計算機叢書

計算機結構

鄭金火　麥松梅

興業圖書股份有限公司

計 算 機 叢 書

計 算 機 結 構

（集合語言）

鄭 金 火　麥 松 梅

興業圖書股份有限公司

中華民國六十五年三月一日　一版

計算機結構

特價：壹佰貳拾元

編著者：鄭　金　火・麥　松　梅
發行人：王　　志　　康
出版登記證局版台業字第〇四一〇號
出版者：興業圖書股份有限公司
印刷者：廣　益　印　書　局
地　址：台中市中正路九十七號
打字者：義　德　春　打　字　行
地　址：台南市正興街二十四號
發行所：興業圖書股份有限公司
地　址：台南市勝利路一一八號
郵政劃撥南字 31573 號

篇者的話

最近幾年來，計算機科學的進展，一日千里，其中尤以迷你計算機（mini-computer）之銷售與應用，更是一枝獨秀。而在台灣現在所使用的迷你計算機，以惠普（HP）公司所出之迷你計算機系列爲最多。而在使用及研究迷你計算時，最先接觸的問題就是計算機語言，尤其是集合語言（Assemble language）。惠普公司所提供有關集合語言的資料，固是汗牛充棟，但都是以手册之型式篇排，對一般初學者而言，有不知從何下手的感覺，篇者有見於此，是以將惠普公司所提供的有關參考資料，集合整理，分門別類，使初學者能一目了然。

爲了加深讀者的印象及對各種命令（Instruction）之了解，篇者在本書附上很多實例，除了第七章之輸入及輸出命令外，每章之實例皆以可重置（Relocatable）之型式書寫，至於第七章，爲了使讀者更容易了解各種終端機（Terminals）輸入輸出動作與計算機之配合，是以絕對（Absolute）之程式書寫。

依篇者的想法，學習集合語言，非但爲了解語言本身之用法，最重要的是，一方面可幫助我們了解一些計算機在數學上之邏輯觀念，另一方面可幫助我們了解一些計算之動作步驟及基本原理，故讀書學習集合語言時，不要只是注重程式本身之設計，並應同時注意整個計算機之結構原理。

篇者在篇著本書時，承蒙耿興玲及蔡秀珠兩老師的幫忙及指正，在此並致謝意。篇者才識有限，書內錯誤之處，在所難免，但望高明之士，不吝指教，以做爲再版之參考。

編者　六十五年二月

目　　錄

第一章　計算機結構簡介

第二章　HP2100系統簡介

第三章　記憶器相關命令

第四章　記發器相關命令

第五章　擴張算術和浮動小數點命令

第六章　假擬命令

第七章　輸入，輸出

第八章　程式之設計技巧

附錄

第一章　計算機結構簡介

1－1　導言

一提到電子計算機（digital computer，簡稱 computer），我們就想像到一個巨大的匣子，有一連串的開關、按扭、閃亮的指示燈，你叫它做什麼就能做什麼，故有"電腦"之稱。其實，除非具有下列兩種要素，它是沒有任何用處的；第一使用計算機的人能使用某種邏輯的程序使計算機能夠接受此命，第二計算機與使用者之間必需有一翻譯器（Translator），將計算機所接受的命令變爲計算機所能執行的工作。無論再大型再複雜的計算機，它所能做的就是接受一連串不同的輸入組合，更簡單的說就是接受一連串非常簡單不同電子信號，而計算機如何將此簡單的電子信號，改變爲使用人的命令，此即爲翻譯器的功用，而使用人的命令，亦需有一定格式，此即牽涉的有關計算機的語言問題，各種不同的命令格式就變成各種不同的語言程式，而計算機對各種使用者的語言程式（programming），必需各具有不同的翻譯器，故一部計算機能做什麼工作，能接受何種語言程式，與計算機本身構造並無直接關係。

