Timer

Hua-Hsi Tseng
曾華璽

Networked Embedded Applications and Technologies Lab

Department of Computer Science and Information Engineering
National Cheng Kung University, TAIWAN

Outline

- Introduction to Timer
  - PIC18F4520 Internal Oscillator
- Timer1
- Timer2

Introduction to Timer

Timer1
Timer2

What's Timer?

- Timer簡單的說就是一個會持續不斷的計算進入Timer中Clock數量的模組。

- Timer一般有兩種稱呼Timer或Counter。
  - 輸入的Clock時間未知:僅能得知計數到多少個Clock,稱為Counter。
  - 輸入的Clock時間已知:可以進一步換算出時間, 稱為Timer。

- 8-bit MCU的Timer為遞增型,也就是會固定從”0”開始計數。模組可以設定計數到多少Clock後要通知CPU。

Interrupt & Timer

- 通知: 就是中斷旗標(Interrupt Flag) 。當數到設定值時, 中斷旗標會被模組設定為”1”, 對CPU發出中斷請求 (Interrupt Request)。
  - 如果此時中斷致能沒開, 則必須由程式自行檢查事件是否發生。
  - 如果此時中斷致能有開, 就會自動進入中斷服務常式。

- 舉例:Timer0的中斷控制在 INTCON register中控制

REGISTER 9-1: 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/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
bit 7							bit 0

bit 2 **TMR0IF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

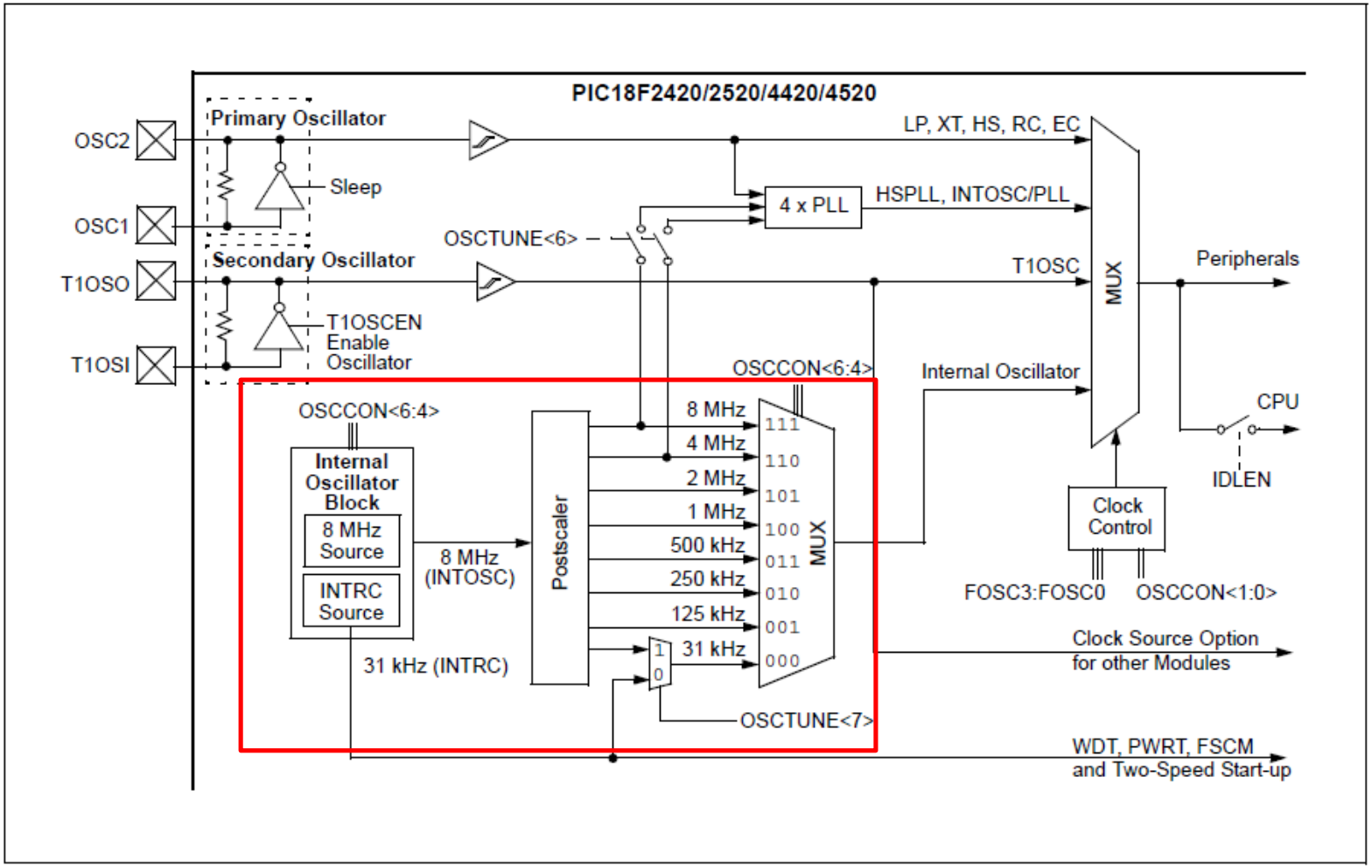
bit 5 **TMR0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 overflow interrupt
0 = Disables the TMR0 overflow interrupt

