```
 ;Included from IO.asm
 .if SUPPORT_HARD_DISK

 ;*****************************************************************
 ;
 ; Hard Disk logic
 ;
 ;   IMPORTANT - ALL IO READS (but NOT writes) ARE INVERTED BY THE CPU BOAR
 ;
 ;   IMPORTANT - THIS LOGIC ASSUMES A HARD DRIVE EXISTS, DON'T CALL IF NOT
 ;
 ;   Boot Area Layout (sectors)
 ;   --------------------------
 ;
 ;      ____
 ;     |____|   <-- last sector number reported by hard disk pointed to by
 ;     |____|       and the end of the HDBOOTLEN boot area (end of boot cod
 ;     |____|
 ;      ...
 ;      ____
 ;     |____|
 ;     |____|   <-- HDBOOTLEN bytes below last sector (start of boot code)
 ;     |____|   <-- Boot Info Sector pointed to by 'HDBASect'
 ;     |____|   <-- last sector number reported to the OS as usable
 ;
 ;*****************************************************************

     HDBOOTLEN = (16 * 1024)             ;size of hard disk boot area to re
                                         ;  at the end of the logical drive


     .area   RAM     (REL)

     .globl HDStatus, HDBPS, HDNumSect, HDSPB
     .globl HDBANSect, HDBASect, HDGetStatus

 HDSBuffer:                              ;hard disk get stats command reply buf
     HDStatus:   .ds     1               ;  status byte
     HDBPS:      .ds     2               ;  bytes per sector
     HDNumSect:  .ds     2               ;  #of sectors on the drive
     HDSPB:      .ds     1               ;  sectors per block (flash writes occ

 ;Populated after reading the hard disk status record
 HDBANSect:      .ds     2               ;#of boot area sectors based on sector
 HDBASect:       .ds     2               ;sector number of first sector in boot
```

```
BootInfo:
BILoadAddress:  .ds     2           ;memory address boot image came from
BIFiller:       .ds     126         ;TODO RE-WRITE READ BOOT INFO LOGIC TO


    .area   CODE    (REL)               ;program area CODE is relative

    .globl HDReadData, HDWriteData
    .globl HDReadSector, HDWriteSector
    .globl BootDrive, SysDrive, HDGetInfo


HDGetStatus:
    ld      a,(HDBase)              ;
    ld      c,a                     ;  c = hard disk command port
    ld      a,#HDCMDGS              ;
    out     (c),a                   ;  issue get status command

    ld      hl,#HDSBuffer           ;
    ld      de,#HDCMDGSL            ;
    call    HDReadData              ;  read HDCMDGSL bytes into buffer

    ;  NOW FIGURE OUT WHERE THE BOOT SECTOR IS ON THIS DRIVE


    ld      de,#HDBOOTLEN           ;  get size of hard disk boot sector (
    ld      hl,(HDBPS)              ;  get #of bytes per sector - ASSUME T

;TODO HANDLE THIS BETTER - REPORT AN ERROR OR SOMETHING
    ld      a,h                     ;
    or      l                       ;
    jr      nz,XP5                  ;  if (reported bytes per sector == 0)
    ld      hl,#128                 ;    force BPS to a reasonable value s
XP5:                                ;  endif

                                    ;  Calculate hl = HDBOOTLEN / (HDBPS)
                                    ;  since (HDBPS) should always be a po
                                    ;  be accomplished more efficiently by
                                    ;  numerator and denominator until the
                                    ;  denominator.
