

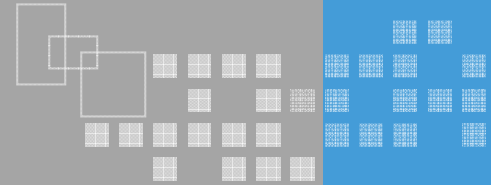
第7章

利用邏輯閘作組合電路 設計與模擬

7.2 具有閘扇入數限制的電路設計

❖ 扇入數(fan-in)：每個閘輸入的最大數目。

範例



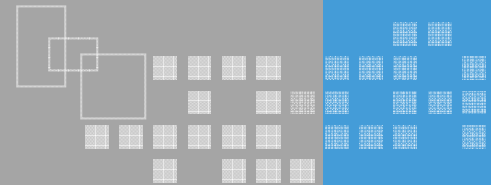
❖ 用三輸入的NOR閘實現 $f(a, b, c, d) = \Sigma m(0, 3, 4, 5, 8, 9, 10, 14, 15)$ 。

f 的圖：

$ab \backslash cd$	00	01	11	10
00	1	1	0	1
01	0	1	0	1
11	1	0	1	0
10	0	0	1	1

$$f' = a'b'c'd + ab'cd + abc' + a'bc + a'cd'$$

範例



- ❖ 由上面表示式可知，一個二階電路需要兩個四輸入閘和一個五輸入閘。這個 f' 的表示式被分解以降低最大的閘輸入數成為三個，然後再取它的補數：

$$f' = b'd(a'c' + ac) + a'c(b + d') + abc'$$

$$f = [b + d' + (a + c)(a' + c')][a + c' + b'd][a' + b' + c]$$

範例

❖ 圖7-1所示為所得的NOR閘電路。

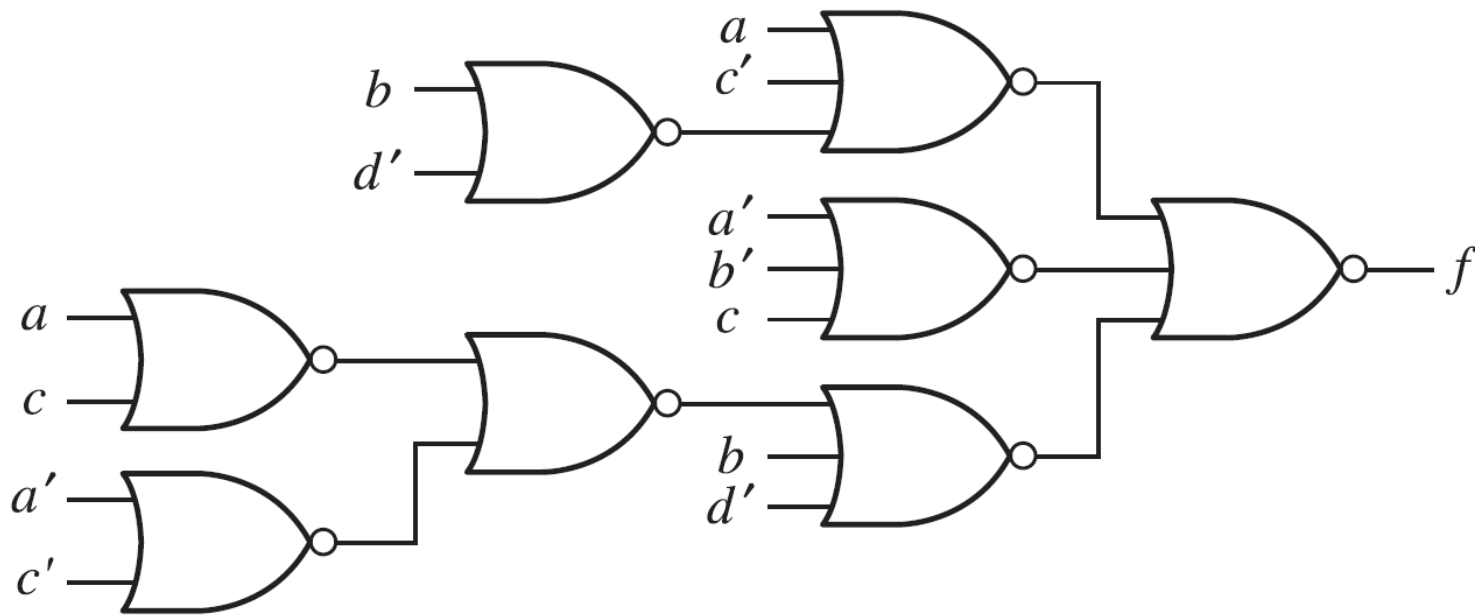
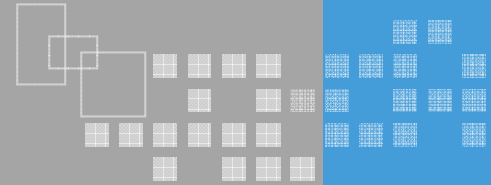
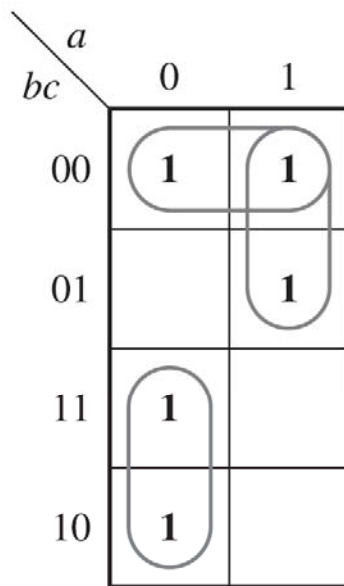


