第一章　微處理機導論

1-1　微處理機之發展史

1-2　微處理機之方塊圖

1-3　微處理機之基本結構

1-4　微處理機指令之提取、解碼及執行

1-1 微處理機的發展史

廠商 / 位元	Intel	AMD	Cyrix	Motorola	Zilog	MOS Technology
4 bit	4004					
8 bit	8008 8080			6800	Z80	6502
16 bit	8086/88 80286			68000 68010		
32 bit	80386 80486	5X86	5X86			
64 bit	Pentium (P5/P54C) Pentium Pro (P6） Pentium MMX Pentium II Pentium III Pentium 4 Pentium D/EE Core 2 Due Core 2 Quad Core 2 Extreme	5K86 (K5/K6) K6-2 Athlon (K7) Athlon (K7-5) Athlon 64 (K8) Athlon FX (K8) Athlon X2 (K8) Phenom X2 (K10) Phenom X3 (K10) Phenom X4 (K10)	6X86 MII			

1-1 微處理機的發展史

- Intel 80286 的記憶體定址工作模式分為二種

  ☆真實模式：向下相容 8bit 的 8086 程式，控制 1MB 記憶體

  ☆保護模式：可以執行 16bit 程式，控制 16MB 記憶體

- Intel 80386DX 率先使用管線（Pipeline）式解碼、執行架構

  CPU 內部各個單元電路不必等待共用匯流排傳輸

  以專線傳輸達到分工，每一分工專線稱為一個階層（Stage）

  當分工專線到七個階層就稱為超管線（Super-Pipeline）架構

- Intel 80486 是整合 80386 CPU、80387 FPU、8KB L1 Cache

  80486DX2 以倍頻技術將內部頻率提升為外部匯流排時脈的二倍

1-1 微處理機的發展史

- Intel Pentium 提出超純量（Super Scalar）架構設計
  可使 CPU 在一個週期內，同時執行多道指令、算術運算處理
  並支援雙 CPU 的平行處理作業
- Intel Pentium Pro 開始內建第二階層快取記憶體（L2 Cache）
- Intel Pentium MMX 增加 57 組 MMX 指令
  最佳化處理多媒體指令集，提升音波合成、繪圖輸出執行速度
- Intel Pentium II 將 CPU 與 L2 Cache 晶片分開包裝
  減少 IC 生產失敗率，達到降低 CPU 成本
- Intel Pentium III 製程技術由 0.25μm進步到 0.18μm
  CPU 時脈速度突破到 GHz，進入高速 CPU 時代

1-1 微處理機的發展史

- Intel Pentium 4 以後 CPU 使用 Netburst 架構，核心技術包含
  - ☆雙倍頻算術邏輯單元（Double Pumped ALU）
    算術邏輯單元在時脈的正緣與負緣都產生觸發
  - ☆執行微碼追蹤快取（Execution Trace Cache）技術
    CPU 指令直接在 L1 Cache 內解碼，加快指令處理速度
  - ☆四泵浦匯流排（Quad Pumped Bus）架構
    資料匯流排在同時脈內，以倍頻技術同時傳輸四筆 64bit 資料
    例如：實際只有 100MHz 外頻可使前端匯流排速度達 400MHz
  - ☆超執行緒（Hyper Threading，HT）：模擬雙核心 CPU 技術
    可在同一時間內處理二個應用程式執行緒

1-1 微處理機的發展史

- Intel Core 2 系列 CPU 使用 Intel Core 架構，核心技術包含
  - ☆多核心設計

    之前 CPU 提高工作時脈增加效能，會導致高溫、高耗電現象

    現在以製程技術將多個核心做在同一個 CPU 裡

    每個核心 L1 Cache 互相連接，可共同分享 L2 Cache

    達到最高效能功率消耗比（Performance per watt）
  - ☆虛擬化技術（Virtualization Technology，VT）

    可將單個 CPU 模擬成多個 CPU 來執行

    允許一個平臺同時運行多個作業系統

    應用程式可在獨立空間內執行互不影響，顯著提高電腦效率

1-1 微處理機的發展史

- 學後評量：

C 1、下列何者全是 32 位元 CPU？

（A）80386、6502、Z80　　（B）68020、80386、68000

（C）68020、80386、80486　　（D）80860、Pentium、K6

A 2、可以支援雙 CPU 或多 CPU 之作業系統的 CPU，稱為？

（A）平行處理架構　　（B）超純量架構

（C）超管線架構　　（D）超執行緒架構

D 3、下列何者不會改善 CPU 的執行速度？

（A）增加資料匯流排寬度　　（B）提升基本時脈頻率

（C）管線技術改良　　（D）增加外部記憶體容量

第一章　微處理機導論

1-1　微處理機之發展史

1-2　微處理機之方塊圖

1-3　微處理機之基本結構

1-4　微處理機指令之提取、解碼及執行

1-2 微處理機的方塊圖

- 電腦系統的中央處理單元（Central Processing Unit，CPU）

  俗稱微處理機（Microprocessor），CPU 內部三種電路結構：

  控制電路、算術邏輯運算電路、暫存器電路

  ☆控制電路：負責電腦指令解碼及系統執行控制

  ☆算術邏輯運算電路：負責電腦算術運算及邏輯運算處理

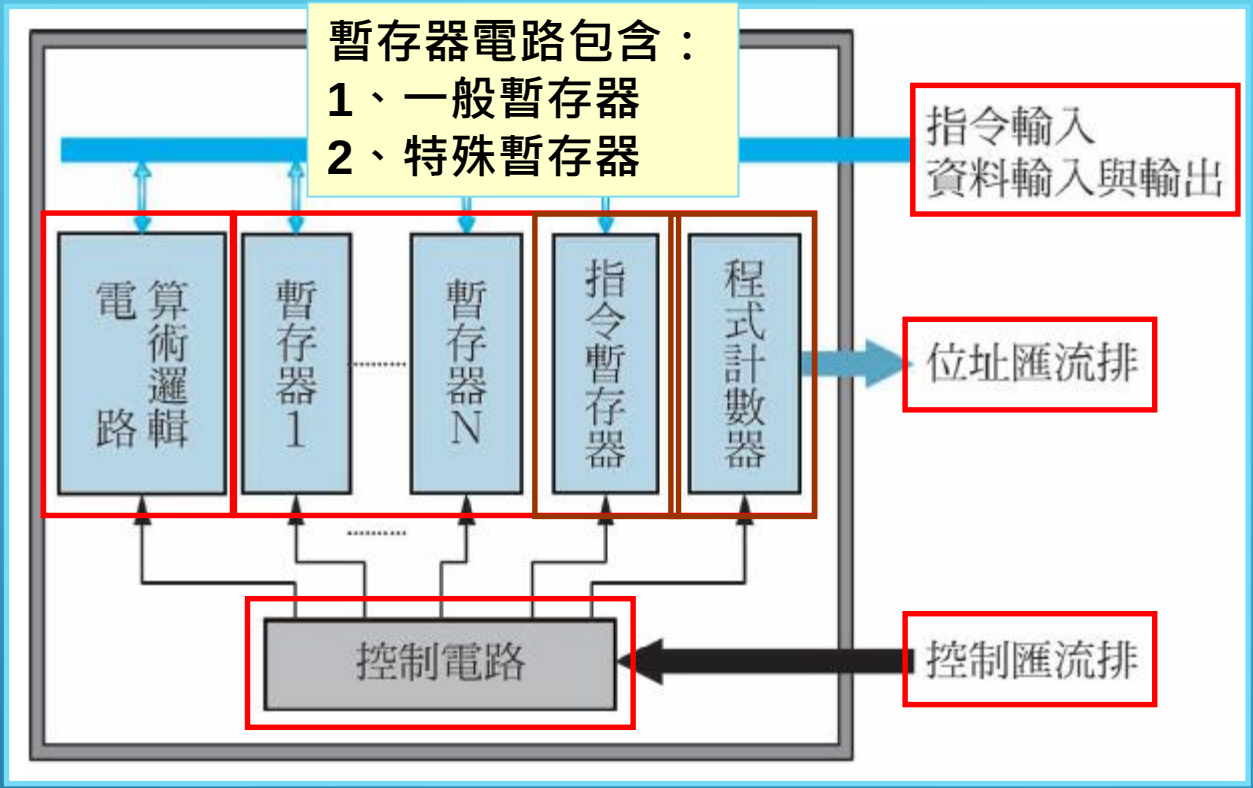
  ☆暫存器電路：負責電腦暫時儲存指令、資料的地方

- CPU 指令、資料傳輸與控制都是透過共用匯流排線（Bus）

  共用匯流排線可節省傳輸線路，目前匯流排線有三種不同用途：

  資料匯流排、位址匯流排、控制匯流排

1-2 微處理機的方塊圖



CPU 指令與資料都存在外部記憶體，依靠位址線來規劃指令、資料存放位置

程式計數器（Program Counter，PC）透過位址匯流排到記憶體取得指令或資料，PC 內容是 CPU 下一次要讀取的記憶體位址，讀取後 PC 累加 1

指令暫存器（Instruction Register，IR）儲存從資料匯流排輸入的指令

1-2 微處理機的方塊圖

- 學後評量：

B 1、下列何者並不包含於 CPU 內？

（A）控制單元　　（B）輸入/輸出單元

（C）算術邏輯單元　　（D）程式計數器

C 2、用以指定待執行指令所在位址的是？

（A）指令暫存器　（B）資料計數器　（C）程式計數器　（D）累加器

第一章　微處理機導論

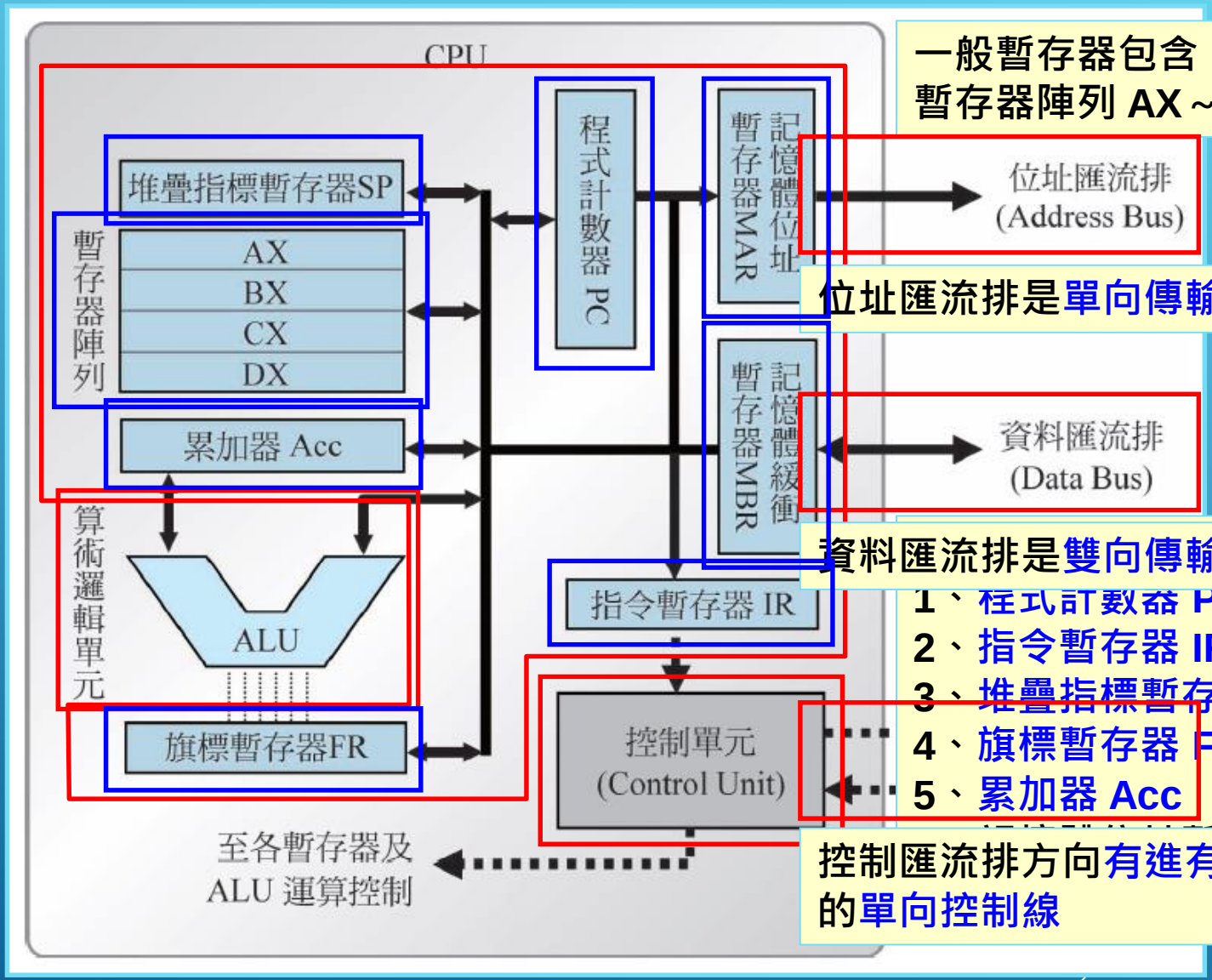
1-1　微處理機之發展史

1-2　微處理機之方塊圖

1-3　微處理機之基本結構

1-4　微處理機指令之提取、解碼及執行

1-3 微處理機的基本結構



一般暫存器包含：
暫存器陣列 AX～DX

位址匯流排是單向傳輸線

資料匯流排是雙向傳輸線

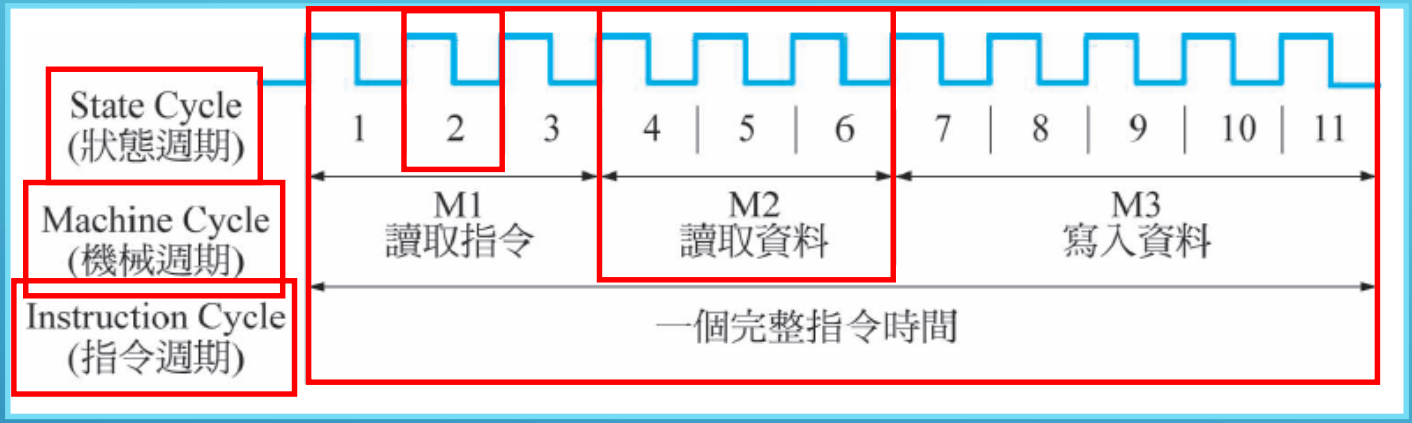
1、程式計數器 PC
2、指令暫存器 IR
3、堆疊指標暫存器 SP
4、旗標暫存器 FR
5、累加器 Acc

MAR
MBR

控制匯流排方向有進有出的單向控制線

1-3 微處理機的基本結構

- 控制單元（Control Unit，CU）：



☆**狀態週期**（State Cycle）： CPU 的**時脈**（**Clock**）信號

Clock 是 CPU 工作頻率的倒數，為 CPU 最小的單位時間

☆**機械週期**（Machine Cycle）：CPU 完成一個小動作的時間

一個小動作機械週期通常需花幾個狀態週期的時間

☆**指令週期**（Instruction Cycle）：CPU 執行一個完整指令的時間

一個完整指令週期包含**指令讀取**、**資料讀取**、**指令分析**與**執行**

1-3 微處理機的基本結構

☆主機板的時脈頻率稱為外頻 或 前端匯流排頻率（FSB）

外頻太高容易產生電磁輻射干擾，所以一般設計的比較低

早期 80X86 外頻只有 66MHz / 100MHz / 133MHz

目前 Core 2 外頻已提升到 1066MHz / 1333MHz / 1600MHz

☆ 80486 以後採用倍頻技術，來提高 CPU 內部的時脈頻率

例如：Pentium III 800 是指 CPU 內部工作頻率為 800MHz

可搭配外頻 100MHz 與 8 倍頻、外頻 133MHz 與 6倍頻

➢算數邏輯單元（Arithmetic Logic Unit，ALU ）：

☆負責算術運算與邏輯運算處理，運算結果會存回累加器 Acc

☆ 80X86 CPU 外部搭配 80X87 輔助處理器，加強運算速度

☆ 80486、Pentium 以後的 CPU 就已經內含輔助處理器

1-3 微處理機的基本結構

- 暫存器（Register）：

  暫存器功用相當於記憶體，是 CPU 存放資料與特定用途控制

  暫存器與記憶體差異：暫存器的容量小、成本高、執行速度快

  ☆程式計數器（Program Counter，PC）：

  儲存 CPU 下一次要到外部記憶體中執行的位址，

  控制 CPU 執行動作順序，程式計數器會以 PC＋1 方式遞增

  遇到跳躍、中斷、副程式呼叫時，會改變原來遞增的執行順序

  ☆指令暫存器（Instruction Register，IR）：

  是 CPU 從外部記憶體中讀入指令存放的地方

  採用佇列（Queue）結構具有先入先出（FIFO）特性

  Intel Pentium 以後的 CPU 將指令佇列稱為 L1 Cache

1-3 微處理機的基本結構

☆堆疊指標暫存器（Stack Pointer Register，SP）：

指示堆疊區中儲存的位址，堆疊區在外部記憶體中有專屬範圍

堆疊（Stack）結構具有後入先出（LIFO）的特性

當程式計數器 PC 遇到中斷、副程式呼叫時

將返回要繼續執行的位址、旗標暫存器內容先存入堆疊區中

等返回時即可從堆疊中取出原先存入值，繼續下一個動作執行

☆旗標暫存器（Flag Register，FR）：

設定 CPU 執行的工作模式、記錄 CPU 指令執行完的狀態值

旗標是以位元為單位，做為 CPU 執行下一個指令的參考

例如：運算結果是否為 0、正負符號、是否進位、是否溢位

週邊設備是否有中斷要求

1-3 微處理機的基本結構

☆累加器（Accumulator，Acc）

算術邏輯單元的必要暫存器，儲存運算後的結果

☆記憶體位址暫存器（Memory Address Register，MAR）

CPU 要對外部記憶體溝通的位址，先暫時存放的暫存器

☆記憶體緩衝暫存器（Memory Buffer Register，MBR）

CPU 與外部記憶體間讀入/寫出的資料，先暫時存放的暫存器

➢匯流排（Bus）

☆位址匯流排是單向傳輸，提供 RAM 或 I/O 位址的連接管道

☆資料匯流排是雙向傳輸，CPU 與 RAM、I/O 傳送資料的管道

☆控制匯流排是有進有出的單向傳輸

包含系統時脈、讀寫控制、介面控制、中斷要求、重置控制等

1-3 微處理機的基本結構

- 學後評量：

D 1、電腦系統的時脈（Clock）週期，是 CPU 最小的單位時間，可決定 CPU 的執行速度，一般稱時脈週期為？

（A）指令週期　（B）機械週期　（C）工作週期　（D）狀態週期

D 2、CPU 從記憶體讀入指令後，在等待控制單元解碼之前，是存放在？

（A）程式計數器　（B）記憶體緩衝暫存器

（C）堆疊器　（D）指令暫存器

A 3、Queue 的資料存取順序為？

（A）先入先出　（B）先入後出　（C）後入先出　（D）隨機存取

第一章　微處理機導論

1-1　微處理機之發展史

1-2　微處理機之方塊圖

1-3　微處理機之基本結構

1-4　微處理機指令之提取、解碼及執行

1-4 微處理機指令的提取、解碼及執行

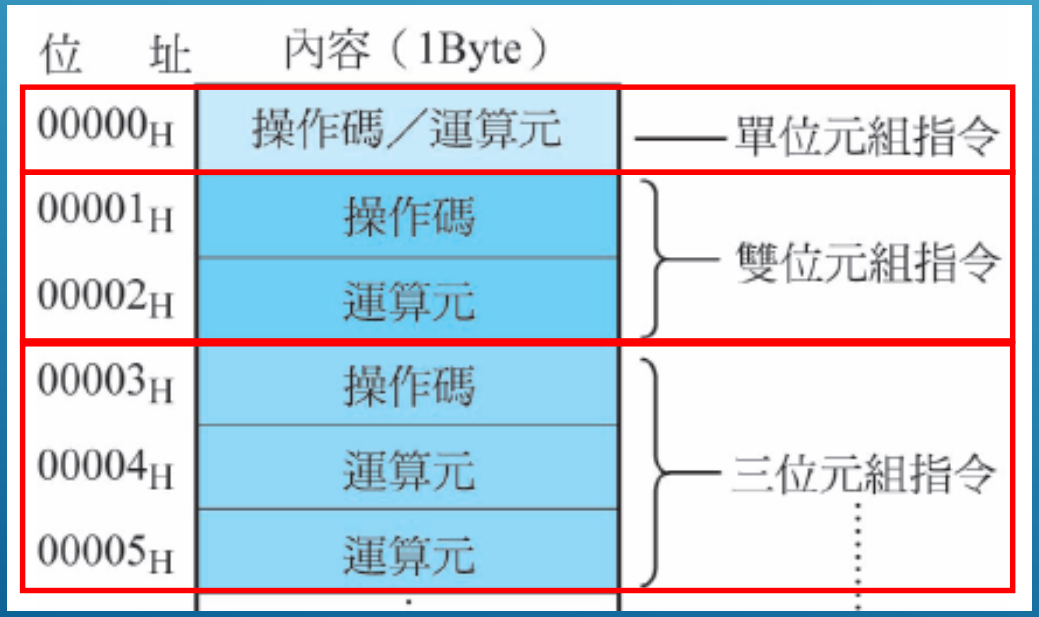
- **CPU 指令的基本格式：**

CPU 是連續、重複到記憶體提取指令，給控制單元解碼並執行

CPU 指令都是一串二進位編碼，分為操作碼、運算元二部份

☆操作碼（Operation Code，OP Code）：定義指令動作

☆運算元（Operand）：提供指令所需的運算資料或位址



操作碼與運算元隱含一起
只佔 1B 記憶體容量

操作碼佔 1B 記憶體容量
運算元佔 1B 記憶體容量

操作碼佔 1B 記憶體容量
運算元佔 2B 記憶體容量

1-4 微處理機指令的提取、解碼及執行

- 低階程式語言：

☆機械語言（Machine Language）：

CPU 指令由二進位編碼構成，CPU 唯一可辨識、執行的語言

在所有程式語言中唯一不需翻譯，執行速度最快

☆組合語言（Assembly Language）：

由於機械語言不容易被程式設計師辨認及記憶

CPU 指令碼改用簡單英文縮寫或符號所形成的程式語言

CPU 無法直接辨識、執行，須先經過組譯器（Assembler）

翻譯成機械語言才能由 CPU 執行，執行速度僅次於機械語言

1-4 微處理機指令的提取、解碼及執行

- 組合語言指令格式分為四個部分：標記、操作碼、運算元、註解
  - ☆標記：在組譯之後相當於程式在記憶體中的位址
    提供指令分歧、跳躍、副程式呼叫使用
    標記屬於非必要欄位可省略不用
  - ☆操作碼：定義指令動作，操作碼屬於必要欄位不可空白
  - ☆運算元：提供指令運算所需的資料或資料所在的位址
  - ☆註解：用於編寫程式說明，增加程式可讀性
    組譯器不會做翻譯動作，註解使用時須以「；」開頭
    註解屬於非必要欄位可省略不用。

1-4 微處理機指令的提取、解碼及執行

- **CPU 指令提取（Fetch）：**

  CPU 到外部記憶體中，將操作碼讀入 CPU 的指令暫存器存放

指令提取任何指令動作的時序步驟完全相同

指令執行任何指令動作由控制單元解碼，時序步驟不完全相同

T1：MAR ← PC

T2：PC ← PC + 1

T3：MBR ← Mem（MAR）

T4：IR ← MBR

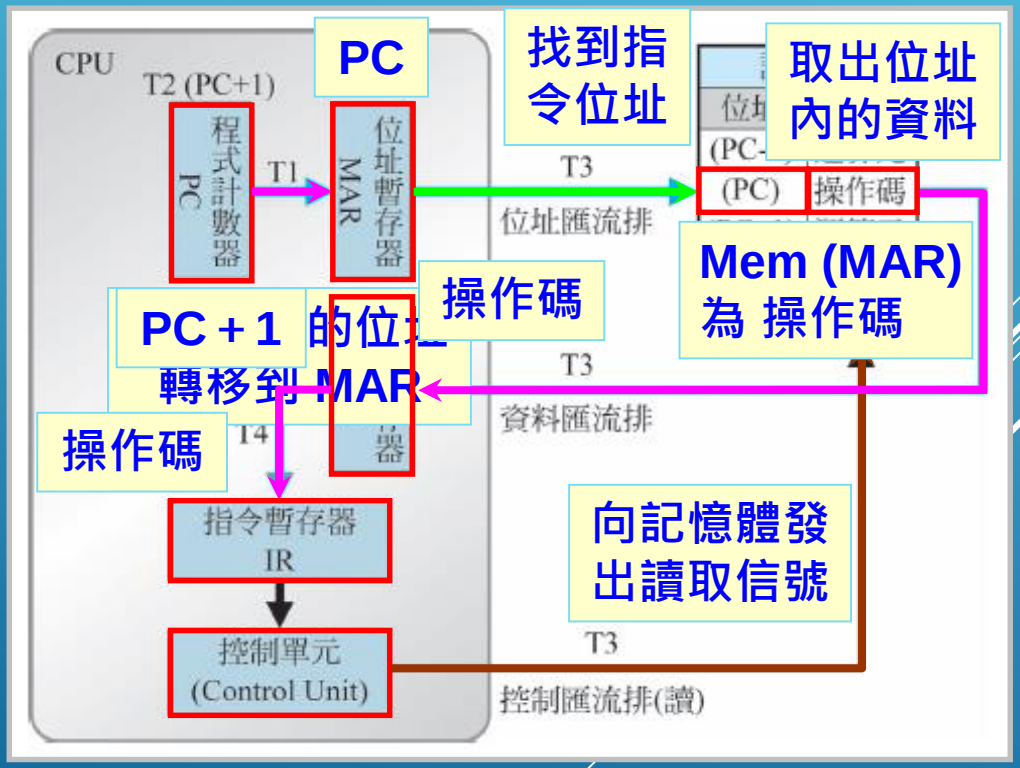
1、CPU 透過 MAR 將 PC 的位址送到位址匯流排

記憶體中指令操作碼的位址

程式計數器內容會自動增加 1

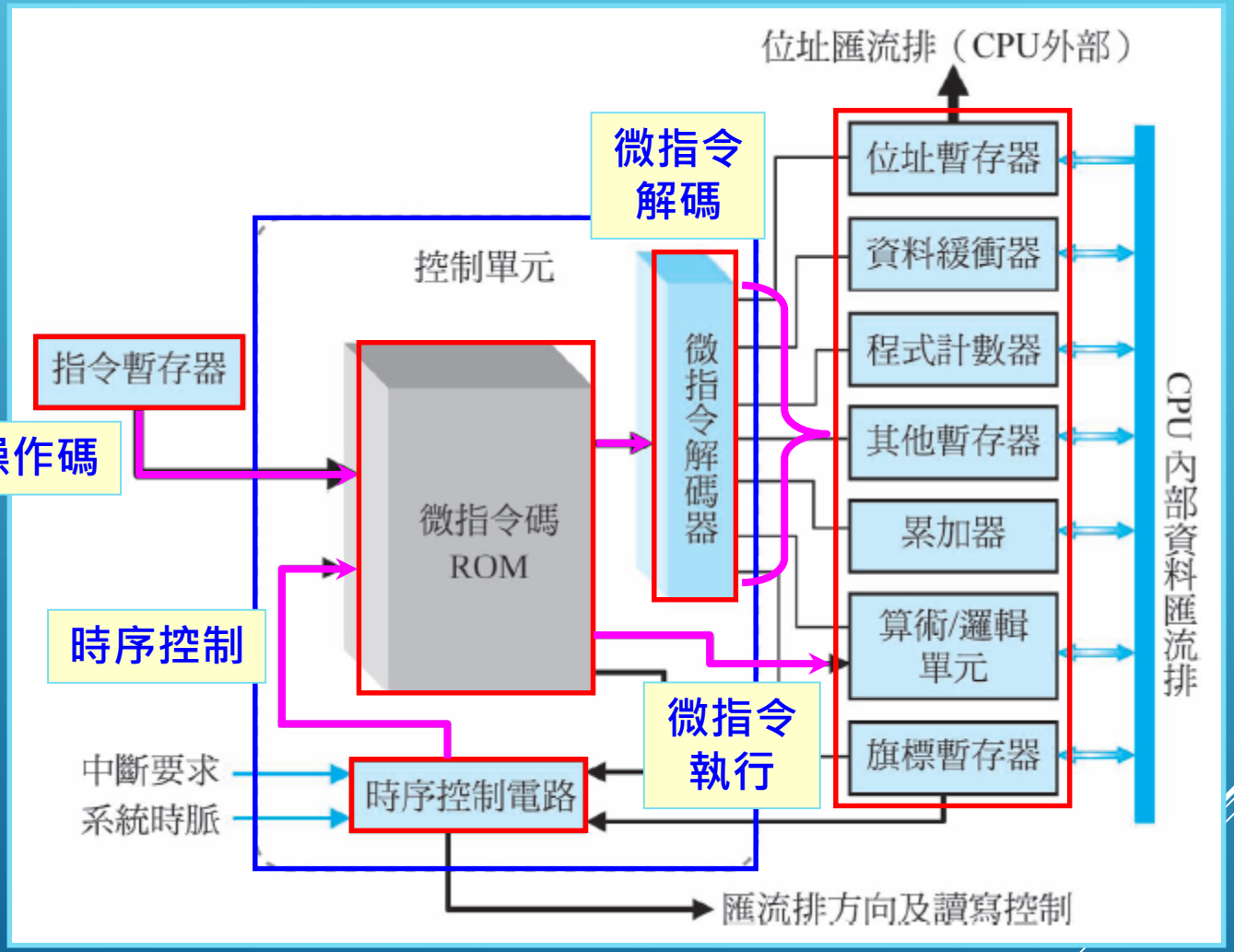
以便讀取下一指令、資料

或尚未讀完的指令、資料



1-4 微處理機指令的提取、解碼及執行

- CPU 的微碼控制單元簡化圖



1-4 微處理機指令的提取、解碼及執行

- CPU 指令解碼（Decode）：

☆微動作（Micro-Operation）：

每個機械週期的解碼動作，可在控制單元中規劃硬體電路完成

某個暫存器將資料寫入匯流排或致能算術邏輯單元做一個運算

☆微指令（Micro-Instruction）：

對於含有複雜演繹過程的指令，硬體設計難度、成本相對增高

則帶入軟體技術方式完成控制

☆微程式（Micro-Program）：

在控制單元的 ROM 中選擇不同微指令區構成一連串微指令碼

☆可微程式化（Micro-Programmable）微處理機：

在控制單元的微程式在生產後，還可以修改的微處理機

1-4 微處理機指令的提取、解碼及執行

- **CPU 指令執行（Execute ）：**

不同指令的提取時序步驟相同，但是執行時序步驟不會完全相同

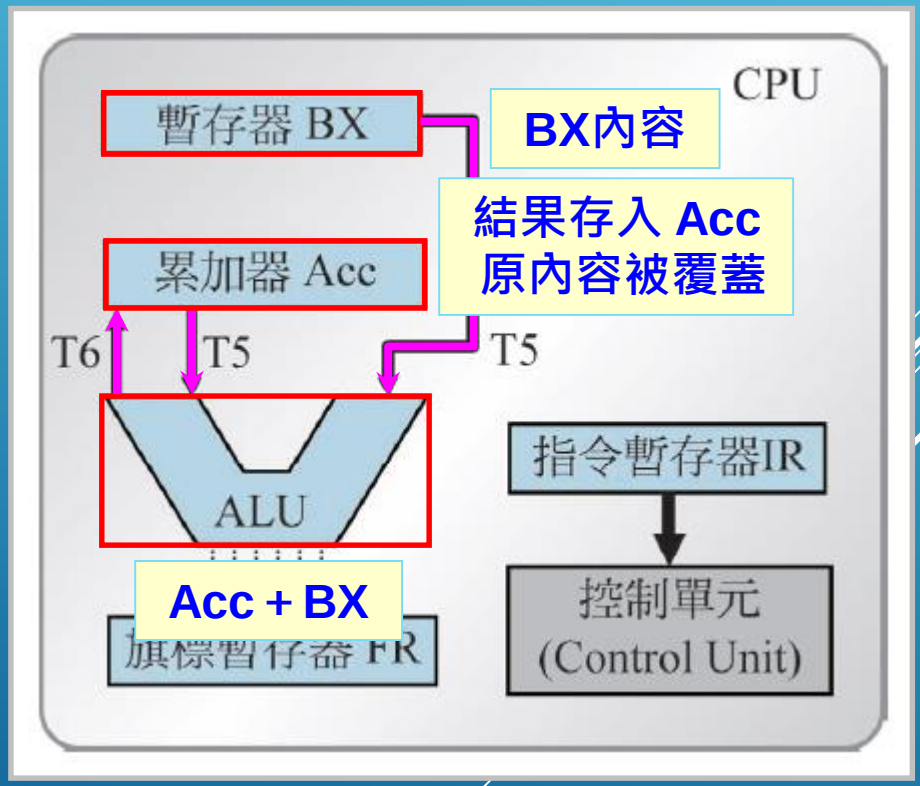
☆執行 ADD　Acc，BX 指令：

指令動作：CPU 內部的一個加法運算（Acc ← Acc＋BX）

T5：ALU ← Acc + BX

T6：Acc ← ALU

BX 暫存器與累加器 Acc 的內容
將 ALU 運算結果轉移到 Acc



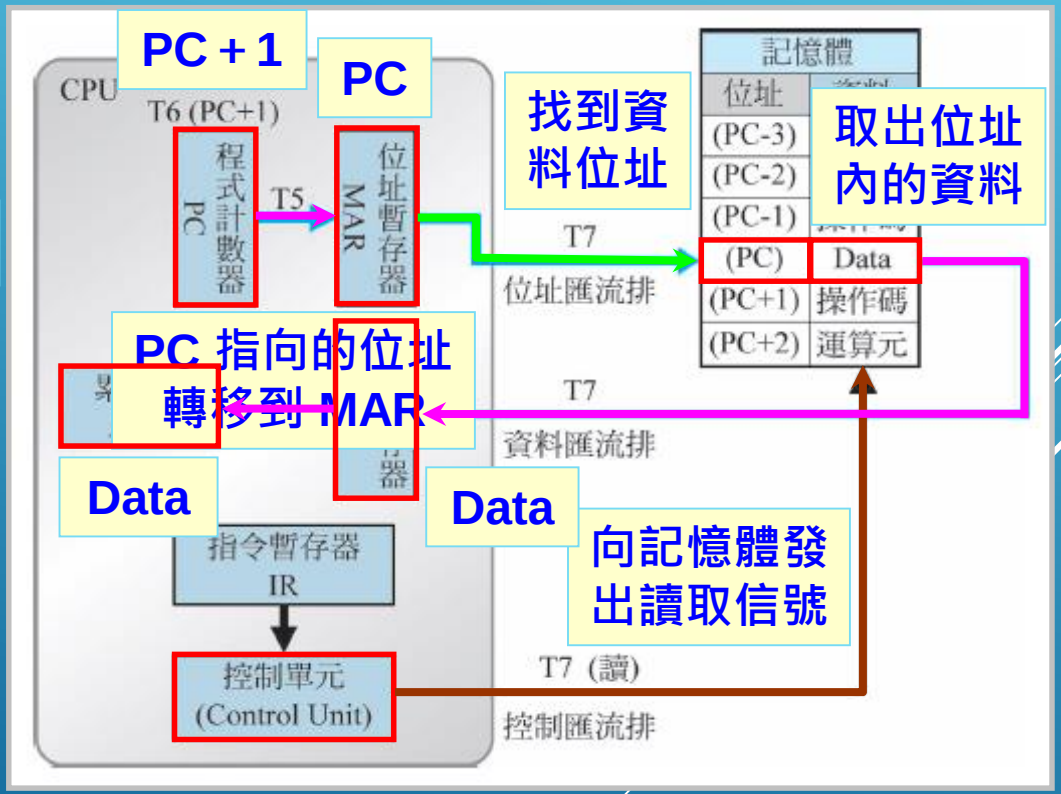
1-4 微處理機指令的提取、解碼及執行

☆執行 MOV　Acc，Data 指令：

指令動作：外部記憶體資料讀入到 CPU 內部暫存器

CPU 將記憶體中的資料 Data 搬移到累加器 Acc 內存放

T5：MAR ← PC
T6：PC ← PC + 1
T7：MBR ← Data
T8：Acc ← MBR
1、CPU 透過 MAR 將 PC 的位址送到位址匯流排
讀取記憶體資料 Data 的位址
程式計數器內容會自動增加 1
