LED Driver (detailed operation of creating a new project instance)

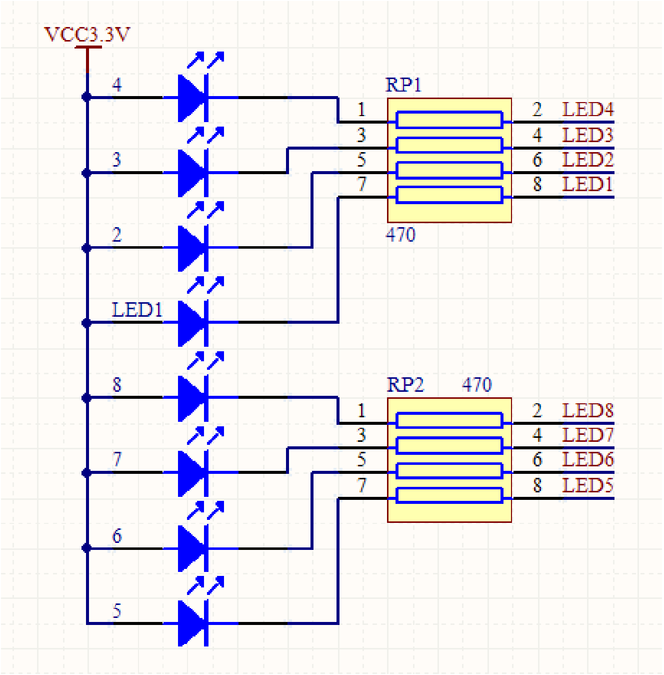
The most important thing to learn FPGA knowledge is to master the basic theory and focus on practice and hands-on. The following will take LED driver as an example to write the basic process of FPGA development as detailed as possible and be familiar with using Quartus II software to understand and master these skills.

Speaking of LED drivers, those who learn electronics are no strangers. In my opinion, it is equivalent to the first program "Hello world!" compiled to introduce CS. Although it is simple, you can still learn a lot of basic knowledge from it. Usually, a certain current and voltage difference must pass through the LED's two ends to make it emit light. It cannot be directly lit by connecting the power supply and ground at both ends, and a current limiting resistor needs to be added to control the current flowing through the LED to not be too large.

The following part is written with the development board of EP2C5Q208 as the basis, and some of the contents differ from the student board. Please make some changes as appropriate.

There are a total of 8 light-emitting diodes LED1-LED8 on the bottom plate of the Ruizhi FPGA development board (Note: 2 light-emitting diodes are on the core board)

LED1, LED2, and the bottom board LED1, LED2 are connected to the same FPGA pins, we first need to complete a task: Light the 8 LEDs at intervals. The hardware schematic diagram is as follows:



If you want to light up these LEDs, you only need to output a low-level “0” to the FPGA pin connected to it to achieve this function. If we use the continuous assignment statement assign to achieve, write the program code as follows:

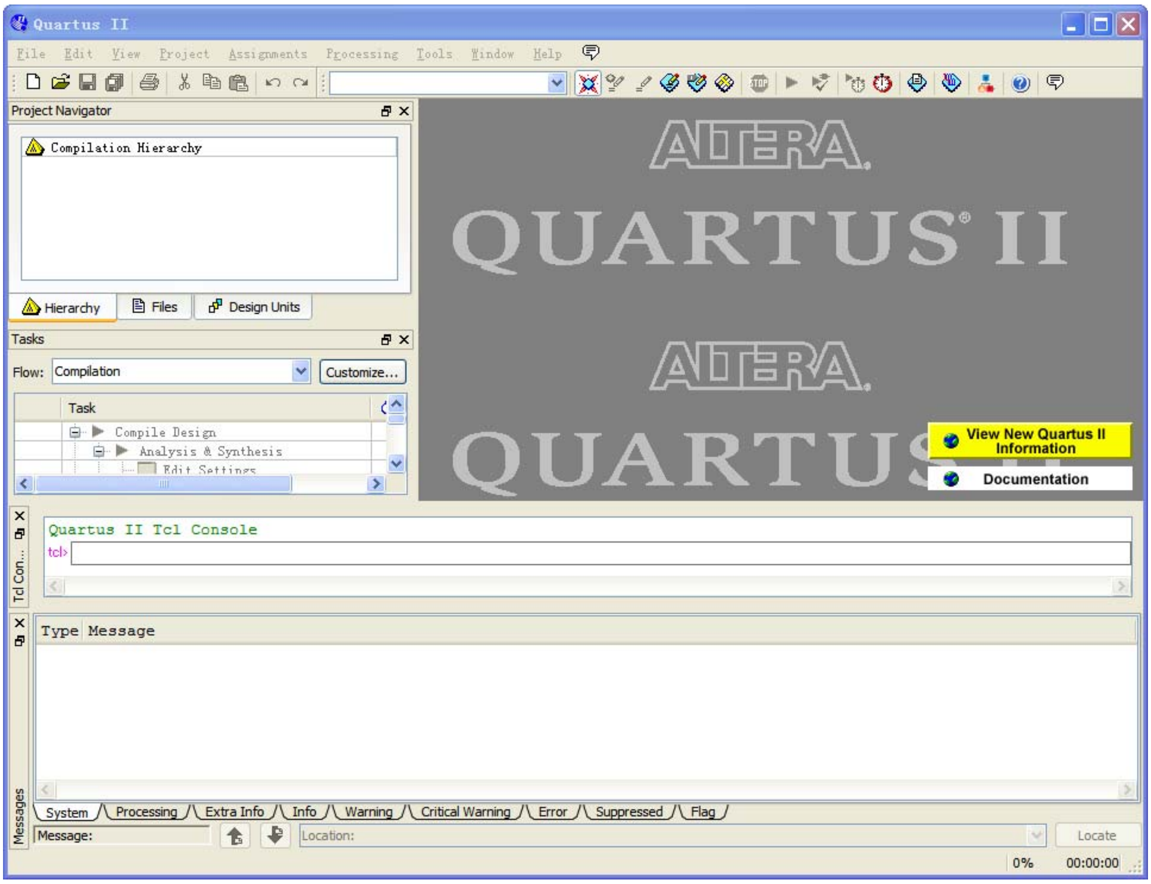
```
module led_light(led);
output[7:0] led;
assign led=8'b10101010;
endmodule
```

Next, we will use this simple example to carry out the whole process of the new project, which is very valuable for complete beginners, and I hope everyone can master it proficiently!

Starting any design in Quartus II is a project. During the engineering design process, many files with the same name with different extensions will be generated, which are placed in the same folder for unified management. Therefore, different design projects are best placed in different folders. In this example, the folder D:\FPGA_study is established as the working library so that the relevant files in the design process can be stored here.

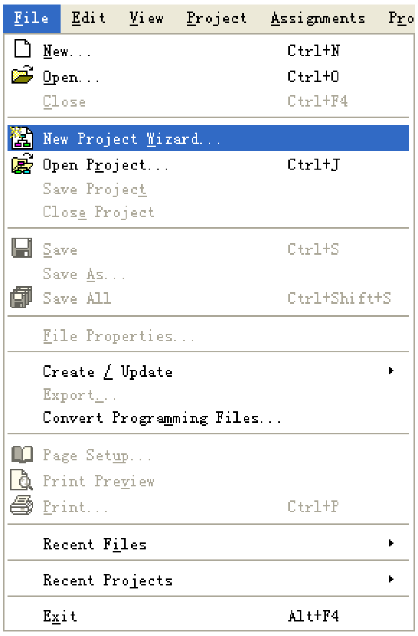
Start Quartus II

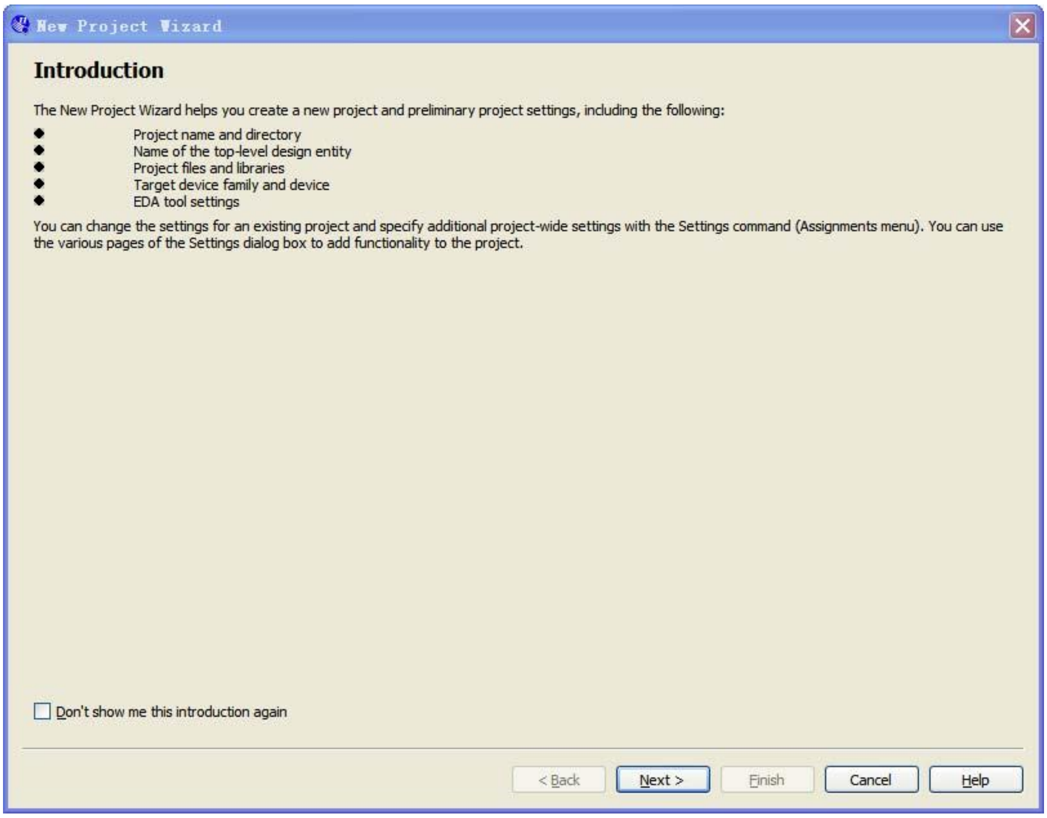
Double-click the Quartus II icon on the desktop or click the start button and select Quartus II 9 in the program menu to start Quartus II. It’s initial interface is shown in the figure



