```
                    00002
   000.001          00003   DEBUG          EQU      1                       DON'T ASSEMBLE FOR DEBUG
                    00004   ***  SYSTEM I/O HANDLER.
                    00005   *
                    00006   *    J. G. L., 10/77
                    00007   *
                    00008   *    COPYRIGHT HEATH COMPANY.
                    00009   *
                    00010   *    G CHANDLER        78/10    Maintenance Release
                    00011   *                      78/04\
                    00012   *                      79/05>   Release  #50.04.00
                    00013   *                      79/06/
                    00014   *                      79/10    Release  #50.05.00
                    00015   *                      80       Release  #50.06.00
                    00016   *                                /2.0a/ = /80.09.gc/
                    00017   *                                /2.0b/ = /80.10.gc/
                    00018   *                                /2.0c/ = /80.11.gc/
                    00019   *



                    00021   ***  THE SYSTEM I/O HANDLER HANDLES SYSTEM REQUESTS FOR
                    00022   *    READS AND WRITES.
                    00023   *
                    00024   *    IF A MASS STORAGE DEVICE, THIS DOES THE CORRECT STORAGE
                    00025   *    MANAGEMENT. IF A SERIAL DEVICE, THE COMMAND IS PASSED
                    00026   *    ONTO THE DEVICE DRIVER.
```

```
                    00029
                    00030   **   MACHINE INSTRUCTIONS
                    00031
   000.376          00032   MI.CPI        EQU      376Q
   000.303          00033   MI.JMP        EQU      303Q
   000.311          00034   MI.RET        EQU      311Q
                    00035
                    00036
                    00037   **   SYSTEM SYMBOLS
   042.200          00038        XTEXT    U8250
                    00039X



                    00041X  **   8250 UART CONTROL AND BIT DEFINITIONS.
                    00042X
   000.350          00043X  SC.ACE        EQU      350Q              SYSTEM CONSOLE PORT IF 8250 ACE
   000.156          00044X  AC.DLY        EQU      110               220 MIL. SEC. DELAY FOR 8250
                    00045X
   000.000          00046X  UR.RBR        EQU      0                 RECEIVER BUFFER REGISTER (READ ONLY)
                    00047X
   000.000          00048X  UR.THR        EQU      0                 TRANSMITTER HOLDING REGISTER (WRITE ONLY)
                    00049X
   000.000          00050X  UR.DLL        EQU      0                 DIVISOR LATCH (LEAST SIGNIFICANT)
                    00051X
   000.001          00052X  UR.DLM        EQU      1                 DIVISOR LATCH (MOST SIGNIFICANT)
                    00053X
   000.001          00054X  UR.IER        EQU      1                 INTERRUPT ENABLE REGISTER
   000.001          00055X  UC.EDA        EQU      00000001B ENABLE RECEIVED DATA AVAILABLE INTERRUPT
   000.002          00056X  UC.TRE        EQU      00000010B ENABLE TRANSMIT HOLD REGISTER EMPTY INTERRUPT
   000.004          00057X  UC.RSI        EQU      00000100B ENABLE RECEIVE STATUS INTERRUPT
   000.010          00058X  UC.MSI        EQU      00001000B ENABLE MODEM STATUS INTERRUPT
                    00059X
   000.002          00060X  UR.IIR        EQU      2                 INTERRUPT IDENTIFICATION REGISTER
   000.001          00061X  UC.IIP        EQU      00000001B INVERTED INTERRUPT PENDING (0 MEANS PENDING)
   000.006          00062X  UC.IID        EQU      00000110B INTERRUPT ID
                    00063X
   000.003          00064X  UR.LCR        EQU      3                 LINE CONTROL REGISTER
   000.000          00065X  UC.5BW        EQU      00000000B 5 BIT WORDS
   000.001          00066X  UC.6BW        EQU      00000001B 6 BIT WORDS
   000.002          00067X  UC.7BW        EQU      00000010B 7 BIT WORDS
   000.003          00068X  UC.8BW        EQU      00000011B 8 BIT WORDS
   000.004          00069X  UC.2SB        EQU      00000100B TWO STOP BITS SELECTED
   000.010          00070X  UC.PEN        EQU      00001000B PARITY COMPUTATION ENABLED
   000.020          00071X  UC.EPS        EQU      00010000B EVEN PARITY SELECT
   000.040          00072X  UC.SKP        EQU      00100000B STICK PARITY
   000.100          00073X  UC.SB         EQU      01000000B SET BREAK
   000.200          00074X  UC.DLA        EQU      10000000B DIVISOR LATCH ACCESS
                    00075X
   000.004          00076X  UR.MCR        EQU      4                 MODEM CONTROL REGISTER
   000.001          00077X  UC.DTR        EQU      00000001B DATA TERMINAL READY
   000.002          00078X  UC.RTS        EQU      00000010B REQUEST TO SEND
   000.004          00079X  UC.OU1        EQU      00000100B OUT 1
   000.010          00080X  UC.OU2        EQU      00001000B OUT 2
   000.020          00081X  UC.LOO        EQU      00010000B LOOP
                    00082X
```

```
   000.005             00083X  UR.LSR        EQU      5                LINE STATUS REGISTER
   000.001             00084X  UC.DR         EQU      00000001B DATA READY
   000.002             00085X  UC.OR         EQU      00000010B OVERRUN
   000.004             00086X  UC.PE         EQU      00000100B PARITY ERROR
   000.010             00087X  UC.FE         EQU      00001000B FRAMING ERROR
   000.020             00088X  UC.BI         EQU      00010000B BREAK INTERRUPT
   000.040             00089X  UC.THE        EQU      00100000B TRANSMITTER HOLDING REGISTER EMPTY
   000.100             00090X  UC.TSE        EQU      01000000B TRANSMITTER SHIFT REGISTER EMPTY
                       00091X
   000.006             00092X  UR.MSR        EQU      6                MODEM STATUS REGISTER
   000.001             00093X  UC.DCS        EQU      00000001B DELTA CLEAR TO SEND
   000.002             00094X  UC.DDR        EQU      00000010B DELTA DATA SET READY
   000.004             00095X  UC.TER        EQU      00000100B TRAILING EDGE OF RING
   000.010             00096X  UC.DRL        EQU      00001000B DELTA RECEIVE LINE SIGNAL DETECT
   000.020             00097X  UC.CTS        EQU      00010000B CLEAR TO SEND
   000.040             00098X  UC.DSR        EQU      00100000B DATA SET READY
   000.100             00099X  UC.RI         EQU      01000000B RING INDICATOR
   000.200             00100X  UC.RLS        EQU      10000000B RECEIVED LINE SIGNAL DETECT
   042.200             00101         XTEXT   U8251
                       00102X
```

```
                    00105X  **   8251 USART BIT DEFINITIONS.
                    00106X  *
                    00107X
                    00108X  **   PORT ADDRESSES
                    00109X
   000.000          00110X  UDR  EQU      0                DATA REGISTER IS EVEN
   000.001          00111X  USR  EQU      1                STATUS REGISTER IS NEXT
                    00112X
   000.372          00113X  SC.UART       EQU      372Q              CONSOLE USART ADDRESS (IFF 8251)
                    00114X
                    00115X
                    00116X  **   MODE INSTRUCTION CONTROL BITS.
                    00117X
   000.100          00118X  UMI.1B        EQU      01000000B1 STOP BIT
   000.200          00119X  UMI.HB        EQU      10000000B1 1/2 STOP BITS
   000.300          00120X  UMI.2B        EQU      11000000B2 STOP BITS
   000.040          00121X  UMI.PE        EQU      00100000BEVEN PARITY
   000.020          00122X  UMI.PA        EQU      00010000BUSE PARITY
   000.000          00123X  UMI.L5        EQU      00000000B5 BIT CHARACTERS
   000.004          00124X  UMI.L6        EQU      00000100B6 BIT CHARACTERS
   000.010          00125X  UMI.L7        EQU      00001000B7 BIT CHARACTERS
   000.014          00126X  UMI.L8        EQU      00001100B8 BIT CHARACTERS
   000.001          00127X  UMI.1X        EQU      00000001BCLOCK X 1
   000.002          00128X  UMI.16X EQU   00000010BCLOCK X 16
   000.003          00129X  UMI.64X EQU   00000011BCLOCK X 64
                    00130X
                    00131X  **   COMMAND INSTRUCTION BITS.
                    00132X
   000.100          00133X  UCI.IR        EQU      01000000BINTERNAL RESET
   000.040          00134X  UCI.RO        EQU      00100000BREADER-ON CONTROL FLAG
   000.020          00135X  UCI.ER        EQU      00010000BERROR RESET
   000.004          00136X  UCI.RE        EQU      00000100BRECEIVE ENABLE
   000.002          00137X  UCI.IE        EQU      00000010BENABLE INTERRUPTS FLAG
   000.001          00138X  UCI.TE        EQU      00000001BTRANSMIT ENABLE
                    00139X
                    00140X  **   STATUS READ COMMAND BITS.
                    00141X
   000.100          00142X  USR.BD        EQU      01000000BBreak Detect                        /80.08.gc/
   000.040          00143X  USR.FE        EQU      00100000BFRAMING ERROR
   000.020          00144X  USR.OE        EQU      00010000BOVERRUN ERROR
   000.010          00145X  USR.PE        EQU      00001000BPARITY ERROR
   000.004          00146X  USR.TXE EQU   00000100BTRANSMITTER EMPTY
   000.002          00147X  USR.RXR EQU   00000010BRECEIVER READY
   000.001          00148X  USR.TXR EQU   00000001BTRANSMITTER READY
   042.200          00149         XTEXT    H17DEF
                    00150X



                    00152X  **   H17 CONTROL INFORMATION.
                    00153X
   000.177          00154X  DP.DC         EQU      07FH              DISK CONTROL PORT
                    00155X
   000.001          00156X  DF.HD         EQU      00000001BHOLE DETECT
   000.002          00157X  DF.T0         EQU      00000010BTRACK 0 DETECT
   000.004          00158X  DF.WP         EQU      00000100BWRITE PROTECT
   000.010          00159X  DF.SD         EQU      00001000BSYNC DETECT
```

```
                    00160X
   000.001          00161X  DF.WG         EQU      00000001B WRITE GATE ENABLE
   000.002          00162X  DF.DS0        EQU      00000010B DRIVE SELECT 0
   000.004          00163X  DF.DS1        EQU      00000100B DRIVE SELECT 1
   000.010          00164X  DF.DS2        EQU      00001000B DRIVE SELECT 2
   000.020          00165X  DF.MO         EQU      00010000B MOTOR ON (BOTH DRIVES)
   000.040          00166X  DF.DI         EQU      00100000B DIRECTION (0=OUT)
   000.100          00167X  DF.ST         EQU      01000000B STEP COMMAND (ACTIVE HIGH)
   000.200          00168X  DF.WR         EQU      10000000B WRITE ENABLE RAM
                    00169X
                    00170X
                    00171X
                    00172X  *    Drives other than Wangco's need a delay after write before step
                    00173X
   000.173          00174X  H17SDL        EQU      900/15*1024/500+1  H17 step delay, 900 mic sec /80.06.gc/
                    00175X  *                                        = 900/15*2.048
                    00176X
                    00177X
                    00178X
                    00179X  **   DISK UART PORTS AND CONTROL FLAGS.
                    00180X
   000.174          00181X  UP.DP         EQU      07CH              DATA PORT
   000.175          00182X  UP.FC         EQU      07DH              FILL CHARACTER
   000.175          00183X  UP.ST         EQU      07DH              STATUS FLAGS
   000.176          00184X  UP.SC         EQU      07EH              SYN CHARACTER (OUTPUT)
   000.176          00185X  UP.SR         EQU      07EH              SYNC RESET (INPUT)
                    00186X
   000.001          00187X  UF.RDA        EQU      00000001B RECEIVE DATA AVAILABLE
   000.002          00188X  UF.ROR        EQU      00000010B RECEIVER OVERRUN
   000.004          00189X  UF.RPE        EQU      00000100B RECEIVER PARITY ERROR
   000.100          00190X  UF.FCT        EQU      01000000B FILL CHAR TRANSMITTED
   000.200          00191X  UF.TBM        EQU      10000000B TRANSMITTER BUFFER EMPTY
                    00192X
                    00193X
                    00194X
                    00195X  **   CHARACTER DEFINITIONS.
                    00196X
   000.375          00197X  C.DSYN        EQU      0FDH              PREFIX SYNC CHARACTER
   042.200          00198         XTEXT    ASCII
                    00199X


                    00201X  **   ASCII CHARACTER EQUIVALENCES.
                    00202X
   000.015          00203X  CR   EQU      13                CARRIAGE RETURN
   000.012          00204X  LF   EQU      10                LINE FEED
   000.200          00205X  NULL EQU      200Q              PAD CHARACTER
   000.000          00206X  NUL2 EQU      0
   000.007          00207X  BELL EQU      7                 BELL CHARACTER
   000.177          00208X  RUBOUT        EQU      177Q
   000.010          00209X  BKSP EQU      10Q               CTL-H
   000.026          00210X  C.SYN         EQU      26Q               SYNC
   000.002          00211X  C.STX         EQU      2                 STX
   000.047          00212X  QUOTE         EQU      47Q
   000.011          00213X  TAB  EQU      11Q
   000.033          00214X  ESC  EQU      33Q
   000.012          00215X  NL   EQU      12Q               NEW LINE (HDOS SYSTEMS)
```

```
   000.212             00216X  ENL  EQU      NL+200Q          NL + END-OF-LINE-FLAG
   000.014             00217X  FF   EQU      14Q              FORM FEED
   000.001             00218X  CTLA EQU      01Q              CTL-A
   000.002             00219X  CTLB EQU      02Q              CTL-B
   000.003             00220X  CTLC EQU      03Q              CTL-C
   000.004             00221X  CTLD EQU      04Q              CTL-D
   000.017             00222X  CTLO EQU      17Q              CTL-O
   000.020             00223X  CTLP EQU      20Q              CTL-P
   000.021             00224X  CTLQ EQU      21Q              CTL-Q
   000.023             00225X  CTLS EQU      23Q              CTL-S
   000.032             00226X  CTLZ EQU      32Q              CTL-Z
   042.200             00227       XTEXT    MTR
                       00228X
```

```
                    00231X  **   MTR - PAM/8 EQUIVALENCES.
                    00232X  *
                    00233X  *    THIS DECK CONTAINS SYMBOLIC DEFINITIONS USED TO
                    00234X  *    MAKE USE OF THE PAM/8 CODE AND CONTROL BYTES.



                    00236X  **   IO PORTS
                    00237X
  000.360           00238X  IP.PAD        EQU      360Q              PAD INPUT PORT
  000.360           00239X  OP.CTL        EQU      360Q              CONTROL OUTPUT PORT
  000.360           00240X  OP.DIG        EQU      360Q              DIGIT SELECT OUTPUT PORT
  000.361           00241X  OP.SEG        EQU      361Q              SEGMENT SELECT OUTPUT PORT
  000.362           00242X  IP.CON        EQU      362Q              H-88/H-89/HA-8-8 Configuration      /80.07.gc/
  000.362           00243X  OP2.CTL       EQU      362Q              H-88/H-89/HA-8-8 Control Port       /80.07.gc/



                    00245X  **   FRONT PANEL CONTROL BITS.                              /80.07.gc/
                    00246X  *
                    00247X  *    CB.*  set in OP.CTL
                    00248X  *    CB2.* set in OP2.CTL
                    00249X  *
                    00250X
  000.020           00251X  CB.SSI        EQU      00010000B SINGLE STEP INTERRUPT
  000.040           00252X  CB.MTL        EQU      00100000B MONITOR LIGHT
  000.100           00253X  CB.CLI        EQU      01000000B CLOCK INTERRUPT ENABLE
  000.200           00254X  CB.SPK        EQU      10000000B SPEAKER ENABLE
                    00255X
  000.001           00256X  CB2.SSI       EQU      00000001B Single Step Interrupt
  000.002           00257X  CB2.CLI       EQU      00000010B Clock Interrupt Enable
  000.040           00258X  CB2.ORG       EQU      00100000B ORG 0 Select
  000.100           00259X  CB2.SID       EQU      01000000B Side 1 Select



                    00261X  **   Secondary Control Bits
                    00262X



                    00264X  **   MONITOR MODE FLAGS.
                    00265X
  000.000           00266X  DM.MR         EQU      0                 MEMORY READ
  000.001           00267X  DM.MW         EQU      1                 MEMORY WRITE
  000.002           00268X  DM.RR         EQU      2                 REGISTER READ
  000.003           00269X  DM.RW         EQU      3                 REGISTER WRITE



                    00271X  **   USER OPTION BITS.
                    00272X  *
                    00273X  *    THESE BITS ARE SET IN CELL .MFLAG.
                    00274X
  000.200           00275X  UO.HLT        EQU      10000000B DISABLE HALT PROCESSING
  000.100           00276X  UO.NFR        EQU      CB.CLI            NO REFRESH OF FRONT PANEL
  000.002           00277X  UO.DDU        EQU      00000010B DISABLE DISPLAY UPDATE
```

```
   000.001             00278X  UO.CLK        EQU      00000001B ALLOW PRIVATE INTERRUPT PROCESSING



                       00280X  **   MONITOR IDENTIFICATION FLAGS
                       00281X  *
                       00282X  *    THESE BYTES IDENTIFY THE ROM MONITOR.
                       00283X  *    THEY ARE THE VARIOUS VALUES OF LOCATION .IDENT
                       00284X
   000.021             00285X  M.PAM8        EQU      021Q              'LXI' INSTRUCTION AT 000.000 IN PAM-8
   000.303             00286X  M.FOX         EQU      303Q              'JMP' INSTRUCTION AT 000.000 IN FOX ROM



                       00288X  **   Configuration Flags                                          /80.07.gc/
                       00289X  *
                       00290X  *    These bits are read in IP.CON.
                       00291X  *
                       00292X
   000.003             00293X  CN.174M       EQU      00000011B Port 174Q Device-Type Mask
   000.014             00294X  CN.170M       EQU      00001100B Port 170Q Device-Type Mask
   000.020             00295X  CN.PRI        EQU      00010000B Primary/Secondary:  1=>primary == 170Q
   000.040             00296X  CN.MEM        EQU      00100000B Memory Test/Normal Switch:  0=>Test; 1=>Normal
   000.100             00297X  CN.BAU        EQU      01000000B Baud Rate:  0=>9600; 1=>19,200
   000.200             00298X  CN.ABO        EQU      10000000B Auto-Boot:  1=>Auto-Boot
                       00299X
   000.000             00300X  CND.H17       EQU      00B               H-17 Disk,            Valid only in CN.174M
   000.000             00301X  CND.NDI       EQU      00B               No Device Installed,  Valid only in CN.170M
   000.001             00302X  CND.H47       EQU      01B               H-47 Disk



                       00304X  **   ROUTINE ENTRY POINTS.
                       00305X  *
                       00306X
   000.000             00307X  .IDENT        EQU      0000A             IDENTIFICATION LOCATION
   000.053             00308X  .DLY EQU      0053A            DELAY
   001.267             00309X  .LOAD         EQU      1267A             TAPE LOAD
   001.374             00310X  .DUMP         EQU      1374A             TAPE DUMP
   002.136             00311X  .ALARM        EQU      2136A             ALARM ROUTINE
   002.140             00312X  .HORN         EQU      2140A             HORN
   002.172             00313X  .CTC EQU      2172A            CHECK TAPE CHECKSUM
   002.205             00314X  .TPERR        EQU      2205A             TAPE ERROR ROUTINE
   002.264             00315X  .PCHL         EQU      2264A             PCHL INSTRUCTION
   002.265             00316X  .SRS EQU      2265A            SCAN RECORD START
   002.325             00317X  .RNP EQU      2325A            READ NEXT PAIR
   002.331             00318X  .RNB EQU      2331A            READ NEXT BYTE
   002.347             00319X  .CRC EQU      2347A            CRC-16 CALCULATOR
   003.017             00320X  .WNP EQU      3017A            WRITE NEXT PAIR
   003.024             00321X  .WNB EQU      3024A            WRITE NEXT BYTE
   003.122             00322X  .DOD EQU      3122A            DECODE FOR OCTAL DISPLAY
   003.260             00323X  .RCK EQU      3260A            READ CONSOLE KEYSET
   003.356             00324X  .DODA         EQU      3356A             SEGMENT CODE TABLE
```

```
                    00326X  **   RAM CELLS USED BY H8MTR.
                    00327X  *
                    00328X
   040.000          00329X  .START        EQU      40000A            START DUMP ADDRESS
   040.002          00330X  .IOWRK        EQU      40002A            IN OR OUT INSTRUCTION
   040.005          00331X  .REGI         EQU      40005A            DISPLAYED REGISTER INDEX
   040.006          00332X  .DSPROT EQU   40006A           PERIOD FLAG BYTE
   040.007          00333X  .DSPMOD       EQU      40007A            DISPLAY MODE
   040.010          00334X  .MFLAG         EQU      40010A            USER OPTION BYTE
   040.011          00335X  .CTLFLG EQU   40011A           PANEL CONTROL BYTE
   040.013          00336X  .ALEDS         EQU      40013A            ABUSS LEDS
   040.021          00337X  .DLEDS         EQU      40021A            DBUSS LEDS
   040.024          00338X  .ABUSS         EQU      40024A            ABUSS REGISTER
   040.027          00339X  .CRCSUM EQU   40027A           CRCSUM WORD
   040.031          00340X  .TPERRX EQU   40031A           TAPE ERROR EXIT VECTOR
   040.033          00341X  .TICCNT EQU   40033A           CLOCK TICK COUNTER
   040.035          00342X  .REGPTR EQU   40035A           REGISTER POINTER
   040.037          00343X  .UIVEC        EQU      40037A            USER INTERRUPT VECTORS
   040.064          00344X  .NMIRET       EQU      40064A            H88/H89 NMI Return Address  /80.07.gc/
   040.066          00345X  .CTL2FL       EQU      40066A            OP2.CTL Control Byte                /80.07.gc/
   042.200          00346       XTEXT    HDSROM
                    00347X



                    00349X  **   HDOS H17 ROM ENTRY POINTS.
   031.253          00350X       ORG      31253A
                    00351X  *DWRITE EQU   *                         Obsolete          /80.04.gc/
   031.253          00352X       DS       31256A-31253A
                    00353X  *DREAD        EQU      *                          Obsolete          /80.04.gc/
   031.256          00354X       DS       31275A-31256A
   031.275          00355X  S.READ        EQU      *
   031.275          00356X       DS       31321A-31266A
   031.330          00357X  S.WRITE EQU   *
   031.330          00358X       DS       31325A-31311A
   031.344          00359X  ERR.FNO EQU   *
   031.344          00360X       DS       31331A-31325A
   031.350          00361X  ERR.ILR EQU   *
   031.350          00362X       DS       31335A-31331A
   031.354          00363X  CFF  EQU      *
   031.354          00364X       DS       31363A-31335A
   032.002          00365X  DCA  EQU      *
   032.002          00366X       DS       32114A-31363A
   032.133          00367X  FFB  EQU      *
   032.133          00368X       DS       32166A-32114A
   032.205          00369X  FFL  EQU      *
   032.205          00370X       DS       32204A-32166A
                    00371X  *LDD EQU      *
   032.223          00372X       DS       32372A-32204A+1
   033.012          00373X  LDO  EQU      *
   033.012          00374X       DS       33135A-33002A
   033.145          00375X  PDI  EQU      *
   033.145          00376X       DS       33154A-33124A
   033.175          00377X  REL. EQU      *
   033.175          00378X       DS       33156A-33154A
   033.177          00379X  REL  EQU      *
   033.177          00380X       DS       33212A-33156A
```

```
   033.233             00381X  TFE  EQU      *
   033.233             00382X       DS       33232A-33206A
   033.257             00383X  RUC  EQU      *
                       00384X
   037.132             00385X  BOOTA         EQU      37132A            Boot Vectors                          /80.06.gc/
   000.130             00386X  BOOTAL        EQU      00130A            Length of boot vectors                /80.06.gc/
                       00387X
   034.031             00388X  CLOCK         EQU      34031A            Clock vector                          /80.06.GC/
   033.257             00389        XTEXT    FILDEF



                       00391X  **   FILDEF - FILE TYPE DEFINITIONS.
                       00392X  *
                       00393X  *    DB       377Q,FT.XXX
                       00394X
                       00395X
   000.000             00396X  FT.ABS        EQU      0                 ABSOLUTE BINARY
   000.001             00397X  FT.PIC        EQU      1                 POSITION INDEPENDANT CODE
   000.002             00398X  FT.REL        EQU      2                 RELOCATABLE CODE
   000.003             00399X  FT.BAC        EQU      3                 COMPILED BASIC CODE
   033.257             00400        XTEXT    HOSDEF
                       00401X



                       00403X  **   HOSDEF - DEFINE HOS PARAMETER.
                       00404X  *
                       00405X
                       00406X
   000.040             00407X  VERS EQU      2*16+0            VERSION 2.0
                       00408X
   000.377             00409X  SYSCALL       EQU      377Q              SYSCALL INSTRUCTION
                       00410X
                       00411X
   000.000             00412X       ORG      0
                       00413X
                       00414X  *    RESIDENT FUNCTIONS
                       00415X
   000.000             00416X  .EXIT         DS       1                 EXIT (MUST BE FIRST)
   000.001             00417X  .SCIN         DS       1                 SCIN
   000.002             00418X  .SCOUT        DS       1                 SCOUT
   000.003             00419X  .PRINT        DS       1                 PRINT
   000.004             00420X  .READ         DS       1                 READ
   000.005             00421X  .WRITE        DS       1                 WRITE
   000.006             00422X  .CONSL        DS       1                 SET/CLEAR CONSOLE OPTIONS
   000.007             00423X  .CLRCO        DS       1                 CLEAR CONSOLE BUFFER
   000.010             00424X  .LOADO        DS       1                 LOAD AN OVERLAY
   000.011             00425X  .VERS         DS       1                 RETURN HDOS VERSION NUMBER
   000.012             00426X  .SYSRES       DS       1                 PRECEDING FUNCTIONS ARE RESIDENT
                       00427X
                       00428X
                       00429X  *    *HDOSOVL0.SYS*  FUNCTIONS
                       00430X
   000.040             00431X       ORG      40A
                       00432X
```

```
   000.040             00433X  .LINK         DS       1                LINK  (MUST BE FIRST)
   000.041             00434X  .CTLC         DS       1                CTL-C
   000.042             00435X  .OPENR        DS       1                OPENR
   000.043             00436X  .OPENW        DS       1                OPENW
   000.044             00437X  .OPENU        DS       1                OPENU
   000.045             00438X  .OPENC        DS       1                OPENC
   000.046             00439X  .CLOSE        DS       1                CLOSE
   000.047             00440X  .POSIT        DS       1                POSITION
   000.050             00441X  .DELET        DS       1                DELETE
   000.051             00442X  .RENAM        DS       1                RENAME
   000.052             00443X  .SETTP        DS       1                SETTOP
   000.053             00444X  .DECODE       DS       1                NAME DECODE
   000.054             00445X  .NAME         DS       1                GET FILE NAME FROM CHANNEL
   000.055             00446X  .CLEAR        DS       1                CLEAR CHAN
   000.056             00447X  .CLEARA       DS       1                CLEAR ALL CHANS
   000.057             00448X  .ERROR        DS       1                LOOKUP ERROR
   000.060             00449X  .CHFLG        DS       1                CHANGE FLAGS
   000.061             00450X  .DISMT        DS       1                FLAG SYSTEM DISK DISMOUNTED
   000.062             00451X  .LOADD        DS       1                LOAD DEVICE DRIVER
   000.063             00452X  .OPEN         DS       1                Parametrized Open
                       00453X
                       00454X
                       00455X  *    *HDOSOVL1.SYS*  FUNCTIONS
                       00456X
   000.200             00457X       ORG      200Q
                       00458X
   000.200             00459X  .MOUNT        DS       1                MOUNT  (MUST BE FIRST)
   000.201             00460X  .DMOUN        DS       1                DISMOUNT
   000.202             00461X  .MONMS        DS       1                MOUNT/NO MESSAGE
   000.203             00462X  .DMNMS        DS       1                DISMOUNT/NO MESSAGE
   000.204             00463X  .RESET        DS       1                RESET = DISMOUNT/MOUNT OF UNIT
   000.205             00464X  .CLEAN        DS       1                Clean device
   000.206             00465X  .DAD DS       1                Dismount All Disks          /80.08.gc/
   000.207             00466        XTEXT    OVLDEF
                       00467X




                       00469X  **   OVERLAY TABLE ENTRYS.
                       00470X
   000.000             00471X       ORG      0
                       00472X
   000.000             00473X  OVL.COD       DS       2                FIRST SECTOR OF OVERLAY CODE
   000.002             00474X  OVL.SIZ       DS       2                OVERLAY SIZE
   000.004             00475X  OVL.ENT       DS       2                OVERLAY ENTRY POINT
   000.006             00476X  OVL.FLB       DS       1                OVERLAY FLAG BYTE
   000.007             00477X       DS       1                DUMMY BYTE TO ROUND TABLE SIZE UP TO 8
   000.010             00478X  OVL.ENS       EQU      *                OVERLAY ENTRY SIZE
                       00479X
                       00480X  *    OVERLAY INDICES
                       00481X
   000.000             00482X       ORG      0
                       00483X
   000.000             00484X  OVL0 DS       1
   000.001             00485X  OVL1 DS       1
   000.002             00486        XTEXT    DEVDEF
```

```
                    00487X



                    00489X  **   DEVICE TABLE ENTRYS.
                    00490X
   000.000          00491X       ORG      0
                    00492X
   000.000          00493X  DEV.NAM       DS       2                DEVICE NAME
   000.000          00494X  DV.EL         EQU      00000000B END OF DEVICE LIST FLAG
   000.001          00495X  DV.NU         EQU      00000001B DEVICE ENTRY NOT IN USE
                    00496X
   000.002          00497X  DEV.RES       DS       1                DRIVER RESIDENSE CODE
   000.001          00498X  DR.IM         EQU      00000001B DRIVER IN MEMORY
   000.002          00499X  DR.PR         EQU      00000010B DRIVER PERMINANTLY RESIDENT
                    00500X
   000.003          00501X  DEV.JMP       DS       1                JMP TO PROCESSOR
   000.004          00502X  DEV.DDA       DS       2                DRIVER ADDRESS
   000.006          00503X  DEV.FLG       DS       1                FLAG BYTE
   000.001          00504X  DT.DD         EQU      00000001B DIRECTORY DEVICE
   000.002          00505X  DT.CR         EQU      00000010B CAPABLE OF READ OPERATION
   000.004          00506X  DT.CW         EQU      00000100B CAPABLE OF WRITE OPERATION
   000.010          00507X  DT.RN         EQU      00001000B Capable of random access    /80.02.gc/
   000.020          00508X  DT.CH         EQU      00010000B Capable of Character mode   /80.02.gc/
                    00509X
   000.007          00510X  DEV.MUM       DS       1                MOUNTED UNIT MASK
   000.010          00511X  DEV.MNU       DS       1                MAXIMUM NUMBER OF UNITS
   000.011          00512X  DEV.UNT       DS       2                ADDRESS OF UNIT SPECIFIC DATA TABLE
                    00513X
   000.013          00514X  DEV.DVL       DS       2                DRIVER BYTE LENGTH
   000.015          00515X  DEV.DVG       DS       1                DRIVER ROUTINE GROUP ADDRESS
                    00516X
   000.016          00517X  DEVELEN       EQU      *                DEVICE TABLE ENTRY LENGTH



                    00519X  **   UNIT SPECIFIC DEVICE DATA TABLE ENTRIES
                    00520X
   000.000          00521X       ORG      0
                    00522X
   000.000          00523X  UNT.FLG       DS       1                UNIT SPECIFIC  *DEV.FLG*
   000.001          00524X  UNT.SPG       DS       1                Sectors Per Group           /80.04.GC/
   000.002          00525X  UNT.GRT       DS       2                ADDRESS OF GROUP RESERVATION TABLE (IF DT.DD)
   000.004          00526X  UNT.GTS       DS       2                GRT SECTOR NUMBER
   000.006          00527X  UNT.DIS       DS       2                DIRECTORY FIRST SECTOR NUMBER
                    00528X
   000.010          00529X  UNT.SIZ       EQU      *                SIZE OF UNIT SPECIFIC DATA TABLE PER UNIT
   000.010          00530        XTEXT    DIRDEF
                    00531X
```

```
                    00533X  **   DIRECTORY ENTRY FORMAT.
                    00534X
   000.000          00535X       ORG      0
                    00536X
                    00537X
   000.377          00538X  DF.EMP        EQU      377Q              FLAGS ENTRY EMPTY
   000.376          00539X  DF.CLR        EQU      376Q              FLAGS ENTRY EMPTY, REST OF DIR ALSO CLEAR
                    00540X
   000.000          00541X  DIR.NAM       DS       8                 NAME
   000.010          00542X  DIR.EXT       DS       3                 EXTENSION
   000.013          00543X  DIR.PRO       DS       1                 PROJECT
   000.014          00544X  DIR.VER       DS       1                 VERSION
   000.015          00545X  DIRIDL        EQU      *                 FILE IDENTIFICATION LENGTH
                    00546X
   000.015          00547X  DIR.CLU       DS       1                 CLUSTER FACTOR
   000.016          00548X  DIR.FLG       DS       1                 FLAGS
   000.017          00549X       DS       1               RESERVED
   000.020          00550X  DIR.FGN       DS       1                 FIRST GROUP NUMBER
   000.021          00551X  DIR.LGN       DS       1                 LAST GROUP NUMBER
   000.022          00552X  DIR.LSI       DS       1                 LAST SECTOR INDEX (IN LAST GROUP)
   000.023          00553X  DIR.CRD       DS       2                 CREATION DATE
   000.025          00554X  DIR.ALD       DS       2                 LAST ALTERATION DATE
                    00555X
   000.027          00556X  DIRELEN       EQU      *                 DIRECTORY ENTRY LENGTH
   000.027          00557        XTEXT    DISDEF



                    00559X  **   DIRECTORY BLOCK FORMAT.
                    00560X
   000.000          00561X       ORG      0
                    00562X
   000.000          00563X  DIS.ENT EQU    *                 FIRST ENTRY ADDRESS
   000.000          00564X       DS       22*DIRELEN        22 DIRECTORY ENTRYS PER BLOCK
   001.372          00565X       DS       1                 0 BYTE = END OF ENTRYS IN THIS BLOCK
                    00566X
   001.373          00567X       ORG      512-5             AT END OF BLOCK
   001.373          00568X  DIS.ENL DS     1                 LENGTH OF EACH ENTRY (=DIRELEN)
   001.374          00569X  DIS.SEC DS     2                 BLOCK # OF THIS BLOCK,
   001.376          00570X  DIS.LNK DS     2                 BLOCK # OF NEXT BLOCK, =0 IF THIS IS LAST
   002.000          00571        XTEXT    IOCDEF
                    00572X



                    00574X  **   I/O CHANNEL DEFINITIONS.
                    00575X
   000.000          00576X       ORG      0
                    00577X
   000.000          00578X  IOC.LNK       DS       2                 ADDRESS OF NEXT CHANNEL, =0 IF LAST
   000.002          00579X  IOC.DDA       DS       2                 THREAD JUMP TO DEVICE DRIVER (VIA DEV TABLE)
                    00580X
                    00581X
   000.004          00582X  IOC.FLG       DS       1                 FILE TYPE FLAGS
   000.001          00583X  FT.DD         EQU      00000001B=1 IF DIRECTORY DEVICE
   000.002          00584X  FT.OR         EQU      00000010B=1 IF OPEN FOR READ
   000.004          00585X  FT.OW         EQU      00000100B=1 IF OPEN FOR WRITE
```

```
   000.010             00586X  FT.OU         EQU      00001000B=1 IF OPEN FOR UPDATE
   000.020             00587X  FT.OC         EQU      00010000B=1 IF OPEN FOR CHARACTER MODE        /80.02.GC/
   000.003             00588X  IOC.SQL       EQU      *-IOC.DDALENGTH OF INFO FOR SEQUENTIAL FILE (FROM IOC)
                       00589X
   000.005             00590X  IOC.GRT       DS       2                ADDRESS OF GROUP RESERVATION TABLE
   000.007             00591X  IOC.SPG       DS       1                SECTORS PER GROUP, THIS DEVICE
   000.010             00592X  IOC.CGN       DS       1                CURRENT GROUP NUMBER
   000.011             00593X  IOC.CSI       DS       1                CURRENT SECTOR INDEX (IN CURRENT GROUP)
   000.012             00594X  IOC.LGN       DS       1                LAST GROUP NUMBER
   000.013             00595X  IOC.LSI       DS       1                LAST SECTOR INDEX (IN LAST GROUP)
   000.010             00596X  IOC.DRL       EQU      *-IOC.FLGLENGTH OF INFO NORMALLY COPIED BACK TO
                       00597X  *                               THE CHANNEL TABLE
   000.014             00598X  IOC.DTA       DS       2                DEVICE TABLE ADDRESS FOR THIS DEVICE
   000.016             00599X  IOC.DES       DS       2                SECTOR NUMBER OF DIRECTORY ENTRY
   000.020             00600X  IOC.DEV       DS       2                DEVICE CODE
   000.022             00601X  IOC.UNI       DS       1                UNIT NUMBER (0-9)
   000.021             00602X  IOC.DIL       EQU      *-IOC.DDALENGTH OF INFO FOR DIRECTORY FILE (FROM IOC)
                       00603X
   000.023             00604X  IOC.DIR       DS       DIRELEN          DIRECTORY ENTRY
                       00605X
   000.052             00606X  IOCELEN       EQU      *                IOC ENTRY LENGTH
                       00607X
   000.001             00608X  IOCCTD        EQU      1                INDEX OF USER CHANNEL #0 IN CHANTAB (FIRST = 0)
   000.052             00609       XTEXT     DDDEF
                       00610X



                       00612X  **   DEVICE DRIVER COMMUNICATION FLAGS.
                       00613X  *
                       00614X
   000.000             00615X      ORG       0
                       00616X
   000.000             00617X  DC.REA        DS       1                READ
   000.001             00618X  DC.WRI        DS       1                WRITE
   000.002             00619X  DC.RER        DS       1                READ REGARDLESS
   000.003             00620X  DC.OPR        DS       1                OPEN FOR READ
   000.004             00621X  DC.OPW        DS       1                OPEN FOR WRITE
   000.005             00622X  DC.OPU        DS       1                OPEN FOR UPDATE
   000.006             00623X  DC.CLO        DS       1                CLOSE
   000.007             00624X  DC.ABT        DS       1                ABORT
   000.010             00625X  DC.MOU        DS       1                MOUNT DEVICE
   000.011             00626X  DC.LOD        DS       1                LOAD DEVICE DRIVER
   000.012             00627X  DC.RDY        DS       1                Device Ready                          /80.04.GC/
   000.013             00628X  DC.MAX        DS       1                MAXIMUM ENTRY INDEX
   000.014             00629       XTEXT     ECDEF



                       00631X  **   ERROR CODE DEFINITIONS.
                       00632X
   000.000             00633X      ORG       0
   000.000             00634X      DS        1               NO ERROR #0
   000.001             00635X  EC.EOF        DS       1                END OF FILE
   000.002             00636X  EC.EOM        DS       1                END OF MEDIA
   000.003             00637X  EC.ILC        DS       1                ILLEGAL SYSCALL CODE
   000.004             00638X  EC.CNA        DS       1                CHANNEL NOT AVAILABLE
```

```
   000.005              00639X  EC.DNS        DS       1                 DEVICE NOT SUITABLE
   000.006              00640X  EC.IDN        DS       1                 ILLEGAL DEVICE NAME
   000.007              00641X  EC.IFN        DS       1                 ILLEGAL FILE NAME
   000.010              00642X  EC.NRD        DS       1                 NO ROOM FOR DEVICE DRIVER
   000.011              00643X  EC.FNO        DS       1                 CHANNEL NOT OPEN
   000.012              00644X  EC.ILR        DS       1                 ILLEGAL REQUEST
   000.013              00645X  EC.FUC        DS       1                 FILE USAGE CONFLICT
   000.014              00646X  EC.FNF        DS       1                 FILE NAME NOT FOUND
   000.015              00647X  EC.UND        DS       1                 UNKNOWN DEVICE
   000.016              00648X  EC.ICN        DS       1                 ILLEGAL CHANNEL NUMBER
   000.017              00649X  EC.DIF        DS       1                 DIRECTORY FULL
   000.020              00650X  EC.IFC        DS       1                 ILLEGAL FILE CONTENTS
   000.021              00651X  EC.NEM        DS       1                 NOT ENOUGH MEMORY
   000.022              00652X  EC.RF         DS       1                 READ FAILURE
   000.023              00653X  EC.WF         DS       1                 WRITE FAILURE
   000.024              00654X  EC.WPV        DS       1                 WRITE PROTECTION VIOLATION
   000.025              00655X  EC.WP         DS       1                 DISK WRITE PROTECTED
   000.026              00656X  EC.FAP        DS       1                 FILE ALREADY PRESENT
   000.027              00657X  EC.DDA        DS       1                 DEVICE DRIVER ABORT
   000.030              00658X  EC.FL         DS       1                 FILE LOCKED
   000.031              00659X  EC.FAO        DS       1                 FILE ALREADY OPEN
   000.032              00660X  EC.IS         DS       1                 ILLEGAL SWITCH
   000.033              00661X  EC.UUN        DS       1                 UNKNOWN UNIT NUMBER
   000.034              00662X  EC.FNR        DS       1                 FILE NAME REQUIRED
   000.035              00663X  EC.DIW        DS       1                 DEVICE IS NOT WRITABLE (OR WRITE LOCKED)
   000.036              00664X  EC.UNA        DS       1                 UNIT NOT AVAILABLE
   000.037              00665X  EC.ILV        DS       1                 ILLEGAL VALUE
   000.040              00666X  EC.ILO        DS       1                 ILLEGAL OPTION
   000.041              00667X  EC.VPM        DS       1                 VOLUME PRESENTLY MOUNTED ON DEVICE
   000.042              00668X  EC.NVM        DS       1                 NO VOLUME PRESENTLY MOUNTED
   000.043              00669X  EC.FOD        DS       1                 FILE OPEN ON DEVICE
   000.044              00670X  EC.NPM        DS       1                 NO PROVISIONS MADE FOR REMOUNTING MORE DISKS
   000.045              00671X  EC.DNI        DS       1                 DISK NOT INITIALIZED
   000.046              00672X  EC.DNR        DS       1                 DISK IS NOT READABLE
   000.047              00673X  EC.DSC        DS       1                 DISK STRUCTURE IS CORRUPT
   000.050              00674X  EC.NCV        DS       1                 NOT CORRECT VERSION OF HDOS
   000.051              00675X  EC.NOS        DS       1                 NO OPERATING SYSTEM MOUNTED
   000.052              00676X  EC.IOI        DS       1                 ILLEGAL OVERLAY INDEX
   000.053              00677X  EC.OTL        DS       1                 OVERLAY TO LARGE
   000.054              00678         XTEXT    DDFDEF
                        00679X


                        00681X  **   DIRECTORY DEVICE FORMAT DEFINITION.                    /80.09.gc/
                        00682X  *
                        00683X  *    Modified: Sep-80
                        00684X  *                       No longer require 2 sectors per group
                        00685X  *                       Reserved Group Table dynamically allocated
                        00686X  *
                        00687X
   000.000              00688X        ORG      0
                        00689X
   000.000              00690X  DDF.BOO        DS       9                 2K BOOT PROGRAM
   000.011              00691X  DDF.BOL        EQU      *                 LENGTH OF BOOT
   000.011              00692X  DDF.LAB        DS       1                 LABEL SECTOR
   000.012              00693X  DDF.USR        DS       0                 BEGINNING OF OPEN SPACE
```

```
   000.012             00694        XTEXT    LABDEF
                       00695X



                       00697X  **   DISK LABEL SECTOR FORMATS.
                       00698X
   000.000             00699X       ORG      0
   000.000             00700X  LAB.SER       DS       1                 SERIAL NUMBER OF VOLUME
   000.001             00701X  LAB.IND       DS       2                 INITIALIZATION DATE
   000.003             00702X  LAB.DIS       DS       2                 SECTOR NUMBER OF 1ST DIRECTORY SECTOR
   000.005             00703X  LAB.GRT       DS       2                 INDEX OF GRT SECTOR
   000.007             00704X  LAB.SPG       DS       1                 SECTORS PER GROUP
                       00705X
   000.000             00706X  LAB.DAT       EQU      0                 DATA VOLUME ONLY
   000.001             00707X  LAB.SYS       EQU      1                 SYSTEM VOLUME
   000.002             00708X  LAB.NOD       EQU      2                 => LAB.NOD MEANS VOLUME HAS NO DIRECTORY
                       00709X
   000.010             00710X  LAB.VLT       DS       1                 VOLUME TYPE
   000.011             00711X  LAB.VER       DS       1                 VERSION OF INIT17 THAT INITED DISK
                       00712X
   000.012             00713X  LAB.RGT       DS       2                 RGT sector number          /80.06.gc/
                       00714X
   000.014             00715X  LAB.VPR       EQU      *                 Volume dependant data                /80.05.gc/
   000.014             00716X  LAB.SIZ       DS       2                  Volume Size (Bytes/256)   /80.05.gc/
   000.016             00717X  LAB.PSS       DS       2                  Physical Sector Size              /80.05.gc/
   000.020             00718X  LAB.VFL       DS       1                  Volume dependant Flags            /80.09.gc/
   000.001             00719X  VFL.NSD       EQU      00000001B  Number of Sides:  1 => 2  /80.09.gc/
   000.005             00720X  LAB.VPL       EQU      *-LAB.VPRLength of volume dependant data     /80.05.gc/
                       00721X
   000.000             00722X       ERRMI    5-LAB.VPL                                    /80.05.gc/
   000.021             00723X       DS       5-LAB.VPLReserved                 /80.05.gc/
                       00724X
   000.021             00725X  LAB.LAB       DS       60                LABEL
   000.074             00726X  LAB.LBL       EQU      *-LAB.LABLABEL LENGTH
   000.115             00727X       DS       2                Reserved for 0 bytes               /80.09.gc/
                       00728X
   000.117             00729X  LAB.AUX       EQU      *                 Auxiliary Data                       /80.09.gc/
   000.117             00730X  LAB.SPT       DS       1                  Sectors per Track         /80.09.gc/
   000.001             00731X  LAB.AXL       EQU      *-LAB.AUXLength of Aux. Data                 /80.09.gc/
   000.120             00732        XTEXT    PICDEF
                       00733X



                       00735X  **   PIC FORMAT EQUIVALENCES.
                       00736X
   000.000             00737X       ORG      0
                       00738X
   000.000             00739X  PIC.ID        DS       1                 377Q = BINARY FILE FLAG
   000.001             00740X       DS       1                FILE TYPE (FT.PIC)
   000.002             00741X  PIC.LEN       DS       2                 LENGTH OF ENTIRE RECORD
   000.004             00742X  PIC.PTR       DS       2                 INDEX OF START OF PIC TABLE
                       00743X
   000.006             00744X  PIC.COD       DS       0                 CODE STARTS HERE
   000.006             00745        XTEXT    DVDDEF
                       00746X
```

```
                    00748X  **   DEVICE DRIVER EQUIVALENCES.
                    00749X
   000.307          00750X  DVDFLV        EQU      307Q              DEVICE DRIVER FLAG VALUE
                    00751X
   000.006          00752X        ORG      PIC.COD           STARTS AT PIC CODE AREA
                    00753X
   000.006          00754X  DVD.DVD       DS       1                 MUST BE DVDFLV, FLAGS TO HDOS AS DRIVER
   000.007          00755X  DVD.CAP       DS       1                 DEVICE CAPABILITY FLAG
   000.010          00756X  DVD.MUM       DS       1                 MOUNTED UNIT MASK
   000.011          00757X  DVD.MNU       DS       1                 MAXIMUM NUMBER OF UNITS
   000.012          00758X  DVD.UFL       DS       8                 UNIT SUB-CAPABILITY FLAGS FOR UNITS 0-7
   000.022          00759X  DVD.SET       DS       1                 = DVDFLV IFF DRIVER WILL TAKE SET OPTIONS
   000.023          00760X  DVD.INP       DS       2                 Pointer to Init Code               /80.07.gc/
   000.025          00761X        DS       22               RESERVED, MUST BE 0                  /80.07.gc/
   000.053          00762X  DVD.STE       EQU      *                 ENTRY FOR 'SET' INVOCATION
                    00763X
   002.000          00764X  DVD.ENT       EQU      2000A             DRIVER ENTRY POINT (MUST BE MULT OF 256)
   000.053          00765         XTEXT    DIFDEF




                    00767X  **   DIRECTORY FILE FLAGS.
                    00768X
   000.200          00769X  DIF.SYS EQU   10000000B          SYSTEM FILE
   000.100          00770X  DIF.LOC EQU   01000000B          LOCKED FOR CHANGE
   000.040          00771X  DIF.WP        EQU      00100000B         WRITE PROTECTED
   000.020          00772X  DIF.CNT EQU   00010000B          CONTIGUOUS FILE
   000.053          00773         XTEXT    NAMDEF




                    00775X  **   SYSTEM FILE NAME CONVENTIONS
                    00776X  *
                    00777X  *    RGT      .SYS              RESERVED GROUP TABLE (1 SECTOR)
                    00778X  *    GRT      .SYS              GROUP RESERVATION TABLE (1 SECTOR)
                    00779X  *    DIRECT   .SYS              DIRECTORY
                    00780X  *    HOS      .SYS              SYSTEM IMAGE PROGRAM FOR SYSTEM
   000.053          00781         XTEXT    MTRDEF




                    00783X  **   HDOS MONITOR PRIVATE RAM AREA DEFINITIONS.
