



組合邏輯電路設計

SSI執行

3-1 簡介

數位系統大致上可分成：

1.組合邏輯(Combinational logic):

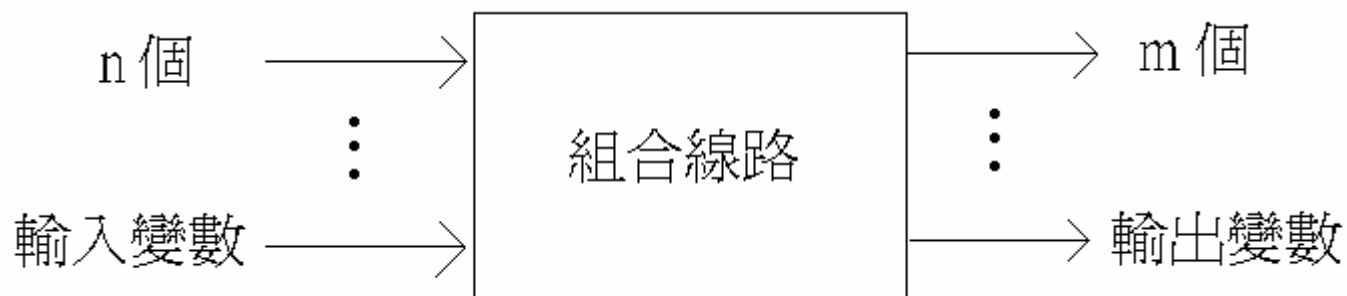
輸出可以直接由輸入組合的形式表示，不涉及過去以輸入情況，亦即只與電路輸入有關而與時間無關。

2.序向邏輯(Sequential logic):

輸出除了和目前的輸入狀態有關外，也與先前的輸出狀態有關。亦即除了具有邏輯閘之外，尚有記憶裝置(記憶體)。

3-2 組合邏輯電路設計

1. 組合線路方塊圖



2. 設計步驟：規格描述→邏輯圖完成

- 1.問題要明確

- 2.決定輸入、輸出數

- 3.每一個輸入、輸出給一標識字母

- 4.導出真值表(或交換表示)以定義輸入、輸出關係

- 5.布氏函數化簡

- 6.畫出邏輯圖(即執行交換函數)

3.設計例子：加法器、減法器

例1.半加器(HA, Half-Adder) — 能處理兩個bit的相加
解：能處理 2 bit, 故有 2^2 種輸入變化

真值表：

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$C = x \cdot y$$

$$S = x \oplus y$$

(exclusive OR)

例2.全加器(FA, Full Adder) —能處理 3 個 bit 相加
解:3 bit 有 $2^3 = 8$ 種變化

x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$\begin{aligned}C &= x'y z + x y' z + x y z' + x y z \\ &= x y + z(x \oplus y)\end{aligned}$$

$$\begin{aligned}S &= x'y'z + x'y z' + x y' z' + x y z \\ &= (x'y' + x y)z + z'(x'y + x y') \\ &= (x \odot y)z + z'(x \oplus y) \\ &= x \oplus y \oplus z\end{aligned}$$

例3. 半減器—能處理兩位元相減

解：

真值表：

x	y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

$$B = x'y$$


$$D = x \oplus y$$

例4.全減器 — 能處理有前級借位時,兩位元相減
解:

真值表:

x	y	z	B	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

輸出函數值 = $x - y - z$
若有借位, 即 $z=1$
則 $B=1$, x 加2
若無借位即 $z=0$
相當於半減器情形


$$\begin{aligned} B &= x' y' z + x' y z' + x' y z + x y z \\ &= x' (y \oplus z) + y z \end{aligned}$$

$$\begin{aligned} D &= x' y' z + x' y z' + x y' z' + x y z \\ &= (x' y' + x y) z + (x' y + x y') z' \\ &= (x \odot y) z + (x \oplus y) z' \\ &= x \oplus y \oplus z \end{aligned}$$

比較： $D = S$, B 相似 C , 只 x 改 x' , 則全加器變成全減器。

3-3 組合邏輯電路的執行

目前常用以(組合或序向)邏輯電路執行方式，若依數位IC的包裝密度可以區分為：

1. 小型積體電路(Small-scale integration, SSI):

