```
                                ;
                                ;*********************************************************
                                ;*                                                       *
                                ;*          UNIVERSAL BASIC I/O SYSTEM (BIOS)            *
                                ;*                     Vers. 3.0                         *
                                ;*                                                       *
                                ;*      A,B = 5 Inc. 32,10 sec/trk 256 byte/sec Hard    *
                                ;*      C,D = 5 Inc. 32,10 sec/trk 256 byte/sec (DS)    *
                                ;*      E   = 5 Inc. 10 sec/trk    256 byte/sec (SS)    *
                                ;*                                                       *
                                ;*********************************************************
                                ;
                                        title   Bios HDD3.0 for CP/M 2.2 with Hard-Disk Basf 6182.
                                        subttl  Copyright Costantino Haritakis Last rev 05/01/1986 12:30.
                                ;       Programmer: Costantino Haritakis.
                                ;
                         C              include SIZE.CPM ; Get cp/m size
  003B                   C              msize   equ     59 ; CP/M memory size in kilobyte
                                ;
  4844                          vers    equ     'HD'    ; Single side version
  001E                          rev     equ     30      ; CBIOS revision number
                                ;
                                ;
                                ;       Boolean scalar constants
  0000                          false   equ     0
  00FF                          true    equ     0ffh
                                ;
                                ;
                                ; ***   I/O Devices     ***
  0001                          TTY     equ     01b     ; CON:
  0000                          RDR     equ     false   ; Undefinited
  0000                          PUN     equ     false   ; Undefinited
  0002                          LST     equ     10b     ; LST:
                                ;
                                ;       Default Value for I/O byte
  0081                          DftI.O  equ     (LST shl 6) or (RDR shl 4) or (PUN shl 2) or (TTY)
                                        ;
                                        page    62
                                ;
                                ;
                                ;
                                ;*****************************************************
                                ;*                                                   *
                                ;*             ASCII EQUIVALENTS                     *
                                ;*                                                   *
                                ;*****************************************************
                                ;
 0007                           bell    equ     'G'-'@'         ; ring beeper
 0008                           backsp  equ     'H'-'@'         ; back space char.
 0009                           tab     equ     'I'-'@'         ; tabulation char.
 000A                           lf      equ     'J'-'@'         ; line-feed char.
 000C                           ffeed   equ     'L'-'@'         ; form feed char.
 000D                           cr      equ     'M'-'@'         ; carriage-return char.
 0013                           pfx     equ     'S'-'@'         ; attributes pfx
 0042                           rever   equ     'B'             ; Reverse On    (^SB)
 0043                           flash   equ     'C'             ; Flash On      (^SC)
 0040                           norm    equ     '@'             ; Normal        (^S@)
 0020                           space   equ     ' '             ; space char.
 0024                           endmsg  equ     '$'             ; end of print message
                                ;
                                ;
                                ;
                                ;*****************************************************
                                ;*                                                   *
                                ;*             Rom routines address                  *
                                ;*                                                   *
                                ;*****************************************************
                                ;
 F000                           rom     equ     0F000h          ; <--- rom starting address
 F003                           cin     equ     rom+3           ; console input
 F006                           cout    equ     rom+6           ; console output
 F009                           csts    equ     rom+9           ; console status
 F00C                           lout    equ     rom+12          ; printer output
 F00F                           lsts    equ     rom+15          ; printer status
 F012                           fdios   equ     rom+18          ; fdd I/O 128 byte
 F015                           fdiod   equ     rom+21          ; fdd I/O 256 byte
 F018                           wdini   equ     rom+24          ; wdd initialization
 F01B                           wdio    equ     rom+27          ; wdd I/O 256 byte
 F01E                           strout  equ     rom+30          ; print string .DE until $
 F01E                           print   equ     strout          ; sinonime
 F021                           bootrom equ     rom+33          ; load BIOS and go to wboote
 F024                           printat equ     rom+36          ; print str. -> DE at -> HL cursor
 F027                           movcurs equ     rom+39          ; move cursor at -> HL
 F02A                           vidinit equ     rom+42          ; initialize video
 F02D                           CompFlg equ     rom+45          ; Version Number
                                ;
                                        page
                                ;
                                ;*******************************************************
                                ;*              SYSTEM CONSTANTS                       *
                                ;*******************************************************
                                ;
 003B                           cmsize  equu    msize   ; cp/m size in kbyte
                                ;
                                ;       bias is address offset from 3400h for memory system
                                ;       than 16k (referred to as "b" throughout the next)
                                ;
 9C00                           bias    equ     (cmsize-20)*1024;
 D000                           ccp     equ     3400h+bias      ; base of ccp
 D806                           bdos    equ     ccp+806h        ; base of bdos
 E600                           bios    equ     ccp+1600h       ; base of bios
                                ;
                                ;
 1600                           cpml    equ     bios-ccp        ; lenght (in bytes) of cp/m system (ccp + bdos)
 0600                           biosl   equ     600h            ; lenght (in bytes) of standard bios
 0016                           cpmsiz  equ     cpml/secsiz     ; lenght (sector numbers) of cp/m (ccp + bdos)
 0006                           biossiz equ     biosl/secsiz    ; lenght (sector numbers) of bios
                                ;
                                ;
                                ;*******************************************************
                                ;*              Reserved Locations in Page Zero        *
                                ;*******************************************************
                                ;
 0000                           PZero   equ     0000            ; Start CP/M Zero Page
 0000                           JWBoot  equ     PZero           ; Jump to WBoot
 0003                           iobyte  equ     PZero+3         ; intel I/O byte
 0004                           CurDsk  equ     PZero+4         ; cp/m logical disk number
 0005                           JBdos   equ     PZero+5         ; Jump to BDOS
 0080                           defdma  equ     PZero+80h       ; cp/m defalt dma adrs
 0080                           stack   equ     0080h           ; wboot stack pointer
 1000                           stack1  equ     1000h           ; ipl stack pointer
                                ;
                                ;
                                ;*******************************************************
                                ;*                                                     *
                                ;*      BDOS constants on entry to write               *
                                ;*                                                     *
                                ;*******************************************************
                                ;
 0000                           wrall   equ     0               ; write to allocated
 0001                           wrdir   equ     1               ; write to directory
 0002                           wrual   equ     2               ; write to unallocated
                                ;
                                        page
                                ;
                                ;*****************************************************
                                ;*                                                   *
                                ;*      I P L  Double Side Floppy Version            *
                                ;*                                                   *
                                ;*****************************************************
                                ;
                                ;       this program loaded in ram by rom boot, load the cp/m
                                ;       bios, set bios sysflag and go to wboote
                                ;
                                        subttl  IPL for DS Floppy - OMICRON BIOS 3.0 with Hard-Disk BASF 6182
                                ;
 0000'                                  Aseg
                                        Org     100h
                                        .phase  1000h           ; origin of IPL
                                ;
                                ;
 1000                           wdbboot:
                                        ; entry point for bios boot from hard disk
 1000    C3 1009                        jp      wdbbt1          ; jump to hard bios boot
                                ;
 1003                           fdbboot:
                                        ; entry point for bios boot from floppy disk
 1003    C3 103A                        jp      fdbbt1          ; jump to floppy bios boot
 1006                           iplmsg:
                                        ; message for ipl checking
 1006    49 50 4C                       defb    'IPL'
                                ;
                                ;
 1009                           wdbbt1:
                                        ; load bios from hard disk
 1009    21 1060                        ld      hl,bbtdsk       ; H.L = bios boot r/w para pointer
 100C    CD F01B                        call    wdio            ; read bios
 100F    B7                             or      a               ; read error ?
