PIC18 Timer Programming

Hsiao-Lung Chan
Dept Electrical Engineering
Chang Gung University, Taiwan
chanhl@mail.cgu.edu.tw

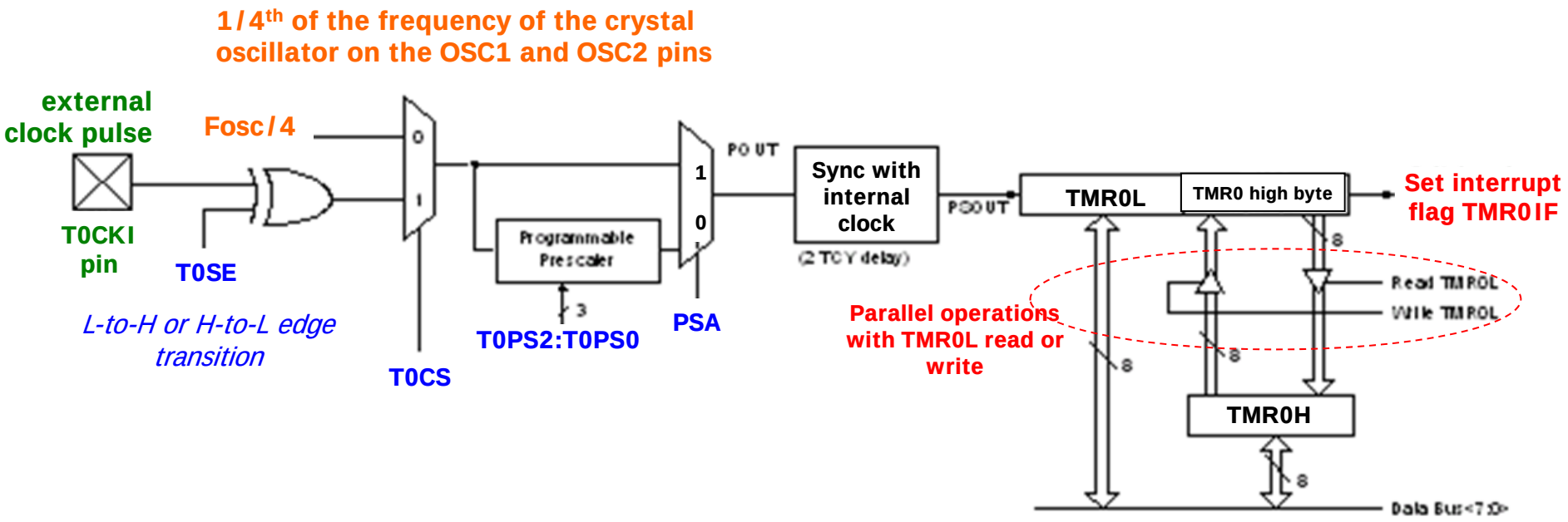
Objectives

- Introduce PIC18 timers
  - Timer 0, Timer 1, Timer 2, Timer 3
- Generate a time delay using a timer
- Generate a square wave using a timer
- Count external pulses using a timer
- Use C as well as assembly in timer programming

Functions of PIC18 timer

- Generate a time delay (計時器)
  - Use internal clock pulse, Fosc/4
- As a counter to count events happening outside the microcontroller (計數器)
  - Use external clock pulse