PIC18F4520 Timer Module

- Timer0:
  - Timer0 可設定為8-bit 或16-bit 模式
  - 計時器產生溢位時FFh to 00h (FFFFh to 0000h)，即產生中斷
- Timer1 and Timer3:
  - 16-bit 模式的計數器或計時器
  - 由兩個可讀/寫的8-bit 計數器串聯而成
  - 預除器有四種選擇：÷1，÷2，÷4，or ÷8
  - 當計數器或計時器產生溢位時FFFFh to 0000h，即產生中斷
- Timer2:
  - 8-bit 模式的計時器，有預除器及後除器之功能
  - TMR2 會自動加一並與設定的值相比；若相等則送出訊號至後除器或產生中斷，並自動將自己清除為零，重新計時

PIC18F4520 Internal Oscillator

FIGURE 2-8: PIC18F2420/2520/4420/4520 CLOCK DIAGRAM



PIC18F4520 Internal Oscillator

- 程式設定範例:

```
3    ; CONFIG1H
4      CONFIG  OSC = INTIO67
```

R 23-1: CONFIG1H: CONFIGURATION REGISTER 1 HIGH (BYTE ADDRESS 300001h)

R/P-0	R/P-0	U-0	U-0	R/P-0	R/P-1	R/P-1	R/P-1
IESO	FCMEN	—	—	FOSC3	FOSC2	FOSC1	FOSC0
bit 7							bit 0

bit 7 **IESO:** Internal/External Oscillator Switchover bit
1 = Oscillator Switchover mode enabled
0 = Oscillator Switchover mode disabled

bit 6 **FCMEN:** Fail-Safe Clock Monitor Enable bit
1 = Fail-Safe Clock Monitor enabled
0 = Fail-Safe Clock Monitor disabled

bit 5-4 **Unimplemented:** Read as '0'

bit 3-0 **FOSC3:FOSC0:** Oscillator Selection bits
11xx = External RC oscillator, CLKO function on RA6
101x = External RC oscillator, CLKO function on RA6
1001 = Internal oscillator block, CLKO function on RA6, port function on RA7
1000 = Internal oscillator block, port function on RA6 and RA7
0111 = External RC oscillator, port function on RA6
0110 = HS oscillator, PLL enabled (Clock Frequency = 4 x FOSC1)
0101 = EC oscillator, port function on RA6
0100 = EC oscillator, CLKO function on RA6
0011 = External RC oscillator, CLKO function on RA6
0010 = HS oscillator
0001 = XT oscillator
0000 = LP oscillator

PIC18F4520 Internal Oscillator

- Default Internal Oscillator Frequency: 1MHz

2-2: OSCCON REGISTER

R/W-0	R/W-1	R/W-0	R/W-0	R[(1)]	R-0	R/W-0	R/W-0
IDLEN	IRCF2	IRCF1	IRCF0	OSTS	IOFS	SCS1	SCS0
bit 7							bit 0

bit 7 **IDLEN:** Idle Enable bit
1 = Device enters Idle mode on SLEEP instruction
0 = Device enters Sleep mode on SLEEP instruction

bit 6-4 **IRCF2:IRCF0:** Internal Oscillator Frequency Select bits
111 = 8 MHz (INTOSC drives clock directly)
110 = 4 MHz
101 = 2 MHz
100 = 1 MHz[(3)]
011 = 500 kHz
010 = 250 kHz
001 = 125 kHz
000 = 31 kHz (from either INTOSC/256 or INTRC directly)[(2)]

bit 3 **OSTS:** Oscillator Start-up Time-out Status bit[(1)]
1 = Oscillator start-up time-out timer has expired; primary oscillator is running
0 = Oscillator start-up time-out timer is running; primary oscillator is not ready

