MICROCHIP

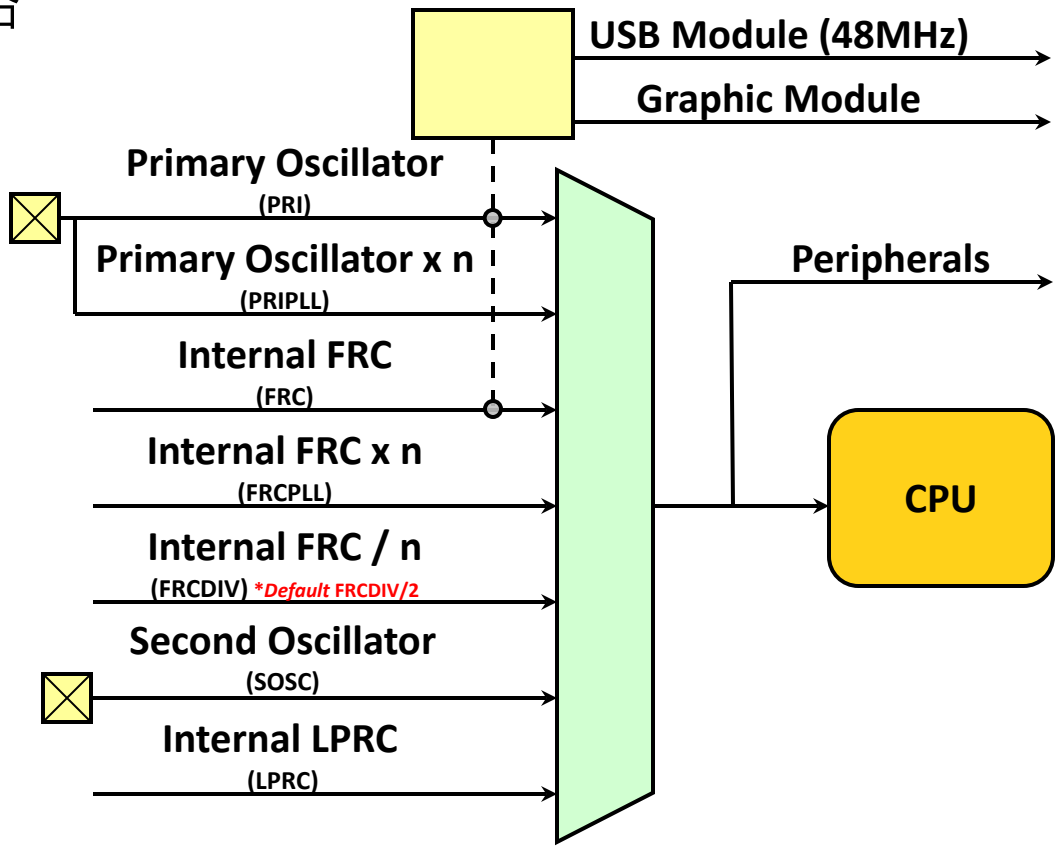
Regional Training Centers

Section 7
Oscillator and Configuration Bits

MICROCHIP
Regional Training Centers

Microchip 16-Bits 時脈架構

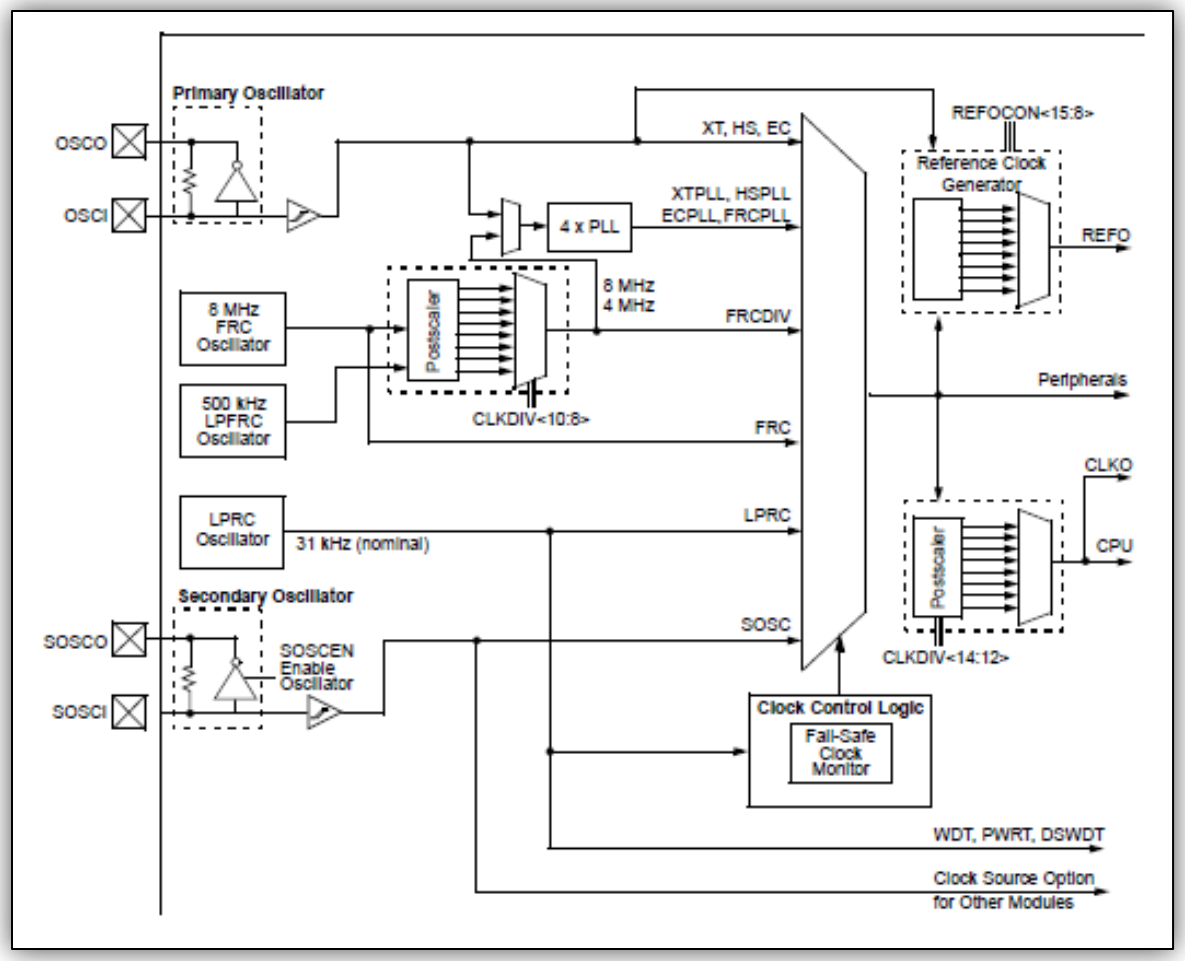
- 回憶下前面的說明, Microchip 16-Bits MCU具有多種時脈來源可選擇。可以使用內部RC或者外部的Crystal, Oscillator。輸入的時脈也可透過內部電路進行倍頻(PLL)或者除頻。
- **時脈來源的設定, 必須正確如果設定錯誤, CPU會以不正確頻率執行, 導致程式運作上的時序不對, 或者完全沒有接收到時脈, 導致CPU無法運作。**
- 特定系列具有USB模組及LCD模組的時脈, 電路結構會更複雜。



