



# 布林代數與基本邏輯閘

# 概 述

- ◆ **數學** (Mathematics) 是人類在進行分析與設計工作之最佳輔助工具，而數位系統設計亦須有一套**完整之數學理論**來支持。
- ◆ 英國數學家 **George Boole** 在 1854 年提出一套**數學系統**，來解決分析與設計數位電路之方法，稱為**布林代數** (Boolean Algebra)。
- ◆ 為了分析與設計數位系統，德國數學家 **C. E. Shannon**，在 1938 年發展一套兩值布林代數，又被稱為**交換代數** (Switching Algebra)，並證明**兩值布林代數**，可充分代表數位系統之特性，因此**兩值布林代數**已被發展成為數位系統分析與設計之數學基礎。
- ◆ 本章將首先探討兩值布林代數 (簡稱為**布林函數**) 之**特性**、**基本定理**與**表示法**後，再說明實現數位系統之基本元件--**基本邏輯閘** (Logic Gate) 之**符號**、**邏輯運算式**與**基本特性**與說明使用基本邏輯閘來實現布林函數 (Boolean Function)組成邏輯電路之方法，以構成**完整之數位系統**。最後再以電路之觀點，討論上述基本邏輯閘之**電氣特性**。

# 布林代數之性質

- ◆ 布林代數與一般演繹性之代數系統一樣，它亦可用一組基本元素之集合、一組運算子與許多未證明之定理與假設來定義。

- ◆ 布林代數可用下式來表示

$$B = f(w, x, y, z, \dots)$$

其中  $B$  為輸出變數  $w$ 、 $x$ 、 $y$ 、 $z$  等輸入變數之函數，即  $B$  之值是由  $w$ 、 $x$ 、 $y$ 、 $z$  等變數來決定。

- ◆ 構成布林代數之基本元素，僅被允許為 0 與 1 兩個可能元素之一，且所有數位系統之輸入與輸出變數，亦僅被限制為 0 與 1 兩個可能邏輯值之一，而這兩個邏輯值 (0 與 1) 是分別用來代表數位系統之低電位與高電位的電壓位準，故足以說明布林代數與數位系統間之直接對應關係。

# 布林代數之定義

- ◆ 兩值布林代數之輸入與輸出變數僅允許 **0 與 1 兩個元素**，因組成布林代數為此兩元素，即  $B = \{0,1\}$  集合，與 **AND** ( 積 ;  $\cdot$  )、**OR** ( 和 ;  $+$  ) 與 **NOT** ( 反 ;  $\square$  ) 三個基本**二元邏輯運算子** (Binary Logic Operator) 所組成之代數架構。
- ◆ 布林代數亦應是一套完整之代數系統，故布林代數  $B$  應為 0 與 1 等兩值所組成之集合，即  $B = \{0,1\}$ ，因此對於  $B$  中之所有元素，應滿足下列**六項基本假設**：
  1. **封閉性**(Closure Properties)：對集合  $B$  中的任一個元素而言，經二元運算子運算後所得到之元素，亦應為集合  $B$  之元素。
  2. **單位元素** (Identity Unit)：若集合  $B$  存在一個 0 或 1 之元素，則經邏輯運算後，會滿足  $x \cdot 1 = 1 \cdot x = x$  與  $x + 0 = 0 + x = x$ 。
  3. **交換律** (Commutative Law)：對任意兩個元素  $x, y \in B$  而言，若存在交換律，則應滿足  $x \cdot y = y \cdot x$  與  $x + y = y + x$ 。

4. **分配律** (Distributive Law) : 對任意三個元素  $x, y, z \in B$  而言，若存在分配律，則應滿足  $x \cdot (y + z) = x \cdot y + x \cdot z$  與  $x + (y \cdot z) = (x + y) \cdot (x + z)$  。
5. **反元素** (Inverse Element) : 對集合  $B$  中的任一個元素而言，若存在  $x \in B$ ，亦存在  $\bar{x} \in B$ 。
6. 布林代數  $B$  至少存在兩個不同之元素  $x \cdot y \in B$ ，且  $x \neq y$ ，即兩值布林代數有兩個元素 1 與 0，且  $1 \neq 0$ 。

5. **反元素** (Inverse Element): 對集合  $B$  中的任一個元素而言, 若存在  $x \in B$ , 亦存在  $\bar{x} \in B$ 。

6. 布林代數  $B$  至少存在兩個不同之元素  $x \cdot y \in B$ ，且  $x \neq y$ ，即兩值布林代數有兩個元素 1 與 0，且  $1 \neq 0$ 。

### 證明假說(4)之真值表

[illegible]



# 布林代數之基本定理

- ◆ **定理** (Theorems) 經常是由一套**假設與真實的陳述交叉證明**而得到的，因此根據前面所得之假說，可以整理出許多布林函數之恆等式，而利用這些布林恆等式便可推演出許多**布林定理** (Boolean Theorems)。
- ◆ 接著列出許多常用之布林定理，並分為**單變數**與**多變數**兩個部分來討論如下：

## 1. 單變數布林定理

定 理 名 稱	單變數之 AND 運算	單變數之 OR 運算
(1) 吸收定理	$x \cdot 1 = x$	$x + 0 = x$
(2) 邊界定理	$x \cdot 0 = 0$	$x + 1 = 1$
(3) 全等定理	$x \cdot x = x$	$x + x = x$
(4) 補數定理	$x \cdot \bar{x} = 0$	$x + \bar{x} = 1$
(5) 自補定理	$\bar{\bar{x}} = x$	

## 2. 多變數布林定理

定理名稱	多變數之 AND 運算	多變數之 OR 運算
(1) 交換律	$x \cdot y = y \cdot x$	$x + y = y + x$
(2) 結合律	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$	$x + (y + z) = (x + y) + z$
(3) 分配律	$x \cdot (y + z) = x \cdot y + x \cdot z$	$x + (y \cdot z) = (x + y) \cdot (x + z)$
(4) 消去律	$x \cdot (x + y) = x$	$x + x \cdot y = x$
(5) 狄摩根定律	$\overline{x + y} = \bar{x} \cdot \bar{y}$	$\overline{x \cdot y} = \bar{x} + \bar{y}$

◆ **狄摩根定律** (DeMorgan's Theorem) 之變數，亦可適用於  $n$  個變數之布林代數式，即下面之布林等式亦可成立，即

$$(a) \quad \overline{x + y + \dots + n} = \bar{x} \cdot \bar{y} \cdot \dots \cdot \bar{n}$$

$$(b) \quad \overline{x \cdot y \cdot \dots \cdot n} = \bar{x} + \bar{y} + \dots + \bar{n}$$

