(1)

半減器的最終結果是由輸入的x、y值來決定得到的D值和是否要進行借位的B值，所以由此概念可得到以下的真值表:

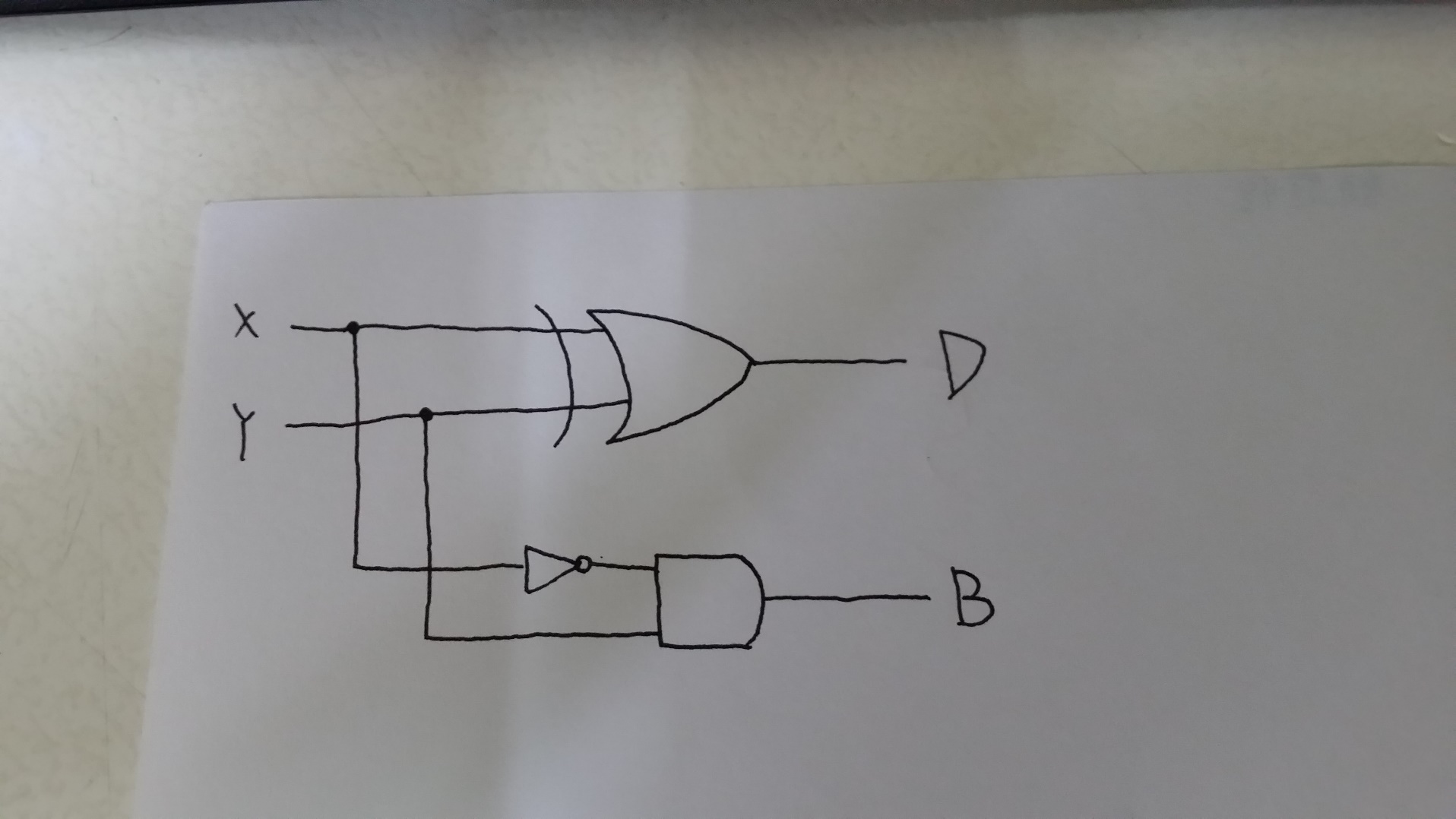
|  |  |  |  |
| --- | --- | --- | --- |
| x | y | B | D |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 |