MICROCHIP
Regional Training Centers

PIC24F16KA102's Oscillator

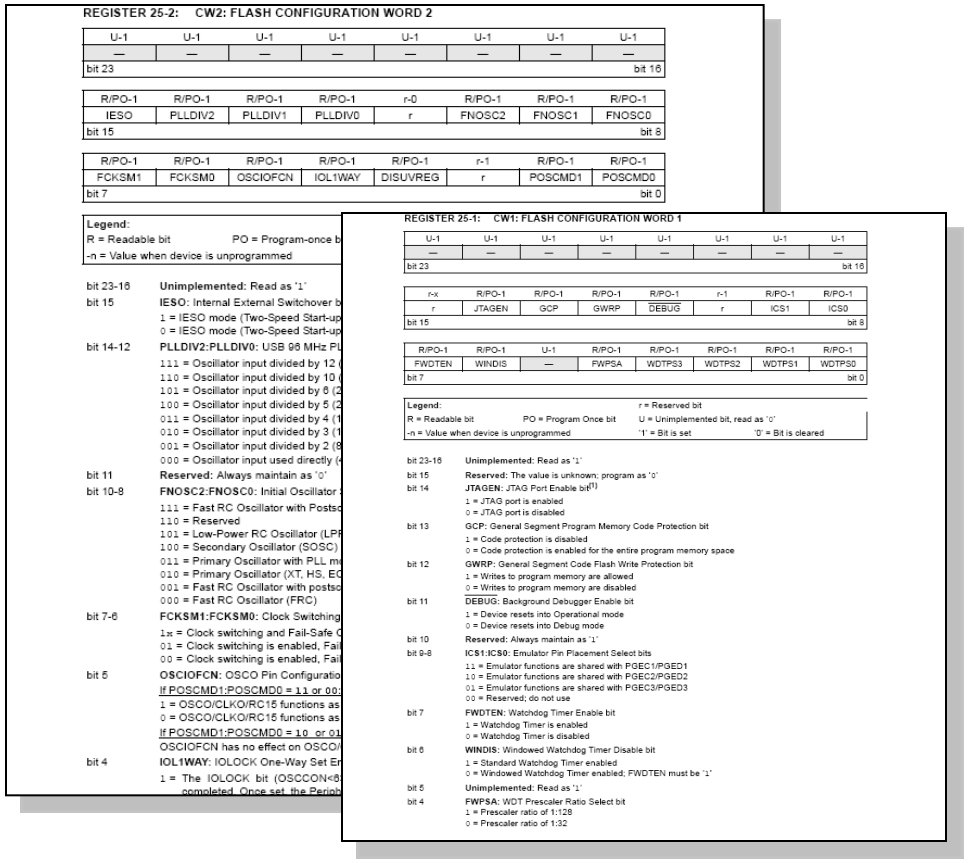
- PIC24F16KA102的Oscillator Block Diagram如圖所示。
- 時脈來源的設定必需透過設定Configuration Bits。



MICROCHIP
Regional Training Centers

Configuration Bits

- Configuration Bits是用來設定MCU中一些重要的模式。其中包含如:CPU的時脈來源, 除錯工具使用的接腳, Watchdog, 程式碼保護, 程式碼防寫等設定。
- **其中又以CPU的時脈來源, 除錯工具使用的接腳設定最重要, 時脈來源設定錯誤則CPU無法如預期運作, 除錯工具使用的接腳設定錯誤, 則無法開啟除錯模式。**
- 參考Datasheet的Special Feature章節可以看到完整的說明。



