

微電腦程式的作法

賴耿陽 譯

	JZ	RAM1
	INX	H
2	JMP	RAM1
	MOV	A, H
13	CMP	D
	JNZ	RAM2
	AND	CHECK
READ	LXI	H, E
	MVI	M, O
	LHLD	STA
	MOV	A, M
	CMP	L
	JZ	RAM
	MOV	C, I
	B	
	LXI	H,
	INR	M
	MOV	

本書特色

由假設讀者對微電腦軟體一無所知而解說。

並以 Intel-8080 為例說明命令。

對各命令的解說附有演習問題和解答，容易瞭解。

並附錄 Intel-8080 的命令組和變換表，以便利用。

微電腦系列叢書

(1) 微電腦機器的設計

(2) 微電腦的製作法

(3) 微電腦的活用法

(4) 微電腦的開發技巧

(5) 微電腦程式的作法

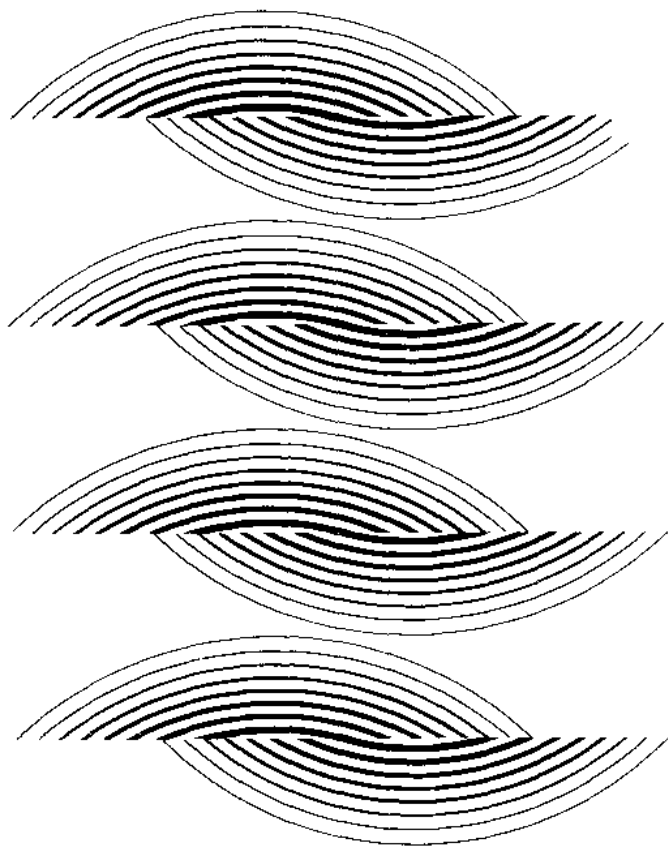
(6) 微處理機控制的設計

特價壹佰參拾元

* 出版者：王家出版社 * 地址：台南市新和路三十三號 * 電話：〇六二一 二一三八三八 * 出版登記證：字號行政院新
聞局局版台業字第〇四一六號 * 郵政劃撥帳號：二一八一七號 * 發行人：王惠月 版權所有翻印必究 * 印刷者：
美光彩色印刷股份有限公司 * 地址：台南市新和路四號 中華民國六十九年十月四日行

微電腦程式的作法

賴耿陽編譯



王富出版社印行

序

複雜者昂貴，簡單者便宜，建設高樓大廈需要大筆經費，小木屋就很經濟。

但是，微電腦的問世却打破此一千古不變的原則，微電腦小而便宜，但內容很複雜，有衆多機能，一個小小的東西，却有高度工作能力。

學會微電腦技術有何利益？這可列出一組損益答案，不過，不拘利益如何，一個人若能全神貫注於某一事物，在其中所體會的樂趣不就夠了嗎？

筆者深願有更多的人共享微電腦的樂趣，編書的宗旨即在促進普及。

各LSI製造廠推出各種實驗機組或適於OEM的組件，業餘的微電腦愛好者或可不必對硬體費太多心神。

促進軟體——程式計劃的普及已是今後的重要問題。

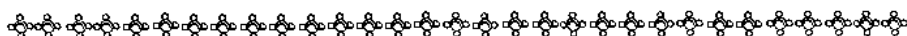
本書假定讀者對微電腦的軟體一無所知，先說明微電腦的骨架，其次以Intel-8080為例說明命令，說明若干命令後，以學得的命令練習組合程式，並以實際的微電腦實習。後半部假設擁有一部RAM型微電腦，敘述監視程式的作法和若干例子。

