LAB - 04

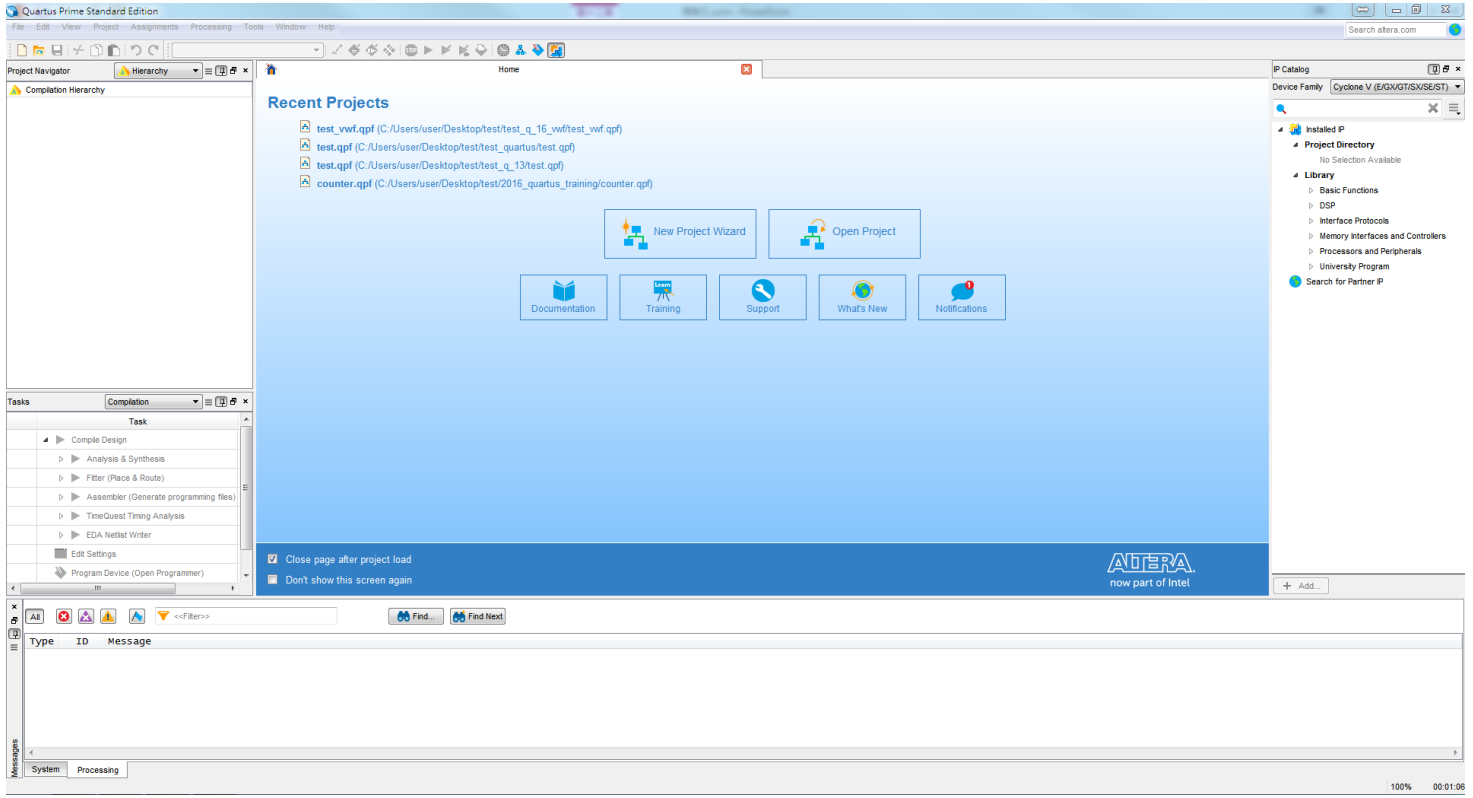
陳培殷
國立成功大學　資訊工程系

Outline

- Video preview for 晶片實現+HDL介紹(Parts I~II)
- Quartus II tutorial
- Introduction to DE0-CV
- Programming DE0-CV
- Lab I--full adder implement
- Lab II – full adder to DE0-CV
- Appendix- Quartus II simulation

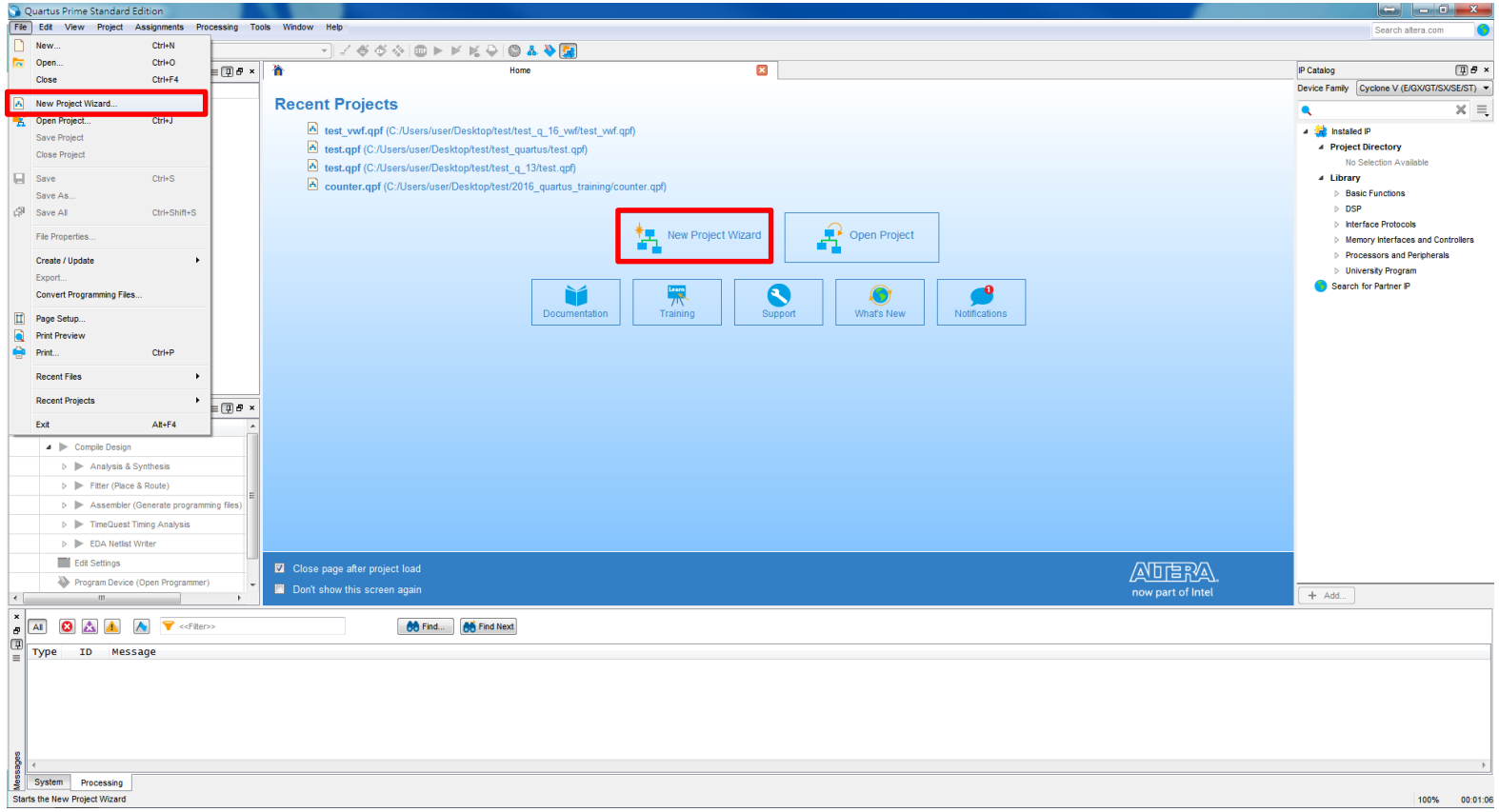
Quartus II Tutorial (1/10)

- **Getting Started –**
  - **Start the Quartus II software**



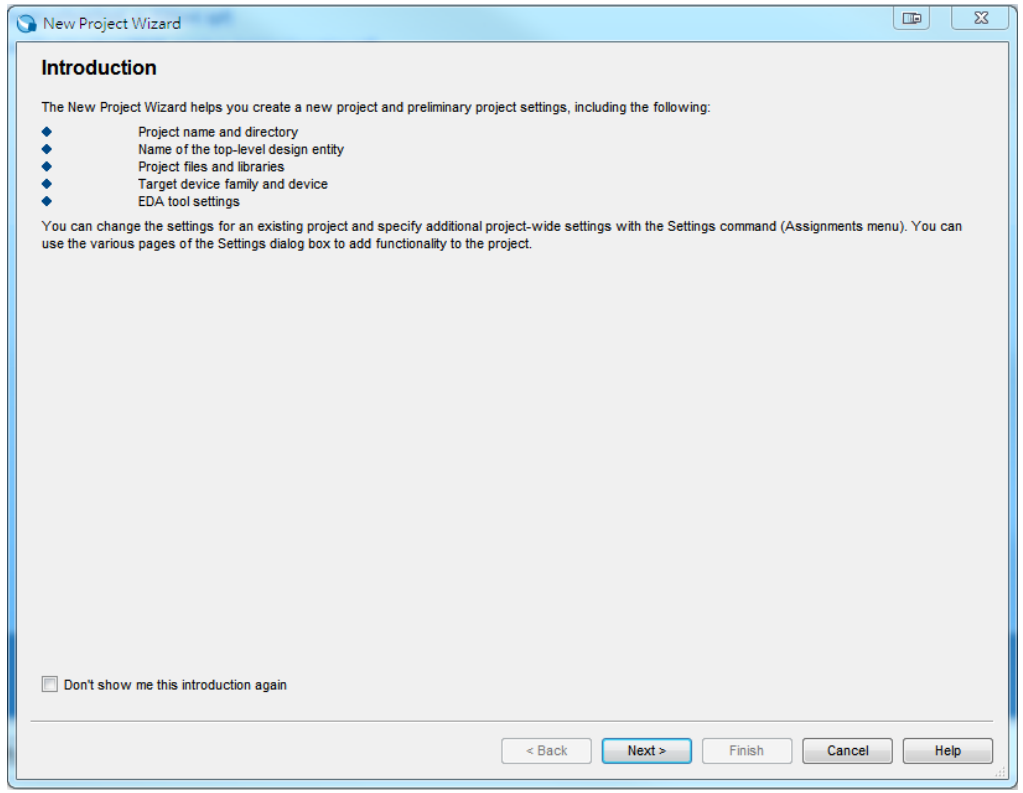
Quartus II Tutorial (2/10)

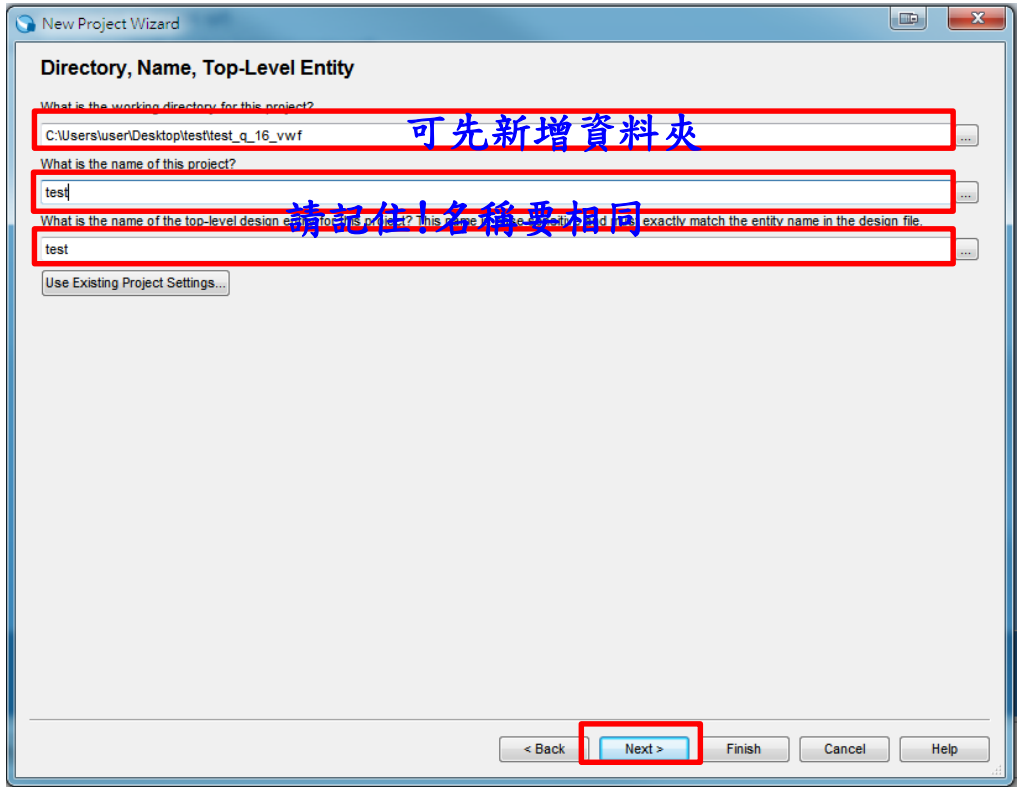
- **Create a New Project –**
  - **Open New Project Wizard (File → New Project Wizard…)**



Quartus II Tutorial (3/10)

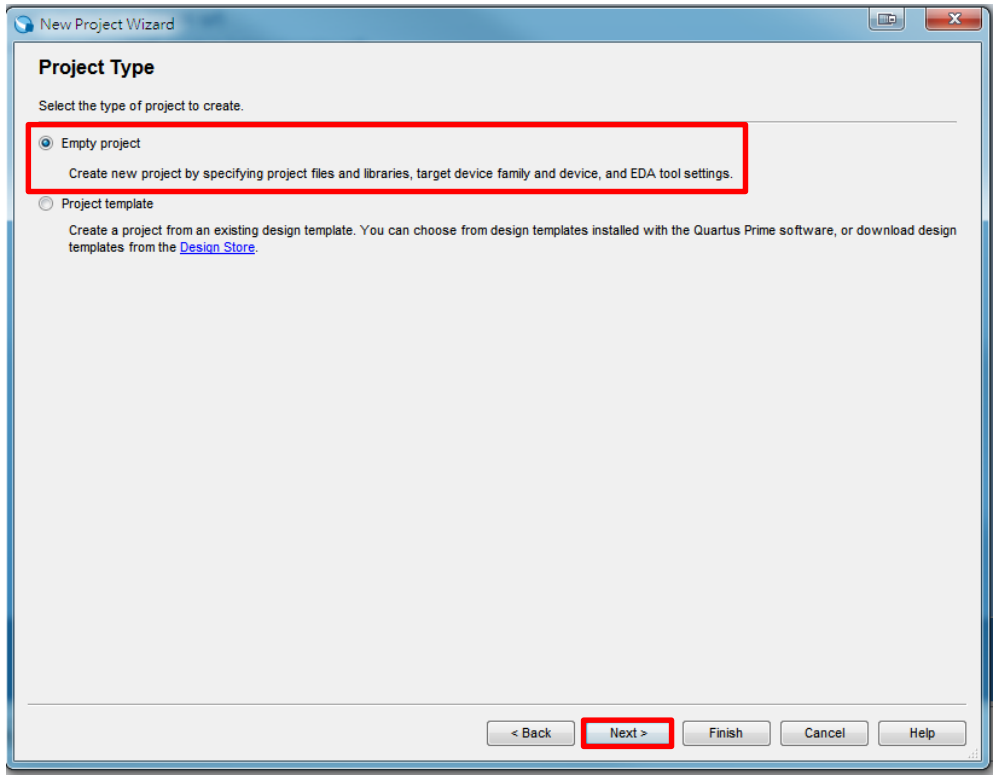
- **Specify the working directory and the name of the project**

