```
000.001          PUBLIC  EQU     1                NOT PUBLIC
                 ***     INITIALIZE - INITIALIZE DISK.
                 *
                 *        J. G. L.,   10/77
                 *
                 *       COPYRIGHT 10/77, HEATH CO.
                 *       COPYRIGHT 79/04, HEATH CO.
                 *
                 *          G. C.,        78/10        Maintenence release
                 *                        79/04        Renamed INIT from INIT17 for H89
                 *                                     compatibility reasons.
                 *                        80/05        Arbitrary Mass Storage
                 *                        /2.0b/  = /80.10.sc/
                 *


                 ***     INITIALIZE INITIALIZES A NEW DISK VOLUME.
                 *
                 *       ACTIONS TAKEN:
                 *
                 *       1) REQUEST MEDIA INSERTION
                 *       2) CHECK FOR PROPER HOLE PATTERN
                 *       3) READ LABEL AND TYPE INFORMATION
                 *       4) ASK FOR # OF TRACKS
                 *       5) ASK FOR DISK SERIAL NUMBER AND LABEL
                 *       6) ZERO AND INITIALIZE TRACKS
                 *       7) ASK FOR LIST OF BAD SECTORS
                 *       8) FORMAT DEVICE
                 *       9) PROMPT RESTORATION OF SYSTEM DEVICE


000.303          MI.JMP  EQU     303Q

000.002          SPG     EQU     2               2 SECTORS PER GROUP
001.220          VOLSIZ  EQU     400             400 SECTORS PER VOLUME

000.001          CN.FDP  EQU     1               Parameter File Channel

000.004          DVT.MAX EQU     4               Maximum of 4 Device Entries
```

```
     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.gc/
        000.362                    57X OP2.CTL  EQU     362Q            H-88/H-89/HA-8-8 Control Port   /80.07.gc/


                                   59X **       FRONT PANEL CONTROL BITS.                               /80.07.gc/
                                   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.gc/
                              103X *
                              104X *        These bits are read in IP.CON.
                              105X *
                              106X
     000.003                  107X CN.174M EQU      00000011B       Port 174Q Device-Type Mask
     000.014                  108X CN.170M EQU      00001100B       Port 170Q 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.NDI 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
```

```
002.347                 133X .CRC     EQU     2347A          CRC-16 CALCULATOR
003.017                 134X .WNP     EQU     3017A          WRITE NEXT PAIR
003.024                 135X .WNB     EQU     3024A          WRITE NEXT BYTE
003.122                 136X .DOD     EQU     3122A          DECODE FOR OCTAL DISPLAY
003.260                 137X .RCK     EQU     3260A          READ CONSOLE KEYSET
003.356                 138X .DODA    EQU     3356A          SEGMENT CODE TABLE


                        140X **       RAM CELLS USED BY H8MTR.
                        141X *
                        142X
040.000                 143X .START   EQU     40000A         START DUMP ADDRESS
040.002                 144X .IOWRK   EQU     40002A         IN OR OUT INSTRUCTION
040.005                 145X .REGI    EQU     40005A         DISPLAYED REGISTER INDEX
040.006                 146X .DSPROT  EQU     40006A         PERIOD FLAG BYTE
040.007                 147X .DSPMOD  EQU     40007A         DISPLAY MODE
040.010                 148X .MFLAG   EQU     40010A         USER OPTION BYTE
040.011                 149X .CTLFLG  EQU     40011A         PANEL CONTROL BYTE
040.013                 150X .ALEDS   EQU     40013A         ABUSS LEDS
040.021                 151X .DLEDS   EQU     40021A         DBUSS LEDS
040.024                 152X .ABUSS   EQU     40024A         ABUSS REGISTER
040.027                 153X .CRCSUM  EQU     40027A         CRCSUM WORD
040.031                 154X .TPERRX  EQU     40031A         TAPE ERROR EXIT VECTOR
040.033                 155X .TICCNT  EQU     40033A         CLOCK TICK COUNTER
040.035                 156X .REGPTR  EQU     40035A         REGISTER POINTER
040.037                 157X .UIVEC   EQU     40037A         USER INTERRUPT VECTORS
040.064                 158X .NMIRET  EQU     40064A         H88/H89 NMI Return Address      /80.07.sc/
040.066                 159X .CTL2FL  EQU     40066A         OP2.CTL Control Byte            /80.07.sc/
000.000                 160           XTEXT   BOODEF



                        162X **       BOODEF - SPECIAL BOOT-HDOS INTERFACE DEFINTIONS.        /80.05.sc/
                        163X
051.000                 164X SB.ORG   EQU     51000A         ORG FOR LOAD OF INITIAL HDOS.SAV
014.000                 165X SB.OVMX  EQU     14000A         SIZE OF HOLD AREA FOR SWAPPED USER CODE
                        166X *                               (=MAX SIZE OF HDOSOVL.SYS)
                        167X
042.200                 168X          ORG     42200A
                        169X
042.200                 170X SB.BOO   DS      3              Jump to Boot routine
042.203                 171X SB.VER   DS      1              Version of INIT that built disk
042.204                 172X SB.FLG   DS      1              Boot Flags
000.001                 173X BFLG.A   EQU     00000001B      Auto-Boot:  1 => Boot
042.205                 174X SB.BAU   DS      2              Baud Rate Divisor (0=>ignore)
042.207                 175X SB.DAT   DS      2              Default Date
000.027                 176X          ERRMI   SB.BOO+32-*
042.211                 177X          DS      SB.BOO+32-*    Reserved
042.240                 178X SB.BPE   EQU     *              End of BOOT-parameters
                        179X
042.240                 180X SB.DRV   DS      SB.BOO+512-*   Primary Boot
                        181X
044.200                 182X SB.SDB   EQU     *              Secondary Boot
044.200                 183           XTEXT   ECDEF
```

```
                          185X **       ERROR CODE DEFINITIONS.
                          186X
  000.000                 187X          ORG     0
  000.000                 188X          DS      1                NO ERROR #0
  000.001                 189X EC.EOF   DS      1                END OF FILE
  000.002                 190X EC.EOM   DS      1                END OF MEDIA
  000.003                 191X EC.ILC   DS      1                ILLEGAL SYSCALL CODE
  000.004                 192X EC.CNA   DS      1                CHANNEL NOT AVAILABLE
  000.005                 193X EC.DNS   DS      1                DEVICE NOT SUITABLE
  000.006                 194X EC.IDN   DS      1                ILLEGAL DEVICE NAME
  000.007                 195X EC.IFN   DS      1                ILLEGAL FILE NAME
  000.010                 196X EC.NRD   DS      1                NO ROOM FOR DEVICE DRIVER
  000.011                 197X EC.FNO   DS      1                CHANNEL NOT OPEN
  000.012                 198X EC.ILR   DS      1                ILLEGAL REQUEST
  000.013                 199X EC.FUC   DS      1                FILE USAGE CONFLICT
  000.014                 200X EC.FNF   DS      1                FILE NAME NOT FOUND
  000.015                 201X EC.UND   DS      1                UNKNOWN DEVICE
  000.016                 202X EC.ICN   DS      1                ILLEGAL CHANNEL NUMBER
  000.017                 203X EC.DIF   DS      1                DIRECTORY FULL
  000.020                 204X EC.IFC   DS      1                ILLEGAL FILE CONTENTS
  000.021                 205X EC.NEM   DS      1                NOT ENOUGH MEMORY
  000.022                 206X EC.RF    DS      1                READ FAILURE
  000.023                 207X EC.WF    DS      1                WRITE FAILURE
  000.024                 208X EC.WPV   DS      1                WRITE PROTECTION VIOLATION
  000.025                 209X EC.WP    DS      1                DISK WRITE PROTECTED
  000.026                 210X EC.FAP   DS      1                FILE ALREADY PRESENT
  000.027                 211X EC.DDA   DS      1                DEVICE DRIVER ABORT
  000.030                 212X EC.FL    DS      1                FILE LOCKED
  000.031                 213X EC.FAO   DS      1                FILE ALREADY OPEN
  000.032                 214X EC.IS    DS      1                ILLEGAL SWITCH
  000.033                 215X EC.UUN   DS      1                UNKNOWN UNIT NUMBER
  000.034                 216X EC.FNR   DS      1                FILE NAME REQUIRED
  000.035                 217X EC.DIW   DS      1                DEVICE IS NOT WRITABLE (OR WRITE LOCKED)
  000.036                 218X EC.UNA   DS      1                UNIT NOT AVAILABLE
  000.037                 219X EC.ILV   DS      1                ILLEGAL VALUE
  000.040                 220X EC.ILO   DS      1                ILLEGAL OPTION
  000.041                 221X EC.VPM   DS      1                VOLUME PRESENTLY MOUNTED ON DEVICE
  000.042                 222X EC.NVM   DS      1                NO VOLUME PRESENTLY MOUNTED
  000.043                 223X EC.FOD   DS      1                FILE OPEN ON DEVICE
  000.044                 224X EC.NPM   DS      1                NO PROVISIONS MADE FOR REMOUNTING MORE DISKS
  000.045                 225X EC.DNI   DS      1                DISK NOT INITIALIZED
  000.046                 226X EC.DNR   DS      1                DISK IS NOT READABLE
  000.047                 227X EC.DSC   DS      1                DISK STRUCTURE IS CORRUPT
  000.050                 228X EC.NCV   DS      1                NOT CORRECT VERSION OF HDOS
  000.051                 229X EC.NOS   DS      1                NO OPERATING SYSTEM MOUNTED
  000.052                 230X EC.IOI   DS      1                ILLEGAL OVERLAY INDEX
  000.053                 231X EC.OTL   DS      1                OVERLAY TO LARGE
                          232  *        XTEXT   H17DEF
  000.054                 233           XTEXT   U8251
```

```
                        236X **       8251 USART BIT DEFINITIONS.
                        237X *
                        238X
                        239X **       PORT ADDRESSES
                        240X
   000.000              241X UDR      EQU     0               DATA REGISTER IS EVEN
   000.001              242X USR      EQU     1               STATUS REGISTER IS NEXT
                        243X
   000.372              244X SC.UART EQU      372Q            CONSOLE USART ADDRESS (IFF 8251)
                        245X
                        246X
                        247X **       MODE INSTRUCTION CONTROL BITS.
                        248X
   000.100              249X UMI.1B  EQU      01000000B       1 STOP BIT
   000.200              250X UMI.HB  EQU      10000000B       1 1/2 STOP BITS
   000.300              251X UMI.2B  EQU      11000000B       2 STOP BITS
   000.040              252X UMI.PE  EQU      00100000B       EVEN PARITY
   000.020              253X UMI.PA  EQU      00010000B       USE PARITY
   000.000              254X UMI.L5  EQU      00000000B       5 BIT CHARACTERS
   000.004              255X UMI.L6  EQU      00000100B       6 BIT CHARACTERS
   000.010              256X UMI.L7  EQU      00001000B       7 BIT CHARACTERS
   000.014              257X UMI.L8  EQU      00001100B       8 BIT CHARACTERS
   000.001              258X UMI.1X  EQU      00000001B       CLOCK X 1
   000.002              259X UMI.16X EQU      00000010B       CLOCK X 16
   000.003              260X UMI.64X EQU      00000011B       CLOCK X 64
                        261X
                        262X **       COMMAND INSTRUCTION BITS.
                        263X
   000.100              264X UCI.IR  EQU      01000000B       INTERNAL RESET
   000.040              265X UCI.RO  EQU      00100000B       READER-ON CONTROL FLAG
   000.020              266X UCI.ER  EQU      00010000B       ERROR RESET
   000.004              267X UCI.RE  EQU      00000100B       RECEIVE ENABLE
   000.002              268X UCI.IE  EQU      00000010B       ENABLE INTERRUPTS FLAG
   000.001              269X UCI.TE  EQU      00000001B       TRANSMIT ENABLE
                        270X
                        271X **       STATUS READ COMMAND BITS.
                        272X
   000.100              273X USR.BD  EQU      01000000B       Break Detect                    /80.08.sc/
   000.040              274X USR.FE  EQU      00100000B       FRAMING ERROR
   000.020              275X USR.OE  EQU      00010000B       OVERRUN ERROR
   000.010              276X USR.PE  EQU      00001000B       PARITY ERROR
   000.004              277X USR.TXE EQU      00000100B       TRANSMITTER EMPTY
   000.002              278X USR.RXR EQU      00000010B       RECEIVER READY
   000.001              279X USR.TXR EQU      00000001B       TRANSMITTER READY
   000.054              280           XTEXT   U8250


                        282X **       8250 UART CONTROL AND BIT DEFINITIONS.
                        283X
   000.350              284X SC.ACE  EQU      350Q            SYSTEM CONSOLE PORT IF 8250 ACE
   000.156              285X AC.DLY  EQU      110             220 MIL. SEC. DELAY FOR 8250
                        286X
   000.000              287X UR.RBR  EQU      0               RECEIVER BUFFER REGISTER (READ ONLY)
                        288X
```

```
     000.000                     289X UR.THR   EQU     0              TRANSMITTER HOLDING REGISTER (WRITE ONLY)
                                 290X
     000.000                     291X UR.DLL   EQU     0              DIVISOR LATCH (LEAST SIGNIFICANT)
                                 292X
     000.001                     293X UR.DLM   EQU     1              DIVISOR LATCH (MOST SIGNIFICANT)
                                 294X
     000.001                     295X UR.IER   EQU     1              INTERRUPT ENABLE REGISTER
     000.001                     296X UC.EDA   EQU     00000001B       ENABLE RECEIVED DATA AVAILABLE INTERRUPT
     000.002                     297X UC.TRE   EQU     00000010B       ENABLE TRANSMIT HOLD REGISTER EMPTY INTERRUPT
     000.004                     298X UC.RSI   EQU     00000100B       ENABLE RECEIVE STATUS INTERRUPT
     000.010                     299X UC.MSI   EQU     00001000B       ENABLE MODEM STATUS INTERRUPT
                                 300X
     000.002                     301X UR.IIR   EQU     2              INTERRUPT IDENTIFICATION REGISTER
     000.001                     302X UC.IIP   EQU     00000001B       INVERTED INTERRUPT PENDING (0 MEANS PENDING)
     000.006                     303X UC.IID   EQU     00000110B       INTERRUPT ID
                                 304X
     000.003                     305X UR.LCR   EQU     3              LINE CONTROL REGISTER
     000.000                     306X UC.5BW   EQU     00000000B       5 BIT WORDS
     000.001                     307X UC.6BW   EQU     00000001B       6 BIT WORDS
     000.002                     308X UC.7BW   EQU     00000010B       7 BIT WORDS
     000.003                     309X UC.8BW   EQU     00000011B       8 BIT WORDS
     000.004                     310X UC.2SB   EQU     00000100B       TWO STOP BITS SELECTED
     000.010                     311X UC.PEN   EQU     00001000B       PARITY COMPUTATION ENABLED
     000.020                     312X UC.EPS   EQU     00010000B       EVEN PARITY SELECT
     000.040                     313X UC.SKP   EQU     00100000B       STICK PARITY
     000.100                     314X UC.SB    EQU     01000000B       SET BREAK
     000.200                     315X UC.DLA   EQU     10000000B       DIVISOR LATCH ACCESS
                                 316X
     000.004                     317X UR.MCR   EQU     4              MODEM CONTROL REGISTER
     000.001                     318X UC.DTR   EQU     00000001B       DATA TERMINAL READY
     000.002                     319X UC.RTS   EQU     00000010B       REQUEST TO SEND
     000.004                     320X UC.OU1   EQU     00000100B       OUT 1
     000.010                     321X UC.OU2   EQU     00001000B       OUT 2
     000.020                     322X UC.LOO   EQU     00010000B       LOOP
                                 323X
     000.005                     324X UR.LSR   EQU     5              LINE STATUS REGISTER
     000.001                     325X UC.DR    EQU     00000001B       DATA READY
     000.002                     326X UC.OR    EQU     00000010B       OVERRUN
     000.004                     327X UC.PE    EQU     00000100B       PARITY ERROR
     000.010                     328X UC.FE    EQU     00001000B       FRAMING ERROR
     000.020                     329X UC.BT    EQU     00010000B       BREAK INTERRUPT
     000.040                     330X UC.THE   EQU     00100000B       TRANSMITTER HOLDING REGISTER EMPTY
     000.100                     331X UC.TSE   EQU     01000000B       TRANSMITTER SHIFT REGISTER EMPTY
                                 332X
     000.006                     333X UR.MSR   EQU     6              MODEM STATUS REGISTER
     000.001                     334X UC.DCS   EQU     00000001B       DELTA CLEAR TO SEND
     000.002                     335X UC.DDR   EQU     00000010B       DELTA DATA SET READY
     000.004                     336X UC.TER   EQU     00000100B       TRAILING EDGE OF RING
     000.010                     337X UC.DRL   EQU     00001000B       DELTA RECEIVE LINE SIGNAL DETECT
     000.020                     338X UC.CTS   EQU     00010000B       CLEAR TO SEND
     000.040                     339X UC.DSR   EQU     00100000B       DATA SET READY
     000.100                     340X UC.RI    EQU     01000000B       RING INDICATOR
     000.200                     341X UC.RLS   EQU     10000000B       RECEIVED LINE SIGNAL DETECT
     000.054                     342           XTEXT   DDDEF          DEVICE DRIVER CONSTANTS
```

```
                        344X **       DEVICE DRIVER COMMUNICATION FLAGS.
                        345X *
                        346X
         000.000        347X          ORG     0
                        348X
         000.000        349X DC.REA   DS      1               READ
         000.001        350X DC.WRI   DS      1               WRITE
         000.002        351X DC.RER   DS      1               READ REGARDLESS
         000.003        352X DC.OPR   DS      1               OPEN FOR READ
         000.004        353X DC.OPW   DS      1               OPEN FOR WRITE
         000.005        354X DC.OPU   DS      1               OPEN FOR UPDATE
         000.006        355X DC.CLO   DS      1               CLOSE
         000.007        356X DC.ABT   DS      1               ABORT
         000.010        357X DC.MOU   DS      1               MOUNT DEVICE
         000.011        358X DC.LOD   DS      1               LOAD DEVICE DRIVER
         000.012        359X DC.RDY   DS      1               Device Ready                /80.04.GC/
         000.013        360X DC.MAX   DS      1               MAXIMUM ENTRY INDEX
         000.014        361           XTEXT   FILDEF


                        363X **       FILDEF - FILE TYPE DEFINITIONS.
                        364X *
                        365X *        DB      377Q,FT.XXX
                        366X
                        367X
         000.000        368X FT.ABS   EQU     0               ABSOLUTE BINARY
         000.001        369X FT.PIC   EQU     1               POSITION INDEPENDANT CODE
         000.002        370X FT.REL   EQU     2               RELOCATABLE CODE
         000.003        371X FT.BAC   EQU     3               COMPILED BASIC CODE
         000.014        372           XTEXT   ABSDEF


                        374X **       ABS FORMAT EQUIVALENCES.
                        375X
         000.000        376X          ORG     0
                        377X
         000.000        378X ABS.ID   DS      1               377Q = BINARY FILE FLAG
         000.001        379X          DS      1               FILE TYPE (FT.ABS)
         000.002        380X ABS.LDA  DS      2               LOAD ADDRESS
         000.004        381X ABS.LEN  DS      2               LENGTH OF ENTIRE RECORD
         000.006        382X ABS.ENT  DS      2               ENTRY POINT
                        383X
         000.010        384X ABS.COD  DS      0               CODE STARTS HERE
         000.010        385           XTEXT   PICDEF
```

```
                     387X **      PIC FORMAT EQUIVALENCES.
                     388X
000.000              389X         ORG     0
                     390X
000.000              391X PIC.ID  DS      1                 377Q = BINARY FILE FLAG
000.001              392X         DS      1                 FILE TYPE (FT.PIC)
000.002              393X PIC.LEN DS      2                 LENGTH OF ENTIRE RECORD
000.004              394X PIC.PTR DS      2                 INDEX OF START OF PIC TABLE
                     395X
000.006              396X PIC.COD DS      0                 CODE STARTS HERE
000.006              397          XTEXT   DIRDEF


                     399X **      DIRECTORY ENTRY FORMAT.
                     400X
000.000              401X         ORG     0
                     402X
                     403X
000.377              404X DF.EMP  EQU     377Q              FLAGS ENTRY EMPTY
000.376              405X DF.CLR  EQU     376Q              FLAGS ENTRY EMPTY, REST OF DIR ALSO CLEAR
                     406X
000.000              407X DIR.NAM DS      8                 NAME
000.010              408X DIR.EXT DS      3                 EXTENSION
000.013              409X DIR.PRO DS      1                 PROJECT
000.014              410X DIR.VER DS      1                 VERSION
000.015              411X DIRIDL  EQU     *                 FILE IDENTIFICATION LENGTH
                     412X
000.015              413X DIR.CLU DS      1                 CLUSTER FACTOR
000.016              414X DIR.FLG DS      1                 FLAGS
000.017              415X         DS      1                 RESERVED
000.020              416X DIR.FGN DS      1                 FIRST GROUP NUMBER
000.021              417X DIR.LGN DS      1                 LAST GROUP NUMBER
000.022              418X DIR.LSI DS      1                 LAST SECTOR INDEX (IN LAST GROUP)
000.023              419X DIR.CRD DS      2                 CREATION DATE
000.025              420X DIR.ALD DS      2                 LAST ALTERATION DATE
                     421X
000.027              422X DIRELEN EQU     *                 DIRECTORY ENTRY LENGTH
000.027              423          XTEXT   DISDEF


                     425X **      DIRECTORY BLOCK FORMAT.
                     426X
000.000              427X         ORG     0
                     428X
000.000              429X DIS.ENT EQU     *                 FIRST ENTRY ADDRESS
000.000              430X         DS      22*DIRELEN        22 DIRECTORY ENTRYS PER BLOCK
001.372              431X         DS      1                 0 BYTE = END OF ENTRYS IN THIS BLOCK
                     432X
001.373              433X         ORG     512-5             AT END OF BLOCK
001.373              434X DIS.ENL DS      1                 LENGTH OF EACH ENTRY (=DIRELEN)
001.374              435X DIS.SEC DS      2                 BLOCK # OF THIS BLOCK,
001.376              436X DIS.LNK DS      2                 BLOCK # OF NEXT BLOCK, =0 IF THIS IS LAST
002.000              437          XTEXT   DEVDEF
```

```
                        439X **       DEVICE TABLE ENTRYS.
                        440X
   000.000              441X          ORG     0
                        442X
   000.000              443X DEV.NAM DS       2             DEVICE NAME
   000.000              444X DV.EL   EQU      00000000B     END OF DEVICE LIST FLAG
   000.001              445X DV.NU   EQU      00000001B     DEVICE ENTRY NOT IN USE
                        446X
   000.002              447X DEV.RES DS       1             DRIVER RESIDENSE CODE
   000.001              448X DR.IM   EQU      00000001B     DRIVER IN MEMORY
   000.002              449X DR.PR   EQU      00000010B     DRIVER PERMINANTLY RESIDENT
                        450X
   000.003              451X DEV.JMP DS       1             JMP TO PROCESSOR
   000.004              452X DEV.DDA DS       2             DRIVER ADDRESS
   000.006              453X DEV.FLG DS       1             FLAG BYTE
   000.001              454X DT.DD   EQU      00000001B     DIRECTORY DEVICE
   000.002              455X DT.CR   EQU      00000010B     CAPABLE OF READ OPERATION
   000.004              456X DT.CW   EQU      00000100B     CAPABLE OF WRITE OPERATION
   000.010              457X DT.RN   EQU      00001000B     Capable of random access        /80.02.sc/
   000.020              458X DT.CH   EQU      00010000B     Capable of Character mode       /80.02.sc/
                        459X
   000.007              460X DEV.MUM DS       1             MOUNTED UNIT MASK
   000.010              461X DEV.MNU DS       1             MAXIMUM NUMBER OF UNITS
   000.011              462X DEV.UNT DS       2             ADDRESS OF UNIT SPECIFIC DATA TABLE
                        463X
   000.013              464X DEV.DVL DS       2             DRIVER BYTE LENGTH
   000.015              465X DEV.DVG DS       1             DRIVER ROUTINE GROUP ADDRESS
                        466X
   000.016              467X DEVELEN EQU      *             DEVICE TABLE ENTRY LENGTH



                        469X **       UNIT SPECIFIC DEVICE DATA TABLE ENTRIES
                        470X
   000.000              471X          ORG     0
                        472X
   000.000              473X UNT.FLG DS       1             UNIT SPECIFIC  *DEV.FLG*
   000.001              474X UNT.SPG DS       1             Sectors Per Group               /80.04.GC/
   000.002              475X UNT.GRT DS       2             ADDRESS OF GROUP RESERVATION TABLE (IF DT.DD)
   000.004              476X UNT.GTS DS       2             GRT SECTOR NUMBER
   000.006              477X UNT.DIS DS       2             DIRECTORY FIRST SECTOR NUMBER
                        478X
   000.010              479X UNT.SIZ EQU      *             SIZE OF UNIT SPECIFIC DATA TABLE PER UNIT
   000.010              480          XTEXT   DDFDEF



                        482X **       DIRECTORY DEVICE FORMAT DEFINITION.                     /80.09.sc/
                        483X *
                        484X *        Modified:       Sep-80
                        485X *                         No longer require 2 sectors per group
                        486X *                         Reserved Group Table dynamically allocated
                        487X *
                        488X
   000.000              489X          ORG     0
```

```
                              490X
    000.000                   491X DDF.BOO DS      9                2K BOOT PROGRAM
    000.011                   492X DDF.BOL EQU     *                LENGTH OF BOOT
    000.011                   493X DDF.LAB DS      1                LABEL SECTOR
    000.012                   494X DDF.USR DS      0                BEGINNING OF OPEN SPACE
    000.012                   495          XTEXT   DVDDEF


                              497X **      DEVICE DRIVER EQUIVALENCES.
                              498X
    000.307                   499X DVDFLV  EQU     307Q             DEVICE DRIVER FLAG VALUE
                              500X
    000.006                   501X         ORG     PIC.COD          STARTS AT PIC CODE AREA
                              502X
    000.006                   503X DVD.DVD DS      1                MUST BE DVDFLV, FLAGS TO HDOS AS DRIVER
    000.007                   504X DVD.CAP DS      1                DEVICE CAPABILITY FLAG
    000.010                   505X DVD.MUM DS      1                MOUNTED UNIT MASK
    000.011                   506X DVD.MNU DS      1                MAXIMUM NUMBER OF UNITS
    000.012                   507X DVD.UFL DS      8                UNIT SUB-CAPABILITY FLAGS FOR UNITS 0-7
    000.022                   508X DVD.SET DS      1                = DVDFLV IFF DRIVER WILL TAKE SET OPTIONS
    000.023                   509X DVD.INF DS      2                Pointer to Init Code            /80.07.sc/
    000.025                   510X         DS      22               RESERVED, MUST BE 0             /80.07.sc/
    000.053                   511X DVD.STE EQU     *                ENTRY FOR 'SET' INVOCATION
                              512X
    002.000                   513X DVD.ENT EQU     2000A            DRIVER ENTRY POINT (MUST BE MULT OF 256)
    000.053                   514          XTEXT   FBDEF


                              516X **      FILE BLOCK DEFINITIONS.
                              517X
    000.000                   518X         ORG     0
    000.000                   519X FB.CHA  DS      1                CHANNEL NUMBER
    000.001                   520X FB.FLG  DS      1                FLAGS
    000.002                   521X FB.FWA  DS      2                BUFFER FWA
    000.004                   522X FB.PTR  DS      2                BUFFER POINTER
    000.006                   523X FB.LIM  DS      2                LIMIT OF DATA IN BUFFER (READ OPERATIONS)
    000.010                   524X FB.LWA  DS      2                LWA OF BUFFER
    000.012                   525X FB.NAM  DS      4+8+4+1          NAME OF FILE
    000.021                   526X FB.NAML EQU     *-FB.NAM
    000.033                   527X FBENL   EQU     *                ENTRY LENGTH
    000.033                   528          XTEXT   IOCDEF


                              530X **      I/O CHANNEL DEFINITIONS.
                              531X
    000.000                   532X         ORG     0
                              533X
    000.000                   534X IOC.LNK DS      2                ADDRESS OF NEXT CHANNEL, =0 IF LAST
    000.002                   535X IOC.DDA DS      2                THREAD JUMP TO DEVICE DRIVER (VIA DEV TABLE)
                              536X
    000.004                   537X IOC.FLG DS      1                FILE TYPE FLAGS
```

```
     000.001             538X FT.DD    EQU    00000001B     =1 IF DIRECTORY DEVICE
     000.002             539X FT.OR    EQU    00000010B     =1 IF OPEN FOR READ
     000.004             540X FT.OW    EQU    00000100B     =1 IF OPEN FOR WRITE
     000.010             541X FT.OU    EQU    00001000B     =1 IF OPEN FOR UPDATE
     000.020             542X FT.OC    EQU    00010000B     =1 IF OPEN FOR CHARACTER MODE   /80.02.GC/
     000.003             543X IOC.SQL  EQU    *-IOC.DDA     LENGTH OF INFO FOR SEQUENTIAL FILE (FROM IOC)
                         544X
     000.005             545X IOC.GRT  DS     2             ADDRESS OF GROUP RESERVATION TABLE
     000.007             546X IOC.SPG  DS     1             SECTORS PER GROUP, THIS DEVICE
     000.010             547X IOC.CGN  DS     1             CURRENT GROUP NUMBER
     000.011             548X IOC.CSI  DS     1             CURRENT SECTOR INDEX (IN CURRENT GROUP)
     000.012             549X IOC.LGN  DS     1             LAST GROUP NUMBER
     000.013             550X IOC.LSI  DS     1             LAST SECTOR INDEX (IN LAST GROUP)
     000.010             551X IOC.DRL  EQU    *-IOC.FLG     LENGTH OF INFO NORMALLY COPIED BACK TO
                         552X *                             THE CHANNEL TABLE
     000.014             553X IOC.DTA  DS     2             DEVICE TABLE ADDRESS FOR THIS DEVICE
     000.016             554X IOC.DES  DS     2             SECTOR NUMBER OF DIRECTORY ENTRY
     000.020             555X IOC.DEV  DS     2             DEVICE CODE
     000.022             556X IOC.UNI  DS     1             UNIT NUMBER (0-9)
     000.021             557X IOC.DIL  EQU    *-IOC.DDA     LENGTH OF INFO FOR DIRECTORY FILE (FROM IOC)
                         558X
     000.023             559X IOC.DIR  DS     DIRELEN       DIRECTORY ENTRY
                         560X
     000.052             561X IOCELEN  EQU    *             IOC ENTRY LENGTH
                         562X
     000.001             563X IOCCTD   EQU    1             INDEX OF USER CHANNEL #0 IN CHANTAB (FIRST = 0)
     000.052             564           XTEXT  LABDEF


                         566X **       DISK LABEL SECTOR FORMATS.
                         567X
     000.000             568X          ORG    0
     000.000             569X LAB.SER  DS     1             SERIAL NUMBER OF VOLUME
     000.001             570X LAB.IND  DS     2             INITIALIZATION DATE
     000.003             571X LAB.DIS  DS     2             SECTOR NUMBER OF 1ST DIRECTORY SECTOR
     000.005             572X LAB.GRT  DS     2             INDEX OF GRT SECTOR
     000.007             573X LAB.SPG  DS     1             SECTORS PER GROUP
                         574X
     000.000             575X LAB.DAT  EQU    0             DATA VOLUME ONLY
     000.001             576X LAB.SYS  EQU    1             SYSTEM VOLUME
     000.002             577X LAB.NOD  EQU    2             => LAB.NOD MEANS VOLUME HAS NO DIRECTORY
                         578X
     000.010             579X LAB.VLT  DS     1             VOLUME TYPE
     000.011             580X LAB.VER  DS     1             VERSION OF INIT17 THAT INITED DISK
                         581X
     000.012             582X LAB.RGT  DS     2             RGT sector number                 /80.06.sc/
                         583X
     000.014             584X LAB.VPR  EQU    *             Volume dependant data             /80.05.sc/
     000.014             585X LAB.SIZ  DS     2              Volume Size (Bytes/256)          /80.05.sc/
     000.016             586X LAB.PSS  DS     2              Physical Sector Size             /80.05.sc/
     000.020             587X LAB.VFL  DS     1              Volume dependant Flags           /80.09.sc/
     000.001             588X VFL.NSD  EQU    00000001B       Number of Sides:  1 => 2        /80.09.sc/
     000.005             589X LAB.VPL  EQU    *-LAB.VPR     Length of volume dependant data /80.05.sc/
                         590X
     000.000             591X          ERRMI  5-LAB.VPL                                       /80.05.sc/
     000.021              592X          DS      5-LAB.VPL       Reserved                        /80.05.sc/
                          593X
     000.021              594X LAB.LAB  DS      60              LABEL
     000.074              595X LAB.LBL  EQU     *-LAB.LAB       LABEL LENGTH
     000.115              596X          DS      2               Reserved for 0 bytes            /80.09.sc/
                          597X
     000.117              598X LAB.AUX  EQU     *               Auxiliary Data                  /80.09.sc/
     000.117              599X LAB.SPT  DS      1                Sectors per Track              /80.09.sc/
     000.001              600X LAB.AXL  EQU     *-LAB.AUX       Length of Aux. Data             /80.09.sc/
     000.120              601           XTEXT   MTRDEF                                          /80.09.GC/



                          603X **       HDOS MONITOR PRIVATE RAM AREA DEFINITIONS.
                          604X
     000.000              605X          ORG     0
     000.000              606X M.SYSM   DS      1               SYSCALL ITTERATION COUNT
     000.001              607X M.SALO   DS      1               STAND-ALONE FLAG
     000.002              608X M.CSLC   DS      1               LINES IN CONSOLE BUFFER
     000.003              609X M.CPRE   DS      1               CONSOLE PREVIOUS CHARACTER
     000.004              610X M.CRUB   DS      1               CONSOLE RUBOUT FLAG
     000.005              611X M.CINT   DS      1               CONSOLE INTERRUPT FLAG
     000.006              612X M.CIN    DS      2               CONSOLE CB IN POINTER
     000.010              613X M.COUT   DS      2               CONSOLE CB OUT POINTER
     000.012              614X M.CFWA   DS      2               CONSOLE CB FWA POINTER
     000.014              615X M.CLWA   DS      2               CONSOLE CB LWA POINTER
     000.016              616X M.CDLY   DS      1               CONSOLE PAD CHARACTER COUNT
     000.017              617X M.CDCA   DS      2               ADDRESS OF CHARACTER BEING PADDED
     000.021              618X M.SUNI   DS      1               System Unit Number              /80.05.sc/
     000.022              619X M.SYDD   DS      2               Address of Raw System Driver    /80.09.sc/
     000.024              620           XTEXT   DIFDEF



                          622X **       DIRECTORY FILE FLAGS.
                          623X
     000.200              624X DIF.SYS  EQU     10000000B               SYSTEM FILE
     000.100              625X DIF.LOC  EQU     01000000B               LOCKED FOR CHANGE
     000.040              626X DIF.WP   EQU     00100000B               WRITE PROTECTED
     000.020              627X DIF.CNT  EQU     00010000B               CONTIGUOUS FILE
                          628X
     000.024              629           XTEXT   NAMDEF



                          631X **       SYSTEM FILE NAME CONVENTIONS
                          632X *
                          633X *        RGT     .SYS            RESERVED GROUP TABLE (1 SECTOR)
                          634X *        GRT     .SYS            GROUP RESERVATION TABLE (1 SECTOR)
                          635X *        DIRECT  .SYS            DIRECTORY
                          636X *        HOS     .SYS            SYSTEM IMAGE PROGRAM FOR SYSTEM
                          637X
     000.024              638           XTEXT   OVLDEF
```

```
                         640X **       OVERLAY TABLE ENTRYS.
                         641X
   000.000               642X          ORG     0
                         643X
   000.000               644X OVL.COD DS       2                     FIRST SECTOR OF OVERLAY CODE
   000.002               645X OVL.SIZ DS       2                     OVERLAY SIZE
   000.004               646X OVL.ENT DS       2                     OVERLAY ENTRY POINT
   000.006               647X OVL.FLB DS       1                     OVERLAY FLAG BYTE
   000.007               648X         DS       1                     DUMMY BYTE TO ROUND TABLE SIZE UP TO 8
   000.010               649X OVL.ENS EQU      *                     OVERLAY ENTRY SIZE
                         650X
                         651X *        OVERLAY INDICES
                         652X
   000.000               653X          ORG     0
                         654X
   000.000               655X OVL0     DS      1
   000.001               656X OVL1     DS      1
   000.002               657           XTEXT   HDSROM


                         659X **       HDOS H17 ROM ENTRY POINTS.
   031.253               660X          ORG     31253A
                         661X *DWRITE EQU      *                 Obsolete                /80.04.sc/
   031.253               662X          DS      31256A-31253A
                         663X *DREAD  EQU      *                 Obsolete                /80.04.sc/
   031.256               664X          DS      31275A-31256A
   031.275               665X S.READ  EQU      *
   031.275               666X          DS      31321A-31266A
   031.330               667X S.WRITE EQU      *
   031.330               668X          DS      31325A-31311A
   031.344               669X ERR.FNO EQU      *
   031.344               670X          DS      31331A-31325A
   031.350               671X ERR.ILR EQU      *
   031.350               672X          DS      31335A-31331A
   031.354               673X CFF      EQU     *
   031.354               674X          DS      31363A-31335A
   032.002               675X DCA      EQU     *
   032.002               676X          DS      32114A-31363A
   032.133               677X FFB      EQU     *
   032.133               678X          DS      32166A-32114A
   032.205               679X FFL      EQU     *
   032.205               680X          DS      32204A-32166A
                         681X *LDD     EQU     *
   032.223               682X          DS      32372A-32204A+1
   033.012               683X LDO      EQU     *
   033.012               684X          DS      33135A-33002A
   033.145               685X PDI      EQU     *
   033.145               686X          DS      33154A-33124A
   033.175               687X REL.     EQU     *
   033.175               688X          DS      33156A-33154A
   033.177               689X REL      EQU     *
   033.177               690X          DS      33212A-33156A
   033.233               691X TFE      EQU     *
   033.233               692X          DS      33232A-33206A
   033.257               693X RUC      EQU     *
```

```
                         694X
      037.132            695X BOOTA   EQU     37132A          Boot Vectors                     /80.06.gc/
      000.130            696X BOOTAL  EQU     00130A          Length of boot vectors           /80.06.gc/
                         697X
      034.031            698X CLOCK   EQU     34031A          Clock vector                     /80.06.GC/
      033.257            699          XTEXT   HOSEQU


                         701X **      HDOS SYSTEM EQUIVALENCES.
                         702X *
                         703X
      024.000            704X S.GRT0  EQU     24000A          SYSTEM AREA FOR  GRT0
      025.000            705X S.GRT1  EQU     25000A          SYSTEM AREA FOR  GRT1
      026.000            706X S.GRT2  EQU     26000A          SYSTEM AREA FOR  GRT2
                         707X
      030.000            708X ROMBOOT EQU     30000A          ROM BOOT ENTRY
                         709X
      040.100            710X         ORG     40100A          FREE SPACE FROM PAM-8
                         711X
      040.100            712X         DS      8               JUMP TO SYSTEM EXIT
      040.110            713X D.CON   DS      16              DISK CONSTANTS
      040.130            714X SYDD    EQU     *               SYSTEM DISK ENTRY POINT
      040.130            715X D.VEC   DS      24*3            SYSTEM ROM ENTRY VECTORS
      040.240            716X D.RAM   DS      31              SYSTEM ROM WORK AREA
      040.277            717X S.VAL   DS      36              SYSTEM VALUES
      040.343            718X S.INT   DS      115             SYSTEM INTERNAL WORK AREAS
      041.126            719X         DS      16
      041.146            720X S.SOVR  DS      2               STACK OVERFLOW WARNING
      041.150            721X         DS      42200A-*        SYSTEM STACK
      001.032            722X STACKL  EQU     *-S.SOVR        STACK SIZE
                         723X
      042.200            724X STACK   EQU     *               LWA+1 SYSTEM STACK
      042.200            725X USERFWA EQU     *               USER FWA
      042.200            726          XTEXT   HOSDEF


                         728X **      HOSDEF - DEFINE HOS PARAMETER.
                         729X *
                         730X
                         731X
      000.040            732X VERS    EQU     2*16+0          VERSION 2.0
                         733X
      000.377            734X SYSCALL EQU     377Q            SYSCALL INSTRUCTION
                         735X
                         736X
      000.000            737X         ORG     0
                         738X
                         739X *       RESIDENT FUNCTIONS
                         740X
      000.000            741X .EXIT   DS      1               EXIT (MUST BE FIRST)
      000.001            742X .SCIN   DS      1               SCIN
      000.002            743X .SCOUT  DS      1               SCOUT
      000.003            744X .PRINT  DS      1               PRINT
```

```
    000.004              745X .READ   DS      1               READ
    000.005              746X .WRITE  DS      1               WRITE
    000.006              747X .CONSL  DS      1               SET/CLEAR CONSOLE OPTIONS
    000.007              748X .CLRCO  DS      1               CLEAR CONSOLE BUFFER
    000.010              749X .LOADO  DS      1               LOAD AN OVERLAY
    000.011              750X .VERS   DS      1               RETURN HDOS VERSION NUMBER
    000.012              751X .SYSRES DS      1               PRECEDING FUNCTIONS ARE RESIDENT
                         752X
                         753X
                         754X *       *HDOSOVL0.SYS*  FUNCTIONS
                         755X
    000.040              756X         ORG     40A
                         757X
    000.040              758X .LINK   DS      1               LINK  (MUST BE FIRST)
    000.041              759X .CTLC   DS      1               CTL-C
    000.042              760X .OPENR  DS      1               OPENR
    000.043              761X .OPENW  DS      1               OPENW
    000.044              762X .OPENU  DS      1               OPENU
    000.045              763X .OPENC  DS      1               OPENC
    000.046              764X .CLOSE  DS      1               CLOSE
    000.047              765X .POSIT  DS      1               POSITION
    000.050              766X .DELET  DS      1               DELETE
    000.051              767X .RENAM  DS      1               RENAME
    000.052              768X .SETTP  DS      1               SETTOP
    000.053              769X .DECODE DS      1               NAME DECODE
    000.054              770X .NAME   DS      1               GET FILE NAME FROM CHANNEL
    000.055              771X .CLEAR  DS      1               CLEAR CHAN
    000.056              772X .CLEARA DS      1               CLEAR ALL CHANS
    000.057              773X .ERROR  DS      1               LOOKUP ERROR
    000.060              774X .CHFLG  DS      1               CHANGE FLAGS
    000.061              775X .DISMT  DS      1               FLAG SYSTEM DISK DISMOUNTED
    000.062              776X .LOADD  DS      1               LOAD DEVICE DRIVER
    000.063              777X .OPEN   DS      1               Parametrized Open
                         778X
                         779X
                         780X *       *HDOSOVL1.SYS*  FUNCTIONS
                         781X
    000.200              782X         ORG     200Q
                         783X
    000.200              784X .MOUNT  DS      1               MOUNT  (MUST BE FIRST)
    000.201              785X .DMOUN  DS      1               DISMOUNT
    000.202              786X .MONMS  DS      1               MOUNT/NO MESSAGE
    000.203              787X .DMNMS  DS      1               DISMOUNT/NO MESSAGE
    000.204              788X .RESET  DS      1               RESET = DISMOUNT/MOUNT OF UNIT
    000.205              789X .CLEAN  DS      1               Clean device
    000.206              790X .DAD    DS      1               Dismount All Disks            /80.08.sc/
    000.207              791          XTEXT   EDRAM
```

```
                      793X **       EDRAM - DISK RAM WORKAREA DEFINITION.
