

1.1 Lab 1: Schematic Entry Using the Quartus II Development Software

Purpose

In this lab you will learn how to use the Altera Quartus II development software to design a simple combinational circuit.

Introduction

The Altera Quartus II development software and the DE1 development board provide all of the necessary tools for implementing and trying out all of the circuits, including building the final general-purpose microprocessor. The Quartus II software offers a completely integrated development tool and easy-to-use graphical-user interface for the design, and synthesis of digital logic circuits. Together with the DE1 development board, these circuits can actually be implemented in hardware. The main component on the DE1 development board is a field programmable gate array (FPGA) chip which is capable of implementing very complex digital logic circuits. After synthesizing a circuit and downloading it onto the FPGA, you can see the operation of the circuit in hardware.

The Web Edition version of the Quartus II software can be downloaded for free from the Altera website at www.altera.com. This lab assumes that you are familiar with the Windows environment, and that the Quartus II software has already been installed on your computer. The rest of this lab will provide a step-by-step instruction for the schematic entry of a 2-input AND gate circuit.

1.1.1 Starting Quartus II

After the successful installation of the Quartus II software, there should be a link to the program under the Windows' **Start** button named **Quartus II 12.1 Web Edition**. Click on this link to start the program. You should see the main Quartus II window similar to Figure 1.

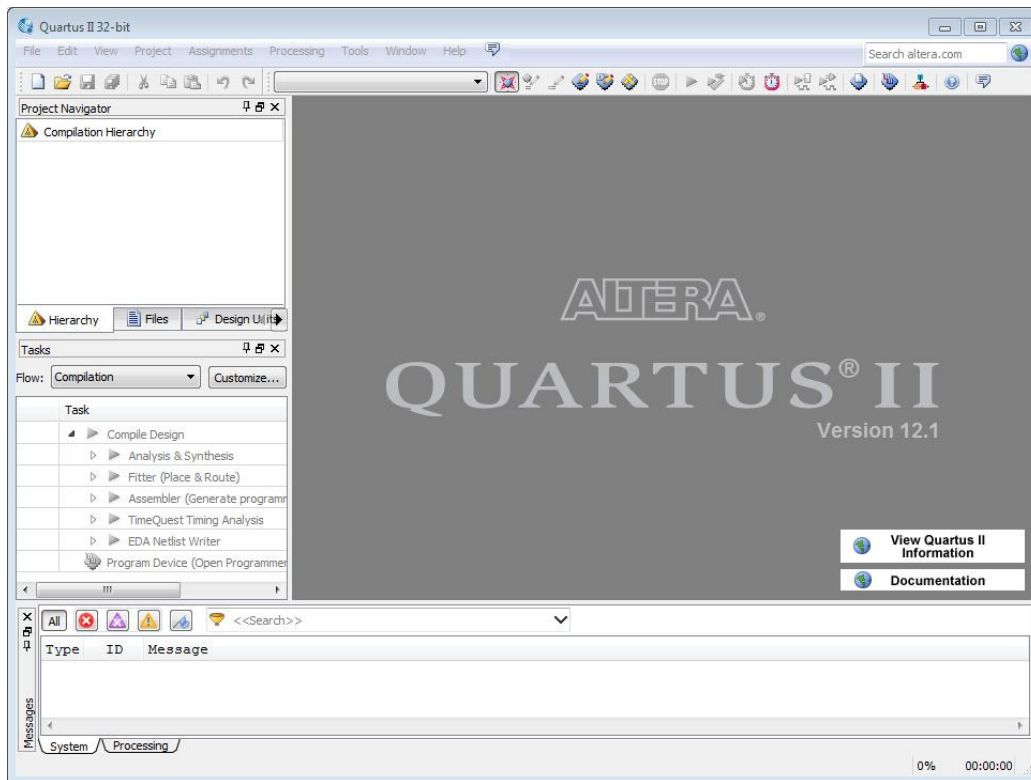


Figure 1: The Quartus II main window.

1.1.2 Creating a New Project

Each circuit design in Quartus II is called a project. Each project should be placed in its own folder, since the program creates many associated working files for a project. Perform the following steps to create a new project and a new folder for storing the project files.

From the Quartus II menu, select **File > New Project Wizard**. If the **New Project Wizard Introduction** screen appears and you don't want to see it again the next time you start the new project wizard, you can select the check box that says **Don't show me this introduction again**, and then click **Next** to go to the next screen. You should see the **New Project Wizard: Directory, Name, Top-Level Entity [page 1 of 5]** window as shown in Figure 2.

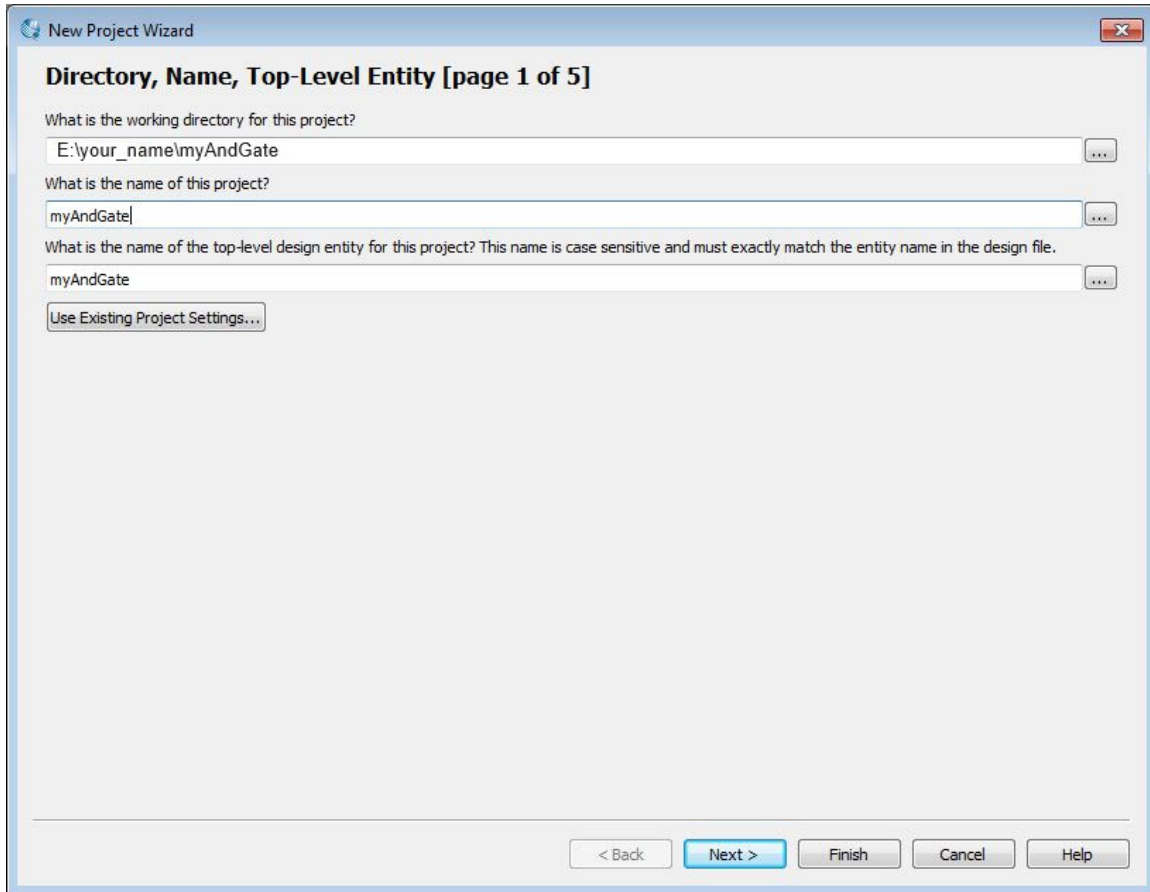



Figure 2: The New Project Wizard: Directory, Name, Top-Level Entity window with the working directory, the project name, and the top-level entity name filled in.

