Timers in PIC

An Introduction

Virendra Singh

Computer Architecture and Dependable Systems Lab

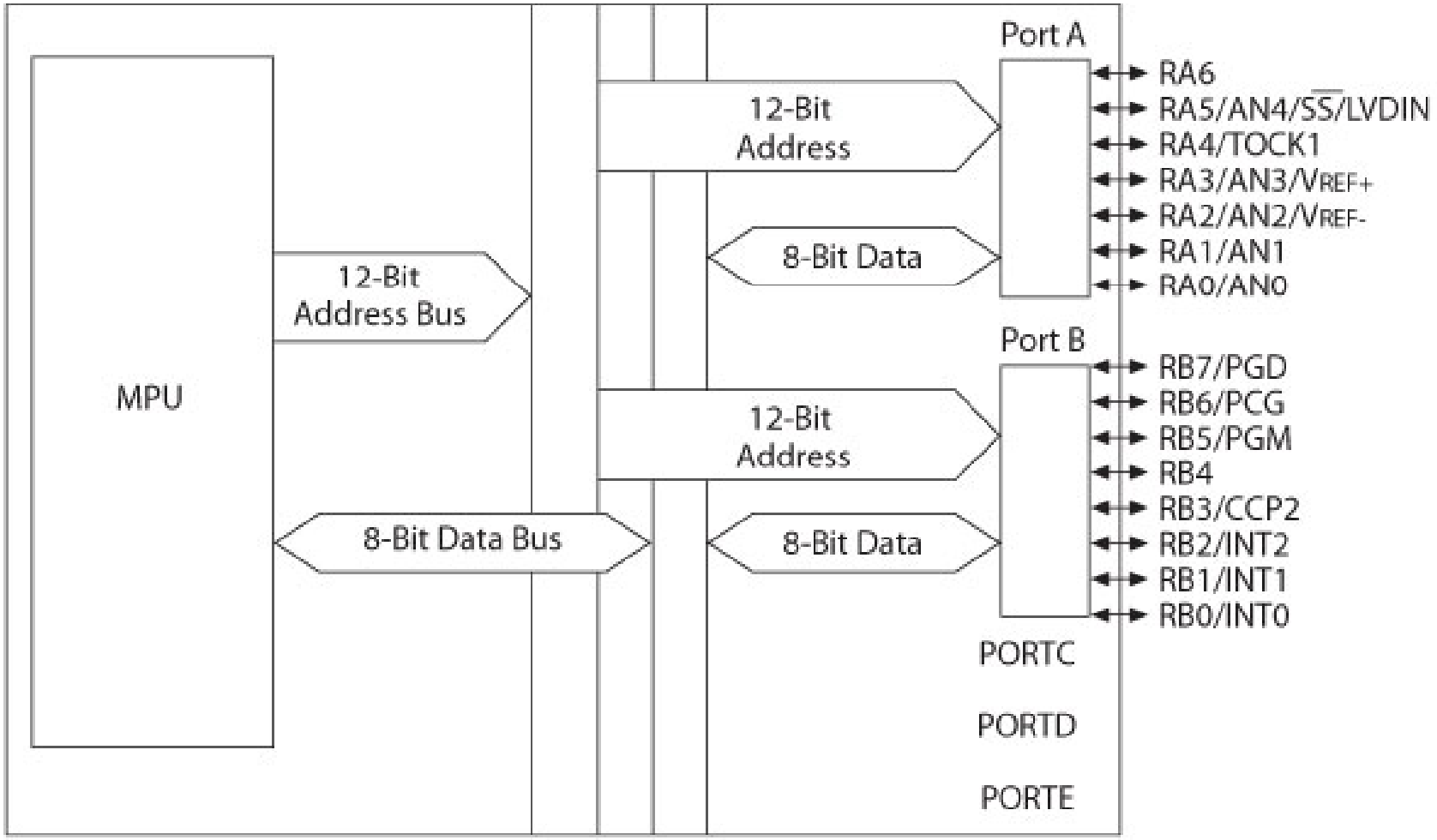
Department of Electrical Engineering

Indian Institute of Technology Bombay

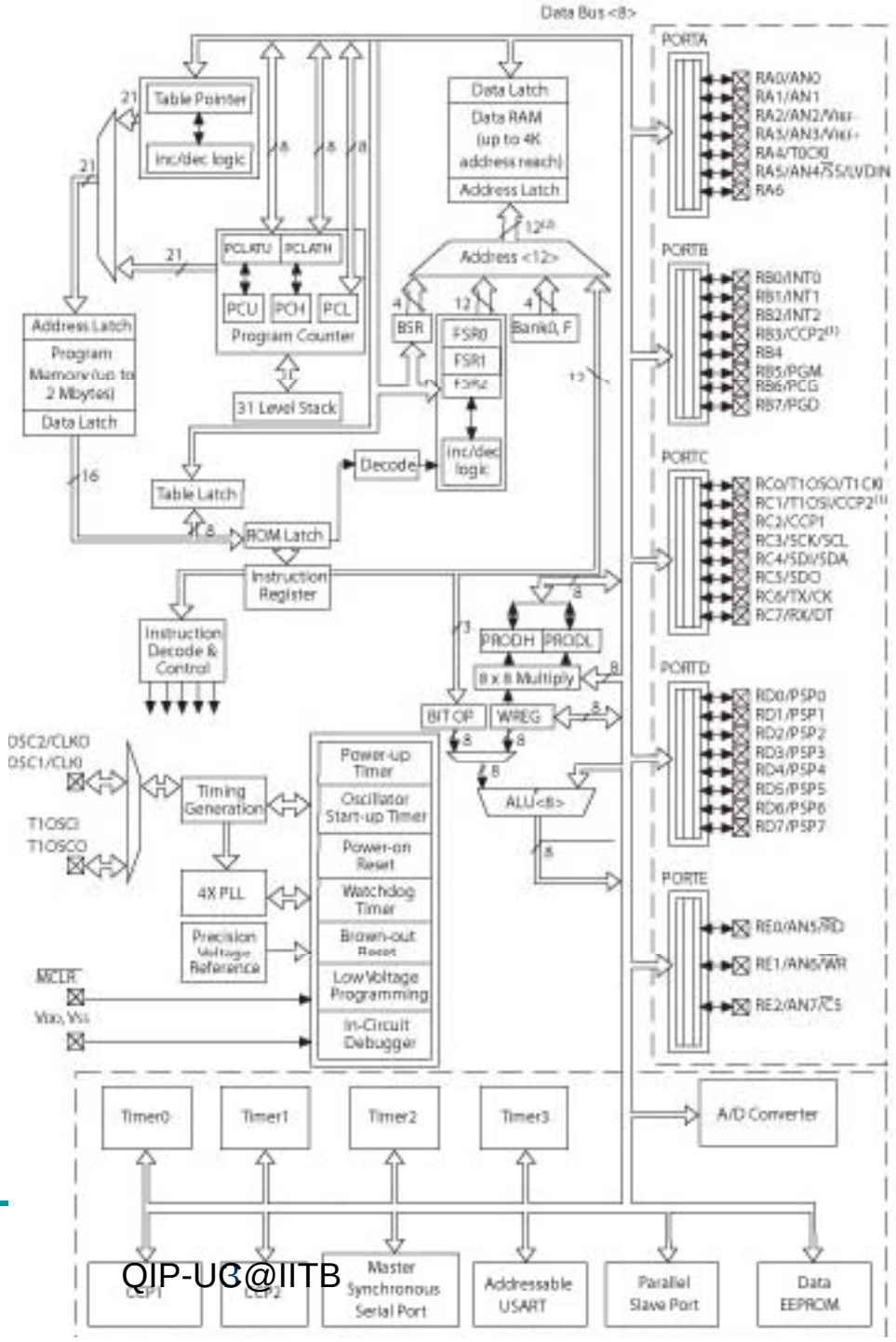
http://www.ee.iitb.ac.in/~viren/

E-mail: viren@ee.iitb.ac.in

I/O Ports A and B



PIC18F4X2 Architecture Block Diagram



Introduction

- The PIC18 has from two (2) to five (5) timers depending on the family member. They are referred to as Timer 0, 1, 2, 3 and 4.
- What Timers can do? 1) Generate a time delay 2) As a Counter to count events happening outside the microcontroller.
- PIC18F4550 has 4 Timers.

