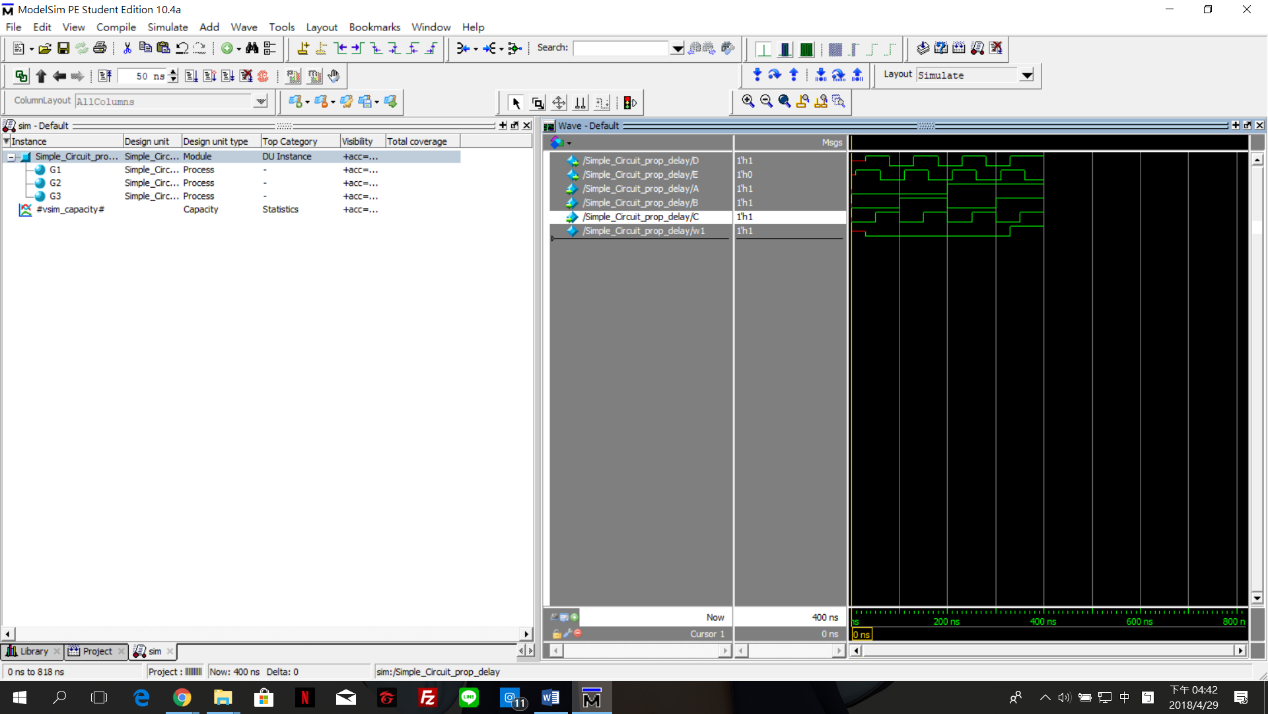
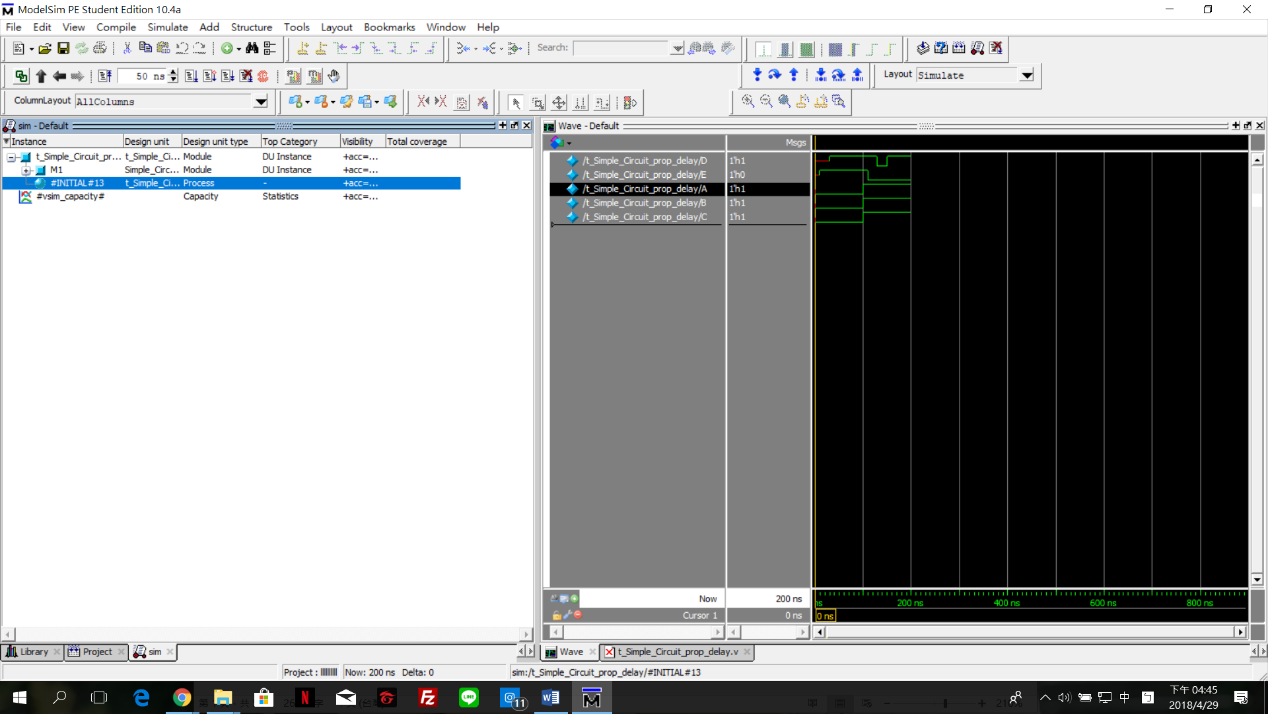
數電作業Lab1

