



LAB - 06

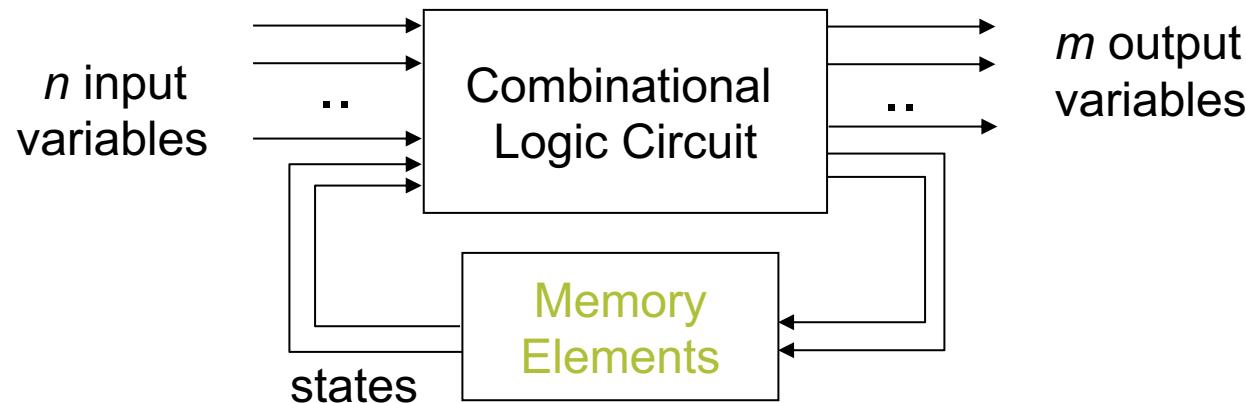
陳培殷
國立成功大學 資訊工程系

Outline

- Video preview for 晶片實現 + HDL介紹 (Parts VI)
 - Sequential Circuit
 - Lab I – Counter (sequential circuit)
-

Sequential Circuit (1/3)

A sequential circuit is a system whose outputs at any time are determined *from the present combination of inputs and the previous inputs or outputs.*



- Sequential components contain memory elements
- The output values of sequential components depend on the input values and the values stored in the memory elements

Sequential Circuit (2/3)

```
1 module sequential_circuit(clk, reset);
2   input clk, reset;
3
4   always@(posedge clk or negedge reset)
5   begin
6
7     if(!reset)
8       begin
9         // Initialization
10      end
11    else
12      begin
13        // Circuit functionality
14      end
15
16  end
17
18 endmodule
19
```

正緣同步電路

低位準非同步重置

Sequential Circuit (3/3)

■ Blocking statement vs nonblocking statement

```
27 module combinational_circuit(in, a, b);
28
29 input in;
30 output reg [3:0] a,b;
31
32 input in;
33 output reg [3:0] a,b;
34
35 always@(*)
36 begin
37
38 if(in == 0)
39 begin
40     a = 4'h1;
41     b = 4'hf;
42
43     a = b; // a -> 4'hf
44     b = a; // b -> 4'hf
45 end
46 else
47 begin
48     a = 4'h1;
49     b = 4'hf;
50
51     b = a; // b -> 4'h1
52     a = b; // a -> 4'h1
53 end
54
55 end
56
57 endmodule
```

```
1  module sequential_circuit(clk, reset);
2  input clk, reset;
3
4  reg [3:0] a,b;
5
6  always@(posedge clk or negedge reset)
7  begin
8
9      if(!reset)
10 begin
11         a <= 4'h1;
12         b <= 4'hf;
13     end
14     else
15 begin
16         a <= b; // a -> 4'hf
17         b <= a; // b -> 4'h1
18     end
19
20 end
21
22 endmodule
```

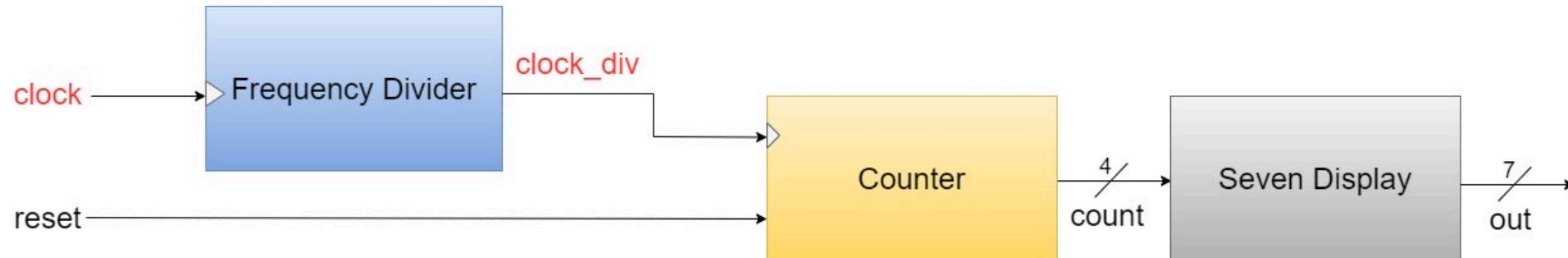
Lab I (1/3)

■ 請設計一個具備下列功能的計數器：

- 正緣同步電路，低位準非同步重置
- 當reset訊號為0，將目前狀態暫停，輸出維持0
- 當reset訊號為1，**每秒計數加1**
 - EX : 0 -> 1 -> 2 -> -> E -> F -> 0 ->
- FPGA版之clock頻率為50MHz，需藉由除頻器將clock訊號降為1Hz
 - 實現方式為透過一個計數器，計算經過幾個**時脈正緣**，當計數到 50×10^6 即代表經過一秒

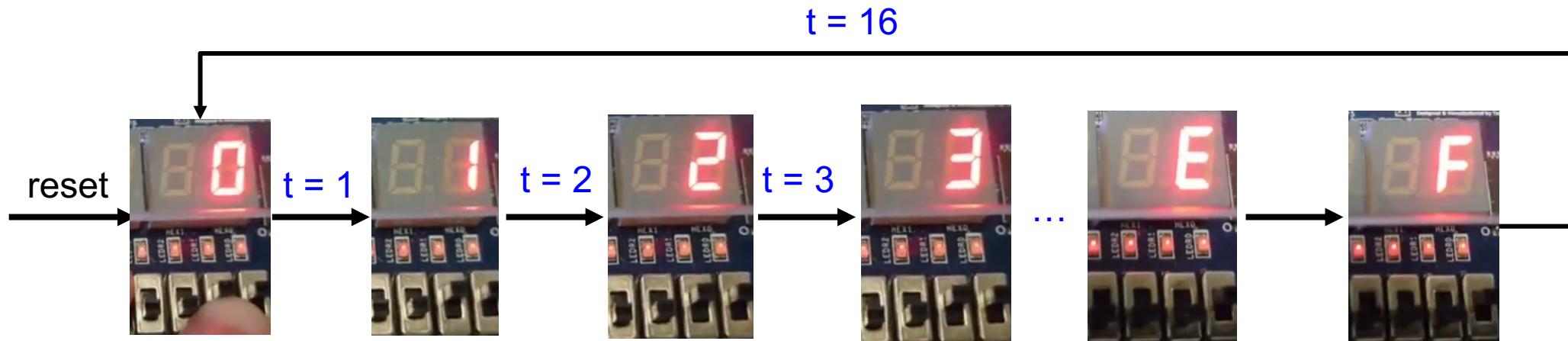
Lab I (2/3)

- 請將計數的數值顯示於七段顯示器
- 系統架構圖請參考下方
 - Input : **clock(CLOCK_50)**, reset(SW0)
 - Output : out(7 bits, HEX06~HEX00)
- 可使用 **structural description** 設計，其中，除頻器及計數模組為循序電路，七段顯示器控制模組則為組合電路



Lab I (3/3)

■ 輸出範例



Lab - Hint(1/2)

■ Frequency Divider (sequential circuit)

- 將clock頻率從50MHz降為1Hz

■ Counter (sequential circuit)

- 每秒進行計數

■ Seven Display (combinational circuit)

- 將Counter數值轉為七段顯示器控制訊號

■ Module呼叫範例 (Structural description)

- **FrequencyDivider** u_FreqDiv (.clk(clock), .reset(reset), .clk_div(clock_div));
- **Counter** u_counter(.clk(clock_div), .reset(reset), .count(count));
- **SevenDisplay** u_display(.count(count), .out(out));

.port_name(signal_name)

