Quartus II Introduction Using Schematic Design

This tutorial presents an introduction to the Quartus II 9.1 CAD system. It gives a general overview of a typical CAD flow for designing circuits that are implemented by using FPGA devices, and shows how this flow is realized in the Quartus II software. The design process is illustrated by giving step-by-step instructions for using the Quartus II software to implement a very simple circuit in an Altera FPGA device.

The Quartus II system includes full support for all of the popular methods of entering a description of the desired circuit into a CAD system. This tutorial makes use of the schematic design entry method, in which the user draws a graphical diagram of the circuit. Two other versions of this tutorial are also available, which use the Verilog and VHDL hardware description languages, respectively.

The last step in the design process involves configuring the designed circuit in an actual FPGA device. To show how this is done, it is assumed that the user has access to the Altera DE2-115 Development and Education board connected to a computer that has Quartus II software installed. A reader who does not have access to the DE2-115 board will still find the tutorial useful to learn how the FPGA programming and configuration task is performed.

The screen captures in the tutorial were obtained using the Quartus II version 9.1; if other versions of the software are used, some of the images may be slightly different.

Contents:

Computer Aided Design (CAD) software makes it easy to implement a desired logic circuit by using a programmable logic device, such as a field programmable gate array (FPGA) chip. A typical FPGA CAD flow is illustrated in Figure 1.

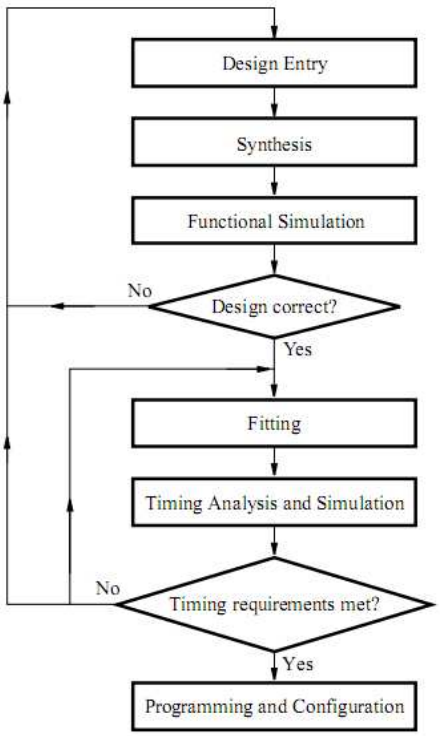


Figure 1.Typical CAD flow.

The CAD flow involves the following steps:

- **Design Entry** – the desired circuit is specified either by means of a schematic diagram, or by using a hardware description language, such as Verilog or VHDL.
- **Synthesis** – the entered design is synthesized into a circuit that consists of the logic elements (LEs) provided in the FPGA chip
- **Functional Simulation** – the synthesized circuit is tested to verify its functional correctness; this simulation does not take into account any timing issues
- **Fitting** – the CAD Fitter tool determines the placement of the LEs defined in the netlist into the LEs in an actual FPGA chip; it also chooses routing wires in the chip to make the required connections between specific LEs
- **Timing Analysis** – propagation delays along the various paths in the fitted circuit are analyzed to provide an indication of the expected performance of the circuit
- **Timing Simulation** – the fitted circuit is tested to verify both its functional correctness and timing
- **Programming and Configuration** – the designed circuit is implemented in a physical FPGA chip by programming the configuration switches that configure the LEs and establish the required wiring connections

This tutorial introduces the basic features of the Quartus II software. It shows how the software can be used to design and implement a circuit specified by means of a schematic diagram. It makes use of the graphical user interface to invoke the Quartus II commands. Doing this tutorial, the reader will learn about:

- **Creating a project**
- **Entering a schematic diagram**
- **Synthesizing a circuit from the schematic diagram**
- **Fitting a synthesized circuit into an Altera FPGA**
- **Assigning the circuit inputs and outputs to specific pins on the FPGA**
- **Simulating the designed circuit**
- **Programming and configuring the FPGA chip on Altera's DE2-115 board**

1 Getting Started

Each logic circuit, or subcircuit, being designed with Quartus II software is called a project. The software works on one project at a time and keeps all information for that project in a single directory (folder) in the file system.

To begin a new logic circuit design, the first step is to create a directory to hold its files. To hold the design files for this tutorial, we will use a directory introtutorial. The running example for this tutorial is a simple circuit for two-way light control.

Start the Quartus II software. You should see a display similar to the one in Figure 2. This display consists of several windows that provide access to all the features of Quartus II software, which the user selects with the computer mouse. Most of the commands provided by Quartus II software can be accessed by using a set of menus that are located below the title bar. For example, in Figure 2 clicking the left mouse button on the menu name **File** opens the menu shown in Figure 3. Clicking the left mouse button on the entry **Exit** exits from Quartus II software. In general, whenever the mouse is used to select something, the left button is used. Hence we will not normally specify which button to press. In the few cases when it is necessary to use the right mouse button, it will be specified explicitly.

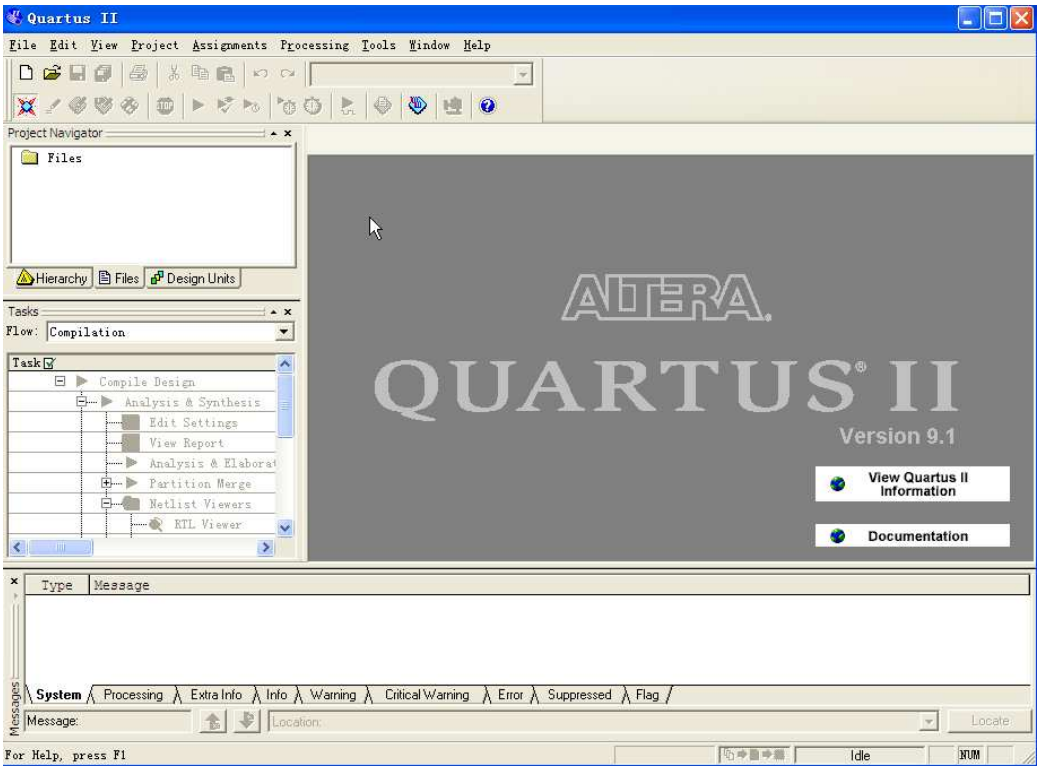


Figure 2.The main Quartus II display.

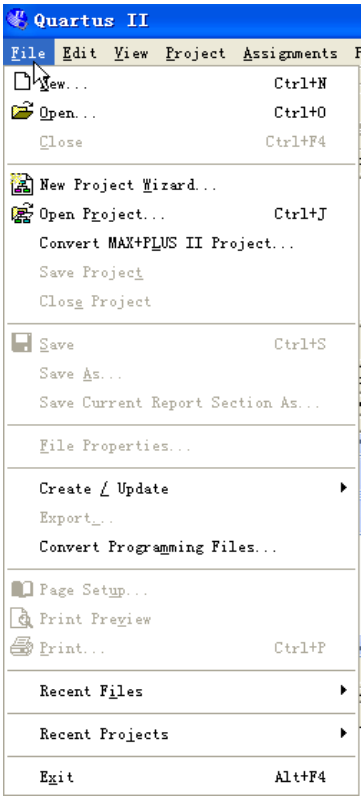


Figure 3.An example of the File menu.

For some commands it is necessary to access two or more menus in sequence. We use the convention **Menu1 > Menu2 > Item to indicate** that to select the desired command the user should first click the left mouse button on **Menu1**, then within this menu click on **Menu2**, and then within Menu2 click on Item. For example, **File > Exit** uses the mouse to exit from the system. Many commands can be invoked by clicking on an icon displayed in one of the toolbars. To see the command associated with an icon, position the mouse over the icon and a tooltip will appear that displays the command name.

1.1 Quartus II Online Help

Quartus II software provides comprehensive online documentation that answers many of the questions that may arise when using the software. The documentation is accessed from the menu in the **Help** window. To get some idea of the extent of documentation provided, it is worthwhile for the reader to browse through the Help menu. For instance, selecting **Help > How to Use Help** gives an indication of what type of help is provided.

The user can quickly search through the Help topics by selecting **Help > Search**, which opens a dialog box into which key words can be entered. Another method, context-sensitive help, is provided for quickly finding documentation for specific topics. While using most applications, pressing the **F1** function key on the keyboard opens a Help display that shows the commands available for the application.

2 Starting a New Project

To start working on a new design we first have to define a new design project. Quartus II software makes the designer's task easy by providing support in the form of a wizard. Create a new project as follows:

1. Select **File > New Project Wizard** to reach the window in Figure 4, which indicates the capability of this wizard. You can skip this window in subsequent projects by checking the box Don't show me this introduction again. Press **Next** to get the window shown in Figure 5.

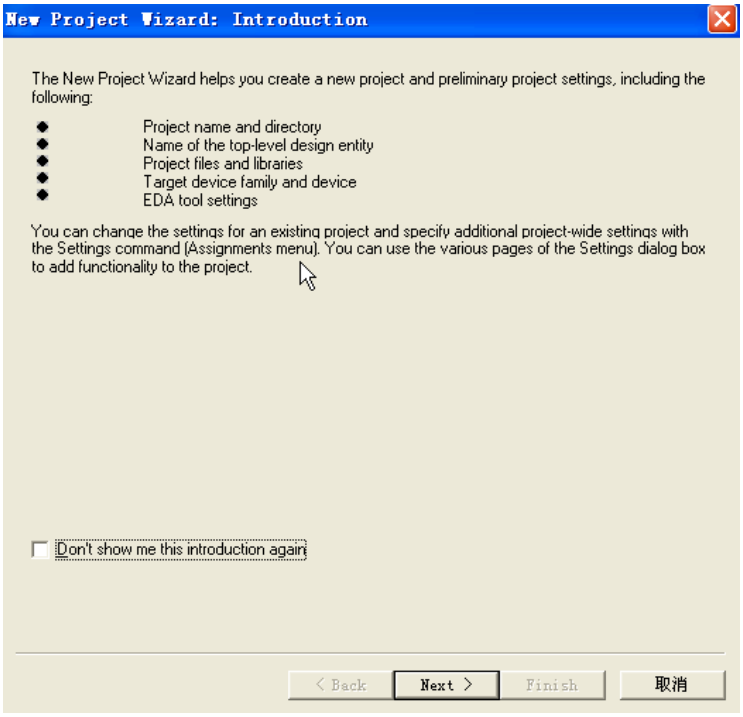


Figure 4.Tasks performed by the wizard.

