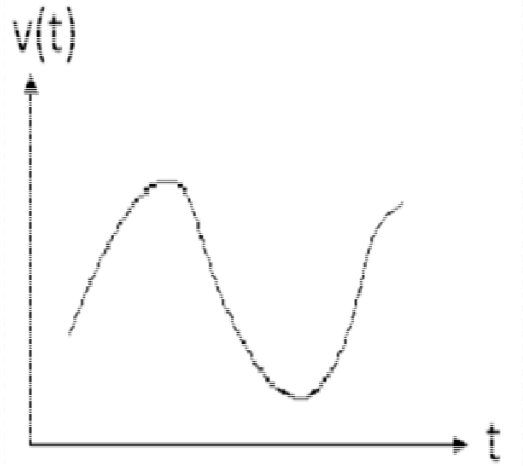
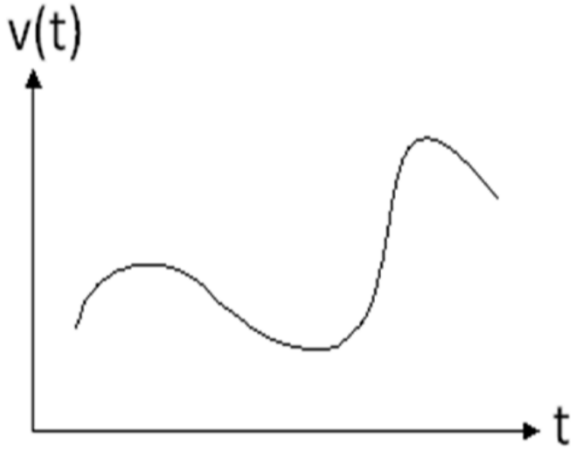
第1章

簡介：數字系統與轉換

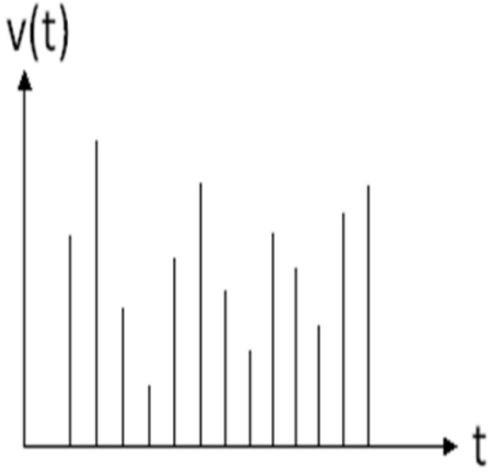
Digital Systems and Switching Circuits



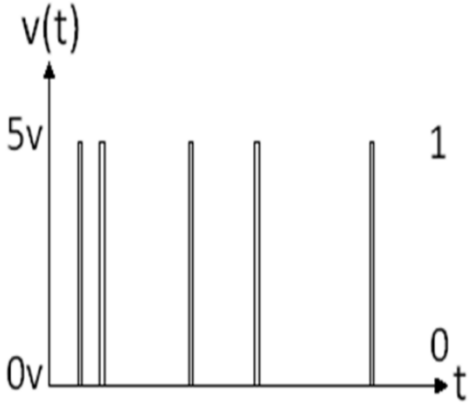
- Continuous signal



Analog signal



- Discrete signal



Digital signal

1.1 數位系統和交換電路

- 數位系統的設計大略可以分成三個部分：系統設計、邏輯設計和電路設計。
  - **系統設計**（system design）包括將整個系統分割成子系統，並且設定每個子系統的特性。
  - **邏輯設計**（logic design）包括決定如何連結內部基本的邏輯架構方塊以執行特定的功能，邏輯設計的例子像決定邏輯閘和正反器的內部連結以執行二進位加法。
  - **電路設計**（circuit design）包括指定特定的元件，如電阻、二極體和電晶體之內部連結，以構成邏輯閘、正反器或其他邏輯架構方塊。