module_name unit_name

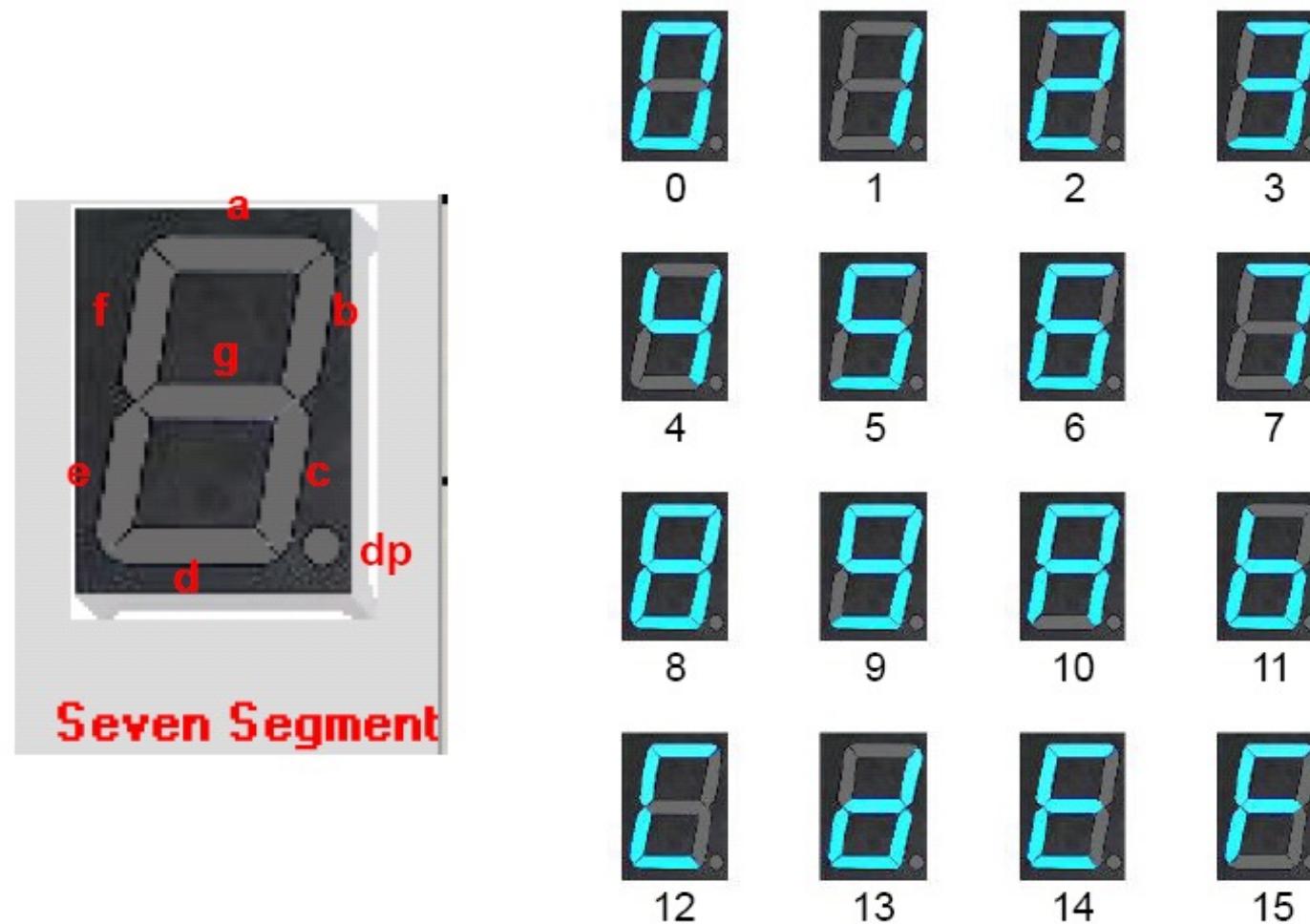
Lab – Hint(2/2)

■ 除頻器範例：

- 每0.5秒改變一次訊號(div_clk)

```
1 `define TimeExpire 32'd25000000
2
3 module clk_div(clk,rst,div_clk);
4   input clk,rst;
5   output div_clk;
6
7   reg div_clk;
8   reg [31:0]count;
9
10  always@ (posedge clk)
11    begin
12      if(!rst) 低位準同步reset
13        begin
14          count <= 32'd0;
15          div_clk <= 1'b0;
16        end
17      else
18        begin
19          if(count == `TimeExpire)
20            begin
21              count <= 32'd0;
22              div_clk <= ~div_clk;
23            end
24          else
25            begin
26              count <= count + 32'd1;
27            end
28        end
29    end
30
31 endmodule
```

Seven-segment display(1/2)



Seven-segment display (2/2)

- 0 is on, 1 is off
- dp is useless in DE0-CV board

- Ex:  out=7'b1000000;
g=1

- Ex:  out=7'b0010010;
b=1, e=1

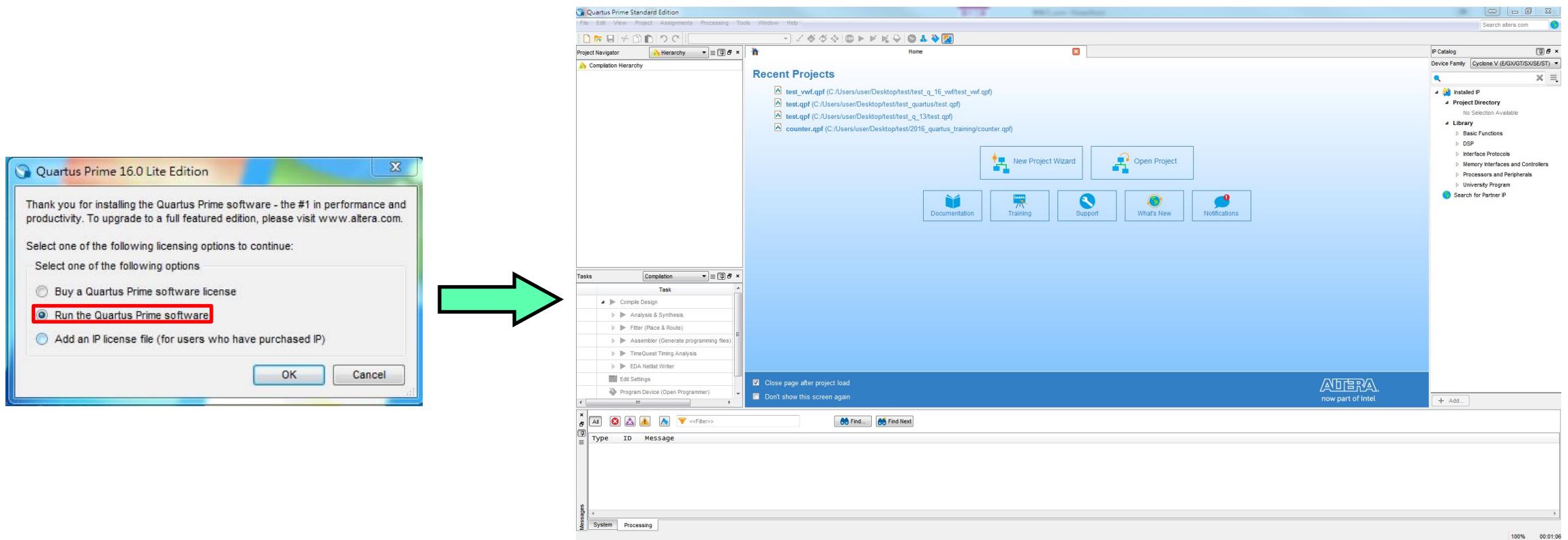
Notice

- 請勿命名中文或數字開頭的資料夾
- Device family 請確認與 FPGA Chip 符合 (**5CEFA4F23C7**)
- Top module name & Project name 需要一致

Quartus II Tutorial (1/10)

