第6章

多階閘電路／NAND和NOR閘

6.1 多階閘電路

- **階數**（level）：在一個電路的輸入和輸出之間所能串接的最大閘數稱為閘的階數。
  1. **AND-OR 電路**
  2. **OR-AND 電路**
  3. **OR-AND-OR 電路**
  4. **AND和OR閘電路**

6.1 多階閘電路

- 對於一個AND-OR 電路，我們可以利用分解其所導出的積項和表示式來增加它的階數。
- 對一個OR-AND 電路，我們可以乘開其所導出的和項積表示式的某些項目來增加它的階數。
- 一個電路的閘數、閘輸入數和階數可以由其相對應的表示式觀察得到。

6.1 多階閘電路

$$Z = (AB + C)[(D + E) + FG] + H$$
$$= AB(D + E) + C(D + E) + ABFG + CFG + H$$

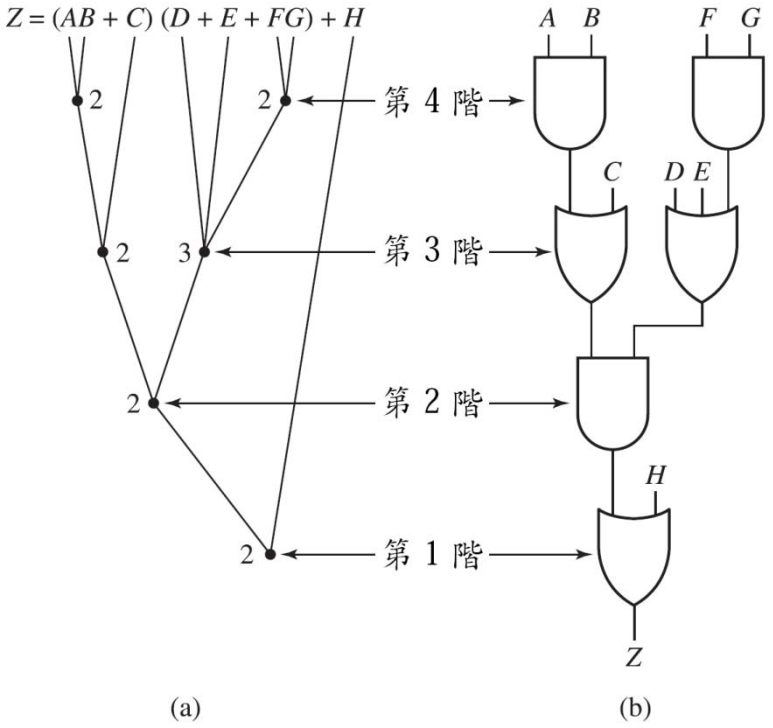


圖 6-1 實現 Z 的四階電路

在Z 表示式下面所畫的樹狀圖指出相對應的電路有四階、六個閘和13個閘輸入。

6.1 多階閘電路

❖最後的電路需要三階、六個閘和19個閘輸入。

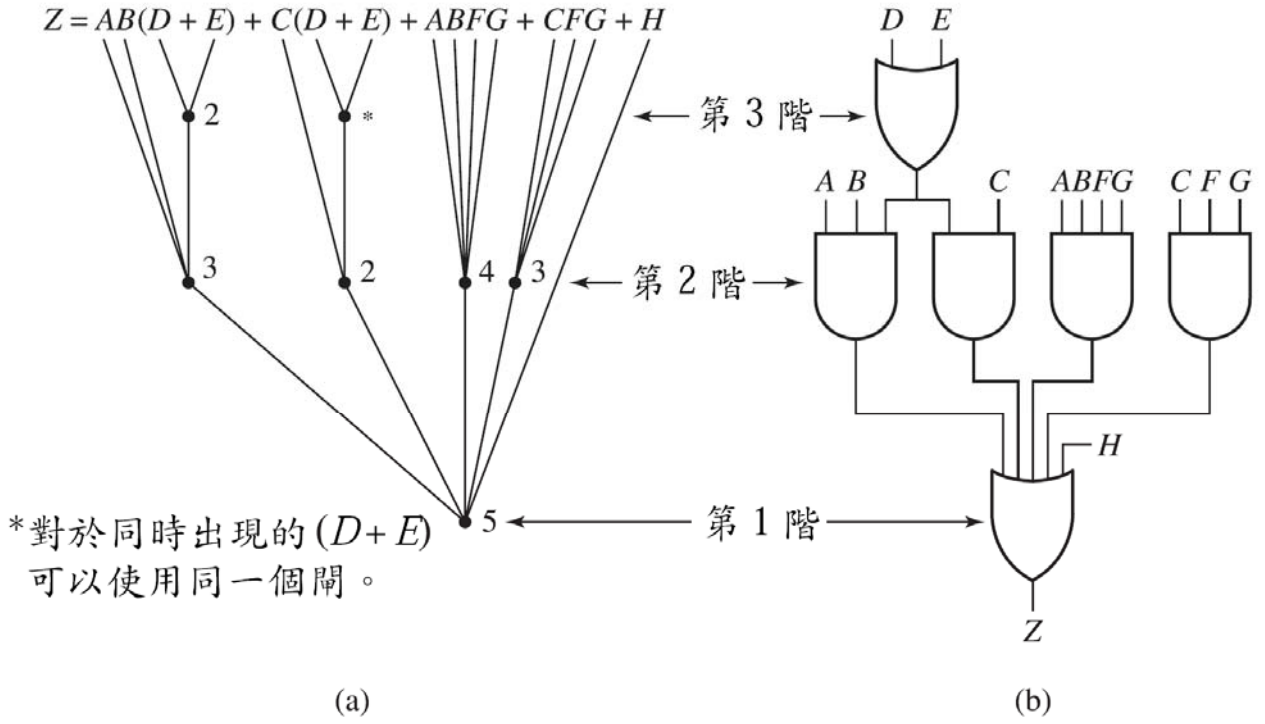


圖 6-2 實現 Z 的三階電路

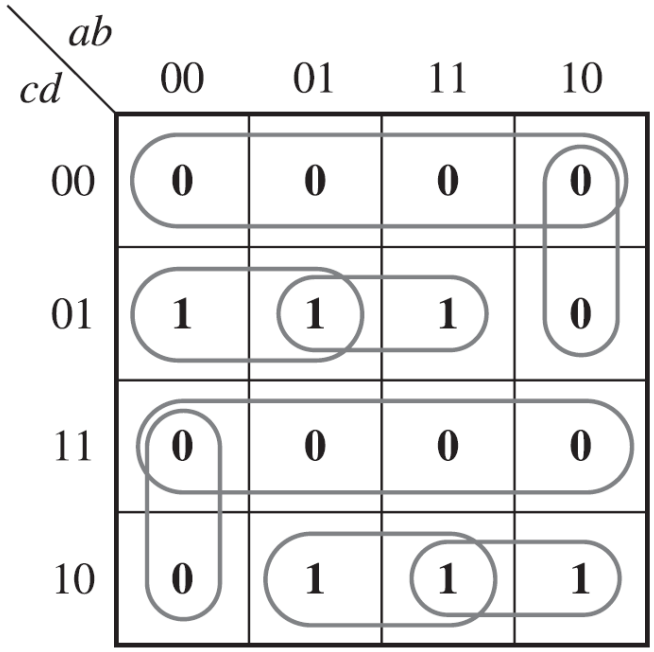
範例：利用AND 和OR閘設計多階的範例

❖ 問題：找出一個AND 和OR閘電路實現：

$$f(a,b,c,d) = \sum m(1,5,6,10,13,14)$$

範例：利用AND和OR閘設計多階的範例

❖解答：首先利用卡諾圖（圖6-3）化簡f：



