ECE336 – MICROPROCESSORS I

WEEK 10

PIC16F84 INTERRUPTS

INTERRUPTS

- A single microcontroller can serve several devices. In the interrupt method, whenever any device needs the microcontroller’s service, the device notifies it by sending an interrupt signal.
- Upon receiving an interrupt signal, the microcontroller stops whatever it is doing and serves the device.
- The program associated with the interrupt is called the interrupt service routine (ISR).

Interrupt Service Routine

- For every interrupt, there must be a interrupt service routine (ISR), or interrupt handler. When an interrupt is invoked, the microcontroller runs the interrupt service routine.
- Generally, in most microprocessors, for every interrupt there is a fixed location in memory that holds the address of its ISR. In PIC16F84 there is only one location for the interrupt, location 004 (program memory address of the location).

Steps in executing an interrupt

Upon activation of an interrupt, the microcontroller goes through the following steps:

1. It finishes the instruction it is executing and saves the address of the next instruction (program counter) on the stack register.
2. It jumps to a fixed location in memory (address of the ISR)
3. The microcontroller gets the address of the ISR. It starts to execute the interrupt service subroutine until it reaches the last instruction of the subroutine, which is RETFIE (return from interrupt exit).
4. Upon executing the RETFIE instruction, the microcontroller returns to the place where it was interrupted. First, it gets the program counter (PC) address from the stack then it starts to execute from that address.

Sources of Interrupts in the PIC16F84

There are many sources of interrupts in PIC16F84, depending on which peripheral is incorporated into the chip. The following are some of the most widely used sources of interrupts in the PIC16F84.

1. External hardware interrupt. Pin RB0 (PORTB. 0) is for the external hardware interrupt INT.
2. There is an interrupt for timer overflow. This is the software interrupt or internal interrupt (when timer register TMR0 counts from h'FF' to h'00').
3. The PORTB-Change interrupt.(RB4,RB5,RB6, and RB7 can be used as interrupt sources. If any bit is changed, interrupt occurs.)
4. EEPROM-interrupt.(It occurs when the writing process is finished)

INTCON REGISTER

The interrupts must be enabled by software in order for the microcontroller to respond to them. The D7 of the INTCON (Interrupt Control) register is responsible for enabling and disabling the interrupts globally. The following figure shows the INTCON register with all interrupt control bits.

INTCON REGISTER

GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF

GIE (Global interrupt enable)

GIE=0, disables all interrupts.

GIE=1, interrupts are allowed to happen. Each interrupt source is anabled by setting the corresponding interrupt enable bit.

EEIE (EEPROM interrupt enable)

EEIE=0, disables EEPROM interrupt

EEIE=1, enables EEPROM interrupt

TOIE (TMR0 interrupt enable)

TOIE=0, disables TMR0 overflow interrupt

TOIE=1, enables TMR0 overflow interrupt

INTE (external interrupt enable)

INTE=0, disables external interrupt

INTE=1, enables external interrupt

RBIE (PORTB interrupt enable for RB4,RB5,RB6,RB7)

RBIE=0, disables PORTB interrupt

RBIE=1, enables PORTB interrupt

These bits, along with the GIE, must be set high for an interrupt to be responded to.Upon activition of the interrupt, the GIE bit is cleared by the PIC16 itself to make sure another interrupt cannot interrupt the microcontroller while it is servicing the current one. At the end of the ISR, the RETFIE instruction will make GIE=1 to allow another interrupt to come in.

FLAGS OF INTERRUPTS

If TOIF =1 there is TMR0 interrupt .

If INTF=1 there is external interrupt.

If RBIF=1 tehre is PORTB interrupt.

```
;==GENERAL STRUCTURE FOR AN ASSEMBLY PROGRAM WITH EXTERNAL
INTERRUPT SUBROUTINE

        LIST P=16F84A
        INCLUDE "P16F84A.INC"
        ORG     0X000   ; address of the main program
        GOTO    START
        ORG     0X004   ;address of the ISR
        GOTO    MY_ISR
START
        BSF      INTCON, GIE       ;global interrupt enable
        BSF     INTCON, INTE       ; external interrupt enable
................

LOOP
        GOTO     LOOP
MY_ISR
        BCF     INTCON, INTF       ; clear interrupt flag
......................
        RETFIE
        END
```

External interrupts and OPTION register

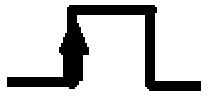
For external interrupts,

1. RB0 must be input.

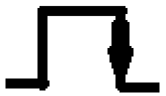
2. INTE must be 1.