本書所述的程式未必可照抄而實用，不過，確是公開筆者開發的程式，足供參考。

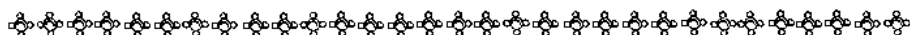
最後敬獻讀者一句話「不管別人怎樣，試作自己的程式，以朝氣和冒險心向新鮮事、未知物挑戰！」

1980年6月

編者



目 錄



第 1 章 導 論

1.1	電腦的動作與結構·····	7
1.2	電腦與情報·····	10
1.3	程式的輸入方法·····	12
1.4	數字表示法·····	13
1.5	程式作成用程式·····	21

第 2 章 命令與程式計劃的基礎

2.1	本書討論的微電腦構成·····	25
2.2	轉送命令·····	28
2.3	增量、減量命令·····	40
2.4	算術演算命令·····	43
2.5	跳越命令·····	54
2.6	邏輯演算命令·····	61
2.7	10 進演算命令·····	72
2.8	副常式與堆疊命令·····	76
2.9	擠入命令及其他·····	87

第 3 章 輸入出程式與管理監視程式

6 目 次

3.1	程式開發法的選擇.....	93
3.2	十六進法加感程式.....	95
3.3	鍵盤加感程式.....	101
3.4	記憶傾銷.....	107
3.5	記憶打孔與紙帶確認程式.....	113
3.6	監視程式.....	121

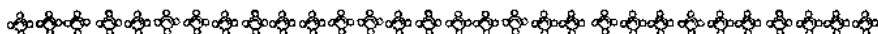
第 4 章 應用程式例

4.1	前言.....	127
4.2	整數的平方根.....	127
4.3	RAM 試驗程式.....	132
4.4	數字的變換程式.....	135
4.5	標籤表處理程式.....	138
後記.....		145
參考資料.....		151
附錄.....		153



第 1 章

導論



1.1 電腦的動作與結構

本章為初次試寫電腦程式的讀者說明電腦的結構。

要使電腦工作時，須有程式（program），程式須依序排列該電腦的命令。

例如想作一道菜，却不知怎麼下手，於是參考食譜，上面寫著「預先在青豌豆上撒鹽，將 3 杯水煮沸，沸騰後加 1 匙鹽，投入青豌豆……」，此時，作菜的人便是電腦，食譜便是程式，若將食譜的文章分解成電腦的命令，則如下示：

命令 1. 在青豌豆上撒鹽

命令 2. 使 3 杯水沸騰

命令 3. 將青豌豆投入其中

電腦的程式也與此相同，例如

0000	AF		XRA	A
0001	210600		LXI	H,WS
0004	77		MOV	M,A
0005	76		HLT	
0006		WS	DS	1
			END	

上示程式的意義容後解說，第 1 行的 XRA A 是命令「使電腦的累積器（accumulator）和進位（carry）為零」，試與食譜的「在青豌豆上撒鹽」比較一下，同是指令某人的動作。

8 第1章 導論

電腦為邏輯性機械，不可有曖昧的動作，所以電腦可實行的動作須預先清楚規定，電腦所能實行的動作集合為命令（instruction）。

用電腦執行一工作時，須分解工作的實行順序，寫成計算機命令的集結。但在分解順序時，須先瞭解電腦有那些命令，宛如作菜時，須先知道冰箱中有什麼材料。

瞭解電腦命令的過程就如同研讀六法全書，很難令人意趣橫生，但不管有無興趣，反正非懂不可，不歷經單調乏味的學習過程，無法學成電腦程式計劃技術。

吾人很難獨自學成單調的技術，不論如何清楚掌握觀念，總會在途中遺忘，所以最好跟隨優秀的指導者學習，學習效率最好，費用雖較高，獲益却較大。

不能如此時，不妨糾集數人，結伴研習，互切互磋。

無論如何，使用某電腦時，須先解讀該電腦的命令定義。

原則上，電腦是逐一依序實行命令，所以命令須依實行順序排成一行，通常命令是從紙的上方往下列出。如此依序排列命令者稱為程式（program），編列程式之事稱為程式計劃（programming），負責此事的人稱為程式計劃員（programmer）。

