SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

19ECB211 – MICROCONTROLLER PROGRAMMING & INTERFACING

II YEAR IV SEM

UNIT II – PIC TIMER, SERIAL PORT AND INTERRUPT

TOPIC 6 – PIC Interrupts

Interrupt

- Microcontrollers are used to perform a set of pre-defined (programmed) activates which triggers the necessary outputs based on the input.
- But, while your Microcontroller is busy with executing one piece of code there might be an emergency situation where other piece of your code needs immediate attention.
- This other piece of code that needs immediate attention should be treated as an interrupt.

Interrupt -Example

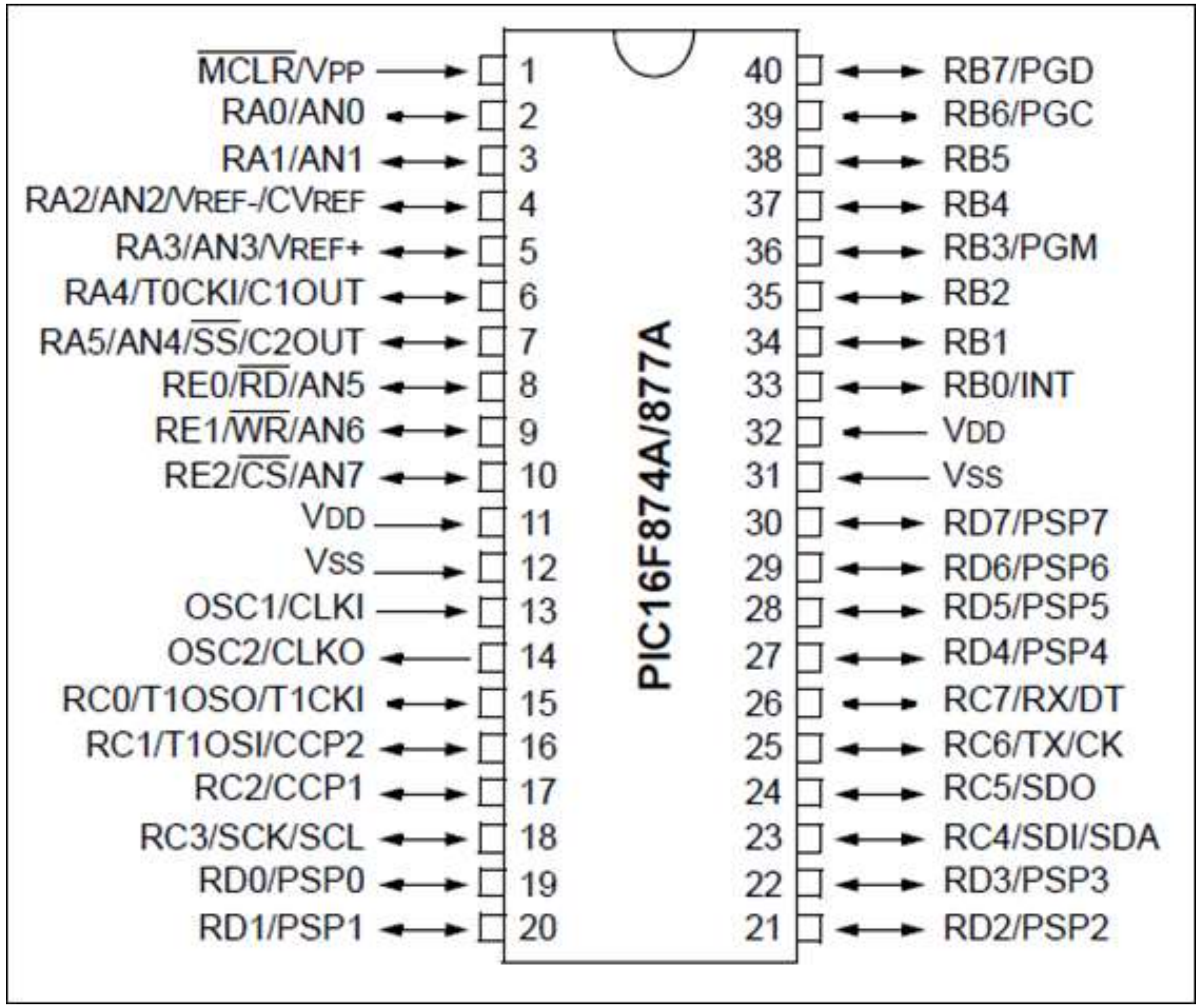
- Let us consider that you are playing your favourite game on your mobile and the controller (assumption) inside your phone is busy throwing all the graphics that is needed for you to enjoy the game.
- But, suddenly your girlfriend calls to your number.
- Now, the worst thing to happen is your mobiles controller to neglecting your girlfriends call since you are busy playing a game.
- To prevent this nightmare from happening we use something called interrupts.

Types of Interrupts

- In Microcontrollers there are two main **types of interrupts**.
- External Interrupt and Internal Interrupt.
- The internal Interrupts occur inside the Microntroller for performing a task, for example Timer Interrupts, ADC Interrupts etc..
- These interrupts are triggered by the software to complete the Timer operation or ADC operation respectively.
- The **external interrupt** is the one that can get triggered by the user.
- when an interrupt is triggered we should notify about it from the Interrupt Service Routine **ISR** and then continue back to incrementing the numbers.

