Altera Quartus II Tutorial

Quartus II is a sophisticated CAD system. As most commercial CAD tools are continuously being improved and updated, Quartus II has gone through a number of releases. The version known as Quartus II 4.2 is used in this tutorial. For simplicity, in our discussion we will refer to this software package simply as Quartus II.

In this tutorial we introduce the design of logic circuits using Quartus II. Step-by-step instructions are presented for performing design entry with two methods: using schematic capture and writing VHDL code. The tutorial also illustrates the process of simulation.

1. Introduction

This tutorial assumes that the reader has access to a computer on which Quartus II is installed. Instructions for installing Quartus II are provided with the software. The Quartus II software will run on several different types of computer systems. For this tutorial a computer running a Microsoft operating system (Linux, Windows NT, Windows 2000, or Windows XP) is assumed. Although Quartus II operates similarly on all of the supported types of computers, there are some minor differences. A reader who is not using a Microsoft Windows operating system may experience some slight discrepancies from this tutorial. Examples of potential differences are the locations of files in the computer's file system and the exact appearance of windows displayed by the software. All such discrepancies are minor and will not affect the reader's ability to follow the tutorial.

This tutorial does not describe how to use the operating system provided on the computer. We assume that the reader already knows how to perform actions such as running programs, operating a mouse, moving, resizing, minimizing and maximizing windows, creating directories (folders) and files, and the like. A reader who is not familiar with these procedures will need to learn how to use the computer's operating system before proceeding.

Getting Started

Each logic circuit, or subcircuit, being designed in Quartus II is called a *project*. The software works on one project at a time and keeps all information for that project in a single directory in the file system (we use the traditional term *directory* for a location in the file system, but in Microsoft Windows the term *folder* is used). 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 *tutorial*. The location and name of the directory is not important; hence the reader may use any valid directory.

Start the Quartus II software. You should see a display similar to the one in Figure B.1. This display consists of several windows that provide access to all features of Quartus II, which the user selects with the computer mouse.

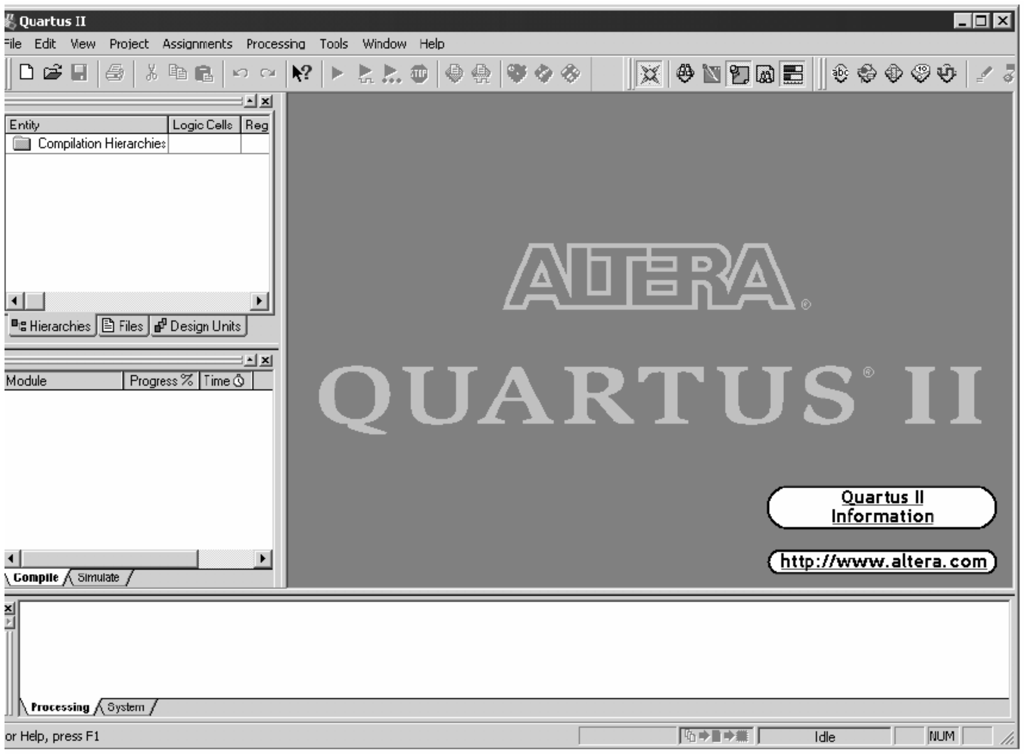


Figure 1: Altera Quartus II main window

Most of the commands provided by Quartus II can be accessed by using a set of menus that are located below the title bar. For example, in Figure 1 clicking the left mouse button on the menu named File opens the menu shown in Figure 2. Clicking the left mouse button on the item Exit exits from Quartus II. In general, whenever the mouse is employed 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. 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 (Menu1| Menu2) uses the mouse to exit from the Quartus II system.

Quartus II On-Line Help

Quartus II provides comprehensive on-line 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 topics. For instance, selecting Help | How to Use Help gives an indication of what type of help is provided.

