CH 5 數碼、文數字碼、錯誤偵測與更正碼

A. BCD 碼 – 4bit 表示 0~9，1010 之後不使用(6 個)

B. BCD 碼變形(Excess – 3 code) – BCD+ 3 : 0010~1100(0000~0010 & 1101~1111 : 6 個不用)

C. 8、4、-2、-1 碼 – 4bit

D. Gray Code 格雷碼 – 最右邊補 0，做 XOR

E. ASCII - 1 parity check bit + 7 bits 表英文數字 = 8 bits；數字(48) < 大寫字母(65) < 小寫字母(97)

F. Extended BCDIC(EBCDIC) – 8 bits；小寫 < 大寫 < 數字

G. Unicode – 2 bytes (16bits)；前 128 與 ACSII 相同

H. Big5 code

I. Parity Check – 只偵測錯誤；even parity : 偶數個 1；odd parity : 奇數個 1；將各 bit XOR 即可

J. CRC – 只偵測錯誤；$M(x)$ % $G(x)$ = check sum； 送→ $M(x)$ – (XOR) checksum；收→除$G(x)$ = 0(無問題)，反之

K. Hamming Code – 1 個 bit 出錯可知道，2 個無法；出現在 2 的次方位置

L. Hamming Distance – min{ 一組漢明距離}；n 個 error，H.D 需 ≥ n + 1(偵測)；n 個 error，H.D 需 ≥ 2n + 1(更正)

---

1. 數碼

- **BCD 碼**

(一) Def: 以4個bits表示十進制中之0~9位數值.

(二) 例: ① $(395)_{10} \to (0011\ 1001\ 0101)_{BCD}$

② $(671)_{10} \to (0110\ 0111\ 0001)_{BCD}$

③ $(1000\ 0110\ 0011)_{BCD} \to (863)_{10}$

④ $(0111\ 0110)_{BCD} \to (76)_{10}$

(三) 特性: 有6種組合不用 $(10)_{10} \sim (15)_{10}$ ⇒ 1010, 1011, 1100, 1101, 1110, 1111

例: 下列哪個非BCD碼?

(A) 10011001 (B) 00111001 (C) 01111001 (D) 10001100

Ans: (D)

Decimal	BCD Code	Decimal	BCD Code
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0101	8	1000
4	0100	9	1001

- 加三碼**(Excess – 3)code**

(一) Def: 以4個bits表示十進制中之0~9位數值.

(二) 例: ① $(395)_{10} \to (0011\ 1001\ 0101)_{BCD}$

② $(671)_{10} \to (0110\ 0111\ 0001)_{BCD}$

③ $(1000\ 0110\ 0011)_{BCD} \to (863)_{10}$

④ $(0111\ 0110)_{BCD} \to (76)_{10}$

(三) 特性: 有6種組合不用 $(10)_{10} \sim (15)_{10}$ ⇒ 1010, 1011, 1100, 1101, 1110, 1111

例: 下列哪個非BCD碼?

(A) 10011001 (B) 00111001 (C) 01111001 (D) 10001100

Ans: (D)

- **8、4、-2、-1** 碼

(一) Def：也是BCD碼的變形之一. 用4 bits表0~9位數值.

(二)

十進制	8	4	-2	-1
0	0	0	0	0
1	0	1	1	1
2	0	1	1	0
3	0	1	0	1
4	0	1	0	0
5	1	0	1	1
6	1	0	1	0
7	1	0	0	1
8	1	0	0	0
9	1	1	1	1

$4*1+(-2)*1+(-1)*1=1$

(三) $(157)_{10} \rightarrow (011101111001)_{8.4.-2.-1}$

$(111101011010)_{8.4.-2.-1} \rightarrow (936)_{10}$

- **Gray code 格雷碼**

(一) 用於通訊傳輸. 通常非用於數值計算方面。

(二) 特點：① 連續的數值
② 互為補數
} 只差1個Bit值不同

(三) 十進制 → Gray Code

Steps：① 十進制 → 二進制 $(b_n b_{n-1} \dots b_1 b_0)_2$

② 最左邊補1個"0"