                    00784X
   000.000          00785X        ORG      0
   000.000          00786X  M.SYSM        DS       1                 SYSCALL ITTERATION COUNT
   000.001          00787X  M.SALO        DS       1                 STAND-ALONE FLAG
   000.002          00788X  M.CSLC        DS       1                 LINES IN CONSOLE BUFFER
   000.003          00789X  M.CPRE        DS       1                 CONSOLE PREVIOUS CHARACTER
   000.004          00790X  M.CRUB        DS       1                 CONSOLE RUBOUT FLAG
   000.005          00791X  M.CINT        DS       1                 CONSOLE INTERRUPT FLAG
   000.006          00792X  M.CIN         DS       2                 CONSOLE CB IN POINTER
   000.010          00793X  M.COUT        DS       2                 CONSOLE CB OUT POINTER
   000.012          00794X  M.CFWA        DS       2                 CONSOLE CB FWA POINTER
   000.014          00795X  M.CLWA        DS       2                 CONSOLE CB LWA POINTER
```

```
   000.016             00796X  M.CDLY        DS       1                CONSOLE PAD CHARACTER COUNT
   000.017             00797X  M.CDCA        DS       2                ADDRESS OF CHARACTER BEING PADDED
   000.021             00798X  M.SUNI        DS       1                System Unit NUmber          /80.05.gc/
   000.022             00799X  M.SYDD        DS       2                Address of Raw System Driver         /80.09.gc/
   000.024             00800       XTEXT    FLTDEF



                       00802X  **   FLTDEF - DEFAULT SECTOR DEFINITIONS
                       00803X
   000.000             00804X      ORG      0
   000.000             00805X  FLT.CTY DS    1                CONSOLE TYPE FLAGS (FOR S.CONTY)
   000.001             00806X  FLT.CWI DS    1                CONSOLE WIDTH (FOR S.CONWI)
   000.002             00807X  FLT.CFC DS    1                CONSOLE FILL CHARACTERS NEEDED
   000.003             00808X  FLT.CRF DS    1                CONSOLE CHARACTER REQUIRING FILL(377Q IF NONE)
   000.004             00809X  FLT.MNC DS    1                MAXIMUM NUMBER OF I/O CHANNELS
   000.005             00810X      DS       1                Hold Place (Formerly Track Delay) /80.06.gc/
   000.006             00811X  FLT.CDB DS    1                CONSOLE DEFINITION BYTE
   000.007             00812X  FLT.CBD DS    2                CONSOLE BAUD RATE
   000.011             00813X  FLT.BOP DS    1                BOOTUP FLAGS
   000.012             00814X  FLT.SAL DS    1                STAND-ALONE FLAG( != 0      => CAN GO STAND-ALONE)
                       00815X
   000.013             00816X  FLT.PBO DS    1                Permanent Boot Options              /80.08.gc/
   000.001             00817X  PBO.DAT EQU   00000001BNo-Date: 0=> No Date                /80.08.gc/
   000.014             00818       XTEXT    HOSEQU
                       00819X



                       00821X  **   HDOS SYSTEM EQUIVALENCES.
                       00822X  *
                       00823X
   024.000             00824X  S.GRT0        EQU      24000A           SYSTEM AREA FOR  GRT0
   025.000             00825X  S.GRT1        EQU      25000A           SYSTEM AREA FOR  GRT1
   026.000             00826X  S.GRT2        EQU      26000A           SYSTEM AREA FOR  GRT2
                       00827X
   030.000             00828X  ROMBOOT       EQU      30000A           ROM BOOT ENTRY
                       00829X
   040.100             00830X      ORG      40100A           FREE SPACE FROM PAM-8
                       00831X
   040.100             00832X      DS       8                JUMP TO SYSTEM EXIT
   040.110             00833X  D.CON         DS       16               DISK CONSTANTS
   040.130             00834X  SYDD EQU      *                SYSTEM DISK ENTRY POINT
   040.130             00835X  D.VEC         DS       24*3             SYSTEM ROM ENTRY VECTORS
   040.240             00836X  D.RAM         DS       31               SYSTEM ROM WORK AREA
   040.277             00837X  S.VAL         DS       36               SYSTEM VALUES
   040.343             00838X  S.INT         DS       115              SYSTEM INTERNAL WORK AREAS
   041.126             00839X      DS       16
   041.146             00840X  S.SOVR        DS       2                STACK OVERFLOW WARNING
   041.150             00841X      DS       42200A-* SYSTEM STACK
   001.032             00842X  STACKL        EQU      *-S.SOVR STACK SIZE
                       00843X
   042.200             00844X  STACK         EQU      *                LWA+1 SYSTEM STACK
   042.200             00845X  USERFWA       EQU      *                USER FWA
   042.200             00846       XTEXT    ESVAL
```

```
                     00847X



                     00849X  **   S.VAL - SYSTEM VALUE DEFINTIONS.
                     00850X  *
                     00851X  *    THESE VALUES ARE SET AND MAINTAINED BY THE SYSTEM.
                     00852X  *
                     00853X  *    THE DECK HOSEQU MUST BE MODIFIED WHEN THIS IS MODIFIED.
                     00854X
                     00855X
   040.277           00856X       ORG      S.VAL
                     00857X
   040.277           00858X  S.DATE        DS       9                SYSTEM DATE (IN ASCII)
   040.310           00859X  S.DATC        DS       2                CODED DATE
   040.312           00860X  S.TIME        DS       4                TIME FROM MIDNIGHT (IN TICS)
   040.316           00861X  S.HIMEM       DS       2                HARDWARE HIGH MEMORY ADRESS+1
                     00862X
   040.320           00863X  S.SYSM        DS       2                FWA RESIDENT SYSTEM
                     00864X
   040.322           00865X  S.USRM        DS       2                LWA USER MEMORY
                     00866X
   040.324           00867X  S.OMAX        DS       2                MAX OVERLAY SIZE FOR SYSTEM
                     00868X
                     00869X
                     00870X  **   THE FOLLOWING FIVE CELLS SHOULD BE MODIFIED/READ ONLY VIA THE .CONSL SYSCALL
                     00871X
   000.200           00872X  CSL.ECH       EQU      10000000B SUPPRESS ECHO
   000.004           00873X  CSL.RAW       EQU      00000100B Raw Mode I/O                        /80.09.gc/
   000.002           00874X  CSL.WRP       EQU      00000010B WRAP LINES AT WIDTH
   000.001           00875X  CSL.CHR       EQU      00000001B OPERATE IN CHARACTER MODE
                     00876X
   000.000           00877X  I.CSLMD       EQU      0                S.CSLMD IS FIRST BYTE
   040.326           00878X  S.CSLMD       DS       1                CONSOLE MODE
                     00879X
   000.200           00880X  CTP.BKS       EQU      10000000B TERMINAL PROCESSES BACKSPACES
   000.100           00881X  CTP.FF        EQU      01000000B Terminal Processes Form-Feed        /80.09.gc/
   000.040           00882X  CTP.MLI       EQU      00100000B MAP LOWER CASE TO UPPER ON INPUT
   000.020           00883X  CTP.MLO       EQU      00010000B MAP LOWER CASE TO UPPER ON OUTPUT
   000.010           00884X  CTP.2SB       EQU      00001000B TERMINAL NEEDS TWO STOP BITS
   000.002           00885X  CTP.BKM       EQU      00000010B MAP BKSP (UPON INPUT) TO RUBOUT
   000.001           00886X  CTP.TAB       EQU      00000001B TERMINAL SUPPORTS TAB CHARACTERS
                     00887X
   000.001           00888X  I.CONTY       EQU      1                S.CONTY IS 2ND BYTE
   000.000           00889X       ERRNZ    *-S.CSLMD-I.CONTY
   040.327           00890X  S.CONTY       DS       1                CONSOLE TYPE FLAGS
   000.002           00891X  I.CUSOR       EQU      2                S.CUSOR IS 3RD BYTE
   000.000           00892X       ERRNZ    *-S.CSLMD-I.CUSOR
   040.330           00893X  S.CUSOR       DS       1                CURRENT CURSOR POSITION
   000.003           00894X  I.CONWI       EQU      3                S.CONWI IS 4TH BYTE
   000.000           00895X       ERRNZ    *-S.CSLMD-I.CONWI
   040.331           00896X  S.CONWI       DS       1                CONSOLE WIDTH
                     00897X
   000.001           00898X  CO.FLG        EQU      00000001B CTL-O FLAG
   000.200           00899X  CS.FLG        EQU      10000000B CTL-S FLAG
                     00900X
```

```
   000.004             00901X  I.CONFL       EQU      4                 S.CONFL IS 5TH BYTE
   000.000             00902X        ERRNZ    *-S.CSLMD-I.CONFL
   040.332             00903X  S.CONFL       DS       1                 CONSOLE FLAGS
                       00904X
   040.333             00905X  S.CAADR       DS       2                 ADDRESS FOR ABORT PROCESSING (>256 IF VALID)
   040.335             00906X  S.CCTAB       DS       6                 ADDR FOR CTL-A, CTL-B, CTL-C PROCESSING
   040.343             00907         XTEXT    ESINT



                       00909X  **   S.INT - SYSTEM INTERNAL WORKAREA DEFINITIONS.
                       00910X  *
                       00911X  *    THESE CELLS ARE REFERENCED BY OVERLAYS AND MAIN CODE, AND
                       00912X  *    MUST THEREFORE RESIDE IN FIXED LOW MEMORY.
                       00913X
                       00914X
   040.343             00915X        ORG      S.INT
                       00916X
                       00917X  **   CONSOLE STATUS FLAGS
                       00918X
   040.343             00919X  S.CDB          DS       1                 CONSOLE DESCRIPTOR BYTE
   000.000             00920X  CDB.H85       EQU      00000000B
   000.001             00921X  CDB.H84       EQU      00000001B=0 IF H8-5, =1 IF H8-4
   040.344             00922X  S.BAUD        DS       2                 [0-14]  H8-4 BAUD RATE, =0 IF H8-5
                       00923X  *                                 [15]    =1 IF BAUD RATE => 2 STOP BITS
                       00924X
                       00925X  **   TABLE ADDRESS WORDS
                       00926X
   040.346             00927X  S.DLINK       DS       2                 ADDRESS OF DATA IN HDOS CODE
   040.350             00928X  S.OFWA        DS       2                 FWA  OVERLAY  TABLE
   040.352             00929X  S.CFWA        DS       2                 FWA  CHANNEL  TABLE
   040.354             00930X  S.DFWA        DS       2                 FWA  DEVICE   TABLE
   040.356             00931X  S.RFWA        DS       2                 FWA  RESIDENT HDOS CODE
                       00932X
                       00933X  **   DEVICE DRIVER DELAYED LOAD FLAGS
                       00934X
   040.360             00935X  S.DDLDA       DS       2                 DRIVER LOAD ADDRESS (HIGH BYTE=0 IF NO LOAD PENDING)
   040.362             00936X  S.DDLEN       DS       2                 CODE LENGTH IN BYTES
   040.364             00937X  S.DDGRP       DS       1                 GROUP NUMBER FOR DRIVER
   040.365             00938X        DS       1                HOLD PLACE
                       00939X  *S.DDSEC      DS       2                 SECTOR NUMBER FOR DRIVER ( * OBSOLETE ! * )
   040.366             00940X  S.DDDTA       DS       2                 DEVICE'S ADDRESS IN DEVLST +DEV.RES
   040.370             00941X  S.DDOPC       DS       1                 OPEN OPCODE PENDEDING
                       00942X
                       00943X  **   OVERLAY MANAGEMENT FLAGS
                       00944X
   000.001             00945X  OVL.IN        EQU      00000001B IN MEMORY
   000.002             00946X  OVL.RES       EQU      00000010B PERMINANTLY RESIDENT
   000.014             00947X  OVL.NUM       EQU      00001100B OVERLAY NUMBER MASK
   000.200             00948X  OVL.UCS       EQU      10000000B USER CODE SWAPPED FOR OVERLAY
                       00949X
   040.371             00950X  S.OVLFL       DS       1                 OVERLAY FLAG
   040.372             00951X  S.UCSF        DS       2                 FWA SWAPPED USER CODE
   040.374             00952X  S.UCSL        DS       2                 LENGTH SWAPPED USER CODE
   040.376             00953X  S.OVLS        DS       2                 SIZE OF OVERLAY CODE
   041.000             00954X  S.OVLE        DS       2                 ENTRY POINT OF OVERLAY CODE
```

```
                    00955X
   041.002          00956X  S.SSN         DS       2                SWAP AREA SECTOR NUMBER
   041.004          00957X  S.OSN         DS       2                OVERLAY SECTOR NUMBER
                    00958X
                    00959X  *    SYSCALL PROCESSING WORK AREAS
                    00960X
   041.006          00961X  S.CACC        DS       1                (ACC) UPON SYSCALL
   041.007          00962X  S.CODE        DS       1                SYSCALL INDEX IN PROGRESS
                    00963X
                    00964X  *    JUMPS TO ROUTINES IN RESIDENT HDOS CODE
                    00965X
   041.010          00966X  S.JUMPS       DS       0                START OF DUMP VECTORS
   041.010          00967X  S.SDD         DS       3                JUMP TO STAND-IN DEVICE DRIVER
   041.013          00968X  S.FASER       DS       3                JUMP TO FATSERR (FATAL SYSTEM ERROR)
   041.016          00969X  S.DIREA       DS       3                JUMP TO DIREAD (DISK FILE READ)
   041.021          00970X  S.FCI         DS       3                JUMP TO FCI (FETCH CHANNEL INFO)
   041.024          00971X  S.SCI         DS       3                JUMP TO SCI (STORE CHANNEL INFO)
   041.027          00972X  S.GUP         DS       3                JUMP TO GUP (GET UNIT POINTER)
                    00973X
   041.032          00974X  S.MOUNT       DS       1                <>0 IF THE SYSTEM DISK IS MOUNTED
   041.033          00975X  S.DCS         DS       1                DEFAULT CLUSTER SIZE-1
                    00976X
   041.034          00977X  S.BOOTF       DS       1                BOOT FLAGS
   000.001          00978X  BOOT.P        EQU      00000001BEXECUTE PROLOGUE UPON BOOTUP
                    00979X
                    00980X  *    STACK VALUE SAVED FOR OVERLAY SYSCALLS
                    00981X
   041.035          00982X  S.OVSTK       DS       2                VALUE OF SP UPON SYSCALLS USING OVERLAY
                    00983X
   041.037          00984X        DS       1                RESERVED



                    00986X  **   ACTIVE I/O AREA.
                    00987X  *
                    00988X  *    THE AIO.XXX AREA CONTAINS INFORMATION ABOUT THE I/O OPERATION
                    00989X  *    CURRENTLY BEING PERFORMED. THE INFORMATION IS OBTAINED FROM
                    00990X  *    THE CHANNEL TABLE, AND WILL BE RESTORED THERE WHEN DONE.
                    00991X  *
                    00992X  *    NORMALLY, THE AIO.XXX INFORMATION WOULD BE OBTAINED DIRECTLY
                    00993X  *    FROM VARIOUS SYSTEM TABLES VIA POINTER REGISTERS. SINCE THE
                    00994X  *    8080 HAS NO GOOD INDEXED ADDRESSING, THE DATA IS MANUALLY
                    00995X  *    COPIED INTO THE AIO.XXX CELLS BEFORE PROCESSING, AND
                    00996X  *    BACKDATED AFTER PROCESSING.
                    00997X
   041.040          00998X  AIO.VEC       DS       3                JUMP INSTRUCTION
   041.041          00999X  AIO.DDA       EQU      *-2              DEVICE DRIVER ADDRESS
   041.043          01000X  AIO.FLG       DS       1                FLAG BYTE
   041.044          01001X  AIO.GRT       DS       2                ADDRESS OF GROUP RESERV TABLE
   041.046          01002X  AIO.SPG       DS       1                SECTORS PER GROUP
   041.047          01003X  AIO.CGN       DS       1                CURRENT GROUP NUMBER
   041.050          01004X  AIO.CSI       DS       1                CURRENT SECTOR INDEX
   041.051          01005X  AIO.LGN       DS       1                LAST GROUP NUMBER
   041.052          01006X  AIO.LSI       DS       1                LAST SECTOR INDEX
   041.053          01007X  AIO.DTA       DS       2                DEVICE TABLE ADDRESS
   041.055          01008X  AIO.DES       DS       2                DIRECTORY SECTOR
```

```
   041.057              01009X  AIO.DEV       DS       2                 DEVICE CODE
   041.061              01010X  AIO.UNI       DS       1                 UNIT NUMBER (0-9)
                        01011X
   041.062              01012X  AIO.DIR       DS       DIRELEN           DIRECTORY ENTRY
                        01013X
   041.111              01014X  AIO.CNT       DS       1                 SECTOR COUNT
   041.112              01015X  AIO.EOM       DS       1                 END OF MEDIA FLAG
   041.113              01016X  AIO.EOF       DS       1                 END OF FILE FLAG
   041.114              01017X  AIO.TFP       DS       2                 TEMP FILE POINTERS
   041.116              01018X  AIO.CHA       DS       2                 ADDRESS OF CHANNEL BLOCK (IOC.DDA)




   041.120              01020X  S.BDA         DS       1                 Boot Device Address (Setup by ROM) /80.09.gc/
   041.121              01021X  S.SCR         DS       2                 SYSTEM SCRATCH AREA ADDRESS
   041.123              01022         XTEXT    BOODEF
                        01023X




                        01025X  **   BOODEF - SPECIAL BOOT-HDOS INTERFACE DEFINTIONS.       /80.05.gc/
                        01026X
   051.000              01027X  SB.ORG        EQU      51000A            ORG FOR LOAD OF INITIAL HDOS.SAV
   014.000              01028X  SB.OVMX       EQU      14000A            SIZE OF HOLD AREA FOR SWAPPED USER CODE
                        01029X  *                                 (=MAX SIZE OF HDOSOVL.SYS)
                        01030X
   042.200              01031X        ORG     42200A
                        01032X
   042.200              01033X  SB.BOO        DS       3                 Jump to Boot routine
   042.203              01034X  SB.VER        DS       1                 Version of INIT that built disk
   042.204              01035X  SB.FLG        DS       1                 Boot Flags
   000.001              01036X  BFLG.A        EQU      00000001BAuto-Boot:  1 => Boot
   042.205              01037X  SB.BAU        DS       2                 Baud Rate Divisor (0=>ignore)
   042.207              01038X  SB.DAT        DS       2                 Default Date
   000.027              01039X        ERRMI   SB.BOO+32-*
   042.211              01040X        DS      SB.BOO+32-*       Reserved
   042.240              01041X  SB.BPE        EQU      *                 End of BOOT-parameters
                        01042X
   042.240              01043X  SB.DRV        DS       SB.BOO+512-*      Primary Boot
                        01044X
   044.200              01045X  SB.SDB        EQU      *                 Secondary Boot
                        01046
                        01047
                        01048
                        01049   *    CODE     P,SB.ORG          POSITION INDEPENDANT CODE
   051.006              01050         CODE     P,SB.ORG+6                Heath ASM includes header   /09.13.GFR/
   051.006              01051         CODE     -R                        THIS CODE WILL NOT BE RELOCATED
                        01052
                        01055
                        01056   **   TEMP INITIALIZE
                        01057
   051.006  303 025 051 01058   HOSBOOT JMP    HOSBOO1           PERFORM BOOT
                        01059
                        01060   *    DEFAULT VALUES FOR SYSTEM
                        01061
```

```
   051.011              01062   HOSTAB        DS       0                 DEFAULT VALUE TABLE
   000.000              01063         ERRNZ    *-HOSTAB-FLT.CTY
                        01064   *             SETUP S.CONTY TO MAP LOWER CASE, ALLOW BKSP, USE 2 STOP BITS.
   051.011  072         01065         DB       CTP.MLO+CTP.MLI+CTP.BKM+CTP.2SB
   000.000              01066         ERRNZ    *-HOSTAB-FLT.CWI
   051.012  120         01067         DB       80                S.CONWI
   000.000              01068         ERRNZ    *-HOSTAB-FLT.CFC
   051.013  004         01069         DB       4                 NUMBER OF FILL CHARACTERS
   000.000              01070         ERRNZ    *-HOSTAB-FLT.CRF
   051.014  015         01071         DB       CR                CHARACTER TO BE FILLED
   000.000              01072         ERRNZ    *-HOSTAB-FLT.MNC
   051.015  006         01073         DB       6                 NUMBER OF CHANNELS
   051.016  000         01074         DB       0                 Hold Place
   000.000              01075         ERRNZ    *-HOSTAB-FLT.CDB
   051.017  000         01076         DB       CDB.H85           H8-F CONSOLE
   000.000              01077         ERRNZ    *-HOSTAB-FLT.CBD
   051.020  000 200     01078         DW       200000A           BAUD => 2 STOP BITS FOR H8-5
   000.000              01079         ERRNZ    *-HOSTAB-FLT.BOP
   051.022  000         01080         DB       0                 BOOT OPTION FLAGS
   000.000              01081         ERRNZ    *-HOSTAB-FLT.SAL
   051.023  000         01082         DB       0                 STAND-ALONE OPTION
   000.000              01083         ERRNZ    *-HOSTAB-FLT.PBO
   051.024  001         01084         DB       PBO.DAT           Permanent Boot Options          /80.08.gc/
                        01085
                        01086   *     END OF DEFAULT TABLE. START OF BOOT CODE
                        01087
   051.025              01088   HOSBOO1 EQU   *
   051.025  061 200 042 01089         LXI      SP,STACK  SET UP THE NEW STACK
   000.001              01090         IF       DEBUG
                        01094         ENDIF
                        01095
                        01096   *     BOOT CODE
                        01097
   051.030  315 154 051 01098         CALL     SBD               Save Boot Data                  /80.08.gc/
   051.033  041 312 040 01099         LXI      H,S.TIME
   051.036  006 204     01100         MVI      B,AIO.CHA-S.TIME
   051.040  315 212 031 01101         CALL     $ZERO             ZERO OUT LOTS OF MEMORY
   051.043  315 337 052 01102         CALL     SDV               SETUP DEFAULT SYSTEM VALUES
   051.046  315 217 051 01103         CALL     RRH               RELOCATE RESIDENT HDOS CODE
   051.051  315 113 052 01104         CALL     SRR               SET UP ROM REPLACEMENTS
   051.054  315 114 052 01105         CALL     SLR               SET LOW MEMORY REFERENCES
   051.057  315 337 052 01106         CALL     SDV               SETUP DEFAULT SYSTEM VALUES
   051.062  315 021 053 01107         CALL     SCD               SETUP CONSOLE DRIVER
   051.065  315 215 053 01108         CALL     GVM               GIVE VERSION MESSAGE            /80.05.gc/
   051.070  315 270 053 01109         CALL     FSM               Fake System Mount          /80.05.gc/
   051.073  315 356 053 01110         CALL     LSO               LOCATE SYSTEM OVERLAYS          /80.05.gc/
   051.076  315 364 054 01111         CALL     SDT               SETUP DEVICE TABLES             /80.05.gc/
   051.101  315 377 060 01112         CALL     SSD               SET SYSTEM DATE                 /80.05.gc/
   051.104  315 164 056 01113         CALL     MSD               Mount System Diskette           /80.05.gc/
   051.107  315 323 061 01114         CALL     UBP               Update Boot Parameters          /80.06.gc/
                        01115
   051.112  072 034 041 01116         LDA      S.BOOTF
   051.115  346 001     01117         ANI      BOOT.P
   051.117  312 127 051 01118         JZ       HOSB2             IGNORE PROLOGUE FILE
                        01119
   051.122  041 133 051 01120         LXI      H,HOSBA
   051.125  377 040     01121         SCALL    .LINK             TRY TO LINK TO PROLOGUE
```

```
                        01122
   051.127  076 001     01123   HOSB2         MVI      A,1               COULDN'T FINE PROFILE, SO TRY NORMAL
   051.131  377 000     01124         SCALL    .EXIT
                        01125
   051.133  123 131     01126   HOSBA         DB       'SY'                                                  /80.05.gc/
   051.135  060 072 120 01127         DB       '0:PROLOGUE.SYS',0 PROLOGUE FILE             /80.05.gc/
```

```
                     01130   **   SBD      - Save Boot Data                              /80.05.GC/
                     01131   *
                     01132   *    SBD saves the data determined at boot time.
                     01133   *
                     01134
  051.154  072 343 040 01135   SBD  LDA      S.CDB
  051.157  062 017 051 01136        STA      HOSTAB+FLT.CDB
                     01137
  051.162  052 344 040 01138        LHLD     S.BAUD
  051.165  042 020 051 01139        SHLD     HOSTAB+FLT.CBD
                     01140
  051.170  041 011 051 01141        LXI      H,HOSTAB+FLT.CTY
  051.173  072 327 040 01142        LDA      S.CONTY
  051.176  346 010     01143        ANI      CTP.2SB
  051.200  266         01144        ORA      M
  051.201  167         01145        MOV      M,A
                     01146
  051.202  072 034 041 01147        LDA      S.BOOTF
  051.205  062 022 051 01148        STA      HOSTAB+FLT.BOP     SAVE THE BOOT FLAGS
                     01149
  051.210  072 061 041 01150        LDA      AIO.UNI
  051.213  062 321 077 01151        STA      SUNIT              Save the system unit number
                     01152
  051.216  311         01153        RET
```

```
                     01156   **   RRH - RELOCATE CODE.
                     01157   *
                     01158   *    RRH IS CALLED TO RELOCATE THE HOS CODE INTO HIGH MEMORY.
                     01159   *
                     01160   *    Modified:Aug-90
                     01161   *                         64K RAM System support
                     01162   *
                     01163   *    ENTRY    NONE
                     01164   *    EXIT     (DE) = DISPLACEMENT FACTOR
                     01165   *    USES     ALL
                     01166
   051.217           01167   RRH  EQU      *
   051.217  041 240 051 01168        LXI      H,RRH2             START AT RRH2
   051.222  056 000     01169        MVI      L,0                START AT 256 BOUNDARY
   051.224  044         01170   RRH1 INR      H                  TRY NEXT BLOCK
   051.225  312 237 051 01171        JZ       RRH1.5             Wrap through high memory    /80.08.gc/
   051.230  176         01172        MOV      A,M
   051.231  064         01173        INR      M
   051.232  276         01174        CMP      M
   051.233  167         01175        MOV      M,A                RESTORE
   051.234  302 224 051 01176        JNE      RRH1               WAS RAM
   051.237  053         01177   RRH1.5       DCX      H                 (HL) = HIGHMEM                        /80.08.gc/
                     01178
                     01179   *    (HL) = HIGHMEM ADDRESS
                     01180
   000.001           01181        IF       DEBUG
                     01186        ELSE
   051.240           01187   RRH2 EQU      *
                     01188        ENDIF
   051.240  042 316 040 01189   RRH2.5       SHLD     S.HIMEM           SET HARDWARE HIGH MEM
   051.243  043         01190        INX      H                  (HL) = LWA+1
   051.244  174         01191        MOV      A,H
   051.245  326 040     01192        SUI      40Q
   051.247  037         01193        RAR
   051.250  037         01194        RAR
   051.251  346 077     01195        ANI      77Q                (A) = # OF K
   051.253  137         01196        MOV      E,A
   051.254  026 000     01197        MVI      D,0
   051.256  315 107 064 01198        CALL     $TYPET
   051.261  000 123 131 01199        DB       0,'SYSTEM HAS',' '+200Q
   051.275  076 002     01200        MVI      A,2
   051.277  315 316 060 01201        CALL     TDD                TYPE NUMBER OF K
   051.302  315 107 064 01202        CALL     $TYPET
   051.305  113 040 117 01203        DB       'K OF RAM',200Q
   051.316  072 317 040 01204        LDA      S.HIMEM+1(A) = SIZE
   051.321  326 177     01205        SUI      24*4+40Q-1         24 K
   051.323  322 003 052 01206        JNC      RRH3               ENOUGH ROOM                                   /2.0a/
   051.326  315 107 064 01207        CALL     $TYPET
   051.331  007 077 060 01208        DB       BELL,'?01 HDOS REQUIRES AT LEAST 24K!',0,BELL+200Q     /2.0a/
   051.373  257         01209        XRA      A                                                             /2.0a/
   051.374  323 351     01210        OUT      SC.ACE+UR.IER      Turn off Interrupts                        /2.0a/
   051.376  323 373     01211        OUT      SC.UART+USR                                                   /2.0a/
   052.000  303 000 052 01212        JMP      *                  Wait for the rapture                       /2.0a/
                     01213
                     01214   *    HAVE ENOUGH ROOM
                     01215
   052.003  052 316 040 01216   RRH3 LHLD     S.HIMEM
```

```
                    01217   *    LXI      D,FWASYS-LWASYS-4                          /79.11.GC/
052.006  021 215 362 01218        LXI      D,FWASYS-LWASYS+1                          /79.11.GC/
052.011  031         01219        DAD      D                 (HL) = NEW FWASYS
052.012  021 342 064 01220        LXI      D,FWAREL
052.015  175         01221        MOV      A,L
052.016  223         01222        SUB      E
052.017  117         01223        MOV      C,A
052.020  174         01224        MOV      A,H
052.021  232         01225        SBB      D
052.022  107         01226        MOV      B,A               (BC) = DISPLACEMENT
052.023  305         01227        PUSH     B                 SAVE
052.024  001 164 015 01228        LXI      B,LWASYS-FWASYS (BC) = SYSTEM RESIDENSE LENGTH
052.027  315 252 030 01229        CALL     $MOVE             MOVE INTO PLACE
                    01230
                    01231   *    RELOCATE REFERENCEES
                    01232
052.032  321         01233        POP      D                 (DE) = RELOCATION FACTOR
052.033  052 004 051 01234        LHLD     SB.ORG+PIC.PTR
052.036  001 000 051 01235        LXI      B,SB.ORG
052.041  011         01236        DAD      B                 (HL) = REL TABLE ADDRESS
                    01237
                    01238   *    RELOCATE CELLS IN BOOT CODE ITSELF
                    01239
052.042  325         01240   RRH4 PUSH     D                 SAVE RELOCATION FACTOR
052.043  136         01241        MOV      E,M
052.044  043         01242        INX      H
052.045  126         01243        MOV      D,M
052.046  043         01244        INX      H                 (DE) = REL ADDRESS OF WORD TO RELOCATE
052.047  172         01245        MOV      A,D
052.050  263         01246        ORA      E
052.051  312 066 052 01247        JZ       RRH6              ALL DONE
                    01248
                    01249   *    SEE IF ADDRESS IS BEYOND FWAREL
                    01250
052.054  001 342 064 01251        LXI      B,FWAREL          (BC) = BREAK BETWEEN ABS PRESET AND REL HDOS
052.057  173         01252        MOV      A,E
052.060  221         01253        SUB      C
052.061  172         01254        MOV      A,D
052.062  230         01255        SBB      B
052.063  332 076 052 01256        JC       RRH5              NOT BEYOND
                    01257
                    01258   *    LET REL ROUTINE RELOCATE REST OF CODE
                    01259
052.066  001 376 377 01260   RRH6 LXI      B,-2
052.071  011         01261        DAD      B                 BACKUP (HL)
052.072  301         01262        POP      B                 (BC) = REL FACTOR
052.073  303 175 033 01263        JMP      REL.              RELOCATE AND EXIT
                    01264
                    01265   *    (DE) = INDEX OF WORD TO RELOCATE
                    01266   *    (HL) = RELOCATION TABLE ADDRESS
                    01267   *    (BC) = CODE DISPLACEMENT FACTOR
                    01268   *    ((SP)) = CODE RELOCATION FACTOR
                    01269
052.076  343         01270   RRH5 XTHL                      (HL) = CODE RELOCATION FACTOR
052.077  032         01271        LDAX     D
052.100  205         01272        ADD      L                 RELOCATE WORD OF CODE
052.101  022         01273        STAX     D
```

```
   052.102  023         01274        INX      D
   052.103  032         01275        LDAX     D
   052.104  214         01276        ADC      H
   052.105  022         01277        STAX     D               RELOCATE
   052.106  353         01278        XCHG                     (DE) = RELOCATION FACTOR
   052.107  341         01279        POP      H               (HL) = RELOCATION TABLE ENTRY ADDRESS
   052.110  303 042 052 01280        JMP      RRH4            DO IT AGAIN
```

```
                     01283   **   SRR      -  SET UP ROM REPLACEMENTS                  /80.06.gc/
                     01284   *
                     01285   *    SET UP RAM REPLACEMENTS FOR THE ROM CODE.
                     01286   *
                     01287   *    The H17 specific ROM replacement has been moved to
                     01288   *    the H17 device driver.
                     01289   *                                        G. Chandler, 80.06.19
                     01290   *
                     01291
  052.113            01292   SRR  EQU      *
                     01293
  052.113  311      01294        RET
```

```
                     01297   **   SLR - SETUP LOW MEMORY REFERENCES.
                     01298   *
                     01299
   052.114           01300   SLR  EQU      *
   052.114  041 342 064 01301        LXI      H,SYSCAL
   052.117  042 062 040 01302        SHLD     .UIVEC+18+1        SETUP SYSCALL LINKAGE
   052.122  076 201     01303        MVI      A,UO.CLK+UO.HLT    DISABLE HALT PROCESSING &
   052.124  062 010 040 01304        STA      .MFLAG             REQUEST CLOCK INTERRUPTS
                     01305
                     01306   *    SETUP EXIT VECTOR AT 40100A
                     01307
   052.127  315 223 062 01308        CALL     $MOVEL
   052.132  010 000 305 01309        DW       SLRAL,SLRA,40100A                          /80.06.gc/
                     01310
                     01311   *    SETUP LOW-MEMORY STUFF
                     01312
   052.140  076 003     01313        MVI      A,4-1              (A) = DEFAULT CLUSTER-1
   052.142  062 033 041 01314        STA      S.DCS              SET DEFAULT CLUSTER SIZE
                     01315
   052.145  041 342 064 01316        LXI      H,FWASYS
   052.150  042 320 040 01317        SHLD     S.SYSM             SET SYSTEM FWA
   052.153  042 356 040 01318        SHLD     S.RFWA             SET RESIDENT CODE FWA
   052.156  041 200 042 01319        LXI      H,USERFWA
   052.161  042 322 040 01320        SHLD     S.USRM             SET LWA USER MEMORY
                     01321
   052.164  257         01322        XRA      A
   052.165  062 326 040 01323        STA      S.CSLMD            CLEAR CONSOLE MODE
   052.170  062 330 040 01324        STA      S.CUSOR            CLEAR CURRSOR ADDRESS
   052.173  257         01325        XRA      A
   052.174  062 332 040 01326        STA      S.CONFL            CLEAR CONSOLE FLAGS                  /80.06.gc/
                     01327
   052.177  041 300 077 01328        LXI      H,HIGHDAT
   052.202  042 346 040 01329        SHLD     S.DLINK            SET DATA LINK
   052.205  041 036 076 01330        LXI      H,OVLTAB
   052.210  042 350 040 01331        SHLD     S.OFWA
   052.213  041 232 076 01332        LXI      H,CHANTAB
   052.216  042 352 040 01333        SHLD     S.CFWA
   052.221  041 056 076 01334        LXI      H,DEVLST
   052.224  042 354 040 01335        SHLD     S.DFWA
                     01336
   052.227  257         01337        XRA      A
   052.230  062 371 040 01338        STA      S.OVLFL            CLEAR OVL RESIDENCE
   052.233  076 060     01339        MVI      A,CTP.MLI+CTP.MLO
   052.235  062 327 040 01340        STA      S.CONTY            INITIALIZE CONSOLE TYPE
                     01341
   052.240  041 126 100 01342        LXI      H,SECSCR
   052.243  042 121 041 01343        SHLD     S.SCR              SET UP OF SYSTEM SCRATCH POINTER
                     01344
                     01345   *    SETUP JUMP VECTORS
                     01346
   052.246  076 303     01347        MVI      A,MI.JMP
   052.250  062 040 041 01348        STA      AIO.VEC
   052.253  315 223 062 01349        CALL     $MOVEL
   052.256  022 000 315 01350        DW       SLRBL,SLRB,S.JUMPS SETUP JUMP VECTORS
                     01351
   052.264  257         01352        XRA      A                  System Unit                      /80.05.gc/
   052.265  062 061 041 01353        STA      AIO.UNI            Set the Boot Unit          /80.05.gc/
```

```
   052.270  052 131 040 01354        LHLD     SYDD+1                                          /80.05.gc/
   052.273  042 322 077 01355        SHLD     MSYDD             Save current SYDD          /80.05.gc/
   052.276  041 202 075 01356        LXI      H,ISY                                           /80.05.gc/
   052.301  042 131 040 01357        SHLD     SYDD+1            Stuff new SYD                 /80.05.gc/
   052.304  311         01358        RET
                         01359
   052.305              01360   SLRA DS       0                 CODE FOR 40100A
   052.305  257         01361        XRA      A
   052.306  062 300 077 01362        STA      SYSMODE
   052.311  076 001      01363        MVI      A,1               FLAT RESET
   052.313  377 000      01364        SCALL    .EXIT
   000.010              01365   SLRAL        EQU      *-SLRA
   377.377              01366        ERRPL    SLRAL-9           ONLY ROOM FOR 8 BYTES
                         01367
   052.315              01368   SLRB DS       0                 JUMP VECTOR CONTENTS
   000.000              01369        ERRNZ    *-SLRB+S.JUMPS-S.SDD
   052.315  303 373 074 01370        JMP      SDD
   000.000              01371        ERRNZ    *-SLRB+S.JUMPS-S.FASER
   052.320  303 115 066 01372        JMP      FATSERR
   000.000              01373        ERRNZ    *-SLRB+S.JUMPS-S.DIREA
   052.323  303 107 072 01374        JMP      DIREAD
   000.000              01375        ERRNZ    *-SLRB+S.JUMPS-S.FCI
   052.326  303 256 073 01376        JMP      FCI
   000.000              01377        ERRNZ    *-SLRB+S.JUMPS-S.SCI
   052.331  303 347 074 01378        JMP      SCI
   000.000              01379        ERRNZ    *-SLRB+S.JUMPS-S.GUP
   052.334  303 035 075 01380        JMP      GUP
   000.022              01381   SLRBL        EQU      *-SLRB
```

```
                     01384   **   SDV - SETUP SYSTEM DEFAULT VALUES.
                     01385   *
                     01386   *    SDV SETS UP THE SYSTEM DEFAULT VALUES CONTAINED IN *SHOTAB*,
                     01387   *    AS DESCRIBED IN *FLTDEF.COM*
                     01388   *
                     01389   *    THESE VALUES CAN BE SET IN THE HDOS.SYS BINARY BY THE *SET*
                     01390   *    UTILITY, AND ARE PROPIGATED INTO THE PROPER SPOTS AT
                     01391   *    BOOT TIME.
                     01392   *
                     01393   *    ENTRY    NONE
                     01394   *    EXIT     NONE
                     01395   *    USES     ALL
                     01396
                     01397
  052.337  072 011 051 01398   SDV  LDA      HOSTAB+FLT.CTY    CONSOLE TYPE FLAGS
  052.342  062 327 040 01399        STA      S.CONTY
  052.345  072 012 051 01400        LDA      HOSTAB+FLT.CWI    (A) = CONSOLE WIDTH
  052.350  062 331 040 01401        STA      S.CONWI
  052.353  072 013 051 01402        LDA      HOSTAB+FLT.CFC    (A) = # OF FILL CHARACTERS NEEDED
  052.356  062 316 077 01403        STA      CSLDLY            SET PAD DELAY
  052.361  072 014 051 01404        LDA      HOSTAB+FLT.CRF
  052.364  052 317 077 01405        LHLD     CSLDCA            (HL) = ADDRESS FOR CHARACTER NEEDING PAD
  052.367  167         01406        MOV      M,A               SET CHARACTER                          /80.06.gc/
  052.370  072 017 051 01407        LDA      HOSTAB+FLT.CDB                                           /80.06.gc/
  052.373  062 343 040 01408        STA      S.CDB             SET CONSOLE DEFINITION BYTE
  052.376  052 020 051 01409        LHLD     HOSTAB+FLT.CBD
  053.001  042 344 040 01410        SHLD     S.BAUD            SET CONSOLE BAUD RATE
  053.004  072 022 051 01411        LDA      HOSTAB+FLT.BOP
  053.007  062 034 041 01412        STA      S.BOOTF           SET UP BOOT FLAGS
  053.012  072 023 051 01413        LDA      HOSTAB+FLT.SAL
  053.015  062 301 077 01414        STA      SALONE            SET UP STAND-ALONE FLAG
  053.020  311         01415        RET
```

```
                     01418   **   SCD - SETUP CONSOLE DRIVER.
                     01419   *
                     01420   *    SCD SETS UP INTERRUPT VECTORS FOR CONSOLE INPUT, AND
                     01421   *    SETS UP THE USART
                     01422
  053.021            01423   SCD  EQU      *
  000.001            01424        IF       DEBUG
                     01428        ENDIF
  053.021  041 130 067 01429        LXI      H,SCINI
  053.024  042 046 040 01430        SHLD     .UIVEC+7  SETUP VECTOR
  053.027  315 036 053 01431        CALL     SCU
  053.032  315 173 053 01432        CALL     ECI
  053.035  311         01433        RET
  053.036              01434        XTEXT    SCU



                     01436X  **   SCU - SETUP CONSOLE USART.
                     01437X  *
                     01438X  *    SCU CONFIGURES THE CONSOLE USART.
                     01439X  *
                     01440X  *    IF  8250
                     01441X  *      THEN   PORT = 372-3Q
                     01442X  *      ELSE   PORT = 340-7Q
                     01443X  *
                     01444X  *
                     01445X  *    ENTRY    NONE
                     01446X  *    EXIT     NONE
                     01447X  *    USES     A,F,(BC),(HL)
                     01448X
                     01449X
  053.036  072 343 040 01450X  SCU  LDA      S.CDB
  053.041  376 001     01451X       CPI      CDB.H84
  053.043  312 106 053 01452X       JZ       SCU1                        IF  8250
                     01453X
                     01454X  *    PRESET 8251
                     01455X
  053.046  076 201     01456X       MVI      A,201Q
  053.050  323 373     01457X       OUT      SC.UART+USR                 GET USART IN KNOWN STATE
  053.052  323 373     01458X       OUT      SC.UART+USR
  053.054  323 373     01459X       OUT      SC.UART+USR
  053.056  323 373     01460X       OUT      SC.UART+USR
  053.060  076 100     01461X       MVI      A,UCI.IR          RESET
  053.062  323 373     01462X       OUT      SC.UART+USR
  053.064  072 327 040 01463X       LDA      S.CONTY
  053.067  346 010     01464X       ANI      CTP.2SB
  000.000              01465X       ERRNZ    CTP.2SB*16+UMI.1B-UMI.2B
  053.071  007         01466X       RLC
  053.072  007         01467X       RLC
  053.073  007         01468X       RLC
  053.074  007         01469X       RLC
  053.075  366 116     01470X       ORI      UMI.1B+UMI.L8+UMI.16X
  053.077  323 373     01471X       OUT      SC.UART+USR
  053.101  076 025     01472X       MVI      A,UCI.ER+UCI.RE+UCI.TE
  053.103  323 373     01473X       OUT      SC.UART+USR
  053.105  311         01474X       RET
```

```
                    01475X
                    01476X  *    IS 8250
                    01477X
   053.106  333 355     01478X  SCU1 IN       SC.ACE+UR.LSR                                          /80.01.GC/
   053.110  346 100     01479X       ANI      UC.TSE            CHECK FOR SHIFT EMPTY                /80.01.GC/
   053.112  312 106 053 01480X       JZ       SCU1
                    01481X
   053.115  257         01482X       XRA      A                                                      /79.01.GC/
   053.116  323 351     01483X       OUT      SC.ACE+UR.IER     TURN OFF ANY INTERRUPTS     /79.01.GC/
   053.120  076 020     01484X       MVI      A,UC.LOO                                      /79.01.GC/
   053.122  323 354     01485X       OUT      SC.ACE+UR.MCR                                          /79.01.GC/
   053.124  052 344 040 01486X       LHLD     S.BAUD
   053.127  076 200     01487X       MVI      A,UC.DLA
   053.131  323 353     01488X       OUT      SC.ACE+UR.LCR     ACCESS DIVISOR LATCHES
   053.133  175         01489X       MOV      A,L
   053.134  323 350     01490X       OUT      SC.ACE+UR.DLL     SET LEAST SIGNIFICANT
   053.136  174         01491X       MOV      A,H
   053.137  346 177     01492X       ANI      177Q              TRIM STOP BITS
   053.141  323 351     01493X       OUT      SC.ACE+UR.DLM     SET MOST SIGNIFICANT
   053.143  072 327 040 01494X       LDA      S.CONTY
   053.146  346 010     01495X       ANI      CTP.2SB
   053.150  017         01496X       RRC
   000.000              01497X       ERRNZ    CTP.2SB/2-UC.2SB
   000.000              01498X       ERRNZ    UC.2SB-4 (A) = UC.2SB IF 2 STOP BITS
   053.151  366 003     01499X       ORI      UC.8BW            8 BIT WORDS
   053.153  323 353     01500X       OUT      SC.ACE+UR.LCR
   053.155  076 156     01501X       MVI      A,AC.DLY                                      /79.01.GC/
   053.157  315 053 000 01502X       CALL     .DLY                                                   /79.01.GC/
   053.162  333 350     01503X       IN       SC.ACE+UR.RBR  GOBBLE ANY TRASH               /79.01.GC/
   053.164  333 354     01504X       IN       SC.ACE+UR.MCR                                          /79.01.GC/
   053.166  346 357     01505X       ANI      377Q-UC.LOO                                            /79.01.GC/
   053.170  323 354     01506X       OUT      SC.ACE+UR.MCR                                          /79.01.GC/
   053.172  311         01507X       RET
   053.173              01508        XTEXT    ECI



                    01510X  **   ECI - ENABLE CONSOLE INTERRUPTS
                    01511X  *
                    01512X  *    ENTRY    NONE
                    01513X  *    EXIT     NONE
                    01514X  *    USES     (PSW)
                    01515X  *
                    01516X
   053.173  072 343 040 01517X  ECI  LDA      S.CDB
   053.176  376 001     01518X       CPI      CDB.H84
   053.200  312 210 053 01519X       JZ       ECI1                        IF  8250
                    01520X
                    01521X  *    HAVE 8251
                    01522X
   053.203  076 027     01523X       MVI      A,UCI.RE+UCI.TE+UCI.ER+UCI.IE
   053.205  323 373     01524X       OUT      SC.UART+USR
   053.207  311         01525X       RET
                    01526X
                    01527X  *    HAVE 8250
                    01528X
```

```
   053.210  076 001     01529X  ECI1 MVI      A,UC.EDA
   053.212  323 351     01530X       OUT      SC.ACE+UR.IER
   053.214  311         01531X       RET
```

```
                     01534   **   GVM - GIVE VERSION MESSAGE.
                     01535   *
                     01536   *    ENTRY    NONE
                     01537   *    EXIT     NONE
                     01538   *    USES     ALL
                     01539
                     01540
  053.215  315 173 053 01541   GVM  CALL     ECI
  053.220  315 136 031 01542        CALL     $TYPTX
  053.223  012 110 104 01543        DB       NL,'HDOS Version '
                     01544   *    DB       VERS/16+'0','.',VERS&00001111B+'0'            /13.09.GFR/
  053.241  062 056 060 01545        DB       '2.0'    hard wired due to ASM limitations    /13.09.GFR/
  053.244  012 040 111 01546        DB       NL,' Issue # 50.06.00',ENL
  053.267  311         01547        RET
```

```
                     01550   **   FSM      - Fake System Mount                                  /80.05.gc/
                     01551   *
                     01552   *    FSM reads in just enough of the system paramaters
                     01553   *    for LSO to locate the overlays.  Then the overlays
                     01554   *    can really mount the diskettes once the drivers
                     01555   *    have also been found.
                     01556   *
                     01557
  001.051            01558        ERRMI    *-LABELE Label buffer will overlay   /80.05.gc/
                     01559
  053.270  076 007   01560   FSM  MVI      A,DC.ABT
  053.272  315 130 040 01561        CALL     SYDD               ABORT DRIVER
                     01562
                     01563   *    Fetch the Label ( and consequently the parameters )
                     01564
  053.275  001 000 001 01565        LXI      B,256
  053.300  021 217 051 01566        LXI      D,LABEL
  053.303  041 011 000 01567        LXI      H,DDF.LAB
  053.306  076 002     01568        MVI      A,DC.RER READ REGARDLESS
  053.310  315 130 040 01569        CALL     SYDD
  053.313  334 111 057 01570        CC       BOOTERR            BAD ERROR
                     01571
                     01572   *    Mount the device so the volume parameters are set-up
                     01573
  053.316  072 217 051 01574        LDA      LABEL+LAB.SER
  053.321  157         01575        MOV      L,A
  053.322  046 000     01576        MVI      H,0                HL = serial number
  053.324  076 010     01577        MVI      A,DC.MOU
  053.326  315 130 040 01578        CALL     SYDD               Mount the unit
  053.331  334 111 057 01579        CC       BOOTERR            Bad error
                     01580
                     01581   *    Save the parameters for others to find
                     01582
  053.334  052 224 051 01583        LHLD     LABEL+LAB.GRT
  053.337  042 044 041 01584        SHLD     AIO.GRT            GRT address
                     01585
  053.342  072 226 051 01586        LDA      LABEL+LAB.SPG
  053.345  042 046 041 01587        SHLD     AIO.SPG            Sectors/Group
                     01588
  053.350  076 001     01589        MVI      A,1
  053.352  062 032 041 01590        STA      S.MOUNT            FLAG SYSTEM MOUNTED
  053.355  311         01591        RET
```

```
                     01594   **   LSO - LOCATE SYSTEM OVERLAY.
                     01595   *
                     01596   *    LSO LOCATES THE SYSTEM OVERLAYS:
                     01597   *             *HDOSOVL0.SYS*
                     01598   *             *HDOSOVL1.SYS*
                     01599   *
                     01600   *    AND SETS UP POINTERS AND OTHER TABLE DATA TO BOTH.
