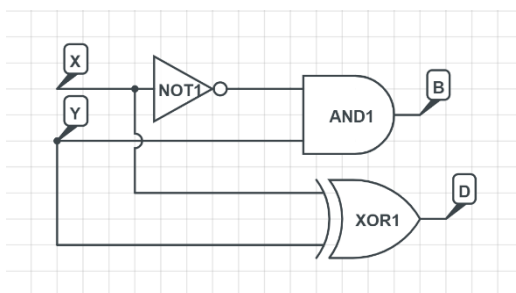


(1) 半減器真值表：

X	Y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

布林代數式： $B = X'Y$      $D = X \text{ xor } Y$

邏輯電路圖：



波形圖：

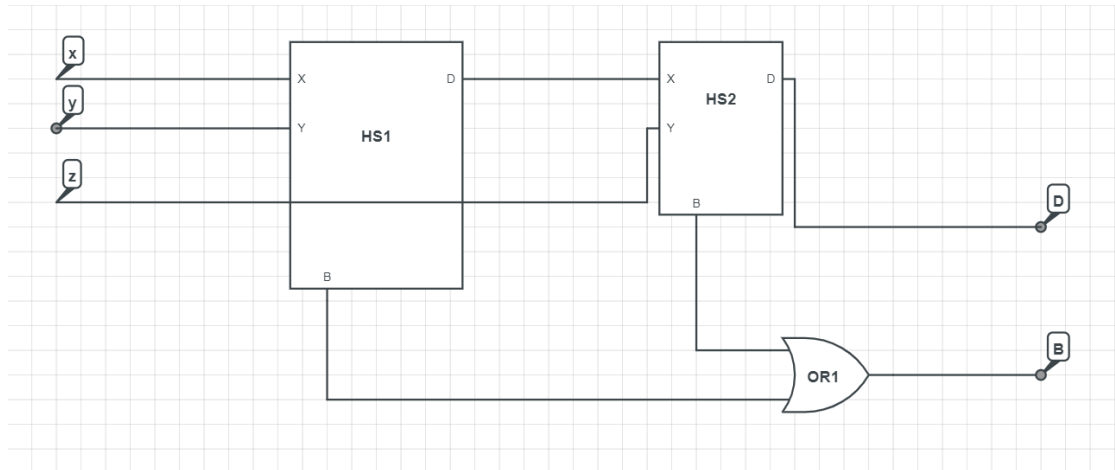


波形圖顯示結果與真值表相符

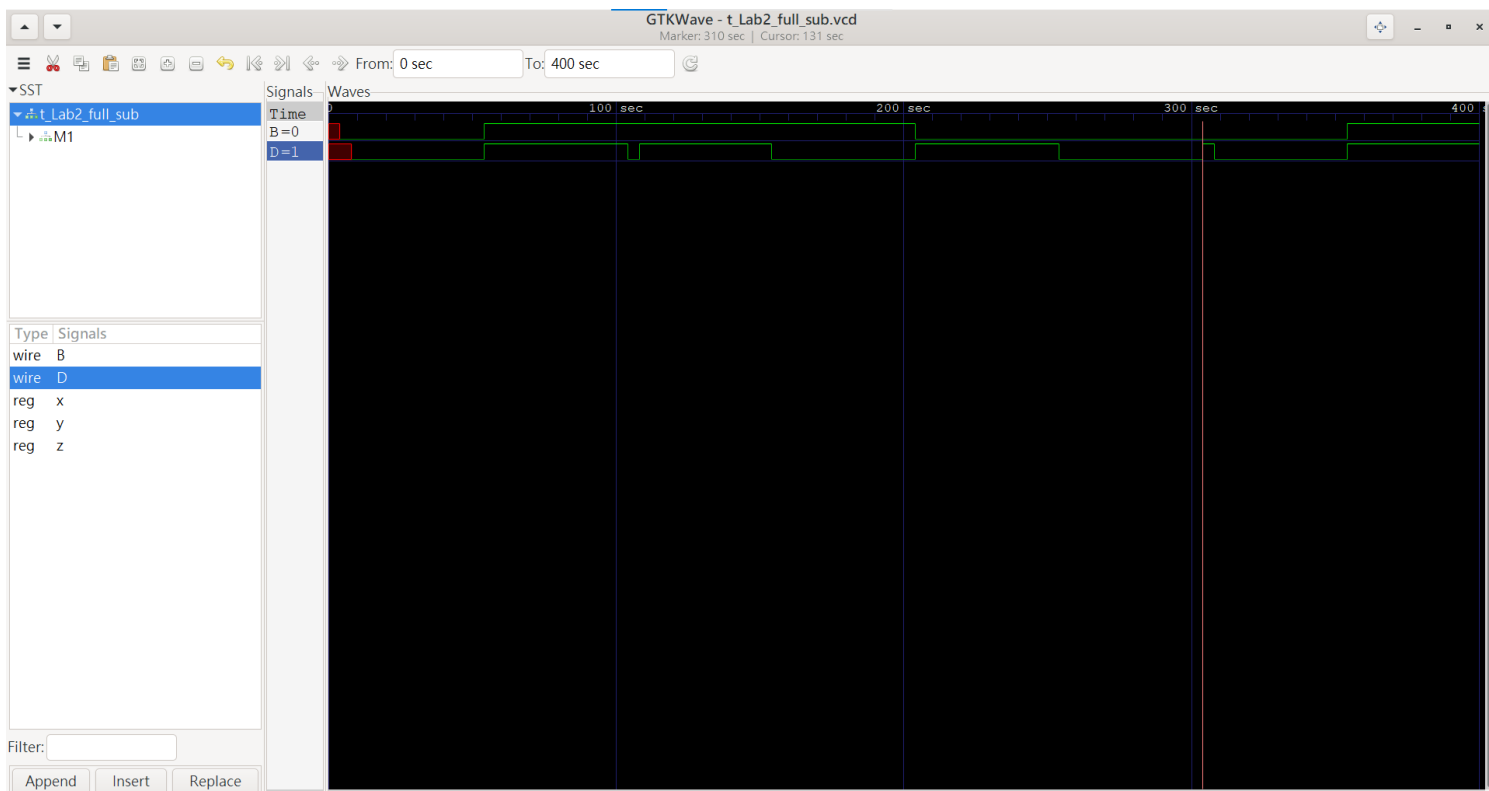
Propagation delay:    B:2 time units    D:4 time units