In the figure, click the left button under the File tab to open the menu shown in Figure 3, and click the new project wizard.

Click File->New project wizard.. to create a new project led_light





Introduction

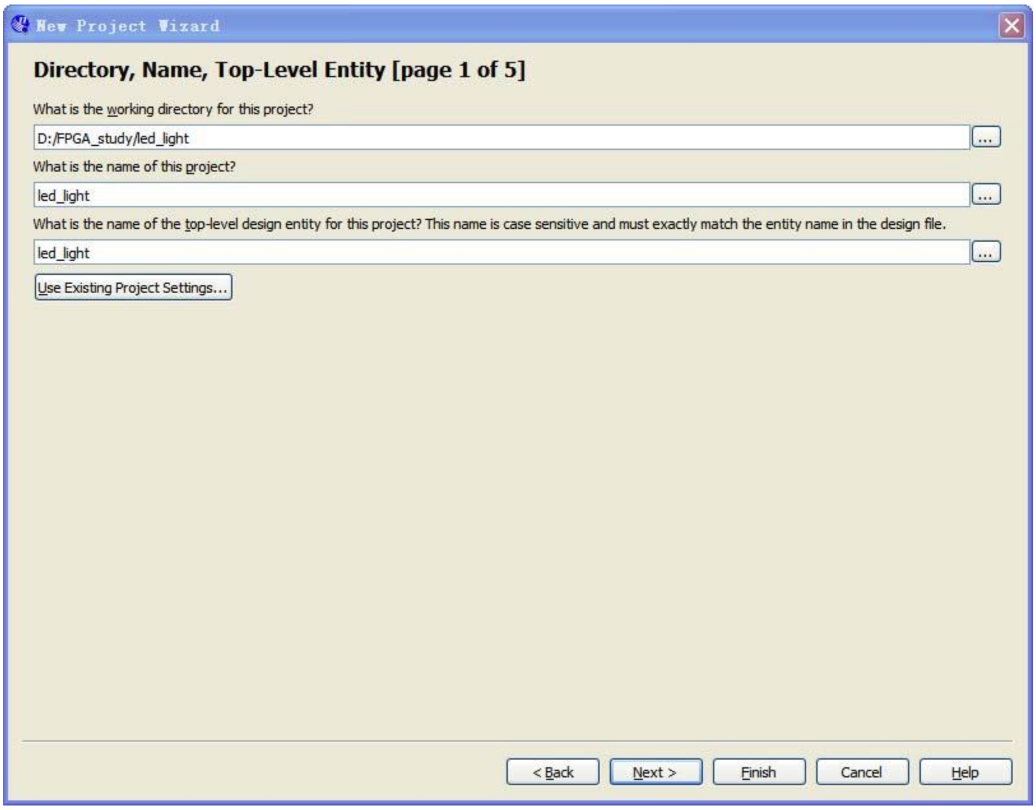
The New Project Wizard helps you create a new project and preliminary project settings, including the following:

- Project name and directory
- Name of the top-level design entity
- Project files and libraries
- Target device family and device
- EDA tool settings

You can change the settings for an existing project and specify additional project-wide settings with the Settings command (Assignments menu). You can use the various pages of the Settings dialog box to add functionality to the project.

□ Don't show me this introduction again

Click Next, go to the next step, enter the basic settings dialog box



New Project Wizard

Directory, Name, Top-Level Entity [page 1 of 5]

What is the working directory for this project?

D:/FPGA_study/led_light

What is the name of this project?

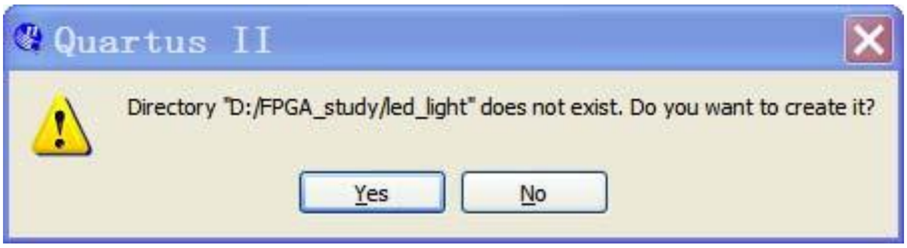
led_light

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

led_light

Enter the address of the work library folder in the input box of the first column. The input box in the second column requires the name of the project to be entered. Generally, the name of the top-level file can be used as the project name. Here we write “led_light”. The input box of the third column requires the name of the top-level design file entity. In fact, you will find that when you write the second column, the system automatically completes the third column, which is also led_light. When finished, just click Finish.

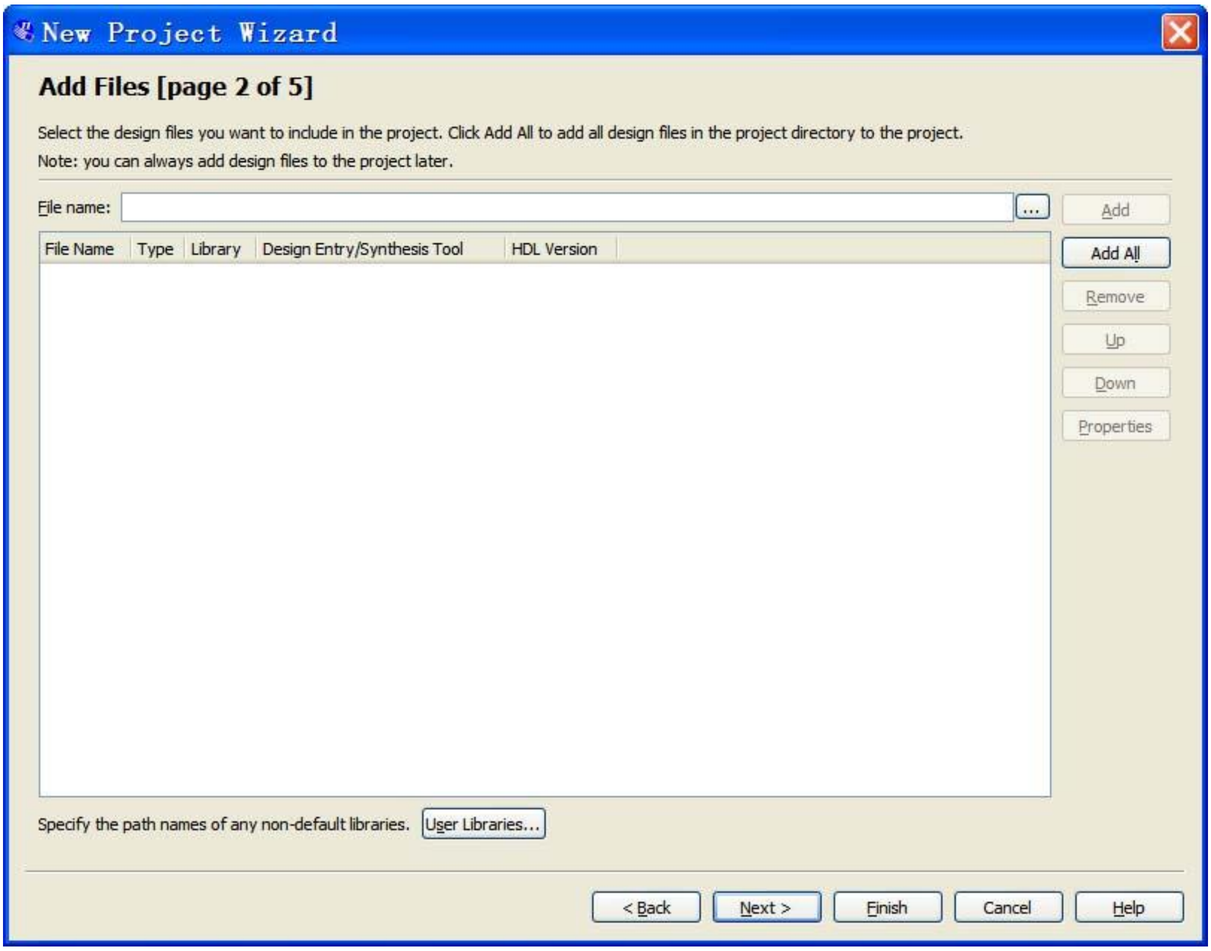
Because the folder has not been created, Quartus II pops up a dialog box asking whether to create the desired folder, as shown in the figure.



