```
                        2
                        3
                        4  ***     TEST47  -  FLOPPY DISK DIAGNOSTIC.
                        5  *
                        6  *        J. G. L.,  11/11/77
                        7  *
                        8  *       FOR HEATH COMPANY
                        9  *       COPYRIGHT HEATH COMPANY, 1977, 1979
                       10  *
                       11  *            G. C.,      78/09   Maintenence release
                       12  *                        79/04   Renamed *TEST* from *TEST17*
                       13  *            W. Z.,      80/07   Renamed *TEST17* from *TEST*
                       14  *                                Features added.
                       15  *            W. Z.,      80/07   Modified to do tests for H47,
                       16  *                                Modified version call *TEST47*.
                       17  *
                       18
                       19  ***     TEST47  -  FLOPPY DISK DIAGNOSTIC.
                       20  *
                       21  *       THIS DIAGNOSTIC RUNS STAND ALONE, AFTER BEING LOADED VIA
                       22  *       HDOS.  NO HDOS OVERLAY ROUTINES ARE USED, AND TEST EXITS TO
                       23  *       THE ROM BOOT.
                       24  *
                       25  *       THE USER IS GIVEN THESE OPTIONS:
                       26  *
                       27  *       D - PERFORM GENERAL DRIVE DIAGNOSTIC
                       28  *       M - PERFORM MEDIA CHECK
                       29  *       E - EXIT AND RE-BOOT THE OPERATING SYSTEM
                       30  *       C - CLEAN DRIVE HEAD                                /071080/
                       31  *       A - ALIGN DRIVE HEAD
                       32  *       R - HARDCOPY REPORT                                 /071080/
                       33  *       U - UNIT SELECT
                       34  *
                       35  *       ANY DIAGNOSTIC CAN BE ABORTED PREMATURELY VIA A CTL-C.
                       36
                       37
                       38
000.001                39  .DEBUG. EQU     1               NOT IN DEBUG MODE
```

```
                        41
000.000                 42          XTEXT   MTR
```

```
                        45X **      MTR - PAM/8 EQUIVALENCES.
                        46X *
                        47X *       THIS DECK CONTAINS SYMBOLIC DEFINITIONS USED TO
                        48X *       MAKE USE OF THE PAM/8 CODE AND CONTROL BYTES.


                        50X **      IO PORTS
                        51X
    000.360             52X IP.PAD  EQU     360Q        PAD INPUT PORT
    000.360             53X OP.CTL  EQU     360Q        CONTROL OUTPUT PORT
    000.360             54X OP.DIG  EQU     360Q        DIGIT SELECT OUTPUT PORT
    000.361             55X OP.SEG  EQU     361Q        SEGMENT SELECT OUTPUT PORT
    000.362             56X IP.CON  EQU     362Q        H-88/H-89/HA-8-8 Configuration  /80.07.sc/
    000.362             57X OP2.CTL EQU     362Q        H-88/H-89/HA-8-8 Control Port   /80.07.sc/


                        59X **      FRONT PANEL CONTROL BITS.                           /80.07.sc/
                        60X *
                        61X *       CB.*  set in OP.CTL
                        62X *       CB2,* set in OP2.CTL
                        63X *
                        64X
    000.020             65X CB.SSI  EQU     00010000B   SINGLE STEP INTERRUPT
    000.040             66X CB.MTL  EQU     00100000B   MONITOR LIGHT
    000.100             67X CB.CLI  EQU     01000000B   CLOCK INTERRUPT ENABLE
    000.200             68X CB.SPK  EQU     10000000B   SPEAKER ENABLE
                        69X
    000.001             70X CB2.SSI EQU     00000001B   Single Step Interrupt
    000.002             71X CB2.CLI EQU     00000010B   Clock Interrupt Enable
    000.040             72X CB2.ORG EQU     00100000B   ORG 0 Select
    000.100             73X CB2.SID EQU     01000000B   Side 1 Select


                        75X **      Secondary Control Bits
                        76X


                        78X **      MONITOR MODE FLAGS.
                        79X
    000.000             80X DM.MR   EQU     0           MEMORY READ
    000.001             81X DM.MW   EQU     1           MEMORY WRITE
    000.002             82X DM.RR   EQU     2           REGISTER READ
    000.003             83X DM.RW   EQU     3           REGISTER WRITE
```

```
                        85X **      USER OPTION BITS.
                        86X *
                        87X *       THESE BITS ARE SET IN CELL .MFLAG.
                        88X
000.200                 89X UO.HLT  EQU     10000000B       DISABLE HALT PROCESSING
000.100                 90X UO.NFR  EQU     CB.CLI          NO REFRESH OF FRONT PANEL
000.002                 91X UO.DDU  EQU     00000010B       DISABLE DISPLAY UPDATE
000.001                 92X UO.CLK  EQU     00000001B       ALLOW PRIVATE INTERRUPT PROCESSING

                        94X **      MONITOR IDENTIFICATION FLAGS
                        95X *
                        96X *       THESE BYTES IDENTIFY THE ROM MONITOR.
                        97X *       THEY ARE THE VARIOUS VALUES OF LOCATION .IDENT
                        98X
000.021                 99X M.PAM8  EQU     021Q            'LXI' INSTRUCTION AT 000.000 IN PAM-8
000.303                100X M.FOX   EQU     303Q            'JMP' INSTRUCTION AT 000.000 IN FOX ROM

                        102X **     Configuration Flags                                 /80.07.sc/
                        103X *
                        104X *      These bits are read in IP.CON.
                        105X *
                        106X
000.003                 107X CN.174M EQU    00000011B       Port 1740 Device-Type Mask
000.014                 108X CN.170M EQU    00001100B       Port 1700 Device-Type Mask
000.020                 109X CN.PRI  EQU    00010000B       Primary/Secondary:  1=>primary == 170Q
000.040                 110X CN.MEM  EQU    00100000B       Memory Test/Normal Switch:  0=>Test; 1=>Normal
000.100                 111X CN.BAU  EQU    01000000B       Baud Rate:  0=>9600; 1=>19,200
000.200                 112X CN.ABO  EQU    10000000B       Auto-Boot:  1=>Auto-Boot
                        113X
000.000                 114X CND.H17 EQU    00B             H-17 Disk,              Valid only in CN.174M
000.000                 115X CND.ND1 EQU    00B             No Device Installed,    Valid only in CN.170M
000.001                 116X CND.H47 EQU    01B             H-47 Disk

                        118X **     ROUTINE ENTRY POINTS.
                        119X *
                        120X
000.000                 121X .IDENT  EQU    0000A           IDENTIFICATION LOCATION
000.053                 122X .DLY    EQU    0053A           DELAY
001.267                 123X .LOAD   EQU    1267A           TAPE LOAD
001.374                 124X .DUMP   EQU    1374A           TAPE DUMP
002.136                 125X .ALARM  EQU    2136A           ALARM ROUTINE
002.140                 126X .HORN   EQU    2140A           HORN
002.172                 127X .CTC    EQU    2172A           CHECK TAPE CHECKSUM
002.205                 128X .TPERR  EQU    2205A           TAPE ERROR ROUTINE
002.264                 129X .PCHL   EQU    2264A           PCHL INSTRUCTION
002.265                 130X .SRS    EQU    2265A           SCAN RECORD START
002.325                 131X .RNP    EQU    2325A           READ NEXT PAIR
002.331                 132X .RNB    EQU    2331A           READ NEXT BYTE
```

```
                    133X .CRC    EQU     2347A           CRC-16 CALCULATOR
                    134X .WNP    EQU     3017A           WRITE NEXT PAIR
                    135X .WNB    EQU     3024A           WRITE NEXT BYTE
                    136X .DOD    EQU     3122A           DECODE FOR OCTAL DISPLAY
                    137X .RCK    EQU     3260A           READ CONSOLE KEYSET
                    138X .DODA   EQU     3356A           SEGMENT CODE TABLE


                    140X **      RAM CELLS USED BY H8MTR.
                    141X *
                    142X
                    143X .START  EQU     40000A          START DUMP ADDRESS
                    144X .IOWRK  EQU     40002A          IN OR OUT INSTRUCTION
                    145X .REGI   EQU     40005A          DISPLAYED REGISTER INDEX
                    146X .DSPROT EQU     40006A          PERIOD FLAG BYTE
                    147X .DSPMOD EQU     40007A          DISPLAY MODE
                    148X .MFLAG  EQU     40010A          USER OPTION BYTE
                    149X .CTLFLG EQU     40011A          PANEL CONTROL BYTE
                    150X .ALEDS  EQU     40013A          ABUSS LEDS
                    151X .DLEDS  EQU     40021A          DBUSS LEDS
                    152X .ABUSS  EQU     40024A          ABUSS REGISTER
                    153X .CRCSUM EQU     40027A          CRCSUM WORD
                    154X .TPERRX EQU     40031A          TAPE ERROR EXIT VECTOR
                    155X .TICCNT EQU     40033A          CLOCK TICK COUNTER
                    156X .REGPTR EQU     40035A          REGISTER POINTER
                    157X .UIVEC  EQU     40037A          USER INTERRUPT VECTORS
                    158X .NMIRET EQU     40064A          H88/H89 NMI Return Address      /80.07.sc/
                    159X .CTL2FL EQU     40066A          OP2.CTL Control Byte            /80.07.sc/
                    160          XTEXT   ASCII


                    162X **      ASCII CHARACTER EQUIVALENCES.
                    163X
                    164X CR      EQU     13              CARRIAGE RETURN
                    165X LF      EQU     10              LINE FEED
                    166X NULL    EQU     200Q            PAD CHARACTER
                    167X NUL2    EQU     0
                    168X BELL    EQU     7               BELL CHARACTER
                    169X RUBOUT  EQU     177Q
                    170X BKSP    EQU     10Q             CTL-H
                    171X C.SYN   EQU     26Q             SYNC
                    172X C.STX   EQU     2               STX
                    173X QUOTE   EQU     47Q
                    174X TAB     EQU     11Q
                    175X ESC     EQU     33Q
                    176X NL      EQU     12Q             NEW LINE (HDOS SYSTEMS)
                    177X ENL     EQU     NL+200Q         NL + END-OF-LINE-FLAG
                    178X FF      EQU     14Q             FORM FEED
                    179X CTLA    EQU     01Q             CTL-A
                    180X CTLB    EQU     02Q             CTL-B
                    181X CTLC    EQU     03Q             CTL-C
                    182X CTLD    EQU     04Q             CTL-D
                    183X CTLO    EQU     17Q             CTL-O
                    184X CTLP    EQU     20Q             CTL-P
                    185X CTLQ    EQU     21Q             CTL-Q
```

```
000.023                 186X CTLS   EQU     23Q             CTL-S
000.032                 187X CTLZ   EQU     32Q             CTL-Z
000.000                 188         XTEXT   H47PAR

                        190X **     H47PAR  -  H47 Parameters
                        191X *
                        192X
000.015                 193X NSPTS  EQU     13              Sectors/Track Single Density    [1-13]
000.032                 194X NSPTD  EQU     26              Sectors/Track Double Density    [1-26]
                        195X
000.115                 196X NTRK   EQU     76+1            Number of Tracks                [0-76]
000.000                 197         XTEXT   H47DEF

                        199X **     H47DEF  -  H47 Constant Definitions
                        200X *

                        202X *      Z-80 INSTRUCTIONS
                        203X
242.355                 204X M.INI  EQU     10100010B*256+11101101B         INI   INSTRUCTION
243.355                 205X M.OUTI EQU     10100011B*256+11101101B         OUTI  INSTRUCTION

                        207X **     DISK INTERFACE CONSTANTS
                        208X *
                        209X
000.170                 210X D.STA  EQU     170Q                INTERFACE STATUS PORT
000.171                 211X D.DAT  EQU     D.STA+1             DATA PORT
                        212X
000.001                 213X S.ERR  EQU     00000001B           ERROR BIT
000.040                 214X S.DON  EQU     00100000B           DONE
000.100                 215X S.IEN  EQU     01000000B           INTERRUPT ENABLE
000.200                 216X S.DTR  EQU     10000000B           DATA TERMINAL REQUEST
                        217X
000.002                 218X S.SW0  EQU     00000010B           DIP SWITCH:  0
000.004                 219X S.SW1  EQU     00000100B           DIP SWITCH:  1
000.010                 220X S.SW2  EQU     00001000B           DIP SWITCH:  2
000.020                 221X S.SW3  EQU     00010000B           DIP SWITCH:  3
                        222X
000.002                 223X W.RES  EQU     00000010B           RESET COMMAND
```

```
                        225X **     STATUS BYTE FLAGS
                        226X *
                        227X
000.200                 228X SB.UNR EQU 10000000B   UNIT NOT READY
000.100                 229X SB.WPD EQU 01000000B   WRITE PROTECTED DRIVE
000.040                 230X SB.DLD EQU 00100000B   DELETED DATA
000.020                 231X SB.NRF EQU 00010000B   NO RECORD FOUND
000.010                 232X SB.CRC EQU 00001000B   CRC ERROR
000.004                 233X SB.LTD EQU 00000100B   LATE DATA
000.002                 234X SB.ILC EQU 00000010B   ILLEGAL COMMAND
000.001                 235X SB.BTO EQU 00000001B   BAD TRACK OVERFLOW


                        237X **     AUXILLARY STATUS BYTE FLAGS
                        238X *
                        239X
000.100                 240X AS.ODD EQU 01000000B   TRACK 0 DOUBLE DENSITY
000.040                 241X AS.1DD EQU 00100000B   TRACK 1-76 DOUBLE DENSITY
000.020                 242X AS.S1A EQU 00010000B   SIDE 1 AVAILABLE
000.003                 243X AS.SLM EQU 00000011B   SECTOR LENGTH MASK


                        245X **     DISK COMMANDS
                        246X *
                        247X
000.000                 248X        ORG 0
000.000                 249X DD.BOOT DS  1          BOOT
000.001                 250X DD.RST  DS  1          READ STATUS
000.002                 251X DD.RAS  DS  1          READ AUX. STATUS
000.003                 252X DD.LSC  DS  1          LOAD SECTOR COUNT
000.004                 253X DD.RAD  DS  1          READ ADDRESS OF LAST SECTOR ACCESSED
000.005                 254X DD.REA  DS  1          READ SECTORS
000.006                 255X DD.WRI  DS  1          WRITE SECTORS
000.007                 256X DD.REAB DS  1          READ SECTORS BUFFERED
000.010                 257X DD.WRIB DS  1          WRITE SECTORS BUFFERED
000.011                 258X DD.WRD  DS  1          DD.WRI + DELETED
000.012                 259X DD.WRBD DS  1          DD.WRIB + DELETED
000.013                 260X DD.CPY  DS  1          COPY
000.014                 261X DD.FRM0 DS  1          FORMAT IBM SD
000.015                 262X DD.FRM1 DS  1          FORMAT SD
000.016                 263X DD.FRM2 DS  1          FORMAT IBM DD
000.017                 264X DD.FRM3 DS  1          FORMAT DD
000.020                 265X DD.RRDY DS  1          Read Ready (conflict with DD.SPF0)
```

```
                        267X **     Special De-Bus Functions
                        268X *
                        269X
    000.020             270X        ORG     010H
    000.020             271X DD.SPF0 DS     1                   SPECIAL FUNCTION 0
    000.021             272X DD.SPF1 DS     1                   SPECIAL FUNCTION 1
    000.022             273X DD.SPF2 DS     1                   SPECIAL FUNCTION 2
    000.023             274X DD.SPF3 DS     1                   SPECIAL FUNCTION 3
    000.024             275X DD.SPF4 DS     1                   SPECIAL FUNCTION 4
    000.025             276X DD.SPF5 DS     1                   SPECIAL FUNCTION 5


                        278X **     Special Heath Functions
                        279X *
                        280X
    000.200             281X        ORG     080H
    000.200             282X DD.SDC  DS     1                   SET DRIVE CHARACTERISTICS
    000.201             283X DD.ST   DS     1                   SEEK TO TRACK
    000.202             284X DD.DS   DS     1                   DISK STATUS
    000.203             285X DD.RDL  DS     1                   READ LOGICAL
    000.204             286X DD.WTL  DS     1                   WRITE LOGICAL
    000.205             287X DD.RDBL DS     1                   READ BUFFERED LOGICAL
    000.206             288X DD.WTBL DS     1                   WRITE BUFFERED LOGICAL
    000.207             289X DD.WTDL DS     1                   WRITE DELETED DATA LOGICAL
    000.210             290X DD.WDLB DS     1                   WRITE BUFFERED DELETED DATA LOGICAL


                        292X **     Useful Flags
                        293X *
                        294X
    000.000             295X UNT.0  EQU     00000000B           Unit: 0
    000.040             296X UNT.1  EQU     00100000B           Unit: 1
    000.100             297X UNT.2  EQU     01000000B           Unit: 2
    000.140             298X UNT.3  EQU     01100000B           Unit: 3
                        299X
    000.140             300X UNT.M  EQU     UNT.0!UNT.1!UNT.2!UNT.3 Unit Mask
                        301X
                        302X
                        303X
    000.000             304X SID.0  EQU     00000000B           Side: 0
    000.200             305X SID.1  EQU     10000000B           Side: 1
                        306X
    000.200             307X SID.M  EQU     SID.0!SID.1         Side Mask
                        308X
                        309X
                        310X
    000.037             311X SEC.M  EQU     00011111B           Track Mask
                        312X
                        313X
                        314X
    004.000             315X SSIZ.M EQU     1024                Maximum Sector Size
                        316X
```

```
                        317X
                        318X *C.128  EQU     128
                        319X *C.256  EQU     256
                        320X *C.24   EQU     24
    000.211             321          XTEXT   DIRDEF


                        323X **      DIRECTORY ENTRY FORMAT.
                        324X
    000.000             325X         ORG     0
                        326X
                        327X
    000.377             328X DF.EMP  EQU     377Q            FLAGS ENTRY EMPTY
    000.376             329X DF.CLR  EQU     376Q            FLAGS ENTRY EMPTY, REST OF DIR ALSO CLEAR
                        330X
    000.000             331X DIR.NAM DS      8               NAME
    000.010             332X DIR.EXT DS      3               EXTENSION
    000.013             333X DIR.PRO DS      1               PROJECT
    000.014             334X DIR.VER DS      1               VERSION
    000.015             335X DIR1DL  EQU     *               FILE IDENTIFICATION LENGTH
                        336X
    000.015             337X DIR.CLU DS      1               CLUSTER FACTOR
    000.016             338X DIR.FLG DS      1               FLAGS
    000.017             339X         DS      1               RESERVED
    000.020             340X DIR.FGN DS      1               FIRST GROUP NUMBER
    000.021             341X DIR.LGN DS      1               LAST GROUP NUMBER
    000.022             342X DIR.LSI DS      1               LAST SECTOR INDEX (IN LAST GROUP)
    000.023             343X DIR.CRD DS      2               CREATION DATE
    000.025             344X DIR.ALD DS      2               LAST ALTERATION DATE
                        345X
    000.027             346X DIRELEN EQU     *               DIRECTORY ENTRY LENGTH
    000.027             347          XTEXT   DDFDEF


                        349X **      DIRECTORY DEVICE FORMAT DEFINITION.                    /80.09.SC/
                        350X *
                        351X *       Modified:      Sep-80
                        352X *                      No longer require 2 sectors per group
                        353X *                       Reserved Group Table dynamically allocated
                        354X *
                        355X
    000.000             356X         ORG     0
                        357X
    000.000             358X DDF.BOO DS      9               2K BOOT PROGRAM
    000.011             359X DDF.BOL EQU     *               LENGTH OF BOOT
    000.011             360X DDF.LAB DS      1               LABEL SECTOR
    000.012             361X DDF.USR DS      0               BEGINNING OF OPEN SPACE
    000.012             362          XTEXT   LABDEF
```

```
                        364X **       DISK LABEL SECTOR FORMATS.
                        365X
000.000                 366X          ORG     0
000.000                 367X LAB.SER  DS      1               SERIAL NUMBER OF VOLUME
000.001                 368X LAB.IND  DS      2               INITIALIZATION DATE
000.003                 369X LAB.DIS  DS      2               SECTOR NUMBER OF 1ST DIRECTORY SECTOR
000.005                 370X LAB.GRT  DS      2               INDEX OF GRT SECTOR
000.007                 371X LAB.SPG  DS      1               SECTORS PER GROUP
                        372X
000.000                 373X LAB.DAT  EQU     0               DATA VOLUME ONLY
000.001                 374X LAB.SYS  EQU     1               SYSTEM VOLUME
000.002                 375X LAB.NOD  EQU     2               => LAB.NOD MEANS VOLUME HAS NO DIRECTORY
                        376X
000.010                 377X LAB.VLT  DS      1               VOLUME TYPE
000.011                 378X LAB.VER  DS      1               VERSION OF INIT17 THAT INITED DISK
                        379X
000.012                 380X LAB.RGT  DS      2               RGT sector number                       /80.06.sc/
                        381X
000.014                 382X LAB.VPR  EQU     *               Volume dependant data                   /80.05.sc/
000.014                 383X LAB.SIZ  DS      2                Volume Size (Bytes/256)                /80.05.sc/
000.016                 384X LAB.PSS  DS      2                Physical Sector Size                   /80.05.sc/
000.020                 385X LAB.VFL  DS      1                Volume dependant Flags                 /80.09.sc/
000.001                 386X VFL.NSD  EQU     00000001B         Number of Sides: 1 => 2               /80.09.sc/
000.005                 387X LAB.VPL  EQU     *-LAB.VPR       Length of volume dependant data         /80.05.sc/
                        388X
000.000                 389X          ERRMI   5-LAB.VPL                                               /80.05.sc/
000.021                 390X          DS      5-LAB.VPL       Reserved                                /80.05.sc/
                        391X
000.021                 392X LAB.LAB  DS      60              LABEL
000.074                 393X LAB.LBL  EQU     *-LAB.LAB       LABEL LENGTH
000.115                 394X          DS      2               Reserved for 0 bytes                    /80.09.sc/
                        395X
000.117                 396X LAB.AUX  EQU     *               Auxiliary Data                          /80.09.sc/
000.117                 397X LAB.SPT  DS      1                Sectors per Track                      /80.09.sc/
000.001                 398X LAB.AXL  EQU     *-LAB.AUX       Length of AUX. Data                     /80.09.sc/
000.120                 399           XTEXT   OVLDEF



                        401X **       OVERLAY TABLE ENTRYS.
                        402X
000.000                 403X          ORG     0
                        404X
000.000                 405X OVL.COD  DS      2               FIRST SECTOR OF OVERLAY CODE
000.002                 406X OVL.SIZ  DS      2               OVERLAY SIZE
000.004                 407X OVL.ENT  DS      2               OVERLAY ENTRY POINT
000.006                 408X OVL.FLB  DS      1               OVERLAY FLAG BYTE
000.007                 409X          DS      1               DUMMY BYTE TO ROUND TABLE SIZE UP TO 8
000.010                 410X OVL.ENS  EQU     *               OVERLAY ENTRY SIZE
                        411X
                        412X *        OVERLAY INDICES
                        413X
000.000                 414X          ORG     0
                        415X
000.000                 416X OVL0     DS      1
```

```
000.001                 417X OVL1   DS      1
000.002                 418         XTEXT   DDDEF


                        420X **     DEVICE DRIVER COMMUNICATION FLAGS.
                        421X *
                        422X
000.000                 423X        ORG     0
                        424X
000.000                 425X DC.REA DS      1           READ
000.001                 426X DC.WRI DS      1           WRITE
000.002                 427X DC.RER DS      1           READ REGARDLESS
000.003                 428X DC.OPR DS      1           OPEN FOR READ
000.004                 429X DC.OPW DS      1           OPEN FOR WRITE
000.005                 430X DC.OPU DS      1           OPEN FOR UPDATE
000.006                 431X DC.CLO DS      1           CLOSE
000.007                 432X DC.ABT DS      1           ABORT
000.010                 433X DC.MOU DS      1           MOUNT DEVICE
000.011                 434X DC.LOD DS      1           LOAD DEVICE DRIVER
000.012                 435X DC.RDY DS      1           Device Ready                    /80.04.GC/
000.013                 436X DC.MAX DS      1           MAXIMUM ENTRY INDEX
000.014                 437         XTEXT   HOSEQU


                        439X **     HDOS SYSTEM EQUIVALENCES.
                        440X *
                        441X
024.000                 442X S.GRT0 EQU     24000A      SYSTEM AREA FOR  GRT0
025.000                 443X S.GRT1 EQU     25000A      SYSTEM AREA FOR  GRT1
026.000                 444X S.GRT2 EQU     26000A      SYSTEM AREA FOR  GRT2
                        445X
030.000                 446X ROMBOOT EQU    30000A      ROM BOOT ENTRY
                        447X
040.100                 448X        ORG     40100A      FREE SPACE FROM PAM-8
                        449X
040.100                 450X        DS      8           JUMP TO SYSTEM EXIT
040.110                 451X D.CON  DS      16          DISK CONSTANTS
040.130                 452X SYDD   EQU     *           SYSTEM DISK ENTRY POINT
040.130                 453X D.VEC  DS      24*3        SYSTEM ROM ENTRY VECTORS
040.240                 454X D.RAM  DS      31          SYSTEM ROM WORK AREA
040.277                 455X S.VAL  DS      36          SYSTEM VALUES
040.343                 456X S.INT  DS      115         SYSTEM INTERNAL WORK AREAS
041.126                 457X        DS      16
041.146                 458X S.SOVR DS      2           STACK OVERFLOW WARNING
041.150                 459X        DS      42200A-*    SYSTEM STACK
001.032                 460X STACKL EQU     *-S.SOVR    STACK SIZE
                        461X
042.200                 462X STACK  EQU     *           LWA+1 SYSTEM STACK
042.200                 463X USERFWA EQU    *           USER FWA
042.200                 464         XTEXT   ESVAL
```

```
                    466X **     S.VAL - SYSTEM VALUE DEFINTIONS.
                    467X *
                    468X *      THESE VALUES ARE SET AND MAINTAINED BY THE SYSTEM.
                    469X *
                    470X *      THE DECK HOSEQU MUST BE MODIFIED WHEN THIS IS MODIFIED.
                    471X
                    472X
040.277             473X        ORG     S.VAL
                    474X
040.277             475X S.DATE DS      9               SYSTEM DATE (IN ASCII)
040.310             476X S.DATC DS      2               CODED DATE
040.312             477X S.TIME DS      4               TIME FROM MIDNIGHT (IN TICS)
040.316             478X S.HIMEM DS     2               HARDWARE HIGH MEMORY ADRESS+1
                    479X
040.320             480X S.SYSM DS      2               FWA RESIDENT SYSTEM
                    481X
040.322             482X S.USRM DS      2               LWA USER MEMORY
                    483X
040.324             484X S.OMAX DS      2               MAX OVERLAY SIZE FOR SYSTEM
                    485X
                    486X
                    487X **     THE FOLLOWING FIVE CELLS SHOULD BE MODIFIED/READ ONLY VIA THE .CONSL SYSCALL
                    488X
000.200             489X CSL.ECH EQU    10000000B       SUPPRESS ECHO
000.004             490X CSL.RAW EQU    00000100B       Raw Mode I/O                    /80.09.sc/
000.002             491X CSL.WRP EQU    00000010B       WRAP LINES AT WIDTH
000.001             492X CSL.CHR EQU    00000001B       OPERATE IN CHARACTER MODE
                    493X
000.000             494X I.CSLMD EQU    0               S.CSLMD IS FIRST BYTE
040.326             495X S.CSLMD DS     1               CONSOLE MODE
                    496X
000.200             497X CTP.BKS EQU    10000000B       TERMINAL PROCESSES BACKSPACES
000.100             498X CTP.FF EQU     01000000B       Terminal Processes Form-Feed    /80.09.sc/
000.040             499X CTP.MLI EQU    00100000B       MAP LOWER CASE TO UPPER ON INPUT
000.020             500X CTP.MLO EQU    00010000B       MAP LOWER CASE TO UPPER ON OUTPUT
000.010             501X CTP.2SB EQU    00001000B       TERMINAL NEEDS TWO STOP BITS
000.002             502X CTP.BKM EQU    00000010B       MAP BKSP (UPON INPUT) TO RUBOUT
000.001             503X CTP.TAB EQU    00000001B       TERMINAL SUPPORTS TAB CHARACTERS
                    504X
000.001             505X I.CONTY EQU    1               S.CONTY IS 2ND BYTE
000.000             506X        ERRNZ   *-S.CSLMD-I.CONTY
040.327             507X S.CONTY DS     1               CONSOLE TYPE FLAGS
000.002             508X I.CUSOR EQU    2               S.CUSOR IS 3RD BYTE
000.000             509X        ERRNZ   *-S.CSLMD-I.CUSOR
040.330             510X S.CUSOR DS     1               CURRENT CURSOR POSITION
000.003             511X I.CONW1 EQU    3               S.CONW1 IS 4TH BYTE
000.000             512X        ERRNZ   *-S.CSLMD-I.CONW1
040.331             513X S.CONW1 DS     1               CONSOLE WIDTH
                    514X
000.001             515X CO.FLG EQU     00000001B       CTL-O FLAG
000.200             516X CS.FLG EQU     10000000B       CTL-S FLAG
                    517X
000.004             518X I.CONFL EQU    4               S.CONFL IS 5TH BYTE
000.000             519X        ERRNZ   *-S.CSLMD-I.CONFL
040.332             520X S.CONFL DS     1               CONSOLE FLAGS
                    521X
```

```
040.333                 522X S.CAADR DS      2          ADDRESS FOR ABORT PROCESSING (>256 IF VALID)
040.335                 523X S.CCTAB DS      6          ADDR FOR CTL-A, CTL-B, CTL-C PROCESSING
040.343                 524          XTEXT   ABSDEF


                        526X **      ABS FORMAT EQUIVALENCES.
                        527X
000.000                 528X         ORG     0
                        529X
000.000                 530X ABS.ID  DS      1          377Q = BINARY FILE FLAG
000.001                 531X         DS      1          FILE TYPE (FT.ABS)
000.002                 532X ABS.LDA DS      2          LOAD ADDRESS
000.004                 533X ABS.LEN DS      2          LENGTH OF ENTIRE RECORD
000.006                 534X ABS.ENT DS      2          ENTRY POINT
                        535X
000.010                 536X ABS.COD DS      0          CODE STARTS HERE
000.010                 537          XTEXT   FILDEF


                        539X **      FILDEF - FILE TYPE DEFINITIONS.
                        540X *
                        541X *       DB      377Q,FT.XXX
                        542X
                        543X
000.000                 544X FT.ABS  EQU     0          ABSOLUTE BINARY
000.001                 545X FT.PIC  EQU     1          POSITION INDEPENDANT CODE
000.002                 546X FT.REL  EQU     2          RELOCATABLE CODE
000.003                 547X FT.BAC  EQU     3          COMPILED BASIC CODE
000.010                 548          XTEXT   DEVDEF


                        550X **      DEVICE TABLE ENTRYS.
                        551X
000.000                 552X         ORG     0
                        553X
000.000                 554X DEV.NAM DS      2          DEVICE NAME
000.000                 555X DV.EL   EQU     00000000B  END OF DEVICE LIST FLAG
000.001                 556X DV.NU   EQU     00000001B  DEVICE ENTRY NOT IN USE
                        557X
000.002                 558X DEV.RES DS      1          DRIVER RESIDENSE CODE
000.001                 559X DR.IM   EQU     00000001B  DRIVER IN MEMORY
000.002                 560X DR.PR   EQU     00000010B  DRIVER PERMINANTLY RESIDENT
                        561X
000.003                 562X DEV.JMP DS      1          JMP TO PROCESSOR
000.004                 563X DEV.DDA DS      2          DRIVER ADDRESS
000.006                 564X DEV.FLG DS      1          FLAG BYTE
000.001                 565X DT.DD   EQU     00000001B  DIRECTORY DEVICE
000.002                 566X DT.CR   EQU     00000010B  CAPABLE OF READ OPERATION
000.004                 567X DT.CW   EQU     00000100B  CAPABLE OF WRITE OPERATION
000.010                 568X DT.RN   EQU     00001000B  Capable of random access        /80.02.sc/
000.020                 569X DT.CH   EQU     00010000B  Capable of Character mode       /80.02.sc/
```

```
                    570X
000.007             571X DEV.MUM DS      1          MOUNTED UNIT MASK
000.010             572X DEV.MNU DS      1          MAXIMUM NUMBER OF UNITS
000.011             573X DEV.UNT DS      2          ADDRESS OF UNIT SPECIFIC DATA TABLE
                    574X
000.013             575X DEV.DVL DS      2          DRIVER BYTE LENGTH
000.015             576X DEV.DVG DS      1          DRIVER ROUTINE GROUP ADDRESS
                    577X
000.016             578X DEVELEN EQU     *          DEVICE TABLE ENTRY LENGTH


                    580X **      UNIT SPECIFIC DEVICE DATA TABLE ENTRIES
                    581X
000.000             582X         ORG     0
                    583X
000.000             584X UNT.FLG DS      1          UNIT SPECIFIC  *DEV.FLG*
000.001             585X UNT.SPG DS      1          Sectors Per Group                /80.04.6C/
000.002             586X UNT.GRT DS      2          ADDRESS OF GROUP RESERVATION TABLE (IF DT.DD)
000.004             587X UNT.GTS DS      2          GRT SECTOR NUMBER
000.006             588X UNT.DIS DS      2          DIRECTORY FIRST SECTOR NUMBER
                    589X
000.010             590X UNT.SIZ EQU     *          SIZE OF UNIT SPECIFIC DATA TABLE PER UNIT
000.010             591          XTEXT   ESINT


                    593X **      S.INT - SYSTEM INTERNAL WORKAREA DEFINITIONS.
                    594X *
                    595X *       THESE CELLS ARE REFERENCED BY OVERLAYS AND MAIN CODE, AND
                    596X *       MUST THEREFORE RESIDE IN FIXED LOW MEMORY.
                    597X
                    598X
040.343             599X         ORG     S.INT
                    600X
                    601X **      CONSOLE STATUS FLAGS
                    602X
040.343             603X S.CDB   DS      1          CONSOLE DESCRIPTOR BYTE
000.000             604X CDB.H85 EQU     00000000B
000.001             605X CDB.H84 EQU     00000001B  =0 IF H8-5, =1 IF H8-4
040.344             606X S.BAUD  DS      2          [0-14]  H8-4 BAUD RATE, =0 IF H8-5
                    607X *                          [15]    =1 IF BAUD RATE => 2 STOP BITS
                    608X
                    609X **      TABLE ADDRESS WORDS
                    610X
040.346             611X S.DLINK DS      2          ADDRESS OF DATA IN HDOS CODE
040.350             612X S.OFWA  DS      2          FWA  OVERLAY  TABLE
040.352             613X S.CFWA  DS      2          FWA  CHANNEL  TABLE
040.354             614X S.DFWA  DS      2          FWA  DEVICE   TABLE
040.356             615X S.RFWA  DS      2          FWA  RESIDENT HDOS CODE
                    616X
                    617X **      DEVICE DRIVER DELAYED LOAD FLAGS
                    618X
040.360             619X S.DDLDA DS      2          DRIVER LOAD ADDRESS (HIGH BYTE=0 IF NO LOAD PENDING)
```

```
     040.362                620X S.DDLEN DS      2              CODE LENGTH IN BYTES
     040.364                621X S.DDGRP DS      1              GROUP NUMBER FOR DRIVER
     040.365                622X         DS      1              HOLD PLACE
                            623X *S,DDSEC        DS      2              SECTOR NUMBER FOR DRIVER (*OBSOLETE!*)
     040.366                624X S.DDDTA DS      2              DEVICE'S ADDRESS IN DEVLST +DEV.RES
     040.370                625X S.DDOPC DS      1              OPEN OPCODE PENDEDING
                            626X
                            627X **      OVERLAY MANAGEMENT FLAGS
                            628X
     000.001                629X OVL.IN  EQU     00000001B      IN MEMORY
     000.002                630X OVL.RES EQU     00000010B      PERMINANTLY RESIDENT
     000.014                631X OVL.NUM EQU     00001100B      OVERLAY NUMBER MASK
     000.200                632X OVL.UCS EQU     10000000B      USER CODE SWAPPED FOR OVERLAY
                            633X
     040.371                634X S.OVLFL DS      1              OVERLAY FLAG
     040.372                635X S.UCSF  DS      2              FWA SWAPPED USER CODE
     040.374                636X S.UCSL  DS      2              LENGTH SWAPPED USER CODE
     040.376                637X S.OVLS  DS      2              SIZE OF OVERLAY CODE
     041.000                638X S.OVLE  DS      2              ENTRY POINT OF OVERLAY CODE
                            639X
     041.002                640X S.SSN   DS      2              SWAP AREA SECTOR NUMBER
     041.004                641X S.OSN   DS      2              OVERLAY SECTOR NUMBER
                            642X
                            643X *       SYSCALL PROCESSING WORK AREAS
                            644X
     041.006                645X S.CACC  DS      1              (ACC) UPON SYSCALL
     041.007                646X S.CODE  DS      1              SYSCALL INDEX IN PROGRESS
                            647X
                            648X *       JUMPS TO ROUTINES IN RESIDENT HDOS CODE
                            649X
     041.010                650X S.JUMPS DS      0              START OF DUMP VECTORS
     041.010                651X S.SDD   DS      3              JUMP TO STAND-IN DEVICE DRIVER
     041.013                652X S.FASER DS      3              JUMP TO FATSERR (FATAL SYSTEM ERROR)
     041.016                653X S.DIREA DS      3              JUMP TO DIREAD (DISK FILE READ)
     041.021                654X S.FCI   DS      3              JUMP TO FCI (FETCH CHANNEL INFO)
     041.024                655X S.SCI   DS      3              JUMP TO SCI (STORE CHANNEL INFO)
     041.027                656X S.GUP   DS      3              JUMP TO GUP (GET UNIT POINTER)
                            657X
     041.032                658X S.MOUNT DS      1              <>0 IF THE SYSTEM DISK IS MOUNTED
     041.033                659X S.DCS   DS      1              DEFAULT CLUSTER SIZE-1
                            660X
     041.034                661X S.BOOTF DS      1              BOOT FLAGS
     000.001                662X BOOT.P  EQU     00000001B      EXECUTE PROLOGUE UPON BOOTUP
                            663X
                            664X *       STACK VALUE SAVED FOR OVERLAY SYSCALLS
                            665X
     041.035                666X S.OVSTK DS      2              VALUE OF SP UPON SYSCALLS USING OVERLAY
                            667X
     041.037                668X         DS      1              RESERVED
```

```
                        670X **      ACTIVE I/O AREA.
                        671X *
                        672X *       THE AIO.XXX AREA CONTAINS INFORMATION ABOUT THE I/O OPERATION
                        673X *       CURRENTLY BEING PERFORMED. THE INFORMATION IS OBTAINED FROM
                        674X *       THE CHANNEL TABLE, AND WILL BE RESTORED THERE WHEN DONE.
                        675X *
                        676X *       NORMALLY, THE AIO.XXX INFORMATION WOULD BE OBTAINED DIRECTLY
                        677X *       FROM VARIOUS SYSTEM TABLES VIA POINTER REGISTERS. SINCE THE
                        678X *       8080 HAS NO GOOD INDEXED ADDRESSING, THE DATA IS MANUALLY
                        679X *       COPIED INTO THE AIO.XXX CELLS BEFORE PROCESSING, AND
                        680X *       BACKDATED AFTER PROCESSING.