每一個晶片或包裝中含有邏輯閘數且少於10個的IC。典型IC為AND、OR、NOT、NAND、NOR與XOR等基本邏輯電路。

2. 中型積體電路(Medium-scale integration, MSI):

邏輯閘數目介於10與100之間的IC，典型的IC晶片為加法器(adder)、多工器(multiplexer)與計數器(counter)等電路。

3. 大型積體電路(large-scale integration, LSI):

邏輯閘數目介於100與1000之間的IC。典型的IC晶片為記憶體器、微處理器與週邊裝置(peripheral)。

4. 超大型積體電路(very large-scale integration, VLSI):

每一個晶片中含有邏輯閘數目在1000個以上的IC。典型以IC晶片為微算機, 大的計算機組件等。

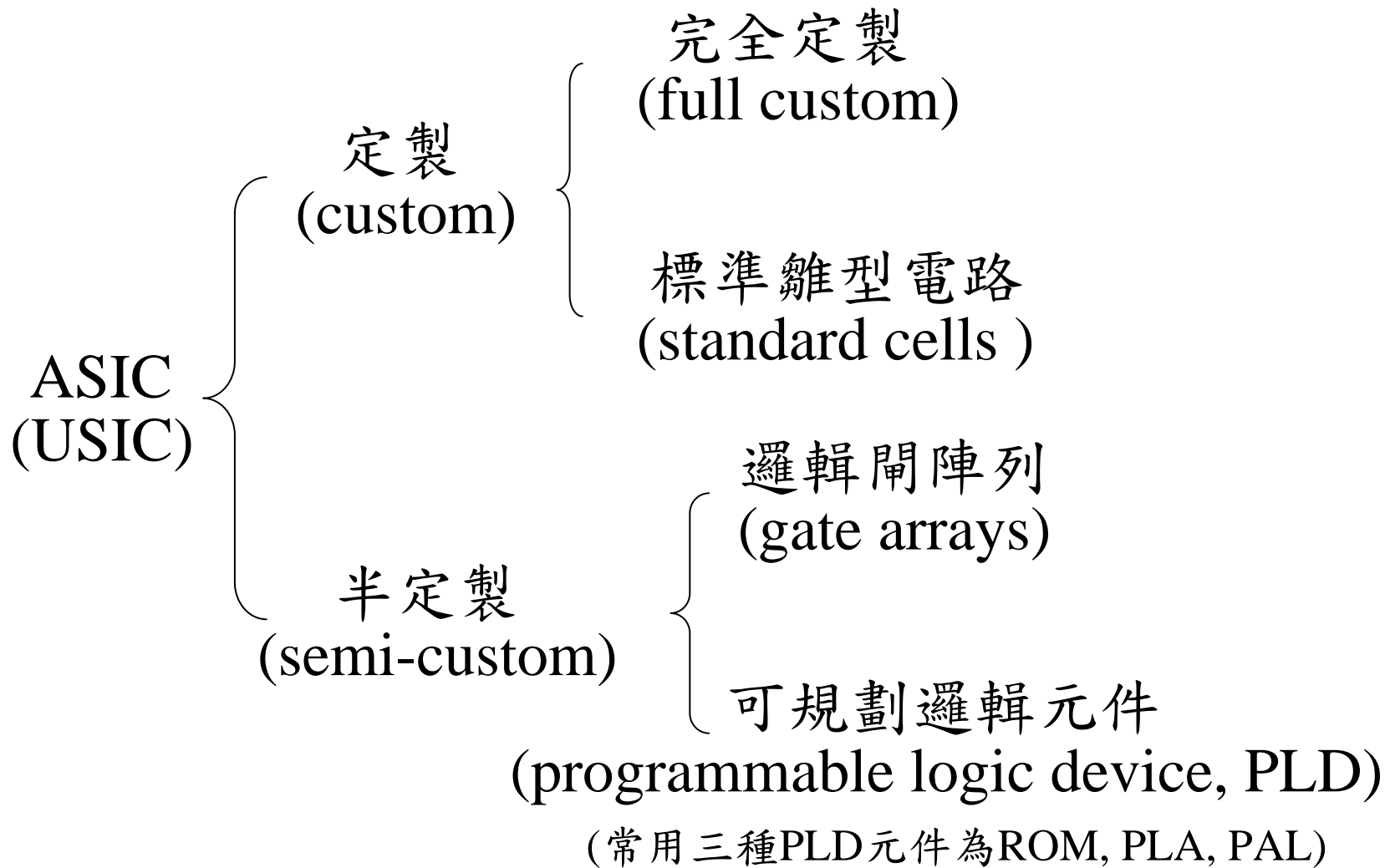


若以IC晶片的規格定義方式, 則分為下列兩類:

1. 標準規格IC (standard IC 或稱 catalog IC)

例如74系列的SSI與MSI等層

2. 應用規格IC (application specific IC, 簡稱ASIC, 或稱 User specific IC, USIC):



除PLD外，其他元件電路均相當複雜，在實際應用設計數位系統時，須藉CAD軟體資助。

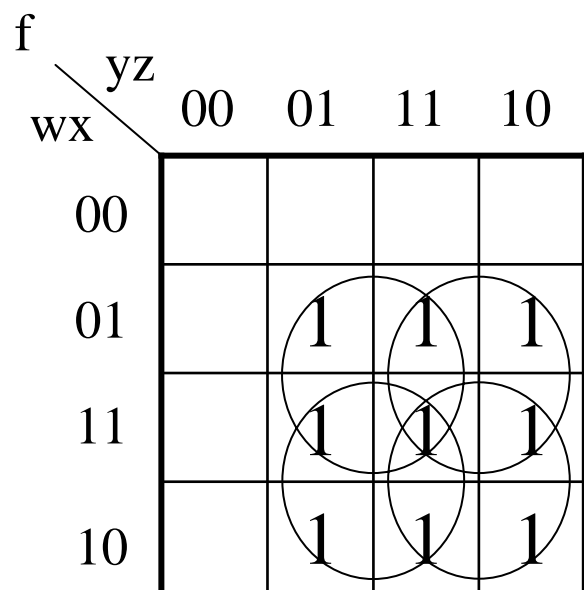
3-4 以SSI執行組合電路

1. 閘的最佳選擇：

聰明的設計師總是在尚未開始接線前，先列出欲執行交換函數最簡的 SOP 及 POS 兩種形式的式子，再以基本邏輯閘(AND、OR、NOR、NAND)執行。

