

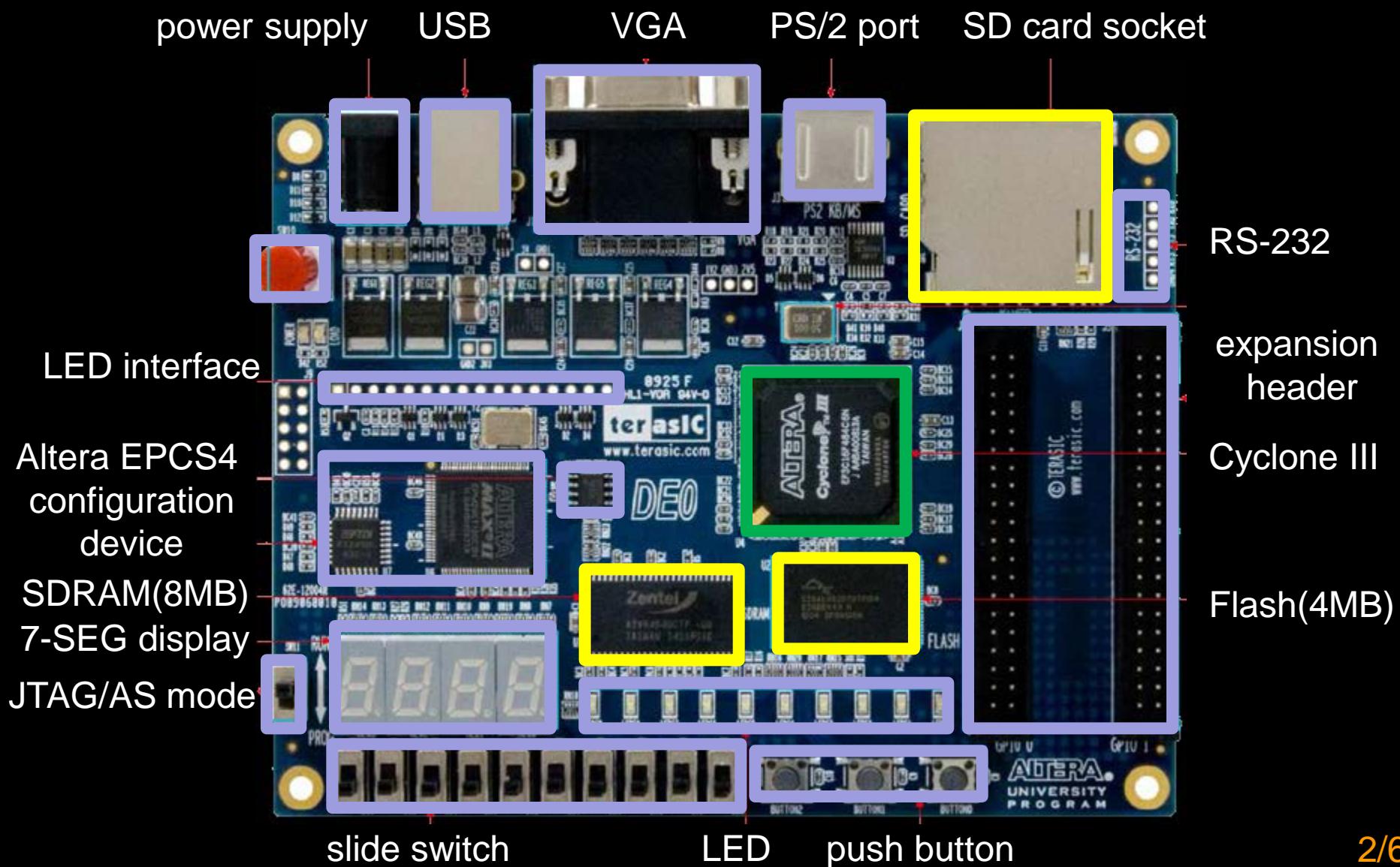
FPGA Design Tutorial

2016/02/26 YZU CSE

Outline

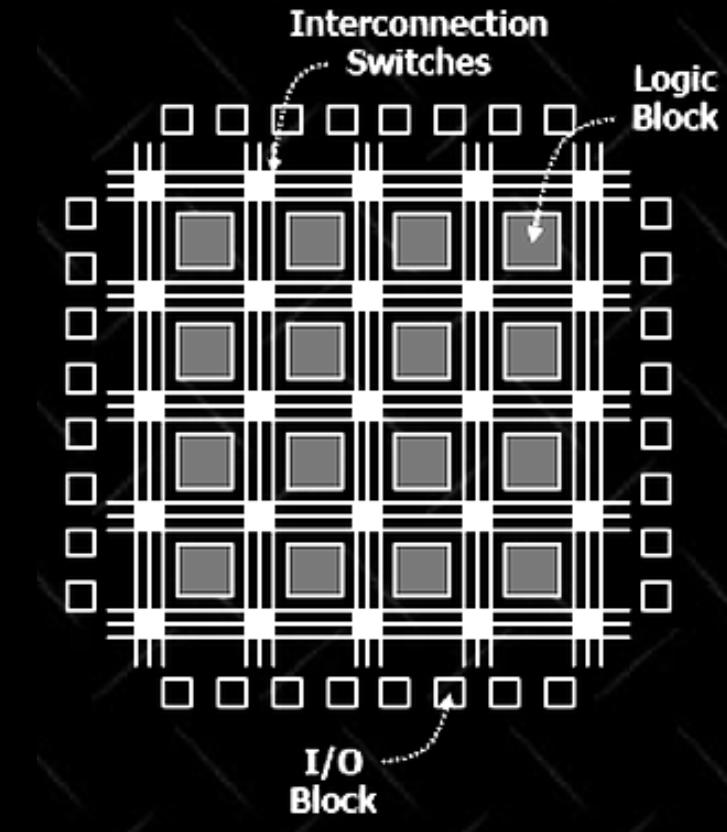
- **Introduction - Altera DE0 FPGA Board**
- **FPGA Design with Quartus II**
 - Lab1: Schematic Design Flow
 - Lab2: Dip switch to LED
 - Lab3: Dip switch to 7-seg (Binary to Decimal)

Layout and Component of DE0 Board



Field-programmable Gate Array (FPGA)

- **Advantage:**
 - Programmability
 - Short turnaround time
 - Low manufacturing cost
- **Disadvantage:**
 - Low performance
 - High power consumption
 - High unit cost
- **Why do we use FPGA for this course?**



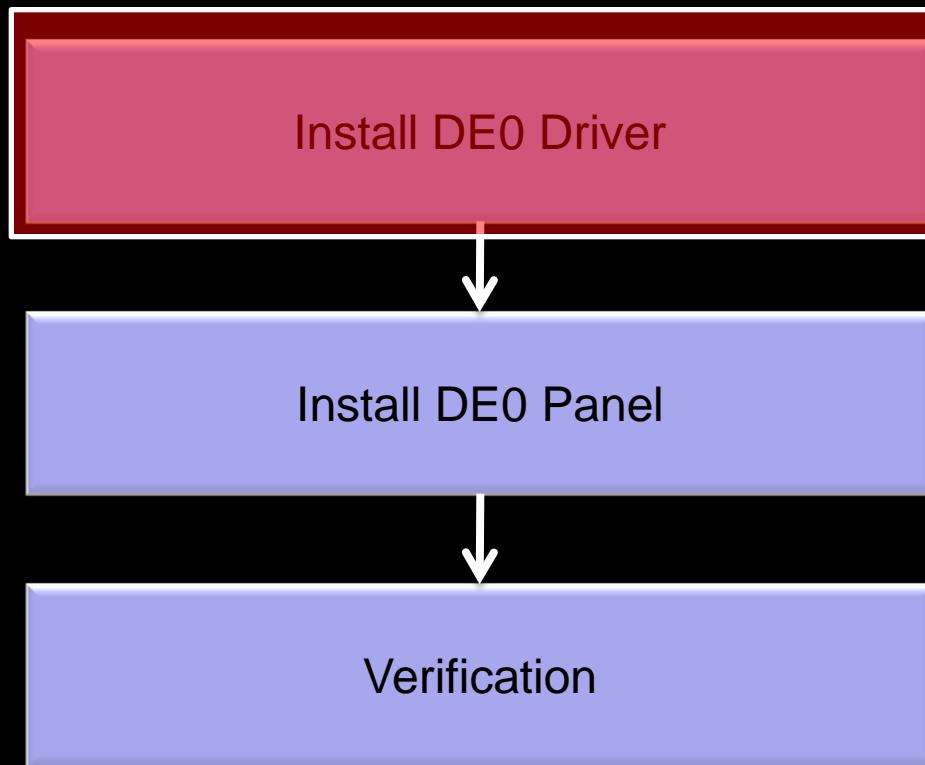
Course Syllabus - 1

- [01] **FPGA Design Flow (Combination Circuit)**
- [02] **Sequential Circuits and MegaFunction**
- [03] **Sequential Circuits and MegaFunction II**
- [04] **ModelSim and Testbench**
- [05] **Basic System with SOPC and Software Build Tools**
- [06] **SDRAM and FLASH Memory**
- [07] **NiosII Assembly Programming**
- [08] **Custom Instruction**
- [09] **Timer – Interrupt and Polling**
- [10] **Altera University Program - VGA**
- [11] **Altera University Program - ColorVGA and PS/2**

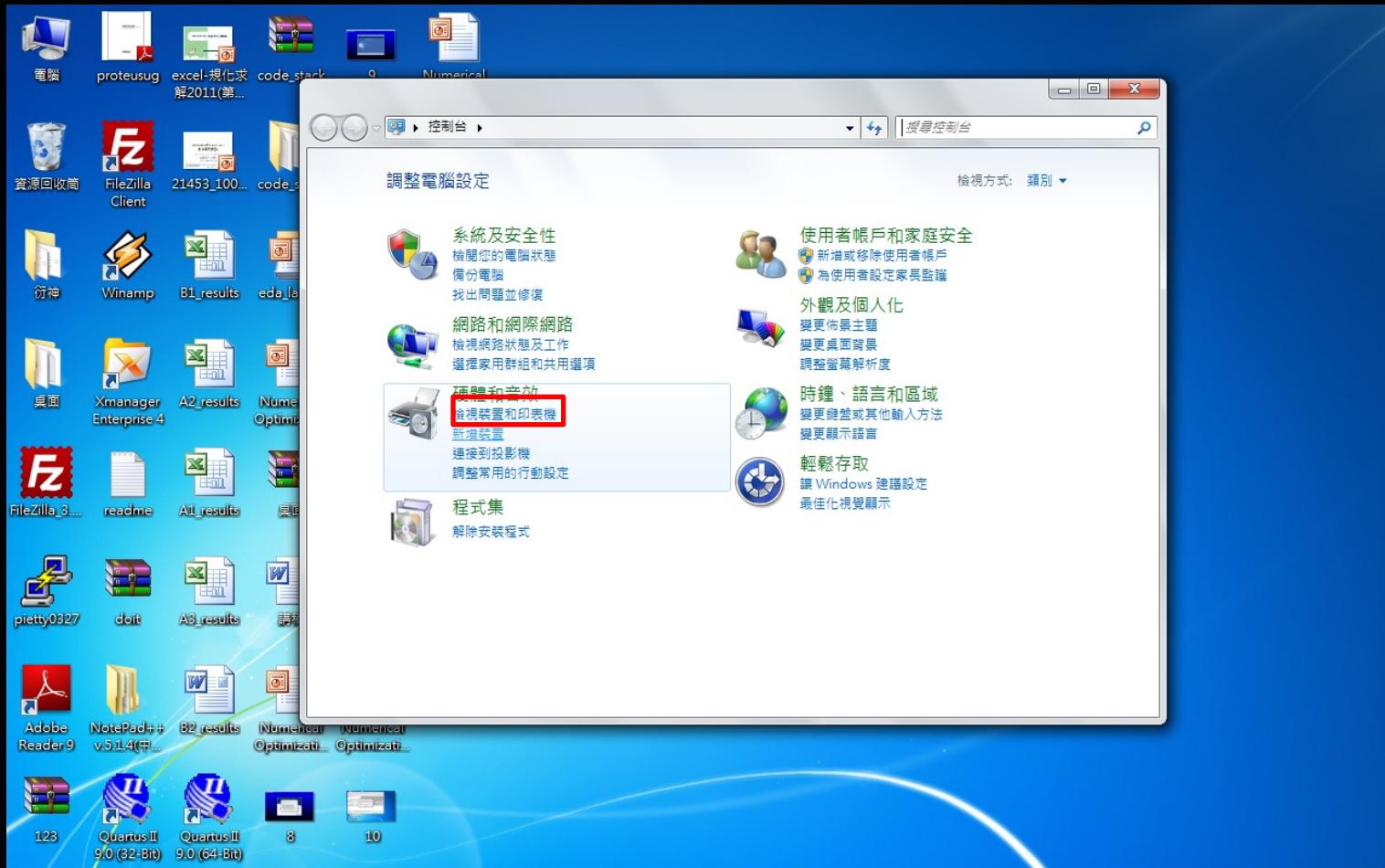
Course Syllabus - 2

- [12] uClinux Kernel Configuration and Compilation
- [13] U-BOOT
- [14] Advanced DE0
- [15] Linux Driver

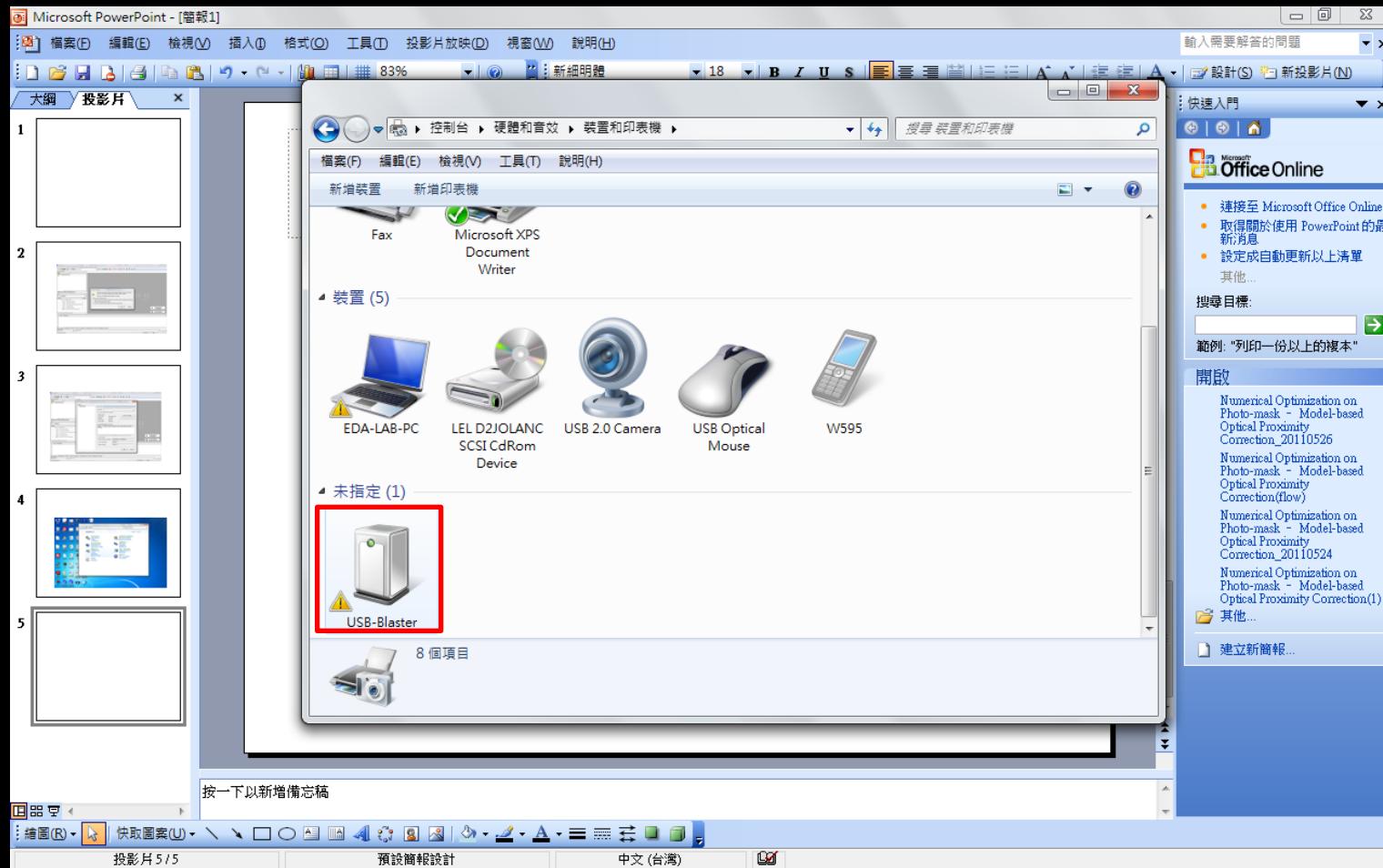
Getting Started with Altera DE0 Board



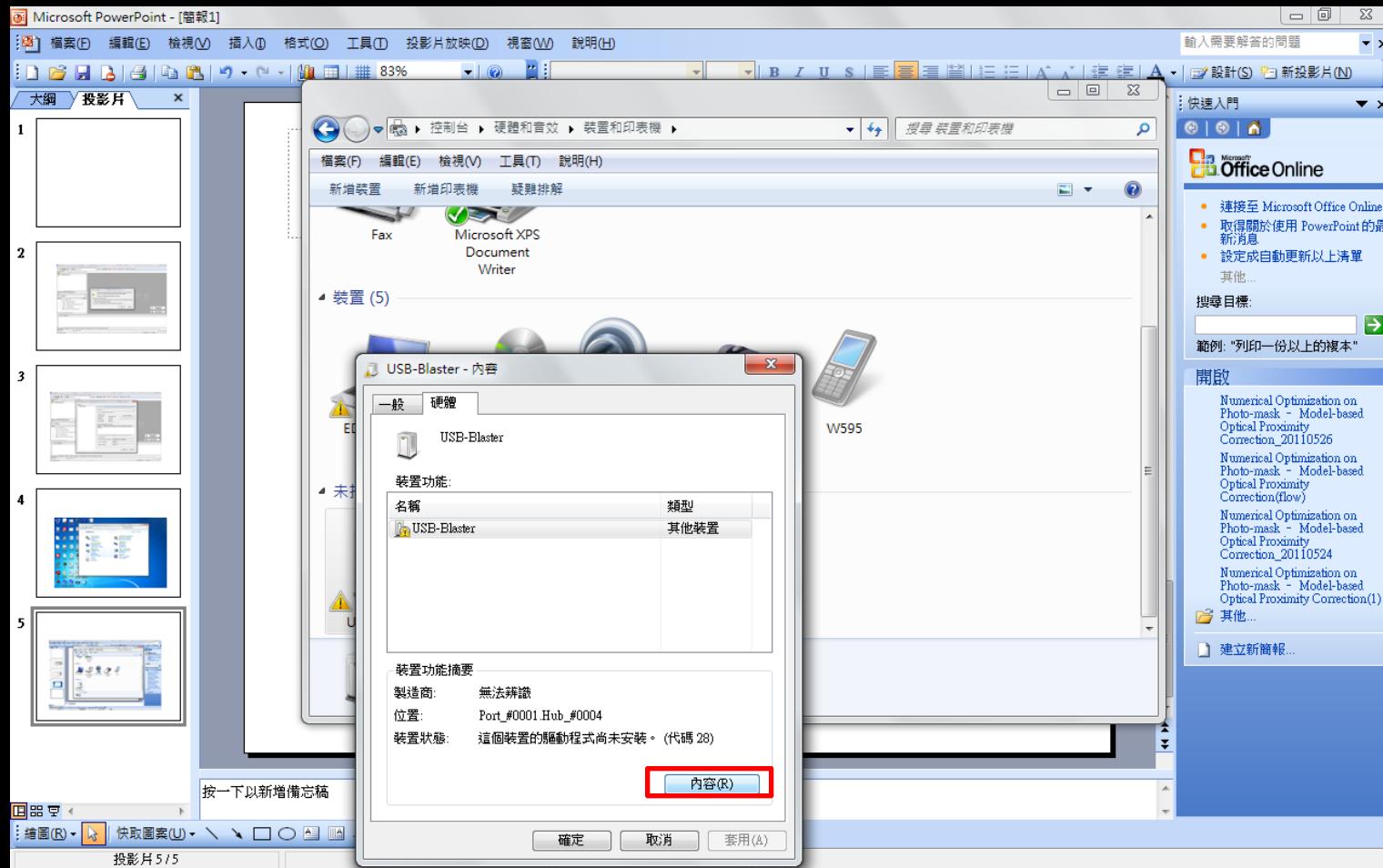
DE0 Driver Installation (1/7)



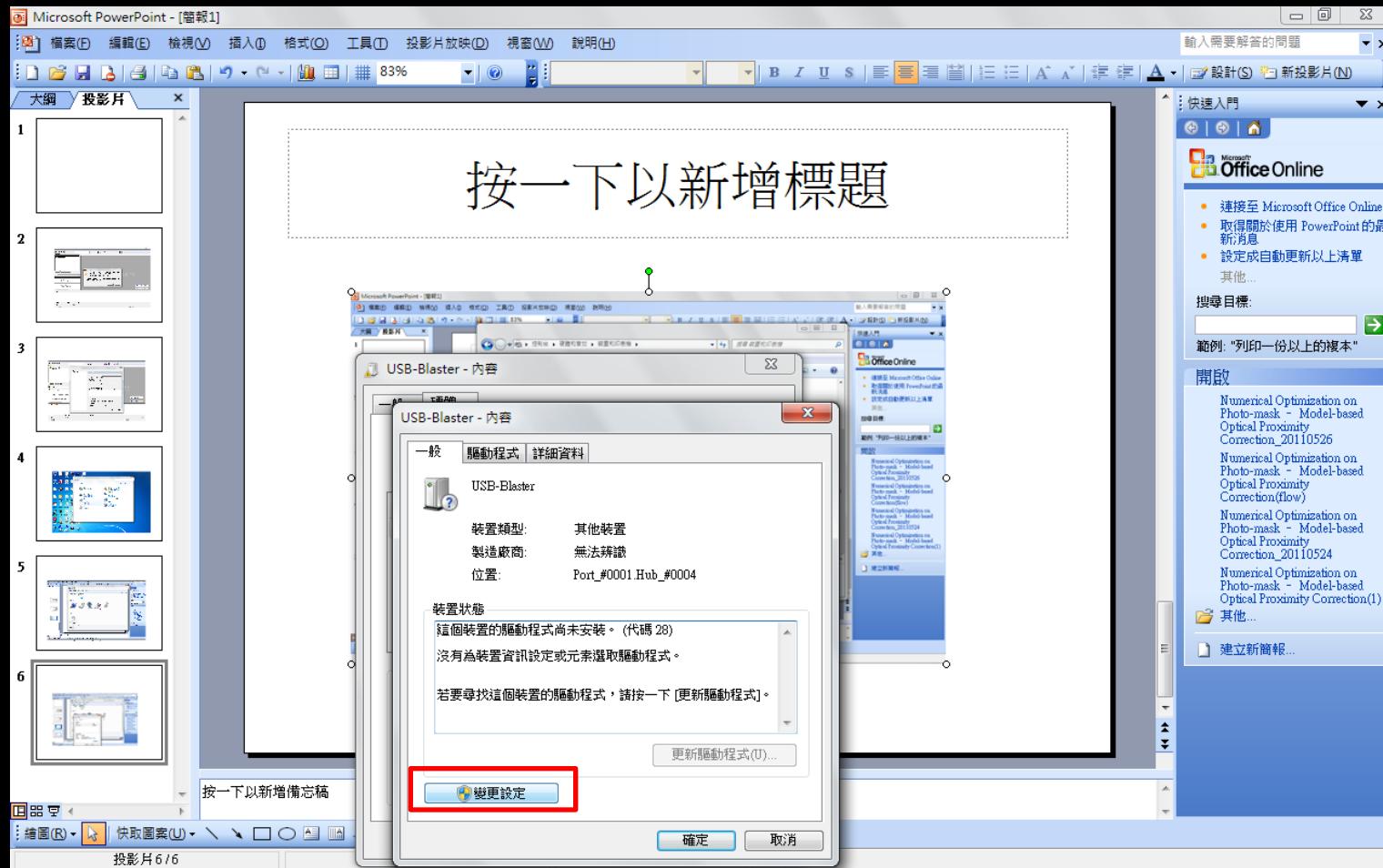
DE0 Driver Installation (2/7)



