



正確學會

改訂新版 デジタル回路と Verilog HDL

Verilog

的16堂課

第 1 章

基本數位邏輯電路

本投影片（下稱教用資源）僅授權給採用教用資源相關之旗標書籍為教科書之授課老師（下稱老師）專用，老師為教學使用之目的，得摘錄、編輯、重製教用資源（但使用量不得超過各該教用資源內容之80%）以製作為輔助教學之教學投影片，並於授課時搭配旗標書籍公開播放，但不得為網際網路公開傳輸之遠距教學、網路教學等之使用；除此之外，老師不得再授權予任何第三人使用，並不得將依此授權所製作之教學投影片之相關著作物移作他用。

本章重點

- 1.1 基本邏輯電路
- 1.2 電路構成
- 1.3 延伸學習

1.1 基本邏輯電路

- 「H(高電位)」 \rightarrow '真' (也就是 1)
- 「L(低電位)」 \rightarrow '假' (也就是 0)

1.1.1 AND 電路

- 全部輸入皆為 'H' 的時候才會輸出 'H'

B	A	Y
L	L	L
L	H	L
H	L	L
H	H	H



雙輸入

C	B	A	Y
L	L	L	L
L	L	H	L
L	H	L	L
L	H	H	L
H	L	L	L
H	L	H	L
H	H	L	L
H	H	H	H

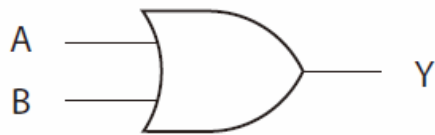


三輸入

1.1.2 OR 電路

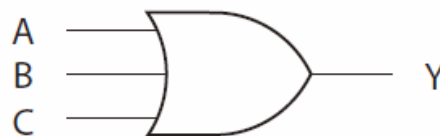
- 只要任何一個輸入為 'H' 的時候, 輸出就會為 'H'
- 全部輸入都為 'L' 才會輸出 'L'

B	A	Y
L	L	L
L	H	H
H	L	H
H	H	H



雙輸入

C	B	A	Y
L	L	L	L
L	L	H	H
L	H	L	H
L	H	H	H
H	L	L	H
H	L	H	H
H	H	L	H
H	H	H	H

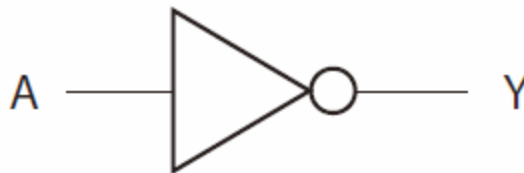


三輸入

1.1.3 NOT 電路

- 當輸入為 'H' 時輸出為 'L'；輸入為 'L' 的時候輸出則為 'H'
- 輸出的邏輯會跟輸入相反

A	Y
L	H
H	L



1.1.4 NAND 電路

- 在 AND 電路的輸出接上 NOT 電路
- 全部輸入皆為 'L' 的時候才會輸出 'H'

B	A	Y
L	L	H
L	H	L
H	L	L
H	H	L



1.1.5 NOR 電路

- 在 OR 電路的輸出接上 NOT 電路
- 只要任何一個輸入為 'L' 的時候，輸出就會為 'H'

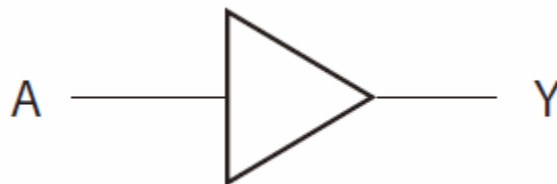
B	A	Y
L	L	H
L	H	H
H	L	H
H	H	L



1.1.6 緩衝器電路

- 將輸入的邏輯原封不動的輸出
- 用來調整不同裝置間的電位差異

A	Y
L	L
H	H



1.2 電路構成

- 邏輯電路分為：

- ▣ 組合電路 (combination logical circuit)：由邏輯閘組合的電路，根據輸入的值就可以推斷出輸出的結果，不受前一個狀態的影響
- ▣ 序向電路 (sequential logical circuit)：由組合電路跟記憶電路所構成，輸出的結果會根據輸入的值與電路現在的狀況而改變

