

Digital Logic Design

數位邏輯設計 **全**

第 6 章 組合邏輯電路設計及應用

6-1 組合邏輯電路設計步驟

6-2 加法器及減法器

6-3 二進碼十進數 (BCD) 加法器

6-4 解碼器及編碼器

6-5 多工器及解多工器

6-6 比較器

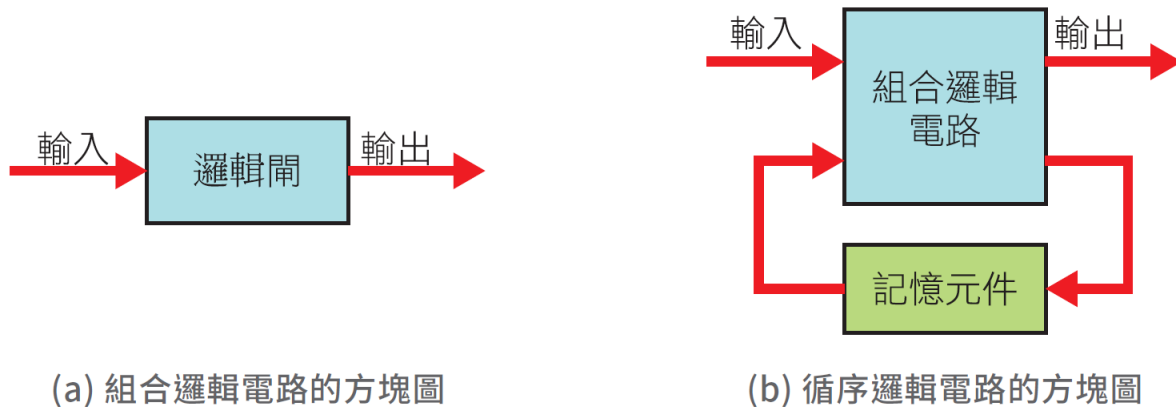
6-7 應用實例的認識—
多工器的應用



第 6 章 組合邏輯電路設計及應用

數位邏輯電路可以區分為組合邏輯（combinational logic）電路與循序邏輯（sequential logic）電路兩種類型。組合邏輯電路是由各種邏輯閘所組成，其輸出的狀態只與目前的輸入狀態有關，而與前一個輸出狀態無關，如圖6-1(a)所示。而循序邏輯電路是由組合邏輯電路與記憶元件（即正反器）所組成，如圖6-1(b)所示，由於記憶元件可以儲存前一個輸出的狀態，所以循序邏輯的輸出狀態是由前一個輸出狀態與目前的輸入狀態來決定，以達到循序的功能。

第 6 章 組合邏輯電路設計及應用



▲ 圖 6-1 組合邏輯與循序邏輯電路的方塊圖

本章將針對組合邏輯電路設計及應用做詳細的介紹，而正反器與循序邏輯電路將在第七、八章中介紹。

6-1 組合邏輯電路設計步驟

組合邏輯電路設計是將文字敘述的問題轉換成組合邏輯電路圖，其設計的步驟如下：

1. 決定輸入與輸出變數的個數與名稱。
2. 定義輸入與輸出變數。
3. 依題意建立真值表。
4. 利用布林定理或卡諾圖，化簡輸出函數為最簡布林代數式。
5. 畫出組合邏輯電路圖。

6-1 組合邏輯電路設計步驟

例題 6-1 三人投票器

設計一個三人投票器。

解

1. 決定輸入與輸出變數的個數與名稱：

(1) 輸入變數3 個： A 、 B 、 C 。

(2) 輸出變數1 個： F 。

2. 定義輸入與輸出變數：

(1) 輸入變數 A 、 B 、 C ：贊成以1 表示，不贊成以0 表示。

(2) 輸出變數 F ：表決結果通過以1 表示，未通過以0 表示。

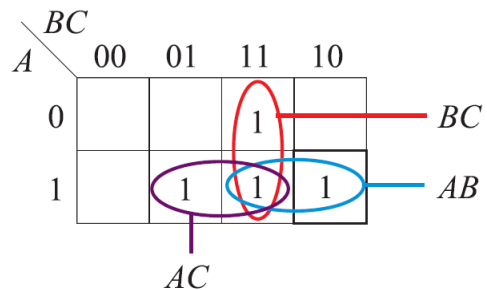
6-1 組合邏輯電路設計步驟

3. 依題意建立真值表：

輸入			輸出
A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

6-1 組合邏輯電路設計步驟

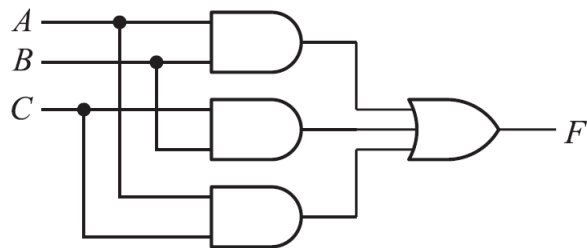
4. 利用卡諾圖，化簡輸出函數 F 為最簡SOP 布林代數式。



$$\therefore F(A, B, C) = AB + BC + AC$$

6-1 組合邏輯電路設計步驟

5. 畫出組合邏輯電路圖：



演練 1

設計一個兩輸入XOR 閘。

6-2 加法器及減法器

根據上一節所敘述的組合邏輯電路設計步驟，我們可以設計出各種不同的組合邏輯應用電路。一般而言，常用的組合邏輯應用電路有加法器、減法器、解碼器、編碼器、多工器、解多工器與比較器等。

一 加法器

兩個二進位數字相加，若和大於或等於2 即進位。我們以兩個1 位元的二進位數相加，以及兩個4 位元的二進位數相加為例，說明二進位的加法運算，如表6-1 所示。

6-2 加法器及減法器

▼ 表 6-1 二進位的加法運算

兩個 1 位元的二進位數相加				兩個 4 位元的二進位數相加			
$\begin{array}{r} 0 \\ + 0 \\ \hline 0 \end{array}$	$\begin{array}{r} 0 \\ + 1 \\ \hline 1 \end{array}$	$\begin{array}{r} 1 \\ + 0 \\ \hline 1 \end{array}$	$\begin{array}{r} 1 \\ + 1 \\ \hline 10 \end{array}$	$\begin{array}{r} 111 \leftarrow \text{前一級進位} \\ 1101 \leftarrow \text{被加數} \\ + 0111 \leftarrow \text{加數} \\ \hline 10100 \end{array}$	$\begin{array}{r} 10100 \\ \hline 10100 \end{array}$	$\begin{array}{r} 10100 \\ \hline 10100 \end{array}$	$\begin{array}{r} 10100 \\ \hline 10100 \end{array}$
↑	↑	↑	↑	進位 → 1	↑	↑	↑
半	半	半	半		全	全	半
加	加	加	加		加	加	加
器	器	器	器		器	器	器

執行二進位加法運算的電路稱為加法器。二進位加法器有半加器（half adder；簡稱HA）與全加器（full adder；簡稱FA）兩種，另外執行兩個多位元的二進位數加法運算的電路稱為並加器，分別說明如下。

6-2 加法器及減法器

(一) 半加器

執行兩個一位元（被加數與加數）的二進位數加法運算之電路稱為半加器（HA），它有被加數與加數兩個輸入端，以及進位（carry）與和（sum）兩個輸出端。半加器設計的步驟說明如下：

1. 決定輸入與輸出變數的個數與名稱：
 - (1) 輸入變數2 個：被加數 A 、加數 B 。
 - (2) 輸出變數2 個：進位 C 、和 S 。

6-2 加法器及減法器

2. 定義輸入與輸出變數：

$$\begin{array}{r}
 A \leftarrow \text{被加數} \\
 + \quad B \leftarrow \text{加數} \\
 \hline
 C \quad S \\
 \uparrow \quad \uparrow \\
 \text{進位} \quad \text{和}
 \end{array}$$

3. 依題意建立真值表，如表6-2 所示。

▼ 表 6-2 半加器的真值表

輸入		輸出	
被加數	加數	進位	和
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

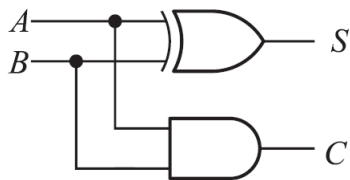
6-2 加法器及減法器

4. 依據表6-2 之真值表，寫出輸出函數C 與S 的最簡SOP 布林代數式：

$$\text{進位 } C(A, B) = AB$$

$$\text{和 } S(A, B) = \bar{A}B + A\bar{B} = A \oplus B$$

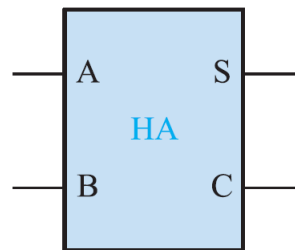
5. 畫出組合邏輯電路圖：如圖6-2 所示。



▲ 圖 6-2 半加器的邏輯電路圖

6-2 加法器及減法器

6. 半加器的符號：如圖6-3 所示。



▲ 圖 6-3 半加器的符號

6-2 加法器及減法器

(二) 全加器

執行兩個一位元與前一級進位（共三個1 位元）的二進位數加法運算電路稱為全加器（FA），它具有被加數、加數與前一級進位三個輸入端，以及進位與和兩個輸出端。全加器設計的步驟說明如下：

1. 決定輸入與輸出變數的個數與名稱：

(1) 輸入變數3 個：被加數 A 、加數 B 、前一級進位 C_i 。

(2) 輸出變數2 個：進位 C_o 、和 S

6-2 加法器及減法器

2. 定義輸入與輸出變數：

$$\begin{array}{rcl}
 & C_i & \leftarrow \text{前一級進位} \\
 & A & \leftarrow \text{被加數} \\
 + & B & \leftarrow \text{加數} \\
 \hline
 C_o & S & \\
 \uparrow & \uparrow & \\
 \text{進} & \text{和} & \\
 \text{位} & &
 \end{array}$$

6-2 加法器及減法器

3. 依題意建立真值表，如表6-3所示。

▼ 表 6-3 全加器的真值表

輸入			輸出	
被加數	加數	前一級進位	進位	和
A	B	C_i	C_o	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

6-2 加法器及減法器

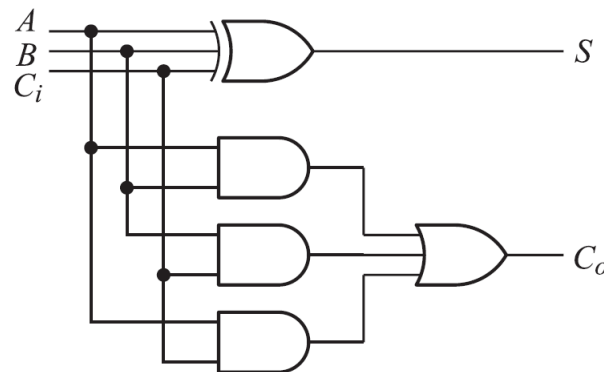
4. 利用卡諾圖，化簡輸出函數 C_o 與 S 為最簡SOP 布林代數式：
如表6-4 所示。

▼ 表 6-4 輸出函數 C_o 與 S 的最簡布林代數式

	進位 C_o	和 S
卡諾圖		
最簡布林代數式	$\therefore \text{進位 } C_o(A, B, C_i) = AB + BC_i + AC_i$	$\begin{aligned} \therefore \text{和 } S(A, B, C_i) &= \overline{A}\overline{B}C_i + \overline{A}B\overline{C_i} + A\overline{B}\overline{C_i} + ABC_i = A \oplus B \oplus C_i \\ &(\text{因為當輸入端有奇數個 } 1 \text{ 時，輸出端為 } 1) \end{aligned}$

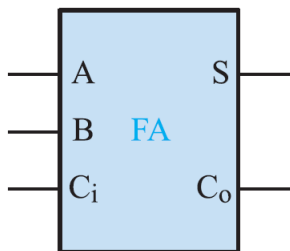
6-2 加法器及減法器

5. 畫出組合邏輯電路圖：如圖6-4 所示。



▲ 圖 6-4 全加器的邏輯電路圖

6. 全加器的符號：如圖6-5 所示。



▲ 圖 6-5 全加器的符號

6-2 加法器及減法器

7. 全加器也可以由兩個半加器與一個OR 閘組成。

依據表 6-3 之真值表，寫出輸出端進位 C_o 與和 S 的布林代數式為

$$\begin{aligned}\text{進位 } C_0(A, B, C_i) &= \overline{A}BC_i + A\overline{B}C_i + AB\overline{C}_i + ABC_i \\ &= AB(\overline{C}_i + C_i) + C_i(\overline{A}B + A\overline{B}) \\ &= AB \cdot 1 + C_i(A \oplus B)\end{aligned}$$

6-2 加法器及減法器

$$C_o(A, B, C_i) = AB + C_i(A \oplus B)$$

6-1 式

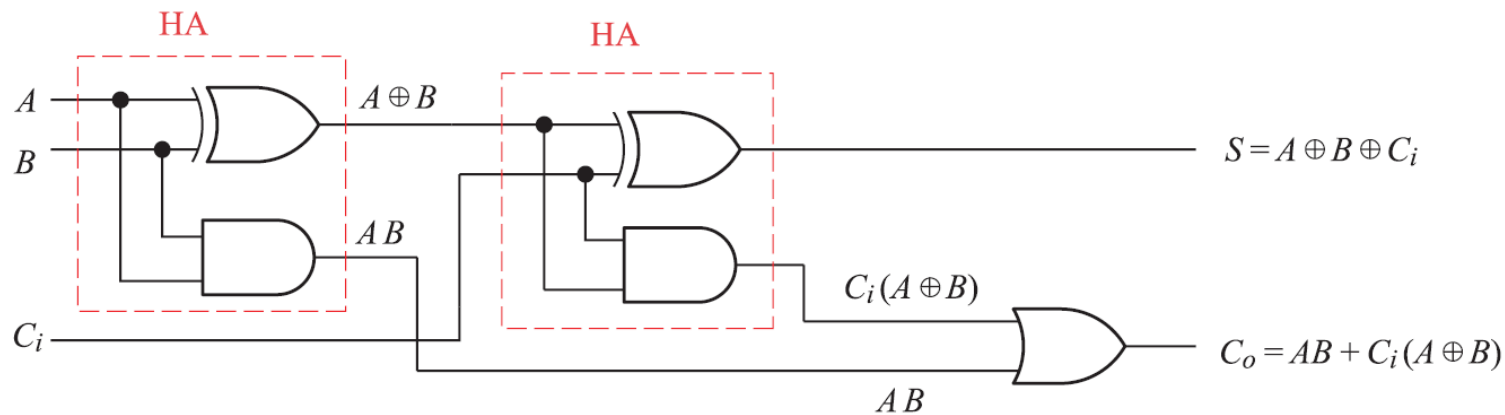
和

$$S(A, B, C_i) = A \oplus B \oplus C_i = (A \oplus B) \oplus C_i$$

6-2 式

6-2 加法器及減法器

根據6-1 式與6-2 式，可以畫出由兩個半加器與一個OR 閘組成之全加器，如圖6-6 所示。



▲ 圖 6-6 兩個半加器與一個 OR 閘組成的全加器

6-2 加法器及減法器

例題 6-2 全加器

若全加器輸入端 $A = 1$ ， $B = 0$ ， $C_i = 1$ ，其輸出端進位 C_o 與和 S 的結果為何？

解

$$\begin{array}{r}
 A \\
 B \\
 + C_i \\
 \hline
 C_o \quad S
 \end{array}
 \longrightarrow
 \begin{array}{r}
 1 \\
 0 \\
 + 1 \\
 \hline
 10
 \end{array}$$

進位 $C_o = 1$ ；和 $S = 0$ 。

演練 2

若全加器輸入端 $A = 1$ ， $B = 1$ ， $C_i = 1$ ，其輸出端進位 C_o 與和 S 的結果為何？

6-2 加法器及減法器

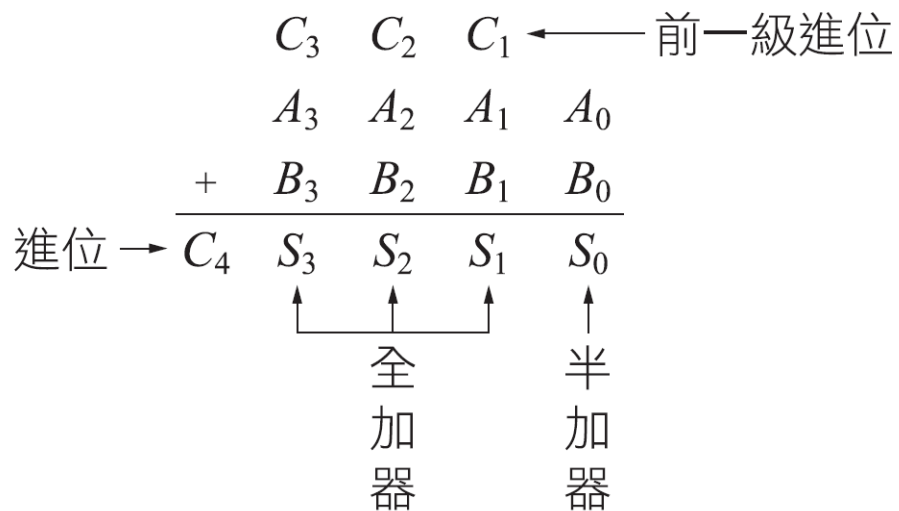
(三) 並加器

全加器一般很少單獨使用，因為它只能執行兩個一位元與前一級進位的二進位數加法運算，若要執行多個位元的二進位數加法運算，則需要將數個全加器並列以串接方式連接在一起才能完成，我們稱之為並加器（**parallel adder**）。我們以四位元並加器為例，說明並加器的原理。

