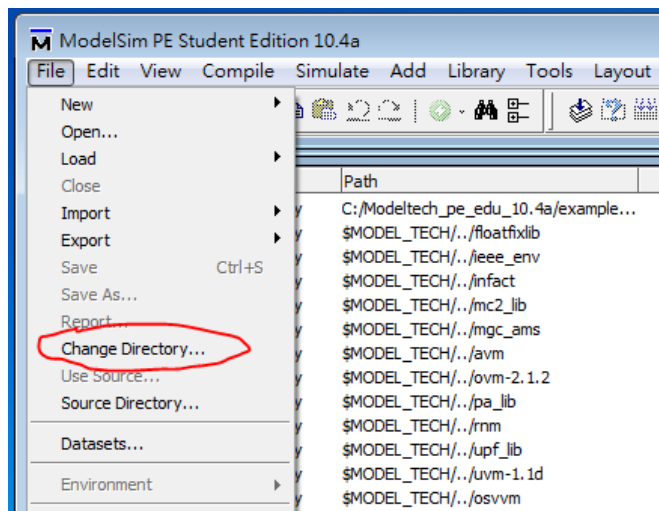


# Modelsim 建立專案教學

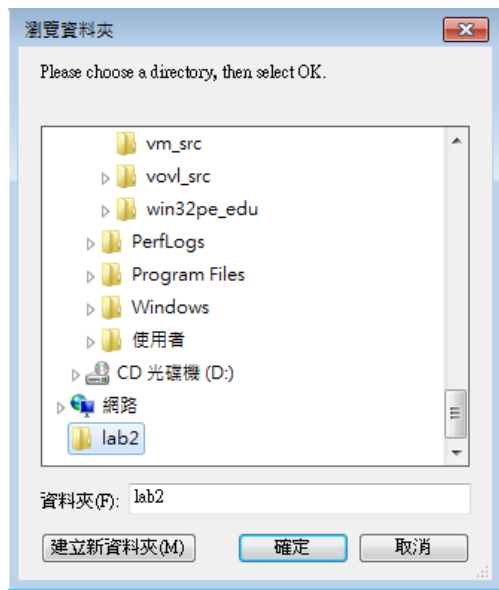
建一個資料夾，並將檔案放入此資料夾



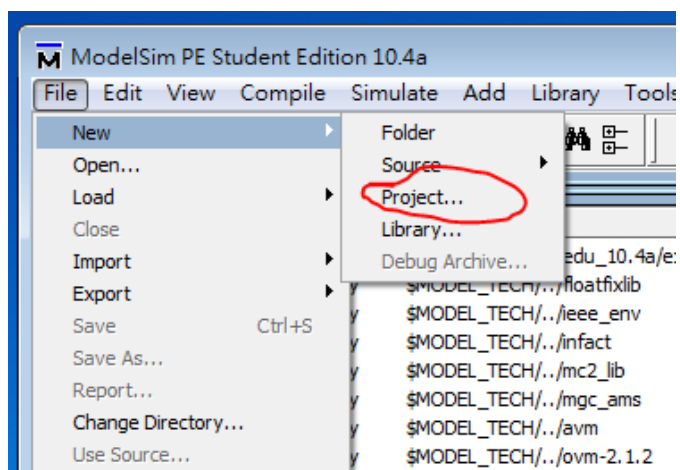
File -> change directory 點選自己要建專案的資料夾



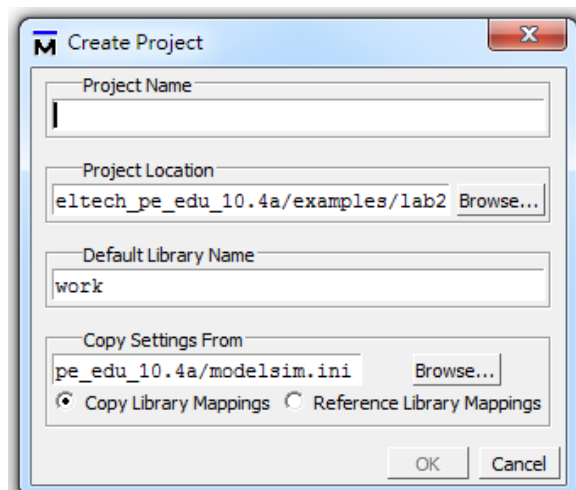
選擇剛剛建立的資料夾



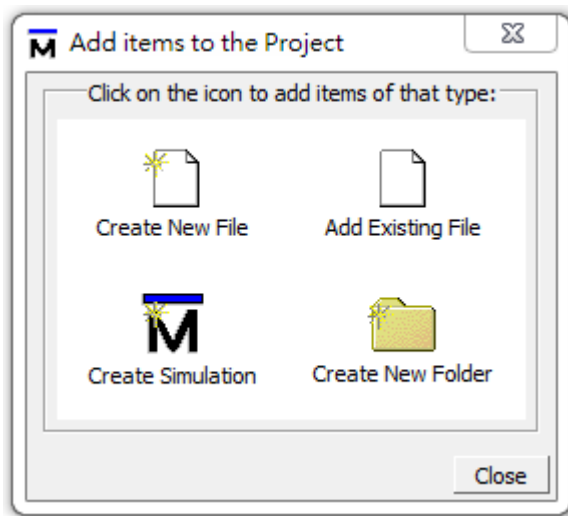
點下 File -> New -> Project



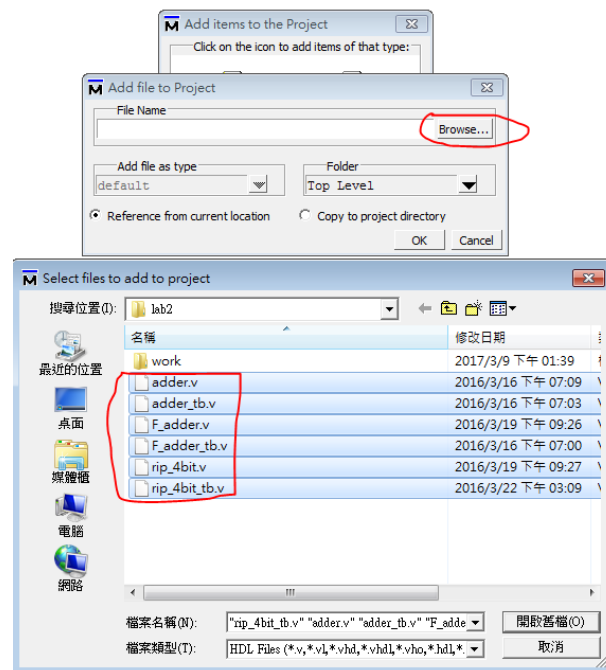
取個名字(e. g. adder, lab2...)



Add existing file



選取檔案：



半加器範例：

```