                      794X *
                      795X *        ZEROED UPON BOOTING UP.
                      796X *
                      797X *        HOSEQU MUST BE CHANGED WHEN THIS DECK IS CHANGED.
                      798X
                      799X
   040.240            800X          ORG     D.RAM
                      801X
   040.240            802X D.TT     DS      1                TARGET TRACK (CURRENT OPERATION)
   040.241            803X D.TS     DS      1                TARGET SECTOR (CURRENT OPERATION)
                      804X
   040.242            805X D.DVCTL  DS      1                DEVICE CONTROL BYTE
                      806X
   040.243            807X D.DLYMO  DS      1                MOTOR ON DELAY COUNT
   040.244            808X D.DLYHS  DS      1                HEAD SETTLE DELAY COUNTER
                      809X
   040.245            810X D.TRKPT  DS      2                ADDRESS IN D.DRVTB FOR TRACK NUMBER
   040.247            811X D.VOLPT  DS      2                ADDRESS IN D.DRVTB FOR VOLUME NUMBER
                      812X
   040.251            813X D.DRVTB  DS      2*4              TRACK NUMBER AND VOLUME NUMBER FOR 4 DRIVES
                      814X
   040.261            815X D.HECNT  DS      1                HARD ERROR COUNT
   040.262            816X D.SECNT  DS      2                SOFT ERROR COUNT
   040.264            817X D.OECNT  DS      1                OPERATION ERROR COUNT
                      818X
                      819X *        GLOBAL DISK ERROR COUNTERS
                      820X
   040.265            821X D.ERR    DS      0                BEGINNING OF ERROR BLOCK
   040.265            822X D.E.MDS  DS      1                MISSING DATA SYNC
   040.266            823X D.E.HSY  DS      1                MISSING HEADER SYNC
   040.267            824X D.E.CHK  DS      1                DATA CHECKSUM
   040.270            825X D.E.HCK  DS      1                HEADER CHECKSUM
   040.271            826X D.E.VOL  DS      1                WRONG VOLUME NUMBER
   040.272            827X D.E.TRK  DS      1                BAD TRACK SEEK
   040.273            828X D.ERRL   DS      0                LIMIT OF ERROR COUNTERS
                      829X
                      830X *        I/O OPERATION COUNTS
                      831X
   040.273            832X D.OPR    DS      2
   040.275            833X D.OPW    DS      2
                      834X
   000.037            835X D.RAML   EQU     *-D.RAM
   040.277            836           XTEXT   EDVEC



                      838X **       JMP VECTORS FOR ROM CODE
                      839X *
                      840X *        SEE DISK ROM FOR ADDRESSES
                      841X *
                      842X *        HOSEQU MUST BE ALTERED WHEN THIS TABLE IS ALTERED.
                      843X
   040.130            844X          ORG     D.VEC
                      845X
```

```
        040.130                  846X D.SYDD   DS       3                  JMP   R.SYDD (MUST BE FIRST)
        040.133                  847X D.MOUNT  DS       3                  JMP   R.MOUNT
        040.136                  848X D.XOK    DS       3                  JMP   R.XOK
        040.141                  849X D.ABORT  DS       3                  JMP   R.ABORT
        040.144                  850X D.XIT    DS       3                  JMP   R.XIT
        040.147                  851X D.READ   DS       3                  JMP   R.READ
        040.152                  852X D.READR  DS       3                  JMP   R.READR
        040.155                  853X D.WRITE  DS       3                  JMP   R.WRITE
        040.160                  854X D.CDE    DS       3                  JMP   R.CDE
        040.163                  855X D.DTS    DS       3                  JMP   R.DTS
        040.166                  856X D.SDT    DS       3                  JMP   R.SDT
        040.171                  857X D.MAI    DS       3                  JMP   R.MAI
        040.174                  858X D.MAO    DS       3                  JMP   R.MAO
        040.177                  859X D.LPS    DS       3                  JMP   R.LPS
        040.202                  860X D.RDB    DS       3                  JMP   R.RDB
        040.205                  861X D.SDP    DS       3                  JMP   R.SDP
        040.210                  862X D.STS    DS       3                  JMP   R.STS
        040.213                  863X D.STZ    DS       3                  JMP   R.STZ
        040.216                  864X D.UDLY   DS       3                  JMP   R.UDLY
        040.221                  865X D.WSC    DS       3                  JMP   R.WSC
        040.224                  866X D.WSP    DS       3                  JMP   R.WSP
        040.227                  867X D.WNB    DS       3                  JMP   R.WNB
        040.232                  868X D.ERRT   DS       3                  JMP   R.ERRT
        040.235                  869X D.DLY    DS       3                  JMP   R.DLY
        040.240                  870           XTEXT    ESVAL



                                 872X **       S.VAL - SYSTEM VALUE DEFINTIONS.
                                 873X *
                                 874X *        THESE VALUES ARE SET AND MAINTAINED BY THE SYSTEM.
                                 875X *
                                 876X *        THE DECK HOSEQU MUST BE MODIFIED WHEN THIS IS MODIFIED.
                                 877X
                                 878X
        040.277                  879X          ORG      S.VAL
                                 880X
        040.277                  881X S.DATE   DS       9               SYSTEM DATE (IN ASCII)
        040.310                  882X S.DATC   DS       2               CODED DATE
        040.312                  883X S.TIME   DS       4               TIME FROM MIDNIGHT (IN TICS)
        040.316                  884X S.HIMEM  DS       2               HARDWARE HIGH MEMORY ADRESS+1
                                 885X
        040.320                  886X S.SYSM   DS       2               FWA RESIDENT SYSTEM
                                 887X
        040.322                  888X S.USRM   DS       2               LWA USER MEMORY
                                 889X
        040.324                  890X S.OMAX   DS       2               MAX OVERLAY SIZE FOR SYSTEM
                                 891X
                                 892X
                                 893X **       THE FOLLOWING FIVE CELLS SHOULD BE MODIFIED/READ ONLY VIA THE .CONSL SYSCALL
                                 894X
        000.200                  895X CSL.ECH  EQU      10000000B       SUPPRESS ECHO
        000.004                  896X CSL.RAW  EQU      00000100B       Raw Mode I/O                  /80.09.sc/
        000.002                  897X CSL.WRP  EQU      00000010B       WRAP LINES AT WIDTH
        000.001                  898X CSL.CHR  EQU      00000001B       OPERATE IN CHARACTER MODE
```

```
                     899X
 000.000             900X I.CSLMD EQU     0              S.CSLMD IS FIRST BYTE
 040.326             901X S.CSLMD DS      1              CONSOLE MODE
                     902X
 000.200             903X CTP.BKS EQU     10000000B      TERMINAL PROCESSES BACKSPACES
 000.100             904X CTP.FF  EQU     01000000B      Terminal Processes Form-Feed    /80.09.sc/
 000.040             905X CTP.MLI EQU     00100000B      MAP LOWER CASE TO UPPER ON INPUT
 000.020             906X CTP.MLO EQU     00010000B      MAP LOWER CASE TO UPPER ON OUTPUT
 000.010             907X CTP.2SB EQU     00001000B      TERMINAL NEEDS TWO STOP BITS
 000.002             908X CTP.BKM EQU     00000010B      MAP BKSP (UPON INPUT) TO RUBOUT
 000.001             909X CTP.TAB EQU     00000001B      TERMINAL SUPPORTS TAB CHARACTERS
                     910X
 000.001             911X I.CONTY EQU     1              S.CONTY IS 2ND BYTE
 000.000             912X         ERRNZ   *-S.CSLMD-I.CONTY
 040.327             913X S.CONTY DS      1              CONSOLE TYPE FLAGS
 000.002             914X I.CUSOR EQU     2              S.CUSOR IS 3RD BYTE
 000.000             915X         ERRNZ   *-S.CSLMD-I.CUSOR
 040.330             916X S.CUSOR DS      1              CURRENT CURSOR POSITION
 000.003             917X I.CONWI EQU     3              S.CONWI IS 4TH BYTE
 000.000             918X         ERRNZ   *-S.CSLMD-I.CONWI
 040.331             919X S.CONWI DS      1              CONSOLE WIDTH
                     920X
 000.001             921X CO.FLG  EQU     00000001B      CTL-O FLAG
 000.200             922X CS.FLG  EQU     10000000B      CTL-S FLAG
                     923X
 000.004             924X I.CONFL EQU     4              S.CONFL IS 5TH BYTE
 000.000             925X         ERRNZ   *-S.CSLMD-I.CONFL
 040.332             926X S.CONFL DS      1              CONSOLE FLAGS
                     927X
 040.333             928X S.CAADR DS      2              ADDRESS FOR ABORT PROCESSING (>256 IF VALID)
 040.335             929X S.CCTAB DS      6              ADDR FOR CTL-A, CTL-B, CTL-C PROCESSING
 040.343             930          XTEXT   ESINT



                     932X **      S.INT - SYSTEM INTERNAL WORKAREA DEFINITIONS.
                     933X *
                     934X *       THESE CELLS ARE REFERENCED BY OVERLAYS AND MAIN CODE, AND
                     935X *       MUST THEREFORE RESIDE IN FIXED LOW MEMORY.
                     936X
                     937X
 040.343             938X         ORG     S.INT
                     939X
                     940X **      CONSOLE STATUS FLAGS
                     941X
 040.343             942X S.CDB   DS      1              CONSOLE DESCRIPTOR BYTE
 000.000             943X CDB.H85 EQU     00000000B
 000.001             944X CDB.H84 EQU     00000001B      =0 IF H8-5, =1 IF H8-4
 040.344             945X S.BAUD  DS      2              [0-14]  H8-4 BAUD RATE, =0 IF H8-5
                     946X *                              [15]     =1 IF BAUD RATE => 2 STOP BITS
                     947X
                     948X **      TABLE ADDRESS WORDS
                     949X
 040.346             950X S.DLINK DS      2              ADDRESS OF DATA IN HDOS CODE
 040.350             951X S.OFWA  DS      2              FWA  OVERLAY  TABLE
```

```
040.352              952X S.CFWA  DS       2                FWA  CHANNEL  TABLE
040.354              953X S.DFWA  DS       2                FWA  DEVICE   TABLE
040.356              954X S.RFWA  DS       2                FWA  RESIDENT HDOS CODE
                     955X
                     956X **      DEVICE DRIVER DELAYED LOAD FLAGS
                     957X
040.360              958X S.DDLDA DS       2                DRIVER LOAD ADDRESS (HIGH BYTE=0 IF NO LOAD PENDING)
040.362              959X S.DDLEN DS       2                CODE LENGTH IN BYTES
040.364              960X S.DDGRP DS       1                GROUP NUMBER FOR DRIVER
040.365              961X         DS       1                HOLD PLACE
                     962X *S.DDSEC         DS      2                SECTOR NUMBER FOR DRIVER ( * OBSOLETE ! * )
040.366              963X S.DDDTA DS       2                DEVICE'S ADDRESS IN DEVLST +DEV.RES
040.370              964X S.DDOPC DS       1                OPEN OPCODE PENDEDING
                     965X
                     966X **      OVERLAY MANAGEMENT FLAGS
                     967X
000.001              968X OVL.IN  EQU      00000001B        IN MEMORY
000.002              969X OVL.RES EQU      00000010B        PERMINANTLY RESIDENT
000.014              970X OVL.NUM EQU      00001100B        OVERLAY NUMBER MASK
000.200              971X OVL.UCS EQU      10000000B        USER CODE SWAPPED FOR OVERLAY
                     972X
040.371              973X S.OVLFL DS       1                OVERLAY FLAG
040.372              974X S.UCSF  DS       2                FWA SWAPPED USER CODE
040.374              975X S.UCSL  DS       2                LENGTH SWAPPED USER CODE
040.376              976X S.OVLS  DS       2                SIZE OF OVERLAY CODE
041.000              977X S.OVLE  DS       2                ENTRY POINT OF OVERLAY CODE
                     978X
041.002              979X S.SSN   DS       2                SWAP AREA SECTOR NUMBER
041.004              980X S.OSN   DS       2                OVERLAY SECTOR NUMBER
                     981X
                     982X *       SYSCALL PROCESSING WORK AREAS
                     983X
041.006              984X S.CACC  DS       1                (ACC) UPON SYSCALL
041.007              985X S.CODE  DS       1                SYSCALL INDEX IN PROGRESS
                     986X
                     987X *       JUMPS TO ROUTINES IN RESIDENT HDOS CODE
                     988X
041.010              989X S.JUMPS DS       0                START OF DUMP VECTORS
041.010              990X S.SDD   DS       3                JUMP TO STAND-IN DEVICE DRIVER
041.013              991X S.FASER DS       3                JUMP TO FATSERR (FATAL SYSTEM ERROR)
041.016              992X S.DIREA DS       3                JUMP TO DIREAD (DISK FILE READ)
041.021              993X S.FCI   DS       3                JUMP TO FCI (FETCH CHANNEL INFO)
041.024              994X S.SCI   DS       3                JUMP TO SCI (STORE CHANNEL INFO)
041.027              995X S.GUP   DS       3                JUMP TO GUP (GET UNIT POINTER)
                     996X
041.032              997X S.MOUNT DS       1                <>0 IF THE SYSTEM DISK IS MOUNTED
041.033              998X S.DCS   DS       1                DEFAULT CLUSTER SIZE-1
                     999X
041.034             1000X S.BOOTF DS       1                BOOT FLAGS
000.001             1001X BOOT.P  EQU      00000001B        EXECUTE PROLOGUE UPON BOOTUP
                    1002X
                    1003X *       STACK VALUE SAVED FOR OVERLAY SYSCALLS
                    1004X
041.035             1005X S.OVSTK DS       2                VALUE OF SP UPON SYSCALLS USING OVERLAY
                    1006X
041.037             1007X         DS       1                RESERVED
```

```
                 **      ACTIVE I/O AREA.
                 *
                 *       THE AIO.XXX AREA CONTAINS INFORMATION ABOUT THE I/O OPERATION
                 *       CURRENTLY BEING PERFORMED. THE INFORMATION IS OBTAINED FROM
                 *       THE CHANNEL TABLE, AND WILL BE RESTORED THERE WHEN DONE.
                 *
                 *       NORMALLY, THE AIO.XXX INFORMATION WOULD BE OBTAINED DIRECTLY
                 *       FROM VARIOUS SYSTEM TABLES VIA POINTER REGISTERS. SINCE THE
                 *       8080 HAS NO GOOD INDEXED ADDRESSING, THE DATA IS MANUALLY
                 *       COPIED INTO THE AIO.XXX CELLS BEFORE PROCESSING, AND
                 *       BACKDATED AFTER PROCESSING.

041.040          AIO.VEC DS      3               JUMP INSTRUCTION
041.041          AIO.DDA EQU     *-2             DEVICE DRIVER ADDRESS
041.043          AIO.FLG DS      1               FLAG BYTE
041.044          AIO.GRT DS      2               ADDRESS OF GROUP RESERV TABLE
041.046          AIO.SPG DS      1               SECTORS PER GROUP
041.047          AIO.CGN DS      1               CURRENT GROUP NUMBER
041.050          AIO.CSI DS      1               CURRENT SECTOR INDEX
041.051          AIO.LGN DS      1               LAST GROUP NUMBER
041.052          AIO.LSI DS      1               LAST SECTOR INDEX
041.053          AIO.DTA DS      2               DEVICE TABLE ADDRESS
041.055          AIO.DES DS      2               DIRECTORY SECTOR
041.057          AIO.DEV DS      2               DEVICE CODE
041.061          AIO.UNI DS      1               UNIT NUMBER (0-9)

041.062          AIO.DIR DS      DIRELEN         DIRECTORY ENTRY

041.111          AIO.CNT DS      1               SECTOR COUNT
041.112          AIO.EOM DS      1               END OF MEDIA FLAG
041.113          AIO.EOF DS      1               END OF FILE FLAG
041.114          AIO.TFP DS      2               TEMP FILE POINTERS
041.116          AIO.CHA DS      2               ADDRESS OF CHANNEL BLOCK (IOC.DDA)


041.120          S.BDA   DS      1               Boot Device Address (Setup by ROM) /80.09.sc/
041.121          S.SCR   DS      2               SYSTEM SCRATCH AREA ADDRESS
041.123                  XTEXT   H17ROM


                 **      H17 ROM DEFINITIONS

036.235          R.WHD   EQU     36235A
036.271          R.WNH   EQU     36271A
035.303          R.DLY   EQU     35303A
041.123                  XTEXT   ASCII
```

```
                        1054X **       ASCII CHARACTER EQUIVALENCES.
                        1055X
     000.015            1056X CR       EQU      13                CARRIAGE RETURN
     000.012            1057X LF       EQU      10                LINE FEED
     000.200            1058X NULL     EQU      200Q              PAD CHARACTER
     000.000            1059X NUL2     EQU      0
     000.007            1060X BELL     EQU      7                 BELL CHARACTER
     000.177            1061X RUBOUT   EQU      177Q
     000.010            1062X BKSP     EQU      10Q               CTL-H
     000.026            1063X C.SYN    EQU      26Q               SYNC
     000.002            1064X C.STX    EQU      2                 STX
     000.047            1065X QUOTE    EQU      47Q
     000.011            1066X TAB      EQU      11Q
     000.033            1067X ESC      EQU      33Q
     000.012            1068X NL       EQU      12Q               NEW LINE (HDOS SYSTEMS)
     000.212            1069X ENL      EQU      NL+200Q           NL + END-OF-LINE-FLAG
     000.014            1070X FF       EQU      14Q               FORM FEED
     000.001            1071X CTLA     EQU      01Q               CTL-A
     000.002            1072X CTLB     EQU      02Q               CTL-B
     000.003            1073X CTLC     EQU      03Q               CTL-C
     000.004            1074X CTLD     EQU      04Q               CTL-D
     000.017            1075X CTLO     EQU      17Q               CTL-O
     000.020            1076X CTLP     EQU      20Q               CTL-P
     000.021            1077X CTLQ     EQU      21Q               CTL-Q
     000.023            1078X CTLS     EQU      23Q               CTL-S
     000.032            1079X CTLZ     EQU      32Q               CTL-Z
     041.123            1080           XTEXT    INIDEF



                        1082X **       INIDEF  -  Init Sub-Function Definitions
                        1083X *
                        1084X
     000.000            1085X          ORG      0
                        1086X
     000.000            1087X INI.CMV  DS       1                 Check Media Validity
     000.001            1088X INI.IDS  DS       1                 Initialize Disk Surface
     000.002            1089X INI.DBI  DS       1                 Directory Block Interleave
     000.003            1090X INI.PAR  DS       1                 Parameters
                        1091X
     000.004            1092X INI.MAX  EQU      *



                        1094X **       Definition of Shared Routines
                        1095X *
                        1096X
     054.000            1097X INITVEC  EQU      54000A            Vector Address
                        1098X
     054.000            1099X          ORG      INITVEC
                        1100X
     054.000            1101X $$VER    DS       1                 Version of Init
                        1102X
     054.001            1103X $$BITC   DS       3                 Bit CLEAR
     054.004          $$BITS  DS      3            Bit SET

     054.007          $$BUFF  DS      2            256 byte buffer

     054.011          $$CHL   DS      3            Complement HL

     054.014          $$CNO   DS      3            Check  NO

     054.017          $$CYS   DS      3            Check  YES

     054.022          $$DRVR  DS      3            Device Driver

     054.025          $$DRVR. DS      3            Device Driver with ERROR detection

     054.030          $$ITL.  DS      3            Input Text Line

     054.033          $$MOVE  DS      3            Move bytes

     054.036          $$TBRA  DS      3            Table Branch

     054.041          $$TYPTX DS      3            Type Text

     054.044          $$VSN   DS      3            Volume Serial Number

     054.047          $$MAX   EQU     *
```

```
044.170                      ORG     SB.SDB-ABS.COD



044.170  377 000             DB      377Q,FT.ABS
044.172  200 044             DW      SB.SDB           LOAD ADDR
044.174  211 033             DW      MEML-SB.SDB      SIZE
044.176  332 072             DW      PRSS             ENTRY

                             LON     C


                     **      S0BOOT  - SECTOR 0 BOOT ROUTINE.
                     *
                     *       THIS BOOT STARTS AT SECTOR 0 ON EVERY INITIALIZED
                     *       DISK, AND OCCUPIES THE FIRST 9 SECTORS OF THE DISK.
                     *
                     *       IT IS BROUGHT IN BY THE H17 ROM.



044.200  061 200 042 S0BOOT  LXI     SP,STACK         Insure valid STACK         /80.06.sc/
044.203  041 031 034         LXI     H,CLOCK                                     /80.06.sc/
044.206  042 040 040         SHLD    .UIVEC+1         Insure valid CLOCK         /80.06.sc/
044.211  257                 XRA     A                                           /80.09.sc/
044.212  062 174 053         STA     ABF              Time-Out initially valid   /80.09.sc/

044.215  315 210 050         CALL    FCU              FIND CONSOLE USART         /80.05.GC/

044.220  061 200 042 S0BOOTX LXI     SP,STACK
044.223  257                 XRA     A
044.224  062 062 041         STA     AIO.DIR          AM WORKING WITH NO FILES YET
044.227  315 027 047         CALL    $TYPET
044.232  000 012 101         DB      0,LF,'ACTION? <BOOT>',' '+200Q              /80.09.sc/

                     *       GET REPLY. MAY BE:
                     *
                     *       BOOT
                     *       CHECK
                     *       HELP
                     *       Ignore                                              /80.05.sc/

044.253  016 074     S0BOOTY MVI     C,60             C = Time-Out Counter       /80.08.sc/
044.255  315 303 050         CALL    RCC              INPUT TASK TIME            /80.08.sc/
044.260  332 333 044         JC      S0BOOT1          Output Message at new Baud-Rate /80.08.sc/
044.263  107                 MOV     B,A              Save character if any      /80.08.sc/
044.264  171                 MOV     A,C                                         /80.08.sc/
044.265  247                 ANA     A                                           /80.09.sc/
044.266  302 303 044         JNZ     S0BOOT0          Not Time-Out               /80.09.sc/

044.271  041 174 053         LXI     H,ABF                                       /80.09.sc/
044.274  266                 ORA     M                Check Auto-Boot Validity   /80.09.sc/
044.275  312 076 045         JZ      S0BOOT2          Time-Out                   /80.08.sc/
044.300  303 253 044         JMP     S0BOOTY                                     /80.09.sc/

044.303  170         S0BOOT0 MOV     A,B              A  = saved character       /80.08.sc/
044.304  315 071 052         CALL    $MCU             MAP TO UPPER CASE
```

```
044.307  376 015                CPI     CR
044.311  312 076 045            JE      SOBOOT2         IS BOOT
044.314  376 102                CPI     'B'
044.316  312 076 045            JE      SOBOOT2         IS BOOT
044.321  376 103                CPI     'C'
044.323  312 146 045            JE      SOBOOT4         IS CHECK
044.326  376 111                CPI     'I'
044.330  312 170 045            JE      SOBOOT5         IS IGNORE

                       *        ASSUME HELP

044.333  315 027 047   SOBOOT1  CALL    $TYPET
044.336  110 105 114            DB      'HELP',0,0
044.344  114 105 107            DB      'LEGAL COMMANDS:',0
044.364  102 117 117            DB      'BOOT   - BOOT HDOS',0
045.007  103 110 105            DB      'CHECK  - SECTOR CHECKSUMS',0
045.041  110 105 114            DB      'HELP   - PRINT THIS LIST',0
000.001                         IF      PUBLIC
                                DB      'IGNORE - IGNORE PROLOGUE FILE'
                                ENDIF
045.072  200                    DB      200Q
045.073  303 220 044            JMP     SOBOOTX         TRY AGAIN

                       *        IS BOOT

045.076  041 174 053   SOBOOT2  LXI     H,ABF                                           /80.09.sc/
045.101  066 001                MVI     M,1             Subsequent Auto-Boots are invalid /80.09.sc/
045.103  315 027 047            CALL    $TYPET
045.106  102 117 117            DB      'BOOT',200Q
045.113  072 034 041            LDA     S.BOOTF
045.116  366 001                ORI     BOOT.P          FLAG PROLOGUE EXECUTION UPON BOOTUP

045.120  062 034 041   SOBOOT3  STA     S.BOOTF                                         /80.09.sc/
045.123  315 335 047            CALL    MSD             MOUNT THIS DISK
045.126  315 212 045            CALL    LEP             LOAD AND EXECUTE PROGRAM
045.131  110 104 117            DB      'HDOS',0,0,0,0
045.141  123 131 123            DB      'SYS',0,0

                       *        IS CHECK

045.146  315 335 047   SOBOOT4  CALL    MSD             Mount the System Disk           /80.09.sc/
045.151  315 027 047            CALL    $TYPET
045.154  103 110 105            DB      'CHECK',200Q
045.162  315 102 052            CALL    CDC             COMPUTE DISK CCHECKS
045.165  303 220 044            JMP     SOBOOTX         TRY AGAIN

                       *        IS IGNORE

045.170  315 027 047   SOBOOT5  CALL    $TYPET
045.173  111 107 116            DB      'IGNORE',200Q
045.202  072 034 041            LDA     S.BOOTF
045.205  346 376                ANI     377Q-BOOT.P     Turn off Prologue at BOOT       /80.09.sc/
045.207  303 120 045            JMP     SOBOOT3
```

```
                      **      LEP IS CALLED TO LOAD AND EXECUTE A DISK FILE.
                      *
                      *       THE DISKS DIRECTORY IS SEARCHED FOR THE APPROPRIATE FILE NAME.
                      *       IF FOUND, IT IS LOADED INTO MEMORY AT SB.ORG AND EXECUTED.
                      *
                      *       IF NOT FOUND, TYPE ERROR MESSAGE:
                      *
                      *       NEEDED FILE * FNAME * IS MISSING
                      *
                      *       AND RETURN TO SOBOOT.
                      *
                      *       IF ERROR IN READING THE FILE, TYPE
                      *
                      *       'DISK READ ERROR IN FILE * FNAME *'
                      *
                      *       AND RE-BOOT.
                      *
                      *       ENTRY   ((SP)) = FILE NAME
                      *       EXIT    TO SB.ORG IF LOAD SUCCESSFUL,
                      *               TO ROMBOOT IF READ ERROR,
                      *               TO SOBOOTX IF FILE MISSING
                      *       USES    ALL


045.212  321          LEP     POP     D                 (DE) = NAME ADDRESS
045.213  001 015 000          LXI     B,DIRIDL
045.216  041 062 041          LXI     H,AIO.DIR
045.221  315 252 030          CALL    $MOVE             MOVE IN NAME
045.224  072 215 053          LDA     BLABEL+LAB.VLT            (A) = VOLUME TYPE
045.227  247                  ANA     A
000.000                       ERRNZ   LAB.DAT
045.230  312 214 046          JZ      LEP5              IS DATA DISK
045.233  075                  DCR     A
000.000                       ERRNZ   LAB.SYS-1
045.234  302 070 046          JNZ     LEP6              IS GARBAGE DISK
045.237  001 015 000          LXI     B,DIRIDL
045.242  052 210 053          LHLD    BLABEL+LAB.DIS
045.245  315 146 047          CALL    LDE.,             LOAD ENTRY
045.250  322 323 045          JNC     LEP1              FOUND

                      *       COULDNT FIND IT

045.253  315 027 047          CALL    $TYPET
045.256  007 077 060          DB      BELL,'?00 REQUIRED FILE',' '+200Q
045.301  315 375 046          CALL    TFN               TYPE FILE NAME
045.304  315 027 047          CALL    $TYPET
045.307  040 115 111          DB      ' MISSING',BELL+200Q
045.320  303 220 044          JMP     SOBOOTX

                      *       GOT DIRECTORY ENTRY. TRY TO READ IT

045.323  021 016 000  LEP1    LXI     D,DIR.FLG
045.326  031                  DAD     D                 (HL) = ADDRESS OF FLG
045.327  176                  MOV     A,M
045.330  346 020              ANI     DIF.CNT
045.332  312 004 046          JZ      LEP4              NOT CONTIGUOUG
```

```
000.000                         ERRNZ   DIR.FGN-DIR.FLG-2
045.335   043                   INX     H
045.336   043                   INX     H                 (HL) = $DIR.FGN
045.337   136                   MOV     E,M
045.340   026 000               MVI     D,0               (DE) = GROUP NUMBER
045.342   072 214 053           LDA     BLABEL+LAB.SPG    (A) = SECTORS PER GROUP
045.345   315 007 031           CALL    $MU86             (HL) = SECTOR NUMBER FOR FILE

                        *       (HL) = SECTOR NUMBER FOR FILE

045.350   001 000 001   LEP3    LXI     B,256
045.353   021 000 051           LXI     D,SB.ORG
045.356   345                   PUSH    H
045.357   315 232 047           CALL    READD             READ DISK
045.362   052 002 051           LHLD    SB.ORG+PIC.LEN
045.365   053                   DCX     H                 (HL) = SECTOR COUNT
045.366   104                   MOV     B,H
045.367   016 000               MVI     C,0
045.371   341                   POP     H                 (HL) = SECTOR NUMBER OF FWA
045.372   043                   INX     H                 ALREADY READ 1
045.373   021 000 052           LXI     D,SB.ORG+256
045.376   315 232 047           CALL    READD             READ THE REMAINDER
046.001   303 006 051           JMP     SB.ORG+PIC.COD    ALL OK, EXECUTE IT

                        *       FILE NOT CONTIGUOUS

046.004   315 027 047   LEP4    CALL    $TYPET
046.007   000 007 077           DB      0,BELL,'?00 THIS DISK HAS NOT BEEN PROPERLY SYSGENED.',BELL,200Q
                        *       JMP     LEP6                                                /80.09.sc/
000.000                         ERRNZ   *-LEP6                                              /80.09.sc/

                        *       GARBAGE DISK.                                               /80.09.sc/

046.070   315 027 047   LEP6    CALL    $TYPET
046.073   000 007 077           DB      0,BELL,'?00 THIS DISK MUST BE INITIALIZED AND THEN SYSGENED'
046.160   000 102 105           DB      0,'BEFORE IT CAN BE USED.',BELL,200Q                /80.09.sc/
046.211   303 220 044           JMP     SOBOOTX

                        *       IS DATA DISK. NOT YET SYSGENED

046.214   315 027 047   LEP5    CALL    $TYPET
046.217   000 007 077           DB      0,BELL,'?00 THIS DISK MUST BE SYSGENED BEFORE IT CAN BE USED.',BELL,200Q
046.310   303 220 044           JMP     SOBOOTX
```

```
      046.313              1344          XTEXT   ICTT


                           1346X **      $ICTT - INPUT FROM CONSOLE TASK TIME.
                           1347X *
                           1348X *       $ICTT IS A TASK-TIME CONSOLE INPUT ROUTINE, WHICH
                           1349X *       PERFORMS SIMPLE SINGLE CHARACTER INPUTS.
                           1350X *
                           1351X *       IT IS CALLED DURING BOOT OPERATIONS, AND BY SPECIAL ROUTINES
                           1352X *       WHICH MAY BE RUNNING IN ENVIRONMENTS WHERE KEYBOARD INTERRUPTS
                           1353X *       ARE UNDESIRABLE.
                           1354X *
                           1355X *       Modified to handle H8-4 ports by G. Chandler, 1-SEP-78
                           1356X *       This routine assumes that the ports have been previously initialized,
                           1357X *       and that S.CDB has been previously initialized.
                           1358X *
                           1359X *       ENTRY   NONE
                           1360X *       EXIT    (A) = CHARACTER
                           1361X *       USES    A,F
                           1362X
                           1363X
      046.313  315 325 046 1364X $ICTT   CALL    $ICTT.
      046.316  332 313 046 1365X         JC      $ICTT
      046.321  315 353 046 1366X         CALL    $ICTT..
      046.324  311         1367X         RET
                           1368X
      046.325  072 343 040 1369X $ICTT.  LDA     S.CDB
      046.330  376 001     1370X         CPI     CDB.H84
      046.332  312 344 046 1371X         JZ      ICTT2           IF  H8-4 PORT
                           1372X
                           1373X *       HAVE 8251 FOR CONSOLE
                           1374X
      046.335  333 373     1375X ICTT1   IN      SC.UART+USR
      046.337  346 002     1376X         ANI     USR.RXR
      046.341  300         1377X         RNZ                     READY
                           1378X
      046.342  067         1379X         STC                     FLAG NOT READY
      046.343  311         1380X         RET
                           1381X
                           1382X *       HAVE 8250 PORT FOR CONSOLE
                           1383X
      046.344  333 355     1384X ICTT2   IN      SC.ACE+UR.LSR
      046.346  346 001     1385X         ANI     UC.DR
      046.350  300         1386X         RNZ                     READY
                           1387X
      046.351  067         1388X         STC                     FLAG NOT READY
      046.352  311         1389X         RET
                           1390X
      046.353  072 343 040 1391X $ICTT.. LDA     S.CDB
      046.356  376 001     1392X         CPI     CDB.H84
      046.360  312 370 046 1393X         JZ      ICTT3
                           1394X
                           1395X *       HAVE 8251 FOR CONSOLE
                           1396X
```

```
       046.363  333 372      1397X            IN      SC.UART+UDR
       046.365  346 177      1398X            ANI     177Q
       046.367  311          1399X            RET
                             1400X
                             1401X *          HAVE 8250 FOR CONSOLE
                             1402X
       046.370  333 350      1403X ICTT3      IN      SC.ACE+UR.RBR
       046.372  346 177      1404X            ANI     177Q
       046.374  311          1405X            RET



                             1407  **         TFN - TYPE FILE NAME.
                             1408  *
                             1409  *          TFN TYPES THE FILE WHOSE NAME APPEARS IN AIO.XXX
                             1410  *
                             1411  *          ENTRY   NONE
                             1412  *          EXIT    NONE
                             1413  *          USES    A,F,B,H,L
                             1414
                             1415
       046.375  041 062 041  1416  TFN        LXI     H,AIO.DIR+DIR.NAM
       047.000  006 010      1417             MVI     B,8
       047.002  315 014 047  1418             CALL    TFN1                TYPE NAME
       047.005  076 056      1419             MVI     A,'.'
       047.007  315 064 047  1420             CALL    $TYPEC.
       047.012  006 003      1421             MVI     B,3
                             1422
       047.014  176          1423  TFN1       MOV     A,M
       047.015  247          1424             ANA     A
       047.016  304 064 047  1425             CNZ     $TYPEC.
       047.021  043          1426             INX     H
       047.022  005          1427             DCR     B
       047.023  302 014 047  1428             JNZ     TFN1
       047.026  311          1429             RET
       047.027               1430             XTEXT   MU86



                             1432X **         $MU86 - MULTIPLY 8X16 UNSIGNED.
                             1433X *
                             1434X *          $MU86 MULTIPLIES A 16 BIT VALUE BY A 8
                             1435X *          BIT VALUE.
                             1436X *
                             1437X *          ENTRY   (A) = MULTIPLIER
                             1438X *                  (DE) = MULTIPLICAND
                             1439X *          EXIT    (HL) = RESULT
                             1440X *                  'Z' SET IF NOT OVERFLOW
                             1441X *          USES    A,F,H,L
                             1442X
                             1443X
       031.007               1444X $MU86      EQU     31007A              IN H17 ROM
       047.027               1445             XTEXT   TYPET
```

```
                            1447X **       $TYPET - TYPE TEXT.
                            1448X *
                            1449X *        $TYPET IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE
                            1450X *        AT TASK TIME RATHER THAN AT INTERRUPT TIME.
                            1451X *
                            1452X *        IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED,
                            1453X *        A BYTE WITH THE 200Q BIT SET IS THE LAST BYTE OF THE MESSAGE.
                            1454X *
                            1455X *        This routine modified to accomodate H8-4 ports by G.Chandler, 1-SEP-78.
                            1456X *        This routine assumes that the ports have been previously initialized,
                            1457X *        and that S.CDB has been previously initialized.
                            1458X *
                            1459X *        ENTRY    (RET) = TEXT
                            1460X *        EXIT     TO (RET+LENGTH)
                            1461X *        USES     A,F
                            1462X
                            1463X
    047.027  343            1464X $TYPET   XTHL                        (HL) = TEXT ADDRESS
    047.030  315 035 047    1465X          CALL     $TYPET.            TYPE IT
    047.033  343            1466X          XTHL
    047.034  311            1467X          RET
                            1468X
    047.035  176            1469X $TYPET.  MOV      A,M
    047.036  346 177        1470X          ANI      177Q
    047.040  304 064 047    1471X          CNZ      $TYPEC.            IF NOT CRLF
    047.043  247            1472X          ANA      A
    047.044  314 055 047    1473X          CZ       $TYPET1            IS CRLF
    047.047  276            1474X          CMP      M
    047.050  043            1475X          INX      H
    047.051  300            1476X          RNE                         WAS 200 BIT SET
    047.052  303 035 047    1477X          JMP      $TYPET.
                            1478X
                            1479X *        TYPE CRLF
                            1480X
    047.055  315 027 047    1481X $TYPET1  CALL     $TYPET
    047.060  015 212        1482X          DB       CR,LF+200Q
    047.062  257            1483X          XRA      A                  RESTORE (A)
    047.063  311            1484X          RET



                            1486X **       $TYPEC. - TYPE SINGLE CHARACTER.
                            1487X *
                            1488X *        IF CR, PADD WITH 4 ZERO BYTES
                            1489X *
                            1490X *        ENTRY    (A) = CHARACTER
                            1491X *        EXIT     (A) = CHARACTER
                            1492X *        USES     A,F
                            1493X
                            1494X
    047.064  365            1495X $TYPEC.  PUSH     PSW                SAVE CHAR
    047.065  072 343 040    1496X          LDA      S.CDB
    047.070  376 001        1497X          CPI      CDB.H84
    047.072  312 112 047    1498X          JZ       TYPEC2             IF H8-4 PORT
                            1499X
```

```
                          1500X *       HAVE 8251 PORT FOR CONSOLE
                          1501X
047.075  333 373          1502X TYPEC1  IN      SC.UART+USR
047.077  346 001          1503X         ANI     USR.TXR
047.101  312 075 047      1504X         JZ      TYPEC1          NOT READY
047.104  361              1505X         POP     PSW
047.105  323 372          1506X         OUT     SC.UART+UDR
047.107  303 124 047      1507X         JMP     TYPEC3
                          1508X
                          1509X *       HAVE 8250 PORT FOR CONSOLE
                          1510X
047.112  333 355          1511X TYPEC2  IN      SC.ACE+UR.LSR
047.114  346 040          1512X         ANI     UC.THE
047.116  312 112 047      1513X         JZ      TYPEC2          NOT READY
047.121  361              1514X         POP     PSW
047.122  323 350          1515X         OUT     SC.ACE+UR.THR
                          1516X
047.124  376 015          1517X TYPEC3  CPI     CR
047.126  300              1518X         RNE                     NOT CR
                          1519X
                          1520X *       IS CR. PADD 4 TIMES
                          1521X
047.127  076 004          1522X         MVI     A,4
047.131  365              1523X TYPEC4  PUSH    PSW
047.132  257              1524X         XRA     A
047.133  315 064 047      1525X         CALL    $TYPEC.
047.136  361              1526X         POP     PSW
047.137  075              1527X         DCR     A
047.140  302 131 047      1528X         JNZ     TYPEC4
047.143  076 015          1529X         MVI     A,CR
047.145  311              1530X         RET
047.146                   1531          XTEXT   MOVE


                          1533X **      $MOVE - MOVE DATA
                          1534X *
                          1535X *       $MOVE MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
                          1536X *       IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
                          1537X *       FIRST TO LAST.
                          1538X *
                          1539X *       IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
                          1540X *       LAST TO FIRST.
                          1541X *
                          1542X *       THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
                          1543X *
                          1544X *       ENTRY   (BC) = COUNT
                          1545X *               (DE) = FROM
                          1546X *               (HL) = TO
                          1547X *       EXIT    MOVED
                          1548X *               (DE) = ADDRESS OF NEXT FROM BYTE
                          1549X *               (HL) = ADDRESS OF NEXT *TO* BYTE
                          1550X *               'C' CLEAR
                          1551X *       USES    ALL
                          1552X
```

```
                      1553X
030.252               1554X $MOVE   EQU     30252A          IN H17 ROM
047.146               1555          XTEXT   COMP


                      1557X **      $COMP - COMPARE TWO CHARACTER STRINGS.
                      1558X *
                      1559X *       $COMP COMPARES TWO BYTE STRINGS.
                      1560X *
                      1561X *       ENTRY   (C) = COMPARE COUNT
                      1562X *               (DE) = FWA OF STRING #1
                      1563X *               (HL) = FWA OF STRING #2
                      1564X *       EXIT    'Z' CLEAR, IS MIS-MATCH
                      1565X *                (C) = LENGTH REMAINING
                      1566X *                (DE) = ADDRESS OF MISMATCH IN STRING#1
                      1567X *                (HL) = ADDRESS OF MISMATCH IN STRING #2
                      1568X *               'C' SET, HAVE MATCH
                      1569X *                (C) = 0
                      1570X *                (DE) = (DE) + (OC)
                      1571X *                (HL) = (HL) + (OC)
                      1572X *       USES    A,F,C,D,E,H,L
                      1573X
                      1574X
030.060               1575X $COMP   EQU     30060A          IN H17 ROM
047.146               1576          XTEXT   DADA2


                      1578X **      $DADA. - ADD (0,A) TO (H,L)
                      1579X *
                      1580X *       ENTRY   NONE
                      1581X *       EXIT    (HL) = (HL) + (0A)
                      1582X *       USES    A,F,H,L
                      1583X
                      1584X
030.101               1585X $DADA.  EQU     30101A          IN H17 ROM
                      1586


                      1588  **      LDE - LOCATE DIRECTORY ENTRY.
                      1589  *
                      1590  *       LDE LOCATES A DIRECTORY ENTRY CORRESPONDING TO THE AIO.DIR ENTRY.