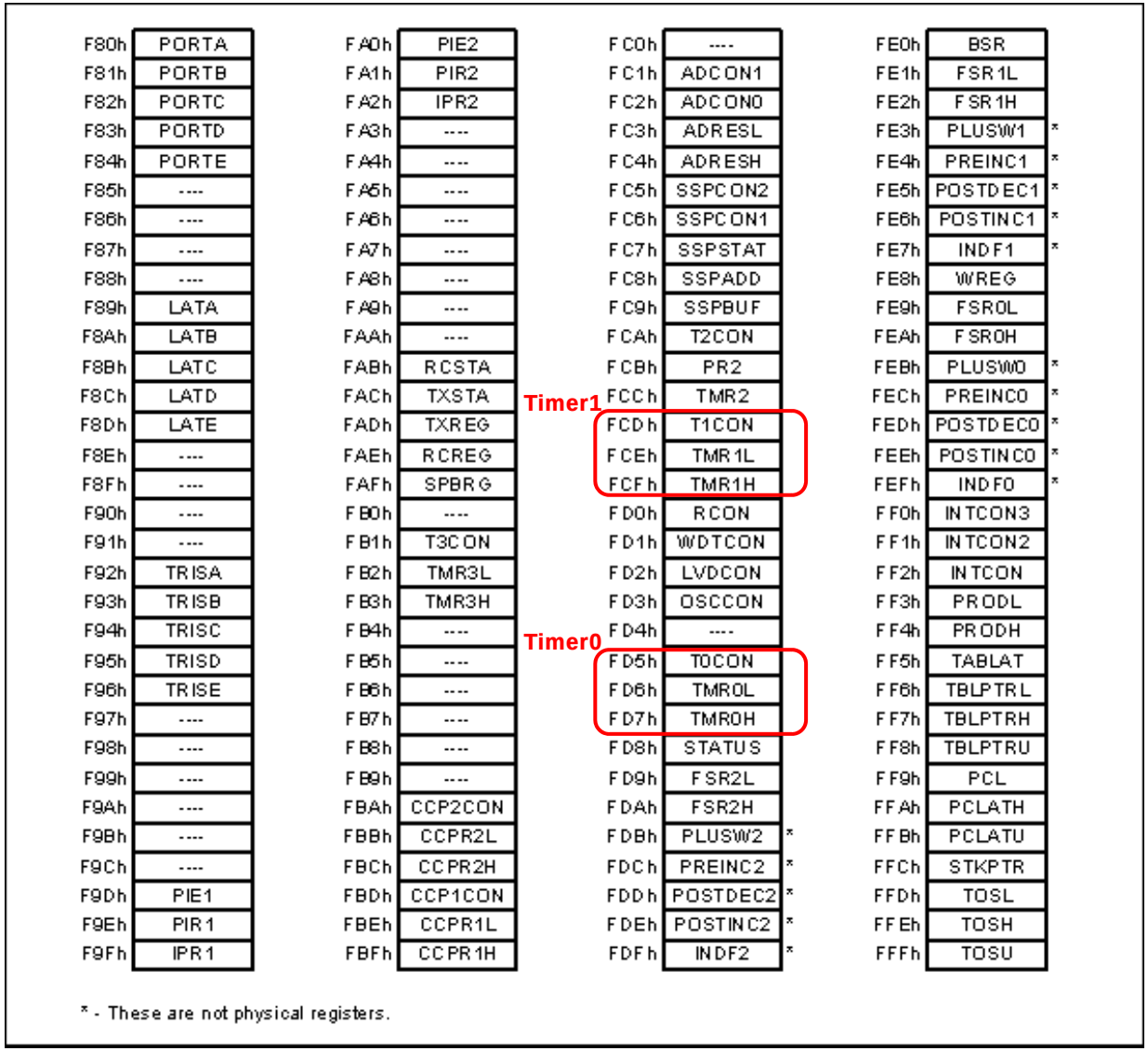
Timer0 16-bit block diagram



- Should load TMR0H first, and then TMR0L

(Figure 9-5)