module adder(input1,input2,cout,sum);

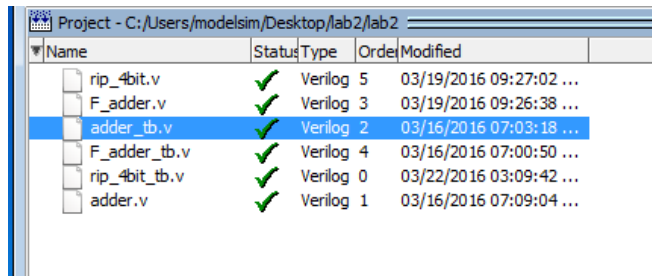
    input input1;
    input input2;

    output cout;
    output sum;

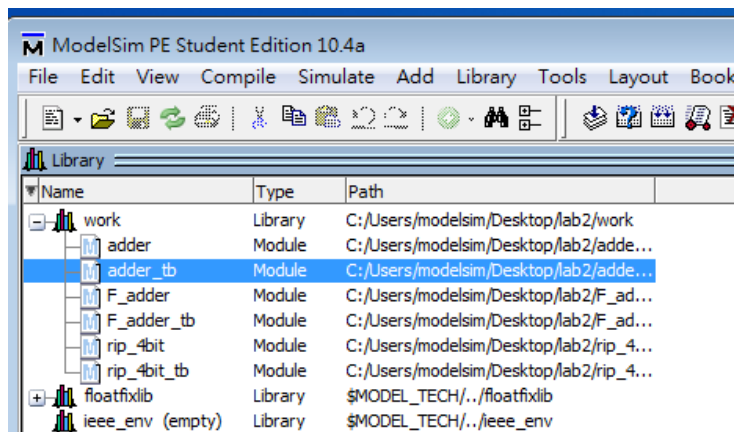
    assign sum = input1 ^ input2;
    assign cout = input1 & input2;

endmodule
```

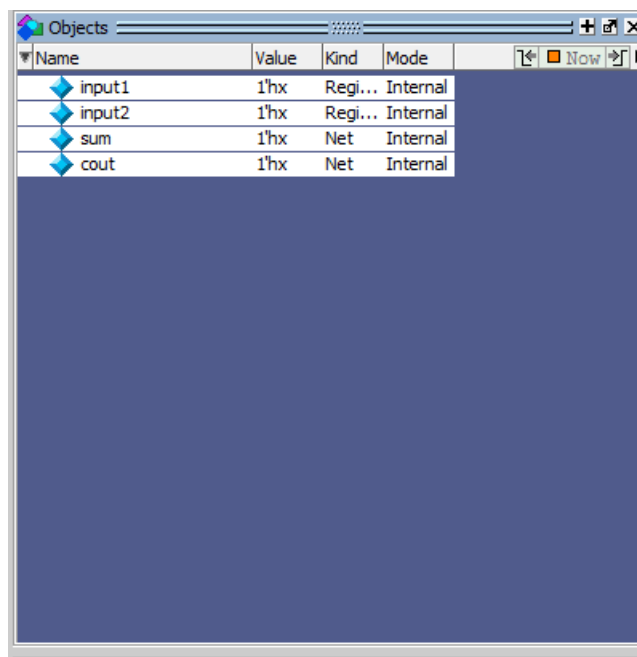
打完後對著檔案名稱點右鍵(compile -> compile selected file 每次修改檔案，都要儲存並做此步驟直到檔案打勾勾)，接著在空白處點右鍵新增助教提供的 tb 檔做驗證(add to project -> existing file 一樣要 compile 到打勾勾)