- (2) 以第一個半減器輸出的 D 拿到第二個半減器去當被減數，減出來的 D 就會變成全減器的 D，而第一個半減器輸出的 B 和第二個半減器輸出的 B 做 or 可以得到全減器有沒有借位，也就是全減器的 B。

電路方塊圖：



波形圖：

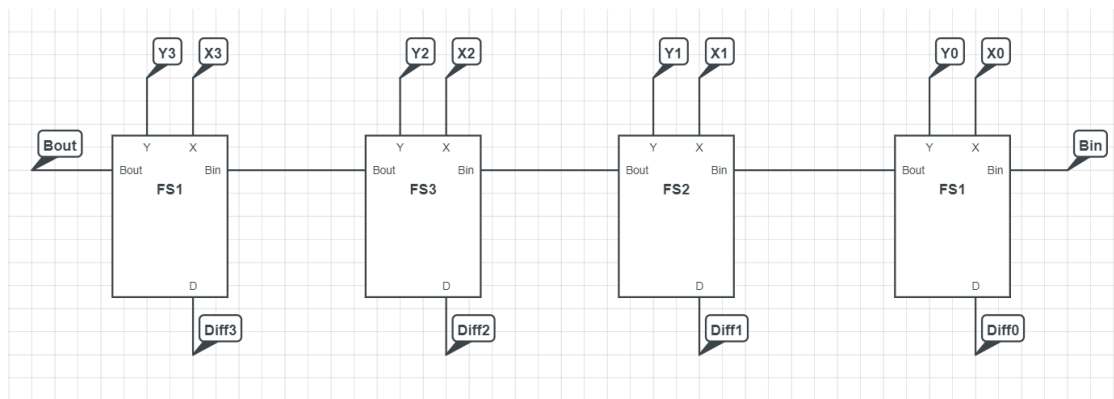


波形圖顯示結果與真值表相符

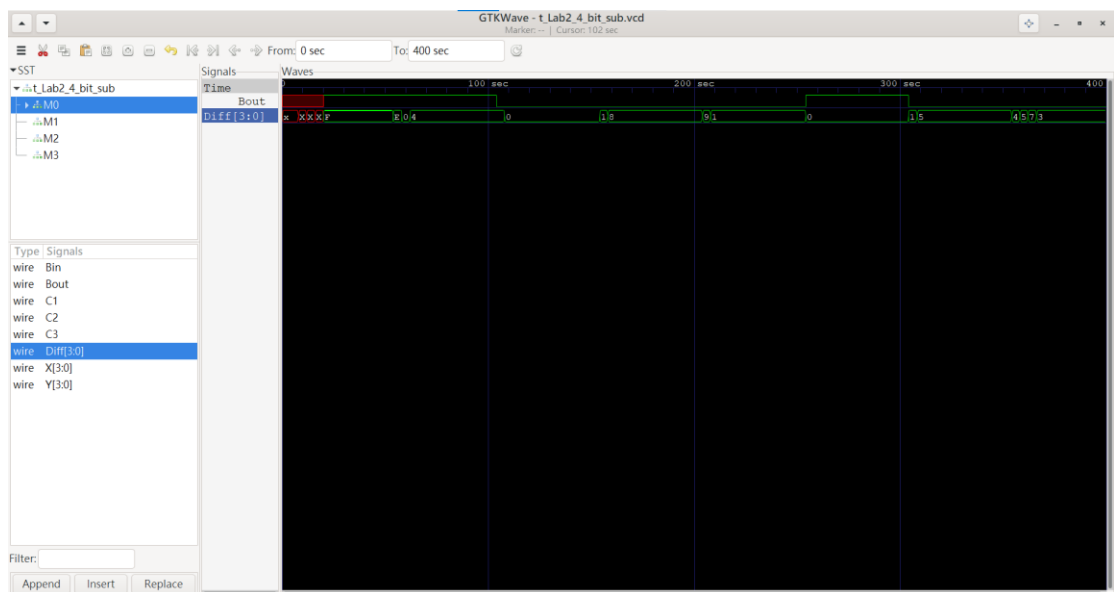
Propagation delay: B:4 time units D:8 time units

- (3) 將 4 個全減器串在一起，每個全減器的輸出 D 依序為 D0、D1、D2、D3，而輸出 B 將成為下一個全減器的 Bin，最後一個全減器的輸出 B 將會成為 Bout。