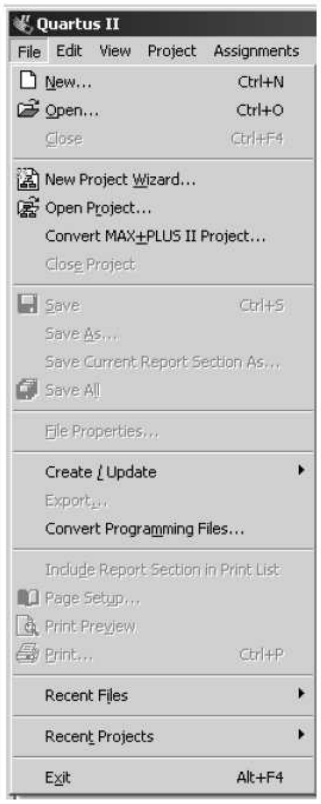
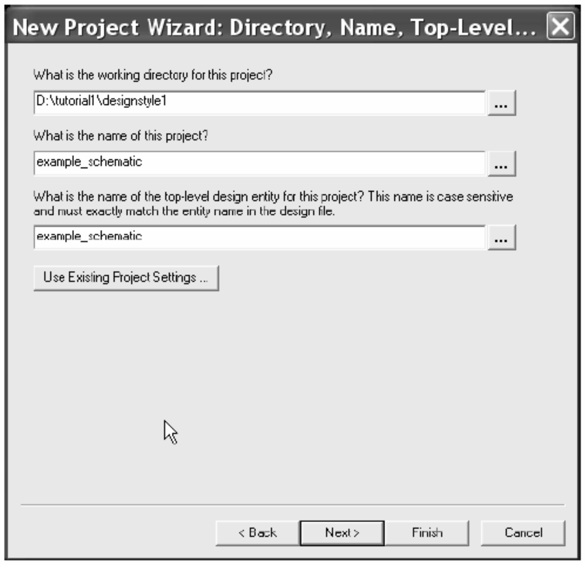


Figure 2: An example of the File menu.

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 any application, pressing the F1 function key on the keyboard opens a Help display that shows the commands available for that application.

2. Starting a New Project

To start working on a new design we first have to define a new *design project*. Quartus II makes the designer's task easier by providing support in the form of a *wizard*. Select File | New Project Wizard to reach a window that indicates the capability of this wizard. Press Next to get the window shown in Figure 3. Set the working directory to be */users/class/ee3109/ed.../tutorial/*, the directory should be set for you already. The project must have a name, which may optionally be the same as the name of the directory. We have chosen the name *example schematic* because our first example involves design entry by means of schematic capture. Usually the name of the schematic must be same as the name of the top-level entity. Press Next. Since we have not yet created the directory, Quartus II displays the pop-up box in Figure 4 asking if it should create the desired directory. Click Yes, which leads to the window in Figure 5. In this window the designer can specify which existing files (if any) should be included in the project. We have no existing files, so click Finish.


New Project Wizard: Directory, Name, Top-Level...

What is the working directory for this project?

D:\tutorial1\designstyle1

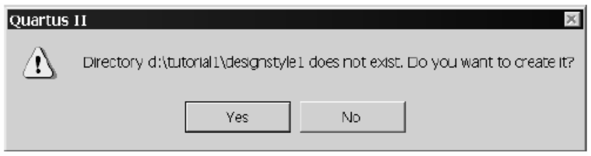
What is the name of this project?

example_schematic

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.

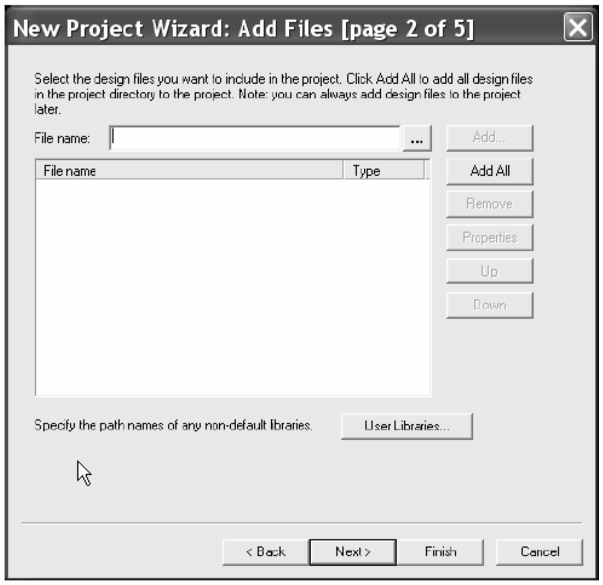
example_schematic

Figure 3: Specifying the project directory and name


Quartus II

Directory d:\tutorial1\designstyle1 does not exist. Do you want to create it?

Figure 4: Quartus II can create the desired directory


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

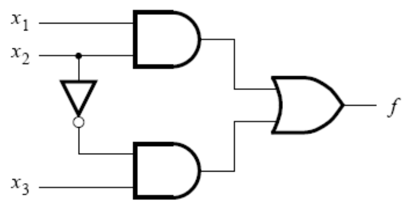
Specify the path names of any non-default libraries.

Figure 5: A window for inclusion of design files

If you press Next, another window appears, which allows the designer to specify the type of device in which the designed circuit will be implemented. For the purpose of this tutorial the choice of device is unimportant. The device will be chosen automatically. We do not need to choose a specific device, so click on the selection Auto device selected by the Fitter from the ’Available devices’ list.

3. Design Entry Using Schematic Capture

As explained before, commonly used design entry methods include schematic capture and VHDL code. This section illustrates the process of using the schematic capture tool provided in Quartus II, which is called the Block Editor. As a simple example, we will draw a schematic for the logic function $f = x_1x_2 + x_2x_3$. A circuit diagram for f was shown in Figure 6a. The truth table for f is given in Figure 6b. After creating the schematic, we show how to use the simulator in Quartus II to verify the correctness of the designed circuit.



(a) Circuit

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

(b) Truth table

Figure 6: The logic function that to be implemented.

Using the Block Editor

The first step is to draw the schematic. In the Quartus II display select File | New. A window that appears, shown in Figure 7, allows the designer to choose the type of file that should be created. The possible file types include schematics, VHDL code, and other hardware description language files such as Verilog and AHDL (Altera's proprietary HDL). It is also possible to use a third-party synthesis tool to generate a file that represents the circuit in a standard format called EDIF (Electronic Design Interface Format). The EDIF standard provides a convenient mechanism for exchanging information between EDA tools. Since we want to illustrate the schematic-entry approach in this section, choose Block Diagram/Schematic File and click OK. This selection opens the Block Editor window shown on the right side of Figure 8. Drawing a circuit in this window will produce the desired block diagram file, example_schematic.bdf. Resize the window to its maximum size to make it easier to draw the circuit.

