Unit 1：數字系統與數碼系統

Section 1: 數字系統

- 十進位(Decimal) 0,1,2,3,4,5,6,7,8,9
- 二進位(Binary) 0,1
- 八進位(Octal) 0,1,2,3,4,5,6,7
- 十六進位(Hexadecimal) 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

數制間的轉換:

整數部分: 用短除法連除 2(或 8 或 16)，由下而上寫出餘數

小數部分: 用乘法乘 2(或 8 或 16)，直到小數為 0。由上而下寫出乘積的整數。

範例 1: 將十進位數字 0 到 16 分別以十進位（Decimal），二進位（Binary），八進位（Octal），及十六進位（Hexadecimal）表示出來。

解答：

1. 十進位達 10 即進位，而二進位達 2 即進，八進位達 8 即進位，十六進位達 16 即進位。
2. 十六進位，由於基底爲 16，故其數元分別 0,1,2,……,14,15，但 10 到 15 由於有兩個數元，故借來 A,B,C,D,E,F 等數元以分別表示 10 到 15。
3. 若是二進位，則其一個數元（digit）被稱爲二進數元或位元（binary digit, or bit）。

十進制（Decimal）	二進制（Binary）	八進制（Octal）	十六進制（Hexadecimal）
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

範例 2：試將下列數字表示成十進位數字：
① $(903.87)_{10}$ ② $(10101.01)_2$ ③ $(754.34)_8$ ④ $(ABC.5)_{16}$

解答：

① $(903.87)_{10} = 9\times 10^2 + 0\times 10^1 + 3\times 10^0 + 8\times 10^{-1} + 7\times 10^{-2}$
$= 900 + 3 + 0.8 + 0.07$
$= 903.87$

② $(10101.01)_2 = 1\times 2^4 + 0\times 2^3 + 1\times 2^2 + 0\times 2^1 + 1\times 2^0 + 0\times 2^{-1} + 1\times 2^{-2}$
$= 16 + 4 + 1 + 0.25$
$= 21.25$

③ $(754.34)_8 = 7\times 8^2 + 5\times 8^1 + 4\times 8^0 + 3\times 8^{-1} + 4\times 8^{-2}$
$= 448 + 40 + 4 + 0.375 + 0.0625$
$= 492.4375$

④ $(ABC.5)_{16} = 10\times 16^2 + 11\times 16^1 + 12\times 16^0 + 5\times 16^{-1}$
$= 2560 + 176 + 12 + 0.3125$
$= 2748.3125$

範例 3： 試將 $(18.75)_{10}$ 轉成二進位數字，或以二進位來表示之。

解答：

$(18.75)_{10} = (18)_{10} + (0.75)_{10}$

我們首先將整數部分與小數部分分開，並分別找出它們的二進位表示法，再合併之。

（一）整數部分：用除 2 的方式	（二）小數部分：用乘 2 的方式
2 | 18 --- 0 (18÷2 的餘數) 1 | 9 --- 1 (9÷2 的餘數) 2 | 4 --- 0 (4÷2 的餘數) 2 | 2 --- 0 (2÷2 的餘數) 2 | 1 --- 1 ← MSD 0 所以 $(18)_{10} = (10010)_2$	0.75 × 2 小數點後第一位 $\underline{1}.50$ 0.50 × 2 小數點後第二位 $\underline{1}.00$ 當小數部分＝0 時則停止。 所以 $(0.75)_{10} = (0.11)_2$

將整數及小數合併，$(18.75)_{10} = (10010.11)_2$

讀者可自行練習將 $(13.3)_{10}$ 以二進位來表示。下面範例說明如何將十進位數字轉成八進位數字。

範例 5：試將 $(43969.75)_{10}$ 以十六進位表示之。

解答：

$(43969.75)_{10} = (43969)_{10} + (0.75)_{10}$

（一）整數部分：用除 16 的方式

```
16 | 43969   --- 1 (43969÷16 的餘數)
16 |  2748   --- 12 (2748÷16 的餘數)
16 |   171   --- 11 (2171÷16 的餘數)
16 |    10   ---10 ← MSD
         0
```

故 $(43969)_{10} = (ABC1)_{16}$

（二）小數部分：用乘 16 的方式

```
        0.75
  ×       16
```

小數點後第一位 12.000

↑因=0，故停止

$(0.75)_{10} = (0.C)_{16}$

將整數與小數合併 $(43969.75)_{10} = (ABC1.C)_{16}$

讀者可自行練習將 $(1000.625)_{10}$ 轉換成十六進位。

範例 6：① 將二進位數字 $(10101.0101)_2$ 以八進位表示之。
② 將八進位數字 $(123.45)_8$ 轉成二進位數字。

解答：

① 將二進位轉成八進位，其做法是以小數點爲分界，分別往左右每三個位元爲一組，若不夠三位元則補 0，之後每一組的三個位元轉成八進位的數元。

```
  0 1 0 1 0 1 . 0 1 0 1 0 0
  |___| |___|   |___| |___|
⇨   2     5   .   2     4
```

