SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

COIMBATORE-35

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

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

COURSE NAME: 19EEB303 / Microcontroller and its Applications

III YEAR / VI SEMESTER

Unit II – **PIC MICROCONTROLLER**

Topic TIMER AND COUNTER

Timers and Counters

- PIC Microcontroller has three Timers/
- Counters:
- Timer 0: 8-bit register TMR0
  - Described in section 11 of RM
- Timer 1: 16-bit register TMR1H | TMR1L
  - Described in section 12 of RM
- Timer 2: 8-bit register TMR2
  - Described in section 13 of RM

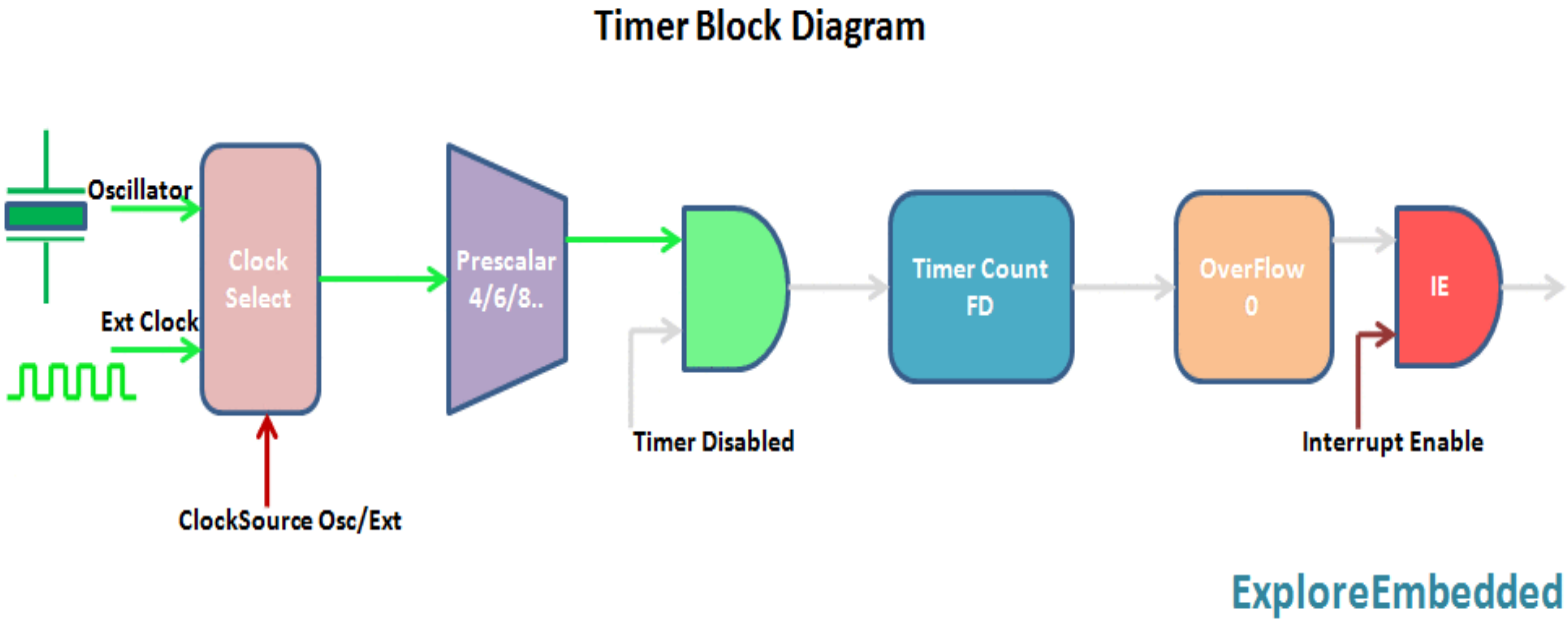
Timers and Counters

Timer Basics

As the name suggests these are used to measure the time or generate the accurate time delay. The microcontroller can also generate/measure the required time delays by running loops, but the timer/counter relieves the CPU from that redundant and repetitive task, allowing it to allocate maximum processing time for other tasks.

Timer is nothing but a simple binary counter that can be configured to count clock pulses(Internal/External). Once it reaches the Max value, it will roll back to zero setting up an **OverFlow** flag and generates the interrupt if enabled.