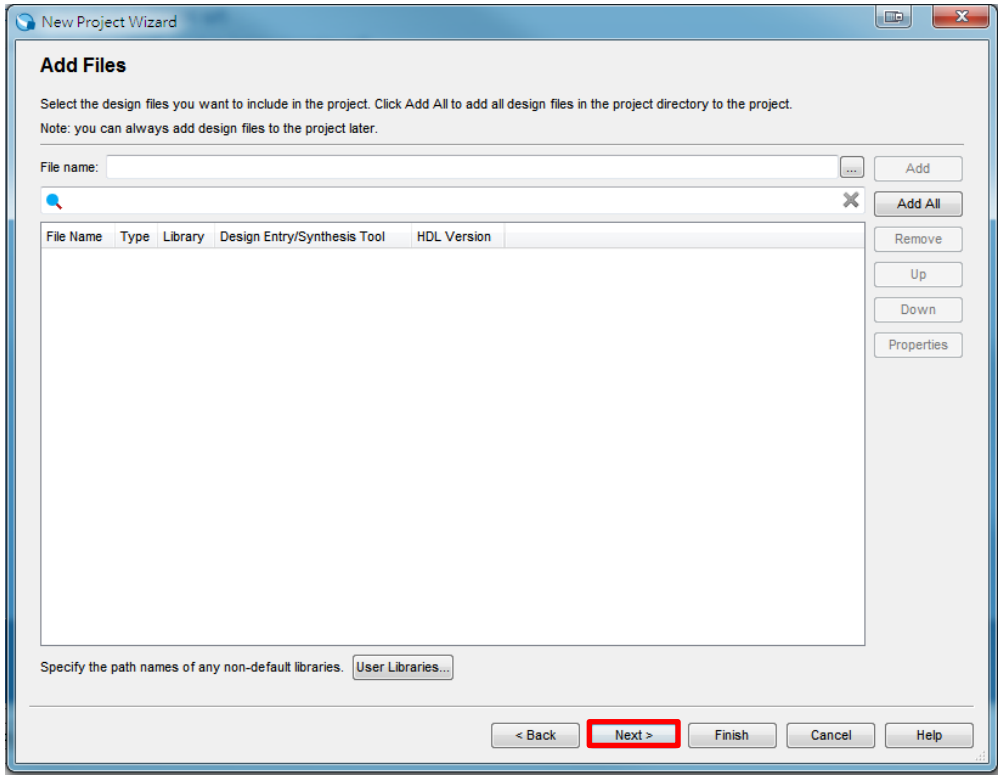




Quartus II Tutorial (4/10)

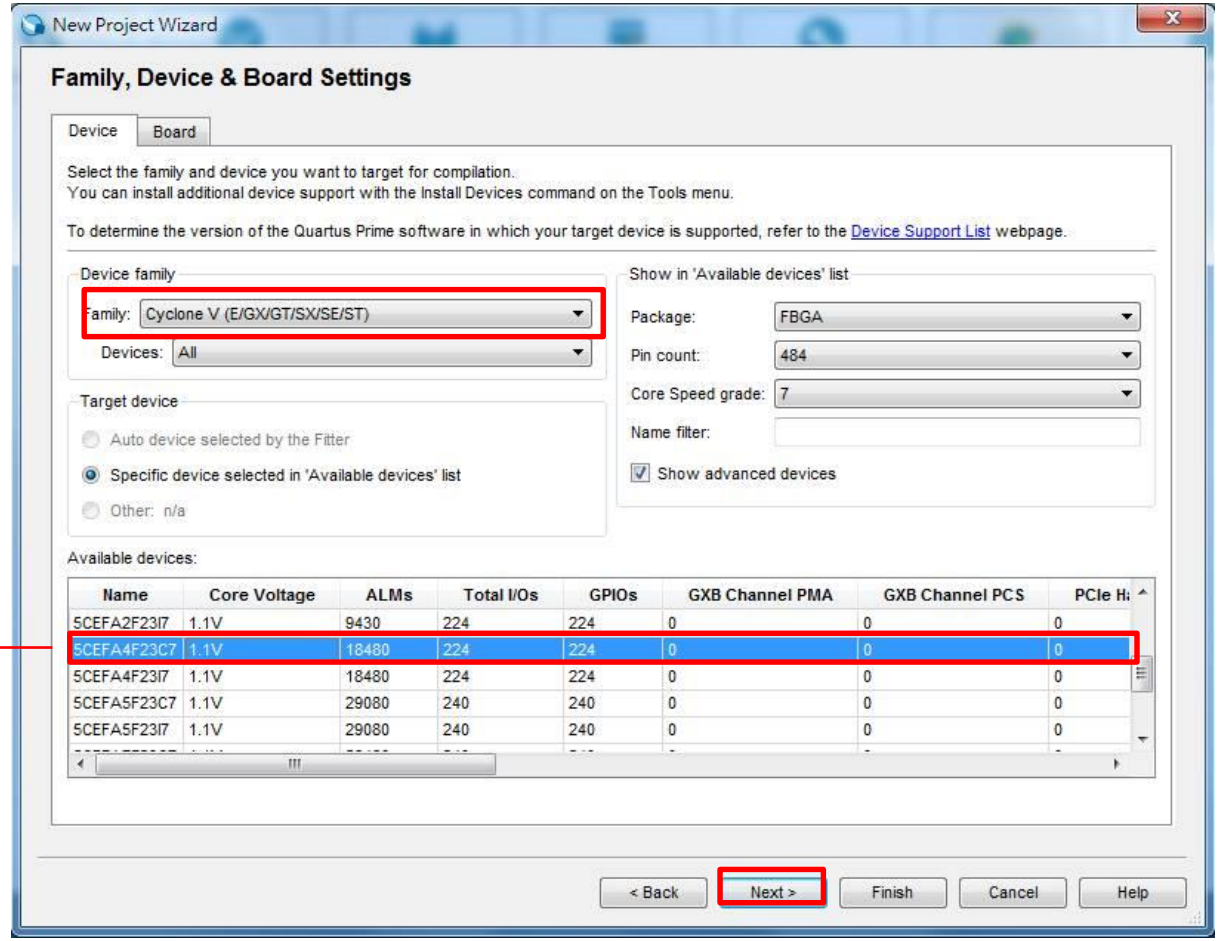
- Select “Empty project”. Then, click “Next”.
- Select design files. Or click “Next” to skip this step.





Quartus II Tutorial (5/10)

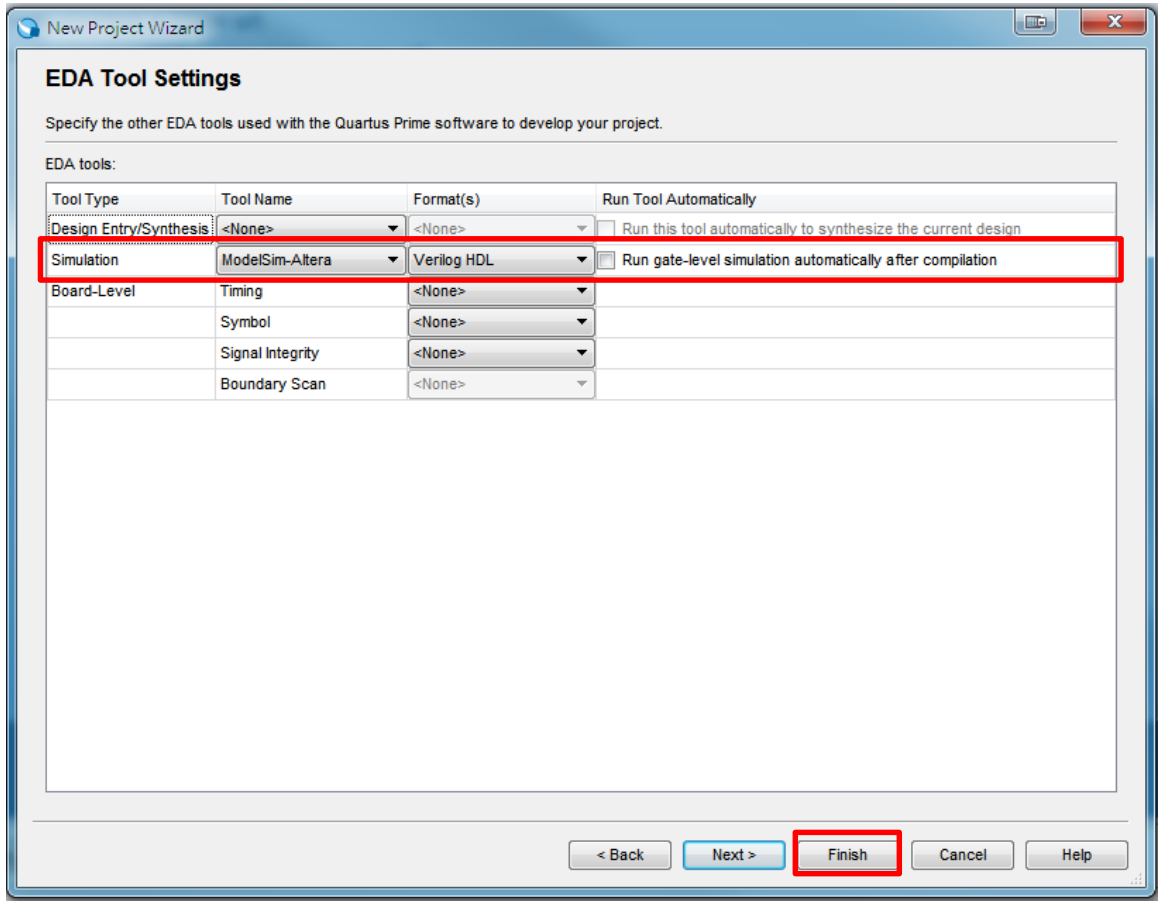
- **Specify device settings - (DE0-CV Device family are used). Click "Next."**



5CEFA4F23C7

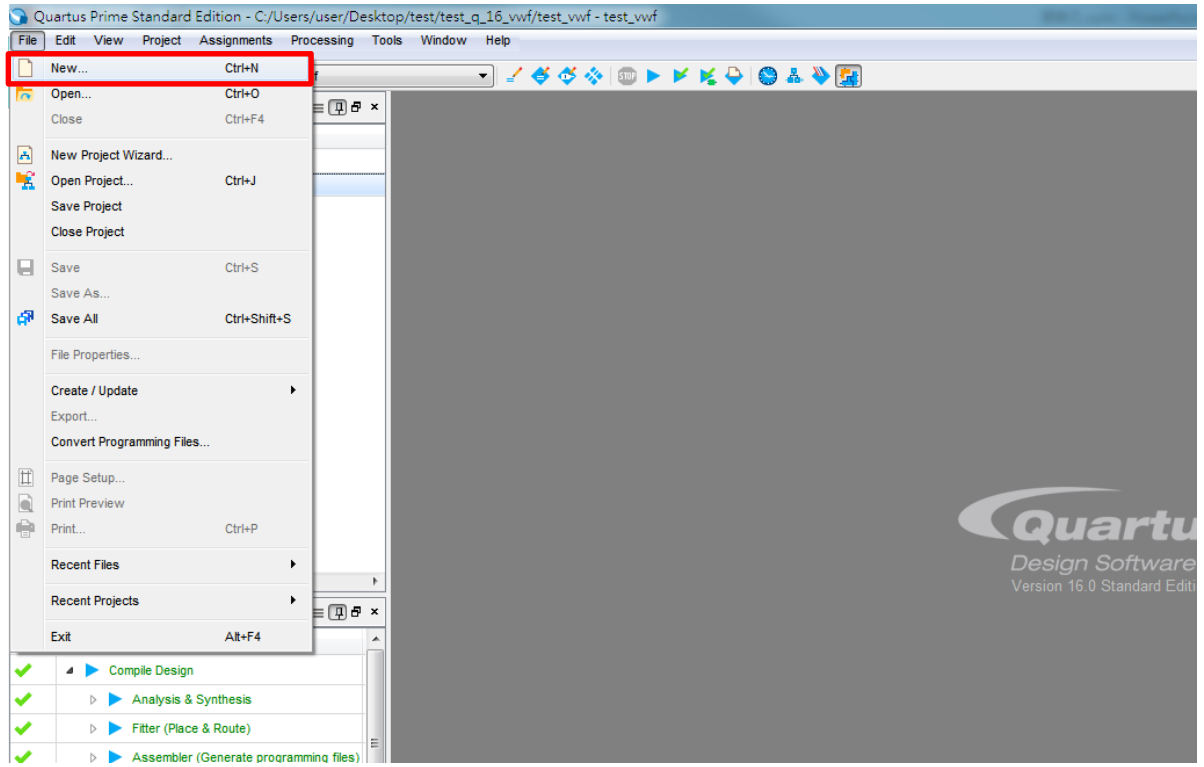
Quartus II Tutorial (6/10)

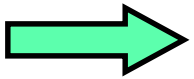
- **Specify EDA Tool – (Modelsim-Altera is selected for simulation). Click "Finish."**

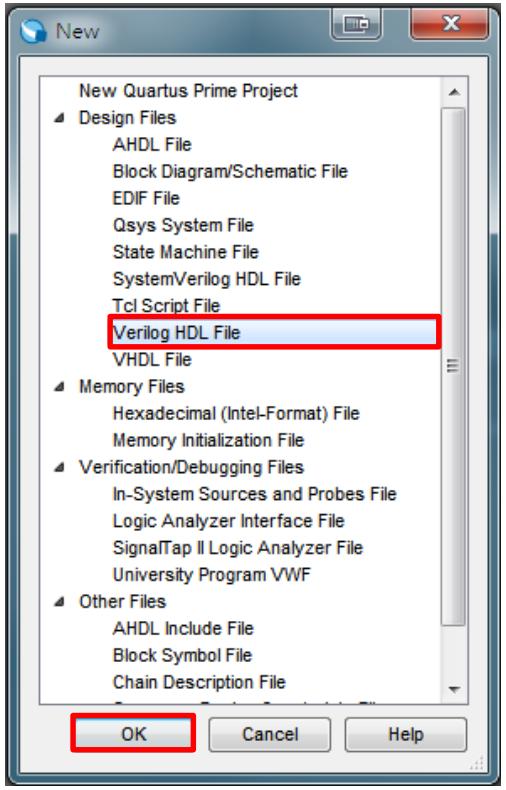


Quartus II Tutorial (7/10)

- **Edit a new file by opening a Verilog HDL file**
  - **(File → New → Verilog HDL File → OK)**







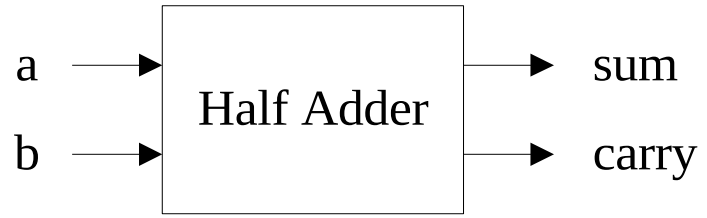
Quartus II Tutorial (8/10)

- **Write Verilog code**

Top module name 一定要跟 Project name 相同 !!