Type in the directory for storing your project. You can also click on the  icon next to it to browse to the directory. This is VERY IMPORTANT.

- For this lab, type in `E:\your_name\myAndGate` to create a folder named `myAndGate` inside a folder with your name on the E drive.

You also need to give the project a name.

- For this lab, type in the project name `myAndGate`.

A project may have more than one design file. Whether your project has one or more files, you need to specify which design file is the top-level design entity. The default name given is the same as the project name. However, you can use a different name.

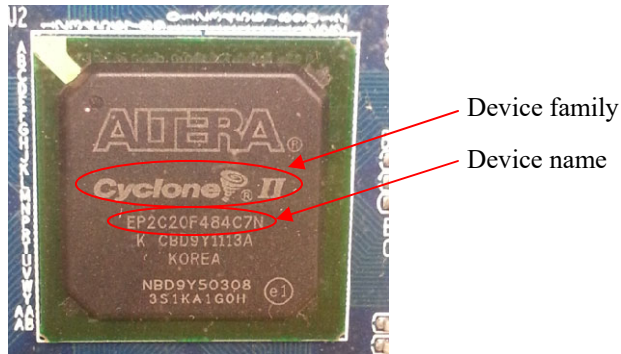
- For this lab, leave the top-level file name as `myAndGate`, and click **Next** to continue to the next window.

Since the directory `c:\myAndGate` does not yet exist, Quartus II will inform you of that and asks whether you want to create this new directory. Click **Yes** to create the directory.

In the **New Project Wizard: Add Files [page 2 of 5]** window, you can optionally add existing circuit source files associated with your project. For example, if you have a source file created in another project and want to use it in this project, you can specify that here.

- We are starting a new project and do not yet have any source files, so click **Next** to continue to the next window.

In the **New Project Wizard: Family & Device Settings [page 3 of 5]** window as shown in Figure 3, we select the target FPGA device that we will be using to implement the circuit on. You need to find the device family and name of the FPGA chip that is used on your development board. The FPGA chip will look something similar to the following. The device family for the FPGA chip in the picture is Cyclone II and the name is EP2C20F484C7N.



The DE1 development board uses the Cyclone II **EP2C20F484C7N** FPGA chip.

The DE2 development board uses the Cyclone II **EP2C35F672C6** FPGA chip.

If you are using a different FPGA chip then you need to make the appropriate changes in the next two lines of instructions.

- In the **Device Family** drop-down box, select **Cyclone II** or the one that matches your FPGA chip.
- In the **Available devices** list, select the device EP2C20F484C7, EP2C35F672C6 or the one that matches your FPGA chip. If this device is not listed, then you need to reinstall the Quartus II program with the Cyclone II device family option checked.
- Click **Next** to continue to the next window.

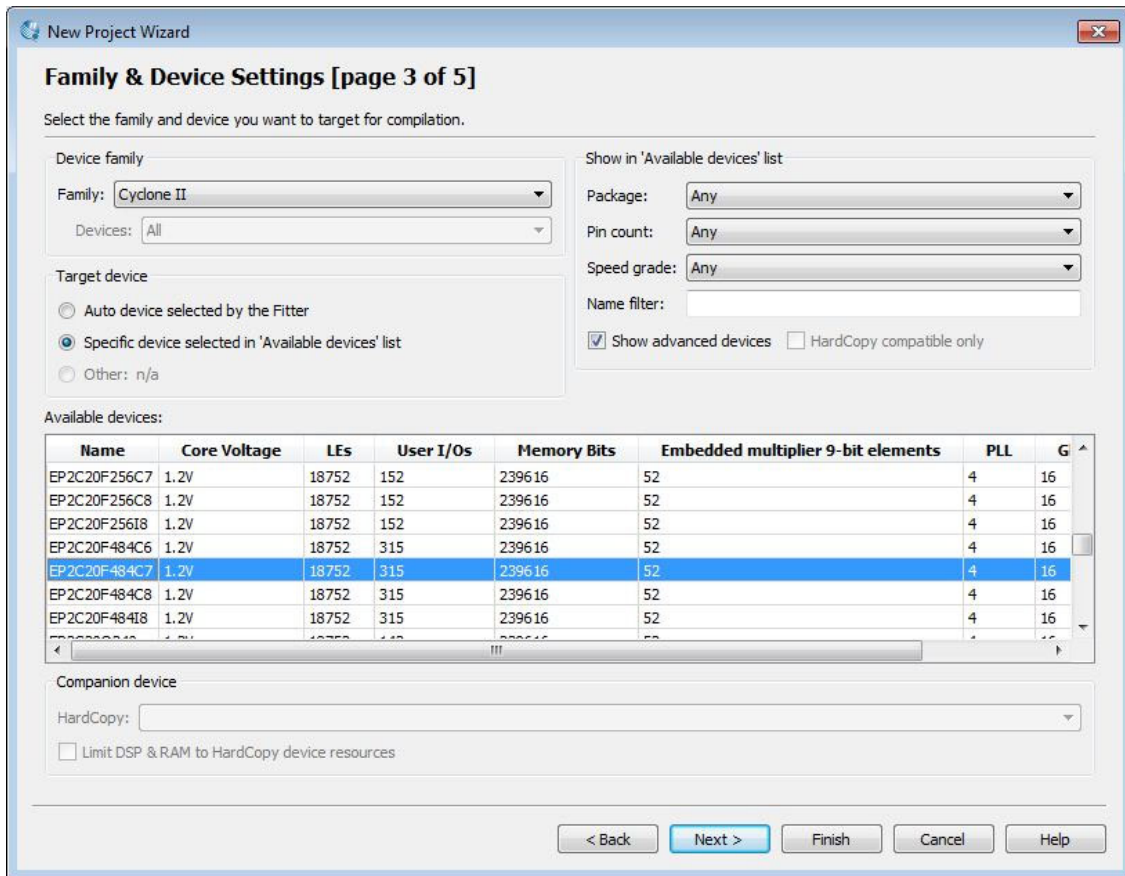


Figure 3: The New Project Wizard: Family & Device Settings window with the device EP2C20F484C7 selected.

In the next **New Project Wizard: EDA Tool Settings [page 4 of 5]** window, we do not have any EDA tools to use for this project, so click **Next** to continue to the next window.

The final window is a summary of the choices that you have just made. Click **Finish** to create your new project.

1.1.3 Using the Graphic Editor

After creating a new project, we are now ready to start the Schematic Block Editor for manually drawing the schematic circuit.

Starting the Graphic Editor

From the Quartus II menu, select **File > New**. Under **Design Files**, select **Block Diagram/Schematic File**, and then click **OK**. You should see the **Graphic Editor** window similar to the one shown in Figure 4. Any circuit diagram can be drawn in this Graphic Editor window.

