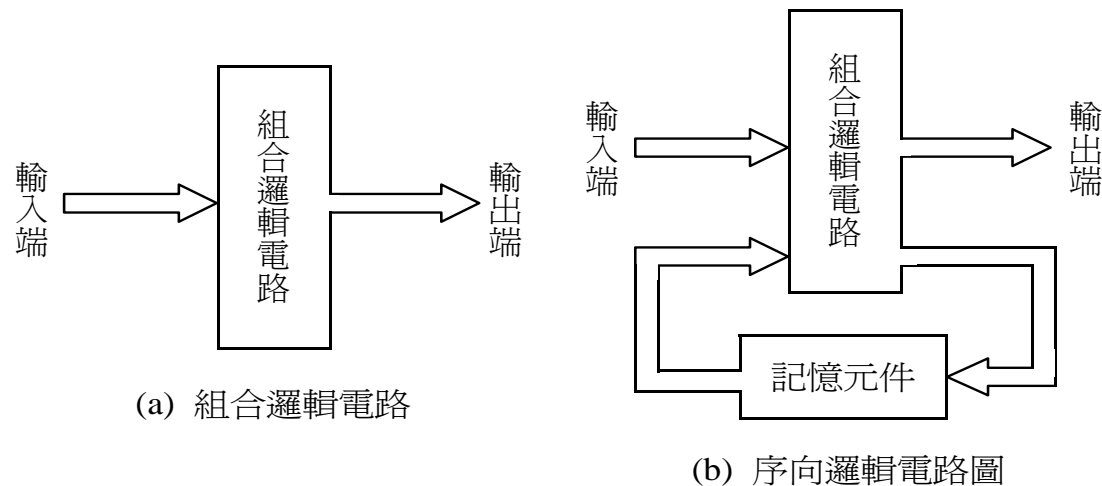




組合邏輯電路之分析與設計

概 述

- ◆ **數位電路** (Digital Circuits) 可分成**組合邏輯電路** (Combinational Logic Circuit) 與**序向邏輯電路** (Sequential Logic Circuit) 兩大類。
- ◆ **組合邏輯電路**是由**基本邏輯閘** (Logic Gates) 所組成的，此種電路之**輸出是所有輸入的組合**，即**輸出狀態**可完全由**目前之輸入狀態**來決定；**序向邏輯電路**是由**基本邏輯閘**與**記憶元件** (Memory Elements) 所組成的，此種電路有記憶元件存在，故輸出不僅需考慮**目前之輸入狀態**，且亦與**上一個輸出狀態**有關，因此序向電路之輸出狀態與時間有關。



- ◆ 本學期將分三個部分來討論**組合邏輯電路之設計**，首先討論**組合邏輯電路之基本觀念、設計技巧與時序分析**後，再討論**算數運算與資料處理**等兩種典型之組合邏輯電路的設計方法。

組合邏輯電路之基本觀念

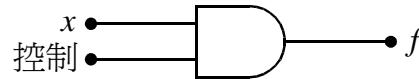
- ◆ 組合邏輯電路包含 n 個輸入變數 $x_{n-1}、\dots、x_1、x_0$ ，一組基本邏輯閘所組成之電路與 m 個輸出變數 $f_{m-1}、\dots、f_1、f_0$ 。當電路之輸入端 (Input Port) 接受二元性訊號 (Binary Signal) 時，這些二元性訊號經過組合邏輯閘電路之運算後，最後於輸出端 (Output Port) 產生實際所需之二元性訊號，如下圖所示。



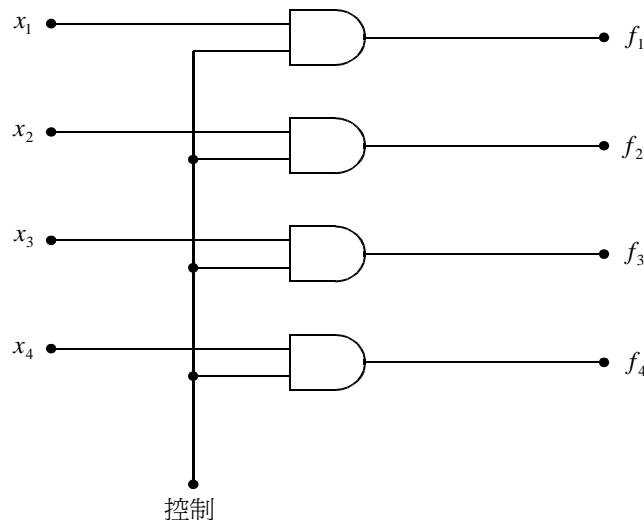
- ◆ 對 n 個變數之布林函數而言，因每個變數皆可以補數 (Complement) 或非補數 (Noncomplement) 之型態出現，故可產生 2^n 種變數組合。因此對組合邏輯電路而言，若電路有 n 個輸入變數，將可產生 2^n 種可能之二元性 (二進位) 輸入組合。對輸入變數之任何一個組合，皆可對應至一種可能的輸出值，故組合邏輯電路可以用真值表 (True Table) 來加以定義，即真值表可用來描述所有可能輸入變數組合與輸出間之關係。
- ◆ 具有 n 個輸入與 m 個輸出之組合邏輯電路可用 m 個布林函數來描述，即每一個輸出函數，皆可使用不多於 2^n 個輸入變數組合之布林函數來表示。

基本邏輯閘之應用(AND 閘)

- ◆ **邏輯閘** (Logic Gates) 為構成數位電路之基本元件，故應先瞭解**邏輯閘之簡單應用**後，再進一步**設計組合邏輯電路** (Combinational Logic Circuit)，以加強讀者對基本邏輯閘功能之認識。
- ◆ 使用 **AND 閘** 做**低態禁制** (Inhibit) 電路：當「控制 = 1」時， $f = x$ (輸入訊號 x **可通過** AND 閘至輸出端)；若「控制 = 0」， $f = 0$ (輸入訊號 x **無法通過** AND 閘至輸出端)。

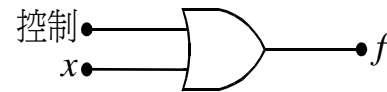


- ◆ 使用一個 **AND 閘** 僅能做一個位元之低態禁制電路，若欲對 n 個位元進行低態禁制，則只要使用 n 個 **AND 閘** 即可，而 4 位元之低態禁制電路，如下圖所示。

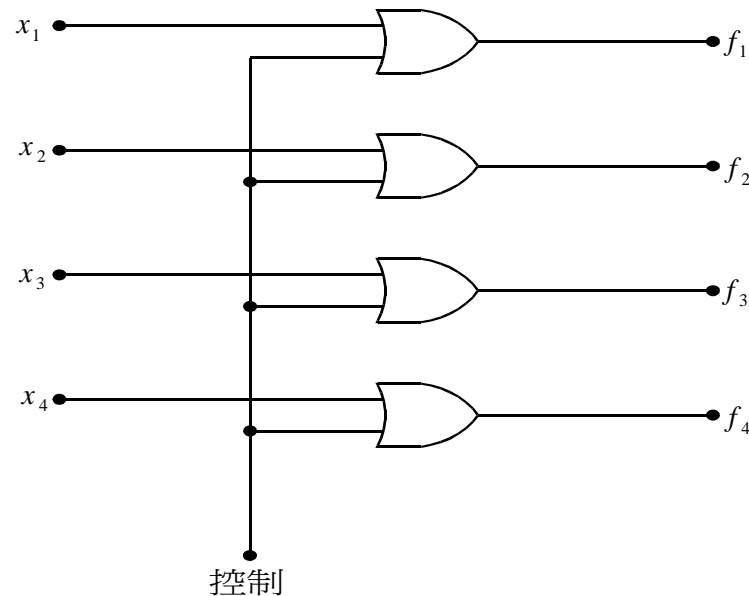


基本邏輯閘之應用(OR 閘)

- ◆ 使用 **OR 閘** 做**高態禁制** (Inhibit) 電路：當「控制=1」時， $f = 1$ (輸出端被**固定**為高態)；若「控制=0」， $f = x$ (輸入訊號 x **可通過** OR 閘至輸出端)。