圖 7-1

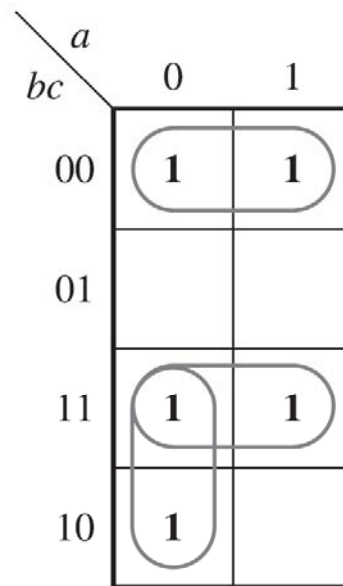
範例



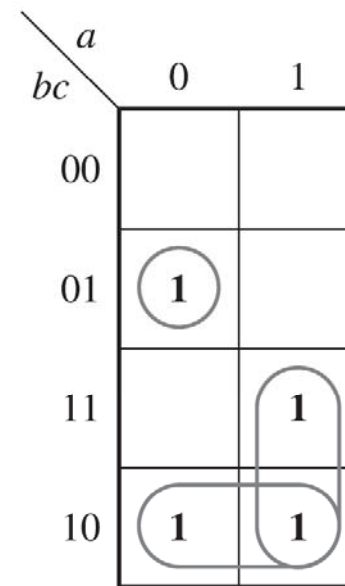
❖ 只用二輸入的NAND 閘和反相器實現圖7-2中的函數。



$$f_1 = \Sigma m(0, 2, 3, 4, 5)$$



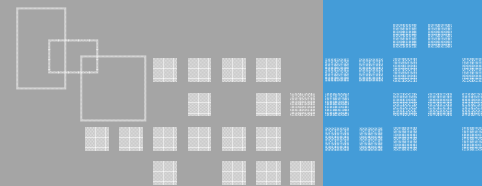
$$f_2 = \Sigma m(0, 2, 3, 4, 7)$$



$$f_3 = \Sigma m(1, 2, 6, 7)$$

圖 7-2

範例



❖ 假如我們個別簡化每個函數，則所得結果是：

$$f_1 = b'c' + ab' + a'b$$

$$f_2 = b'c' + bc + a'b$$

$$f_3 = a'b'c + ab + bc'$$

❖ 每個函數需要一個三輸入的OR閘，所以我們將分解以減少閘輸入的數目。

$$f_1 = b'(\underline{a + c'}) + \underline{a'b}$$

$$f_2 = b(a' + c) + b'c' \quad \text{或} \quad f_2 = (b' + c)(b + c') + \underline{a'b}$$

$$f_3 = a'b'c + b(\underline{a + c'})$$

範例

❖ 藉由下式從 f_3 消掉剩下的三輸入閘：

$$a'b'c = a'(b'c) = a'(b + c)'$$

❖ 圖7-3(a)所示為使用共同項 $a'b$ 和 $a + c'$ 所得的電路。

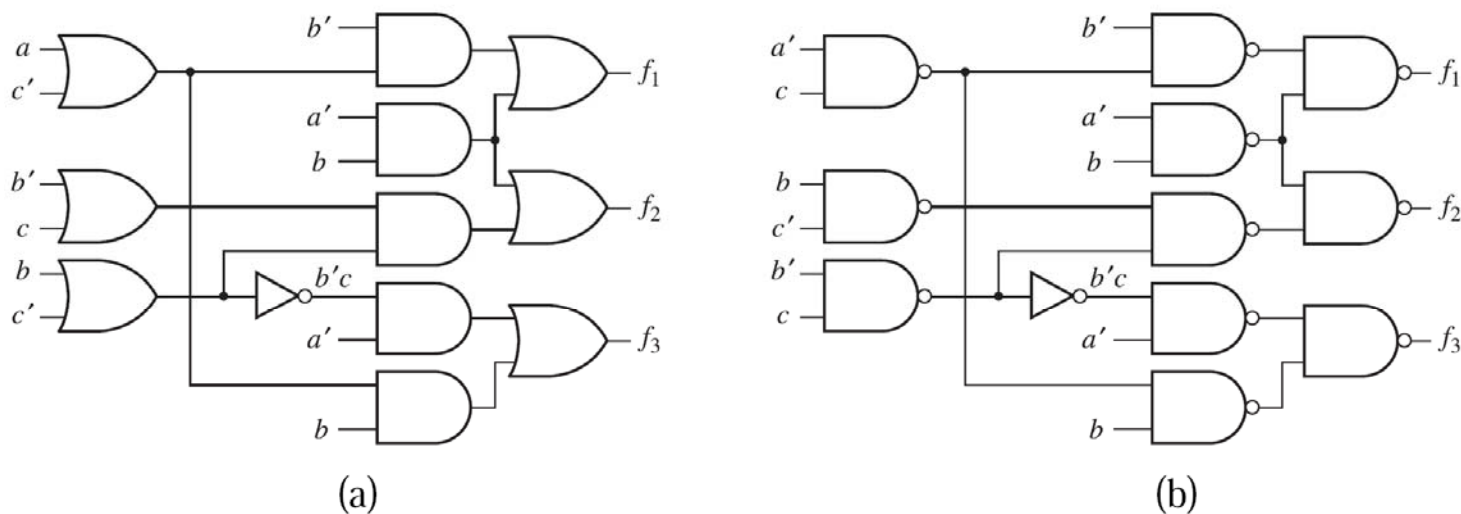


圖 7-3 圖 7-2 的實現

7.3 閘延遲與時序圖

❖ 如果輸出變化相對於輸入變化的延遲時間為 ε ，則稱這個邏輯閘的**傳播延遲**（propagation delay）為 ε 。

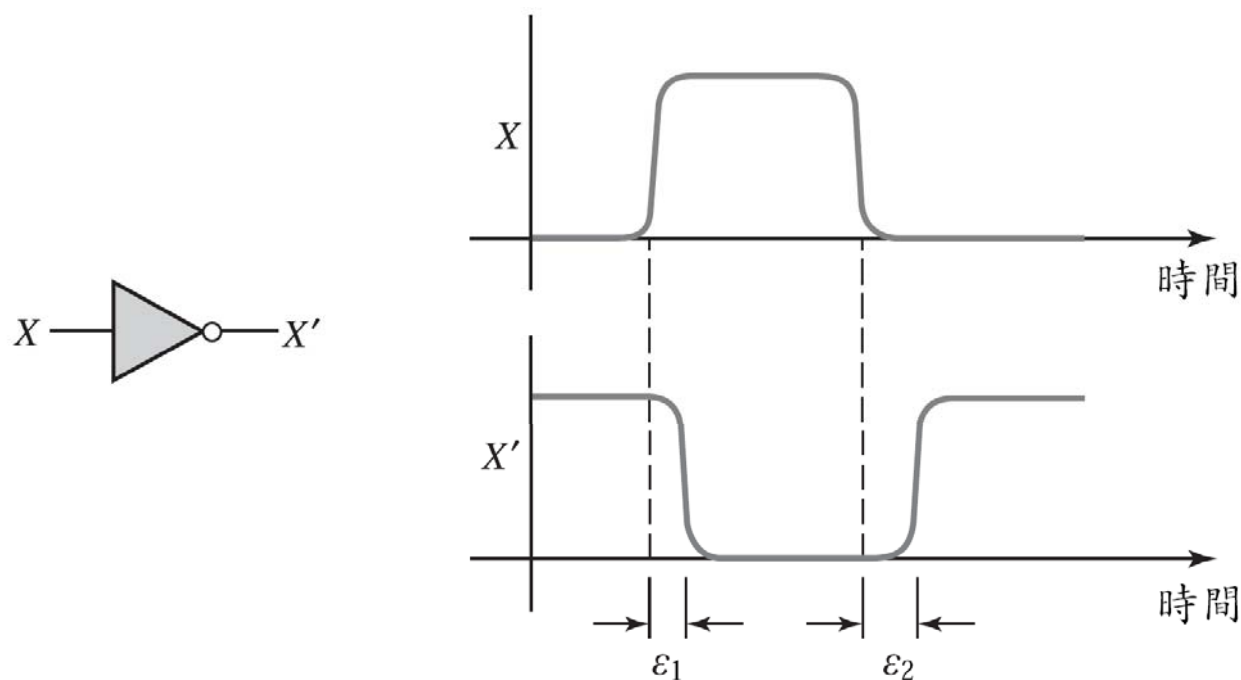
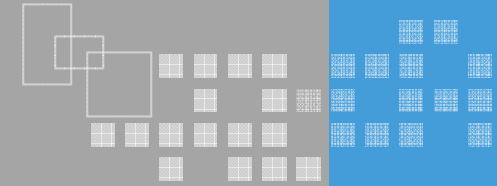