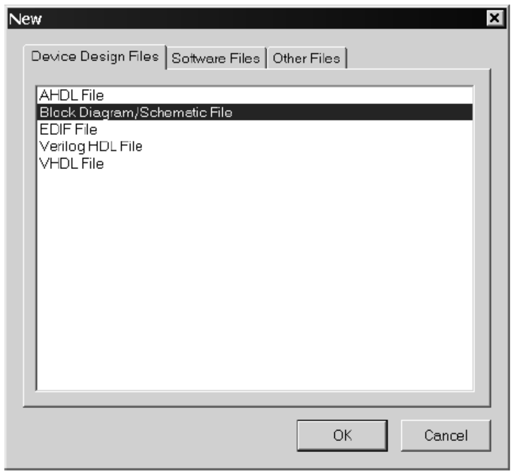


Figure 7: Choosing the type of design file.

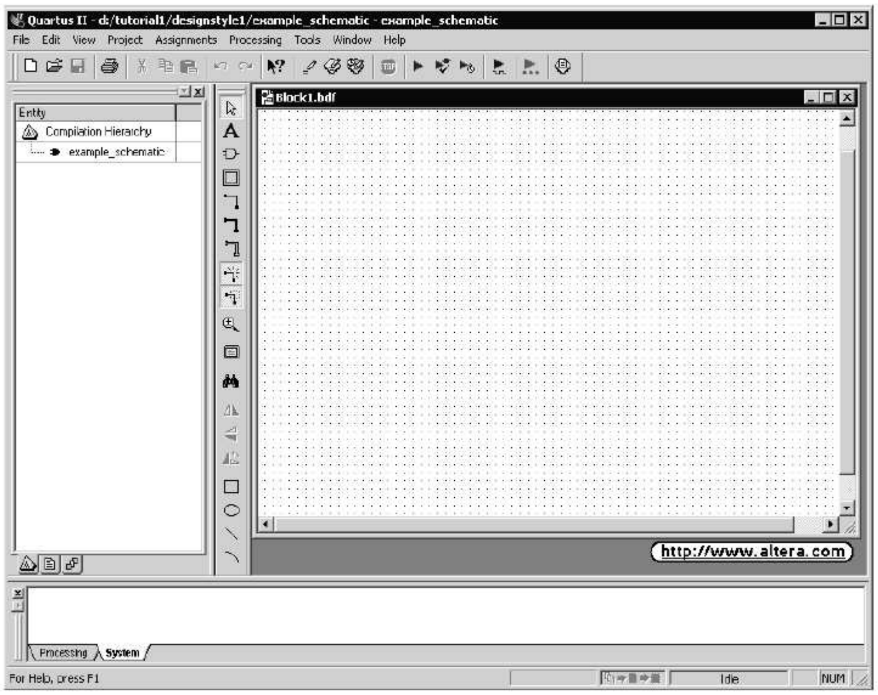

Quartus II - d:/tutorial1/designstyle1/example_schematic - example_schematic
File Edit View Project Assignments Processing Tools Window Help
Entity
Compilation Hierarchy
example_schematic
Block1.bdf
http://www.altera.com
Processing
System
For Help, press F1
Idle
NUM

Figure 8: Block Editor Window

Importing Logic Gate Symbols

The Block Editor provides several libraries that contain circuit elements which can be imported into a schematic. For our simple example we will use a library called *primitives*, which contains basic logic gates. To access the library, double-click on the blank space inside the Block Editor display to open the window in Figure 9 (another way to open this window is to select Edit | Insert Symbol or by clicking on the AND gate symbol in the toolbar). In the figure, the box labeled Libraries lists several libraries that are provided with Quartus II. To expand the list, click on the small + symbol next to c:/altera/libraries, then click on the + next to primitives, and finally click on the + next to logic. Now, double-click on the *and2* symbol to import it into the schematic (you can alternatively click on *and2* and then click OK). A two-input AND-gate symbol now appears in the Block Editor window. Using the mouse, move the symbol to the position where it should appear in the diagram and place it there by clicking the mouse.

Any symbol in a schematic can be selected by using the mouse. Position the mouse pointer on top of the AND-gate symbol in the schematic and click the mouse to select it. The symbol is highlighted in color. To move a symbol, select it and, while continuing to press the mouse button, drag the mouse to move the symbol. To make it easier to position the graphical symbols, a grid of guidelines can be displayed in the Block Editor window by selecting View | Show Guidelines.

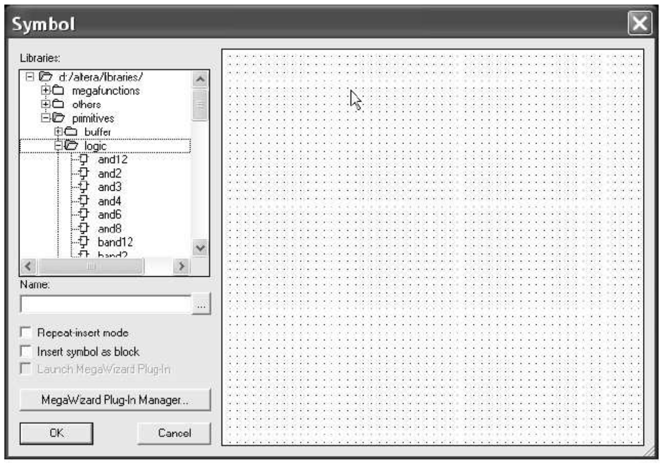


Figure 9: Selection of logic symbols

The logic function *f* requires a second two-input AND gate, a two-input OR gate, and a NOT gate. Use the following steps to import them into the schematic.

Position the mouse pointer over the AND-gate symbol that has already been imported. Press and hold down the Ctrl keyboard key and click and drag the mouse on the AND-gate symbol. The Block Editor automatically imports a second instance of the AND-gate symbol. This shortcut procedure for making a copy of a circuit element is convenient when you need many instances of the same element in a schematic. Of course, an alternative approach is to import each instance of the symbol by opening the primitive's library as described above.

