



數字系統與數碼

1-1 簡介

數位系統 — 是指一個處理有限數字(或數值)資料集合的系統。

它也常稱為數位邏輯、交換電路、或邏輯電路。

◎數位系統 (Digital System) — 用以處理一些以數位型式表示的物理量之裝置組合。

數位系統主要應用 — 數位計算機的設計、數位儀器、
數位通信設備、數位控制電路及
打字機等。

常見的類比系統有 — 電話系統、磁帶錄放音設備及
汽車行程表。

最大的不同 — 類比 ≡ 連續的
數位 ≡ 斷續的

<1>在數位系統中，常用的數目系統 (Number System) 有：

二進制 (Binary)、八進制 (Octal)、十進制 (Decimal) 與 十六進制 (Hexadecimal) 等。


註：數位電路和系統的工作仍是採用二進制。

使用八進制、十六進制只是使系統操作者方便。

ie：用來有效地表示更大的二進制數字。

理由 (優點)： 1.更適合且更有效率

2.其間轉換容易



在電腦中，二進制數字可能表示：

(1)實際的數值資料

(2)數字對應於記憶體中的位址 (address)

(3)一個指令碼 (Instruction code)

(4)代表字母或其他非數值字元的代碼


(5)一組表達電腦內部或外部裝置之狀態 (states) 的位元

1-2 數字系統

數字系統 — 二進數、八進數、十進數
、十六進數

<2> 相互轉換與四則運算 (+、-、×、÷)

相互轉換：小基底 → 大基底
大基底 → 小基底



提要： 小 \rightarrow 大 大 \rightarrow 小

2 \rightarrow 8 16 \rightarrow 10

2 \rightarrow 10 16 \rightarrow 8

2 \rightarrow 16 16 \rightarrow 2

8 \rightarrow 10 10 \rightarrow 8

8 \rightarrow 16 10 \rightarrow 2

10 \rightarrow 16 8 \rightarrow 2

直接轉換：如 $2 \leftrightarrow 8$ 、 $2 \leftrightarrow 16$

(目視法：取3或4 bit 為一組)。

間接轉換：如 $8 \leftrightarrow 16$ ，先把題目化成二進數後，再依直接轉換要訣處理。

運算轉換：無法使用上述簡便方法，必須用乘、除法加以運算。

如 $2 \leftrightarrow 10$ 、 $8 \leftrightarrow 10$ 、 $10 \leftrightarrow 16$

例子：將 177_{10} 轉換為二進制等效值。

Sol1：

$$\begin{array}{r|l} 2 & 177 \\ \hline & 88 \\ 2 & 44 \\ \hline & 22 \\ 2 & 11 \\ \hline & 5 \\ 2 & 2 \\ \hline & 1 \end{array} \begin{array}{l} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{array} \uparrow$$

$$\begin{aligned} 177_{10} &= \underline{10} \underline{110} \underline{001}_2 \\ &= 261_8 \end{aligned}$$

Sol2:

$$\begin{array}{r|l} 8 & 177 \\ \hline & 22 \\ 8 & 2 \\ \hline & 6 \end{array} \begin{array}{l} 1 \\ 6 \end{array} \uparrow$$

$$\begin{aligned} 177_{10} &= 261_8 \\ &= \underline{010} \underline{110} \underline{001}_2 \end{aligned}$$

*採用 $10 \rightarrow 8$ ，再 $8 \rightarrow 2$ ，通常比直接由 $10 \rightarrow 2$ 快，特別是大數目時。

相同地，先將 $2 \rightarrow 8$ ，再 $8 \rightarrow 10$ 也是較快的方法。

1-3 數碼系統

亦為二進碼使用在數位系統中以代表數字性資料
(如 BCD Code)或非數字性資料(如 ASCII Code)



BCD碼：

如果每個十進制數字的位數以等效的二進制表示，
此過程產生的碼謂之。

例子：將 0110100000111001 (BCD) 轉換為等效的
十進制。

Sol : 0110 1000 0011 1001
 6 8 3 9

優點：易於與十進制作轉換或反轉換。

加三碼：

對任何二進位代表之BCD碼再加上3

即 $0011_{(2)}$ 謂之。

優點：為一自補碼 (self complementary codes)

亦即要求這些碼的9之補數時，只須將

“0”換成“1”，“1”換成“0”即可。

XS-3 碼與十進數對照表

十進數	XS-3碼
0	0011
1	0100
2	0101
3	0110
4	0111
5	1000
6	1001
7	1010
8	1011
9	1100



格雷碼 (Gray Code)：

主要特點是在數碼變換過程中，每次只有一個位數變化。

亦即屬於最小變化碼 (minimum – change code)。

因此格雷碼不適用於算術運算

∴ 其是一種非加權碼(unweighted code)

而適用於資料的傳輸。

例子：將 $1001_{(2)}$ 化成 Gray Code

Sol :

1	0	0	1
↓	↘	↘	↘
1	1	0	1

例子：將 Gray Code 1101 化成 二進數

Sol :

1	1	0	1
↓	↘	↘	↘
1	0	0	1

文字碼：

- 除了數字資料外，一個電腦必須能夠處理非數字的資料。
- 換句話說，一個電腦必須認識代表字母、標點符號和其他特殊字元的數碼，就如同其認識數字一樣。這些數碼謂之。
- 最常用的文字碼是美國標準資訊交換碼 (American Standard Code for Information Interchange，簡稱ASCII(讀音是 askee))
ASCII碼是用來傳送文字資訊在電腦和輸入/輸出裝置之間，如終端機或印表機。電腦也能儲存由操作員經鍵盤輸入之訊號。ASCII為目前使用最廣泛的通信碼。

