

工業用書

微電腦硬體與軟體基礎

電腦語言中心編輯部編



電腦語言中心出版

微電腦 硬體與軟體基礎

電腦語言中心編輯部編

電腦語言中心出版

微電腦硬體與軟體基礎

編 者：電腦語言中心編輯部

出版者：電 腦 語 言 中 心

發行者：

九龍彩虹道400號六樓

印刷者：合興隆印刷公司

香港仔宏利工業大廈七樓

定價港幣 ·\$ 22.00

序 文

「微電腦」是我們最近常聽到的新名詞。很多家庭用品都使用微電腦，因而它使我們生活更加豐富與幸福。

目前許多工業新產品都利用更進步的微電腦為製造的基本條件，不但在銷售數量上大為增加，也給我們帶來了很多意想不到的方便。

但微電腦到底是什麼東西呢？雖然有很多人想知道它，但以它是新的科技知識，而現有的參考書籍又過於專門化，在觀念上以為它是高深又複雜的專門知識，因而不得不放棄了研習它的嚮往。筆者為使一般人也能理解與接近電腦，而以簡明易懂的方式，撰寫此書。

本書在硬體方面利用很多圖片說明，在基礎上易於使人了解；同時在軟體方面，也利用模型簡單說明其機能。相信它能對想學習微電腦的人有所幫助。

目 錄

第 1 章 電腦是什麼

§ 1 電腦的語言	2
① 兩者選一就是電腦的資訊	2
② 數元和資訊量是什麼？	2
③ 電腦資訊的單位	5
④ 電腦的語言是“0”和“1”	6
⑤ 要使電腦工作需要程式	8
⑥ 以人的語言編寫程式	9
⑦ 組合語言是什麼？	11
⑧ 高階語言是什麼？	13
§ 2 微電腦的基本構成	16
① 電腦和電子計算機	16
② 電腦是怎樣構成的？	18
③ 微電腦是什麼？	23
④ 從利用方面看微電腦和電腦的差別	25

第 2 章 構成微電腦的要素

§ 1 構成微電腦的 I C	28
① L S I 就是大規模的 I C	28
② 利用電壓的高低傳達資訊	29
③ 利用二極體傳達資訊的方法	31
④ 利用電晶體（雙極型）傳達資訊的方法	32
⑤ 利用 M O S 型 F E T 傳達資訊的方法	34

§ 2 IC的基本邏輯電路	36
[1] 邏輯和邏輯電路	36
[2] AND電路(或稱為邏輯乘積電路)是什麼?	37
[3] OR電路(或稱為邏輯加電路)是什麼?	39
[4] NOT電路(或稱為反電路)是什麼?	42
§ 3 微電腦常用的應用邏輯電路	44
[1] NAND電路是什麼?	44
[2] NOR電路是什麼?	46
[3] Ex OR電路, Ex NOR電路是什麼?	47
§ 4 記憶的真面目—正反器的基本	50
[1] 波蘭電路和正反器	50
[2] RS - FF是什麼?—NPN電晶體	51
[3] 利用MOS的RS - FF	56
§ 5 各式各樣的正反器	59
[1] RST - FF是什麼?	59
[2] JK - FF是什麼?	63
[3] 主-附屬JK - FF是什麼?	65
[4] D - FF是什麼?	66
[5] T - FF是什麼?	69

第 3 章 微電腦電路的製作法

§ 1 簡單的邏輯電路的製作法	72
[1] Ex NOR電路的製作法	72
[2] Ex OR電路的製作法	74
§ 2 電路的經濟製作法—布耳代數	76
[1] 布耳代數是什麼?	76

[2] Ex NOR 電路的 NAND 化	80
[3] 稍為複雜電路的簡化和 NAND 化	81
§ 3 電路的經濟製作法—范氏圖和坎諾圖	84
[1] 范氏圖法的使用法	84
[2] 坎諾圖法的使用法	85
§ 4 加器的原理	89
[1] 半加器的原理	90
[2] 全加器的原理	92

第 4 章 微電腦的機能和裝置

§ 1 對學習微電腦硬體的前言	98
[1] 控制程式和靴帶式	98
[2] 關於指令的執行方法	99
§ 2 記憶器概論—記憶格	105
[1] 雙極型記憶格	105
[2] MOS 靜態型記憶格	108
[3] MOS 動態型記憶格	109
§ 3 記憶器概論—RAM 和 ROM	113
[1] RAM 是怎樣動作？	113
[2] ROM 是怎樣的記憶器？	116
[3] PROM	117
[4] ROM 是怎樣動作？	119
§ 4 編碼器電路和解碼器電路	121
[1] 編碼器電路是什麼？	122
[2] 解碼器電路是什麼？	123
§ 5 CPU 的內部構成—ALU 和累積器	128