Timers and Counters



Timers and Counters

Timer 0

The TMR0 module is an 8-bit timer/counter with the following features:

- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select
- Interrupt on overflow from FFh to 00h
- Edge select for external clock

Timer0 Registers

The below table shows the registers associated with PIC16f877A Timer0 module.

Register	Description
OPTION_REG	This registers is used to configure the TIMER0 Prescalar, Clock Source etc
TMR0	This register holds the timer count value which will be incremented depending on prescalar configuration
INTCON	This register contains the Timer0 overflow flag(TMR0IF) and corresponding Inetrrupt Enable flag(TMR0IE).

Timer 0

RBPU: NA for Timers
INTEDG: NA for Timers
T0CS: TMR0 Clock Source Select bit
1-Transition on T0CKI pin
0-Internal instruction cycle clock (CLKO)
T0SE: TMR0 Source Edge Select bit
1-Increment on high-to-low transition on T0CKI pin
0-Increment on low-to-high transition on T0CKI pin
PSA: Prescaler Assignment bit
1-Prescaler is assigned to the WDT
0-Prescaler is assigned to the Timer0

OPTION _REG							
7	6	5	4	3	2	1	0
RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

Timer 0

RBPU: NA for Timers

INTEDG: NA for Timers

T0CS: TMR0 Clock Source Select bit
1-Transition on T0CKI pin
0-Internal instruction cycle clock (CLKO)

T0SE: TMR0 Source Edge Select bit
1-Increment on high-to-low transition on T0CKI pin
0-Increment on low-to-high transition on T0CKI pin

PSA: Prescaler Assignment bit
1-Prescaler is assigned to the WDT
0-Prescaler is assigned to the Timer0