Quartus II

Directory "D:/FPGA_study/led_light" does not exist. Do you want to create it?

Clicking Yes will bring up the window shown below. Click “Yes”



New Project Wizard

Add Files [page 2 of 5]

Select the design files you want to include in the project. Click Add All to add all design files in the project directory to the project.
Note: you can always add design files to the project later.

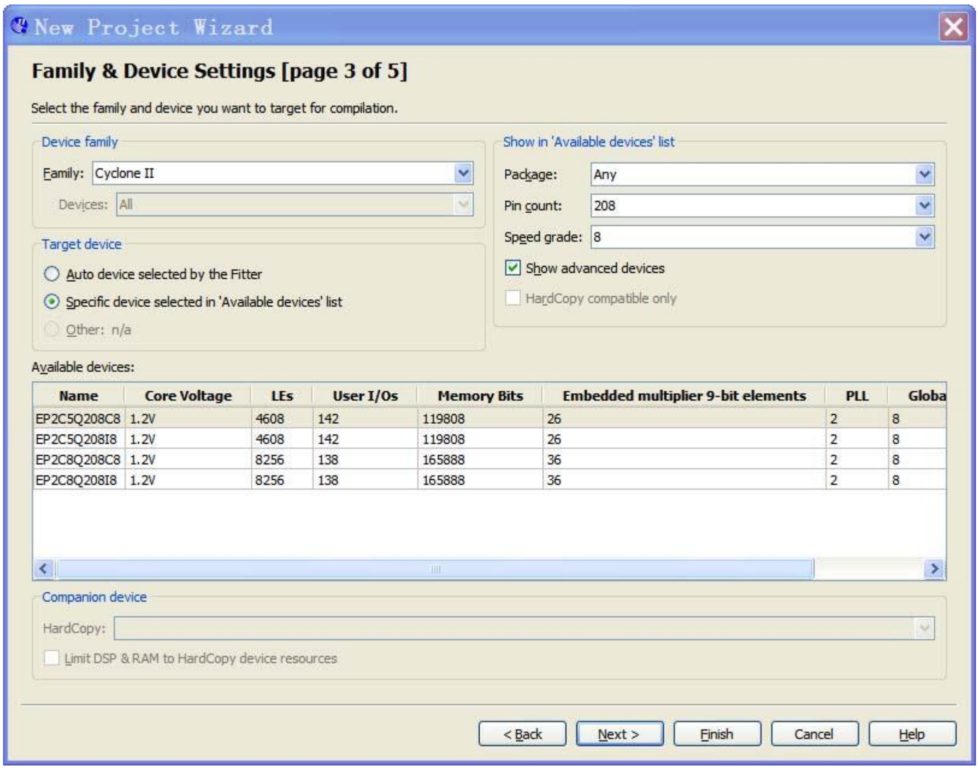
File name:

File Name	Type	Library	Design Entry/Synthesis Tool	HDL Version

Specify the path names of any non-default libraries.

Because there is no specific design verilog file, let’s click [next]

Select the FPGA device used and make some necessary configurations below. For example, the development board uses EP2C5Q208C8, click the menu Assignments->Device.., the following dialog box appears.



Family & Device Settings [page 3 of 5]

Select the family and device you want to target for compilation.

Device family

Family: Cyclone II

Devices: All

Target device

○ Auto device selected by the Fitter

● Specific device selected in 'Available devices' list

○ Other: n/a

Show in 'Available devices' list

Package: Any

Pin count: 208

Speed grade: 8

☑ Show advanced devices

□ HardCopy compatible only

Available devices:

Name	Core Voltage	LEs	User I/Os	Memory Bits	Embedded multiplier 9-bit elements	PLL	Globa
EP2C5Q208C8	1.2V	4608	142	119808	26	2	8
EP2C5Q208I8	1.2V	4608	142	119808	26	2	8
EP2C8Q208C8	1.2V	8256	138	165888	36	2	8
EP2C8Q208I8	1.2V	8256	138	165888	36	2	8

Companion device

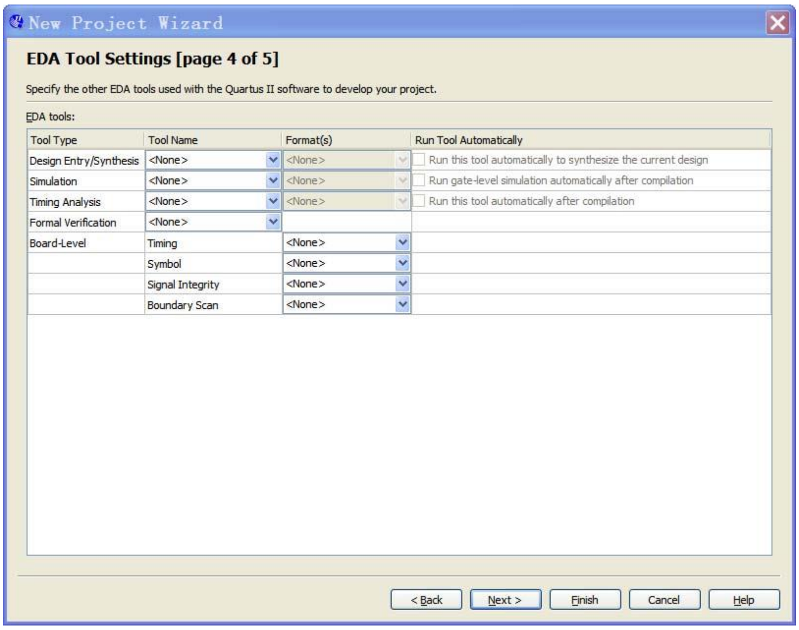
HardCopy:

□ Limit DSP & RAM to HardCopy device resources

With several options for filtering, we can choose according to our own understanding, such as:

EP2C5Q208C8 belongs to Cyclone II system, select Cyclone II in Device Family;
EP2C5Q208C8 has a total of 208 pins and the Pin count is 208;
EP2C5Q208C8 device speed grade is 8, Speed grade choose 8;

At this time, the devices that meet the above conditions have been listed in the Available device list, then we select EP2C5Q208C8, click OK to confirm

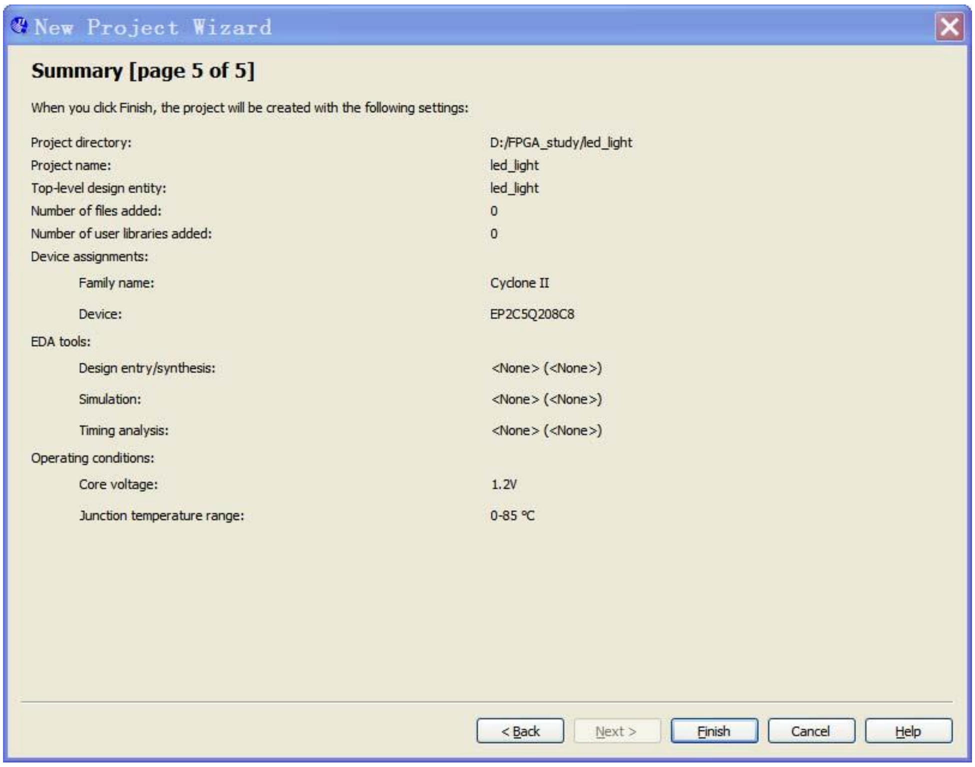


EDA Tool Settings [page 4 of 5]

Specify the other EDA tools used with the Quartus II software to develop your project.

