

IBM PC 第二版

8088 組合語言程式

陳廷光 編譯

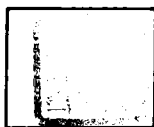


松崗電腦圖書資料有限公司

IBM PC

8088組合語言程式

版權所有



翻印必究

每本定價 180 元整

書號：2101104

譯著者：陳 廷 光

發行人：朱 小 珍

發行所：松崗電腦圖書資料有限公司

台北市仁愛路二段一一〇號三樓

總經理：松崗電腦圖書資料有限公司

台北市仁愛路二段一一〇號三樓

電話：3930249-3930255-3930258

郵政劃撥：0109030-8

印刷者：東崗印刷設計股份有限公司

台北市仁愛路二段一一〇號三樓

電話：3930249-3930255-3930258

中華民國七十三年二月初版

中華民國七十四年四月第二版

本出版社經行政院新聞局核准登記，

登記號碼為局版台業字第三一九六號

原著者序

今天無論在家庭或公司，微電腦的數量都以前所未有的速度增加。現有的科技，使得桌上型電腦也擁有甚強的演算功能，而成本却比一輛新車還低。市面上固然有琳琅滿目的軟體，可供這些機器執行各類工作，但是許多微電腦的使用者，最後却都希望設計他們自己的軟體。自製軟體的優點是可以針對作者本身的需求而適切剪裁。經過設計軟體後，程式師將更深入了解機器，而拉近他跟機器間的距離。

IBM 個人電腦可說是時下最新且功能最強微電腦之一，本書即是為介紹如何使用它設計機器語言軟體而寫的。連帶使本書討論的是組成此個人電腦心臟的今日科技晶華：Intel 8088微處理機。

熟悉諸如培基及巴斯可高階語言的程式師，將發現本書會引導您輕易的登進設計組合語言程式的更高藝術境界。有經驗的組合語言程式設計師，把 8088 納入其智識寶庫必將有所受益。

本書尚包括有關個人電腦內部結構的詳細資訊，對 IBM 個人電腦的程式設計師而言，這些資訊是很有價值的。

我們衷心感謝 Intel 公司及 Howard W. Sams & Co 公司的協助，使本書得以付梓。

David C. Willien
Jeffrey I. Krantz

目 錄

第一章 引 言	1
使用培基語言的困難.....	1
有權則有責.....	3
前 瞻.....	3
第二章 基本概念	5
資訊之電子表達方式.....	5
二進數與十六進數加法.....	11
儲存資訊於主記憶體.....	14
中央處理器的角色.....	15
爲何我們需要組合語言.....	16
第三章 8088結構	21
8088 暫存器組	22
記憶地址分段.....	23
8088 指令集	28
資料定址模式.....	28
堆疊動作.....	31
輸出入與其他的資料傳遞指令.....	33
算術指令與旗誌暫存器.....	35

邏輯指令.....	38
字串操縱指令.....	42
控制移轉指令.....	50
無條件跳躍指令.....	51
條件跳躍指令.....	54
重複控制指令.....	57
副常式.....	59
中斷 (Interrupt)	62
處理器控制指令.....	65
結 語.....	66

第四章 BIOS. DOS. 與巨編譯器..... 67

開 機.....	67
執行我們的程式.....	68
代擬 (Pseudo) 指令.....	71
定義資料之代擬命令.....	72
程序 (Procedures)	73
段定義與定址.....	75
DOS 連接法則	77
程式實例.....	78
建立程式.....	82
BIOS 常式	86
編譯器算子 (ASSEMBLER .OPERTORS)	87

第五章 PC系統電路板..... 91

匯流排 (BUS) 觀念	91
----------------------	----

主記憶支援.....	93
系統支援裝置.....	95
8259 中斷控制器	97
8255 可程式規劃之週邊中介	99
鍵 盤.....	101
8253 計時器 (TIMER)	109
產生聲音效果.....	120
結 論.....	126

第六章 單色、彩色/圖形及印字機匹配器..... 127

單色顯示.....	127
6845 CRT 控制器	135
彩色 / 圖形監視器匹配器.....	151
文數字符模式.....	153
圖形模式.....	163
印字機中介.....	174

第七章 串聯通訊..... 179

串聯與並聯之比較.....	179
異步串聯通訊協定.....	180
通用異步發送接收器 (UART)	182
調變解調器 (MODEM)	186
實體中介.....	188
串聯 I / O BIOS 呼叫	190
程式之規劃 8250	192
起始 8250	194