(1) 2A(a)

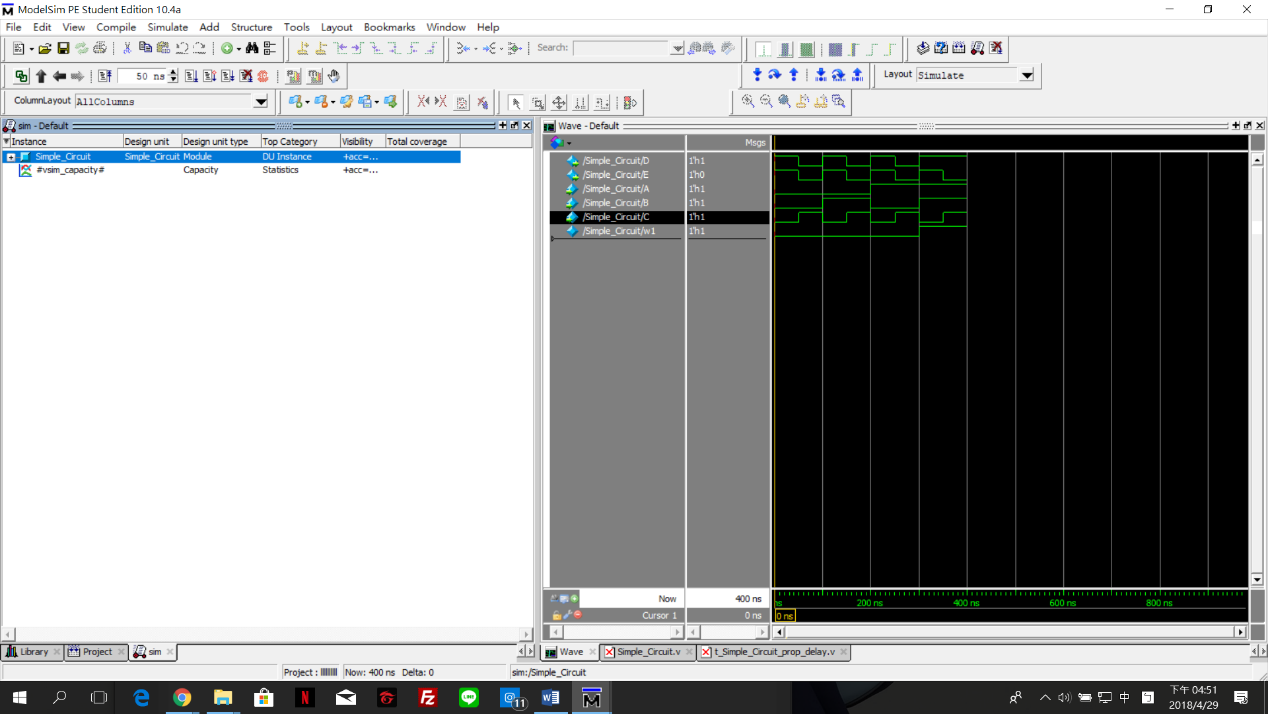
Simple\_Circuit\_prop\_delay.V ABC=000~111



