CPE 2211 COMPUTER ENGINEERING LAB

EXPERIMENT 2 LAB MANUAL

Revised: J. Tichenor FS19

SCHEMATIC CAPTURE AND LOGIC SIMULATION

(ELECTRONIC DESIGN AUTOMATION: EDA)

OBJECTIVES

In this experiment you will

- Become familiar with two common electronic design automation tools: Schematic Capture and Logic Simulation.
- Use the suite of EDA tools supplied by Altera: Quartus-II and ModelSim.
- Enter a circuit schematic and simulate to verify correctness.

LAB REPORTS

The format of lab reports should be such that the information can be used to reproduce the lab, including what values were used in a circuit, why the values were used, how the values were determined, and any results and observations made. This lab manual will be used as a guide for what calculations need to be made, what values need to be recorded, and various other questions. The lab report does not need to repeat everything from the manual verbatim, but it does need to include enough information for a 3rd party to be able to use the report to obtain the same observations and answers. Throughout the lab manual, in the Preliminary (if there is one), and in the Procedure, there are areas designated by **Qxx followed by a question or statement**. These areas will be **bold**, and the lab TA will be looking for an answer or image for each. These answers or images are to be included in the lab report. The lab TA will let you know if the lab report will be paper form, or if you will be able to submit electronically.

PRELIMINARY

Read through the tutorial to familiarize yourself with the steps you will be performing. You may want to highlight these steps in your lab book before coming to lab. Write truth tables for the four primitive gates used in the tutorial. Find data sheets for the parts on the internet.

PROCEDURE

Work through the tutorial in the lab. Take notes as you are going through the procedures. You may want to refer back to your notes later when doing other labs. You might want to use a highlighter in this lab so you can find important parts quickly later on, you will be using the information learned herein in future labs. When finished with the tutorial, submit a screen shot of the schematic, and ModelSim output results along with any other material your TA might require.

TUTORIAL

Using Schematic Capture with Altera Quartus-II

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 12.0 is used in this tutorial. For simplicity, in our discussion we will refer to this software package simply as Quartus II. In this tutorial the design of logic circuits using Quartus II is introduced. Step-by-step instructions are presented for performing design entry with two methods: using schematic capture and writing Verilog code, as well as with a combination of the two.

1. Introduction

This tutorial assumes that the reader has access to a computer on which Quartus II is installed or accessible via AppsAnywhere. You can find Quartus II in AppsAnywhere as shown in Figure 1.

Figure 1: Quartus II Launch in AppsAnywhere.

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 computers 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.

1.1 Getting Started

Each logic circuit, or sub-circuit, 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 or folder in the file system. To begin a new logic circuit design, the first step is to create a directory to hold its files. The project should be stored on the S drive so that access to the project can occur on any ECE CLC computer. Please create a folder named CpE2211 in your S drive and a subfolder named lab2 inside it.

Upon starting the Quartus II software you will see a display similar to the one in Figure 2. This display will allow you to initiate the new project wizard or open an existing project. If you don't see this display you will be seeing the main window of Quartus II (Figure 3).

Figure 2: Quartus II Startup window.

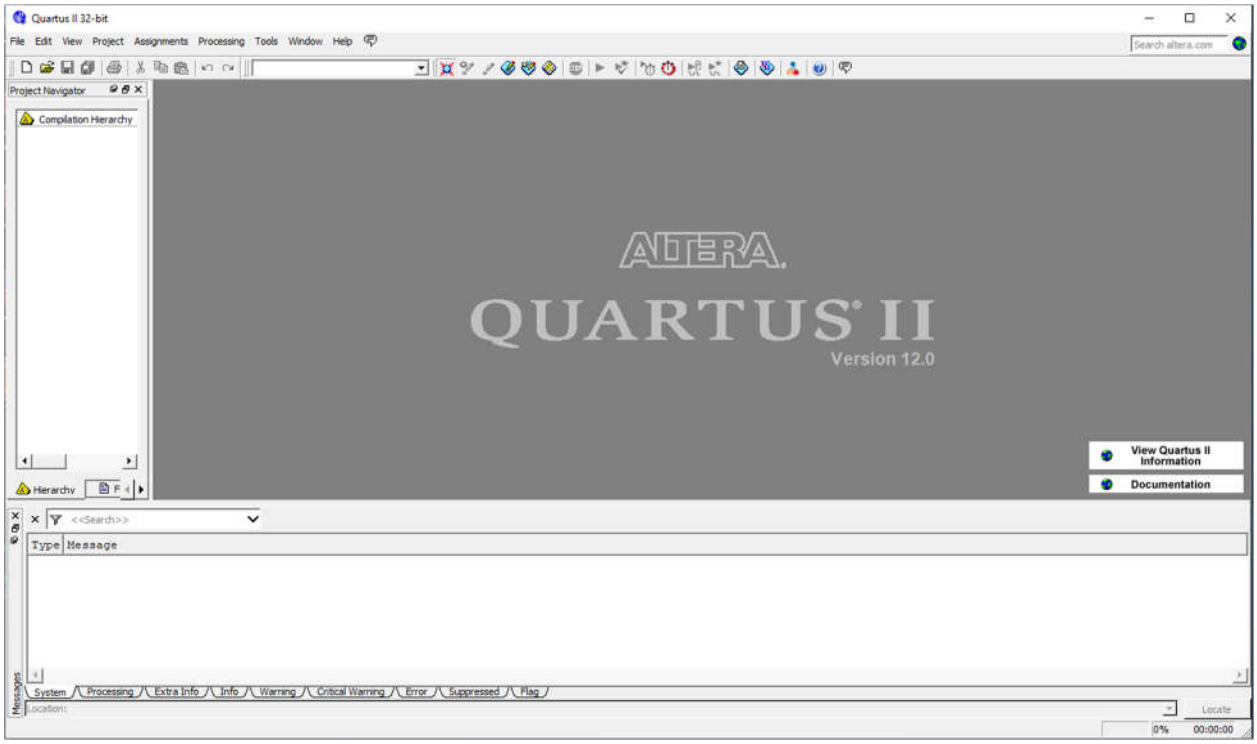


Figure 3: Quartus II main window.

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 3 clicking the mouse button on the menu named **File** opens the menu shown in Figure 4. In general, whenever the mouse is employed to select something the *left* button is used, and therefore the mouse button to press will not be specified. 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. The convention **Menu1 | Menu2 | Item** will be used 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 Quartus II system. Many Quartus II commands have an associated icon displayed in one of the toolbars. To see the list of available toolbars, select **Tools | Customize | Toolbars**. Once a toolbar is opened, it can be moved with the mouse, and icons can be dragged from one toolbar to another. To see the Quartus II command associated with an icon, position the mouse cursor on top of the icon and a tooltip will appear that displays the command name.

It is possible to modify the appearance of the Quartus II display in Figure 3 in many ways. Section 6 shows how to move, resize, close, and open windows within the main Quartus II display.

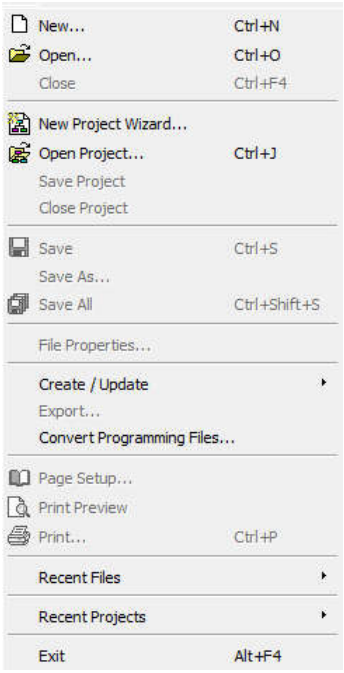


*Figure 4: **File** menu pop-up.*

1.2 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.

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 5. Set the working directory to be *S:/CpE2211/lab2*. The project must have a name, which may optionally be the same as the name of the directory. We have chosen the name *lab2*. Observe that Quartus II automatically suggests that the name *lab2* be also the name of the top-level design entity in the project. This is a reasonable suggestion, but it can be ignored if the user wants to use a different name.

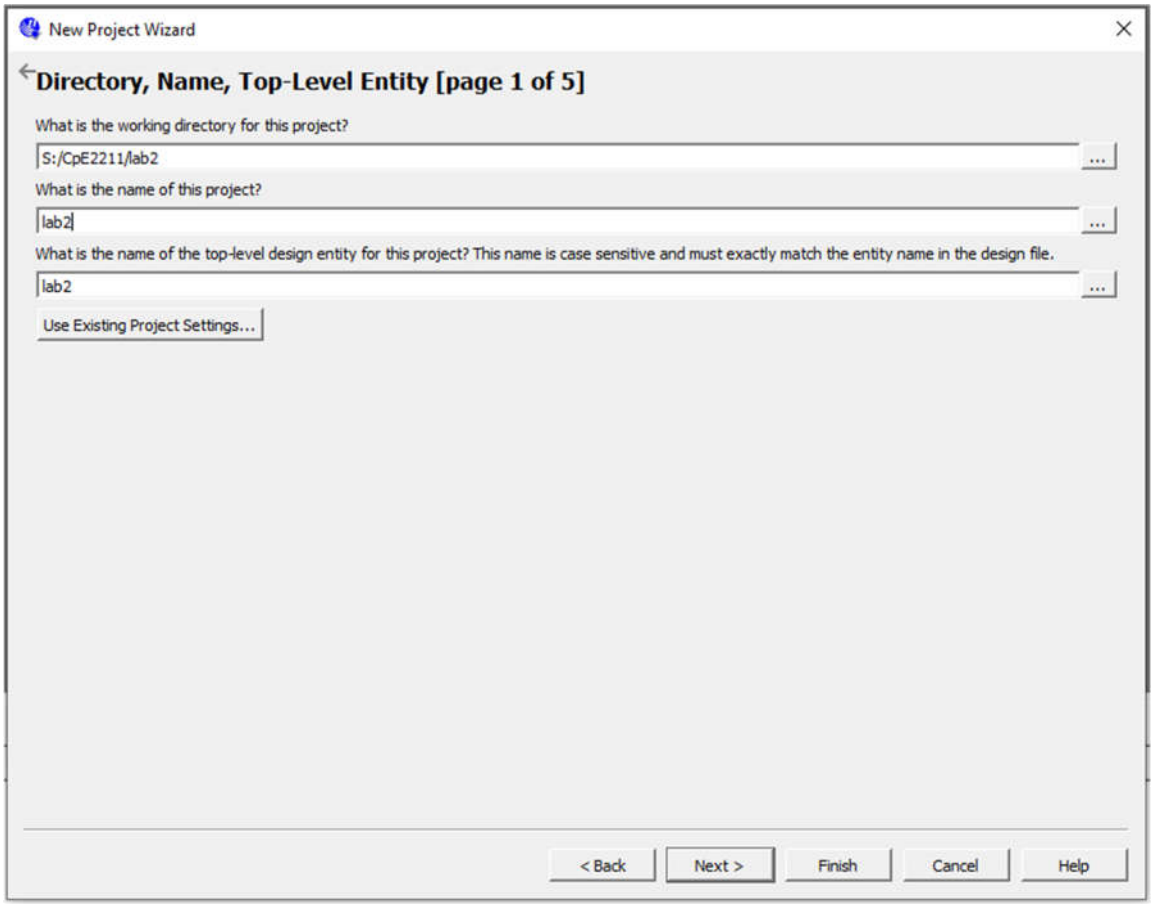


Figure 5: New Project Wizard, Directory, Name, Top-Level Entity.

Press **Next**, 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 **Next**.