$$f = a'c'd + bc'd + bcd' + acd' \qquad (6\text{-}1)$$

圖 6-3

範例：利用AND 和OR閘設計多階的範例

- 這可以直接得到一個二階的AND-OR閘電路（圖6-4）：

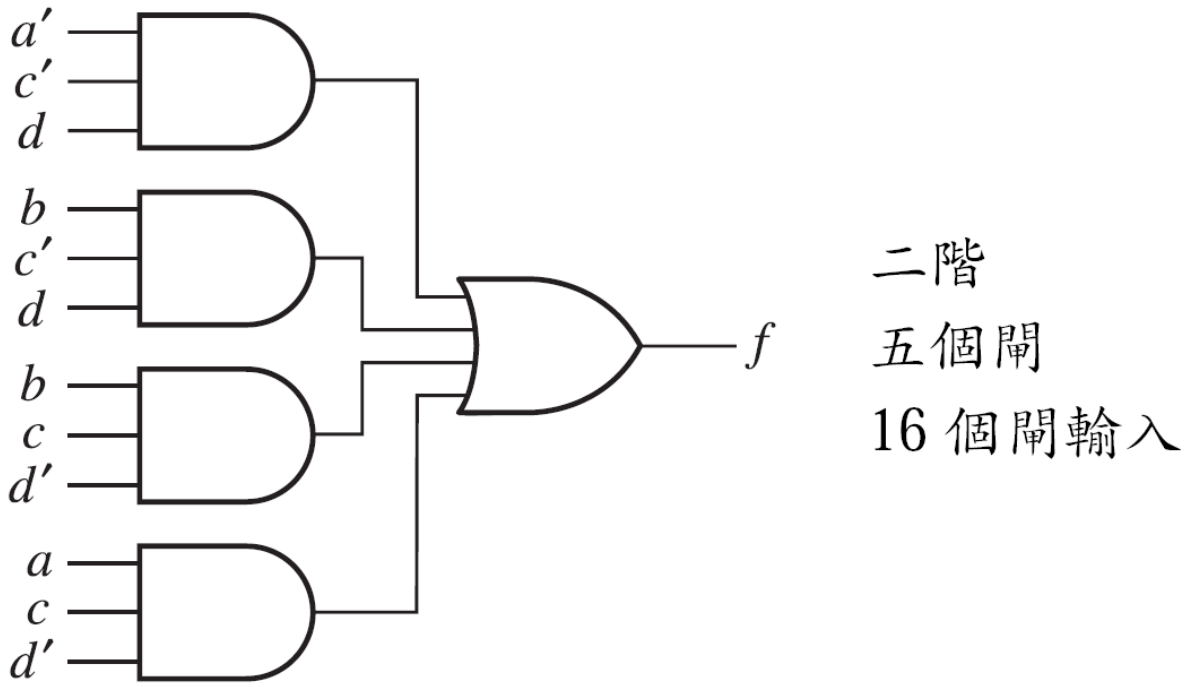


圖 6-4

範例：利用AND 和OR閘設計多階的範例

- 分解(6-1) 式得到：

$$f = c'd(a'+b)+cd'(a+b) \qquad (6\text{-}2)$$

- 從上式可以得到下面三階的OR-AND-OR閘電路（圖6-5）：

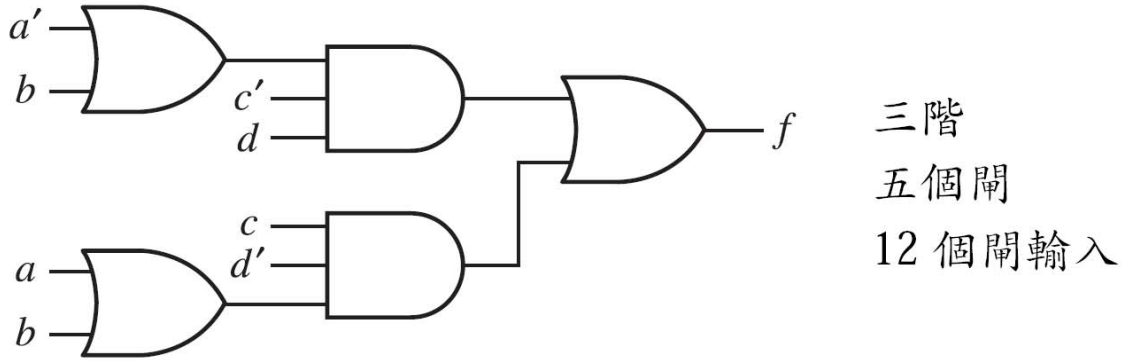


圖 6-5

範例：利用AND 和OR閘設計多階的範例

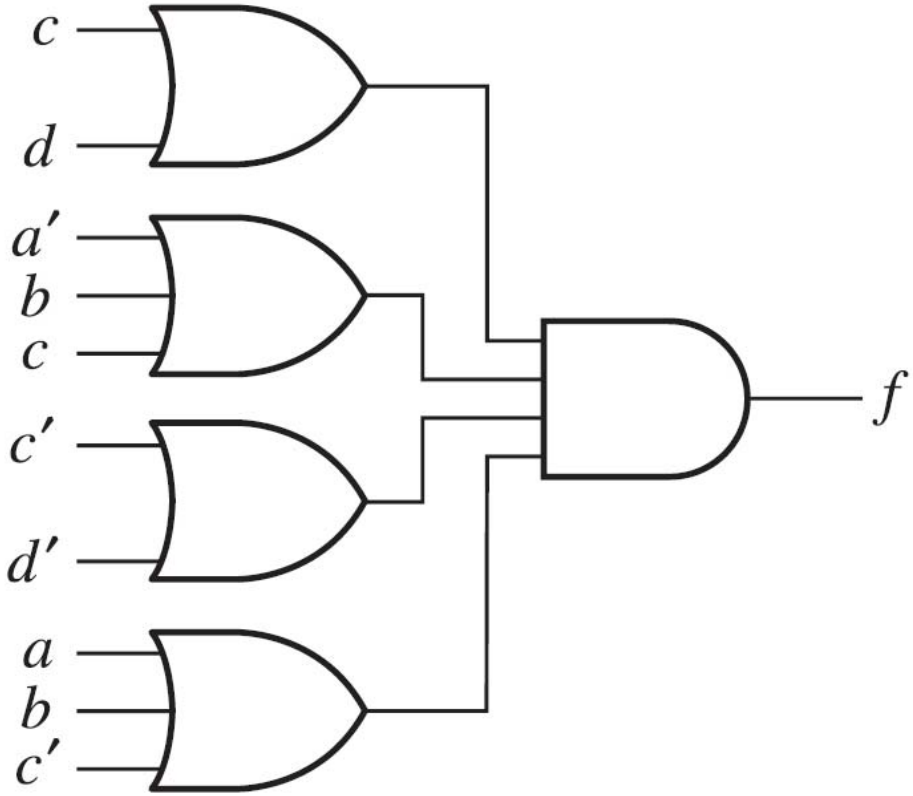
- 利用卡諾圖中0的部分得到如下所示的函數：

$$f' = c'd' + ab'c' + cd + a'b'c \qquad (6\text{-}3)$$

$$f = (c+d)(a'+b+c)(c'+d')(a+b+c') \qquad (6\text{-}4)$$

- 從(6-4) 式可以直接得到一個二階的OR-AND電路（圖6-6）。

範例：利用AND 和OR閘設計多階的範例



二階

五個閘

14 個閘輸入

圖 6-6

範例：利用AND和OR閘設計多階的範例

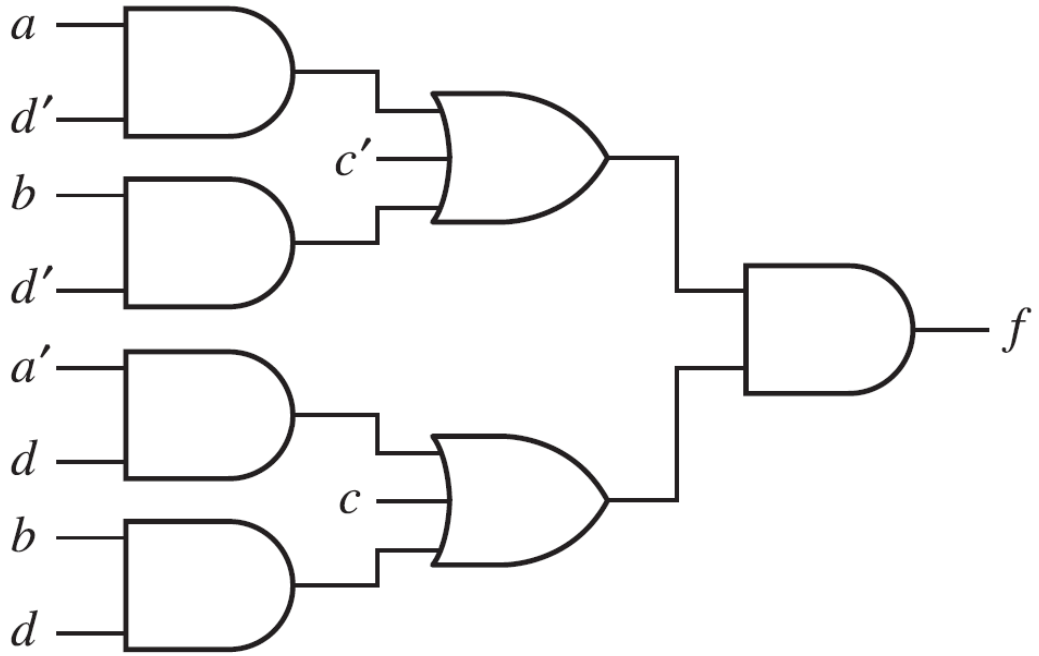
- 利用$(X + Y)(X + Z) = X + YZ$對(6-4)式作部分乘開，則可以得到