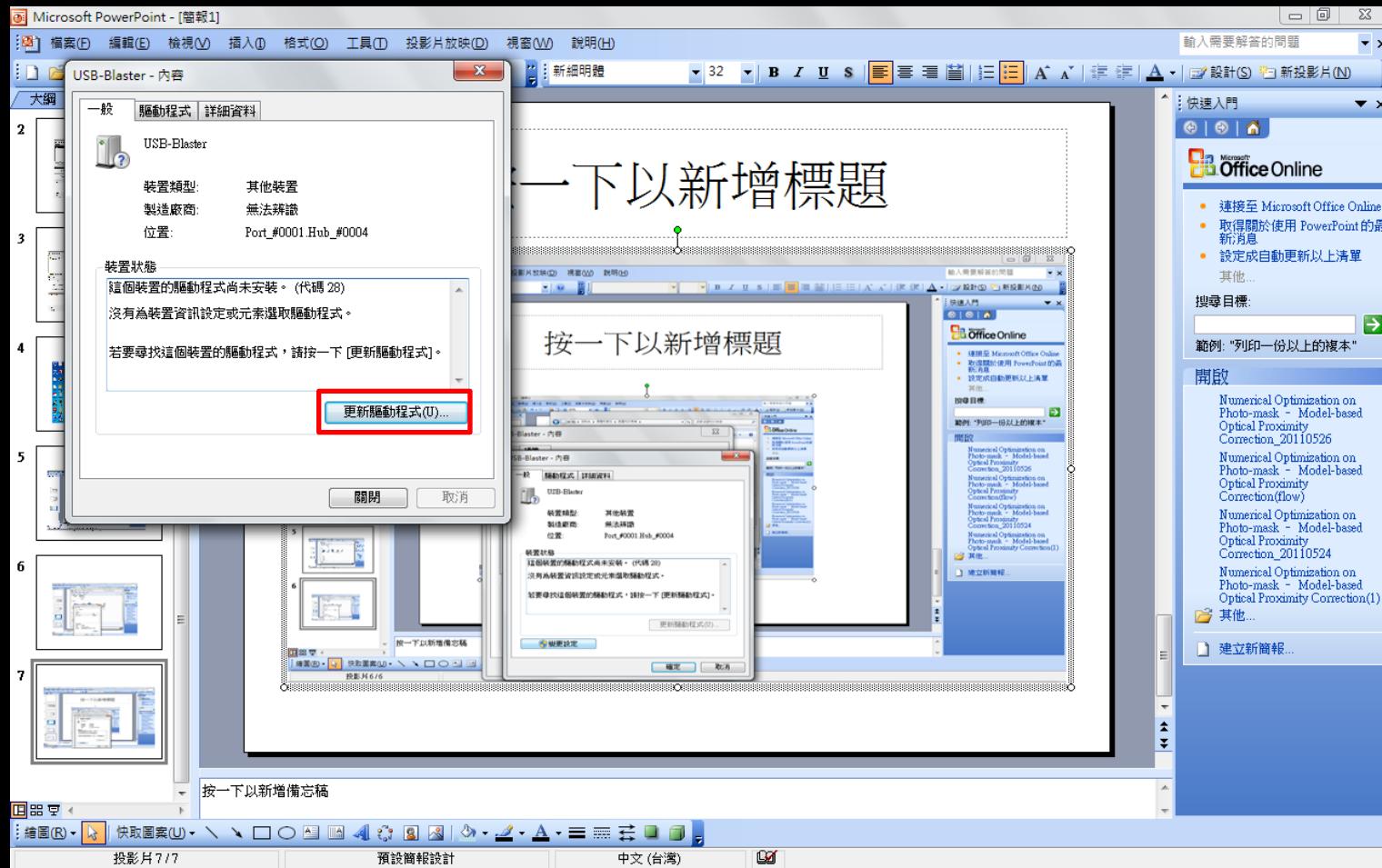
DE0 Driver Installation (3/7)



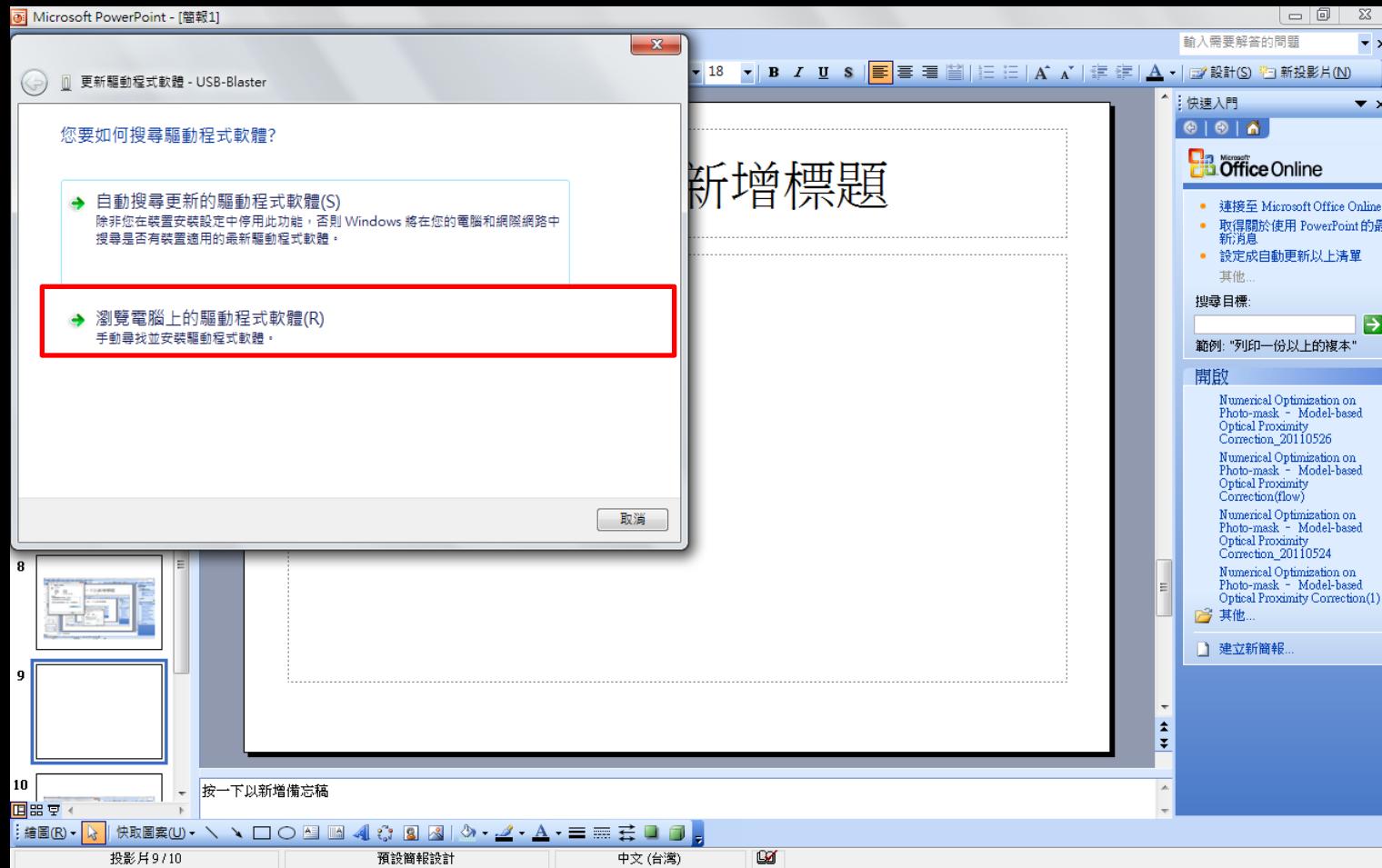
DE0 Driver Installation (4/7)



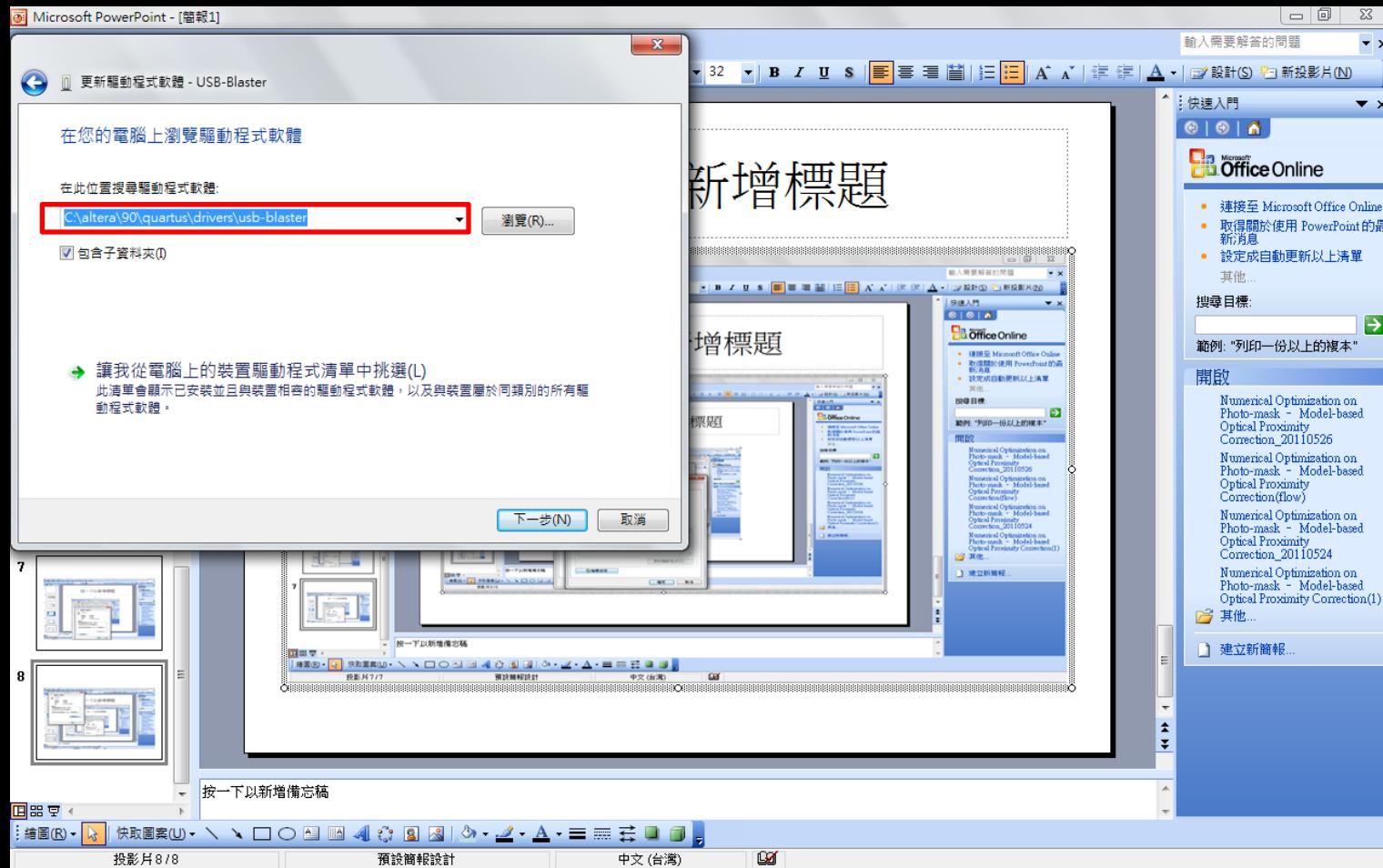
DE0 Driver Installation (5/7)



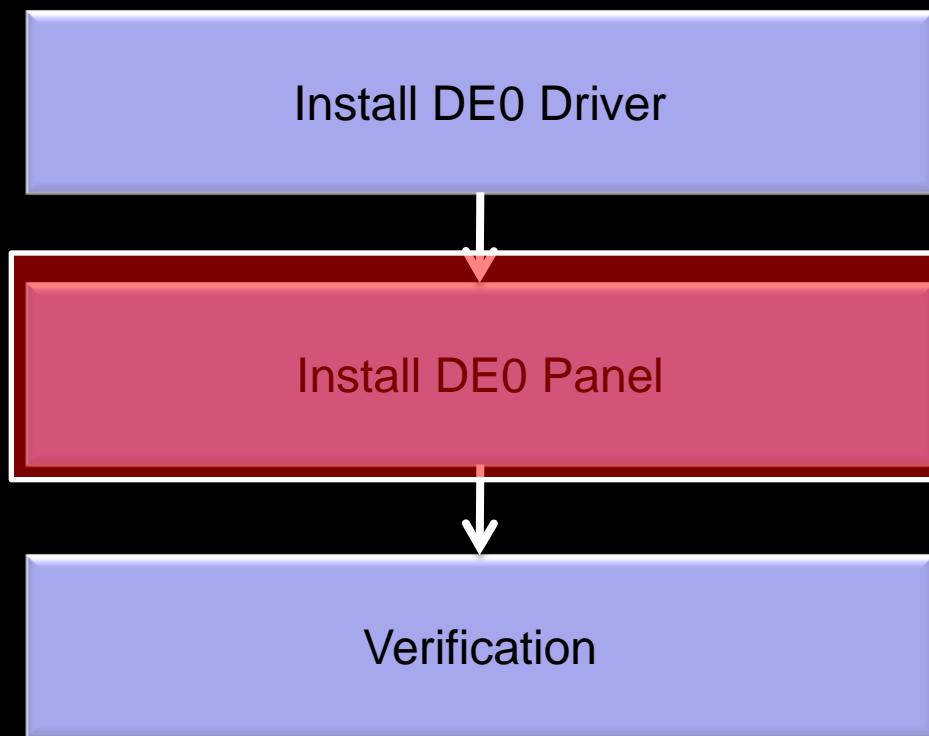
DE0 Driver Installation (6/7)



DE0 Driver Installation (7/7)

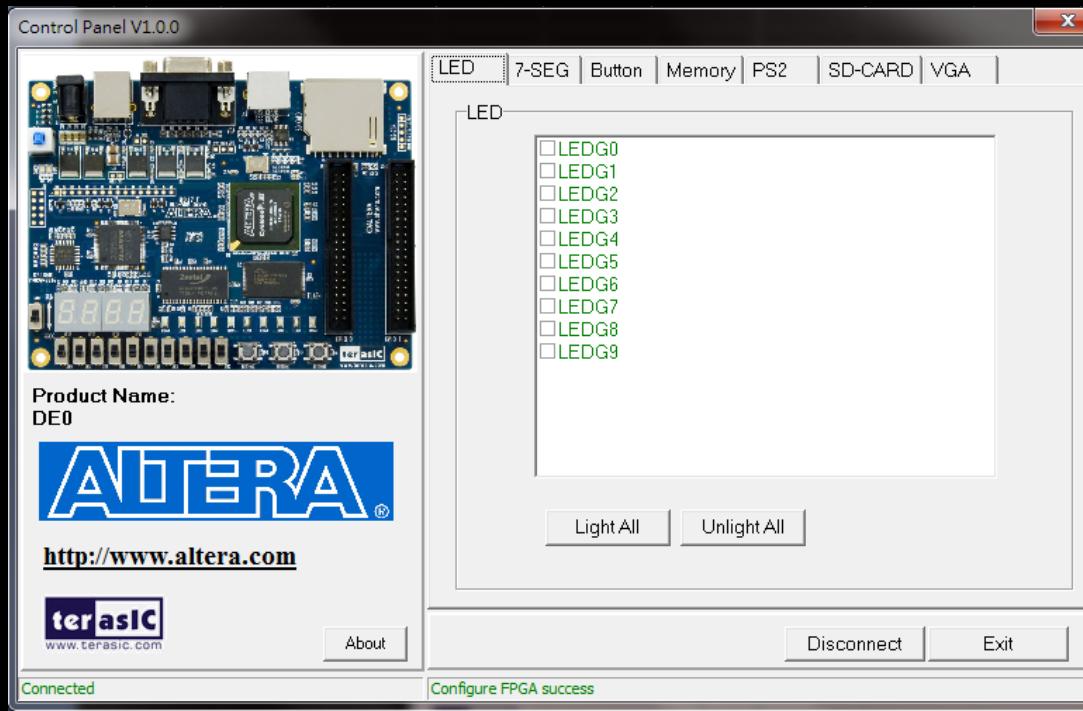


Getting Started with Altera DE0 Board

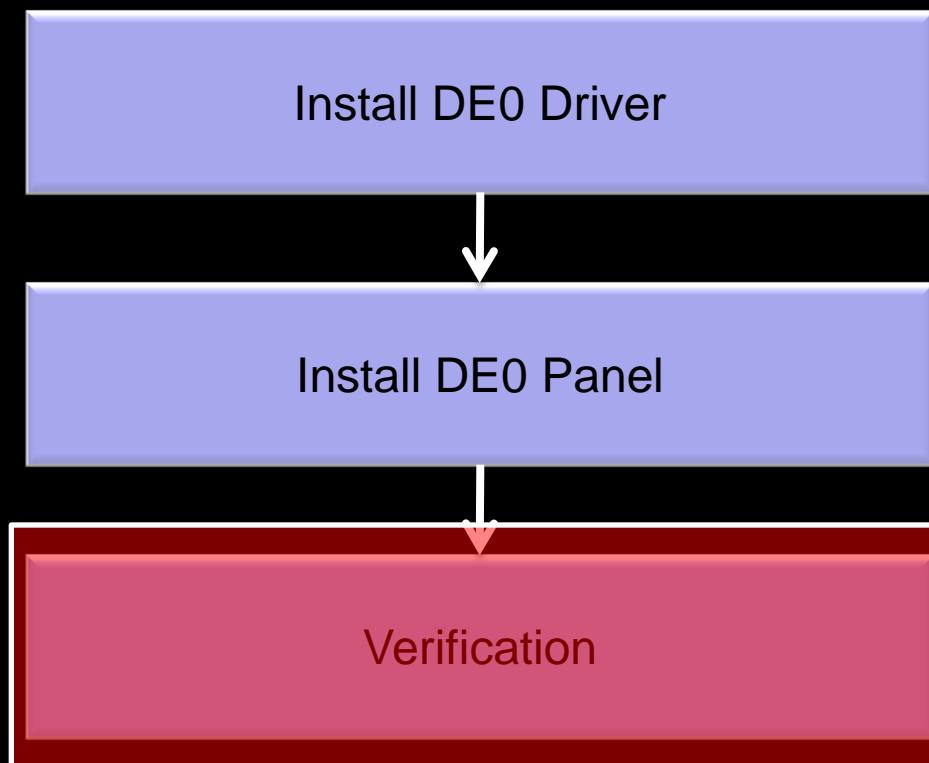


DE0 Control Panel

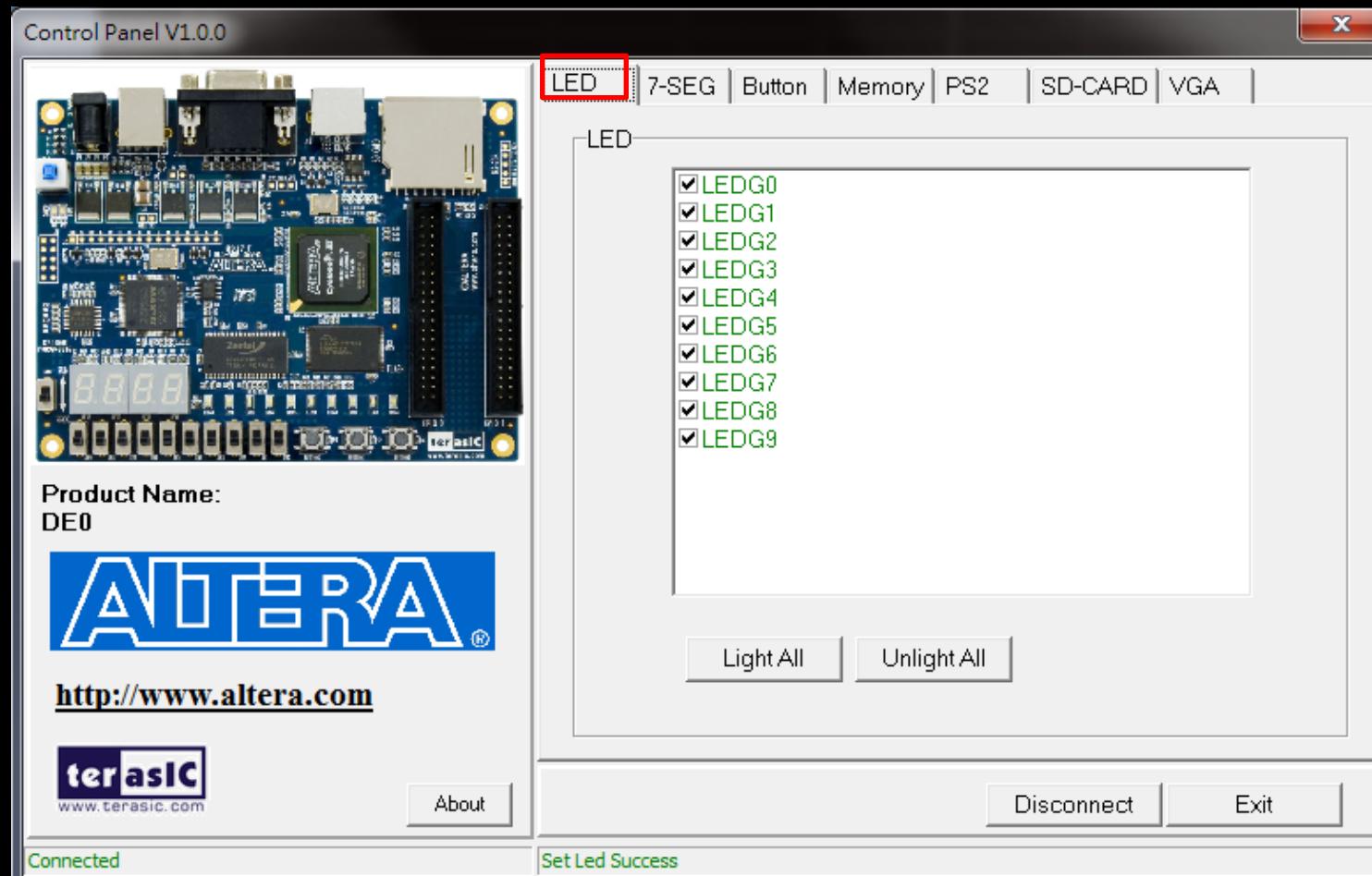
- Copy the “Control_panel” from “DE0 CD_ROM” to C:\altera\10.1\quartus
- Click “DE0_ControlPanel.exe”
- Click “Connect”