EDA tools:

Tool Type	Tool Name	Format(s)	Run Tool Automatically
Design Entry/Synthesis	<None>	<None>	□ Run this tool automatically to synthesize the current design
Simulation	<None>	<None>	□ Run gate-level simulation automatically after compilation
Timing Analysis	<None>	<None>	□ Run this tool automatically after compilation
Formal Verification	<None>		
Board-Level	Timing	<None>	
	Symbol	<None>	
	Signal Integrity	<None>	
	Boundary Scan	<None>	



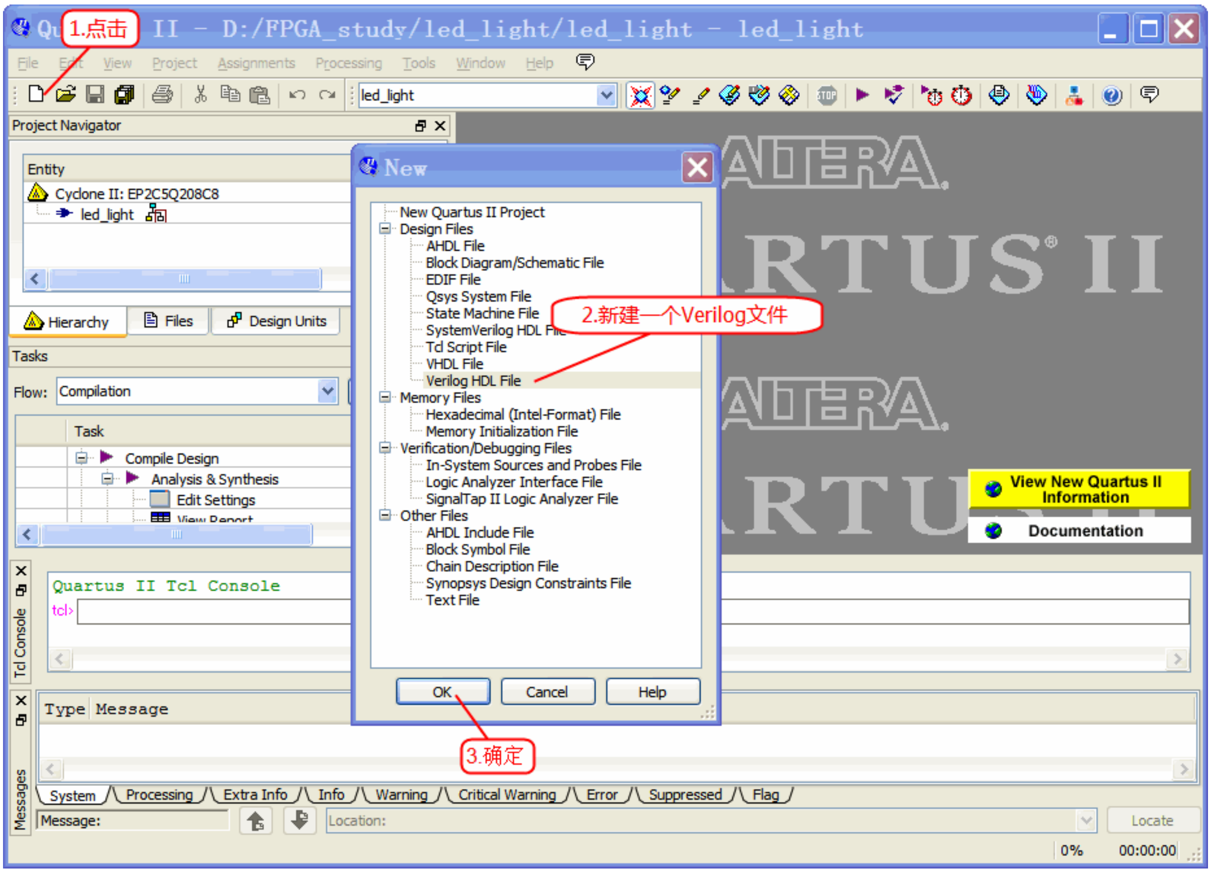
New Project Wizard

Summary [page 5 of 5]

When you click Finish, the project will be created with the following settings:

Project directory:	D:/FPGA_study/led_light
Project name:	led_light
Top-level design entity:	led_light
Number of files added:	0
Number of user libraries added:	0
Device assignments:	
Family name:	Cyclone II
Device:	EP2C5Q208C8
EDA tools:	
Design entry/synthesis:	<None> (<None>)
Simulation:	<None> (<None>)
Timing analysis:	<None> (<None>)
Operating conditions:	
Core voltage:	1.2V
Junction temperature range:	0-85 ℃

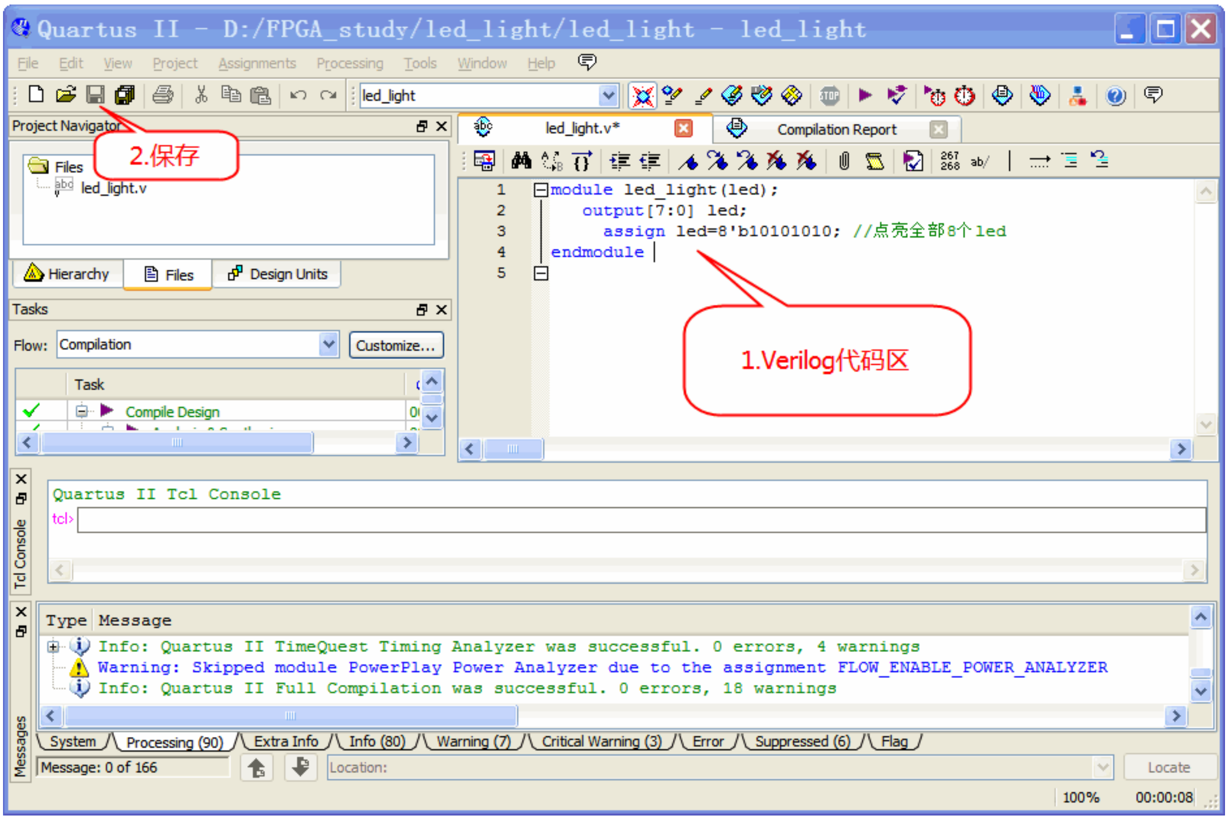
Click File->New.. to create a new Verilog HDL source program file led_light.v under the project (or directly click the shortcut icon, as shown in the figure below), enter the source code and save it. A dialog box for saving will pop up, we found that the saved file directory is already under led_light, and the file name is also led_light.v by default, no need to change it, just save it.