Registers of the timer



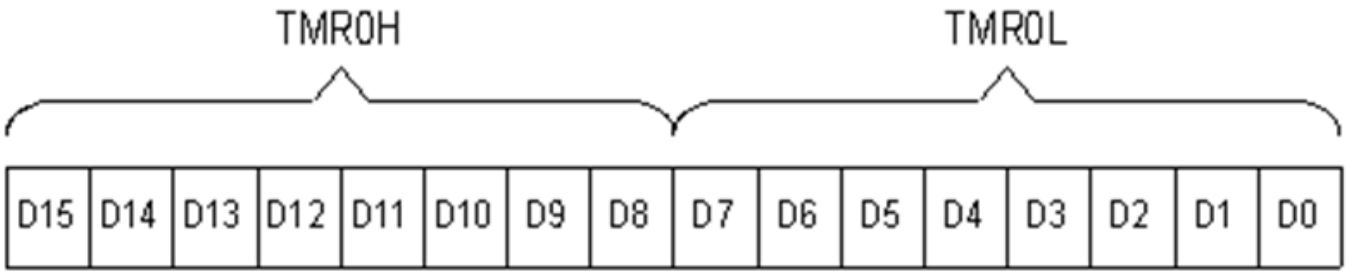
F80h	PORTA	FA0h	PIE2	FC0h	----		FE0h	BSR			
F81h	PORTB	FA1h	PIR2	FC1h	ADCON1		FE1h	FSR1L			
F82h	PORTC	FA2h	IPR2	FC2h	ADCON0		FE2h	FSR1H			
F83h	PORTD	FA3h	----	FC3h	ADRESL		FE3h	PLUSW1	*		
F84h	PORTE	FA4h	----	FC4h	ADRESH		FE4h	PREINC1	*		
F85h	----	FA5h	----	FC5h	SSPCON2		FE5h	POSTDEC1	*		
F86h	----	FA6h	----	FC6h	SSPCON1		FE6h	POSTINC1	*		
F87h	----	FA7h	----	FC7h	SSPSTAT		FE7h	INDF1	*		
F88h	----	FA8h	----	FC8h	SSPADD		FE8h	WREG			
F89h	LATA	FA9h	----	FC9h	SSPBUF		FE9h	FSR0L			
F8Ah	LATB	FAAh	----	FCAh	T2CON		FEAh	FSR0H			
F8Bh	LATC	FABh	RCSTA	FCBh	PR2		FEBh	PLUSW0	*		
F8Ch	LATD	FACh	TXSTA	FCCh	TMR2	Timer1	FECh	PREINC0	*		
F8Dh	LATE	FADh	TXREG	FCDh	T1CON		FEDh	POSTDEC0	*		
F8Eh	----	FAEh	RCREG	FCEh	TMR1L		FEEh	POSTINC0	*		
F8Fh	----	FAFh	SPBRG	FCFh	TMR1H		FEFh	INDF0	*		
F90h	----	FB0h	----	FD0h	RCON		FF0h	INTCON3			
F91h	----	FB1h	T3CON	FD1h	WDTCON		FF1h	INTCON2			
F92h	TRISA	FB2h	TMR3L	FD2h	LVDCON		FF2h	INTCON			
F93h	TRISB	FB3h	TMR3H	FD3h	OSCCON		FF3h	PRODL			
F94h	TRISC	FB4h	----	FD4h	----	Timer0	FF4h	PRODH			
F95h	TRISD	FB5h	----	FD5h	T0CON		FF5h	TABLAT			
F96h	TRISE	FB6h	----	FD6h	TMR0L		FF6h	TBLPTRL			
F97h	----	FB7h	----	FD7h	TMR0H		FF7h	TBLPTRH			
F98h	----	FB8h	----	FD8h	STATUS		FF8h	TBLPTRU			
F99h	----	FB9h	----	FD9h	FSR2L		FF9h	PCL			
F9Ah	----	FBAh	CCP2CON	FDAh	FSR2H		FFAh	PCLATH			
F9Bh	----	FBBh	CCPR2L	FDBh	PLUSW2	*	FFBh	PCLATU			
F9Ch	----	FBCh	CCPR2H	FDCh	PREINC2	*	FFCh	STKPTR			
F9Dh	PIE1	FBDh	CCP1CON	FDDh	POSTDEC2	*	FFDh	TOSL			
F9Eh	PIR1	FBEh	CCPR1L	FDEh	POSTINC2	*	FFEh	TOSH			
F9Fh	IPR1	FBFh	CCPR1H	FDFh	INDF2	*	FFFh	TOSU			

* - These are not physical registers.

Special function registers (SFRs)

Timer0 registers

- Timer0 can be used as an 8-bit or 16-bit timer
- Can be assessed like any SFRs
  - MOVWF TMR0L
  - MOVFF TMR0H, PORTC



(Figure 9-1)

T0CON (Timer0 control) register

- Set various timer operation mode

TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0

TMR0ON D7 Timer0 ON and OFF control bit
1 = Enable (start) Timer0
0 = Stop Timer0

T0SBIT D6 Timer0 8-bit/16-bit selector bit
1 = Timer0 is configured as an 8-bit timer/counter.
0 = Timer0 is configured as a 16-bit timer/counter.

T0CS D5 Timer0 clock source select bit
1 = External clock from RA4/T0CKI pin
0 = Internal clock (Fosc/4 from XTAL oscillator)

T0SE D4 Timer0 source edge select bit
1 = Increment on H-to-L transition on T0CKI pin
0 = Increment on L-to-H transition on T0CKI pin

PSA D3 Timer0 prescaler assignment bit
1 = Timer0 clock input bypasses prescaler.
0 = Timer0 clock input comes from prescaler output.

T0PS2:T0PS0 D2 D1 D0 Timer0 prescaler selector

0 0 0 = 1:2	Prescale value (Fosc / 4 / 2)
0 0 1 = 1:4	Prescale value (Fosc / 4 / 4)
0 1 0 = 1:8	Prescale value (Fosc / 4 / 8)
0 1 1 = 1:16	Prescale value (Fosc / 4 / 16)
1 0 0 = 1:32	Prescale value (Fosc / 4 / 32)
1 0 1 = 1:64	Prescale value (Fosc / 4 / 64)
1 1 0 = 1:128	Prescale value (Fosc / 4 / 128)
1 1 1 = 1:256	Prescale value (Fosc / 4 / 256)