MICROCHIP
Regional Training Centers

XC16 Configuration Bits的支援

- 不同系列的Configuration Bits設定方法是一樣的, 但使用參數與定義可能不同。必須搭配參照Datasheet及MCU的說明檔(**config_index.html**)來設定。
- 以PICF16KA102為例,要設定為內部FRC要使用以下方法:

  ***#pragma config* FNOSC = FRC**

Oscillator Select

FNOSC = FRC	Fast RC oscillator (FRC)
FNOSC = FRCPLL	Fast RC oscillator with divide-by-N with PLL module (FRCDIV+PLL)
FNOSC = PRI	Primary oscillator (XT, HS, EC)
FNOSC = PRIPLL	Primary oscillator with PLL module (HS+PLL, EC+PLL)
FNOSC = SOSC	Secondary oscillator (SOSC)
FNOSC = LPRC	Low-Power RC oscillator (LPRC)
FNOSC = LPFRC	500 kHz Low-Power FRC oscillator with divide-by-N (LPFRCDIV)
FNOSC = FRCDIV	8 MHz FRC oscillator with divide-by-N (FRCDIV)

REGISTER 26-2: FGS: GENERAL SEGMENT CONFIGURATION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/C-1	R/C-1
—	—	—	—	—	—	GSS0	GWRP
bit 7							bit 0

Legend:
R = Readable bit; C = Clearable bit; U = Unimplemented bit, read as '0'
-n = Value at POR; '1' = Bit is set; '0' = Bit is cleared; x = Bit is unknown

bit 7-2 **Unimplemented:** Read as '0'
bit 1 **GSS0:** General Segment Code Flash Code Protection bit
1 = No protection
0 = Standard security is enabled
bit 0 **GWRP:** General Segment Code Flash Write Protection bit
1 = General segment may be written
0 = General segment is write-protected

REGISTER 26-3: FOSCSEL: OSCILLATOR SELECTION CONFIGURATION REGISTER

R/P-1	U-0	U-0	U-0	U-0	R/P-1	R/P-1	R/P-1
IESO	—	—	—	—	FNOSC2	FNOSC1	FNOSC0
bit 7							bit 0

Legend:
R = Readable bit; P = Programmable bit; U = Unimplemented bit, read as '0'
-n = Value at POR; '1' = Bit is set; '0' = Bit is cleared; x = Bit is unknown

bit 7 **IESO:** Internal External Switchover bit
1 = Internal External Switchover mode is enabled (Two-Speed Start-up is enabled)
0 = Internal External Switchover mode is disabled (Two-Speed Start-up is disabled)
bit 6-3 **Unimplemented:** Read as '0'
bit 2-0 **FNOSC<2:0>:** Oscillator Selection bits
000 = Fast RC Oscillator (FRC)
001 = Fast RC Oscillator with divide-by-N with PLL module (FRCDIV+PLL)
010 = Primary Oscillator (XT, HS, EC)
011 = Primary Oscillator with PLL module (HS+PLL, EC+PLL)
100 = Secondary Oscillator (SOSC)
101 = Low-Power RC Oscillator (LPRC)
110 = 500 kHz Low-Power FRC Oscillator with divide-by-N (LPFRCDIV)
111 = 8 MHz FRC Oscillator with divide-by-N (FRCDIV)

XC16 Configuration Bits的支援

- 要設定除錯介面，則使用以下方法設定:

  ***#pragma config* ICS = PGx3**

Alternate I2C1 Pin Mapping bit

I2C1SEL = SEC	Alternate location for SCL1/SDA1 pins
I2C1SEL = PRI	Default location for SCL1/SDA1 pins

Brown-out Reset Voltage bits

BORV = LPBOR	Low-power Brown-Out Reset occurs around 2.0V
BORV = V27	Brown-out Reset set to Highest Voltage (2.7V)
BORV = V20	Brown-out Reset set at 2.0V
BORV = V18	Brown-out Reset set to lowest voltage (1.8V)

MCLR Pin Enable bit

MCLRE = OFF	RA5 input pin enabled; MCLR disabled
MCLRE = ON	MCLR pin enabled; RA5 input pin disabled

ICD Pin Placement Select bits

ICS = PGx3	PGC3/PGD3 are used for programming and debugging the device
ICS = PGx2	PGC2/PGD2 are used for programming and debugging the device
ICS = PGx1	PGC1/PGD1 are used for programming and debugging the device

Set Clip On Emulation bit

COE = ON	Device will reset into Clip On Emulation Mode
COE = OFF	Device will reset into Operational Mode

Background Debugger Enable bit

BKBUG = ON	Background debugger functions enabled
BKBUG = OFF	Background debugger disabled

Deep Sleep Watchdog Timer Postscale Select bits

DSWDTPS = DSWDTPS0	1 : 2 (2.1 ms)
DSWDTPS = DSWDTPS1	1 : 8 (8.3 ms)