由前所述，有關計算機的使用，牽涉到三部份，剛品（Hardware），軟品（Software）和程式（Programming）。

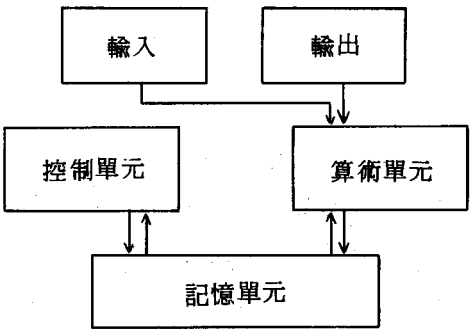
1－2　剛品（Hardware）

所謂計算機的剛品，就是指有關計算機的機械與電氣，設計與裝置亦即是計算機構造，一部計算機設計製造生產後，其剛品就定型了，除非將其電路等用機械的方法改裝，不然是不能隨便更改的。

計算機其剛品大約可分爲四部份：

1. 控制單元（Control element）：控制資料數據的移轉及各種操作之執行。
2. 算術單元（Arithmetic element）：做各種算術（如＋，－，×，÷）及邏輯（如AND，OR，NDT）運算。
3. 記憶單元（Memory element）：用於貯存所需的資料。
4. 輸入／輸出單元（Inpnt／outpnt element）：用於將資料傳送

（送入或送出）到計算機的記憶單元，傳送的外圍裝置如讀紙帶機（Paper tape readers），打孔機（Punches），印刷機（Printers）等等。其構成以圖表示如下：



1－3　軟品（Software）

計算機剛品所能認識的信息只不過是一連串開與關的電流或電壓脈衝，此開或關所產生的電壓階層，常用來代表數學或邏輯數的0和1，有所謂正邏輯與副邏輯之分別，所謂正邏輯，是以大電壓表大數小電壓表小數（如以－3V表1,－5V表0），而負邏輯是以小電壓表大數大電壓表小數（如以＋5V表0而＋3V表1）。而每個表0或1之電壓或電流就叫一數元（bit）而計算機都是用不同的0與1（即電壓之高低）來代表一符號或操作某一動作，此0與1之一連串組合就叫做數位（digit），此即是數位計算機（digital compnter）名稱的來源。但是如果叫一般使用的人寫一連串0，1計算機所能明瞭的語言是非常不簡單的事，故才有“翻譯器”（Translator）的設計。

軟品（Software）一詞，如“翻譯器”（Translator），控制系統（Control system）是用來有別於有實際裝置的剛品，軟品的翻譯器，控制系統是一連串的計算機命令，通常貯存在低帶、磁帶、磁碟等媒介上，要用時再從媒介送入計算機。

翻譯器（Translator）是接受某種人爲的計算機語言，而再將其變成計算機所能執行的機械語言（machine language），翻譯器大體可分爲兩類：

1. 組合器（assembler）：使每一命令能用一縮小的助憶碼（mn-

emonic code)表示,通常一助憶碼就譯成一機械命令，但是大部份的組合都含有一些假命令(Pseudo instrnction)和副程式(subroutine call)，這些都能產生很多的機械命令去執行某些特殊的工作。

2. 編輯器(Compilers)：高級語言的表示較接近人類習見的字句與公式，一命令的指述可以變成很多機械的命令。組合器一般是根據計算機之機械動作而設計的，而編輯器是根據使用人慣用的語言型式而設計的，故使用編輯器的語言程式比較接近使用人的語言方式，而使用組合器的語言程式，比較接近計算機的動作方式。例如互傳(FORTRAN 或 FORmula TRANslation)，很容易表示科學上複雜的數學公式(ALGOL，或 ALGOrithmic Language)是用來表示各種數值過程(numerical processes)的精確語言。

另外一種軟品是控制系統(control system)或叫模擬系統(monitor system)又叫操作系統(operating system)。控制系統是用來編排組合器和編輯器所產生的程式，例如做爲配置(loading)和錯誤檢查(error detection)幫助之用。

1－4　做程式（Programming）

做程式包括兩部分：

1. 分析問題而決定獲得答案的必需步驟。
2. 編纂最適宜於軟品語言的解決程序。

1－5　計算機語言的分類

用來做程式的計算機語言，種類非常多，根據非正式的統計，約有一千之多，當然，一般人所習用的，大體上不超過十種，其中用的最多的，便是互傳(Fortran)，再次爲便習(Basic)等等大體上分爲兩大類：