PIC 16F877 A Interrupts

- From this each interrupt should be enabled individually
- RB0(EXTERNAL)
- PORT B CHANGE
- TMR O OVERFLOW
- 14 PERIPHERAL INTERRUPT



When the Interrupt occurs.,

- Code stops executing
- Flags get set
- All interrupts disabled
- Value of PC put on stack
- PC moves to IS vector (line O4h)
- In location 04H, put GOTO ISR
- End ISR with RETFIE (return from interrupt)

Interrupt Service Routine

- The interrupts will always be active listing for some particular actions to happen and when they occur they execute a piece of code and then gets back to the normal function.
- This piece of code is called the **interrupt service routine (ISR)**.
- One practical project in which interrupt is mandatory is "Digital Speedometer and Odometer Circuit using PIC Microcontroller"

Types of Interrupts in PIC 16F877A

PIC 16F877A has the following 15 interrupt sources :

- External
- Timer 0
- Timer 1
- RB Port Change
- Parallel Slave Port Read/Write
- A/D Converter
- USART Receive
- USART Transmit
- Synchronous Serial Port
- CCP1 (Capture, Compare, PWM)
- CCP2 (Capture, Compare, PWM)
- TMR2 to PR2 Match
- Comparator
- EEPROM Write Operation
- Bus Collision

Registers for Interrupts

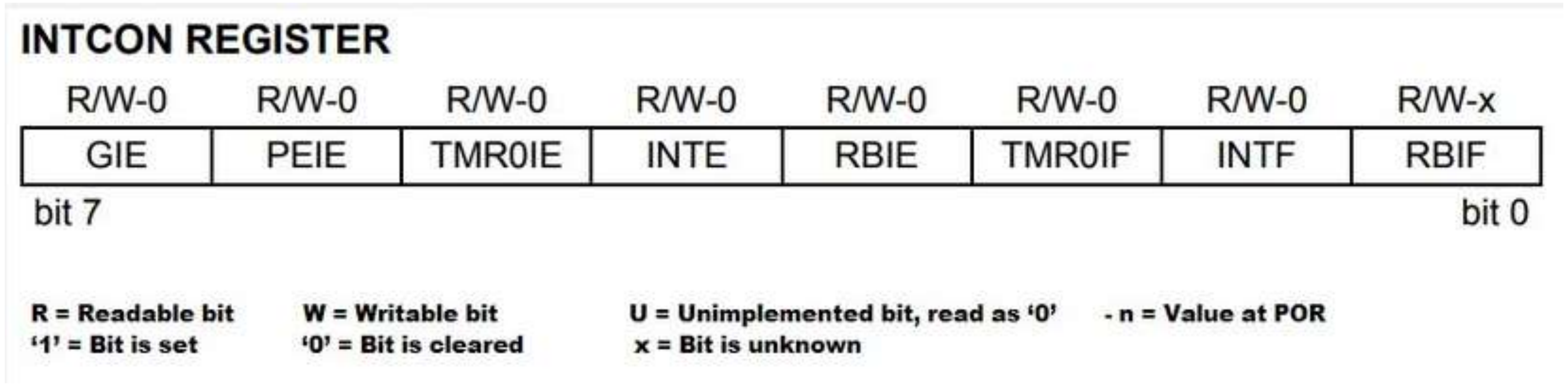
The 5 registers that used to control the operation of Interrupts in PIC 16F877A Microcontroller :

- INTCON
- PIE1
- PIR1
- PIE2
- PIR2

Interrupts

INTCON Register

The INTCON register is a readable and writable register, which contains various enable and flag bits for the TMR0 register overflow, RB port change and external RB0/INT pin interrupts.



INTCON REGISTER

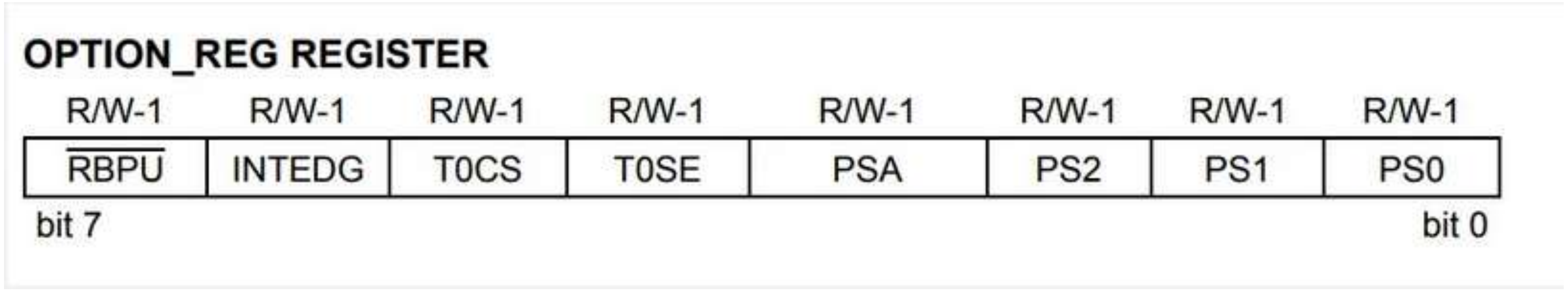
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF
bit 7							bit 0