REGISTER 26-7: FICD: IN-CIRCUIT DEBUGGER CONFIGURATION REGISTER

R/P-1	U-0	U-0	U-0	U-0	U-0	R/P-1	R/P-1
DEBUG	—	—	—	—	—	FICD1	FICD0
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **DEBUG**: Background Debugger Enable bit
1 = Background debugger is disabled
0 = Background debugger functions are enabled

bit 6-2 **Unimplemented**: Read as '0'

bit 1-0 **FICD<1:0:>** ICD Pin Select bits
11 = PGC1/PGD1 are used for programming and debugging the device
10 = PGC2/PGD2 are used for programming and debugging the device
01 = PGC3/PGD3 are used for programming and debugging the device
00 = Reserved; do not use

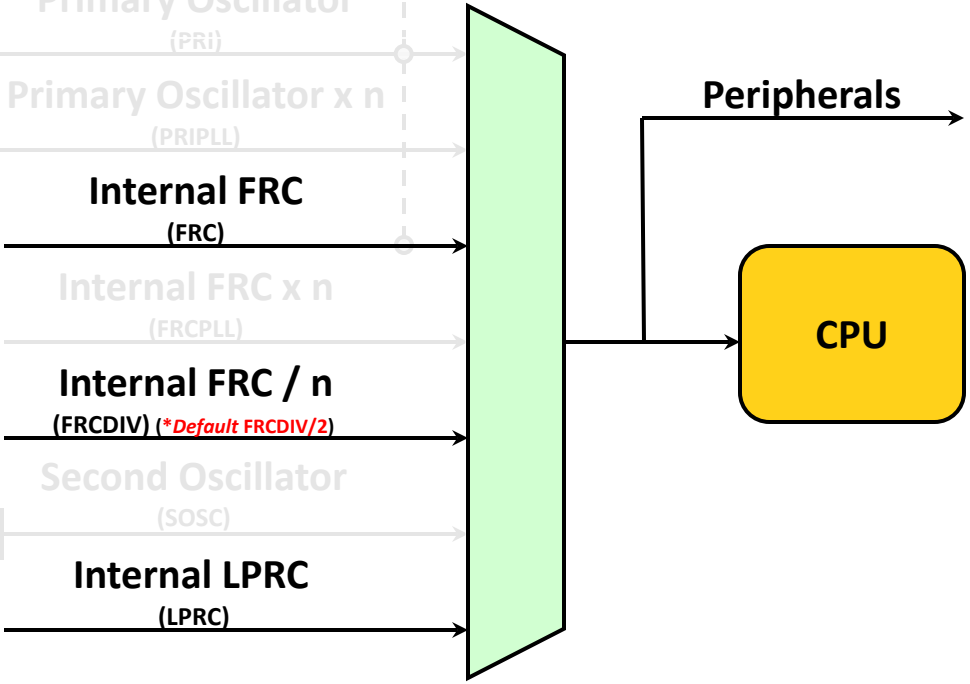
Clock Source (FRC, FRCDIV, LPRC)

- PIC24的Oscillator來源可以選擇使用內部FRC,內部FRC/n或Low Power RC。PIC24內部的FRC,工作頻率為8MHz, LPRC為31KHz。
- 要選擇使用內部FRC/LPRC時,可以在Configuration Bits中將FNOSC <2:0>設定為111b,101b或000b。

***#pragma config* FNOSC = FRC**
***#pragma config* FNOSC = FRCDIV**
***#pragma config* FNOSC = LPRC**

- **FNOSC預設FRCDIV(FRC/n, n(RCDIV)預設= 1 (/ 2)。**

FNOSC<2:0>: Oscillator Selection bits
000 = Fast RC Oscillator (FRC)
001 = Fast RC Oscillator with divide-by-N with PLL module (FRCDIV+PLL)
010 = Primary Oscillator (XT, HS, EC)
011 = Primary Oscillator with PLL module (HS+PLL, EC+PLL)
100 = Secondary Oscillator (SOSC)
101 = Low-Power RC Oscillator (LPRC)
110 = 500 kHz Low-Power FRC Oscillator with divide-by-N (LPFRCDIV)
111 = 8 MHz FRC Oscillator with divide-by-N (FRCDIV)