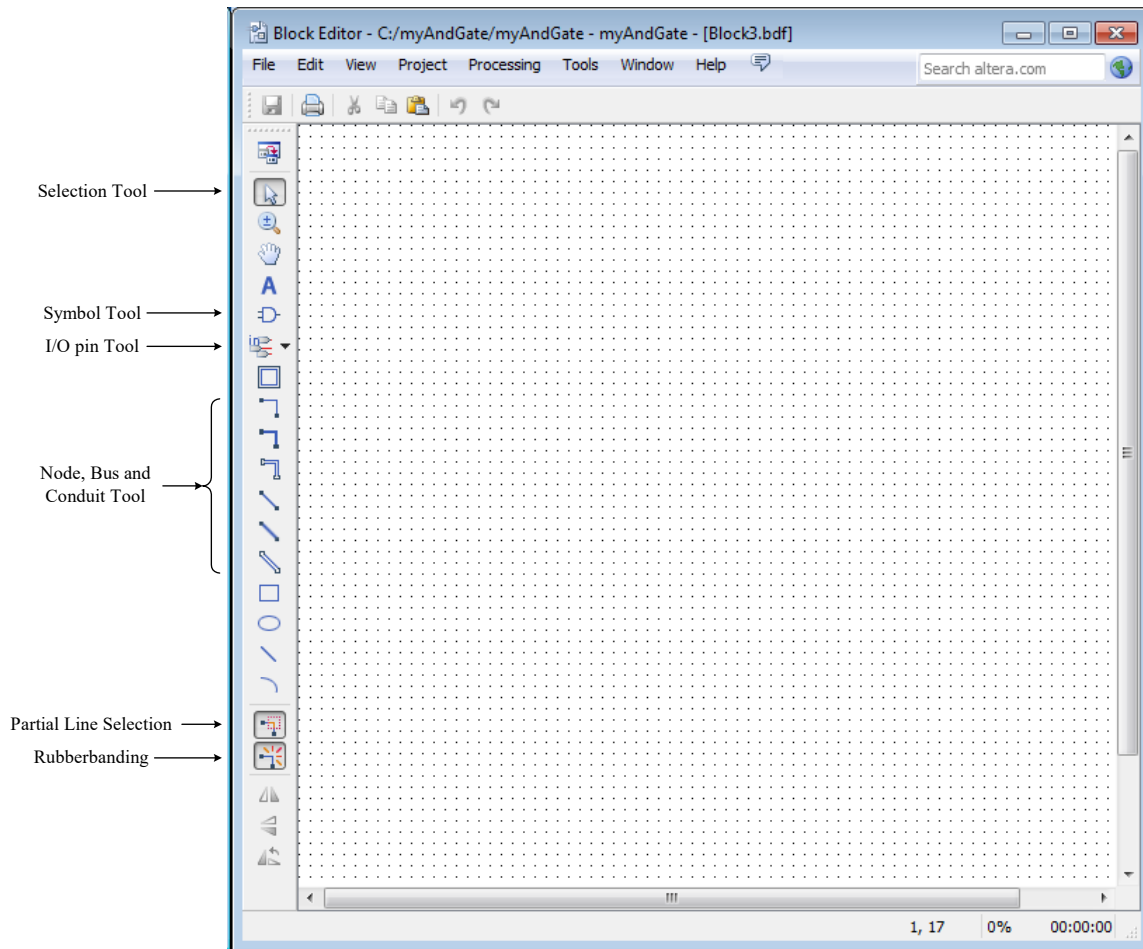


Figure 4: The Graphic Editor window with the graphics toolbar showing on the left.

Drawing Tools

In Figure 4, the tools for drawing circuits in the Block Editor are shown in the toolbar on the left side. The default location for this tool bar is at the top. There are the standard tools such as text writing, zoom, flip and rotate, and line and shape drawing. The main tool that you will use is the Selection tool. This selection tool allows you to perform many different operations depending on the context in which it is used. Two main operations performed by this tool are selecting objects and making connections between logic symbols. The Symbol tool allows you to select and use logic symbols from the library or from your own design files. The six Node, Bus and Conduit tools allow you to draw connection lines that are not connected to another object. The Partial Line Selection and Rubberbanding buttons turn on or off these functions. When rubberbanding is turned on, connection lines are adjusted automatically when symbols are moved from one location to another. When rubberbanding is turned off, moving a symbol will not affect the lines connected to it.

Inserting Logic Symbols

To insert a logic symbol, first select the Selection tool, and then double-click on an empty spot in the Block Editor window. You should see the Symbol window as shown in Figure 5.

- Alternatively, you can click on the Symbol tool icon in the toolbar to bring up the Symbol window.

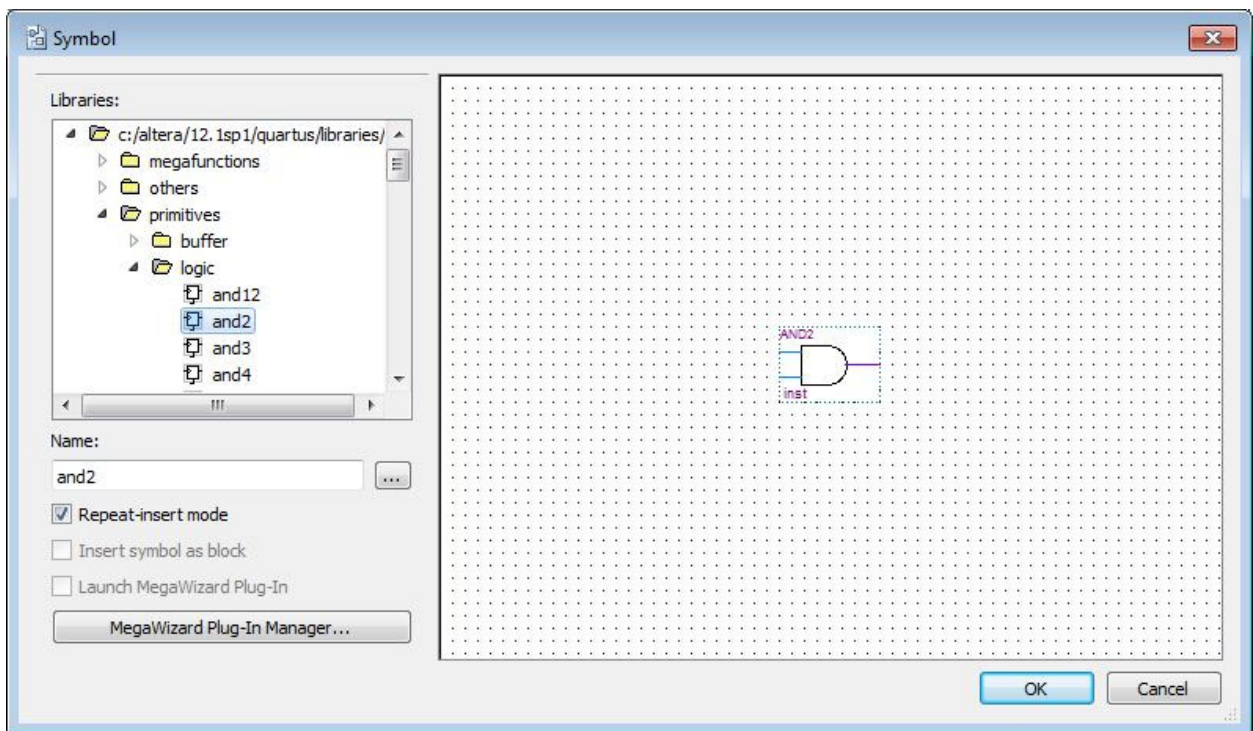


Figure 5: The Symbol selector window.