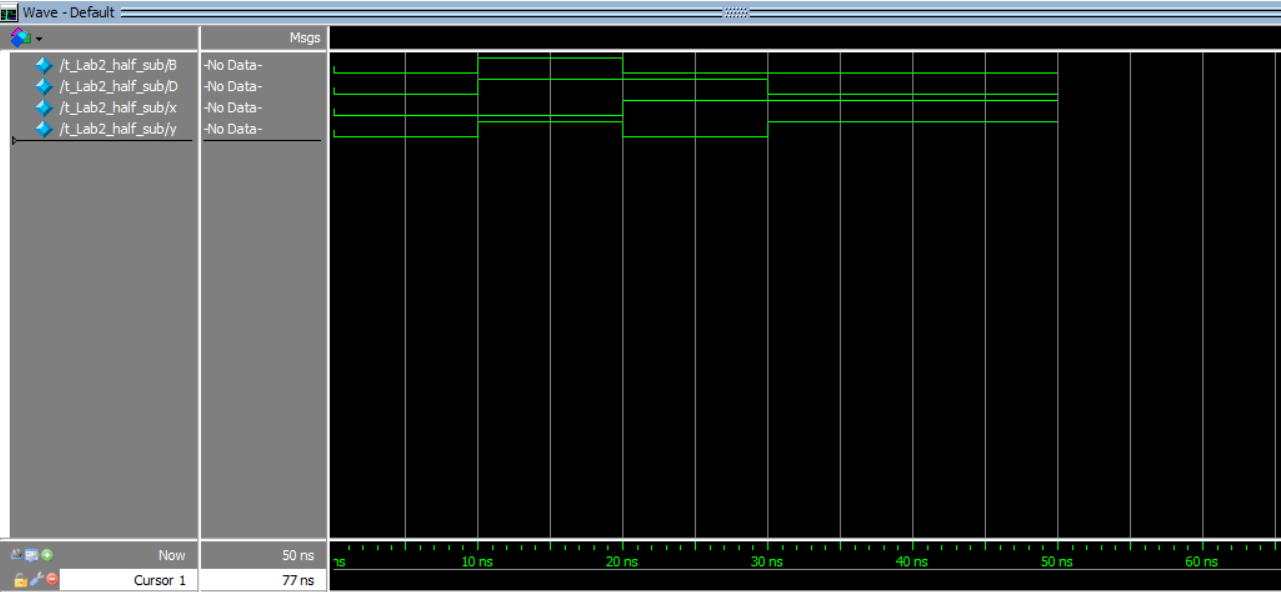
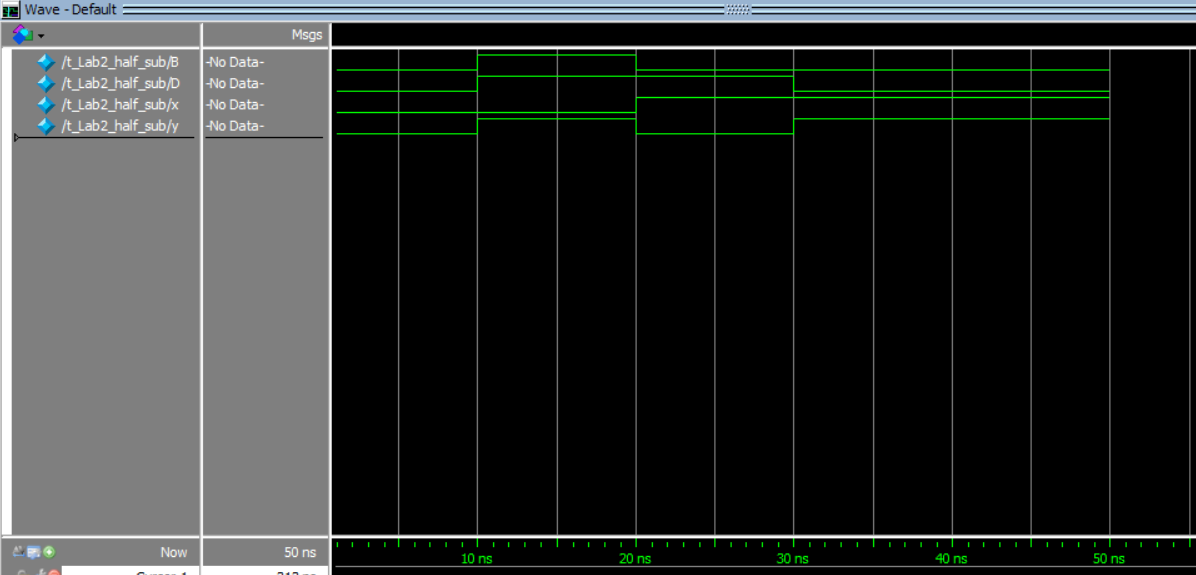
藉由真值表可推出B和D的布林代數式:

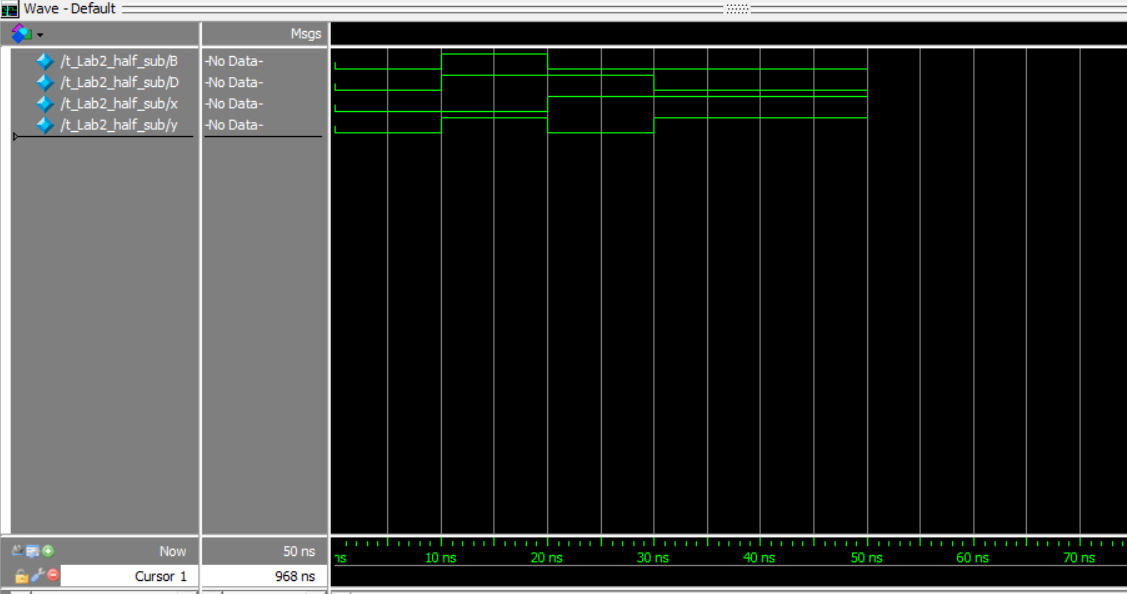
B = (!x) & y ;

D = x ⊕ y;

並可繪出以下的電路圖:



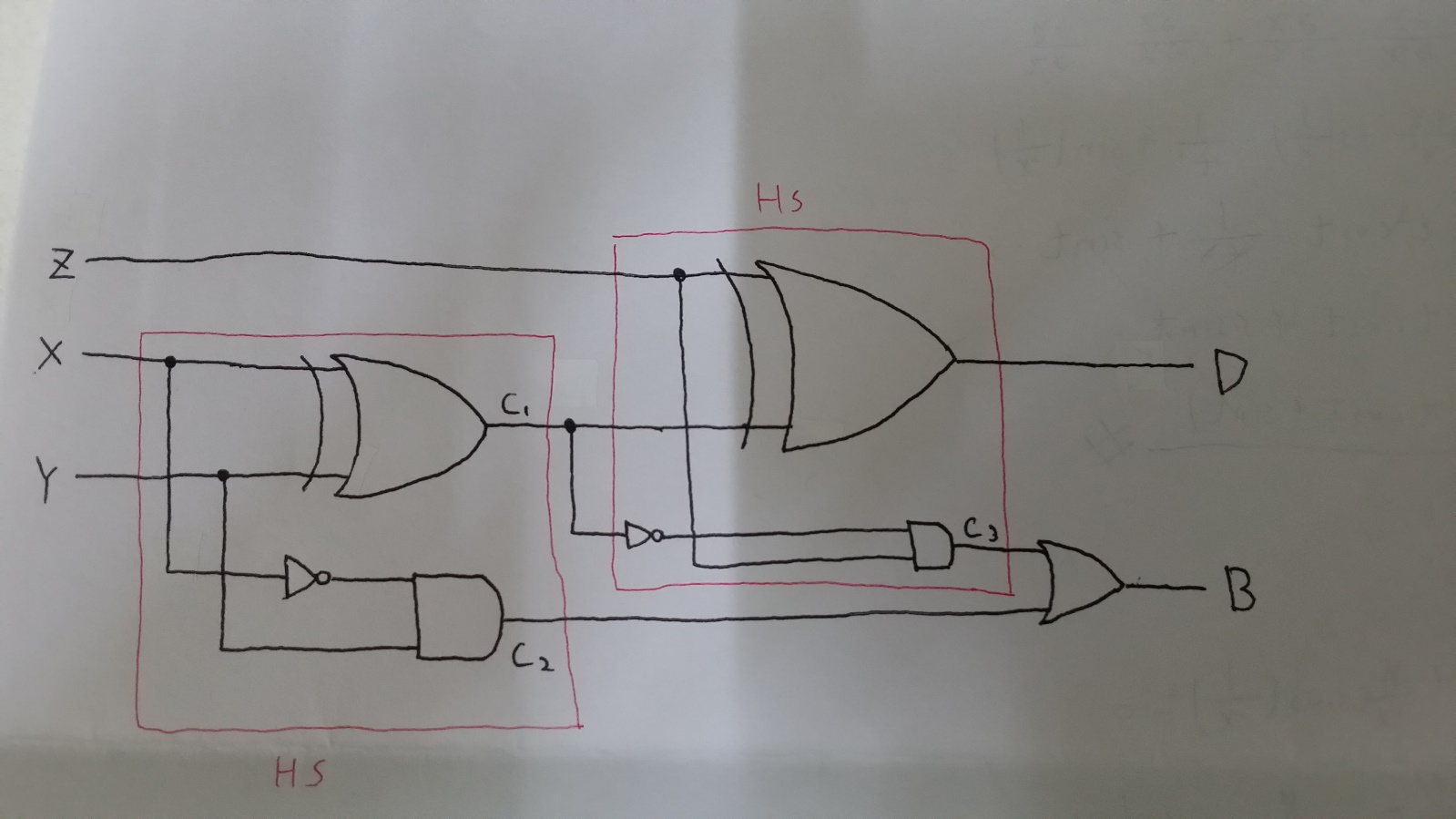
而模擬所得的波形圖:



用三種不同方式寫出的模組進行模擬後，得到的波型圖並無差異，他們的結果也符合上述真值表依不同x、y得到B、D的值一樣，所以這三個模擬結果都是正確的。

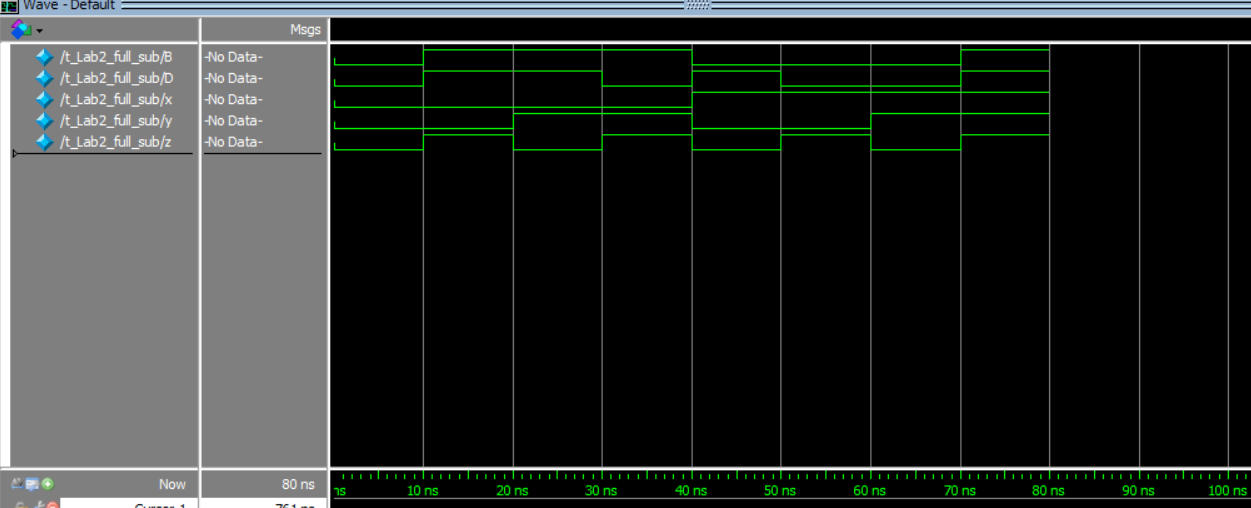
(2)

全減器的電路圖如下:



要建構一個全減器可以利用兩個半減器來組成，先將x,y的數值放入其中一個半減器，就會得到C1和C2，這時再將C1和z放入第二個半減器裡，就能得到D和C3，最後只要將C2和C3放入一個OR的電路器中，就能得到B的值。