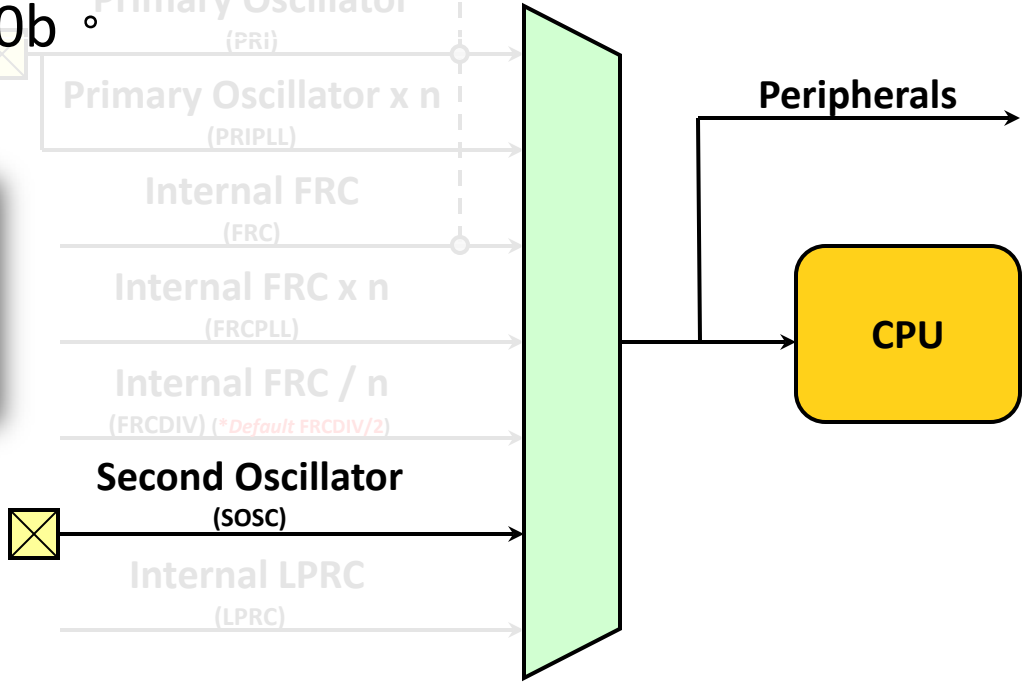


Clock Source (SOSC)

- PIC24的Oscillator來源也可以選擇透過外部的第二組時脈來源接腳(SOSC)取得。
- Second Osc.通常是用來接低速的Crystal,如:32.768KHz。要選擇使用Second Osc.時, 可以在Configuration Bits中將FNOSC<2:0>設定為100b。

***#pragma config* FNOSC = SOSC**

FNOSC<2:0>: Oscillator Selection bits
000 = Fast RC Oscillator (FRC)
001 = Fast RC Oscillator with divide-by-N with PLL module (FRCDIV+PLL)
010 = Primary Oscillator (XT, HS, EC)
011 = Primary Oscillator with PLL module (HS+PLL, EC+PLL)
100 = Secondary Oscillator (SOSC)
101 = Low-Power RC Oscillator (LPRC)
110 = 500 kHz Low-Power FRC Oscillator with divide-by-N (LPFRCDIV)
111 = 8 MHz FRC Oscillator with divide-by-N (FRCDIV)



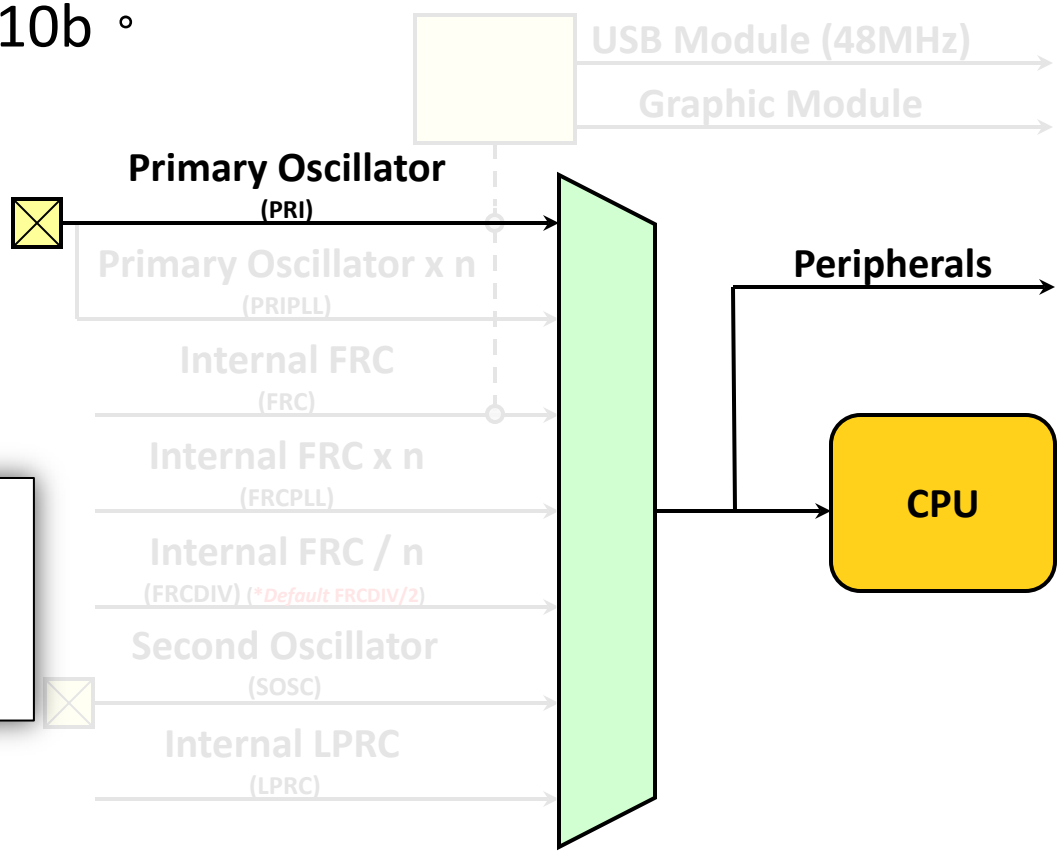
Clock Source (PRI)

- PIC24也可選擇透過外部取得時脈。直接使用Crystal的震盪頻率輸入。要選擇使用Primary Osc.時, 可以在Configuration Bits中將FNOSC<2:0>設定為010b。