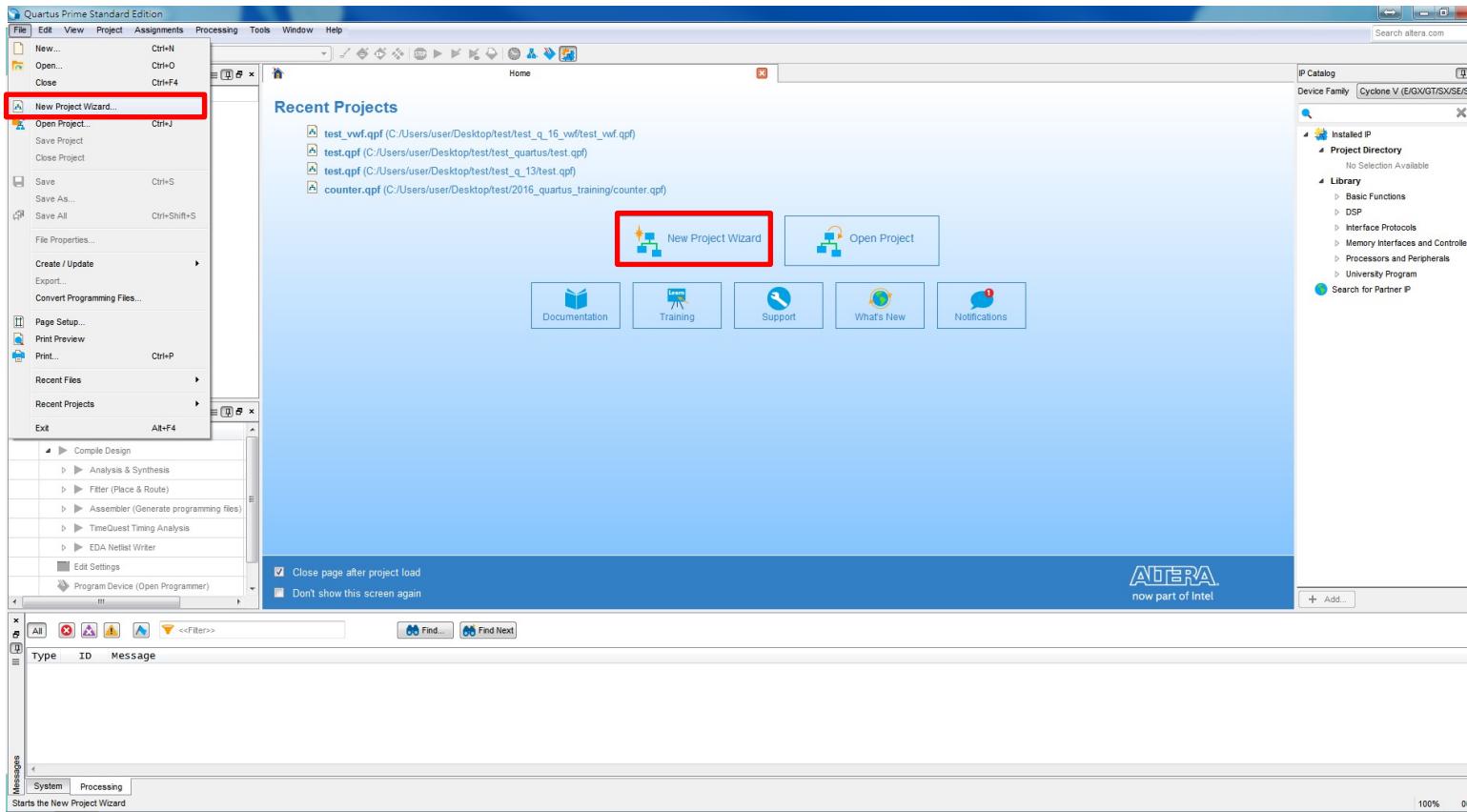
■ Getting Started –

- Start the Quartus II software



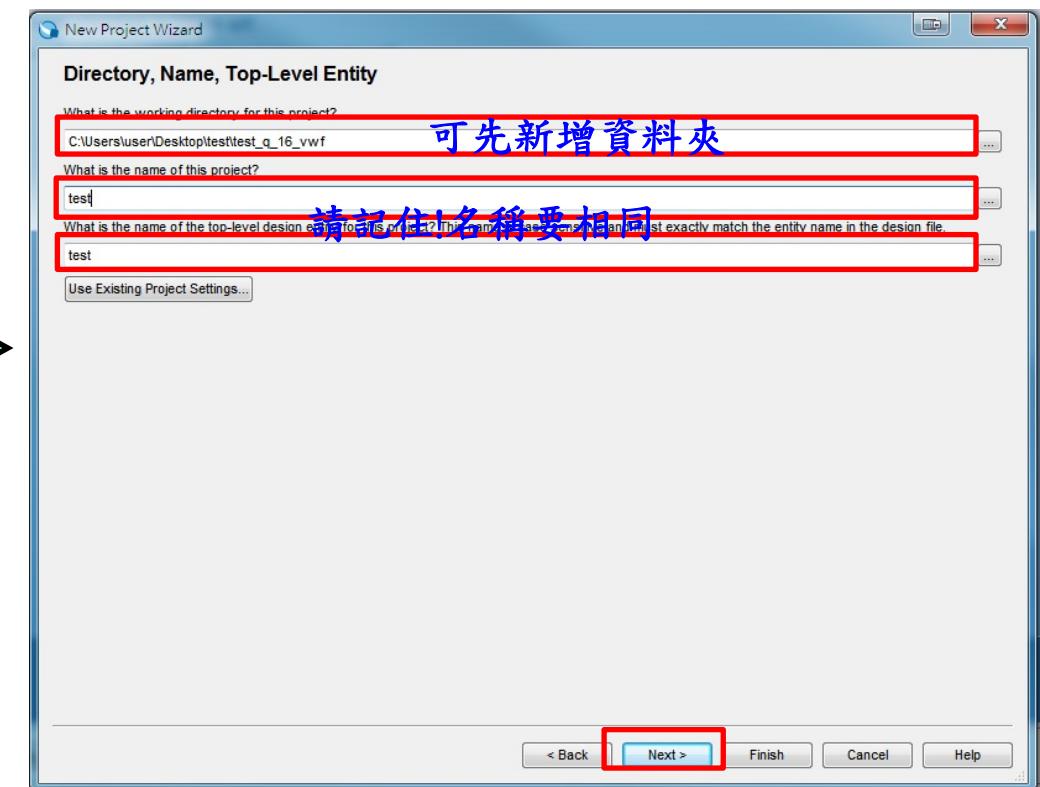
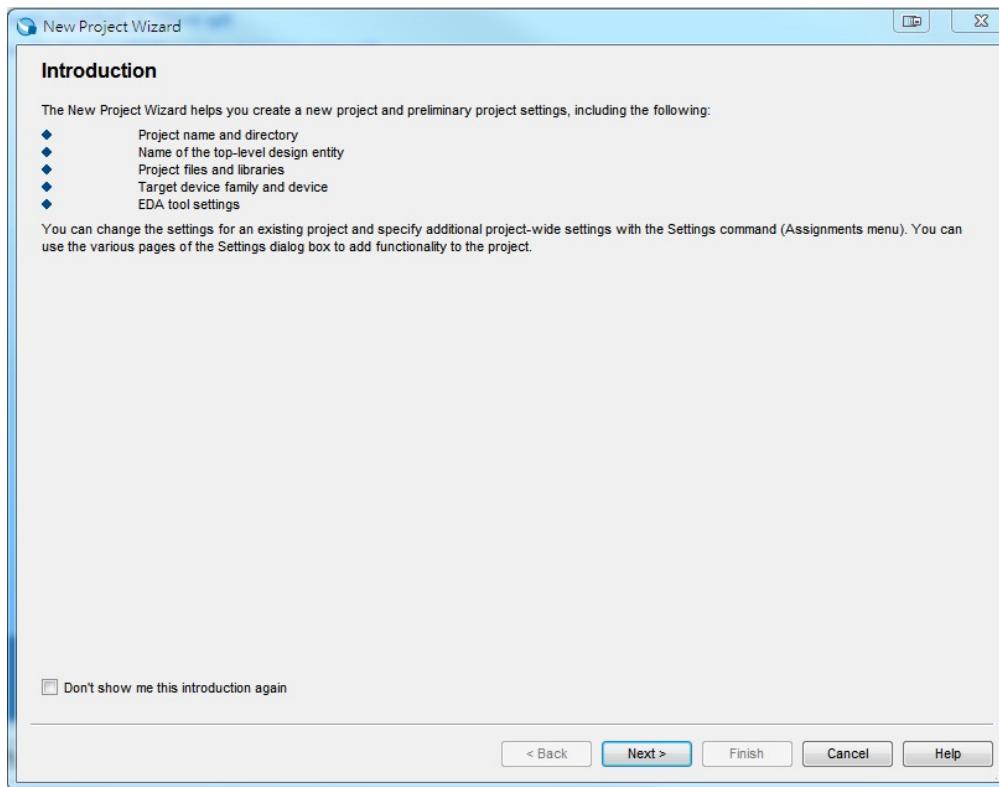
Quartus II Tutorial (2/10)

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



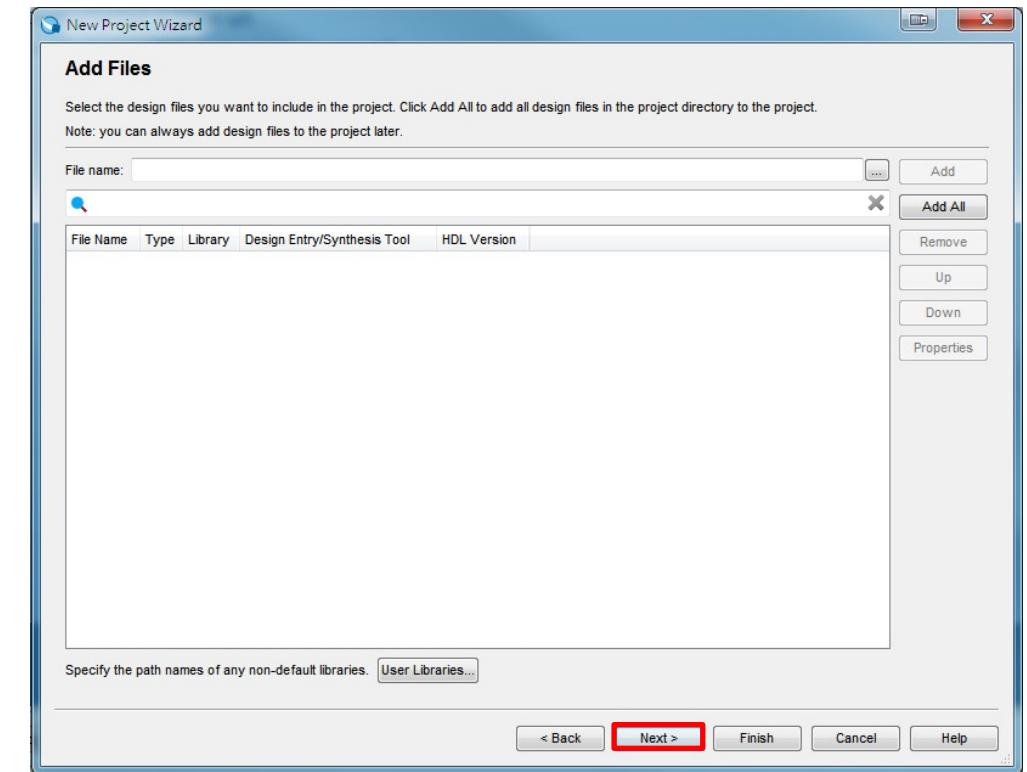
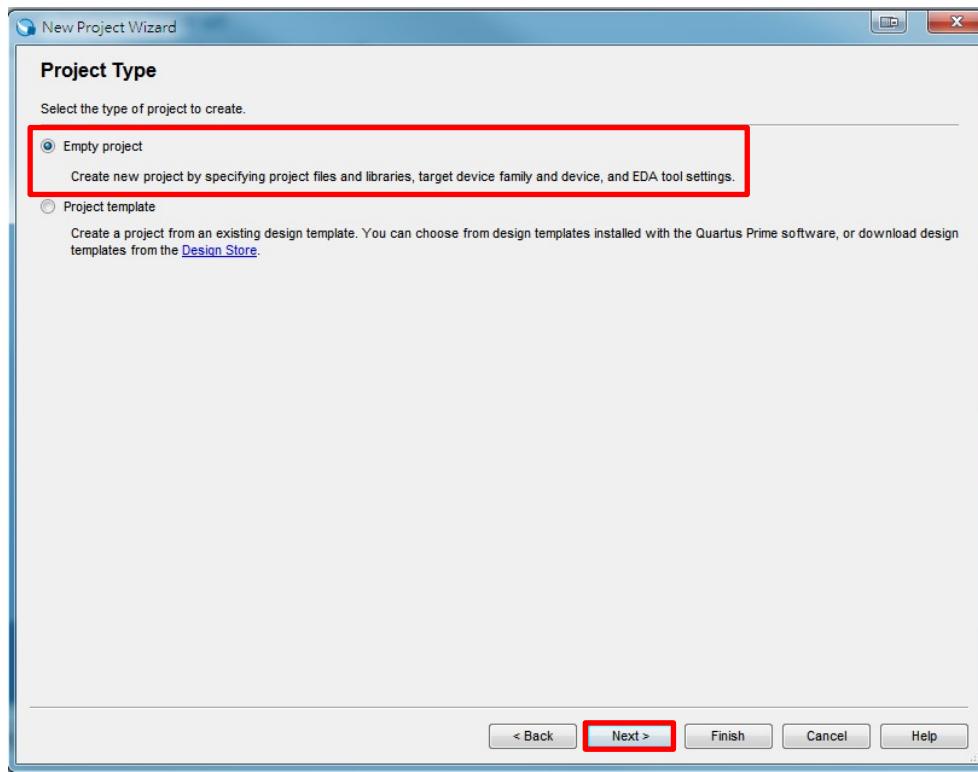
Quartus II Tutorial (3/10)