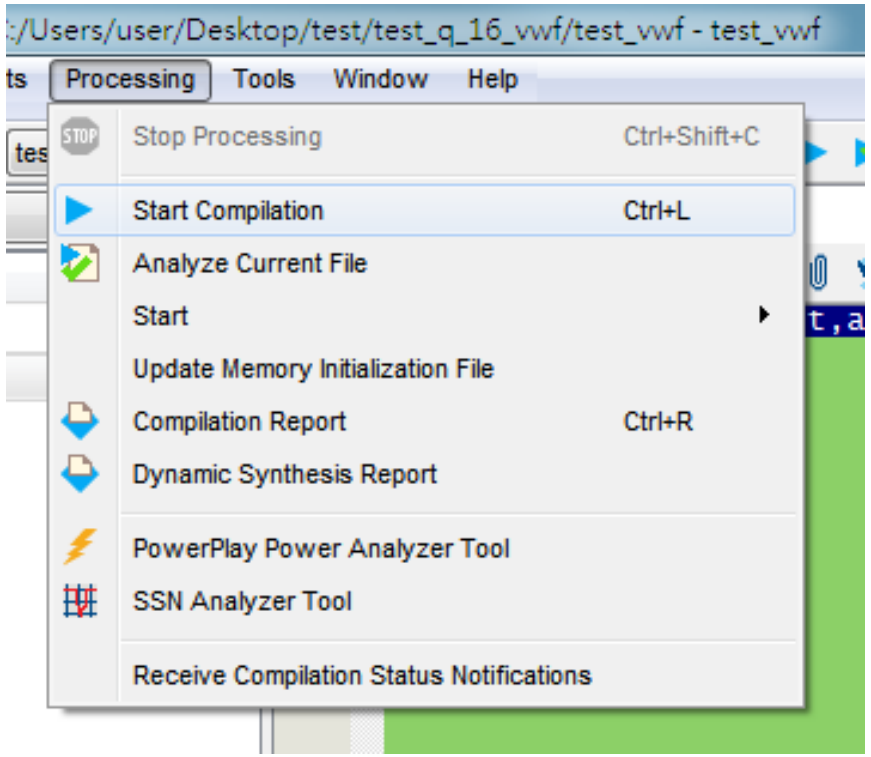
```
1: //File Name：Half_Adder.v
2: module    Half_Adder(a, b, sum, carry);
3: input     a, b;
4: output    sum, carry;
5:
6: assign sum = a ^ b;
7: assign carry = a & b;
8:
9: endmodule
```

輸入(input)		輸出(output)	
被加數(a)	加數(b)	和(sum)	進位(carry)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



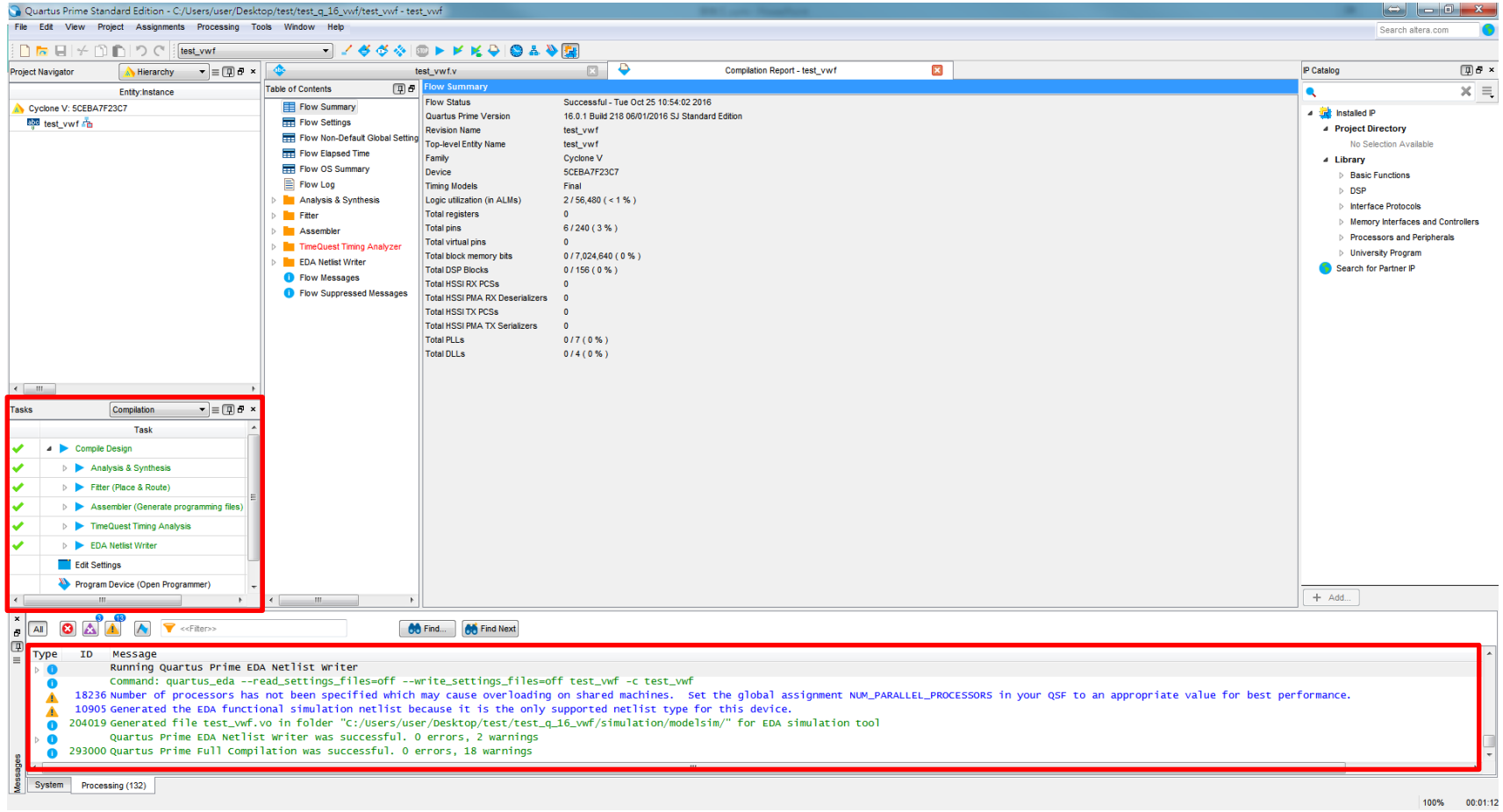
Quartus II Tutorial (9/10)

- **Compiling the Designed Circuit　(synthesis 合成)**
  - **(Processing → Start Compilation)**



Quartus II Tutorial (10/10)

- **Successful compilation**



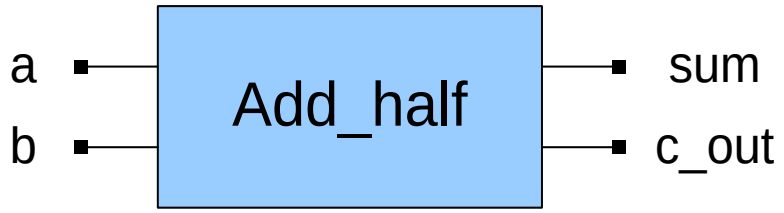
Half Adder

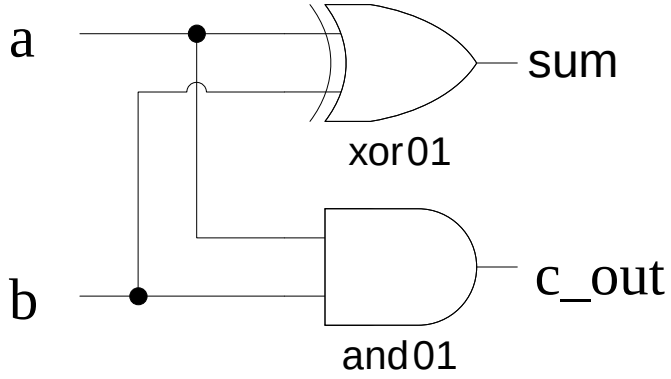
a\b	0	1
0	0	1
1	1	0

$sum = a \oplus b$

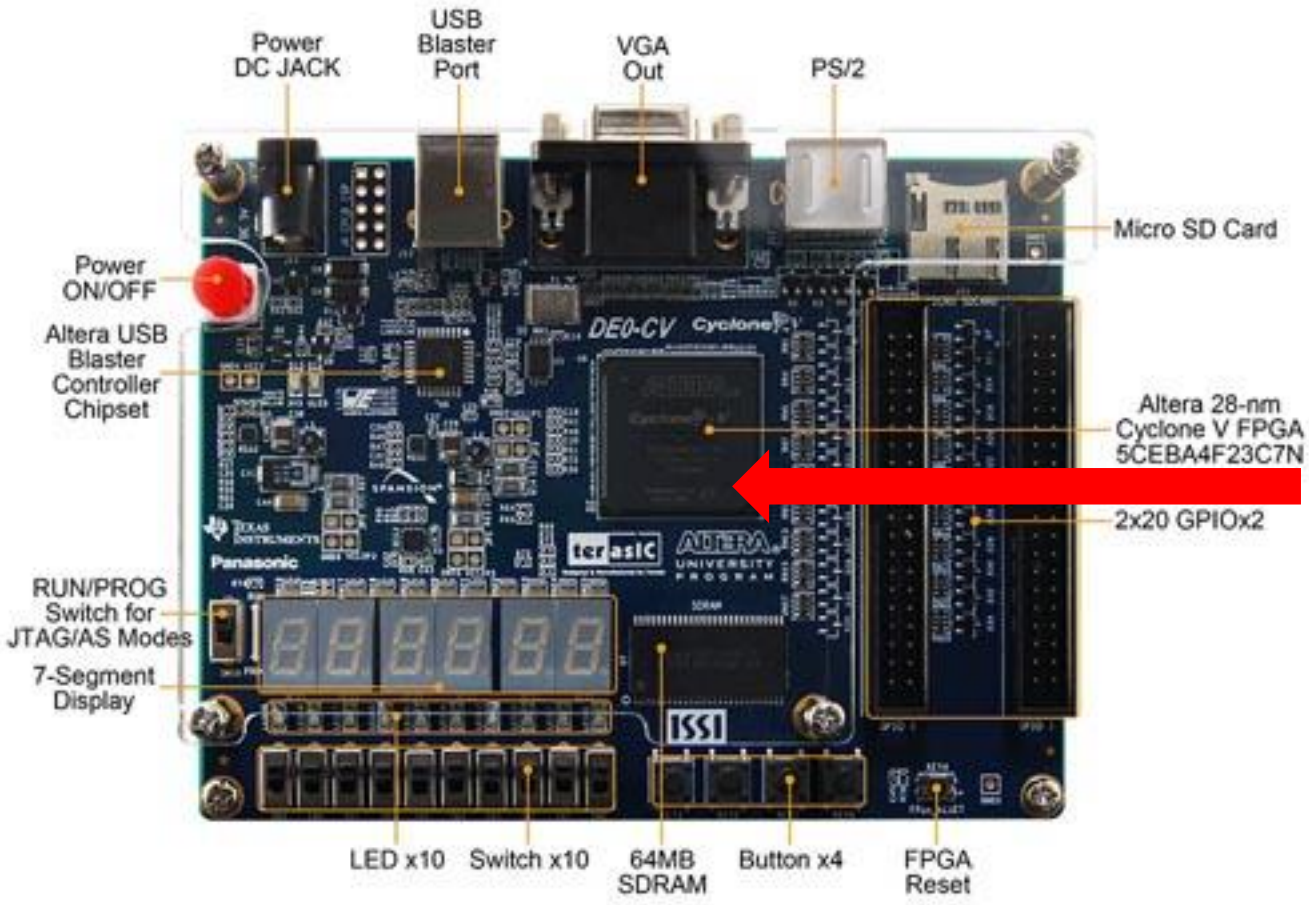
a\b	0	1
0	0	0
1	0	1

$c\_out = ab$





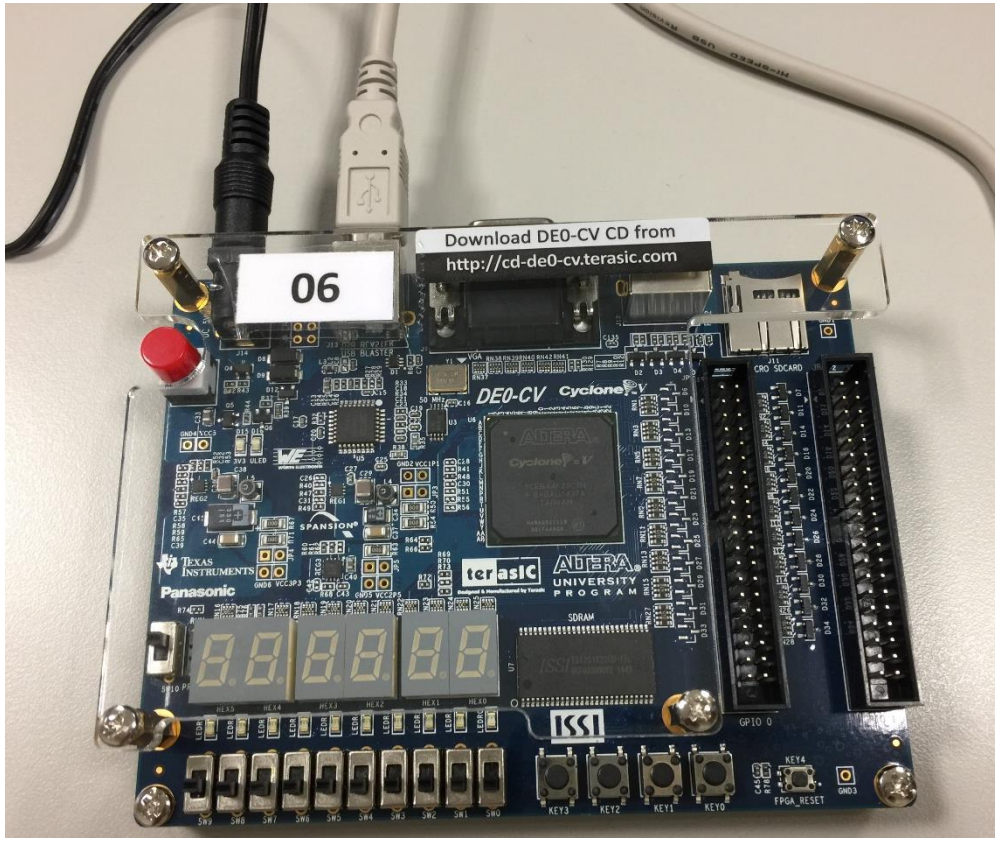
Introduction to DE0-CV(1/3)



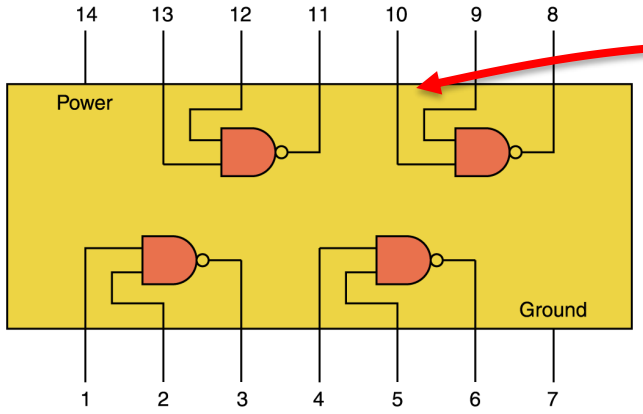
1. DE0-CV is a FPGA simulation board used to simulate the user-defined circuit.
2. You can program you circuit into the FPGA in the board for simulation.

Introduction to DE0-CV (2/3)

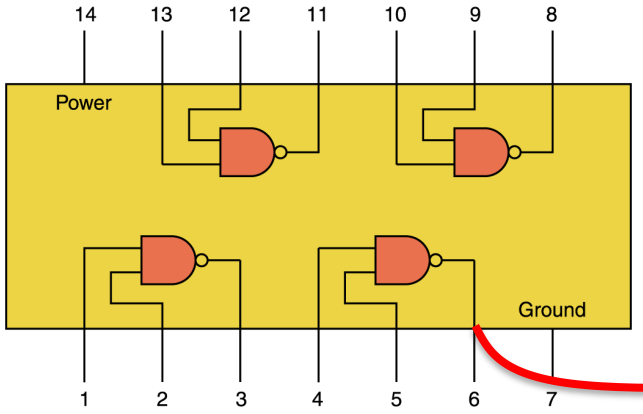
電源　USB



Introduction to DE0-CV (3/3)



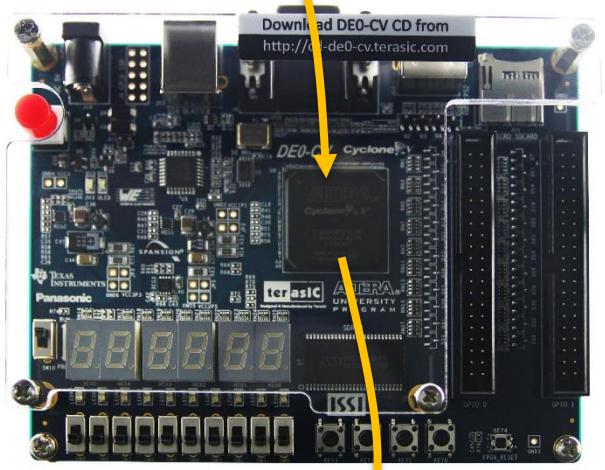
your circuit



inputs

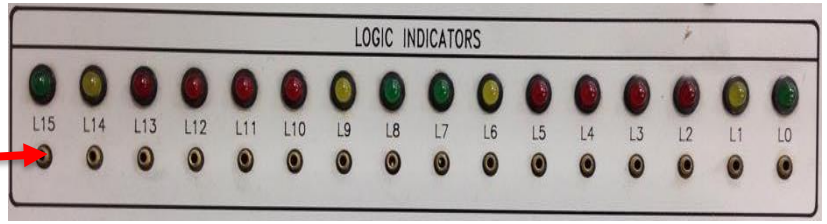
FPGA

Simulation



Traditional simulation

Solderless Breadboard



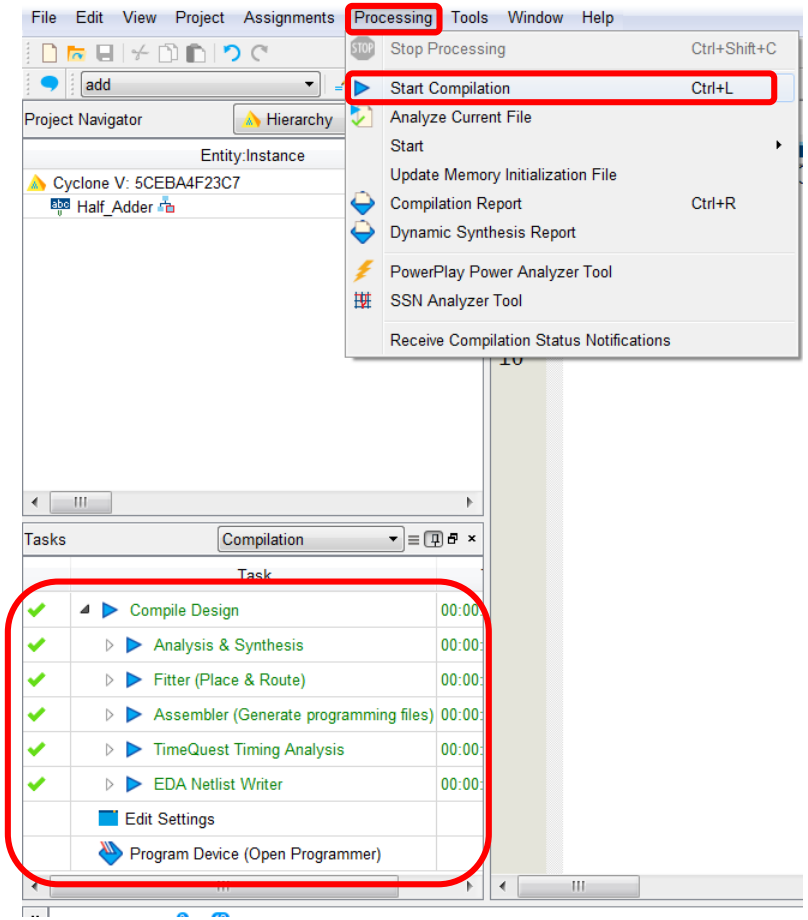
outputs

Programming DE0-CV (1/13)

```
 1   module   Half_Adder(a, b, sum, carry);
 2
 3   input a,b;
 4   output   sum, carry;
 5
 6   and(carry,a,b);
 7   xor(sum,a,b);
 8
 9   endmodule
10
```