與 8250 通訊	197
8250 中斷	198
簡單的終端機程式一例	201
第八章 磁碟輸出入	207
磁片剖析	207
磁片讀寫技巧	209
BOS 管理下的磁碟 I/O	209
循序讀寫實例	215
透過 BIOS 的磁碟 I/O	221
BIOS 磁碟 I/O 實例：讀取檔案目錄	224
附錄A	231
8088 指令集	231
附錄B	251
參考資料	251

第一章 引 言

IBM公司於 1981 年八月推出其個人電腦，暱稱為“IBM PC”。IBM在此型個人電腦中，引入幾種相當進步的設計；其中最重要的，就是採用 Intel 8088 微處理機作中央處理單元。

8088 雖然為 8 秘（bit，俗譯為位元、數元、或筆，本書採音譯）之晶片，但它所能支援的架構，却直追其大哥 16 秘的 8086。因此這種個人電腦（以下簡稱 PC）遂具有使用高達 1024KB 大量記憶的能力；而大多數的微電腦充其量也只能處理 64KB 而已。IBM乃利用此特點，在 PC 上提供一强大有力的培基語言解譯器，燒製於唯讀記憶（ROM）內。無怪乎大多數的使用者都採用培基語言為 PC 設計程式。

使用培基語言的困難

誠然，培基語言易學易用，但亦有其缺點。培基語言的程式設計師，把 PC 當成能夠執行培基陳述（statement）和函數的電腦。其實 PC 只能執行 8088 中央處理器所提供的功能而已。機器語言遠比我們所熟悉的培基陳述原始。例如，培基陳述的 PRINT，就是由上百道機器指令組成的。當電腦執行培基程式時，均先把收到的每一道陳述先予以解譯；然後由一連串適切的機器指令達成該陳述所規定的功能，此過程稱為解譯執行（interpretive execution），如圖 1-1 所示，其先天的缺陷是速度太慢。

2 8088 組合語言程式

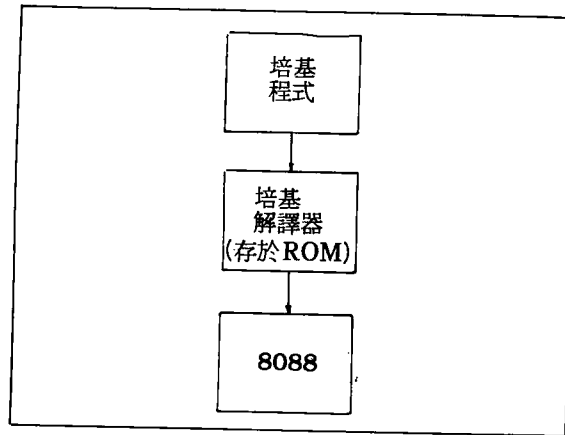


圖 1-1 解譯執行方式

假如我們用機器語言寫成一程式，則此程式將可直接由 8088 中央處理器執行。這種機器語言程式，將比相同功能的培基程式快上數倍，因為每一道培基陳述所須的解碼步驟，在機器程式中都省掉了。此種情況例示如圖 1-2，由機器指令組成的程式，是用組合語言（*assembler language*）寫成的。

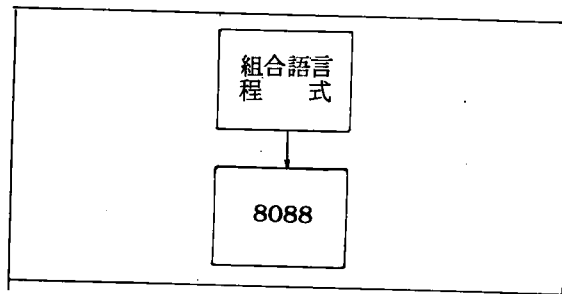


圖 1-2 編譯器 - 程式執行方式

執行速度較快並不是組合語言的唯一優點。如前所提，培基程式師把電腦視作培基電腦，意即程式師只能在培基語言所提供的特性範圍內施展身手。組合語言程式師則從最低階看待電腦，故能充分應用

硬體所包含的一切特性。看到自己所寫的組合語言程式，在 PC 上跑（即執行之謂），將是一種極為賞心悅目的經驗；因為，這時你知道促使機器工作的每一個電路，都在你的掌握之下。

有權則有責

組合語言的能力固然強大，其操縱運用是需要一番代價的（天下沒有不勞而獲的事情）。欲以組合語言運用自如的寫程式，非對電腦的內部組件嫻熟於胸不可；其中最重要者，首推 8088 微處理機。我們將先研究 8088 的內部結構，再學習它所能執行的種種指令。