t\_Simple\_Circuit\_prop\_delay.V



Simple\_Circuit.V ABC=000~111



Simple\_Circuit\_prop\_delay.V 為有延遲的完整圖

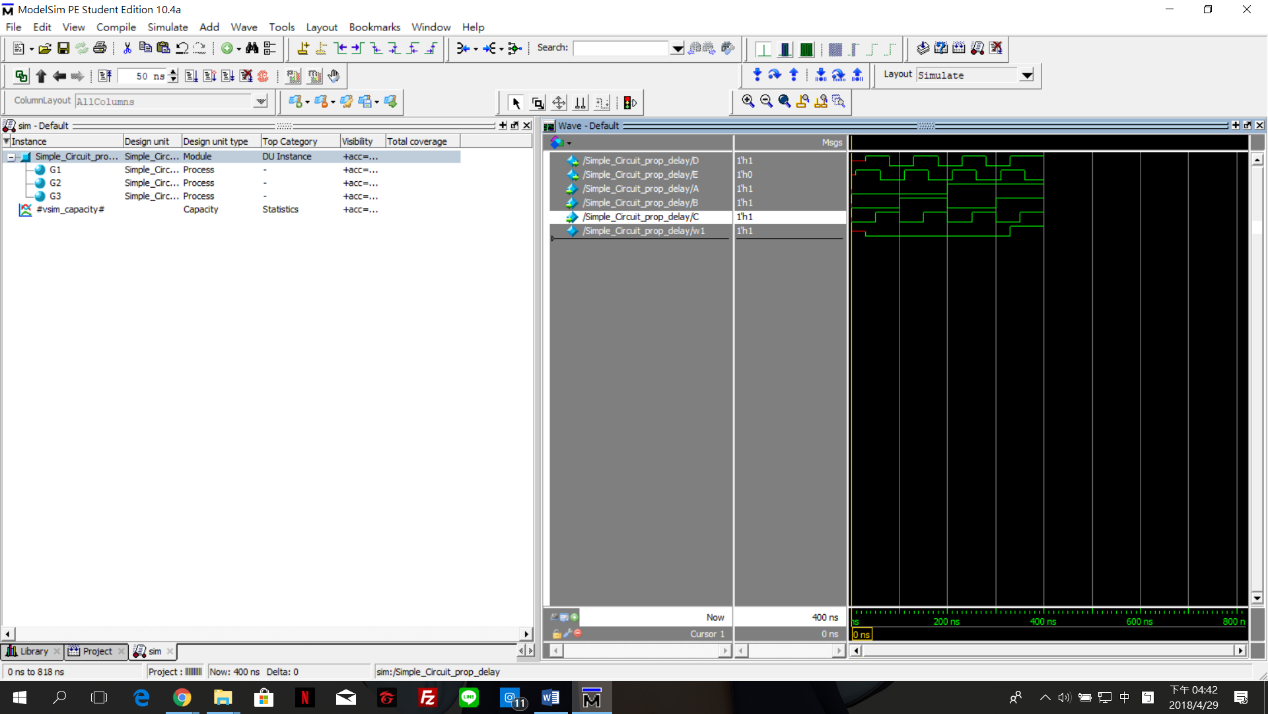
t\_Simple\_Circuit\_prop\_delay.V 為只有ABC=000,111有延遲的圖

Simple\_Circuit.V 為沒有延遲的完整圖

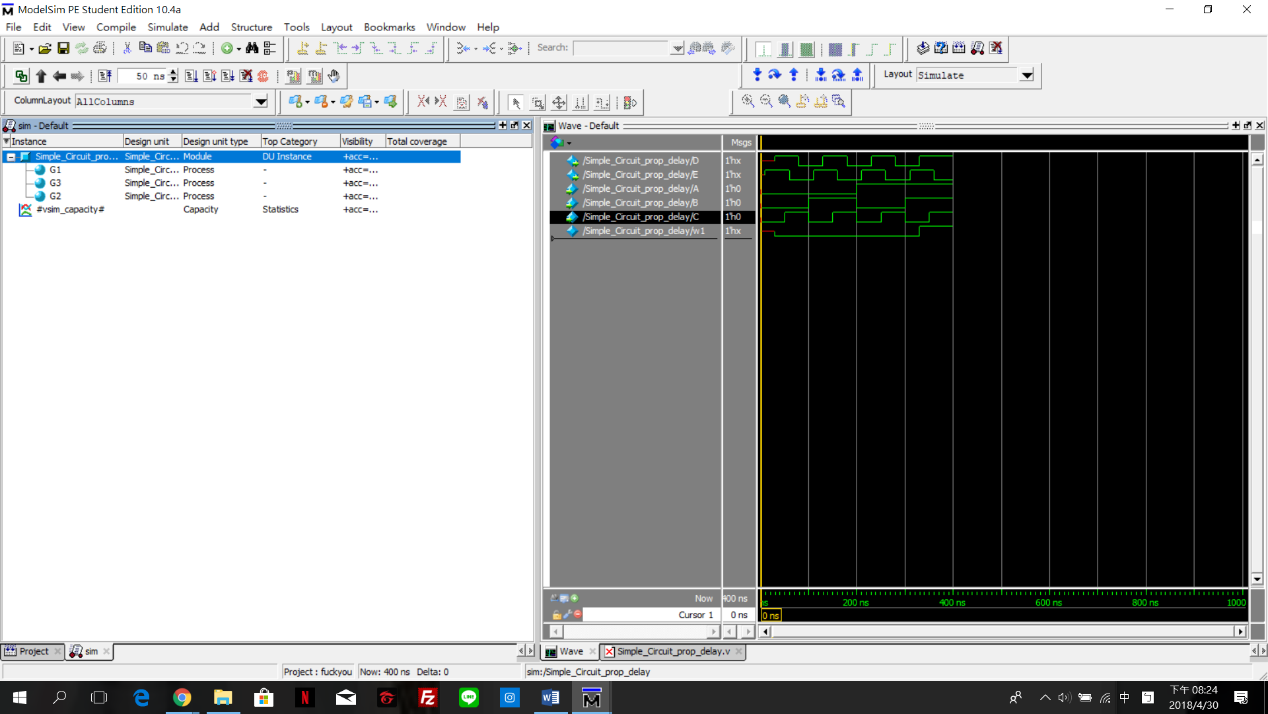
差別在於沒有prop的值會直接出現，沒有延遲，但有prop的要等一段單位為ns的時間後才會有值。

