Chapter 13

PIC Family Microcontroller

Lesson 14

Interrupts

INTCON— Interrupt Control-Status bits Register

- Used for global, peripheral, interrupt enable bits for three interrupts (TMR0, external interrupt and RB port change and three flag bits for TMR0, INT external at RB0 pin and RB port change interrupt)

INTCON

- GIE bit b7: 1 enable global interrupts for interrupts which are unmasked, disable all interrupts when 0,
- PEIE bit b6: Peripheral interrupt enables all unmasked peripheral interrupts when 1, disable when 0
- T0IE bit b5: TMR0 overflow interrupt enable when 1, disable when 0
- INTIE bit b4: RB0/INT external interrupt enable when 1, disable when 0

INTCON

- RBIE bit b3: RB port change interrupt enable when 1, disable when 0
- T0IF bit b2: TMR0 overflow interrupt flag (1 on overflow and must be cleared by software)
- INTIF bit b1: RB0/INT external interrupt flag (1 on interrupt and must be cleared by software)
- RBIF bit b0: RB port change interrupt flag (1 on interrupt and must be cleared by software)

PIE1 (Peripheral Interrupt Enable) at 0x8C

- bit 0: TMR1 overflow enable
- bit 1: TMR2 overflow enable
- bit 2: CCP1 compare/capture /PWM enable
- bit 3: SSPIE synchronous serial port
- bit 4: USART transmit interrupt
- bit 5: USART receive interrupt
- bit 6: ADC interrupt enable
- bit 7: PSP Read/Write interrupt

PIE2 at 0x8D

- b2-b1 and b7 not implemented
- bit 0: CCP2 compare/capture /PWM enable
- bit 3: Bus collision interrupt
- bit 4: EEPROM write interrupt

PIR1 at 0x0C

- b0= 1 means TMR1 overflowed
- b1= 1 means TMR2 overflowed
- b2=1 means capture or compare-match occurred
- b3= 1 means SPI/I2C transmission or reception taken place
- b4=1 USART Tx buffer empty
- b5=1 USART receive buffer full
- b6=1 AD conversion complete
- b7= 1 PSP read or write taken place

PIR2

- b2-b1 and b7 not implemented
- CCP2IF TMR1 capture or match occurred
- Bus collision occurred at SSP in I^2C mode
- EEPROM write completed

ISR Start

- GIE bit resets
- Disables the further interrupt by any source during execution of the ISR
- The 13-bit return address is pushed onto stack
- The PC bits loaded from interrupt vector address 0x0004
- The ISR finds the flags, which are set and executes the service routine for one of interrupt source
- The flag resets by the software
- This prevents repeated service for the same interrupt event and also prepares the source for next interrupt

ISR Return

- Return from interrupt instruction (RETFIE) executes in an ISR when there is return from the ISR
- The GIE bit also sets on return
- This re-enables interrupt

Context Save

- *context save* means the process of executing instructions that enables the retrieval of the registers and other variables on return from the ISR

Saving a Context

- No PUSH or POP instruction in the instruction set
- PC value the only one saved onto the stack by the hardware
- If interrupt service is not masked fro an interrupt source, the ISR executes on interrupt
- W-register and STATUS and some other registers of presently running program often saved by software instructions in the ISR
- W-register or STATUS might be required during running of the ISR

Summary

We learnt

- Register used for control of interrupts (Enabling and disabling)
- INTCON
- PIE1
- PIE2

We learnt

- Registers used for status of interrupts (flags)
- PR1
- PR2

We learnt

- Saving of context done by instructions

End of Lesson 14 on

Interrupts