



MICROCHIP

Regional Training Centers

Section 1

Microchip 16-Bits

Microcontroller Architecture

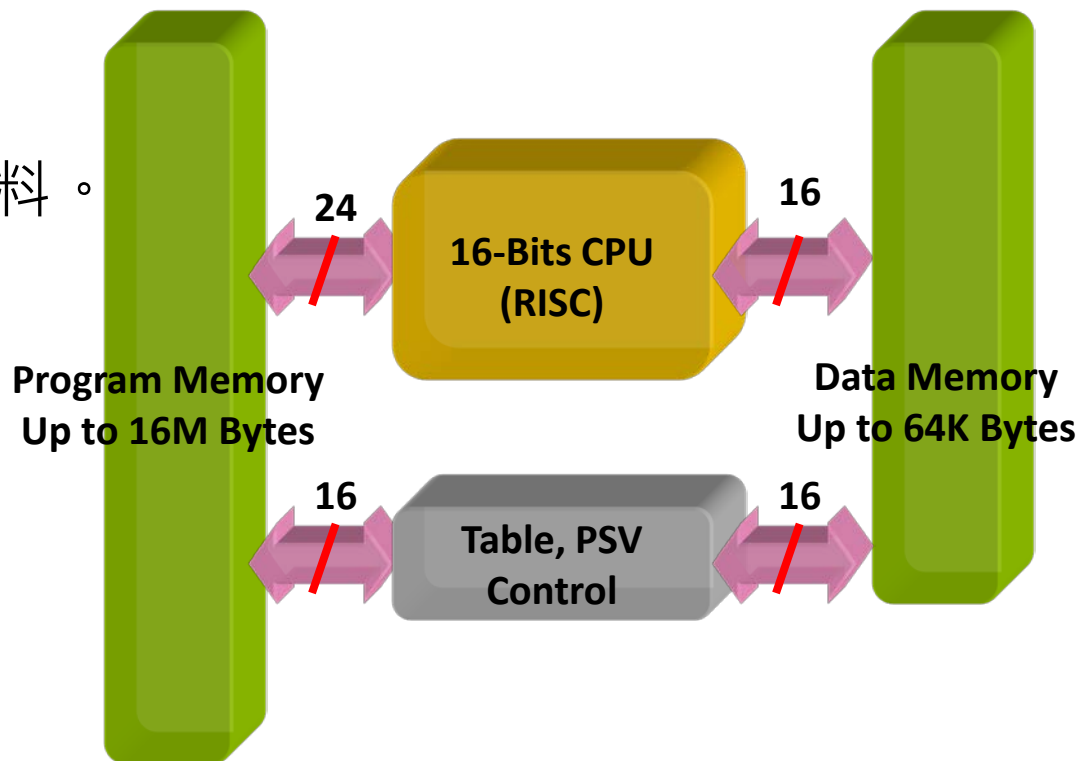
Microchip 16 Bits MCU Series

Microchip 16 Bits MCU包括以下系列:

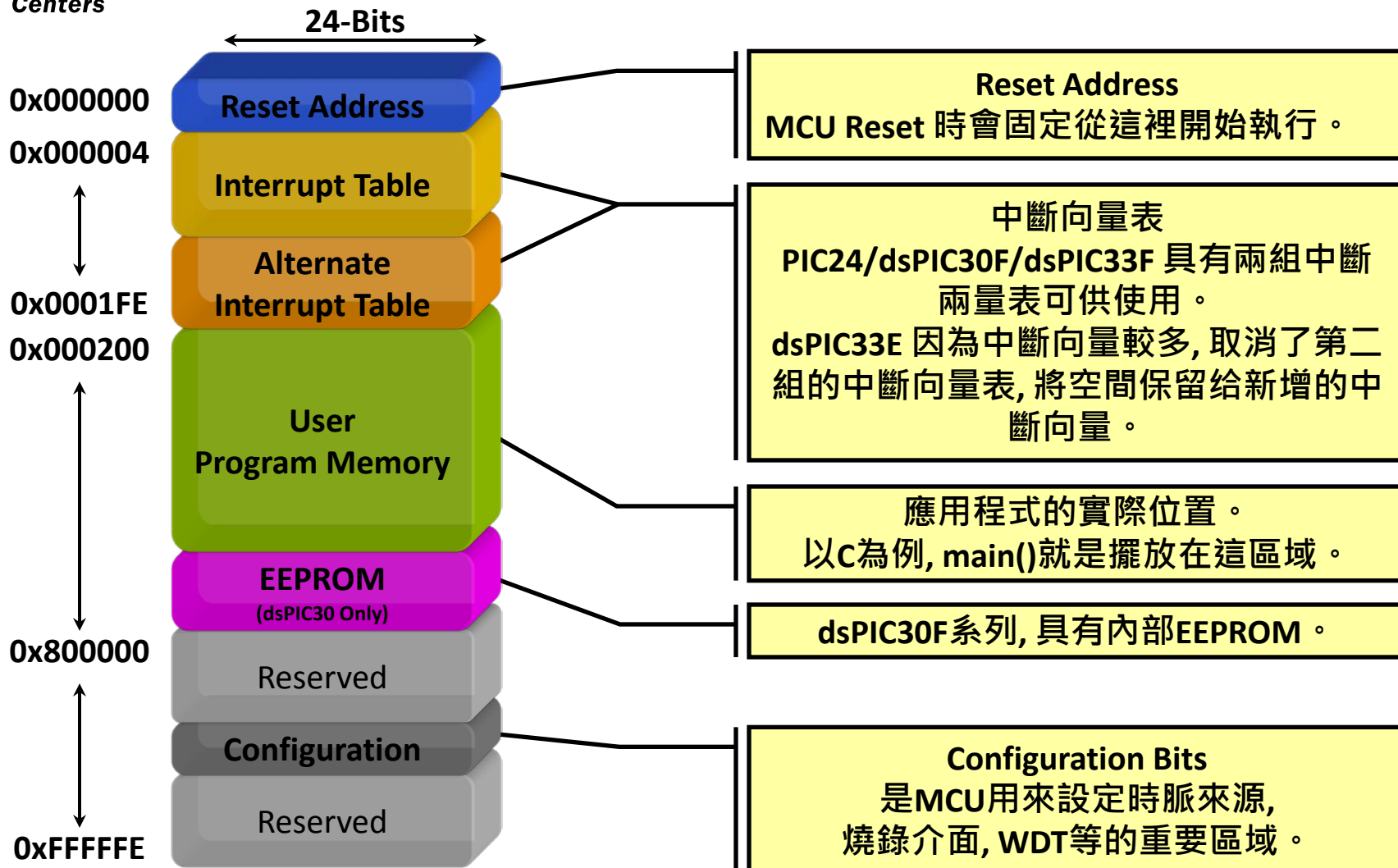
- **PIC24FJ (16 MIPS)**
 - 具備優異的價格與性能比.
- **PIC24HJ (40 MIPS)**
 - 加入 DMA 功能, 提高執行效能到 40 MIPS 。
- **PIC24EP(70 MIPS)**
 - 新一代MCU, 提高執行效能到 70 MIPS 。
- **dsPIC30F (30 MIPS)**
 - 具備數位訊號運算能力。
- **dsPIC33FJ (40 MIPS)**
 - 加入 DMA 功能, 且具備數位訊號的運算能力, 提高執行效能到 40 MIPS 。
- **dsPIC33EP(60 MIPS)**
 - 新一代MCU, 具備USB, 數位訊號運算能力. 提高執行效能到 60 MIPS 。

Harvard Architecture

- Microchip 16-Bits MCU採用哈佛(Harvard)架構。Program, Data Memory 有各自獨立的匯流排。程式與資料可以同時間存取, 提高效率。
- 16-Bits資料寬度:
CPU單次可存取16-Bits資料。
- 24-Bits指令寬度:
指令包含運算碼及資料或位址, 指令的編碼寬度採用24-Bits, 增加定址範圍及彈性。
- 使用Table, PSV Control, 提高Program Memory 的存取效率。



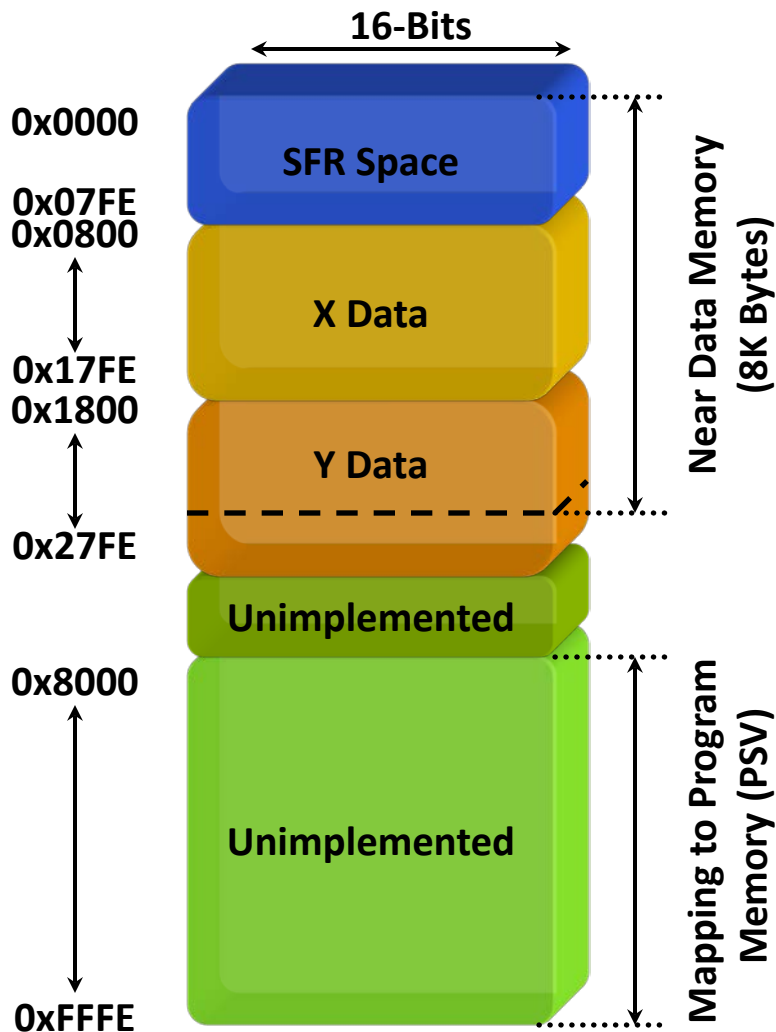
Program Memory Mapping



Program Memory Mapping

- 16-Bbits MCU 的 PC 為 23-Bits, 定址範圍為8M Words。一個指令(Instruction Word)會佔用2個Words, 所以總共可定址到 4M Instruction Words。
- 中斷向量表由0x000004開始, **中斷向量表中並不是放指令, 而是位址。該位址是各種不同中斷源, 其中斷的服務函式的進入點。中斷向量表由0x000004 到 0x0001FE。** User的程式則由0x000200開始放置。(中斷架構後續章節會詳細說明)
- Reset Address(0x000000)通常是放入goto xxxxxx指令。xxxxxx是0x000200後的位址。也就是重置後, 程式會接著跳到使用者的程式區塊, 執行使用者的程式。
- 中斷向量表中只能放位址。如果程式goto到中斷向量表, 會出現不可以預期的情況, 此時CPU將會自動產生Address Error的Trap(Trap, Non-Mask Interrupt, NMI, 不可遮罩中斷)。

Data Memory Mapping



資料記憶體(Data Memory)的實際大小依不同型號有所不同。

資料記憶體(Data Memory)前2K Bytes為特殊功能暫存器區。

X Data, Y Data是提供DSP Engine進行Dual Access用, dsPIC獨有。PIC24沒有DSP Engine, 所以沒有X Data, Y Data的區別, 全部都是X Data。

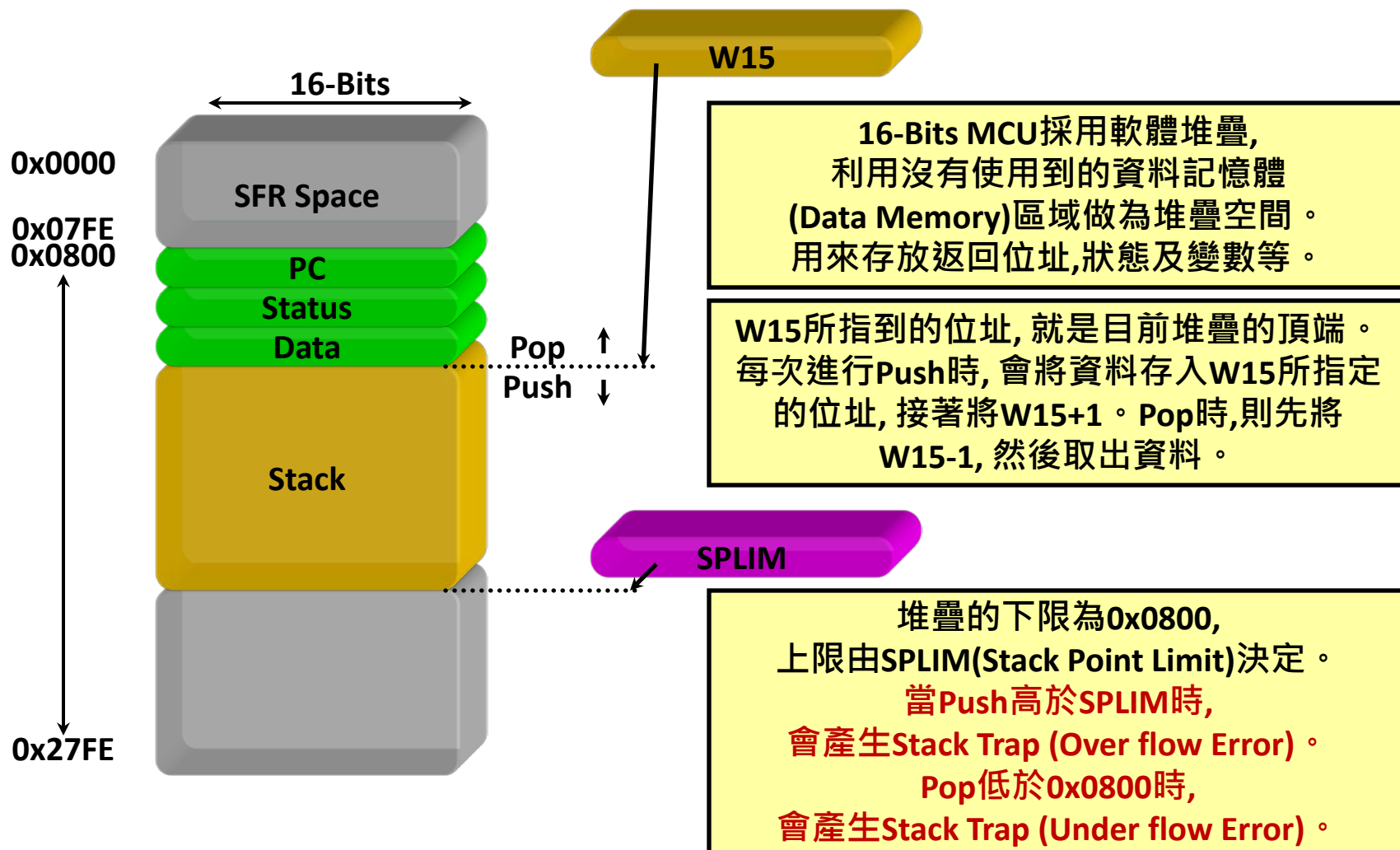
前8K Bytes定義為Near Data Memory。整個64K Bytes定義為Far Data Memory。