(2) 2A(b)

原Simple\_Circuit\_prop\_delay.V ABC=000~111



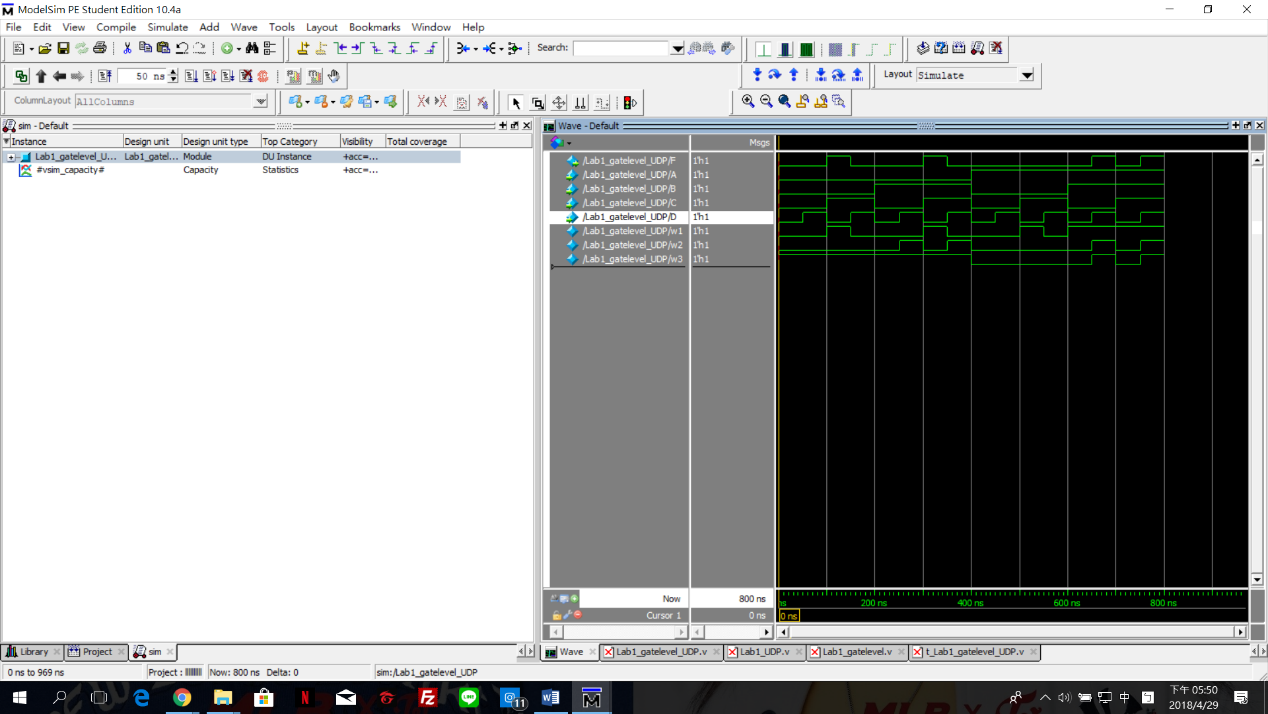
後Simple\_Circuit\_prop\_delay.V ABC=000~111