                      1591  *
                      1592  *       ENTRY   (BC) = NUMBER OF CHARACTERS TO MATCH ON
                      1593  *       EXIT    'C' CLEAR IF FOUND
                      1594  *                AIO.DES SETUP
                      1595  *                (HL) = ADDRESS OF DIRECTORY ENTRY IN BUFF
                      1596  *               'C' SET IF NOT FOUND
                      1597  *                (A) = CODE
                      1598  *       USES    ALL
                      1599
```

```


                        **      ENTRY FOR (HL) = SECTOR NUMBER TO START WITH

047.146  305            LDE..   PUSH    B                  SAVE COUNT
047.147  001 000 002            LXI     B,512
047.152  021 205 054            LXI     D,BUFF
047.155  042 055 041            SHLD    AIO.DES            ASSUME WILL FIND IN THIS BLOCK
047.160  315 232 047            CALL    READD              READ FRM DEVICE
047.163  301                    POP     B                  RESTORE (BC)

                        *       SCAN SECTOR FOR INFO

047.164  041 205 054            LXI     H,DIS.ENT+BUFF

                        *       COMPARE

047.167  021 062 041    LDE3    LXI     D,AIO.DIR
047.172  305                    PUSH    B                  SAVE COPY OF (BC)
047.173  345                    PUSH    H                  SAVE ADDRESS
047.174  315 060 030            CALL    $COMP              COMPARE
047.177  341                    POP     H
047.200  301                    POP     B                  (BC) = COMPARE COUNT
047.201  310                    RE                         GOT MATCH
047.202  021 027 000            LXI     D,DIRELEN          MISSED, SCAN TO NEXT ENTRY
047.205  031                    DAD     D
047.206  176                    MOV     A,M
047.207  247                    ANA     A
047.210  302 167 047            JNZ     LDE3               MORE IN SECTOR

                        *       DIDNT FIND IT IN THIS SECTOR, TRY NEXT

047.213  052 203 056            LHLD    DIS.LNK+BUFF
047.216  042 055 041            SHLD    AIO.DES            SET POSSIBLE SECTOR INDEX
047.221  174                    MOV     A,H
047.222  265                    ORA     L
047.223  302 146 047            JNZ     LDE..              HAVE MORE SECTORS
047.226  076 014                MVI     A,EC.FNF           FILE NOT FOUND
047.230  067                    STC
047.231  311                    RET



                        **      READD - READ DISK.
                        *
                        *       READD CALLS THE SYSTEM DEVICE DRIVER FOR A
                        *       READ OPERATION.
                        *
                        *       IF AN ERROR OCCURS, A MESSAGE IS PRINTED, AND THE
                        *       BOOT OPERATION RESTARTS.
                        *
                        *       ENTRY   REGISTERS SET FOR READ
                        *       EXIT    FROM SYDD
                        *       USES    ALL

                        1653
047.232  257            1654 READD   XRA    A               A = 0                      /80.09.sc/
000.000                 1655         ERRNZ  DC.REA                                     /80.09.sc/
047.233  315 130 040    1656 READD1  CALL   SYDD            ISSUE READ                      /2.0b/
047.236  320            1657         RNC                    ALL OK
                        1658
                        1659 *       READ ERROR
                        1660
047.237  315 027 047    1661 READERR CALL   $TYPET
047.242  000 000 007    1662         DB     0,0,BELL,'?00 DISK READ ERROR DURING BOOT.',0
047.306  007 102 117    1663         DB     BELL,'BOOT ABORTED.',200Q                       /2.0b/
047.325  303 000 000    1664         JMP    0                                               /2.0b/
                        1665
047.330  076 002        1666 READR   MVI    A,DC.RER                                        /2.0b/
047.332  303 233 047    1667         JMP    READD1          Read Regardless                 /2.0b/


                        1669 **      MSD - MOUNT SYSTEM DISK.
                        1670 *
                        1671 *       MSD MOUNTS THE SYSTEM DISK.
                        1672 *
                        1673 *       1) ABORT DRIVER
                        1674 *       2) READ BLABEL RECORD
                        1675 *       3) SET VOLUME NUMBER FOR DIRIVER
                        1676 *       4) INITIALIZE DEVICE TABLE
                        1677 *       5) BUILD GRT
                        1678
                        1679
047.335  076 007        1680 MSD     MVI    A,DC.ABT
047.337  315 130 040    1681         CALL   SYDD            ABORT DRIVER
047.342  056 000        1682         MVI    L,0             Mount volume 0                  /2.0b/
047.344  315 013 050    1683         CALL   MSD.                                            /2.0b/
                        1684
047.347  001 000 001    1685         LXI    B,256
047.352  021 205 053    1686         LXI    D,BLABEL
047.355  041 011 000    1687         LXI    H,DDF.LAB
047.360  315 330 047    1688         CALL   READR           Read Regardless                 /2.0b/
                        1689
                        1690 *       CALL DEVICE MOUNT ROUTINE
                        1691
047.363  052 205 053    1692         LHLD   BLABEL+LAB.SER  L = Volume Number               /2.0b/
047.366  315 013 050    1693         CALL   MSD.            Mount the volume                /2.0b/
                        1694
                        1695 *       Patch Label for old diskettes              /80.09.sc/
                        1696
047.371  072 216 053    1697         LDA    BLABEL+LAB.VER
047.374  376 027        1698         CPI    017H
047.376  320            1699         RNC                    Version >= 1.7
                        1700
047.377  041 220 001    1701         LXI    H,400
050.002  042 221 053    1702         SHLD   BLABEL+LAB.SIZ  Force size of 400
050.005  076 012        1703         MVI    A,10
050.007  062 324 053    1704         STA    BLABEL+LAB.SPT  10 Sectors per Track
050.012  311            1705         RET
```

```

                          *       Mount the volume                                             /2.0b/

050.013  046 000          MSD.    MVI     H,0                                                  /2.0b/
050.015  076 010                  MVI     A,DC.MOU                                             /2.0b/
050.017  315 130 040              CALL    SYDD                                                 /2.0b/
050.022  334 237 047              CC      READERR                                              /2.0b/
050.025  311                      RET                                                          /2.0b/

050.026                           DS      32              PATCH AREA                           /2.0b/

                          **      ALL CODE FOLLOWING MAY BE OVERLAID BY THE HDOS.SYS
                          *       PROGRAM DURING BOOT.

377.066                           ERRPL   *-SB.ORG        MUST BE BEFORE ORG ADDRESS


                          **      CBS     - Check Break Status                         /80.08.sc/
                          *
                          *       CBS checks the break status to see if the user
                          *       has hit the break key.  If so, it is assumed
                          *       that the user wishes to re-determine the baud-rate.
                          *
                          *       This routine also waits for a character from the
                          *       selected device.
                          *
                          *       ENTRY:  C       = Time-Out Counter
                          *               S.CDB   = Console definition byte
                          *
                          *       EXIT:   PSW     = 'Z' if
                          *                               Time-Out
                          *                               character ready
                          *                         'NZ' if
                          *                               Baud-Rate changed
                          *
                          *       USES:   PSW,BC
                          *

050.066  072 343 040      CBS     LDA     S.CDB
050.071  247                      ANA     A
000.000                           ERRNZ   CDB.H85
050.072  302 115 050              JNZ     CBS2            Not an 8251

050.075  315 163 050      CBS1    CALL    CBS4
050.100  310                      RZ                      Time-Out
050.101  333 373                  IN      SC.UART+USR
050.103  346 102                  ANI     USR.BD+USR.RXR  Check for Break, OR character
050.105  312 075 050              JZ      CBS1
050.110  346 100                  ANI     USR.BD          Check for Break only
050.112  303 132 050              JMP     CBS3

050.115  315 163 050      CBS2    CALL    CBS4
050.120  310                      RZ                      Time-Out
```

```
050.121  333 355                   IN      SC.ACE+UR.LSR
050.123  346 031                   ANI     UC.BI+UC.DR+UC.FE       Check: Break, Frame error, character
050.125  312 115 050               JZ      CBS2
050.130  346 030                   ANI     UC.BI+UC.FE             Check: Break, Frame error

050.132  310               CBS3    RZ                      No Break, so must be character

050.133  365               CBS.    PUSH    PSW             Re-Determine Baud-Rate
050.134  305                       PUSH    B
050.135  345                       PUSH    H

050.136  052 205 042               LHLD    SB.BAU
050.141  345                       PUSH    H
050.142  041 000 000               LXI     H,0
050.145  042 205 042               SHLD    SB.BAU          Over-Ride Default Baud-Rate
050.150  315 210 050               CALL    FCU
050.153  341                       POP     H
050.154  042 205 042               SHLD    SB.BAU          Restore old Baud-Rate

050.157  341                       POP     H
050.160  301                       POP     B
050.161  361                       POP     PSW
050.162  311                       RET

050.163  072 033 040       CBS4    LDA     .TICCNT
050.166  247                       ANA     A
050.167  312 174 050               JZ      CBS5            1/2 Second Interval
050.172  107                       MOV     B,A             B = last TicCnt
050.173  311                       RET

050.174  170               CBS5    MOV     A,B             A = Last TicCnt
050.175  247                       ANA     A
050.176  302 204 050               JNZ     CBS6            This 1/2 Second Not counted yet
050.201  366 001                   ORI     1
050.203  311                       RET                     No Time-Out Yet

050.204  006 000           CBS6    MVI     B,0             Flag this TicCnt counted
050.206  015                       DCR     C               Count the 1/2 second
050.207  311                       RET



                           **      FCU - FIND CONSOLE USART.
                           *
                           *       FCU FINDS AND CONFIGURES THE CONSOLE USART.
                           *
                           *       ENTRY   NONE
                           *       EXIT    NONE
                           *       USES    A,F,(BC),(HL)
                           *

050.210  257               FCU     XRA     A
050.211  323 351                   OUT     SC.ACE+UR.IER           OFF INTERRUPTS
050.213  323 373                   OUT     SC.UART+USR             OFF INTERRUPTS

```

```
                         1812  *       SEE IF WE HAVE AN 8250
                         1813
050.215  076 003         1814          MVI     A,UC.8BW
050.217  323 353         1815          OUT     SC.ACE+UR.LCR
050.221  333 353         1816          IN      SC.ACE+UR.LCR
050.223  376 003         1817          CPI     UC.8BW          SEE IF UNCHANGED
050.225  076 000         1818          MVI     A,CDB.H85
050.227  041 000 000     1819          LXI     H,0
050.232  302 272 050     1820          JNE     FCU1            IS 8251
                         1821
050.235  052 205 042     1822          LHLD    SB.BAU                                     /80.08.sc/
050.240  174             1823          MOV     A,H                                        /80.08.sc/
050.241  265             1824          ORA     L                                          /80.08.sc/
050.242  076 001         1825          MVI     A,CDB.H84                                  /80.08.sc/
050.244  302 272 050     1826          JNZ     FCU1            Default Baud-Rate Provided /80.08.sc/
                         1827
050.247  315 000 051     1828          CALL    ABR             AUTO SET BAUD RATE
050.252  174             1829          MOV     A,H
050.253  346 200         1830          ANI     10000000B
050.255  312 270 050     1831          JZ      FCU0
050.260  072 327 040     1832          LDA     S.CONTY
050.263  366 010         1833          ORI     CTP.2SB
050.265  062 327 040     1834          STA     S.CONTY         SET TWO STOP BITS
050.270  076 001         1835  FCU0    MVI     A,CDB.H84
                         1836
                         1837  *       HAVE TYPE AND BAUDRATE.
                         1838  *       (A) = S.CDB VALUE
                         1839  *       (HL) = BAUD RATE (0 IF 8251)
                         1840
050.272  042 344 040     1841  FCU1    SHLD    S.BAUD
050.275  062 343 040     1842          STA     S.CDB
050.300  303 334 051     1843          JMP     SCU                                        /80.09.sc/



                         1845  **      RCC      - Read Console Character
                         1846  *
                         1847  *       RCC reads a console character from the console
                         1848  *       flagged by S.CDB.  The difference between this
                         1849  *       read and that of $ICTT is that the break status
                         1850  *       of each of the UARTS is checked while waiting
                         1851  *       for a character.
                         1852  *
                         1853  *       ENTRY:  BC       = Time-Out Counter
                         1854  *               S.CDB set up
                         1855  *
                         1856  *       EXIT:   PSW      = 'C' if
                         1857  *                                 Baud-Rate Changed
                         1858  *                            'NC' if
                         1859  *                                 Character
                         1860  *                                 Time-Out
                         1861  *                              C = Time-Out counter == 0 for Time-Out
                         1862  *                              A  = Character read
                         1863  *
                         1864  *       USES:   PSW,BC
```

```
                              1865 *
                              1866
      050.303  315 066 050    1867 RCC       CALL    CBS
      050.306  067            1868           STC
      050.307  300            1869           RNZ                     No character read, UART re-initialized
                              1870
      050.310  171            1871           MOV     A,C
      050.311  247            1872           ANA     A
      050.312  310            1873           RZ                      Time-Out
                              1874
      050.313  303 313 046    1875           JMP     $ICTT           Input the character
      050.316                 1876           XTEXT   BCRC


                              1878X **       $BCRC - GENERATE CRC16 ON A BLOCK OF DATA.
                              1879X *
                              1880X *        *** WARNING ***
                              1881X *
                              1882X *        THIS CRC-16 IS NOT COMPATIBLE WITH THE ONE
                              1883X *        PRODUCED BY PAM-8, AND THE DECK CRC.COM!
                              1884X *
                              1885X *        ENTRY   (BC) = BYTE COUNT
                              1886X *                (HL) = ADDRESS
                              1887X *                (DE) = CRC ACCUMULATOR
                              1888X *        EXIT    (HL) = (HL)+(BC)
                              1889X *                (DE) = NEW CRC
                              1890X *        USES    ALL
                              1891X
                              1892X
      050.316  170            1893X $BCRC    MOV     A,B
      050.317  261            1894X          ORA     C
      050.320  310            1895X          RZ                      NO MORE
      050.321  176            1896X          MOV     A,M             (A) = NEW BYTE
      050.322  345            1897X          PUSH    H
      050.323  305            1898X          PUSH    B               SAVE REGISTERS
      050.324  253            1899X          XRA     E
      050.325  107            1900X          MOV     B,A
      050.326  017            1901X          RRC
      050.327  017            1902X          RRC
      050.330  017            1903X          RRC
      050.331  017            1904X          RRC
      050.332  117            1905X          MOV     C,A
      050.333  250            1906X          XRA     B
      050.334  346 360        1907X          ANI     0F0H
      050.336  252            1908X          XRA     D
      050.337  157            1909X          MOV     L,A
      050.340  171            1910X          MOV     A,C
      050.341  007            1911X          RLC
      050.342  346 037        1912X          ANI     1FH
      050.344  255            1913X          XRA     L
      050.345  157            1914X          MOV     L,A
      050.346  170            1915X          MOV     A,B
      050.347  007            1916X          RLC
      050.350  346 001        1917X          ANI     1
```

```
050.352  252          1918X          XRA     D
050.353  255          1919X          XRA     L
050.354  127          1920X          MOV     D,A
050.355  171          1921X          MOV     A,C
050.356  346 360      1922X          ANI     0F0H
050.360  250          1923X          XRA     B
050.361  137          1924X          MOV     E,A
050.362  171          1925X          MOV     A,C
050.363  250          1926X          XRA     B
050.364  007          1927X          RLC
050.365  346 340      1928X          ANI     0E0H
050.367  253          1929X          XRA     E
050.370  137          1930X          MOV     E,A
050.371  301          1931X          POP     B
050.372  341          1932X          POP     H
050.373  043          1933X          INX     H
050.374  013          1934X          DCX     B
050.375  303 316 050  1935X          JMP     $BCRC
051.000               1936           XTEXT   ABR              INCLUDE HERE TO BE OVERLAID



                      1938X **       ABR - AUTO BAUD RATE SELECTION.
                      1939X *
                      1940X *        ABR READS CHARACTERS FROM THE SYSTEM CONSOLE ACE UNTIL
                      1941X *        THE CURRENT BAUD RATE IS DETERMINED.
                      1942X *
                      1943X *        ENTRY   NONE
                      1944X *        EXIT    (HL) = BAUD RATE DIVISOR
                      1945X *                ACE SETUP WITH BAUD RATE, NO INTERRUPTS
                      1946X *        USES    ALL
                      1947X
                      1948X
051.000               1949X ABR      EQU     *
                      1950X
                      1951X *        INITIALIZE LED DISPLAY FOR PROMPT
                      1952X
051.000  072 010 040  1953X ABR0.1   LDA     .MFLAG                                /79.01.GC/
051.003  365          1954X          PUSH    PSW                                   /79.01.GC/
051.004  366 002      1955X          ORI     UO.DDU                                /79.01.GC/
051.006  062 010 040  1956X          STA     .MFLAG                                /79.01.GC/
051.011  001 011 000  1957X          LXI     B,9
051.014  021 231 051  1958X          LXI     D,ABR.A
051.017  041 013 040  1959X          LXI     H,.ALEDS
051.022  315 252 030  1960X          CALL    $MOVE
051.025  021 013 040  1961X          LXI     D,.ALEDS
051.030  076 144      1962X          MVI     A,100
051.032  315 140 002  1963X          CALL    .HORN
                      1964X
051.035  041 242 051  1965X          LXI     H,TABLE
                      1966X
051.040  257          1967X ABR0.3   XRA     A                                     /79.01.GC/
051.041  323 351      1968X          OUT     SC.ACE+UR.IER                         /79.01.GC/
051.043  076 020      1969X          MVI     A,UC.LOO                              /79.01.GC/
051.045  323 354      1970X          OUT     SC.ACE+UR.MCR   SET LOOP BACK         /79.01.GC/
```

```
051.047  076 200      1971X         MVI     A,UC.DLA
051.051  323 353      1972X         OUT     SC.ACE+UR.LCR    LINE CONTROL ACCESS
051.053  176          1973X         MOV     A,M
051.054  043          1974X         INX     H
051.055  323 350      1975X         OUT     SC.ACE+UR.DLL    DIVISOR LEAST SIGNIFICANT
051.057  176          1976X         MOV     A,M
051.060  346 177      1977X         ANI     177Q             CLEAR STOP BITS FLAG
051.062  323 351      1978X         OUT     SC.ACE+UR.DLM    DIVISOR MOST SIGNIFICANT
051.064  276          1979X         CMP     M                SEE IF 2 STOP BITS
051.065  043          1980X         INX     H
051.066  076 003      1981X         MVI     A,UC.8BW         ASSUME 8 BIT WORDS, 1 STOP
051.070  312 075 051  1982X         JE      ABR0.5
051.073  076 007      1983X         MVI     A,UC.8BW+UC.2SB  SET 2 STOP BITS
051.075  323 353      1984X ABR0.5  OUT     SC.ACE+UR.LCR    LINE CONTROL ACCESS
051.077  076 156      1985X         MVI     A,AC.DLY                                      /79.01.GC/
051.101  315 053 000  1986X         CALL    .DLY             WAIT FOR 8250 TO SETTLE      /79.01.GC/
051.104  333 354      1987X         IN      SC.ACE+UR.MCR                                 /79.01.GC/
051.106  346 357      1988X         ANI     377Q-UC.LOO                                   /79.01.GC/
051.110  323 354      1989X         OUT     SC.ACE+UR.MCR    TURN OFF LOOP                /79.01.GC/
                      1990X
                      1991X *       WAIT FOR CHARACTER TO BE HIT
                      1992X
051.112  333 350      1993X ABR0    IN      SC.ACE+UR.RBR    GOBBLE OVERRUN
051.114  333 355      1994X ABR1    IN      SC.ACE+UR.LSR
000.000               1995X         ERRNZ   UC.OR-2
051.116  037          1996X         RAR
051.117  037          1997X         RAR
051.120  332 112 051  1998X         JC      ABR0             OVERRUN
051.123  027          1999X         RAL
051.124  027          2000X         RAL
051.125  346 015      2001X         ANI     UC.DR+UC.PE+UC.FE
051.127  312 114 051  2002X         JZ      ABR1             NOTHING YET
051.132  365          2003X         PUSH    PSW
051.133  032          2004X         LDAX    D                ECHO ' ' AS '.' ON LEDS
051.134  346 177      2005X         ANI     01111111B        TURN ON '.'
051.136  022          2006X         STAX    D
051.137  023          2007X         INX     D
051.140  361          2008X         POP     PSW
051.141  346 010      2009X         ANI     UC.FE
051.143  302 163 051  2010X         JNZ     ABR3             USER IS SLOWER THAN THIS
051.146  333 350      2011X         IN      SC.ACE+UR.RBR    GET DATA
051.150  346 177      2012X         ANI     177Q             TRIM
051.152  376 040      2013X         CPI     ' '
051.154  312 200 051  2014X         JE      ABR5
                      2015X
                      2016X *       USER IS FASTER THAN WE ARE. FOLLOW FASTER LINKAGE
                      2017X
051.157  156          2018X ABR2    MOV     L,M              FOLLOW LINK
051.160  303 040 051  2019X         JMP     ABR0.3           TRY AGAIN
                      2020X
                      2021X *       USER IS SLOWER THAN WE ARE. READ NEXT CHARACTER
                      2022X
051.163  076 067      2023X ABR3    MVI     A,110/2
051.165  315 053 000  2024X         CALL    .DLY             WAIT FOR THINGS TO SETTLE OUT
051.170  333 350      2025X         IN      SC.ACE+UR.RBR
051.172  333 355      2026X         IN      SC.ACE+UR.LSR
```

```
051.174  043          2027X          INX      H
051.175  303 157 051  2028X          JMP      ABR2
                      2029X
                      2030X *        FOUND THE BAUD RATE. RETURN WITH ANSWERS
                      2031X
051.200  021 013 040  2032X ABR5     LXI      D,.ALEDS                        BLANK DISPLAY
051.203  006 011      2033X          MVI      B,9
051.205  076 377      2034X          MVI      A,377A
051.207  022          2035X ABR5.1   STAX     D
051.210  023          2036X          INX      D
051.211  005          2037X          DCR      B
051.212  302 207 051  2038X          JNZ      ABR5.1
051.215  053          2039X          DCX      H
051.216  126          2040X          MOV      D,M
051.217  053          2041X          DCX      H
051.220  136          2042X          MOV      E,M
051.221  353          2043X          XCHG                (HL) = BAUD RATE
051.222  333 350      2044X          IN       SC.ACE+UR.RBR  GOBBLE THE GARBAGE           /79.01.GC/
051.224  361          2045X          POP      PSW                                         /79.01.GC/
051.225  062 010 040  2046X          STA      .MFLAG                                      /79.01.GC/
051.230  311          2047X          RET
                      2048X
051.231  244 230 220  2049X ABR.A    DB       244Q,230Q,220Q,215Q,214Q,377Q,377Q,377Q,377Q  'SPACE' FOR LEDS
                      2050X



                      2052X **       BAUD RATE SELECTION TREE.
                      2053X *
051.242               2054X TABLE    DS       0              START OF BAUD TABLE
                      2055X
051.242  060 000      2056X          DW       000060A        2400 BAUD
051.244  246          2057X          DB       #T9600         USER IS FASTER
051.245  252          2058X          DB       #T600          USER IS SLOWER
                      2059X
                      2060X *        2ND TRY GROUPS
                      2061X
051.246  014 000      2062X T9600    DW       000014A        9600 BAUD
051.250  256          2063X          DB       #T19200        USER IS FASTER
051.251  262          2064X          DB       #T4800         USER IS SLOWER
                      2065X
051.252  300 000      2066X T600     DW       000300A        600 BAUD
051.254  266          2067X          DB       #T1200         USER IS FASTER
051.255  272          2068X          DB       #T300          USER IS SLOWER
                      2069X
                      2070X *        3RD TRY GROUPS
                      2071X
051.256  006 000      2072X T19200   DW       000006A        19200 BAUD
051.260  242          2073X          DB       #TABLE         USER IS FASTER, MUST BE SCREWED UP
051.261  242          2074X          DB       #TABLE         USER IS SLOWER, MUST BE SCREWED UP
                      2075X
051.262  030 000      2076X T4800    DW       000030A        4800 BAUD
051.264  276          2077X          DB       #T7200         USER IS FASTER
051.265  302          2078X          DB       #T3600         USER IS SLOWER
                      2079X
```

```
051.266  140 000      2080X T1200   DW      000140A         1200 BAUD
051.270  306          2081X         DB      #T1800          USER IS FASTER
051.271  242          2082X         DB      #TABLE          USER IS SLOWER, MUST BE SCREWED UP
                      2083X
051.272  200 001      2084X T300    DW      001200A         300 BAUD
051.274  242          2085X         DB      #TABLE          USER IS FASTER, MUST BE SCREWED UP
051.275  312          2086X         DB      #T110           USER IS SLOWER
                      2087X
                      2088X *       4TH TRY GROUPS
                      2089X
051.276  020 000      2090X T7200   DW      000020A         7200 BAUD
051.300  242          2091X         DB      #TABLE          USER IS FASTER, MUST BE SCREWED UP
051.301  242          2092X         DB      #TABLE          USER IS SLOWER, MUST BE SCREWED UP
                      2093X
051.302  040 000      2094X T3600   DW      000040A         3600 BAUD
051.304  242          2095X         DB      #TABLE          USER IS FASTER, MUST BE SCREWED UP
051.305  242          2096X         DB      #TABLE          USER IS SLOWER, MUST BE SCREWED UP
                      2097X
051.306  100 000      2098X T1800   DW      000100A         1800 BAUD
051.310  242          2099X         DB      #TABLE          USER IS FASTER, MUST BE SCREWED UP
051.311  242          2100X         DB      #TABLE          USER IS SLOWER, MUST BE SCREWED UP
                      2101X
051.312  027 204      2102X T110    DW      204027A         110 BAUD
051.314  316          2103X         DB      #T150           USER IS FASTER
051.315  322          2104X         DB      #T75            USER IS SLOWER
                      2105X
                      2106X *       5TH TRY GROUPS
                      2107X
051.316  000 003      2108X T150    DW      003000A         150 BAUD
051.320  242          2109X         DB      #TABLE          USER IS FASTER, MUST BE SCREWED UP
051.321  242          2110X         DB      #TABLE          USER IS SLOWER, MUST BE SCREWED UP
                      2111X
051.322  000 006      2112X T75     DW      006000A         75 BAUD
051.324  242          2113X         DB      #TABLE          USER IS FASTER, MUST BE SCREWED UP
051.325  242          2114X         DB      #TABLE          USER IS SLOWER, MUST BE SCREWED UP
                      2115X
000.051               2116X .       SET     */256
000.000               2117X         ERRNZ   TABLE/256-.     MUST BE IN SAME PAGE
051.326               2118          XTEXT   HLCPDE          Make sure only ABR uses this!
                      2119X **      HLCPDE  - (HL) COMPARED TO (DE)
                      2120X *
                      2121X *       THIS ROUTINE IS DOUBLE WORD COMPARE OF REGISTER PAIRS (DE) AND (HL).
                      2122X *
                      2123X *       ENTRY:  (HL)&(DE) SET UP
                      2124X *
                      2125X *       EXIT:   (PSW)    =
                      2126X *                                'Z' SET   IF (HL) =  (DE)
                      2127X *                                'C' SET   IF (HL) <  (DE)
                      2128X *                                'C' CLEAR IF (HL) >= (DE)
                      2129X *
                      2130X *
                      2131X *       USES:   (PSW)
                      2132X *
                      2133X
051.326  174          2134X HLCPDE  MOV     A,H
051.327  272          2135X         CMP     D       'C' SET => (A) < (D)
```

```
051.330  300          2136X          RNZ
051.331  175          2137X          MOV     A,L
051.332  273          2138X          CMP     E          'C' SET => (L) < (E)
051.333  311          2139X          RET
051.334               2140           XTEXT   SCU


                      2142X **       SCU - SETUP CONSOLE USART.
                      2143X *
                      2144X *        SCU CONFIGURES THE CONSOLE USART.
                      2145X *
                      2146X *        IF  8250
                      2147X *          THEN  PORT = 372-3Q
                      2148X *          ELSE  PORT = 340-7Q
                      2149X *
                      2150X *
                      2151X *        ENTRY   NONE
                      2152X *        EXIT    NONE
                      2153X *        USES    A,F,(BC),(HL)
                      2154X
                      2155X
051.334  072 343 040  2156X SCU      LDA     S.CDB
051.337  376 001      2157X          CPI     CDB.H84
051.341  312 004 052  2158X          JZ      SCU1                     IF  8250
                      2159X
                      2160X *        PRESET 8251
                      2161X
051.344  076 201      2162X          MVI     A,201Q
051.346  323 373      2163X          OUT     SC.UART+USR              GET USART IN KNOWN STATE
051.350  323 373      2164X          OUT     SC.UART+USR
051.352  323 373      2165X          OUT     SC.UART+USR
051.354  323 373      2166X          OUT     SC.UART+USR
051.356  076 100      2167X          MVI     A,UCI.IR                 RESET
051.360  323 373      2168X          OUT     SC.UART+USR
051.362  072 327 040  2169X          LDA     S.CONTY
051.365  346 010      2170X          ANI     CTP.2SB
000.000               2171X          ERRNZ   CTP.2SB*16+UMI.1B-UMI.2B
051.367  007          2172X          RLC
051.370  007          2173X          RLC
051.371  007          2174X          RLC
051.372  007          2175X          RLC
051.373  366 116      2176X          ORI     UMI.1B+UMI.L8+UMI.16X
051.375  323 373      2177X          OUT     SC.UART+USR
051.377  076 025      2178X          MVI     A,UCI.ER+UCI.RE+UCI.TE
052.001  323 373      2179X          OUT     SC.UART+USR
052.003  311          2180X          RET
                      2181X
                      2182X *        IS 8250
                      2183X
052.004  333 355      2184X SCU1     IN      SC.ACE+UR.LSR                                    /80.01.GC/
052.006  346 100      2185X          ANI     UC.TSE          CHECK FOR SHIFT EMPTY            /80.01.GC/
052.010  312 004 052  2186X          JZ      SCU1                                             /80.01.GC/
                      2187X
052.013  257          2188X          XRA     A                                                /79.01.GC/
```

```
052.014  323 351      2189X       OUT     SC.ACE+UR.IER    TURN OFF ANY INTERRUPTS         /79.01.GC/
052.016  076 020      2190X       MVI     A,UC.LOO                                         /79.01.GC/
052.020  323 354      2191X       OUT     SC.ACE+UR.MCR                                    /79.01.GC/
052.022  052 344 040  2192X       LHLD    S.BAUD
052.025  076 200      2193X       MVI     A,UC.DLA
052.027  323 353      2194X       OUT     SC.ACE+UR.LCR    ACCESS DIVISOR LATCHES
052.031  175          2195X       MOV     A,L
052.032  323 350      2196X       OUT     SC.ACE+UR.DLL    SET LEAST SIGNIFICANT
052.034  174          2197X       MOV     A,H
052.035  346 177      2198X       ANI     177Q             TRIM STOP BITS
052.037  323 351      2199X       OUT     SC.ACE+UR.DLM    SET MOST SIGNIFICANT
052.041  072 327 040  2200X       LDA     S.CONTY
052.044  346 010      2201X       ANI     CTP.2SB
052.046  017          2202X       RRC
000.000               2203X       ERRNZ   CTP.2SB/2-UC.2SB
000.000               2204X       ERRNZ   UC.2SB-4         (A) = UC.2SB IF 2 STOP BITS
052.047  366 003      2205X       ORI     UC.8BW           8 BIT WORDS
052.051  323 353      2206X       OUT     SC.ACE+UR.LCR
052.053  076 156      2207X       MVI     A,AC.DLY                                         /79.01.GC/
052.055  315 053 000  2208X       CALL    .DLY                                             /79.01.GC/
052.060  333 350      2209X       IN      SC.ACE+UR.RBR    GOBBLE ANY TRASH                /79.01.GC/
052.062  333 354      2210X       IN      SC.ACE+UR.MCR                                    /79.01.GC/
052.064  346 357      2211X       ANI     377Q-UC.LOO                                      /79.01.GC/
052.066  323 354      2212X       OUT     SC.ACE+UR.MCR                                    /79.01.GC/
052.070  311          2213X       RET
052.071               2214        XTEXT   MCU              INCLUDED HERE TO BE USED AT BOOT-UP



                      2216X **    MCU - MAP LOWER CASE TO UPPER CASE.
                      2217X *
                      2218X *     MCU MAPS A LOWER CASE ALPHABETIC TO UPPER
                      2219X *     CASE.
                      2220X *
                      2221X *     ENTRY   (A) = CHARACTER
                      2222X *     EXIT    (A) = CHARACTER RESULT
                      2223X *     USES    A,F
                      2224X
                      2225X
052.071  376 141      2226X $MCU  CPI     'a'
052.073  330          2227X       RC                       NOT LOWER CASE
052.074  376 173      2228X       CPI     'z'+1
052.076  320          2229X       RNC                      NOT LOWER CASE
052.077  326 040      2230X       SUI     'a'-'A'
052.101  311          2231X       RET
```

```
                         **      THIS ROUTINE IS OVERLAID BY THE HDOS.SYS PRORGRAM DURING BOOT.

                         **      CDC - COMPUTE DISK CHECKSUMS.
                         *
                         *       CDC READS EACH DISK SECTOR (EXCEPT FOR TRACK 0) TO SEE IF
                         *       THERE ARE ANY PROBLEMS.
                         *
                         *       THE CHECKSUM OF EACH SECTOR IS PRINTED, TOGETHER
                         *       WITH THE TOTAL CHECKSUM FOR EACH TRACK, AND THE VOLUME CHECKSUM.
                         *


052.102  041 000 000     CDC     LXI     H,0
052.105  042 175 053             SHLD    CDCA            VOLUME
052.110  042 177 053             SHLD    CDCB            TRACK
052.113  044                     INR     H
052.114  042 203 053             SHLD    CDCE            SET TRACK AND SECTOR NUMBERS
052.117  052 324 053             LHLD    BLABEL+LAB.SPT                          /80.09.sc/
052.122  046 000                 MVI     H,0             HL = Track 1, sector 0  /80.09.sc/
052.124  042 201 053             SHLD    CDCD            SET FIRST SECTOR NUMBER

                         *       READ TRACK

052.127  021 205 054     CDC1    LXI     D,BUFF                                  /80.09.sc/
052.132  072 324 053             LDA     BLABEL+LAB.SPT                          /80.09.sc/
052.135  107                     MOV     B,A                                     /80.09.sc/
052.136  016 000                 MVI     C,0             BC = Byte Count         /80.09.sc/
052.140  076 000                 MVI     A,DC.REA
052.142  315 130 040             CALL    SYDD            READ DISK
052.145  322 174 052             JNC     CDC2            NO ERROR

                         *       READ ERROR

052.150  315 027 047             CALL    $TYPET
052.153  007 077 060             DB      BELL,'?00 * ERROR * ',BELL,200Q

                         *       CRC SECTOR

052.174  315 027 047     CDC2    CALL    $TYPET
052.177  123 105 103             DB      'SECTOR',' '+200Q
052.206  052 201 053             LHLD    CDCD
052.211  353                     XCHG
052.212  076 004                 MVI     A,4             !!!! Big disks !!!!     /80.09.sc/
052.214  315 103 053             CALL    TTDD
052.217  315 027 047             CALL    $TYPET
052.222  040 075 240             DB      ' =',' '+200Q
052.225  052 177 053             LHLD    CDCB
052.230  072 203 053             LDA     CDCE
052.233  315 063 053             CALL    CSC             COMPUTE TRACK CRC
052.236  042 177 053             SHLD    CDCB
052.241  052 175 053             LHLD    CDCA
052.244  072 203 053             LDA     CDCE
052.247  315 063 053             CALL    CSC             COMPUTE VOLUME CRC
052.252  042 175 053             SHLD    CDCA
052.255  041 000 000             LXI     H,0
052.260  072 203 053             LDA     CDCE
```

```
052.263  315 063 053  2290          CALL    CSC               COMPUTE SECTOR CRC
052.266  353          2291          XCHG
052.267  076 005      2292          MVI     A,5
052.271  315 164 053  2293          CALL    TTDDCR            TYPE SECTOR CHECKSUM
052.274  052 201 053  2294          LHLD    CDCD              (HL) = SECTOR COUNT
052.277  043          2295          INX     H
052.300  042 201 053  2296          SHLD    CDCD
052.303  041 203 053  2297          LXI     H,CDCE
052.306  315 325 046  2298          CALL    $ICTT.            CHECK FOR CHARACTER
052.311  332 324 052  2299          JC      CDC3              NO CHARACTER WAS HIT
052.314  315 353 046  2300          CALL    $ICTT..           GET CHARACTER
052.317  376 003      2301          CPI     CTLC
052.321  312 054 053  2302          JE      CDC4              CTL-C HIT
052.324  064          2303  CDC3    INR     M                 COUNT SECTOR
052.325  072 324 053  2304          LDA     BLABEL+LAB.SPT                              /80.09.sc/
052.330  226          2305          SUB     M
052.331  302 174 052  2306          JNE     CDC2              MORE ON TRACK
                      2307
                      2308  *       HAVE COMPLETED TRACK
                      2309
052.334  167          2310          MOV     M,A
052.335  043          2311          INX     H
052.336  064          2312          INR     M                 COUNT TRACK
052.337  315 027 047  2313          CALL    $TYPET
052.342  124 122 101  2314          DB      'TRACK TOTAL ',200Q+' '
052.357  052 177 053  2315          LHLD    CDCB
052.362  353          2316          XCHG
052.363  041 000 000  2317          LXI     H,0
052.366  042 177 053  2318          SHLD    CDCB              RESET COUNT
052.371  076 005      2319          MVI     A,5
052.373  315 164 053  2320          CALL    TTDDCR
                      2321
052.376  052 221 053  2322          LHLD    BLABEL+LAB.SIZ                              /80.09.sc/
053.001  353          2323          XCHG                      DE = Volume Size          /80.09.sc/
053.002  052 201 053  2324          LHLD    CDCD              HL = Next Sector          /80.09.sc/
053.005  315 326 051  2325          CALL    HLCPDE                                      /80.09.sc/
053.010  332 127 052  2326          JC      CDC1              Not Done                  /80.09.GC/
                      2327
                      2328  *       ALL DONE
                      2329
053.013  315 027 047  2330          CALL    $TYPET
053.016  000 040 126  2331          DB      0,' VOLUME TOTAL CRC =',200Q+' '
053.043  052 175 053  2332          LHLD    CDCA
053.046  353          2333          XCHG
053.047  076 005      2334          MVI     A,5
053.051  303 164 053  2335          JMP     TTDDCR                                      /80.09.sc/
                      2336
                      2337  *       CTL-C STRUCK
                      2338
053.054  315 027 047  2339  CDC4    CALL    $TYPET
053.057  136 103 200  2340          DB      '^C',200Q
053.062  311          2341          RET                                                 /80.08.sc/
```

```
                           **      CSC - COMPUTE SECTOR CRC.
                           *
                           *       CSC IS CALLED TO COMPUTE THE CRC OVER A SECTOR'S WORTH OF
                           *       DATA.
                           *
                           *       ENTRY   (HL) = CURRENT CRC VALUE
                           *               BUFF = BUFFER WITH SECTORS OF DATA
                           *               (A) = INDEX (0 TO 9) OF SECTOR IN BUFF
                           *       EXIT    (HL) = UPDATED CRC VALUE
                           *       USES    ALL
                           *

053.063  353               CSC     XCHG              (DE) = CRC VALUE
053.064  041 205 054               LXI     H,BUFF
053.067  204                       ADD     H
053.070  147                       MOV     H,A               (HL) = ADDRESS OF SECTOR
053.071  001 000 001               LXI     B,256             (BC) = COUNT
053.074  315 316 050               CALL    $BCRC             BLOCK CRC IT
053.077  353                       XCHG                      (HL) = RESULTANT CRC
053.100  311                       RET


                           **      TTDD - TYPE DECIMAL DIGITS.
                           *
                           *       TTDD TYPES A 16 BIT VALUE AS 1 TO 5 DECIMAL DIGITS.
                           *
                           *       ENTRY   (D,E) = VALUE
                           *               (A) = DIGIT COUNT
                           *       EXIT    VALUE TYPED.
                           *       USES    A,B,C,F


053.101  076 005           TTDD.   MVI     A,5
053.103  345               TTDD    PUSH    H
053.104  365               TTDD1   PUSH    PSW
053.105  041 150 053               LXI     H,TTDDA-2
053.110  007                       RLC                       (A) = DIGIT NUMBER*2
053.111  315 101 030               CALL    $DADA.
053.114  176                       MOV     A,M
053.115  043                       INX     H
053.116  146                       MOV     H,M
053.117  157                       MOV     L,A               (HL) = MULTIPLE OF 10
053.120  353                       XCHG                      (DE) = DEVISOR, (HL) = VALUE
053.121  076 377                   MVI     A,377Q
053.123  031               TTDD2   DAD     D
053.124  074                       INR     A
053.125  332 123 053               JC      TTDD2             IF MORE TO GO
053.130  306 060                   ADI     '0'
053.132  315 064 047               CALL    $TYPEC.           TYPE DIGIT
053.135  175                       MOV     A,L
053.136  223                       SUB     E
053.137  137                       MOV     E,A               REMOVE EXTRA SUBTRACTION
053.140  174                       MOV     A,H
053.141  232                       SBB     D
```

```
053.142  127             2396           MOV     D,A
053.143  361             2397           POP     PSW
053.144  075             2398           DCR     A
053.145  302 104 053     2399           JNZ     TTDD1           IF MORE DIGITS
053.150  341             2400           POP     H
053.151  311             2401           RET                     EXIT
                         2402
053.152                  2403  TTDDA    EQU     *
053.152  377 377         2404           DW      -1
053.154  366 377         2405           DW      -10
053.156  234 377         2406           DW      -100
053.160  030 374         2407           DW      -1000
053.162  360 330         2408           DW      -10000



                         2410  **       TTDDCR - TYPE DECIMAL DIGITS, THEN CRLF.
