dsPIC30F Peripheral Module

General Purpose Timers

16 & 32-bit Timers

Timers / Counters Overview

- Five 16-bit General Purpose Timers / Counters
  - Similar functionality between all 5 timers
- Period Registers for each
  - Interrupt generation on match
  - Reset Timer Counter on match
- Gated Timer operation on each
  - Interrupt on falling edge of gate
- Prescale Selectable
  - 1, 8, 64, 256
- Four of these timers can be combined to form two 32-bit timers/counters

TimerX Control Register

- Timer function enabled with TON
- Timer can select internal or external clocks

  TCS=0 : Internal; TCS=1 : External
- Timer can pre-scale input clock by 1,8,64,256

  TCKPS<1:0> - Timer pre-scale settings
- Timer can count internal clocks based on an external gate signal

  TGATE=1 : Gated time accumulation mode
- TSYNC bit is for Timer1 (Type A) only
  - TSYNC = 1 : Synchronize external clock input
- T32 bit is for Timer2 & Timer4 (Type B)
- Both TSYNC & T32 aren't available on Timer3 & Timer5 (Type C)

TxCON Register

TON	-	TSIDL	-	-	-	-	-
bit15	14	13	12	11	10	9	bit8

-	TGATE	TCKPS<1:0>		T32	TSYNC	TCS	-
bit7	6	5	4	3	2	1	bit0

TimerX External Clock

- Timers differ on handling of external clock
  - TIMER1 is asynchronous
    - Input clock frequency < 25MHz
  - TIMER2 and TIMER4 synchronize the clock after the prescaler
    - Input clock frequency < (prescale * 1/2 Fcy)
  - TIMER3 and TIMER5 synchronize the clock at the input of the timer
    - Input clock frequency < 1/2 Fcy

Timer1 Asynchronous Operation

- Timer1 count can sync with external clock or not
- Timer1 has the external gate control for gated Tcy
- Timer1 can count while the device sleeps (async mode)
- Timer1 + LP Oscillator = Real-Time Clock
  - Low-power operation
  - External 32KHz crystal oscillator
  - LP Oscillator can serve as system clock

Timer1 Relate Register

T1CON Register (Type A Time Base)

TON	-	TSIDL	-	-	-	-	-
bit15	14	13	12	11	10	9	bit8
-	TGATE	TCKPS<1:0>		-	TSYNC	TCS	-
bit7	6	5	4	3	2	1	bit0

IFS0 Register

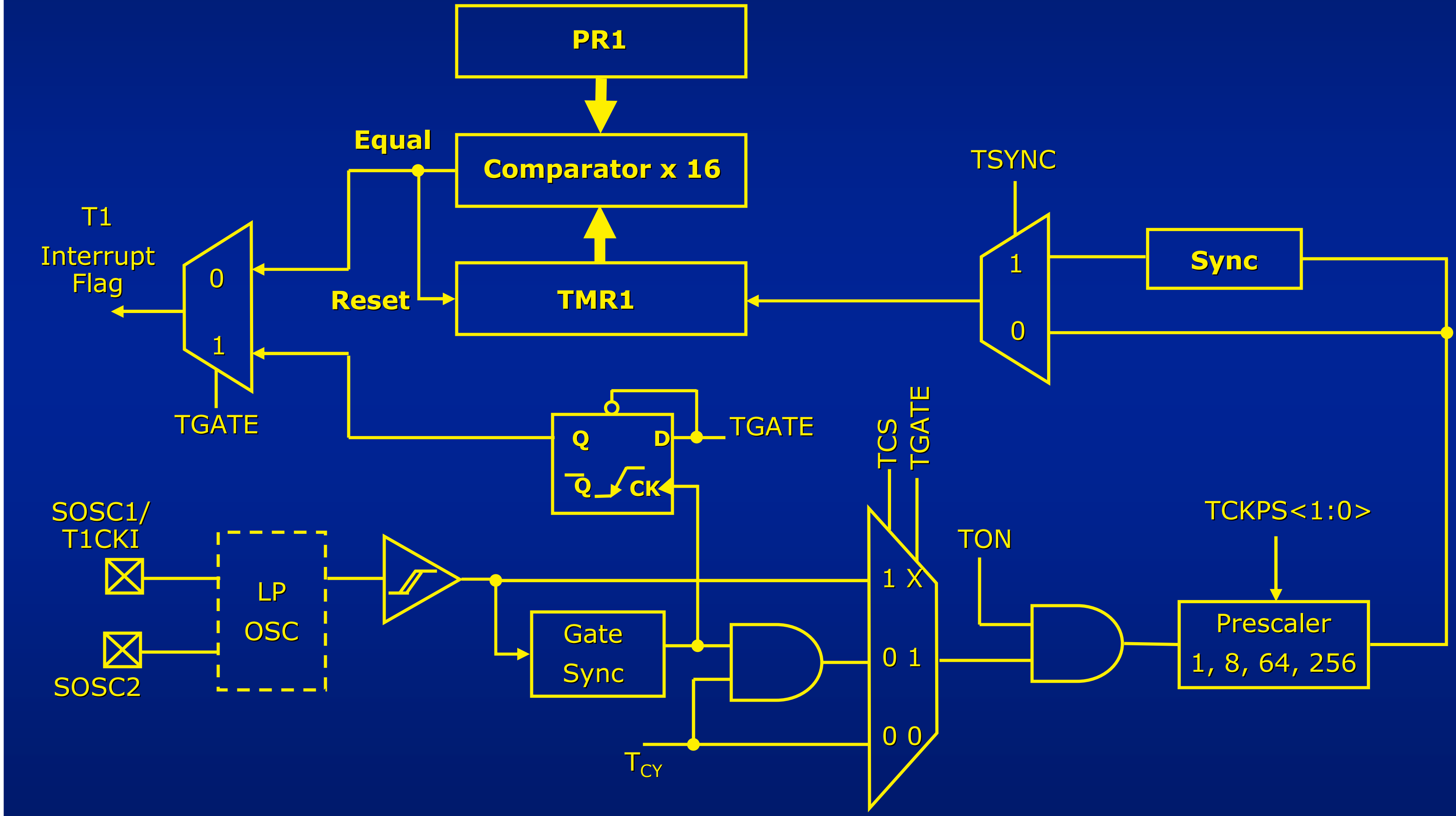
CNIF	BCLIF	I2CIF	NVMIF	ADIF	U1TXIF	U1RXIF	SPIIF
bit15	14	13	12	11	10	9	bit8
T3IF	T2IF	OC2IF	IC2IF	T1IF	OC1IF	IC1IF	INT0IF
bit7	6	5	4	3	2	1	bit0

ICE0 Register

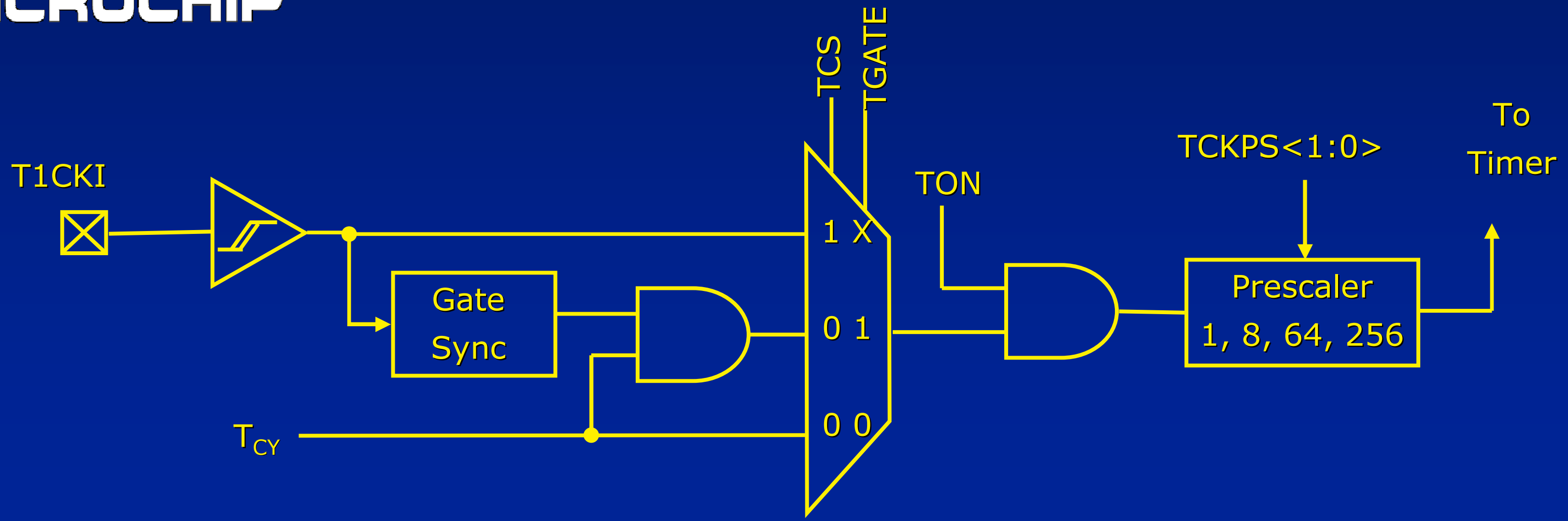
CNIE	BCLIE	I2CIE	NVMIE	ADIE	U1TXIE	U1RXIE	SPIIE
bit15	14	13	12	11	10	9	bit8
T3IE	T2IE	OC2IE	IC2IE	T1IE	OC1IE	IC1IE	INT0IE
bit7	6	5	4	3	2	1	bit0

IPC0 Register : TIP<2:0> for the interrupt priority control of Timer1 (Default are 4)
TMR1 Register : Timer1 count register (Reset at 0x0000 , count up)
PR1 Register : Timer1 period register

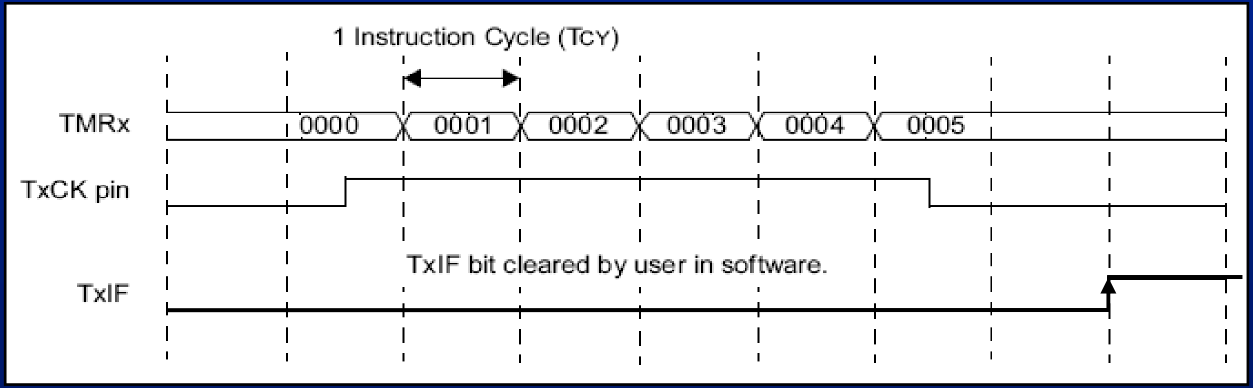
Timer 1 方塊圖
(Type A)



閘控時間的累計

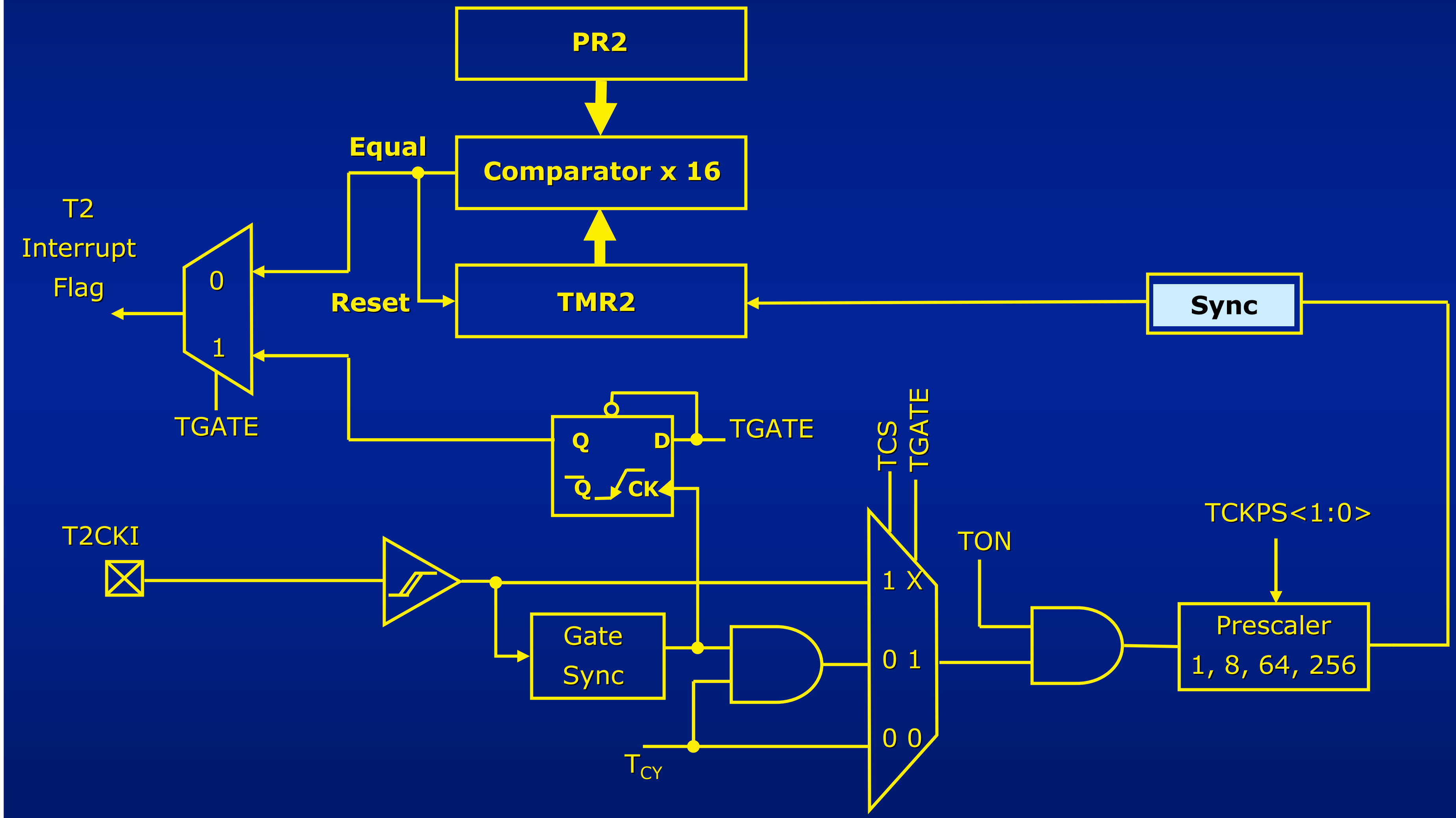


- T1CKI 的輸入在 Hi 的週期裡 Timer1 會以 Tcy 的頻率來計數， 當 T1CKI 由 Hi 變 Low 時會產生中斷；T1CKI 為 Low 時停止計數。

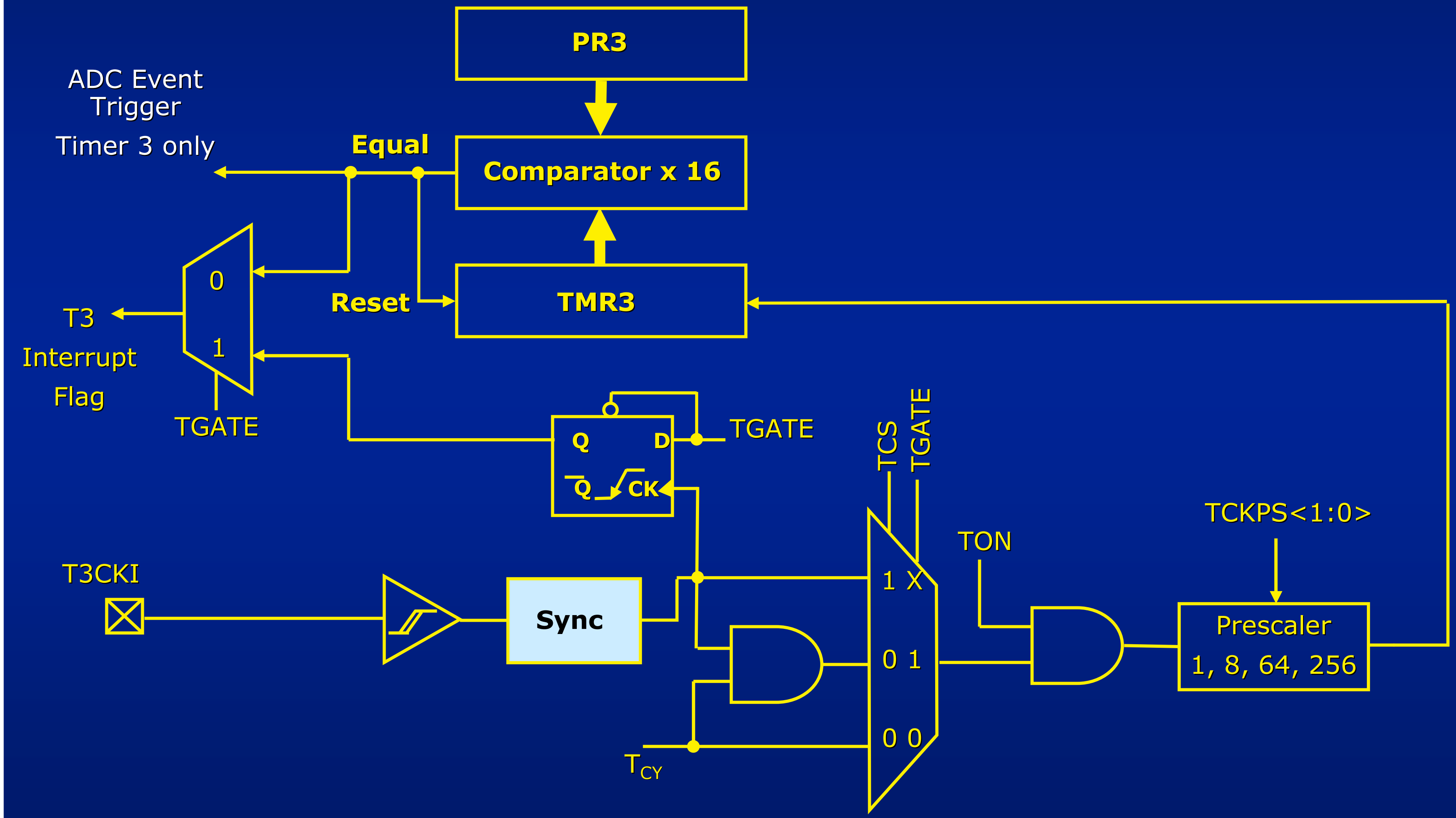


1 Instruction Cycle (TCY)
TMRx
0000
0001
0002
0003
0004
0005
TxCK pin
TxIF bit cleared by user in software.
TxIF

Timer 2/Timer 4 方塊圖
(Type B)



Timer 3/Timer 5 方塊圖
(Type C)



Concatenating Timers 32-bit Operation

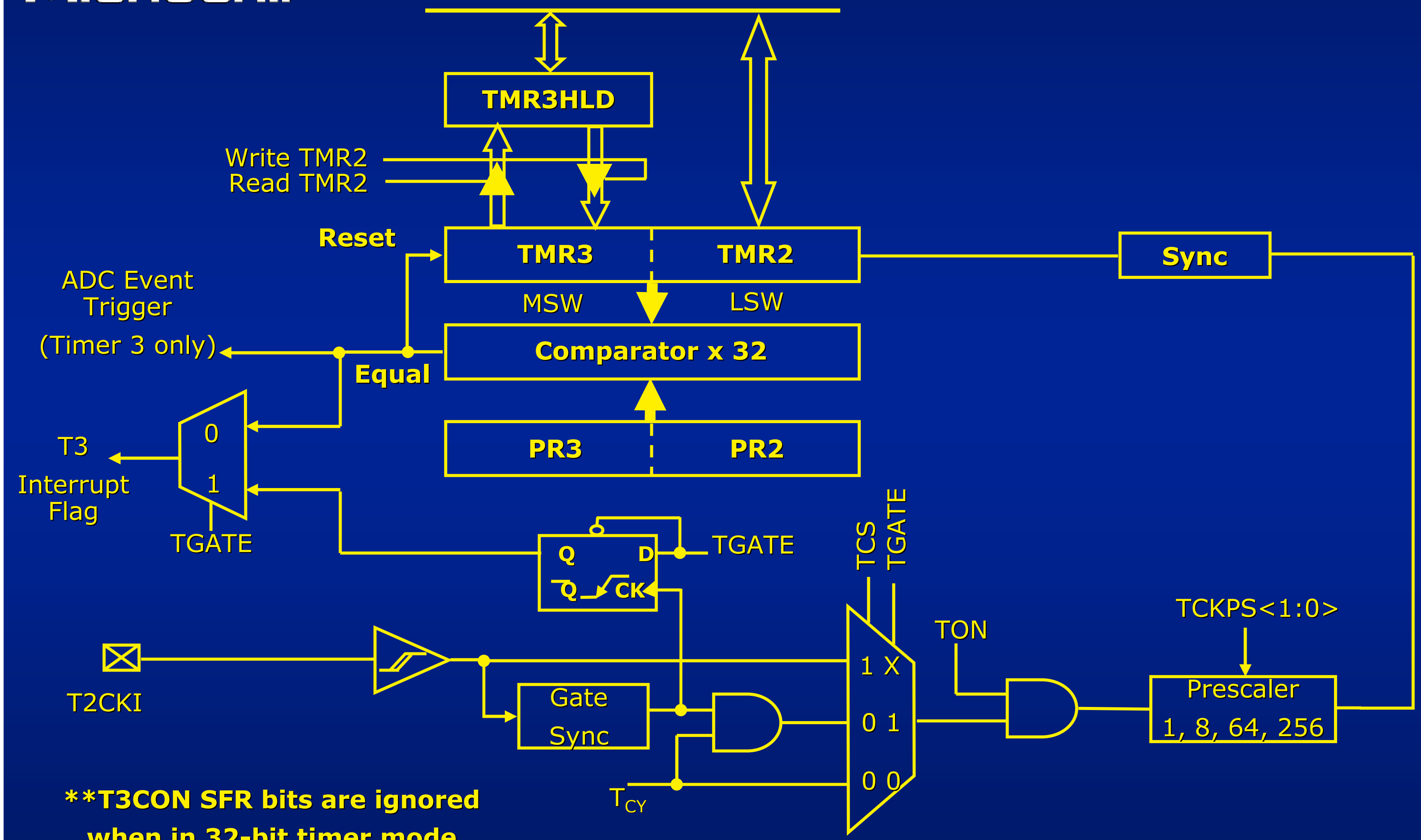
- Timer3/Timer2 pair into 32-bit Timer
- Timer5/Timer4 pair into 32-bit Timer
- 16-bit writes buffered to allow 32-bit counter updates
- T32=1 : 32-bit Timer select
- Concatenated timer has same functionality as Timer2

T2CON Register

TON	-	TSIDL	-	-	-	-	-
bit15	14	13	12	11	10	9	bit8

-	TGATE	TCKPS<1:0>		T32	-	TCS	-
bit7	6	5	4	3	2	1	bit0

32-bit Timer Block Diagram



****T3CON SFR bits are ignored when in 32-bit timer mode**

32-bit Timers Operation

- Timer config. bit, T32(T2CON<3>) must be set to “1” for a 32-bit timer/counter operation.
- As 32-bit mode, all control bits are respective to the T2CON (Timer2 & 3) or T4CON (Timer4 & 5)
- 32-bit mode need to be synchronized between LS and MSWord of the 32-bit timer
  - Assume TMR3:TMR2 is a 32-bit timer pair.
  - Read the LSWord (TMR2) will automatically transfer the TMR3 into the TMR3HLD
  - Write the MSWord of 32-bir value to the TMR3HLD firstly then write the LSWord to the TMR2

Timer 其它注意事項

- Only Timer1 can work in the SLEEP mode with asynchronize setting
- All Timers can work with IDEL mode when the TSIDL bit was cleared.
- Input Capture & Output Compare can select one of two timers (TMR2 or TMR3) as their time base
- Type C timer can generated the AD conversion trigger signal on a period match in 16 or 32-bit mode
  - T32 bit = 0, Timer3 compare ends sampling and start conversion
  - T32 bit = 1, Timer2 and Timer3 AD event trigger signal
- Timer can assign for the Interrupt input pin with rising edge
  - Prescaler 1:1
  - Timerx = PRx – 0x01 (PRX does not equal “0”)

有關 16-bit Peripheral Libraries

- Peripheral Library 的檔案路徑：
  - C:\Program Files\Microchip\MPLAB C30\lib\dsPIC30F\
    - libp30F4011-coff.a
- Microchip 所提供的週邊原始資料庫程式碼：
  - C:\Program Files\Microchip\MPLAB C30\src\peripheral_30F_24H_33F\src\pmc
- 周邊資料庫一些相關 header file
  - …\MPLAB C30\support\peripheral_30F_24H_33F
    - timer.h , ADC10.h , uart.h ,…
- Linker script file “p30f4011.gld” 會自動加入
  - …\MPLAB C30\support\dsPIC30F\gld

C Libraries of Timer

- OpenTimer1 (*unsigned int config. , unsigned int period*)
  - 這函數設定 Timer1 模組 (16-bit)
  - 相關的定義參考 “ timer.h” 或 C:\Program Files\Microchip\MPLAB C30\docs\periph_lib\dsPIC30F_dsPIC33F_PIC24H_Timers_Help.htm
  - 不同的 Timer 使用不同的定義名稱，參考 timer.h
  - 詳細資訊請參考 : C:\Program Files\Microchip\MPLAB C30\docs\ 16-Bit_Language_Tools_Libraries_51456E.pdf

  Example:

  ```
  OpenTimer1    ( T1_ON & T1_GATE_OFF & T1_PS_1_256 &
                  T1_SYNC_EXT_OFF & T1_SOURCE_INT ,
                  32768 );
  ```

- ConfigIntTimer1 (*unsigned int config.*)
  - 此函數設定中斷功能及優先權

  Example:

  ```
  ConfigIntTimer1 ( T1_INT_PERIO_6 & T1_INT_ON );
  ```

其他常用的 Timer 函數庫

- WriteTimer1 (*unsigned int timer*)
  - writes the contents of the 16-bit Timer Register
- ReadTimer1 (*unsigned int timer*)
  - Reads the contents of the 16-bit Timer Register
- CloseTimer1 (*void*)
  - Turns off the Timer1
- Other 32-bit Timer Functions
  - OpenTimer23 (*unsigned int config. , unsigned int period*)
  - ConfigIntTimer23 (*unsigned int config.*)
  - WriteTimer23 (*unsigned long timer*)
  - ReadTimer23 (*unsigned long timer*)
  - CloseTimer23 (*void*)

有用的Timer1 巨集定義 (marco)

- EnableIntT1 : 開啓 Timer1 的中斷
- DisableIntT1: 關閉 Timer1 的中斷
- SetPriorityIntT1: 設定或變更 Timer1 中斷優先權
  - 使用語法: SetPriorityIntT1 (7) ;

Timer1 實驗 LAB1

- 使用 Timer1 (10mS) 當做按鍵掃描的基準及處理按鍵的彈跳問題
  - 掃描兩個數位按鍵 (SW5 & SW6)
  - 掃描四個 ADC 類比按鍵 (SW1 ~ SW4)
    - 類比按鍵的 AD 轉換來自 Timer3 間隔每 3mS
  - 按鍵掃描中斷時間 10mS
  - 先抓按鍵編號再轉成 ASCII code
  - 在 LCD 上顯示各按鍵的次數
- 更多有關 Timer 應用的想法 …
  - How to do the more event in the background
  - Look the Timer Interrupt provided the powerful function
  - If I want to doing more real time TASK …

Timer1 中斷時間的算法

```
void Init_Timer1(void)
{
        ConfigIntTimer1( T1_INT_PRIOR_7 & T1_INT_ON ) ;    // Timer1 的中斷優先等級設 7 (最高)
                                                           // Timer1 的中斷 ON

        OpenTimer1 ( T1_ON & T1_IDLE_STOP & T1_GATE_OFF &       // Timer1 Period as 10mS
                        T1_PS_1_1 & T1_SYNC_EXT_OFF & T1_SOURCE_INT ,
                         (long) (( Fosc/4 / 1000) * 10) /1) ;
```

時間計算方式:

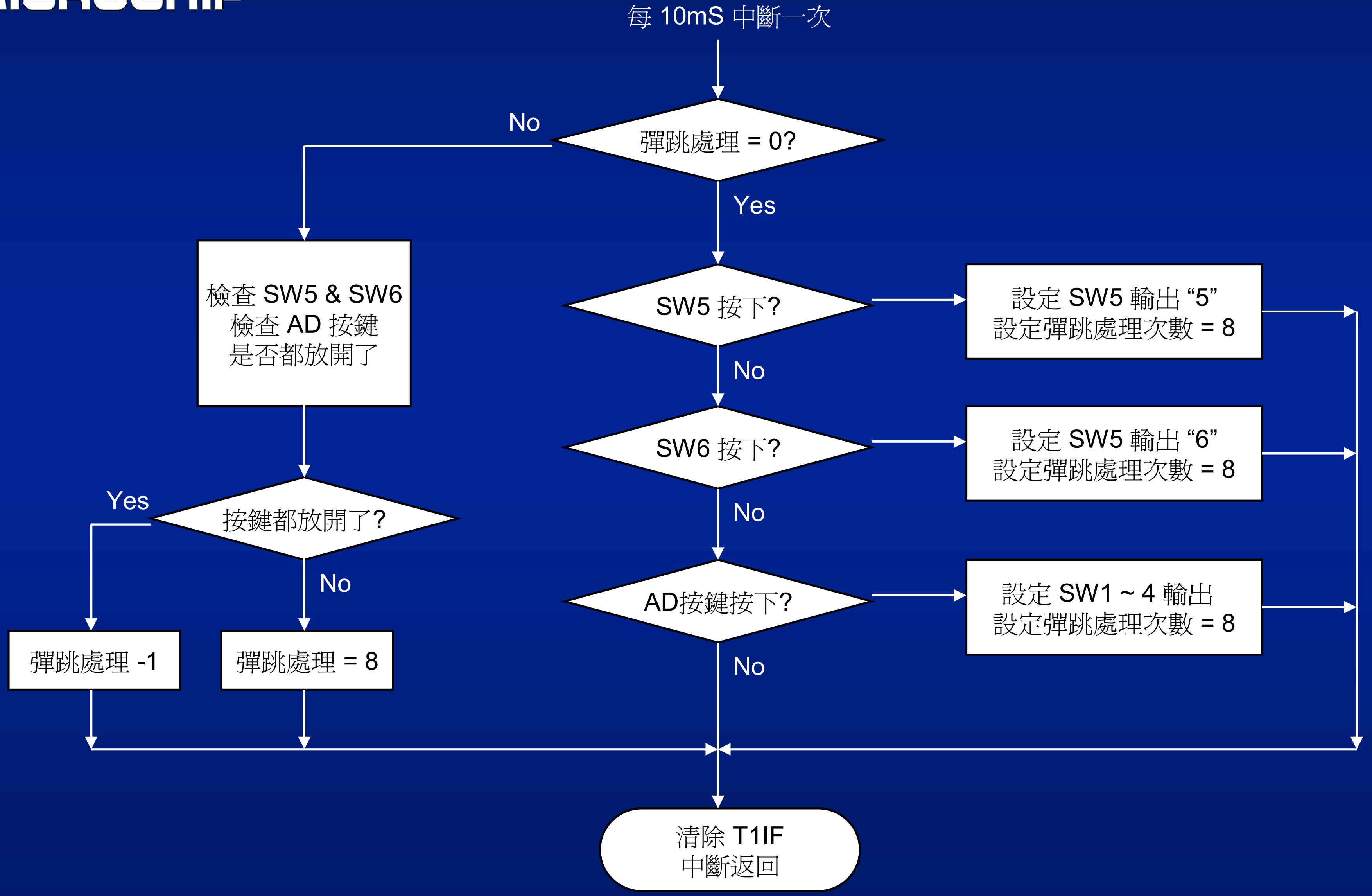
- Fosc = 7372800Hz x 16 = 117964800 Hz (CPU 執行頻率)。
- Fosc/4 爲取得進入Timer1 的 Fcy 頻率。
- Fcy 再除以 1000 後即可將 MHz (uS) 變成 KHz (mS) 。
- 最後面的 /1 爲預除器的比率 (T1_PS_1_1)，如需較長的延遲將此預除器變大即可。前面所計算出的數值基本上已經是 1mS 的計時的值 (Period)，如果要延遲 10mS 的話只要將此 1mS 的 Period 的值再乘以 10 即可。
- 如果乘以 250 呢?

AD 類比按鍵偵測

```
ADCValue = ADCBUF0 >> 6 ;    // ADC 轉換結果除 64

if (ADCValue == 0)   (AD_Key_Input = ‘1’) ;   // SW1 被按下，傳回 0x31
else
{
    if(ADCValue <= 8) (AD_Key_Input = ‘2’) ; // SW2 被按下，傳回 0x32
    else
    {    if(ADCValue <= 10) (AD_Key_Input = ‘3’) ; // SW3按下，傳回 0x32
         else
         {  if(ADCValue <= 12) (AD_Key_Input = '4') ;
            // SW4 被按下，傳回 0x34 (ASCII Code ‘4’)
            else (AD_Key_Input = 0x00) ;
            //沒有類比按鍵被按下，回傳 0x00 的值
         }
    }
}
```

使用 Timer1 處理按鍵彈跳



每 10mS 中斷一次

彈跳處理 = 0?

No

Yes

檢查 SW5 & SW6
檢查 AD 按鍵
是否都放開了

按鍵都放開了?

Yes

No

彈跳處理 -1

彈跳處理 = 8

SW5 按下?

設定 SW5 輸出 “5”
設定彈跳處理次數 = 8

No

SW6 按下?

設定 SW5 輸出 “6”
設定彈跳處理次數 = 8

No

AD按鍵按下?

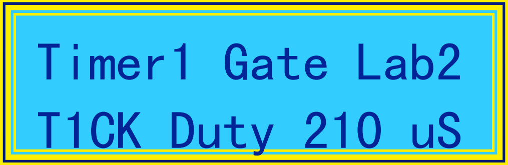
設定 SW1 ~ 4 輸出
設定彈跳處理次數 = 8

No

清除 T1IF
中斷返回

Timer1 閘控實驗 LAB 2

- 使用 Timer1 的閘控功能來量測脈衝的時間
- T1CK/RC14 (pin 16) 爲輸入腳位
- 以指令週期 Fcy 做爲量測的參考時間(頻率)
- RB4 (pin 6) 提供可變週期的脈衝(訊號來自 QEI 產生器的 QEA)
  - ❖ VR3 change the Duty and Frequency
  - ❖ Make sure the switch 1 is ON position of DSW4
- 使用杜邦線跳接 : RC14 (pin16) 接到 RB4 (pin6)
  - ❖ RB4 provide a variable signal from PIC16F684
- 量測輸入的 high duty 的時間並顯示在 LCD 上



```
Timer1 Gate Lab2
T1CK Duty 210 uS
```

Input Signal Duty on LCD Display

閘控設定與換算

- 設定 Timer1 爲閘控時間量測並使用閘控中斷觸發模式：

```
ConfigIntTimer1( T1_INT_PRIOR_7 & T1_INT_ON ) ;
// Timer1 的中斷優先等級設 7 (最高)， Timer1 的中斷 ON

OpenTimer1( T1_ON & T1_IDLE_STOP & T1_GATE_ON &
// Timer1 Period as 10mS設爲閘控量測模式
                  T1_PS_1_1 & T1_SYNC_EXT_OFF & T1_SOURCE_INT , 0xFFFF);
```

- 計算脈衝 Hi Period 的時間:

```
DisableIntT1;       // 暫時停止 Timer1 閘控計時中斷功能
Period = Period / 14.745600 ;       // 1uS 會有 14.7456 (Fcy) 個計數單位
// 將所量到的 Period 値除以 Fcy 後就可以得到脈衝的時間(uS) 。
update_LCD( );      // 將時間轉成 ASCII Code 後以 us 爲單位顯示在 LCD
WriteTimer1(0);     //  Timer1 歸零
EnableIntT1;        //  允許  Timer1 値控計時功能
```