Laboratory 4

認識Verilog 與

Xilinx & SMIMS 軟體介紹與操作

Department of Electrical Engineering
National Cheng Kung University

實驗目的

- 瞭解相關FPGA IDE環境操作
- 學習利用Verilog模擬邏輯電路與驗證

使用器材

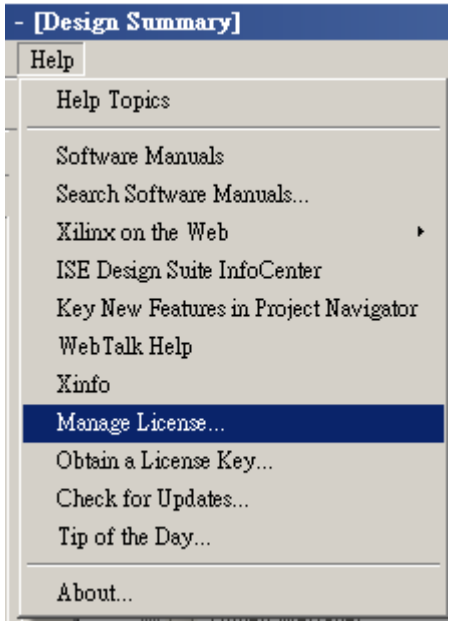
- 桌上型電腦
- Xilinx FPGA 板

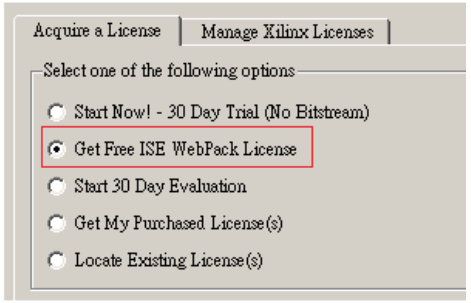
IDE安裝

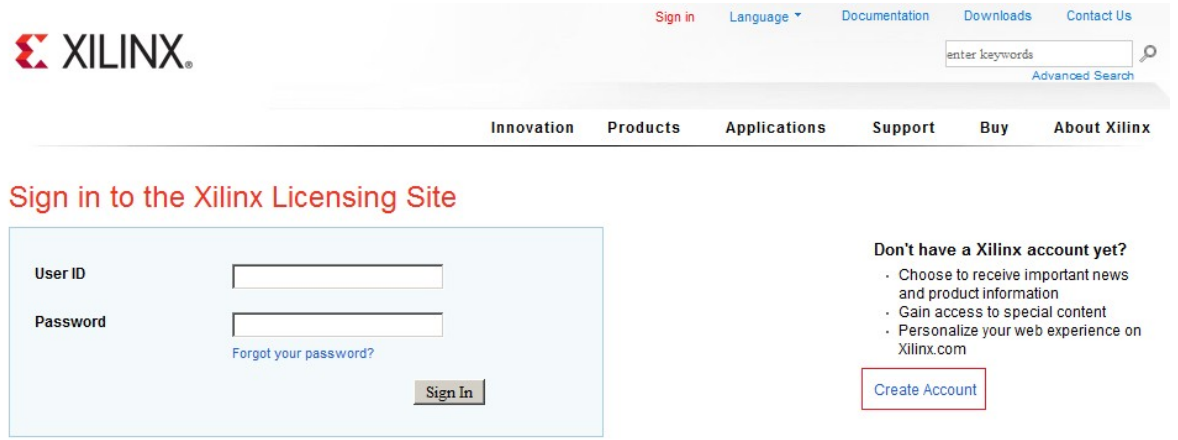
以下說明Xilinx IDE 相關安裝說明

Xilinx License (1/3)

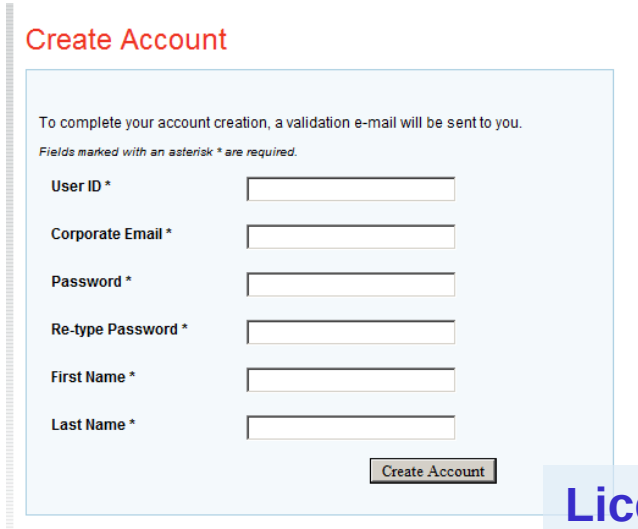
- 選擇help->manage license
- 選擇get free ISE WebPack License 3. 進入Xilinx網頁填寫資料註冊



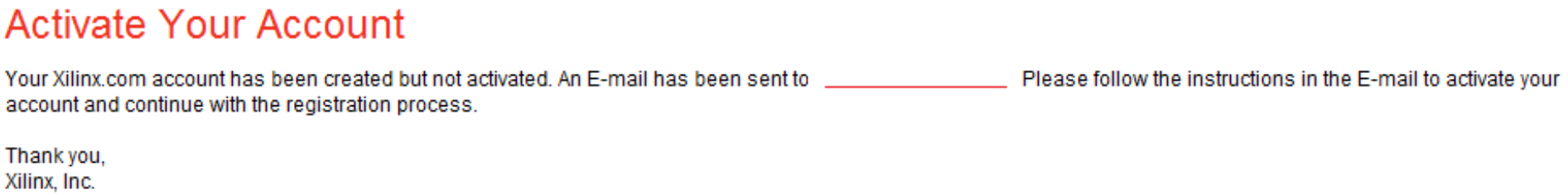




Xilinx License (2/3)



License檔案會寄到此處填的email地址



Xilinx License (3/3)

Create a New License File

Create a new license file by making your product selections from the table below. [?]

	Product	Type	License	Available Seats	Status	Subscription End Date
☐	ISE Design Suite: 30-Day Evaluation, Node-Locked License	Evaluation	Node	1/1	Current	30 days
☑	ISE Design Suite: WebPACK License	No Charge	Node	1/1	Current	None

登入後在此頁面選擇框起的選項，license檔案會寄到註冊時填的信箱

Generate Node-Locked License

Congratulations

Your new license file has been successfully generated and e-mailed to _____ You can also view the license file under the Manage Licenses tab.

Please add this sender (xilinx.notification@entitlenow.com) to your address book.

