

電腦 2000

第二冊



郭永強

613195074
Hkd 86

G634.67/035

wres
DA
76
K861
1999
v.2
G2

電腦 2000

第二冊



二冊 (國中)



郭永強

B.Sc., B.Ed., M.Sc., DIC, SMACE, MBCS



© 郭永強 1999

版權所有

如未經本出版社書面同意，不得以任何方式翻印或
轉載本書中任何部分之文字及圖片

英文版 1995 年初版
1999 年第二版

中文版 1999 年初版

ISBN 962 03 2063 8

出版

麥克米倫出版 (中國) 有限公司
香港九龍觀塘鴻圖道 51 號
保華企業中心十八樓 1812 室
電話：2811 8781

印刷 香港

前言

本課程是根據香港中學會考電腦科的要求而編寫；課程分為三部份：第一冊和第二冊覆蓋信息處理及電腦系統，而第三冊則討論 Pascal 程序編寫。

在編寫本書的過程中，本人盡力糾正類似教材中一些常見的弱點。本人還盡力引入一些新的特點，希望同樣受到老師和同學的歡迎。

課本內容

教授電腦科概念的次序，經常引起爭議。本人在本書中稍微偏離了課程綱要中的教學次序，相信這會使教學和學習變得更容易。本人還特別強調在將來考試中會佔重要地位的課題，例如網絡與數據庫等。

本書的特點

本書尤其著重練習題的設計；每章末端共有三種類型的練習題：關鍵概念問題、結構問題及歷屆的香港中學會考試題。

在第二冊的最後三章中，本人略作改動，引入「處境任務」，以鼓勵學生觀察電腦在當今社會的應用情況。另有一節專門討論「個案研究」。

關鍵概念問題： 這些練習測試學生對各章所出現的關鍵術語和關鍵概念的理解程度。如果學生能夠回答這些問題，便可以確信自己已掌握了有關章節的基本知識。練習題還提供了有關句子結構方面的訓練。

結構問題： 這些習題著重日常的電腦概念和應用。練習通常與學生的日常生活有關，使這些練習題又生動又有趣。更重要的是，這些練習還反映了香港中學會考的重點改變。

第二冊中的 **個案研究** 是香港中學會考的特色。本書有專門一節介紹這方面的例題。這些練習按難易程度分級，因此學生可以在整個學習過程中不斷練習。

致謝

感謝下列人士在本書的編寫過程中所給予的支持與協助。

LAU HO KWAN、LEE CHEUK HUNG、LEUNG YIU KEI、YUE MEI FONG

作者和出版社衷心感謝為本書提供精彩圖片之有關機構、團體和個人：

Hong Kong and Shanghai Banking Corporation Limited

Rainbow

Ronald Grant Picture Library

Science Photo Library

The Image Bank

作者和出版者衷心感謝香港考試局，准許本書轉載歷屆香港中學會考的試題。

其他照片由以下人士提供：Chung Wing Keung 及 Sammy Ho。

為了聯絡所引圖片的版權擁有人，我們已盡了最大努力，但倘若到目前為止仍無法和有關的版權擁有人接觸，以致某些圖片的版權問題懸而未決，我們將願意和合法版權人以誠懇態度合理解決。

目錄

言簡

10 數據表示法

10.1	用二進制數將信息編碼	1
10.2	二進制數系	3
10.3	將十進制數轉換為二進制數	4
10.4	其他數系	5
10.5	二進制小數	7
10.6	整數的表示	8
10.7	定點表示法	11
10.8	浮點表示法	12
10.9	字母數字數據和中文字的表示	14
10.10	奇偶檢查	16

11 電腦操作

11.1	一個簡單電腦模型	21
11.2	指令週期	25
11.3	機器碼	26
11.4	中央處理器的操作	27
11.5	匯編語言	30
11.6	更多關於匯編程序的資料	31