3. Bit_6 of the OPTION register (INTEDG) is the interrupt edge select bit;

If INTEDG= 1, interrupt occurs rising edge of the signal.



If INTEDG= 0, interrupt occurs falling edge of the signal.



Depends on the hardware, INTEDG must be 0 or 1.

** To protect the contents of the W register and STATUS register, interrupt subroutine should be written as;

```
........
        ORG     h'004'
        GOTO MY_ISR
........
MY_ISR
        MOVWF           SAVE_W          ; SAVE_W=W_initial
        SWAPF           STATUS,W        ;W=SWAP STATUS_initial
        MOVWF           SAVE_S ; SAVE_S=SWAP STATUS_initial
        ..........
        SWAPF           SAVE_S, W  ;W=STATUS
        MOVWF           STATUS      ;STATUS=STATUS_initial
        SWAPF           SAVE_W, F  ;W=SWAP W_initial
        SWAPF           SAVE_W, W  ;W=W_initial
        RETFIE
```

Exp. Write a program that when RA1 is pressed, RB1 is on. If there is an interrupt from RB0/INT (for falling edge), RB2 is toggled.

Exp: Trace the following program and write the output.

```
                LIST P=16F84A
                INCLUDE "P16F84A.INC"
                ORG             0X000           ; address of the main program
                GOTO            START
                ORG             0X004           ;address of the ISR
                BTFSS           PORTA,2
                GOTO            MY_ISR1
                BTFSS           PORTA,4
                GOTO            MY_ISR2
                GOTO            MY_ISR3
START
                CLRF            PORTB
                BSF             STATUS,RP0
                CLRF            TRISB
                MOVLW           h’FF’
                MOVWF           TRISA
                BCF             OPTION_REG,INTEDG
                BCF             STATUS,RP0
                BCF             INTCON,INTF
                BSF              INTCON, GIE    ;global interrupt enable
                BSF             INTCON, INTE    ; external interrupt enable
LOOP
                GOTO             LOOP
MY_ISR1
                BCF             INTCON, INTF    ; clear interrupt flag
                BSF             PORTB,1
                RETFIE
MY_ISR2
                BCF             INTCON, INTF    ; clear interrupt flag
                BSF             PORTB,2
                RETFIE
MY_ISR3
                BCF             INTCON, INTF    ; clear interrupt flag
                BSF             PORTB,3
                RETFIE
                END
```

Exp: Write a program that checks all the interrupt sources and executes the related interrupt subroutines.

```
        LIST P=16F84A
        INCLUDE "P16F84A.INC"
        ORG     0X000    ; address of the main program
        GOTO    START
        ORG     0X004    ;address of the ISR
        BTFSC   INTCON,INTF
        GOTO    MY_ISR_RB0_INT
        BTFSC   INTCON,RBIF
        GOTO    MY_ISR_RB
        BTFSC   INTCON,TOIF
        GOTO    MY_ISR_TO
        GOTO    MY_ISR_EE
START
        BSF      INTCON, GIE       ;global interrupt enable
        BSF     INTCON, INTE       ; external interrupt enable
        BSF      INTCON, RBIE      ;PORTB_change interrupt enable
        BSF     INTCON, TOIE       ; Timer overflow interrupt enable
        BSF     INTCON, EEIE       ;eeprom interrupt enable
..............................
LOOP
        GOTO     LOOP
```

```
MY_ISR_RB0_INT
        BCF     INTCON, INTF    ; clear interrupt flag
        BSF     PORTB,1
        RETFIE
MY_ISR_RB
        BCF     INTCON, RBIF    ; clear interrupt flag
        BSF     PORTB,2
        RETFIE
MY_ISR_TO
        BCF     INTCON, TOIF    ; clear interrupt flag
        BSF     PORTB,3
        RETFIE
MY_ISR_EE
        BSF     PORTB,0
        RETFIE
        END
```