圖 7-4 反相器中的傳播延遲

7.3 閘延遲與時序圖



❖ **時序圖** (timing diagrams)

❖ 假設每個邏輯閘的傳播延遲為20奈秒 (nanoseconds, ns)。

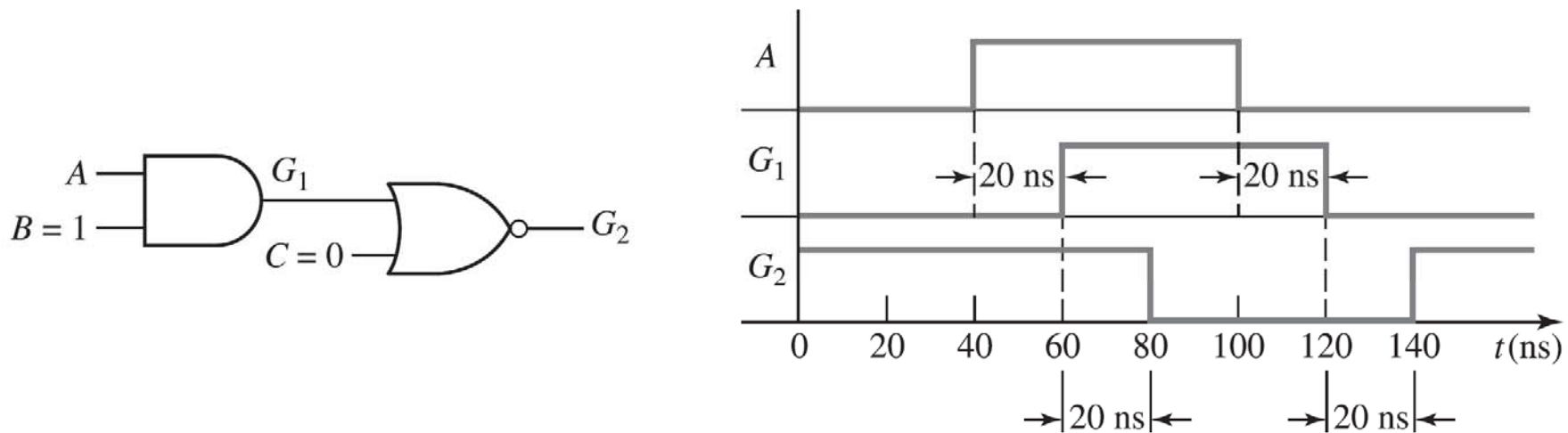
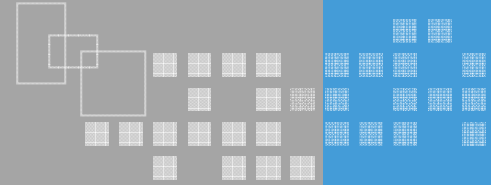


圖 7-5 AND-NOR 電路的時序圖

7.3 閘延遲與時序圖



- ❖ 這個時序圖指出，當閘輸入 B 和 C 維持在常數值 1 和 0，且輸入 A 在 $t = 40\text{ ns}$ 變成 1，然後在 $t = 100\text{ ns}$ 又回到 0 時，分別發生了什麼事。閘 G_1 的輸出在 A 變化之後 20 ns 發生改變，且閘 G_2 的輸出在閘 G_1 變化之後 20 ns 發生改變。

7.3 閘延遲與時序圖

❖ 圖7-6 所示為加上一個延遲元件電路的時序圖

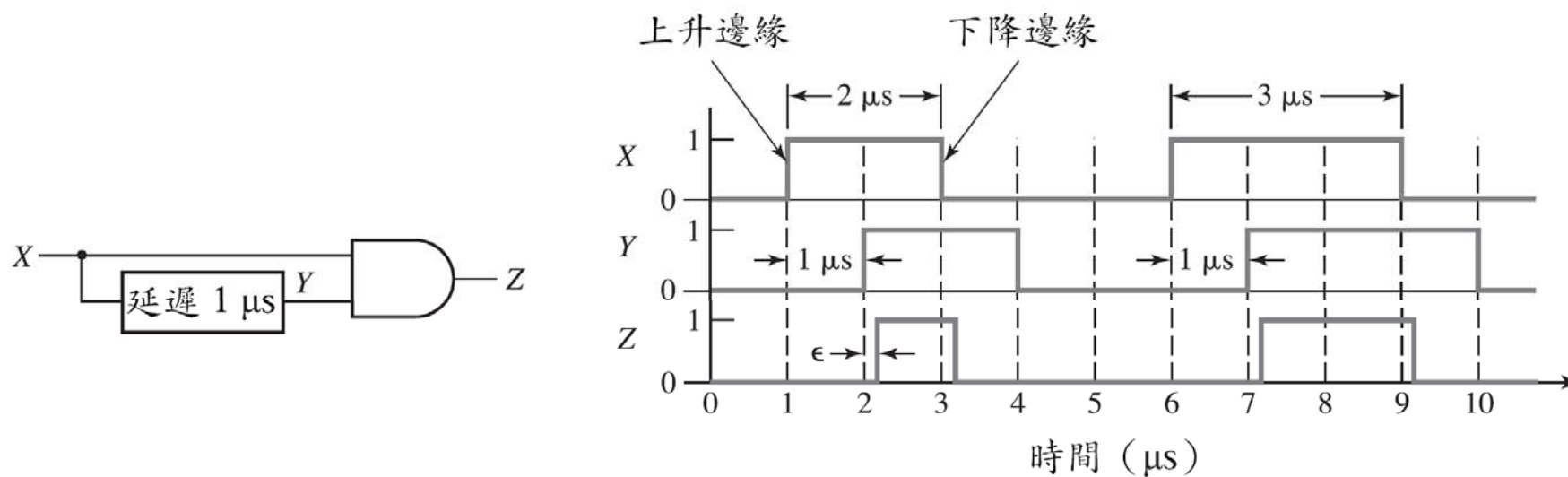
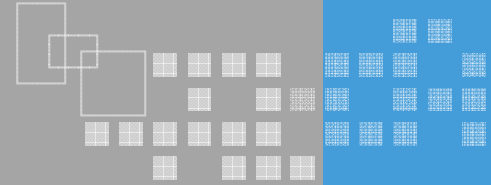