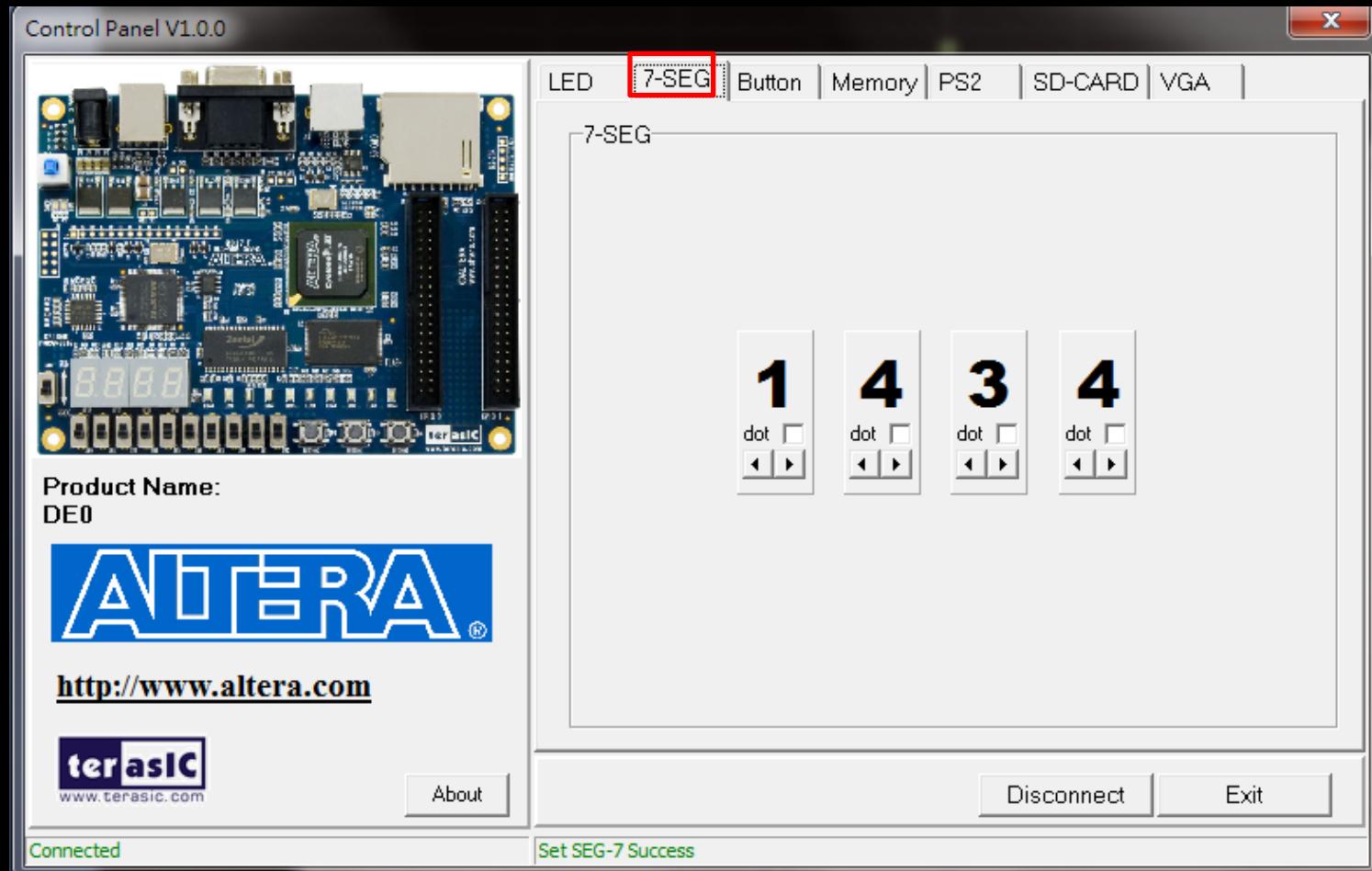
Getting Started with Altera DE0 Board



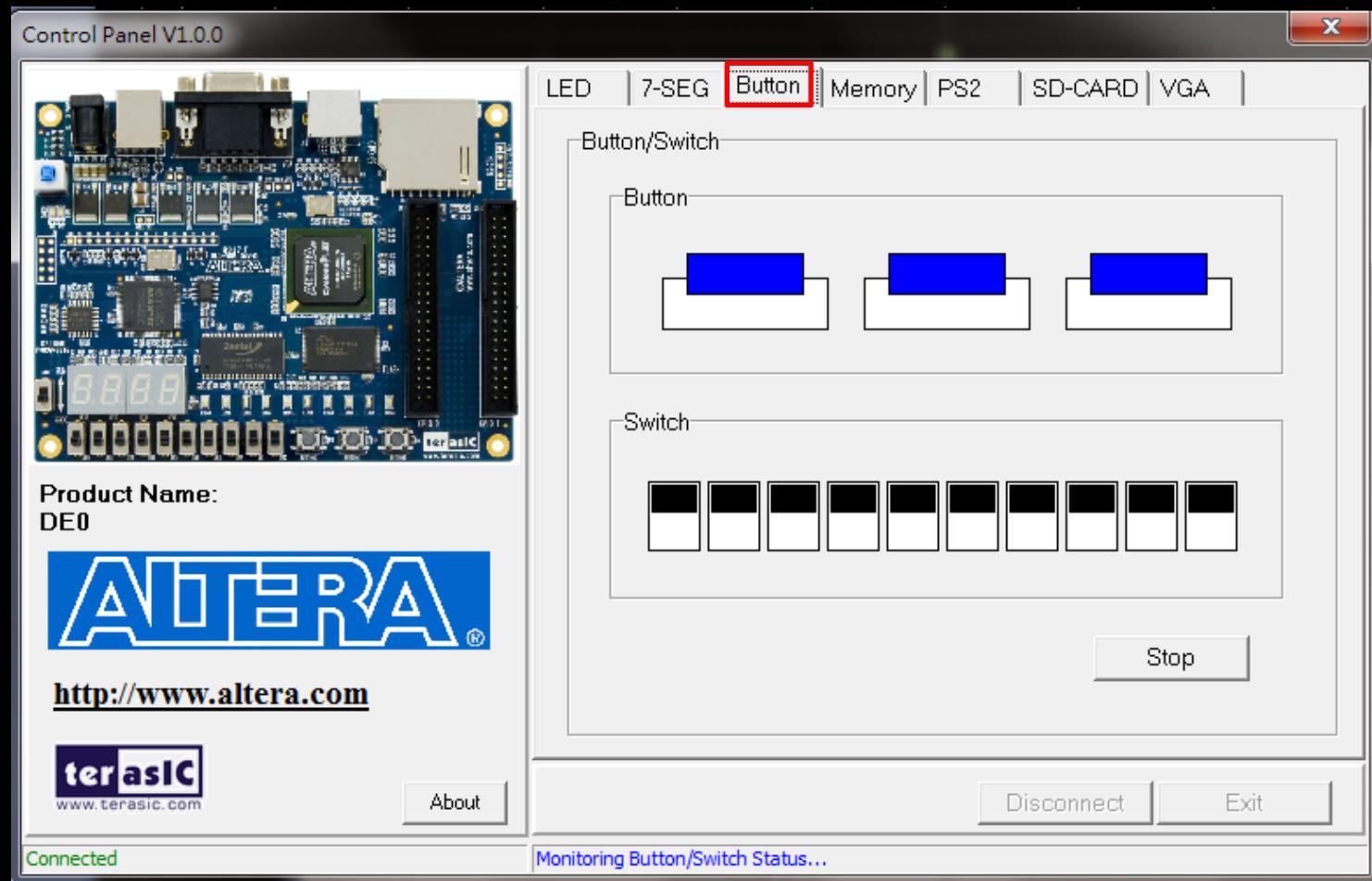
Verification of LED



Verification of 7-SEG



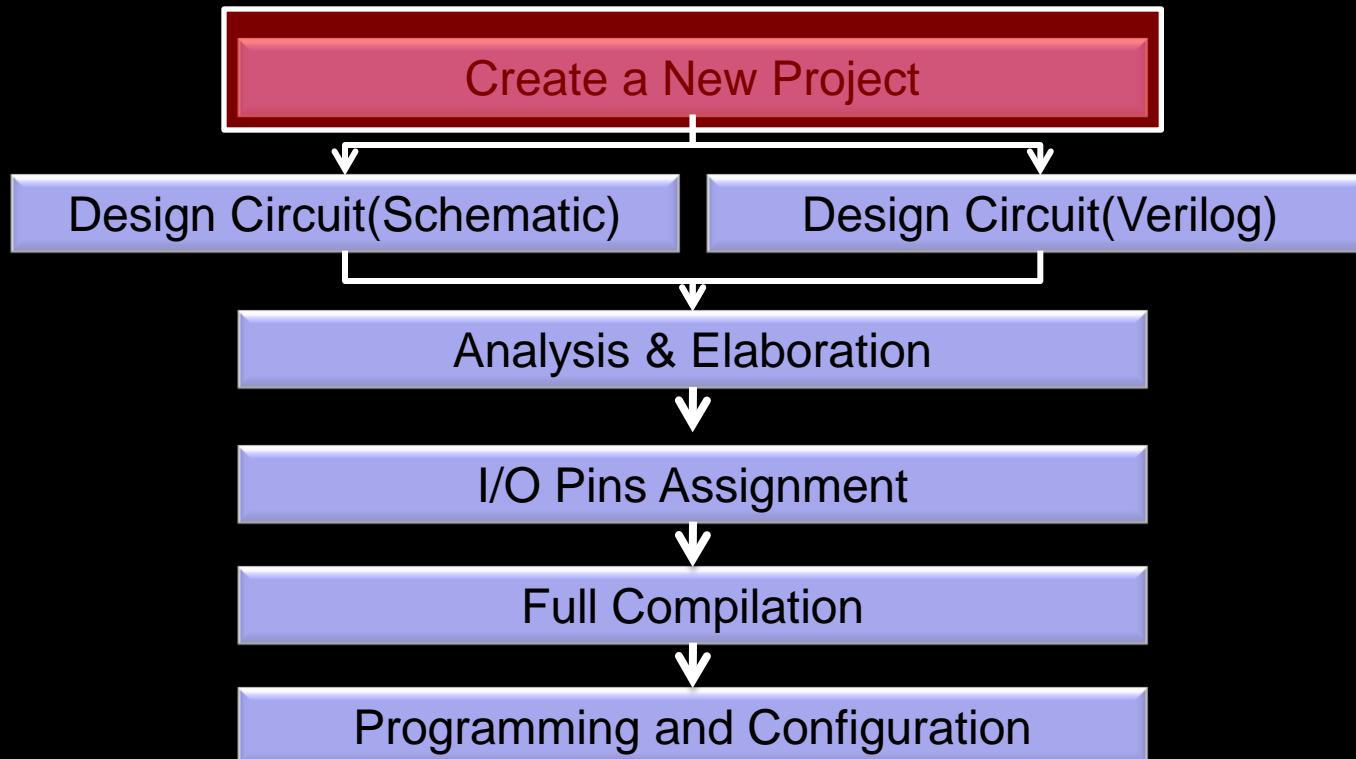
Verification of Button



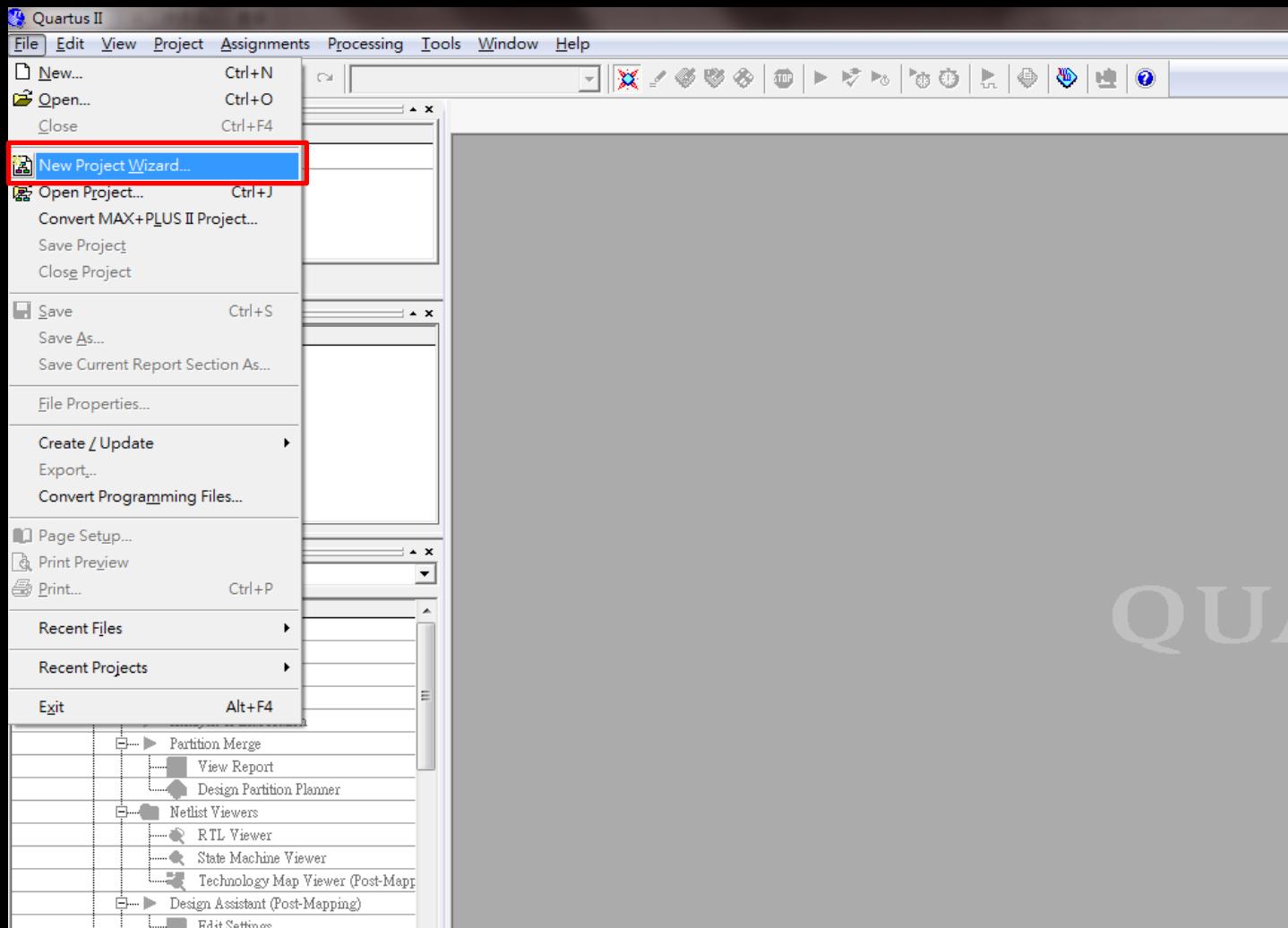
Outline

- **Introduction - Altera DE0 FPGA Board**
- **FPGA Design with Quartus II**
 - Lab1: Schematic Design Flow
 - Lab2: Dip switch to LED
 - Lab3: Dip switch to 7-seg (Binary to Decimal)