# 布林代數之運算優先順序

- ◆ 同一般之代數系統，布林代數運算式之運算子，亦有**運算優先順序** (Priority) 之問題，讀者必須熟記這些運算優先順序，以免在進行**邏輯運算**與**利用邏輯閘來實現布林函數**時，發生錯誤之情況。
- ◆ 處理布林代數之運算式時，運算子之**運算優先順序** (Priority) 為 (1) **括號**，(2) **NOT**，(3) **AND**，(4) **OR**。
- ◆ 換言之，當布林代數之運算式中，**括號之運算優先權為最高**，依次是 NOT、AND，而**運算優先權最低者為 OR**。

# 布林函數之表示法(積項與和項)

- ◆ **函數** (Function) 是由變數、運算子、括號與等號所組成的表示式，而布林函數之邏輯運算子，僅為 NOT、AND 與 OR 等三種，因此當確定變數之數目與名稱後，運用上述三個運算子，便可得到適當之**布林函數** (Boolean Function) 表示式。
- ◆ 布林函數表示式中，變數  $x$  可以用**非補數**  $x$  或**補數**  $\bar{x}$  之形式出現，其中  $x$  與  $\bar{x}$  為**相同之變數**，但兩者是代表**不同之字母變數**。
- ◆ 當兩個或兩個以上之變數，以 **AND 運算子** 所組合成之布林代數式稱為**積項** (Product Terms)，如  $x \cdot y$ 、 $\bar{x} \cdot y$ 、 $x \cdot \bar{y}$  與  $\bar{x} \cdot \bar{y}$  等表示式稱之。
- ◆ 當兩個或兩個以上之變數，以 **OR 運算子** 所組合成之布林代數式稱為**和項** (Sum Terms)，如  $x + y$ 、 $x + \bar{y}$ 、 $\bar{x} + y$  與  $\bar{x} + \bar{y}$  等表示式稱之。

# 最小項與最大項

- ◆ **積項** (Product Terms) 是由  $n$  個不同變數所組成之布林代數表示式，稱為  $n$  個變數之**最小項** (Minterms) 或稱為**標準積** (Standard Product)，亦即**每個積項皆包含到  $n$  個變數**(每個變數僅可以補數或非補數形式出現一次 ) 稱之，由此可知，對  $n$  個變數而言，最多可組成  $2^n$  個最小項。
- ◆ **和項** (Sum Terms) 是由  $n$  個不同變數所組成之布林代數表示式，稱為  $n$  個變數之**最大項** (Maxterms) 或稱為**標準和** (Standard Sum)，即**每個和項皆包含到  $n$  個變數** ( 每個變數亦僅可以補數或非補數形式出現一次 ) 稱之，相同的，對  $n$  個變數而言，最多亦可組成  $2^n$  個最大項。
- ◆ 對  $n$  個變數之布林函數而言，此  $n$  個變數中的每一個二進位數組合，皆定義一個**最小項**與**最大項**；反之，每個最小項與最大項亦定義一個**二進位數字**，且**最大項與最小項互為補數** (Complement)。
- ◆ 為了方便起見，**最小項以小寫之  $m$  來表示**；而**最大項用大寫之  $M$  來表示**，並於英文字母之

下標，分別用其代表二進位數字之等效十進位數作為區分。

- ◆ 接著舉出由 3 個變數 ( $x$ 、 $y$ 、 $z$ ) 所組合成之 8 ( $2^3 = 8$ ) 個最小項與最大項表示法，如下表所示。

十進位數	$x$	$y$	$z$	最 小 項	最 大 項
0	0	0	0	$\bar{x} \cdot \bar{y} \cdot \bar{z} \ (m_0)$	$x + y + z \ (M_0)$
1	0	0	1	$\bar{x} \cdot \bar{y} \cdot z \ (m_1)$	$x + y + \bar{z} \ (M_1)$
2	0	1	0	$\bar{x} \cdot y \cdot \bar{z} \ (m_2)$	$x + \bar{y} + z \ (M_2)$
3	0	1	1	$\bar{x} \cdot y \cdot z \ (m_3)$	$x + \bar{y} + \bar{z} \ (M_3)$
4	1	0	0	$x \cdot \bar{y} \cdot \bar{z} \ (m_4)$	$\bar{x} + y + z \ (M_4)$
5	1	0	1	$x \cdot \bar{y} \cdot z \ (m_5)$	$\bar{x} + y + \bar{z} \ (M_5)$
6	1	1	0	$x \cdot y \cdot \bar{z} \ (m_6)$	$\bar{x} + \bar{y} + z \ (M_6)$
7	1	1	1	$x \cdot y \cdot z \ (m_7)$	$\bar{x} + \bar{y} + \bar{z} \ (M_7)$

- ◆ 二進位數字與最小項之關係：當二進位數字之位元值為 0 時，則其相對應之變數取補數形式；而二進位數字之位元值為 1 時，則其相對應之變數取非補數之形式，最後再將所取得之變數用 AND 組合起來即可。對  $n$  個變數之最小項而言，只在其所對應之二進位數字組合之情況，最小項之值才為 1，而於其它可能組合之情況，最小項之值皆為 0。

- ◆ **二進位數字與最大項之關係**：當二進位數字之位元值為 0 時，則其相對應之變數取**非補數**之形式；而二進位數字之位元值為 1 時，則其相對應之變數取**補數**形式，最後再將所取得之變數用 **OR 組合**起來即可。對  $n$  個變數之最大項而言，只在其**所對應之二進位數字組合之情況**，**最大項之值才為 0**，而於其它可能組合情況之最大項皆為 1。
- ◆ 若以**最小項之和**來表示**布林函數**時，稱為**積項之和** (Sun of Product ; **SOP**)，即在真值表中，**輸出值為 1** 之所有最小項，用 **OR** 連結起來。
- ◆ 若以**最大項之積**來表示**布林函數**時，稱為**和項之積**(Product of Sun ; **POS**)，即在真值表中，**輸出值為 0** 之所有最大項，用 **AND** 連結起來。
- ◆ 對 **SOP 或 POS** 之表示式而言，若所有個別之**積項與和項**，均分別用**最小項與最大項**來表示時，稱為**正規形式** (Canonical Form) 或**標準形式** (Standard Form) 表示法。
- ◆ 標準形式之 SOP 與 POS 表示法之所有**積項與和項**，皆**包含到函數之所有變數**，故往往**不會是最簡單**之布林函數表示法，但在某些數位系統設計之應用上，此種表示法具有其相當好之用處。

## 例題 2-6

試將布林函數  $f(x, y, z) = x \cdot \bar{y} + \bar{x} \cdot z + x \cdot y \cdot \bar{z}$  表示成為積項之和的標準形式。