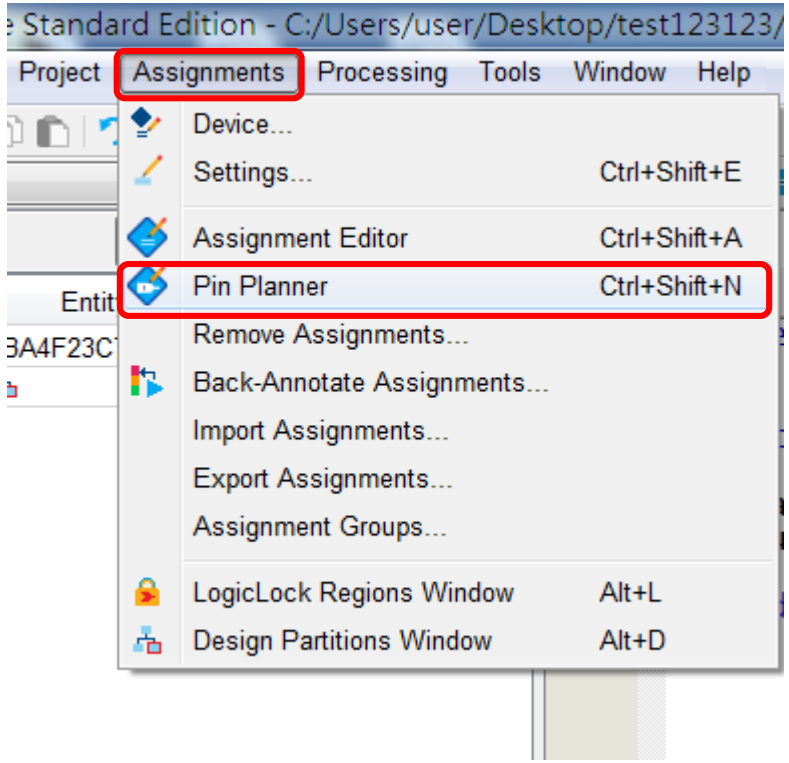
Programming DE0-CV (2/13)

- Start compilation



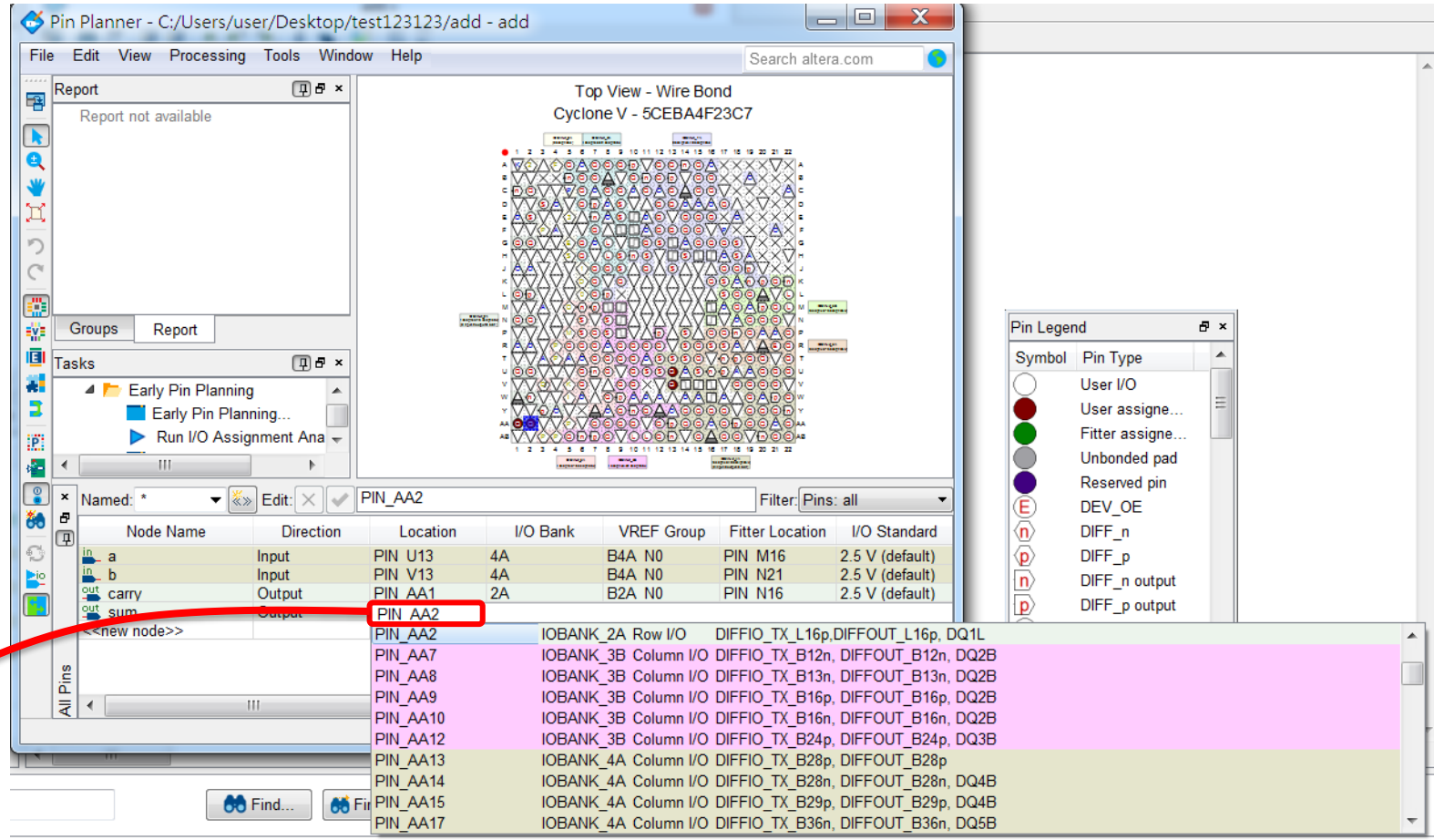
Programming DE0-CV (3/13)

- Open Pin Planner



Programming DE0-CV (4/13)

- Pin assignment



Programming DE0-CV (5/13)

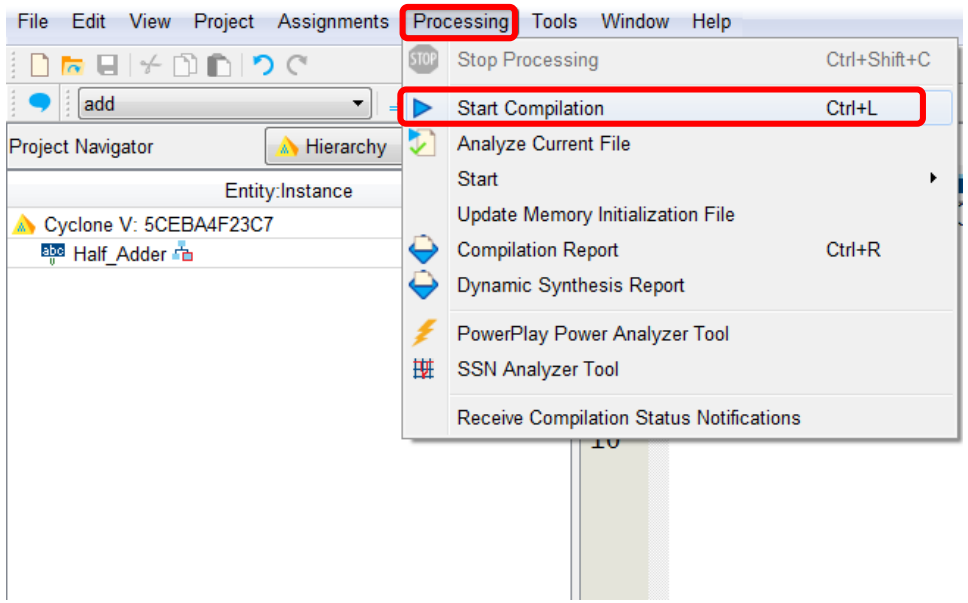
- Assign pin location to all inputs and outputs

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard
in a	Input	PIN U13 SW0	4A	B4A N0	PIN M16	2.5 V (default)
in b	Input	PIN V13 SW1	4A	B4A N0	PIN N21	2.5 V (default)
out carry	Output	PIN AA1 LED1	2A	B2A N0	PIN N16	2.5 V (default)
out sum	Output	PIN AA2 LED0				

  - Please refer to DE0_pin.xls for pin location assignment

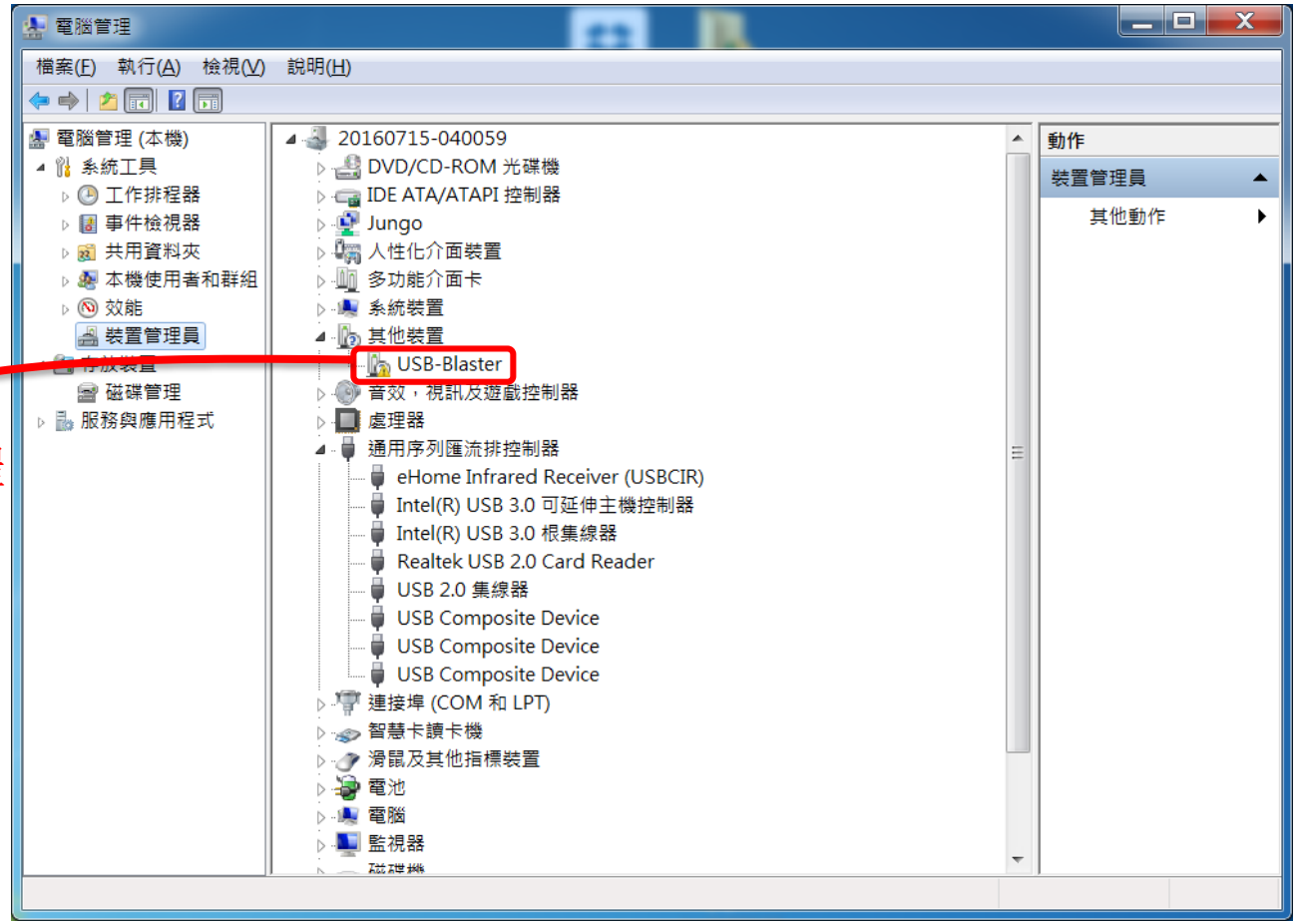
Programming DE0-CV (6/13)

- Start compilation

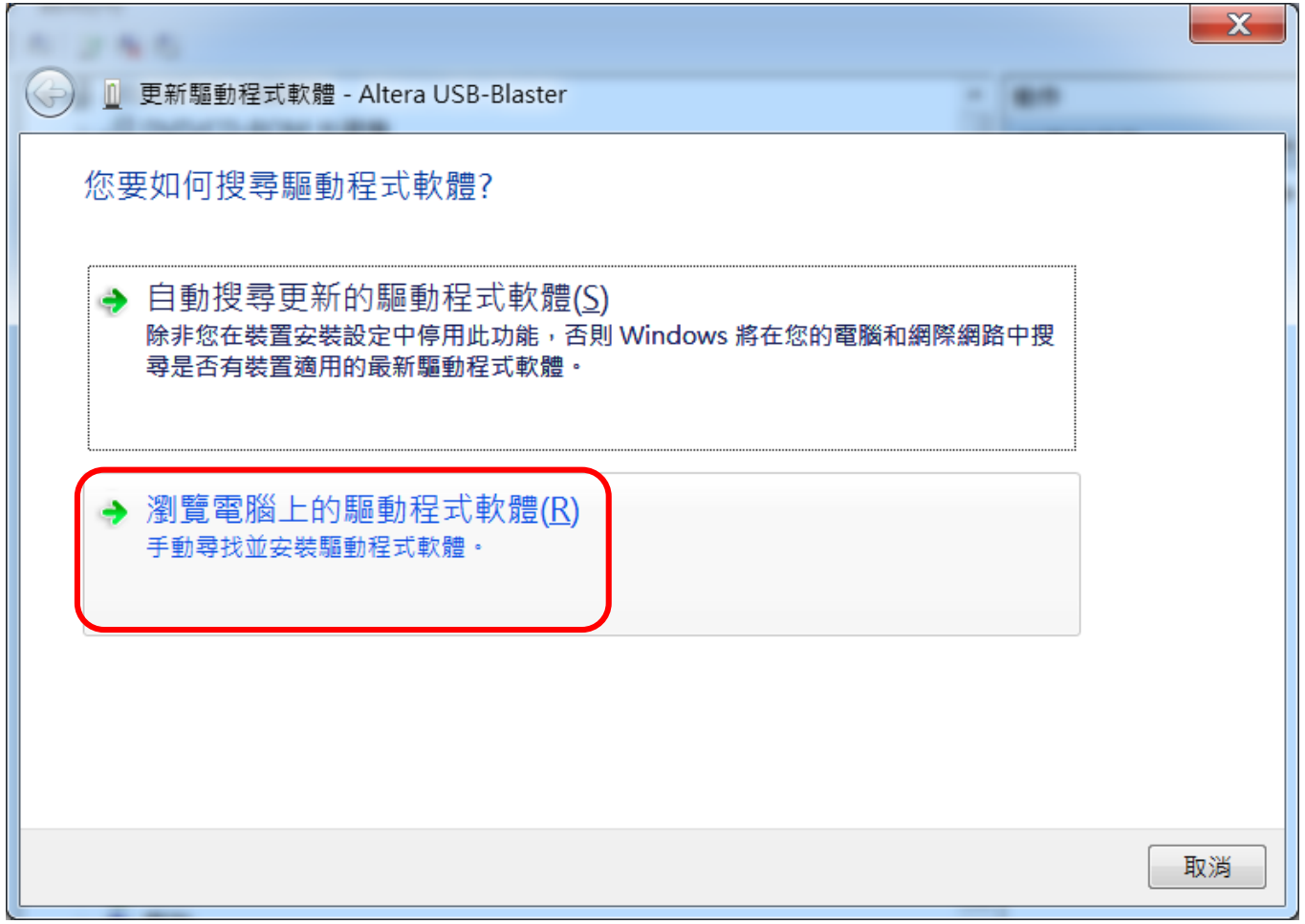


Programming DE0-CV (7/13)