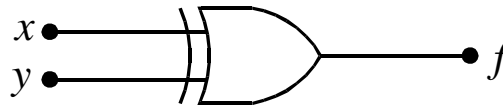


- ◆ 使用**一個 OR 閘** 僅能做**一個位元**之高態禁制電路，若欲對 n 個位元進行高態禁制，則只要使用 n 個 **OR 閘** 即可，而 **4 位元**之高態禁制電路，如下圖所示。

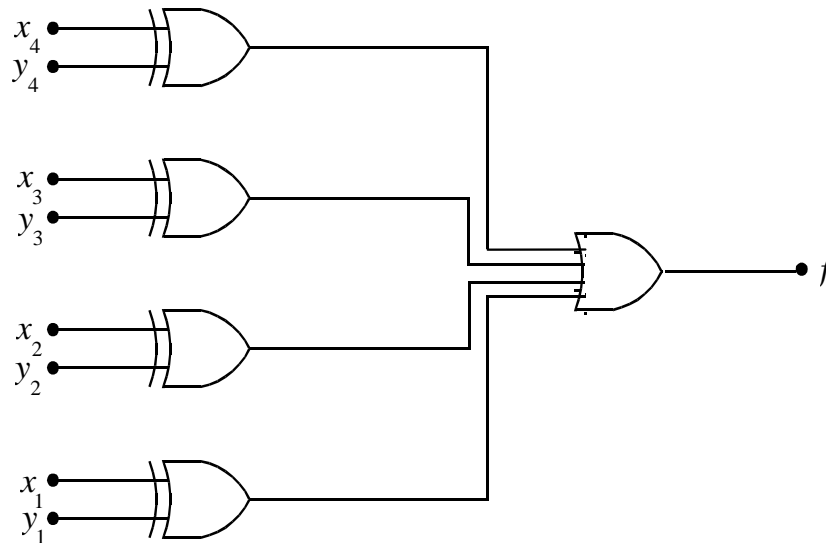


基本邏輯閘之應用(XOR 閘)

- ◆ **XOR(互斥或)閘**可用來決定兩個二進位數值**是否相等** (僅可比較兩個二進位數相等或不相等，而無法用來比較兩數之大小)，如下圖所示。當 $x \neq y$ 時， $f = 1$ ；若 $x = y$ 時， $f = 0$ 。



- ◆ 欲對 n 個位元進行比較運算時，可使用 n 個 XOR 閘，若兩個 4 位元輸入訊號分別為 $x = x_4x_3x_2x_1$ 與 $y = y_4y_3y_2y_1$ ，當 $x \neq y$ 時，輸出 $f = 1$ ；若 $x = y$ 時，輸出 $f = 0$ ，如下圖所示。



組合邏輯電路之設計步驟

◆ 組合邏輯電路之設計是由問題的描述開始，而以得到最簡之邏輯電路(最少的邏輯閘)為結果。

組合邏輯電路之輸出與輸入間具有直接的關係，故可遵循下列步驟來設計滿足實際需求之邏輯電路：

1. 分析實際之需求，以決定電路所需輸入與輸出變數的數目，並賦予每個變數不同之名稱(符號)。
2. 藉由問題所描述之邏輯功能，利用真值表定義輸出與輸入間之關係。
3. 利用函數之化簡法，求取最簡之布林函數表示式，以降低邏輯電路所需之硬體成本。
4. 將化簡所得之布林函數式，繪出實際所需之邏輯電路圖。
5. 驗證所得邏輯電路功能之正確性。

組合邏輯電路之設計原則

- ◆ 為了降低實現所設計的組合邏輯電路所需硬體成本與符合實際需求，除依照上述步驟來設計邏輯電路外，亦須考慮下列之條件：
 1. 使用數目最少之邏輯閘。
 2. 盡量減少每個邏輯閘之輸入接腳數。
 3. 盡量減少訊號經過邏輯電路（由輸入端到達輸出端）之階層數，以縮短訊號之傳輸延遲時間 (Propagation Delay Time)，提高電路之運算速度。
 4. 邏輯閘之位置應予適當的安排，使彼此間之連接線 (Interconnections) 最少為原則。
 5. 每個邏輯閘之扇入 (Fan-In) 與扇出 (Fan-Out) 能力之限制必須加以考慮。
- ◆ 當使用邏輯閘來實現數位電路時，雖然無法完全滿足上述條件，但設計者可視實際之需求，而做不同之選擇，但在進行布林函數化簡時，亦應參考上述之條件來進行化簡工作。

例題 4-1

試設計一個組合邏輯電路，其輸出是由 x 、 y 與 z 三個輸入所控制之投票器，當大部份輸入為邏輯 1（高態）時，則輸出為邏輯 1；否則輸出為邏輯 0，試設計此組合邏輯電路。

解

1. 此電路需 3 個輸入變數，分別標示為 x 、 y 與 z 等 3 個符號與 1 個輸出變數，標示為 $f(x, y, z)$ 之符號。
2. 因輸入僅有 3 個變數，當有 2 個或 2 個以上之輸入為邏輯 1 時，即可在輸出得到邏輯 1，否則輸出為邏輯 0，故可用下列真值表來定義輸出與輸入間的關係，如下表所示。

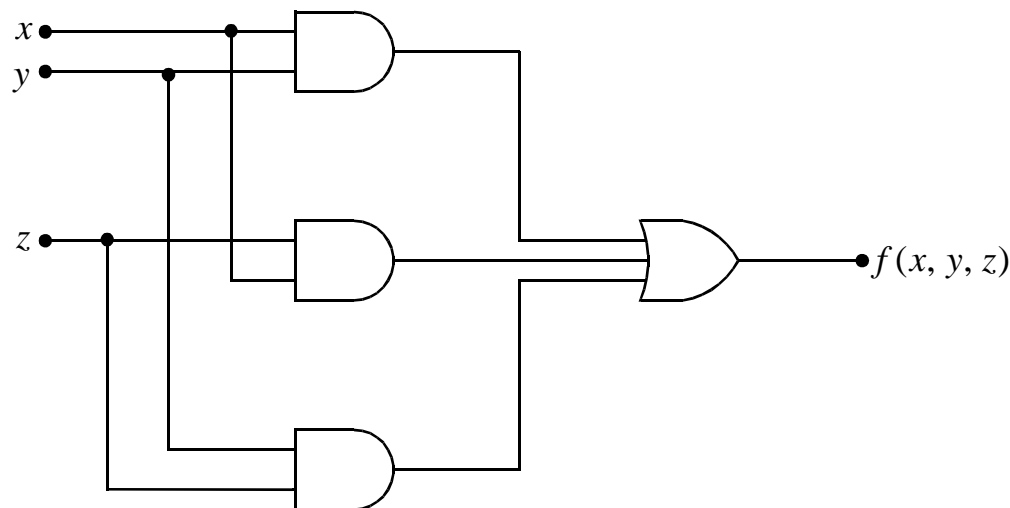
x	y	z	$f(x, y, z)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

3、 利用卡諾圖 (Karnaugh Map) 來化簡上面真值表，如下圖所示，以求得最簡之布林函數式

x \ yz	00	01	11	10
0	0	0	1	0
1	0	1	1	1

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

4. 最後使用邏輯閘來實現輸出 $f(x, y, z)$ 之布林函數式，即可繪出投票器之組合邏輯電路，如下圖所示。



例題 4-2

試設計一組合邏輯電路，它接受 3 位元之二進位數字，而產生一個等於輸入數平方的輸出之二進位數字。

解

1. 此電路需 3 個輸入變數，分別標示為 x 、 y 與 z 等 3 個符號，而輸出最大值為 $(7)_{10}$ 之平方值為 $(49)_{10}$ ，

因 $(49)_{10}$ 必需用 6 個位元之二進位數才足夠表示，故所設計之組合邏輯電路需有 6 個輸出，分別標示為 f_5 、 f_4 、 f_3 、 f_2 、 f_1 與 f_0 等 6 個符號。

2. 因所求之輸出為輸入之平方，故可以下列真值表來定義輸出與輸入間的關係。

x	y	z	f_5	f_4	f_3	f_2	f_1	f_0
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1
0	1	0	0	0	0	1	0	0
0	1	1	0	0	1	0	0	1
1	0	0	0	1	0	0	0	0
1	0	1	0	1	1	0	0	1
1	1	0	1	0	0	1	0	0
1	1	1	1	1	0	0	0	1