$$f = [c + d(a' + b)][c' + d'(a + b)] \qquad (6\text{-}5)$$

$$f = (c + a'd + bd)(c' + ad' + bd') \qquad (6\text{-}6)$$

範例：利用AND 和OR閘設計多階的範例

- 由此可以直接得到一個三階的AND-OR-AND電路（圖6-7）：



三階
七個閘
16 個閘輸入

圖 6-7

範例：利用AND 和OR閘設計多階的範例

- 對於這個特別的範例，最好的二階解是輸出閘為AND閘（圖6-6），且最好的三階解是輸出為OR 閘（圖6-5）。

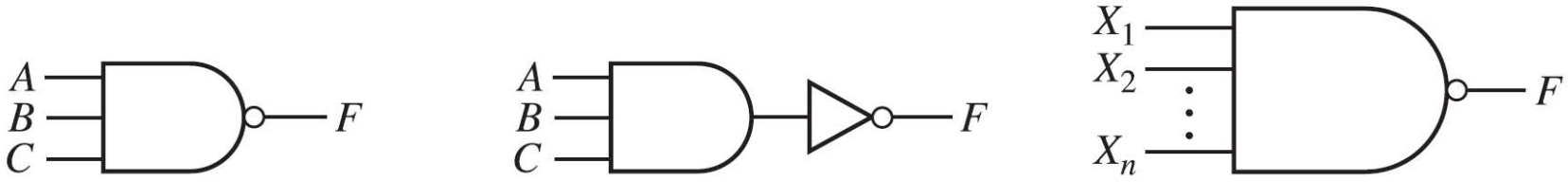
6.2 NAND和NOR閘

- NAND閘：一個AND閘後面接著一個反相器。
- 閘的輸出是：

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

- n 輸入NAND閘的輸出是：

$$F = (X_1 X_2 \ldots X_n)' = X_1' + X_2' + \ldots + X_n' \qquad (6\text{-}8)$$



(a) 三輸入 NAND 閘　(b) 等效 NAND 閘　(c) n 輸入 NAND 閘

圖 6-8 NAND 閘

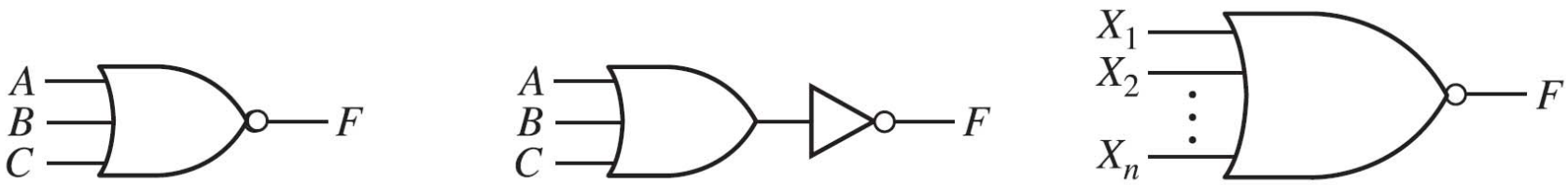
6.2 NAND和NOR閘

- NOR閘：一個OR 閘後面接著一個反相器。
- 閘的輸出為：

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

- n 輸入NOR閘的輸出為：

$$F = (X_1 + X_2 ... + X_n)' = X_1'X_2' \dots X_n' \qquad (6\text{-}9)$$



(a) 三輸入 NOR 閘　(b) 等效 NOR 閘　(c) n 輸入 NOR 閘

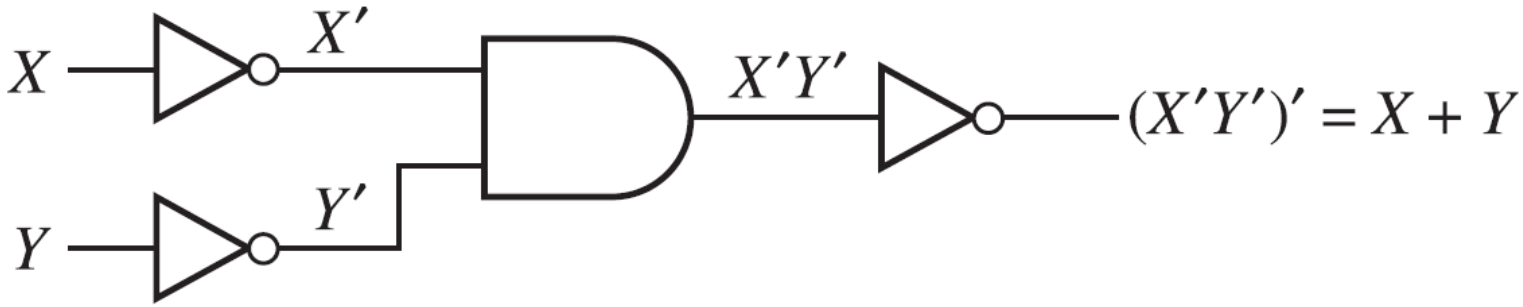
圖 6-9 NOR 閘

6.2 NAND和NOR閘

- 假如任何布林函數可以用一組邏輯運算來表示，則此邏輯運算組合稱為**功能完備**（functionally complete）。
- AND、OR 和NOT 組合明顯是功能完備，因為任何函數都可以被表示成積項和的形式，
- 假如所有的交換函數可以使用一組閘集合來實現，則此組邏輯閘集合為功能完備。
- 任何邏輯閘集合可以實現AND、OR和NOT者都是功能完備。

6.2 NAND和NOR閘

- AND和NOT是一組功能完備的閘集合，因為使用AND和NOT可以實現OR運算：



- 假如單一個閘本身形成一個功能完備的集合，則任何交換函數可以只使用這種型式的閘來實現。

6.2 NAND和NOR閘

- NOT、AND和OR可以只使用NAND 閘來實現。
- 同理，任何函數可以只使用NOR閘來實現。

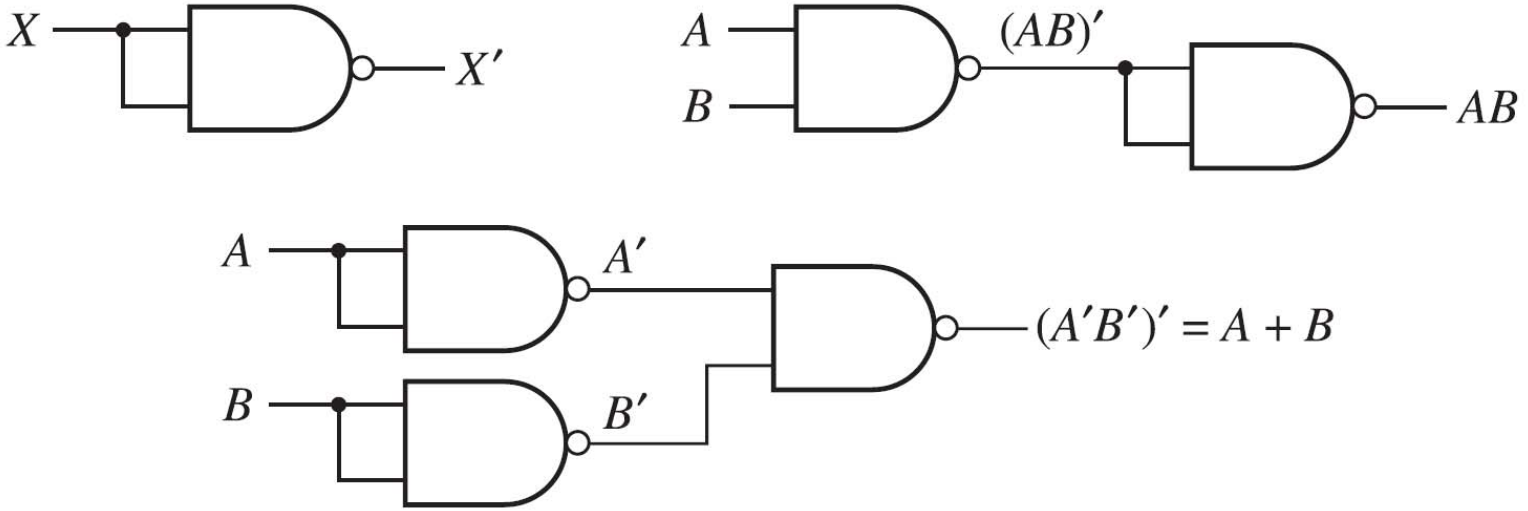


圖 6-10 用 NAND 閘來實現 NOT、AND 和 OR

6.2 NAND和NOR閘

- 決定一組已知的閘集合是否功能完備：
  - 首先，寫出由每個閘所實現函數的最簡積項和表示式。假如在表示式中沒有出現補數，則NOT 不能被實現，且此集合不是功能完備。
  - 假如在表示式中有補數出現，則通常可以由適當地選擇相對應閘的輸入來實現NOT。
  - 其次，嘗試實現AND或OR，牢記現在可以用NOT。只要AND 或OR 其中之一可以實現，雖然沒有其他明顯直接的方法，則通常可以利用笛摩根定律來實現另外一個。例如，假如可以使用OR 和NOT，則可以由$XY = (X' + Y')'$來實現AND。

6.3 二階NAND和NOR閘電路的設計

- 一個包含AND 和OR 閘的二階電路可以很容易地轉換成由NAND 閘或NOR 閘所組成的二階電路。這個轉換可以由$F = (F')'$，然後應用笛摩根定律來實現：

$$(X_1 + X_2 + \cdots + X_n)' = X_1'X_2'\ldots X_n' \qquad (6\text{-}11)$$

$$(X_1X_2\ldots X_n)' = X_1' + X_2' + \cdots + X_n' \qquad (6\text{-}12)$$

6.3 二階NAND和NOR閘電路的設計

❖ 下面的例子說明一個最簡積項和形式轉換成一些其他二階形式的過程：

$$F = A + BC' + B'CD = [(A + BC' + B'CD)']' \quad (6\text{-}13)$$

$$= [A' \cdot (BC')' \cdot (B'CD)']' \quad (由\ (6\text{-}11)\ 式) \quad (6\text{-}14)$$

$$= [A' \cdot (B' + C) \cdot (B + C' + D')]' \quad (由\ (6\text{-}12)\ 式) \quad (6\text{-}15)$$

$$= A + (B' + C)' + (B + C' + D')' \quad (由\ (6\text{-}12)\ 式) \quad (6\text{-}16)$$

(6-13) 式、(6-14) 式、(6-15) 式和(6-16) 式分別代表AND-OR、NANDNAND、OR-NAND 和NOR-OR形式，如圖6-11所示。

6.3 二階NAND和NOR閘電路的設計

- 由卡諾圖得到最簡和項積之後，F 可以被寫成下列的二階形式：

$$F=(A+B+C)(A+B'+C')(A+C'+D) \quad (6\text{-}18)$$

$$=\{[(A+B+C)(A+B'+C')(A+C'+D)]'\}'$$

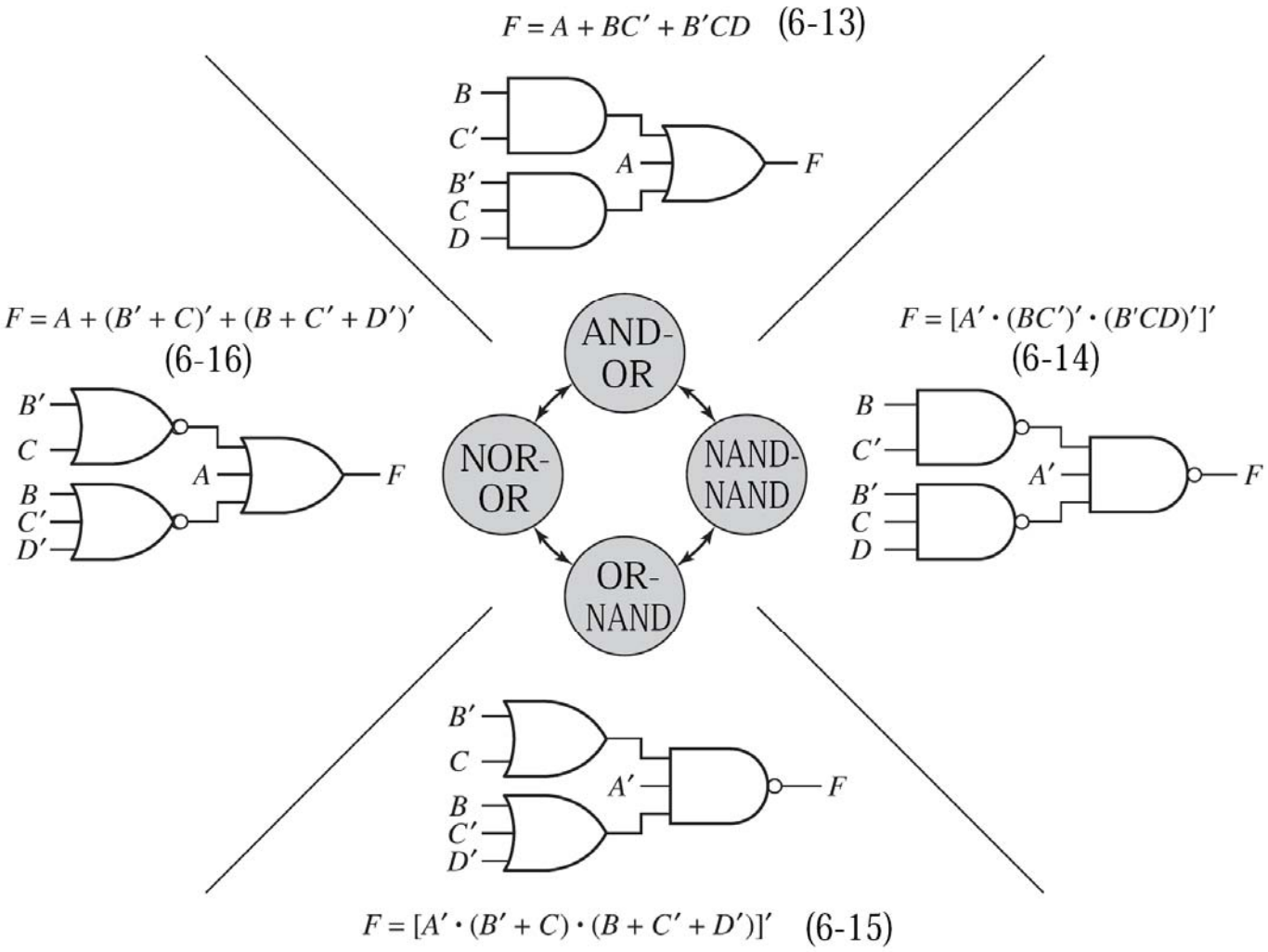
$$=[(A+B+C)'+(A+B'+C')'+(A+C'+D)']' \quad (由\ (6\text{-}12)\ 式) \quad (6\text{-}19)$$

$$=(A'B'C'+A'BC+A'CD')' \quad (由\ (6\text{-}11)\ 式) \quad (6\text{-}20)$$

$$=(A'B'C')'\cdot(A'BC)'\cdot(A'CD')' \quad (由\ (6\text{-}11)\ 式) \quad (6\text{-}21)$$

