

Operating Systems

作業系統

HARDWARE
電腦硬體

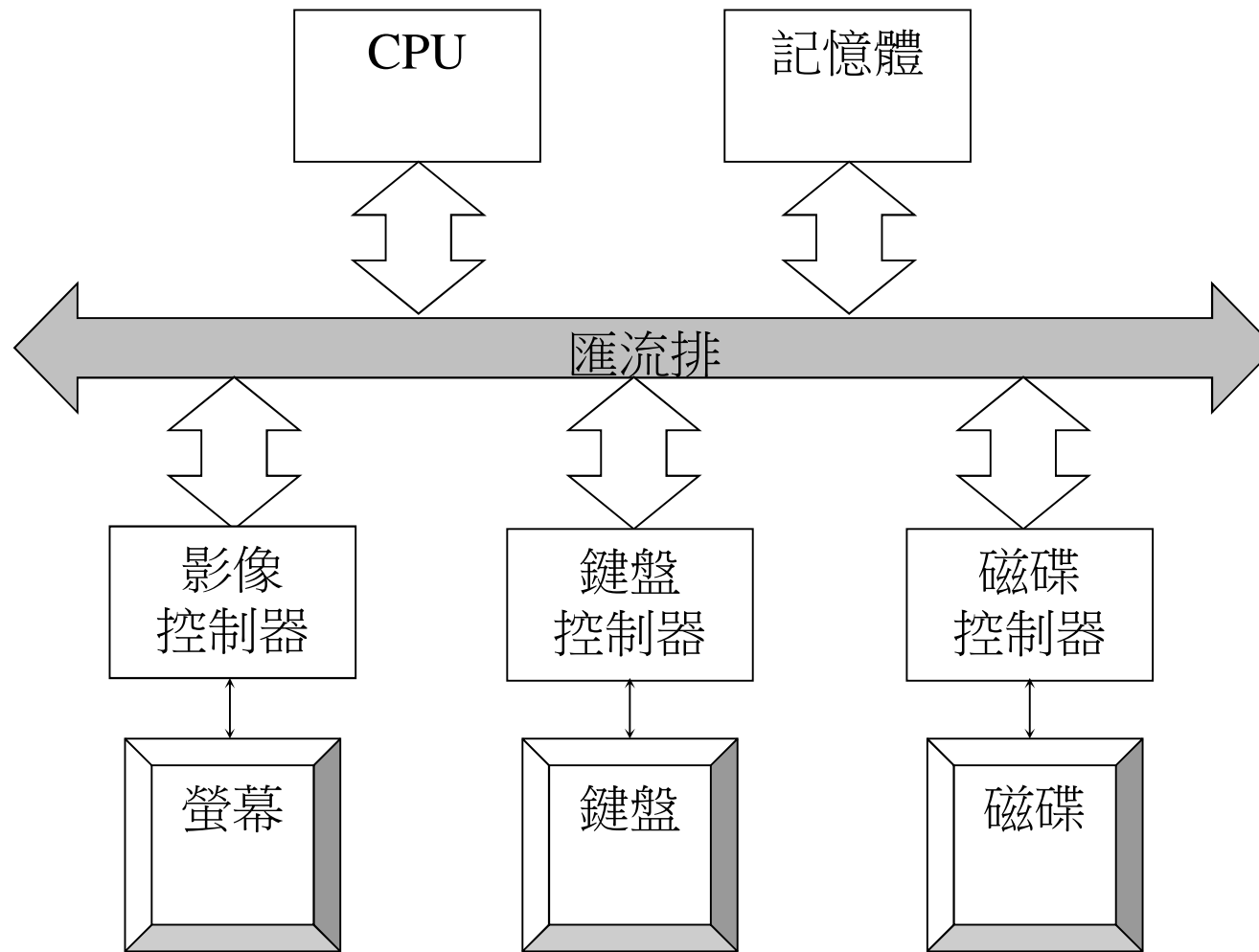
硬體結構

- CPU
 - 暫存器
 - 范紐曼機
- 儲存裝置
- I/O 結構
- 儲存階層
- 硬體保護
- 摘要

中央處理單元 (Central Process Unit, CPU)

- CPU就是中央處理器
 - 是電腦系統的**心臟**
 - 專門處理各種**運算**
 - 負責與週邊設備**溝通**
- 有效利用CPU是提高系統效能的關鍵之一
 - 軟體—**CPU 排程**
 - 硬體—階層式架構

電腦與週邊



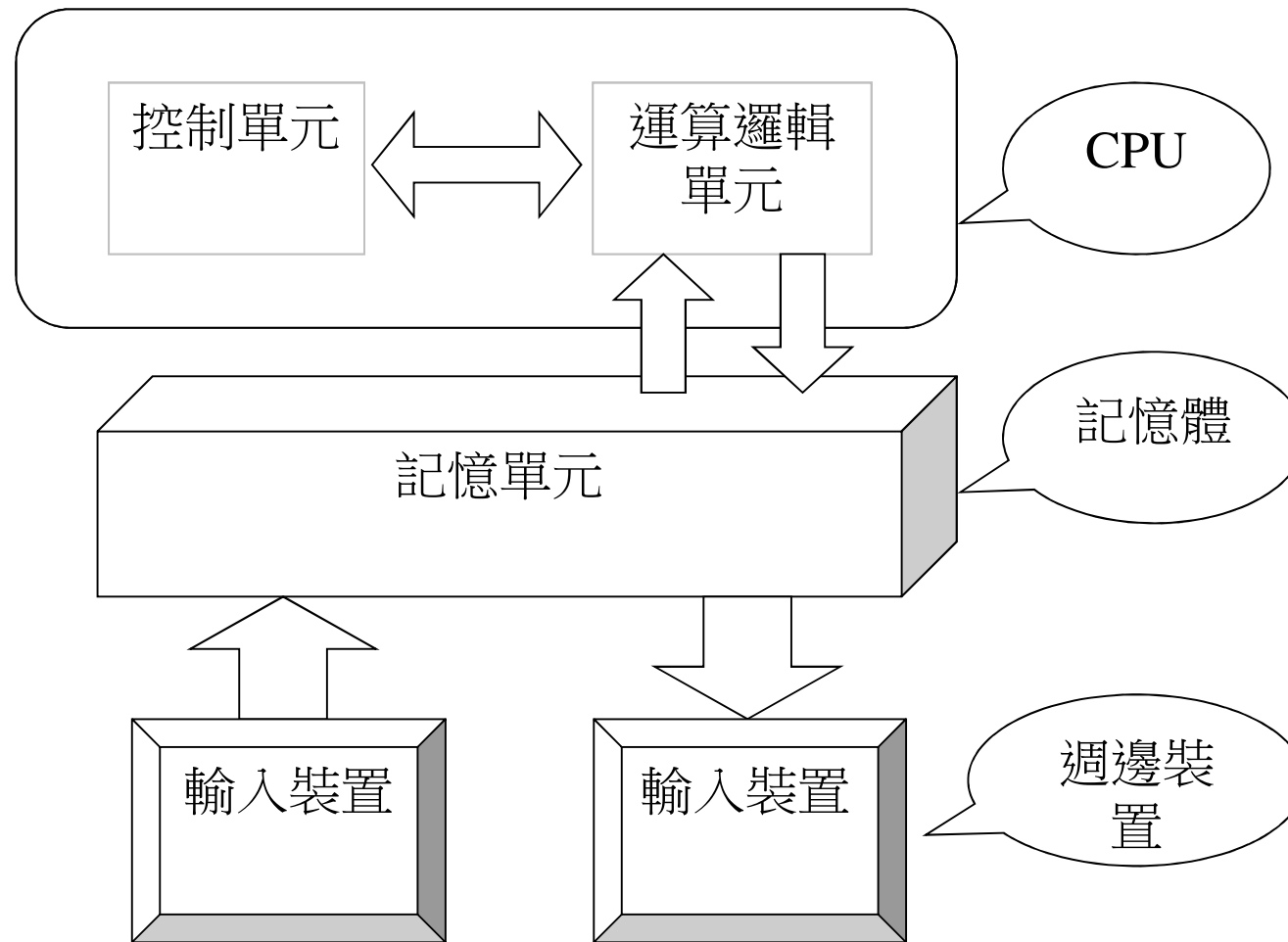
暫存器 (Register)