1.2.1 組合電路及設計方法

- 組合電路設計法
 - ▣ 加法標準型設計法 (2 段邏輯設計法)
 - ▣ 多段邏輯設計法
- 多段邏輯設計法使用的邏輯閘會比加法標準型設計法少

1.2.2 正邏輯與負邏輯

- 正邏輯 (positive logic)

H (高電位) 為 '真' (也就是 '1')

L (低電位) 為 '假' (也就是 '0')

- 負邏輯 (negative logic)

H (高電位) 為 '假' (也就是 '0')

L (低電位) 為 '真' (也就是 '1')

1.2.3 正/負邏輯與 AND/OR 的關係

- (1) AND 電路的變換

- ▣ 正邏輯：

$$Y = A \cdot B$$

- ▣ 負邏輯：

$$\overline{Y} = \overline{A \cdot B}$$

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

兩邊都取否定

- 負邏輯信號當輸入, AND 電路的電路符號如下：



- 正邏輯信號為輸入時, 以 AND 原理來動作且輸出為正邏輯
- 負邏輯信號為輸入時, 以 OR 原理來動作且輸出為負邏輯

(2) OR 電路的變換

- 正邏輯

- ▣ $Y = A + B$

- 負邏輯

$$\begin{aligned}\overline{Y} &= \overline{A + B} \\ &= \overline{A} \cdot \overline{B}\end{aligned}$$