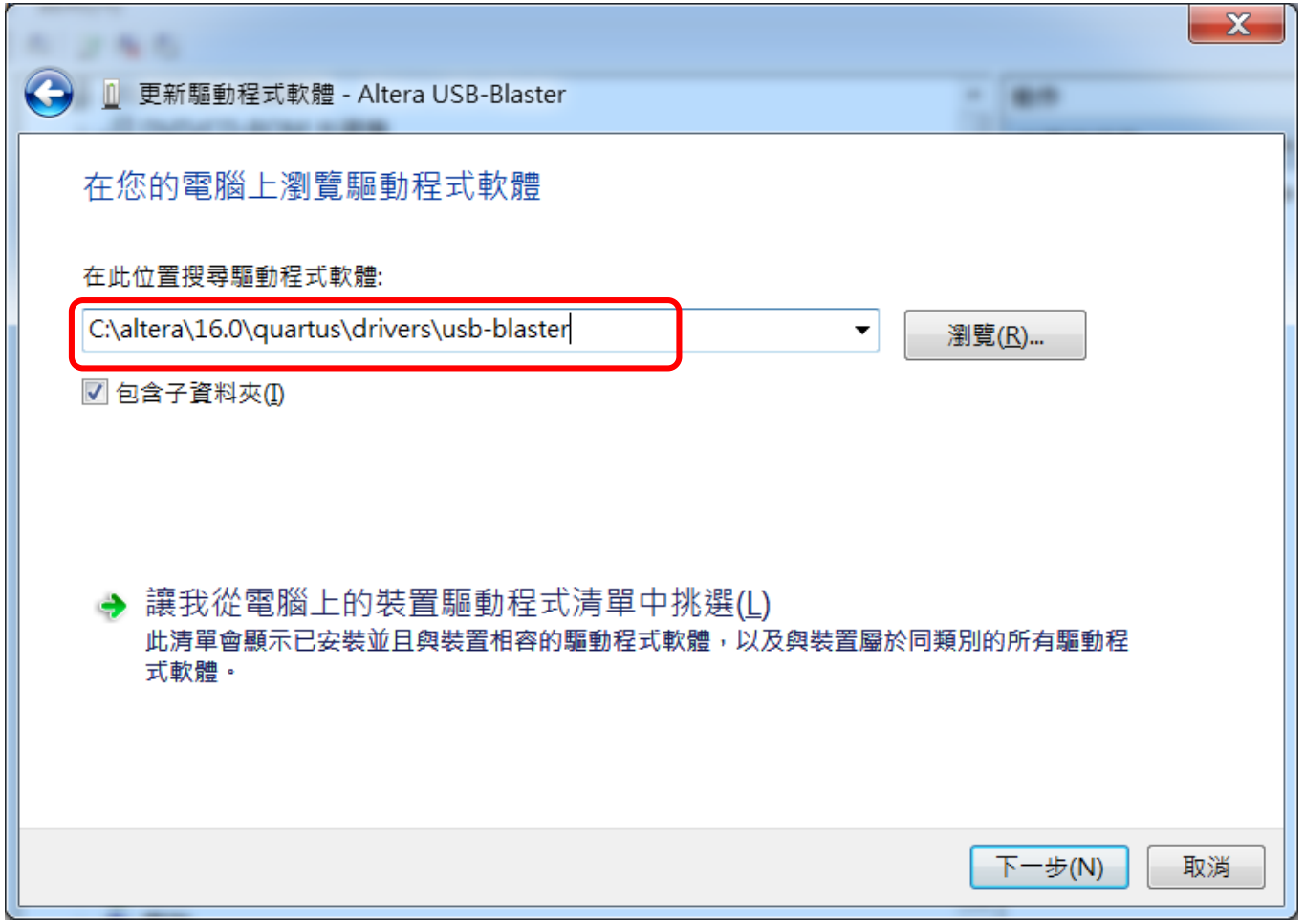
右鍵選更新驅動程式軟體



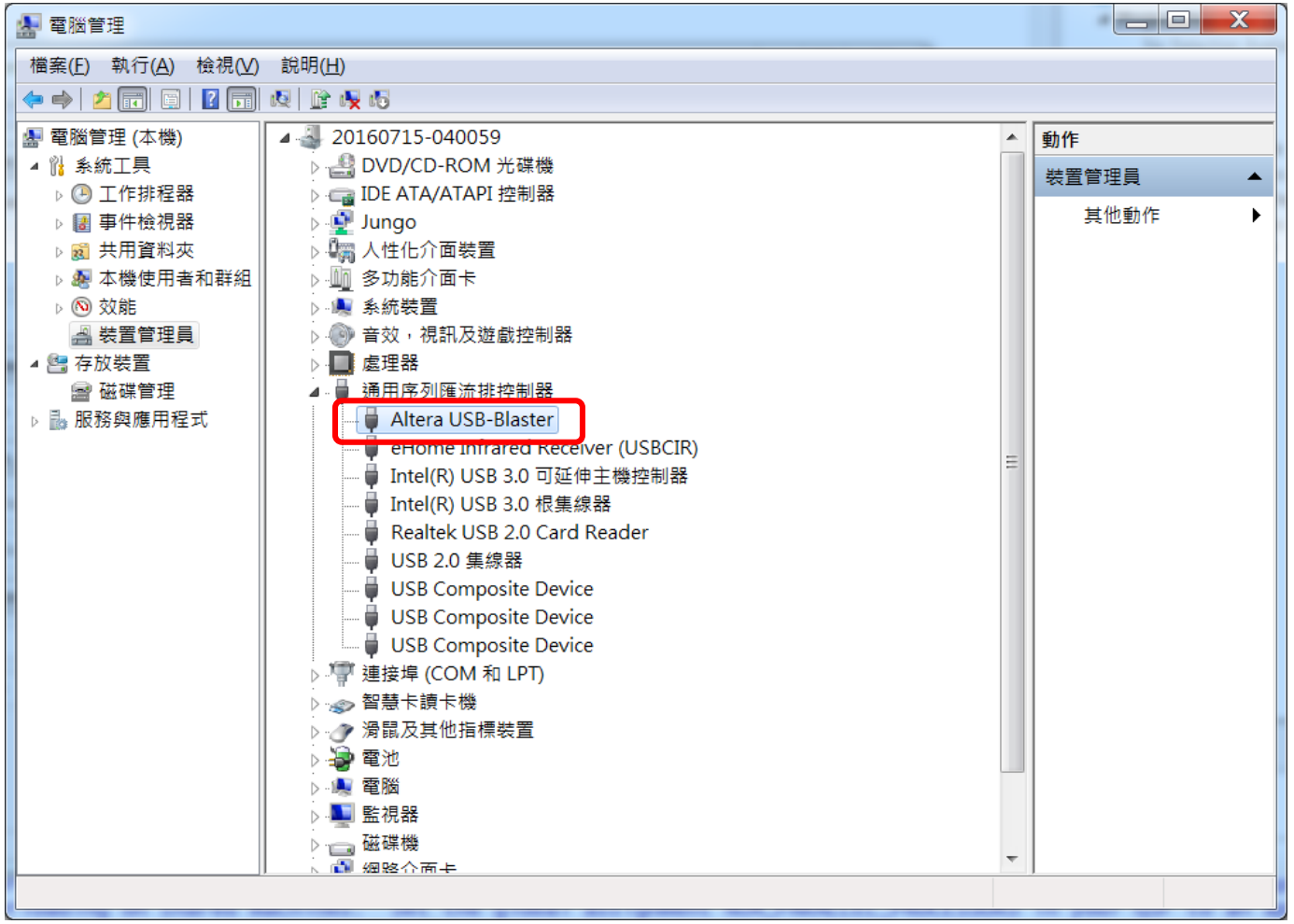
Programming DE0-CV (8/13)



Programming DE0-CV (9/13)

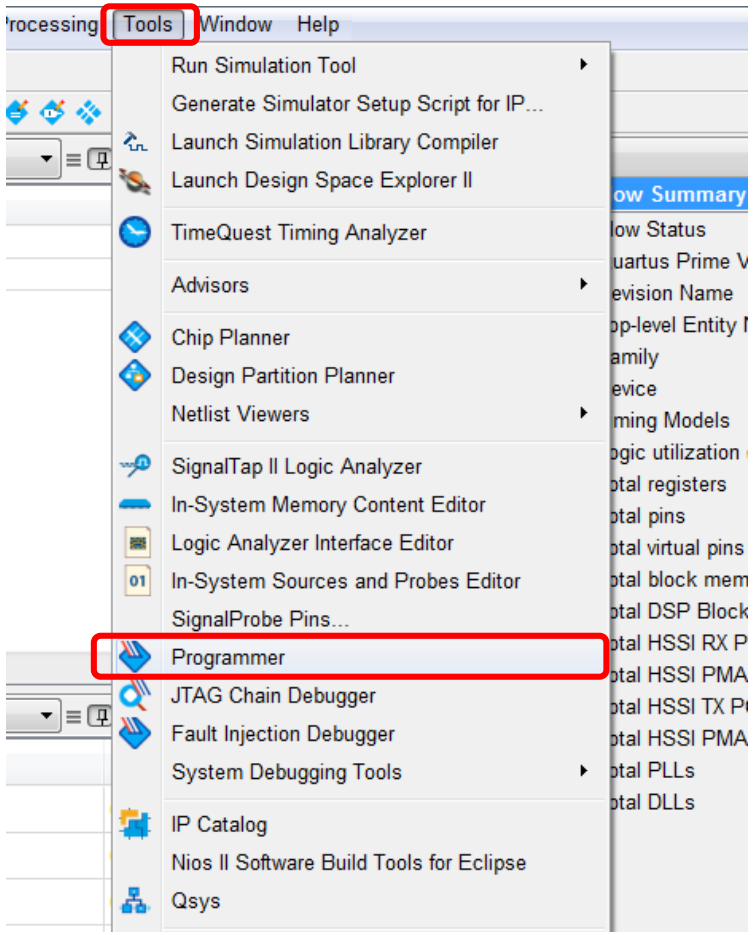


Programming DE0-CV (10/13)



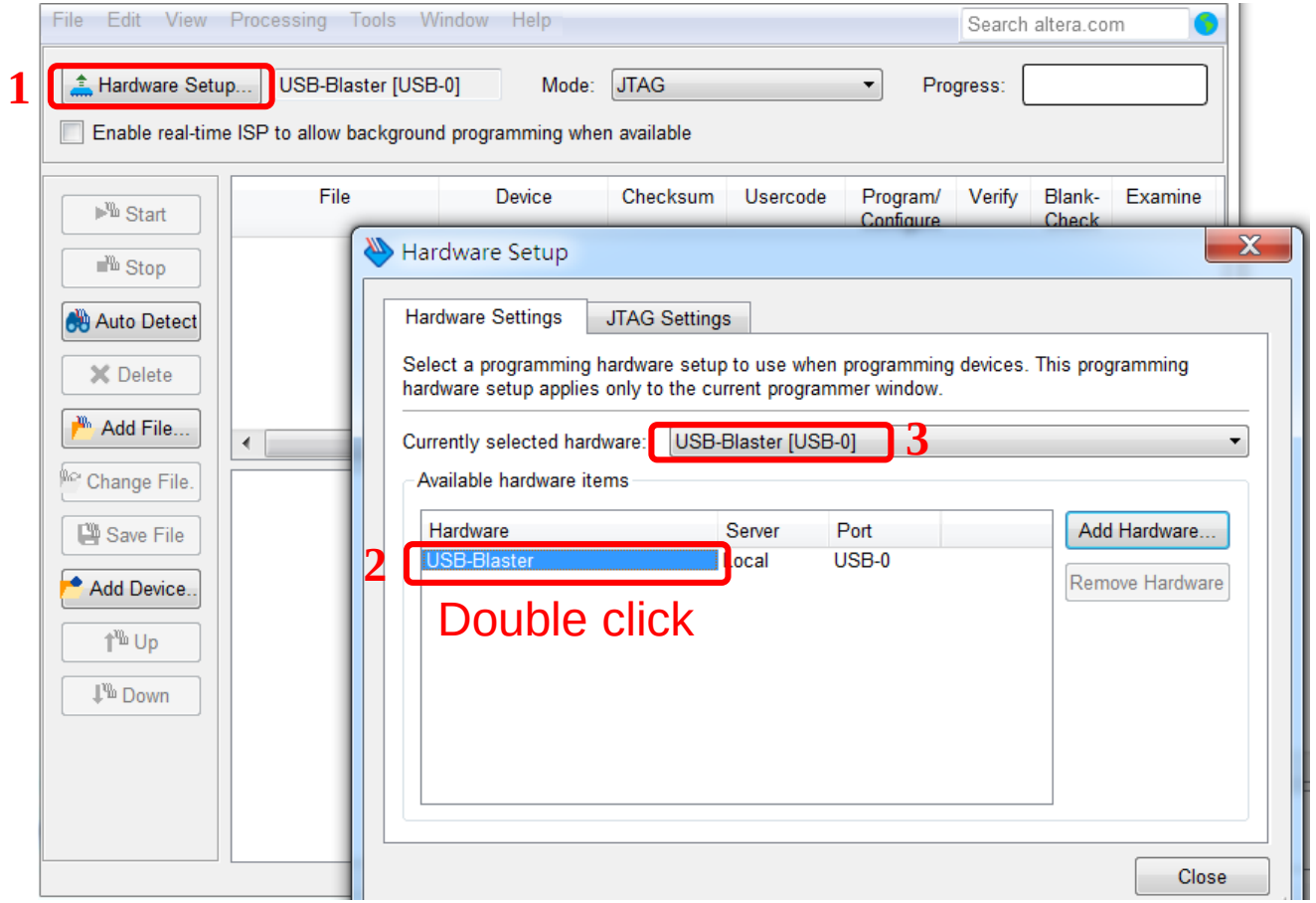
Programming DE0-CV (11/13)

- Programming device



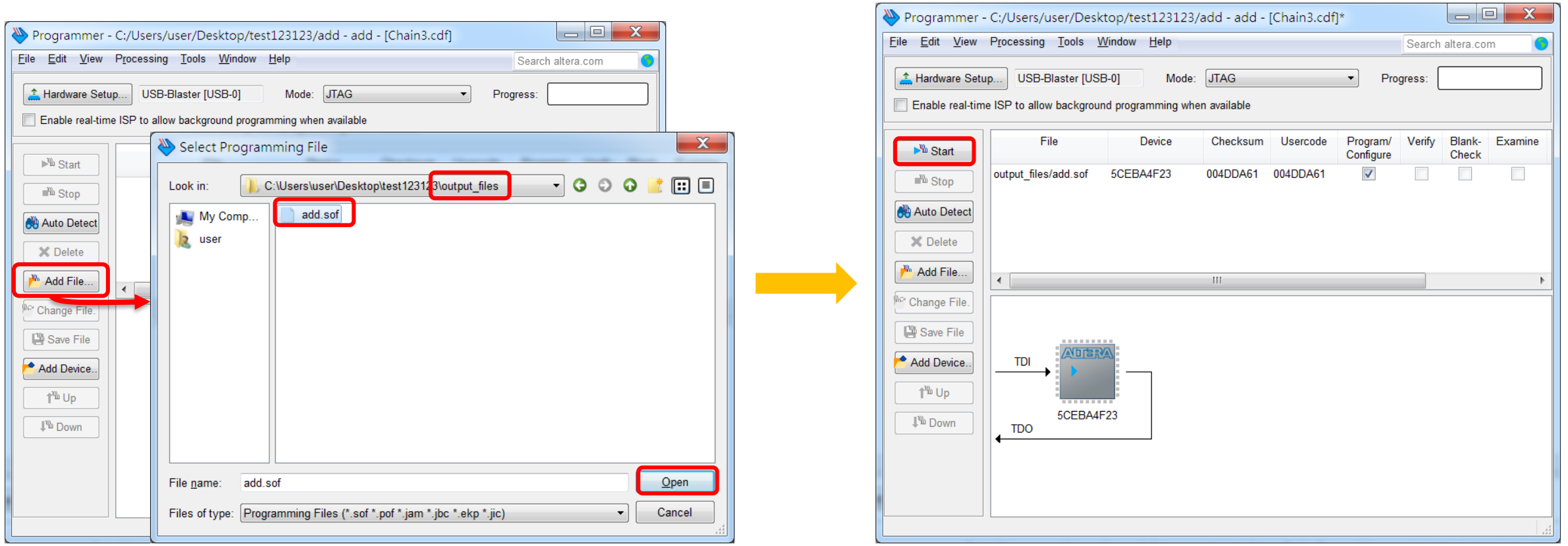
Programming DE0-CV (12/13)