6-2 加法器及減法器

1. 四位元並加器的定義

兩個四位元的二進位數加法運算，舉例說明如下：



6-2 加法器及減法器

其中 A_3 、 A_2 、 A_1 、 A_0 為四個位元的被加數。

B_3 、 B_2 、 B_1 、 B_0 為四個位元的加數。

C_3 、 C_2 、 C_1 為前一級的進位。

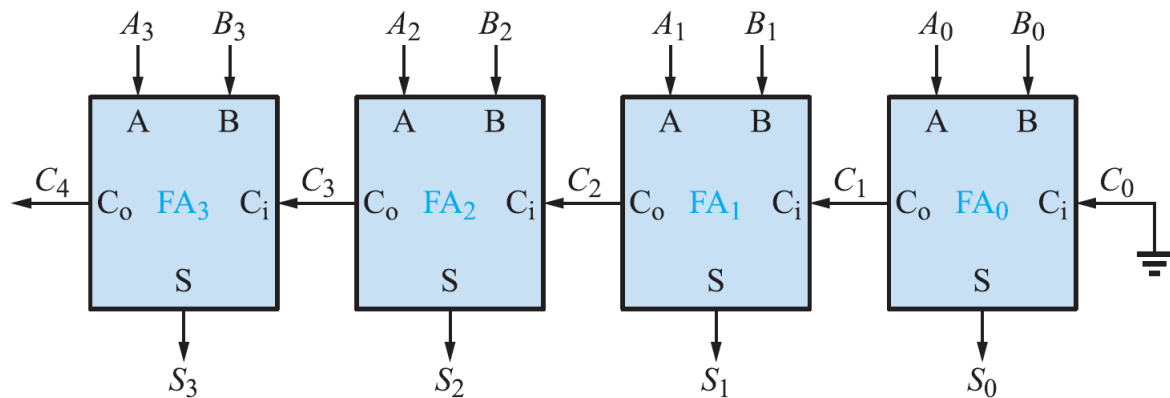
S_3 、 S_2 、 S_1 、 S_0 為四個位元的總和。

C_4 為最高位元的進位。

6-2 加法器及減法器

2. 四位元並加器的電路圖

依據上述的定義，我們可以利用三個全加器與一個半加器，或四個全加器組成一個四位元並加器，如圖6-7 所示，其中最右邊的全加器 FA_0 只有執行兩個1 位元（ A_0 與 B_0 ）的二進位數相加，並無前一級進位輸入，故將前一級的進位輸入 C_0 接地，相當於半加器。



▲ 圖 6-7 四位元並加器的電路圖

6-2 加法器及減法器

3. 四位元並加器的電路原理

如圖6-7所示，其被加數 $A_3A_2A_1A_0$ 與加數 $B_3B_2B_1B_0$ 雖同時送至全加器相加，但是必須先將最低有效位元（LSB） A_0 與 B_0 經過全加器 FA_0 相加後，和為 S_0 ，其進位 C_1 再與 A_1 與 B_1 經過全加器 FA_1 相加後，以此類推，才能得到四個位元的總和 S_3 、 S_2 、 S_1 、 S_0 與最高位元的進位 C_4 ，所以四位元並加器的進位有傳遞延遲現象，因此降低了並加器的運算速度。

為了解決此一問題，可以將所有位元的被加數與加數，直接透過組合邏輯電路運算，以取得正確的進位，就可以提高並加器的運算速度，稱之為前瞻式進位（**look ahead carry**），例如TTL的7483就是採用前瞻式進位的四位元並加器。

6-2 加法器及減法器

二 減法器

執行二進位減法運算的電路稱為減法器。二進位減法器有半減器（half subtractor；簡稱HS）與全減器（full subtractor；簡稱FS）兩種，另外執行兩個多位元的二進位數減法運算的電路稱為並減器，分別說明如下。

（一）半減器

執行兩個一位元（被減數與減數）的二進位數減法運算之電路稱為半減器（HS），它具有被減數與減數兩個輸入端，以及借位（borrow）與差（difference）兩個輸出端。半減器設計的步驟說明如下：

6-2 加法器及減法器

1. 決定輸入與輸出變數的個數與名稱：
 - (1) 輸入變數2 個：被減數 X 、減數 Y 。
 - (2) 輸出變數2 個：借位 B 、差 D 。
2. 定義輸入與輸出變數：

$$\begin{array}{r}
 X \leftarrow \text{被減數} \\
 - \quad Y \leftarrow \text{減數} \\
 \hline
 B \quad D \\
 \uparrow \quad \uparrow \\
 \text{借位} \quad \text{差}
 \end{array}$$

6-2 加法器及減法器

3. 依題意建立真值表，如表6-5 所示。

▼ 表 6-5 半減器的真值表

輸入		輸出	
被減數	減數	借位	差
X	Y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

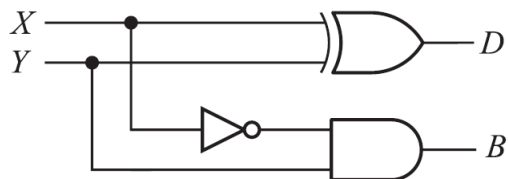
4. 依據表6-5 之真值表，寫出輸出函數 B 與 D 的最簡SOP 布林代數式：

$$\text{借位 } B(X,Y) = \overline{X}Y$$

$$\text{差 } D(X,Y) = \overline{X}Y + X\overline{Y} = X \oplus Y$$

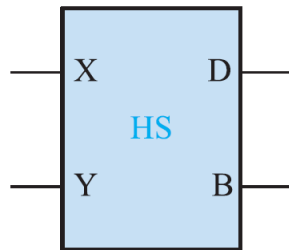
6-2 加法器及減法器

5. 畫出組合邏輯電路圖：如圖6-8 所示。



▲ 圖 6-8 半減器的邏輯電路圖

6. 半減器的符號：如圖6-9 所示。



▲ 圖 6-9 半減器的符號

6-2 加法器及減法器

(二) 全減器

執行兩個一位元與前一級借位（共三個1 位元）的二進位數減法運算之電路稱為全減器（*FS*），它具有被減數、減數與前一級借位三個輸入端，

以及借位與差兩個輸出端。全減器設計的步驟說明如下：

1. 決定輸入與輸出變數的個數與名稱：

(1) 輸入變數3 個：被減數 X 、減數 Y 、前一級借位 B_i 。

(2) 輸出變數2 個：借位 B_o 、差 D 。

6-2 加法器及減法器

2. 定義輸入與輸出變數：

$$\begin{array}{r}
 X \\
 - Y \\
 \hline
 B_o \quad D \\
 \uparrow \quad \uparrow \\
 \text{借位} \quad \text{差}
 \end{array}
 \quad
 \begin{array}{l}
 \text{---} B_i \leftarrow \text{前一級借位} \\
 \text{---} \leftarrow \text{被減數} \\
 \text{---} \leftarrow \text{減數}
 \end{array}$$

6-2 加法器及減法器

3. 依題意建立真值表，如表6-6 所示。

▼ 表 6-6 全減器的真值表

輸入			輸出	
被減數	減數	前一級 借位	借位	差
X	Y	B_i	B_o	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

6-2 加法器及減法器

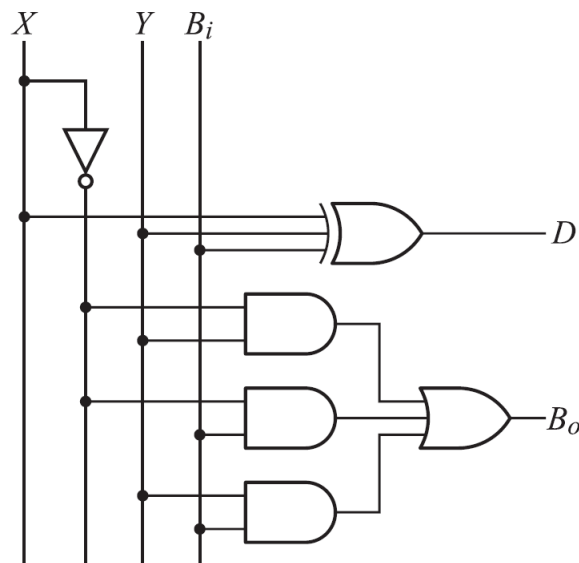
4. 利用卡諾圖，化簡輸出函數 B_o 與 D 為最簡SOP 布林代數式：
如表6-7 所示。

▼ 表 6-7 輸出函數 B_o 與 D 的最簡布林代數式

	借位 B_o	差 D
卡諾圖		
最簡布林代數式	$\therefore \text{借位 } B_o(X, Y, B_i) = \overline{X}Y + \overline{X}B_i + YB_i$	$\begin{aligned} \therefore \text{差 } D(X, Y, B_i) &= \overline{X}\overline{Y}B_i + \overline{X}Y\overline{B_i} + X\overline{Y}\overline{B_i} + XYB_i = X \oplus Y \oplus B_i \\ &\text{(因為當輸入端有奇數個 1 時，輸出端為 1)} \end{aligned}$

6-2 加法器及減法器

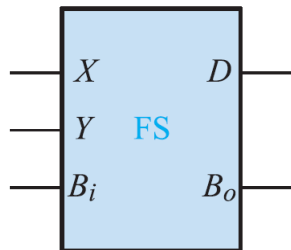
5. 畫出組合邏輯電路圖：如圖6-10 所示。



▲ 圖 6-10 全減器的邏輯電路圖

6-2 加法器及減法器

6. 全減器的符號：如圖6-11 所示。



▲ 圖 6-11 全減器的符號

6-2 加法器及減法器

7. 全減器也可以由兩個半減器與一個OR 閘組成。

依據表 6-6 之真值表，寫出輸出端借位 B_o 與差 D 的布林代數式為

$$\begin{aligned}
 \text{借位 } B_o(X, Y, B_i) &= \overline{X}YB_i + \overline{X}Y\overline{B_i} + \overline{X}YB_i + XYB_i \\
 &= \overline{X}Y(\overline{B_i} + B_i) + B_i(\overline{X}Y + XY) \\
 &= \overline{X}Y \cdot 1 + B_i(\overline{X \oplus Y})
 \end{aligned}$$

6-2 加法器及減法器

$$B_o(X, Y, B_i) = \overline{X}Y + B_i(\overline{X \oplus Y})$$

6-3 式

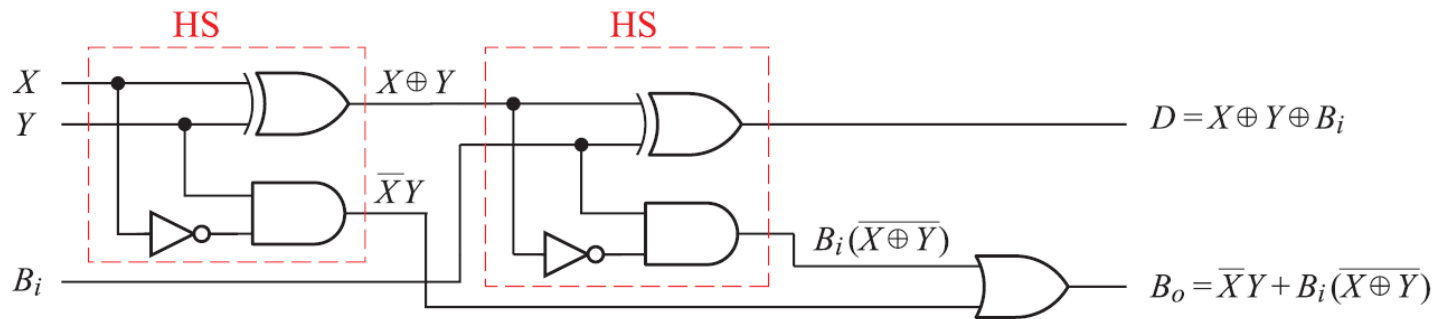
差

$$D(X, Y, B_i) = X \oplus Y \oplus B_i = (X \oplus Y) \oplus B_i$$

6-4 式

6-2 加法器及減法器

根據6-3 式與6-4 式，可以畫出由兩個半減器與一個OR 閘組成之全減器，如圖6-12 所示。



▲ 圖 6-12 兩個半減器與一個 OR 閘組成的全減器

6-2 加法器及減法器

例題 6-3 全減器

如圖6-12 所示之全減器，若輸入端 $X = 0$ ， $Y = 0$ ， $B_i = 1$ ，其輸出端借位 B_o 與差 D 的結果為何？

解

$$\begin{array}{r}
 \begin{array}{r}
 \overset{\textcolor{red}{\curvearrowright} B_i}{X} \\
 - \quad Y \\
 \hline
 B_o \quad D
 \end{array}
 \longrightarrow
 \begin{array}{r}
 \overset{\textcolor{red}{\curvearrowright} 1}{0} \\
 - \quad 0 \\
 \hline
 1 \quad 1
 \end{array}
 \end{array}$$

借位 $B_o = 1$ ；差 $D = 1$ 。

演練 3

如圖6-12 所示之全減器，若輸入端 $X = 0$ ， $Y = 1$ ， $B_i = 1$ ，其輸出端借位 B_o 與差 D 的結果為何？

6-2 加法器及減法器

(三) 並減器

全減器一般很少單獨使用，因為它只能執行兩個一位元與前一級借位的二進位數減法運算。通常在數位電路或計算機中，若要執行多個位元的二進位數減法運算，一般是利用加法器來做減法運算，以達到簡化電路結構的目的，常用的減法器有1 的補數並減器與2 的補數並減器兩種，分別說明如下。

6-2 加法器及減法器

1. 四位元1 的補數並減器

有關1 的補數減法運算可以參考第5 章的5-3 節「二進位減法」之內容。因為兩數相減之差 = 被減數 - 減數

$$= A - B$$

$$= A + (-B)$$

$$= A + (B \text{ 取1 的補數})$$

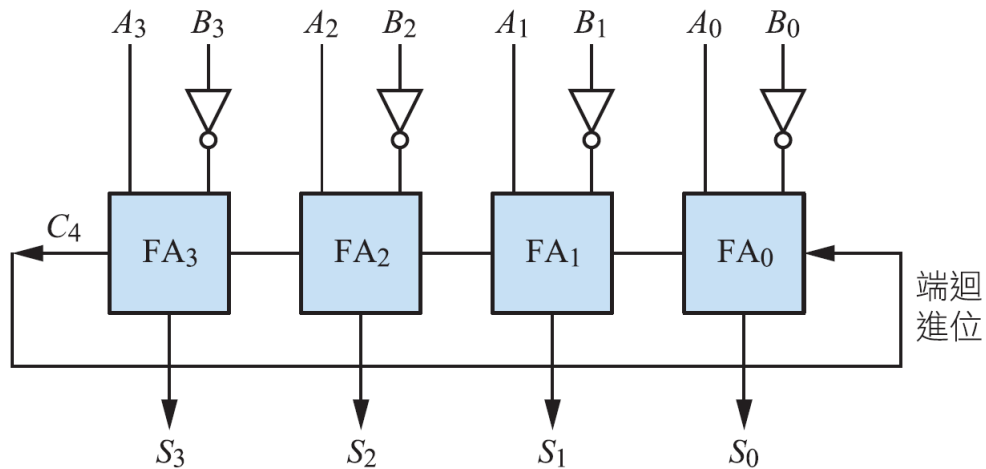
$$= A + \overline{B}$$

6-2 加法器及減法器

所以1的補數減法器只要將減數 B 取1的補數（即 $0 \rightarrow 1$ ， $1 \rightarrow 0$ ），再與被減數 A 相加，即可完成1的補數減法運算，若有進位產生時，則需將進位與和的最低有效位元（**LSB**）相加，以獲得正確的答案，稱之為端迴進位（**EAC**）。在數位電路中，由於1的補數減法運算電路較複雜，故較少使用。

圖6-13 是利用四位元並加器與NOT 閘所組成的四位元1的補數並減器。首先利用4個反閘將減數 $B_3B_2B_1B_0$ 取1的補數，即 $\overline{B_3}\overline{B_2}\overline{B_1}\overline{B_0}$ ，再與被減數 $A_3A_2A_1A_0$ 相加，若 C_4 有進位產生時，則需將進位 C_4 與最低有效位元（**LSB**）相加，以獲得正確的答案，即端迴進位。

6-2 加法器及減法器



▲ 圖 6-13 四位元 1 的補數並減器

6-2 加法器及減法器

例題 6-4 四位元1 的補數減法

如圖6-13 所示電路，若 $A_3A_2A_1A_0 = 0110_{(2)}$ ， $B_3B_2B_1B_0 = 0010_{(2)}$ ，則其輸出端總和 S_3 、 S_2 、 S_1 、 S_0 為多少？

解

1. 先將減數 $0010_{(2)}$ 取1 的補數 $= 1101_{(2)}$ 。
2. 再與被減數 $0110_{(2)}$ 相加。
3. 若 C_4 有進位產生時，則需將進位 C_4 與和的最低有效位元（LSB）相加，且其結果為正數。

6-2 加法器及減法器

$$\begin{array}{r}
 0110_{(2)} \\
 - 0010_{(2)} \\
 \hline
 \end{array}
 \Rightarrow \text{取1's}
 \begin{array}{r}
 0110_{(2)} \\
 + 1101_{(2)} \\
 \hline
 \end{array}$$