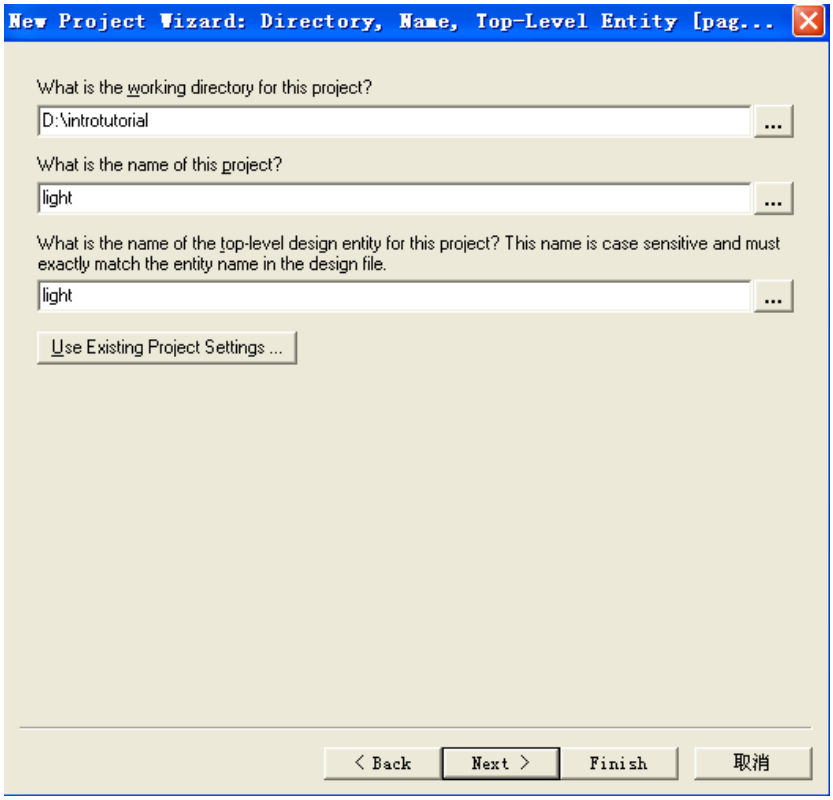


Figure 5.Creation of a new project.

2. Set the working directory to be introtutorial; of course, you can use some other directory name of your choice if you prefer. The project must have a name, which is usually the same as the top-level design entity that will be included in the project. Choose light as the name for both the project and the top-level entity, as shown in Figure 5. Press **Next**. Since we have not yet created the directory introtutorial, Quartus II software displays the pop-up box in Figure 6 asking if it should create the desired directory. Click **Yes**, which leads to the window in Figure 7.

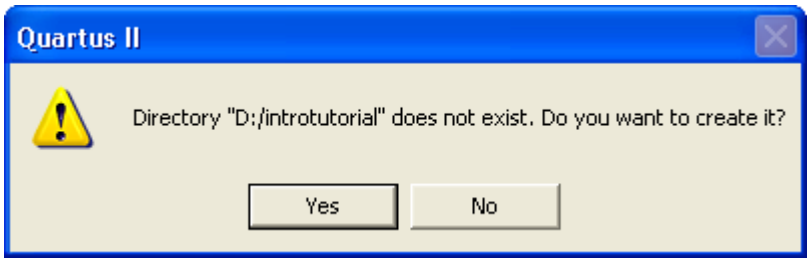


Figure 6.Quartus II software can create a new directory for the project.

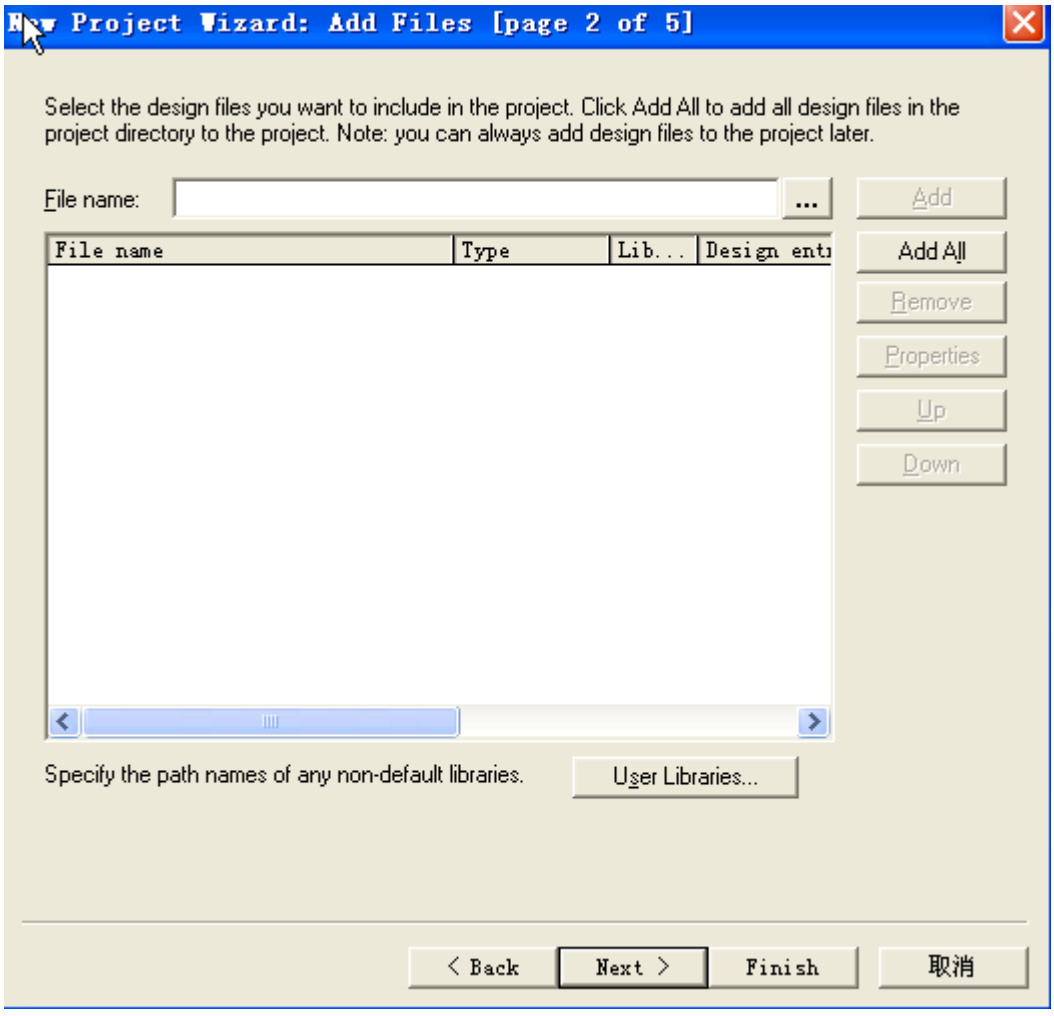


Figure 7.The wizard can include user-specified design files.

3. The wizard makes it easy to specify which existing files (if any) should be included in the project. Assuming that we do not have any existing files, click **Next**, which leads to the window in Figure 8.

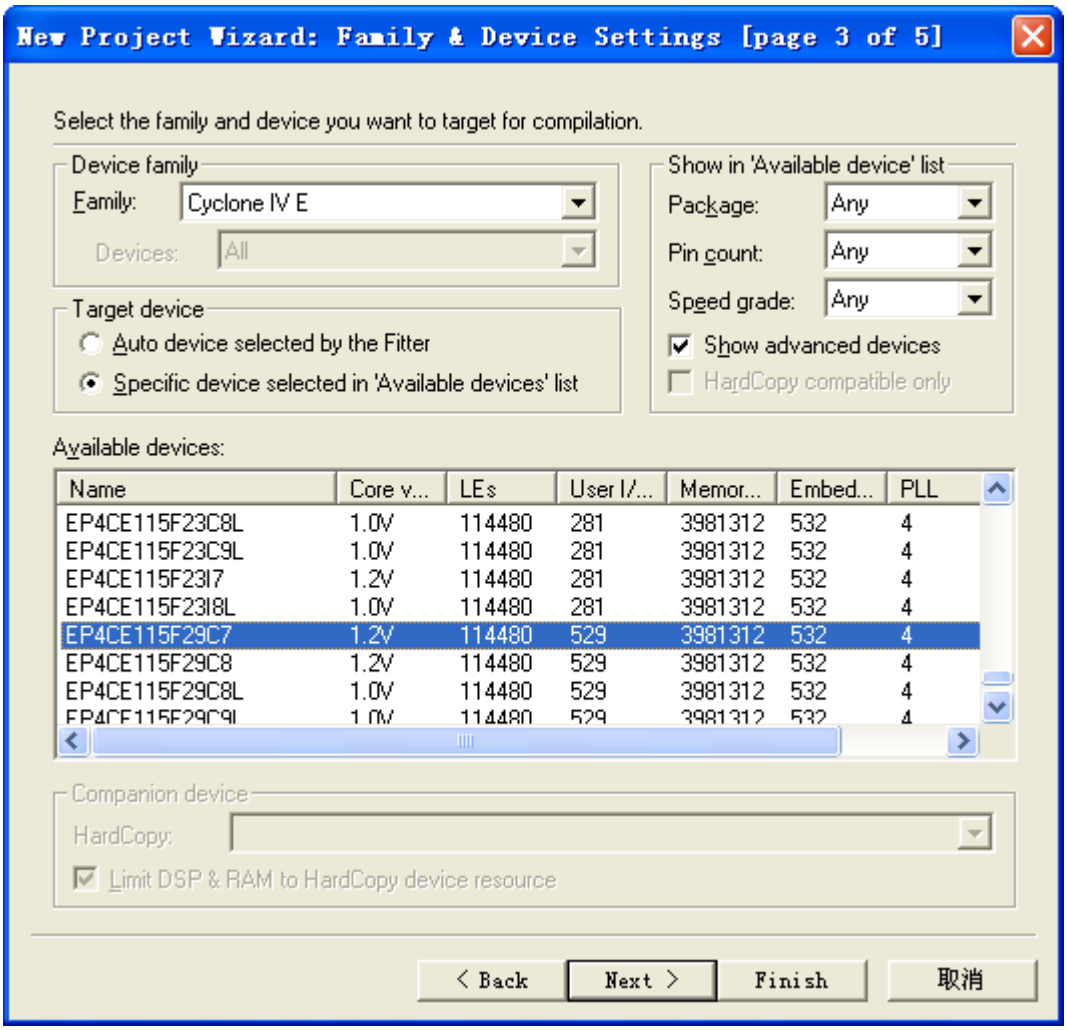


Figure 8.Choose the device family and a specific device.

4. We have to specify the type of device in which the designed circuit will be implemented. Choose Cyclone IV as the target device family. We can let Quartus II software select a specific device in the family, or we can choose the device explicitly. We will take the latter approach. From the list of available devices, choose the device called EP4CE115F29C7 which is the FPGA used on Altera's DE2-115 board. Press **Next**, which opens the window in Figure 9.

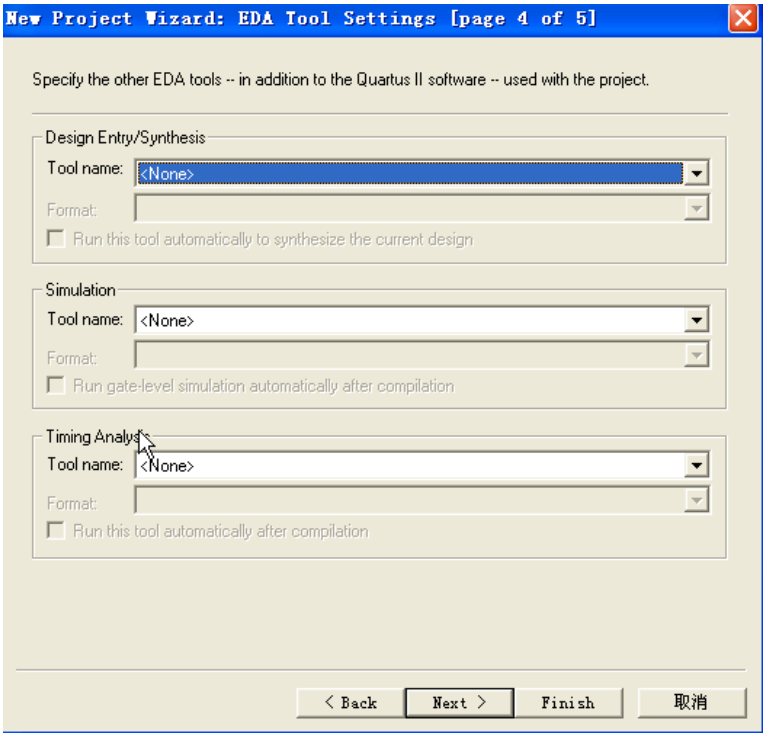


Figure 9.Other EDA tools can be specified.