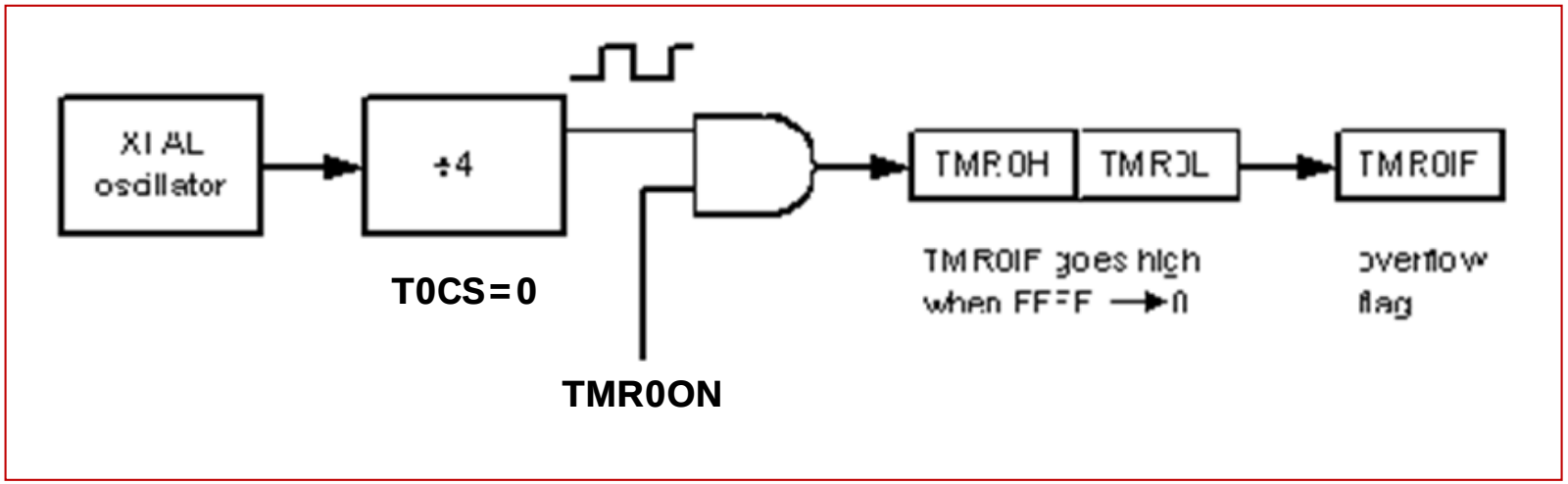
(Figure 9-2)

TMR0IF flag bit in INTCON (interrupt control register)

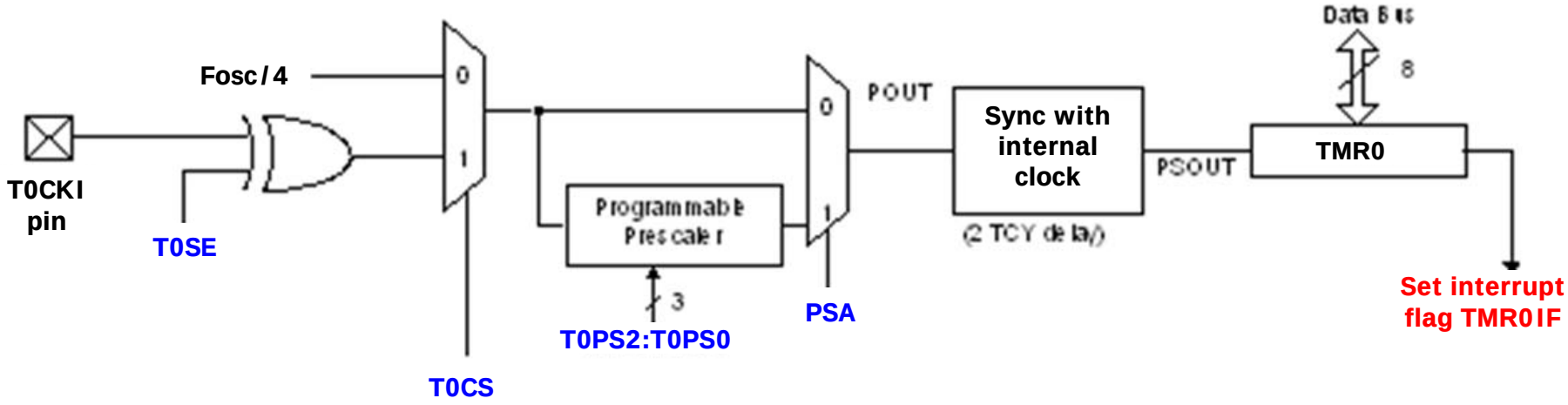
					TMR0IF		

TMR0IF D2 Timer0 interrupt overflow flag bit
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. **(Figure 9-3,4)**



8-bit mode programming of Timer0



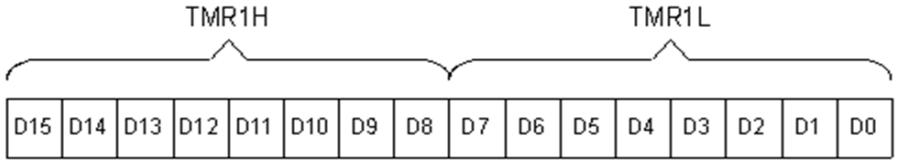
(Figure 9-7)