Timer0 Timer1

Timer2 Timer3

Introduction

- Every timer needs a clock pulse to tick. The clock source can be internal or external.
- Internal clock: The 1/4th of the frequency of the crystal oscillator on the OSC1 and OSC2 pins (Fosc/4) is fed into the timer. Therefore, it is used for time delay generation. This is called a timer.
- External clock: Fed pulses thru' one of the PIC18's pins: This is called a counter.

Introduction

- Many of the PIC18 timers are 16 bits wide.
- Each 16-bit timer is accessed as two separate register, low byte (TMRxL) and high byte (TMRxH)
- Each timer also has the TCON (Timer Control) register for setting modes of operation.

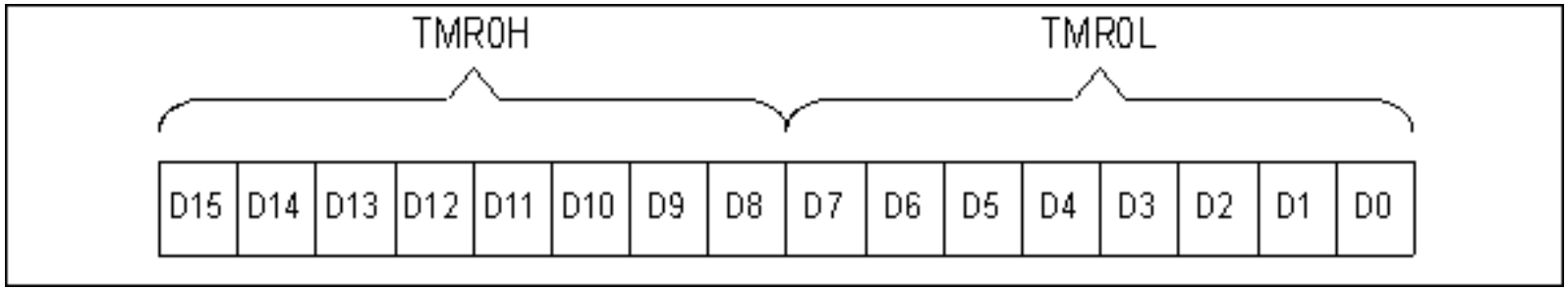
Timer0 Module

The Timer0 module incorporates the following features:

- Software selectable operation as a timer or counter in both 8-bit or 16-bit modes
- Dedicated 8-bit, software programmable prescaler
- Selectable clock source (internal or external)
- Edge select for external clock
- Interrupt-on-overflow

Timer0 Registers and Programming

- Timer0 can be used as an 8-bit or 16-bit timer
- The 16-bit register of Timer0 is accessed as low byte (TMR0L) and high byte (TMR0H)



Timer0 High and Low Registers

Timer0 Control Register (T0CON)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
bit 7							bit 0

bit 7 **TMR0ON:** Timer0 On/Off Control bit
1 = Enables Timer0
0 = Stops Timer0

bit 6 **T08BIT**: Timer0 8-Bit/16-Bit Control bit
1 = Timer0 is configured as an 8-bit timer/counter
0 = Timer0 is configured as a 16-bit timer/counter

bit 5 **T0CS**: Timer0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (CLKO)

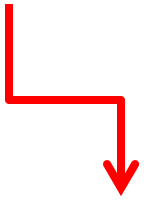
bit 4 **T0SE**: Timer0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA**: Timer0 Prescaler Assignment bit
1 = TImer0 prescaler is not assigned. Timer0 clock input bypasses prescaler.
0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.

bit 2-0 **T0PS2:T0PS0**: Timer0 Prescaler Select bits
111 = 1:256 Prescale value
110 = 1:128 Prescale value
101 = 1:64 Prescale value
100 = 1:32 Prescale value
011 = 1:16 Prescale value
010 = 1:8 Prescale value
001 = 1:4 Prescale value
000 = 1:2 Prescale value