5. The user can specify any third-party tools that should be used. A commonly used term for CAD software for electronic circuits is EDA tools, where the acronym stands for Electronic Design Automation. This term is used in Quartus II messages that refer to third-party tools, which are the tools developed and marketed by companies other than Altera. Since we will rely solely on Quartus II tools, we will not choose any other tools. Press **Next**.

6. A summary of the chosen settings appears in the screen shown in Figure 10. Press **Finish**, which returns to the main Quartus II window, but with light specified as the new project, in the display title bar, as indicated in Figure 11.

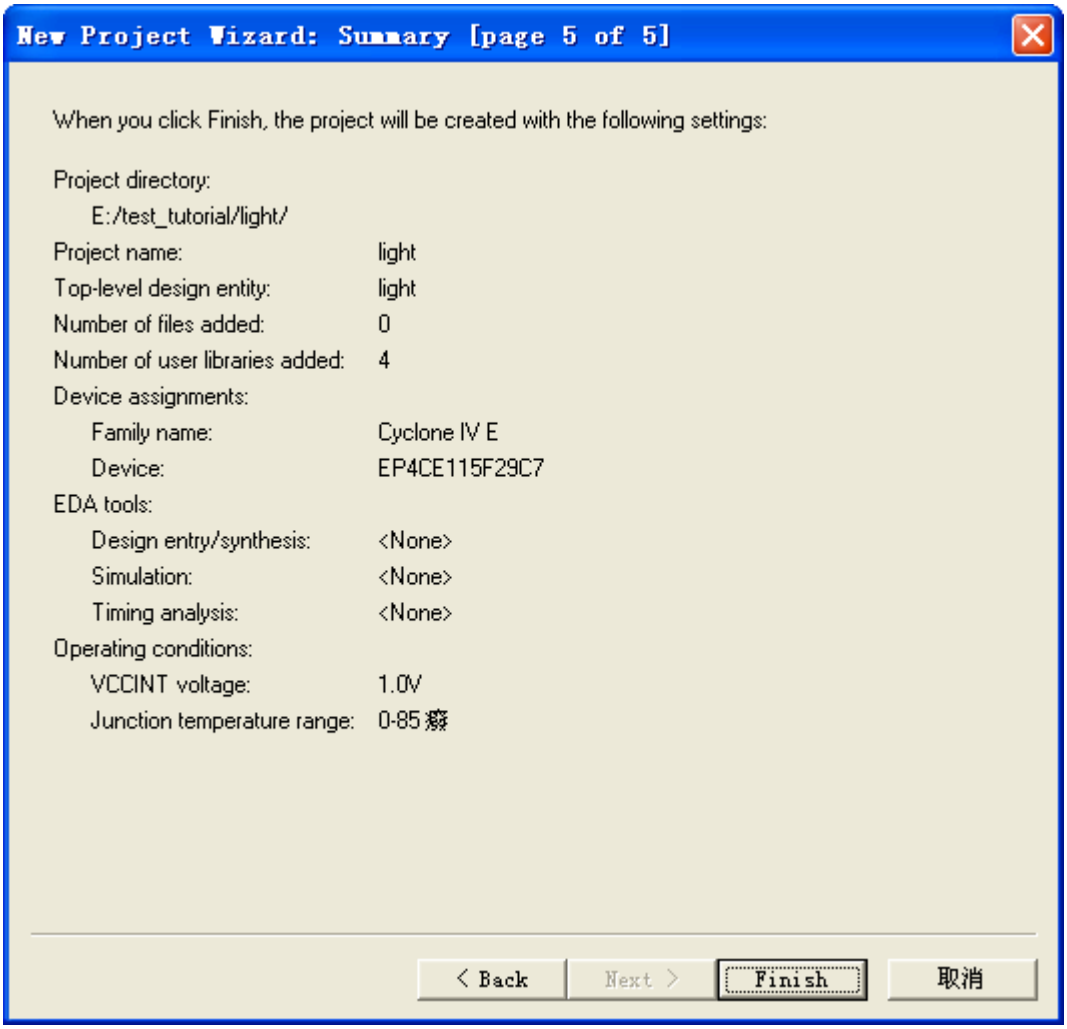


Figure 10.Summary of the project settings.

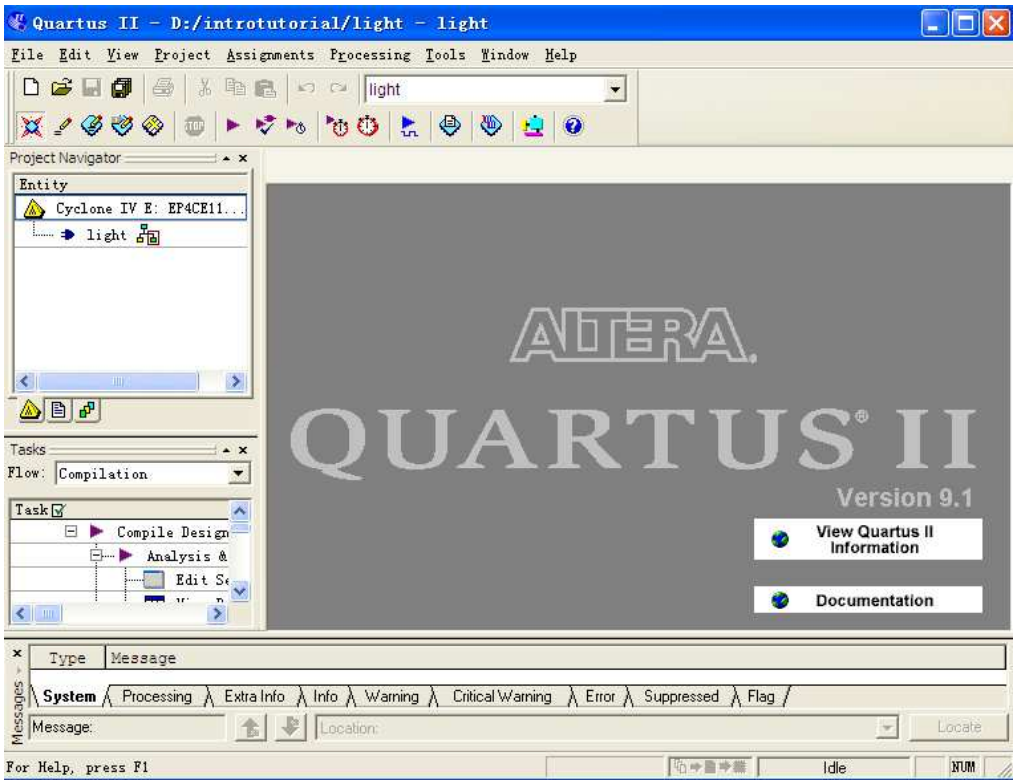
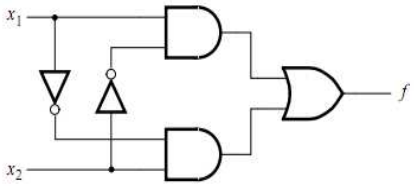


Figure 11.The Quartus II display for the created project.

3 Design Entry Using the Graphic Editor

As a design example, we will use the two-way light controller circuit shown in Figure 12. The circuit can be used to control a single light from either of the two switches, x1and x2, where a closed switch corresponds to the logic value 1. The truth table for the circuit is also given in the figure. Note that this is just the Exclusive-OR function of the inputs x1and x2, but we will implement it using the gates shown.



x_1	x_2	f
0	0	0
0	1	1
1	0	1
1	1	0

Figure 12.The light controller circuit.

The Quartus II Graphic Editor can be used to specify a circuit in the form of a block diagram. Select **File > New** to get the window in Figure 13, choose Block Diagram/Schematic File, and click OK. This opens the Graphic Editor window. The first step is to specify a name for the file that will be created. Select **File > Save As** to open the pop-up box depicted in Figure 14. In the box labeled **Save as type choose Block Diagram/Schematic File** (*.bdf). In the box labeled **File name** type light, to match the name given in Figure 5, which was specified when the project was created. Put a checkmark in the box **Add file to current project**. Click

Save, which puts the file into the directory introtutorial and leads to the Graphic Editor window displayed in Figure 15.

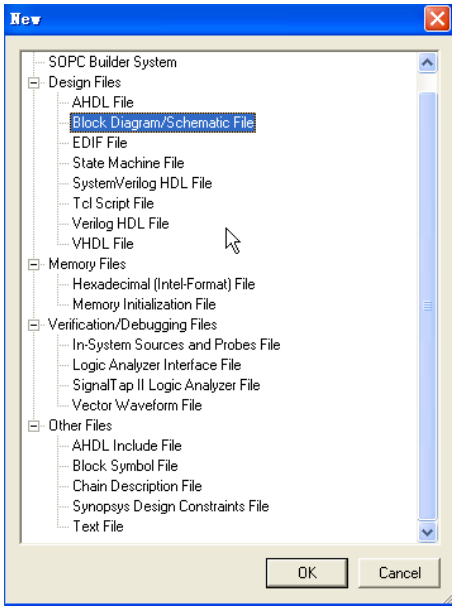


Figure 13.Choose to prepare a block diagram.

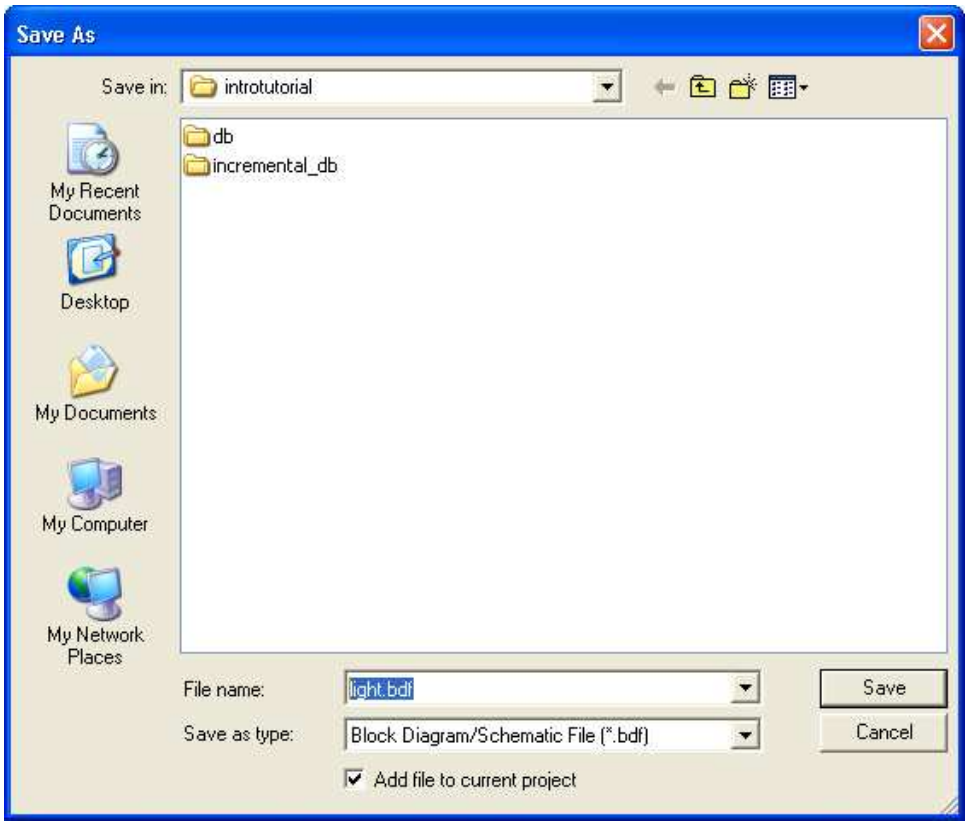


Figure 14.Name the file.

