

(1)

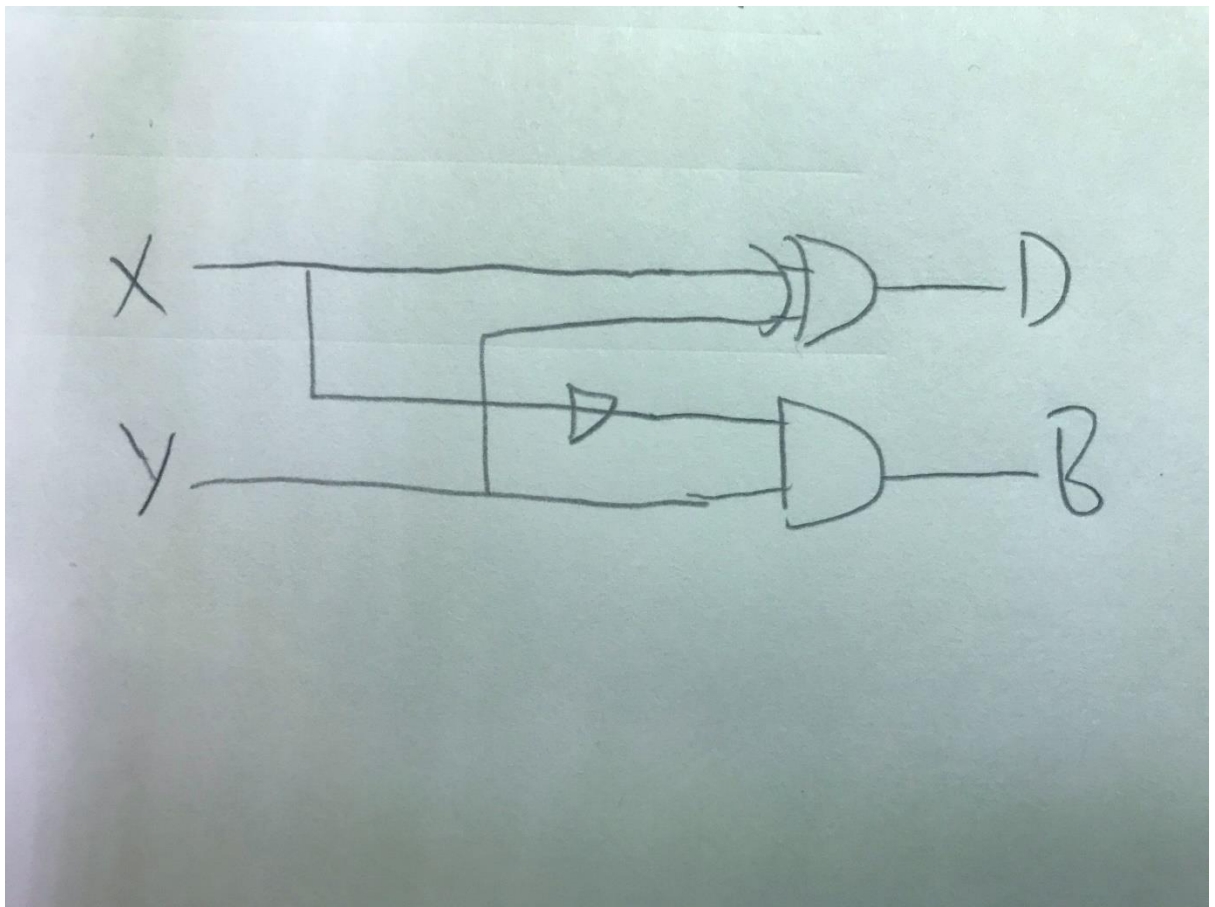
半減器之真值表

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