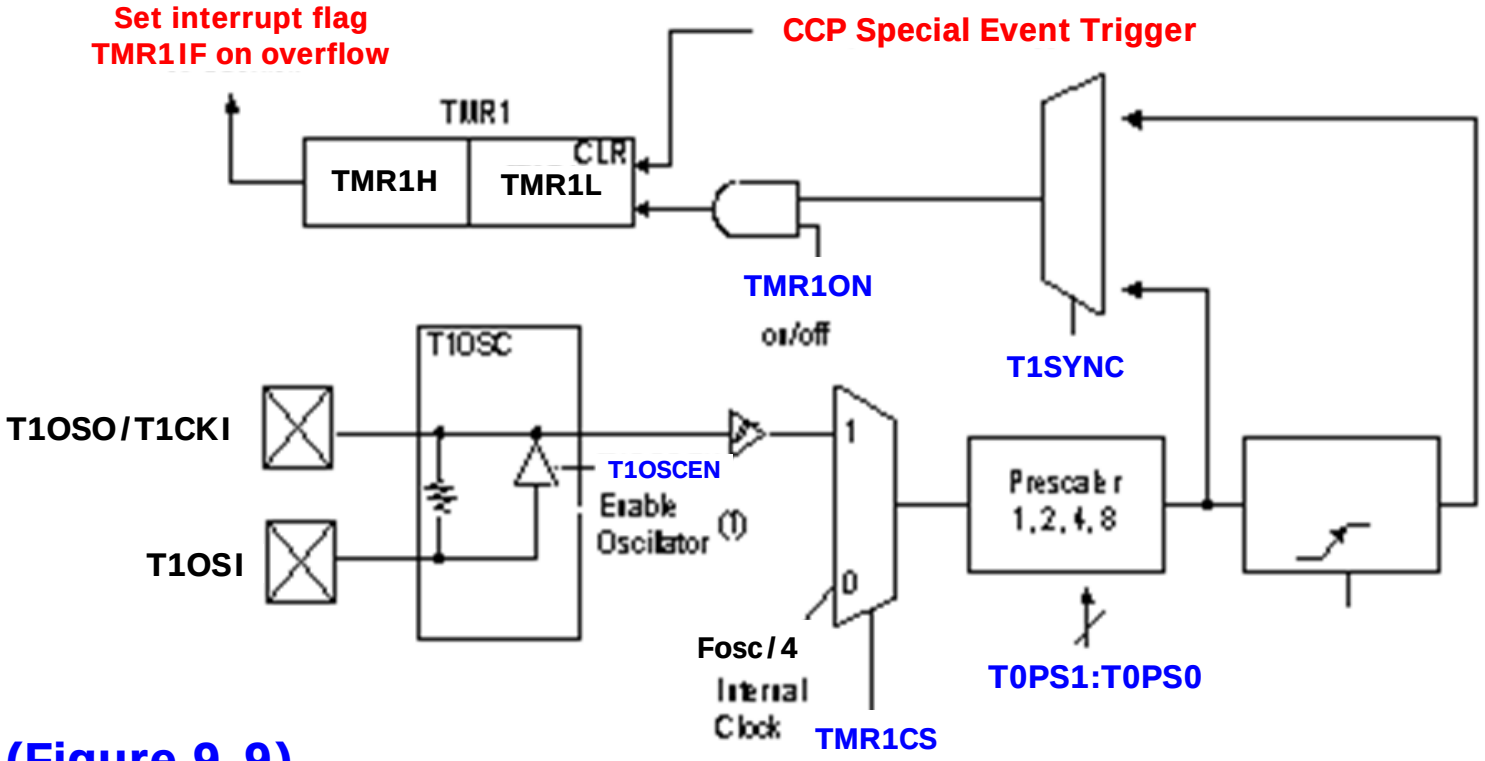
Examples using Timer 0

- Assembly programming
  - Find frequency of square wave using 16-bit timer (Example 9-7)
  - Find frequency of square wave using 8-bit timer (Example 9-19)
- C programming
  - Toggle RB5 every 50 ms using 16-bit timer (Example 9-29)
  - Generate 2-Hz square wave using 8-bit timer (Example 9-30)

Timer1

- Only support 16-bit mode





(Figure 9-9)

T1CON (Timer1 control) register

RD16	---	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON

RD16 D7 16-bit read/write enable bit
1 = Timer1 16-bit is accessible in one 16-bit operation.
0 = Timer1 16-bit is accessible in two 8-bit operations.