12 程序語言

12.1	低級語言	47
12.2	高級語言	49
12.3	高級和低級語言的比較	51
12.4	翻譯程序	52
12.5	程序語言的發展趨勢	53

13 操作系統

13.1	操作系統的必要性	59
13.2	電腦的分層觀點	60
13.3	操作系統的功能	61
13.4	操作系統的類型	61
13.5	微型電腦操作系統	65

14 微型電腦應用軟件包

14.1	一般軟件包常見的功能和操作	70
14.2	中文輸入法	74

14.3	文字處理	76
14.4	試算表	82
14.5	數據庫管理系統	88
14.6	圖形	92
14.7	互聯網	97

15 電腦在現代世界中的應用 103

15.1	電腦的應用	103
15.2	未來的電腦	114

16 電腦對社會的衝擊 119

16.1	無需鈔票的社會	119
16.2	電腦與就業	120
16.3	電腦與健康	121
16.4	易用的電腦系統	122
16.5	信息的表達	124
16.6	關於電腦道德的問題	126

個案研究 133

1	影視中心	133
2	陳醫生的診所	134
3	文林書店	135
4	電子轉賬	136
5	電費單	137
6	旅行團訂位系統	137
7	學校中的電腦	138
8	ABC 化妝品店	139
9	宇宙書店	141
10	舊生會	142
11	班會	145
12	NTS 程序	147

附錄 A 150

A.1	存貯程序的概念	150
A.2	存貯數據的概念	154
A.3	電腦的發展	157

附錄 B 162

索引 163

練習題答案 165

個案研究的答案 170

電腦處理兩種基本的數據類型：**數字數據**和**字母數字數據**。數字數據指那些能作算術處理的數字，而字母數字數據則是由一組包括字母、數字和特殊字符(例如標點符號)，等字符所組成的。

本章說明不同的數據是如何在電腦系統表示。

10.1 用二進制數字將信息編碼

電腦中的所有設備都是雙態設備。例如，開關不是「開」就是「關」，一個信號是一個脈沖或沒有脈沖(見圖 10.1)。兩種狀態通常由「1」及「0」表示，「1」代表「開」，而「0」代表「關」。0 或 1 稱為二進制數字或**位**，因為只有兩種選擇。一系列的二進制數字，也叫**位模式**，可以表達一組信息。