License File Details
Node License
Host ID: ANY

Products
ISE Design Suite: WebPACK License (No Charge): 1 seats

Design

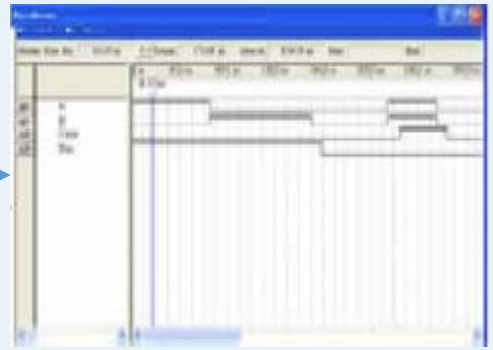
Compilation

Simulation

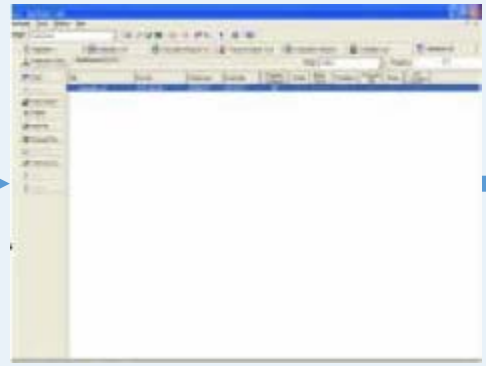
Verification

HDL Model

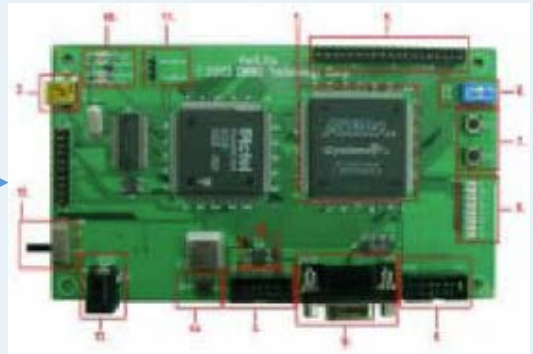
Compiler



Timing Analysis



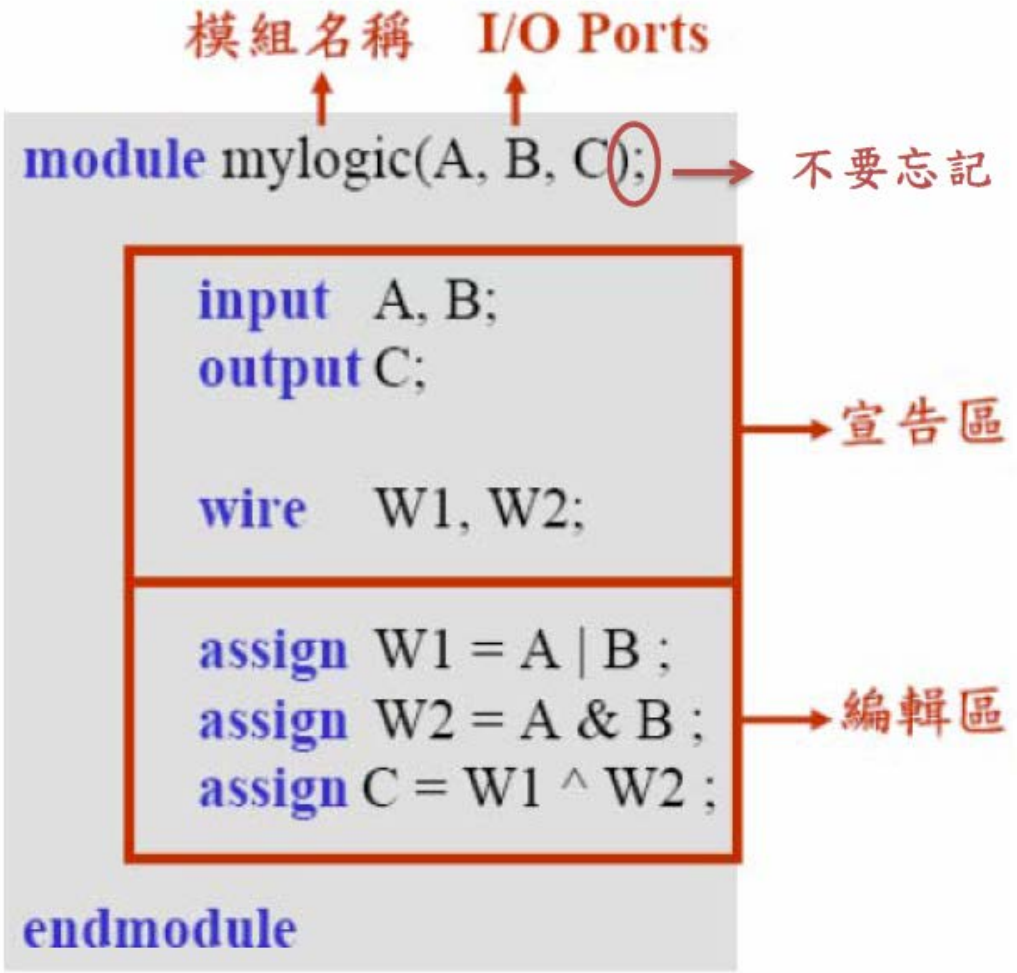
Program FPGA



SMIMS VeriLite

認識Verilog

認識Verilog


模組名稱
I/O Ports
module mylogic(A, B, C);
不要忘記
input A, B;
output C;
wire W1, W2;
宣告區
assign W1 = A | B ;
assign W2 = A & B ;
assign C = W1 ^ W2 ;
編輯區
endmodule

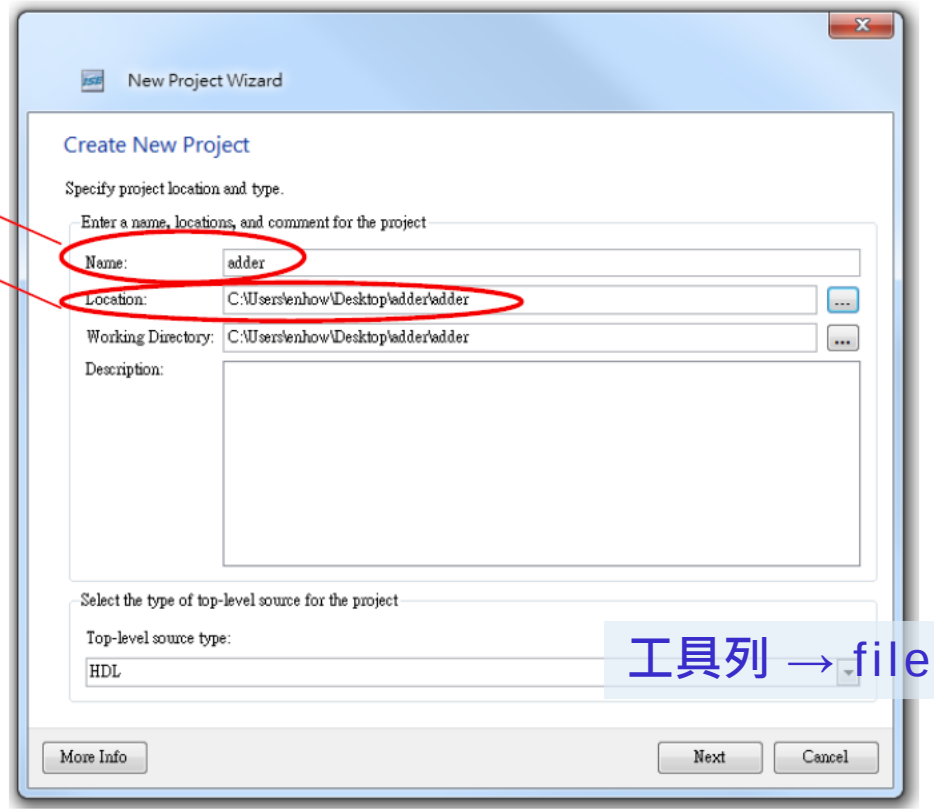
位元運算符號

邏輯閘	符號
AND	&
OR	|
NOT	~
XOR	^

例子:

基本專案建立與驗證

建立新專案(1/2)

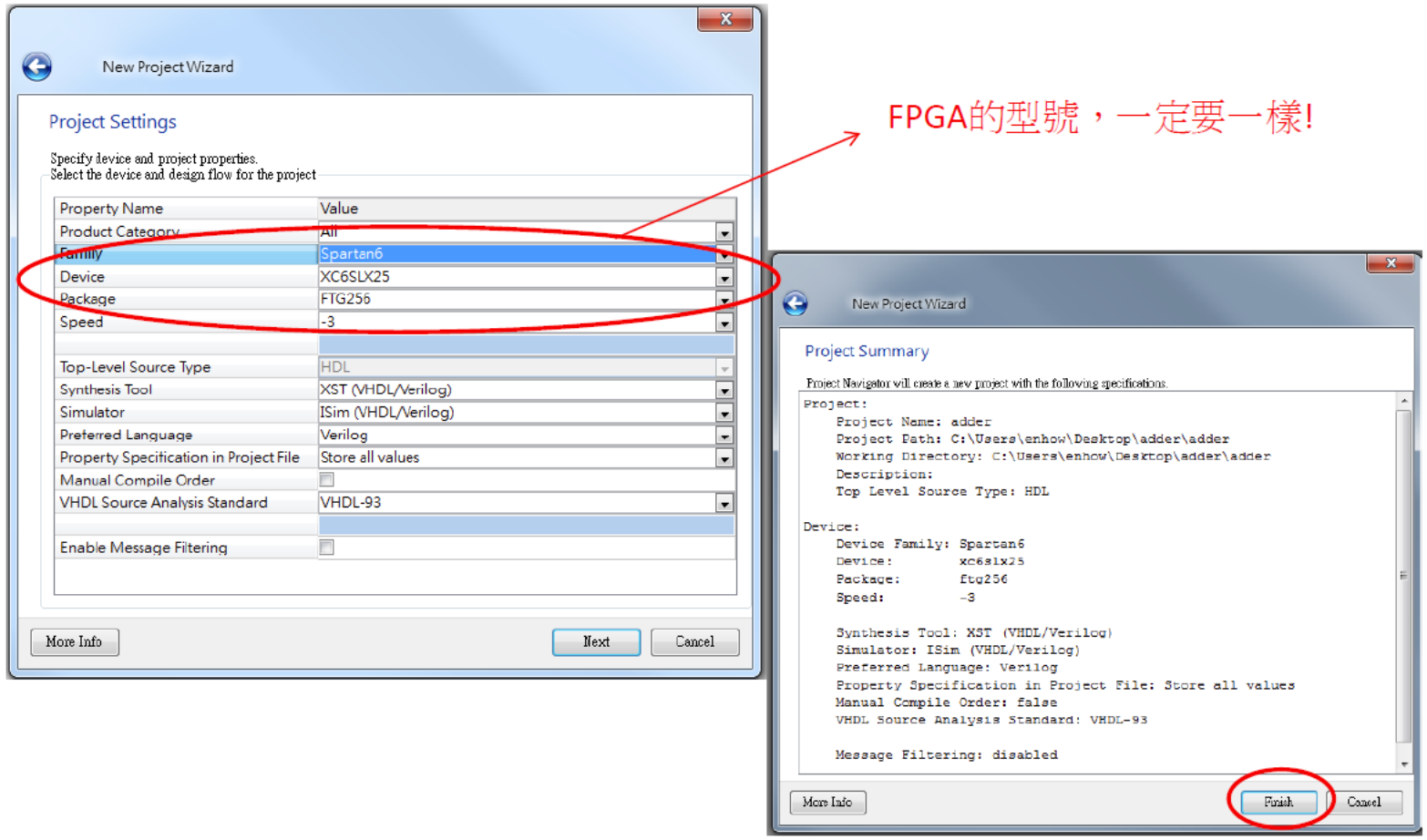


專案名稱

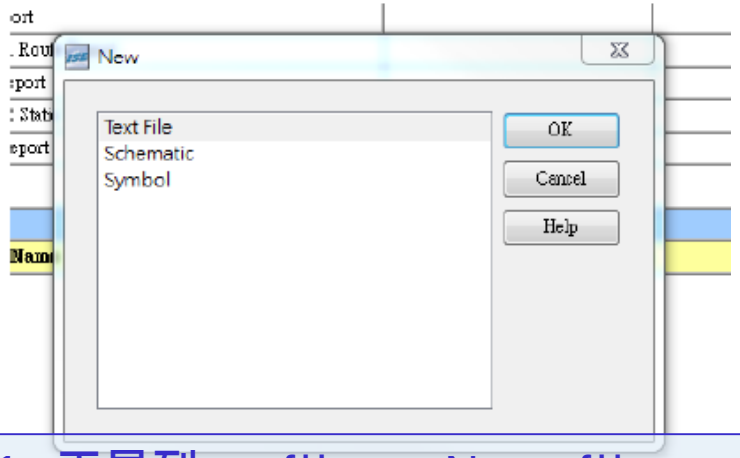
專案資料夾

工具列 → file → New Project

建立新專案(2/2)


New Project Wizard
Project Settings
Specify device and project properties.
Select the device and design flow for the project
Property Name
Value
Product Category
All
Family
Spartan6
Device
XC6SLX25
Package
FTG256
Speed
-3
Top-Level Source Type
HDL
Synthesis Tool
XST (VHDL/Verilog)
Simulator
ISim (VHDL/Verilog)
Preferred Language
Verilog
Property Specification in Project File
Store all values
Manual Compile Order
VHDL Source Analysis Standard
VHDL-93
Enable Message Filtering
More Info
Next
Cancel
FPGA的型號，一定要一樣!
New Project Wizard
Project Summary
Project Navigator will create a new project with the following specifications:
Project:
Project Name: adder
Project Path: C:\Users\enhow\Desktop\adder\adder
Working Directory: C:\Users\enhow\Desktop\adder\adder
Description:
Top Level Source Type: HDL
Device:
Device Family: Spartan6
Device: xc6slx25
Package: ftg256
Speed: -3
Synthesis Tool: XST (VHDL/Verilog)
Simulator: ISim (VHDL/Verilog)
Preferred Language: Verilog
Property Specification in Project File: Store all values
Manual Compile Order: false
VHDL Source Analysis Standard: VHDL-93
Message Filtering: disabled
More Info
Finish
Cancel

編輯Verilog檔 (1/2)