解

布林函數  $f(x, y, z)$  包含  $x$ 、 $y$  與  $z$  等 3 個變數，因其中有兩個積項未包含 3 個變數，故必須設法將未包含 3 個變數之積項，填滿成為 3 個變數之最小項。而填滿這些積項成為最小項之方法，在此提供兩種方法：

1. 利用代數演繹法來進行，首先保留已經為最小項之積項，而對非最小項之積項，補上所缺少變數的補數與非補數之和以補足所缺少之變數(若缺少變數  $x$ ，則補上  $\bar{x} + x$ )，最後再消去重複之最小項，即為所求。

$$\begin{aligned} f(x, y, z) &= x \cdot \bar{y} + \bar{x} \cdot z + x \cdot y \cdot \bar{z} = (x \cdot \bar{y}) \cdot 1 + (\bar{x} \cdot z) \cdot 1 + x \cdot y \cdot z \\ &= (x \cdot \bar{y}) \cdot (z + \bar{z}) + (\bar{x} \cdot z) \cdot (y + \bar{y}) + x \cdot y \cdot \bar{z} \\ &= x \cdot \bar{y} \cdot z + x \cdot \bar{y} \cdot \bar{z} + \bar{x} \cdot y \cdot z + \bar{x} \cdot \bar{y} \cdot z + x \cdot y \cdot \bar{z} \end{aligned}$$

2. 保留已經為最小項之積項，接著將缺少變數之積項，用對應之二進位數字來表示，並將缺少之變數補上 0 與 1 兩個數( 即代表所缺少變數之補數與非補數形式 )，最後再將二進位數字還原為相對應之最小項，並消去重複之最小項後，即為所求。

(1)、第一個積項  $x \cdot \bar{y}$  缺少  $z$ ，故先將  $x$  定為 1， $y$  定為 0 後，再將  $z$  分別補上 0 與 1。

$x$	$y$	$z$
1	0	0
1	0	1

$$x \cdot \bar{y} = x \cdot \bar{y} \cdot \bar{z} + x \cdot \bar{y} \cdot z$$

(2)、第二個積項  $\bar{x} \cdot z$  缺少  $y$ ，故先將  $x$  定為 0， $z$  定為 1 後，再將  $y$  分別補上 0 與 1。

$x$	$y$	$z$
0	0	1
0	1	1

$$\bar{x} \cdot z = \bar{x} \cdot \bar{y} \cdot z + \bar{x} \cdot y \cdot z$$

(3)、第三個積項  $x \cdot y \cdot \bar{z}$  已是最小項，故不作任何更動。

最後將上面所得之最小項用 OR 連結起來，並消去重複之最小項，即可得

$$f(x, y, z) = x \cdot \bar{y} \cdot \bar{z} + x \cdot \bar{y} \cdot z + \bar{x} \cdot \bar{y} \cdot z + \bar{x} \cdot y \cdot z + x \cdot y \cdot \bar{z}$$

## 例題 2-7

試將布林函數  $f(x, y, z) = (x + \bar{y}) \cdot (\bar{x} + z) \cdot (x + y + \bar{z})$  表示成為和項之積的標準形式。

解

布林函數  $f(x, y, z)$  包含  $x$ 、 $y$ 、 $z$  等 3 個變數，因有兩個和項未包含 3 個變數，故必須設法將未包含 3 個變數之和項，填滿成為 3 個變數之最大項。而填滿這些和項成為最大項之方法，在此亦提供兩種方法：

1. 利用代數演繹法來進行，做法為首先保留已是最大項之和項，而對非最大項之和項，補上所缺少變數之補數與非補數之積，以補足所缺少之變數(若缺少變數  $x$ ，則加上  $\bar{x} \cdot x$ )，最後再消去重複之最大項後，即為所求：

$$\begin{aligned} f(x, y, z) &= (x + \bar{y}) \cdot (\bar{x} + z) \cdot (x + y + \bar{z}) = [(x + \bar{y}) + 0] \cdot [(\bar{x} + z) + 0] \cdot (x + y + \bar{z}) \\ &= [(x + \bar{y}) + \bar{z} \cdot z] \cdot [(\bar{x} + z) + \bar{y} \cdot y] \cdot (x + y + \bar{z}) \\ &= (x + \bar{y} + \bar{z}) \cdot (x + \bar{y} + z) \cdot (\bar{x} + \bar{y} + z) \cdot (\bar{x} + y + z) \cdot (x + y + \bar{z}) \end{aligned}$$

2. 保留已是最大項之和項，接著將缺少變數之和項，用對應之二進位數字來表示，並將缺少之變數補上 0 與 1 兩個數（即代表所缺少變數之補數與非補數形式），最後再將二進位數字還原為相對應之最大項，並消去重複之最大項後，即為所求。

(1)、第一個和項  $x + \bar{y}$  缺少  $z$ ，故先將  $x$  定為 0， $y$  定為 1 後，再將  $z$  分別補上 0 與 1。

$x$	$y$	$z$
0	1	0
0	1	1

$$x + \bar{y} = (x + \bar{y} + \bar{z}) \cdot (x + \bar{y} + z)$$

(2)、第二個和項  $\bar{x} + z$  缺少  $y$ ，故先將  $x$  定為 1， $z$  定為 0 後，再將  $y$  分別補上 0 與 1。

$x$	$y$	$z$
1	0	0
1	1	0

$$\bar{x} + z = (\bar{x} + \bar{y} + z) \cdot (\bar{x} + y + z)$$

(3)、第三個和項  $(x + y + \bar{z})$  已是最大項，故不作任何更動。

最後將上面所得之最大項用 AND 連結起來，並消去重複之最大項，即可得

$$f(x, y, z) = (x + y + \bar{z}) \cdot (x + \bar{y} + z) \cdot (x + \bar{y} + \bar{z}) \cdot (\bar{x} + y + z) \cdot (\bar{x} + \bar{y} + z)$$

## 布林函數之表示法(積項之和)

- ◆ 根據例題 2-6 所得之布林函數，可得 3 種常用積項之和(SOP)的布林函數式表示法，如下所示：

$$\begin{aligned}f(x, y, z) &= \bar{x} \cdot \bar{y} \cdot z + \bar{x} \cdot y \cdot z + x \cdot \bar{y} \cdot \bar{z} + x \cdot \bar{y} \cdot z + x \cdot y \cdot \bar{z} \\&= m_1 + m_3 + m_4 + m_5 + m_6 \\&= \sum(1, 3, 4, 5, 6)\end{aligned}$$