例題： $f(w, x, y, z) = \sum (5, 6, 7, 9, 10, 11, 13, 14, 15)$
(a) 求最簡 SOP 及 POS 形式 (b) 比較兩種形式，
以做閘的最佳選擇

解：



$$f = xz + xy + wz + wy$$

sop 形式須 5 個 Gate
且含有 4 個和項

$$f(w, x, y, z) = \Pi(0, 1, 2, 3, 4, 8, 12)$$

Truth Table for $f(w, x, y, z)$:

$wx \backslash yz$	00	01	11	10
00	0	0	0	0
01	0			
11	0			
10	0			

The table shows that the function f is 0 for all combinations of w, x, y, z where w and x are both 0, or w and x are both 1. This is represented by the ellipses in the original image.

$$f = (w + x)(y + z)$$

pos形式(較佳)

只需 3 個 Gate 且只含有兩個和項與積項故符合交換函數化簡的基本觀念, 獲得了最少的邏輯閘數目與邏輯閘的扇入數目。

2. 多重輸出的化簡

多輸出函數的化簡程序相當複雜, 尤其是輸出函數很多時更是困難, 甚至使用數位計算機時亦然。因此, 實際上在設計多輸出交換電路時必須在電路成本與設計時所花費的時間(即代價)上做取捨, 以下試舉兩個例子以說明其基本觀念。