  ***#pragma config* FNOSC = PRI**

- Primary Osc.通常是用來連接高速的Crystal。其最高的輸入頻率被限定在25MHz以下。

FNOSC<2:0>: Oscillator Selection bits
000 = Fast RC Oscillator (FRC)
001 = Fast RC Oscillator with divide-by-N with PLL module (FRCDIV+PLL)
010 = Primary Oscillator (XT, HS, EC)
011 = Primary Oscillator with PLL module (HS+PLL, EC+PLL)
100 = Secondary Oscillator (SOSC)
101 = Low-Power RC Oscillator (LPRC)
110 = 500 kHz Low-Power FRC Oscillator with divide-by-N (LPFRCDIV)
111 = 8 MHz FRC Oscillator with divide-by-N (FRCDIV)

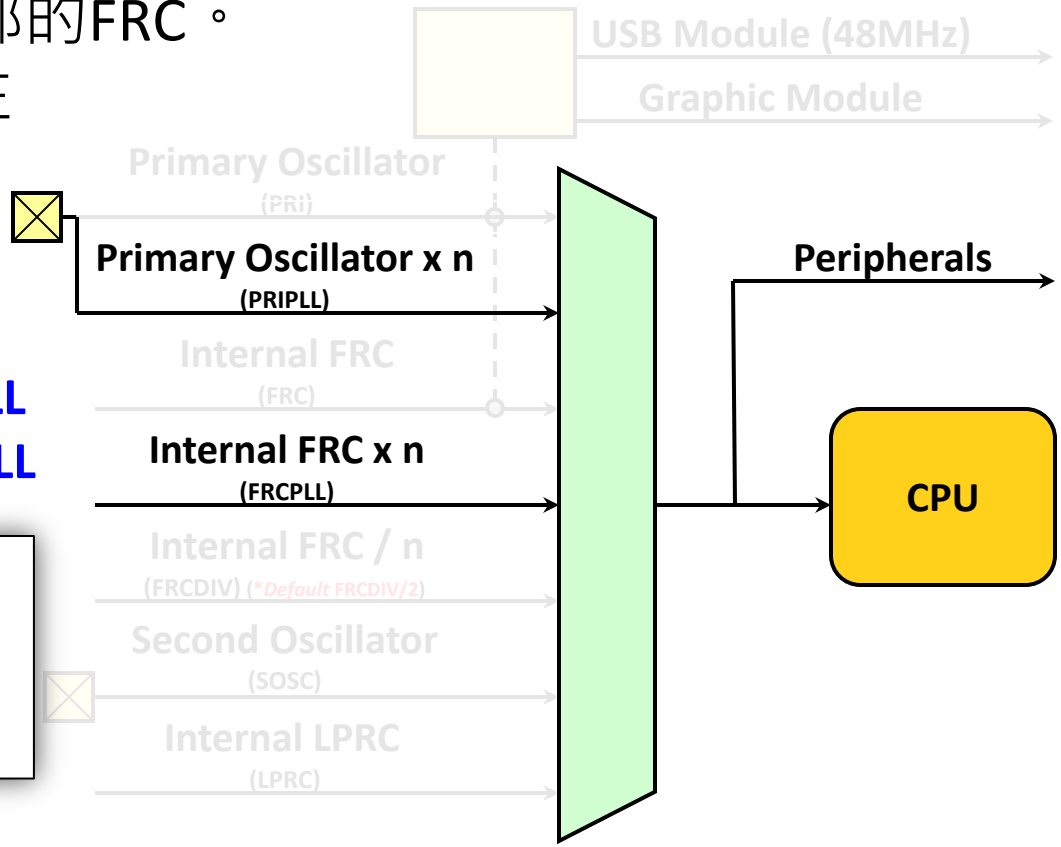


Clock Source (PRIPLL, FRCPLL)

- 除了以上幾種模式,大多數的PIC24應用, 都會透過內部的倍頻線路(PLL, Phase Locked Loop)來提高頻率。倍頻線路可以接受外部的震盪時脈或者內部的FRC。
  要選擇使用PLL時, 可以在Configuration Bits中將FNOSC<2:0>設定為011b或001b。

