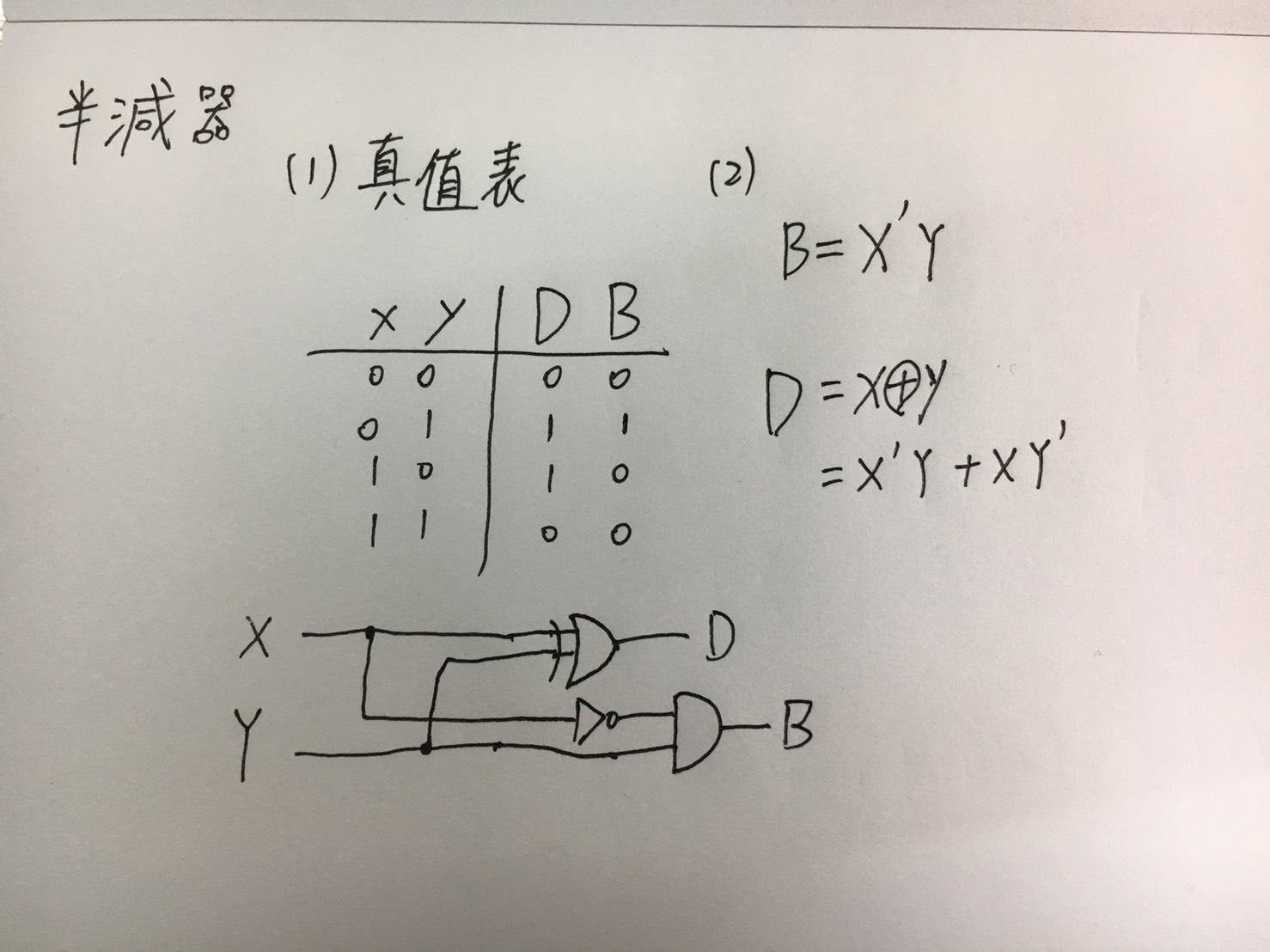
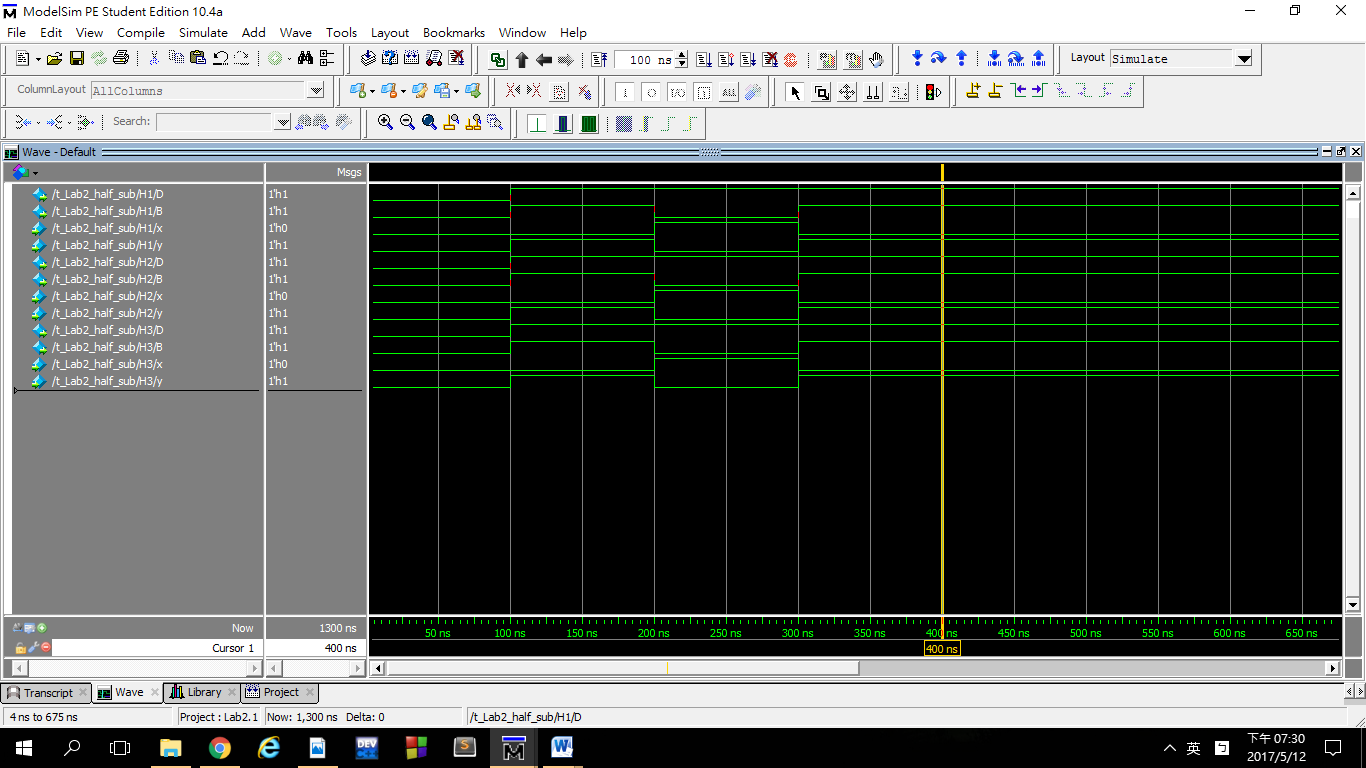
數位電路設計Lab2 0516072 洪立宇