例題1: 試執行下列多輸出交換函數:

$$f_1(x, y, z) = \Sigma(2, 4, 6, 7)$$

$$f_2(x, y, z) = \Sigma(4, 7)$$

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

		yz			
		00	01	11	10
x	0				
	1	1		1	

$$f_1 = x z' + x y + y z'$$

須5個AND, 2個OR,

$$f_1 = x y' z' + x y z + y z'$$

只需3個AND 與 2個OR

$$f_2 = x y' z' + x y z$$
$$f_2 = x y' z' + x y z$$

例題2: 試執行下列多重輸出交換函數:

$$f_1 = \Sigma (4, 5, 12, 13, 15)$$

$$f_2 = \Sigma (6, 14, 15)$$

f_1		yz			
		00	01	11	10
wx	00				
	01	1	1		
	11	1	1	1	
	10				

f_2		yz			
		00	01	11	10
wx	00				
	01				1
	11			1	1
	10				

$$f_1 = x y' + w x z \quad f_1 = x y' + w x y z$$

需4個AND, 2個 OR 只需3個AND, 2個 OR

$$f_2 = x y z' + w x y$$

$$f_2 = x y z' + w x y z$$

尋找共用項之基本觀念：

1. 先找出函數間非共有半主體
2. 小心檢視相適用方程式或利用函數乘積觀念尋找可能共用項

如例1: $f_1 - f_2 = \Sigma(4, 7)$

例2: $f_1 - f_2 = \Sigma(15)$

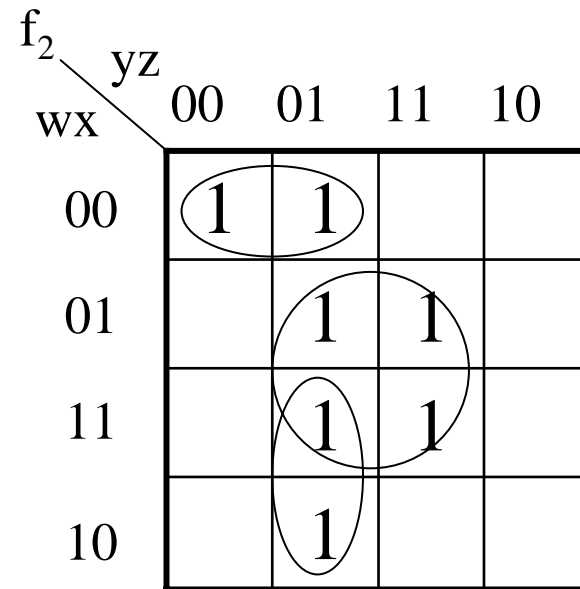
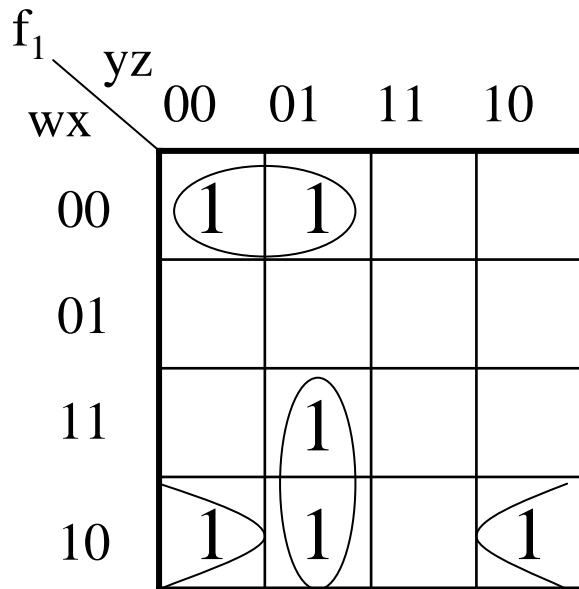
3-5 結論

在設計一個數位電路(或系統)時,應盡量使用具有較大功能的IC,以減少IC的數目,因而減少外加接線數目,與降低成本及增加可靠度。亦即在設計一個數位系統時,應儘可能使用MSI或LSI電路,而SSI電路,則只用來當作電路間的介面。

pb. 試化簡下列多輸出交換函數

$$f_1 = \Sigma (0, 1, 8, 9, 10, 13)$$

$$f_2 = \Sigma (0, 1, 5, 7, 9, 13, 15)$$



$$f_1 \cdot f_2 = \Sigma (0, 1, 9, 13)$$

pb. 利用多重輸出觀念，考慮全加器之S與C、全減器之B與D。