1. 工具列 → file → New file

2. Design & Testbench then save

```v
module adder(a,b,c);
input a,b;

output c;
assign c=a+b;

endmodule
```

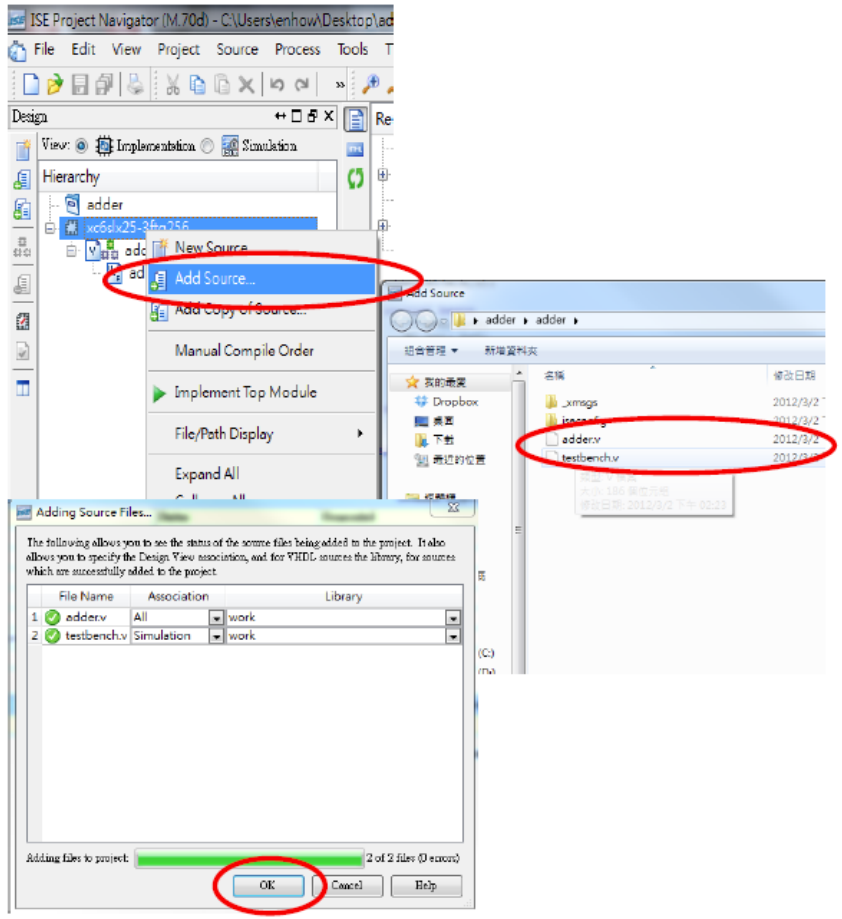
adder.v

```v
module testbench();

   reg a,b;
   wire c;

   adder X1(a,b,c);

   initial begin
         a=0; b=0;
      #1 a=1; b=0;
      #1 a=0; b=1;
      #1 a=1; b=1;
      #1 $finish;

   end

endmodule
```

testbench.v

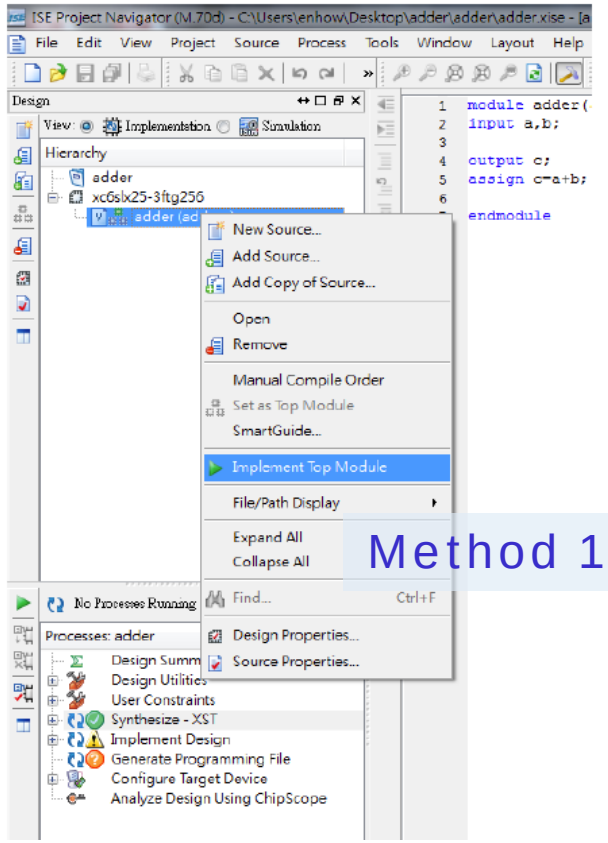
編輯Verilog檔 (2/2)