 1010    20 1B                          jr      nz,bbterr       ; yes, then reinitialize system
                                                                ; A = 0 because not error occurs
 1012                           bbtok:
                                        ; bios has been loaded
 1012    32 E973                        ld      (sysflag),a     ; set bios system flag
 1015    DD 21 E974                     ld      ix,vidares      ; init video table
 1019    CD F02A                        call    vidinit
 101C    21 0003                        ld      hl,iobyte       ; point to iobyte
 101F    36 81                          ld      (hl),DftI.O     ; value for i/o byte (lst:=lpt:)
 1021    23                             inc     hl              ; point to logdsk
 1022    36 00                          ld      (hl),0          ; set cp/m logical disk = 0
 1024    11 106E                        ld      de,cpmmsg       ; D.E = cp/m message
 1027    CD F01E                        call    strout          ; print it
 102A    C3 E603                        jp      wboote          ; jump to bios wboote
                                        ;
                                        ; error in reading BIOS
                                        ;
 102D                           bbterr:
 102D    11 10A5                        ld      de,bbtermsg     ; D.E = bios boot error message
 1030    CD F01E                        call    strout          ; print it
 1033                           wait1cr:
 1033    CD F003                        call    cin             ; wait one char.
 1036   FE 0D                           cp      cr              ; cr ?
 1038   20 F9                           jr      nz,wait1cr      ;
                                        ;
 103A                           fdbbt1:
                                        ; load bios from floppy disk
 103A   21 1061                         ld      hl,bbttrk       ; H.L = track para pointer
 103D   36 01                           ld      (hl),1          ; floppy disk bios is in track 1 (DS)
 103F   06 06                           ld      b,biossiz       ; bios size in sector number
 1041   11 1067                         ld      de,bbtxlt-1     ; D.E = sector translate table for bios boot
 1044                           fdbbt2:
 1044   13                              inc     de              ; point to next sector
 1045   C5                              push    bc              ; save sector count
 1046   D5                              push    de              ; save xlt1 pointer
 1047   1A                              ld      a,(de)          ; load physical sector
 1048   32 1063                         ld      (bbtsec),a      ; set physical sector number
 104B   21 1060                         ld      hl,bbtdsk       ; H.L = boot para adrs
 104E   CD F015                         call    fdiod           ; read 256 byte
 1051   D1                              pop     de              ;
 1052   C1                              pop     bc              ;
 1053   B7                              or      a               ; read error ?
 1054   20 D7                           jr      nz,bbterr       ; yes, then reinitialize system
 1056   21 1065                         ld      hl,bbtdma+1     ; HL.= high byte current dma adrs
 1059   34                              inc     (hl)            ; set next dma adrs
                                                                ; bios boot end ?
 105A   10 E8                           djnz    fdbbt2          ; no, loop again
 105C   3C                              inc     a               ; A=1 for fdd boot
 105D   C3 1012                         jp      bbtok           ; then go to bios boot ok
                                ;
                                ;
                                        ; bios boot r/w para table (initially for wdd)
                                ;
 1060   00                      bbtdsk: defb    0               ; dsk-0 sid 0
 1061   0000                    bbttrk: defw    0               ; cylinder number
 1063   18                      bbtsec: defb    24              ; sector number (for wdd)
 1064   E600                    bbtdma: defw    bios            ; bios start address
 1066   00                      btprw:  defb    0               ; read operation
                                ;
 1067   06                      wdbloc: defb    biossiz         ; for wdd boot (6 sec. to load)
                                ;
 1068                           bbtxlt:
                                        ; sector translate table for floppy disk (256 byte/sec)
                                        ; the first two sector are occuped by ccp + bdos
                                        ; than bbtxlt starts at 4' sector
                                ;
 1068   09 05 02 08                     defb    9,5,2,8,4,10
 106C   04 0A
                                        ;
 106E                           cpmmsg:
 106E   0C 0D 0A 0A                     defb    ffeed,cr,lf,lf,pfx,'B ',(cmsize+1)/10+'0'
 1072   13 42 20 36
 1076   30 4B 20 4F                     defb    (cmsize+1) mod 10+'0','K OMICRON CP/M System - '
 107A   4D 49 43 52
 107E   4F 4E 20 43
 1082   50 2F 4D 20
 1086   53 79 73 74
 108A   65 6D 20 2D
 108E   20
 108F   76 65 72 73                     defb    'vers ',high vers,low vers
 1093    20 48 44
 1096    20 72 65 76                    defb    ' rev ',rev/10+'0','.',rev mod 10+'0',' ',pfx,'@'
 109A    20 33 2E 30
 109E    20 13 40
 10A1    0D 0A 07 24                    defb    cr,lf,bell,endmsg
                                ;
                                ;
 10A5                           bbtermsg:
 10A5    07 0D 0A 43                    defb    bell,cr,lf,'Cannot load your BIOS.'