- CPU 與暫存器
 - CPU 要對任何資料作運算前，必須先將資料載入到**暫存器**中。
 - 暫存器功用與記憶體類似，是**最接近 CPU 的記憶體**。
 - CPU 存取暫存器的速度相當快，所以把資料存到暫存器之後再作運算，可以加快資料處理的速度。
- **特殊用途**暫存器與一般用途暫存器
 - **特殊用途暫存器**：用來**控制硬體**所提供的特殊功能或是有特殊用途的暫存器。
 - 如 Intel 486 相容的架構下有 4 個特殊用途的暫存器 — CR0、CR1、CR2、CR3。
 - **一般用途暫存器**：一般用途的暫存器可以用來存放**資料**或是**記憶體**的位址。
 - 如 Intel 486 相容架構下的 EAX、EBX、ECX、EDX。

范紐曼機 (Von Neumann)

- 范紐曼機是美國普林斯頓大學的范紐曼博士所提出來的電腦系統架構。
- 他將電腦大致分為 5 個單元
 - 算術邏輯單元 (Arithmetic Logic Unit, ALU)
 - 控制單元 (Control Unit)
 - 記憶單元 (Memory)
 - 輸入單元 (Input)
 - 輸出單元 (Output)
- CPU 是由**算術邏輯單元**及**控制單元**組合而成的。

范紐曼機 (續)



硬體結構

- CPU
- 儲存裝置
 - 記憶體
 - 快取
 - 磁碟
 - 磁帶
- I/O 結構
- 儲存階層
- 硬體保護
- 摘要

儲存裝置 (Storage)

- 主記憶體通常會有下列的**缺點**
 - 主記憶體通常不大，無法將所有要執行的程式同時載入。
 - 主記憶體大多是屬於**揮發性記憶體**。
- 因此電腦系統提供**輔助記憶體**來彌補主記憶體的不足。
- 輔助記憶體是用來長期儲存大量的資料，如
 - 磁碟
 - 光碟
 - 磁帶
 - 軟碟
- 不同的裝置其存取速度、容量、體積大小、價格及物理性質也大不相同。

記憶體 (Memory)

- 記憶體大致上分為 ROM 與 RAM 兩種。
- 下面介紹幾種不同類型的記憶體

記憶體名稱	讀取速度	寫入速度	可寫入次數	揮發性記憶體
masked ROM	快	不能	0	否
PROM	快	不能	1	否
EPROM	快	不能	許多	否
Flash	快	慢	10,000	否
EEPROM	慢	慢	1,000,000	否
RAM	非常快	非常快	無限次	是
附註：此處的寫入速度是指在電腦上寫入的時間，以燒錄機寫入皆標為不能				

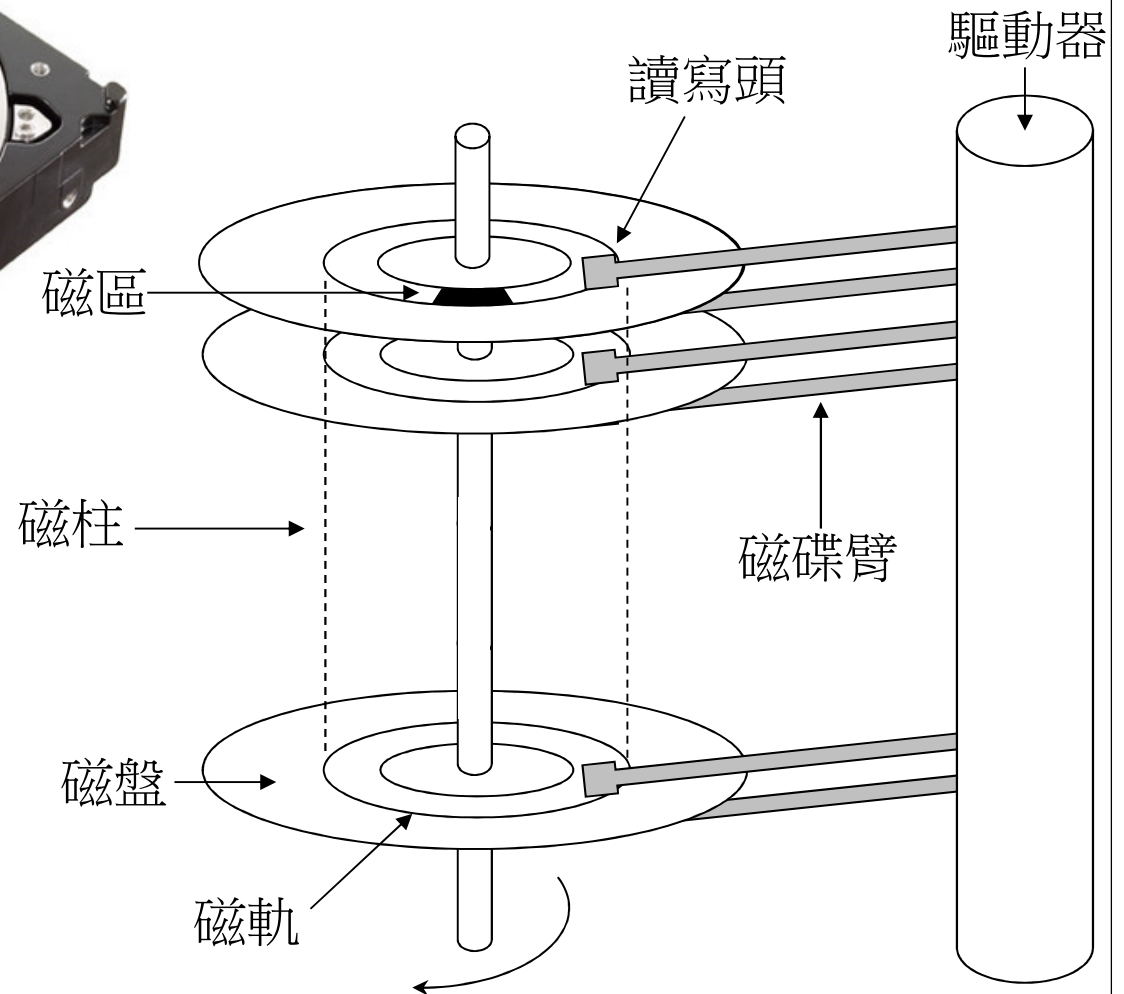
快取 (Cache)