故 $(10101.0101)_2 = (25.24)_8$

② 將八進位轉成二進位，其做法是將每個八位數元以三位元的二進位數元表示之即可。

$$(123.45)_8 = (\underbrace{001}_{1}\ \underbrace{010}_{2}\ \underbrace{011}_{3}\ .\ \underbrace{100}_{4}\ \underbrace{101}_{5})_2$$

$$= (1010011.100101)_2$$

範例 7：① 試將 $(11101.0101)_2$ 以十六進位表示之。 ② 試將 $(2A5.A8)_{16}$ 轉成二進位數字。

解答：

① 與由二進位轉成八進位的方法類似，不過二進位轉十六進位，要四個位元組成一組，不足四個則補 0。

$$\underbrace{0001}_{1}\ \underbrace{1101}_{D}\ .\ \underbrace{0101}_{5}$$

故 $(11101.0101)_2 = (1D.5)_{16}$

② 將十六進位的每個位元轉成以四位元的二進位即可。

$$(2A5.A8)_{16} = (\underbrace{0010}_{2}\ \underbrace{1010}_{A}\ \underbrace{0101}_{5}.\underbrace{1010}_{A}\ \underbrace{1000}_{8})_2$$
$$= (1010100101.10101)_2$$

最左邊與最右邊的 "0" 皆可略去。

範例 8：① 試將 $(365.5)_8$ 轉成十六進位數字。 ② 試將 $(293.AC)_{16}$ 轉成八進位數字。

解答:

① 將八進位數字先轉成二進位數字，再由二進位轉成十六進位即可。

$$(365.5)_8 = (\underline{011}\ \underline{110}\ \underline{101}\ .\ \underline{101})_2$$
$$= (0\ \underbrace{1111}_{F}\ \underbrace{0101}_{5}\ .\ \underbrace{1010}_{A})_2$$
$$= (F5.A)_{16}$$

② 先將十六進位轉成二進位，之後再轉成八進位即可。

$$(293.AC)_{16} = (\underline{0010}\ \underline{1001}\ \underline{0011}\ .\ \underline{1010}\ \underline{1100})_2$$
$$= (\underbrace{001}_{1}\ \underbrace{010}_{2}\ \underbrace{010}_{2}\ \underbrace{011}_{3}\ .\ \underbrace{101}_{5}\ \underbrace{011}_{3})_2$$
$$= (1223.53)_8$$

練習:

1.計算 0.828125_{10}=___________$_{16}$

2.計算 BCD_{16}=_______$_{2}$=__________$_{10}$

3.計算 1023_{10}=___________$_{16}$

4.計算 10010.101111_2=_______$_{8}$=_______$_{16}$

5.計算 571.2438=_________$_{10}$

6.某資料存於某記憶體位址由 $B08D_{16}$ 到 $BCBB_{16}$，問其容量為幾 KB? _______

¤ 補數的運算

二進位的補數分下列兩種

- 1's 補數:即為原數的相反(0 變 1，1 變 0)
- 2's 補數(2 進位的負數表示法)= 該數的 1's 補數+1

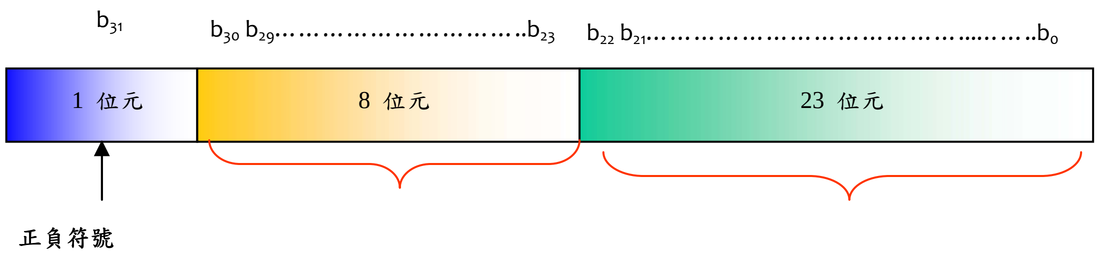
2's 補數表示法:若為 n bits，可表示的範圍為$-(2)^n \sim +(2)^n-1$