                        681X
    041,040             682X AIO.VEC DS      3            JUMP INSTRUCTION
    041.041             683X AIO.DDA EQU     *-2          DEVICE DRIVER ADDRESS
    041,043             684X AIO.FLG DS      1            FLAG BYTE
    041.044             685X AIO.GRT DS      2            ADDRESS OF GROUP RESERV TABLE
    041,046             686X AIO.SPG DS      1            SECTORS PER GROUP
    041.047             687X AIO.CGN DS      1            CURRENT GROUP NUMBER
    041,050             688X AIO.CSI DS      1            CURRENT SECTOR INDEX
    041.051             689X AIO.LGN DS      1            LAST GROUP NUMBER
    041,052             690X AIO.LSI DS      1            LAST SECTOR INDEX
    041.053             691X AIO.DTA DS      2            DEVICE TABLE ADDRESS
    041,055             692X AIO.DES DS      2            DIRECTORY SECTOR
    041.057             693X AIO.DEV DS      2            DEVICE CODE
    041,061             694X AIO.UNI DS      1            UNIT NUMBER (0-9)
                        695X
    041,062             696X AIO.DIR DS      DIRELEN      DIRECTORY ENTRY
                        697X
    041,111             698X AIO.CNT DS      1            SECTOR COUNT
    041.112             699X AIO.EOM DS      1            END OF MEDIA FLAG
    041,113             700X AIO.EOF DS      1            END OF FILE FLAG
    041.114             701X AIO.TFP DS      2            TEMP FILE POINTERS
    041,116             702X AIO.CHA DS      2            ADDRESS OF CHANNEL BLOCK (IOC.DDA)


    041.120             704X S.BDA   DS      1            Boot Device Address (Setup by ROM) /80.09.sc/
    041,121             705X S.SCR   DS      2            SYSTEM SCRATCH AREA ADDRESS
    041.123             706          XTEXT   ECDEF


                        708X **      ERROR CODE DEFINITIONS.
                        709X
    000.000             710X         ORG     0
    000,000             711X         DS      1            NO ERROR #0
    000.001             712X EC.EOF  DS      1            END OF FILE
    000,002             713X EC.EOM  DS      1            END OF MEDIA
    000.003             714X EC.ILC  DS      1            ILLEGAL SYSCALL CODE
    000,004             715X EC.CNA  DS      1            CHANNEL NOT AVAILABLE
    000.005             716X EC.DNS  DS      1            DEVICE NOT SUITABLE
    000,006             717X EC.IDN  DS      1            ILLEGAL DEVICE NAME
    000.007             718X EC.IFN  DS      1            ILLEGAL FILE NAME
    000,010             719X EC.NRD  DS      1            NO ROOM FOR DEVICE DRIVER
    000.011             720X EC.FNO  DS      1            CHANNEL NOT OPEN
```

```
000.012                721X EC.ILR  DS      1          ILLEGAL REQUEST
000.013                722X EC.FUC  DS      1          FILE USAGE CONFLICT
000.014                723X EC.FNF  DS      1          FILE NAME NOT FOUND
000.015                724X EC.UND  DS      1          UNKNOWN DEVICE
000.016                725X EC.ICN  DS      1          ILLEGAL CHANNEL NUMBER
000.017                726X EC.DIF  DS      1          DIRECTORY FULL
000.020                727X EC.IFC  DS      1          ILLEGAL FILE CONTENTS
000.021                728X EC.NEM  DS      1          NOT ENOUGH MEMORY
000.022                729X EC.RF   DS      1          READ FAILURE
000.023                730X EC.WF   DS      1          WRITE FAILURE
000.024                731X EC.WPV  DS      1          WRITE PROTECTION VIOLATION
000.025                732X EC.WP   DS      1          DISK WRITE PROTECTED
000.026                733X EC.FAP  DS      1          FILE ALREADY PRESENT
000.027                734X EC.DDA  DS      1          DEVICE DRIVER ABORT
000.030                735X EC.FL   DS      1          FILE LOCKED
000.031                736X EC.FAO  DS      1          FILE ALREADY OPEN
000.032                737X EC.IS   DS      1          ILLEGAL SWITCH
000.033                738X EC.UUN  DS      1          UNKNOWN UNIT NUMBER
000.034                739X EC.FNR  DS      1          FILE NAME REQUIRED
000.035                740X EC.DIW  DS      1          DEVICE IS NOT WRITABLE (OR WRITE LOCKED)
000.036                741X EC.UNA  DS      1          UNIT NOT AVAILABLE
000.037                742X EC.ILV  DS      1          ILLEGAL VALUE
000.040                743X EC.ILO  DS      1          ILLEGAL OPTION
000.041                744X EC.VPM  DS      1          VOLUME PRESENTLY MOUNTED ON DEVICE
000.042                745X EC.NVM  DS      1          NO VOLUME PRESENTLY MOUNTED
000.043                746X EC.FOD  DS      1          FILE OPEN ON DEVICE
000.044                747X EC.NPM  DS      1          NO PROVISIONS MADE FOR REMOUNTING MORE DISKS
000.045                748X EC.DNI  DS      1          DISK NOT INITIALIZED
000.046                749X EC.DNR  DS      1          DISK IS NOT READABLE
000.047                750X EC.DSC  DS      1          DISK STRUCTURE IS CORRUPT
000.050                751X EC.NCV  DS      1          NOT CORRECT VERSION OF HDOS
000.051                752X EC.NOS  DS      1          NO OPERATING SYSTEM MOUNTED
000.052                753X EC.IOI  DS      1          ILLEGAL OVERLAY INDEX
000.053                754X EC.OIL  DS      1          OVERLAY TO LARGE
000.054                755          XTEXT   HOSDEF


                       757X **      HOSDEF - DEFINE HOS PARAMETER.
                       758X *
                       759X
                       760X
000.040                761X VERS    EQU     2*16+0     VERSION 2.0
                       762X
000.377                763X SYSCALL EQU     377Q       SYSCALL INSTRUCTION
                       764X
                       765X
000.000                766X         ORG     0
                       767X
                       768X *       RESIDENT FUNCTIONS
                       769X
000.000                770X .EXIT   DS      1          EXIT (MUST BE FIRST)
000.001                771X .SCIN   DS      1          SCIN
000.002                772X .SCOUT  DS      1          SCOUT
000.003                773X .PRINT  DS      1          PRINT
000.004                774X .READ   DS      1          READ
```

```
000.005                 775X .WRITE  DS      1               WRITE
000.006                 776X .CONSL  DS      1               SET/CLEAR CONSOLE OPTIONS
000.007                 777X .CLRCO  DS      1               CLEAR CONSOLE BUFFER
000.010                 778X .LOADO  DS      1               LOAD AN OVERLAY
000.011                 779X .VERS   DS      1               RETURN HDOS VERSION NUMBER
000.012                 780X .SYSRES DS      1               PRECEDING FUNCTIONS ARE RESIDENT
                        781X
                        782X
                        783X *       *HDOSOVL0.SYS*  FUNCTIONS
                        784X
000.040                 785X         ORG     40A
                        786X
000.040                 787X .LINK   DS      1               LINK  (MUST BE FIRST)
000.041                 788X .CTLC   DS      1               CTL-C
000.042                 789X .OPENR  DS      1               OPENR
000.043                 790X .OPENW  DS      1               OPENW
000.044                 791X .OPENU  DS      1               OPENU
000.045                 792X .OPENC  DS      1               OPENC
000.046                 793X .CLOSE  DS      1               CLOSE
000.047                 794X .POSIT  DS      1               POSITION
000.050                 795X .DELET  DS      1               DELETE
000.051                 796X .RENAM  DS      1               RENAME
000.052                 797X .SETTP  DS      1               SETTOP
000.053                 798X .DECODE DS      1               NAME DECODE
000.054                 799X .NAME   DS      1               GET FILE NAME FROM CHANNEL
000.055                 800X .CLEAR  DS      1               CLEAR CHAN
000.056                 801X .CLEARA DS      1               CLEAR ALL CHANS
000.057                 802X .ERROR  DS      1               LOOKUP ERROR
000.060                 803X .CHFLG  DS      1               CHANGE FLAGS
000.061                 804X .DISM)  DS      1               FLAG SYSTEM DISK DISMOUNTED
000.062                 805X .LOADD  DS      1               LOAD DEVICE DRIVER
000.063                 806X .OPEN   DS      1               Parametrized Open
                        807X
                        808X
                        809X *       *HDOSOVL1.SYS*  FUNCTIONS
                        810X
000.200                 811X         ORG     2000
                        812X
000.200                 813X .MOUNT  DS      1               MOUNT  (MUST BE FIRST)
000.201                 814X .DMOUN  DS      1               DISMOUNT
000.202                 815X .MONMS  DS      1               MOUNT/NO MESSAGE
000.203                 816X .DMNMS  DS      1               DISMOUNT/NO MESSAGE
000.204                 817X .RESET  DS      1               RESET = DISMOUNT/MOUNT OF UNIT
000.205                 818X .CLEAN  DS      1               Clean device
000.206                 819X .DAD    DS      1               Dismount All Disks              /80.08.sc/
000.207                 820          XTEXT   FBDEF                                           /071080/
```

```
                        822X **     FILE BLOCK DEFINITIONS.
                        823X
000.000                 824X        ORG     0
000.000                 825X FB.CHA DS      1           CHANNEL NUMBER
000.001                 826X FB.FLG DS      1           FLAGS
000.002                 827X FB.FWA DS      2           BUFFER FWA
000.004                 828X FB.PTR DS      2           BUFFER POINTER
000.006                 829X FB.LIM DS      2           LIMIT OF DATA IN BUFFER (READ OPERATIONS)
000.010                 830X FB.LWA DS      2           LWA OF BUFFER
000.012                 831X FB.NAM DS      4+8+4+1     NAME OF FILE
000.021                 832X FB.NAML EQU    *-FB.NAM
000.033                 833X FBENL  EQU     *           ENTRY LENGTH
000.033                 834         XTEXT   IOCDEF                                  /071080/


                        836X **     I/O CHANNEL DEFINITIONS.
                        837X
000.000                 838X        ORG     0
                        839X
000.000                 840X IOC.LNK DS     2           ADDRESS OF NEXT CHANNEL, =0 IF LAST
000.002                 841X IOC.DDA DS     2           THREAD JUMP TO DEVICE DRIVER (VIA DEV TABLE)
                        842X
000.004                 843X IOC.FLG DS     1           FILE TYPE FLAGS
000.001                 844X FT.DD  EQU     00000001B   =1 IF DIRECTORY DEVICE
000.002                 845X FT.OR  EQU     00000010B   =1 IF OPEN FOR READ
000.004                 846X FT.OW  EQU     00000100B   =1 IF OPEN FOR WRITE
000.010                 847X FT.OU  EQU     00001000B   =1 IF OPEN FOR UPDATE
000.020                 848X FT.OC  EQU     00010000B   =1 IF OPEN FOR CHARACTER MODE   /80.02.GC/
000.003                 849X IOC.SQL EQU    *-IOC.DDA   LENGTH OF INFO FOR SEQUENTIAL FILE (FROM IOC)
                        850X
000.005                 851X IOC.GRT DS     2           ADDRESS OF GROUP RESERVATION TABLE
000.007                 852X IOC.SPG DS     1           SECTORS PER GROUP, THIS DEVICE
000.010                 853X IOC.CGN DS     1           CURRENT GROUP NUMBER
000.011                 854X IOC.CSI DS     1           CURRENT SECTOR INDEX (IN CURRENT GROUP)
000.012                 855X IOC.LGN DS     1           LAST GROUP NUMBER
000.013                 856X IOC.LSI DS     1           LAST SECTOR INDEX (IN LAST GROUP)
000.010                 857X IOC.DRL EQU    *-IOC.FLG   LENGTH OF INFO NORMALLY COPIED BACK TO
                        858X *                          THE CHANNEL TABLE
000.014                 859X IOC.DTA DS     2           DEVICE TABLE ADDRESS FOR THIS DEVICE
000.016                 860X IOC.DES DS     2           SECTOR NUMBER OF DIRECTORY ENTRY
000.020                 861X IOC.DEV DS     2           DEVICE CODE
000.022                 862X IOC.UNI DS     1           UNIT NUMBER (0-9)
000.021                 863X IOC.DIL EQU    *-IOC.DDA   LENGTH OF INFO FOR DIRECTORY FILE (FROM IOC)
                        864X
000.023                 865X IOC.DIR DS     DIRELEN     DIRECTORY ENTRY
                        866X
000.052                 867X IOCELEN EQU    *           IOC ENTRY LENGTH
                        868X
000.001                 869X IOCCID EQU     1           INDEX OF USER CHANNEL #0 IN CHANTAB (FIRST = 0)
                        870
                        871
042.170                 872         ORG     USERFWA-ABS.COD
042.170 377 000         873         DB      377Q,FT.ABS
042.172 200 042         874         DW      USERFWA     LOAD ADDR
042.174 043 033         875         DW      MEML-USERFWA SIZE
```

042.176 200 042 876 877 DW TEST ENTRY

```
                                880  **      TEST47
                                881  *
                                882  *
                                883
                                884  *       TEST RUNS AN EXTENSIVE TEST ON A HDOS 8" FLOPPY DISK.
                                885  *
                                886
                                887
042.200                         888  TEST    EQU     *
                                889
042.200  076 000               890          MVI     A,OVLO                                  /072080/
042.202  377 010               891          DB      SYSCALL,.LOADO
042.204  076 001               892          MVI     A,OVL1
042.206  377 010               893          DB      SYSCALL,.LOADO                          /072080/
                                894
042.210  041 230 100            895          LXI     H,RMEML         ENOUGH MEMORY           /071080/
042.213  377 052               896          DB      SYSCALL,.SETTP    TO GET STARTED
042.215  322 226 042            897          JNC     TEST0           BR IF YES
042.220  076 021               898          MVI     A,EC.NEM        NOT ENOUGH MEMORY
042.222  067                   899          STC
042.223  303 140 044            900          JMP     ERROR
                                901
042.226                         902  TEST0   EQU     *                                       /071080/
042.226  377 011               903          DB      SYSCALL,.VERS
042.230  332 240 042            904          JC      TEST1           NO VERSION SYSTEM CALL
042.233  376 040               905          CPI     VERS
042.235  312 246 042            906          JZ      TEST2           IS CORRECT VERSION OF HDOS
042.240  076 050               907  TEST1   MVI     A,EC.NCV        NOT CORRECT VERSION OF HDOS
042.242  067                   908          STC
042.243  303 140 044            909          JMP     ERROR
                                910
042.246  076 377               911  TEST2   MVI     A,377Q
042.250  377 055               912          DB      SYSCALL,.CLEAR  CLEAR THE CHANNEL THAT WE CAME IN ON
042.252  257                   913          XRA     A
042.253  062 326 040            914          STA     S.CSLMD         SET CONSOLE MODE
042.256  062 007 040            915          STA     .DSPMOD         DISPLAY MEMORY
042.261  363                   916          DI
042.262  072 010 040            917          LDA     .MFLAG
042.265  346 275               918          ANI     377Q-UO.DDU-UO.NFR
042.267  062 010 040            919          STA     .MFLAG          ALLOW DISPLAY
042.272  373                   920          EI
                                921
                                922  *       DISMOUNT SYSTEM DISKS
                                923
042.273  315 337 073            924          CALL    $DOS            DISMOUNT OPERATING SYSTEM
042.276  332 140 044            925          JC      ERROR
042.301                         926  MOUNT   EQU     *               ENTRY HERE TO MOUNT NEW DISK
042.301  041 301 042            927          LXI     H,MOUNT
042.304  076 003               928          MVI     A,CTLC
042.306  377 041               929          DB      SYSCALL,.CTLC   SETUP CTL-C PROCESSING
042.310  315 102 054            930          CALL    DUN             DETERMINE UNIT NUMBER TO WORK OVER
                                931
                                932  *       SETUP USE OF READ ROUTINE
                                933
042.313  052 033 040            934          LHLD    .TICCNT
042.316  042 113 075            935          SHLD    RSEED
```

```
042.321  041 000 000   938          LXI     H,0
042.324  042 111 075   937          SHLD    PASS            SET PASS NUMBER
                       938
                       939  *       START TESTS
                       940
042.327                941  RESTART EQU     *                                       *071080*
042.327  041 327 042   942          LXI     H,RESTART
042.332  076 003       943          MVI     A,CTLC
042.334  377 041       944          DB      SYSCALL,.CTLC   SET CTL-C PROCESSING
042.336  061 200 042   945          LXI     SP,STACK        RESET STACK
042.341  041 327 042   946          LXI     H,RESTART
042.344  345           947          PUSH    H               SET *RETURN ADDRESS*
042.345  076 377       948          MVI     A,377Q
042.347  062 006 040   949          STA     .DSPROT         OFF FP PERIODS
042.352  076 201       950          MVI     A,UO.CLK+UO.HLT
042.354  062 010 040   951          STA     .MFLAG          ENABLE CLOCK INTERRUPTS
042.357  076 007       952          MVI     A,DC.ABT
042.361  315 157 062   953          CALL    DDRV            ABORT DISK              *071080*
042.364  377 007       954          DB      SYSCALL,.CLRCO  CLEAR CONSOLE
042.366  315 136 031   955          CALL    $TYPTX
042.371  012 106 165   956          DB      NL,'Functions Available:',NL,NL         /072080/
043.020  104 040 055   957          DB      'D - General Drive Checkout       '
043.066  103 040 055   958          DB      'C - Clean Drive Head',NL
043.113  115 040 055   959          DB      'M - Media Check (Sector Validity)'
043.161  101 040 055   960          DB      'A - Align Drive Head',NL
043.206  125 040 055   961          DB      'U - Select Another Drive Unit    '
043.254  122 040 055   962          DB      'R - Hardcopy Report',NL
043.300  105 040 055   963          DB      'E - Exit to Boot Program',NL
043.331  012 103 124   964          DB      NL,'CTRL-C Cancels the Test in Progress.',ENL   /072080/
043.377  315 075 070   965          CALL    $CCO            CLEAR CTL-O
044.002  315 136 031   966          CALL    $TYPTX
044.005  040 117 160   967          DB      ' Option:',' '+2000
044.016  041 203 075   968          LXI     H,LINE
044.021  315 202 070   969          CALL    $RTL.           READ LINE IN UPPER CASE
044.024  176           970          MOV     A,M
044.025  247           971          ANA     A
044.026  312 327 042   972          JZ      RESTART         NO GOOD REPLY
044.031  041 121 044   973          LXI     H,DIAGA
044.034  315 247 070   974          CALL    $TBLS           FIND IN TABLE
044.037  312 077 044   975          JE      DIAG2
044.042  315 136 031   976          CALL    $TYPTX
044.045  007 111 114   977          DB      BELL,'ILLEGAL OPTION:',' '+2000
044.066  072 203 075   978          LDA     LINE
044.071  315 306 070   979          CALL    $WCHAR
044.074  303 327 042   980          JMP     RESTART
                       981
                       982  *       PERFORM DIAGNOSTIC
                       983
044.077                984  DIAG2   EQU     *                                       *071080*
044.077  176           985          MOV     A,M             (A) = INDEX
044.100  315 061 031   986          CALL    $TJMP
044.103  204 051       987          DW      DRIVE           DRIVE DIAGNOSTIC
044.105  135 052       988          DW      MEDIA           MEDIA CHECK
044.107  171 044       989          DW      EXIT            EXIT DIAGNOSTIC
044.111  301 042       990          DW      MOUNT           SELECT NEW DRIVE
044.113  177 044       991          DW      CLEAN           CLEAN HEAD              /071080/
```

```
044.115  075 045        992          DW      ALIGN           ALIGN HEAD
044.117  277 046        993          DW      REPORT          HARDCOPY REPORT                 /071080/
                        994
                        995
044.121  104 000        996  DIAGA   DB      'D',0                                           /072080/
044.123  115 001        997          DB      'M',1
044.125  105 002        998          DB      'E',2
044.127  125 003        999          DB      'U',3
044.131  103 004       1000          DB      'C',4
044.133  101 005       1001          DB      'A',5
044.135  122 006       1002          DB      'R',6                                           /072080/
044.137  000            1003          DB      0


                       1005  **      ERROR - DISK ERROR OCCURRED BEFORE DISKS DISMOUNTED.
                       1006  *
                       1007
                       1008
044.140  365           1009  ERROR   PUSH    PSW             SAVE CODE
044.141  315 075 070   1010          CALL    $CCO
044.144  315 136 031   1011          CALL    $TYPTX
044.147  012 002 105   1012          DB      NL,BELL,'ERROR -'+'0'+200Q
044.161  046 007       1013          MVI     H,BELL
044.163  361           1014          POP     PSW
044.164  377 057       1015          DB      SYSCALL,,ERROR
044.166  303 171 044   1016          JMP     EXIT


                       1018  **      EXIT - EXIT DIAGNOSTIC.
                       1019  *
                       1020  *       GIVE HIM TIME TO INSERT A DISK, THEN BOOT.
                       1021
                       1022
044.171                1023  EXIT    EQU     *                                               /071080/
044.171  315 303 047  1024          CALL    RPT8            CLOSE HARDCOPY DEVICE           /071080/
044.174  257           1025          XRA     A
044.175  377 000       1026          DB      SYSCALL,,EXIT   LET  *HDOS*  TAKE CARE OF THE ERROR STUFF
```

```
                        1029  ***     CLEAN - CLEAN DRIVE HEAD.
                        1030  *
                        1031
044.177                 1032  CLEAN   EQU     *
044.177  315 075 070   1033          CALL    $CCO
044.202  315 136 031   1034          CALL    $TYPTX
044.205  012 115 141   1035          DB      NL,'Make sure that a Cleaning Diskette is '
044.254  151 156 163   1036          DB      'inserted in drive',' '+80H
044.276  072 110 075   1037          LDA     MYUNIT
044.301  306 060       1038          ADI     '0'
044.303  315 306 070   1039          CALL    $WCHAR
044.306  315 136 031   1040          CALL    $TYPTX
044.311  056 012 110   1041          DB      '.',NL,'Hit return when ready',7,'?'+80H
                        1042
044.341                 1043  CLN0    EQU     *
044.341  315 300 070   1044          CALL    $RCHAR
044.344  376 012       1045          CPI     NL
044.346  302 341 044   1046          JNZ     CLN0
                        1047
044.351  257           1048          XRA     A
044.352  062 107 074   1049          STA     INTDSK          FLAG INITIALIZED DISK NOT MOUNTED
                        1050
                        1051  *       USE SPECIAL FUNCTIONS IN CONTROLLER TO SWEEP HEAD
                        1052  *       WITH A CLEANING DISKETTE INSTALLED.
                        1053
                        1054  *       PRELIMINARY CALL FOR SPECIAL FUNCTION ALIGN.
                        1055
044.355  072 110 075   1056          LDA     MYUNIT          GET UNIT #
044.360  062 061 041   1057          STA     AIO.UNI         PUT UNIT # IN HDOS TABLE
044.363  076 000       1058          MVI     A,SID.0                                  /091280/
044.365  062 122 075   1059          STA     SIDE            SIDE 0
044.370  257           1060          XRA     A
044.371  062 126 075   1061          STA     TRACK           TRACK 0
044.374  076 001       1062          MVI     A,1
044.376  062 121 075   1063          STA     SECTOR          SECTOR 1                 /091280/
045.001  315 327 065   1064          CALL    COM             COMMAND
045.004  021           1065          DB      11H
045.005  315 275 067   1066          CALL    TRK             TRACK
045.010  315 104 067   1067          CALL    SUS..           SIDE/UNIT/SECTOR         *091280*
045.013  315 171 067   1068          CALL    TEB             CHECK FOR ERROR
045.016  330           1069          RC                      RETURN IF ERROR
                        1070
                        1071  *       EXIT PRELIMINARY CALL.
                        1072
045.017  076 377       1073          MVI     A,0FFH          INVALID TRACK
045.021  315 022 066   1074          CALL    OUT               CAUSES EXIT
045.024  315 367 067   1075          CALL    WDN             WAIT FOR DONE USING TIMEOUT
045.027  330           1076          RC                      RETURN IF DONE NOT INDICATED
                        1077
                        1078  *       ISSUE CLEANING COMMAND.  MAKE 2 PASSES.
                        1079
045.030  076 002       1080          MVI     A,2
045.032  062 074 045   1081          STA     CLNA            LOOP COUNTER
                        1082
045.035                 1083  CLN3    EQU     *
045.035  315 327 065   1084          CALL    COM             COMMAND
```

```
045.040  021          1085          DB      11H
045.041  076 377      1086          MVI     A,0FFH          SPECIAL BYTE
045.043  315 022 066  1087          CALL    OUT
045.046  315 104 067  1088          CALL    SUS,,           SIDE/UNIT/SECTOR                *091280*
045.051  315 171 067  1089          CALL    TEB             CHECK FOR ERROR
045.054  330          1090          RC                      RETURN IF ERROR
                      1091
                      1092  *       WAIT UNTIL DONE, DON'T USE TIMEOUT,
                      1093
045.055               1094  CLN5    EQU     *
045.055  333 170      1095          IN      D.STA
045.057  346 040      1096          ANI     S,DON
045.061  312 055 045  1097          JZ      CLN5            BR IF NOT DONE
                      1098
045.064  041 074 045  1099          LXI     H,CLNA
045.067  065          1100          DCR     M
045.070  302 035 045  1101          JNZ     CLN3
                      1102
045.073  311          1103          RET
                      1104
045.074               1105  CLNA    DS      1                                       /072080/
```

```
                                1108  ***     ALIGN - ALIGN DRIVE HEAD.
                                1109  *
                                1110
045.075                         1111  ALIGN   EQU     *
045.075  315 075 070            1112          CALL    $CCO
045.100  315 136 031            1113          CALL    $TYPTX
045.103  012 115 141            1114          DB      NL,'Make sure that an Alignment Diskette is '
045.154  151 156 163            1115          DB      'inserted in drive',' '+80H
045.176  072 110 075            1116          LDA     MYUNIT
045.201  306 060                1117          ADI     '0'
045.203  315 306 070            1118          CALL    $WCHAR
045.206  315 136 031            1119          CALL    $TYPTX
045.211  056 012 110            1120          DB      '.';NL,'Hit return when ready',7,'?'+80H
                                1121
045.241                         1122  ALN$    EQU     *
045.241  315 300 070            1123          CALL    $RCHAR
045.244  376 012                1124          CPI     NL
045.246  302 241 045            1125          JNZ     ALN$
                                1126
045.251  257                   1127          XRA     A                                       /072080/
045.252  062 275 046            1128          STA     ALND            CLEAR FLAG              /072080/
045.255  062 107 074            1129          STA     INTDSK                                  *091280*
                                1130
                                1131  *       ASK WHICH SIDE TO USE.                          /091280/
                                1132
045.260                         1133  ALN$$   EQU     *
045.260  315 075 070            1134          CALL    $CCO
045.263  315 136 031            1135          CALL    $TYPTX
045.266  012 105 116            1136          DB      NL,'ENTER SIDE # <1>?',' '+80H
045.311  076 002                1137          MVI     A,2
045.313  041 203 075            1138          LXI     H,LINE
045.316  315 066 064            1139          CALL    $ETL.
045.321  332 260 045            1140          JC      ALN$$
045.324  056 001                1141          MVI     L,1             ASSUME DEFAULT
045.326  075                    1142          DCR     A
045.327  312 343 045            1143          JZ      ALN$$.1         BR IF DEFAULT REQUESTED
045.332  041 203 075            1144          LXI     H,LINE
045.335  315 277 073            1145          CALL    $PDD            CONVERT CHARACTER TO DECIMAL
045.340  332 260 045            1146          JC      ALN$$           BR IF ERRONEOUS
045.343                         1147  ALN$$.1 EQU     *
045.343  175                    1148          MOV     A,L
045.344  247                    1149          ANA     A
045.345  312 260 045            1150          JZ      ALN$$           BR IF SIDE # TOO SMALL
045.350  376 003                1151          CPI     2+1
045.352  322 260 045            1152          JNC     ALN$$           BR IF SIDE # TOO LARGE
045.355  056 000                1153          MVI     L,SID.0         ASSUME SIDE 1
045.357  376 002                1154          CPI     2
045.361  302 366 045            1155          JNE     ALN$$.2         BR IF SIDE 2 NOT REQUESTED
045.364  056 200                1156          MVI     L,SID.1
045.366                         1157  ALN$$.2 EQU     *
045.366  175                    1158          MOV     A,L
045.367  062 276 046            1159          STA     ALND            SAVE FOR LATER          /091280/
                                1160
                                1161  *       ASK FOR WHICH TRACK TO POSITION HEAD TO.
                                1162
045.372                         1163  ALN0    EQU     *
```

```
045.372  257           1164           XRA     A
045.373  062 274 046   1165           STA     ALNB            CLEAR FLAG
045.376  076 003       1166           MVI     A,CTLC          CTL-C IN RESPONSE TO QUESTION
046.000  041 327 042   1167           LXI     H,RESTART       RESULTS IN TERMINATION
046.003  377 041       1168           DB      SYSCALL,.CTLC       OF ALIGNMENT REQUEST
                       1169
046.005               1170  ALN0.5   EQU     *
046.005  315 075 070   1171           CALL    $CCO
046.010  315 136 031   1172           CALL    $TYPTX
046.013  012 105 116   1173           DB      NL,'ENTER TRACK # (<0>?):',' '+80H
046.037  076 003       1174           MVI     A,3
046.041  041 203 075   1175           LXI     H,LINE
046.044  315 066 064   1176           CALL    $ETL.           INPUT VALUE
046.047  332 005 046   1177           JC      ALN0.5          BR IF TOO MANY CHARACTERS
046.052  056 000       1178           MVI     L,0             ASSUME DEFAULT
046.054  075           1179           DCR     A
046.055  312 071 046   1180           JZ      ALN0.7          BR IF DEFAULT REQUESTED
046.060  041 203 075   1181           LXI     H,LINE
046.063  315 277 073   1182           CALL    $PDD            CONVERT CHARACTER TO DECIMAL
046.066  332 005 046   1183           JC      ALN0.5          BR IF ERRONEOUS
046.071               1184  ALN0.7   EQU     *
046.071  175           1185           MOV     A,L
046.072  376 115       1186           CPI     NTRK                                    *072080*
046.074  322 005 046   1187           JNC     ALN0.5          BR IF TRACK # TOO LARGE
046.077  062 273 046   1188           STA     ALNA            SAVE FOR LATER
                       1189
                       1190  *        POSITION HEAD AND KEEP LOADED.
                       1191
046.102  072 275 046   1192           LDA     ALNC                                    /072080/
046.105  247           1193           ANA     A
046.106  302 165 046   1194           JNZ     ALN1.5          BR IF NOT 1ST TRACK #
                       1195
                       1196  *        1ST CALL TO ALIGN SPECIAL FUNCTION.
                       1197
046.111  074           1198           INR     A
046.112  062 275 046   1199           STA     ALNC            SET FLAG
046.115  072 110 075   1200           LDA     MYUNIT          GET UNIT #
046.120  062 061 041   1201           STA     AIO.UNI         PUT UNIT # IN HDOS TABLE
046.123  072 276 046   1202           LDA     ALND                                    /091280/
046.126  062 122 075   1203           STA     SIDE            SET SIDE PARM FOR REQ SIDE
046.131  076 001       1204           MVI     A,1
046.133  062 121 075   1205           STA     SECTOR          SECTOR MUST NOT BE 0    /091280/
046.136  072 273 046   1206           LDA     ALNA
046.141  062 126 075   1207           STA     TRACK           SET TRACK PARM FOR REQ TRACK
046.144  315 327 065   1208           CALL    COM             COMMAND
046.147  021           1209           DB      11H
046.150  315 275 067   1210           CALL    TRK             TRACK
046.153  315 104 067   1211           CALL    SUS..           SIDE/UNIT/SECTOR        *091280*
046.156  315 171 067   1212           CALL    TEB             CHECK FOR ERROR
046.161  330           1213           RC                      RETURN IF ERROR
046.162  303 202 046   1214           JMP     ALN2            GO WAIT
                       1215
                       1216  *        2ND AND SUBSEQUENT TRACK REQUESTS.
                       1217
046.165               1218  ALN1.5   EQU     *
046.165  072 273 046   1219           LDA     ALNA
```

```
046.170  062 126 075  1220          STA     TRACK           SET TRACK PARAMETER
046.173  315 275 067  1221          CALL    TRK             MOVE TO REQ TRACK
046.176  315 171 067  1222          CALL    TEB             CHECK FOR ERROR
046.201  330          1223          RC                      RETURN IF ERROR
                      1224
                      1225  *       WAIT UNTIL REQUEST FOR ANOTHER TRACK.
                      1226
046.202               1227  ALN2    EQU     *                                       /072080/
046.202  076 003      1228          MVI     A,CTLC          CTL-C ONCE AT DESIRED TRACK
046.204  041 265 046  1229          LXI     H,ALN4            MEANS USER WANTS TO
046.207  377 041      1230          DB      SYSCALL,.CTLC       REQUEST ANOTHER TRACK
                      1231
046.211  315 136 031  1232          CALL    $TYPTX
046.214  103 124 114  1233          DB      'CTL-C TO REQUEST ANOTHER TRACK',ENL
                      1234
046.253               1235  ALN3    EQU     *
046.253  072 274 046  1236          LDA     ALNB
046.256  247          1237          ANA     A
046.257  302 372 045  1238          JNZ     ALN0            USER WANTS TO REQUEST ANOTHER TRACK
046.262  303 253 046  1239          JMP     ALN3
                      1240
                      1241  *       SET FLAG TO INDICATE USER WANTS ANOTHER TRACK.
                      1242
046.265               1243  ALN4    EQU     *
046.265  076 001      1244          MVI     A,1
046.267  062 274 046  1245          STA     ALNB
046.272  311          1246          RET
                      1247
046.273               1248  ALNA    DS      1               SAVE SLOT FOR TRACK NUMBER
046.274               1249  ALNB    DS      1               FLAG != <>0 USER WANTS ANOTHER TRACK
046.275               1250  ALNC    DS      1               FLAG != <>0 NOT 1ST REQUEST     *072080*
046.276               1251  ALND    DS      1               SAVE SLOT FOR SIDE VALUE        *091280*
```

```
                                1254  ***     REPORT - PRODUCE HARD COPY REPORT.
                                1255  *
                                1256
046.277                         1257  REPORT  EQU     *
                                1258
                                1259  *       CLOSE PRESENT HARDCOPY OUTPUT IF THERE IS ANY.
                                1260
046.277 315 303 047             1261          CALL    RPT8
                                1262
                                1263  *       ASK USER IF HE WANTS HARDCOPY OPTION TURNED ON/OFF.
                                1264
046.302 315 075 070             1265          CALL    $CCO
046.305 315 136 031             1266          CALL    $TYPTX
046.310 012 110 141             1267          DB      NL,'Hardcopy Report Option (ON/OFF) <OFF>?',' '+80H
046.360 076 004                 1268          MVI     A,4
046.362 041 203 075             1269          LXI     H,LINE
046.365 315 066 064             1270          CALL    $ETL.
046.370 041 203 075             1271          LXI     H,LINE
046.373 315 151 070             1272          CALL    $MLU
046.376 041 203 075             1273          LXI     H,LINE
047.001 176                     1274          MOV     A,M
047.002 376 117                 1275          CPI     'O'
047.004 300                     1276          RNZ
047.005 043                     1277          INX     H
047.006 176                     1278          MOV     A,M
047.007 376 116                 1279          CPI     'N'
047.011 300                     1280          RNZ
047.012 043                     1281          INX     H
047.013 176                     1282          MOV     A,M
047.014 247                     1283          ANA     A
047.015 300                     1284          RNZ                     RETURN IF REPLY IS NOT 'ON'
                                1285
                                1286  *       GET HARDCOPY DEVICE NAME.
                                1287
047.016                         1288  RPT1    EQU     *
047.016 315 136 031             1289          CALL    $TYPTX
047.021 012 105 156             1290          DB      NL,'Enter Hardcopy Device Name <LP:>?',' '+80H
047.064 076 005                 1291          MVI     A,5
047.066 041 203 075             1292          LXI     H,LINE
047.071 315 066 064             1293          CALL    $ETL.
047.074 332 016 047             1294          JC      RPT1            TOO MANY CHARACTERS
047.077 041 203 075             1295          LXI     H,LINE
047.102 315 151 070             1296          CALL    $MLU
047.105 001 177 051             1297          LXI     B,RPTG
047.110 021 174 051             1298          LXI     D,RPTF
047.113 041 203 075             1299          LXI     H,LINE
047.116 315 141 073             1300          CALL    DDS
047.121 332 016 047             1301          JC      RPT1            NOT A VALID DEVICE NAME SYNTAX
                                1302
                                1303  *       OPEN NEW HARDCOPY DEVICE.
                                1304
047.124 072 201 051             1305          LDA     RPTG+2          CONVERT UNIT NUMBER
047.127 306 060                 1306          ADI     '0'               FROM
047.131 062 201 051             1307          STA     RPTG+2              BINARY TO CHARACTER
047.134 315 126 070             1308          CALL    $MOVEL          MOVE DEVICE NAME TO FILE BLOCK
047.137 005 000 177             1309          DW      5,RPTG,RPTD+FB.NAM
```

```
                        1310
047.145  041 141 050  1311          LXI     H,RPTD
047.150  315 177 071  1312          CALL    $FOPEW.         OPEN
047.153  332 326 047  1313          JC      RPT9            BR IF ERROR
                        1314
                        1315  *     REQUEST TITLE FOR TEST OUTPUT.
                        1316
047.156  076 040      1317          MVI     A,' '
047.160  001 062 000  1318          LXI     B,RPTA1L
047.163  041 021 050  1319          LXI     H,RPTA1
047.166  315 147 064  1320          CALL    FILL            BLANK FILL
047.171  315 136 031  1321          CALL    $TYPTX
047.174  105 156 164  1322          DB      'Enter Title for Test Hardcopy Report',NL,'?'+80H
047.242  076 062      1323          MVI     A,RPTA1L
047.244  041 021 050  1324          LXI     H,RPTA1
047.247  315 077 064  1325          CALL    $ETL
047.252  315 126 070  1326          CALL    $MOVEL
047.255  011 000 277  1327          DW      9,S.DATE,RPTA2  MOVE TODAY'S DATE TO HEADING
                        1328
                        1329  *     SET FLAG TO INDICATE HARDCOPY REPORT TO BE PRODUCED.
                        1330
047.263  076 001      1331          MVI     A,1
047.265  062 140 050  1332          STA     RPTC
                        1333
047.270  062 345 064  1334          STA     FNPA            SET FLAG FOR FNP
047.273  257          1335          XRA     A
047.274  062 127 065  1336          STA     WRTLA           INIT LINE COUNTER
047.277  062 137 050  1337          STA     RPTB            INIT PAGE COUNTER
047.302  311          1338          RET
                        1339
                        1340  *     CLOSE HARDCOPY DEVICE.
                        1341
047.303                 1342  RPT8    EQU     *
047.303  072 140 050  1343          LDA     RPTC
047.306  247          1344          ANA     A
047.307  310          1345          RZ                      NOT IN USE
047.310  315 330 064  1346          CALL    FNP.            FORCE NEW PAGE W/O HEADING
047.313  041 141 050  1347          LXI     H,RPTD
047.316  315 316 071  1348          CALL    $FCLO.          CLOSE
047.321  257          1349          XRA     A
047.322  062 140 050  1350          STA     RPTC            INDICATE HARDCOPY DEVICE NOT IN USE
047.325  311          1351          RET
                        1352
                        1353  *     ERROR ON OPENING HARDCOPY OUTPUT DEVICE.
                        1354
047.326                 1355  RPT9    EQU     *
047.326  315 136 031  1356          CALL    $TYPTX
047.331  012 007 125  1357          DB      NL,BELL,'Unable to Open Hardcopy Device'
047.371  012 122 145  1358          DB      NL,'Report Request Denied',ENL
050.020  311          1359          RET
                        1360
050.021                 1361  RPTA    DS      0               HEADING
050.021                 1362  RPTA1   DS      50                TITLE
000.062                 1363  RPTA1L  EQU     *-RPTA1
050.103  040 040 040    1364          DB      '   '
050.110                 1365  RPTA2   DS      9                 TODAY'S DATE
```

```
050.121  040 040 040  1366          DB      '    '
050.126  120 101 107  1367          DB      'PAGE '
050.133               1368  RPTA3   DS      2               PAGE NUMBER
050.135  012 012       1369          DB      NL,NL
000.116               1370  RPTAL   EQU     *-RPTA
050.137  000          1371  RPTB    DB      0               PAGE COUNTER
050.140  000          1372  RPTC    DB      0               FLAG := <>0 PRODUCE HARDCOPY OUTPUT
050.141               1373  RPTD    DS      0               HARDCOPY FILE BLOCK
050.141  001          1374          DB      1                 CHANNEL # 1
050.142  000          1375          DB      0                 FLAGS
050.143  174 050      1376          DW      RPTE
050.145  174 050      1377          DW      RPTE
050.147  174 050      1378          DW      RPTE
050.151  174 051      1379          DW      RPTE+RPTEL
050.153               1380          DS      FB.NAML           HARDCOPY FILENAME (DEVICE)
000.000               1381          ERRNZ   *-RPTD-FBENL
050.174               1382  RPTE    DS      256             HARDCOPY BUFFER
001.000               1383  RPTEL   EQU     *-RPTE
051.174  114 120 060  1384  RPTF    DB      'LPO'           DEFAULT HARDCOPY DEVICE
051.177  104 104 116  1385  RPTG    DB      'DDN:',0                                        /071080/
```

```
                        1389  ***     DRIVE - PERFORM GENERAL DRIVE TESTS
                        1390  *
                        1391  *       DRIVE PERFORMS A GENERAL DRIVE DIAGNOSTIC BY
                        1392  *       A SERIES OF 7 TESTS:
                        1393  *
                        1394  *       A) WRITE ALL ZEROS
                        1395  *       B) READ ALL ZEROS
                        1396  *       C) WRITE ALL ONES
                        1397  *       D) READ ALL ONES
                        1398  *       E) WRITE ID PATTERN
                        1399  *       F) READ ID PATTERN
                        1400  *       G) RANDOM READ/WRITE TEST
                        1401  *
                        1402  *       BEFORE EACH TEST IS STARTED, ITS LETTER IS TYPED. IF ANY
                        1403  *       ERRORS OCCURR DURING THAT PASS, THE NUMBER IS TYPED AS
                        1404  *       HHH, WHERE HHH = HARD ERROR COUNT.