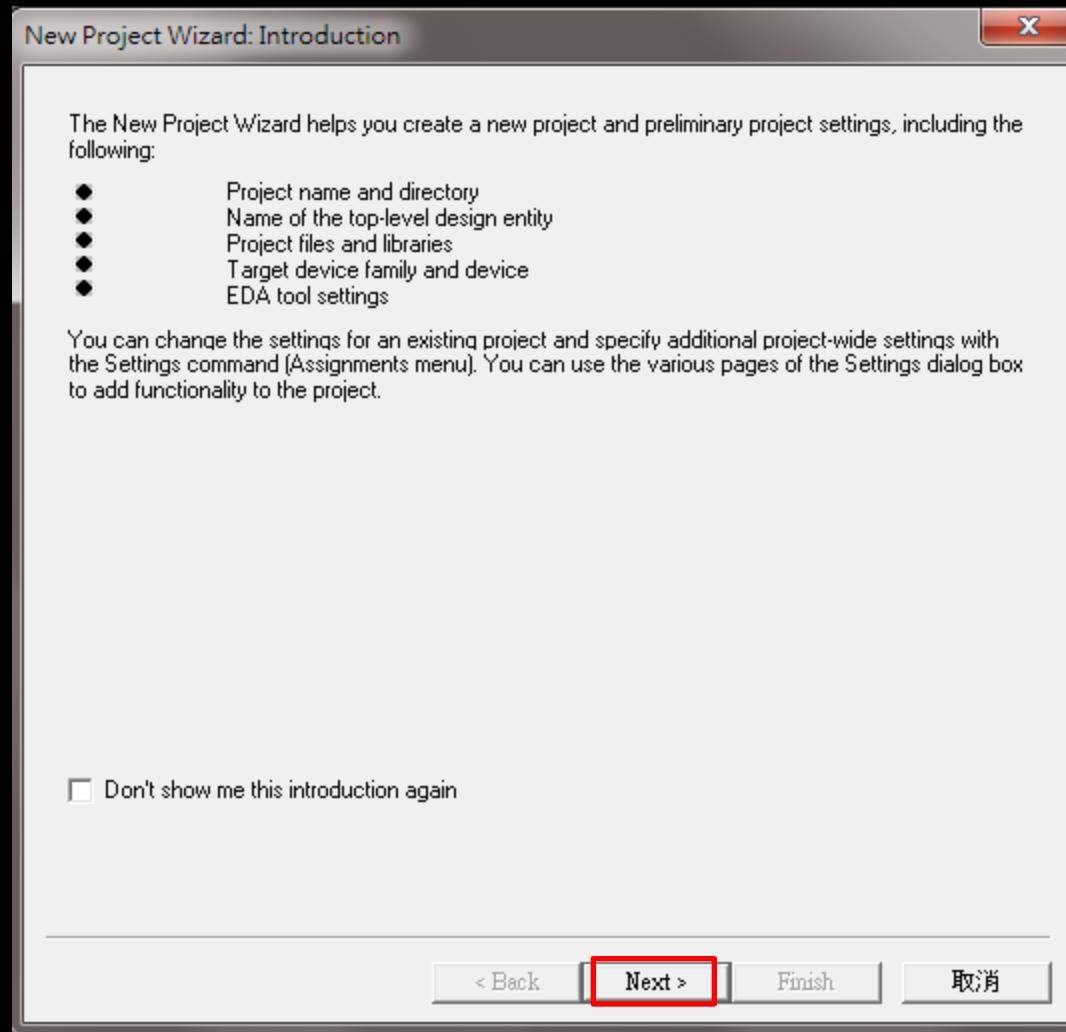
Flow of FPGA Design with Quartus II



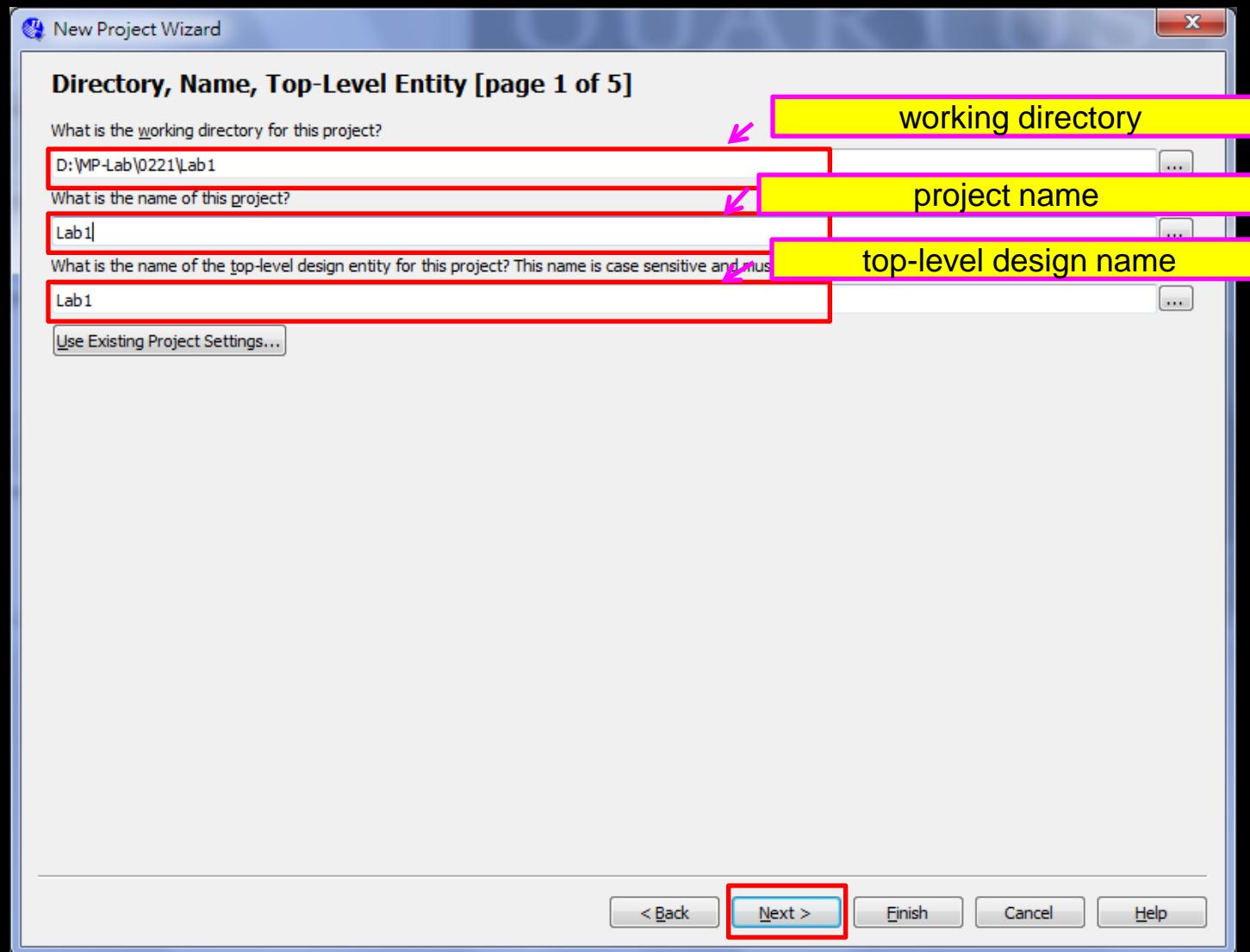
Creating a New Project



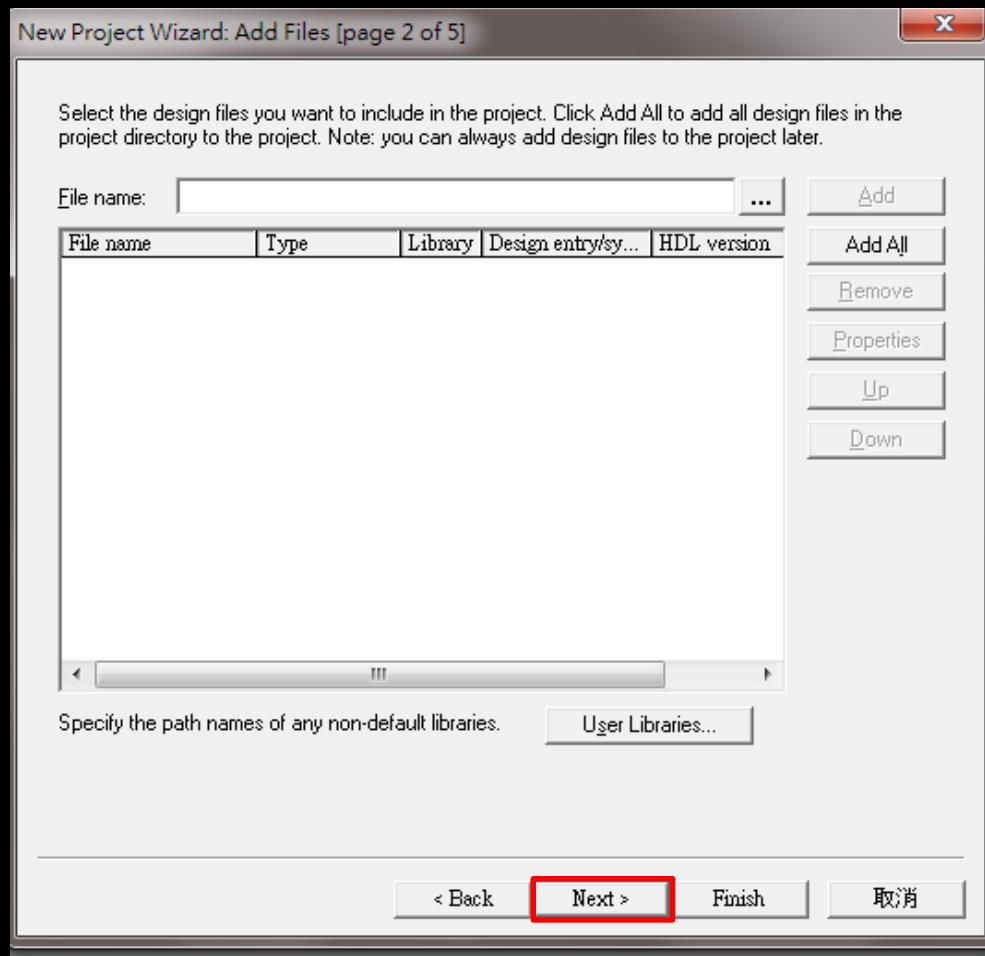
New Project Wizard: Introduction



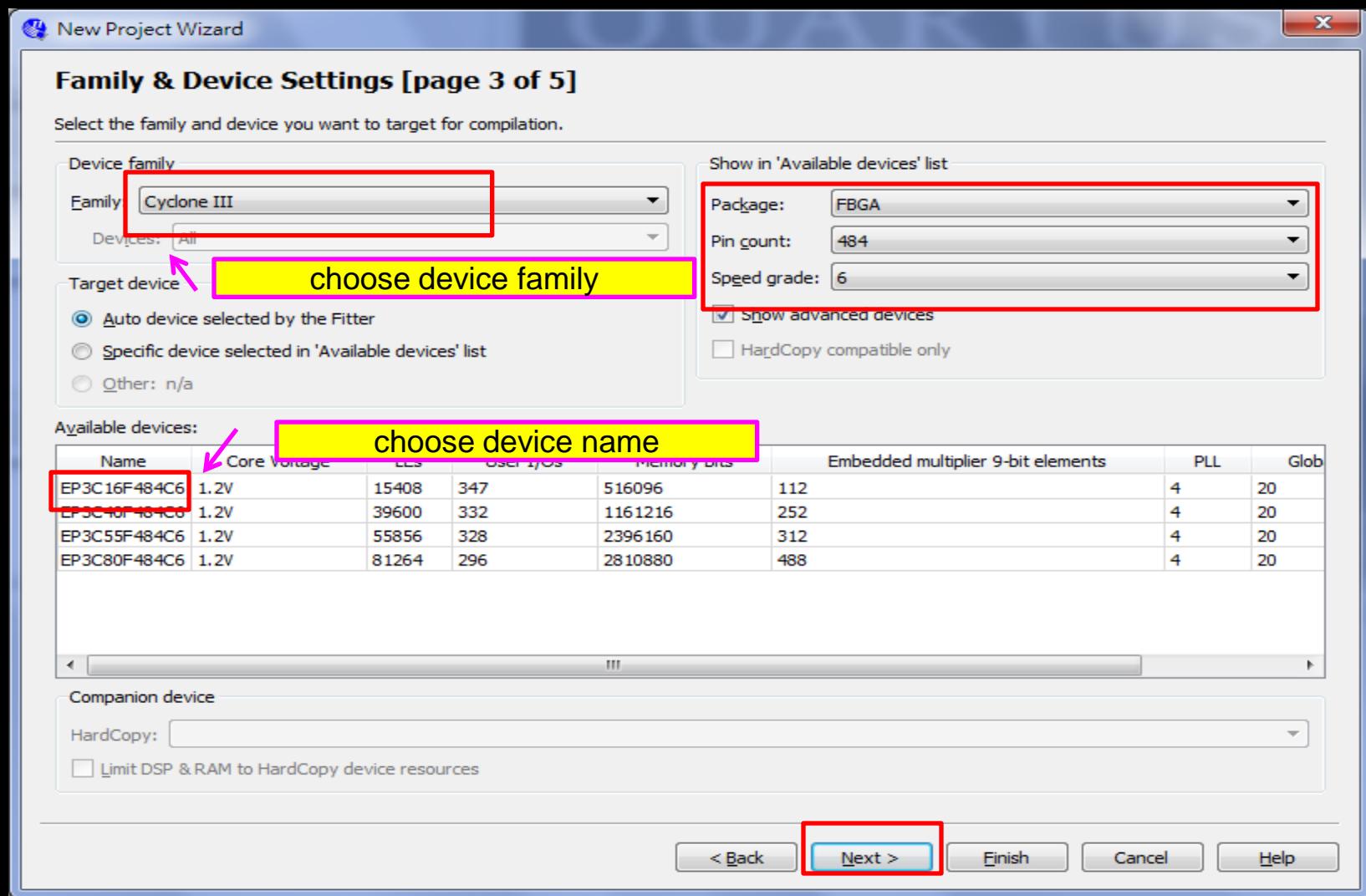
New Project Wizard (1/5)



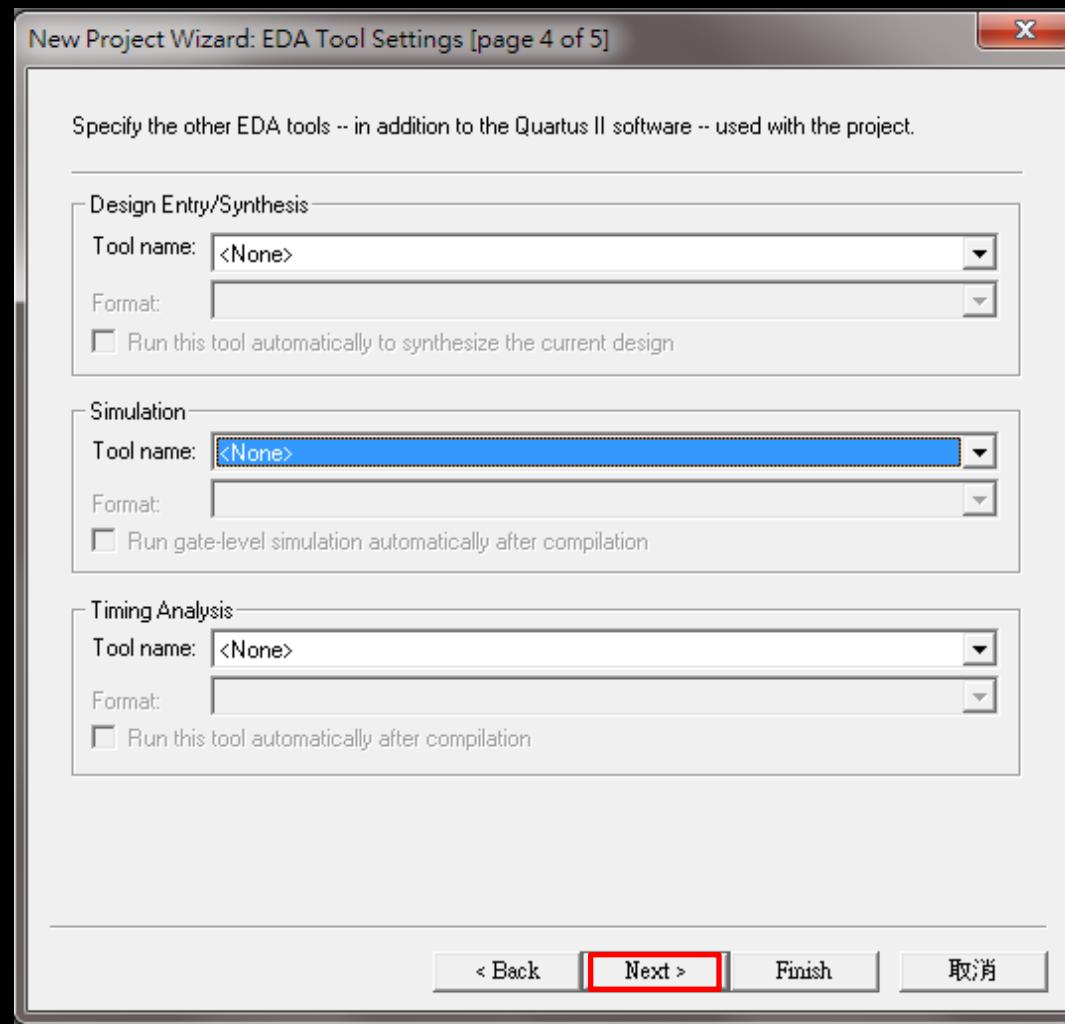
New Project Wizard (2/5)



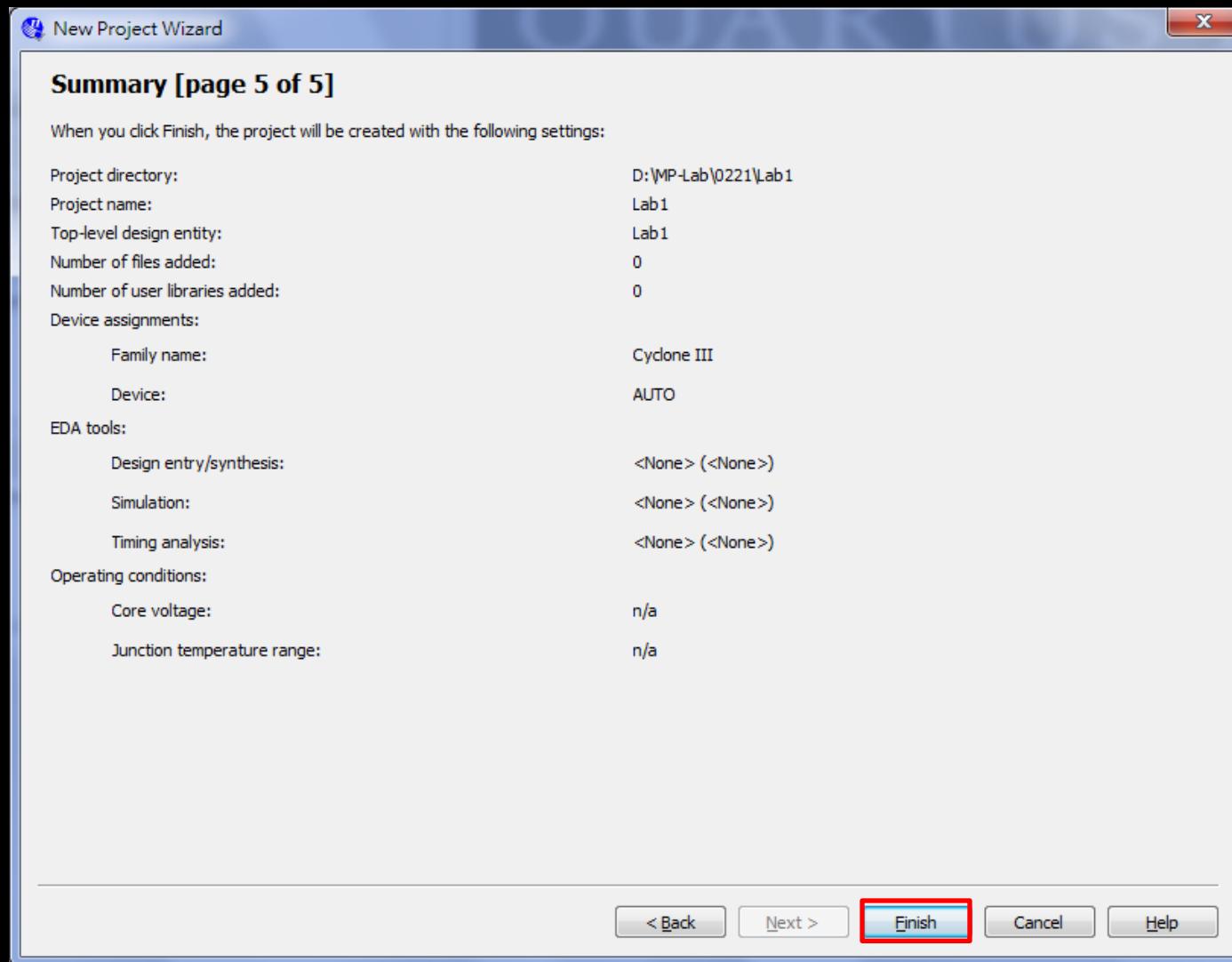
New Project Wizard (3/5)



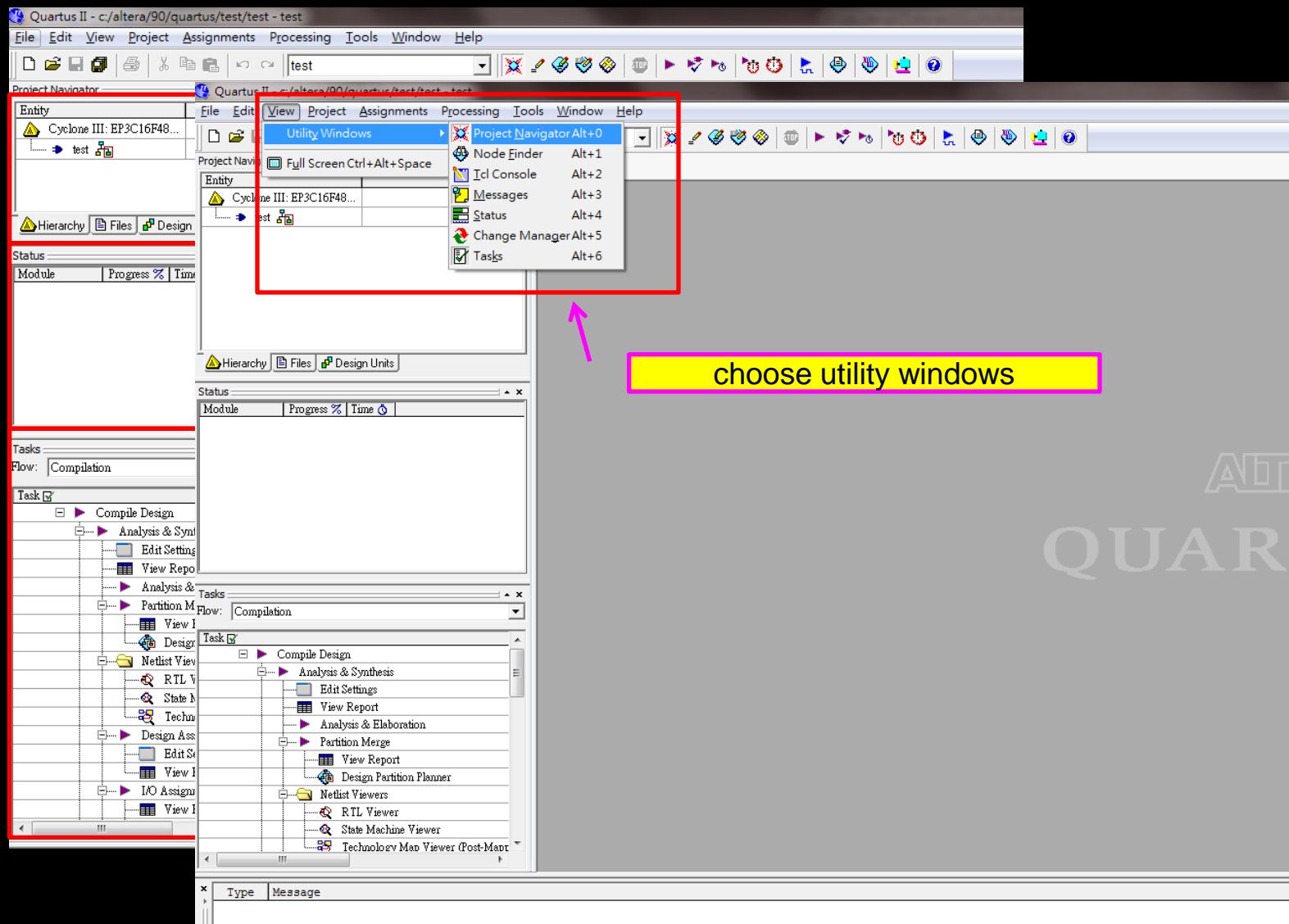
New Project Wizard (4/5)



New Project Wizard (5/5)