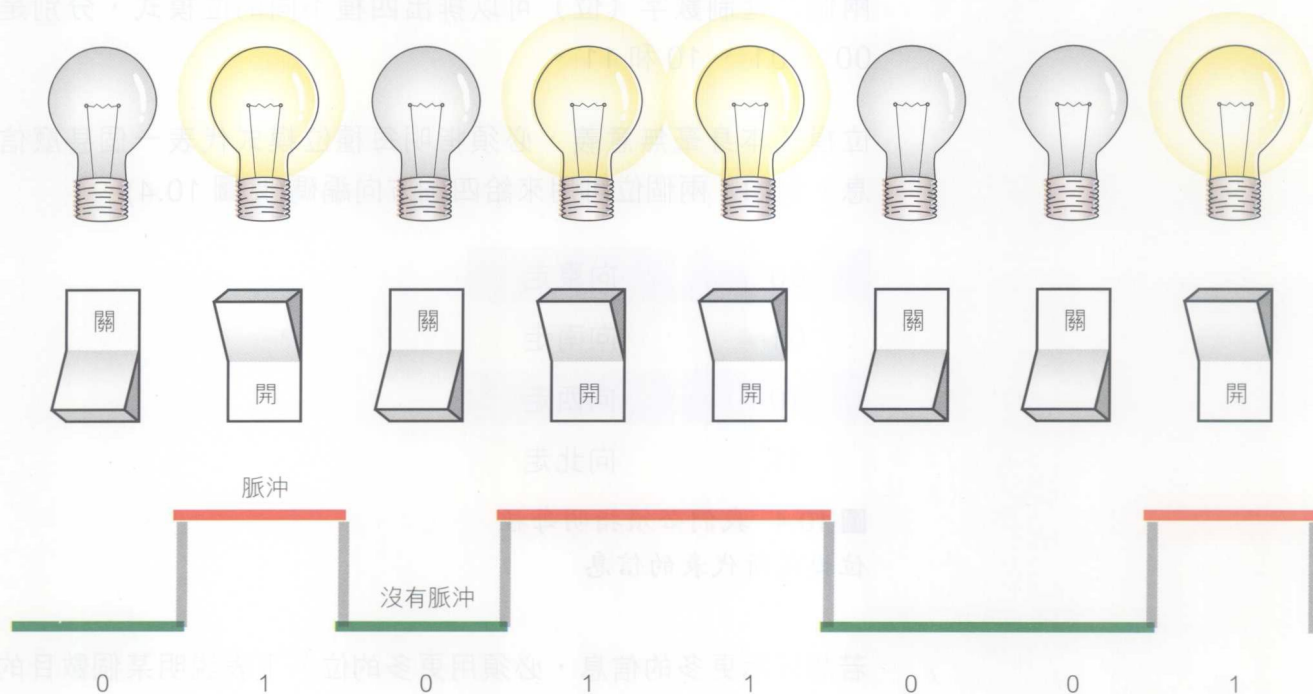


圖 10.1 雙態設備。開關不是「開」就是「關」

為了理解位模式如何表達信息，試假設有人利用一個燈泡向另一個人傳送信號。他可以傳送兩種不同的信號：一個信息用開燈來表示，而另一個信息用關燈來表示(見圖 10.2)。

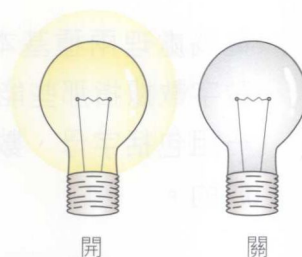


圖 10.2 用一個燈泡可以發送兩種信號。「開」燈表示一件事情，「關」燈表示另一件事情

若他有兩個燈泡將會如何？他可以發送四種不同的信號(見圖 10.3)。



圖 10.3 用兩個燈泡可以傳送四種不同的信號

兩個二進制數字（位）可以排出四種不同的位模式，分別是 00、01、10 和 11。

位模式本身毫無意義，必須指明每種位模式代表一個甚麼信息。例如，兩個位可用來給四個方向編碼(見圖 10.4)。

00	向東走
01	向南走
10	向西走
11	向北走

圖 10.4 我們必須指明每種位模式所代表的信息

若想表示更多的信息，必須用更多的位。下表說明某個數目的位所能代表的最大代碼數目。

位數	最大代碼數目	
1	2	$= 2^1 = 2$
2	2×2	$= 2^2 = 4$
3	$2 \times 2 \times 2$	$= 2^3 = 8$
4	$2 \times 2 \times 2 \times 2$	$= 2^4 = 16$
5	$2 \times 2 \times 2 \times 2 \times 2$	$= 2^5 = 32$
6	$2 \times 2 \times 2 \times 2 \times 2 \times 2$	$= 2^6 = 64$
7	$2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2$	$= 2^7 = 128$
8	$2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2$	$= 2^8 = 256$

10.2 二進制數系

在日常生活中，我們最熟悉的是十進制數系，所用的數字為 0、1、2、3、4、5、6、7、8、9。十進制數字 1628 代表八個一、二個十、六個一百和一個一千。

即是

$$1628_{(10)} = 1 \times 10^3 + 6 \times 10^2 + 2 \times 10^1 + 8 \times 10^0$$

	千位	百位	十位	個位
位值	1000	100	10	1
數字	1	6	2	8

我們可以看出，任何數字的值根據它在數字中的位置來決定。十進制數的位值是個位、十位、百位、千位等等(從右開始)。

在電腦中，數字數據用二進制形式表示。二進制數與十進制數的組成方式相同，不同的是位置上的 10 的乘方被 2 的乘方所代替，它的位值是個位、兩位、四位、八位、十六位等等。