波型圖如下:

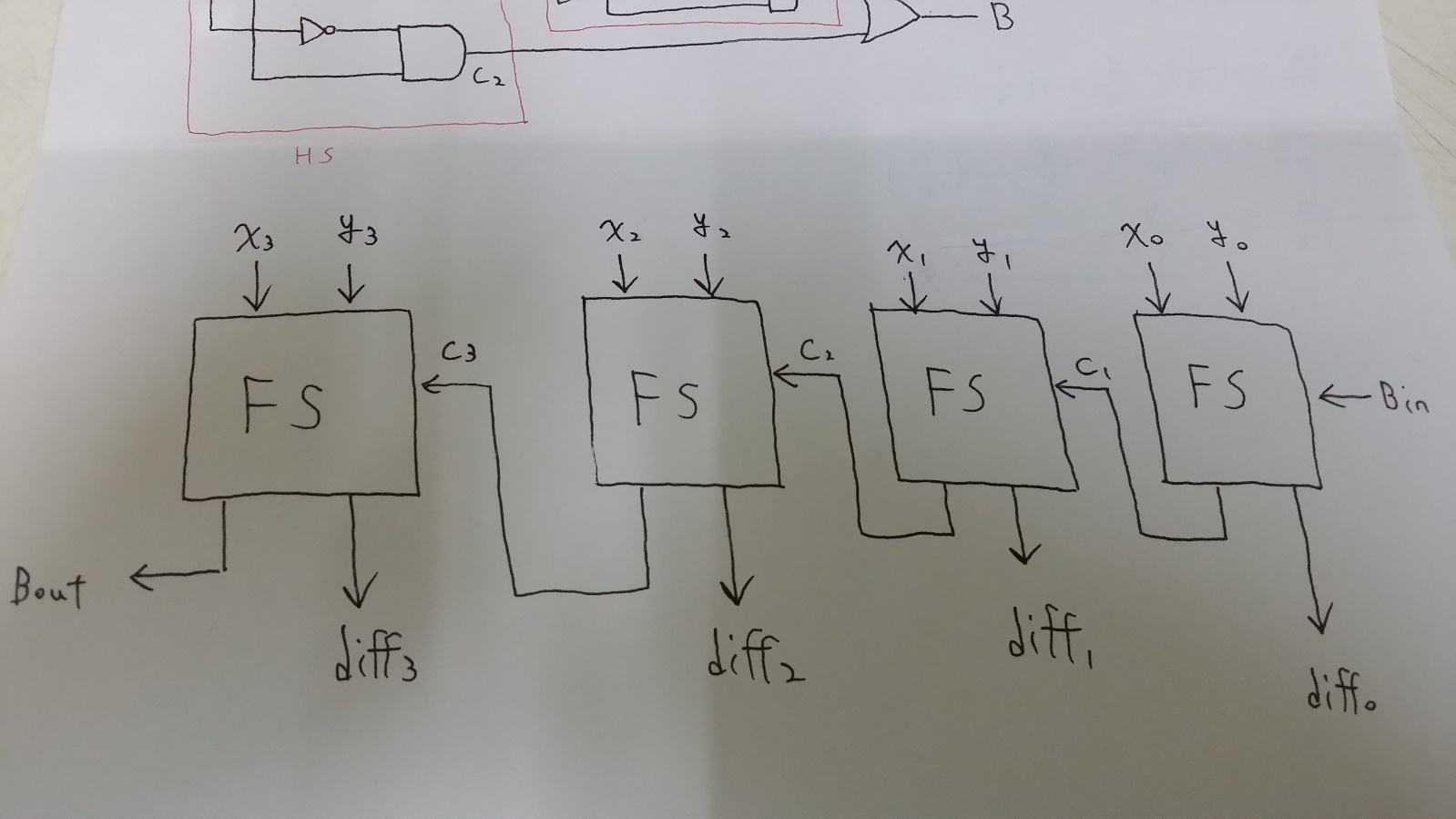


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | y | z | B | D |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

我利用以上的真值表來進行全減器的電路測試，從波型圖的結果來看，我所得到的模擬結果與真值表上B和D依照不同的x、y、z所對應的值完全相同，所以全減器的模擬結果是正確的。