PC 系統內尚有許多其他的組件，也是我們所必須研究的。例如，有一只積路晶片，可用來精確地測定事件的時間；它是由程式控制的，質言之，在程式的控制下，該晶片可執行各種不同的計時功能。然而，要能夠控制它，除了須要了解其內部結構外，它如何跟其他部門結合成一整體系統，也應曉暢。此外還有許多控制揚聲器（喇叭）、處理中斷（interrupt）、保持螢幕顯示等工作之特殊晶片或電路，若我們要控制它們，則對渠等非加以徹底認識不可。

再者，我們尚須明白系統支援程式。這些程式是隨同 PC 一併出售的，其中有些永久性的儲存於機器內的 ROM，其餘的部分則提供於 DOS 磁片上。這些支援程式建立起的軟體環境，使我們得以在 PC 上寫出並執行我們的組合語言程式。我們所寫的任何程式，都必須在此環境下執行，因此我們應該領悟它。

前 瞻

以下各章將涵蓋上面所提及的所有課題。使用本書毋須具備組合

語言經驗，但是讀者應熟識設計電腦程式所涉及的概念，以及至少通曉一種程式語言，如培基亦可。

第二章討論二進位與十六進位數目系統，並介紹組合語言的結構。若讀者已精通別種機器的組合語言，則讀者或許可跳過此章（應請注意者：許多常見組合語言的慣例並不適用於 8088）。

8088 的結構敘述於第三章。此處我們將網羅暫存器、記憶定址技巧、以及此強力微處理機之指令集（instruction set）。我們將在第四章學習如何使用 IBM 的巨組譯器（Macro Assembler）。我們也將以組合語言程式師的觀點，探討 IBM DOS。程式實例從本章開始陸續出現，這些程式例可實際輸入你的 PC 並加以執行；所需硬體包括至少 64KB 的記憶、一個磁碟機、單色匹配器（monochrome adapter）與螢幕、以及 IBM DOS 和 IBM 巨組合組譯器程式。讀者應熟練於標準 DOS 之操作（諸如磁碟檔案之建立與維護）。瞭解 LINED 程式如何輸入（建立）及修改程式內容，也是應該具備的。

第五章描述 PC 系統電路板，此板是個人電腦的心臟。我們將在這一章，學習如何控制機器的中斷結構、鍵盤以及計時器（timer）機構。第六章包括單色及彩色顯示匹配器。欲使程式例能顯示彩色結果，除了匹配器外，當然還需系統具備彩色螢幕才行。

第七章剖析串聯通訊（serial communication）的主要觀念，並說明如何使用異步（asynchronous）通訊匹配器。具有跟遠處電腦通訊的能力，將使 PC 的應用邁入嶄新的境界。此章有一程式例把 PC 轉換成一資料終端機，此程式須配合異步通訊匹配器才能執行。

最後，第八章專注於磁碟之輸入與輸出。兩種不同層次的磁碟操作，將在此討論。較高的層次讓我們能操縱標準的 DOS 磁碟檔，較低的層次則讓我們能直接進入磁碟的磁軌（track）與磁節（sector）存取資料。〔註：磁節或譯磁區、磁扇〕

第二章 基本概念

如前所述，組合語言的程式設計師，需深切了解電腦的內部組件，才能勝任。我們將在這一章研究流行微電腦的基本組成方塊，從而更深入的認識這些微電腦是怎樣工作的。

今日的微電腦可按其邏輯功能分成四個基本部分，如圖 2-1 所示

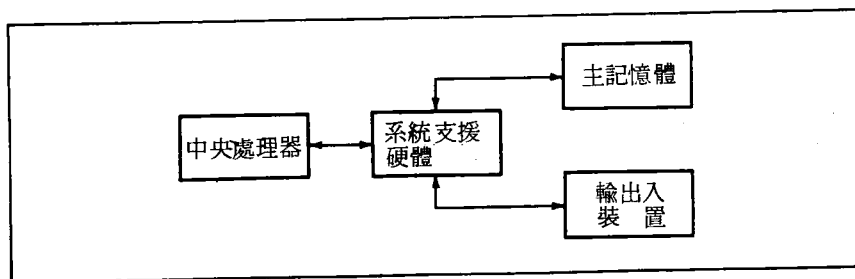


圖 2-1 微電腦的基本部分

。中央處理器是系統的心臟，構成電腦程式的指令，就是由中央處理器擷取及執行的。中央處理器的資料出入通路由系統支援硬體提供，系統支援硬體也掌管諸如追蹤中斷，供應電腦時鐘“脈搏”等各種雜事。主記憶用來儲存資訊，俾讓中央處理器隨時存取。輸出入裝置提供系統與外部世界之中介（interface）。鍵盤與螢幕是最常見的輸出入裝置，不過，大容量資料儲存裝置如磁碟和磁帶機，也屬於輸出入裝置的範疇。