端迴進位 $\boxed{1}0011_{(2)}$

$+ \quad \quad \quad \rightarrow 1$

$0100_{(2)} \leftarrow \text{解答}$

\uparrow 符號位元 = 0，表示結果為正數

$$\therefore 0110_{(2)} - 0010_{(2)} = 0100_{(2)} = 4_{(10)}$$

所以總和 $S_3S_2S_1S_0 = 0100$ 。

演練 4

如圖6-13 所示電路，若 $A_3A_2A_1A_0 = 0010_{(2)}$ ， $B_3B_2B_1B_0 = 0110_{(2)}$ ，則其輸出端總和 S_3 、 S_2 、 S_1 、 S_0 為多少？

6-2 加法器及減法器

2. 四位元2 的補數並減器

有關2 的補數減法運算可以參考第5 章的5-3 節「二進位減法」之內容。因為兩數相減之差 = 被減數 - 減數

$$= A - B$$

$$= A + (-B)$$

$$= A + (B \text{ 取2 的補數})$$

$$= A + (B \text{ 取1 的補數} + 1)$$

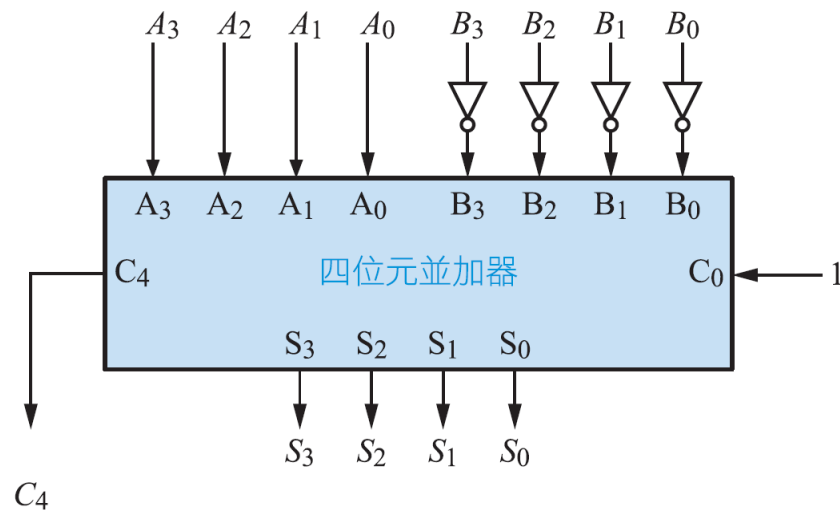
$$= A + (\overline{B} + 1)$$

6-2 加法器及減法器

所以2的補數減法器只要先將減數 B 取1的補數（即 $0 \rightarrow 1$ ， $1 \rightarrow 0$ ）且加1之後，再與被減數 A 相加，即可完成2的補數減法運算，若有進位產生時，則直接捨去進位後其結果即為解答。由於2的補數減法運算電路較簡單，所以目前在數位電路與計算機中，大多採用2的補數減法運算。

圖6-14 是利用四位元並加器7483與NOT閘所組成的四位元2的補數並減器。首先利用4個反閘將減數 $B_3B_2B_1B_0$ 取1的補數，即 $B_3B_2B_1B_0$ ，再與被減數 $A_3A_2A_1A_0$ 及 $C_0 = 1$ 相加，即可獲得兩數相減之差。

6-2 加法器及減法器



▲ 圖 6-14 四位元 2 的補數並減器

6-2 加法器及減法器

例題 6-5 四位元2 的補數減法

如圖6-14 所示電路，若 $A_3A_2A_1A_0 = 0110_{(2)}$ ， $B_3B_2B_1B_0 = 0010_{(2)}$ ，則其輸出端最高位元的進位 C_4 與總和 S_3 、 S_2 、 S_1 、 S_0 為多少？

解

1. 先將減數 $0010_{(2)}$ 取1 的補數 $= 1101_{(2)}$ 。
2. 再與被減數 $0110_{(2)}$ 與1 相加。
3. 若 C_4 有進位產生時，則直接捨去 C_4 後，其結果即為解答，且為正數。

6-2 加法器及減法器

$$\begin{array}{r}
 0110_{(2)} \\
 - 0010_{(2)} \\
 \hline
 \end{array}
 \Rightarrow \text{取1's} \quad
 \begin{array}{r}
 \overset{1}{0}110_{(2)} \\
 + 1101_{(2)} \\
 \hline
 \end{array}$$

捨進位 $\boxed{1}0100_{(2)} \leftarrow$ 解答

↑ 符號位元 = 0，表示結果為正數

$$\therefore 0110_{(2)} - 0010_{(2)} = 0100_{(2)} = 4_{(10)}$$

所以最高位元的進位 $C_4 = 1$ ，而總和 $S_3S_2S_1S_0 = 0100$ 。

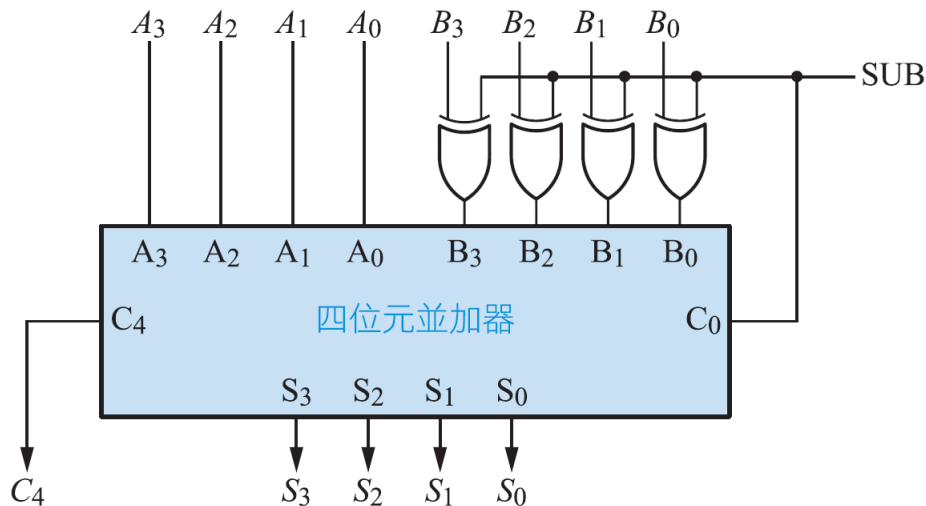
演練 5

如圖6-14 所示電路，若 $A_3A_2A_1A_0 = 0010_{(2)}$ ， $B_3B_2B_1B_0 = 0110_{(2)}$ ，則其輸出端最高位元的進位 C_4 與總和 S_3 、 S_2 、 S_1 、 S_0 為多少？

6-2 加法器及減法器

3. 四位元2 的補數並加 / 減法器

圖6-14 所示之電路，只能當成減法器使用，非常可惜，若將其稍加修改之後，即可當成加 / 減法器使用。圖6-15 是利用四位元並加器7483與XOR 閘組成一個四位元2 的補數並加 / 減法器。



▲ 圖 6-15 四位元 2 的補數並加 / 減法器

6-2 加法器及減法器

其電路原理說明如下：

- (1) 當 **SUB = 0** 時， $C_0 = 0$ ，XOR 閘的輸出端與減數 B 相同，仍為 $B_3B_2B_1B_0$ ，所以執行 $A_3A_2A_1A_0 + B_3B_2B_1B_0$ 之運算，即加法運算 $A + B$ 。
- (2) 當 **SUB = 1** 時， $C_0 = 1$ ，XOR 閘的輸出端與減數 B 反相，即為 $\overline{B_3}\overline{B_2}\overline{B_1}\overline{B_0}$ ，所以執行 $A_3A_2A_1A_0 + \overline{B_3}\overline{B_2}\overline{B_1}\overline{B_0} + 1$ 之運算，相當於 $A + (\overline{B} + 1) = A + (B \text{ 取 } 1 \text{ 的補數} + 1) = A + (B \text{ 取 } 2 \text{ 的補數}) = A + (-B) = A - B$ ，即減法運算 $A - B$ 。

6-2 加法器及減法器

例題 6-6 四位元2 的補數加減法

如圖6-15 所示電路，若 $A_3A_2A_1A_0 = 0010_{(2)}$ ， $B_3B_2B_1B_0 = 0101_{(2)}$ ，則當 $SUB = 0$ 與 $SUB = 1$ 時，其輸出端最高位元的進位 C_4 與總和 S_3 、 S_2 、 S_1 、 S_0 為多少？

解

1. 當 $SUB = 0$ 時，執行加法運算 $A + B$ ，即 $0010_{(2)} + 0101_{(2)}$ 。

$$\begin{array}{r} 0010_{(2)} \\ + 0101_{(2)} \\ \hline 0111_{(2)} \end{array}$$

$$\therefore 0010_{(2)} + 0101_{(2)} = 0111_{(2)} = 7_{(10)}$$

所以最高位元的進位 $C_4 = 0$ ，而總和 $S_3S_2S_1S_0 = 0111$ 。

6-2 加法器及減法器

2. 當 **SUB = 1** 時，執行減法運算 $A + (\bar{B} + 1) = A - B$ ，即 $0010_{(2)} - 0101_{(2)}$ 。

(1) 先將減數 $0101_{(2)}$ 取1的補數 = $1010_{(2)}$

(2) 再與被減數 $0010_{(2)}$ 與1相加

$$\begin{array}{r} 0010_{(2)} \\ - 0101_{(2)} \\ \hline \end{array}$$

⇒ 取1's

符號位元 = 1，表示結果為負數 → $1101_{(2)}$ ← 解答

$$\begin{array}{r} 1 \\ 0010_{(2)} \\ + 1010_{(2)} \\ \hline 1101_{(2)} \\ \downarrow \\ = -0011_{(2)} = -3_{(10)} \end{array}$$

取2's

$$\therefore 0010_{(2)} - 0101_{(2)} = 1101_{(2)} = -3_{(10)}$$

所以最高位元的進位 $C_4 = 0$ ，而總和 $S_3S_2S_1S_0 = 1101$ 。

6-2 加法器及減法器

演練 6

如圖6-15 所示電路，若 $A_3A_2A_1A_0 = 0001_{(2)}$ ， $B_3B_2B_1B_0 = 0011_{(2)}$ ，則當 $SUB = 0$ 與 $SUB = 1$ 時，其輸出端最高位元的進位 C_4 與總和 S_3 、 S_2 、 S_1 、 S_0 為多少？

6-3 二進碼十進數 (BCD) 加法器

有關BCD的說明可以參考第5章的5-4節「二進碼十進數 (BCD)」之內容，BCD是以4位元的二進位碼 (0000 ~ 1001) 來表示十進位的數字 (0 ~ 9)。

一 BCD 加法運算

BCD的加法運算雖然與二進位的加法運算類似，但是因為BCD的數值只有 $0_{(10)}$ 至 $9_{(10)}$ ，所以BCD的加法運算必須適度地做修正，才能得到正確的結果，其修正的步驟敘述如下：

- (一) 將兩個BCD碼，以四個位元為一組進行加法運算。
- (二) 若相加之和大於**9**（即 $10_{(10)} \sim 15_{(10)}$ ）或有進位（ $C_4 = 1$ ，即 $16_{(10)} \sim 19_{(10)}$ ）時，則和必須再加上**6**（即 $0110_{(2)}$ ）進行修正，以得到正確的解答。

6-3 二進碼十進數 (BCD) 加法器

例題 6-7 BCD 加法運算

執行下列BCD 加法運算，其結果為何？

$$1. 6_{(10)} + 8_{(10)} = 0110_{(BCD)} + 1000_{(BCD)}$$

$$2. 9_{(10)} + 8_{(10)} = 1001_{(BCD)} + 1000_{(BCD)}$$

解

十進位加法	BCD 加法
$\begin{array}{r} 6_{(10)} \\ + 8_{(10)} \\ \hline 14_{(10)} \end{array}$	$\begin{array}{r} 0110_{(BCD)} \\ + 1000_{(BCD)} \\ \hline 1110 \text{ (和>9)} \\ + 0110 \text{ (+6)} \\ \hline 0001\ 0100_{(BCD)} \end{array}$

$$\therefore 6_{(10)} + 8_{(10)} = 0110_{(BCD)} + 1000_{(BCD)} = 0001\ 0100_{(BCD)}$$

6-3 二進碼十進數 (BCD) 加法器

2.

十進位加法	BCD 加法
$\begin{array}{r} 9_{(10)} \\ + 8_{(10)} \\ \hline 17_{(10)} \end{array}$	$\begin{array}{r} 1001_{(BCD)} \\ + 1000_{(BCD)} \\ \hline (進位C_4=1) \rightarrow 10001 \\ + 0110 \text{ (+6)} \\ \hline 0001\ 0111_{(BCD)} \end{array}$

$$\therefore 9_{(10)} + 8_{(10)} = 1001_{(BCD)} + 1000_{(BCD)} = 0001\ 0111_{(BCD)}$$

演練 7

執行 $3_{(10)} + 7_{(10)} = 0011_{(BCD)} + 0111_{(BCD)}$ 的BCD 加法運算，其結果為何？

6-3 二進碼十進數 (BCD) 加法器

▼ 表 6-8 兩個 BCD 相加的結果

二 BCD 加法器

因為BCD 的數值只有 $0_{(10)}$ 至 $9_{(10)}$ ，所以兩個最大的BCD相加，加上前一級的進位，絕對不會超過 $19_{(10)}$ ($\because 9 + 9 + 1 = 19$)，表6-8 所示，列出兩個BCD 相加的結果。

十進位數	二進位加法					BCD 加法				
	C_4	S_3	S_2	S_1	S_0	Y_4	Z_3	Z_2	Z_1	Z_0
0		0	0	0	0		0	0	0	0
1		0	0	0	1		0	0	0	1
2		0	0	1	0		0	0	1	0
3		0	0	1	1		0	0	1	1
4		0	1	0	0		0	1	0	0
5		0	1	0	1		0	1	0	1
6		0	1	1	0		0	1	1	0
7		0	1	1	1		0	1	1	1
8		1	0	0	0		1	0	0	0
9		1	0	0	1		1	0	0	1
10		1	0	1	0	1	0	0	0	0
11		1	0	1	1	1	0	0	0	1
12		1	1	0	0	1	0	0	1	0
13		1	1	0	1	1	0	0	1	1
14		1	1	1	0	1	0	1	0	0
15		1	1	1	1	1	0	1	0	1
16	1	0	0	0	0	1	0	1	1	0
17	1	0	0	0	1	1	0	1	1	1
18	1	0	0	1	0	1	1	0	0	0
19	1	0	0	1	1	1	1	0	0	1

$S_3S_1 = 1$

$S_3S_2 = 1$

$C_4 = 1$

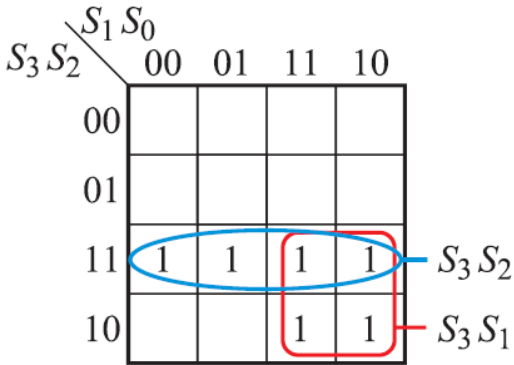
6-3 二進碼十進數 (BCD) 加法器

由表6-8可知，當兩個BCD碼相加時，若和大於9（即 $10_{(10)} \sim 15_{(10)}$ ）或和有進位（ $C_4 = 1$ ，即 $16_{(10)} \sim 19_{(10)}$ ）時，則和必須再加6（即 $0110_{(2)}$ ）進行修正。因此利用兩個四位元並加器配合加6的調整電路，即可組成BCD加法器，其電路設計之方法說明如下：

- (一) 首先利用一個四位元的二進位並加器7483（主加法器）作兩BCD數的相加，其相加的結果為 $C_4S_3S_2S_1S_0$ 。
- (二) 必須設計一個「超9與進位鑑別電路」，當兩數相加之和 $S_3S_2S_1S_0$ 大於9或 C_4 有進位時，則超9與進位鑑別電路之輸出 $Y_4 = 1$ ，如表6-9所示。

6-3 二進碼十進數 (BCD) 加法器

▼ 表 6-9 超 9 與進位鑑別電路

兩數相加	若和 $S_3S_2S_1S_0$ 大於 9 (即 $10_{(10)} \sim 15_{(10)}$)	若 C_4 有進位 (即 $16_{(10)} \sim 19_{(10)}$)
布林代數式	$F(S_3, S_2, S_1, S_0) = \Sigma(10, 11, 12, 13, 14, 15)$	$C_4 = 1$
最簡布林代數式	 <p> $F(S_3, S_2, S_1, S_0)$ $= \Sigma(10, 11, 12, 13, 14, 15) = S_3S_1 + S_3S_2$ </p>	$C_4 = 1$