Available symbol libraries are listed in the Libraries box. These libraries include the standard primitive gates, standard combinational and sequential components, and your own logic symbols located in the current project directory.

All of the basic logic gates, latches, flip-flops, and input and output connectors that you need are located in the **primitives** folder. If this folder is not listed, then click on the plus (+) sign to expand the libraries folder. Within the **primitives** folder are several subfolders. The basic gates are in the **logic** subfolder; the latches and flip-flops are in the **storage** subfolder; and the input and output connectors are in

the **pin** subfolder. Your own circuits, if there are any that you want to reuse in building larger circuits, will be listed in the **Project** folder.

Expand the **logic** subfolder by clicking on the plus sign next to it to see a list of logic gate symbols available in that library. The logic symbols are sorted in alphabetical order. Select the logic symbol name that you want to use, or alternatively, you can just type in the name of the logic symbol in the **Name** field. Click on the **OK** button to insert the symbol in the Graphic Editor. If the Repeat-insert mode box is checked, then you can insert several instances of the same symbol until you press the **Esc** key.

For this lab, insert the following symbols into the Graphic Editor:

- A 2-input AND gate (**and2**) found in the **logic** subfolder.
- An input signal connector (**input**) found in the **pin** subfolder.
- An output signal connector (**output**) found in the **pin** subfolder.

A unique number is assigned to each instance of a symbol and is written at the lower-left corner of the symbol. This number is used only as a reference number in the output netlist and report files. The numbers that you see may be different from those in the examples.

Selecting, Moving, Copying, and Deleting Logic Symbols

To select a logic symbol in the Block Editor, simply click on the symbol using the Selection tool. You can also select multiple symbols by holding down the **Ctrl** key while you select the symbols. An alternative method is to trace a rectangle around the objects that you want to select. All objects inside the rectangle will be selected.

To de-select a symbol, simply click on an empty spot in the Block Editor.

To move a symbol, simply drag the symbol.

To copy a symbol, first select it and then perform the Copy and Paste operations. An alternative method is to hold down the **Ctrl** key while you drag the symbol.

To delete a symbol, first select it and then press the **Delete** key.

To rotate a symbol, right-click on the symbol, select **Rotate by Degrees** from the pop-up menu, and select the angle to rotate the symbol. Alternatively, you can first select the symbol and then click on one of the Flip or Rotate buttons on the tool bar.

Perform the following operations for this lab:

- Make a copy of the 2-input and gate
- Make two more copies of the input signal connector
- Position the symbols similar to Figure 6

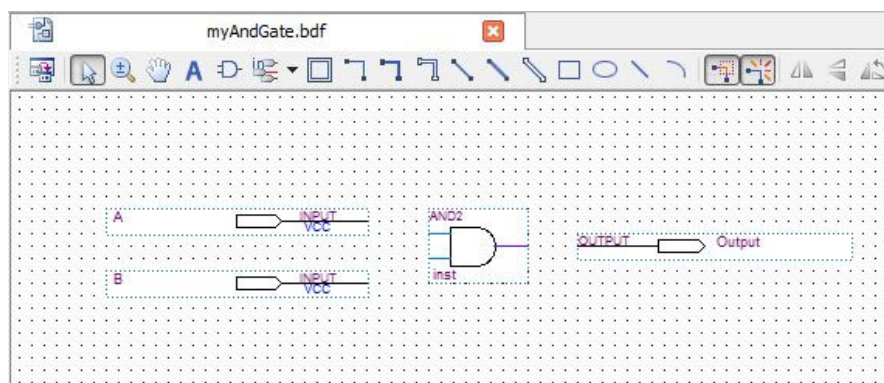


Figure 6: Symbol placements for the 2-to-1 multiplexer circuit.

Making and Naming Connections

To make a connection between two connection points, use the Selection tool and drag from one connection point to the second connection point. Notice that, when you position the pointer to a connection point, the arrow pointer changes to a crosshair.

To change the direction of a connection line while dragging the line, simply release and press the mouse button again, and then continue to drag the connection line.

You can also make a connection between two connection points by moving a symbol so that its connection point touches the connection point of the second symbol. With rubberbanding turned on, you can now move one symbol away from the second symbol, and a connection line is automatically drawn between them.

If you want to make a connection line that does not start from a symbol connection point, you will need to use either the Orthogonal Node tool or the Diagonal Node tool instead of the Selection tool.

Do not use the Line tool to make connections; this tool is only for drawing lines and not actually making a connection.

Once a connection is made to a symbol, you can move the symbol to another location, and the connection line is adjusted automatically if the rubberbanding function is turned on. However, if the rubberbanding function is turned off, the connection will be broken if the symbol is moved.

To make a connection between two lines that cross each other as shown in Figure 7, you need to use the Selection tool. Right-click on the junction point (i.e., the point where the two lines cross) and then select from the pop-up menu **Toggle Connection Dot**. You can repeat the same process to remove the connection point.



Figure 7: Making or deleting a connection point.

To select a line segment, simply single click on it. To select the entire line (with several line segments connected in different directions), you need to double-click on it.

Use the Orthogonal Bus tool to draw a bus connection.

To change a single node line to a bus line, right-click on the line and select **Bus Line** from the pop-up menu. Select **Node Line** from the pop-up menu to change it back to a node line.

A bus must also have a name and a width associated with it. To name a connection line, right-click on the line that you want to name. In the pop-up menu, select Properties and then type in the name and the width for the bus in the Name box. For example, `data[7..0]` is an 8-bit bus with the name `data`, as shown in Figure 8.

To change the name, just double-click on the name and edit it.

To connect one line to a bus, connect a single line to the bus, and then give it the same name as the bus with the line index appended to it. For example, `data[2]`, is bit two of the `data` bus, as shown in Figure 8.

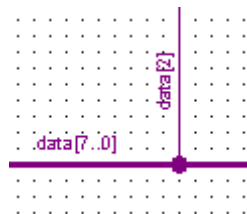


Figure 8: A single line connected to an 8-bit bus with the name `data`.

To check whether a name is attached correctly to a line, select the line, and the name that is attached to the line will also be selected.

All input and output signals in a circuit must be connected to an input and output signal connector respectively. To name an input or output signal connector, select its name label by single-clicking it, and then double-clicking it. You can now type in the new name. Pressing the **Enter** key will move the text entry cursor to the name label for the symbol below the current symbol. Alternatively, you can select the input or output connector and then double-click on. The Properties window for that pin will open up which allows you to enter the pin name, among other things.

A bus line connected to an input or output connector must have the same bus width as the connector.

For this lab, perform the following operations to look like Figure 9:

- Name the two input connectors A and B.
- Name the output connector Output.

Select **File > Save** to save the design file. Type in `myAndGate` for the filename. The default file extension is `.bdf` (for block design file). Recall that when we created the project, we had specified `myAndGate` as the top-level filename. We will now use this file as the top-level source file.