3. 加入 Project

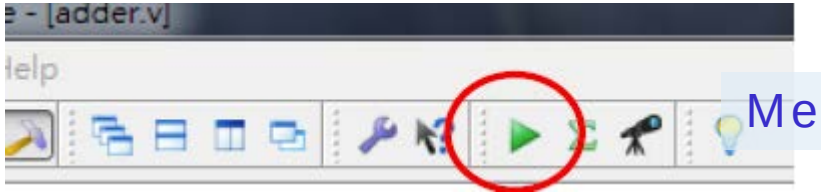


Compiler

Implement top module

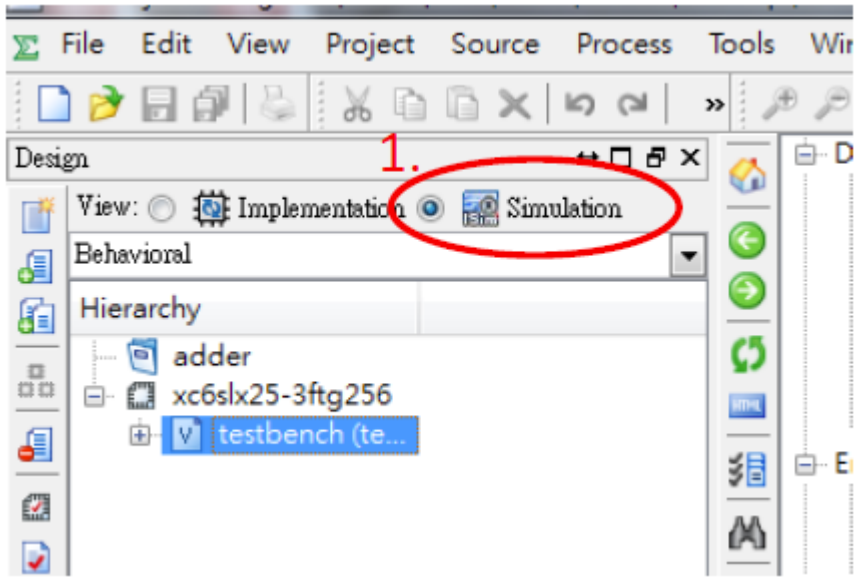


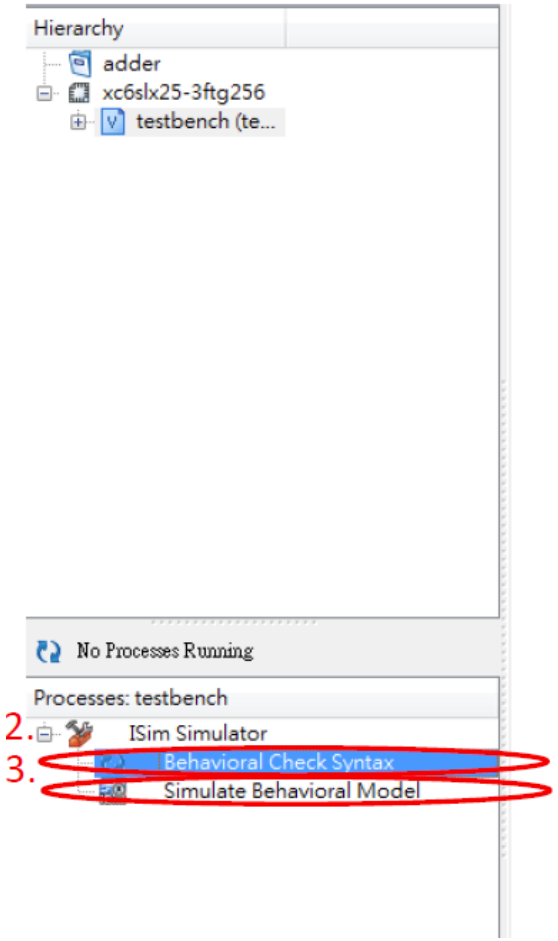
Method 1



Method 2

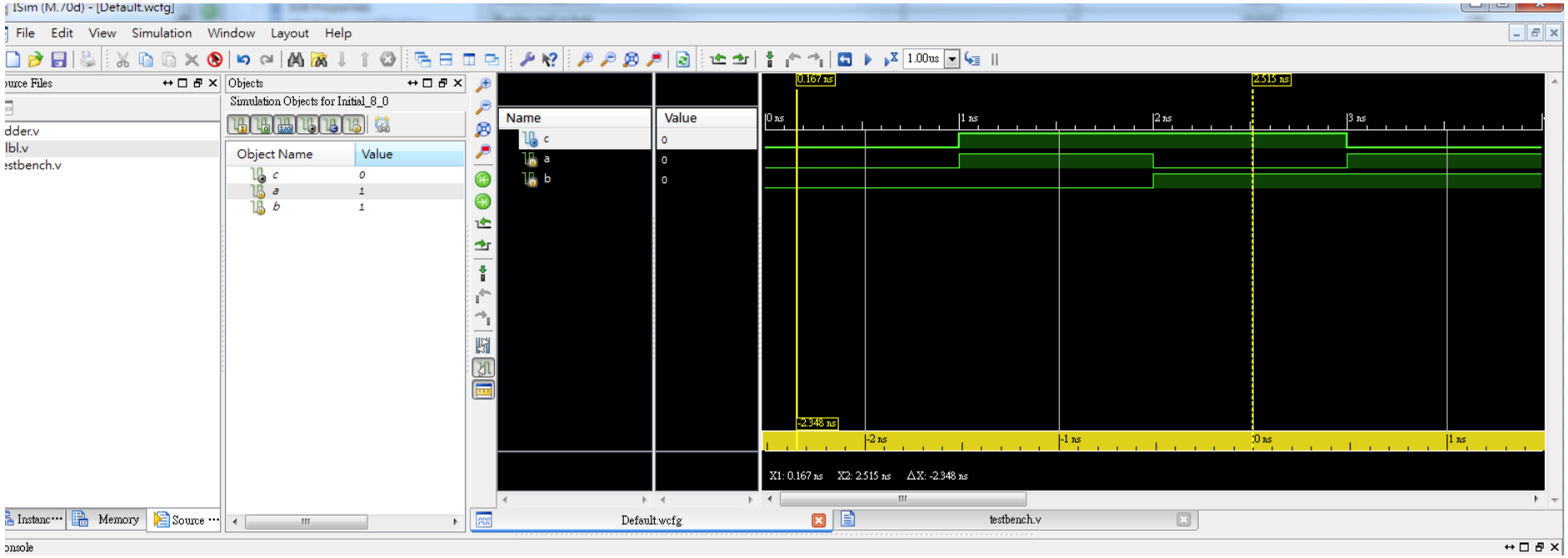
Simulation


File Edit View Project Source Process Tools
Design
1.
View: Implementation Simulation
Behavioral
Hierarchy
adder
xc6slx25-3ftg256
testbench (te...


Hierarchy
adder
xc6slx25-3ftg256
testbench (te...
No Processes Running
Processes: testbench
2.
ISim Simulator
3.
Behavioral Check Syntax
Simulate Behavioral Model

Simulation Result

之後就會跳出模擬波形的視窗



Design Summary/Report

工作列 → Project → Design Summary/Report

adder Project Status (03/02/2012 - 15:33:28)

Project File:	adder.xise	Parser Errors:	No Errors
Module Name:	adder	Implementation State:	Programming File Generated
Target Device:	xc6slx25-3ftg256	• Errors:	No Errors
Product Version:	ISE 12.3	• Warnings:	No Warnings
Design Goal:	Balanced	• Routing Results:	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	0 (Timing Report)

Timing Report

Device Utilization Summary [-]

Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	0	30,064	0%	
Number of Slice LUTs	1	15,032	1%	
Number used as logic	1	15,032	1%	
Number using O6 output only	1			
Number using O5 output only	0			
Number using O5 and O6	0			
Number used as ROM	0			
Number used as Memory	0	3,664	0%	
Number of occupied Slices	1	3,758	1%	
Number of LUT Flip Flop pairs used	1			
Number with an unused Flip Flop	1	1	100%	
Number with an unused LUT	0	1	0%	
Number of fully used LUT-FF pairs	0	1	0%	
Number of slice register sites lost to control set restrictions	0	30,064	0%	
Number of bonded IOBs	3	186	1%	
Number of RAMB16BWERs	0	52	0%	
Number of RAMB8BWERs	0	104	0%	
Number of BUFIO2/BUFIO2_2CLKs	0	32	0%	
Number of BUFIO2FB/BUFIO2FB_2CLKs	0	32	0%	
Number of BUFG/BUFGMUXs	0	16	0%	

一些詳細的報告

Detailed Reports [-]

Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	週五 三月 2 15:32:13 2012	0	0	0
Translation Report	Current	週五 三月 2 15:32:17 2012	0	0	0
Map Report	Current	週五 三月 2 15:32:34 2012	0	0	6 Infos (0 new)
Place and Route Report	Current	週五 三月 2 15:32:51 2012	0	0	2 Infos (0 new)
Power Report					
Post-PAR Static Timing Report	Current	週五 三月 2 15:32:58 2012	0	0	3 Infos (0 new)
Bitgen Report	Current	週五 三月 2 15:33:17 2012	0	0	0