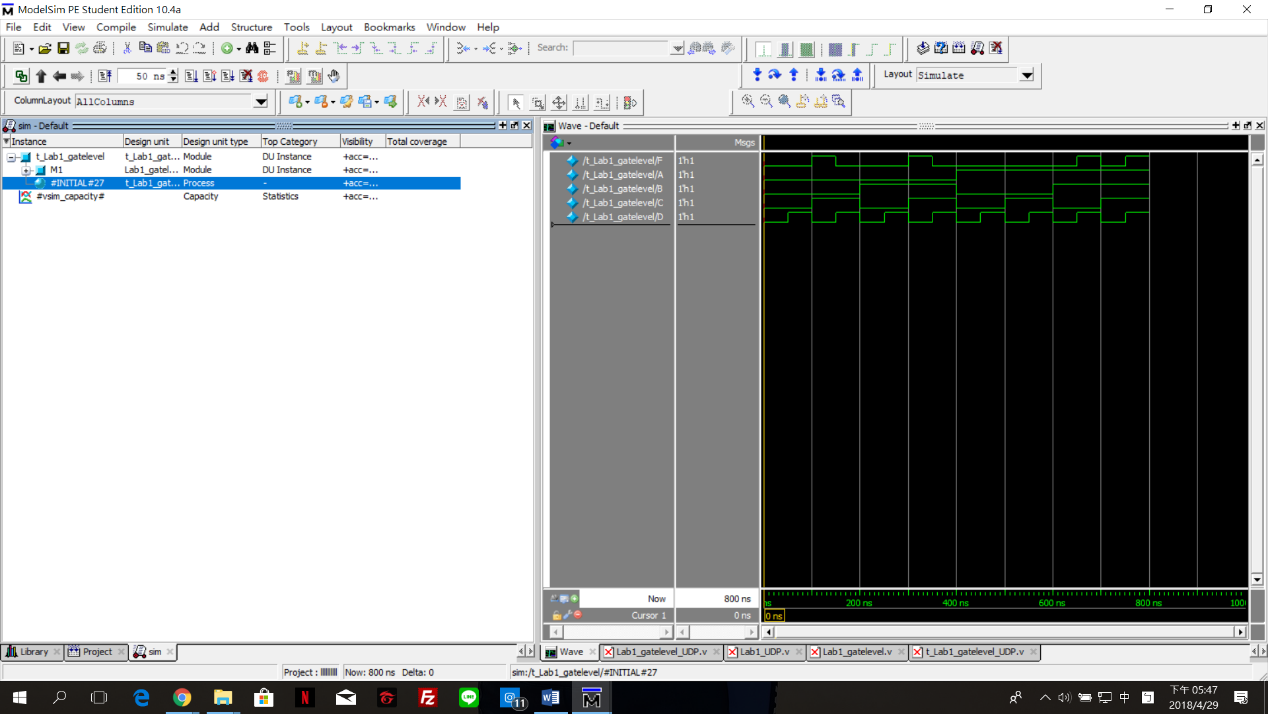
原本與後來的完全一樣，因為只有輸入的順序不一樣，不會影響輸出。

(3) 2B(a)