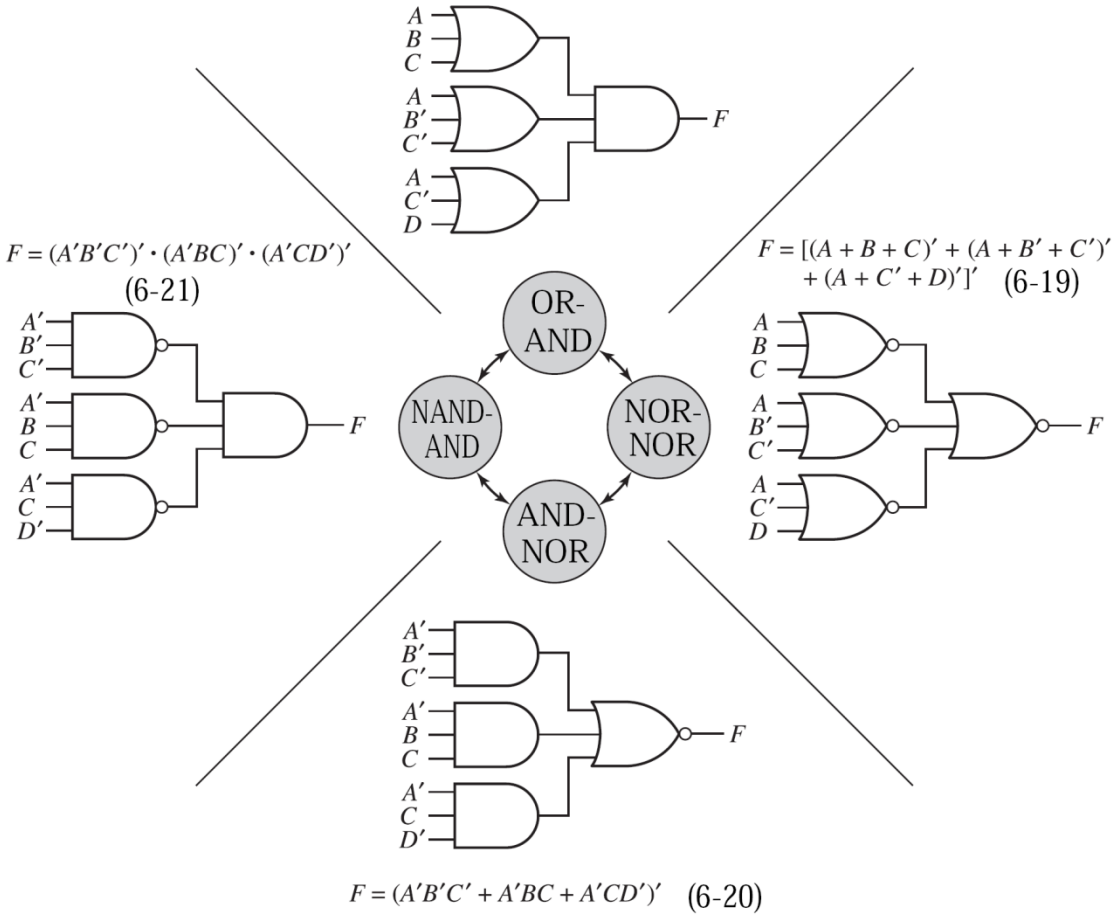
- (6-18) 式、(6-19) 式、(6-20) 式和(6-21) 式分別表示OR-AND、NOR-NOR、AND-NOR 和NAND-AND 形式，如圖6-11所示。

6.3 二階NAND和NOR閘電路的設計



6.3 二階NAND和NOR閘電路的設計

$$F = (A + B + C)(A + B' + C')(A + C' + D) \quad (6\text{-}18)$$



$F = (A'B'C')' \cdot (A'BC)' \cdot (A'CD')'$ (6-21)

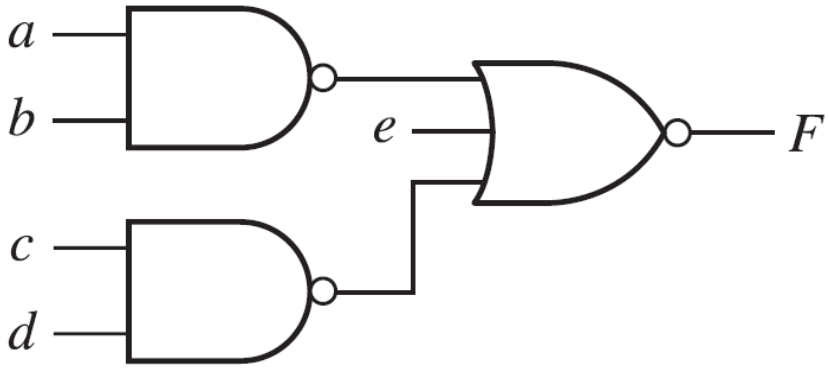
$F = [(A + B + C)' + (A + B' + C')' + (A + C' + D)']'$ (6-19)

$$F = (A'B'C' + A'BC + A'CD')' \quad (6\text{-}20)$$

圖 6-11 二階電路的八個基本形式

6.3 二階NAND和NOR閘電路的設計

- 其他八個可能的二階形式（AND-AND、OR-OR、OR-NOR、ANDNAND、NAND-NOR、NOR-NAND 等）是退化的形式，它們不能實現所有的交換函數。考慮下列NAND-NOR電路的例子：



$$F = [(ab)' + (cd)' + e]' = abcde'$$

- NAND-NOR 形式只能形成文字字元的積而不是積項和。

6.3 二階NAND和NOR閘電路的設計

- 設計一個最簡的二階NAND-NAND電路的步驟：
  1. 求出F的最簡積項和表示式。
  2. 畫出相對應的二階AND-OR電路。
  3. 以NAND 閘取代所有的閘，且閘與閘之間的連接方式保持不變。假如輸出閘有任何單一文字字元作為輸入，則將文字字元取補數。

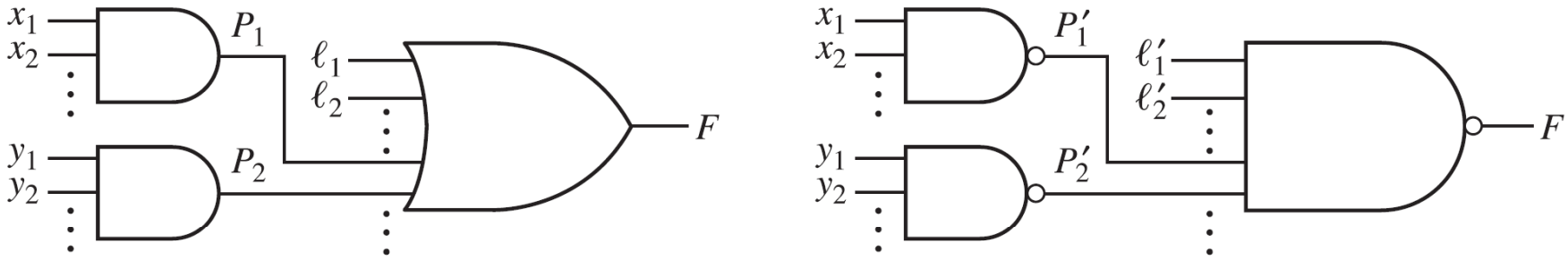
6.3 二階NAND和NOR閘電路的設計

- 圖6-12說明步驟3的轉換方式

$$F = \ell_1 + \ell_2 + \cdots P_1 + P_2 + \cdots$$

- 應用笛摩根定理之後，

$$F = (\ell_1'\ell_2'\cdots P_1'P_2'\cdots)'$$



(a) 轉換前　　(b) 轉換後

圖 6-12　AND-OR 到 NAND-NAND 的轉換

6.3 二階NAND和NOR閘電路的設計

❖設計一個最簡的二階NOR-NOR電路的步驟：

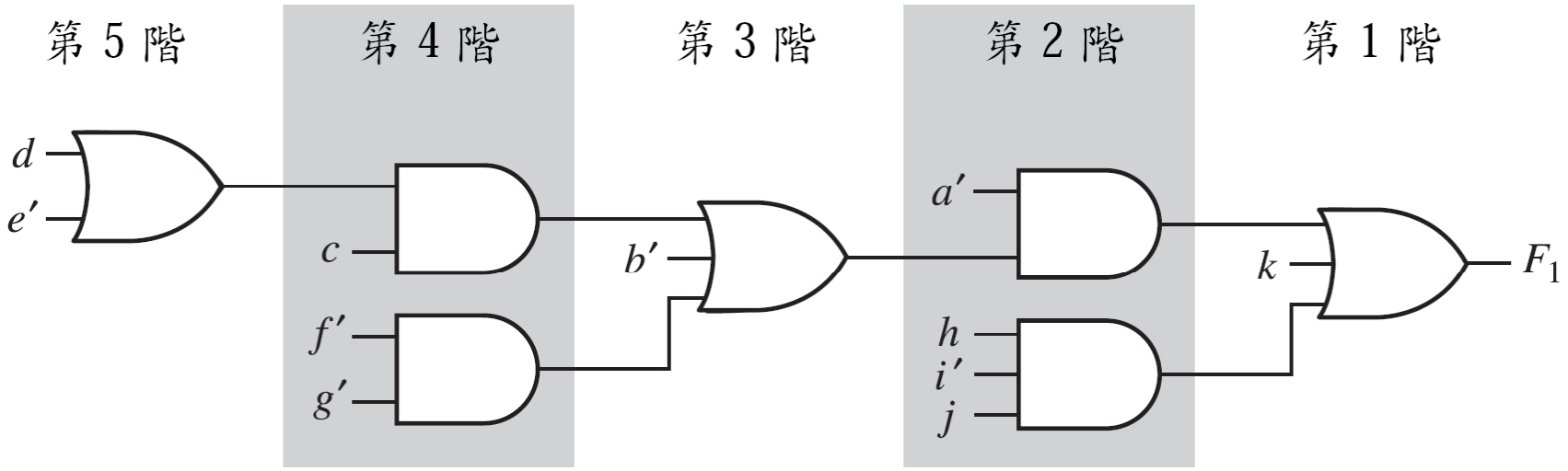
1. 求出F的最簡和項積表示式。
2. 畫出其相對應的二階OR-AND電路。
3. 用NOR 閘取代所有的閘，且閘與閘之間的接連方式保持不變。假如輸出閘上有任何單一的文字字元作為輸入，則將文字字元取補數。