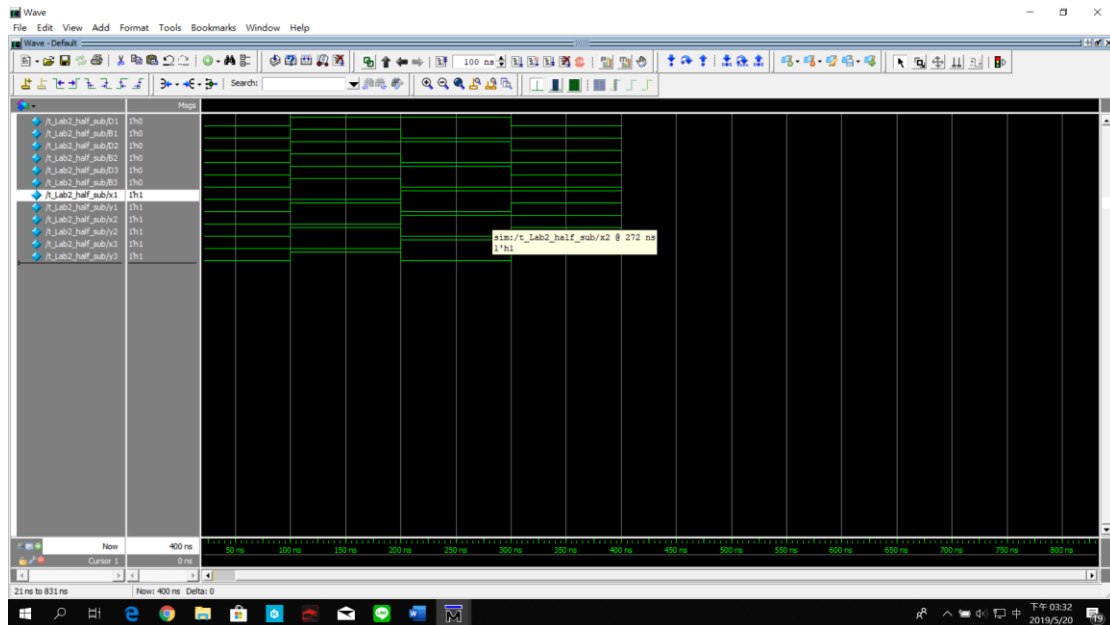
布林代數式

$$D = x' y + x y' = x \oplus y; B = x' y;$$

邏輯電路圖



波形圖



三者的波形圖和真值表皆相同且正確。

(2)