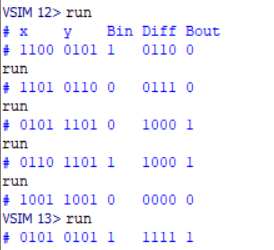
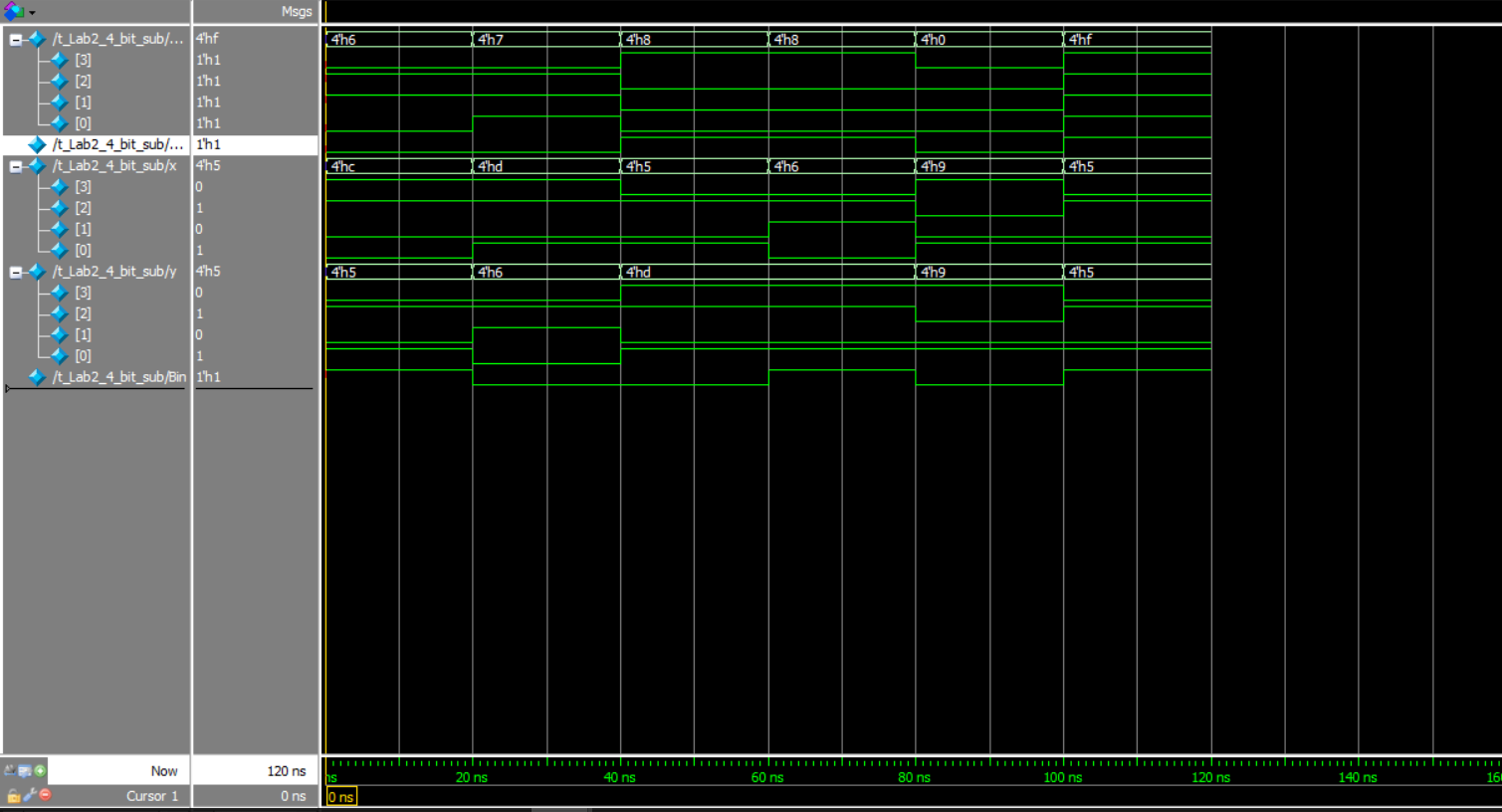
(3)

電路圖如下:



利用第一個全減器和一開始輸入的Bin、x[0]、y[0]來算出餘數diff[0]和是否要向前一位借位的C1，並將得到的C1與x[1]、y[1]再放入第二個全減器中，就能得到餘數diff[1]和是否要借位的C2，將此動作重複進行，就能得到最後的餘數diff[3]和借位數Bout。

波型圖如下:



我在模擬時讓程式顯示每次輸入的x、y、Bin和輸出的Diff和Bout值，並從顯示的數值來對應上面的波型圖，發現對應出來的結果與波型圖上每個數值的變化完全相同，所以我這次的模擬結果是正確的。

(4)

Pi = Xi ⊕ Yi;

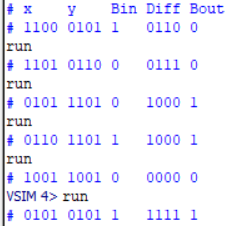
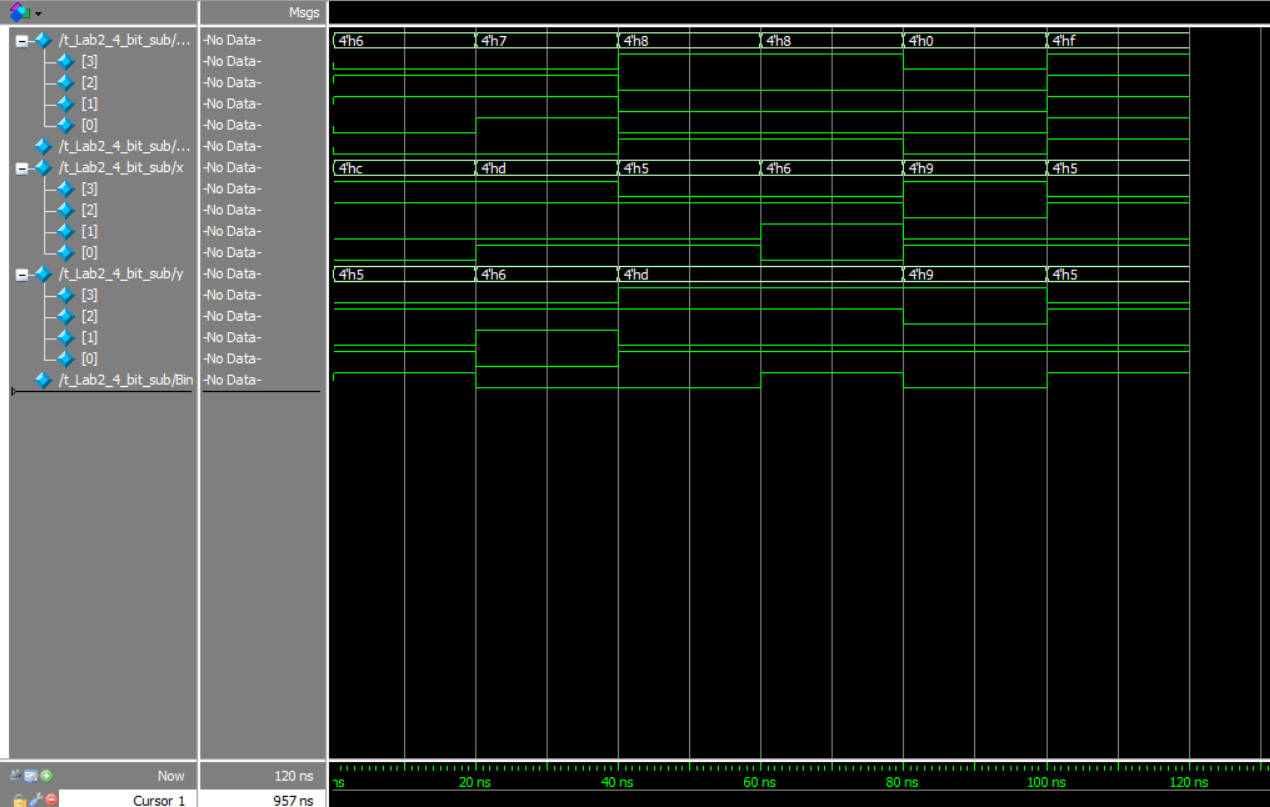
Gi = (!Xi) & Yi;

