Interrupts

ELEC 330

Digital Systems Engineering

Dr. Ron Hayne

Images Courtesy of Ramesh Gaonkar and Delmar Learning

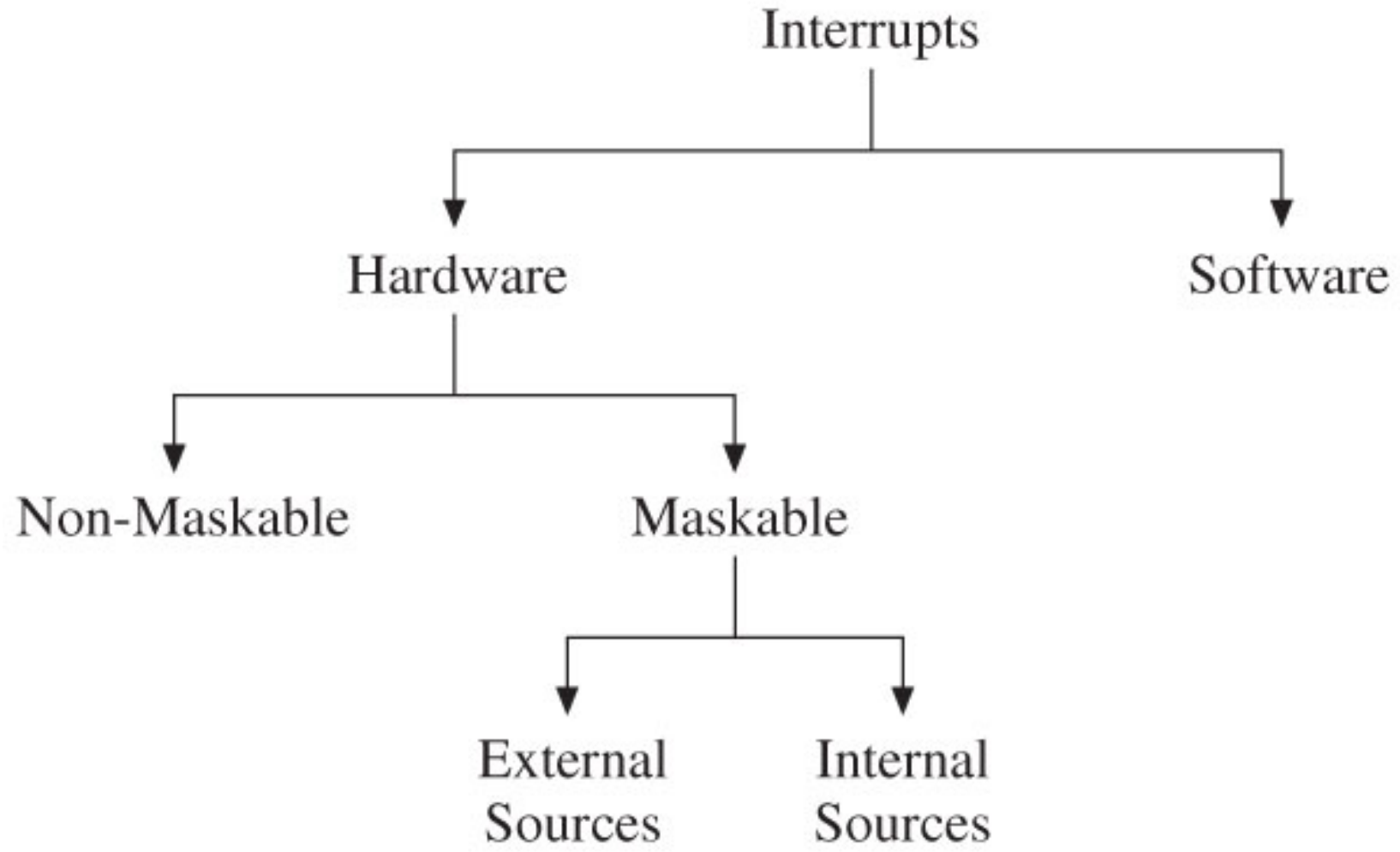
THE CITADEL

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

Basic Concepts of Interrupts

- An interrupt is a communication process
  - A device
    - Requests the MPU to stop processing
    - Internal or external
  - The MPU
    - Acknowledges the request
    - Attends to the request
    - Goes back to processing where it was interrupted

Types of Interrupts



MPU Response to Interrupts

- When interrupts are enabled
  - MPU checks interrupt request flag at the end of each instruction
- If interrupt request is present, the MPU
  - Resets the interrupt flag
  - Saves the return address on the stack
- MPU redirected to appropriate memory location
  - Interrupt vectors
- Interrupt service routine (ISR) meets request
- MPU returns to where it was interrupted
  - Specific return instruction