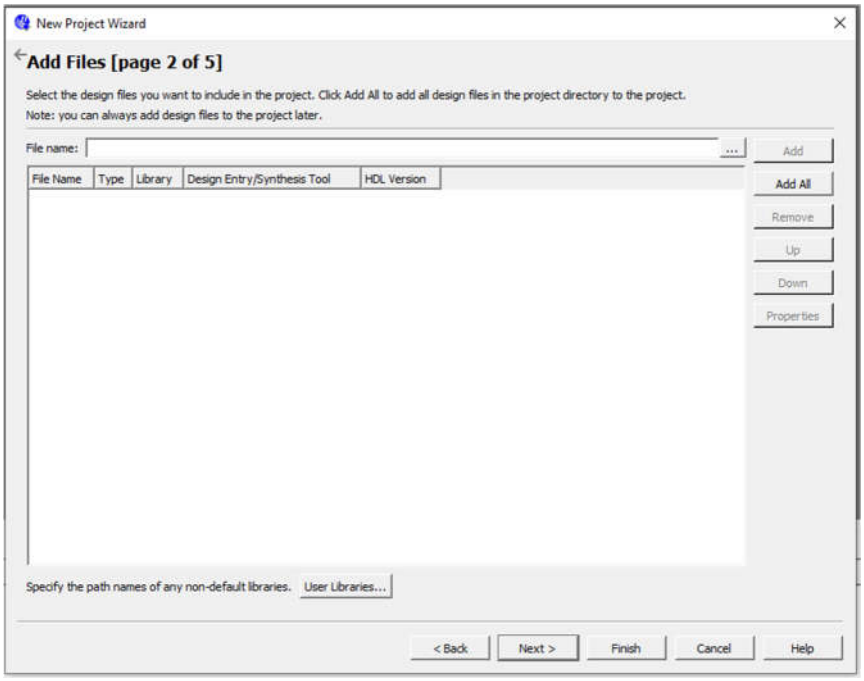


Figure 6: New Project Wizard, Add Files.

Now, the window in Figure 7 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. Choose the device family called Cyclone II, which is the FPGA provided for use in the CpE2211 lab (the Altera DE2 development board). Select **EP2C35F672C6** from the **Available devices** list.

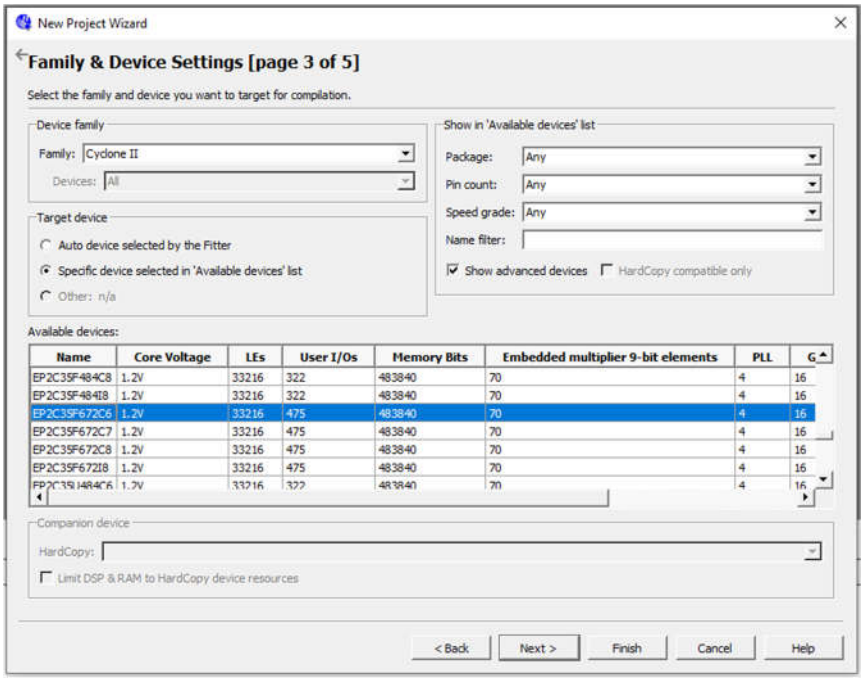


Figure 7: New Project Wizard, Family & Device Settings.

Press **Next** to go to the window shown in Figure 8. Here, we can specify third-party CAD tools (i.e. those that are not a part of Quartus II software) that should be used. In this tutorial the term CAD tools refers to software packages developed for use in Computer Aided Design. Another term for software of this type is EDA tools, where the acronym stands for Electronic Design Automation. This term is mostly used in Quartus II messages that refer to third party tools, which are the tools developed and marketed by companies other than Altera. You will see that ModelSim by Altera is selected by default for simulation. Clicking **Next** will show you a summary of the project settings similar to Figure 9. Click finish to end the new project wizard.

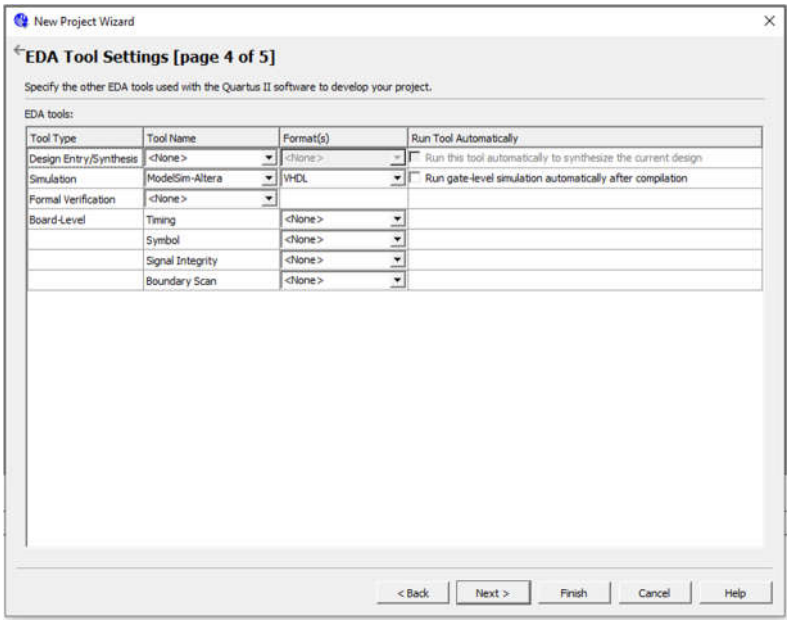


Figure 8: New Project Wizard, EDA Tool Settings.

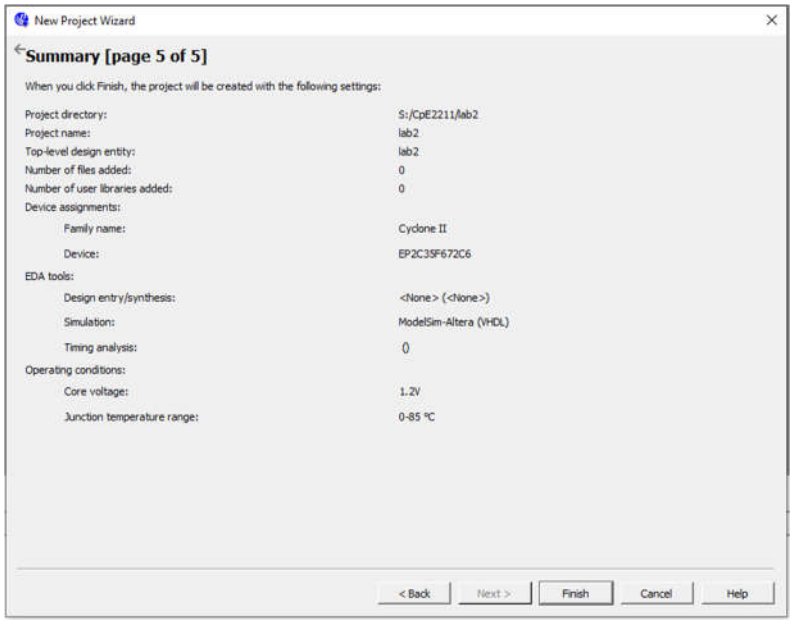


Figure 9: New Project Wizard, Summary.

3. Design Entry Using Schematic Capture

This section illustrates the process of using the schematic capture tool provided in Quartus II. And a simple example schematic using a 4 logic function including; AND, NAND, NOR, and OR will be drawn. After creating the schematic, ModelSim will be used to verify the correctness of the circuit.

3.1 Using the Block Diagram/Schematic File Editor

The first step is to draw the schematic. In the Quartus II display select **File | New**. The window shown in Figure 10 will appear, allowing the designer to choose the type of file to be created. The possible file types include schematics, Verilog code, and other hardware description language files such as VHDL and AHDL (Altera's proprietary Hardware Description Language or 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. Choose **Block Diagram/Schematic File** and click OK. This selection opens the Block Diagram/Schematic Editor window shown on the right side of Figure 11. Drawing a circuit in this window will produce the desired block diagram file.

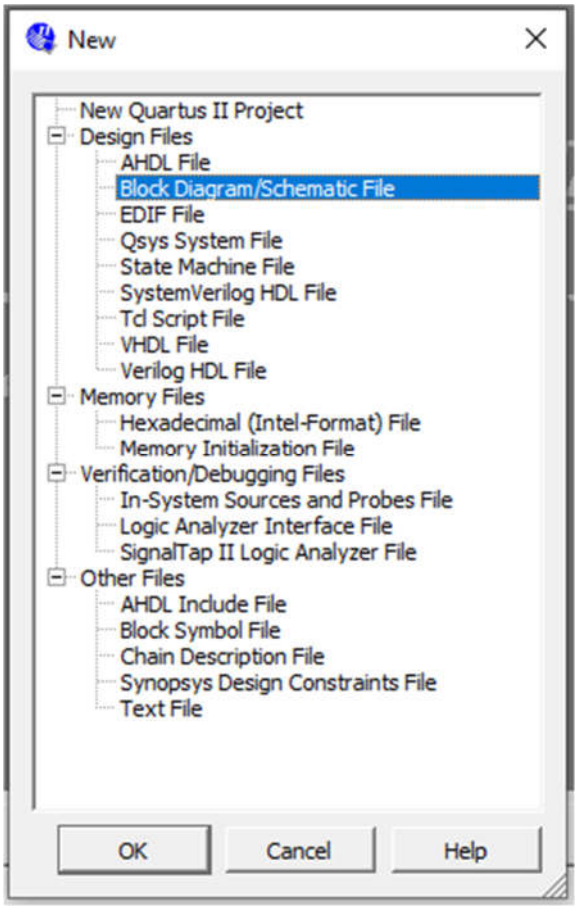


Figure 10: Block Diagram/Schematic File.