- ◆ 除利用上面 3 種布林函數之 SOP 表示法，亦可使用第 4 種表示法，即將上述之布林函數式，以真值表方式來表示，如下所示：

$x$	$y$	$z$	$f(x, y, z)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

- ◆ 以上為布林函數的 4 種積項之和 (SOP) 的表示方法。

## 布林函數之表示法(和項之積)

- ◆ 根據例題 2-7 所得之布林函數，可得 3 種常用和項之積(POS)的布林函數式表示法，如下所示：

$$\begin{aligned}f(x, y, z) &= (x + y + \bar{z}) \cdot (x + \bar{y} + z) \cdot (x + \bar{y} + \bar{z}) \cdot (\bar{x} + y + z) \cdot (\bar{x} + \bar{y} + z) \\&= M_1 \cdot M_2 \cdot M_3 \cdot M_4 \cdot M_6 \\&= \prod (1, 2, 3, 4, 6)\end{aligned}$$

- ◆ 除利用上面 3 種布林函數之 POS 表示法，亦可使用第 4 種表示法，即將上述之布林函數式，以真值表方式來表示，如下所示：

$x$	$y$	$z$	$f(x, y, z)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

- ◆ 以上為布林函數的 4 種和項之積(POS) 的表示方法。

# 標準積項之和(SOP)與標準和項之積(POS)的互換

- ◆ 積項之和(SOP)與和項之積(POS)兩種形式，皆可分別用來表示所有布林函數，作為設計數位系統之工具。雖然這兩種表示法，在形式上有所差異，但彼此間亦可相互轉換。
- ◆ 若能使用適當的表示法，來設計數位系統，則可適度降低硬體電路之複雜度，如此便可提高所設計的數位系統之經濟效益。

## 例題 2-8

試將標準積項之和  $f(x, y, z) = \bar{x} \cdot \bar{y} \cdot z + \bar{x} \cdot y \cdot z + x \cdot \bar{y} \cdot \bar{z} + x \cdot \bar{y} \cdot z + x \cdot y \cdot \bar{z}$  之布林函數表示式，轉換成為標準和項之積表示式。

解

若取出布林函數值  $f(x, y, z)$  為 0 之部分，即可得原函數之補數  $\overline{f(x, y, z)} = \bar{x} \cdot \bar{y} \cdot \bar{z} + \bar{x} \cdot y \cdot \bar{z} + x \cdot y \cdot z$ 。接著利用狄摩根定律，以求取原函數之補數  $\overline{f(x, y, z)}$  的補數  $\overline{\overline{f(x, y, z)}}$  如下

$$\begin{aligned}\overline{\overline{f(x, y, z)}} &= \overline{f(x, y, z)} = \overline{\bar{x} \cdot \bar{y} \cdot \bar{z} + \bar{x} \cdot y \cdot \bar{z} + x \cdot y \cdot z} = \overline{(\bar{x} \cdot \bar{y} \cdot \bar{z}) \cdot (\bar{x} \cdot y \cdot \bar{z}) \cdot (x \cdot y \cdot z)} \\ &= (\bar{\bar{x}} + \bar{\bar{y}} + \bar{\bar{z}}) \cdot (\bar{\bar{x}} + \bar{y} + \bar{\bar{z}}) \cdot (\bar{x} + \bar{y} + \bar{z}) \\ &= (x + y + z) \cdot (x + \bar{y} + z) \cdot (\bar{x} + \bar{y} + \bar{z}) \\ &= M_0 \cdot M_2 \cdot M_7 = \prod(0, 2, 7)\end{aligned}$$

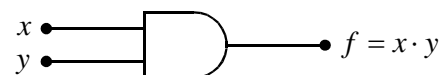
故原函數  $f(x, y, z) = \bar{x} \cdot \bar{y} \cdot z + \bar{x} \cdot y \cdot z + x \cdot \bar{y} \cdot \bar{z} + x \cdot \bar{y} \cdot z + x \cdot y \cdot \bar{z} = \sum(1, 3, 4, 5, 6) = \prod(0, 2, 7)$ 。

欲將標準積項之和 (SOP) 轉換成標準和項之積 (POS) 之方法，只須將  $\Sigma$  與  $\Pi$  互換，並將括號內之等效十進位數字換成原來缺少之等效十進位數後，即為所求（反之，欲將標準和項之積 (POS) 轉換成為標準積項之和 (SOP) 的方法，亦可採用相同之方法），而等效十進數字之範圍是從 0 至  $2^n - 1$ ，其中  $n$  為所求布林函數之變數數目。

# 基本邏輯閘

- ◆ 組成數位電路 (Digital Circuit) 之基本元件為**邏輯閘** (Logic Gate)，也就是說**邏輯閘可用來實現任何布林函數** (Boolean Function)，以構成一個**完整之數位系統** (Digital System)。
- ◆ 布林函數可以用 **AND、OR 與 NOT 三個運算子**完全表示出來，故此**三個邏輯閘亦可用來實現任何布林函數** (Boolean Function)。
- ◆ 在實際數位電路設計上，亦必須**考慮實體元件(邏輯閘)之實現技術**與電路設計常會用到之邏輯運算等問題，故除了上述三種基本邏輯閘外，亦有幾種由 **AND、OR 與 NOT 三個邏輯閘**所合成之常用邏輯閘，如 **NAND、NOR、XOR 與 XNOR** 等四種常用之合成邏輯閘，以簡化繪製與製作邏輯電路之複雜度。
- ◆ 接著將這**七種常用邏輯閘之邏輯符號、布林函數表示式、邏輯運算式與真值表**分別說明於下：