例:請以 8bits 表示下列資料

	1's 補數	2's 補數
(1) 00110010		
(2) 10101100		
(3) 11010011		

¤ 浮點數表示法

- 表示實數資料
  - ❑ 單倍精確浮點數：32 位元。
  - ❑ 雙倍精確浮點數：64 位元。
  - ❑ 延伸精確浮點數：80 位元。
- 表示法



- 說明
  - ❑ 正/負符號
    - ➢ b31=0 表示此實數為正數；b31=1 表示此實數為負數。
  - ❑ 偏差指數
    - ➢ 8 位元表示的非負整數值為 0~255。
    - ➢ 實數可由很小至很大，故需要正、負二種指數，因此以 127 為指數偏差值，實際的指數值=偏差指數-127。
      - ◆ 偏差指數的範圍為 127～255，則代表真正指數值為 0～128。
      - ◆ 偏差指數介於 126～0 之間，則代表真正指數值介於-1～-127 之間。
      - ◆ 偏差指數是 132，其真正指數則為 5；偏差指數是 120，其真正指數即是-7。
  - ❑ 小數部分
    - ➢ 此處的小數部分是經過正規化(normalization)後的小數。由於它有 23 位元，所以可準確到小數點後 23 位。

範例 19：-0.111×2^{-4}之浮點表示法為何?

解答:

因此數是負值，故 $b_{31} = 1$。並將-0.111×2^{-4}正規化成-1.11×2^{-5}，故其指數為-5。但因指數偏差值為 127，故偏差指數是$-5 + 127 = 122$，表示成八位元的二進數字為 01111010。有效小數部分則為 $11\underline{0 \cdots 0}$(21 個 0)。所以此數被儲存為 1 01111010 11000000000000000000000。

範例 20： ±0 之浮點表示法為何?

解答:

因 0 無法寫成正規化形式，故需以特殊情形待之。最簡單的方法即是令三部分皆為 0。故±0 是以 0 0000000 00000000000000000000000。

Section 2: 數碼系統

⌖ 種類

8-4-2-1 碼、二五碼、環形計數碼、五取二碼、超三碼、Gray Code(格雷碼)、BCD 碼、標準 BCD 碼、ASCII 碼、EBCDIC 碼、其它。

⌖ BCD 碼:每個十進位數可用 4bit 一組的二進數來表示。

十進數	BCD 碼
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

練習:

1. 轉換 BCD 碼為十進位數:

100101_{BCD}=______________$_{10}$

11001_{BCD}=______________$_{10}$

011000011001_{BCD}=__________$_{10}$

2. 轉換十進位為 BCD 碼:

104_{10}=__________________$_{BCD}$

547_{10}=__________________$_{BCD}$

359_{10}=__________________$_{BCD}$

⌗ ASCII:美國資訊標準交換碼

一般使用 7 個 bit 來表示一個字元(character)。是目前使用最廣泛的通信碼。

⌗ Gray Code:通常用於資料的傳輸，不適合用於算術運算。(每一次只有一個位元異動)。

- ❖ 任何連續的兩個二進位表示法，只有一個位元不相同；其餘相同。
- ❖ 用二個位元來表示整數 0，1，2，3，

  方法一：

  即 G1=｛0=00，1=01，2=11，3=10｝

  方法二：

  即 G2=｛00=10，1=11，2=01，3=00｝。
- ❖ 學者研究出一種二進碼，稱為反射葛雷碼（Reflected Gray code），其編碼方式唯一而且有系統，故廣泛應用在計算機領域。

十進位	**Gray Code**	二進位
0	0000	0000
1	0001	0001
2	0011	0010
3	0010	0011
4	0110	0100
5	0111	0101
6	0101	0110
7	0100	0111
8	1100	1000
9	1101	1001
10	1111	1010
11	1110	1011
12	1010	1100
13	1011	1101
14	1001	1110
15	1000	1111

二進位化為 Gray Code:

	B_8	B_4	B_2	B_1
二進數	1	0	0	1
		⊕	⊕	⊕
Gray Code	1	1	0	1

⊕：代表 XOR (即二者相異才為 1)

範例17：試將15以4位元反射葛雷碼表示之。

解答：

1. 先將15轉成4位元的二進位數字 $= (1111)_2$
2. 將二進位數字轉成反射葛雷碼：

   將反射葛雷碼的位元由左到右依次產生。

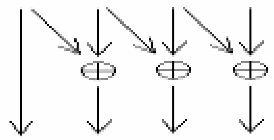
   最左邊的位元 = 二進位數字最左邊的位元。

   左邊第二個位元 = 二進位數字左邊第二個位元 ⊕ 最左邊的位元

   左邊第三個位元 = 二進位數字左邊第三個位元 ⊕ 左邊第二個位元

   最右邊的位元 = 二進位數字左邊第四個位元 ⊕ 左邊第三個位元

二進位數字： 1 1 1 1



反射葛雷碼： 1 0 0 0

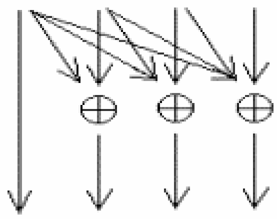
故得15的反射葛雷碼爲 1000

範例18：試問4位元反射葛雷碼1100，其相對應的十進位數字爲何？

解答：

1. 由 N_{RG} 先轉成 N_2，其方法如下：

N_{RG} = 1 1 0 0



N_2 = 1 0 0 0

2. 將二進位數字轉成十進位數字

$N_2 = (1000)_2 = 8$

故反射葛雷碼1100代表十進位數字8。

練習:

1. 十進位 5 的 Gray Code 為________
2. 二進位 101101 的 Gray Code 為__________

¤ 霍夫曼碼(Huffman Code)

- 不固定長度的編碼方式，符號編碼長度與出現頻率成反比。
- 編碼步驟
  - ❑ 找出所有符號出現頻率。
  - ❑ 將頻率最低的兩者相加得出另一個頻率。

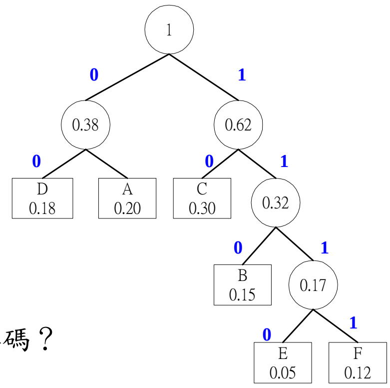
- ❑ 重覆以上第二步驟，將最低兩個頻率相加，直到只剩下一個頻率為止。
- ❑ 根據合併關係分配 0 與 1，而形成一棵編碼樹。

⌖ 實例—編碼

- ❑ 假設編碼系統有 A, B,C, D, E, F 等六個符號，期出現頻率依序為 0.2, 0.15, 0.3, 0.18, 0.05, 0.12，試設計霍夫曼碼？
- ❑ 編碼結果

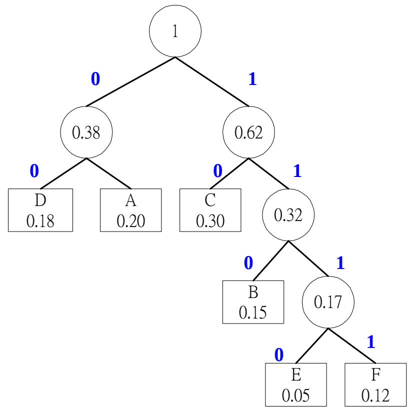
  A: 01; B:110; C:10

  D:00; E:1110;F:1111
- ❑ 總共所需位元：17 bits。



⌖ 實例—解碼

- ❑ 請依照上一題所設計霍夫曼碼，將 111110010000110 進行解碼？
- ❑ 解碼結果
  - ❖ **FCADDB**



¤ 數碼檢查與更正

⌖ 方法:

- ❑ 同位元檢查(Parity check)—又分為奇數同位元檢查和偶數同位元檢查
- ❑ 定數檢查(fixed-count check)
- ❑ Cyclic Redundancy Check(CRC)
- ❑ 漢明碼(Hamming Code)

⌖ 同位元檢查

主要原理是於資料碼後自動加一個 parity bit (同位元)，使此資料碼內"1"的個數形成奇數或偶數。

若採奇同位元數碼檢查法，則 parity bit 和資料碼內 1 的個數總和應為奇數。

反之，則為偶同位元數碼檢查法。

例 1:使用奇同位:

資料碼	同位位元
1101101	0
1011001	1

例 2:使用偶同位:

資料碼	同位位元
1101101	1
1011001	0

【練習】:

1. 採用奇同位元檢查法傳送 7 位元資料，以下為接收端收到的各筆資料，何者可確知在傳送中有錯誤發生？ (A)11100000 (B)10110000 (C)10001111 (D)10101010。

2. 經加上偶同位元檢查後，下列代碼何者正確？ (A)1010111 (B)0110111 (C)1111001 (D)1011001。

Hamming Code

漢明碼為一個兼具自動錯誤偵測與更正一個 bit 的雙種功能，若用 7 個 bit，4 個代表 data，3 個代表檢查碼。

bit position	7	6	5	4	3	2	1	其中
Data	M_4	M_3	M_2		M_1			$c_1=M_4\oplus M_2\oplus M_1$ $c_2=M_4\oplus M_3\oplus M_1$ $c_3=M_4\oplus M_3\oplus M_2$
Check bit				C_3		C_2	C_1	

Hamming Distance:

代表兩組信號不同 bit 個數之和。

如:A=$(10101)_2$　B=$(11110)_2$，　因為有 3 個位元不同，所以 Hamming Distance=3.