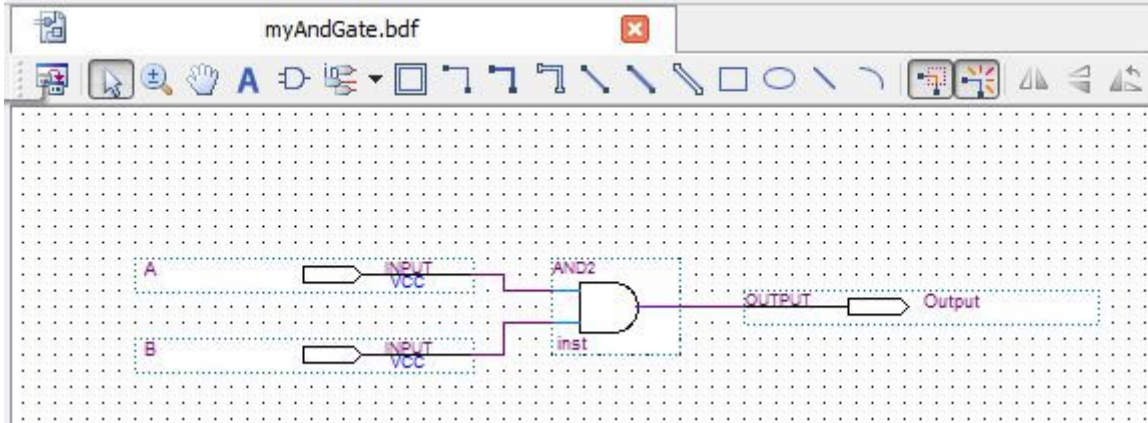


Figure 9: Connections and names for the `myAndGate` circuit.

Selecting, Moving and Deleting Connection Lines

To select a straight connection line segment, just single-click on it.

To select an entire connection line with horizontal and vertical segments, just double-click on it.

To select a portion of a line segment, first turn on the Use Partial Line Selection button, and then drag a rectangle around the line segment. Only the portion of the line segment that is inside the rectangle will be selected.

After a line is selected, it can be moved by dragging.

After a line is selected, it can be deleted by pressing the **Delete** key.

1.1.4 Managing Files in a Project

A project may have one or more design files associated with it.

Design Files in a Project

To see the files that are currently associated with a project, click on the **Files** tab in the **Project Navigator** window. The Project Navigator shown in Figure 10 shows that this project has only one file named `myAndGate.bdf`.

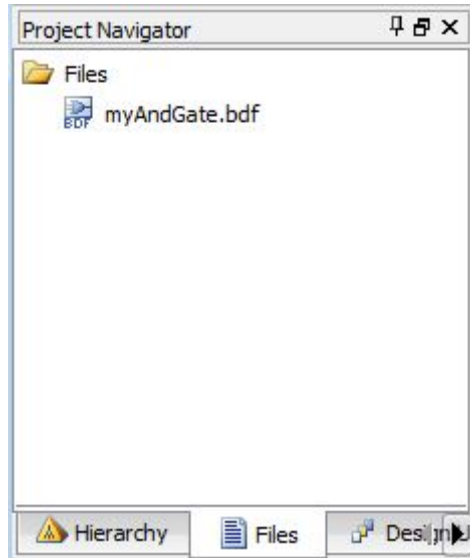


Figure 10: Files associated with a project as shown in the Project Navigator window.

Opening a Design File

To open a design file, simply double-click on the file that is listed in the Project Navigator window. Depending on the type of file, the associated editor will be used. The Block Editor is used to edit a Block Diagram/Schematic File, and a text editor is used to edit a VHDL or Verilog text file.

Creating a New Design File

To create a new schematic drawing design file, select **File > New** from the Quartus II menu. In the **Device Design Files** tab, select **Block Diagram/Schematic File**. After you save this file, the file is automatically added into the project.

Adding Design Files to a Project

To add an existing design file to the current project, select **Project > Add/Remove Files in Project** from the Quartus II menu. Alternatively, you can right-click on the folder icon labeled **Files** in the Project Navigator window, and then select **Add/Remove Files in Project** from the pop-up menu.

This will bring up the **Files Category** under the **Settings** window. From this window, you can choose additional files to be added into the project by either manually typing in the file name or browsing to the directory and then selecting it. Click on the **Add** button to add individual files, or click on the **Add All** button to add all of the files in the selected directory.

Deleting Design Files from a Project

To delete a design file from a project, simply select it in the Project Navigator window, and then press the **Delete** key. Alternatively, you can right-click on the file that you want to delete, and then select **Remove File from Project** from the pop-up menu.

Setting the Top-Level Entity Design File


- When you first created a new project, you also had to specify the name of the top-level design file. If you want to change the top-level entity to another design file, you can do so by right-clicking on the file that you want to be the top-level entity in the Project Navigator window. From the pop-up menu, select **Set as Top-Level Entity**.

Saving the Project

Select **File > Save Project** to save the project and all of its associated files.

1.1.5 Analysis and Synthesis

After drawing your circuit with the Graphic Editor, the next step is to analyze and synthesize it. During this step, Quartus II collects all of the necessary information about your circuit, and produces a netlist for it.

- From the Quartus II menu, select **Processing | Start | Start Analysis & Synthesis** to synthesize the circuit. Alternatively, you can click on the icon .
- If there are no errors in your circuit, you should see the message “Quartus II Analysis & Synthesis was successful” in the Message window at the bottom.

If there are errors then they will be reported in the Message window and highlighted in red. You can double-click on the error message to see where the error is in the circuit. Go back and double check your circuit with the one shown in **Figure 9**: Connections and names for the `myAndGate` circuit.

- to correct all of the errors.

1.1.6 Creating and Using a Logic Symbol

If you want to use a circuit as part of another circuit in a schematic drawing, you can create a logic symbol for this circuit. Logic symbols are like black boxes that hide the details of a circuit. Only the input and output signals for the circuit are shown. The input and output signals for the logic symbol are obtained directly from the input and output signal connectors that are connected in the circuit.

To create a logic symbol for a circuit, first select the Block Editor window containing the circuit that you want as the active window. Select **File > Create/Update > Create Symbol Files for Current File**. The name of this symbol file will be the same as the name of the current active circuit diagram in the Graphic Editor, but with the file extension **.bsf** (for block symbol file).

You can view and edit the logic symbol by first opening the **.bsf** file. Select **File > Open** and type in the filename. Click on the **Open** button. A window similar to Figure 11 will open showing the logic symbol.

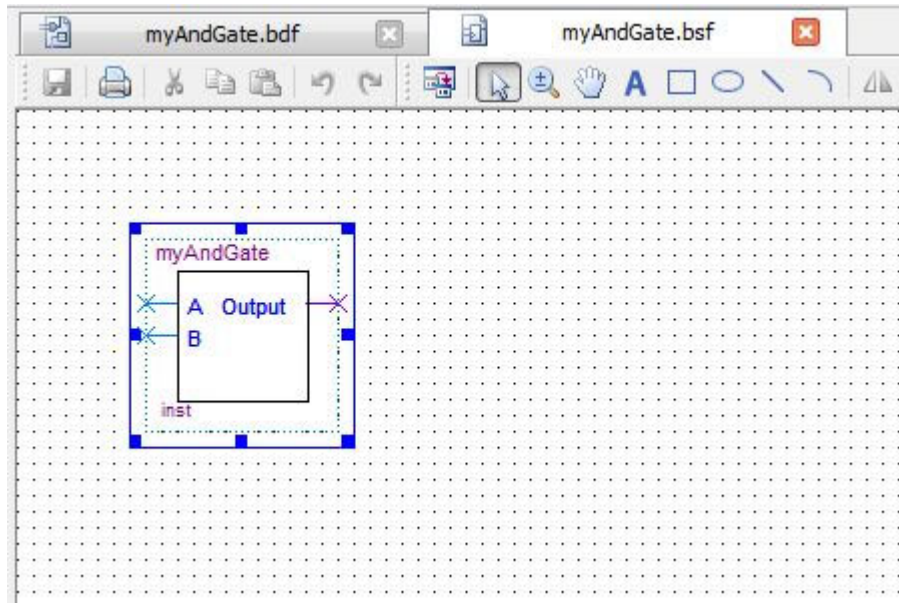


Figure 11: Logic symbol of the myAndGate circuit.