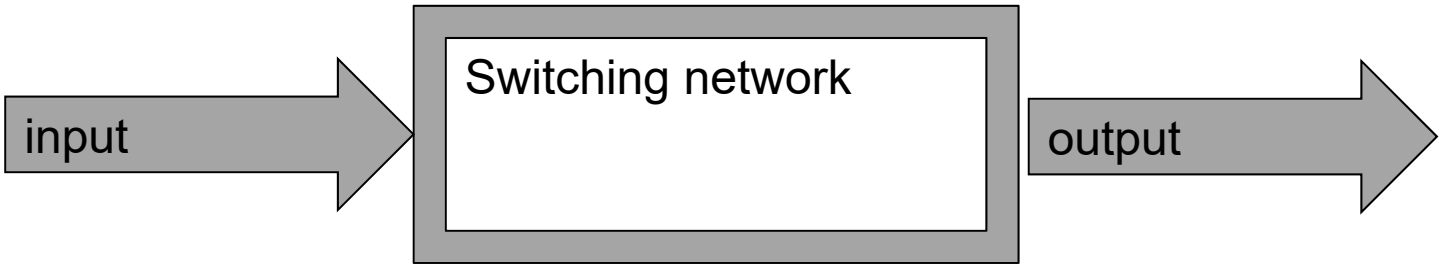
1.1 數位系統和交換電路

- 許多數位系統的子系統採用如圖1-1 **交換電路**（switching circuit）的形式，交換電路具有一個或多個離散的輸入值和輸出值。

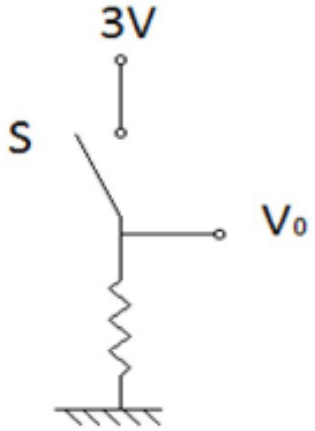
圖 1-1 交換電路

- 在本書中，將學到兩種類型的交換電路：**組合式**（combinational）與**序向式**（sequential）。

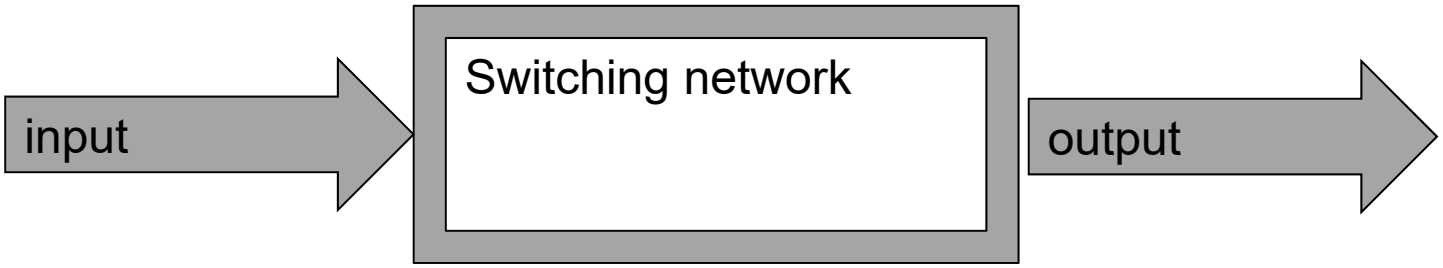
Digital Systems and Switching Circuits



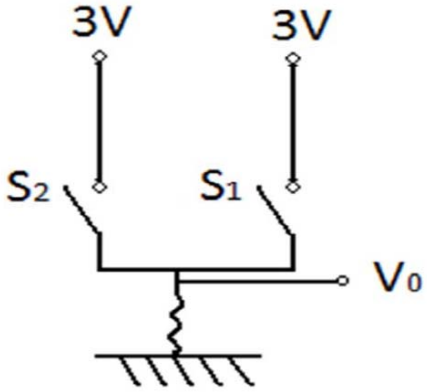
- S ON, V0= “1” (3V)
- S OFF, V0= “0” (0V)