1. 半減器設計:真值表、布林代數式、邏輯電路圖

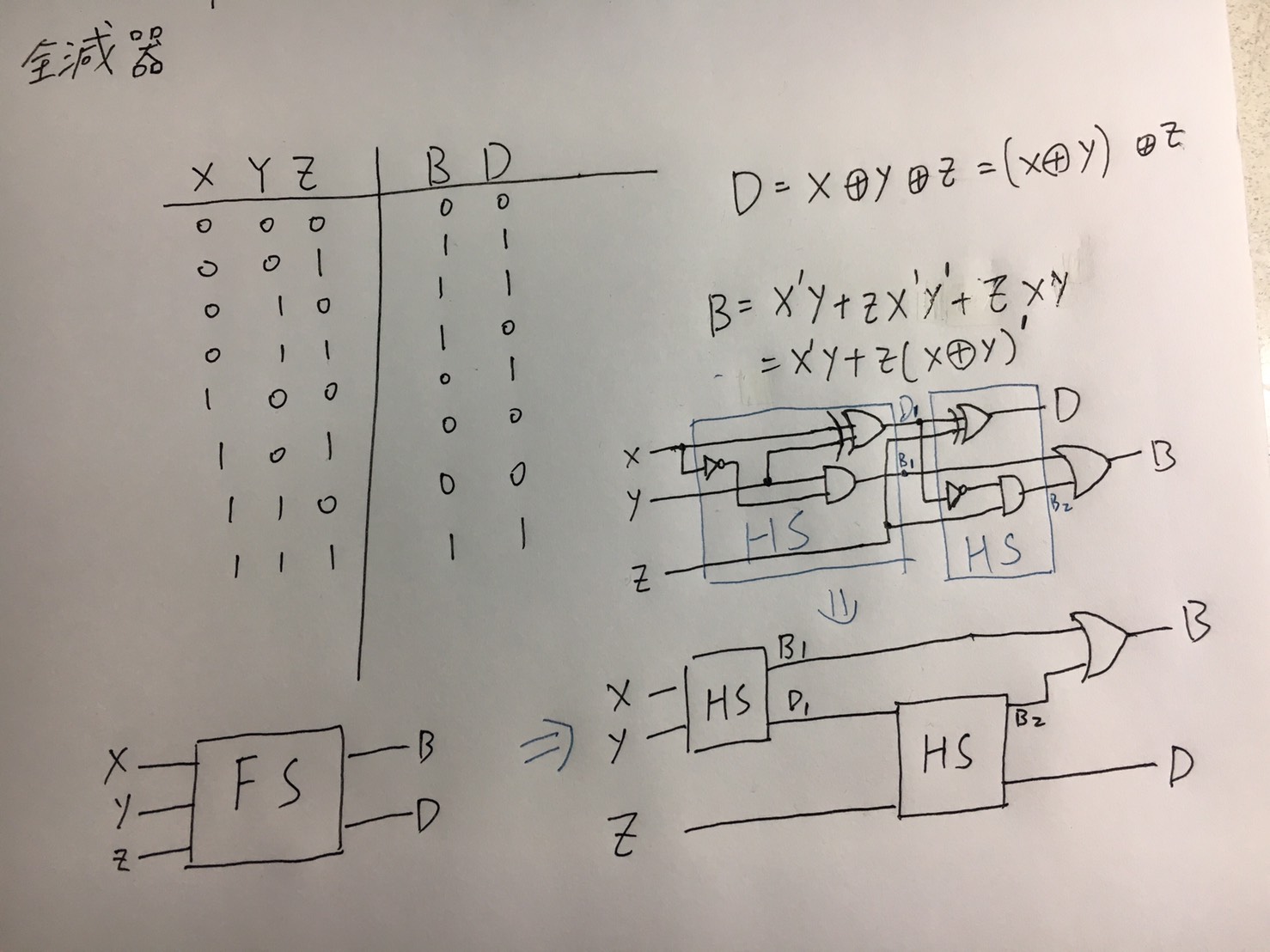
2個輸入x,y 和 Difference(差值) Borrow-out (借位)