6-3 二進碼十進數 (BCD) 加法器

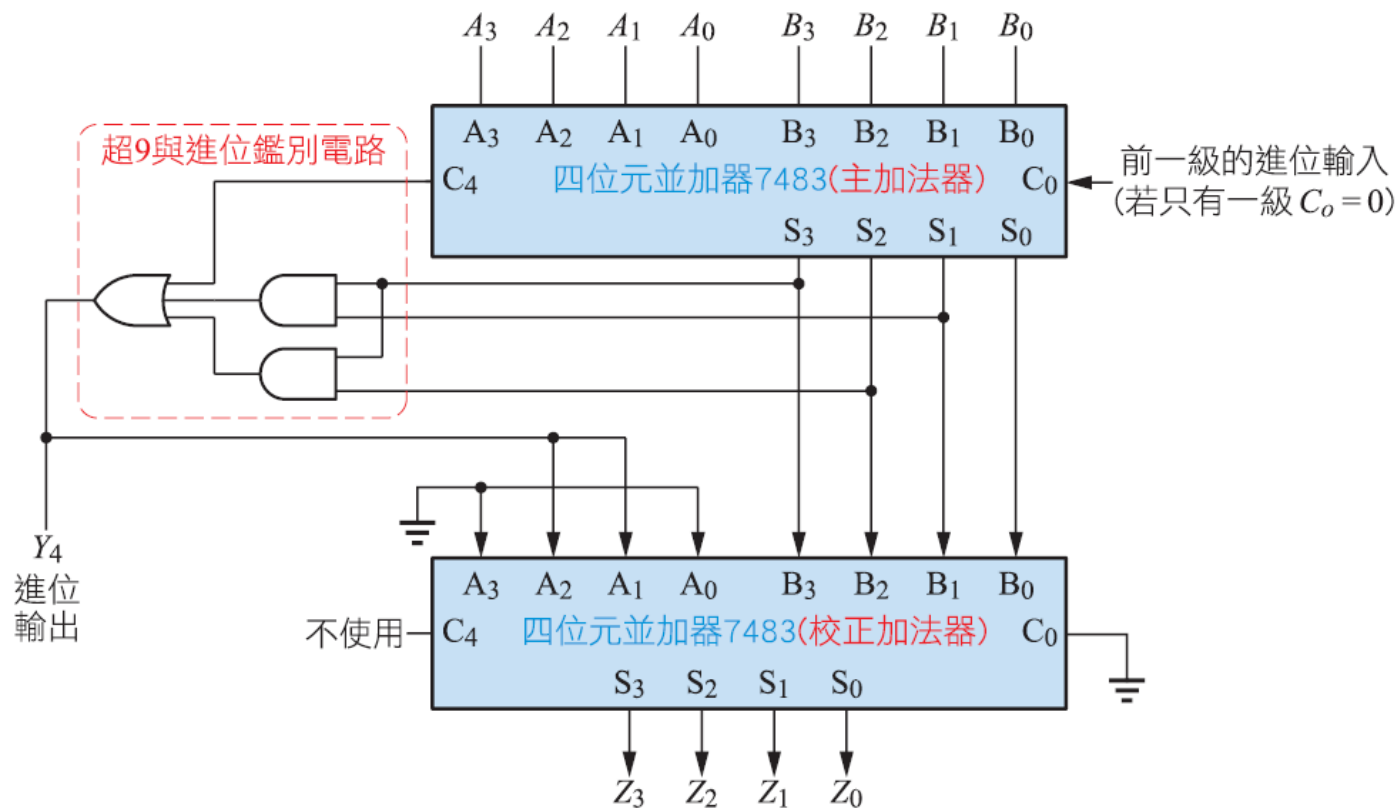
所以「超 9 與進位鑑別電路」之輸出端最簡布林代數式為

$$Y_4 = S_3S_1 + S_3S_2 + C_4$$

6-5 式

(三) 如圖6-16 所示為BCD 加法器，當「超9 與進位鑑別電路」之輸出端 $Y_4 = 1$ ，此時和 $S_3S_2S_1S_0$ 必須經過校正加法器做加6（即 $0110_{(2)}$ ）的修正，所得的結果才是正確的BCD 碼，即進位為 Y_4 ，和為 $Z_3Z_2Z_1Z_0$ 。

6-3 二進碼十進數 (BCD) 加法器



▲ 圖 6-16 BCD 加法器

6-3 二進碼十進數 (BCD) 加法器

例題 6-8 BCD 加法運算

如圖6-16所示之BCD加法器，若 $A_3A_2A_1A_0 = 0101(\text{BCD})$ ， $B_3B_2B_1B_0 = 0111(\text{BCD})$ ，則其輸出端 Y_4 與總和 Z_3 、 Z_2 、 Z_1 、 Z_0 為多少？

解

1. 主加法器：

$$\begin{aligned} \text{主加法器之輸出 } C_4S_3S_2S_1S_0 &= A_3A_2A_1A_0 + B_3B_2B_1B_0 + C_0 \\ &= 0101 + 0111 + 0 \\ &= 01100 \quad (\text{和 } S_3S_2S_1S_0 > 9) \end{aligned}$$

所以「超9與進位鑑別電路」之輸出端 $Y_4 = S_3S_1 + S_3S_2 + C_4 = 1 \cdot 0 + 1 \cdot 1 + 0 = 1$

6-3 二進碼十進數 (BCD) 加法器

2. 校正加法器：

因 $Y_4 = 1$ ，所以校正加法器之 $A_3A_2A_1A_0 = 0110$ ，

而校正加法器之 $B_3B_2B_1B_0 =$ 主加法器之 $S_3S_2S_1S_0 = 1100$ ，

因此校正加法器執行加 6 (即 $0110_{(2)}$) 的修正，

所以校正加法器之輸出 $C_4Z_3Z_2Z_1Z_0 = A_3A_2A_1A_0 + B_3B_2B_1B_0 + C_0$
 $= 0110 + 1100 + 0 = 10010$ ，

其中 $C_4 = 1$ 捨棄不用。

故 BCD 加法器的解答 $Y_4Z_3Z_2Z_1Z_0 = 10010$ 。

$\therefore 5_{(10)} + 7_{(10)} = 0101_{(\text{BCD})} + 0111_{(\text{BCD})} = 1\ 0010_{(\text{BCD})}$ 。

6-3 二進碼十進數 (BCD) 加法器

演練 8

執行 $9_{(10)} + 9_{(10)} = 1001_{(BCD)} + 1001_{(BCD)}$ 的BCD 加法運算，其結果為何？

6-4 解碼器及編碼器

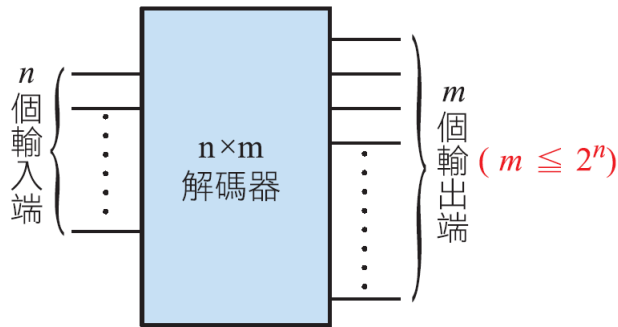
通常數位電路是使用二進位數來處理資料，而人類卻習慣使用十進位數來表示資料，因此解碼器（decoder）正好可以將二進位轉換為十進位，方便人類閱讀；相反的，編碼器（encoder）可以將十進位轉換為二進位，方便數位電路處理，分別說明如下：

一 解碼器

解碼器（**decoder**）可以將二進位轉換為十進位，一般而言， n 位元的二進位數可以表示 2^n 種狀態，所以若解碼器有 n 個輸入端，則其輸出端有 m 個輸出，且 $m=2^n$ ，稱之為 n 對 m 之全解碼器（full decoder），例如二對四解碼器、三對八解碼器等。

6-4 解碼器及編碼器

若解碼器有 n 個輸入端，則其輸出端有 m 個輸出，且 $m < 2^n$ ，稱之為 n 對 m 之部分解碼器（partial decoder），例如BCD對十進位解碼器、BCD對7段顯示器之解碼器等。解碼器的符號如圖6-17所示。



▲ 圖 6-17 n 對 m 解碼器的符號

我們以二對四解碼器、三對八解碼器為例，來說明全解碼器的原理。

6-4 解碼器及編碼器

(一) 二對四解碼器

二對四解碼器可區分為高態動作與低態動作兩種。

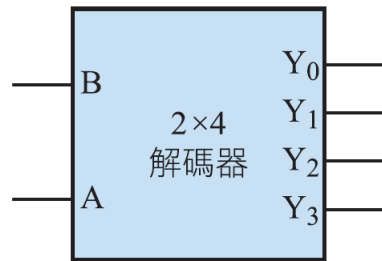
1. 高態動作的二對四解碼器

高態動作的二對四解碼器，有2個輸入端，因為 $2^2=4$ ，所以有4個輸出端，其符號如圖6-18所示，高態動作的二對四解碼器之設計的步驟說明如下：

(1) 決定輸入與輸出變數的個數與名稱：

輸入變數2個： B 、 A ，其中 B 為最高有效位元。

輸出變數4個： Y_0 、 Y_1 、 Y_2 、 Y_3 。



▲ 圖 6-18 二對四解碼器的符號

6-4 解碼器及編碼器

(2) 定義輸入與輸出變數：

無論輸入端 B 、 A 為何種狀態，每次只能有一個輸出端為1，其它皆為0。

(3) 依題意建立真值表，如表6-10 所示。

▼ 表 6-10 二對四解碼器的真值表

輸入		輸出			
(MSB) B	(LSB) A	Y_0	Y_1	Y_2	Y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

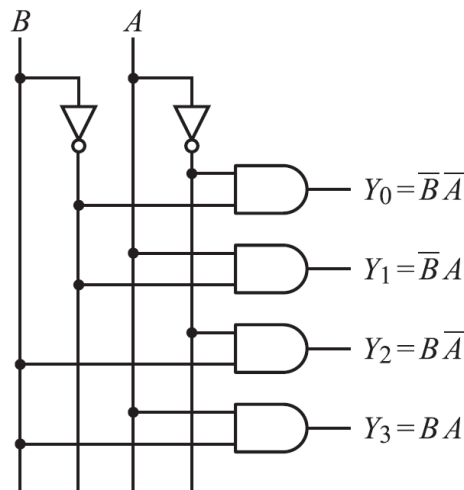
6-4 解碼器及編碼器

(4) 依據表6-10 的真值表，寫出輸出端 $Y_0 \sim Y_3$ 的最簡SOP 布林代數式：

$$Y_0 = \overline{B}\overline{A} \quad Y_1 = \overline{B}A \quad Y_2 = B\overline{A} \quad Y_3 = BA$$

所以高態動作的二對四解碼器有2 個輸入變數，其輸出端可以產生 $2^2 = 4$ 個標準積項（最小項）。

(5) 畫出組合邏輯電路圖：如圖6-19 所示。



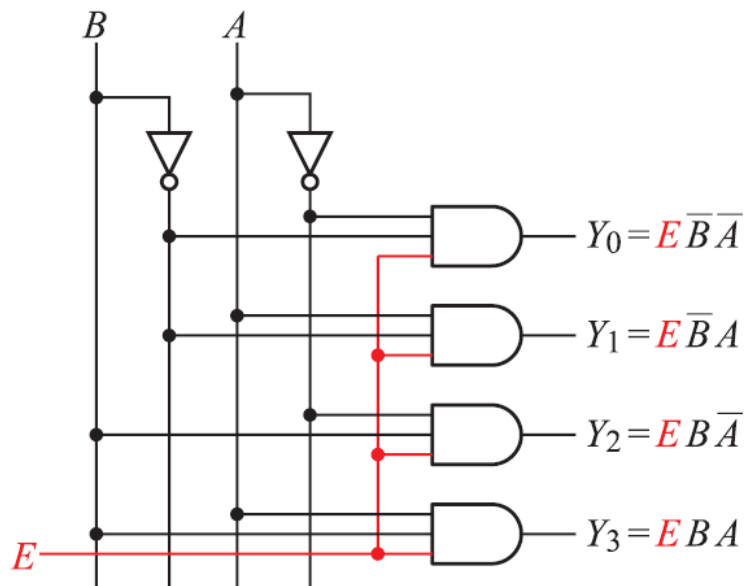
▲ 圖 6-19 二對四解碼器的邏輯電路圖

6-4 解碼器及編碼器

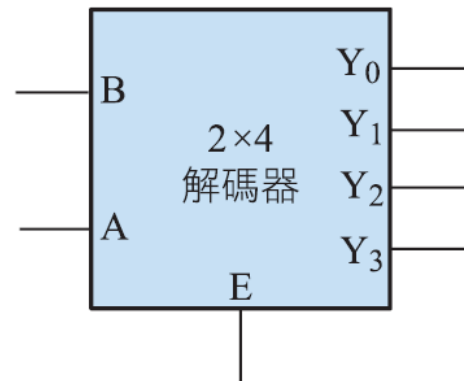
(6) 具有致能（或閃控）控制

在解碼器中，通常包含一個或多個致能（enable；簡稱E）或閃控（strobe；簡稱G）控制輸入端，可以控制解碼器是否接受輸入端的信號。致能控制有高態動作與低態動作兩種類型。如圖6-20 所示為具有高態動作致能控制的二對四解碼器，當輸入致能端 $E = 1$ 時，解碼器能正常解碼，其輸出端 $Y_0 \sim Y_3$ 則視輸入選擇線 B 、 A 而決定何者為1；當輸入致能端 $E = 0$ 時，解碼器不能正常解碼，無論輸入選擇線 B 、 A 為何種狀態，其輸出端 $Y_0 \sim Y_3$ 全為0。通常二對四解碼器IC 的致能控制為低態動作，只要將圖6-20(a) 之輸入致能端 E 接一個反閘即可。

6-4 解碼器及編碼器



(a) 電路圖



(b) 符號

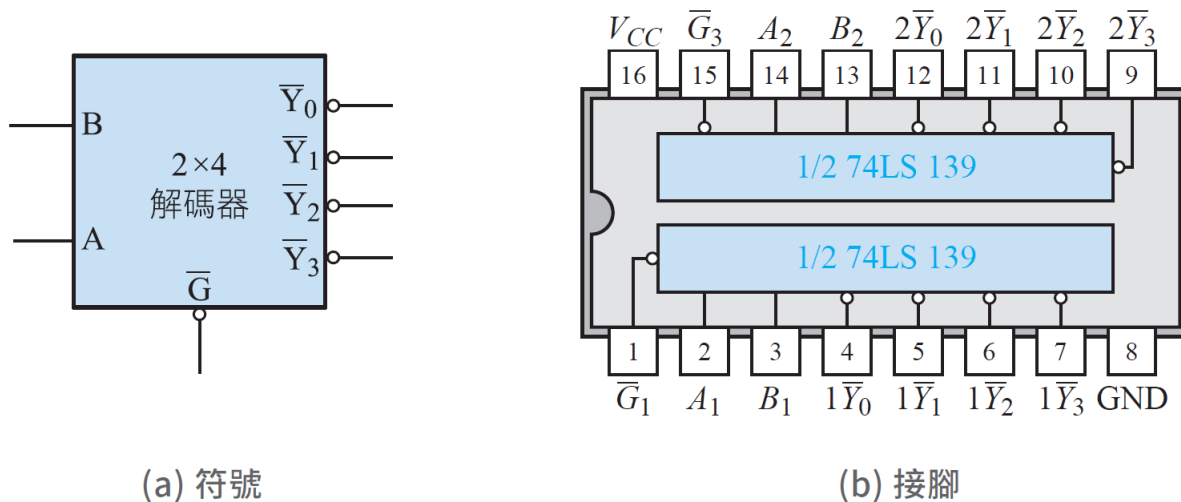
▲ 圖 6-20 具有高態動作致能控制的二對四解碼器

6-4 解碼器及編碼器

2. 低態動作的二對四解碼器（具有致能控制）

一般常用的低態動作之二對四解碼器為74139，它除了具有2個輸入端與4個輸出端之外，還多了一個低態動作的致能控制輸入端 \overline{G} ，故符號與接腳圖中以小圓圈表示低態動作，如圖6-21所示。

6-4 解碼器及編碼器



▲ 圖 6-21 低態動作的二對四解碼器 74139 的符號與接腳

6-4 解碼器及編碼器

74139 的真值表如表6-11 所示，其工作原理說明如下：

- (1) 當輸入致能端 $\overline{G} = 0$ 時，74139 能正常解碼，其輸出端 $\overline{Y}_0 \sim \overline{Y}_3$ 則視輸入選擇線 BA 而決定何者為0（低態動作），而且每次只能有一個輸出端為0，其它輸出端皆為1。
- (2) 當輸入致能端 $\overline{G} = 1$ 時，74139 不能正常解碼，無論輸入選擇線 B 、 A 為何種狀態（以 \times 表示），其輸出端 $\overline{Y}_0 \sim \overline{Y}_3$ 全為1。

6-4 解碼器及編碼器

▼ 表 6-11 低態動作的二對四解碼器 74139 的真值表

輸入			輸出				說明
致能	選擇線		\overline{Y}_0	\overline{Y}_1	\overline{Y}_2	\overline{Y}_3	
\overline{G}	(MSB) B	(LSB) A					
0	0	0	0	1	1	1	正常解碼
	0	1	1	0	1	1	
	1	0	1	1	0	1	
	1	1	1	1	1	0	
1	×	×	1	1	1	1	輸出全為 1

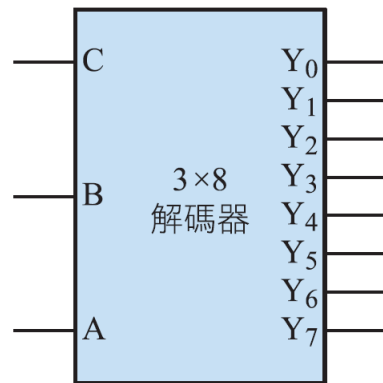
6-4 解碼器及編碼器

(二) 三對八解碼器

三對八解碼器可區分為高態動作與低態動作兩種。

1. 高態動作的三對八解碼器

高態動作的三對八解碼器，有3 個輸入端，因為 $2^3 = 8$ ，所以有8 個輸出端，其符號如圖6-22 所示，高態動作的三對八解碼器之設計的步驟說明如下：



▲ 圖 6-22 三對八解碼器的符號

6-4 解碼器及編碼器

(1) 決定輸入與輸出變數的個數與名稱：

輸入變數3個： C 、 B 、 A ，其中 C 為最高有效位元（MSB）。

輸出變數8個： Y_0 、 Y_1 、 Y_2 、...、 Y_7 。

(2) 定義輸入與輸出變數：

無論輸入端 C 、 B 、 A 為何種狀態，每次只能有一個輸出端為1，其他皆為0。

(3) 依題意建立真值表，如表6-12所示。

6-4 解碼器及編碼器

▼ 表 6-12 高態動作的三對八解碼器之真值表

輸入			輸出							
(MSB) C	B	(LSB) A	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

