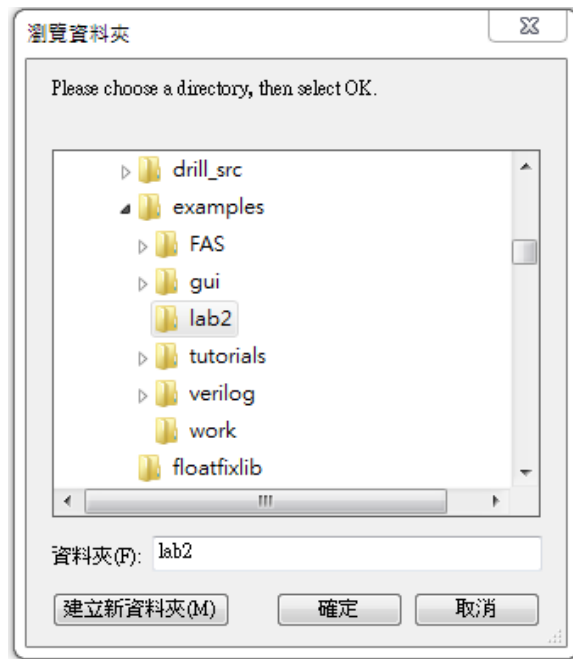
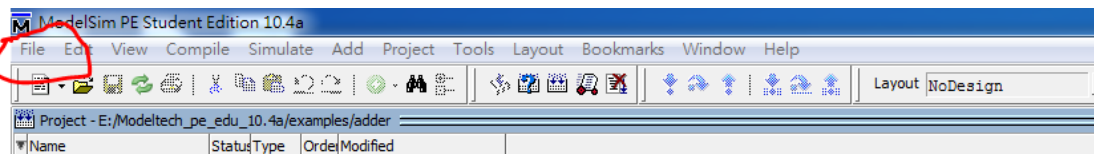


Modelsim 建立專案教學

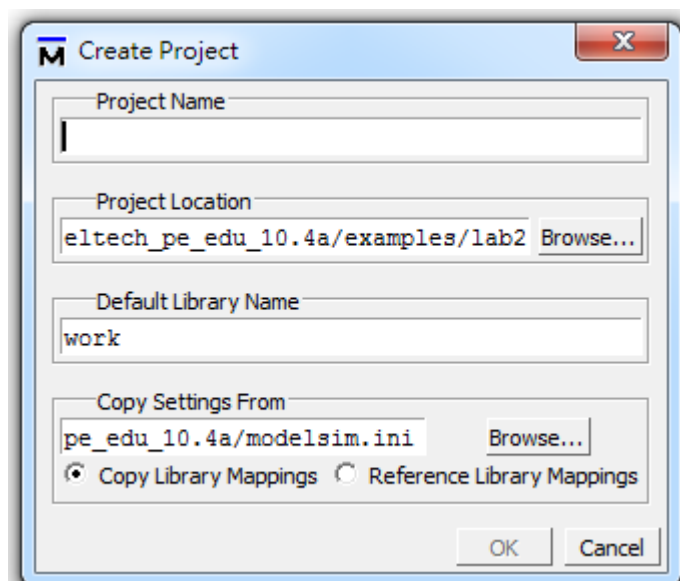
File -> change directory 點選自己要建專案的資料夾



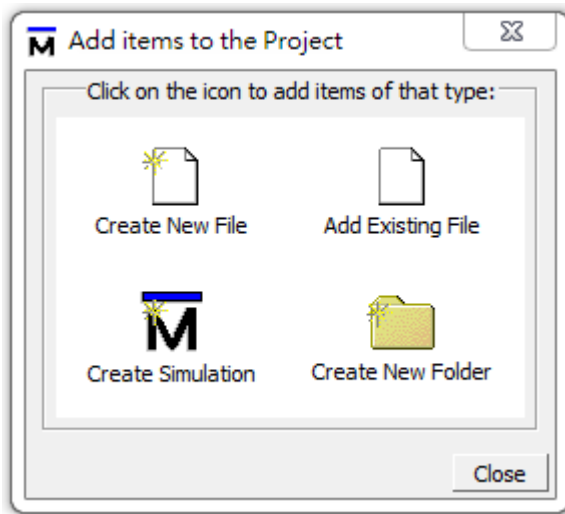
點下 File -> New -> Project



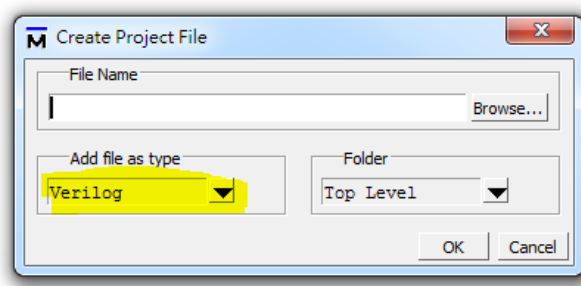
取個名字(e. g. adder)以及設定路徑



Create new file (若有已經打好的 .v 檔則是點 add existing file)