6.4 多階NAND和NOR閘電路的設計

- 下面的步驟可以用來設計多階NAND閘電路：
  1. 化簡所要實現的交換函數。
  2. 設計一個多階的AND 和OR 閘電路。輸出閘必須是OR 閘。AND 閘的輸出不能作為AND 閘的輸入；OR 閘的輸出不能作為OR 閘的輸入。
  3. 由輸出閘當作第1階開始作階層編號。以NAND閘取代所有的閘，閘與閘之間的連接方式保持不變。且第2、4、6... 階的輸入保持不變，而將第1、3、5... 階的文字字元輸入取反相。

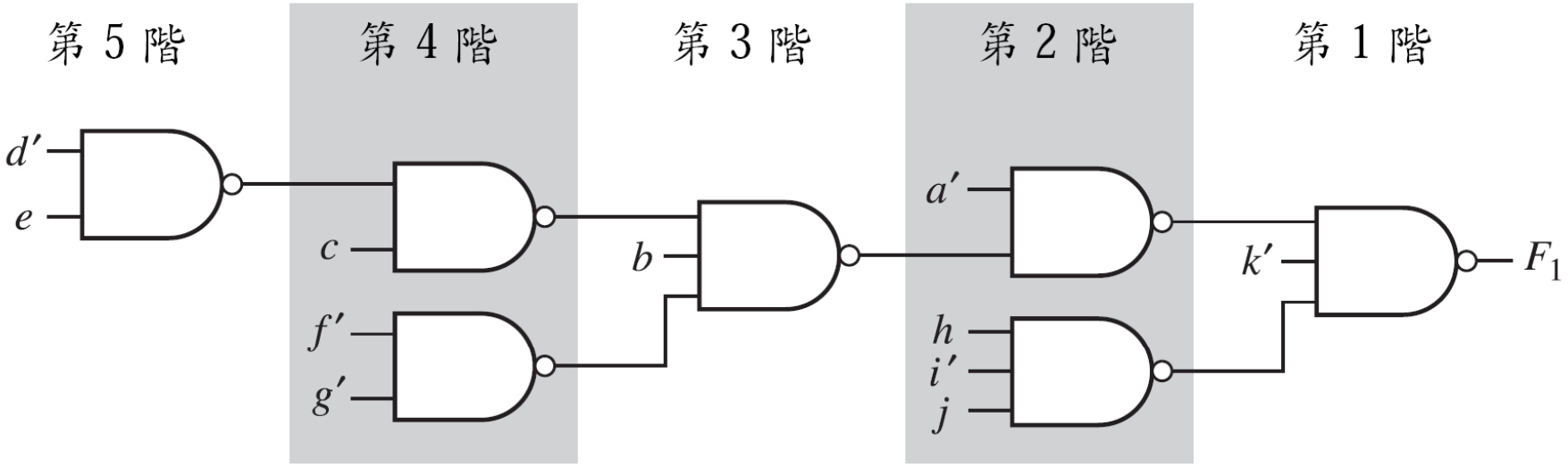
範例

$$F_1 = a'[b' + c(d + e') + f'g'] + hi'j + k$$



(a) AND-OR 網路

範例



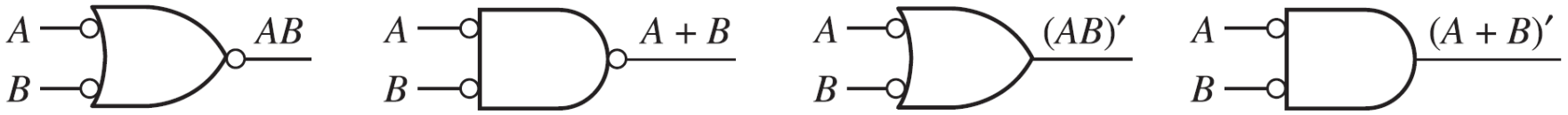
(b) NAND 網路

圖 6-13 多階電路轉換成 NAND 閘

6.5 使用其他閘符號作電路轉換

❖ 反相器:

❖ AND、OR、NAND和NOR閘：



$AB = (A' + B')'$ (a) AND

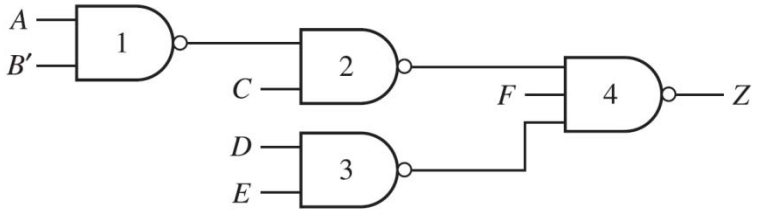
$A + B = (A'B')'$ (b) OR

$(AB)' = A' + B'$ (c) NAND

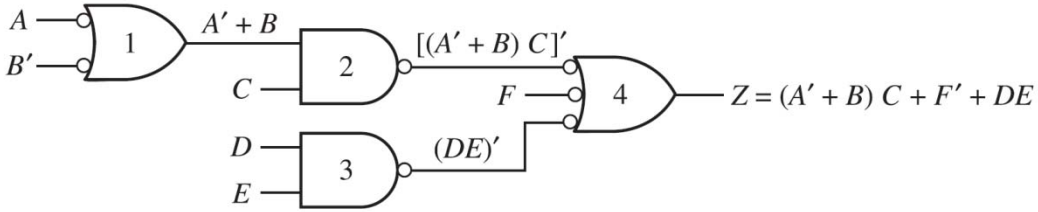
$(A + B)' = A'B'$ (d) NOR

圖 6-14 另一種的閘符號

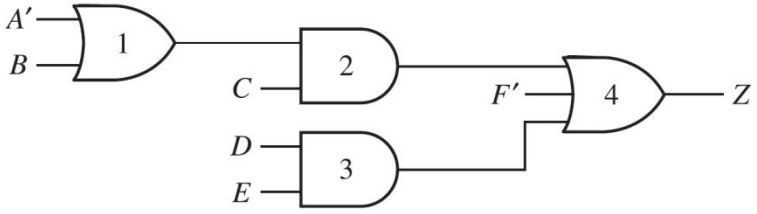
6.5 使用其他閘符號作電路轉換



(a) NAND 閘網路



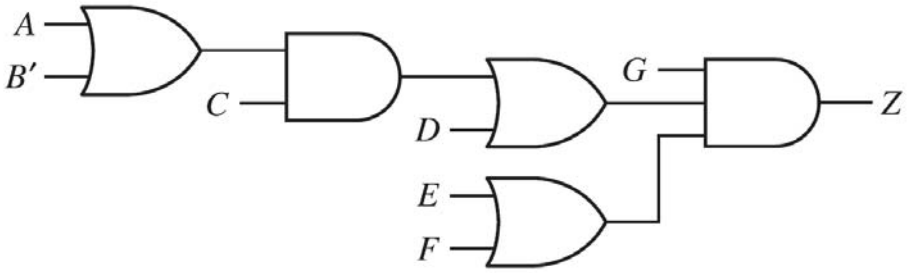
(b) NAND 閘網路的另一種形式



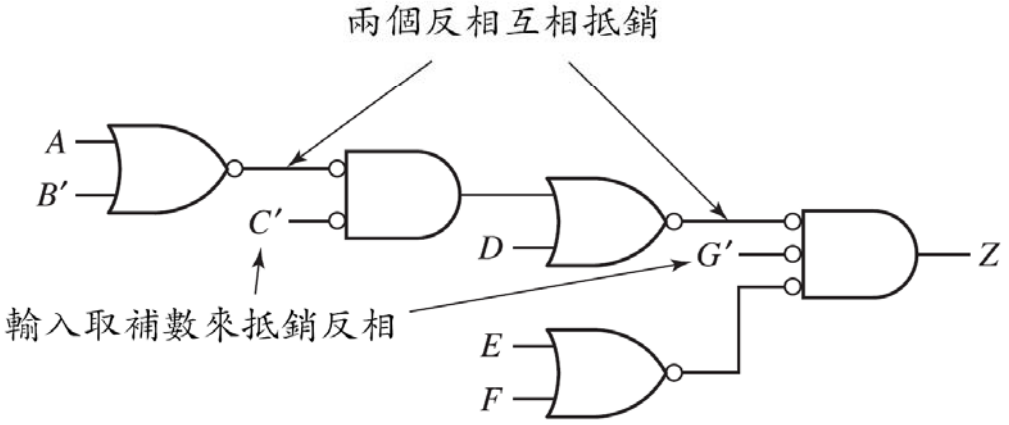
(c) 等效的 AND-OR 網路

圖 6-15 NAND 閘電路的轉換

6.5 使用其他閘符號作電路轉換



(a) 具有 OR 和 AND 閘的電路



(b) 等效 NOR 閘電路

圖 6-16 NOR 閘的轉換

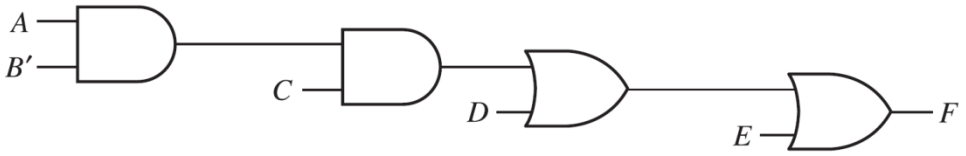
6.5 使用其他閘符號作電路轉換

- 即使AND 和OR 閘沒有交替，我們仍然可以轉換一個AND-OR 電路成為一個NAND 或NOR 電路，下面的步驟可以用來轉換成為一個NAND（或NOR）電路：
  1. 利用在AND 閘的輸出加上反相小圓，將所有AND 閘轉換成NAND閘。利用在OR 閘的輸入加上反相小圓，將所有OR 閘轉換成NAND閘。（要轉換成NOR，則要在所有OR 閘的輸出及所有AND 閘的輸入加入反相小圓。）