                         2411  *
                         2412  *        ENTRY   SAME AS TTDD
                         2413  *        EXIT    SAME AS TD
                         2414  *        USES    SAME AS TTDD
                         2415
                         2416
053.164  315 103 053     2417  TTDDCR   CALL    TTDD
053.167  315 027 047     2418           CALL    $TYPET
053.172  200             2419           DB      200Q
053.173  311             2420           RET
```

```
      007.000                2423  SOBOOTL EQU    *-SOBOOT+255/256*256    SOBOOT LENGTH IN SECTORS *256
P     000.000                2424          ERRPL  SB.SDB-SB.BOO+SOBOOTL+255/256-DDF.BOL           /80.09.sc/
                             2425
      053.174  000           2426  ABF     DB     0               Auto Boot Flag: != 0 => Illegal /80.09.sc/
      053.175  000 000       2427  CDCA    DW     0               VOLUME CRC                      /80.09.sc/
      053.177  000 000       2428  CDCB    DW     0               TRACK CRC                       /80.09.sc/
      053.201  000 000       2429  CDCD    DW     0               SECTOR NUMBER                   /80.09.sc/
      053.203  000           2430  CDCE    DB     0               SECTOR NUMBER                   /80.09.sc/
      053.204  000           2431          DB     0               TRACK NUMBER                    /80.09.sc/
                             2432
      053.205                2433  BLABEL  EQU    *               LABEL BUFFER STARTS HERE
      054.205                2434  BUFF    EQU    BLABEL+256      10 SECTOR BUFFER
      066.205                2435  BOOEND  EQU    10*256+BUFF     BUFFER ENDS HERE
```

```
                               ***     INIT Vectors for Shared Routines
                               *

 000.173                               ERRMI   INITVEC-*

 054.000                               ORG     INITVEC

 000.000                               ERRNZ   *-$$VER
 054.000  040                          DB      VERS

 000.000                               ERRNZ   *-$$BITC
 054.001  303 212 066                  JMP     BITC

 000.000                               ERRNZ   *-$$BITS
 054.004  303 231 066                  JMP     BITS

 000.000                               ERRNZ   *-$$BUFF
 054.007  000 074                      DW      GRTAB

 000.000                               ERRNZ   *-$$CHL
 054.011  303 224 030                  JMP     $CHL

 000.000                               ERRNZ   *-$$CNO
 054.014  303 060 064                  JMP     CNO

 000.000                               ERRNZ   *-$$CYS
 054.017  303 076 064                  JMP     CYS

 000.000                               ERRNZ   *-$$DRVR
 054.022  303 241 064                  JMP     DRIVER

 000.000                               ERRNZ   *-$$DRVR.
 054.025  303 271 064                  JMP     DRIVER.

 000.000                               ERRNZ   *-$$ITL.
 054.030  303 334 067                  JMP     $ITL.

 000.000                               ERRNZ   *-$$MOVE
 054.033  303 252 030                  JMP     $MOVE

 000.000                               ERRNZ   *-$$TBRA
 054.036  303 076 031                  JMP     $TBRA

 000.000                               ERRNZ   *-$$TYPTX
 054.041  303 136 031                  JMP     $TYPTX

 000.000                               ERRNZ   *-$$VSN
 054.044  303 205 066                  JMP     VSN
 000.000                               ERRNZ   *-$$MAX
```

```
                      ***      INIT - MAIN INITIALIZE LOOP.
                      *

                               LOF       C                    RESTORE LISTING CONTROL

                      *        RESTART HERE TO INIT NEW DISK

054.047  061 200 042  RESTART  LXI       SP,STACK                                          /80.05.GC/
054.052  315 243 054           CALL      AMW                  ASK IF MORE WORK WANTED
054.055  302 200 054           JNZ       EXIT                 ALL DONE

054.060  061 200 042  INIT0    LXI       SP,STACK                                          /80.05.GC/
054.063  315 372 054           CALL      PDN                  Parse Device Name            /80.05.GC/
054.066  332 047 054           JC        RESTART                                           /80.05.GC/

054.071  061 200 042  INIT1    LXI       SP,STACK                                          /80.05.GC/
054.074  315 224 055           CALL      RMI                  REQUEST MEDIA INSERTION      /80.05.GC/
054.077  332 060 054           JC        INIT0                                             /80.05.GC/

054.102  315 251 056           CALL      AAL                  ASK ABOUT LABEL
054.105  302 071 054           JNZ       INIT1                Not OKAY to INIT diskette    /80.05.GC/

054.110  315 256 057           CALL      GVI                  GET VOLUME ID
054.113  315 133 060           CALL      IDS                  INIT DISK SURFACE
054.116  332 071 054           JC        INIT1                Error Initializing Disk      /80.09.sc/
054.121  315 244 060           CALL      GBL                  GET BAD SECTOR LIST

054.124  315 200 061           CALL      FOV                  FORMAT VOLUME
054.127  332 071 054           JC        INIT1                ERROR                        /80.05.GC/

054.132  315 136 031           CALL      $TYPTX                                            /80.05.GC/
054.135  012                   DB        NL                                                /80.05.GC/
054.136  040 104 151           DB        ' Disk Initialization complete.',ENL              /80.05.GC/
054.175  303 071 054           JMP       INIT1                                             /80.05.GC/

                      *        EXIT

054.200  257          EXIT     XRA       A
054.201  377 000      EXIT.    DB        SYSCALL,.EXIT   LET  *HDOS*  HANDLE THE EXIT PARAMETERS



                      **       ERROR - ERROR ENCOUNTERED.                                  /80.05.sc/

054.203  315 213 054  ERROR    CALL      ERROR1                                            /80.05.GC/
054.206  076 001               MVI       A,1                  Abort                        /80.05.sc/
054.210  303 201 054           JMP       EXIT.                                             /80.05.GC/

054.213  365          ERROR1   PUSH      PSW                                               /80.05.GC/
054.214  315 250 066           CALL      $CCO                 CLEAR CTL-O
054.217  315 136 031           CALL      $TYPTX
054.222  012 007 105           DB        NL,BELL,'ERROR - ',' '+200Q
054.235  361                   POP       PSW
054.236  046 012               MVI       H,NL
054.240  377 057               SCALL     .ERROR                                            /80.05.GC/
```

```
    054.242  311          2542          RET                                                     /80.05.GC/
```

```
                          2546  **       AMW - ASK FOR MORE WORK.                                    /80.05.GC/
                          2547  *
                          2548  *        AMW SEES IF THE USER WANTS TO CONTINUE THE PROCESS.
                          2549  *
                          2550  *        ENTRY   NONE
                          2551  *        EXIT    'Z' CLEAR IF  NO MORE WORK
                          2552  *                'Z' SET IF MORE WORK
                          2553  *        USES    ALL
                          2554
                          2555
054.243                   2556  AMW      EQU     *                                                   /80.05.GC/
                          2557
054.243  315 250 066      2558  AMW1     CALL    $CCO            CLEAR CTL-O
054.246  315 136 031      2559           CALL    $TYPTX
054.251  012              2560           DB      NL
054.252  104 157 040      2561           DB      'Do you have any more disks to initialize (YES/NO) <NO> '
054.341  277              2562           DB      '?'+200Q
054.342  315 334 067      2563           CALL    $ITL.
054.345  332 243 054      2564           JC      AMW1            CTL-D HIT                           /80.05.GC/
                          2565
054.350  176              2566           MOV     A,M             (A) = REPLY
054.351  247              2567           ANA     A
054.352  312 367 054      2568           JZ      AMW2            took the default
                          2569
054.355  315 076 064      2570           CALL    CYS             Check for Yes
054.360  310              2571           RE                      IS YES
                          2572
054.361  315 060 064      2573           CALL    CNO             Check for NO
054.364  302 243 054      2574           JNZ     AMW1            Ask Again
                          2575
054.367  366 001          2576  AMW2     ORI     1               ANSWER IS NO
054.371  311              2577           RET
```

```
                        **      PDN     - Parse Device Name                          /80.50.sc/
                        *
                        *       PDN parses the device name, assuming the default of SY0:.
                        *
                        *       ENTRY:  NONE
                        *
                        *       EXIT:   PSW     = 'C' SET   if CTL-D hit
                                                  'C' CLEAR if valid device input
                                                   CRNDEV = address of device entry
                                                   UNIT   = unit
                        *
                        *       USES:   ALL
                        *

054.372  315 250 066    PDN     CALL    $CCO                Clear CTL-O
054.375  315 136 031            CALL    $TYPTX
055.000  012                    DB      NL
055.001  104 145 166            DB      'Device<SY0:>?',' '+200Q

055.017  315 342 067            CALL    $ITL
055.022  330                    RC                          CTL-D hit

055.023  041 064 070            LXI     H,ITLA              HL = address of device specification
055.026  001 213 055    PDN.    LXI     B,PDNA              Decode area
055.031  021 216 055            LXI     D,PDNC              Default device
055.034  315 121 064            CALL    DDS
055.037  332 147 055            JC      PDN3

055.042  021 213 055            LXI     D,PDNA
055.045  041 205 070            LXI     H,DEVTAB

055.050  325            PDN1    PUSH    D
055.051  021 255 070            LXI     D,DEVTABE
055.054  315 326 051            CALL    HLCPDE
055.057  321                    POP     D
055.060  322 147 055            JNC     PDN3

055.063  016 002                MVI     C,2
055.065  325                    PUSH    D
055.066  345                    PUSH    H
055.067  315 060 030            CALL    $COMP
055.072  341                    POP     H
055.073  321                    POP     D                   Save the pointers
055.074  312 106 055            JZ      PDN2                Found Device

                        *       Device not found in this entry

055.077  001 012 000            LXI     B,DVT.ENL
055.102  011                    DAD     B
055.103  303 050 055            JMP     PDN1

                        *       Device entry found

055.106  042 043 070    PDN2    SHLD    CRNDEV
055.111  072 215 055            LDA     PDNB
055.114  062 063 070            STA     UNIT
```

```
055.117  107                  MOV     B,A              B  = unit number
055.120  315 234 030          CALL    $INDL
055.123  004 000              DW      DVT.ADR
055.125  353                  XCHG                     HL = device table pointer
055.126  315 344 066          CALL    $INDLB
055.131  010 000              DW      DEV.MNU
055.133  075                  DCR     A
055.134  270                  CMP     B
055.135  332 147 055          JC      PDN3             Unit Number is NOT valid

055.140  315 170 065          CALL    RPB              Relocate Primary Boot
055.143  315 362 064          CALL    IVP              Initialize Volume Parameters
055.146  311                  RET

                    *         Illegal Device Specification

055.147  315 136 031    PDN3  CALL    $TYPTX
055.152  012                  DB      NL
055.153  111 154 154          DB      'Illegal Device Specification'
055.207  212                  DB      ENL
055.210  303 372 054          JMP     PDN              get another device

055.213  130 130        PDNA  DB      'XX'
055.215  000            PDNB  DB      0
000.000                       ERRNZ   IOC.UNI-IOC.DEV-2        2 byte device
000.000                       ERRNZ   IOC.DIR-IOC.UNI-1        1 byte unit

055.216  123 131 060    PDNC  DB      'SY0',0,0,0      Default device
```

```
                         **      RMI - REQUEST MEDIA INSERTION.                           /80.05.sc/
                         *
                         *       RMI PROMPTS THE USR TO LOAD THE DESIRED MEDIA.
                         *
                         *       WHEN IN PLACE, RMI CHECKS THE HOLE TIMINGS
                         *
                         *       ENTRY   NONE
                         *
                         *       EXIT    PSW     = 'C' SET   if ERROR
                         *                         'C' CLEAR if NO ERROR
                         *
                         *       USES    ALL
                         *

055.224                  RMI     EQU     *

                         *       Stuff Device Descriptors

055.224  052 043 070             LHLD    CRNDEV
055.227  315 234 030             CALL    $INDL
055.232  000 000                 DW      DVT.NAM
055.234  353                     XCHG
055.235  042 341 055             SHLD    RMIA
055.240  072 063 070             LDA     UNIT
055.243  062 061 041             STA     AIO.UNI                                          /80.09.GC/
055.246  306 060                 ADI     '0'
055.250  062 343 055             STA     RMIB

                         *       Prompt for the new volume

055.253  315 250 066     RMI1    CALL    $CCO                                             /80.05.GC/
055.256  315 136 031             CALL    $TYPTX
055.261  012                     DB      NL
055.262  040 111 156             DB      ' Insert the volume you wish to initialize into '
055.341  123 131         RMIA    DB      'SY'
055.343  060 072 073     RMIB    DB      '0:;',NL
055.347  040 162 145             DB      ' remember, any data on this volume will be destroyed.',NL
056.035  012                     DB      NL
056.036  110 151 164             DB      'Hit RETURN when ready.',NL
056.065  122 145 141             DB      'Ready?',' '+200Q

056.074  315 342 067             CALL    $ITL
056.077  330                     RC                      Abort

056.100  315 320 065             CALL    SUBFUN          Check Media Validity
056.103  000                     DB      INI.CMV
056.104  320                     RNC                     Good

                         *       ERROR IN MEDIA FORMAT.

056.105  315 136 031             CALL    $TYPTX
056.110  012 007 040             DB      NL,BELL,' Wrong type of media, media inserted improperly, or'
056.175  012 040 155             DB      NL,' media damaged. Check it and try again.',ENL
056.246  303 253 055             JMP     RMI1                                             /80.05.GC/
```

```
                        **      AAL - ASK ABOUT LABEL.
                        *
                        *       AAL ATTEMPTS TO READ THE VOLUME LABEL RECORD. IF SUCCESSFUL, THE
                        *       VOLUME # AND LABEL ARE PRINTED. THE USER GETS A CHANCE TO
                        *       CHICKEN OUT.
                        *
                        *       ENTRY   NONE
                        *
                        *       EXIT:   PSW     = 'Z' SET   if OKAY to init diskette    /80.05.sc/
                        *                         'Z' CLEAR if NOT OKAY to init diskette /80.05.sc/
                        *
                        *       USES    ALL
                        *

056.251  315 241 064    AAL     CALL    DRIVER          RESET DISK                      /80.09.GC/
056.254  007                    DB      DC.ABT                                          /80.05.GC/

056.255  041 000 000            LXI     H,0                                             /80.09.sc/
056.260  315 062 065            CALL    MOUNT..         Set Volume 0                    /80.09.sc/

056.263  315 136 031            CALL    $TYPTX                                          /80.05.sc/
056.266  012 124 150            DB      NL,'The volume now in the drive ...',ENL
056.327  021 000 075            LXI     D,RGTAB         USE RGTAB FOR SCRATCH BUFFER
056.332  001 000 001            LXI     B,1000A
056.335  041 011 000            LXI     H,DDF.LAB
056.340  315 241 064            CALL    DRIVER          READ LABEL SECTOR               /80.05.GC/
056.343  002                    DB      DC.RER                                          /80.05.GC/
056.344  322 030 057            JNC     AAL1            OK

                        *       COULDNT READ IT

056.347  315 136 031            CALL    $TYPTX
056.352  101 160 160            DB      'Apparently has not been initialized before',ENL
057.025  303 114 057            JMP     AAL2

                        *       COULD READ IT. TYPE DISK NUMBER AND LABEL

057.030  315 136 031    AAL1    CALL    $TYPTX
057.033  151 163 040            DB      'is volume #',' '+200Q
057.047  072 000 075            LDA     RGTAB+LAB.SER
057.052  137                    MOV     E,A
057.053  026 000                MVI     D,0
057.055  076 003                MVI     A,3
057.057  315 352 067            CALL    $TDD            TYPE DECIMAL DIGITS
057.062  315 136 031            CALL    $TYPTX
057.065  012 040 114            DB      NL,' Label = ','"'+200Q
057.100  076 074                MVI     A,LAB.LBL
057.102  041 021 075            LXI     H,RGTAB+LAB.LAB
057.105  315 241 067            CALL    $TYPL.          TYPE LABEL
057.110  315 136 031            CALL    $TYPTX
057.113  242                    DB      '"'+200Q

                        *       GIVE CHANCE TO CHICKEN OUT

057.114  315 250 066    AAL2    CALL    $CCO            CLEAR CTL-O
057.117  315 136 031            CALL    $TYPTX
```

```
     057.122  012 124 171  2778            DB      NL,'Type NO to cancel, type YES to erase and initialize the disk. (YES/NO) ','?'+200
Q
     057.233  315 334 067  2779            CALL    $ITL.
     057.236  332 253 057  2780            JC      AAL3            CTL-D hit                       /80.05.sc/
                           2781
     057.241  315 076 064  2782            CALL    CYS                                             /80.05.GC/
     057.244  310          2783            RZ                      IS OKAY TO INIT                 /80.05.GC/
                           2784
     057.245  315 060 064  2785            CALL    CNO                                             /80.05.GC/
     057.250  302 114 057  2786            JNZ     AAL2            Not a valid response            /80.05.GC/
                           2787
     057.253  366 001      2788  AAL3      ORI     1                                               /80.05.GC/
     057.255  311          2789            RET                     Exit without OK                 /80.05.sc/
```

```
                      **      GVI - GET VOLUME ID.
                      *
                      *       GVI GETS A NEW VOLUME SERIAL NUMBER AND AN OPTIONAL VOLUME LABEL.
                      *
                      *       ENTRY   NONE
                      *       EXIT    LABEL+LAB.SER = VOLUME SERIAL #
                      *               LABEL+LAB.LAB = 60 CHARACTER VOLUME LABEL
                      *       USES    ALL


057.256               GVI     EQU     *
057.256  315 250 066          CALL    $CCO                CLEAR CTL-O
057.261  315 136 031          CALL    $TYPTX
057.264  012 105 156          DB      NL,'Enter a unique volume serial number from 1 to 255:',' '+200Q
057.350  315 326 066          CALL    $IDN                INPUT DECIMAL NUMBER
057.353  332 256 057          JC      GVI                 BAD RANGE
057.356  174                  MOV     A,H
057.357  267                  ORA     A
057.360  302 256 057          JNZ     GVI                 TOO LARGE
057.363  265                  ORA     L
057.364  312 256 057          JZ      GVI                 ZERO ILLEGAL
057.367  062 256 070          STA     LABEL+LAB.SER

                      *       GET LABEL

057.372  315 250 066  GVI1    CALL    $CCO                CLEAR CTL-O
057.375  315 136 031          CALL    $TYPTX
060.000  105 156 164          DB      'Enter a volume label of 60 characters or less',ENL
060.056  315 342 067          CALL    $ITL
060.061  353                  XCHG                        (DE) = LINE ADDRESS
060.062  041 277 070          LXI     H,LABEL+LAB.LAB (HL) = TO ADDRESS

                      *       COPY FROM LINE TO LABEL+LAB.LAB, BLANK FILL OUT TO 60 CHARACTERS

060.065  006 075              MVI     B,61
060.067  032          GVI2    LDAX    D
060.070  167                  MOV     M,A                 COPY
060.071  023                  INX     D
060.072  247                  ANA     A
060.073  312 123 060          JZ      GVI3                END OF LINE
060.076  043                  INX     H                   INCREMENT TO POINTER
060.077  005                  DCR     B
060.100  302 067 060          JNZ     GVI2                MORE TO TO
060.103  315 136 031          CALL    $TYPTX
060.106  124 157 157          DB      'Too Long.',ENL
060.120  303 372 057          JMP     GVI1

060.123  066 040      GVI3    MVI     M,' '               BLANK REST OF LABEL
060.125  043                  INX     H
060.126  005                  DCR     B
060.127  302 123 060          JNZ     GVI3
060.132  311                  RET
```

```
                          **      IDS - INITIALIZE DISK SURFACE.                          /80.05.sc/
                          *
                          *       IDS IS CALLED TO INITIALIZE THE DISK SURFACE WITH RECORD LABELS,
                          *       AND A TEST PATTERN.
                          *
                          *       ENTRY   NONE
                          *
                          *       EXIT    PSW      = 'C'    if ERROR
                          *                          'NC'   if NO Error
                          *
                          *       USES    ALL
                          *

060.133  315 201 060      IDS     CALL    IDS.
060.136  320                      RNC                       No Errors

060.137  315 136 031              CALL    $TYPTX
060.142  012 007 125              DB      NL,BELL,'Unable to format this disk',ENL
060.177  067                      STC                       Flag Error
060.200  311                      RET


060.201  315 241 064      IDS.    CALL    DRIVER
060.204  007                      DB      DC.ABT

060.205  315 320 065              CALL    SUBFUN
060.210  001                      DB      INI.IDS
060.211  330                      RC

060.212  315 241 064              CALL    DRIVER            Abort disk
060.215  007                      DB      DC.ABT
060.216  330                      RC

060.217  315 367 065              CALL    SVP

                          *       WRITE DUMY BOOTSTRAP

060.222  315 055 065              CALL    MOUNT.            Set Volume 0
060.225  330                      RC

060.226  001 000 011              LXI     B,SOBOOTL+SB.SDB-SB.BOO
060.231  021 200 042              LXI     D,SB.BOO
060.234  041 000 000              LXI     H,0
060.237  315 271 064              CALL    DRIVER.           Write Boot
060.242  001                      DB      DC.WRI
060.243  311                      RET
```

```
                            **      GBL - GET BAD SECTOR LIST.
                            *
                            *       GBL GETS A LIST OF BAD SECTORS, AND FLAGS THE BYTES IN THE RGTAB.
                            *
                            *       ENTRY   NONE
                            *       EXIT    RGTAB SETUP
                            *       USES    ALL


060.244                     GBL     EQU     *

060.244  041 000 075                LXI     H,RGTAB
060.247  072 055 070                LDA     NGROUPS         A  = Number of Valid groups      /80.05.sc/
060.252  066 001            GBL1    MVI     M,1             FLAG UNUSED
060.254  043                        INX     H
060.255  075                        DCR     A
060.256  302 252 060                JNZ     GBL1            ZERO TABLE

060.261  175                GBL1.5  MOV     A,L                                              /80.05.GC/
060.262  247                        ANA     A                                                /80.05.GC/
060.263  312 274 060                JZ      GBL2            This assumes RGTAB is 256 bytes  /80.05.sc/
060.266  066 377                    MVI     M,377Q
060.270  043                        INX     H
060.271  303 261 060                JMP     GBL1.5                                           /80.05.GC/

060.274                     GBL2    EQU     *                                                /80.05.GC/

000.001                             IF      1                                                /80.05.GC/
                                    ANA     A               Clear CARRY                      /80.05.sc/

                                    JNC     GBL4            DONT ALLOW THIS QUESTION
                                    CALL    $CCO            CLEAR CTL-O
                                    CALL    $TYPTX
                                    DB      NL,'RESTRICT TO 35 TRACKS (YES/NO) <NO> ?',' '+200Q
                                    CALL    $ITL
                                    MOV     A,M
                                    ANA     A
                                    JZ      GBL4            IS DEFAULT <NO>
                                    CPI     'N'
                                    JE      GBL4            IS NO
                                    CPI     CR
                                    JE      GBL4            IS NO
                                    CPI     'Y'
                                    JNE     GBL2            TRY AGAIN

                            *       RESTRICT TO 35, FLAG LAST 5 BAD

                                    LXI     H,35*10/2+RGTAB
                                    MVI     A,5*10/2
                            GBL3    MVI     M,-1
                                    INX     H
                                    DCR     A
                                    JNZ     GBL3            FLAG BAD
                                    ERRNZ   *-GBL4                                           /80.05.GC/
                                    ENDIF                                                    /80.05.GC/

```

```
                             *       GET BAD SECTOR LIST

060.274  315 250 066         GBL4    CALL    $CCO              CLEAR CTL-O
060.277  315 136 031                 CALL    $TYPTX
060.302  012 105 156                 DB      NL,'Enter the numbers of the bad sectors one at a time.'
060.366  040 040 110                 DB      '  Hit RETURN',NL
061.003  141 146 164                 DB      'after each entry, and when finished.',ENL

061.050  315 250 066         GBL5    CALL    $CCO              CLEAR CTL-O
061.053  315 136 031                 CALL    $TYPTX
061.056  123 145 143                 DB      'Sector?',' '+200Q
061.066  315 326 066                 CALL    $IDN              INPUT DECIMAL NUBER
061.071  332 050 061                 JC      GBL5              Some error getting number       /80.05.sc/
061.074  353                         XCHG                      DE = sector number              /80.05.sc/
061.075  052 272 070                 LHLD    LABEL+LAB.SIZ                                     /80.05.sc/
061.100  053                         DCX     H                 HL = Maximum Sector Number      /80.05.sc/
061.101  315 326 051                 CALL    HLCPDE                                            /80.05.sc/
061.104  322 145 061                 JNC     GBL7              HL >= DE                        /80.05.sc/

061.107  315 136 031         GBL6    CALL    $TYPTX                                            /80.05.sc/
061.112  040 040 111                 DB      '  Illegal Sector Number',ENL
061.142  303 050 061                 JMP     GBL5

061.145  172                 GBL7    MOV     A,D
061.146  263                         ORA     E
061.147  310                         RZ                        CARRIAGE RETURN

061.150  041 365 377                 LXI     H,-11
061.153  031                         DAD     D
061.154  322 107 061                 JNC     GBL6              BAD NUMBER

                             *       FLAG SECTOR BAD

061.157  072 265 070                 LDA     LABEL+LAB.SPG     A  = Sectors/group              /80.05.sc/
061.162  102                         MOV     B,D                                               /80.05.GC/
061.163  113                         MOV     C,E               BC = Sector Number              /80.05.sc/
061.164  315 354 064                 CALL    DU68              HL = BC/A                       /80.05.sc/

061.167  021 000 075                 LXI     D,RGTAB
061.172  031                         DAD     D
061.173  066 377                     MVI     M,-1              FLAG BAD
061.175  303 050 061                 JMP     GBL5              GET MORE
```

```
                            **      FOV - FORMAT VOLUME.
                            *
                            *       FOV FORMATS THE NEW VOLUME.
                            *
                            *       IT ALLOCATES SPACE FOR THE DIRECTORY, BUILDS THE
                            *       DIRECTORY STRUCTURE, AND BUILDS THE
                            *       RGT AND THE GRT.
                            *
                            *       EXIT:   PSW     = 'C' SET   if ERROR                    /80.05.sc/
                                                      'C' CLEAR if No ERROR                 /80.05.sc/
                            *

061.200                     FOV     EQU     *

                            *       FLAG (IN THE RGT) THE LOW SECTORS RESERVED
                            *       FOR THE SYSTEM.

061.200  041 000 000                LXI     H,0
061.203  042 000 075                SHLD    RGTAB           FIRST TWO BLOCKS ARE SPECIAL

061.206  072 056 070                LDA     NSPCGRP         A  = Number of special groups
061.211  075                        DCR     A
061.212  075                        DCR     A               First 2 already taken care of
061.213  041 002 075                LXI     H,RGTAB+2
061.216  312 230 061        FOV1    JZ      FOV2            All done
061.221  066 377                    MVI     M,377Q          Flag Reserved
061.223  043                        INX     H
061.224  075                        DCR     A
061.225  302 216 061                JNZ     FOV1

061.230  315 332 061        FOV2    CALL    ADB             ASSIGN DIRECTORY BLOCKS
061.233  330                        RC                      Error                       /80.05.GC/

                            *       WRITE THE RGT

061.234  315 042 065                CALL    MOUNT                                       /80.05.GC/

061.237  001 000 001                LXI     B,256                                       /80.05.GC/
061.242  021 000 075                LXI     D,RGTAB
061.245  052 270 070                LHLD    LABEL+LAB.RGT   HL = RGT address            /80.06.sc/
061.250  315 271 064                CALL    DRIVER.         WRITE IT                    /80.05.GC/
061.253  001                        DB      DC.WRI                                      /80.05.GC/

                            *       WRITE THE LABEL SECTOR

061.254  052 310 040                LHLD    S.DATC
061.257  042 257 070                SHLD    LABEL+LAB.IND

061.262  315 055 065                CALL    MOUNT.          Set Volume to Zero          /80.05.sc/

061.265  001 000 001                LXI     B,256                                       /80.05.sc/
061.270  021 256 070                LXI     D,LABEL
061.273  041 011 000                LXI     H,DDF.LAB
061.276  315 271 064                CALL    DRIVER.         WRITE LABEL                 /80.05.GC/
061.301  001                        DB      DC.WRI                                      /80.05.GC/

```

```
                      *       WRITE THE GRT

061.302  315 213 062          CALL    BGT             BUILD THE GRT FIRST
061.305  315 042 065          CALL    MOUNT                                           /80.05.GC/

061.310  001 000 001          LXI     B,256
061.313  021 000 074          LXI     D,GRTAB
061.316  052 263 070          LHLD    LABEL+LAB.GRT   (HL) = GRT SECTOR INDEX
061.321  315 271 064          CALL    DRIVER.         WRITE IT                        /80.05.GC/
061.324  001                  DB      DC.WRI                                          /80.05.GC/

                      *       INITIALIZE THE DIRECTORY

061.325  315 341 062          CALL    IDD             INITIALIZE DEVICE DIRECTORY
061.330  247                  ANA     A               Clear ERROR                     /80.05.sc/
061.331  311                  RET
```

```
                               3068  **      ADB - ASSIGN DIRECTORY BLOCKS.                         /80.06.sc/
                               3069  *
                               3070  *       ADB LOCATES CONTIGUOUS SECTORS TO HOLD THE DIRECTORY
                               3071  *       AND THE GRT TABLE.
                               3072  *
                               3073  *       ENTRY   NONE
                               3074  *       EXIT    PSW      = 'C' SET   if ERROR
                               3075  *                          'C' CLEAR if NO ERROR
                               3076  *
                               3077  *               GRTBLK = BLOCK INDEX OF GRT
                               3078  *               DIRBLK = BLOCK INDEX OF DIRECTORY FIRST BLOCK
                               3079  *               LABEL+LAB.GRT = GRT SECTOR INDEX
                               3080  *               LABEL+LAB.DIS = DIRECTORY START SECTOR INDEX (NOT 1ST DIR SECTOR!)
                               3081  *       USES    ALL
                               3082
                               3083
       061.332                 3084  ADB     EQU     *
                               3085
                               3086  *       Compute Directory File Parameters
                               3087
       061.332  072 265 070    3088          LDA     LABEL+LAB.SPG
       061.335  021 013 000    3089          LXI     D,22/2
       000.000                 3090          ERRNZ   DIS.ENL-DIS.ENT/DIRELEN-22
       061.340  315 007 031    3091          CALL    $MU86
       061.343  353            3092          XCHG                      DE = direct. entries per volume block
                               3093
       061.344  072 055 070    3094          LDA     NGROUPS
       061.347  052 056 070    3095          LHLD    NSPCGRP
       061.352  225            3096          SUB     L                 Don't count special groups in max. num. files
       061.353  117            3097          MOV     C,A
       061.354  006 000        3098          MVI     B,0               BC = number of groups = max. num. files
       061.356  305            3099          PUSH    B
       061.357  315 106 030    3100          CALL    $DU66             HL = number of dir. blocks = DE / SPG
       061.362  301            3101          POP     B
                               3102
       061.363  175            3103          MOV     A,L
       061.364  074            3104          INR     A                 Add one to round up
       061.365  074            3105          INR     A                 Add one for GRT
       061.366  062 054 070    3106          STA     NDIRBLK
                               3107
                               3108  *       Find contiguous blocks for directory
                               3109
       061.371  021 003 000    3110          LXI     D,3
       061.374  315 106 030    3111          CALL    $DU66             HL = groups/3
       061.377  021 000 075    3112          LXI     D,RGTAB
       062.002  031            3113          DAD     D                 HL = address to start looking at
                               3114
       062.003  016 001        3115          MVI     C,1               Flag 0 passes through entire GRT
                               3116
       062.005  072 054 070    3117  ADB1    LDA     NDIRBLK
       062.010  107            3118          MOV     B,A               B  = number of directory blocks sought
       062.011  042 211 062    3119          SHLD    ADBA              Save initial Table pointer
                               3120
       062.014  176            3121  ADB2    MOV     A,M               A  = RESERVATION BYTE
       062.015  043            3122          INX     H
       062.016  075            3123          DCR     A
```

```
062.017  302 031 062             JNZ      ADB3             RESERVED
062.022  005                     DCR      B                COUNT GOT ONE
062.023  302 014 062             JNZ      ADB2             NEED MORE
062.026  303 137 062             JMP      ADB4             GOT OUR BLOCKS

                         *       RAN INTO A BAD BLOCK, START LOOKING OVER AGAIN

062.031  345             ADB3    PUSH     H
062.032  072 055 070             LDA      NGROUPS          A  = max. group number + 1
062.035  052 054 070             LHLD     NDIRBLK
062.040  225                     SUB      L                A  = A - NDIRBLK
062.041  341                     POP      H

062.042  225                     SUB      L
062.043  322 005 062             JNC      ADB1             STILL ENOUGH FOR A CHANCE

062.046  041 002 075             LXI      H,RGTAB+2
062.051  015                     DCR      C
062.052  362 005 062             JP       ADB1             Start from the beginning this time

062.055  315 136 031             CALL     $TYPTX           Unable to find enough in a row
062.060  007 040 126             DB       BELL,' Volume too decrepid for use. Try another.',BELL,ENL
062.135  067                     STC                       Flag problems
062.136  311                     RET

                         *       GOT THE TRACKS. (HL) = INDEX OF FIRST GRT+11

062.137  053             ADB4    DCX      H                (HL) = INDEX IF LAST GRT
062.140  175                     MOV      A,L
062.141  062 047 070             STA      GRTBLK           use last block for GRT

062.144  046 000                 MVI      H,0
062.146  315 044 064             CALL     BTS              HL = sector number
062.151  042 263 070             SHLD     LABEL+LAB.GRT    SET GRT ADDRESS

062.154  072 054 070             LDA      NDIRBLK
062.157  075                     DCR      A                Count the GRT block
062.160  062 054 070             STA      NDIRBLK

062.163  052 211 062             LHLD     ADBA             L  = block number for first directory block

                         *       SETUP POINTERS FOR LABEL AND INIT PROGRAM.

062.166  175                     MOV      A,L              (A) = DIRECTORY BLOCK INDEX
062.167  062 050 070             STA      DIRBLK
062.172  052 045 070             LHLD     DBI
062.175  206                     ADD      M
062.176  157                     MOV      L,A              HL = index of first directory block

062.177  046 000                 MVI      H,0
062.201  315 044 064             CALL     BTS              HL = sector index of directory
062.204  042 261 070             SHLD     LABEL+LAB.DIS

062.207  247                     ANA      A                Clear ERROR flag
062.210  311                     RET
```

```
     062.211  000 000      3180  ADBA    DW       0                   Save first directory block


                           3182  **      BGT - BUILD GRT TABLE.                             /80.06.GC/
                           3183  *
                           3184  *       BGT BUILDS THE GRT FROM THE RGT, ENTERING BLOCK CHAINS
                           3185  *       FOR THE DIRECTORY, THE RGT AND THE GRT FILES.
                           3186  *
                           3187  *       FIRST, THE BLOCKS USED IN THE RGT, GRT, AND DIRECTORY
                           3188  *       FILES ARE LINKED UP.
                           3189  *
                           3190  *       THEN, ALL FREE BLOCKS LEFT (INDEX = 1) ARE LINKED INTO
                           3191  *       THE FREE CHAIN.
                           3192  *
                           3193  *       ENTRY   NONE
                           3194  *       EXIT    NONE
                           3195  *       USES    ALL
                           3196
                           3197
     062.213  001 000 001  3198  BGT     LXI      B,256
     062.216  021 000 075  3199          LXI      D,RGTAB
     062.221  041 000 074  3200          LXI      H,GRTAB
     062.224  315 252 030  3201          CALL     $MOVE               COPY RGTAB INTO GRTAB
     062.227  052 270 070  3202          LHLD     LABEL+LAB.RGT
     062.232  315 301 065  3203          CALL     STB                 HL = RGT block number
     062.235  021 000 074  3204          LXI      D,GRTAB
     062.240  031          3205          DAD      D                   HL = table address
     062.241  066 000      3206          MVI      M,0                 Reserve the block for the RGT file
                           3207
                           3208  *       Link the directory blocks together in required order
                           3209
     062.243  072 054 070  3210          LDA      NDIRBLK
     062.246  107          3211          MOV      B,A                 B  = block count
     062.247  052 045 070  3212          LHLD     DBI
     062.252  353          3213          XCHG                         DE = Directory block interleave table
     062.253  052 050 070  3214          LHLD     DIRBLK              L  = first directory block
     062.256  046 074      3215          MVI      H,GRTAB/256
     062.260  115          3216          MOV      C,L                 C  = directory block base address
     062.261  032          3217          LDAX     D
     062.262  023          3218          INX      D
     062.263  201          3219          ADD      C
     062.264  157          3220          MOV      L,A                 HL = address of first directory block
                           3221
     062.265  005          3222          DCR      B                   last block is special
     062.266  312 302 062  3223          JZ       BGT1.5              only 1 block
                           3224
     062.271  032          3225  BGT1    LDAX     D                   A  = block offset
     062.272  023          3226          INX      D
     062.273  201          3227          ADD      C                   A  = physical block number
     062.274  167          3228          MOV      M,A                 this block points the the next
     062.275  157          3229          MOV      L,A                 HL = address of successor block
     062.276  005          3230          DCR      B
     062.277  302 271 062  3231          JNZ      BGT1                more blocks to do
                           3232
```

```
062.302  066 000          BGT1.5  MVI      M,0             last block points to 00 as final block
062.304  175                      MOV      A,L
062.305  062 051 070              STA      DIRLBLK         SET DIRECTORY LAST BLOCK
062.310  072 047 070              LDA      GRTBLK
062.313  157                      MOV      L,A
062.314  066 000                  MVI      M,0             GRT BLOCK CHAINS TO NONE

                          *       CHAIN ALL UNUSED (001) BLOCKS TOGETHER.
                          *
                          *       PUT START POINTER IN BLOCK 00, LAST BLOCK POINTS TO 00

062.316  016 000                  MVI      C,0             (C) = NEXT FREE BLOCK
062.320  041 377 074              LXI      H,GRTAB+255     WORK FROM THE BACK
062.323  076 001          BGT2    MVI      A,1
062.325  276                      CMP      M
062.326  302 333 062              JNE      BGT3            NOT FREE
062.331  161                      MOV      M,C             LINK ON FRONT OF CHAIN
062.332  115                      MOV      C,L             IS NOW FRONT OF CHAIN
062.333  055              BGT3    DCR      L
062.334  302 323 062              JNZ      BGT2            MORE TO GO
062.337  161                      MOV      M,C             SET POINTER IN GROUP 00
062.340  311                      RET



                          **      IDD - INITIALIZE DEVICE DIRECTORY.                            /80.06.sc/
                          *
                          *       IDD INITIALIZES THE DEVICE DIRECTORY BY WRITTING THE DIRECTORY
                          *       BLOCK 1 CLEAR, BLOCKS 3 - N EMPTY, AND BLOCK 2 CONTAINING
                          *       THE FILES
                          *
                          *       RGT      .SYS
                          *       GRT      .SYS
                          *       DIRECT   .SYS
                          *
                          *       THE DIRECTORYS BLOCKS ARE INTERNALLY LINKED TO FOLLOW THEIR POSITION
                          *       IN THE GRT.
                          *
                          *       ENTRY    NONE
                          *       EXIT     NONE
                          *       USES     ALL
                          *

062.341  076 002          IDD     MVI      A,2
062.343  062 350 063              STA      IDDD            SET COUNT FOR SECOND DIRECTORY BLOCK
062.346  006 377                  MVI      B,DF.EMP        SET ENTRYS EMPTY
062.350  315 012 064              CALL     PDS             PRESET DIRECTORY SECTOR
062.353  072 265 070              LDA      LABEL+LAB.SPG
062.356  346 376                  ANI      377Q-1
062.360  017                      RRC
062.361  062 351 063              STA      IDDE            Initialize directory blocks per GRT block
062.364  062 352 063              STA      IDDF            Initialize number this GRT block

                          *       WRITE DIRECTORY BLOCKS, IN ORDER

```

```
062.367  076 027      3286          MVI     A,DIRELEN
062.371  062 176 073  3287          STA     IDDA+DIS.ENL    SET DIRECTORY ENTRY LENGTH
062.374  052 261 070  3288          LHLD    LABEL+LAB.DIS   (HL) = INDEX OF FIRST BLOCK
                      3289
062.377  042 177 073  3290  IDD2    SHLD    DIS.SEC+IDDA    SET IN DIRECTORY ENTRY
063.002  315 353 063  3291          CALL    ANS             Advance to the next sector
063.005  042 201 073  3292          SHLD    DIS.LNK+IDDA    SET IN BLOCK
063.010  041 350 063  3293          LXI     H,IDDD
063.013  065          3294          DCR     M               COUNT SECTOR
063.014  302 166 063  3295          JNZ     IDD4            IS NOT SECTOR 2
                      3296
                      3297  *       IS SECTOR 2, ADD SPECIAL FILES TO THE END OF IT.