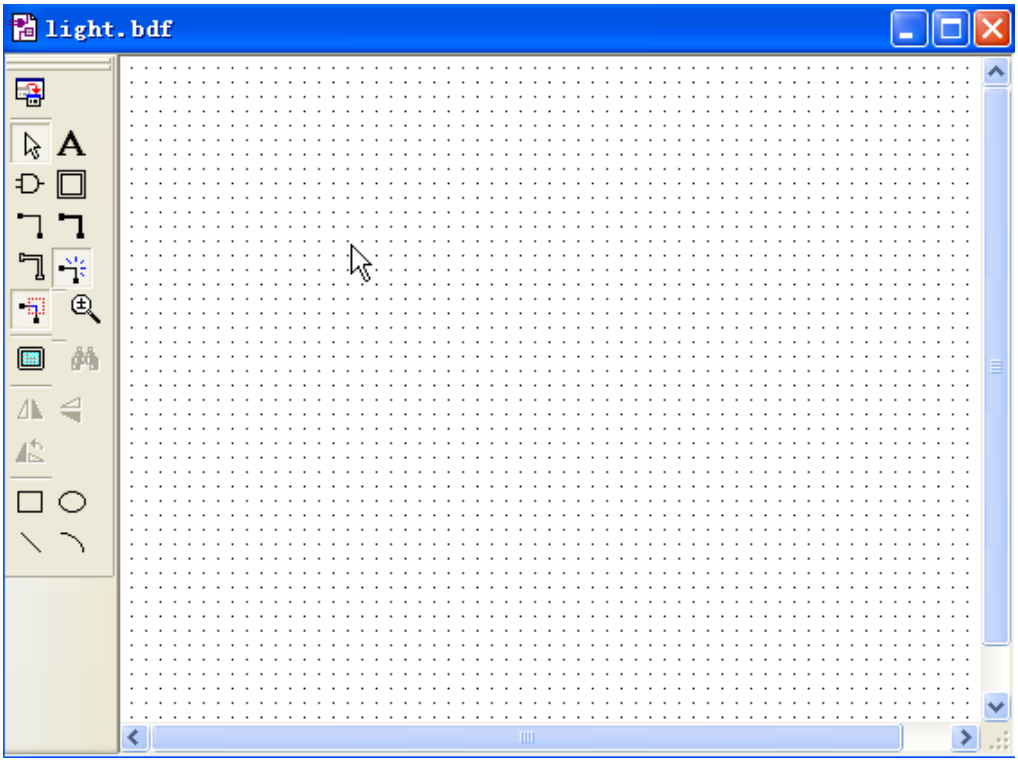


Figure 15.Graphic Editor Window.

3.1 Importing Logic-Gate Symbols

The Graphic Editor provides a number of libraries which include circuit elements that can be imported into a schematic. Double-click on the blank space in the Graphic Editor window, or click on the icon in the toolbar that looks like an **AND** gate. A pop-up box in Figure 16 will appear.

Expand the hierarchy in the Libraries box as shown in the figure. First expand libraries, then expand the library primitives, followed by expanding the library logic which comprises the logic gates. Select **and2**, which is a two-input **AND** gate, and click **OK**. Now, the **AND** gate symbol will appear in the Graphic Editor window. Using the mouse, move the symbol to a desirable location and click to place it there. Import the second **AND** gate, which can be done simply by positioning the mouse pointer over the existing **AND-gate** symbol, right-clicking, and dragging to make a copy of the symbol. A symbol in the Graphic Editor window can be moved by clicking on it and dragging it to a new location with the mouse button pressed. Next, select **or2** from the library and import the **OR** gate into the diagram. Then, select not and import two instances of the **NOT** gate. Rotate the **NOT** gates into proper position by using the “Rotate left 90" icon. Arrange the gates as shown in Figure 17.

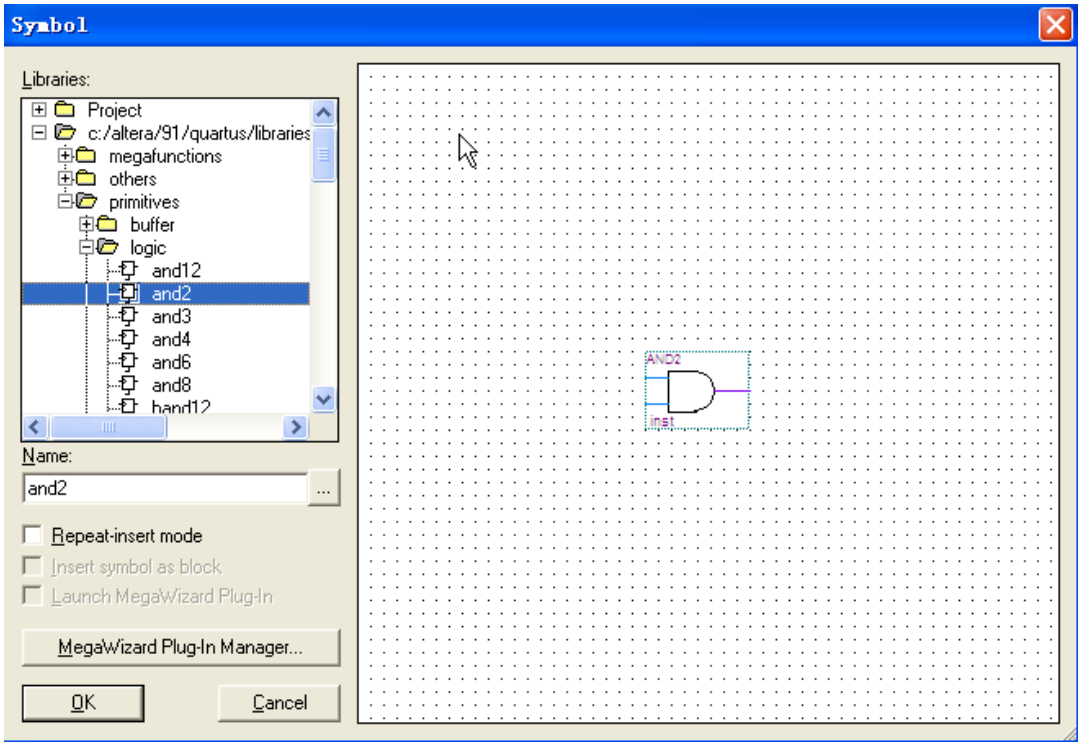


Figure 16.Choose a symbol from the library.

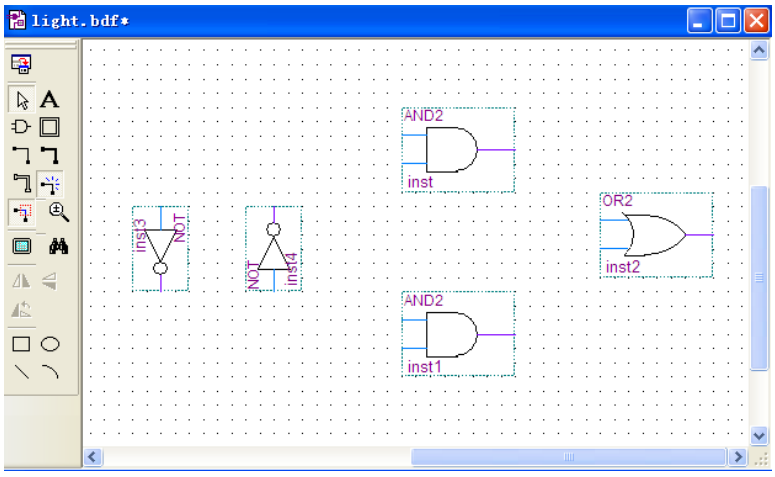


Figure 17.Import the gate symbols into the Graphic Editor window.

3.2 Importing Input and Output Symbols

Having entered the logic-gate symbols, it is now necessary to enter the symbols that

represent the input and output ports of the circuit. Use the same procedure as for importing the gates, but choose the port symbols from the library primitives/pin. Import two instances of the input port and one instance of the output port, to obtain the image in Figure 18.

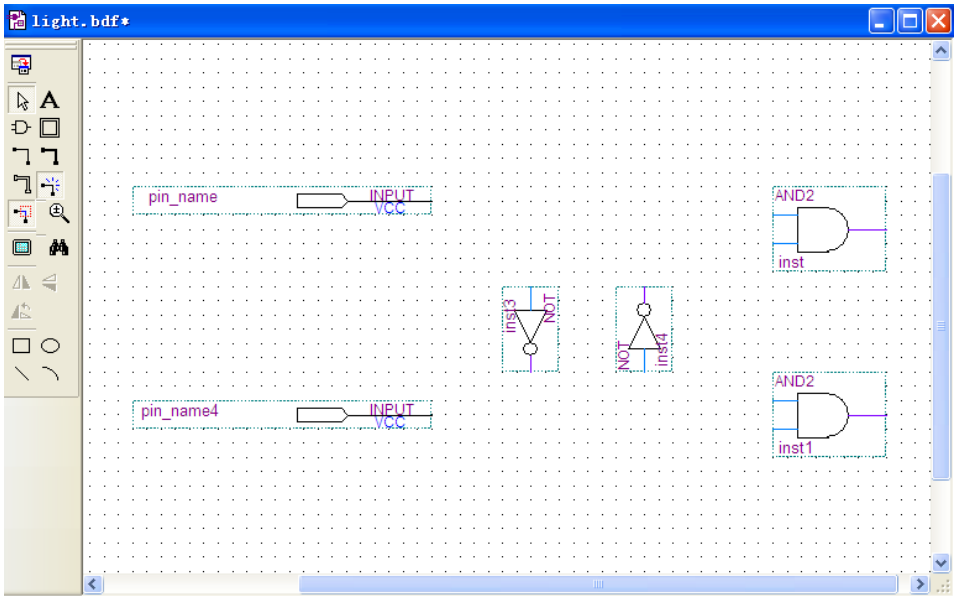


Figure 18.Import the input and output pins.

Assign names to the input and output symbols as follows. Point to the word pin_name on the top input symbol and double-click the mouse. The dialog box in Figure 19 will appear. Type the pin name, x1, and click **OK**. Similarly, assign the name x2 to the other input and f to the output.

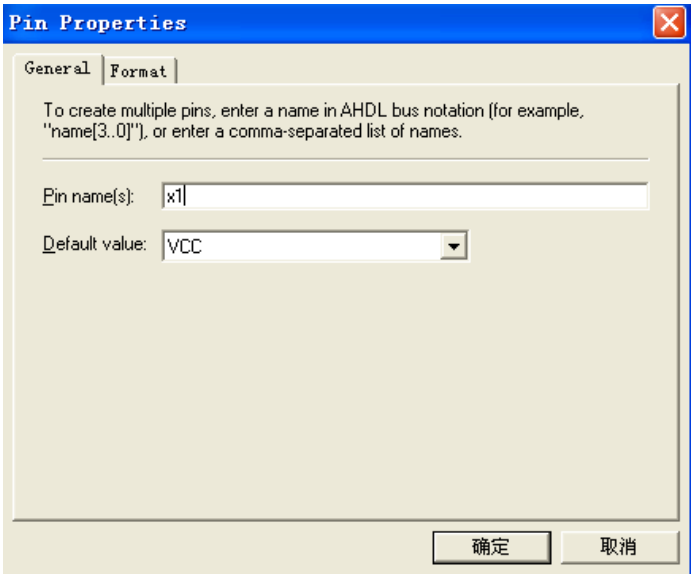


Figure 19.Naming of a pin.

3.3 Connecting Nodes with Wires

The symbols in the diagram have to be connected by drawing lines (wires). Click on the icon  in the toolbar to activate the Orthogonal Node Tool. Position the mouse pointer over the right edge of the x1 input pin. Click and hold the mouse button and drag the mouse to the right until the drawn line reaches the pinstub on the top input of the **AND** gate. Release the mouse button, which leaves the line connecting the two pinstubs. Next, draw a wire from the input pinstub of the leftmost **NOT** gate to touch the wire that was drawn above it. Note that a dot will appear indicating a connection between the two wires. Use the same procedure to draw the remaining wires in the circuit. If a mistake is made, a wire can be selected by clicking on it, and removed by pressing the Delete key on the keyboard. Upon completing the diagram, click on the  icon, to activate the Selection and Smart Drawing Tool.

Now, changes in the appearance of the diagram can be made by selecting a particular symbol or wire and either moving it to a different location or deleting it. The final diagram is shown in Figure 20; save it.