兩邊都取負數

- 負邏輯信號當輸入, OR 電路的電路符號如下：



- 正邏輯輸入, 以 OR 原理來動作且輸出為正邏輯
- 負邏輯輸入, 以 AND 原理來動作且輸出為負邏輯

(3) NAND 電路的變換

- 正邏輯

$$Y = \overline{A \cdot B}$$



- 負邏輯

$$\overline{Y} = \overline{\overline{A} \cdot \overline{B}}$$

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

$$Y = A + B$$



- 正邏輯輸入，以 OR 原理來動作且輸出為正邏輯
- 負邏輯輸入，以 AND 原理來動作且輸出為負邏輯

(4) NOR 電路的變換

- 正邏輯

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



- 負邏輯

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

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

$$Y = \overline{\overline{A} \cdot \overline{B}}$$



- 正邏輯輸入，以 AND 原理來動作且輸出為正邏輯
- 負邏輯輸入，以 OR 原理來動作且輸出為負邏輯

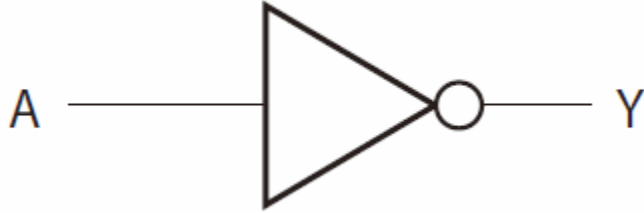
小結

歸納 (1)~(4) 可得

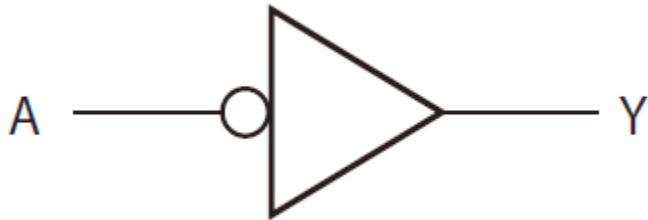
- 以負邏輯信號為輸入時, AND 跟 OR 機能會互換

(5) NOT 電路的變換

- 正邏輯

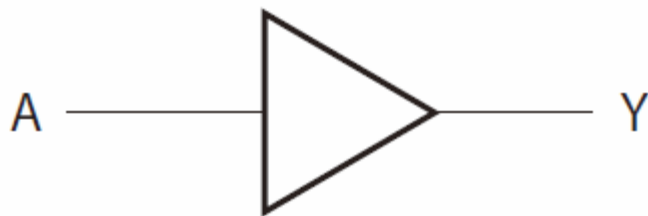


- 負邏輯

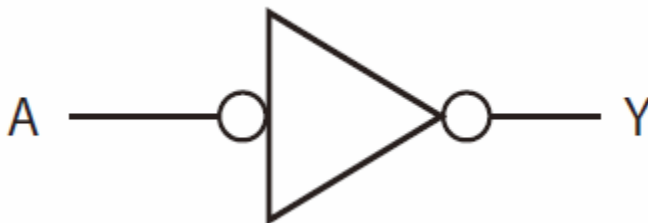


(6) 緩衝器電路的變換

- 正邏輯



- 負邏輯

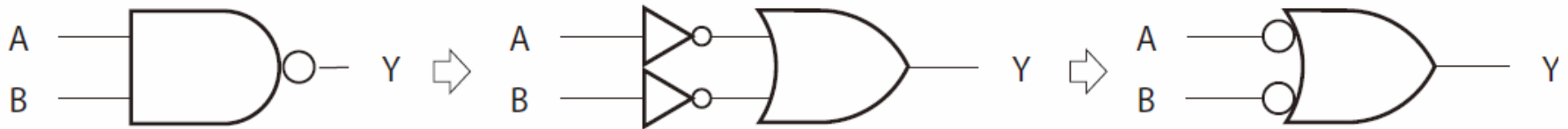


1.2.4 根據笛摩根定律來做電路變換

- 使用「笛摩根定律」，AND 跟 OR 電路可以相互變換

(1) 從 NAND 電路變換為 OR 電路

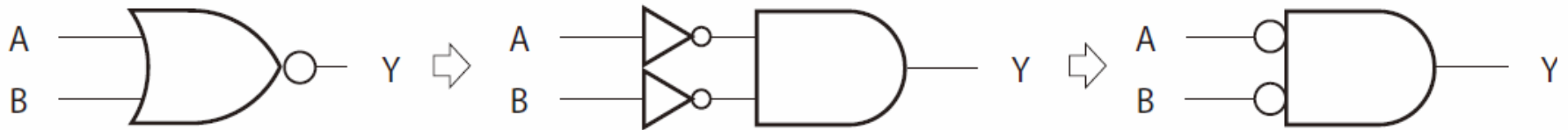
$$Y = \overline{A \cdot B}$$
$$= \overline{A} + \overline{B}$$



(a) 反向 OR 電路

(2) 從 NOR 電路變換為 AND 電路

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



(b) 反向 AND 電路

1.2.5 以笛摩根定律把所有的邏輯都變換為 NAND 電路

- 依據之前的推導, 任何邏輯電路都可用 NAND 電路構成等效電路
- NAND 電路延遲時間較短且耗電量也較低

1.3 延伸學習

- 正邏輯與負邏輯的邏輯符號要怎樣分辨

項目	輸入	功能	輸出	使用電路	邏輯符號
1	正邏輯	AND	正邏輯	AND	圖 1.8
2	正邏輯	AND	負邏輯	NAND	圖 1.12
3	負邏輯	AND	正邏輯	NOR	圖 1.15
4	負邏輯	AND	負邏輯	OR	圖 1.11
5	正邏輯	OR	正邏輯	OR	圖 1.10
6	正邏輯	OR	負邏輯	NOR	圖 1.14
7	負邏輯	OR	正邏輯	NAND	圖 1.13
8	負邏輯	OR	負邏輯	AND	圖 1.9
9	正邏輯	NOT	負邏輯	NOT	圖 1.16
10	負邏輯	NOT	正邏輯	NOT	圖 1.17
11	正邏輯	BUF	負邏輯	BUF	圖 1.18
12	負邏輯	BUF	正邏輯	BUF	圖 1.19

- 電路符號的‘○’：
 - ▣ 代表 NOT 電路
 - ▣ 代表 NOT 的功能
 - ▣ 代表負邏輯