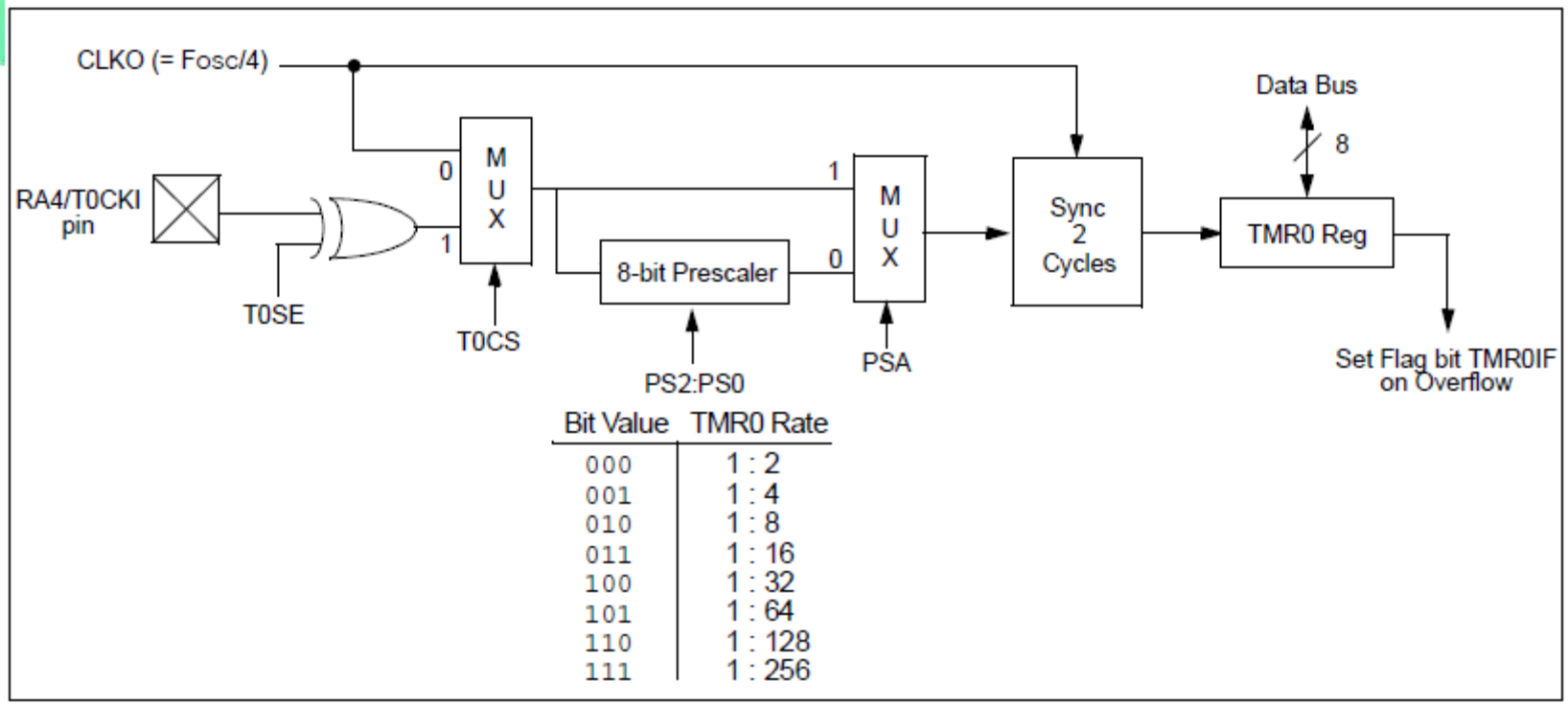
Timer 0

PS2:PS0: Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

Timer 0

Timer0 Block Diagram Excluding WDT



Bit Value	TMR0 Rate
000	1 : 2
001	1 : 4
010	1 : 8
011	1 : 16
100	1 : 32
101	1 : 64
110	1 : 128
111	1 : 256

Delay Calculations for 1ms @20Mhz with Prescalar as 32:

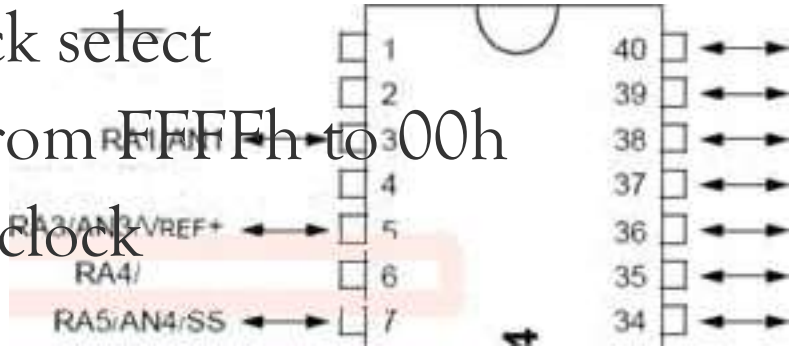
RegValue = 256-(Delay * Fosc)/(Prescalar*4)) = 256-((1ms * 20Mhz)/(32*4)) = 256-156=100

Timer 1

Timer 1

The timer TMR1 module is an 16-bit timer/counter with the following features:

- 16-bit timer/counter with two 8-Bit register TMR1H/TMR1L
- Readable and writable
- software programmable prescaler upto 1:8
- Internal or external clock select
- Interrupt on overflow from FFFFh to 0000h
- Edge select for external clock



Timer 1

imer1 Registers

The below table shows the registers associated with PIC16f877A Timer1 module.

Register	Description
T1CON:	This registers is used to configure the TIMER1 Prescalar, Clock Source etc
TMRIH	This register holds the higher 8-bits of timer value. TMR1H and TMR1L are used in pair to increment from 0000 - FFFFh
TMRIL	This register holds the lower 8-bits of timer value. TMR1H and TMR1L are used in pair to increment from 0000 - FFFFh
PIR1	This register contains the Timer1 overflow flag(TMR1IF).
PIE1	This register contains the Timer1 Interrupt Enable flag(TMR1IE).