 10A9    61 6E 6E 6F
 10AD    74 20 6C 6F
 10B1    61 64 20 79
 10B5    6F 75 72 20
 10B9    42 49 4F 53
 10BD    2E
 10BE    0D 0A 53 65                    defb    cr,lf,'Set new system diskette in disk A,'
 10C2    74 20 6E 65
 10C6    77 20 73 79
 10CA    73 74 65 6D
 10CE    20 64 69 73
 10D2    6B 65 74 74
 10D6    65 20 69 6E
 10DA    20 64 69 73
 10DE    6B 20 41 2C
 10E2    0D 0A 6F 6B                    defb    cr,lf,'ok push return. ',endmsg
 10E6    20 70 75 73
 10EA    68 20 72 65
 10EE    74 75 72 6E
 10F2    2E 20 24
                                ;
 10F5                           freipl  equ     $
                                        if      $ lt wdbboot+256
 000B                                   frebIPL equ     wdbboot+256-$   ; free space on IPL ram
 10F5                                   defs    frebIPL
                                        else
                                        if2
                                        .printx *** WARNING: IPL overflow reserved space ***
                                        endif
                                        endif
                                ;
                                        .dephase
                                ;
                                ;
                                ;
                                ;
                                ;*****************************************************
                                ;*                                                   *
                                ;*                     BIOS                          *
                                ;*                                                   *
                                ;*****************************************************
                                ;
                                subttl  Copyright Costantino Haritakis Last rev 05/01/1986 12:30
                                ;
                                ;       jump vector for individual subroutines
                                ;
                                        .phase  bios            ; origin of this program
                                ;
 E600    C3 E6E7                        jp      boot            ; cold start
 E603                           wboote:
 E603    C3 E6ED                        jp      wboot           ; warm start
 E606    C3 F009                        jp      csts            ; console status        (ROM)
 E609    C3 F003                        jp      cin             ; console character in  (ROM)
 E60C    C3 F006                        jp      cout            ; console charecter out (ROM)
 E60F    C3 E7A0                        jp      list            ; list character out
 E612    C3 E7BC                        jp      punch           ; punch character out
 E615    C3 E7BC                        jp      reader          ; reader character in
 E618    C3 E7EE                        jp      home            ; move head to home position
 E61B    C3 E7CD                        jp      seldsk          ; select disk
 E61E    C3 E7F1                        jp      settrk          ; set track number
 E621    C3 E802                        jp      setsec          ; set sector number
 E624    C3 E807                        jp      setdma          ; set dma address
 E627    C3 E80C                        jp      read            ; read disk
 E62A    C3 E811                        jp      write           ; write disk
 E62D    C3 E7AE                        jp      listst          ; return list status
 E630    C3 E7F6                        jp      sectran         ; sector translate
                                        ;
                                        page
                                ;
                                ;*******************************************************
                                ;* D P B T A B L E                                     *
                                ;*                                                     *
                                ;*      W/F  Size  B/S   S/T  Trk  Hds  R/T  Capacity   *
                                ;* -    --- ------ ---   --   ---  ---  --- ----------  *
                                ;* A: = wdd   5"   256   32   153   2    2  2432 Kbyte  *
                                ;* B: = wdd   5"   256   32   153   2    0  2448 Kbyte  *
                                ;* C: = fdd   5"   256   10    40   2    4   188 Kbyte  *
                                ;* D: = fdd   5"   256   10    40   2    4   188 Kbyte  *
                                ;* E: = fdd   5"   256   10    40   1    3    92 Kbyte  *
                                ;*                                                     *
                                ;*******************************************************
                                ;
                                ;
                                ;*******************************************************
                                ;*              Disk constants                         *
                                ;*******************************************************
                                ;
 0100                           secsiz  equ     256             ; byte/sector (256)
 0003                           fddsiz  equ     3               ; fdd number on system
 000A                           fddsec  equ     10              ; fdd sec/trk (10 sec/trk -256 byte-)
 0002                           wddsiz  equ     2               ; wdd number on system
 0020                           wddsec  equ     32              ; wdd sec/trk (32 sec/trk -256 byte-)
                                ;
 0005                           maxdsk  equ     fddsiz+wddsiz   ; max disk on system
                                ;
                                ;
 0002                           cpmblk  equ     secsiz/128      ; r/w buffer size
 0001                           secmsk  equ     cpmblk-1        ; sector mask
 0014                           fddspt  equ     fddsec*cpmblk   ; cp/m fdd sec/trk (20)
 0040                           wddspt  equ     wddsec*cpmblk   ; cp/m wdd sec/trk (64)
                                ;
                                ;
                                ;
                                ;*******************************************************
                                ;*              D P B   Table                          *
                                ;*******************************************************
                                ;
 E633                           dpbase  equ     $       ; base of disk parameter header
                                ;
                                        ; dpe0,dpe1 = disk parameter header for hard disk
 E633                           dpe0:
 E633    0000 0000                      defw    xlt0,0000h      ; no translate table
 E637    0000 0000                      defw    0000h,0000h     ; scratch area
 E63B    EA98 E6AB                      defw    dirbuf,dpb0     ; dir buff,parm block
 E63F    EB64 EB18                      defw    csv0,alv0       ; check,alloc vector
 E643                           dpe1:
 E643    0000 0000                      defw    xlt0,0000h      ; no translate table
 E647    0000 0000                      defw    0000h,0000h     ; scratch area
 E64B    EA98 E6BA                      defw    dirbuf,dpb01    ; dir buff,parm block
 E64F    EBB1 EB64                      defw    csv1,alv1       ; check,alloc vector
                                ;
                                        ; dpe2,dpe3 = disk parameter header for floppy disk (256 byte/sec)
 E653                           dpe2:   ; 256 byte/sec - Double Side
 E653    E683 0000                      defw    xlt1,0000h      ; translate table
 E657    0000 0000                     defw    0000h,0000h     ; scratch area
 E65B    EA98 E6C9                     defw    dirbuf,dpb1     ; dir buff,parm block
 E65F    EBBD EBB1                     defw    csv2,alv2       ; check,alloc vector
                                       ;
 E663                          dpe3:   ; 256 byte/sec - Double Side
 E663    E683 0000                     defw    xlt1,0000h      ; translate table
 E667    0000 0000                     defw    0000h,0000h     ; scratch area
 E66B    EA98 E6C9                     defw    dirbuf,dpb1     ; dir buff,parm block
 E66F    EBD9 EBCD                     defw    csv3,alv3       ; check,alloc vector
                                       ;
                                       ;
 E673                          dpe4:   ; 256 byte/sec - Single Side
 E673    E683 0000                     defw    xlt1,0000h      ; translate table
 E677    0000 0000                     defw    0000h,0000h     ; scratch area
 E67B    EA98 E6D8                     defw    dirbuf,dpb2     ; dir buff,parm block
 E67F    EBEF EBE9                     defw    csv4,alv4       ; check,alloc vector
                               ;
                                       page
```

```
                                ;
 0000                           xlt0    equ     0               ; no sector translate for hard disk
                                ;
 E683                           xlt1:
                                        ; sector translate table for floppy disk (256 byte/sec)
 E683    01 02 0D 0E                    defb    1,2,13,14,5,6,17,18,9,10,3,4,15,16,7,8,19,20,11,12
 E687    05 06 11 12
 E68B    09 0A 03 04
 E68F    0F 10 07 08
 E693    13 14 0B 0C
 E697    15 16 21 22                    defb    21,22,33,34,25,26,37,38,29,30,23,24,35,36,27,28,39,40,31,32
 E69B    19 1A 25 26
 E69F    1D 1E 17 18
 E6A3    23 24 1B 1C
 E6A7    27 28 1F 20
                                ;
                                ;
                                ;
 E6AB                           dpb0:
                                        ; disk parameter block for hard disk 0 (256 byte/sector 1 res. trk)
 E6AB    0080                           defw    128             ; SPT (sec/trk) (32 sect * (256/128) * 2 side)
 E6AD    05                             defb    5               ; BSH
 E6AE    1F                             defb    31              ; BLM
 E6AF    01                             defb    1               ; EXM (extent mask)
 E6B0    025F                           defw    607             ; DSM (disk size in BLS units - 1) (2432 kbyte)
 E6B2    03FF                           defw    1023            ; DRM (directory elements - 1)
 E6B4    FF                             defb    11111111b       ; AL0
 E6B5    00                             defb    00000000b       ; AL1
 E6B6    0000                           defw    0               ; CKS disk fixed, no dir. check vector
 E6B8    0001                           defw    1               ; OFF (track offset)
                                ;
 E6BA                           dpb01:
                                        ; disk parameter block for hard disk 1 (256 byte/sector no res. trk)
 E6BA    0080                           defw    128             ; SPT (sec/trk) (32 sect * (256/128) * 2 side)
 E6BC    05                             defb    5               ; BSH
 E6BD    1F                             defb    31              ; BLM
 E6BE    01                             defb    1               ; EXM (extent mask)
 E6BF    0263                           defw    611             ; DSM (disk size in BLS units - 1) (2448 kbyte)
 E6C1    03FF                           defw    1023            ; DRM (directory elements - 1)
 E6C3    FF                             defb    11111111b       ; AL0
 E6C4    00                             defb    00000000b       ; AL1
 E6C5    0000                           defw    0               ; CKS disk fixed, no dir. check vector
 E6C7    0000                           defw    0               ; OFF (no track offset)
                                ;
 E6C9                           dpb1:
                                        ; disk parameter block for floppy disk
                                        ; 256 byte/sector - Double Side
 E6C9    0028                           defw    40              ; SPT (sec/trk) (10 sect * (256/128) * 2 side)
 E6CB    04                             defb    4               ; BSH
 E6CC    0F                             defb    15              ; BLM
 E6CD    01                             defb    1               ; EXM (extent mask)
 E6CE    005E                           defw    94              ; DSM (disk size in BLS unit) (90 kbyte)
 E6D0    003F                           defw    63              ; DRM (directory elements - 1)
 E6D2    80                             defb    10000000b       ; AL0
 E6D3    00                             defb    00000000b       ; AL1
 E6D4    0010                           defw    16              ; CKS = (DRM + 1)/4  (size dir. check vect.)