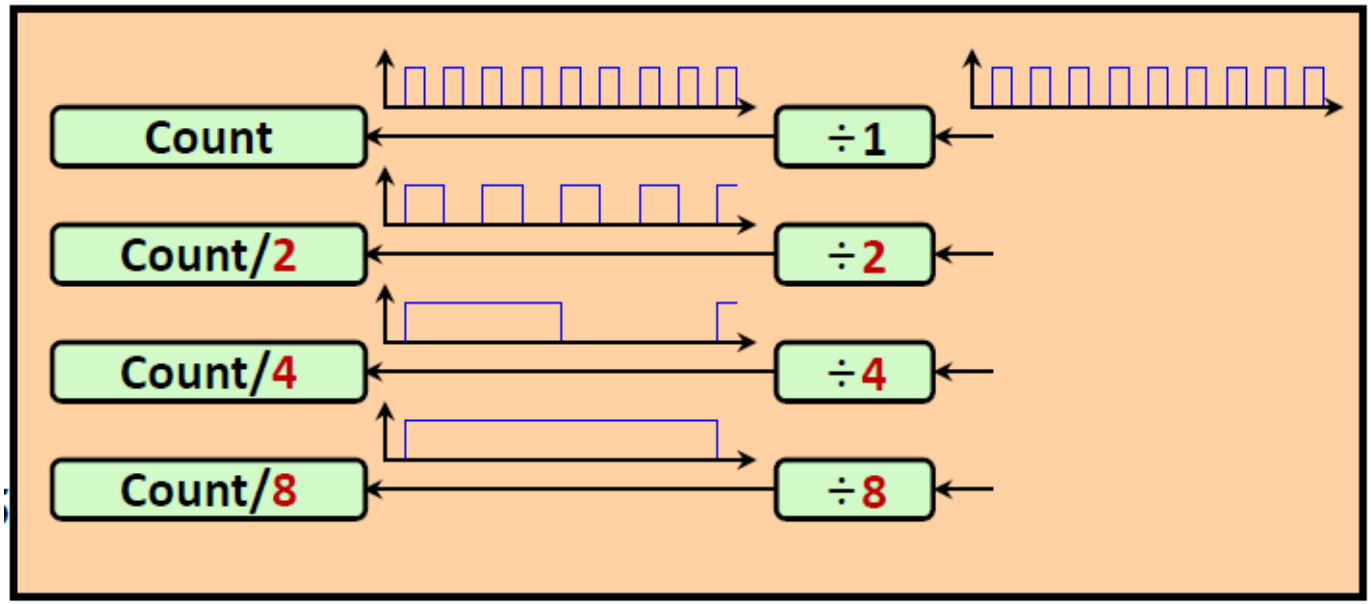
bit 2 **IOFS:** INTOSC Frequency Stable bit
1 = INTOSC frequency is stable
0 = INTOSC frequency is not stable

bit 1-0 **SCS1:SCS0:** System Clock Select bits
1x = Internal oscillator block
01 = Secondary (Timer1) oscillator
00 = Primary oscillator

Note 1: Reset state depends on state of the IESO configuration bit.
2: Source selected by the INTSRC bit (OSCTUNE<7>), see text.
3: Default output frequency of INTOSC on Reset.

Prescaler & Postscaler

- 16-bit Timer計數範圍0~65,535。如果需要的計數值超過時, 必須透過預除器修正。預除器可以用來擴大計數範圍。
- 舉例:
  - 如果把預除器設為除8(1:8),則變成Clock每8個才會有一個進入TMRx, 意即TMRx數到1,000時,實際的Clock已經產生了8,000個。