[1] A L U (算術和邏輯裝置) — 演算裝置	128
[2] 累加器	132
[3] T 暫存器	133
[4] 旗 標	133
[5] 10 進補正	136
6 暫存器群和位址計數器	138
[1] 暫存器群的任務	138
[2] 通用暫存器	140
[3] W Z 暫存器	141
[4] 程式計數器	141
[5] 堆積指針	142
[6] 波閘 電路	143
[7] 暫存器選擇裝置和多元調節器	144
7 定時和控制裝置	145
[1] 指令的構成	145
[2] 時鐘產生器	147
[3] 需要兩種時鐘的理由	149
[4] 系統控制器	151
[5] 狀態資訊和機器週期	152
[6] 控制信號	155
8 匯流排線	163
[1] 資料匯流排	163
[2] 位址匯流排	164
[3] 控制匯流排	164
9 CPU 內的其他裝置	165
[1] 緩 衝 器	165

[2] 閘鎖裝置.....	166
[3] 移位暫存器.....	167
[4] 計數器電路.....	169
§ 10 微電腦作業的定時.....	173
§ 11 輸出入裝置和分界面.....	177
[1] 輸出入裝置.....	177
[2] 分界面的任務.....	182
[3] 優先間斷控制 P I C.....	184

第 5 章 學習微電腦的軟體

§ 1 利用假想迷你微電腦來了解軟體.....	188
[1] 假想“迷你微電腦”是什麼.....	188
[2] 決定迷你微電腦的指令語句.....	190
[3] 加法的流程圖和程式.....	191
[4] 加法程式的執行方法.....	194
§ 2 在微電腦寫入指令.....	197
[1] 靴帶式是什麼？.....	197
[2] 藉靴帶式編寫輸入常式的程式之方法.....	199
§ 3 重複同樣動作時的程式.....	202
[1] 相加的數目增加時從來的程式就變成複雜.....	202
[2] 迴路方式的程式.....	204
§ 4 輸入常式也改為迴路方式較有利.....	209
[1] 迴路式輸入常式的例子.....	209
§ 5 利用次常式的程式簡化.....	214
[1] 次常式是什麼.....	214
[2] 次常式的結構.....	215

§ 6 組合語言入門.....	218
[1] 組合語言的使用文字.....	219
[2] 組合語言的敘述.....	219
§ 7 迷你微電腦的組合語言.....	222
[1] 使用簡字符號的微電腦指令碼.....	222
[2] 以組合語言編寫的程式.....	224
[3] 含有迴路指令的組合語言程式.....	229
§ 8 8080 A 微電腦的程式.....	231
[1] 2 進數和 8 進數，16 進數之間關係.....	231
[2] 8080 A 微電腦的程式.....	233

第1章 電腦是什麼

目前，因為微電腦的應用不但已經擴大進入到各種領域，並且還繼續不斷地出現裝配微電腦的新產品，以致構成所謂的微電腦風行時期了。

如此受到歡迎的微電腦到底是什麼呢？首先，我們必須回答這個問題。

當然，我們不能對初學的人只說「微電腦就是利用微處理機的電腦」而就算做給他說明清楚了。因此，在第 1 章裡將盡量不使用專用術語來說明微電腦是什麼，使各位讀者能抓住它的概略概念。

當我們登山時，總要先在山腳眺望這座山的全貌，以確認將要上爬的路徑。同樣的，在本章裡將先請各位讀者來抓住微電腦這座山的構成概況。

此後，當我們遇到個別的問題時，方能了解這些問題到底處於全盤中的什麼位置。

§ 1 電腦的語言



☆ 兩者選一

① 兩者選一就是電腦的資訊

在西部片電影裡，常常出現以一枚硬幣的正面或反面打賭的情景。例如牛郎利用拋出硬幣，看它翻出正面或反面，以決定去打劫銀行，或是死了這一條心而到鄰近城市去一般的方法，決定他們命運的情景。

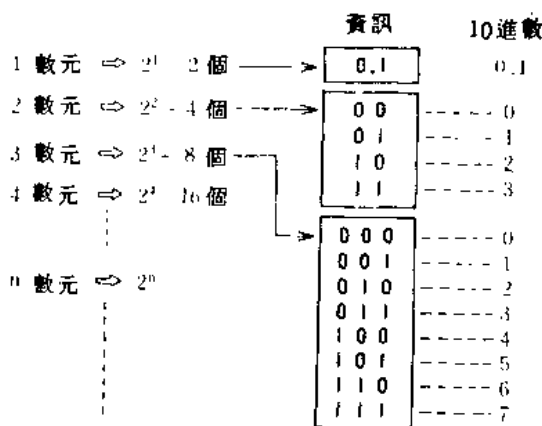
當我們還是小孩子的時候，記得也做過內容較天真的這一類占卦。譬如把木屨高高地向上拋，看它掉下來的時候出現表面或底面，以好玩的心情卜明天的天氣。