                     01601   *
                     01602   *
                     01603   *    IT IS READ, AND THE INFO USED TO SETUP THE CELLS
                     01604   *
                     01605   *    S.OMAX             SYSTEM OVERLAY MAX
                     01606   *    S.SSN              SWAP SECTOR NUMBER
                     01607   *    S.OSN              OVERLAY SECTOR NUMBER
                     01608   *    S.OVLS             OVERLAY SIZE
                     01609   *
                     01610   *    ENTRY    NONE
                     01611   *    EXIT     NONE
                     01612   *    USES     ALL
                     01613
  000.137            01614        ERRMI    *-BUFFE            BUFF will overlay                   /80.05.gc/
                     01615
  053.356  021 332 054 01616   LSO  LXI      D,LSOA
  053.361  315 071 054 01617        CALL     LSO.               (HL) = SECTOR NUMBER
  053.364  042 002 041 01618        SHLD     S.SSN              SET SWAP NUMBER
  053.367  021 014 000 01619        LXI      D,SB.OVMX/256
  053.372  031         01620        DAD      D                  (HL) = SECTOR FOR CODE
  053.373  042 036 076 01621        SHLD     OVL0*OVL.ENS+OVLTAB+OVL.COD
  053.376  315 306 054 01622        CALL     LSO..              (HL) = LENGTH
  054.001  042 324 040 01623        SHLD     S.OMAX             SET OVERLAY MAXIMUM SIZE
  054.004  042 040 076 01624        SHLD     OVL0*OVL.ENS+OVLTAB+OVL.SIZ
                     01625
                     01626   *    SET UP *HDOSOVL2.SYS*
                     01627
  054.007  021 347 054 01628        LXI      D,LSOB
  054.012  315 071 054 01629        CALL     LSO.               (HL) = SECTOR NUMBER FOR CODE
  054.015  042 046 076 01630        SHLD     OVL1*OVL.ENS+OVLTAB+OVL.COD
  054.020  315 306 054 01631        CALL     LSO..              (HL) = LENGTH OF OVERLAY
  054.023  042 050 076 01632        SHLD     OVL1*OVL.ENS+OVLTAB+OVL.SIZ
  054.026  353         01633        XCHG
  054.027  052 324 040 01634        LHLD     S.OMAX
  054.032  315 053 075 01635        CALL     HLCPDE
  054.035  320         01636        RNC                         S.OMAX >= SIZE OF THIS OVERLAY
  054.036  315 136 031 01637        CALL     $TYPTX
  054.041  012 077 060 01638        DB       NL,'?01 Overlay too big',ENL
  054.066  303 111 057 01639        JMP      BOOTERR
                     01640
  054.071  325         01641   LSO. PUSH     D                  SAVE FILE NAME POINTER
  054.072  001 015 000 01642        LXI      B,DIRIDL  (BC) = COUNT
  054.075  041 062 041 01643        LXI      H,AIO.DIR+DIR.NAM
  054.100  315 252 030 01644        CALL     $MOVE              MOVE IN NAME PATTERN
  054.103  001 015 000 01645        LXI      B,DIRIDL  (BC) = MATCH LENGTH
  054.106  052 222 051 01646        LHLD     LABEL+LAB.DIS      (HL) = DIRECTORY SECTOR FWA
  054.111  315 372 057 01647        CALL     LDE..              LOCATE DIRECTORY ENTRY
  054.114  322 164 054 01648        JNC      LSO1               GOTIT
                     01649
                     01650   *    MISSING OVERLAY FILE
```

```
                        01651
   054.117  315 136 031 01652         CALL      $TYPTX
   054.122  012 077 060 01653         DB        NL,'?01 Missing File',' '+200Q
   054.144  321         01654         POP       D                 RESTORE FILE NAME POINTER
   054.145  001 015 000 01655         LXI       B,DIRIDL SET UP COUNT
   054.150  041 062 041 01656         LXI       H,AIO.DIR+DIR.NAM  SET UP DESTINATION FOR FILE NAME
   054.153  315 252 030 01657         CALL      $MOVE             MOVE IN NAME PATTERN
   054.156  315 056 064 01658         CALL      $TFN              TYPE FILE NAME
   054.161  303 111 057 01659         JMP       BOOTERR           ABORT BOOT
                        01660
                        01661    *    FOUND OVERLAY
                        01662
   054.164  321         01663    LSO1 POP       D                 DISCARD FILE NAME POINTER SINCE IT IS FOUND
   054.165  021 016 000 01664         LXI       D,DIR.FLG
   054.170  031         01665         DAD       D
   054.171  176         01666         MOV       A,M               (A) = FLAG BYTE
   054.172  346 020     01667         ANI       DIF.CNT
   054.174  312 213 054 01668         JZ        LSO2              NOT CONTIGUOUS
   000.000              01669         ERRNZ     DIR.FGN-DIR.FLG-2
   054.177  043         01670         INX       H
   054.200  043         01671         INX       H                 (HL0 = #DIR.FGN
   054.201  136         01672         MOV       E,M
   054.202  026 000     01673         MVI       D,0               (DE) = FILE FIRST GROUP NUMBER
   054.204  072 226 051 01674         LDA       LABEL+LAB.SPG
   054.207  315 007 031 01675         CALL      $MU86             (HL) = SECTOR NUMBER
   054.212  311         01676         RET
                        01677
                        01678    *    OVERLAY IS NOT CONTIGUOUS
                        01679
   054.213  315 136 031 01680    LSO2 CALL      $TYPTX
   054.216  012 077 060 01681         DB      NL,'?01 System Not SYSGENed Properly, or Files Damaged.',ENL
   054.303  303 111 057 01682         JMP     BOOTERR
                        01683
   054.306  001 000 001 01684    LSO..          LXI       B,256
   054.311  021 217 052 01685         LXI       D,BUFF
   054.314  315 241 031 01686         CALL      $WER              WRITE ENABLE RAM AREA
   054.317  315 275 031 01687         CALL      S.READ                      READ FROM DISK
   054.322  052 221 052 01688         LHLD      BUFF+PIC.LEN
   054.325  001 010 000 01689         LXI       B,8
   054.330  011         01690         DAD       B
   054.331  311         01691         RET
                        01692
   054.332  110 104 117 01693    LSOA DB        'HDOSOVL0','SYS',0,0        OVERLAY FILE NAME
   000.000              01694         ERRNZ     *-LSOA-DIRIDL     LSOA IS ENTIRE SPECIFICATION
   054.347  110 104 117 01695    LSOB DB        'HDOSOVL1','SYS',0,0
   000.000              01696         ERRNZ     *-LSOB-DIRIDL
```

```
                     01699   **   SDT - SETUP DEVICE TABLE.
                     01700   *
                     01701   *    SDT SCANS THE SYSTEM DISK DIRECTORY LOOKING FOR FILES IN
                     01702   *    THE FORM:
                     01703   *
                     01704   *    XX       .DVD
                     01705   *
                     01706   *    THESE ENTRYS ARE BUILT INTO THE DEVICE TABLE
                     01707   *
                     01708
  010.017            01709        ERRMI    BUFF-42200A       Need to write enable buffer /80.06.gc/
  001.145            01710        ERRMI    *-BUFFE           BUFF will overlay           /80.05.gc/
  026.364            01711        ERRMI    *-SDTAE           SDT will overlay            /80.05.gc/
                     01712
  054.364            01713   SDT  EQU      *
  054.364  052 222 051 01714        LHLD     LABEL+LAB.DIS
  054.367  042 376 025 01715        SHLD     SDTA+DIS.LNK      SET SECTOR NUMBER TO READ
                     01716
                     01717   *    READ NEXT SECTOR
                     01718
  054.372  052 376 025 01719   SDT1 LHLD     SDTA+DIS.LNK
  054.375  174         01720        MOV      A,H
  054.376  265         01721        ORA      L
  054.377  310         01722        RZ                         NO MORE DIRECTORY, AM DONE
  055.000  021 000 024 01723        LXI      D,SDTA
  055.003  001 000 002 01724        LXI      B,512
  055.006  315 241 031 01725        CALL     $WER              WRITE ENABLE RAM
  055.011  315 275 031 01726        CALL     S.READ            READ DIRECTORY
                     01727
                     01728   *    RUN DOWN THROUGH ENTRYS LOOKING FOR XX.DVD
                     01729
  055.014  041 000 024 01730        LXI      H,SDTA
  055.017  176         01731   SDT2 MOV      A,M
  055.020  247         01732        ANA      A
  055.021  312 372 054 01733        JZ       SDT1              END OF SECTOR
  000.000              01734        ERRNZ    DF.EMP-377Q
  055.024  074         01735        INR      A
  055.025  312 064 055 01736        JZ       SDT4              ENTRY IS EMPTY
  000.000              01737        ERRNZ    DF.CLR-376Q
  055.030  074         01738        INR      A
  055.031  310         01739        RZ                         NO MORE IN DIRECTORY
  055.032  345         01740        PUSH     H
  055.033  043         01741        INX      H
  055.034  176         01742        MOV      A,M
  055.035  247         01743        ANA      A
  055.036  312 063 055 01744        JZ       SDT3              IS ONE-CHARACTER NAME
  055.041  043         01745        INX      H
  055.042  021 075 055 01746        LXI      D,SDTB
  055.045  001 013 000 01747        LXI      B,SDTBL
  055.050  315 060 030 01748        CALL     $COMP             COMPARE
  055.053  302 063 055 01749        JNE      SDT3              NOT MATCH
                     01750
                     01751   *    GOT ONE
                     01752
  055.056  341         01753        POP      H
  055.057  345         01754        PUSH     H                 (HL) = ENTRY FWA
  055.060  315 110 055 01755        CALL     EDL               ENTER DRIVER IN LIST
```

```
                     01756
                     01757   *    TRY ANOTHER ENTRY
                     01758
   055.063  341      01759   SDT3 POP      H                  (HL) = ENTRY FWA
   055.064  072 373 025 01760   SDT4 LDA      SDTA+DIS.ENL
   055.067  315 101 030 01761        CALL     $DADA.             ADVANCE
   055.072  303 017 055 01762        JMP      SDT2               TRY NEXT
                     01763
   055.075  000 000 000 01764   SDTB DB       0,0,0,0,0,0,'DVD',0,0               REQUIRED EXTENSION
   000.013              01765   SDTBL        EQU      *-SDTB             LENGTH OF PATTERN



                     01767   **   EDL - ENTER DEVICE IN DEVICE LIST.
                     01768   *
                     01769   *    EDL ENTERS DEVICE DRIVER INFORMATION INTO THE
                     01770   *    DEVLST.
                     01771   *
                     01772   *    THE FILE IS READ TO SETUP THE DEVICE TABLE ENTRY.
                     01773   *
                     01774   *    ENTRY    (HL) = FWA DIRECTORY ENTRY FOR DRIVER
                     01775   *    EXIT     DRIVER IN DEVLST IF ALL OK
                     01776   *             DRIVER IGNORED IF PROBLEMS
                     01777   *    USES     ALL
                     01778
                     01779
   055.110  136      01780   EDL  MOV      E,M
   055.111  043      01781        INX      H
   055.112  126      01782        MOV      D,M                (DE) = NAME
   055.113  353      01783        XCHG
   055.114  042 146 056 01784        SHLD     EDLNAM             SET NAME FIELD IN DEVLST ENTRY
   055.117  042 137 056 01785        SHLD     EDLD               SET NAME FOR MESSAGE                /80.05.gc/
                     01786
                     01787   *    SETUP SECTOR ADDRESS FOR DRIVER
                     01788
   055.122  041 017 000 01789        LXI      H,DIR.FGN-1
   055.125  031      01790        DAD      D                  (HL) = #DIR.FGN
   055.126  176      01791        MOV      A,M                (A) = FIRST GROUP
   055.127  062 163 056 01792        STA      EDLDVG             SET DRIVER FIRST GROUP
                     01793
                     01794   *    READ FIRST SECTOR OF DRIVER FILE
                     01795
   055.132  137      01796        MOV      E,A
   055.133  026 000   01797        MVI      D,0                (DE) = GROUP
   055.135  072 226 051 01798        LDA      LABEL+LAB.SPG      A  = Sectors/Group          /80.05.gc/
   055.140  315 007 031 01799        CALL     $MU86              (HL) = SECTOR ADDRESS OF 1ST GROUP
   055.143  021 217 052 01800        LXI      D,BUFF
   055.146  001 000 001 01801        LXI      B,256
   055.151  315 275 031 01802        CALL     S.READ             READ IT                             /80.06.gc/
                     01803
                     01804   *    SEE IF PIC FILE
                     01805
   055.154  052 217 052 01806        LHLD     BUFF
   055.157  054      01807        INR      L
   055.160  302 063 056 01808        JNZ      EDL5               NOT BINARY
   055.163  076 001   01809        MVI      A,FT.PIC
```

```
   055.165  274         01810        CMP      H
   055.166  302 063 056 01811        JNE      EDL5              NOT PIC
                        01812
                        01813   *    SET DEVICE CAPABILITY BYTE
                        01814
   055.171  072 225 052 01815        LDA      BUFF+DVD.DVD      (A) = DRIVER FLAG
   055.174  376 307     01816        CPI      DVDFLV            SEE IF DRIVER
   055.176  302 063 056 01817        JNE      EDL5              NOT DRIVER
   055.201  072 226 052 01818        LDA      BUFF+DVD.CAP
   055.204  062 154 056 01819        STA      EDLCAP            SET DEVICE CAPABILITY FLAGS
   055.207  072 227 052 01820        LDA      BUFF+DVD.MUM
   055.212  062 155 056 01821        STA      EDLMUM            SET UP MOUNTED UNITS MASK
   055.215  072 230 052 01822        LDA      BUFF+DVD.MNU
   055.220  062 156 056 01823        STA      EDLMNU            SET MAXIMUM NUMBER OF UNITS
                        01824
                        01825   *    ALLOCATE UNIT DESCRIPTOR TABLES
                        01826
   055.223  072 230 052 01827        LDA      BUFF+DVD.MNU      A  = MAX. NUMBER OF UNITS
   055.226  021 010 000 01828        LXI      D,UNT.SIZ
   055.231  315 007 031 01829        CALL     $MU86             HL = MEMORY TO ALLOCATE
   055.234  315 224 030 01830        CALL     $CHL
   055.237  353         01831        XCHG
   055.240  052 356 040 01832        LHLD     S.RFWA
   055.243  031         01833        DAD      D                 HL = NEW FWA
                        01834
   055.244  042 356 040 01835        SHLD     S.RFWA
   055.247  042 320 040 01836        SHLD     S.SYSM
   055.252  042 157 056 01837        SHLD     EDLPTR
                        01838
                        01839   *    INITIALIZE THE UNIT DESCRIPTOR TABLE
                        01840
   055.255  072 226 052 01841        LDA      BUFF+DVD.CAP
   055.260  107         01842        MOV      B,A               B  = DEVICE CAPABILITY FLAGS
   055.261  021 231 052 01843        LXI      D,BUFF+DVD.UFL
   055.264  072 230 052 01844        LDA      BUFF+DVD.MNU
                        01845
   055.267  075         01846   EDL0 DCR      A
   055.270  372 316 055 01847        JM       EDL0.5            FINISHED WITH THE UNITS
                        01848
   055.273  365         01849        PUSH     PSW
   055.274  032         01850        LDAX     D                 A  = FLAG VALUE FOR THIS UNIT
   055.275  240         01851        ANA      B                 MAP OUT ILLEGAL BITS
   055.276  315 143 075 01852        CALL     $INDSB
   055.301  000 000     01853        DW       UNT.FLG
   055.303  325         01854        PUSH     D
   055.304  021 010 000 01855        LXI      D,UNT.SIZ
   055.307  031         01856        DAD      D                 HL = NEXT UNIT DESCRIPTOR
   055.310  321         01857        POP      D
   055.311  023         01858        INX      D                 MOVE TO NEXT UNIT
   055.312  361         01859        POP      PSW
   055.313  303 267 055 01860        JMP      EDL0
                        01861
   055.316              01862   EDL0.5       EQU      *
                        01863
                        01864   *    SET LENGTH
                        01865
   055.316  052 223 052 01866        LHLD     BUFF+PIC.PTR      (HL) = CODE LENGTH
```

```
   055.321  001 000 376 01867        LXI      B,-DVD.ENT
   055.324  011         01868        DAD      B                 (HL = LEN OF DRIVER CODE
   055.325  322 063 056 01869        JNC      EDL5              TOO SMALL
   055.330  042 161 056 01870        SHLD     EDLDVL            SET DRIVER LENGTH
                        01871
                        01872   *    HAVE BUILT ENTRY FOR DEVLST. INSERT
                        01873
   055.333  052 354 040 01874        LHLD     S.DFWA
   055.336  006 006     01875        MVI      B,DEVCNT-1        (B) = MAX DRIVER COUNT
   000.005              01876        ERRMI    DEVCNT-2          REQUIRE 2
   055.340  021 016 000 01877        LXI      D,DEVELEN
                        01878
   055.343  031         01879   EDL1 DAD      D                 (HL) = ADDRESS OF NEXT ENTRY
   055.344  176         01880        MOV      A,M
   055.345  267         01881        ORA      A
   000.000              01882        ERRNZ    DV.EL             DEVICE END OF LIST FLAG
   055.346  312 047 056 01883        JZ       EDL3              GOT ONE
   055.351  005         01884        DCR      B
   055.352  302 343 055 01885        JNZ      EDL1              TRY NEXT
                        01886
                        01887   *    NO ROOM FOR IT.
                        01888
   055.355  315 136 031 01889        CALL     $TYPTX
   055.360  012 007 077 01890        DB       NL,BELL,'?01 Too Many Device Drivers.',' '+200Q
   056.017  041 133 056 01891   EDL2 LXI      H,EDLB            TYPE NAME
   056.022  076 012     01892        MVI      A,10
   056.024  315 226 064 01893        CALL     $TYPCC            TYPE NAME
   056.027  315 136 031 01894        CALL     $TYPTX
   056.032  040 055 040 01895        DB       ' - Ignored.',ENL
   056.046  311         01896        RET
                        01897
                        01898   *    GOT SPOT. PUT IT IN
                        01899
   056.047  021 146 056 01900   EDL3 LXI      D,EDLDEV
   056.052  001 016 000 01901        LXI      B,DEVELEN
   056.055  315 252 030 01902        CALL     $MOVE             COPY INTO TABLE
   056.060  066 000     01903        MVI      M,0               CLEAR NEXT ENTRY
   056.062  311         01904        RET                        RETURN
                        01905
                        01906   *    ERROR IN DRIVER FORMAT
                        01907
   056.063  315 136 031 01908   EDL5 CALL     $TYPTX
   056.066  012 007 077 01909        DB       NL,BELL,'?01 Format Error in Driver File',' '+200Q
   056.130  303 017 056 01910        JMP      EDL2
                        01911
   056.133  123 131     01912   EDLB DB       'SY'              System Device                        /80.05.gc/
   056.135  060 072     01913   EDLC DB       '0:'              Unit Number                          /80.05.gc/
   056.137  130 130     01914   EDLD DB       'XX'              Device Name                          /80.05.gc/
   056.141  056 104 126 01915        DB       '.DVD',0
                        01916
   056.146              01917   EDLDEV       EQU      *
   000.000              01918        ERRNZ    *-EDLDEV-DEV.NAM
   056.146  040 040     01919   EDLNAM       DB       '  '               DEVICE NAME
   000.000              01920        ERRNZ    *-EDLDEV-DEV.RES
   056.150  000         01921        DB       0                 NOT RESIDENT
   000.000              01922        ERRNZ    *-EDLDEV-DEV.JMP
   056.151  303         01923        DB       303Q              JUMP OPCODE
```

```
   000.000             01924        ERRNZ    *-EDLDEV-DEV.DDA
   056.152  373 074    01925        DW       SDD               DRIVER ADDRESS (STAND-IN DEVICE DRIVER)
   000.000             01926        ERRNZ    *-EDLDEV-DEV.FLG
   056.154  000        01927   EDLCAP        DB       0                FLAGS
   000.000             01928        ERRNZ    *-EDLDEV-DEV.MUM
   056.155  000        01929   EDLMUM        DB       0                MOUNTED UNIT MASK
   000.000             01930        ERRNZ    *-EDLDEV-DEV.MNU
   056.156  001        01931   EDLMNU        DB       1                MAXIMUM NUMBER OF UNITS
   000.000             01932        ERRNZ    *-EDLDEV-DEV.UNT
   056.157  000 000    01933   EDLPTR        DW       0                UNIT POINTER
   000.000             01934        ERRNZ    *-EDLDEV-DEV.DVL
   056.161  000 000    01935   EDLDVL        DW       0                DRIVER LENGTH
   000.000             01936        ERRNZ    *-EDLDEV-DEV.DVG
   056.163  000        01937   EDLDVG        DB       0                DRIVER SECTOR FIRST GROUP NUMBER
   000.000             01938        ERRNZ    *-EDLDEV-DEVELEN
```

```
                     01941   **   MSD      - Mount System Diskette                                /80.05.gc/
                     01942   *
                     01943   *    MSD mounts the system diskette.  It invokes the overlays
                     01944   *    hence, requires that they already be found via FSM and
                     01945   *    LSO.
                     01946   *
                     01947
  056.164  315 041 064 01948   MSD  CALL     $CRLF              for aesthetics
                     01949
                     01950   *    Find the device table entry for SY:
                     01951
  056.167  001 131 123 01952        LXI      B,'SY'
  056.172  052 354 040 01953        LHLD     S.DFWA             HL = address of device table
                     01954
  056.175  176         01955   MSD1 MOV      A,M
  056.176  247         01956        ANA      A
  000.000              01957        ERRNZ    DV.EL
  056.177  312 034 057 01958        JZ       MSD5               At the end of the list without finding SY:
                     01959
  056.202  270         01960        CMP      B
  056.203  043         01961        INX      H
  056.204  302 223 056 01962        JNZ      MSD2               Is not SY:
  056.207  176         01963        MOV      A,M
  056.210  271         01964        CMP      C
  056.211  302 223 056 01965        JNZ      MSD2               Is not SY:
                     01966
                     01967   *    SY: is found
                     01968
  056.214  053         01969        DCX      H
  056.215  042 377 073 01970        SHLD     GSPA               Save pointer to SY: entry
  056.220  303 232 056 01971        JMP      MSD3
                     01972
                     01973   *    SY is NOT found
                     01974
  056.223  021 015 000 01975   MSD2 LXI      D,DEVELEN-1
  056.226  031         01976        DAD      D
  056.227  303 175 056 01977        JMP      MSD1               Try the next entry
                     01978
  056.232  315 063 060 01979   MSD3 CALL     LSD                Load the SYstem Device Driver
  056.235  332 034 057 01980        JC       MSD5               Error
  056.240  072 217 051 01981        LDA      LABEL+LAB.SER
  056.243  157         01982        MOV      L,A
  056.244  046 000     01983        MVI      H,0                HL = Volume Number
  056.246  076 010     01984        MVI      A,DC.MOU
  056.250  315 130 040 01985        CALL     SYDD               Set up the volume parameters for read
  056.253  332 034 057 01986        JC       MSD5
                     01987
                     01988   *    Mount the volume
                     01989
  056.256  041 104 057 01990        LXI      H,MSDA
  056.261  377 200     01991        SCALL    .MOUNT             Mount the volume
                     01992
  056.263  365         01993        PUSH     PSW
  056.264  315 041 064 01994        CALL     $CRLF              for aesthetics
  056.267  361         01995        POP      PSW
  056.270  320         01996        RNC                         No ERROR in mount
                     01997
```

```
                        01998   *    An error in mount
                        01999
   056.271  376 047     02000        CPI       EC.DSC
   056.273  302 034 057 02001        JNZ       MSD5
   056.276  315 136 031 02002        CALL      $TYPTX
   056.301  012 007     02003        DB        NL,BELL
   056.303  077 060 061 02004        DB        '?01 Disk Structure is Corrupt.',NL
   056.342  103 157 156 02005        DB        'Contact Heath Technical Correspondence for Assistance.',ENL
   057.031  303 156 057 02006        JMP       BOOTABT
                        02007
                        02008   *    Bad ERROR
                        02009
   057.034  315 136 031 02010   MSD5 CALL      $TYPTX
   057.037  012 077 060 02011        DB        NL,'?01 Unable To Mount System Disk.',ENL
   057.101  303 156 057 02012        JMP       BOOTABT
                        02013
   057.104  123 131     02014   MSDA DB        'SY'              Device specification
   057.106  060 072 000 02015        DB        '0:',0            Unit specification




                        02017   **   BOOTERR - ERROR DURING BOOT.
                        02018   *
   057.111  315 136 031 02019   BOOTERR CALL  $TYPTX
   057.114  012 007 077 02020        DB        NL,BELL,'?01 Disk I/O Error During Boot.',ENL
   000.000              02021        ERRNZ     *-BOOTABT



                        02023   **   BOOTABT - ABORT BOOT.
                        02024   *
                        02025   *
                        02026
   057.156  315 136 031 02027   BOOTABT CALL  $TYPTX
   057.161  040 040 102 02028        DB        '  Boot Aborted. Will Restart ..','.'+200Q
   057.221  303 000 030 02029        JMP       30000A
```

```
                     02032   **   AGT      - Allocate GRT Table                                  /80.04.gc/
                     02033   *
                     02034   *    AGT allocates enough space for each of the GRT's
                     02035   *    required by a device driver.
                     02036   *
                     02037   *    ********
                     02038   *    * Note *
                     02039   *    ********
                     02040   *
                     02041   *             This problem should be addressed more aggresively
                     02042   *             in that GRT's should not necessarily be on page
                     02043   *             boudaries, however, this would necessitate replacing
                     02044   *             much code and sacrificing the efficienty of the
                     02045   *             allocation algorithem.  For now, we will waste
                     02046   *             the extra memory, however, at some time this
                     02047   *             restriction should be removed.
                     02048   *                                          G. CHandler 80.04.29
                     02049   *
                     02050   *    ENTY:    AIO.DATA initialized
                     02051   *             DE       = device driver length
                     02052   *             HL       = DEV.DVG for this device
                     02053   *
                     02054   *    EXIT:    HL       = device load address
                     02055   *             PSW      = 'C' clear if no errors
                     02056   *                        unit GRT pointers initalized
                     02057   *                      = 'C' set   if not enough room
                     02058   *
                     02059   *    USES:    PSW, HL
                     02060   *
                     02061
057.224  325         02062   AGT  PUSH     D                 save driver length
057.225  315 066 075 02063        CALL     $INDLB
057.230  371 377     02064        DW       DEV.FLG-DEV.DVG
057.232  346 001     02065        ANI      DT.DD
057.234  062 357 057 02066        STA      AGTA              save directory device flag
057.237  312 254 057 02067        JZ       AGT1              Not a directory device
                     02068
                     02069   *    Compute total load length
                     02070
057.242  315 066 075 02071        CALL     $INDLB
057.245  373 377     02072        DW       DEV.MNU-DEV.DVG
057.247  062 360 057 02073        STA      AGTB              save the maximum number of units
057.252  202         02074        ADD      D                 add the tables to the end of the driver
057.253  127         02075        MOV      D,A
                     02076
                     02077   *    Compute load address
057.254  052 320 040 02078   AGT1 LHLD     S.SYSM
                     02079
057.257  175         02080        MOV      A,L
057.260  223         02081        SUB      E
057.261  157         02082        MOV      L,A
057.262  174         02083        MOV      A,H
057.263  232         02084        SBB      D
057.264  147         02085        MOV      H,A               HL = HL - DE
                     02086
057.265  321         02087        POP      D
057.266  330         02088        RC                         error
```

```
                    02089
   057.267  072 357 057 02090        LDA      AGTA
   057.272  247         02091        ANA      A
   057.273  310         02092        RZ                        Not a directory device
                    02093
                    02094   *    Kludge the load address to force GRT's on page boundaries
                    02095
   057.274  173         02096        MOV      A,E
   057.275  205         02097        ADD      L
   057.276  057         02098        CMA      A
   057.277  305         02099        PUSH     B
   057.300  006 377     02100        MVI      B,377Q           sign extend offset !
   057.302  117         02101        MOV      C,A
   057.303  011         02102        DAD      B                Adjust HL to force GRT's on page boundaries
   057.304  301         02103        POP      B
   057.305  043         02104        INX      H
                    02105
   057.306  072 360 057 02106        LDA      AGTB             A  = maximum number of units
   057.311  247         02107        ANA      A
   057.312  310         02108        RZ                        No units for some reason?
                    02109
                    02110   *    Initialize the unit pointers
                    02111
   057.313  325         02112        PUSH     D                save load length
   057.314  345         02113        PUSH     H                save load address
   057.315  031         02114        DAD      D
   057.316  353         02115        XCHG                      DE = address of first GRT to allocate
                    02116
   057.317  325         02117        PUSH     D
   057.320  052 053 041 02118        LHLD     AIO.DTA
   057.323  021 011 000 02119        LXI      D,DEV.UNT
   057.326  031         02120        DAD      D                HL = address of UNIT table pointer
   057.327  321         02121        POP      D
                    02122
   057.330  075         02123   AGT2 DCR      A
   057.331  372 354 057 02124        JM       AGT3             all finished
                    02125
   057.334  365         02126        PUSH     PSW
   057.335  345         02127        PUSH     H
   057.336  315 027 041 02128        CALL     S.GUP            HL = unit table pointer
   057.341  315 107 075 02129        CALL     $INDS
   057.344  002 000     02130        DW       UNT.GRT          initialize the GRT pointer
   057.346  341         02131        POP      H
   057.347  361         02132        POP      PSW
                    02133
   057.350  024         02134        INR      D                advance to the next GRT pointer
   057.351  303 330 057 02135        JMP      AGT2
                    02136
   057.354  341         02137   AGT3 POP      H                restore load address
   057.355  321         02138        POP      D                restore load length
   057.356  311         02139        RET
                    02140
   057.357  000         02141   AGTA DB       0                Directory Device flag
   057.360  000         02142   AGTB DB       0                Maximum Number of Units
```

```
                     02144   **   LDE - LOCATE DIRECTORY ENTRY.
                     02145   *
                     02146   *    LDE LOCATES A DIRECTORY ENTRY CORRESPONDING TO THE AIO.DIR ENTRY.
                     02147   *
                     02148   *    ENTRY    (BC) = NUMBER OF CHARACTERS TO MATCH ON
                     02149   *    EXIT     'C' CLEAR IF FOUND
                     02150   *             AIO.DES SETUP
                     02151   *             (HL) = ADDRESS OF DIRECTORY ENTRY IN SECSCR
                     02152   *             'C' SET IF NOT FOUND
                     02153   *             (A) = CODE
                     02154   *    USES     ALL
                     02155
                     02156
  057.361  001 015 000 02157   LDE. LXI      B,DIRIDL ENTRY FOR FULL NAME COMPARE
                     02158
  057.364  052 222 051 02159   LDE  LHLD     LABEL+LAB.DIS     HL = Directory Sector               /80.05.gc/
  057.367  042 055 041 02160        SHLD     AIO.DES                                              /80.05.gc/
                     02161
                     02162   **   ENTRY FOR (HL) = SECTOR NUMBER TO START WITH
                     02163
  057.372  305         02164   LDE..         PUSH     B                 SAVE COUNT
  057.373  001 000 002 02165        LXI      B,512
  057.376  021 126 100 02166        LXI      D,SECSCR
  060.001  042 055 041 02167        SHLD     AIO.DES           Assume will find in this block     /80.05.gc/
  060.004  315 275 031 02168        CALL     S.READ            READ (know is system device)       /80.05.gc/
  060.007  301         02169        POP      B                 RESTORE (BC)
                     02170
                     02171   *    SCAN SECTOR FOR INFO
                     02172
  060.010  041 126 100 02173        LXI      H,DIS.ENT+SECSCR
                     02174
                     02175   *    COMPARE
                     02176
  060.013  021 062 041 02177   LDE3 LXI      D,AIO.DIR+DIR.NAM
  060.016  176         02178        MOV      A,M
  060.017  247         02179        ANA      A
  060.020  372 033 060 02180        JM       LDE3.5            NO ENTRY
  060.023  305         02181        PUSH     B                 SAVE COPY OF (BC)
  060.024  345         02182        PUSH     H                 SAVE ADDRESS
  060.025  315 060 030 02183        CALL     $COMP             COMPARE
  060.030  341         02184        POP      H
  060.031  301         02185        POP      B                 (BC) = COMPARE COUNT
  060.032  310         02186        RE                         GOT MATCH
                     02187
  060.033  021 027 000 02188   LDE3.5       LXI      D,DIRELENMISSED, SCAN TO NEXT ENTRY
  060.036  031         02189        DAD      D
  060.037  176         02190        MOV      A,M
  060.040  247         02191        ANA      A
  060.041  302 013 060 02192        JNZ      LDE3              MORE IN SECTOR
                     02193
                     02194   *    DIDNT FIND IT IN THIS SECTOR, TRY NEXT
                     02195
  060.044  052 124 102 02196        LHLD     DIS.LNK+SECSCR
  060.047  042 055 041 02197        SHLD     AIO.DES           SET POSSIBLE SECTOR INDEX   /80.05.gc/
  060.052  174         02198        MOV      A,H
  060.053  265         02199        ORA      L
  060.054  302 372 057 02200        JNZ      LDE..             HAVE MORE SECTORS
```

```
060.057  076 014     02201        MVI      A,EC.FNF FILE NOT FOUND
060.061  067         02202        STC
060.062  311         02203        RET



                     02205   **   LSD      - Load System Device
                     02206   *
                     02207   *    LSD loads the system device driver.  Once again, this
                     02208   *    is somewhat of a kludge since the system device is not
                     02209   *    mounted yet, but ...
                     02210   *
                     02211   *    EXIT:    PSW      = 'C' Clear if NO Error
                     02212   *                         'C' Set   if    Error
                     02213   *
                     02214
060.063              02215   LSD  EQU      *
                     02216
                     02217   *    Kludge GRT Pointers for initial load of SY.DVD
                     02218
060.063  041 000 052 02219        LXI      H,MSDB
060.066  042 204 074 02220        SHLD     LDD8A              Stuff system GRT address
060.071  072 226 051 02221        LDA      LABEL+LAB.SPG
060.074  062 206 074 02222        STA      LDD8B              Stuff system SPG
                     02223
060.077  001 000 001 02224        LXI      B,256
060.102  021 000 052 02225        LXI      D,MSDB
060.105  052 224 051 02226        LHLD     LABEL+LAB.GRT
060.110  315 275 031 02227        CALL     S.READ             Initialize temporary GRT
060.113  330         02228        RC
                     02229
                     02230   *    Load Driver
                     02231
060.114  052 377 073 02232        LHLD     GSPA
060.117  042 053 041 02233        SHLD     AIO.DTA
060.122  021 002 000 02234        LXI      D,DEV.RES
060.125  031         02235        DAD      D
060.126  353         02236        XCHG                        DE = $DEV.RES
060.127  315 231 060 02237        CALL     RDL                Request the driver load
060.132  330         02238        RC                          ERROR
                     02239
060.133  076 011     02240        MVI      A,DC.LOD
060.135  062 370 040 02241        STA      S.DDOPC            Set the OPEN code for the driver call
060.140  315 001 074 02242        CALL     LDD                Load the driver
                     02243
                     02244   *    Fix system device driver routine addresses
                     02245
060.143  315 362 073 02246        CALL     GSP
060.146  315 234 030 02247        CALL     $INDL              DE = system GRT address
060.151  002 000     02248        DW       UNT.GRT
060.153  353         02249        XCHG
060.154  042 204 074 02250        SHLD     LDD8A              Adjust temporary kludget GRT to the real thing
                     02251
060.157  052 377 073 02252        LHLD     GSPA               HL = SY: device table entry
060.162  315 234 030 02253        CALL     $INDL              DE = driver address
060.165  004 000     02254        DW       DEV.DDA
```

```
060.167  353          02255        XCHG
060.170  042 322 077  02256        SHLD     MSYDD             Stuff the Real Driver address
060.173  042 356 040  02257        SHLD     S.RFWA            FOrce as part of HDOS
060.176  353          02258        XCHG
                      02259
060.177  021 202 075  02260        LXI      D,ISY
060.202  315 107 075  02261        CALL     $INDS             Stuff mapped address in table
060.205  004 000      02262        DW       DEV.DDA
060.207  353          02263        XCHG
                      02264
                      02265   *    Fix device parameters & pointers
                      02266
060.210  052 377 073  02267        LHLD     GSPA
060.213  315 066 075  02268        CALL     $INDLB            A  = DEV.RES flag
060.216  002 000      02269        DW       DEV.RES
060.220  366 002      02270        ORI      DR.PR             Flag driver permanently resident
060.222  315 143 075  02271        CALL     $INDSB
060.225  002 000      02272        DW       DEV.RES
                      02273
060.227  247          02274        ANA      A                 CLEAR 'C'
060.230  311          02275        RET



                      02277   **   RDL - REQUEST DEVICE DRIVER.
                      02278   *
                      02279   *    RDL SETS A REQUEST FOR THE LOADING OF A DEVICE DRIVER.
                      02280   *
                      02281   *    THE DRIVER IS LOADED INTO MEMORY JUST BELOW *S.SYSM*.
                      02282   *
                      02283   *    ENTRY    (DE) = $DEV.RES
                      02284   *    EXIT     'C' SET IF ERROR
                      02285   *              (A) = ERROR CODE
                      02286   *             'C' CLEAR IF OK
                      02287   *             DEVLST POINTERS SET
                      02288   *    USES     A,F,B,C,H,L
                      02289
                      02290
060.231  325          02291   RDL  PUSH     D                 SAVE (DE)
060.232  353          02292        XCHG                       (HL) = $DEV.RES
060.233  042 366 040  02293        SHLD     S.DDDTA           SET DEVICE TABLE ADDRESS (OF DEV.RES)
060.236  021 011 000  02294        LXI      D,DEV.DVL-DEV.RES
060.241  031          02295        DAD      D                 (HL) = ADDRESS OF LENGTH
060.242  136          02296        MOV      E,M
060.243  043          02297        INX      H
060.244  126          02298        MOV      D,M               (DE) = LEN OF DRIVER
000.000               02299        ERRNZ    DEV.DVG-DEV.DVL-2
060.245  043          02300        INX      H
                      02301
060.246  176          02302        MOV      A,M               (A) = (DEV.DVG)
060.247  062 364 040  02303        STA      S.DDGRP           SET GROUP FOR FILE
                      02304
060.252  315 224 057  02305        CALL     AGT               HL = load address          /80.04.gc/
                      02306
060.255  076 010      02307        MVI      A,EC.NRD NO ROOM FOR DRIVER
060.257  332 312 060  02308        JC       RDL1              ERROR
```

```
                     02309
                     02310   *    SEE IF THIS IS ABOVE THE USER HIMEM
                     02311
060.262  353        02312        XCHG                        (DE) = NEW S.SYSM
060.263  042 362 040 02313        SHLD     S.DDLEN            SET LENGTH OF LOAD
060.266  052 322 040 02314        LHLD     S.USRM
060.271  043        02315        INX      H
                     02316
060.272  173        02317        MOV      A,E
060.273  225        02318        SUB      L
060.274  172        02319        MOV      A,D
060.275  234        02320        SBB      H
060.276  076 010     02321        MVI      A,EC.NRD
060.300  332 312 060 02322        JC       RDL1               NO ROOM
                     02323
060.303  353        02324        XCHG                        (HL) = NEW S.SYSM
060.304  042 320 040 02325        SHLD     S.SYSM
060.307  042 360 040 02326        SHLD     S.DDLDA            SET LOAD ADDR
                     02327
060.312  321        02328   RDL1 POP      D                  RESTORE (DE)
060.313  311        02329        RET



                     02331   **   TDD - TYPE DECIMAL DIGITS.
                     02332   *
                     02333   *    TDD TYPES A 16 BIT VALUE AS 1 TO 5 DECIMAL DIGITS.
                     02334   *
                     02335   *    ENTRY    (D,E) = VALUE
                     02336   *             (A) = DIGIT COUNT
                     02337   *    EXIT     VALUE TYPED.
                     02338   *    USES     A,B,C,F
                     02339
                     02340
060.314  076 005     02341   TDD. MVI      A,5
060.316  345        02342   TDD  PUSH     H
060.317  365        02343   TDD1 PUSH     PSW
060.320  041 363 060 02344        LXI      H,TDDA-2
060.323  007        02345        RLC                         (A) - DIGIT NUMBER*2
060.324  315 072 030 02346        CALL     $DADA
060.327  176        02347        MOV      A,M
060.330  043        02348        INX      H
060.331  146        02349        MOV      H,M
060.332  157        02350        MOV      L,A                (HL) = MULTIPLE OF 10
060.333  353        02351        XCHG                        (DE) = DEVISOR, (HL) = VALUE
060.334  076 377     02352        MVI      A,377Q
060.336  031        02353   TDD2 DAD      D
060.337  074        02354        INR      A
060.340  332 336 060 02355        JC       TDD2               IF MORE TO GO
060.343  306 060     02356        ADI      '0'
060.345  315 144 064 02357        CALL     $TYPEC.            TYPE DIGIT
060.350  175        02358        MOV      A,L
060.351  223        02359        SUB      E
060.352  137        02360        MOV      E,A                REMOVE EXTRA SUBTRACTION
060.353  174        02361        MOV      A,H
060.354  232        02362        SBB      D
```

```
   060.355  127          02363        MOV      D,A
   060.356  361          02364        POP      PSW
   060.357  075          02365        DCR      A
   060.360  302 317 060  02366        JNZ      TDD1              IF MORE DIGITS
   060.363  341          02367        POP      H
   060.364  311          02368        RET                        EXIT
                         02369
   060.365               02370   TDDA EQU      *
   060.365  377 377      02371        DW       -1
   060.367  366 377      02372        DW       -10
   060.371  234 377      02373        DW       -100
   060.373  030 374      02374        DW       -1000
   060.375  360 330      02375        DW       -10000



                         02377   **   SSD - SET SYSTEM DATE.                                          /80.08.gc/
                         02378   *
                         02379   *    SSD PROMPTS THE USER AS
                         02380   *
                         02381   *    DATE (DD-MMM-YY)?
                         02382   *
                         02383   *    THE 'DD-MMM-YY' FIELD IS REPLACED BY THE CURRENT
                         02384   *    SYSTEM DATE, IF A VALID ONE IS IN MEMORY.
                         02385   *
                         02386   *    Modified: If a valid date is not in memory, use
                         02387   *                      the default from the disk being booted.
                         02388   *                                                                    /80.08.gc/
                         02389   *
                         02390   *    IN THIS CASE, HITTING 'CR' IN REPLY CAUSES THE CURRENT DATE
                         02391   *    TO REMAIN.
                         02392   *
                         02393   *    ENTRY    NONE
                         02394   *    EXIT     NONE
                         02395   *    USES ALL
                         02396   *
                         02397
   060.377  072 024 051  02398   SSD  LDA      HOSTAB+FLT.PBO
   061.002  346 001      02399        ANI      PBO.DAT
   061.004  302 015 061  02400        JNZ      SSD0              User Wants the Date
                         02401
   061.007  041 000 000  02402        LXI      H,0
   061.012  303 310 061  02403        JMP      SSD8              Stuff the Date
                         02404
   061.015  257          02405   SSD0 XRA      A
   061.016  062 322 061  02406        STA      SSDA              Flag default legal
   061.021  052 310 040  02407        LHLD     S.DATC
   061.024  353          02408        XCHG                       DE = Compressed Date
   061.025  041 217 052  02409        LXI      H,SSDB            HL = Date FIeld
   061.030  172          02410        MOV      A,D
   061.031  263          02411        ORA      E
   061.032  312 062 061  02412        JZ       SSD1              No-Date is Illegal Here
   061.035  315 170 063  02413        CALL     $DAD
   061.040  332 062 061  02414        JC       SSD1              Illegal Coded Date
                         02415
                         02416   *    Compare Decoded date against RAM date
```

```
                    02417
061.043  001 011 000 02418        LXI     B,9
061.046  021 277 040 02419        LXI     D,S.DATE
061.051  041 217 052 02420        LXI     H,SSDB
061.054  315 060 030 02421        CALL    $COMP
061.057  312 114 061 02422        JZ      SSD2            Date field assumed legal
                    02423
                    02424   *     Set a new default date
                    02425
061.062  052 207 042 02426   SSD1 LHLD    SB.DAT
061.065  042 310 040 02427        SHLD    S.DATC          Set new default date
061.070  353         02428        XCHG                    DE = Compressed Date
061.071  041 277 040 02429        LXI     H,S.DATE HL = Date FIeld
061.074  172         02430        MOV     A,D
061.075  263         02431        ORA     E
061.076  312 107 061 02432        JZ      SSD1.5          No-Date
061.101  315 170 063 02433        CALL    $DAD
061.104  322 114 061 02434        JNC     SSD2            Legal default date
061.107  076 001     02435   SSD1.5       MVI     A,1
061.111  062 322 061 02436        STA     SSDA            Set Default Illegal
                    02437
                    02438   *     Get date from user
                    02439
061.114  315 136 031 02440   SSD2 CALL    $TYPTX
061.117  104 141 164 02441        DB      'Date ','('+200Q
061.125  072 322 061 02442        LDA     SSDA
061.130  247         02443        ANA     A
061.131  312 153 061 02444        JZ      SSD3            Legal Default Date
                    02445
061.134  315 136 031 02446        CALL    $TYPTX
061.137  104 104 055 02447        DB      'DD-MMM-Y','Y'+200Q
061.150  303 163 061 02448        JMP     SSD4
                    02449
061.153  076 011     02450   SSD3 MVI     A,9
061.155  041 277 040 02451        LXI     H,S.DATE
061.160  315 226 064 02452        CALL    $TYPCC          Type a good date
                    02453
                    02454   *     GET REPLY
                    02455
061.163  315 136 031 02456   SSD4 CALL    $TYPTX
061.166  051 077 240 02457        DB      ')?',' '+200Q
061.171  041 217 052 02458        LXI     H,SSDB
061.174  315 257 062 02459        CALL    $RTL.           READ TEXT LINE (UPPER CASE)
061.177  332 114 061 02460        JC      SSD2            CTL-D STRUCK
061.202  176         02461        MOV     A,M
061.203  247         02462        ANA     A
061.204  302 217 061 02463        JNZ     SSD5            GIVEN REPLY
                    02464
                    02465   *     HE DEFAULTED. SEE IF DEFAULT ALLOWED
                    02466
061.207  072 322 061 02467        LDA     SSDA
061.212  247         02468        ANA     A
061.213  310         02469        RZ                      DEFAULT OK
061.214  303 225 061 02470        JMP     SSD6            MAKE IT MORE CLEAR WHAT WE WANT
                    02471
                    02472   *     CACK DATE
                    02473
```

```
   061.217  315 324 062 02474   SSD5 CALL     $CAD              CONVERT AUGUSTAN DATE
   061.222  322 307 061 02475        JNC      SSD7              DATE GOOD
                        02476
                        02477   *    HIS REPLY BAD. TRY AGAIN
                        02478
   061.225  315 136 031 02479   SSD6 CALL     $TYPTX
   061.230  007 040 105 02480        DB       BELL,' ENTER DATE AS DD-MMM-YY (I.E., 02-JUL-77)',ENL
   061.304  303 114 061 02481        JMP      SSD2              TRY AGAIN
                        02482
                        02483   *    DATE IS GOOD. SETUP TWO DATE FIELDS FOR SYSTEM
                        02484
   061.307  353         02485   SSD7 XCHG
   061.310  042 310 040 02486   SSD8 SHLD     S.DATC            SET DATE CODE
   061.313  353         02487        XCHG
   061.314  041 277 040 02488        LXI      H,S.DATE
   061.317  303 170 063 02489        JMP      $DAD              DECODE DATE INTO ASCII AND RETURN
                        02490
   061.322  000         02491   SSDA DB       0                 =0 IFF DEFAULT DATE ALLOWED



                        02493   **   UBP      - Update Boot Parameters
                        02494   *
                        02495   *    UBP updates the boot parameters by rewriting sector
                        02496   *    zero.  Since track 0 is written as volume zero, the
                        02497   *    volume number must be temporarily adjusted.
                        02498   *
                        02499   *    ENTRY:   S.DATC   = Current Compressed Date
                        02500   *
                        02501   *    EXIT:    None
                        02502   *
                        02503   *    USES:    ALL
                        02504   *
                        02505
   061.323  072 203 042 02506   UBP  LDA      SB.VER
   061.326  376 040     02507        CPI      VERS
   061.330  300         02508        RNZ                        Only Update if versions match
                        02509
   061.331  001 035 000 02510        LXI      B,UBPAL
   061.334  021 203 042 02511        LXI      D,SB.VER
   061.337  041 074 062 02512        LXI      H,UBPA
   061.342  315 252 030 02513        CALL     $MOVE             Get the original parameters
                        02514
   061.345  052 310 040 02515        LHLD     S.DATC
   061.350  042 100 062 02516        SHLD     UBPA-SB.VER+SB.DAT Update Date
   061.353  052 344 040 02517        LHLD     S.BAUD
   061.356  042 076 062 02518        SHLD     UBPA-SB.VER+SB.BAU Update Baud-Rate
                        02519
   061.361  016 035     02520        MVI      C,UBPAL
   061.363  021 203 042 02521        LXI      D,SB.VER
   061.366  041 074 062 02522        LXI      H,UBPA
   061.371  315 060 030 02523        CALL     $COMP
   061.374  310         02524        RZ                         No change in parameters
                        02525
   061.375  001 035 000 02526        LXI      B,UBPAL
   062.000  021 074 062 02527        LXI      D,UBPA
```

```
   062.003  041 203 042 02528        LXI     H,SB.VER
   062.006  315 252 030 02529        CALL    $MOVE           Move the new data into parameter area
                        02530
   062.011  041 000 000 02531        LXI     H,0
   062.014  076 010     02532        MVI     A,DC.MOU
   062.016  315 130 040 02533        CALL    SYDD            Mount the disk as volume zero
   062.021  334 013 041 02534        CC      S.FASER
                        02535
   062.024  001 000 001 02536        LXI     B,256
   062.027  021 200 042 02537        LXI     D,SB.BOO
   062.032  041 000 000 02538        LXI     H,0
   062.035  076 001     02539        MVI     A,DC.WRI
   062.037  315 130 040 02540        CALL    SYDD            Re-Write the Sector
   062.042  322 055 062 02541        JNC     UBP1
   062.045  376 025     02542        CPI     EC.WP
   062.047  312 055 062 02543        JZ      UBP1            Ignore Write-Protect Error
                        02544
   062.052  315 013 041 02545        CALL    S.FASER         Other errors are fatal
                        02546
   062.055  072 217 051 02547   UBP1 LDA     LABEL+LAB.SER
   062.060  157         02548        MOV     L,A             L = Volume number
   062.061  046 000     02549        MVI     H,0
   062.063  076 010     02550        MVI     A,DC.MOU
   062.065  315 130 040 02551        CALL    SYDD            Re-Mount the disk
   062.070  334 013 041 02552        CC      S.FASER
                        02553
   062.073  311         02554        RET
                        02555
   062.074              02556   UBPA DS      SB.BPE-SB.VER   Reserve Space for temporary
   000.035              02557   UBPAL        EQU     *-UBPA
   062.131              02558        XTEXT   CDEHL



                        02560X  **   $CDEHL   - COMPARE (DE) TO (HL)
                        02561X  *
                        02562X  *    $CDEHL COMPARES (DE) TO (HL) FOR EQUALITY.
                        02563X  *
                        02564X  *    ENTRY    NONE
                        02565X  *    EXIT     'Z' SET IF (DE) = (HL)
                        02566X  *    USES     A,F
                        02567X
   030.216              02568X  $CDEHL       EQU     30216A          IN H17 ROM
   062.131              02569        XTEXT   MCU
                        02570X



                        02572X  **   MCU - MAP LOWER CASE TO UPPER CASE.
                        02573X  *
                        02574X  *    MCU MAPS A LOWER CASE ALPHABETIC TO UPPER
                        02575X  *    CASE.
                        02576X  *
                        02577X  *    ENTRY    (A) = CHARACTER
                        02578X  *    EXIT     (A) = CHARACTER RESULT
```

```
                     02579X  *    USES     A,F
                     02580X
                     02581X
062.131  376 141     02582X  $MCU CPI      'a'
062.133  330         02583X       RC                        NOT LOWER CASE
062.134  376 173     02584X       CPI      'z'+1
062.136  320         02585X       RNC                       NOT LOWER CASE
062.137  326 040     02586X       SUI      'a'-'A'
062.141  311         02587X       RET
062.142              02588        XTEXT    MLU



                     02590X  **   MLU - MAP LOWER CASE LINE TO UPPER CASE.
                     02591X  *
                     02592X  *    MLU MAPS THE LOWER CASE ALPHABETICS IN A LINE TO UPPER CASE.