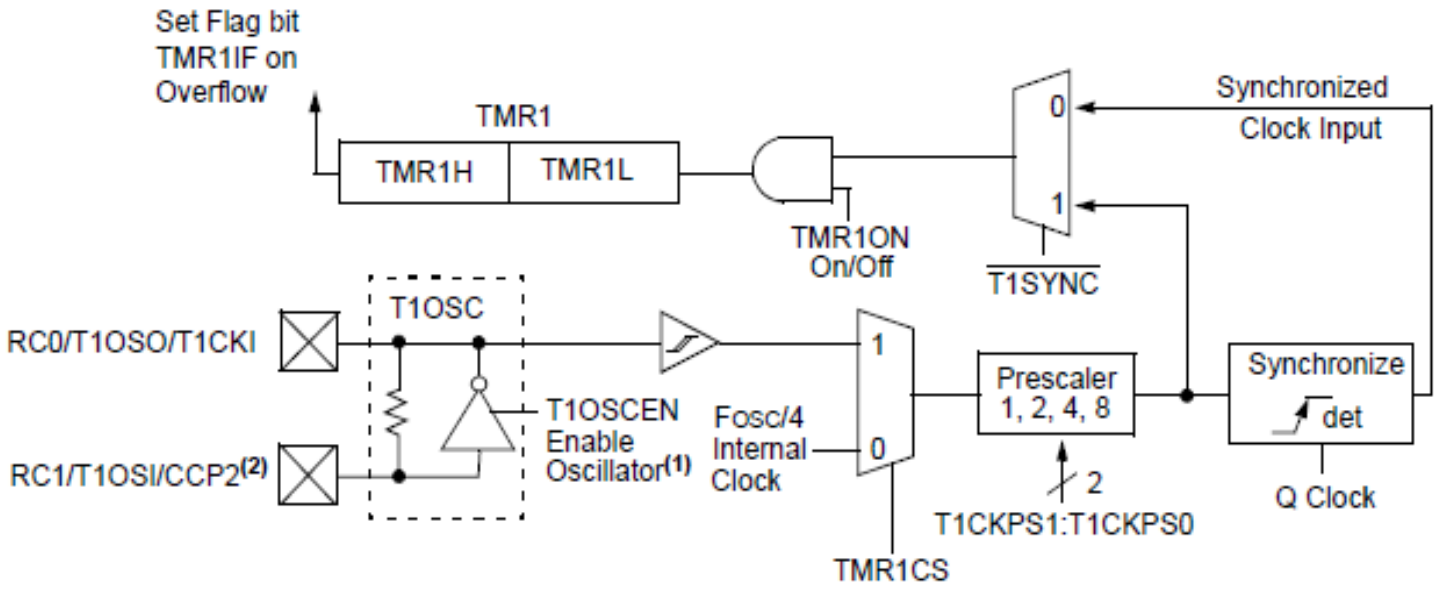
Timer 1

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

T1CON							
7	6	5	4	3	2	1	0
—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON

Timer 1

TIMER1 BLOCK DIAGRAM



Note 1: When the T1OSCEN bit is cleared, the inverter is turned off. This eliminates power drain.

Delay Calculations for 100ms @20Mhz with Prescalar as 8:

RegValue = 65536-(Delay * Fosc)/(Prescalar*4)) = 65536-((100ms * 20Mhz)/(8*4)) = 3036 = 0x0BDC

Timer 2

Timer 2

The TImer2 module is an 8-bit timer/counter with the following features:

- 8-bit timer/counter
- Readable and writable
- Software programmable prescaler/PostScaler upto 1:16
- Interrupt on overflow from FFh to 00h

Timer2 Registers

The below table shows the registers associated with PIC16f877A Timer0 module.

Timer 2

Register	Description
T2CON	This registers is used to configure the TIMER2 Prescalar, Clock Source etc
TMR2	This register holds the timer count value which will be incremented depending on prescalar configuration
PIR1	This register contains the Timer2 overflow flag(TMR2IF).
PIE1	This register contains the Timer2 Interrupt Enable flag(TMR2IE).