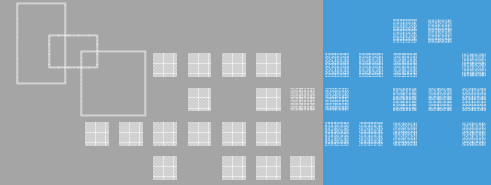
圖 7-6 具有延遲元件的電路的時序圖

7.4 組合邏輯中的雜訊



- ❖ 當組合電路中的輸入變化時，不想要的交換暫態可能出現在電路的輸出端。
- ❖ **靜態1-雜訊**（static 1-hazard）：
 - 若因任何單一輸入的變化及傳播延遲之組合的影響，一個電路的輸出應該維持常數1，但卻暫時地變成0時，我們稱此電路有靜態1-雜訊。
- ❖ **靜態0-雜訊**（static 0-hazard）：
 - 如果輸出應該維持於常數0，卻暫時地變成1時，我們稱此電路有靜態0-雜訊。

7.4 組合邏輯中的雜訊



❖ 動態雜訊 (dynamic hazard) :

- 若當輸出被設成由0變成1（或由1變0），輸出可能變動三次或更多次時，我們稱此電路有動態雜訊。

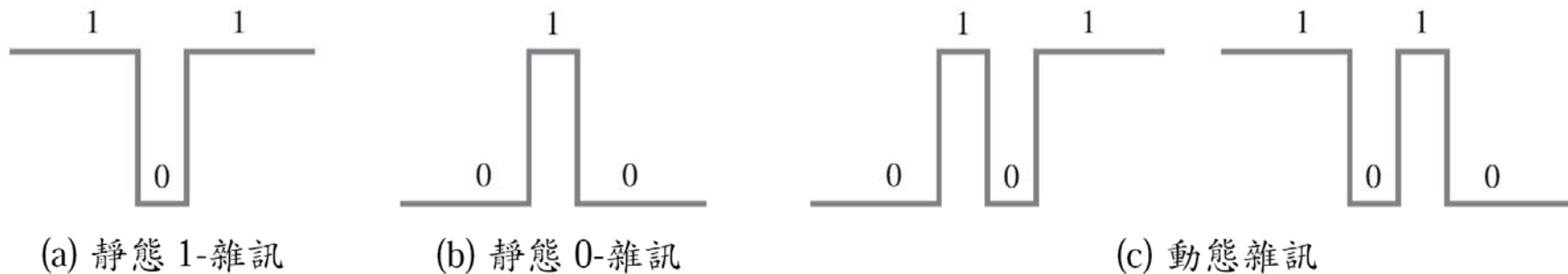
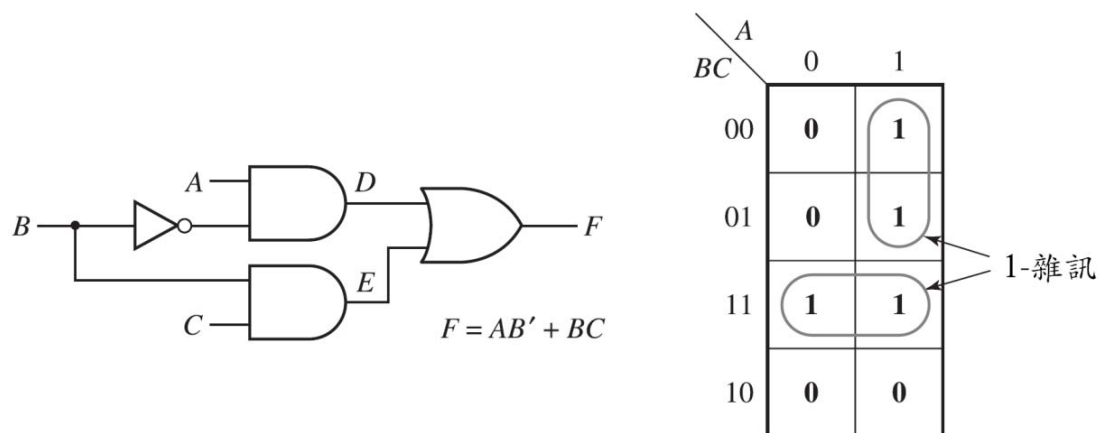


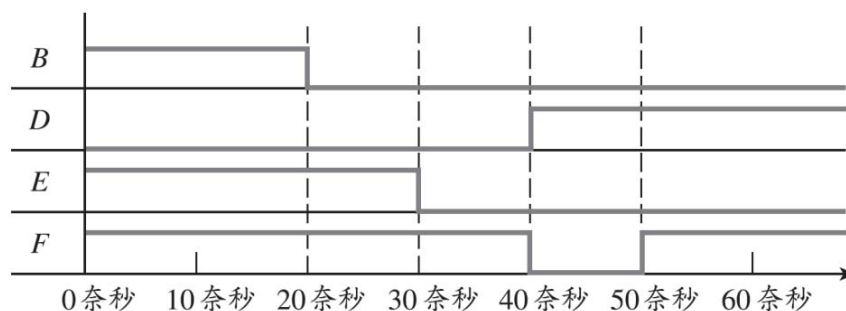
圖 7-7 雜訊的型態

7.4 組合邏輯中的雜訊

❖ 具有靜態1-雜訊的電路



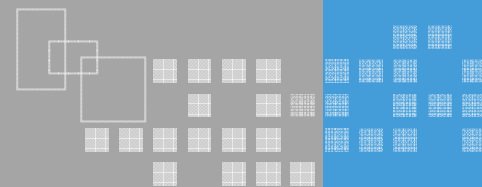
(a) 具有靜態 1-雜訊的電路



(b) 時序圖

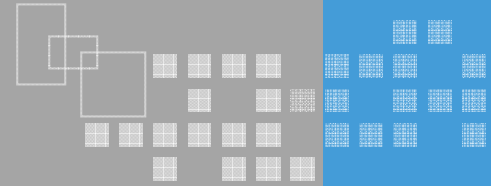
圖 7-8 1-雜訊的偵測

7.4 組合邏輯中的雜訊



- ❖ 假如 $A = C = 1$ ，則 $F = B + B' = 1$ ，所以當 B 由 1 變成 0 時，輸出 F 應該維持常數 1。但是，如圖 7-8(b) 所示，若是每個閘有傳播延遲 10 奈秒，則 E 在 D 變成 1 之前會變成 0，導致暫時 0 [由於 1-雜訊引起一個**突波** (glitch)] 出現在輸出 F 。注意，在 B 變成 0 之後的右邊，反相器的輸入 (B) 以及輸出 (B') 同時為 0，一直到傳播延遲過去為止，在這段期間，在 F 等式中的兩項同時為 0，所以 F 暫時變成 0。

7.4 組合邏輯中的雜訊



❖ 在二階ANDOR電路中偵測出雜訊：

1. 寫出電路的積項和表示式。
2. 將每一項畫在卡諾圖中，並且將它圈起來。
3. 假如任何相鄰的兩個1 沒有被相同的圈所包含，則在這兩個1 之間存在一個靜態1-雜訊的暫態。對於 n 變數圖，當一個變數改變而其他 $n - 1$ 個變數保持常數時，則會發生這樣的暫態。

7.4 組合邏輯中的雜訊

- ❖ 假如我們在圖7-8(a)中加入一個小圓，然後在電路中加入相對應的閘（圖7-9），這樣可以消去雜訊，當 B 改變時 AC 項保持1，所以在輸出沒有突波出現，注意此時 F 不再是最簡積項和。