## 1. AND ( 及 ) 閘

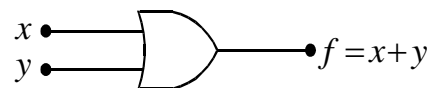


$x$	$y$	$f$
0	0	0
0	1	0
1	0	0
1	1	1

邏輯符號與布林函數式

真 值 表

## 2. OR ( 或 ) 閘

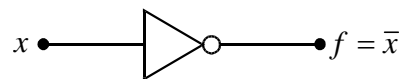


$x$	$y$	$f$
0	0	0
0	1	1
1	0	1
1	1	1

邏輯符號與布林函數式

真 值 表

### 3. NOT ( 反 ) 閘

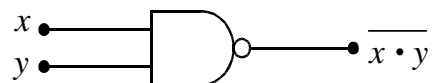


邏輯符號與布林函數式

$x$	$f$
0	1
1	0

真 值 表

### 4. NAND ( 反及 ) 閘

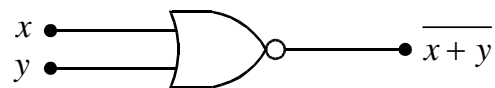


邏輯符號與布林函數式

$x$	$y$	$f$
0	0	1
0	1	1
1	0	1
1	1	0

真 值 表

## 5. NOR ( 反或 ) 閘

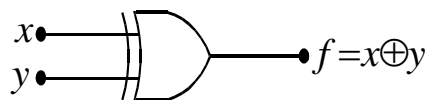


$x$	$y$	$f$
0	0	1
0	1	0
1	0	0
1	1	0

邏輯符號與布林函數式

真 值 表

## 6. XOR ( 互斥或 ) 閘

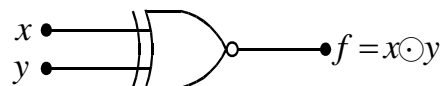


$x$	$y$	$f$
0	0	0
0	1	1
1	0	1
1	1	0

邏輯符號與布林函數式

真 值 表

## 7. XNOR ( 反互斥或 ) 閘



$x$	$y$	$f$
0	0	1
0	1	0
1	0	0
1	1	1

邏輯符號與布林函數式

真 值 表

註：上述真值表中之邏輯 0 與邏輯 1 兩值，皆分別代表一個物理量，而定義這些物理量之方法，可分為正邏輯系統 (Positive Logic System) 與負邏輯系統 (Negative Logic System) 兩種。對正邏輯系統而言，以低電位代表邏輯 0，高電位代表邏輯 1；而負邏輯系統正好與正邏輯系統相反。不論使用正邏輯系統或負邏輯系統來設計數位電路，所得之結果完全相同，但於設計電路時，兩系統切不可混合使用。為避免混淆起見，除非特別聲明，本書僅考慮使用正邏輯系統來設計數位電路。

# 布林函數之邏輯閘網路的實現

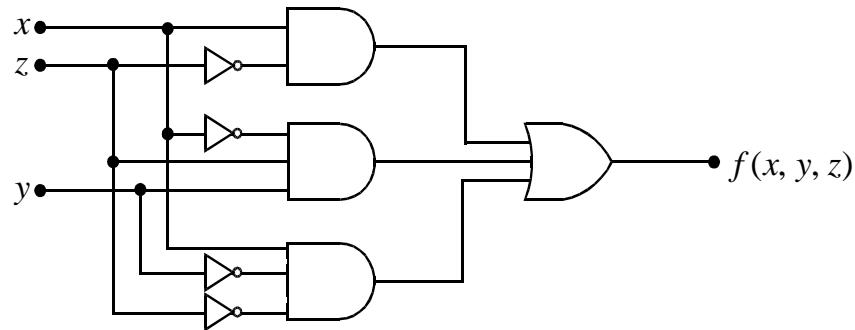
- ◆ 布林函數之運算式，可以完全利用 AND、OR 與 NOT 三個邏輯運算子，以組成積項之和 (SOP) 或和項之積 (POS) 的表示式，因此 AND 閘、OR 閘與 NOT 閘等三個基本邏輯閘 (Logic Gate)，便可用來實現任何布林函數之邏輯閘網路( 亦可稱為邏輯電路；Logic Circuits)，以構成完整之數位系統。
- ◆ 利用邏輯閘來實現布林函數時，首先要考慮為運算優先權 (Priority) 之問題，即運算優先權愈高者，應放置於較接近邏輯閘網路之輸入端，而運算優先權愈低者，則應放置愈靠近邏輯閘網路之輸出端。

## 例題 2-9

試利用邏輯閘網路來實現  $f(x, y, z) = x \cdot \bar{z} + \bar{x} \cdot y \cdot z + x \cdot \bar{y} \cdot \bar{z}$  之布林函數表示式。

解

因反運算 (NOT) 之運算優先權最高，而積項 (AND) 之運算優先權比和項 (OR) 高，故應先對 NOT 進行運算後，接著為 AND 運算，最後再進行 OR 運算。故繪成邏輯閘網路時，NOT 閘應最接近輸入端，接著為 AND 閘，最後才是 OR 閘，如下圖所示。

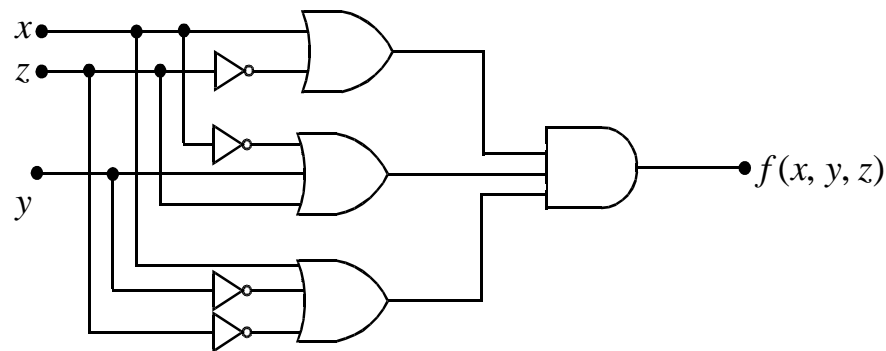


## 例題 2-10

試利用邏輯閘網路來實現  $f(x, y, z) = (x + \bar{z}) \cdot (\bar{x} + y + z) \cdot (x + \bar{y} + \bar{z})$  之布林函數表示式。

解

因括號之運算優先權最高，故應先進行 NOT 運算後，接著進行 OR 運算，最後再進行 AND 運算。故繪成邏輯閘網路時，NOT 閘最接近輸入端，接著為 OR 閘，最後才是 AND 閘，即可得到  $f(x, y, z)$  之邏輯閘網路，如下圖所示。

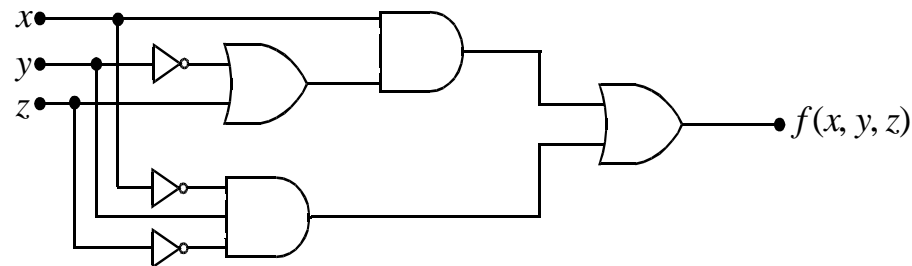


## 例題 2-11

試利用邏輯閘網路來實現  $f(x, y, z) = x \cdot (\bar{y} + z) + \bar{x} \cdot y \cdot \bar{z}$  之布林函數表示式。