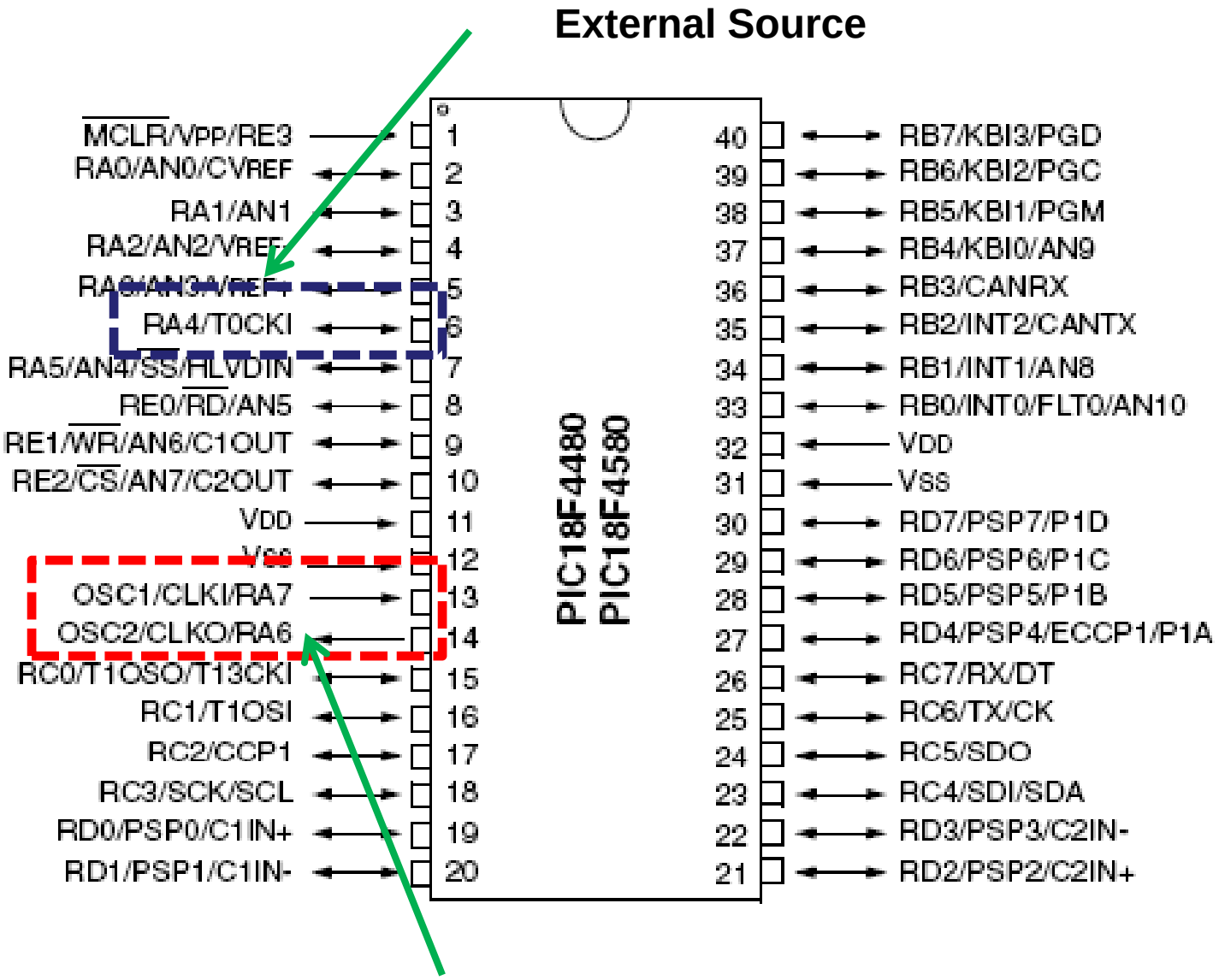
Timer0 Clock Source (T0CS)

- T0CS or bit 5 is used to decide whether the clock source is internal (Fosc/4) or external
- T0CS = 0 ; The Fosc/4 is used as clock source
- T0CS = 1 ; The clock source is external and comes from the RA4/T0CKI (pin 6)