1.点击

2.新建一个Verilog文件

3.确定

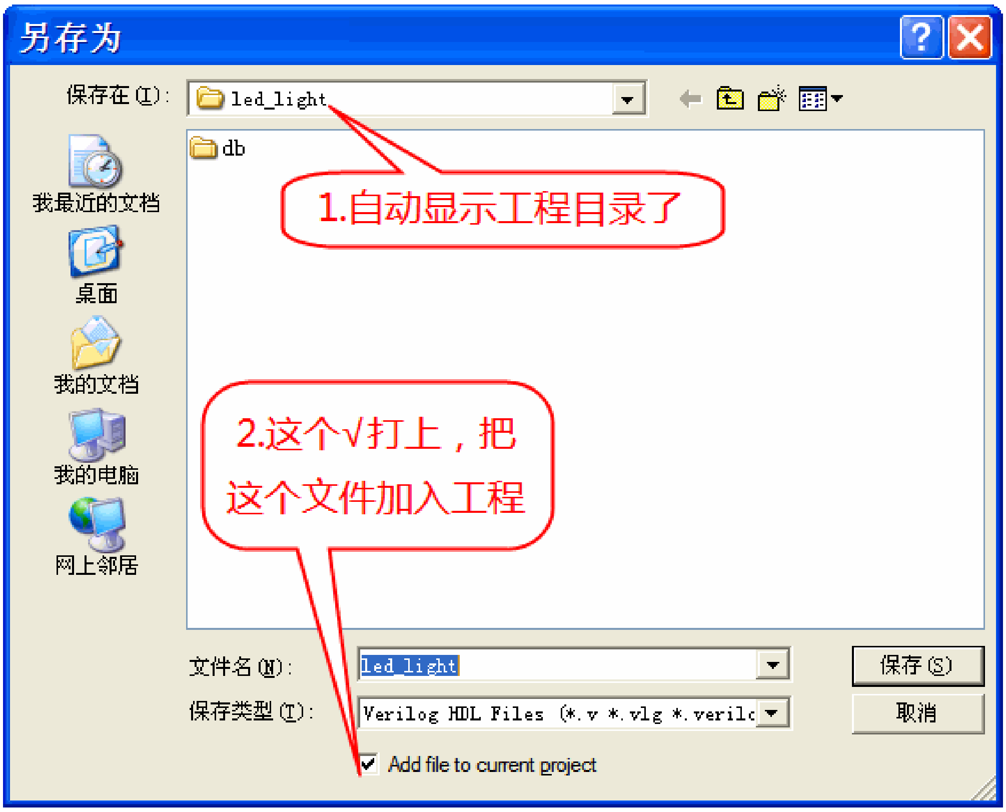


```
module led_light(led);
   output[7:0] led;
     assign led=8'b10101010; //点亮全部8个led
endmodule
```

```
Info: Quartus II TimeQuest Timing Analyzer was successful. 0 errors, 4 warnings
Warning: Skipped module PowerPlay Power Analyzer due to the assignment FLOW_ENABLE_POWER_ANALYZER
Info: Quartus II Full Compilation was successful. 0 errors, 18 warnings
```

2.保存

1.Verilog代码区



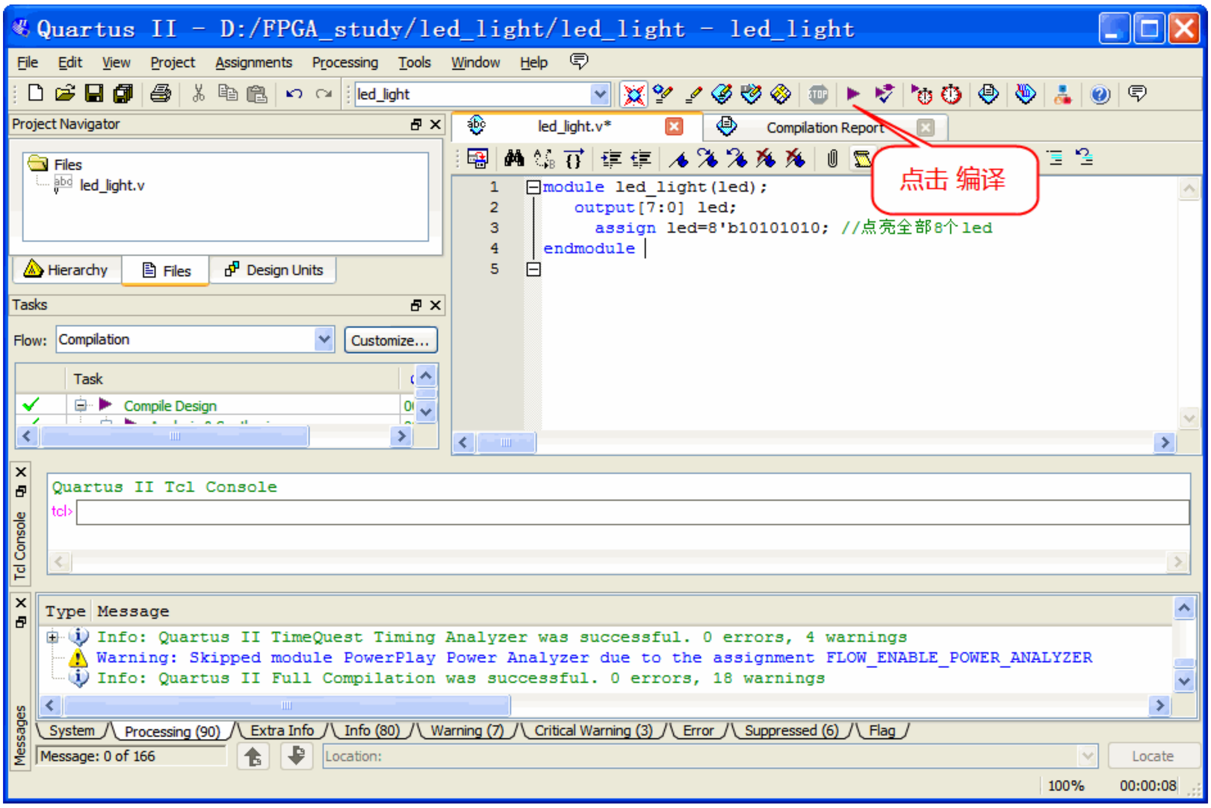
文件名(N): led_light

保存类型(T): Verilog HDL Files (*.v *.vlg *.verilo

☑ Add file to current project

1.自动显示工程目录了

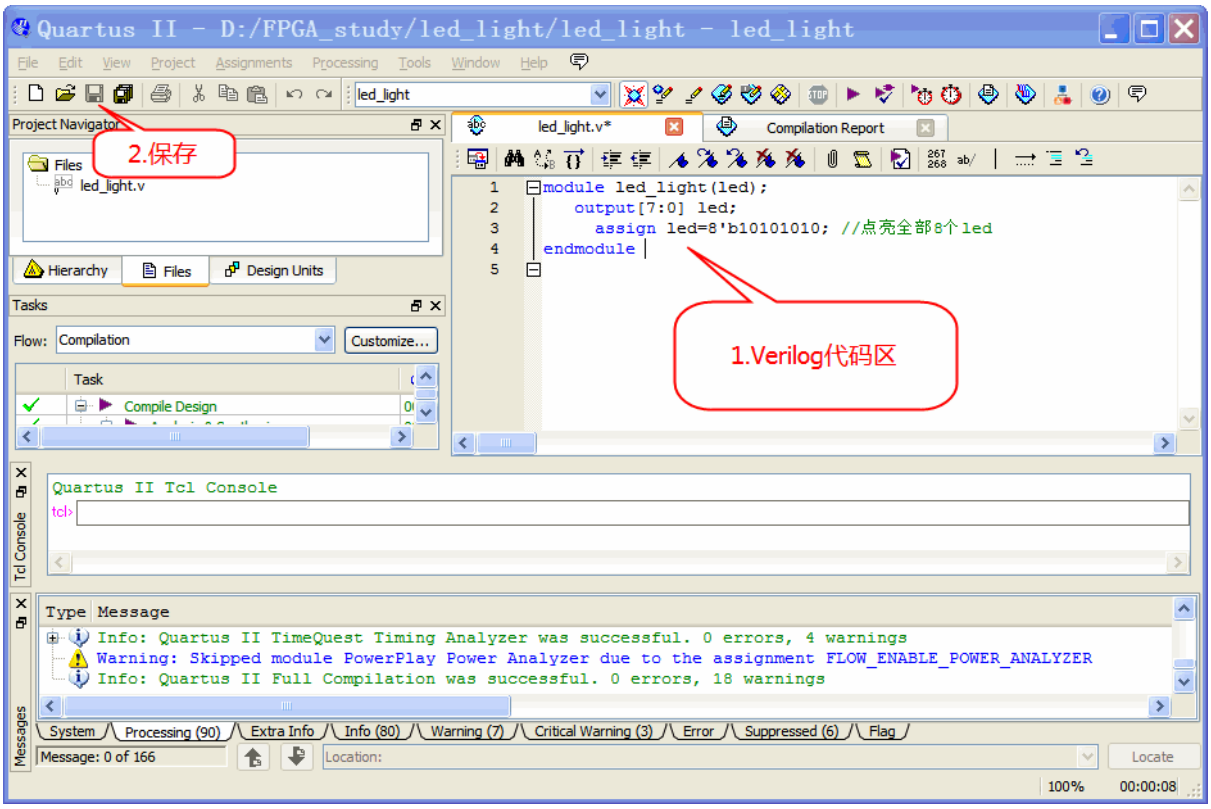
2.这个√打上，把这个文件加入工程



点击 编译

```
module led_light(led);
   output[7:0] led;
     assign led=8'b10101010; //点亮全部8个led
endmodule
```

```
Info: Quartus II TimeQuest Timing Analyzer was successful. 0 errors, 4 warnings
Warning: Skipped module PowerPlay Power Analyzer due to the assignment FLOW_ENABLE_POWER_ANALYZER
Info: Quartus II Full Compilation was successful. 0 errors, 18 warnings
```