Lab1\_gatelevel.V ABCD=0000~1111



t\_Lab1\_gatelevel.V

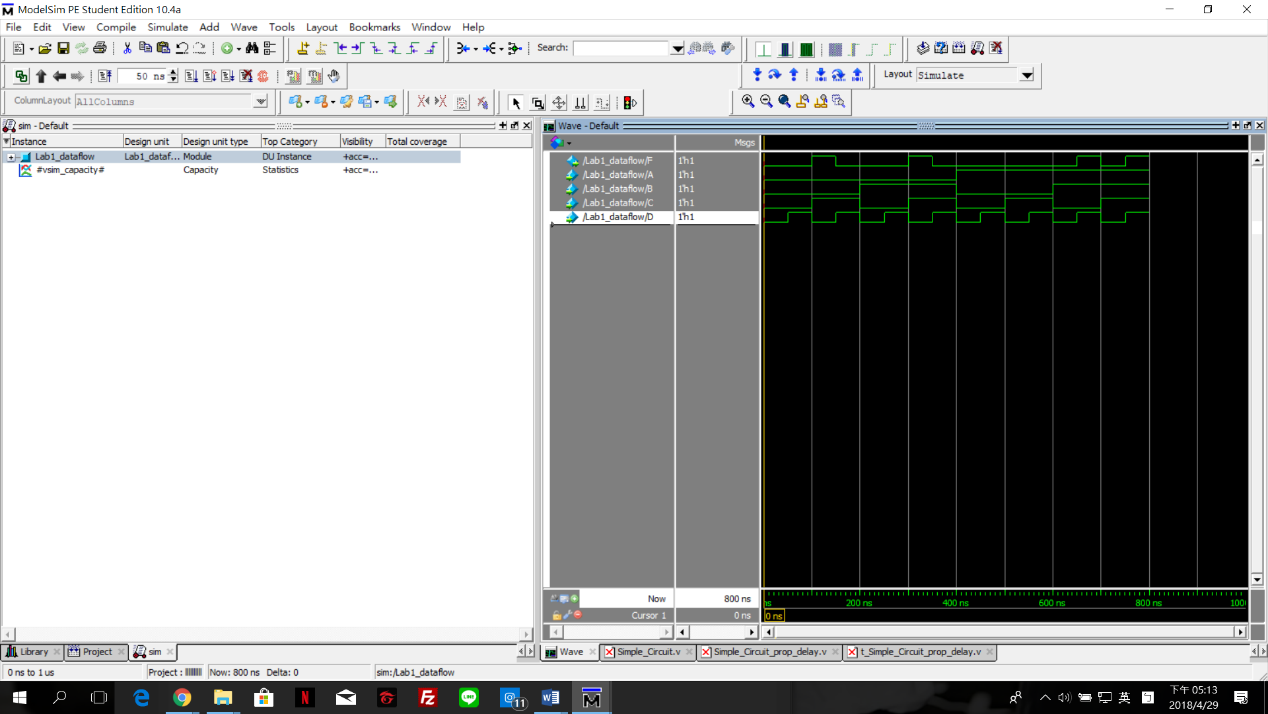


Lab1\_gatelevel.V與t\_Lab1\_gatelevel.V之ABCDF的值一致->模擬成功。

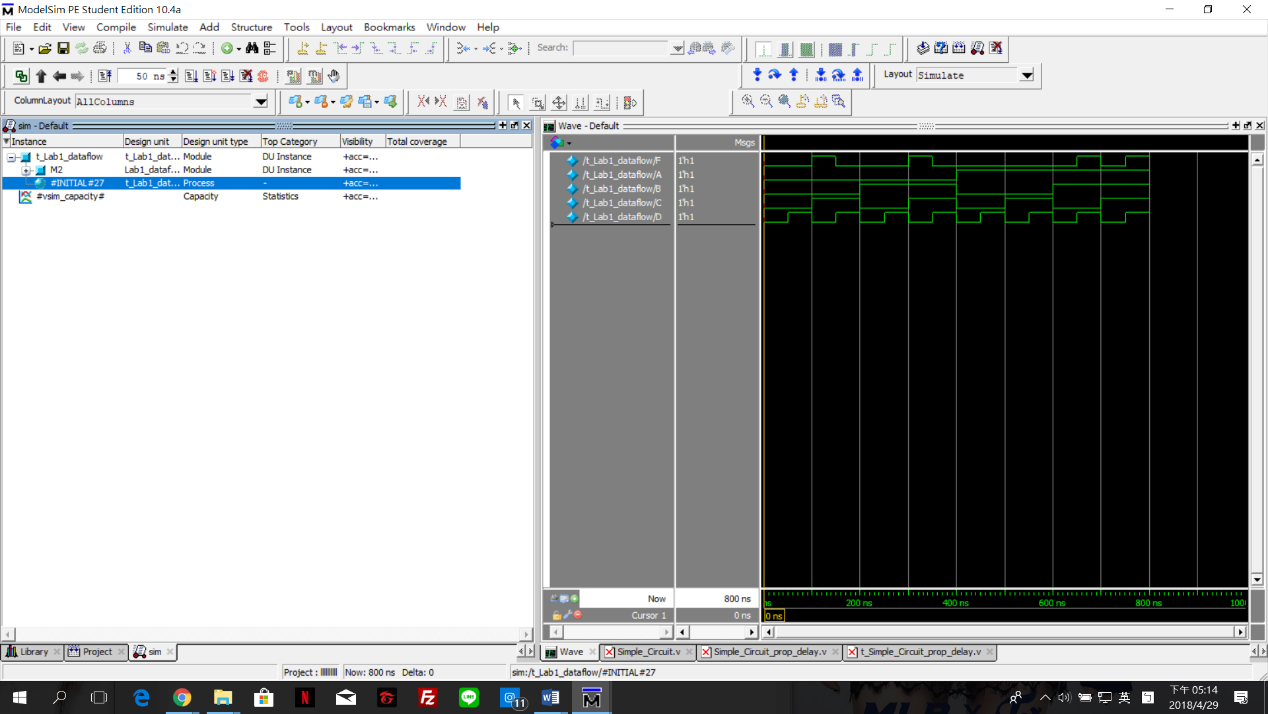
與它該有的truth table一樣->成功

(4) 2B(b)