                                    ;
SDDiv1:                             ;  while (hl bit 0 is not set)
    bit     0,l                     ;
    jr      nz,SDDiv2               ;
```

```
    xor     a                       ;    clear C flag
    rr      d                       ;
    rr      e                       ;    de = de / 2
    xor     a                       ;    clear C flag
    rr      h                       ;
    rr      l                       ;    hl = hl / 2
    jr      SDDiv1                  ;
SDDiv2:                             ;  end while

    inc     de                      ;  add one more sector to store the bo

    ld      (HDBANSect),de          ;  store #of 'HDBPS' byte sectors in h

    ld      hl,(HDNumSect)          ;  boot sectors are at the end of the
    xor     a                       ;  clear C flag
    sbc     hl,de                   ;  hl = first sector number of boot re

    ld      (HDBASect),hl           ;  store sector number of first sector

    ret


;HDWriteData
;
; Write a block of data to the hard disk data port
;
; hl = ptr to data string
; de = length of data string
;
HDWriteData:                        ;
    ld      a,(HDBase)              ;get hard drive
    add     a,#ExpDataOffset        ;  data offset register
    ld      c,a                     ;  into 'c' register
HDWDLoop:                           ;repeat
    ld      a,(hl)                  ;  get next data byte
    inc     hl                      ;  adjust pointer
    out     (c),a                   ;  write to data port
    dec     de                      ;
    ld      a,d                     ;  de--
    or      e                       ;
    jr      nz,HDWDLoop             ;until (data is written)
    ret


;HDReadData
```

```
;
; Read a block of data from the hard disk data port
;
; hl = ptr to buffer to receive the data
; de = length of data string
;
HDReadData:                         ;
    ld      a,(HDBase)              ;get hard drive
    add     a,#ExpDataOffset        ;  data offset register
    ld      c,a                     ;  into 'c' register
HDRDLoop:                           ;repeat
    in      a,(c)                   ;  read data port
    cpl                             ;  invert it - Digital Group I/O reads
    ld      (hl),a                  ;  store next data byte
    inc     hl                      ;  adjust pointer
    dec     de                      ;
    ld      a,d                     ;  de--
    or      e                       ;
    jr      nz,HDRDLoop             ;until (data is read)
    ret


;HDWriteSector
;
; Write a sector of data to the specified hard disk sector
;
; hl = ptr to data string
; de = sector number
HDWriteSector:                      ;write a sector of data to the hard di
    ld      a,(HDBase)              ;
    ld      c,a                     ;c = hard disk command port
    ld      a,#HDCMDWS              ;
    out     (c),a                   ;issue write sector command
    ld      a,e                     ;
    out     (c),a                   ;issue sector# LSB to command port
    ld      a,d                     ;
    out     (c),a                   ;issue sector# MSB to command port
    ld      de,(HDBPS)              ;get #of bytes in sector
    ld      a,e                     ;
    out     (c),a                   ;issue length LSB to command port
    ld      a,d                     ;
    out     (c),a                   ;issue length MSB to command port
    call    HDWriteData             ;write the sector data pointed to by h
    ret                             ;
```

```
;HDReadSector
;
; Read a sector of data from the specified hard disk sector
;
; hl = ptr to data string
; de = sector number
HDReadSector:                       ;read a sector of data from the hard d
    ld      a,(HDBase)              ;
    ld      c,a                     ;c = hard disk command port
    ld      a,#HDCMDRS              ;
    out     (c),a                   ;issue read sector command
    ld      a,e                     ;
    out     (c),a                   ;issue sector# LSB to command port
    ld      a,d                     ;
    out     (c),a                   ;issue sector# MSB to command port
    ld      de,(HDBPS)              ;get #of bytes in sector
    ld      a,e                     ;
    out     (c),a                   ;issue length LSB to command port
    ld      a,d                     ;
    out     (c),a                   ;issue length MSB to command port
    call    HDReadData              ;read the sector data into the buffer
    ret                             ;


;-----------------------------------------------------------------------
;SysDrive & BootDrive
;
;Copy 16K from the user specified memory location to the hard disk boot ar
;so that the Boot ROM can boot an OS at power up.  The companion logic (be
;BootDrive is used to restore the image to memory and execute it.