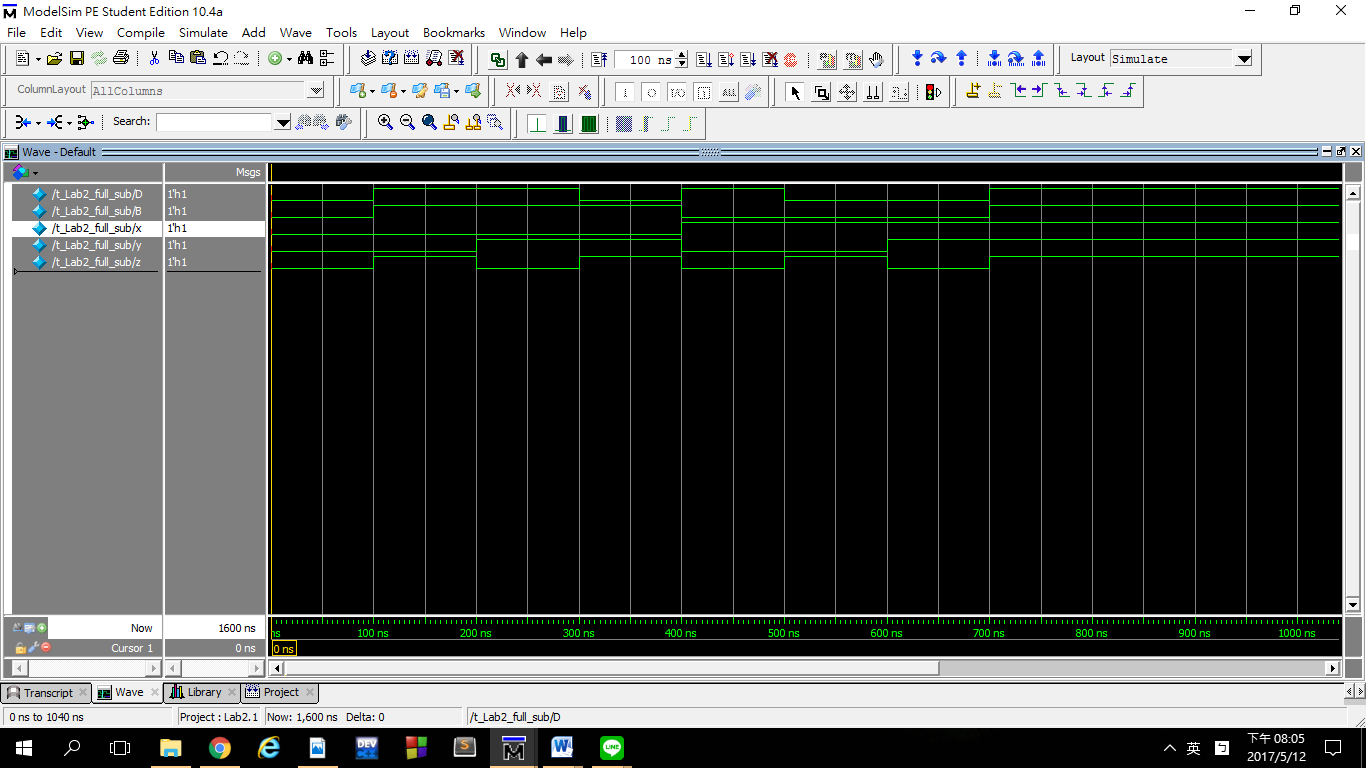


H1是Gatelevel的測試圖，H2是dataflow，H3是behavior，三種寫法出來的波型圖都是一樣的，所以正確。

1. 用半減器建構全減器

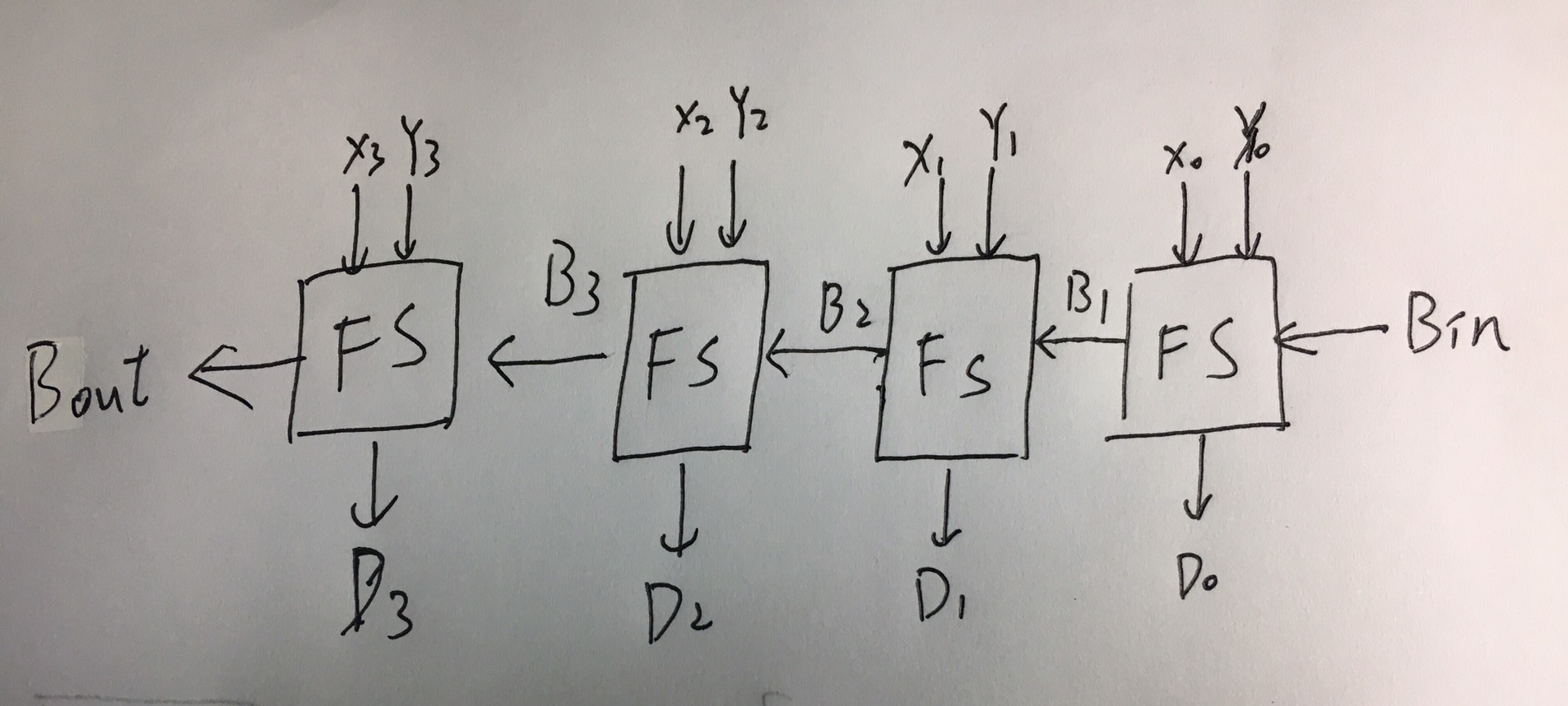


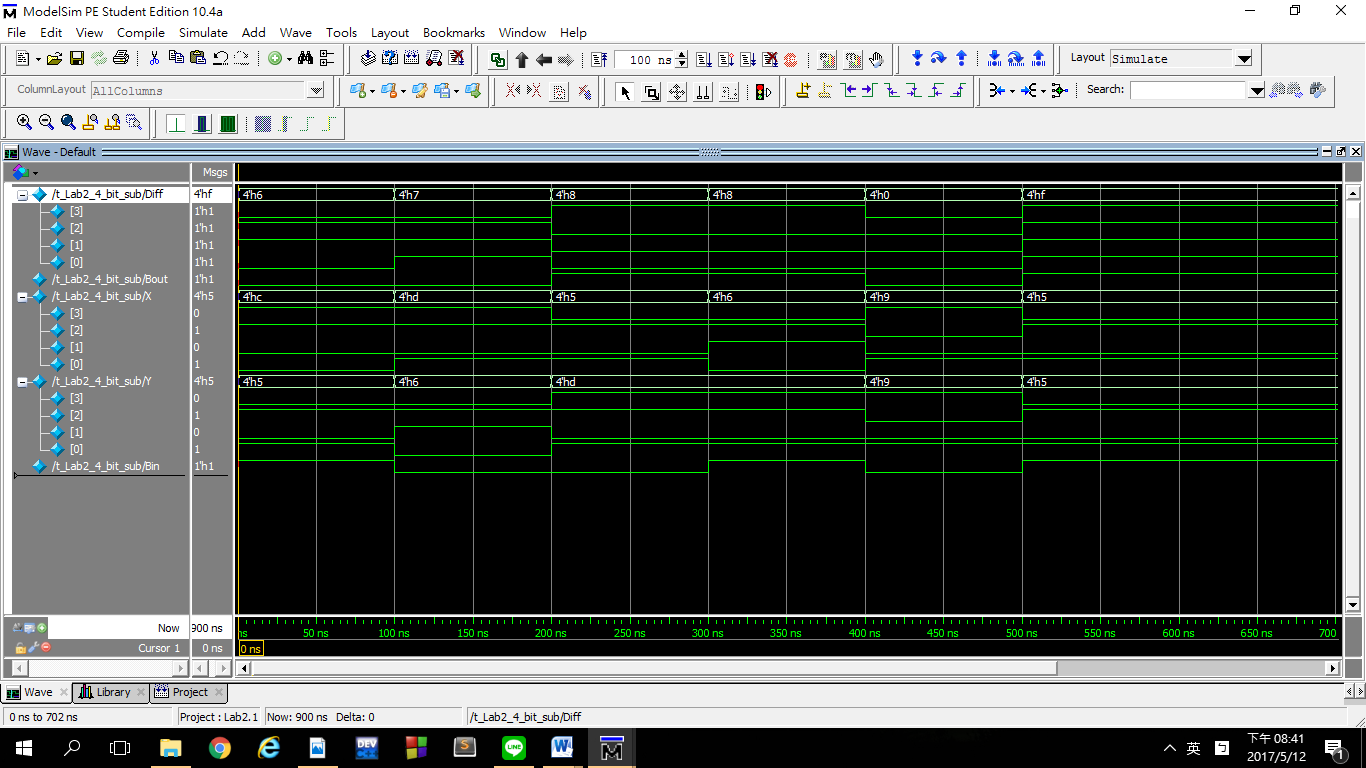
首先先畫出全減器真值表，然後用真值表寫出D,B的布林代數式，然後畫出此邏輯電路圖，會發現全減器會是用兩個半減器加一個OR-Gate組成，因此可以用兩個半減器來做一個全減器。