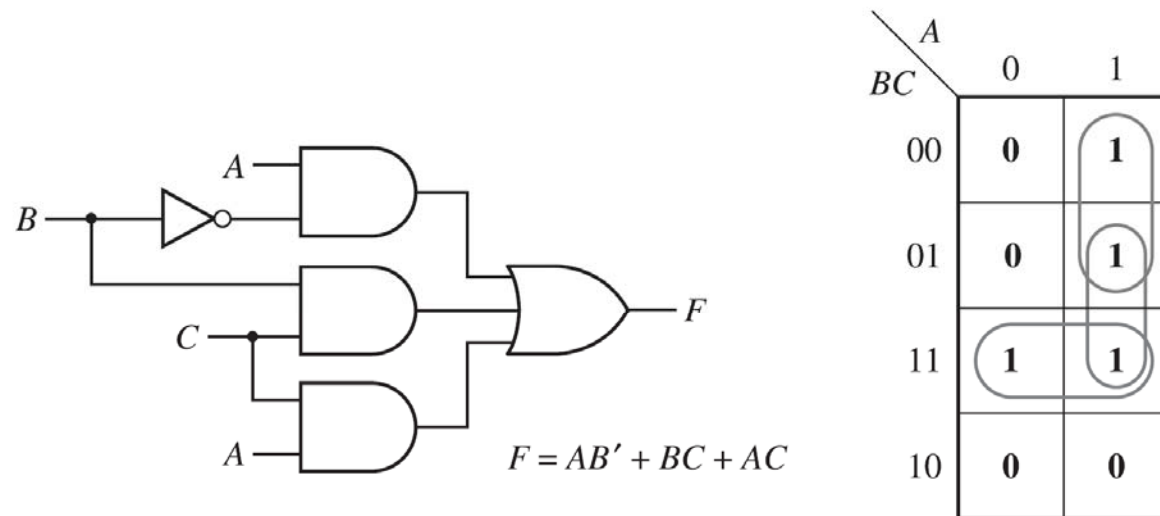
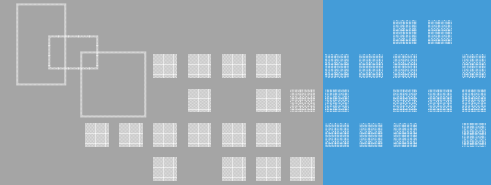


圖 7-9 具雜訊移除的電路

7.4 組合邏輯中的雜訊

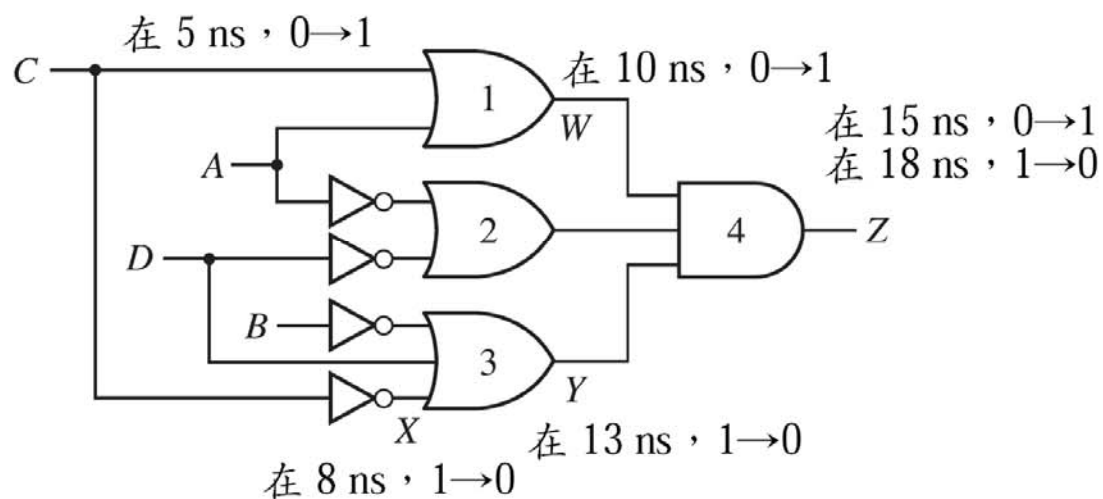
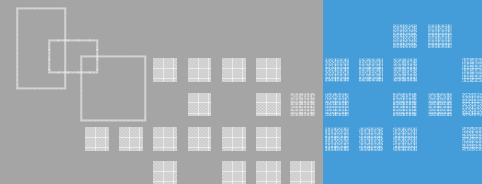


- ❖ 圖7-10(a) 所示為一具有好幾個0-雜訊的電路，此電路輸出的和項積表示為：

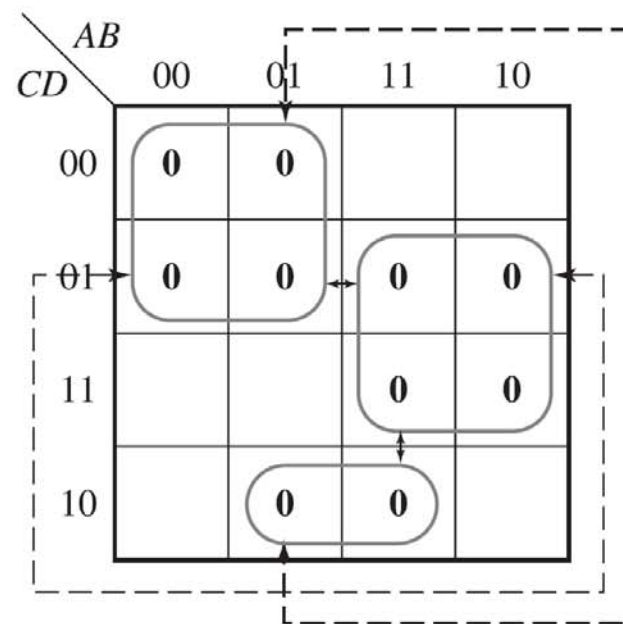
$$F = (A + C)(A' + D')(B' + C' + D)$$

- ❖ 這個函數的卡諾圖（圖7-10(b)）顯示有四對相鄰的0如箭頭所指的並沒有被共同的迴圈所包含，因此每一對相對應一個0-雜訊。例如：當 $A = 0$ 、 $B = 1$ 、 $D = 0$ 且 C 由0變成1時，對於某些閘延遲的組合，輸出 Z 可能會出現**尖波**（spike）。

