```
; Simple Z80 Boot Loader routine
; to be built with ZCC (version 3.09) tools
; available here: http://www.z80.info/z80sdt.htm
; Note: tools will only run in 16bit windows (ex: XP)
;
; build using the following commands:
;     set path=%path%;c:\z80\zcc\exe
;     asz80 -l bootrom.asm
;     aslink -i bootrom.rel
;     del bootrom.hex
;     ren a.ihx bootrom.hex

    .area   RAM     (REL)

    .globl ExpIDArray, KybdBase, HDBase


;***************************************************************************
;
;   IMPORTANT - ALL IO READS (but NOT writes) ARE INVERTED BY THE CPU BOAR
;
;***************************************************************************


; Expansion slot IDs discovered
ExpIDArray:  .ds    6

KybdBase:    .ds    1                   ;0x00 if no keyboard found, otherw
HDBase:      .ds    1                   ;0x00 if no hard disk found, other


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

    .globl DetectExpansion, ReadKybd

    .include ".\Hardware.asm"

;table of expansion port base addresses
ExpansionPorts:
    .db     Exp0Base, Exp1Base, Exp2Base, Exp3Base, Exp4Base, Exp5Base, 0x


;-------------------------------------
;GetPortBase
;
```

```
;Get port base address of feature
;
;Input: A = feature code to find
;            (see Hardware.asm)
;
;all registers destroyed
;port value returned in 'A' register
;  (0x00 if no feature available)
;-------------------------------------

GetPortBase:
    ld      c,a
    ld      de,#ExpIDArray
    ld      hl,#ExpansionPorts
    ld      b,#NumExpPorts
GPBLoop:                            ;Repeat
    ld      a,(de)                  ;  get next expansion port ID
    cp      a,c                     ;  check to see if this is the request
    jr      z,GPBFound              ;  *break out if a keyboard was found
    inc     hl                      ;
    inc     de                      ;  adjust the pointers
    djnz    GPBLoop                 ;Until (feature port found OR no more
    xor     a                       ;clear the return code indicating no f
    jr      GPBNotFound             ;
GPBFound:                           ;If (feature found)
    ld      a,(hl)                  ;  get the expansion port base address
GPBNotFound:                        ;endif
    ret

;-------------------------------------
;ReadKybd
;
;Get available keyboard key pressed
;
;all registers destroyed
;key value returned in 'A' register
;  (0x00 if no key available)
;-------------------------------------

ReadKybd:
    ld      a,(KybdBase)            ;
    or      a                       ;
    jr      z,RKNotFound            ;if (keyboard exists)
    add     a,#ExpDataOffset        ;  add in the offset to get to the dat
    ld      c,a                     ;
```

```
    in      a,(c)                   ;  read in the expansion port data reg
    cpl                             ;  invert it (all I/O reads are invert
RKNotFound:                         ;endif
    ret                             ;


;-------------------------------------
;DetectExpansion
;
;Detect expansion cards that are plugged in
;
;all registers destroyed
;-------------------------------------

DetectExpansion:
    xor     a                       ;
    ld      (KybdBase),a            ;default to no keyboard found
    ld      (HDBase),a              ;default to no hard disk found
    ld      de,#ExpIDArray
    ld      hl,#ExpansionPorts
    ld      b,#NumExpPorts
DEPLoop:                            ;Repeat
    ld      a,(hl)                  ;  get the next expansion port base ad
    add     a,#ExpIDOffset          ;  add in the offset to get to the ID
    ld      c,a                     ;
    in      a,(c)                   ;  read in the expansion port ID regis
    cpl                             ;  invert it (all I/O reads are invert
    ld      (de),a                  ;  store it in the ExpIDArray

    cp      a,#ExpIDKybd            ;  check if this is a keyboard
    jr      nz,DENKybd              ;  if (found a keyboard)
    ld      a,(hl)                  ;    get it's base register
    ld      (KybdBase),a            ;    store it for later
DENKybd:                            ;  endif

.if SUPPORT_HARD_DISK
    cp      a,#ExpIDHD              ;  check if this is a hard disk
    jr      nz,DENHD                ;  if (found a hard disk)
    ld      a,(hl)                  ;    get it's base register
    ld      (HDBase),a              ;    store it for later
DENHD:                              ;  endif
.endif

    inc     hl                      ;
    inc     de                      ;  adjust the pointers
```

```
    djnz    DEPLoop                 ;Until (all ports read in)

.if SUPPORT_HARD_DISK
    ld      a,(HDBase)              ;
    or      a                       ;
    jr      z,DENHD2                ;if (a hard disk was found)
    call    HDGetStatus             ;  read in the hard disk configuration
DENHD2:                             ;endif
.endif

    ret

    .include ".\HardDisk.asm"
```