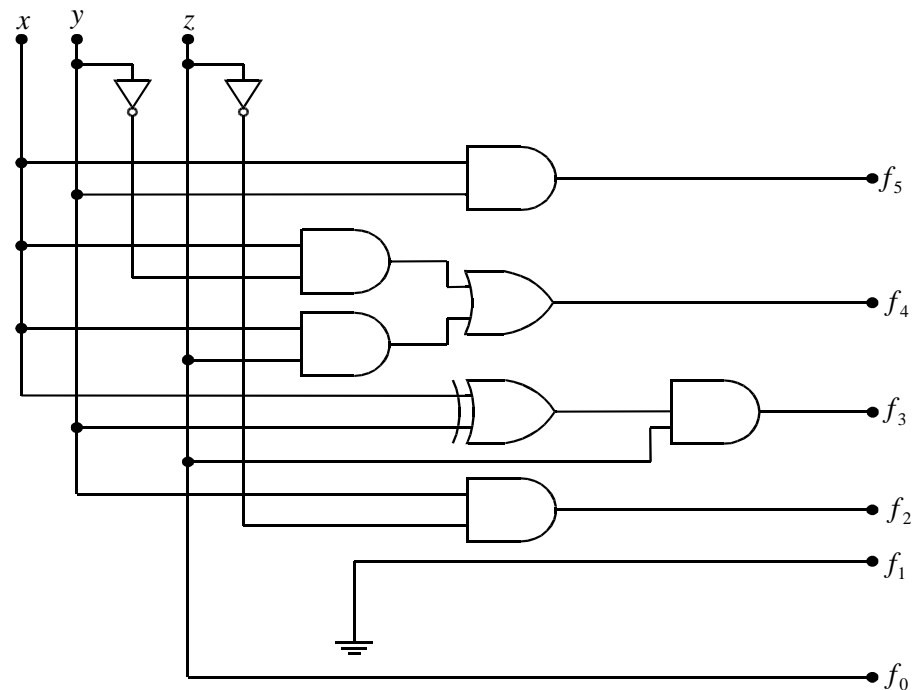
3. 利用卡諾圖化簡真值表，以求得 f_5 、 f_4 、 f_3 、 f_2 、 f_1 與 f_0 等 6 個輸出之最簡布林函數式如下：

$$f_5 = x \cdot y \quad , \quad f_4 = x \cdot \bar{y} + x \cdot z$$

$$f_3 = \bar{x} \cdot y \cdot z + x \cdot \bar{y} \cdot z = z \cdot (x \oplus y) \quad , \quad f_2 = y \cdot \bar{z}$$

$$f_1 = 0 \quad , \quad f_0 = z$$

4. 最後使用邏輯閘來實現 f_5 、 f_4 、 f_3 、 f_2 、 f_1 與 f_0 等 6 個輸出布林函數式，即可繪出平方產生器之組合邏輯電路，如下圖所示。



具有扇入限制之組合邏輯電路設計

- ◆ 每個邏輯閘之輸入端所能允許的最大數目(扇入)是有一定的限制，而使用扇入愈少之邏輯閘來設計電路，必須分解所得之布林函數式，以滿足實際之情況，因此可能會導致訊號由輸入至輸出需經過較多層之邏輯閘，即可得較多階層(Level)之邏輯電路，以導致傳遞延遲時間較長之問題發生。

例題 4-3

試使用最大扇入數為 2 之邏輯閘來設計一個輸入為 4 位元之二進位數，而輸出為輸入之 2 的補數的組合邏輯電路。

解

1. 此電路有 4 個輸入變數，分別標明為 w 、 x 、 y 與 z 等 4 個符號，因 4 位元二進位數之 2 的補數亦為 4 個位元，故所設計之組合邏輯電路，亦為 4 個輸出變數，分別標示為 f_4 、 f_3 、 f_2 與 f_1 等 4 個符號。
2. 因所求之輸出為輸入的 2 的補數，故可用下列真值表來定義輸出與輸入間的關係，如下表所示。
3. 利用卡諾圖來化簡上面真值表，可求得 f_4 、 f_3 、 f_2 與 f_1 等 4 個輸出之最簡布林函數式如下：

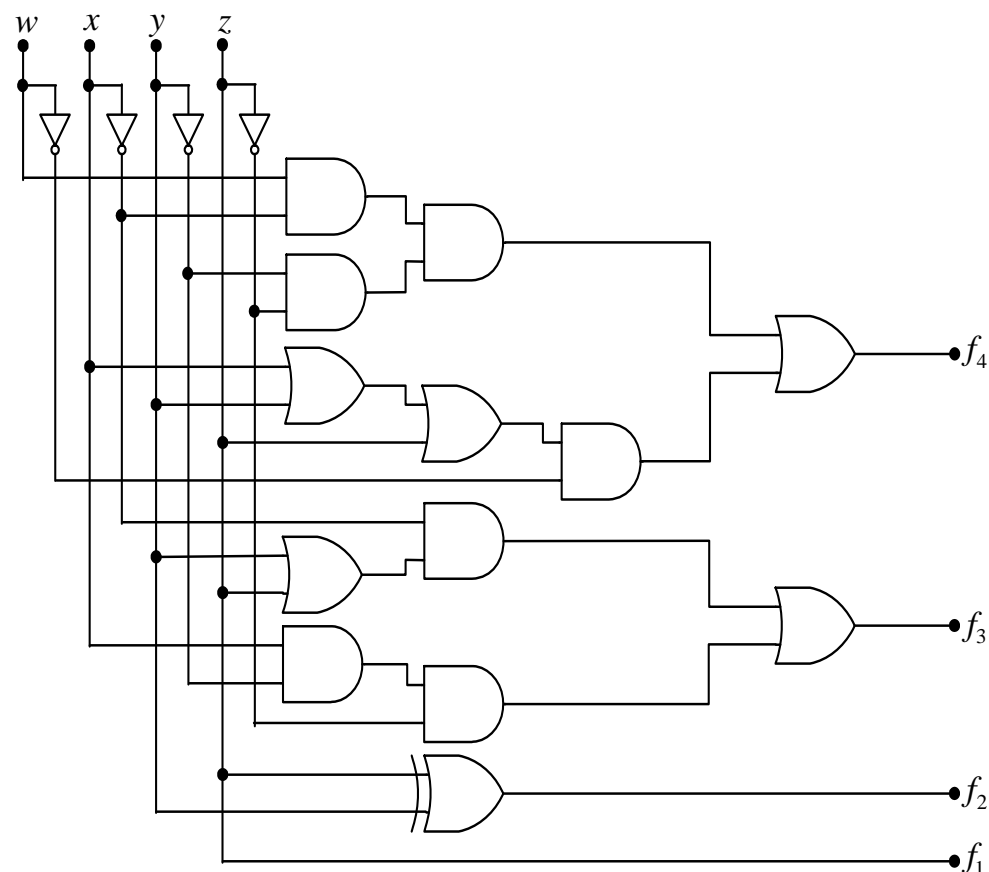
$$f_4 = w \cdot \bar{x} \cdot \bar{y} \cdot \bar{z} + \bar{w} \cdot z + \bar{w} \cdot y + \bar{w} \cdot x = w \cdot \bar{x} \cdot \bar{y} \cdot \bar{z} + \bar{w} \cdot (x + y + z)$$