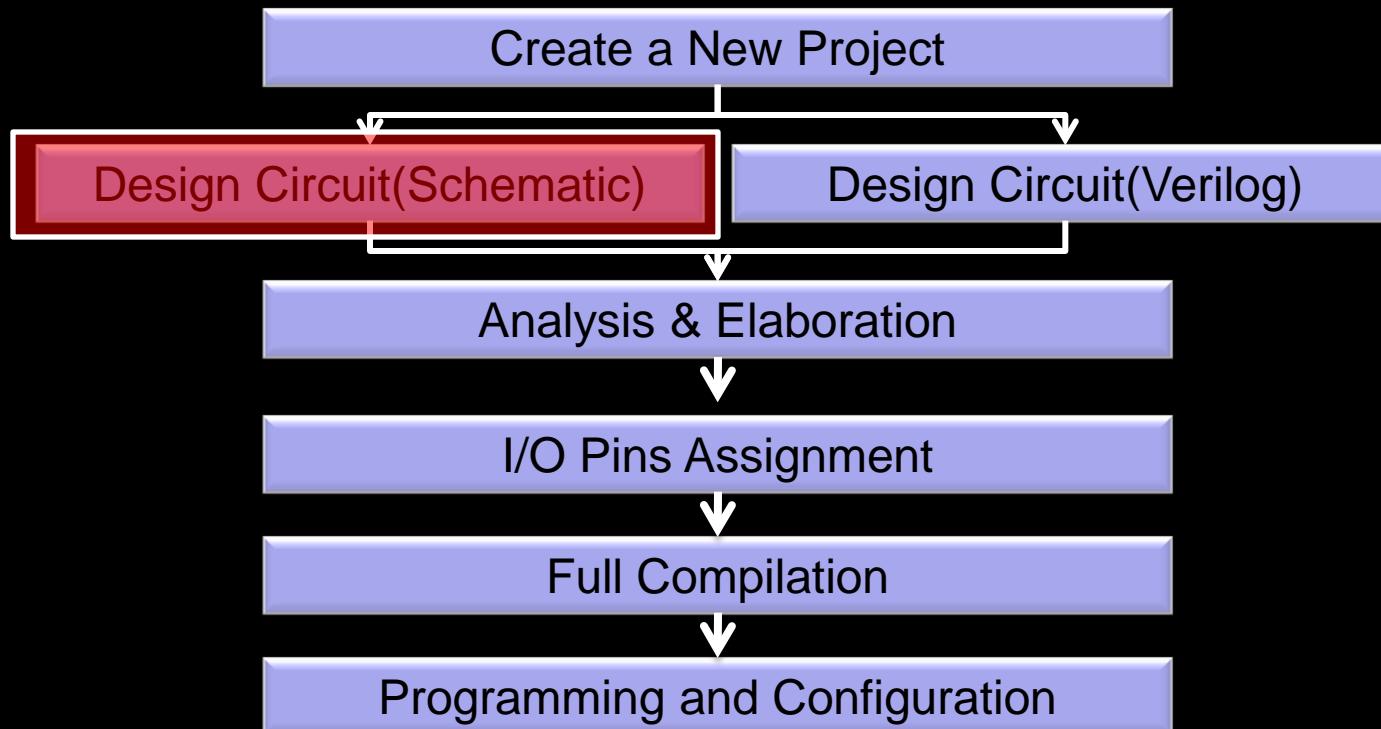


View Utility

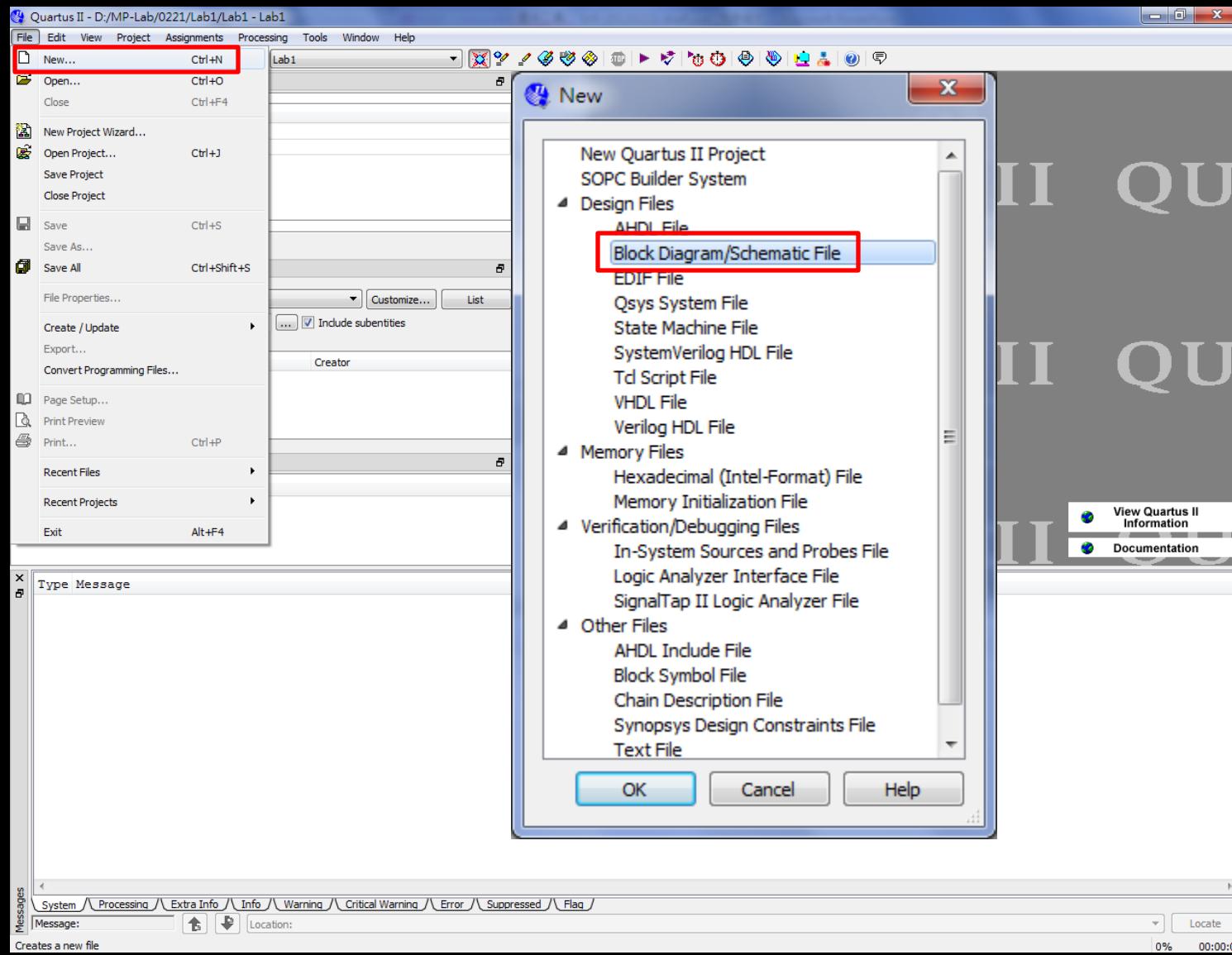


choose utility windows

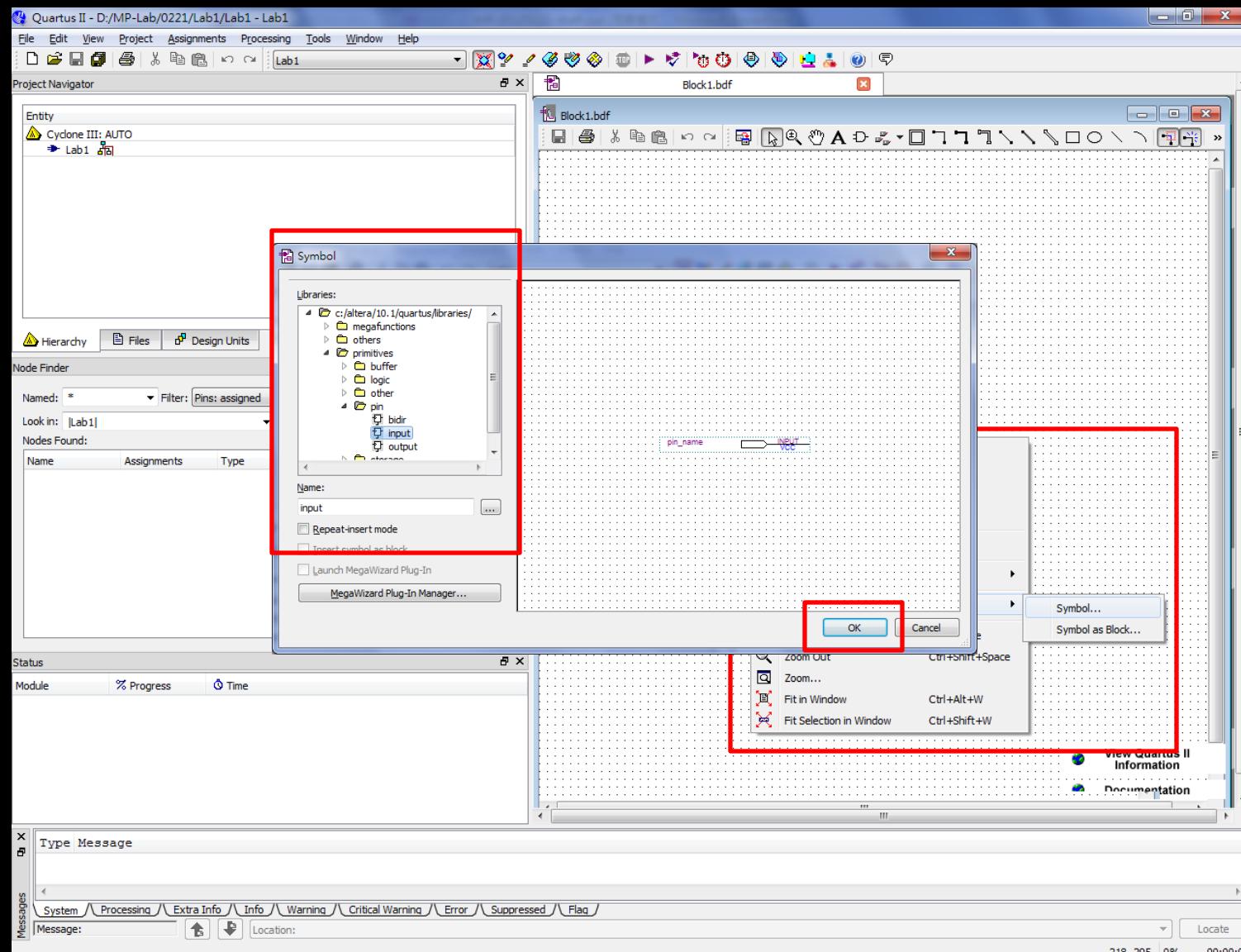
Flow of FPGA Design with Quartus II



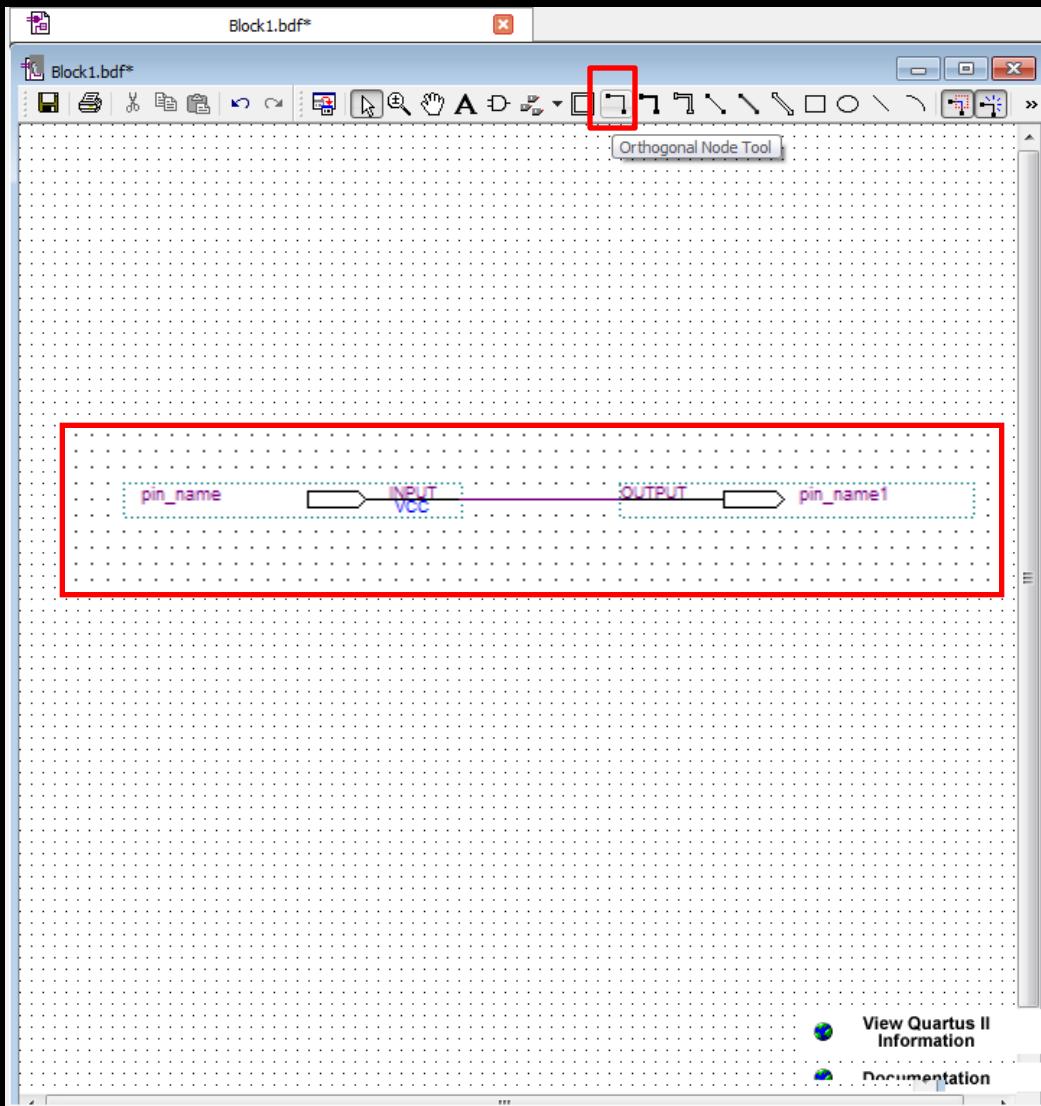
Lab1: Create a Schematic File



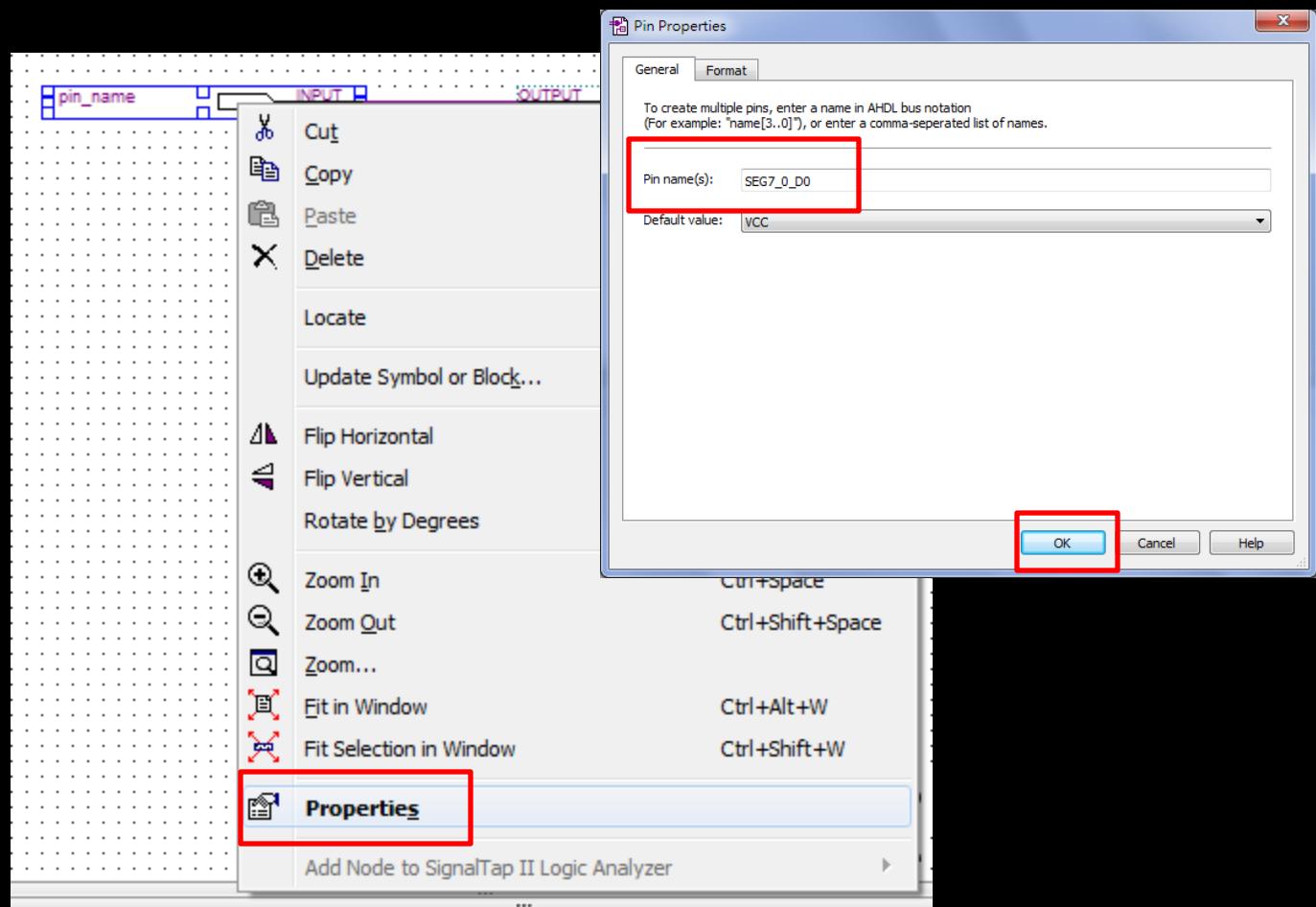
Insert Symbols



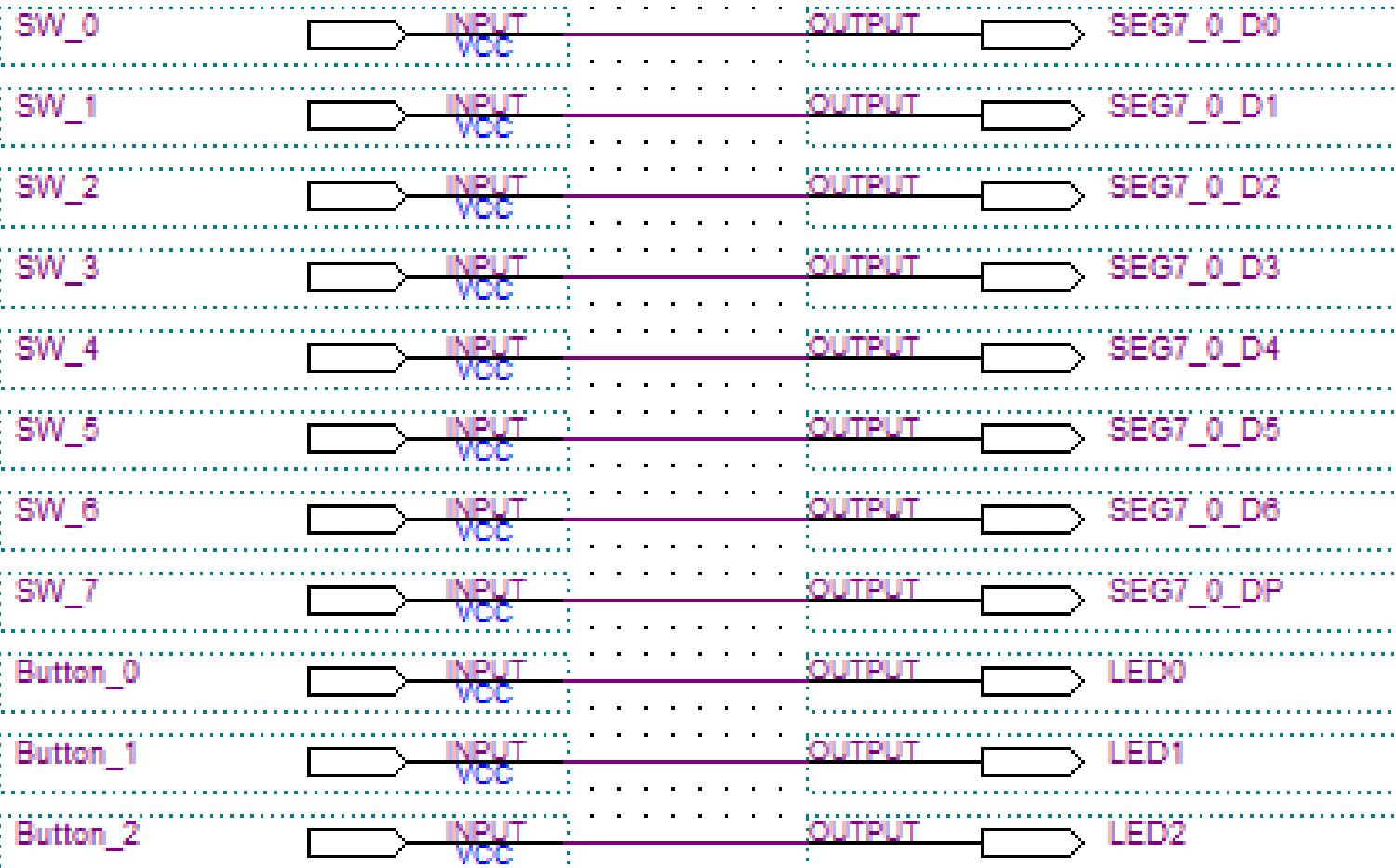
Add Wires



Modify Pin Names



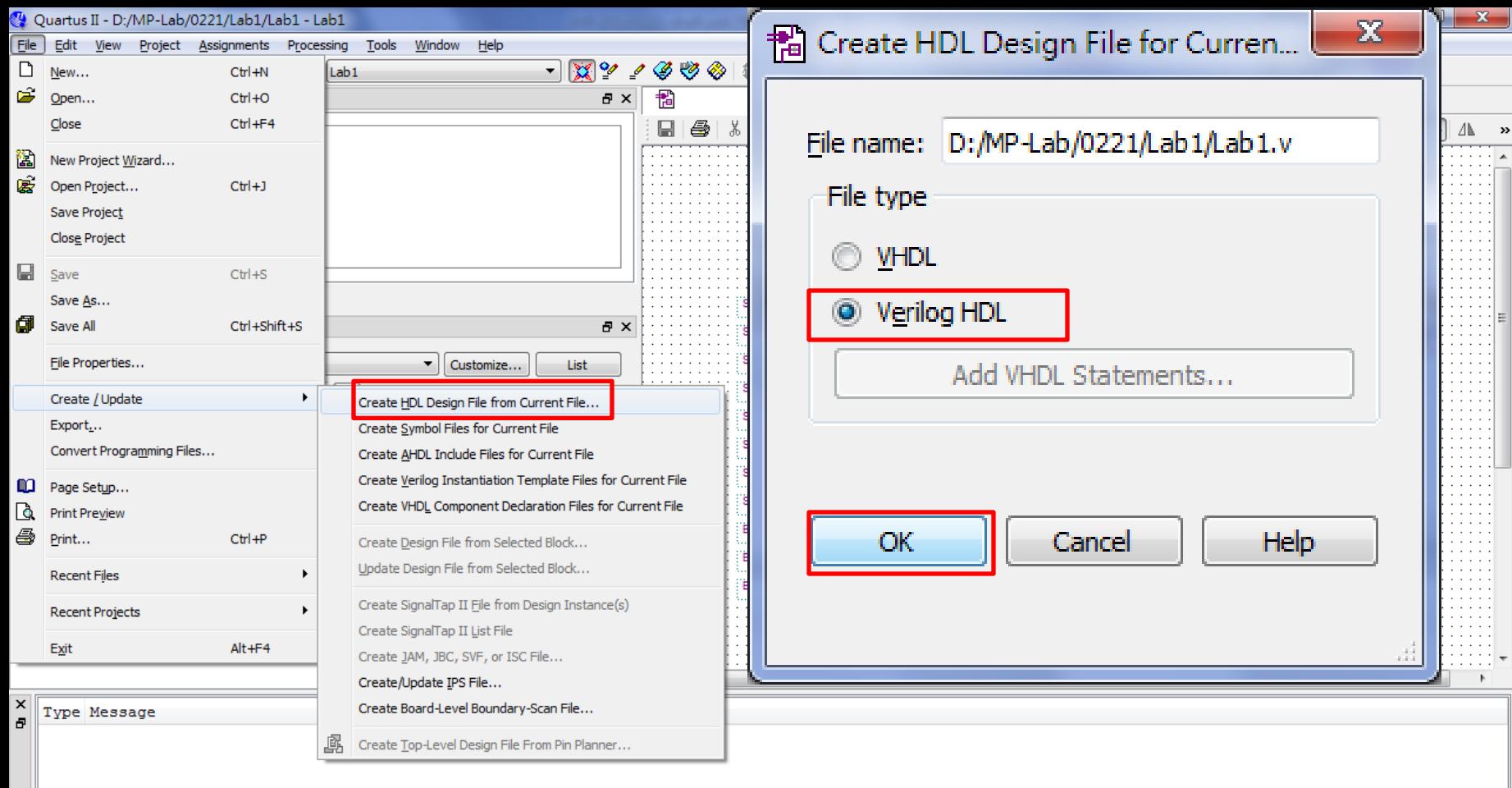
Schematic Design Result



Schematic Design Result (cont.)



TIP: Transfereing Schematic to Verilog Code



TIP: Transfereing Schematic to Verilog Code (cont.)

The screenshot shows the Quartus II software interface with the following windows:

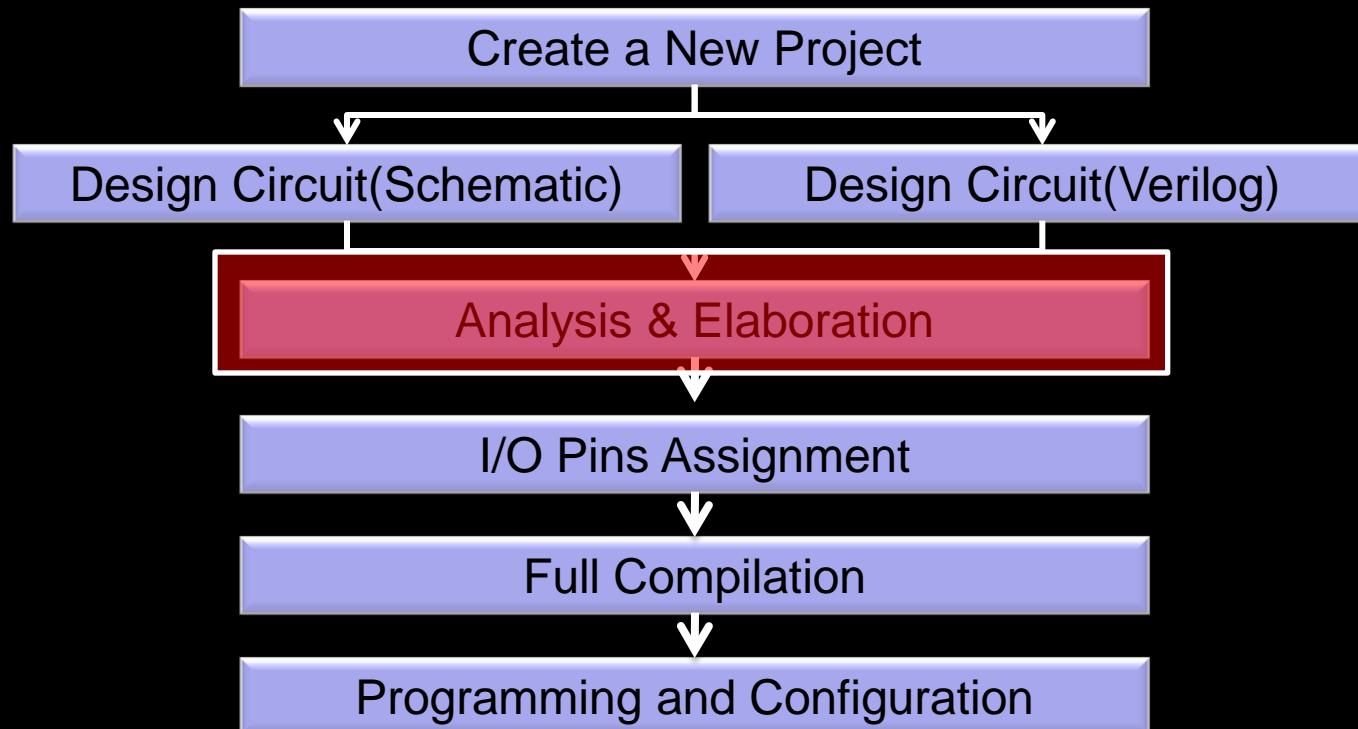
- Project Navigator:** Shows a folder named "Lab1.bdf".
- Node Finder:** Set to look in "Lab1" and filter by "Pins: assigned".
- Status:** Shows a progress bar at 100% completion.
- Editor:** Displays Verilog code for module Lab1. A red box highlights the top portion of the code, which contains copyright and program information.

```
// Copyright (C) 1991-2010 Altera Corporation
// Your use of Altera Corporation's design tools, logic functions
// and other software and tools, and its AMPP partner logic
// functions, and any output files from any of the foregoing
// (including device programming or simulation files), and any
// associated documentation or information are expressly subject
// to the terms and conditions of the Altera Program License
// Subscription Agreement, Altera MegaCore Function License
// Agreement, or other applicable license agreement, including,
// without limitation, that your use is for the sole purpose of
// programming logic devices manufactured by Altera and sold by
// Altera or its authorized distributors. Please refer to the
// applicable agreement for further details.

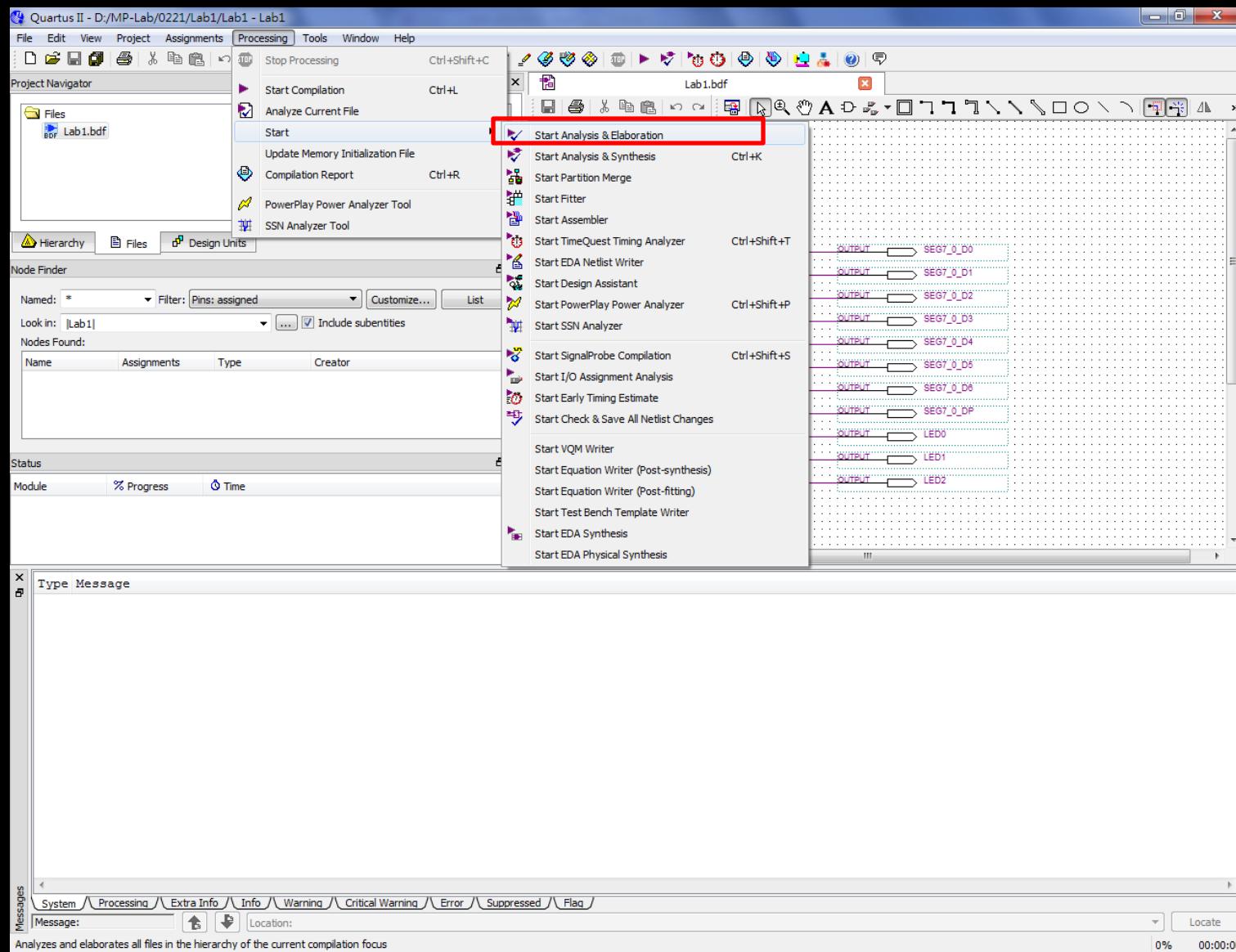
// PROGRAM      "Quartus II"
// VERSION      "Version 10.1 Build 153 11/29/2010 SJ Full Version"
// CREATED      "Mon Feb 20 00:04:17 2012"

module Lab1(
    SW_0,
    SW_1,
    SW_2,
    SW_3,
    SW_4,
    SW_5,
    SW_6,
    SW_7,
    Button 0.
```