```

```
  E6D6    0002                         defw    2               ; OFF (track offset)
                               ;
  E6D8                         dpb2:
                                       ; disk parameter block for floppy disk
                                       ; 256 byte/sector - Single Side
  E6D8    0014                         defw    20              ; SPT (sec/trk)
  E6DA    04                           defb    4               ; BSH
  E6DB    0F                           defb    15              ; BLM
  E6DC    01                           defb    1               ; EXM (extent mask)
  E6DD    002D                         defw    45              ; DSM (disk size in BLS unit) (90 kbyte)
  E6DF    003F                         defw    63              ; DRM (directory elements - 1)
  E6E1    80                           defb    10000000b       ; AL0
  E6E2    00                           defb    00000000b       ; AL1
  E6E3    0010                         defw    16              ; CKS = (DRM + 1)/4  (size dir. check vect.)
  E6E5    0003                         defw    3               ; OFF (track offset)
                               ;
                                       ;
                                       page
```

```
                                ;
                                ;****************************************************
                                ;* B O O T                                          *
                                ;*              Exec a Cold Boot                    *
                                ;****************************************************
                                ;
  E6E7                          boot:
                                        ; set A = sysflag and go to bootrom
  E6E7    3A E973                       ld      a,(sysflag)     ; if A = 0 then load IPL from WDD
  E6EA    C3 F021                       jp      bootrom         ; else from FDD
                                ;
                                ;
                                ;****************************************************
                                ;* W B O O T                                        *
                                ;*              Load bdos + ccp                     *
                                ;*              From wdd or Double/Side Fdd         *
                                ;****************************************************
                                ;
  E6ED                          wboot:
  E6ED    31 0080                       ld      sp,stack        ; set stack pointer
  E6F0    CD E89B                       call    WrtPng          ; Write any pending sector
  E6F3    21 0004                       ld      hl,CurDsk       ; point to cp/m log disk
  E6F6    7E                            ld      a,(hl)          ; load cp/m logical disk
  E6F7    E6 0F                         and     00001111b       ; mask out User
  E6F9    FE 05                         cp      maxdsk          ; disk overflow ?
  E6FB    38 01                         jr      c,wb_1          ; no, then go to wboot1
  E6FD                          wb_0:
  E6FD    74                            ld      (hl),h          ; else clear cp/m log disk
                                                                ; H=0
  E6FE                          wb_1:   ;
                                        ; Set parameter,
                                        ; then load from Hard or Floppy Disk
  E6FE    6C                            ld      l,h             ; H was 0 -> HL=0
  E6FF    22 E98E                       ld      (PrePhy),hl     ; Dsk 0 - side 0 & low Track=0
  E702    26 02                         ld      h,2             ; Sector #2
  E704    22 E990                       ld      (PreTrk+1),hl   ; Set High Trk=0 & Sector #2
  E707    26 16                         ld      h,cpmsiz        ; ccp + bdos size in sectors number
  E709    22 E994                       ld      (PreR.W),hl     ; set Read op. and # of sec (for wdd)
  E70C    21 D000                       ld      hl,ccp          ; Cp/m starting add
  E70F    22 E992                       ld      (PreDma),hl     ; set it
                                        ;
                                        ; Hard or Floppy ?
                                        ;
  E712    3A E973                       ld      a,(sysflag)     ; load system flag
  E715    B7                            or      a               ; sysflag = 0 ?
  E716    20 0B                         jr      nz,fd_wb        ; no, load from floppy
  E718                          wd_wb:
                                        ; load from hard disk
  E718    21 E98E                       ld      hl,PrePhy       ; H.L = wdd boot para adrs
  E71B    CD F01B                       call    wdio            ; call wdd read
  E71E    B7                            or      a               ; wdd i/o error ?
  E71F    20 74                         jr      nz,exboot       ; yes, then retry
  E721    18 35                         jr      syschk          ; no, then go to system check
                                ;
  E723                          fd_wb:
                                        ; load cp/m from floppy disk
```

```
 E723    06 16                    ld      b,cpmsiz        ; ccp + bdos size in sector number
 E725    11 E78C                  ld      de,wbxlt+1      ; D.E = sector translate table
 E728                     fd_wb.3:
 E728    C5                       push    bc              ; save sector count
 E729    D5                       push    de              ; save xlt1 pointer
 E72A    1A                       ld      a,(de)          ; load physical sector
 E72B    32 E991                  ld      (PreSec),a      ; set physical sector number
 E72E    21 E98E                  ld      hl,PrePhy       ; H.L = boot para adrs
 E731    CD F015                  call    fdiod           ; read 256 byte
 E734    D1                       pop     de              ;
 E735    C1                       pop     bc              ;
 E736    B7                       or      a               ; read error ?
 E737    20 5C                    jr      nz,exboot       ; yes, then retry
 E739    21 E993                  ld      hl,PreDma+1     ; HL = high current dma adrs
 E73C    34                       inc     (hl)            ; DMA=DMA+256
 E73D    05                       dec     b               ; warm boot end ?
 E73E    28 18                    jr      z,syschk        ; yes, then go to system check
 E740    13                       inc     de              ; xlt1 pointer + 1
 E741    7B                       ld      a,e             ;
 E742    FE 95                    cp      low (wbxlt+fddsec); end of track ?