                      3298
063.017  052 270 070  3299          LHLD    LABEL+LAB.RGT
063.022  315 301 065  3300          CALL    STB
063.025  175          3301          MOV     A,L
063.026  062 234 063  3302          STA     IDDC0           RGT start block
063.031  062 235 063  3303          STA     IDDC1           RGT last block
                      3304
063.034  052 310 040  3305          LHLD    S.DATC
063.037  042 237 063  3306          SHLD    IDDC2           SET CREATION AND ALTERATION DATES FOR ALL
063.042  042 241 063  3307          SHLD    IDDC2+2
063.045  042 266 063  3308          SHLD    IDDC4
063.050  042 270 063  3309          SHLD    IDDC4+2
063.053  042 315 063  3310          SHLD    IDDC6
063.056  042 317 063  3311          SHLD    IDDC6+2
                      3312
                      3313  *       SET DIRECTORY POINTERS TO GRT BLOCK AND DIRECTORY
                      3314
063.061  072 047 070  3315          LDA     GRTBLK
063.064  062 263 063  3316          STA     IDDC3           SET GRT BLOCK
063.067  062 264 063  3317          STA     IDDC3+1         ONLY ONE BLOCK IN FILE
063.072  076 001      3318          MVI     A,1
063.074  062 265 063  3319          STA     IDDC3+2         ONLY ONE SECTOR IN FILE
                      3320
063.077  052 261 070  3321          LHLD    LABEL+LAB.DIS
063.102  315 301 065  3322          CALL    STB             HL = block number            /80.05.sc/
063.105  175          3323          MOV     A,L             A  = group number direct. start /80.05.sc/
063.106  062 312 063  3324          STA     IDDC5
063.111  072 051 070  3325          LDA     DIRLBLK
063.114  062 313 063  3326          STA     IDDC5+1         SET GROUP NUMBER OF LAST BLOCK
063.117  072 265 070  3327          LDA     LABEL+LAB.SPG                                /80.06.sc/
063.122  062 314 063  3328          STA     IDDC5+2         SET LAST SECTOR INDEX
                      3329
063.125  001 134 000  3330          LXI     B,IDDCL
063.130  021 214 063  3331          LXI     D,IDDC
063.133  041 041 073  3332          LXI     H,IDDB+IDDBL-IDDCL
063.136  315 252 030  3333          CALL    $MOVE           MOVE ENTRYS INTO BLOCK
063.141  001 000 002  3334          LXI     B,512
063.144  021 203 071  3335          LXI     D,IDDA
063.147  052 177 073  3336          LHLD    DIS.SEC+IDDA
063.152  315 271 064  3337          CALL    DRIVER.         WRITE BLOCK AND EXIT         /80.05.GC/
063.155  001          3338          DB      DC.WRI                                       /80.05.GC/
                      3339
                      3340  *       HAVE WRITTEN 1ST AND 2ND DIRECTORY BLOCKS, FLAG
                      3341  *       ALL REMAINING BLOCKS EMPTY
```

```
                           3342
063.156  006 376           3343            MVI     B,DF.CLR         FLAG REST OF DIRECTORY EMPTY
063.160  315 012 064       3344            CALL    PDS              PRESET DIRECTORY SECTOR
063.163  303 203 063       3345            JMP     IDD5             SKIP RE-WRITTING 2ND SECTOR
                           3346
                           3347  *         WRITE BLOCK TO DISK
                           3348
063.166  001 000 002       3349  IDD4      LXI     B,512
063.171  021 203 071       3350            LXI     D,IDDA
063.174  052 177 073       3351            LHLD    DIS.SEC+IDDA
063.177  315 271 064       3352            CALL    DRIVER.          WRITE BLOCK                        /80.05.GC/
063.202  001               3353            DB      DC.WRI                                              /80.05.GC/
                           3354
063.203  052 201 073       3355  IDD5      LHLD    DIS.LNK+IDDA
063.206  174               3356            MOV     A,H
063.207  265               3357            ORA     L
063.210  302 377 062       3358            JNZ     IDD2             NOT LAST ONE, DO SOME MORE
063.213  311               3359            RET
                           3360
                           3361
                           3362  **        DIRECTORY ENTRYS FOR SPECIAL FILES
                           3363
063.214                    3364  IDDC      DS      0
000.000                    3365            ERRNZ   *-IDDC-DIR.NAM
063.214  122 107 124       3366            DB      'RGT',0,0,0,0,0 RGT.SYS
000.000                    3367            ERRNZ   *-IDDC-DIR.EXT
063.224  123 131 123       3368            DB      'SYS'
000.000                    3369            ERRNZ   *-IDDC-DIR.PRO
063.227  000 000           3370            DB      0,0              VERSION AND PPN
000.000                    3371            ERRNZ   *-IDDC-DIR.CLU
063.231  000               3372            DB      0                CLUSTER
000.000                    3373            ERRNZ   *-IDDC-DIR.FLG
063.232  360               3374            DB      DIF.SYS+DIF.LOC+DIF.CNT+DIF.WP  SET UNCHANGABLY WRITE-PROTECTED
063.233  000               3375            DB      0                UNUSED
000.000                    3376            ERRNZ   *-IDDC-DIR.FGN
063.234  000               3377  IDDC0     DB      0                FIRST GROUP                        /80.06.GC/
000.000                    3378            ERRNZ   *-IDDC-DIR.LGN
063.235  000               3379  IDDC1     DB      0                LAST GROUP                         /80.06.GC/
000.000                    3380            ERRNZ   *-IDDC-DIR.LSI
063.236  001               3381            DB      1                SECTOR INDEX OF EOF
000.000                    3382            ERRNZ   *-IDDC-DIR.CRD
063.237                    3383  IDDC2     DS      4                CREATION AND ALTERATION DATE
                           3384
063.243  107 122 124       3385            DB      'GRT',0,0,0,0,0          GRT.SYS
063.253  123 131 123       3386            DB      'SYS'
063.256  000 000           3387            DB      0,0              PPN, VERSION
063.260  000               3388            DB      0                CLUSTER
063.261  360               3389            DB      DIF.SYS+DIF.LOC+DIF.CNT+DIF.WP  UNCHANGABLY WRITE-PROTECT
063.262  000               3390            DB      0                UNUSED
063.263                    3391  IDDC3     DS      3                FIRST GROUP, LAST GROUP, LAST SECTOR
063.266                    3392  IDDC4     DS      4                CREATION AND ALTERATION DATE
                           3393
063.272  104 111 122       3394            DB      'DIRECT',0,0             DIRECT.SYS
063.302  123 131 123       3395            DB      'SYS'
063.305  000 000           3396            DB      0,0              PPN, VERSION
063.307  000               3397            DB      0                CLUSTER
```

```
063.310  340                   DB      DIF.SYS+DIF.LOC+DIF.WP  UNCHANGABLY WRITE PROTECT
063.311  000                   DB      0               UNUSED
063.312               IDDC5    DS      3               FIRST GROUP, LAST GROUP, LAST SECTOR
063.315               IDDC6    DS      4               CREATION AND ALTERATION DATE
063.321  376                   DB      DF.CLR          LAST SPOT IN 2ND BLOCK IS EMPTY
063.322  107 101 103           DB      'GAC / HEATH CO.'
063.341                        DS      DIRELEN-1-15    REST OF ENTRY MEANINGLESS

000.134               IDDCL    EQU     *-IDDC
000.000                        ERRNZ   4*DIRELEN-IDDCL         SHOULD BE FOUR ENTRYS

063.350  000          IDDD     DB      0               DIRECTORY BLOCK COUNTER
063.351  000          IDDE     DB      0               Directory blocks per GRT block  /80.06.sc/
063.352  000          IDDF     DB      0               Remaining directory blocks block /80.06.sc/



                      **       ANS     - Advance to Next Sector
                      *
                      *        ANS updates the pointer to the current sector, to
                      *        point to the next sector.
                      *
                      *        ENTRY:  HL      = current sector
                      *                IDDE, IDDF initialized
                      *
                      *        EXIT:   HL      = next sector
                      *                IDDF updated
                      *
                      *        USES:   PSW,HL
                      *

063.353  072 352 063  ANS      LDA     IDDF
063.356  075                   DCR     A               Count this directory block
063.357  062 352 063           STA     IDDF
063.362  312 370 063           JZ      ANS1            need a new block

                      *        More available directory blocks in this GRT block

063.365  043                   INX     H
063.366  043                   INX     H
063.367  311                   RET

063.370  072 351 063  ANS1     LDA     IDDE
063.373  062 352 063           STA     IDDF            Initialize count for next GRT block
063.376  315 301 065           CALL    STB             HL = block number
064.001  046 074               MVI     H,GRTAB/256
064.003  156                   MOV     L,M             L  = next block number
064.004  046 000               MVI     H,0
064.006  315 044 064           CALL    BTS             convert it back to a sector number
064.011  311                   RET
```

```
                        3447  **       PDS - PRESET DIRECTORY SECTOR.
                        3448  *
                        3449  *        PDS BUILDS A DIRECTORY BLOCK INTO *IDDB*  FULL OF DIRECTORY
                        3450  *                ENTRYS IN THE FORMAT:
                        3451  *
                        3452  *        FLAG,0,0, . . . ,0
                        3453  *
                        3454  *        WHERE FLAG = SOME  SUPPLIED VALUE.
                        3455  *
                        3456  *        ENTRY   (B) = FLAG BYTE
                        3457  *        EXIT    NONE
                        3458  *        USES    ALL
                        3459
                        3460
064.012  041 203 071    3461  PDS      LXI     H,IDDB
064.015  021 372 001    3462           LXI     D,IDDBL
064.020  160            3463  PDS1     MOV     M,B             SET FIRST BYTE
064.021  043            3464           INX     H
064.022  033            3465           DCX     D
                        3466
                        3467  *        NOW ZERO REST OF ENTRY
                        3468
064.023  016 026        3469           MVI     C,DIRELEN-1
064.025  066 000        3470  PDS2     MVI     M,0             ZERO IT
064.027  043            3471           INX     H
064.030  033            3472           DCX     D               COUNT BYTE FROM DIRECTORY BLOCK
064.031  015            3473           DCR     C               COUNT BYTE FROM DIRECTORY ENTRY
064.032  302 025 064    3474           JNZ     PDS2
064.035  172            3475           MOV     A,D
064.036  263            3476           ORA     E
064.037  302 020 064    3477           JNZ     PDS1
064.042  167            3478           MOV     M,A             ZERO BYTE FOLLOWS ENTRYS
064.043  311            3479           RET                     EXIT
```

```
                              **      BTS     -  Block To Sector                         /80.05.sc/
                              *
                              *       BTS converts a block number to a sector number.
                              *
                              *       ENTRY:  HL       = block number
                              *
                              *       EXIT:   HL       = sector number
                              *
                              *       USES:   HL
                              *

064.044  365                  BTS     PUSH    PSW
064.045  325                          PUSH    D
064.046  353                          XCHG             DE = multiplicand
064.047  072 265 070                  LDA     LABEL+LAB.SPG
064.052  315 007 031                  CALL    $MU86    HL = DE * A
064.055  321                          POP     D
064.056  361                          POP     PSW
064.057  311                          RET


                              **      CNO     -  Check NO
                              *
                              *       CNO checks a line of text to verify a  *NO*  answer.
                              *
                              *       ENTRY:  HL       = address of line
                              *
                              *       EXIT:   PSW      = 'Z' SET    if  NO
                              *                            'Z' CLEAR if NOT  NO
                              *
                              *       USES:   PSW
                              *

064.060  305                  CNO     PUSH    B
064.061  325                          PUSH    D
064.062  345                          PUSH    H

064.063  021 073 064                  LXI     D,CNOA
064.066  016 003                      MVI     C,CNOAL
064.070  303 106 064                  JMP     CYS1

064.073  116 117 000          CNOA    DB      'NO',0
000.003                       CNOAL   EQU     *-CNOA


                              **      CYS     -  Check YES
                              *
                              *       CYS inputs a line of text to verify a  *YES*  answer.
                              *       Any answer except 'YES' is regarded as  *NO*.
                              *
                              *       ENTRY:  HL       = address of line
                              *
```

```
                       3533  *       EXIT:   PSW     = 'Z' SET   if YES
                       3534  *                         'Z' CLEAR if NO
                       3535  *
                       3536  *       USES:   PSW
                       3537  *
                       3538
064.076  305           3539  CYS     PUSH    B
064.077  325           3540          PUSH    D
064.100  345           3541          PUSH    H
                       3542
064.101  021 115 064   3543          LXI     D,CYSA
064.104  016 004       3544          MVI     C,CYSAL
064.106  315 060 030   3545  CYS1    CALL    $COMP
                       3546
064.111  341           3547          POP     H
064.112  321           3548          POP     D
064.113  301           3549          POP     B
064.114  311           3550          RET
                       3551
064.115  131 105 123   3552  CYSA    DB      'YES',0
000.004                3553  CYSAL   EQU     *-CYSA
064.121                3554          XTEXT   DDS



                       3556X **      DDS     -  Decode Device Specification               /80.05.sc/
                       3557X *
                       3558X *       DDS decodes the device specification, returning a two character
                       3559X *       device name, and one byte unit number.
                       3560X *
                       3561X *
                       3562X *       ENTRY:  BC      = Address of destination fields
                       3563X *               DE      = Address of default
                       3564X *               HL      = Address of string specifier
                       3565X *
                       3566X *       EXIT:   PSW     = 'C' SET   if ERROR
                       3567X *                         'C' CLEAR if NO ERROR
                       3568X *
                       3569X *       USES:   ALL
                       3570X *
                       3571X
064.121                3572X DDS     EQU     *
                       3573X
                       3574X *       Initialize the fields to the defaults
                       3575X
064.121  305           3576X         PUSH    B
064.122  315 231 064   3577X         CALL    DDS3
064.125  315 231 064   3578X         CALL    DDS3
064.130  032           3579X         LDAX    D
064.131  326 060       3580X         SUI     '0'
064.133  002           3581X         STAX    B
064.134  301           3582X         POP     B
                       3583X
064.135  176           3584X         MOV     A,M
064.136  247           3585X         ANA     A
```

```
064.137  310          3586X          RZ                     took the default
                      3587X
                      3588X *        Check the supplied name
                      3589X
064.140  315 145 067  3590X          CALL   $SOB            skip the whitespace
064.143  315 212 064  3591X          CALL   DDS2
064.146  330          3592X          RC                     Not alpha
064.147  315 212 064  3593X          CALL   DDS2
064.152  330          3594X          RC                     Not alpha
                      3595X
064.153  176          3596X          MOV    A,M
064.154  376 072      3597X          CPI    ':'
064.156  076 000      3598X          MVI    A,0             assume unit 0
064.160  312 174 064  3599X          JZ     DDS1            default to unit 0
                      3600X
                      3601X *        Check for a valid digit
                      3602X
064.163  176          3603X          MOV    A,M
064.164  326 060      3604X          SUI    '0'
064.166  330          3605X          RC                     Not digit
064.167  376 010      3606X          CPI    7+1
064.171  077          3607X          CMC
064.172  330          3608X          RC                     digit too large
064.173  043          3609X          INX    H
                      3610X
064.174  002          3611X DDS1     STAX   B
064.175  003          3612X          INX    B
064.176  176          3613X          MOV    A,M
064.177  043          3614X          INX    H
064.200  376 072      3615X          CPI    ':'
064.202  067          3616X          STC
064.203  300          3617X          RNZ                    requires ':'
                      3618X
064.204  176          3619X          MOV    A,M
064.205  247          3620X          ANA    A
064.206  067          3621X          STC
064.207  300          3622X          RNZ                    require 'NULL'
                      3623X
064.210  247          3624X          ANA    A               Clear ERROR flag
064.211  311          3625X          RET
                      3626X
064.212  176          3627X DDS2     MOV    A,M
064.213  043          3628X          INX    H
064.214  315 071 052  3629X          CALL   $MCU
064.217  376 101      3630X          CPI    'A'
064.221  330          3631X          RC                     Not alpha
                      3632X
064.222  376 133      3633X          CPI    'Z'+1
064.224  077          3634X          CMC
064.225  330          3635X          RC                     Not alpha
                      3636X
064.226  002          3637X          STAX   B
064.227  003          3638X          INX    B               replace the default char
064.230  311          3639X          RET
                      3640X
064.231  032          3641X DDS3     LDAX   D
```

```
064.232  023                           INX     D
064.233  315 071 052                   CALL    $MCU                  Map to upper case
064.236  002                           STAX    B
064.237  003                           INX     B
064.240  311                           RET
000.000                                ERRNZ   IOC.UNI-IOC.DEV-2           2 byte device
000.000                                ERRNZ   IOC.DIR-IOC.UNI-1           1 byte unit



                            **         DRIVER
                            *
                            *          DRIVER calls the device driver for the currently seclected
                            *

064.241  343                DRIVER     XTHL
064.242  176                           MOV     A,M                   A  = driver request
064.243  043                           INX     H
064.244  343                           XTHL

064.245  345                DRIVER1    PUSH    H
064.246  365                           PUSH    PSW
064.247  325                           PUSH    D
064.250  052 043 070                   LHLD    CRNDEV
064.253  315 234 030                   CALL    $INDL
064.256  004 000                       DW      DVT.ADR
064.260  353                           XCHG                          HL = address of device table entry
064.261  021 003 000                   LXI     D,DEV.JMP
064.264  031                           DAD     D                     HL = address of jump vector
064.265  321                           POP     D
064.266  361                           POP     PSW
064.267  343                           XTHL                          restore original HL
064.270  311                           RET                           enter the device driver

064.271  343                DRIVER.    XTHL
064.272  176                           MOV     A,M                   A  = driver request
064.273  043                           INX     H
064.274  343                           XTHL
064.275  315 245 064                   CALL    DRIVER1
064.300  320                           RNC                           No error

064.301  315 136 031                   CALL    $TYPTX
064.304  007 125 156                   DB      BELL,'Unable to Read/Write this diskette',BELL,ENL
064.351  303 047 054                   JMP     RESTART



                            **         DU68    - Divide Unsigned (16 bit/8 bit)
                            *
                            *          DU68 performs a 16 bit X 8 bit unsigned division.
                            *
                            *          ENTRY:  A        = divisor
                            *                  BC       = dividend
                            *
```

```
                               3692  *        EXIT:    See $DU66
                               3693  *
                               3694  *        USES:    ALL
                               3695  *
                               3696
    064.354  137               3697  DU68     MOV      E,A
    064.355  026 000           3698           MVI      D,0
    064.357  303 106 030       3699           JMP      $DU66



                               3701  **       IVP      - Initialize Volume Parameters
                               3702  *
                               3703  *        IVP initializes the volume parameters in the label
                               3704  *        sectors.  This routine must be called after the boot
                               3705  *        device is mounted, since relocating the primary boot
                               3706  *        device destroys the default area.
                               3707  *
                               3708  *        ENTRY:   none
                               3709  *
                               3710  *        EXIT:    None
                               3711  *
                               3712  *        USES:    All
                               3713  *
                               3714
    064.362  052 310 040       3715  IVP      LHLD     S.DATC
    064.365  042 011 065       3716           SHLD     IVPB             Initialize default date to today's
    064.370  001 035 000       3717           LXI      B,IVPAL
    064.373  021 005 065       3718           LXI      D,IVPA
    064.376  041 203 042       3719           LXI      H,SB.VER
    065.001  315 252 030       3720           CALL     $MOVE
    065.004  311               3721           RET
                               3722
    065.005                    3723  IVPA     DS       0
    022.202                    3724  .        SET      *-SB.VER
                               3725
    000.000                    3726           ERRNZ    *-.-SB.VER
    065.005  040               3727           DB       VERS             Version of INIT that initialized the disk
                               3728
    000.000                    3729           ERRNZ    *-.-SB.FLG
    065.006  000               3730           DB       0                Flags
                               3731
    000.000                    3732           ERRNZ    *-.-SB.BAU
    065.007  000 000           3733           DW       0                Baud Rate
                               3734
    000.000                    3735           ERRNZ    *-.-SB.DAT
    065.011  000 000           3736  IVPB     DW       0                Default Date
                               3737
    065.013  000 000 000       3738           DW       0,0,0,0,0,0,0,0,0,0,0
    065.041  000               3739           DB       0
                               3740
    000.000                    3741           ERRNZ    *-.-SB.BPE
    000.035                    3742  IVPAL    EQU      *-IVPA
```

```
                          **      MOUNT                                                   /80.05.sc/
                          *
                          *       MOUNT issues the mount system call to the device driver.
                          *       In the case of the H17, this sets up the all important
                          *       volume number.
                          *
                          *       ENTRY:  L        = Volume Number
                          *
                          *       EXIT:   NONE
                          *
                          *       USES:   ALL
                          *

065.042  365              MOUNT   PUSH    PSW
065.043  072 256 070              LDA     LABEL+LAB.SER
065.046  157                      MOV     L,A               L = Volume Number
065.047  361                      POP     PSW

065.050  315 271 064      MOUNT1  CALL    DRIVER.
065.053  010                      DB      DC.MOU
065.054  311                      RET

065.055  056 000          MOUNT.  MVI     L,0               Used for forcing volume 0 to write track 0
065.057  303 050 065              JMP     MOUNT1

065.062  315 241 064      MOUNT.. CALL    DRIVER            Mount without error detection
065.065  010                      DB      DC.MOU
065.066  311                      RET


                          **      MSU     - Map System Unit                               /80.09.sc/
                          *
                          *       MSU maps the system unit to match the performance of
                          *       ISY in HDOS.SYS.  That is, it corrects for the fact
                          *       that a unit other than 0 may have been booted from.
                          *       This may seem like somewhatof a kludge, but it seems
                          *       reasonable.  After mapping the unit, it calls the
                          *       SY: subfunction processor, and then un-maps the unit
                          *       after the subfunction processor exits.
                          *
                          *       NOTE:   The sub-function processor temporarily zeroes
                          *               the boot unit so that if the function processor
                          *               calls the driver, the unit will not be mapped
                          *               twice.  After processing the sub-function, the
                          *               mapping factor is restored.
                          *
                          *       ENTRY:  AIO.UNI = Unit Number
                          *
                          *       EXIT:   AIO.UNI = Unit Number, with MSUA called
                          *
                          *       USES:   NONE
                          *

065.067  345              MSU     PUSH    H
```

```
065.070  365            3797                PUSH    PSW
065.071  052 346 040    3798                LHLD    S.DLINK
065.074  257            3799                XRA     A
065.075  315 021 067    3800                CALL    $INDSB           No more mapping of system device
065.100  021 000        3801                DW      M.SUNI            because already mapped here
065.102  072 061 041    3802                LDA     AIO.UNI
065.105  147            3803                MOV     H,A
065.106  361            3804                POP     PSW
065.107  343            3805                XTHL                     Save current device specification
                        3806
065.110  365            3807                PUSH    PSW
065.111  325            3808                PUSH    D
065.112  345            3809                PUSH    H
                        3810
065.113  072 060 070    3811                LDA     SYMNU
065.116  127            3812                MOV     D,A              D = max num of units
065.117  072 061 041    3813                LDA     AIO.UNI
065.122  052 057 070    3814                LHLD    SUNIT
065.125  205            3815                ADD     L                A = AIO.UNI+SUNIT
065.126  222            3816                SUB     D
065.127  322 133 065    3817                JNC     MSU1
065.132  202            3818                ADD     D                A = A mod D
065.133                 3819       MSU1     EQU     *
                        3820
065.133  062 061 041    3821                STA     AIO.UNI
                        3822
065.136  341            3823                POP     H
065.137  321            3824                POP     D
065.140  361            3825                POP     PSW
                        3826
065.141  315 377 377    3827                CALL    -1
065.142                 3828       MSUA     EQU     *-2
                        3829
065.144  343            3830                XTHL
065.145  365            3831                PUSH    PSW
065.146  174            3832                MOV     A,H
065.147  062 061 041    3833                STA     AIO.UNI          Replace the original value
065.152  072 057 070    3834                LDA     SUNIT
065.155  052 346 040    3835                LHLD    S.DLINK
065.160  315 021 067    3836                CALL    $INDSB
065.163  021 000        3837                DW      M.SUNI           Restore original mapping factor
065.165  361            3838                POP     PSW
065.166  341            3839                POP     H                Restore exit HL
065.167  311            3840                RET



                        3842       **       RPB     - Relocate Primary Boot                       /80.06.gc/
                        3843       *
                        3844       *        RPB relocates the primary boot to the beginning of the boot
                        3845       *        track.
                        3846       *
                        3847       *        ENTRY:  CRNDEV  = pointer to internal device table entry
                        3848       *
                        3849       *        EXIT:   Primary Boot Relocated
```

```
                              3850  *
                              3851  *        USES:    ALL
                              3852  *
                              3853
     065.170  001 000 002     3854  RPB      LXI      B,SB.SDB-SB.BOO          BC = move COUNT
     002.000                  3855  .        SET      SB.SDB-SB.BOO/256*256
     000.000                  3856           ERRNZ    SB.SDB-SB.BOO-.          Must be a multiple of 256
     065.173  052 043 070     3857           LHLD     CRNDEV
     065.176  315 234 030     3858           CALL     $INDL                    DE = FROM address
     065.201  006 000         3859           DW       DVT.PAR
     065.203  041 200 042     3860           LXI      H,SB.BOO                 HL = DESTINATION address
     065.206  315 252 030     3861           CALL     $MOVE
                              3862
                              3863  *        Relocate the Primary Boot
                              3864
     065.211  052 043 070     3865           LHLD     CRNDEV
     065.214  345             3866           PUSH     H
     065.215  315 234 030     3867           CALL     $INDL
     065.220  006 000         3868           DW       DVT.PAR
     065.222  353             3869           XCHG
     065.223  315 224 030     3870           CALL     $CHL             HL = - HL
     065.226  021 200 042     3871           LXI      D,SB.BOO
     065.231  031             3872           DAD      D
     065.232  104             3873           MOV      B,H
     065.233  115             3874           MOV      C,L              BC = Relocation Constant
     065.234  341             3875           POP      H
                              3876
     065.235  315 234 030     3877           CALL     $INDL
     065.240  010 000         3878           DW       DVT.PIC
     065.242  353             3879           XCHG                      HL = PIC Table address
                              3880
     065.243  136             3881  RPB1     MOV      E,M
     065.244  043             3882           INX      H
     065.245  126             3883           MOV      D,M              DE = Relocation Address
     065.246  043             3884           INX      H
                              3885
     065.247  172             3886           MOV      A,D
     065.250  263             3887           ORA      E
     065.251  310             3888           RZ                        At the end of the PIC table
                              3889
     065.252  345             3890           PUSH     H
     065.253  041 200 044     3891           LXI      H,SB.SDB
     065.256  315 326 051     3892           CALL     HLCPDE
     065.261  341             3893           POP      H
     065.262  332 243 065     3894           JC       RPB1             Address is not in PRIMARY Boot
                              3895
     065.265  353             3896           XCHG                      HL = address of word to relocate
     065.266  176             3897           MOV      A,M
     065.267  201             3898           ADD      C
     065.270  167             3899           MOV      M,A
     065.271  043             3900           INX      H
     065.272  176             3901           MOV      A,M
     065.273  210             3902           ADC      B
     065.274  167             3903           MOV      M,A              Relocate the word
     065.275  353             3904           XCHG
     065.276  303 243 065     3905           JMP      RPB1             Relocate the next word
```

```
                                3907  **        STB     - Sector To Block                                   /80.05.sc/
                                3908  *
                                3909  *         STB converts a sector number to the corresponding block number.
                                3910  *
                                3911  *         ENTRY:  HL      = sector number
                                3912  *
                                3913  *         EXIT:   HL      = block number
                                3914  *
                                3915  *         USES:   HL
                                3916  *
                                3917
        065.301  365            3918  STB       PUSH    PSW
        065.302  305            3919            PUSH    B
        065.303  325            3920            PUSH    D
        065.304  104            3921            MOV     B,H
        065.305  115            3922            MOV     C,L             BC = block number
        065.306  072 265 070    3923            LDA     LABEL+LAB.SPG
        065.311  315 354 064    3924            CALL    DU68            HL = BC/A
        065.314  321            3925            POP     D
        065.315  301            3926            POP     B
        065.316  361            3927            POP     PSW
        065.317  311            3928            RET



                                3930  **        SUBFUN  -  Sub-Function
                                3931  *
                                3932  *         SUBFUN is the INIT sub-function processor
                                3933  *
                                3934
        065.320  343            3935  SUBFUN    XTHL
        065.321  176            3936            MOV     A,M             A  = sub-function
        065.322  043            3937            INX     H
        065.323  343            3938            XTHL
                                3939
        065.324  345            3940            PUSH    H
        065.325  325            3941            PUSH    D
        065.326  365            3942            PUSH    PSW
                                3943
        065.327  052 061 070    3944            LHLD    SYPTR
        065.332  353            3945            XCHG
        065.333  052 043 070    3946            LHLD    CRNDEV
        065.336  315 326 051    3947            CALL    HLCPDE
        065.341  041 067 065    3948            LXI     H,MSU
        065.344  312 363 065    3949            JZ      SUBFUN1         Processing device SY:
                                3950
        065.347  052 043 070    3951            LHLD    CRNDEV
        065.352  315 234 030    3952            CALL    $INDL
        065.355  006 000        3953            DW      DVT.PAR
        065.357  041 000 002    3954            LXI     H,SB.SDB-SB.BOO
        065.362  031            3955            DAD     D               The sub-functions begin there
                                3956
        065.363  361            3957  SUBFUN1   POP     PSW
        065.364  321            3958            POP     D
        065.365  343            3959            XTHL
```

```
065.366  311                     RET                         enter the drivers sub-processor



                         **      SVP     - Set-Up Volume Parameters
                         *
                         *       SVP sets up the volume parameters in the label sector.
                         *
                         *       ENTRY:  NONE
                         *
                         *       EXIT:   LABEL Device Dependant volume parameters initialized
                         *
                         *       USES:   ALL
                         *

065.367                  SVP     EQU     *

                         *       Fetch Device Dependant Parameters

065.367  315 320 065             CALL    SUBFUN          HL = address of the parameters
065.372  003                     DB      INI.PAR

065.373  353                     XCHG                    DE = current volume parameters
065.374  041 272 070             LXI     H,LABEL+LAB.VPR
065.377  001 005 000             LXI     B,LAB.VPL
066.002  315 252 030             CALL    $MOVE
066.005  041 375 070             LXI     H,LABEL+LAB.AUX
066.010  001 001 000             LXI     B,LAB.AXL
066.013  315 252 030             CALL    $MOVE           Set up auxiliary parameters

                         *       Compute the rest

066.016  052 272 070             LHLD    LABEL+LAB.SIZ   HL = volume size
066.021  021 377 000             LXI     D,255           This assumes the sector size is 256
066.024  031                     DAD     D
066.025  174                     MOV     A,H             A  = cluster factor
066.026  074                     INR     A
066.027  346 376                 ANI     377Q-1          Force cluster to a multiple of 2
066.031  062 265 070             STA     LABEL+LAB.SPG   Stuff sector/group value

066.034  052 272 070             LHLD    LABEL+LAB.SIZ
066.037  104                     MOV     B,H
066.040  115                     MOV     C,L             BC = volume size
066.041  315 354 064             CALL    DU68            HL = number of groups = BC/A

066.044  174                     MOV     A,H
066.045  247                     ANA     A
066.046  302 135 066             JNZ     SVP2            Too many groups (should never happen)

066.051  175                     MOV     A,L
066.052  062 055 070             STA     NGROUPS         Save the number of groups
066.055  315 044 064             CALL    BTS             HL = first illegal sect. num. = Num. of sect.
066.060  042 272 070             SHLD    LABEL+LAB.SIZ   Update size to the HDOS size

                         *       Compute Low-Order Space Reservation
```

```
                       4013
066.063  041 012 000   4014           LXI     H,DDF.USR        Reserve the system data area
066.066  072 265 070   4015           LDA     LABEL+LAB.SPG
066.071  075           4016           DCR     A
066.072  315 072 030   4017           CALL    $DADA            Round up to a totally free block
066.075  315 301 065   4018           CALL    STB
066.100  175           4019           MOV     A,L
066.101  376 002       4020           CPI     2
066.103  322 110 066   4021           JNC     SVP1             More than 2 are reserved
                       4022
066.106  076 002       4023           MVI     A,2              Force the reservation of at least 2 groups
066.110                4024  SVP1     EQU     *
                       4025
066.110  062 056 070   4026           STA     NSPCGRP          Reserve these groups as special
                       4027
066.113  157           4028           MOV     L,A
066.114  046 000       4029           MVI     H,0              HL = RGT block number
066.116  315 044 064   4030           CALL    BTS              Use this sector for the RGT
066.121  042 270 070   4031           SHLD    LABEL+LAB.RGT    Save the RGT address in the label
                       4032
                       4033  *        Fetch the Interleave pointer
                       4034
066.124  315 320 065   4035           CALL    SUBFUN
066.127  002           4036           DB      INI,DBI
066.130  042 045 070   4037           SHLD    DBI              Save directory block interleave
                       4038
066.133  247           4039           ANA     A                Clear any errors
066.134  311           4040           RET
                       4041
066.135  315 136 031   4042  SVP2     CALL    $TYPTX
066.140  012           4043           DB      NL
066.141  124 157 157   4044           DB      'Too many groups on this volume.',ENL
066.201  303 047 054   4045           JMP     RESTART
066.204  311           4046           RET



                       4048  **       VSN     - Volume Serial Number
                       4049  *
                       4050  *        VSN returns the volume serial number
                       4051  *
                       4052  *        ENTRY:  NONE
                       4053  *
                       4054  *        EXIT:   A       = volume serial number
                       4055  *
                       4056  *        USES:   PSW
                       4057  *
                       4058
066.205  072 256 070   4059  VSN      LDA     LABEL+LAB.SER
066.210  247           4060           ANA     A
066.211  311           4061           RET
```

```
   066.212             4064          XTEXT   SAVALL


                       4066X **      $RSTALL - RESTORE ALL REGISTERS.
                       4067X *
                       4068X *       $RSTALL RESTORES ALL THE REGISTERS OFF THE STACK, AND
                       4069X *       RETURNS TO THE PREVIOUS CALLER.
                       4070X *
                       4071X *       ENTRY   (SP) = PSW
                       4072X *               (SP+2) = BC
                       4073X *               (SP+4) = DE
                       4074X *               (SP+6) = HL
                       4075X *               (SP+8) = RET
                       4076X *       EXIT    TO *RET*, REGISTERS RESTORED
                       4077X *       USES    ALL
                       4078X
                       4079X
   031.047             4080X $RSTALL EQU     31047A          IN H17 ROM


                       4082X **      $SAVALL - SAVE ALL REGISTERS ON STACK.
                       4083X *
                       4084X *       $SAVALL SAVES ALL THE REGISTERS ON THE STACK.
                       4085X *
                       4086X *       ENTRY   NONE
                       4087X *       EXIT    (SP) = PSW
                       4088X *               (SP+2) = BC
                       4089X *               (SP+4) = DE
                       4090X *               (SP+6) = HL
                       4091X *       USES    H,L
                       4092X
                       4093X
   031.054             4094X $SAVALL EQU     31054A          IN H17 ROM
   066.212             4095          XTEXT   BITC


                       4097X **      BITC    -  BIT CLEAR
                       4098X *
                       4099X *       BITC CLEARS THE SPECIFIED BIT IN THE ACCUMULATOR.
                       4100X *
                       4101X *       ENTRY:  A       = ORIGINAL  A
                       4102X *               B       = BIT NUMBER TO CLEAR ( 7=HIGH,...,0=LOW )
                       4103X *
                       4104X *       EXIT:   A       = ORIGINAL  A WITH BIT(B) CLEARED
                       4105X *
                       4106X *       USES:   PSW
                       4107X *
                       4108X
   066.212  305        4109X BITC    PUSH    B
                       4110X
   066.213  365        4111X         PUSH    PSW
   066.214  004        4112X         INR     B
```

```
    066.215  076 177      4113X          MVI     A,01111111B
    066.217  007          4114X BITC1    RLC
    066.220  005          4115X          DCR     B
    066.221  302 217 066  4116X          JNZ     BITC1
                          4117X
    066.224  117          4118X          MOV     C,A
    066.225  361          4119X          POP     PSW
    066.226  241          4120X          ANA     C
                          4121X
    066.227  301          4122X          POP     B
    066.230  311          4123X          RET
    066.231               4124           XTEXT   BITS



                          4126X **       BITS     -  BIT SET
                          4127X *
                          4128X *        BITS SETS THE SPECIFIED BIT IN THE ACCUMULATOR.
                          4129X *
                          4130X *        ENTRY:  A        = ORIGINAL  A
                          4131X *                B        = NUMBER OF BIT TO SET ( 7=HIGH,...,0=LOW )
                          4132X *
                          4133X *        EXIT:   A        = ORIGINAL  A  WITH  BIT(B) SET
                          4134X *
                          4135X *        USES:   PSW
                          4136X *
                          4137X
    066.231  305          4138X BITS     PUSH    B
                          4139X
    066.232  365          4140X          PUSH    PSW
    066.233  076 200      4141X          MVI     A,10000000B
    066.235  004          4142X          INR     B
    066.236  007          4143X BITS1    RLC
    066.237  005          4144X          DCR     B
    066.240  302 236 066  4145X          JNZ     BITS1
                          4146X
    066.243  117          4147X          MOV     C,A
    066.244  361          4148X          POP     PSW
    066.245  261          4149X          ORA     C
                          4150X
    066.246  301          4151X          POP     BC
    066.247  311          4152X          RET
    066.250               4153           XTEXT   CCO



                          4155X **       $CCO - CLEAR CONTROL-O
                          4156X *
                          4157X *        $CCO IS CALLED TO CLEAR THE EFFECT OF THE CTL-O CHARACTER.
                          4158X *
                          4159X *        ENTRY   NONE
                          4160X *        EXIT    NONE
                          4161X *        USES    NONE
                          4162X
```

```
                       4163X
066.250  315 054 031   4164X $CCO     CALL    $SAVALL         SAVE REGISTERS
066.253  076 004       4165X          MVI     A,I.CONFL
066.255  001 001 000   4166X          LXI     B,CO.FLG        CLEAR CO.FLG
066.260  377 006       4167X          DB      SYSCALL,.CONSL
066.262  303 047 031   4168X          JMP     $RSTALL         RESTORE REGISTERS AND RETURN
066.265                4169           XTEXT   CDEHL



                       4171X **       $CDEHL - COMPARE (DE) TO (HL)
                       4172X *
                       4173X *        $CDEHL COMPARES (DE) TO (HL) FOR EQUALITY.
                       4174X *
                       4175X *        ENTRY   NONE
                       4176X *        EXIT    'Z' SET IF (DE) = (HL)
                       4177X *        USES    A,F
                       4178X
                       4179X
030.216                4180X $CDEHL   EQU     30216A          IN H17 ROM
066.265                4181           XTEXT   CHL



                       4183X **       $CHL - COMPLEMENT (HL).
                       4184X *
                       4185X *        (HL) = -(HL)            TWO'S COMPLEMENT
                       4186X *
                       4187X *        ENTRY   NONE
                       4188X *        EXIT    NONE
                       4189X *        USES    A,F,H,L
                       4190X
                       4191X
030.224                4192X $CHL     EQU     30224A          IN H17 ROM
066.265                4193           XTEXT   DTB



                       4195X **       $DTB - DELETE TRAILING BLANKS.
                       4196X *
                       4197X *        $DTB DELETES THE TRAILING BLANKS FROM A CODED LINE.
                       4198X *
                       4199X *        ENTRY   (HL) = LINE FWA
                       4200X *        EXIT    (A) = LENGTH OF RESULT (ENCLUDING 00 TERMINATOR BYTE)
                       4201X *        USES    A,F
                       4202X
                       4203X
066.265  325           4204X $DTB     PUSH    D               SAVE (DE)
066.266  124           4205X          MOV     D,H
066.267  135           4206X          MOV     E,L             (DE) = FWA
066.270  033           4207X          DCX     D               (DE) = FWA-1
066.271  176           4208X $DTB1    MOV     A,M
066.272  043           4209X          INX     H
```

```
   066.273  247              4210X          ANA     A               FIND END OF LINE
   066.274  302 271 066      4211X          JNZ     $DTB1
   066.277  053              4212X          DCX     H               (HL) = ADDRESS OF TERMINATING ZERO BYTE
                             4213X
                             4214X *        GOT END OF LINE. DELETE TRAILING BLANKS
                             4215X
   066.300  053              4216X $DTB2    DCX     H               BACKUP ONE CHARACTER
   066.301  315 216 030      4217X          CALL    $CDEHL
   066.304  312 315 066      4218X          JE      $DTB3           GONE PAST FRONT OF LINE, MUST BE ALL BLANKS
   066.307  176              4219X          MOV     A,M
   066.310  376 040          4220X          CPI     ' '
   066.312  312 300 066      4221X          JE      $DTB2           GOT BLANK
                             4222X
                             4223X *        HAVE TRIMED LINE. COMPUTE LENGTH
                             4224X
   066.315  043              4225X $DTB3    INX     H
   066.316  066 000          4226X          MVI     M,0             TERMINATE LINE
   066.320  175              4227X          MOV     A,L
   066.321  223              4228X          SUB     E               (A) = LENGTH +1 (FOR 00 BYTE)
   066.322  353              4229X          XCHG
   066.323  043              4230X          INX     H               (HL) = LINE FWA
   066.324  321              4231X          POP     D               RESTORE (DE)
   066.325  311              4232X          RET
   066.326                   4233           XTEXT   DU66



                             4235X **       $DU66 - UNSIGNED 16 / 16 DIVIDE.
                             4236X *
                             4237X *        (HL) = (BC)/(DE)
                             4238X *
                             4239X *        ENTRY   (BC), (DE) PRESET
                             4240X *        EXIT    (HL) = RESULT
                             4241X *                (DE) = REMAINDER
                             4242X *        USES    ALL
                             4243X
                             4244X
   030.106                   4245X $DU66    EQU     30106A          IN H17 ROM
   066.326                   4246           XTEXT   HLIHL



                             4248X **       $HLIHL - LOAD HL INDIRECT THROUGH HL.
                             4249X *
                             4250X *        (HL) = ((HL))
                             4251X *
                             4252X *        ENTRY   NONE
                             4253X *        EXIT    NONE
                             4254X *        USES    A,H,L
                             4255X
   030.211                   4256X $HLIHL   EQU     30211A          IN H17 ROM
   066.326                   4257           XTEXT   IDN
```

```
                         4259X **       $IDN - INPUT DECIMAL NUMBER.
                         4260X *
                         4261X *        $IDN IS CALLED TO INPUT A DECIMAL NUMBER FROM THE CONSOLE.
                         4262X *
                         4263X *        AN ENTIRE LINE IS ACCEPTED, AND THEN THE NUMBER IS DECODED.
                         4264X *
                         4265X *        RUBOUTS AND BACKSPACES MAY BE USED DURING ENTRY.
                         4266X *
                         4267X *        ENTRY   NONE
                         4268X *        EXIT    'C' CLEAR IF OK
                         4269X *                 (HL) = NUMBER
                         4270X *                'C' SET IF SOME ERROR
                         4271X *        USES    ALL
                         4272X
                         4273X
066.326  041 064 070     4274X $IDN     LXI     H,ITLA
066.331  315 076 067     4275X          CALL    $RTL            READ LINE
066.334  303 274 067     4276X          JMP     $PDD            PACK DECIMAL DIGITS
066.337                  4277           XTEXT   ILDEHL



                         4279X **       ILDEHL  -  INDEXED LOAD OF DE FROM HL
                         4280X *
                         4281X *        'DE' GET THE FULL WORD VALUE POINTED TO BY 'HL', AND 'HL' IS
                         4282X *        INCREMENTED BY TWO.
                         4283X *
                         4284X *        ENTRY:  HL       = ADDRESS OF FULL WORD VALUE
                         4285X *
                         4286X *        EXIT:   DE       = (HL)
                         4287X *                HL       = HL + 2
                         4288X *
                         4289X *        USES:   DE
                         4290X *
                         4291X
066.337  136             4292X ILDEHL   MOV     E,M
066.340  043             4293X          INX     H
066.341  126             4294X          MOV     D,M
066.342  043             4295X          INX     H
066.343  311             4296X          RET
066.344                  4297           XTEXT   INDL



                         4299X **       $INDL - INDEXED LOAD.
                         4300X *
                         4301X *        $INDL LOADS DE WITH THE TWO BYTES AT (HL)+DISPLACMENT
                         4302X *
                         4303X *        THIS ACTS AS AN INDEXED FULL WORD LOAD.
                         4304X *
                         4305X *        (DE) = ( (HL) + DSPLACEMENT )
                         4306X *
                         4307X *        ENTRY   ((RET)) = DISPLACMENT (FULL WORD)
                         4308X *                (HL) = TABLE ADDRESS
```

```
                                 4309X *       EXIT     TO (RET+2)
                                 4310X *       USES     A,F,D,E
                                 4311X
                                 4312X
        030.234                  4313X $INDL   EQU      30234A           IN H17 ROM
        066.344                  4314          XTEXT    INDXX


                                 4316X **      $INDLB  -  INDEXED LOAD BYTE
                                 4317X *
                                 4318X *       BYTE INDEXED LOAD PRIMITIVE
                                 4319X *
                                 4320X *       ENTRY:   HL       = BASE ADDRESS
                                 4321X *                (RET)    = FULL WORD RELOCATION
                                 4322X *
                                 4323X *       EXIT:    A        = ( HL + (RET) )
                                 4324X *
                                 4325X *       USES:    A
                                 4326X *
                                 4327X
        066.344  353             4328X $INDLB  XCHG                      DE = BASE
        066.345  343             4329X         XTHL                      SAVE  .DE.
        066.346  325             4330X         PUSH     D                SAVE  BASE
        066.347  305             4331X         PUSH     B                SAVE  .BC.