- 快取是**提昇系統效能**的重要機制
 - 存取記憶體必須透過**匯流排**傳送資料到**暫存器**內，所以需要花費許多時間。
 - 快取是加在 CPU 與主記憶體間的快速記憶體，當 CPU 在存取主記憶體的資料時，會複製一份相同的資料到快取之中，等 CPU 下次讀取相同一段記憶體位址的資料時，就能夠直接從快取中讀出。
- 快取的**速度**與**成本**皆高於**主記憶體**。
- 小心選擇快取的大小及好的管理策略，可以讓 80% ~ 99% 的資料都能在快取中找到。

磁碟 (Disk)

- 磁碟是電腦系統中常見的輔助記憶體，由下列機構所組成
 - 磁盤
 - 磁軌
 - 磁區
 - 磁柱
 - 讀寫頭
 - 磁碟臂
 - 驅動器
- 影響磁碟讀取速度的因素
 - 傳送速率
 - 定位時間，或稱為隨機存取時間
 - 搜尋時間
 - 旋轉延遲

磁碟構造圖



磁帶 (Tape)

- 磁帶是早期就有的輔助記憶體
 - 以**循序**的方式來存取資料，較花時間。
 - 可以儲存大量資料。
 - 主要的用途是**備份大型系統的資料**。

硬體結構

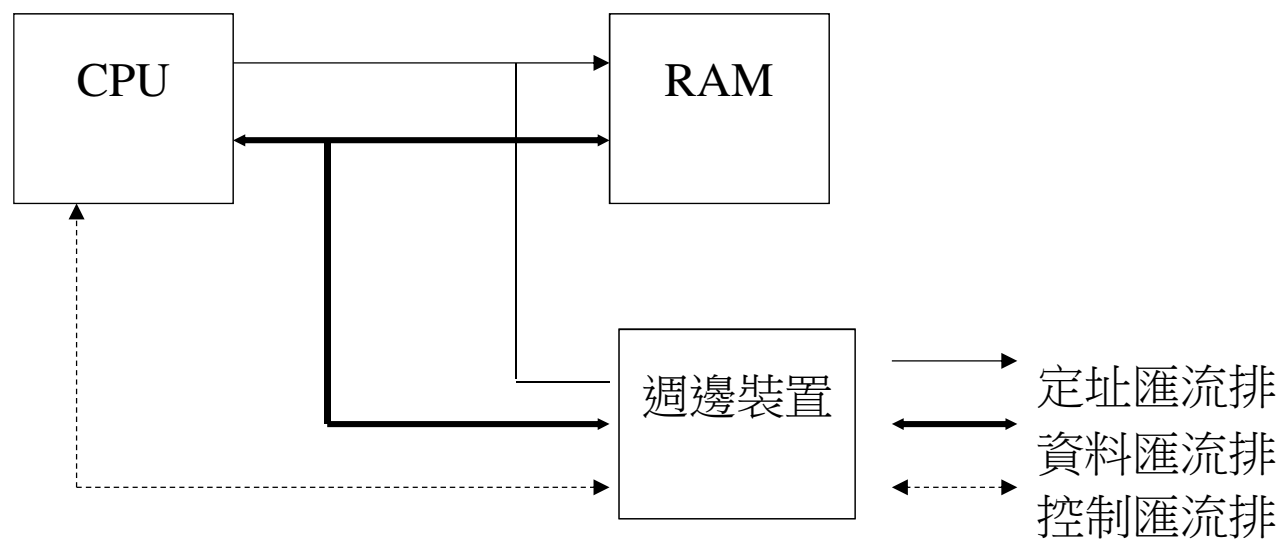
- CPU
- 儲存裝置
- I/O 結構
 - 匯流排
 - 控制器
 - I/O 中斷
 - 直接存取記憶體
- 儲存階層
- 硬體保護
- 摘要

I/O結構

- CPU 和週邊設備有兩種溝通方式
 - I/O 對映
 - CPU 透過特殊的指令控制週邊設備。
 - 記憶體對映 I/O
 - CPU 與**週邊設備**間建立起相同的記憶體位址空間，彼此就以這段位址傳遞指令與資料。

匯流排 (Bus)