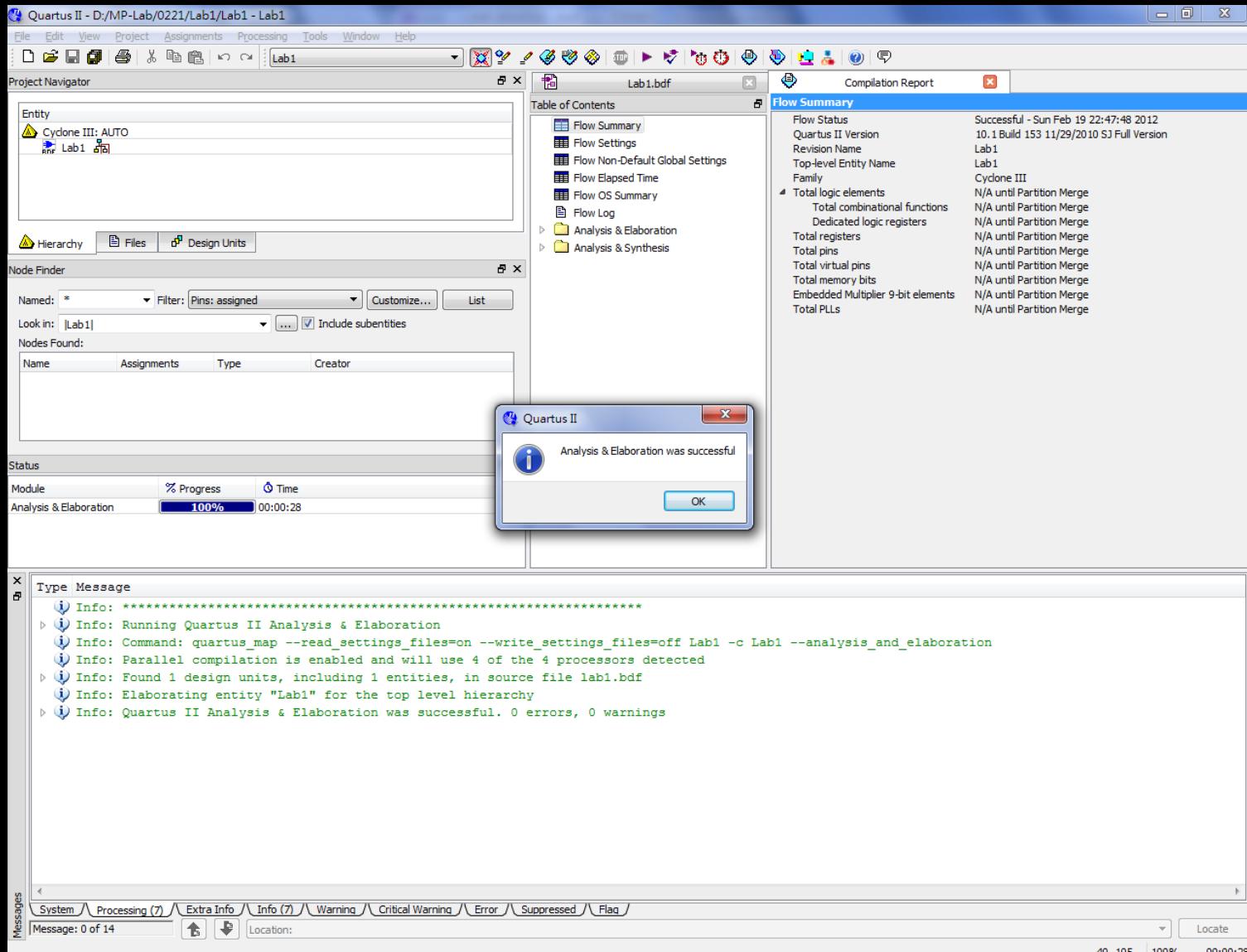
Flow of FPGA Design with Quartus II



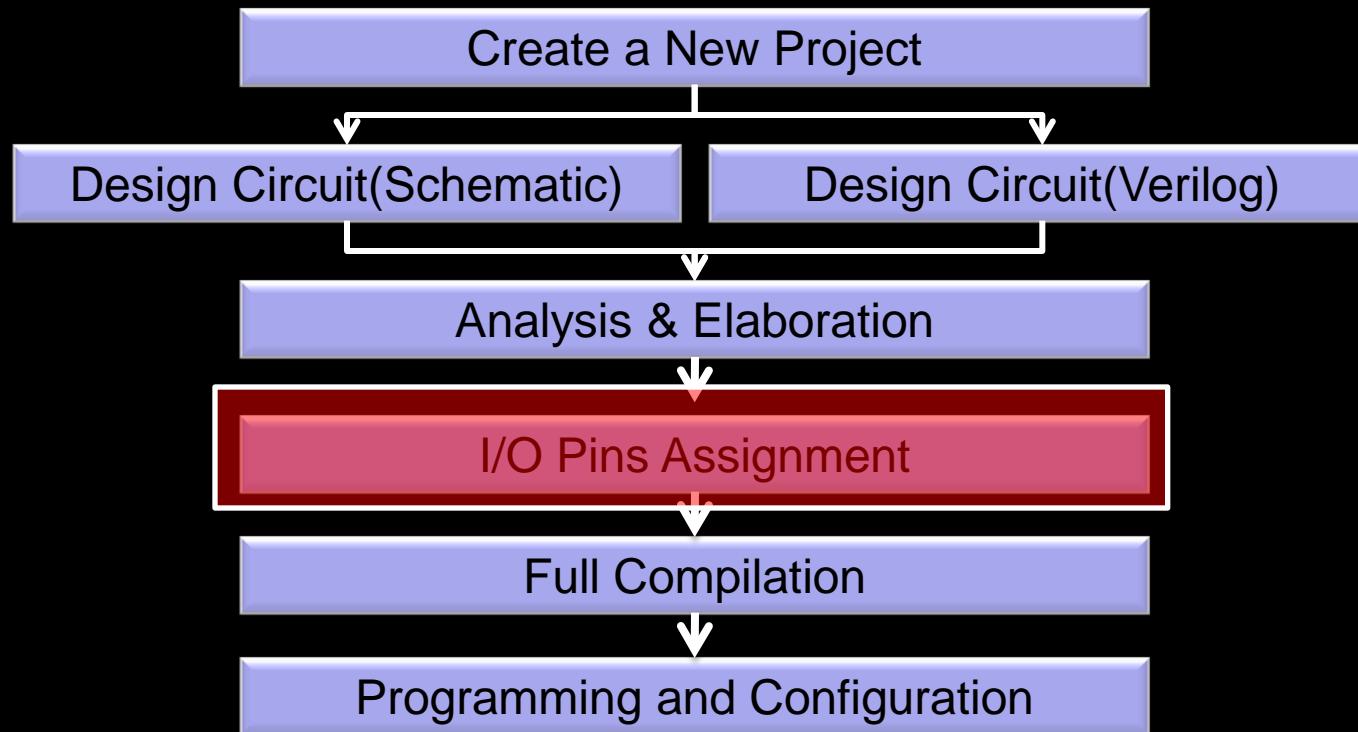
Analysis & Elaboration



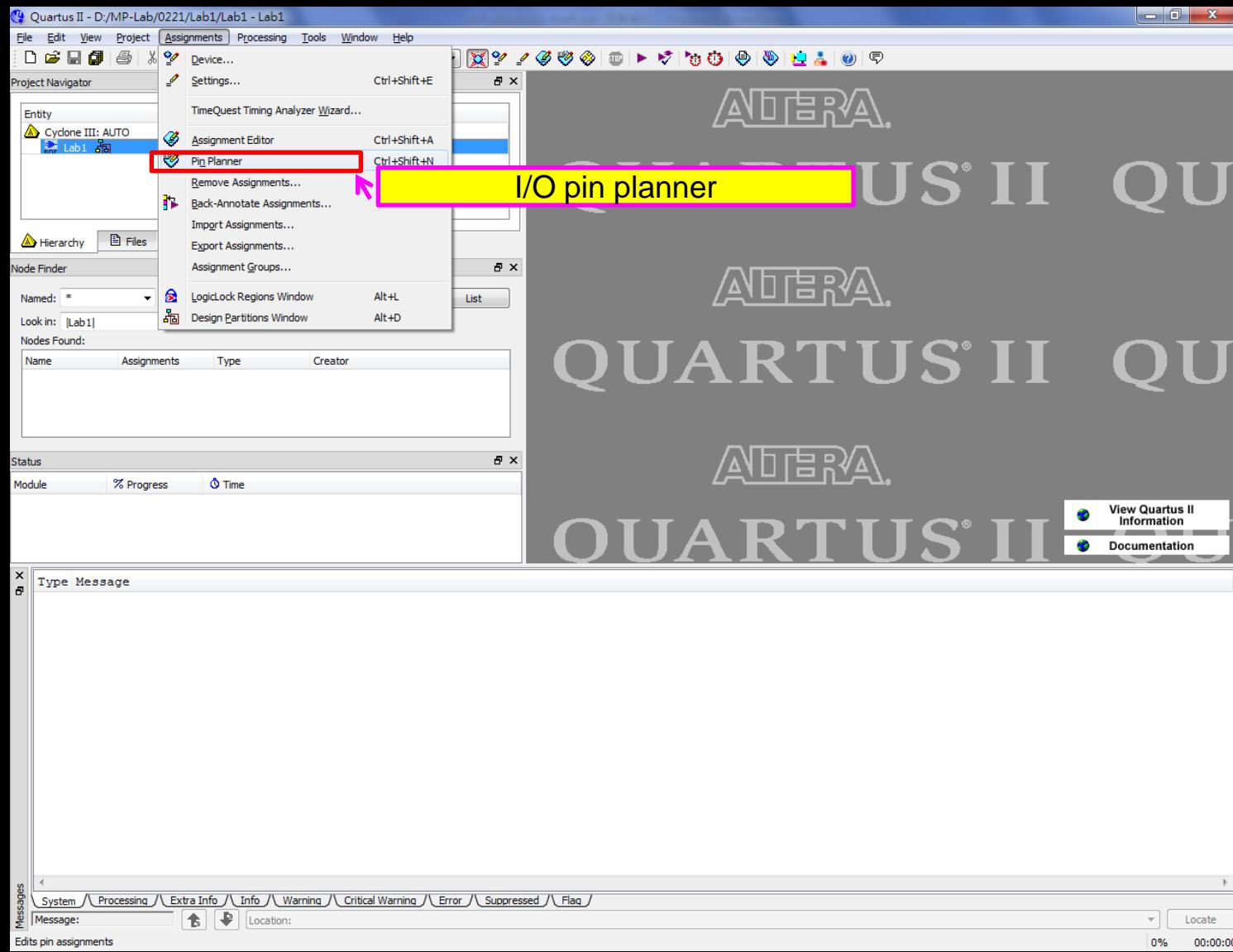
Success of Analysis & Elaboration



Flow of FPGA Design with Quartus II



Pin Planner



Assign I/O Pins

Pin Planner - D:/MP-Lab/0221/Lab1/Lab1 - Lab1

File Edit View Processing Tools Window

Groups
Named: * <<new group>>

Node Name Direction Location

Top View - Wire Bond
Cyclone III - EP3C16F484C6

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

A B C D E F G H J K L M N P R T U V W Y AA AB

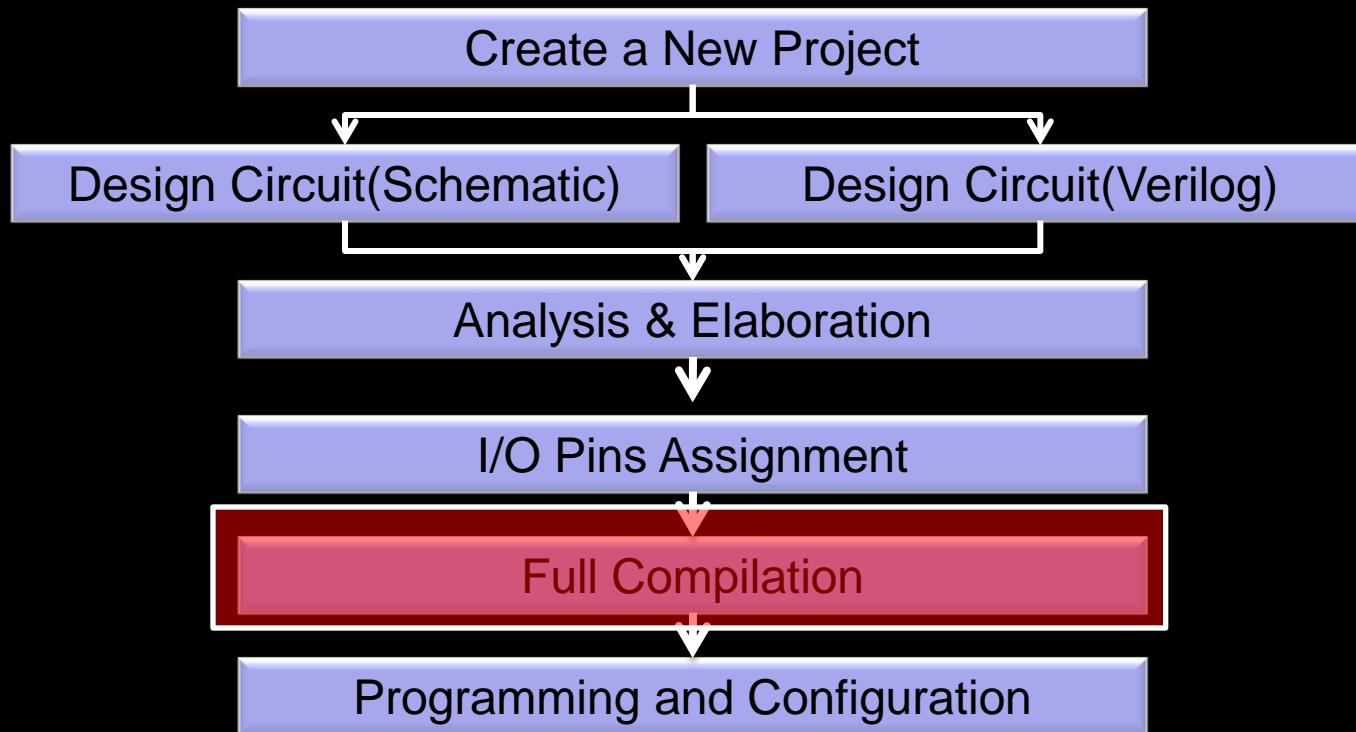
See user manual for pin assignment

Node Name	Direction	Location	I/O Standard	Reserved
Button_0	Input		2.5 V (default)	
Button_1	Input		2.5 V (default)	
Button_2	Input		2.5 V (default)	
LED0	Output		2.5 V (default)	
LED1	Output		2.5 V (default)	
LED2	Output		2.5 V (default)	
SEG7_0_D0	Output		2.5 V (default)	
SEG7_0_D1	Output		2.5 V (default)	
SEG7_0_D2	Output		2.5 V (default)	
SEG7_0_D3	Output		2.5 V (default)	
SEG7_0_D4	Output		2.5 V (default)	
SEG7_0_D5	Output		2.5 V (default)	
SEG7_0_D6	Output		2.5 V (default)	
SEG7_0_DP	Output		2.5 V (default)	
SW_0	Input		2.5 V (default)	
SW_1	Input		2.5 V (default)	
SW_2	Input		2.5 V (default)	
SW_3	Input		2.5 V (default)	
SW_4	Input		2.5 V (default)	
SW_5	Input		2.5 V (default)	