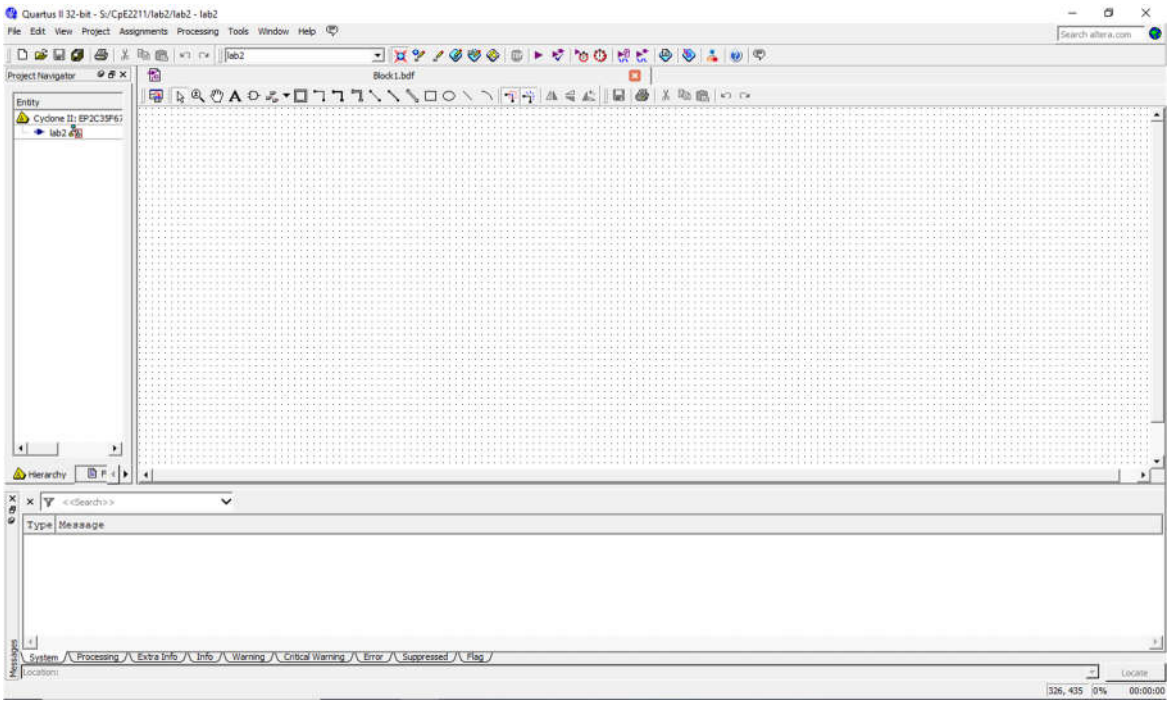


Figure 11: Block Diagram/Schematic Editor.

Importing Logic Gate Symbols

The Block Diagram/Schematic Editor provides several libraries that contain circuit elements which can be imported into a schematic. For our simple example use a library called *primitives*, which contains basic logic gates, VCC source, GND, and Input/Ouput pins. To access the library, **double-click on the blank space inside the Block Editor** display to open the window in Figure 12 (another way to open this window is to select the AND gate icon from the toolbar as shown in Figure 13).

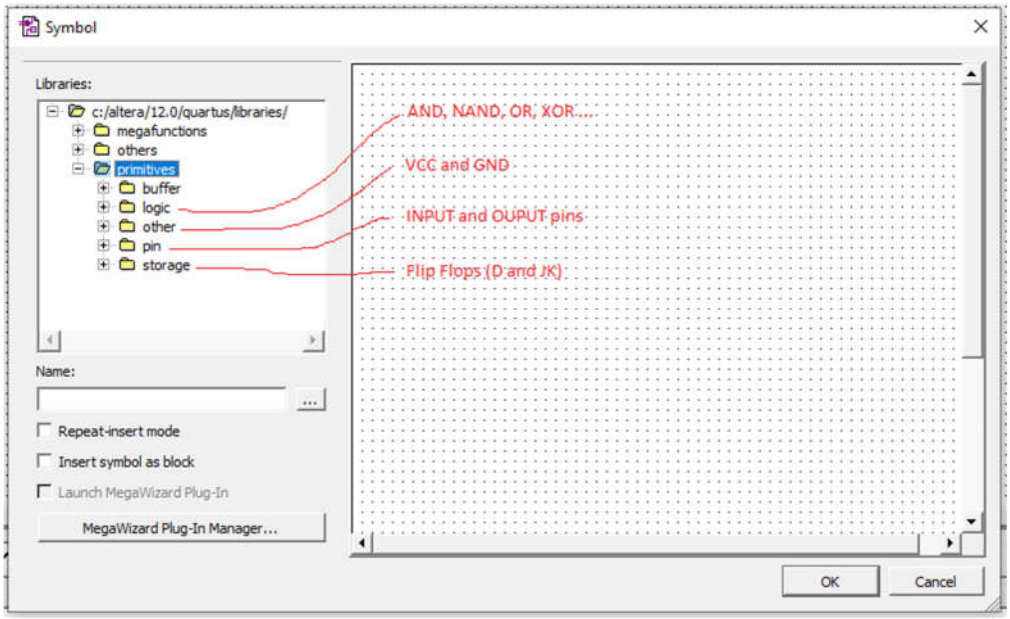


Figure 12: Primitives for schematic entry.

Figure 13: AND symbol selection from toolbar for Primitives.

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/12.0/quartus/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 Diagram/Schematic 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. Notice in Figure 12 that other useful item locations are highlighted in red. These include VCC and GND, Input/Output pins, and Flip-Flops. These will be needed in later labs.

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 by selecting **View | Show Guidelines**.

The logic function also requires a two-input NAND gate, a two-input OR gate, and a two input NOR gate. To import the OR-gate symbol, again double-click on a blank space in the Block Diagram/Schematic 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. 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 14.

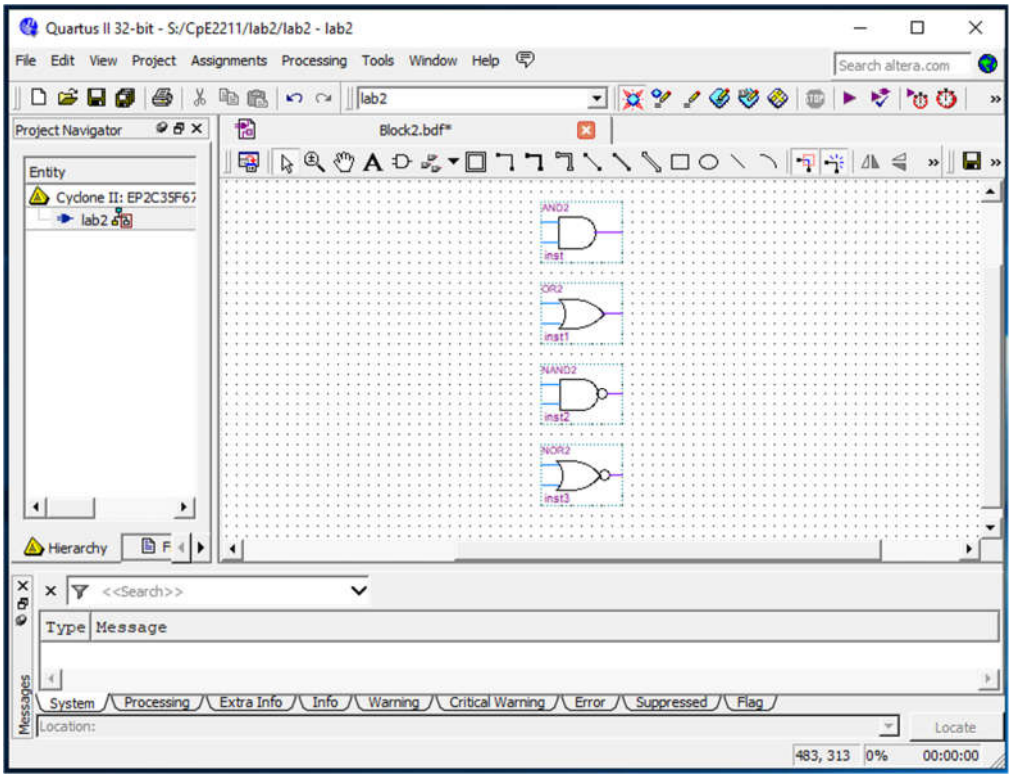


Figure 14: Schematic containing the 4 logic gates, AND, OR, NAND, and NOR.

The following steps can be used to copy a component already placed in 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 Diagram/Schematic Editor automatically imports a second instance of the AND-gate symbol.

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 primitives' 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. To represent the output of the circuit, open the primitives' library and import the symbol named ***output***. Arrange the symbols to appear as illustrated in Figure 15.

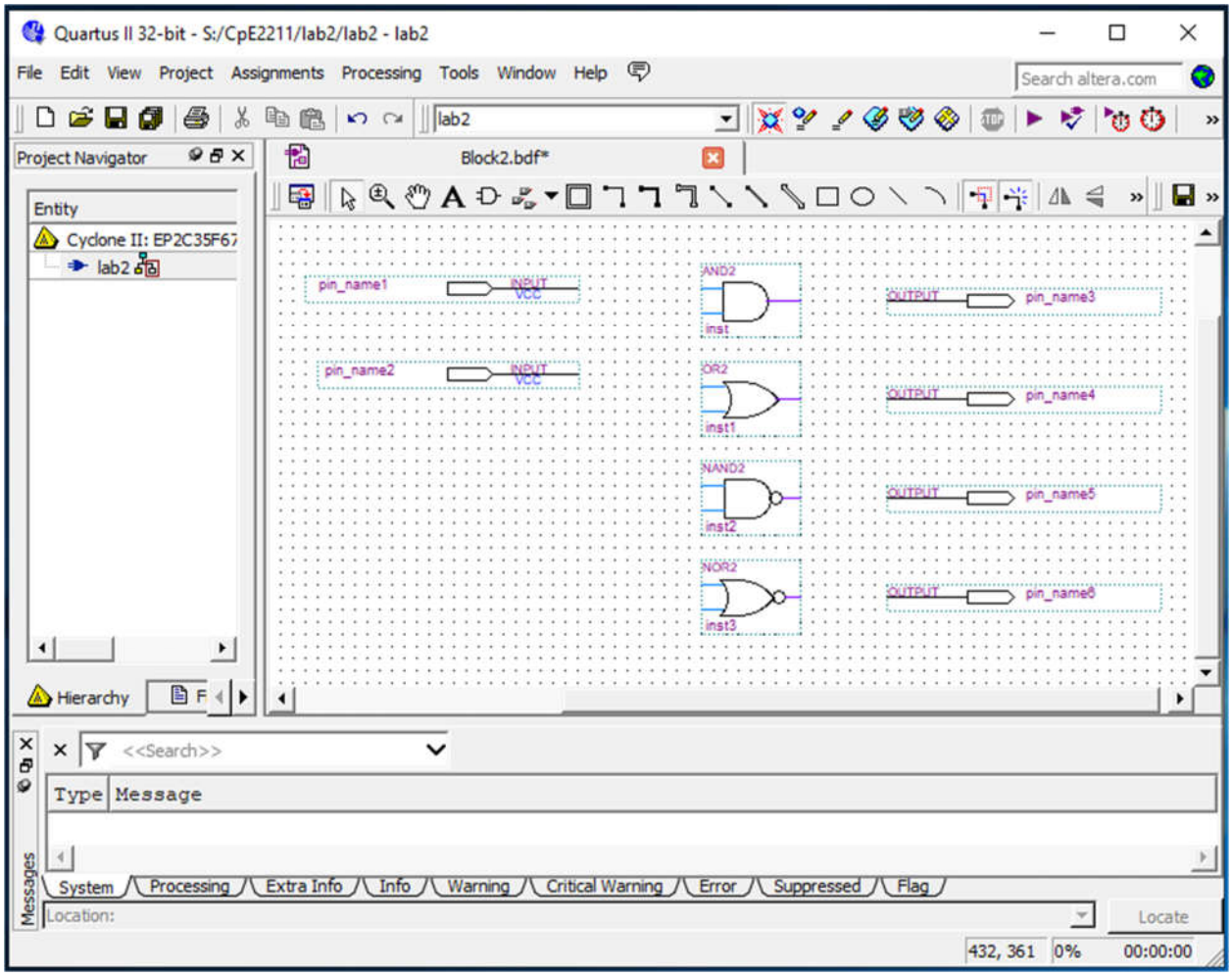


Figure 14: Schematic with Input and Output pins added.

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 **a** and **b** as the input 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. Finally, assign the name ***and2***, ***nand2***, ***or2*** and ***nor2*** to the output pins.