                     02593X  *
                     02594X  *    ENTRY    (HL)  = LINE FWA
                     02595X  *    EXIT     NONE
                     02596X  *    USES     NONE
                     02597X
                     02598X
062.142  365         02599X  $MLU PUSH     PSW               SAVE (PSW)
062.143  345         02600X       PUSH     H                 SAVE FWA
062.144  053         02601X       DCX      H                 ANTICIPATE INX H
062.145  043         02602X  $MLU1        INX      H
062.146  176         02603X       MOV      A,M               (A)= CHARACTER
062.147  315 131 062 02604X       CALL     $MCU              MAP CHAR TO UPPER
062.152  167         02605X       MOV      M,A
062.153  247         02606X       ANA      A
062.154  302 145 062 02607X       JNZ      $MLU1             MORE TO GO
062.157  341         02608X       POP      H                 RESTORE (HL)
062.160  361         02609X       POP      PSW               RESTORE (PSW)
062.161  311         02610X       RET
062.162              02611        XTEXT    DTB



                     02613X  **   $DTB - DELETE TRAILING BLANKS.
                     02614X  *
                     02615X  *    $DTB DELETES THE TRAILING BLANKS FROM A CODED LINE.
                     02616X  *
                     02617X  *    ENTRY    (HL) = LINE FWA
                     02618X  *    EXIT     (A) = LENGTH OF RESULT (ENCLUDING 00 TERMINATOR BYTE)
                     02619X  *    USES     A,F
                     02620X
                     02621X
062.162  325         02622X  $DTB PUSH     D                 SAVE (DE)
062.163  124         02623X       MOV      D,H
062.164  135         02624X       MOV      E,L               (DE) = FWA
062.165  033         02625X       DCX      D                 (DE) = FWA-1
062.166  176         02626X  $DTB1        MOV      A,M
062.167  043         02627X       INX      H
062.170  247         02628X       ANA      A                 FIND END OF LINE
062.171  302 166 062 02629X       JNZ      $DTB1
```

```
   062.174  053          02630X       DCX      H                 (HL) = ADDRESS OF TERMINATING ZERO BYTE
                         02631X
                         02632X  *    GOT END OF LINE. DELETE TRAILING BLANKS
                         02633X
   062.175  053          02634X  $DTB2         DCX      H                 BACKUP ONE CHARACTER
   062.176  315 216 030  02635X       CALL     $CDEHL
   062.201  312 212 062  02636X       JE       $DTB3             GONE PAST FRONT OF LINE, MUST BE ALL BLANKS
   062.204  176          02637X       MOV      A,M
   062.205  376 040      02638X       CPI      ' '
   062.207  312 175 062  02639X       JE       $DTB2             GOT BLANK
                         02640X
                         02641X  *    HAVE TRIMED LINE. COMPUTE LENGTH
                         02642X
   062.212  043          02643X  $DTB3         INX      H
   062.213  066 000      02644X       MVI      M,0               TERMINATE LINE
   062.215  175          02645X       MOV      A,L
   062.216  223          02646X       SUB      E                 (A) = LENGTH +1 (FOR 00 BYTE)
   062.217  353          02647X       XCHG
   062.220  043          02648X       INX      H                 (HL) = LINE FWA
   062.221  321          02649X       POP      D                 RESTORE (DE)
   062.222  311          02650X       RET
   062.223               02651        XTEXT    MOVEL



                         02653X  **   $MOVEL- MOVE DATA
                         02654X  *
                         02655X  *    $MOVEL MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
                         02656X  *    IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
                         02657X  *    FIRST TO LAST.
                         02658X  *
                         02659X  *    IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
                         02660X  *    LAST TO FIRST.
                         02661X  *
                         02662X  *    THIS IS DONE SO THAT AN OVERLAPPED MOVE WILL NOT 'RIPPLE'.
                         02663X  *
                         02664X  *    CALL     $MOVEL
                         02665X  *    DW       COUNT
                         02666X  *    DW       FROM
                         02667X  *    DW       TO
                         02668X  *
                         02669X  *    ENTRY    ((SP)) = RET
                         02670X  *             (RET+0) = COUNT (WORD VALUE)
                         02671X  *             (RET+2) = FROM
                         02672X  *             (RET+4) = TO
                         02673X  *    EXIT     TO (RET+6)
                         02674X  *             (DE) = ADDRESS OF NEXT FROM BYTE
                         02675X  *             (HL) = ADDRESS OF NEXT *TO* BYTE
                         02676X  *             'C' CLEAR
                         02677X  *    USES     ALL
                         02678X
                         02679X
   062.223  341          02680X  $MOVEL        POP      H                 (HL) = RET
   062.224  116          02681X       MOV      C,M
   062.225  043          02682X       INX      H
   062.226  106          02683X       MOV      B,M               (BC) = COUNT
```

```
   062.227  043          02684X       INX      H
   062.230  136          02685X       MOV      E,M
   062.231  043          02686X       INX      H
   062.232  126          02687X       MOV      D,M               (DE) = FROM
   062.233  043          02688X       INX      H
   062.234  325          02689X       PUSH     D                 ((SP)) = FROM
   062.235  136          02690X       MOV      E,M
   062.236  043          02691X       INX      H
   062.237  126          02692X       MOV      D,M               (DE) = TO
   062.240  043          02693X       INX      H
   062.241  343          02694X       XTHL                       ((SP)) = RET, (HL) = FROM
   062.242  353          02695X       XCHG                       (DE) = FROM , (HL) = TO
   062.243  303 252 030 02696X       JMP      $MOVE             MOVE IT
   062.246              02697        XTEXT    RCHAR
                        02698X



                        02700X  **   $RCHAR - READ SINGLE CHARACTER FROM CONSOLE.
                        02701X  *
                        02702X  *    ENTRY    NONE
                        02703X  *    EXIT     (A) = CHARACTER
                        02704X  *    USES     A,F
                        02705X
                        02706X
   062.246  377 001     02707X  $RCHAR        DB       SYSCALL,.SCIN
   062.250  332 246 062 02708X       JC       $RCHAR            NOT READY
   062.253  311          02709X       RET
                        02710X
   062.254  377 002     02711X  $WCHAR        DB       SYSCALL,.SCOUT
   062.256  311          02712X       RET
   062.257              02713        XTEXT    MU10



                        02715X  **   $MU10 - MULTIPLY UNSIGNED 16 BIT QUANTITY BY 10.
                        02716X  *
                        02717X  *    (HL) = (DE)*10
                        02718X  *
                        02719X  *    ENTRY    (DE) = MULTIPLIER
                        02720X  *    EXIT     'C' CLEAR IF OK
                        02721X  *              (HL) = PRODUCT
                        02722X  *             'C' SET IF ERROR
                        02723X  *    USES     D,E,H,L,F
                        02724X
                        02725X
   030.324              02726X  $MU10         EQU      30324A            IN H17 ROM
   062.257              02727        XTEXT    RTL
                        02728X
```

```
                     02730X  **   $RTL - READ TEXT LINE.
                     02731X  *
                     02732X  *    $RTL READS A LINE FROM THE TERMINAL.
                     02733X  *
                     02734X  *    CHARACTER ARE ACCEPTED FROM THE TERMINAL, RUBOUT AND BACKSPACE
                     02735X  *    CHARACTERS ARE PROCESSED. WHEN A CARRIAGE RETURN IS ENTERED,
                     02736X  *    $RTL RETURNS.
                     02737X  *
                     02738X  *    ENTRY    (HL) = BUFFER FWA
                     02739X  *    EXIT     'C' CLEAR IF OK
                     02740X  *              DATA IN BUFFER
                     02741X  *              (A) = TEXT LENGTH
                     02742X  *             'C' SET IF CTL-D STRUCK
                     02743X  *    USES     A,F
                     02744X
                     02745X
  062.257  315 266 062 02746X  $RTL.         CALL     $RTL              $RTL IN UPPER CASE
  062.262  330         02747X       RC                          CTL-D
  062.263  303 142 062 02748X       JMP      $MLU               MAP LINE TO UPPER CASE
                     02749X
  062.266              02750X  $RTL EQU      *
  062.266  345         02751X       PUSH     H                  SAVE FWA
  062.267  315 246 062 02752X  $RTL1         CALL     $RCHAR
  062.272  376 004     02753X       CPI      CTLD
  062.274  312 321 062 02754X       JE       $RTL2              CTL-D STRUCK
  062.277  167         02755X       MOV      M,A
  062.300  043         02756X       INX      H
  062.301  376 012     02757X       CPI      NL
  062.303  302 267 062 02758X       JNE      $RTL1
  062.306  053         02759X       DCX      H
  062.307  066 000     02760X       MVI      M,0
  062.311  043         02761X       INX      H
                     02762X
                     02763X  *    ALL DONE. COMPUTE LENGTH
                     02764X
  062.312  353         02765X       XCHG                        (DE) = LWA+1
  062.313  343         02766X       XTHL                        (HL) = FWA
  062.314  173         02767X       MOV      A,E
  062.315  225         02768X       SUB      L                  (A) = LENGTH
  062.316  247         02769X       ANA      A                  CLEAR CARRY
  062.317  321         02770X       POP      D                  RESTORE (DE)
  062.320  311         02771X       RET
                     02772X
                     02773X  *    CTL-D STRUCK
                     02774X
  062.321  341         02775X  $RTL2         POP      H                  (HL) = FWA
  062.322  067         02776X       STC
  062.323  311         02777X       RET
  062.324              02778        XTEXT    CAD
```

```
                     02780X  **   $CAD - CODE AUGUSTAN DATE.
                     02781X  *
                     02782X  *    $CAD IS CALLED TO CODE AN AUGUSTAN DATE INTO THE FORM:
                     02783X  *
                     02784X  *
                     02785X  *    ----------------------------------------
                     02786X  *    I  0  I  6 BITS  I  4 BITS  I  5 BITS  I
                     02787X  *    ----------------------------------------
                     02788X  *             YEAR-70       MON  DAY
                     02789X  *              1-63         1-121-31
                     02790X  *
                     02791X  *    FROM THE FORM:
                     02792X  *
                     02793X  *    DD-MMM-YY
                     02794X  *
                     02795X  *    ENTRY    (HL) = ADDRESS OF STRING
                     02796X  *    EXIT     'C' CLEAR IF OK
                     02797X  *              (DE) = 15 BIT VALUE
                     02798X  *              (HL) ADVANCED PAST '-YY'
                     02799X  *             'C' SET IF ERROR
                     02800X  *    USES     ALL
                     02801X
                     02802X
  062.324  345       02803X  $CAD PUSH     H                                                    /80.08/GC/
  062.325  016 011   02804X       MVI      C,CADBL                                    /80.08.GC/
  062.327  021 157 063 02805X       LXI      D,CADB                                               /80.08.GC/
  062.332  315 060 030 02806X       CALL     $COMP                                                /80.08.GC/
  062.335  302 346 062 02807X       JNZ      CAD0              Is not 'No-Date'         /80.08.GC/
  062.340  321       02808X       POP      D                                                    /80.08.GC/
  062.341  021 000 000 02809X       LXI      D,0               0  =>  No Date                   /80.08.gc/
  062.344  247       02810X       ANA      A                 Clear 'C'                /80.08.gc/
  062.345  311       02811X       RET                                                           /80.08.GC/
                     02812X
  062.346  341       02813X  CAD0 POP      H                                                    /80.08.GC/
  062.347  315 004 064 02814X       CALL     $DDD              DECODE DECIMAL DIGITS
  062.352  330       02815X       RC                         ERROR
  062.353  172       02816X       MOV      A,D
  062.354  247       02817X       ANA      A
  062.355  067       02818X       STC                        ASSUME TOO LARGE
  062.356  300       02819X       RNZ                        TOO LARGE
  062.357  173       02820X       MOV      A,E
  062.360  247       02821X       ANA      A
  062.361  067       02822X       STC
  062.362  310       02823X       RZ                         TOO SMALL FOR DD
  062.363  376 040   02824X       CPI      32
  062.365  077       02825X       CMC
  062.366  330       02826X       RC                         TOO LARGE
  062.367  353       02827X       XCHG                       (HL) = DAY
  062.370  076 040   02828X       MVI      A,100000B
  062.372  205       02829X       ADD      L
  062.373  157       02830X       MOV      L,A               COUNT 1ST MONTH
  062.374  353       02831X       XCHG                       (DE) = DD*16+1, (HL) = ADDRESS
                     02832X
                     02833X  *    DECODE MONTH
                     02834X
  062.375  325       02835X       PUSH     D                 SAVE DD*16+1
  062.376  176       02836X       MOV      A,M
```

```
   062.377  043          02837X       INX      H
   063.000  376 055      02838X       CPI      '-'
   063.002  302 044 063  02839X       JNE      CAD2              FORMAT ERROR
   063.005  021 112 063  02840X       LXI      D,CADA            (DE) = MONTH TABLE ADDRESS
   063.010  001 003 000  02841X  CAD1 LXI      B,3
   063.013  345          02842X       PUSH     H                 SAVE TEXT ADDRESS, CADA ADDRESS
   063.014  325          02843X       PUSH     D
   063.015  315 060 030  02844X       CALL     $COMP             COMPARE
   063.020  321          02845X       POP      D                 (DE) = *CADA* ADDRESS
   063.021  312 047 063  02846X       JE       CAD3              GOT MONTH
   063.024  341          02847X       POP      H                 (HL) = BUFFER ADDRESS OF MMM-YY
   063.025  023          02848X       INX      D
   063.026  023          02849X       INX      D
   063.027  023          02850X       INX      D                 TRY NEXT MONTH
   063.030  343          02851X       XTHL
   063.031  076 040      02852X       MVI      A,100000B
   063.033  315 101 030  02853X       CALL     $DADA.            COUNT MONTH
   063.036  343          02854X       XTHL
   063.037  032          02855X       LDAX     D                 (A) = ENTRY IN CADA
   063.040  247          02856X       ANA      A
   063.041  302 010 063  02857X       JNZ      CAD1              MORE MONTHS TO GO
                         02858X
                         02859X  *    ERROR
                         02860X
   063.044  341          02861X  CAD2 POP      H                 CLEAR STACK
   063.045  067          02862X       STC
   063.046  311          02863X       RET                        FLAG ERROR
                         02864X
                         02865X  *    CRACK -YY
                         02866X
   063.047  301          02867X  CAD3 POP      B                 DISCARD ADDRESS IF MMM-YY
   063.050  176          02868X       MOV      A,M
   063.051  376 055      02869X       CPI      '-'
   063.053  302 044 063  02870X       JNE      CAD2              NOT -
   063.056  043          02871X       INX      H
   063.057  315 004 064  02872X       CALL     $DDD              DECODE DECIMAL DIGITS
   063.062  332 044 063  02873X       JC       CAD2              IF ERROR
   063.065  172          02874X       MOV      A,D
   063.066  247          02875X       ANA      A
   063.067  302 044 063  02876X       JNZ      CAD2              ERROR
   063.072  173          02877X       MOV      A,E               (A) = YEAR
   063.073  326 106      02878X       SUI      70                SUBTRACT DISPLACEMENT
   063.075  332 044 063  02879X       JC       CAD2              ERROR
   063.100  376 077      02880X       CPI      63
   063.102  322 044 063  02881X       JNC      CAD2              TOO LARGE
   063.105  321          02882X       POP      D                 (DE) = MONTH AND DAY
   063.106  207          02883X       ADD      A                 (A) = YEAR*2
   063.107  202          02884X       ADD      D
   063.110  127          02885X       MOV      D,A               MERGE WITH REST OF IT
   063.111  311          02886X       RET
                         02887X
   063.112               02888X  CADA DS       0                 TABLE OF MONTHS
   063.112  112 101 116  02889X       DB       'JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC',0
                         02890X
   063.157  040 116 157  02891X  CADB DB       ' No-Date '                                          /80.08.GC/
   000.011               02892X  CADBL        EQU      *-CADB                                              /80.08.GC/
   063.170               02893       XTEXT    DAD
```

```
                     02895X  **   $DAD - DECODE AUGUSTAN DATE.
                     02896X  *
                     02897X  *    $DAD DECODES A 15 BIT DATE CODE OF THE FORMAT:
                     02898X  *
                     02899X  *    ----------------------------------------
                     02900X  *    I  0  I  6 BITS  I  4 BITS  I  5 BITS  I
                     02901X  *    ----------------------------------------
                     02902X  *              YEAR-70       MON  DAY
                     02903X  *               1-63         1-121-31
                     02904X  *
                     02905X  *    TO THE FORM:
                     02906X  *
                     02907X  *    DD-MMM-YY
                     02908X  *
                     02909X  *    ENTRY    (DE) = 15 BIT VALUE
                     02910X  *             (HL) = ADDRESS FOR DECODE
                     02911X  *    EXIT     'C' CLEAR IF OK
                     02912X  *              (DE) = (DE)+9
                     02913X  *             'C' SET IF ERROR
                     02914X  *
                     02915X  *    USES     ALL
                     02916X
                     02917X
063.170  172         02918X  $DAD MOV      A,D                                                    /80.08.gc/
063.171  263         02919X       ORA      E                                                      /80.08.gc/
063.172  312 316 063 02920X       JZ       DAD2              No-Date                   /80.08.gc/
                     02921X
063.175  102         02922X       MOV      B,D
063.176  113         02923X       MOV      C,E
063.177  021 040 000 02924X       LXI      D,32
063.202  345         02925X       PUSH     H                 SAVE ADDRESS
063.203  315 106 030 02926X       CALL     $DU66             (DE) = DAY, (HL) = YEAR & MONTH
063.206  343         02927X       XTHL                       (HL) = ADDRESS
063.207  102         02928X       MOV      B,D
063.210  113         02929X       MOV      C,E
063.211  173         02930X       MOV      A,E
063.212  247         02931X       ANA      A
063.213  312 313 063 02932X       JZ       DAD1              BAD VALUE
063.216  076 002     02933X       MVI      A,2
063.220  315 157 031 02934X       CALL     $UDD              UNPACK DAY
063.223  066 055     02935X       MVI      M,'-'
063.225  043         02936X       INX      H
063.226  301         02937X       POP      B                 (BC) = YEAR & MONTH
063.227  021 020 000 02938X       LXI      D,16
063.232  345         02939X       PUSH     H                 SAVE ADDRESS
063.233  315 106 030 02940X       CALL     $DU66
063.236  343         02941X       XTHL                       (HL) = ADDRESS, ((SP)) = YEAR
063.237  173         02942X       MOV      A,E
063.240  207         02943X       ADD      A
063.241  203         02944X       ADD      E                 (A) = 3*MONTH
063.242  312 313 063 02945X       JZ       DAD1              BAD VALUE
063.245  376 047     02946X       CPI      13*3
063.247  322 313 063 02947X       JNC      DAD1              TOO LARGE
063.252  353         02948X       XCHG                       (DE) = ADDRESS
063.253  041 324 063 02949X       LXI      H,DADB-3
063.256  315 101 030 02950X       CALL     $DADA.            (HL) = ADDRESS OF MONTH
063.261  001 003 000 02951X       LXI      B,3
```

```
   063.264  353          02952X       XCHG                        (HL) = BUFFER ADDR, (DE) = ADDR IN 'DADB'
   063.265  315 252 030  02953X       CALL     $MOVE              MOVE MONTH IN
   063.270  066 055      02954X       MVI      M,'-'
   063.272  043          02955X       INX      H
   063.273  301          02956X       POP      B                  (BC) = YEAR
   063.274  171          02957X       MOV      A,C
   063.275  306 106      02958X       ADI      70
   063.277  376 144      02959X       CPI      100
   063.301  077          02960X       CMC
   063.302  330          02961X       RC                          TOO LARGE
   063.303  117          02962X       MOV      C,A                (BC) = YEAR
   063.304  076 002      02963X       MVI      A,2
   063.306  315 157 031  02964X       CALL     $UDD               UNPACK YEAR
   063.311  247          02965X       ANA      A
   063.312  311          02966X       RET
                         02967X
                         02968X  *    ILLEGAL FORMAT. (NOT ALL ILLEGALS EXIT HERE!)
                         02969X
   063.313  341          02970X  DAD1 POP      H                  RESTORE STACK
   063.314  067          02971X       STC                         FLAG ERROR
   063.315  311          02972X       RET
                         02973X
                         02974X  *    No-Date                                                 /80.08.gc/
                         02975X
   063.316  001 011 000  02976X  DAD2 LXI      B,DADCL                                         /80.08.gc/
   063.321  021 373 063  02977X       LXI      D,DADC                                                  /80.08.gc/
   063.324  303 252 030  02978X       JMP      $MOVE                                                   /80.08.gc/
                         02979X
   063.327  112 141 156  02980X  DADB DB       'JanFebMarAprMayJunJulAugSepOctNovDec'
                         02981X
   063.373  040 116 157  02982X  DADC DB       ' No-Date '                                             /80.08.gc/
   000.011               02983X  DADCL        EQU      *-DADC                                                 /80.08.gc/
                         02984X
                         02985X
   064.004               02986       XTEXT    DU66
                         02987X




                         02989X  **   $DU66 - UNSIGNED 16 / 16 DIVIDE.
                         02990X  *
                         02991X  *    (HL) = (BC)/(DE)
                         02992X  *
                         02993X  *    ENTRY    (BC), (DE) PRESET
                         02994X  *    EXIT     (HL) = RESULT
                         02995X  *             (DE) = REMAINDER
                         02996X  *    USES     ALL
                         02997X
                         02998X
   030.106               02999X  $DU66        EQU      30106A            IN H17 ROM
   064.004               03000       XTEXT    DDD
```

```
                    03002X  **   $DDD - DECODE DECIMAL DIGITS.
                    03003X  *
                    03004X  *    $DDD DECODES A STRING OF DECIMAL DIGITS INTO A DECIMAL INTEGER.
                    03005X  *
                    03006X  *    THE CHARACTERS ARE TAKEN OUT OF MEMORY. CONVERSION STOPS WITH THE
                    03007X  *    FIRST NON-DIGIT CHARACTER FOUND.
                    03008X  *
                    03009X  *    ENTRY    (HL) = ADDRESS OF CHARACTERS
                    03010X  *    EXIT     'C' CLEAR IF OK
                    03011X  *              (DE) = NUMBER
                    03012X  *              (HL) = INDEX OF FIRST NON-DIGIT ENCOUNTERED
                    03013X  *             'C' SET IF ERROR
                    03014X  *    USES     A,F,D,E,H,L
                    03015X
                    03016X
  064.004  021 000 000 03017X  $DDD LXI      D,0               (DE) = ACCUM
                    03018X
  064.007  176         03019X  $DDD1         MOV      A,M
  064.010  326 060     03020X       SUI      '0'
  064.012  077         03021X       CMC
  064.013  320         03022X       RNC                        TOO SMALL
  064.014  376 012     03023X       CPI      10
  064.016  320         03024X       RNC                        TOO LARGE
  064.017  043         03025X       INX      H                 ADVANCE ADDRESS
  064.020  345         03026X       PUSH     H                 SAVE (HL)
  064.021  315 324 030 03027X       CALL     $MU10             (HL) = ACCUM*10
  064.024  353         03028X       XCHG                       (DE) = ACCUM
  064.025  341         03029X       POP      H                 (HL) = ADDRESS OF STRING
  064.026  330         03030X       RC                         OVERFLOW
  064.027  203         03031X       ADD      E
  064.030  137         03032X       MOV      E,A
  064.031  076 000     03033X       MVI      A,0
  064.033  212         03034X       ADC      D
  064.034  127         03035X       MOV      D,A
  064.035  322 007 064 03036X       JNC      $DDD1             NOT OVERFLOW
  064.040  311         03037X       RET
  064.041              03038        XTEXT    UDD



                    03040X  **   $UDD - UNPACK DECIMAL DIGITS.
                    03041X  *
                    03042X  *    UDD CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF
                    03043X  *    DECIMAL DIGITS. THE RESULT IS ZERO FILLED.
                    03044X  *
                    03045X  *    ENTRY    (B,C) = ADDRESS VALUE
                    03046X  *             (A) = DIGIT COUNT
                    03047X  *             (H,L) = MEMORY ADDRESS
                    03048X  *    EXIT     (HL) = (HL) + (A)
                    03049X  *    USES     ALL
                    03050X
                    03051X
  031.157              03052X  $UDD EQU      31157A            IN H17 ROM
  064.041              03053        XTEXT    DADA
                    03054X
```

```
                    03056X  **   $DADA - PERFORM (H,L) = (H,L) + (0,A)
                    03057X  *
                    03058X  *    ENTRY    (H,L) = BEFORE VALUE
                    03059X  *             (A) = BEFORE VALUE
                    03060X  *    EXIT     (H,L) = (H,L) + (0,A)
                    03061X  *             'C' SET IF OVERFLOW
                    03062X  *    USES     F,H,L
                    03063X
                    03064X
  030.072           03065X  $DADA         EQU      30072A            IN H17 ROM
  064.041           03066        XTEXT    CRLF



                    03068X  **   $CRLF - TYPE CARRIAGE RETURN/ LINE FEED
                    03069X  *
                    03070X  *    $CRLF IS USED TO GENERATE PADDED CRLF'S.
                    03071X  *
                    03072X  *    ENTRY    NONE
                    03073X  *    EXIT     (A) = 0
                    03074X  *    USES     A,F
                    03075X
                    03076X
  064.041  076 012  03077X  $CRLF         MVI      A,NL
  064.043  377 002  03078X       DB       SYSCALL,.SCOUT
  064.045  257      03079X       XRA      A
  064.046  311      03080X       RET
  064.047           03081        XTEXT    TYPT2



                    03083X  **   $TYPTX - TYPE TEXT.
                    03084X  *
                    03085X  *    $TYPTX IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE.
                    03086X  *
                    03087X  *    IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED,
                    03088X  *    A BYTE WITH THE 200Q BIT SET IS THE LAST BYITE IN THE MESSAGE.
                    03089X  *
                    03090X  *    ENTRY    (RET) = TEXT
                    03091X  *    EXIT     TO (RET+LENGTH)
                    03092X  *    USES     A,F
                    03093X
                    03094X
  031.136           03095X  $TYPTX        EQU      31136A            IN H17 ROM
                    03096X
  031.144           03097X  $TYPTX. EQU   31144A            IN H17 ROM
  064.047           03098        XTEXT    TYPCH
```

```
                     03100X  **   $TYPCH - TYPE SINGLE CHARACTER.
                     03101X  *
                     03102X  *    ENTRY    (RET) = CHARACTER
                     03103X  *    EXIT     TO (RET)+1
                     03104X  *             (A) = CHARACTER TYPED
                     03105X
                     03106X
  064.047  343       03107X  $TYPCH        XTHL                        (HL) = RETURN ADDRESS
  064.050  176       03108X       MOV      A,M               (A) = CHARACTER
  064.051  043       03109X       INX      H
  064.052  343       03110X       XTHL                       RESTORE ADVANCED EXIT ADDRESS
                     03111X
                     03112X  **   $TYPC. - TYPE SINGLE CHARACTER.
                     03113X  *
                     03114X  *    ENTRY    (A) = CHARACTER
                     03115X  *    EXIT     TO (RET)
                     03116X
  064.053  377 002   03117X  $TYPC.        SCALL    .SCOUT
  064.055  311       03118X       RET
  064.056            03119        XTEXT    TFN



                     03121X  **   $TFN - TYPE FILE NAME.
                     03122X  *
                     03123X  *    $TFN TYPES THE FILE WHOSE NAME APPEARS IN AIO.XXX
                     03124X  *
                     03125X  *    ENTRY    NONE
                     03126X  *    EXIT     NONE
                     03127X  *    USES     A,F,B,H,L
                     03128X
                     03129X
  064.056  041 062 041 03130X  $TFN LXI      H,AIO.DIR+DIR.NAM
  064.061  006 010   03131X  $TFN.          MVI      B,8                                          /80.07.GC/
  064.063  315 074 064 03132X       CALL     $TFN1             TYPE NAME
  064.066  315 047 064 03133X       CALL     $TYPCH
  064.071  056       03134X       DB       '.'
  064.072  006 003   03135X       MVI      B,3
                     03136X
  064.074  176       03137X  $TFN1         MOV      A,M
  064.075  247       03138X       ANA      A
  064.076  304 053 064 03139X       CNZ      $TYPC.
  064.101  043       03140X       INX      H
  064.102  005       03141X       DCR      B
  064.103  302 074 064 03142X       JNZ      $TFN1
  064.106  311       03143X       RET
  064.107            03144        XTEXT    TYPET
```

```
                    03146X  **   $TYPET - TYPE TEXT.
                    03147X  *
                    03148X  *    $TYPET IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE
                    03149X  *    AT TASK TIME RATHER THAN AT INTERRUPT TIME.
                    03150X  *
                    03151X  *    IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED,
                    03152X  *    A BYTE WITH THE 200Q BIT SET IS THE LAST BYTE OF THE MESSAGE.
                    03153X  *
                    03154X  *    This routine modified to accomodate H8-4 ports by G.Chandler, 1-SEP-78
                    03155X  *    This routine assumes that the ports have been previously initialized,
                    03156X  *    and that S.CDB has been previously initialized.
                    03157X  *
                    03158X  *    ENTRY    (RET) = TEXT
                    03159X  *    EXIT     TO (RET+LENGTH)
                    03160X  *    USES     A,F
                    03161X
                    03162X
064.107  343       03163X  $TYPET        XTHL                        (HL) = TEXT ADDRESS
064.110  315 115 064 03164X       CALL     $TYPET.            TYPE IT
064.113  343       03165X       XTHL
064.114  311       03166X       RET
                    03167X
064.115  176       03168X  $TYPET.       MOV      A,M
064.116  346 177   03169X       ANI      177Q
064.120  304 144 064 03170X       CNZ      $TYPEC.            IF NOT CRLF
064.123  247       03171X       ANA      A
064.124  314 135 064 03172X       CZ       $TYPET1            IS CRLF
064.127  276       03173X       CMP      M
064.130  043       03174X       INX      H
064.131  300       03175X       RNE                         WAS 200 BIT SET
064.132  303 115 064 03176X       JMP      $TYPET.
                    03177X
                    03178X  *    TYPE CRLF
                    03179X
064.135  315 107 064 03180X  $TYPET1       CALL     $TYPET
064.140  015 212   03181X       DB       CR,LF+200Q
064.142  257       03182X       XRA      A                  RESTORE (A)
064.143  311       03183X       RET



                    03185X  **   $TYPEC. - TYPE SINGLE CHARACTER.
                    03186X  *
                    03187X  *    IF CR, PADD WITH 4 ZERO BYTES
                    03188X  *
                    03189X  *    ENTRY    (A) = CHARACTER
                    03190X  *    EXIT     (A) = CHARACTER
                    03191X  *    USES     A,F
                    03192X
                    03193X
064.144  365       03194X  $TYPEC.       PUSH     PSW                SAVE CHAR
064.145  072 343 040 03195X       LDA      S.CDB
064.150  376 001   03196X       CPI      CDB.H84
064.152  312 172 064 03197X       JZ       TYPEC2             IF H8-4 PORT
                    03198X
                    03199X  *    HAVE 8251 PORT FOR CONSOLE
```

```
                    03200X
   064.155  333 373     03201X  TYPEC1        IN       SC.UART+USR
   064.157  346 001     03202X        ANI      USR.TXR
   064.161  312 155 064 03203X        JZ       TYPEC1            NOT READY
   064.164  361         03204X        POP      PSW
   064.165  323 372     03205X        OUT      SC.UART+UDR
   064.167  303 204 064 03206X        JMP      TYPEC3
                        03207X
                        03208X  *     HAVE 8250 PORT FOR CONSOLE
                        03209X
   064.172  333 355     03210X  TYPEC2        IN       SC.ACE+UR.LSR
   064.174  346 040     03211X        ANI      UC.THE
   064.176  312 172 064 03212X        JZ       TYPEC2            NOT READY
   064.201  361         03213X        POP      PSW
   064.202  323 350     03214X        OUT      SC.ACE+UR.THR
                        03215X
   064.204  376 015     03216X  TYPEC3        CPI      CR
   064.206  300         03217X        RNE                        NOT CR
                        03218X
                        03219X  *     IS CR, PADD 4 TIMES
                        03220X
   064.207  076 004     03221X        MVI      A,4
   064.211  365         03222X  TYPEC4        PUSH     PSW
   064.212  257         03223X        XRA      A
   064.213  315 144 064 03224X        CALL     $TYPEC.
   064.216  361         03225X        POP      PSW
   064.217  075         03226X        DCR      A
   064.220  302 211 064 03227X        JNZ      TYPEC4
   064.223  076 015     03228X        MVI      A,CR
   064.225  311         03229X        RET
   064.226               03230         XTEXT    MU86
                        03231X



                        03233X  **    $MU86 - MULTIPLY 8X16 UNSIGNED.
                        03234X  *
                        03235X  *     $MU86 MULTIPLIES A 16 BIT VALUE BY A 8
                        03236X  *     BIT VALUE.
                        03237X  *
                        03238X  *     ENTRY     (A) = MULTIPLIER
                        03239X  *               (DE) = MULTIPLICAND
                        03240X  *     EXIT      (HL) = RESULT
                        03241X  *               'Z' SET IF NOT OVERFLOW
                        03242X  *     USES      A,F,H,L
                        03243X
   031.007              03244X  $MU86           EQU      31007A            IN H17 ROM
   064.226              03245         XTEXT    TYPCC
```

```
                    03247X  **   $TYPCC - TYPE A CHARACTER STRING BY COUNT.
                    03248X  *
                    03249X  *    $TYPCC TYPES A STRING OF CHARACTERS. THE CALLER SUPPLIES
                    03250X  *    THE CHARACTER ADDRESS AND COUNT.
                    03251X  *
                    03252X  *    ENTRY    (HL) = ADDRESS
                    03253X  *             (A) = COUNT
                    03254X  *    EXIT     (HL) = LAST CHARACTER ADDRESS+1
                    03255X  *    USES     A,F,H,L
                    03256X
                    03257X
064.226             03258X  $TYPCC        EQU      *
064.226  247       03259X       ANA      A
064.227  310       03260X       RZ                        NOTHING TO TYPE
064.230  365       03261X       PUSH     PSW              SAVE COUNT
064.231  176       03262X       MOV      A,M              (A) = CHARACTER
064.232  043       03263X       INX      H
064.233  377 002   03264X       DB       SYSCALL,.SCOUT
064.235  361       03265X       POP      PSW
064.236  075       03266X       DCR      A
064.237  303 226 064 03267X       JMP      $TYPCC



                    03269   **   PATCH - PATCH AREA
                    03270
064.242             03271   PATCH         DS       0
064.242  014 012 101 03272        DB       FF,NL,'ANOTHER FINE HEATH SOFTWARE PRODUCT'
                    03273
000.033             03274        ERRMI    64-*+PATCH                                    /79.06.gc/
064.307             03275        DS       64-*+PATCH                                    /79.06.gc/



                    03277   ********************************************************************************
                    03278   ********************************************************************************
                    03279   **                                                                            **
                    03280   **   BE VERY CAREFUL ABOUT THE PLACEMENT OF THESE BUFFERS, AND NOTE      **
                    03281   **   THAT THE  *LABEL*  BUFFER OVERLAYS CODE.  ( MAKE SURE THAT THE      **
                    03282   **   CODE WHICH IS OVERLAID IS NO LONGER NEEDED AT OVERLAY TIME.)        **
                    03283   **                                                  G. Chandler                **
                    03284   **                                                          79.11.gc     **
                    03285   **                                                                       **
                    03286   **            And even more overlap.                        80.05.gc     **
                    03287   **                                                                       **
                    03288   ********************************************************************************
                    03289   ********************************************************************************
051.217             03290   LABEL         EQU      RRH               256 BYTE BUFFER                    /79.11.GC/
052.217             03291   LABELE        EQU      LABEL+256                                    /80.0F.gc/
                    03292
052.217             03293   BUFF EQU      LABELE            256 BYTE BUFFER                    /80.05.gc/
053.217             03294   BUFFE         EQU      BUFF+256                                    /80.05.gc/
                    03295
024.000             03296   SDTA EQU      S.GRT0            512 BYTE BUFFER                    /80.05.gc/
026.000             03297   SDTAE         EQU      SDTA+512                                    /80.05.gc/
```

```
                    03298
   052.217          03299   SSDB EQU      BUFF              DATE BUFFER                          /79.12.GC/
                    03300
   052.000          03301   MSDB EQU      LABEL+255/256*256 Temporary GRT              /80.06.gc/
   064.342          03302   OVBUFE        EQU      *                END OF OVERLAID BUFFERS
                    03303
                    03304   **   WE MUST MAKE SURE THAT THERE IS ENOUGH MEMORY IN 8K SO THAT
                    03305   *    THE RESIDENT CODE WILL BE MOVED COMPLETELY ABOVE 'OVBUFE'
                    03306
P  375.226          03307        ERRMI    100000A-LENSYS-OVBUFE-20    NOT ENOUGH ROOM FOR EVERYBODY
```

```
                    03310   ***  SYSCALL DISPATCH.
                    03311   *
                    03312   *    THE SYSCALL DISPATCH HANDLER IS ENTERED VIA A SYSCALL INSTRUCTION.
                    03313   *
                    03314   *    IF THE PROCESSOR IS IN RESIDENT CODE, IT IS CALLED.
                    03315   *
                    03316   *    ALL CALLS WHICH INVOKE THE OVERLAY CODE HAVE THEIR STACK POINTER
                    03317   *    VALUE SAVED. THIS IS A KLUDGE FOR STACK PRESERVATION VIA 'LINK'
                    03318   *
                    03319   *    IF THE REQUIRED OVERLAY IS RESIDENT, IT IS CALLED.
                    03320   *
                    03321   *    IF THE OVERLAY IS NOT RESIDENT, LOAD IT, RELOCATE IT, AND CALL IT.
                    03322   *
                    03323   *    ENTRY    (SP) = RET
                    03324   *             (RET) = SYSCALL INDEX
                    03325   *    EXIT     'C' SET IF ILLEGAL CODE
                    03326   *              (A) = EC.ILC
                    03327   *             TO PROCESSOR IF A GOOD LOAD
                    03328   *             (SP) = PSW
                    03329   *             (SP+2) = RETURN ADDRESS (ADVANCED PAST CODE)
                    03330   *    USES     A,F
                    03331
                    03332
064.342             03333   FWAREL       EQU      *                ABS ADDRESS TO START RELOCATION
064.342             03334        CODE    +R               REMAINING CELLS ARE RELOCATED
064.342             03335   FWASYS       EQU      *                SYSTEM FWA
                    03336
064.342             03337   SYSCAL       EQU      *
064.342  062 006 041 03338        STA     S.CACC           SAVE (A)
064.345  343         03339        XTHL
064.346  176         03340        MOV     A,M              (A) = CODE
064.347  062 007 041 03341        STA     S.CODE           SET SYSTEM CODE
064.352  043         03342        INX     H                ADVANCE RETURN ADDRESS
064.353  343         03343        XTHL
                    03344
                    03345   *    Special case calls which may need mass storage device drivers   /2.0b/
                    03346
064.354  376 200     03347        CPI     .MOUNT                                                  /2.0b/
064.356  312 373 064 03348        JZ      SYS0                                                    /2.0b/
064.361  376 202     03349        CPI     .MONMS                                                  /2.0b/
064.363  312 373 064 03350        JZ      SYS0                                                    /2.0b/
064.366  376 204     03351        CPI     .RESET                                                  /2.0b/
064.370  302 025 065 03352        JNZ     SYS2                                                    /2.0b/
       /2.0b/
064.373  072 007 041 03353   SYS0 LDA     S.CODE                                                  /2.0b/
064.376  365         03354        PUSH    PSW                                             /80.04.GC/
064.377  072 006 041 03355        LDA     S.CACC                                          /80.04.GC/
065.002  365         03356        PUSH    PSW                                             /80.04.GC/
065.003  345         03357        PUSH    H                                               /80.04.GC/
065.004  377 062     03358        SCALL   .LOADD           preface .MOUNT by .LOADD   /80.04.GC/
065.006  341         03359        POP     H                restore device descriptor  /80.04.GC/
065.007  322 015 065 03360        JNC     SYS1                                            /80.04.GC/
                    03361
                    03362   *    Discard saved original parameters                   /80.04.gc/
                    03363
065.012  301         03364        POP     B                                               /80.04.GC/
065.013  301         03365        POP     B                                               /80.04.GC/
065.014  311         03366        RET                      exit with error                /80.04.GC/
```

```
                     03367
                     03368   *    Restore original parameters                          /80.04.gc/
                     03369
   065.015  361      03370   SYS1 POP      PSW                                              /80.04.GC/
   065.016  062 006 041 03371        STA      S.CACC                                        /80.04.GC/
   065.021  361      03372        POP      PSW                                              /80.04.GC/
   065.022  062 007 041 03373        STA      S.CODE                                        /80.04.GC/
   065.025           03374   SYS2 EQU      *
                     03375
   000.001           03376        IF DEBUG
                     03381        ENDIF
                     03382
   065.025  345      03383        PUSH     H
   065.026  041 371 040 03384        LXI      H,S.OVLFLSTORE S.OVLFL ON STACK WITHOUT
   065.031  146      03385        MOV      H,M                DAMAGING REGISTERS
   065.032  343      03386        XTHL
   065.033  315 127 065 03387   SYS3 CALL     SYSCAL0            CALL ALL SYSCALLS TO RETURN HERE
                     03388
                     03389   *    ALL SYSCALLS RETURN HERE.
                     03390   *
                     03391   *    LOAD ANY POSTPONED DEVICE DRIVERS, AND SEEZ IF A CTL-C OR CTL-Z
                     03392   *    WAS STRUCK.
                     03393
   065.036  365      03394        PUSH     PSW
   065.037  072 300 077 03395        LDA      SYSMODE
   065.042  075      03396        DCR      A
   065.043  302 055 065 03397        JNZ      SYS4               DONT RESTORE USER IF NOT FIRST LEVEL CALL
   065.046  072 371 040 03398        LDA      S.OVLFL
   000.000           03399        ERRNZ    OVL.UCS-200Q
   065.051  247      03400        ANA      A
   065.052  374 257 033 03401        CM       RUC                RESTORE USER CODE, IF SWAPPED
                     03402
   065.055  361      03403   SYS4 POP      PSW
   065.056  343      03404        XTHL                        (H) = OLD S.OVLFL
   065.057  365      03405        PUSH     PSW
   065.060  174      03406        MOV      A,H
   065.061  346 002  03407        ANI      OVL.RES
   065.063  312 101 065 03408        JZ       SYS5               WAS NOT PERMANENTLY RESIDENT BEFORE
   065.066  174      03409        MOV      A,H
   065.067  346 014  03410        ANI      OVL.NUM
   065.071  017      03411        RRC
   065.072  017      03412        RRC
   000.000           03413        ERRNZ    OVL.NUM-00001100B
   065.073  315 265 065 03414        CALL     LDON               WAS PERM. RESIDENT BEFORE
   065.076  334 115 066 03415        CC       FATSERR            OVERLAY WAS TOO BIG
                     03416
   065.101  072 361 040 03417   SYS5 LDA      S.DDLDA+1
   065.104  247      03418        ANA      A
   065.105  304 001 074 03419        CNZ      LDD                LOAD DEVICE DRIVER IF PENDING
   065.110  041 300 077 03420        LXI      H,SYSMODE
   065.113  363      03421        DI                          LOCK OUT CONSOLE INTERRUPTS UNTIL *CPA*
   065.114  065      03422        DCR      M                  DECREMENT NESTED SYSCAL COUNT
   065.115  314 222 031 03423        CZ       $WDR               IF RETURNING TO USER, WRITE DISABLE ROM
   065.120  315 230 073 03424        CALL     CPA                CHECK PENDING ABORT
   065.123  361      03425        POP      PSW
   065.124  341      03426        POP      H                  RESTORE USER (HL)
   065.125  373      03427        EI
```

```
   065.126  311          03428        RET                         EXIT
                         03429
                         03430
   000.001               03431        IF       DEBUG
                         03440        ENDIF



   065.127               03442   SYSCAL0 EQU   *
   065.127  345          03443        PUSH     H                  SAVE (HL)
   065.130  041 300 077  03444        LXI      H,SYSMODE
   065.133  064          03445        INR      M                  COUNT NESTED SYSCALL
   065.134  315 241 031  03446        CALL     $WER               WRITE ENABLE RAM AREA
   065.137  376 040      03447        CPI      .LINK
   065.141  322 176 065  03448        JNC      SYSCAL2            IS IN OVERLAY
   065.144  376 012      03449        CPI      .SYSRES
   065.146  332 156 065  03450        JC       SYSCAL1            IS RESIDENT
   065.151  076 003      03451        MVI      A,EC.ILC
   065.153  067          03452        STC
   065.154  341          03453        POP      H                  RESTORE (HL)
   065.155  311          03454        RET                         ERROR
                         03455
                         03456   *    DISPATCH RESIDENT CALLS
                         03457
   065.156  041 241 065  03458   SYSCAL1 LXI   H,SYSCALA
   065.161  207          03459        ADD      A                  (A) = CODE*2
   065.162  315 101 030  03460        CALL     $DADA.             (HL) = TABLE ADDRESS
   065.165  176          03461        MOV      A,M
   065.166  043          03462        INX      H
   065.167  146          03463        MOV      H,M
   065.170  157          03464        MOV      L,A                (HL) = CODE ADDRESS
   065.171  343          03465        XTHL                        PUT ON STACK
   065.172  072 006 041  03466        LDA      S.CACC             (A) = (ACC) UPON CALL
   065.175  311          03467        RET                         ENTER PROCESSOR CODE
                         03468
                         03469   *    DISPATCH OVERLAID CALLS
                         03470
   065.176  041 010 000  03471   SYSCAL2 LXI   H,8
   065.201  071          03472        DAD      SP
   065.202  042 035 041  03473        SHLD     S.OVSTK            SAVE STACK VALUE
   065.205  365          03474        PUSH     PSW                SAVE CODE
                         03475
   065.206  376 200      03476        CPI      .MOUNT
   065.210  322 220 065  03477        JNC      SYSCAL3            SECOND OVERLAY REQUIRED
   065.213  076 000      03478        MVI      A,OVL0             HDOSOVL .SYS
   065.215  303 222 065  03479        JMP      SYSCAL4
                         03480
   065.220  076 001      03481   SYSCAL3 MVI   A,OVL1             HDOSOVL2.SYS
                         03482
   065.222  315 265 065  03483   SYSCAL4 CALL  LDON               LOAD INDEXED OVERLAY
   065.225  334 115 066  03484        CC       FATSERR            OVERLAY TOO BIG
                         03485
                         03486   *    OVERLAY IS NOW LOADED
                         03487
   065.230  315 262 074  03488        CALL     OTI
   065.233  004 000      03489        DW       OVL.ENT            (HL) = ADDRESS OF ENTRY POINT
   065.235  315 211 030  03490        CALL     $HLIHL             (HL) = ENTRY POINT
```

```
   065.240  351         03491        PCHL                      ENTER CODE


                        03493   **   TABLE OF SYSCALL ROUTINES.
                        03494   *
                        03495   *    DW       ADDR             ENTRY ADDRESS
                        03496
                        03497
   065.241              03498   SYSCALA DS    0
                        03499
   065.241  171 066     03500        DW       EXIT             RETURN TO MONITOR
                        03501
   065.243  014 067     03502        DW       SCIN             READ FROM SYSTEM CONSOLE
                        03503
   065.245  304 070     03504        DW       SCOUT            WRITE TO SYSTEM CONSOLE
                        03505
   065.247  340 071     03506        DW       PRINT            WRITE LINE TO SYSTEM CONSOLE
                        03507
   065.251  254 071     03508        DW       READ             READ DATA
                        03509
   065.253  305 071     03510        DW       WRITE            WRITE DATA
                        03511
   065.255  354 071     03512        DW       CONSL            SET/READ CONSOLE OPTIONS
   065.257  002 072     03513        DW       CLRCO            CLEAR CONSOLE TYPE AHEAD
                        03514
   065.261  030 072     03515        DW       LOADO            LOAD SPECIFIED OVERLAY
                        03516
   065.263  103 072     03517        DW       VERSN
```

```
                     03519   **   LDON     - LOAD OVERLAY BY NUMBER
                     03520   *
                     03521   *    LOAD THE SPECIFIED OVERLAY ACCORDING TO THE NUMBER SPECIFIED.
                     03522   *    THE NUMBER CORRESPONDS TO THE INDEX IN TABLE SYSCALB,
                     03523   *    SET THE ENTRY POINT AN D FLAG BYTE IN THE OVERLAY TABLE.
                     03524   *
                     03525   *    IF THE OVERLAY IS ALREADY PRESENT, IT IS NOT LOADED.
                     03526   *
                     03527   *    IF A SMALLER OVERLAY IS ALREADY LOADED, IT IS TAKEN AS
                     03528   *    A FATAL SYSTEM ERROR.
                     03529   *
                     03530   *    ************************************************************************
                     03531   *    *                                                                      *
                     03532   *    *        OVERLAID CALLS TO OTHER OVERLAYS WILL PROBABLY NOT WORK      *
                     03533   *    *                                                                      *
                     03534   *    ************************************************************************
                     03535   *
                     03536   *
                     03537   *    ENTRY:   (A)      = INDEX OF OVERLAY TO BE LOADED
                     03538   *
                     03539   *    EXIT:    (PSW)    = 'C' CLEAR IF NO ERROR
                     03540   *                      = 'C' SET   IF    ERROR
                     03541   *                         (A)  = ERROR CODE
                     03542   *
                     03543   *    USES:    (FLAGS)
                     03544   *
                     03545
065.265  305        03546   LDON PUSH     B
065.266  325        03547        PUSH     D
065.267  345        03548        PUSH     H
                     03549
065.270  376 002    03550        CPI      OVLMAX             INDEX IS TOO BIG
065.272  322 101 066 03551        JNC      LDON5
                     03552
                     03553   *    CHECK TO SEE IF OVERLAY IS PERMANENTLY RESIDENT
                     03554
065.275  365        03555        PUSH     PSW                SAVE OVERLAY INDEX
065.276  315 262 074 03556        CALL     OTI
065.301  006 000    03557        DW       OVL.FLB            (HL) = ADDRESS OF FLAG BYTE
065.303  176        03558        MOV      A,M
065.304  346 002    03559        ANI      OVL.RES
065.306  302 343 065 03560        JNZ      LDON0              OVERLAY IS PERMANENTLY RESIDENT
065.311  361        03561        POP      PSW                RESTORE OVERLAY INDEX
                     03562
                     03563   *    CHECK TO SEE IF OVERLAY IS PRESENTLY IN MEMORY
                     03564
065.312  365        03565        PUSH     PSW                SAVE OVERLAY INDEX
065.313  207        03566        ADD      A
065.314  207        03567        ADD      A                  A = A*4
000.000             03568        ERRNZ    OVL.NUM-00001100B
065.315  107        03569        MOV      B,A                (B) = OVERLAY SOUGHT
065.316  072 371 040 03570        LDA      S.OVLFL
065.321  037        03571        RAR
065.322  322 351 065 03572        JNC      LDON2              NO OVERLAY LOADED
000.000             03573        ERRNZ    OVL.IN-1
                     03574
                     03575   *    CHECK TO SEE IF CURRENT OVERLAY IS THE ONE SOUGHT
```

```
                    03576
065.325  027       03577        RAL
065.326  346 014   03578        ANI      OVL.NUM
065.330  270       03579        CMP      B
065.331  312 343 065 03580       JZ       LDON0              CURRENT == SOUGHT
065.334  052 376 040 03581       LHLD     S.OVLS
065.337  353       03582        XCHG                        (DE) = OLD OVERLAY SIZE
065.340  303 354 065 03583       JMP      LDON3
                    03584
065.343  361       03585   LDON0        POP      PSW
065.344  247       03586        ANA      A                  CLEAR CARRY
065.345  341       03587   LDON1        POP      H
065.346  321       03588        POP      D
065.347  301       03589        POP      B
065.350  311       03590        RET
                    03591
                    03592   *    LOAD THE NEW OVERLAY
065.351  021 377 377 03593   LDON2        LXI      D,377377A LARGE (DE) IF NO PRESENT OVERLAYS
065.354  072 032 041 03594   LDON3        LDA      S.MOUNT
065.357  247       03595        ANA      A
065.360  076 051   03596        MVI      A,EC.NOS  NO OPERATING SYSTEM
065.362  312 107 066 03597       JZ       LDON6              NO O.S.