Timing Report

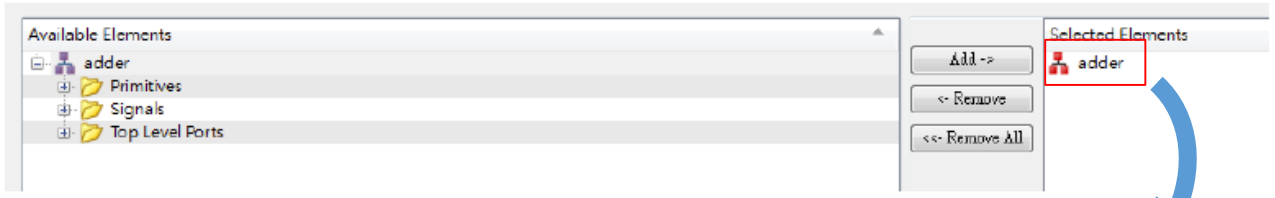
```
Environment Variable      Effect
--------------------      ------
NONE                      No environment variables were set
--------------------------------------------------------------------------------

INFO:Timing:2698 - No timing constraints found, doing default enumeration.
INFO:Timing:2752 - To get complete path coverage, use the unconstrained paths
   option. All paths that are not constrained will be reported in the
   unconstrained paths section(s) of the report.
INFO:Timing:3339 - The clock-to-out numbers in this timing report are based on
   a 50 Ohm transmission line loading model.  For the details of this model,
   and for more information on accounting for different loading conditions,
   please see the device datasheet.


Data Sheet report:
-----------------
All values displayed in nanoseconds (ns)

Pad to Pad
+---------------+---------------+---------+
Source Pad      |Destination Pad|  Delay  |
+---------------+---------------+---------+
a               |c              |    6.923|
b               |c              |    6.917|
+---------------+---------------+---------+


Analysis completed Fri Mar 02 15:32:58 2012
--------------------------------------------------------------------------------

Trace Settings:
------------------------
Trace Settings

Peak Memory Usage: 227 MB
```

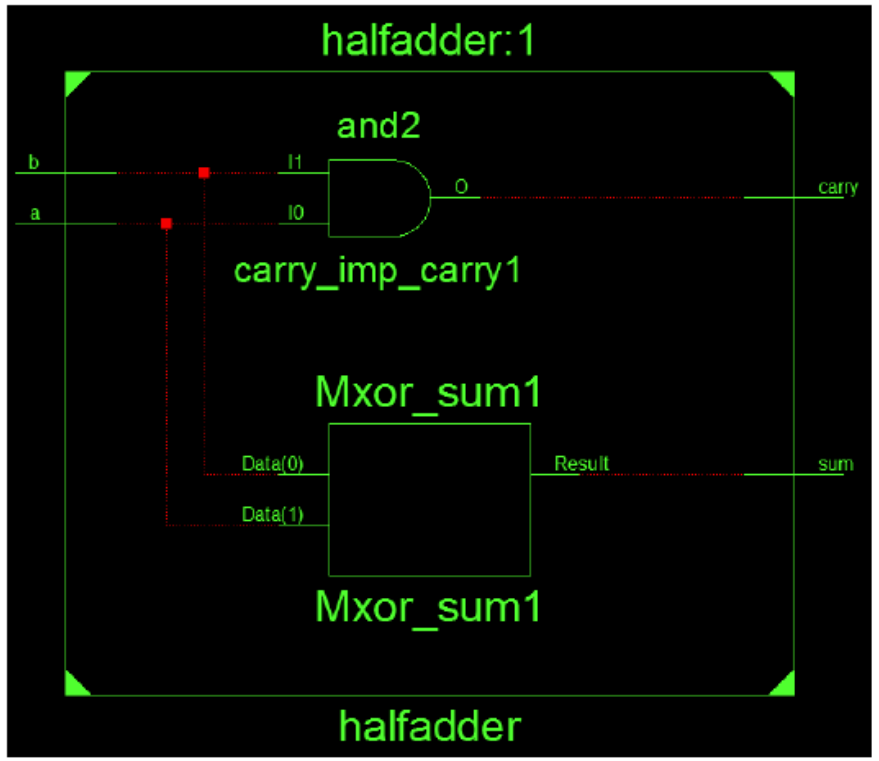
可以看到path delay

RTL viewer

工作列
→ Tools
→ Schematic Viewer
→ RTL

可觀看所設計電路Gate-Level diagram



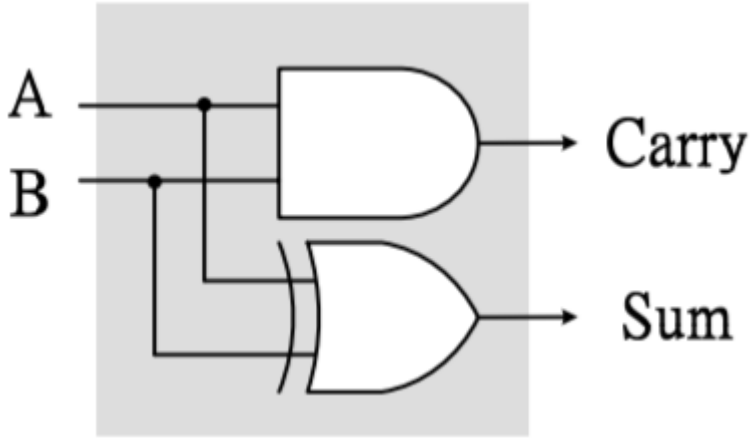


實作題

本次實作分為兩個基本實作與一個挑戰實作

實作題(一): 半加器(1/6)

➢利用Verilog的基本邏輯語法組合出下圖的電路並且驗證



(halfadder.v)

邏輯閘	符號
AND	&
OR	|
NOT	~
XOR	^

實作題(一): 半加器(2/6)

步驟:

1. 建立 **C:\logiclab\<學號>\lab4_1** 資料夾。
2. 依上面投影片所示，建立一個 project。
3. 在ISE開啟新的Verilog檔案，並把名稱設為**halfadder.v**，儲存在上述資料夾。

實作題(一): 半加器(3/6)

4. 透過Verilog的語法設計指定的邏輯元件。

請同學先鍵入下列程式碼，並且將空白處填完後，再按一次存檔

```
module halfadder(A, B, Carry, Sum);

    input A,B;
    output Carry, Sum;

    assign Carry =  A & B;
    assign Sum =          ;

endmodule
```

實作題(一): 半加器(4/6)

5. 在ISE再開啟新的Verilog檔案，並把名稱設為**testbench.v**，儲存在上述資料夾。

6. 鍵入右邊程式碼，再按一次存檔

```
module testbench();

    reg A, B;
    wire carry, sum;

    halfadder N1(A, B, carry, sum);

    initial begin
        A = 0; B = 0;
    #1  A = 1; B = 0;
    #1  A = 0; B = 1;
    #1  A = 1; B = 1;
    #1 $finish;
    end

endmodule
```

實作題(一): 半加器(5/6)

