

## Chapter 3 微處理機軟體發展流程

### ➤ 選擇題 - (每題 2 分，共 50 分)

- 3-1 ( B ) 1. 指令計數器 (Program Counter, PC) 是一個暫存器，下列敘述，何者正確？
- (A) 計算已執行過的程式指令的數目  
 (B) 記載下一個執行指令的位址  
 (C) 記載程式中可被執行的數目  
 (D) 計數同一種運算的指令數目。
- ( A ) 2. 下列暫存器何者儲存 ALU 運算後的狀態，例如進位、溢位、零值、負值等？
- (A) 旗標 (狀態) 暫存器 (B) 指令暫存器  
 (C) 程式計數器 (D) 堆疊。
- ( B ) 3. 微電腦系統中，當呼叫副程式、產生中斷等狀況下，會把目前的程式位址及旗號存起來，等回歸時可再取回，此時程式計數器的值和旗號是存於何處？
- (A) 控制單元 (B) 堆疊 (C) 通用暫存器 (D) 虛擬記憶體。
- 3-2 ( D ) 4. 使用何種定址法，獲得運算元的速度最快？
- (A) 直接索引定址法 (B) 直接定址法  
 (C) 索引定址法 (D) 立即定址法。
- ( B ) 5. 某一種使用單位址指令的電腦，其指令結構只有二個欄位：運算碼和位址碼，每一指令佔用 24 bits，若此電腦之指令組共有 120 種運算，而指令是使用直接定址模式，則可定址的記憶體空間大小 (單位：byte) 最接近於 (A) 64K (B) 128K (C) 1M (D) 16M。
- ⊙ 單位址指令表示只有 1 個運算碼欄位與 1 個運算元欄位。共有  $120 < 2^7$  個指令，所以需要 7 位元運算碼欄位。剩下  $(24 - 7) = 17\text{bits}$  為運算元欄位，可定址  $2^{17} = 128\text{K}$ 。故選擇 (B)。
- ( A ) 6. CPU 具有各種不同拿取運算元的方法，稱為
- (A) 定址模式 (addressing mode)  
 (B) 指令類型 (instrument type)  
 (C) 提取運算元  
 (D) 指令執行



( B ) 11. 在 AX 暫存器中 MSB 為位元 15，而 LSB 為位元 0，請問要將 AX 的位元 10、位元 3 及位元 2 的值設為 1 而不改變其它位元的值時，應該使用下列何種指令？

- (A) AND AX, 040CH                      (B) OR AX, 040CH  
(C) XOR AX, 040CH                      (D) MOV AX, 040CH。

⊙ 解 OR 運算具有 SET (設定) 功能，與 0000 0100 0000 1100 進行 OR 運算，即可不改變其他位元值，並設定位元 10、位元 3 及位元 2 的值為 1。

( B ) 12. 執行下列 80×86 指令後，AX 的值應是多少？

- (A) 32 H (B) 26 H (C) 78 H (D) 85 H。

MOV AX, 13H

MOV CX, AX

ADD AX, CX

⊙ 解 程式執行分析過程如下表所示，

MOV AX, 13H	AX = 13H
MOV CX, AX	CX = 13H
ADD AX, CX	AX = AX + CX = 26H

( A ) 13. 若暫存器 A, B, C 初始值分別為 30, 20, 10，今依序執行 PUSH A, PUSH B, PUSH C, POP A, POP B, POP C 後，暫存器 A, B, C 的內容是什麼？

- (A) 10, 20, 30                      (B) 20, 10, 30  
(C) 20, 30, 10                      (D) 30, 10, 20。

( C ) 14. 80×86 執行 8 位元的相加運算，下列哪一種運算結果將使溢位旗標 ( overflow flag, OF ) 被設定成 1 ？

- (A) 7FH + FFH                      (B) 7FH + B9H  
(C) 72H + 55H                      (D) BFH + CFH。

⊙ 解 溢位旗標 = OF = CF7 ⊕ CF6，其中 CF7 為 bit7 是否進位？CF6 為 bit6 是否進位？其中只有 (C) 72H + 55H = 0111 0010 + 0101 0101 之 CF6 = 1、CF7 = 0，所以 OF 為 1，產生溢位了！

( C ) 15. 兩個 8-bit 暫存器：AL 與 BL，其內容分別為 AL = 3BH、BL = 0FH。若將這兩個暫存器進行 NAND 之邏輯處理後，則其結果為何？ (A) C0H (B) 3BH (C) F4H (D) 0BH。

( B ) 16. 如果將一個 2 進位的數同時放入暫存器 A 及暫存器 B，現將 A 左移一位後，再加上 B 的內容，其結果等於原數乘多少？（不考慮溢位） (A) 2 (B) 3 (C) 4 (D) 5。