■ Specify the working directory and the name of the project



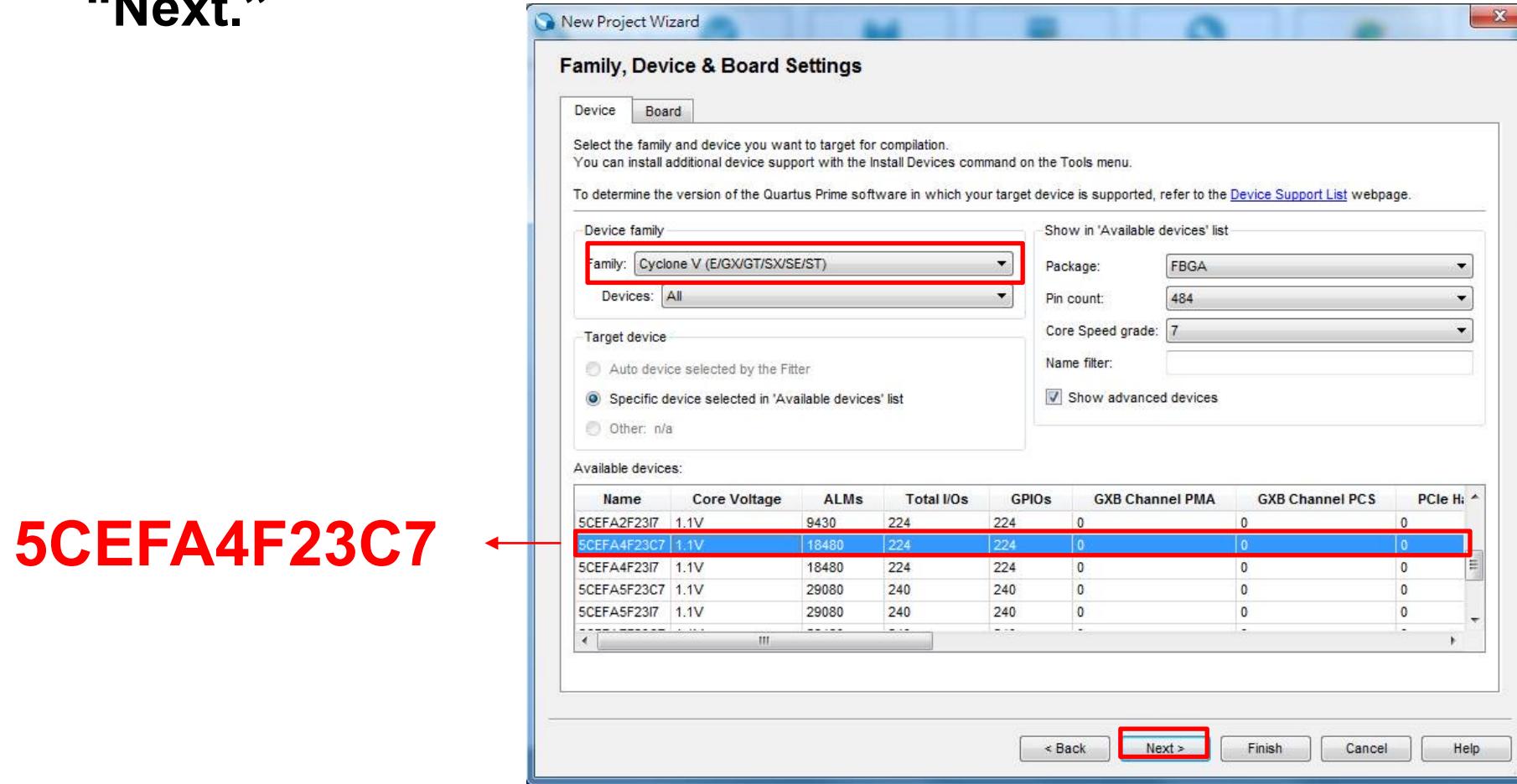
Quartus II Tutorial (4/10)

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



Quartus II Tutorial (5/10)

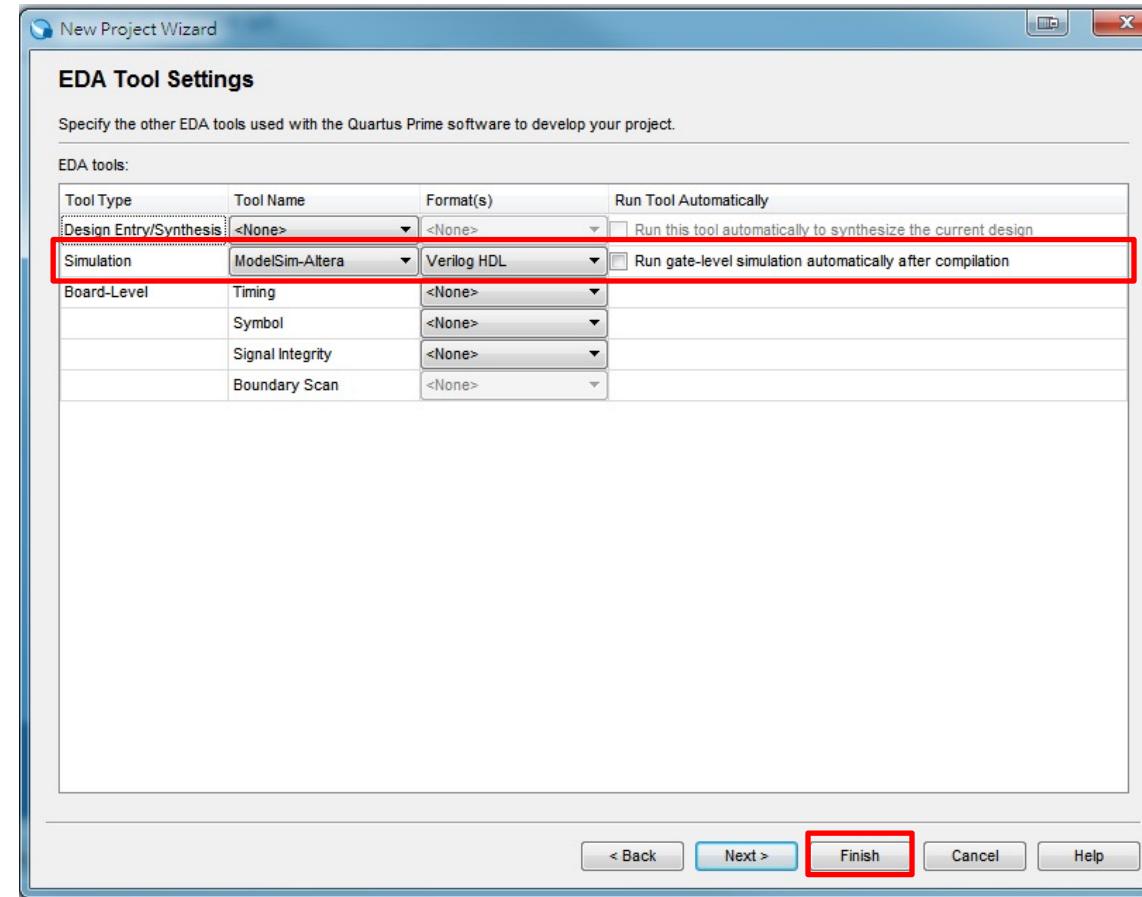
- Specify device settings - (DE0-CV Device family are used). Click “Next.”



5CEFA4F23C7

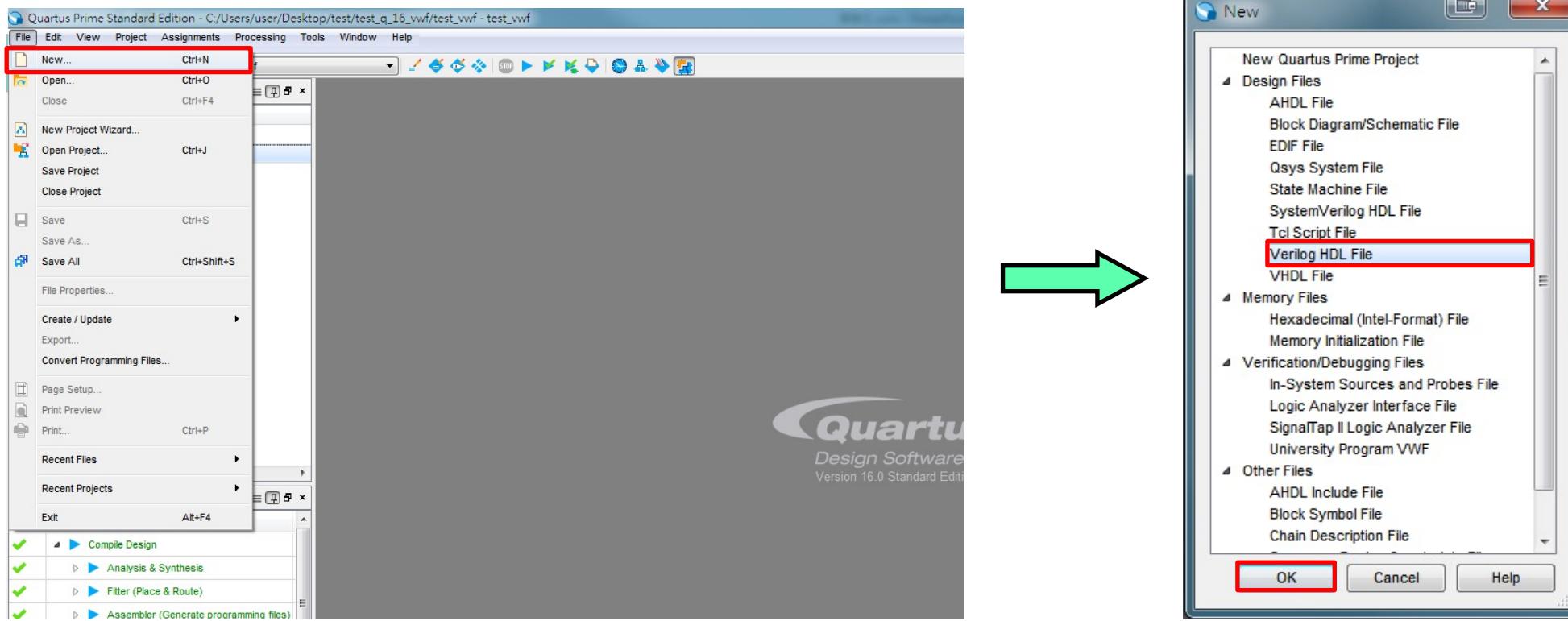
Quartus II Tutorial (6/10)

- 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 !!