Lab1\_ dataflow.V ABCD=0000~1111



t\_Lab1\_dataflow.V

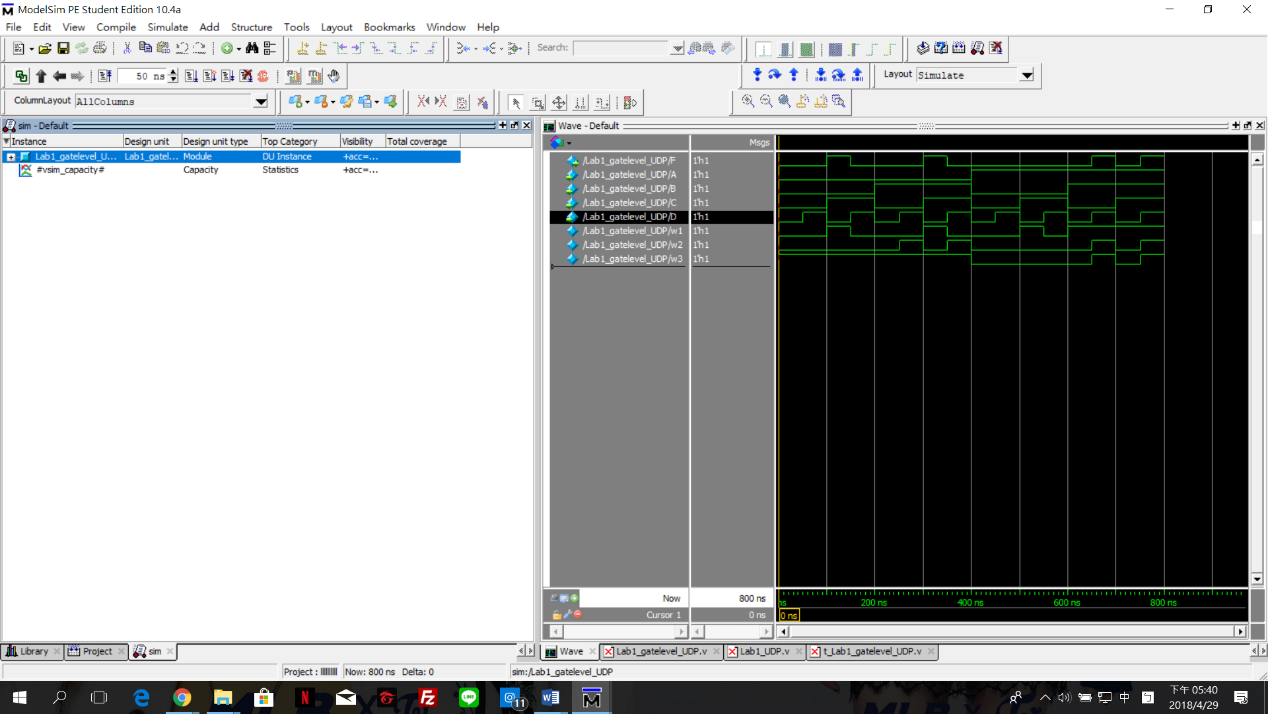


Lab1\_ dataflow.V與t\_Lab1\_ dataflow.V之ABCDF的值一致->模擬成功。

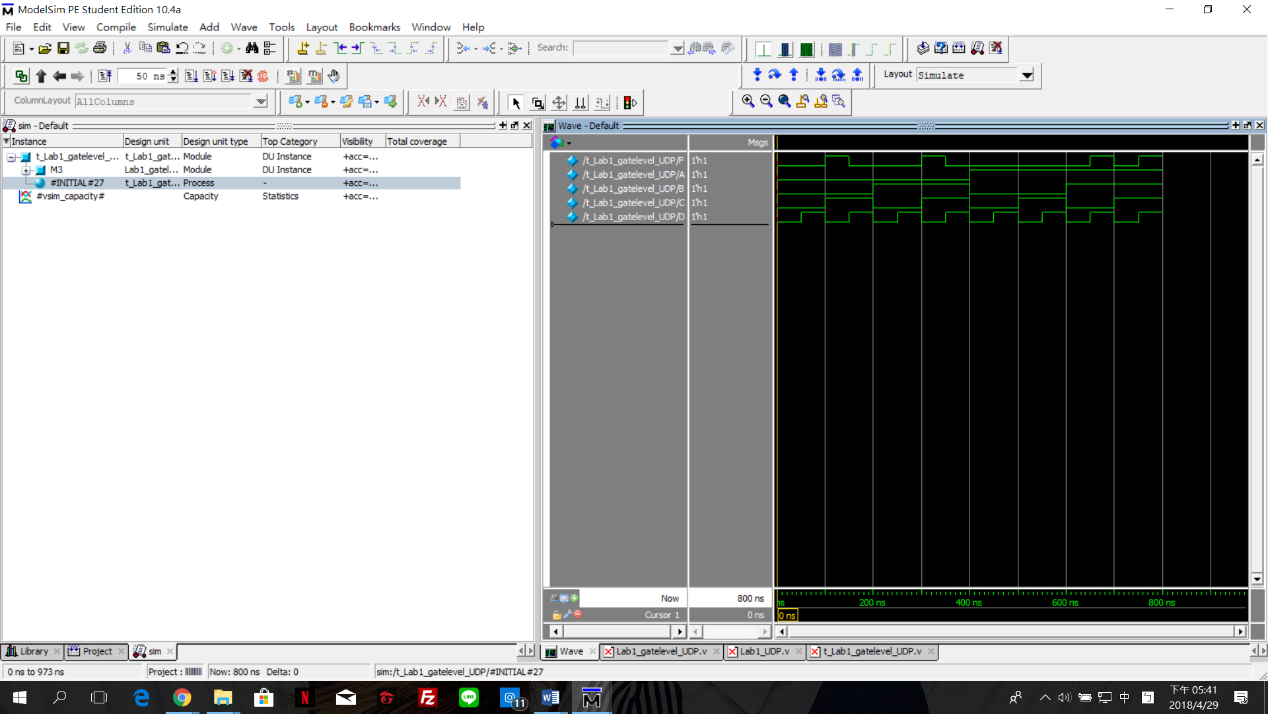
與它該有的truth table一樣->成功

(5) 2B(c)