Try it

- 觀察設定CONFIG1H後OSCCON Register的預設值和每個bit代表的意義與變化
  - CONFIG OSC = INTIO67

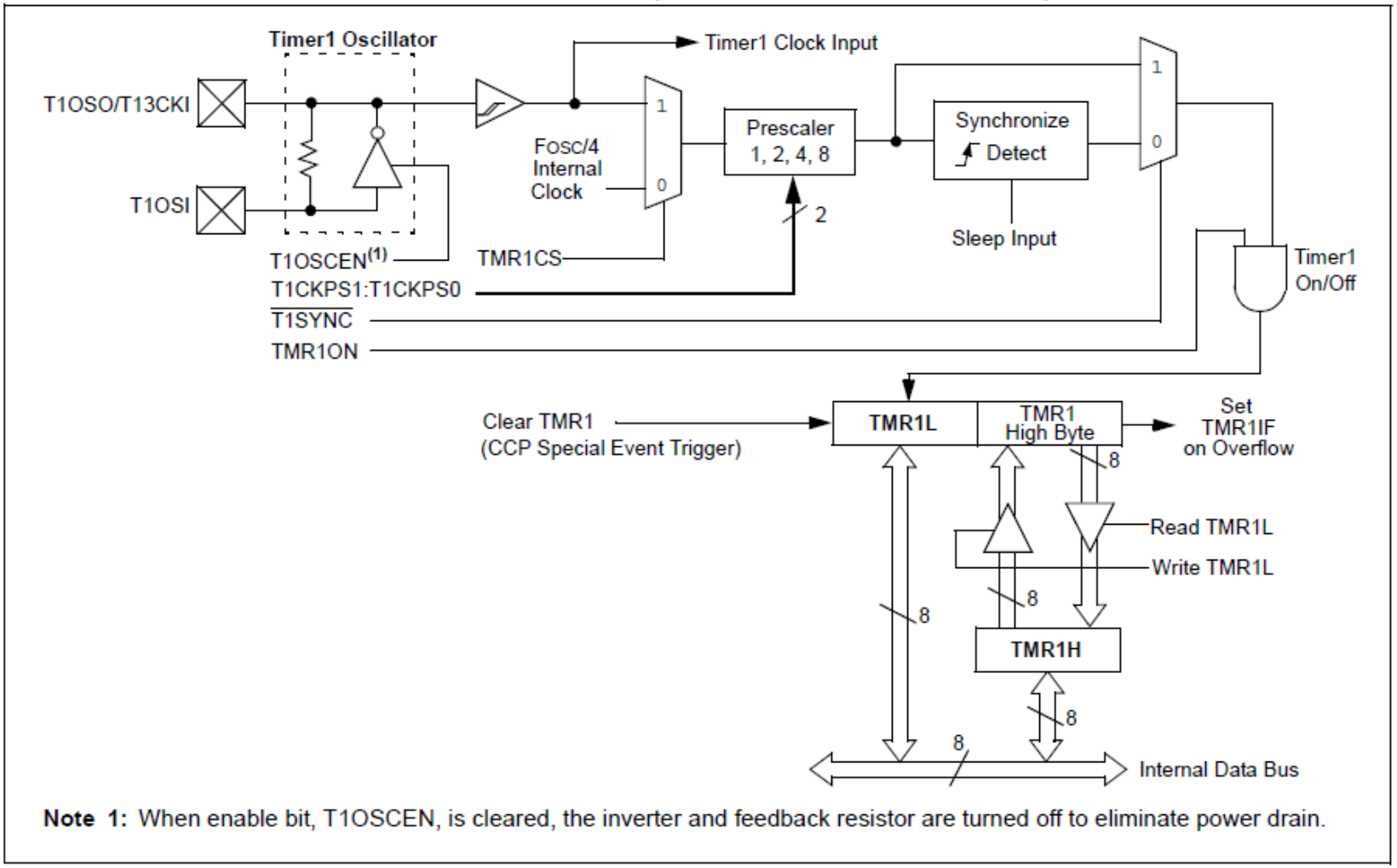
Introduction to Timer

Timer1

Timer2

Timer1 Block diagram

FIGURE 12-2: TIMER1 BLOCK DIAGRAM (16-BIT READ/WRITE MODE)



Note 1: When enable bit, T1OSCEN, is cleared, the inverter and feedback resistor are turned off to eliminate power drain.

Timer1 Control Register: T1CON

- Time1 Clock Soucre:
  - Internal Clock(FOSC/4)
  - Default frequency: 1MHz/4
- T1CKPS1:T1CKPS0
  - Timer1 Input Clock Prescale Select bits
- TMR1CS:
  - Time1 Clock Source Select bit
- TMR1ON
  - Timer1 On bit

:R 12-1: T1CON: 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

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

Registers Associated with TIMER1

- TMR1L,TMR1H: 目前Timer數到多少的暫存器, FFFFh to 0000h發生overflow時即產生中斷。
- PIR1: (TMR1IF) TMR1 Overflow Interrupt Flag bit
- PIE1: (TMR1IE) TMR1 Overflow Interrupt Enable bit
- IPR1: (TMR1IP) TMR1 Overflow Interrupt Priority bit