Bi+1 = Bi & (!Pi) | Gi;

Di = Pi ⊕ Bi;

藉由這種關係就能得到各個位置Di、Bi的數值，把最後得到的數值整理後，就能得到兩個4bit的二進位數字相減後的值。

波型圖:



我使用的測試值與四位元漣波借位減法器所用的相同，而最後得到的結果與上一題的結果完全相同，最後的Bout與每個Diff值都是該X、Y相減後所應得到的數值，所以這次模擬結果沒出問題。

(5)

從優先編碼器之真值表可以推出A[1]、A[0]、V的布林代數式:

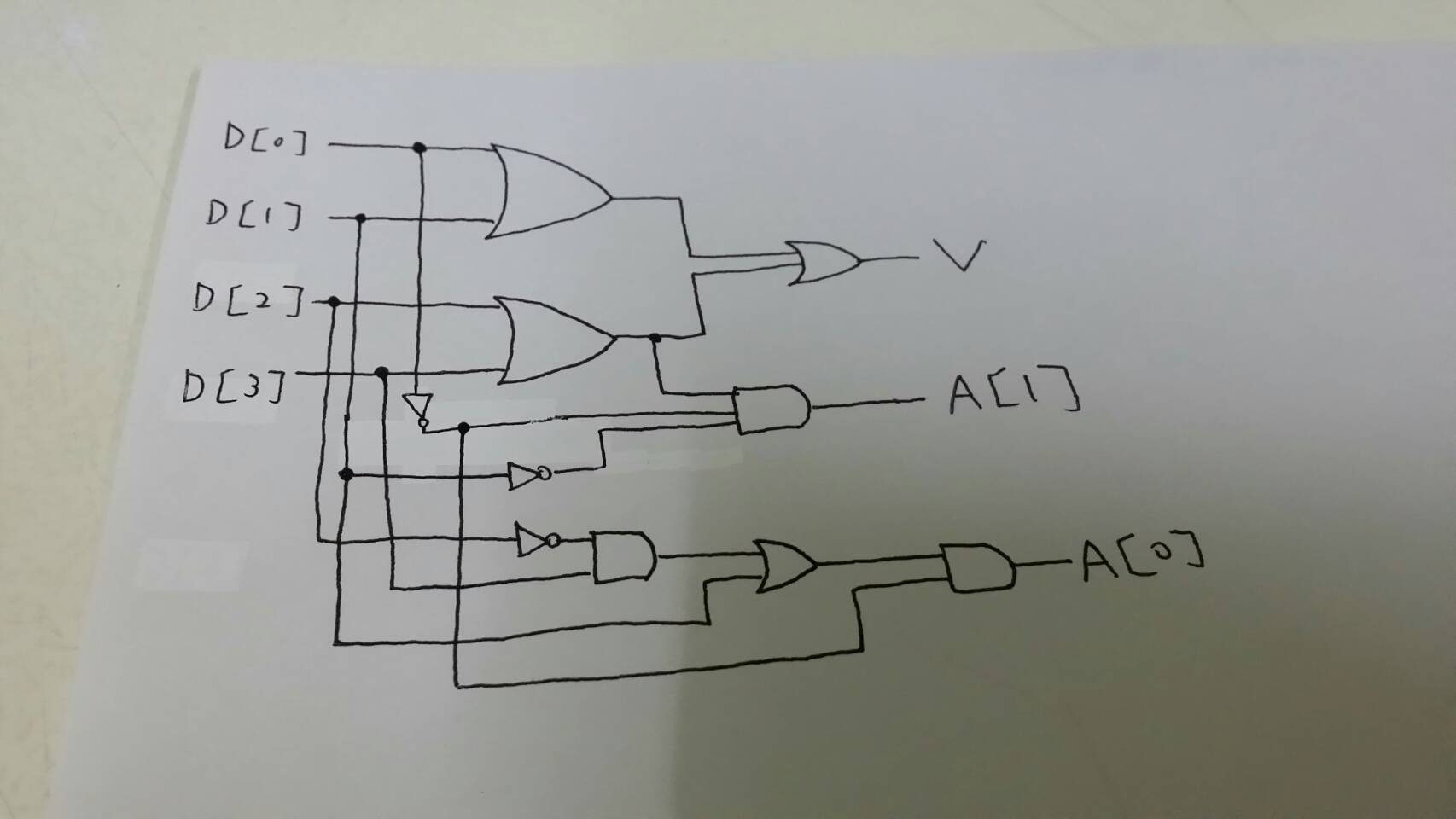
V = D[3] + D[2] + D[1] + D[0];