 E744    20 E2                    jr      nz,fd_wb.3      ; no, load next sector
 E746    11 E78B                  ld      de,wbxlt        ; pointer to start xlt table
 E749    21 E98E                  ld      hl,PrePhy       ; point to side
 E74C    CB 66                    bit     4,(hl)          ; check for side 1
 E74E    CB E6                    set     4,(hl)          ; setting for side 1
 E750    28 D6                    jr      z,fd_wb.3       ; was side 0 then side 1
 E752    CB A6                    res     4,(hl)          ; set side 0
 E754    23                       inc     hl              ; point to track
 E755    34                       inc     (hl)            ; track = track + 1
 E756    18 D0                    jr      fd_wb.3         ; load first sector to next track
                                  ;
                                  ; CP/M has been loaded
 E758                     syschk:
                                  ; cp/m system check
 E758    3A D002                  ld      a,(ccp+2)       ; load third data of cp/m
 E75B    FE D3                    cp      high (ccp+35Ch) ; check for correct jp andress
 E75D    20 36                    jr      nz,exboot       ; no, error
 E75F    3E C3                    ld      a,0c3h          ; jump command
 E761    32 0000                  ld      (0000),a        ; location 0000h
 E764    21 E603                  ld      hl,wboote       ; wboot address
 E767    22 0001                  ld      (0001),hl       ;
 E76A    32 0005                  ld      (0005),a        ; location 0005h
 E76D    21 D806                  ld      hl,bdos         ; bdos address
 E770    22 0006                  ld      (0006),hl       ;
 E773    3E FF                    ld      a,0ffh          ; A = 0ffh
 E775    32 E98D                  ld      (PreDsk),a      ; Physic disk para -> 'ff'
 E778    21 E998                  ld      hl,defbuf       ; Default Buffer
 E77B    22 E992                  ld      (PreDma),hl     ; set it
 E77E    01 0080                  ld      bc,defdma       ; BC = defalt dma adrs
 E781    CD E807                  call    setdma          ; cp/m dma = 0080h
 E784    3A 0004                  ld      a,(CurDsk)      ; load cp/m Default disk
 E787    4F                       ld      c,a             ;
 E788    C3 D000                  jp      ccp             ; and jump to ccp
                          ;
 E78B                     wbxlt:
 E78B    01 07 03 09              defb    1,7,3,9,5,2,8,4,10,6 ; Skew factor table for fdd wboot
 E78F    05 02 08 04
 E793    0A 06
```

```
                                ;
                                ;
 E795                           exboot:
 E795                           exbot1:
 E795     11 E8EF                       ld      de,nosysmsg     ; D.E = no system message
 E798     CD E8E4                       call    msgcr           ; print it and wait cr
 E79B     3E 01                         ld      a,1             ; set A = 1 for IPL boot from floppy
 E79D     C3 F021                       jp      bootrom         ; jump to rom boot
                                ;
                                ;
                                        ;
                                        page
                                ;
                                ;****************************************************
                                ;*                                                  *
                                ;*      *** Logical Peripheral Device Sub ***       *
                                ;*                                                  *
                                ;****************************************************
                                ;
                                ; *** Console Subroutine Jump directing to rom ***
                                ;
                                ;
                                ;
                                ;****************************************************
                                ;* L i s t                                          *
                                ;*      Write C caracter on printer                 *
                                ;****************************************************
                                ;
  E7A0                          list:
  E7A0    3A 0003                       ld      a,(iobyte)      ; load intel i/o byte
  E7A3    E6 C0                         and     11000000b       ; mask bit 6,7
  E7A5    FE 80                         cp      080h            ;
  E7A7    DA F006                       jp      c,cout          ; jump rom console output
  E7AA    CA F00C                       jp      z,lout          ; jump printer output
  E7AD    C9                            ret                     ; no device, data lost
                                ;       jr      notdev          ; jump no device
                                ;
                                ;
                                ;****************************************************
                                ;* L i s t S t                                      *
                                ;*      Return printer status                       *
                                ;****************************************************
                                ;
                                ;
  E7AE                          listst:
  E7AE    3A 0003                       ld      a,(iobyte)      ; load intel i/o byte
  E7B1    E6 C0                         and     11000000b       ; mask bit 6,7
  E7B3    FE 80                         cp      080h            ;
  E7B5    DA F009                       jp      c,csts          ; jump rom console status
  E7B8    CA F00F                       jp      z,lsts          ; jump printer status
  E7BB    C9                            ret                     ; no device, now ret 11000000b
                                        ;jr     notdev          ; jump no device
                                ;
                                ;
                                ;****************************************************
                                ;* P u n c h                                        *
                                ;*      Puncher output                              *
                                ;****************************************************
                                ;
  E7BC                          punch:
                                        if      PUN             ; if PUNcher exists
                                        ld      a,(iobyte)      ; load intel i/o byte
                                        and     00110000b       ; mask bit 4,5
                                        cp      00010000b       ;
                                        jp      c,cout          ; = TTY: jump rom console output
                                                                ; = PTP:
                                        jp      nz,notdev       ; else no device exist
                                        jp      0000            ; spare jump
                                ; start of PTP: dev subroutine
                                ret                     ;
                                endif                   ; else no sub go to reader
                        ;
                        ;
                        ;*********************************************************
                        ;* R e a d e r                                           *
                        ;*      Reader input                                     *
                        ;*********************************************************
                        ;
 E7BC                   reader:
                                if      RDR             ; if ReaDeR exists
                                ld      a,(iobyte)      ; load intel i/o byte
                                and     00001100b       ; mask bit 2,3
                                cp      00000100b       ;
                                jp      c,cin           ; = TTY: jump rom console input
                                                        ; = PTR:
                                jr      nz,notdev       ; else no device exists
                                ; start of PTR: dev subroutine
                                jp      0000            ; spare jump
                                ELSE                    ; if no device
                                ;
                                ; if NO DEVICE
 E7BC    3E 1A                  ld      a,'Z'-'@'       ; set ^z = EOF
 E7BE    C9                     ret                     ; end
                                endif
                        ;
 E7BF                   notdev:
                                ; print not device message and go to cpm
 E7BF    3E 81                  ld      a,DftI.O        ; set default i/o byte
 E7C1    32 0003                ld      (iobyte),a      ;
 E7C4    11 E962                ld      de,ndevmsg      ; D.E = no device msg
 E7C7    CD F01E                call    strout          ; print it
 E7CA    C3 E6ED                jp      wboot           ; return to cp/m
                        ;

                        ;
                                page
```

```
                                  ;
                                  ;*****************************************************
                                  ;*               Disk I/O Subroutine                 *
                                  ;*****************************************************
                                  ;
                                  ;
                                  ;*****************************************************
                                  ;* S e l D s k                                       *
                                  ;*               Select logical disk from reg. C     *
                                  ;*               Ret HL=.DPB or 0 if error           *
                                  ;*****************************************************
                                  ;
 E7CD                             SelDsk:
 E7CD    21 0000                          ld      hl,0            ; return 0000h if error
 E7D0    79                               ld      a,c             ;
 E7D1    FE 05                            cp      maxdsk          ; too large ?