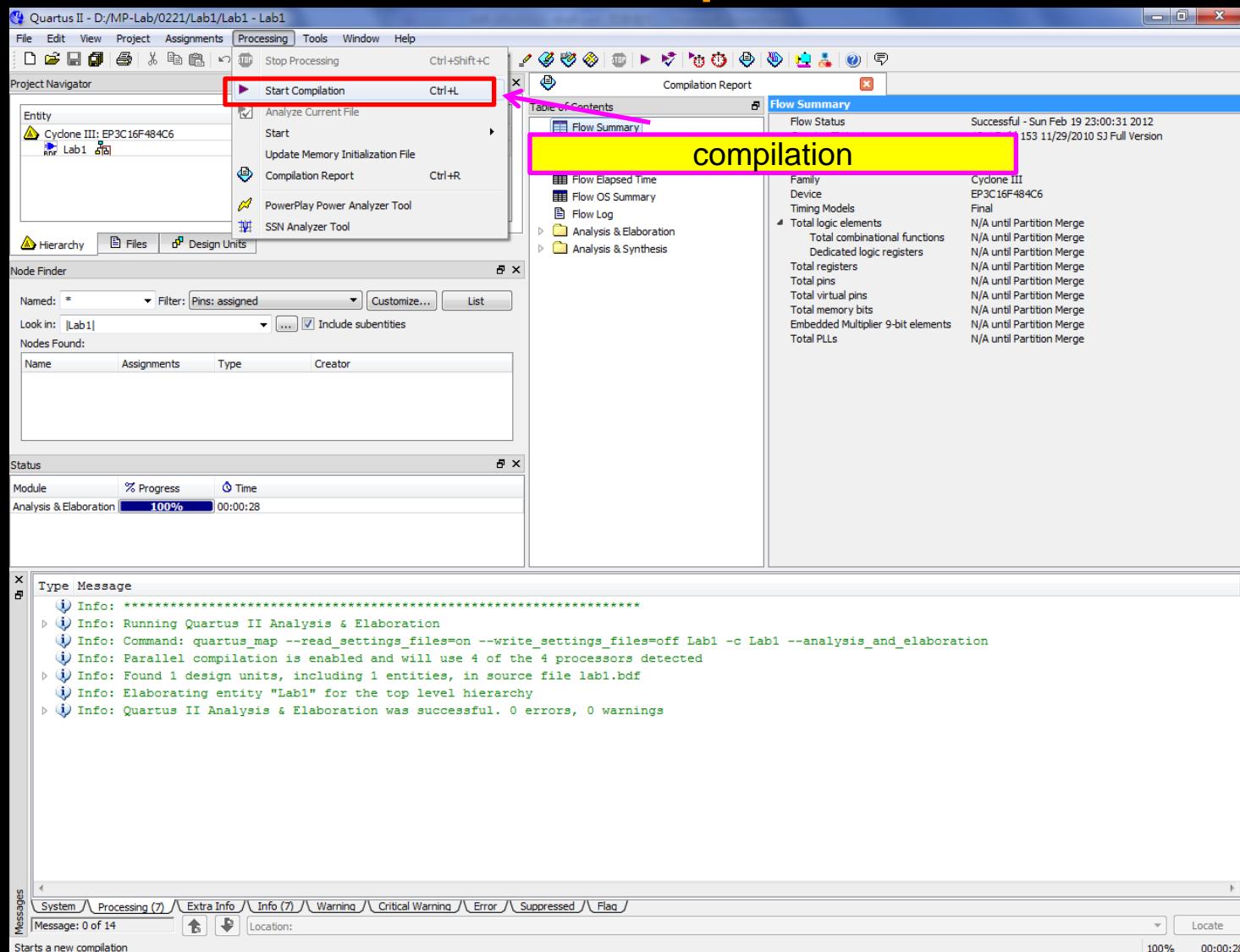
0% 00:00:00

44/62

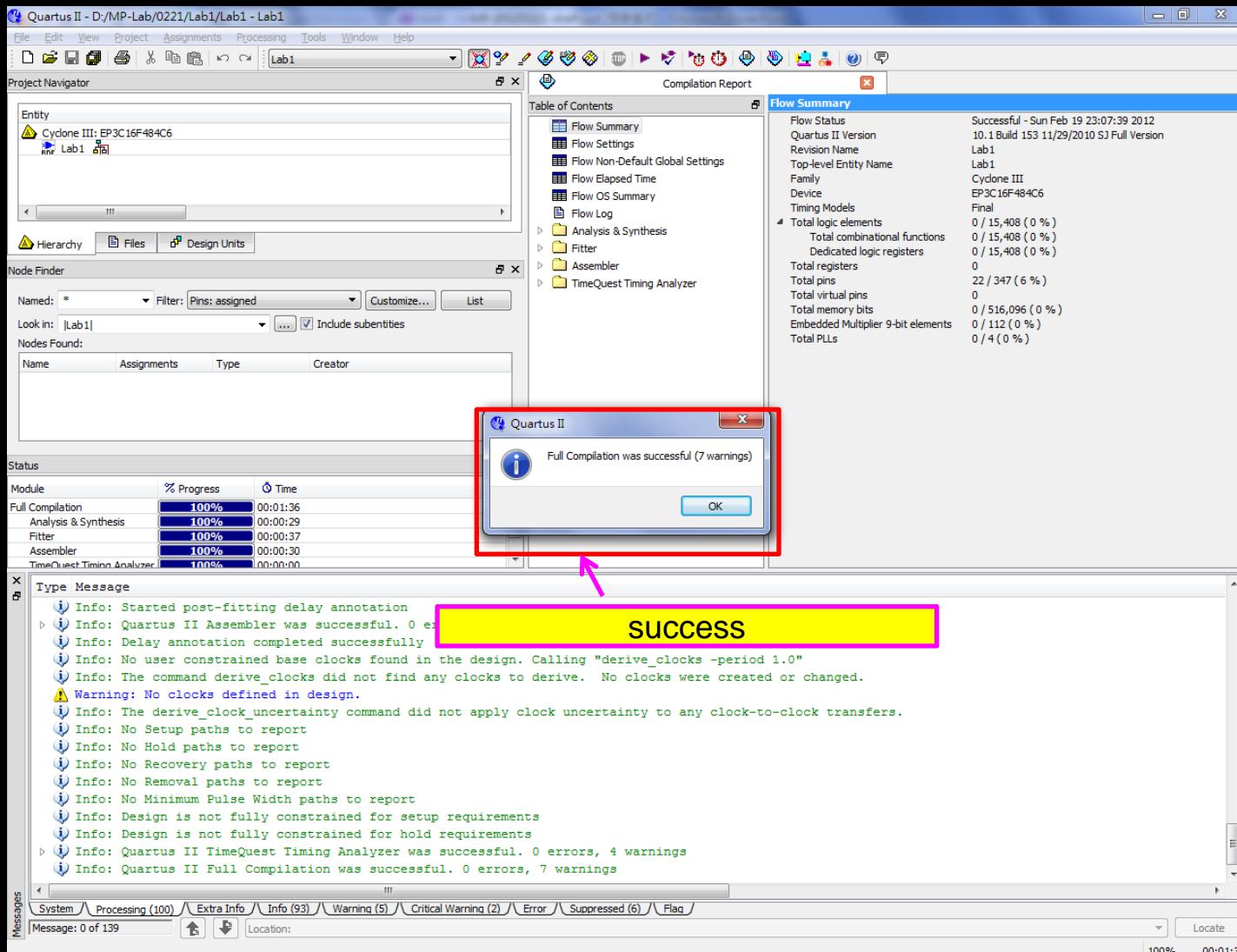
Flow of FPGA Design with Quartus II



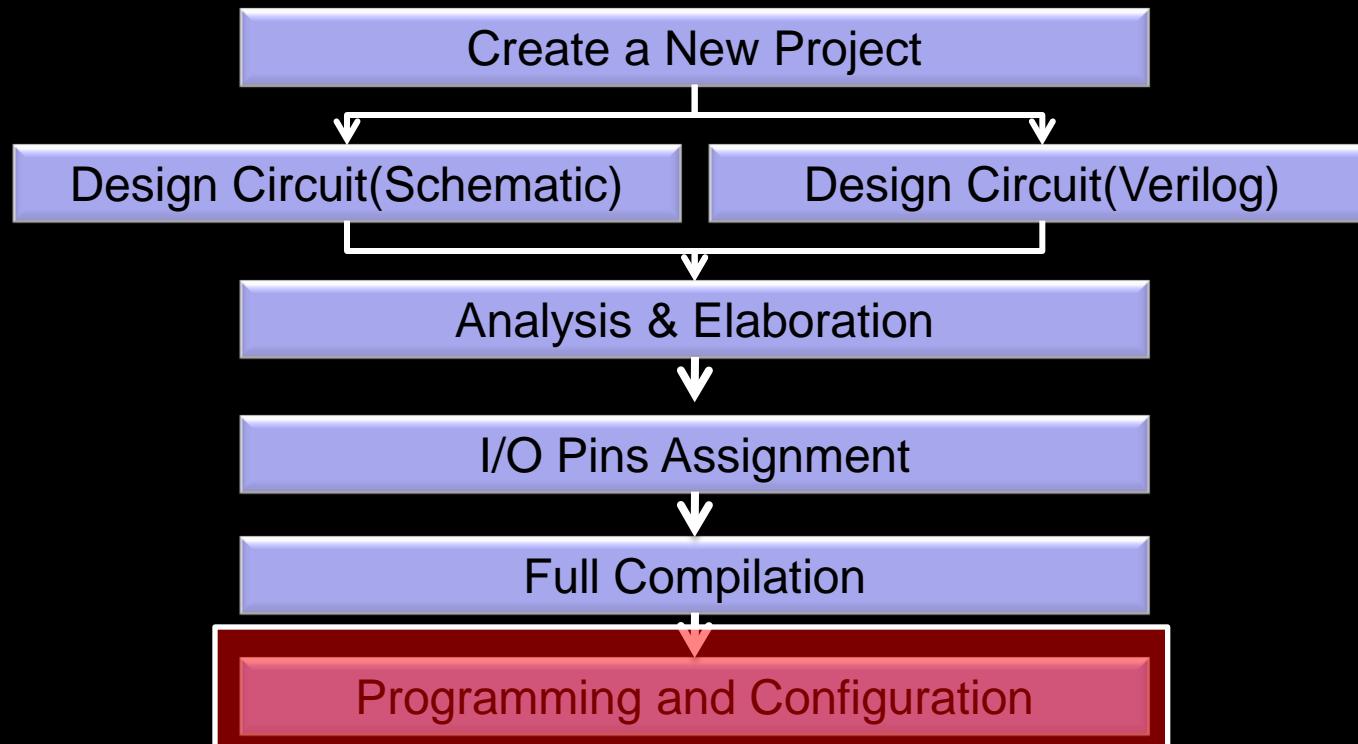
Start Compilation



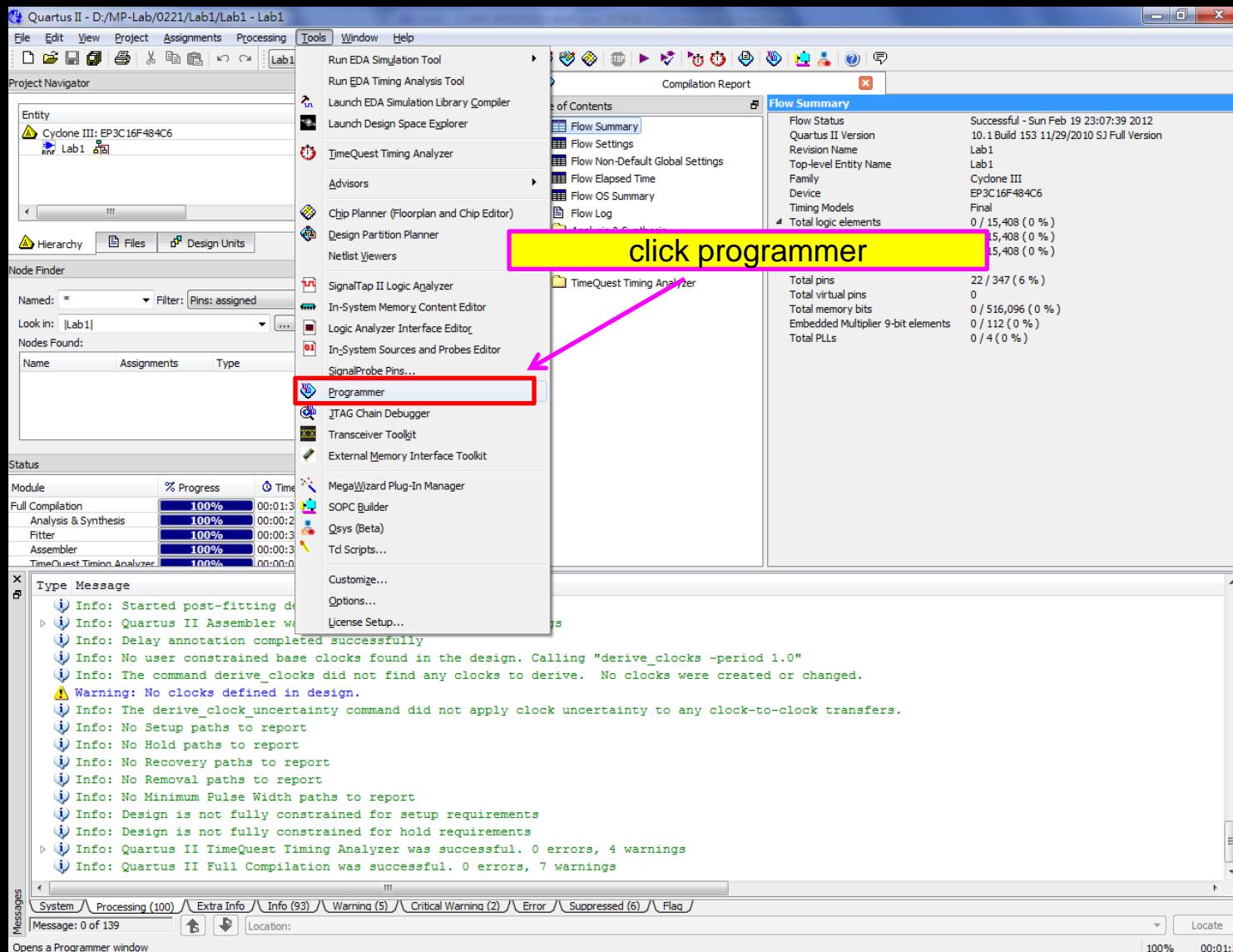
Success of Compilation



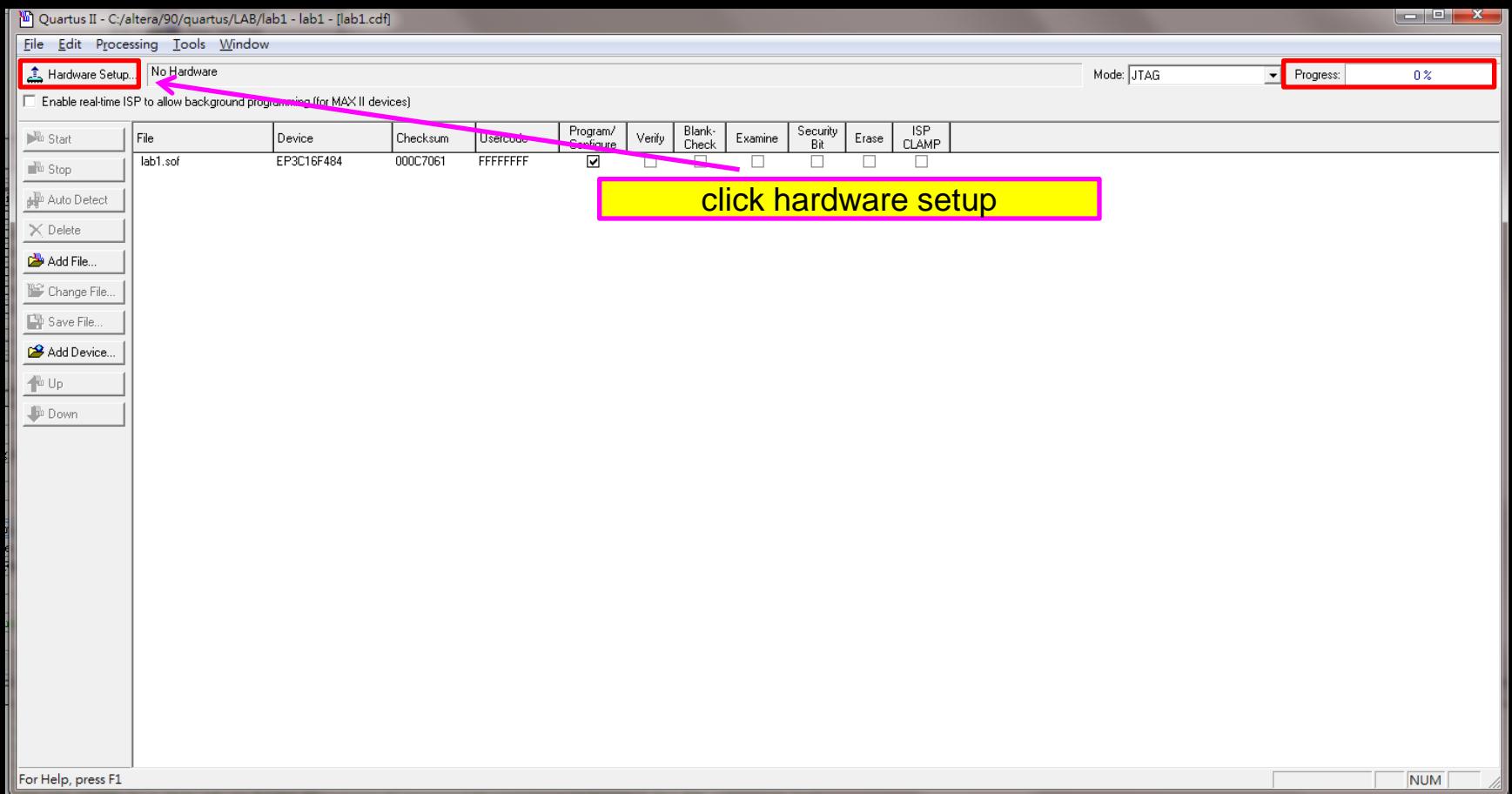
Flow of FPGA Design with Quartus II



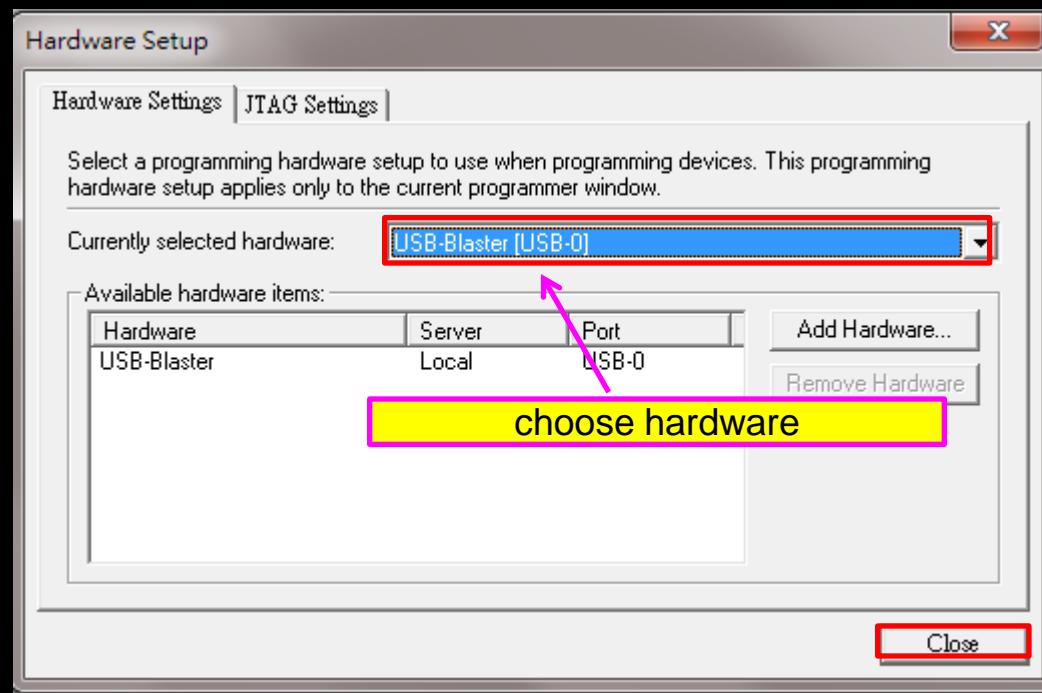
Programmer



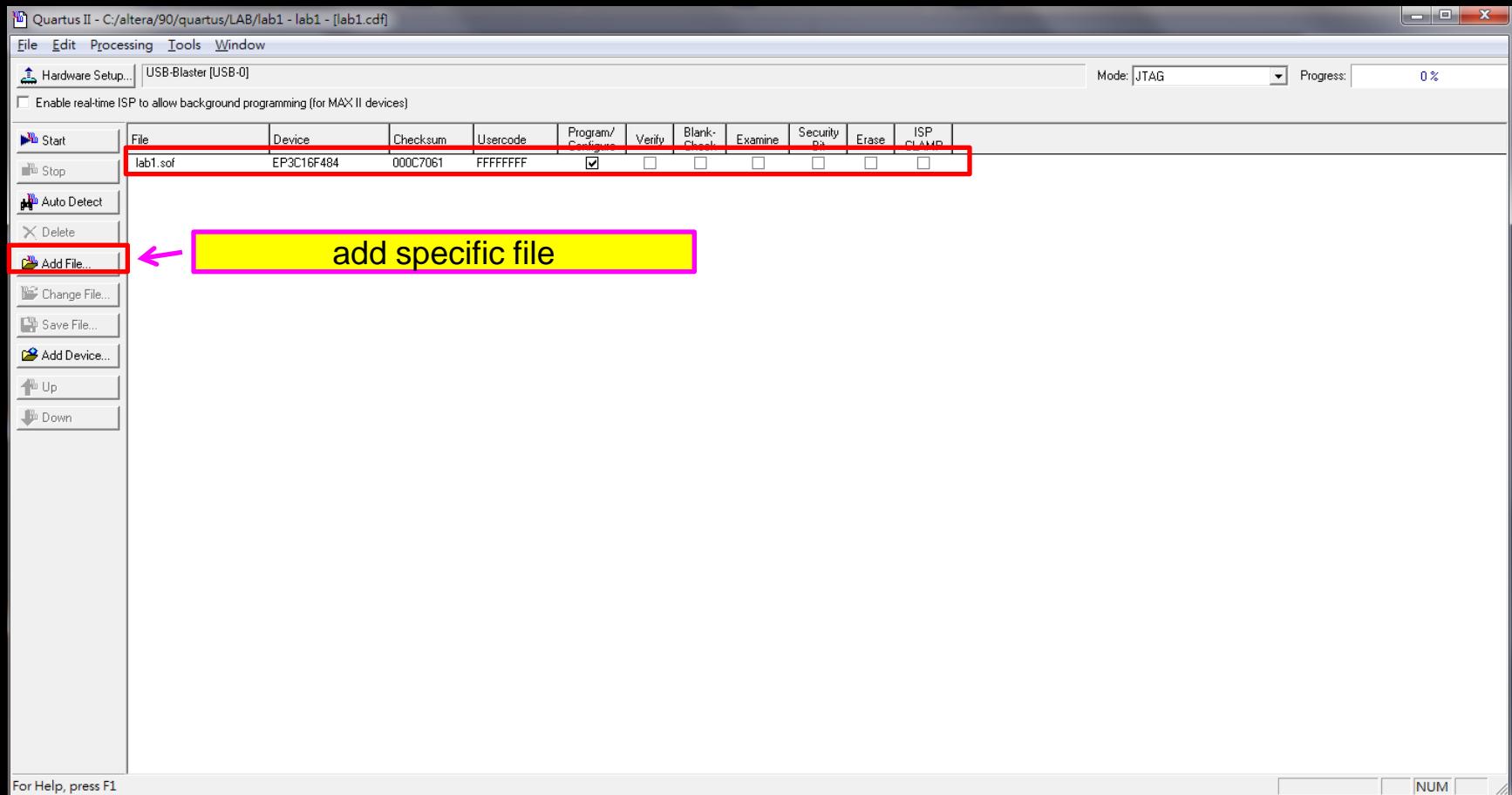
Hardware Setup



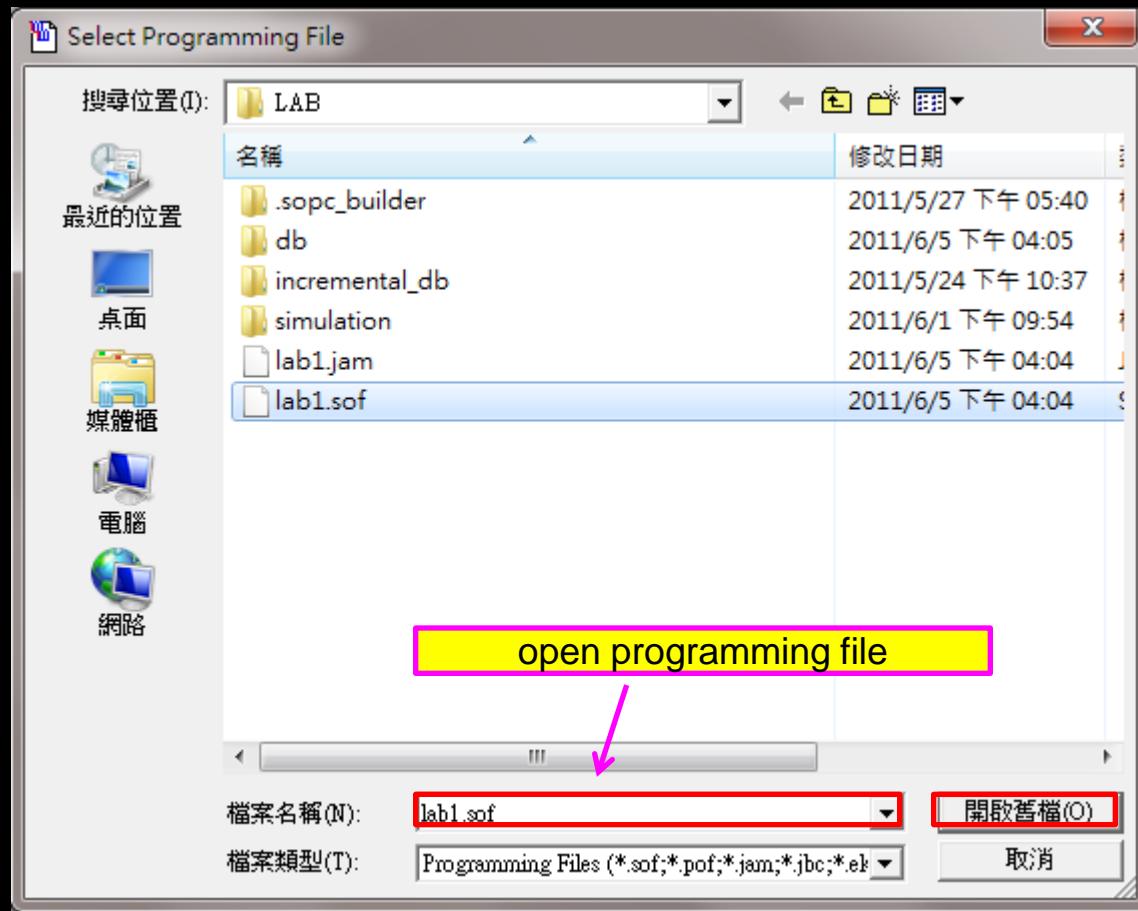
Add Hardware



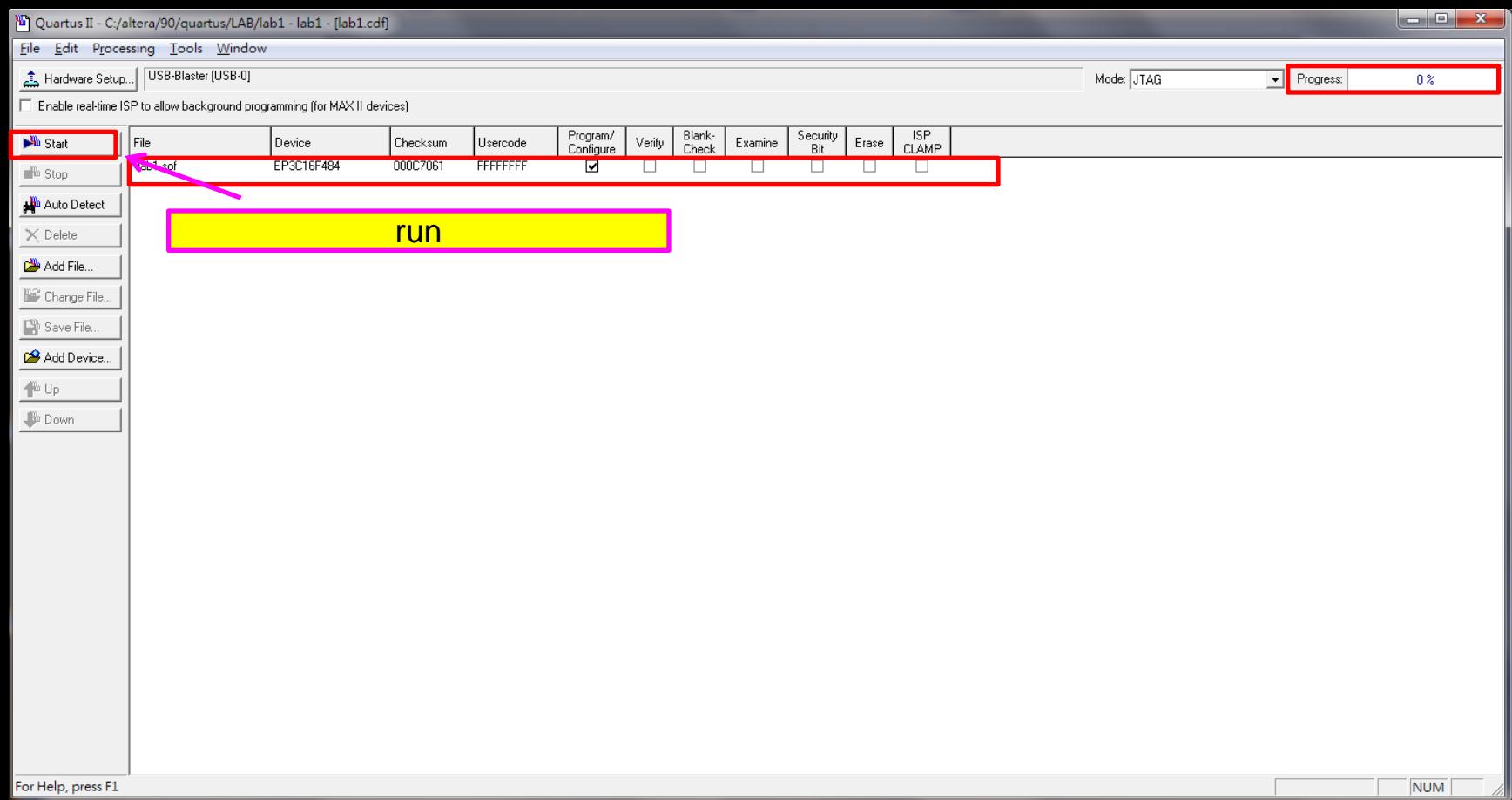
Design Window



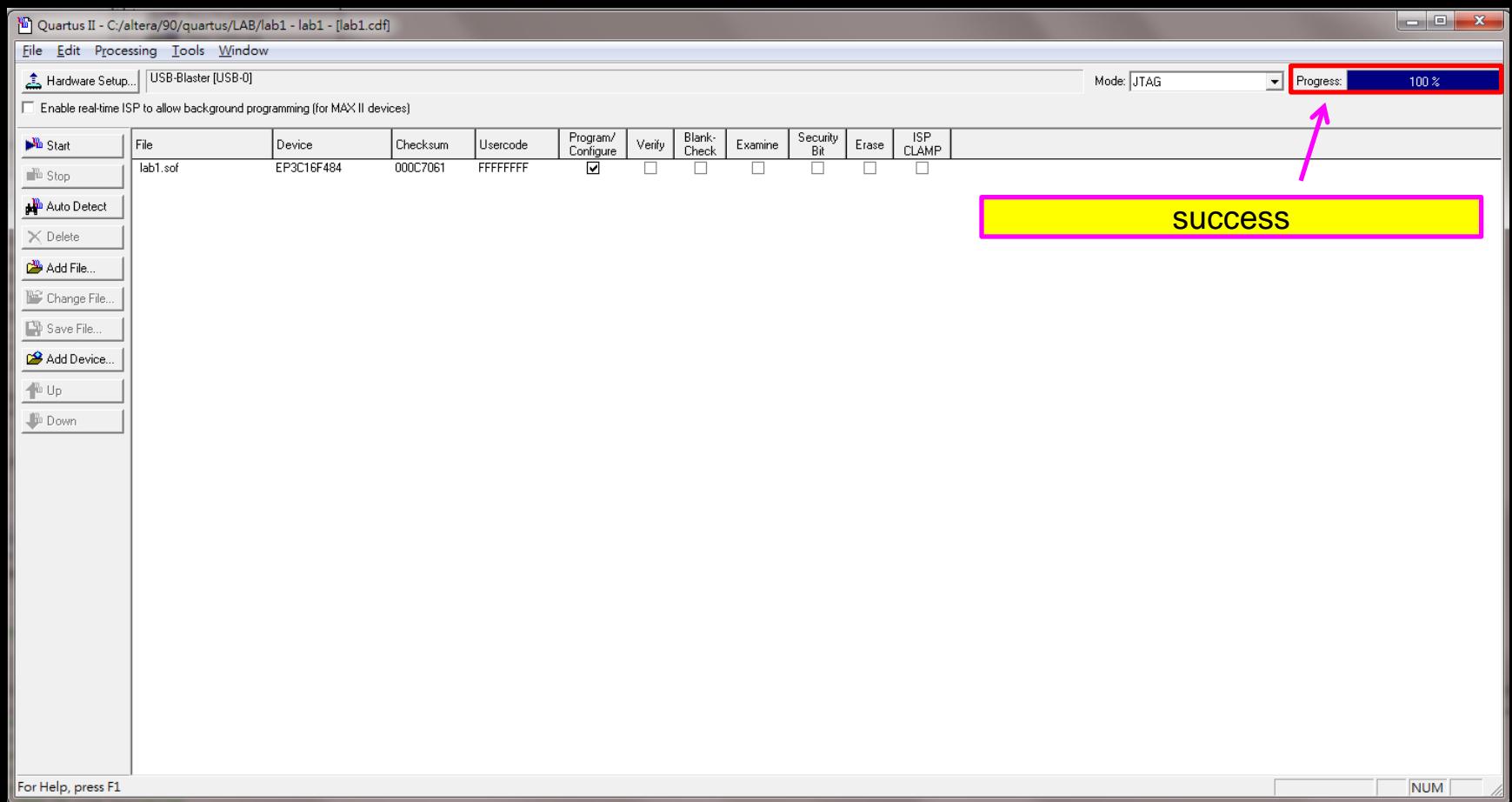
Read Programming File



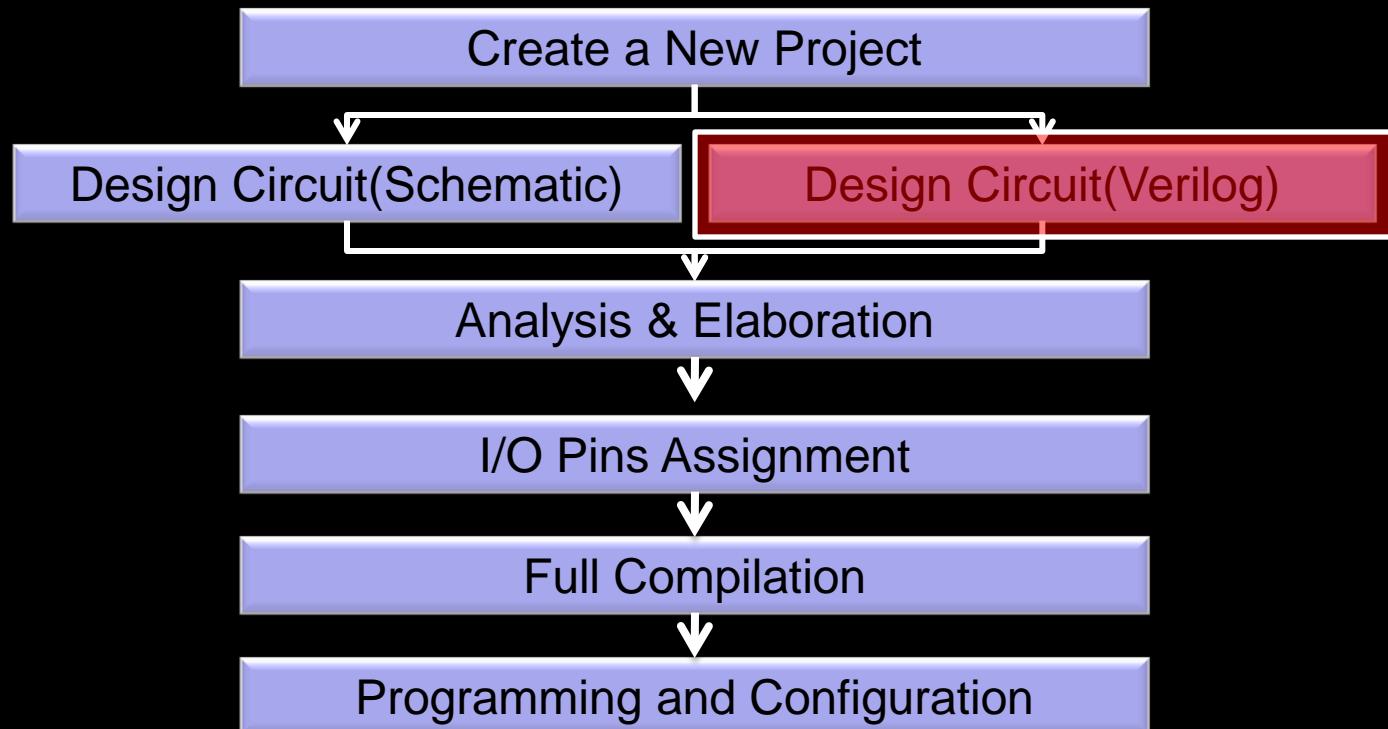
Run Programming File



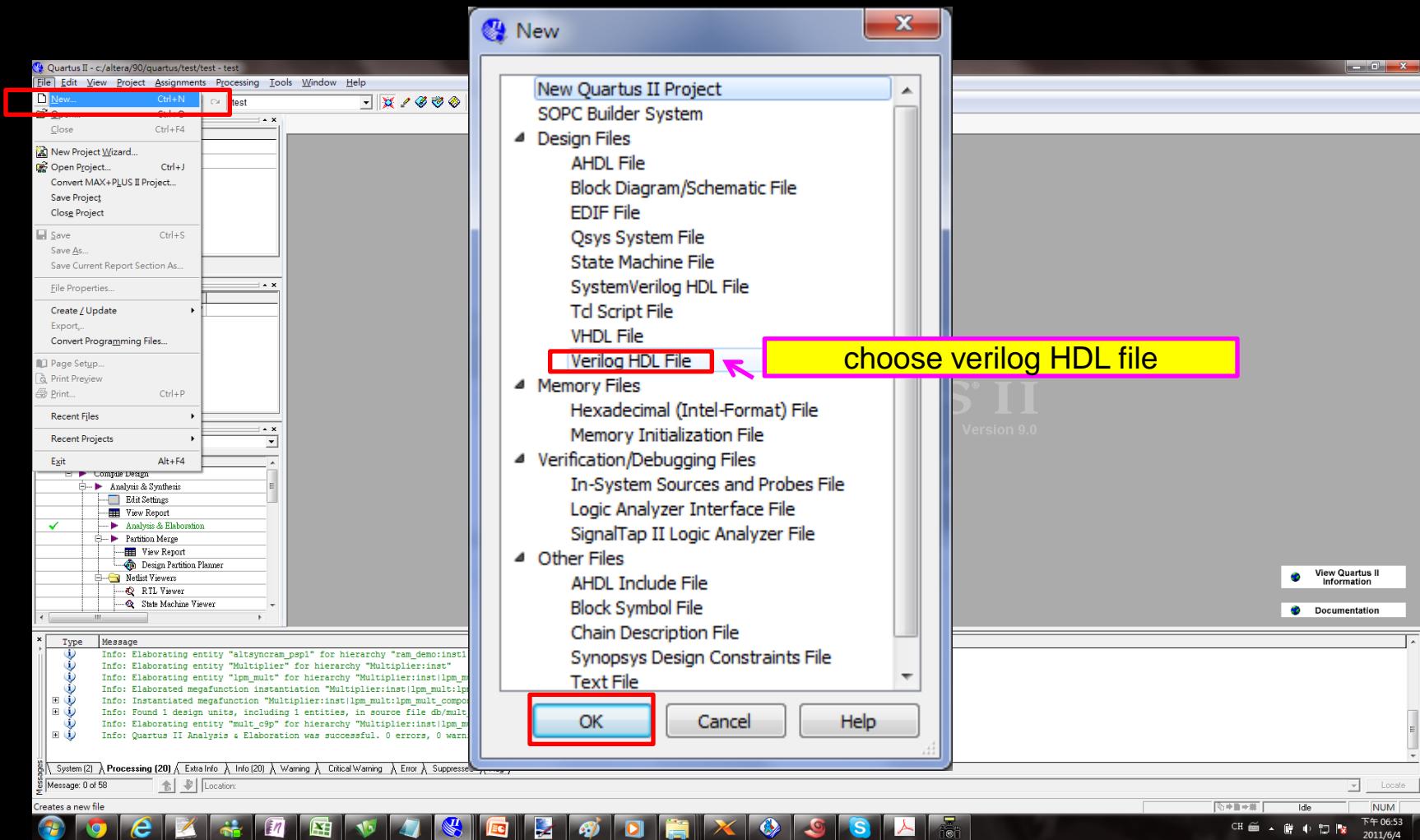
Success of Configuration



Flow of FPGA Design with Quartus II



Logic Design(Verilog File)



Lab2: Switch2Led

The screenshot shows the Quartus II software interface with the project "Lab2" open. The "Switch2Led.v" file is selected in the editor window. A red box highlights the Verilog code for the module definition:

```
1 module Switch2Led(Sw,Led);
2   input [9:0] Sw;
3   output [9:0] Led;
4   reg [9:0] Led;
5   always@ (Sw)
6   begin
7     Led[9:0] = Sw[9:0];
8   end
9 endmodule
10
```

The rest of the interface includes the Project Navigator, Node Finder, Status bar, and Messages panel.

Top-level Module (Lab 2)

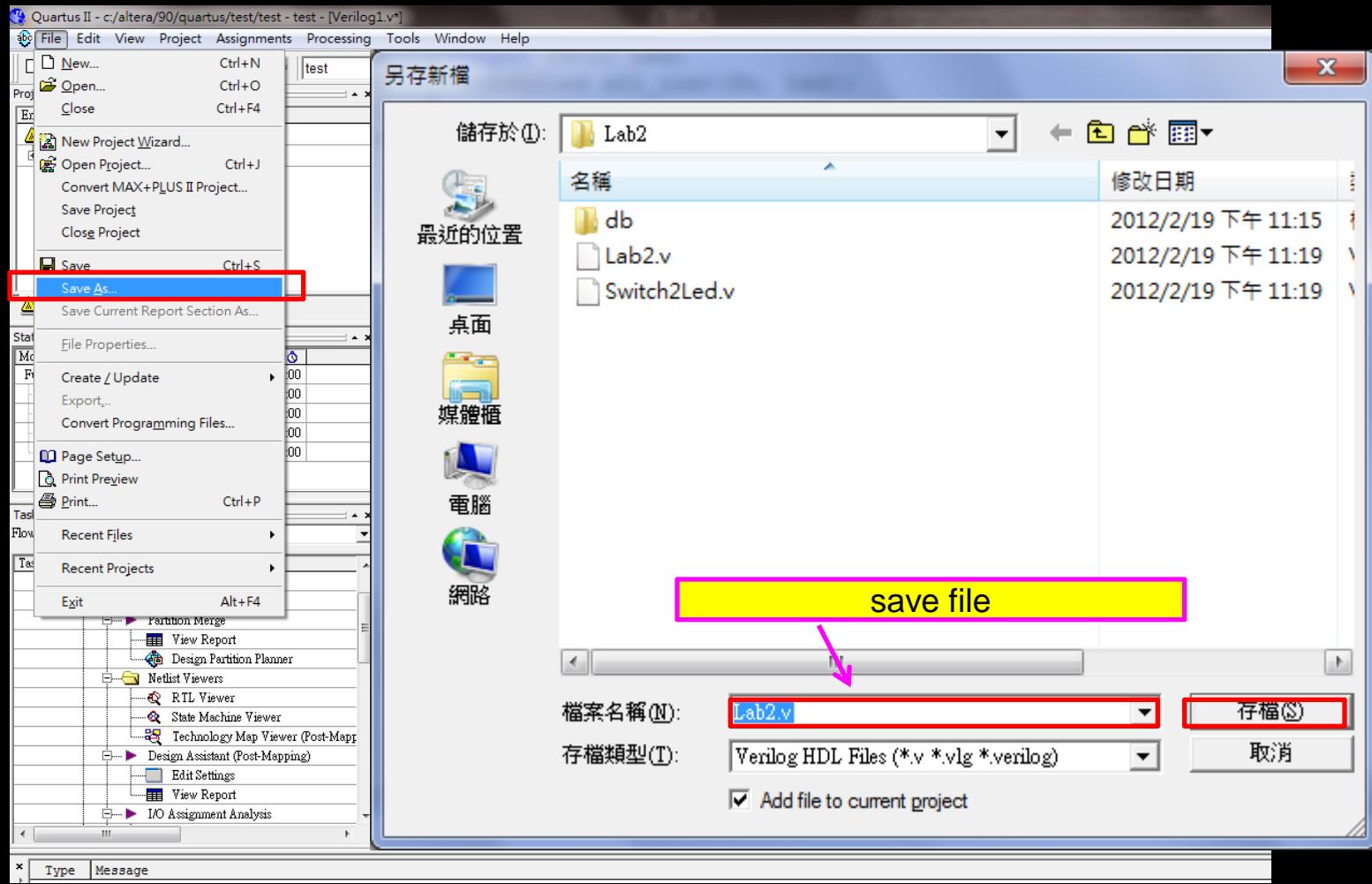
The screenshot shows the Quartus II software interface with the following components:

- Project Navigator**: Shows the project structure with an entity named "Lab2".
- Code Editor (Lab2.v*)**: Displays the Verilog code:

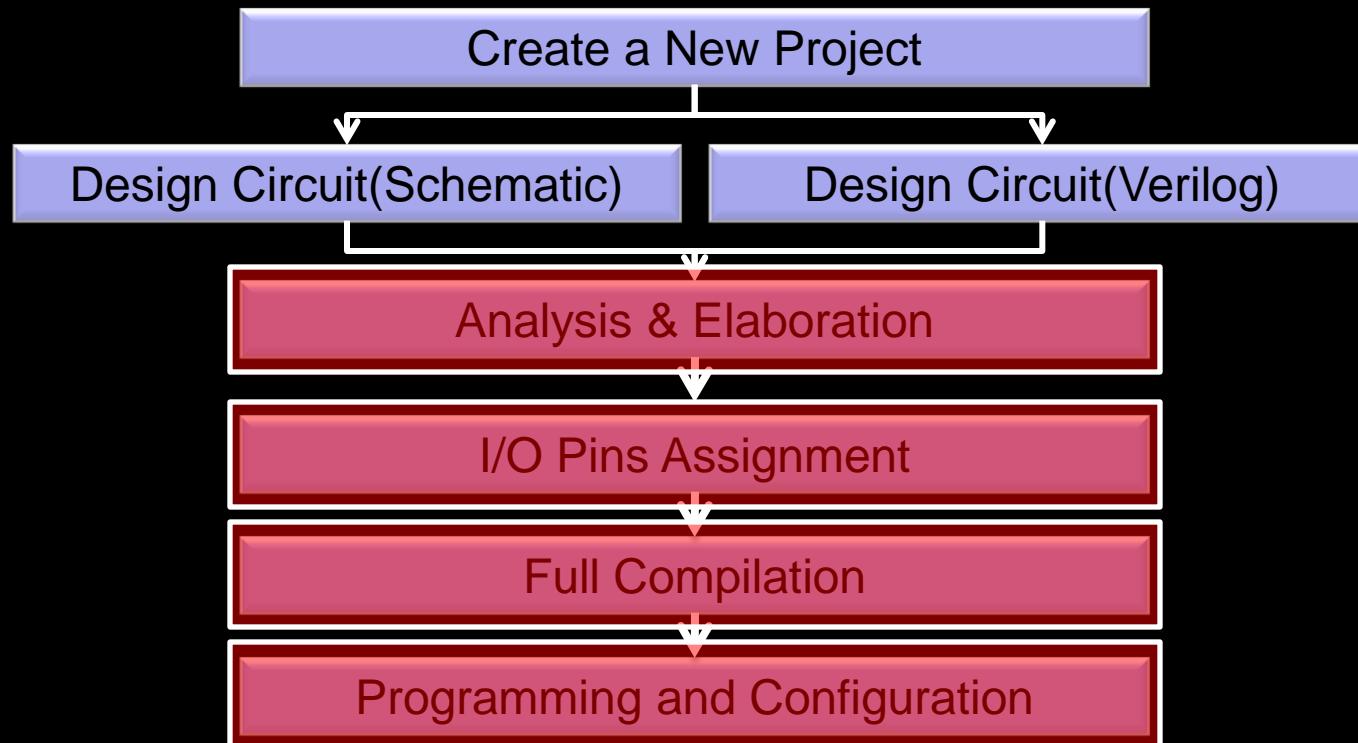
```
1 module Lab2(Sw, Led);
2   input [9:0] Sw;
3   output [9:0] Led;
4   Switch2Led s2l_inst(Sw, Led);