解

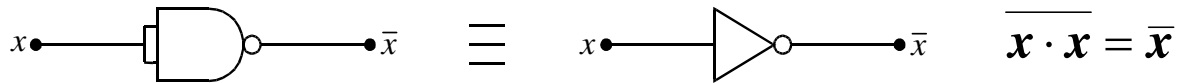
因括號之運算優先權最高，故應先進行 NOT 運算後，接著進行 OR 運算，最後再進行 AND 運算。  
故繪成邏輯閘網路時，NOT 閘最接近輸入端，接著為 OR 閘，最後才是 AND 閘，即可得到  $f(x, y, z)$  之邏輯閘網路，如下圖所示。



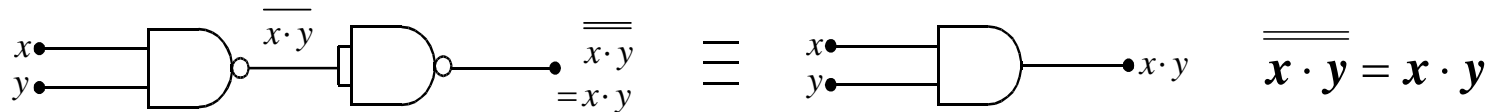
## 邏輯閘之取代(NAND 閘)

- ◆ 因製作 NAND 閘和 NOR 閘的硬體電路比 AND 閘、OR 閘較為簡單，所以利用一些布林代數運算式之轉換（如狄摩根定理）後，便可使用 NAND 閘或 NOR 閘來取代 AND 閘、OR 閘與 NOT 閘之功能，來實現任何布林函數表示式，以設計完整之數位系統。
- ◆ 使用 NAND 取代 NOT 閘、AND 閘與 OR 閘

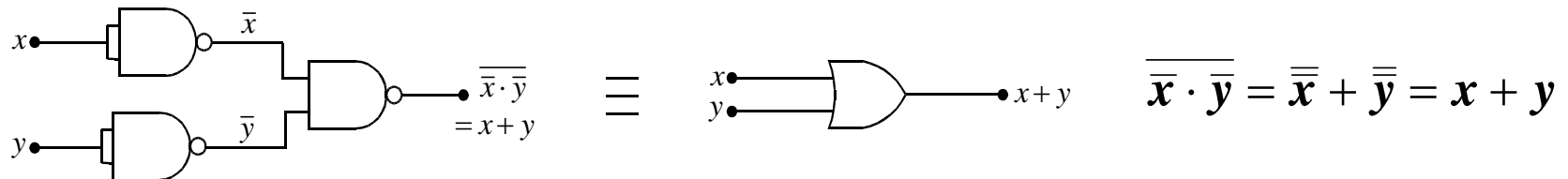
## 1. 使用 NAND 閘取代 NOT 閘



## 2. 使用 NAND 閘取代 AND 閘

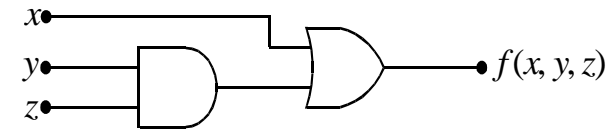


### 3. 使用 NAND 閘取代 OR 閘



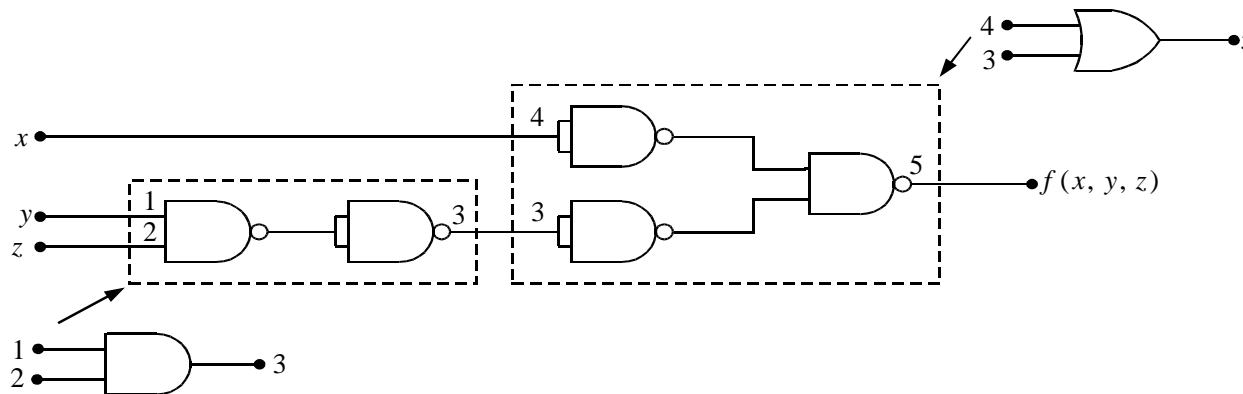
## 例題 2-12

試利用 NAND 閘來實現邏輯函數  $f(x, y, z) = x + y \cdot z$  。

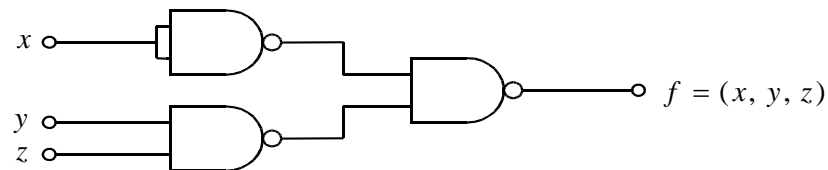


解

由輸入端開始直到輸出端為止，即設法將每一個邏輯閘直接轉換成為 NAND 閘，如下圖所示。



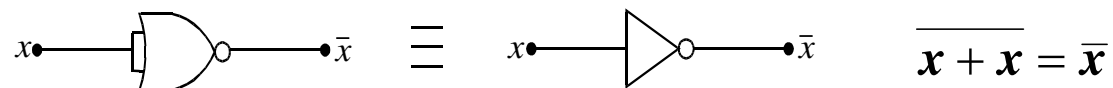
直接將兩個連續串接之反相閘刪除 (即  $\overline{\overline{x}} = x$ )，即可得 NAND 所構成之邏輯電路，如下圖所示。