                        1405  *
                        1406  *       ENTRY   NONE
                        1407  *       EXIT    TO RESTART VIA CTL-C
                        1408  *       USES    ALL
                        1409
                        1410
051.204                 1411  DRIVE   EQU     *
                        1412
                        1413  *       READ AND ZAP DISK LABEL SECTOR.                 /101080/
                        1414
051.204  315 103 056   1415          CALL    RL
051.207  315 051 057   1416          CALL    ZL                                      /101080/
                        1417
                        1418  *       SET CTL-C TO ABORT TESTS AND JUMP TO RESTART.   /071080/
                        1419
051.212  076 003       1420          MVI     A,CTLC
051.214  041 347 051   1421          LXI     H,TEST5
051.217  377 041       1422          DB      SYSCALL,.CTLC
                        1423
051.221  041 046 052   1424          LXI     H,DRIVEA
051.224  315 350 064   1425          CALL    PMSG                                    /071080/
051.227  257           1426          XRA     A
051.230  062 111 075   1427          STA     PASS            CLEAR PASS NUMBER
051.233  315 326 061   1428  DRIVE1  CALL    CEC             CLEAR ERROR COUNTS
051.236  041 126 075   1429          LXI     H,TRACK                                 *072080*
051.241  042 024 040   1430          SHLD    .ABUSS          SET TRACK ON DISPLAY
051.244  315 357 051   1431          CALL    TESTA           WRITE A'S
051.247  315 367 061   1432          CALL    PSE             PRINT SIGNIFICAT ERRORS
051.252  315 367 051   1433          CALL    TESTB
051.255  315 367 061   1434          CALL    PSE
051.260  315 377 051   1435          CALL    TESTC
051.263  315 367 061   1436          CALL    PSE
051.266  315 010 052   1437          CALL    TESTD
051.271  315 367 061   1438          CALL    PSE
051.274  315 021 052   1439          CALL    TESTE
051.277  315 367 061   1440          CALL    PSE
051.302  315 030 052   1441          CALL    TESTF
051.305  315 367 061   1442          CALL    PSE
051.310  315 037 052   1443          CALL    TESTG
051.313  315 367 061   1444          CALL    PSE
```

```
051.316  041 111 075  1445          LXI     H,PASS
051.321  064          1446          INR     M
051.322  176          1447          MOV     A,M
051.323  376 004      1448          CPI     4
051.325  306 060      1449          ADI     '0'
051.327  365          1450          PUSH    PSW             SAVE CODE
051.330  062 132 052  1451          STA     DRIVEB1                         /071080/
051.333  041 115 052  1452          LXI     H,DRIVEB
051.336  315 350 064  1453          CALL    PMSG            TYPE/PRINT MSG  /071080/
051.341  361          1454          POP     PSW
051.342  376 063      1455          CPI     '3'
051.344  302 233 051  1456          JNE     DRIVE1
                      1457
                      1458  *       ABORT TEST.                             /071080/
                      1459
051.347               1460  TEST5   EQU     *
051.347  076 007      1461          MVI     A,DC.ABT
051.351  315 157 062  1462          CALL    DDRV
                      1463
051.354  303 327 042  1464          JMP     RESTART                         /071080/


                      1466  **      TESTA - WRITE ALL ZEROS
                      1467  *
                      1468
051.357               1469  TESTA   EQU     *                               /071080/
051.357  315 371 064  1470          CALL    PMSG.                           /071080/
051.362  301          1471          DB      'A'+200Q
051.363  257          1472          XRA     A
051.364  303 051 060  1473          JMP     WCP             WRITE CONSTANT PATTERN


                      1475  **      TESTB - READ ALL ZEROS
                      1476
051.367               1477  TESTB   EQU     *                               /071080/
051.367  315 371 064  1478          CALL    PMSG.                           /071080/
051.372  302          1479          DB      'B'+200Q
051.373  257          1480          XRA     A                               *072080*
051.374  303 231 057  1481          JMP     CCP             CHECK FOR CONSTANT PATTERN


                      1483  **      TESTC - WRITE ALL ONES
                      1484
051.377               1485  TESTC   EQU     *                               /071080/
051.377  315 371 064  1486          CALL    PMSG.                           /071080/
052.002  303          1487          DB      'C'+200Q
052.003  076 377      1488          MVI     A,377Q
052.005  303 051 060  1489          JMP     WCP             WRITE CONSTANT PATTERN
```

```
                        1491  **      TESTD - READ ALL ONES
                        1492
052.010                 1493  TESTD   EQU     *                                       /071080/
052.010 315 371 064     1494          CALL    PMSG.                                   /071080/
052.013 304             1495          DB      'D'+200Q
052.014 076 377         1496          MVI     A,377Q                                  *072080*
052.016 303 231 057     1497          JMP     CCP             CHECK FOR CONSTANT PATTERN

                        1499  **      TESTE - WRITE ID PATTERN
                        1500
052.021                 1501  TESTE   EQU     *                                       /071080/
052.021 315 371 064     1502          CALL    PMSG.                                   /071080/
052.024 305             1503          DB      'E'+200Q
052.025 303 040 061     1504          JMP     WIP             WRITE ID PATTERN

                        1506  **      TESTF - READ ID PATTERN
                        1507
052.030                 1508  TESTF   EQU     *                                       /071080/
052.030 315 371 064     1509          CALL    PMSG.                                   /071080/
052.033 306             1510          DB      'F'+200Q
052.034 303 173 061     1511          JMP     CIP             CHECK ID PATTERN

                        1513  **      TESTG - RANDOM SEEK TEST
                        1514
                        1515
052.037                 1516  TESTG   EQU     *                                       /071080/
052.037 315 371 064     1517          CALL    PMSG.                                   /071080/
052.042 307             1518          DB      'G'+200Q
052.043 303 224 060     1519          JMP     RRT             RANDOM READ/WRITE TEST
                        1520
052.046                 1521  DRIVEA  DS      0                                       /071080/
052.046 012 063 040     1522          DB      NL,'3 Pass General Drive Test for Unit '
052.112                 1523  DRIVEA1 DS      1
052.113 012 200         1524          DB      NL,80H
052.115                 1525  DRIVEB  DS      0
052.115 040 105 156     1526          DB      ' End of Pass '
052.132                 1527  DRIVEB1 DS      1
052.133 012 200         1528          DB      NL,80H                                  /071080/
```

```
                                1532  **      MEDIA - CHECK SECTOR VALIDITY.                          /072080/
                                1533  *
                                1534  *       MEDIA CHECKS ALL NON-RESERVED (WRITABLE) SECTORS.
                                1535  *
                                1536  *       EACH SECTOR IS WRITTEN WITH ALL ZEROS, ALL ONES,
                                1537  *       THEN A FENCE PATTERN.
                                1538  *
                                1539  *       FOR EACH WRITE AND READ OPERATION, IF AN ERROR OCCURRS, THEN
                                1540  *       THE SECTOR IS MARKED IN A TABLE TO BE REPORTED AS BAD.
                                1541  *
                                1542  *       ENTRY   NONE
                                1543  *       EXIT    NONE
                                1544  *       USES    ALL
                                1545
      052.135                   1546  MEDIA   EQU     *
                                1547
                                1548  *       READ AND ZAP DISK LABEL SECTOR.                         /101080/
                                1549
      052.135  315 103 056      1550          CALL    RL
      052.140  315 051 057      1551          CALL    ZL                                              /101080/
                                1552
                                1553  *       SET CTRL-C TO ABORT AND JUMP TO RESTART.                /071080/
                                1554
      052.143  076 003          1555          MVI     A,CTLC
      052.145  041 330 052      1556          LXI     H,MEDIA8
      052.150  377 041          1557          DB      SYSCALL+.CTLC                                   /071080/
                                1558
      052.152  041 000 000      1559          LXI     H,0                                             /072080/
      052.155  042 370 052      1560          SHLD    MEDIAA          CLEAR BAD SECTOR COUNT
      052.160  257              1561          XRA     A
      052.161  001 365 001      1562          LXI     B,SECERRL
      052.164  041 243 076      1563          LXI     H,SECERR
      052.167  315 147 064      1564          CALL    FILL            CLEAR BAD SECTOR TABLE          /072080/
      052.172  076 001          1565          MVI     A,1
      052.174  062 025 040      1566          STA     .ABUSS+1        SET PASS
      052.177  041 000 000      1567          LXI     H,0                                             *090980*
      052.202  315 103 053      1568          CALL    CSV             CHECK SECTOR VALIDTY WITH OS
      052.205  041 025 040      1569          LXI     H,.ABUSS+1
      052.210  064              1570          INR     M
      052.211  041 377 377      1571          LXI     H,0FFFFH                                        *090980*
      052.214  315 103 053      1572          CALL    CSV             CHECK SECTOR VALIDITY WITH 1'S
      052.217  041 025 040      1573          LXI     H,.ABUSS+1
      052.222  064              1574          INR     M
      052.223  041 333 311      1575          LXI     H,0C9DBH                                        /090980/
      052.226  315 103 053      1576          CALL    CSV             CHECK VALIDITY WITH DBC9 HEX    /090980/
                                1577
                                1578  *       REPORT BADDIES BY SCANNING THROUGH TABLE AND REPORTING
                                1579  *       THE SECTORS ON WHICH HARD ERRORS OCCURRED.
                                1580
      052.231  041 372 052      1581          LXI     H,MEDIAB                                        /071080/
      052.234  315 012 065      1582          CALL    WRTL                                            /071080/
                                1583
      052.237  001 000 000      1584          LXI     B,0             SECTOR # COUNTER                /072080/
      052.242  021 365 001      1585          LXI     D,SECERRL       LENGTH OF BAD SECTOR TABLE
      052.245  041 243 076      1586          LXI     H,SECERR        FWA OF BAD SECTOR TABLE
                                1587
```

```
                                1588  *       THE TABLE IS GROUPED 8 SECTORS PER BYTE.
                                1589
        052.250                 1590  MEDIA5  EQU     *
        052.250  325           1591          PUSH    D
        052.251  126           1592          MOV     D,M             GET A BYTE FROM TABLE
        052.252  043           1593          INX     H
        052.253  345           1594          PUSH    H
                                1595
                                1596  *       ROTATE BYTE AND LOOK AT ALL EIGHT BITS.
                                1597  *       IF A BIT .EQ. 1 THEN A HARD ERROR OCCURRED ON THAT SECTOR.
                                1598
        052.254  036 001        1599          MVI     E,1
        052.256  046 010        1600          MVI     H,8
        052.260                 1601  MEDIA6  EQU     *
        052.260  172           1602          MOV     A,D
        052.261  243           1603          ANA     E
        052.262  304 340 052    1604          CNZ     MEDIA10         REPORT ERROR
        052.265  173           1605          MOV     A,E
        052.266  007           1606          RLC
        052.267  137           1607          MOV     E,A
        052.270  003           1608          INX     B               BUMP SECTOR # COUNTER
        052.271  045           1609          DCR     H
        052.272  302 260 052    1610          JNZ     MEDIA6
                                1611
        052.275  341           1612          POP     H
        052.276  321           1613          POP     D
        052.277  033           1614          DCX     D
        052.300  172           1615          MOV     A,D
        052.301  263           1616          ORA     E
        052.302  302 250 052    1617          JNZ     MEDIA5          BR IF MORE TO GO
                                1618
                                1619  *       SUMMARY MESSAGE
                                1620
        052.305                 1621  MEDIA7  EQU     *
        052.305  052 370 052    1622          LHLD    MEDIAA
        052.310  104           1623          MOV     B,H
        052.311  115           1624          MOV     C,L
        052.312  041 024 053    1625          LXI     H,MEDIAC1
        052.315  076 004        1626          MVI     A,4
        052.317  315 311 070    1627          CALL    $UDDN
        052.322  041 023 053    1628          LXI     H,MEDIAC
        052.325  315 350 064    1629          CALL    PMSG                                    /072080/
                                1630
                                1631  *       ABORT TEST.
                                1632
        052.330                 1633  MEDIA8  EQU     *
        052.330  076 007        1634          MVI     A,DC.ABT
        052.332  315 157 062    1635          CALL    DDRV
                                1636
        052.335  303 327 042    1637          JMP     RESTART
```

```
                        1639  **      REPORT ERROR
                        1640  *
                        1641  *       (BC) = SECTOR NUMBER
                        1642  *       USES    NONE
                        1643
                        1644
 052.340  315 054 031   1645  MEDIA10 CALL    $SAVALL
 052.343  076 004       1646          MVI     A,4                                 *072080*
 052.345  041 065 053   1647          LXI     H,MEDIAD1
 052.350  315 311 070   1648          CALL    $UDDN
 052.353  041 056 053   1649          LXI     H,MEDIAD
 052.356  315 350 064   1650          CALL    PMSG                                /071080/
 052.361  041 370 052   1651          LXI     H,MEDIAA
 052.364  064           1652          INR     M               COUNT BAD SECTOR
 052.365  303 047 031   1653          JMP     $RSTALL         RESTORE AND EXIT
                        1654
 052.370               1655  MEDIAA  DS      2               ERROR COUNT         *072080*
 052.372               1656  MEDIAB  DS      0                                   /071080/
 052.372  012 115 145   1657          DB      NL,'Media Check for Unit '
 053.020               1658  MEDIAB1 DS      1
 053.021  012 200       1659          DB      NL,80H
 053.023               1660  MEDIAC  DS      0
 053.023  012           1661          DB      NL
 053.024               1662  MEDIAC1 DS      4                                   *072080*
 053.030  040 102 141   1663          DB      ' Bad Sectors Located',NL,80H
 053.056               1664  MEDIAD  DS      0
 053.056  123 145 143   1665          DB      'Sector '
 053.065               1666  MEDIAD1 DS      4                                   *072080*
 053.071  040 151 163   1667          DB      ' is Bad.',NL,80H                   /071080/


                        1669  **      CSV - CHECK SECTOR VALIDITY.
                        1670  *
                        1671  *       CSV CHECKS A DISK VOLUME FOR VALIDITY OVER THE
                        1672  *       PATTERN.
                        1673  *
                        1674  *       THE GIVEN BYTE IS WRITTEN TO EACH SECTOR, THEN READ BACK.
                        1675  *
                        1676  *       ANY ERRORS ARE RECORDED IN 'SECERR'.
                        1677  *
                        1678  *
                        1679  *       ENTRY   (A) = PATTERN
                        1680  *       EXIT    NONE
                        1681  *       USES    ALL
                        1682
                        1683
 053.103               1684  CSV     EQU     *                                   /072080/
 053.103  042 041 054   1685          SHLD    CSVB            SAVE PATTERN        /090980/
                        1686
 053.106  353           1687          XCHG
 053.107  001 200 000   1688          LXI     B,128
 053.112  041 243 075   1689          LXI     H,BUFF
 053.115  315 163 064   1690          CALL    FILLW           FILL BUFFER WITH PATTERN  /090980/
                        1691
```

```
                                1692  *       TRY WRITE
                                1693
053.120  001 003 005           1694          LXI     B,5*256+3       PREPARE INTERLEAVE TABLE        /091180/
053.123  041 006 054           1695          LXI     H,CSVD
053.126  315 310 062           1696          CALL    INTRLV                                          /091180/
                                1697
053.131  076 001               1698          MVI     A,DC.WRI        WRITE OP CODE
053.133  315 151 053           1699          CALL    CSV2            DO IT
                                1700
                                1701  *       TRY READ
                                1702
053.136  001 003 005           1703          LXI     B,5*256+3       PREPARE INTERLEAVE TABLE        /091180/
053.141  041 006 054           1704          LXI     H,CSVD
053.144  315 310 062           1705          CALL    INTRLV                                          /091180/
                                1706
053.147  076 000               1707          MVI     A,DC.REA        READ OP CODE
                                1708  *       JMP     CSV2            DO READ AND EXIT
000.000                         1709          ERRNZ   CSV2-*


                                1711  **      CSV2 - READ/WRITE PASS
                                1712  *
                                1713
053.151                         1714  CSV2    EQU     *
053.151  062 001 054           1715          STA     CSVA            SAVE READ/WRITE CODE
053.154  041 000 000           1716          LXI     H,0             INIT SECTOR #
053.157  257                   1717          XRA     A
053.160  062 024 040           1718          STA     .ABUSS          INIT TRACK # ON DISPLAY
                                1719
                                1720  *       CHECK IF ANY MORE SECTORS TO TEST.
                                1721  *       CAN'T GO PAST VOLUME SIZE.
                                1722
053.163                         1723  CSV2.1  EQU     *
053.163  042 002 054           1724          SHLD    CSVB            SAVE SECTOR # OF 1ST SECTOR ON TRACK
053.166  315 067 054           1725          CALL    CHKWR2
053.171  320                   1726          RNC                     RETURN IF ALL DONE
                                1727
                                1728  *       INITIALIZE INTERLEAVE POINTER.
                                1729
053.172  041 006 054           1730          LXI     H,CSVD
053.175  345                   1731          PUSH    H
                                1732
                                1733  *       READ OR WRITE NEXT SECTOR.
                                1734
053.176                         1735  CSV4    EQU     *
053.176  341                   1736          POP     H               GET INTERLEAVE POINTER
053.177  176                   1737          MOV     A,M
053.200  247                   1738          ANA     A
053.201  372 340 053           1739          JM      CSV7            AT END OF TABLE
053.204  043                   1740          INX     H
053.205  345                   1741          PUSH    H               UPDATE INTERLEAVE POINTER
053.206  052 002 054           1742          LHLD    CSVB            SECTOR # = 1ST SECTOR OF TRACK
053.211  315 101 030           1743          CALL    $DADA.            + INTERLEAVE VALUE
053.214  315 043 054           1744          CALL    CHKWR           CHECK IF WRITEABLE SECTOR
053.217  332 176 053           1745          JC      CSV4
```

```
053.222  042 004 054  1746            SHLD    CSVC            SAVE SECTOR #
                      1747
053.225  072 001 054  1748            LDA     CSVA            GET READ/WRITE CODE
053.230  001 000 001  1749            LXI     B,256           COUNT
053.233  021 243 075  1750            LXI     D,BUFF            BUFFER ADDR
053.236  315 157 062  1751            CALL    DDRV
053.241  332 310 053  1752            JC      CSV5            BR IF ERROR
                      1753
                      1754  *         IF READ, THEN SEE IF DATA MATCHES SAVED PATTERN.
                      1755  *         THIS CHECKS FOR ERRORS NOT CAUGHT BY H47 HARDWARE.
                      1756
053.244  072 001 054  1757            LDA     CSVA
053.247  376 000      1758            CPI     DC.REA
053.251  302 176 053  1759            JNZ     CSV4            BR IF NOT READ
                      1760
053.254  006 200      1761            MVI     B,128
053.256  052 041 054  1762            LHLD    CSVG                                    /090980/
053.261  353          1763            XCHG                                            /090980/
053.262  041 243 075  1764            LXI     H,BUFF
                      1765
053.265               1766  CSV4.7    EQU     *
053.265  173          1767            MOV     A,E                                     *090980*
053.266  276          1768            CMP     M
053.267  302 310 053  1769            JNZ     CSV5
053.272  043          1770            INX     H
053.273  172          1771            MOV     A,D                                     *090980*
053.274  276          1772            CMP     M
053.275  302 310 053  1773            JNZ     CSV5
053.300  043          1774            INX     H
053.301  005          1775            DCR     B
053.302  302 265 053  1776            JNZ     CSV4.7
                      1777
053.305  303 176 053  1778            JMP     CSV4
                      1779
                      1780  *         FLAG ERROR FOR THIS SECTOR.
                      1781
053.310               1782  CSV5      EQU     *
000.001               1783            IF      .DEBUG.
                      1784            LHLD    CSVC
                      1785            CALL    DBGO
                      1786            ENDIF
                      1787
053.310  052 004 054  1788            LHLD    CSVC            GET SECTOR #
053.313  104          1789            MOV     B,H             DIVIDE SECTOR #
053.314  115          1790            MOV     C,L               BY 8
053.315  021 010 000  1791            LXI     D,8                 QUOTIENT = DISPLACEMENT INTO TABLE
053.320  315 106 030  1792            CALL    $DU66                 REMAINDER = BIT #
053.323  001 243 076  1793            LXI     B,SECERR
053.326  011          1794            DAD     B               ADDR OF ENTRY IN TABLE
053.327  176          1795            MOV     A,M             GET IT
053.330  103          1796            MOV     B,E
053.331  315 056 070  1797            CALL    BITS            SET BIT
053.334  167          1798            MOV     M,A             UPDATE TABLE
                      1799
053.335  303 176 053  1800            JMP     CSV4            ON TO NEXT SECTOR
                      1801
```

```
                        1802  *       ADVANCE TO NEXT TRACK.
                        1803
053.340                 1804  CSV7    EQU     *
053.340  052 002 054    1805          LHLD    CSVB
053.343  072 124 075    1806          LDA     SPT                                     /090980/
053.346  315 101 030    1807          CALL    $DADA.                                  /090980/
053.351  072 130 074    1808          LDA     LABEL+LAB.VFL                           /090980/
053.354  346 001        1809          ANI     VFL.NSD
053.356  017            1810          RRC
053.357  107            1811          MOV     B,A                                     /090980/
053.360  072 122 075    1812          LDA     SIDE
053.363  270            1813          CMP     B               Q. TIME TO UPDATE DISPLAY /090980/
053.364  302 163 053    1814          JNZ     CSV2.1              BR IF NOT
053.367  072 024 040    1815          LDA     .ABUSS
053.372  074            1816          INR     A               UPDATE TRACK # ON DISPLAY
053.373  062 024 040    1817          STA     .ABUSS
053.376  303 163 053    1818          JMP     CSV2.1                                  /072080/
                        1819
054.001  000            1820  CSVA    DB      0               READ/WRITE CODE
054.002                 1821  CSVB    DS      2               SECTOR # OF 1ST SECTOR ON TRACK /072080/
054.004                 1822  CSVC    DS      2               SECTOR # BEING WRITTEN OR READ
054.006                 1823  CSVD    DS      0               SECTOR INTERLEAVE TABLE
054.006  000 004 010    1824          DB      0,4,8,12,16,20,24,2,6,10,14,18,22
054.023  001 005 011    1825          DB      1,5,9,13,17,21,25,3,7,11,15,19,23,80H
054.041                 1826  CSVG    DS      2               PATTERN                 /072080/


                        1828  **      CHKWR - CHECK WRITEABLE.                        /072080/
                        1829  *
                        1830  *       CHKWR CHECKS IF SECTOR # IS IN THE WRITEABLE RANGE
                        1831  *       OF SECTOR #'S.
                        1832  *
                        1833  *       SCT1WR <= N < VOLUME SIZE
                        1834  *
                        1835
054.043                 1836  CHKWR   EQU     *
054.043  315 054 054    1837          CALL    CHKWR1
054.046  330            1838          RC
054.047  315 067 054    1839          CALL    CHKWR2
054.052  077            1840          CMC
054.053  311            1841          RET
                        1842
                        1843  *       COMPARE SECTOR # TO SCT1WR.
                        1844
054.054                 1845  CHKWR1  EQU     *
054.054  325            1846          PUSH    D
054.055  353            1847          XCHG
054.056  052 115 075    1848          LHLD    SCT1WR
054.061  353            1849          XCHG
054.062  315 120 070    1850          CALL    HLCPDE
054.065  321            1851          POP     D
054.066  311            1852          RET
                        1853
                        1854  *       COMPARE SECTOR # TO VOLUME SIZE.
```

```
                          1855
054.067                   1856  CHKWR2  EQU     *
054.067  325             1857          PUSH    D
054.070  353             1858          XCHG
054.071  052 124 074     1859          LHLD    LABEL+LAB.SIZ
054.074  353             1860          XCHG
054.075  315 120 070     1861          CALL    HLCPDE
054.100  321             1862          POP     D
054.101  311             1863          RET                                   /072080/
```

```
                                1867  **      DUN - DETERMINE UNIT NUMBER.
                                1868  *
                                1869  *       DUN DISCOVERS THE UNIT NUMBER TO DIAGNOSE, AFTER SUITABLE
                                1870  *       REDUNDANT WARNINGS.
                                1871  *
                                1872  *       ENTRY   NONE
                                1873  *       EXIT    TO CALLER WITH UNIT = NUMBER IF OK
                                1874  *               TO SYSTEM IF USER CHICKENS OUT
                                1875  *       USES    ALL
                                1876
                                1877
054.102  315 136 031            1878  DUN     CALL    $TYPTX
054.105  012 011 011            1879          DB      NL,TAB,TAB,TAB,'      ^','TEST47'
054.123  012 011 011            1880          DB      NL,TAB,TAB,TAB,'Version:  ',VERS/16+'0','.',VERS&0FH+'0'
054.144  012 011 011            1881          DB      NL,TAB,TAB,'      ','Issue  #50.06.00',ENL        *071080*
                                1882
                                1883  *       WARN HIM ABOUT THE FACTS OF LIFE
                                1884
054.176  315 136 031            1885  DUN1    CALL    $TYPTX
054.201  007 012 011            1886          DB      BELL,NL,TAB,'This  program tests your disk system.  Certain'
054.262  040 164 145            1887          DB      ' tests'
054.270  012 144 145            1888          DB      NL,'destroy the '
054.305  144 141 164            1889          DB      'data on the volume under test.  This volume must'
054.365  012 150 141            1890          DB      NL,'have been '
055.000  151 156 151            1891          DB      'initialized at least once, and may have to be'
055.062  012 162 145            1892          DB      NL,'reinitialized'
055.100  040 142 145            1893          DB      ' before being used for anything else.',ENL
055.146  315 075 070            1894          CALL    $CCO
055.151  315 136 031            1895          CALL    $TYPTX
055.154  012 120 162            1896          DB      NL,'Proceed (Yes/No)?',' '+2000
055.177  315 340 061            1897          CALL    CYR             CHECK FOR YES REPLY
055.202  302 171 044            1898          JNE     EXIT            TRY AGAIN
                                1899
                                1900  *       HE'S BEEN WARNED. FIND OUT WHICH UNIT HE WANTS
                                1901
055.205  315 075 070            1902  DUN2    CALL    $CCO
055.210  315 136 031            1903          CALL    $TYPTX
055.213  012 127 150            1904          DB      NL,'Which Drive (0/1) ?',' '+2000               *072080*
055.240  041 203 075            1905          LXI     H,LINE
055.243  315 211 070            1906          CALL    $RTL
055.246  176                    1907          MOV     A,M
055.247  326 060                1908          SUI     '0'
055.251  332 205 055            1909          JC      DUN2
055.254  376 002                1910          CPI     2                                               *072080*
055.256  322 205 055            1911          JNC     DUN2
055.261  062 110 075            1912          STA     MYUNIT
055.264  306 060                1913          ADI     '0'             PLACE UNIT # IN MESSAGE         /071080/
055.266  062 112 052            1914          STA     DRIVEA1
055.271  062 020 053            1915          STA     MEDIAB1
055.274  257                    1916          XRA     A
055.275  062 107 074            1917          STA     INTDSK          FLAG INITIALIZED DISK NOT MOUNTED
055.300  311                    1918          RET                                                     /071080/
```

```
                                1921  **      RID - REQUIRE INITIALIZED DISK TO BE MOUNTED.
                                1922  *
                                1923
055,301                         1924  RID     EQU     *
055.301  072 107 074           1925          LDA     INTDSK
055,304  247                   1926          ANA     A
055.305  300                   1927          RNZ                     INITIALIZED DISK ALREADY MOUNTED
                                1928
                                1929  *       ASK USER TO MOUNT INITIALIZED DISK.
                                1930
056.306  315 075 070           1931          CALL    $CCO
056,311  315 136 031           1932          CALL    $TYPTX
056.314  012 111 156           1933          DB      NL,'Insert the Diskette you wish to use for this test'
056,376  012 151 156           1934          DB      NL,'into drive',' '+200Q
056.012  072 110 075           1935          LDA     MYUNIT
056.015  306 060               1936          ADI     '0'
056.017  315 306 070           1937          CALL    $WCHAR
056.022  315 136 031           1938          CALL    $TYPTX
056.025  072 054 040           1939          DB      ':',', and hit RETURN.'
056.047  012 040 122           1940          DB      NL,' Ready',7,'?'+200Q
056.057                         1941  WARN2.5 EQU     *
056.057  315 300 070           1942          CALL    $RCHAR
056.062  376 012               1943          CPI     NL
056.064  302 057 056           1944          JNZ     WARN2.5
056.067  076 001               1945          MVI     A,1
056.071  062 107 074           1946          STA     INTDSK  SHOW USER SAYS INITIALIZED DISK IS MOUNTED
                                1947
                                1948  *       MOUNT DISK.
                                1949
056.074  056 000               1950          MVI     L,0
056.076  076 010               1951          MVI     A,DC.MOU
056.100  303 157 062           1952          JMP     DDRV            MOUNT DISK AND RETURN
```

```
                                1955  **      RL/ZL - READ AND ZAP LABEL SECTOR.
                                1956  *
                                1957  *       RL READS THE DEVICE'S LABEL SECTOR.
                                1958  *       ZL WRITES A SPECIAL 'DESTROYED BY *DIAG*' LABEL.
                                1959  *         THIS LABEL HAS A ZERO BYTE AS IT'S FIRST CHARACTER,
                                1960  *         SO THAT THE BOOT AND MOUNT ROUTINES WILL KNOW
                                1961  *         ITS A BADDIE.
                                1962  *
                                1963  *       ENTRY   UNIT = UNIT NUMBER
                                1964  *       EXIT    NONE
                                1965  *       USES    ALL
                                1966
                                1967
056.103                         1968  RL      EQU     *
056.103  315 301 055           1969          CALL    RID             REQUIRES INITIALIZED DISK MOUNTED
056.106  076 000               1970          MVI     A,DC.REA
056.110  001 000 001           1971          LXI     B,256
056.113  021 110 074           1972          LXI     D,LABEL
056.116  041 011 000           1973          LXI     H,DDF.LAB
056.121  315 157 062           1974          CALL    DDRV            READ LABEL SECTOR
056.124  332 216 056           1975          JC      RL9                                     /072080/
                                1976
056.127  072 121 074           1977          LDA     LABEL+LAB.VER
056.132  376 027               1978          CPI     17H
056.134  322 145 056           1979          JNC     RL1             DISK INTILIAZED BY VER 1.7 OR LATER
                                1980
                                1981  *       ON VERSIONS PRIOR TO VER 1.7 THE DATA I NEED FROM THE LABEL
                                1982  *       IS NOT THERE. THEREFORE, USE DEFAULT VALUES.
                                1983
056.137  041 240 017           1984          LXI     H,4000          VOLUME SIZE
056.142  042 124 074           1985          SHLD    LABEL+LAB.SIZ
                                1986
                                1987  *       CALCULATE 1ST WRITEABLE SECTOR.  I CAN'T WRITE OVER SYSTEM DATA.
                                1988
056.145                         1989  RL1     EQU     *
056.145  041 012 000           1990          LXI     H,DDF.USR       CAN'T USE SYSTEM AREA
056.150  072 117 074           1991          LDA     LABEL+LAB.SPG
056.153  075                   1992          DCR     A
056.154  315 072 030           1993          CALL    $DADA           ROUND UP TO A TOTALLY FREE BLOCK
056.157  104                   1994          MOV     B,H
056.160  115                   1995          MOV     C,L
056.161  072 117 074           1996          LDA     LABEL+LAB.SPG
056.164  137                   1997          MOV     E,A
056.165  026 000               1998          MVI     D,0
056.167  315 106 030           1999          CALL    $DU66           CONVERT SECTOR TO GROUP #
056.172  175                   2000          MOV     A,L
056.173  376 002               2001          CPI     2
056.175  322 202 056           2002          JNC     RL1.5           CAN'T TOUCH AT LEAST 1ST TWO GROUPS
056.200  056 002               2003          MVI     L,2
056.202                         2004  RL1.5   EQU     *
056.202  072 117 074           2005          LDA     LABEL+LAB.SPG
056.205  353                   2006          XCHG
056.206  315 007 031           2007          CALL    $MU86           CONVERT GROUP TO SECTOR #
056.211  042 115 075           2008          SHLD    SCT1WR          SAVE RESULT FOR LATER USE
                                2009
                                2010  *       RETURN TO CALLER.
```

```
                                2011
056.214  247                   2012            ANA     A
056.215  311                   2013            RET                                             /072080/
                                2014
                                2015  *         CAN'T EVEN READ DISK LABEL.  GOT SERIOUS PROBLEMS.
                                2016
056.216                         2017  RL9       EQU     *                                       *072080*
056.216  315 136 031           2018            CALL    $TYPTX
056.221  007 012 125           2019            DB      BELL,NL,'Unable To Read This Disk At All.'
056.263  012 122 145           2020            DB      NL,'Remember That The Disks Must Be Initialized '
056.340  102 171 040           2021            DB      'By The "INIT"',NL
056.356  120 162 157           2022            DB      'Program Before They Can Be Used By This '
057.026  104 151 141           2023            DB      'Diagnostic.',ENL
057.042  257                   2024            XRA     A
057.043  062 107 074           2025            STA     INTDSK          SAY NO INITIALIZED DISK MOUNTED
057.046  303 327 042           2026            JMP     RESTART
                                2027
057.051                         2028  ZL        EQU     *
057.051  076 040               2029            MVI     A,' '
057.053  001 074 000           2030            LXI     B,LAB.LBL
057.056  041 131 074           2031            LXI     H,LABEL+LAB.LAB
057.061  315 147 064           2032            CALL    FILL
057.064  315 126 070           2033            CALL    $MOVEL
057.067  037 000 172           2034            DW      RZLAL,RZLA,LABEL+LAB.LAB                MOVE IN NEW LABEL
057.075  076 002               2035            MVI     A,LAB.NOD
057.077  062 120 074           2036            STA     LABEL+LAB.VLI   SET NO DIRECTORY ON THIS VOLUME
057.102  076 001               2037            MVI     A,DC.WR1
057.104  001 000 001           2038            LXI     B,256
057.107  021 110 074           2039            LXI     D,LABEL
057.112  041 011 000           2040            LXI     H,DDF.LAB                               /071080/
057.115  315 157 062           2041            CALL    DDRV            WRITE IT                /101080/
057.120  320                   2042            RNC
                                2043
057.121  315 136 031           2044            CALL    $TYPTX
057.124  007 012 125           2045            DB      BELL,NL,'Unable To Write On This Disk',ENL
057.163  257                   2046            XRA     A
057.164  062 107 074           2047            STA     INTDSK          SAY NO INITIALIZED DISK MOUNTED
057.167  303 327 042           2048            JMP     RESTART                                 /101080/
                                2049
057.172  124 150 151           2050  RZLA      DB      'This disk was erased by "TEST"',0
000.037                         2051  RZLAL     EQU     *-RZLA
```

```
                        2055  **      CCP - CHECK FOR CONSTANT PATTERN.
                        2056  *
                        2057  *       CCP CHECKS FOR A CONSTANT ONE BYTE PATTERN OVER THE
                        2058  *       ENTIRE CODED DISK SURFACE.
                        2059  *
                        2060  *       ENTRY   (A) = PATTERN
                        2061  *       EXIT    NONE
                        2062  *       USES    ALL
                        2063
057.231                 2064  CCP     EQU     *                                       /072080/
057.231 062 050 060     2065          STA     CCPC            SAVE PATTERN
                        2066
057.234 001 003 005     2067          LXI     B,5*256+3       PREPARE INTERLEAVE TABLE  /091180/
057.237 041 013 060     2068          LXI     H,CCPA
057.242 315 310 062     2069          CALL    INTRLV                                  /091180/
                        2070
057.245 041 000 000     2071          LXI     H,0             INITIAL SECTOR #
                        2072
                        2073  *       CHECK IF ANY MORE SECTORS TO TEST.
                        2074  *       CAN'T GO PAST VOLUME SIZE.
                        2075
057.250                 2076  CCP1    EQU     *
057.250 042 046 060     2077          SHLD    CCPB            SAVE SECTOR # OF 1ST SECTOR ON TRACK
057.253 315 067 054     2078          CALL    CHKWR2
057.256 320             2079          RNC                     RETURN IF ALL DONE
                        2080
                        2081  *       INITIALIZE INTERLEAVE TABLE POINTER.
                        2082
057.257 021 013 060     2083          LXI     D,CCPA
057.262 325             2084          PUSH    D
                        2085
                        2086  *       READ NEXT SECTOR.
                        2087
057.263                 2088  CCP2    EQU     *
057.263 341             2089          POP     H               GET INTERLEAVE TABLE POINTER
057.264 176             2090          MOV     A,M             GET INTERLEAVE VALUE
057.265 247             2091          ANA     A
057.266 372 377 057     2092          JM      CCP5            AT END OF TABLE
057.271 043             2093          INX     H
057.272 345             2094          PUSH    H               UPDATE INTERLEAVE TABLE POINTER
057.273 052 046 060     2095          LHLD    CCPB            SECTOR # = 1ST SECTOR OF TRACK
057.276 315 101 030     2096          CALL    $DADA.            + INTERLEAVE VALUE
057.301 315 043 054     2097          CALL    CHKWR           CHECK IF WRITEABLE SECTOR
057.304 332 263 057     2098          JC      CCP2              BR IF NOT
                        2099
057.307 076 000         2100          MVI     A,DC.REA        READ OP CODE
057.311 001 000 001     2101          LXI     B,256             COUNT
057.314 021 243 075     2102          LXI     D,BUFF              BUFFER ADDR
057.317 315 133 062     2103          CALL    DDRV.                 DO READ AND COUNT HARD ERRORS
057.322 332 263 057     2104          JC      CCP2                    BR IF ERROR
                        2105
                        2106  *       SCAN BUFFER JUST READ AND SEE IF HARDWARE DIDN'T CATCH ERROR.
                        2107
057.325 072 050 060     2108          LDA     CCPC            RETRIEVE PATTERN
057.330 006 200         2109          MVI     B,128
057.332 041 243 075     2110          LXI     H,BUFF
```

```
                                2111
057.335                         2112  CCP3    EQU     *
057.335  276                   2113          CMP     M
057.336  302 356 057           2114          JNZ     CCP4
057.341  043                   2115          INX     H
057.342  276                   2116          CMP     M
057.343  302 356 057           2117          JNZ     CCP4
057.346  043                   2118          INX     H
057.347  005                   2119          DCR     B
057.350  302 335 057           2120          JNZ     CCP3
057.353  303 263 057           2121          JMP     CCP2
                                2122
                                2123  *       RECORD DATA MISMATCH ERROR AS HARD ERROR.
                                2124
057.356                         2125  CCP4    EQU     *
057.356  315 361 061           2126          CALL    IERR1
057.361  072 106 074           2127          LDA     HERRS
057.364  306 001               2128          ADI     1
057.366  332 263 057           2129          JC      CCP2
057.371  062 106 074           2130          STA     HERRS
057.374  303 263 057           2131          JMP     CCP2
                                2132
                                2133  *       ADVANCE TO NEXT TRACK.