	2^5	2^4	2^3	2^2	2^1	2^0
位值	32	16	8	4	2	1
二進制數字	1	0	1	1	1	0

$$\begin{aligned}
 101110_{(2)} &= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^1 + 0 \times 2^0 \\
 &= 32 + 0 + 8 + 5 + 2 + 0 \\
 &= 46_{(10)}
 \end{aligned}$$

10.3 將十進制數轉換為二進制數

從前一節的計算得知，只需把二進制數中所有數字為 1 的位值相加，便可求出等同的十進制數。現在將討論把十進制數轉換成與之相等的二進制數的兩種方法。

方法一

此方法利用簡單的減法。假設把 50 轉換成二進制數。

- 1 寫出一列位值，而最大的值剛好超過你要轉換的數(請看下面)。

64 32 16 8 4 2 1

- 2 在能夠從原來的數中減去的最大數下寫上 1。在這例子中，將 50 減去 32，得出餘數 18。

64 32 16 8 4 2 1
1

- 3 移到另一可以從餘數 18 減去的位值。這次，減去 16 後，餘數為 2。繼續此步驟直到最後的餘數為 0。

64 32 16 8 4 2 1
1 1 1

- 4 在未加數字的位值上寫上 0，於是 $50_{(10)} = 110010_{(2)}$

方法二

另一種方法是重覆除以 2 直到商為 0。在此處用了整數除法，餘數記錄如下。

2) 50	餘數為 0	
2) 25	餘數為 1	
2) 12	餘數為 0	
2) 6	餘數為 0	
2) 3	餘數為 1	
2) 1	餘數為 1	
0		

於是 $50_{(10)} = 110010_{(2)}$

等同的二進制數是從下向上讀的餘數列。

10.4 其他數系

對電腦來說，用二進制數工作既容易又方便，但對人來說則比較難。由於人們處理較長的位模式容易出錯，所以會採用八進制和十六進制表示法，來作為二進制的簡略表達方式。

八進制數系

之所以稱為八進制數系是因為這系統僅用 8 個數字：0、1、2、3、4、5、6、7。八進制的位值是 8 的乘方。

位值	8^2	8^1	8^0
八進制數字	3	4	6

$$\begin{aligned}
 346_{(8)} &= 3 \times 8^2 + 4 \times 8^1 + 6 \times 8^0 \\
 &= 192 + 32 + 6 \\
 &= 230_{(10)}
 \end{aligned}$$

與八進制數相應的二進制數如下所示：

八進制數	0	1	2	3	4	5	6	7
二進制數	000	001	010	011	100	101	110	111

十六進制數系

在十六進制數系中，用到了 16 個符號：0 至 9 和 A 至 F (A 表示 10，B 表示 11，如此類推直到 F 表示 15)。十六進制的位值是 16 的乘方。

位值	16^2	16^1	16^0
十六進制數字	1	F	A

$$\begin{aligned}
 1FA_{(16)} &= 1 \times 16^2 + 15 \times 16^1 + 10 \times 16^0 \\
 &= 256 + 240 + 10 \\
 &= 506_{(10)}
 \end{aligned}$$

與十六進制數對應的二進制數如下所示：

十六進數	0	1	2	3	4	5	6	7
二進制數	0000	0001	0010	0011	0100	0101	0110	0111
十六制數	8	9	A	B	C	D	E	F
二進數	1000	1001	1010	1011	1100	1101	1110	1111

二進制、八進制、十六進制數之間的轉換

二進制與十六進制間的轉換非常容易。要把二進制數轉換成十六進制數，只需要將二進制數從右向左，分成四個數字一組，然後寫下與每組四位相應的十六進制數即可（請看下面的例子）。