漢明碼：

兼具有自動錯誤偵測與更正一個bit功能。

漢明碼的理論基礎：

(1)除了 m 個位元外，再加上 k 個檢查位元同時傳送

(2)由 k 的檢查位元可推得一個二進制數：

$D_{k-1} \cdots D_0$ 用以指出該組資訊中那一位元是錯誤的。我們稱此二進制位數為錯誤指標(Error Index)

(3) 在 $(m + k)$ 個資訊位元中的任一單一錯誤必須要能由錯誤指標指出。而且，錯誤指標必須保留當其本身均為 0 時，代表沒有任何錯誤的情況發生。

因此我們可以得 m 與 k 之間的關係為

$$\times \quad 2^k \geq m + k + 1$$



編出漢明碼六大步驟：（資料位元=4為例子）

- 1.決定那些位置上是檢查位元 k ，那些是資料位元 m
- 2.按順序將資料位元放入
- 3.取1、3、5及7位置上的位元組成第一組同位檢查
- 4.取2、3、6及7位置上的位元組成第二組同位檢查
- 5.取4、5、6及7位置上的位元組成第三組同位檢查
- 6.編成完整的漢明碼

例子：將二進制信號 0100 組成漢明碼

Sol：

$$\text{由 } 2^k \geq m+k+1$$

$$2^k \geq 4+k+1 \Rightarrow k=3$$

步驟：

1. $k=3$ ，分別置於 2^0 ， 2^1 ， 2^2

$$k+m=3+4=7$$

2. 位置 $X_1^{\downarrow 2^0}$ $X_2^{\downarrow 2^1}$ X_3 $X_4^{\downarrow 2^2}$ X_5 X_6 X_7

k_0	k_1	m_3	k_2	m_2	m_1	m_0
		0		1	0	0

$$\begin{aligned}
 3. C_1 &= X_1 \oplus X_3 \oplus X_5 \oplus X_7 \\
 &= k_0 \oplus 0 \oplus 1 \oplus 0 = 0 \\
 &\Rightarrow k_0 = 1
 \end{aligned}$$

$$\begin{aligned}
 4. C_2 &= X_2 \oplus X_3 \oplus X_6 \oplus X_7 \\
 &= k_1 \oplus 0 \oplus 0 \oplus 0 = 0 \\
 &\Rightarrow k_1 = 0
 \end{aligned}$$

$$\begin{aligned}
 5. C_3 &= X_4 \oplus X_5 \oplus X_6 \oplus X_7 \\
 &= k_2 \oplus 1 \oplus 0 \oplus 0 = 0 \\
 &\Rightarrow k_2 = 1
 \end{aligned}$$

$$\begin{array}{ccccccc}
 6. & k_0 & k_1 & m_3 & k_2 & m_2 & m_1 & m_0 \\
 & 1 & 0 & 0 & 1 & 1 & 0 & 0
 \end{array}$$

步驟 3、4、5 由下得知

	k_2	k_1	k_0
X_0	0	0	0
X_1	0	0	1
X_2	0	1	0
X_3	0	1	1
X_4	1	0	0
X_5	1	0	1
X_6	1	1	0
X_7	1	1	1

例子：試偵錯 Hamming Code 為 1101101 之第幾位元錯誤，其正確值為多少？


Sol：

X_1	X_2	X_3	X_4	X_5	X_6	X_7
2^0	2^1		2^2			
k_0	k_1	m_3	k_2	m_2	m_1	m_0
1	1	0	1	1	0	1

$$\begin{aligned} C_1 &= X_1 \oplus X_3 \oplus X_5 \oplus X_7 \\ &= 1 \oplus 0 \oplus 1 \oplus 1 = 1 = D_0 \end{aligned}$$

$$\begin{aligned}
 C_2 &= X_2 \oplus X_3 \oplus X_6 \oplus X_7 \\
 &= 1 \oplus 0 \oplus 0 \oplus 1 = 0 = D_1 \\
 C_3 &= X_4 \oplus X_5 \oplus X_6 \oplus X_7 \\
 &= 1 \oplus 1 \oplus 0 \oplus 1 = 1 = D_2
 \end{aligned}$$

求得錯誤指標為 101，亦即第五位置上發生錯誤，
 所以正確的漢明碼為 1 1 0 1 0 0 1



相關專有名詞：

加權碼(Weighted)：數碼的一種，其碼群中的位置均有一個固定的加權(即比重)

如：8-4-2-1 碼。

非加權碼(Unweighted Code)：數碼的一種，其碼群中的位元位置，並沒有特殊的權重。

如：XS-3 碼、Gray Code 及 ASCII Code

最低位元(Least Significant Bit ; LSB)：以二進制表示數量，在其最右邊(權重最小)的位元。

EX：

1011 LSB

最低位數(Least Significant Digit ; LSD)：在一特定的數字中，權位最小的位數。

EX：

765 LSD

pb：寫出下列16進位的加與乘的結果

$$<1> 5A+16C \quad <2> 28 \times B5$$

〈要訣：先化成二進位求解，再換回答案！〉

$$\text{Ans} : <1> 106_{(16)} \quad <2> 1048_{(16)}$$

pb：其資料存於其記憶體位址由 $(B08D)_{16}$ 到 $(BCBB)_{16}$ ，試問其容量為幾KB？

$$\text{Ans} : 3.046 \text{ KB}$$

Hamming Code 另解：

編碼：位置	1	2	3	4	5	6	7
	k_0	k_1	m_3	k_2	m_2	m_1	m_0

將 m 位元中為 1 處，化成等效位置之二進數，再將之互斥或求得 $k_2 k_1 k_0$

如前例 $m_2=1$ 為 position 5 $\Rightarrow 101$

偵錯：將 Hamming Code 中 position 為 1 處化為 position 等效二進數，再全數互斥或求得 error index