- Hardware setup: add USB-Blaster



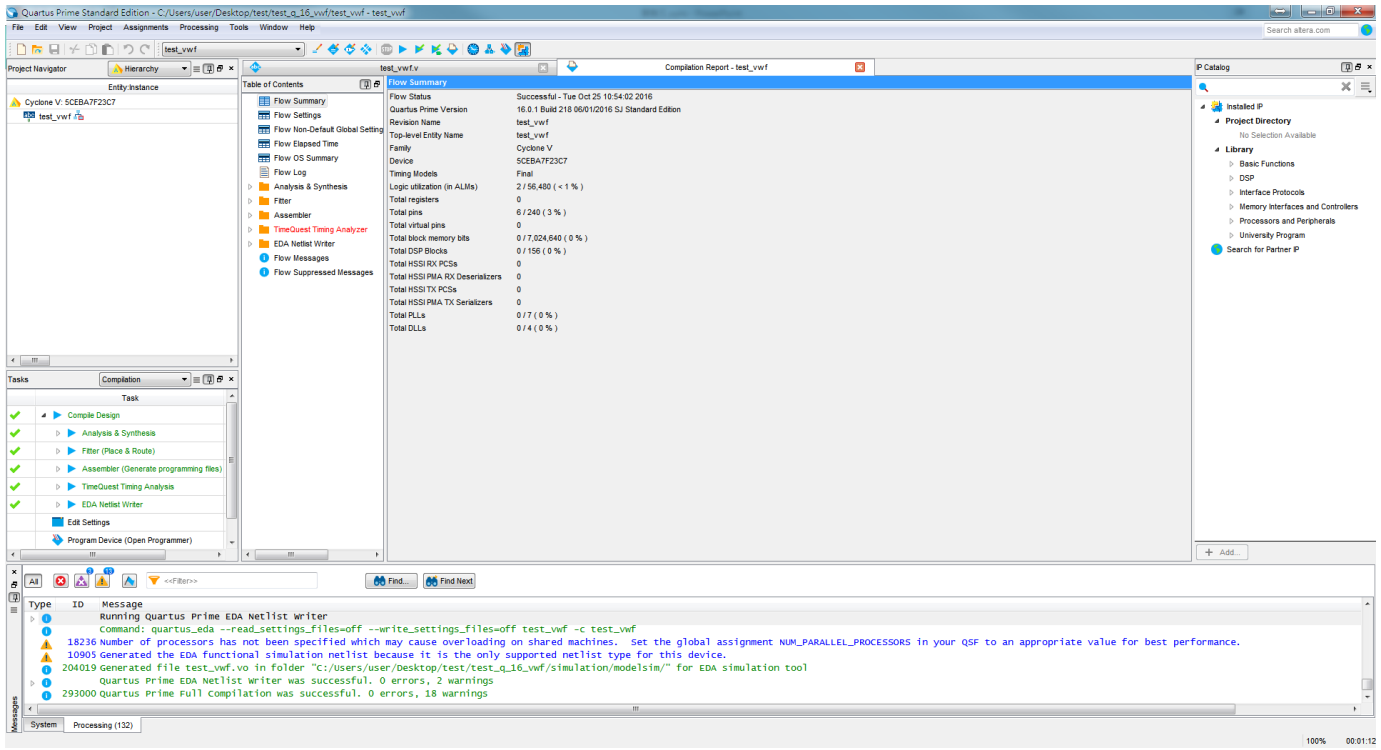
Programming DE0-CV (13/13)

- Programming device



Lab I

- Using Verilog (gate level model) to implement a 1-bit Full Adder (2 half adder + 1 or gate)
- Successfully compile the circuit



Lab I-Hint Structural (gate-level) description

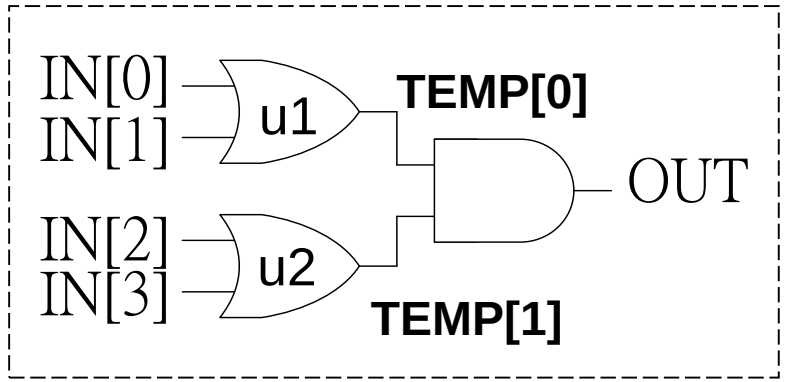
Verilog allows three kinds of descriptions for circuits:

Structural description:

```
1. module OR_AND_STRUCTURAL(IN,OUT);

2. input   [3:0]   IN;
3. output          OUT;
4. wire    [1:0]   TEMP;

5. or u1(TEMP[0], IN[0], IN[1]);
6. or u2(TEMP[1], IN[2], IN[3]);
7. and (OUT, TEMP[0], TEMP[1]);
8. endmodule
```



Synthesized **(synthesis) +**
optimized by tools

Full Adder (1/2)

ab\c_in	0	1
00	0	1
01	1	0
11	0	1
10	1	0

ab\c_in	0	1
00	0	0
01	0	1
11	1	1
10	0	1

sum

$$= \bar{a}\bar{b}c\_in + a\bar{b}\bar{c}\_in + \bar{a}bc\_in + abc\_in$$
$$= (\bar{a}b + a\bar{b})\bar{c}\_in + (\bar{a}\bar{b} + ab)c\_in$$
$$= (a \oplus b)\bar{c}\_in + (\overline{a \oplus b})c\_in$$
$$= (a \oplus b) \oplus c\_in$$

c_out

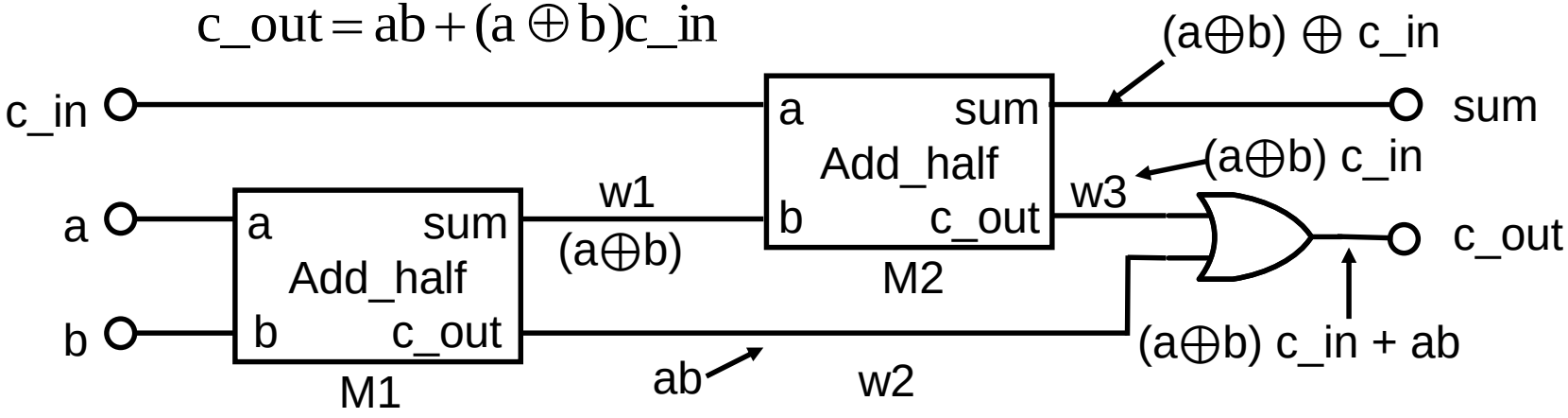
$$= ab + \bar{a}bc\_in + a\bar{b}c\_in$$
$$= ab + (\bar{a}b + a\bar{b})c\_in$$
$$= ab + (a \oplus b)c\_in$$

Full Adder (2/2)

You can implement a full adder with (2 half adder and 1 or gate).

$$sum = (a \oplus b) \oplus c\_in$$

$$c\_out = ab + (a \oplus b)c\_in$$



Lab II

- Use the result of Lab I to implement on the DE0-CV

SW2 SW1 SW0 LED0 LED1

- There are three 1 bit input C_in, X , Y and two output S and C

C_in	X	Y	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

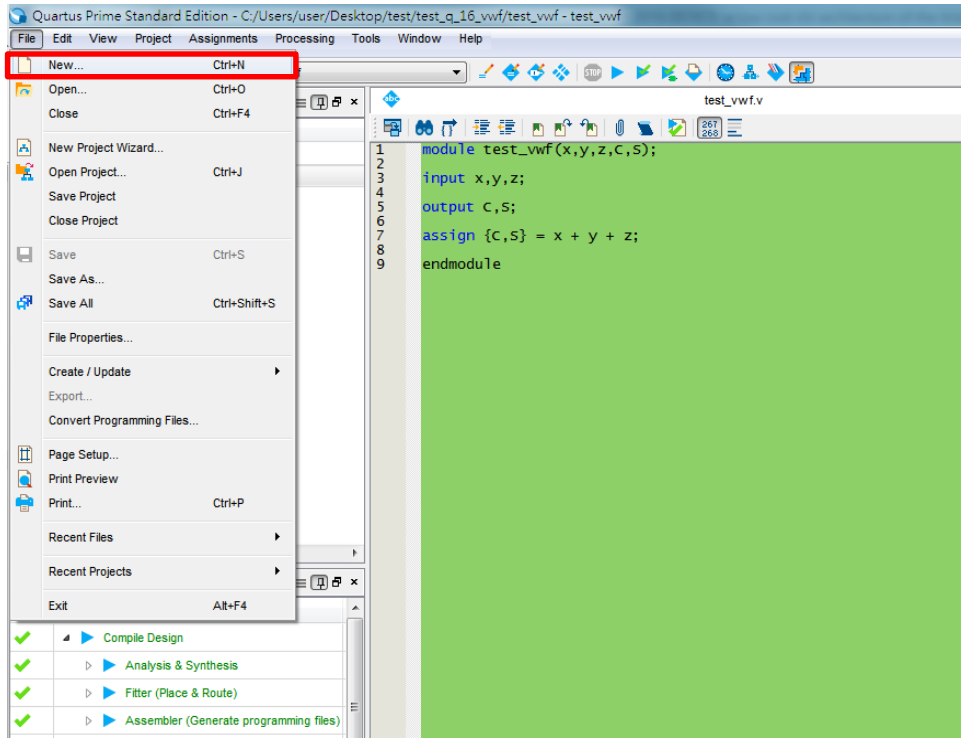
Notice

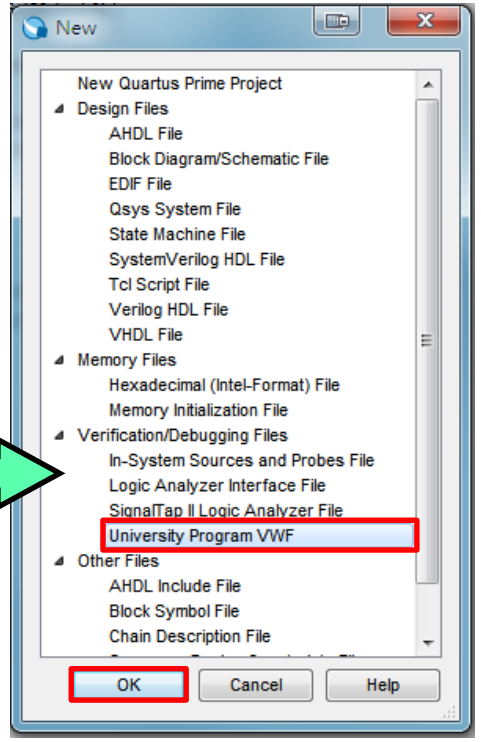
- 請勿命名中文資料夾
- Device family 請確認與 FPGA Chip 符合 (5CEBA4F23C7)
- Top module name & Project name 需要一致
- 確認 module … endmodule 為keyword 變成藍色字體

Appendix

Quartus II Simulation (1/12)

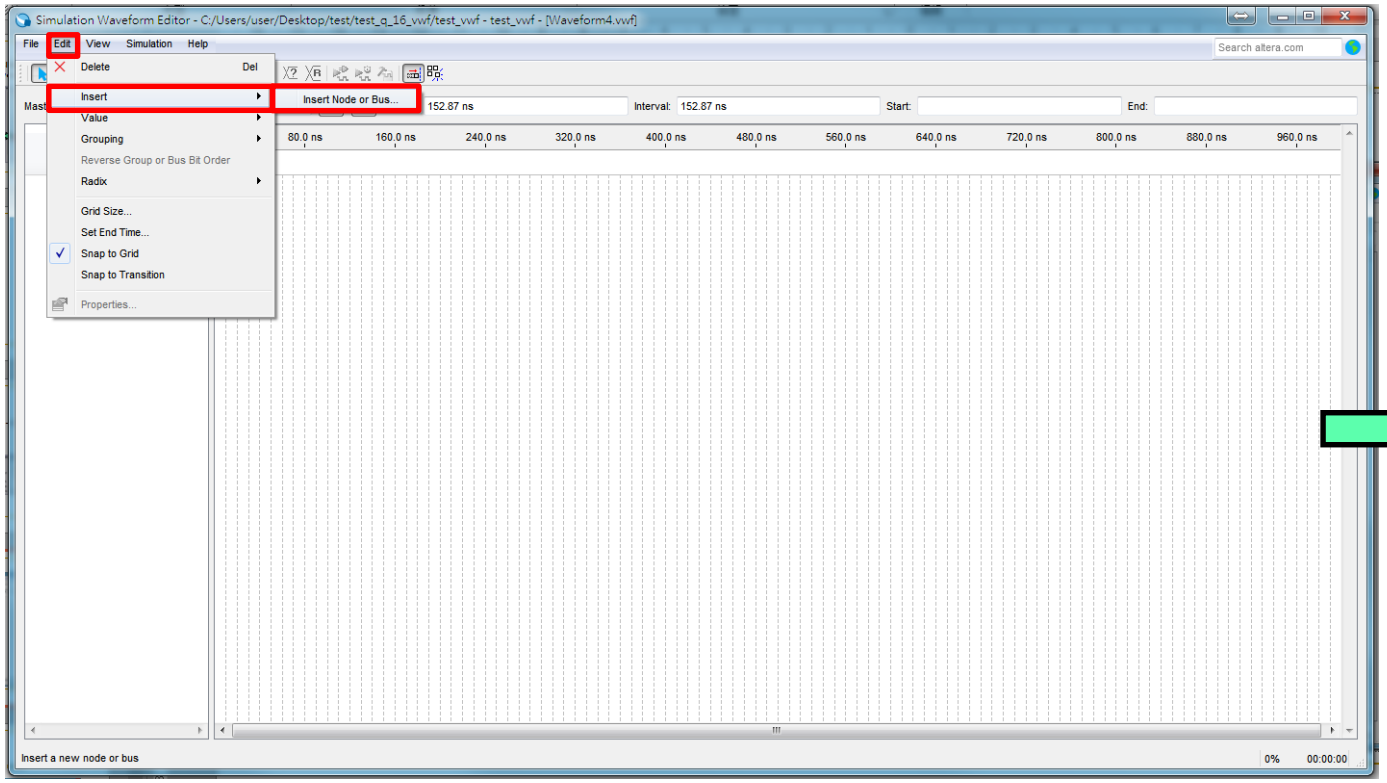
- **Simulating/Verifying the designed Circuit (half adder)**
  - **Using the Waveform Editor(File → New → Vector Waveform File)**