Connecting Nodes with Wires

The next step is to draw lines (wires) to connect the symbols in the schematic together. Move the mouse pointer on top of the **a** 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 pinstub, on the right edge of the **a** 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. Another way to begin wiring is to select the icon highlighted in Figure 15.

Figure 15: Wire tool initiated by selecting the icon in red circle.

Connect the input symbol for **a** to the AND gate at the top of the schematic as follows. While the mouse is pointing at the pinstub on the **a** 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 **b** input symbol to the other input on the AND gate. Finally, the circuit appears as illustrated in Figure 16. 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**. Save the schematic using **File | Save As** and choose the name example schematic. Note that the saved file is called **lab02.bdf**.

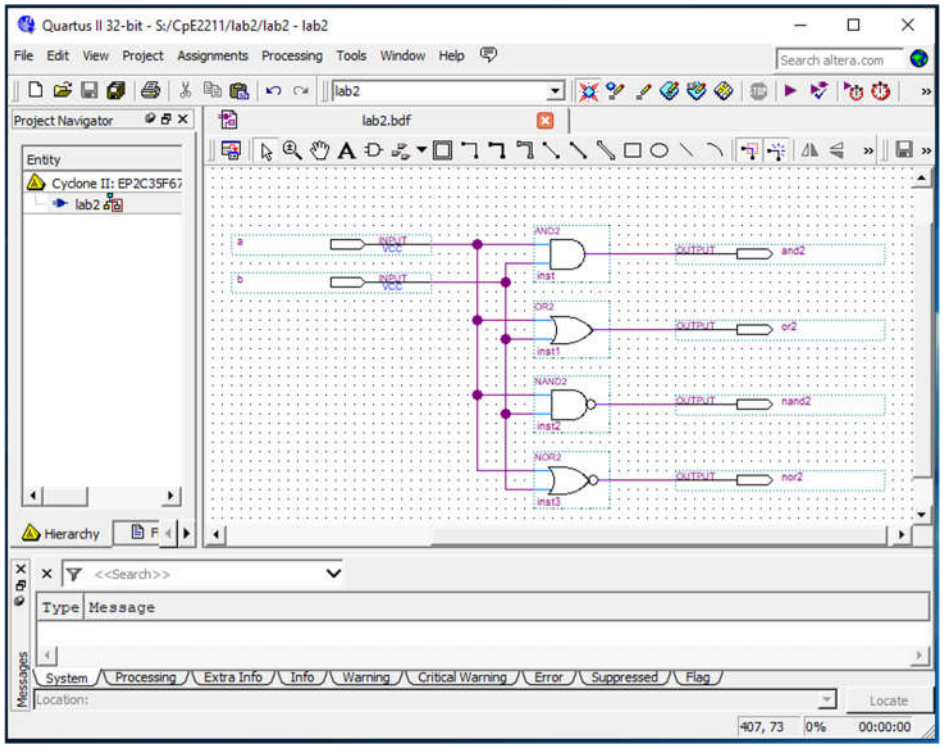


Figure 16: Wired schematic.

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 is a default setting. There is a

rubberbanding icon (shown in Figure 17). 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.

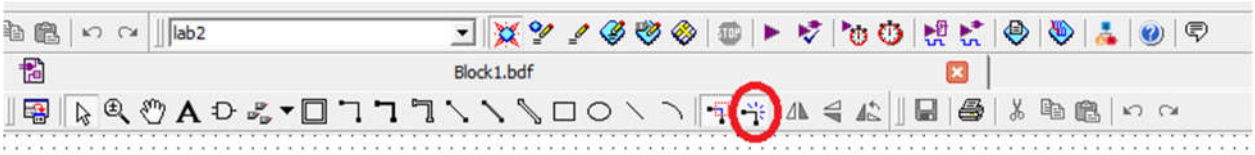


Figure 17: Rubberbanding tool highlighted by red circle.

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.

3.2 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

The CAD tools available in Quartus II are divided into a number of modules. Select **Processing | Start** to show the window in Figure 18, which contain all modules. The **Start Analysis & Synthesis** module performs the synthesis step in Quartus II. It produces a circuit of logic elements, where each element can be directly implemented in the target chip. The **Start Fitter** module determines the exact location on the chip where each of these elements produced by synthesis will be implemented.

These Quartus II modules are controlled by an application program called the Compiler. The Compiler can be used to run a single module at a time, or it can invoke multiple modules in sequence. There are several ways to access the Compiler in the Quartus II user interface. Pressing the **Start Compilation** button will run Analysis & Synthesis, Fitter, Assembler, and Timing Analyzer in sequence.

Another convenient way of accessing the Compiler is to use the **start compilation** icon  from the toolbar. The command for running the synthesis module is **Processing | Start | Start Analysis & Synthesis**. Part of the synthesis module can also be invoked by using the command **Processing | Start | Start Analysis & Elaboration**. This command runs only the early part of synthesis, which checks the design project for syntax errors, and identifies the major sub-design names that are present in the project.

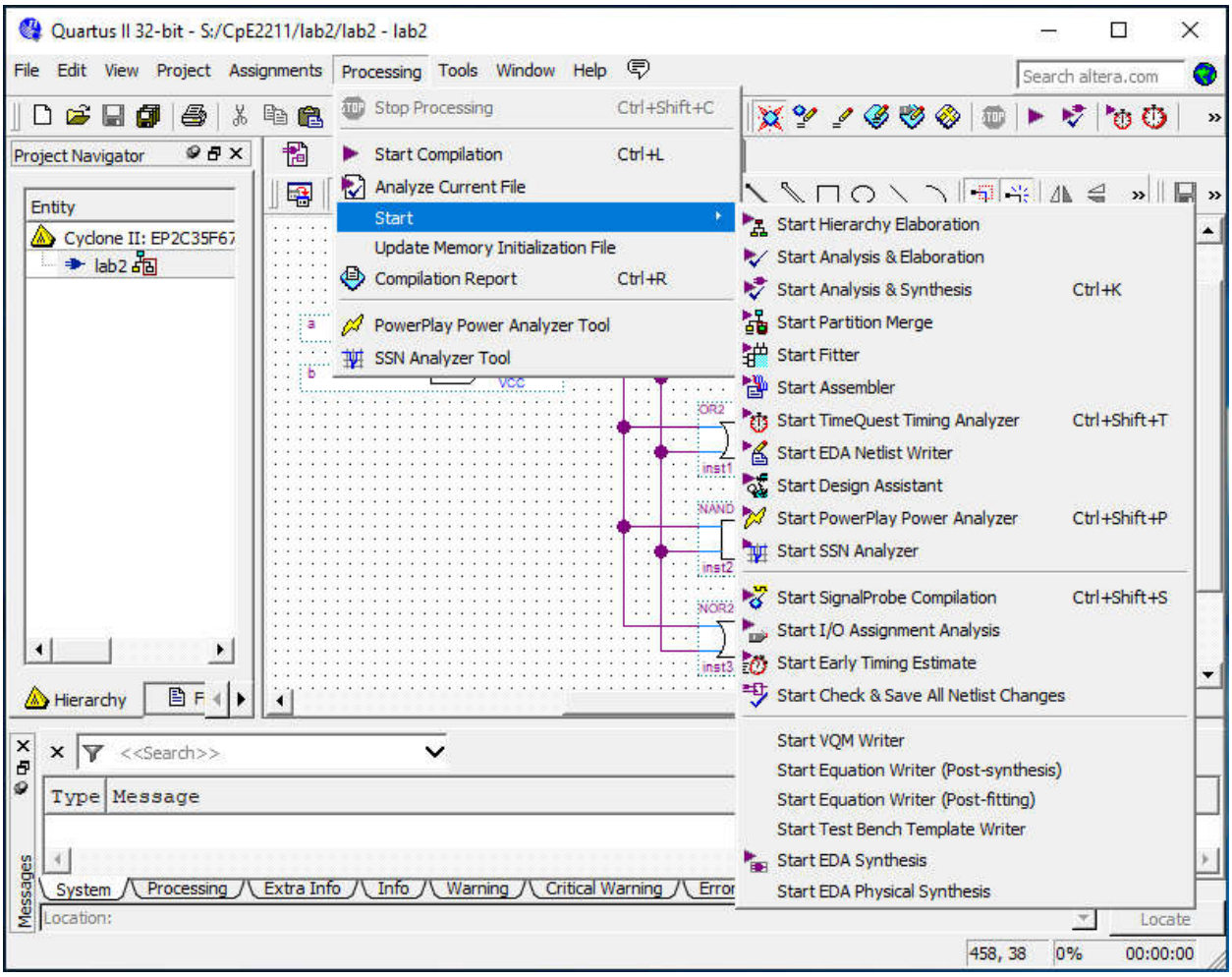


Figure 18: Processing modules.

An efficient way of using the CAD tools is to run only the modules that are needed at any particular phase of the design process. This approach is pragmatic because some of the CAD tools may require a long time, on the order of hours, to complete when processing a large design project. The purpose of this tutorial is to perform a functional simulation of the schematic, and since only the output of synthesis is needed to perform this task, only the synthesis module need be ran.

Select **Processing | Start | Start Analysis & Synthesis**, use the corresponding icon in the toolbar, or use the shortcut Ctrl-k. As the compilation proceeds, its progress is reported in the lower right corner of the Quartus II display, and also in the Status utility window on the left side (if this window is not open it can be accessed by selecting **View | Utility Windows | Status**). Successful (or unsuccessful) compilation is indicated in a pop-up box. Acknowledge it by clicking **OK** and examine the compilation report depicted in Figure 19 (if the report is not already opened, it can be accessed by clicking on the **Report** icon in the Compiler Tool window, using the appropriate icon in the toolbar, or by selecting **Processing | Compilation Report**.