現在已經變成穿鞋的時代了，而因為鞋會橫倒，出現既不是表面也不是底面的中間狀態，所以這種占卦變成不太好做了。

因為電腦是電的機器，所以將以開關的 ON · OFF，就是以電流有否流通的狀態，構成這兩者選一的狀態。

② 數元和資訊量是什麼？

我們把表示這種兩者選一狀態的單位叫做數元 (bit)。



☆ 將 1 個數叫做數元

圖 1 數元和資訊量

數元 (bit) 係 binary digit 的縮寫，是意味着 2 進數。關於 2 進數，在次一頁將有說明。先簡單說，就是以“0”和“1”的兩種數表示兩者選一的方法。

譬如說，假如以“0”和“1”表示 1 枚硬幣的反面和正面，那麼所表示出來的兩種資訊就成為 0 或是 1 的 1 位數。我們將這種 1 位數叫做 1 數元。

其次，假如我們使用 2 枚硬幣，就變成怎樣呢？

兩枚硬幣的所有排列方法有：（反反），（反正），（正反），（正正）等 4 種類。也就是說，此時我們能傳達 4 種類的資訊。如果以“0”和“1”的 2 進數對應這些反・正面排列，就得以 00, 01, 10, 11 表示。

這種情形和 1 枚硬幣的情形不同。各別資訊是以 2 位數表示它。我們將這 2 位的 2 進數叫做 2 數元。

同樣的，當我們使用 3 枚硬幣時，也就是說以 3 數元表示時，就

會有多少種類的資訊數目呢？請各位讀者自己想一想看。

圖 1 係表示數元和資訊量之間的關係。

■ 10 淮數和 2 淮數

我們日常使用的數是屬於所謂的10進數。在10進數，當個位數成為10時就上至十位，十位數再成為10時就上進至百位。這樣當一個位數成為10時就依次上進一位。

這是因為在10進數中，有1的數位，10的數位，100的數位等定位（加權）的關係。

譬如說，以定位（加權）的方法寫出 351，就成為：

$$351 = 3 \times \underline{100} + 5 \times \underline{10} + 1 \times \underline{1} = 300 + 50 + 1$$

這裡的 $100 (=10^2)$, $10 (=10^1)$, $1 (=10^0)$ 就是數位(加權)。

那麼，當我們擴大這種想法時，就能以每位數成爲 2 就上進一位的方法表示一種數。

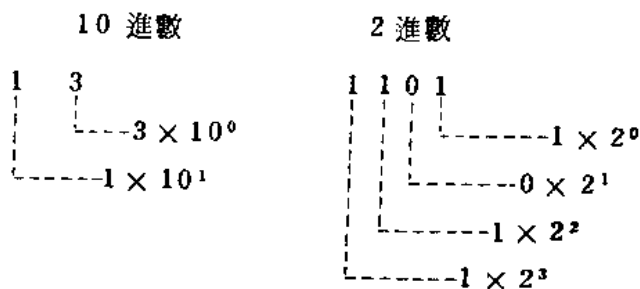
以這樣的方法表示出來的數就是 2 進數。

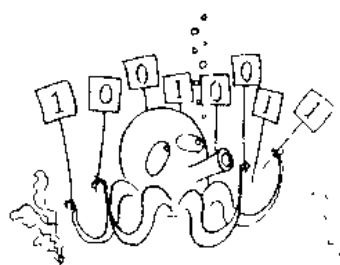
例如，以 2 進數表示 13，就等於：

$$13 = 1 \times \underline{8} + 1 \times \underline{4} + 0 \times 2 + 1 \times 1 = 8 + 4 + 0 + 1$$

這裡的 8, 4, 2, 1 或是 $2^3, 2^2, 2^1, 2^0$ 就是定位 (加權)。

因此，2 進數的 13，就能以 10 進數的時候一般，繼續使用 0 和 1，表示如下：





☆ 1 數元組



☆ 1 數元小組

③ 電腦資訊的單位

如同前述，當數元組愈增加，能傳達的資訊量亦愈增多。

譬如說，大型電腦是以 64 數元做為資訊量的 1 單位，因此能擁有 2^{64} ($\approx 1.84 \times 10^{19}$) 種類的龐大資訊量。