 E7D3    D0                               ret     nc              ; leave HL = 0000
                                  ;
 E7D4    FE 04                            cp      4               ; if Disk # > D:           nc = maggiore
 E7D6    30 09                            jr      nc,SDsk.1       ; then no swapping         C = minore
 E7D8    3A E973      Se disk#            ld      a,(sysflag)     ; load system flag
 E7DB    B7           e'                  or      a               ; if system flag = 0 then disk    set z flag se
                      maggiore                                    ; A,B = hard disk; C,D = floppy disk   sysflag = 1
 E7DC    79           di:                 ld      a,c             ;       restore disk # on a      ovvero se boot = fl
 E7DD    28 02        4 = D:              jr      z,SDsk.1        ; yes, if so
 E7DF    EE 02                            xor     00000010b       ; else A,B -> C,D and vice-versa      ∅ xoR 2 = 2
                                                                  ; A,B = floppy disk; C,D = hard disk  1 xoR 2 = 3
 E7E1                             SDsk.1:                                                               2 xoR 2 = ∅
 E7E1    32 E985                          ld      (LogDsk),a      ; set logical disk number             3 xoR 2 = 1
 E7E4    6F                               ld      l,a             ; L = disk number
                                          rept    4
                                          add     hl,hl           ; HL = disk number * 16
                                          endm
 E7E5    29                +              add     hl,hl           ; HL = disk number * 16
 E7E6    29                +              add     hl,hl           ; HL = disk number * 16
 E7E7    29                +              add     hl,hl           ; HL = disk number * 16
 E7E8    29                +              add     hl,hl           ; HL = disk number * 16
 E7E9    11 E633                          ld      de,dpbase       ;
 E7EC    19                               add     hl,de           ; H.L disk table adrs
 E7ED    C9                               ret
                                  ;
                                  ;
                                  ;*****************************************************
                                  ;* H O M E                                           *
                                  ;*               Select logical track 0              *
                                  ;*****************************************************
                                  ;
 E7EE                             Home:
 E7EE    01 0000                          ld      bc,0            ; Track #0000
                                  ;
                                  ;
                                  ;*****************************************************
                                  ;* S e t T r k                                       *
                                  ;*               Select logical track from reg.s BC  *
                                  ;*****************************************************
                                  ;
```

```
  E7F1                          SetTrk:
                                ;
  E7F1   ED 43 E987                     ld      (LogTrk),bc      ; Save low and high byte
  E7F5   C9                             ret                      ;
                                ;
                                ;
                                ;**********************************************************
                                ;* S e c T r a n                                         *
                                ;*              Translate the BC sector using trans       *
                                ;*              table pointed by DE                       *
                                ;**********************************************************
                                ;
  E7F6                          SecTran:
  E7F6   EB                             ex      de,hl            ; H.L = sectran table adrs
  E7F7   7D                             ld      a,l              ; check for -> 0000
  E7F8   B4                             or      h                ; this means no sec tran
  E7F9   09                             add     hl,bc            ; compute sector (BC = sec num)
  E7FA   28 04                          jr      z,Strn_5         ; no sec tran
  E7FC   6E                             ld      l,(hl)           ; get trans sector
  E7FD   26 00                          ld      h,0              ; high = 0
  E7FF   C9                             ret                      ; done
  E800   2C                     Strn_5: inc     l                ; convert to base 1
  E801   C9                             ret
                                ;
                                ;
                                ;**********************************************************
                                ;* S e t S e c                                           *
                                ;*              Set sector from registers BC              *
                                ;**********************************************************
                                ;
  E802                          SetSec:
  E802   79                             ld      a,c              ; Only low byte
  E803   32 E984                        ld      (LogSec),a       ; because sector < 256
  E806   C9                             ret                      ;
                                ;
                                ;
                                ;**********************************************************
                                ;* S e t D M A                                           *
                                ;*              Set DMA address from registers BC         *
                                ;**********************************************************
                                ;
  E807                          SetDma:
  E807   ED 43 E98A                     ld      (LogDMA),bc      ; set logical DMA
  E80B   C9                             ret
                                ;
                                ;
                                ;**********************************************************
                                ;* R e a d                                               *
                                ;*              Read sector specified by prev param       *
                                ;*              @ spec DMA (ret A=-1 if error)            *
                                ;**********************************************************
                                ;
  E80C                          read:
  E80C   AF                             xor     a                ; set disk read operation
  E80D   0E 02                          ld      c,wrual          ; write type (to unallocated)
  E80F   18 02                          jr      rw00             ;
                                ;
                                ;
                                ;****************************************************
                                ;* W r i t e                                        *
                                ;*              Write sector specified by prev param *
                                ;*              from spec DMA (ret A=-1 if error)    *
                                ;****************************************************
                                ;
 E811                           write:
 E811   3E 01                           ld      a,1             ; set write operation
                                ;
                                ;****************************************************
                                ;* R W 2 5 6   -      Read o Write 256 byte/sec dsk  *
                                ;****************************************************
                                ;
 E813   32 E98C                 rw00:   ld      (LogR.W),a      ; set read or write operation
 E816   79                              ld      a,c             ; get &
 E817   32 E997                         ld      (WrType),a      ; set CP/M write type
 E81A   11 E985                         ld      de,LogDsk       ; DE. LogDsk
 E81D   1A                              ld      a,(de)          ; get disk number
 E81E   26 40                           ld      h,wddspt        ; if disk number is 0 or 1
 E820   FE 02                           cp      wddsiz          ; then H = wdd sector/track
 E822   38 02                           jr      c,R256.1        ;
 E824   26 14                           ld      h,fddspt        ; else H = fdd sector/track
 E826   1B                      R256.1: dec     de              ; DE. LogSec
 E827   1A                              ld      a,(de)          ; Get Logical Sector
 E828   3D                              dec     a               ; to base 0
 E829   2E 00                           ld      l,0             ; initial side = 0
 E82B                           R256.2:
 E82B   BC                              cp      h               ; repeat until
 E82C   38 04                           jr      c,R256.3        ; log sec < sec/trk
 E82E   2C                              inc     l               ; side up
 E82F   94                              sub     h               ; log sec = log sec - sec/trk
 E830   18 F9                           jr      R256.2          ; retry
 E832                           R256.3:
 E832   B7                              or      a               ; carry = 0
 E833   1F                              rra                     ; A = A/2
 E834   3C                              inc     a               ; to base 1
 E835   32 E989                         ld      (PhySec),a      ; Set physical sector
                                        rept    4               ; move side number
                                        sla     l               ; to bit 4
                                        endm                    ; L = side number
 E838   CB 25           +               sla     l               ; to bit 4
 E83A   CB 25           +               sla     l               ; to bit 4
 E83C   CB 25           +               sla     l               ; to bit 4
 E83E   CB 25           +               sla     l               ; to bit 4
 E840   13                              inc     de              ; DE. LogDsk
 E841   1A                              ld      a,(de)          ; get LogDsk
 E842   E6 01                           and     1               ; only unit number
 E844   B5                              or      l               ; merge side
 E845   32 E986                         ld      (PhyDsk),a      ; set unit and side
                                        ;
 E848   06 05                           ld      b,5             ; byte count for old-new para compare
                                                                ; D.E => CP/M   Disk para (new)
 E84A   21 E98D                         ld      hl,PreDsk       ; H.L =>        Disk para (old)
 E84D                           rw01:
                                        ; compare old para with new para (dsk,sid,trk,sec)
 E84D   1A                              ld      a,(de)          ; A = new para
 E84E   BE                              cp      (hl)            ; (hl) = old para
 E84F   20 06                           jr      nz,wtchk        ; new <> old
 E851   23                              inc     hl              ; next para adrs
 E852   13                              inc     de              ;
 E853   10 F8                           djnz    rw01            ; repeat until end para
 E855   18 14                           jr      match           ; dsk,sid,trk,sec equ
 E857                           wtchk:
 E857   CD E89B                         call    WrtPng          ; Write Pending Sectors
 E85A   C0                              ret     nz              ; return if error
                                        ;
 E85B   01 0005                         ld      bc,5            ; 5 parameters
 E85E   21 E985                         ld      hl,LogDsk       ; H.L = new para adrs
 E861   11 E98D                         ld      de,PreDsk       ; D.E = old para adrs
 E864   ED B0                           ldir                    ; new para -> old para
 E866   CD E8A7                         call    diskrd          ; disk read
 E869   B7                              or      a               ; read error ?