                                2134
057.377                         2135  CCP5    EQU     *
057.377  052 046 060           2136          LHLD    CCPB
060.002  072 124 075           2137          LDA     SPT                             /090980/
060.005  315 101 030           2138          CALL    $DADA.                          /090980/
060.010  303 250 057           2139          JMP     CCP1
                                2140
060.013                         2141  CCPA    DS      0               INTERLEAVE TABLE
060.013  000 005 012           2142          DB      0,5,10,15,20,25,4,9,14,19,24,3,8
060.030  015 022 027           2143          DB      13,18,23,2,7,12,17,22,1,6,11,16,21,80H
                                2144
060.046                         2145  CCPB    DS      2               1ST SECTOR OF TRACK
060.050                         2146  CCPC    DS      1               PATTERN         /072080/
```

```
                                2149  **      WCP - WRITE CONSTANT PATTERN.
                                2150  *
                                2151  *       WCP WRITES A CONSTANT ONE BYTE PATTERN TO THE DISK.
                                2152  *
                                2153  *       ENTRY   (A) = BYTE
                                2154  *       EXIT    NONE
                                2155  *       USES    ALL
                                2156
                                2157
060.051                         2158  WCP     EQU     *                                       /072080/
                                2159
                                2160  *       FILL BUFFER WITH PATTERN.
                                2161
060.051  001 000 001           2162          LXI     B,256
060.054  041 243 075           2163          LXI     H,BUFF
060.057  315 147 064           2164          CALL    FILL
                                2165
                                2166  *       WRITE PATTERN TO DISK.
                                2167
060.062  001 003 005           2168          LXI     B,5*256+3       PREPARE INTERLEAVE TABLE  /091180/
060.065  041 167 060           2169          LXI     H,WCPA
060.070  315 310 062           2170          CALL    INTRLV                                    /091180/
                                2171
060.073  041 000 000           2172          LXI     H,0             INIT SECTOR #
                                2173
                                2174  *       CHECK IF ANY MORE SECTORS TO TEST.
                                2175  *       CAN'T GO PAST VOLUME SIZE.
                                2176
060.076                         2177  WCP1    EQU     *
060.076  042 222 060           2178          SHLD    WCPB            SAVE SECTOR # OF 1ST SECTOR ON TRACK
060.101  315 067 054           2179          CALL    CHKWR2
060.104  320                   2180          RNC                     RETURN IF ALL DONE
                                2181
                                2182  *       INITIALIZE INTERLEAVE TABLE POINTER.
                                2183
060.105  021 167 060           2184          LXI     D,WCPA
060.110  325                   2185          PUSH    D
                                2186
                                2187  *       WRITE NEXT SECTOR.
                                2188
060.111                         2189  WCP2    EQU     *
060.111  341                   2190          POP     H               GET INTERLEAVE TABLE POINTER
060.112  176                   2191          MOV     A,M             GET INTERLEAVE VALUE
060.113  247                   2192          ANA     A
060.114  372 153 060           2193          JM      WCP3            AT END OF TABLE
060.117  043                   2194          INX     H
060.120  345                   2195          PUSH    H               UPDATE INTERLEAVE TABLE POINTER
060.121  052 222 060           2196          LHLD    WCPB            SECTOR # = 1ST SECTOR OF TRACK
060.124  315 101 030           2197          CALL    $DADA.            + INTERLEAVE VALUE
060.127  315 043 054           2198          CALL    CHKWR           CHECK IF WRITEABLE SECTOR
060.132  332 111 060           2199          JC      WCP2              BR IF NOT
                                2200
060.135  076 001               2201          MVI     A,DC.WRI        WRITE OP CODE
060.137  001 000 001           2202          LXI     B,256             COUNT
060.142  021 243 075           2203          LXI     D,BUFF              BUFFER ADDR
060.145  315 133 062           2204          CALL    DDRV.                 DO WRITE AND COUNT HARD ERRORS
```

```
060.150  303 111 060  2205           JMP     WCP2
                      2206
                      2207  *        ADVANCE TO NEXT TRACK.
                      2208
060.153               2209  WCP3     EQU     *
060.153  052 222 060  2210           LHLD    WCPB
060.156  072 124 075  2211           LDA     SPT                          /090980/
060.161  315 101 030  2212           CALL    $DADA.                       /090980/
060.164  303 076 060  2213           JMP     WCP1
                      2214
060.167               2215  WCPA     DS      0               INTERLEAVE TABLE
060.167  000 004 010  2216           DB      0,4,8,12,16,20,24,2,6,10,14,18,22
060.204  001 005 011  2217           DB      1,5,9,13,17,21,25,3,7,11,15,19,23,80H
                      2218
060.222               2219  WCPB     DS      2               1ST SECTOR # OF TRACK        /072080/
```

```
                        2223  **      RRT - RANDOM READ/WRITE TEST
                        2224  *
                        2225  *       RRT RANDOLY SELECTS A SECTOR, AND READS OR
                        2226  *       WRITES IT.
                        2227  *
                        2228  *       EVERY 8 TRYS, RRT CAUSES THE HEAD TO UNLOAD.
                        2229  *
                        2230  *       RRT KEEPS TRACK OF THOSE WHICH HAVE BEEN WRITTEN.
                        2231  *       A SECTOR HAS EITHER BEEN WRITTEN WITH A MODIFIED BIT PATTERN,
                        2232  *       OR A REGULAR BIT PATTERN.
                        2233
                        2234
060.224                 2235  RRT     EQU     *                                       /072080/
                        2236
                        2237  *       ZERO TAG TABLE.
                        2238
060.224  257           2239          XRA     A
060.225  001 365 001    2240          LXI     B,RRTAL
060.230  041 243 076    2241          LXI     H,RRTA
060.233  315 147 064    2242          CALL    FILL
                        2243
                        2244  *       RANDOM SELECTION.
                        2245
060.236  041 350 003    2246          LXI     H,1000          TRY 1000 OF EM
060.241  042 036 061    2247          SHLD    RRTB
                        2248
060.244  315 364 070    2249  RRT00   CALL    $RND            GET RANDOM NUMBER
060.247  174            2250          MOV     A,H
060.250  247            2251          ANA     A               CLEAR CARRY
060.251  037            2252          RAR
060.252  147            2253          MOV     H,A
060.253  175            2254          MOV     A,L
060.254  037            2255          RAR
060.255  157            2256          MOV     L,A
060.256  365            2257          PUSH    PSW             SAVE R/W FLAG
060.257  104            2258          MOV     B,H
060.260  115            2259          MOV     C,L
060.261  052 124 074    2260          LHLD    LABEL+LAB.SIZ
060.264  353            2261          XCHG
060.265  315 106 030    2262          CALL    $DU66           GET SECTOR MODULO VOLUME SIZE
060.270  353            2263          XCHG                    HL=MODULO  DE=QUOTIENT
                        2264
                        2265  *       SECTOR # MUST BE WRITEABLE.
                        2266
060.271  315 043 054    2267          CALL    CHKWR
060.274  322 303 060    2268          JNC     RRT1.3
060.277  361            2269          POP     PSW
060.300  303 244 060    2270          JMP     RRT00           RE-TRY
                        2271
060.303  361            2272  RRT1.3  POP     PSW             'C' SET IF WRITE
060.304  315 346 060    2273          CALL    RRT1.5
060.307  052 036 061    2274          LHLD    RRTB
060.312  053            2275          DCX     H
060.313  042 036 061    2276          SHLD    RRTB
060.316  175            2277          MOV     A,L
060.317  346 007        2278          ANI     07H                                     /072080/
```

```
060.321  302 340 060  2279          JNZ     RRT1.4          NOT TIME
060.324  345          2280          PUSH    H
060.325  076 233      2281          MVI     A,620/4         A DELAY
060.327  315 053 000  2282          CALL    ,DLY              OF 620 MSEC
060.332  076 233      2283          MVI     A,620/4             SHOULD CAUSE
060.334  315 053 000  2284          CALL    ,DLY                  THE HEAD TO UNLOAD
060.337  341          2285          POP     H
060.340               2286  RRT1.4  EQU     *                                       /072080/
060.340  174          2287          MOV     A,H
060.341  265          2288          ORA     L
060.342  302 244 060  2289          JNZ     RRT00           TRY AGAIN
060.345  311          2290          RET
                      2291
060.346  322 002 061  2292  RRT1.5  JNC     RRT2            IS READ
                      2293
                      2294  *       IS WRITE
                      2295
060.351  345          2296          PUSH    H
060.352  104          2297          MOV     B,H             DIVIDE SECTOR #
060.353  115          2298          MOV     C,L               BY 8
060.354  021 010 000  2299          LXI     D,8                 QUOTIENT = DISPLACEMENT INTO TAG TABLE
060.357  315 106 030  2300          CALL    $DU66               REMAINDER = BIT #
060.362  001 243 076  2301          LXI     B,RRTA
060.365  011          2302          DAD     B               ADDR OF ENTRY IN TAG TABLE
060.366  176          2303          MOV     A,M             GET IT
060.367  103          2304          MOV     B,E
060.370  315 056 070  2305          CALL    BITS            SET BIT
060.373  167          2306          MOV     M,A             UPDATE TAG TABLE
060.374  341          2307          POP     H
060.375  076 001      2308          MVI     A,1
060.377  303 041 063  2309          JMP     WLP             WRITE LABEL PATTERN
                      2310
                      2311  *       IS READ
                      2312
061.002               2313  RRT2    EQU     *
061.002  345          2314          PUSH    H
061.003  104          2315          MOV     B,H             DIVIDE SECTOR #
061.004  115          2316          MOV     C,L               BY 8
061.005  021 010 000  2317          LXI     D,8                 QUOTIENT = DISPLACEMENT INTO TAG TABLE
061.010  315 106 030  2318          CALL    $DU66               REMAINDER = BIT #
061.013  001 243 076  2319          LXI     B,RRTA
061.016  011          2320          DAD     B               ADDR OF ENTRY IN TAG TABLE
061.017  257          2321          XRA     A
061.020  103          2322          MOV     B,E
061.021  315 056 070  2323          CALL    BITS            SET MASK
061.024  246          2324          ANA     M               (A) = 0 IF UNMODDED, <>0 IF MODDED
061.025  312 032 061  2325          JZ      RRT2.5
061.030  076 001      2326          MVI     A,1             USE 1 FOR ALL MODDED TAGS
061.032               2327  RRT2.5  EQU     *
061.032  341          2328          POP     H
061.033  303 030 062  2329          JMP     RLP             READ LABEL PATTERN              /072080/
                      2330
061.036  000 000      2331  RRTB    DW      0               ITTERATION COUNT
```

```
                                2335  **      WIP - WRITE ID PATTERN.
                                2336  *
                                2337  *       WIP WRITES THE FIXED ID PATTERN TO ALL SECTORS
                                2338  *
                                2339  *       ENTRY   NONE
                                2340  *       EXIT    NONE
                                2341  *       USES    ALL
                                2342
061.040                         2343  WIP     EQU     *                                       /072080/
                                2344
061.040  001 003 005           2345          LXI     B,5*25643       PREPARE INTERLEAVE TABLE  /091180/
061.043  041 136 061           2346          LXI     H,WIPA
061.046  315 310 062           2347          CALL    INTRLV                                    /091180/
                                2348
061.051  041 000 000           2349          LXI     H,0             INIT SECTOR #
                                2350
                                2351  *       CHECK IF ANY MORE SECTORS TO TEST.
                                2352  *       CAN'T GO PAST VOLUME SIZE.
                                2353
061.054                         2354  WIP1    EQU     *
061.054  042 171 061           2355          SHLD    WIPB            SAVE SECTOR # OF 1ST SECTOR ON TRACK
061.057  315 067 054           2356          CALL    CHKWR2
061.062  320                   2357          RNC                     RETURN IF ALL DONE
                                2358
                                2359  *       INITIALIZE INTERLEAVE TABLE POINTER.
                                2360
061.063  021 136 061           2361          LXI     D,WIPA
061.066  325                   2362          PUSH    D
                                2363
                                2364  *       WRITE NEXT SECTOR.
                                2365
061.067                         2366  WIP2    EQU     *
061.067  341                   2367          POP     H               GET INTERLEAVE TABLE POINTER
061.070  176                   2368          MOV     A,M             GET INTERLEAVE VALUE
061.071  247                   2369          ANA     A
061.072  372 122 061           2370          JM      WIP3            AT END OF TABLE
061.075  043                   2371          INX     H
061.076  345                   2372          PUSH    H               UPDATE INTERLEAVE TABLE POINTER
061.077  052 171 061           2373          LHLD    WIPB            SECTOR # = 1ST SECTOR OF TRACK
061.102  315 101 030           2374          CALL    $DADA.            + INTERLEAVE VALUE
061.105  315 043 054           2375          CALL    CHKWR           CHECK IF WRITEABLE SECTOR
061.110  332 067 061           2376          JC      WIP2              BR IF NOT
                                2377
061.113  257                   2378          XRA     A               INDICATE REGULAR BIT PATTERN
061.114  315 041 063           2379          CALL    WLP             WRITE LABEL PATTERN
061.117  303 067 061           2380          JMP     WIP2
                                2381
                                2382  *       ADVANCE TO NEXT TRACK.
                                2383
061.122                         2384  WIP3    EQU     *
061.122  052 171 061           2385          LHLD    WIPB
061.125  072 124 075           2386          LDA     SPT                                       /090980/
061.130  315 101 030           2387          CALL    $DADA.                                    /090980/
061.133  303 054 061           2388          JMP     WIP1
                                2389
061.136                         2390  WIPA    DS      0               INTERLEAVE TABLE
```

```
061.136  000 004 010  2391          DB      0,4,8,12,16,20,24,2,6,10,14,18,22
061.153  001 005 011  2392          DB      1,5,9,13,17,21,25,3,7,11,15,19,23,80H
                      2393
061.171              2394  WIPB    DS      2               1ST SECTOR # OF TRACK       /072080/
```

```
                        2398  **      CIP - READ ID PATTERN.
                        2399  *
                        2400  *       CIP READS THE FIXED ID PATTERN OF ALL SECTORS
                        2401  *
                        2402  *
                        2403  *       ENTRY   NONE
                        2404  *       EXIT    NONE
                        2405  *       USES    ALL
                        2406
                        2407
061.173                 2408  CIP     EQU     *                                       /072080/
                        2409
061.173 001 004 006     2410          LXI     B,6*256+4       PREPARE INTERLEAVE TABLE /091180/
061.176 041 271 061     2411          LXI     H,CIPA
061.201 315 310 062     2412          CALL    INTRLV                                  /091180/
                        2413
061.204 041 000 000     2414          LXI     H,0             INIT SECTOR #
                        2415
                        2416  *       CHECK IF ANY MORE SECTORS TO TEST.
                        2417  *       CAN'T GO PAST VOLUME SIZE.
                        2418
061.207                 2419  CIP1    EQU     *
061.207 042 324 061     2420          SHLD    CIPB            SAVE SECTOR # OF 1ST SECTOR ON TRACK
061.212 315 067 054     2421          CALL    CHKWR2
061.215 320             2422          RNC                     RETURN IF ALL DONE
                        2423
                        2424  *       INITIALIZE INTERLEAVE TABLE POINTER.
                        2425
061.216 021 271 061     2426          LXI     D,CIPA
061.221 325             2427          PUSH    D
                        2428
                        2429  *       READ NEXT SECTOR.
                        2430
061.222                 2431  CIP2    EQU     *
061.222 341             2432          POP     H               GET INTERLEAVE TABLE POINTER
061.223 176             2433          MOV     A,M             GET INTERLEAVE VALUE
061.224 247             2434          ANA     A
061.225 372 255 061     2435          JM      CIP3            AT END OF TABLE
061.230 043             2436          INX     H
061.231 345             2437          PUSH    H               UPDATE INTERLEAVE TABLE POINTER
061.232 052 324 061     2438          LHLD    CIPB            SECTOR # = 1ST SECTOR OF TRACK
061.235 315 101 030     2439          CALL    $DADA,            + INTERLEAVE VALUE
061.240 315 043 054     2440          CALL    CHKWR           CHECK IF WRITEABLE SECTOR
061.243 332 222 061     2441          JC      CIP2              BR IF NOT
                        2442
061.246 257             2443          XRA     A               INDICATE REGULAR PATTERN
061.247 315 030 062     2444          CALL    RLP             READ LABEL PATTERN
061.252 303 222 061     2445          JMP     CIP2
                        2446
                        2447  *       ADVANCE TO NEXT SECTOR.
                        2448
061.255                 2449  CIP3    EQU     *
061.255 052 324 061     2450          LHLD    CIPB
061.260 072 124 075     2451          LDA     SPT                                     /090980/
061.263 315 101 030     2452          CALL    $DADA,                                  /090980/
061.266 303 207 061     2453          JMP     CIP1
```

```
                                2454
061.271                         2455  CIPA    DS      0               INTERLEAVE TABLE
061.271  000 005 012            2456          DB      0,5,10,15,20,25,4,9,14,19,24,3,8
061.306  015 022 027            2457          DB      13,18,23,2,7,12,17,22,1,6,11,16,21,80H
                                2458
061.324                         2459  CIPB    DS      2               1ST SECTOR # OF TRACK   /072080/
```

```
                                2463  **      CEC - CLEAR ERROR COUNT.
                                2464  *
                                2465  *       CEC CLEARS THE HARD ERROR COUNTER.                      *072080*
                                2466  *
                                2467  *       ENTRY   NONE
                                2468  *       EXIT    NONE
                                2469  *       USES    NONE
                                2470
                                2471
061.326  315 054 031           2472  CEC     CALL    $SAVALL         SAVE REGS
061.331  257                   2473          XRA     A
061.332  062 106 074           2474          STA     HERRS           CLEAR HARD ERRORS               *072080*
061.335  303 047 031           2475          JMP     $RSTALL         RESTORE AND EXIT


                                2477  **      CYR - CHECK FOR YES REPLY.
                                2478  *
                                2479  *       CYR READS A LINE FROM THE CONSOLE, AND CHECKS TO SEE IF IT
                                2480  *       STARTED WITH THE CHARACTERS 'YES'
                                2481  *
                                2482  *       ENTRY   NONE
                                2483  *       EXIT    'Z' SET IF YES
                                2484  *               'Z' CLEAR IF NOT
                                2485  *       USES    ALL
                                2486
                                2487
061.340  041 203 075           2488  CYR     LXI     H,LINE
061.343  315 202 070           2489          CALL    $RTL.           READ LINE
061.346  021 356 061           2490          LXI     D,CYRA
061.351  016 003               2491          MVI     C,3
061.353  303 060 030           2492          JMP     $COMP           COMPARE AND EXIT
                                2493
061.356  131 105 123           2494  CYRA    DB      'YES'


                                2496  **      IERR - INTERNAL ERROR
                                2497  *
                                2498  *       DATA ERROR GOT PAST CHECKSUM
                                2499
                                2500
061.361  315 054 031           2501  IERR1   CALL    $SAVALL
000.001                         2502          IF      .DEBUG.         PRINT MESSAGE IF DEBUGGING
                                2503          CALL    $TYPTX
                                2504          DB      NL,'INTERNAL ERROR #1. CONTACT TECHNICAL CORRESPONDENCE'
                                2505          DB      NL,'FOR ASSISTANCE.',ENL
                                2506          ENDIF
061.364  303 047 031           2507          JMP     $RSTALL
```

```
                                2509  **      PSE - PRINT SIGNIFICANT ERRORS.
                                2510  *
                                2511  *       PSE PRINTS AN ERROR COUNT IFF A NUMBER OF                *072080*
                                2512  *       ERRORS HAS OCCURED.
                                2513  *
                                2514  *       THE ERROR COUNT IS ZEROED WHEN PSE EXITS.                *072080*
                                2515  *
                                2516  *       ENTRY   NONE
                                2517  *       EXIT    NONE
                                2518  *       USES    ALL
                                2519
                                2520
061.367  072 106 074           2521  PSE     LDA     HERRS                                            *072080*
061.372  247                   2522          ANA     A
061.373  310                   2523          RZ                      NO ERRORS TO REPORT              *072080*
                                2524
                                2525  *       HE LOOSES, PRINT AN ERROR COUNT
                                2526
061.374  072 106 074           2527  PSE1    LDA     HERRS                                            *072080*
061.377  117                   2528          MOV     C,A
062.000  006 000               2529          MVI     B,0
062.002  041 024 062           2530          LXI     H,PSEB
062.005  076 003               2531          MVI     A,3
062.007  315 311 070           2532          CALL    $UDDN           UNPACK HARD COUNT
062.012  041 023 062           2533          LXI     H,PSEA
062.015  315 350 064           2534          CALL    PMSG            TYPE/PRINT MESSAGE               *071080*
062.020  303 326 061           2535          JMP     CEC             CLEAR ERROR COUNT AND EXIT
                                2536
062.023  040                   2537  PSEA    DB      ' '             ERROR MESSAGE
062.024  110 110 110           2538  PSEB    DB      'HHH'           HARD COUNT                       *072080*
062.027  240                   2539          DB      ' '+80H


                                2541  **      RLP - READ LABEL PATTERN
                                2542  *
                                2543  *       RLP READS A SECTOR, AND CHECKS THE LABEL PATTERN AND THE
                                2544  *       TYPE PATTERN
                                2545  *
                                2546  *       ENTRY   (A) = TYPE
                                2547  *               (HL) = BLOCK NUMBER
                                2548  *       EXIT    NONE
                                2549  *       USES    A,F,B,C,D,E
                                2550
                                2551
062.030  042 066 063           2552  RLP     SHLD    WLPB
062.033  062 070 063           2553          STA     WLPC
062.036  076 000               2554  RLP0    MVI     A,DC.REA
062.040  001 000 001           2555          LXI     B,256
062.043  021 243 075           2556          LXI     D,BUFF
062.046  315 133 062           2557          CALL    DDRV.                                            *071080*
062.051  332 106 062           2558          JC      RLP2            HARD ERROR, DONT CHECK
062.054  041 243 075           2559          LXI     H,BUFF
062.057  021 066 063           2560          LXI     D,WLPB
062.062  006 200               2561          MVI     B,128           (B) = COUNT                      /072080/
```

```
062.064  032            2562  RLP1    LDAX    D
062.065  276            2563          CMP     M
062.066  302 112 062     2564          JNE     RLPERR
062.071  043            2565          INX     H
062.072  023            2566          INX     D
062.073  032            2567          LDAX    D
062.074  276            2568          CMP     M
062.075  302 112 062     2569          JNE     RLPERR
062.100  043            2570          INX     H
062.101  023            2571          INX     D
062.102  005            2572          DCR     B
062.103  302 064 062     2573          JNZ     RLP1                                    /072080/
062.106  052 066 063     2574  RLP2    LHLD    WLPB
062.111  311            2575          RET                     ALL OK
                        2576
062.112  315 361 061     2577  RLPERR  CALL    IERR1           COUNT IT
000.001                 2578          IF      .DEBUG.
                        2579          LHLD    WLPB
                        2580          CALL    DBGO
                        2581          ENDIF
062.115  072 106 074     2582          LDA     HERRS                                   /072080/
062.120  306 001         2583          ADI     1
062.122  332 106 062     2584          JC      RLP2
062.125  062 106 074     2585          STA     HERRS                                   /072080/
062.130  303 106 062     2586          JMP     RLP2
                        2587


                        2589  **      DDRV. - USE DEVICE DRIVER AND FLAG HARD ERRORS.
                        2590
062.133                 2591  DDRV.   EQU     *
062.133  345            2592          PUSH    H
062.134  315 157 062     2593          CALL    DDRV
062.137  341            2594          POP     H
062.140  320            2595          RNC                     ALL OK
062.141  365            2596          PUSH    PSW             SAVE CODE
000.001                 2597          IF      .DEBUG.
                        2598          CALL    DBGO
                        2599          ENDIF
062.142  072 106 074     2600          LDA     HERRS                                   /072080/
062.145  306 001         2601          ADI     1
062.147  332 155 062     2602          JC      DDRV1.
062.152  062 106 074     2603          STA     HERRS                                   /072080/
062.155  361            2604  DDRV1.  POP     PSW             RESTORE CODE
062.156  311            2605          RET
```

```
                                2607  **      DDRV - DEVICE DRIVER.
                                2608  *
                                2609  *       USE DEVICE ROUTINES IN COMMON DECK H47LIB.                *072080*
                                2610  *
                                2611
062.157                         2612  DDRV    EQU     *
062.157  365                   2613          PUSH    PSW                                               /072080/
062.160  072 110 075           2614          LDA     MYUNIT
062.163  062 061 041           2615          STA     AIO.UNI
062.166  361                   2616          POP     PSW
                                2617
062.167  376 007               2618          CPI     DC.ABT
062.171  312 316 066           2619          JZ      RST             ABORT
                                2620
062.174  376 000               2621          CPI     DC.REA
062.176  312 156 065           2622          JZ      BLKRD           READ
                                2623
062.201  376 001               2624          CPI     DC.WR1
062.203  312 132 065           2625          JZ      BLKWT           WRITE
                                2626
062.206  376 010               2627          CPI     DC.MOU
062.210  312 252 062           2628          JZ      DDRV1           MOUNT
                                2629
                                2630  *       ISSUE BAD NEWS MESSAGE SINCE CODE WAS WRITTEN
                                2631  *       TO ONLY NEED THE ABOVE 3 DEVICE DRIVER ENTRY POINTS.
                                2632
062.213  315 371 064           2633          CALL    PMSG.
062.216  007 012 111           2634          DB      BELL,NL,'Internal Error -- DDRV',NL,80H
062.250  067                   2635          STC
                                2636
                                2637
062.251  311                   2638          RET
                                2639
062.252                         2640  DDRV1   EQU     *
062.252  315 257 066           2641          CALL    RAS             READ AUXILARY STATUS              /090980/
062.255  330                   2642          RC                      RET IF ERROR
                                2643
062.256  107                   2644          MOV     B,A             PUT AUX. STATUS IN TABLE
062.257  072 061 041           2645          LDA     AIO.UNI
062.262  041 127 075           2646          LXI     H,STATBL
062.265  315 101 030           2647          CALL    $DADA.
062.270  160                   2648          MOV     M,B                                               /090980/
                                2649
062.271  247                   2650          ANA     A
062.272  311                   2651          RET                                                       /072080/


                                2653  **      FAS - FETCH AUXILARY STATUS
                                2654  *
                                2655  *       FAS FETCHES THE AUILARY STATUS BYTE FROM THE STATUS
                                2656  *       TABLE FOR THE APPROPRIATE UNIT.
                                2657  *
                                2658  *       ENTRY   'AIO.UNI' = UNIT NUMBER
                                2659  *       EXIT    (A) = AUXILARY STATUS BYTE
```

```
                        2660  *       USES    A,F
                        2661  *
                        2662
        062.273         2663  FAS     EQU     *
        062.273  345   2664          PUSH    H
        062.274  072 061 041  2665          LDA     AIO.UNI
        062.277  041 127 075  2666          LXI     H,STATBL
        062.302  315 101 030  2667          CALL    $DADA.
        062.305  176         2668          MOV     A,M
        062.306  341         2669          POP     H
        062.307  311         2670          RET                                             /090980/

                        2672  **      INTRLV - PREPARE INTERLEAVE TABLE.
                        2673  *
                        2674  *       INTRLV CONSTRUCTS A SECTOR INTERLEAVE TABLE.
                        2675  *
                        2676  *       ENTRY   (BC) = INTERLEAVE FACTOR
                        2677  *                      (B) = FACTOR FOR DOUBLE DENSITY
                        2678  *                      (C) = FACTOR FOR SINGLE DENSITY
                        2679  *               (HL) = FWA OF TABLE
                        2680  *               'SPT' = NUMBER OF SECTORS PER TRACK
                        2681  *       EXIT    NONE
                        2682  *       USES    ALL
                        2683  *
                        2684  *       -- NOTE --
                        2685  *
                        2686  *       THE H47 DRIVER PLACES THE VALUE IN 'SPT' AND 'AUXSTAT'.  THEREFORE,
                        2687  *       THESE ARE ONLY VALID IF A READ OR WRITE HAS BEEN DONE TO THE DISK.
                        2688  *       SINCE ALL CODE WHICH USES THIS ROUTINE SHOULD HAVE READ THE
                        2689  *       LABEL SECTOR, I AM ASSURED THAT THEY ARE VALID.
                        2690  *
        062.310         2691  INTRLV  EQU     *
        062.310  042 037 063  2692          SHLD    INTRLVA         SAVE FWA
                        2693
        062.313  072 117 075  2694          LDA     AUXSTAT         DETERMINE WHICH DENSITY
        062.316  346 100      2695          ANI     A$.ODD
        062.320  302 324 062  2696          JNZ     INTRLVO         DOUBLE DENSITY
        062.323  101         2697          MOV     B,C             USE SINGLE DENSITY FACTOR
                        2698
        000.001         2699          IF      .DEBUG.
                        2700  INTRLVO EQU     *
                        2701          CALL    $TYPTX
                        2702          DB      NL,'ENTER INTERLEAVE FACTOR?',' '+80H
                        2703          LXI     H,LINE
                        2704          CALL    $RTL
                        2705          LDA     LINE
                        2706          SUI     '0'
                        2707          MOV     B,A
                        2708
                        2709          ELSE
        062.324         2710  INTRLVO EQU     *
                        2711          ENDIF
                        2712
```

```
062.324  016 000       2713            MVI     C,0             (C) = TABLE DISPLACEMENT
062.326  121           2714            MOV     D,C             (D) = SECTOR #
                       2715
                       2716  *         CHECK IF DONE.
                       2717
062.327                2718  INTRLV1   EQU     *
062.327  171           2719            MOV     A,C
062.330  052 124 075    2720            LHLD    SPT
062.333  275           2721            CMP     L
062.334  322 026 063    2722            JNC     INTRLV6         AM DONE
                       2723
                       2724  *         DETERMINE SECTOR #.  REG D WILL HOLD THIS VALUE.
                       2725  *         CHECK THAT THE SECTOR # HAS NOT ALREADY BEEN USED.
                       2726  *         IF THAT SECTOR HAS ALREADY BEEN USED, THEN INCREMENT
                       2727  *         AND CHECK AGAIN.
                       2728
062.337                2729  INTRLV2   EQU     *
062.337  131           2730            MOV     E,C
                       2731
062.340                2732  INTRLV3   EQU     *
062.340  035           2733            DCR     E
062.341  173           2734            MOV     A,E
062.342  173           2735            MOV     A,E
062.343  247           2736            ANA     A
062.344  372 377 062    2737            JM      INTRLV4         DONE CHECKING AGAINST PREVIOUS ENTRIES
                       2738
062.347  052 037 063    2739            LHLD    INTRLVA         CHECK THIS ENTRY FOR COLLISION
062.352  315 101 030    2740            CALL    $DADA.
062.355  176           2741            MOV     A,M
062.356  272           2742            CMP     D
062.357  302 340 062    2743            JNE     INTRLV3         NO COLLISION GO CHECK NEXT ENTRY
                       2744
062.362  024           2745            INR     D               SECTOR # USED PREVIOUSLY - TRY NEXT HIGHER
                       2746
062.363  172           2747            MOV     A,D             DO MODULO NUMBER OF SECTORS PER TRACK
062.364  052 124 075    2748            LHLD    SPT
062.367  225           2749            SUB     L
062.370  332 337 062    2750            JC      INTRLV2
062.373  127           2751            MOV     D,A
062.374  303 337 062    2752            JMP     INTRLV2
                       2753
                       2754  *         NEXT SECTOR # DETERMINED.  PLACE IT IN TABLE.
                       2755
062.377                2756  INTRLV4   EQU     *
062.377  052 037 063    2757            LHLD    INTRLVA
063.002  171           2758            MOV     A,C
063.003  315 101 030    2759            CALL    $DADA.
063.006  162           2760            MOV     M,D
                       2761
                       2762  *         ADD INTERLEAVE FACTOR AND GO AGAIN.
                       2763
063.007  014           2764            INR     C               INCREMENT TABLE DISPLACEMENT
                       2765
063.010  172           2766            MOV     A,D             ADD INTERLEAVE FACTOR
063.011  200           2767            ADD     B
063.012  127           2768            MOV     D,A
```

```
                        2769
063.013 052 124 075     2770            LHLD    SPT             DO MODULO NUMBER OF SECTORS PER TRACK
063.016 225             2771            SUB     L
063.017 332 327 062     2772            JC      INTRLV1
063.022 127             2773            MOV     D,A
063.023 303 327 062     2774            JMP     INTRLV1
                        2775
                        2776    *       ALL DONE.  MARK END OF TABLE WITH 80H.
                        2777
063.026                 2778    INTRLV6 EQU     *
063.026 052 037 063     2779            LHLD    INTRLVA
063.031 315 101 030     2780            CALL    $DADA.
063.034 066 200         2781            MVI     M,80H
                        2782
000.001                 2783            IF      .DEBUG.
                        2784            MOV     C,B
                        2785            MVI     B,0
                        2786            MVI     A,2
                        2787            LXI     H,INTRLVB
                        2788            CALL    $UDDN
                        2789            CALL    $TYPTX
                        2790            DB      'PREPARED INTERLEAVE TABLE - FACTOR OF '
                        2791    INTRLVB DS      2
                        2792            DB      ENL
                        2793            LDA     SPT
                        2794            MOV     B,A
                        2795            LHLD    INTRLVA
                        2796    INTRLV7 EQU     *
                        2797            PUSH    B
                        2798            PUSH    H
                        2799            MVI     A,2
                        2800            MVI     B,0
                        2801            MOV     C,M
                        2802            LXI     H,INTRLVC
                        2803            CALL    $UDDN
                        2804            CALL    $TYPTX
                        2805    INTRLVC DS      2
                        2806            DB      ','+80H
                        2807            POP     H
                        2808            POP     B
                        2809            INX     H
                        2810            DCR     B
                        2811            JNZ     INTRLV7
                        2812            CALL    $TYPTX
                        2813            DB      8,' ',NL,ENL
                        2814            ENDIF
                        2815
063.036 311             2816            RET
                        2817
063.037                 2818    INTRLVA DS      2               FWA OF TABLE SAVE SLOT          /091180/
```

```
                                2820  **      WLP - WRITE LABEL PATTERN.
                                2821  *
                                2822  *       WLP WRITES TO A SECTOR A LABEL PATTERN.
                                2823  *
                                2824  *       THE PATTERN IS:
                                2825  *
                                2826  *       DW      SECTOR NUMBER
                                2827  *       DB      FLAG BYTE
                                2828  *       DS      256-3           VARIOUS PATTERNS
                                2829  *
                                2830  *       ENTRY   (A) = FLAG BYTE
                                2831  *               (HL) = SECTOR NUMBER
                                2832  *       EXIT    NONE
                                2833  *       USES    A,F,B,C,D,E
                                2834
                                2835
063.041  042 066 063  2836  WLP     SHLD    WLPB
063.044  062 070 063  2837          STA     WLPC
063.047  076 001      2838          MVI     A,DC.WRI
063.051  001 000 001  2839          LXI     B,256
063.054  021 066 063  2840          LXI     D,WLPB
063.057  315 133 062  2841          CALL    DDRV.                                   *071080*
063.062  052 066 063  2842          LHLD    WLPB
063.065  311          2843          RET
                      2844
063.066  000 000      2845  WLPB    DW      0               BLOCK NUMBER
063.070  000          2846  WLPC    DB      0               ID BYTE
063.071  001 002 004  2847          DB      1,2,4,8,16,32,64,128
063.101  377 376 374  2848          DB      -1,-2,-4,-8,-16,-32,-64,-128
063.111  000 377 000  2849          DB      0,-1,0,-1,0,-1,0,-1,0,-1,0,-1,0,-1,0,-1
063.131  360 360 360  2850          DB      360Q,360Q,360Q,360Q,360Q,360Q,360Q,360Q,360Q,360Q
063.143  360 360 360  2851          DB      360Q,360Q,360Q,360Q,360Q,360Q,360Q
063.151  017 017 017  2852          DB      17Q,17Q,17Q,17Q,17Q,17Q,17Q,17Q,17Q,17Q,17Q,17Q,17Q,17Q,17Q,17Q
063.171  377 377 377  2853          DB      -1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1
063.211  000 000 000  2854          DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
063.231  000 001 002  2855          DB      0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
063.251  020 021 022  2856          DB      16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31
063.271  040 041 042  2857          DB      32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47
063.311  060 061 062  2858          DB      48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63
063.331  106 107 110  2859          DB      70,71,72,73,74,75,76,77,78,79
063.343  120 121 122  2860          DB      80,81,82,83,84,85,86,87,88,89
063.355  132 133 134  2861          DB      90,91,92,93,94,95,96,97,98,99
063.367  144 145 146  2862          DB      100,101,102,103,104,105,106,107,108,109
064.001  156 157 160  2863          DB      110,111,112,113,114,115,116,117,118,119
064.013  170 171 172  2864          DB      120,121,122,123,124,125,126,127,128,129
064.025  202 203 204  2865          DB      130,131,132,133,134,135,136,137,138,139
064.037  214 215 216  2866          DB      140,141,142,143,144,145,146,147,148,149
064.051  226 227 230  2867          DB      150,151,152,153,154,155,156,157,158,159
064.063  240 241 242  2868          DB      160,161,162,163,164,165,166,167,168,169
064.075              2869          DS      256-*+WLPB              FINISH BLOCK
000.001              2870          IF      .DEBUG.
                     2871  DBG     SPACE   4,10
                     2872  ***     DBGO - PRINT SECTOR # WHERE ERROR OCCURRED.
                     2873  *
                     2874  *       ENTRY - (HL) = SECTOR #
                     2875  *       USES - NONE
```

```
                        2876  *
                        2877
                        2878  DBGO    EQU     *
                        2879          PUSH    PSW
                        2880          PUSH    B
                        2881          PUSH    D
                        2882          PUSH    H
                        2883          MOV     B,H
                        2884          MOV     C,L
                        2885          MVI     A,4
                        2886          LXI     H,DBGA1
                        2887          CALL    $UDDN
                        2888          LXI     H,DBGA
                        2889          CALL    PMSG
                        2890          POP     H
                        2891          POP     D
                        2892          POP     B
                        2893          POP     PSW
                        2894          RET
                        2895
                        2896  DBGA    DS      0
                        2897          DB      ' Error on sector # '
                        2898  DBGA1   DS      4
                        2899          DB      NL,80H
                        2900          ENDIF


                        2902  ***     ETL - ENTER LINE.
                        2903  *
                        2904  *       $ETL  - READS A LINE OF DATA INTO A BUFFER UNTIL A
                        2905  *               <CR> IS ENTERED.  DATA IS ONLY PLACED INTO
                        2906  *               THE BUFFER UNTIL THE BUFFER IS FULL.  EXCESS
                        2907  *               CHARACTERS ARE THROWN AWAY.
                        2908  *
                        2909  *       $ETL. - PERFORMS THE SAME FUNCTION AS $ETL AND APPENDS
                        2910  *               A NULL BYTE TO THE END OF THE BUFFER.
                        2911  *
                        2912  *       ENTRY - (A) LENGTH OF BUFFER
                        2913  *               (HL) ADDR OF BUFFER
                        2914  *       EXIT  - (A) # OF CHARACTERS READ
                        2915  *               (HL) ADDR OF LAST BYTE PLACED IN BUFFER + 1
                        2916  *               PSW/C =0 OK
                        2917  *                     =1 # OF CHARACTERS ENTERED EXCEEDED
                        2918  *                        LENGTH OF BUFFER
                        2919  *       USES  - A,H,L,PSW
                        2920  *
                        2921
064.066                 2922  $ETL.   EQU     *
064.066  075           2923          DCR     A               PREDECREMENT FOR NULL BYTE
064.067  315 077 064   2924          CALL    $ETL
064.072  066 000       2925          MVI     M,0
064.074  043           2926          INX     H
064.075  074           2927          INR     A
064.076  311           2928          RET
```

```
                                2929
064.077                         2930  $ETL      EQU     *
064.077  305                    2931            PUSH    B               SAVE BC
064.100  365                    2932            PUSH    PSW             SAVE BUFFER LENGTH
064.101  107                    2933            MOV     B,A
                                2934
                                2935  *         READ CHARACTERS AND PLACE THEM INTO THE BUFFER UNTIL
                                2936  *           1) <CR> ENTERED
                                2937  *           2) BUFFER IS FULL
                                2938
064.102                         2939  $ETL0     EQU     *
064.102  315 300 070           2940            CALL    $RCHAR
064.105  376 012               2941            CPI     NL
064.107  312 127 064           2942            JZ      $ETL1           <CR> ENTERED (HDOS TRANSLATES TO <NL>)
064.112  117                    2943            MOV     C,A
064.113  170                    2944            MOV     A,B
064.114  326 001               2945            SUI     1
064.116  332 133 064           2946            JC      $ETL2           BUFFER FULL
064.121  107                    2947            MOV     B,A
064.122  161                    2948            MOV     M,C
064.123  043                    2949            INX     H
064.124  303 102 064           2950            JMP     $ETL0
                                2951
                                2952  *         <CR> ENTERED -- DETERMINE NUMBER OF CHARACTERS ENTERED AND RETURN
                                2953
064.127                         2954  $ETL1     EQU     *
064.127  361                    2955            POP     PSW
064.130  220                    2956            SUB     B
064.131  301                    2957            POP     B
064.132  311                    2958            RET
                                2959
                                2960  *         MORE CHARACTERS ENTERED THAN BUFFER CAN HOLD.