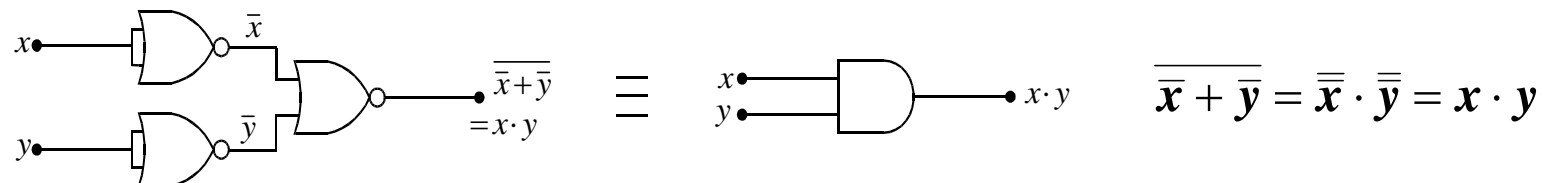
# 邏輯閘之取代(NOR 閘)

## ◆ 使用 NOR 取代 NOT 閘、AND 閘與 OR 閘

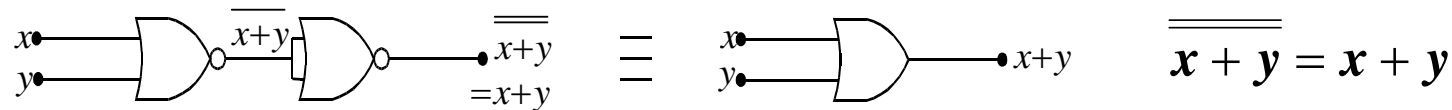
### 1. 使用 NOR 閘取代 NOT 閘



### 2. 使用 NOR 閘取代 AND 閘

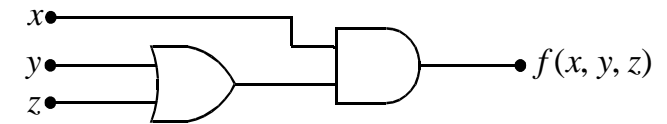


### 3. 使用 NOR 閘取代 OR 閘



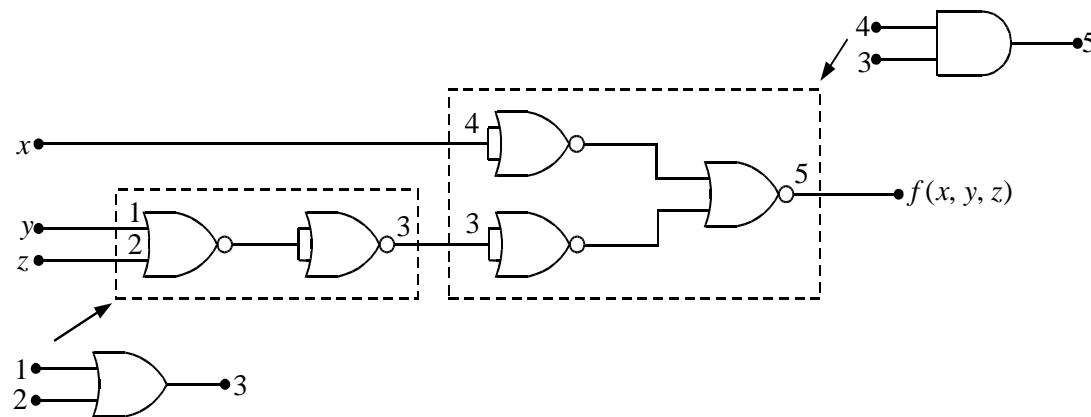
## 例題 2-13

試利用 NOR 閘來實現邏輯函數  $f(x, y, z) = x \cdot (y + z)$ 。

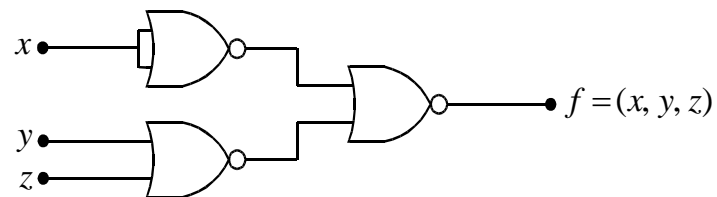


解

由輸入端開始直到輸出端為止，即設法將每一個邏輯閘直接轉換成為 NOR 閘，如下圖所示。



直接將兩個連續串接之反相閘刪除 (即  $\overline{\overline{x}} = x$ )，即可得 NOR 所構成之邏輯電路，如下圖所示。



# 邏輯閘之基本特性

- ◆ 邏輯閘是用來實現數位系統之基本元件，目前構成這些邏輯閘皆為**半導體材料**，大致上可分為使用**電源電壓為 5V 之雙載子技術**(此類之 IC 編號大多以 54 和 74 系列為主)與使用**電源電壓範圍較寬(0 至 18V)之單載子技術**(此類之 IC 編號大多以 40 系列為主)等兩種，所組成之積體電路邏輯族。
- ◆ 邏輯閘所處理之資料僅為 **0 與 1 等兩個邏輯值**，而這兩個邏輯值，可用**電壓或電流之大小來區分**，即邏輯 0 與邏輯 1 分別代表**某一個電壓或電流之範圍**，如下表所示。

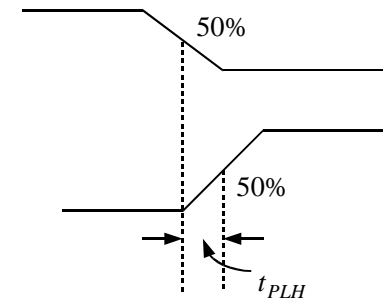
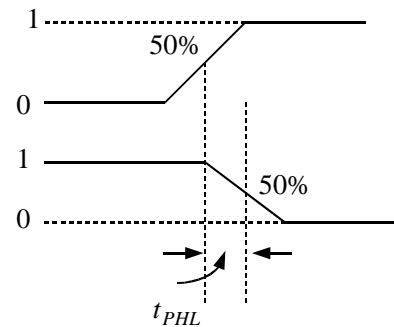
IC 類別		TTL	CMOS
電源電壓		$V_{CC} = 5V$	$V_{DD} = 3 \sim 18V$
輸入	高電壓範圍(邏輯 1)	$2.0 \sim 5V$	$0.7V_{DD} \sim V_{DD}$
	模糊範圍	$0.8 \sim 2V$	$0.3V_{DD} \sim 0.7V_{DD}$
	低電壓範圍(邏輯 0)	$0 \sim 0.8V$	$0 \sim 0.3V_{DD}$
輸出	高電壓範圍(邏輯 1)	$2.4 \sim 5V$	$0.95V_{DD}$
	模糊範圍	$0.4 \sim 2.4V$	$0.1 \sim 0.95V_{DD}$
	低電壓範圍(邏輯 0)	$0 \sim 0.4V$	$0.1V$