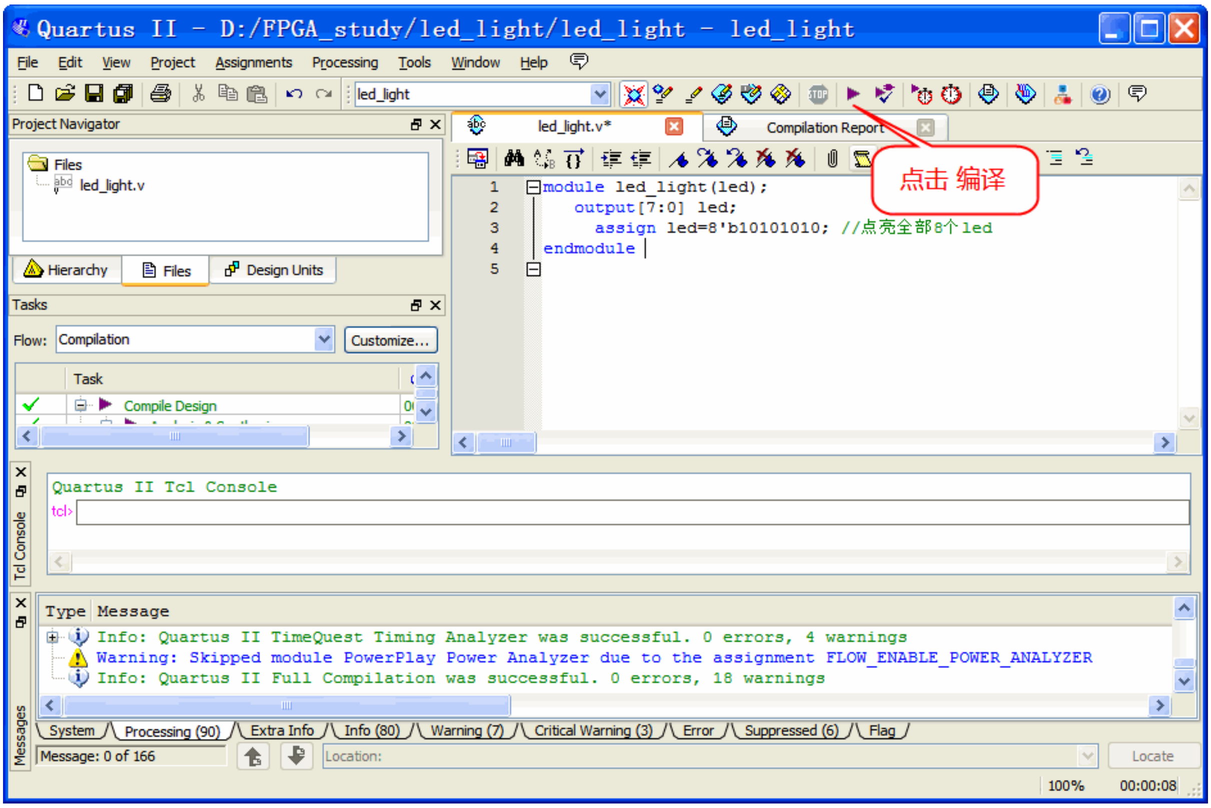


2.保存

1.Verilog代码区

```
module led_light(led);
   output[7:0] led;
     assign led=8'b10101010; //点亮全部8个led
endmodule
```

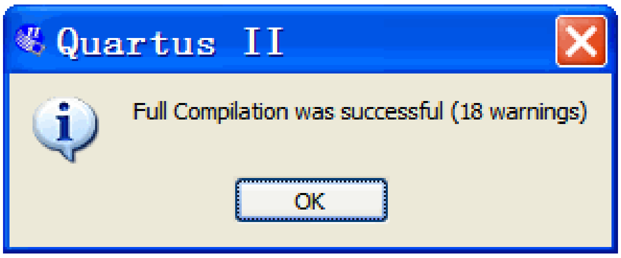
```
Info: Quartus II TimeQuest Timing Analyzer was successful. 0 errors, 4 warnings
Warning: Skipped module PowerPlay Power Analyzer due to the assignment FLOW_ENABLE_POWER_ANALYZER
Info: Quartus II Full Compilation was successful. 0 errors, 18 warnings
```



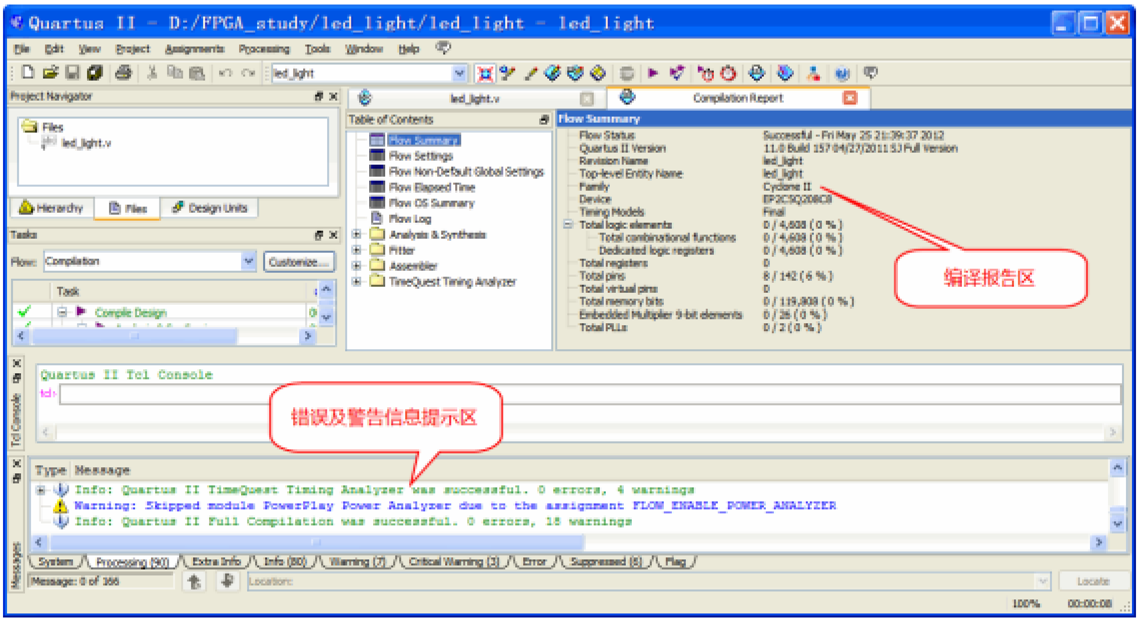
```
module led_light(led);
    output[7:0] led;
      assign led=8'b10101010; //点亮全部8个led
endmodule
```

```
Quartus II Tcl Console
tcl>
```

```
Type Message
Info: Quartus II TimeQuest Timing Analyzer was successful. 0 errors, 4 warnings
Warning: Skipped module PowerPlay Power Analyzer due to the assignment FLOW_ENABLE_POWER_ANALYZER
Info: Quartus II Full Compilation was successful. 0 errors, 18 warnings
```



Full Compilation was successful (18 warnings)