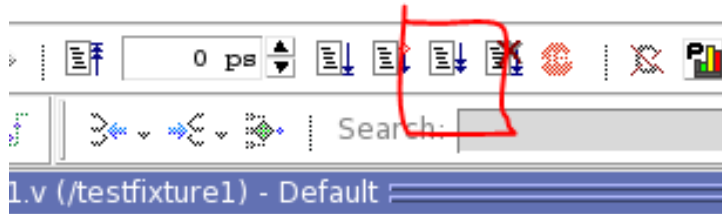
對著 adder\_tb 檔點左鍵兩下，跳出視窗按 yes



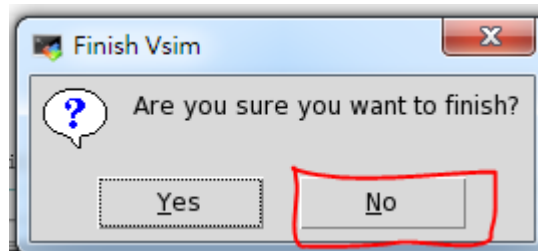
選取後按右鍵(add to -> wave -> selected signals)



接著點下 Run all 即可看波形



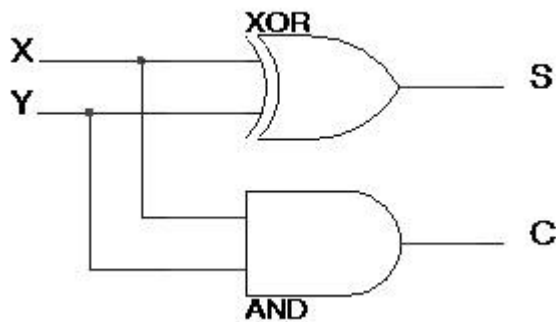
若出現 Are you sure you want to finish? 按 NO



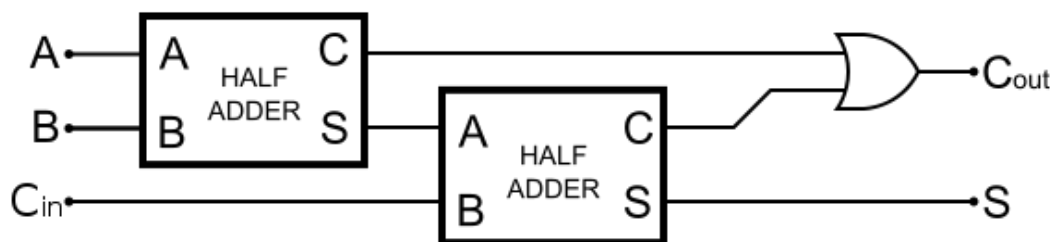
練習：

利用上面給的 half adder module，先修改 F\_adder.v 做出 full adder module，再修改 rip\_4bit.v 做出 4bit ripple carry adder  
(用附上的 rip\_4bit\_tb.v 做驗證，不需要修改 tb 檔)

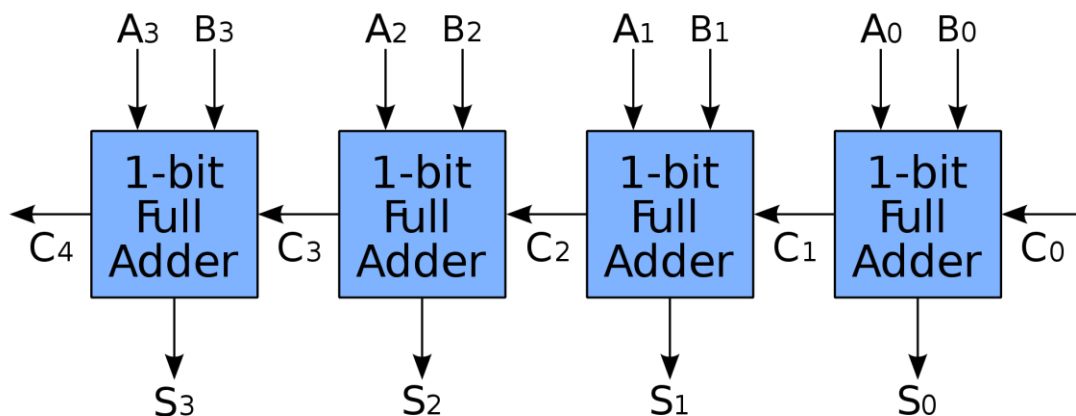
Reference：



full adder module：



4bit ripple carry adder：



運算子參考：

~ NOT

& AND

| OR

^ XOR

如何呼叫 module :

可以參考像是 tb 檔裡面的

若要呼叫 adder

則

```
adder add(.input1(input1),.input2(input2),.cout(cout),.sum(sum));
```

add 是自己取的 module 名稱(也可以叫 add1, add2...)

括弧外的 input1 是要呼叫的 module adder 裡線的名稱

括弧內的 input1 是 module 自己要接的線的名稱