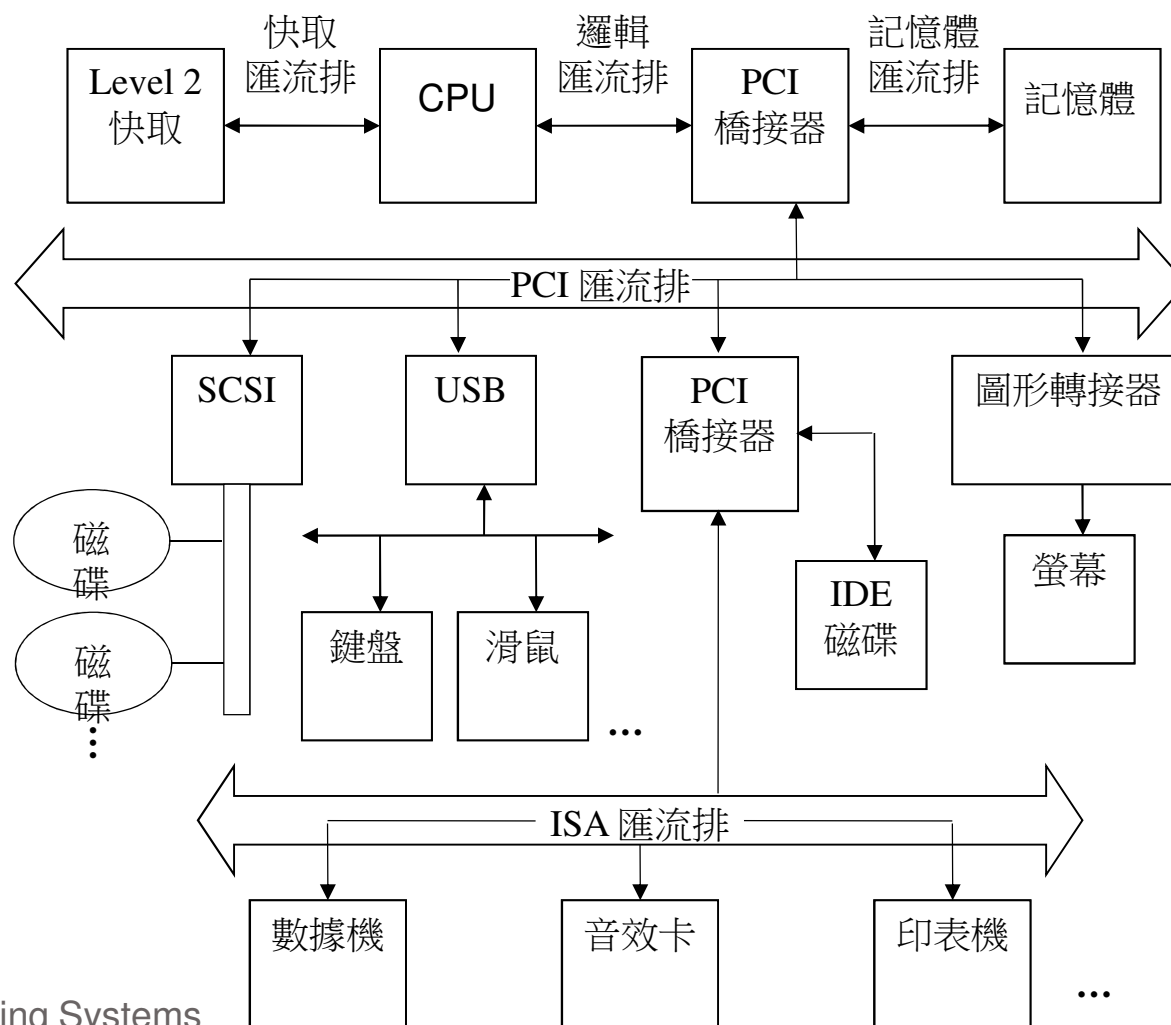
- 匯流排可以分成 3 個主要的部分
 - 定址匯流排
 - 資料匯流排
 - 控制匯流排



匯流排

- 下面是幾種常見的匯流排規格
 - ISA
 - PCI
 - IDE
 - USB
 - SCSI

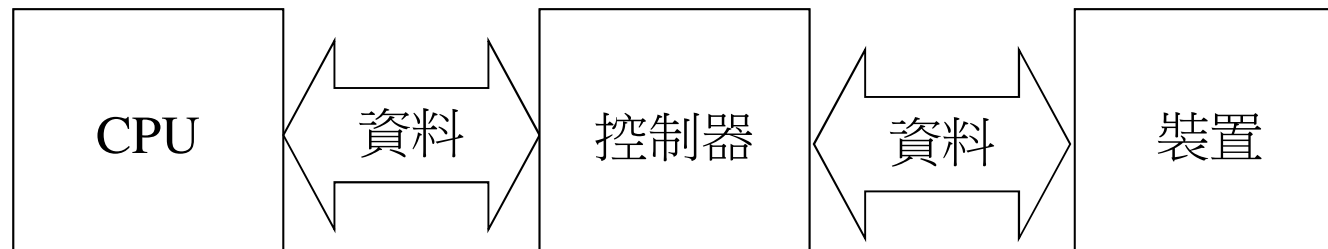
Pentium 系統結構圖



控制器

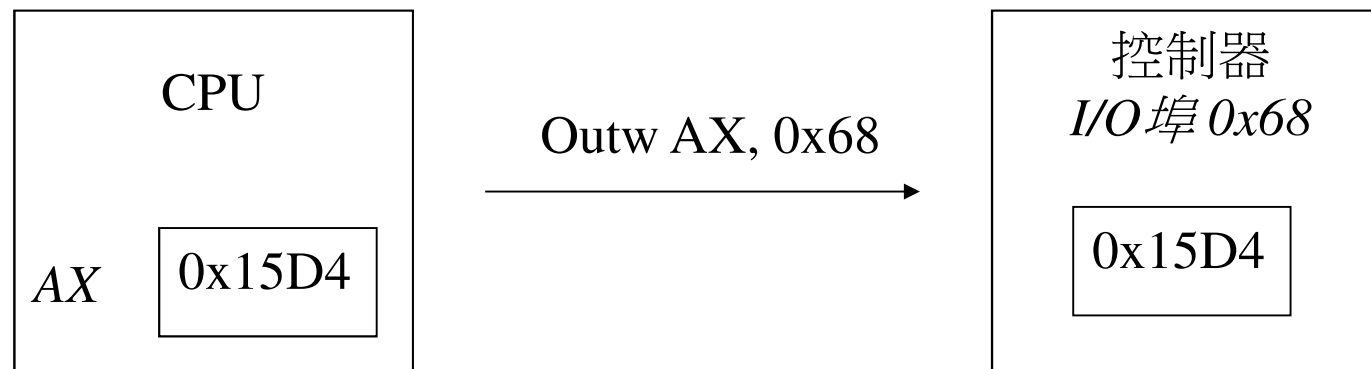
- 一個週邊設備包括了兩個部分
 - 控制器
 - 裝置本身
- 控制器會有一些**暫存器**可以用來控制裝置
 - 驅動程式必須要有能力去存取這些暫存器
 - I/O 對映與記憶體對映 I/O 兩種方式就是將資料或指令寫入這些暫存器內來驅動裝置