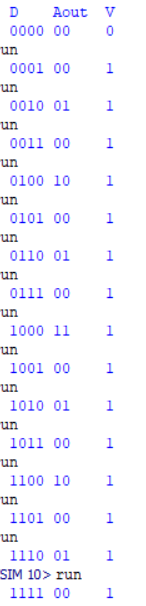
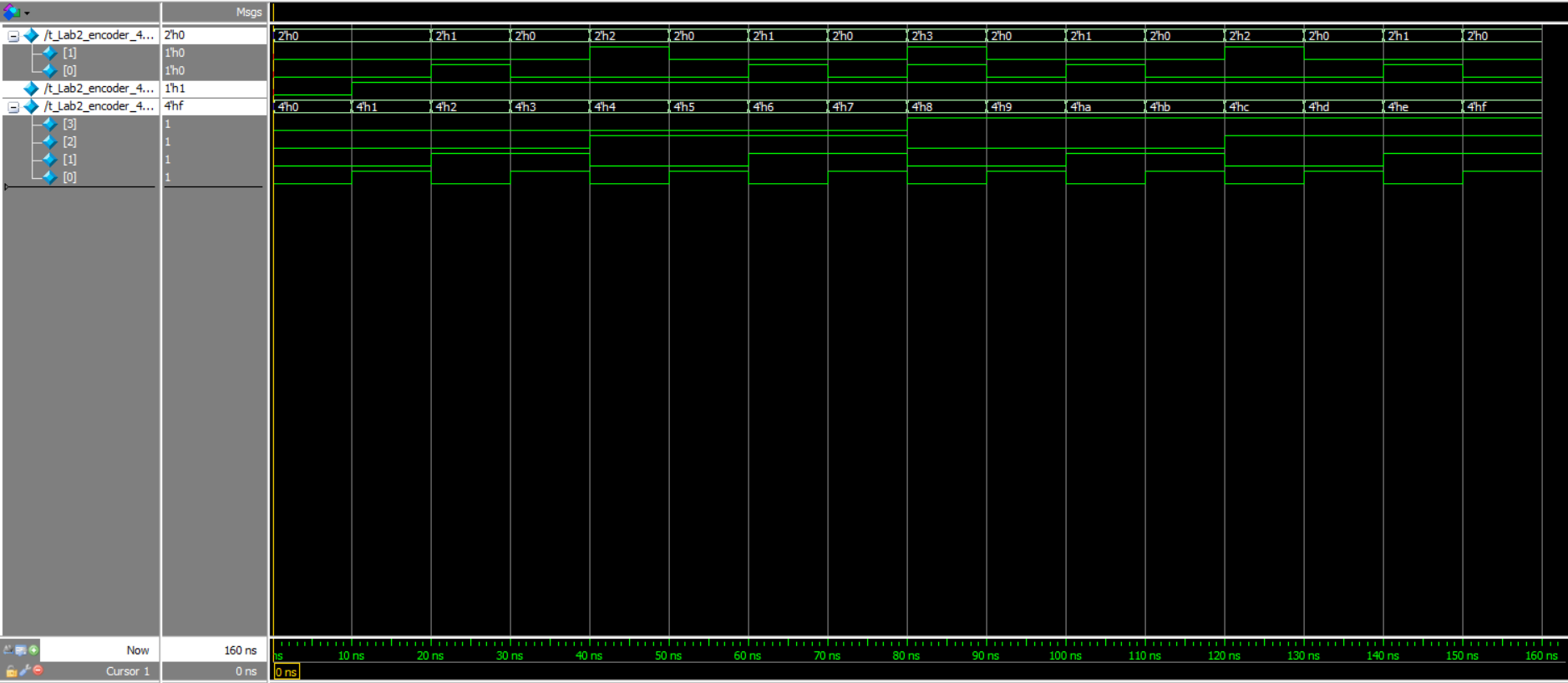
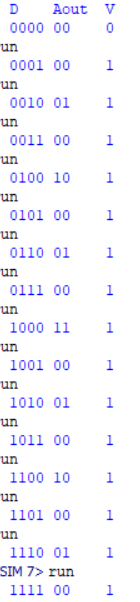
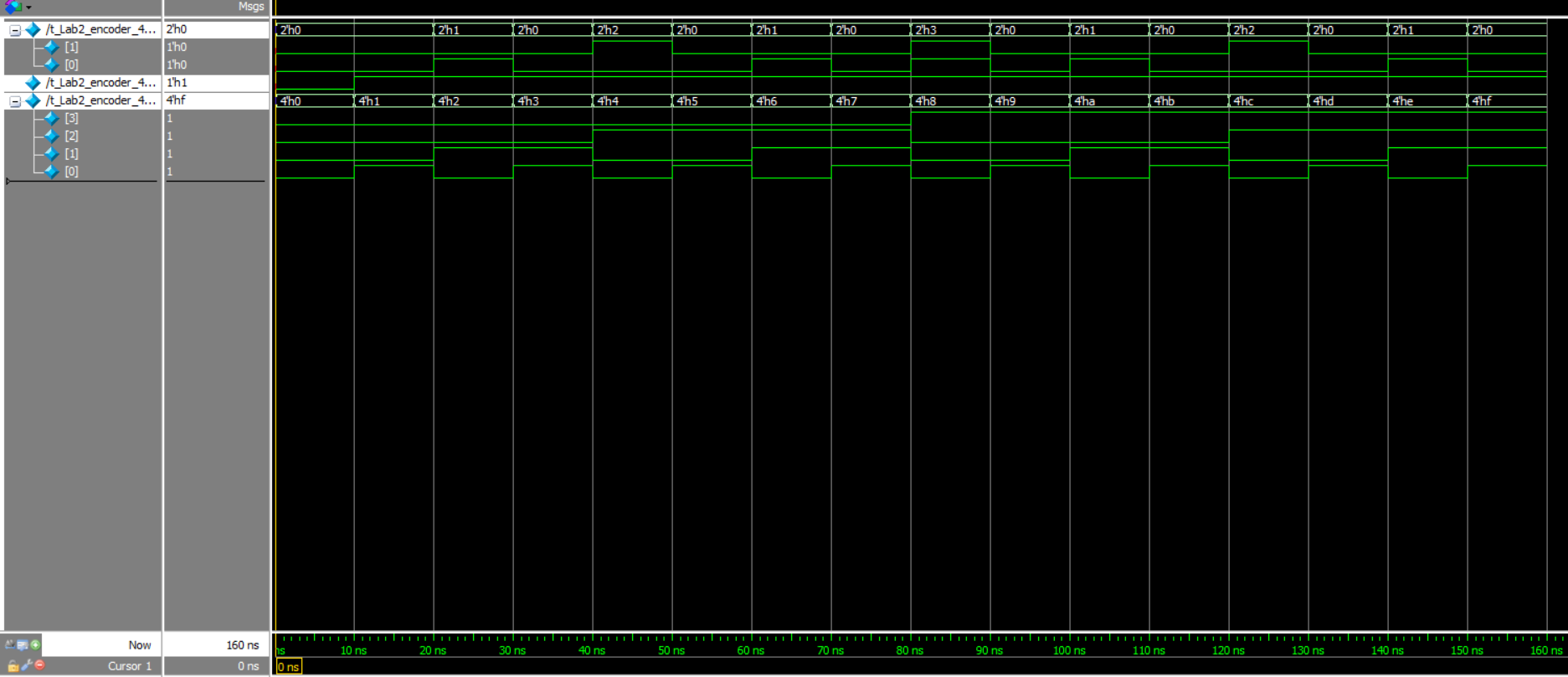
A[0] = (!D[0])&&(D[1] + D[3]&&(!D[2]));