1. 高級語言(High- level language)：包括

互傳(Fortran)
便習(Basic)
賈伯(COBAL)
PL／I
ALGOL

2. 低級語言（Low-Level language）：包括

集合語言（assembler language）
機械語言（machinc language）

一般來講，使用高級語言而做程式時較簡單方便，且不要對計算機之原理與構造有任何了解，而使用低級語言做程式時，非但必需對所使用的計算的原理與構造有相當認識而且做程式之步驟非常的煩雜，這樣說來，我們利用低級語言（集合語言）做程式，究竟有什麼好處呢？

1-6 集合語言（Assembley）

即然集合語言難學，用來做程式又麻煩，那麼我們爲什麼要學習集合語言，用集合語言做程式呢！有下列幾點原因。

1. 有很多程式，很難甚至不能用高級語言，如互傳，便習去做，而需用集合語言去做。
2. 在控制方面，如計算機輸入及輸出之細部控制，非用集合語言不可
3. 可從集合語言的命令程序，進而了解計算機之動作原理。

但學習集合語言需對計算機之剛品與軟品有相當認識，下列數節，是針對這些需要而寫。

1-7 記發器（Register）

記發器是一套貯存代表數字0，1的電路裝置，其0，1（代表高，低電階）數字，可從別的記發器或記憶器（Memory）移入，亦可從此記發器移到別的記發器或記憶器，在同一記發器中，其0，1的排列，亦可作各種改變的動作，如移位（shift）旋移（Rotate）等，此等動作，將在後節詳述之。

1-8 二進制（The Binary System）

以前計算機之基本單元乃繼電器（Relay）及開關，而開關或替續器之操作，在本質上，即是基本二進位制。那就是說接通時可看作1，而扳斷時，則可看作零。至於現代大多數計算機的主要電路單元，則都採用電晶體及積體電路（I.C.）。由於在可靠性方面的要求，使得設計家們採用這些裝置時，只使它們處於兩種狀態之一，即是完全接通，或完全不通。故此型電路，與用開關的電燈電路仍很相似，即在任一時間，電燈（或電泡）只有亮（接通爲1），或不亮（不通爲零）。

1－9　二進制之表示方法

二進位數系也像十進制一樣，用了相同的位置記數法，以下即20 個二進位數之例。

十進位數		二進位數	十進位數		二進位數
1	=	1	11	=	1011
2	=	10	12	=	1100
3	=	11	13	=	1101
4	=	100	14	=	1110
5	=	101	15	=	1111
6	=	110	16	=	10000
7	=	111	17	=	10001
8	=	1000	18	=	10010
9	=	1001	19	=	10011
10	=	1010	20	=	10100

雖然二者同用位置記數法，但十進制所用的是10 乘冪 ，而二進制的則是2的乘冪。正如以前的說明，數目 125 的實在意義是 $(1\times10^2)+(2\times10^1)+(5\times10^0)$。在二進制中，同樣的數目（125）則是以1111101表示的，其意爲 $(1\times2^6)+(1\times2^5)+(1\times2^4)+(1\times2^3)+(1\times2^2)+(0\times2^1)+(1\times2^0)$。

因此二進位的數值的表示，是 $a_1 2^{n-1}+a_2 2^{n-2}+\cdots+a_n$，其中 $a_1, a_2, a_3 \cdots a_n$ 之 a_n爲 1 或 0 ，而 n 則爲自二進制之小數點起，向左數至頂端 數字之數目。

下列爲表示由二進位數目換算爲十進制之例：

$$101=(1\times2^{3-1})+(0\times2^{3-2})+(1\times2^{3-3})$$
$$=(1\times2^2)+(0\times2^1)+(1\times2^0)$$
$$=4+1=5$$

$$1001=(1\times2^{4-1})+(0\times2^{4-2})+(0\times2^{4-3})+(1\times2^{4-4})$$
$$=(1\times2^3)+(0\times2^2)+(0\times2^1)+(1\times2^0)$$
$$=8+1=9$$

$$11.011=(1\times2^{2-1})+(1\times2^{2-2})+(0\times2^{2-3})+(1\times2^{2-4})+(1\times5^{2-5})$$
$$=(1\times2^1)+(1\times2^0)+(0\times2^{-1})+(1\times2^{-2})+(1\times2^{-3})$$
$$=2+1+\frac{1}{4}+\frac{1}{8}=3\frac{3}{8}$$