065.365  170       03598        MOV      A,B                (A)  = OVERLAY INDEX * 4
065.366  017       03599        RRC
065.367  017       03600        RRC                         (A)  = OVERLAY INDEX
065.370  315 262 074 03601       CALL     OTI
065.373  000 000   03602        DW       OVL.COD            (HL) = ADDRESS OF CODE ENTRY
065.375  345       03603        PUSH     H
065.376  315 211 030 03604       CALL     $HLIHL
066.001  042 004 041 03605       SHLD     S.OSN              SET NEW OVERLAY SECTOR NUMBER
066.004  341       03606        POP      H
066.005  043       03607        INX      H
066.006  043       03608        INX      H
000.000             03609        ERRNZ    OVL.SIZ-OVL.COD-2
066.007  315 211 030 03610       CALL     $HLIHL
066.012  072 371 040 03611       LDA      S.OVLFL
066.015  346 200   03612        ANI      OVL.UCS
066.017  312 040 066 03613       JZ       LDON4              NO USER CODE SWAPPED
066.022  315 053 075 03614       CALL     HLCPDE
066.025  312 040 066 03615       JZ       LDON4              NEW SIZE = PRESENT SIZE
066.030  332 040 066 03616       JC       LDON4              NEW SIZE < PRESENT SIZE
066.033  076 053   03617        MVI      A,EC.OTL  NEW SIZE > PRESENT SIZE
066.035  303 107 066 03618       JMP      LDON6
                    03619
                    03620   *    SET ENTRY POINT AND FLAG OVERLAY 'IN MEMORY'
                    03621
066.040  042 376 040 03622   LDON4        SHLD     S.OVLS             SET NEW OVERLAY SIZE
066.043  315 012 033 03623       CALL     LDO
066.046  072 371 040 03624       LDA      S.OVLFL
066.051  346 363   03625        ANI      377Q-OVL.NUM
066.053  260       03626        ORA      B
066.054  062 371 040 03627       STA      S.OVLFL            SET OVERLAY NUMBER IN FLAG BYTE
066.057  361       03628        POP      PSW                RESTORE OVERLAY INDEX
066.060  365       03629        PUSH     PSW                SAVE OVERLAY INDEX
066.061  315 262 074 03630       CALL     OTI                OVERLAY TABLE INDEXING
066.064  004 000   03631        DW       OVL.ENT            (HL) = ADDRESS OF THIS OVERLAY'S OVL.ENT BYTE
066.066  353       03632        XCHG
```

```
   066.067  052 000 041 03633        LHLD     S.OVLE
   066.072  353         03634        XCHG                        (DE) = OVERLAY ENTRY ADDRESS
   066.073  163         03635        MOV      M,E
   066.074  043         03636        INX      H
   066.075  162         03637        MOV      M,D                SET OVERLAY ENTRY ADDRESS IN OVERLAY TABLE
   066.076  303 343 065 03638        JMP      LDON0              RETURN
                        03639
   066.101  076 052     03640   LDON5          MVI      A,EC.IOI  ILLEGAL OVERLAY INDEX
   066.103  067         03641        STC                         FLAG ERROR
   066.104  303 345 065 03642        JMP      LDON1
                        03643
   066.107  063         03644   LDON6          INX      SP
   066.110  063         03645        INX      SP                 REMOVE OLD (PSW) FROM STACK
   066.111  067         03646        STC                         FLAG ERROR
   066.112  303 345 065 03647        JMP      LDON1



                        03649   *    FATAL SYSTEM ERROR
                        03650
   066.115  315 136 031 03651   FATSERR CALL  $TYPTX
   066.120  012 007 077 03652        DB       NL,BELL,'?02 FATAL SYSTEM ERROR ?',BELL,ENL
   066.154  257         03653   FATSER1 XRA   A
   066.155  062 010 040 03654        STA      .MFLAG
   066.160  323 373     03655        OUT      SC.UART+USR        CLEAR CONSOLE UART
   066.162  323 351     03656        OUT      SC.ACE+UR.IER
   066.164  373         03657        EI
   066.165  166         03658        HLT
   066.166  303 154 066 03659        JMP      FATSER1
```

```
                    03662   ***  EXIT - EXIT USER PROGRAM.
                    03663   *
                    03664   *    EXIT IS CALLED TO RETURN CONTROL TO THE SYSTEM COMMAND
                    03665   *    PROGRAM.
                    03666   *
                    03667   *    MVI      A,FLAG            =0 FOR NORMAL, =1 FOR ABORT
                    03668   *    DB       SYSCALL,.EXIT
                    03669   *
                    03670   *    FOR A NORMAL EXIT, THE CONTROL CHARACTER VECTORS ARE CLEARED.
                    03671   *    AND SYSCMD IS ENTERED.
                    03672   *
                    03673   *    FOR AN ABORT EXIT, THE DISK DRIVER IS RESET.
                    03674   *
                    03675   *    /79.06.gc/         IF  ( NO SYSTEM DISK  AND  S.ALONE IS SET)
                    03676   *                                    OR
                    03677   *                           ( SYSTEM DISK IS STILL MOUNTED )
                    03678   *
                    03679   *                           NORMAL LINK TO *SYSCMD.SYS*
                    03680   *
                    03681   *                       ELSE
                    03682   *
                    03683   *                         EXIT TO REBOOT CODE
                    03684   *
                    03685
                    03686
  066.171           03687   EXIT EQU      *
  066.171  061 200 042 03688        LXI      SP,STACK          RESET STACK                /79.12.GC/
  066.174  365       03689        PUSH     PSW                         SAVE CODE FOR LINKED PROGRAM
  066.175  247       03690        ANA      A                           SET CONDITION CODES
  066.176  076 201   03691        MVI      A,UO.CLK+UO.HLT
  066.200  062 010 040 03692      STA      .MFLAG                      REFRESH MFLAG
  066.203  312 217 066 03693      JZ       EXIT1                       NOT TO ABORT
                    03694
  066.206  257       03695        XRA      A                           System unit                /80.05.gc/
                    03696   *    LDA      SUNIT
  066.207  062 061 041 03697      STA      AIO.UNI                     SET SYSTEM DISK
  066.212  076 007   03698        MVI      A,DC.ABT
  066.214  315 130 040 03699      CALL     SYDD                        ABORT SYSTEM DISK
                    03700
  066.217  377 056   03701   EXIT1        SCALL    .CLEARA                    CLEAR ALL BUT THE LINK CHANNEL
  066.221  072 032 041 03702        LDA      S.MOUNT
  066.224  247       03703        ANA      A
  066.225  302 237 066 03704      JNZ      EXIT2                       SYSTEM IS MOUNTED
  066.230  072 301 077 03705      LDA      SALONE
  066.233  247       03706        ANA      A
  066.234  312 260 066 03707      JZ       EXIT3                       STAND-ALONE SWITCH IS NOT SET
                    03708
                    03709   *    LOAD EXIT OVERLAY
                    03710
  066.237  361       03711   EXIT2        POP      PSW                        RESTORE LINK CODE
  066.240  041 373 066 03712      LXI      H,EXITA
  066.243  061 200 042 03713      LXI      SP,STACK          RESET STACK
  066.246  377 040   03714        SCALL    .LINK                       LINK TO EXIT PROCESSOR
                    03715
                    03716   *    COULD NOT LINK
                    03717
  066.250  365       03718        PUSH     PSW                         SAVE CODE
```

```
   066.251  072 032 041 03719        LDA      S.MOUNT
   066.254  247         03720        ANA      A
   066.255  302 324 066 03721        JNZ      EXIT4                     CONSIDERED FATAL BECAUSE SYSTEM DISK
                        03722
   066.260  041 013 067 03723   EXIT3          LXI      H,EXITC                                                    /2.0a/
   066.263  377 003     03724        SCALL    .PRINT                    MAKE SURE WE ARE ON A NEW LINE
   066.265  076 377     03725        MVI      A,-1
   066.267  377 055     03726        SCALL    .CLEAR                    CLEAR THE LINK CHANNEL
                        03727
                        03728   *    Boot a new disk                                                        /2.0a/
   066.271  257         03729        XRA      A
   000.000              03730        ERRNZ    OVL0
   066.272  377 010     03731        SCALL    .LOADO                    Load *HDOSOVL0*
   066.274  334 115 066 03732        CC       FATSERR
   066.277  076 001     03733        MVI      A,OVL1
   066.301  377 010     03734        SCALL    .LOADO                    Load *HDOSOVL1*
   066.303  334 115 066 03735        CC       FATSERR
   066.306  377 206     03736        SCALL    .DAD                      Dismount all disks
   066.310  334 115 066 03737        CC       FATSERR
   066.313  315 007 073 03738        CALL     BND                       Read in the boot track
   066.316  061 200 042 03739        LXI      SP,STACK                                          /2.0c/
   066.321  303 200 042 03740        JMP      SB.BOO
                        03741
                        03742   *    ERROR    -  COULD NOT LINK TO   *SY0:SYSCMD.SYS*
                        03743
   066.324  041 354 066 03744   EXIT4          LXI      H,EXITB
   066.327  377 003     03745        SCALL    .PRINT                    PRINT MESSAGE
   066.331  361         03746        POP      PSW                       (A) = CODE
   066.332  046 000     03747        MVI      H,0
   066.334  377 057     03748        SCALL    .ERROR                    TYPE ERROR
   066.336  315 115 066 03749        CALL     FATSERR                   HALT
                        03750
   066.341  377 201     03751   EXIT5          SCALL    .DMOUN
   066.343  320         03752        RNC                                NO ERROR
   066.344  376 042     03753        CPI      EC.NVM
   066.346  310         03754        RZ                                 NO VOLUME MOUNTED NOT CONSIDERED FATAL
   066.347  377 057     03755        SCALL    .ERROR
   066.351  315 115 066 03756        CALL     FATSERR                   HALT
   066.354  012 007 077 03757   EXITB          DB       NL,BELL,'?02 Cant Run '
   066.373  123 131     03758   EXITA          DB       'SY'                        System Device             /80.05.gc/
   066.375  060 072 123 03759        DB       '0:SYSCMD.SYS',0,ENL      System Unit               /80.05.gc
   067.013  212         03760   EXITC          DB       ENL                                                        /2.0a/
```

```
                    03763   ***  SCIN - SYSTEM CONSOLE INPUT.
                    03764   *
                    03765   *    SCIN TAKES A SINGLE CHARACTER FROM THE CONSOLE INPUT
                    03766   *    BUFFER, IF ANY ARE AVAILABLE.
                    03767   *
                    03768   * L1 DB       SYSCALL,.SCIN
                    03769   *    JC       L1                CHARACTER NOT READY
                    03770   *
                    03771   *    ENTRY    NONE
                    03772   *    EXIT     'C' SET IF NO CHARACTER
                    03773   *             'C' CLEAR IF CHARACTER
                    03774   *              (A) =CHARACTER
                    03775   *    USES     A,F
                    03776
                    03777
067.014             03778   SCIN EQU      *
067.014  072 326 040 03779        LDA      S.CSLMD
000.000             03780        ERRNZ    CSL.CHR-1
067.017  037       03781        RAR
067.020  345       03782        PUSH     H                 SAVE (HL)
067.021  315 027 067 03783        CALL     SCIN1             GET CHARACTER
067.024  373       03784        EI
067.025  341       03785        POP      H
067.026  311       03786        RET
                    03787
                    03788   **   GET CHARACTER FROM BUFFER.
                    03789
067.027  363       03790   SCIN1         DI
067.030  332 041 067 03791        JC       SCIN2             NOT LINE MODE
                    03792
                    03793   *    LINE INPUT FORM
                    03794
067.033  072 302 077 03795        LDA      CSLLCNT
067.036  326 001     03796        SUI      1                 'C' SET IF NO LINES
067.040  330       03797        RC                         NO LINE YET
                    03798
                    03799   *    TAKE CHARACTER
                    03800
067.041  052 310 077 03801   SCIN2         LHLD     SCIOUT
067.044  072 306 077 03802        LDA      SCIIN
067.047  275       03803        CMP      L                 SEE IF EMPTY
067.050  067       03804        STC
067.051  310       03805        RE                         EMPTY
067.052  176       03806        MOV      A,M               (A) = CHARACTER
067.053  365       03807        PUSH     PSW
067.054  315 255 072 03808        CALL     ABP               ADVANCE BUFFER POINTER
067.057  042 310 077 03809        SHLD     SCIOUT            UPDATE POINTER
067.062  361       03810        POP      PSW               (A) - CHARACTER READ
                    03811
                    03812   *    MAP LOWER CASE TO UPPER, IF 'CTP.MLI' SET
                    03813
067.063  376 141     03814        CPI      'a'               LOWER CASE 'A'
067.065  332 104 067 03815        JC       SCIN2.5           NOT LWOER CASE
067.070  376 173     03816        CPI      'z'+1             LOWER CASE 'Z'
067.072  322 104 067 03817        JNC      SCIN2.5           NOT LOWER CASE
067.075  147       03818        MOV      H,A               (H) = CHARACTER
067.076  072 327 040 03819        LDA      S.CONTY
```

```
   000.000             03820        ERRNZ    'a'-'A'-CTP.MLI
   067.101  346 040    03821        ANI      CTP.MLI           (A) = 040Q IF TO MAP
   067.103  254        03822        XRA      H                 (A) = MAPPED CHARACTER
   067.104  376 012    03823   SCIN2.5 CPI   NL
   067.106  312 120 067 03824        JE       SCIN3             IS NEW LINE
   067.111  376 004    03825        CPI      CTLD              SEE IF CTLD
   067.113  312 120 067 03826        JE       SCIN3             Is New Line                          /2.0a/
   067.116  247        03827        ANA      A                 F = 'NC', A = character              /2.0a/
   067.117  311        03828        RET                                                             /2.0a/
                       03829
                       03830   *    END OF LOGICAL LINE
   067.120  041 302 077 03831   SCIN3         LXI      H,CSLLCNT
   067.123  065        03832        DCR      M                 COUNT LINE
   067.124  360        03833        RP                         NOT UNDERFLOW
   067.125  066 000    03834        MVI      M,0
   067.127  311        03835        RET
```

```
                     03838   **   SCINI - SYSTEM CONSOLE INPUT INTERRUPT.
                     03839   *
                     03840
                     03841
   067.130           03842   SCINI         EQU      *
   067.130  365      03843        PUSH     PSW
   067.131  345      03844        PUSH     H
   067.132  315 144 067 03845        CALL     SCINI0            PROCESS CHARACTER
   067.135  341      03846        POP      H
   067.136  315 230 073 03847        CALL     CPA               CHECK FOR PENDING ABORT
   067.141  361      03848        POP      PSW
   067.142  373      03849        EI
   067.143  311      03850        RET                        EXIT
                     03851
                     03852   *    PROCESS CHARACTER INTERRUPT
                     03853
   067.144  072 343 040 03854   SCINI0        LDA      S.CDB
   067.147  376 001     03855        CPI      CDB.H84
   067.151  312 161 067 03856        JZ       SCINI01                  IF  8250
                     03857
                     03858   *    HAVE 8251
   067.154  333 372     03859        IN       SC.UART+UDR
   067.156  303 163 067 03860        JMP      SCINI02
                     03861
                     03862   *    HAVE 8250
                     03863
   067.161  333 350     03864   SCINI01 IN    SC.ACE+UR.RBR
                     03865
   067.163  346 177     03866   SCINI02 ANI   177Q              TRIM PARITY
   067.165  315 242 071 03867        CALL     CRM               Check Raw Mode                          /2.0a/
   067.170  302 266 067 03868        JNZ      SCINI4            Ignore special character proc.          /2.0a
                     03869
   067.173  247      03870        ANA      A                                                         /2.0a/
   067.174  310      03871        RZ                         NULL CHARACTER
   067.175  376 012     03872        CPI      LF
   067.177  310      03873        RE                         IGNORE LINE-FEEDS
                     03874
                     03875   *    SEE IF SPECIAL CONTROL CHARACTER:
                     03876   *
                     03877   *    CTL-A,B,C, CTL-Z, CTL-O, CTL-P, CTL-Q, CTL-S
                     03878
   067.200  376 032     03879        CPI      CTLZ
   067.202  312 214 067 03880        JE       SCINI2            CTL-Z
   067.205  376 004     03881        CPI      04
   067.207  322 223 067 03882        JNC      SCINI3            NOT CTL-A, CTL-B, OR CTL-C
   067.212  356 002     03883        XRI      2                 CANCEL EFFECT OF NEXT INSTRUCTION
                     03884
                     03885   *    HAVE CTL-A,B,C OR CTL-Z
                     03886
   067.214  356 002     03887   SCINI2        XRI      2                 REMOVE '2' BIT IN ^Z (32Q -> 30Q)
   067.216  346 013     03888        ANI      CC.FLG+CZ.FLG     MASK OFF FLAG
   067.220  303 167 070 03889        JMP      PSC               PROCESS SPECIAL CHARACTER AND EXIT
                     03890
                     03891   *    SEE IF CTL-O THROUGH CTL-S
                     03892
   067.223  062 303 077 03893   SCINI3        STA      SCIPRE            SET PREVIOUS CHARACTER
   067.226  376 017     03894        CPI      CTLO
```

```
   067.230  332 266 067 03895        JC       SCINI4             NONE OF THESE
   067.233  376 024     03896        CPI      'T'-'@'
   067.235  322 266 067 03897        JNC      SCINI4             NONE OF THESE
   067.240  376 022     03898        CPI      'R'-'@'
   067.242  312 266 067 03899        JE       SCINI4             DONT TAKE CTL-R
                        03900
                        03901   *    IS CTL-O THROUGH CTL-S
                        03902
   067.245  207         03903   SCINI35 ADD   A                  (A) = 2 * CODE                        /80.04.GC/
   067.246  041 117 070 03904        LXI      H,SCINIA-'O'-'O'+'@'+'@'    (HL) = TABLE FWA - BIAS
   067.251  315 101 030 03905        CALL     $DADA.             (HL) = TABLE ADDRESS
   067.254  072 332 040 03906        LDA      S.CONFL
   067.257  246         03907        ANA      M                  CLEAR BITS
   067.260  043         03908        INX      H
   067.261  256         03909        XRA      M                  SET BITS
   067.262  062 332 040 03910        STA      S.CONFL
   067.265  311         03911        RET                         DONE
                        03912
                        03913   *    IS NOT AN 'ANYTIME' CONTROL CHARACTER. SEE IF LINE MODE
                        03914
   067.266  147         03915   SCINI4        MOV      H,A               (H) = CHARACTER
   067.267  072 326 040 03916        LDA      S.CSLMD
   000.000              03917        ERRNZ    CSL.CHR-1
   067.272  037         03918        RAR
   067.273  332 034 070 03919        JC       SCINI8             IS CHARACTER MODE
                        03920
                        03921   *    IS LINE MODE. SEE IF RUBOUT OR CTL-U
                        03922
   067.276  072 327 040 03923        LDA      S.CONTY
   067.301  346 002     03924        ANI      CTP.BKM            SEE IF MAPPING BKSP TO RUBOUT
   067.303  312 314 067 03925        JZ       SCIN4.3            NOT MAPPING
   067.306  076 010     03926        MVI      A,BKSP
   067.310  274         03927        CMP      H
   067.311  312 321 067 03928        JE       SCIN4.5            IS BKSP/RUBOUT
   067.314  174         03929   SCIN4.3 MOV   A,H
   067.315  074         03930        INR      A
   067.316  362 361 067 03931        JP       SCINI6             NOT RUBOUT
                        03932
                        03933   *    IS RUBOUT. TYPE FLAGS AND REMOVE CHARACTER
                        03934
   067.321  315 311 074 03935   SCIN4.5 CALL  RRC                REMOVE REGULAR CHARACTER
   067.324  310         03936        RE                          NONE TO REMOVE
   067.325  147         03937        MOV      H,A               (H) = REMOVED CHARACTER
   067.326  072 327 040 03938        LDA      S.CONTY
   000.000              03939        ERRNZ    CTP.BKS-200Q
   067.331  247         03940        ANA      A
   067.332  372 136 070 03941        JM       SCINI11            CAN BACKSPACE: ECHO <BKSP BLANK BKSP>
   067.335  072 304 077 03942        LDA      CSLRBF
   067.340  356 057     03943        XRI      '/'
   067.342  312 355 067 03944        JZ       SCINI5             ALREADY SET
   067.345  062 304 077 03945        STA      CSLRBF
   067.350  345         03946        PUSH     H
   067.351  315 326 070 03947        CALL     SCOUT1             TYPE '/'
   067.354  341         03948        POP      H
   067.355  174         03949   SCINI5        MOV      A,H
   067.356  303 326 070 03950        JMP      SCOUT1             ECHO CHARACTER
                        03951
```

```
   067.361  072 304 077 03952   SCINI6        LDA      CSLRBF
   067.364  356 057     03953         XRI      '/'                SEE IF RUBOUT PENDING
   067.366  302 003 070 03954         JNZ      SCINI65            NOT PENDING
   067.371  062 304 077 03955         STA      CSLRBF             SLEAR FLAG
   067.374  076 057     03956         MVI      A,'/'
   067.376  345         03957         PUSH     H
   067.377  315 326 070 03958         CALL     SCOUT1
   070.002  341         03959         POP      H
   070.003  174         03960   SCINI65 MOV    A,H                (A) = INPUT CHARACTER
   070.004  376 025     03961         CPI      'U'-'@'
   070.006  302 034 070 03962         JNE      SCINI8             NOT CTL-U
                        03963
                        03964   *     IS CTL-U
                        03965
   070.011  315 311 074 03966   SCINI7        CALL     RRC                REMOVE REGULAR CHARACTER
   070.014  302 011 070 03967         JNZ      SCINI7             MORE TO GO
   070.017  076 136     03968         MVI      A,'^'
   070.021  315 345 070 03969         CALL     SCOUT2             TYPE ^
   070.024  076 125     03970         MVI      A,'U'
   070.026  315 345 070 03971         CALL     SCOUT2             TYPE 'U'
   070.031  303 225 071 03972         JMP      CRLF               NEW LINE AND EXIT
                        03973
                        03974   *     HAVE REGULAR CHARACTER. STORE IF ROOM
                        03975
   070.034  345         03976   SCINI8        PUSH     H                  SAVE CHAR
   070.035  052 306 077 03977         LHLD     SCIIN
   070.040  345         03978         PUSH     H                  SAVE IN POINTER
   070.041  315 255 072 03979         CALL     ABP                ADVANCE BUFFER POINTER
   070.044  072 310 077 03980         LDA      SCIOUT
   070.047  275         03981         CMP      L
   070.050  302 062 070 03982         JNE      SCINI9             HAVE ROOM
                        03983
                        03984   *     TOO FULL. BEEP CHARACTER
                        03985
   070.053  341         03986         POP      H
   070.054  361         03987         POP      PSW
   070.055  076 007     03988         MVI      A,BELL
   070.057  303 326 070 03989         JMP      SCOUT1             BEEP
                        03990
                        03991   *     HAVE ROOM. WILL STORE CHARACTER
                        03992
   070.062  042 306 077 03993   SCINI9        SHLD     SCIIN
   070.065  341         03994         POP      H                  (HL) = POINTER
   070.066  361         03995         POP      PSW                (A) = CHAR
   070.067  167         03996         MOV      M,A                STORE
   070.070  376 004     03997         CPI      CTLD
   070.072  312 120 070 03998         JE       SCINI95            IS CTL-D
                        03999
                        04000   *     Conditionally map CR to NL
                        04001
   070.075  315 242 071 04002         CALL     CRM                Check Raw Mode
   070.100  302 113 070 04003         JNZ      SCINI93            Don't map because of raw mode
   070.103  376 015     04004         CPI      CR
   070.105  302 113 070 04005         JNE      SCINI93            NOT CR
   070.110  076 012     04006         MVI      A,NL
   070.112  167         04007         MOV      M,A                STORE NL
                        04008
```

```
   070.113  376 012     04009   SCINI93 CPI   NL
   070.115  302 124 070 04010         JNZ      SCINI10           Not the end of a line
                        04011
                        04012   *    HAVE SEEN END OF LOGICAL LINE
                        04013
   070.120  041 302 077 04014   SCINI95 LXI   H,CSLLCNT
   070.123  064         04015         INR      M                 COUNT LINE
   070.124  147         04016   SCINI10 MOV   H,A               (H) = CHAR
                        04017
                        04018   *    SEE IF TO ECHO
                        04019
   070.125  072 326 040 04020         LDA      S.CSLMD
   000.000              04021         ERRNZ    CSL.ECH-200Q
   070.130  027         04022         RAL
   070.131  174         04023         MOV      A,H               (A) = CHA
   070.132  322 326 070 04024         JNC      SCOUT1            AM TO ECHO
   070.135  311         04025         RET                        SUPRESS ECHO
                        04026
                        04027   *    HAVE BACKSPACE FOR TERMINAL WITH BACKSPACE CAPABILITY:
                        04028   *    ISSUE BKSP, BLANK, BKSP
                        04029
   070.136  076 010     04030   SCINI11 MVI   A,BKSP
   070.140  315 345 070 04031         CALL     SCOUT2
   070.143  076 040     04032         MVI      A,' '
   070.145  315 345 070 04033         CALL     SCOUT2            PRINT BKSP, <BLANK>, BKSP
   070.150  076 010     04034         MVI      A,BKSP
   070.152  303 345 070 04035         JMP      SCOUT2            PRINT AND EXIT
                        04036
                        04037
                        04038   **   PROCESSING FOR CTL-P, CTL-O, CTL-Q, CTL-S
                        04039   *
                        04040   *    FIRST BYTE = CLEAR MASK FOR S.CONFL
                        04041   *    2ND BYTE = XOR MASK FOR S.CONFL
                        04042
   070.155  377 001     04043   SCINIA        DB       -1,CO.FLG CTL-O
   070.157  376 000     04044         DB       377Q-CO.FLG,0     CTL-P
   070.161  177 000     04045         DB       377Q-CS.FLG,0     CTL-Q
   070.163  377 000     04046         DB       -1,0              CTL-R
   070.165  177 200     04047         DB       377Q-CS.FLG,CS.FLG CTL-S



                        04049   **   PSC - PROCESS SPECIAL CHARACTER.
                        04050   *
                        04051   *    PSC IS CALLED WHEN A SPECIAL INTERRUPT CHARACTER IS DETECTED
                        04052   *    (CTL-A, CTL-B, CTL-C, CTL-Z). PSC DECIDES IF SPECIAL
                        04053   *    SERVICE WILL BE NEEDED (IF REQUESTED BY USER FOR CTL-A, -B, AND -C,
                        04054   *    OR UPON 2ND CTL-Z)
                        04055   *
                        04056   *    IF SERVICE IS NEEDED, THE SERVICE ADDRESS IS STORED IN
                        04057   *    S.CADDR.
                        04058   *
                        04059   *    ENTRY    (A) = CHARACTER DETECTED
                        04060   *    EXIT     S.CADDR > 256 IF PROCESSING NEEDED
                        04061   *    USES     A,F,H,L
                        04062
```

```
                    04063
   070.167  305     04064   PSC  PUSH     B
   070.170  315 175 070 04065        CALL     PSC1              PROCESS
   070.173  301     04066        POP      B
   070.174  311     04067        RET
                    04068
   070.175  107     04069   PSC1 MOV      B,A               (B) = CHARACTER
   070.176  346 010 04070        ANI      CZ.FLG
   070.200  302 226 070 04071        JNZ      PSC2              IS CTL-Z
                    04072
                    04073   *    IS CTL-A, -B, OR -C
                    04074
   070.203  170     04075        MOV      A,B
   070.204  207     04076        ADD      A                 (A) = 2*CODE
   070.205  310     04077        RZ                         NONE
   070.206  041 333 040 04078        LXI      H,S.CCTAB-2
   070.211  315 101 030 04079        CALL     $DADA.            (HL) = ADDRESS OF ADDRESS
   070.214  315 211 030 04080        CALL     $HLIHL
   070.217  174     04081        MOV      A,H
   070.220  247     04082        ANA      A
   070.221  310     04083        RZ                         NONE TO SET
   070.222  042 333 040 04084        SHLD     S.CAADR           SET CONSOLE ABORT ADDRESS
   070.225  311     04085        RET
                    04086
                    04087   *    IS CTL-Z
                    04088
   070.226  072 032 041 04089   PSC2 LDA      S.MOUNT
   070.231  247     04090        ANA      A
   070.232  310     04091        RZ                         SYSTEM NOT MOUNTED, IGNORE
   070.233  373     04092        EI
   070.234  076 136 04093        MVI      A,'^'
   070.236  315 326 070 04094        CALL     SCOUT1
   070.241  076 132 04095        MVI      A,'Z'
   070.243  315 326 070 04096        CALL     SCOUT1
   070.246  072 303 077 04097        LDA      SCIPRE
   070.251  376 032 04098        CPI      CTLZ
   070.253  312 270 070 04099        JE       PSC3              2ND CTL-Z
   070.256  076 032 04100        MVI      A,CTLZ
   070.260  062 303 077 04101        STA      SCIPRE            SET CTL-Z AS PREVIOUS CHARACTER
                    04102
                    04103   *    TYPE '?' WARNING
                    04104
   070.263  076 077 04105        MVI      A,'?'
   070.265  303 326 070 04106        JMP      SCOUT1            OUTPUT AND RETURN
                    04107
                    04108   *    2ND CTL-Z HIT
                    04109
   070.270  076 021 04110   PSC3 MVI      A,CTLQ                                        /80.04.gc/
   070.272  315 245 067 04111        CALL     SCINI35           simulate CTL-Q              /80.04.gc/
   070.275  041 171 066 04112        LXI      H,EXIT
   070.300  042 333 040 04113        SHLD     S.CAADR           CONSOLE ABORT ADDRESS
   070.303  311     04114        RET
```

```
                     04117   **   SCOUT - SYSTEM CONSOLE OUTPUT.
                     04118   *
                     04119   *    SCOUT OUTPUTS A SINGLE CHARACTER TO THE CONSOLE. CURSOR POSITOINING
                     04120   *    IS KEPT TRACK OF, A 'NL' CHARACTER INDICATES A NEW LINE, 'CR' AND
                     04121   *    'LF' CHARACTERS SHOULD NOT BE USED.
                     04122   *
                     04123   *    NOTE THAT THERE ARE SOME GAMES PLAYED WITH THE PARITY BIT.
                     04124   *    SEE *CRLF* FOR DISCUSSION.
                     04125   *
                     04126   *    MVI      A,CHAR
                     04127   *    DB       SCALL,.SCOUT
                     04128   *
                     04129   *    ENTRY    (A) = CHARACTER
                     04130   *    EXIT     (A) = CHARACTER
                     04131   *    USES     NONE
                     04132
                     04133
 070.304             04134   SCOUT         EQU      *
 070.304  365        04135        PUSH     PSW               SAVE CHAR
 070.305  345        04136        PUSH     H                 SAVE (HL)
 070.306  147        04137        MOV      H,A               (A) = CHARACTER
 070.307  072 332 040 04138   SCOUT0        LDA      S.CONFL
 000.000             04139        ERRNZ    CS.FLG-200Q
 070.312  247        04140        ANA      A
 070.313  372 307 070 04141        JM       SCOUT0            AM IN WAIT MODE
 000.000             04142        ERRNZ    CO.FLG-1
 070.316  037        04143        RAR
 070.317  174        04144        MOV      A,H               (A) = CHARACTER
 070.320  324 326 070 04145        CNC      SCOUT1            PERFORM I/O IF NOT CTL-O
 070.323  341        04146        POP      H
 070.324  361        04147        POP      PSW
 070.325  311        04148        RET



                     04150   **   SCOUT1 - OUTPUT CHARACTER.
                     04151   *
                     04152   *    SCOUT1 IGNORES CTL-O AND CTL-S, AND IS USED BY HDOS CODE
                     04153   *    WHICH MUST NOT BE HELD UP.
                     04154   *
                     04155   *    SCOUT1 MAY BE CALLED WITH INTERRUPTS DISABLED.
                     04156   *
                     04157   *    ENTRY    (A) = CHARACTER
                     04158   *    EXIT     NONE
                     04159   *    USES     A,F,H,L
                     04160
 070.326  346 177     04161   SCOUT1        ANI      177Q              TRIM
                     04162
                     04163   *    IF LOWER CASE MAPPING TURNED ON, DOIT
                     04164
 070.330  376 140     04165        CPI      140Q
 070.332  332 345 070 04166        JC       SCOUT2            NOT LOWER CASE
 070.335  147        04167        MOV      H,A
 070.336  072 327 040 04168        LDA      S.CONTY
 070.341  207        04169        ADD      A
 070.342  346 040     04170        ANI      CTP.MLO*2(A) = 040Q IF MAPPING
```

```
   070.344  254          04171        XRA      H                 CLEA BIT IF MAPPING
                         04172
                         04173
                         04174   **   SOME ROUTINES CALL HERE (SCOUT ITSELF, RECURSIVELY)
                         04175   *    TO OUTPUT CHARACTERS WITHOUT THE CPU OVERHEAD OF  SCOUT OR
                         04176   *    SCOUT1.
                         04177
   070.345  376 011      04178   SCOUT2        CPI      TAB
   070.347  302 002 071  04179        JNE      SCOUT4            NOT TAB
                         04180
                         04181   *    HAVE TAB, EXPAND TO COLUMN
                         04182
   070.352  072 327 040  04183        LDA      S.CONTY
   000.000               04184        ERRNZ    CTP.TAB-1
   070.355  037          04185        RAR
   070.356  076 011      04186        MVI      A,TAB
   070.360  332 002 071  04187        JC       SCOUT4            TERMINAL WILL TAKE TABS
   070.363  076 040      04188   SCOUT3        MVI      A,' '
   070.365  315 002 071  04189        CALL     SCOUT4            TYPE BLANK
   070.370  072 330 040  04190        LDA      S.CUSOR
   070.373  075          04191        DCR      A
   070.374  346 007      04192        ANI      7
   070.376  302 363 070  04193        JNZ      SCOUT3            NOT TO FIELD
   071.001  311          04194        RET                        DONE
                         04195
                         04196   *    TYPE CHARACTER. (A) =CHARACTER
                         04197
   071.002  315 242 071  04198   SCOUT4        CALL     CRM              Check Raw Mode                          /2.0a/
   071.005  302 015 071  04199        JNZ      SCOUT45           Don't map NL                          /2.0a/
   071.010  376 012      04200        CPI      NL
   071.012  312 225 071  04201        JE       CRLF              IS A NEW LINE
                         04202
   071.015  041 330 040  04203   SCOUT45 LXI   H,S.CUSOR(M) = CONSOLE CURSOR POINTER                 /2.0a/
   071.020  376 015      04204        CPI      CR
   071.022  302 027 071  04205        JNE      SCOUT5            NOT CR
   071.025  066 001      04206        MVI      M,1               CLEAR POINTER
   071.027  376 010      04207   SCOUT5        CPI      BKSP
   071.031  302 042 071  04208        JNE      SCOUT6            IS NOT BKSP
   071.034  065          04209        DCR      M
   071.035  302 042 071  04210        JNZ      SCOUT6            NOT UNDERFLOW
   071.040  066 001      04211        MVI      M,1               RESET
   071.042  376 011      04212   SCOUT6        CPI      TAB
   071.044  302 060 071  04213        JNE      SCOUT7            NOT TAB
   071.047  176          04214        MOV      A,M
   071.050  306 007      04215        ADI      7Q
   071.052  346 370      04216        ANI      370Q
   071.054  074          04217        INR      A                 ADJUST COLUMN COUNT TO NEXT TAB
   071.055  167          04218        MOV      M,A
   071.056  076 011      04219        MVI      A,TAB
   071.060  346 177      04220   SCOUT7        ANI      177Q              TRIM TO 7 BITS
   071.062  376 040      04221        CPI      ' '               SEE IF PRINTING
   071.064  332 101 071  04222        JC       SCOUT8            NOT PRINTING
                         04223
                         04224   *    CHECK FOR CHARACTER WRAP?
                         04225
   071.067  365          04226        PUSH     PSW               SAVE CURRENT CHARACTER
   071.070  043          04227        INX      H
```

```
   000.000             04228        ERRNZ    S.CONWI-S.CUSOR-1
   071.071  176        04229        MOV      A,M               (A) = CONSOLE WIDTH
   071.072  053        04230        DCX      H
   000.000             04231        ERRNZ    S.CONWI-S.CUSOR-1
   071.073  276        04232        CMP      M
   071.074  334 225 071 04233        CC       CRLF              WIDTH < S.CURSOR AND ABOUT TO OUTPUT A PRINTING
   071.077  361        04234        POP      PSW                CHARACTER
                       04235
   071.100  064        04236        INR      M                 COUNT CHARACTER
                       04237
                       04238   *    OUTPUT CHARACTER.
                       04239
   071.101  147        04240   SCOUT8       MOV      H,A              Save Character                              /2.0a/
   071.102  072 327 040 04241        LDA      S.CONTY                                                  /2.0a/
   071.105  346 100    04242        ANI      CTP.FF            Check for Form Feed Proc.          /2.0a/
   071.107  174        04243        MOV      A,H               Restore Character                  /2.0a/
   071.110  302 120 071 04244        JNZ      SCOUT85           Don't Check Form Feed Proc.        /2.0a/
   071.113  376 014    04245        CPI      FF
   071.115  312 166 071 04246        JE       SCOUT10           IS FORM FEED
                       04247
   071.120  365        04248   SCOUT85 PUSH PSW               SAVE CHAR
   071.121  072 343 040 04249   SCOUT9       LDA      S.CDB
   071.124  376 001    04250        CPI      CDB.H84
   071.126  312 146 071 04251        JZ       SCOUT92           IF 8250
                       04252
                       04253   *    HAVE 8251
                       04254
   071.131  333 373    04255   SCOUT91      IN       SC.UART+USR
   071.133  346 001    04256        ANI      USR.TXR
   071.135  312 131 071 04257        JZ       SCOUT91           NOT READY
   071.140  361        04258        POP      PSW
   071.141  323 372    04259        OUT      SC.UART+UDR
   071.143  303 160 071 04260        JMP      SCOUT95
                       04261
                       04262   *    HAVE 8250
                       04263
   071.146  333 355    04264   SCOUT92 IN   SC.ACE+UR.LSR
   071.150  346 040    04265        ANI      UC.THE
   071.152  312 146 071 04266        JZ       SCOUT92
   071.155  361        04267        POP      PSW
   071.156  323 350    04268        OUT      SC.ACE+UR.THR
                       04269
   071.160  376 377    04270   SCOUT95 CPI  377Q              SEE IF TO PAD
   071.161             04271   SCOUTA       EQU      *-1
   071.162  314 204 071 04272        CE       SCDLY             MUST DELAY FOR PADS
   071.165  311        04273        RET                                                      /79.04.GC/
                       04274
                       04275   *    IS FORM FEED
                       04276
   071.166  076 012    04277   SCOUT10 MVI  A,10
   071.170  365        04278   SCOUT11 PUSH PSW               SAVE LINE FEED COUNT
   071.171  076 212    04279        MVI      A,LF+200Q
   071.173  315 345 070 04280        CALL     SCOUT2            OUTPUT LINE FEED
   071.176  361        04281        POP      PSW
   071.177  075        04282        DCR      A
   071.200  302 170 071 04283        JNZ      SCOUT11           MORE TO GO
   071.203  311        04284        RET
```

```
                     04286   **   SCDLY - ISSUE DELAY (VIA 00 CHARACTERS)
                     04287   *
                     04288   *    ENTRY    NONE
                     04289   *    EXIT     NONE
                     04290   *    USES     A,F
                     04291
                     04292
071.204  072 316 077 04293   SCDLY         LDA      CSLDLY
071.207  247         04294   SCDLY1        ANA      A
071.210  310         04295        RZ                          NO MORE PADS
071.211  365         04296        PUSH     PSW
071.212  345         04297        PUSH     H                  SAVE REGISTERS
071.213  257         04298        XRA      A
071.214  315 345 070 04299        CALL     SCOUT2             WRITE PAD
071.217  341         04300        POP      H
071.220  361         04301        POP      PSW
071.221  075         04302        DCR      A
071.222  303 207 071 04303        JMP      SCDLY1             DELAY UNTIL DONE



                     04305   **   CRLF - START NEW LINE.
                     04306   *
                     04307   *    NOTE THAT CRLF DOESNT WANT THE 'LF' TO BE TAKEN AS A
                     04308   *    'NL', AND THUS TRIGGER A RECURSIVE LOOP. WE CAN GET AROUND THAT
                     04309   *    BY SETTING THE PARITY BIT FOR IT, SO THAT IT FAILS THE
                     04310   *    CPI      NO
                     04311   *    TEST. THE PARITY BIT IS STRIPPED (AGAIN, FOR MOST) BEFORE
                     04312   *    THE CHAR IS PASSED TO THE USART.
                     04313   *    ENTRY    NONE
                     04314   *    EXIT     NONE
                     04315   *    USES     A,F
                     04316
                     04317
071.225  345         04318   CRLF PUSH     H                  SAVE (HL)
071.226  076 015     04319        MVI      A,CR
071.230  315 326 070 04320        CALL     SCOUT1
071.233  076 212     04321        MVI      A,LF+200Q
071.235  315 345 070 04322        CALL     SCOUT2             OUTPUT IT
071.240  341         04323        POP      H
071.241  311         04324        RET



                     04326   **   CRM      - Check Raw Mode
                     04327   *
                     04328   *    CRM checks the raw mode flag.  This routine should be
                     04329   *    called before certain character mapping and processint
                     04330   *    is done
                     04331   *
                     04332   *    ENTRY:   S.CSLMD = Current Console Mode
                     04333   *
                     04334   *    EXIT:    PSW      = 'Z'    If NOT Raw Mode
                     04335   *                        'NZ'  if     Raw Mode
                     04336   *
```

```
                     04337   *    USES:    F
                     04338   *
                     04339
   071.242  345      04340   CRM  PUSH     H
   071.243  147      04341        MOV      H,A
   071.244  072 326 040 04342        LDA      S.CSLMD          A = Current Console Mode
   071.247  346 004  04343        ANI      CSL.RAW          Set Flags accordingly
   071.251  174      04344        MOV      A,H              Restore A
   071.252  341      04345        POP      H
   071.253  311      04346        RET
```

```
                     04349   ***  READ - PROCESS READ SYSCALL.
                     04350   *
                     04351   *    READ PROCESSES READ SYSCALLS. IF A SERIAL DEVICE, PASS TO
                     04352   *    DRIVER. IF A STORAGE DEVICE, HANDLE STORAGE MAPPING.
                     04353   *
                     04354   *    MVI      A,CHAN
                     04355   *    LXI      B,COUNT            MUST BE MULTIPLE OF 256
                     04356   *    LXI      D,ADDR
                     04357   *    DB       SYSCALL,.READ      READ DATA FROM CHANNEL
                     04358   *
                     04359   *    ENTRY    (A) = I/O CHANNEL NUMBER
                     04360   *             (BC) = DATA COUNT
                     04361   *             (DE) = ADDRESS FOR DATA
                     04362   *    EXIT     (BC) = COUNT LEFT
                     04363   *             (DE) = NEXT UNUSED ADDRESS
                     04364   *             'C' CLEAR IF ALL OK
                     04365   *             'C' SET IF ERROR
                     04366   *              (A) = ERROR CODE
                     04367   *    USES     ALL
                     04368
                     04369
  071.254  315 256 073 04370   READ CALL     FCI                FETCH CHANNEL INFO
  071.257  330         04371        RC                         ERROR
  071.260  247         04372        ANA      A
  071.261  312 344 031 04373        JZ       ERR.FNO            FILE NOT OPEN
  000.000              04374        ERRNZ    FT.OR-2
  071.264  037         04375        RAR
  071.265  037         04376        RAR
  071.266  322 350 031 04377        JNC      ERR.ILR            ILLEGAL REQUEST
  000.000              04378        ERRNZ    FT.DD-1
  071.271  027         04379        RAL
  071.272  076 000     04380        MVI      A,DC.REA  (A) = DEVICE CODE
  071.274  322 040 041 04381        JNC      AIO.VEC            IF NOT DIRECTORY DEVICE, CALL DRIVER
  071.277  315 107 072 04382        CALL     DIREAD             DIRECTORIED READ
  071.302  303 347 074 04383        JMP      SCI                STORE CHANNEL INFORMATION AND EXIT
```

```
                     04386   ***  WRITE - PROCESS WRITE SCALL.
                     04387   *
                     04388   *    MVI      A,CHAN
                     04389   *    LXI      B,COUNT           MUST BE MULTIPLE OF 256
                     04390   *    LXI      D,ADDR
                     04391   *    DB       SYSCALL,.WRITE    WRITE DATA TO CHANNEL
                     04392   *
                     04393   *    ENTRY    (A) = CHANNEL #
                     04394   *             (BC) = DATA COUNT
                     04395   *             (DE) = ADDRESS
                     04396   *    EXIT     (BC) = COUNT LEFT
                     04397   *             (DE) = NEXT ADDRESS
                     04398   *             'C' CLEAR IF OK
                     04399   *             'C' SET IF ERROR
                     04400   *              (A) = ERROR CODE
                     04401   *    USES     ALL
                     04402
                     04403
  071.305  315 256 073 04404   WRITE         CALL     FCI               FETCH CHANNEL INFORMATION
  071.310  330         04405        RC                          ERROR
  071.311  247         04406        ANA      A
  071.312  312 344 031 04407        JZ       ERR.FNO           FILE NOT OPEN
  071.315  147         04408        MOV      H,A               SAVE COPY IN H
  071.316  346 004     04409        ANI      FT.OW             SEE IF OPEN FOR WRITE
  071.320  312 350 031 04410        JZ       ERR.ILR           ILLEGAL REQUEST
  071.323  174         04411        MOV      A,H
  000.000              04412        ERRNZ    FT.DD-1
  071.324  037         04413        RAR
  071.325  076 001     04414        MVI      A,DC.WRI  REQUEST WRITE
  071.327  322 040 041 04415        JNC      AIO.VEC           NOT A DIRECTORY DEVICE
  071.332  315 167 072 04416        CALL     DIWRITE           DIRECTORIED WRITE
  071.335  303 347 074 04417        JMP      SCI               STORE CHANNEL INFO
```

```
                     04420   ***  PRINT - PRINT CONSOLE LINE.
                     04421   *
                     04422   *    PRINT CAUSES A CODED LINE TO BE PRINTED AT THE CONSOLE.
                     04423   *
                     04424   *    LXI      H,LINEADDR
                     04425   *    DB       SYSCALL,.PRINT
                     04426   *
                     04427   *    THE LAST CHARACTER IN THE LINE SHOULD HAVE THE 200Q BIT SET.
                     04428   *
                     04429   *    ENTRY    (HL) = LINE ADDRESS
                     04430   *    EXIT     (HL) = LWA OF MESSAGE +1
                     04431   *    USES     A,F,H,L
                     04432
                     04433
071.340  176         04434   PRINT    MOV      A,M              (A) = CODE
071.341  346 177     04435            ANI      177Q             CLEAR FLAG BIT
071.343  315 304 070 04436            CALL     SCOUT            TYPE IT
071.346  276         04437            CMP      M
071.347  043         04438            INX      H
071.350  312 340 071 04439            JE       PRINT            NOT 200Q SET
071.353  311         04440            RET
```

```
                     04443   ***  CONSL - SET AND CLEAR CONSOLE FLAGS.
                     04444   *
                     04445   *    CONSL IS CALLED TO SET, CLEAR, OR READ BITS IN THE VARIOUS
                     04446   *    CONSOLE FLAGS.
                     04447   *
                     04448   *    THE CALLER PASSES AN INDES INTO THE PROPER FLAG, A
                     04449   *    MASK TO INDICATE THE EFFECTED BITS, AND A SET OF NEW VALUES
                     04450   *    FOR THOSE BITS.
                     04451   *
                     04452   *    INDEX =
                     04453   *
                     04454   *    O        S.CSLMD
                     04455   *    1        S.CONTY
                     04456   *    2        S.CUSOR
                     04457   *    3        S.CONWI
                     04458   *    4        S.CONFL
                     04459   *
                     04460   *    ENTRY    (A) = INDEX
                     04461   *             (B) = NEW VALUES
                     04462   *             (C) = MASK ('1' BIT FOR EVERY BIT TO CHANGE)
                     04463   *    EXIT     'C' CLEAR IF NO ERROR
                     04464   *              (A) = NEW VALUE
                     04465   *             'C' SET IF ERROR
                     04466   *              (A) = ERROR CODE
                     04467   *    USES     ALL
                     04468
                     04469
  071.354            04470   CONSL         EQU      *
  071.354  376 005   04471         CPI      5
  071.356  322 350 031 04472         JNC      ERR.ILR            ILLEGAL REQUEST
  071.361  041 326 040 04473         LXI      H,S.CSLMD
  071.364  315 101 030 04474         CALL     $DADA.             (HL) = ADDRESS FOR BYTE
  071.367  171         04475         MOV      A,C
  071.370  240         04476         ANA      B                  CLEAR (B) TO THE BITS TO BE ALTERED
  071.371  107         04477         MOV      B,A
  071.372  171         04478         MOV      A,C
  071.373  057         04479         CMA                         (A) = -MASK
  071.374  363         04480         DI                          INTERLOCK CONSOLE
  071.375  246         04481         ANA      M                  CLEAR EFFECTED BITS
  071.376  260         04482         ORA      B                  SET NEW VALUES
  071.377  167         04483         MOV      M,A                REPLACE
  072.000  373         04484         EI
  072.001  311         04485         RET
```

```
                     04488   ***  CLRCO - CLEAR CONSOLE BUFFERS.
                     04489   *
                     04490   *    CLRCO CLEARS THE CONSOLE TYPE-AHEAD BUFFER.
                     04491   *    CTL-O AND CTL-S FLAGS ARE ALSO CLEARED.