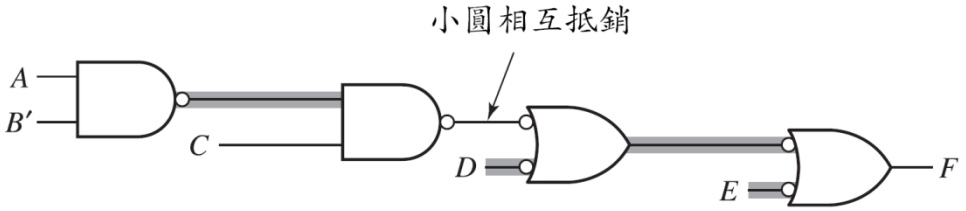
6.5 使用其他閘符號作電路轉換

2. 當一個反相輸出推動一個反相輸入時，因為兩個反相互相抵銷，所以無須額外的動作。
3. 當一個非反相閘輸出推動一個反相閘輸入或相反時，則加入一個反相器使得小圓被消掉。（視需要選擇一個在輸入或輸出有小圓的反相器。）
4. 當一個變數推動一個反相輸入時，則將變數取補數（或加入一個反相器）使得補數消掉在輸入端的反相。

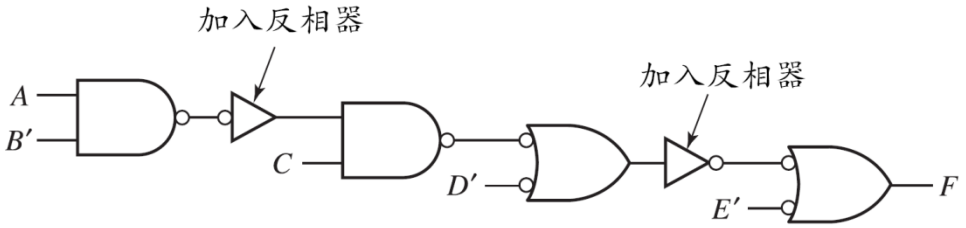
6.5 使用其他閘符號作電路轉換



(a) AND-OR 網路



(b) NAND 轉換的第一個步驟



(c) 完成轉換

圖 6-17 AND-OR 電路到 NAND 閘的轉換

6.5 使用其他閘符號作電路轉換

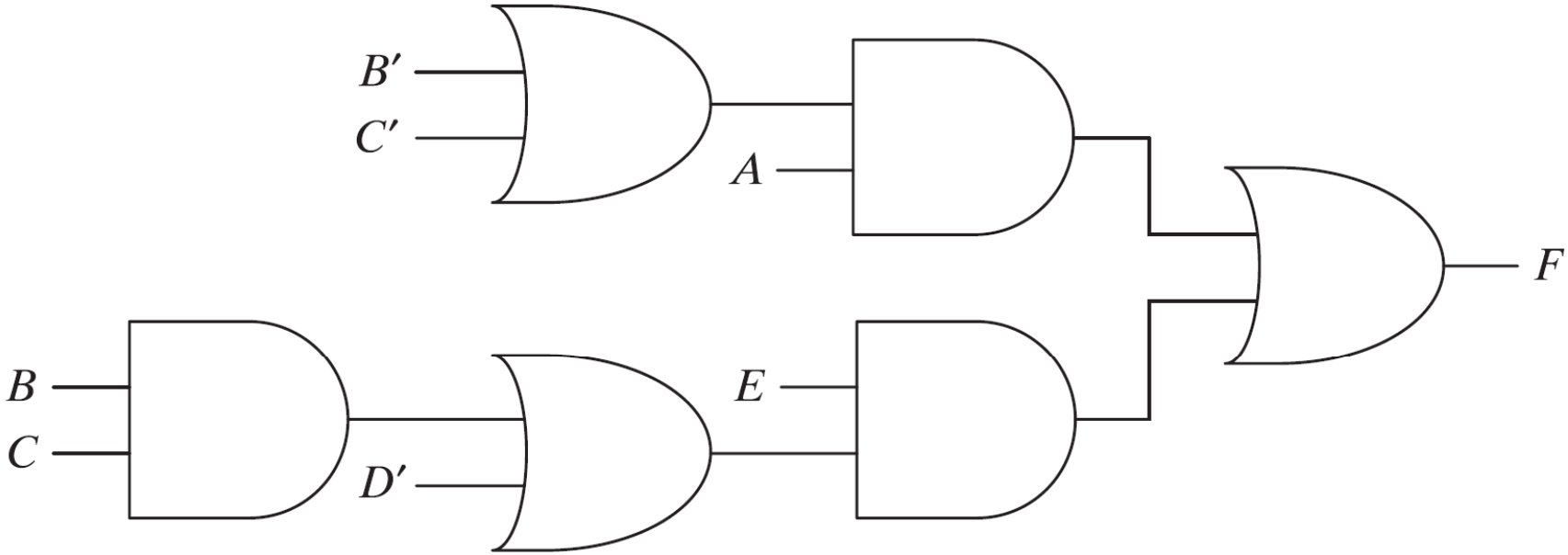


圖 6-18 有限的扇入數電路

6.5 使用其他閘符號作電路轉換

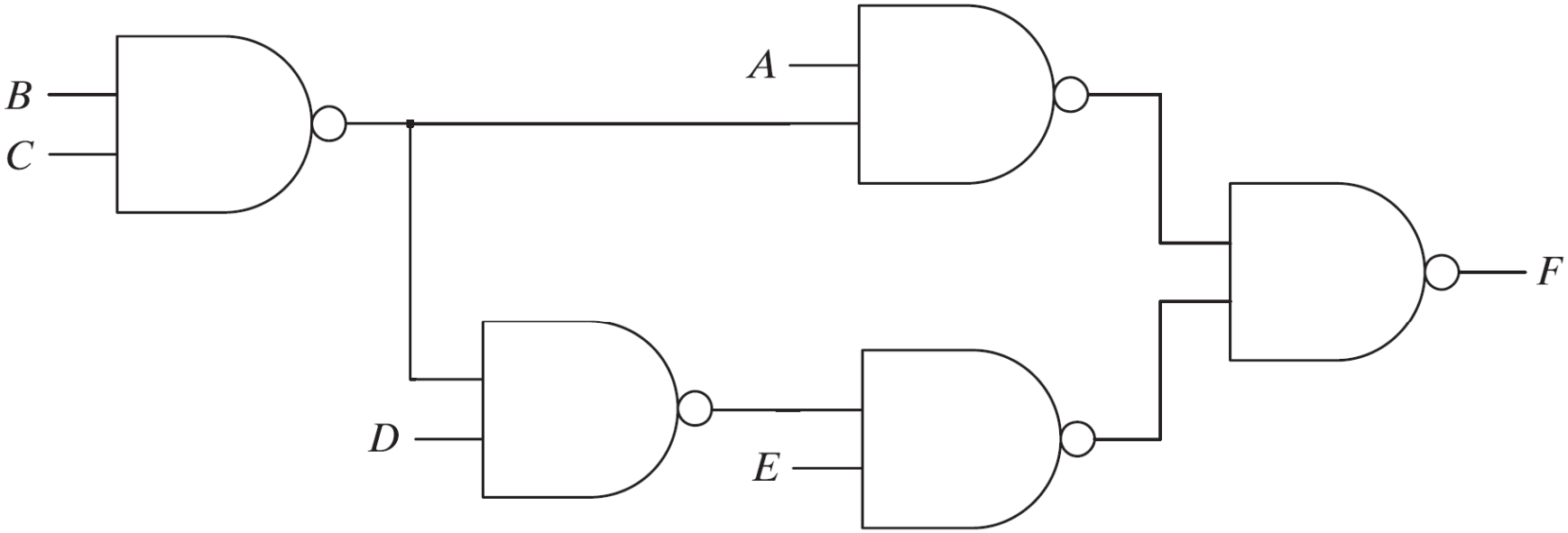


圖 6-19　圖 6-18 等效的 NAND 閘電路

6.6 二階多重輸出電路的設計

- 使用四個輸入和三個輸出設計一個電路實現下列函數：

$$F_1(A,B,C,D) = \sum m(11,12,13,14,15)$$
$$F_2(A,B,C,D) = \sum m(3,7,11,12,13,15)$$
$$F_3(A,B,C,D) = \sum m(3,7,12,13,14,15) \qquad (6\text{-}24)$$