現假設作成一程式，欲將之輸入電腦，獲得結果，應怎麼辦？

先要知道電腦命令實行過程的大概，如圖 1.1 所示，電腦分為記憶情報的部份與演算部，記憶裝置的英文為 memory，演算部稱為中央處理裝置（central processing unit, CPU），記憶部宛如廚房的

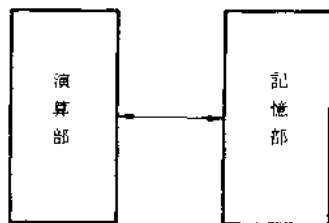


圖 1.1 電腦部的構成

冰箱，演算部可說是調理台。

程式先存入記憶部中，就像從超級市場買菜回來後，先存入冰箱一般，電腦的記憶部不像冰箱那麼亂，而是區間整然，命令存入各區間中，如圖1.2所示，各區間有編號，此稱記憶部的位址（address），例如位址0、位址1……位址150……等。

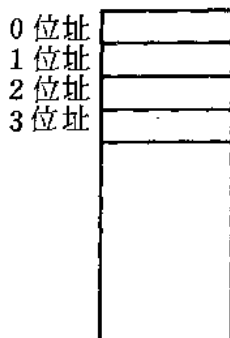


圖 1.2 記憶部的構造

微電腦（micro computer）大都從位址0的命令開始實行，所以作成程式後，須從記憶部的位址0依序存入，後面會說明如何將程式存入記憶部。

其次談電腦的演算部，CPU 從記憶部第一道命令（否則，CPU 不知工作內容），取出命令後，該處寫明 CPU 應做何事，CPU 進行該命令規定的動作，結束該命令的實行，着手從記憶部取出次一命令。如此依序實行命令，最後停止命令來到時，CPU 在該處結束動作，如此結束一項計算。

CPU 從記憶部取出命令的過程稱為取出循環（fetch cycle），實行命令的過程稱為命令的實行循環（execution cycle）。

CPU 中有臨時記憶取自記憶部之情報的少量記憶回路，特稱暫存器（register），例如從記憶部取出命令時，須將它保存到命令實行終了，臨時記憶此命令的回路稱為命令暫存器（instruction register）。

CPU 中，有記憶現正實行幾號位址命令之情報的暫存器，此稱程式計數器（program counter：PC）。

CPU 的主要機能是調理情報，亦即實行加減算之類的演算及AND、OR之類的邏輯演算，故有可實行此種演算的回路，這

些回路稱為ALU (arithmetic and logic unit)，這些回路動作的嚴密定義容後詳述。

1.2 電腦與情報

任何機械所能處理的事物都有限度，菜刀可切菜切肉，却切不了石頭，電腦也一樣，電腦只能處理特定的形式，此形式以外者概不接受。

現在的電腦大都以電路組成，所以電腦只接受規格化的電氣信號，但在作程式的立場，即使不考慮電氣條件，也可行程式計劃，所以不必詳述電路。

構成電腦的回路元件稱為邏輯回路或數位回路等，有興趣製作微電腦的人，可另參考專書。

電腦通常有高電壓與低電壓2種不同的電壓，例如+5伏特與零伏特，依此電壓為+5伏特或零伏特而決定電腦中的回路狀態，現將+5伏特稱為狀態的“1”，零伏特稱為狀態的“0”，結果，依電腦中暫存器或記憶部為1或0而決定電腦的狀態。

電腦所能接受的是“1”或“0”的情報，反過來說，欲令電腦工作時，命令、資料 (data) 都須改為1或0的形式。電腦工作的結果原則上也以1或0的形式送出。

現設有1條信號線，此信號線在某時刻所能取得的狀態為1或0，可藉信號表現2個不同的事象。若有2條信號線，可表現4個不同的事象。考慮信號的順序而設00，01，10，11四種符號。若有3條獨立的信號線，可作成000，001，010，011，100，101，110，111八個互相不同的符號。一般上， n 條獨立的信號線最多可作成 2^n 個符號。