Lab1\_ gatelevel\_UDP.V ABCD=0000~1111



t\_Lab1\_gatelevel\_UDP.V



Lab1\_ gatelevel\_UDP.V與t\_Lab1\_ gatelevel\_UDP.V之ABCDF的值一致->模擬成功。

與它該有的truth table一樣->成功

(3),(4),(5)該有的truth table:

A B C D | F A B C D | F

0 0 0 0 0 1 0 0 0 0

0 0 0 1 0 1 0 0 1 0

0 0 1 0 1 1 0 1 0 0

0 0 1 1 0 1 0 1 1 0

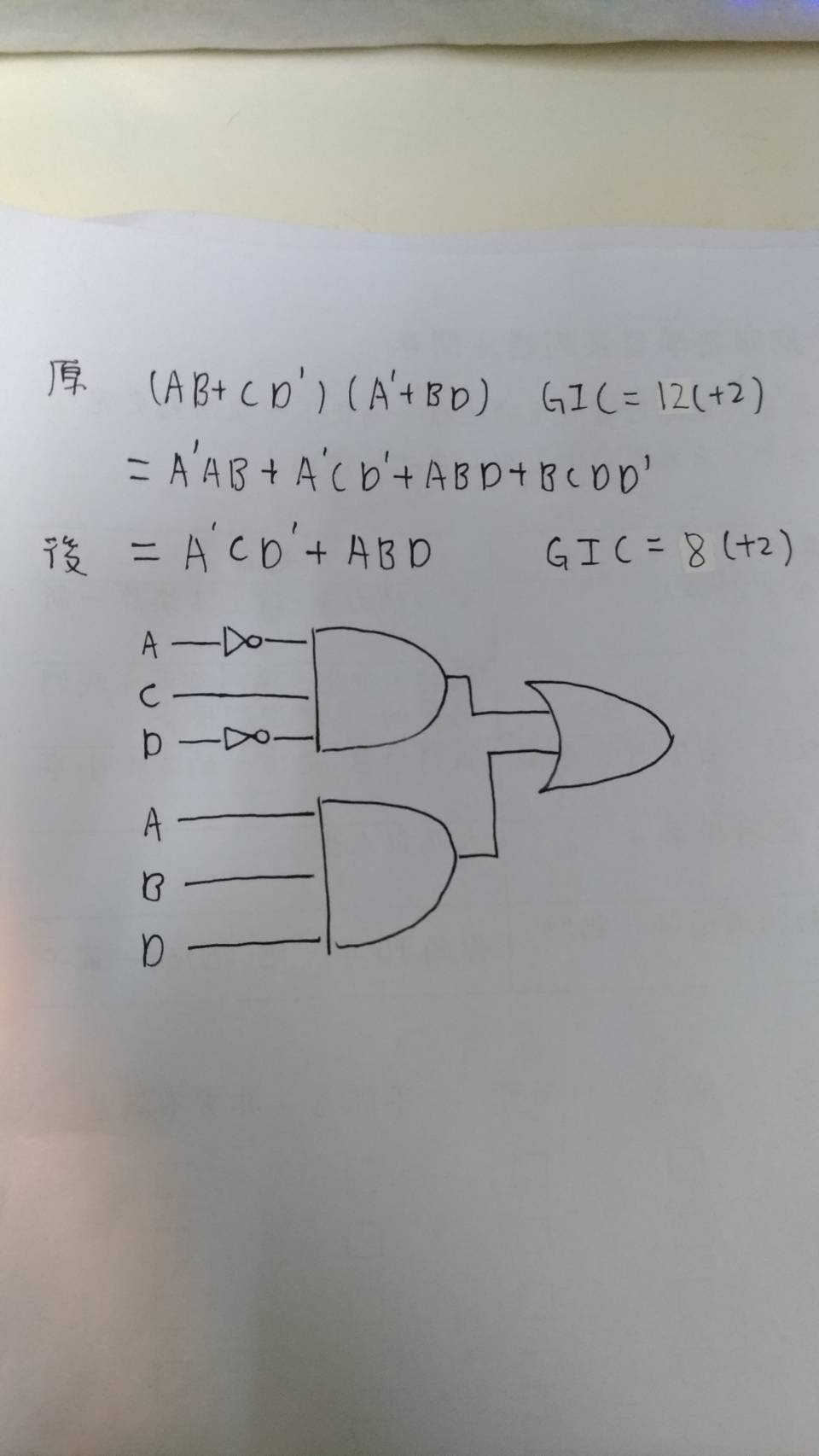
0 1 0 0 0 1 1 0 0 0

0 1 0 1 0 1 1 0 1 1

0 1 1 0 1 1 1 1 0 0

0 1 1 1 0 1 1 1 1 1

(6)圖1不是最少GIC的情況



(7)心得:這個程式語言很容易理解，直觀而且很簡單，寫模組的方式和在C中寫function的感覺很像，基本電路不需要都用NAND做成真的很方便。只不過也許是因為是很久之前的程式，介面操作起來很不方便，而且與WIN10有點不容，介面用起來比較難上手。