Digital Systems and Switching Circuits

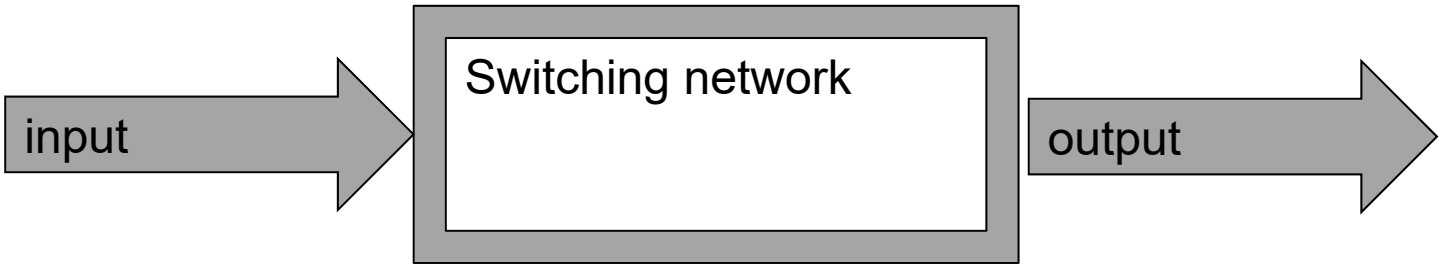


- Question:

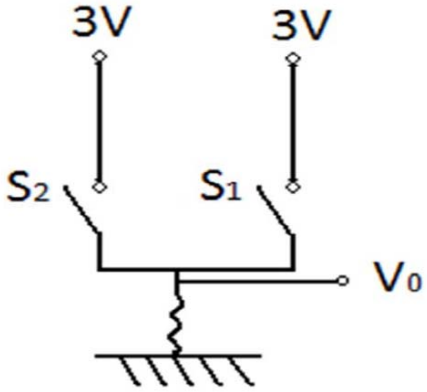


- S1 on, S2 on V0 = ____.
- S1 on, S2 off V0 = ____.
- S1 off, S2 on V0 = ____.
- S1 off, S2 off V0 = ____.

Digital Systems and Switching Circuits



- Ans:
- S1 on, S2 on V0=(3V) “1”
- S1 on, S2 off V0=(3V) “1”
- S1 off, S2 on V0=(3V) “1”
- S1 off, S2 off V0=(0V) “0”



1.2 數字系統與轉換

❖位置記號法（positional notation）

- 10的乘冪：

$$953.78_{10} = 9\times10^{2} + 5\times10^{1} + 3\times10^{0} + 7\times10^{-1} + 8\times10^{-2}$$

- 2 的乘冪：：

$$1011.11_{2} = 1\times2^{3} + 0\times2^{2} + 1\times2^{1} + 1\times2^{0} + 1\times2^{-1} + 1\times2^{-2}$$

$$= 8 + 0 + 2 + 1 + \frac{1}{2} + \frac{1}{4} = 11\frac{3}{4} = 11.75_{10}$$

1.2 數字系統與轉換

❖ R的乘冪：

- 任意一個正整數R（$R>1$）都可以被當作數字系統的**基數**（radix）或**底數**（base）

$$N=\left(a_4a_3a_2a_1a_0.a_{-1}a_{-2}a_{-3}\right)_R$$
$$=a_4\times R^4+a_3\times R^3+a_2\times R^2+a_1\times R^1+a_0\times R^0$$
$$+a_{-1}\times R^{-1}+a_{-2}\times R^{-2}+a_{-3}\times R^{-3}$$

其中a_i是R^i的係數，且$0\le a_i\le R-1$。

1.2 數字系統與轉換

❖ 例如：

$$147.3_8 = 1\times8^2 + 4\times8^1 + 7\times8^0 + 3\times8^{-1} = 64 + 32 + 7 + \frac{3}{8}$$
$$= 103.375_{10}$$