宜注意者，分數，或一般所稱之小數之製成，與在十進制時是相同的，如：在十進制中

$$0\cdot123=(1\times10^{-1})+(2\times10^{-2})+(3\times10^{-3})$$

在二進制中

$$0\cdot101=(1\times2^{-1})+(0\times2^{-2})+(1\times2^{-3})$$

1－10　二進位加法與減法

(Binary Addition and Subtraction)

二進位加法的演算，與十進位加法的情形，是相同的。實際上二進位的計算，更易於學習，以下即二進位加法表之全部

$0+0=0$
$0+1=1$
$1+1=0$　但需加上進位之1

「進位」(Carry-overs)的演算，也與十進制中相同。由於1是二進制中最大的數字，任何比1爲大的和皆需進位。例如二進位數100加100即需將兩個1在第三位中相加，而後向左進一位，因1＋1＝0但需加上進位之1，故100加100之和爲1000。以下再舉三個二進位加法的例子：

十進位	二進位	十進位	二進位	十進位	二進位
				$3\frac{1}{4}$	
5	101	15	1111	$5\frac{3}{4}$	11·01
6	110	20	10100		101·11
11	1011	35	100011	9	1001·00

減法乃加法之逆算，爲了要減，必需先制定小數減去大數的程序。在二進位制中，只有當從0減去1時才有此情形，其差爲1，但必需向左邊一位數借1。以下即爲二進位減法表：

$0-0=0$
$1-0=1$
$1-1=0$
$0-1=1$　但需借1

茲再舉幾個例子，當能使讀者對二進位減法，有更清楚之瞭解。

十進位	二進位	十進位	二進位	十進位	二進位
9	1001	16	10000	6 ¼	110.01
−5	−001	−3	−11	−4 ½	−100.1
4	100	13	1101	1 ¾	1.11

1－11　二進位乘法與除法

(Binary Multiplication and Division)

二進位數乘法表是很短的，它只有三條，不像十進位系法有 100 條。茲將二進位乘法寫於下：

$$0 \times 0 = 0$$
$$0 \times 1 = 0$$
$$1 \times 1 = 1$$

下列三個例子，已把二進位乘法的每一種演算都敍述了，若乘數爲 1，則其積只需照被乘數寫出即足，若乘數爲零時不論被乘數爲何，其積應爲 0。顯然地，這樣的演算，其每一部都是很容易的。

十進位	二進位	十進位	二進位	十進位	二進位
12	1100	102	1100110	1.25	1.01
×10	×1010	×8	× 1000	×2.5	10.1
120	0000	816	1100110000	625	101
	1100			250	101
	0000			3.125	11.001
	1100				
	1111000				

二進位除法也是很簡單的。也像在十進制（或其他各制）中一樣，但是被零除是無意義的。其除法表如下：

$$0 \div 1 = 0$$
$$1 \div 1 = 1$$

茲舉除法的兩個例子如下：

十進位	二進位	十進位	二進位
5	101	2·416…	10·0110101…
5)25	101)11001	12)29·0000	1100)11101·00
	101	24	1100
	101	50	10100
	101	48	1100
		20	10000
		12	1100
		80	10000
		72	1100
		8	…

例二的商，如欲由二進位數換算爲十進位數時，其程序如下：

$$
\begin{aligned}
10\cdot0110101 = {} & 1\times2^{1} && = 2.0\\
& 0\times2^{0} && = 0.0\\
& 0\times2^{-1} && = 0.0\\
& 1\times2^{-2} && = 0.25\\
& 1\times2^{-3} && = 0.125\\
& 0\times2^{-4} && = 0.0\\
& 1\times2^{-5} && = 0.03125\\
& 0\times2^{-6} && = 0.0\\
& 1\times2^{-7} && = 0.0078125\\
& 0\times2^{-8} && = 0.0\\
& 0\times2^{-9} && = 0.001953125\\
& && \overline{\quad 2.416015625}
\end{aligned}
$$