                     04492   *
                     04493   *    ENTRY    NONE
                     04494   *    EXIT     NONE
                     04495   *    USES     ALLL
                     04496
                     04497
  072.002  363       04498   CLRCO         DI
  072.003  041 324 077 04499         LXI      H,CSLIBUF
  072.006  042 306 077 04500         SHLD     SCIIN
  072.011  042 310 077 04501         SHLD     SCIOUT            CLEAR POINTER
  072.014  257       04502         XRA      A
  072.015  062 302 077 04503         STA      CSLLCNT           CLEAR LINE COUNT
  072.020  062 304 077 04504         STA      CSLRBF            CLEAR RUBOUT BUFFER
  072.023  062 332 040 04505         STA      S.CONFL           CLEAR CTL-O AND CTL-S
  072.026  373       04506         EI
  072.027  311       04507         RET
```

```
                     04510   ***  LOADO     - LOAD SPECIFIED OVERLAY
                     04511   *
                     04512   *    LOADO LOADS THE OVERLAY SPECIFIED THROUGH THE INDES.
                     04513   *
                     04514   *            OVERLAY        INDEX
                     04515   *          ----------------------
                     04516   *            HDOSOVL           0
                     04517   *            HDOSOVL2 1
                     04518   *
                     04519   *    ************************************************************************
                     04520   *    *                                                                      *
                     04521   *    *        NOTE:    THIS CALL SHOULD NOT BE MADE FROM ANOTHER OVERLAY    *
                     04522   *    *                 UNLESS IT IS THE OVERLAY TO BE LOADED                *
                     04523   *    *                                                                      *
                     04524   *    ************************************************************************
                     04525   *
                     04526   *    ENTRY:   (A)       = OVERLAY INDEX
                     04527   *
                     04528   *    EXIT:    (PSW)     ='C' CLEAR IF NO ERRORS
                     04529   *                         'C' SET   IF    ERRORS
                     04530   *                          (A)  = ERROR CODE
                     04531   *
                     04532   *    USES:    ALL
                     04533   *
                     04534
  072.030  365       04535   LOADO          PUSH     PSW               SAVE THE OVERLAY INDEX
  072.031  315 265 065 04536        CALL     LDON              LOAD THE SPECIFIED OVERLAY
  072.034  332 100 072 04537        JC       LOADO2            ERROR
  072.037  072 371 040 04538        LDA      S.OVLFL
  072.042  346 200     04539        ANI      OVL.UCS
  072.044  302 076 072 04540        JNZ      LOADO1            USER CODE IS SWAPPED
  072.047  361         04541        POP      PSW               RESTORE OVERLAY INDEX
  072.050  315 262 074 04542        CALL     OTI               OVERLAY TABLE INDEX
  072.053  006 000     04543        DW       OVL.FLB           (HL) = ADDRESS OF FLAG BYTE
  072.055  176         04544        MOV      A,M
  072.056  346 002     04545        ANI      OVL.RES
  072.060  300         04546        RNZ                        IT IS ALREADY RESIDENT
  072.061  176         04547        MOV      A,M
  072.062  366 002     04548        ORI      OVL.RES
  072.064  167         04549        MOV      M,A               FLAG OVERLAY AS PERM RESIDENT
  072.065  053         04550        DCX      H
  072.066  053         04551        DCX      H                 (HL) = OVERLAY ENTRY POINT
  000.000               04552        ERRNZ    OVL.FLB-OVL.ENT-2
  072.067  315 211 030 04553        CALL     $HLIHL            (HL) = ENTRY ADDRESS
  072.072  042 320 040 04554        SHLD     S.SYSM            SET OVERLAY ENTRY POINT AS HDOS LOWER BOUND
  072.075  311         04555        RET
                       04556
  072.076  076 021     04557   LOADO1         MVI      A,EC.NEM NOT ENOUGH MEMORY
  072.100  067         04558   LOADO2         STC                        FLAG ERROR
  072.101  341         04559        POP      H                 REMOVE SAVED OVERLAY INDEX
  072.102  311         04560        RET
```

```
                     04563   **   VERSN    - RETURN HDOS VERSION NUMBER'
                     04564   *
                     04565   *    VERSN RETURNS THE HDOS VERSION NUMBER AS A ONE BYTE BCD NUMBER.
                     04566   *    A DECIMAL IS ASSUMED BETWEEN THE HIGH AND LOW ORDER NIBBLES.
                     04567   *
                     04568   *
                     04569   *    ENTRY    NONE
                     04570   *
                     04571   *    EXIT     (PSW)    = (A) = VERSION NUMBER
                     04572   *
                     04573   *    USES     (PSW)
                     04574   *
                     04575
  072.103  076 040    04576   VERSN        MVI      A,VERS
  072.105  247        04577        ANA      A                CLEAR CARRY
  072.106  311        04578        RET
```

```
                     04581   **   DIREAD - DIRECTORIED READ.
                     04582   *
                     04583   *    DIREAD REASD THE SPECIFIED NUMBER OF SECTORS FROM A
                     04584   *    DIRECTORIED DEVICE. THE DATA IS RAD FROM THE CURRENT
                     04585   *    FILE POSITION.
                     04586   *
                     04587   *    ENTRY    (B) = SECTOR COUNT
                     04588   *             (C) = 9
                     04589   *             (DE) = ADDRESS FOR DATA
                     04590   *             AIO.XXX SETUP
                     04591   *    EXIT     (BC) = COUNT LEFT
                     04592   *             (DE) = NEXT FREE ADDRESS
                     04593   *             'C' CLEAR IF OK
                     04594   *             'C' SET OF ERROR
                     04595   *              (A) = CODE
                     04596   *    USES     ALL
                     04597
                     04598
  072.107            04599   DIREAD       EQU      *
  072.107  170      04600        MOV      A,B
  072.110  247      04601        ANA      A
  072.111  310      04602        RZ                        NOTHING TO READ
  072.112  325      04603        PUSH     D                SAVE (DE)
  072.113  315 002 032 04604     CALL     DCA              DETERMINE CONTINUOUS AREA
  072.116  321      04605        POP      D
  072.117  072 113 041 04606     LDA      AIO.EOF
  072.122  037      04607        RAR
  072.123  330      04608        RC                        EXIT IF EOF
  072.124  305      04609        PUSH     B
  072.125  315 145 033 04610     CALL     PDI              PREPARE DEVICE I/O
  000.000            04611        ERRNZ    DC.REA
  072.130  315 140 072 04612     CALL     DIREAD1          PERFORM I/O
  072.133  301      04613        POP      B
  072.134  322 107 072 04614     JNC      DIREAD           IF NOT ERROR
  072.137  311      04615        RET



                     04617   **   DIREAD1 - PERFORM I/O
                     04618   *
                     04619   *    DIREAD1 CALLS THE I/O DRIVER, AFTER COMPUTING THE COMPLETION ADDRESS
                     04620   *    (WHICH THE DRIVER WILL NOT RETURN)
                     04621   *
                     04622   *    ENTRY    (A) = OPERATION CODE
                     04623   *             (BC) = COUNT
                     04624   *             (DE) = ADDRESS
                     04625   *             (HL) = SECTOR NUMBER
                     04626   *    EXIT     (PSW) AS FOM DRIVER
                     04627   *             (BC) AS FROM DRIVER
                     04628   *             (DE) = (BC ON ENTRY) + (DE ON ENTRY)
                     04629   *             (HL) AS FROM DRIVER
                     04630   *    USES     ALL
                     04631
                     04632
  072.140  353      04633   DIREAD1 XCHG                   (HL) = I/O ADDRESS
  072.141  345      04634        PUSH     H                SAVE
  072.142  011      04635        DAD      B                (HL) = NEW ADDRESS
```

```
   072.143  343         04636        XTHL                       (HL) = I/O ADDRESS ((SP)) = NEW ADDRESS
   072.144  353         04637        XCHG                       RESTORE AS UPON ENTRY
   072.145  315 040 041 04638        CALL     AIO.VEC           CALL DRIVER
   072.150  321         04639        POP      D                 (DE) = NEW ADDRESS
   072.151  311         04640        RET



                        04642   **   DIWRITE - DIRECTORY DEVICE WRITE.
                        04643   *
                        04644   *    DIWRITE WRITES THE SPECIFIED NUMBER OF SECTORS TO A DIRECTORIED
                        04645   *    DEVICE.
                        04646   *
                        04647   *    ENTRY    (B) = COUNT
                        04648   *             (C) = 0
                        04649   *             (DE) = TEXT ADDRESS
                        04650   *             AIO.XXX SETUP
                        04651   *    EXIT     (BC) = COUNT LEFT
                        04652   *             (DE) = ADDRESS
                        04653   *             'C' CLEAR, IF OK
                        04654   *             'C' SET IF ERROR
                        04655   *              (A) = ERROR CODE
                        04656   *    USES ALL
                        04657
                        04658
   072.152  305         04659   DWRIT1   PUSH     B                 SAVE COUNT
   072.153  315 145 033 04660        CALL     PDI               PREPARE FOR DEVICE I/O
   072.156  076 001     04661        MVI      A,DC.WRI
   072.160  315 140 072 04662        CALL     DIREAD1           PERFORM I/O
   072.163  301         04663        POP      B                 (BC) = COUNT LEFT
   072.164  076 023     04664        MVI      A,EC.WF           WRITE FAIL (IF CARRY SET)
   072.166  330         04665        RC                         RETURN IF ERROR
                        04666
   072.167              04667   DIWRITE EQU   *
   072.167  170         04668        MOV      A,B
   072.170  247         04669        ANA      A
   072.171  310         04670        RZ                         NO MORE
   072.172  325         04671        PUSH     D
   072.173  315 002 032 04672        CALL     DCA               DETERMINE CONTIGUOUS AREA
   072.176  321         04673        POP      D
   072.177  072 113 041 04674        LDA      AIO.EOF
   072.202  037         04675        RAR
   072.203  322 152 072 04676        JNC      DWRIT1            IF NOT EOF
                        04677
                        04678   *    MUST APPEND SECTORS TO END OF THE FILE.
                        04679   *    ALLOCATE THE SPACE.
                        04680
   072.206              04681   DWRIT2   EQU      *
   072.206  170         04682        MOV      A,B
   072.207  247         04683        ANA      A
   072.210  310         04684        RZ                         NO MORE
   072.211  052 116 041 04685        LHLD     AIO.CHA
   072.214  076 037     04686        MVI      A,IOC.DIR+DIR.FLG-IOC.DDA
   072.216  315 101 030 04687        CALL     $DADA.            (HL) = #DIR.FLG IN CHANNEL
   072.221  176         04688        MOV      A,M
   072.222  346 357     04689        ANI      377Q-DIF.CNT      IS NOT CONTIGUOUS ANY MORE (IF IT EVER WAS)
```

```
   072.224  167         04690        MOV      M,A
   072.225  325         04691        PUSH     D
   072.226  315 267 072 04692        CALL     ACA               ALLOCATE CONTINUOUS AREA
   072.231  321         04693        POP      D
   072.232  072 112 041 04694        LDA      AIO.EOM
   072.235  037         04695        RAR
   072.236  330         04696        RC                         EXIT IF EOM
                        04697
                        04698   *    NOT OUT OF SPACE. WRITE IT
                        04699
   072.237  305         04700        PUSH     B
   072.240  315 145 033 04701        CALL     PDI               PREPARE DEVICE I/O
   072.243  076 001     04702        MVI      A,DC.WRI
   072.245  315 140 072 04703        CALL     DIREAD1           PERFORM I/O
   072.250  301         04704        POP      B
   072.251  322 206 072 04705        JNC      DWRIT2            GO AGAIN
   072.254  311         04706        RET                        RETURN WITH ERROR CODE
```

```
                     04709   **   ABP - ADVANCE BUFFER POINTERS.
                     04710   *
                     04711   *    ABP ADVANCES THE BUFFER POINTER TO THE NEXT BYTE. IF THE
                     04712   *    POINTER OVERFLOWS, IT IS WRAPPED.
                     04713   *
                     04714   *    ENTRY    (HL) = POINTER
                     04715   *    EXIT     (HL) = POINTER TO NEXT
                     04716   *    USES     A,F,H,L
                     04717
                     04718
072.255  043       04719   ABP  INX      H                 INCREMENT
072.256  072 314 077 04720        LDA      SCILWA
072.261  275       04721        CMP      L
072.262  300       04722        RNE                        NOT OVER END
072.263  052 312 077 04723        LHLD     SCIFWA
072.266  311       04724        RET



                     04726   **   ACA - ALLOCATE CONTINUOUS AREA
                     04727   *
                     04728   *    ACA IS CALLED TO APPEND SECTORS TO THE END OF A FILE.
                     04729   *    IT ALLOCATES AS MANY CONTINUOUS SECTORS AS IT CAN UNTIL
                     04730   *    ENOUGH ARE ALLOCATED, OR A BREAK IN THE CONTINUITY IS REQUIRED.
                     04731   *
                     04732   *    FIRST, THE REMAINING SECTORS IN THE GROUP ARE USED.
                     04733   *    2ND, ACA ATTEMPTS TO OBTAIN THE IMMEDIATELY FOLLOWING GROUP.
                     04734   *    3RD, ACA TRYS TO LOCATE A VIRGIN CLUSTER
                     04735   *    4TH, ACA TAKES ANY FREE GROUPS.
                     04736   *
                     04737   *    ENTRY    (B) = SECTOR COUNT
                     04738   *             AIO.XXX SETUP
                     04739   *    EXIT     (B) = SECTORS NOT ALLOCATED
                     04740   *             AIO.CNT = AMOUNT ALLOCATED
                     04741   *             AIO.EOM = EC.EOM*2+1 IF END OF MEDIA
                     04742   *             AIO.LGN, AIO.LST UPDATED FOR ADDITIONS
                     04743   *             AIO,CGN, AIO.CSI = AIO.LGN, AIO.LSI
                     04744   *             AIO.TFP = SETUP WITH GROUP AND INDEX OF START OF AREA
                     04745   *    USES     ALL
                     04746
                     04747
072.267  016 000     04748   ACA  MVI      C,0               (C) = COUNT ALLOCATED
072.271  052 051 041 04749   ACA0 LHLD     AIO.LGN           (L) = AIO.LGN, (H) = AIO.LSI
000.000              04750        ERRNZ    AIO.LSI-AIO.LGN-1
072.274  042 114 041 04751        SHLD     AIO.TFP           SAVE WRITE ADDRESS
                     04752
072.277  041 052 041 04753   ACA1 LXI      H,AIO.LSI(M) = LAST SECTOR INDEX
072.302  072 046 041 04754        LDA      AIO.SPG           (A) = SECTORS PER GROUP
072.305  226       04755        SUB      M                 (A) = SECTORS LEFT IN GROUP
072.306  312 332 072 04756        JZ       ACA3              NONE LEFT
072.311  270       04757        CMP      B
072.312  332 316 072 04758        JC       ACA2              NOT TOO MANY IN GROUP FOR NEED
072.315  170       04759        MOV      A,B               DONT TAKE MORE THAN WE NEED
072.316  127       04760   ACA2 MOV      D,A               (D) = AMOUNT IN GROUP
072.317  206       04761        ADD      M
072.320  167       04762        MOV      M,A               ADVANCE AIO.LSI
```

```
072.321  172          04763        MOV      A,D               (A) = AMOUNT ALLOCATED FROM GROUP
072.322  201          04764        ADD      C
072.323  117          04765        MOV      C,A               ADVANCE TOTAL ALLOCATED COUNT
072.324  170          04766        MOV      A,B
072.325  222          04767        SUB      D                 DECREMENT NEEDED COUNT
072.326  107          04768        MOV      B,A
072.327  312 374 072  04769        JZ       ACA9              GOT ALL WE NEED
                      04770
                      04771   *    FINISHING THE GROUP WASENT ENOUGH. TRY TO GET THE FOLLOWING
                      04772   *    GROUP.
                      04773
072.332  056 051      04774   ACA3 MVI      L,#AIO.LGN        (HL) = #AIO.LGN
000.041               04775   .    SET      AIO.LGN/256
000.000               04776        ERRNZ    AIO.LSI/256-.     MUST BE IN SAME PAGE
072.334  126          04777        MOV      D,M               (D) = AIO.LGN
072.335  024          04778        INR      D                 (D) = FOLLOWING GROUP #
072.336  315 133 032  04779        CALL     FFB               FIND FREE BLOCK
072.341  332 362 072  04780        JC       ACA8              END OF MEDIA
072.344  302 374 072  04781        JNZ      ACA9              COULDNT GET ONE CONTIGUOUS
                      04782
                      04783   *    GOT A BLOCK. CHAIN IT TO THE FILE
                      04784
072.347  315 354 031  04785        CALL     CFF               CHAIN FREE BLOCK TO FILE
072.352  171          04786        MOV      A,C
072.353  247          04787        ANA      A
072.354  312 271 072  04788        JZ       ACA0              AM STILL LOOKING FOR THE START
072.357  303 277 072  04789        JMP      ACA1              GO SOME MORE
                      04790
                      04791
                      04792   **   END OF MEDIA EXIT. FLAG EOM IF COULDNT ALLOCATE ANY
                      04793
072.362  171          04794   ACA8 MOV      A,C
072.363  247          04795        ANA      A
072.364  302 374 072  04796        JNZ      ACA9              GIVE HIM WHAT WE DID GET, ANYWAY...
072.367  076 005      04797        MVI      A,EC.EOM*2+1
072.371  062 112 041  04798        STA      AIO.EOM           SET EOM
                      04799
                      04800   **   NORMAL EXIT. (C) = AMOUNT ALLOCATED
                      04801
072.374  171          04802   ACA9 MOV      A,C
072.375  062 111 041  04803        STA      AIO.CNT           SET COUNT
073.000  052 051 041  04804        LHLD     AIO.LGN
073.003  042 047 041  04805        SHLD     AIO.CGN           UPDATE CURRENT=LAST
000.000               04806        ERRNZ    AIO.LSI-AIO.LGN-1
000.000               04807        ERRNZ    AIO.CSI-AIO.CGN-1
073.006  311          04808        RET



                      04810   **   BND      - Boot New Disk                                    /2.0a/
                      04811   *
                      04812   *    BND boots a new disk when it is time to re-boot
                      04813   *
                      04814   *    ENTRY:   NONE
                      04815   *
                      04816   *    EXIT:    To FATSERR if Error
```

```
                     04817   *
                     04818   *    USES:    ALL
                     04819   *
                     04820
   073.007  377 007     04821   BND  SCALL    .CLRCO             Cleary Type-Ahead
   073.011  315 136 031 04822        CALL     $TYPTX
   073.014  012         04823        DB       NL
   073.015  111 156 163 04824        DB       'Install a bootable disk in SY0:.  Hit RETURN to reboot:'
   073.104  240         04825        DB       ' '+200Q
                     04826
   073.105  377 001     04827   BND1 SCALL    .SCIN              Wait for a character
   073.107  332 105 073 04828        JC       BND1
   073.112  376 012     04829        CPI      NL                 Wait for a Newline
   073.114  302 105 073 04830        JNZ      BND1
                     04831
   073.117  257         04832   BND. XRA      A
   073.120  062 061 041 04833        STA      AIO.UNI            Force Unit 0
                     04834
   073.123  076 012     04835   BND2 MVI      A,DC.RDY
   073.125  315 130 040 04836        CALL     SYDD
   073.130  332 123 073 04837        JC       BND2               Wait for device to go ready
                     04838
   073.133  041 000 000 04839        LXI      H,0
   073.136  076 010     04840        MVI      A,DC.MOU
   073.140  315 130 040 04841        CALL     SYDD               Mount the possibly new volume
   073.143  334 115 066 04842        CC       FATSERR
                     04843
   073.146  001 000 011 04844        LXI      B,DDF.BOL*256      BC = Boot Code Maximum Length
   073.151  021 200 042 04845        LXI      D,SB.BOO  DE = Boot Code address
   073.154  041 000 000 04846        LXI      H,DDF.BOOHL = Boot SEctor
   073.157  076 002     04847        MVI      A,DC.RER Read Regardless
   073.161  315 130 040 04848        CALL     SYDD
   073.164  334 115 066 04849        CC       FATSERR
                     04850
   073.167  072 321 077 04851        LDA      SUNIT
   073.172  062 061 041 04852        STA      AIO.UNI  Restore Physical Unit Number
                     04853
                     04854   *    Restore original vectors for old versions of HDOS               /2.0c/
   073.175  001 130 000 04855        LXI      B,BOOTAL
   073.200  021 132 037 04856        LXI      D,BOOTA
   073.203  041 110 040 04857        LXI      H,D.CON
   073.206  315 252 030 04858        CALL     $MOVE              Move the ROM vectors into RAM
                     04859
   073.211  257         04860        XRA      A                  Clear Console
   073.212  323 351     04861        OUT      SC.ACE+UR.IER
   073.214  323 373     04862        OUT      SC.UART+USR
   073.216  311         04863        RET



                     04865   **   BTS      - Block to Sector                              /80.06.gc/
                     04866   *
                     04867   *    Convert a block number to the corresponding sector
                     04868   *    number.
                     04869   *
                     04870   *    ENTRY:   A        = Sectors Per Group
```

```
                     04871   *             HL       = Block Number
                     04872   *
                     04873   *    EXIT:    HL       = Sector Number
                     04874   *
                     04875   *    USES:    HL
                     04876   *
                     04877
   073.217  365      04878   BTS  PUSH     PSW
   073.220  325      04879        PUSH     D
   073.221  353      04880        XCHG                        DE = Block Number
   073.222  315 007 031 04881     CALL     $MU86              HL = A * DE
   073.225  321      04882        POP      D
   073.226  361      04883        POP      PSW
   073.227  311      04884        RET



                     04886   **   CPA- CHECK FOR PENDING ABORT.
                     04887   *
                     04888   *    CPA IS CALLED WHEN A CONSOLE ABORT MAY BE PROCESSED.
                     04889   *    IF THE SYSTEM IS READY, AND AN ABORT
                     04890   *    IS PENDING, PROCESS IT.
                     04891   *
                     04892   *    CPA SHOULD BE CALLED WITH INTERRUPTS DISABLED, SO THAT
                     04893   *    ANOTHER INTERRUPT CHARACTER CANNOT OCCUR DURING CPA PROCESSING.
                     04894   *    THIS GUARANTEES THAT A USER PROGRAM WILL BE 'INTERRUPTED' WITH
                     04895   *    THE PROGRAM COUNTER IN THE USER CODE, NEVER IN HDOS CODE.
                     04896   *
                     04897   *    UPON ENTRY TO THE USER INTERRUPT ROUTINE,
                     04898   *
                     04899   *    ((SP)+0) = RETURN ADDRESS (IF USER WISHES TO RESUME NORMAL PROCESSING)
                     04900   *    ((SP)+2) = USER PSW
                     04901   *    ((SP)+4) = USER INTERRUPTED ADDRESS
                     04902   *
                     04903   *    THE USER REGISTER VALUES FOR B,C,D,E,H, AND L ARE STILL
                     04904   *    IN THE REGISTERS.
                     04905   *
                     04906   *    ENTRY    ((SP+0) = RETURN ADDRESS
                     04907   *             ((SP+2) = USER PSW
                     04908   *             ((SP+4) = USER INTERRUPTED ADDRESS
                     04909   *    EXIT     TO *RED* IF NONE, OR DISABLED
                     04910   *             TO PROCESSOR IF READY AND OK
                     04911   *    USES     A,F
                     04912
                     04913
                     04914
   073.230  072 300 077 04915   CPA  LDA      SYSMODE
   073.233  247      04916        ANA      A
   073.234  300      04917        RNZ                         IN SYSCALL MODE
                     04918
                     04919   *    WILL ALLOW PROCESSING
                     04920
   073.235  072 334 040 04921        LDA      S.CAADR+1(A) = HIGH BYTE ABORT ADDRESS
   073.240  247      04922        ANA      A
   073.241  310      04923        RZ                          NO ABORT PENDING
                     04924
```

```
                     04925   *    HAVE ABORT. PROCESS IT
                     04926
   073.242  345      04927        PUSH     H
   073.243  052 333 040 04928     LHLD     S.CAADR           (HL) = ADDRESS FOR JUMP
   073.246  257      04929        XRA      A
   073.247  062 334 040 04930     STA      S.CAADR+1 CLEAR
   073.252  074      04931        INR      A                 SET (A) <>.
   073.253  343      04932        XTHL                       RESTORE (HL), SET PROCESSOR
   073.254  373      04933        EI
   073.255  311      04934        RET                        ENTER ROUTINE



                     04936   **   FCI - FETCH CHANNEL INFORMATION.
                     04937   *
                     04938   *    FCI COPIES THE CHANNEL INFORMATION FROM THE
                     04939   *    CHANNEL TABLE INTO THE ACTIVE I/O TABLE.
                     04940   *
                     04941   *    AIO.VEC = DRIVER ADDRESS
                     04942   *    AIO.XXX SETUP IF DIRECTORY DEVICE
                     04943   *    AIO.CTA = ADDRESS OF CHANNEL AREA
                     04944   *
                     04945   *    ENTRY    (A) = CHANNEL NUMBER
                     04946   *    EXIT     (A) = CHANNEL STATUS BYTE
                     04947   *             (HL) = ADDRESS OF FILE STATUS BYTE
                     04948   *             'C' SET OF ERROR
                     04949   *              (A) = ERROR CODE
                     04950   *    USES     A,F,H,L
                     04951
                     04952
   073.256  052 352 040 04953   FCI  LHLD     S.CFWA            (HL) = CHANNEL TABLE FWA
   000.000           04954        ERRNZ    IOCCTD-1  CHANNEL 377Q IS FIRST IN LIST
   073.261  074      04955        INR      A                 (A) = INDEX OF CHANNEL IN CHANTAB
   073.262  365      04956        PUSH     PSW               SAVE INDEX
   073.263  361      04957   FCI1 POP      PSW               (A) = INDEX
   073.264  247      04958        ANA      A
   073.265  312 307 073 04959        JZ       FCI2              GOT IT
   073.270  075      04960        DCR      A                 DECREMENT COUNT
   073.271  365      04961        PUSH     PSW               SAVE INDEX
   073.272  176      04962        MOV      A,M
   073.273  043      04963        INX      H
   073.274  146      04964        MOV      H,M
   073.275  157      04965        MOV      L,A               FOLLOW LINK
   073.276  264      04966        ORA      H
   073.277  302 263 073 04967        JNZ      FCI1              MORE TO FOLLOW
                     04968
                     04969   *    CHANNEL DOES NOT EXIST. FLAG ERROR
                     04970
   073.302  361      04971        POP      PSW
   073.303  076 016   04972        MVI      A,EC.ICN  ILLEGAL CHANNEL NUMBER
   073.305  067      04973        STC
   073.306  311      04974        RET
                     04975
                     04976   *    GOT CHANNEL
                     04977
   073.307  305      04978   FCI2 PUSH     B
```

```
   073.310  325          04979        PUSH     D                  SAVE REGISTERS
   073.311  315 066 075  04980        CALL     $INDLB             A  = UNIT CODE                      /80.02.GC/
   073.314  022 000      04981        DW       IOC.UNI                                                /80.02.GC/
   073.316  062 061 041  04982        STA      AIO.UNI            INSURE UNIT SET UP FOR SEQ. /80.02.GC/
   073.321  043          04983        INX      H
   073.322  043          04984        INX      H                  MOVE PAST LINK
   000.000               04985        ERRNZ    IOC.DDA-2 POINT TO DDA
   073.323  042 116 041  04986        SHLD     AIO.CHA            SET BLOCK ADDRESS
   073.326  043          04987        INX      H
   073.327  043          04988        INX      H
   000.000               04989        ERRNZ    IOC.FLG-IOC.DDA-2  (HL) = #IOC.DDA
   073.330  345          04990        PUSH     H                  SAVE ADDRESS
   073.331  176          04991        MOV      A,M                (A) = TYPE
   073.332  346 001      04992        ANI      FT.DD              SEE IF DIRECTORY TREE
   073.334  053          04993        DCX      H
   073.335  053          04994        DCX      H
   000.000               04995        ERRNZ    IOC.DDA-IOC.FLG+2  (HL) = #IOC.DDA
   073.336  353          04996        XCHG
   073.337  041 041 041  04997        LXI      H,AIO.DDA
   073.342  001 003 000  04998        LXI      B,IOC.SQL
   073.345  312 352 073  04999        JZ       FCI3               IS SEQUENTIAL
   073.350  016 021      05000        MVI      C,IOC.DIL IS DIRECTORY
   073.352  315 252 030  05001   FCI3 CALL     $MOVE              MOVE DATA
   073.355  341          05002        POP      H
   073.356  176          05003        MOV      A,M                A = FLAG
   073.357  321          05004        POP      D
   073.360  301          05005        POP      B
   073.361  311          05006        RET



                         05008   **   GSP      GET SYSTEM POINTER
                         05009   *
                         05010   *    GET THE SYSTEM POINTER
                         05011   *
                         05012   *
                         05013   *    ENTRY:   GSPA     = SY: device table entry address     /80.06.gc/
                         05014   *
                         05015   *    EXIT:    HL       = SYSTEM DEVICE UNIT POINTER
                         05016   *
                         05017   *    USES:    PSW,HL
                         05018   *
                         05019
   073.362  052 377 073  05020   GSP  LHLD     GSPA               HL = SY: device table entry /80.06.gc/
   073.365  325          05021        PUSH     D
   073.366  021 011 000  05022        LXI      D,DEV.UNT HL = POINTER TO UNIT TABLE POINTER
   073.371  031          05023        DAD      D
   073.372  321          05024        POP      D
   073.373  257          05025        XRA      A                                                      /80.05.gc/
                         05026   *    LDA      SUNIT              A  = system Unit            /80.05.gc/
   073.374  303 035 075  05027        JMP      GUP
                         05028
   073.377  000 000      05029   GSPA DW       0                  Saved SY: device table entry        /80.06.gc/
```

```
                     05031   **   LDD - LOAD DEVICE DRIVER.
                     05032   *
                     05033   *    LDD IS CALLED TO PERFORM THE SUSPENDED LOAD OF A DEVICE DRIVER.
                     05034   *
                     05035   *    IF SOME OVL CODE WISHES TO LOAD A DEVICE DRIVER, IT MUST
                     05036   *    SUSPEND THE REQUEST, SINCE THE DEVICE DRIVER WILL OVERLAY THE
                     05037   *    OVL CODE. AFTER THE OVL CODE EXITS, THE RESIDENT CODE WILL CALL
                     05038   *    LDD TO PERFORM THE ACTUAL LOAD, OVER THE OVL.
                     05039   *
                     05040   *
                     05041   *    This code has been modified to reduce the dependencies
                     05042   *    on a cluster size of 2.  the GRT address and SPG are
                     05043   *    not dynamically computed so that the mount of SY0: may
                     05044   *    be kludged by pre-stuffing them.
                     05045   *                                        G. Chandler       80.06.19
                     05046   *
                     05047   *
                     05048   *    ENTRY    DD.IOC = POINTER TO IOC.DDA
                     05049   *             DD.LDA = LOAD ADDRESS
                     05050   *             DD.LEN = LOAD LENGTH
                     05051   *             DD.GRP = SECTOR INDEX ON SYSTEM DEVICE
                     05052   *             DD.DTA = DEV.RES ADDRESS
                     05053   *             DD.OPE = OPEN CODE (DC.OPR, DC.OPW, DC.OPU)
                     05054   *
                     05055   *             LDD8A    = system GRT address                          /80.06.GC/
                     05056   *             LDD8B    = system SPG                                  /80.06.GC/
                     05057   *
                     05058   *    EXIT     OVL CODE DESTROYED
                     05059   *    USES     NONE
                     05060
                     05061
  074.001  315 054 031 05062   LDD  CALL     $SAVALL           SAVE REGS
                     05063
                     05064   *    CLEAR OVL RESIDENT FLAG
                     05065
  074.004  041 371 040 05066        LXI      H,S.OVLFL
  074.007  176         05067        MOV      A,M
  074.010  346 376     05068        ANI      377Q-OVL.IN
  074.012  167         05069        MOV      M,A               CLEAR IN FLAG
                     05070
                     05071   *    LOAD OVERLAY
                     05072
  074.013  072 206 074 05073        LDA      LDD8B             LDD8B = System SPG          /80.06.gc/
  074.016  346 376     05074        ANI      376Q                                          /80.06.GC/
  074.020  017         05075        RRC                        A  = 512 byte reads/group   /80.06.gc/
  074.021  062 261 074 05076        STA      LDD9B             Initialize Current Set Index  /80.06.gc/
  074.024  062 260 074 05077        STA      LDD9A             Initialize Sets/group         /80.06.gc/
  074.027  052 364 040 05078        LHLD     S.DDGRP                                         /80.06.GC/
  074.032  046 000     05079        MVI      H,0               HL = driver group number    /80.06.gc/
  074.034  072 206 074 05080        LDA      LDD8B             A  = SPG                    /80.06.gc/
  074.037  315 217 073 05081        CALL     BTS               HL = sector number          /80.06.gc/
  074.042  315 211 074 05082        CALL     LDD9              ignore parameters           /80.06.gc/
  000.000              05083        ERRNZ    DVD.ENT-2000A                                   /80.06.GC/
  074.045  042 207 074 05084        SHLD     LDD8C             Safe it for later           /80.06.gc/
                     05085
  074.050  052 362 040 05086        LHLD     S.DDLEN           (HL) = LENGTH
  074.053  104         05087        MOV      B,H
```

```
   074.054  115          05088        MOV      C,L               (BC) = LENGTH
   074.055  052 360 040  05089        LHLD     S.DDLDA           (HL) = LOAD ADDRESS
   074.060  345          05090        PUSH     H                 SAVE FOR LATER
   074.061  353          05091        XCHG
   074.062  041 125 102  05092        LXI      H,SECSCR+511      FORCE NEW DISK READ RIGHT AWAY
                         05093
                         05094   *    LOAD BINARY
                         05095
   074.065  315 136 074  05096   LDD2 CALL     LDD8              FIND NEXT BYTE
   074.070  176          05097        MOV      A,M               (A) = NEXT BYTE
   074.071  022          05098        STAX     D                 COPY
   074.072  023          05099        INX      D
   074.073  013          05100        DCX      B
   074.074  170          05101        MOV      A,B
   074.075  261          05102        ORA      C
   074.076  302 065 074  05103        JNZ      LDD2              MORE TO GO
                         05104
                         05105   *    CODE ALL LOADED. RELOCATE IT
                         05106
   074.101  301          05107        POP      B                 (BC) = REL FACTOR
   074.102  005          05108        DCR      B
   074.103  005          05109        DCR      B
   000.000                05110        ERRNZ    DVD.ENT-2000A     ASSUME DRIVER ENTRY = 2000A
   074.104  315 136 074  05111   LDD3 CALL     LDD8
   074.107  136          05112        MOV      E,M
   074.110  315 136 074  05113        CALL     LDD8
   074.113  126          05114        MOV      D,M               (DE) = REL ADDRESS OF WORD TO RELOCATE
   074.114  172          05115        MOV      A,D
   074.115  263          05116        ORA      E
   074.116  312 323 032  05117        JZ       LDD4              ALL DONE
   074.121  353          05118        XCHG                       (HL) = REL ADDRESS OF WORD TO RELOCATE
   074.122  011          05119        DAD      B                 (HL) = ABS ADDRESS OF WORD TO RELOCATE
   074.123  176          05120        MOV      A,M
   074.124  201          05121        ADD      C
   074.125  167          05122        MOV      M,A
   074.126  043          05123        INX      H
   074.127  176          05124        MOV      A,M
   074.130  210          05125        ADC      B
   074.131  167          05126        MOV      M,A
   074.132  353          05127        XCHG                       RESTORE (HL)
   074.133  303 104 074  05128        JMP      LDD3
                         05129
                         05130   *    SETUP ENTRY ADDRESSES IN TABLES
                         05131
   032.323               05132   LDD4 EQU      32323A            USE WHATS IN ROM
   032.361               05133   PCHL EQU      32361A            USE PCHL IN ROM


                         05135   **   LDD8 - READ A BYTE FROM THE FILE.                    /80.06.gc/
                         05136   *
                         05137   *    This code has been modified to read device drivers off of
                         05138   *    system volumes with many differant SPG factors.  Unfortunately,
                         05139   *    if a new volume is mounted, the system volume stuff will not
                         05140   *    be updated.
                         05141   *
                         05142   *    If the sector pointer were not on an even boundary, or
                         05143   *    the cluster factor were not a multiple of 2, this code
```

```
                     05144   *    would not work well.  Therefore, it is up to INIT to
                     05145   *    enforce even clusters.
                     05146   *
                     05147   *
                     05148   *    ENTRY    (HL) = SECSCR POINTER OF CURRENT BYTE
                     05149   *
                     05150   *             LDD8A    = address of system device GRT              /80.06.gc/
                     05151   *             LDD8B    = Sectors/Group on system volume    /80.06.gc/
                     05152   *             LDD8C    = current sector number                     /80.06.gc/
                     05153   *             LDD9A and LDD9B as required by LDD9          /80.06.GC/
                     05154   *
                     05155   *    EXIT     (HL) = ADDRESS OF NEXT BYTE
                     05156   *
                     05157   *    USES     A,F,H,L
                     05158   *
                     05159
074.136  054         05160   LDD8 INR      L                POINT TO NEXT BYTE
074.137  300         05161        RNZ                      GOT IT
                     05162
074.140  044         05163        INR      H               MAYBE IN NEXT GROUP
074.141  345         05164        PUSH     H                                                    /79.11.GC/
074.142  041 126 101 05165        LXI      H,SECSCR+256                                         /79.11.GC/
074.145  174         05166        MOV      A,H                                                  /79.11.GC/
074.146  341         05167        POP      H                                                    /79.11.GC/
                     05168   *    MVI      A,SECSCR+256/256                            /79.11.GC/
074.147  274         05169        CMP      H
074.150  310         05170        RE                        OK, IN SECOND SECTOR NOW
                     05171
                     05172   *    MUST READ ANOTHER
                     05173
074.151  305         05174        PUSH     B
074.152  325         05175        PUSH     D
074.153  021 126 100 05176        LXI      D,SECSCR DE = address
074.156  001 000 002 05177        LXI      B,512            (BC) = COUNT
074.161  052 207 074 05178        LHLD     LDD8C            HL = Sector Number          /80.06.GC/
074.164  325         05179        PUSH     D                SAVE #SECSCR                        /80.06.GC/
074.165  345         05180        PUSH     H                                                    /80.06.GC/
074.166  315 275 031 05181        CALL     S.READ           READ IT                             /80.06.gc/
074.171  341         05182        POP      H                HL = sector number          /80.06.gc/
074.172  315 211 074 05183        CALL     LDD9                                                 /80.06.gc/
074.175  042 207 074 05184        SHLD     LDD8C            Update Sector Number                /80.06.gc/
074.200  341         05185        POP      H                HL = SECSCR                         /80.06.GC/
074.201  321         05186        POP      D                RESTORE (DE) AND (BC)
074.202  301         05187        POP      B
074.203  311         05188        RET
                     05189
074.204  000 000     05190   LDD8A        DW       0               System GRT address          /80.06.gc/
074.206  000         05191   LDD8B        DB       0               System Sectors/Group                /80.06.gc/
074.207  000 000     05192   LDD8C        DW       0               Current Sector Number               /80.06.gc/
```

```
                        05194   **   LDD9                                                          /80.06.GC/
                        05195   *
                        05196   *    LDD9 advances the current driver sector pointer
                        05197   *    to the next multiple of 2 sectors.  Not this routine
                        05198   *    will not work if the cluster is odd, or the *set*
                        05199   *    code is not a multiple of 512.
                        05200   *
                        05201   *    ENTRY:   HL       = Current Sector Number
                        05202   *             LDD8A, LDD8B, and LDD8C as required by LDD8
                        05203   *             LDD9A    = 512 bytes reads per group
                        05204   *             LDD9B    = read index current group
                        05205   *
                        05206   *    EXIT:    HL       = Next Sector Number
                        05207   *
                        05208   *    USES:    PSW,HL
                        05209   *
   002.000              05210   .    SET      DVD.ENT/512*512
   000.000              05211        ERRNZ    DVD.ENT-.          Must be a multiple of 512
                        05212
   074.211  072 261 074 05213   LDD9 LDA      LDD9B
   074.214  075         05214        DCR      A                  Count these sectors
   074.215  062 261 074 05215        STA      LDD9B
   074.220  312 226 074 05216        JZ       LDD10              At the end of this group
                        05217
                        05218   *    More available sectors in this group
                        05219
   074.223  043         05220        INX      H
   074.224  043         05221        INX      H
   074.225  311         05222        RET
                        05223
                        05224   *    Need to find the next Group
                        05225
   074.226  072 260 074 05226   LDD10          LDA      LDD9A
   074.231  062 261 074 05227        STA      LDD9B              Reset Counter
   074.234  072 206 074 05228        LDA      LDD8B              A  = SPG
   074.237  315 016 075 05229        CALL     STB                HL = Block Number
   074.242  072 205 074 05230        LDA      LDD8A+1
   074.245  147         05231        MOV      H,A                HL = GRT address
   074.246  156         05232        MOV      L,M                L  = Next Block Number
   074.247  046 000     05233        MVI      H,0
   074.251  072 206 074 05234        LDA      LDD8B              A  = SPG
   074.254  315 217 073 05235        CALL     BTS                HL = sector Number
   074.257  311         05236        RET
                        05237
   074.260  000         05238   LDD9A          DB       0                 Number of 512 byte reads per group
   074.261  000         05239   LDD9B          DB       0                 Current read index



                        05241   **   OTI      - OVERLAY TABLE INDEX
                        05242   *
                        05243   *    OTI COMPUTES THE OVERLAY TABLE INDEX ADDRESS BASED ON THE OVERLAY
                        05244   *    INDEX, (AS DEFINED IN LOADO,) AND THE OFFSET INTO THE TABLE ENTRY.
                        05245   *
                        05246   *    USE:     CALL     OTI
                        05247   *             DW       offset
```

```
                     05248   *
                     05249   *
                     05250   *    ENTRY:    (A)       = OVERLAY INDEX
                     05251   *
                     05252   *    EXIT:     (HL)      = ADDRESS OF THE SPECIFIED TABLE ENTRY
                     05253   *
                     05254   *    USES:     (PSW),(HL)
                     05255   *
                     05256
  074.262  207       05257   OTI  ADD       A                 (A) = 2*(A)
  074.263  207       05258        ADD       A                 (A) = 4*(A)
  074.264  207       05259        ADD       A                 (A) = 8*(A)
  000.000            05260        ERRNZ     OVL.ENS-8
  074.265  041 036 076 05261        LXI       H,OVLTAB  TABLE FIRST WORD ADDRESS
  074.270  315 101 030 05262        CALL      $DADA.            (HL) = TABLE ENTRY ADDRESS
  074.273  353         05263        XCHG
  074.274  343         05264        XTHL                        SAVE (DE)
  074.275  325         05265        PUSH      D                 SAVE TABLE ENTRY ADDRESS
  074.276  136         05266        MOV       E,M
  074.277  043         05267        INX       H
  074.300  126         05268        MOV       D,M               (DE) = TABLE ENTRY OFFSET
  074.301  043         05269        INX       H                 (HL) = RETURN ADDRESS
  074.302  343         05270        XTHL                        (HL) = TABLE ENTRY ADDRESS
  074.303  031         05271        DAD       D                 (HL) = TABLE ENTRY OFFSET ADDRESS
  074.304  321         05272        POP       D                 (DE) = RETURN ADDRESS
  074.305  353         05273        XCHG                        (HL) = RETURN ADDRESS
  074.306  343         05274        XTHL                        (HL) = OLD (DE)
  074.307  353         05275        XCHG                        (HL) = TAB. ENTRY OFFSET ADDR, (DE = OLD (DE)
  074.310  311         05276        RET



                     05278   **   RRC - REMOVE REGULAR CHARACTER
                     05279   *
                     05280   *    RRC REMOVES THE LAST CHARACTER IN THE INPUT CIRCULAR BUFFER,
                     05281   *    IF IT IS NOT A NEW-LINE CHARACTER (00).
                     05282   *
                     05283   *    ENTRY     NONE
                     05284   *    EXIT      'Z' SET IF NO CHARACTERS, OR LAST ONE IS '00
                     05285   *              'Z' CLEAR IF GOT CHARACTER
                     05286   *               (A) = CHARACTER
                     05287   *    USES      A,F,H,L
                     05288
                     05289
  074.311  052 306 077 05290   RRC  LHLD      SCIIN
  074.314  072 310 077 05291        LDA       SCIOUT
  074.317  275         05292        CMP       L
  074.320  310         05293        RE                          NONE
  074.321  072 312 077 05294        LDA       SCIFWA
  074.324  275         05295        CMP       L
  074.325  302 333 074 05296        JNE       RRC1              NOT AT BEGINNING
  074.330  052 314 077 05297        LHLD      SCILWA
  074.333  053         05298   RRC1 DCX       H                 DECREMENT POINTER
  074.334  176         05299        MOV       A,M               (A) = VALUE
  074.335  376 012     05300        CPI       NL
  074.337  310         05301        RE                          IS END OF LINE
```

```
   074.340  376 004     05302        CPI      CTLD
   074.342  310         05303        RE                        IS END OF FILE
   074.343  042 306 077 05304        SHLD     SCIIN            UPDATE POINTER
   074.346  311         05305        RET



                        05307   **   SCI - STORE CHANNEL INFORMATION.
                        05308   *
                        05309   *    SCI SAVES THE ACTIVE CHANNEL INFORMATION BACK
                        05310   *    INTO THE CHANNEL BLOCK.
                        05311   *
                        05312   *    ENTRY    NONE
                        05313   *    EXIT     NONE
                        05314   *    USES     NONE
                        05315
                        05316
   074.347  315 054 031 05317   SCI  CALL     $SAVALL
   074.352  052 116 041 05318        LHLD     AIO.CHA
   000.000              05319        ERRNZ    IOC.FLG-IOC.DDA-2
   074.355  043         05320        INX      H
   074.356  043         05321        INX      H                 (HL) = IOB.FLG ADDRESS
   074.357  001 010 000 05322        LXI      B,IOC.DRL(BC) = LEN
   074.362  021 043 041 05323        LXI      D,AIO.FLG
   074.365  315 252 030 05324        CALL     $MOVE             MOVE DATA
   074.370  303 047 031 05325        JMP      $RSTALL           RESTORE ALL REGS



                        05327   **   SDD - STAND-IN DEVICE DRIVER.
                        05328   *
                        05329   *    SDD IS SETUP AS THE DEVICE DRIVER ADDRESS FOR DRIVERS WHICH
                        05330   *    ARE NOT IN MEMROY. IF THE REQUEST IS AN OPEN, POSTPONE IT
                        05331   *    UNTIL 'LDD' LOADS THE OVERLAY. OTHERWISE, IS A FATAL
                        05332   *    SYSTEM ERROR.
                        05333   *
                        05334   *    ENTRY    (A) = CODE
                        05335   *    EXIT     NONE
                        05336   *    USES     A,F
                        05337
                        05338
   074.373  376 011     05339   SDD  CPI      DC.LOD            check for load                    /80.04.gc/
   074.375  312 012 075 05340        JZ       SDD1                                                /80.04.gc/
                        05341
   075.000  376 003     05342        CPI      DC.OPR
   075.002  334 115 066 05343        CC       FATSERR
   075.005  376 006     05344        CPI      DC.OPU+1
   075.007  324 115 066 05345        CNC      FATSERR
                        05346
   075.012  062 370 040 05347   SDD1 STA      S.DDOPC           SET CODE                 /80.04.gc/
   075.015  311         05348        RET
```

```
                     05350   **   STB      - Sector to Block                          /80.06.gc/
                     05351   *
                     05352   *    STB converts a sector number to the corresponding
                     05353   *    block number
                     05354   *
                     05355   *    ENTRY:   A        = Sectors Per Group
                     05356   *             HL       = Sector Number
                     05357   *
                     05358   *    EXIT:    HL       = Block Number
                     05359   *
                     05360   *    USES:    HL
                     05361   *
                     05362
  075.016  365       05363   STB  PUSH     PSW
  075.017  305       05364        PUSH     B
  075.020  325       05365        PUSH     D
  075.021  104       05366        MOV      B,H
  075.022  115       05367        MOV      C,L              BC = sector number
  075.023  026 000   05368        MVI      D,0
  075.025  137       05369        MOV      E,A              DE = sectors/group
  075.026  315 106 030 05370      CALL     $DU66            HL = BC / DE
  075.031  321       05371        POP      D
  075.032  301       05372        POP      B
  075.033  361       05373        POP      PSW
  075.034  311       05374        RET
```

```
   075.035              05377        XTEXT    TBRA
                        05378X



                        05380X  **   $TBRA - BRANCH RELATIVE THOUGH TABLE.
                        05381X  *
                        05382X  *    $TBRA USES THE SUPPLIED INDEX TO SELECT A BYTE FROM THE
                        05383X  *    JUMP TABLE. THE CONTENTS OF THIS BYTE ARE ADDED TO THE
                        05384X  *    ADDRESS OF THE BYTE, YEILDING THE PROCESSOR ADDRESS.
                        05385X  *
                        05386X  *    CALL     $TBRA
                        05387X  *    DB       LAB1-*            INDEX = 0 FOR LAB1
                        05388X  *    DB       LAB2-*            INDEX = 1 FOR LAB2
                        05389X  *    DB       LABN-*            INDEX = N-1 FOR LABN
                        05390X  *
                        05391X  *    ENTRY    (A) = INDEX
                        05392X  *             (RET) = TABLE FWA
                        05393X  *    EXIT     TO COMPUTED ADDRESS
                        05394X  *    USES     F,H,L
                        05395X
                        05396X
   031.076              05397X  $TBRA        EQU      31076A            IN H17 ROM
   075.035              05398        XTEXT    GUP



                        05400X  **   GUP      - GET UNIT POINTER                          /80.04.GC/
                        05401X  *
                        05402X  *    GET THE UNIT SPECIFIC DATA POINTER FOR THE SPECIFIED UNIT
                        05403X  *
                        05404X  *
                        05405X  *    ENTRY:   A        = UNIT NUMBER
                        05406X  *             HL       = ADDRESS OF UNIT TABLE
                        05407X  *
                        05408X  *    EXIT:    HL       = ADDRESS OF TABLE ENTRY FOR SPECIFIED UNIT
                        05409X  *
                        05410X  *    USES:    PSW,HL
                        05411X  *
                        05412X
   075.035  365         05413X  GUP  PUSH     PSW
   075.036  315 211 030 05414X       CALL     $HLIHL            HL = POINTER TO UNIT TABLE
   075.041  361         05415X       POP      PSW               SAVE A
                        05416X
   075.042  346 007     05417X       ANI      007A              Map out any extra bits
   075.044  007         05418X       RLC
   075.045  007         05419X       RLC
   075.046  007         05420X       RLC
   000.000              05421X       ERRNZ    UNT.SIZ-8
                        05422X
   075.047  315 101 030 05423X       CALL     $DADA.            HL - HL + A
   075.052  311         05424X       RET
   075.053              05425        XTEXT    HLCPDE
                        05426X
                        05427X  **   HLCPDE   - (HL) COMPARED TO (DE)
```

```
                    05428X  *
                    05429X  *    THIS ROUTINE IS DOUBLE WORD COMPARE OF REGISTER PAIRS (DE) AND (HL).