❖ 通常使用字母來代表大於9的數字。例如：在十六進位（底數為16）中，A表示10_{10}、B 表示11_{10}、C 表示12_{10}、D 表示13_{10}、E 表示14_{10} 及F 表示15_{10}。

$$A2F_{16} = 10\times16^2 + 2\times16^1 + 15\times16^0 = 2560 + 32 + 15 = 2607_{10}$$

1.2 數字系統與轉換

❖利用除法將十進位的整數轉換成底數R。

$$N=\left(a_n a_{n-1}\cdots a_2 a_1 a_0\right)_R = a_n R^n + a_{n-1}R^{n-1} + \cdots + a_2 R^2 + a_1 R^1 + a_0$$

$$\frac{N}{R} = a_n R^{n-1} + a_{n-1}R^{n-2} + \cdots + a_2 R^1 + a_1 = Q_1 \quad \text{，餘數} a_0$$

$$\frac{Q_1}{R} = a_n R^{n-2} + a_{n-1}R^{n-3} + \cdots + a_3 R^1 + a_2 = Q_2 \quad \text{，餘數} a_1$$

$$\frac{Q_2}{R} = a_n R^{n-3} + a_{n-1}R^{n-4} + \cdots + a_3 = Q_3 \quad \text{，餘數} a_2$$

1.2 數字系統與轉換

- 這個步驟持續下去，直到最後求得a_n。
- 注意到每次除法所得的餘數是一個所要的數字，並且最先求得的是**最低有效數元**（least significant digit）。

範例

❖ 將 53_{10} 轉換成二進位數

$$
\begin{array}{rll}
2 \,\big/\!\underline{53} & & \\
2 \,\big/\!\underline{26} & \text{餘數} = 1 = a_0 & \\
2 \,\big/\!\underline{13} & \text{餘數} = 0 = a_1 & \\
2 \,\big/\!\underline{6} & \text{餘數} = 1 = a_2 & 53_{10} = 110101_2 \\
2 \,\big/\!\underline{3} & \text{餘數} = 0 = a_3 & \\
2 \,\big/\!\underline{1} & \text{餘數} = 1 = a_4 & \\
0 & \text{餘數} = 1 = a_5 &
\end{array}
$$

1.2 數字系統與轉換

- 可以利用連續乘以R的乘法，將十進位的小數轉換成底數R。

$$F = \left(0.a_{-1}a_{-2}a_{-3}\cdots a_{-m}\right)_{\mathrm{R}} = a_{-1}R^{-1} + a_{-2}R^{-2} + a_{-3}R^{-3} + \cdots + a_{-m}R^{-m}$$

$$FR = a_{-1} + a_{-2}R^{-1} + a_{-3}R^{-2} + \cdots + a_{-m}R^{-m+1} = a_{-1} + F_1$$

$$F_1R = a_{-2} + a_{-3}R^{-1} + \cdots + a_{-m}R^{-m+2} = a_{-2} + F_2$$

$$F_2R = a_{-3} + \cdots + a_{-m}R^{-m+3} = a_{-3} + F_3$$

- 這個步驟持續下去，直到求得足夠的位數。注意到每次所得的整數部分是一個所要的數字，並且最先求得的是**最高有效數元**（most significant digit）。

範例

❖ 將 0.625_{10} 轉換成二進位數。

$$
\begin{array}{rrrr}
F = & 0.625 & F_1 = & 0.250 & F_2 = & 0.500 \\
\times & 2 & \times & 2 & \times & 2 \\
\hline
 & 1.250 & & 0.500 & & 1.000 \\
 & (a_{-1} = 1) & & (a_{-2} = 0) & & (a_{-3} = 1)
\end{array}
\qquad 0.625_{10} = 0.101_2
$$