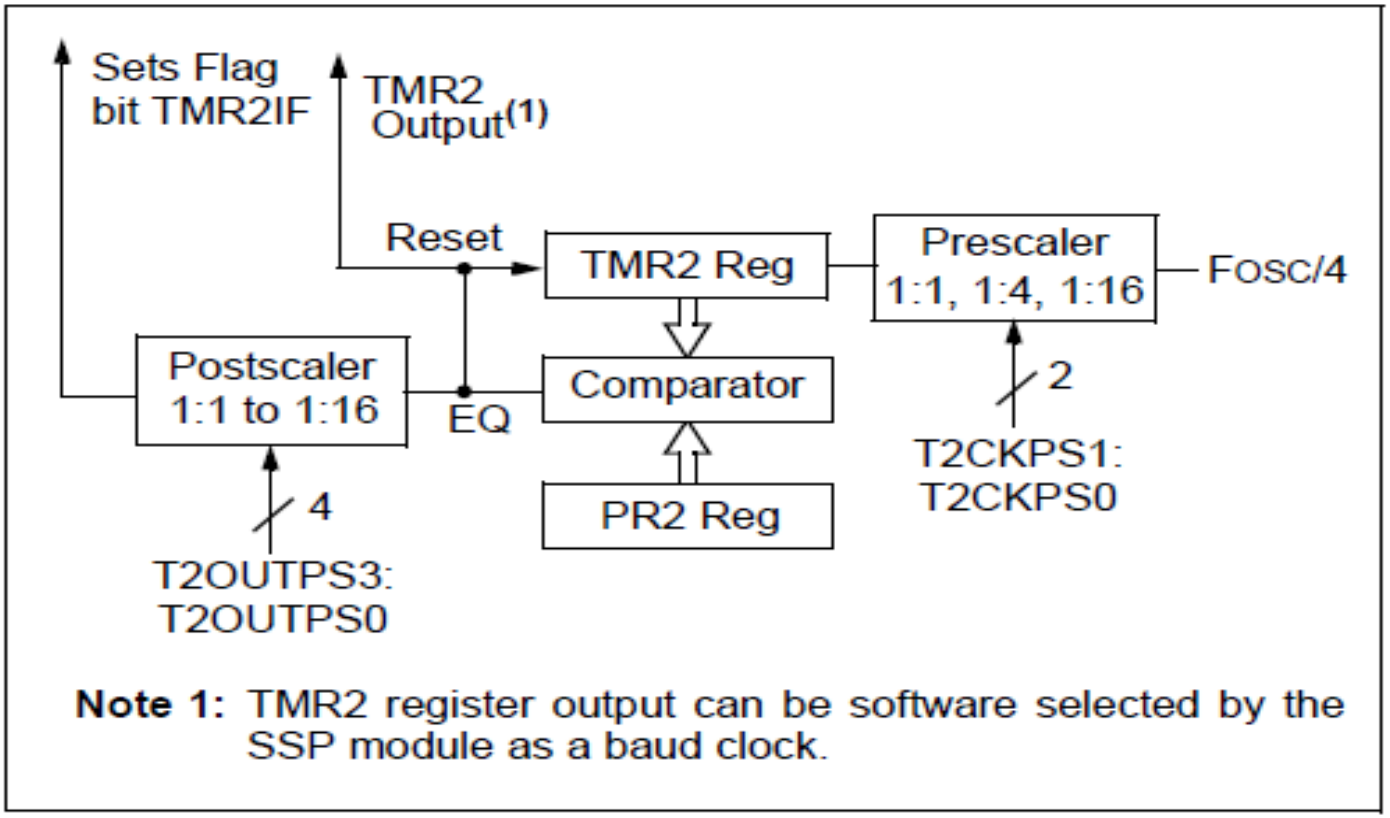
Timer 2

TOUTPS3:TOUTPS0: Timer2 Output Postscale Select bits
0000 = 1:1 postscale
0001 = 1:2 postscale
0010 = 1:3 postscale
•
•
•
1111 = 1:16 postscale

T2CON							
7	6	5	4	3	2	1	0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0

Timer 2

TIMER2 BLOCK DIAGRAM



Note 1: TMR2 register output can be software selected by the SSP module as a baud clock.

Timer 2

Generating 1sec delay using Timer2:

As the timer2 is 8-bit and supports 1:16 prescalar, it is not possible to directly generate the delay of 1sec. The max delay with 1:16 prescaler will be:

Delay = 256 * (Prescaler*4)/Fosc = 256 * 16*4/20Mhz=819us

Now 500us can be generated using timers which will be used to increment a counter 2000 times to get 1sec delay.

Delay Calculations for 500usec @20Mhz with Prescalar as 16:

RegValue = 256-(Delay * Fosc)/(Prescalar*4)) = 256-((500us* 20Mhz)/(16*4)) = 256-156=100