電路方塊圖：



波形圖：



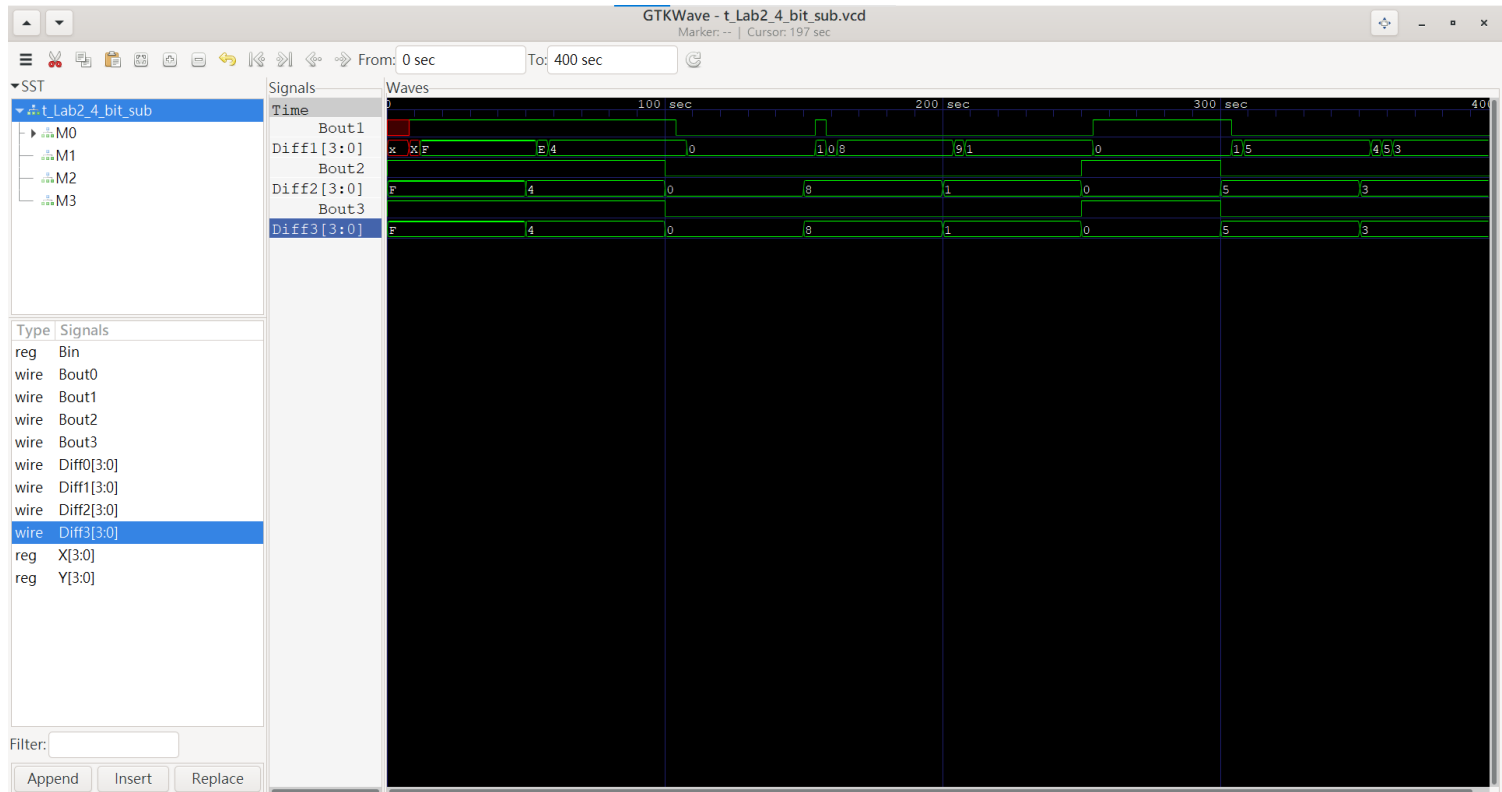
波形圖顯示結果與真值表相符

Propagation delay: Bout:20 time units D:20 time units

- (4) 每個 Bin 不須等到上一個全減器的 Bout 做完就可以得到值，所以可以減少 propagation delay

布林代數式： $P_i = X_i \text{ xor } Y_i$     $G_i = X_i'Y_i$     $B_{i+1} = G_i + P_i'B_i$     $D_i = P_i \text{ xor } B_i$

波形圖：



1:gatelevel 2:dataflow 3:behavior

因為 gatelevel 有 propagation delay 但是在沒有 delay 時的輸出值三者是相同的，因此波形圖是正確的。

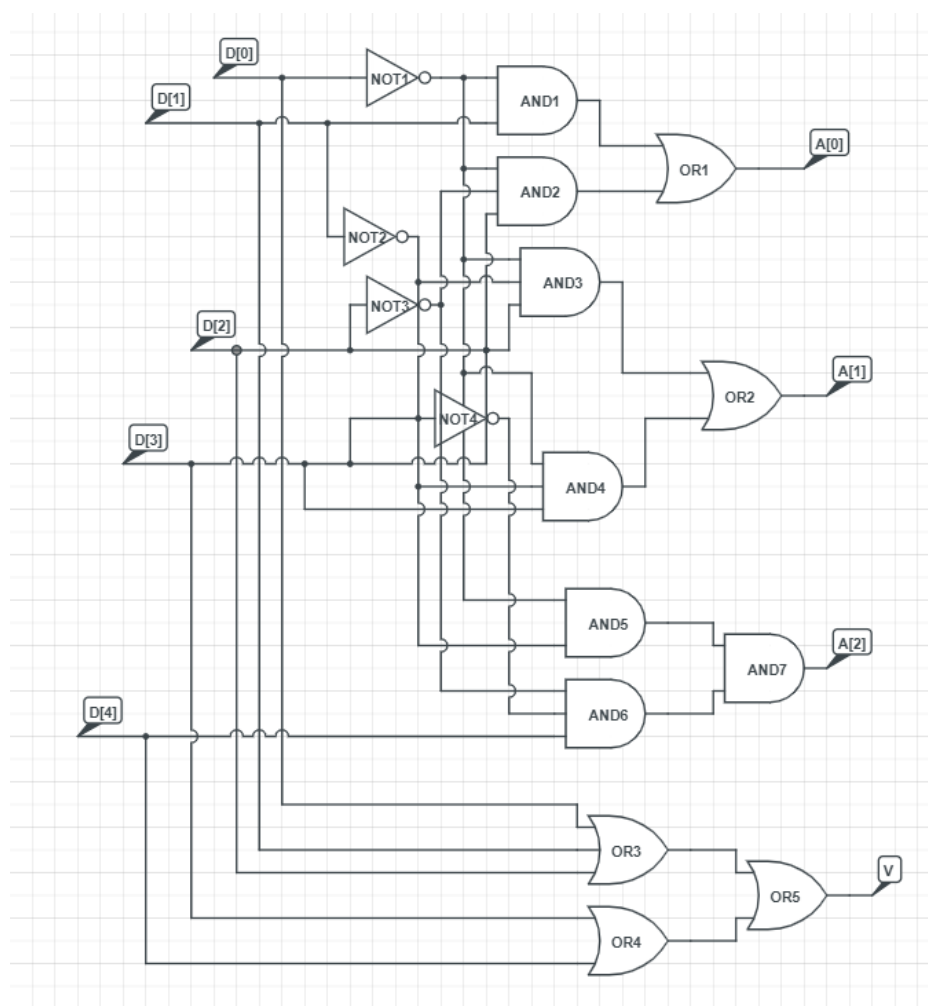
Gate-level modeling 的 propagation delay: Bout:8 time units D:12 time units