- 首先，每個函數可以被個別實現，卡諾圖、函數及最後所得的電路列在圖6-20和圖6-21中，這個電路需要九個閘和21個閘輸入。

6.6 二階多重輸出電路的設計

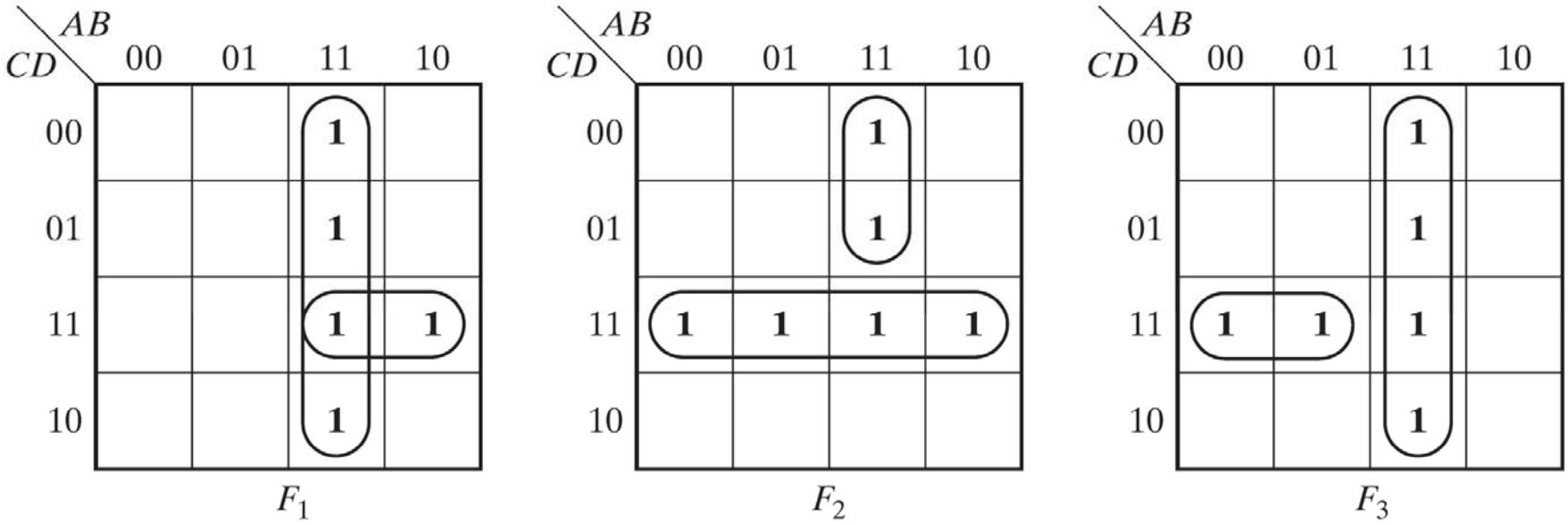


圖 6-20 (6-24) 式的卡諾圖

6.6 二階多重輸出電路的設計

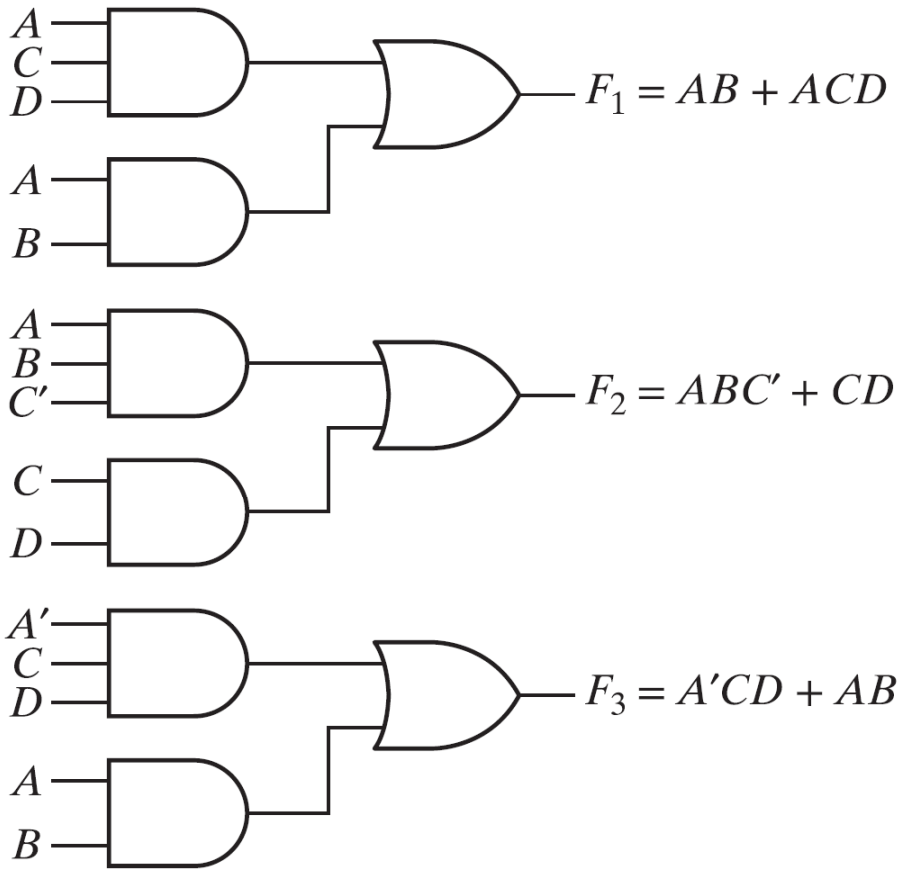


圖 6-21 (6-24) 式的電路

6.6 二階多重輸出電路的設計

- 在F_1及F_3中對AB使用同一個閘，且將F_2中的CD用$A'CD+ACD$來取代，則電路可以簡化為七個閘和18個閘輸入。

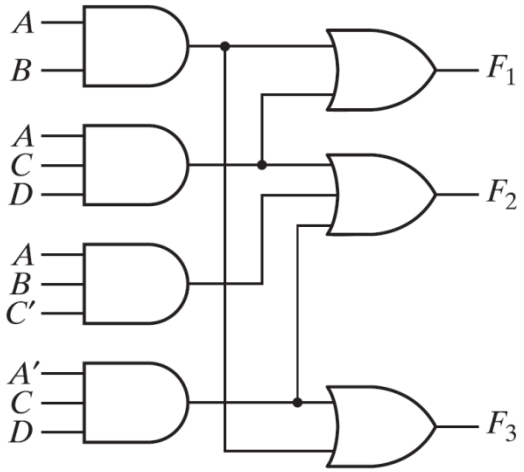


圖 6-22 (6-24) 式的多重輸出電路

- 在實現多重輸出電路，對每個函數使用最簡質含項的和並不一定會使整個電路合乎最經濟的原則。

6.6 二階多重輸出電路的設計

❖設計一個四輸入、三輸出的電路實現下列函數。

$$f_1 = \sum m(2,3,5,7,8,9,10,11,13,15)$$
$$f_2 = \sum m(2,3,5,6,7,10,11,14,15)$$
$$f_3 = \sum m(6,7,8,9,13,14,15) \qquad (6\text{-}25)$$

6.6 二階多重輸出電路的設計

- 首先，我們畫出f_1、f_2和f_3的圖（圖6-23）。假如每個函數被分別化簡，所得的結果為：

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

$$f_2 = c + a'bd$$

$$f_3 = bc + ab'c' + \left\{ \begin{array}{c} abd \\ 或 \\ ac'd \end{array} \right\} \begin{array}{l} 十個閘 \\ 25個閘輸入 \end{array} \qquad [6\text{-}25(a)]$$

6.6 二階多重輸出電路的設計

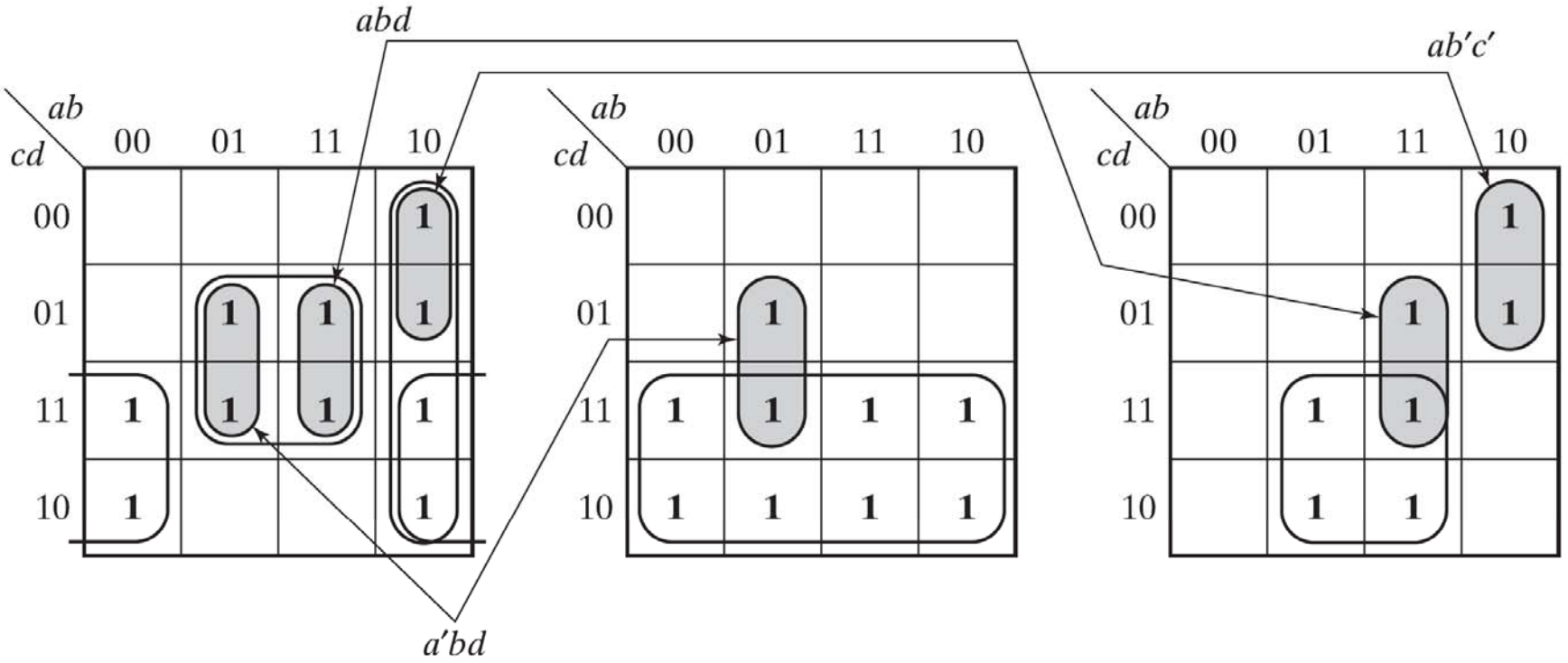


圖 6-23

6.6 二階多重輸出電路的設計

- 觀察這些圖之後，我們發現$a'bd$（從f_2）、abd（從f_3）及（從f_3）可以用在f_1中。假如用$a'bd + abd$取代bd，則可以省去需要實現bd的閘數。因為m_{10}和m_{11}在f_1中已經被$b'c$包含，$ab'c'$（從f_3）可以用來包含m_8和m_9，且可以消去需要實現ab'的閘。因此，最簡解是：