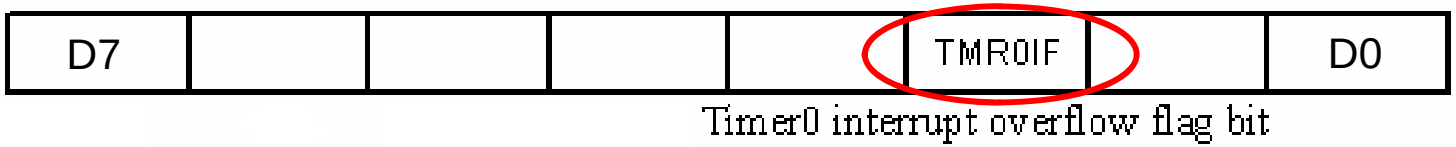
Used as an event counter

Timer0 Clock Source (T0CS)



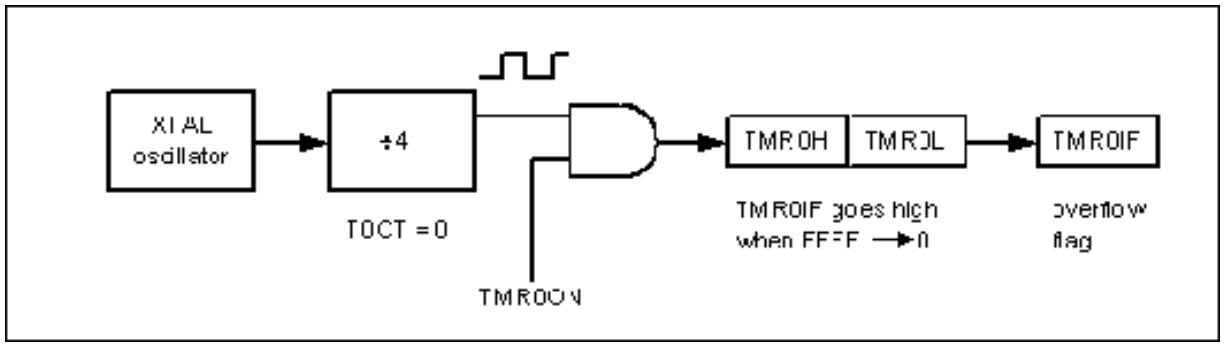
INTCON (Interrupt Control)

- TMR0IF Flag Bit



TMR0IF 0 = Timer0 did not overflow.
1 = Timer0 has overflowed (FFFF to 0000, or FF to 00 in 8-bit mode).

The importance of TMR0IF: In 16-bit mode, when TMR0H:TMR0L overflows from FFFF to 0000 this flag is raised. In 8-bit, it is raised when the timer goes from FF to 00. We monitor this flag bit before we reload the TMR0H:TMR0L registers.



Tomer0 Overflow Flag

16-bit Timer Programming

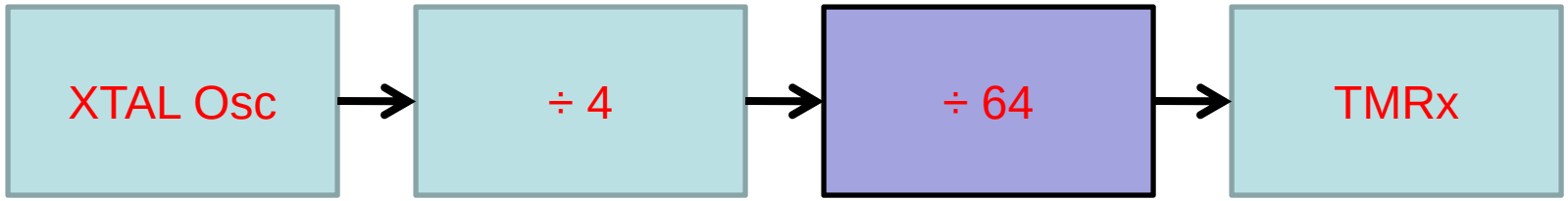
- It allows values of 0000H to FFFFH to be loaded into the registers TMR0H and TMR0L
- After loaded, the timer must be started (BSF T0CON, TMR0ON)
- It start to count up until it reaches its limit of FFFFH. When it rolls over from FFFFH to 0000H, it sets HIGH a flag bit (TMR0IF)
- Repeat the process:
  - 1 Reload the TMR0H and TMR0L
  - 2 TMR0IF flag must be reset to 0