7.4 組合邏輯中的雜訊



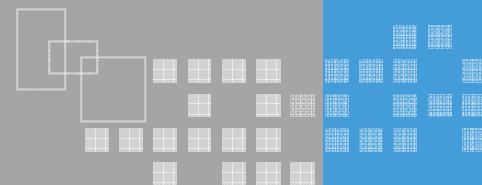
(a) 具靜態 0-雜訊的電路



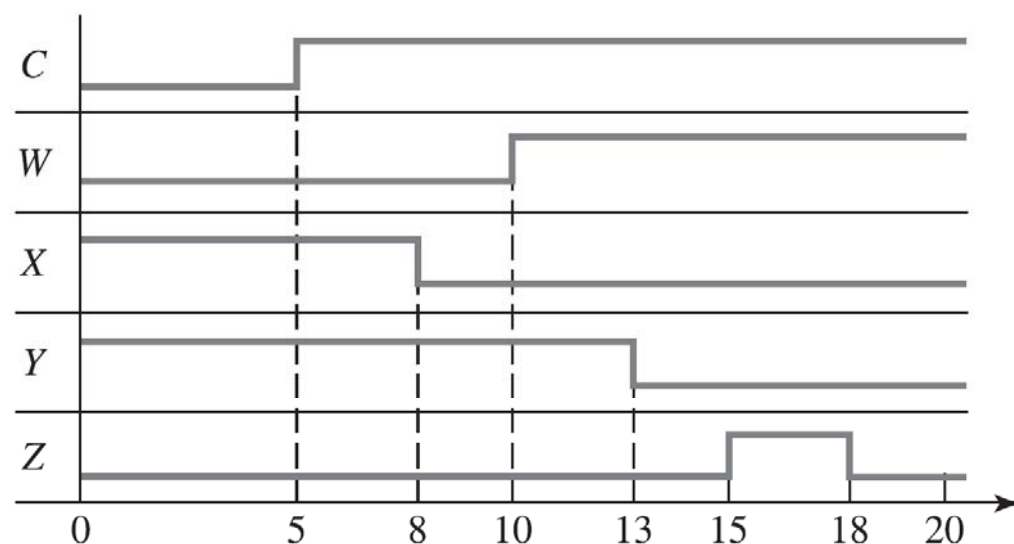
(b) 圖 (a) 中電路的卡諾圖

圖 7-10 靜態 0-雜訊的偵測

7.4 組合邏輯中的雜訊



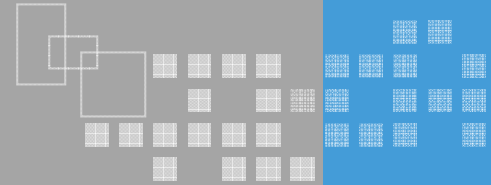
- ❖ 圖7-10(c)的時序圖，假設每個反相器的閾延遲為3 ns，且每個AND閘及每個OR閘的閾延遲為5 ns，說明了這個結果。



(c) 說明圖 (a) 0-雜訊的時序圖

圖 7-10 靜態 0-雜訊的偵測

7.4 組合邏輯中的雜訊



- ❖ 藉由迴圈上額外的質含項來包含那些沒有被共同迴圈所包含之相鄰的0而消去0-雜訊。這個需要三個額外的迴圈，如圖7-11所示。所得方程式為：

$$F = (A + C)(A' + D')(B' + C' + D)(C + D')(A + B' + D)(A' + B' + C')$$

7.4 組合邏輯中的雜訊

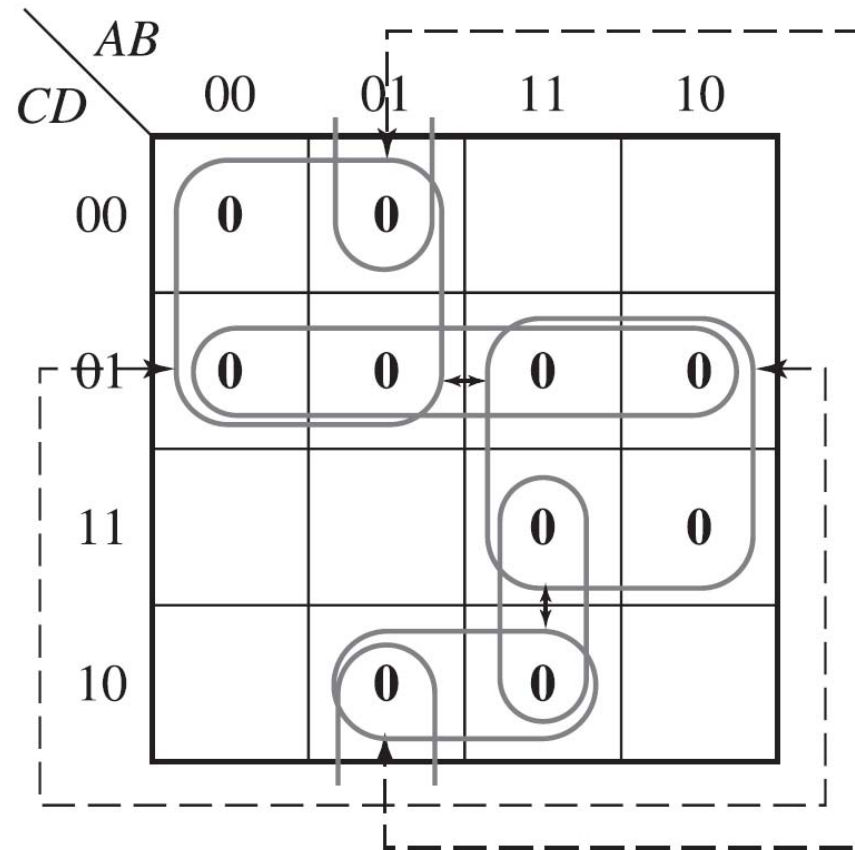
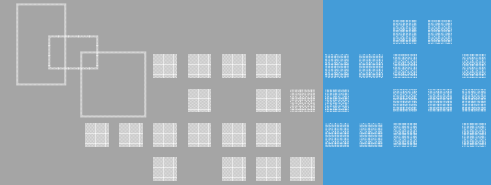