反之，欲作成 m 個符號時，找出滿足 $2^{n-1} < m \leq 2^n$ 的 n ，

它就是必要的信號線數，例如 $m = 15$ 時，15 大於 8 而小於 16，所以 $2^3 < 15 < 2^4$ 的式子成立，需要 4 條信號線，一般寫成

$$n = \log_2 m$$

此處所謂的信號線數取為情報的基本單位，稱為 bit（位元），區別 15 個物品時需要 4 bit 的情報。有 250 個物品，欲設 1 與 0 的符號區別時，試問需幾 bit 的情報。

電腦處理情報的形式有各種形式，現代的電腦（包括微電腦）可並列處理若干情報，例如普通的迷你電腦可並列處理 16 bit 的情報，大型電腦可同時處理 32 bit、50 bit 或更多的情報。

微電腦通常總括處理 8 bit 的情報，8 bit 特別稱為 1 byte（數元組），所以微電腦又稱 byte machine。也有的微電腦將 4 bit、12 bit、16 bit 視同一情報單位，在此只考慮 8 bit 的微電腦。

情報處理為 8 bit（1 byte）的微電腦如圖 1.3 所示，CPU 與記憶部之間有 8 條線，CPU 經此線從記憶部取出 8 bit 的情報，或寫入記憶部。傳送情報的線成束者稱為巴士（bus），圖 1.3 的巴士載有資料，故稱資料巴士（data bus）。情報在寫入時從 CPU 部傳送到記憶部，讀出時，從記憶

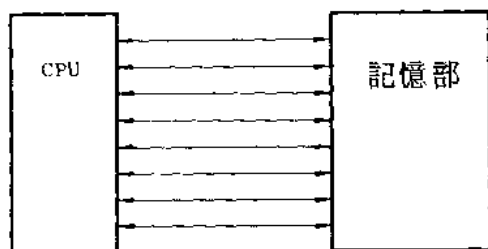


圖 1.3 8 條資料巴士

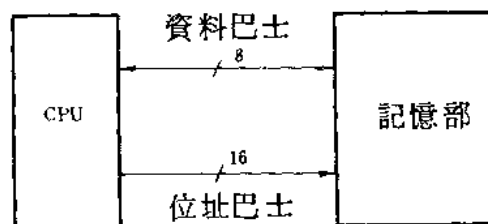


圖 1.4 位址巴士與資料巴士

部傳送到 CPU 部。如此在兩端授受情報的巴士稱為雙方向性巴士 (bidirectional bus)。

CPU 與記憶部授受情報時，要指定與記憶部的何一 cell 授受，傳送此情報的束線稱為位址巴士 (address bus)，如圖 1.4 所示，位址巴士從 CPU 將情報送到記憶部，以指示資料巴士授受情報的位址，例如位址巴士的情報指定位址 1，則 CPU 與記憶部的位址 1 授受資料。

如此圖所示，若有 16 條位址巴士，全部可指定 2^{16} 個記憶位址，若計算 2^{10} ，則成 1024，情報的 1024 稱為 1 K，若用此單位，有 16 bit 位址巴士的微電腦最大有 64 K byte 的記憶，此時，實際的記憶位址數為 $64 \times 1024 = 65536$ ，可約略記成六萬四千位址。

當然最多可設置此數字的記憶，實用微電腦的記憶數因系統而異，實際使用時查明記憶容量而程式計劃即可。

1.3 程式的輸入方法

微電腦是依序實行存入記憶部的程式的機械，那麼，最初如何將程式存入記憶部。

首先說明微電腦的記憶由何種元件組成，微電腦的記憶製造技術現仍在發展中，未完全確定。

目前用為微電腦記憶部的 LSI 有 RAM 和 ROM，RAM 為 Random Access Memory 第 1 字母的縮寫，可行寫入與讀出雙方的操作，用於記憶資料之類每次處理都變化的情報。