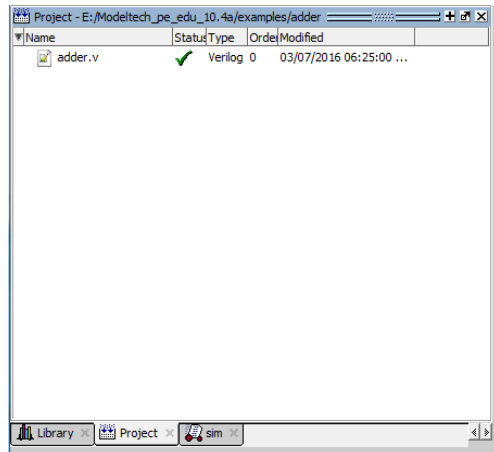
命名檔案名稱 (e. g. adder.v 不一定要與專案名稱同，但記得加上.v 副檔名)
以及選取 verilog



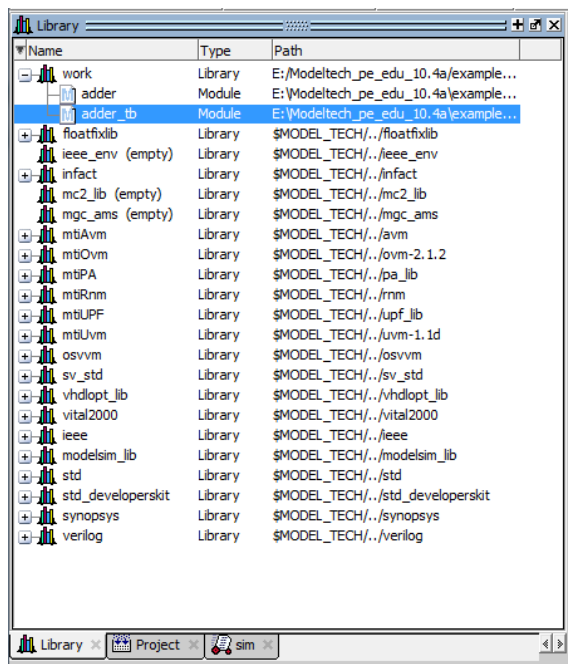
半加器範例：

```
module adder(input1,input2,cout,sum);  
  
    input input1;  
    input input2;  
  
    output cout;  
    output sum;  
  
    assign sum = input1 ^ input2;  
    assign cout = input1 & input2;  
  
endmodule
```

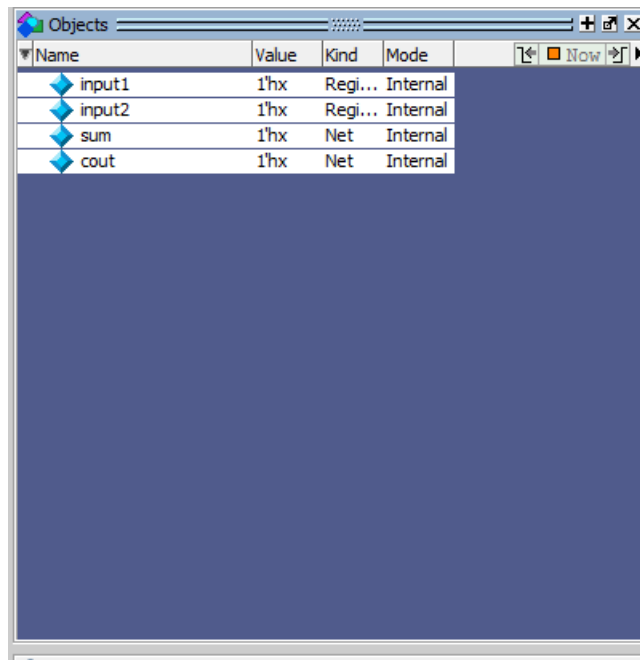
打完後對著檔案名稱點右鍵(compile -> compile selected file 每次修改檔案，都要儲存並做此步驟直到檔案打勾勾)，接著在空白處點右鍵新增助教提供的tb檔做驗證(add to project -> existing file 一樣要compile到打勾勾)