範例 1

把 $11011011000_{(2)}$ 轉換成十六進制數。

$\underbrace{110}_{6}$ $\underbrace{1101}_{D}$ $\underbrace{1000}_{8}$

從右向左分成 4 個數字一組

每組對應的十六進制數字

把二進制數轉換成八進制數與前者相似，不過這次是分成三個數字一組。

範例 2

把 $5AF_{(16)}$ 轉換成二進制數，然後轉成八進制數。

$\underbrace{101}_{2}$ $\underbrace{1010}_{6}$ $\underbrace{1111}_{7}$

把十六進制數字分開

與之相應的二進制數

每組 3 位的八進制數字

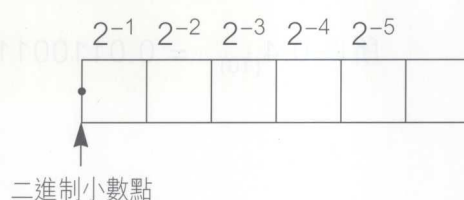
10.5 二進制小數

對整數來說，十進制的小數，可以表示成 10 乘方的和的形式。例如：

$$0.345_{(10)} = 3 \times 10^{-1} + 4 \times 10^{-2} + 5 \times 10^{-3}$$

相似的原則也可用在二進制小數中，例如：

$$\begin{aligned} 0.1101_{(2)} &= 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} \\ &= 0.5 + 0.25 + 0 + 0.0625 \\ &= 0.8125_{(10)} \end{aligned}$$



$$2^{-1} = 0.5$$

$$2^{-2} = 0.25$$

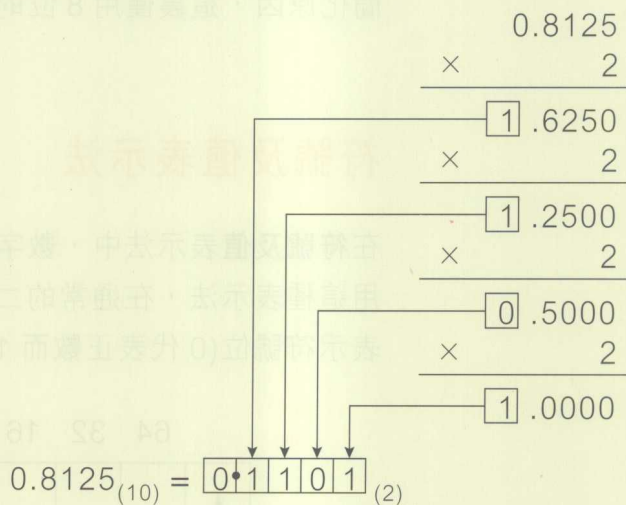
$$2^{-3} = 0.125$$

$$2^{-4} = 0.0625$$

$$2^{-5} = 0.03125$$

要把一個十進制小數轉換成二進制數，需要反覆乘以 2。假設我們要把 0.8125 轉換成二進制數。

把 0.8125 轉換成二進制數。



每次相乘之後，在小數點之前的數字(0 或 1)，將成為二進制小數的下一位值。只有小數部分在每一遍中乘 2，直到小數部分變成 0。下頁有另一個例子。

把 0.4 轉換成二進制小數。

$$\begin{array}{r}
 0.4 \\
 \times 2 \\
 \hline
 0.8 \\
 \times 2 \\
 \hline
 1.6 \\
 \times 2 \\
 \hline
 1.2 \\
 \times 2 \\
 \hline
 0.4
 \end{array}$$