                                 4332X
        066.350  116             4333X         MOV      C,M
        066.351  043             4334X         INX      H
        066.352  106             4335X         MOV      B,M              BC = OFFSET
        066.353  043             4336X         INX      H                HL =  .RET.
                                 4337X
        066.354  353             4338X         XCHG                      HL = BASE
        066.355  011             4339X         DAD      B                HL = BASE + OFFSET
        066.356  176             4340X         MOV      A,M              A  = ( BASE + OFFSET )
        066.357  353             4341X         XCHG                      HL =  .RET.
                                 4342X
        066.360  301             4343X         POP      B                RESTORE  .BC.
        066.361  321             4344X         POP      D                RESTORE  BASE
        066.362  343             4345X         XTHL                      HL = .DE. ; (SP) = .RET.
        066.363  353             4346X         XCHG                      DE = .DE. ; HL = BASE
        066.364  311             4347X         RET


                                 4349X **      $INDS   -  INDEXED STORE
                                 4350X *
                                 4351X *       INDEXED STORE PRIMITIVE.
                                 4352X *
                                 4353X *       ENTRY:   HL = BASE ADDRESS
                                 4354X *                DE = VALUE TO STORE
                                 4355X *
                                 4356X *       EXIT:    ( HL + (RET) ) = DE
                                 4357X *
                                 4358X *       USES:    NONE
```

```
                             4359X *
                             4360X
       066.365  315 263 067  4361X $INDS   CALL    XCHGBC
       066.370  343          4362X         XTHL                    SAVE  .BC.
       066.371  325          4363X         PUSH    D
       066.372  315 337 066  4364X         CALL    ILDEHL          DE = OFFSET
       066.375  315 263 067  4365X         CALL    XCHGBC          BC = .RET.
       067.000  353          4366X         XCHG                    DE = BASE ; HL = OFFSET
       067.001  031          4367X         DAD     D               HL = BASE + OFFSET
       067.002  353          4368X         XCHG
       067.003  343          4369X         XTHL                    SAVE  BASE
       067.004  353          4370X         XCHG                    DE = VALUE
       067.005  315 042 067  4371X         CALL    ISDEHL
       067.010  341          4372X         POP     H               HL = BASE
       067.011  315 263 067  4373X         CALL    XCHGBC
       067.014  343          4374X         XTHL                    RESTORE  .BC.
       067.015  315 263 067  4375X         CALL    XCHGBC
       067.020  311          4376X         RET



                             4378X **      $INDSB  -  INDEXED BYTE STORE
                             4379X *
                             4380X *       INDEXED BYTE STORE.
                             4381X *
                             4382X *       ENTRY:  A        = VALUE TO STORE
                             4383X *               HL       = BASE ADDRESS
                             4384X *               (RET)    = OFFSET
                             4385X *
                             4386X *       EXIT:   NONE
                             4387X *
                             4388X *       USES:   PSW
                             4389X *
                             4390X
       067.021  353          4391X $INDSB  XCHG                    DE = BASE
       067.022  343          4392X         XTHL                    SAVE  .DE.
       067.023  325          4393X         PUSH    D               SAVE  BASE
       067.024  305          4394X         PUSH    B               SAVE  .BC.
                             4395X
       067.025  116          4396X         MOV     C,M
       067.026  043          4397X         INX     H
       067.027  106          4398X         MOV     B,M             BC = OFFSET
       067.030  043          4399X         INX     H               HL =  .RET.
                             4400X
       067.031  353          4401X         XCHG                    HL = BASE
       067.032  011          4402X         DAD     B               HL = BASE + OFFSET
       067.033  167          4403X         MOV     M,A             ( BASE + OFFSET ) = A
       067.034  353          4404X         XCHG
                             4405X
       067.035  301          4406X         POP     B               RESTORE  .BC.
       067.036  321          4407X         POP     D               RESTORE  BASE
       067.037  343          4408X         XTHL                    HL = .DE. ; (SP) = .RET.
       067.040  353          4409X         XCHG                    DE = .DE. ; HL = BASE
       067.041  311          4410X         RET
       067.042               4411          XTEXT   ISDEHL
```

```
                          4413X **       ISDEHL  -  INDEXED STORE OF DE AT HL
                          4414X *
                          4415X *        STORE 'DE' AT THE ADDRESS POINTED TO BY 'HL', AND INCREMENT 'HL'
                          4416X *        BY 2.
                          4417X *
                          4418X *        ENTRY:  DE       = VALUE
                          4419X *                HL       = ADDRESS OF VALUE
                          4420X *
                          4421X *        EXIT:   (HL)     = DE
                          4422X *                HL       = HL + 2
                          4423X *
                          4424X *        USES:   HL
                          4425X *
                          4426X
 067.042  163             4427X ISDEHL   MOV     M,E
 067.043  043             4428X          INX     H
 067.044  162             4429X          MOV     M,D
 067.045  043             4430X          INX     H
 067.046  311             4431X          RET
 067.047                  4432           XTEXT   MLU



                          4434X **       MLU - MAP LOWER CASE LINE TO UPPER CASE.
                          4435X *
                          4436X *        MLU MAPS THE LOWER CASE ALPHABETICS IN A LINE TO UPPER CASE.
                          4437X *
                          4438X *        ENTRY   (HL)  = LINE FWA
                          4439X *        EXIT    NONE
                          4440X *        USES    NONE
                          4441X
                          4442X
 067.047  365             4443X $MLU     PUSH    PSW               SAVE (PSW)
 067.050  345             4444X          PUSH    H                 SAVE FWA
 067.051  053             4445X          DCX     H                 ANTICIPATE INX H
 067.052  043             4446X $MLU1    INX     H
 067.053  176             4447X          MOV     A,M               (A)= CHARACTER
 067.054  315 071 052     4448X          CALL    $MCU              MAP CHAR TO UPPER
 067.057  167             4449X          MOV     M,A
 067.060  247             4450X          ANA     A
 067.061  302 052 067     4451X          JNZ     $MLU1             MORE TO GO
 067.064  341             4452X          POP     H                 RESTORE (HL)
 067.065  361             4453X          POP     PSW               RESTORE (PSW)
 067.066  311             4454X          RET
 067.067                  4455           XTEXT   RTL
```

```
                    4457X **      $RTL - READ TEXT LINE.
                    4458X *
                    4459X *       $RTL READS A LINE FROM THE TERMINAL.
                    4460X *
                    4461X *       CHARACTER ARE ACCEPTED FROM THE TERMINAL, RUBOUT AND BACKSPACE
                    4462X *       CHARACTERS ARE PROCESSED. WHEN A CARRIAGE RETURN IS ENTERED,
                    4463X *       $RTL RETURNS.
                    4464X *
                    4465X *       ENTRY   (HL) = BUFFER FWA
                    4466X *       EXIT    'C' CLEAR IF OK
                    4467X *                DATA IN BUFFER
                    4468X *                (A) = TEXT LENGTH
                    4469X *                'C' SET IF CTL-D STRUCK
                    4470X *       USES    A,F
                    4471X
                    4472X
067.067  315 076 067 4473X $RTL.  CALL    $RTL            $RTL IN UPPER CASE
067.072  330         4474X        RC                      CTL-D
067.073  303 047 067 4475X        JMP     $MLU            MAP LINE TO UPPER CASE
                    4476X
067.076              4477X $RTL   EQU     *
067.076  345         4478X        PUSH    H               SAVE FWA
067.077  315 134 067 4479X $RTL1  CALL    $RCHAR
067.102  376 004     4480X        CPI     CTLD
067.104  312 131 067 4481X        JE      $RTL2           CTL-D STRUCK
067.107  167         4482X        MOV     M,A
067.110  043         4483X        INX     H
067.111  376 012     4484X        CPI     NL
067.113  302 077 067 4485X        JNE     $RTL1
067.116  053         4486X        DCX     H
067.117  066 000     4487X        MVI     M,0
067.121  043         4488X        INX     H
                    4489X
                    4490X *       ALL DONE. COMPUTE LENGTH
                    4491X
067.122  353         4492X        XCHG                    (DE) = LWA+1
067.123  343         4493X        XTHL                    (HL) = FWA
067.124  173         4494X        MOV     A,E
067.125  225         4495X        SUB     L               (A) = LENGTH
067.126  247         4496X        ANA     A               CLEAR CARRY
067.127  321         4497X        POP     D               RESTORE (DE)
067.130  311         4498X        RET
                    4499X
                    4500X *       CTL-D STRUCK
                    4501X
067.131  341         4502X $RTL2  POP     H               (HL) = FWA
067.132  067         4503X        STC
067.133  311         4504X        RET
067.134              4505         XTEXT   WER
```

```
                                  4507X **       $WER - WRITE ENABLE RAM.
                                  4508X *
                                  4509X *        $WER IS CALLED TO ENABLE WRITTING TO THE H17 CONTROLLER'S
                                  4510X *        RAM AREA.
                                  4511X *
                                  4512X *        ENTRY   NONE
                                  4513X *        EXIT    NONE
                                  4514X *        USES    NONE
                                  4515X
                                  4516X
     031.241                      4517X $WER     EQU     31241A          IN H17 ROM



                                  4519X **       $WDR - WRITE DISABLE RAM.
                                  4520X *
                                  4521X *        $WDR IS CALLED TO DISABLE WRITTING TO THE H17 CONTROLLER'S
                                  4522X *        RAM AREA.
                                  4523X *
                                  4524X *        ENTRY   NONE
                                  4525X *        EXIT    NONE
                                  4526X *        USES    NONE
                                  4527X
                                  4528X
     031.222                      4529X $WDR     EQU     31222A          IN H17 ROM
     067.134                      4530           XTEXT   RCHAR



                                  4532X **       $RCHAR - READ SINGLE CHARACTER FROM CONSOLE.
                                  4533X *
                                  4534X *        ENTRY   NONE
                                  4535X *        EXIT    (A) = CHARACTER
                                  4536X *        USES    A,F
                                  4537X
                                  4538X
     067.134  377 001             4539X $RCHAR   DB      SYSCALL,.SCIN
     067.136  332 134 067         4540X          JC      $RCHAR          NOT READY
     067.141  311                 4541X          RET
                                  4542X
     067.142  377 002             4543X $WCHAR   DB      SYSCALL,.SCOUT
     067.144  311                 4544X          RET
     067.145                      4545           XTEXT   SOB



                                  4547X **       $SOB - SKIP OVER BLANKS.
                                  4548X *
                                  4549X *        $SOB IS CALLED TO SKIP AN ARBITRARILY LONG STRING OF BLANKS AND TABS.
                                  4550X *
                                  4551X *        ENTRY   (HL) = FWA OF (POSSIBLE) BLANK STRING
                                  4552X *        EXIT    (HL) = LWA+1 OF BLANK STRING (UNCHANGED IF NO BLANKS)
                                  4553X *                (A) = FIRST NON-BLANK, NON-TAB CHARACTER EEN
```

```
                              4554X *       USES     A,F,H,L
                              4555X
                              4556X
    067.145  053              4557X $SOB    DCX      H               PRE-DECREMENT
    067.146  043              4558X $SOB1   INX      H
    067.147  176              4559X         MOV      A,M
    067.150  376 040          4560X         CPI      ' '
    067.152  312 146 067      4561X         JE       $SOB1           GOT BLANK
    067.155  376 011          4562X         CPI      TAB
    067.157  312 146 067      4563X         JE       $SOB1           GOT TAB
    067.162  311              4564X         RET
    067.163                   4565          XTEXT    TBLS


                              4567X **      $TBLS - TABLE SEARCH
                              4568X *
                              4569X *       TABLE FORMAT
                              4570X *
                              4571X *       DB       KEY1,VAL1,
                              4572X *       .        .
                              4573X *       .        .
                              4574X *       DB       KEYN,VALN
                              4575X *       DB       0
                              4576X *
                              4577X *       ENTRY    (A) = PATTERN
                              4578X *                (H,L) = TABLE FWA
                              4579X *       EXIT     (A) = PATTERN IF FOUND
                              4580X *                'Z' SET IF FOUND
                              4581X *                'Z' CLEAR IF NOT FOUND OR PATTERN=0          /78.10.GC/
                              4582X *       USES     A,F,H,L
                              4583X
                              4584X
    067.163  305              4585X $TBLS   PUSH     B
    067.164  376 000          4586X         CPI      0                                            /78.10.GC/
    067.166  312 210 067      4587X         JZ       TBL2                                         /78.10.GC/
    067.171  107              4588X         MOV      B,A
    067.172  176              4589X TBL1    MOV      A,M             (A) = CHARACTER
    067.173  043              4590X         INX      H
    067.174  270              4591X         CMP      B
    067.175  312 212 067      4592X         JZ       TBL3            IF MATCH
    067.200  247              4593X         ANA      A
    067.201  043              4594X         INX      H               SKIP PAST
    067.202  302 172 067      4595X         JNZ      TBL1            IF NOT END OF TABLE
    067.205  053              4596X         DCX      H
    067.206  053              4597X         DCX      H
    067.207  257              4598X         XRA      A               SET TO ZERO FOR OLD USERS    /78.10.GC/
    067.210  376 001          4599X TBL2    CPI      1               CLEAR ZERO                   /78.10.GC/
                              4600X
                              4601X *       DONE
                              4602X
    067.212  301              4603X TBL3    POP      B
    067.213  311              4604X         RET
    067.214                   4605          XTEXT    TBRA
```

```
                                 4607X **       $TBRA - BRANCH RELATIVE THOUGH TABLE.
                                 4608X *
                                 4609X *        $TBRA USES THE SUPPLIED INDEX TO SELECT A BYTE FROM THE
                                 4610X *        JUMP TABLE. THE CONTENTS OF THIS BYTE ARE ADDED TO THE
                                 4611X *        ADDRESS OF THE BYTE, YEILDING THE PROCESSOR ADDRESS.
                                 4612X *
                                 4613X *        CALL    $TBRA
                                 4614X *        DB      LAB1-*          INDEX = 0 FOR LAB1
                                 4615X *        DB      LAB2-*          INDEX = 1 FOR LAB2
                                 4616X *        DB      LABN-*          INDEX = N-1 FOR LABN
                                 4617X *
                                 4618X *        ENTRY   (A) = INDEX
                                 4619X *                (RET) = TABLE FWA
                                 4620X *        EXIT    TO COMPUTED ADDRESS
                                 4621X *        USES    F,H,L
                                 4622X
                                 4623X
        031.076                  4624X $TBRA    EQU     31076A          IN H17 ROM
        067.214                  4625           XTEXT   TYPCH



                                 4627X **       $TYPCH - TYPE SINGLE CHARACTER.
                                 4628X *
                                 4629X *        ENTRY   (RET) = CHARACTER
                                 4630X *        EXIT    TO (RET)+1
                                 4631X *                (A) = CHARACTER TYPED
                                 4632X
                                 4633X
        067.214  343             4634X $TYPCH   XTHL                    (HL) = RETURN ADDRESS
        067.215  176             4635X          MOV     A,M             (A) = CHARACTER
        067.216  043             4636X          INX     H
        067.217  343             4637X          XTHL                    RESTORE ADVANCED EXIT ADDRESS
                                 4638X
                                 4639X **       $TYPC. - TYPE SINGLE CHARACTER.
                                 4640X *
                                 4641X *        ENTRY   (A) = CHARACTER
                                 4642X *        EXIT    TO (RET)
                                 4643X
        067.220  377 002         4644X $TYPC.   DB      SYSCALL,.SCOUT
        067.222  311             4645X          RET
        067.223                  4646           XTEXT   TYPT2



                                 4648X **       $TYPTX - TYPE TEXT.
                                 4649X *
                                 4650X *        $TYPTX IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE.
                                 4651X *
                                 4652X *        IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED,
                                 4653X *        A BYTE WITH THE 200Q BIT SET IS THE LAST BYTE IN THE MESSAGE.
                                 4654X *
                                 4655X *        ENTRY   (RET) = TEXT
                                 4656X *        EXIT    TO (RET+LENGTH)
```

```
                            4657X *       USES    A,F
                            4658X
                            4659X
    031.136                 4660X $TYPTX  EQU     31136A          IN H17 ROM
                            4661X
    031.144                 4662X $TYPTX. EQU     31144A          IN H17 ROM
    000.001                 4663  $CMP$   EQU     1                       NO COMPRESSED TEXT
    067.223                 4664          XTEXT   TYPLN



                            4666X **      $TYPLN - TYPE LINE.
                            4667X *
                            4668X *       $TYPLN IS CALLED TO TYPE A LINE OF TEXT. ZERO BYTES ARE
                            4669X *       TAKEN AS CRLF (WITH THE PROPER PADDING)
                            4670X *
                            4671X *       CALL    $TYPLN
                            4672X *       DB      N               BYTE COUNT OF FOLLOWING MESSAGE
                            4673X *       DB      'N-CHARACTER MESSAGE'
                            4674X *
                            4675X *       ENTRY   (RET) = TEXT COUNT
                            4676X *               (RET)+1 - (RET)+N = TEXT
                            4677X *       EXIT    TO (RET)+N+1
                            4678X *       USES    A,F
                            4679X *
                            4680X
                            4681X
    067.223  343            4682X $TYPLN. XTHL                    (H,L) = COUNT ADDRESS
    067.224  176            4683X         MOV     A,M             (A) = COUNT
    067.225  043            4684X         INX     H               (H,L)  = TEXT ADDRESS
    067.226  345            4685X         PUSH    H               SAVE TEXT FWA
    067.227  315 072 030    4686X         CALL    $DADA           CALCULATE RETURN ADDRESS
    067.232  343            4687X         XTHL                    (HL) = TEXT ADDRE
    067.233  315 241 067    4688X         CALL    $TYPL.          OUTPUT LINE
    067.236  341            4689X         POP     H               (HL) = RETURN ADDRESS
    067.237  343            4690X         XTHL                    RESTORE (HL); SET RETURN ADDRESS
    067.240  311            4691X         RET
                            4692X
                            4693X **      $TYPL. - TYPE LINE.
                            4694X *
                            4695X *       ENTRY   (HL) = ADDRESS
                            4696X *               (A) = COUNT
                            4697X *       EXIT    NONE
                            4698X *       USES    A,F,H,L
                            4699X
    067.241                 4700X $TYPL.  EQU     *
    067.241  247            4701X         ANA     A
    067.242  310            4702X         RZ                      NOTHING TO TYPE
    067.243  365            4703X         PUSH    PSW             SAVE COUNT
    067.244  176            4704X         MOV     A,M             (A) = CHARACTER
    067.245  043            4705X         INX     H
    067.246  247            4706X         ANA     A
    000.001                 4707X         IF      $CMP$           IF HAVE COMPRESSED SPACES
                            4708X         JM      TPL2            IS COMPRESSED SPACE
                            4709X         ENDIF
    067.247  314 033 070  4710X          CZ      $CRLF
    067.252  315 220 067  4711X          CALL    $TYPC.          TYPE CHARACTER
    067.255  361          4712X TPL1     POP     PSW
    067.256  075          4713X          DCR     A
    067.257  302 241 067  4714X          JNZ     $TYPL.
    067.262  311          4715X          RET
    000.001               4716X          IF      $CMP$           IF COMPRESSED TEXT
                          4717X
                          4718X *        HAVE COMPRESSED SPACE.
                          4719X
                          4720X TPL2     DCR     A
                          4721X          CP      $TYPCH          TYPE 00 IF CHARACTER WAS 200Q
                          4722X          DB      0
                          4723X          ANA     A               SET CODES
                          4724X TPL3     JP      TPL1            ALL EXPANDED
                          4725X          PUSH    PSW             SAVE COUNT
                          4726X          CALL    $TYPCH
                          4727X          DB      ' '
                          4728X          POP     PSW
                          4729X          DCR     A
                          4730X          JMP     TPL3
                          4731X          ENDIF
    067.263               4732           XTEXT   UDD



                          4734X **       $UDD - UNPACK DECIMAL DIGITS.
                          4735X *
                          4736X *        UDD CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF
                          4737X *        DECIMAL DIGITS. THE RESULT IS ZERO FILLED.
                          4738X *
                          4739X *        ENTRY   (B,C) = ADDRESS VALUE
                          4740X *                (A) = DIGIT COUNT
                          4741X *                (H,L) = MEMORY ADDRESS
                          4742X *        EXIT    (HL) = (HL) + (A)
                          4743X *        USES    ALL
                          4744X
                          4745X
    031.157               4746X $UDD     EQU     31157A          IN H17 ROM
    067.263               4747           XTEXT   XCHGBC



                          4749X **       XCHGBC  -  XCHG BC
                          4750X *
                          4751X *        EXCHANGE THE 'BC' REGISTER PAIR WITH THE 'HL' REGISTER PAIR.
                          4752X *
                          4753X *        ENTRY:  BC      = ORIGINAL BC
                          4754X *                HL      = ORIGINAL HL
                          4755X *
                          4756X *        EXIT:   BC      = ORIGINAL HL
                          4757X *                HL      = ORIGINAL BC
                          4758X *
                          4759X *        USES:   BC,HL
```

```
                      4760X *
                      4761X
067.263  365          4762X XCHGBC  PUSH    PSW
067.264  170          4763X         MOV     A,B
067.265  104          4764X         MOV     B,H
067.266  147          4765X         MOV     H,A
067.267  171          4766X         MOV     A,C
067.270  115          4767X         MOV     C,L
067.271  157          4768X         MOV     L,A
067.272  361          4769X         POP     PSW
067.273  311          4770X         RET
067.274               4771          XTEXT   PDD


                      4773X **      $PDD - PACK DECIMAL DIGITS.
                      4774X *
                      4775X *       $PDD PACKS A STRING OF DECIMAL DIGITS INTO A DECIMAL INTEGER.
                      4776X *
                      4777X *       THE CHARACTERS MUST BE IN MEMORY, AND BE IMMEDIATELY FOLLOWED BY A
                      4778X *       00 BYTE.
                      4779X *
                      4780X *       ENTRY   (HL) = ADDRESS OF CHARACTERS
                      4781X *       EXIT    'C' CLEAR IF OK
                      4782X *                (HL) = NUMBER
                      4783X *               'C' SET IF ERROR
                      4784X *       USES    A,F,D,E,H,L
                      4785X
                      4786X
067.274  353          4787X $PDD    XCHG                    (DE) = TEXT ADDRESS
067.275  041 000 000  4788X         LXI     H,0             (HL) = ACCUM
                      4789X
067.300  032          4790X $PDD1   LDAX    D
067.301  023          4791X         INX     D               ADVANCE ADDRESS
067.302  247          4792X         ANA     A
067.303  310          4793X         RZ                      ALL DONE
067.304  326 060      4794X         SUI     '0'
067.306  330          4795X         RC                      TOO SMALL
067.307  376 012      4796X         CPI     10
067.311  077          4797X         CMC
067.312  330          4798X         RC                      TOO SMALL
067.313  325          4799X         PUSH    D               SAVE (DE)
067.314  353          4800X         XCHG
067.315  315 324 030  4801X         CALL    $MU10
067.320  321          4802X         POP     D
067.321  330          4803X         RC                      OVERFLOW
067.322  205          4804X         ADD     L
067.323  157          4805X         MOV     L,A
067.324  076 000      4806X         MVI     A,0
067.326  214          4807X         ADC     H
067.327  147          4808X         MOV     H,A
067.330  322 300 067  4809X         JNC     $PDD1           NOT OVERFLOW
067.333  311          4810X         RET
067.334               4811          XTEXT   MU10
```

```
                           4813X **       $MU10 - MULTIPLY UNSIGNED 16 BIT QUANTITY BY 10.
                           4814X *
                           4815X *        (HL) = (DE)*10
                           4816X *
                           4817X *        ENTRY    (DE) = MULTIPLIER
                           4818X *        EXIT     'C' CLEAR IF OK
                           4819X *                  (HL) = PRODUCT
                           4820X *                 'C' SET IF ERROR
                           4821X *        USES     D,E,H,L,F
                           4822X
                           4823X
   030.324                 4824X $MU10    EQU      30324A          IN H17 ROM
   067.334                 4825           XTEXT    ITL


                           4827X **       $ITL - INPUT TEXT LINE.
                           4828X *
                           4829X *        $ITL INPUTS A LINE FROM THE TERMINAL.
                           4830X *
                           4831X *        CHARACTER ARE ACCEPTED FROM THE TERMINAL, RUBOUT AND BACKSPACE
                           4832X *        CHARACTERS ARE PROCESSED. WHEN A CARRIAGE RETURN IS ENTERED,
                           4833X *        $ITL RETURNS.
                           4834X *
                           4835X *        ENTRY    NONE
                           4836X *        EXIT     (HL) = $ITLA
                           4837X *                 (A) = TEXT LENGTH
                           4838X *        USES     A,F,H,L
                           4839X
                           4840X
   067.334  315 342 067    4841X $ITL.    CALL     $ITL            INPUT LINE IN UPPER CASE
   067.337  303 047 067    4842X          JMP      $MLU            MAP LINE TO UPPER
                           4843X
   067.342  041 064 070    4844X $ITL     LXI      H,ITLA
   067.345  303 076 067    4845X          JMP      $RTL            READ TEXT LINE
   067.350                 4846           XTEXT    TDD


                           4848X **       $TDD - TYPE DECIMAL DIGITS.
                           4849X *
                           4850X *        $TDD TYPES A 16 BIT VALUE AS 1 TO 5 DECIMAL DIGITS.
                           4851X *
                           4852X *        ENTRY    (D,E) = VALUE
                           4853X *                 (A) = DIGIT COUNT
                           4854X *        EXIT     VALUE TYPED.
                           4855X *        USES     A,B,C,F
                           4856X
                           4857X
   067.350  076 005        4858X $TDD.    MVI      A,5
   067.352  345            4859X $TDD     PUSH     H
   067.353  365            4860X TDD1     PUSH     PSW
   067.354  041 017 070    4861X          LXI      H,TDDA-2
   067.357  007            4862X          RLC                      (A) = DIGIT NUMBER*2
```

```
     067.360  315 101 030  4863X          CALL     $DADA.
     067.363  176          4864X          MOV      A,M
     067.364  043          4865X          INX      H
     067.365  146          4866X          MOV      H,M
     067.366  157          4867X          MOV      L,A              (HL) = MULTIPLE OF 10
     067.367  353          4868X          XCHG                      (DE) = DEVISOR, (HL) = VALUE
     067.370  076 377      4869X          MVI      A,377Q
     067.372  031          4870X TDD2     DAD      D
     067.373  074          4871X          INR      A
     067.374  332 372 067  4872X          JC       TDD2             IF MORE TO GO
     067.377  306 060      4873X          ADI      '0'
     070.001  315 220 067  4874X          CALL     $TYPC.           TYPE DIGIT
     070.004  175          4875X          MOV      A,L
     070.005  223          4876X          SUB      E
     070.006  137          4877X          MOV      E,A              REMOVE EXTRA SUBTRACTION
     070.007  174          4878X          MOV      A,H
     070.010  232          4879X          SBB      D
     070.011  127          4880X          MOV      D,A
     070.012  361          4881X          POP      PSW
     070.013  075          4882X          DCR      A
     070.014  302 353 067  4883X          JNZ      TDD1             IF MORE DIGITS
     070.017  341          4884X          POP      H
     070.020  311          4885X          RET                       EXIT
                           4886X
     070.021               4887X TDDA     EQU      *
     070.021  377 377      4888X          DW       -1
     070.023  366 377      4889X          DW       -10
     070.025  234 377      4890X          DW       -100
     070.027  030 374      4891X          DW       -1000
     070.031  360 330      4892X          DW       -10000
     070.033               4893           XTEXT    DADA



                           4895X **       $DADA - PERFORM (H,L) = (H,L) + (0,A)
                           4896X *
                           4897X *        ENTRY    (H,L) = BEFORE VALUE
                           4898X *                 (A) = BEFORE VALUE
                           4899X *        EXIT     (H,L) = (H,L) + (0,A)
                           4900X *                 'C' SET IF OVERFLOW
                           4901X *        USES     F,H,L
                           4902X
                           4903X
     030.072               4904X $DADA    EQU      30072A           IN H17 ROM
     070.033               4905           XTEXT    CRLF
```

```
                        4907X **       $CRLF - TYPE CARRIAGE RETURN/ LINE FEED
                        4908X *
                        4909X *        $CRLF IS USED TO GENERATE PADDED CRLF'S.
                        4910X *
                        4911X *        ENTRY   NONE
                        4912X *        EXIT    (A) = 0
                        4913X *        USES    A,F
                        4914X
                        4915X
     070.033  076 012   4916X $CRLF    MVI     A,NL
     070.035  377 002   4917X          DB      SYSCALL,.SCOUT
     070.037  257       4918X          XRA     A
     070.040  311       4919X          RET
```

```
                              4922
                              4923
    070.041  011 102          4924  BUFLIM  DW      RMEML            Buffer Limit                       /80.05.GC/
    070.043  205 070          4925  CRNDEV  DW      DEVTAB           Current Device Pointer             /80.05.GC/
    070.045  000 000          4926  DBI     DW      0                Pointer to dir. block interleave /80.06.sc/
    070.047  000              4927  GRTBLK  DB      0                BLOCK NUMBER CONTAINING GRT
    070.050  000              4928  DIRBLK  DB      0                BLOCK NUMBER OF FIRST DIRECTORY BLOCK
    070.051  000              4929  DIRLBLK DB      0                BLOCK NUMBER OF LAST DIRECTORY BLOCK
    070.052  000 000          4930  MEMLIM  DW      0                Memory Limit                       /80.05.sc/
    070.054  000              4931  NDIRBLK DB      0                Number of Directory Blocks         /80.06.sc/
    070.055  000              4932  NGROUPS DB      0                Number of Valid Groups             /80.05.sc/
    070.056  000              4933  NSPCGRP DB      0                Number of Special Groups (>=2)     /80.06.sc/
    070.057  000              4934  SUNIT   DB      0                System Unit                        /80.09.sc/
    070.060  000              4935  SYMNU   DB      0                SY: Maximum Number of Units        /80.09.sc/
    070.061  000 000          4936  SYPTR   DW      0                SY: Internal Device Table Ptr.     /80.09.sc/
    070.063  000              4937  UNIT    DB      0                Unit Number                        /80.05.sc/
                              4938
    070.064                   4939  LINE    DS      81
    000.120                   4940  LINEL   EQU     *-LINE-1
    070.064                   4941  ITLA    EQU     LINE             USE LINE FOR BUFFER


                              4943  **      DEVTAB  -  Device Table
                              4944  *
                              4945
    070.205                   4946  DEVTAB  DS      0
    070.205                   4947  .       SET     *
                              4948
    000.000                   4949  DVT.NAM EQU     *-.
    070.205  170 170 072      4950          DB      'xx',':',0       Device Name
                              4951
    000.004                   4952  DVT.ADR EQU     *-.
    070.211  000 000          4953          DW      0                Device Table Entry Address
                              4954
    000.006                   4955  DVT.PAR EQU     *-.
    070.213  000 000          4956          DW      0                Parameter File Pointer
                              4957
    000.010                   4958  DVT.PIC EQU     *-.
    070.215  000 000          4959          DW      0                PIC Table Address
                              4960
    000.012                   4961  DVT.ENL EQU     *-.              Device Table Entry Length
                              4962
    070.217                   4963          DS      DVT.MAX-1*DVT.ENL        Room for more device entries
                              4964
    070.255  000              4965  DEVTABE DB      0                Zero at the end of the list
```

```
                              4967  **      DISK LABEL
                              4968
      070.256                 4969  LABEL   DS      0
      000.000                 4970          ERRNZ   *-LABEL-LAB.SER
      070.256  000            4971          DB      0               SERIAL NUMBER
      000.000                 4972          ERRNZ   *-LABEL-LAB.IND
      070.257  000 000        4973          DW      0               CREATION DATE
      000.000                 4974          ERRNZ   *-LABEL-LAB.DIS
      070.261  000 000        4975          DW      0               INDEX OF FIRST DIRECTORY SECTOR
      000.000                 4976          ERRNZ   *-LABEL-LAB.GRT
      070.263  000 000        4977          DW      0               GRT TABLE INDEX
      000.000                 4978          ERRNZ   *-LABEL-LAB.SPG
      070.265  000            4979          DB      0               TWO SECTORS PER GROUP
      000.000                 4980          ERRNZ   *-LABEL-LAB.VLT
      070.266  000            4981          DB      LAB.DAT         VOLUME TYPE
      000.000                 4982          ERRNZ   *-LABEL-LAB.VER
      070.267  040            4983          DB      VERS            VERSION OF INIT17 TO INIT DISK
                              4984
      000.000                 4985          ERRNZ   *-LABEL-LAB.RGT                             /80.06.GC/
      070.270  000 000        4986          DW      0               Sector Address of RGT       /80.06.GC/
                              4987
      000.000                 4988          ERRNZ   *-LABEL-LAB.VPR                             /80.05.sc/
      000.000                 4989          ERRNZ   *-LABEL-LAB.SIZ                             /80.05.sc/
      070.272  000 000        4990          DW      0               Volume Size                 /80.05.sc/
      000.000                 4991          ERRNZ   *-LABEL-LAB.PSS                             /80.05.sc/
      070.274  000 000        4992          DW      0               Physical Sector Size        /80.05.sc/
      000.000                 4993          ERRNZ   *-LABEL-LAB.VFL                             /80.05.sc/
      070.276  000            4994          DB      0               Device dependant flags      /80.05.sc/
      000.000                 4995          ERRNZ   *-LABEL-LAB.VPR-LAB.VPL                     /80.05.sc/
                              4996
      000.000                 4997          ERRNZ   *-LABEL-LAB.LAB
      070.277                 4998          DS      60              LABEL
      070.373  000 000        4999          DB      0,0             END OF LABEL
                              5000
      000.000                 5001          ERRNZ   *-LABEL-LAB.AUX                             /80.09.sc/
      070.375                 5002          DS      LAB.AXL         Reserve Space for Aux. Param  /80.09.sc/
                              5003
      070.376  040 040 040    5004          DB      '    '                                      /80.09.sc/
      071.002  015 012        5005          DB      CR,LF
      071.004  123 131 123    5006          DB      'SYSTEM COPYRIGHT HEATH CO., 10/1977, 79/4'
      071.055  015 012        5007          DB      CR,LF
      071.057  040 102 131    5008          DB      ' BY JGL, 10/1977'
      071.077  057 147 143    5009          DB      '/sc '
      071.103                 5010          DS      0               END OF LABEL
                              5011
                              5012
                              5013
      071.103                 5014  PATCH   DS      64              PATCH AREA
                              5015
```

```
071.203                  5018  OVRLAY  EQU     *               Start of buffers initially code /80.05.GC/


                         5020  **      PRS - PRESET PROGRAM.
                         5021  *
                         5022  *       PRS PERFORMS ANY INITIALIZATION TASKS, AND  INFORMS THE
                         5023  *       USER OF THE FACTS OF LIFE CONCERNING INITIALIZATION/.
                         5024  *
                         5025  *       ENTRY   NONE
                         5026  *       EXIT    TO SYSTEM IF USER CHICKENS OUT
                         5027  *               TO CALLER IF OK
                         5028  *       USES    ALL
                         5029
                         5030
071.203  315 250 066     5031  PRS     CALL    $CCO            CLEAR CTL-O
071.206  315 136 031     5032          CALL    $TYPTX
071.211  012 011 011     5033          DB      NL,TAB,TAB,TAB,'     ','INIT'
071.225  012 011 011     5034          DB      NL,TAB,TAB,TAB,'Version:  ',VERS/16+'0','.',VERS&00001111B+'0'
071.246  012 011 011     5035          DB      NL,TAB,TAB,'        ','Issue:  #50.06.00'
071.300  012             5036          DB      NL
071.301  012 011 124     5037          DB      NL,TAB,'This  routine is used to initialize HDOS floppy'
071.362  040 144 151     5038          DB      ' disks.'
071.371  012 111 164     5039          DB      NL,'It is  a  stand-alone utility,  and  will destroy any'
072.057  040 146 151     5040          DB      ' files on'
072.070  012 164 150     5041          DB      NL,'the disks it initializes.  Do not attempt to use this'
072.156  040 040 160     5042          DB      '  program'
072.167  012 165 156     5043          DB      NL,'until you have studied the appropriate manual.'
072.246  212             5044          DB      ENL
                         5045
072.247  315 250 066     5046  PRS1    CALL    $CCO            CLEAR CTL-O                     /80.05.GC/
072.252  315 136 031     5047          CALL    $TYPTX
072.255  012 120 162     5048          DB      NL,'Proceed (YES/NO) <NO> ','?'+200Q
072.305  315 334 067     5049          CALL    $ITL.
                         5050
072.310  176             5051          MOV     A,M
072.311  247             5052          ANA     A
072.312  312 200 054     5053          JZ      EXIT            WANTS TO EXIT
072.315  315 060 064     5054          CALL    CNO                                             /80.05.sc/
072.320  312 200 054     5055          JZ      EXIT            Wants to Exit                   /80.05.sc/
                         5056
072.323  315 076 064     5057          CALL    CYS                                             /80.05.sc/
072.326  302 247 072     5058          JNZ     PRS1            Not a legal answer              /80.05.sc/
                         5059
072.331  311             5060          RET                     Wants to proceed                /80.05.sc/
```

```
                     **       PRSS     - Preset Preset                                 /80.05.GC/
                     *
                     *        PRSS is the initial preset code executed only once
                     *        when INIT is entered.
                     *

072.332 377 011      PRSS     SCALL    .VERS
072.334 332 056 073           JC       PRSS1           NO .VERS SYSTEM CALL
072.337 376 040               CPI      VERS
072.341 302 056 073           JNZ      PRSS1           VERSIONS DO NOT MATCH

072.344 041 000 000           LXI      H,0
072.347 071                   DAD      SP              HL = Stack
072.350 315 145 067           CALL     $SOB            Skip blanks
072.353 371                   SPHL                     Save New Stack
072.354 176                   MOV      A,M
072.355 247                   ANA      A
072.356 312 367 072           JZ       PRSS0           Null Line
072.361 021 200 042           LXI      D,STACK
072.364 315 326 051           CALL     HLCPDE
072.367 314 203 071  PRSS0    CZ       PRS             Make sure we really want this!

072.372 076 000               MVI      A,OVL0
072.374 377 010               SCALL    .LOADO          LOAD  *HDOSOVL0.SYS*
072.376 076 001               MVI      A,OVL1
073.000 377 010               SCALL    .LOADO          LOAD  *HDOSOVL1.SYS*

073.002 315 064 073           CALL     FDP             fetch device parameters
073.005 315 123 075           CALL     PSD             Patch System Device

073.010 076 377               MVI      A,377Q
073.012 377 046               SCALL    .CLOSE  CLOSE THE CHANNEL WE CAME IN ON
073.014 257                   XRA      A
073.015 062 326 040           STA      S.CSLMD         SETUP CONSOLE MODE

073.020 315 027 076           CALL     $DOS            DISMOUNT OPERATING SYSTEM
073.023 332 203 054           JC       ERROR           ERROR

                     *        Check for data passed on the command line

073.026 041 000 000           LXI      H,0
073.031 071                   DAD      SP              HL = SP
073.032 176                   MOV      A,M
073.033 247                   ANA      A
073.034 312 060 054           JZ       INIT0           Null Line on Stack, ask user for device
073.037 021 200 042           LXI      D,STACK
073.042 315 326 051           CALL     HLCPDE
073.045 312 060 054           JZ       INIT0           Ask user for device

                     *        Force a default device specification from the command line

073.050 315 026 055           CALL     PDN.
073.053 303 071 054           JMP      INIT1           do it!

                     *        Error

```

```
073.056  076 050      5118  PRSS1   MVI      A,EC.NCV
073.060  067          5119          STC
073.061  303 203 054  5120          JMP      ERROR


                      5122  **      FDP      -  Fetch Device Parameters                /80.05.GC/
                      5123  *
                      5124  *       FDP fetches all of the data from each of the device
                      5125  *       parameter files
                      5126  *
                      5127
073.064  052 354 040  5128  FDP     LHLD     S.DFWA
                      5129
                      5130  *       Fetch parameter files
                      5131
073.067  176          5132  FDP1    MOV      A,M
000.000               5133          ERRNZ    DEV.NAM
073.070  376 000      5134          CPI      DV.EL
073.072  312 125 073  5135          JZ       FDP3             At the end of the device list
                      5136
073.075  376 001      5137          CPI      DV.NU
073.077  312 116 073  5138          JZ       FDP2             device entry not in use
                      5139
073.102  315 344 066  5140          CALL     $INDLB
073.105  006 000      5141          DW       DEV.FLG
073.107  346 001      5142          ANI      DT.DD
073.111  345          5143          PUSH     H
073.112  304 131 073  5144          CNZ      FDP5             fetch all of the parameters
073.115  341          5145          POP      H
                      5146
073.116  021 016 000  5147  FDP2    LXI      D,DEVELEN
073.121  031          5148          DAD      D
073.122  303 067 073  5149          JMP      FDP1
                      5150
                      5151  *       Flag the end of the device list for sure
                      5152
073.125  315 262 074  5153  FDP3    CALL     FDP9
073.130  311          5154          RET


                      5156  **      FDP5
                      5157  *
                      5158  *       really fetch the data
                      5159  *
                      5160
073.131  315 047 075  5161  FDP5    CALL     FMM
                      5162
073.134  315 234 030  5163          CALL     $INDL
073.137  000 000      5164          DW       DEV.NAM
073.141  353          5165          XCHG
073.142  042 353 074  5166          SHLD     FDPC             set device name in file look-up
073.145  042 043 075  5167          SHLD     FDPF             set .loadd name
```

```
073.150  353          5168       XCHG
                      5169
073.151  345          5170       PUSH    H
073.152  052 043 070  5171       LHLD    CRNDEV
073.155  315 365 066  5172       CALL    $INDS           set name in internal device table
073.160  000 000      5173       DW      DVT.NAM
073.162  321          5174       POP     D
073.163  315 365 066  5175       CALL    $INDS           set device table address
073.166  004 000      5176       DW      DVT.ADR
                      5177
                      5178  *    Check for table overflow
                      5179
073.170  052 043 070  5180       LHLD    CRNDEV
073.173  021 255 070  5181       LXI     D,DEVTABE
073.176  315 326 051  5182       CALL    HLCPDE
073.201  322 270 074  5183       JNC     FDP10           Table overflow
                      5184
                      5185  *    Read the Device driver parameter file
                      5186
073.204  021 362 074  5187       LXI     D,FDPD          DE = address of defaults
073.207  041 341 074  5188       LXI     H,FDPB          HL = file block
073.212  315 350 076  5189       CALL    $FOPER.