To import the OR-gate symbol, again double-click on a blank space in the Block Editor to get to the primitives library. Use the scroll bar to scroll down through the list of gates to find the symbol named *or2*. Import this symbol into the schematic. Next import the NOT gate using the same procedure. To orient the NOT gate so that it points downward, as depicted in Figure B.8a, select the NOT-gate symbol and then use the command Edit | Rotate by Degrees | 270 to rotate the symbol 270 degrees counterclockwise. The symbols in the schematic can be moved by selecting them and dragging the mouse, as explained above. More than one symbol can be selected at the same time by clicking the mouse and dragging an outline around the symbols. The selected symbols are moved together by clicking on any one of them and moving it.

Experiment with this procedure. Arrange the symbols so that the schematic appears similar to the one in Figure 10.

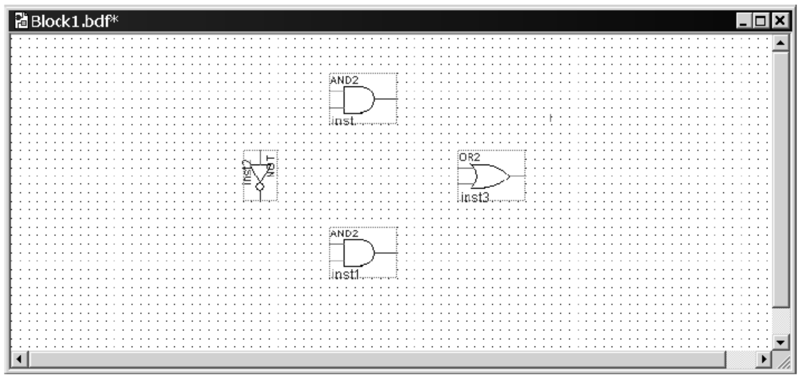


Figure 10: Imported gate symbols

Importing Input and Output Symbols

Now that the logic-gate symbols have been entered, it is necessary to import symbols to represent the input and output ports of the circuit. Open the primitive's library again. Scroll down past the gates until you reach *pins*. Import the symbol named *input* into the schematic. Import two additional instances of the input symbol, having a total of 3 input pins. To represent the output of the circuit, open the primitive's library and import the symbol named *output*. Arrange the symbols to appear as illustrated in Figure 11.

Assigning Names to Input and Output Symbols

Point to the word pin name on the input pin symbol in the upper-left corner of the schematic and double-click the mouse. The pin name is selected, allowing a new pin name to be typed. Type x1 as the pin name. Hitting carriage return immediately after typing the pin name causes the mouse focus to move to the pin directly below, the one currently being named. This method can be used to name any number of pins. Assign the names x2 and x3 to the middle and bottom input pins, respectively. Finally, assign the name *f* to the output pin. The pin names should be displayed as shown in Figure B.15.

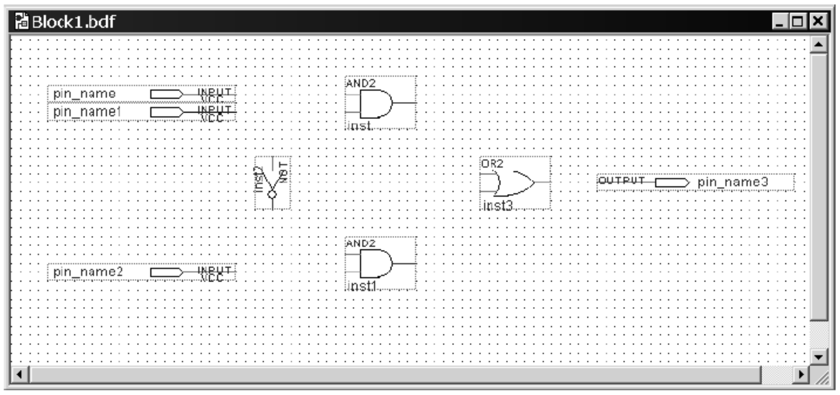


Figure 11: The desired arrangement of gates and pins

Connecting Nodes with Wires

The next step is to draw lines (wires) to connect the symbols in the schematic together. Click on the icon that looks like a big arrowhead in the vertical toolbar. This icon is called the Selection and Smart Drawing tool, and it allows the Block Editor to change automatically between the modes of selecting a symbol on the screen or drawing wires to interconnect symbols. The appropriate mode is chosen depending on where the mouse is pointing.

Move the mouse pointer on top of the x1 input symbol. When pointing anywhere on the symbol except at the right edge, the mouse pointer appears as crossed arrowheads. This indicates that the symbol will be selected if the mouse button is pressed. Move the mouse to point to the small line, called a *pin stub*, on the right edge of the x1 input symbol. The mouse pointer changes to a crosshair, which allows a wire to be drawn to connect the pinstub to another location in the schematic. A connection between two or more pinstubs in a schematic is called a *node*. The name derives from electrical terminology, where the term *node* refers to any number of points in a circuit that are connected together by wires.

Connect the input symbol for x1 to the AND gate at the top of the schematic as follows. While the mouse is pointing at the pinstub on the x1 symbol, click and hold the mouse button. Drag the mouse to the right until the line (wire) that is drawn reaches the pinstub on the top input of the AND gate; then release the button. The two pinstubs are now connected and represent a single node in the circuit. Use the same procedure to draw a wire from the pinstub on the x2 input symbol to the other input on the AND gate. Then draw a wire from the pinstub on the input of the NOT gate upward until it reaches the wire connecting x2 to the AND gate. Release the mouse button and observe that a connecting dot is drawn automatically. The three pinstubs corresponding to the x2 input symbol, the AND-gate input, and the NOT-gate input now represent a single node in the circuit. Figure 12 shows a magnified view of the part of the schematic that contains the connections drawn so far. To increase or decrease the portion of the schematic displayed on the screen, use the icon that looks like a magnifying glass in the toolbar.