控制器是 CPU 與週邊設備溝通的橋樑



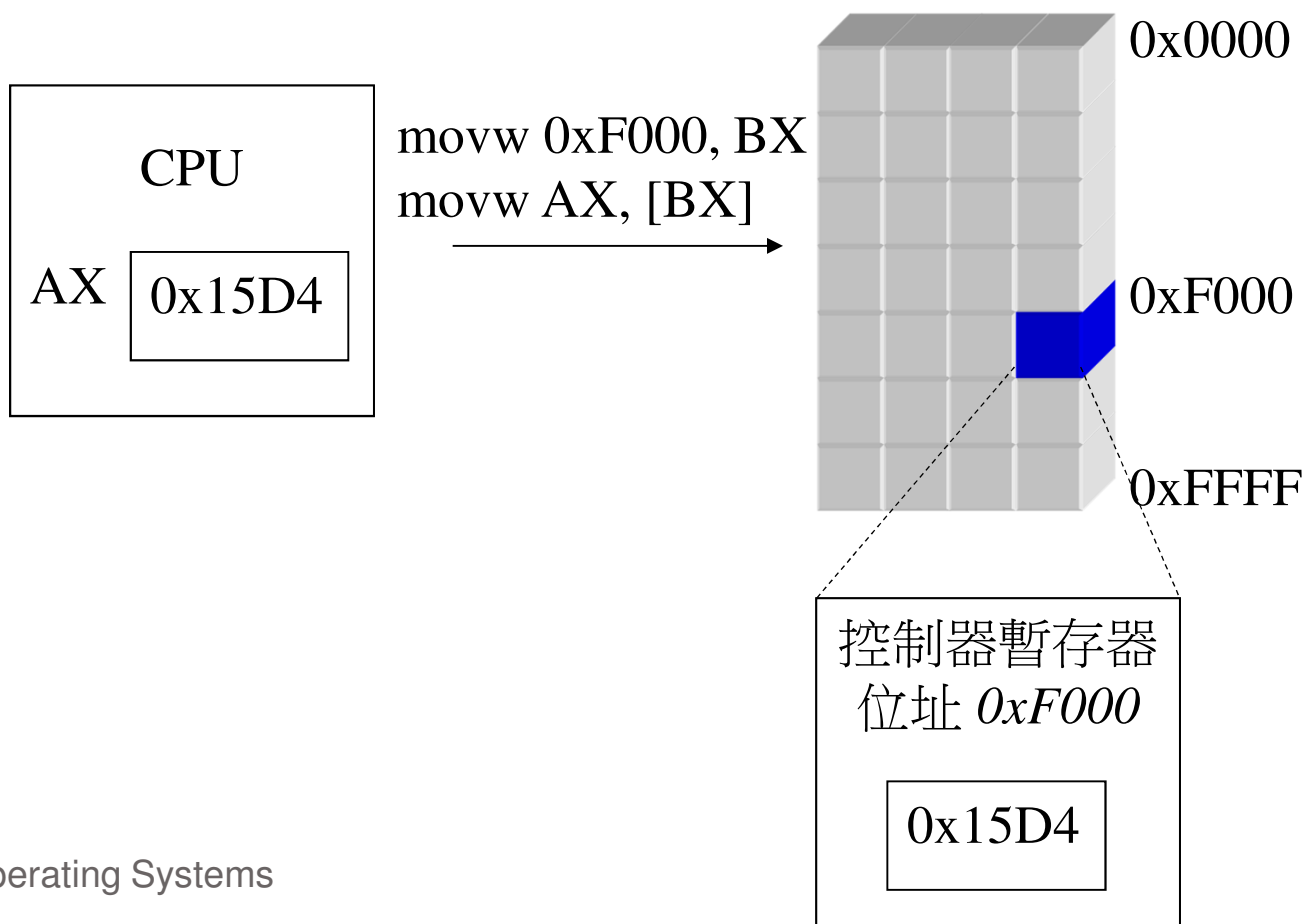
I/O 對映

- 每一個控制器上的暫存器都被給定一個特殊的 I/O 埠。
- Intel 的 IN 跟 OUT 指令可以用來分別**讀出**或**寫入**暫存器的值。



記憶體對映 I/O

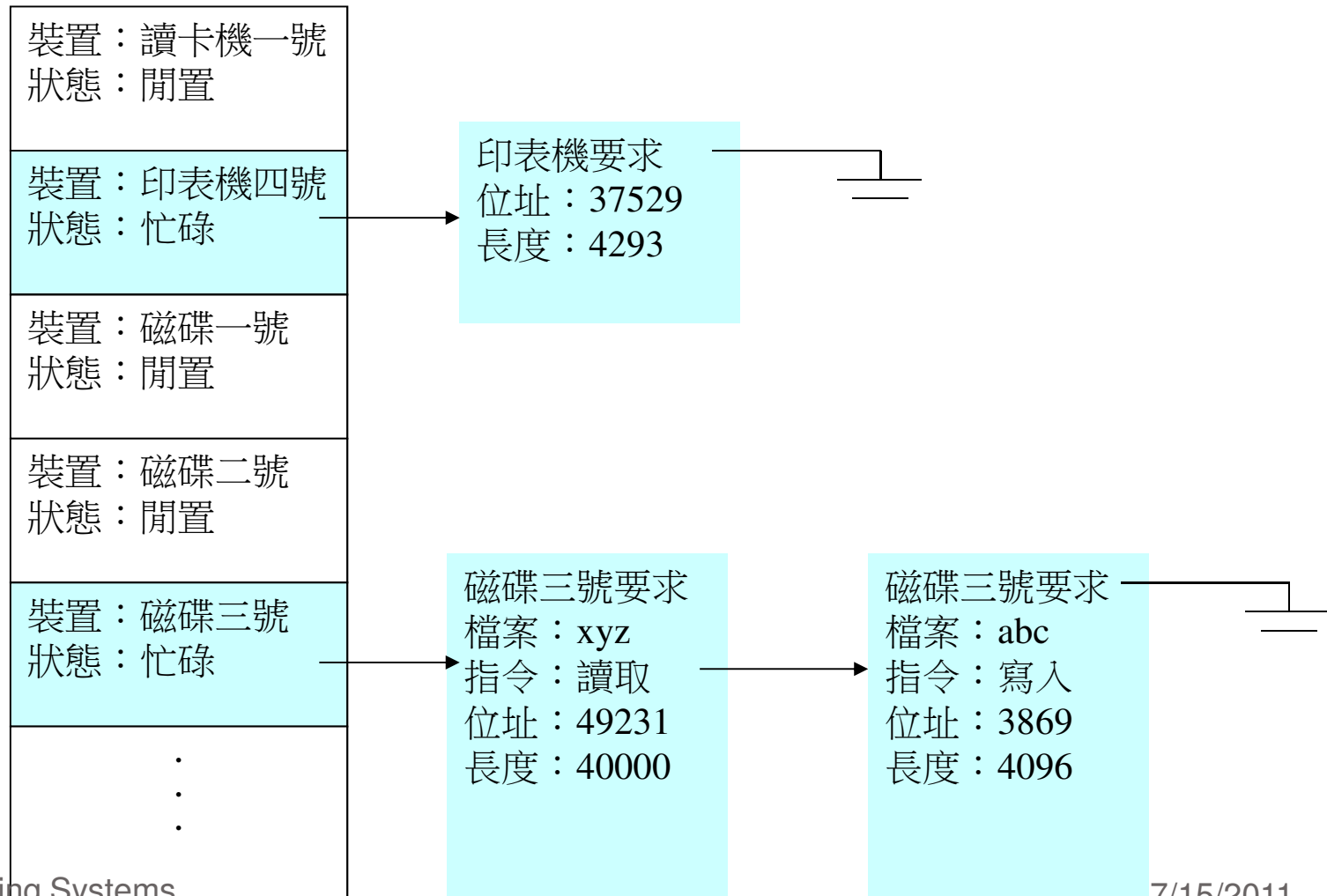
- 記憶體對映 I/O 是將週邊設備的暫存器映對到記憶體位址空間。
- CPU 在存取這些暫存器時，就像是在存取記憶體裡面的值一樣。