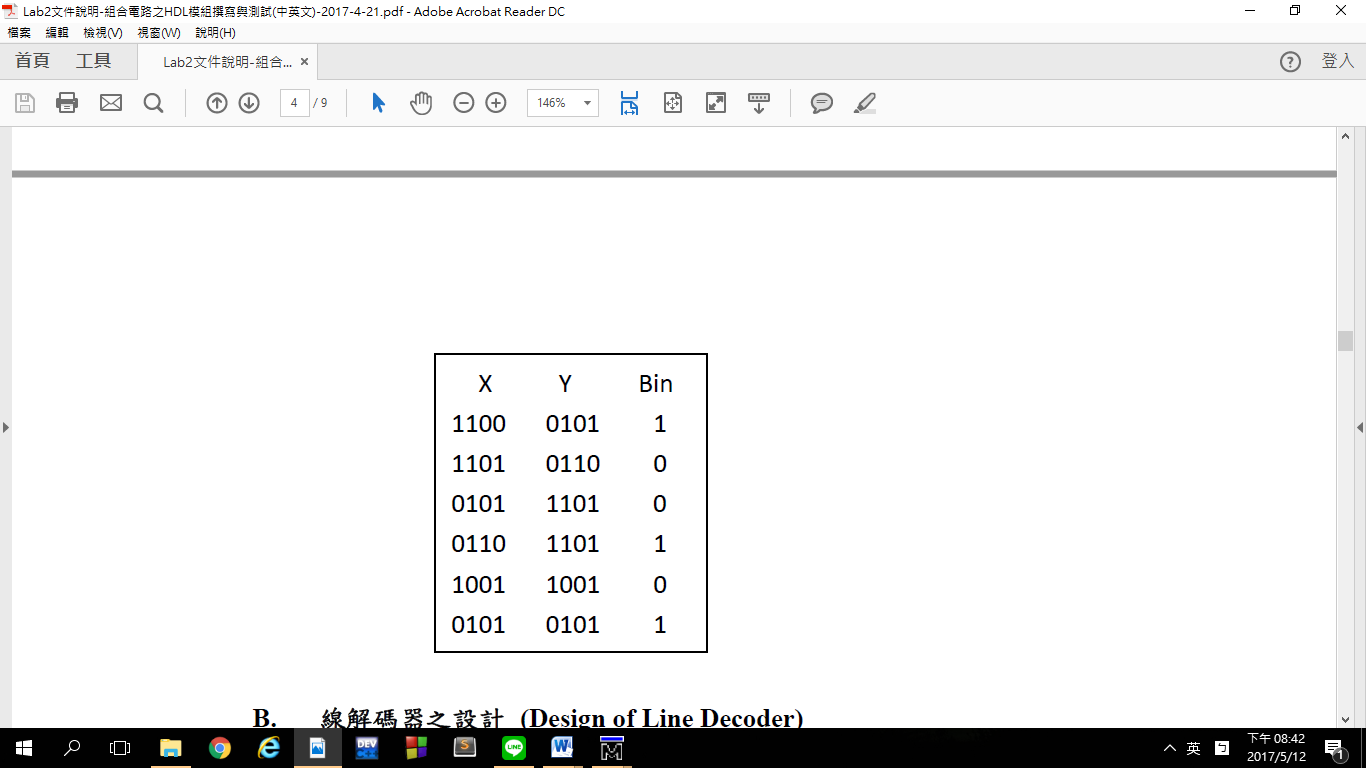


此圖對照上圖中的真值表，其值相同，所以正確。

1. 用全減器來建構四位元漣波借位減法器

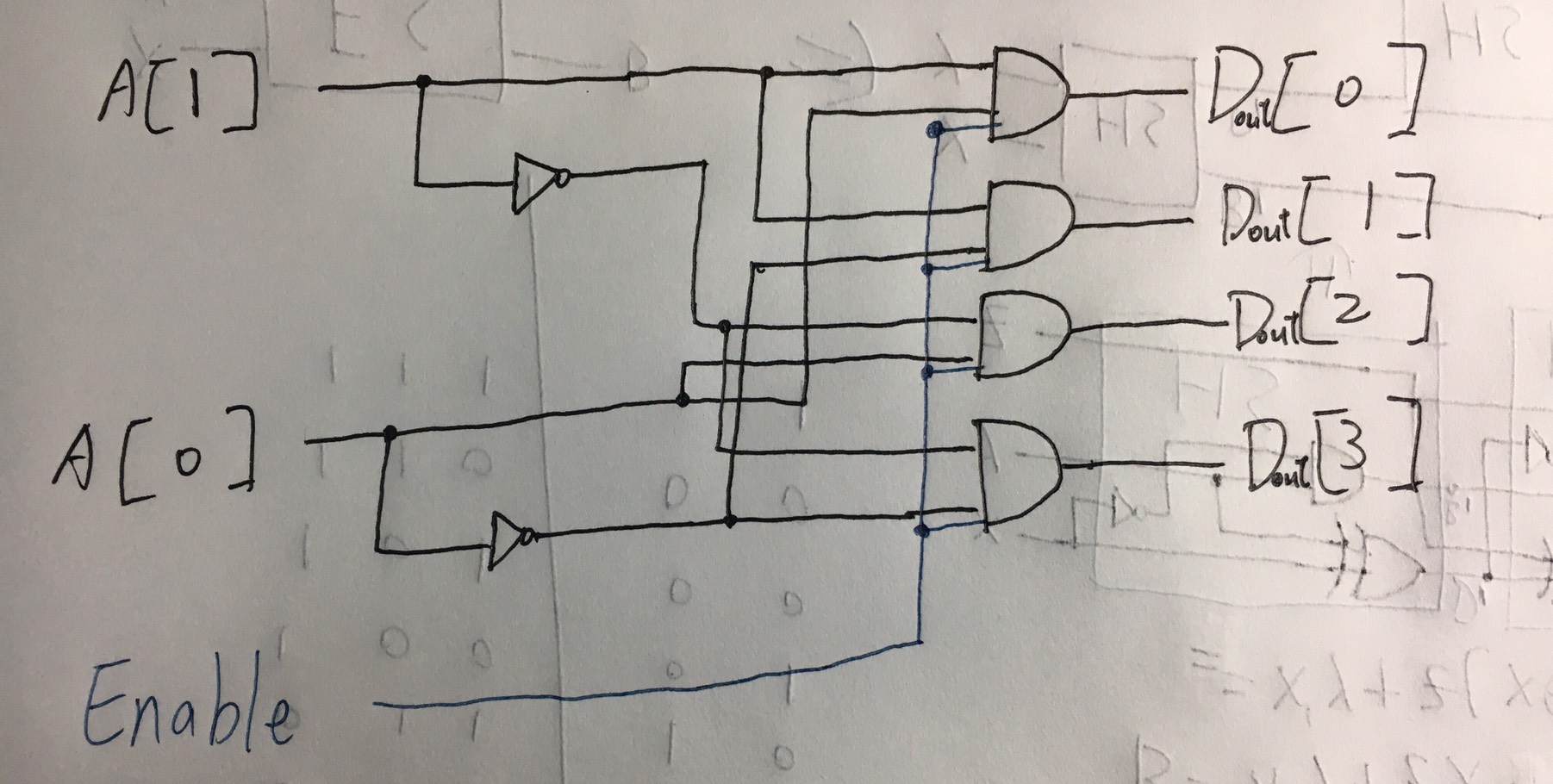