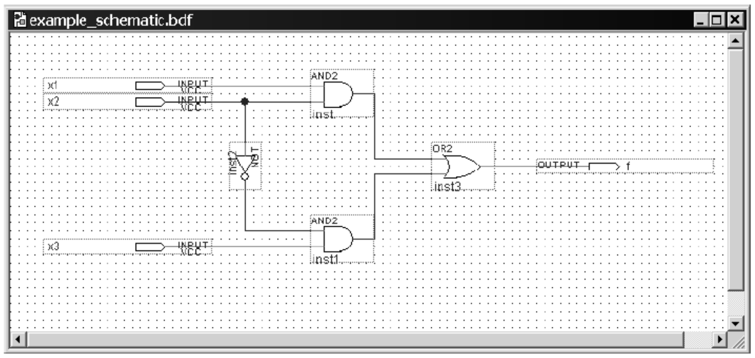


Figure 12: The completed schematic

To complete the schematic, connect the output of the NOT gate to the lower AND gate and connect the input symbol for x3 to that AND gate as well. Connect the outputs of the two AND gates to the OR gate and connect the OR gate to the *f* output symbol. If any mistakes are made while connecting the symbols, erroneous wires can be selected with the mouse and then removed by pressing the Delete key or by selecting Edit | Delete. The finished schematic is depicted in Figure 12. Save the schematic using File | Save As and choose the name *example schematic*. Note that the saved file is called *example_ schematic.bdf*.

Try to rearrange the layout of the circuit by selecting one of the gates and moving it. Observe that as you move the gate symbol all connecting wires are adjusted automatically. This takes place because Quartus II has a feature called *rubberbanding* which was activated by default when you chose to use the Selection and Smart Drawing tool. There is a rubberbanding icon, which is the icon in the toolbar that looks like an L-shaped wire with small tick marks on the corner. Observe that this icon is highlighted to indicate the use of rubberbanding. Turn the icon off and move one of the gates to see the effect of this feature.

Since our example schematic is quite simple, it is easy to draw all the wires in the circuit without producing a messy diagram. However, in larger schematics some nodes that have to be connected may be far apart, in which case it is awkward to draw wires between them. In such cases the nodes are connected by assigning labels to them, instead of drawing wires. See Help for a more detailed description.

4. Synthesizing a Circuit from the Schematic

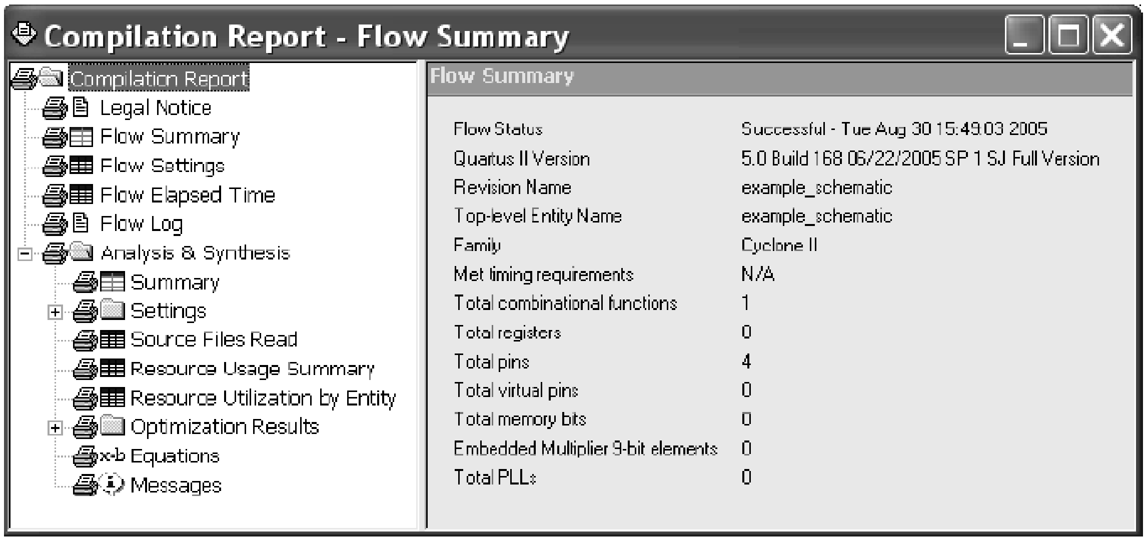
After a schematic is entered into a CAD system, it is processed by a number of CAD tools. The first step in the CAD flow uses the synthesis tool to translate the schematic into logic expressions. Then, the next step in the synthesis process, called technology mapping, determines how each logic expression should be implemented in the logic elements available in the target chip.

Using the Compiler

Run the Compiler by selecting Processing | Start Compilation, or by using the toolbar icon that looks like a solid purple triangle. As the compilation moves through various stages, its progress is reported in the window in the bottom left corner. Successful (or unsuccessful) compilation is indicated in a pop-up box. Acknowledge it by clicking OK and examine the compilation report depicted in Figure 13. The report summary indicates that only a miniscule amount of chip resources are needed to implement this tiny circuit on the selected FPGA chip.

The compilation report provides a lot of information that may be of interest to the designer. For example, the detailed implementation in the form of synthesized logic expressions can be seen by selecting Fitter | Fitter Equations in the compilation report directory. Note that the equation that Quartus II used to implement our circuit is

$$f = x_1(x_3 + x_2) + x_1x_3x_2$$



Flow Status	Successful - Tue Aug 30 15:49:03 2005
Quartus II Version	5.0 Build 168 06/22/2005 SP 1 SJ Full Version
Revision Name	example_schematic
Top-level Entity Name	example_schematic
Family	Cyclone II
Met timing requirements	N/A
Total combinational functions	1
Total registers	0
Total pins	4
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0

Figure 13: The compilation report summary

(Quartus II denotes AND as &, OR as #, and NOT as!). This is not the simplest expression that one may expect, namely

$$f = x_1x_2 + x_2x_3$$

But both expressions represent the same function and since the logic elements (LEs) are four-input lookup (truth) tables, called LUTs, in the chosen FPGA chip our function is implemented using a single LE in either case. The compilation report can be opened at any time by selecting Processing | Compilation Report or by clicking on the corresponding toolbar icon which looks like a white sheet on top of a blue chip.