073.215  332 013 074  5190       JC      FDP6
                      5191
073.220  315 345 075  5192       CALL    VFT             Verify File Type, and find Parameters
073.223  332 062 074  5193       JC      FDP7
                      5194
073.226  001 006 000  5195       LXI     B,PIC.COD
073.231  021 370 074  5196       LXI     D,FDPE
073.234  041 341 074  5197       LXI     H,FDPB
073.237  315 076 077  5198       CALL    $FREAB.         Read the header information
073.242  332 062 074  5199       JC      FDP7
                      5200
073.245  072 370 074  5201       LDA     FDPE+PIC.ID
073.250  376 377      5202       CPI     377Q
073.252  302 062 074  5203       JNZ     FDP7            The file is not in BINARY format
                      5204
073.255  072 371 074  5205       LDA     FDPE+PIC.ID+1
073.260  376 001      5206       CPI     FT.PIC
073.262  302 062 074  5207       JNZ     FDP7            The file is not in PIC format
                      5208
073.265  052 372 074  5209       LHLD    FDPE+PIC.LEN    HL = length of entire record
073.270  353          5210       XCHG
073.271  052 041 070  5211       LHLD    BUFLIM
073.274  031          5212       DAD     D
073.275  353          5213       XCHG                    DE = Prospective buffer end
073.276  052 052 070  5214       LHLD    MEMLIM
073.301  315 326 051  5215       CALL    HLCPDE
073.304  076 021      5216       MVI     A,EC.NEM
073.306  332 203 054  5217       JC      ERROR
                      5218
073.311  052 041 070  5219       LHLD    BUFLIM
073.314  353          5220       XCHG                    DE = FWA for current entry
073.315  042 041 070  5221       SHLD    BUFLIM          Allocate the new memory
                      5222
073.320  052 043 070  5223       LHLD    CRNDEV
```

```
073.323  315 365 066  5224            CALL    $INDS           Save FWA of Parameter file
073.326  006 000      5225            DW      DVT.PAR
                      5226
                      5227  *         Read the rest of the file
                      5228
073.330  052 372 074  5229            LHLD    FDPE+PIC.LEN
073.333  104          5230            MOV     B,H
073.334  115          5231            MOV     C,L             BC = byte count
073.335  041 341 074  5232            LXI     H,FDPB
073.340  315 076 077  5233            CALL    $FREAB.
073.343  332 062 074  5234            JC      FDP7            Error
                      5235
073.346  041 341 074  5236            LXI     H,FDPB
073.351  315 225 076  5237            CALL    $FCLO.
073.354  332 062 074  5238            JC      FDP7
                      5239
073.357  315 241 075  5240            CALL    RDP             Relocate device parameters
                      5241
                      5242  *         Load device driver
                      5243
073.362  315 076 075  5244            CALL    FNM             reduce memory consumption
073.365  041 043 075  5245            LXI     H,FDPF
073.370  377 062      5246            SCALL   .LOADD          load device driver
073.372  332 143 074  5247            JC      FDP8            ERROR
                      5248
                      5249  *         Advance current device table
                      5250
073.375  052 043 070  5251            LHLD    CRNDEV
074.000  021 012 000  5252            LXI     D,DVT.ENL
074.003  031          5253            DAD     D
074.004  042 043 070  5254            SHLD    CRNDEV
                      5255
074.007  315 262 074  5256            CALL    FDP9
                      5257
074.012  311          5258            RET
                      5259
                      5260  *         Parameter file not found
                      5261
074.013  315 262 074  5262  FDP6      CALL    FDP9
074.016  315 136 031  5263            CALL    $TYPTX
074.021  116 157 040  5264            DB      'No INIT parameter file found,',' '+200Q
074.057  303 307 075  5265            JMP     TDI
                      5266
                      5267  *         Illegal format for parameter file, or error while reading data
                      5268
074.062  315 240 074  5269  FDP7      CALL    FDP9..
074.065  315 136 031  5270            CALL    $TYPTX
074.070  111 154 154  5271            DB      'Illegal format for INIT parameter file,',' '+200Q
074.140  303 307 075  5272            JMP     TDI
                      5273
                      5274  *         Error fetching the device driver
                      5275
074.143  315 262 074  5276  FDP8      CALL    FDP9
074.146  315 144 031  5277            CALL    $TYPTX.
074.151  116 157 164  5278            DB      'Not enough memory for device driver and parameters,',' '+200Q
074.235  303 307 075  5279            JMP     TDI
```

```

                                 *       Flag the end of the devices

074.240  041 341 074      FDP9..  LXI     H,FDPB
074.243  315 225 076              CALL    $FCLO.

074.246  052 043 070      FDP9.   LHLD    CRNDEV
074.251  315 234 030              CALL    $INDL
074.254  004 000                  DW      DVT.ADR
074.256  353                      XCHG
074.257  042 041 070              SHLD    BUFLIM              return the buffer space

074.262  052 043 070      FDP9    LHLD    CRNDEV
074.265  257                      XRA     A
074.266  167                      MOV     M,A
074.267  311                      RET


                                 *       Table overflow

074.270  315 262 074      FDP10   CALL    FDP9
074.273  315 136 031              CALL    $TYPTX
074.276  111 156 164              DB      'Internal device table overflow,',' '+200Q
074.336  303 307 075              JMP     TDI

                                 *FDPA    DW      0                   Device Table Address

074.341                   FDPB    DS      0
074.341  001                      DB      CN.FDP              channel
074.342  000                      DB      0                   flags
074.343  011 100                  DW      FDPBUF
074.345  011 100                  DW      FDPBUF
074.347  011 100                  DW      FDPBUF
074.351  011 102                  DW      FDPBUF+FDPBUFL
074.353  170 170 056      FDPC    DB      'xx.DVD',0          parameter file name

074.362  123 131 060      FDPD    DB      'SY0',0,0,0         parameter file defaults

074.370                   FDPE    DS      DVD.STE             temp holding for parameter file descriptors

075.043  170 170 072      FDPF    DB      'xx:',0             Device name for load



                                 **      FMM     - Fetch Maximum Memory
                                 *
                                 *       FMM requests the maximum allowable memory.
                                 *
                                 *       ENTRY:  NONE
                                 *
                                 *       EXIT:   NONE
                                 *
                                 *       USES:   NONE
                                 *

```

```
075.047  315 054 031  5333 FMM      CALL     $SAVALL
075.052  041 377 377  5334          LXI      H,-1
075.055  377 052      5335          SCALL    .SETTP
075.057  021 366 377  5336          LXI      D,-10
075.062  031          5337          DAD      D
075.063  042 052 070  5338          SHLD     MEMLIM
075.066  377 052      5339          SCALL    .SETTP
075.070  332 203 054  5340          JC       ERROR            Should never happen
075.073  303 047 031  5341          JMP      $RSTALL



                      5343 **       FNM      -  Fetch Minimal Memory
                      5344 *
                      5345 *        FNM reduces the allocated memory to the minimal amount
                      5346 *        as defined by BUFLIM.
                      5347 *
                      5348 *        ENTRY:   BUFLIM initializes
                      5349 *
                      5350 *        EXIT:    NONE
                      5351 *
                      5352 *        USES:    NONE
                      5353 *
                      5354
075.076  315 054 031  5355 FNM      CALL     $SAVALL
075.101  052 041 070  5356          LHLD     BUFLIM
075.104  021 012 000  5357          LXI      D,10
075.107  031          5358          DAD      D
075.110  042 052 070  5359          SHLD     MEMLIM
075.113  377 052      5360          SCALL    .SETTP
075.115  332 203 054  5361          JC       ERROR
075.120  303 047 031  5362          JMP      $RSTALL



                      5364 **       PSD      -  Patch System Driver
                      5365 *
                      5366 *        PSD patches the system disk subfunction caller to
                      5367 *        perform any mappings left over from boot.  See MSU.
                      5368 *
                      5369 *        ENTRY:   Internal Table Initialized
                      5370 *
                      5371 *        EXIT:    NONE
                      5372 *
                      5373 *        USES:    ALL
                      5374 *
                      5375
075.123  041 205 070  5376 PSD      LXI      H,DEVTAB
                      5377
075.126  016 004      5378 PSD1     MVI      C,PSDAL
075.130  021 235 075  5379          LXI      D,PSDA
075.133  345          5380          PUSH     H
075.134  315 060 030  5381          CALL     $COMP
075.137  341          5382          POP      H
```

```
075.140  312 161 075                JZ      PSD2            Have found SY:

075.143  001 012 000                LXI     B,DVT.ENL
075.146  011                        DAD     B
075.147  021 255 070                LXI     D,DEVTABE
075.152  315 326 051                CALL    HLCPDE
075.155  332 126 075                JC      PSD1            Not to the end of the table yet

075.160  311                        RET                     Entry Not found

075.161  042 061 070          PSD2  SHLD    SYPTR           Save Device Pointer for later
075.164  345                        PUSH    H
075.165  315 234 030                CALL    $INDL
075.170  006 000                    DW      DVT.PAR
075.172  353                        XCHG
075.173  001 000 002                LXI     B,SB.SDB-SB.BOO
075.176  011                        DAD     B               Use the Subfunction Entry-Point
075.177  042 142 065                SHLD    MSUA
075.202  341                        POP     H               HL = Internal Device Entry

075.203  315 234 030                CALL    $INDL
075.206  004 000                    DW      DVT.ADR
075.210  353                        XCHG
075.211  315 344 066                CALL    $INDLB
075.214  010 000                    DW      DEV.MNU
075.216  062 060 070                STA     SYMNU           Save Maximum number of units

075.221  052 346 040                LHLD    S.DLINK
075.224  315 344 066                CALL    $INDLB          Save System Boot Unit
075.227  021 000                    DW      M.SUNI
075.231  062 057 070                STA     SUNIT

075.234  311                        RET

075.235  123 131 072          PSDA  DB      'SY:',0
000.004                       PSDAL EQU     *-PSDA



                              **    RDP     -  Relocate Device Paramters
                              *
                              *     RDP relocates the Device Parameters.
                              *
                              *     ENTRY:  NONE
                              *
                              *     EXIT:   Device Parameters Relocated
                              *
                              *     USES:   NONE
                              *

075.241  052 043 070          RDP   LHLD    CRNDEV
075.244  315 234 030                CALL    $INDL
075.247  006 000                    DW      DVT.PAR         DE = Parameter Address

075.251  041 200 335                LXI     H,-SB.BOO
```

```
075.254  031          5436          DAD     D
075.255  104          5437          MOV     B,H
075.256  115          5438          MOV     C,L             BC = Displacement
                      5439
075.257  325          5440          PUSH    D
075.260  052 374 074  5441          LHLD    FDPE+PIC.PTR
075.263  031          5442          DAD     D               HL = Address of PIC Table
075.264  021 372 377  5443          LXI     D,-PIC.COD
075.267  031          5444          DAD     D               Account for Table space elsewhere
075.270  353          5445          XCHG
075.271  052 043 070  5446          LHLD    CRNDEV
075.274  315 365 066  5447          CALL    $INDS           Save PIC Table Address
075.277  010 000      5448          DW      DVT.PIC
075.301  353          5449          XCHG                    HL = Address of PIC Table
075.302  321          5450          POP     D
                      5451
075.303  315 175 033  5452          CALL    REL.            Relocate it!
075.306  311          5453          RET


                      5455 **       TDI     -  Type Device Ignored
                      5456 *
                      5457 *        TDI types the device ignored error message, and identifies
                      5458 *        the ignored device.
                      5459 *
                      5460
075.307  052 353 074  5461 TDI      LHLD    FDPC
075.312  042 327 075  5462          SHLD    TDIA
075.315  315 136 031  5463          CALL    $TYPTX
075.320  144 145 166  5464          DB      'device '
075.327  170 170 072  5465 TDIA     DB      'xx: ignored',ENL
075.343  247          5466          ANA     A
075.344  311          5467          RET


                      5469 **       VFT     - Verify File Type
                      5470 *
                      5471 *        VFT verifies that the Driver/Init File is of the required
                      5472 *        type.
                      5473 *
                      5474 *        ENTRY:  File open with file block FDPB
                      5475 *
                      5476 *        EXIT:   PSW     = 'C' CLEAR if File OK
                      5477 *                           File positioned at beginning of
                      5478 *                           parameters, file buffer cleared
                      5479 *                         'C' SET   if File NOT OK
                      5480 *        USES:   ALL
                      5481 *
                      5482
075.345  001 053 000  5483 VFT      LXI     B,DVD.STE
075.350  021 370 074  5484          LXI     D,VFTA
075.353  041 341 074  5485          LXI     H,FDPB
```

```
        075.356  315 076 077  5486              CALL     $FREAB.            Read the header information
        075.361  330          5487              RC
                              5488
        075.362  072 370 074  5489              LDA      VFTA+PIC.ID
        075.365  376 377      5490              CPI      377Q
        075.367  067          5491              STC
        075.370  300          5492              RNZ                         The File is NOT in BINARY format
                              5493
        075.371  072 371 074  5494              LDA      VFTA+PIC.ID+1
        075.374  376 001      5495              CPI      FT.PIC
        075.376  067          5496              STC
        075.377  300          5497              RNZ                         The file is NOT in PIC format
                              5498
        076.000  052 013 075  5499              LHLD     VFTA+DVD.INP
        076.003  174          5500              MOV      A,H
        076.004  265          5501              ORA      L
        076.005  067          5502              STC
        076.006  310          5503              RZ                          No INIT parameter file present
                              5504
        076.007  104          5505              MOV      B,H
        076.010  115          5506              MOV      C,L
        076.011  072 341 074  5507              LDA      FDPB+FB.CHA
        076.014  377 047      5508              SCALL    .POSIT             Position at the beginning of parameters
        076.016  330          5509              RC
        076.017  041 341 074  5510              LXI      H,FDPB
        076.022  315 176 076  5511              CALL     $FCLEAR            Clear up the buffers
                              5512
        076.025  247          5513              ANA      A                  Clear 'C'
        076.026  311          5514              RET
                              5515
        074.370               5516      VFTA    EQU      FDPE               Use FDPE buffer
```

```
    076.027                 5520           XTEXT    DOS


                            5522X **       $DOS - DISMOUNT OPERATING SYSTEM.
                            5523X *
                            5524X *        $DOS disounts all units of all directory devices        /80.04.sc/
                            5525X *
                            5526X *        THE USER IS MESSAGED ABOUT THE DISKS, AND THE OPERATING
                            5527X *        SYSTEM IS NOTIFIED.
                            5528X *
                            5529X *
                            5530X *        ENTRY    NONE
                            5531X *
                            5532X *        EXIT     (PSW)   = 'C' CLEAR IF NO ERROR
                            5533X *                           'C' SET   IF   ERROR
                            5534X *                            (A)  = ERROR CODE
                            5535X *
                            5536X *        USES     ALL
                            5537X *
                            5538X
    076.027  315 136 031    5539X $DOS     CALL     $TYPTX
    076.032  012 007 104    5540X          DB       NL,BELL,'Dismounting All Disks:',NL,ENL
                            5541X
    076.064  315 161 076    5542X          CALL     $DOS.
    076.067  330            5543X          RC
                            5544X
    076.070  315 136 031    5545X          CALL     $TYPTX
    076.073  012 122 145    5546X          DB       NL,'Remove the Disk(s). Hit RETURN when ready:',' '+200Q
                            5547X
    076.147  315 134 067    5548X DOS1     CALL     $RCHAR              READ CHARACTER
    076.152  376 012        5549X          CPI      NL
    076.154  302 147 076    5550X          JNE      DOS1
                            5551X
    076.157  247            5552X          ANA      A                   CLEAR CARRY
    076.160  311            5553X          RET


    076.161  076 000        5555X $DOS.    MVI      A,OVL0
    076.163  377 010        5556X          SCALL    .LOADO
    076.165  330            5557X          RC
                            5558X
    076.166  076 001        5559X          MVI      A,OVL1
    076.170  377 010        5560X          SCALL    .LOADO
    076.172  330            5561X          RC
                            5562X
    076.173  377 206        5563X          SCALL    .DAD                Dismount all Disks              /80.09.sc/
    076.175  311            5564X          RET
    076.176                 5565           XTEXT    FCLEAR
```

```
                              5567X **      $FCLEAR - CLEAR FILE BLOCK.
                              5568X *
                              5569X *       $FCLEAR CLEARS OUT A FILE BLOCK BY SETTING THE POINTERS TO
                              5570X *       EMPTY, AND CLEARING ANY ERROR OR EOF FLAGS.
                              5571X *
                              5572X *       THE DISK (OR WHATEVER) FILE IS NOT POSITIONED, READ, WRITEN
                              5573X *       OPENED OR CLOSED.
                              5574X *
                              5575X *       ENTRY   (HL) = FB ADDRESS
                              5576X *       EXIT    NONE
                              5577X *       USES    A,F,B,C
                              5578X
                              5579X
       076.176                5580X $FCLEAR EQU     *
       076.176  345           5581X         PUSH    H                SAVE FILE BLOCK ADDRESS
       000.000                5582X         ERRNZ   FB.FLG-FB.CHA-1
       076.177  043           5583X         INX     H
       000.000                5584X         ERRNZ   FB.FWA-FB.FLG-1
       076.200  043           5585X         INX     H                (HL) = #FB.FWA
       076.201  116           5586X         MOV     C,M
       076.202  043           5587X         INX     H
       076.203  106           5588X         MOV     B,M              (BC) = FB.FWA
       076.204  043           5589X         INX     H
       000.000                5590X         ERRNZ   FB.PTR-FB.FWA-2
       076.205  161           5591X         MOV     M,C              SET FB.PTR = FB.FWA
       076.206  043           5592X         INX     H
       076.207  160           5593X         MOV     M,B
       076.210  043           5594X         INX     H
       000.000                5595X         ERRNZ   FB.LIM-FB.PTR-2
       076.211  161           5596X         MOV     M,C              SET FB.LIM = FB.FWA
       076.212  043           5597X         INX     H
       076.213  160           5598X         MOV     M,B
       076.214  341           5599X         POP     H                (HL) = FB FWA
       076.215  311           5600X         RET
       076.216                5601          XTEXT   FCLO



                              5603X **      $FCLO - CLOSE FILE BLOCK.
                              5604X *
                              5605X *       $FCLO IS CALLED TO TERMINATE PROCESSING THROUGH A FILE
                              5606X *       BLOCK.
                              5607X *
                              5608X *       ENTRY   (HL) = FILE BLOCK ADDRESS
                              5609X *       EXIT    TO $FERROR IF ERROR
                              5610X *                TO CALLER IF OK
                              5611X *       USES    A,F,B,C,D,E
                              5612X
                              5613X
       076.216  315 225 076   5614X $FCLO   CALL    $FCLO.
       076.221  320           5615X         RNC                      NO ERROR
       076.222  303 203 054   5616X         JMP     $FERROR
                              5617X
       076.225  345           5618X $FCLO.  PUSH    H                SAVE FILE BLOCK ADDRESS
       000.000                5619X         ERRNZ   FB.FLG-1
```

```
     076.226  043              5620X           INX     H                (HL) = $FB.FLG
     076.227  176              5621X           MOV     A,M
     076.230  066 000          5622X           MVI     M,0              CLEAR FLAG
     076.232  247              5623X           ANA     A
     076.233  312 321 076      5624X           JZ      $FCLO4           FILE NOT OPEN
     076.236  346 004          5625X           ANI     FT.OW
     076.240  312 313 076      5626X           JZ      $FCLO3           NO WRITING, NO FLUSHING NEEDED
                               5627X
                               5628X *         WAS OPEN FOR WRITE. SEE IF NEED FLUSH THE LAST SECTOR
                               5629X
     076.243  315 234 030      5630X           CALL    $INDL
     076.246  003 000          5631X           DW      FB.PTR-FB.FLG
     076.250  325              5632X           PUSH    D                SAVE (FB.PTR)
     076.251  315 234 030      5633X           CALL    $INDL            (DE) = (FB.FWA)
     076.254  001 000          5634X           DW      FB.FWA-FB.FLG
     076.256  341              5635X           POP     H                (HL) = (FB.PTR)
     076.257  175              5636X           MOV     A,L
     076.260  223              5637X           SUB     E
     076.261  117              5638X           MOV     C,A
     076.262  174              5639X           MOV     A,H
     076.263  232              5640X           SBB     D
     076.264  107              5641X           MOV     B,A              (BC) = AMOUNT IN BLOCK
     076.265  261              5642X           ORA     C
     076.266  312 313 076      5643X           JZ      $FCLO3           NONE TO FLUSH
                               5644X
                               5645X *         NEED TO FLUSH BUFFER
                               5646X *
                               5647X *         (BC) = DATA AMOUNT
                               5648X *         (DE) = FWA
                               5649X *         (HL) = LWA+1
                               5650X
     076.271  171              5651X           MOV     A,C
     076.272  247              5652X           ANA     A
     076.273  312 306 076      5653X           JZ      $FCLO2           DONT HAVE PARTIAL SECTOR
                               5654X
                               5655X *         ZERO FILL PARTIAL SECTOR
                               5656X
     076.276  066 000          5657X $FCLO1    MVI     M,0
     076.300  043              5658X           INX     H
     076.301  014              5659X           INR     C
     076.302  302 276 076      5660X           JNZ     $FCLO1
     076.305  004              5661X           INR     B                COUNT ANOTHER FULL SECTOR
     076.306  341              5662X $FCLO2    POP     H                (HL) = FB FWA
     076.307  176              5663X           MOV     A,M              (A) = CHANNEL NUMBER
     000.000                   5664X           ERRNZ   FB.CHA
     076.310  345              5665X           PUSH    H
     076.311  377 005          5666X           DB      SYSCALL,.WRITE       FLUSH
                               5667X
                               5668X *         READY TO CLOSE FILE.
                               5669X *
                               5670X *         'C' SET IF ERROR
                               5671X *          (A) = ERROR CODE
                               5672X
     076.313  341              5673X $FCLO3    POP     H                (HL) = FILE BLOCK ADDRESS
     076.314  330              5674X           RC                       ERROR
     000.000                   5675X           ERRNZ   FB.CHA
```

```
076.315  176            5676X         MOV    A,M              (A) = CHANNEL NUMBER
076.316  345            5677X         PUSH   H
076.317  377 046        5678X         DB     SYSCALL,.CLOSE   CLOSE CHANNEL
076.321  341            5679X $FCLO4  POP    H                (HL) = FILE BLOCK ADDRESS
076.322  311            5680X         RET
054.203                 5681  $FERROR EQU    ERROR
076.323                 5682          XTEXT  FOPE



                        5684X **      $FOPEx - OPEN FILE BLOCK FOR I/O
                        5685X *
                        5686X *       $FOPEx IS CALLED BEFORE ANY I/O IS DONE VIA A
                        5687X *       FILE BLOCK. $FOPEx SETS UP THE FILE BLOCK, AND OPENS
                        5688X *       THE FILE VIA *HDOS*.
                        5689X *
                        5690X *       ENTRY  (DE) = ADDRESS OF DEFAULT BLOCK
                        5691X *              (HL) = ADDRESS OF FILE BLOCK
                        5692X *       EXIT   TO $FERROR IF ERROR
                        5693X *              TO CALLER IF OK
                        5694X *       USES   A,F,B,C,D,E
                        5695X
                        5696X
076.323  315 350 076    5697X $FOPER  CALL   $FOPER.
076.326  320            5698X         RNC
076.327  303 203 054    5699X         JMP    $FERROR          IN ERROR
                        5700X
076.332  315 353 076    5701X $FOPEW  CALL   $FOPEW.
076.335  320            5702X         RNC
076.336  303 203 054    5703X         JMP    $FERROR          IN ERROR
                        5704X
076.341  315 356 076    5705X $FOPEU  CALL   $FOPEU.
076.344  320            5706X         RNC
076.345  303 203 054    5707X         JMP    $FERROR          IN ERROR
                        5708X
                        5709X
076.350  076 002        5710X $FOPER. MVI    A,FT.OR          FILE TYPE OF OPEN FOR READ
076.352  001            5711X         DB     001Q             LXI,B TO SKIP NEXT MVI
076.353  076 004        5712X $FOPEW. MVI    A,FT.OW          OPEN FOR WRITE
076.355  001            5713X         DB     001Q             LXI,B TO SKIP NEXT MIV
076.356  076 006        5714X $FOPEU. MVI    A,FT.OR+FT.OW
                        5715X
                        5716X *       (A) = FILE FLAGS
                        5717X
076.360  345            5718X         PUSH   H                SAVE FILE BLOCK ADDRESS
076.361  365            5719X         PUSH   PSW              SAVE NEW FLAGS
000.000                 5720X         ERRNZ  FB.CHA
076.362  106            5721X         MOV    B,M              (B) = CHANNEL NUMBER
076.363  305            5722X         PUSH   B                SAVE HANNEL NUMBER
000.000                 5723X         ERRNZ  FB.FLG-FB.CHA-1
076.364  043            5724X         INX    H
076.365  117            5725X         MOV    C,A              (C) = NEW FILE FLAGS
076.366  176            5726X         MOV    A,M              (A) = CURRENT TYPE
076.367  247            5727X         ANA    A
076.370  171            5728X         MOV    A,C              (A) = NEW FLAGS TO BE SET
```

```
  076.371  312 003 077  5729X          JZ      $FOPE1          NOT ALREADY OPEN
                        5730X
                        5731X *        ALREADY OPEN. SQUACK
                        5732X
  076.374  301          5733X          POP     B               RESTORE (BC)
  076.375  361          5734X          POP     PSW             DISCARD NEW FLAGS
  076.376  341          5735X          POP     H               (HL) = FB ADDRESS
  076.377  076 031      5736X          MVI     A,EC.FAO        FILE ALREADY OPEN
  077.001  067          5737X          STC
  077.002  311          5738X          RET
                        5739X
  000.000               5740X          ERRNZ   FB.FWA-FB.FLG-1
  077.003  043          5741X $FOPE1   INX     H               (HL) = #FB.FWA
  077.004  116          5742X          MOV     C,M
  077.005  043          5743X          INX     H
  077.006  106          5744X          MOV     B,M             (BC) = FB.FWA
  077.007  043          5745X          INX     H
  000.000               5746X          ERRNZ   FB.PTR-FB.FWA-2
  077.010  161          5747X          MOV     M,C             SET FB.PTR = FB.FWA
  077.011  043          5748X          INX     H
  077.012  160          5749X          MOV     M,B
  077.013  043          5750X          INX     H
  000.000               5751X          ERRNZ   FB.LIM-FB.PTR-2
  077.014  161          5752X          MOV     M,C             SET FB.LIM = FB.FWA
  077.015  043          5753X          INX     H
  077.016  160          5754X          MOV     M,B
  077.017  043          5755X          INX     H
  000.000               5756X          ERRNZ   FB.NAM-FB.LIM-4
  077.020  043          5757X          INX     H
  077.021  043          5758X          INX     H               (HL) = #FB.NAM
                        5759X
                        5760X *        FILE BLOCK POINTERS SETUP. OPEN FILE
                        5761X
  077.022  345          5762X          PUSH    H               SAVE NEW ADDRESS FOR NAME
  077.023  041 054 077  5763X          LXI     H,$FOPEB
  077.026  247          5764X          ANA     A                                           /78.10.GC/
  077.027  312 036 077  5765X          JZ      $FOPE2
  000.000               5766X          ERRNZ   .EXIT
  077.032  315 163 067  5767X          CALL    $TBLS           FIND CODE
  077.035  176          5768X          MOV     A,M
  077.036  062 044 077  5769X $FOPE2   STA     $FOPEA          SET SYSCALL CODE
  077.041  341          5770X          POP     H               (HL) = #FB.NAM
  077.042  361          5771X          POP     PSW             (A) = CHANNEL NUMBER
  077.043  377 000      5772X          DB      SYSCALL,.EXIT
  077.044               5773X $FOPEA   EQU     *-1             SYSCALL CODE
  077.045  321          5774X          POP     D               (D) = NEW FLAG
  077.046  341          5775X          POP     H               (HL) = FILE BLOCK ADDRESS
  077.047  330          5776X          RC                      EXIT IF ERROR
  077.050  043          5777X          INX     H
  000.000               5778X          ERRNZ   FB.FLG-1
  077.051  162          5779X          MOV     M,D             SET NEW FLAGS
  077.052  053          5780X          DCX     H               RESTORE (HL)
  077.053  311          5781X          RET
                        5782X
  077.054  002 042      5783X $FOPEB   DB      FT.OR,.OPENR    TABLE OF SYSCALL CODES
  077.056  004 043      5784X          DB      FT.OW,.OPENW
```

```
077.060  006 044      5785X          DB     FT.OR+FT.OW,.OPENU
077.062  000          5786X          DB     0                SHOULD NOT OCCUR
077.063               5787           XTEXT  FREAB



                      5789X **       $FREAB - READ BYTES FROM FILE BUFFER.
                      5790X *
                      5791X *        $FREAB IS CALLED TO READ A NUMBER OF BYTES FROM A FILE BUFFER.
                      5792X *
                      5793X *        ENTRY  (BC) = BYTE COUNT
                      5794X *               (DE) = FWA FOR BYTES
                      5795X *               (HL) = ADDRESS OF FILE BUFFER
                      5796X *        EXIT   TO *FERROR* IF ERROR
                      5797X *               TO CALLER IF OK
                      5798X *                (BC) = UNREAD BYTE COUNT (ONLY IF EOF)
                      5799X *                (DE) = ADDRESS OF FIRST UNUSED BYTE
                      5800X *                'C' SET IF EOF DURING READ
                      5801X *        USES   A,F,B,C,D,E
                      5802X
                      5803X
077.063  315 076 077  5804X $FREAB   CALL   $FREAB.
077.066  320          5805X          RNC                     RETURN IF OK
077.067  376 001      5806X          CPI    EC.EOF
077.071  302 203 054  5807X          JNE    $FERROR          ERROR IS NOT EOF
077.074  067          5808X          STC
077.075  311          5809X          RET                     ERROR IS SIMPLY EOF
                      5810X
                      5811X
077.076               5812X $FREAB.  EQU    *
077.076  257          5813X          XRA    A
077.077  062 010 100  5814X          STA    EOFFLG           CLEAR EOF FLAG
077.102  345          5815X          PUSH   H
077.103  315 234 077  5816X          CALL   CBT              COPY BUFFER POINTERS TO TEMP CELLS
                      5817X
                      5818X *        COPY DATA FROM BUFFER TO TARGET
                      5819X
077.106  325          5820X $REAB2   PUSH   D                SAVE TARGET ADDRESS
077.107  072 377 077  5821X          LDA    T.FLG
077.112  346 002      5822X          ANI    FT.OR
077.114  076 011      5823X          MVI    A,EC.FNO         ASSUME FILE NOT OPEN FOR READ
077.116  067          5824X          STC
077.117  312 227 077  5825X          JZ     $REAB8           NOT OPEN FOR READ
077.122  170          5826X          MOV    A,B
077.123  261          5827X          ORA    C
077.124  312 227 077  5828X          JZ     $REAB8           ALL DONE
                      5829X
                      5830X *        COMPUTE MIN( DATA IN BUFFER, DATA REQUESTED)
                      5831X
077.127  052 002 100  5832X $REAB3   LHLD   T.PTR
077.132  353          5833X          XCHG                    (DE) = (FB.PTR) = ADDRESS OF DATA
077.133  052 004 100  5834X          LHLD   T.LIM            (HL) = LIMIT ADDRESS
077.136  175          5835X          MOV    A,L
077.137  223          5836X          SUB    E
077.140  157          5837X          MOV    L,A
```

```
077.141  174          5838X         MOV     A,H
077.142  232          5839X         SBB     D
077.143  147          5840X         MOV     H,A              (HL) = NUMBER OF BYTES IN BUFFER
077.144  171          5841X         MOV     A,C
077.145  225          5842X         SUB     L                COMPARE REQUESTED TO AVAILABLE
077.146  170          5843X         MOV     A,B
077.147  234          5844X         SBB     H
077.150  322 155 077  5845X         JNC     $REAB4           MORE REQUESTED THEN AVAILABLE
077.153  140          5846X         MOV     H,B
077.154  151          5847X         MOV     L,C              LIMIT TRANSFER TO REQUEST COUNT
077.155  174          5848X $REAB4  MOV     A,H
077.156  265          5849X         ORA     L
077.157  302 173 077  5850X         JNZ     $REAB6           SOME IN BUFFER
                      5851X
                      5852X *       BUFFER IS EMPTY. RE-FILL IT
                      5853X
077.162  315 314 077  5854X         CALL    $FFB             FILL FILE BUFFER
077.165  332 227 077  5855X         JC      $REAB8           ERROR CONDITION
077.170  303 127 077  5856X         JMP     $REAB3           COUNT NEW DATA
                      5857X
                      5858X *       GOT THE DATA. MOVE IT FROM BUFFER TO TARGET
                      5859X *
                      5860X *       (BC) = REQUESTED COUNT
                      5861X *       (DE) = FROM
                      5862X *       (HL) = COUNT
                      5863X *       ((SP)) = TO
                      5864X
077.173  171          5865X $REAB6  MOV     A,C
077.174  225          5866X         SUB     L
077.175  117          5867X         MOV     C,A
077.176  170          5868X         MOV     A,B
077.177  234          5869X         SBB     H
077.200  107          5870X         MOV     B,A              REMOVE BYTES ABOUT TO BE MOVED FROM REQUEST COUNT
077.201  305          5871X         PUSH    B
077.202  343          5872X         XTHL                     (HL) = REMAINING REQUEST COUNT
077.203  301          5873X         POP     B                (BC) = COUNT FOR THIS COPY
077.204  343          5874X         XTHL                     (HL) = TARGET ADDR, ((SP)) = REMAINING REQ. COUNT
077.205  032          5875X $REAB7  LDAX    D
077.206  167          5876X         MOV     M,A
077.207  023          5877X         INX     D
077.210  043          5878X         INX     H
077.211  013          5879X         DCX     B
077.212  170          5880X         MOV     A,B
077.213  261          5881X         ORA     C
077.214  302 205 077  5882X         JNZ     $REAB7           MORE TO GO
077.217  353          5883X         XCHG
077.220  042 002 100  5884X         SHLD    T.PTR            UPDATE POINTER
077.223  301          5885X         POP     B                (BC) = REMAINING COUNT
077.224  303 106 077  5886X         JMP     $REAB2           SEE IF MORE IN BUFFER
                      5887X
                      5888X *       READ COMPLETE.
                      5889X *
                      5890X *       (PSW) = COMPLETION FLAGS
                      5891X
077.227  321          5892X $REAB8  POP     D                RESTORE TARGET ADDRESS
077.230  341          5893X         POP     H
```

```
077.231  303 262 077  5894X          JMP     CTB             COPY TEMP POINTERS BACK TO BLOCK, EXIT
077.234               5895           XTEXT   FUTIL



                      5897X **       $FUTIL - UTILITY ROUTINES FOR FILE BLOCK ROUTINES.
                      5898X
                      5899X **       CBT - COPY BLOCK POINTERS TO TEMP CELLS.
                      5900X *
                      5901X *        ENTRY   (HL) = FILE BLOK FWA
                      5902X *        EXIT    NONE
                      5903X *        USES    A,F,H,L
                      5904X
077.234  325          5905X CBT      PUSH    D
077.235  305          5906X          PUSH    B               SAVE REGISTERS
000.000               5907X          ERRNZ   TLEN-10         ASSUME 10 BYTES TO MOVE
077.236  021 376 077  5908X          LXI     D,T.CHA         (DE) = TARGET FOR MOVE
077.241  006 005      5909X          MVI     B,10/2
077.243  176          5910X CBT1     MOV     A,M             COPY FILE BUFFER INTO WORK AREA
077.244  022          5911X          STAX    D
077.245  043          5912X          INX     H
077.246  023          5913X          INX     D
077.247  176          5914X          MOV     A,M
077.250  022          5915X          STAX    D
077.251  043          5916X          INX     H
077.252  023          5917X          INX     D
077.253  005          5918X          DCR     B
077.254  302 243 077  5919X          JNZ     CBT1            MORE TO GO
077.257  301          5920X          POP     B
077.260  321          5921X          POP     D               (DE) = DATA TARGET ADDRESS
077.261  311          5922X          RET
                      5923X
                      5924X
                      5925X **       CTB - COPY TEMP CELLS BACK TO FILE BLOCK.
                      5926X *
                      5927X *        ENTRY   (HL) = FILE BLOCK ADDRESS
                      5928X *        EXIT    NONE
                      5929X *        USES    NONE
                      5930X
077.262  365          5931X CTB      PUSH    PSW
077.263  325          5932X          PUSH    D
077.264  305          5933X          PUSH    B
077.265  345          5934X          PUSH    H               SAVE REGISTERS
077.266  006 004      5935X          MVI     B,8/2
077.270  021 376 077  5936X          LXI     D,T.CHA
077.273  032          5937X CTB1     LDAX    D
077.274  167          5938X          MOV     M,A
077.275  023          5939X          INX     D
077.276  043          5940X          INX     H
077.277  032          5941X          LDAX    D
077.300  167          5942X          MOV     M,A
077.301  023          5943X          INX     D
077.302  043          5944X          INX     H
077.303  005          5945X          DCR     B
077.304  302 273 077  5946X          JNZ     CTB1            RESTORE FILE BUFFER VALUES
```

```
077.307  341             5947X         POP      H
077.310  301             5948X         POP      B
077.311  321             5949X         POP      D
077.312  361             5950X         POP      PSW
077.313  311             5951X         RET


                         5953X **      $FFB - FILE FILE BUFFER.
                         5954X *
                         5955X *       $FFB FILLS THE FILE BUFFER BY READING FROM THE FILE.
                         5956X *
                         5957X *       ENTRY    NONE
                         5958X *       EXIT     'C' SET IF READ INCOMPLETE
                         5959X *                 (A) = ERROR CODE
                         5960X *                'C' CLEAR IF READ COMPLETEE
                         5961X *                 DATA IN BUFFER
                         5962X *       USES     A,F,D,E,H,L
                         5963X
                         5964X
077.314  072 010 100     5965X $FFB    LDA      EOFFLG
077.317  037             5966X         RAR
077.320  330             5967X         RC                       EOF
                         5968X
                         5969X *       CAN READ MORE. DO SO
                         5970X
077.321  305             5971X         PUSH     B               SAVE COUNT
077.322  052 000 100     5972X         LHLD     T.FWA
077.325  042 002 100     5973X         SHLD     T.PTR           CLEAR REMOVAL POINTER
077.330  353             5974X         XCHG
077.331  052 006 100     5975X         LHLD     T.LWA
077.334  042 004 100     5976X         SHLD     T.LIM           SET DATA LIMIT
077.337  175             5977X         MOV      A,L
077.340  223             5978X         SUB      E
077.341  117             5979X         MOV      C,A
077.342  174             5980X         MOV      A,H
077.343  232             5981X         SBB      D
077.344  107             5982X         MOV      B,A             (BC) = ROOM IN BUFFER
077.345  072 376 077     5983X         LDA      T.CHA
077.350  377 004         5984X         DB       SYSCALL,.READ   READ BUFFER
077.352  120             5985X         MOV      D,B             (D) = SECTORS UNREAD
077.353  301             5986X         POP      B               (BC) = DESIRED COUNT
077.354  320             5987X         RNC                      GOT THE DATA
                         5988X
                         5989X *       ERROR ON READ. SEE IF EOF
                         5990X
077.355  027             5991X         RAL
077.356  062 010 100     5992X         STA      EOFFLG          SET EOF, WE HOPE
077.361  376 003         5993X         CPI      EC.EOF*2+1
077.363  037             5994X         RAR
077.364  300             5995X         RNE                      IS NOT EOF, RETURN NOW!
077.365  072 005 100     5996X         LDA      T.LIM+1
077.370  222             5997X         SUB      D
077.371  062 005 100     5998X         STA      T.LIM+1         SET AMOUNT OF DATA WE DID GET
077.374  247             5999X         ANA      A
```

```
     077.375  311            6000X           RET                         EXIT WITH DATA
                             6001X
                             6002X
                             6003X **        TEMP CELLS TO HOLD FILE BLOCK POINTERS DURING I/O
                             6004X
     000.000                 6005X           ERRNZ   FB.CHA
     077.376  000            6006X T.CHA     DB      0                   CHANNEL NUMBER
     000.000                 6007X           ERRNZ   *-T.CHA-FB.FLG
     077.377  000            6008X T.FLG     DB      0                   FLAG BYTE
     000.000                 6009X           ERRNZ   *-T.CHA-FB.FWA
     100.000  000 000        6010X T.FWA     DW      0
     000.000                 6011X           ERRNZ   *-T.CHA-FB.PTR
     100.002  000 000        6012X T.PTR     DW      0
     000.000                 6013X           ERRNZ   *-T.CHA-FB.LIM
     100.004  000 000        6014X T.LIM     DW      0
     000.000                 6015X           ERRNZ   *-T.CHA-FB.LWA
     100.006  000 000        6016X T.LWA     DW      0
     000.012                 6017X TLEN      EQU     *-T.CHA             LENGTH OF TEMP CELLS
                             6018X
     100.010  000            6019X EOFFLG    DB      0
```

```
                         6022
  100.011                6023  MEML    EQU     *               MAX MEMORY FOR LOAD          /80.05.sc/
                         6024
                         6025  **      Overlaid PRSS Buffers
                         6026  *
                         6027
  100.011                6028  FDPBUF  DS      0
  100.011                6029          DS      512             FDP Buffer
  002.000                6030  FDPBUFL EQU     *-FDPBUF
                         6031
                         6032
  102.011                6033  RMEML   DS      0               RUNNING MEMORY MAX LEN       /80.05.sc/
```

```
     071.203                6036            ORG     OVRLAY            reuse the ephemeral code space
                            6037
                            6038    **      DIRECTORY SECTOR
                            6039
     071.203                6040    IDDA    DS      0                 DIRECTORY SECTOR FWA
                            6041
     071.203                6042    IDDB    DS      22*DIRELEN        SPACE FOR REST OF BLOCK
     001.372                6043    IDDBL   EQU     *-IDDB
     073.175                6044            DS      1                 SPACE FOR 00 BYTE
     000.000                6045            ERRNZ   *-IDDA-DIS.ENL
     073.176                6046            DS      1                 LENGTH OF ENTRYS (DIRELEN)
     000.000                6047            ERRNZ   *-IDDA-DIS.SEC
     073.177                6048            DS      2                 SECTOR NUMBER OF BEGINNING OF THIS BLOCK
     000.000                6049            ERRNZ   *-IDDA-DIS.LNK
     073.201                6050            DS      2                 SECTOR NUMBER OF BEGINNING OF NEXT BLOCK
     000.000                6051            ERRNZ   *-IDDA-512        SHOULD FILL BLOCK
                            6052
                            6053    **      RGTAB AND GRTAB MUST BE ON EVEN PAGE BOUNDARYS.