6-4 解碼器及編碼器

(4) 依據表6-12 的真值表，寫出輸出端 $Y_0 \sim Y_7$ 的最簡SOP 布林代數式：

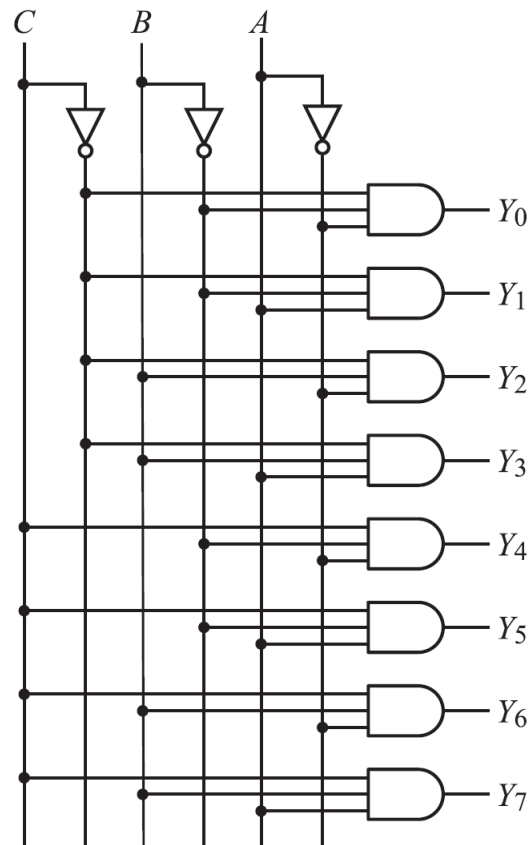
$$Y_0 = \overline{CBA} \quad Y_1 = \overline{CBA} \quad Y_2 = \overline{CBA} \quad Y_3 = \overline{CBA} \quad Y_4 = C\overline{B}\overline{A}$$

$$Y_5 = C\overline{B}A \quad Y_6 = CBA \quad Y_7 = CBA$$

所以高態動作的三對八解碼器有3 個輸入變數，其輸出端可以產生 $2^3 = 8$ 個標準積項（最小項）。

6-4 解碼器及編碼器

(5) 畫出組合邏輯電路圖
：如圖6-23 所示。



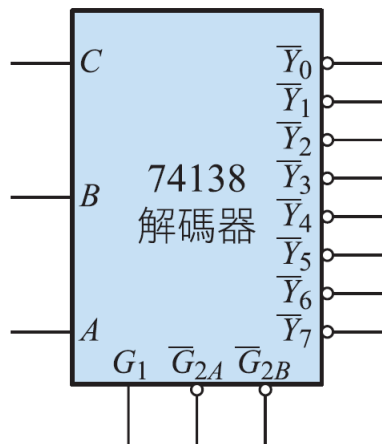
▲ 圖 6-23 三對八解碼器的邏輯電路圖

6-4 解碼器及編碼器

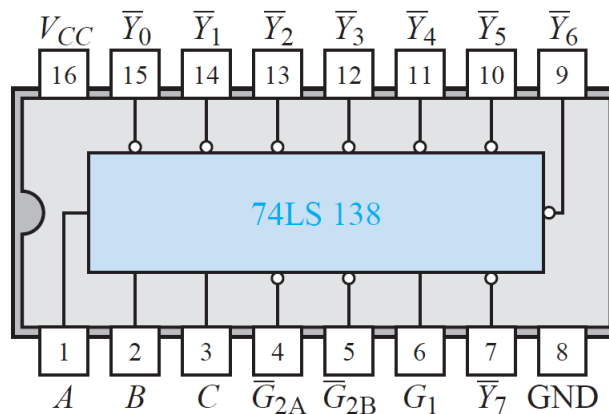
2. 低態動作的三對八解碼器（具有致能控制）

一般常用的低態動作之三對八解碼器為74138，它除了具有3個輸入端與8個輸出端之外，還多了一個高態動作的致能控制輸入端 G_1 ，以及二個低態動作的致能控制輸入端 \overline{G}_{2A} 與 \overline{G}_{2B} ，其符號與接腳如圖 6-24所示。

6-4 解碼器及編碼器



(a) 符號



(b) 接腳

▲ 圖 6-24 低態動作的三對八解碼器 74138 的符號與接腳

6-4 解碼器及編碼器

74138 的真值表如表6-13 所示，其工作原理說明如下：

- (1) 當輸入致能端 $G_1 \equiv 1$ 、 $\overline{G}_{2A} = 0$ 與 $\overline{G}_{2B} = 0$ 時，74138 才能正常解碼，其輸出端 $Y_0 \sim Y_7$ 則視輸入選擇線 CBA 而決定何者為0（低態動作），而且每次只能有一個輸出端為0，其它輸出端皆為1。
- (2) 當輸入致能端 $G_1 = 0$ 或 $\overline{G}_{2A} = 1$ 或 $\overline{G}_{2B} = 1$ 時，74138 皆不能正常解碼，無論輸入選擇線 C 、 B 、 A 為何種狀態（以 x 表示），其輸出端 $\overline{Y}_0 \sim \overline{Y}_7$ 全為1。

6-4 解碼器及編碼器

▼ 表 6-13 低態動作的三對八解碼器 74138 的真值表

輸入						輸出								說明
致能			選擇線			\overline{Y}_0	\overline{Y}_1	\overline{Y}_2	\overline{Y}_3	\overline{Y}_4	\overline{Y}_5	\overline{Y}_6	\overline{Y}_7	
G_1	\overline{G}_{2A}	\overline{G}_{2B}	(MSB) C	B	(LSB) A									
1	0	0	0	0	0	0	1	1	1	1	1	1	1	正常解碼
			0	0	1	1	0	1	1	1	1	1	1	
			0	1	0	1	1	0	1	1	1	1	1	
			0	1	1	1	1	0	1	1	1	1	1	
			1	0	0	1	1	1	0	1	1	1	1	
			1	0	1	1	1	1	1	0	1	1	1	
			1	1	0	1	1	1	1	1	0	1	1	
			1	1	1	1	1	1	1	1	1	1	0	
0	×	×	×	×	×	1	1	1	1	1	1	1	輸出高態	
×	1	×												
×	×	1												

6-4 解碼器及編碼器

例題 6-9 解碼器設計布林函數

試利用：

1. 高態動作的三對八解碼器與邏輯閘
2. 低態動作的三對八解碼器（74138）與邏輯閘

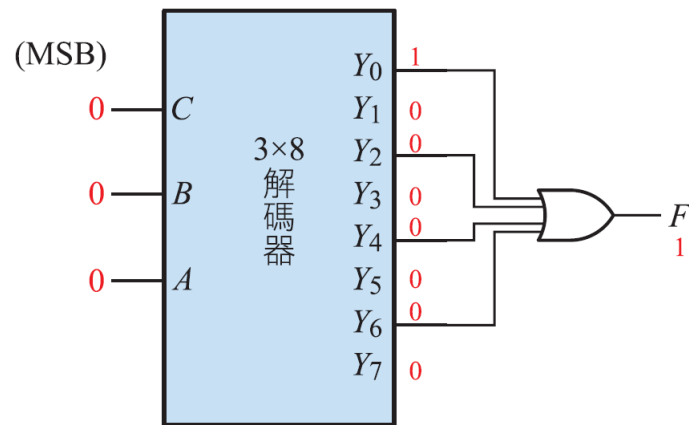
分別設計函數 $F(C,B,A) = \overline{CBA} + \overline{CBA} + \overline{CBA} + \overline{CBA} = \Sigma(0,2,4,6)$ 。

6-4 解碼器及編碼器

解

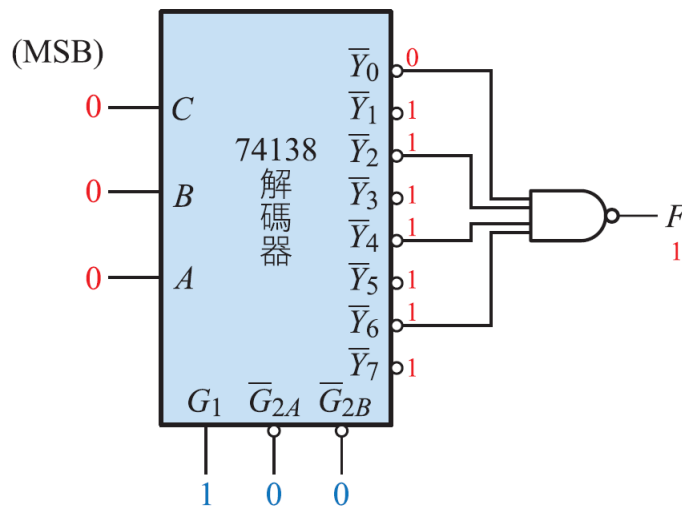
1. 高態動作的三對八解碼器：將解碼器的輸出端 Y_0 、 Y_2 、 Y_4 、 Y_6 以OR 閘連接起來，因為OR 閘只要有一個輸入端為1，其輸出端即為1，所以可得到 $F(C, B, A) = \Sigma(0, 2, 4, 6)$ ，如下圖所示。

如上圖所示，當輸入端 $CBA = 000$ 時，解碼器的輸出端 $Y_0 = 1$ ，其它 $Y_1 \sim Y_7$ 皆為 0，所以OR 閘的輸出端 F 為1，其它以此類推。



6-4 解碼器及編碼器

2. 低態動作的三對八解碼器：將解碼器的輸出端 \bar{Y}_0 、 \bar{Y}_2 、 \bar{Y}_4 、 \bar{Y}_6 以NAND 閘連接起來，因為NAND閘只要有一個輸入端為0，其輸出端即為1，所以可得到 $F(C, B, A) = \Sigma(0, 2, 4, 6)$ ，如下圖所示。



6-4 解碼器及編碼器

如上圖所示，當輸入端 $CBA = 000$ 時，解碼器的輸出端 $\bar{Y}_0 = 0$ ，其它 $\bar{Y}_1 \sim \bar{Y}_7$ 皆為1，所以NAND 閘的輸出端 F 為1，其它以此類推。

因此只要將解碼器搭配適當的邏輯閘，即可以實現布林代數式，以完成組合邏輯電路的設計。

演練 9

試利用：

1. 高態動作的三對八解碼器與邏輯閘
2. 低態動作的三對八解碼器（74138）與邏輯閘

分別設計一個全加器，其和 $S(C_i, B, A) = \Sigma(1, 2, 4, 7)$ ，其進位 $C_o(C_i, B, A) = \Sigma(3, 5, 6, 7)$ 。

6-4 解碼器及編碼器

(三) BCD 對7 段顯示器的解碼器

我們以BCD 對7 段顯示器的解碼器為例，說明部分解碼器的原理。

1. 7 段顯示器

7 段顯示器是使用7 個線段來顯示十進位的數字0 ~ 9，其結構與顯示字型如圖6-25 所示，通常在一般電路中是使用發光二極體（LED）所製成的7 段顯示器，可分為共陽極（common anode）與共陰極（common cathode）兩種。

6-4 解碼器及編碼器

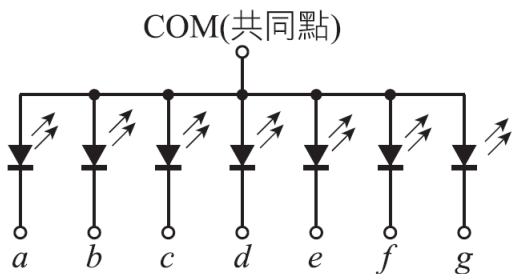


▲ 圖 6-25 7 段顯示器之結構與顯示字型

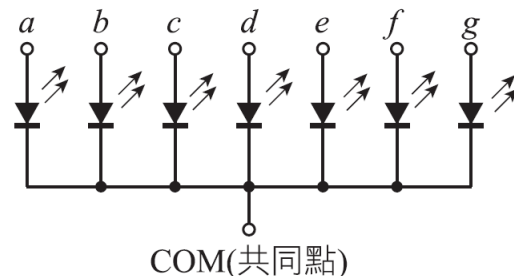
所謂共陽極7段顯示器是將內部所有**LED**的陽極全部都接在一起，成為一個共同點（common），如圖6-26(a)所示，而每個LED的陰極則分別接成 a 、 b 、 c 、 d 、 e 、 f 、 g 等7個端點。當共同點（common）接正電壓 $+V_{CC}$ （相當於1）時，只要將 $a \sim g$ 段中任何一個端點接地（相當於0），則該段LED會發亮（即接0發亮

6-4 解碼器及編碼器

反之，共陰極7段顯示器是將內部所有**LED**的陰極全部都接在一起，成為一個共同點（common），如圖6-26(b)所示，而每個LED的陽極則分別接成 a 、 b 、 c 、 d 、 e 、 f 、 g 等7個端點。當共同點（common）接地（相當於0）時，只要 $a \sim g$ 段中任何一個端點接正電壓 $+V_{CC}$ （相當於1），則該段LED會發亮（即接1發亮）。



(a) 共陽極 7 段顯示器之結構



(b) 共陰極 7 段顯示器之結構

▲ 圖 6-26 7 段顯示器之結構

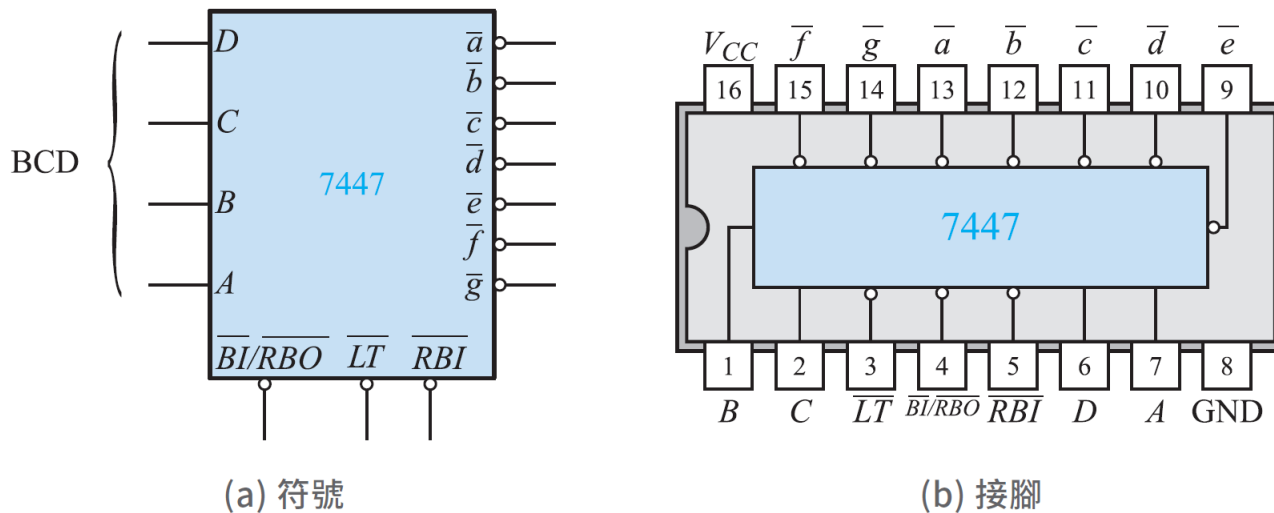
6-4 解碼器及編碼器

2. BCD 對7 段顯示器的解碼器 — 7447

常用的BCD 對7 段顯示器之解碼器有兩種，一種為推動共陽極7段顯示器（低態動作），例如TTL IC 中的**7446**、**7447**、**74246**、**74247**等；另一種為推動共陰極7 段顯示器（高態動作），例如TTL IC 中的**7448**、**7449**、**74248** 等，以及CMOS IC 中的**4511**。

我們以7447 為例，說明BCD 對7 段顯示器的解碼器原理，其符號與接腳如圖6-27 所示。

6-4 解碼器及編碼器



▲ 圖 6-27 7447 符號與接腳

6-4 解碼器及編碼器

如表6-14 所示為7447 解碼器之真值表，由真值表可知，當7447 處於正常的解碼狀態下（即 $\overline{LT}=1$ ， $\overline{RBI}=1$ ， $\overline{BI}/\overline{RBO}=1$ ），若輸入端 $DCBA$ 為0000 時，則輸出端 $abcdefg$ 為0000001，使七段顯示器顯示"0" 字型；若輸入端 $DCBA$ 為0001 時，則輸出端 $abcdefg$ 為1001111，使七段顯示器顯示"1" 字型；其他以此類推，如圖6-28 所示為7447 所有顯示的字型。