Errors

Quartus II displays messages produced during compilation in the Messages window. This window is at the bottom of the Quartus II display in Figure 1. If the schematic is drawn correctly, one of the messages will state that the compilation was successful and that there are no errors or warnings.

To see what happens if an error is made, remove the wire that connects input x3 to the bottom AND gate and compile the modified schematic. Now, the compilation is not successful and two error messages are displayed. The first tells the designer that the affected AND gate is missing a source. The second states that there is one error and one warning. In a large circuit it may be difficult to find the location of an error. Quartus II provides help whereby if the user double-clicks on the error message, the corresponding location (AND gate in our case) will be highlighted. Reconnect the removed wire and recompile the corrected circuit.

5. Simulating the Designed Circuit

Quartus II includes a simulation tool that can be used to simulate the behavior of the designed circuit. Before the circuit can be simulated, it is necessary to create the desired waveforms, called *test vectors*, to represent the input signals. We will use the Quartus II Waveform Editor to draw test vectors.

Using the Waveform Editor

Open the Waveform Editor window by selecting File|New, which gives the window in Figure 7. Click on the Other Files tab to reach the window displayed in Figure 14. Choose Vector Waveform File and click OK.

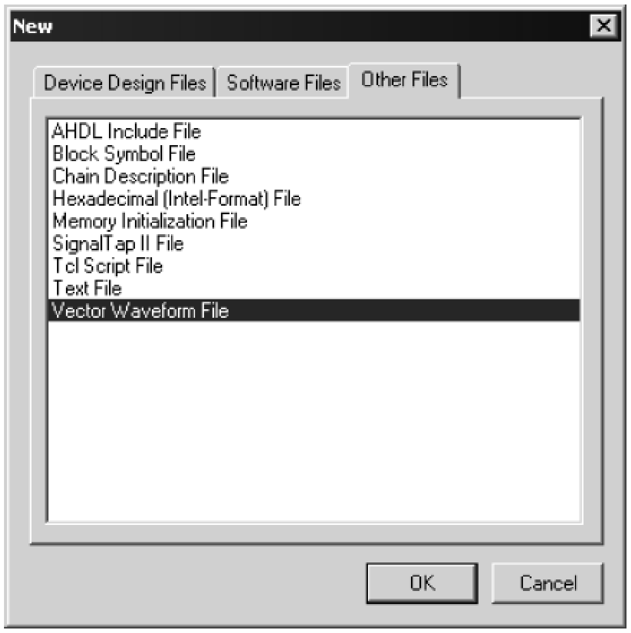


Figure 14: Choose to prepare a test-vector file.

The Waveform Editor window is depicted in Figure 15. Save the file under the name *example_ schematic.vwf*, and note that this changes the name in the displayed window. Set the desired simulation to run from 0 to 160 ns by selecting Edit | End Time and entering 160 ns in the dialog box that pops up. Select View | Fit in Window to display the entire simulation range of 0 to 160 ns in the window. You may want to resize the window to its maximum size.

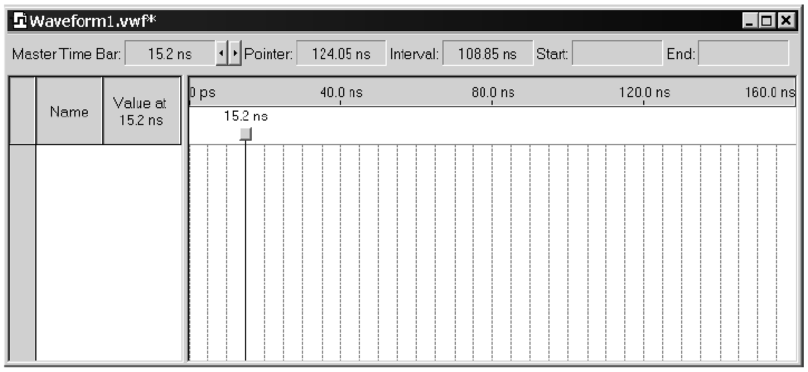
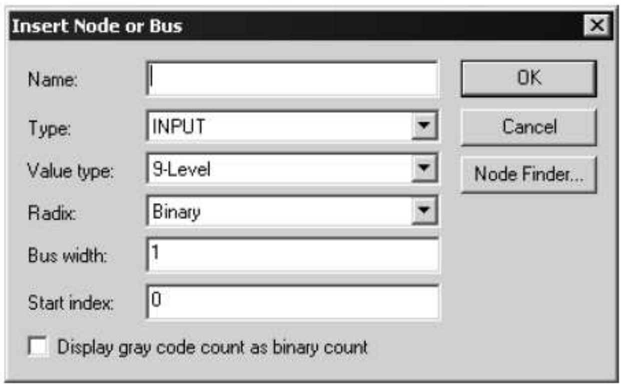


Figure 15: The waveform editor window.

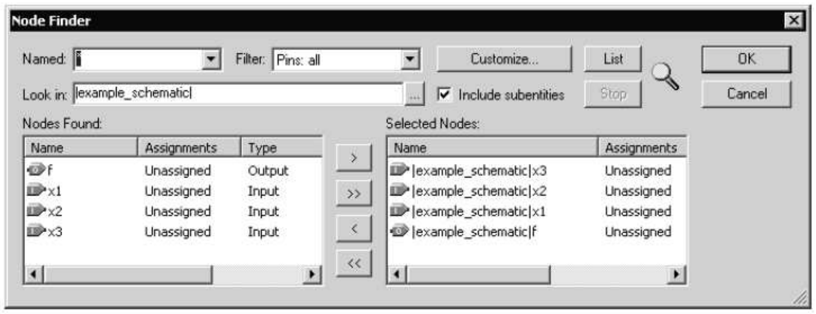
Next, we want to include the input and output nodes of the circuit to be simulated. This is done by using the Node Finder utility. Click Edit | Insert Node or Bus to open the window in Figure 16. It is possible to type the name of a signal (pin) into the Name box, but it is more convenient to click on the button labeled Node Finder to open the window in Figure 17. The Node Finder utility has a filter used to indicate what type of nodes are to be found. Since we are interested in input and output pins, set the filter to Pins: all. Click the List button to find the input and output nodes.


