

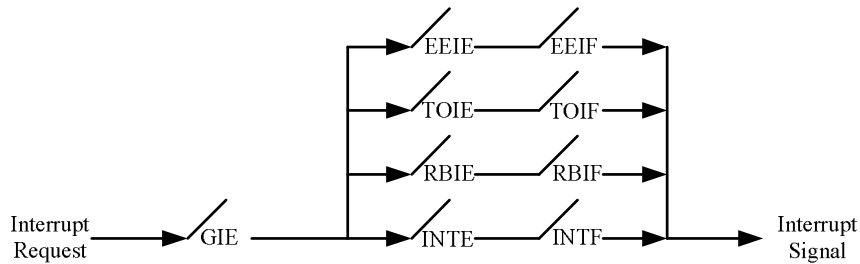
**ECE 425 Introductions to Microprocessors**  
**Laboratory Work 8**

**Objective:**

- 1) Understanding Interrupts.
- 2) MPLAB Stimulus Facility.
- 3) Interrupt Simulation Using Stimulus.

**Preparation:**

To generate an interrupt signal, some flags and some enable bits should have been set. This is illustrated in the figure below.



- 1) Set INT0 as interrupt source in stimulus. Using stimulus trace the following program. What happens to the value of program counter when interrupt occurs?

```
LIST P=16F84A
INCLUDE "P16f84A.INC"

__config __CP_OFF&_WDT_OFF&_XT_OSC

    org 0x00;
    goto start
    org 0x04;
    goto my_ISR

start
    bsf    STATUS, RP0;
    movlw b'00000001';
    movwf TRISB;
    bcf    OPTION_REG, INTEDG;
    bcf    STATUS, RP0;
    clrf   PORTB;

    bsf    INTCON, GIE;
    bsf    INTCON, INTE;
    bcf    INTCON, INTF;

loop   nop;
        nop;
        nop;
        nop;
    goto loop;

my_ISR
    bcf INTCON, INTF;
    nop;
    nop;
    nop;
    nop;
    retfie;

end
```

- 2) Using stimulus trace the following program. Give only one interrupt. Why program enters into an infinite loop, comment on it.

```
LIST P=16F84A
INCLUDE "P16f84A.INC"

__config __CP_OFF&_WDT_OFF&_XT_OSC

    org 0x00;
    goto start
    org 0x04;

start

    bsf    STATUS, RP0;
    movlw  b'00000001';
    movwf  TRISB;
    bcf    OPTION_REG, INTEDG;
    bcf    STATUS, RP0;
    clrf   PORTB;

    bsf    INTCON, GIE;
    bsf    INTCON, INTE;
    bcf    INTCON, INTF;

loop

    nop;
    nop;
    nop;
    nop;

goto loop;

my_ISR

    bcf INTCON, INTF;
    nop;
    nop;
    nop;
    nop;

    retfie;

end
```

- 3) Using stimulus trace the following program. Give only one interrupt. Why program enters into an infinite loop, comment on it. What is the difference from the previous question?

```
LIST P=16F84A
INCLUDE "P16f84A.INC"

__config __CP_OFF&_WDT_OFF&_XT_OSC

    org 0x00;
    goto start
    org 0x04;
    goto my_ISR

start
    bsf    STATUS, RP0;
    movlw b'00000001';
    movwf TRISB;
    bcf    OPTION_REG, INTEDG;
    bcf    STATUS, RP0;
    clrf   PORTB;

    bsf    INTCON, GIE;
    bsf    INTCON, INTE;
    bcf    INTCON, INTF;

loop
    nop;
    nop;
    nop;
    nop;

    goto loop;

my_ISR

    nop;
    nop;
    nop;
    nop;

    retfie;

end
```

- 4) PORTB pins 4, 5, 6, and 7 can be used as interrupt sources, i.e., RB4, RB5, RB6, RB7, can be used as interrupt sources. The following program uses RB4, RB5, RB6, and RB7 as interrupt sources. State change in one of RB4, RB5, RB6, and RB7 can trigger an interrupt. Trace the following program using stimulus. Send interrupt signals and note the values of RB4, RB5, RB6, and RB7.

```

LIST P=16F84A
INCLUDE "P16f84A.INC"

__config __CP_OFF&__WDT_OFF&__XT_OSC

    org 0x00;
    goto start
    org 0x04;
    goto ISR_RB_4_5_6_7;

start
    clrf    PORTB;
    bsf    STATUS, RP0;
    movlw  0xF0;
    movwf  TRISB;
    bcf    OPTION_REG, INTEDG;
    bcf    OPTION_REG, 7;
    bcf    STATUS, RP0;
    clrf    PORTB;

    bsf    INTCON, GIE;
    bsf    INTCON, RBIE;
    bcf    INTCON, RBIF;

loop
    nop;
    nop;
    nop;
    nop;

goto loop;

ISR_RB_4_5_6_7

    bsf PORTB, 0;
    bsf PORTB, 1;
    nop;
    nop;

    bcf INTCON, RBIF;
    retfie;

end

```

- 5) More than one interrupt sources may be enabled. Trace the following program using stimulus. Give two different interrupts at different times, and comment on the results.

```
LIST P=16F84A
INCLUDE "P16f84A.INC"

__config __CP_OFF&__WDT_OFF&__XT_OSC

    org 0x00;
    goto start
    org 0x04;
    btfsc INTCON, INTF;
    goto ISR_RB0_INT0;
    goto ISR_RB_4_5_6_7;

start    clrf    PORTB;
        bsf    STATUS, RP0;
        movlw 0xF1;
        movwf TRISB;
        bcf    OPTION_REG, INTEDG;
        bcf    OPTION_REG, 7;
        bcf    STATUS, RP0;
        clrf    PORTB;

        bsf    INTCON, GIE;
        bsf    INTCON, INTE;
        bsf    INTCON, RBIE;
        bcf    INTCON, INTF;
        bcf    INTCON, RBIF;
        bsf    INTCON, GIE;

loop     nop;
        nop;

        goto loop;

ISR_RB0_INT0

        bcf INTCON, INTF;
        nop;
        nop;
        nop;
        retfie;

ISR_RB_4_5_6_7

        btfsc PORTB, 7;
        nop;

        btfsc PORTB, 6;
        nop;

        btfsc PORTB, 5;
        nop;

        btfsc PORTB, 4;
        nop;

        bcf INTCON, RBIF;
        retfie;
end
```

**Laboratory Work:**

- 1) Trace program segments in preparation 1-5 and comment on the results.
- 2) Write a program that starts counting from zero, and each time an interrupt is received from INTO count value is incremented and displayed at PORTB

**During your LAB work show every step that you complete to the LAB assistant. Get a copy of assembly files you write during the LAB hour via a flash disk for future reference.**