( B ) 17. 關於精簡指令集微處理機 (RISC CPU) 以及複雜指令集微處理機 (CISC CPU) 之敘述，下列何者不正確？

(A) 一般而言，RISC CPU 之定址模式遠比 CISC CPU 之定址模式少

(B) RISC CPU 之指令編碼長度不固定，導致解碼器之設計複雜

(C) 對於 CISC CPU 而言，不同的指令，需要不同的時脈週期來完成

(D) 相較於 RISC CPU，CISC CPU 在實作管線化資料路徑時有較高的設計困難度。

解 RISC 指令編碼長度固定，所以指令解碼電路較單純。

3-4 ( A ) 18. 下列程式語言，何者屬低階語言？

(A) 組合語言 (B) BASIC (C) Visual Basic (D) C++。

( C ) 19. 下列有關組合語言假指令 (pseudo instruction) 的敘述，何者有誤？

(A) 可以增加程式的清楚度

(B) 可以設定組譯器組譯時的選項

(C) 假指令本身有相對應的機器碼 (machine code)

(D) EQU 為假指令。

( B ) 20. 把組合語言程式翻譯成二進制機械語言的程式稱為

(A) 編譯器 (compiler) (B) 組譯器 (assembler)

(C) 直譯器 (interpreter) (D) 函式庫 (library)。

( C ) 21. 「編譯器」主要的功能為何？

(A) 將組合語言程式碼轉譯成機器碼

(B) 將程式重新定址

(C) 將高階語言程式碼轉譯成機器碼

(D) 連結互相呼叫的程式。

解 將組合語言程式碼轉譯成機器碼者，稱為組譯器 (Assembler) 將高階語言程式碼轉譯成機器碼者，稱為編譯器 (Compiler)。

( B ) 22. 組合語言指令格式中，下列哪一個欄位不可省略？

(A) 標記欄 (B) 指令碼欄 (C) 注解元欄 (D) 全都不可省略。

3-5 ( D ) 23. 作業系統的功能不包含

- (A) 分配及管理系統資源
- (B) 建立使用者介面
- (C) 執行應用軟體，並提供執行時期所需之服務
- (D) 文書處理。

( C ) 24. 下列有關「載入程式 (Loader)」的敘述中，何者是正確的？

- (A) 可檢查原始程式是否有語法 (Syntax) 上的錯誤
- (B) 將原始程式編譯成目的程式
- (C) 將目的程式載入主記憶體中
- (D) 執行目的程式。

解 (A) 檢查原始程式是否有語法錯誤的是「編譯程式」、  
(B) 將原始程式編譯成目的程式的是「編譯程式」。

( A ) 25. 下列何者是電腦系統架構的最底層？

- (A) 硬體 (B) 系統程式 (C) 應用程式 (D) 作業系統。

解 電腦系統最底層為硬體，但沒有作業系統也無法運作。層級為 硬體 > 作業程式 > 應用程式。系統程式包括作業系統、公用程式和程式開發工具。

### ➤ 問答題 - (每題 10 分，共 50 分)

1. 請依據圖 3-1 微處理機架構簡圖，描述指令週期的處理步驟。

解 指令週期可概分為提取週期與執行週期，處理步驟如下。

#### 1. 提取週期

步驟	工作說明
①	PC 存放「下一個 CPU 要執行的指令位址」，當提取週期開始時，會先將 PC 內容放置 MAR，作為取得下一個要執行指令的記憶體位址。
②	控制單元產生記憶體讀取信號，讀取 MAR 所指位置內容至 MDR。
③	指令格式中，運算碼部分由 MDR 輸入 IR
④	IR 內容送入 ID 進行指令解碼，以解碼指令的工作，解碼出數個微碼，再送至控制信號產生電路。
⑤	控制信號產生電路產生一連串可達成指令功能的控制信號，負責執行指令提的所有控制動作。
⑥	將 PC 加 1，指向下一個 CPU 要執行的指令位置。

## 2. 執行週期

步驟	工作說明
①	提取在記憶體內的運算元，指令需要由記憶體提取運算元內容，由控制單元計算運算元有效位址，將位址放置 MAR。
②	控制信號產生電路產生記憶體讀取信號，讀取 MAR 所指位置內容至 MDR。
③	MDR 中的資料（運算元）載入 ALU 中的累積器 A 或 B 暫存器，以進行運算動作。控制信號完成資料傳輸及必要條件
④	ALU 執行算術、邏輯運算
⑤	指令執行結果可能放在暫存器或回存記憶體。如需要回存運算結果到記憶體，則將運算結果放至 MDR。
⑥	配合 MAR 指定位置與 MDR 回存資料，產生寫入信號，完成回存結果。

## 2. 請簡述常見的定址模式與特色。

解 常見的定址模式與特色整理如下表所示。

定址模式	特色與說明
立即定址	運算元在指令中，如指令中的 3CH。沒有記憶體存取，速度較快，但是運算元長度受限。
直接定址	運算元在所指的記憶體位址中。因指令長度固定，位址空間（長度）受限
間接定址	運算元有效位址在所指記憶體位址中。可使用位址空間大，但多次記憶體存取速度較慢
暫存器定址	雖然無記憶體存取，但位址空間受限
暫存器間接定址	位址空間大，但額外記憶體存取速度較慢
位移定址	存取位址有彈性，但較複雜。又可分為相對定址、基底暫存器定址、索引定址三種方式
堆疊定址	是一種隱含定址的操作方式，指令中沒有記憶體位址