PIC18 Interrupts

- PIC18 Microcontroller family
  - Has multiple sources that can send interrupt requests
  - Does not have any non-maskable or software interrupts
    - All interrupts are maskable hardware
  - Has a priority scheme divided into two groups
    - High priority and low priority
  - Uses many Special Function Registers (SFRs) to implement the interrupt process

PIC18 Interrupt Sources

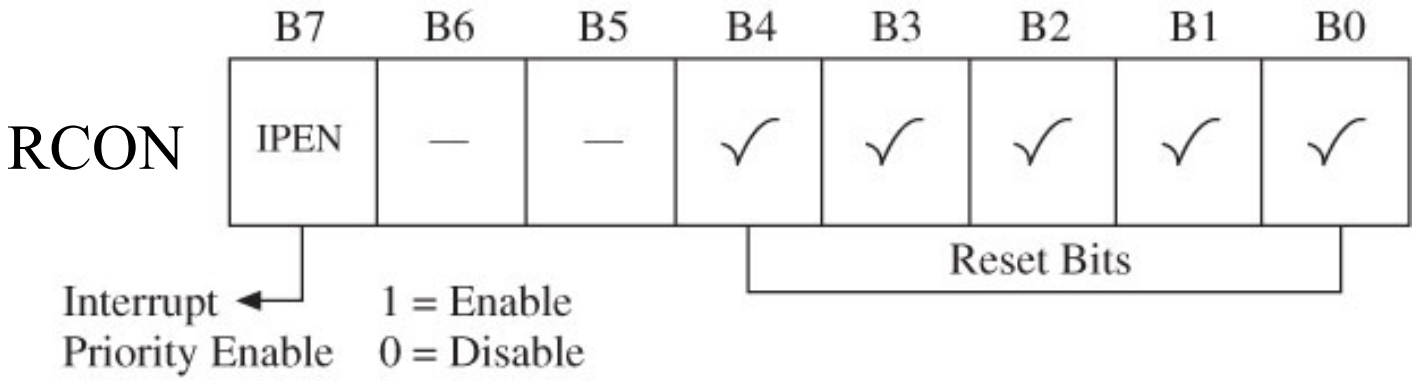
- External sources
  - Three pins of PORTB
    - RB0/INTO, RB1/INT1,and RB2/INT2
    - Can be used to connect external interrupting sources
      - Keypads or switches
  - PORTB Interrupt (RBI)
    - Change in logic levels of pins RB4-RB7
- Internal peripheral sources
  - Examples
    - Timers
    - A/D Converter
    - Serial I/O

PIC18 Interrupt Sources

- Special Function Registers (SFRs)
  - RCON
    - Priority Enable
  - INTCON
    - External interrupt sources
  - IPR, PIE, and PIR
    - Internal peripheral interrupts
- Valid interrupt
  - Interrupt request bit (flag)
  - Interrupt enable bit
  - Priority bit

Interrupt Priority

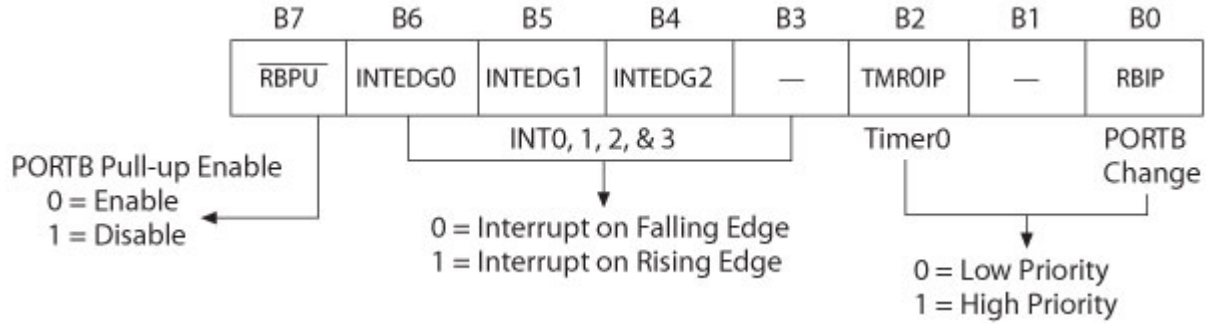
- Interrupt priorities
  - High-priority interrupt vector 000008_H
  - Low-priority interrupt vector 000018_H
  - A high-priority interrupt can interrupt a low-priority interrupt in progress.
  - Interrupt priority enable
    - Bit7 (IPEN) in RCON register