 E86A   C0                              ret     nz              ; error return
 E86B                           match:
 E86B   3A E984                         ld      a,(LogSec)      ; load logical sector
 E86E   3D                              dec     a               ; convert to base 0
 E86F   E6 01                           and     secmsk          ; sector mask
 E871   67                              ld      h,a
 E872   2E 00                           ld      1,0             ; get high or low buff adrs
 E874   CB 3C                           srl     h               ; HL=HL*128=(*256)/2
 E876   CB 1D                           rr      1
 E878   11 E998                         ld      de,defbuf       ; D.E = phys sector buff start adrs
 E87B   19                              add     hl,de           ; H.L = log sector buff start adrs
 E87C   ED 5B E98A                      ld      de,(LogDma)     ; D.E = user dma adrs
 E880   01 0080                         ld      bc,128          ; BC = moving count
 E883   3A E98C                         ld      a,(LogR.W)      ; load r/w flag
 E886   B7                              or      a               ; read ?
 E887   28 04                           jr      z,rwbuf         ;
 E889   32 E996                         ld      (WrtFlg),a      ; write flag on (A=1)
 E88C   EB                              ex      de,hl           ; H.L = user dma adrs
 E88D                           rwbuf:
 E88D   ED B0                           ldir                    ; move (hl) to (de)
 E88F   3A E997                         ld      a,(WrType)      ; load write type
 E892   FE 01                           cp      wrdir           ; directory write ?
 E894   3E 00                           ld      a,0             ; prepare no errors
 E896   CC E89B                         call    z,WrtPng        ; yes, write Phys sector
 E899   B7                              or      a               ; set flags
 E89A   C9                              ret                     ; return status (A)

                                ;
                                ;
                                ;*****************************************************
                                ;* W r t P n g                                       *
                                ;*              Check for pending Sectors            *
                                ;*              Write if active                      *
                                ;*****************************************************
                                ;
 E89B                           WrtPng:
 E89B   21 E996                         ld      hl,WrtFlg
 E89E   7E                              ld      a,(hl)          ; get flag
 E89F   36 00                           ld      (hl),0          ; & clear
 E8A1   B7                              or      a               ; was active ?
 E8A2   C8                              ret     z               ; no, return
 E8A3   CD E8AA                         call    diskwt          ; yes, write flush data
 E8A6   C9                              ret                     ; return status & flag

                                ;
                                ;
                                ;*****************************************************
                                ;* D i s k R d                                       *
                                ;*              Read Physical Sector                 *
                                ;*****************************************************
                                ;
 E8A7                           diskrd:
                                        ; disk read
 E8A7   AF                              xor     a               ; 0 = read
 E8A8   18 02                           jr      rdwt
                                ;
                                ;
                                ;*****************************************************
                                ;* D i s k W t                                       *
                                ;*              Write Physical Sector                *
                                ;*****************************************************
                                ;
 E8AA                           diskwt:
                                        ; disk write
 E8AA   3E 01                           ld      a,1             ; 1 = write
 E8AC                           rdwt:
 E8AC   32 E994                         ld      (PreR.W),a      ; set r/w para
 E8AF                           rdwt0:
 E8AF   21 E98E                         ld      hl,PrePhy       ; H.L = i/o para adrs
 E8B2   3A E98D                         ld      a,(PreDsk)      ; load i/o unit number
 E8B5   FE 02                           cp      wddsiz          ; wdd i/o ?
 E8B7   30 12                           jr      nc,fdrdwt       ; no, then fdd i/o
 E8B9                           wdrdwt:
 E8B9   3E 01                           ld      a,1             ; one sector to wdd i/o
 E8BB   32 E995                         ld      (PreBlk),a      ; set wdd sector block
 E8BE   CD F01B                         call    wdio            ; exec. wdd i/o
 E8C1   B7                              or      a               ; i/o error ?
 E8C2   C8                              ret     z               ; no, then normal return
 E8C3                           NoBuff:
 E8C3                           rdwterr:
 E8C3   3E FF                           ld      a,0ffh          ; set no sector buffered
 E8C5   32 E98D                         ld      (PreDsk),a      ;
 E8C8   E6 01                           and     1               ; A=1
 E8CA   C9                              ret
                                ;
                                ;
 E8CB                           fdrdwt:
 E8CB   CD F015                         call    fdiod           ; r/w 256 byte
 E8CE   B7                              or      a               ; fdd i/o error ?
 E8CF   C8                              ret     z               ; no, then normal return
 E8D0   11 E925                         ld      de,ioerrmsg     ; D.E = Disk err message
 E8D3   CD F01E                         call    strout          ; print it
 E8D6   CD F003                         call    cin             ; wait one char.
 E8D9   FE 0D                           cp      cr              ; is return ?
 E8DB   28 D2                           jr      z,rdwt0         ; yes, then retry
 E8DD   FE 03                           cp      'C'-'@'         ; in cntrl C ?
 E8DF   20 E2                           jr      nz,NoBuff       ; Set Error and no sector buff
 E8E1   C3 E603                         jp      wboote          ; else go to wboot
                                ;
                                ;
                                ;
 E8E4                           msgcr:
                                        ; print string pointed by DE and wait cr
 E8E4   CD F01E                         call    strout          ; print it
 E8E7                         waitcr:
 E8E7    CD F003                      call    cin             ; wait one char.
 E8EA    FE 0D                        cp      cr              ; cr ?
 E8EC    20 F9                        jr      nz,waitcr       ;
 E8EE    C9                           ret
                                      ;
                                      page
```

```
                                ;
                                ;*********************************************************
                                ;*                                                       *
                                ;*              Initialized RAM data areas               *
                                ;*                                                       *
                                ;*********************************************************
                                ;
  E8EF                          nosysmsg:
  E8EF    0D 0A 07 53                   defb    cr,lf,bell,'Set system diskette in disk A,',cr,lf
  E8F3    65 74 20 73
  E8F7    79 73 74 65
  E8FB    6D 20 64 69
  E8FF    73 6B 65 74
  E903    74 65 20 69
  E907    6E 20 64 69
  E90B    73 6B 20 41
  E90F    2C 0D 0A
  E912    74 68 65 6E                   defb    'then push return. ',endmsg
  E916    20 70 75 73
  E91A    68 20 72 65
  E91E    74 75 72 6E
  E922    2E 20 24
                                ;
  E925                          ioerrmsg:
  E925    0D 0A 07 44                   defb    cr,lf,bell,'DISK ERROR',cr,lf
  E929    49 53 4B 20
  E92D    45 52 52 4F
  E931    52 0D 0A
  E934    3C 52 45 54                   defb    '<RETURN> retry, ^C abort, any key to continue'
  E938    55 52 4E 3E
  E93C    20 72 65 74
  E940    72 79 2C 20
  E944    5E 43 20 61
  E948    62 6F 72 74
  E94C    2C 20 61 6E
  E950    79 20 6B 65
  E954    79 20 74 6F
  E958    20 63 6F 6E
  E95C    74 69 6E 75
  E960    65
  E961    24                            defb    endmsg
                                ;
  E962                          ndevmsg:
  E962    0D 0A 07 2E                   defb    cr,lf,bell,'.NO Device.',cr,lf,endmsg
  E966    4E 4F 20 44
  E96A    65 76 69 63
  E96E    65 2E 0D 0A
  E972    24
                                ;
                                ;
  E973                          sysflag:
  E973    00                            defb    0               ; system flag for disk assignement
                                ;
  E974                          vidareas:
                                        ; video routine data areas
  E974                                  defs    16
```

```
                                ;
                                ; Logical Parameter Table
                                ;
 E984   01                      LogSec: defb    1               ; CP/M logical Sector number
 E985   00                      LogDsk: defb    0               ; CP/M logical Disk number
 E986   00                      PhyDsk: defb    0               ; Physical Disk Number
 E987   0000                    LogTrk: defw    0000            ; Physical Track Number
 E989   01                      PhySec: defb    1               ; Physical Sector Number
 E98A   0080                    LogDma: defw    0080h           ; CP/M logical Dma address
 E98C   00                      LogR.W: defb    0               ; CP/M logical R/W Flag
                                ;
                                ; Previous Parameter Table
                                ;
 E98D   FF                      PreDsk: defb    0ffh            ; Previous CP/M Disk
 E98E   00                      PrePhy: defb    0               ; Previous Phys Disk
 E98F   0000                    PreTrk: defw    0000            ; Previous Phys=logical Track
 E991   01                      PreSec: defb    1               ; Previous Phys Sector
 E992   E998                    PreDma: defw    defbuf          ; Physical DMA add
 E994   00                      PreR.W: defb    0               ; Phys R/W operation
 E995   01                      PreBlk: defb    1               ; Phys # of Sectors (for wdd)
 E996   00                      WrtFlg: defb    0               ; Write Pending Flag
 E997   01                      WrType: defb    1               ; BDos Write Type
                                ;
                                ;
                                ;
                                ;*******************************************************
                                ;*                                                     *
                                ;*              Disk data areas                        *
                                ;*                                                     *
                                ;*******************************************************
                                ;
 E998                           defbuf: defs    secsiz          ; defalt i/o dma address
 EA98                           dirbuf: defs    128             ; directory buffer
                                ;
                                ;
                                        ; wdd alloc and check vector
                                ;
 EB18                           alv0:   defs    76              ; alloc vector 0 (1215K/8)+1
 EB64                           csv0:   defs    0               ; no check vector 0
                                ;
 EB64                           alv1:   defs    77              ; alloc vector 1 (1223K/8)+1
 EBB1                           csv1:   defs    0               ; no check vector 1
                                ;
                                ;
                                        ; fdd alloc and check vector
                                ;
 EBB1                           alv2:   defs    12              ; alloc vector 2
 EBBD                           csv2:   defs    16              ; check vector 2
                                ;
 EBCD                           alv3:   defs    12              ; alloc vector 3
 EBD9                           csv3:   defs    16              ; check vector 3
                                ;
                                ;
 EBE9                           alv4:   defs    6               ; alloc vector 4
 EBEF                           csv4:   defs    16              ; check vector 4
                                ;
                                ;
                                ;
 EBFF                           freeram equ     $
                                        if      $ lt bios+600h
 0001                           freebyt equ     bios+600h-$     ; free space on bios ram
 EBFF                                   defs    freebyt
                                        else
                                        if2
                                        .printx *** WARNING: BIOS overflow reserved space ***
                                        endif
                                        endif
                                        page
```

```
                        .dephase                ; end of bios + data areas
                        end     100h
```

```
Macros:

Symbols:
ALV0    EB18    ALV1    EB64    ALV2    EBB1    ALV3    EBCD
ALV4    EBE9    BACKSP  0008    BBTDMA  1064    BBTDSK  1060
BBTERM  10A5    BBTERR  102D    BBTOK   1012    BBTSEC  1063
BBTTRK  1061    BBTXLT  1068    BDOS    D806    BELL    0007
BIAS    9C00    BIOS    E600    BIOSL   0600    BIOSSI  0006
BOOT    E6E7    BOOTRO  F021    BTPRW   1066    CCP     D000
CIN     F003    CMSIZE  003B    COMPFL  F02D    COUT    F006
CPMBLK  0002    CPML    1600    CPMMSG  106E    CPMSIZ  0016
CR      000D    CSTS    F009    CSV0    EB64    CSV1    EBB1
CSV2    EBBD    CSV3    EBD9    CSV4    EBEF    CURDSK  0004
DEFBUF  E998    DEFDMA  0080    DFTI.0  0081    DIRBUF  EA98
DISKRD  E8A7    DISKWT  E8AA    DPB0    E6AB    DPB01   E6BA
DPB1    E6C9    DPB2    E6D8    DPBASE  E633    DPE0    E633
DPE1    E643    DPE2    E653    DPE3    E663    DPE4    E673
ENDMSG  0024    EXBOOT  E795    EXBOT1  E795    FALSE   0000
FDBBOO  1003    FDBBT1  103A    FDBBT2  1044    FDDSEC  000A
FDDSIZ  0003    FDDSPT  0014    FDIOD   F015    FDIOS   F012
FDRDWT  E8CB    FD_WB   E723    FD_WB.  E728    FFEED   000C
FLASH   0043    FREBIP  000B    FREEBY  0001    FREERA  EBFF
FREIPL  10F5    HOME    E7EE    IOBYTE  0003    IOERRM  E925
IPLMSG  1006    JBDOS   0005    JWBOOT  0000    LF      000A
LIST    E7A0    LISTST  E7AE    LOGDMA  E98A    LOGDSK  E985
LOGR.W  E98C    LOGSEC  E984    LOGTRK  E987    LOUT    F00C
LST     0002    LSTS    F00F    MATCH   E86B    MAXDSK  0005
MOVCUR  F027    MSGCR   E8E4    MSIZE   003B    NDEVMS  E962
NOBUFF  E8C3    NORM    0040    NOSYSM  E8EF    NOTDEV  E7BF
PFX     0013    PHYDSK  E986    PHYSEC  E989    PREBLK  E995
PREDMA  E992    PREDSK  E98D    PREPHY  E98E    PRER.W  E994
PRESEC  E991    PRETRK  E98F    PRINT   F01E    PRINTA  F024
PUN     0000    PUNCH   E7BC    PZERO   0000    R256.1  E826
R256.2  E82B    R256.3  E832    RDR     0000    RDWT    E8AC
RDWT0   E8AF    RDWTER  E8C3    READ    E80C    READER  E7BC
REV     001E    REVER   0042    ROM     F000    RW00    E813
RW01    E84D    RWBUF   E88D    SDSK.1  E7E1    SECMSK  0001
SECSIZ  0100    SECTRA  E7F6    SELDSK  E7CD    SETDMA  E807
SETSEC  E802    SETTRK  E7F1    SPACE   0020    STACK   0080
STACK1  1000    STRN_5  E800    STROUT  F01E    SYSCHK  E758
SYSFLA  E973    TAB     0009    TRUE    00FF    TTY     0001
VERS    4844    VIDARE  E974    VIDINI  F02A    WAIT1C  1033
WAITCR  E8E7    WBOOT   E6ED    WBOOTE  E603    WBXLT   E78B
WB_0    E6FD    WB_1    E6FE    WDBBOO  1000    WDBBT1  1009
WDBLOC  1067    WDDSEC  0020    WDDSIZ  0002    WDDSPT  0040
WDINI   F018    WDIO    F01B    WDRDWT  E8B9    WD_WB   E718
WRALL   0000    WRDIR   0001    WRITE   E811    WRTFLG  E996
WRTPNG  E89B    WRTYPE  E997    WRUAL   0002    WTCHK   E857
XLT0    0000    XLT1    E683



No  Fatal error(s)
```