因此二進位數10·0110101，即約等於十進位數2·416。

1－12　將十進位數換算爲二進位數

(Converting Decimdal Numbers to Binary)

將十進位數換算爲二進位數的方法，有好些種。第一種也是最明顯的方法，就是減去十進位數中所有2的乘冪，直至無所剩餘爲止。首先要減去其中最大的2乘冪，再減其次，依此類推。例如要把十進位整數25換算爲二進位數系時，首先要找到25中最大2的乘冪，此即$2^4=16$，於是$25-16=9$。9中最大2的乘冪爲2^3，即8，再減去此數即只餘1，或2^0，故25可表示爲11001。

用這種方法來換算數字，是既乏味而費力的。在數目少時，照此方法，用人力來算尚還可以，但是却很少用在大數目的。故在大數目上要採用另一方法，此法爲將該數種複地除以2，於是其各項之餘數，即求爲二進位數之各系數。但需注意，所推演出的二進位數，要由下端向上寫去。

$$
\begin{aligned}
125 \div 2 &= 62 + \text{餘數}\ 1\\
62 \div 2 &= 31 + \text{餘數}\ 0\\
31 \div 2 &= 15 + \text{餘數}\ 1\\
15 \div 2 &= 7 + \text{餘數}\ 1\\
7 \div 2 &= 3 + \text{餘數}\ 1\\
3 \div 2 &= 1 + \text{餘數}\ 1\\
1 + 2 &= 0 + \text{餘數}\ 1
\end{aligned}
$$

故125之二進位表示法，即由上表之下端向上端寫出，即1111101。此結果可校對如下：

$$
\begin{aligned}
1 \times 2^6 &= 64\\
1 \times 2^5 &= 32\\
1 \times 2^4 &= 16\\
1 \times 2^3 &= 8\\
1 \times 2^2 &= 4\\
1 \times 2^1 &= 0\\
1 \times 2^0 &= 1\\
&\overline{\quad 125}
\end{aligned}
$$

此種方法，不適用於帶小數的數目。若仍要用此法，則必需先將該數目分爲整數部分及小數部分，然後才能應用上述的方法。如102·274 即可分爲102及 0·274 ，然後再個別的求出其二進位數，再將此兩部份合起來即得。

十進位小數，可用好幾種方法換算之爲二進位小數。其中最明顯的方法，即是從十進位小數中減去最高的2乘冪（乘冪爲負數）。于是再從其差減去次一最高的2乘冪，如此演算下去直至再無餘數，或已達到所需之精確度時爲止。

$$0\cdot875-(1\times2^{-1})=0\cdot875-0\cdot5=0\cdot375$$
$$0\cdot375-(1\times2^{-2})=0\cdot375-0\cdot25=0\cdot125$$
$$0\cdot125-(1\times2^{-3})=0\cdot125-0\cdot125=0$$

因此將十進位數0·875，換算成二進位數後，即是0·111。對於更長的小數，則可將小數重複地以2乘之，可更簡單。在乘以2後，若在小數點左端出現一個1，則此1即可加於欲換算成之二進位小數之右端。若乘以2後，十進位數小數點之左端餘數0，則二進位之右端即可加以0。以下即爲換算十進位數0·4375爲二進位數之例：

	二進位表示法
$2\times0\cdot4375=0\cdot8750$	·0
$2\times0\cdot875=1\cdot750$	·01
$2\times0\cdot75=1\cdot50$	·011
$2\times0\cdot5=1\cdot0$	·0111

故對十進位數0·4375之二進位之表示，爲0·0111。

1－13　二進位數字之表示法

（Representation of Binary Numbers）

在一架電子數位計算機中，其資料之處理，是藉交換（Switching）與貯存電的信號而執行的。使用二進制操作的計算機，必需使其信號，只能表示兩個數值中之一個，此即謂，每一信號不表示0，即表示1。

圖1－1(a)即說明利用直流電（d－c）之不同電位（原文爲Lewel，意爲水平，以形狀得名，以後即用電位）在一條線上表示二進位數的方法。這種方法是以負d－c電壓表示0，而以正d－c電壓表1。此圖顯示用一個－10伏特d－c信號，以表示0，而用一個＋10伏特d－c信號，以表示