### 如何以半減器建構全減器

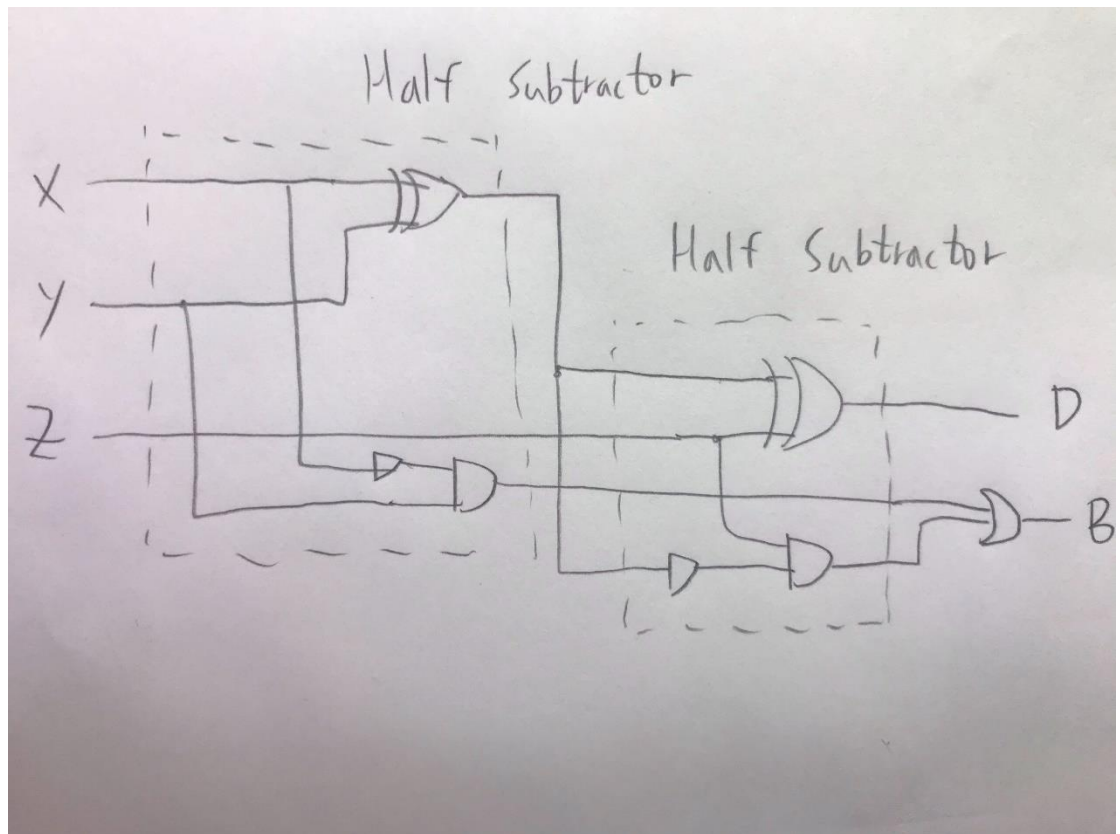
由全減器輸出的布林函數化簡後，

$$D = x' y' z + x' y z' + x y' z' + x y z \text{ 可化簡成 } (x \oplus y) \oplus z$$

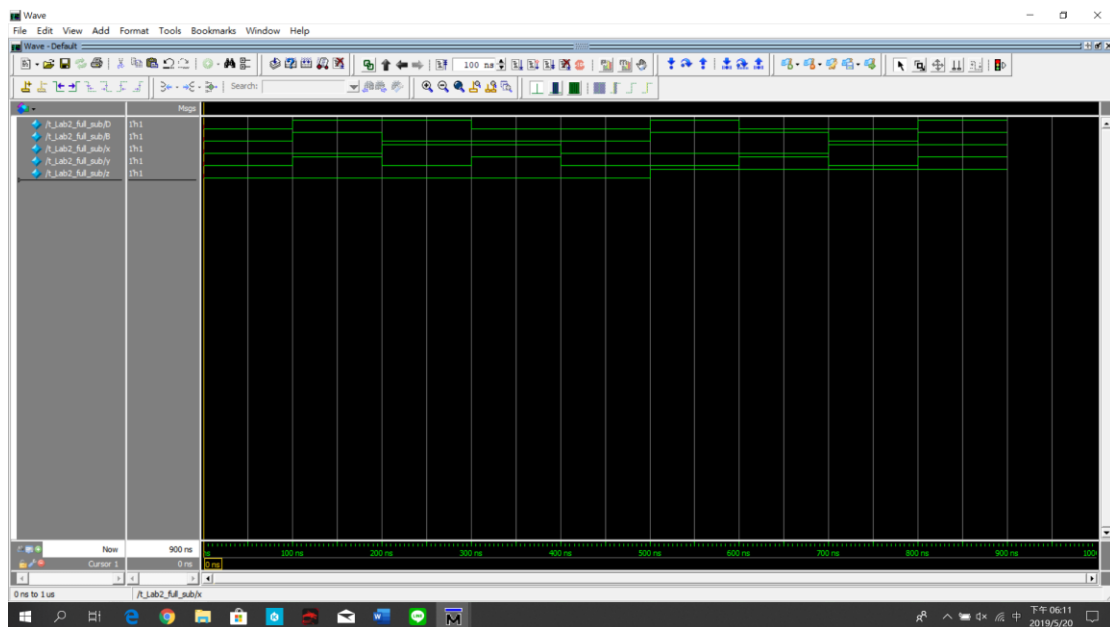
$$B = x' z + x' y + y z \text{ 可化簡成 } (x \oplus y)' z + x' y$$

能以兩個半減器和一個 or 閘來表示。第一個半減器的輸入為 x, y，輸出為  $x \oplus y$ ,  $x' y$ ， $x \oplus y$  與 z 取 xor 後可得 D， $x \oplus y$  與 z 取 and 再對  $x' y$  取 or 可得到 B。