圖 7-11 圖 7-10 之移除雜訊的卡諾圖

7.4 組合邏輯中的雜訊

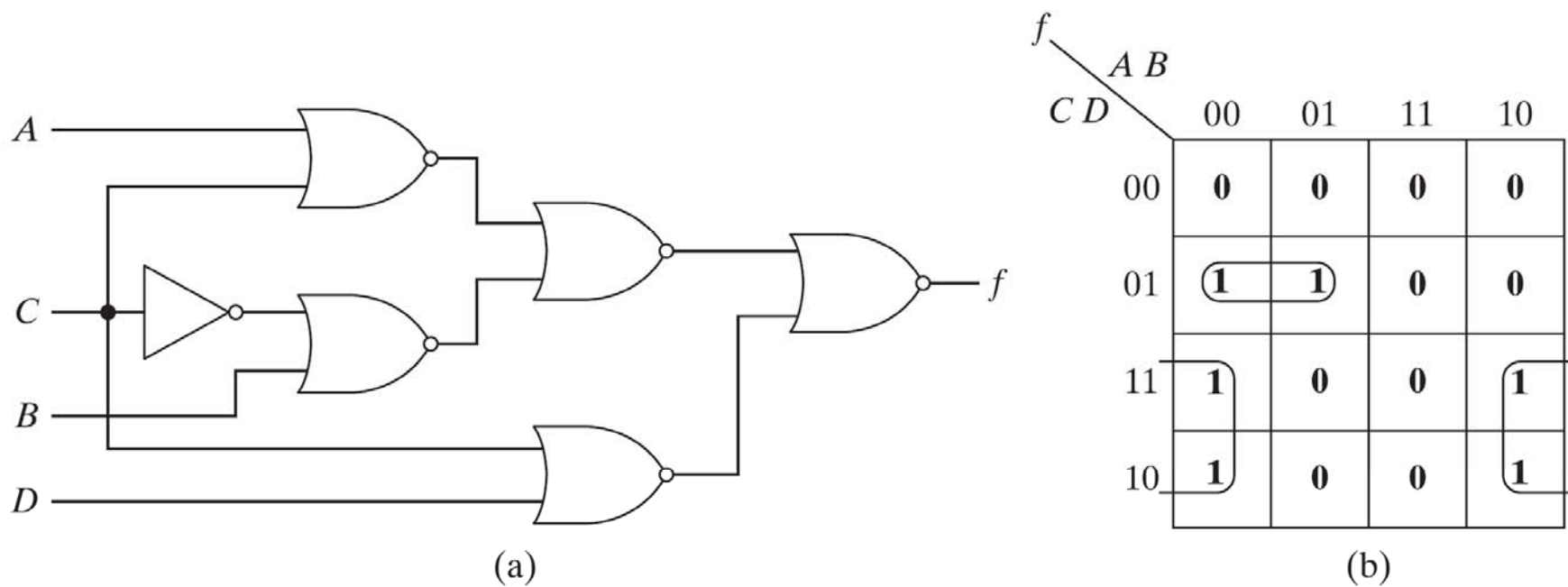
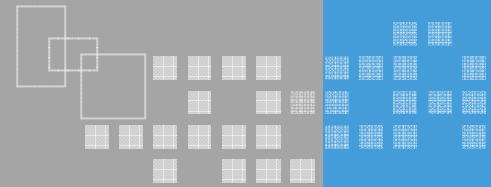
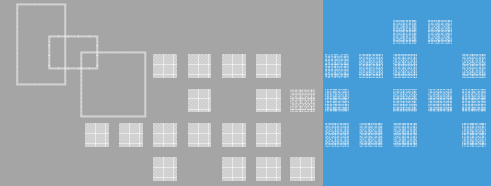


圖 7-12 雜訊的例子

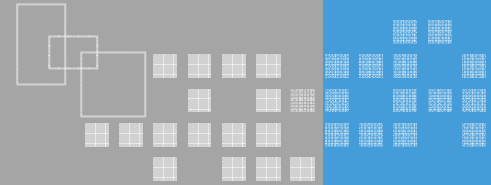
7.4 組合邏輯中的雜訊



❖ 要設計一個沒有靜態以及動態雜訊的電路，可使用下列的步驟：

1. 對於那些只被1的項所包含之每一對相鄰的1的輸出找出其積項和表示式 (F^t)。(所有質含項的和通常滿足這個條件。)則以這個 F^t 為基礎所建立的二階AND-OR電路將是沒有靜態1-雜訊、0-雜訊及動態雜訊。
2. 假如需要不同形式的電路，則利用簡單的分解、笛摩根定理等，將 F^t 轉換成所要的形式，對於每一個 x_i 和 x'_i 當成獨立的變數來處理，以免引進雜訊。

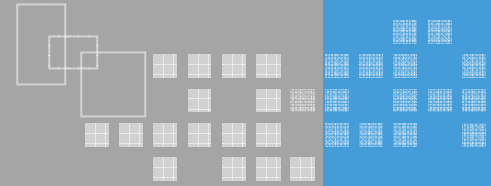
7.5 邏輯電路的模擬與測試



❖ 邏輯設計過程中，要作模擬的幾個理由：

- (1) 證明這個設計是邏輯正確的。
- (2) 證明邏輯信號的時序是正確的。
- (3) 在電路中錯誤元件的模擬有助於求出電路的測試方法。

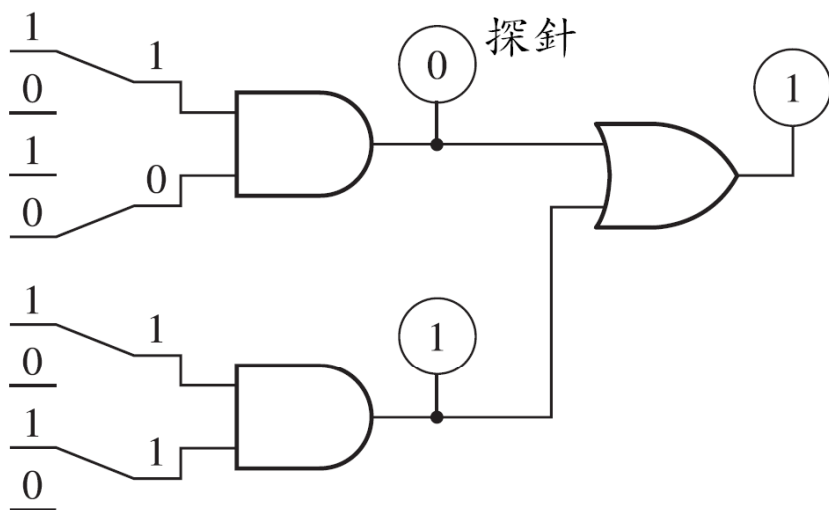
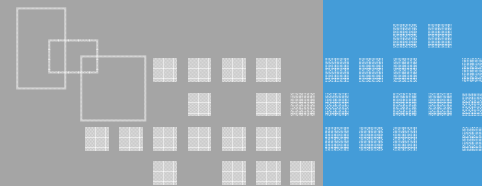
7.5 邏輯電路的模擬與測試



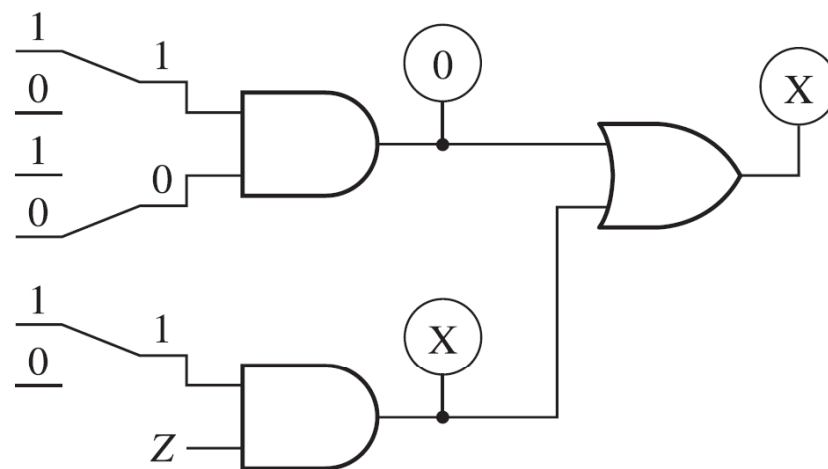
❖ 一個簡單的組合式邏輯模擬器工作如下：

1. 電路輸入被用在電路中第一組的閘，且計算這些閘的輸出。
2. 在上一個步驟中，有改變的閘輸出被接到下一層的閘輸入。假如任何閘的輸入已經改變了，則計算該閘的輸出。
3. 重作步驟2，直到閘輸入沒有發生改變為止。電路就在**穩態**（steady state）的狀況下，且可以讀出輸出值。
4. 當電路輸入改變時，重作步驟1 到3。