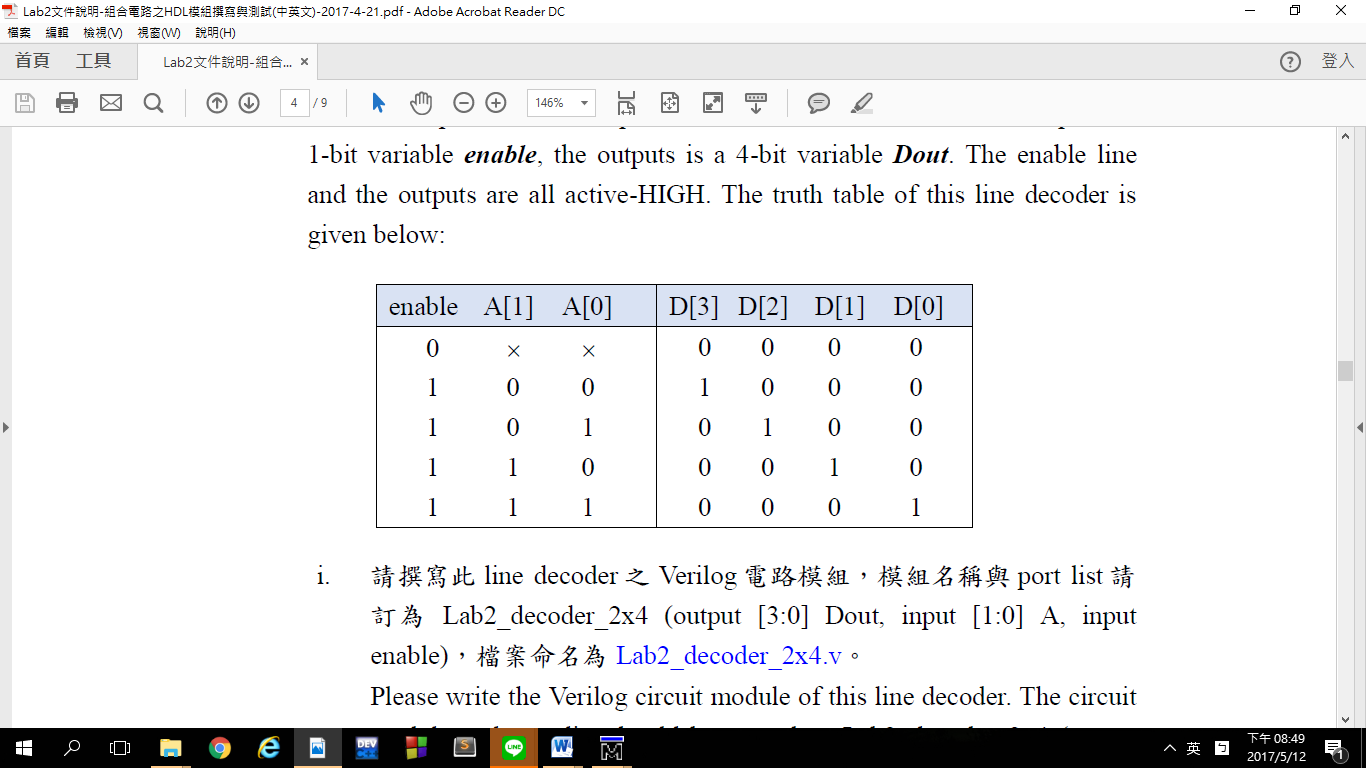




用上圖真值表做測試，結果符合，所以正確。

1. 2-to-4線解碼器之電路設計:

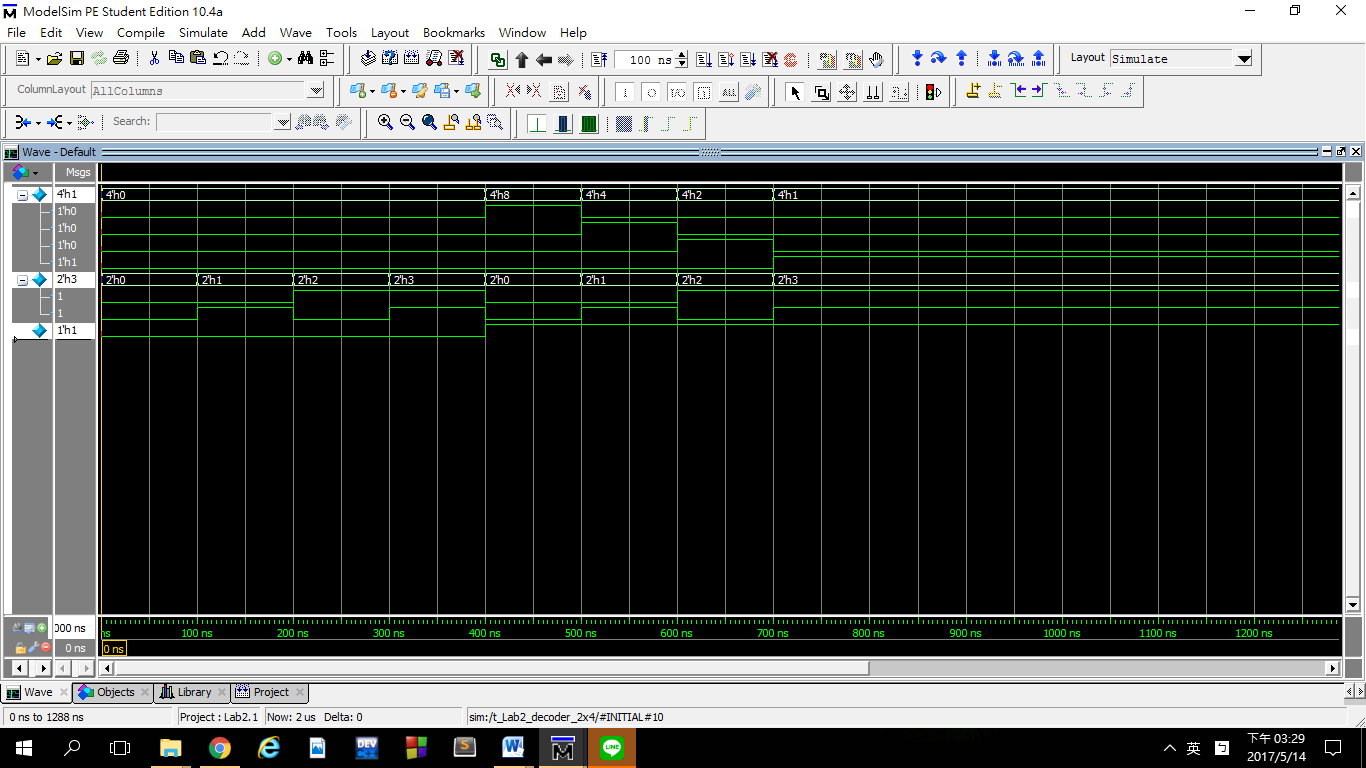




用此真值表，來推

當enable=0時，D[0]到D[3]都不工作

當enable=1時，找D[0]到D[3]的1值And在一起，也就是Minterm



此圖測試

當enable=0時A=00,01,10,11時的Dout 每100ns變化一次，可是看到上圖Dout 完全沒變，

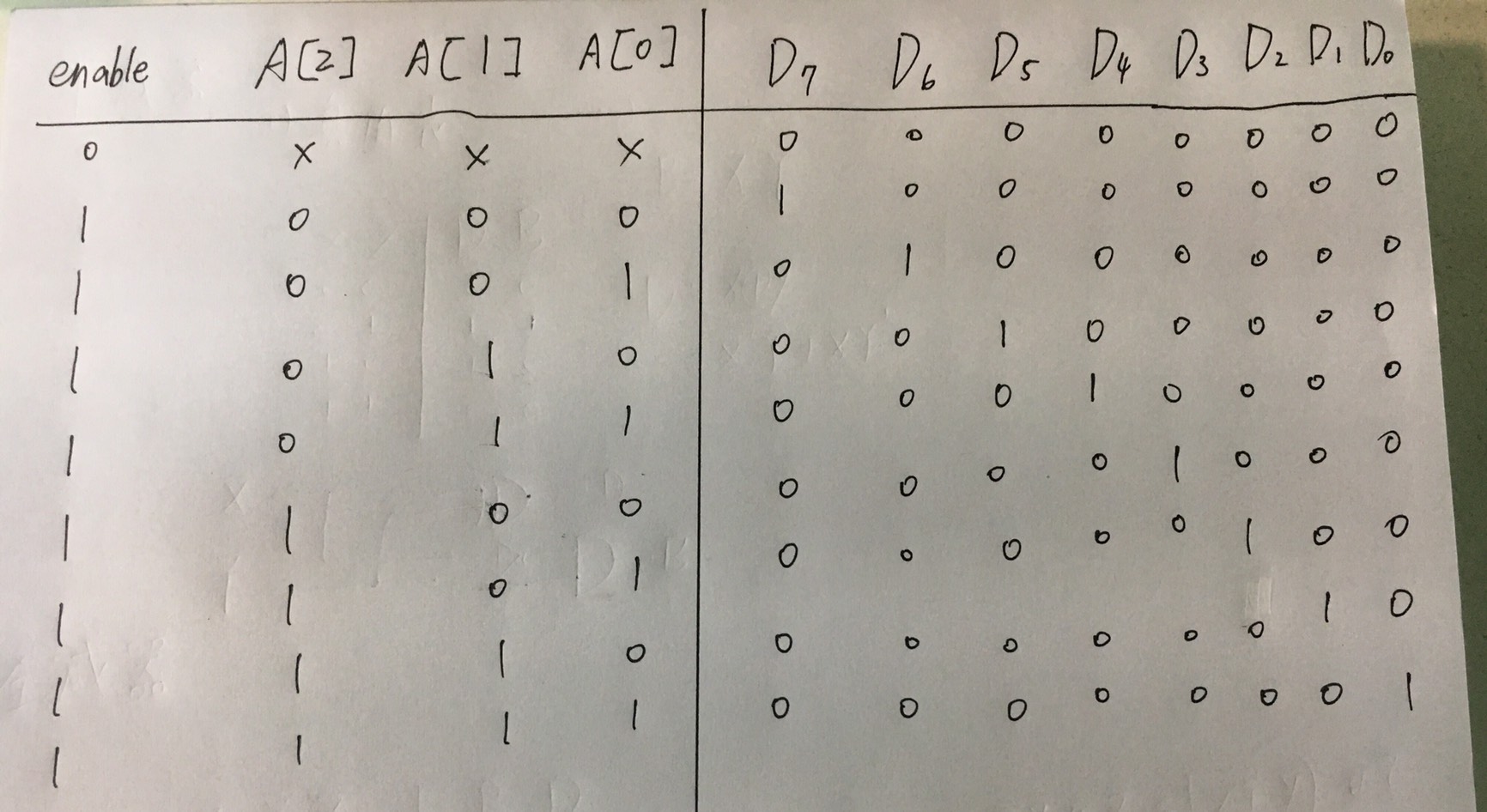
當enable=1時A=00,01,10,11時的Dout 每100ns變化一次，可看到上圖Dout的變化和真質表一樣，所以正確。