$$f_1 = \underline{a'bd} + \underline{abd} + \underline{ab'c'} + b'c$$

$$f_2 = c + \underline{a'bd}$$

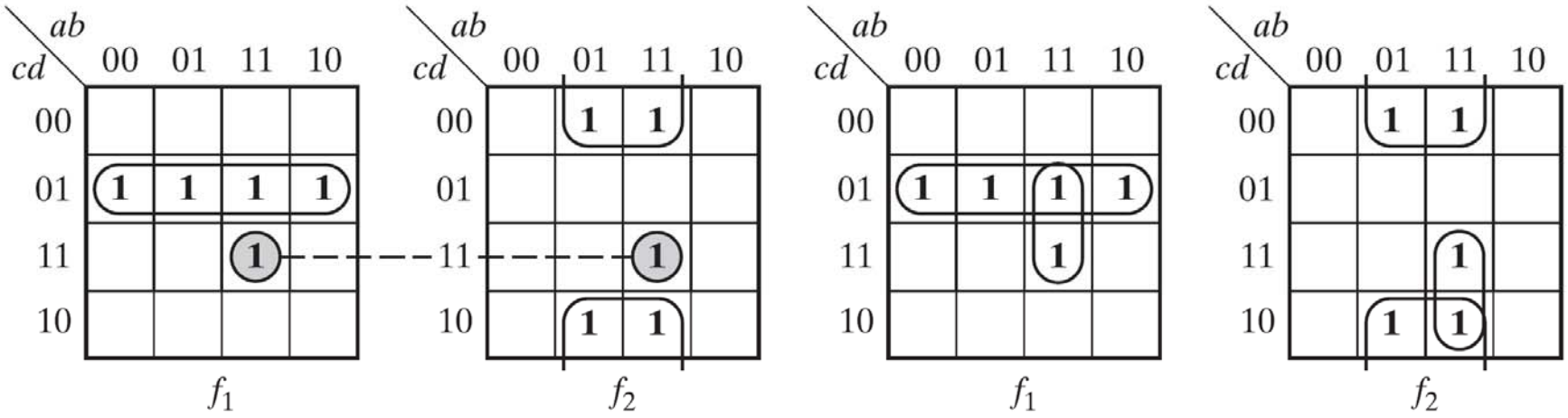
八個閘

$$f_3 = bc + \underline{ab'c'} + \underline{abd}$$

22個閘輸入　[6-25(b)]

6.6 二階多重輸出電路的設計

- 當設計多重輸出電路時，有時候最好不要將1 與相鄰的1 合併，參考在圖6-24的例子可以說明。

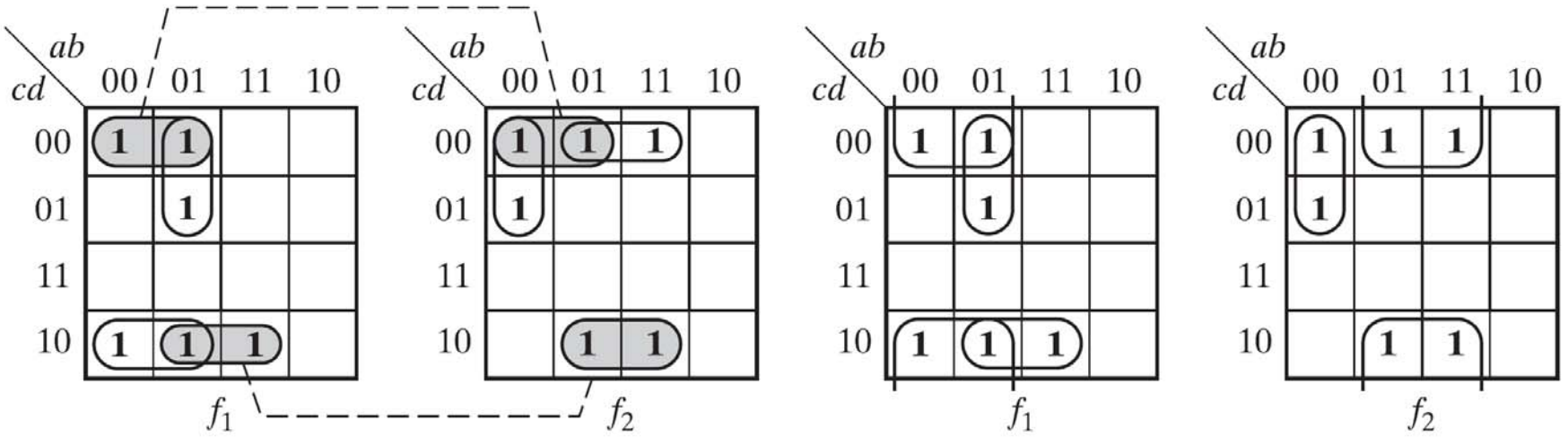


(a) 最佳解　　(b) 需要額外閘的解

圖 6-24

6.6 二階多重輸出電路的設計

- 共同項最多的解不一定是最好的解，參考在圖6-25的例子可以說明。



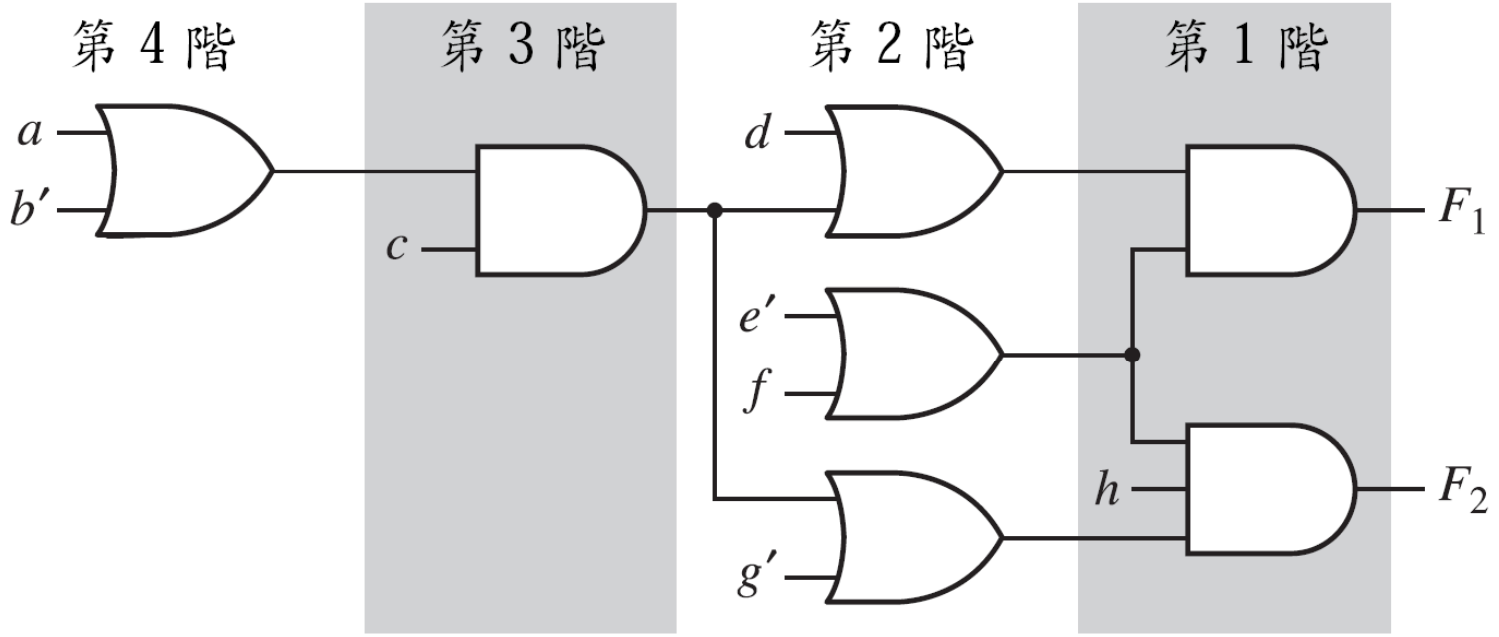
(a) 共同項最多的解
需要八個閘、26 個閘輸入

(b) 最佳解需要七個閘、
18 個閘輸入且沒有共同項

圖 6-25

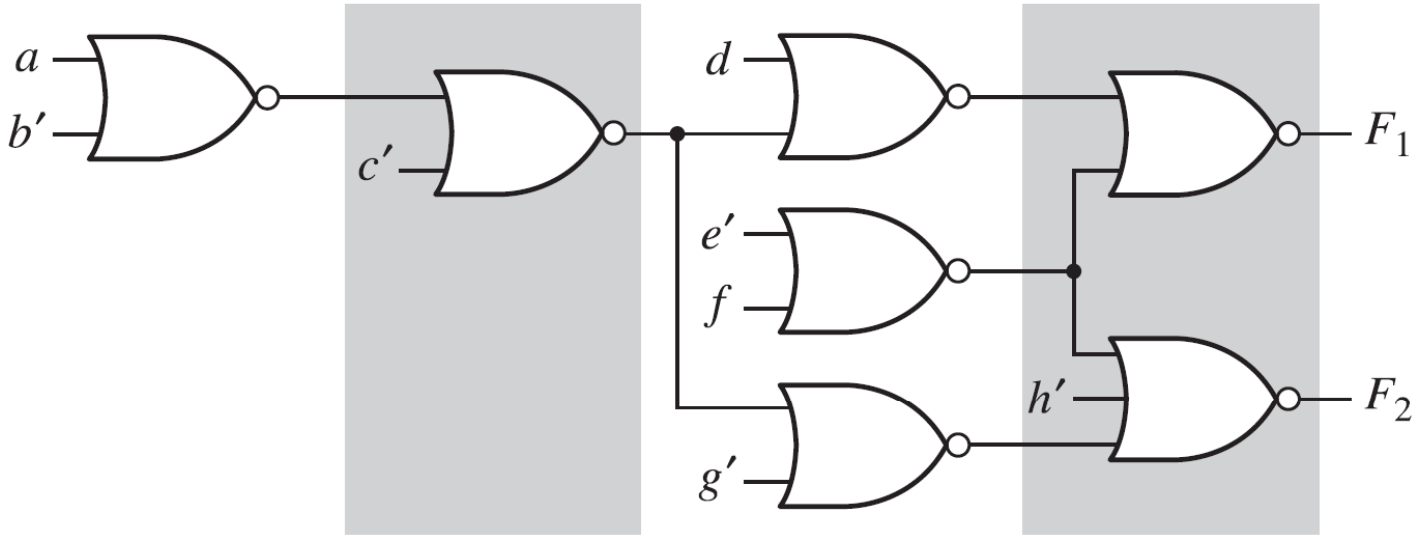
6.7 多重輸出NAND和NOR閘電路

$$F_1 = [(a + b')c + d](e' + f) \qquad F_2 = [(a + b')c + g'](e' + f)h$$



(a) NAND 和 OR 閘網路

6.7 多重輸出NAND和NOR閘電路



(b) NOR 網路

圖 6-26 多階電路轉換成 NOR 閘