Microprogram Controller

CSIE 1-B 110502567 蔡淵丞

1. **微程式控制的原理、定義**

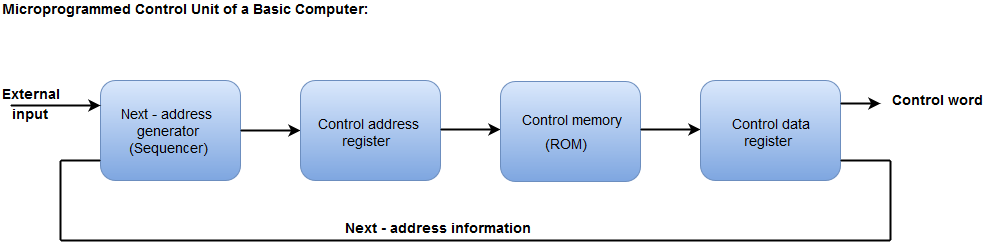
控制器 ( Control Unit ) 在CPU中扮演著指揮官的角色，其功能最主要就是管理及控制計算機內的各個子系統。

不同於硬連線控制器 ( Hardwired CU ) 將程式以組合邏輯電路的形式設定並儲存的方式，一個微程式控制單元 ( Microprogrammed CU) 將所有的微指令儲存於內部存儲器中，在執行時取出各種指令來產生所需要的控制信號。