- (5) 因為是低位順序優先，所以從最小的開始判斷，如果有，其他就 don't care，如果全部都沒有的話 V 就是 0，反之為 1。

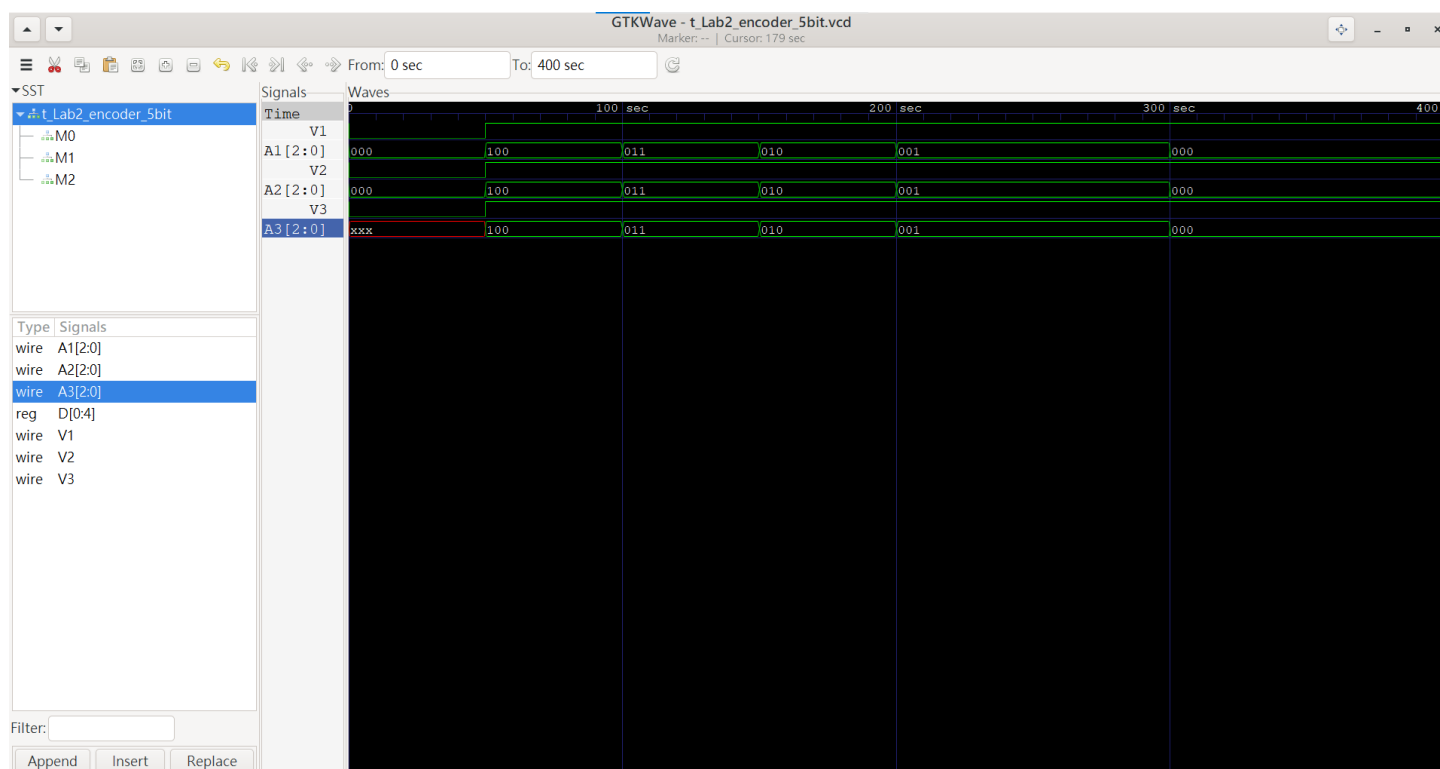
$A[0] = D[0]'D[1] + D[0]'D[2]'D[3]$     $A[1] = D[0]'D[1]'D[2] + D[0]'D[1]'D[3]$

$A[2] = D[0]'D[1]'D[2]'D[3]'D[4]$

電路方塊圖：



波形圖：



1:gatelevel 2:dataflow 3:behavior

除了  $V=0$  時 gatelevel 和 dataflow 會不定、behavior 會顯示 don't care 外，三者的波形圖結果一樣，故結果正確。

- (6) 這次的 lab 感覺比上次複雜，尤其是畫電路方塊圖要花更多時間，還有 gatelevel modeling 只要寫錯就很難找錯的地方，需要 debug 很久。