Technical Note

Setting Interrupts in Z80 Mode on eZ80190, eZ80L92, eZ80F92, and eZ80F93 Devices

TN002302-1203

General Overview

This Technical Note describes how to set maskable interrupts for the eZ80190, eZ80L92, eZ80F92, and eZ80F93 devices in Z80 mode. The document discusses how to relocate the interrupt vector table and map interrupt service routines in the interrupt vector table.

A broader discussion about this topic covers the entire family of eZ80® devices in both ADL and Z80 modes. Please refer to the ZiLOG Application Note titled *Setting Interrupts with the eZ80® CPU* (AN0170).

Discussion

All maskable interrupts for the eZ80® family of devices use the eZ80® CPU's vectored interrupt function. The eZ80F91-based interrupt vector locations have a 24-bit address. The remainder of the eZ80® devices have a 16-bit address. In eZ80F91-based applications that run exclusively in Z80 mode, the interrupt vector address is {MBASE, I[7:1], IVECT[8:0]}. For other eZ80® CPU–based applications, the interrupt vector address is {MBASE, I[7:0], IVECT[7:0]}. A 16-bit word is fetched from the interrupt vector address and loaded into the lower two bytes of the Program Counter, PC [15:0].

In Z80 mode, the upper byte of the I Register bits, [15:8], is not used.

Because the ZDS II C compiler does not support Z80 mode, locating the interrupt vector table and writing the interrupt service routine must be performed in Assembly language.

Relocating the Interrupt Vector Table

In eZ80F91-based applications that run exclusively in Z80 mode, the interrupt vector address is {MBASE, I[7:1], IVECT[8:0]}. For other eZ80® CPU–based applications, which are discussed in this Technical Note, the interrupt vector address is {MBASE, I[7:0], IVECT[7:0]}. A 16-bit word is fetched from the interrupt vector address and loaded into the lower two bytes of the Program Counter, PC [15:0].

In Z80 mode, the upper byte of the I Register bits, [15:8], is not used.

Because the ZDS II C compiler does not support Z80 mode, locating the interrupt vector table and writing the interrupt service routine must be performed in Assembly language.

Relocating the Interrupt Vector Table

The start-up code presented below defines the interrupt vector table for the remainder of the eZ80® devices, in Z80 mode. Each entry is a 16-bit address pointing into the `__vector`

segment. This segment must be aligned to the 256 byte boundary of RAM and must reside in the lower 64KB of memory.

```
;*****************************************************************
;*****************************************************************
. assume ADL = 0;
NUM_VECTORS     EQU     64    ; Initialize all of the interrupt vector
                              ; location
. def  __vector table ;
define __vectab, space=RAM, align= 256
.sect  "__vectab"
ORG    %0300                  ; Base address of the Interrupt
                              ; vectors.
;*****************************************************************
;The interrupt vector table is mapped to ADDRESS  0300h; the ORG
;directive is used. However, this address should be the multiple of 256.
;*****************************************************************
;Set the interrupt to mode 2; load the interrupt base address to 8 bit I
;register. In this example, 03h is moved to I register. At the beginning
;of the program, the user should take care of initializing  MBASE
;register with an appropriate value.
;*****************************************************************

im 2                                  ; Interrupt mode 2
ld a, __vector_table >> 8 & 0ffh      ;
ld i,a                                ; Load interrtup vector base
;*****************************************************************
*;*****************************************************************
```

Note: The jump table concept for relocating the interrupt vector table cannot be applied in Z80 mode.

Mapping the ISR Location in the Interrupt Vector Table

The TIMER0 (PRT 0) interrupt vector location for the eZ80F92 device is at `0Ah`. Assuming that its interrupt service routine resides at the two-byte address location `1234h`, the TIMER0 interrupt service routine's address for the eZ80F92 device is stored as follows:

```
{MBASE, I Register [7:1], 0XXh}   ------------>   34h
{MBASE, I Register [7:1], 0XXh}   ------------>   12h
```

Writing the Interrupt Service Routine

The example code below illustrates how to write an interrupt service routine in Z80 mode for all of the eZ80® devices.

```
_ISR_Timer0:
  DI;
  EXX
```

```
  EX AF, AF'
------
------
------
------
EXX
EX AF, AF'
EI;
RETI
```

The following code illustrates how to load the `ISR_TIMER0` address at the eZ80F92 MCU's TIMER0 (PRT 0) interrupt vector location at `0Ah`. This code can be added to the user generated assembly program.

```
TIMER0 EQU %0A    // Vector offset for TIMER0(PRT 0)
                                          // for eZ80F92 is 0Ah
ld hl, VECTOR_TIMER0; Timer0
ld bc,__vector_table
add     hl,bc
ld      bc, _ISR_Timer0
ld      (hl), bc
```

This publication is subject to replacement by a later edition. To determine whether a later edition exists, or to request copies of publications, contact:

ZiLOG Worldwide Headquarters
532 Race Street
San Jose, CA 95126
Telephone: 408.558.8500
Fax: 408.558.8300
www.zilog.com

Information Integrity

The information contained within this document has been verified according to the general principles of electrical and mechanical engineering. Any applicable source code illustrated in the document was either written by an authorized ZiLOG employee or licensed consultant. Permission to use these codes in any form, besides the intended application, must be approved through a license agreement between both parties. ZiLOG will not be responsible for any code(s) used beyond the intended application. Contact the local ZiLOG Sales Office to obtain necessary license agreements.

Document Disclaimer