資訊之電子表達方式

描繪於圖 2-1 之微電腦系統已能迅速的、精確的操縱及儲存資訊。爲了在系統之下發展及執行正確的程式，我們應知曉資訊在系統內是如何表示的。

我們均習於使用十進制數目系統，表達數目資訊。十進制的基本單位“十進制數字”(decimal digit)，由 0、1、2、3、4、5、6、7、8、9 十個數目字組成。但是在電腦內，資訊之儲存係以電子信號之存在與否來代表，所以僅可能有兩個不同的數字，而不是十個。我們稱此兩個數字爲 0 與 1。電腦內的所有資訊，都是用此兩種不同數字的種種組合儲存的。

因爲只有兩個數字，故我們說電腦之數目系統爲二進制，是以我們使用秘(bit, binary digit 之縮字)代表電腦資訊的儲存單位，換言之，秘之值只能爲 0 或 1 二者而已。

顯然的，我們必須使用多個秘才能用來代表 1 以上的數目。假如我們把兩個秘擺在一起，則我們可以獲得四種不同的組合：兩秘同爲 1、兩秘同爲 0、前秘爲 0 後秘爲 1、及前秘爲 1 後秘爲 0。若我們加上第三秘，則我們將有八種不同的組合，因而可用來代表 0 到 7 的數目。顯而易見的，每增加一秘，則所能代表的數目範圍亦跟著倍增；是故四秘時可代表到 16 種不同的數值(圖 2-2)。

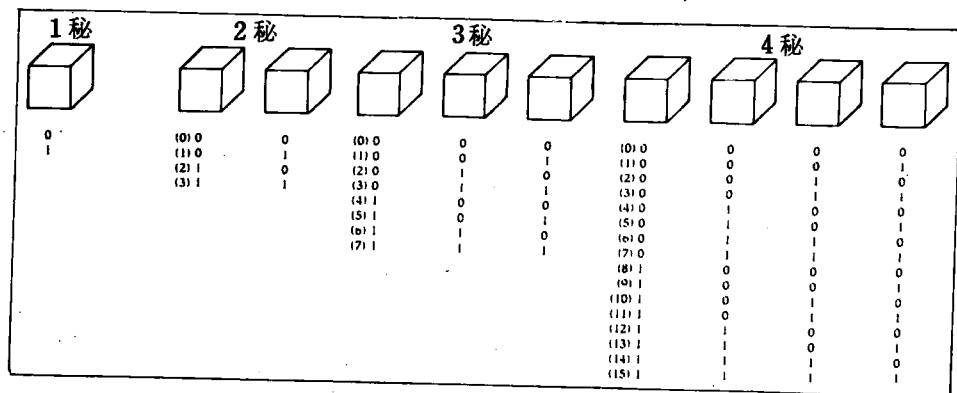


圖 2-2 用秘組合代表數目資料

一旦我們定下代表某種資訊的秘數，則所能代表該資訊的總項數，即隨著確定。這種不同的秘組合可用來代表從 0 至某一上限的數目，如圖 2-2 所示。但是秘組合並不僅僅限制於數目的代表而已。例如，設某物體的顏色可能為紅、藍、綠、黃四色之一，而我們欲儲存其顏色值，則我們可使用兩個秘代表，如圖 2-3 所示。圖中，利用四種不同的秘組合所產生的各值，分別代表一顏色。

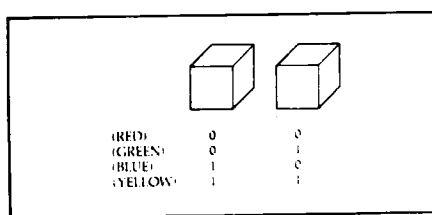


圖 2-3 用 2 秘代表四種不同的彩色

若干年前，前輩們很明智的決定，若要電腦真正實用，則必須讓我們能以通用的文字與符號而不是永無止盡的數目串，來跟電腦通訊。欲達成此目的，需要一套編碼方法使能分別代表各個英文字母、數目字、及標點符號；此三者合計約有一百個不同的字符。採用七個秘的組合，可代表到 128 個值，如表 2-1 所示，把大小寫字母、數目字、標點符號、及若干控制碼等均包羅在內。這些字符的代表碼業已標準化，且為業界所通用，它們就是所謂的美國標準資訊交換碼，或簡作 ASCII。此種標準化，帶來莫大的好處，因為實用上，任何一部電腦都將認識 1000001 的二進位碼為大寫的字母 A。