                                2961  *         THROW AWAY CHARACTERS AND WAIT FOR <CR> TO BE ENTERED.
                                2962
064.133                         2963  $ETL2     EQU     *
064.133  315 300 070           2964            CALL    $RCHAR
064.136  376 012               2965            CPI     NL
064.140  302 133 064           2966            JNZ     $ETL2
064.143  361                    2967            POP     PSW
064.144  301                    2968            POP     B
064.145  067                    2969            STC
064.146  311                    2970            RET


                                2972  ***       FILL - FILL MEMORY WITH BYTE
                                2973  *
                                2974  *         FILL FILLS MEMORY WITH A CONSTANT BYTE VALUE.
                                2975  *
                                2976  *         ENTRY - (A)  = CONSTANT BYTE VALUE
                                2977  *                 (BC) = COUNT
                                2978  *                 (HL) = FWA OF MEMORY
                                2979  *         EXIT  - (HL) = LWA+1
                                2980  *         USES  - A,B,C,H,L,PSW
                                2981  *
```

```
                                2982
064.147                         2983  FILL    EQU     *
064.147  325                   2984          PUSH    D
064.150  127                   2985          MOV     D,A
064.151                         2986  FILL1   EQU     *
064.151  162                   2987          MOV     M,D
064.152  043                   2988          INX     H
064.153  013                   2989          DCX     B
064.154  170                   2990          MOV     A,B
064.155  261                   2991          ORA     C
064.156  302 151 064           2992          JNZ     FILL1
064.161  321                   2993          POP     D
064.162  311                   2994          RET


                                2996  ***     FILLW - FILL MEMORY WITH WORD
                                2997  *
                                2998  *       FILLW FILLS MEMORY WITH A CONSTANT WORD VALUE.
                                2999  *
                                3000  *       ENTRY - (BC) = COUNT
                                3001  *               (DE) = CONSTANT WORD VALUE
                                3002  *               (HL) = FWA OF MEMORY
                                3003  *       EXIT  - (HL) = LWA+1
                                3004  *       USES  - A,B,C,H,L,PSW
                                3005  *
                                3006
064.163                         3007  FILLW   EQU     *
064.163  163                   3008          MOV     M,E
064.164  043                   3009          INX     H
064.165  162                   3010          MOV     M,D
064.166  043                   3011          INX     H
064.167  013                   3012          DCX     B
064.170  170                   3013          MOV     A,B
064.171  261                   3014          ORA     C
064.172  302 163 064           3015          JNZ     FILLW
064.175  311                   3016          RET


                                3018  ***     FNP - FORCE NEW PAGE
                                3019  *
                                3020  *       FNP -- FORCE NEW PAGE AND PRINT HEADING.
                                3021  *
                                3022  *       FNP. - FORCE NEW PAGE W/O PRINTING HEADING.
                                3023  *
                                3024  *       ENTRY - NONE
                                3025  *       EXIT  - NONE
                                3026  *       USES  - ALL
                                3027  *
                                3028
                                3029  *       FORCE NEW PAGE WITH HEADING.
                                3030
064.176                         3031  FNP     EQU     *
```

```
064.176  076 001       3032           MVI     A,1
064.200  062 346 064   3033           STA     FNPB            INDICATE PRINT HEADING
                       3034
                       3035  *        IF 1ST TIME HERE, THEN ASSUME PAPER IS AT TOP OF NEW FORM.
                       3036  *        DON'T ISSUE FORMFEED.
                       3037
064.203                3038  FNP0     EQU     *
064.203  072 345 064   3039           LDA     FNPA
064.206  247           3040           ANA     A
064.207  312 221 064   3041           JZ      FNP1            NOT 1ST TIME HERE
064.212  257           3042           XRA     A
064.213  062 345 064   3043           STA     FNPA            CLEAR 1ST TIME FLAG
064.216  303 235 064   3044           JMP     FNP2
                       3045
                       3046  *        ISSUE FORMFEED TO DEVICE.
                       3047
064.221                3048  FNP1     EQU     *
064.221  001 001 000   3049           LXI     B,1
064.224  021 347 064   3050           LXI     D,FNPC
064.227  041 141 050   3051           LXI     H,RPTD
064.232  315 200 072   3052           CALL    $FWRIB.
                       3053
                       3054  *        PRINT HEADING IF DESIRED.
                       3055
064.235                3056  FNP2     EQU     *
064.235  072 346 064   3057           LDA     FNPB
064.240  247           3058           ANA     A
064.241  310           3059           RZ                      HEADING NOT DESIRED
064.242  072 137 050   3060           LDA     RPTB
064.245  306 001       3061           ADI     1
064.247  047           3062           DAA
064.250  062 137 050   3063           STA     RPTB            INCREMENT PAGE NUMBER
064.253  107           3064           MOV     B,A
064.254  037           3065           RAR
064.255  037           3066           RAR
064.256  037           3067           RAR
064.257  037           3068           RAR
064.260  346 017       3069           ANI     0FH
064.262  302 267 064   3070           JNZ     FNP3
064.265  076 360       3071           MVI     A,'0'-'0'
064.267                3072  FNP3     EQU     *
064.267  306 060       3073           ADI     '0'
064.271  062 133 050   3074           STA     RPTA3           CONVERT 10'S DIGIT TO ASCII
064.274  170           3075           MOV     A,B
064.275  346 017       3076           ANI     0FH
064.277  306 060       3077           ADI     '0'
064.301  062 134 050   3078           STA     RPTA3+1         CONVERT UNIT'S DIGIT TO ASCII
064.304  001 116 000   3079           LXI     B,RPTAL
064.307  021 021 050   3080           LXI     D,RPTA
064.312  041 141 050   3081           LXI     H,RPTD
064.315  315 200 072   3082           CALL    $FWRIB.         WRITE HEADING
064.320  076 074       3083           MVI     A,FNPD
064.322  326 003       3084           SUI     3
064.324  062 127 065   3085           STA     WRTLA           RESET LINE COUNTER - PAGE HEADING
064.327  311           3086           RET
                       3087
```

```
                                3088  *       FORCE NEW PAGE WITHOUT HEADING.
                                3089
064.330                         3090  FNP.    EQU     *
064.330  257                   3091          XRA     A
064.331  062 346 064           3092          STA     FNPB            INDICATE NO HEADING
064.334  315 203 064           3093          CALL    FNPO            FORCE NEW PAGE
064.337  076 074               3094          MVI     A,FNPD
064.341  062 127 065           3095          STA     WRTLA           RESET LINE COUNTER
064.344  311                   3096          RET
                                3097
064.345  001                   3098  FNPA    DB      1               FLAG := <>0 1ST TIME HERE
064.346                         3099  FNPB    DS      1               FLAG := <>0 PRINT HEADING
064.347  014                   3100  FNPC    DB      FF              FORMFEED
000.074                         3101  FNPD    EQU     60              LINES/PAGE


                                3103  ***     PMSG - PRINT MESSAGE.
                                3104  *
                                3105  *       PMSG -- PRINT MESSAGE ON BOTH THE TERMINAL AND THE
                                3106  *                HARDCOPY DEVICE.
                                3107  *
                                3108  *       ENTRY - (HL) = FWA OF MESSAGE
                                3109  *       USES  - NONE
                                3110  *
                                3111
064.350                         3112  PMSG    EQU     *
064.350  365                   3113          PUSH    PSW
064.351  305                   3114          PUSH    B
064.352  325                   3115          PUSH    D
064.353  345                   3116          PUSH    H
064.354  315 144 031           3117          CALL    $TYPTX.         PRINT LINE ON TERMINAL
064.357  341                   3118          POP     H
064.360  345                   3119          PUSH    H
064.361  315 012 065           3120          CALL    WRTL            PRINT LINE ON HARDCOPY DEVICE
064.364  341                   3121          POP     H
064.365  321                   3122          POP     D
064.366  301                   3123          POP     B
064.367  361                   3124          POP     PSW
064.370  311                   3125          RET
                                3126
                                3127  *       PMSG. - PRINT MESSAGE ON BOTH THE TERMINAL AND
                                3128  *               THE HARDCOPY DEVICE
                                3129  *
                                3130  *       ENTRY - (SP) = FWA OF MESSAGE
                                3131  *       EXIT  - (SP) = LWA+1 OF MESSAGE
                                3132  *       USES  - NONE
                                3133
064.371                         3134  PMSG.   EQU     *
064.371  343                   3135          XTHL
064.372  365                   3136          PUSH    PSW
064.373  305                   3137          PUSH    B
064.374  325                   3138          PUSH    D
064.375  345                   3139          PUSH    H
064.376  315 012 065           3140          CALL    WRTL            PRINT MESSAGE ON HARDCOPY DEVICE
```

```
065.001  341            3141            POP     H
065.002  315 144 031    3142            CALL    $TYPTX.         PRINT MESSAGE ON TERMINAL
065.005  321            3143            POP     D
065.006  301            3144            POP     B
065.007  361            3145            POP     PSW
065.010  343            3146            XTHL
065.011  311            3147            RET


                        3149  ***       WRTL - WRITE LINES ON HARDCOPY DEVICE.
                        3150  *
                        3151  *         WRTL -- WRITES LINES TO THE HARDCOPY DEVICE KEEPING
                        3152  *                 TRACK OF A LINE COUNTER.  WHEN THE LINE
                        3153  *                 COUNTER REACHES ZERO AND ANOTHER LINE
                        3154  *                 IS TO BE PRINTED, A NEW PAGE IS FORCED.
                        3155  *
                        3156  *         ENTRY - (HL) = FWA OF BUFFER
                        3157  *         EXIT  - (HL) = LWA+1
                        3158  *         USES  - ALL
                        3159  *
                        3160  *         NOTE:
                        3161  *           THE LAST BYTE TO BE WRITTEN IS INDICATED BY SETTING THE
                        3162  *             SIGN BIT OF THE BYTE TO A 1.  THE <NL> CHARACTER SHOULD
                        3163  *           NOT HAVE THE SIGN BIT SET, SINCE SOME DEVICE DRIVERS
                        3164  *             ONLY CHECK FOR <NL> BUT NOT <NL>+80H.  THIS CAN CAUSE
                        3165  *           PROBLEMS IF THE DEVICE DRIVER WANTS TO TRANSLATE <NL>'S
                        3166  *             TO <CR><LF>.
                        3167  *
                        3168
065.012                 3169  WRTL      EQU     *
065.012  072 140 050    3170            LDA     RPTC
065.015  247            3171            ANA     A
065.016  310            3172            RZ                      HARDCOPY NOT REQUESTED
                        3173
065.017                 3174  WRTL0     EQU     *
065.017  042 130 065    3175            SHLD    WRTLB           SAVE BUFFER POINTER
065.022  072 127 065    3176            LDA     WRTLA
065.025  247            3177            ANA     A
065.026  302 036 065    3178            JNZ     WRTL1           NOT TIME FOR NEW PAGE
065.031  345            3179            PUSH    H
065.032  315 176 064    3180            CALL    FNP             FORCE NEW PAGE WITH HEADING
065.035  341            3181            POP     H
                        3182
                        3183  *         SCAN BUFFER FOR NEXT <NL> OR END OF BUFFER.
                        3184
065.036                 3185  WRTL1     EQU     *
065.036  176            3186            MOV     A,M
065.037  043            3187            INX     H
065.040  107            3188            MOV     B,A
065.041  346 177        3189            ANI     07FH
065.043  376 012        3190            CPI     NL
065.045  312 061 065    3191            JZ      WRTL2           BR IF <NL>
065.050  170            3192            MOV     A,B
065.051  247            3193            ANA     A
```

```
065.052  362 036 065  3194           JP      WRTL1           NOT END OF BUFFER
                      3195
                      3196  *       END OF BUFFER.  WRITE WHAT WE HAVE.
                      3197
065.055  315 101 065  3198           CALL    WRTL3
065.060  311          3199           RET
                      3200
                      3201  *       WRITE A LINE OF DATA AND UPDATE LINE COUNTER.
                      3202
065.061               3203  WRTL2    EQU     *
065.061  315 101 065  3204           CALL    WRTL3           WRITE LINE
065.064  072 127 065  3205           LDA     WRTLA
065.067  075          3206           DCR     A
065.070  062 127 065  3207           STA     WRTLA           UPDATE LINE COUNTER
065.073  170          3208           MOV     A,B             0, <NL> ALSO
065.074  247          3209           ANA     A                 END OF BUFFER
065.075  362 017 065  3210           JP      WRTL0               BR IF NOT
065.100  311          3211           RET
                      3212
                      3213  *       WRITE  TO THE HARDCOPY DEVICE.
                      3214
065.101               3215  WRTL3    EQU     *
065.101  305          3216           PUSH    B
065.102  345          3217           PUSH    H
065.103  353          3218           XCHG                    (DE)=LWA+1
065.104  052 130 065  3219           LHLD    WRTLB           (HL)=FWA
065.107  353          3220           XCHG                    (DE)=FWA (HL)=LWA+1
                      3221
                      3222  *       CALCULATE COUNT.
                      3223
065.110  175          3224           MOV     A,L
065.111  223          3225           SUB     E
065.112  117          3226           MOV     C,A
065.113  174          3227           MOV     A,H
065.114  232          3228           SBB     D
065.115  107          3229           MOV     B,A
                      3230
065.116  041 141 050  3231           LXI     H,RPTD
065.121  315 200 072  3232           CALL    $FWRIB.
065.124  341          3233           POP     H
065.125  301          3234           POP     B
065.126  311          3235           RET
                      3236
065.127  000          3237  WRTLA    DB      0               LINE COUNTER
065.130               3238  WRTLB    DS      2               BUFFER POINTER                  /071080/


000.000               3240  .BLKW    EQU     0                                               /072080/
000.001               3241  .SMALL   EQU     1
065.132               3242           XTEXT   H47LIB                                          /072080/
```

```
                        3245X ***     Assembly Constants
                        3246X *
                        3247X *
                        3248X *       .BLKW   Used to conditional WRITE operations in and out.
                        3249X *
                        3250X *       .SMALL  Used to conditional BLK operations out
                        3251X *               memory, or minimal run-time.
                        3252X *

000.001                 3254X         IF      .SMALL
                        3255X         ELSE

                        3257X **      BLK     -  Block
                        3258X *
                        3259X *       BLK repeatedly READ/WRITEs the data until all of the data
                        3260X *       is transfered.  Data is always transfered so that it will
                        3261X *       not wrap over a single side track boundary, so as to avoid
                        3262X *       the multiple sector algorithem problem.
                        3263X *
                        3264X *
                        3265X *       ENTRY:  BC      = total count (should be a multiple of 256)
                        3266X *               DE      = buffer address
                        3267X *               HL      = block number
                        3268X *
                        3269X *       EXIT:   PSW     = 'C' set   if    error
                        3270X *                         'C' clear if no error
                        3271X *
                        3272X *       USES:   ALL
                        3273X *
                        3274X
065.132  076 010       3275X BLKWT   MVI     A,DD.WRIB       block write
065.134  062 326 065   3276X         STA     BLKB
065.137  315 354 066   3277X         CALL    SDE             Default error is WRITE
065.142  023           3278X         DB      EC.WF
065.143  345           3279X         PUSH    H
065.144  041 040 066   3280X         LXI     H,OUTB
065.147  042 324 065   3281X         SHLD    BLKA            set block operation as output
065.152  341           3282X         POP     H
065.153  303 177 065   3283X         JMP     BLK1
                        3284X
065.156  076 007       3285X BLKRD   MVI     A,DD.REAB       block read
065.160  062 326 065   3286X         STA     BLKB
065.163  315 354 066   3287X         CALL    SDE             Default error is READ
065.166  022           3288X         DB      EC.RF
065.167  345           3289X         PUSH    H
065.170  041 142 066   3290X         LXI     H,PINB
065.173  042 324 065   3291X         SHLD    BLKA            set block operation as input
065.176  341           3292X         POP     H
000.000                 3293X         ERRNZ   *-BLK1
                        3294X
```

```
065.177  315 364 066  3295X BLK1   CALL   SDP            Set-Up device parameters
065.202  072 274 067  3296X         LDA    DEFERR
065.205  330          3297X         RC                    Illegal parameters
                       3298X
065.206  170          3299X BLK2   MOV    A,B
065.207  261          3300X         ORA    C
065.210  310          3301X         RZ                    all finished with the block operation
                       3302X
065.211  305          3303X         PUSH   B
065.212  315 237 065  3304X         CALL   BLK3
065.215  301          3305X         POP    B
065.216  330          3306X         RC                    Error
                       3307X
065.217  171          3308X         MOV    A,C
065.220  225          3309X         SUB    L
065.221  117          3310X         MOV    C,A
065.222  170          3311X         MOV    A,B            decrement the bytes read count
065.223  234          3312X         SBB    H
065.224  107          3313X         MOV    B,A
                       3314X
065.225  315 303 067  3315X         CALL   USN            Update sector number
                       3316X
065.230  315 367 067  3317X         CALL   WDN
065.233  322 206 065  3318X         JNC    BLK2           No errors
                       3319X
065.236  311          3320X         RET                   ERROR waiting for DONE


                       3322X **     BLK3
                       3323X *
                       3324X *      EXIT:  HL      = bytes actually read
                       3325X *
                       3326X
065.237                3327X BLK3   EQU    *
                       3328X
                       3329X *      Compute Transfer Size
                       3330X
065.237  140          3331X         MOV    H,B            HL = BC ; Initialize byte count
065.240  151          3332X         MOV    L,C
                       3333X
065.241  171          3334X         MOV    A,C
065.242  247          3335X         ANA    A
065.243  312 247 065  3336X         JZ     BLK4
065.246  004          3337X         INR    B              round sector count up for partial sector
065.247                3338X BLK4   EQU    *
                       3339X
065.247  170          3340X         MOV    A,B
065.250  062 125 075  3341X         STA    STC            Initialize sector count
                       3342X
065.253  345          3343X         PUSH   H
065.254  072 124 075  3344X         LDA    SPT            A  = Sectors per Track
065.257  052 121 075  3345X         LHLD   SECTOR
065.262  225          3346X         SUB    L
065.263  074          3347X         INR    A              A  = maximum number of sectors left this track
```

```
065.264  270          3348X          CMP     B
065.265  341          3349X          POP     H
065.266  322 277 065  3350X          JNC     BLK5            Can read all that we need
                      3351X
065.271  062 125 075  3352X          STA     STC             Update sector count for end of track
065.274  147          3353X          MOV     H,A
065.275  056 000      3354X          MVI     L,0             HL = count for the rest of this track
065.277               3355X BLK5     EQU     *
                      3356X
065.277  315 365 065  3357X          CALL    LSC             Set the transfer count
065.302  330          3358X          RC                      ERROR
                      3359X
065.303  072 326 065  3360X          LDA     BLKB
065.306  315 333 065  3361X          CALL    COM.            command
065.311  330          3362X          RC                      ERROR issuing command
                      3363X
065.312  315 275 067  3364X          CALL    TRK             track
065.315  315 112 067  3365X          CALL    SUS             side/unit/sector
065.320  330          3366X          RC                      ERROR bit is set
                      3367X
065.321  104          3368X          MOV     B,H
065.322  115          3369X          MOV     C,L             BC = actual byte transfer count
065.323  303 377 377  3370X          JMP     -1              Enter data transfer processor
065.324               3371X BLKA     EQU     *-2
                      3372X
065.326  000          3373X BLKB     DB      0               Transfer Command (Read/Write)
001.000               3374X BLKC     EQU     256


                      3376X **       COM     - Command
                      3377X *
                      3378X *        COM outputs a command byte.
                      3379X *
                      3380X *
                      3381X *        ENTRY:  A       = command
                      3382X *
                      3383X *        EXIT:   PSW     = 'C' SET   IF    ERROR
                      3384X *                        = 'C' CLEAR IF NO ERROR
                      3385X *
                      3386X *        USES:   PSW
                      3387X *
                      3388X
065.327  343          3389X COM      XTHL
065.330  176          3390X          MOV     A,M             Fetch the command byte
065.331  043          3391X          INX     H
065.332  343          3392X          XTHL
                      3393X
065.333  365          3394X COM.     PUSH    PSW
065.334  315 367 067  3395X          CALL    WDN
065.337  332 352 065  3396X          JC      COM1            ERROR
065.342  361          3397X          POP     PSW
065.343  315 032 066  3398X COM..    CALL    OUT.
065.346  315 355 065  3399X          CALL    DLY
065.351  311          3400X          RET
```

```
                          3401X
065.352  063            3402X COM1    INX     SP
065.353  063            3403X         INX     SP
065.354  311            3404X         RET                     Return with error from WDN


                          3406X **      DLY     -  Delay
                          3407X *
                          3408X *       DLY delays for a short time.
                          3409X *
                          3410X *       ENTRY:  NONE
                          3411X *
                          3412X *       EXIT:   A       = 0
                          3413X *
                          3414X *       USES:   PSW
                          3415X *
                          3416X
065.355  076 040          3417X DLY     MVI     A,040Q
065.357  247              3418X         ANA     A               F = 'NC'
065.360  075              3419X DLY1    DCR     A
065.361  302 360 065      3420X         JNZ     DLY1            Wait longer
065.364  311              3421X         RET


                          3423X **      LSC     -  Load Sector Count
                          3424X *
                          3425X *       LSC loads the sector count for subsequent operations.
                          3426X *
                          3427X *       ENTRY:  AUXSTAT initialized
                          3428X *
                          3429X *       EXIT:   PSW     = 'C' CLEAR if NO ERROR
                          3430X *                         'C' SET   if    ERROR
                          3431X *
                          3432X *       USES:   PSW
                          3433X *
                          3434X
065.365  315 327 065      3435X LSC     CALL    COM
065.370  003              3436X         DB      DD.LSC
065.371  330              3437X         RC                      Tough luck if COM returns errors
                          3438X
065.372  257              3439X         XRA     A               High order byte
065.373  315 022 066      3440X         CALL    OUT
065.376  330              3441X         RC
                          3442X
065.377  072 117 075      3443X         LDA     AUXSTAT
066.002  346 100          3444X         ANI     AS.ODD          Check for Double Density
066.004  072 125 075      3445X         LDA     STC
066.007  302 013 066      3446X         JNZ     LSC1            Double Density
066.012  207              3447X         ADD     A               Double Count for single density 128 byte sect.
066.013  315 022 066      3448X LSC1    CALL    OUT             Low order byte
066.016  330              3449X         RC
                          3450X
```

```
066.017  303 367 067  3451X          JMP     WDN


                      3453X **       OUT     - Output
                      3454X *
                      3455X *        OUT outputs a byte to the port with a *S,DTR* handshake.
                      3456X *
                      3457X *
                      3458X *        ENTRY:  A       = byte
                      3459X *
                      3460X *        EXIT:   PSW     = 'C'   if ERROR
                      3461X *                         'NC'  if NO Error, no byte output
                      3462X *
                      3463X *        USES:   PSW
                      3464X *
                      3465X
066.022  365          3466X OUT      PUSH    PSW
066.023  315 020 070  3467X          CALL    WTR             Wait for DTR
066.026  332 035 066  3468X          JC      OUTO
066.031  361          3469X          POP     PSW
                      3470X
066.032  323 171      3471X OUT,     OUT     D.DAT           OUTPUT TO THE DATA PORT
066.034  311          3472X          RET
                      3473X
066.035  063          3474X OUTO     INX     SP              Return with error from WTR
066.036  063          3475X          INX     SP
066.037  311          3476X          RET
                      3477X          ENDIF


000.000               3479X          IF      .BLKW


                      3481X **       OUTB    - Output Block
                      3482X *
                      3483X *        OUTB outputs a block.  This is one of the more critical
                      3484X *        routines as far as time goes in transfering data.  This
                      3485X *        routine should be highly tuned.
                      3486X *
                      3487X *
                      3488X *        ENTRY:  BC      =  count
                      3489X *                DE      =  buffer address
                      3490X *
                      3491X *        EXIT:   PSW     =  'C' clear if NO error
                      3492X *                        =  'C' set   if    error
                      3493X *                           BC = count remaining
                      3494X *
                      3495X *        USES:   ALL
                      3496X *
                      3497X
```

```
066.040  170           3498X OUTB   MOV     A,B
066.041  261           3499X        ORA     C
066.042  312 367 067   3500X        JZ      WDN             Finished, wait for DONE
                       3501X
066.045  305           3502X        PUSH    B
066.046  315 066 066   3503X        CALL    OUT128
066.051  301           3504X        POP     B
066.052  330           3505X        RC                      Transfer Error
                       3506X
066.053  345           3507X        PUSH    H
066.054  041 200 377   3508X        LXI     H,-128
066.057  011           3509X        DAD     B
066.060  104           3510X        MOV     B,H
066.061  115           3511X        MOV     C,L
066.062  341           3512X        POP     H
066.063  303 040 066   3513X        JMP     OUTB


                       3515X **     OUT128  - OUT 128
                       3516X *
                       3517X *      OUT128 outputs 128 bytes from the data input port.
                       3518X *      The first and last bytes are transfered via hand-
                       3519X *      shake, the rest are transferred as good.  The
                       3520X *      reason 128 was chosen, as it is the minimum trans-
                       3521X *      fer size if the H47 code is somehow lost.
                       3522X *
                       3523X *      NOTE:   This code assumes that the H47 accepts
                       3524X *              bytes sufficiently fast in the MAIN loop.
                       3525X *
                       3526X *      ENTRY:  DE      = buffer
                       3527X *
                       3528X *      EXIT:   PSW     = 'C' CLEAR if NO error
                       3529X *                        DE = DE advanced
                       3530X *                    'C' SET   if    error
                       3531X *                        A  = Error Code
                       3532X *
                       3533X *      USES:   PSW,BC,DE
                       3534X *
                       3535X
066.066  315 020 070   3536X OUT128  CALL    WTR
066.071  330           3537X        RC                      Synchronization error
                       3538X
                       3539X *      Output 127 bytes
                       3540X
066.072  016 177       3541X        MVI     C,127
066.074  333 170       3542X OUT1   IN      D.STA
066.076  346 240       3543X        ANI     S.DTR+S.DON
066.100  372 111 066   3544X        JM      OUT2            S.DTR is set, is not done
000.000                3545X        ERRNZ   S.DTR-200Q
                       3546X
066.103  312 074 066   3547X        JZ      OUT1            is not done yet
                       3548X
066.106  303 161 067   3549X        JMP     TEB.            Generate error, and examine status
                       3550X
```

```
066.111  032          3551X OUT2   LDAX    D
066.112  323 171      3552X        OUT     D.DAT          output a byte
066.114  023          3553X        INX     D
066.115  015          3554X        DCR     C
066.116  302 074 066  3555X        JNZ     OUT1
                      3556X
                      3557X *      Handshake last byte
                      3558X
066.121  315 020 070  3559X        CALL    WTR
066.124  330          3560X        RC                     Synchronization error
                      3561X
066.125  032          3562X        LDAX    D
066.126  323 171      3563X        OUT     D.DAT
066.130  023          3564X        INX     D
066.131  247          3565X        ANA     A              Clear 'C'
066.132  311          3566X        RET
                      3567X        ENDIF
000.001               3568X        IF      .SMALL
                      3569X        ELSE


                      3571X **     PIN     =   Input
                      3572X *
                      3573X *      PIN inputs a byte from the data data port.
                      3574X *
                      3575X *
                      3576X *      ENTRY:  NONE
                      3577X *
                      3578X *      EXIT:   PSW     = 'C'   if ERROR
                      3579X *                          A   = Error Code
                      3580X *                        'NC'  if NO Error
                      3581X *                          A   = Byte
                      3582X *
                      3583X *      USES:   PSW
                      3584X *
                      3585X
066.133  315 020 070  3586X PIN    CALL    WTR            Wait for DTR
066.136  330          3587X        RC
                      3588X
066.137  333 171      3589X PIN.   IN      D.DAT
066.141  311          3590X        RET


                      3592X **     PINB    =   Input Block
                      3593X *
                      3594X *      PINB inputs a block.  This is one of the more critical
                      3595X *      routines as far as time goes in transferins data.  This
                      3596X *      routine should be highly tuned.
                      3597X *
                      3598X *
                      3599X *      ENTRY:  BC      =  count
                      3600X *                 DE      =  buffer address


```

```
                        3601X *
                        3602X *       EXIT:   PSW     =  'C' clear if NO error
                        3603X *                        =  'C' set   if   error
                        3604X *                          A  = Error Code
                        3605X *                          BC = count remaining
                        3606X *
                        3607X *       USES:   ALL
                        3608X *
                        3609X
066.142  170           3610X PINB    MOV     A,B
066.143  247           3611X         ANA     A
066.144  312 161 066   3612X         JZ      PINB2           Need less than one sector
                        3613X
066.147  305           3614X PINB1   PUSH    B
066.150  315 222 066   3615X         CALL    PIN256          read one sector
066.153  301           3616X         POP     B
066.154  330           3617X         RC                      ERROR
                        3618X
066.155  005           3619X         DCR     B               Count the bytes read
066.156  302 147 066   3620X         JNZ     PINB1
                        3621X
066.161  171           3622X PINB2   MOV     A,C
066.162  247           3623X         ANA     A
066.163  312 367 067   3624X         JZ      WDN             Finished, don't need partial sector
                        3625X
                        3626X *       Read any partial sectors
                        3627X
066.166  305           3628X         PUSH    B               C  = bytes left to read in partial sector
066.167  315 230 066   3629X         CALL    PIN1.           Read a partial sector
066.172  301           3630X         POP     B
066.173  330           3631X         RC                      ERROR
                        3632X
066.174  333 170       3633X PINB3   IN      D.STA
066.176  346 240       3634X         ANI     S.DTR+S.DON
066.200  372 211 066   3635X         JM      PINB4           H47 has a byte
000.000                 3636X         ERRNZ   S.DTR-200Q
                        3637X
066.203  312 174 066   3638X         JZ      PINB3           DONE is not set
                        3639X
066.206  303 161 067   3640X         JMP     TEB.            Generate error, and look at status
                        3641X
066.211  333 171       3642X PINB4   IN      D.DAT           Eat the byte
066.213  014           3643X         INR     C
066.214  302 174 066   3644X         JNZ     PINB3
                        3645X
066.217  303 367 067   3646X         JMP     WDN             DONE accepting bytes
```

```
                        3648X **      PIN256  - PIN 256
                        3649X *
                        3650X *       PIN256 inputs 256 bytes from the data input port.
                        3651X *       S.DTR must be set before any bytes may be transfered.
                        3652X *       This is one of the more critical routines, and should
                        3653X *       be highly tuned.
                        3654X *
                        3655X *       ENTRY:  DE      = buffer
                        3656X *
                        3657X *       EXIT:   PSW     = 'C' CLEAR if NO error
                        3658X *                         DE = DE advanced
                        3659X *                       'C' SET   if   error
                        3660X *
                        3661X *       USES:   PSW,BC,DE
                        3662X *
                        3663X
066.222 315 020 070     3664X PIN256  CALL    WTR
066.225 330             3665X         RC                      Synchronization ERROR
                        3666X
                        3667X *       Accept 256 bytes
                        3668X
066.226 016 000         3669X         MVI     C,0             Set count to 256
066.230 333 170         3670X PIN1.   IN      D.STA
066.232 346 240         3671X         ANI     S.DTR+S.DON
066.234 372 245 066     3672X         JM      PIN2            H47 has a byte
000.000                 3673X         ERRNZ   S.DTR-2000
                        3674X
066.237 312 230 066     3675X         JZ      PIN1.           done is not set
                        3676X
066.242 303 161 067     3677X         JMP     IER.            Generate Error on pre-mature done
                        3678X
066.245 333 171         3679X PIN2    IN      D.DAT
066.247 022             3680X         STAX    D
066.250 023             3681X         INX     D
066.251 015             3682X         DCR     C
066.252 302 230 066     3683X         JNZ     PIN1.
                        3684X
066.255 247             3685X         ANA     A               Clear 'C'
066.256 311             3686X         RET


                        3688X **      RAS     - Read Auxiliary Status
                        3689X *
                        3690X *       RAS reads the auxiliary status for the unit specified
                        3691X *       in AIO.UNI.
                        3692X *
                        3693X *       ENTRY:  AIO.UNI = Device Unit
                        3694X *
                        3695X *       EXIT:   PSW     = 'C'   if ERROR
                        3696X *                           A = error code
                        3697X *                         'NC'  if NO error
                        3698X *
                        3699X *       USES:   PSW,HL,BC
                        3700X *
```

```
                                3701X
066.257  041 000 000  3702X RAS      LXI     H,0
066.262  042 121 075  3703X          SHLD    SECTOR          Zero initial parameters
000.000               3704X          ERRNZ   SIDE-SECTOR-1
000.000               3705X          ERRNZ   SID.0
                      3706X
066.265  315 327 065  3707X          CALL    COM             Output original command
066.270  002          3708X          DB      DD.RAS
066.271  330          3709X          RC
066.272  315 104 067  3710X          CALL    SUS..           Unit number
066.275  330          3711X          RC
066.276  315 133 066  3712X          CALL    PIN             A  = Aux. Status byte
066.301  330          3713X          RC
                      3714X
066.302  365          3715X          PUSH    PSW
066.303  315 367 067  3716X          CALL    WDN             Wait for DONE
066.306  332 313 066  3717X          JC      RAS1
                      3718X
066.311  361          3719X          POP     PSW
066.312  311          3720X          RET                     NO Error, so return with 'NC' and A
                      3721X
066.313  063          3722X RAS1     INX     SP
066.314  063          3723X          INX     SP              Discard saved A
066.315  311          3724X          RET                     Exit with WDN return values


                      3726X **       RST     -  Reset
                      3727X *
                      3728X *        RST reset the device.
                      3729X *
                      3730X *
                      3731X *        ENTRY:  NONE
                      3732X *
                      3733X *        EXIT:   NONE
                      3734X *
                      3735X *        USES:   PSW
                      3736X *
                      3737X
066.316  305          3738X RST      PUSH    B
066.317  315 324 066  3739X          CALL    RST.
066.322  301          3740X          POP     B
066.323  311          3741X          RET
                      3742X
066.324  076 002      3743X RST.     MVI     A,W.RES
066.326  323 170      3744X          OUT     D.STA
066.330  315 355 065  3745X          CALL    DLY
                      3746X
                      3747X *        Wait for DONE
                      3748X
066.333  001 000 000  3749X          LXI     B,RSTA
066.336  013          3750X RST1     DCX     B
066.337  170          3751X          MOV     A,B
066.340  261          3752X          ORA     C
066.341  312 161 067  3753X          JZ      TEB.            Set error flags
```

```
                                3754X
066,344  333 170                3755X           IN      D.STA
066,346  346 040                3756X           ANI     S.DON
066,350  302 336 066            3757X           JNZ     RST1            Wait some more
                                3758X
066,353  311                    3759X           RET
                                3760X
000,000                         3761X RSTA      EQU     0               Time-Out Counter


                                3763X **        SDE     -  Set Default Error
                                3764X *
                                3765X *         SDE sets the default error to the specified one
                                3766X *
                                3767X *         ENTRY:  (SP)    = default error
                                3768X *
                                3769X *         EXIT:   (SP) advanced to the RETurn address
                                3770X *
                                3771X *         USES:   PSW
                                3772X *
                                3773X
066,354  343                    3774X SDE       XTHL
066,355  176                    3775X           MOV     A,M
066,356  043                    3776X           INX     H
066,357  062 274 067            3777X           STA     DEFERR
066,362  343                    3778X           XTHL
066,363  311                    3779X           RET
                                3780X           ENDIF


                                3782X **        SDP     -  Set-up Device Parameters
                                3783X *
                                3784X *         SDP sets up the device TRACK, SIDE, and SECTOR from the
                                3785X *         sector number.
                                3786X *
                                3787X *         IF      .SMALL, this code assumes that AUXSTAT is initialized.
                                3788X *
                                3789X *         ENTRY:  HL      = sector number
                                3790X *
                                3791X *         EXIT:   TRACK, SIDE, and SECTOR initialized for the
                                3792X *                 transfer
                                3793X *
                                3794X *         USES:   PSW,HL
                                3795X *
                                3796X
066,364                         3797X SDP       EQU     *
066,364  305                    3798X           PUSH    B
066,365  325                    3799X           PUSH    D
                                3800X
066,366  315 374 066            3801X           CALL    SDP.
                                3802X
066,371  321                    3803X           POP     D
```

```
066.372  301            3804X          POP     B
066.373  311            3805X          RET
                        3806X
066.374  104            3807X SDP.     MOV     B,H
066.375  115            3808X          MOV     C,L             BC = sector number
066.376  257            3809X          XRA     A
000.000                 3810X          ERRNZ   SID.0
066.377  062 122 075    3811X          STA     SIDE            Initialize Side Byte
                        3812X
000.001                 3813X          IF      .SMALL
                        3814X          LDA     AUXSTAT         A  = Alternate Status
                        3815X          ELSE
067.002  315 273 062    3816X          CALL    FAS             A  = Alternate Status
067.005  062 117 075    3817X          STA     AUXSTAT
                        3818X          ENDIF
                        3819X
067.010  346 100        3820X          ANI     AS.ODD          Track 0 is the real clue
067.012  076 015        3821X          MVI     A,NSPTS
067.014  312 020 067    3822X          JZ      SDP1            Is Single Density
067.017  207            3823X          ADD     A               A = 2 * A
000.000                 3824X          ERRNZ   NSPTS*2-NSPTD
067.020  062 124 075    3825X SDP1     STA     SPT             Save Sectors per Track
                        3826X
067.023  157            3827X          MOV     L,A
067.024  046 000        3828X          MVI     H,0             HL = Sectors per Track
067.026  072 117 075    3829X          LDA     AUXSTAT
067.031  346 020        3830X          ANI     AS.SIA
067.033  312 037 067    3831X          JZ      SDP2            Only 1 Side
067.036  051            3832X          DAD     H               HL = 2 * HL
067.037                 3833X SDP2     EQU     *
000.001                 3834X          IF      .SMALL
                        3835X          ELSE
067.037  175            3836X          MOV     A,L
067.040  062 123 075    3837X          STA     SPC             Save sectors per cylinder
                        3838X          ENDIF
                        3839X
067.043  353            3840X          XCHG                    DE = Sectors per Cylinder
067.044  315 106 030    3841X          CALL    $DV66           HL = BC/DE = Track Number
                        3842X
067.047  175            3843X          MOV     A,L
067.050  062 126 075    3844X          STA     TRACK           Assume Track is Good (Let H47 flag errors)
067.053  173            3845X          MOV     A,E
000.001                 3846X          IF      .SMALL
                        3847X          ELSE
067.054  062 120 075    3848X          STA     CSN             Save cylinder sector number
                        3849X          ENDIF
067.057  074            3850X          INR     A               Range for sector is [1-NSPTx]
067.060  062 121 075    3851X          STA     SECTOR
                        3852X
067.063  041 124 075    3853X          LXI     H,SPT
067.066  276            3854X          CMP     M
067.067  310            3855X          RZ                      Is on Side 0
067.070  077            3856X          CMC
067.071  320            3857X          RNC                     Is on Side 0
                        3858X
067.072  226            3859X          SUB     M
```

```
067.073  062 121 075  3860X          STA     SECTOR          Compute Real sector number
067.076  076 200      3861X          MVI     A,SID.1
067.100  062 122 075  3862X          STA     SIDE            Use side 1
067.103  311          3863X          RET
000.001               3864X          IF      .SMALL
                      3865X          ELSE


                      3867X **       SUS     -  Side Unit Sector
                      3868X *
                      3869X *        SUS outputs the Side/Unit/Sector byte.  It assumes that
                      3870X *        SIDE, AIO,UNI, and SECTOR are already initialized.