7.5 邏輯電路的模擬與測試



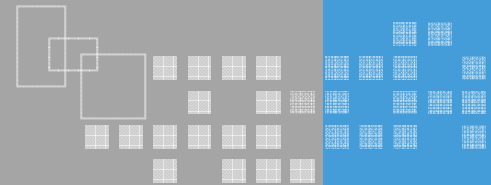
(a) 顯示開關的模擬螢幕



(b) 沒有開輸入的模擬螢幕

圖 7-13

7.5 邏輯電路的模擬與測試



❖ 表7-1所示為對於四值邏輯模擬之AND和OR函數

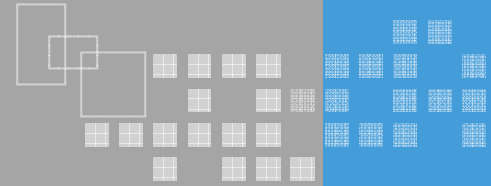


表 7-1 對於四值模擬的 AND 和 OR 函數

\cdot	0	1	X	Z
0	0	0	0	0
1	0	1	X	X
X	0	X	X	X
Z	0	X	X	X

$+$	0	1	X	Z
0	0	1	X	X
1	0	1	1	1
X	X	1	X	X
Z	X	1	X	X

7.5 邏輯電路的模擬與測試



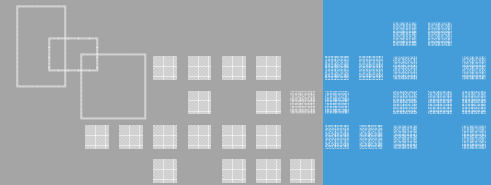
❖ 假如對某些輸入值的組合而言，電路輸出是錯誤的，這也許是因為下面幾個可能的原因：

1. 不正確的設計。
2. 閘連接錯誤。
3. 錯誤的輸入信號送到電路中。

❖ 假如這個電路是建立在實驗室中，其他可能的原因包括：

4. 不良的閘。
5. 不良的連接線。

範例



❖ 用圖7-14的電路來實現函數 $F = AB(C'D + CD') + A'B'(C + D)$

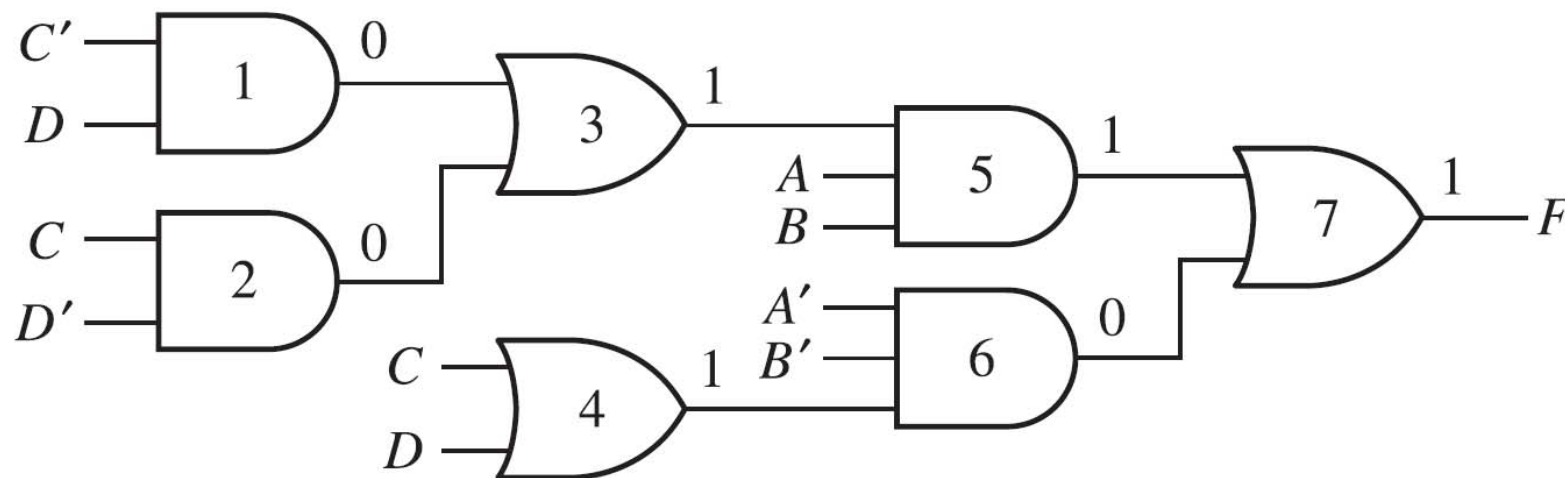
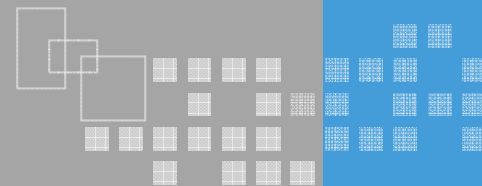


圖 7-14 具有不正確輸出的邏輯電路

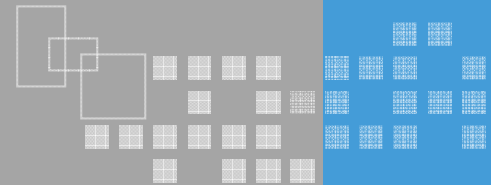
範例



❖ 當某個學生在實驗室中建立這個電路時，發現當 $A = B = C = D = 1$ 時，輸出 F 的值錯誤，且閘的輸出如圖7-13中所示。 F 不正確值的原因可以由下面來決定：

1. 閘7 (F) 的輸出是錯誤的，但是這個錯誤的輸出和到閘7的輸入一致，也就是說 $1 + 0 = 1$ 。因此，到閘7的某個輸入是錯誤的。
2. 為了使閘7有正確的輸出 ($F = 0$)，兩個輸入必須都是0。因此，閘5的輸出是錯誤的。然而，閘5的輸出和輸入一致，因為 $1 \cdot 1 \cdot 1 = 1$ 。因此，到閘5的某個輸入是錯誤的。

範例



3. 如果閘3 的輸出不是錯的，就是到閘5 的輸入 A 或 B 是錯的，因為 $C'D + CD' = 0$ ，所以閘3 的輸出是錯的。
4. 閘3 的輸出與閘1 和閘2 的輸出不一致，因為 $0 + 0 \neq 1$ 。因此，不是閘3 某個輸入連接錯誤，或是閘3 不良，就是到閘3 的某個輸入接腳是不良的。