5 endmodule
```

Annotations with red arrows point from the text "Module" to the word "endmodule" and from "Module instance" to the line "Switch2Led s2l_inst(Sw, Led);".
- Hierarchy**, **Files**, and **Design Units** tabs in the Project Navigator.
- Tasks**: A list of compilation steps including Compile Design, Analysis & Synthesis, and various post-mapping and place & route tasks.

Save As Verilog File



Flow of FPGA Design with Quartus II



Lab3: Binary to Decimal

```
1 module Lab3(Sw, Seg7);
2   input [9:0] Sw;
3   output [31:0] Seg7;
4   Bin2Dec b2d_inst(Sw, Seg7);
5 endmodule
6
```

Top module

```
1 module Bin2Dec(Sw,Seg7);
2   input [9:0] Sw;
3   output [31:0] Seg7;
4   reg [31:0] Seg7;
5   always@(Sw)
6   begin
7     Seg7 = numDecode(Sw);
8   end
9
10 function [31:0] numDecode;
11   input [9:0] Sw;
12   integer num;
13 begin
14   integer n0, n1,n2,n3;
15   num = Sw;
16   n0 = num/1000;
17   n1 = (num%1000)/100;
18   n2 = (num%100)/10;
19   n3 = (num%10);
20   numDecode[31:24] = seg7Decode(n0);
21   numDecode[23:16] = seg7Decode(n1);
22   numDecode[15:8] = seg7Decode(n2);
23   numDecode[7:0] =seg7Decode(n3);
24 end
25 endfunction
26
```

```
27 function [7:0] seg7Decode;
28   input [31:0] num;
29 begin
30   case(num)
31     0 : seg7Decode = 8'b11000000; // 0
32     1 : seg7Decode = 8'b11111001; // 1
33     2 : seg7Decode = 8'b10100100; // 2
34     3 : seg7Decode = 8'b10110000; // 3
35     4 : seg7Decode = 8'b10011001; // 4
36     5 : seg7Decode = 8'b10010010; // 5
37     6 : seg7Decode = 8'b10000010; // 6
38     7 : seg7Decode = 8'b11011000; // 7
39     8 : seg7Decode = 8'b10000000; // 8
40     9 : seg7Decode = 8'b10010000; // 9
41   default :
42     seg7Decode = 8'b11111111;
43   endcase
44 end
45 endfunction
46
47 endmodule
```

Sub module