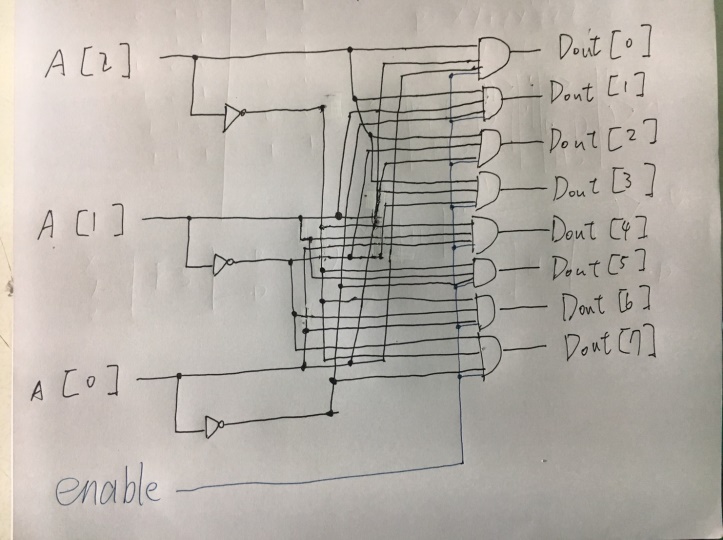
1. 3-to-8 解碼器之電路設計，首先寫出真值表如下圖，然後推出

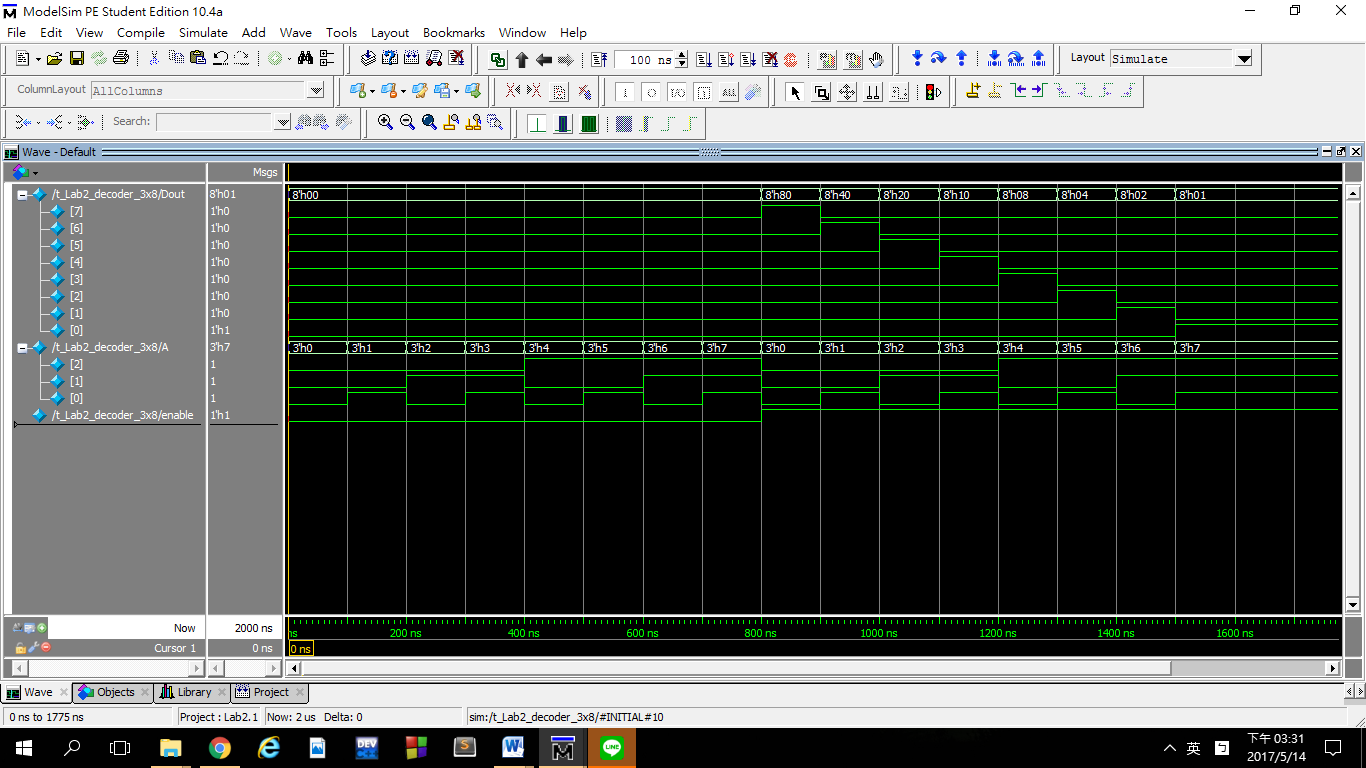
當enable=0時，D[0]到D[7]都不工作

當enable=1時，找D[0]到D[7]的1值And在一起，也就是Minterm，

並用此真值表畫出下圖電路。





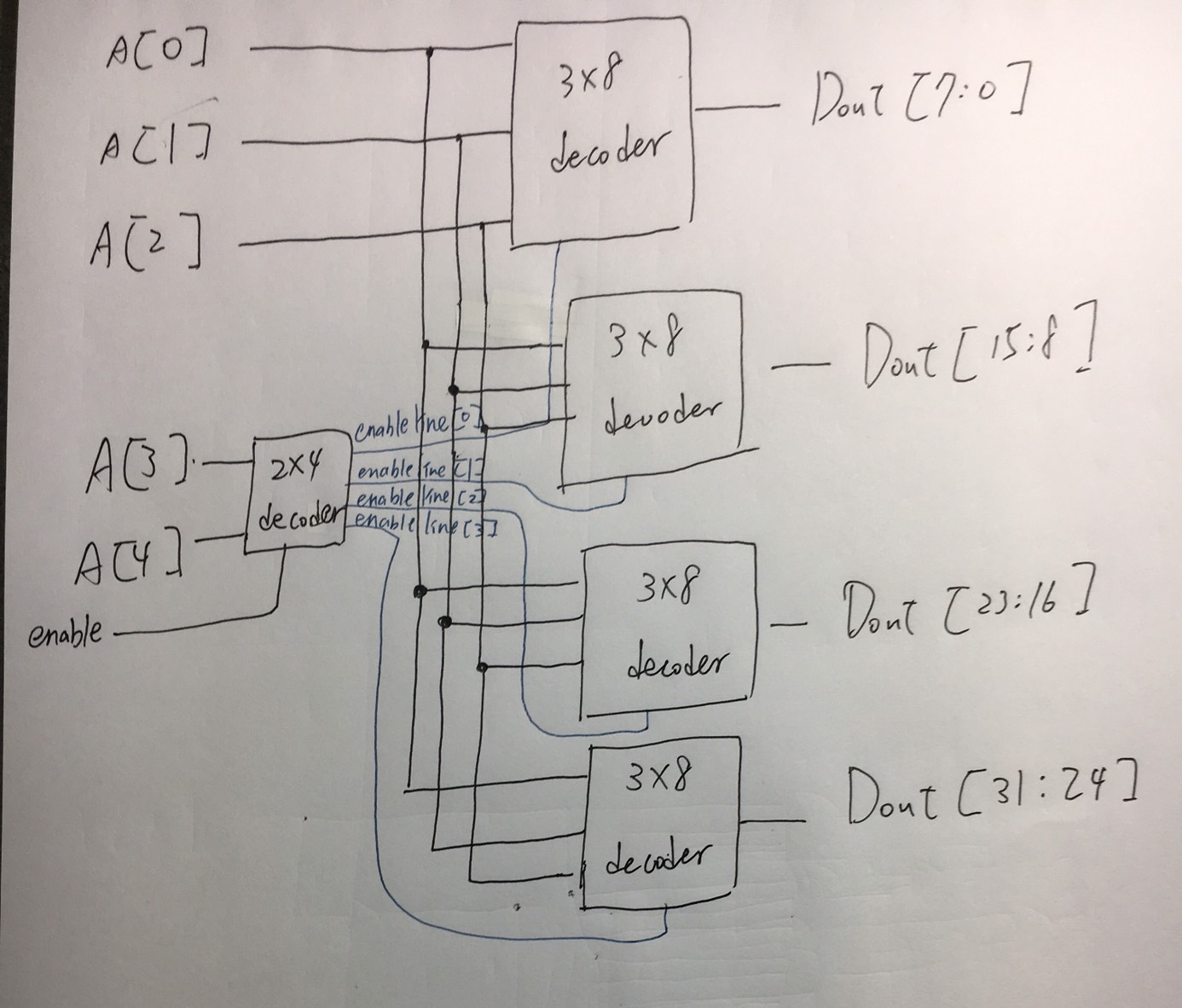


此圖測試

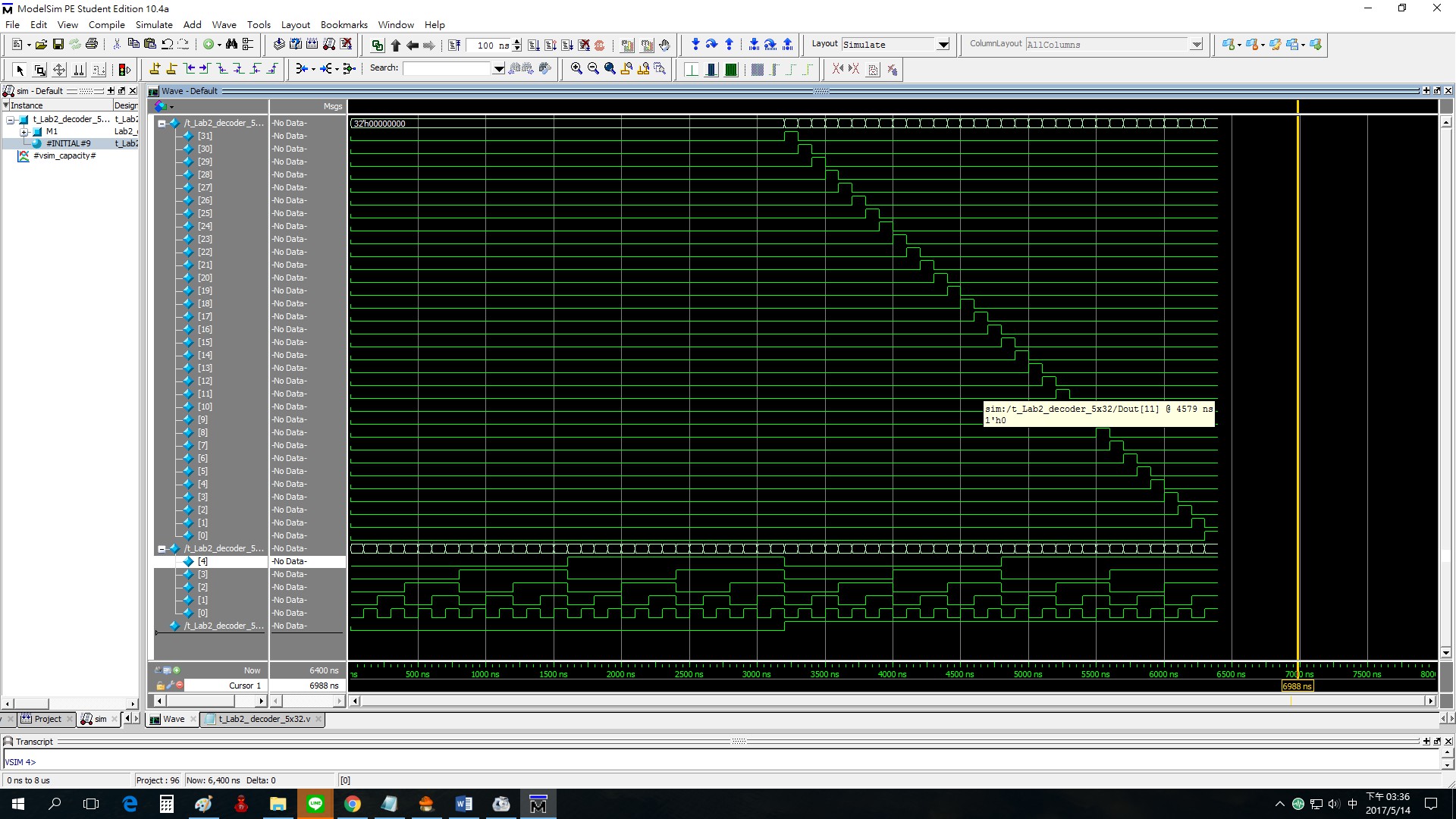
當enable=0時A=000,001,010,011,100,101,110,111 時的Dout 每100ns變化一次，可是看到上圖Dout 完全沒變，

當enable=1時A=000,001,010,011,100,101,110,111 時的Dout 每100ns變化一次，可看到上圖Dout的變化和真值表一樣，所以正確。

1. 5-to-32線解碼器之電路設計



首先設計讓5x8 decoder的三個輸入分別接到4個3x8 decoder的輸入，然後再用剩下兩個5x8 decoder的輸入和enable，來當2x4 decoder的輸入，所以2x4 decoder會有四個輸出，分別接到4個3x8 decoder，當enable訊號，所以enable就可以控制整個5x8 decoder，也就完成了5x8 decoder。