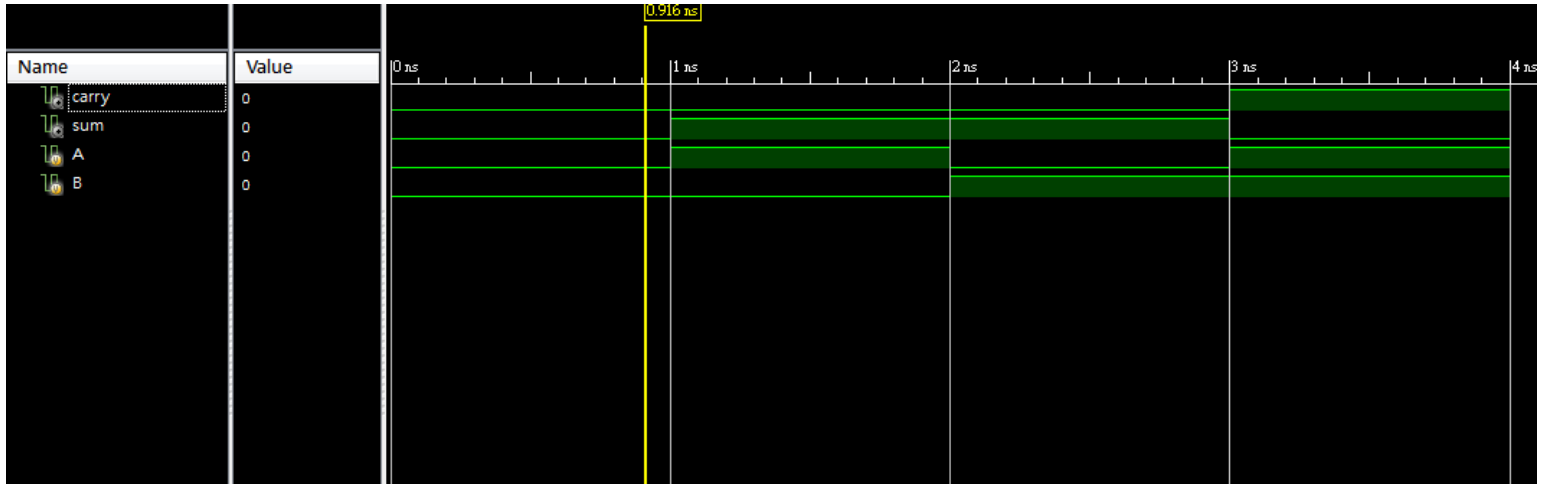
7. 依投影片所示，將**halfadder.v**和**testbench.v**都加入project。

8. 依投影片所示開始編譯檔案，Compiler後會顯示是否出現Errors、Warnings，若有Error發生，可直接用滑鼠點錯誤訊息兩 下即可直接跳至錯誤處。

實作題(一): 半加器(6/6)

9. 驗證結果是否符合預期。

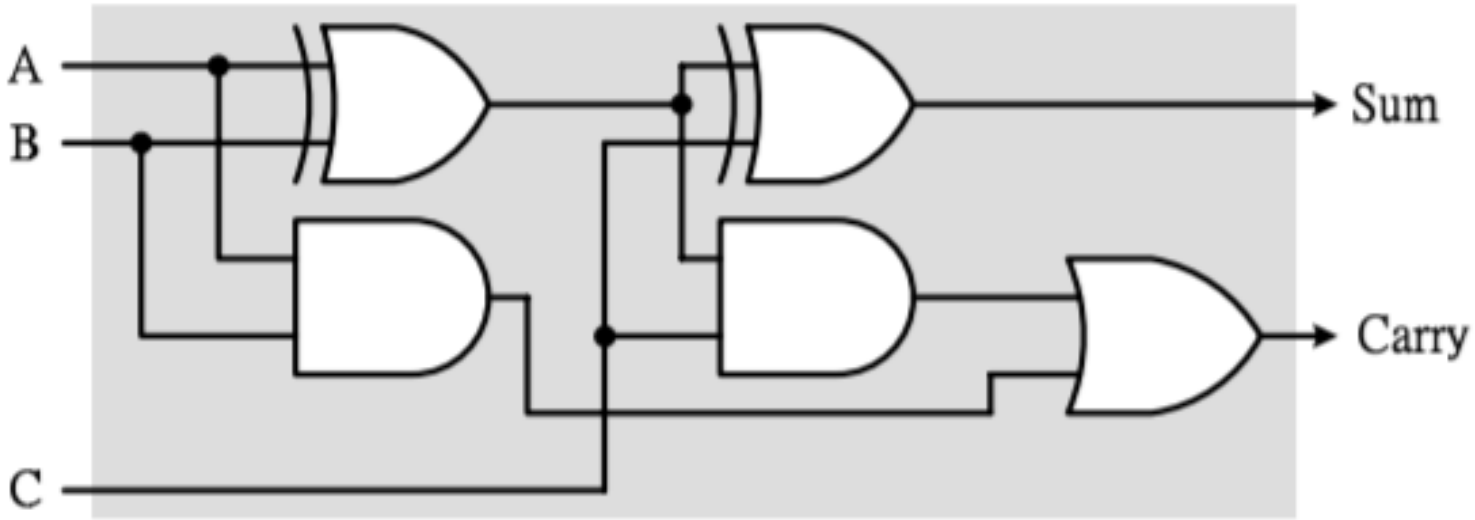
依投影片所示，執行電路模擬並觀察波形，如下圖。



A	B	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

實作題(二): 全加器(1/6)

- 利用Verilog的基本邏輯語法組合出下圖的電路並且驗證



(fulladder.v)

實作題(二): 全加器(2/6)

步驟

1. 建立 **C:\logiclab\<學號>\lab4_2** 資料夾。
2. 依投影片所示，建立一個project。
3. 在ISE開啟新的Verilog檔案，並把名稱設為**fulladder.v**， 儲存在先前lab4_2的資料夾。

實作題(二): 全加器(3/6)

4. 透過Verilog的基本邏輯語法設計指定的邏輯元件。

請同學先鍵入下列程式碼，並且將空白處填完後，再按一次存檔

```
module fulladder(A, B, C, Carry, Sum);

    input A,B;
    output Carry, Sum;

    assign Carry =                ;
    assign Sum =                  ;

endmodule
```

實作題(二): 全加器(4/6)

5. 在ISE再開啟新的Verilog檔案，並把名稱設為**testbench.v**， 儲存在上述資料夾。

6. 鍵入右邊程式碼，再按一次存檔。

```
module testbench();

    reg a, b, c;
    wire ca, s;

    fulladder N1(a, b, c, ca, s);

    initial begin
        a = 0; b = 0; c = 0;
    #1  a = 0; b = 0; c = 1;
    #1  a = 0; b = 1; c = 0;
    #1  a = 0; b = 1; c = 1;
    #1  a = 1; b = 0; c = 0;
    #1  a = 1; b = 0; c = 1;
    #1  a = 1; b = 1; c = 0;
    #1  a = 1; b = 1; c = 1;
    #1 $finish;
    end

endmodule
```