I/O 中斷

- 一個 I/O 動作包含了下面的步驟
 - CPU 將資料載入到**裝置控制器**的**暫存器**中。
 - 裝置控制器依照暫存器裡面的值，讓裝置動作。
 - 發出一個**中斷訊號**通知 CPU 工作已經完成了。
- 有兩種等待中斷訊號的方式
 - 同步 I/O
 - 非同步 I/O
 - 需要記錄所有**週邊裝置**的**狀態**，如裝置狀態表。

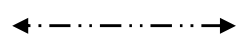
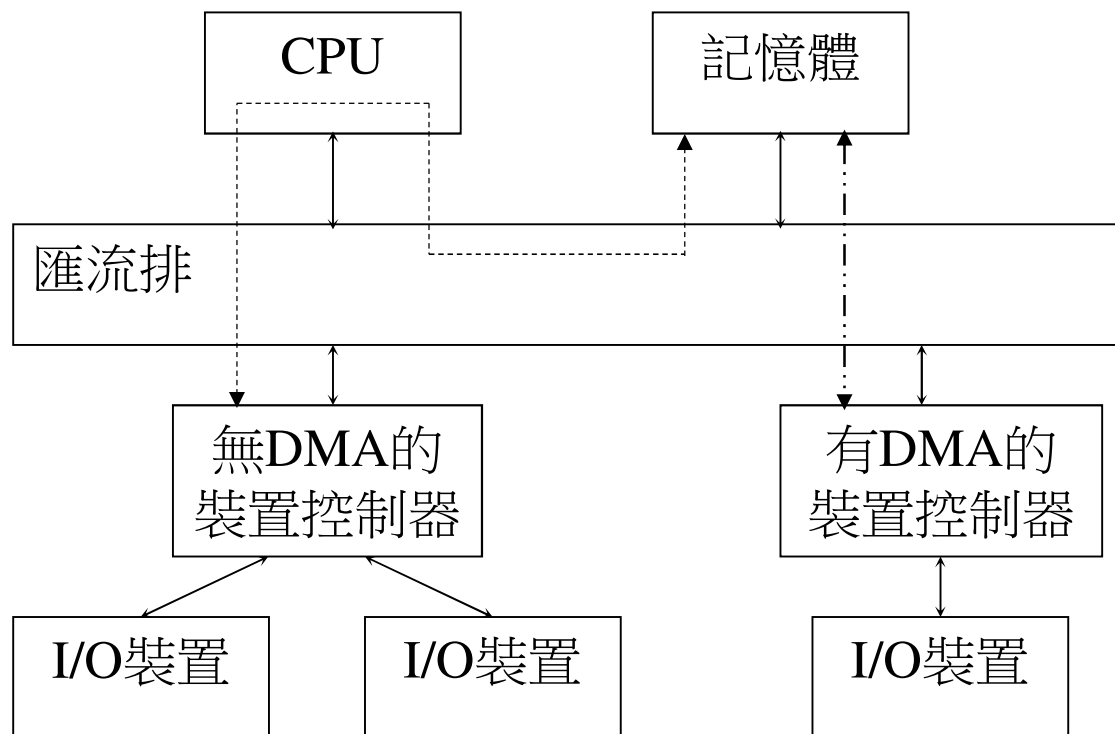
裝置狀態表



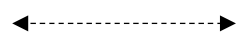
直接記憶體存取 (DMA)

- 沒有 DMA 的控制器若想要存取記憶體中的資料，就必須依賴 CPU 的幫忙。
- DMA 控制器的兩個**特性**
 - 可以**不經由CPU**就能存取到記憶體。
 - 一次可以處理**一整個區塊**的資料。

DMA存取路徑



DMA存取路徑



無DMA存取路徑



資料可傳遞路徑

硬體結構

- CPU
- 儲存裝置
- I/O 結構
- 儲存階層
 - 連貫性
 - 一致性
- 硬體保護
- 摘要

儲存階層

存取時間

1 ns

2 ns

10 ns

10 ms

100 s



暫存器

快取

主記憶體

磁碟

磁帶

容量大小

小於1 KB

1 MB

64-1 GB

5-100 GB

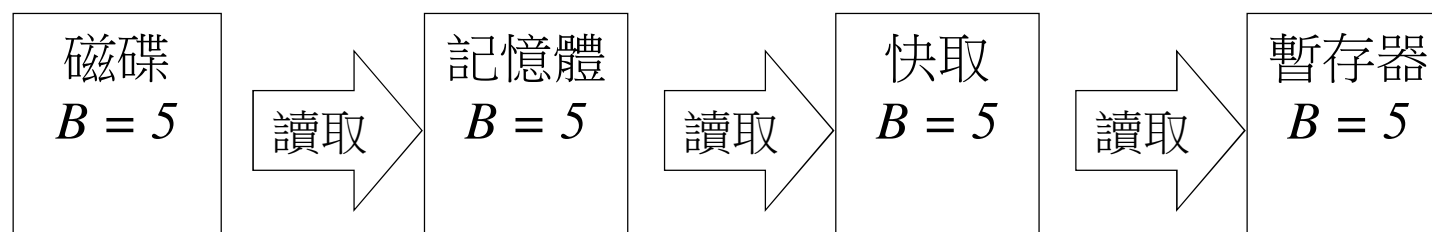
20-120 GB

儲存階層

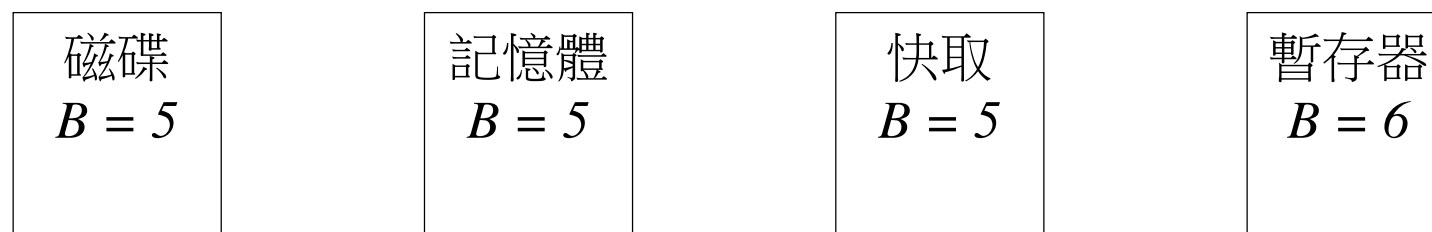
- 在儲存階層中，在愈高層的裝置其價格愈高，但是速度也愈快。
- 從儲存階層往下看，儲存裝置的速度變得愈來愈慢，價格也往下降。
- 以效能來考量
 - 希望所需要用的資料在上層就可以找到
 - 因此出現了**快取機制**及許多**快取演算法**