此圖測試

當enable=0時A從00000每次增加1bit的1到11111的變化，並且每100ns變化一次，可是看到上圖Dout 完全沒變，

當enable=1時A從00000每次增加1bit的1到11111的變化，

並且每100ns變化一次，可看到上圖Dout的變化，

A[4]A[3]兩位數00電路跑向D[7:0]，01電路跑向D[15:8]，10電路跑向D[23:16]，11電路跑向D[31:24]，而A[2:0]則代表0~7就是在分別在D[7:0]、D[15:8]、D[23:16]、D[31:24]中的第0~7個，ex當A=01001就是D[15:8] 中的第1個，也就是D[9]。

(7)

這次的Lab讓我學習到如何使用Behavior 的寫法，讓我多學會一種描述電路的方式，也讓我學會用已經寫好的電路來去寫比較大的電路，也讓我體會到自己手繪設計電路果然不是這麼容易，但是只要跟著步驟，寫真值表、寫出布林代數式，就可以畫出來，並且寫成硬體描述語言，真的很酷。這次遇到的困難是在最後那個5\*32 decoder的設計，在這個上面花了特別多的時間去想要如何接才能組出這個電路，不過就整體而言已經比上次寫Lab1時更加熟練，語法上有更熟悉一些，希望下次的Lab可以再接再厲，對數位電路設計有更深入的了解。