Step to Program Timer0 in 16-bit Mode

1) Load the value into the T0CON register
2) Load register TMR0H followed by register TMR0L
3) Start the timer
4) Keep monitoring the timer flag (TMR0IF)
5) Stop the timer
6) Clear the TMR0IF flag for the next round
7) Go back to the step 2)

Prescaller and Generating a Large Time Delay

- The time delay depends on two factors,
  - The crystal frequency
  - The timer's 16-bit register
- We can use the prescaler option in the T0CON register to increase the delay by reducing the period
- Prescaler option from 2 to 256



8-bit Mode Programming of Timer0

- Set the T0CON value register indicating 8-bit mode
- Load the TMR0L register only!
- Start the timer
- Keep monitoring the timer flag (TMR0IF)
- Stop the timer
- Clear the timer flag
- Reload TMR0L

Please refer Example 9-16

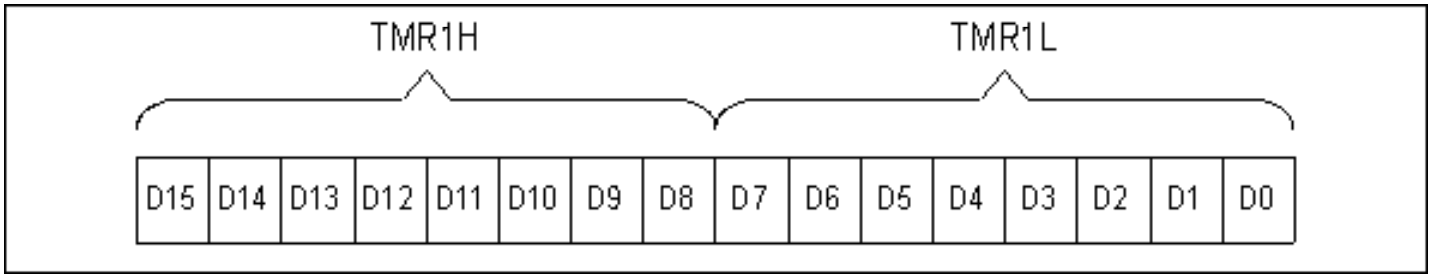
Timer1 Module

The Timer1 timer/counter module incorporates these features:

- Software selectable operation as a 16-bit timer or counter
- Readable and writable 8-bit registers (TMR1H and TMR1L)
- Selectable clock source (internal or external) with device clock or Timer1 oscillator internal options
- Interrupt-on-overflow
- Module Reset on CCP Special Event Trigger
- Device clock status flag (T1RUN)

Timer1 Programming

- 16-bit wide
- Consists of a low-byte (TMR1L) and a high-byte (TMR1H) register
- Can be used as 16-bit timer only!



Timer1

Important Registers:

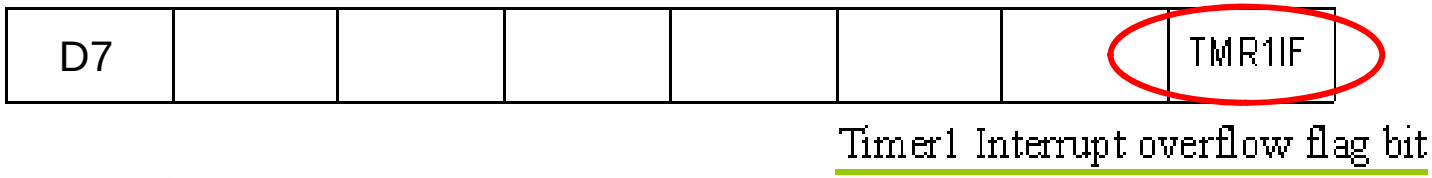
i) T1CON (Timer1 Control Register)

- To start & stop Timer1 and other configurations

ii) TMR1H:TMR1L (for counting purposes)

- Act as counting buffer

iii) PIR1 (Peripheral Interrupt Request Register 1)



TMR1IF 0 = Timer1 did not overflow
1 = Timer1 has overflowed (FFFF to 0000).