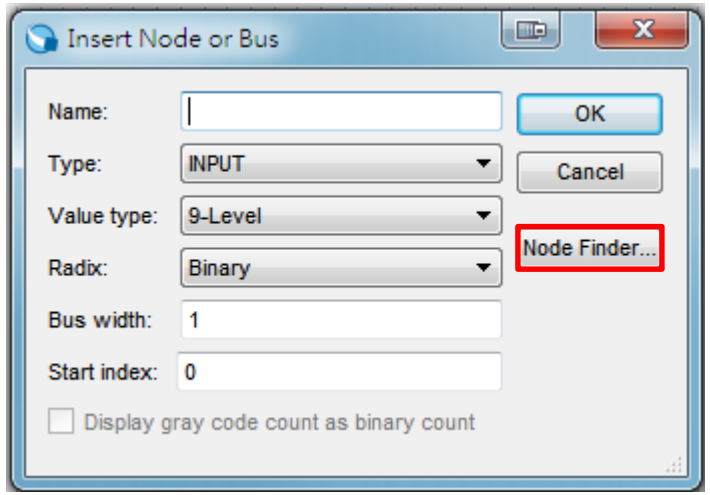




Quartus II Simulation (2/12)

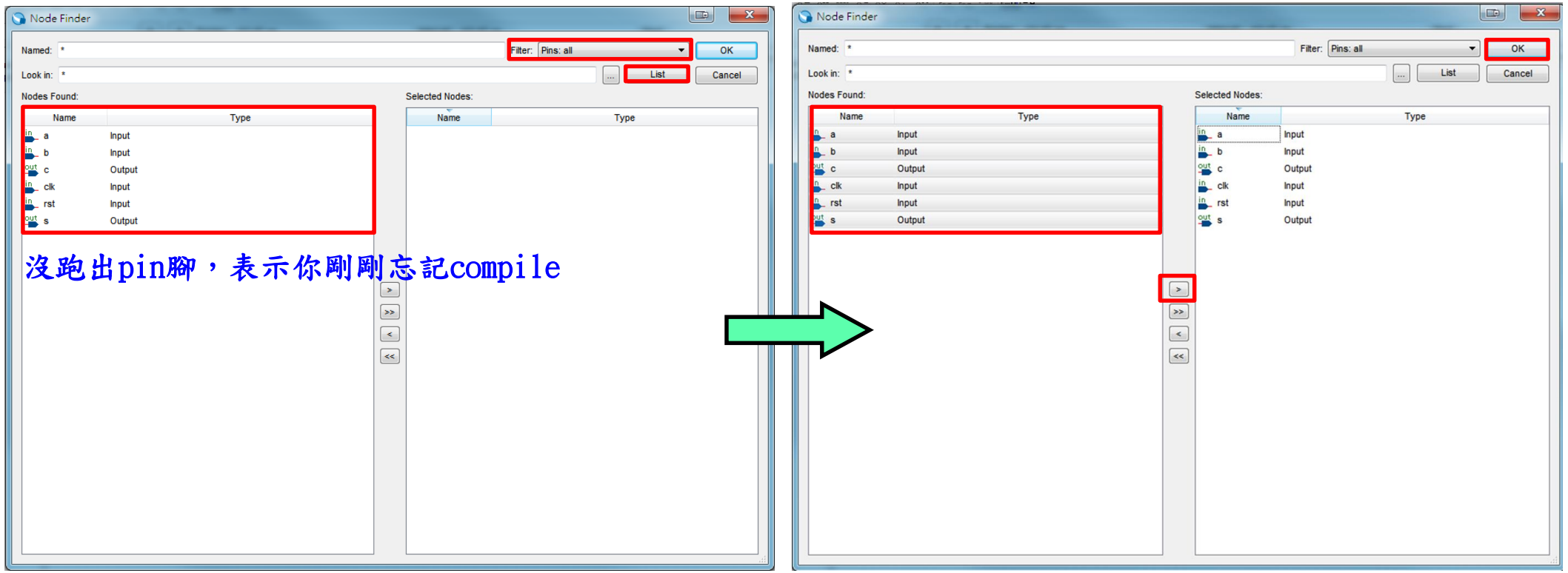
- **Simulating the Designed Circuit**
  - **Use node finder to find all the I/O pins (Edit → Insert → Insert Node or Bus…)**





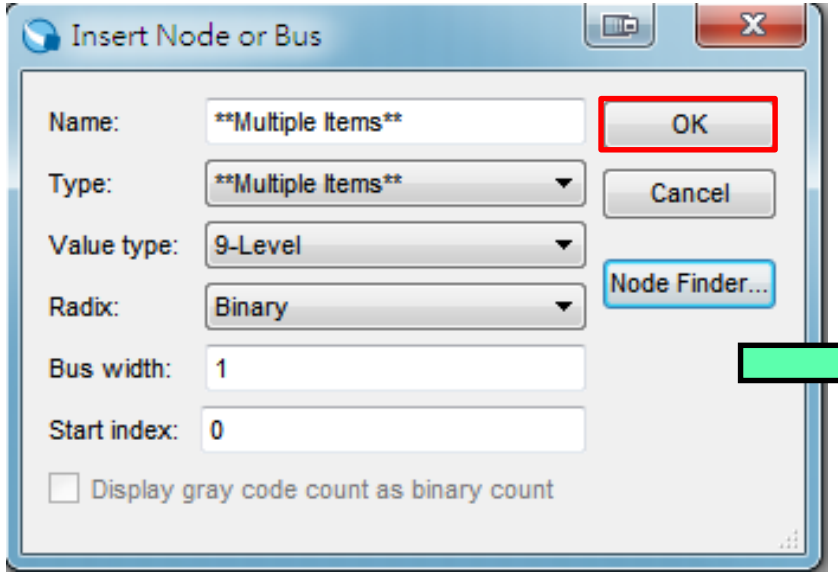
Quartus II Simulation (3/12)

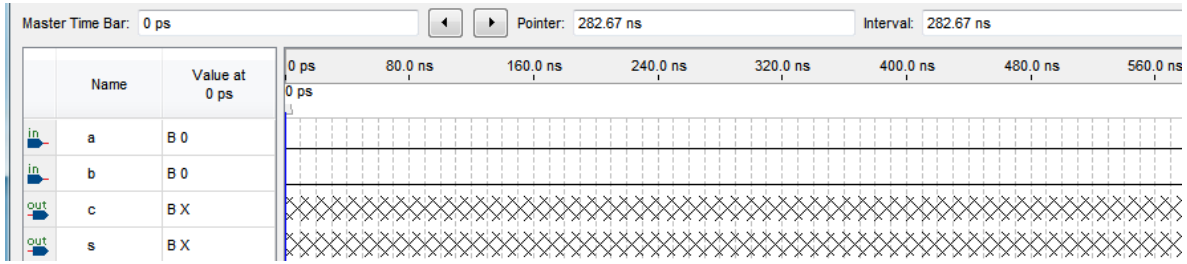
- **Simulating the Designed Circuit**
  - **Selecting nodes to insert into the Waveform Editor**



Quartus II Simulation (4/12)

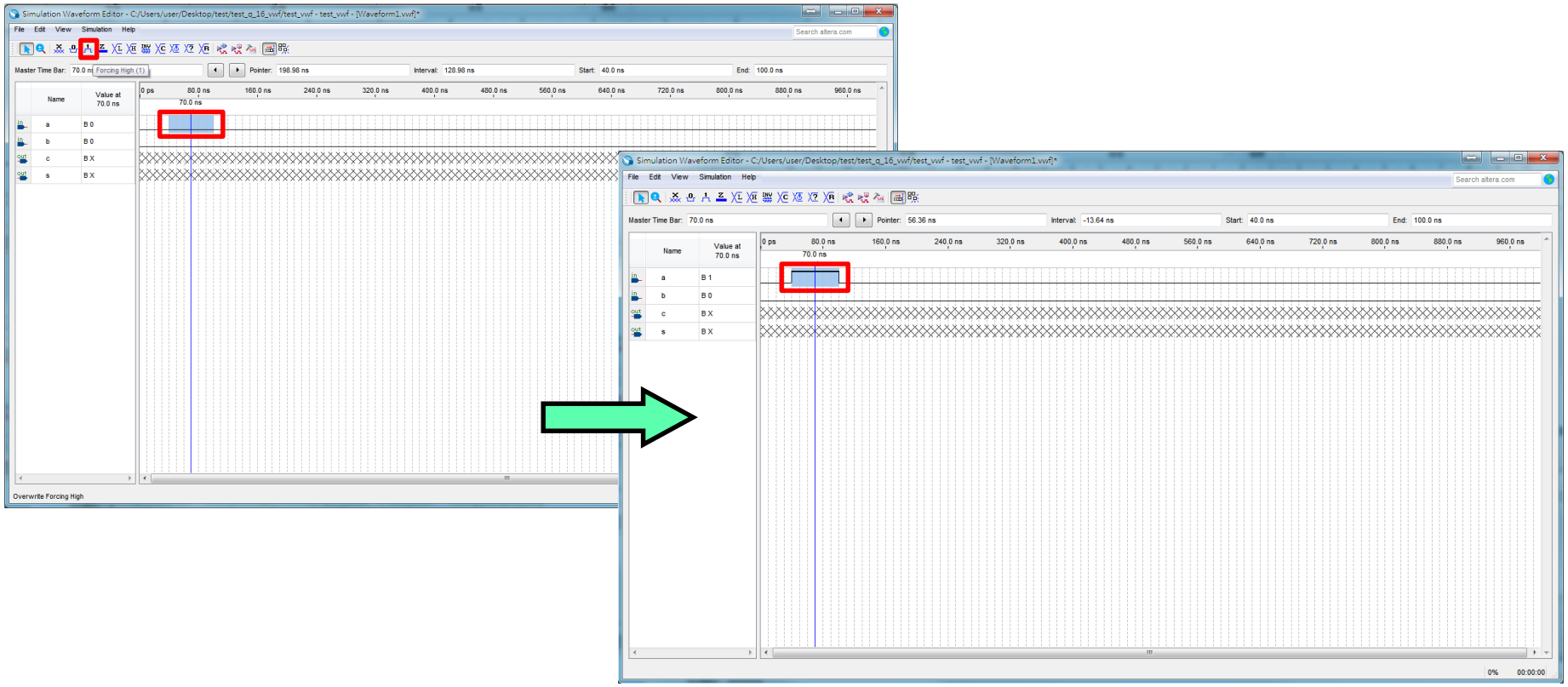
- **Simulating the Designed Circuit**
  - **Selecting nodes to insert into the Waveform Editor**





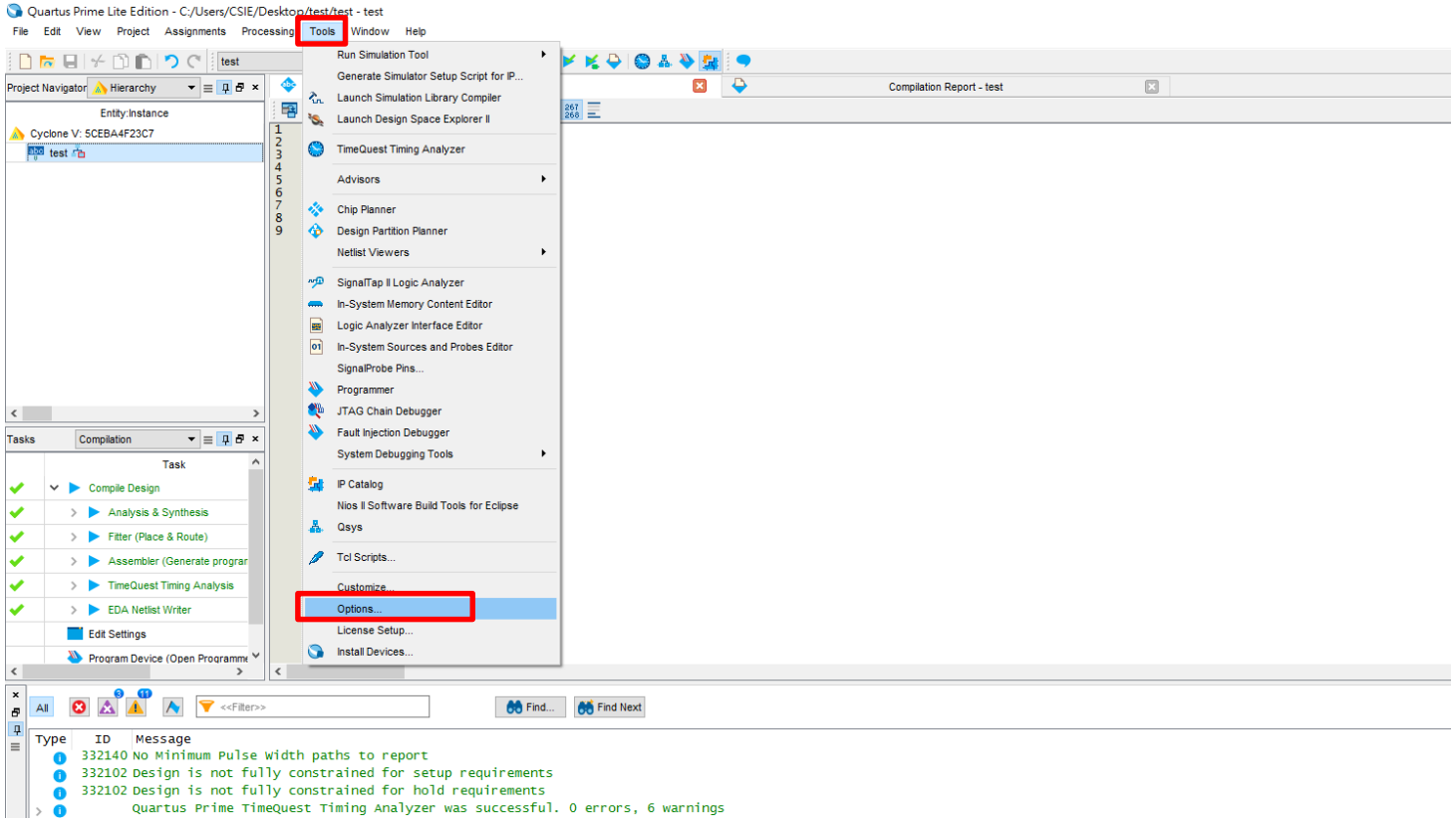
Quartus II Simulation (5/12)

- **Simulating the Designed Circuit**
  - **Select and edit waveform**



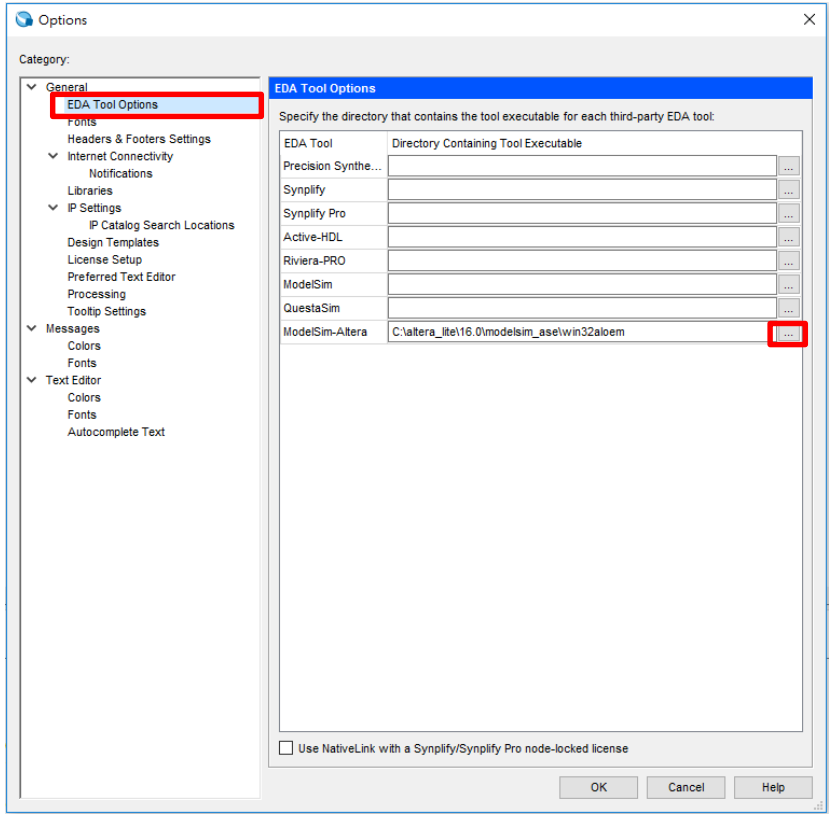
Quartus II Simulation (6/12)

- **Performing the Simulation**
  - **Modify default settings (Tools → Opions →)**