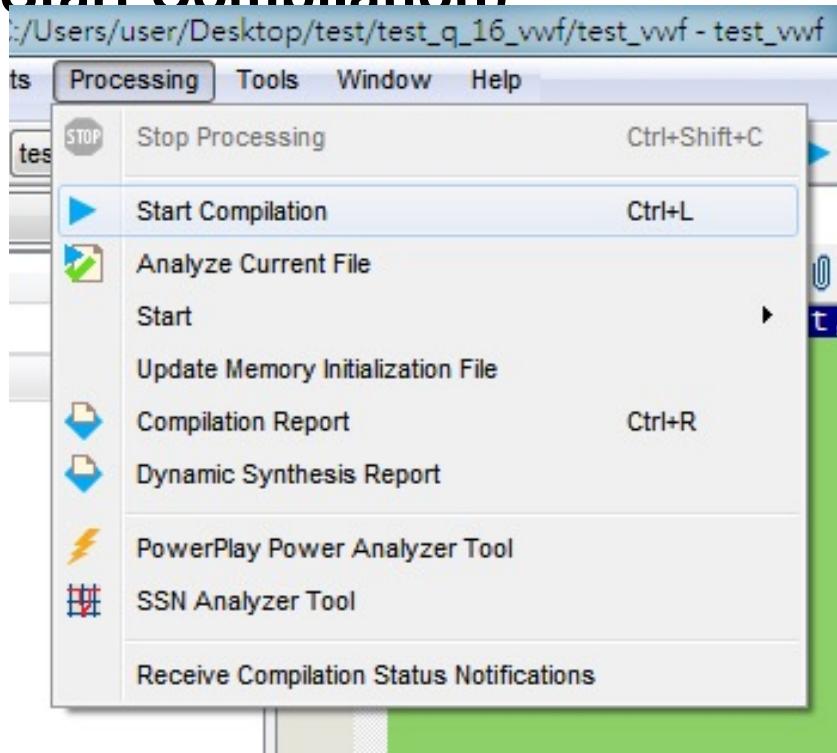
```
1: //File Name : Half_Adder.v
2: module Half_Adder(a, b, sum, carry);
3:   input a, b;
4:   output sum, carry;
5:
6:   assign sum = a ^ b;
7:   assign carry = a & b;
8:
9: endmodule
```

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



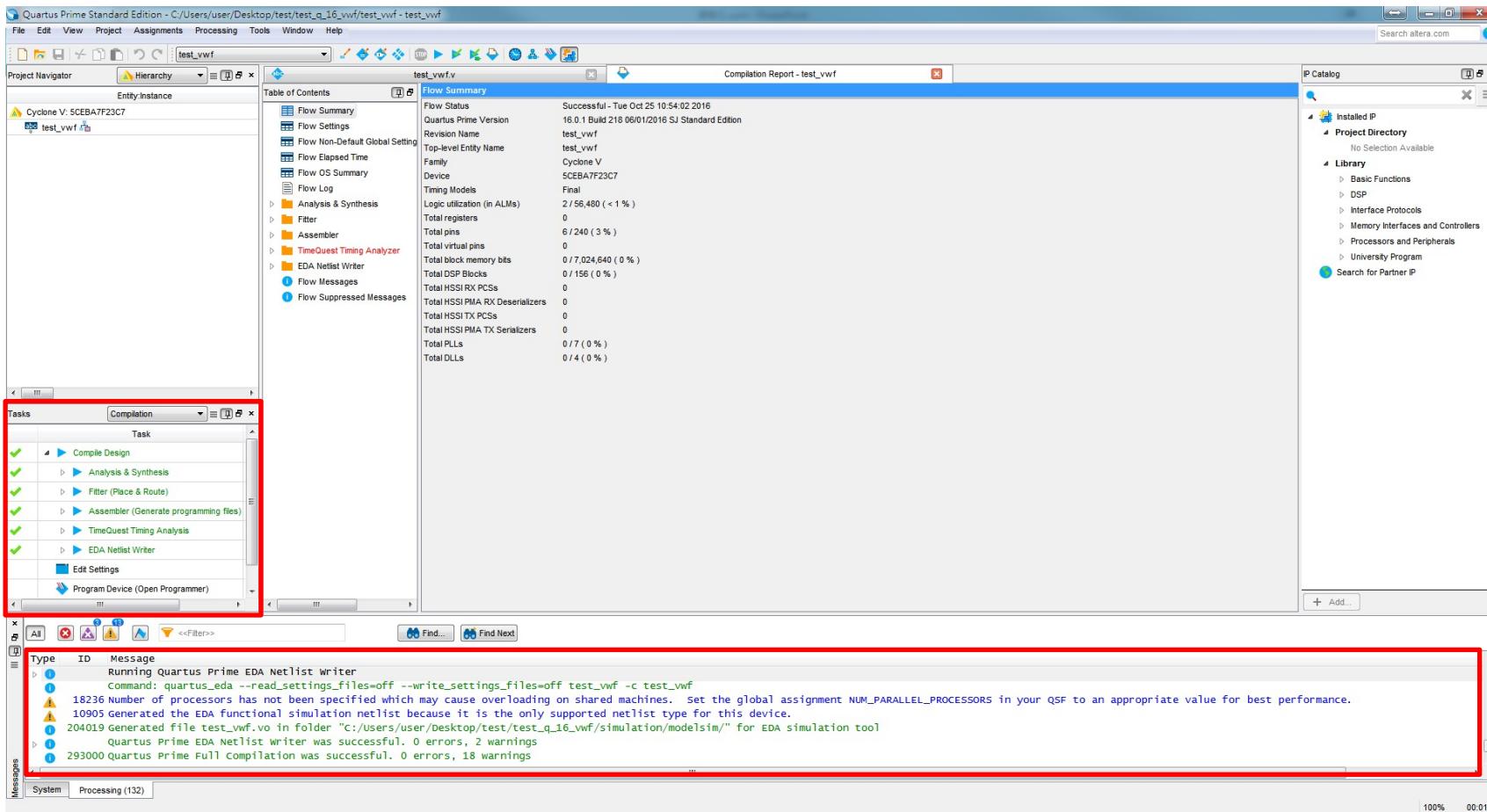
Quartus II Tutorial (9/10)

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



Quartus II Tutorial (10/10)

■ Successful compilation

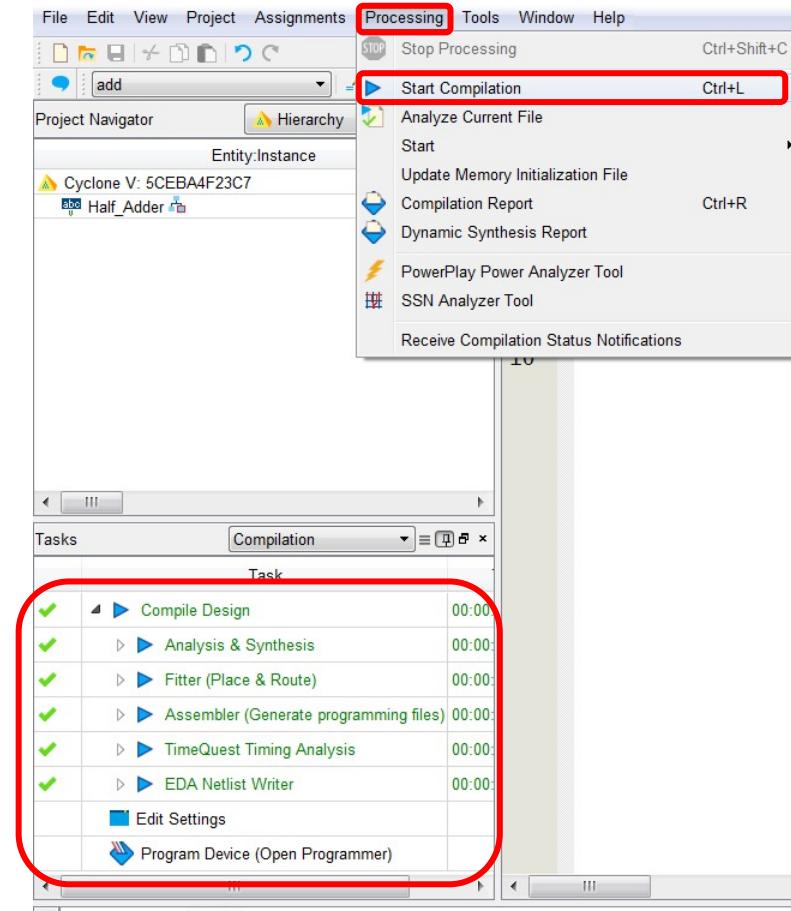


Programming DE0-CV (1/13)

```
1 module Half_Adder(a, b, sum, carry);
2   input a,b;
3   output sum, carry;
4   assign sum = a ^ b;
5   assign carry = a & b;
6 endmodule
```