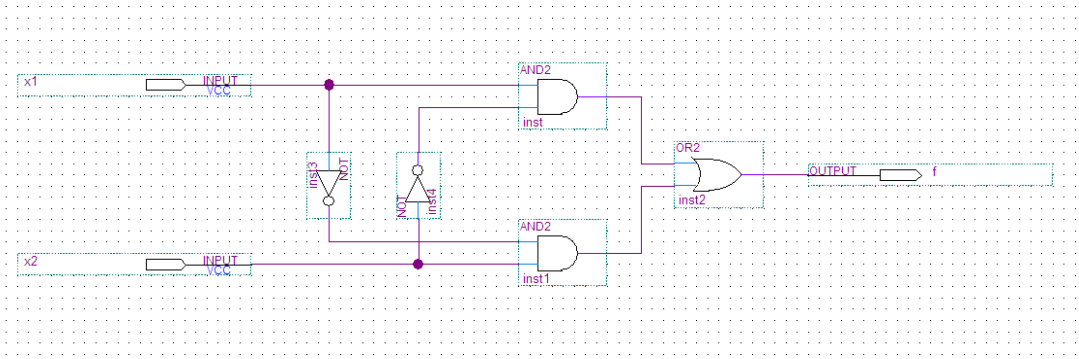


Figure 20.The completed schematic diagram.

4 Compiling the Designed Circuit

The entered schematic diagram file, light.bdf, is processed by several Quartus II tools that analyze the file, synthesize the circuit, and generate an implementation of it for the target chip. These tools are controlled by the application program called the Compiler.

Run the Compiler by selecting **Processing > Start Compilation**, or by clicking on the toolbar  icon that looks like a purple triangle. As the compilation moves through various stages, its progress is reported in a window on the left side of the Quartus II display. Successful (or unsuccessful) compilation is indicated in a pop-up box.

Acknowledge it by clicking **OK**, which leads to the Quartus II display in Figure 21. In the message window, at the bottom of the figure, various messages are displayed. In case of errors, there will be appropriate messages given.

When the compilation is finished, a compilation report is produced. A window showing this report is opened automatically, as seen in Figure 21. The window can be resized, maximized, or closed in the normal way, and it can be opened at any time either by selecting **Processing > Compilation Report** or by clicking on the icon .

The report includes a number of sections listed on the left side of its window. Figure 21 displays the Compiler Flow Summary section, which indicates that only one logic element and

three pins are needed to implement this tiny circuit on the selected FPGA chip.

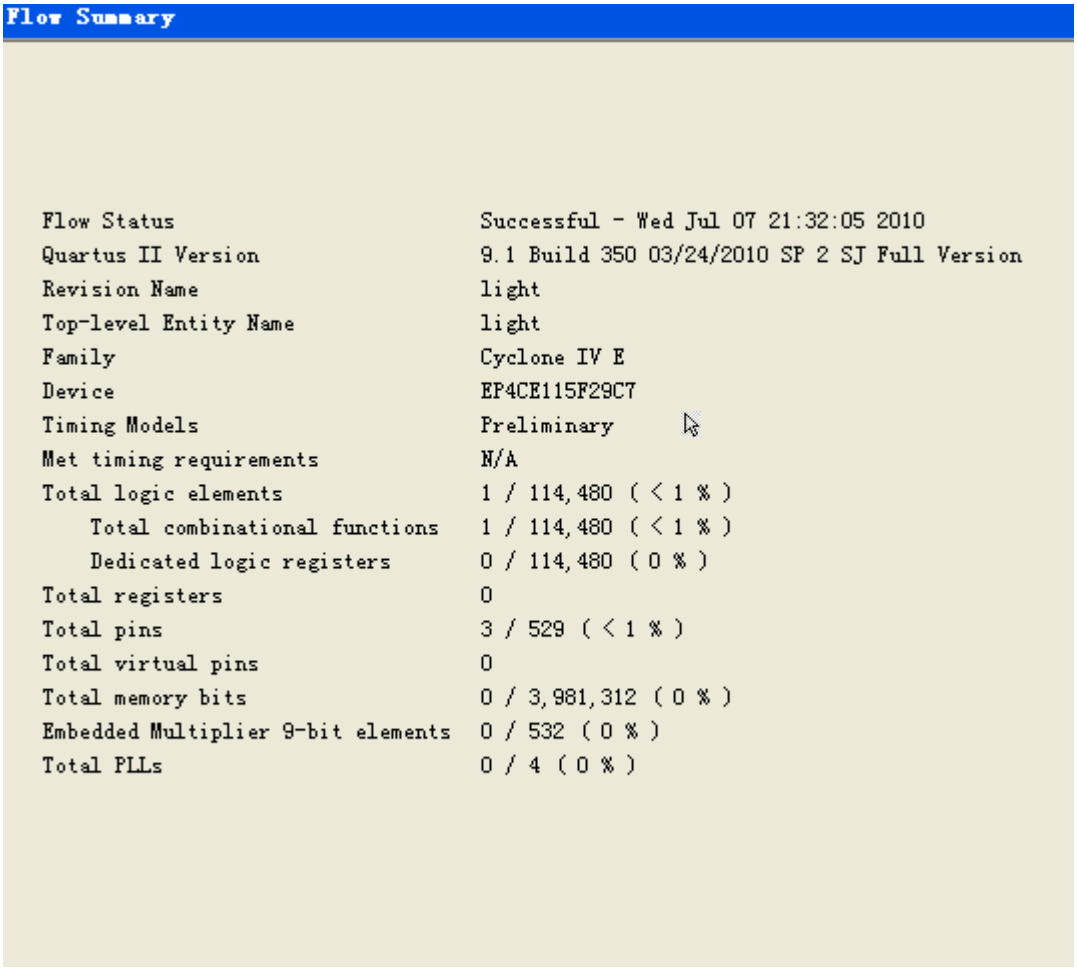


Figure 21.Display after a successful compilation.

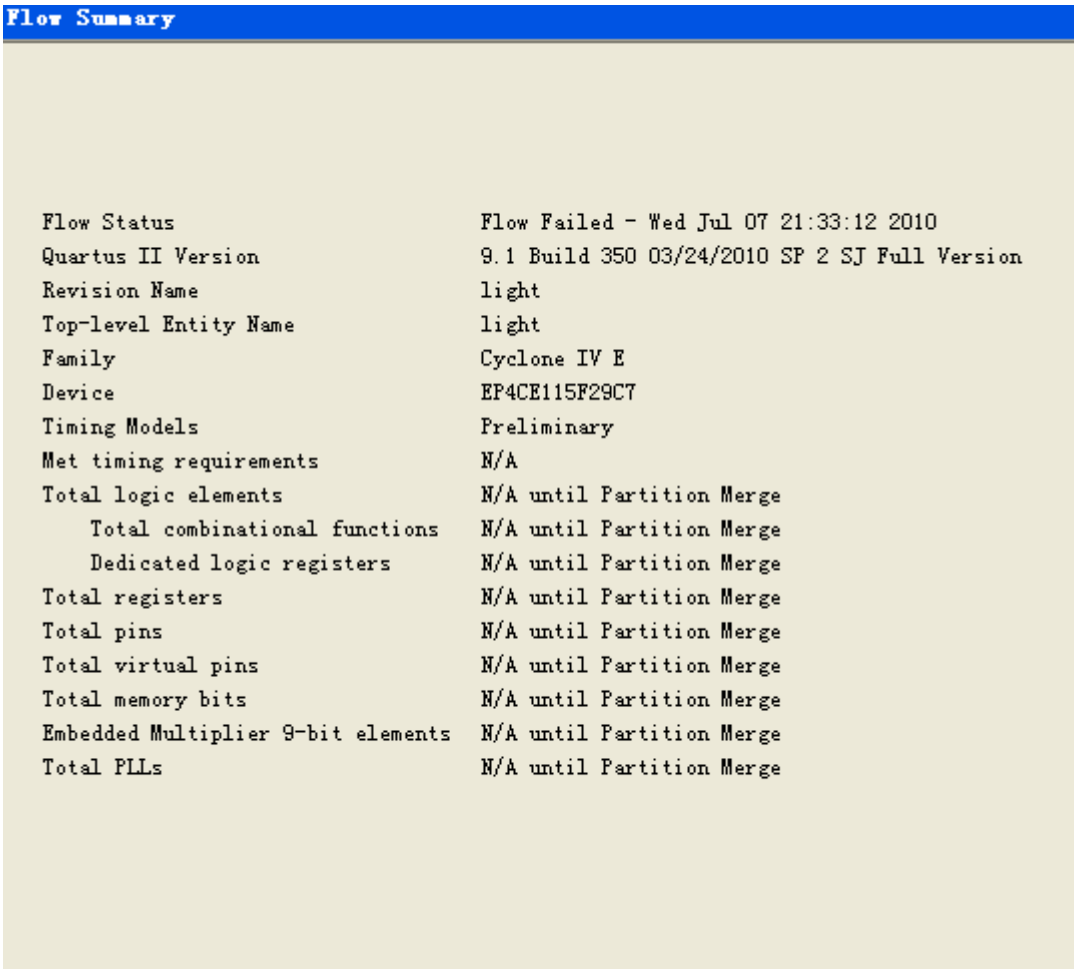
4.1 Errors

Quartus II software displays messages produced during compilation in the Messages window. If the block diagram design file is correct, one of the messages will state that the compilation was successful and that there are no errors.

If the Compiler does not report zero errors, then there is at least one mistake in the schematic entry. In this case a message corresponding to each error found will be displayed in the Messages window. Double-clicking on an error message will highlight the offending part of the circuit in the Graphic Editor window. Similarly, the Compiler may display some warning messages. Their details can be explored in the same way as in the case of error messages. The user can obtain more information about a specific error or warning message by selecting the message and pressing the **F1** function key.

To see the effect of an error, open the file light.bdf. Remove the wire connecting the output of the top **AND** gate to the **OR** gate. To do this, click on the icon, click the mouse on the wire to be removed (to select it) and press Delete. Compile the erroneous design by clicking on

the icon. A pop-up box will ask if the changes made to the light.bdf file should be saved; click **Yes**. After trying to compile the circuit, Quartus II software will display a pop-up box indicating that the compilation was not successful. Acknowledge it by clicking **OK**. The compilation report summary, given in Figure 22, now confirms the failed result. Expand the **Analysis & Synthesis** part of the report and then select **Messages** to have the messages displayed as shown in Figure 23. Double-click on the first error message, which states that one of the nodes is missing a source. Quartus II software responds by displaying the light.bdf schematic and highlighting the OR gate which is affected by the error, as shown in Figure 24. Correct the error and recompile the design.



Flow Summary

Flow Status	Flow Failed - Wed Jul 07 21:33:12 2010
Quartus II Version	9.1 Build 350 03/24/2010 SP 2 SJ Full Version
Revision Name	light
Top-level Entity Name	light
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Preliminary
Met timing requirements	N/A
Total logic elements	N/A until Partition Merge
Total combinational functions	N/A until Partition Merge
Dedicated logic registers	N/A until Partition Merge
Total registers	N/A until Partition Merge
Total pins	N/A until Partition Merge
Total virtual pins	N/A until Partition Merge
Total memory bits	N/A until Partition Merge
Embedded Multiplier 9-bit elements	N/A until Partition Merge
Total PLLs	N/A until Partition Merge

Figure 22.Compilation report for the failed design.

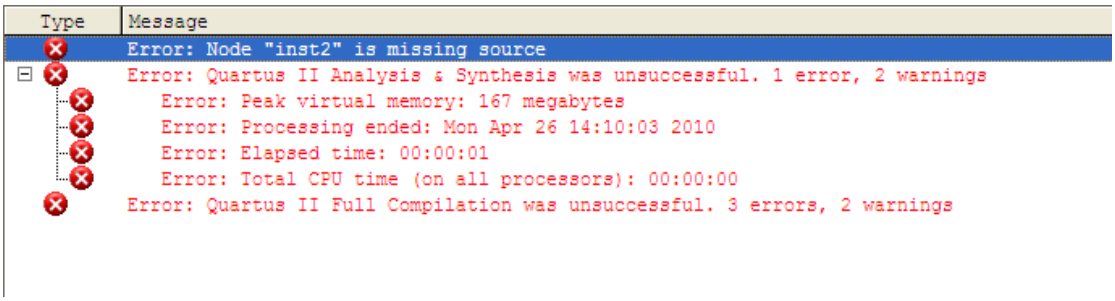


Figure 23.Error messages.