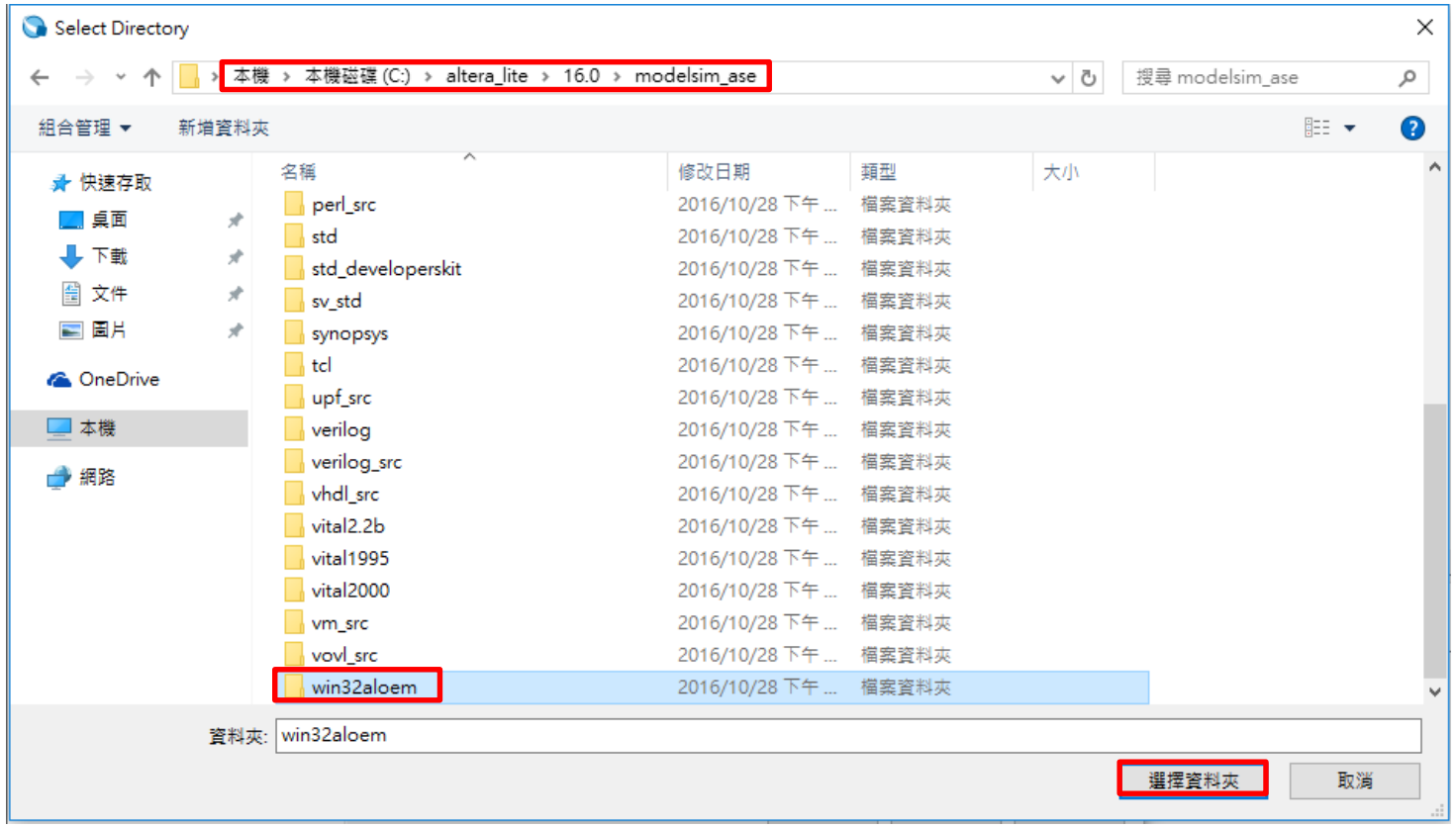
Quartus II Simulation (7/12)

- **Performing the Simulation**
  - **Modify default settings (→EDA Tool Options →ModelSim-Altera →···)**



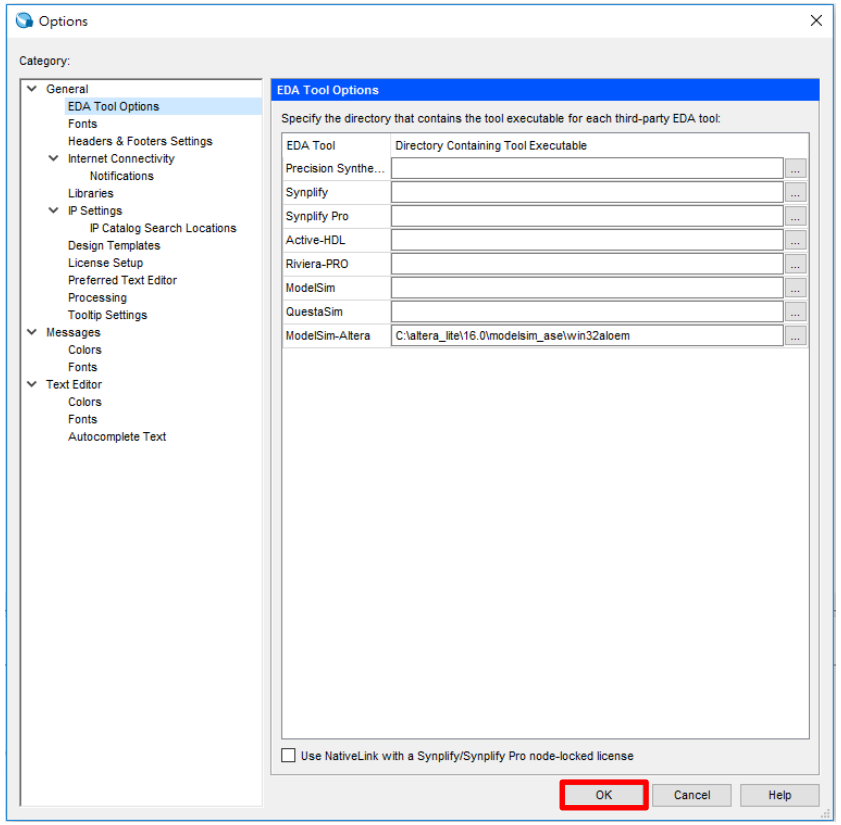
Quartus II Simulation (8/12)

- **Performing the Simulation**
  - **Modify default settings (Select “C:\altera_lite\16.0\modelsim_ase\win32aloem”)**



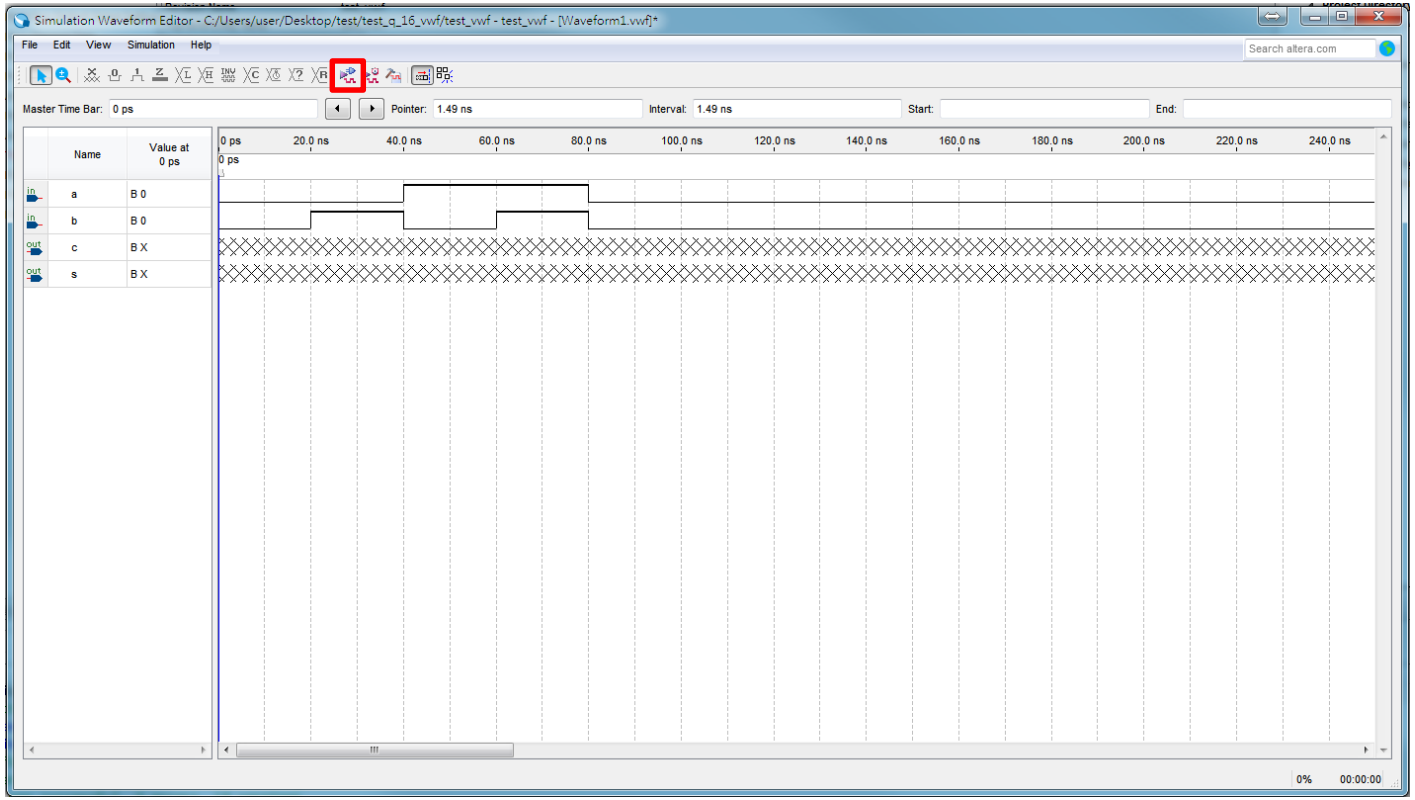
Quartus II Simulation (9/12)

- **Performing the Simulation**
  - **Modify default settings (→ OK)**



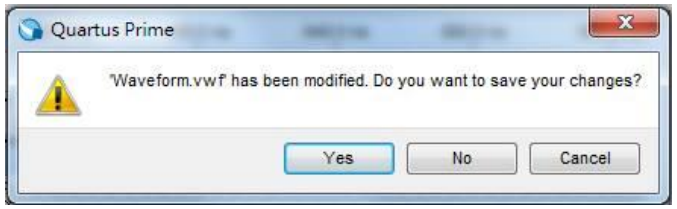
Quartus II Simulation (10/12)

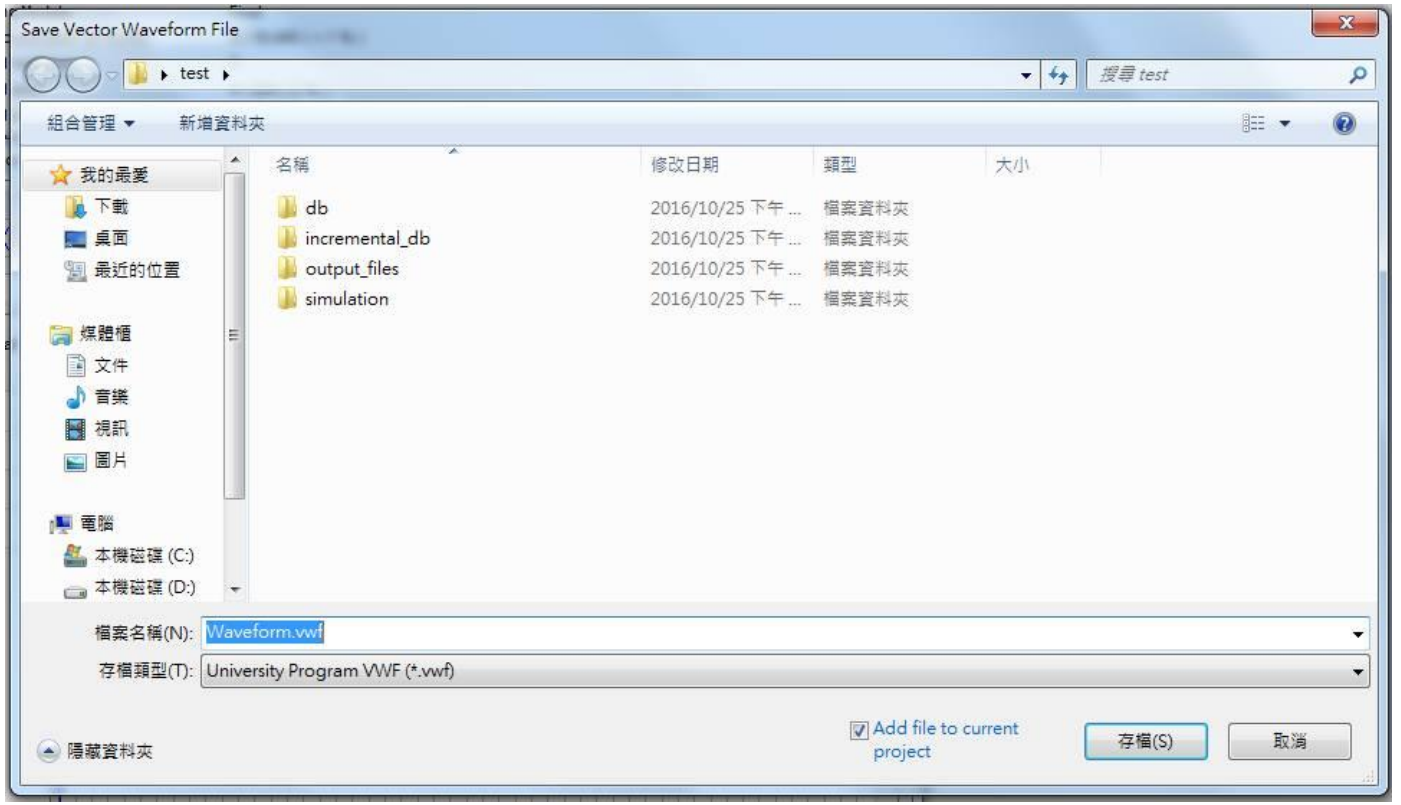
- **Simulating the Designed Circuit**
  - **Setting of test values**
  - **Click “Functional Simulation”**



Quartus II Simulation (11/12)

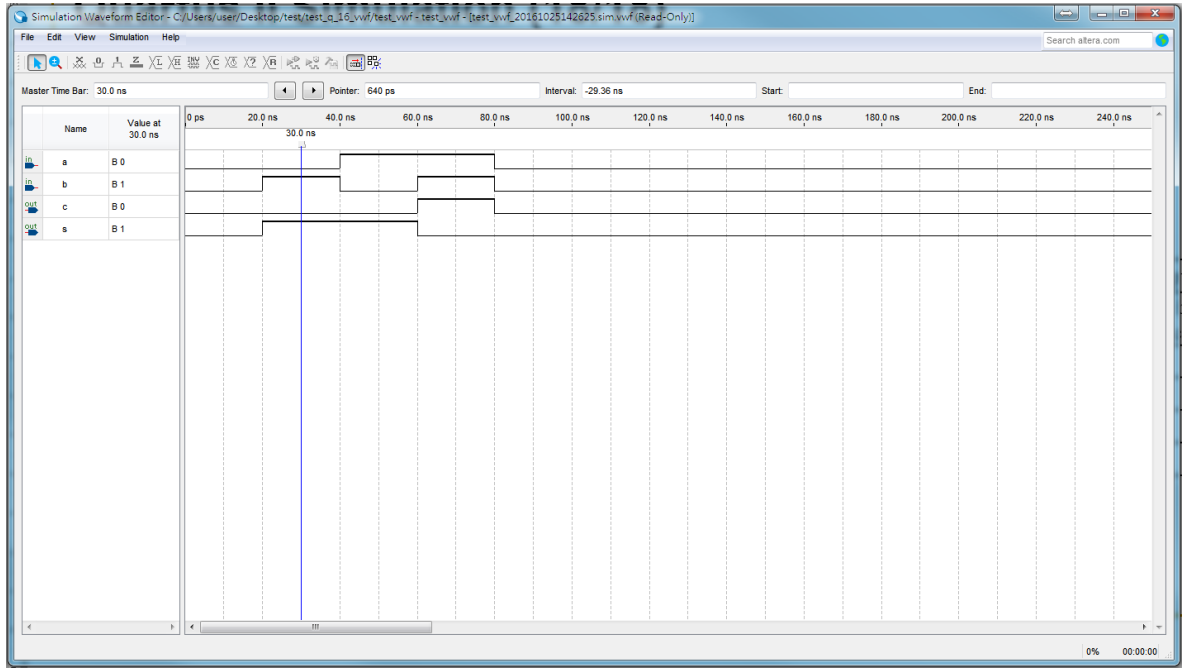
- **Save the Waveform setting.**





Quartus II Simulation (12/12)

- **The result of functional simulation**



輸入(input)		輸出(output)	
被加數(a)	加數(b)	和(sum)	進位(carry)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1