CPU 將 MBR 內容轉移到累加器 Acc 存放



1-4 微處理機指令的提取、解碼及執行

☆執行 MOV （OPR），AX 指令：

指令動作：CPU 內部暫存器資料寫出到外部記憶體

CPU 將暫存器 AX 的內容搬移到記憶體 OPR 位址內存放

T5：MAR ← PC

T6：PC ← PC + 1

T7：MBR ← OPR

T8：MAR ← MBR

T9：MBR ← AX

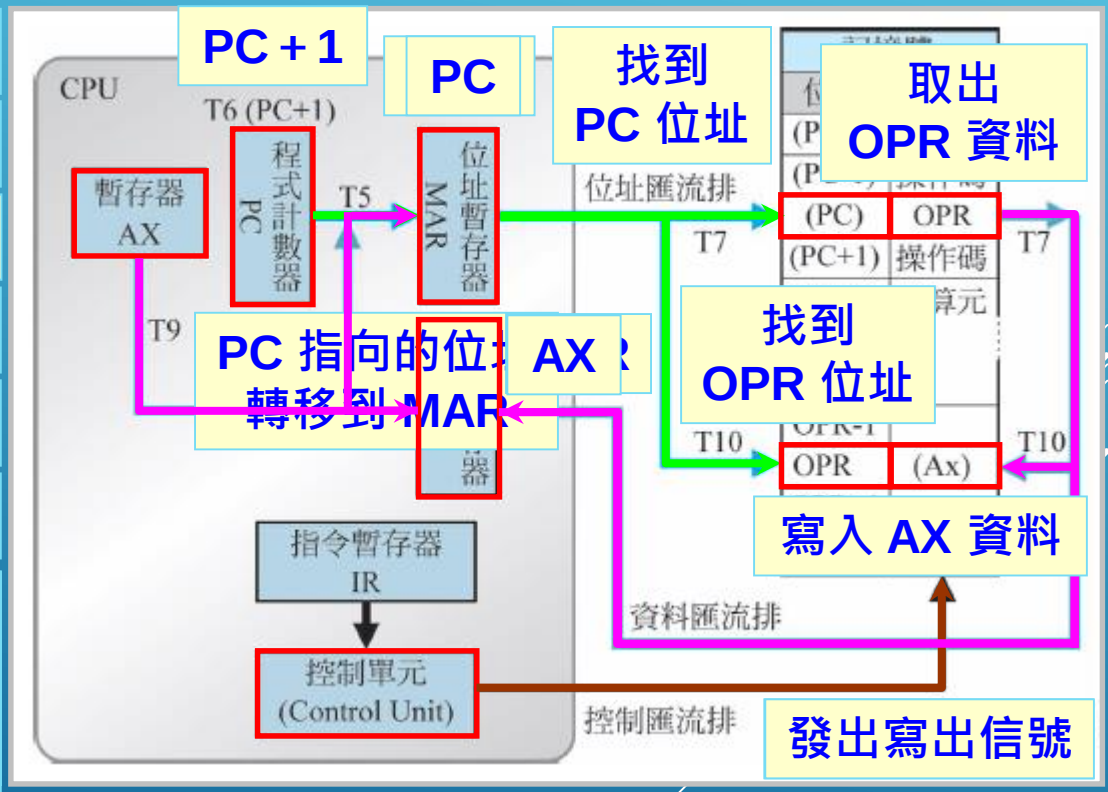
T10：Mem（MAR）← MBR

1、MAR 送出 PC 的位址

MBR 內容 OPR 轉移到 MAR

AX 暫存器的內容轉移到 MBR

寫到 OPR 的位址內



1-4 微處理機指令的提取、解碼及執行

- 學後評量：

D 1、組合語言中何者與程式執行無關？

（A）標記　（B）操作碼　（C）運算元　（D）註解

B 2、微處理機執行一個指令的程序為？

（A）解碼→提取→執行　（B）提取→解碼→執行

（C）組譯→儲存→解碼　（D）組譯→提取→解碼

B 3、CPU 執行完 MOV　Acc，Data 指令後，會做什麼動作？

（A）將 Acc 轉移至 Data 中

（B）將 Data 轉移至 Acc 中

（C）將 Data 所指位址轉移至 MAR 中

（D）將 Data 所指位址的內容轉移至 MBR 中