The placements of the input and output signals can be moved to different locations by dragging the signal connection line around the symbol box. The signal label will also be moved. You can then drag the label to another location if you wish. The size of the symbol can also be changed by dragging the edges of the symbol box.

This new symbol can now be used in the Block Editor. It will show up in the Symbol window under the **Project** folder as shown in Figure 12. You can follow the same steps as discussed earlier for inserting built-in logic symbols to insert this logic symbol into another schematic circuit design.

To use a circuit that is represented by its logic symbol in another project, you need to first copy the **.bsfc** symbol file and the corresponding **.bdf** circuit design file to the other project's directory. It will then be available in the Symbol window inside the Project folder as shown in Figure 12.

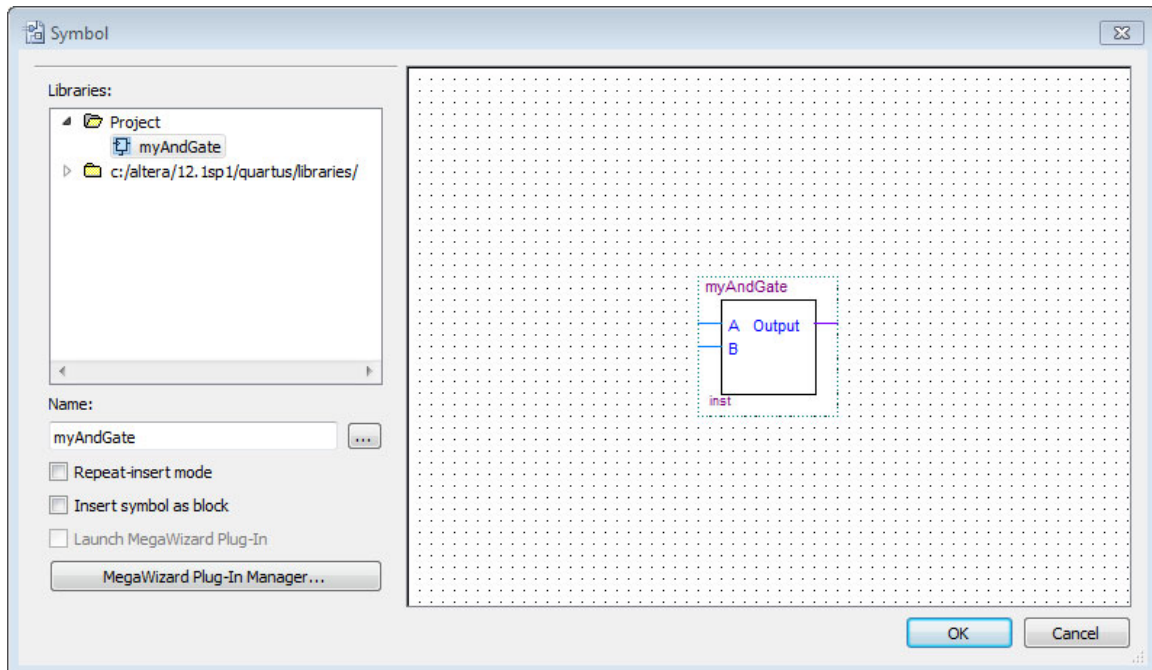



Figure 12: Selecting the myAndGate logic symbol to be inserted into another circuit design.

You can now select and use this component just like the standard components from the library.

1.1.7 Mapping the I/O Signals

Since we want to implement the circuit on a FPGA, we need to assign or map all of the I/O signals from our circuit to the actual pins on the FPGA. The following instructions are for mapping the I/O signals to the Cyclone II EP2C20F484C7 FPGA on the DE1 development board. If you are using a different FPGA development board then you will need to refer to the documentations for your development board for the correct pin assignments.

We will use the Pin Planner to map each of the I/O signals from our circuit to the pins on the Cyclone II chip. From the Quartus II menu, select **Assignments | Pin Planner** to bring up the Pin Planner similar to Figure 13.

1. Alternatively, you can click on the **Pin Planner** icon .
2. All of the available I/O signals from the circuit will be listed under the **Node Name** column. If the I/O signals are not listed, then you need to go back and do the Analysis and Synthesis step in Section 1.1.5.