;
;SysDrive is invoked from the Boot ROM Monitor code
;BootDrive is invoked from the main power up logic
;
;input - ParseData = start of 16K block of memory
;
;-----------------------------------------------------------------------

SysDrive:
    ld      a,(HDBase)                  ;check to see if hard disk exi
    or      a                           ;
    jr      z,SysNHD                    ;if (hard drive exists)
    ld      de,(HDBASect)               ;  get 1st sector of boot area
    ld      hl,(ParseData)              ;  get user specified RAM addr
```

```
    ld      (BILoadAddress),hl          ;  store boot image RAM addres
    ld      hl,#BootInfo                ;  point to boot info area
    call    HDWriteSector               ;  write out the boot info

    ld      de,(HDBASect)               ;  get 1st sector of boot area
    inc     de                          ;  point past the boot info se
    ld      hl,(ParseData)              ;  get user specified RAM addr
    ld      bc,(HDBANSect)              ;  get size of HD boot area in
    dec     bc                          ;  remove the boot info sector
SDLoop:                                 ;  repeat
    push    bc                          ;
    push    de                          ;
    push    hl                          ;
    call    HDWriteSector               ;    write out a sector of the
    ld      a,#0x2e                     ;    get a period
    call    CmdWRA                      ;    display it for status
    pop     hl                          ;

    ld      de,(HDBPS)                  ;
    add     hl,de                       ;    hl += #of bytes per secto

    pop     de                          ;
    pop     bc                          ;

    inc     de                          ;    sector number++

    dec     bc                          ;    #of sectors--
    ld      a,b                         ;
    or      a,c                         ;
    jr      nz,SDLoop                   ;  until (all sectors have bee

SysNHD:                                 ;endif
    ret


BootDrive:
    ld      a,(HDBase)                  ;check to see if hard disk exi
    or      a                           ;
    jr      z,BDNHD                     ;if (hard drive exists)

    ; Read in the Boot Info sector

    ld      de,(HDBASect)               ;  get 1st sector of boot area
    ld      hl,#BootInfo                ;  point to boot info area
    call    HDReadSector                ;  write in the boot info
```

```
    ; Read in all of the code sectors

    ld      hl,(BILoadAddress)              ;  get destination location fo
    ld      de,(HDBASect)                   ;  get 1st sector of boot area
    inc     de                              ;  point past the boot info se
    ld      bc,(HDBANSect)                  ;  get size of HD boot area in
    dec     bc                              ;  remove the boot info sector
BDLoop:                                     ;  repeat
    push    bc                              ;
    push    de                              ;
    push    hl                              ;
    call    HDReadSector                    ;    read the next sector of t
    ld      a,#0x2e                         ;    get a period
    call    CmdWRA                          ;    display it for status
    pop     hl                              ;

    ld      de,(HDBPS)                      ;
    add     hl,de                           ;    hl += #of bytes per secto

    pop     de                              ;
    pop     bc                              ;

    inc     de                              ;    sector number++

    dec     bc                              ;    #of sectors--
    ld      a,b                             ;
    or      a,c                             ;
    jr      nz,BDLoop                       ;  until (all sectors have bee

    ld      hl,(BILoadAddress)              ;  get destination location fo
    jp     (hl)                             ;  *EXIT the boot rom logic an

BDNHD:                                      ;endif
    ret


;-------------------------------------------------------------------------
;HDGetInfo
;
;Get information about the drive
;
; input A = drive (0-A:, 1-B:, 2-C:, ...)
; output A = 0 on success, !0 = error code
;        BC = last usable sector number
```

```
;       DE = #of bytes per sector
;-------------------------------------------------------------------------

HDGetInfo:                                  ;get drive info
    cp      #2                              ;
    jr      nz,GINC                         ;if (requesting 'C' drive info
    ld      bc,(HDBASect)                   ;  get first sector of boot ar
    dec     bc                              ;  indicate sector just below
    ld      de,(HDBPS)                      ;  get #of bytes per sector
    xor     a                               ;  indicate success
    jr      GIExit                          ;
GINC:                                       ;else
    ld      a,#1                            ;  indicate drive not supporte
GIExit:                                     ;endif
    ret                                     ;



.endif
```