## 3. 試著比較 CISC 與 RISC 兩種不同指令集電腦的不同。

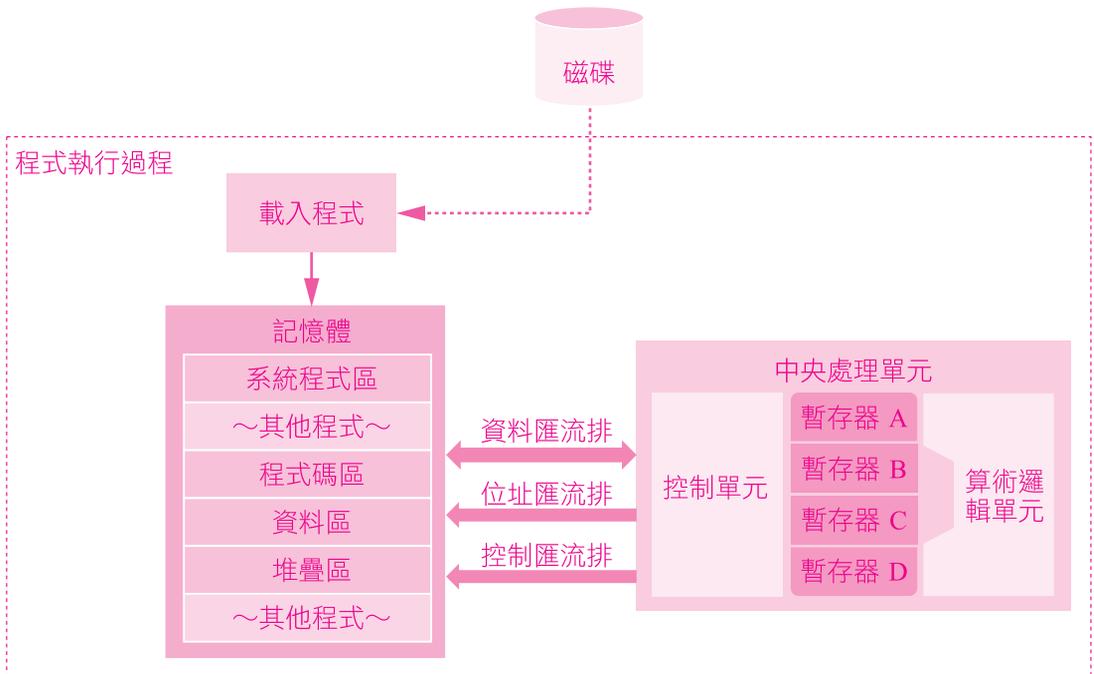
解 針對兩種指令集特性的不同，整理如下表，指令長度、指令數目、定址模式等特性分別比較兩者的不同。

指令集類型	CISC	RISC
比較項目	複雜指令集	精簡指令集
指令長度	可變長度（格式）	固定長度（格式）
指令數與定址模式	多	少
每個指令執行週期	不一定	一個時脈週期
參考（存取）記憶體	任何指令都可以	只有 LOAD 與 STORE 可以存取記憶體
暫存器集合	一組暫存器	多組大量暫存器
指令管線化	無	高度管線化設計
執行指令	由微程式執行指令	指令由硬體直接執行
缺點	複雜的微程式與硬體	複雜的編譯器與軟體

4. 請說明一個程式的執行過程。

解 執行程式的過程如下：

1. 載入程式 (Loader) 將可執行程式載入主記憶體中，並安排多個不同特性的記憶體區塊，包括程式區、變數區、堆疊區等。
2. 載入後中央處理單元就可以開始執行程式了，從第一道指令開始，進行提取、解碼與執行等過程，完成指令工作，再執行第二道指令以此類推。
3. 提取至指令暫存器，再經控制單元的指令解碼器進行解碼成數個微運算，再由控制單元指揮暫存器與算術邏輯單元進行這些微運算，最後將運算結果回存到暫存器，有些指令則需要回寫到記憶體中。



5. 請比較組譯、編譯與直譯的不同。

解 組譯、編譯與直譯三者的特色比較如下表所示。

翻譯方式	語言翻譯程式	使用的程式語言		特色
組譯	組譯器 (Assembler)	低階	組合語言	1. 不同的 CPU，指令不同，組譯器也不同 2. 會產生目的檔 (.obj)
編譯	編譯器 (Compiler)	高階	C、C++、PASCAL、DELPHI	1. 大多為高階程式語言採用 2. 偵錯較不易、但執行速度快 3. 會產生目的檔 (.obj)
直譯	直譯器 (Interpreter)	高階	BASIC	1. 易學、適合初學者 2. 偵錯容易、但執行速度慢 3. 不會產生目的檔 (.obj)