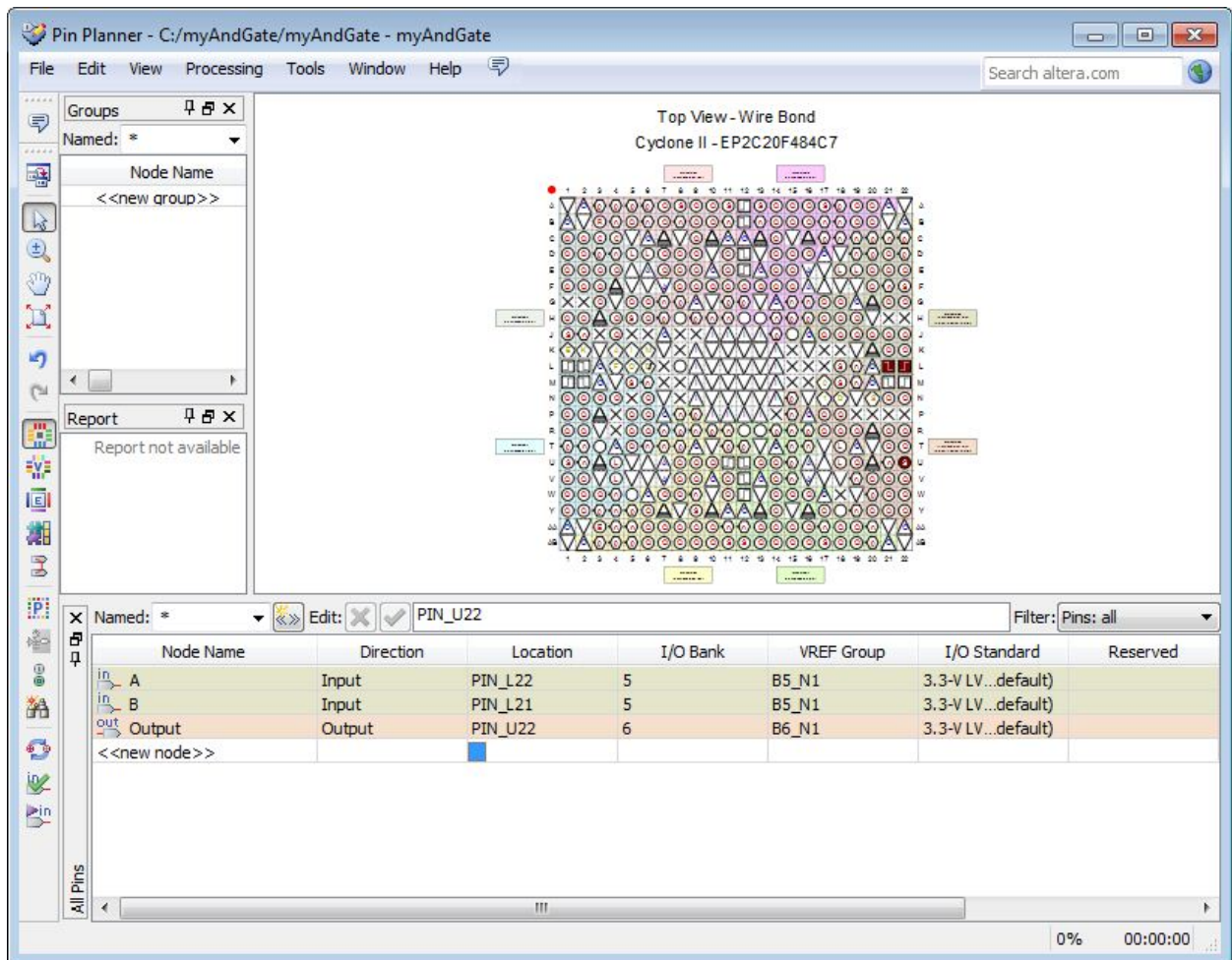


Figure 13: The Pin Planner showing the pin assignments of the EP2C20F484C7 chip.

3. For each I/O signal name, double-click on the cell next to the signal name under the **Location** column to bring up a pop-up list of all the assignable pins from the FPGA. Select the pin number that you want

to assign to that I/O signal. Figure 13 shows that Pin L22 is currently being assigned to the signal A.

- Alternatively, instead of using the pop-up list, you can type in the pin number such as L22.
- Perform the following signal-to-pin mapping for the FPGA chip.

Node Name (I/O Signal)	Location (Pin)
A	L22
B	L21
Output	U22


Faster Alternative Method for Mapping the Pins

A much faster pin mapping method is to directly edit the pin mapping file. The name of this file is <project name>.qsf located in your project directory. Use any text editor such as Windows Notepad to open and edit this file. For each pin mapping you will add a set_location_assignment line at the end of the file. Do not modify or delete any other lines in the file. For our project's pin mappings, we will have the following three lines added to the file.

```
set_location_assignment PIN_L22 -to A
set_location_assignment PIN_L21 -to B
set_location_assignment PIN_U22 -to Output
```

1.1.8 Fitting the Netlist and Pins to the FPGA

Now that we have created the netlist for the circuit (from Section 1.1.5), and have mapped all of the I/O signals to the actual pins on the PLD (from Section 1.1.7), the next step is to fit the netlist and the pin assignments to the given FPGA. This requires a full compilation of the circuit, which involves four individual steps: Analysis & Synthesis, Fitter, Assembler, and Timing Analyzer.

1. From the Quartus II menu, select **Processing | Start Compilation** to start the full compilation of the circuit. Alternatively, you can click on the **Start Compilation** icon . The full compilation will automatically go through the four steps. If there are no errors in your design, you should see the message "Full Compilation was successful" at the end of the message listing with 0 errors and many warnings. It is normal to have many warnings, and in most cases, they can safely be ignored.

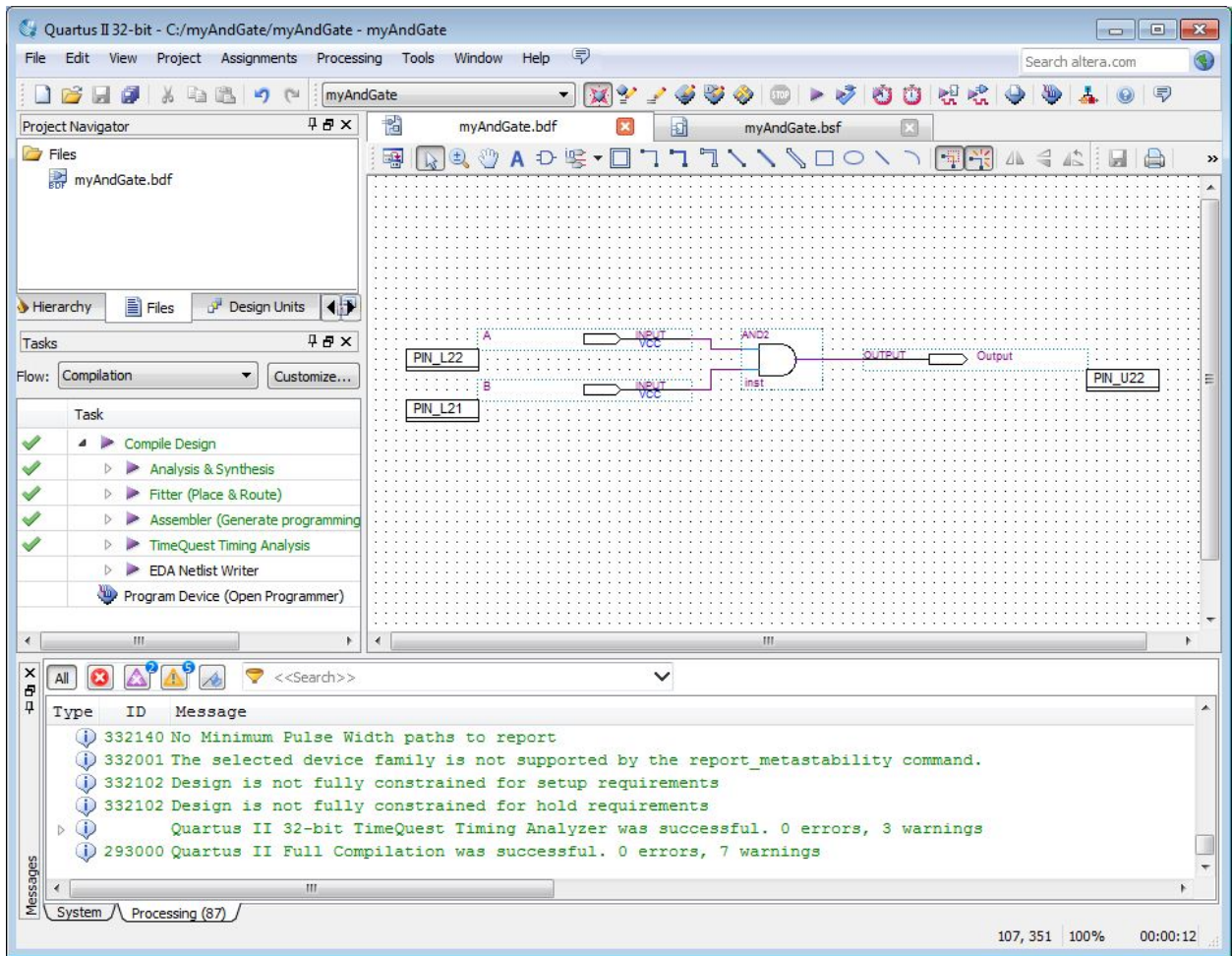




Figure 14: Successful compilation of the circuit.

1.2 Programming the circuit onto the FPGA

1. Plug in the DE1 board to the computer using the USB cable, and turn on the DE1 board.
2. From the Quartus II menu, select **Tools | Programmer** to bring up the Programmer window as shown in Figure 15. Alternatively, you can click on the **Programmer** icon .
3. The device listed next to the **Hardware Setup** button should be USB-Blaster. If it is not then click on the **Hardware Setup** button and change it.
4. Click on the **Start** button  to start the programming of your circuit, and watch the progress bar.
5. When the progress bar reaches 100%, the programming is completed, and there should be a message in the Message window saying that the operation was performed successfully.