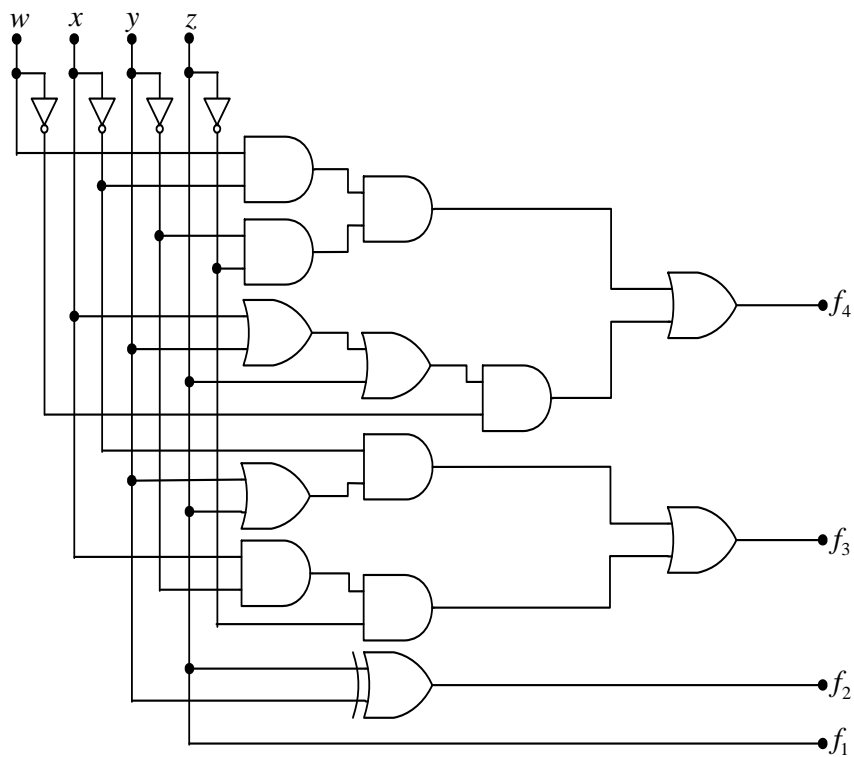
$$f_3 = \bar{x} \cdot z + x \cdot \bar{y} \cdot \bar{z} + \bar{x} \cdot y = \bar{x} \cdot (y + z) + x \cdot \bar{y} \cdot \bar{z}$$

$$f_2 = \bar{y} \cdot z + y \cdot \bar{z} = y \oplus z \quad , \quad f_1 = z$$

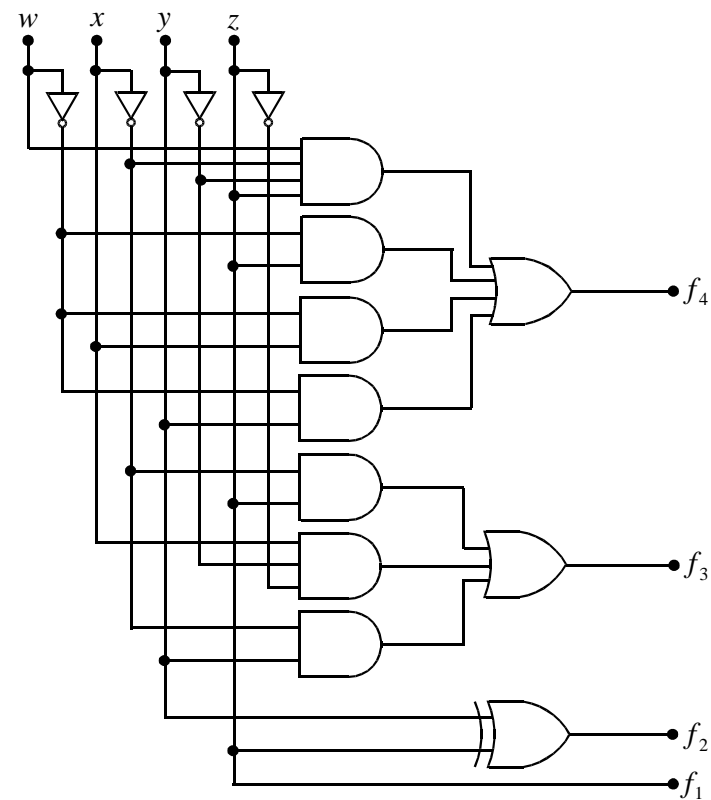
w	x	y	z	f_4	f_3	f_2	f_1
0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1
0	0	1	0	1	1	1	0
0	0	1	1	1	1	0	1
0	1	0	0	1	1	0	0
0	1	0	1	1	0	1	1
0	1	1	0	1	0	1	0
0	1	1	1	1	0	0	1
1	0	0	0	1	0	0	0
1	0	0	1	0	1	1	1
1	0	1	0	0	1	1	0
1	0	1	1	0	1	0	1
1	1	0	0	0	1	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	0	1	0
1	1	1	1	0	0	0	1

4. 最後使用扇入數為 2 之邏輯閘來實現 f_4 、 f_3 、 f_2 與 f_1 等 4 個輸出之布林函數式，即可得 2 的補數產生器之邏輯電路，如下圖所示。





使用扇入數為 2 之邏輯閘來實現 2 的補數產生器

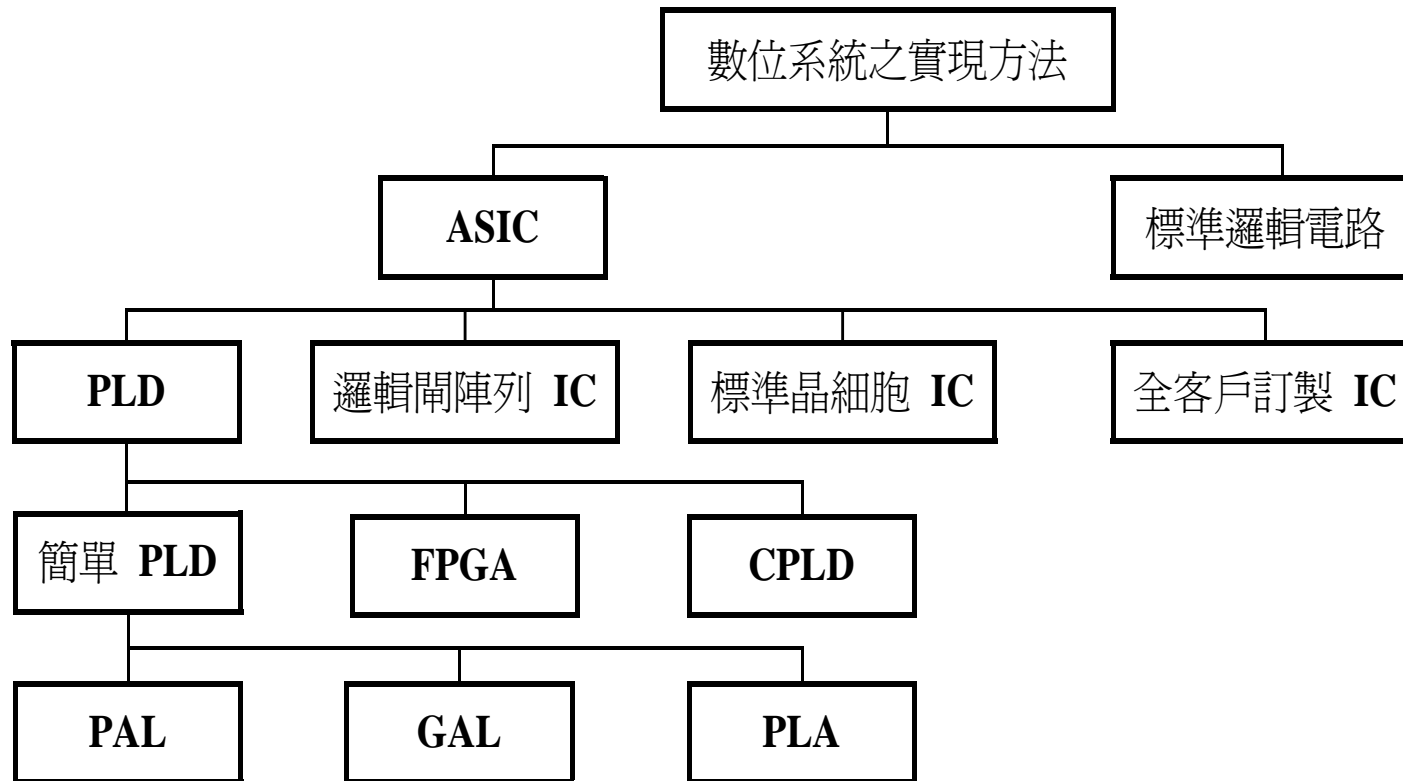


使用無扇入數限制之邏輯閘來實現 2 的補數產生器

邏輯電路之實現方法(概述)