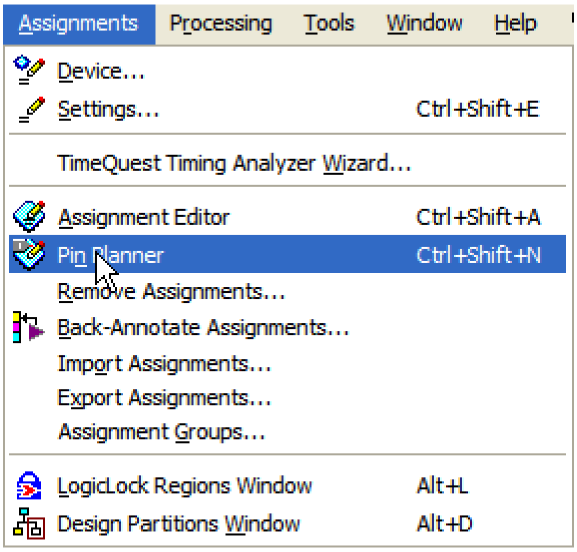


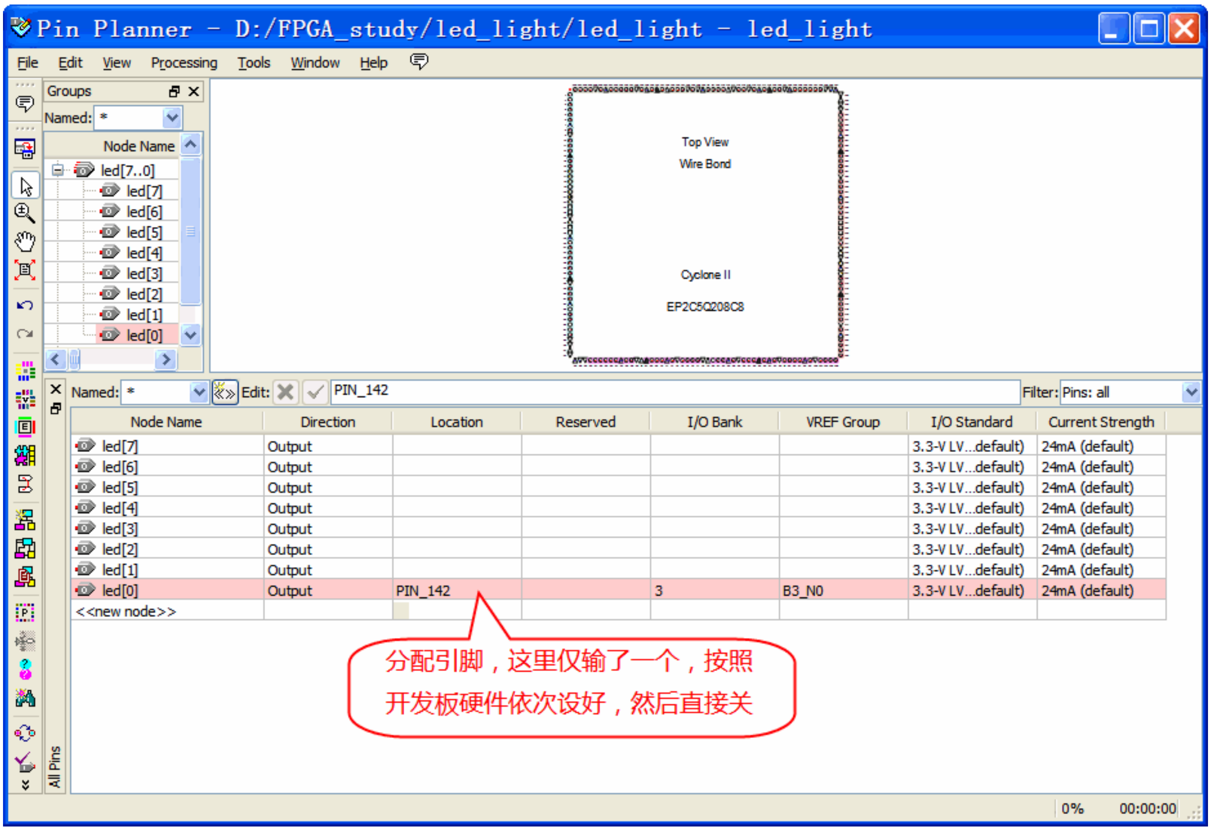
Flow Summary

Flow Status	Successful - Fri May 25 21:39:37 2012
Quartus II Version	11.0 Build 157 04/27/2011 SJ Full Version
Revision Name	led_light
Top-level Entity Name	led_light
Family	Cyclone II
Device	EP2C5Q208C8
Timing Models	Final
Total logic elements	0 / 4,608 (0 %)
Total combinational functions	0 / 4,608 (0 %)
Dedicated logic registers	0 / 4,608 (0 %)
Total registers	0
Total pins	8 / 142 (6 %)
Total virtual pins	0
Total memory bits	0 / 119,808 (0 %)
Embedded Multiplier 9-bit elements	0 / 26 (0 %)
Total PLLs	0 / 2 (0 %)

```
Quartus II Tcl Console
tcl>
```

```
Type Message
Info: Quartus II TimeQuest Timing Analyzer was successful. 0 errors, 4 warnings
Warning: Skipped module PowerPlay Power Analyzer due to the assignment FLOW_ENABLE_POWER_ANALYZER
Info: Quartus II Full Compilation was successful. 0 errors, 18 warnings
```

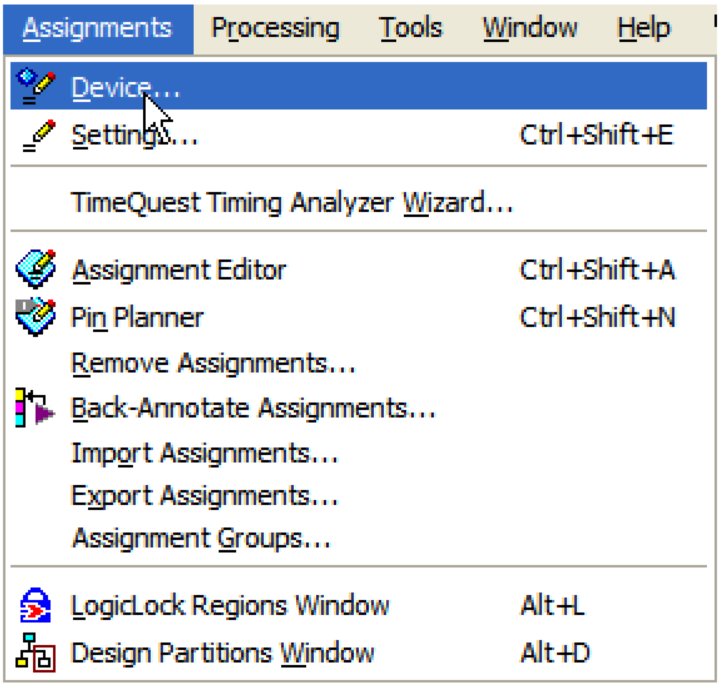


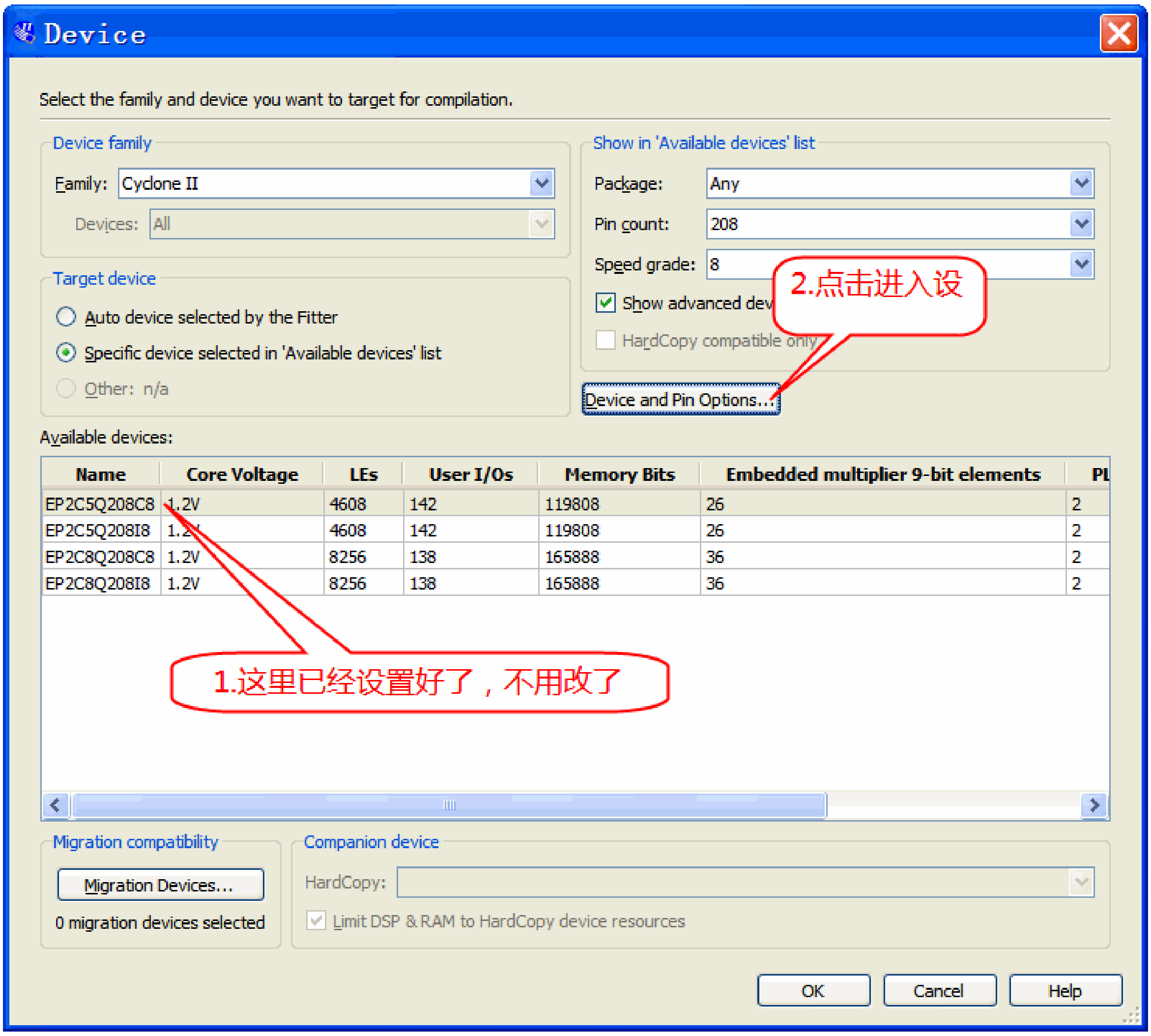


PIN_142

Node Name	Direction	Location	Reserved	I/O Bank	VREF Group	I/O Standard	Current Strength
led[7]	Output					3.3-V LV...default)	24mA (default)
led[6]	Output					3.3-V LV...default)	24mA (default)
led[5]	Output					3.3-V LV...default)	24mA (default)
led[4]	Output					3.3-V LV...default)	24mA (default)
led[3]	Output					3.3-V LV...default)	24mA (default)
led[2]	Output					3.3-V LV...default)	24mA (default)
led[1]	Output					3.3-V LV...default)	24mA (default)
led[0]	Output	PIN_142		3	B3_N0	3.3-V LV...default)	24mA (default)
<<new node>>							

分配引脚，这里仅输了一个，按照开发板硬件依次设好，然后直接关




Device

Select the family and device you want to target for compilation.