R = Readable bit
'1' = Bit is set
W = Writable bit
'0' = Bit is cleared
U = Unimplemented bit, read as '0'
x = Bit is unknown
- n = Value at POR

Interrupts

OPTION_REG

The OPTION_REG Register is a readable and writable register, which contains various control bits to configure the TMR0 prescaler/WDT postscaler (single assignable register known also as the prescaler), the external INT interrupt, TMR0 and the weak pull-ups on PORTB.



OPTION_REG REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

Interrupts

PIE1 Register

The PIE1 register contains the individual enable bits for the peripheral interrupts.

PSPIE: Parallel Slave Port Read/Write Interrupt Enable bit**(1)**

Note **(1)**: PSPIE is reserved on PIC16F873A/876A devices; always maintain this bit clear.

ADIE: A/D Converter Interrupt Enable bit

RCIE: USART Receive Interrupt Enable bit

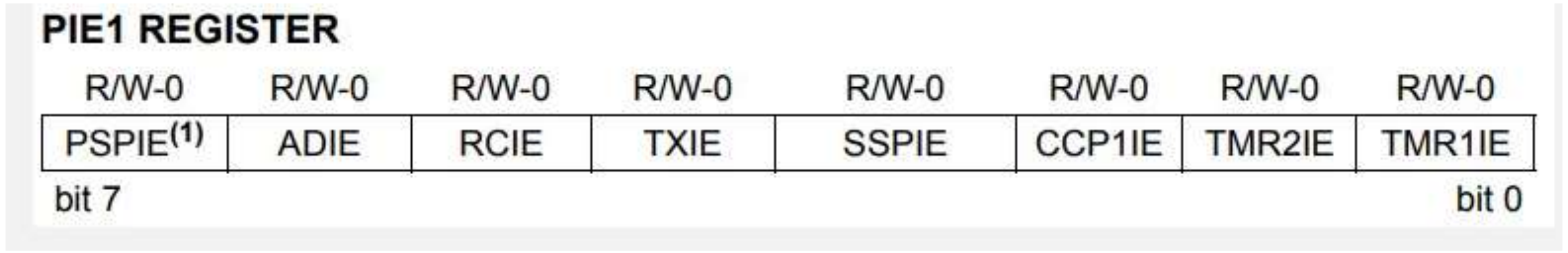
TXIE: USART Transmit Interrupt Enable bit

SSPIE: Synchronous Serial Port Interrupt Enable bit

CCP1IE: CCP1 Interrupt Enable bit

TMR2IE: TMR2 to PR2 Match Interrupt Enable bit

TMR1IE: TMR1 Overflow Interrupt Enable bit



PIE1 REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIE[1]	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

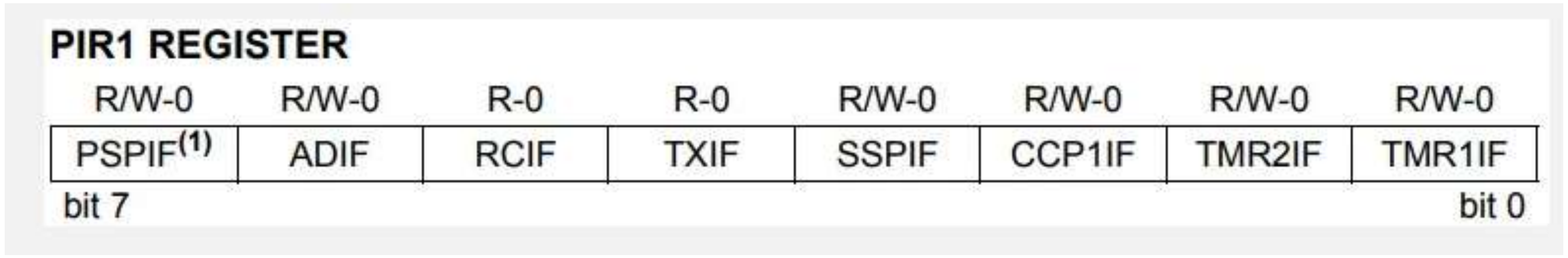
Interrupts

PIR1 Register

The PIR1 register contains the individual flag bits for the peripheral interrupts.

Note: Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>).

User software should ensure the appropriate interrupt bits are clear prior to enabling an interrupt.



PIR1 REGISTER

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF[(1)]	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

References

https://embetronicx.com/tutorials/microcontrollers/pic16f877a/pic16f877a-interrupt-tutorial/

https://www.microcontrollerboard.com/pic_interrupt.html

https://www.youtube.com/watch?v=ZkV4N-fZWBs

Mazidi M. A., McKinlay R. D., Causey D. “PIC Microcontroller And Embedded Systems” Pearson Education International, 2008(Unit I,II,III, IV & V)