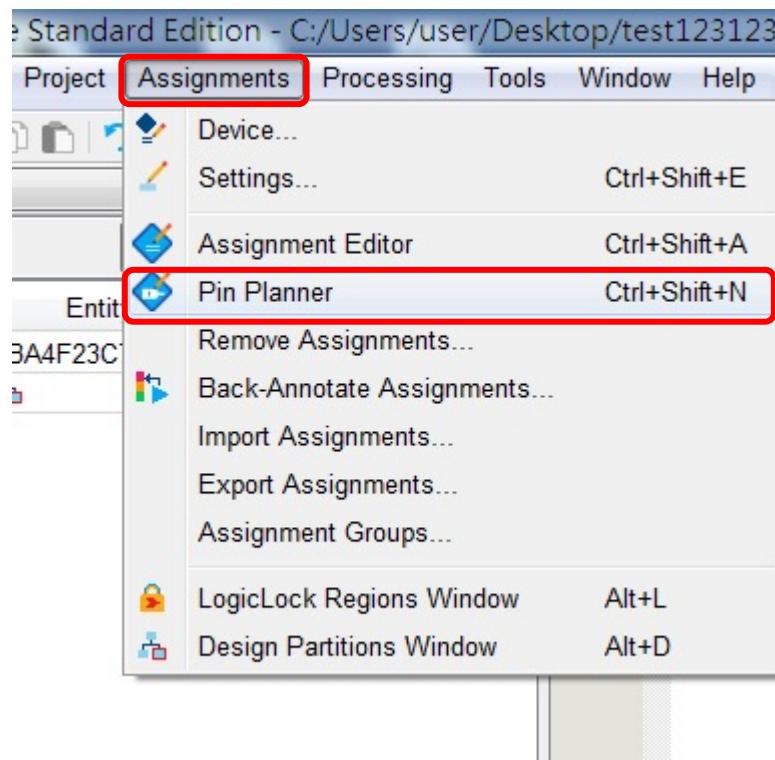
Programming DE0-CV (2/13)

■ Start compilation



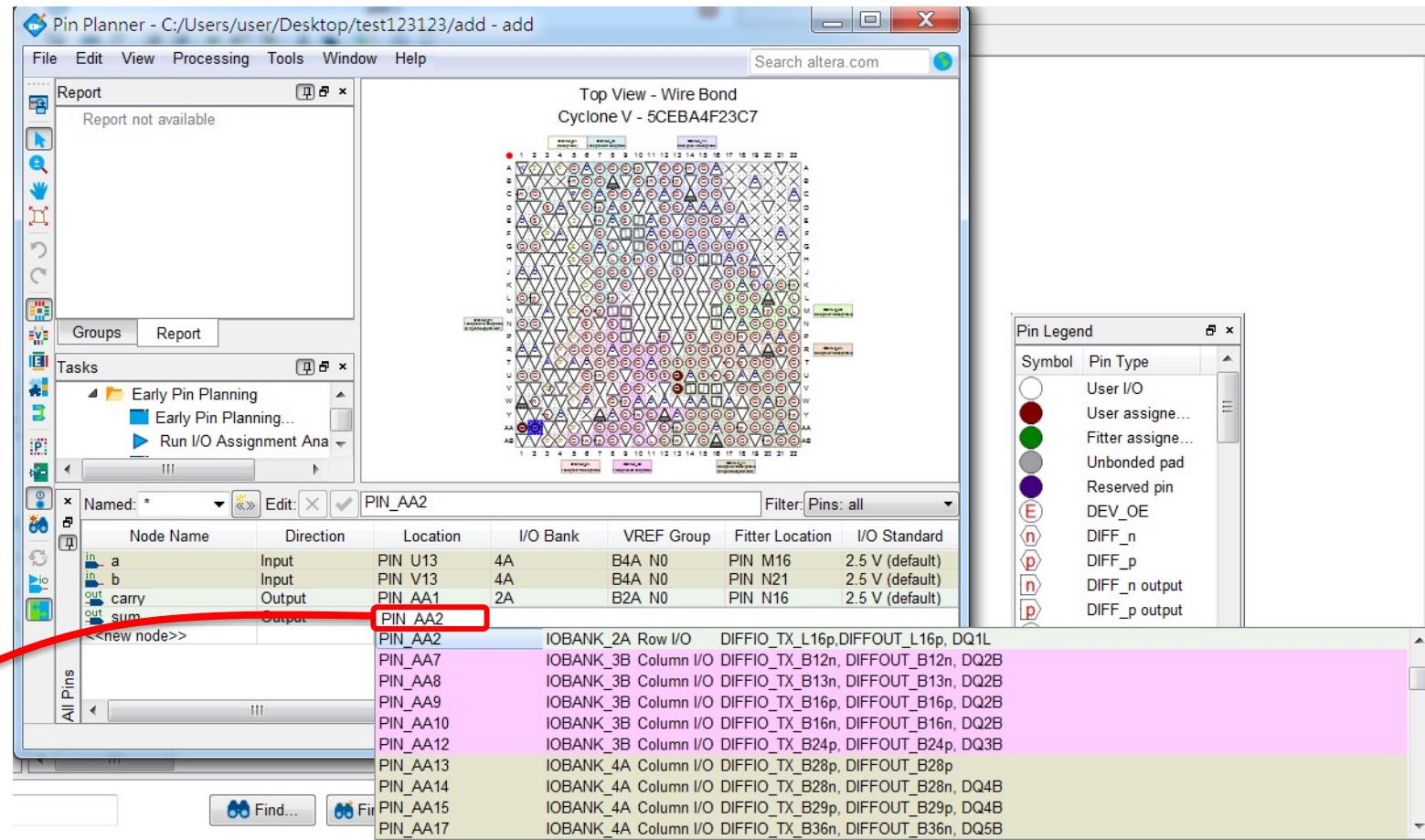
Programming DE0-CV (3/13)

■ Open Pin Planner



Programming DE0-CV (4/13)

■ Pin assignment



Double click

Programming DE0-CV (5/13)

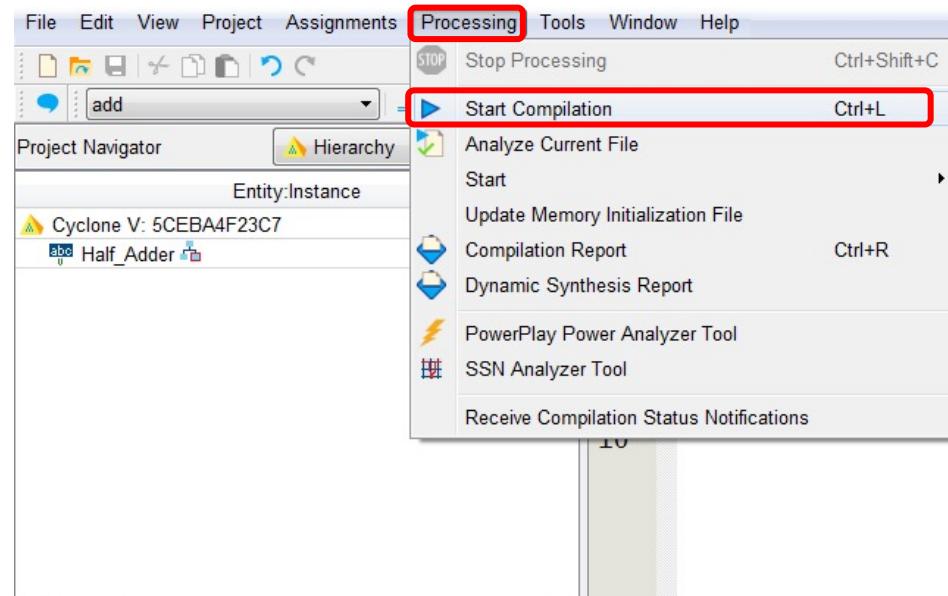
- Assign pin location to all inputs and outputs