Device family

Family: Cyclone II

Devices: All

Target device

○ Auto device selected by the Fitter

● Specific device selected in 'Available devices' list

○ Other: n/a

Show in 'Available devices' list

Package: Any

Pin count: 208

Speed grade: 8

☑ Show advanced dev

□ HardCopy compatible only

2.点击进入设

Device and Pin Options...

Available devices:

Name	Core Voltage	LEs	User I/Os	Memory Bits	Embedded multiplier 9-bit elements	PL
EP2C5Q208C8	1.2V	4608	142	119808	26	2
EP2C5Q208I8	1.2V	4608	142	119808	26	2
EP2C8Q208C8	1.2V	8256	138	165888	36	2
EP2C8Q208I8	1.2V	8256	138	165888	36	2

1.这里已经设置好了，不用改了

Migration compatibility

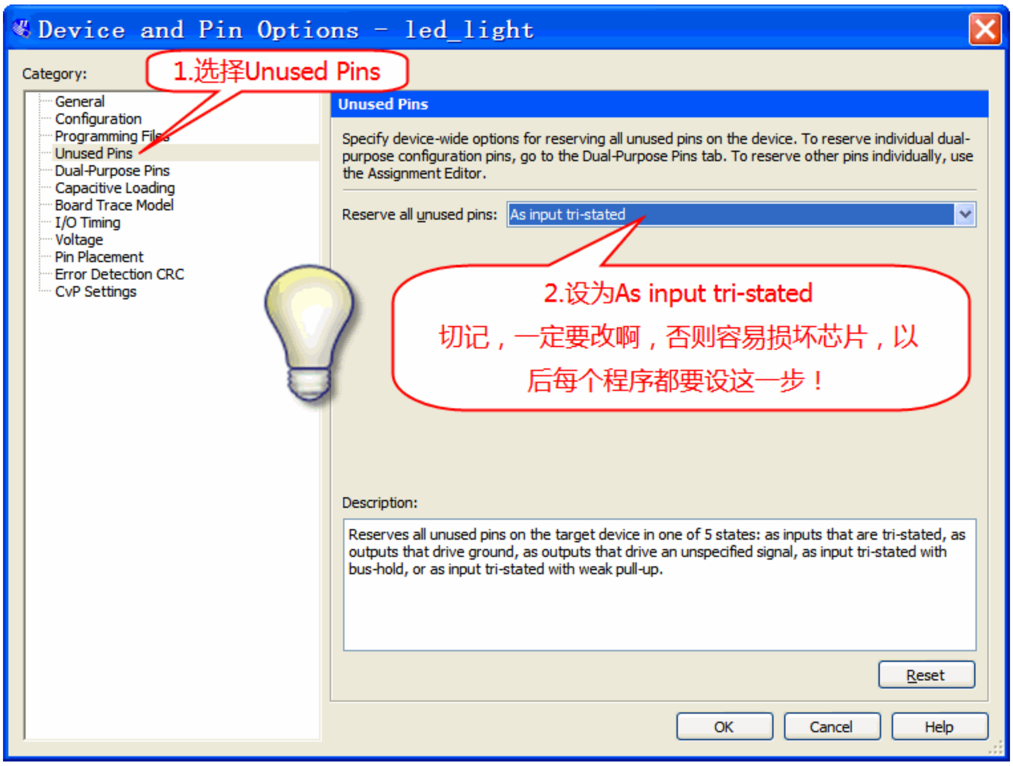
Migration Devices...

0 migration devices selected

Companion device

HardCopy:

☑ Limit DSP & RAM to HardCopy device resources



Device and Pin Options - led_light

1.选择Unused Pins

Category:

Unused Pins

Specify device-wide options for reserving all unused pins on the device. To reserve individual dual-purpose configuration pins, go to the Dual-Purpose Pins tab. To reserve other pins individually, use the Assignment Editor.

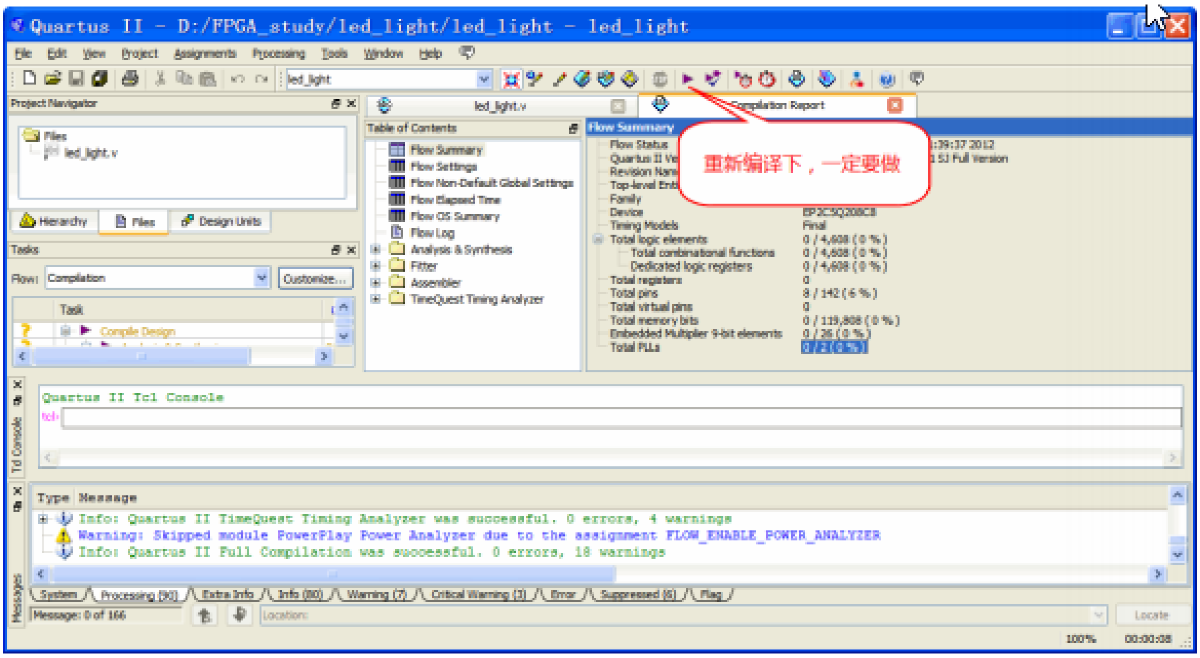
Reserve all unused pins: As input tri-stated

2.设为As input tri-stated

切记，一定要改啊，否则容易损坏芯片，以后每个程序都要设这一步！

Description:

Reserves all unused pins on the target device in one of 5 states: as inputs that are tri-stated, as outputs that drive ground, as outputs that drive an unspecified signal, as input tri-stated with bus-hold, or as input tri-stated with weak pull-up.



Quartus II - D:/FPGA_study/led_light/led_light - led_light

重新编译下，一定要做

Flow Summary

Flow Status	:39:37 2012
Quartus II Ve	1 SJ Full Version
Revision Nam	
Top-level Enti	
Family	
Device	EP2C5Q208C8
Timing Models	Final
Total logic elements	0 / 4,608 (0 %)
Total combinational functions	0 / 4,608 (0 %)
Dedicated logic registers	0 / 4,608 (0 %)
Total registers	0
Total pins	8 / 142 (6 %)
Total virtual pins	0
Total memory bits	0 / 119,808 (0 %)
Embedded Multiplier 9-bit elements	0 / 26 (0 %)
Total PLLs	0 / 2 (0 %)

Quartus II Tcl Console

```
Type Message
Info: Quartus II TimeQuest Timing Analyzer was successful. 0 errors, 4 warnings
Warning: Skipped module PowerPlay Power Analyzer due to the assignment FLOW_ENABLE_POWER_ANALYZER
Info: Quartus II Full Compilation was successful. 0 errors, 18 warnings
```

The Quartus II software displays the generated messages in the Messages window during compilation. If this Verilog is designed correctly, one of the messages will show that the compilation was successful with no errors.

If the compilation is not error-free, then there is at least one error in the Verilog code. In this case, the message corresponding to the error is displayed in the message window. Double-click an error message, and the information will scroll to display the complete and open the Quartus II built-in text compiler, highlighting a line in the Verilog code. Similarly, the compiler also displays many warning messages. The same method works for viewing the full warning message. For an error or warning message, if you want more information, you can select and hold the message and then press and hold the F1 function key to view it.

After the compilation is successful, you can try to download the file to the board, and the method can be operated according to the download method in the previous section.