Insert Node or Bus

Name:	
Type:	INPUT
Value type:	9-Level
Radix:	Binary
Bus width:	1
Start index:	0

☐ Display gray code count as binary count

Figure 16: The Insert Node or Bus dialogue.


Node Finder

Named: | Filter: Pins: all

Look in: |example_schematic| ☑ Include subentities

Nodes Found:

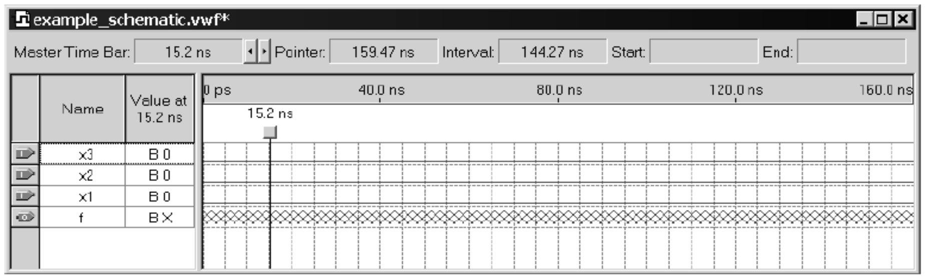
Name	Assignments	Type
f	Unassigned	Output
x1	Unassigned	Input
x2	Unassigned	Input
x3	Unassigned	Input

Selected Nodes:

Name	Assignments
|example_schematic|x3	Unassigned
|example_schematic|x2	Unassigned
|example_schematic|x1	Unassigned
|example_schematic|f	Unassigned

Figure 17: The Node Finder window.

The Node Finder displays on the left side of the window the nodes *f*, x1, x2, and x3. Click on x3 and then click the > sign to add it to the Selected Nodes box on the right side of the figure. Do the same for x2, x1, and *f*. Click OK to close the Node Finder window, and then click OK in the window of Figure 17. This leaves a fully displayed Waveform Editor window, as shown in Figure 18. If you did not select the nodes in the same order as displayed in Figure 18, it is possible to rearrange them.


example_schematic.vwf*

Master Time Bar: 15.2 ns Pointer: 159.47 ns Interval: 144.27 ns Start: End:

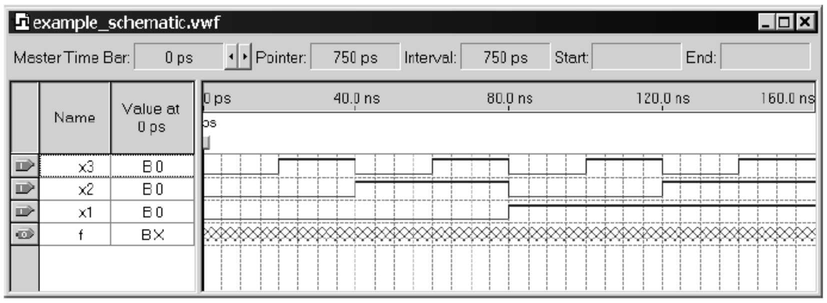
Name	Value at 15.2 ns
x3	B 0
x2	B 0
x1	B 0
f	B X

Figure 18: The nodes needed for simulation.

To move a waveform up or down in the Waveform Editor window, click on the node name (in the Name column) and release the mouse button. The waveform is now highlighted to show the selection. Click again on the waveform and drag it up or down in the Waveform Editor.

We will now specify the logic values to be used for the input signals during simulation. The logic values at the output *f* will be generated automatically by the simulator. To make it easy to draw the desired waveforms, Quartus II displays (by default) the vertical guidelines and provides a drawing feature that snaps on these lines (which can otherwise be invoked by choosing View | Snap to Grid). Observe also a solid vertical line, which can be moved by pointing to its top and dragging it horizontally. We will use this "reference line" in Tutorial 2. The waveforms can be drawn using the Selection tool, which is activated by selecting the icon in the vertical toolbar that looks like a big arrowhead.

To simulate the behavior of a large circuit, it is necessary to apply a sufficient number of input valuations and observe the expected values of the outputs. The number of possible input valuations may be huge, so it is necessary to choose a relatively small (but representative) sample of these input valuations. Our circuit is very small, so it can be simulated fully by applying all eight possible valuations of inputs x1, x2, and x3. Let us apply a new valuation every 20 ns. To start, all inputs are zero. At the 20-ns point we want x3 to go to 1. Click on x3; this highlights the signal and activates the vertical toolbar that allows us to shape the selected waveform. The toolbar provides options such as setting the signal to 0, 1, unknown (X), high impedance (Z), don't care (DC), and inverting its existing value (INV). Observe that the output *f* is displayed as having an unknown value at this time, which is indicated by a hashed pattern. A specific time interval is selected by pressing the mouse on a waveform at the start of the interval and dragging it to its end; the selected interval is highlighted. Select the interval from 20 to 40 ns for x3 and set the signal to 1. Similarly, set x3 to 1 from 60 to 80 ns, 100 to 120 ns, and 140 to 160 ns. Next, set x2 to 1 from 40 to 80 ns, and from 120 to 160 ns. Finally, set x1 to 1 from 80 to 160 ns. Complete the remaining assignments to obtain the image in Figure 19 and save the file.


example_schematic.vwf

Master Time Bar: 0 ps Pointer: 750 ps Interval: 750 ps Start: End:

Name	Value at 0 ps
x3	B 0
x2	B 0
x1	B 0
f	B X

Figure 19: The complete test vectors.