                      3871X *
                      3872X *
                      3873X *        NOTE:   This code no longer masks the fields
                      3874X *                to insure against overflow...Be careful!!!
                      3875X *
                      3876X *
                      3877X *        ENTRY:  SIDE    =  side
                      3878X *                AIO.UNI =  unit number
                      3879X *                SECTOR  =  sector number
                      3880X *
                      3881X *        EXIT:   NONE
                      3882X *
                      3883X *        USES    PSW
                      3884X *
                      3885X
067.104  072 121 075  3886X SUS.     LDA     SECTOR
067.107  303 136 067  3887X          JMP     SUS1            Do not map sector number
                      3888X
067.112  305          3889X SUS      PUSH    B
067.113  315 120 067  3890X          CALL    SUS.
067.116  301          3891X          POP     B
067.117  311          3892X          RET
                      3893X
067.120  072 117 075  3894X SUS.     LDA     AUXSTAT
067.123  346 100      3895X          ANI     A8.ODD
067.125  072 121 075  3896X          LDA     SECTOR          A = Sector
067.130  302 136 067  3897X          JNZ     SUS1            Double Density
067.133  075          3898X          DCR     A
067.134  207          3899X          ADD     A               Map Sector Number
067.135  074          3900X          INR     A
067.136  107          3901X SUS1     MOV     B,A
                      3902X
067.137  072 122 075  3903X          LDA     SIDE
067.142  260          3904X          ORA     B
067.143  107          3905X          MOV     B,A             Accumulate Side
                      3906X
067.144  072 061 041  3907X          LDA     AIO.UNI
067.147  017          3908X          RRC
067.150  017          3909X          RRC
067.151  017          3910X          RRC
000.000               3911X          ERRNZ   UNT.M-96
067.152  260          3912X          ORA     B
```

```
                                3913X
067.153  303 022 066           3914X          JMP     OUT             OUTPUT THE BYTE


                                3916X **       TEB     -  Test Error Bit
                                3917X *
                                3918X *        TEB test for the error bit to be set.  This routine assumes
                                3919X *        that the error bit will already be valid, that is, that the
                                3920X *        caller has already verified *S.DON*.
                                3921X *
                                3922X *        If the error bit is set, a table look-up is performed to find
                                3923X *        the HDOS error.
                                3924X *
                                3925X *
                                3926X *        ENTRY:  NONE
                                3927X *
                                3928X *        EXIT:   PSW     = 'C' clear if no error
                                3929X *                          'C' set   if    error
                                3930X *
                                3931X *        USES:   PSW
                                3932X *
                                3933X
067.156  315 316 066           3934X TEB..    CALL    RST             The system needs cleaning up
                                3935X
067.161  072 274 067           3936X TEB:     LDA     DEFERR
067.164  315 171 067           3937X          CALL    TEB             Check for error other than default
067.167  067                   3938X          STC                     Force at least some error flag
067.170  311                   3939X          RET
                                3940X
067.171  365                   3941X TEB      PUSH    PSW             Look for error in the status port
067.172  333 170               3942X          IN      D.STA
067.174  346 040               3943X          ANI     S.DON
067.176  312 271 067           3944X          JZ      TEB4            DONE is NOT set
                                3945X
067.201  333 170               3946X          IN      D.STA
067.203  346 001               3947X          ANI     S.ERR
067.205  312 271 067           3948X          JZ      TEB4            ERROR is NOT set
                                3949X
                                3950X *        ERROR is set
                                3951X
067.210  361                   3952X          POP     PSW             Discard saved PSW
067.211  076 001               3953X          MVI     A,DD.RST        A  = Command
067.213  315 343 065           3954X          CALL    COM..           Read status (Know Done is already set)
067.216  332 265 067           3955X          JC      TEB3            Things are rapidly disintegrating
                                3956X
                                3957X *        Input the error byte
                                3958X
067.221  305                   3959X          PUSH    B
067.222  001 000 000           3960X          LXI     B,WTRA          Initialize Time-Out Counter
                                3961X
067.225  013                   3962X TEB1     DCX     B
067.226  170                   3963X          MOV     A,B
067.227  261                   3964X          ORA     C
067.230  312 264 067           3965X          JZ      TEB2            Time-Out
```

```
                                3966X
067.233  333 170                3967X           IN      D.STA
067.235  346 040                3968X           ANI     S.DON
067.237  302 264 067            3969X           JNZ     TEB2            Pre-Mature Done
                                3970X
067.242  333 170                3971X           IN      D.STA
067.244  346 200                3972X           ANI     S.DTR
067.246  312 225 067            3973X           JZ      TEB1            No DONE yet
                                3974X
067.251  301                    3975X           POP     B
067.252  315 137 066            3976X           CALL    PIN.            Get the error byte
                                3977X
                                3978X *         Determine HDOS error
                                3979X
067.255  346 100                3980X           ANI     SB.WPD
067.257  076 025                3981X           MVI     A,EC.WP
067.261  067                    3982X           STC
067.262  300                    3983X           RNZ                     Drive was write-protected
                                3984X
067.263  305                    3985X           PUSH    B
067.264  301                    3986X TEB2      POP     B
                                3987X
                                3988X *         Take the default error
                                3989X
067.265  072 274 067            3990X TEB3      LDA     DEFERR          A  = default error
067.270  311                    3991X           RET
                                3992X
                                3993X *         NO Error
                                3994X
067.271  361                    3995X TEB4      POP     PSW             Restore A
067.272  247                    3996X           ANA     A               Clear Error Flag
067.273  311                    3997X           RET
                                3998X
067.274  000                    3999X DEFERR    DB      0               Default Error for anything but write-protect


                                4001X **        TRK     -  Track
                                4002X *
                                4003X *         TRK output the track.
                                4004X *
                                4005X *
                                4006X *         ENTRY:  TRACK   =  track sought
                                4007X *
                                4008X *         EXIT:   NONE
                                4009X *
                                4010X *         USES:   PSW
                                4011X *
                                4012X
067.275  072 126 075            4013X TRK       LDA     TRACK
067.300  303 022 066            4014X           JMP     OUT
```

```
                        4016X **      USN     - Update Sector Number
                        4017X *
                        4018X *       USN updates the sector number to the next group.  SDP
                        4019X *       must have been previously called to initialize all of
                        4020X *       the device parameters.
                        4021X *
                        4022X *       ENTRY:  SPC, CSN, and STC initialized.
                        4023X *
                        4024X *       EXIT:   Parameters updated
                        4025X *
                        4026X *       USES:   PSW,HL
                        4027X *
                        4028X
067.303                 4029X USN     EQU     *
                        4030X
                        4031X *       Compute new sector number
                        4032X
067.303  072 120 075  4033X         LDA     CSN
067.306  052 125 075  4034X         LHLD    STC
067.311  205          4035X         ADD     L
067.312  062 120 075  4036X         STA     CSN             Update cylinder sector number
                        4037X
                        4038X *       Check for cylinder wrap
                        4039X
067.315  052 123 075  4040X         LHLD    SPC
067.320  225          4041X         SUB     L
067.321  332 333 067  4042X         JC      USN1            Is no cylinder wrap
067.324  062 120 075  4043X         STA     CSN             Update cylinder number
067.327  041 126 075  4044X         LXI     H,TRACK
067.332  064          4045X         INR     M               Move to the next track
067.333                 4046X USN1    EQU     *
                        4047X
                        4048X *       Compute physical sector number
                        4049X
067.333  257          4050X         XRA     A
000.000                 4051X         ERRNZ   SID.0
067.334  062 122 075  4052X         STA     SIDE            Assume side 0
067.337  072 120 075  4053X         LDA     CSN
067.342  074          4054X         INR     A               Range for sector number is [1-NSPT+1]
067.343  062 121 075  4055X         STA     SECTOR          Assume on side 0
067.346  041 124 075  4056X         LXI     H,SPT
067.351  276          4057X         CMP     M
067.352  310          4058X         RZ                      Is side 0
067.353  077          4059X         CMC
067.354  320          4060X         RNC                     Is side 0
                        4061X
067.355  226          4062X         SUB     M
067.356  062 121 075  4063X         STA     SECTOR          Is Side 1
067.361  076 200      4064X         MVI     A,SID.1
067.363  062 122 075  4065X         STA     SIDE
067.366  311          4066X         RET
                        4067X         ENDIF
```

```
                                4069X **      WDN     -  Wait for Done
                                4070X *
                                4071X *       WDN waits for the done bit to be asserted.
                                4072X *
                                4073X *
                                4074X *       ENTRY:  NONE
                                4075X *
                                4076X *       EXIT:   PSW     =  'C' clear if NO errors
                                4077X *                       =  'C' set   if    error bit set
                                4078X *                          A  = Error Code
                                4079X *
                                4080X *       USES:   PSW
                                4081X *
                                4082X
067.367  305                   4083X WDN     PUSH    B
067.370  315 375 067           4084X         CALL    WDN.
067.373  301                   4085X         POP     B
067.374  311                   4086X         RET
                                4087X
067.375  001 000 000           4088X WDN.    LXI     B,WDNA          Initialize Time-Out counter
                                4089X
070.000  013                   4090X WDN1    DCX     B
070.001  170                   4091X         MOV     A,B
070.002  261                   4092X         ORA     C
070.003  312 156 067           4093X         JZ      TEB..           Time-out ERROR
                                4094X
070.006  333 170               4095X         IN      D.STA
070.010  346 040               4096X         ANI     S.DON
070.012  312 000 070           4097X         JZ      WDN1            Wait for Done
                                4098X
070.015  303 171 067           4099X         JMP     TEB             Test error bits
                                4100X
000.000                         4101X WDNA    EQU     0               Time-Out Counter
000.001                         4102X         IF      1
                                4103X WND     SPACE   4,10
                                4104X **      WND     -  Wait for Not Done
                                4105X *
                                4106X *       WND waits for the Done bit to be cleared
                                4107X *
                                4108X *
                                4109X *       ENTRY:  NONE
                                4110X *
                                4111X *       EXIT:   NONE
                                4112X *
                                4113X *       USES:   PSW
                                4114X *
                                4115X
                                4116X WND     PUSH    B
                                4117X         CALL    WND.
                                4118X         POP     B
                                4119X         RET
                                4120X
                                4121X WND.    LXI     B,WNDA
                                4122X
                                4123X WND1    DCX     B
                                4124X         MOV     A,B
```

```
                        4125X          ORA     C
                        4126X          JZ      TEB..           Time-Out Error
                        4127X
                        4128X          IN      D.STA
                        4129X          ANI     S.DON
                        4130X          JNZ     WND1            DONE IS STILL HIGH
                        4131X
                        4132X          RET
                        4133X
                        4134X WNDA     EQU     0               Wait for Not Done Time-Out count
                        4135X          ENDIF
000.001                 4136X          IF      .SMALL
                        4137X          ELSE


                        4139X **       WTR     - Wait for Transfer Request
                        4140X *
                        4141X *        WTR waits for a transfer request.  It checks for done
                        4142X *        first, and if it is found flags an error.  The code
                        4143X *        will also time-out waiting for  *S.DTR*.
                        4144X *
                        4145X *        ENTRY:  NONE
                        4146X *
                        4147X *        EXIT:   PSW     = 'C' CLEAR if NO error
                        4148X *                          'C' SET   if    error
                        4149X *
                        4150X *        USES:   PSW
                        4151X *
                        4152X
070.020  305            4153X WTR      PUSH    B
070.021  315 026 070    4154X          CALL    WTR.
070.024  301            4155X          POP     B
070.025  311            4156X          RET
                        4157X
070.026  001 000 000    4158X WTR.     LXI     B,WTRA          Initialize Time-Out Counter
                        4159X
070.031  333 170        4160X WTR1     IN      D.STA
070.033  346 040        4161X          ANI     S.DON
070.035  302 161 067    4162X          JNZ     TEB.            Done means some type of synchronization error
                        4163X
                        4164X *        Wait for Handshake on First Byte, Time-Out if not fast enough
                        4165X
070.040  013            4166X          DCX     B
070.041  170            4167X          MOV     A,B
070.042  261            4168X          ORA     C
070.043  312 156 067    4169X          JZ      TEB..           Time-Out fatal
                        4170X
070.046  333 170        4171X          IN      D.STA
070.050  346 200        4172X          ANI     S.DTR
070.052  312 031 070    4173X          JZ      WTR1            Wait for Data Transfer Request
                        4174X
070.055  311            4175X          RET
                        4176X
000.000                 4177X WTRA     EQU     0               Time-Out Loop control
```

```
                        4178X       ENDIF
```

```
070.056                 4181          XTEXT   BITS


                        4183X **      BITS    -  BIT SET
                        4184X *
                        4185X *       BITS SETS THE SPECIFIED BIT IN THE ACCUMULATOR.
                        4186X *
                        4187X *       ENTRY:  A       = ORIGINAL  A
                        4188X *               B       = NUMBER OF BIT TO SET ( 7=HIGH,...,0=LOW )
                        4189X *
                        4190X *       EXIT:   A       = ORIGINAL  A  WITH  BIT(B) SET
                        4191X *
                        4192X *       USES:   PSW
                        4193X *
                        4194X
070.056  305           4195X BITS    PUSH    B
                        4196X
070.057  365           4197X         PUSH    PSW
070.060  076 200       4198X         MVI     A,10000000B
070.062  004           4199X         INR     B
070.063  007           4200X BITS1   RLC
070.064  005           4201X         DCR     B
070.065  302 063 070   4202X         JNZ     BITS1
                        4203X
070.070  117           4204X         MOV     C,A
070.071  361           4205X         POP     PSW
070.072  261           4206X         ORA     C
                        4207X
070.073  301           4208X         POP     BC
070.074  311           4209X         RET
070.075                 4210          XTEXT   CCO


                        4212X **      $CCO - CLEAR CONTROL-O
                        4213X *
                        4214X *       $CCO IS CALLED TO CLEAR THE EFFECT OF THE CTL-O CHARACTER.
                        4215X *
                        4216X *       ENTRY   NONE
                        4217X *       EXIT    NONE
                        4218X *       USES    NONE
                        4219X
                        4220X
070.075  315 054 031   4221X $CCO    CALL    $SAVALL         SAVE REGISTERS
070.100  076 004       4222X         MVI     A,I.CONFL
070.102  001 001 000   4223X         LXI     B,CO.FLG        CLEAR CO.FLG
070.105  377 006       4224X         DB      SYSCALL,.CONSL
070.107  303 047 031   4225X         JMP     $RSTALL         RESTORE REGISTERS AND RETURN
070.112                 4226          XTEXT   CDEHL
```

```
                        4228X **      $CDEHL - COMPARE (DE) TO (HL)
                        4229X *
                        4230X *       $CDEHL COMPARES (DE) TO (HL) FOR EQUALITY.
                        4231X *
                        4232X *       ENTRY   NONE
                        4233X *       EXIT    'Z' SET IF (DE) = (HL)
                        4234X *       USES    A,F
                        4235X
                        4236X
030.216                 4237X $CDEHL  EQU     30216A          IN H17 ROM
070.112                 4238          XTEXT   COMP


                        4240X **      $COMP - COMPARE TWO CHARACTER STRINGS.
                        4241X *
                        4242X *       $COMP COMPARES TWO BYTE STRINGS.
                        4243X *
                        4244X *       ENTRY   (C) = COMPARE COUNT
                        4245X *               (DE) = FWA OF STRING #1
                        4246X *               (HL) = FWA OF STRING #2
                        4247X *       EXIT    'Z' CLEAR, IS MIS-MATCH
                        4248X *               (C) = LENGTH REMAINING
                        4249X *                (DE) = ADDRESS OF MISMATCH IN STRING#1
                        4250X *               (HL) = ADDRESS OF MISMATCH IN STRING #2
                        4251X *               'C' SET, HAVE MATCH
                        4252X *               (C) = 0
                        4253X *                (DE) = (DE) + (OC)
                        4254X *               (HL) = (HL) + (OC)
                        4255X *       USES    A,F,C,D,E,H,L
                        4256X
                        4257X
030.060                 4258X $COMP   EQU     30060A          IN H17 ROM
070.112                 4259          XTEXT   CRLF


                        4261X **      $CRLF - TYPE CARRIAGE RETURN/LINE FEED
                        4262X *
                        4263X *       $CRLF IS USED TO GENERATE PADDED CRLF'S.
                        4264X *
                        4265X *       ENTRY   NONE
                        4266X *       EXIT    (A) = 0
                        4267X *       USES    A,F
                        4268X
                        4269X
070.112  076 012       4270X $CRLF   MVI     A,NL
070.114  377 002       4271X         DB      SYSCALL,,SCOUT
070.116  257           4272X         XRA     A
070.117  311           4273X         RET
070.120                 4274          XTEXT   DADA2
```

```
                        4276X **      $DADA. - ADD (0,A) TO (H,L)
                        4277X *
                        4278X *       ENTRY   NONE
                        4279X *       EXIT    (HL) = (HL) + (0A)
                        4280X *       USES    A,F,H,L
                        4281X
                        4282X
030.101                 4283X $DADA.  EQU     30101A          IN H17 ROM
070.120                 4284          XTEXT   MOVE



                        4286X **      $MOVE - MOVE DATA
                        4287X *
                        4288X *       $MOVE MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
                        4289X *       IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
                        4290X *       FIRST TO LAST.
                        4291X *
                        4292X *       IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
                        4293X *       LAST TO FIRST.
                        4294X *
                        4295X *       THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
                        4296X *
                        4297X *       ENTRY   (BC) = COUNT
                        4298X *               (DE) = FROM
                        4299X *               (HL) = TO
                        4300X *       EXIT    MOVED
                        4301X *               (DE) = ADDRESS OF NEXT FROM BYTE
                        4302X *               (HL) = ADDRESS OF NEXT *TO* BYTE
                        4303X *               'C' CLEAR
                        4304X *       USES    ALL
                        4305X
                        4306X
030.252                 4307X $MOVE   EQU     30252A          IN H17 ROM
070.120                 4308          XTEXT   HLCPDE                                  /072080/
                        4309X **      HLCPDE  - (HL) COMPARED TO (DE)
                        4310X *
                        4311X *       THIS ROUTINE IS DOUBLE WORD COMPARE OF REGISTER PAIRS (DE) AND (HL).
                        4312X *
                        4313X *       ENTRY:  (HL)&(DE) SET UP
                        4314X *
                        4315X *       EXIT:   (PSW)   =
                        4316X *                       'Z' SET   IF (HL) =  (DE)
                        4317X *                       'C' SET   IF (HL) <  (DE)
                        4318X *                       'C' CLEAR IF (HL) >= (DE)
                        4319X *
                        4320X *
                        4321X *       USES:   (PSW)
                        4322X *
                        4323X
070.120  174            4324X HLCPDE  MOV     A,H
070.121  272            4325X         CMP     D       'C' SET => (A) < (D)
070.122  300            4326X         RNZ
070.123  175            4327X         MOV     A,L
070.124  273            4328X         CMP     E       'C' SET => (L) < (E)
```

```
070.125  311          4329X          RET
070.126               4330           XTEXT   MU86                                    /072080/


                      4332X **       $MU86 - MULTIPLY 8X16 UNSIGNED.
                      4333X *
                      4334X *        $MU86 MULTIPLIES A 16 BIT VALUE BY A 8
                      4335X *        BIT VALUE.
                      4336X *
                      4337X *        ENTRY   (A) = MULTIPLIER
                      4338X *                (DE) = MULTIPLICAND
                      4339X *        EXIT    (HL) = RESULT
                      4340X *                'Z' SET IF NOT OVERFLOW
                      4341X *        USES    A,F,H,L
                      4342X
                      4343X
031.007               4344X $MU86    EQU     31007A          IN H17 ROM
070.126               4345           XTEXT   DU66


                      4347X **       $DU66 - UNSIGNED 16 / 16 DIVIDE.
                      4348X *
                      4349X *        (HL) = (BC)/(DE)
                      4350X *
                      4351X *        ENTRY   (BC), (DE) PRESET
                      4352X *        EXIT    (HL) = RESULT
                      4353X *                (DE) = REMAINDER
                      4354X *        USES    ALL
                      4355X
                      4356X
030.106               4357X $DU66    EQU     30106A          IN H17 ROM
070.126               4358           XTEXT   MOVEL


                      4360X **       $MOVEL - MOVE DATA
                      4361X *
                      4362X *        $MOVEL MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
                      4363X *        IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
                      4364X *        FIRST TO LAST.
                      4365X *
                      4366X *        IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
                      4367X *        LAST TO FIRST.
                      4368X *
                      4369X *        THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
                      4370X *
                      4371X *        CALL    $MOVEL
                      4372X *        DW      COUNT
                      4373X *        DW      FROM
                      4374X *        DW      TO
                      4375X *
```

```
                        4376X *         ENTRY   ((SP)) = RET
                        4377X *                 (RET+0) = COUNT (WORD VALUE)
                        4378X *                 (RET+2) = FROM
                        4379X *                 (RET+4) = TO
                        4380X *         EXIT    TO (RET+6)
                        4381X *                 (DE) = ADDRESS OF NEXT FROM BYTE
                        4382X *                 (HL) = ADDRESS OF NEXT *TO* BYTE
                        4383X *                 'C' CLEAR
                        4384X *         USES    ALL
                        4385X
                        4386X
070.126  341           4387X $MOVEL  POP     H               (HL) = RET
070.127  116           4388X         MOV     C,M
070.130  043           4389X         INX     H
070.131  106           4390X         MOV     B,M             (BC) = COUNT
070.132  043           4391X         INX     H
070.133  136           4392X         MOV     E,M
070.134  043           4393X         INX     H
070.135  126           4394X         MOV     D,M             (DE) = FROM
070.136  043           4395X         INX     H
070.137  325           4396X         PUSH    D               ((SP)) = FROM
070.140  136           4397X         MOV     E,M
070.141  043           4398X         INX     H
070.142  126           4399X         MOV     D,M             (DE) = TO
070.143  043           4400X         INX     H
070.144  343           4401X         XTHL                    ((SP)) = RET, (HL) = FROM
070.145  353           4402X         XCHG                    (DE) = FROM , (HL) = TO
070.146  303 252 030   4403X         JMP     $MOVE           MOVE IT
070.151               4404          XTEXT   SAVALL


                        4406X **        $RSTALL - RESTORE ALL REGISTERS.
                        4407X *
                        4408X *         $RSTALL RESTORES ALL THE REGISTERS OFF THE STACK, AND
                        4409X *         RETURNS TO THE PREVIOUS CALLER.
                        4410X *
                        4411X *         ENTRY   (SP) = PSW
                        4412X *                 (SP+2) = BC
                        4413X *                 (SP+4) = DE
                        4414X *                 (SP+6) = HL
                        4415X *                 (SP+8) = RET
                        4416X *         EXIT    TO *RET*, REGISTERS RESTORED
                        4417X *         USES    ALL
                        4418X
                        4419X
031.047                 4420X $RSTALL EQU     3104?A          IN H17 ROM
```

```
                        4422X **      $SAVALL - SAVE ALL REGISTERS ON STACK.
                        4423X *
                        4424X *       $SAVALL SAVES ALL THE REGISTERS ON THE STACK.
                        4425X *
                        4426X *       ENTRY   NONE
                        4427X *       EXIT    (SP) = PSW
                        4428X *               (SP+2) = BC
                        4429X *               (SP+4) = DE
                        4430X *               (SP+6) = HL
                        4431X *       USES    H,L
                        4432X
                        4433X
    031.054             4434X $SAVALL EQU     31054A          IN H17 ROM
    070.151             4435          XTEXT   TJMP


                        4437X **      $TJMP - TABLE JUMP.
                        4438X *
                        4439X *       USAGE
                        4440X *
                        4441X *       CALL      $TJMP             (A) = INDEX
                        4442X *       DW        ADDR1
                        4443X *       .           .
                        4444X *       .           .
                        4445X *       .           .
                        4446X *       DW        ADDRN
                        4447X *
                        4448X *       ENTRY     (A) = INDEX
                        4449X *       EXIT       TO PROCESSOR
                        4450X *                 (A) = INDEX*2
                        4451X *       USES       NONE.
                        4452X
                        4453X
    031.061             4454X $TJMP   EQU     31061A          IN H17 ROM, (A) = INDEX*2
                        4455X
    031.062             4456X $TJMP.  EQU     31062A          IN H17 ROM
    070.151             4457          XTEXT   MLU


                        4459X **      MLU - MAP LOWER CASE LINE TO UPPER CASE.
                        4460X *
                        4461X *       MLU MAPS THE LOWER CASE ALPHABETICS IN A LINE TO UPPER CASE.
                        4462X *
                        4463X *       ENTRY   (HL) = LINE FWA
                        4464X *       EXIT    NONE
                        4465X *       USES    NONE
                        4466X
                        4467X
    070.151  365       4468X $MLU    PUSH    PSW             SAVE (PSW)
    070.152  345       4469X         PUSH    H               SAVE FWA
    070.153  053       4470X         DCX     H               ANTICIPATE INX H
    070.154  043       4471X $MLU1   INX     H
```

```
070.155  176           4472X          MOV     A,M             (A)= CHARACTER
070.156  315 171 070   4473X          CALL    $MCU            MAP CHAR TO UPPER
070.161  167           4474X          MOV     M,A
070.162  247           4475X          ANA     A
070.163  302 154 070   4476X          JNZ     $MLU1           MORE TO GO
070.166  341           4477X          POP     H               RESTORE (HL)
070.167  361           4478X          POP     PSW             RESTORE (PSW)
070.170  311           4479X          RET
070.171                4480           XTEXT   MCU


                       4482X **       MCU - MAP LOWER CASE TO UPPER CASE.
                       4483X *
                       4484X *        MCU MAPS A LOWER CASE ALPHABETIC TO UPPER
                       4485X *        CASE.
                       4486X *
                       4487X *        ENTRY   (A) = CHARACTER
                       4488X *        EXIT    (A) = CHARACTER RESULT
                       4489X *        USES    A,F
                       4490X
                       4491X
070.171  376 141       4492X $MCU     CPI     'a'
070.173  330           4493X          RC                      NOT LOWER CASE
070.174  376 173       4494X          CPI     'z'+1
070.176  320           4495X          RNC                     NOT LOWER CASE
070.177  326 040       4496X          SUI     'a'-'A'
070.201  311           4497X          RET
070.202                4498           XTEXT   RTL


                       4500X **       $RTL - READ TEXT LINE.
                       4501X *
                       4502X *        $RTL READS A LINE FROM THE TERMINAL.
                       4503X *
                       4504X *        CHARACTER ARE ACCEPTED FROM THE TERMINAL, RUBOUT AND BACKSPACE
                       4505X *        CHARACTERS ARE PROCESSED. WHEN A CARRIAGE RETURN IS ENTERED,
                       4506X *        $RTL RETURNS.
                       4507X *
                       4508X *        ENTRY   (HL) = BUFFER FWA
                       4509X *        EXIT    'C' CLEAR IF OK
                       4510X *                 DATA IN BUFFER
                       4511X *                 (A) = TEXT LENGTH
                       4512X *                 'C' SET IF CTL-D STRUCK
                       4513X *        USES    A,F
                       4514X
                       4515X
070.202  315 211 070   4516X $RTL.    CALL    $RTL            $RTL IN UPPER CASE
070.205  330           4517X          RC                      CTL-D
070.206  303 151 070   4518X          JMP     $MLU            MAP LINE TO UPPER CASE
                       4519X
070.211                4520X $RTL     EQU     *
070.211  345           4521X          PUSH    H               SAVE FWA
```

```
070.212  315 300 070  4522X $RTL1   CALL    $RCHAR
070.215  376 004      4523X         CPI     CTLD
070.217  312 244 070  4524X         JE      $RTL2          CTL-D STRUCK
070.222  167          4525X         MOV     M,A
070.223  043          4526X         INX     H
070.224  376 012      4527X         CPI     NL
070.226  302 212 070  4528X         JNE     $RTL1
070.231  053          4529X         DCX     H
070.232  066 000      4530X         MVI     M,0
070.234  043          4531X         INX     H
                      4532X
                      4533X *       ALL DONE, COMPUTE LENGTH
                      4534X
070.235  353          4535X         XCHG                   (DE) = LWA+1
070.236  343          4536X         XTHL                   (HL) = FWA
070.237  173          4537X         MOV     A,E
070.240  225          4538X         SUB     L              (A) = LENGTH
070.241  247          4539X         ANA     A              CLEAR CARRY
070.242  321          4540X         POP     D              RESTORE (DE)
070.243  311          4541X         RET
                      4542X
                      4543X *       CTL-D STRUCK
                      4544X
070.244  341          4545X $RTL2   POP     H              (HL) = FWA
070.245  067          4546X         STC
070.246  311          4547X         RET
070.247               4548          XTEXT   TBLS


                      4550X **      $TBLS - TABLE SEARCH
                      4551X *
                      4552X *       TABLE FORMAT
                      4553X *
                      4554X *       DB      KEY1,VAL1,
                      4555X *       .       .
                      4556X *       .       .
                      4557X *       DB      KEYN,VALN
                      4558X *       DB      0
                      4559X *
                      4560X *       ENTRY   (A) = PATTERN
                      4561X *               (H,L) = TABLE FWA
                      4562X *       EXIT    (A) = PATTERN IF FOUND
                      4563X *               'Z' SET IF FOUND
                      4564X *               'Z' CLEAR IF NOT FOUND OR PATTERN=0          /78.10.GC/
                      4565X *       USES    A,F,H,L
                      4566X
                      4567X
070.247  305          4568X $TBLS   PUSH    B
070.250  376 000      4569X         CPI     0                                            /78.10.GC/
070.252  312 274 070  4570X         JZ      TBL2                                         /78.10.GC/
070.255  107          4571X         MOV     B,A
070.256  176          4572X TBL1    MOV     A,M            (A) = CHARACTER
070.257  043          4573X         INX     H
070.260  270          4574X         CMP     B
```

```
070.261  312 276 070  4575X          JZ      TBL3            IF MATCH
070.264  247          4576X          ANA     A
070.265  043          4577X          INX     H               SKIP PAST
070.266  302 256 070  4578X          JNZ     TBL1            IF NOT END OF TABLE
070.271  053          4579X          DCX     H
070.272  053          4580X          DCX     H
070.273  257          4581X          XRA     A               SET TO ZERO FOR OLD USERS       /78.10.GC/
070.274  376 001      4582X TBL2     CPI     1               CLEAR ZERO                      /78.10.GC/
                      4583X
                      4584X *        DONE
                      4585X
070.276  301          4586X TBL3     POP     B
070.277  311          4587X          RET
070.300               4588           XTEXT   RCHAR


                      4590X **       $RCHAR - READ SINGLE CHARACTER FROM CONSOLE.
                      4591X *
                      4592X *        ENTRY   NONE
                      4593X *        EXIT    (A) = CHARACTER
                      4594X *        USES    A,F
                      4595X
                      4596X
070.300  377 001      4597X $RCHAR   DB      SYSCALL,.SCIN
070.302  332 300 070  4598X          JC      $RCHAR          NOT READY
070.305  311          4599X          RET
                      4600X
070.306  377 002      4601X $WCHAR   DB      SYSCALL,.SCOUT
070.310  311          4602X          RET
070.311               4603           XTEXT   DADA


                      4605X **       $DADA - PERFORM (H,L) = (H,L) + (0,A)
                      4606X *
                      4607X *        ENTRY   (H,L) = BEFORE VALUE
                      4608X *                (A) = BEFORE VALUE
                      4609X *        EXIT    (H,L) = (H,L) + (0,A)
                      4610X *                'C' SET IF OVERFLOW
                      4611X *        USES    F,H,L
                      4612X
                      4613X
030.072               4614X $DADA    EQU     30072A          IN H17 ROM
070.311               4615           XTEXT   UDUN
```

```
                                4617X **      $UDDN - UNPACK DECIMAL DIGITS.
                                4618X *
                                4619X *       UDDN CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF
                                4620X *       DECIMAL DIGITS. THE RESULT IS NULL FILLED TO THE LEFT.
                                4621X *
                                4622X *       ENTRY   (B,C) = ADDRESS VALUE
                                4623X *               (A) = DIGIT COUNT
                                4624X *               (H,L) = MEMORY ADDRESS
                                4625X *       EXIT    (HL) = (HL) + (A)
                                4626X *       USES    ALL
                                4627X
                                4628X
070.311                         4629X $UDDN   EQU     *
070.311 315 072 030             4630X         CALL    $DADA
070.314 345                     4631X         PUSH    H               SAVE FINAL (H,L) VALUE
                                4632X
070.315 365                     4633X UDDN1   PUSH    PSW
070.316 345                     4634X         PUSH    H
070.317 021 012 000             4635X         LXI     D,10
070.322 315 106 030             4636X         CALL    $DV66           (H,L) = VALUE/10
070.325 104                     4637X         MOV     B,H
070.326 115                     4638X         MOV     C,L             (BC) = QUOTIENT
070.327 341                     4639X         POP     H
070.330 076 060                 4640X         MVI     A,'0'
070.332 203                     4641X         ADD     E               ADD REMAINDER
070.333 053                     4642X         DCX     H
070.334 167                     4643X         MOV     M,A             STORE DIGIT
070.335 170                     4644X         MOV     A,B
070.336 261                     4645X         ORA     C
070.337 312 351 070             4646X         JZ      UDDN2           ALL ZEROS
070.342 361                     4647X         POP     PSW
070.343 075                     4648X         DCR     A
070.344 302 315 070             4649X         JNZ     UDDN1           IF MORE TO GO
                                4650X
                                4651X *       ALL DONE. EXIT
                                4652X
070.347 341                     4653X UDDN1.5 POP     H               RESTORE H
070.350 311                     4654X         RET                     RETURN
                                4655X
                                4656X *       DIGITS LEADING THIS ONE ARE ZERO. STORE NULLS INSTEAD.
                                4657X
070.351 361                     4658X UDDN2   POP     PSW
070.352 075                     4659X UDDN3   DCR     A
070.353 312 347 070             4660X         JE      UDDN1.5         ALL DONE
070.356 053                     4661X         DCX     H
070.357 066 000                 4662X         MVI     M,0
070.361 303 352 070             4663X         JMP     UDDN3
070.364                         4664          XTEXT   RND
```

```
                                4666X **      $RND - COMPUTE TAUSWORTH 15 BIT RANDOM NUMBER
                                4667X *
                                4668X *       $RND COMPUTES A RANDOM NUMBER USING RSEED
                                4669X *       AS THE SEED.
                                4670X *
                                4671X *       ENTRY   (RSEED) = NON-ZERO SEED(16 BIT)
                                4672X *       EXIT    (HL) = RANDOM NUMBER
                                4673X *       USES    A,F,H,L
                                4674X
                                4675X
     070.364  052 113 075       4676X $RND    LHLD    RSEED           (HL) = SEED
     070.367  325               4677X         PUSH    D               SAVE (DE)
     070.370  026 017           4678X         MVI     D,15            (D) = BIT COUNT
                                4679X
     070.372  174               4680X RND1    MOV     A,H             SHIFT RIGHT ONE
     070.373  247               4681X         ANA     A
     070.374  037               4682X         RAR
     070.375  147               4683X         MOV     H,A
     070.376  175               4684X         MOV     A,L
     070.377  037               4685X         RAR
     071.000  157               4686X         MOV     L,A
     071.001  027               4687X         RAL                     'C' = 1
     071.002  027               4688X         RAL
     071.003  027               4689X         RAL
     071.004  027               4690X         RAL                     'C' = 10Q
     071.005  255               4691X         XRA     L               XOR WITH VALUE
     071.006  027               4692X         RAL
     071.007  027               4693X         RAL
     071.010  027               4694X         RAL
     071.011  346 100           4695X         ANI     100Q
     071.013  264               4696X         ORA     H               INSERT IN LEFT
     071.014  147               4697X         MOV     H,A
     071.015  025               4698X         DCR     D
     071.016  302 372 070       4699X         JNZ     RND1            MORE TO GO
     071.021  042 113 075       4700X         SHLD    RSEED           SAVE SEED
     071.024  321               4701X         POP     D               RESTORE (DE)
                                4702X
     071.025  311               4703X         RET                     EXIT
     071.026                    4704          XTEXT   TYPT2


                                4706X **      $TYPTX - TYPE TEXT.
                                4707X *
                                4708X *       $TYPTX IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE.
                                4709X *
                                4710X *       IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED,
                                4711X *       A BYTE WITH THE 200Q BIT SET IS THE LAST BYTE IN THE MESSAGE.
                                4712X *
                                4713X *       ENTRY   (RET) = TEXT
                                4714X *       EXIT    TO (RET+LENGTH)
                                4715X *       USES    A,F
                                4716X
                                4717X
     031.136                    4718X $TYPTX  EQU     31136A          IN H17 ROM
```

```
                        4719X
031.144                 4720X $TYPTX. EQU   31144A          IN H17 ROM
071.026                 4721          XTEXT INDL                                /071080/


                        4723X **      $INDL - INDEXED LOAD.
                        4724X *
                        4725X *       $INDL LOADS DE WITH THE TWO BYTES AT (HL)+DISPLACMENT
                        4726X *
                        4727X *       THIS ACTS AS AN INDEXED FULL WORD LOAD.
                        4728X *
                        4729X *       (DE) = ( (HL) + DSPLACEMENT )
                        4730X *
                        4731X *       ENTRY   ((RET)) = DISPLACMENT (FULL WORD)
                        4732X *                 (HL) = TABLE ADDRESS
                        4733X *       EXIT    TO (RET+2)
                        4734X *       USES    A,F,D,E
                        4735X
                        4736X
030.234                 4737X $INDL   EQU   30234A          IN H17 ROM
071.026                 4738          XTEXT INDXX


                        4740X **      $INDLB  - INDEXED LOAD BYTE
                        4741X *
                        4742X *       BYTE INDEXED LOAD PRIMITIVE
                        4743X *
                        4744X *       ENTRY:  HL      = BASE ADDRESS
                        4745X *               (RET)   = FULL WORD RELOCATION
                        4746X *
                        4747X *       EXIT:   A       = ( HL + (RET) )
                        4748X *
                        4749X *       USES:   A
                        4750X *
                        4751X
071.026  353           4752X $INDLB  XCHG                DE = BASE
071.027  343           4753X         XTHL                SAVE  .DE.
071.030  325           4754X         PUSH  D             SAVE  BASE
071.031  305           4755X         PUSH  B             SAVE  .BC.
                        4756X
071.032  116           4757X         MOV   C,M
071.033  043           4758X         INX   H
071.034  106           4759X         MOV   B,M           BC = OFFSET
071.035  043           4760X         INX   H             HL = .RET.
                        4761X
071.036  353           4762X         XCHG                HL = BASE
071.037  011           4763X         DAD   B             HL = BASE + OFFSET
071.040  176           4764X         MOV   A,M           A = ( BASE + OFFSET )
071.041  353           4765X         XCHG                HL = .RET.
                        4766X
071.042  301           4767X         POP   B             RESTORE  .BC.
071.043  321           4768X         POP   D             RESTORE  BASE
```

```
071.044  343          4769X          XTHL                    HL = .DE. ; (SP) = .RET.
071.045  353          4770X          XCHG                    DE = .DE. ; HL = BASE
071.046  311          4771X          RET


                      4773X **       $INDS   -  INDEXED STORE
                      4774X *
                      4775X *        INDEXED STORE PRIMITIVE.
                      4776X *
                      4777X *        ENTRY:  HL = BASE ADDRESS
                      4778X *                DE = VALUE TO STORE
                      4779X *
                      4780X *        EXIT:   (HL + (RET)) = DE
                      4781X *
                      4782X *        USES:   NONE
                      4783X *
                      4784X
071.047  315 136 071  4785X $INDS    CALL    XCHGBC
071.052  343          4786X          XTHL                    SAVE  .BC.
071.053  325          4787X          PUSH    D
071.054  315 124 071  4788X          CALL    ILDEHL          DE = OFFSET
071.057  315 136 071  4789X          CALL    XCHGBC          BC = .RET.
071.062  353          4790X          XCHG                    DE = BASE ; HL = OFFSET
071.063  031          4791X          DAD     D               HL = BASE + OFFSET
071.064  353          4792X          XCHG
071.065  343          4793X          XTHL                    SAVE  BASE
071.066  353          4794X          XCHG                    DE = VALUE
071.067  315 131 071  4795X          CALL    ISDEHL
071.072  341          4796X          POP     H               HL = BASE
071.073  315 136 071  4797X          CALL    XCHGBC
071.076  343          4798X          XTHL                    RESTORE  .BC.