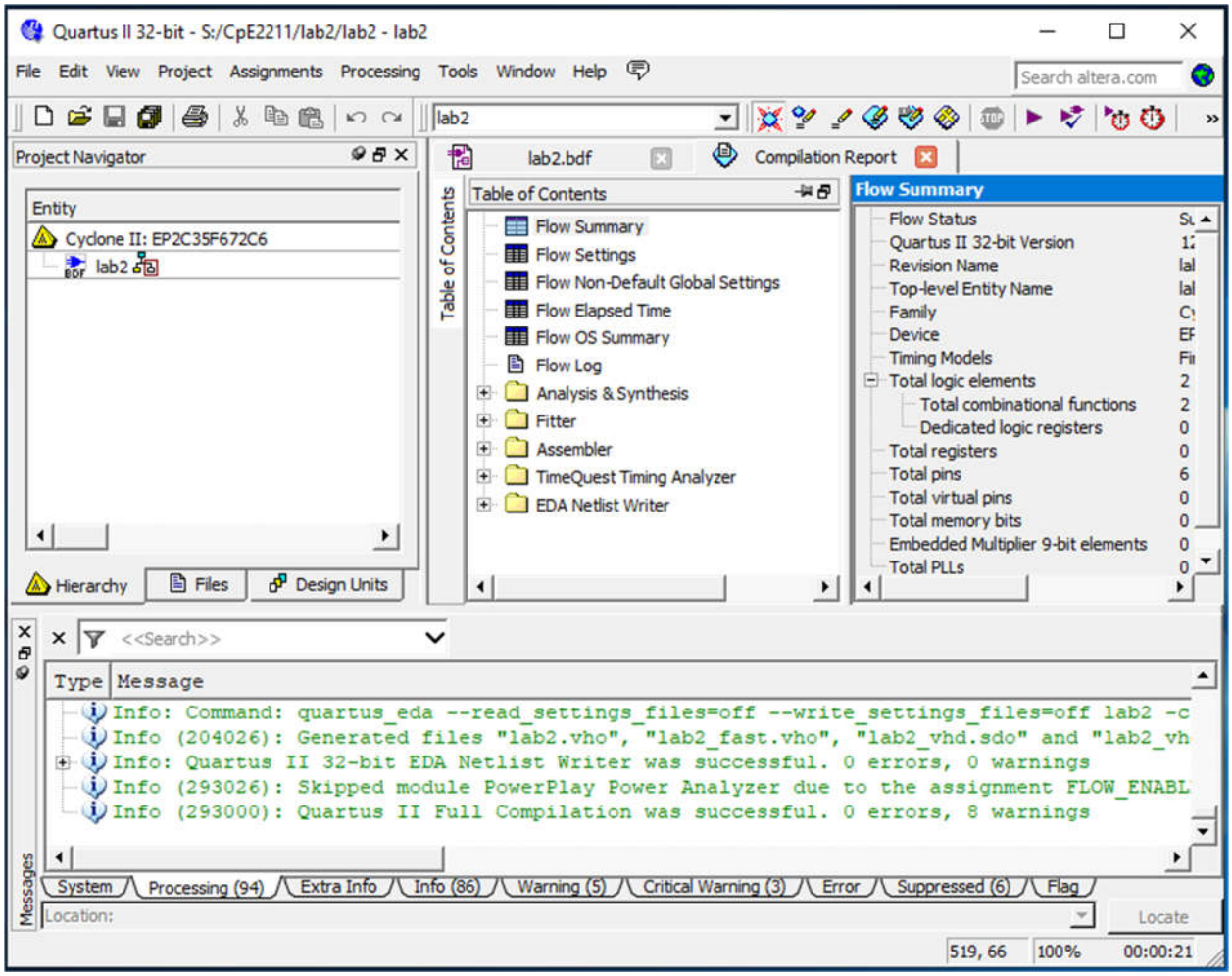


Figure 19: Compilation Report.

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 19. 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 a 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.

4. Create VHDL code from Schematic

Upon getting successful compilation, go to **File**, click **Create/Update** and **Create HDL Design File from Current File** as shown in Figure 20. Another window (Figure 21) will pop up; choose **VHDL** in the **File type**. Then save the lab02.vhd file in the required directory.

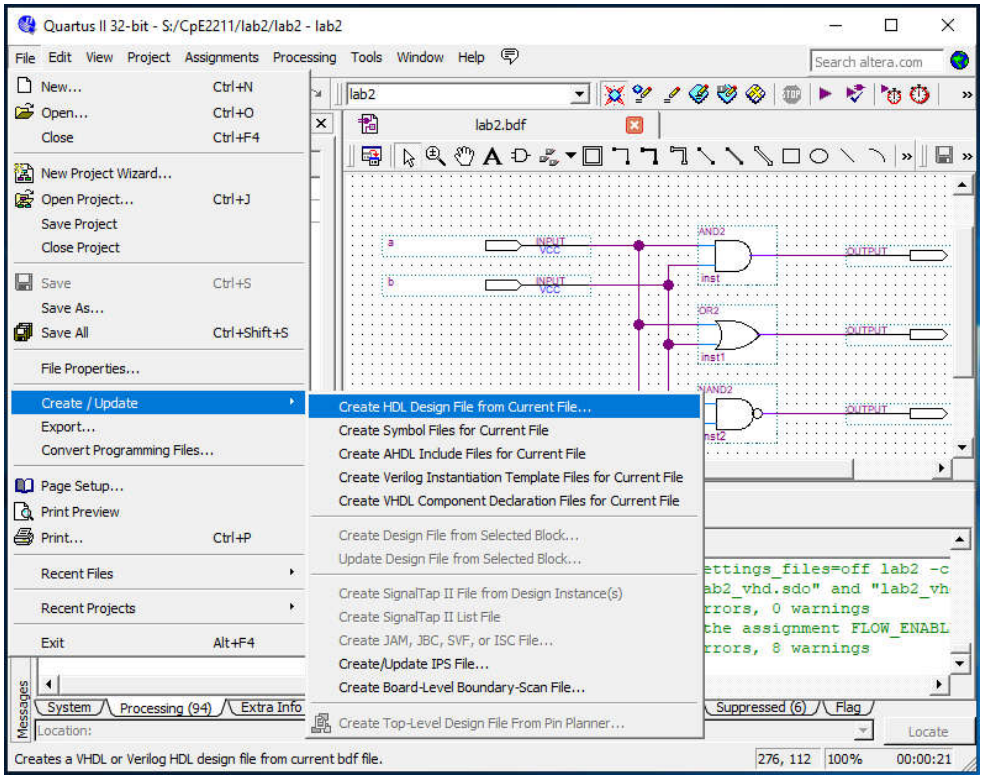


Figure 20: Create/Update, Create HDL Design File from Current File…

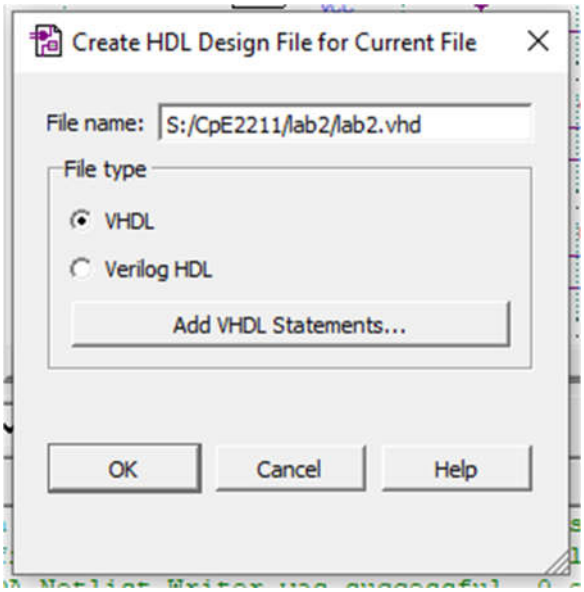


Figure 21: Create HDL Design File for Current File, Select VHDL.

TUTORIAL

Simulation using ModelSim-Altera

Very High Speed Integrated Circuit Hardware Description Language, or VHDL, is an increasingly important tool in digital circuit design used for automated specification and testing of digital systems. VHDL code can be created automatically from the schematic in Quartus II. You can perform a functional and/or a timing simulation of a Quartus II-generated design with the Mentor Graphics ModelSim-Altera software (OEM) or the ModelSim PE or SE (non-OEM) software. The ModelSim software is a dual-language simulator, meaning you can simulate designs containing either Verilog HDL, VHDL, or both. You can use designs in which a Verilog HDL module instantiates VHDL entities or a VHDL module instantiates Verilog HDL entities. You can also use the ModelSim software to perform a functional or timing simulation of Excalibur designs using the bus functional model or full stripe model, respectively.

1. Getting Started

This tutorial assumes that the reader has access to a computer on which ModelSim is installed or accessible via AppsAnywhere. You can find ModelSim in AppsAnywhere as shown in Figure 22.

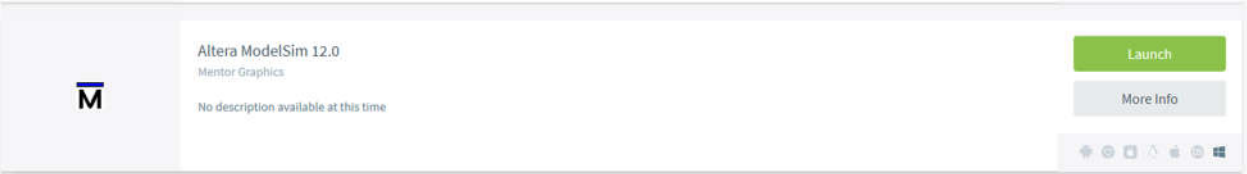


Figure 22: ModelSim Launch in AppsAnywhere.

Start the ModelSim software, and you should see a display similar to the one in Figure 23. Select **File | New | Project** to create a new project. This opens the **Create Project** dialog (shown in Figure 24), where you can specify a project name, location, and default library name. You should select the **Project Location** of the project you created in Quarus II (***S:/CpE2211/lab2*** for this tutorial***)***. The **Default Library Name** "work" is fine to keep. This dialog also allows you to reference library settings from a selected .ini file or copy them directly into the project.

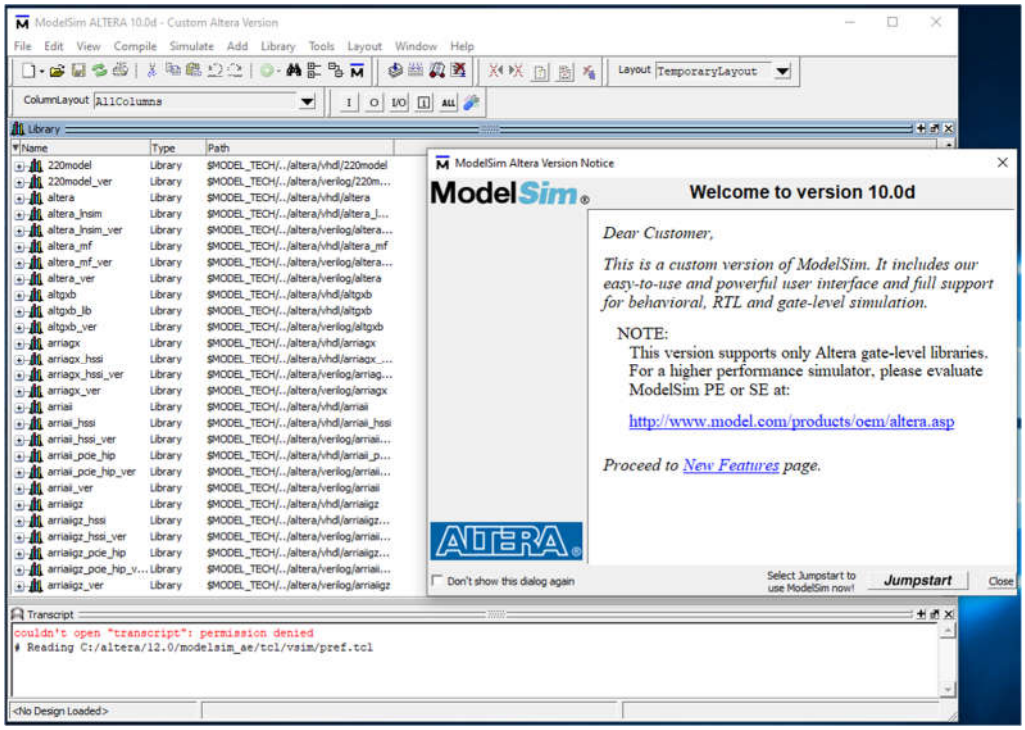


Figure 23: ModelSim startup.

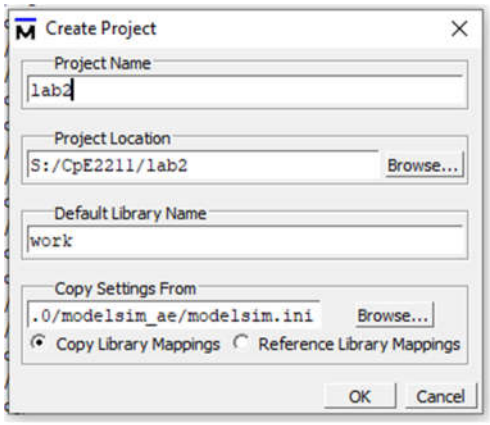


Figure 24: ModelSim, Create Project popup.

After selecting **OK**, you will see a blank Project tab in the Workspace pane of the Main window (Figure 25) and the **Add Items to the Project** dialog (Figure 26).