### 電路方塊圖



波形圖



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

1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

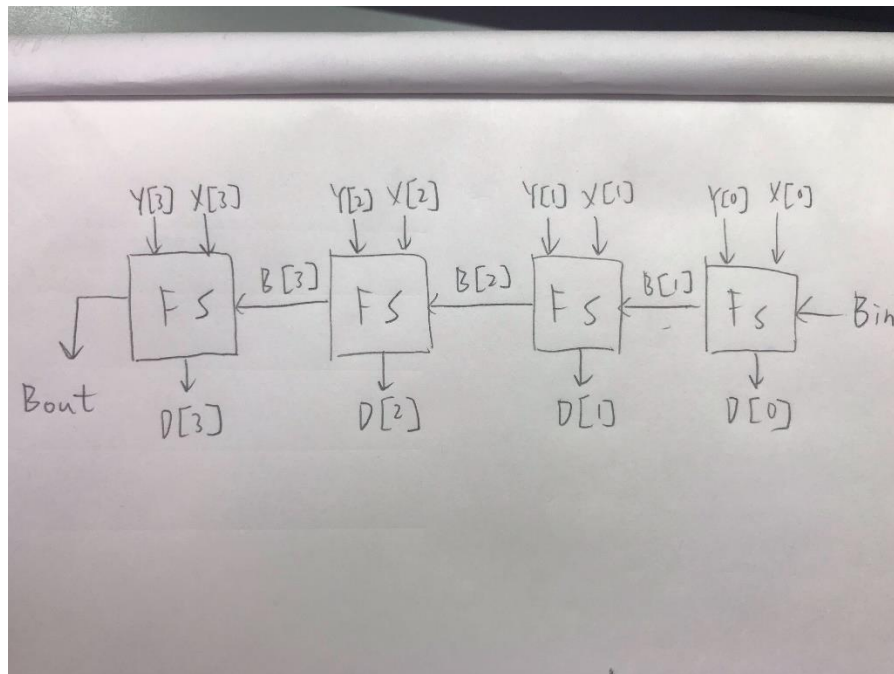
由真值表來看波形圖正確

(3)

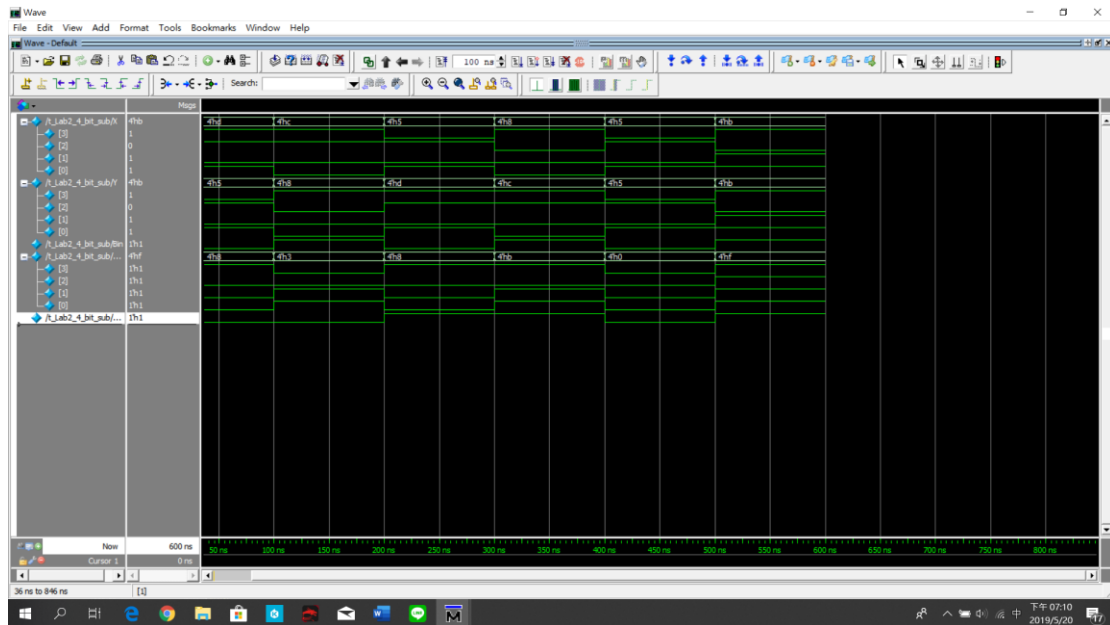
### 電路設計

四個全減器並排後，將第一個全減器的輸出(Bout)接上第二個全減器的輸入(Bin)，可連接成四位元的減法器。

### 電路方塊圖



### 波形圖



X	Y	Bin	Diff	Bout
1101	0101	0	1000	0
1100	1000	1	0011	0
0101	1101	0	1000	1
1000	1100	1	1011	1
0101	0101	0	0000	0
1011	1011	1	1111	1

由真值表來看波形圖正確

(4)

### Boolean fuction

$i = 0, 1, 2, 3$ ，令

$$P[i] = x[i] \oplus y[i]; G[i] = x[i] \cdot y[i]; B[i+1] = G[i] + P[i] \cdot B[i]; D[i] = B[i] \oplus P[i]$$