我們又回到了原來的情況，其中十進制小數是 0.4。這 0110 的模式將一遍一遍地重覆。

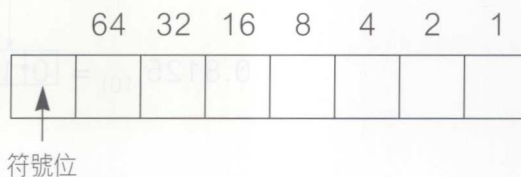
所以 $0.4_{(10)} = 0.011001100110 \dots$

10.6 整數的表示

本節和接下來的兩節將介紹電腦貯存數字數據的方式。通常，一部電腦用固定數目的位來貯存數字，可以是 8 位、16 位或 32 位的。用的位越多，所能貯存的數字的範圍也越大。為了簡化原因，這裏僅用 8 位的字來講解。

符號及值表示法

在符號及值表示法中，數字的符號和它的值是分開表示的。使用這種表示法，在通常的二進制位值之外，還有最高有效位來表示符號位(0 代表正數而 1 代表負數)。請看下面：



例如，使用 8 位，

$$+ 67_{(10)} = 01000011$$

$$- 67_{(10)} = 11000011$$

二進制補碼表示法

在二進制補碼表示法中，我們使用通常的二進制位值，但最高有效位表示了負值。

-128	64	32	16	8	4	2	1	
0	0	1	0	0	1	0	0	$= 36_{(10)}$
1	1	0	1	1	1	0	0	$= -36_{(10)}$
0	1	1	1	1	1	1	1	$= 127_{(10)}$
1	0	0	0	0	0	0	0	$= -128_{(10)}$

用二進制補碼，就可很容易地把正數變成相應的負數，反之亦然。方法如下：

- 1 把所有的 0 變成 1 並把所有的 1 變成 0。
- 2 將所得代碼再加 1。

範例 3

	-128	64	32	16	8	4	2	1
$50_{(10)}$	0	0	1	1	0	0	1	0
換位	1	1	0	0	1	1	0	1
加 1								1
	1	1	0	0	1	1	1	0
$= -50_{(10)}$								

範例 4

	-128	64	32	16	8	4	2	1
$-83_{(10)}$	1	0	1	0	1	1	0	1
換位	0	1	0	1	0	0	1	0
加 1								1
	0	1	0	1	0	0	1	1
$= 83_{(10)}$								

利用二進制補碼進行二進制運算

下面的四個例子說明用 8 位字進行二進制的加法和減法。減法由加法與互補來進行。

(a)	50	00110010	(b)	50	00110010
	- 83	+ 10101101		+ 83	+ 01010011
	- 33	11011111 = -33		133	10000101 = -123
		(正確)			(不正確)
(c)	83	01010011	(d)	- 83	10101101
	- 50	+ 11001110		- 50	+ 11001110
	33	100100001 = 33		- 133	101111011 = 123
		(正確)			(不正確)
	捨去 1			捨去 1	

在(c)和(d)兩種情況中，最高有效位都有一個 1 的進位，該位應捨棄，因為採用的是 8 位二進制數。

在(b)和(d)兩種情況中的結果並不正確；這是溢出誤差的例子。情況(b)的答案 133 太大，不能用二進制補碼表示法貯存在 8 位的字中。另一方面，情況(d)的答案 -133 太小，未能貯存。兩個答案都在所能表示的數字範圍之外。這處的範圍為 -128 至 127。

在兩個正數的和變成負值時，便會出現溢出。同樣地，在兩個負數的和變成正值時，便會出現誤差。為了容納較大的值，我們需要更多的位。

符號數值表示法與二進制補碼表示法的比較

符號數值表示法與二進制補碼表示法的比較如表 10.1 所示。

符號數值表示法	二進制補碼表示法
零有兩種表示 00000000 (+0) 10000000 (-0)	零僅有一種表示 00000000
$-2^7 + 1 \leq \text{範圍} \leq 2^7 - 1$	$-2^7 \leq \text{範圍} \leq 2^7 - 1$
算術運算時需用複雜的電路。當兩個整數相加，電腦首先要檢查符號位。	減法可用加法進行，所以可以用一般的電路。

表 10.1 符號數值表示法與二進制補碼表示法的比較