  ***#pragma config* FNOSC = PRIPLL**

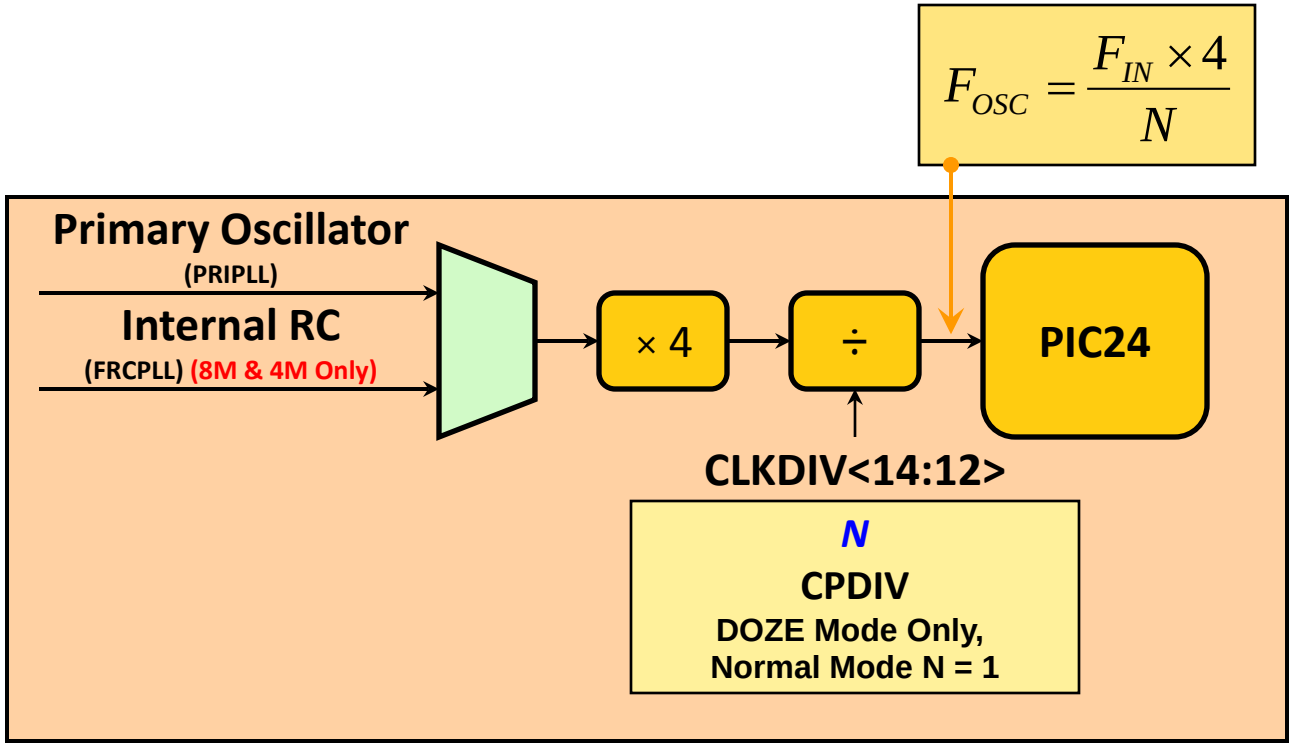
  ***#pragma config* FNOSC = FRCPLL**



FNOSC<2:0>: Oscillator Selection bits
000 = Fast RC Oscillator (FRC)
001 = Fast RC Oscillator with divide-by-N with PLL module (FRCDIV+PLL)
010 = Primary Oscillator (XT, HS, EC)
011 = Primary Oscillator with PLL module (HS+PLL, EC+PLL)
100 = Secondary Oscillator (SOSC)
101 = Low-Power RC Oscillator (LPRC)
110 = 500 kHz Low-Power FRC Oscillator with divide-by-N (LPFRCDIV)
111 = 8 MHz FRC Oscillator with divide-by-N (FRCDIV)

Phase Locked Loop

- PIC24F16KA102的PLL的架構非常簡單, 固定4x的倍頻。

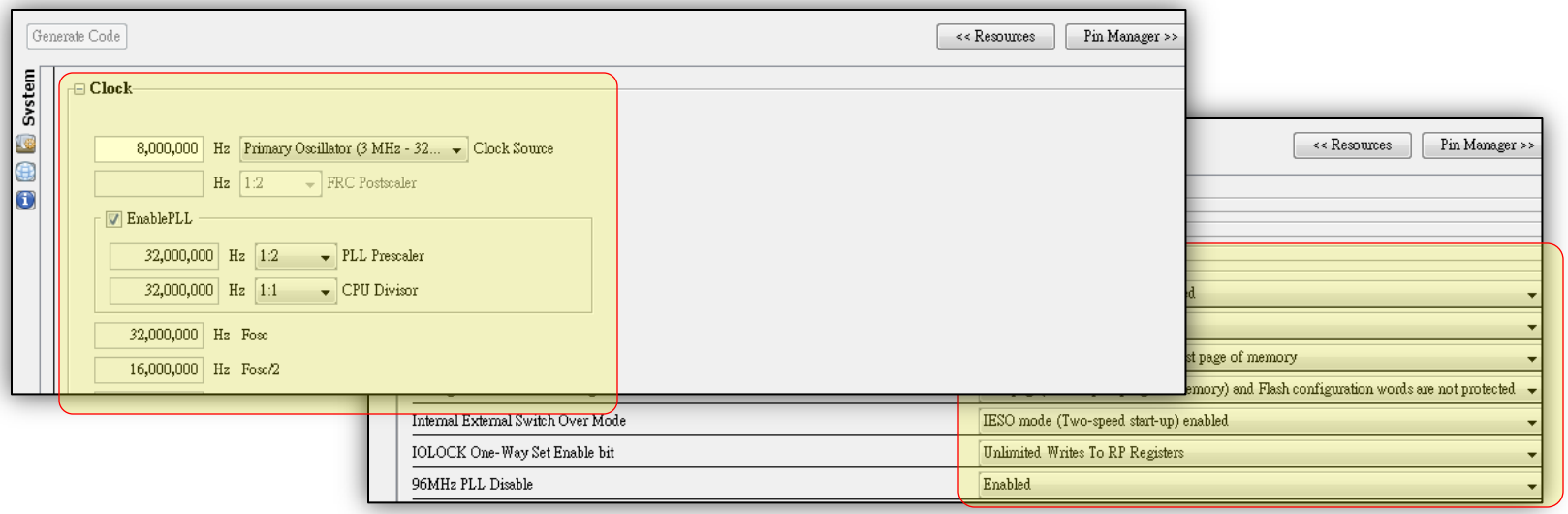


Configuration Bits Setting

- 以前我們要自己在程式中,加入各項Configuration Bits的設定。

```
#pragma config FNOSC = FRCDIV
#pragma config IESO = OFF
#pragma config ICS = PGx3
#pragma config BOREN = BOR3
#pragma config …
#pragma config …
```