TABLE 12-2: REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	52
PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	52
IPR1	PSPIP(1)	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	52
TMR1L	Timer1 Register, Low Byte								50
TMR1H	Timer1 Register, High Byte								50
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	50

Try it

- 開啟Timer1 ON, TMR1IE, TMR1IP並設立中斷點在interrupt,觀察中斷時是否TMR1IF有立起來

- 嘗試若關掉TMR1IE後1.是否會進中斷2.是否flag TMR1IF有立起來

```
int tick_count=0x0;
void main(void) {
    T1CON = 0x01;           //Enable timer1
    PIE1bits.TMR1IE = 1;    //Timer 1 interrupt enable
    RCONbits.IPEN=0x01;     // Enable Interrupt priority
    IPR1bits.TMR1IP=0x01;   // TMR1 high priority ,TMR1 Overflow Interrupt Priority bit
    INTCONbits.GIE = 1;
    PIR1bits.TMR1IF = 0;

    while(1);
}
```

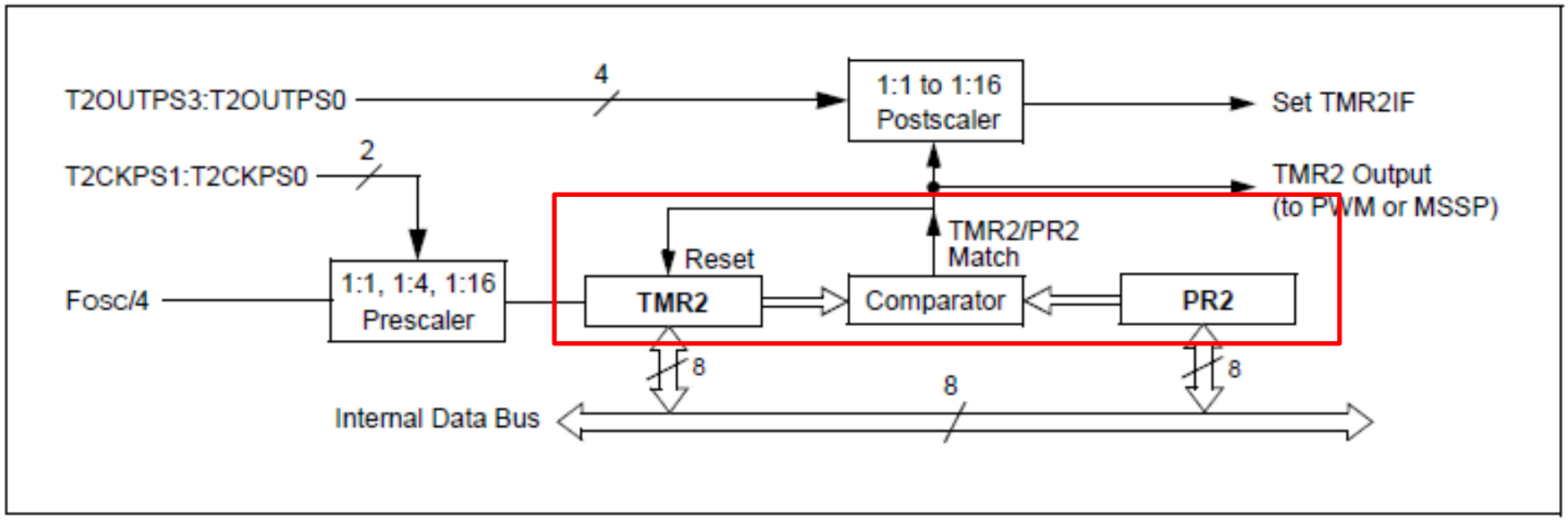
Introduction to Timer
Timer1

Timer2

Timer2 Block diagram

- Time2 Clock Soucre:
  - Internal Clock(FOSC/4)
  - Default frequency: 1MHz/4

FIGURE 13-1: TIMER2 BLOCK DIAGRAM



Timer2 Control Register: T2CON

TER 13-1: T2CON: TIMER2 CONTROL REGISTER

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

bit 7 **Unimplemented:** Read as '0'