- ◆ 當依組合邏輯電路設計步驟得到布林函數式後，接下來需考慮採用何種方法來實現邏輯電路，以製作出完整之數位系統，以符合實際之需求。由於目前積體電路製程技術之持續進步，因此 IC 已普遍使用於數位系統設計上，且面對如此快速變化之環境，電路設計者必須慎選研發工具與電子產品，以提高產品設計效率與產品上市時間。
- ◆ 實現邏輯電路之 IC 大致上可分為兩大類，而這兩大類之特性與優缺點歸納於下：
 1. 標準邏輯 IC (Standard Logic IC)：如 TTL/CMOS 等系列之 IC，使用這些元件來實現邏輯電路，具有價格便宜與元件取得容易之優點；但這些邏輯元件之功能固定，當需實現較複雜之邏輯電路時，會造成電路成本增加、接線繁雜、檢修困難、穩定度不佳與電路修改彈性不佳等缺點。
 2. 特殊應用規格晶片 ASIC(Application Specific Integrated Circuits)：此種技術來實現邏輯電路，可能會產生邏輯密度較低與操作速度較慢之缺點，但因此種技術是將所有邏輯閘集中在同一個晶片 (Chip) 上，故可得到穩定度高、保密性高與邏輯功能更改較具彈性等優點。

- ◆ 實現數位電路積體化，大致上可分為兩大類，如下圖所示。



邏輯電路之實現方法(ASIC)

- ◆ 因積體電路製程技術之進步，為儘可能縮短產品從設計至上市時間，目前 ASIC 之設計技術已廣泛使用於數位電路設計上。一般而言，目前 ASIC 之設計方法可分為下列 4 種技術：

1. 可程式邏輯元件 (Programmable Logic Device; PLD)：藉由將許多可規劃 AND-OR 功能之邏輯元件或可規劃之邏輯方塊 (Configuration Logic Block)，以陣列之方式排列在晶片上，且這些可規劃邏輯元件彼此間之連接關係亦是可程式的，故使用者可依電路之實際需求來定義邏輯功能，以構成一個完整之數位系統。
2. 閘陣列 (Gate-Array) IC：將單一功能之邏輯元件做成單一晶細胞 (Cell)，再將這些晶細胞以陣列之方式排列在晶片上，接著依實際之需要規劃各個晶細胞之邏輯功能與彼此間之連接關係，以構成一個完整之數位系統。
3. 標準晶細胞 (Cell-Based) IC：將各種具有不同邏輯功能之晶細胞組合在同一個晶片上，使用者可藉規劃這些晶細胞間之連接關係，即可用來定義晶片所實現之邏輯功能，以構成一個完整之數位系統。
4. 全客戶訂製 (Full-Custom) IC：電路所需之邏輯運算，皆委託設計廠商依照實際之需求，將所有之邏輯元件製作在同一個晶片上，再經過完整之光罩 (Mask) 流程製成 IC，以構成一個完整之數位系統。

組合邏輯電路之時序分析

- ◆ 當訊號進入邏輯閘後，欲在輸出得到邏輯運算後之結果，往往因物理的限制，而產生一段傳遞延遲時間 (Propagation Delay Time)，而傳遞延遲時間可由 IC 使用手冊查得。
- ◆ 當組合邏輯電路之輸入訊號的邏輯位準改變時，因輸入訊號往往會經由許多不同路徑到達輸出端，而每條路徑所經過之邏輯閘個數(階層數)可能不完全相同，故對每個路徑而言，皆可能會有不同之傳遞延遲時間發生。
- ◆ 當輸入訊號通過不同之路徑到達輸出端，因每條路經過不同邏輯閘層次，而造成傳遞延遲時間不同之問題，將會導致訊號到達輸出端之時間誤差，因此會有暫時性之錯誤情況發生，這種現象稱為突波 (Hazard)。
- ◆ 突波現象可能會導致在設計組合邏輯電路時，電路之輸出不完全是輸入之函數，故必須考慮邏輯閘之傳遞時延遲間對輸出之影響，以設計出能符合實際需求之數位電路。
- ◆ 組合邏輯電路之突波可分為靜態突波 (Static Hazard) 與動態突波 (Dynamic Hazard) 等兩種，而這兩種突波皆為暫時性的，亦可視為邏輯電路之暫態響應 (Transient Response)。

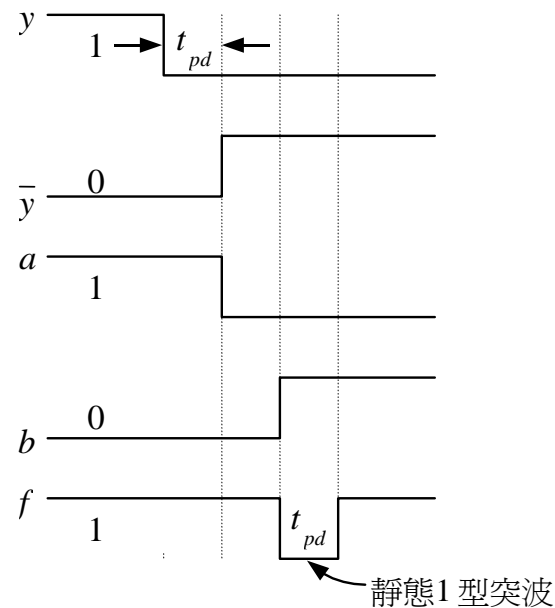
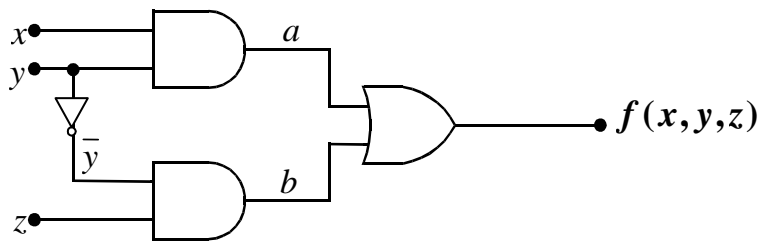
靜態突波(概述)

- ◆ **靜態突波** (Static Hazard) 是指輸出邏輯位準不應有變動，因受邏輯閘**傳遞延遲時間**的影響，而造成**輸出邏輯位準短暫之跳動**。而靜態突波可分為**靜態 1 型突波** (Static-1 Hazard) 與**靜態 0 型突波** (Static-0 Hazard) 等兩種。
- ◆ 某些輸入訊號之邏輯位準改變時，理論上，輸出端之**邏輯 1** 應保持不變，因邏輯閘傳遞延遲之影響，使輸出訊號離開穩態的邏輯 1，而**暫時轉變為邏輯 0**，經過一段時間後，再回到穩態之邏輯 1，則表示此組合邏輯電路含有**靜態 1 型突波** (Static-1 Hazard)。
- ◆ 當某些輸入訊號之邏輯位準改變時，理論上，輸出端之**邏輯 0** 應保持不變，因邏輯閘傳遞延遲之影響，使輸出訊號會離開穩態的邏輯 0，而**暫時轉變為邏輯 1**，經過一段時間後，再回到穩態之邏輯 0，則表示此組合邏輯電路含有**靜態 0 型突波** (Static-0 Hazard)。

靜態 1 型突波之產生原因

- ◆ 以下圖之組合邏輯電路為例，以說明靜態 1 型突波之產生原因如下：

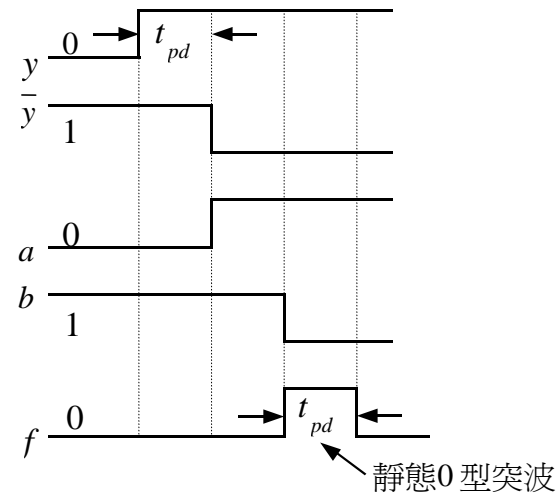
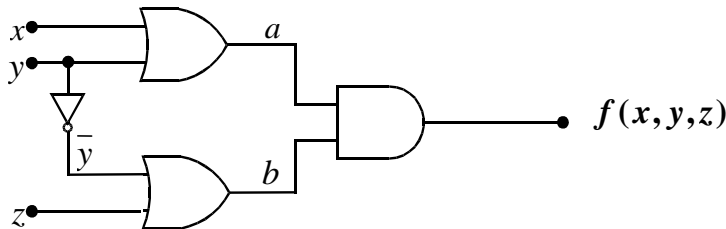
在穩定狀態下，若輸入 $x = y = z = 1$ 時，依邏輯運算可得輸出 $f = 1$ 。當輸入 y 之邏輯值由 1 變成 0 時，理論上，輸出 f 應依然保持邏輯 1，但因邏輯閘傳遞延遲時間的影響，輸出 f 會有一短暫時間 (t_{pd}) 轉變為邏輯 0，經過一段時間後，才會再回到穩定的邏輯 1，如左圖所示，而此種現象表示含有靜態 1 型突波。