微程式控制的優勢則在於其硬體結構只需設計一次，若要進行不同操作只需更改微指令集的順序，讓軟體工程師能專注在指令操作流程上。這就是程式設計。

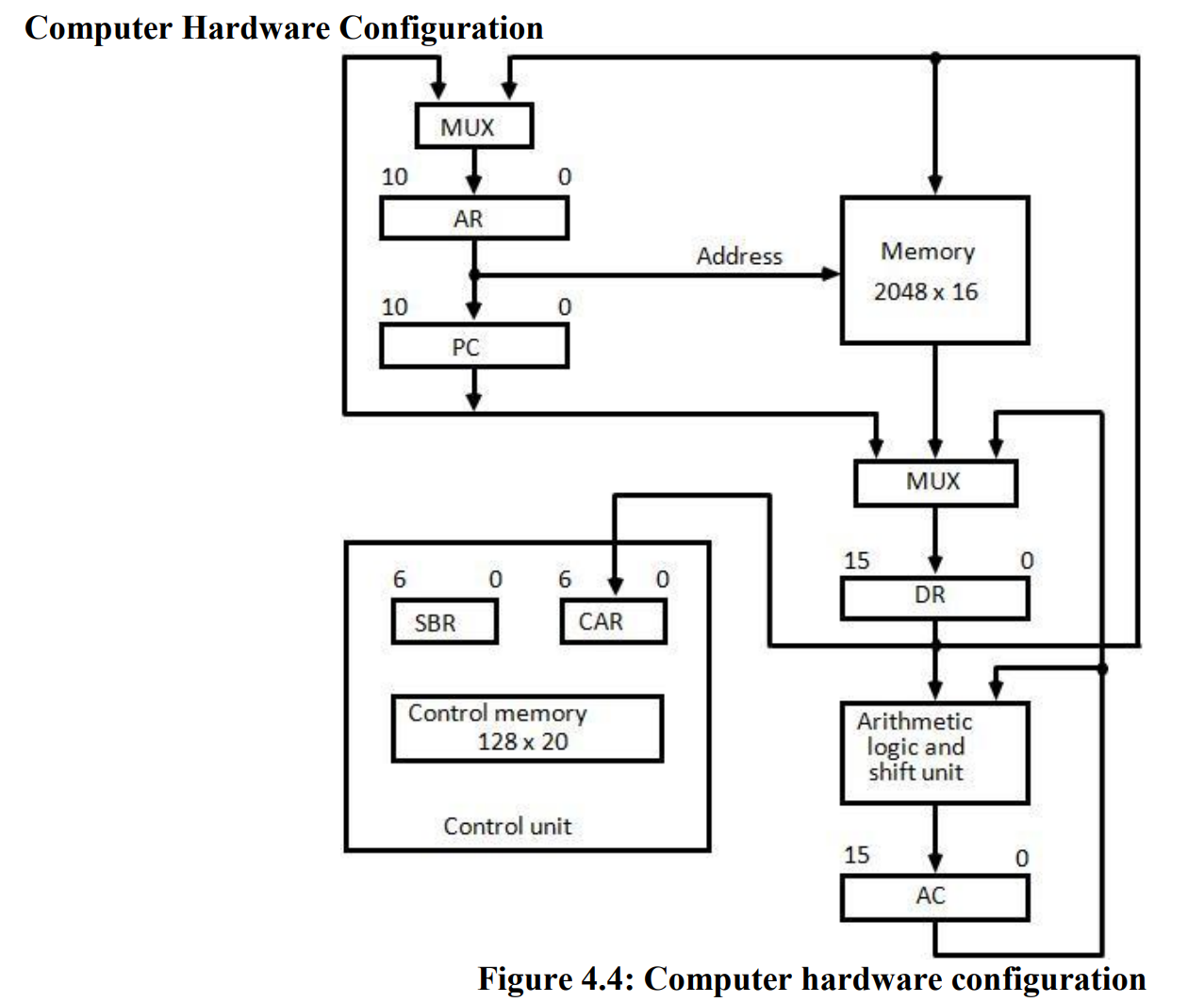
1. **架構圖與架構說明**

**微指令 (Microinstruction)** 是由若干微操作 (Microoperation)組成的一系列操作，每一條微指令都以特定位址儲存於程式記憶體中。



* + 此循環會不斷的取得下一個微指令的位址並輸出其對應的微指令
  + 程式記憶體 ( Control Memory ) 是一個唯讀的記憶體，其中儲存了所有的微指令
  + 在微指令執行的同時，下一條位址會被讀取進位址暫存器 ( Control address register ) 中

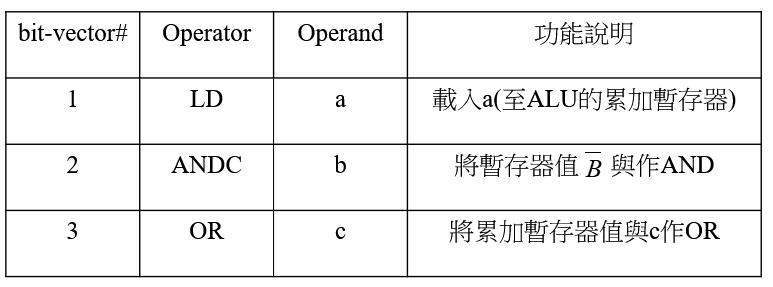
每當程式運行時，微程式控制器就以微指令的形式控制算術邏輯單元、輸入輸出匯流等，藉此達到以軟體控制硬體。



* + 記憶體
    1. 主記憶體 ( Main Memory ) 儲存程式跟資料
    2. 程式記憶體 ( Control Memory ) 儲存微程式
  + 暫存器
    1. 累加暫存器 ( Accumulator ) 儲存ALU的運算結果
    2. 程式計數器 ( Program Counter ) 儲存將要讀取程式的記憶體位址
    3. 位址暫存器 ( Address Register ) 儲存正在讀取程式的記憶體位址
    4. 數據暫存器 ( Data Register ) 儲存從記憶體取出的指令
  + 多工器選擇不同的訊號輸入至ALU
  + ALU 執行微操作並將結果存於AC

1. **軟、硬體運作機制**

假設我們現在要運行一個簡單的邏輯運算 : F = AB’ + C，此程式會先大略分成三步驟如下 :



步驟一

PC指向bit-vector#1。

a 的位址會被輸入至MUX中，MUX將a 輸出至ALU的B輸入端。

LD 這個操作指令會從ROM中輸出至ALU作為選擇訊號。

ALU將a存入AC。

PC指向bit-vector#2。

步驟二

b的位址會被輸入至MUX中，MUX將b輸出至ALU 的B輸入端。

ANDC 這個操作指令會從ROM中輸出至ALU作為選擇訊號。

ALU 將AB’存入AC。

PC指向bit-vector#3。

步驟三

c的位址會被輸入至MUX中，MUX將c輸出至ALU 的B輸入端。

OR 這個操作指令會從ROM中輸出至ALU作為選擇訊號。

ALU 將AB’輸出。

以上這些步驟不同的點只在於輸入輸出的訊號及順序，而繁瑣且重複的低階機械操作則由內部存儲器儲存，我們只需思考高階的邏輯運算過程即可。

微程式控制器簡單來說就是一個CPU內部的小型CPU，他為我們實現了真正意義上的軟體工程，讓控制電路不再只是連接邏輯門。透過把繁複的步驟模組化(微操作->微指令->微程式)讓程式設計師能夠愈來愈有效率的設計更龐大的系統，微程式在電腦科學的地位不可抹滅。

1. **Reference**

[Control unit - Wikipedia](https://en.wikipedia.org/wiki/Control_unit)

[Design of Control Unit | Computer Organization and Architecture Tutorial - javatpoint](https://www.javatpoint.com/design-of-control-unit)

[挑战408——组成原理（25）——微程序控制器 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/58841915)