D6 Not used

T1CKPS2:T1CKPS0 D5 D4 Timer1 prescaler selector
0 0 = 1:1 Prescale value
0 1 = 1:2 Prescale value
1 0 = 1:4 Prescale value
1 1 = 1:8 Prescale value

T1OSCEN D3 Timer1 oscillator enable bit
1 = Timer1 oscillator is enabled.
0 = Timer1 oscillator is shutoff

T1SYNC D2 Timer1 synchronization (used only when TMR1CS = 1 for counter mode to synchronize external clock input)
If TMR1CS = 0 this bit is not used.

TMR1CS D1 Timer1 clock source select bit
1 = External clock from pin RC0/T1CKI
0 = Internal clock (Fosc/4 from XTAL)

TMR1ON D0 Timer1 ON and OFF control bit
1 = Enable (start) Timer1
0 = Stop Timer1

(Figure 9-10)

PIR1 (interrupt control register 1)

TMR1IF D1 Timer1 Interrupt overflow flag bit
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.

(Figure 9-11)

Examples using Timer 1

- Generate a square wave using assembly (Example 9-21)
- Generate a square wave using C (Example 9-33)

Counter programming

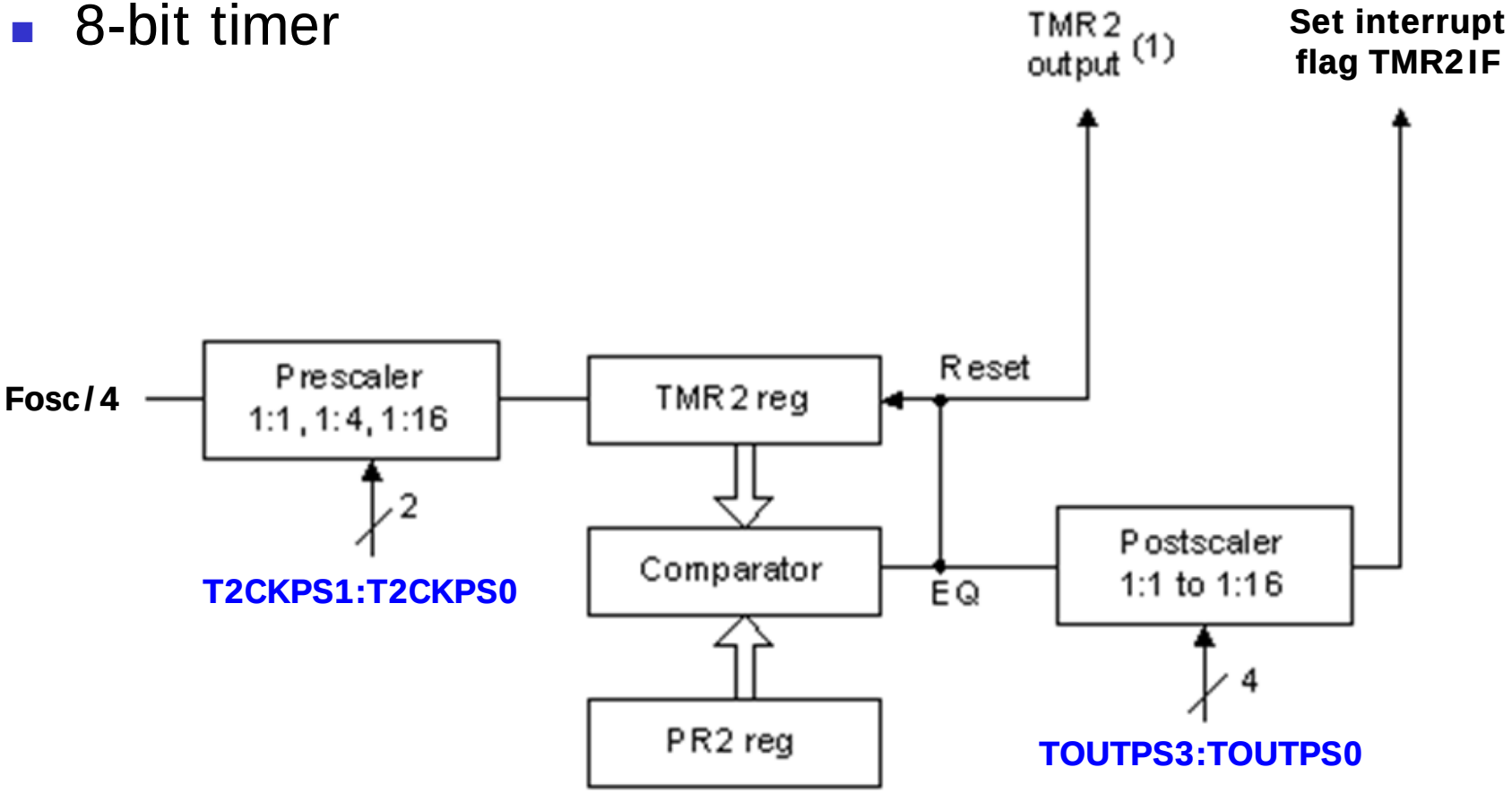
- Using Timer 0
  - Get pulses from T0CKI (Timer0 clock input): RA4 (PORTA.4)
    - Set T0CS (Timer0 clock source) in T0CON to 1
- Using Timer 1
  - External clock is fed into T1CKI pin
    - External clock source option (TMR1CS) is set to 1
    - Timer1 oscillator enable bit (T1OSCEN) is set to 1