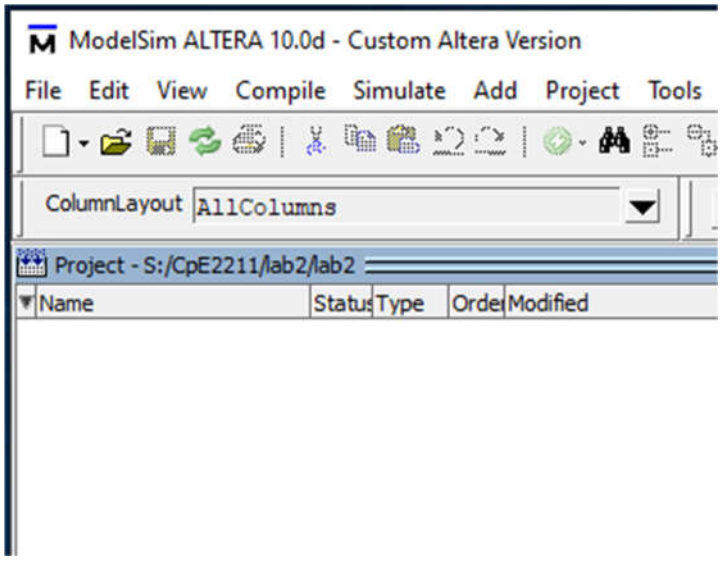


Figure: 25: Project tab.

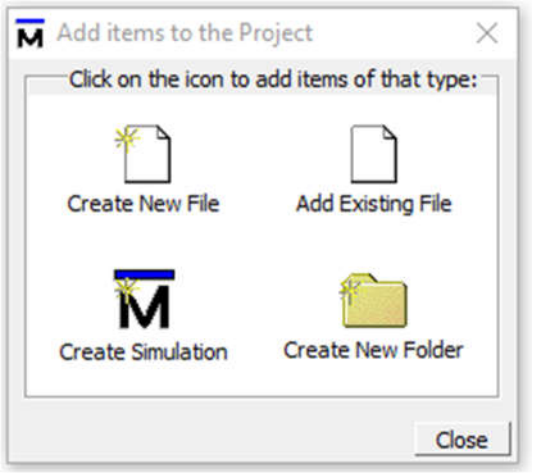


Figure 26: Add items to the Project popup.

Adding Items to the Project

The **Add Items to the Project** dialog includes these options:

- **Create New File** — Create a new VHDL, Verilog, Tcl, or text file using the Source editor.
- **Add Existing File** — Add an existing file.
- **Create Simulation** — Create a Simulation Configuration that specifies source files and simulator options.
- **Create New Folder** — Create an organization folder.

By clicking on **Add Existing File** or selecting **Project** | **Add to Project** | **Existing File** or by right clicking in the Project tab workspace and selecting **Add to Project** | **Existing File,** you will see the **Add file to Project** dialogue as shown in Figure 27. Browse for the vhd file that was created earlier from Quartus II and select it. When you select **OK**, the file(s) is added to the Project tab.

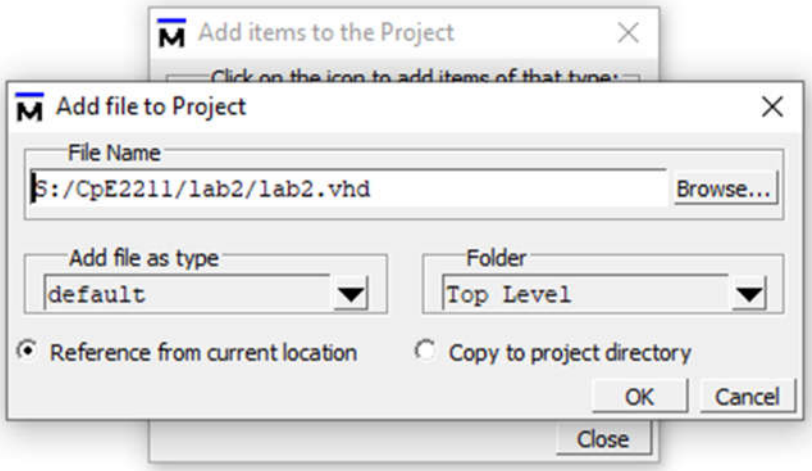


Figure 27: Add VHD file to Project.

Compiling the Files

The question mark in the **Status** column in the Project tab denote either the files haven't been compiled into the project or the source has changed since the last compile. To compile the files, select **Compile | Compile All** or right click in the Project tab workspace and select **Compile | Compile All** (Figure 28). You can also just select **Compile Selected.**

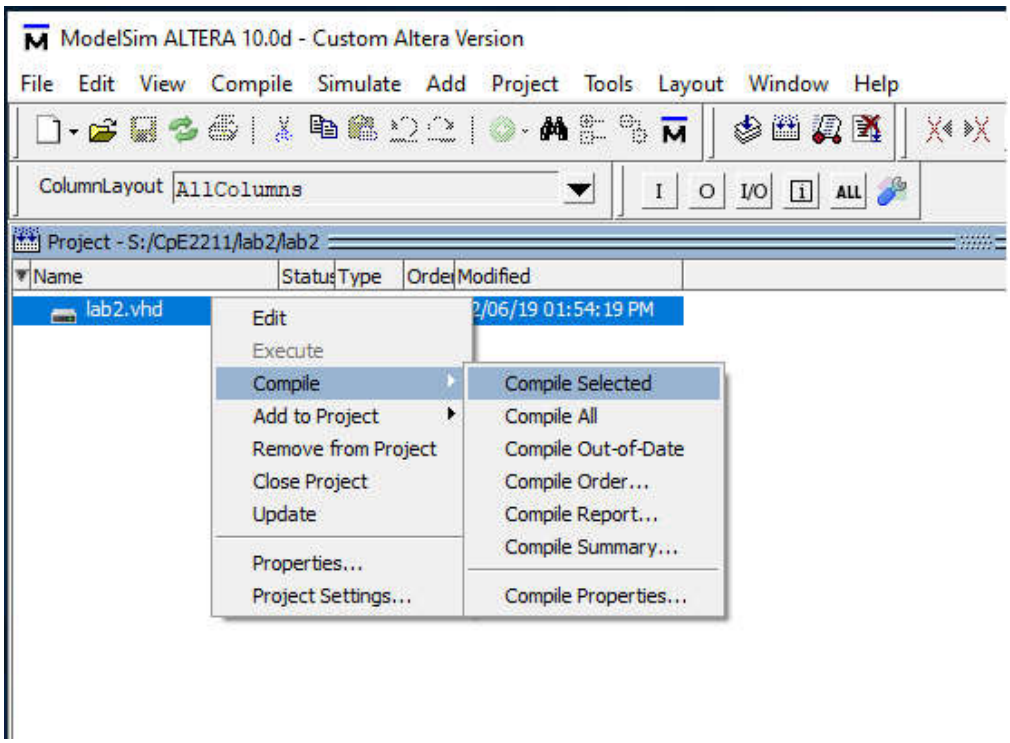


Figure 28: Compiling the VHD file.

Simulating a Design

A simulation can be started by selecting **Simulate | Start Simulation** (Figure 29) from the menu to open the **Start Simulation** dialog (Figure 30). In the **Start Simulation** dialog expand library *work* by clicking the "+", and you will see the compiled design units (lab2 in this tutorial).

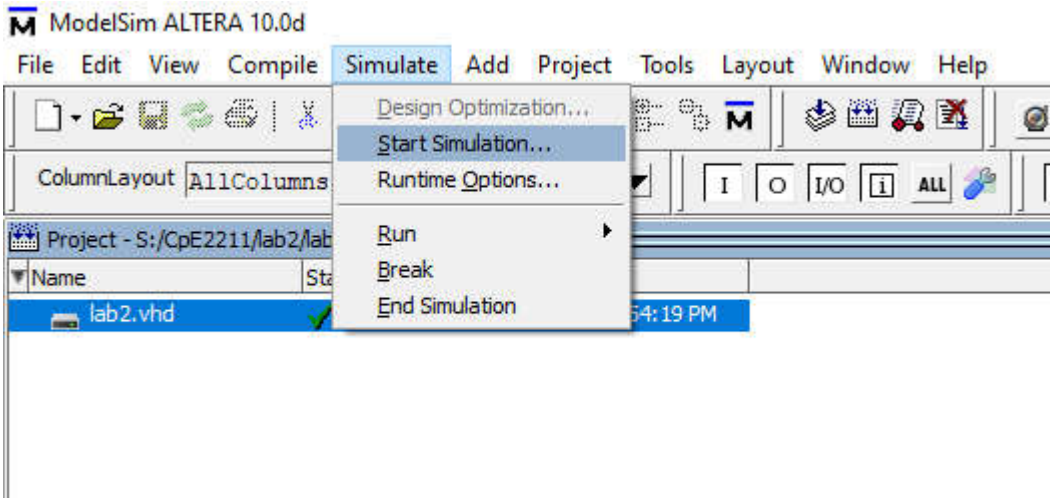


Figure 29: Starting a simulation from menu.

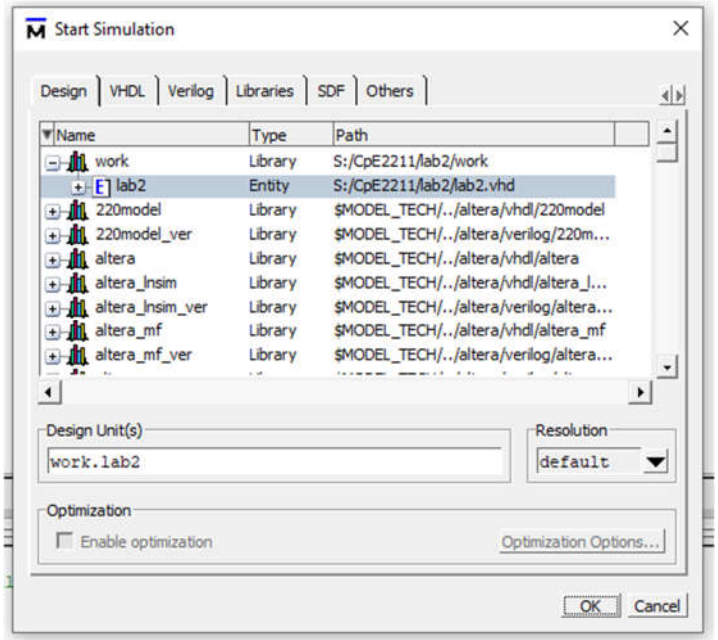
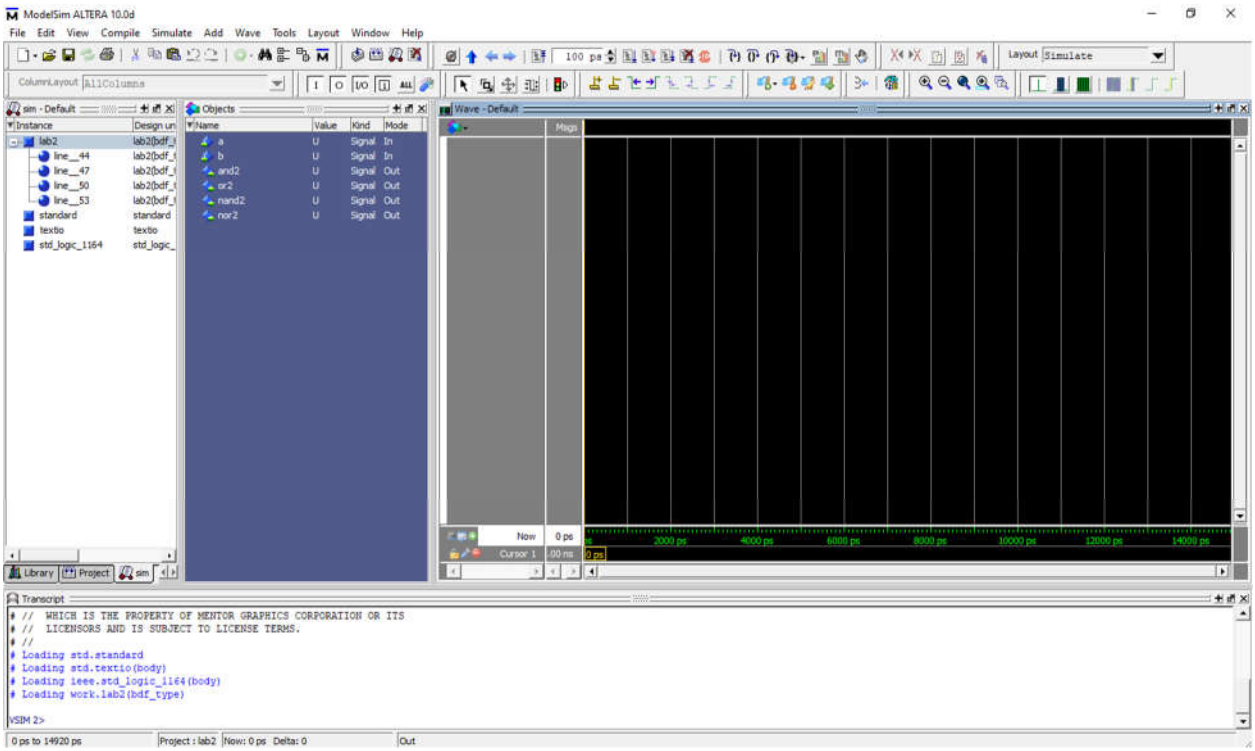


