

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。

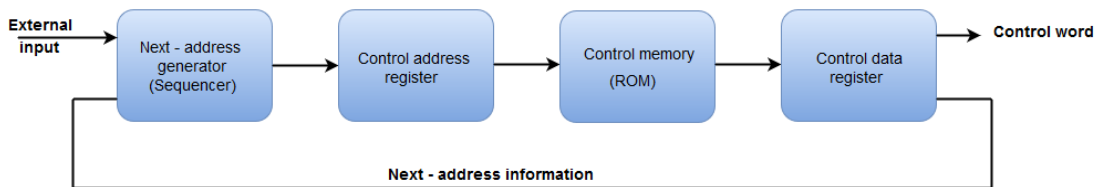
不同於硬連線控制器 (Hardwired CU) 將程式以組合邏輯電路的形式設定並儲存的方式，一個微程式控制單元 (Microprogrammed CU) 將所有的微指令儲存於內部存储器中，在執行時取出各種指令來產生所需要的控制信號。

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。

2. 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。

Microprogrammed Control Unit of a Basic Computer:



- 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。

Computer Hardware Configuration

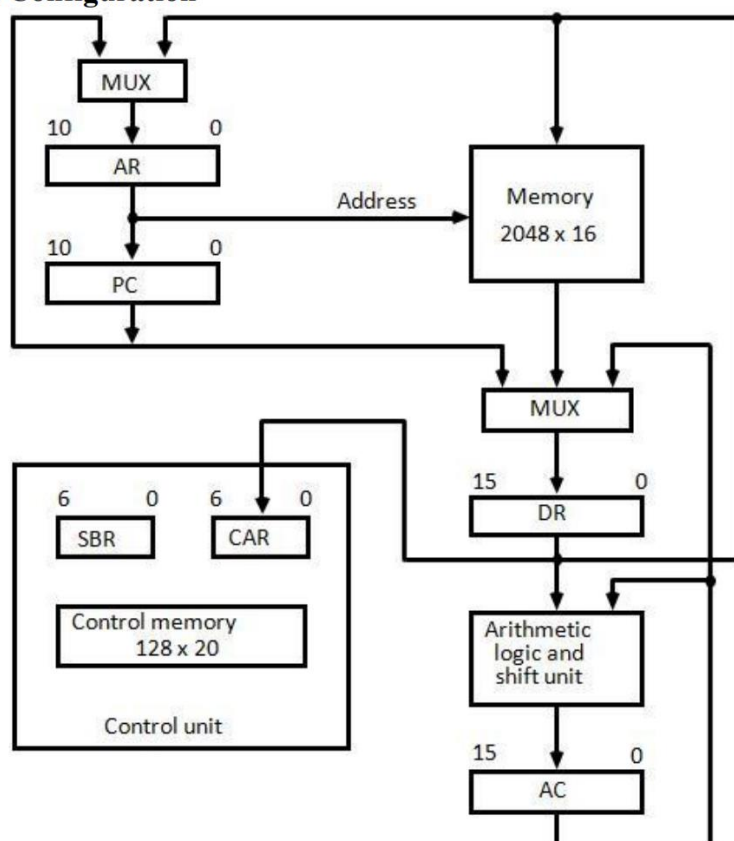


Figure 4.4: Computer hardware configuration

- 記憶體
 - i. 主記憶體 (Main Memory) 儲存程式跟資料
 - ii. 程式記憶體 (Control Memory) 儲存微程式
- 暫存器
 - i. 累加暫存器 (Accumulator) 儲存 ALU 的運算結果
 - ii. 程式計數器 (Program Counter) 儲存將要讀取程式的記憶體位址
 - iii. 位址暫存器 (Address Register) 儲存正在讀取程式的記憶體位址
 - iv. 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

3. 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個步驟如下：

bit-vector#	Operator	Operand	功能說明
1	LD	a	載入a(至ALU的累加暫存器)
2	ANDC	b	將暫存器值 \overline{B} 與作AND
3	OR	c	將累加暫存器值與c作OR

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设计更龐大的系統，微程式在電腦科學的地位不可抹滅。

4. Reference

[Control unit - Wikipedia](#)

[Design of Control Unit | Computer Organization and Architecture Tutorial - javatpoint](#)

[挑战 408——组成原理（25）——微程序控制器 - 知乎 \(zhihu.com\)](#)