連貫性

- 以效能為考量，修改上層裝置的資料，不會每次都寫回下層的裝置。



(a) 執行加法前



(b) 執行加法後

一致性

- 一致性是指系統中的工作在存取記憶體的時候，存取的動作不會互相干擾，而影響到資料的正確性。
- 一致性的問題在下面的情況會變得較為複雜
 - 多 CPU 的環境
 - 快取一致性
 - 分散式系統的環境

第二章 硬體結構

- CPU
- 儲存裝置
- I/O 結構
- 儲存階層
- 硬體保護
 - 雙模式運作
 - I/O 保護
 - CPU 保護
 - 記憶體保護
- 摘要

硬體保護

- 單使用者系統的時代
 - 程式設計師全權控制整個系統
- 多工作業系統的時代
 - 作業系統
- 為了提高系統的使用率，作業系統將系統上的資源同時分享給許多程式使用
 - 許多程式的錯誤只能靠硬體來偵測

雙模式運作

- 兩種執行模式供作業系統使用
 - **使用者**模式
 - **系統**模式
- 需要雙模式運作的兩個原因
 - 保護**共享資源**
 - 避免硬體指令造成傷害

I/O 保護

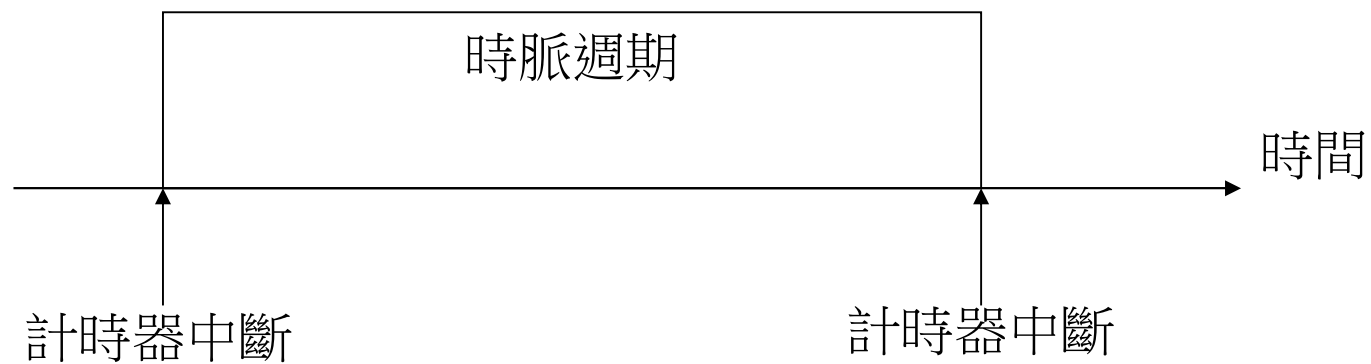
- 避免使用者的程式很有可能發出不合法的 I/O 指令。
- I/O 指令被定義成**特權指令**
 - 使用者不能直接使用 I/O 指令
- 作業系統必須代替使用者處理每個 I/O 指令
 - 需要考量安全問題

CPU 保護

- 如果使用者執行無窮迴圈而拒絕讓出 CPU 的使用權，其他程式就無法被執行。
 - 用**計時器**來保護 CPU
- 倒數計時器
 - 計時器倒數到設定的時間到後發出**中斷**，下次中斷需要由程式重新設定才會再開始進行。
- 週期計時器
 - 會自動重新設定同一段時間後再發生中斷。
- 計時器可以用來實作**分時系統**。