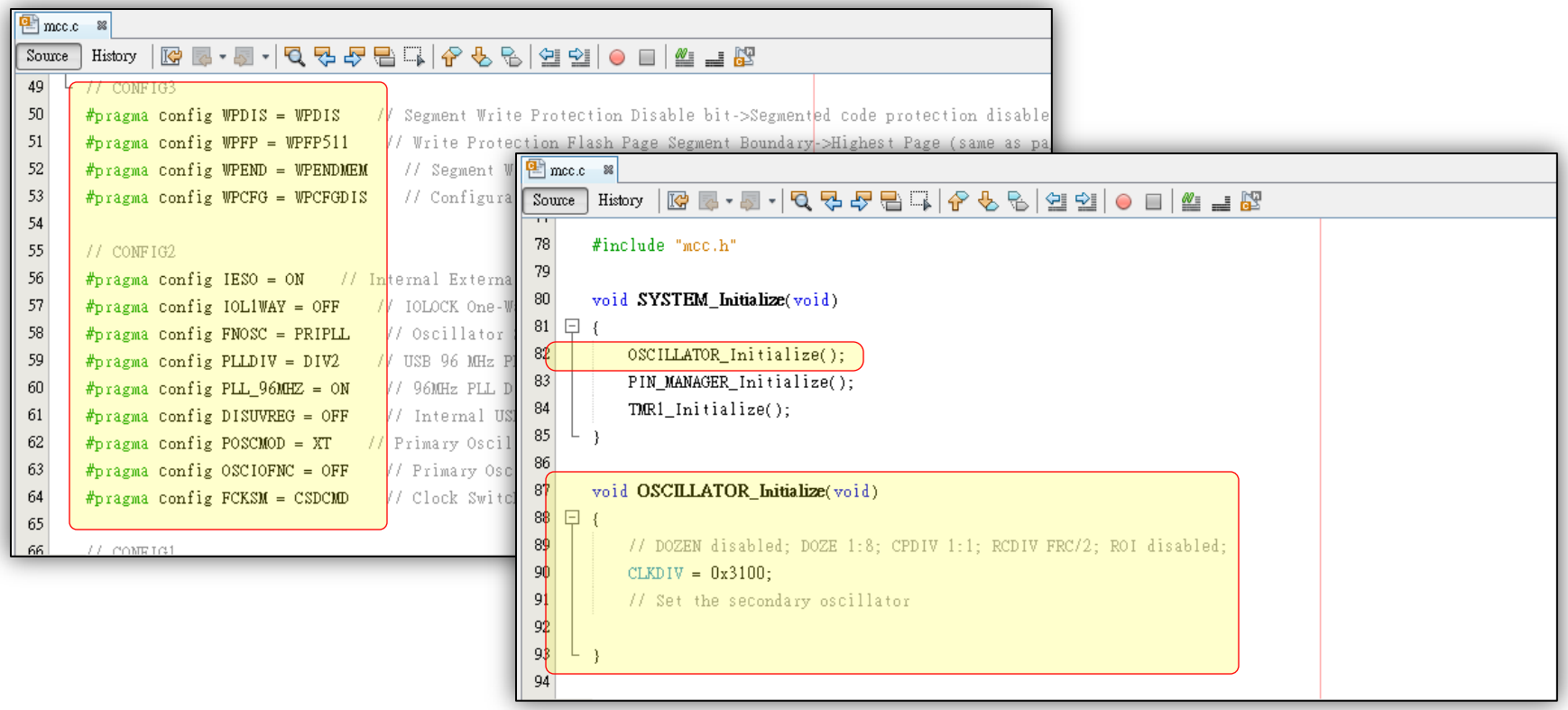
- 現在, 透過MCC, 直接用點選的方式就能搞定。



MICROCHIP
Regional Training Centers

MCC's Configuration Bits Setting

- 執行 **Generate Code** 後, MCC會自動根據選擇, 產生各項Configuration Bits的設定。有些設定需要, 還需要使用程式填入式的的數值, 這部分MCC也會一併完成。



Lab4 – Timer1 Polling PRIPLL

目 標

- 嘗試透過MCC的設定, 改變系統時脈為主要外部時脈輸入(Primary Oscillator), 並開啟PLL, 將系統頻率(F_{OSC})提高到32MHz。除錯介面設定為第一組(PGEC3/PGED3), Watchdog Disable。
- 系統頻率改變後, Timer的設定, 會跟著變動。請再透過MCC調整Lab4中Timer1的設定, 讓Timer1的週期改變成100mS。

- 該如何開始?

Lab4 – Timer1 Polling PRIPLL Step

- 開啟既有專案(C:\Exercises\Exams\Lab4 Timer1 Polling PRIPLL.x)
- 接著開啟開啟MCC, 修改**System**資源, 依照題目要求設定。
- **System**資源修改完成後, 觀察**Timer1**資源的變化, 並依照題目要求再次修改設定。

MICROCHIP
Regional Training Centers

Lab4 – Timer1 Polling PRIPLL

MCC's Setting