靜態 0 型突波之產生原因

- ◆ 以下圖之組合邏輯電路為例，以說明靜態 0 型突波之產生原因如下：

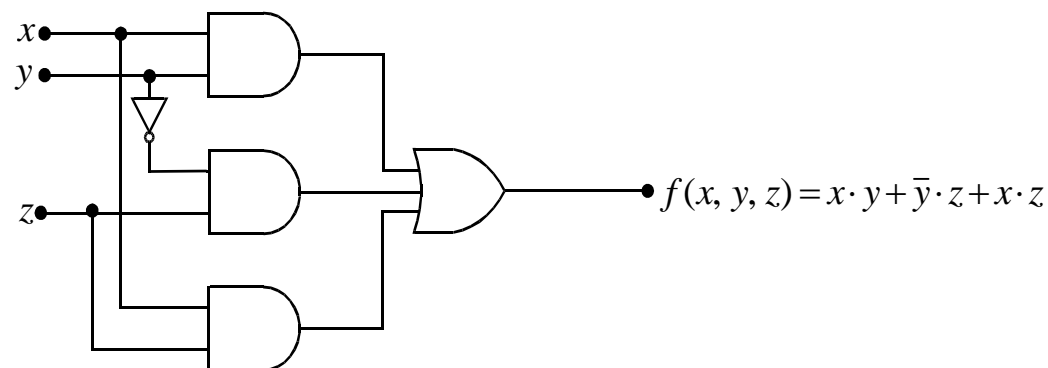
在穩定狀態下，若輸入 $x = y = z = 0$ 時，依邏輯運算可得輸出 $f = 0$ 。當輸入 y 之邏輯位準由 0 變成 1 時，理論上，輸出 f 應依然保持邏輯 0，但因邏輯閘傳遞延遲時間的影響，輸出 f 會有一短暫時間 (t_{pd}) 轉變為邏輯 1，經過一段時間會，再回到穩定的邏輯 0，如左圖所示，而此種現象表示含有靜態 0 型突波。



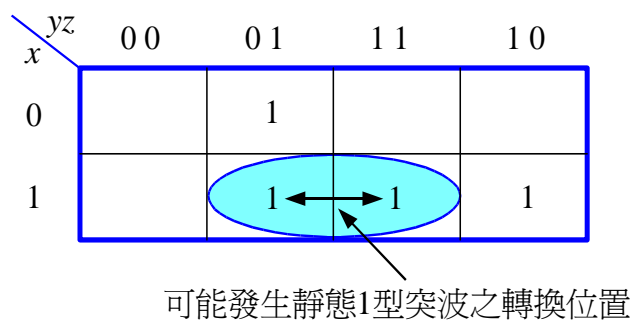
靜態突波之消除

- ◆ 靜態 1 型突波在 $y + \bar{y} = 1$ 之邏輯運算出問題，即在突波發生瞬間會有 $y + \bar{y} = 0$ 違反邏輯運算出現，即 $b = z \cdot \bar{y}$ 與 $a = x \cdot y$ 之運算，應該是同時發生的，但對 y 進行反運算時，需一段傳遞延遲時間，致使 OR 閘之輸入端 a 與 b 有一段時間會同時為 0 之錯誤發生，而造成一個短暫之靜態 1 型突波。
- ◆ 只要可避免 $y + \bar{y} = 0$ 之錯誤情況，即可消除靜態 1 型突波，而欲避免因傳遞延遲所造成之 $y + \bar{y} = 0$ 的錯誤，可考慮多加入一個積項(且加入此積項後之布林函數之輸出必須與原函數相同)，即 $f = x \cdot y + \bar{y} \cdot z + C$ ，其中 C 為需加入之積項。而造成靜態 1 型突波原因與消除方法如下：
 - 1、若 $x \cdot y = \bar{y} \cdot z = 1$ 、而 $C = 0$ 時，當 $x = z = 1$ ，而 y 之邏輯值由 1 跳動至 0 時，便會出現靜態 1 型突波。
 - 2、若加入一積項 C ，且 $C \supseteq xz$ ，則 x 與 z 同時為邏輯 1 時， C 之邏輯值亦為 1，如此便可消除靜態 1 型突波。

- ◆ 由以上之討論可知，將原布林函數更改為 $f = x \cdot y + \bar{y} \cdot z + x \cdot z$ ，即可消除邏輯電路之靜態 1 型突波，而更改後之邏輯電路圖，如下圖所示。

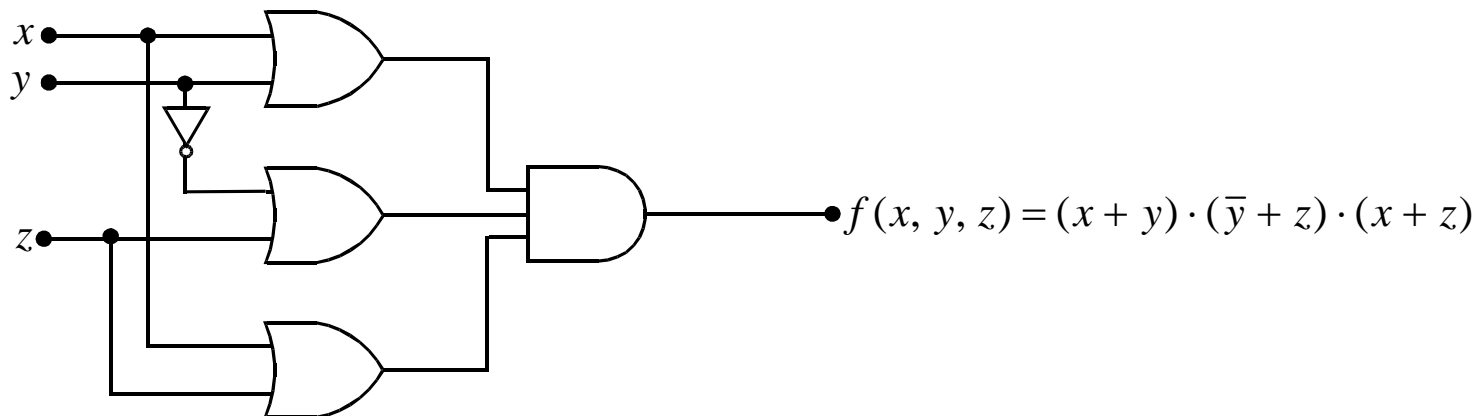


- ◆ 若從卡諾圖之觀點來討論造成靜態 1 型突波原因與消除方法，首先繪出輸出函數 $f = x \cdot y + \bar{y} \cdot z$ 之卡諾圖，如下圖所示。



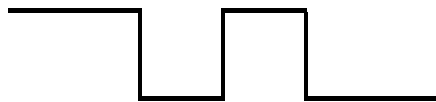
◆ 觀察上圖可知，僅在輸入訊號 x 、 y 、 z 之邏輯值，由 **111 轉態為 101** 時（所圈選之相鄰方格），即積項 $\bar{y} \cdot z$ 轉移至 $x \cdot y$ 時，會發生**靜態 1 型突波**。因此只要將**加入一個積項 $x \cdot z$** ，使當輸入訊號 x 、 y 、 z 之邏輯值由 **111 轉態為 101** 時（反之亦然），即**積項 $x \cdot z$ 可包含這兩個輸入狀態**，如此便可**消除靜態 1 型突波**。

◆ 同理，只要可避免 $y \cdot \bar{y} = 1$ 之錯誤情況發生，即**加入一個可包含 x 與 z 之和項**，且此外加之和項需符合符合 $C \supseteq xz$ 之條件，故可將原布林函數更改為 $f = (x + y) \cdot (\bar{y} + z) \cdot (x + z)$ ，便可**消除靜態 0 型突波**，如右圖所示。