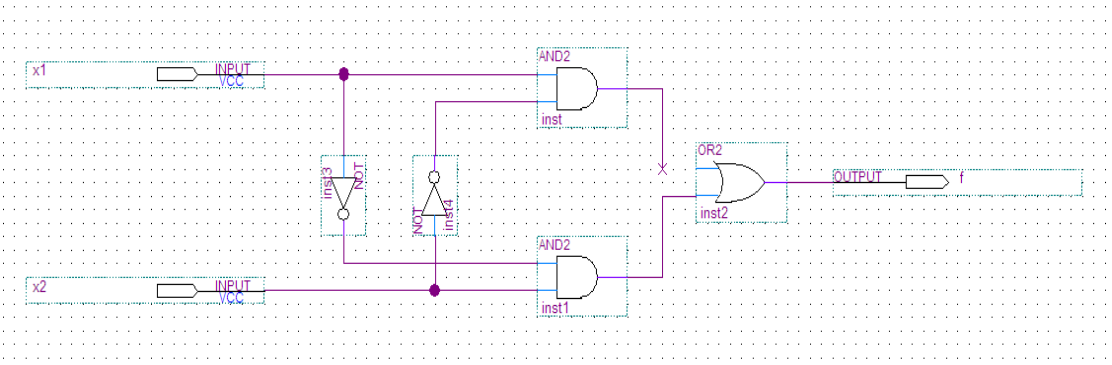


Figure 24.Identifying the location of the error.

5 Pin Assignment

During the compilation above, the Quartus II Compiler was free to choose any pins on the selected FPGA to serve as inputs and outputs. However, the DE2-115 board has hardwired connections between the FPGA pins and the other components on the board. We will use two toggle switches, labeled **SW0** and **SW1**, to provide the external inputs, x1 and x2, to our example circuit. These switches are connected to the FPGA pins **AB28** and **AC28**, respectively. We will connect the output f to the green light-emitting diode labeled LEDG0, which is hardwired to the FPGA pin **E21**.

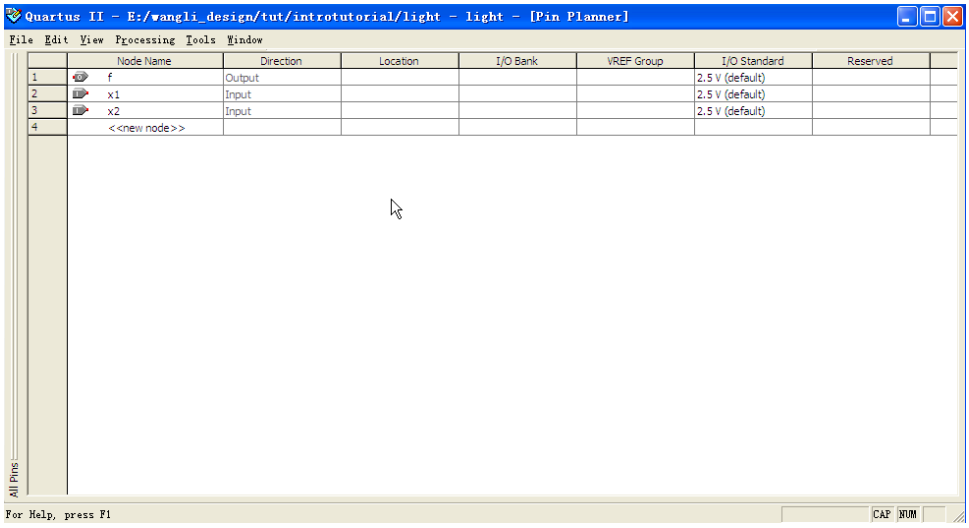


Figure 25.The Assignment Editor window.

Pin assignments are made by using the Assignment Editor. Select **Assignments > Pins** to reach the window in Figure 25. Under **Category** select **Pin**. Double-click on the entry <<new>> which is highlighted in blue in the column labeled To. The drop-down menu in Figure 26 will appear. Click on x1 as the first pin to be assigned; this will enter x1 in the displayed table. Follow this by double-clicking on the box to the right of this new x1 entry, in the column labeled Location. Now, the drop-down menu in Figure 27 appears. Scroll down and select **PIN_AB28**. Instead of scrolling down the menu to find the desired pin, you can just type the name of the pin (**AB28**) in the Location box. Use the same procedure to assign input x2 to pin

AC28 and output f to pin **E21**, which results in the image in Figure 28. To save the assignments made, choose **File > Save**. You can also simply close the Assignment Editor window, in which case a pop-up box will ask if you want to save the changes to assignments; click **Yes**. Recompile the circuit, so that it will be compiled with the correct pin assignments.

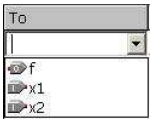


Figure 26.The drop-down menu displays the input and output names.

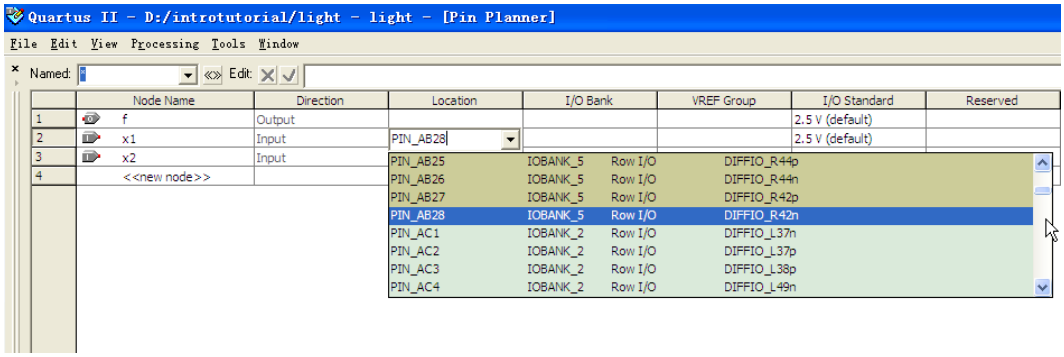


Figure 27.The available pins.

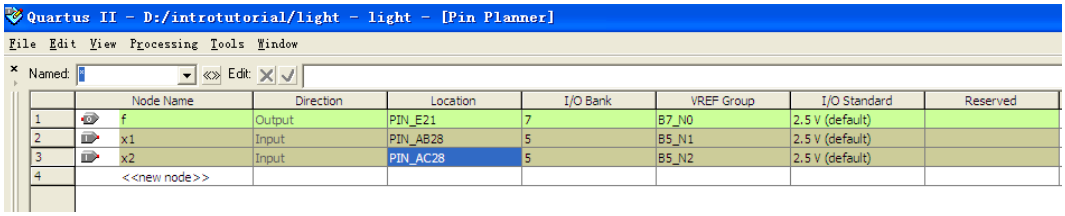


Figure 28.The complete assignment.

The DE2-115 board has fixed pin assignments. Having finished one design, the user will want to use the same pin assignment for subsequent designs. Going through the procedure described above becomes tedious if there are many pins used in the design. A useful Quartus II feature allows the user to both export and import the pin assignments from a special file format, rather than creating them manually using the Assignment Editor. A simple file format that can be used for this purpose is the comma separated value (CSV) format, which is a common text file format that contains comma-delimited values. This file format is often used in conjunction with the Microsoft Excel spreadsheet program, but the file can also be created by

hand using any plain ASCII text editor. The format for the file for our simple project is

To, Location
x1, PIN_AB28
x2, PIN_AC28
f, PIN_E21

By adding lines to the file, any number of pin assignments can be created. Such csv files can be imported into any design project.

If you created a pin assignment for a particular project, you can export it for use in a different project. To see how this is done, open again the Assignment Editor to reach the window in Figure 28. Now, select **File > Export** which leads to the window in Figure 29. Here, the file light.csv is available for export. Click on **Export**. If you now look in the directory introtutorial, you will see that the file light.csv has been created.

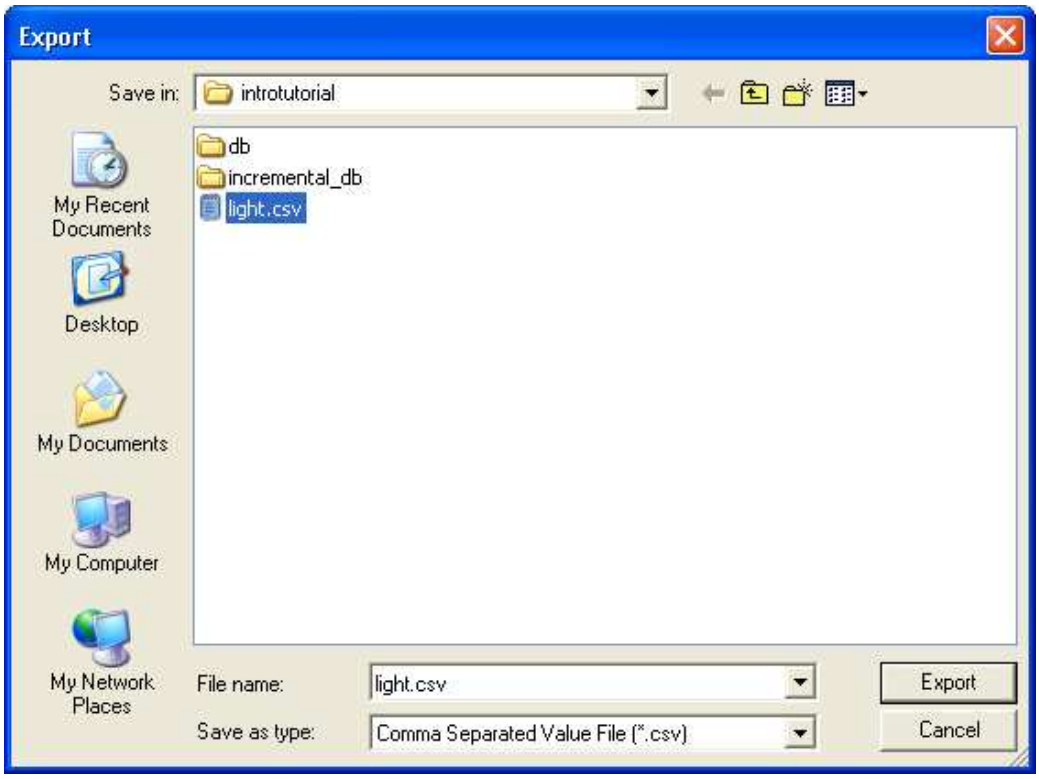


Figure 29.Exporting the pin assignment.

You can import a pin assignment by choosing **Assignments > Import Assignments**. This opens the dialogue in Figure 30 to select the file to import. Type the name of the file, including the csv extension and the full path to the directory that holds the file, in the File Name box and press **OK**. Of course, you can also browse to find the desired file.

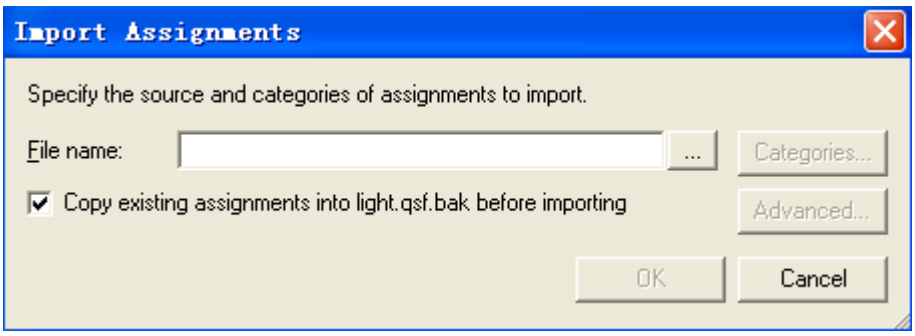


Figure 30.Importing the pin assignment.

For convenience when using large designs, all relevant pin assignments for the DE2-115 board are given in the file called DE2-115_pin_assignments.csv in the directory DE2-115_tutorials\design_files, which is included on the **CD-ROM** that accompanies the DE2-115 board and can also be found on Altera's DE2-115 web pages. This file uses the names found in the DE2-115 User Manual. If we wanted to make the pin assignments for our example circuit by importing this file, then we would have to use the same names in our Block Diagram/Schematic design file; namely, SW [0], SW [1] and LEDG [0] for x1, x2 and f, respectively. Since these signals are specified in the DE2-115_pin_assignments.csv file as elements of vectors SW and LEDG, we must refer to them in the same way in our design file. For example, in the DE2-115_pin_assignments.csv file the 18 toggle switches are called SW [17] to SW [0]. In a design file they can also be referred to as a vector SW [17..0].