Examples for counter programming

- Assembly programming
  - Count pulse using 8-bit Timer 0 (Example 9-26)
  - Count pulse using 16-bit Timer 1 (Example 9-27)
- C programming
  - Count pulse using 16-bit Timer 0 (Example 9-35)

Timer2 block diagram

- 8-bit timer



(Figure 9-12)

T2CON (Timer2 control) register

TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0

D7 Not used

TOUTPS3:TOUTPS0 D6–D3 Timer2 Output Postcale Select bits
- 00 0 0 = 1:1 Postscale value
- 00 0 1 = 1:2 Postscale value
- 00 1 0 = 1:3 Postscale value
- 00 1 1 = 1:4 Postscale value

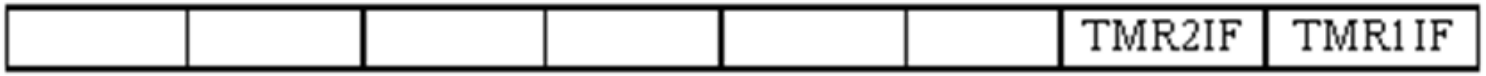
- 11 1 0 = 1:15 Postscale value
- 11 1 1 = 1:16 Postscale value

TMR2ON D2 Timer2 ON and OFF Control bit
- 1 = Enable (Start) Timer2
- 0 = Stop Timer2

T2CKPS1:T2CKPS0 D1–D0 Timer2 Clock Prescale Select bits
- 0 0 = Prescale is 1
- 0 1 = Prescale is 4
- 1 x = Prescale is 16

(Figure 9-13)

PIR1 (interrupt control 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.

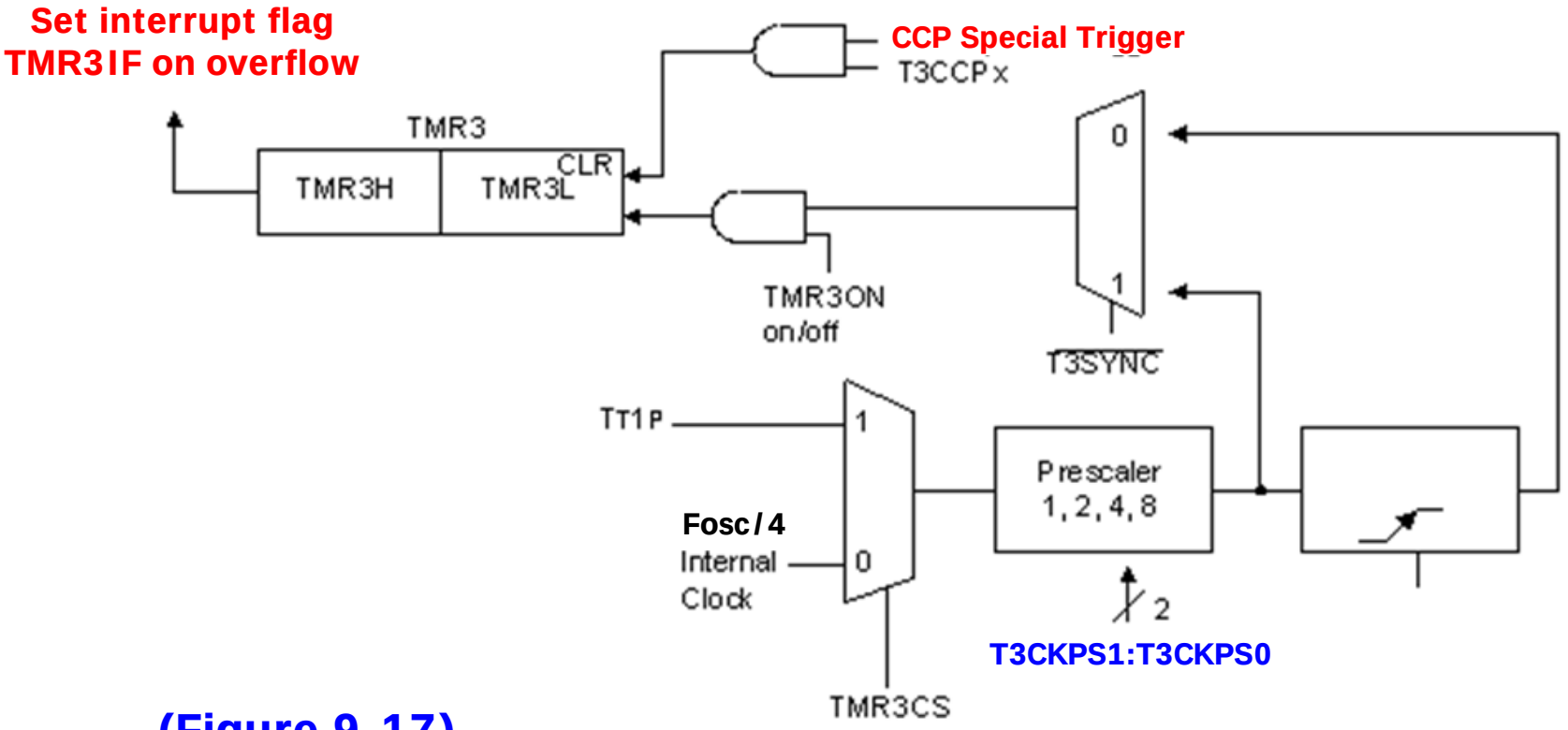
(Figure 9-14)