對著 tb 檔點左鍵兩下，跳出視窗按 yes

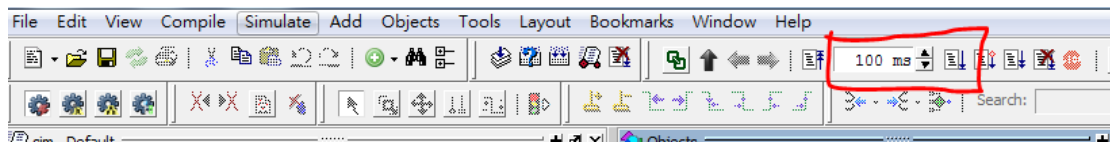


選取後按右鍵(add to -> wave -> selected signals)



| Name | Value | Kind | Mode |
|--------|-------|---------|----------|
| input1 | 1'hx | Regi... | Internal |
| input2 | 1'hx | Regi... | Internal |
| sum | 1'hx | Net | Internal |
| cout | 1'hx | Net | Internal |

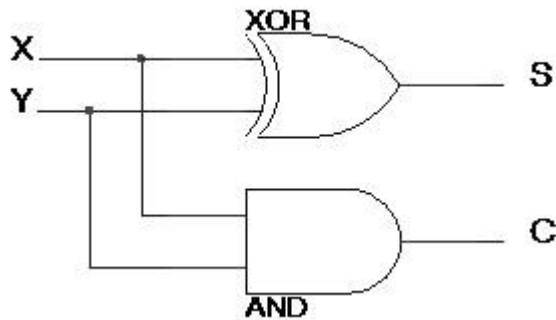
接著點下 Run 即可看波形（時間可以設定 100ms，一次跑完）



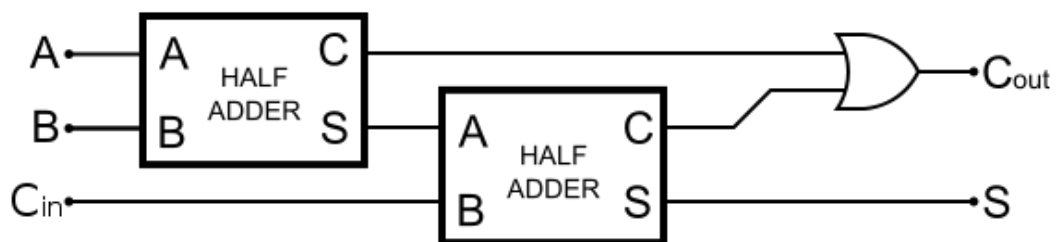
練習：

利用上面給的 half adder module，先做出 full adder module，再做出 4bit ripple carry adder（用附上的 rip_4bit_tb.v 做驗證）

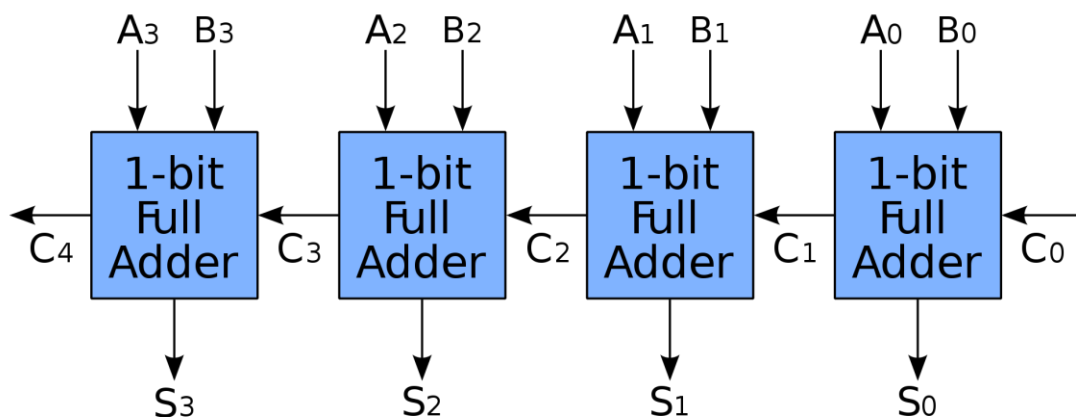
Reference：



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 是自己宣告的module(e.g. add1 add2)

括弧外的input1 為要呼叫的module **adder** 的線的名稱

括弧內的input1 為自己module **adder_tb** 線的名稱