範例

❖ 將 0.7_{10} 轉換成二進位數。

$$
\begin{array}{r}
0.7 \\
\underline{\quad\quad 2} \\
(1)\ 0.4 \\
\underline{\quad\quad 2} \\
(0)\ 0.8 \\
\underline{\quad\quad 2} \\
(1)\ 0.6 \\
\underline{\quad\quad 2} \\
(1)\ 0.2 \\
\underline{\quad\quad 2} \\
(0)\ 0.4 \quad \leftarrow \text{從此處開始循環，因為之前就已經出現0.4} \\
\underline{\quad\quad 2} \\
(0)\ 0.8
\end{array}
$$

$$0.7_{10} = 0.1\ \underline{0110}\ \underline{0110}\ \underline{0110}\ldots_2$$

範例

❖ 將 231.3_4 轉換成底數7。

$$231.3_4 = 2\times 16 + 3\times 4 + 1 + \frac{3}{4} = 45.75_{10}$$

$$\begin{array}{r|l} 7 & 45 \\ \hline 7 & 6 \quad \text{餘數} = 3 \\ \hline & 0 \quad \text{餘數} = 6 \end{array}$$

$$\begin{array}{rr} & 0.75 \\ & 7 \\ \hline (5) & 0.25 \\ & 7 \\ \hline (1) & 0.75 \\ & 7 \\ \hline (5) & 0.25 \\ & 7 \\ \hline (1) & 0.75 \end{array}$$

$$45.75_{10} = 63.5151\ldots_7$$

1.2 數字系統與轉換

❖ 二進位到十六進位（或反過來）的轉換

$$1001101.010111_2 = \underbrace{0100}_{4}\ \underbrace{1101}_{D}.\underbrace{0101}_{5}\ \underbrace{1100}_{C} = 4D.5C_{16} \tag{1-1}$$

1.3 二進制算術

❖ 二進位數的加法表如下：

$$0+0=0$$

$$0+1=1$$

$$1+0=1$$

$$1+1=0$$ 同時進位1到下一行

範例

❖ 用二進制執行13_{10}與11_{10}相加。

$$\begin{array}{l} \quad\quad 1111 \leftarrow\!\!-\!\!- \text{進位} \\ 13_{10} = \quad 1101 \\ \underline{11_{10} = \quad 1011} \\ \quad\quad 11000 = 24_{10} \end{array}$$

1.3 二進制算術

❖ 二進位數的減法表如下：

$$0-0=0$$

$$0-1=1 \quad \text{同時從上一行借位1}$$

$$1-0=1$$

$$1-1=0$$

範例　二進位減法的範例

(a)

1←——(顯示從第3行借位)

```
  11101
 –10011
 ------
   1010
```

(b)

1111←——借位

```
  10000
 –   11
 ------
   1101
```

(c)

111←——借位

```
  111001
 –  1011
 -------
  101110
```

1.3 二進制算術

❖ 二進位數的乘法表如下所示：

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

1.3 二進制算術

❖ 底下的例子說明13_{10} 乘以11_{10} 的二進位乘法運算：

$$
\begin{array}{r}
1101 \\
\times\quad 1011 \\
\hline
1101 \\
1101 \\
0000 \\
1101 \\
\hline
10001111 = 143_{10}
\end{array}
$$

1.3 二進制算術

❖ 底下的例子說明145_{10}除以11_{10}的二進位除法運算：

```
          1101
    ┌──────────
1011│ 10010001
     1011
     ----
      1110
      1011
      ----
       1101
       1011
       ----
         10
```

所得的商為 1101，餘數為 10。

2 補數法的數字

- 2 補數系統：
  - 正數N表示法是一個0接著如符號與大小法表示的大小N。
  - 負數–N是用其2補數N^*來表示。
- 若字組長度為n位元，則正整數N 的2 補數定義如下：

$$N^* = 2^n - N \qquad (1\text{-}2)$$

- 表1-1顯示n=4 的結果，如表1-1所示，2 補數法從–1到– 7的負數可以取其正數1 到7 的2補數而得（亦即用16去減）。

1.4 負數的表示法

表 1-1　有號二進制整數（字組長度：$n=4$）

$+N$	正整數（各系統）	$-N$	負整數		
			符號與大小	2 補數 N^*	1 補數 $\overline{N}$
+0	0000	−0	1000	——	1111
+1	0001	−1	1001	1111	1110
+2	0010	−2	1010	1110	1101
+3	0011	−3	1011	1101	1100
+4	0100	−4	1100	1100	1011
+5	0101	−5	1101	1011	1010
+6	0110	−6	1110	1010	1001
+7	0111	−7	1111	1001	1000
		−8	——	1000	——