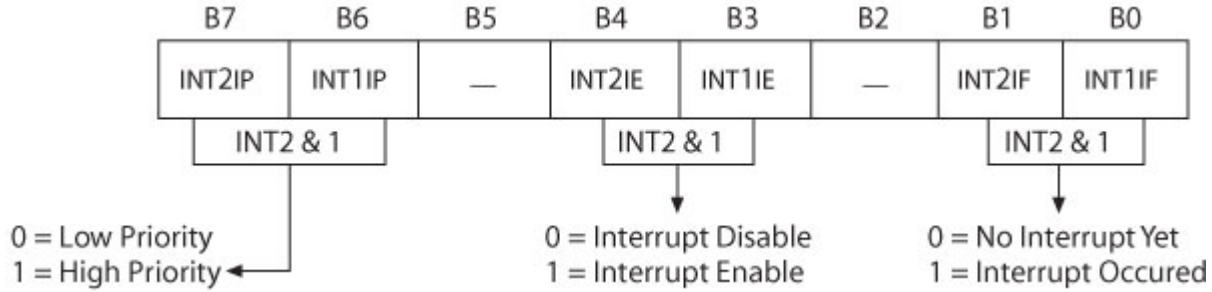
External Interrupts

- INTCON Register

- INTCON2 Register



- INTCON3 Register



Interrupt Service Routine (ISR)

- Similar to a subroutine
- Attends to the request of an interrupting source
  - Clears the interrupt flag
  - Should save register contents that may be affected by the code in the ISR
  - Must be terminated with the instruction RETFIE
- When an interrupt occurs, the MPU:
  - Completes the instruction being executed
  - Disables global interrupt enable
  - Places the return address on the stack

Interrupt Service Routine (ISR)

- High-priority interrupts
  - The contents of W, STATUS, and BSR registers are automatically saved into respective shadow registers.
- Low-priority interrupts
  - These registers must be saved as a part of the ISR
    - If they are affected

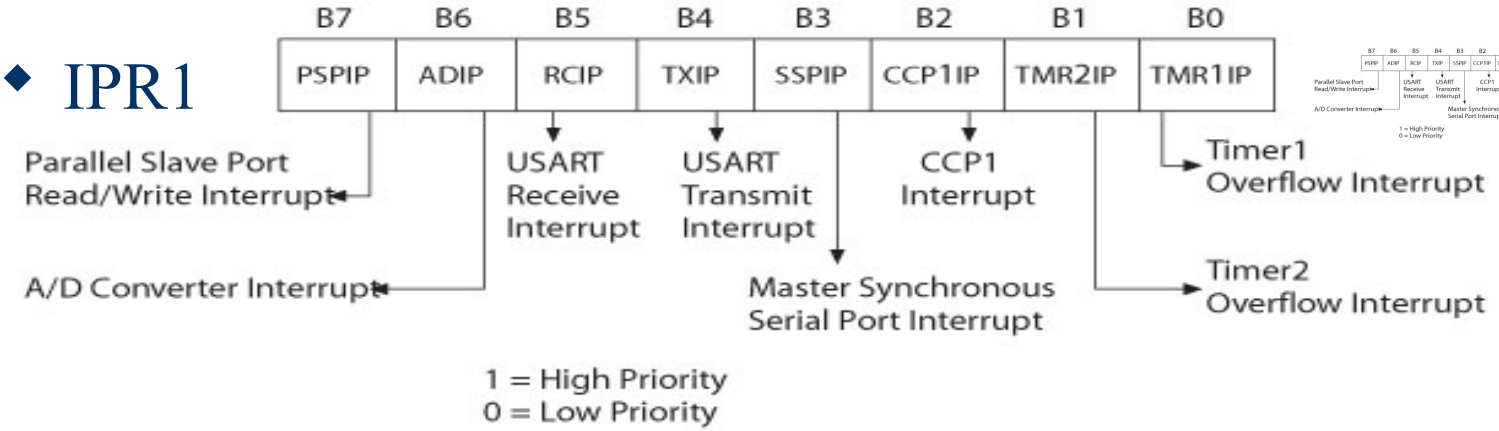
- RETFIE [s] ;Return from interrupt
- RETFIE FAST ;FAST equivalent to s = 1
  - If s =1: MPU also retrieves the contents of W, BSR, and STATUS registers