6 Programming and Configuring the FPGA Device

The FPGA device must be programmed and configured to implement the designed circuit. The required configuration file is generated by the Quartus II Compiler's Assembler module. Altera's DE2-115 board allows the configuration to be done in two different ways, known as JTAG and AS modes. The configuration data is transferred from the host computer (which runs the Quartus II software) to the board by means of a cable that connects a USB port on the host computer to the leftmost USB connector on the board. To use this connection, it is necessary to have the USB-Blaster driver installed. If this driver is not already installed, consult the tutorial Getting Started with Altera's DE2-115 Board for information about installing the driver. Before using the board, make sure that the USB cable is properly connected and turn on the power supply switch on the board.

In the JTAG mode, the configuration data is loaded directly into the FPGA device. The acronym JTAG stands for Joint Test Action Group. This group defined a simple way for testing digital circuits and loading data into them, which became an IEEE standard. If the FPGA is configured in this manner, it will retain its configuration as long as the power remains turned on. The configuration information is lost when the power is turned off. The second possibility is to use the Active Serial (AS) mode. In this case, a configuration device that includes some flash memory is used to store the configuration data. Quartus II software places the configuration data into the configuration device on the DE2-115 board. Then, this data is loaded into the FPGA upon power-up or reconfiguration.

Thus, the FPGA need not be configured by the Quartus II software if the power is turned off and on. The choice between the two modes is made by the **RUN/PROG** switch on the DE2-115 board. The RUN position selects the JTAG mode, while the **PROG** position selects the AS mode.

6.1 JTAG Programming

The programming and configuration task is performed as follows. Flip the **RUN/PROG** switch into the RUN position. Select **Tools > Programmer** to reach the window in Figure 31. Here it is necessary to specify the programming hardware and the mode that should be used. If not already chosen by default, select JTAG in the Mode box. Also, if the USB-Blaster is not chosen by default, press the **Hardware Setup**... button and select the USB-Blaster in the window that pops up, as shown in Figure 32.

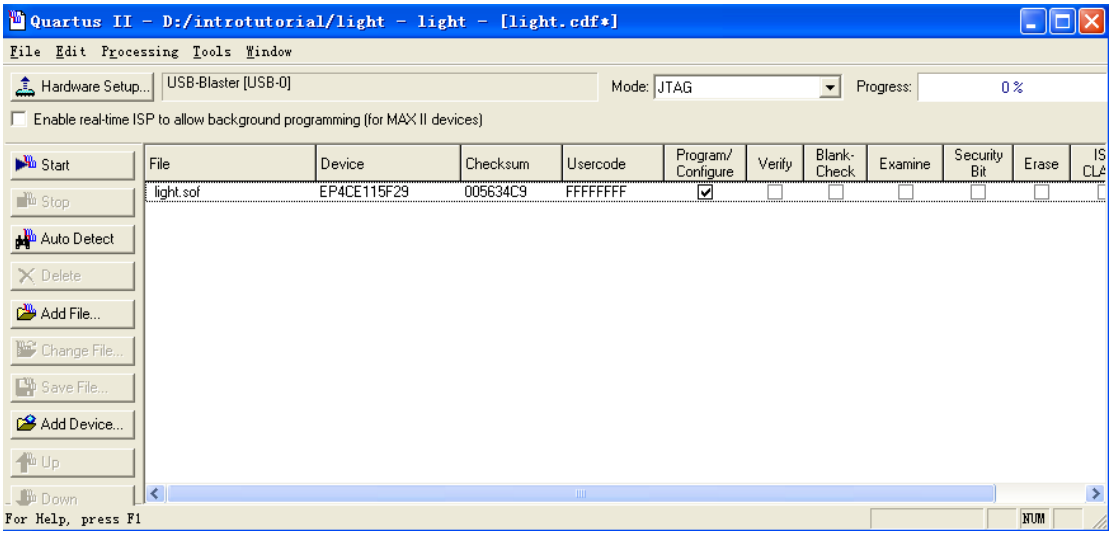


Figure 31.The Programmer window.

Observe that the configuration file light.sof is listed in the window in Figure 32. If the file is not already listed, then click **Add File** and select it. This is a binary file produced by the Compiler's Assembler module, which contains the data needed to configure the FPGA device. The extension .sof stands for SRAM Object File. Note also that the device selected is EP4CE115F29C7, which is the FPGA device used on the DE2-115 board. Click on the **Program/Configure** check box, as shown in Figure 33.

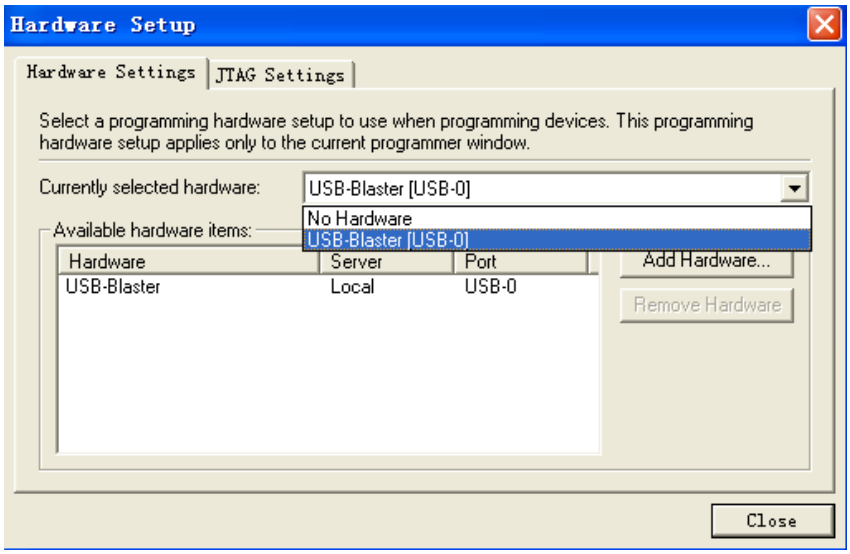


Figure 32.The Hardware Setup window.

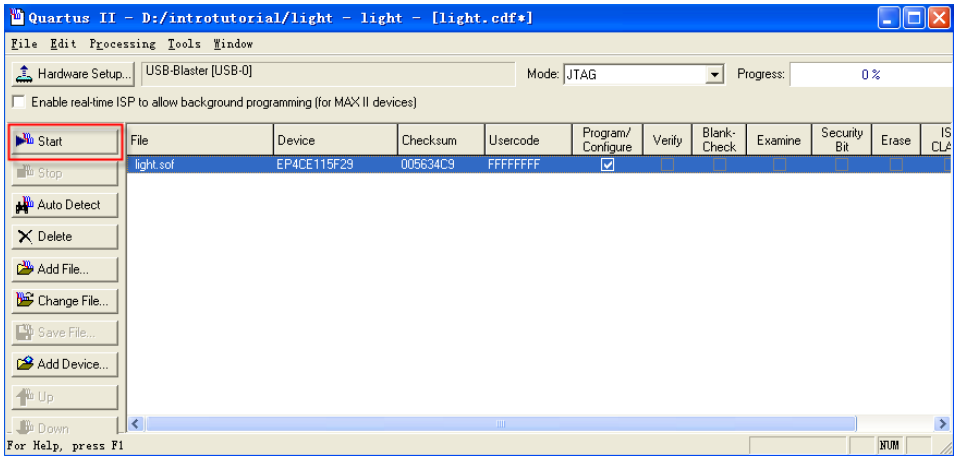


Figure 33.The updated Programmer window.

Now, press **Start** in the window in Figure 33. An LED on the board will light up when the configuration data has been downloaded successfully. If you see an error reported by Quartus II software indicating that programming failed, then check to ensure that the board is properly powered on.

6.2 Active Serial Mode Programming

In this case, the configuration data has to be loaded into the configuration device on the DE2-115 board, which is identified by the name EPCS64. To specify the required configuration device select **Assignments > Device**, which leads to the window in Figure 34. Click on the **Device & Pin Options** button to reach the window in Figure 35. Now, click on the **Configuration** tab to obtain the window in Figure 36. In the Configuration device box (which may be set to Auto) choose **EPCS64** and click **OK**. Upon returning to the window in Figure 37, click **OK**. Recompile the designed circuit.

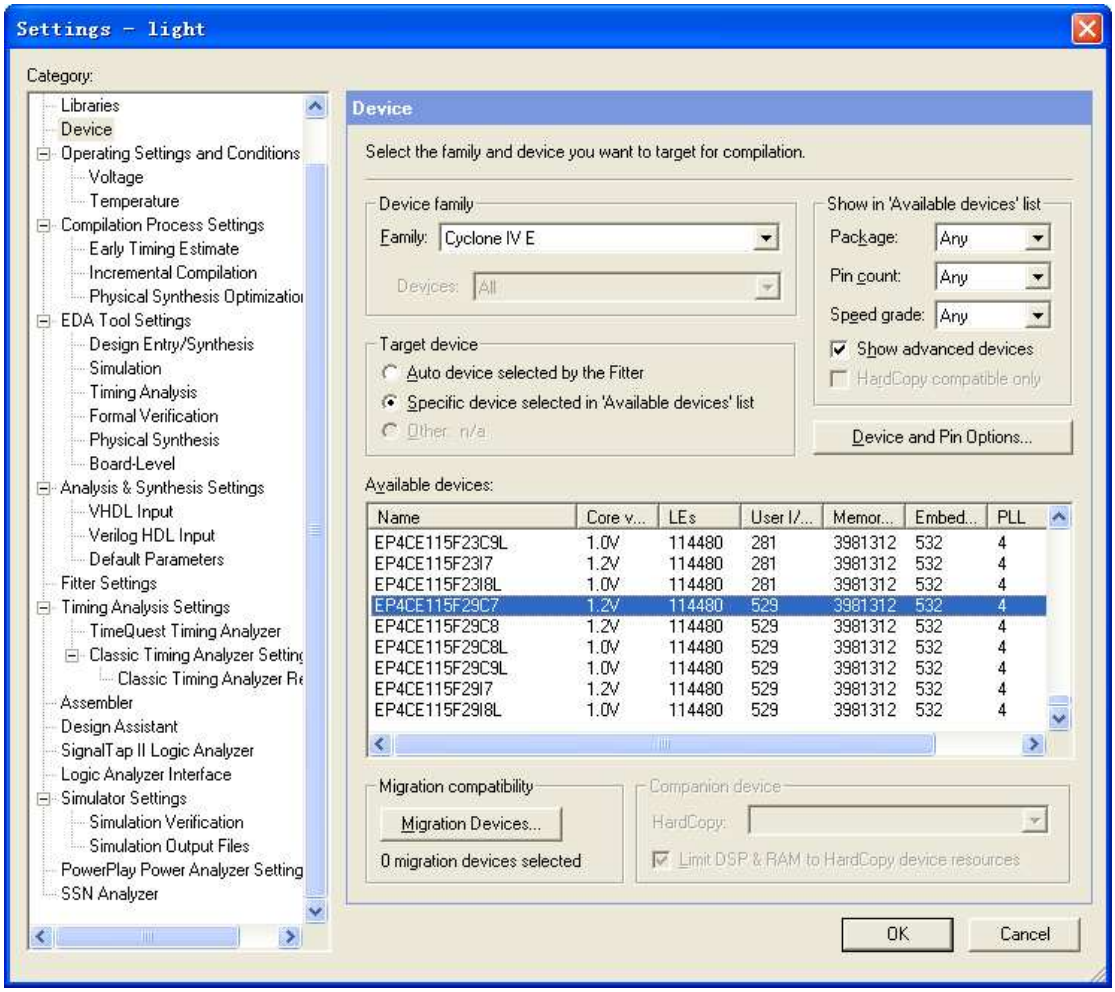


Figure 34.The Device Settings window.