                    05430X  *
                    05431X  *    ENTRY:    (HL)&(DE) SET UP
                    05432X  *
                    05433X  *    EXIT:     (PSW)     =
                    05434X  *                         'Z' SET   IF (HL) =  (DE)
                    05435X  *                         'C' SET   IF (HL) <  (DE)
                    05436X  *                         'C' CLEAR IF (HL) >= (DE)
                    05437X  *
                    05438X  *
                    05439X  *    USES:     (PSW)
                    05440X  *
                    05441X
  075.053  174      05442X  HLCPDE        MOV       A,H
  075.054  272      05443X        CMP       D         'C' SET => (A) < (D)
  075.055  300      05444X        RNZ
  075.056  175      05445X        MOV       A,L
  075.057  273      05446X        CMP       E         'C' SET => (L) < (E)
  075.060  311      05447X        RET
  075.061           05448         XTEXT     HLIHL




                    05450X  **   $HLIHL - LOAD HL INDIRECT THROUGH HL.
                    05451X  *
                    05452X  *    (HL) = ((HL))
                    05453X  *
                    05454X  *    ENTRY     NONE
                    05455X  *    EXIT      NONE
                    05456X  *    USES      A,H,L
                    05457X
  030.211           05458X  $HLIHL        EQU       30211A            IN H17 ROM
  075.061           05459         XTEXT     ILDEHL




                    05461X  **   ILDELH   -  INDEXED LOAD OF DE FROM HL
                    05462X  *
                    05463X  *    'DE' GET THE FULL WORD VALUE POINTED TO BY 'HL', AND 'HL' IS
                    05464X  *    INCREMENTED BY TWO.
                    05465X  *
                    05466X  *    ENTRY:    HL        = ADDRESS OF FULL WORD VALUE
                    05467X  *
                    05468X  *    EXIT:     DE        = (HL)
                    05469X  *              HL        = HL + 2
                    05470X  *
                    05471X  *    USES:     DE
                    05472X  *
                    05473X
  075.061  136      05474X  ILDEHL        MOV       E,M
  075.062  043      05475X        INX       H
  075.063  126      05476X        MOV       D,M
  075.064  043      05477X        INX       H
  075.065  311      05478X        RET
```

```
   075.066              05479        XTEXT    INDL



                        05481X  **   $INDL - INDEXED LOAD.
                        05482X  *
                        05483X  *    $INDL LOADS DE WITH THE TWO BYTES AT (HL)+DISPLACEMENT
                        05484X  *
                        05485X  *    THIS ACTS AS AN INDEXED FULL WORD LOAD.
                        05486X  *
                        05487X  *    (DE) = ( (HL) + DISPLACEMENT )
                        05488X  *
                        05489X  *    ENTRY    ((RET)) = DISPLACEMENT (FULL WORD)
                        05490X  *             (HL) = TABLE ADDRESS
                        05491X  *    EXIT     TO (RET+2)
                        05492X  *    USES     A,F,D,E
                        05493X
                        05494X
   030.234              05495X  $INDL         EQU      30234A            IN H17 ROM
   075.066              05496        XTEXT    INDXX



                        05498X  **      $INDLB  -  INDEXED LOAD BYTE
                        05499X  *
                        05500X  *       BYTE INDEXED LOAD PRIMITIVE
                        05501X  *
                        05502X  *       ENTRY:  HL      = BASE ADDRESS
                        05503X  *               (RET)   = FULL WORD RELOCATION
                        05504X  *
                        05505X  *       EXIT:   A       = ( HL + (RET) )
                        05506X  *
                        05507X  *       USES:   A
                        05508X  *
                        05509X
   075.066  353         05510X  $INDLB  XCHG                     DE = BASE
   075.067  343         05511X          XTHL                     SAVE .DE.
   075.070  325         05512X          PUSH    D                SAVE BASE
   075.071  305         05513X          PUSH    B                SAVE .BC.
                        05514X
   075.072  116         05515X          MOV     C,M
   075.073  043         05516X          INX     H
   075.074  106         05517X          MOV     B,M              BC = OFFSET
   075.075  043         05518X          INX     H                HL =  .RET.
                        05519X
   075.076  353         05520X          XCHG                     HL = BASE
   075.077  011         05521X          DAD     B                HL = BASE + OFFSET
   075.100  176         05522X          MOV     A,M              A  = ( BASE + OFFSET )
   075.101  353         05523X          XCHG                     HL =  .RET.
                        05524X
   075.102  301         05525X          POP     B                RESTORE  .BC.
   075.103  321         05526X          POP     D                RESTORE  BASE
   075.104  343         05527X          XTHL                     HL = .DE. ; (SP) = .RET.
   075.105  353         05528X          XCHG                     DE = .DE. ; HL = BASE
   075.106  311         05529X          RET
```

```
                     05531X  **      $INDS   -  INDEXED STORE
                     05532X  *
                     05533X  *       INDEXED STORE PRIMITIVE.
                     05534X  *
                     05535X  *       ENTRY:  HL = BASE ADDRESS
                     05536X  *               DE = VALUE TO STORE
                     05537X  *
                     05538X  *       EXIT:   ( HL + (RET) ) = DE
                     05539X  *
                     05540X  *       USES:   NONE
                     05541X  *
                     05542X
  075.107  315 171 075 05543X  $INDS   CALL    XCHGBC
  075.112  343         05544X          XTHL                    SAVE  .BC.
  075.113  325         05545X          PUSH    D
  075.114  315 061 075 05546X          CALL    ILDEHL          DE = OFFSET
  075.117  315 171 075 05547X          CALL    XCHGBC          BC = .RET.
  075.122  353         05548X          XCHG                    DE = BASE ; HL = OFFSET
  075.123  031         05549X          DAD     D               HL = BASE + OFFSET
  075.124  353         05550X          XCHG
  075.125  343         05551X          XTHL                    SAVE BASE
  075.126  353         05552X          XCHG                    DE = VALUE
  075.127  315 164 075 05553X          CALL    ISDEHL
  075.132  341         05554X          POP     H               H = BASE
  075.133  315 171 075 05555X          CALL    XCHGBC
  075.136  343         05556X          XTHL                    RESTORE  .BC.
  075.137  315 171 075 05557X          CALL    XCHGBC
  075.142  311         05558X          RET



                     05560X  **      $INDSB  -  INDEXED BYTE STORE
                     05561X  *
                     05562X  *       INDEXED BYTE STORE.
                     05563X  *
                     05564X  *       ENTRY:  A       = VALUE TO STORE
                     05565X  *               HL      = BASE ADDRESS
                     05566X  *               (RET)   = OFFSET
                     05567X  *
                     05568X  *       EXIT:   NONE
                     05569X  *
                     05570X  *       USES:   PSW
                     05571X  *
                     05572X
  075.143  353         05573X  $INDSB  XCHG                    DE = BASE
  075.144  343         05574X          XTHL                    SAVE  .DE.
  075.145  325         05575X          PUSH    D               SAVE  BASE
  075.146  305         05576X          PUSH    B               SAVE  .BC.
                       05577X
  075.147  116         05578X          MOV     C,M
  075.150  043         05579X          INX     H
  075.151  106         05580X          MOV     B,M             BC = OFFSET
  075.152  043         05581X          INX     H               HL =  .RET.
                       05582X
  075.153  353         05583X          XCHG                    HL = BASE
  075.154  011         05584X          DAD     B               HL = BASE + OFFSET
```

```
   075.155  167         05585X          MOV     M,A             ( BASE + OFFSET ) = A
   075.156  353         05586X          XCHG
                        05587X
   075.157  301         05588X          POP     B               RESTORE  .BC.
   075.160  321         05589X          POP     D               RESTORE  BASE
   075.161  343         05590X          XTHL                    HL = .DE. ; (SP) = .RET.
   075.162  353         05591X          XCHG                    DE = .DE. ; HL = BASE
   075.163  311         05592X          RET
   075.164              05593       XTEXT    ISDEHL



                        05595X  **      ISDEHL  -  INDEXED STORE OF DE AT HL
                        05596X  *
                        05597X  *       STORE 'DE' AT THE ADDRESS POINTED TO BY 'HL', AND INCREMENT 'HL'
                        05598X  *       BY 2.
                        05599X  *
                        05600X  *       ENTRY:  DE      = VALUE
                        05601X  *               HL      = ADDRESS OF VALUE
                        05602X  *
                        05603X  *       EXIT:   (HL)    = DE
                        05604X  *               HL      = HL + 2
                        05605X  *
                        05606X  *       USES:   HL
                        05607X  *
                        05608X
   075.164  163         05609X  ISDEHL  MOV     M,E
   075.165  043         05610X          INX     H
   075.166  162         05611X          MOV     M,D
   075.167  043         05612X          INX     H
   075.170  311         05613X          RET
   075.171              05614       XTEXT    MOVE
                        05615X



                        05617X  **   $MOVE - MOVE DATA
                        05618X  *
                        05619X  *    $MOVE MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
                        05620X  *    IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
                        05621X  *    FIRST TO LAST.
                        05622X  *
                        05623X  *    IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
                        05624X  *    LAST TO FIRST.
                        05625X  *
                        05626X  *    THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
                        05627X  *
                        05628X  *    ENTRY    (BC) = COUNT
                        05629X  *             (DE) = FROM
                        05630X  *             (HL) = TO
                        05631X  *    EXIT     MOVED
                        05632X  *             (DE) = ADDRESS OF NEXT FROM BYTE
                        05633X  *             (HL) = ADDRESS OF NEXT *TO* BYTE
                        05634X  *             'C' CLEAR
                        05635X  *    USES     ALL
```

```
                    05636X
                    05637X
   030.252          05638X  $MOVE         EQU      30252A            IN H17 ROM
   075.171          05639         XTEXT    DADA2



                    05641X  **   $DADA. - ADD (0,A) TO (H,L)
                    05642X  *
                    05643X  *    ENTRY    NONE
                    05644X  *    EXIT     (HL) = (HL) + (0A)
                    05645X  *    USES     A,F,H,L
                    05646X
                    05647X
   030.101          05648X  $DADA.        EQU      30101A            IN H17 ROM
   075.171          05649         XTEXT    SAVALL
                    05650X



                    05652X  **   $RSTALL - RESTORE ALL REGISTERS.
                    05653X  *
                    05654X  *    $RSTALL RESTORES ALL THE REGISTERS OFF THE STACK, AND
                    05655X  *    RETURNS TO THE PREVIOUS CALLER.
                    05656X  *
                    05657X  *    ENTRY    (SP) = PSW
                    05658X  *             (SP+2) = BC
                    05659X  *             (SP+4) = DE
                    05660X  *             (SP+6) = HL
                    05661X  *             (SP+8) = RET
                    05662X  *    EXIT     TO *RET*, REGISTERS RESTORED
                    05663X  *    USES     ALL
                    05664X
                    05665X
   031.047          05666X  $RSTALL       EQU      31047A            IN H17 ROM



                    05668X  **   $SAVALL - SAVE ALL REGISTERS ON STACK.
                    05669X  *
                    05670X  *    $SAVALL SAVES ALL THE REGISTERS ON THE STACK.
                    05671X  *
                    05672X  *    ENTRY    NONE
                    05673X  *    EXIT     (SP) = PSW
                    05674X  *             (SP+2) = BC
                    05675X  *             (SP+4) = DE
                    05676X  *             (SP+6) = HL
                    05677X  *    USES     H,L
                    05678X
                    05679X
   031.054          05680X  $SAVALL       EQU      31054A            IN H17 ROM
   075.171          05681         XTEXT    COMP
```

```
                     05683X  **   $COMP - COMPARE TWO CHARACTER STRINGS.
                     05684X  *
                     05685X  *    $COMP COMPARES TWO BYTE STRINGS.
                     05686X  *
                     05687X  *    ENTRY     (C) = COMPARE COUNT
                     05688X  *              (DE) = FWA OF STRING #1
                     05689X  *              (HL) = FWA OF STRING #2
                     05690X  *    EXIT      'Z' CLEAR, IS MIS-MATCH
                     05691X  *               (C) = LENGTH REMAINING
                     05692X  *               (DE) = ADDRESS OF MISMATCH IN STRING#1
                     05693X  *               (HL) = ADDRESS OF MISMATCH IN STRING #2
                     05694X  *              'C' SET, HAVE MATCH
                     05695X  *               (C) = 0
                     05696X  *               (DE) = (DE) + (0C)
                     05697X  *               (HL) = (HL) = (0C)
                     05698X  *    USES      A,F,C,D,E,H,L
                     05699X
                     05700X
  030.060            05701X  $COMP          EQU      30060A            IN H17 ROM
  075.171            05702          XTEXT    XCHGBC



                     05704X  **   XCHGBC    -  XCHG BC
                     05705X  *
                     05706X  *    EXCHANGE THE 'BC' REGISTER PAIR WITH THE 'HL' REGISTER PAIR.
                     05707X  *
                     05708X  *    ENTRY:    BC       = ORIGINAL BC
                     05709X  *              HL       = ORIGINAL HL
                     05710X  *
                     05711X  *    EXIT:     BC       = ORIGINAL HL
                     05712X  *              HL       = ORIGINAL BC
                     05713X  *
                     05714X  *    USES:     BC,HL
                     05715X  *
                     05716X
  075.171  365       05717X  XCHGBC         PUSH     PSW
  075.172  170       05718X          MOV      A,B
  075.173  104       05719X          MOV      B,H
  075.174  147       05720X          MOV      H,A
  075.175  171       05721X          MOV      A,C
  075.176  115       05722X          MOV      C,L
  075.177  157       05723X          MOV      L,A
  075.200  361       05724X          POP      PSW
  075.201  311       05725X          RET
  075.202            05726          XTEXT    ZERO



                     05728X  **   $ZERO - ZERO MEMORY
                     05729X  *
                     05730X  *    $ZERO ZEROS A BLOCK OF MEMORY.
                     05731X  *
                     05732X  *    ENTRY     (HL) = ADDRESS
                     05733X  *               (B) = COUNT
```

```
                    05734X  *    EXIT     (A) = 0
                    05735X  *    USES     A,B,F,H,L
                    05736X
   031.212          05737X  $ZERO         EQU      031212A           ; IN H17 ROM
   075.202          05738        XTEXT    WER



                    05740X  **   $WER - WRITE ENABLE RAM.
                    05741X  *
                    05742X  *    $WER IS CALLED TO ENABLE WRITTING TO THE H17 CONTROLLER'S
                    05743X  *    RAM AREA.
                    05744X  *
                    05745X  *    ENTRY    NONE
                    05746X  *    EXIT     NONE
                    05747X  *    USES     NONE
                    05748X
                    05749X
   031.241          05750X  $WER EQU      31241A            IN H17 ROM



                    05752X  **   $WDR - WRITE DISABLE RAM.
                    05753X  *
                    05754X  *    $WDR IS CALLED TO DISABLE WRITTING TO THE H17 CONTROLLER'S
                    05755X  *    RAM AREA.
                    05756X  *
                    05757X  *    ENTRY    NONE
                    05758X  *    EXIT     NONE
                    05759X  *    USES     NONE
                    05760X
   031.222          05761X  $WDR EQU      31222A            IN H17 ROM
   075.202          05762        XTEXT    CHL
                    05763X



                    05765X  **   $CHL - COMPLEMENT (HL).
                    05766X  *
                    05767X  *    (HL) = -(HL)              TWO'S COMPLEMENT
                    05768X  *
                    05769X  *    ENTRY    NONE
                    05770X  *    EXIT     NONE
                    05771X  *    USES     A,F,H,L
                    05772X
                    05773X
   030.224          05774X  $CHL EQU      30224A            IN H17 ROM
```

```
                     05777   **   THE FOLLOWING ROUTINES ARE REPLACEMENTS FOR THE H17 ROM CODE.



                     05779   **   ISY      - Internal SY Device Driver
                     05780   *
                     05781   *    ISY maps the logical system device units into the real
                     05782   *    physical units known by the driver.  This is done so
                     05783   *    that any device may be booted.
                     05784   *
                     05785   *    ENTRY:   NONE
                     05786   *
                     05787   *    EXIT:    NONE
                     05788   *
                     05789   *    USES:    NONE
                     05790   *
                     05791
075.202  345        05792   ISY  PUSH     H
075.203  041 061 041 05793        LXI      H,AIO.UNI
075.206  146        05794        MOV      H,M
075.207  343        05795        XTHL                        Save current device specification
                     05796
075.210  365        05797        PUSH     PSW
075.211  325        05798        PUSH     D
075.212  345        05799        PUSH     H
                     05800
075.213  052 377 073 05801        LHLD     GSPA
075.216  315 066 075 05802        CALL     $INDLB
075.221  010 000     05803        DW       DEV.MNU
075.223  127        05804        MOV      D,A                D = max num of units
                     05805
075.224  072 061 041 05806        LDA      AIO.UNI
075.227  052 321 077 05807        LHLD     SUNIT
075.232  205        05808        ADD      L                  A = AIO.UNI+SUNIT
                     05809
075.233  222        05810        SUB      D
075.234  322 240 075 05811        JNC      ISY1
075.237  202        05812        ADD      D                  A = A mod D
075.240             05813   ISY1 EQU      *
                     05814
075.240  062 061 041 05815        STA      AIO.UNI
                     05816
075.243  341        05817        POP      H
075.244  321        05818        POP      D
075.245  361        05819        POP      PSW
                     05820
075.246  345        05821        PUSH     H
075.247  041 261 075 05822        LXI      H,ISY2
075.252  343        05823        XTHL                        Set up RETurn address
075.253  345        05824        PUSH     H
075.254  052 322 077 05825        LHLD     MSYDD
075.257  343        05826        XTHL                        Set up Driver address on Stack
075.260  311        05827        RET                         Call Driver
                     05828
075.261  343        05829   ISY2 XTHL
075.262  365        05830        PUSH     PSW
```

```
   075.263  174         05831        MOV      A,H
   075.264  062 061 041 05832        STA      AIO.UNI           Replace the original value
   075.267  361         05833        POP      PSW
   075.270  341         05834        POP      H                 Restore exit HL
   075.271  311         05835        RET
```

```
                     05838   ***  TTDVD - RESIDENT TT DEVICE DRIVER.*
                     05839
                     05840
   075.272           05841   TTDVD         EQU      *
   075.272  315 076 031 05842         CALL     $TBRA
                     05843
   075.275  041      05844   TTDVD0        DB       TTREAD-* READ
   075.276  114      05845         DB       TTWRITE-*WRITE
   075.277  011      05846         DB       TTABT-*           READR
   075.300  014      05847         DB       TTOPE-*           OPENR
   075.301  013      05848         DB       TTOPE-*           OPENW
   075.302  006      05849         DB       TTABT-*           OPENU
   075.303  026      05850         DB       TTNOP-*           CLOSE
   075.304  025      05851         DB       TTNOP-*           ABORT
   075.305  003      05852         DB       TTABT-*           MOUNT
   075.306  023      05853         DB       TTNOP-*           LOAD                                  /80.04.GC/
   075.307  022      05854         DB       TTNOP-*           Ready                                 /80.06.gc/
   000.000           05855         ERRNZ    *-TTDVD0-DC.MAX   Handle ALL requests                 /80.06.gc/
                     05856
   075.310  076 027  05857   TTABT         MVI      A,EC.DDA DEVICE DRIVER ABORT
   075.312  067      05858         STC
   075.313  311      05859         RET
                     05860
   075.314  072 332 040 05861   TTOPE         LDA      S.CONFL
   075.317  346 376  05862         ANI      377Q-CO.FLG               CLEAR CTL-O
   075.321  062 332 040 05863         STA      S.CONFL
                     05864   *     LDA      S.CSLMD                                               /79.02.04.GC/
                     05865   *     ANI      CSL.ECH           PRESERVE ECHO BIT          /79.02.04.GC/
                     05866   *     ORI      CSL.WRP           SET WRAP MODE                       /79.02.04.GC/
                     05867   *     MVI      A,CSL.WRP                                    /79.02.GC/
                     05868   *     STA      S.CSLMD           SET WRAP MODE                       /79.04.GC/
   075.324  257      05869         XRA      A
   075.325  062 011 076 05870         STA      EOFFLG            CLEAR EOF ON INPUT FLAG
   075.330  311      05871         RET
                     05872
   075.331  247      05873   TTNOP         ANA      A
   075.332  311      05874         RET                        DO NOTHING



                     05876   **   TTREAD - READ
                     05877   *
                     05878
   075.333  022      05879   TTR2 STAX     D                 STORE CHAR
   075.334  023      05880         INX      D
   075.335  013      05881         DCX      B
                     05882
   075.336           05883   TTREAD        EQU      *
   075.336  072 011 076 05884         LDA      EOFFLG
   075.341  037      05885         RAR
   075.342  330      05886         RC                         IS EOF
                     05887
   075.343  170      05888         MOV      A,B
   075.344  261      05889         ORA      C
   075.345  310      05890         RZ                         ALL DONE
                     05891
                     05892   *     TAKE  A CHAR
```

```
                        05893
   075.346  072 334 040 05894   TTR1 LDA      S.CAADR+1
   075.351  247         05895        ANA      A
   075.352  302 367 075 05896        JNZ      TTREOF
   075.355  377 001     05897        SCALL    .SCIN                                          /80.06.GC/
   075.357  332 346 075 05898        JC       TTR1              WAIT TILL GOTIT
   075.362  376 004     05899        CPI      CTLD                                           /80.06.GC/
   075.364  302 333 075 05900        JNE      TTR2              NOT CTL-D
                        05901
                        05902   *    HAVE EOF CHARACTER. FILL THIS SECTOR WITH 0'S
                        05903
   075.367  076 003     05904   TTREOF       MVI      A,EC.EOF*2+1
   075.371  062 011 076 05905        STA      EOFFLG            FLAG EOF
   075.374  257         05906   TTR4 XRA      A
   075.375  022         05907        STAX     D                 STORE 0
   075.376  023         05908        INX      D
   075.377  013         05909        DCX      B
   076.000  171         05910        MOV      A,C
   076.001  261         05911        ORA      C
   076.002  302 374 075 05912        JNZ      TTR4
   076.005  076 001     05913        MVI      A,EC.EOF
   076.007  067         05914        STC
   076.010  311         05915        RET
                        05916
                        05917
   076.011  000         05918   EOFFLG       DB       0                EOF FLAG



   076.012  072 334 040 05920   TTWRITE LDA   S.CAADR+1
   076.015  247         05921        ANA      A
   076.016  300         05922        RNZ                        ALL DONE
   076.017  170         05923        MOV      A,B
   076.020  261         05924        ORA      C
   076.021  310         05925        RZ                         ALL DONE
   076.022  032         05926        LDAX     D
   076.023  247         05927        ANA      A
   076.024  312 031 076 05928        JZ       TTW2              NULL CHARACTER
   076.027  377 002     05929        SCALL    .SCOUT                                         /80.06.GC/
   076.031  023         05930   TTW2 INX      D
   076.032  013         05931        DCX      B
   076.033  303 012 076 05932        JMP      TTWRITE
```

```
                     05935   **   RELOCATABLE RAM CELLS.
                     05936   *
                     05937   *    THESE CELLS RESIDE AT THE TOP OF THE MONITOR.
                     05938



                     05940   **   TABLE OF OVERLAY DATA
                     05941   *
                     05942   *    THIS TABLE IS GENERATED AT BOOT-UP TIME
                     05943   *
                     05944
                     05945
   000.002           05946   OVLCNT        EQU      2
                     05947
   076.036           05948   OVLTAB        EQU      *
                     05949
   076.036           05950        DS       OVL.ENS           OVERLAY *HDOSOVL .SYS*
                     05951
   076.046           05952        DS       OVL.ENS                   OVERLAY *HDOSOVL2.SYS*
                     05953
   000.002           05954   OVLMAX        EQU      *-OVLTAB/OVL.ENS
                     05955
   000.000           05956        ERRMI    OVLMAX-OVLCNT



                     05958   **   DEVICE LIST
                     05959
   000.007           05960   DEVCNT        EQU      7                 INITIALLY 7 DEVICES
                     05961
   076.056           05962   DEVLST        DS       0                 DEVICE TABLE




   076.056           05964   TTDEV         DS       0                 TT DEVICE TABLE ENTRY
   000.000           05965        ERRNZ    *-TTDEV-DEV.NAM
   076.056  124 124  05966        DB       'TT'              DEVICE NAME
   000.000           05967        ERRNZ    *-TTDEV-DEV.RES
   076.060  003      05968        DB       DR.IM+DR.PR       PERMENANTLY RESIDENT
   000.000           05969        ERRNZ    *-TTDEV-DEV.JMP
   076.061  303      05970        DB       303Q              JUMP OPCODE
   000.000           05971        ERRNZ    *-TTDEV-DEV.DDA
   076.062  272 075  05972        DW       TTDVD             DRIVER ADDRESS
   000.000           05973        ERRNZ    *-TTDEV-DEV.FLG
   076.064  006      05974        DB       DT.CR+DT.CW
   000.000           05975        ERRNZ    *-TTDEV-DEV.MUM
   076.065  001      05976        DB       1                 MOUNTED MASK
   000.000           05977        ERRNZ    *-TTDEV-DEV.MNU
   076.066  001      05978        DB       1                 MAXIMUM NUMBER OF UNITS
   000.000           05979        ERRNZ    *-TTDEV-DEV.UNT
   076.067  222 076  05980        DW       TTOUNT            UNIT TABLE
   000.000           05981        ERRNZ    *-TTDEV-DEV.DVL
   076.071  000 000  05982        DW       0                 DRIVER LENGTH
   000.000           05983        ERRNZ    *-TTDEV-DEV.DVG
   076.073  000      05984        DB       0                 DRIVER GROUP NUMBER
```

```
   000.000              05985        ERRNZ    *-TTDEV-DEVELEN




   076.074  000         05987        DB       0                NO MORE DEVICES
   076.075              05988        DS       DEVCNT-1*DEVELEN ROOM FOR 5 MORE DEVICES
   076.221  000         05989        DB       0                BYTE USED IF LAST DEVLST ENTRY USED




   076.222              05991   TTOUNT       DS       0
                        05992
   000.000              05993        ERRNZ    UNT.FLG-0
   076.222  006         05994        DB       DT.CR+DT.CW              TT0:
   076.223              05995        DS       UNT.SIZ-1




                        05997   **   INITIAL CHANNEL TABLE.
                        05998
   000.006              05999   CHANCNT EQU   6                6 CHANNELS
                        06000
   076.232              06001   CHANTAB EQU   *
                        06002
                        06003   *    NOTE THAT THE FIRST CHANNEL IS CHANNEL 377Q, WHICH IS THE
                        06004   *    OVERLAY CHANNEL. THE .CLEARA FUNCTION ASSUMES THIS, AS
                        06005   *    DOES FCI.
                        06006
   076.232  304 076     06007        DW       *+IOCELEN LINK - CHANNEL 377
   076.234  000 000 000 06008        DB       0,0,0
   000.000              06009        ERRNZ    IOCCTD-1 USER CHANNEL #0 FOLLOWS
   076.237              06010        DS       IOCELEN-5
   076.304  356 076     06011        DW       *+IOCELEN LINK - CHANNEL 0
   076.306  000 000 000 06012        DB       0,0,0
   076.311              06013        DS       IOCELEN-5
   076.356  030 077     06014        DW       *+IOCELEN LINK - CHANNEL 1
   076.360  000 000 000 06015        DB       0,0,0
   076.363              06016        DS       IOCELEN-5
   077.030  102 077     06017        DW       *+IOCELEN LINK - CHANNEL 2
   077.032  000 000 000 06018        DB       0,0,0
   077.035              06019        DS       IOCELEN-5
   077.102  154 077     06020        DW       *+IOCELEN LINK - CHANNEL 3
   077.104  000 000 000 06021        DB       0,0,0
   077.107              06022        DS       IOCELEN-5
   077.154  226 077     06023        DW       *+IOCELEN LINK - CHANNEL 4
   077.156  000 000 000 06024        DB       0,0,0
   077.161              06025        DS       IOCELEN-5
   077.226  000 000     06026        DW       0                NULL LINK - CHANNEL 5
   077.230  000 000 000 06027        DB       0,0,0
   077.233              06028        DS       IOCELEN-5
                        06029
                        06030   *    OVL LOAD ADDRESS
                        06031
   077.300              06032   HIGHDAT EQU   *
```

```
                     06033
                     06034
                     06035   **   SYSTEM MODE. NON-ZERO WHEN PROCESSING SYSCALL.
                     06036
   000.000           06037        ERRNZ    *-HIGHDAT-M.SYSM
   077.300  000      06038   SYSMODE DB    0
                     06039
   000.000           06040        ERRNZ    *-HIGHDAT-M.SALO
   077.301  000      06041   SALONE       DB       0                STAND ALONE FLAG != 0 => CAN GO STAND ALONE
                     06042
   000.000           06043        ERRNZ    *-HIGHDAT-M.CSLC
   077.302  000      06044   CSLLCNT DB    0                LINES ENTERED IN BUFFER
   000.000           06045        ERRNZ    *-HIGHDAT-M.CPRE
   077.303  000      06046   SCIPRE       DB       0                PREVIOUSLY INPUT CHARACTER
   000.000           06047        ERRNZ    *-HIGHDAT-M.CRUB
   077.304  000      06048   CSLRBF       DB       0                RUBOUT FLAG
                     06049
   000.003           06050   CC.FLG       EQU      00000011B CTL CHARACTERS FLAG
   000.010           06051   CZ.FLG       EQU      00001000B CTL-Z FLAG
   000.000           06052        ERRNZ    *-HIGHDAT-M.CINT
   077.305  000      06053   SCINTFL DB    0                SYSTEM CONSOLE INTERRUPT FLAGS
                     06054
   000.000           06055        ERRNZ    *-HIGHDAT-M.CIN
   077.306  324 077  06056   SCIIN        DW       CSLIBUF          IN POINTER
   000.000           06057        ERRNZ    *-HIGHDAT-M.COUT
   077.310  324 077  06058   SCIOUT       DW       CSLIBUF          OUT POINTER
   000.000           06059        ERRNZ    *-HIGHDAT-M.CFWA
   077.312  324 077  06060   SCIFWA       DW       CSLIBUF
   000.000           06061        ERRNZ    *-HIGHDAT-M.CLWA
   077.314  071 100  06062   SCILWA       DW       CSLIBFE          END POINTER
                     06063
   000.000           06064        ERRNZ    *-HIGHDAT-M.CDLY
   077.316  004      06065   CSLDLY       DB       4                PAD CHARACTER COUNT
   000.000           06066        ERRNZ    *-HIGHDAT-M.CDCA
   077.317  161 071  06067   CSLDCA       DW       SCOUTA           ADDRESS OF DELAY CHARACTER
                     06068
   000.000           06069        ERRNZ    *-HIGHDAT-M.SUNI                          /80.05.gc/
   077.321  000      06070   SUNIT        DB       0                System Unit Number          /80.05.gc/
   000.000           06071        ERRNZ    *-HIGHDAT-M.SYDD                          /2.0a/
   077.322  000 000  06072   MSYDD        DW       0                System Device Driver                        /2.0a/
                     06073
   077.324           06074   CSLIBUF DS    101
   100.071           06075   CSLIBFE DS    0                END OF BUFFER
                     06076
                     06077   *    PATCH AREA
                     06078
   100.071  275 246 337 06079        DB       377Q-'B',377Q-'Y',377Q-' ',377Q-'G',377Q-'A',377Q-'C',377Q-' '
   100.100  266 261 337 06080        DB       377Q-'I',377Q-'N',377Q-' ',377Q-'R',377Q-'E',377Q-'M',377Q-'E'
   100.107  262 275 255 06081        DB       377Q-'M',377Q-'B',377Q-'R',377Q-'A',377Q-'N',377Q-'C',377Q-'E'
   100.116  337 260 271 06082        DB       377Q-' ',377Q-'O',377Q-'F',377Q-' ',377Q-'J',377Q-'G',377Q-'L'
   100.125  014      06083        DB       FF
                     06084
   100.126           06085   SECSCR       EQU      *                SYSTEM SCRATCH AREA
   100.126           06086        DS       512
                     06087
   102.126           06088   LWASYS       EQU      *                END OF MONITOR
   015.164           06089   LENSYS       EQU      LWASYS-FWASYS
```

```
                     06090
                     06091   *    PATCH AREA FOR RELOCATION TABLE
                     06092
  102.126            06093        DS       16
                     06094
                     06095
                     06096
                     06097        LON      G
                     06098        LON      C
  102.146  214 051 115 06099        END Statement Missing
           052 146 052
           200 052 206
           052 214 052
           222 052 241
           052 274 052
           277 052 307
           052 316 052
           321 052 324
           052 327 052
           332 052 335
           052 357 052
           365 052 016
           053 022 053
           374 053 005
           054 016 054
           024 054 033
           054 277 055
           152 056 216
           056 226 057
           243 057 342
           057 377 057
           011 060 045
           060 067 060
           075 060 115
           060 141 060
           144 060 155
           060 160 060
           171 060 200
           060 203 060
           211 060 214
           060 223 060
           357 064 364
           064 371 064
           010 065 034
           065 040 065
           044 065 064
           065 074 065
           077 065 106
           065 111 065
           121 065 131
           065 142 065
           147 065 157
           065 211 065
           216 065 223
           065 226 065
           231 065 241
           065 243 065
```

```
            245 065 247
            065 251 065
            253 065 255
            065 257 065
            261 065 263
            065 273 065
            277 065 307
            065 323 065
            332 065 341
            065 363 065
            371 065 020
            066 023 066
            026 066 031
            066 036 066
            062 066 077
            066 105 066
            113 066 167
            066 204 066
            226 066 231
            066 235 066
            241 066 256
            066 261 066
            275 066 304
            066 311 066
            314 066 325
            066 337 066
            352 066 022
            067 031 067
            034 067 042
            067 045 067
            055 067 060
            067 066 067
            073 067 107
            067 114 067
            121 067 133
            067 137 067
            152 067 157
            067 166 067
            171 067 203
            067 210 067
            221 067 224
            067 231 067
            236 067 243
            067 247 067
            274 067 304
            067 312 067
            317 067 322
            067 333 067
            336 067 343
            067 346 067
            352 067 357
            067 362 067
            367 067 372
            067 000 070
            007 070 012
            070 015 070
            022 070 027
```

```
            070 032 070
            036 070 042
            070 045 070
            051 070 060
            070 063 070
            073 070 076
            070 101 070
            106 070 116
            070 121 070
            133 070 141
            070 146 070
            153 070 171
            070 201 070
            237 070 244
            070 247 070
            254 070 261
            070 266 070
            273 070 276
            070 314 070
            321 070 333
            070 350 070
            361 070 366
            070 377 070
            003 071 006
            071 013 071
            023 071 032
            071 036 071
            045 071 065
            071 075 071
            111 071 116
            071 127 071
            136 071 144
            071 153 071
            163 071 174
            071 201 071
            205 071 215
            071 223 071
            231 071 236
            071 255 071
            300 071 303
            071 306 071
            333 071 336
            071 344 071
            351 071 004
            072 007 072
            012 072 016
            072 021 072
            032 072 035
            072 045 072
            051 072 131
            072 135 072
            161 072 204
            072 227 072
            246 072 252
            072 257 072
            264 072 307
            072 313 072
```

```
            330 072 342
            072 345 072
            355 072 360
            072 365 072
            110 073 115
            073 131 073
            144 073 165
            073 170 073
            231 073 266
            073 300 073
            312 073 346
            073 363 073
            375 073 014
            074 022 074
            025 074 035
            074 040 074
            043 074 046
            074 063 074
            066 074 077
            074 105 074
            111 074 134
            074 143 074
            154 074 162
            074 173 074
            176 074 212
            074 216 074
            221 074 227
            074 232 074
            235 074 240
            074 243 074
            252 074 255
            074 266 074
            312 074 315
            074 322 074
            326 074 331
            074 344 074
            376 074 003
            075 010 075
            110 075 115
            075 120 075
            130 075 134
            075 140 075
            214 075 217
            075 230 075
            235 075 250
            075 255 075
            326 075 337
            075 353 075
            360 075 365
            075 372 075
            003 076 025
            076 034 076
            062 076 067
            076 232 076
            304 076 356
            076 030 077
            102 077 154
```

```
            077 306 077
            310 077 312
            077 314 077
            317 077 000
            000


06099 Statements Assembled
27350 Bytes Free
00001 Errors Detected
```

```
  $CAD    062324      $CDEHL  030216      $CHL    030224      $COMP   030060      $CRLF   064041      $DAD    063170
  $DADA   030072      $DADA.  030101      $DDD    064004      $DDD1   064007      $DTB    062162      $DTB1   062166
  $DTB2   062175      $DTB3   062212      $DU66   030106      $HLIHL  030211      $INDL   030234      $INDLB  075066'
  $INDS   075107'     $INDSB  075143'     $MCU    062131      $MLU    062142      $MLU1   062145      $MOVE   030252
  $MOVEL  062223      $MU10   030324      $MU86   031007      $RCHAR  062246      $RSTALL 031047      $RTL    062266
  $RTL.   062257      $RTL1   062267      $RTL2   062321      $SAVALL 031054      $TBRA   031076      $TFN    064056
  $TFN.   064061      $TFN1   064074      $TYPC.  064053      $TYPCC  064226      $TYPCH  064047      $TYPEC. 064144
  $TYPET  064107      $TYPET. 064115      $TYPET1 064135      $TYPTX  031136      $TYPTX. 031144      $UDD    031157
  $WCHAR  062254      $WDR    031222      $WER    031241      $ZERO   031212      .       002000      .ABUSS  040024
  .ALARM  002136      .ALEDS  040013      .CHFLG  000060      .CLEAN  000205      .CLEAR  000055      .CLEARA 000056
  .CLOSE  000046      .CLRCO  000007      .CONSL  000006      .CRC    002347      .CRCSUM 040027      .CTC    002172
  .CTL2FL 040066      .CTLC   000041      .CTLFLG 040011      .DAD    000206      .DECODE 000053      .DELET  000050
  .DISMT  000061      .DLEDS  040021      .DLY    000053      .DMNMS  000203      .DMOUN  000201      .DOD    003122
  .DODA   003356      .DSPMOD 040007      .DSPROT 040006      .DUMP   001374      .ERROR  000057      .EXIT   000000
  .HORN   002140      .IDENT  000000      .IOWRK  040002      .LINK   000040      .LOAD   001267      .LOADD  000062
  .LOADO  000010      .MFLAG  040010      .MONMS  000202      .MOUNT  000200      .NAME   000054      .NMIRET 040064
  .OPEN   000063      .OPENC  000045      .OPENR  000042      .OPENU  000044      .OPENW  000043      .PCHL   002264
  .POSIT  000047      .PRINT  000003      .RCK    003260      .READ   000004      .REGI   040005      .REGPTR 040035
  .RENAM  000051      .RESET  000204      .RNB    002331      .RNP    002325      .SCIN   000001      .SCOUT  000002
  .SETTP  000052      .SRS    002265      .START  040000      .SYSRES 000012      .TICCNT 040033      .TPERR  002205
  .TPERRX 040031      .UIVEC  040037      .VERS   000011      .WNB    003024      .WNP    003017      .WRITE  000005
  ABP     072255'     AC.DLY  000156      ACA     072267'     ACA0    072271'     ACA1    072277'     ACA2    072316'
  ACA3    072332'     ACA8    072362'     ACA9    072374'     AGT     057224      AGT1    057254      AGT2    057330
  AGT3    057354      AGTA    057357      AGTB    057360      AIO.CGN 041047      AIO.CHA 041116      AIO.CNT 041111
  AIO.CSI 041050      AIO.DDA 041041      AIO.DES 041055      AIO.DEV 041057      AIO.DIR 041062      AIO.DTA 041053
  AIO.EOF 041113      AIO.EOM 041112      AIO.FLG 041043      AIO.GRT 041044      AIO.LGN 041051      AIO.LSI 041052
  AIO.SPG 041046      AIO.TFP 041114      AIO.UNI 041061      AIO.VEC 041040      BELL    000007      BFLG.A  000001
  BKSP    000010      BND     073007'     BND.    073117'     BND1    073105'     BND2    073123'     BOOT.P  000001
  BOOTA   037132      BOOTABT 057156      BOOTAL  000130      BOOTERR 057111      BTS     073217'     BUFF    052217
  BUFFE   053217      C.DSYN  000375      C.STX   000002      C.SYN   000026      CAD0    062346      CAD1    063010
  CAD2    063044      CAD3    063047      CADA    063112      CADB    063157      CADBL   000011      CB.CLI  000100
  CB.MTL  000040      CB.SPK  000200      CB.SSI  000020      CB2.CLI 000002      CB2.ORG 000040      CB2.SID 000100
  CB2.SSI 000001      CC.FLG  000003      CDB.H84 000001      CDB.H85 000000      CFF     031354      CHANCNT 000006
  CHANTAB 076232'     CLOCK   034031      CLRCO   072002'     CN.170M 000014      CN.174M 000003      CN.ABO  000200
  CN.BAU  000100      CN.MEM  000040      CN.PRI  000020      CND.H17 000000      CND.H47 000001      CND.NDI 000000
  CO.FLG  000001      CONSL   071354'     CPA     073230'     CR      000015      CRLF    071225'     CRM     071242'
  CS.FLG  000200      CSL.CHR 000001      CSL.ECH 000200      CSL.RAW 000004      CSL.WRP 000002      CSLDCA  077317'
  CSLDLY  077316'     CSLIBFE 100071'     CSLIBUF 077324'     CSLLCNT 077302'     CSLRBF  077304'     CTLA    000001
  CTLB    000002      CTLC    000003      CTLD    000004      CTLO    000017      CTLP    000020      CTLQ    000021
  CTLS    000023      CTLZ    000032      CTP.2SB 000010      CTP.BKM 000002      CTP.BKS 000200      CTP.FF  000100
  CTP.MLI 000040      CTP.MLO 000020      CTP.TAB 000001      CZ.FLG  000010      D.CON   040110      D.RAM   040240
  D.VEC   040130      DAD1    063313      DAD2    063316      DADB    063327      DADC    063373      DADCL   000011
  DC.ABT  000007      DC.CLO  000006      DC.LOD  000011      DC.MAX  000013      DC.MOU  000010      DC.OPR  000003
  DC.OPU  000005      DC.OPW  000004      DC.RDY  000012      DC.REA  000000      DC.RER  000002      DC.WRI  000001
  DCA     032002      DDF.BOL 000011      DDF.BOO 000000      DDF.LAB 000011      DDF.USR 000012      DEBUG   000001
  DEV.DDA 000004      DEV.DVG 000015      DEV.DVL 000013      DEV.FLG 000006      DEV.JMP 000003      DEV.MNU 000010
  DEV.MUM 000007      DEV.NAM 000000      DEV.RES 000002      DEV.UNT 000011      DEVCNT  000007      DEVELEN 000016
  DEVLST  076056'     DF.CLR  000376      DF.DI   000040      DF.DS0  000002      DF.DS1  000004      DF.DS2  000010
  DF.EMP  000377      DF.HD   000001      DF.MO   000020      DF.SD   000010      DF.ST   000100      DF.T0   000002
  DF.WG   000001      DF.WP   000004      DF.WR   000200      DIF.CNT 000020      DIF.LOC 000100      DIF.SYS 000200
  DIF.WP  000040      DIR.ALD 000025      DIR.CLU 000015      DIR.CRD 000023      DIR.EXT 000010      DIR.FGN 000020
  DIR.FLG 000016      DIR.LGN 000021      DIR.LSI 000022      DIR.NAM 000000      DIR.PRO 000013      DIR.VER 000014
  DIREAD  072107'     DIREAD1 072140'     DIRELEN 000027      DIRIDL  000015      DIS.ENL 001373      DIS.ENT 000000
  DIS.LNK 001376      DIS.SEC 001374      DIWRITE 072167'     DM.MR   000000      DM.MW   000001      DM.RR   000002
  DM.RW   000003      DP.DC   000177      DR.IM   000001      DR.PR   000002      DT.CH   000020      DT.CR   000002
  DT.CW   000004      DT.DD   000001      DT.RN   000010      DV.EL   000000      DV.NU   000001      DVD.CAP 000007
  DVD.DVD 000006      DVD.ENT 002000      DVD.INP 000023      DVD.MNU 000011      DVD.MUM 000010      DVD.SET 000022
```

```
  DVD.STE 000053      DVD.UFL 000012      DVDFLV  000307      DWRIT1  072152'     DWRIT2  072206'     EC.CNA  000004
  EC.DDA  000027      EC.DIF  000017      EC.DIW  000035      EC.DNI  000045      EC.DNR  000046      EC.DNS  000005
  EC.DSC  000047      EC.EOF  000001      EC.EOM  000002      EC.FAO  000031      EC.FAP  000026      EC.FL   000030
  EC.FNF  000014      EC.FNO  000011      EC.FNR  000034      EC.FOD  000043      EC.FUC  000013      EC.ICN  000016
  EC.IDN  000006      EC.IFC  000020      EC.IFN  000007      EC.ILC  000003      EC.ILO  000040      EC.ILR  000012
  EC.ILV  000037      EC.IOI  000052      EC.IS   000032      EC.NCV  000050      EC.NEM  000021      EC.NOS  000051
  EC.NPM  000044      EC.NRD  000010      EC.NVM  000042      EC.OTL  000053      EC.RF   000022      EC.UNA  000036
  EC.UND  000015      EC.UUN  000033      EC.VPM  000041      EC.WF   000023      EC.WP   000025      EC.WPV  000024
  ECI     053173      ECI1    053210      EDL     055110      EDL0    055267      EDL0.5  055316      EDL1    055343
  EDL2    056017      EDL3    056047      EDL5    056063      EDLB    056133      EDLC    056135      EDLCAP  056154
  EDLD    056137      EDLDEV  056146      EDLDVG  056163      EDLDVL  056161      EDLMNU  056156      EDLMUM  056155
  EDLNAM  056146      EDLPTR  056157      ENL     000212      EOFFLG  076011'     ERR.FNO 031344      ERR.ILR 031350
  ESC     000033      EXIT    066171'     EXIT1   066217'     EXIT2   066237'     EXIT3   066260'     EXIT4   066324'
  EXIT5   066341'     EXITA   066373'     EXITB   066354'     EXITC   067013'     FATSER1 066154'     FATSERR 066115'
  FCI     073256'     FCI1    073263'     FCI2    073307'     FCI3    073352'     FF      000014      FFB     032133
  FFL     032205      FLT.BOP 000011      FLT.CBD 000007      FLT.CDB 000006      FLT.CFC 000002      FLT.CRF 000003
  FLT.CTY 000000      FLT.CWI 000001      FLT.MNC 000004      FLT.PBO 000013      FLT.SAL 000012      FSM     053270
  FT.ABS  000000      FT.BAC  000003      FT.DD   000001      FT.OC   000020      FT.OR   000002      FT.OU   000010
  FT.OW   000004      FT.PIC  000001      FT.REL  000002      FWAREL  064342      FWASYS  064342'     GSP     073362'
  GSPA    073377'     GUP     075035'     GVM     053215      H17SDL  000173      HIGHDAT 077300'     HLCPDE  075053'
  HOSB2   051127      HOSBA   051133      HOSBOO1 051025      HOSBOOT 051006      HOSTAB  051011      I.CONFL 000004
  I.CONTY 000001      I.CONWI 000003      I.CSLMD 000000      I.CUSOR 000002      ILDEHL  075061'     IOC.CGN 000010
  IOC.CSI 000011      IOC.DDA 000002      IOC.DES 000016      IOC.DEV 000020      IOC.DIL 000021      IOC.DIR 000023
  IOC.DRL 000010      IOC.DTA 000014      IOC.FLG 000004      IOC.GRT 000005      IOC.LGN 000012      IOC.LNK 000000
  IOC.LSI 000013      IOC.SPG 000007      IOC.SQL 000003      IOC.UNI 000022      IOCCTD  000001      IOCELEN 000052
  IP.CON  000362      IP.PAD  000360      ISDEHL  075164'     ISY     075202'     ISY1    075240'     ISY2    075261'
  LAB.AUX 000117      LAB.AXL 000001      LAB.DAT 000000      LAB.DIS 000003      LAB.GRT 000005      LAB.IND 000001
  LAB.LAB 000021      LAB.LBL 000074      LAB.NOD 000002      LAB.PSS 000016      LAB.RGT 000012      LAB.SER 000000
  LAB.SIZ 000014      LAB.SPG 000007      LAB.SPT 000117      LAB.SYS 000001      LAB.VER 000011      LAB.VFL 000020
  LAB.VLT 000010      LAB.VPL 000005      LAB.VPR 000014      LABEL   051217      LABELE  052217      LDD     074001'
  LDD10   074226'     LDD2    074065'     LDD3    074104'     LDD4    032323      LDD8    074136'     LDD8A   074204'
  LDD8B   074206'     LDD8C   074207'     LDD9    074211'     LDD9A   074260'     LDD9B   074261'     LDE     057364
  LDE.    057361      LDE..   057372      LDE3    060013      LDE3.5  060033      LDO     033012      LDON    065265'