時脈週期

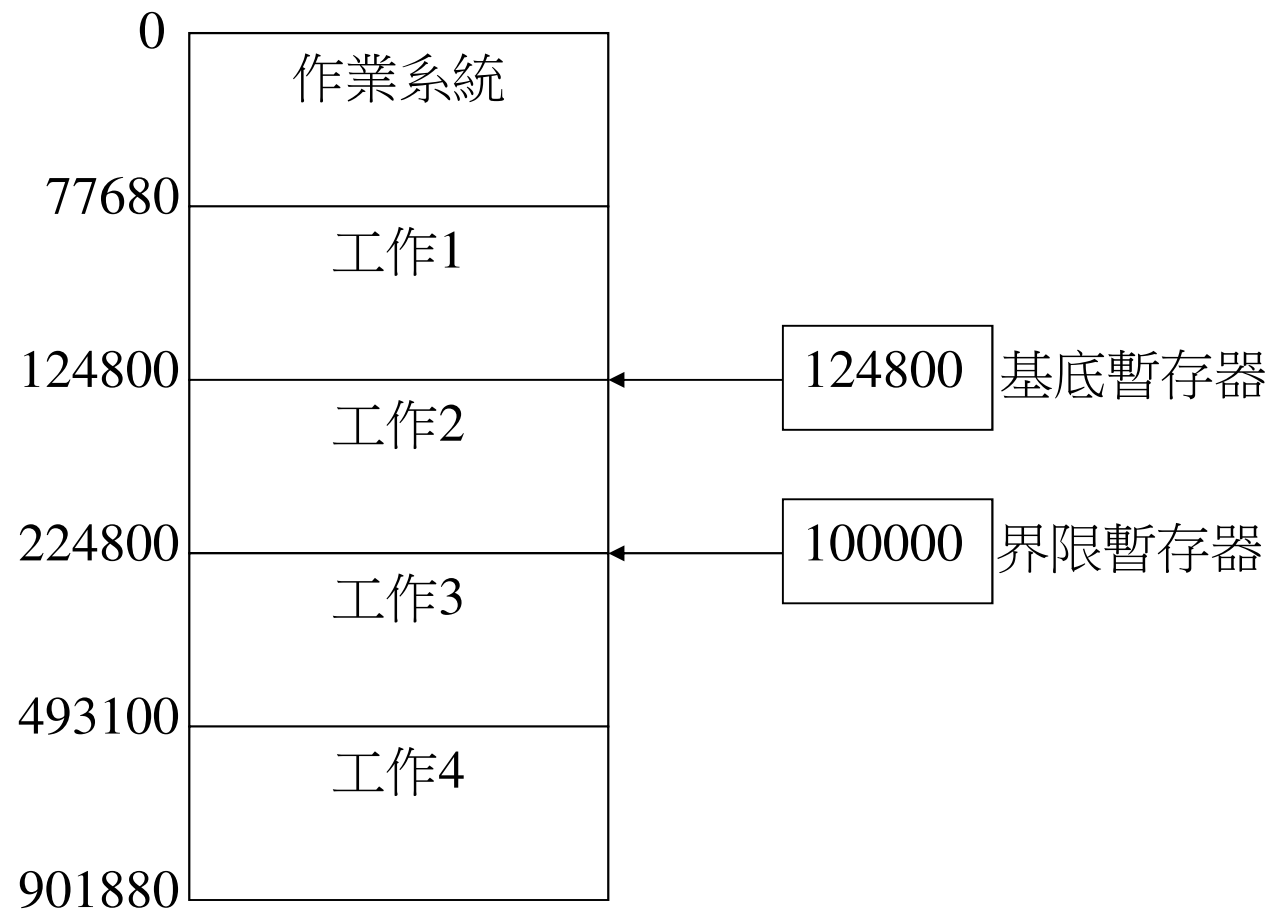
- 週期計時器的週期又稱為時脈週期



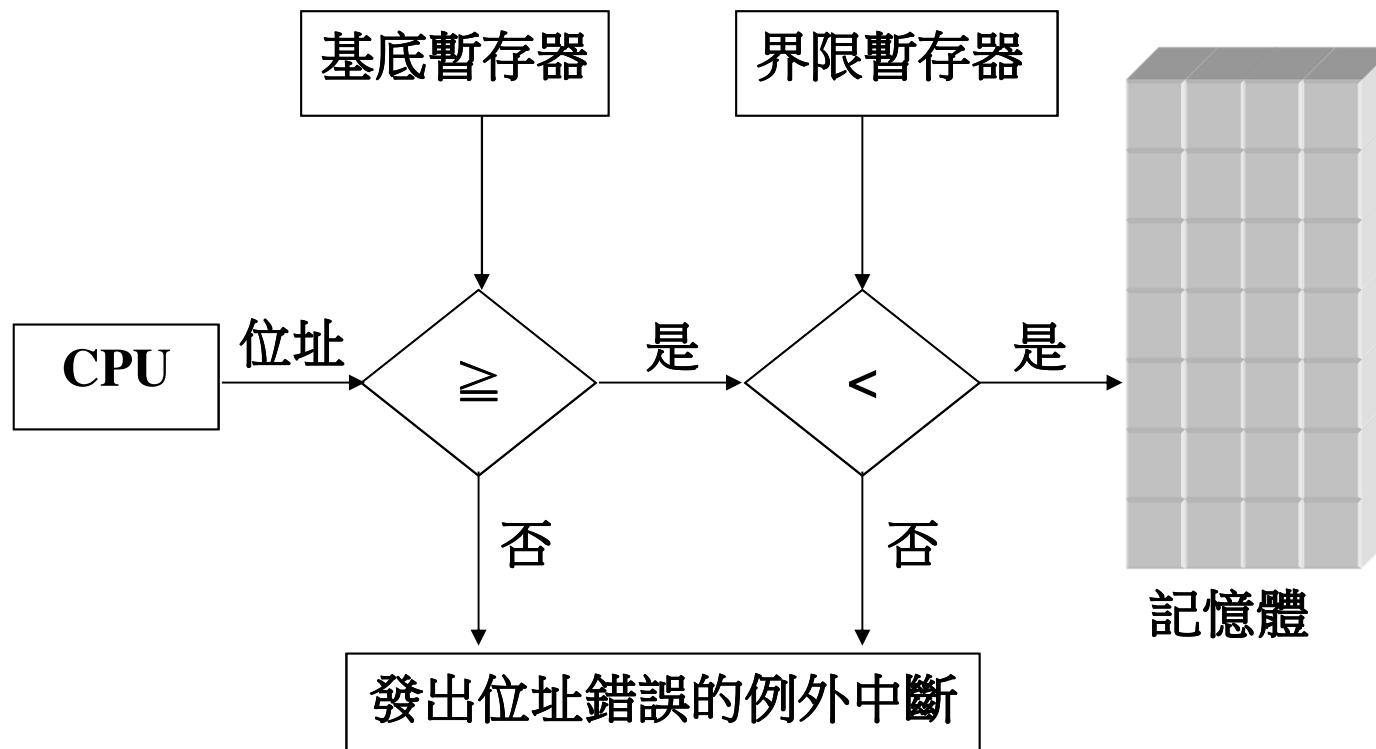
記憶體保護

- MS-DOS 是個很簡單的作業系統
 - 使用者程式可以修作業系統在記憶體中的程式碼或是資料區段。
- 可以用額外的硬體來解決
 - **基底**暫存器
 - **界限**暫存器
- 只有**特權指令**才能夠修改基底暫存器跟界限暫存器

基底暫存器與界限暫存器



基底暫存器與界限暫存器的硬體 記憶體位址偵測



摘要 (1)

- 作業系統與硬體
 - 硬體的架構決定了作業系統各方面實作的方法。
 - 使用非同步 I/O，讓 CPU 與週邊裝置的工作能夠同時進行，以增進效率。
 - CPU 忙碌於執行工作時，DMA 分攤 CPU 傳遞資料到記憶體的工作。
- CPU
 - 執行指令和存取記憶體是 CPU 對外控制的兩個管道。
 - 記憶體對映 I/O 可以讓 CPU 直接控制週邊裝置，否則就必須靠特殊的指令。
 - 主動的方式— CPU 以輪詢的方式查看週邊裝置是否需要服務。
 - 被動的方式—週邊裝置以中斷通知 CPU 目前有工作需要進行。

摘要 (2)

- 儲存階層
 - 在儲存階層的頂端，其執行速度與價格較高
 - 隨著階層往下，儲存裝置的速度與價格也遞減。
 - 除了速度與價格需要考量之外，也要考慮每種儲存裝置的物理特性
 - 揮發性的記憶體在斷電後資料就會消失，但往往其執行速度也比較快。
 - 非揮發性記憶體就比較適合用來儲存資料。

摘要 (3)

- 硬體保護
 - 雙模式運作—避免使用者的程式干擾系統的運作。
 - I/O 保護—特殊指令成為特權指令。
 - CPU 保護—使用計時器防止 CPU 被單一使用者佔據。
 - 記憶體保護—利用特殊的基底暫存器與界限暫存器加以保護記憶體。