6-4 解碼器及編碼器

♥ 表 6-14 7447 的真值表

十進制值與 控制功能	輸入						$\overline{BI} / \overline{RBO}$	輸出							備註
	\overline{LT}	\overline{RBI}	D	C	B	A		\overline{a}	\overline{b}	\overline{c}	\overline{d}	\overline{e}	\overline{f}	\overline{g}	
0	H	H	L	L	L	L	H	L	L	L	L	L	L	H	顯示 0
1	H	×	L	L	L	H	H	H	L	L	H	H	H	H	顯示 1
2	H	×	L	L	H	L	H	L	L	H	L	L	H	L	顯示 2
3	H	×	L	L	H	H	H	L	L	L	L	H	H	L	顯示 3
4	H	×	L	H	L	L	H	H	L	L	H	H	L	L	顯示 4
5	H	×	L	H	L	H	H	L	H	L	L	H	L	L	顯示 5
6	H	×	L	H	H	L	H	H	H	L	L	L	L	L	顯示 6
7	H	×	L	H	H	H	H	L	L	L	H	H	H	H	顯示 7
8	H	×	H	L	L	L	H	L	L	L	L	L	L	L	顯示 8
9	H	×	H	L	L	H	H	L	L	L	H	H	L	L	顯示 9
10	H	×	H	L	H	L	H	H	H	H	L	L	H	L	
11	H	×	H	L	H	H	H	H	H	L	L	H	H	L	
12	H	×	H	H	L	L	H	H	L	H	H	H	L	L	
13	H	×	H	H	L	H	H	L	H	H	L	H	L	L	
14	H	×	H	H	H	L	H	H	H	H	L	L	L	L	
15	H	×	H	H	H	H	H	H	H	H	H	H	H	H	
$\overline{BI} / \overline{RBO}$	×	×	×	×	×	×	L	H	H	H	H	H	H	H	全暗
\overline{LT}	L	×	×	×	×	×	H	L	L	L	L	L	L	L	全亮
\overline{RBI}	H	L	L	L	L	L	L	H	H	H	H	H	H	H	不顯示 0



▲ 圖 6-28 7447 所有顯示的字型

6-4 解碼器及編碼器

在表6-14的真值表中，有 \overline{LT} 、 \overline{RBI} 與 $\overline{BI} / \overline{RBO}$ 三個控制接腳，其中 $\overline{BI} / \overline{RBO}$ 的優先權最高，其次為 \overline{LT} ，而 \overline{RBI} 的優先權最低，其接腳功能說明如下：

- (1) $\overline{BI} / \overline{RBO}$ ：遮沒輸入／漣波遮沒輸出（blanking input／ripple blanking output）控制端，此控制端同時具有輸入與輸出功能。若 $\overline{BI} / \overline{RBO} = 1$ ，則當作輸入端使用，此時7447進入完全遮沒狀態，不管 \overline{LT} 、 \overline{RBI} 或輸入端 $DCBA$ 為何種狀態（以×表示），其輸出端 $\overline{abcdefg}$ 為1111111，使七段顯示器全不亮；反之，當 $\overline{BI} / \overline{RBO} = 0$ 時，則7447正常解碼。

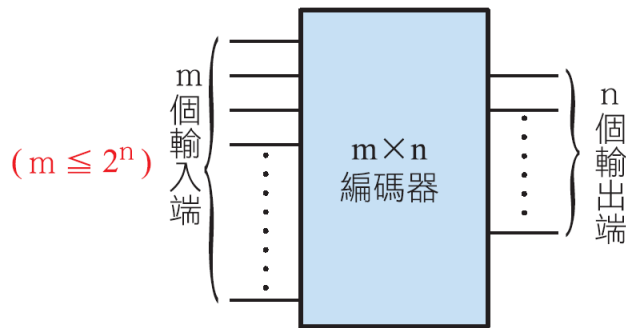
6-4 解碼器及編碼器

- (2) \overline{LT} : 燈泡測試 (lamp test) 輸入端。當 $\overline{BI} / \overline{RB0} = 1$ 且 $\overline{LT} = 0$ 時，不管 \overline{RBI} 或輸入端 $DCBA$ 為何種狀態 (以 x 表示)，7447 處於燈泡測試的情況，其輸出端 $abcdefg$ 為 0000000，使七段顯示器全亮，顯示 "8" 字型；反之，當 $\overline{LT} = 1$ 時，則 7447 正常解碼。
- (3) \overline{RBI} : 漣波遮沒輸入 (ripple blanking input) 控制端，又稱為零遮沒控制端。當 $\overline{LT} = 1$ 、 $\overline{RBI} = 0$ 且輸入端 $DCBA$ 為 0000 時，7447 進入漣波遮沒狀態，其輸出端 $abcdefg$ 為 1111111，使七段顯示器不顯示 "0" 字型，此時 $\overline{BI} / \overline{RB0}$ 可作為零遮沒輸出指示用，所以 $\overline{BI} / \overline{RB0}$ 輸出 0，以傳遞給下一級 7447 使用；反之，當 $\overline{RBI} = 1$ 且輸入端 $DCBA$ 為 0000 時，則輸出端 $abcdefg$ 為 0000001，使七段顯示器顯示 "0" 字型。

6-4 解碼器及編碼器

二 編碼器

編碼器（**encoder**）的功能與上一節所敘述的解碼器恰好相反，它是將人類熟悉的十進位轉換為適合電腦處理的二進位，所以若編碼器有 m 個輸入端，則其輸出端有 n 個輸出，且 $m \leq 2^n$ ，稱之為 m 對 n 編碼器，其符號如圖6-29所示。



▲ 圖 6-29 m 對 n 編碼器的符號

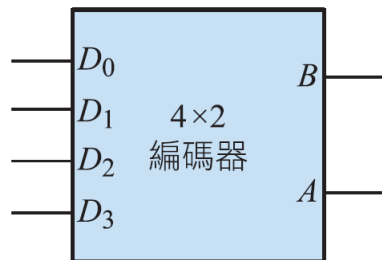
6-4 解碼器及編碼器

編碼器可以區分為基本型編碼器與優先編碼器等兩種類型。

(一) 基本型編碼器－四對二編碼器

我們以四對二編碼器為例，來說明基本型編碼器的原理。

四對二編碼器有4 個輸入端，因為 $4 = 2^2$ ，所以有2 個輸出端，其符號如圖6-30 所示，四對二編碼器之設計的步驟說明如下：



▲ 圖 6-30 四對二編碼器的符號

6-4 解碼器及編碼器

1. 決定輸入與輸出變數的個數與名稱：
 - (1) 輸入變數4 個： D_0 、 D_1 、 D_2 、 D_3 。
 - (2) 輸出變數2 個： B 、 A ，其中 B 為最高有效位元。
2. 定義輸入與輸出變數，每次只能有一個輸入端為1，不能有其他種情況發生。
3. 依題意建立真值表，如表6-15 所示。

▼ 表 6-15 四對二編碼器的真值表

輸入				輸出	
D_0	D_1	D_2	D_3	(MSB) B	(LSB) A
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

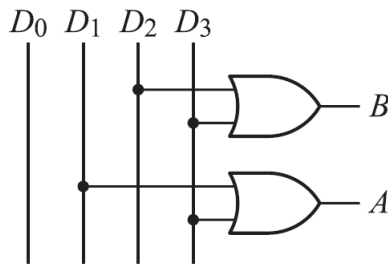
6-4 解碼器及編碼器

4. 依據表6-15 的真值表，寫出輸出端B 與A 的最簡SOP 布林代數式：

$$B = D_2 + D_3$$

$$A = D_1 + D_3$$

5. 畫出組合邏輯電路圖：如圖6-31 所示。



▲ 圖 6-31 四對二編碼器的邏輯電路圖

6-4 解碼器及編碼器

(二) 優先編碼器

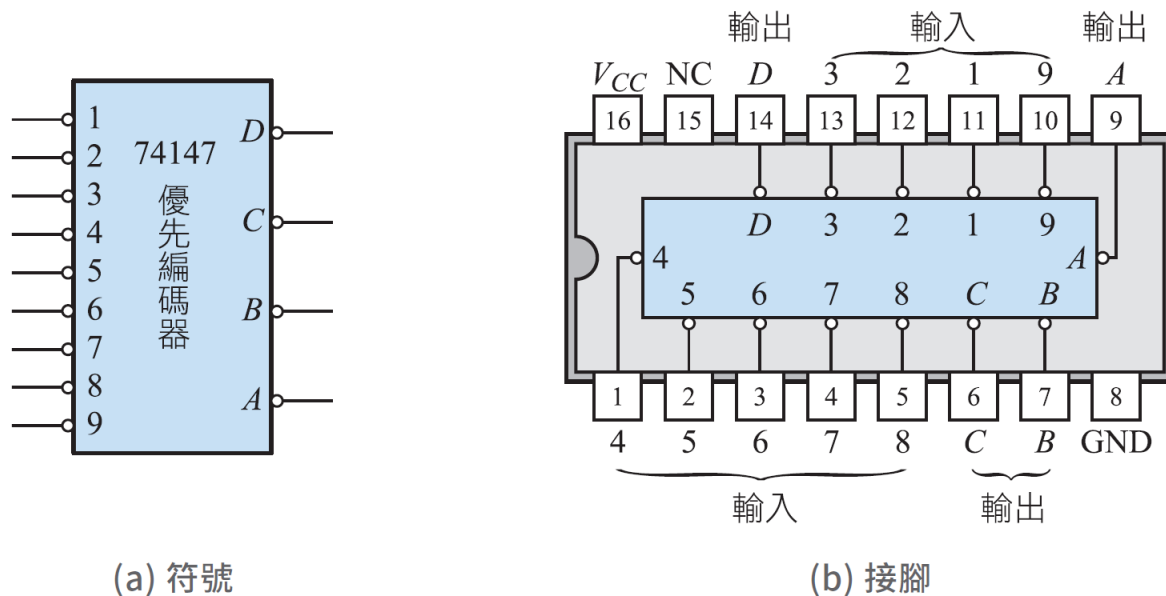
前面所介紹的基本型編碼器是高態動作，每次只能有一個輸入端為1，其輸出端才能得到正確的編碼，如果同時有多個輸入端為1時，則輸出端無法得到正確的編碼。為了解決類似的問題，我們必須定義每個輸入端的優先權，只要同時有多個輸入端為1時，會針對優先權較高的輸入端進行編碼，這就是優先編碼器。

在TTL IC 中，一般常用的優先編碼器有十對四的優先編碼器74147 及八對三的優先編碼器74148，我們以74147 來說明優先編碼器的原理。

1. 優先編碼器（低態動作） - 74147

6-4 解碼器及編碼器

74147 的符號與接腳如圖6-32 所示，其輸入端與輸出端之小圓圈代表低態動作，而74147 的真值表如表6-16 所示。



▲ 圖 6-32 74147 的符號與接腳

6-4 解碼器及編碼器

♥ 表 6-16 74147 的真值表

輸入									輸出			
1	2	3	4	5	6	7	8	9	<i>D</i>	<i>C</i>	<i>B</i>	<i>A</i>
<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>
×	×	×	×	×	×	×	×	<i>L</i>	<i>L</i>	<i>H</i>	<i>H</i>	<i>L</i>
×	×	×	×	×	×	×	<i>L</i>	<i>H</i>	<i>L</i>	<i>H</i>	<i>H</i>	<i>H</i>
×	×	×	×	×	×	<i>L</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>L</i>	<i>L</i>	<i>L</i>
×	×	×	×	×	<i>L</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>L</i>	<i>L</i>	<i>H</i>
×	×	×	×	<i>L</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>L</i>	<i>H</i>	<i>L</i>
×	×	×	<i>L</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>L</i>	<i>H</i>	<i>H</i>
×	×	<i>L</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>L</i>	<i>L</i>
×	<i>L</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>L</i>	<i>H</i>
<i>L</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>	<i>L</i>

6-4 解碼器及編碼器

雖然74147 只有九個輸入端與四個輸出端，但是從真值表中可以看出它有十種輸入的狀態，所以是十對四的優先編碼器，其工作原理說明如下：

- (1) 74147 的輸入為1、2、...、9 號，當提出編碼要求時，其輸入為0（低態動作），所以從真值表中可以發現輸入9 號的優先權最高，而輸入1 號的優先權最低。
- (2) 當全部的輸入1 ~ 9 號皆為1 時，可以視同無任何輸入提出編碼要求，所以輸出 $DCBA$ 皆為高態1111。
- (3) 當輸入9 號提出編碼要求為0 時（低態動作），不管其它輸入為何種狀態（以 x 表示），輸出 $DCBA$ 編碼為0110。

6-4 解碼器及編碼器

- (4) 當輸入9 號為1 且輸入8 號提出編碼要求為0 時（低態動作），不管其它輸入為何種狀態（以x 表示），輸出 $DCBA$ 編碼為0111。
- (5) 其它以此類推。

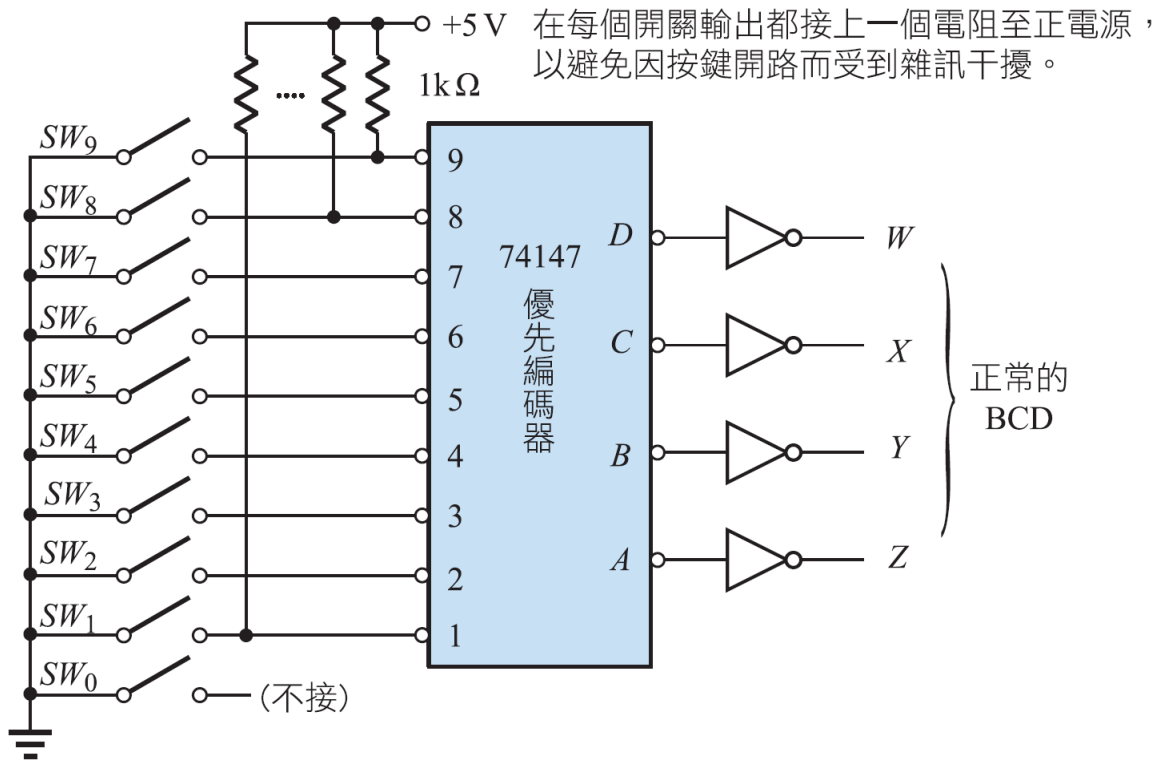
2. 74147 的應用 - 十進位按鍵轉換為 BCD 之優先編碼器

由於74147 的輸出是採補數輸出（低態動作），故欲得到正確的二進位值，必須再加反閘以得到正確的 BCD 碼，如圖6-33 所示為十進位按鍵轉換為 BCD 之優先編碼器。

6-4 解碼器及編碼器

假設開關 SW_7 與 SW_4 同時被按下(ON)時，由於輸入7號的優先權較高，所以編碼器74147的輸出 $DCBA$ 編碼為1000，經過反閘之後的輸出 $WXYZ$ 為 $0111_{(BCD)}$ （即 $7_{(10)}$ ）；假設只有開關 SW_0 被按下(ON)時，全部的輸入1～9號皆為1，所以編碼器74147的輸出 $DCBA$ 編碼為1111，經過反閘之後的輸出 $WXYZ$ 為 $0000_{(BCD)}$ （即 $0_{(10)}$ ）。

6-4 解碼器及編碼器



▲ 圖 6-33 十進位按鍵轉換為 BCD 之優先編碼器

6-4 解碼器及編碼器

例題 6-10 74147 優先編碼器

如圖6-33所示為十進位按鍵轉換為 BCD 之優先編碼器，當開關 SW_5 與 SW_2 同時被按下（ON）時，其輸出 $WXYZ$ 為多少？

解 假設開關 SW_5 與 SW_2 同時被按下（ON）時，由於輸入5號的優先權較高，所以編碼器74147的輸出 $DCBA$ 編碼為1010，經過反閘之後的輸出 $WXYZ$ 為0101_(BCD)（即5₍₁₀₎）。

演練 10

如圖6-33所示為十進位按鍵轉換為 BCD 之優先編碼器，當開關 SW_8 與 SW_3 同時被按下（ON）時，其輸出 $WXYZ$ 為多少？

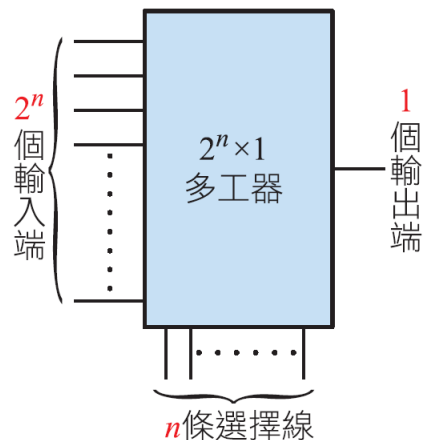
6-5 多工器及解多工器

多工器（multiplexer；簡稱MUX）又稱為資料選擇器（data selector），顧名思義，它能從多個輸入端中，選擇其中一個輸入端信號來輸出；而解多工器（demultiplexer；簡稱DEMUX）的功能與多工器恰好相反，它能將唯一的輸入端信號傳送給多個輸出端中的其中一個，故解多工器又稱為資料分配器（data distributor）。