Figure 30: Selecting file for simulation.

A new tab named *sim* appears that shows the structure of the active simulation. Depending on the computer you are working on, the workspaces that open upon starting a simulation may vary. The workspace that are necessary are; **Project/Library/Sim, Objects, Wave, and Transcript**, as shown in Figure 31. In Figure 31, the left most workspace is the **Project/Library/Sim** workspace and at the bottom of it are tabs that let you select which to view.

The blue **Objects** workspace is next to the right, and the **Wave** workspace is to the far right. The **Transcript** workspace is at the bottom.



Figuer 31: Simulation workspaces.

If a workspace does not appear upon starting a simulation, it can be turned on by selecting **View | workspace of choice**, as demonstrated in Figure 32.

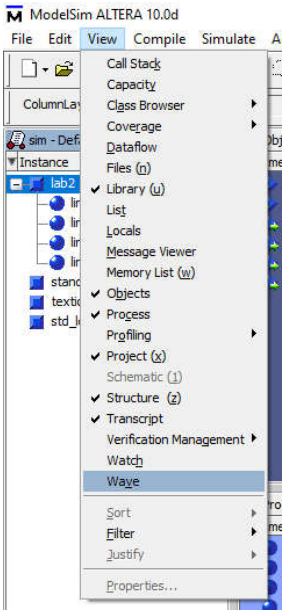


Figure 32: Selecting the Wave workspace to view.

At this point you are ready to run the simulation and analyze your results. You often do this by adding signals to the **Wave** window and running the simulation for a given period of time. View the waveforms of the selected signals by selecting, in the **Sim** window, right click on lab2, **Add | To Wave | All items in region** (as shown in Figure 33). Signals can also be added individually or in select groups from the Objects window by holding down the **Ctrl** key and selecting the signals of interest then right clicking over one and selecting **Add | To Wave | Selected Signals** (as shown in Figure 34).

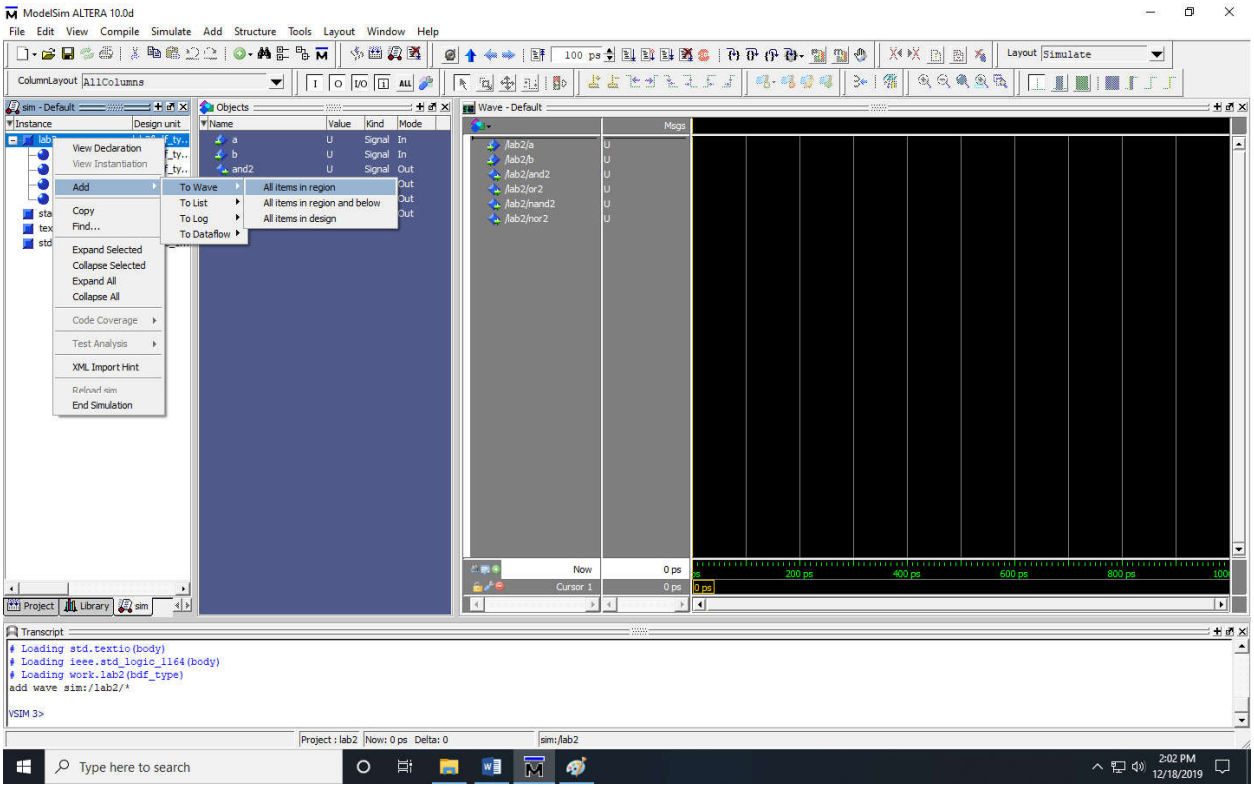


Figure 33: Adding signals to Wave window from Sim window.

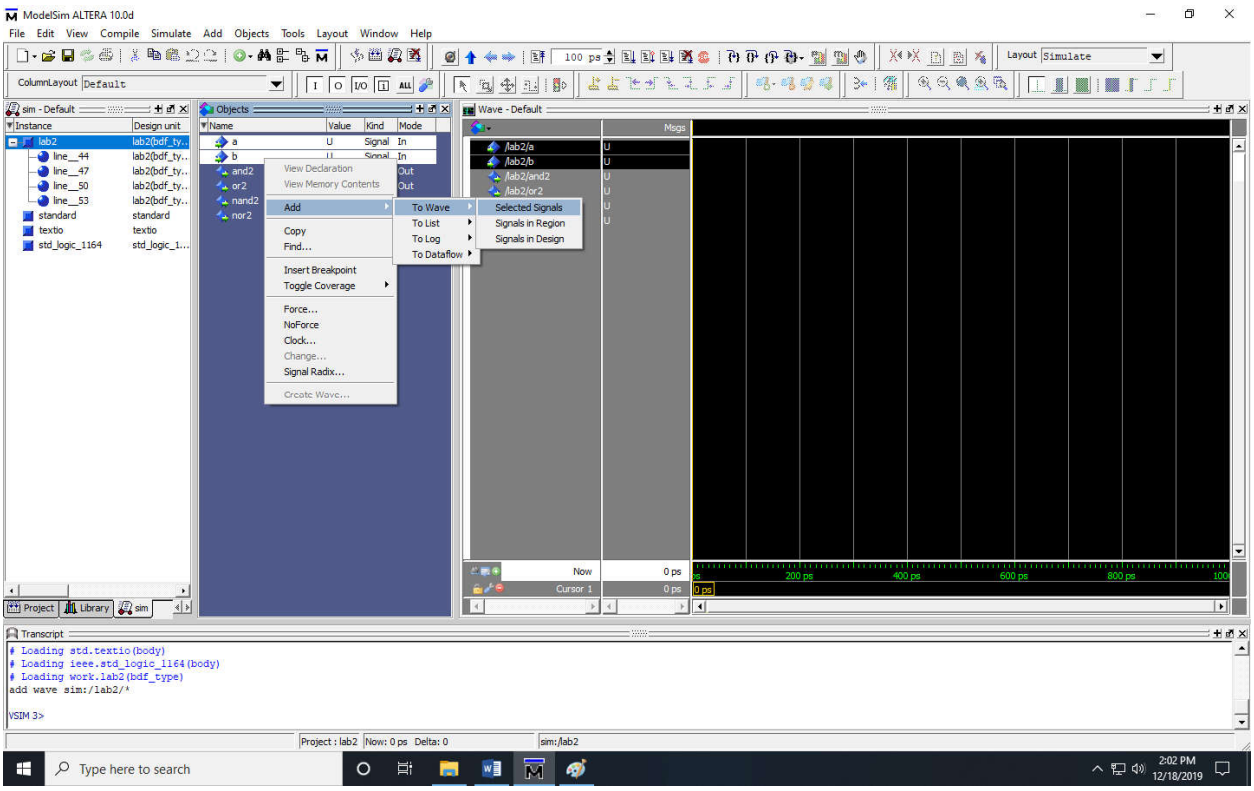


Figure 34: Adding signals to Wave window from Objects window.

The Wave window looks like a graph, with the signals on the vertical axis and the time along the horizontal axis as shown in Figure 35.

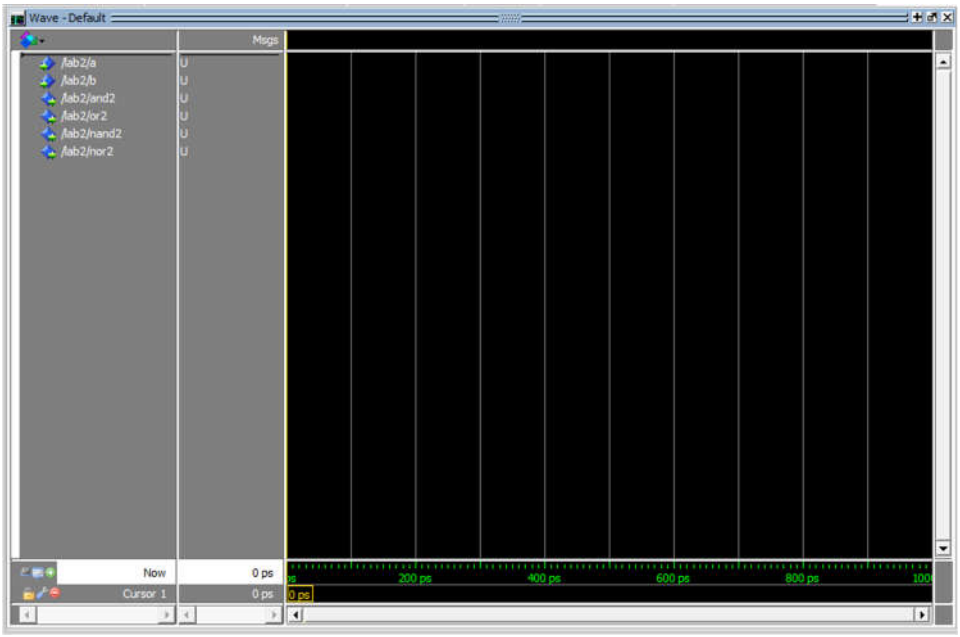


Figure 35: Wave window with signals along left edge, and time along horizontal axis.

