**EENG 3040 Microprocessors**

**In Class Exercise 6**

**Interrupts**

**To be done during the lecture period on September 14, 2015**

* Describe the purpose of each of the bits of the INTCON register in your own words

|  |  |
| --- | --- |
| Bit | Purpose |
| 0 | PORTB Change Interrupt Flag - there was a state change |
| 1 | External Interrupt Flag - external interrupt occurred |
| 2 | Timer0 Overflow Interrupt Flag - the timer overflowed |
| 3 | PORTB Change Interrupt Enable |
| 4 | INT External Interrupt Enable |
| 5 | Timer0 Overflow Interrupt Enable |
| 6 | Peripheral Interrupt Enable |
| 7 | Global Interrupt Enable - gotta turn this on to use anything else |

* What is the difference between an enable bit and a flag bit?

Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit. You should make sure the appropriate interrupt flag bits are clear before enabling an interrupt.

* Write an interrupt service routine (ISR) that does the following:
* Stores the contents of the W register into a variable named WCopy
* Stores the contents of the STATUS register into a variable named StatCopy
* Checks to see if the value on PORTB is equal to 7, if it is, increment a variable named COUNT
* Restore the value of the STATUS register
* Restore the value of the W register
* Return from interrupt

<https://gist.github.com/marioIncandeza/b9ee58616c906ad7a02f> ;for nicer formatting

|  |
| --- |
| ORG 000H |
|  |

|  |
| --- |
|  |
| GOTO INITIALIZE |
|  |
|  |
|  |
|  |
|  |
| ORG 004H |
|  |
|  |
| GOTO INCREMENTER |
|  |
|  |
|  |
|  |
|  |
| INITIALIZE |
|  |
|  |
| BCF STATUS,RP0 |
|  |
|  |
| BCF STATUS,RP1 |
|  |
| ;Since the upper 16 bytes of all GPR banks are common in the PIC16F882/883/884/886/887, temporary holding registers, W\_TEMP and STATUS\_TEMP, should be placed in here. These 16 locations do not require banking and therefore, make it easier to context save and restore. |
|  |
|  |
|  |
| WCOPY EQU 70H |
|  |
|  |
| STATCOPY EQU 71H |
|  |
|  |
| COUNT EQU 72H |
|  |
|  |
|  |
|  |
|  |
| BSF INTCON,GIE |
|  |
|  |
| BSF INTCON,3 |
|  |
|  |
|  |
|  |
|  |
| BSF STATUS,RP0 |
|  |
|  |
| CLRF TRISB |
|  |
|  |
| COMF TRISB,1 |
|  |
|  |
| CLRF IOCB |
|  |
|  |
| COMF IOCB,1 |
|  |
|  |
|  |
|  |
|  |
| BSF STATUS,RP1 |
|  |
|  |
| CLRF ANSEL |
|  |
|  |
| CLRF ANSELH |
|  |
|  |
|  |
|  |
|  |
| GOTO MAIN  MAIN  NOP  NOP  NOP  GOTO MAIN |
|  |
|  |
|  |
|  |
|  |
| INCREMENTER |
|  |
|  |
| MOVWF WCOPY |
|  |
|  |
| SWAPF STATUS,W |
|  |
|  |
| MOVWF STATCOPY |
|  |
|  |
|  |
|  |
|  |
| BCF STATUS,Z |
|  |
|  |
| MOVLW 007H |
|  |
|  |
| XORFW PORTB,0 |
|  |
|  |
| BTFSC STATUS,Z |
|  |
|  |
| INCF COUNT,1 |
|  |
|  |
|  |
|  |
|  |
| SWAPF STATCOPY,W ;I don't know how the swap is working here, but I trust the code in the datasheet |
|  |
|  |
| MOVWF STATUS |
|  |
|  |
| SWAPF WCOPY,F |
|  |
|  |
| SWAPF WCOPY,W |
|  |
|  |
| RETFIE |
|  |

|  |
| --- |
|  |

* What values would need to be in the INTCON and OPTION\_REG in order to have an interrupt configured for the rising edge of RB0? If a particular bit does not matter, indicate it as an X.

|  |  |
| --- | --- |
| INTCON | 1XXX 1XXX |
| OPTION\_REG | X1XX XXXX |