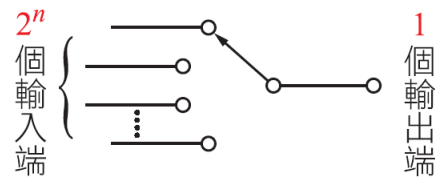
一 多工器

多工器能從多個輸入端中，選擇其中一個輸入端信號來輸出，因此多工器若有 2^n 個輸入端，就必須有 n 條選擇線才能選擇其中一個輸入信號來輸出，例如一個16對1的多工器，因為 $16 = 2^4$ ，所以須有4條選擇線。因此一個多工器具有 2^n 個輸入端、 n 條選擇線及一個輸出端，其符號與等效開關圖如圖6-34所示。

6-5 多工器及解多工器



(a) 符號



(b) 等效開關

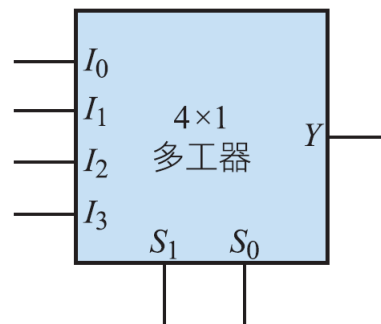
▲ 圖 6-34 多工器的符號與等效開關

6-5 多工器及解多工器

多工器的應用很廣泛，例如家中音響的音源輸入選擇開關，可以選擇雷射唱片、廣播電台或電視機等多個音源信號中的其中一個信號，輸出到放大器。我們以四對一多工器為例，來說明多工器的原理。

(一) 四對一多工器

四對一多工器具有4 個輸入端，因為 $4=2^2$ ，所以需要2 條選擇線及1 個輸出端，其符號如圖6-35 所示，四對一多工器設計的步驟說明如下：



▲ 圖 6-35 四對一多工器的符號

6-5 多工器及解多工器

1. 決定輸入與輸出變數的個數與名稱：

- (1) 輸入變數4 個： I_0 、 I_1 、 I_2 、 I_3 。
- (2) 選擇線2 個： S_1 、 S_0 ，其中 S_1 為最高有效位元。
- (3) 輸出變數1 個： Y 。

2. 定義輸入與輸出變數：

- (1) 當選擇線 $S_1 S_0 = 00$ 時，多工器的輸出端 $Y = I_0$ 。
- (2) 當選擇線 $S_1 S_0 = 01$ 時，多工器的輸出端 $Y = I_1$ ，其他以此類推。

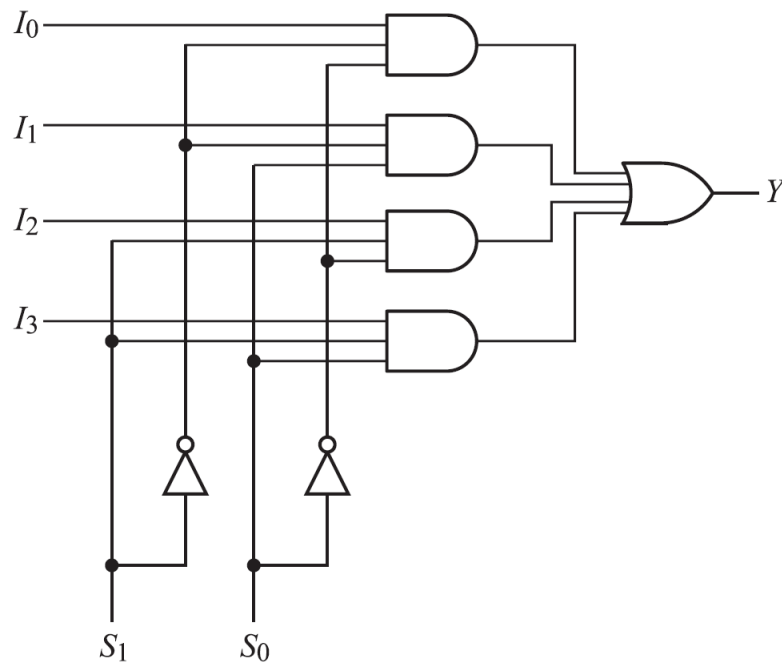
6-5 多工器及解多工器

3. 依題意建立真值表，如表6-17 所示。
4. 依據表6-17 的真值表，寫出輸出端 Y 的最簡SOP 布林代數式：
$$Y = \overline{S_1}\overline{S_0}I_0 + \overline{S_1}S_0I_1 + S_1\overline{S_0}I_2 + S_1S_0I_3$$
5. 畫出組合邏輯電路圖：如圖6-36 所示。

6-5 多工器及解多工器

▼ 表 6-17 四對一多工器的真值表

選擇線		輸出 Y	說明
S_1	S_0		
0	0	I_0	$Y = I_0$
0	1	I_1	$Y = I_1$
1	0	I_2	$Y = I_2$
1	1	I_3	$Y = I_3$



▲ 圖 6-36 四對一多工器的邏輯電路圖

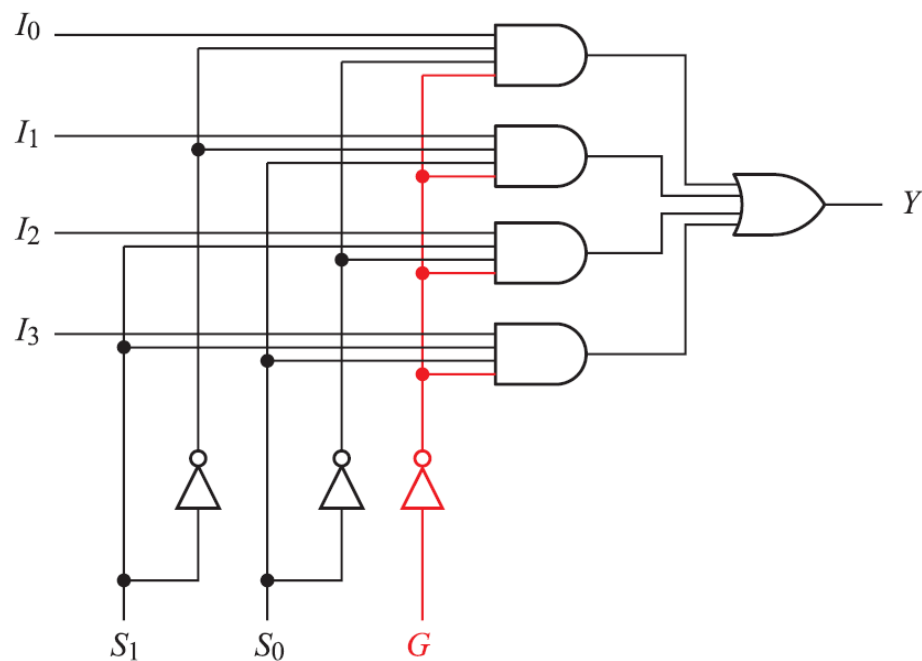
6-5 多工器及解多工器

(二) 具有閃控（或致能）控制的四對一多工器

大部分的多工器與解碼器相同，都具有閃控（strobe）或致能（enable）控制輸入端，以控制多工器是否接受輸入端的信號。閃控控制有高態動作與低態動作兩種類型，如圖6-37 所示為具有低態動作閃控控制的四對一多工器，當輸入閃控端 $G = 0$ 時，多工器才能依輸入選擇線 $S_1 S_0$ 選擇多個輸入端中一個輸入端信號來輸出；反之，當輸入閃控端 $G = 1$ 時，多工器無論輸入選擇線 $S_1 S_0$ 為何種狀態（0 或1），其輸出端 Y 皆為0。其輸出端 Y 的最簡布林代數式為

$$Y = \overline{G}\overline{S_1}\overline{S_0}I_0 + \overline{G}\overline{S_1}S_0I_1 + \overline{G}S_1\overline{S_0}I_2 + \overline{G}S_1S_0I_3$$

6-5 多工器及解多工器



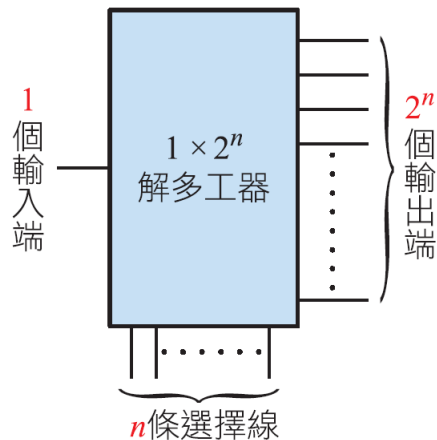
▲ 圖 6-37 具有低態動作閃控控制的四對一多工器

6-5 多工器及解多工器

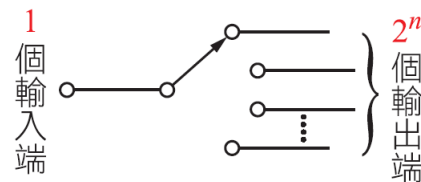
二 解多工器

解多工器能將唯一的輸入端信號傳送給多個輸出端中的其中一個，因此解多工器若有 $2n$ 個輸出端，則必須有 n 條選擇線才能將唯一的輸入信號傳輸給其中一個輸出端，所以解多工器具有一個輸入端、 n 條選擇線及 $2n$ 個輸出端，其符號與等效開關圖如圖 6-38 所示。

6-5 多工器及解多工器



(a) 符號



(b) 等效開關

▲ 圖 6-38 解多工器的符號與等效開關

6-5 多工器及解多工器

我們以一對四解多工器為例，來說明解多工器的原理。

(一) 一對四解多工器

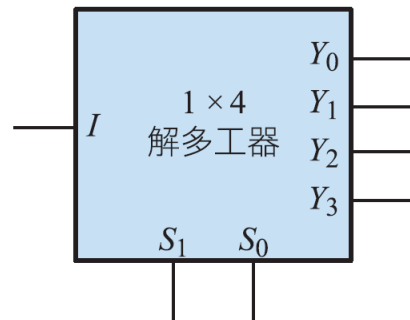
一對四解多工器具有1個輸入端、4個輸出端，因為 $4 = 2^2$ ，所以需要2條選擇線，其符號如圖6-39所示，一對四解多工器設計的步驟說明如下：

1. 決定輸入與輸出變數的個數與名稱：

(1) 輸入變數1個： I 。

(2) 輸入選擇線2條： S_1 、 S_0 ，其中 S_1 為最高有效位元。

(3) 輸出變數4個： Y_0 、 Y_1 、 Y_2 、 Y_3 。



▲ 圖 6-39 一對四解多工器的符號

6-5 多工器及解多工器

2. 定義輸入與輸出變數：

- (1) 當選擇線 $S_1 S_0 = 00$ 時，解多工器的輸出端 $Y_0 = I$ ，其它輸出端皆為0。
 - (2) 當選擇線 $S_1 S_0 = 01$ 時，解多工器的輸出端 $Y_1 = I$ ，其它輸出端皆為0，其他以此類推。
3. 依題意建立真值表，如表6-18 所示。

6-5 多工器及解多工器

▼ 表 6-18 一對四解多工器的真值表

選擇線		輸出				說明
S_1	S_0	Y_0	Y_1	Y_2	Y_3	
0	0	I	0	0	0	$Y_0 = I$
0	1	0	I	0	0	$Y_1 = I$
1	0	0	0	I	0	$Y_2 = I$
1	1	0	0	0	I	$Y_3 = I$

4. 依據表6-18 的真值表，寫出輸出端 $Y_0 \sim Y_3$ 的最簡SOP 布林代數式：

$$Y_0 = \overline{S_1} \overline{S_0} I$$

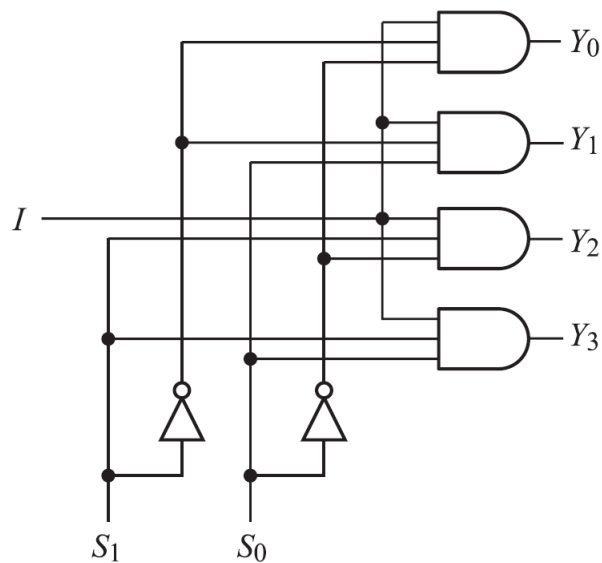
$$Y_1 = \overline{S_1} S_0 I$$

$$Y_2 = S_1 \overline{S_0} I$$

$$Y_3 = S_1 S_0 I$$

6-5 多工器及解多工器

5. 畫出組合邏輯電路圖：如圖6-40 所示。



▲ 圖 6-40 一對四解多工器的電路圖

6-5 多工器及解多工器

(二) 以解碼器完成解多工器

解多工器亦可由具有致能控制的解碼器來完成，如圖6-20所示電路，只要將解碼器的輸入致能端 E 當成解多工器的資料輸入端 I ，將 B 、 A 當成輸入選擇線 S_1S_0 ，而 $Y_0 \sim Y_3$ 為4個輸出端，其電路功能就是一對四解多工器了。

所以低態動作的三對八解碼器 74138 也可以當成一對八解多工器使用，只要將解碼器的輸入致能端 \overline{G}_{2A} 當成解多工器的資料輸入端 I ，且 $G_1 = 1$ 、 $\overline{G}_{2B} = 0$ ，將 C 、 B 、 A 當成輸入選擇線 $S_2S_1S_0$ ，而 $\overline{Y}_0 \sim \overline{Y}_7$ 為8個輸出端，其電路功能就是一對八解多工器了。

6-6 比較器

在日常生活中，我們常將十進位數做數值大小的比較，在數位電路中，我們也需將二進位數做比較，以決定驅動電路是否運作。例如除濕機的濕度感測器偵測到目前的濕度高於設定數值時，除濕機便開始運轉，使濕度下降，當濕度感測器偵測到目前的濕度低於設定數值時，除濕機便停止運轉。此時就需使用比較器（comparator）作為資料大小的比較，所以我們以二進位的一位元比較器、二位元比較器為例，來說明比較器的原理。

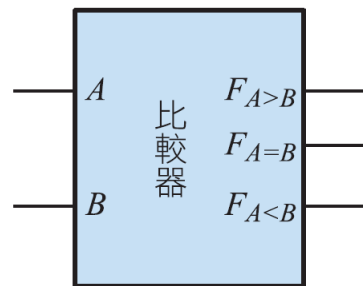
6-6 比較器

一 一位元比較器

一位元比較器有2個輸入端、3個輸出端，其符號如圖6-41所示，一位元比較器設計的步驟說明如下：

(一) 決定輸入與輸出變數的個數與名稱：

1. 輸入變數2個： A 、 B 。
2. 輸出變數3個： $F_{A>B}$ 、 $F_{A=B}$ 、 $F_{A<B}$ 。



▲ 圖 6-41 一位元比較器的符號

6-6 比較器

(二) 定義輸入與輸出變數：

1. 當輸入端 $A > B$ 時，輸出端 $F_{A > B} = 1$ ，其它 $F_{A = B}$ 、 $F_{A < B}$ 皆為0。
2. 當輸入端 $A = B$ 時，輸出端 $F_{A = B} = 1$ ，其它 $F_{A > B}$ 、 $F_{A < B}$ 皆為0。
3. 當輸入端 $A < B$ 時，輸出端 $F_{A < B} = 1$ ，其它 $F_{A > B}$ 、 $F_{A = B}$ 皆為0。

(三) 依題意建立真值表，如表6-19所示。

▼ 表 6-19 一位元比較器的真值表

輸入		輸出		
A	B	$F_{A > B}$	$F_{A = B}$	$F_{A < B}$
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

6-6 比較器

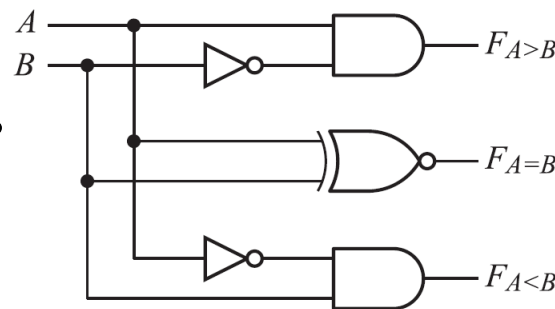
(四) 依據表6-19 的真值表，寫出輸出端 $F_{A>B}$ 、 $F_{A=B}$ 、 $F_{A<B}$ 的最簡 SOP 布林代數式：

$$F_{A>B} = A\bar{B}$$

$$F_{A=B} = \overline{AB} + AB = \overline{A \oplus B} = A \odot B$$

$$F_{A<B} = \bar{A}B$$

(五) 畫出組合邏輯電路圖：如圖6-42 所示。

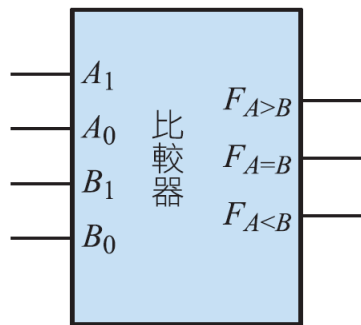


▲ 圖 6-42 一位元比較器的邏輯電路圖

6-6 比較器

二 二位元比較器

二位元比較器與一位元比較器類似，只是由最高有效位元開始，逐次比較各位元之大小。二位元比較器有4 個輸入端、3 個輸出端，其符號如圖6-43所示，二位元比較器設計的步驟說明如下：