Note：$\oplus$ 互斥或 (exclusive-or)
相同為0. 相異為1

$0 \; b_n \; b_{n-1} \cdots b_1 \; b_0$

$\oplus \; \oplus \; \cdots \; \oplus$

$(G_n \; G_{n-1} \cdots G_0)_{Gray}$

例：十進制 → Gray Code

(1) $(120)_{10}$

0 1 1 1 1 0 0 0

$(1000100)_{Gray}$

(2) 4 bits

① $(3)_{10} \rightarrow (0010)_{Gray}$

Ans: ∅ 0 0 1 1

$(0010)_{Gray}$

② $(4)_{10} \rightarrow (0110)_{Gray}$

Ans: ∅ 0 1 0 0

$(0110)_{Gray}$

連續數字只差1 bits不同

15's 互為15補

③ $(12)_{10} \rightarrow (1010)_{Gray}$

④ $(11)_{10} \rightarrow (1110)_{Gray}$

∅ 1 1 0 0

$(1010)_{Gray}$

∅ 1 0 1 1

$(1110)_{Gray}$

連續數字只差1 bit不同.

2. 文數字碼

- **ASCII Code**

⇨ 1 parity check bit + 7 bits 表英文數字 = 8 bits = 1 Byte

(一) 全名: American Standard Code for Information Interchange

(二) 理論上用7個bits表示英文字母(大.小寫)及特殊符號(e.g. !, <, >, =, ...)

及數字(∅~9) 但實際上電腦內部儲存是多加一個同位位元

Parity Check Bit ∴ 是 8 bits = 1 byte

基準值(小 → 大) :

(1) 0 → 48 (10 進位)

(2) A → 65 (10 進位)

(3) a → 97 (10 進位)

⇨ 數字(48) < 大寫字母(65) < 小寫字母(97)

- **EBCDIC code**

(一) 為 IBM 大型主机採用

(二) 8 bits 代表一个文數字

(三) 編碼序：小寫 < 大寫 < 數字.

- **Unicode(Universal Code)萬國碼**

(一) 解決多國語系轉換

(二) 用 2 bytes (16 bits) 來表示每个文數字.

其中前128个字元. 跟 ASCII 字元一樣.

- 中文內碼**(Big – 5 code)**

(一) 使用 2 bytes 表達一个中文字

Note: 外碼 → 輸入法 e.g. 注音. 倉頡.

儲存在電腦內部 = 內碼

3. 錯誤碼偵測

- **Parity check(bit 是 1 個數)**

(1) **偶同位元(Even parity)**：如果一組給定資料位中 1 的個數是奇數，補一個 bit 為 1，使得總的 1 的個數是偶數

(2) **奇同位元(Odd parity)**：如果給定一組資料位中 1 的個數是奇數，補一個 bit 為 0，使得總的 1 的個數是奇數

(一) Def：分2種 { ① 偶同位 Even Parity
② 奇同位 odd Parity

(1) Even Parity：

Bit值為"1"的個數必須是偶數才算正確，否則有誤。

(2) odd Parity：

Bit值為"1"的個數必須是奇數才算正確.

(二) 例1：求下列 Even Parity check Bit值為？

(1) 0 1 1 0 0 1 1 $\boxed{?}$ Ans：ø (2) 1 1 1 1 1 0 0 $\boxed{?}$ P. Bit Ans：1

例2：下列何者非 Even Parity？

(A) 11111111 (B) 10101010 (C) 01101110 (D) 10101111 Ans：(C)

① 只能偵測錯誤，但無法更正錯誤.

② 製作簡單

③ 應用於 e.g. ASCII碼. RAM內容check, RAID, etc.

☆④ 若有偶數個 Bits 同時出錯, 則會被誤判為正確. 無法偵測出錯誤.

⑤ ⊕ 互斥或是偶同位位元之產生器

⇒ 只要將訊息之各個Bit拿來作 ⊕ 即可得出 Even Parity check Bit值

例： 1 1 0 1 0 0 1 0 之 Even Parity Bit 值 = ? 0

Ans： 1 ⊕ 1 0 1 0 0 1 0

0 0 1 1 1 0 0 #

- **CRC 循環冗餘校驗(Cyclic redundancy check)**

⇨ 送出減完 chekcsum 的 m(x)，再用 g(x)除

(一) 只能偵測錯誤，但無法更正錯誤。

(二) 應用原理：餘式定理

- 假設 Message 為 M(x)，雙方(收、送)共同協議出一個生成函數 G(x)
- 將 $G(x) \overline{) M(x)}$

  餘式 = check sum
- 送方送出 M(x) − 餘式 給收方
- 收方收到後，除以 G(x)，若餘式為 ϕ，則正確，否則有誤。

(三) 基本運作 in CRC

① 位元串列 ⟺ 多項式轉換

e.g. 101101 (5 4 3 2 1 0 方) ⇒ $x^5+x^3+x^2+1$　e.g. x^4+x^3+x+1 ⇒ 11011

② 減法(運算不借位，相當於是⊕互斥或)　e.g.