                            6054
     073.203                6055            DS      *+255/256*256-*
                            6056
     074.000                6057    GRTAB   DS      256               GRT TABLE
     075.000                6058    RGTAB   DS      256               RGT TABLE
                            6059
     004.011                6060            ERRMI   RMEML-*
                            6061
     076.000                6062            END
    ASSEMBLY COMPLETE
     6062 STATEMENTS
        1 ERRORS DETECTED
     7224 BYTES FREE
```

```
$$BITC  054001      1103L      2448
$$BITS  054004      1105L      2451
$$BUFF  054007      1107L      2454
$$CHL   054011      1109L      2457
$$CNO   054014      1111L      2460
$$CYS   054017      1113L      2463
$$DRVR  054022      1115L      2466
$$DRVR. 054025      1117L      2469
$$ITL.  054030      1119L      2472
$$MAX   054047      1129E      2486
$$MOVE  054033      1121L      2475
$$TBRA  054036      1123L      2478
$$TYPTX 054041      1125L      2481
$$VER   054000      1101L      2445
$$VSN   054044      1127L      2484
$BCRC   050316      1893L      1935    2360
$CCO    066250      2536       2558    2594    2696    2776    2803    2817    2951    2957    4164L   5031    5046
$CDEHL  030216      4180E      4217
$CHL    030224      2458       3870    4192E
$CMP$   000001      4663E      4707    4716
$COMP   030060      1575E      1620    2620    3545    5381
$CRLF   070033      4710       4916L
$DADA   030072      4017       4686    4904E
$DADA.  030101      1585E      2379    4863
$DOS    076027      5097       5539L
$DOS.   076161      5542       5555L
$DTB    066265      4204L
$DTB1   066271      4208L      4211
$DTB2   066300      4216L      4221
$DTB3   066315      4218       4225L
$DU66   030106      3100       3111    3699    4245E
$FCLEAR 076176      5511       5580E
$FCLO   076216      5614L
$FCLO.  076225      5237       5284    5614    5618L
$FCLO1  076276      5657L      5660
$FCLO2  076306      5653       5662L
$FCLO3  076313      5626       5643    5673L
$FCLO4  076321      5624       5679L
$FERROR 054203      5616       5681E   5699    5703    5707    5807
$FFB    077314      5854       5965L
$FOPE1  077003      5729       5741L
$FOPE2  077036      5765       5769L
$FOPEA  077044      5769       5773E
$FOPEB  077054      5763       5783L
$FOPER  076323      5697L
$FOPER. 076350      5189       5697    5710L
$FOPEU  076341      5705L
$FOPEU. 076356      5705       5714L
$FOPEW  076332      5701L
$FOPEW. 076353      5701       5712L
$FREAB  077063      5804L
$FREAB. 077076      5198       5233    5486    5804    5812E
$HLIHL  030211      4256E
$ICTT   046313      1364L      1365    1875
$ICTT.  046325      1364       1369L   2298
$ICTT.. 046353      1366       1391L   2300
$IDN    066326      2806       2960    4274L
$INDL   030234      2637       2685    3664    3858    3867    3877    3952    4313E   5163    5287    5395    5403
```

```
 5432     5630      5633
   $INDLB  066344     2640       4328L     5140      5406      5411
   $INDS   066365     4361L      5172      5175      5224      5447
   $INDSB  067021     3800       3836      4391L
   $ITL    067342     2599       2707      2820      4841      4844L
   $ITL.   067334     2473       2563      2779      4841L     5049
   $MCU    052071     1187       2226L     3629      3643      4448
   $MLU    067047     4443L      4475      4842
   $MLU1   067052     4446L      4451
   $MOVE   030252     1270       1554E     1960      2476      3201      3333      3720      3861      3983      3986
   $MU10   030324     4801       4824E
   $MU86   031007     1305       1444E     3091      3498
   $PDD    067274     4276       4787L
   $PDD1   067300     4790L      4809
   $RCHAR  067134     4479       4539L     4540      5548
   $REAB2  077106     5820L      5886
   $REAB3  077127     5832L      5856
   $REAB4  077155     5845       5848L
   $REAB6  077173     5850       5865L
   $REAB7  077205     5875L      5882
   $REAB8  077227     5825       5828      5855      5892L
   $RSTALL 031047     4080E      4168      5341      5362
   $RTL    067076     4275       4473      4477E     4845
   $RTL.   067067     4473L
   $RTL1   067077     4479L      4485
   $RTL2   067131     4481       4502L
   $SAVALL 031054     4094E      4164      5333      5355
   $SOB    067145     3590       4557L     5075
   $SOB1   067146     4558L      4561      4563
   $TBLS   067163     4585L      5767
   $TBRA   031076     2479       4624E
   $TDD    067352     2765       4859L
   $TDD.   067350     4858L
   $TYPC.  067220     4644L      4711      4874
   $TYPCH  067214     4634L
   $TYPEC. 047064     1420       1425      1471      1495L     1525      2390
   $TYPET  047027     1163       1199      1215      1229      1236      1285      1288      1325      1332      1339      1464L     1481
                      1661       2267      2272      2278      2313      2330      2339      2418
   $TYPET. 047035     1465       1469L     1477
   $TYPET1 047055     1473       1481L
   $TYPL.  067241     2770       4688      4700E     4714
   $TYPLN. 067223     4682L
   $TYPTX  031136     2482       2519      2537      2559      2595      2652      2697      2716      2742      2753      2759      2766
                      2771       2777      2804      2818      2835      2862      2952      2958      2968      3144      3681      4042      4660E
                      5032       5047      5263      5270      5301      5463      5539      5545
   $TYPTX. 031144     4662E      5277
   $UDD    031157     4746E
   $WCHAR  067142     4543L
   $WDR    031222     4529E
   $WER    031241     4517E
   .       070205     2116S      2117      3724S     3726      3729      3732      3735      3741      3855S     3856      4947S     4949
                      4952       4955      4958      4961
   .ABUSS  040024      152E
   .ALARM  002136      125E
   .ALEDS  040013      150E      1959      1961      2032
   .CHFLG  000060      774L
   .CLEAN  000205      789L
   .CLEAR  000055      771L
 .CLEARA 000056      772L
 .CLOSE  000046      764L      5093     5678
 .CLRCO  000007      748L
 .CONSL  000006      747L      4167
 .CRC    002347      133E
 .CRCSUM 040027      153E
 .CTC    002172      127E
 .CTL2FL 040066      159E
 .CTLC   000041      759L
 .CTLFLG 040011      149E
 .DAD    000206      790L      5563
 .DECODE 000053      769L
 .DELET  000050      766L
 .DISMT  000061      775L
 .DLEDS  040021      151E
 .DLY    000053      122E      1986     2024     2208
 .DMNMS  000203      787L
 .DMOUN  000201      785L
 .DOD    003122      136E
 .DODA   003356      138E
 .DSPMOD 040007      147E
 .DSPROT 040006      146E
 .DUMP   001374      124E
 .ERROR  000057      773L      2541
 .EXIT   000000      741L      2527     5766     5772
 .HORN   002140      126E      1963
 .IDENT  000000      121E
 .IOWRK  040002      144E
 .LINK   000040      758L
 .LOAD   001267      123E
 .LOADD  000062      776L      5246
 .LOADO  000010      749L      5085     5087     5556     5560
 .MFLAG  040010      148E      1953     1956     2046
 .MONMS  000202      786L
 .MOUNT  000200      784L
 .NAME   000054      770L
 .NMIRET 040064      158E
 .OPEN   000063      777L
 .OPENC  000045      763L
 .OPENR  000042      760L      5783
 .OPENU  000044      762L      5785
 .OPENW  000043      761L      5784
 .PCHL   002264      129E
 .POSIT  000047      765L      5508
 .PRINT  000003      744L
 .RCK    003260      137E
 .READ   000004      745L      5984
 .REGI   040005      145E
 .REGPTR 040035      156E
 .RENAM  000051      767L
 .RESET  000204      788L
 .RNB    002331      132E
 .RNP    002325      131E
 .SCIN   000001      742L      4539
 .SCOUT  000002      743L      4543     4644     4917
 .SETTF  000052      768L      5335     5339     5360
 .SRS    002265      130E
 .START  040000      143E
```

```
.SYSRES 000012     751L
.TICCNT 040033     155E     1783
.TPERR  002205     128E
.TPERRX 040031     154E
.UIVEC  040037     157E     1154
.VERS   000011     750L     5068
.WNB    003024     135E
.WNP    003017     134E
.WRITE  000005     746L     5666
AAL     056251    2508      2736L
AAL1    057030    2749      2759L
AAL2    057114    2755      2776L    2786
AAL3    057253    2780      2788L
ABF     053174    1156      1181     1213     2426L
ABR     051000    1828      1949E
ABR.A   051231    1958      2049L
ABR0    051112    1993L     1998
ABR0.1  051000    1953L
ABR0.3  051040    1967L     2019
ABR0.5  051075    1982      1984L
ABR1    051114    1994L     2002
ABR2    051157    2018L     2028
ABR3    051163    2010      2023L
ABR5    051200    2014      2032L
ABR5.1  051207    2035L     2038
ABS.COD 000010     384L     1132
ABS.ENT 000006     382L
ABS.ID  000000     378L
ABS.LDA 000002     380L
ABS.LEN 000004     381L
AC.DLY  000156     285E     1985     2207
ADB     061332    3023      3084E
ADB1    062005    3117L     3138     3142
ADB2    062014    3121L     3126
ADB3    062031    3124      3131L
ADB4    062137    3127      3151L
ADBA    062211    3119      3163     3180L
AIO.CGN 041047    1026L
AIO.CHA 041116    1041L
AIO.CNT 041111    1037L
AIO.CSI 041050    1027L
AIO.DDA 041041    1022E
AIO.DES 041055    1031L     1607     1633
AIO.DEV 041057    1032L
AIO.DIR 041062    1035L     1162     1269     1416     1617
AIO.DTA 041053    1030L
AIO.EOF 041113    1039L
AIO.EOM 041112    1038L
AIO.FLG 041043    1023L
AIO.GRT 041044    1024L
AIO.LGN 041051    1028L
AIO.LSI 041052    1029L
AIO.SPG 041046    1025L
AIO.TFP 041114    1040L
AIO.UNI 041061    1033L     2690     3802     3813     3821     3833
AIO.VEC 041040    1021L
AMW     054243    2497      2556E
AMW1    054243    2558L     2564     2574
```

```
AMW2    054367      2568        2576L
ANS     063353      3291        3427L
ANS1    063370      3430        3438L
BELL    000007      1060E       1286    1289    1326    1326    1333    1334    1340    1340    1662    1663    2268
                    2268        2538    2717    2863    3145    3145    3682    3682    5540
BFLG.A  000001       173E
BGT     062213      3051        3198L
BGT1    062271      3225L       3231
BGT1.5  062302      3223        3233L
BGT2    062323      3246L       3252
BGT3    062333      3248        3251L
BITC    066212      2449        4109L
BITC1   066217      4114L       4116
BITS    066231      2452        4138L
BITS1   066236      4143L       4145
BKSF    000010      1062E
BLABEL  053205      1271        1279    1304    1686    1692    1697    1702    1704    2251    2258    2304    2322
                    2433E       2434
BOOEND  066205      2435E
BOOT.F  000001      1001E       1218    1239
BOOTA   037132       695E
BOOTAL  000130       696E
BTS     064044      3156        3174    3444    3494L   4009    4030
BUFF    054205      1606        1613    1632    2257    2356    2434E   2435
BUFLIM  070041      4924L       5211    5219    5221    5290    5356
C.STX   000002      1064E
C.SYN   000026      1063E
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
CBS     050066      1744L       1867
CBS.    050133      1766L
CBS1    050075      1749L       1753
CBS2    050115      1747        1757L   1761
CBS3    050132      1755        1764L
CBS4    050163      1749        1757    1783L
CBS5    050174      1785        1789L
CBS6    050204      1791        1795L
CBT     077234      5816        5905L
CBT1    077243      5910L       5919
CDB.H84 000001       944E       1370    1392    1497    1825    1835    2157
CDB.H85 000000       943E       1746    1818
CDC     052102      1231        2246L
CDC1    052127      2257L       2326
CDC2    052174      2263        2272L   2306
CDC3    052324      2299        2303L
CDC4    053054      2302        2339L
CDCA    053175      2247        2284    2287    2332    2427L
CDCB    053177      2248        2280    2283    2315    2318    2428L
CDCD    053201      2253        2274    2294    2296    2324    2429L
CDCE    053203      2250        2281    2285    2289    2297    2430L
CFF     031354       673E
CLOCK   034031       698E       1153
```

```
CN.170M 000014       108E
CN.174M 000003       107E
CN.ABO  000200       112E
CN.BAU  000100       111E
CN.FDP  000001        37E      5308
CN.MEM  000040       110E
CN.PRI  000020       109E
CND.H17 000000       114E
CND.H47 000001       116E
CND.NDI 000000       115E
CNO     064060      2461       2573    2785    3515L   5054
CNOA    064073      3519       3523L   3524
CNOAL   000003      3520       3524E
CO.FLG  000001       921E      4166
CR      000015      1056E      1188    1482    1517    1529    5005    5007
CRNDEV  070043      2633       2684    3663    3857    3865    3946    3951    4925L   5171    5180    5223    5251
                    5254       5286    5292    5431    5446
CS.FLG  000200       922E
CSC     053063      2282       2286    2290    2355L
CSL.CHR 000001       898E
CSL.ECH 000200       895E
CSL.RAW 000004       896E
CSL.WRP 000002       897E
CTB     077262      5894       5931L
CTB1    077273      5937L      5946
CTLA    000001      1071E
CTLB    000002      1072E
CTLC    000003      1073E      2301
CTLD    000004      1074E      4480
CTLO    000017      1075E
CTLP    000020      1076E
CTLQ    000021      1077E
CTLS    000023      1078E
CTLZ    000032      1079E
CTP.2SB 000010       907E      1833    2170    2171    2201    2203
CTP.BKM 000002       908E
CTP.BKS 000200       903E
CTP.FF  000100       904E
CTP.MLI 000040       905E
CTP.MLO 000020       906E
CTP.TAB 000001       909E
CYS     064076      2464       2570    2782    3539L   5057
CYS1    064106      3521       3545L
CYSA    064115      3543       3552L   3553
CYSAL   000004      3544       3553E
D.ABORT 040141       849L
D.CDE   040160       854L
D.CON   040110       713L
D.DLY   040235       869L
D.DLYHS 040244       808L
D.DLYMO 040243       807L
D.DRVTB 040251       813L
D.DTS   040163       855L
D.DVCTL 040242       805L
D.E.CHK 040267       824L
D.E.HCK 040270       825L
D.E.HSY 040266       823L
D.E.MDS 040265       822L
```

```
D.E.TRK  040272      827L
D.E.VOL  040271      826L
D.ERR    040265      821L
D.ERRL   040273      828L
D.ERRT   040232      868L
D.HECNT  040261      815L
D.LPS    040177      859L
D.MAI    040171      857L
D.MAO    040174      858L
D.MOUNT  040133      847L
D.OECNT  040264      817L
D.OPR    040273      832L
D.OPW    040275      833L
D.RAM    040240      716L       800     835
D.RAML   000037      835E
D.RDB    040202      860L
D.READ   040147      851L
D.READR  040152      852L
D.SDP    040205      861L
D.SDT    040166      856L
D.SECNT  040262      816L
D.STS    040210      862L
D.STZ    040213      863L
D.SYDD   040130      846L
D.TRKPT  040245      810L
D.TS     040241      803L
D.TT     040240      802L
D.UDLY   040216      864L
D.VEC    040130      715L       844
D.VOLPT  040247      811L
D.WNB    040227      867L
D.WRITE  040155      853L
D.WSC    040221      865L
D.WSP    040224      866L
D.XIT    040144      850L
D.XOK    040136      848L
DBI      070045     3169       3212    4037    4926L
DC.ABT   000007      356L      1680    2737    2868    2875
DC.CLO   000006      355L
DC.LOD   000011      358L
DC.MAX   000013      360L
DC.MOU   000010      357L      1710    3763    3770
DC.OPR   000003      352L
DC.OPU   000005      354L
DC.OPW   000004      353L
DC.RDY   000012      359L
DC.REA   000000      349L      1655    2261
DC.RER   000002      351L      1666    2748
DC.WRI   000001      350L      2889    3034    3047    3058    3338    3353
DCA      032002      675E
DDF.BOL  000011      492E      2424
DDF.BOU  000000      491L
DDF.LAB  000011      493L      1687    2746    3045
DDF.USR  000012      494L      4014
DDS      064121     2605       3572E
DDS1     064174     3599       3611L
DDS2     064212     3591       3593    3627L
DDS3     064231     3577       3578    3641L
```

```
DEV.DDA  000004      452L
DEV.DVG  000015      465L
DEV.DVL  000013      464L
DEV.FLG  000006      453L      5141
DEV.JMP  000003      451L      3667
DEV.MNU  000010      461L      2641      5407
DEV.MUM  000007      460L
DEV.NAM  000000      443L      5133      5164
DEV.RES  000002      447L
DEV.UNT  000011      462L
DEVELEN  000016      467E      5147
DEVTAB   070205     2609       4925      4946L     5376
DEVTABE  070255     2612       4965L     5181      5387
DF.CLR   000376      405E      3343      3402
DF.EMP   000377      404E      3276
DIF.CNT  000020      627E      1297      3374      3389
DIF.LOC  000100      625E      3374      3389      3398
DIF.SYS  000200      624E      3374      3389      3398
DIF.WP   000040      626E      3374      3389      3398
DIR.ALD  000025      420L
DIR.CLU  000015      413L      3371
DIR.CRD  000023      419L      3382
DIR.EXT  000010      408L      3367
DIR.FGN  000020      416L      1299      3376
DIR.FLG  000016      414L      1294      1299      3373
DIR.LGN  000021      417L      3378
DIR.LSI  000022      418L      3380
DIR.NAM  000000      407L      1416      3365
DIR.PRO  000013      409L      3369
DIR.VER  000014      410L
DIRBLK   070050     3168       3214      4928L
DIRELEN  000027      422E       430       559      1035      1624      3090      3286      3404      3407      3469      6042
DIRIDL   000015      411E      1268      1278
DIRLBLK  070051     3235       3325      4929L
DIS.ENL  001373      434L      3090      3287      6045
DIS.ENT  000000      429E      1613      3090
DIS.LNK  001376      436L      1632      3292      3355      6049
DIS.SEC  001374      435L      3290      3336      3351      6047
DM.MR    000000       80E
DM.MW    000001       81E
DM.RR    000002       82E
DM.RW    000003       83E
DOS1     076147     5548L      5550
DR.IM    000001      448E
DR.PR    000002      449E
DRIVER   064241     2467       2736      2747      2867      2874      3655L     3769
DRIVER.  064271     2470       2888      3033      3046      3057      3337      3352      3674L     3762
DRIVER1  064245     3660L      3678
DT.CH    000020      458E
DT.CR    000002      455E
DT.CW    000004      456E
DT.DD    000001      454E      5142
DT.RN    000010      457E
DU68     064354     2985       3697L     3924      4001
DV.EL    000000      444E      5134
DV.NU    000001      445E      5137
DVD.CAP  000007      504L
DVD.DVD  000006      503L
```

```
  DVD.ENT 002000      513E
  DVD.INP 000023      509L      5499
  DVD.MNU 000011      506L
  DVD.MUM 000010      505L
  DVD.SET 000022      508L
  DVD.STE 000053      511E      5318    5483
  DVD.UFL 000012      507L
  DVDFLV  000307      499E
  DVT.ADR 000004     2638       3665    4952E   5176    5288    5404
  DVT.ENL 000012     2627       4961E   4963    5252    5385
  DVT.MAX 000004       39E      4963
  DVT.NAM 000000     2686       4949E   5173
  DVT.PAR 000006     3859       3868    3953    4955E   5225    5396    5433
  DVT.PIC 000010     3878       4958E   5448
  EC.CNA  000004      192L
  EC.DDA  000027      211L
  EC.DIF  000017      203L
  EC.DIW  000035      217L
  EC.DNI  000045      225L
  EC.DNR  000046      226L
  EC.DNS  000005      193L
  EC.DSC  000047      227L
  EC.EOF  000001      189L      5806    5993
  EC.EOM  000002      190L
  EC.FAO  000031      213L      5736
  EC.FAP  000026      210L
  EC.FL   000030      212L
  EC.FNF  000014      200L      1637
  EC.FNO  000011      197L      5823
  EC.FNR  000034      216L
  EC.FOD  000043      223L
  EC.FUC  000013      199L
  EC.ICN  000016      202L
  EC.IDN  000006      194L
  EC.IFC  000020      204L
  EC.IFN  000007      195L
  EC.ILC  000003      191L
  EC.ILO  000040      220L
  EC.ILR  000012      198L
  EC.ILV  000037      219L
  EC.IOI  000052      230L
  EC.IS   000032      214L
  EC.NCV  000050      228L      5118
  EC.NEM  000021      205L      5216
  EC.NOS  000051      229L
  EC.NPM  000044      224L
  EC.NRD  000010      196L
  EC.NVM  000042      222L
  EC.OTL  000053      231L
  EC.RF   000022      206L
  EC.UNA  000036      218L
  EC.UND  000015      201L
  EC.UUN  000033      215L
  EC.VPM  000041      221L
  EC.WF   000023      207L
  EC.WP   000025      209L
  EC.WPV  000024      208L
  ENL     000212     1069E      2521    2655    2718    2743    2754    2819    2836    2863    2955    2969    3145
```

```
3682    4044     5044     5465      5540
  EOFFLG   100010       5814      5965      5992      6019L
  ERR.FNO  031344        669E
  ERR.ILR  031350        671E
  ERROR    054203       2531L     5098      5120      5217      5340      5361      5681
  ERROR1   054213       2531      2535L
  ESC      000033       1067E
  EXIT     054200       2498      2526L     5053      5055
  EXIT.    054201       2527L     2533
  FB.CHA   000000        519L     5507      5582      5664      5675      5720      5723      6005
  FB.FLG   000001        520L     5582      5584      5619      5631      5634      5723      5740      5778      6007
  FB.FWA   000002        521L     5584      5590      5634      5740      5746      6009
  FB.LIM   000006        523L     5595      5751      5756      6013
  FB.LWA   000010        524L     6015
  FB.NAM   000012        525L      526      5756
  FB.NAML  000021        526E
  FB.PTR   000004        522L     5590      5595      5631      5746      5751      6011
  FBENL    000033        527E
  FCU      050210       1158      1774      1808L
  FCU0     050270       1831      1835L
  FCU1     050272       1820      1826      1841L
  FDP      073064       5089      5128L
  FDP1     073067       5132L     5149
  FDP10    074270       5183      5300L
  FDP2     073116       5138      5147L
  FDP3     073125       5135      5153L
  FDP5     073131       5144      5161L
  FDP6     074013       5190      5262L
  FDP7     074062       5193      5199      5203      5207      5234      5238      5269L
  FDP8     074143       5247      5276L
  FDP9     074262       5153      5256      5262      5276      5292L     5300
  FDP9.    074246       5286L
  FDP9..   074240       5269      5283L
  FDPB     074341       5188      5197      5232      5236      5283      5307L     5485      5507      5510
  FDPBUF   100011       5310      5311      5312      5313      6028L     6030
  FDPBUFL  002000       5313      6030E
  FDPC     074353       5166      5314L     5461
  FDPD     074362       5187      5316L
  FDPE     074370       5196      5201      5205      5209      5229      5318L     5441      5516
  FDPF     075043       5167      5245      5320L
  FF       000014       1070E
  FFB      032133        677E
  FFL      032205        679E
  FMM      075047       5161      5333L
  FNM      075076       5244      5355L
  FOV      061200       2516      3005E
  FOV1     061216       3017L     3021
  FOV2     061230       3017      3023L
  FT.ABS   000000        368E     1135
  FT.BAC   000003        371E
  FT.DD    000001        538E
  FT.OC    000020        542E
  FT.OR    000002        539E     5710      5714      5783      5785      5822
  FT.OU    000010        541E
  FT.OW    000004        540E     5625      5712      5714      5784      5785
  FT.PIC   000001        369E     5206      5495
  FT.REL   000002        370E
  GBL      060244       2514      2902E
```

```
GBL1      060252     2906L      2909
GBL1.5    060261     2911L      2916
GBL2      060274     2913       2918E
GBL4      060274     2951L
GBL5      061050     2957L      2961      2970      2990
GBL6      061107     2968L      2978
GBL7      061145     2966       2972L
GRTAB     074000     2455       3055      3200      3204      3215      3245      3441      6057L
GRTBLK    070047     3153       3236      3315      4927L
GVI       057256     2511       2802E     2807      2810      2812
GVI1      057372     2817L      2837
GVI2      060067     2827L      2834
GVI3      060123     2831       2839L     2842
HLCPDE    051326     2134L      2325      2613      2965      3892      3947      5081      5108      5182      5215      5388
I.CONFL   000004      924E       925      4165
I.CONTY   000001      911E       912
I.CONWI   000003      917E       918
I.CSLMD   000000      900E
I.CUSOR   000002      914E       915
ICTT1     046335     1375L
ICTT2     046344     1371       1384L
ICTT3     046370     1393       1403L
IDD       062341     3062       3274L
IDD2      062377     3290L      3358
IDD4      063166     3295       3349L
IDD5      063203     3345       3355L
IDDA      071203     3287       3290      3292      3335      3336      3350      3351      3355      6040L     6045      6047      6049
                     6051
IDDB      071203     3332       3461      6042L     6043
IDDBL     001372     3332       3462      6043E
IDDC      063214     3331       3364L     3365      3367      3369      3371      3373      3376      3378      3380      3382      3406
IDDC0     063234     3302       3377L
IDDC1     063235     3303       3379L
IDDC2     063237     3306       3307      3383L
IDDC3     063263     3316       3317      3319      3391L
IDDC4     063266     3308       3309      3392L
IDDC5     063312     3324       3326      3328      3400L
IDDC6     063315     3310       3311      3401L
IDDCL     000134     3330       3332      3406E     3407
IDDD      063350     3275       3293      3409L
IDDE      063351     3281       3410L     3438
IDDF      063352     3282       3411L     3427      3429      3439
IDS       060133     2512       2859L
IDS.      060201     2859       2867L
ILDEHL    066337     4292L      4364
INI.CMV   000000     1087L      2711
INI.DBI   000002     1089L      4036
INI.IDS   000001     1088L      2871
INI.MAX   000004     1092E
INI.PAR   000003     1090L      3978
INIT0     054060     2500L      2506      5106      5109
INIT1     054071     2504L      2509      2513      2517      2522      5114
INITVEC   054000     1097E      1099      2441      2443
IOC.CGN   000010      547L
IOC.CSI   000011      548L
IOC.DDA   000002      535L       543       557
IOC.DES   000016      554L
IOC.DEV   000020      555L      2660      3647
IOC.DIL  000021     557E
IOC.DIR  000023     559L      2661       3648
IOC.DRL  000010     551E
IOC.DTA  000014     553L
IOC.FLG  000004     537L       551
IOC.GRT  000005     545L
IOC.LGN  000012     549L
IOC.LNK  000000     534L
IOC.LSI  000013     550L
IOC.SPG  000007     546L
IOC.SQL  000003     543E
IOC.UNI  000022     556L      2660       2661       3647       3648
IOCCTD   000001     563E
IOCELEN  000052     561E
IP.CON   000362      56E
IP.PAD   000360      52E
ISDEHL   067042    4371       4427L
ITLA     070064    2602       4274       4844       4941E
IVP      064362    2647       3715L
IVPA     065005    3718       3723L      3742
IVPAL    000035    3717       3742E
IVPB     065011    3716       3736L
LAB.AUX  000117     598E       600       3984       5001
LAB.AXL  000001     600E      3985       5002
LAB.DAT  000000     575E      1273       4981
LAB.DIS  000003     571L      1279       3175       3288       3321       4974
LAB.GRT  000005     572L      3056       3157       4976
LAB.IND  000001     570L      3039       4972
LAB.LAB  000021     594L       595       2769       2822       4997
LAB.LBL  000074     595E      2768
LAB.NOD  000002     577E
LAB.PSS  000016     586L      4991
LAB.RGT  000012     582L      3032       3202       3299       4031       4985
LAB.SER  000000     569L      1692       2761       2813       3758       4059       4970
LAB.SIZ  000014     585L      1702       2322       2963       3990       3998       4010       4989
LAB.SPG  000007     573L      1304       2982       3088       3278       3327       3497       3923       3996       4015       4978
LAB.SPT  000117     599L      1704       2251       2258       2304
LAB.SYS  000001     576E      1276
LAB.VER  000011     580L      1697       4982
LAB.VFL  000020     587L      4993
LAB.VLT  000010     579L      1271       4980
LAB.VPL  000005     589E       591        592       3982       4995
LAB.VPR  000014     584E       589       3981       4988       4995
LABEL    070256    2813       2822       2963       2982       3032       3039       3044       3056       3088       3157       3175       3202
                   3278       3288       3299       3321       3327       3497       3758       3923       3981       3984       3990       3996       3998
                   4010       4015       4031       4059       4969L      4970       4972       4974       4976       4978       4980       4982       4985
                   4988       4989       4991       4993       4995       4997       5001
LDE..    047146    1280       1604L      1636
LDE3     047167    1617L      1628
LDO      033012     683E
LEP      045212    1222       1267L
LEP1     045323    1281       1294L
LEP3     045350    1309L
LEP4     046004    1298       1325L
LEP5     046214    1274       1339L
LEP6     046070    1277       1328       1332L
LF       000012    1057E      1164       1482       5005       5007
LINE     070064    4939L      4940       4941
```

```
LINEL   000120      4940E
M.CDCA  000017       617L
M.CDLY  000016       616L
M.CFWA  000012       614L
M.CIN   000006       612L
M.CINT  000005       611L
M.CLWA  000014       615L
M.COUT  000010       613L
M.CPRE  000003       609L
M.CRUB  000004       610L
M.CSLC  000002       608L
M.FOX   000303       100E
M.PAM8  000021        99E
M.SALO  000001       607L
M.SUNI  000021       618L      3801    3837    5412
M.SYDD  000022       619L
M.SYSM  000000       606L
MEML    100011      1137       6023E
MEMLIM  070052      4930L      5214    5338    5359
MI.JMP  000303        32E
MOUNT   065042      3028       3052    3757L
MOUNT.  065055      2882       3041    3766L
MOUNT.. 065062      2740       3769L
MOUNT1  065050      3762L      3767
MSD     047335      1221       1228    1680L
MSD.    050013      1683       1693    1709L
MSU     065067      3796L      3948
MSU1    065133      3817       3819E
MSUA    065142      3828E      5400
NDIRBLK 070054      3106       3117    3133    3159    3161    3210    4931L
NGROUPS 070055      2905       3094    3132    4008    4932L
NL      000012      1068E      1069    2520    2538    2540    2560    2596    2653    2698    2701    2702    2703
                    2704       2717    2718    2743    2767    2778    2805    2863    2953    2954    4043    4484    4916
                    5033       5034    5035    5036    5037    5039    5041    5043    5048    5540    5540    5546    5549
NSPCGRP 070056      3013       3095    4026    4933L
NUL2    000000      1059E
NULL    000200      1058E
OP.CTL  000360        53E
OP.DIG  000360        54E
OP.SEG  000361        55E
OP2.CTL 000362        57E
OVL.COD 000000       644L
OVL.ENS 000010       649E
OVL.ENT 000004       646L
OVL.FLB 000006       647L
OVL.IN  000001       968E
OVL.NUM 000014       970E
OVL.RES 000002       969E
OVL.SIZ 000002       645L
OVL.UCS 000200       971E
OVL0    000000       655L      5084    5555
OVL1    000001       656L      5086    5559
OVRLAY  071203      5018E      6036
PATCH   071103      5014L
PDI     033145       685E
PDN     054372      2501       2594L   2656
PDN.    055026      2603L      5113
PDN1    055050      2611L      2629
```

```
PDN2     055106     2623      2633L
PDN3     055147     2606      2615      2644      2652L
PDNA     055213     2603      2608      2658L
PDNB     055215     2634      2659L
PDNC     055216     2604      2663L
PDS      064012     3277      3344      3461L
PDS1     064020     3463L     3477
PDS2     064025     3470L     3474
PIC.COD  000006      396L      501      1321      5195      5443
PIC.ID   000000      391L     5201      5205      5489      5494
PIC.LEN  000002      393L     1313      5209      5229
PIC.PTR  000004      394L     5441
PRS      071203     5031L     5082
PRS1     072247     5046L     5058
PRSS     072332     1138      5068L
PRSS0    072367     5079      5082L
PRSS1    073056     5069      5071      5118L
PSD      075123     5090      5376L
PSD1     075126     5378L     5389
PSD2     075161     5383      5393L
PSDA     075235     5379      5417L     5418
PSDAL    000004     5378      5418E
PUBLIC   000001        1E     1205
QUOTE    000047     1065E
R.DLY    035303     1051E
R.WHD    036235     1049E
R.WNH    036271     1050E
RCC      050303     1174      1867L
RDP      075241     5240      5431L
READD    047232     1312      1320      1608      1654L
READD1   047233     1656L     1667
READERR  047237     1661L     1712
READR    047330     1666L     1688
REL      033177      689E
REL.     033175      687E     5452
RESTART  054047     2496L     2502      3683      4045
RGTAB    075000     2744      2761      2769      2904      2987      3011      3016      3031      3112      3140      3199      6058L
RMEML    102011     4924      6033L     6060
RMI      055224     2505      2680E
RMI1     055253     2696L     2719
RMIA     055341     2688      2700L
RMIB     055343     2692      2701L
ROMBOOT  030000      708E
RPB      065170     2646      3854L
RPB1     065243     3881L     3894      3905
RUBOUT   000177     1061E
RUC      033257      693E
S.BAUD   040344      945L     1841      2192
S.BDA    041120     1043L
S.BOOTF  041034     1000L     1217      1220      1238
S.CAADR  040333      928L
S.CACC   041006      984L
S.CCTAB  040335      929L
S.CDB    040343      942L     1369      1391      1496      1744      1842      2156
S.CFWA   040352      952L
S.CODE   041007      985L
S.CONFL  040332      926L
S.CONTY  040327      913L     1832      1834      2169      2200
```

```
 S.CONWI 040331     919L
 S.CSLMD 040326     901L     912     915     918     925     5095
 S.CUSOR 040330     916L
 S.DATC  040310     882L    3038    3305    3715
 S.DATE  040277     881L
 S.DCS   041033     998L
 S.DDDTA 040366     963L
 S.DDGRP 040364     960L
 S.DDLDA 040360     958L
 S.DDLEN 040362     959L
 S.DDOPC 040370     964L
 S.DFWA  040354     953L    5128
 S.DIREA 041016     992L
 S.DLINK 040346     950L    3798    3835    5410
 S.FASER 041013     991L
 S.FCI   041021     993L
 S.GRT0  024000     704E
 S.GRT1  025000     705E
 S.GRT2  026000     706E
 S.GUP   041027     995L
 S.HIMEM 040316     884L
 S.INT   040343     718L     938
 S.JUMPS 041010     989L
 S.MOUNT 041032     997L
 S.OFWA  040350     951L
 S.OMAX  040324     890L
 S.OSN   041004     980L
 S.OVLE  041000     977L
 S.OVLFL 040371     973L
 S.OVLS  040376     976L
 S.OVSTK 041035    1005L
 S.READ  031275     665E
 S.RFWA  040356     954L
 S.SCI   041024     994L
 S.SCR   041121    1044L
 S.SDD   041010     990L
 S.SOVR  041146     720L     722
 S.SSN   041002     979L
 S.SYSM  040320     886L
 S.TIME  040312     883L
 S.UCSF  040372     974L
 S.UCSL  040374     975L
 S.USRM  040322     888L
 S.VAL   040277     717L     879
 S.WRITE 031330     667E
 SOBOOT  044200    1152L    2423
 SOBOOT0 044303    1179     1186L
 SOBOOT1 044333    1175     1199L
 SOBOOT2 045076    1183     1189    1191    1213L
 SOBOOT3 045120    1220L    1240
 SOBOOT4 045146    1193     1228L
 SOBOOT5 045170    1195     1236L
 SOBOOTL 007000    2423E    2424    2885
 SOBOOTX 044220    1160L    1209    1232    1290    1335    1341
 SOBOOTY 044253    1173L    1184
 SB.BAU  042205     174L    1770    1773    1776    1822    3732
 SB.BOO  042200     170L     176     177     180    2424    2885    2886    3854    3855    3856    3860    3871
                   3954     5398    5435
```

```
SB.BPE    042240    178E     3741
SB.DAT    042207    175L     3735
SB.DRV    042240    180L
SB.FLG    042204    172L     3729
SB.ORG    051000    164E     1310      1313      1319      1321      1720
SB.OVMX   014000    165E
SB.SDB    044200    182E     1132      1136      1137      2424      2885      3854      3855      3856      3891      3954      5398
SB.VER    042203    171L     3719      3724      3726
SC.ACE    000350    284E     1384      1403      1511      1515      1759      1809      1815      1816      1968      1970      1972
                   1975      1978      1984      1987      1989      1993      1994      2011      2025      2026      2044      2184      2189
                   2191      2194      2196      2199      2206      2209      2210      2212
SC.UART   000372    244E     1375      1397      1502      1506      1751      1810      2163      2164      2165      2166      2168
                   2177      2179
SCU       051334   1843      2156L
SCU1      052004   2158      2184L     2186
SPG       000002     34E
STACK     042200    724E     1152      1160      2496      2500      2504      5080      5107
STACKL    001032    722E
STB       065301   3203      3300      3322      3440      3918L     4018
SUBFUN    065320   2710      2870      3935L     3977      4035
SUBFUN1   065363   3949      3957L
SUNIT     070057   3814      3834      4934L     5413
SVP       065367   2878      3973E
SVP1      066110   4021      4024E
SVP2      066135   4005      4042L
SYDD      040130    714E     1656      1681      1711      2262
SYMNU     070060   3811      4935L     5408
SYPTR     070061   3944      4936L     5393
SYSCALL   000377    734E     2527      4167      4539      4543      4644      4917      5666      5678      5772      5984
T.CHA     077376   5908      5936      5983      6006L     6007      6009      6011      6013      6015      6017
T.FLG     077377   5821      6008L
T.FWA     100000   5972      6010L
T.LIM     100004   5834      5976      5996      5998      6014L
T.LWA     100006   5975      6016L
T.PTR     100002   5832      5884      5973      6012L
T110      051312   2086      2102L
T1200     051266   2067      2080L
T150      051316   2103      2108L
T1800     051306   2081      2098L
T19200    051256   2063      2072L
T300      051272   2068      2084L
T3600     051302   2078      2094L
T4800     051262   2064      2076L
T600      051252   2058      2066L
T7200     051276   2077      2090L
T75       051322   2104      2112L
T9600     051246   2057      2062L
TAB       000011   1066E     4562      5033      5033      5033      5034      5034      5034      5035      5035      5037
TABLE     051242   1965      2054L     2073      2074      2082      2085      2091      2092      2095      2096      2099      2100
                   2109      2110      2113      2114      2117
TBL1      067172   4589L     4595
TBL2      067210   4587      4599L
TBL3      067212   4592      4603L
TDD1      067353   4860L     4883
TDD2      067372   4870L     4872
TDDA      070021   4861      4887E
TDI       075307   5265      5272      5279      5303      5461L
TDIA      075327   5462      5465L
```

```
TFE      033233      691E
TFN      046375     1287      1416L
TFN1     047014     1418      1423L    1428
TLEN     000012     5907      6017E
TPL1     067255     4712L
TTDD     053103     2277      2375L    2417
TTDD.    053101     2374L
TTDD1    053104     2376L     2399
TTDD2    053123     2386L     2388
TTDDA    053152     2377      2403E
TTDDCR   053164     2293      2320     2335     2417L
TYPEC1   047075     1502L     1504
TYPEC2   047112     1498      1511L    1513
TYPEC3   047124     1507      1517L
TYPEC4   047131     1523L     1528
UC.2SB   000004      310E     1983     2203     2204
UC.5BW   000000      306E
UC.6BW   000001      307E
UC.7BW   000002      308E
UC.8BW   000003      309E     1814     1817     1981     1983     2205
UC.BI    000020      329E     1760     1762
UC.CTS   000020      338E
UC.DCS   000001      334E
UC.DDR   000002      335E
UC.DLA   000200      315E     1971     2193
UC.DR    000001      325E     1385     1760     2001
UC.DRL   000010      337E
UC.DSR   000040      339E
UC.DTR   000001      318E
UC.EDA   000001      296E
UC.EPS   000020      312E
UC.FE    000010      328E     1760     1762     2001     2009
UC.IID   000006      303E
UC.IIP   000001      302E
UC.LOO   000020      322E     1969     1988     2190     2211
UC.MSI   000010      299E
UC.OR    000002      326E     1995
UC.OU1   000004      320E
UC.OU2   000010      321E
UC.PE    000004      327E     2001
UC.PEN   000010      311E
UC.RI    000100      340E
UC.RLS   000200      341E
UC.RSI   000004      298E
UC.RTS   000002      319E
UC.SB    000100      314E
UC.SKP   000040      313E
UC.TER   000004      336E
UC.THE   000040      330E     1512
UC.TRE   000002      297E
UC.TSE   000100      331E     2185
UCI.ER   000020      266E     2178
UCI.IE   000002      268E
UCI.IR   000100      264E     2167
UCI.RE   000004      267E     2178
UCI.RO   000040      265E
UCI.TE   000001      269E     2178
UDR      000000      241E     1397     1506
```

```
UMI.16X 000002       259E       2176
UMI.1B  000100       249E       2171    2176
UMI.1X  000001       258E
UMI.2B  000300       251E       2171
UMI.64X 000003       260E
UMI.HB  000200       250E
UMI.L5  000000       254E
UMI.L6  000004       255E
UMI.L7  000010       256E
UMI.L8  000014       257E       2176
UMI.PA  000020       253E
UMI.PE  000040       252E
UNIT    070063      2635        2689    4937L
UNT.DIS 000006       477L
UNT.FLG 000000       473L
UNT.GRT 000002       475L
UNT.GTS 000004       476L
UNT.SIZ 000010       479E
UNT.SFG 000001       474L
UO.CLK  000001        92E
UO.DDU  000002        91E       1955
UO.HLT  000200        89E
UO.NFR  000100        90E
UR.DLL  000000       291E       1975    2196
UR.DLM  000001       293E       1978    2199
UR.IER  000001       295E       1809    1968    2189
UR.IIR  000002       301E
UR.LCR  000003       305E       1815    1816    1972    1984    2194    2206
UR.LSR  000005       324E       1384    1511    1759    1994    2026    2184
UR.MCR  000004       317E       1970    1987    1989    2191    2210    2212
UR.MSR  000006       333E
UR.RBR  000000       287E       1403    1993    2011    2025    2044    2209
UR.THR  000000       289E       1515
USERFWA 042200       725E
USR     000001       242E       1375    1502    1751    1810    2163    2164    2165    2166    2168    2177    2179
USR.BD  000100       273E       1752    1754
USR.FE  000040       274E
USR.OE  000020       275E
USR.PE  000010       276E
USR.RXR 000002       278E       1376    1752
USR.TXE 000004       277E
USR.TXR 000001       279E       1503
VERS    000040       732E       2446    3727    4983    5034    5034    5070
VFL.NSD 000001       588E
VFT     075345      5192        5483L
VFTA    074370      5484        5489    5494    5499    5516E
VOLSIZ  001220        35E
VSN     066205      2485        4059L
XCHGBC  067263      4361        4365    4373    4375    4762L

7384 BYTES FREE
```