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

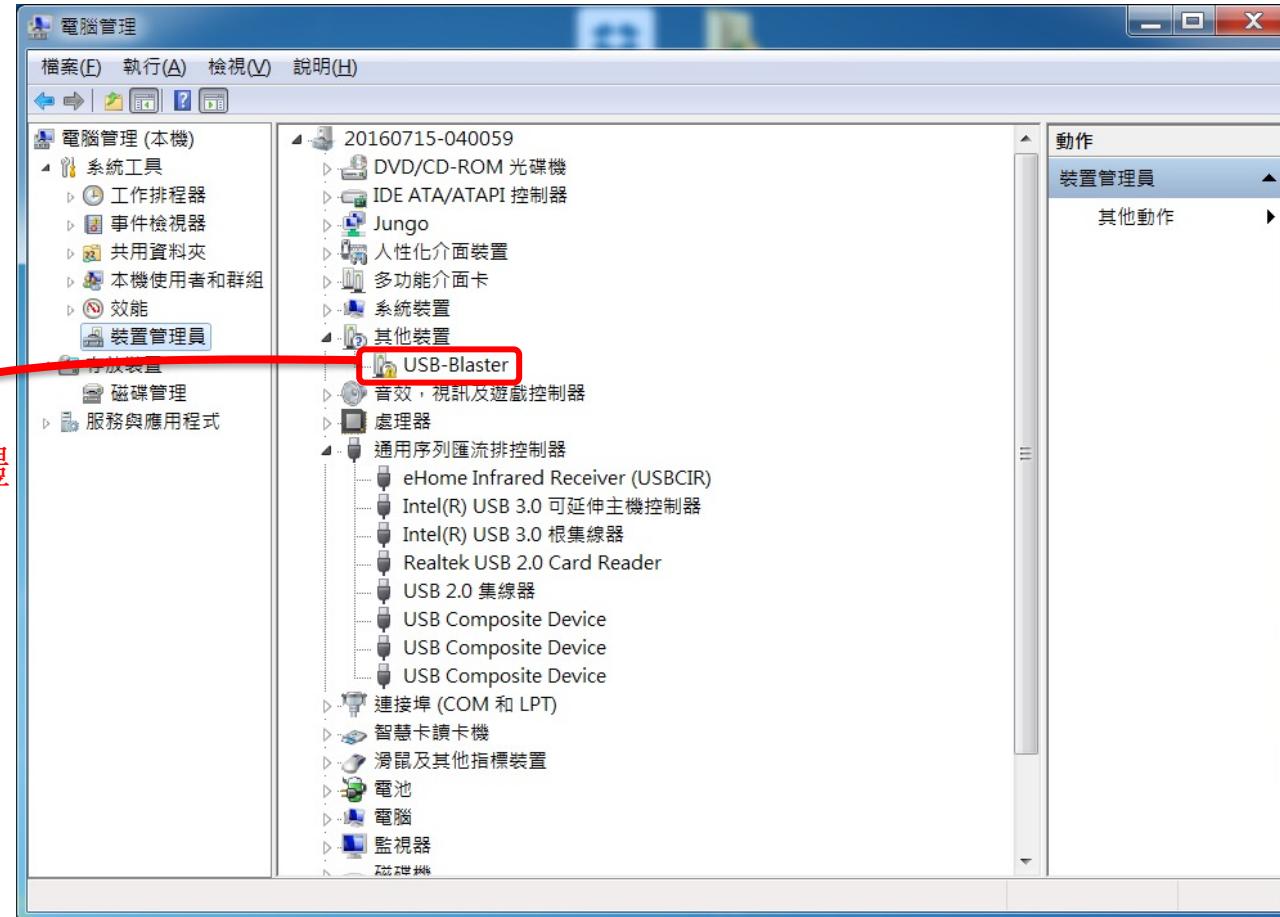
Programming DE0-CV (6/13)

■ Start compilation

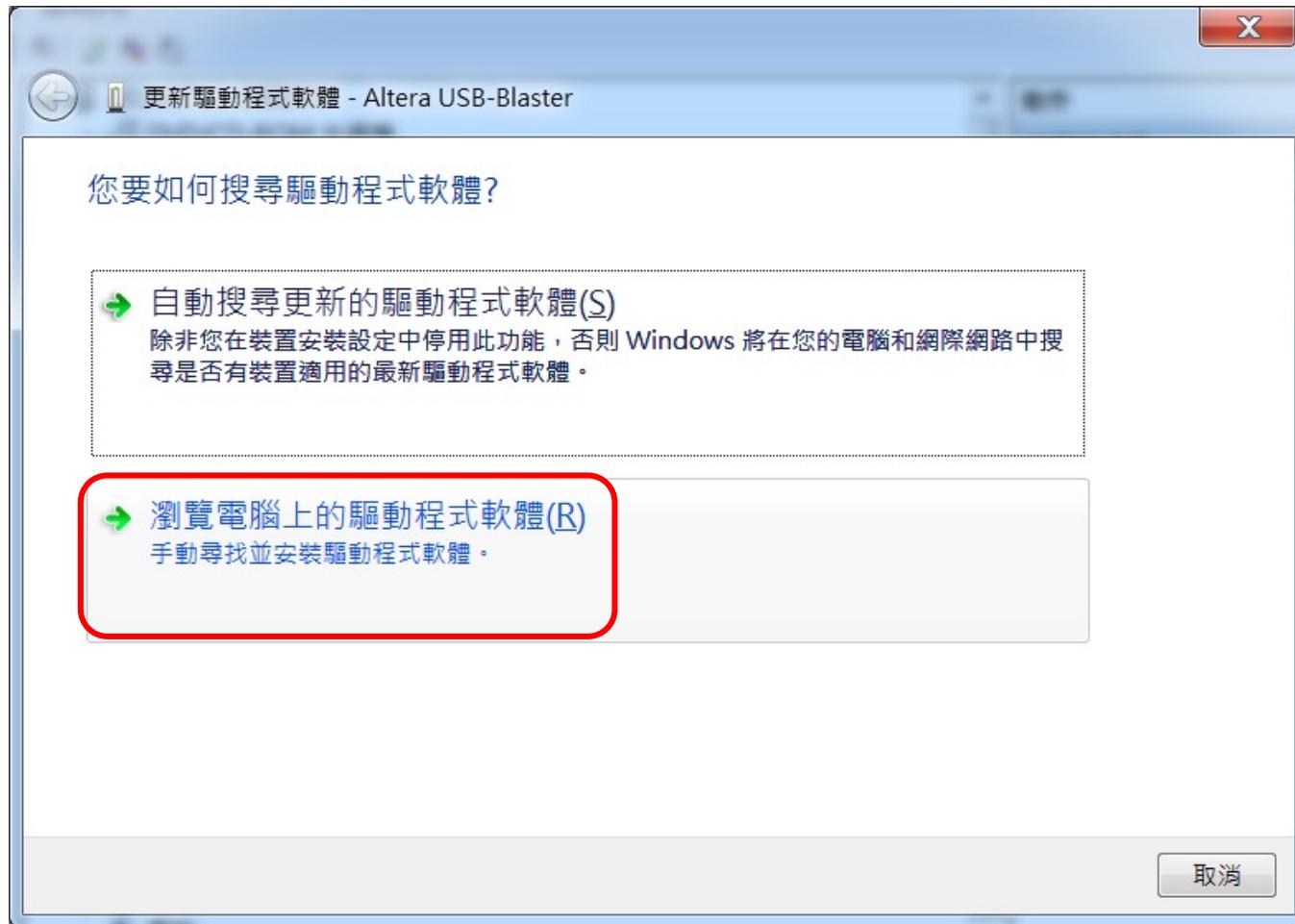


Programming DE0-CV (7/13)