把上述的七秘 ASCII 碼再加一秘，我們將獲得所謂的“拜”（byte，一般譯作位元組或數元組，本書採音譯），電腦系統最常提及的資訊儲存單位。八秘之拜含 256 種不同之值，其前半段可用來代表 ASCII 之字符集合。拜也可用來代表 0 到 255 的數目範圍。當我們需要代表較大的數目時，我們常把兩拜組成一對，形成所謂的“字元”

表 2-1 美國標準資訊交換碼 (ASCII)

二進碼	十六進	字符	二進碼	十六進	字符	二進碼	十六進	字符
0000000	00H		0110000	30H	0	1100000	60H	‘
0000001	01H		0110001	31H	1	1100001	61H	a
0000010	02H		0110010	32H	2	1100010	62H	b
0000011	03H		0110011	33H	3	1100011	63H	c
0000100	04H		0110100	34H	4	1100100	64H	d
0000101	05H		0110101	35H	5	1100101	65H	e
0000110	06H		0110110	36H	6	1100110	66H	f
0000111	07H	<BELL>	0110111	37H	7	1100111	67H	g
0001000	08H	<BKSP>	0111000	38H	8	1101000	68H	h
0001001	09H	<TAB>	0111001	39H	9	1101001	69H	i
0001010	0AH	<LF>	0111010	3AH	:	1101010	6AH	j
0001011	0BH		0111011	3BH	:	1101011	6BH	k
0001100	0CH		0111100	3CH	<	1101100	6CH	l
0001101	0DH	<CR>	0111101	3DH	=	1101101	6DH	m
0001110	0EH		0111110	3EH	>	1101110	6EH	n
0001111	0FH		0111111	3FH	?	1101111	6FH	o
0010000	10H		1000000	40H	@	1110000	70H	p
0010001	11H		1000001	41H	A	1110001	71H	q
0010010	12H		1000010	42H	B	1110010	72H	r
0010011	13H		1000011	43H	C	1110011	73H	s
0010100	14H		1000100	44H	D	1110100	74H	t
0010101	15H		1000101	45H	E	1110101	75H	u
0010110	16H		1000110	46H	F	1110110	76H	v
0010111	17H		1000111	47H	G	1110111	77H	w
0011000	18H		1001000	48H	H	1111000	78H	x
0011001	19H		1001001	49H	I	1111001	79H	y
0011010	1AH		1001010	4AH	J	1111010	7AH	z
0011011	1BH		1001011	4BH	K	1111011	7BH	{
0011100	1CH		1001100	4CH	L	1111100	7CH	
0011101	1DH		1001101	4DH	M	1111101	7DH	}
0011110	1EH		1001110	4EH	N	1111110	7EH	~
0011111	1FH		1001111	4FH	O	1111111	7FH	Δ
0100000	20H	space	1010000	50H	P			
0100001	21H	!	1010001	51H	Q			
0100010	22H	"	1010010	52H	R			
0100011	23H	#	1010011	53H	S			
0100100	24H	\$	1010100	54H	T			
0100101	25H	%	1010101	55H	U			
0100110	26H	&	1010110	56H	V			
0100111	27H	'	1010111	57H	W			
0101000	28H	(	1011000	58H	X			
0101001	29H	)	1011001	59H	Y			
0101010	2AH	*	1011010	5AH	Z			
0101011	2BH	+	1011011	5BH	[			
0101100	2CH	,	1011100	5CH	\			
0101101	2DH	.	1011101	5DH	]			
0101110	2EH	:	1011110	5EH	^			
0101111	2FH	/	1011111	5FH	_			

註：控制碼用<>表示 (BKSP = backspace, TAB = tabulate, LF = line feed, CR = carriage return).