2 補數數字之加法

- n位元**二進位有號數**（signed binary number）的加法。
- $n = 4$時，
  1. 兩個正數相加，和 $< 2^{n-1}$。

+3	0011	
+4	0100	
+7	0111	（正確解）

2 補數數字之加法

2. 兩個正數相加，和 $\geq 2^{n-1}$。

+5　0101

+6　0110

1011 ←——由於溢位，所以答案錯誤

（+11包括符號共須5位元方能表示）

3. 正、負數相加（負數之值較大時）。

+5　0101

–6　1010

–1　1111　（正確解）

2 補數數字之加法

4. 與例3 相同，但正數之值較大時。

$$\begin{array}{lll} -5 & & 1011 \\ \underline{+6} & & \underline{0110} \\ +1 & (1)\ 0001 & \longleftarrow \text{忽略符號位元之進位，為正確解} \\ & & \text{（這不是溢位）} \end{array}$$

5. 兩個負數相加，$|\text{和}| \le 2^{n-1}$。

$$\begin{array}{lll} -3 & & 1011 \\ \underline{-4} & & \underline{0110} \\ -7 & (1)\ 1001 & \longleftarrow \text{若忽略最後之進位，為正確解} \\ & & \text{（這不是溢位）} \end{array}$$

2 補數數字之加法

6. 兩個負數相加，｜和｜$> 2^{n-1}$。

$$
\begin{array}{ll}
-5 & 1011 \\
\underline{-6} & \underline{1010} \\
\quad (1) & 0101 \quad \longleftarrow\text{由於溢位，所以答案錯誤}
\end{array}
$$

（–11包括符號共須5位元方能表示）

2 補數數字之加法

- 證明如下：

例 4：$-A + B$（其中 $B > A$）

$$A^* + B = (2^n - A) + B = 2^n + (B - A) > 2^n$$

- 捨棄最後之進位，相當於減去2^n，所以結果是$(B - A)$為正確解！

2 補數數字之加法

例 5：$-A-B$（其中 $A+B \le 2^{n-1}$）

$$A^* + B^* = (2^n - A) + (2^n - B) = 2^n + 2^n - (A+B)$$

- 捨棄最後之進位，得到$2^n - (A + B) = (A + B)^*$，這正是$-(A + B)$之正確表示式。

1 補數法的數字

❖ 在1 補數系統中，一個負數–N 用N 的1 補數 $\overline{N}$ 表示，定義如下：

$$\overline{N} = (2^n - 1) - N \qquad (1\text{-}4)$$

1 補數數字之加法

- 1 補數數字之加法與2 補數加法非常類似，唯一不同的是1 補數加法並不捨棄最後的進位，而是將進位加在*n* 位元和的最小位元，此稱為**末位遞迴進位**（end-around carry）。

1 補數數字之加法

- 正數相加時的情形與2補數中的例1及例2完全相同，其他的例子分別說明如下（n=4）。

3. 正、負數相加（負數之值較大時）。

$$\begin{array}{rll} +5 & 0101 & \\ \underline{-6} & \underline{1001} & \\ -1 & 1110 & \text{（正確解）} \end{array}$$

4. 與例3 相同，但正數之值較大時。

$$\begin{array}{rll} -5 & 1010 & \\ \underline{+6} & \underline{0110} & \\ (1) & 0000 & \\ & \underline{\longrightarrow 1} & \text{（末位遞迴進位）} \\ & 0001 & \text{（正確解，沒有溢位）} \end{array}$$