▲ 圖 6-43 二位元比較器的符號

6-6 比較器

(一) 決定輸入與輸出變數的個數與名稱：

1. 輸入變數4個： A_1A_0 、 B_1B_0 ，其中 A_1 、 B_1 為最高有效位元。
2. 輸出變數3個： $F_{A>B}$ 、 $F_{A=B}$ 、 $F_{A<B}$ 。

(二) 定義輸入與輸出變數：

1. $A > B$ 的情況有下列二種，其中任何一種皆可：

當輸入端 $A_1 > B_1$ 時，或 $A_1 = B_1$ 且 $A_0 > B_0$ 時，輸出端 $F_{A>B} = 1$ ，其它 $F_{A=B}$ 、 $F_{A<B}$ 皆為0。所以輸出端 $F_{A>B}$ 的布林代數式為

$$F_{A>B} = A_1 \overline{B_1} + (\overline{A_1} \oplus \overline{B_1}) \cdot A_0 \overline{B_0}$$

6-6 比較器

2. $A = B$ 的情況只有一種：

當輸入端 $A_1 = B_1$ 且 $A_0 = B_0$ 時，輸出端 $F_{A=B} = 1$ ，其它 $F_{A>B}$ 、 $F_{A<B}$ 皆為0。所以輸出端 $F_{A=B}$ 的布林代數式為

$$F_{A=B} = \overline{(A_1 \oplus B_1)} \cdot \overline{(A_0 \oplus B_0)}$$

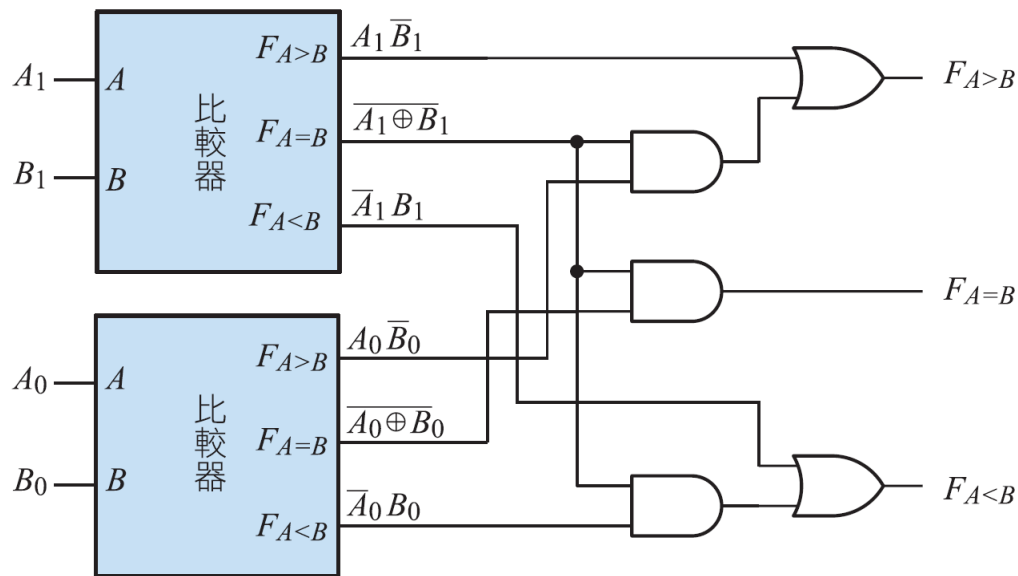
3. $A < B$ 的情況有下列二種，其中任何一種皆可：

當輸入端 $A_1 < B_1$ 時，或 $A_1 = B_1$ 且 $A_0 < B_0$ 時，輸出端 $F_{A<B} = 1$ ，其它 $F_{A>B}$ 、 $F_{A=B}$ 皆為0。所以輸出端 $F_{A<B}$ 的布林代數式為

$$F_{A<B} = A_1 \overline{B_1} + \overline{(A_1 \oplus B_1)} \cdot A_0 \overline{B_0}$$

6-6 比較器

(三) 畫出組合邏輯電路圖：如圖6-44 所示。



▲ 圖 6-44 二位元比較器的邏輯電路圖

6-7 應用實例的認識-多工器的應用

多工器除了具有資料選擇的功能之外，亦可以實現布林代數式完成組合邏輯電路，所以可以減少邏輯閘使用的數量與降低電路的複雜度，達到電路簡化的目的。

6-7 應用實例的認識-多工器的應用

一 多工器電路的分析

一般來說，一個 2^n 對1 的多工器，具有 n 條選擇線，所以可以完成 $n + 1$ 個輸入變數的布林代數式。例如4 對1 的多工器，因為 $4 = 2^2$ ，所以具有2 條資料選擇線，可以表示2 個輸入變數，另外1 個輸入變數接至多工器的輸入端，所以可以完成3 個輸入變數的布林代數式。若要寫出多工器輸出端的布林代數式，其電路分析的步驟如下：

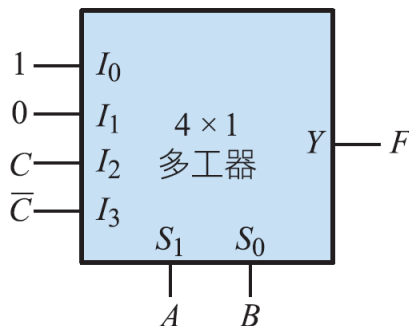
(一) 寫出多工器的真值表。

(二) 依據真值表寫出輸出端的布林代數式。

6-7 應用實例的認識-多工器的應用

例題 6-11 多工器電路的分析

如右圖所示，寫出多工器輸出端 $F(A, B, C)$ 的布林代數式。



6-7 應用實例的認識-多工器的應用

解

1. 寫出多工器的真值表：

選擇線		輸出		積項
S_1	S_0	Y		
A	B			
0	0	I_0	1	$\overline{A}\overline{B} \cdot 1$
0	1	I_1	0	$\overline{A}B \cdot 0$
1	0	I_2	C	$A\overline{B} \cdot C$
1	1	I_3	\overline{C}	$AB \cdot \overline{C}$

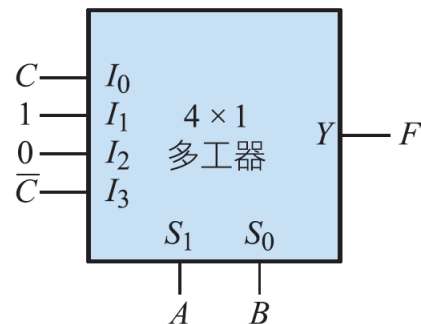
6-7 應用實例的認識-多工器的應用

2. 依據真值表，寫出輸出端 F 的布林代數式為

$$\begin{aligned}
 F(A, B, C) &= \overline{\overline{A}B} \cdot 1 + \overline{\overline{A}B} \cdot 0 + \overline{A}BC + A\overline{B}C \\
 &= \overline{\overline{A}B} + \overline{A}BC + A\overline{B}C \\
 &= \overline{\overline{A}B} + \overline{A}BC + \overline{A}BC + A\overline{B}C \\
 &= \Sigma(0, 1, 5, 6)
 \end{aligned}$$

演練 11

如右圖所示，寫出多工器輸出端 $F(A, B, C)$ 的布林代數式。



6-7 應用實例的認識-多工器的應用

二 以多工器實現布林代數式

若布林代數式有 n 個輸入變數，則以多工器實現布林代數式的方法有下列兩種：

(一) 使用 2^n 對1多工器

布林代數式有 n 個輸入變數，可將 n 個輸入變數直接接至多工器的資料選擇線，所以使用 2^n 對1多工器。例如布林代數式有3個輸入變數，可將3個輸入變數直接接至多工器的資料選擇線，因為 $2^3 = 8$ ，所以使用8對1多工器。舉例說明布林代數式 $F(A, B, C) = \Sigma(1, 2, 5, 7)$ ，使用 2^3 對1多工器完成組合邏輯電路設計步驟。

6-7 應用實例的認識-多工器的應用

1. 寫出布林代數式 $F(A, B, C) = \Sigma(1, 2, 5, 7)$ 之真值表 $F(A, B, C) = \Sigma(1, 2, 5, 7)$ 之真值表，如表6-20 所示。

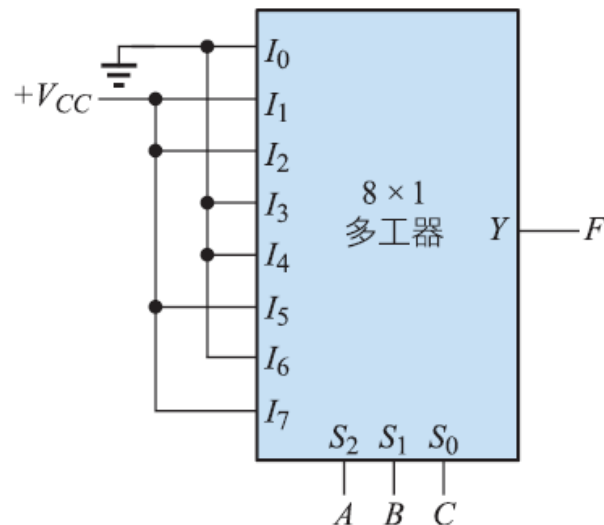
2. 將 n 個輸入變數接至多工器的資料選擇線

將布林代數式的3 個輸入變數 A 、 B 、 C ，依序接至8 對1 多工器的資料選擇線 S_2 、 S_1 、 S_0 ，如圖6-45 所示。

6-7 應用實例的認識-多工器的應用

▼ 表 6-20 $F(A, B, C) = \Sigma(1, 2, 5, 7)$ 之真值表

輸入			輸出
A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



▲ 圖 6-45 多工器組合邏輯電路

6-7 應用實例的認識-多工器的應用

3. 依據真值表，決定多工器輸入端的資料

依據表 6-20 之真值表，決定多工器 $I_0 \sim I_7$ 輸入端的資料為 0（接地）或 1(+ V_{CC})，如圖 6-45 所示。當多工器的選擇線 $S_2S_1S_0 = ABC = 000$ 時，多工器的輸出端 $F = I_0 = 0$ ；當多工器的選擇線 $S_2S_1S_0 = ABC = 001$ 時，多工器的輸出端 $F = I_1 = 1$ ；其它以此類推，如此即可完成布林代數式 $F(A, B, C) = \Sigma(1, 2, 5, 7)$ 之多工器組合邏輯電路。

6-7 應用實例的認識-多工器的應用

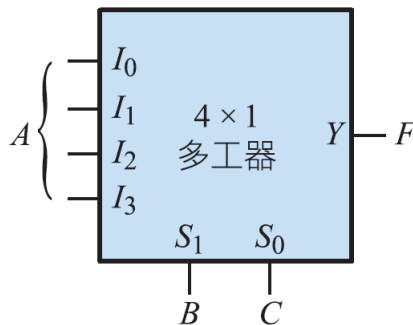
(二) 使用 2^{n-1} 對1 多工器

若要以 2^{n-1} 對1 多工器實現布林代數式，則將其中一個輸入變數當作資料輸入，再將其他輸入變數接至多工器的資料選擇線。舉例說明布林代數式 $F(A, B, C) = \Sigma(1, 2, 5, 7)$ ，使用 2^{3-1} 對1 多工器組合邏輯電路設計步驟。

1. 若布林代數式有 n 個輸入變數，則使用 2^{n-1} 對1 多工器

若布林代數式有 n 個輸入變數，則將其中一個輸入變數當作資料輸入，再將其他的 $n-1$ 個輸入變數接至多工器的資料選擇線，所以使用 2^{n-1} 對1 多工器。例如布林代數式有3 個輸入變數 A 、 B 、 C ，選擇其中一個輸入變數當作資料輸入，再將其他的2 個輸入變數接至多工器的資料選擇線，因為 $2^2 = 4$ ，所以使用4 對1 多工器，如圖6-46 所示。

6-7 應用實例的認識-多工器的應用



▲ 圖 6-46 4 對 1 多工器

2. 任選其中一個輸入變數當作資料輸入，再將其他的 $n - 1$ 個輸入變數接至多工器的資料選擇線

例如布林代數式有 3 個輸入變數 A 、 B 、 C ，選擇其中一個輸入變數 A 當作資料輸入，再將其他的輸入變數 B 、 C 接至多工器的資料選擇線，如圖 6-46 所示。

6-7 應用實例的認識-多工器的應用

3. 寫出執行表，並將布林代數式的最小項所對應的數字圈選起來，以決定輸入端的資料

執行表類似卡諾圖，其差別是變數依序為連續的二進位數值而非格雷碼。例如布林代數式有3 個輸入變數 A 、 B 、 C ，則必須畫出 $2^3 = 8$ 個方格，每一個方格代表一個標準積項(或最小項)，如圖6-47 所示為常用的3 變數執行表（配合圖6-46）。

6-7 應用實例的認識-多工器的應用

如圖6-47所示，若以 A 當多工器的資料輸入，而 BC 接多工器的資料選擇線 S_1S_0 ， BC 兩變數依序為連續的二進位數00、01、10、11，其中0號方格表示輸入變數 ABC 為 $000_{(2)}$ ，即標準積項（或最小項） $ABC = m_0$ ，其它以此類推。首先將布林代數式的最小項所對應的數字1、2、5、7圈選起來，再由下列方法決定輸入端的資料是0、1、 A 或 \bar{A} 。

- (1) 當選擇線 $S_1S_0 = BC = 00$ 時，多工器的輸出端 $F = I_0$ ，因0與4號皆未圈選，故 $I_0 = 0$ 。
- (2) 當選擇線 $S_1S_0 = BC = 01$ 時，多工器的輸出端 $F = I_1$ ，因1與5號皆被圈選，故 $I_1 = 1$ 。

6-7 應用實例的認識-多工器的應用

- (3) 當選擇線 $S_1S_0 = BC = 10$ 時，多工器的輸出端 $F = I_2$ ，因2號被圈選，故 $I_2 = \bar{A}$ 。
- (4) 當選擇線 $S_1S_0 = BC = 11$ 時，多工器的輸出端 $F = I_3$ ，因7號被圈選，故 $I_3 = A$ 。

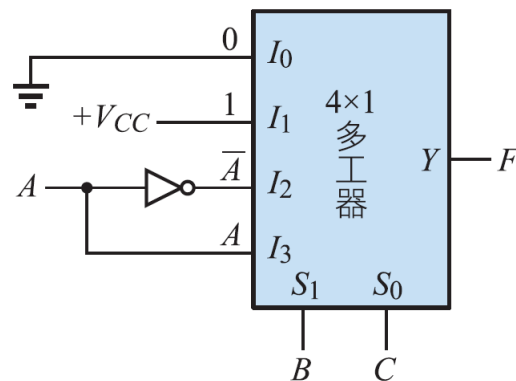
6-7 應用實例的認識-多工器的應用

4. 畫出多工器組合邏輯電路圖，如圖6-48 所示。

資料選擇線
($S_1 S_0$)

		BC			
		00	01	10	11
輸入端 A	0	0	1	2	3
	1	4	5	6	7
		I_0	I_1	I_2	I_3
		0	1	\bar{A}	A

▲ 圖 6-47 執行表



▲ 圖 6-48 多工器組合邏輯電路

6-7 應用實例的認識-多工器的應用

例題 6-12 利用多工器實現布林代數式

利用多工器實現布林代數式 $F(A, B, C) = \Sigma(0, 3, 6, 7)$

解

1. 若布林代數式有3個輸入變數，則選擇線為2條，使用4對1多工器。
2. 選擇其中一個輸入變數A當作資料輸入，再將其他的2個輸入變數BC接至多工器的資料選擇線 S_1S_0 。

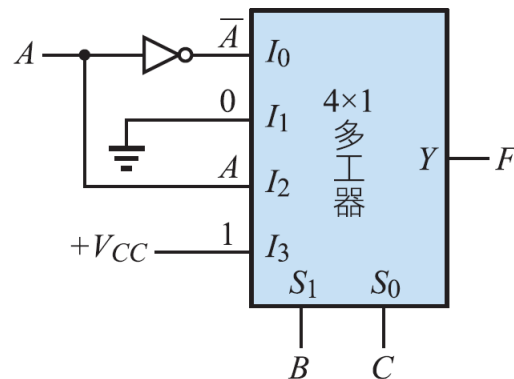
6-7 應用實例的認識-多工器的應用

3. 寫出執行表，並將布林代數式的最小項所對應的數字圈選起來，以決定輸入端的資料。

資料選擇線
($S_1 S_0$)

		BC			
		00	01	10	11
輸入端 A	0	0	1	2	3
	1	4	5	6	7
		\bar{A}	0	A	1

4. 畫出多工器組合邏輯電路圖。



6-7 應用實例的認識-多工器的應用

演練 12

利用多工器實現布林代數式 $F(A, B, C) = \Sigma(0, 3, 6, 7)$ ，以 C 當多工器的資料輸入，其他輸入變數 AB 接至多工器的資料選擇線 S_1S_0 。