RAM 切斷電源開關時，則內容消失，再使電源 ON 時，也不再現原內容，有此性質的記憶稱為揮發性 (volatile) 記憶。只由 RAM 作成微電腦記憶部時，每次切斷電源便使記憶部內部成

白紙狀態，很不方便。

ROM爲 Read Only Memory 的縮寫，在LSI中半永久性寫入情報，即使切斷電源，也不蒸發內容，巧用此點，將常用的程式寫於此ROM中，每次電源ON時，某特定程式即存在記憶部中，驅動該程式，微電腦便開始動作。

讀出專用的記憶部若全然不能寫入，就失去記憶部的意義。依情報寫入ROM中的方法，ROM分若干等級。在LSI製造工程中寫入特定情報者稱爲mask-ROM，用於情報類型（pattern）安定而達某數量的場合。

記憶部中有熔絲（保險絲）之類，以電流將之燒斷而記憶情報的1與0，此種ROM若有流入適當電流的裝置，到處可寫入情報，故稱可在現場寫程式的ROM（field programmable ROM），此ROM有利於少量的場合，但在誤寫程式時，無法修正，將情報寫入ROM的裝置稱爲ROM-writer。

有的ROM可在記憶部寫入或消去程式，稱爲erasable reprogrammable ROM，此型ROM的LSI中央開窗，從此處照射某特定波長的紫外線，即可吹消寫入其中的情報，回復白紙狀態。

用適當的ROM-writer寫入程式時，新情報存入（store）記憶部中，若寫入的程式有錯誤，可再照射紫外線，洗掉（clear）記憶，成爲可再寫入的狀態，如此可數次寫入程式，便於檢查程式。

以新程式的檢查（check）爲主時，宜用以RAM爲主體的微電腦。

1.4 數字表示法

14 第 1 章 導論

微電腦雖小，但總也是電腦的一種，一定有演算數字的機能，下面說明微電腦中的演算方法。

如前所述，電腦中只有高電壓與低電壓、符號只有“1”與“0”之分，只能表示 2 個數，故具備若干信號為 1 個數的單位，此基本單位為 8 bit，如此作成下示 256 種符號。

1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 0

}

1 0 0 0 0 0 0 1

1 0 0 0 0 0 0 0

0 1 1 1 1 1 1 1

}

0 0 0 0 0 0 0 1

0 0 0 0 0 0 0 0

各組可分派任何數字，不過，隨意分派時，記憶很麻煩，數字間的加減算也需要複雜的回路，當然應有規則地分派。

通常電腦中的數字表示是用 2 進法，其表示法異於常用的十進數，不過，對 10 進數成立的法則也大都適用於 2 進數。

例如，欲知 2 進數 00000101 等於 10 進數的多少時，將 2 進數各位的倍數相加即可。

0 0 0 0 0 1 0 1

↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

128 64 32 16 8 4 2 1

宛如 10 進數的個位、十位、百位……，10 進數的第 n 位為

10^{n-1} ，2 進數是 2^{n-1} ，此例中的 1 在 1 與 4 位，故成

$$1 + 4 = 5$$

故知 2 進數的 0000 0101 為 10 進數的 5。

如何將 10 進數改成 2 進數？以 10 進數的 28 為例，因只 28，所以 32、64、128 位一定為零，28 大於 16，所以 16 位須為 1，28 的初步形態為

$$0 \ 0 \ 0 \ 1 \quad \times \times \times \times$$

$28 - 16 = 12$ ，所以其次的 8 位也是 1

$$0 \ 0 \ 0 \ 1 \quad 1 \times \times \times$$

$12 - 8 = 4$ ，所以其次的 4 位也是 1

$$0 \ 0 \ 0 \ 1 \quad 1 \ 1 \times \times$$

$4 - 4 = 0$ ，已不必再標 1，所以 28 表成

$$0 \ 0 \ 0 \ 1 \quad 1 \ 1 \ 0 \ 0$$

筆算法如下

$$2 \overline{) 28} \quad \dots\dots 0$$

$$2 \overline{) 14} \quad \dots\dots 0$$

$$2 \overline{) 7} \quad \dots\dots 1$$

$$2 \overline{) 3} \quad \dots\dots 1$$

$$2 \overline{) 1} \quad \dots\dots 1$$

0