$$\begin{array}{r} 0100 \\ \oplus - 1011 \\ \hline 1111 \end{array}$$

③ 除法時，決定商數的 Bit 值，不是比大小決定

而是以如何消去最高位元(使用⊕)來決定的。

4. 錯誤碼更正

- 漢明碼**(Hamming Code)**

(一) 具更正錯誤之能力，若只有1個 Bit 出錯，則可以指出哪個位置有錯。

故可以予以更正。

(二) Hamming Code Bit (H.C. Bit) 一定要出現在 2 的冪次方的位置 (即 1.2.4.8.16)

例：訊息 = 1100010

則位置

1	2	3	4	5	6	7	8	9	10	11
□	□	1	□	1	0	0	□	0	1	0

→ Default 順序

1	2	3	4	5	6	7	8	9	10	11
□	□	0	□	1	0	0	□	0	1	1

←

(三) 送方如何決定 H.C. Bit 之值？（順序採→）

Bit值為1之位置	$\boxed{8}$	$\boxed{4}$	$\boxed{2}$	$\boxed{1}$
3	0	0	1	1
5	0	1	0	1
10	1	0	1	0
	$\boxed{1}$	$\boxed{1}$	$\boxed{0}$	$\boxed{0}$

↓ 作偶同位.

∴

1	2	3	4	5	6	7	8	9	10	11
$\boxed{0}$	$\boxed{0}$	1	$\boxed{1}$	1	0	0	$\boxed{1}$	0	1	0

$\underbrace{\qquad\qquad\qquad\qquad}_{\text{漢明傳輸碼.}}$

(四) 收方如何驗證對錯？

Bit值為1之位置	H.C. Bit 位置

↓ 作偶同位.

0 0 0 0 ⇒ 零值=正確

1 0 0 1 ⇒ 非零=有誤.

例：

1	2	3 ✓	4 ✓	5 ✓	6 ✓	7	8 ✓	9	10 ✓	11
$\boxed{0}$	$\boxed{0}$	1	$\boxed{1}$	1	1	0	$\boxed{1}$	0	1	0

	8	4	2	1
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
8	1	0	0	0
10	1	0	1	0
	0	1	1	0

↓ 偶同位.

⇒ 非零,為6

第6 bit值有誤

- 漢明距離 **Hamming Distance**

Def：兩個數碼之 H.D. = 兩個數碼之不同 Bit 值之個數.

例：① 求 010000 / 101010 之 H.D.=? 4※

② 10111001 / 00111000 之 H.D=? 2※

一組數碼之H.D.

Def：min.{兩兩數碼之H.D.}取最小值.

例：一組數碼如下：

$$\begin{cases} A = 1111111 \\ B = 0101001 \\ C = 1010110 \\ D = 0011110 \end{cases}$$

求此組數碼之H.D.?

Ans：

$C^4_2=6$

A與B之H.D. = 4
A與C = 3
A與D = 3
B與C = 7
B與D = 4
C與D = 2

取最小值 = 2 ※

記

1. 若要偵測 n 個 errors，則一組數碼之H.D.須 ≥ n+1
2. 若要更正 n 個 errors 〃 ≥ 2n+1

例1：① 若要偵測2个errors 則一組數碼之H.D.至少要 ≥ 3 (2+1=3)

② 更正2个 ≥ 5 (2*2+1)

③ 若一組數碼之H.D. = 7. 則最多可偵測 6 個 errors.

④ 更正 3 個 errors.

例2：一組數碼如下：

$$\begin{cases} A = 1111111 \\ B = 0101001 \\ C = 1010110 \end{cases}$$

(1) 求此組數碼之H.D. 3 ※

(2) 最多可更正?个errors $3=2n+1$ ∴ $n=1$ ※

(3) 若收方收到 0101011 則此code代表?

∵與B只差1个Bit不同. ∴可修正成 B ※