071.077  315 136 071  4799X          CALL    XCHGBC
071.102  311          4800X          RET


                      4802X **       $INDSB  -  INDEXED BYTE STORE
                      4803X *
                      4804X *        INDEXED BYTE STORE.
                      4805X *
                      4806X *        ENTRY:  A       = VALUE TO STORE
                      4807X *                HL      = BASE ADDRESS
                      4808X *                (RET)   = OFFSET
                      4809X *
                      4810X *        EXIT:   NONE
                      4811X *
                      4812X *        USES:   PSW
                      4813X *
                      4814X
071.103  353          4815X $INDSB   XCHG                    DE = BASE
071.104  343          4816X          XTHL                    SAVE  .DE.
071.105  325          4817X          PUSH    D               SAVE  BASE
071.106  305          4818X          PUSH    B               SAVE  .BC.
```

```
                        4819X
071.107 116             4820X           MOV     C,M
071.110 043             4821X           INX     H
071.111 106             4822X           MOV     B,M             BC = OFFSET
071.112 043             4823X           INX     H               HL = .RET.
                        4824X
071.113 353             4825X           XCHG                    HL = BASE
071.114 011             4826X           DAD     B               HL = BASE + OFFSET
071.115 167             4827X           MOV     M,A             ( BASE + OFFSET ) = A
071.116 353             4828X           XCHG
                        4829X
071.117 301             4830X           POP     B               RESTORE .BC.
071.120 321             4831X           POP     D               RESTORE  BASE
071.121 343             4832X           XTHL                    HL = .DE. ; (SP) = .RET.
071.122 353             4833X           XCHG                    DE = .DE. ; HL = BASE
071.123 311             4834X           RET
071.124                 4835            XTEXT   ILDEHL


                        4837X **        ILDEHL  -  INDEXED LOAD OF DE FROM HL
                        4838X *
                        4839X *         'DE' GET THE FULL WORD VALUE POINTED TO BY 'HL', AND 'HL' IS
                        4840X *         INCREMENTED BY TWO.
                        4841X *
                        4842X *         ENTRY:  HL      = ADDRESS OF FULL WORD VALUE
                        4843X *
                        4844X *         EXIT:   DE      = (HL)
                        4845X *                 HL      = HL + 2
                        4846X *
                        4847X *         USES:   DE
                        4848X *
                        4849X
071.124 136             4850X ILDEHL    MOV     E,M
071.125 043             4851X           INX     H
071.126 126             4852X           MOV     D,M
071.127 043             4853X           INX     H
071.130 311             4854X           RET
071.131                 4855            XTEXT   ISDEHL


                        4857X **        ISDEHL  -  INDEXED STORE OF DE AT HL
                        4858X *
                        4859X *         STORE 'DE' AT THE ADDRESS POINTED TO BY 'HL', AND INCREMENT 'HL'
                        4860X *         BY 2.
                        4861X *
                        4862X *         ENTRY:  DE      = VALUE
                        4863X *                 HL      = ADDRESS OF VALUE
                        4864X *
                        4865X *         EXIT:   (HL)    = DE
                        4866X *                 HL      = HL + 2
                        4867X *
                        4868X *         USES:   HL
```

```
                        4869X *
                        4870X
071.131  163           4871X ISDEHL  MOV     M,E
071.132  043           4872X         INX     H
071.133  162           4873X         MOV     M,D
071.134  043           4874X         INX     H
071.135  311           4875X         RET
071.136                4876          XTEXT   XCHGBC



                        4878X **      XCHGBC  -  XCHG BC
                        4879X *
                        4880X *       EXCHANGE THE 'BC' REGISTER PAIR WITH THE 'HL' REGISTER PAIR.
                        4881X *
                        4882X *       ENTRY:  BC      = ORIGINAL BC
                        4883X *               HL      = ORIGINAL HL
                        4884X *
                        4885X *       EXIT:   BC      = ORIGINAL HL
                        4886X *               HL      = ORIGINAL BC
                        4887X *
                        4888X *       USES:   BC,HL
                        4889X *
                        4890X
071.136  365           4891X XCHGBC  PUSH    PSW
071.137  170           4892X         MOV     A,B
071.140  104           4893X         MOV     B,H
071.141  147           4894X         MOV     H,A
071.142  171           4895X         MOV     A,C
071.143  115           4896X         MOV     C,L
071.144  157           4897X         MOV     L,A
071.145  361           4898X         POP     PSW
071.146  311           4899X         RET
071.147                4900          XTEXT   FOPE



                        4902X **      $FOPEx - OPEN FILE BLOCK FOR I/O
                        4903X *
                        4904X *       $FOPEx IS CALLED BEFORE ANY I/O IS DONE VIA A
                        4905X *       FILE BLOCK. $FOPEx SETS UP THE FILE BLOCK, AND OPENS
                        4906X *       THE FILE VIA *HDOS*.
                        4907X *
                        4908X *       ENTRY   (DE) = ADDRESS OF DEFAULT BLOCK
                        4909X *               (HL) = ADDRESS OF FILE BLOCK
                        4910X *       EXIT    TO $FERROR IF ERROR
                        4911X *               TO CALLER IF OK
                        4912X *       USES    A,F,B,C,D,E
                        4913X
                        4914X
071.147  315 174 071   4915X $FOPER  CALL    $FOPER.
071.152  320           4916X         RNC
071.153  303 057 073   4917X         JMP     $FERROR         IN ERROR
                        4918X
```

```
071.156  315 177 071  4919X $FOPEW  CALL   $FOPEW.
071.161  320          4920X         RNC
071.162  303 057 073  4921X         JMP    $FERROR          IN ERROR
                      4922X
071.165  315 202 071  4923X $FOPEU  CALL   $FOPEU.
071.170  320          4924X         RNC
071.171  303 057 073  4925X         JMP    $FERROR          IN ERROR
                      4926X
                      4927X
071.174  076 002      4928X $FOPER.. MVI    A,FT.OR          FILE TYPE OF OPEN FOR READ
071.176  001          4929X         DB     0010             LXI,B TO SKIP NEXT MVI
071.177  076 004      4930X $FOPEW.. MVI    A,FT.OW          OPEN FOR WRITE
071.201  001          4931X         DB     0010             LXI,B TO SKIP NEXT MIV
071.202  076 006      4932X $FOPEU.. MVI    A,FT.OR+FT.OW
                      4933X
                      4934X *       (A) = FILE FLAGS
                      4935X
071.204  345          4936X         PUSH   H                SAVE FILE BLOCK ADDRESS
071.205  365          4937X         PUSH   PSW              SAVE NEW FLAGS
000.000               4938X         ERRNZ  FB.CHA
071.206  106          4939X         MOV    B,M              (B) = CHANNEL NUMBER
071.207  305          4940X         PUSH   B                SAVE HANNEL NUMBER
000.000               4941X         ERRNZ  FB.FLG-FB.CHA-1
071.210  043          4942X         INX    H
071.211  117          4943X         MOV    C,A              (C) = NEW FILE FLAGS
071.212  176          4944X         MOV    A,M              (A) = CURRENT TYPE
071.213  247          4945X         ANA    A
071.214  171          4946X         MOV    A,C              (A) = NEW FLAGS TO BE SET
071.215  312 227 071  4947X         JZ     $FOPE1           NOT ALREADY OPEN
                      4948X
                      4949X *       ALREADY OPEN, SQUACK
                      4950X
071.220  301          4951X         POP    B                RESTORE (BC)
071.221  361          4952X         POP    PSW              DISCARD NEW FLAGS
071.222  341          4953X         POP    H                (HL) = FB ADDRESS
071.223  076 031      4954X         MVI    A,EC.FAO         FILE ALREADY OPEN
071.225  067          4955X         STC
071.226  311          4956X         RET
                      4957X
000.000               4958X         ERRNZ  FB.FWA-FB.FLG-1
071.227  043          4959X $FOPE1  INX    H                (HL) = $FB.FWA
071.230  116          4960X         MOV    C,M
071.231  043          4961X         INX    H
071.232  106          4962X         MOV    B,M              (BC) = FB.FWA
071.233  043          4963X         INX    H
000.000               4964X         ERRNZ  FB.PTR-FB.FWA-2
071.234  161          4965X         MOV    M,C              SET FB.PTR = FB.FWA
071.235  043          4966X         INX    H
071.236  160          4967X         MOV    M,B
071.237  043          4968X         INX    H
000.000               4969X         ERRNZ  FB.LIM-FB.PTR-2
071.240  161          4970X         MOV    M,C              SET FB.LIM = FB.FWA
071.241  043          4971X         INX    H
071.242  160          4972X         MOV    M,B
071.243  043          4973X         INX    H
000.000               4974X         ERRNZ  FB.NAM-FB.LIM-4
```

```
071.244  043          4975X          INX     H
071.245  043          4976X          INX     H               (HL) = $FB.NAM
                      4977X
                      4978X *        FILE BLOCK POINTERS SETUP, OPEN FILE
                      4979X
071.246  345          4980X          PUSH    H               SAVE NEW ADDRESS FOR NAME
071.247  041 300 071  4981X          LXI     H,$FOPEB
071.252  247          4982X          ANA     A                                       /78.10.GC/
071.253  312 262 071  4983X          JZ      $FOPE2
000.000               4984X          ERRNZ   .EXIT
071.256  315 247 070  4985X          CALL    $TBLS           FIND CODE
071.261  176          4986X          MOV     A,M
071.262  062 270 071  4987X $FOPE2   STA     $FOPEA          SET SYSCALL CODE
071.265  341          4988X          POP     H               (HL) = $FB.NAM
071.266  361          4989X          POP     PSW             (A) = CHANNEL NUMBER
071.267  377 000       4990X          DB      SYSCALL,.EXIT
071.270               4991X $FOPEA   EQU     *-1             SYSCALL CODE
071.271  321          4992X          POP     D               (D) = NEW FLAG
071.272  341          4993X          POP     H               (HL) = FILE BLOCK ADDRESS
071.273  330          4994X          RC                      EXIT IF ERROR
071.274  043          4995X          INX     H
000.000               4996X          ERRNZ   FB.FLG-1
071.275  162          4997X          MOV     M,D             SET NEW FLAGS
071.276  053          4998X          DCX     H               RESTORE (HL)
071.277  311          4999X          RET
                      5000X
071.300  002 042      5001X $FOPEB   DB      FT.OR,.OPENR    TABLE OF SYSCALL CODES
071.302  004 043      5002X          DB      FT.OW,.OPENW
071.304  006 044      5003X          DB      FT.OR+FT.OW,.OPENU
071.306  000          5004X          DB      0               SHOULD NOT OCCUR
071.307               5005           XTEXT   FCLO


                      5007X **       $FCLO - CLOSE FILE BLOCK.
                      5008X *
                      5009X *        $FCLO IS CALLED TO TERMINATE PROCESSING THROUGH A FILE
                      5010X *        BLOCK.
                      5011X *
                      5012X *        ENTRY   (HL) = FILE BLOCK ADDRESS
                      5013X *        EXIT    TO $FERROR IF ERROR
                      5014X *                 TO CALLER IF OK
                      5015X *        USES    A,F,B,C,D,E
                      5016X
                      5017X
071.307  315 316 071  5018X $FCLO    CALL    $FCLO.
071.312  320          5019X          RNC                     NO ERROR
071.313  303 057 073  5020X          JMP     $FERROR
                      5021X
071.316  345          5022X $FCLO.   PUSH    H               SAVE FILE BLOCK ADDRESS
000.000               5023X          ERRNZ   FB.FLG-1
071.317  043          5024X          INX     H               (HL) = $FB.FLG
071.320  176          5025X          MOV     A,M
071.321  066 000      5026X          MVI     M,0             CLEAR FLAG
071.323  247          5027X          ANA     A
```

```
071.324  312 012 072  5028X          JZ      $FCLO4          FILE NOT OPEN
071.327  346 004      5029X          ANI     FT.OW
071.331  312 004 072  5030X          JZ      $FCLO3          NO WRITING, NO FLUSHING NEEDED
                      5031X
                      5032X *        WAS OPEN FOR WRITE. SEE IF NEED FLUSH THE LAST SECTOR
                      5033X
071.334  315 234 030  5034X          CALL    $INDL
071.337  003 000      5035X          DW      FB.PTR-FB.FLG
071.341  325          5036X          PUSH    D               SAVE (FB.PTR)
071.342  315 234 030  5037X          CALL    $INDL           (DE) = (FB.FWA)
071.345  001 000      5038X          DW      FB.FWA-FB.FLG
071.347  341          5039X          POP     H               (HL) = (FB.PTR)
071.350  175          5040X          MOV     A,L
071.351  223          5041X          SUB     E
071.352  117          5042X          MOV     C,A
071.353  174          5043X          MOV     A,H
071.354  232          5044X          SBB     D
071.355  107          5045X          MOV     B,A             (BC) = AMOUNT IN BLOCK
071.356  261          5046X          ORA     C
071.357  312 004 072  5047X          JZ      $FCLO3          NONE TO FLUSH
                      5048X
                      5049X *        NEED TO FLUSH BUFFER
                      5050X *
                      5051X *        (BC) = DATA AMOUNT
                      5052X *        (DE) = FWA
                      5053X *        (HL) = LWA+1
                      5054X
071.362  171          5055X          MOV     A,C
071.363  247          5056X          ANA     A
071.364  312 377 071  5057X          JZ      $FCLO2          DONT HAVE PARTIAL SECTOR
                      5058X
                      5059X *        ZERO FILL PARTIAL SECTOR
                      5060X
071.367  066 000      5061X $FCLO1   MVI     M,0
071.371  043          5062X          INX     H
071.372  014          5063X          INR     C
071.373  302 367 071  5064X          JNZ     $FCLO1
071.376  004          5065X          INR     B               COUNT ANOTHER FULL SECTOR
071.377  341          5066X $FCLO2   POP     H               (HL) = FB FWA
072.000  176          5067X          MOV     A,M             (A) = CHANNEL NUMBER
000.000               5068X          ERRNZ   FB.CHA
072.001  345          5069X          PUSH    H
072.002  377 005      5070X          DB      SYSCALL,.WRITE          FLUSH
                      5071X
                      5072X *        READY TO CLOSE FILE.
                      5073X *
                      5074X *        'C' SET IF ERROR
                      5075X *        (A) = ERROR CODE
                      5076X
072.004  341          5077X $FCLO3   POP     H               (HL) = FILE BLOCK ADDRESS
072.005  330          5078X          RC                      ERROR
000.000               5079X          ERRNZ   FB.CHA
072.006  176          5080X          MOV     A,M             (A) = CHANNEL NUMBER
072.007  345          5081X          PUSH    H
072.010  377 046      5082X          DB      SYSCALL,.CLOSE  CLOSE CHANNEL
072.012  341          5083X $FCLO4   POP     H               (HL) = FILE BLOCK ADDRESS
```

```
072.013  311            5084X          RET
072.014                 5085           XTEXT   FUTIL



                        5087X **       $FUTIL - UTILITY ROUTINES FOR FILE BLOCK ROUTINES.
                        5088X
                        5089X **       CBT - COPY BLOCK POINTERS TO TEMP CELLS.
                        5090X *
                        5091X *        ENTRY   (HL) = FILE BLOK FWA
                        5092X *        EXIT    NONE
                        5093X *        USES    A,F,H,L
                        5094X
072.014  325            5095X CBT      PUSH    D
072.015  305            5096X          PUSH    B               SAVE REGISTERS
000.000                 5097X          ERRNZ   TLEN-10         ASSUME 10 BYTES TO MOVE
072.016  021 156 072    5098X          LXI     D,T.CHA         (DE) = TARGET FOR MOVE
072.021  006 005        5099X          MVI     B,10/2
072.023  176            5100X CBT1     MOV     A,M             COPY FILE BUFFER INTO WORK AREA
072.024  022            5101X          STAX    D
072.025  043            5102X          INX     H
072.026  023            5103X          INX     D
072.027  176            5104X          MOV     A,M
072.030  022            5105X          STAX    D
072.031  043            5106X          INX     H
072.032  023            5107X          INX     D
072.033  005            5108X          DCR     B
072.034  302 023 072    5109X          JNZ     CBT1            MORE TO GO
072.037  301            5110X          POP     B
072.040  321            5111X          POP     D               (DE) = DATA TARGET ADDRESS
072.041  311            5112X          RET
                        5113X
                        5114X
                        5115X **       CTB - COPY TEMP CELLS BACK TO FILE BLOCK.
                        5116X *
                        5117X *        ENTRY   (HL) = FILE BLOCK ADDRESS
                        5118X *        EXIT    NONE
                        5119X *        USES    NONE
                        5120X
072.042  365            5121X CTB      PUSH    PSW
072.043  325            5122X          PUSH    D
072.044  305            5123X          PUSH    B
072.045  345            5124X          PUSH    H               SAVE REGISTERS
072.046  006 004        5125X          MVI     B,8/2
072.050  021 156 072    5126X          LXI     D,T.CHA
072.053  032            5127X CTB1     LDAX    D
072.054  167            5128X          MOV     M,A
072.055  023            5129X          INX     D
072.056  043            5130X          INX     H
072.057  032            5131X          LDAX    D
072.060  167            5132X          MOV     M,A
072.061  023            5133X          INX     D
072.062  043            5134X          INX     H
072.063  005            5135X          DCR     B
072.064  302 053 072    5136X          JNZ     CTB1            RESTORE FILE BUFFER VALUES
```

```
072.067  341          5137X          POP     H
072.070  301          5138X          POP     B
072.071  321          5139X          POP     D
072.072  361          5140X          POP     PSW
072.073  311          5141X          RET


                       5143X **       $FFB - FILE FILE BUFFER.
                       5144X *
                       5145X *        $FFB FILLS THE FILE BUFFER BY READING FROM THE FILE.
                       5146X *
                       5147X *        ENTRY   NONE
                       5148X *        EXIT    'C' SET IF READ INCOMPLETE
                       5149X *                  (A) = ERROR CODE
                       5150X *                'C' CLEAR IF READ COMPLETEE
                       5151X *                DATA IN BUFFER
                       5152X *        USES    A,F,D,E,H,L
                       5153X
                       5154X
072.074  072 170 072   5155X $FFB     LDA     EOFFLG
072.077  037           5156X          RAR
072.100  330           5157X          RC                      EOF
                       5158X
                       5159X *        CAN READ MORE, DO SO
                       5160X
072.101  305           5161X          PUSH    B               SAVE COUNT
072.102  052 160 072   5162X          LHLD    T.FWA
072.105  042 162 072   5163X          SHLD    T.PTR           CLEAR REMOVAL POINTER
072.110  353           5164X          XCHG
072.111  052 166 072   5165X          LHLD    T.LWA
072.114  042 164 072   5166X          SHLD    T.LIM           SET DATA LIMIT
072.117  175           5167X          MOV     A,L
072.120  223           5168X          SUB     E
072.121  117           5169X          MOV     C,A
072.122  174           5170X          MOV     A,H
072.123  232           5171X          SBB     D
072.124  107           5172X          MOV     B,A             (BC) = ROOM IN BUFFER
072.125  072 156 072   5173X          LDA     T.CHA
072.130  377 004       5174X          DB      SYSCALL,.READ   READ BUFFER
072.132  120           5175X          MOV     D,B             (D) = SECTORS UNREAD
072.133  301           5176X          POP     B               (BC) = DESIRED COUNT
072.134  320           5177X          RNC                     GOT THE DATA
                       5178X
                       5179X *        ERROR ON READ, SEE IF EOF
                       5180X
072.135  027           5181X          RAL
072.136  062 170 072   5182X          STA     EOFFLG          SET EOF, WE HOPE
072.141  376 003       5183X          CPI     EC,EOF*2+1
072.143  037           5184X          RAR
072.144  300           5185X          RNE                     IS NOT EOF, RETURN NOW!
072.145  072 165 072   5186X          LDA     T.LIM+1
072.150  222           5187X          SUB     D
072.151  062 165 072   5188X          STA     T.LIM+1         SET AMOUNT OF DATA WE DID GET
072.154  247           5189X          ANA     A
```

```
072.155  311          5190X          RET                    EXIT WITH DATA
                       5191X
                       5192X
                       5193X **       TEMP CELLS TO HOLD FILE BLOCK POINTERS DURING I/O
                       5194X
000.000                5195X          ERRNZ   FB.CHA
072.156  000           5196X T.CHA    DB      0              CHANNEL NUMBER
000.000                5197X          ERRNZ   *-T.CHA-FB.FLG
072.157  000           5198X T.FLG    DB      0              FLAG BYTE
000.000                5199X          ERRNZ   *-T.CHA-FB.FWA
072.160  000 000       5200X T.FWA    DW      0
000.000                5201X          ERRNZ   *-T.CHA-FB.PTR
072.162  000 000       5202X T.PTR    DW      0
000.000                5203X          ERRNZ   *-T.CHA-FB.LIM
072.164  000 000       5204X T.LIM    DW      0
000.000                5205X          ERRNZ   *-T.CHA-FB.LWA
072.166  000 000       5206X T.LWA    DW      0
000.012                5207X TLEN     EQU     *-T.CHA        LENGTH OF TEMP CELLS
                       5208X
072.170  000           5209X EOFFLG   DB      0
072.171                5210           XTEXT   FWRIB



                       5212X **       $FWR1B - WRITE BYTES FROM FILE BUFFER.
                       5213X *
                       5214X *        $FWR1B IS CALLED TO WRITE A NUMBER OF BYTES FROM A FILE BUFFER.
                       5215X *
                       5216X *        ENTRY   (BC) = BYTE COUNT
                       5217X *                (DE) = FWA FOR BYTES
                       5218X *                (HL) = ADDRESS OF FILE BUFFER
                       5219X *        EXIT    TO *FERROR* IF ERROR
                       5220X *                TO CALLER IF OK
                       5221X *                 (DE) = ADDRESS OF FIRST UNWRITTEN BYTE
                       5222X *        USES    A,F,B,C,D,E
                       5223X
                       5224X
072.171  315 200 072   5225X $FWRIB  CALL    $FWRIB.
072.174  320           5226X          RNC                    RETURN IF OK
072.175  303 057 073   5227X          JMP     $FERROR        ERROR
                       5228X
                       5229X
072.200                5230X $FWR1B.  EQU     *
072.200  345           5231X          PUSH    H
072.201  315 014 072   5232X          CALL    CBT            COPY BUFFER POINTERS TO TEMP CELLS
                       5233X
                       5234X *        COPY DATA FROM USER AREA TO BUFFER
                       5235X
072.204  325           5236X $WRIB2   PUSH    D              SAVE AREA ADDRESS
072.205  072 157 072   5237X          LDA     T.FLG
072.210  346 004       5238X          ANI     FT.OW          SEE IF OPEN FOR WRITE
072.212  312 346 072   5239X          JZ      $WRIB8         FILE NOT OPEN FOR WRITE
072.215  170           5240X          MOV     A,B
072.216  261           5241X          ORA     C
072.217  312 346 072   5242X          JZ      $WRIB8         ALL DONE
```

```
                        5243X
                        5244X *       COMPUTE MIN( ROOM IN BUFFER, WRITE COUNT REQUESTED)
                        5245X
072.222 052 162 072     5246X $WRIB3  LHLD    T.PTR
072.225 353             5247X         XCHG                    (DE) = (FB.PTR) = ADDRESS OF ROOM
072.226 052 166 072     5248X         LHLD    T.LWA           (HL) = LIMIT ADDRESS
072.231 175             5249X         MOV     A,L
072.232 223             5250X         SUB     E
072.233 157             5251X         MOV     L,A
072.234 174             5252X         MOV     A,H
072.235 232             5253X         SBB     D
072.236 147             5254X         MOV     H,A             (HL) = BYTES OF ROOM IN BUFFER
072.237 171             5255X         MOV     A,C             COMPARE REQUESTED COUNT TO BUFFER ROOM
072.240 225             5256X         SUB     L
072.241 170             5257X         MOV     A,B
072.242 234             5258X         SBB     H
072.243 322 250 072     5259X         JNC     $WRIB4          MORE REQUESTED THEN ROOM
072.246 140             5260X         MOV     H,B
072.247 151             5261X         MOV     L,C             USE REQUESTED COUNT
072.250 174             5262X $WRIB4  MOV     A,H
072.251 265             5263X         ORA     L
072.252 302 312 072     5264X         JNZ     $WRIB6          SOME ROOM IN BUFFER
                        5265X
                        5266X *       BUFFER IS FULL, EMPTY IT
                        5267X
072.255 305             5268X         PUSH    B               SAVE COUNT
072.256 052 160 072     5269X         LHLD    T.FWA
072.261 042 162 072     5270X         SHLD    T.PTR           CLEAR REMOVAL POINTER
072.264 353             5271X         XCHG
072.265 052 166 072     5272X         LHLD    T.LWA
072.270 175             5273X         MOV     A,L
072.271 223             5274X         SUB     E
072.272 117             5275X         MOV     C,A
072.273 174             5276X         MOV     A,H
072.274 232             5277X         SBB     D
072.275 107             5278X         MOV     B,A             (BC) = DATA IN BUFFER
072.276 072 156 072     5279X         LDA     T.CHA
072.301 377 005         5280X         DB      SYSCALL+,WRITE  WRITE BUFFER
072.303 301             5281X         POP     B               (BC) = DESIRED COUNT
072.304 322 222 072     5282X         JNC     $WRIB3          GOT THE DATA
                        5283X
                        5284X *       ERROR ON WRITE.
                        5285X
072.307 303 346 072     5286X         JMP     $WRIB8          HAVE ERROR
                        5287X
                        5288X *       GOT THE DATA, MOVE IT FROM BUFFER TO TARGET
                        5289X *
                        5290X *       (BC) = REQUEST COUNT
                        5291X *       (DE) = TO
                        5292X *       (HL) = COUNT
                        5293X *       ((SP)) = FROM
                        5294X
072.312 171             5295X $WRIB6  MOV     A,C
072.313 225             5296X         SUB     L
072.314 117             5297X         MOV     C,A
072.315 170             5298X         MOV     A,B
```

```
072.316  234            5299X          SBB     H
072.317  107            5300X          MOV     B,A             REMOVE BYTES ABOUT TO BE MOVED FROM REQUEST COUNT
072.320  305            5301X          PUSH    B
072.321  343            5302X          XTHL                    (HL) = REMAINING REQUEST COUNT
072.322  301            5303X          POP     B               (BC) = COUNT FOR THIS COPY
072.323  343            5304X          XTHL                    (HL) = TARGET ADDR, ((SP)) = REMAINING REQ. COUNT
072.324  176            5305X $WRIB7   MOV     A,M
072.325  022            5306X          STAX    D
072.326  023            5307X          INX     D
072.327  043            5308X          INX     H
072.330  013            5309X          DCX     B
072.331  170            5310X          MOV     A,B
072.332  261            5311X          ORA     C
072.333  302 324 072    5312X          JNZ     $WRIB7          MORE TO GO
072.336  353            5313X          XCHG
072.337  042 162 072    5314X          SHLD    T.PTR           UPDATE POINTER
072.342  301            5315X          POP     B               (BC) = REMAINING COUNT
072.343  303 204 072    5316X          JMP     $WRIB2          SEE IF MORE IN BUFFER
                        5317X
                        5318X *        WRITE COMPLETE.
                        5319X *
                        5320X *        (PSW) = COMPLETION FLAGS
                        5321X
072.346  321            5322X $WRIB8   POP     D               RESTORE TARGET ADDRESS
072.347  341            5323X          POP     H
072.350  303 042 072    5324X          JMP     CTB             COPY TEMP POINTERS BACK TO BLOCK, EXIT


                        5326X **       $FWBRK  -  BREAKOUTPUT                                  /80.02.GC/
                        5327X *
                        5328X *        $FWBRK empties the specified buffer by filling it with NULLs
                        5329X *        and then writing it.  Note this is used to insure that block
                        5330X *        mode I/O is output if it is not really a serial device (eg.
                        5331X *        writing to AT: from  *EDIT*.
                        5332X *
                        5333X *
                        5334X *        ENTRY:  HL      =  FILE BLOCK POINTER
                        5335X *
                        5336X *        EXIT:   HL      =  FILE BLOCK POINTER
                        5337X *                TO  $FERROR IF ERROR
                        5338X *
                        5339X *        USES:   PSW,BC,DE
                        5340X *
                        5341X
072.353  315 362 072    5342X $FWBRK   CALL    $FWBRK.
072.356  320            5343X          RNC                     NO ERROR
                        5344X
072.357  303 057 073    5345X          JMP     $FERROR
                        5346X
072.362  345            5347X $FWBRK.  PUSH    H
072.363  315 014 072    5348X          CALL    CBT             COPY BUFFER TO TEMPORARY
072.366  315 376 072    5349X          CALL    $FWBRK1
072.371  341            5350X          POP     H
072.372  315 042 072    5351X          CALL    CTB             COPY TEMPORARY TO BUFFER
```

```
072.375  311           5352X          RET
                       5353X
072.376  052 166 072  5354X $FWBRK1 LHLD    T.LWA
073.001  353          5355X          XCHG                DE = BUFFER LWA
073.002  052 162 072  5356X          LHLD    T.PTR       HL = BUFFER PTR
073.005  173          5357X          MOV     A,E
073.006  225          5358X          SUB     L
073.007  117          5359X          MOV     C,A
073.010  172          5360X          MOV     A,D
073.011  234          5361X          SBB     H
073.012  107          5362X          MOV     B,A         BC = DE - HL
073.013  261          5363X          ORA     C
073.014  310          5364X          RZ                  THE BUFFER IS ALREADY FLUSHED
                       5365X
                       5366X *        FILL THE BUFFER WITH NULLS
                       5367X
073.015  170          5368X FWBRK2   MOV     A,B
073.016  261          5369X          ORA     C
073.017  312 031 073  5370X          JZ      FWBRK3      NO MORE LEFT TO FILL
                       5371X
073.022  066 000      5372X          MVI     M,0
073.024  043          5373X          INX     H
073.025  013          5374X          DCX     B
073.026  303 015 073  5375X          JMP     FWBRK2
                       5376X
073.031  052 160 072  5377X FWBRK3   LHLD    T.FWA
073.034  042 162 072  5378X          SHLD    T.PTR
073.037  353          5379X          XCHG                DE = BUFFER FWA
073.040  052 166 072  5380X          LHLD    T.LWA       HL = BUFFER LWA
073.043  175          5381X          MOV     A,L
073.044  223          5382X          SUB     E
073.045  117          5383X          MOV     C,A
073.046  174          5384X          MOV     A,H
073.047  232          5385X          SBB     D
073.050  107          5386X          MOV     B,A         BC = HL - DE  ( BC = COUNT )
073.051  072 156 072  5387X          LDA     T.CHA
073.054  377 005      5388X          DB      SYSCALL,.WRITE
073.056  311          5389X          RET
073.057               5390           XTEXT   FERROR


                       5392X **       $FERROR - PROCESS FILE ERRORS.
                       5393X *
                       5394X *        $FERROR IS CALLED TO COMPLAIN ABOUT AN ERROR ENCOUNTERED
                       5395X *        WHEN PROCESSING FILES.
                       5396X *
                       5397X *        ENTRY   (A) = ERROR CODE
                       5398X *                (HL) = ADDRESS OF FILE NAME - FB.NAM
                       5399X *        EXIT    TO RESTART
                       5400X *        USES    ALL
                       5401X
                       5402X
073.057  365          5403X $FERROR PUSH    PSW         SAVE CODE
073.060  315 136 031  5404X          CALL    $TYPTX
```

```
073.063  012 007 105  5405X          DB      NL,BELL,'ERROR ON FILE',' '+2000Q
073.103  021 012 000  5406X          LXI     D,FB.NAM
073.106  031          5407X          DAD     D
                      5408X
                      5409X *        PRINT FILE NAME
                      5410X
073.107  176          5411X $FERR1   MOV     A,M
073.110  043          5412X          INX     H                ADVANCE MESSAGE
073.111  247          5413X          ANA     A
073.112  312 123 073  5414X          JZ      $FERR2
073.115  315 306 070  5415X          CALL    $WCHAR
073.120  303 107 073  5416X          JMP     $FERR1
                      5417X
                      5418X *        TYPE ERROR MESSAGE
                      5419X
073.123  315 136 031  5420X $FERR2   CALL    $TYP1X
073.126  040 055 240  5421X          DB      ' -',' '+2000
073.131  046 012       5422X          MVI     H,NL
073.133  361          5423X          POP     PSW              (A) = CODE
073.134  377 057       5424X          DB      SYSCALL,,ERROR
073.136  303 327 042  5425X          JMP     RESTART          EXIT
073.141               5426           XTEXT   DDS


                      5428X **       DDS     -  Decode Device Specification                  /80.05.sc/
                      5429X *
                      5430X *        DDS decodes the device specification, returnins a two character
                      5431X *        device name, and one byte unit number.
                      5432X *
                      5433X *
                      5434X *        ENTRY:  BC      = Address of destination fields
                      5435X *                DE      = Address of default
                      5436X *                HL      = Address of strins specifier
                      5437X *
                      5438X *        EXIT:   PSW     = 'C' SET   if ERROR
                      5439X *                          'C' CLEAR if NO ERROR
                      5440X *
                      5441X *        USES:   ALL
                      5442X *
                      5443X
073.141               5444X DDS      EQU     *
                      5445X
                      5446X *        Initialize the fields to the defaults
                      5447X
073.141  305          5448X          PUSH    B
073.142  315 251 073  5449X          CALL    DDS3
073.145  315 251 073  5450X          CALL    DDS3
073.150  032          5451X          LDAX    D
073.151  326 060      5452X          SUI     '0'
073.153  002          5453X          STAX    B
073.154  301          5454X          POP     B
                      5455X
073.155  176          5456X          MOV     A,M
073.156  247          5457X          ANA     A
```

```
073.157  310          5458X           RZ                      took the default
                       5459X
                       5460X *         Check the supplied name
                       5461X
073.160  315 261 073  5462X           CALL    $SOB            skip the whitespace
073.163  315 232 073  5463X           CALL    DDS2
073.166  330          5464X           RC                      Not alpha
073.167  315 232 073  5465X           CALL    DDS2
073.172  330          5466X           RC                      Not alpha
                       5467X
073.173  176          5468X           MOV     A,M
073.174  326 072      5469X           CPI     ':'
073.176  076 000      5470X           MVI     A,0             assume unit 0
073.200  312 214 073  5471X           JZ      DDS1            default to unit 0
                       5472X
                       5473X *         Check for a valid disit
                       5474X
073.203  176          5475X           MOV     A,M
073.204  326 060      5476X           SUI     '0'
073.206  330          5477X           RC                      Not disit
073.207  376 010      5478X           CPI     7+1
073.211  077          5479X           CMC
073.212  330          5480X           RC                      disit too large
073.213  043          5481X           INX     H
                       5482X
073.214  002          5483X DDS1      STAX    B
073.215  003          5484X           INX     B
073.216  176          5485X           MOV     A,M
073.217  043          5486X           INX     H
073.220  376 072      5487X           CPI     ':'
073.222  067          5488X           STC
073.223  300          5489X           RNZ                     requires ':'
                       5490X
073.224  176          5491X           MOV     A,M
073.225  247          5492X           ANA     A
073.226  067          5493X           STC
073.227  300          5494X           RNZ                     require 'NULL'
                       5495X
073.230  247          5496X           ANA     A               Clear ERROR flag
073.231  311          5497X           RET
                       5498X
073.232  176          5499X DDS2      MOV     A,M
073.233  043          5500X           INX     H
073.234  315 171 070  5501X           CALL    $MCU
073.237  376 101      5502X           CPI     'A'
073.241  330          5503X           RC                      Not alpha
                       5504X
073.242  376 133      5505X           CPI     'Z'+1
073.244  077          5506X           CMC
073.245  330          5507X           RC                      Not alpha
                       5508X
073.246  002          5509X           STAX    B
073.247  003          5510X           INX     B               replace the default char
073.250  311          5511X           RET
                       5512X
073.251  032          5513X DDS3      LDAX    D
```

```
073.252  023              5514X          INX     D
073.253  315 171 070      5515X          CALL    $MCU            Map to upper case
073.256  002              5516X          STAX    B
073.257  003              5517X          INX     B
073.260  311              5518X          RET
000.000                   5519X          ERRNZ   IOC.UNI-IOC.DEV-2       2 byte device
000.000                   5520X          ERRNZ   IOC.DIR-IOC.UNI-1       1 byte unit
073.261                   5521           XTEXT   SOB


                          5523X **       $SOB - SKIP OVER BLANKS.
                          5524X *
                          5525X *        $SOB IS CALLED TO SKIP AN ARBITRARILY LONG STRING OF BLANKS AND TABS.
                          5526X *
                          5527X *        ENTRY   (HL) = FWA OF (POSSIBLE) BLANK STRING
                          5528X *        EXIT    (HL) = LWA+1 OF BLANK STRING (UNCHANGED IF NO BLANKS)
                          5529X *                (A) = FIRST NON-BLANK, NON-TAB CHARACTER EEN
                          5530X *        USES    A,F,H,L
                          5531X
                          5532X
073.261  053              5533X $SOB     DCX     H               PRE-DECREMENT
073.262  043              5534X $SOB1    INX     H
073.263  176              5535X          MOV     A,M
073.264  376 040          5536X          CPI     ' '
073.266  312 262 073      5537X          JE      $SOB1           GOT BLANK
073.271  376 011          5538X          CPI     TAB
073.273  312 262 073      5539X          JE      $SOB1           GOT TAB
073.276  311              5540X          RET
073.277                   5541           XTEXT   PDD


                          5543X **       $PDD - PACK DECIMAL DIGITS.
                          5544X *
                          5545X *        $PDD PACKS A STRING OF DECIMAL DIGITS INTO A DECIMAL INTEGER.
                          5546X *
                          5547X *        THE CHARACTERS MUST BE IN MEMORY, AND BE IMMEDIATELY FOLLOWED BY A
                          5548X *        00 BYTE.
                          5549X *
                          5550X *        ENTRY   (HL) = ADDRESS OF CHARACTERS
                          5551X *        EXIT    'C' CLEAR IF OK
                          5552X *                (HL) = NUMBER
                          5553X *                'C' SET IF ERROR
                          5554X *        USES    A,F,D,E,H,L
                          5555X
                          5556X
073.277  353              5557X $PDD     XCHG                    (DE) = TEXT ADDRESS
073.300  041 000 000      5558X          LXI     H,0             (HL) = ACCUM
                          5559X
073.303  032              5560X $PDD1    LDAX    D
073.304  023              5561X          INX     D               ADVANCE ADDRESS
073.305  247              5562X          ANA     A
073.306  310              5563X          RZ                      ALL DONE
```

```
073.307  326 060      5564X          SUI     '0'
073.311  330          5565X          RC                      TOO SMALL
073.312  376 012      5566X          CPI     10
073.314  077          5567X          CMC
073.315  330          5568X          RC                      TOO SMALL
073.316  325          5569X          PUSH    D               SAVE (DE)
073.317  353          5570X          XCHG
073.320  315 324 030  5571X          CALL    $MU10
073.323  321          5572X          POP     D
073.324  330          5573X          RC                      OVERFLOW
073.325  205          5574X          ADD     L
073.326  157          5575X          MOV     L,A
073.327  076 000      5576X          MVI     A,0
073.331  214          5577X          ADC     H
073.332  147          5578X          MOV     H,A
073.333  322 303 073  5579X          JNC     $PDD1           NOT OVERFLOW
073.336  311          5580X          RET
073.337               5581           XTEXT   MU10                                    /071080/


                      5583X **       $MU10 - MULTIPLY UNSIGNED 16 BIT QUANTITY BY 10.
                      5584X *
                      5585X *        (HL) = (DE)*10
                      5586X *
                      5587X *        ENTRY   (DE) = MULTIPLIER
                      5588X *        EXIT    'C' CLEAR IF OK
                      5589X *                  (HL) = PRODUCT
                      5590X *                 'C' SET IF ERROR
                      5591X *        USES    D,E,H,L,F
                      5592X
                      5593X
030.324               5594X $MU10    EQU     30324A          IN H17 ROM
073.337               5595           XTEXT   DOS             DISMOUNT OPERATING SYSTEM


                      5597X **       $DOS - DISMOUNT OPERATING SYSTEM.