  LDON0   065343'     LDON1   065345'     LDON2   065351'     LDON3   065354'     LDON4   066040'     LDON5   066101'
  LDON6   066107'     LENSYS  015164      LF      000012      LOADO   072030'     LOADO1  072076'     LOADO2  072100'
  LSD     060063      LSO     053356      LSO.    054071      LSO..   054306      LSO1    054164      LSO2    054213
  LSOA    054332      LSOB    054347      LWASYS  102126'     M.CDCA  000017      M.CDLY  000016      M.CFWA  000012
  M.CIN   000006      M.CINT  000005      M.CLWA  000014      M.COUT  000010      M.CPRE  000003      M.CRUB  000004
  M.CSLC  000002      M.FOX   000303      M.PAM8  000021      M.SALO  000001      M.SUNI  000021      M.SYDD  000022
  M.SYSM  000000      MI.CPI  000376      MI.JMP  000303      MI.RET  000311      MSD     056164      MSD1    056175
  MSD2    056223      MSD3    056232      MSD5    057034      MSDA    057104      MSDB    052000      MSYDD   077322'
  NL      000012      NUL2    000000      NULL    000200      OP.CTL  000360      OP.DIG  000360      OP.SEG  000361
  OP2.CTL 000362      OTI     074262'     OVBUFE  064342      OVL.COD 000000      OVL.ENS 000010      OVL.ENT 000004
  OVL.FLB 000006      OVL.IN  000001      OVL.NUM 000014      OVL.RES 000002      OVL.SIZ 000002      OVL.UCS 000200
  OVL0    000000      OVL1    000001      OVLCNT  000002      OVLMAX  000002      OVLTAB  076036'     PATCH   064242
  PBO.DAT 000001      PCHL    032361      PDI     033145      PIC.COD 000006      PIC.ID  000000      PIC.LEN 000002
  PIC.PTR 000004      PRINT   071340'     PSC     070167'     PSC1    070175'     PSC2    070226'     PSC3    070270'
  QUOTE   000047      RDL     060231      RDL1    060312      READ    071254'     REL     033177      REL.    033175
  ROMBOOT 030000      RRC     074311'     RRC1    074333'     RRH     051217      RRH1    051224      RRH1.5  051237
  RRH2    051240      RRH2.5  051240      RRH3    052003      RRH4    052042      RRH5    052076      RRH6    052066
  RUBOUT  000177      RUC     033257      S.BAUD  040344      S.BDA   041120      S.BOOTF 041034      S.CAADR 040333
  S.CACC  041006      S.CCTAB 040335      S.CDB   040343      S.CFWA  040352      S.CODE  041007      S.CONFL 040332
  S.CONTY 040327      S.CONWI 040331      S.CSLMD 040326      S.CUSOR 040330      S.DATC  040310      S.DATE  040277
  S.DCS   041033      S.DDDTA 040366      S.DDGRP 040364      S.DDLDA 040360      S.DDLEN 040362      S.DDOPC 040370
  S.DFWA  040354      S.DIREA 041016      S.DLINK 040346      S.FASER 041013      S.FCI   041021      S.GRT0  024000
  S.GRT1  025000      S.GRT2  026000      S.GUP   041027      S.HIMEM 040316      S.INT   040343      S.JUMPS 041010
  S.MOUNT 041032      S.OFWA  040350      S.OMAX  040324      S.OSN   041004      S.OVLE  041000      S.OVLFL 040371
```

```
  S.OVLS  040376      S.OVSTK 041035      S.READ  031275      S.RFWA  040356      S.SCI   041024      S.SCR   041121
  S.SDD   041010      S.SOVR  041146      S.SSN   041002      S.SYSM  040320      S.TIME  040312      S.UCSF  040372
  S.UCSL  040374      S.USRM  040322      S.VAL   040277      S.WRITE 031330      SALONE  077301'     SB.BAU  042205
  SB.BOO  042200      SB.BPE  042240      SB.DAT  042207      SB.DRV  042240      SB.FLG  042204      SB.ORG  051000
  SB.OVMX 014000      SB.SDB  044200      SB.VER  042203      SBD     051154      SC.ACE  000350      SC.UART 000372
  SCD     053021      SCDLY   071204'     SCDLY1  071207'     SCI     074347'     SCIFWA  077312'     SCIIN   077306'
  SCILWA  077314'     SCIN    067014'     SCIN1   067027'     SCIN2   067041'     SCIN2.5 067104'     SCIN3   067120'
  SCIN4.3 067314'     SCIN4.5 067321'     SCINI   067130'     SCINI0  067144'     SCINI01 067161'     SCINI02 067163'
  SCINI10 070124'     SCINI11 070136'     SCINI2  067214'     SCINI3  067223'     SCINI35 067245'     SCINI4  067266'
  SCINI5  067355'     SCINI6  067361'     SCINI65 070003'     SCINI7  070011'     SCINI8  070034'     SCINI9  070062'
  SCINI93 070113'     SCINI95 070120'     SCINIA  070155'     SCINTFL 077305'     SCIOUT  077310'     SCIPRE  077303'
  SCOUT   070304'     SCOUT0  070307'     SCOUT1  070326'     SCOUT10 071166'     SCOUT11 071170'     SCOUT2  070345'
  SCOUT3  070363'     SCOUT4  071002'     SCOUT45 071015'     SCOUT5  071027'     SCOUT6  071042'     SCOUT7  071060'
  SCOUT8  071101'     SCOUT85 071120'     SCOUT9  071121'     SCOUT91 071131'     SCOUT92 071146'     SCOUT95 071160'
  SCOUTA  071161'     SCU     053036      SCU1    053106      SDD     074373'     SDD1    075012'     SDT     054364
  SDT1    054372      SDT2    055017      SDT3    055063      SDT4    055064      SDTA    024000      SDTAE   026000
  SDTB    055075      SDTBL   000013      SDV     052337      SECSCR  100126'     SLR     052114      SLRA    052305
  SLRAL   000010      SLRB    052315      SLRBL   000022      SRR     052113      SSD     060377      SSD0    061015
  SSD1    061062      SSD1.5  061107      SSD2    061114      SSD3    061153      SSD4    061163      SSD5    061217
  SSD6    061225      SSD7    061307      SSD8    061310      SSDA    061322      SSDB    052217      STACK   042200
  STACKL  001032      STB     075016'     SUNIT   077321'     SYDD    040130      SYS0    064373'     SYS1    065015'
  SYS2    065025'     SYS3    065033'     SYS4    065055'     SYS5    065101'     SYSCAL  064342'     SYSCAL0 065127'
  SYSCAL1 065156'     SYSCAL2 065176'     SYSCAL3 065220'     SYSCAL4 065222'     SYSCALA 065241'     SYSCALL 000377
  SYSMODE 077300'     TAB     000011      TDD     060316      TDD.    060314      TDD1    060317      TDD2    060336
  TDDA    060365      TFE     033233      TTABT   075310'     TTDEV   076056'     TTDVD   075272'     TTDVD0  075275'
  TTNOP   075331'     TTOPE   075314'     TTOUNT  076222'     TTR1    075346'     TTR2    075333'     TTR4    075374'
  TTREAD  075336'     TTREOF  075367'     TTW2    076031'     TTWRITE 076012'     TYPEC1  064155      TYPEC2  064172
  TYPEC3  064204      TYPEC4  064211      UBP     061323      UBP1    062055      UBPA    062074      UBPAL   000035
  UC.2SB  000004      UC.5BW  000000      UC.6BW  000001      UC.7BW  000002      UC.8BW  000003      UC.BI   000020
  UC.CTS  000020      UC.DCS  000001      UC.DDR  000002      UC.DLA  000200      UC.DR   000001      UC.DRL  000010
  UC.DSR  000040      UC.DTR  000001      UC.EDA  000001      UC.EPS  000020      UC.FE   000010      UC.IID  000006
  UC.IIP  000001      UC.LOO  000020      UC.MSI  000010      UC.OR   000002      UC.OU1  000004      UC.OU2  000010
  UC.PE   000004      UC.PEN  000010      UC.RI   000100      UC.RLS  000200      UC.RSI  000004      UC.RTS  000002
  UC.SB   000100      UC.SKP  000040      UC.TER  000004      UC.THE  000040      UC.TRE  000002      UC.TSE  000100
  UCI.ER  000020      UCI.IE  000002      UCI.IR  000100      UCI.RE  000004      UCI.RO  000040      UCI.TE  000001
  UDR     000000      UF.FCT  000100      UF.RDA  000001      UF.ROR  000002      UF.RPE  000004      UF.TBM  000200
  UMI.16X 000002      UMI.1B  000100      UMI.1X  000001      UMI.2B  000300      UMI.64X 000003      UMI.HB  000200
  UMI.L5  000000      UMI.L6  000004      UMI.L7  000010      UMI.L8  000014      UMI.PA  000020      UMI.PE  000040
  UNT.DIS 000006      UNT.FLG 000000      UNT.GRT 000002      UNT.GTS 000004      UNT.SIZ 000010      UNT.SPG 000001
  UO.CLK  000001      UO.DDU  000002      UO.HLT  000200      UO.NFR  000100      UP.DP   000174      UP.FC   000175
  UP.SC   000176      UP.SR   000176      UP.ST   000175      UR.DLL  000000      UR.DLM  000001      UR.IER  000001
  UR.IIR  000002      UR.LCR  000003      UR.LSR  000005      UR.MCR  000004      UR.MSR  000006      UR.RBR  000000
  UR.THR  000000      USERFWA 042200      USR     000001      USR.BD  000100      USR.FE  000040      USR.OE  000020
  USR.PE  000010      USR.RXR 000002      USR.TXE 000004      USR.TXR 000001      VERS    000040      VERSN   072103'
  VFL.NSD 000001      WRITE   071305'     XCHGBC  075171'
```

```
  $CAD    062324       2474      2803L
  $CDEHL  030216       2568E     2635
  $CHL    030224       1830      5774E
  $COMP   030060       1748      2183      2421      2523      2806      2844      5701E
  $CRLF   064041       1948      1994      3077L
  $DAD    063170       2413      2433      2489      2918L
  $DADA   030072       2346      3065E
  $DADA.  030101       1761      2853      2950      3460      3905      4079      4474      4687      5262      5423
                       5648E
  $DDD    064004       2814      2872      3017L
  $DDD1   064007       3019L     3036
  $DTB    062162       2622L
  $DTB1   062166       2626L     2629
  $DTB2   062175       2634L     2639
  $DTB3   062212       2636      2643L
  $DU66   030106       2926      2940      2999E     5370
  $HLIHL  030211       3490      3604      3610      4080      4553      5414      5458E
  $INDL   030234       2247      2253      5495E
  $INDLB  075066'      2063      2071      2268      4980      5510L     5802
  $INDS   075107'      2129      2261      5543L
  $INDSB  075143'      1852      2271      5573L
  $MCU    062131       2582L     2604
  $MLU    062142       2599L     2748
  $MLU1   062145       2602L     2607
  $MOVE   030252       1229      1644      1657      1902      2513      2529      2696      2953      2978      4858
                       5001      5324      5638E
  $MOVEL  062223       1308      1349      2680L
  $MU10   030324       2726E     3027
  $MU86   031007       1675      1799      1829      3244E     4881
  $RCHAR  062246       2707L     2708      2752
  $RSTALL 031047       5325      5666E
  $RTL    062266       2746      2750E
  $RTL.   062257       2459      2746L
  $RTL1   062267       2752L     2758
  $RTL2   062321       2754      2775L
  $SAVALL 031054       5062      5317      5680E
  $TBRA   031076       5397E     5842
  $TFN    064056       1658      3130L
  $TFN.   064061       3131L
  $TFN1   064074       3132      3137L     3142
  $TYPC.  064053       3117L     3139
  $TYPCC  064226       1893      2452      3258E     3267
  $TYPCH  064047       3107L     3133
  $TYPEC. 064144       2357      3170      3194L     3224
  $TYPET  064107       1198      1202      1207      3163L     3180
  $TYPET. 064115       3164      3168L     3176
  $TYPET1 064135       3172      3180L
  $TYPTX  031136       1542      1637      1652      1680      1889      1894      1908      2002      2010      2019
                       2027      2440      2446      2456      2479      3095E     3651      4822
  $TYPTX. 031144       3097E
  $UDD    031157       2934      2964      3052E
  $WCHAR  062254       2711L
  $WDR    031222       3423      5761E
  $WER    031241       1686      1725      3446      5750E
  $ZERO   031212       1101      5737E
  .       002000       4775S     4776      5210S     5211
  .ABUSS  040024        338E
```

```
  .ALARM  002136        311E
  .ALEDS  040013        336E
  .CHFLG  000060        449L
  .CLEAN  000205        464L
  .CLEAR  000055        446L     3726
  .CLEARA 000056        447L     3701
  .CLOSE  000046        439L
  .CLRCO  000007        423L     4821
  .CONSL  000006        422L
  .CRC    002347        319E
  .CRCSUM 040027        339E
  .CTC    002172        313E
  .CTL2FL 040066        345E
  .CTLC   000041        434L
  .CTLFLG 040011        335E
  .DAD    000206        465L     3736
  .DECODE 000053        444L
  .DELET  000050        441L
  .DISMT  000061        450L
  .DLEDS  040021        337E
  .DLY    000053        308E     1502
  .DMNMS  000203        462L
  .DMOUN  000201        460L     3751
  .DOD    003122        322E
  .DODA   003356        324E
  .DSPMOD 040007        333E
  .DSPROT 040006        332E
  .DUMP   001374        310E
  .ERROR  000057        448L     3748      3755
  .EXIT   000000        416L     1124      1364
  .HORN   002140        312E
  .IDENT  000000        307E
  .IOWRK  040002        330E
  .LINK   000040        433L     1121      3447      3714
  .LOAD   001267        309E
  .LOADD  000062        451L     3358
  .LOADO  000010        424L     3731      3734
  .MFLAG  040010        334E     1304      3654      3692
  .MONMS  000202        461L     3349
  .MOUNT  000200        459L     1991      3347      3476
  .NAME   000054        445L
  .NMIRET 040064        344E
  .OPEN   000063        452L
  .OPENC  000045        438L
  .OPENR  000042        435L
  .OPENU  000044        437L
  .OPENW  000043        436L
  .PCHL   002264        315E
  .POSIT  000047        440L
  .PRINT  000003        419L     3724      3745
  .RCK    003260        323E
  .READ   000004        420L
  .REGI   040005        331E
  .REGPTR 040035        342E
  .RENAM  000051        442L
  .RESET  000204        463L     3351
  .RNB    002331        318E
```

```
  .RNP    002325        317E
  .SCIN   000001        417L     2707      4827      5897
  .SCOUT  000002        418L     2711      3078      3117      3264      5929
  .SETTP  000052        443L
  .SRS    002265        316E
  .START  040000        329E
  .SYSRES 000012        426L     3449
  .TICCNT 040033        341E
  .TPERR  002205        314E
  .TPERRX 040031        340E
  .UIVEC  040037        343E     1302      1430
  .VERS   000011        425L
  .WNB    003024        321E
  .WNP    003017        320E
  .WRITE  000005        421L
  ABP     072255'      3808      3979      4719L
  AC.DLY  000156         44E     1501
  ACA     072267'      4692      4748L
  ACA0    072271'      4749L     4788
  ACA1    072277'      4753L     4789
  ACA2    072316'      4758      4760L
  ACA3    072332'      4756      4774L
  ACA8    072362'      4780      4794L
  ACA9    072374'      4769      4781      4796      4802L
  AGT     057224       2062L     2305
  AGT1    057254       2067      2078L
  AGT2    057330       2123L     2135
  AGT3    057354       2124      2137L
  AGTA    057357       2066      2090      2141L
  AGTB    057360       2073      2106      2142L
  AIO.CGN 041047       1003L     4805      4807
  AIO.CHA 041116       1018L     1100      4685      4986      5318
  AIO.CNT 041111       1014L     4803
  AIO.CSI 041050       1004L     4807
  AIO.DDA 041041        999E     4997
  AIO.DES 041055       1008L     2160      2167      2197
  AIO.DEV 041057       1009L
  AIO.DIR 041062       1012L     1643      1656      2177      3130
  AIO.DTA 041053       1007L     2118      2233
  AIO.EOF 041113       1016L     4606      4674
  AIO.EOM 041112       1015L     4694      4798
  AIO.FLG 041043       1000L     5323
  AIO.GRT 041044       1001L     1584
  AIO.LGN 041051       1005L     4749      4750      4774      4775      4804      4806
  AIO.LSI 041052       1006L     4750      4753      4776      4806
  AIO.SPG 041046       1002L     1587      4754
  AIO.TFP 041114       1017L     4751
  AIO.UNI 041061       1010L     1150      1353      3697      4833      4852      4982      5793      5806      5815
                       5832
  AIO.VEC 041040        998L     1348      4381      4415      4638
  BELL    000007        207E     1208      1208      1890      1909      2003      2020      2480      3652      3652
                       3757     3988
  BFLG.A  000001       1036E
  BKSP    000010        209E     3926      4030      4034      4207
  BND     073007'      3738      4821L
  BND.    073117'      4832L
  BND1    073105'      4827L     4828      4830
```

```
  BND2    073123'      4835L     4837
  BOOT.P  000001        978E     1117
  BOOTA   037132        385E     4856
  BOOTABT 057156       2006      2012      2021      2027L
  BOOTAL  000130        386E     4855
  BOOTERR 057111       1570      1579      1639      1659      1682      2019L
  BTS     073217'      4878L     5081      5235
  BUFF    052217       1685      1688      1709      1800      1806      1815      1818      1820      1822      1827
                       1841      1843      1844      1866      3293E     3294      3299
  BUFFE   053217       1614      1710      3294E
  C.DSYN  000375        197E
  C.STX   000002        211E
  C.SYN   000026        210E
  CAD0    062346       2807      2813L
  CAD1    063010       2841L     2857
  CAD2    063044       2839      2861L     2870      2873      2876      2879      2881
  CAD3    063047       2846      2867L
  CADA    063112       2840      2888L
  CADB    063157       2805      2891L     2892
  CADBL   000011       2804      2892E
  CB.CLI  000100        253E      276
  CB.MTL  000040        252E
  CB.SPK  000200        254E
  CB.SSI  000020        251E
  CB2.CLI 000002        257E
  CB2.ORG 000040        258E
  CB2.SID 000100        259E
  CB2.SSI 000001        256E
  CC.FLG  000003       3888      6050E
  CDB.H84 000001        921E     1451      1518      3196      3855      4250
  CDB.H85 000000        920E     1076
  CFF     031354        363E     4785
  CHANCNT 000006       5999E
  CHANTAB 076232'      1332      6001E
  CLOCK   034031        388E
  CLRCO   072002'      3513      4498L
  CN.170M 000014        294E
  CN.174M 000003        293E
  CN.ABO  000200        298E
  CN.BAU  000100        297E
  CN.MEM  000040        296E
  CN.PRI  000020        295E
  CND.H17 000000        300E
  CND.H47 000001        302E
  CND.NDI 000000        301E
  CO.FLG  000001        898E     4043      4044      4142      5862
  CONSL   071354'      3512      4470E
  CPA     073230'      3424      3847      4915L
  CR      000015        203E     1071      3181      3216      3228      4004      4204      4319
  CRLF    071225'      3972      4201      4233      4318L
  CRM     071242'      3867      4002      4198      4340L
  CS.FLG  000200        899E     4045      4047      4047      4139
  CSL.CHR 000001        875E     3780      3917
  CSL.ECH 000200        872E     4021
  CSL.RAW 000004        873E     4343
  CSL.WRP 000002        874E
  CSLDCA  077317'      1405      6067L
```

```
  CSLDLY  077316'      1403      4293      6065L
  CSLIBFE 100071'      6062      6075L
  CSLIBUF 077324'      4499      6056      6058      6060      6074L
  CSLLCNT 077302'      3795      3831      4014      4503      6044L
  CSLRBF  077304'      3942      3945      3952      3955      4504      6048L
  CTLA    000001        218E
  CTLB    000002        219E
  CTLC    000003        220E
  CTLD    000004        221E      2753      3825      3997      5302      5899
  CTLO    000017        222E      3894
  CTLP    000020        223E
  CTLQ    000021        224E      4110
  CTLS    000023        225E
  CTLZ    000032        226E      3879      4098      4100
  CTP.2SB 000010        884E      1065      1143      1464      1465      1495      1497
  CTP.BKM 000002        885E      1065      3924
  CTP.BKS 000200        880E      3939
  CTP.FF  000100        881E      4242
  CTP.MLI 000040        882E      1065      1339      3820      3821
  CTP.MLO 000020        883E      1065      1339      4170
  CTP.TAB 000001        886E      4184
  CZ.FLG  000010       3888       4070      6051E
  D.CON   040110        833L      4857
  D.RAM   040240        836L
  D.VEC   040130        835L
  DAD1    063313       2932       2945      2947      2970L
  DAD2    063316       2920       2976L
  DADB    063327       2949       2980L
  DADC    063373       2977       2982L     2983
  DADCL   000011       2976       2983E
  DC.ABT  000007        624L      1560      3698
  DC.CLO  000006        623L
  DC.LOD  000011        626L      2240      5339
  DC.MAX  000013        628L      5855
  DC.MOU  000010        625L      1577      1984      2532      2550      4840
  DC.OPR  000003        620L      5342
  DC.OPU  000005        622L      5344
  DC.OPW  000004        621L
  DC.RDY  000012        627L      4835
  DC.REA  000000        617L      4380      4611
  DC.RER  000002        619L      1568      4847
  DC.WRI  000001        618L      2539      4414      4661      4702
  DCA     032002        365E      4604      4672
  DDF.BOL 000011        691E      4844
  DDF.BOO 000000        690L      4846
  DDF.LAB 000011        692L      1567
  DDF.USR 000012        693L
  DEBUG   000001          3E      1090      1181      1424      3376      3431
  DEV.DDA 000004        502L      1924      2254      2262      5971
  DEV.DVG 000015        515L      1936      2064      2072      2299      5983
  DEV.DVL 000013        514L      1934      2294      2299      5981
  DEV.FLG 000006        503L      1926      2064      5973
  DEV.JMP 000003        501L      1922      5969
  DEV.MNU 000010        511L      1930      2072      5803      5977
  DEV.MUM 000007        510L      1928      5975
  DEV.NAM 000000        493L      1918      5965
  DEV.RES 000002        497L      1920      2234      2269      2272      2294      5967
```

```
  DEV.UNT 000011        512L     1932      2119      5022      5979
  DEVCNT  000007       1875      1876      5960E     5988
  DEVELEN 000016        517E     1877      1901      1938      1975      5985      5988
  DEVLST  076056'      1334      5962L
  DF.CLR  000376        539E     1737
  DF.DI   000040        166E
  DF.DS0  000002        162E
  DF.DS1  000004        163E
  DF.DS2  000010        164E
  DF.EMP  000377        538E     1734
  DF.HD   000001        156E
  DF.MO   000020        165E
  DF.SD   000010        159E
  DF.ST   000100        167E
  DF.T0   000002        157E
  DF.WG   000001        161E
  DF.WP   000004        158E
  DF.WR   000200        168E
  DIF.CNT 000020        772E     1667      4689
  DIF.LOC 000100        770E
  DIF.SYS 000200        769E
  DIF.WP  000040        771E
  DIR.ALD 000025        554L
  DIR.CLU 000015        547L
  DIR.CRD 000023        553L
  DIR.EXT 000010        542L
  DIR.FGN 000020        550L     1669      1789
  DIR.FLG 000016        548L     1664      1669      4686
  DIR.LGN 000021        551L
  DIR.LSI 000022        552L
  DIR.NAM 000000        541L     1643      1656      2177      3130
  DIR.PRO 000013        543L
  DIR.VER 000014        544L
  DIREAD  072107'      1374      4382      4599E     4614
  DIREAD1 072140'      4612      4633L     4662      4703
  DIRELEN 000027        556E      564       604      1012      2188
  DIRIDL  000015        545E     1642      1645      1655      1694      1696      2157
  DIS.ENL 001373        568L     1760
  DIS.ENT 000000        563E     2173
  DIS.LNK 001376        570L     1715      1719      2196
  DIS.SEC 001374        569L
  DIWRITE 072167'      4416      4667E
  DM.MR   000000        266E
  DM.MW   000001        267E
  DM.RR   000002        268E
  DM.RW   000003        269E
  DP.DC   000177        154E
  DR.IM   000001        498E     5968
  DR.PR   000002        499E     2270      5968
  DT.CH   000020        508E
  DT.CR   000002        505E     5974      5994
  DT.CW   000004        506E     5974      5994
  DT.DD   000001        504E     2065
  DT.RN   000010        507E
  DV.EL   000000        494E     1882      1957
  DV.NU   000001        495E
  DVD.CAP 000007        755L     1818      1841
```

```
  DVD.DVD 000006        754L     1815
  DVD.ENT 002000        764E     1867      5083      5110      5210      5211
  DVD.INP 000023        760L
  DVD.MNU 000011        757L     1822      1827      1844
  DVD.MUM 000010        756L     1820
  DVD.SET 000022        759L
  DVD.STE 000053        762E
  DVD.UFL 000012        758L     1843
  DVDFLV  000307        750E     1816
  DWRIT1  072152'      4659L     4676
  DWRIT2  072206'      4681E     4705
  EC.CNA  000004         638L
  EC.DDA  000027         657L     5857
  EC.DIF  000017         649L
  EC.DIW  000035         663L
  EC.DNI  000045         671L
  EC.DNR  000046         672L
  EC.DNS  000005         639L
  EC.DSC  000047         673L     2000
  EC.EOF  000001         635L     5904      5913
  EC.EOM  000002         636L     4797
  EC.FAO  000031         659L
  EC.FAP  000026         656L
  EC.FL   000030         658L
  EC.FNF  000014         646L     2201
  EC.FNO  000011         643L
  EC.FNR  000034         662L
  EC.FOD  000043         669L
  EC.FUC  000013         645L
  EC.ICN  000016         648L     4972
  EC.IDN  000006         640L
  EC.IFC  000020         650L
  EC.IFN  000007         641L
  EC.ILC  000003         637L     3451
  EC.ILO  000040         666L
  EC.ILR  000012         644L
  EC.ILV  000037         665L
  EC.IOI  000052         676L     3640
  EC.IS   000032         660L
  EC.NCV  000050         674L
  EC.NEM  000021         651L     4557
  EC.NOS  000051         675L     3596
  EC.NPM  000044         670L
  EC.NRD  000010         642L     2307      2321
  EC.NVM  000042         668L     3753
  EC.OTL  000053         677L     3617
  EC.RF   000022         652L
  EC.UNA  000036         664L
  EC.UND  000015         647L
  EC.UUN  000033         661L
  EC.VPM  000041         667L
  EC.WF   000023         653L     4664
  EC.WP   000025         655L     2542
  EC.WPV  000024         654L
  ECI     053173        1432      1517L     1541
  ECI1    053210        1519      1529L
  EDL     055110        1755      1780L
```

```
  EDL0     055267       1846L     1860
  EDL0.5   055316       1847      1862E
  EDL1     055343       1879L     1885
  EDL2     056017       1891L     1910
  EDL3     056047       1883      1900L
  EDL5     056063       1808      1811      1817      1869      1908L
  EDLB     056133       1891      1912L
  EDLC     056135       1913L
  EDLCAP   056154       1819      1927L
  EDLD     056137       1785      1914L
  EDLDEV   056146       1900      1917E     1918      1920      1922      1924      1926      1928      1930      1932
                        1934      1936      1938
  EDLDVG   056163       1792      1937L
  EDLDVL   056161       1870      1935L
  EDLMNU   056156       1823      1931L
  EDLMUM   056155       1821      1929L
  EDLNAM   056146       1784      1919L
  EDLPTR   056157       1837      1933L
  ENL      000212        216E     1546      1638      1681      1895      2005      2011      2020      2480      3652
                        3759      3760
  EOFFLG   076011'      5870      5884      5905      5918L
  ERR.FNO  031344        359E     4373      4407
  ERR.ILR  031350        361E     4377      4410      4472
  ESC      000033        214E
  EXIT     066171'      3500      3687E     4112
  EXIT1    066217'      3693      3701L
  EXIT2    066237'      3704      3711L
  EXIT3    066260'      3707      3723L
  EXIT4    066324'      3721      3744L
  EXIT5    066341'      3751L
  EXITA    066373'      3712      3758L
  EXITB    066354'      3744      3757L
  EXITC    067013'      3723      3760L
  FATSER1  066154'      3653L     3659
  FATSERR  066115'      1372      3415      3484      3651L     3732      3735      3737      3749      3756      4842
                        4849      5343      5345
  FCI      073256'      1376      4370      4404      4953L
  FCI1     073263'      4957L     4967
  FCI2     073307'      4959      4978L
  FCI3     073352'      4999      5001L
  FF       000014        217E     3272      4245      6083
  FFB      032133        367E     4779
  FFL      032205        369E
  FLT.BOP  000011        813L     1079      1148      1411
  FLT.CBD  000007        812L     1077      1139      1409
  FLT.CDB  000006        811L     1075      1136      1407
  FLT.CFC  000002        807L     1068      1402
  FLT.CRF  000003        808L     1070      1404
  FLT.CTY  000000        805L     1063      1141      1398
  FLT.CWI  000001        806L     1066      1400
  FLT.MNC  000004        809L     1072
  FLT.PBO  000013        816L     1083      2398
  FLT.SAL  000012        814L     1081      1413
  FSM      053270       1109      1560L
  FT.ABS   000000        396E
  FT.BAC   000003        399E
  FT.DD    000001        583E     4378      4412      4992
```

```
  FT.OC   000020        587E
  FT.OR   000002        584E     4374
  FT.OU   000010        586E
  FT.OW   000004        585E     4409
  FT.PIC  000001        397E     1809
  FT.REL  000002        398E
  FWAREL  064342       1220      1251      3333E
  FWASYS  064342'      1218      1228      1316      3335E     6089
  GSP     073362'      2246      5020L
  GSPA    073377'      1970      2232      2252      2267      5020      5029L     5801
  GUP     075035'      1380      5027      5413L
  GVM     053215       1108      1541L
  H17SDL  000173        174E
  HIGHDAT 077300'      1328      6032E     6037      6040      6043      6045      6047      6052      6055      6057
                       6059      6061      6064      6066      6069      6071
  HLCPDE  075053'      1635      3614      5442L
  HOSB2   051127       1118      1123L
  HOSBA   051133       1120      1126L
  HOSBOO1 051025       1058      1088E
  HOSBOOT 051006       1058L
  HOSTAB  051011       1062L     1063      1066      1068      1070      1072      1075      1077      1079      1081
                       1083      1136      1139      1141      1148      1398      1400      1402      1404      1407
                       1409      1411      1413      2398
  I.CONFL 000004        901E      902
  I.CONTY 000001        888E      889
  I.CONWI 000003        894E      895
  I.CSLMD 000000        877E
  I.CUSOR 000002        891E      892
  ILDEHL  075061'      5474L     5546
  IOC.CGN 000010        592L
  IOC.CSI 000011        593L
  IOC.DDA 000002        579L      588       602      4686      4985      4989      4995      5319
  IOC.DES 000016        599L
  IOC.DEV 000020        600L
  IOC.DIL 000021        602E     5000
  IOC.DIR 000023        604L     4686
  IOC.DRL 000010        596E     5322
  IOC.DTA 000014        598L
  IOC.FLG 000004        582L      596      4989      4995      5319
  IOC.GRT 000005        590L
  IOC.LGN 000012        594L
  IOC.LNK 000000        578L
  IOC.LSI 000013        595L
  IOC.SPG 000007        591L
  IOC.SQL 000003        588E     4998
  IOC.UNI 000022        601L     4981
  IOCCTD  000001        608E     4954      6009
  IOCELEN 000052        606E     6007      6010      6011      6013      6014      6016      6017      6019      6020
                       6022      6023      6025      6028
  IP.CON  000362        242E
  IP.PAD  000360        238E
  ISDEHL  075164'      5553      5609L
  ISY     075202'      1356      2260      5792L
  ISY1    075240'      5811      5813E
  ISY2    075261'      5822      5829L
  LAB.AUX 000117        729E      731
  LAB.AXL 000001        731E
```

```
  LAB.DAT 000000        706E
  LAB.DIS 000003        702L    1646    1714    2159
  LAB.GRT 000005        703L    1583    2226
  LAB.IND 000001        701L
  LAB.LAB 000021        725L     726
  LAB.LBL 000074        726E
  LAB.NOD 000002        708E
  LAB.PSS 000016        717L
  LAB.RGT 000012        713L
  LAB.SER 000000        700L    1574    1981    2547
  LAB.SIZ 000014        716L
  LAB.SPG 000007        704L    1586    1674    1798    2221
  LAB.SPT 000117        730L
  LAB.SYS 000001        707E
  LAB.VER 000011        711L
  LAB.VFL 000020        718L
  LAB.VLT 000010        710L
  LAB.VPL 000005        720E     722     723
  LAB.VPR 000014        715E     720
  LABEL   051217       1566    1574    1583    1586    1646    1674    1714    1798    1981    2159
                       2221    2226    2547    3290E   3291    3301
  LABELE  052217       1558    3291E   3293
  LDD     074001'      2242    3419    5062L
  LDD10   074226'      5216    5226L
  LDD2    074065'      5096L   5103
  LDD3    074104'      5111L   5128
  LDD4    032323       5117    5132E
  LDD8    074136'      5096    5111    5113    5160L
  LDD8A   074204'      2220    2250    5190L   5230
  LDD8B   074206'      2222    5073    5080    5191L   5228    5234
  LDD8C   074207'      5084    5178    5184    5192L
  LDD9    074211'      5082    5183    5213L
  LDD9A   074260'      5077    5226    5238L
  LDD9B   074261'      5076    5213    5215    5227    5239L
  LDE     057364       2159L
  LDE.    057361       2157L
  LDE..   057372       1647    2164L   2200
  LDE3    060013       2177L   2192
  LDE3.5  060033       2180    2188L
  LDO     033012        373E   3623
  LDON    065265'      3414    3483    3546L   4536
  LDON0   065343'      3560    3580    3585L   3638
  LDON1   065345'      3587L   3642    3647
  LDON2   065351'      3572    3593L
  LDON3   065354'      3583    3594L
  LDON4   066040'      3613    3615    3616    3622L
  LDON5   066101'      3551    3640L
  LDON6   066107'      3597    3618    3644L
  LENSYS  015164       3307    6089E
  LF      000012        204E   3181    3872    4279    4321
  LOADO   072030'      3515    4535L
  LOADO1  072076'      4540    4557L
  LOADO2  072100'      4537    4558L
  LSD     060063       1979    2215E
  LSO     053356       1110    1616L
  LSO.    054071       1617    1629    1641L
  LSO..   054306       1622    1631    1684L
```

```
  LSO1    054164       1648      1663L
  LSO2    054213       1668      1680L
  LSOA    054332       1616      1693L     1694
  LSOB    054347       1628      1695L     1696
  LWASYS  102126'      1218      1228      6088E     6089
  M.CDCA  000017        797L     6066
  M.CDLY  000016        796L     6064
  M.CFWA  000012        794L     6059
  M.CIN   000006        792L     6055
  M.CINT  000005        791L     6052
  M.CLWA  000014        795L     6061
  M.COUT  000010        793L     6057
  M.CPRE  000003        789L     6045
  M.CRUB  000004        790L     6047
  M.CSLC  000002        788L     6043
  M.FOX   000303        286E
  M.PAM8  000021        285E
  M.SALO  000001        787L     6040
  M.SUNI  000021        798L     6069
  M.SYDD  000022        799L     6071
  M.SYSM  000000        786L     6037
  MI.CPI  000376         32E
  MI.JMP  000303         33E     1347
  MI.RET  000311         34E
  MSD     056164       1113      1948L
  MSD1    056175       1955L     1977
  MSD2    056223       1962      1965      1975L
  MSD3    056232       1971      1979L
  MSD5    057034       1958      1980      1986      2001      2010L
  MSDA    057104       1990      2014L
  MSDB    052000       2219      2225      3301E
  MSYDD   077322'      1355      2256      5825      6072L
  NL      000012        215E      216      1543      1546      1638      1653      1681      1890      1909      2003
                       2004      2011      2020      2757      3077      3272      3652      3757      3823      4006
                       4009      4200      4823      4829      5300
  NUL2    000000        206E
  NULL    000200        205E
  OP.CTL  000360        239E
  OP.DIG  000360        240E
  OP.SEG  000361        241E
  OP2.CTL 000362        243E
  OTI     074262'      3488      3556      3601      3630      4542      5257L
  OVBUFE  064342       3302E     3307
  OVL.COD 000000        473L     1621      1630      3602      3609
  OVL.ENS 000010        478E     1621      1624      1630      1632      5260      5950      5952      5954
  OVL.ENT 000004        475L     3489      3631      4552
  OVL.FLB 000006        476L     3557      4543      4552
  OVL.IN  000001        945E     3573      5068
  OVL.NUM 000014        947E     3410      3413      3568      3578      3625
  OVL.RES 000002        946E     3407      3559      4545      4548
  OVL.SIZ 000002        474L     1624      1632      3609
  OVL.UCS 000200        948E     3399      3612      4539
  OVL0    000000        484L     1621      1624      3478      3730
  OVL1    000001        485L     1630      1632      3481      3733
  OVLCNT  000002       5946E     5956
  OVLMAX  000002       3550      5954E     5956
  OVLTAB  076036'      1330      1621      1624      1630      1632      5261      5948E     5954
```

```
  PATCH   064242       3271L     3274      3275
  PBO.DAT 000001        817E     1084      2399
  PCHL    032361       5133E
  PDI     033145        375E     4610      4660      4701
  PIC.COD 000006        744L      752
  PIC.ID  000000        739L
  PIC.LEN 000002        741L     1688
  PIC.PTR 000004        742L     1234      1866
  PRINT   071340'      3506      4434L     4439
  PSC     070167'      3889      4064L
  PSC1    070175'      4065      4069L
  PSC2    070226'      4071      4089L
  PSC3    070270'      4099      4110L
  QUOTE   000047        212E
  RDL     060231       2237      2291L
  RDL1    060312       2308      2322      2328L
  READ    071254'      3508      4370L
  REL     033177        379E
  REL.    033175        377E     1263
  ROMBOOT 030000        828E
  RRC     074311'      3935      3966      5290L
  RRC1    074333'      5296      5298L
  RRH     051217       1103      1167E     3290
  RRH1    051224       1170L     1176
  RRH1.5  051237       1171      1177L
  RRH2    051240       1168      1187E
  RRH2.5  051240       1189L
  RRH3    052003       1206      1216L
  RRH4    052042       1240L     1280
  RRH5    052076       1256      1270L
  RRH6    052066       1247      1260L
  RUBOUT  000177        208E
  RUC     033257        383E     3401
  S.BAUD  040344        922L     1138      1410      1486      2517
  S.BDA   041120       1020L
  S.BOOTF 041034        977L     1116      1147      1412
  S.CAADR 040333        905L     4084      4113      4921      4928      4930      5894      5920
  S.CACC  041006        961L     3338      3355      3371      3466
  S.CCTAB 040335        906L     4078
  S.CDB   040343        919L     1135      1408      1450      1517      3195      3854      4249
  S.CFWA  040352        929L     1333      4953
  S.CODE  041007        962L     3341      3353      3373
  S.CONFL 040332        903L     1326      3906      3910      4138      4505      5861      5863
  S.CONTY 040327        890L     1142      1340      1399      1463      1494      3819      3923      3938      4168
                       4183      4241
  S.CONWI 040331        896L     1401      4228      4231
  S.CSLMD 040326        878L      889       892       895       902      1323      3779      3916      4020      4342
                       4473
  S.CUSOR 040330        893L     1324      4190      4203      4228      4231
  S.DATC  040310        859L     2407      2427      2486      2515
  S.DATE  040277        858L     2419      2429      2451      2488
  S.DCS   041033        975L     1314
  S.DDDTA 040366        940L     2293
  S.DDGRP 040364        937L     2303      5078
  S.DDLDA 040360        935L     2326      3417      5089
  S.DDLEN 040362        936L     2313      5086
  S.DDOPC 040370        941L     2241      5347
```

```
  S.DFWA  040354        930L     1335      1874      1953
  S.DIREA 041016        969L     1373
  S.DLINK 040346        927L     1329
  S.FASER 041013        968L     1371      2534      2545      2552
  S.FCI   041021        970L     1375
  S.GRT0  024000        824E     3296
  S.GRT1  025000        825E
  S.GRT2  026000        826E
  S.GUP   041027        972L     1379      2128
  S.HIMEM 040316        861L     1189      1204      1216
  S.INT   040343        838L      915
  S.JUMPS 041010        966L     1350      1369      1371      1373      1375      1377      1379
  S.MOUNT 041032        974L     1590      3594      3702      3719      4089
  S.OFWA  040350        928L     1331
  S.OMAX  040324        867L     1623      1634
  S.OSN   041004        957L     3605
  S.OVLE  041000        954L     3633
  S.OVLFL 040371        950L     1338      3384      3398      3570      3611      3624      3627      4538      5066
  S.OVLS  040376        953L     3581      3622
  S.OVSTK 041035        982L     3473
  S.READ  031275        355E     1687      1726      1802      2168      2227      5181
  S.RFWA  040356        931L     1318      1832      1835      2257
  S.SCI   041024        971L     1377
  S.SCR   041121       1021L     1343
  S.SDD   041010        967L     1369
  S.SOVR  041146        840L      842
  S.SSN   041002        956L     1618
  S.SYSM  040320        863L     1317      1836      2078      2325      4554
  S.TIME  040312        860L     1099      1100
  S.UCSF  040372        951L
  S.UCSL  040374        952L
  S.USRM  040322        865L     1320      2314
  S.VAL   040277        837L      856
  S.WRITE 031330        357E
  SALONE  077301'      1414      3705      6041L
  SB.BAU  042205       1037L     2518
  SB.BOO  042200       1033L     1039      1040      1043      2537      3740      4845
  SB.BPE  042240       1041E     2556
  SB.DAT  042207       1038L     2426      2516
  SB.DRV  042240       1043L
  SB.FLG  042204       1035L
  SB.ORG  051000       1027E     1050      1234      1235
  SB.OVMX 014000       1028E     1619
  SB.SDB  044200       1045E
  SB.VER  042203       1034L     2506      2511      2516      2518      2521      2528      2556
  SBD     051154       1098      1135L
  SC.ACE  000350         43E     1210      1478      1483      1485      1488      1490      1493      1500      1503
                       1504      1506      1530      3210      3214      3656      3864      4264      4268      4861
  SC.UART 000372        113E     1211      1457      1458      1459      1460      1462      1471      1473      1524
                       3201      3205      3655      3859      4255      4259      4862
  SCD     053021       1107      1423E
  SCDLY   071204'      4272      4293L
  SCDLY1  071207'      4294L     4303
  SCI     074347'      1378      4383      4417      5317L
  SCIFWA  077312'      4723      5294      6060L
  SCIIN   077306'      3802      3977      3993      4500      5290      5304      6056L
  SCILWA  077314'      4720      5297      6062L
```

```
  SCIN    067014'      3502      3778E
  SCIN1   067027'      3783      3790L
  SCIN2   067041'      3791      3801L
  SCIN2.5 067104'      3815      3817      3823L
  SCIN3   067120'      3824      3826      3831L
  SCIN4.3 067314'      3925      3929L
  SCIN4.5 067321'      3928      3935L
  SCINI   067130'      1429      3842E
  SCINI0  067144'      3845      3854L
  SCINI01 067161'      3856      3864L
  SCINI02 067163'      3860      3866L
  SCINI10 070124'      4010      4016L
  SCINI11 070136'      3941      4030L
  SCINI2  067214'      3880      3887L
  SCINI3  067223'      3882      3893L
  SCINI35 067245'      3903L     4111
  SCINI4  067266'      3868      3895      3897      3899      3915L
  SCINI5  067355'      3944      3949L
  SCINI6  067361'      3931      3952L
  SCINI65 070003'      3954      3960L
  SCINI7  070011'      3966L     3967
  SCINI8  070034'      3919      3962      3976L
  SCINI9  070062'      3982      3993L
  SCINI93 070113'      4003      4005      4009L
  SCINI95 070120'      3998      4014L
  SCINIA  070155'      3904      4043L
  SCINTFL 077305'      6053L
  SCIOUT  077310'      3801      3809      3980      4501      5291      6058L
  SCIPRE  077303'      3893      4097      4101      6046L
  SCOUT   070304'      3504      4134E     4436
  SCOUT0  070307'      4138L     4141
  SCOUT1  070326'      3947      3950      3958      3989      4024      4094      4096      4106      4145      4161L
                       4320
  SCOUT10 071166'      4246      4277L
  SCOUT11 071170'      4278L     4283
  SCOUT2  070345'      3969      3971      4031      4033      4035      4166      4178L     4280      4299      4322
  SCOUT3  070363'      4188L     4193
  SCOUT4  071002'      4179      4187      4189      4198L
  SCOUT45 071015'      4199      4203L
  SCOUT5  071027'      4205      4207L
  SCOUT6  071042'      4208      4210      4212L
  SCOUT7  071060'      4213      4220L
  SCOUT8  071101'      4222      4240L
  SCOUT85 071120'      4244      4248L
  SCOUT9  071121'      4249L
  SCOUT91 071131'      4255L     4257
  SCOUT92 071146'      4251      4264L     4266
  SCOUT95 071160'      4260      4270L
  SCOUTA  071161'      4271E     6067
  SCU     053036       1431      1450L
  SCU1    053106       1452      1478L     1480
  SDD     074373'      1370      1925      5339L
  SDD1    075012'      5340      5347L
  SDT     054364       1111      1713E
  SDT1    054372       1719L     1733
  SDT2    055017       1731L     1762
  SDT3    055063       1744      1749      1759L
```

```
  SDT4    055064       1736      1760L
  SDTA    024000       1715      1719      1723      1730      1760      3296E     3297
  SDTAE   026000       1711      3297E
  SDTB    055075       1746      1764L     1765
  SDTBL   000013       1747      1765E
  SDV     052337       1102      1106      1398L
  SECSCR  100126'      1342      2166      2173      2196      5092      5165      5176      6085E
  SLR     052114       1105      1300E
  SLRA    052305       1309      1360L     1365
  SLRAL   000010       1309      1365E     1366
  SLRB    052315       1350      1368L     1369      1371      1373      1375      1377      1379      1381
  SLRBL   000022       1350      1381E
  SRR     052113       1104      1292E
  SSD     060377       1112      2398L
  SSD0    061015       2400      2405L
  SSD1    061062       2412      2414      2426L
  SSD1.5  061107       2432      2435L
  SSD2    061114       2422      2434      2440L     2460      2481
  SSD3    061153       2444      2450L
  SSD4    061163       2448      2456L
  SSD5    061217       2463      2474L
  SSD6    061225       2470      2479L
  SSD7    061307       2475      2485L
  SSD8    061310       2403      2486L
  SSDA    061322       2406      2436      2442      2467      2491L
  SSDB    052217       2409      2420      2458      3299E
  STACK   042200        844E     1089      3688      3713      3739
  STACKL  001032        842E
  STB     075016'      5229      5363L
  SUNIT   077321'      1151      4851      5807      6070L
  SYDD    040130        834E     1354      1357      1561      1569      1578      1985      2533      2540      2551
                       3699      4836      4841      4848
  SYS0    064373'      3348      3350      3353L
  SYS1    065015'      3360      3370L
  SYS2    065025'      3352      3374E
  SYS3    065033'      3387L
  SYS4    065055'      3397      3403L
  SYS5    065101'      3408      3417L
  SYSCAL  064342'      1301      3337E
  SYSCAL0 065127'      3387      3442E
  SYSCAL1 065156'      3450      3458L
  SYSCAL2 065176'      3448      3471L
  SYSCAL3 065220'      3477      3481L
  SYSCAL4 065222'      3479      3483L
  SYSCALA 065241'      3458      3498L
  SYSCALL 000377        409E     2707      2711      3078      3264
  SYSMODE 077300'      1362      3395      3420      3444      4915      6038L
  TAB     000011        213E     4178      4186      4212      4219
  TDD     060316       1201      2342L
  TDD.    060314       2341L
  TDD1    060317       2343L     2366
  TDD2    060336       2353L     2355
  TDDA    060365       2344      2370E
  TFE     033233        381E
  TTABT   075310'      5846      5849      5852      5857L
  TTDEV   076056'      5964L     5965      5967      5969      5971      5973      5975      5977      5979      5981
                       5983      5985
```

```
  TTDVD   075272'      5841E     5972
  TTDVD0  075275'      5844L     5855
  TTNOP   075331'      5850      5851      5853      5854      5873L
  TTOPE   075314'      5847      5848      5861L
  TTOUNT  076222'      5980      5991L
  TTR1    075346'      5894L     5898
  TTR2    075333'      5879L     5900
  TTR4    075374'      5906L     5912
  TTREAD  075336'      5844      5883E
  TTREOF  075367'      5896      5904L
  TTW2    076031'      5928      5930L
  TTWRITE 076012'      5845      5920L     5932
  TYPEC1  064155       3201L     3203
  TYPEC2  064172       3197      3210L     3212
  TYPEC3  064204       3206      3216L
  TYPEC4  064211       3222L     3227
  UBP     061323       1114      2506L
  UBP1    062055       2541      2543      2547L
  UBPA    062074       2512      2516      2518      2522      2527      2556L     2557
  UBPAL   000035       2510      2520      2526      2557E
  UC.2SB  000004         69E     1497      1498
  UC.5BW  000000         65E
  UC.6BW  000001         66E
  UC.7BW  000002         67E
  UC.8BW  000003         68E     1499
  UC.BI   000020         88E
  UC.CTS  000020         97E
  UC.DCS  000001         93E
  UC.DDR  000002         94E
  UC.DLA  000200         74E     1487
  UC.DR   000001         84E
  UC.DRL  000010         96E
  UC.DSR  000040         98E
  UC.DTR  000001         77E
  UC.EDA  000001         55E     1529
  UC.EPS  000020         71E
  UC.FE   000010         87E
  UC.IID  000006         62E
  UC.IIP  000001         61E
  UC.LOO  000020         81E     1484      1505
  UC.MSI  000010         58E
  UC.OR   000002         85E
  UC.OU1  000004         79E
  UC.OU2  000010         80E
  UC.PE   000004         86E
  UC.PEN  000010         70E
  UC.RI   000100         99E
  UC.RLS  000200        100E
  UC.RSI  000004         57E
  UC.RTS  000002         78E
  UC.SB   000100         73E
  UC.SKP  000040         72E
  UC.TER  000004         95E
  UC.THE  000040         89E     3211      4265
  UC.TRE  000002         56E
  UC.TSE  000100         90E     1479
  UCI.ER  000020        135E     1472      1523
```

```
  UCI.IE  000002        137E     1523
  UCI.IR  000100        133E     1461
  UCI.RE  000004        136E     1472      1523
  UCI.RO  000040        134E
  UCI.TE  000001        138E     1472      1523
  UDR     000000        110E     3205      3859      4259
  UF.FCT  000100        190E
  UF.RDA  000001        187E
  UF.ROR  000002        188E
  UF.RPE  000004        189E
  UF.TBM  000200        191E
  UMI.16X 000002        128E     1470
  UMI.1B  000100        118E     1465      1470
  UMI.1X  000001        127E
  UMI.2B  000300        120E     1465
  UMI.64X 000003        129E
  UMI.HB  000200        119E
  UMI.L5  000000        123E
  UMI.L6  000004        124E
  UMI.L7  000010        125E
  UMI.L8  000014        126E     1470
  UMI.PA  000020        122E
  UMI.PE  000040        121E
  UNT.DIS 000006        527L
  UNT.FLG 000000        523L     1853      5993
  UNT.GRT 000002        525L     2130      2248
  UNT.GTS 000004        526L
  UNT.SIZ 000010        529E     1828      1855      5421      5995
  UNT.SPG 000001        524L
  UO.CLK  000001        278E     1303      3691
  UO.DDU  000002        277E
  UO.HLT  000200        275E     1303      3691
  UO.NFR  000100        276E
  UP.DP   000174        181E
  UP.FC   000175        182E
  UP.SC   000176        184E
  UP.SR   000176        185E
  UP.ST   000175        183E
  UR.DLL  000000         50E     1490
  UR.DLM  000001         52E     1493
  UR.IER  000001         54E     1210      1483      1530      3656      4861
  UR.IIR  000002         60E
  UR.LCR  000003         64E     1488      1500
  UR.LSR  000005         83E     1478      3210      4264
  UR.MCR  000004         76E     1485      1504      1506
  UR.MSR  000006         92E
  UR.RBR  000000         46E     1503      3864
  UR.THR  000000         48E     3214      4268
  USERFWA 042200        845E     1319
  USR     000001        111E     1211      1457      1458      1459      1460      1462      1471      1473      1524
                       3201      3655      4255      4862
  USR.BD  000100        142E
  USR.FE  000040        143E
  USR.OE  000020        144E
  USR.PE  000010        145E
  USR.RXR 000002        147E
  USR.TXE 000004        146E
```

```
  USR.TXR 000001        148E     3202      4256
  VERS    000040        407E     2507      4576
  VERSN   072103'      3517      4576L
  VFL.NSD 000001        719E
  WRITE   071305'      3510      4404L
  XCHGBC  075171'      5543      5547      5555      5557      5717L


20850 Bytes Free
```