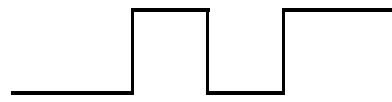


動態突波

- ◆ 若輸入變數之邏輯值轉態期間，輸入訊號可能經過許多不同邏輯閘層次之路徑到達輸出端，導致不同路徑有傳遞延遲時間不同的影響，將會使輸出邏輯值發生多次暫時性變化，此現象稱為組合邏輯之動態突波 (Dynamic Hazard)。
- ◆ 當發生在輸出邏輯值應由 1 轉變為 0 時，輸出邏輯值改變超過 3 次以上，才會回到穩定的邏輯 0，稱為 1 到 0 之動態突波，如下圖 (a) 所示；反之，當發生在輸出邏輯值應由 0 轉變為 1 時，輸出邏輯值改變超過 3 次以上，才會回到穩定的邏輯 1，稱為 0 到 1 之動態突波，如下圖(b)所示。

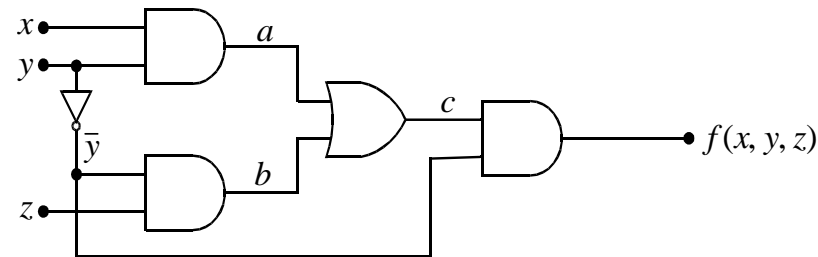


(a) 1 到 0 之動態突波

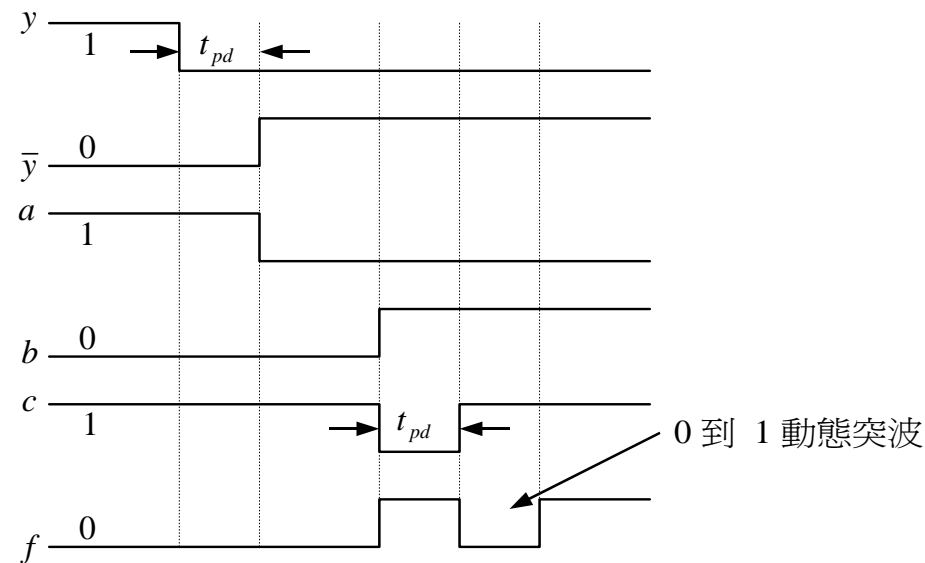


(b) 0 到 1 之動態突波

- ◆ 如下圖所示，在穩定之狀態下，若輸入之邏輯值 $x = y = z = 1$ 時，依邏輯運算可得輸出 $f = 0$ 。

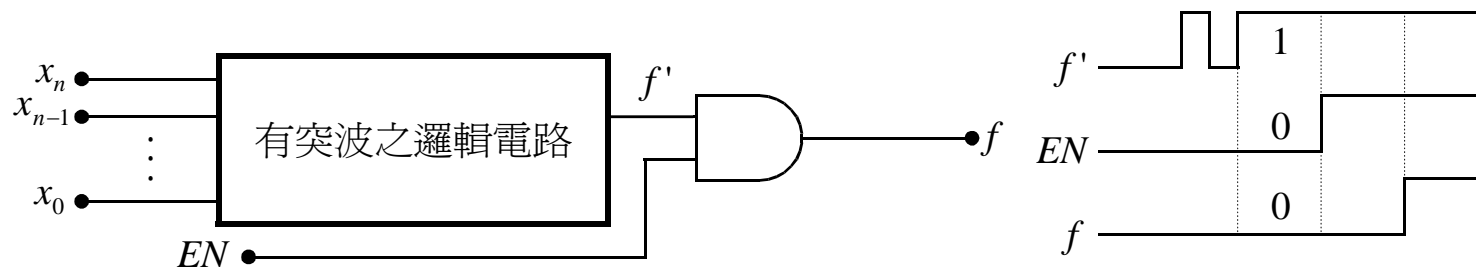


- ◆ 當輸入 y 之邏輯值由 1 轉變為 0 時，理論上，輸出 f 之邏輯值應立即由 0 穩定的轉變為 1，但因輸入 y 有 3 條路徑到達輸出端，且每條路徑所經過之邏輯閘層次皆不相同，由於邏輯閘傳遞延遲時間的影響，輸出 f 之邏輯值會經過 3 次變動後，才會回到穩定之邏輯 1。

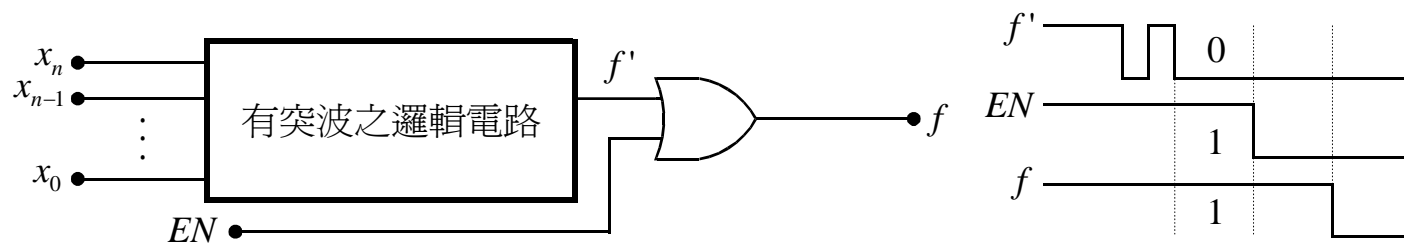


動態突波之消除

- ◆ 多層之組合邏輯電路中，當某些輸入訊號會經過 3 條或 3 條以上之路徑到達輸出端時，便有可能產生動態突波。將多層之邏輯電路，化簡為 SOP (或 POS) 之布林函數式，即可用兩層之邏輯電路來實現布林函數，此為消除組合邏輯電路動態突波最簡單之方法。
- ◆ 若有 multiple 輸入變數一起轉態時，亦可能產生多個靜態與動態突波。欲消除這些突波，必須增加一些額外之積項 (或和項)，如此電路之硬體成本亦會相對增加。本書提供一種不會增加太多硬體成本之條件，即可消除較多突波之組合邏輯電路。若使用外加電路將突波之問題忽略掉，故可同時消除組合邏輯電路之靜態與動態突波的一種較簡單方法，如下圖所示。