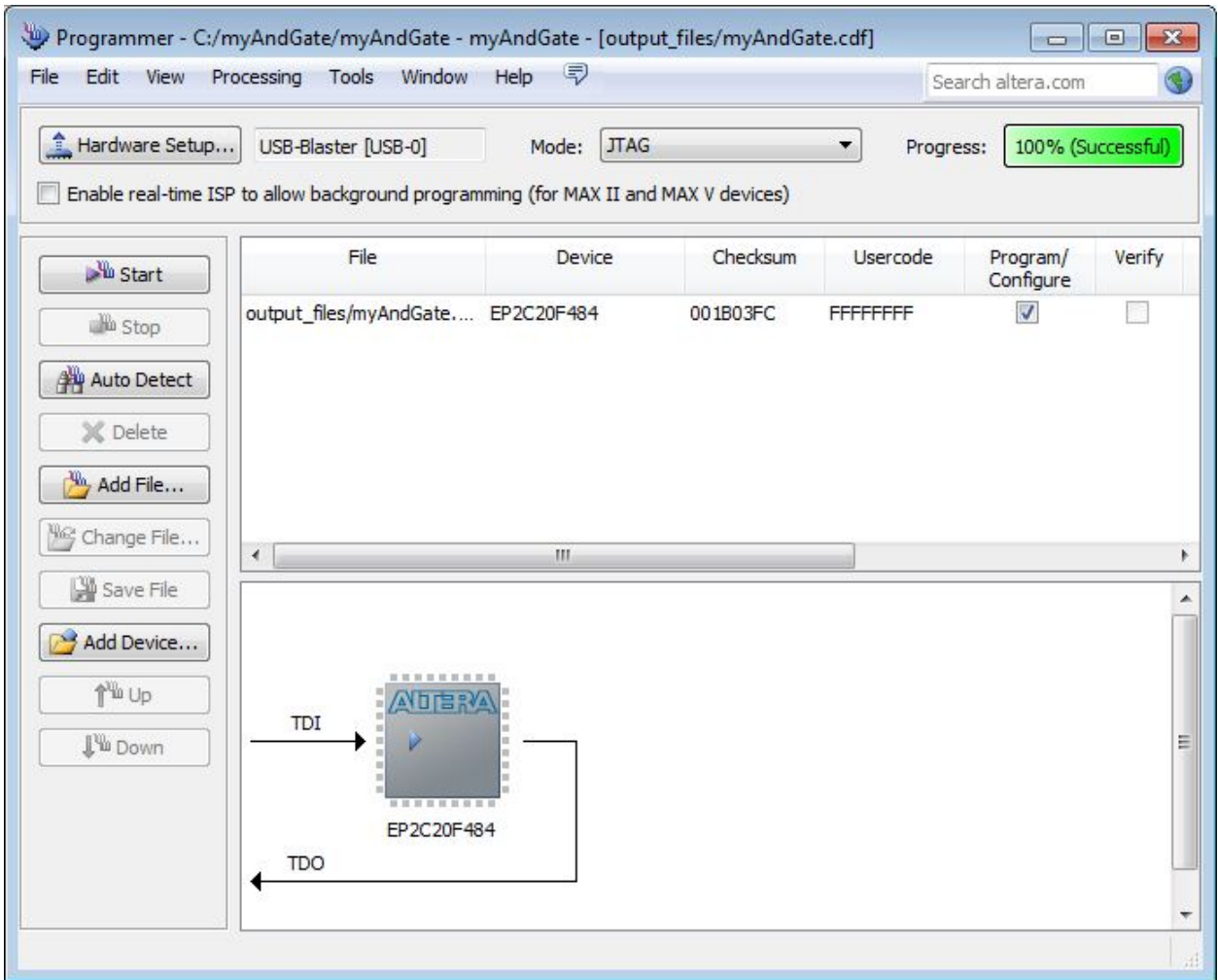


Figure 15: The Programmer window showing that the chip was programmed successfully with your circuit.

1.3 Experiments

1. This question is based on the circuit that you have uploaded to the DE1 board from section 1.2. There are ten switches on the DE1 board. A switch is on when it is in the up position, and off when it is in the down position. Set the two switches (SW0 and SW1) to be either on or off, and observe the green output LED (LEDG0) whether it is on or off. Use a 0 for off and a 1 for on. Write down your observation in the table below. This table that you have derived is known as the *truth table* for the 2-input AND gate.

A	B	Output
0 (off)	0 (off)	
0 (off)	1 (on)	
1 (on)	0 (off)	
1 (on)	1 (on)	

2. Replace the 2-input AND gate in your circuit with a 2-input OR gate (or2). Derive the truth table for the 2-input OR gate by filling in the table below.

A	B	Output
0	0	
0	1	
1	0	
1	1	

3. Replace your circuit with a NOT gate. You'll have to remove one of the input pins. Derive the truth table for the NOT gate by filling in the table below.

A	Output
0	
1	

4. Replace your circuit with a 3-input AND gate. You will need to have three input pins. Refer to the [DE1 pin mappings](#) document to find the pin number for your third switch (SW2). Derive the truth table for the 3-input AND gate by filling in the table below.

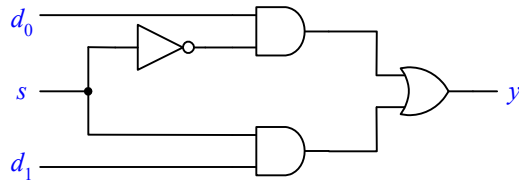
A	B	C	Output
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	

1	1	1	
---	---	---	--

5. Replace your circuit with a 3-input OR gate. Derive the truth table for the 3-input OR gate by filling in the table below.

A	B	C	Output
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

6. Draw and implement the following circuit. This circuit is known as the Multiplexor or Mux for short. Derive the truth table for this circuit. What is happening with the y output? Hint: look at the first four lines of the y output, and then look at the last four lines of the y output.

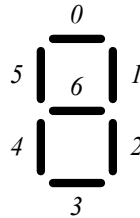


s	d_1	d_0	y (Output)
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

7. Repeat Experiment 1 but use two push buttons (PB0 and PB1) instead of the two switches. What do you notice about the operation of the push buttons?
8. Connect a circuit having one switch and one LED. Make the LED turn on when the switch is on, and off otherwise.
9. Draw some random circuit having three inputs and one output. Randomly connect several AND gates, OR gates and NOT gates together between the inputs and output. Derive the truth table for it.

A	B	C	Output
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

10. Make the seven-segment (HEX) displays to display the number 256 by turning on the appropriate LEDs for each digit. The seven LEDs for each digit is named as follows:



The LEDs in the seven-segment displays are turned on with a 0 rather than a 1 as in the discrete LEDs and turn off with a 1. To get a logic 1 signal use the component VCC from the library. To get a logic 0 signal use the component GND from the library.

The right-most digit on the development board is named HEX0 with the subscripts 0 to 6 matching those in the picture above. The next digit over is HEX1 also with subscripts 0 to 6, etc.

When you use subscripts for your input/output pin names such as HEX0[1] and HEX0[4], you **MUST** also have names from subscript 0 to the highest subscript number 4. In other words, you need to have the pins with the names HEX0[0], HEX0[1], HEX0[2], HEX0[3] and HEX0[4] mapped. Otherwise, your circuit will not work.