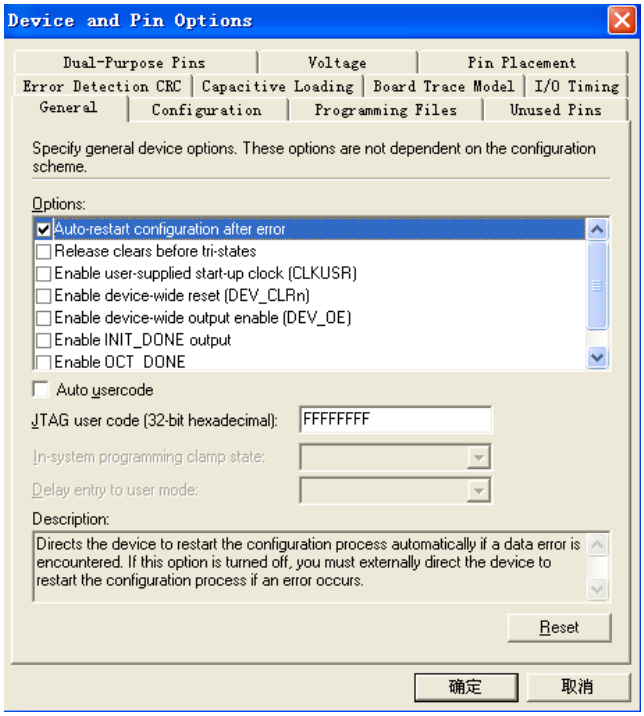


Figure 35.The Options window.

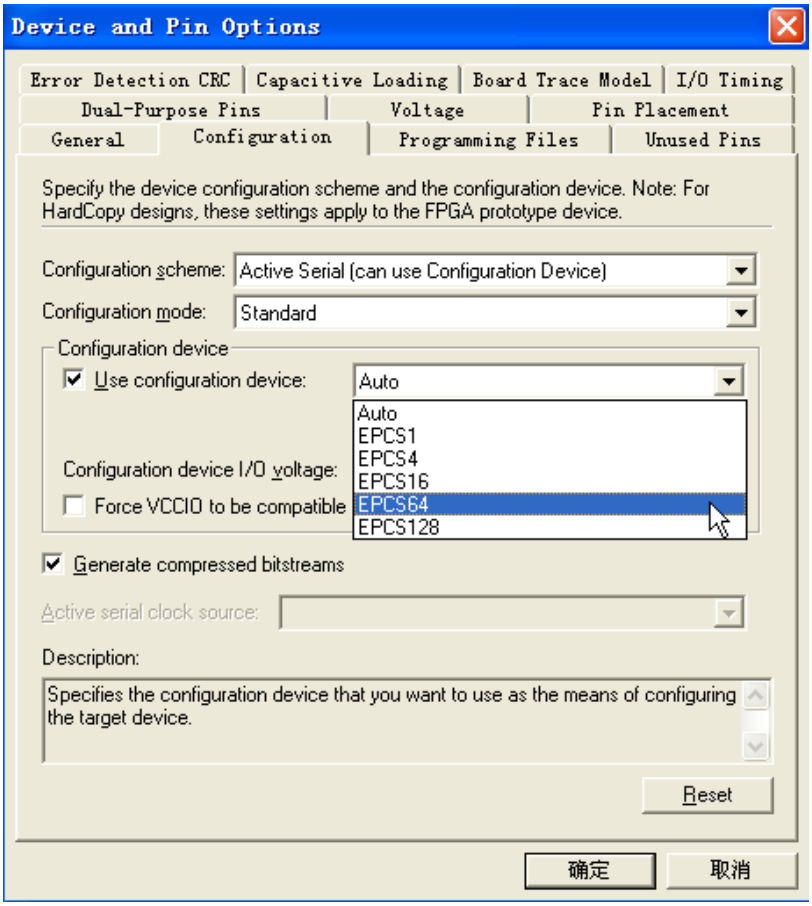


Figure 36.Specifying the configuration device.

The rest of the procedure is similar to the one described above for the JTAG mode. Select **Tools > Programmer** to reach the window in Figure 37. In the Mode box select Active Serial Programming. If you are changing the mode from the previously used JTAG mode, the pop-up box in Figure 37 will appear, asking if you want to clear all devices. Click **Yes**. Now, the Programmer window shown in Figure 38 will appear. Make sure that the Hardware Setup indicates the USB-Blaster. If the configuration file is not already listed in the window, press Add File. The pop-up box in Figure 39 will appear. Select the file **light.pof** in the directory introtutorial and click Open. As a result, the configuration file light.pof will be listed in the window. This is a binary file produced by the Compiler's Assembler module, which contains the data to be loaded into the EPCS64 configuration device. The extension .pof stands for Programmer Object File. Upon returning to the Programmer window, click on the **Program/Configure** check box, as shown in Figure 40.

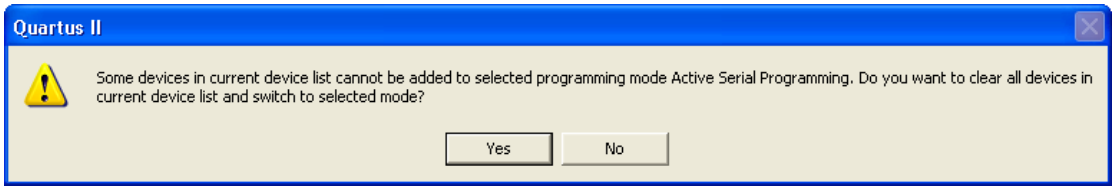


Figure 37.Clear the previously selected devices.

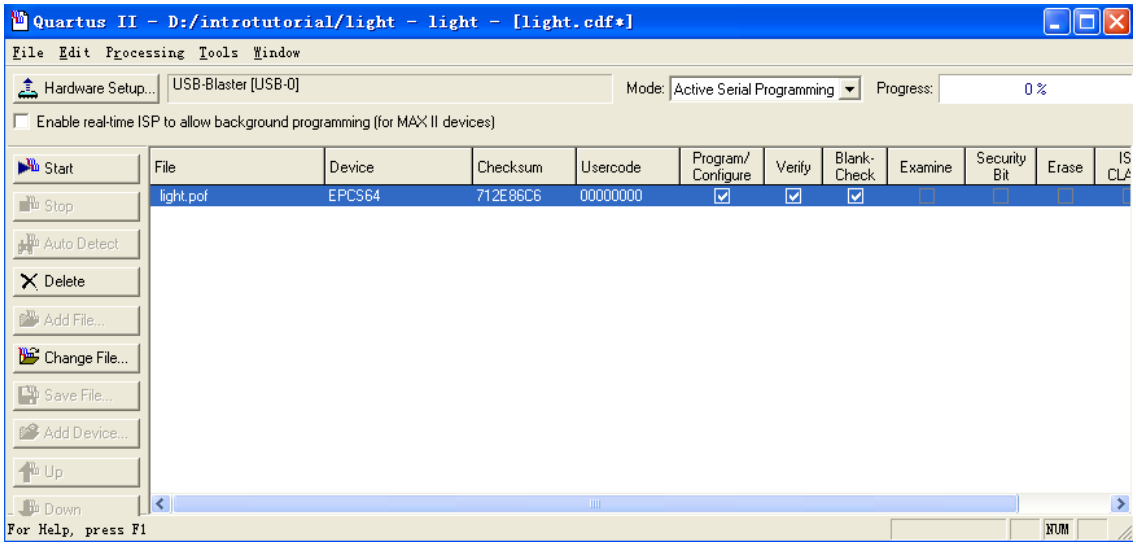


Figure 38.The Programmer window with Active Serial Programming selected.

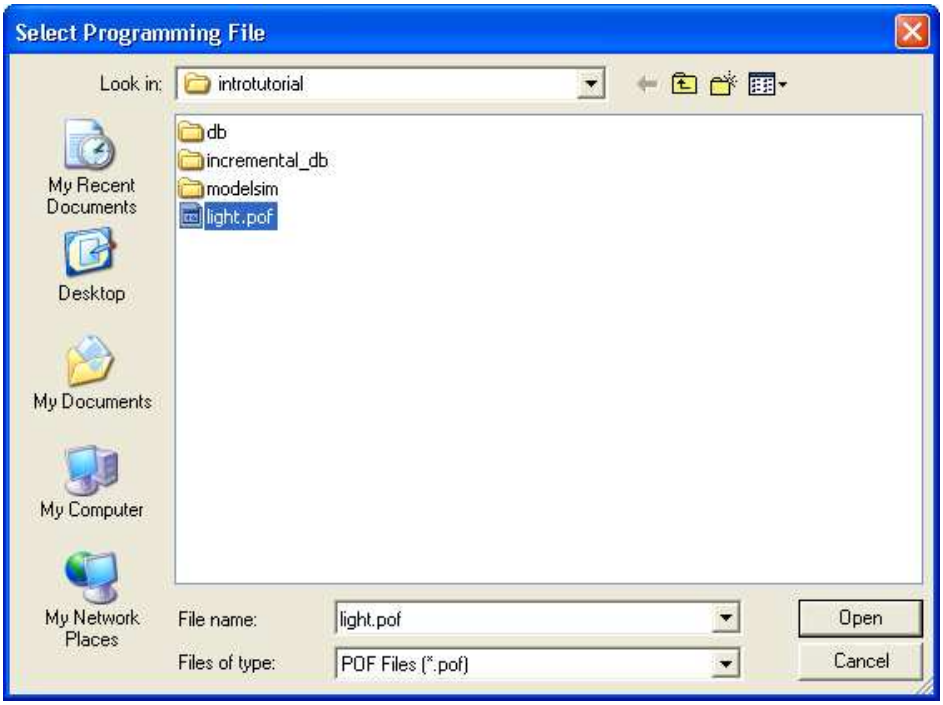


Figure 39.Choose the configuration file.

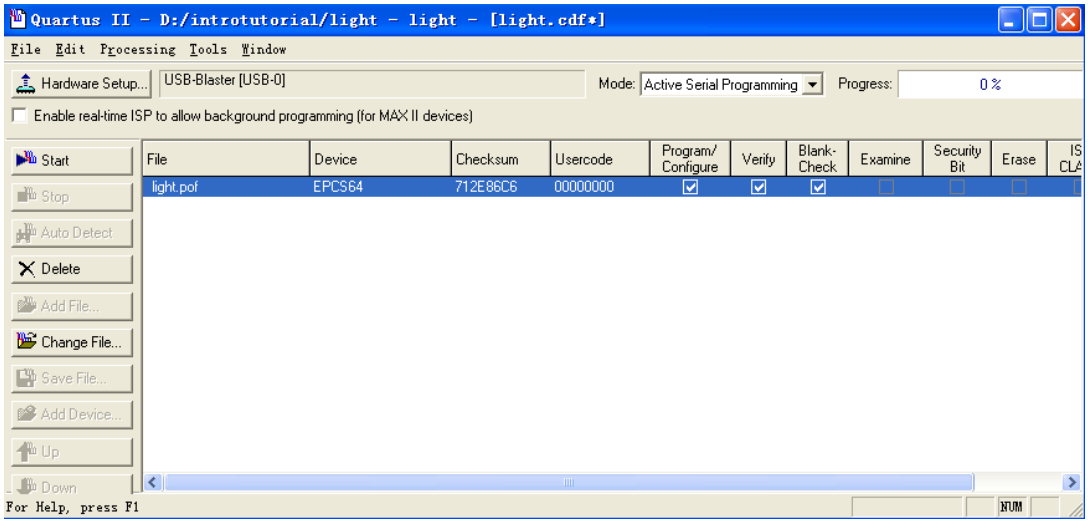


Figure 40.The updated Programmer window.

Flip the **RUN/PROG** switch on the DE2-115 board to the PROG position. Press **Start** in the window in Figure 40. An LED on the board will light up when the configuration data has been downloaded successfully.

7 Testing the Designed Circuit

Having downloaded the configuration data into the FPGA device, you can now test the implemented circuit. Flip the **RUN/PROG** switch to **RUN** position. Try all four valuations of the input variables x1 and x2, by setting the corresponding states of the switches **SW1** and **SW0**.

Verify that the circuit implements the truth table in Figure12.

If you want to make changes in the designed circuit, first close the Programmer window. Then make the desired changes in the **Block Diagram**/**Schematic** file, compile the circuit, and program the board as explained above.