1 補數數字之加法

5. 兩個負數相加，｜和｜＜ 2^{n-1} 。

–3	1100	
–4	1011	
	(1) 0111	
	→ 1	（末位遞迴進位）
	1000	（正確解，沒有溢位）

1 補數數字之加法

6. 兩個負數相加，｜和｜$\geq 2^{n-1}$。

$$
\begin{array}{lll}
-5 & 1010 & \\
\underline{-6} & \underline{1001} & \\
 & (1)\ 0011 & \\
 & \quad \underline{\longrightarrow 1} & \text{（末位遞迴進位）} \\
 & 0100 & \text{（錯誤解，因為有溢位）}
\end{array}
$$

1 補數數字之加法

❖ 證明如下：

例 4：$-A+B$（其中 $B>A$）

$$\overline{A}+B=(2^n-1-A)+B=2^n+(B-A)-1$$

❖ 末位遞迴進位相當於減去2^n後再加1，所以結果是$(B-A)$為正確解。

1 補數數字之加法

例 5：$-A-B$ （$A+B \le 2^{n-1}$ ）

$$\overline{A}+\overline{B}=(2^n-1-A)+(2^n-1-B)=2^n+[2^n-1-(A+B)]-1$$

❖ 末位遞迴進位之後，結果是 $2^n-1-(A+B)=(\overline{A+B})$ ，此為 $-(A+B)$ 之正確表示式。

1.4 負數的表示法

- 以下的例子分別說明字組長度$n = 8$ 時，1補數與2補數之加法運算：
  1. 以1 補數法執行–11 及– 20 之加法運算。

     +11 = 00001011　　+20 = 00010100

     各位元逐一取補數，

     –11以11110100 表示，而–20以11101011 表示

$$\begin{array}{rl} & 11110100 \quad (-11) \\ & \underline{11101011 \quad +(-20)} \\ (1)\ & 11011111 \\ & \underline{\longrightarrow 1 \quad \text{（末位遞迴進位）}} \\ & 11100000 = -31 \end{array}$$

1 補數數字之加法

2. 以2 補數法執行–8 及+19 之加法運算。

+8 = 00001000

將第一個1左側的所有位元取補數， –8表示成11111000

$$\begin{array}{ll} 11111000 & (-8) \\ 00010011 & \underline{+19} \\ \cancel{(1)}\ 00001011 = +11 & \end{array}$$

（捨棄最後之進位）

1.5 二進制碼

- **二進制編碼的十進位**（binary-coded-decimal, BCD），或者更明確地稱為8-4-2-1 BCD。
- **超3 碼**（excess-3 code）
- 8-4-2-1（BCD）碼和6-3-1-1 碼為加權碼
- **5占2碼**（2-out-of-5 code）
- **格雷碼**（Gray code）　（表1-2）
- **美國資訊交換標準碼**（American Standard Code for Information Interchange, ASCII）

1.5 二進制碼

表 1-2 十進位數的二進制碼

十進位數	8-4-2-1 碼（BCD）	6-3-1-1 碼	超 3 碼	5 占 2 碼	格雷碼
0	0000	0000	0011	00011	0000
1	0001	0001	0100	00101	0001
2	0010	0011	0101	00110	0011
3	0011	0100	0110	01001	0010
4	0100	0101	0111	01010	0110
5	0101	0111	1000	01100	1110
6	0110	1000	1001	10001	1010
7	0111	1001	1010	10010	1011
8	1000	1011	1011	10100	1001
9	1001	1100	1100	11000	1000

1.5 二進制碼

- 表1-3列出ASCII 碼的一部分，沒有列出來的編碼組合被用作特殊的控制功能，如「**饋頁**」（form feed）或「**傳輸結束**」（end of transmission）。「Start」這個字用ASCII碼表示如下：