To simulate the design we need to set the input to logic high or logic low at different times, and this can be done by typing commands to the **Transcript** window (Figure 36) or by right-clicking the input name and choosing the option Force (Figure 37).

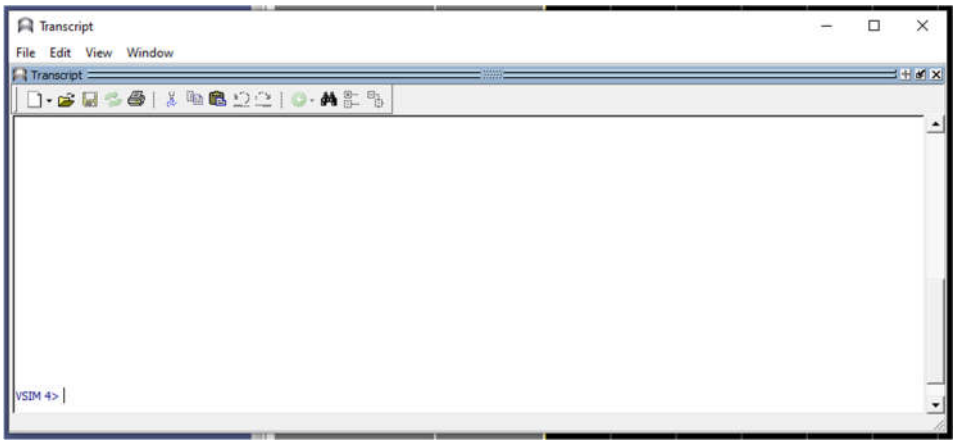


Figure 36: Transcript window.

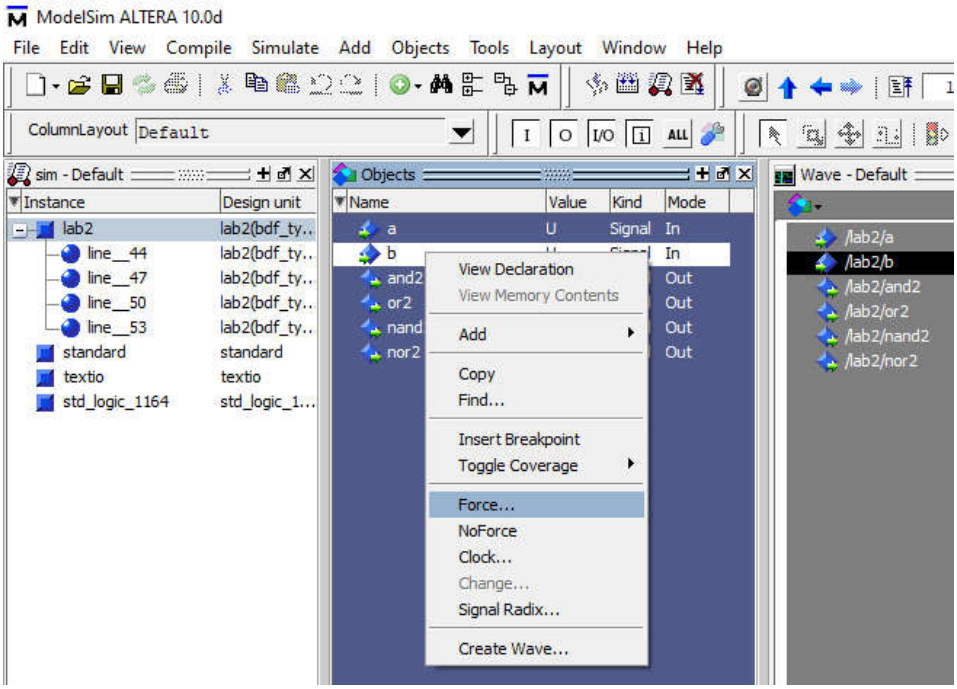


Figure 37: Force option for setting signal to a value.

To use the graphical method right click on the variable of interest in either the **Objects** or **Wave** window, then select **Force**. This will show you a dialog similar to Figure 38. In this window you can specify the **Value** of the input to be 0 or 1, and using the **Delay** value to set the time at which the value is set for that input. The **Value** defaults to U, for unknown.

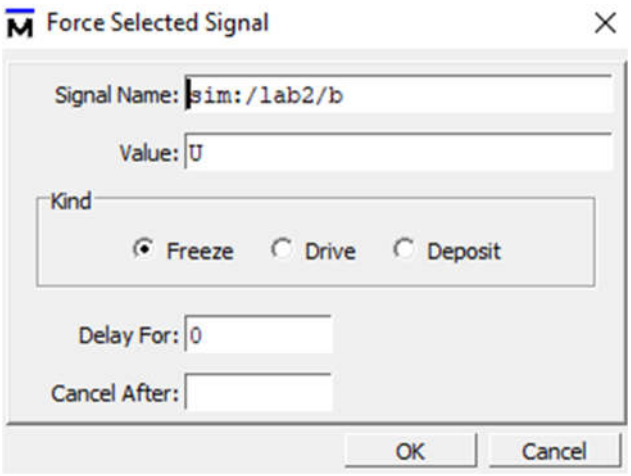


Figure 38: Forcing a signal to a value, and setting a Delay time for when to change value.

Set **a** to 0 at **Delay** times 0 and 100
Set **a** to 1 at **Delay** times 200, 300 and 400
Set **b** to 0 at **Delay** time 0
Set **b** to 1 at **Delay** times 100 and 200
Set **b** to 0 at **Delay** times 300 and 400

You can do this by right clicking **a** or **b** then selecting **Force** and setting the values to 0 or 1 and setting the delay as requested. This can also be done in the Transcript window by entering the following sequence for **a**:

force -freeze sim:/lab02/a 0 0
force -freeze sim:/lab02/a 0 100
force -freeze sim:/lab02/a 1 200
force -freeze sim:/lab02/a 1 300
force -freeze sim:/lab02/a 1 400

force -freeze sim:/lab02/b 0 0
force -freeze sim:/lab02/b 0 300
force -freeze sim:/lab02/b 0 400
force -freeze sim:/lab02/b 1 100
force -freeze sim:/lab02/b 1 200

After setting the values and times for the input signals run the simulation and observe the output. Select **Simulate | Run | Run-All** (Figure 39), or by typing **run -all** in the **Transcript** window.

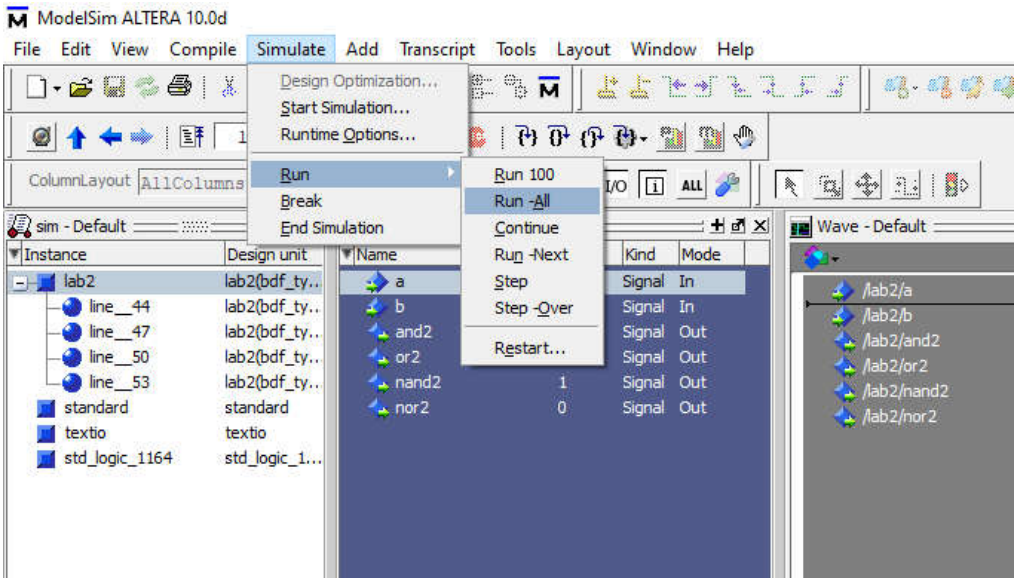


Figure 39: Running simulation from menu.

You should be able to see the following waveform in Figure 40. The yellow line dragged by the mouse tells the values of both input and output. Type **quit –sim will** exit the simulation.

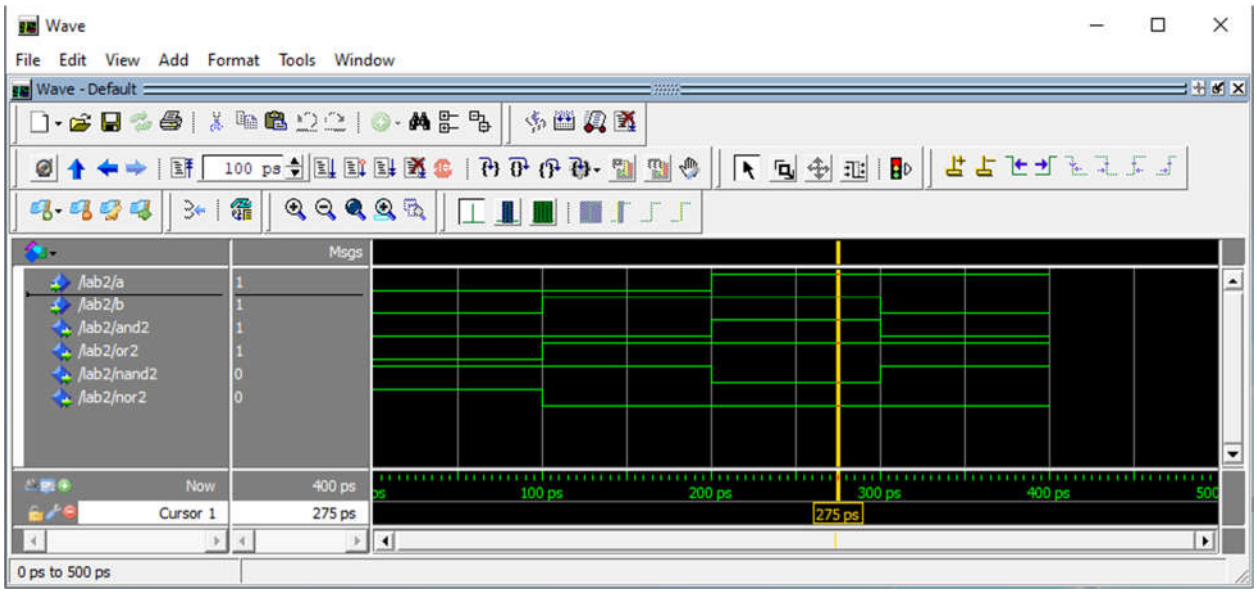


Figure 40: Simulation results.

Congratulations! If you've finished this tutorial you are well on your way to becoming proficient with one of the leading state of the art, industrial strength EDA tools.