# 邏輯閘之扇入、扇出與傳遞延遲時間

- ◆ 由於物理之限制，每一個邏輯閘所允許之輸入端數目會受到一定的限制，即一個邏輯閘能接受之最多輸入端的能力，稱為邏輯閘之扇入 (Fan-In)。
- ◆ 邏輯閘之輸出功率亦有一定之限制，導致在電路中一個邏輯閘之輸出電壓位準可維持在指定範圍內，所能推動相同邏輯閘之數目，稱邏輯閘之扇出 (Fan-Out)。
- ◆ 由於物理之限制，輸出訊號與輸入訊號是無法同時作反應，而是當將新的輸入訊號送到邏輯閘之輸入端時，輸入訊號經過邏輯運算後，會延緩一些時間才會在輸出端產生新的訊號，這段延緩之時間稱為傳遞延遲時間 (Propagation Delay Time)。
- ◆ 邏輯閘之傳遞延遲時間，大致上可用  $t_{PHL}$  與  $t_{PLH}$  等兩個參數來表示，而這兩個參數定義如下。

1.  $t_{PHL}$  : 當輸出電壓由高態 (High) 轉換至低態 (Low) 所需之時間。

2.  $t_{PLH}$  : 當輸出電壓由低態 (Low) 轉換至高態 (High) 所需之時間。



# 問題討論

## 1、試討論布林代數之基本性質

解：

布林代數可用一組基本元素之集合、一組運算子與許多未證明之定理與假設來定義。基本上，布林代數可用下式來表示：

$$B = f(w, x, y, z, \dots)$$

上式亦可被稱為**布林函數** (Boolean Function)，其中  $B$  為輸出變數  $w$ 、 $x$ 、 $y$ 、 $z$  等 4 個輸入變數之函數，即  $B$  之值是由  $w$ 、 $x$ 、 $y$ 、 $z$  等變數來決定，因構成布林代數之基本元素，僅被允許為 0 與 1 兩個可能元素之一，且所有數位系統之輸入與輸出變數，亦僅被限制為 0 與 1 兩個可能邏輯值之一，而這兩個邏輯值 (0 與 1) 是分別用來代表數位系統之**低電位與高電位**的電壓位準。

2、試列出布林函數之**運算優先順序**。

解：

在進行布林代數之運算式時，運算子之運算優先順序 (Priority) 如下：

(1) 括號

(2) NOT

(3) AND

(4) OR。

3、試說明積項、和項、最小項、最大項、積項之和與和項之積之定義。

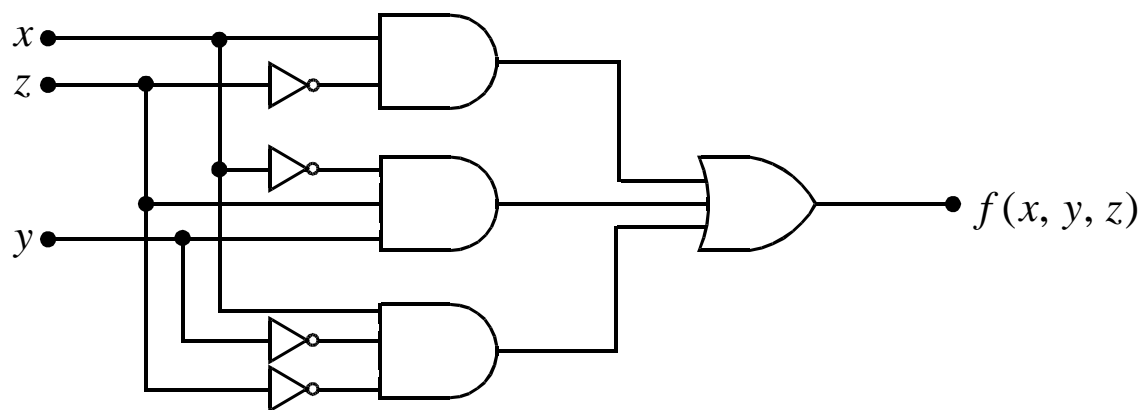
解：

- (1)、**積項**：當兩個或多個變數以 AND 運算子所組合成之布林代數表示式。
- (2)、**和項**：當兩個或多個變數以 OR 運算子所組合成之布林代數表示式。
- (3)、**最小項**：由  $n$  個不同變數所組成之積項表示式，即積項表示式皆包含到  $n$  個變數，且每個變數僅可以補數或非補數形式出現一次。。
- (4)、**最大項**：由  $n$  個不同變數所組成之和項表示式，即和項表示式皆包含到  $n$  個變數，且每個變數僅可以補數或非補數形式出現一次。
- (5)、**積項之和**：以最小項之和來表示布林函數時，稱為積項之和 (SOP)，即在真值表中，輸出值為 1 之所有最小項，用 OR 連結起來。
- (6)、**和項之積**：以最大項之積來表示布林函數時，稱為和項之積 (POS)，即在真值表中，輸出值為 0 之所有最大項，用 AND 連結起來。

4、試討論利用基本邏輯來實現布林代數之邏輯閘網路的基本原則。

解：

利用邏輯閘 (Logic Gate) 來實現布林函數時，首先要考慮為**運算優先權** (Priority) 之問題，即**運算優先權愈高者**，應放置於較**接近邏輯閘網路之輸入端** (Input)，而**運算優先權愈低者**，則應放置**愈靠近邏輯閘網路之輸出端** (Output)。例如實現  $f(x, y, z) = x \cdot \bar{z} + \bar{x} \cdot y \cdot z + x \cdot \bar{y} \cdot \bar{z}$  之邏輯閘網路為



## 5、試討論基本邏輯閘的傳遞延遲時間之意義與種類。

解：

由於物理之限制，輸出訊號與輸入訊號是無法同時作反應，而當將新的輸入訊號送到邏輯閘之輸入端時，輸入訊號經過邏輯運算後，通常會延緩一些時間才可在輸出端產生新的訊號，這段延緩之時間稱為傳遞延遲時間 (Propagation Delay Time)。一般而言，傳遞延遲時間是訊號進入邏輯閘之輸入端時，使邏輯閘內部之電晶體 (Transistor)，由截止進入飽和 (或由飽和進入截止) 所需之時間，因此傳遞延遲時間可用來顯示邏輯電路之運算速度。

在一般邏輯閘之 IC 手冊中，可查閱到關於邏輯閘之傳遞延遲時間，大致上可用  $t_{PHL}$  與  $t_{PLH}$  等兩個參數來表示，接著分別將此兩個參數 (Parameter) 定義如下：

- (1).  $t_{PHL}$ ：當輸出電壓由高態之 50% 轉換至低態之 50% 所需之時間。
- (2).  $t_{PLH}$ ：當輸出電壓由低態之 50% 轉換至高態之 50% 所需之時間。