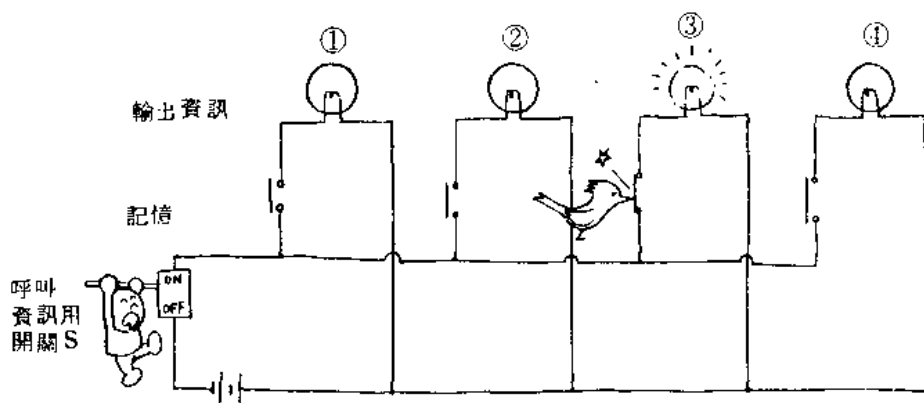
又，連小型化電腦的微電腦也以 16 數元做為單位，因此能傳達 2^{16} ($= 65536$) 種類的不同資訊。

在另一方面，因為微電腦的出生與迷你電腦不同，係以電子計算機為基礎而發展的，所以在誕生的當初是從 4 數元出發，然後慢慢地增加資訊量至 8 數元，12 數元，直到現在出現的如同迷你電腦一般以 16 數位做為單位的微電腦。

從上面的說明，我們能了解微電腦或是電腦是隨着其機械的大小而具有特有的資訊單位。我們把這個單位稱為語句 (word) 或是語句長。就是說，有些語句是把 4 個 0 和 1 排列而成的 4 數元，也有 8 數元做為 1 語句的。

又，以 4 數元做為一個單位，稱為 1 數元小組 (nibble)，以 8 數元做為一個單位，稱為 1 數元組 (byte)。

因此，例如在 16 數元的微電腦，1 語句是由 16 數元，也就是由 2 數元組構成。



④ 電腦的語言是“0”和“1”

大家都知道，電腦目前已經進入各式各樣的領域，進行超人的工作。很意外的，能進行相當複雜工作的電腦，其原理却很簡單，只是最單純的 ON，OFF 開關的組合而已。不過，因為有非常龐大數目的開關，依照目的構成邏輯的組合，所以不但看起來似乎很複雜，而且實際上亦能進行複雜的工作。

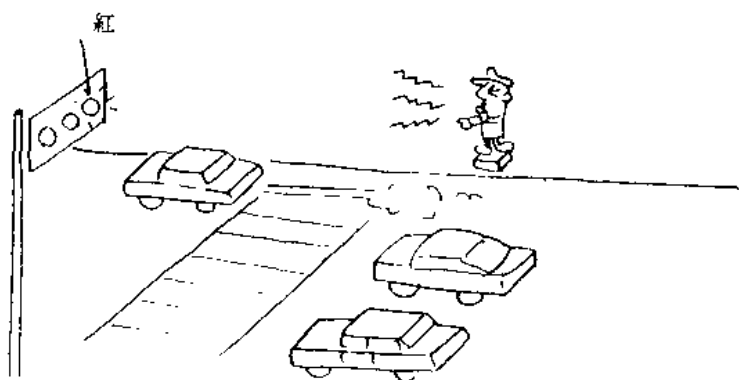
那麼，從這 ON，OFF 開關兩者選一的狀態，是能與 2 進數“1”和“0”相對應的。

因此，我們可以說，電腦係依據“1”和“0”的組合進行所有的工作。

反過來說，電腦是無法做“1”和“0”以外的表現，也無法了解“1”和“0”以外的語言。也就是說，電腦的語言是“1”和“0”。

譬如說，我們來思考上圖的點滅 4 只小燈泡的簡單電路。

在與電源並聯的 4 只小燈泡電路中，當我們按下電源開關 S（呼叫資訊開關）時，在①～④的接點之中，只有 ON 的③號小燈泡將點



☆ 信號是藉着約定成為資訊的

亮。

假如令點亮的小燈泡表示“1”，沒有點亮的小燈泡表示“0”，而從左側排列數字，那麼此時我們就等於輸出了〔0010〕的4數元資訊。

這就是表示我們將在下一章說明的記憶器（memory）的簡單原理。

因此，這電路就是等於藉著①～④接點的開、閉狀態，記憶〔0010〕的資訊。

我們再來看另外一個例子，就是紅、綠、黃的交通信號燈。我們預先知道“當紅燈亮的時候就要停止”“亮綠燈時就前進”等信號所表示的意義。

總而言之，我們先規定約定的事項，使信號變成資訊而具有意義。

到這一節為止，我們把“0”和“1”的排列稱為資訊。但是，正確地說，假如我們不對“0”和“1”的排列規定它所表示的意義，就不能稱這些排列為資訊。它只是一種沒有意義的信號而已。

因此，譬如說，我們需要規定4數元排列的內容，例如〔0000