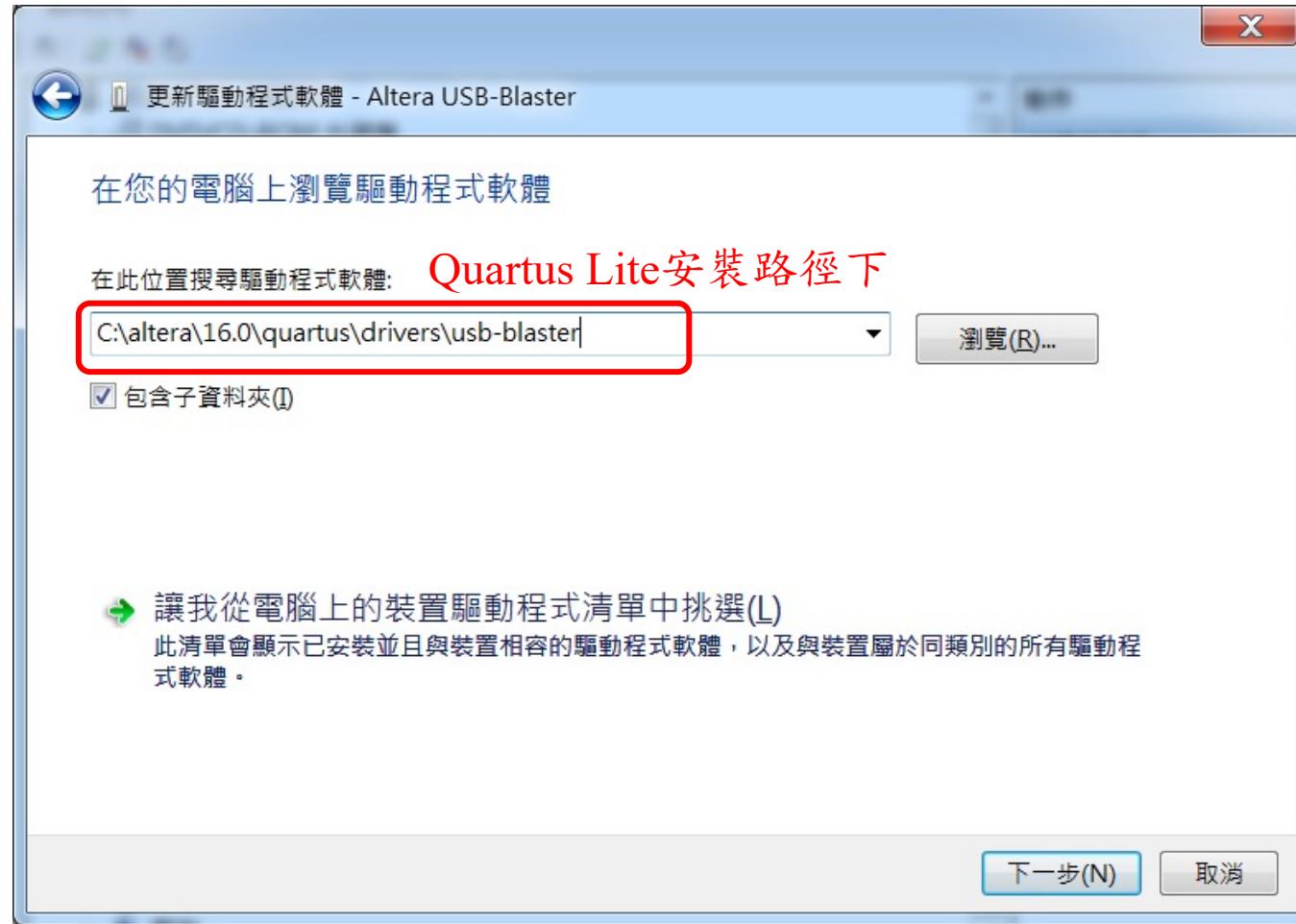
右鍵選更新驅動程式軟體



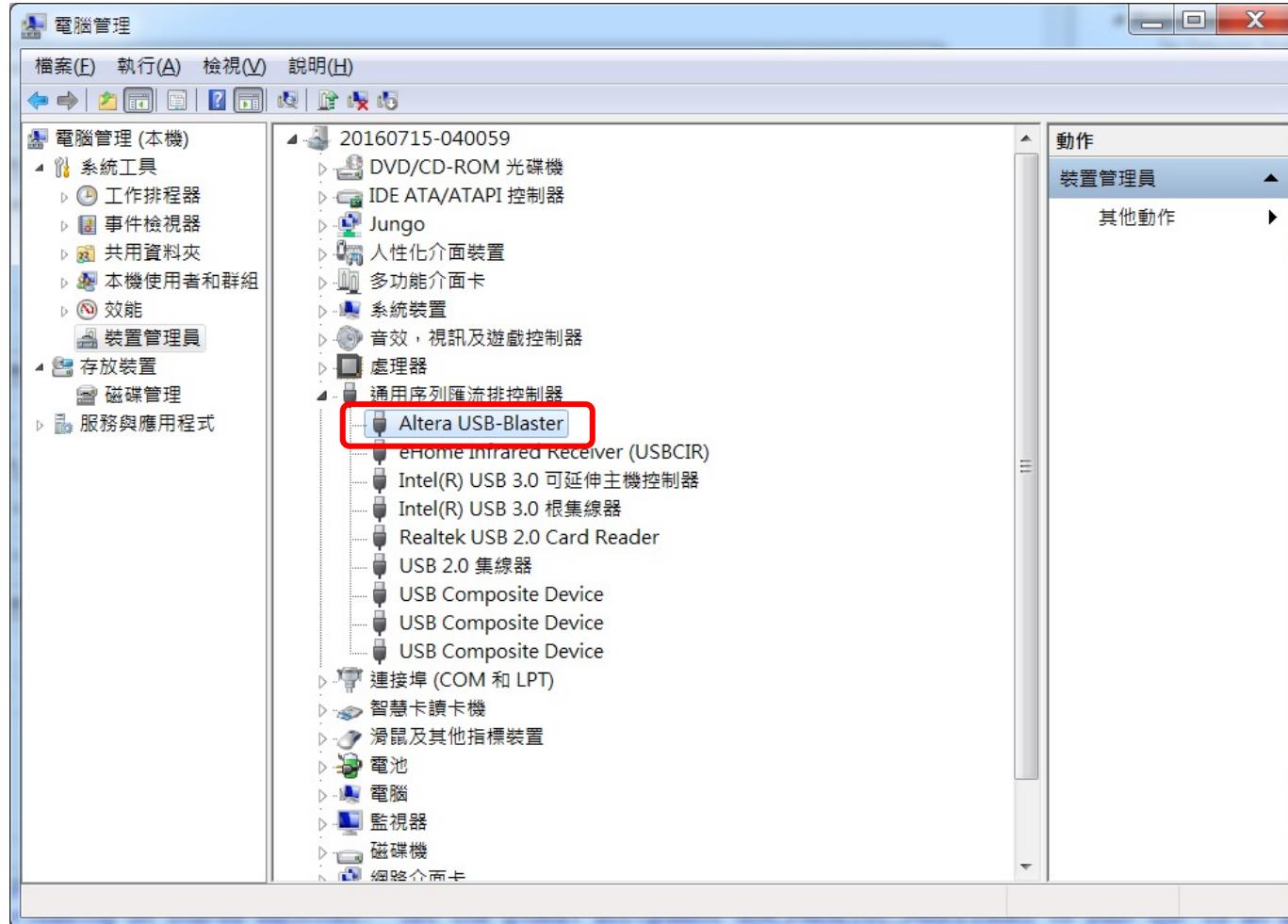
Programming DE0-CV (8/13)



Programming DE0-CV (9/13)

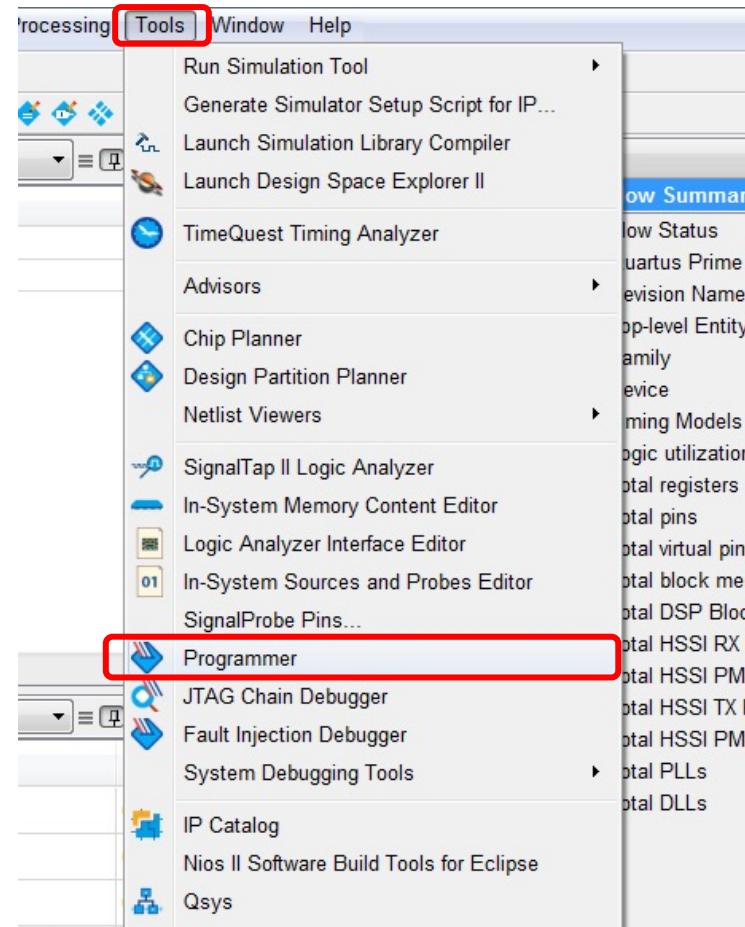


Programming DE0-CV (10/13)



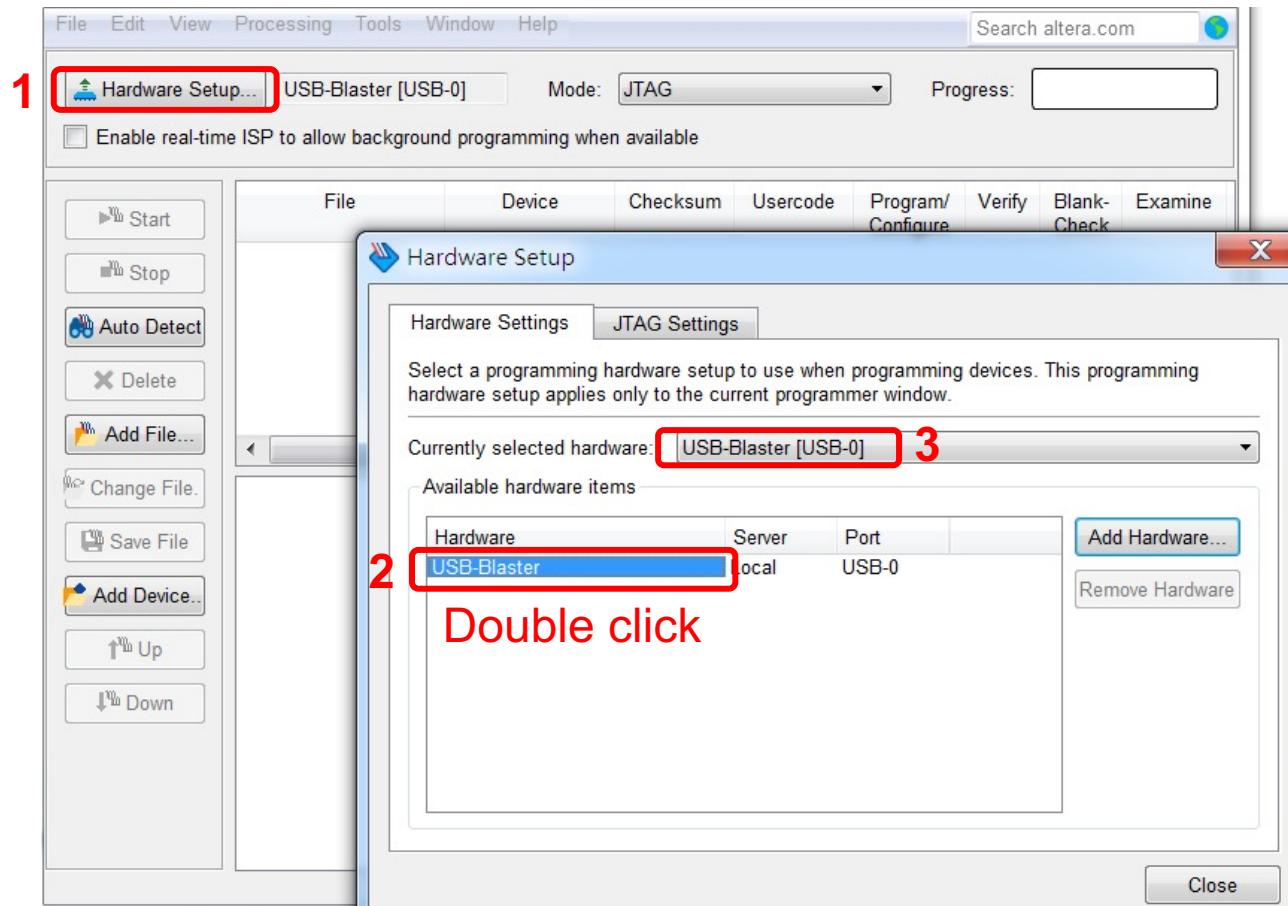
Programming DE0-CV (11/13)

■ Programming device



Programming DE0-CV (12/13)

■ Hardware setup: add USB-Blaster



Programming DE0-CV (13/13)

■ Programming device