Examples using Timer 2

- Turn on after a period of time using assembly (Example 9-38)
- Generate a time delay using C (Example 9-41)

Timer3 block diagram

- 16-bit timer or counter



(Figure 9-17)

T3CON (Timer3 control) register

RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYNC	TMR3CS	TMR3ON

RD16 D7 16-bit read/write enable bit
1 = Timer3 16-bit is accessible in one 16-bit operation.
0 = Timer3 16-bit is accessible in two 8-bit operations.

T3CCP2:T3CCP1 D6 D3 Timer3 and Timer1 to CPPx Enable bits
0 0 = Timer1 is the clock source for compare/capture of the CCP module.
0 1 = Timer3 is the clock source for compare/capture of the CCP2.
Timer1 is the clock source for compare/capture of the CCP1.
1 x = Timer3 is the clock source for compare/capture of the CCP module.

T3CKPS1:T3CKPS0 D5 D4 Timer3 Input Clock Prescaler Selector
0 0 = 1:1 Prescale value
0 1 = 1:2 Prescale value
1 0 = 1:4 Prescale value
1 1 = 1:8 Prescale value

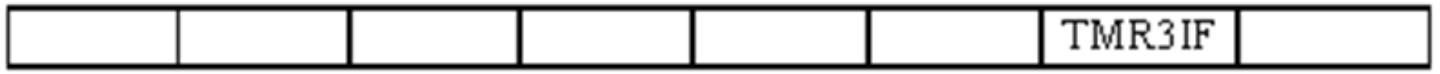
T1SYNC D2 Timer3 external clock input synchronization control bit
Used only when TMR3CS = 1 and clock comes from an external source. If TMR3CS = 0, this bit is not used.
1 = Do not synchronize external clock input
0 = Synchronize external clock input

TMR3CS D1 Timer3 clock source select bit
1 = External clock from pin T1OSI or T1CKI
0 = Internal clock (Fosc/4)

TMR3ON D0 Timer3 On and Off control bit
1 = Enable (start) Timer1
0 = Stop Timer1

(Figure 9-15)

PIR3 (interrupt control register 3)



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.

(Figure 9-16)

Generate a square wave of 50 Hz on the PORTB.5 bit if XTAL = 10 MHz

```
        BCF     TRISB, 5
        MOVLW   0x0         ; Timer3, 16-bit, internal clock, no prescale
        MOVWF   T3CON
HERE    MOVLW   0x9E        FFFFH – 9E58H + 1 = 25,000 clocks
        MOVWF   TMR3H       ➔ time delay = 25,000 x 0.4 μs = 10 ms
        MOVLW   0x58          frequency = 1/(10 ms x 2) = 50 Hz
        MOVWF   TMR3L
        BCF     PIR2, TMR3IF  ; clear Timer3 interrupt flag
        CALL    DELAY
        BTG     PORTB, RB5
        BRA     HERE

DELAY   BSF     T3CON, TMR3ON   ; start Timer3
AGAIN   BTFSS   PIR2, TMR3IF    ; monitor Timer3 interrupt flag
        BRA     AGAIN
        BCF     T3CON, TMR3ON   ; stop Timer3
        RETURN
```

(Example 9-43)

Reference

- M.A. Mazidi, R.D. Mckinlay, D Causey, PIC Microcontroller and Embedded Systems Using Assembly and C for PIC18, Pearson Education Inc., 2008.
- Han-Way Huang, PIC Microcontroller: An Introduction to Software and Hardware Interfacing, Thomson Delmar Learning, 2005.