The importance of TMR1IF: When TMR1H:TMR1L overflows from FFFF to 0000, this flag is raised. We monitor this flag bit before we reload the TMR1H:TMR1L registers.

Timer1 Control Register

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
bit 7							bit 0

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

bit 7 **RD16:** 16-Bit Read/Write Mode Enable bit
1 = Enables register read/write of Timer1 in one 16-bit operation
0 = Enables register read/write of Timer1 in two 8-bit operations

bit 6 **T1RUN:** Timer1 System Clock Status bit
1 = Device clock is derived from Timer1 oscillator
0 = Device clock is derived from another source

bit 5-4 **T1CKPS1:T1CKPS0:** Timer1 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 3 **T1OSCEN:** Timer1 Oscillator Enable bit
1 = Timer1 oscillator is enabled
0 = Timer1 oscillator is shut off
The oscillator inverter and feedback resistor are turned off to eliminate power drain.

bit 2 **$\overline{\text{T1SYNC}}$:** Timer1 External Clock Input Synchronization Select bit
When TMR1CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
When TMR1CS = 0:
This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

bit 1 **TMR1CS:** Timer1 Clock Source Select bit
1 = External clock from pin RC0/T1OSO/T13CKI (on the rising edge)
0 = Internal clock (FOSC/4)

bit 0 **TMR1ON:** Timer1 On bit
1 = Enables Timer1
0 = Stops Timer1

Example

- Write a program to generate a square wave of 50Hz frequency on pin PORTB.5. Use Timer1 and the maximum pre-scaler allowed

Solution:

T = 1/50Hz = 20mS (Square wave)

½ wave = 20mS/2 = 10mS

10mS/0.4uS/8 = 3125 or C325H

Timer1 Register = F3CBH

Example

```
BCF       TRISB,5
MOVLW 0X30
MOVWF T1CON
MOVLW 0XF3
MOVWF TMR1H
MOVLW 0XCB
MOVWF TMR1L
BCF       INTCON,TMR1IF
CALL      DELAY
BTG       PORTB, RB5
BRA       HERE
;--------------------DELAY USING TIMER 1
DELAY     BSF T1CON,TMR1ON
AGAIN     BTFSS INTCON, TMR1IF
                    BRA AGAIN
                    BCF T1CON, TMR1ON
                    RETURN
```

Timer0 & Timer1 as Counter

- Can used as Counters
- Counter0 (Timer0 counter):
  - Count pulses on T0CKI (RA4) pin
- Counter1 (Timer1 counter):
  - Count pulses on T13CKI (RC0) pin

Timer2 Module

- The Timer2 module timer incorporates the following features:
- 8-Bit Timer and Period registers (TMR2 and PR2, respectively)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4 and 1:16)
- Software programmable postscaler (1:1 through 1:16)
- Interrupt on TMR2-to-PR2 match
- Optional use as the shift clock for the MSSP module

Timer2 *(cont'd)*

Important Registers:

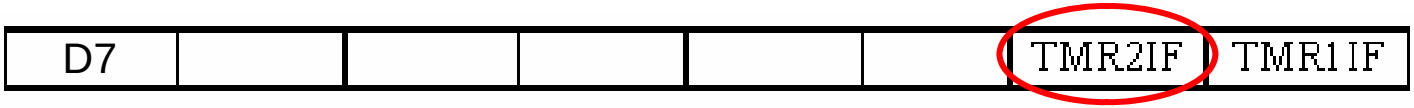
i) T2CON (Timer2 Control Register)

- To start & stop Timer2 and other configurations

ii) PR2 (to set the counting value)

- If TMR2 = PR2; TMR2IF flag is set

iii) PIR1 (Peripheral Interrupt Request Register 1)



TMR2IF Timer2 Interrupt overflow flag bit
0 = TMR2 value is not equal to PR2 register.
1 = TMR2 value is equal to PR2 register.

The location of TMRxIF in the PIR register can vary in future products.