1010011	1110100	1100001	1110010	1110100
S	t	a	r	t

表 1-3 ASCII 碼

字元	ASCII 碼 A_6	A_5	A_4	A_3	A_2	A_1	A_0	字元	ASCII 碼 A_6	A_5	A_4	A_3	A_2	A_1	A_0	字元	ASCII 碼 A_6	A_5	A_4	A_3	A_2	A_1	A_0
space	0	1	0	0	0	0	0	@	1	0	0	0	0	0	0	‘	1	1	0	0	0	0	0
!	0	1	0	0	0	0	1	A	1	0	0	0	0	0	1	a	1	1	0	0	0	0	1
“	0	1	0	0	0	1	0	B	1	0	0	0	0	1	0	b	1	1	0	0	0	1	0
#	0	1	0	0	0	1	1	C	1	0	0	0	0	1	1	c	1	1	0	0	0	1	1
$	0	1	0	0	1	0	0	D	1	0	0	0	1	0	0	d	1	1	0	0	1	0	0
%	0	1	0	0	1	0	1	E	1	0	0	0	1	0	1	e	1	1	0	0	1	0	1
&	0	1	0	0	1	1	0	F	1	0	0	0	1	1	0	f	1	1	0	0	1	1	0
′	0	1	0	0	1	1	1	G	1	0	0	0	1	1	1	g	1	1	0	0	1	1	1
(	0	1	0	1	0	0	0	H	1	0	0	1	0	0	0	h	1	1	0	1	0	0	0
)	0	1	0	1	0	0	1	I	1	0	0	1	0	0	1	i	1	1	0	1	0	0	1
*	0	1	0	1	0	1	0	J	1	0	0	1	0	1	0	j	1	1	0	1	0	1	0
+	0	1	0	1	0	1	1	K	1	0	0	1	0	1	1	k	1	1	0	1	0	1	1
,	0	1	0	1	1	0	0	L	1	0	0	1	1	0	0	l	1	1	0	1	1	0	0
−	0	1	0	1	1	0	1	M	1	0	0	1	1	0	1	m	1	1	0	1	1	0	1
.	0	1	0	1	1	1	0	N	1	0	0	1	1	1	0	n	1	1	0	1	1	1	0
/	0	1	0	1	1	1	1	O	1	0	0	1	1	1	1	o	1	1	0	1	1	1	1
0	0	1	1	0	0	0	0	P	1	0	1	0	0	0	0	p	1	1	1	0	0	0	0
1	0	1	1	0	0	0	1	Q	1	0	1	0	0	0	1	q	1	1	1	0	0	0	1
2	0	1	1	0	0	1	0	R	1	0	1	0	0	1	0	r	1	1	1	0	0	1	0
3	0	1	1	0	0	1	1	S	1	0	1	0	0	1	1	s	1	1	1	0	0	1	1
4	0	1	1	0	1	0	0	T	1	0	1	0	1	0	0	t	1	1	1	0	1	0	0
5	0	1	1	0	1	0	1	U	1	0	1	0	1	0	1	u	1	1	1	0	1	0	1
6	0	1	1	0	1	1	0	V	1	0	1	0	1	1	0	v	1	1	1	0	1	1	0
7	0	1	1	0	1	1	1	W	1	0	1	0	1	1	1	w	1	1	1	0	1	1	1
8	0	1	1	1	0	0	0	X	1	0	1	1	0	0	0	x	1	1	1	1	0	0	0
9	0	1	1	1	0	0	1	Y	1	0	1	1	0	0	1	y	1	1	1	1	0	0	1
:	0	1	1	1	0	1	0	Z	1	0	1	1	0	1	0	z	1	1	1	1	0	1	0
;	0	1	1	1	0	1	1	[	1	0	1	1	0	1	1	{	1	1	1	1	0	1	1
<	0	1	1	1	1	0	0	\	1	0	1	1	1	0	0	|	1	1	1	1	1	0	0
=	0	1	1	1	1	0	1	]	1	0	1	1	1	0	1	}	1	1	1	1	1	0	1
>	0	1	1	1	1	1	0	^	1	0	1	1	1	1	0	~	1	1	1	1	1	1	0
?	0	1	1	1	1	1	1	—	1	0	1	1	1	1	1	delete	1	1	1	1	1	1	1