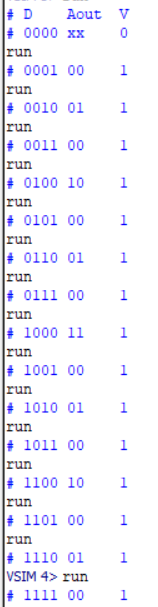
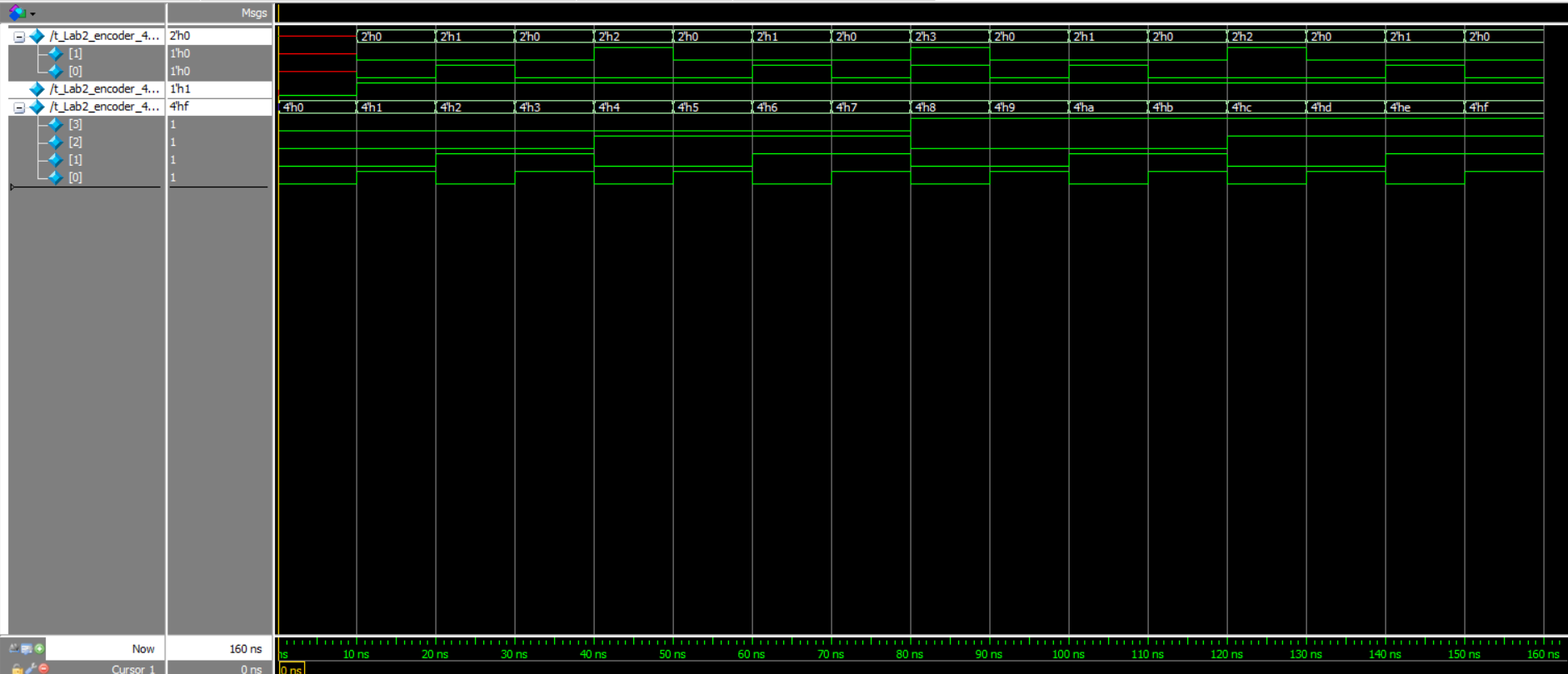
A[1] = (D[2] + D[3])&&(!D[1])&&(!D[0]);

藉由上面的式子就可以寫出優先編碼器的電路。

電路圖:







從真值表來看我這三次所模擬出來的結果都是正確的，但因為用gatelevel和dataflow的方式表示不出當輸入全是0時，輸出應該是X的狀況，所以依照上面所寫的布林代數式，當輸入皆為0時，輸出都會是0，此模擬結果與波型圖的變化一致，所以這些模擬結果都是對的。

(6)

在做Lab1時，我覺得dataflow的方式比其他的方式都來的好用與方便，但到這次的lab我發現當題目中需要出現一些特定值的時候，behavior的方式反而是最好用的，用指定的方式就能得到最後所需的值，我知道這些方式各有各的優點，當在特定時候，其中某一種方式會是解決題目的最佳方法，所以我不應特別排斥某些方法，而應全部都學會，解決問題時，速度才會快且結果正確。