Timer2 *(cont'd)*

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

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

bit 7 **Unimplemented:** Read as 'o'

bit 6-3 **T2OUTPS3:T2OUTPS0**: Timer2 Output Postscale Select bits

0000 = 1:1 Postscale
0001 = 1:2 Postscale
•
•
•
1111 = 1:16 Postscale

bit 2 **TMR2ON**: Timer2 On bit

1 = Timer2 is on
0 = Timer2 is off

bit 1-0 **T2CKPS1:T2CKPS0**: Timer2 Clock Prescale Select bits

00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

Example

- Assume that XTAL=10MHz, write a program to turn on pin PORTB4 when TMR2 reaches value 100 (decimal)

```
BCF     TRISB,4
        BCF     PORTB,4
        MOVLW 0X0
        MOVWF T2CON
        MOVLW 0X0
        MOVWF TMR2
        MOVLW D'100'
        MOVWF PR2
        BCF     PIRI,TMR2IF
        BSF     T2CON, TMR2ON
AGAIN   BTFSS PIRI, TMR2IF
        BRA AGAIN
        BSF PORTB,4
        BCF T2CON,TMR2ON
HERE    BRA HERE
```

Timer3 Module

- The Timer3 module timer/counter incorporates these features:
- Software selectable operation as a 16-bit timer or counter
- Readable and writable 8-bit registers (TMR3H and TMR3L)
- Selectable clock source (internal or external) with device clock or Timer1 oscillator internal options
- Interrupt-on-overflow
- Module Reset on CCP Special Event Trigger

Timer3 *(cont'd)*

Important Registers:

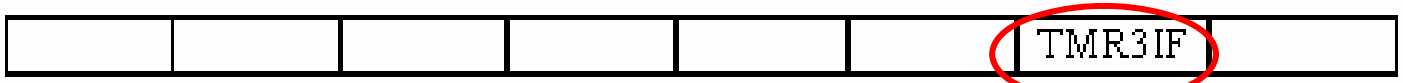
i) T3CON (Timer3 Control Register)

- To start & stop Timer3 and other configurations

ii) TMR3H:TMR3L (for counting purposes)

- Act as counting buffer

iii) PIR2 (Peripheral Interrupt Request Register 2)



TMR3IF Timer3 interrupt overflow flag bit
0 = Timer3 did not overflow
1 = Timer3 has overflowed (FFFF to 0000).

The importance of TMR3IF: In 16-bit mode, when TMR3H:TMR3L overflows from FFFF to 0000, this flag is raised.

The location of TMRxIF in the PIR register can vary in future products.

Timer3 *(cont'd)*

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T3ECCP1(1)	T3CKPS1	T3CKPS0	T3CCP1(1)	$\overline{\text{T3SYNC}}$	TMR3CS	TMR3ON
bit 7							bit 0

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

bit 7 **RD16:** 16-Bit Read/Write Mode Enable bit
1 = Enables register read/write of Timer3 in one 16-bit operation
0 = Enables register read/write of Timer3 in two 8-bit operations

bit 6,3 **T3ECCP1:T3CCP1:** Timer3 and Timer1 to CCP/ECCP Enable bits(1)
1x = Timer3 is the capture/compare clock source for both CCP and ECCP modules
01 = Timer3 is the capture/compare clock source for ECCP;
Timer1 is the capture/compare clock source for CCP
00 = Timer1 is the capture/compare clock source for both CCP and ECCP modules

bit 5-4 **T3CKPS1:T3CKPS0:** Timer3 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 2 **$\overline{\text{T3SYNC}}$:** Timer3 External Clock Input Synchronization Control bit
(Not usable if the device clock comes from Timer1/Timer3.)
When TMR3CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
When TMR3CS = 0:
This bit is ignored. Timer3 uses the internal clock when TMR3CS = 0.

bit 1 **TMR3CS:** Timer3 Clock Source Select bit
1 = External clock input from Timer1 oscillator or T13CKI (on the rising edge after the first falling edge)
0 = Internal clock (Fosc/4)

bit 0 **TMR3ON:** Timer3 On bit
1 = Enables Timer3
0 = Stops Timer3

Summary

- The PIC18 can have up to four or more timers/counters. Depending on the family member
- Timers: Generate Time Delays (using Crystal)
- Counters: Event counter (using Pulse outside)
- Timers are accessed as two 8-bit registers, TMRLx and TMRHx
- Can be used either 8-bit or 16-bit
- Each timer has its own Timer Control register

Thank You