                      5598X *
                      5599X *        $DOS disounts all units of all directory devices        /80.04,sc/
                      5600X *
                      5601X *        THE USER IS MESSAGED ABOUT THE DISKS, AND THE OPERATING
                      5602X *        SYSTEM IS NOTIFIED.
                      5603X *
                      5604X *
                      5605X *        ENTRY   NONE
                      5606X *
                      5607X *        EXIT    (PSW)   = 'C' CLEAR IF NO ERROR
                      5608X *                          'C' SET   IF    ERROR
                      5609X *                         (A)  = ERROR CODE
                      5610X *
                      5611X *        USES    ALL
                      5612X *
                      5613X
```

```
073.337  315 136 031  5614X $DOS    CALL    $TYPTX
073.342  012 007 104  5615X          DB      NL,BELL,'Dismounting All Disks!',NL,ENL
                      5616X
073.374  315 071 074  5617X          CALL    $DOS.
073.377  330          5618X          RC
                      5619X
074.000  315 136 031  5620X          CALL    $TYPTX
074.003  012 122 145  5621X          DB      NL,'Remove the Disk(s). Hit RETURN when ready:',' '+2000
                      5622X
074.057  315 300 070  5623X DOS1     CALL    $RCHAR          READ CHARACTER
074.062  376 012      5624X          CPI     NL
074.064  302 057 074  5625X          JNE     DOS1
                      5626X
074.067  247          5627X          ANA     A               CLEAR CARRY
074.070  311          5628X          RET


074.071  076 000      5630X $DOS.    MVI     A,OVL0
074.073  377 010      5631X          SCALL   .LOADO
074.075  330          5632X          RC
                      5633X
074.076  076 001      5634X          MVI     A,OVL1
074.100  377 010      5635X          SCALL   .LOADO
074.102  330          5636X          RC
                      5637X
074.103  377 206      5638X          SCALL   .DAD            Dismount all Disks              /80.09.sc/
074.105  311          5639X          RET
                      5640
```

```
075.106  000          5643  HERRS   DB      0               HARD ERROR COUNTER
075.107                5644  INTDSK  DS      1       FLAG := <>0 INITIALIZED DISK MOUNTED
075.110                5645  LABEL   DS      256             LABEL SECTOR
075.110  000          5646  MYUNIT  DB      0               UNIT NUMBER REQUESTED
075.111                5647  PASS    DS      2               PASS NUMBER
075.113  000 000       5648  RSEED   DW      0               RANDOM NUMBER SEED
075.115  040 000       5649  SCT1WR  DW      32              1ST WRITEABLE SECTOR
                       5650
075.117  000           5651  AUXSTAT DB      0               AUXILARY STATUS SELECTED UNIT   /090980/
075.120  000           5652  CSN     DB      0               CYLINDER SECTOR NUMBER
075.121  001           5653  SECTOR  DB      1               SECTOR NUMBER
075.122  000           5654  SIDE    DB      0               SIDE NUMBER
075.123  000           5655  SPC     DB      0               SECTORS PER CYLINDER
075.124  000           5656  SPT     DB      0               SECTORS PER TRACK
075.125  000           5657  STC     DB      0               SECTOR TRANSFER COUNT
075.126  000           5658  TRACK   DB      0               TRACK NUMBER
075.127  000 000 000   5659  STATBL  DB      0,0,0,0         AUXILARY STATUS TABLE           /090980/
                       5660
075.133                5661  PATCH   DS      40              PATCH AREA
                       5662
075.203                5663  LINE    DS      32              LINE BUFFER
075.243                5664  MEML    EQU     *               MEM LWA


                       5666  **      MULTI-USE BUFFER
                       5667  *
                       5668  *       THIS FREE SPACE IS USED BY MANY ROUTINES.
                       5669
075.243                5670  BUFF    DS      256             ENOUGH FOR A SECTOR             /072080/
076.243                5671  SECERR  DS      2*NSPTD*NTRK-1/8+1      BAD SECTOR TABLE        *090980*
001.365                5672  SECERRL EQU     *-SECERR        LENGTH OF BAD SECTOR TABLE
076.243                5673  RRTA    EQU     SECERR          RANDOM TAG TABLE
001.365                5674  RRTAL   EQU     SECERRL         RANDOM TAG TABLE LENGTH         /072080/
                       5675
100.230                5676  RMEML   EQU     *               MINIMUM RUN TIME MEMORY LIMIT   *071080*
100.230                5677          END
ASSEMBLY COMPLETE
5677 STATEMENTS
   0 ERRORS DETECTED
8498 BYTES FREE
```

$CCO	070075	965	1010	1033	1112	1134	1171	1265	1894	1902	1931	4221L	
$CDEHL	030216	4237E											
$COMP	030060	2492	4258E										
$CRLF	070112	4270L											
$DADA	030072	1993	4614E	4630									
$DADA.	030101	1743	1807	2096	2138	2197	2212	2374	2387	2439	2452	2647	2667
		2740	2759	2780	4283E								
$DOS	073337	924	5614L										
$DOS.	074071	5617	5630L										
$DU86	030106	1792	1999	2262	2300	2318	3841	4357E	4636				
$ETL	064077	1325	2924	2930E									
$ETL.	064066	1139	1176	1270	1293	2922E							
$ETL0	064102	2939E	2950										
$ETL1	064127	2942	2954E										
$ETL2	064133	2946	2963E	2966									
$FCLO	071307	5018L											
$FCLO.	071316	1348	5018	5022L									
$FCLO1	071367	5061L	5064										
$FCLO2	071377	5057	5066L										
$FCLO3	072004	5030	5047	5077L									
$FCLO4	072012	5028	5083L										
$FERR1	073107	5411L	5416										
$FERR2	073123	5414	5420L										
$FERROR	073057	4917	4921	4925	5020	5227	5345	5403L					
$FFB	072074	5155L											
$FOPE1	071227	4947	4959L										
$FOPE2	071262	4983	4987L										
$FOPEA	071270	4987	4991E										
$FOPEB	071300	4981	5001L										
$FOPER	071147	4915L											
$FOPER.	071174	4915	4928L										
$FOPEU	071165	4923L											
$FOPEU.	071202	4923	4932L										
$FOPEW	071156	4919L											
$FOPEW.	071177	1312	4919	4930L									
$FWBRK	072353	5342L											
$FWBRK.	072362	5342	5347L										
$FWBRK1	072376	5349	5354L										
$FWRIB	072171	5225L											
$FWRIB.	072200	3052	3082	3232	5225	5230E							
$INDL	030234	4737E	5034	5037									
$INDLB	071026	4752L											
$INDS	071047	4785L											
$INDSB	071103	4815L											
$MCU	070171	4473	4492L	5501	5515								
$MLU	070151	1272	1296	4468L	4518								
$MLU1	070154	4471L	4476										
$MOVE	030252	4307E	4403										
$MOVEL	070126	1308	1326	2033	4387L								
$MU10	030324	5571	5594E										
$MU86	031007	2007	4344E										
$PDD	073277	1145	1182	5557L									
$PDD1	073303	5560L	5579										
$RCHAR	070300	1044	1123	1942	2940	2964	4522	4597L	4598	5623			
$RND	070364	2249	4676L										
$RSTALL	031047	1653	2475	2507	4225	4420E							
$RTL	070211	1906	4516	4520E									
$RTL.	070202	989	2489	4516L									

```
$RTL1   070212  4522L   4528
$RTL2   070244  4524    4545L
$SAVALL 031054  1645    2472    2501    4221    4434E
$SOB    073261  5462    5533L
$SOB1   073262  5534L   5537    5539
$TBLS   070247   974    4568L   4985
$TJMP   031061   986    4454E
$TJMP.  031062  4456E
$TYPTX  031136   955     966     976    1011    1034    1040    1113    1119    1135    1172    1232    1266
                1289    1321    1356    1878    1885    1895    1903    1932    1938    2018    2044    4718E   5404
                5420    5614    5620
$TYPTX. 031144  3117    3142    4720E
$UDDN   070311  1627    1648    2532    4629E
$WCHAR  070306   979    1039    1118    1937    4601L   5415
$WRIB2  072204  5236L   5316
$WRIB3  072222  5246L   5282
$WRIB4  072250  5259    5262L
$WRIB6  072312  5264    5295L
$WRIB7  072324  5305L   5312
$WRIB8  072346  5239    5242    5286    5322L
.ABUSS  040024   152E   1430    1566    1569    1573    1718    1815    1817
.ALARM  002136   125E
.ALEDS  040013   150E
.BLKW   000000  3240E   3479
.CHFLG  000060   803L
.CLEAN  000205   818L
.CLEAR  000055   800L    912
.CLEARA 000056   801L
.CLOSE  000046   793L   5082
.CLRCO  000007   777L    954
.CONSL  000006   776L   4224
.CRC    002347   133E
.CRCSUM 040027   153E
.CTC    002172   127E
.CTL2FL 040066   159E
.CTLC   000041   788L    929     944    1168    1230    1422    1557
.CTLFLG 040011   149E
.DAD    000206   819L   5638
.DEBUG. 000001    39E   1783    2502    2578    2597    2699    2783    2870
.DECODE 000053   798L
.DELET  000050   795L
.DISMT  000061   804L
.DLEDS  040021   151E
.DLY    000053   122E   2282    2284
.DMNMS  000203   816L
.DMOUN  000201   814L
.DOD    003122   136E
.DODA   003356   138E
.DSPMOD 040007   147E    915
.DSPROT 040006   146E    949
.DUMP   001374   124E
.ERROR  000057   802L   1015    5424
.EXIT   000000   770L   1026    4984    4990
.HORN   002140   126E
.IDENT  000000   121E
.IOWRK  040002   144E
.LINK   000040   787L
.LOAD   001267   123E
```

```
.LOADD  000062      805L
.LOADO  000010      778L      891     893    5831    5835
.MFLAG  040010      148E      917     919     951
.MONNS  000202      815L
.MOUNT  000200      813L
.NAME   000054      799L
.NMIRET 040064      158E
.OPEN   000063      806L
.OPENC  000045      792L
.OPENR  000042      789L     5001
.OPENU  000044      791L     5003
.OPENW  000043      790L     5002
.PCHL   002264      129E
.POSIT  000047      794L
.PRINT  000003      773L
.RCK    003260      137E
.READ   000004      774L     5174
.REGI   040005      145E
.REGPTR 040035      156E
.RENAM  000051      796L
.RESET  000204      817L
.RNB    002331      132E
.RNP    002325      131E
.SCIN   000001      771L     4597
.SCOUT  000002      772L     4271    4601
.SETTP  000052      797L      896
.SMALL  000001     3241E     3254    3568    3813    3834    3846    3864    4136
.SRS    002265      130E
.START  040000      143E
.SYSRES 000012      780L
.TICCNT 040033      155E      934
.TPERR  002205      128E
.TPERRX 040031      154E
.UIVEC  040037      157E
.VERS   000011      779L      903
.WNB    003024      135E
.WNP    003017      134E
.WRITE  000005      775L     5070    5280    5388
ABS.COD 000010      536L      872
ABS.ENT 000006      534L
ABS.ID  000000      530L
ABS.LDA 000002      532L
ABS.LEN 000004      533L
AIO.CGN 041047      687L
AIO.CHA 041116      702L
AIO.CNT 041111      698L
AIO.CSI 041050      686L
AIO.DDA 041041      683E
AIO.DES 041055      692L
AIO.DEV 041057      693L
AIO.DIR 041062      696L
AIO.DTA 041053      691L
AIO.EOF 041113      700L
AIO.EOM 041112      699L
AIO.FLG 041043      684L
AIO.GRT 041044      685L
AIO.LGN 041051      689L
AIO.LSI 041052      690L
```

```
AIO.SPG 041046   686L
AIO.TFP 041114   701L
AIO.UNI 041061   694L  1057  1201  2615  2645  2665  3907
AIO.VEC 041040   682L
ALIGN   045075   992   1111E
ALN$    045241  1122E  1125
ALN$$   045260  1133E  1140  1146  1150  1152
ALN$$.1 045343  1143   1147E
ALN$$.2 045366  1155   1157E
ALN0    045372  1163E  1238
ALN0.5  046005  1170E  1177  1183  1187
ALN0.7  046071  1180   1184E
ALN1.5  046165  1194   1218E
ALN2    046202  1214   1227E
ALN3    046253  1235E  1239
ALN4    046265  1229   1243E
ALNA    046273  1188   1206  1219  1248L
ALNB    046274  1165   1236  1245  1249L
ALNC    046275  1128   1192  1199  1250L
ALND    046276  1159   1202  1251L
AS.0DD  000100   240E  2695  3444  3820  3895
AS.1DD  000040   241E
AS.S1A  000020   242E  3830
AS.SLM  000003   243E
AUXSTAT 075117  2694   3443  3817  3829  3894  5651L
BELL    000007   168E   977  1012  1013  1357  1886  2019  2045  2634  5405  5615
BITS    070056  1797   2305  2323  4195L
BITS1   070063  4200L  4202
BKSP    000010   170E
BLK1    065177  3283   3293  3295L
BLK2    065206  3299L  3318
BLK3    065237  3304   3327E
BLK4    065247  3336   3338E
BLK5    065277  3350   3355E
BLKA    065324  3281   3291  3371E
BLKB    065326  3276   3286  3360  3373L
BLKC    001000  3374E
BLKRD   065156  2622   3285L
BLKWT   065132  2625   3275L
BOOT.P  000001   662E
BUFF    075243  1689   1750  1764  2102  2110  2163  2203  2556  2559  5670L
C.STX   000002   172E
C.SYN   000026   171E
CB.CLI  000100    67E    90
CB.MTL  000040    66E
CB.SPK  000200    68E
CB.SSI  000020    65E
CB2.CLI 000002    71E
CB2.ORG 000040    72E
CB2.SID 000100    73E
CB2.SSI 000001    70E
CBT     072014  5095L  5232  5348
CBT1    072023  5100L  5109
CCP     057231  1481   1497  2064E
CCP1    057250  2076E  2139
CCP2    057263  2088E  2098  2104  2121  2129  2131
CCP3    057335  2112E  2120
CCP4    057356  2114   2117  2125E
```

```
CCP5     057377    2092      2135E
CCPA     060013    2068      2083      2141L
CCPB     060046    2077      2095      2136      2145L
CCPC     060050    2065      2108      2146L
CDB.H84  000001     605E
CDB.H85  000000     604E
CEC      061326    1428      2472L     2535
CHKWR    054043    1744      1836E     2097      2198      2267      2375      2440
CHKWR1   054054    1837      1845E
CHKWR2   054067    1725      1839      1856E     2078      2179      2356      2421
CIP      061173    1511      2408E
CIP1     061207    2419E     2453
CIP2     061222    2431E     2441      2445
CIP3     061255    2435      2449E
CIPA     061271    2411      2426      2455L
CIPB     061324    2420      2438      2450      2459L
CLEAN    044177     991      1032E
CLN0     044341    1043E     1048
CLN3     045035    1083E     1101
CLN5     045055    1094E     1097
CLNA     045074    1081      1099      1105L
CN.170M  000014     108E
CN.174M  000003     107E
CN.ABO   000200     112E
CN.BAU   000100     111E
CN.MEM   000040     110E
CN.PRI   000020     109E
CND.H17  000000     114E
CND.H47  000001     116E
CND.NDT  000000     115E
CO.FLG   000001     515E      4223
COM      065327    1064      1084      1208      3389L     3435      3707
COM.     065333    3361      3394L
COM..    065343    3398L     3954
COM1     065352    3396      3402L
CR       000015     164E
CS.FLG   000200     516E
CSL.CHR  000001     492E
CSL.ECH  000200     489E
CSL.RAW  000004     490E
CSL.WRP  000002     491E
CSN      075120    3848      4033      4036      4043      4053      5852L
CSV      053103    1568      1572      1576      1684E
CSV2     053151    1699      1709      1714E
CSV2.1   053163    1723E     1814      1818
CSV4     053176    1735E     1745      1759      1778      1800
CSV4.7   053265    1766E     1776
CSV5     053310    1752      1769      1773      1782E
CSV7     053340    1739      1804E
CSVA     054001    1715      1748      1757      1820L
CSVB     054002    1724      1742      1805      1821L
CSVC     054004    1746      1788      1822L
CSVD     054006    1695      1704      1730      1823L
CSVG     054041    1685      1762      1826L
CTB      072042    5121L     5324      5351
CTB1     072053    5127L     5136
CTLA     000001     179E
CTLB     000002     180E
```

CTLC	000003	181E	928	943	1166	1228	1420	1555						
CTLD	000004	182E	4523											
CTLQ	000017	183E												
CTLP	000020	184E												
CTLQ	000021	185E												
CTLS	000023	186E												
CTLZ	000032	187E												
CTP.2SB	000010	501E												
CTP.BKM	000002	502E												
CTP.BKS	000200	497E												
CTP.FF	000100	498E												
CTP.MLI	000040	499E												
CTP.MLO	000020	500E												
CTP.TAB	000001	503E												
CYR	061340	1897	2488L											
CYRA	061356	2490	2494L											
D.CON	040110	451L												
D.DAT	000171	211E	3471	3552	3563	3589	3642	3679						
D.RAM	040240	454L												
D.STA	000170	210E	211	1095	3542	3633	3670	3744	3755	3942	3946	3967	3971	
		4095	4160	4171										
D.VEC	040130	453L												
DC.ABT	000007	432L	952	1461	1634	2618								
DC.CLO	000006	431L												
DC.LOD	000011	434L												
DC.MAX	000013	436L												
DC.MOU	000010	433L	1951	2627										
DC.OPR	000003	428L												
DC.OPU	000005	430L												
DC.OPW	000004	429L												
DC.RDY	000012	435L												
DC.REA	000000	425L	1707	1758	1970	2100	2554	2621						
DC.RER	000002	427L												
DC.WRI	000001	426L	1698	2037	2201	2624	2838							
DD.BOOT	000000	249L												
DD.CPY	000013	260L												
DD.DS	000202	284L												
DD.FRM0	000014	261L												
DD.FRM1	000015	262L												
DD.FRM2	000016	263L												
DD.FRM3	000017	264L												
DD.LSC	000003	252L	3436											
DD.RAD	000004	253L												
DD.RAS	000002	251L	3708											
DD.RDBL	000205	287L												
DD.RDL	000203	285L												
DD.REA	000005	254L												
DD.REAB	000007	256L	3285											
DD.RRDY	000020	265L												
DD.RST	000001	250L	3953											
DD.SDC	000200	282L												
DD.SPF0	000020	271L												
DD.SPF1	000021	272L												
DD.SPF2	000022	273L												
DD.SPF3	000023	274L												
DD.SPF4	000024	275L												
DD.SPF5	000025	276L												
DD.ST	000201	283L												

```
DD.WDLB 000210      290L
DD.WRBD 000012      259L
DD.WRD  000011      258L
DD.WRI  000006      255L
DD.WRIB 000010      257L    3275
DD.WTBL 000206      288L
DD.WTDL 000207      289L
DD.WTL  000204      286L
DDF.BOL 000011      359E
DDF.BUO 000000      358L
DDF.LAB 000011      360L    1973    2040
DDF.USR 000012      361L    1990
DDRV    062157      953     1462    1635    1751    1952    1974    2041    2593    2612E
DDRV.   062133      2103    2204    2557    2591E   2841
DDRV1   062252      2628    2640E
DDRV1.  062155      2602    2604L
DDS     073141      1300    5444E
DDS1    073214      5471    5483L
DDS2    073232      5463    5465    5499L
DDS3    073251      5449    5450    5513L
DEFERR  067274      3296    3777    3936    3990    3999L
DEV.DDA 000004      563L
DEV.DVG 000015      576L
DEV.DVL 000013      575L
DEV.FLG 000006      564L
DEV.JMP 000003      562L
DEV.MNU 000010      572L
DEV.MUM 000007      571L
DEV.NAM 000000      554L
DEV.RES 000002      558L
DEV.UNT 000011      573L
DEVELEN 000016      578E
DF.CLR  000376      329E
DF.EMP  000377      328E
DIAG2   044077      975     984E
DIAGA   044121      973     998L
DIR.ALD 000025      344L
DIR.CLU 000015      337L
DIR.CRD 000023      343L
DIR.EXT 000010      332L
DIR.FGN 000020      340L
DIR.FLG 000016      338L
DIR.LGN 000021      341L
DIR.LSI 000022      342L
DIR.NAM 000000      331L
DIR.PRO 000013      333L
DIR.VER 000014      334L
DIRELEN 000027      346E    696     865
DIRIDL  000015      335E
DLY     065355      3399    3417L   3745
DLY1    065360      3419L   3420
DM.MR   000000       80E
DM.MW   000001       81E
DM.RR   000002       82E
DM.RW   000003       83E
DUS1    074057      5623L   5625
DR.IM   000001      559E
DR.PR   000002      560E
```

```
DRIVE   051204   987     1411E
DRIVE1  051233   1428L   1456
DRIVEA  052046   1424    1521L
DRIVEA1 052112   1523L   1914
DRIVEB  052115   1452    1525L
DRIVEB1 052132   1451    1527L
DT.CH   000020   569E
DT.CR   000002   566E
DT.CW   000004   567E
DT.DD   000001   565E
DT.RN   000010   568E
DUN     054102   930     1878L
DUN1    054176   1885L
DUN2    055205   1902L   1909    1911
DV.EL   000000   555E
DV.NU   000001   556E
EC.CNA  00000A   715L
EC.DDA  000027   734L
EC.DIF  000017   726L
EC.DIW  000035   740L
EC.DNI  000045   748L
EC.DNR  000046   749L
EC.DNS  000005   716L
EC.DSC  000047   750L
EC.EOF  000001   712L    5183
EC.EOM  000002   713L
EC.FAD  000031   736L    4954
EC.FAP  000026   733L
EC.FL   000030   735L
EC.FNF  000014   723L
EC.FND  000011   720L
EC.FNR  000034   739L
EC.FOD  000043   746L
EC.FUC  000013   722L
EC.ICN  000016   725L
EC.IDN  000006   717L
EC.IFC  000020   727L
EC.IFN  000007   718L
EC.ILC  000003   714L
EC.ILO  000040   743L
EC.ILR  000012   721L
EC.ILV  000037   742L
EC.IOI  000052   753L
EC.IS   000032   737L
EC.NCV  000050   751L    907
EC.NEM  000021   728L     898
EC.NOS  000051   752L
EC.NPM  000044   747L
EC.NRD  000010   719L
EC.NVM  000042   745L
EC.OTL  000053   754L
EC.RF   000022   729L    3288
EC.UNA  000036   741L
EC.UND  000015   724L
EC.UUN  000033   738L
EC.VPM  000041   744L
EC.WF   000023   730L    3278
EC.WP   000025   732L    3981
```

```
EC.WPV  000024    731L
ENL     000212    177E    984    1233   1358   1881   1893   2023   2045   5615
EOFFLG  072170   5155    5182   5209L
ERROR   044140    900     909    925    1009L
ESC     000033    175E
EXIT    044171    989    1016   1023E  1898
FAS     062273   2663E   3816
FB.CHA  000000    825L   4938   4941   5068   5079   5195
FB.FLG  000001    826L   4941   4958   4996   5023   5035   5038   5197
FB.FWA  000002    827L   4958   4964   5038   5199
FB.LIM  000006    829L   4969   4974   5203
FB.LWA  000010    830L   5205
FB.NAM  000012    831L    832   1309   4974   5406
FB.NAML 000021    832E   1380
FB.PTR  000004    828L   4964   4969   5035   5201
FBENL   000033    833E   1381
FF      000014    178E   3100
FILL    064147   1320    1564   2032   2164   2242   2983E
FILL1   064151   2986E   2992
FILLW   064163   1690    3007E  3015
FNP     064176   3031E   3180
FNP.    064330   1346    3090E
FNP0    064203   3038E   3093
FNP1    064221   3041    3048E
FNP2    064235   3044    3056E
FNP3    064267   3070    3072E
FNPA    064345   1334    3039   3043   3098L
FNPB    064346   3033    3057   3092   3099L
FNPC    064347   3050    3100L
FNPD    000074   3083    3094   3101E
FT.ABS  000000    544E     873
FT.BAC  000003    547E
FT.DD   000001    844E
FT.OC   000020    848E
FT.OR   000002    845E   4928   4932   5001   5003
FT.OU   000010    847E
FT.OW   000004    846E   4930   4932   5002   5003   5029   5238
FT.PIC  000001    545E
FT.REL  000002    546E
FWBRK2  073015   5368L   5375
FWBRK3  073031   5370    5377L
HERRS   074106   2127    2130   2474   2521   2527   2582   2585   2600   2603   5643L
HLCPDE  070120   1850    1861   4324L
I.CONFL 000004    518E    519   4222
I.CONTY 000001    505E    506
I.CONWI 000003    511E    512
I.CSLMD 000000    494E
I.CUSOR 000002    508E    509
IERR1   061361   2126    2501L  2577
ILDEHL  071124   4788    4850L
INTDSK  074107   1049    1129   1917   1925   1946   2025   2047   5644L
INTRLV  062310   1696    1705   2069   2170   2347   2412   2691E
INTRLV0 062324   2696    2710E
INTRLV1 062327   2718E   2772   2774
INTRLV2 062337   2729E   2750   2752
INTRLV3 062340   2732E   2743
INTRLV4 062377   2737    2756E
INTRLV6 063026   2722    2778E
```

```
INTRLVA 063037  2692  2739  2757  2779  2818L
IOC.CGN 000010  853L
IOC.CSI 000011  854L
IOC.DDA 000002  841L  849   863
IOC.DES 000016  860L
IOC.DEV 000020  861L  5519
IOC.DIL 000021  863E
IOC.DIR 000023  865L  5520
IOC.DRL 000010  857E
IOC.DTA 000014  859L
IOC.FLG 000004  843L  857
IOC.GRT 000005  851L
IOC.LGN 000012  855L
IOC.LNK 000000  840L
IOC.LSI 000013  856L
IOC.SPG 000007  852L
IOC.SQL 000003  849E
IOC.UNI 000022  862L  5519  5520
IOCCTD  000001  869E
IOCELEN 000052  867E
IP.CON  000362  56E
IP.PAD  000360  52E
ISDEHL  071131  4795  4871L
LAB.AUX 000117  396E  398
LAB.AXL 000001  398E
LAB.DAT 000000  373E
LAB.DIS 000003  369L
LAB.GRT 000005  370L
LAB.IND 000001  368L
LAB.LAB 000021  392L  393   2031  2034
LAB.LBL 000074  393E  2030
LAB.NOD 000002  375E  2035
LAB.PSS 000016  384L
LAB.RGT 000012  380L
LAB.SER 000000  367L
LAB.SIZ 000014  383L  1859  1985  2260
LAB.SPG 000007  371L  1991  1996  2005
LAB.SPT 000117  397L
LAB.SYS 000001  374E
LAB.VER 000011  378L  1977
LAB.VFL 000020  385L  1808
LAB.VLT 000010  377L  2036
LAB.VPL 000005  387E  389   390
LAB.VPR 000014  382E  387
LABEL   074110  1808  1859  1972  1977  1985  1991  1996  2005  2031  2034  2036  2039
                2260  5645L
LF      000012  165E
LINE    075203  968   978   1138  1144  1175  1181  1269  1271  1273  1292  1295  1299
                1905  2488  5663L
LSC     065365  3357  3435L
LSC1    066013  3446  3448L
M.FOX   000303  100E
M.INI   242355  204E
M.OUTI  243355  205E
M.PAMB  000021  99E
MEDIA   052135  988   1546E
MEDIA10 052340  1604  1645L
MEDIA5  052250  1590E 1617
```

MEDIA6	052260	1601E	1610											
MEDIA7	052305	1621E												
MEDIA8	052330	1556	1633E											
MEDIAA	052370	1560	1622	1651	1655L									
MEDIAB	052372	1581	1656L											
MEDIAB1	053020	1658L	1915											
MEDIAC	053023	1628	1660L											
MEDIAC1	053024	1625	1662L											
MEDIAD	053056	1649	1664L											
MEDIAD1	053065	1647	1666L											
MEML	075243	875	5664E											
MOUNT	042301	926E	927	990										
MYUNIT	075110	1037	1056	1116	1200	1912	1935	2614	5646L					
NL	000012	176E	177	956	956	956	958	960	962	963	964	1012	1035	
		1041	1045	1114	1120	1124	1136	1173	1267	1290	1322	1357	1358	1369
		1369	1522	1524	1528	1657	1659	1661	1663	1667	1879	1880	1881	1886
		1888	1890	1892	1896	1904	1933	1934	1940	1943	2019	2020	2021	2045
		2634	2634	2941	2965	3190	4270	4527	5405	5422	5615	5615	5621	5624
NSPTD	000032	194E	3824	5671										
NSPTS	000015	193E	3821	3824										
NTRK	000115	196E	1186	5671										
NUL2	000000	167E												
NULL	000200	166E												
OP.CTL	000360	53E												
OP.DIG	000360	54E												
OP.SEG	000361	55E												
OP2.CTL	000362	57E												
OUT	066022	1074	1087	3440	3448	3466L	3914	4014						
OUT.	066032	3398	3471L											
OUT0	066035	3468	3474L											
OUT1	066074	3542L	3547	3555										
OUT128	066066	3503	3536L											
OUT2	066111	3544	3551L											
OUTB	066040	3280	3498L	3513										
OVL.COD	000000	405L												
OVL.ENS	000010	410E												
OVL.ENT	000004	407L												
OVL.FLB	000006	408L												
OVL.IN	000001	629E												
OVL.NUM	000014	631E												
OVL.RES	000002	630E												
OVL.SIZ	000002	406L												
OVL.UCS	000200	632E												
OVL0	000000	416L	890	5630										
OVL1	000001	417L	892	5634										
PASS	075111	937	1427	1445	5647L									
PATCH	075133	5661L												
PIN	066133	3586L	3712											
PIN.	066137	3589L	3976											
PIN1.	066230	3629	3670L	3675	3683									
PIN2	066245	3672	3679L											
PIN256	066222	3615	3664L											
PINB	066142	3290	3610L											
PINB1	066147	3614L	3620											
PINB2	066161	3612	3622L											
PINB3	066174	3633L	3638	3644										
PINB4	066211	3635	3642L											
PMSG	064350	1425	1453	1629	1650	2534	3112E							

```
PMSG,   064371  1470    1478    1486    1494    1502    1509    1517    2633    3134E
PSE     061367  1432    1434    1436    1438    1440    1442    1444    2521L
PSE1    061374  2527L
PSEA    062023  2533    2537L
PSEB    062024  2530    2538L
QUOTE   000047   173E
RAS     066257  2641    3702L
RAS1    066313  3717    3722L
REPORT  046277   993    1257E
RESTART 042327   941E    942     946     972     980    1167    1464    1637    2026    2048    5425
RID     055301  1924E   1969
RL      056103  1415    1550    1968E
RL1     056145  1979    1989E
RL1.5   056202  2002    2004E
RL9     056216  1975    2017E
RLP     062030  2329    2444    2552L
RLP0    062036  2554L
RLP1    062064  2562L   2573
RLP2    062106  2558    2574L   2584    2586
RLPERR  062112  2564    2569    2577L
RMEML   100230   895    5676E
RND1    070372  4680L   4699
ROMBOOT 030000   446E
RPT1    047016  1288E   1294    1301
RPT8    047303  1024    1261    1342E
RPT9    047326  1313    1355E
RPTA    050021  1361L   1370    3080
RPTA1   050021  1319    1324    1362L   1363
RPTA1L  000062  1318    1323    1363E
RPTA2   050110  1327    1365L
RPTA3   050133  1368L   3074    3078
RPTAL   000116  1370E   3079
RPTB    050137  1337    1371L   3060    3063
RPTC    050140  1332    1343    1350    1372L   3170
RPTD    050141  1309    1311    1347    1373L   1381    3051    3081    3231
RPTE    050174  1376    1377    1378    1379    1382L   1383
RPTEL   001000  1379    1383E
RPTF    051174  1298    1384L
RPTG    051177  1297    1305    1307    1309    1385L
RRT     060224  1519    2235E
RRT00   060244  2249L   2270    2289
RRT1.3  060303  2268    2272L
RRT1.4  060340  2279    2286E
RRT1.5  060346  2273    2292L
RRT2    061002  2292    2313E
RRT2.5  061032  2325    2327E
RRTA    076243  2241    2301    2319    5673E
RRTAL   001365  2240    5674E
RRTB    061036  2247    2274    2276    2331L
RSEED   075113   935    4676    4700    5648L
RST     066316  2619    3738L   3934
RST.    066324  3739    3743L
RST1    066336  3750L   3757
RSTA    000000  3749    3761E
RUBOUT  000177   169E
RZLA    057172  2034    2050L   2051
RZLAL   000037  2034    2051E
S.BAUD  040344   606L
```

```
S.BDA   041120      704L
S.BOOTF 041034      661L
S.CAADR 040333      522L
S.CACC  041006      645L
S.CCTAB 040335      523L
S.CDB   040343      603L
S.CFWA  040352      613L
S.CODE  041007      646L
S.CONFL 040332      520L
S.CONTY 040327      507L
S.CONWI 040331      513L
S.CSLMD 040326      495L    506   509   512   519   914
S.CUSOR 040330      510L
S.DATC  040310      476L
S.DATE  040277      475L   1327
S.DCS   041033      659L
S.DDDTA 040366      624L
S.DDGRP 040364      621L
S.DDLDA 040360      619L
S.DDLEN 040362      620L
S.DDOPC 040370      625L
S.DFWA  040354      614L
S.DIREA 041016      653L
S.DLINK 040346      611L
S.DON   000040      214E   1096  3543  3634  3671  3756  3943  3968  4096  4161
S.DTR   000200      216E   3543  3545  3634  3636  3671  3673  3972  4172
S.ERR   000001      213E   3947
S.FASER 041013      652L
S.FCI   041021      654L
S.GRT0  024000      442E
S.GRT1  025000      443E
S.GRT2  026000      444E
S.GUP   041027      656L
S.HIMEM 040316      478L
S.IEN   000100      215E
S.INT   040343      456L    599
S.JUMPS 041010      650L
S.MOUNT 041032      658L
S.OFWA  040350      612L
S.OMAX  040324      484L
S.OSN   041004      641L
S.OVLE  041000      638L
S.OVLFL 040371      634L
S.OVLS  040376      637L
S.OVSTK 041035      666L
S.RFWA  040356      615L
S.SCI   041024      655L
S.SCR   041121      705L
S.SDD   041010      651L
S.SOVR  041146      458L    460
S.SSN   041002      640L
S.SW0   000002      218E
S.SW1   000004      219E
S.SW2   000010      220E
S.SW3   000020      221E
S.SYSM  040320      480L
S.TIME  040312      477L
S.UCSF  040372      635L
```

```
S.UCSL  040374  636L
S.USRM  040322  482L
S.VAL   040277  455L  473
SB.BTD  000001  235E
SB.CRC  000010  232E
SB.DLD  000040  230E
SB.ILC  000002  234E
SB.LTD  000004  233E
SB.NRF  000020  231E
SB.UNR  000200  228E
SB.WPD  000100  229E  3980
SCT1WR  075115  1848  2008  5649L
SDE     066354  3277  3287  3774L
SDP     066364  3295  3797E
SDP.    066374  3801  3807L
SDP1    067020  3822  3825L
SDP2    067037  3831  3833E
SEC.M   000037   311E
SECERR  076243  1563  1586  1793  5671L  5672  5673
SECERRL 001365  1562  1585  5672E  5674
SECTOR  075121  1063  1205  3345  3703  3704  3851  3860  3886  3896  4055  4063  5653L
SID.0   000000   304E   307  1058  1153  3705  3810  4051
SID.1   000200   305E   307  1156  3861  4064
SID.M   000200   307E
SIDE    075122  1059  1203  1812  3704  3811  3862  3903  4052  4065  5654L
SPC     075123  3837  4040  5655L
SPT     075124  1806  2137  2211  2386  2451  2720  2748  2770  3344  3825  3853  4056
                5656L
SSIZ.M  004000   315E
STACK   042200   462E   945
STACKL  001032   460E
STATBL  075127  2646  2666  5659L
STC     075125  3341  3352  3445  4034  5657L
SUS     067112  3365  3889L
SUS.    067120  3890  3894L
SUS..   067104  1067  1088  1211  3710  3886L
SUS1    067136  3887  3897  3901L
SYDD    040130   452E
SYSCALL 000377   763E   891   893   896   903   912   929   944   954  1015  1026  1168
                1230  1422  1557  4224  4271  4597  4601  4990  5070  5082  5174  5280  5388
                5424
T.CHA   072156  5098  5126  5173  5196L  5197  5199  5201  5203  5205  5207  5279  5387
T.FLG   072157  5198L  5237
T.FWA   072160  5162  5200L  5269  5377
T.LIM   072164  5166  5186  5188  5204L
T.LWA   072166  5165  5206L  5248  5272  5354  5380
T.PTR   072162  5163  5202L  5246  5270  5314  5356  5378
TAB     000011   174E  1879  1879  1879  1880  1880  1880  1881  1881  1886  5538
TBL1    070256  4572L  4578
TBL2    070274  4570  4582L
TBL3    070276  4575  4586L
TEB     067171  1068  1089  1212  1222  3937  3941L  4099
TEB.    067161  3549  3640  3677  3753  3936L  4162
TEB..   067156  3934L  4093  4169
TEB1    067225  3962L  3973
TEB2    067264  3965  3969  3986L
TEB3    067265  3955  3990L
TEB4    067271  3944  3948  3995L
```

```
TEST     042200    876      888E
TEST0    042226    897      902E
TEST1    042240    904      907L
TEST2    042246    906      911L
TEST5    051347   1421     1460E
TESTA    051357   1431     1469E
TESTB    051367   1433     1477E
TESTC    051377   1435     1485E
TESTD    052010   1437     1493E
TESTE    052021   1439     1501E
TESTF    052030   1441     1508E
TESTG    052037   1443     1516E
TLEN     000012   5097     5207E
TRACK    075126   1061     1207   1220   1429   3844   4013   4044   5658L
TRK      067275   1066     1210   1221   3364   4013L
UDDN1    070315   4633L    4649
UDDN1.5  070347   4653L    4660
UDDN2    070351   4646     4658L
UDDN3    070352   4659L    4663
UNT.0    000000    295E     300
UNT.1    000040    296E     300
UNT.2    000100    297E     300
UNT.3    000140    298E     300
UNT.DIS  000006    588L
UNT.FLG  000000    584L
UNT.GRT  000002    586L
UNT.GTS  000004    587L
UNT.M    000140    300E    3911
UNT.SIZ  000010    590E
UNT.SPG  000001    585L
UO.CLK   000001     92E     950
UO.DDU   000002     91E     918
UO.HLT   000200     89E     950
UO.NFR   000100     90E     918
USERFWA  042200    463E     872    874    875
USN      067303   3315    4029E
USN1     067333   4042    4046E
VERS     000040    761E     905   1880   1880
VFL.NSD  000001    386E    1809
W.RES    000002    223E    3743
WARN2.5  056057   1941E    1944
WCP      060051   1473    1489   2158E
WCP1     060076   2177E   2213
WCP2     060111   2189E   2199   2205
WCP3     060153   2193    2209E
WCPA     060167   2169    2184   2215L
WCPB     060222   2178    2196   2210   2219L
WDN      067367   1075    3317   3395   3451   3500   3624   3646   3716   4083L
WDN.     067375   4084    4088L
WDN1     070000   4090L   4097
WDNA     000000   4088    4101E
WIP      061040   1504    2343E
WIP1     061054   2354E   2388
WIP2     061067   2366E   2376   2380
WIP3     061122   2370    2384E
WIPA     061136   2346    2361   2390L
WIPB     061171   2355    2373   2385   2394L
WLP      063041   2309    2379   2836L
```

WLPB	063066	2552	2560	2574	2834	2840	2842	2845L	2869
WLPC	063070	2553	2837	2846L					
WRTL	065012	1582	3120	3140	3169E				
WRTL0	065017	3174E	3210						
WRTL1	065036	3178	3185E	3194					
WRTL2	065061	3191	3203E						
WRTL3	065101	3198	3204	3215E					
WRTLA	065127	1336	3085	3095	3176	3205	3207	3237L	
WRTLB	065130	3175	3219	3238L					
WTR	070020	3467	3536	3559	3586	3664	4153L		
WTR.	070026	4154	4158L						
WTR1	070031	4160L	4173						
WTRA	000000	3960	4158	4177E					
XCHGBC	071136	4785	4789	4797	4799	4891L			
ZL	057051	1416	1551	2028E					

11904 BYTES FREE