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

Registers Associated with TIMER2

- PR2: Timer2 Period Register

TABLE 13-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER

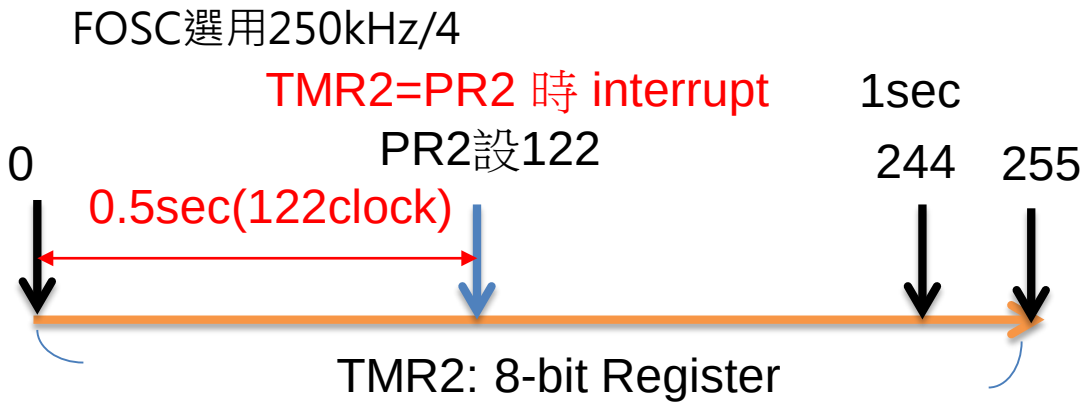
Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	52
PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	52
IPR1	PSPIP(1)	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	52
TMR2	Timer2 Register								50
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	50
PR2	Timer2 Period Register								50

Legend: — = unimplemented, read as '0'. Shaded cells are not used by the Timer2 module.

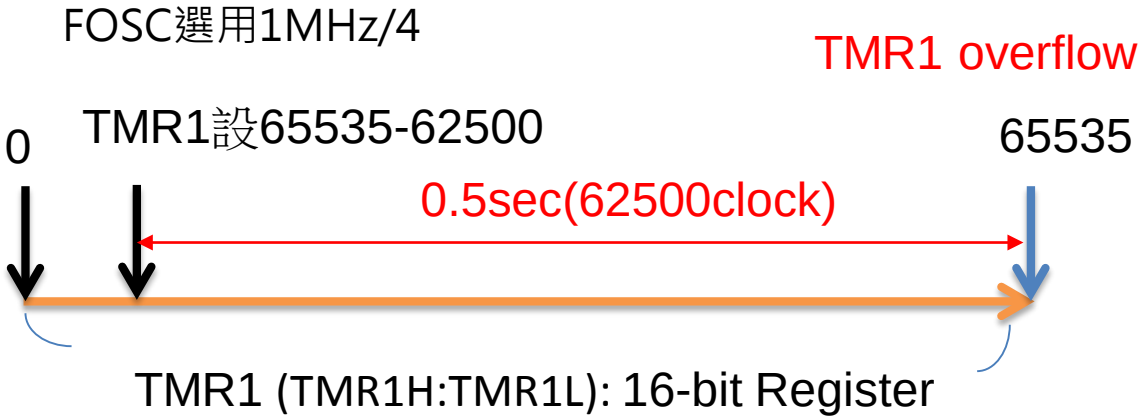
Note 1: These bits are unimplemented on 28-pin devices; always maintain these bits clear.

How to get 0.5 second

1. Timer2 :postscaler和prescaler都選1:16 (In Internal Clock 1秒=62500/16/16=244)



2. Timer1 :除頻器若選用1:2 (In Internal Clock 1秒=250000/2=125000)



MPLAB X IDE - Stopwatch

- You can open the Stopwatch from *Window>Debugging>Stopwatch*
- Stopwath cycle count 只能用simulator來看

```
23
    for(int i=0;i<255;i++){
25          for(int j=0;j<100;j++);
26  }
27
28
    while(1);
```

```
Target halted. Stopwatch cycle count = 14 (14 µs)
Target halted. Stopwatch cycle count = 363642 (363.642 ms)
```

Reference

- PIC18F4520 datasheet
  - http://ww1.microchip.com/downloads/en/devicedoc/39631a.pdf
- Microchip 教材 102ASP Example code
  - http://www.microchip.com.tw/Data_CD/Workshop/8-Bits/102ASP%20PIC18F452.zip
- Microchip 2016 Winter Elite 教材
  - http://www.microchip.com.tw/Data_CD/Workshop/Elite/2016%20Winter%20Elite.zip
- Microchip W401 V3 教材
  - http://www.microchip.com.tw/Data_CD/Workshop/8-Bits/W401%20V3.zip