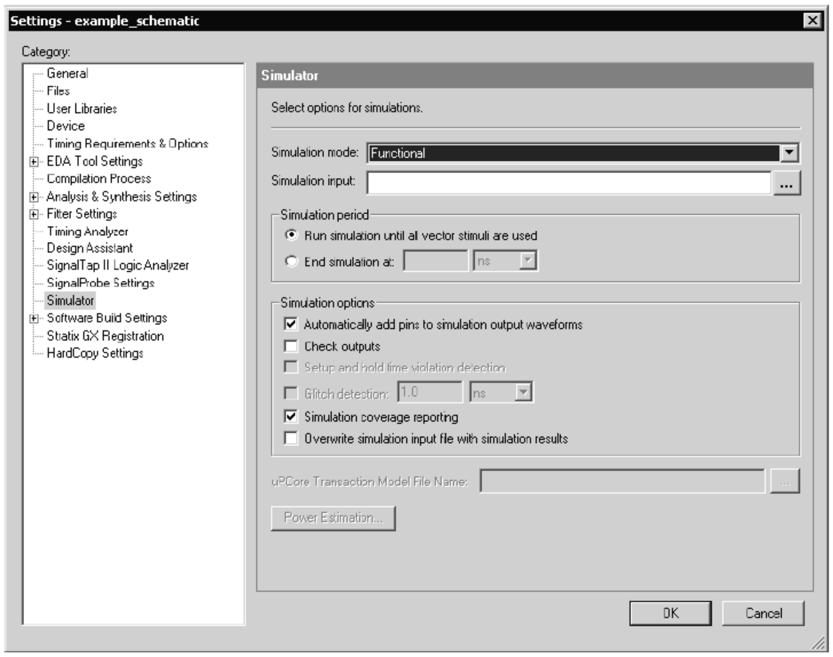
A convenient mechanism for changing the input waveforms is provided by the Waveform Editing tool. The icon for the tool is in the vertical toolbar; it looks like two arrows pointing left and right. When the mouse is dragged over some time interval in which the waveform is 0 (1), the waveform will be changed to 1 (0). Experiment with this feature on signal x3.

Using the Clock

Another convenient way to assign the values to the vectors is by using the clock. As mentioned before the width of each combination is 20ns. Right click the signal x3 and select Value | Clock. Change the time period to 40ns. Repeat the same for x2 with period of 80ns and for x3 with period of 160ns.

Performing the Simulation

A circuit can be simulated in two ways. The simplest way is to assume that logic elements and interconnection wires are perfect, thus causing no delay in propagation of signals through the circuit. This is called *functional simulation*. A more complex alternative is to take all propagation delays into account, which leads to *timing simulation*. Typically, functional simulation is used to verify the functional correctness of a circuit as it is being designed. This takes much less time, because the simulation can be performed simply by using the logic expressions that define the circuit. In this tutorial we will use only the functional simulation.



Settings - example_schematic

Category:
- General
- Files
- User Libraries
- Device
- Timing Requirements & Options
- EDA Tool Settings
- Compilation Process
- Analysis & Synthesis Settings
- Fitter Settings
- Timing Analyzer
- Design Assistant
- SignalTap II Logic Analyzer
- SignalProbe Settings
- Simulator
- Software Build Settings
- Stratix GX Registration
- HardCopy Settings

Simulator

Select options for simulations.

Simulation mode: Functional

Simulation input:

Simulation period
- ● Run simulation until all vector stimuli are used
- ○ End simulation at: ns

Simulation options
- ✓ Automatically add pins to simulation output waveforms
- ☐ Check outputs
- ☐ Setup and hold time violation detection
- ☐ Glitch detection: 1.0 ns
- ✓ Simulation coverage reporting
- ☐ Overwrite simulation input file with simulation results

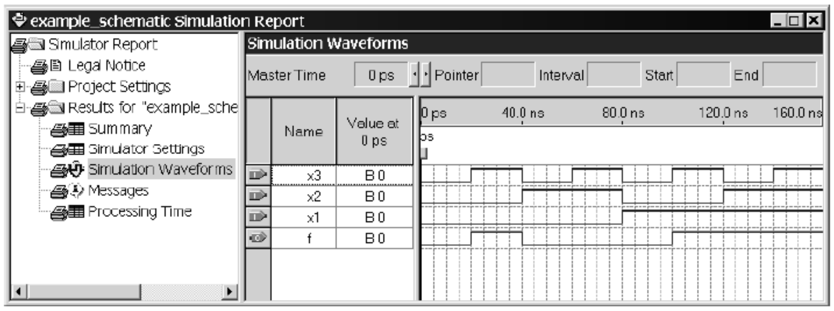
uPCore Transaction Model File Name:

Power Estimation...

OK Cancel

Figure 20: Specifying the simulation mode.

To perform the functional simulation, select Assignments | Settings to open the Settings window. On the left side of this window click on Simulator to display the window in Figure B.23 and choose Functional as the simulation mode. Also select *Overwrite simulation input file with simulation results* by clicking on the box next to it. This will overwrite the input file with results. To complete the set up of the simulator select the command Processing | Generate Functional Simulation Netlist. The Quartus II simulator takes the test inputs and generates the outputs defined in the *example_schematic.vwf* file. Run the simulation by selecting Processing | Start Simulation, or by using the shortcut icon in the toolbar that looks like a blue triangle with a square wave below it. At the end of the simulation, Quartus II indicates its successful completion and displays a simulation report shown in Figure 21. As seen in the figure, the Simulator creates a waveform for the output *f*. The reader should verify that the generated waveform corresponds to the truth table for *f* given in Figure 6b.



Name	Value at 0 ps
x3	B 0
x2	B 0
x1	B 0
f	B 0

Figure 21: The result of functional simulation.