Internal Interrupts

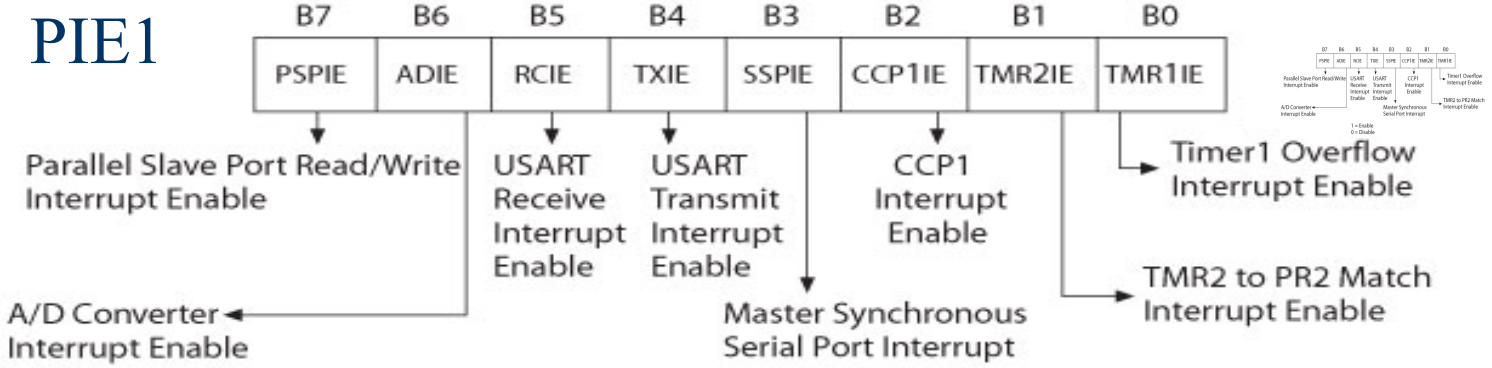
- PIC18 MCU internal interrupt sources
  - Timers
  - A/D converter
  - Serial I/O
- Each interrupt has three bits
  - Interrupt priority bit
  - Interrupt enable bit
  - Interrupt request bit (flag)
- Interrupt registers
  - IPR: Interrupt Priority Register
  - PIE: Peripheral Interrupt Enable
  - PIR: Peripheral Interrupt Request (Flags)

Interrupt Registers

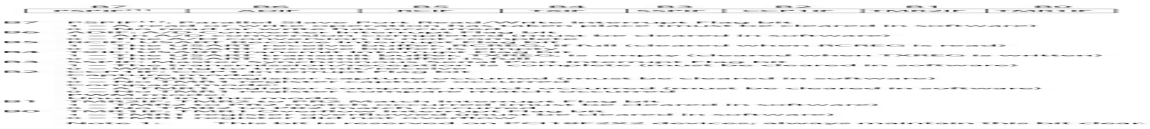
- IPR1



- PIE1



- PIR1



Multiple Interrupt Sources

- All interrupt requests are directed to one of two memory locations (interrupt vectors)
  - 000008_H (high-priority)
  - 000018_H (low-priority)
- When there are multiple requests
  - The interrupt source must be identified by checking the interrupt flags

Illustration

- Problem Statement
  - INT1 set up as a high-priority interrupt
  - Timer1 and Timer2 set up as low-priority
  - Identify the interrupt sources
    - Execute the appropriate interrupt service routines

Illustration

Label	Opcode	Operand	Comments
S_TMP	EQU	0x100	;Temp Registers
W_TMP	EQU	0x101	
	ORG	0x00	
	GOTO	MAIN	
	ORG	0x08	;High-Priority Interrupt Vector
INTCHK:	GOTO	INT1_ISR	
	ORG	0x18	;Low-Priority Interrupt Vector
TIMERCHK:	BTFSC	PIR1,TMR1IF	;Timer1 Flag, Skip if Clear
	GOTO	TMR1_ISR	
	BTFSC	PIR1,TMR2IF	;Timer2 Flag, Skip if Clear
	GOTO	TMR2_ISR	
	RETFIE		

Illustration

Label	Opcode	Operand	Comments
MAIN:			;Main Program goes here
			;Do Something
HERE:	GOTO	HERE	;Wait for an Interrupt
	ORG	0x100	
INT1_ISR:	BCF	INTCON3,INT1IF	;Clear Flag
			;Do Something
	RETFIE	FAST	;Retrieve registers and Return

Illustration

Label	Opcode	Operand	Comments
TMR1_ISR:	MOVFF	STATUS, S_TMP	
	MOVWF	W_TMP	;Save Registers
	BCF	PIR1,TMR1IF	;Clear Flag
			;Do Something
	MOVF	W_TMP,W	;Retrieve Registers
	MOVFF	S_TMP,STATUS	
	RETFIE		;Return from interrupt
TMR2_ISR			;Similar to Timer1