Technical Note

Setting Interrupts in Z80® Mode on the eZ80F91 MCU

TN002204-0607

General Overview

This Technical Note describes how to set maskable interrupts for ZiLOG's eZ80F91 MCU in Z80® mode. The document also describes 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 address and data long (ADL) mode and Z80 mode. For more information, refer to *Setting Interrupts with the eZ80 CPU Application Note (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 remaining 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.

Relocating the Interrupt Vector Table

The start-up code provided below locates the interrupt vector table for the eZ80F91 device in Z80 mode. Each entry is a 16-bit address pointing to the __vecptr segment. This segment must be aligned on the 512 byte page boundary of RAM and must reside in the lower 64 KB of memory.

```
;*****************************************************************
;*****************************************************************
. assume ADL = 0
NUM_VECTORS     EQU     64              ; Initialize all of the interrupt
                                        ; vector location
. def  __vector table                   ;
define __vectab, space=RAM, align= 512
.sect  "__vectab"
ORG %0400                               ;Base address of the
                                        ;Interrupt vectors
;
;*****************************************************************
; If the interrupt vector table is to be mapped at ADDRESS 2048, use the
```

```
; ORG directive as mentioned. However, this address should be a multiple
; of 512.
;*****************************************************************
; Set the interrupt to mode 2; load the interrupt base address to 8 bit I
; register. In this example, 04h 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        ;Difference to ADL mode is
                                        ;highlighted
ld i,a                                  ;Load interrupt vector base
;*****************************************************************
```

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

Mapping the Interrupt Service Routine Location in the Interrupt Vector Table

Consider TIMER0 as an example; TIMER0 interrupt service routine resides at the two-byte address location `1234h`. The TIMER0 interrupt vector location for the eZ80F91 device is at `054h`. The TIMER0 interrupt service routine's address is stored as follows:

```
{MBASE, I Register [7:1], 054h}   ------------>   34h
{MBASE, I Register [7:1], 055h}   ------------>   12h
{MBASE, I Register [7:1], 056h}   ------------>   Not used
{MBASE, I Register [7:1], 057h}   ------------>   Not used
```

The address locations {`MBASE, I Register [7:1], 056h`} and {`MBASE, I Register [7:1], 057h`} are not used.

Writing the Interrupt Service Routine

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

```
_ISR_Timer0:
  DI;
  EXX
  EX AF, AF'
------
------
------
------
EXX
EX AF, AF'
EI;
RETI
```

The following assembly code illustrates how to load the ISR_Timer0 address at the TIMER0 interrupt vector location for the eZ80F91 device in Z80 mode. This code can be added to the generated assembly program.

```
VECTOR_TIMER0 EQU %054                  // Vector offset for the
                                        // TIMER0 ISR for eZ80F91 is
                                        // 054h.


ld hl, VECTOR_TIMER0; Timer0
ld bc,__vector_table
add     hl,bc
ld      bc, _ISR_Timer0
ld      (hl), bc
```

Warning: DO NOT USE IN LIFE SUPPORT

LIFE SUPPORT POLICY

ZiLOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZiLOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer