



# **MICROCHIP**

---

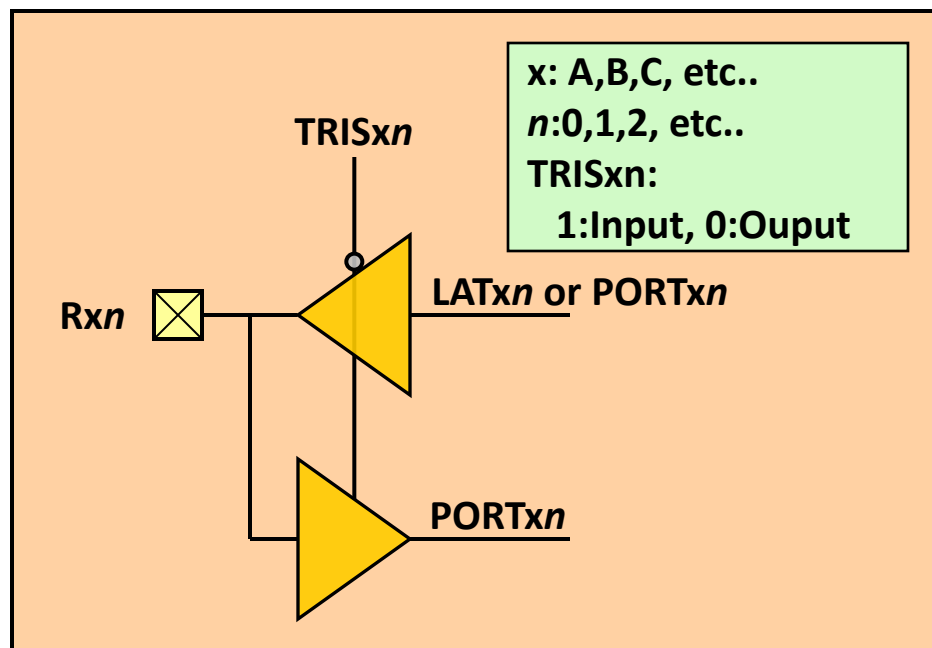
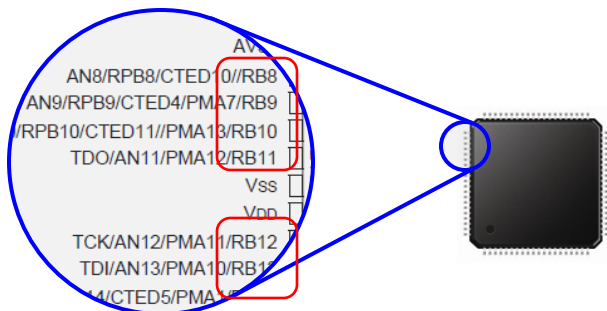
***Regional Training Centers***

## **Section 4**

### **IO Port Architecture**

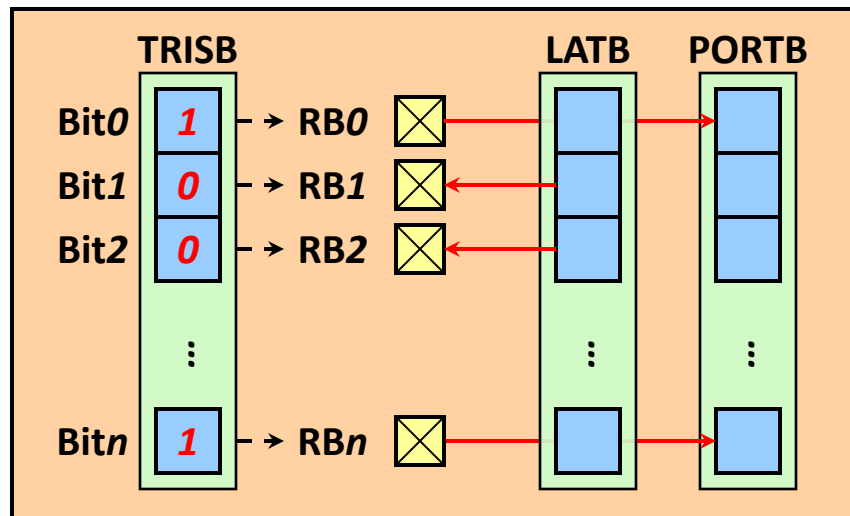
# I/O Port Block Diagram

- 32-Bits MCU的IO Port示意圖, 如圖所示。
- 所有的IO Port都有TRIS, LAT跟PORT特殊功能暫存器。TRIS用來設定IO的方向, 要輸出時(Ex: 控制LED)  $TRIS_{xn}$  必須設為"0"。要輸入(Ex: 讀取按鍵狀態)時  $TRIS_{xn}$  必須設為"1"。
- 設為輸出時, 要輸出的狀態填入  $LAT_{xn}$  或  $PORT_{xn}$ , 對應的接腳  $R_{xn}$  就會有輸出。
- 設為輸入時, 可以讀取  $PORT_{xn}$  取得外部的實際狀態。



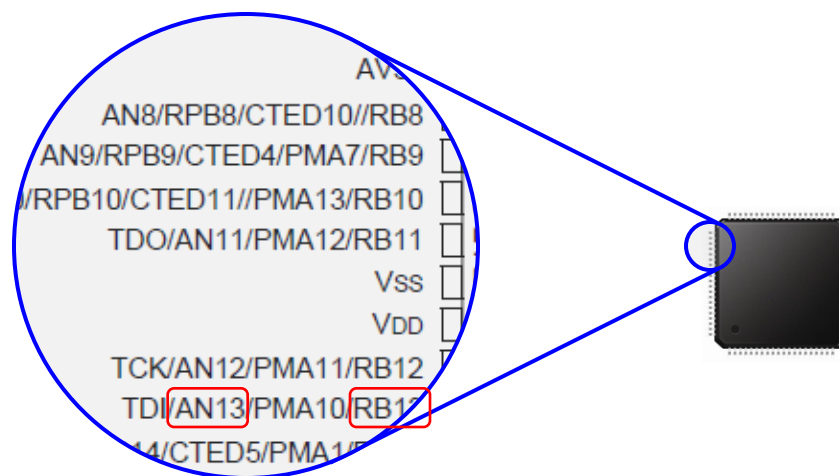
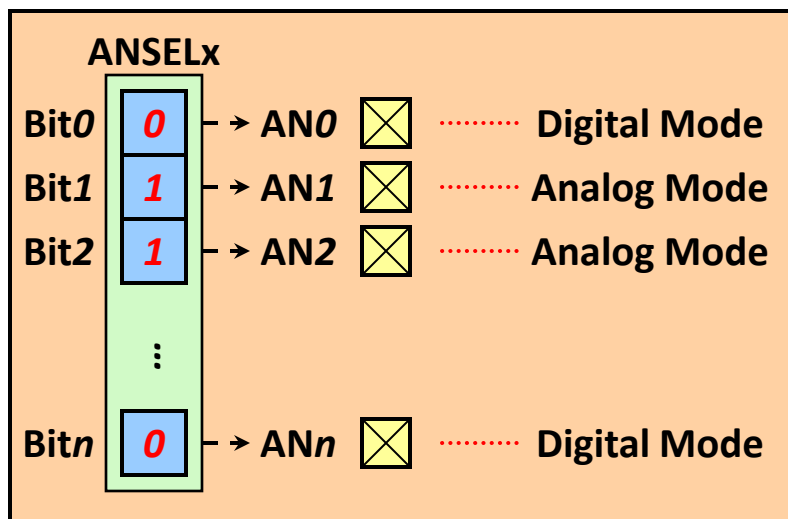
# I/O Port Manipulation

- TRISx, LATx跟PORTx特殊功能暫存器都是一個32-Bits的暫存器, 暫存器中每個Bit都控制著對應的I/O接腳。
- 以IO Port B為例,TRISB, LATB, PORTB的Bit 13,控制RB13接腳。
- 要用RB13控制LED時, 必須先把TRISB的Bit 13設為0(輸出), 然後把要輸出的狀態,填入LATB的Bit 13。
- 要用RB13讀取按鍵狀態時,則必須把TRISB的Bit 13設為1(輸入), 然後就可以從PORTB的Bit 13取得外部的狀態。



# Analog or Digital Mode

- 有些 IO接腳(Rxn)的功能跟類比輸入(ANn)共用的。但MCU在 Reset/Power On後的預設值為Analog Mode, 無法做為數位輸出入使用。所以如果碰到這類接腳, 在使用時必須先取消Analog Mode改為Digital Mode, IO才能正常動作。
- PIC32MX450F2456H透過ANSELx來設定。設0為Digital Mode, 1為Analog Mode。 Ex: `ANSELBbits.ANSB13 = 0; //設定RB13為 Digital Mode`

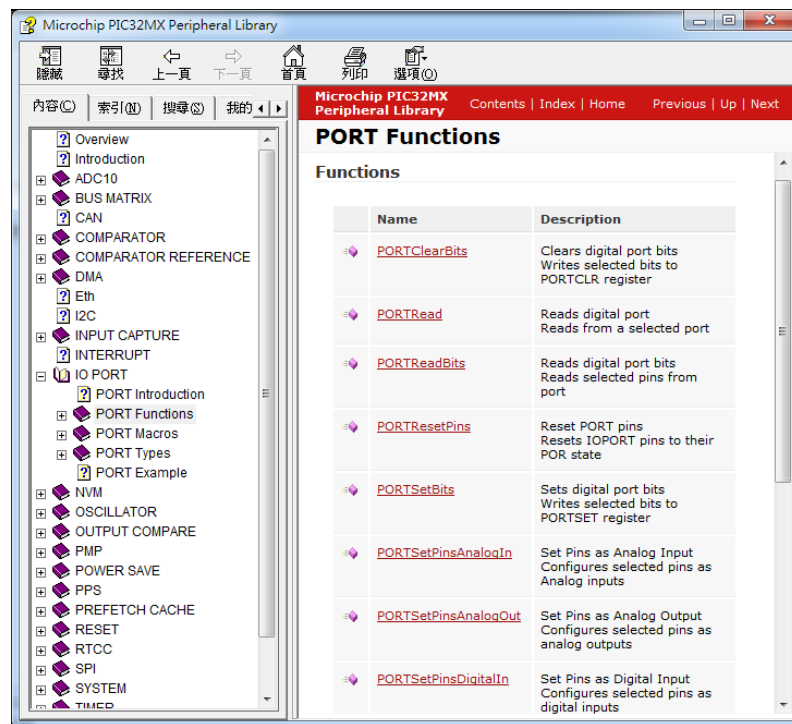


# XC32 I/O Port Function & Macro

- MPLAB XC32提供許多IO Function可供使用:
- |                             |                                       |
|-----------------------------|---------------------------------------|
| mPORTxSetPinsDigitalIn( );  | // 設定對應的I/O Portx Bit為數位輸入            |
| mPORTxSetPinsDigitalOut( ); | // 設定對應的I/O Portx Bit為數位輸出            |
| mPORTxSetPinsAnalogIn( );   | // 設定對應的I/O Portx Bit為類比輸入            |
| mPORTxSetPinsAnalogOut( );  | // 設定對應的I/O Portx Bit為類比輸出            |
| mPORTxSetBits( );           | // 設定I/O為High, Portx Bit = 1          |
| mPORTxClearBits( );         | // 清除I/O為Low, Portx Bit = 0           |
| mPORTxToggleBits( );        | // 反向I/O Portx Bit("1"->"0","0"->"1") |
| PORTRead( , );              | // 讀取I/O Port的資料                      |
| PORTReadBits( , );          | // 讀取I/O Port Bit的資料                  |
| ...                         |                                       |

# XC32 I/O Code Example

- I/O設定為數位輸出, 輸出操作範例  
`mPORTBSetPinsDigitalOut( BIT_12 | BIT_13 );`  
`mPORTBSetBits( BIT_12 );`  
`mPORTBClearBits( BIT_13 );`  
`mPORTBToggleBits( BIT_12 | BIT_13 );`
- I/O設定為數位輸入範例  
`mPORTDSetPinsDigitalIn(BIT_7 | BIT_6);`



- 詳細說明請參閱說明文件

Microchip PIC32MX Peripheral Library.chm。

C:\Program Files (x86)\Microchip\xc32\v1.30\docs\pic32-lib-help\

# Lab2 - BasicIO

- 在Lab1的程式基礎上,嘗試加入IO Port的控制程式。
- 請嘗試控制TRIS $x_n$ , LAT $x_n$ , PORT $x_n$ , 或使用XC32的I/O函數, 來讓RB13接腳可以不斷的轉態(Toggle, 1 $\leftrightarrow$ 0)。時間間隔為500 mS。
- 前面所提過的, 要正確的存取SFRs必須 include MCU的標頭檔  
Ex:#include <p32xxxx.h>。要使用XC32的函數庫時, 必須必須 include 函數庫的標頭檔 Ex:#include <plib.h>。
- 操作SFRs時,
  - 可以使用暫存器, 結構來存取  
Ex:TRISB &= ~0x2000; // Port B Bit13 設為輸出,  
LATB |= 0x2000; // Port B Bit13 輸出High Level。  
Ex:TRISBbits.TRISB13 = 0;  
LATBbits.LATB13= 1;
  - 或透過XC32的I/O函數  
Ex:mPORTBSetPinsDigitalOut(BIT\_13);  
mPORTBToggleBits(BIT\_13);

# Lab2 – Basic IO Step1

- 程式要如何設計才能達到Toggle的功能？

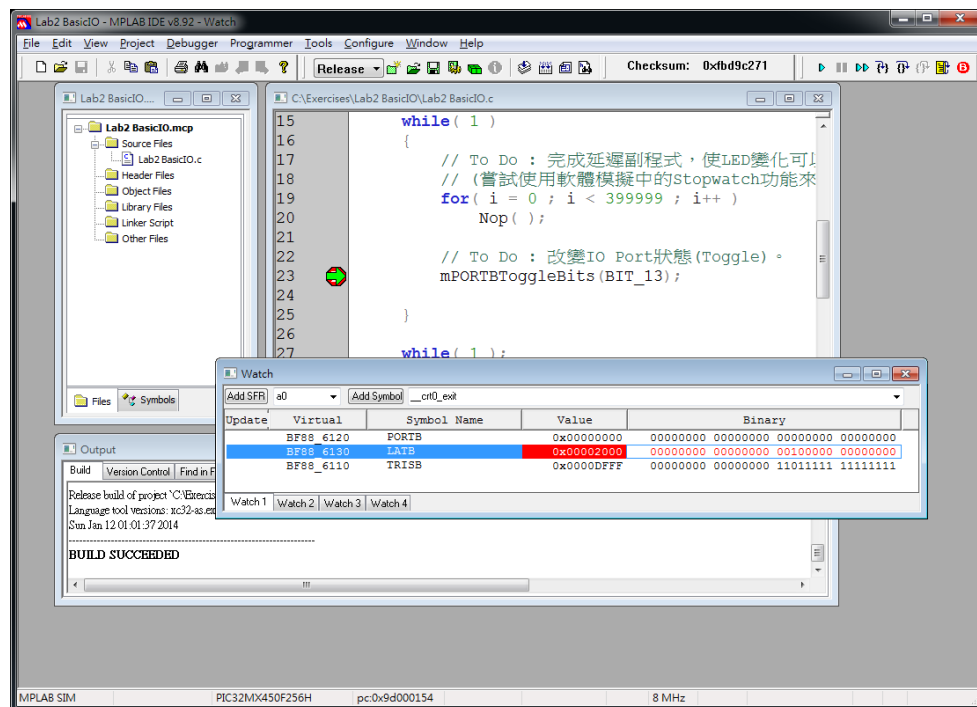
$LATB = LATB \wedge 0x2000;$  // 利用XOR

$LATBbits.LATB13 = !LATBbits.LATB13;$  // 利用NOT

$mPORTBToggleBits(BIT\_13);$

- 如何得知IO的狀態變化？

善用MPLAB SIM與Watch  
Windows觀察程式的變化。

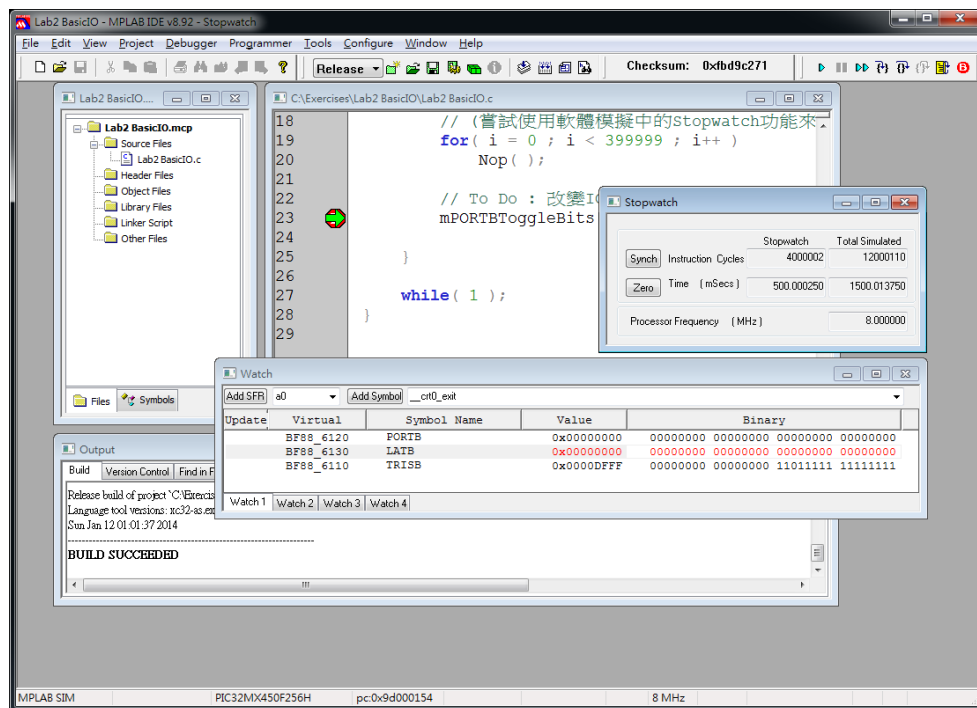




# Lab2 – Basic IO Step2

- 如何控制IO每次Toggle的間隔時間？

控制for迴圈來決定Delay的時間。可以利用Stopwatch來測量迴圈的時間, 計算前必須先設定預計的系統時脈。此處設定 Processor Frequency 為 8MHz。



## Lab2 – Basic IO Step3

- 要達成500mS, Delay的計數值必須不斷變大, 會出現何種問題?

Ex:

```
void main( void )  
{  
    unsigned short i;  
    ...  
    for( i = 0 ; i < 65536 ; i++ )  
        Nop( );  
    ...  
}
```