$B[1:4]$  展開後得到：

$$B[1] = G[0] + P[0] \cdot B[0],$$

$$B[2] = G[1] + P[1] \cdot G[0] + P[1] \cdot P[0] \cdot B[0],$$

$$B[3] = G[2] + P[2] \cdot G[1] + P[2] \cdot P[1] \cdot G[0] + P[2] \cdot P[1] \cdot P[0] \cdot B[0],$$

$$B[4] = G[3] + P[3] \cdot G[2] + P[3] \cdot P[2] \cdot G[1] + P[3] \cdot P[2] \cdot P[1] \cdot G[0] + P[3] \cdot P[2] \cdot P[1] \cdot P[0] \cdot B[0];$$

波形圖



X	Y	Bin	Diff	Bout
1101	0101	0	1000	0
1100	1000	1	0011	0
0101	1101	0	1000	1
1000	1100	1	1011	1
0101	0101	0	0000	0
1011	1011	1	1111	1

由真值表來看波形圖正確

(5)

### 電路設計

將 A[0]~A[7]用 SOP 慢慢化解得出最簡式後，所得出的結果便是答案。

最簡積之和

$$A[6] = D_3' D_2' D_1' + D_3' D_2' D_1 + D_3' D_2' D_0 + D_3' D_1' D_0'$$

$$A[5] = D_3' D_2' D_1' + D_3' D_1' D_0' + D_3' D_2' D_1 + D_3' D_2' D_0'$$

$$A[4] = D_2' D_1' D_0' + D_3' D_1' D_0'$$

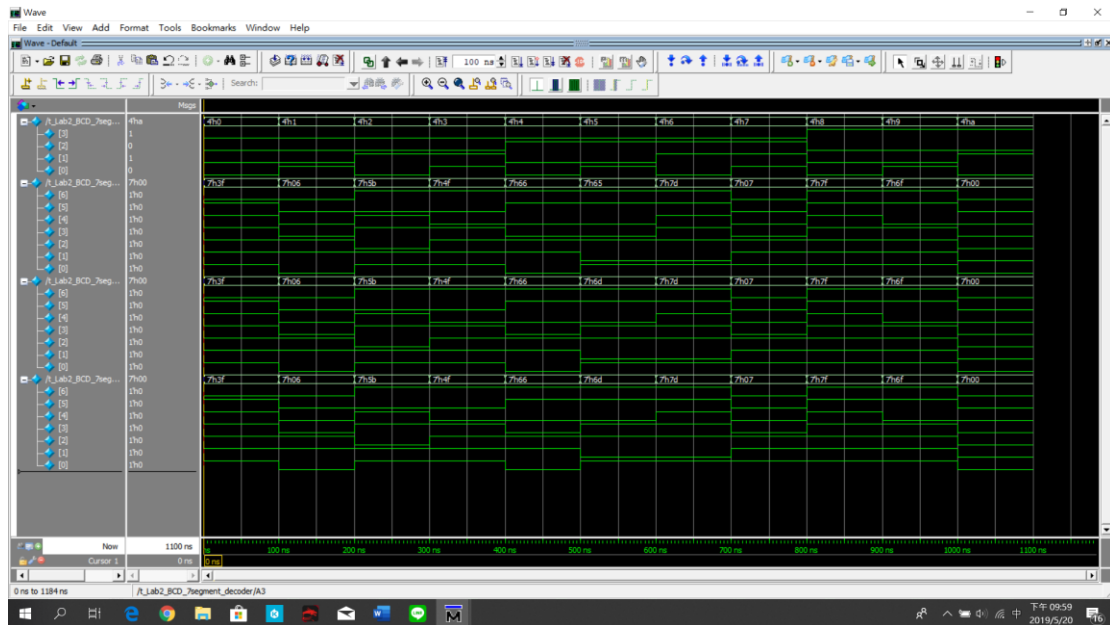
$$A[3] = D_3' D_2' D_1' + D_3' D_2' D_1 + D_3' D_1' D_0' + D_2' D_1' D_0' + D_3' D_2' D_1' D_0'$$

$$A[2] = D_3' D_1' + D_2' D_1' + D_3' D_2 + D_3' D_0$$

$$A[1] = D_3' D_2' + D_2' D_1' + D_3' D_1' D_0' + D_3' D_1' D_0$$

$$A[0] = D_3' D_1 + D_3' D_2' D_1' + D_2' D_1' D_0' + D_3' D_2' D_0$$

波形圖



(6)

這次的作業讓許久沒碰 verilog 的我有了一個好機會可以熟悉一下如何撰寫 verilog，在過程中我覺得最困難的是 decoder 的部分，光化簡 sop 就花了不少時間，再把他打成 verilog 也花了很久時間，而且只要化簡稍有不甚就要檢查許久，十分的累。