(a) 可消除靜態 1 型突波與 0 至 1 動態突波



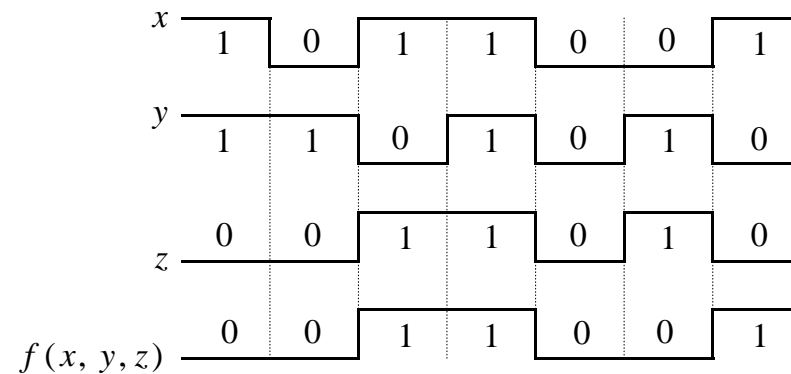
(b)可消除靜態 0 型突波與 1 至 0 動態突波

利用時序波形來設計組合邏輯電路

- ◆ 使用**時序脈波**求解組合邏輯電路之步驟與前面所討論之方法類似，接著將設計步驟列述於下：
 1. 將已知輸入與輸出之時序脈波轉換成與輸出與輸入關係的**真值表**。
 2. 利用函數之化簡法，求取**最簡之布林函數**表示式，以降低邏輯電路所需之硬體成本。
 3. 利用化簡所得之布林函數式，繪出實際所需之**邏輯電路圖**。
 4. 驗證邏輯電路功能之正確性。

例題 4-4

試求可實現下列 3 個輸入變數 (分別為 x 、 y 、 z) 與 1 個輸出變數 $f(x, y, z)$ 時序脈波之組合邏輯電路。



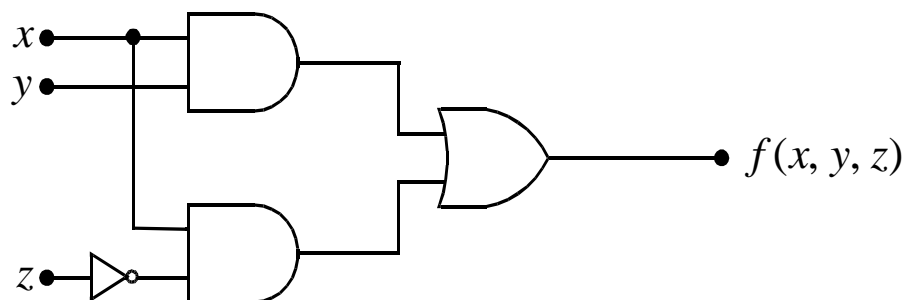
解

1. 觀察已知之輸入與輸出時序脈波，可得所求電路之輸出與輸入關係的真值表如下

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

2. 利用卡諾圖來化簡上面之真值表，可得最簡輸出布林函數式為 $f(x, y, z) = x \cdot y + x \cdot \bar{z}$ 。

3. 使用邏輯閘實現 $f(x, y, z) = x \cdot y + x \cdot \bar{z}$ 之布林函數式，即可繪出所求之組合邏輯電路，如下圖所示。

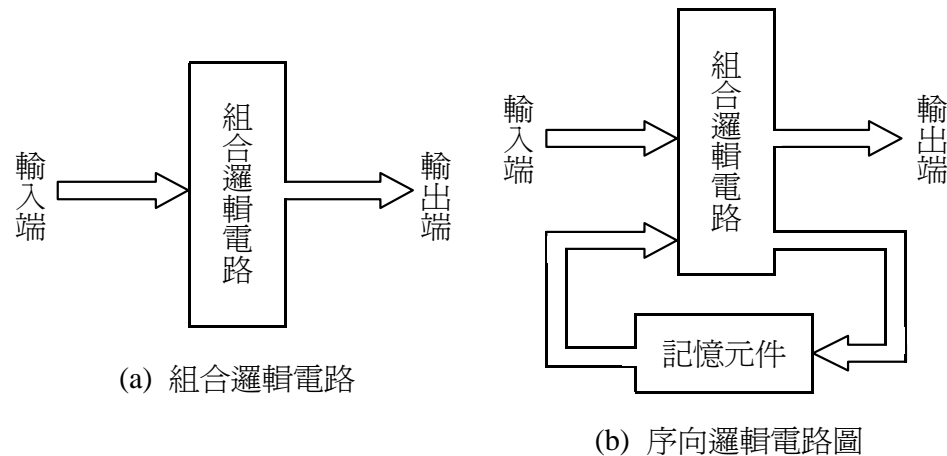


問題討論

1、試分別說明**組合邏輯電路**與**序向邏輯電路**之基本定義。

解：

組合邏輯電路是由基本邏輯閘所組成的，此種電路之輸出是所有輸入的組合，故輸出狀態可完全由**目前之輸入**來決定，組合邏輯電路之示意圖，如下圖(a)所示。序向邏輯電路是由基本邏輯閘與記憶元件所組成的，此種電路有**記憶元件**存在，故輸出不僅需考慮目前之輸入狀態，且亦與上一個輸出狀態有關，因此序向電路之輸出與時間有關，而序向邏輯電路之示意圖，如下圖(b)所示。



2、試簡略說明目前使用 **ASIC 之設計技術**，以實現數位系統之設計方法有那幾種。

解：

因積體電路製程技術之進步，且為儘可能縮短產品從設計至上市時間，目前ASIC之設計技術已廣泛使用於數位電路設計上。一般而言，目前ASIC之設計方法可分為下列4種技術：

- a、**可程式邏輯元件** (Programmable Logic Device; PLD)：藉由將許多可規劃 AND-OR 功能之邏輯元件或可規劃之邏輯方塊 (Configuration Logic Block)，以陣列之方式排列在晶片上，且這些可規劃邏輯元件彼此間之連接關係亦是可程式的，故使用者可依電路之實際需求來定義邏輯功能，以構成一個完整之數位系統。
- b、**閘陣列** (Gate-Array) IC：將單一功能之邏輯元件做成單一晶細胞 (Cell)，再將這些晶細胞以陣列之方式排列在晶片上，接著依實際之需要規劃各個晶細胞之邏輯功能與彼此間之連接關係，以構成一個完整之數位系統。
- c、**標準晶細胞** (Cell-Based) IC：將各種具有不同邏輯功能之晶細胞組合在同一個晶片上，使用者可藉規劃這些晶細胞間之連接關係，即可用來定義晶片所實現之邏輯功能，以構成一個完整之數位系統。
- d、**全客戶訂製** (Full-Custom) IC：電路所需之邏輯運算皆委託設計廠商依照實際之需求，將所有之邏輯元件製作在同一個晶片上，再經過完整之光罩 (Mask) 流程製成IC，以構成一個完整之數位系統。

3、試討論奇同位 (Odd Parity)與偶同位 (Even Parity) 之定義。

解：

為保證所傳送的每筆二進位資料之邏輯 1 個數為奇數或偶數，可在所傳輸之二進位資料加上一個額外位元，此外加位元之邏輯值，可由系統與所傳送之二進位資料來決定，而此外加之位元稱為同位元 P (Parity Bit)。若可使二進位資料有奇數個邏輯 1 稱為奇同位 (Odd Parity)，而有偶數個邏輯 1 稱為偶同位 (Even Parity)。

4、試討論產生靜態 1 型突波 (Static 1 Hazard) 與靜態 0 型突波 (Static 0 Hazard) 之原因。

解：

- (1) **靜態 1 型突波**：當某些輸入訊號之邏輯位準改變時，理論上，輸出端之邏輯 1 應保持不變，因邏輯閘傳遞延遲之影響，使輸出訊號離開穩態的邏輯 1，而暫時轉變為邏輯 0，經過一段時間後，再回到穩態之邏輯 1，則表示此組合邏輯電路含有靜態 1 型突波 (Static-1 Hazard)。
- (2) **靜態 0 型突波**：當某些輸入訊號之邏輯位準改變時，理論上，輸出端之邏輯 0 應保持不變，因邏輯閘傳遞延遲之影響，使輸出訊號會離開穩態的邏輯 0，而暫時轉變為邏輯 1，經過一段時間後，再回到穩態之邏輯 0，則表示此組合邏輯電路含有靜態 0 型突波 (Static-0 Hazard)。

5、試討論產生動態突波 (Dynamic Hazard) 之原因。

解：

當輸入變數之邏輯值轉態期間，輸入訊號可能經過許多不同邏輯閘層次之路徑到達輸出端，理論上，僅可能造成的輸出邏輯值一次之變動，但實際上，在輸入變數之邏輯值轉態期間，輸入訊號可能經過許多不同邏輯閘層次之路徑到達輸出端，因邏輯閘傳遞延遲時間之影響，可能會導致輸出邏輯值發生多次暫時性變化，此現象稱為組合邏輯之動態突波 (Dynamic Hazard)。