或“電腦字”(word)。字元由 16 秘組成，含 65536 種不同之值，常用來代表 0 到 65535 的整數範圍。秘、拜、字元三者之比較示於圖 2-4。注意到，秘的編號係從右而左，且從數目 0 開始。當我們要

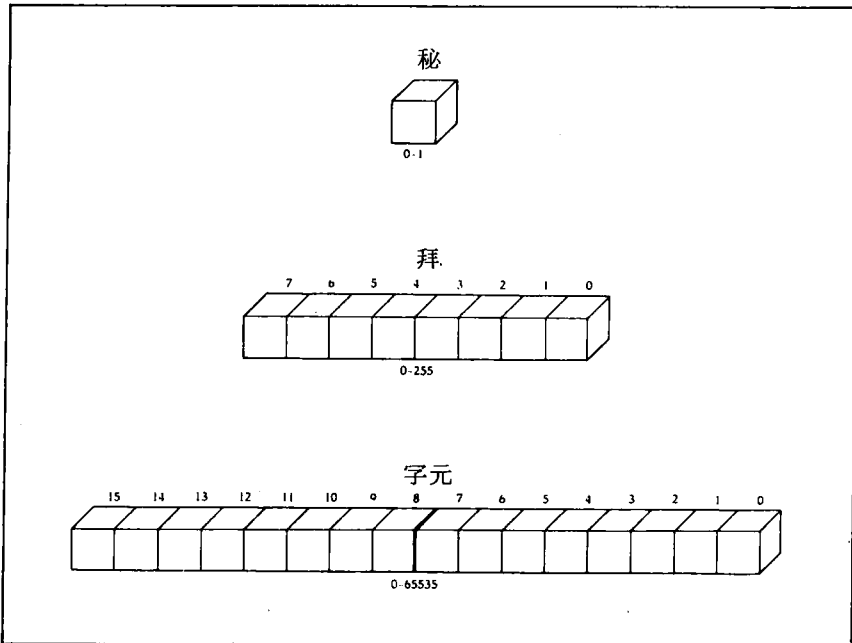


圖 2-4 資訊儲存單位

指明一拜或一字元內的某一秘時，此種編號方式即為我們所將使用者。

我們採用二進制數目以定義每一種秘組合（示於圖 2-4 者）的數值。我們日常所習用的十進制，由十種不同的數目字組成，而各數目字在不同的位置（如個位、十位、千位、萬位等）就代表不同的值（ 10 的指數次方）。是以十進制的 916 是由百位數的 9（ 9×10^2 ）加十位數的 1（ 1×10^1 ），再加上個位數的 6（ 6×10^0 ）而成。在二進制中，只有兩種不同的數字，因此每一秘位分別代表 2 的某一乘幕值。例如二進位數目 101 是由 2^2 位置的 1，加上 2^1 位置的 0，以及 2^0 位置的 1 所組成的。二進數 101 的十進制等值是 5，如圖 2-5

所示。

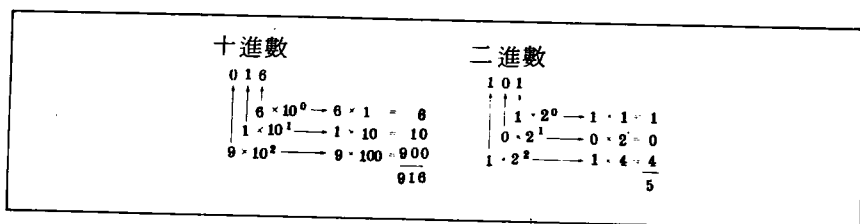


圖 2-5 十進制與二進制

從以上的敘述可看出，當數目愈大時，所需用來代表的秘數就愈多。為避免長串 0 與 1 的夾纏不清，我們乃使用十六進制作為速記。十六進制（常以 Hex 或 H 表示）是以 16 為底的數目系統，因此有十六個數字，如圖 2-6 所示。注意到其前十個數字，仍沿用常見的數目

十六進位	4 秘二進制等值	十進制等值
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

圖 2-6 十六進制

字 0 ~ 9，剩餘的六個則借用前六個英文字母，是故，16 進制之 A 相當於 10，B 相當於 11，以下類推，至最後一數字 F 則相當於 15。16 進制的每一數位代表該數字乘以 16 的指數次方值。故從右而左，我們分別稱呼各數位為 16^0 位、 16^1 位、 16^2 位...等等。為區別起見，本書將在數目之後附上 H，以代表十六進數。例如 2FH 代表 2 乘以 16 加上 F（15）乘以 1，等於十進制之 47。當遇及由數目字 A ~ F 帶頭的十六進數時，我們將在其前再加一前導 0（如 0A2H、0FFH 等）