0x8000 ~ 0xFFFF 並沒有實際的記憶空間, 而是透過PSV映射到程式記憶體。

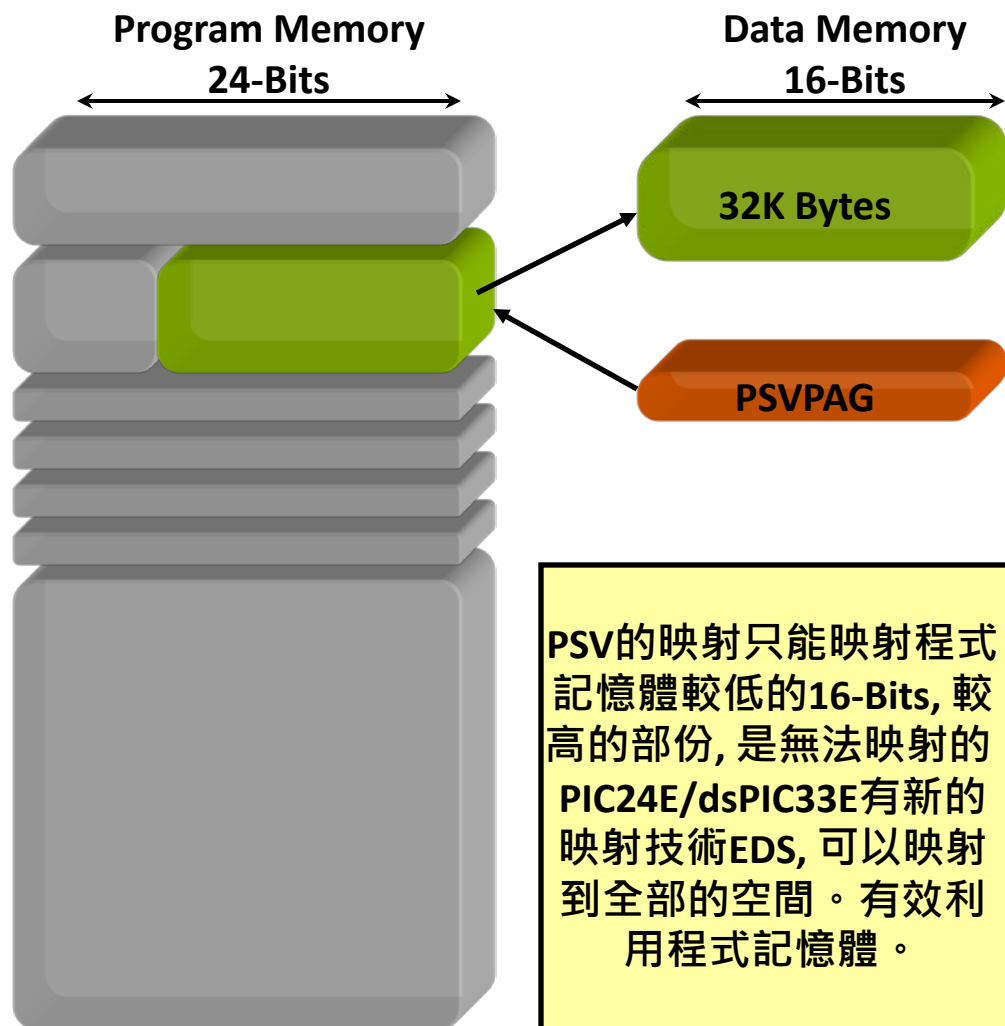
Special Function Registers, SFRs

- 特殊功能暫存器(Special Function Register), 是一個特殊的記憶體區域, 用來控制MCU內部的周邊,如UART, ADC等等。
- 特殊功能暫存器(Special Function Register), 中的每個暫存器都有特定的功能與意義, 例如:某個位址的值可以用來決定UART的鮑率(Baud Rate), 某個位址則存放ADC轉換後的結果。因此特殊功能暫存器的值不可任意的改變, 更不可挪為一般資料記憶體(Data Memory)用。必須詳讀Datasheet後, 再依定義填入正確的設定。
- 一般來說, 同一周邊的控制/狀態等暫存器, 會被安排在鄰近的位址, 所有沒有定義或未使用的位址,預設值都是"0"。

Software Stack



Program Space Visibility



PSV, 是一個利用映射機制讓Data Memory的位址映射到Program Memory的技術。實際的資料存在Program Memory。因此只要透過讀取Data Memory的位址, 便可獲取存在Program Memory的資料。以資料的形式取出。

PSV的映射只能映射程式記憶體較低的16-Bits, 較高的部份, 是無法映射的。PIC24E/dsPIC33E有新的映射技術EDS, 可以映射到全部的空間。有效利用程式記憶體。

程式記憶體以32K Bytes Words為單位, 切割成許多Page。Page直接與資料記憶體0x8000~0xFFFFE的區域做映射。可以透過PSAPAG來選定要映射的PSV Page。

透過PSV可讀取程式記憶體所存放的資料。但PSV只具有讀取的能力, 無法修改(寫入)程式記憶體。

PSV's Advantage

- 使用 PSV可以讓大量的參數資料被快速而有效率地被讀取。
因為已經建立映射, 所以可以直接以讀取資料記憶體(Data Memory)的方式, 來獲取存放在程式記憶體(Program Memory)中的資料。
- PSV常見的使用範例如:
 - 存放數位濾波器的參數, 係數。
 - 存放加解密用的金鑰, 密碼。
 - 存放不會變動的字串, 資料。
 - ...
- **PSV映射的區域, 本質上還是Flash Memory。**
所以只能支援讀取, 不能寫入。
強制寫入時, 系統會出現例外(Trap, Address Error)事件。

Machine & Instruction Cycle

- **機械週期(Machine Cycle) T_{osc} :**
一個時脈(Clock)花費的時間。通常是CPU工作頻率的倒數。
- **指令週期(Instruction Cycle) T_{cy} :**
完成一個指令所需要花費的時間。
- MCU完成一個指令的動作, 通常需要花費好幾的機械週期 (T_{osc})。不同架構, 所需要的時脈(Clock)數量也不同。
- **dsPIC30系列:**
 $1 T_{cy} = 4 T_{osc}$ 。
Ex: 120 MHz => 30 MIPS, $T_{cy} = 33\text{nS}$.
- **PIC24F/PIC24H/dsPIC33F/dsPIC33E系列:**
 $1 T_{cy} = 2 T_{osc}$ 。
Ex: 32 MHz => 16 MIPS, $T_{cy} = 62.5\text{nS}$

Microchip 16 Bits Clock Source

- 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。
(時脈來源的設定會在後續章節說明)
- 時脈來源的設定, 必須正確如果設定錯誤, CPU會接收到錯誤的頻率, 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。