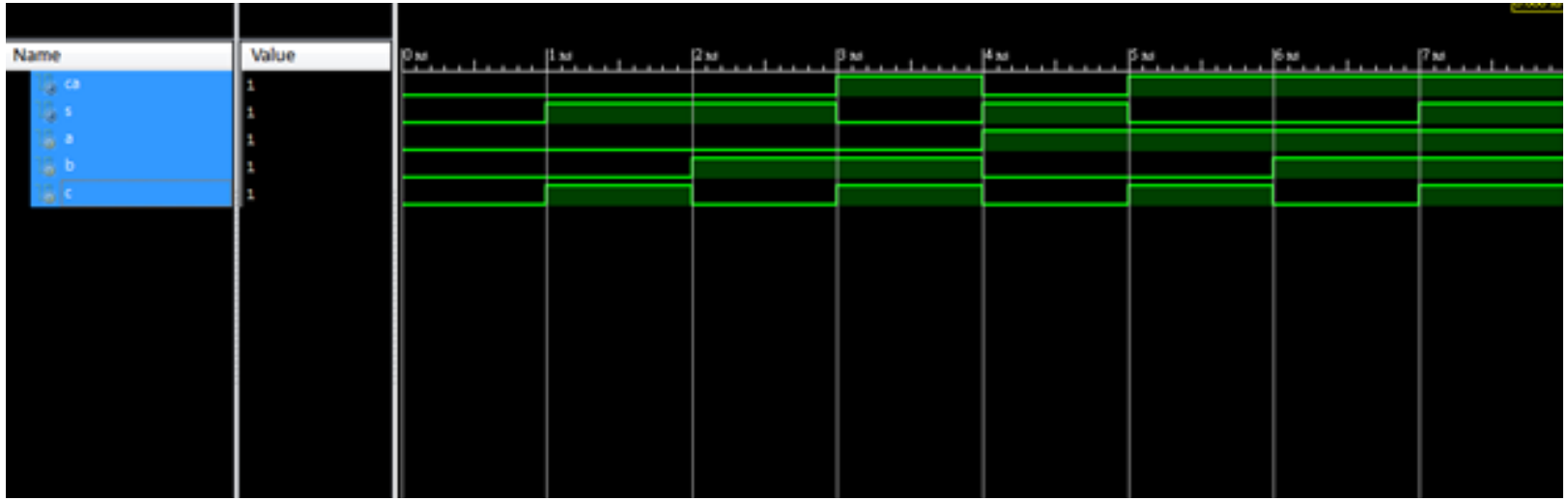
實作題(二): 全加器(5/6)

7. 依投影片所示，將**fulladder.v** 和**testbench.v**都加入project。
8. 依投影片所示開始編譯檔案，Compiler後會顯示是否出現Errors、Warnings，若有Error發生，可直接用滑鼠點錯誤訊息兩 下即可直接跳至錯誤處。

實作題(二): 全加器(6/6)

8. 驗證結果是否符合預期。

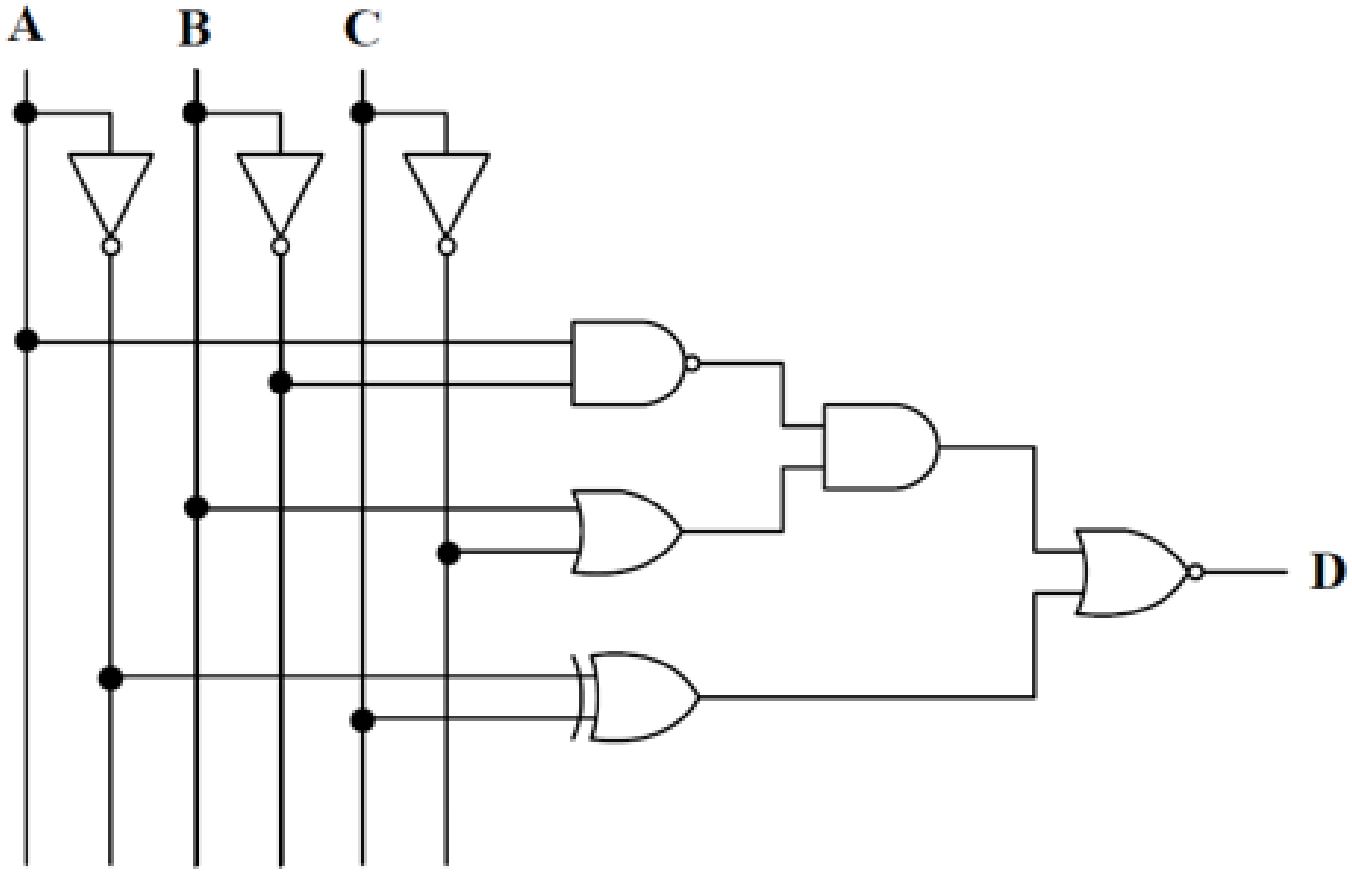
依投影片所示，執行電路模擬並觀察波形，如下圖。



A	B	C	Carry	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

實作題(三): 基本邏輯(1/3)

- 利用Verilog的基本邏輯語法組合出下圖的電路並且驗證



實作題(三): 基本邏輯(2/3)

步驟:

1. 建立**C:\logiclab\<學號>\lab4_3**資料夾。
2. 請使用ISE將上圖用Verilog實現。
3. 設計一個testbench

實作題(三): 基本邏輯(2/3)

4. 將檔案Compile及Simulation，並檢查波形是否與下圖相符。