UNIVERSITY OF CALIFORNIA, DAVIS
Department of Electrical and Computer Engineering

EEC 18   **DIGITAL SYSTEMS I**   **Fall 2021**

**LAB 1: Introduction to Quartus Schematic Capture, ModelSim Simulation and
the Intel/Altera DE10-Lite Board**

Overview

This lab tutorial introduces a few of the tools you will use in EEC 18 including the Quartus Prime design software, the Intel DE10-Lite board, and the ModelSim-Intel simulation software. You will use the Quartus schematic capture tool for design entry and ModelSim to simulate your design. You will verify your design on the Intel DE10-Lite board.

Background

Skim the following references which give important overviews of the hardware and software used in lab.

- Various Intel tutorials, for example, "Quartus Introduction Using Schematic Designs"
  https://www.altera.com/support/training/university/materials-tutorials.html#ifup-tut-dl

- DE-10-Lite User Manual
  Version 1.6, Release Date 10/11/2018
  http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=218&No=1021&PartNo=4

Installing Quartus, ModelSim, and SystemBuilder (optional but highly recommended)

To allow you to work on the CAD component of labs outside of open lab hours, it is highly recommended that you install these tools on your own computer—unfortunately they are available for Windows only.

- ModelSim-Intel simulator
  See handout *EEC_018_Software_Download_Guide.pdf*

- A free version of Quartus Prime *Lite Edition*
  Downloaded with ModelSim as described in *EEC_018_Software_Download_Guide.pdf*

- System Builder
  DE10-Lite CD-ROM, Version 2.0.3, Release Date 10/11/2018
  http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=218&No=1021&PartNo=4

I. Schematic Capture Using Quartus Prime

The ECE Department has Intel's Quartus Prime installed on the Windows workstations in 2110. Log in and create a directory for your projects. For example, you can create a folder **eec18/** in your My Documents folder. Then add a folder **lab1/** under **eec18/** for this lab.

1. Start Quartus.

2. Click on **File > New Project Wizard** to start the project wizard. Click **Next** once you have read the Introduction page.

   - Browse to your working directory such as C:/Users/*name*/Documents/eec18/lab1/

   - Give the project and the top-level design entity the same name, such as lab1. Click **Next** to move to the next page.

- For this tutorial, choose **Empty project** as the Project Type and then click **Next** to move to the next page.

- You don't need to Add Files to this project, so click **Next** to move to the next page.

- Under the Device family section, select **MAX 10 (DA/DF/DC/SA/SC)** as the Family. Under the Target device section, select **Specific device selected in 'Available devices' list** and choose the device **MAX10 10M50DAF484C7G** since this is the FPGA on the DE10-LITE board. The screen should look like Figure 1. Click **Next** to move to the next page.
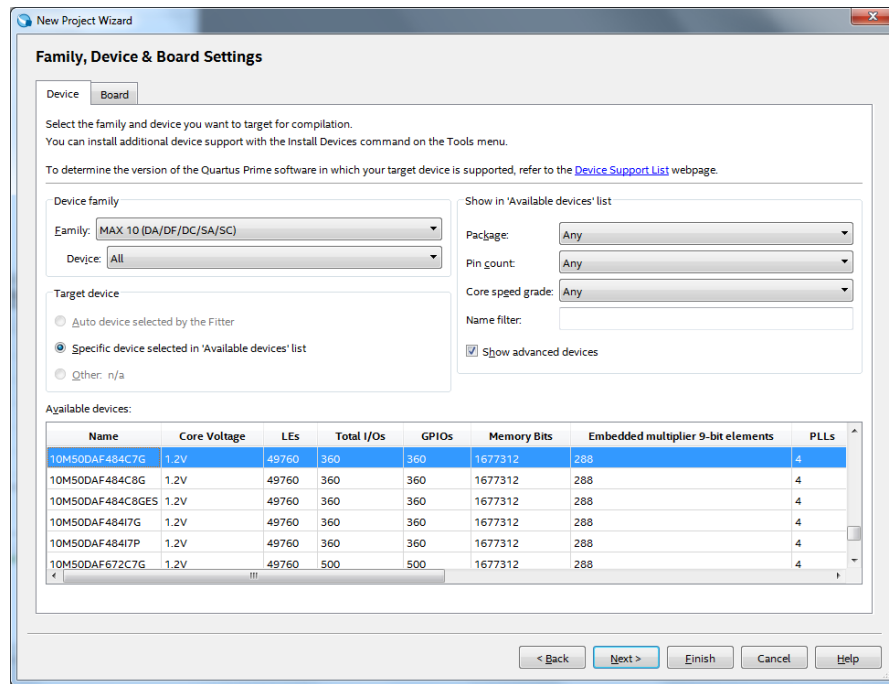

Figure 1 Choosing the Device Family and Device

- For Simulation, select the tool **ModelSim-Altera** from the drop-down box and specify the format as **Verilog**, as shown in Figure 2. Quartus will generate a Verilog netlist from your schematic, allowing you to easily simulate your design. Click **Next**, review the Summary, then click **Finish.**
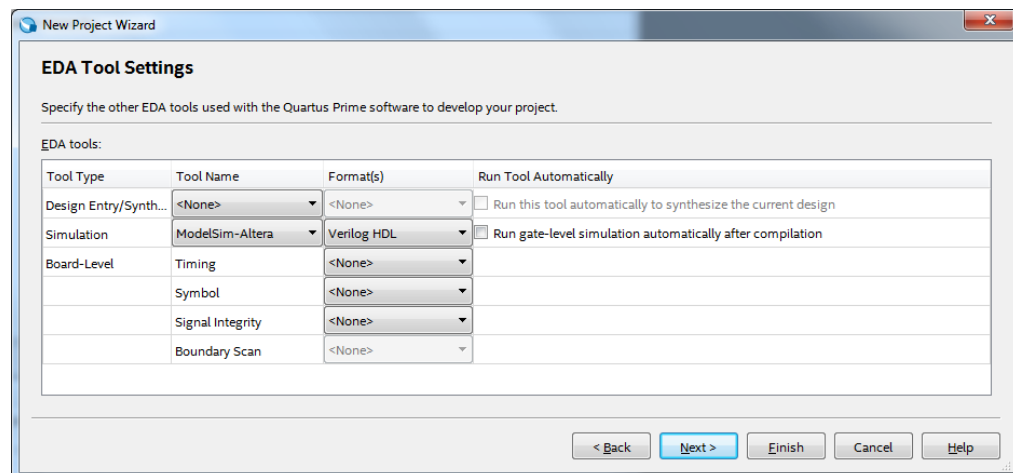

Figure 2. Choosing the ModelSim-Altera simulator

3. Open a new schematic page:

- Click **File > New** and select **Block Diagram/Schematic File** and click **OK**.

- Click **File > Save As…** and specify a file name such as lab1. Make sure the box **Add file to current project** is checked and click **Save**.

4. Start placing components on your blank schematic page:

    - Double-click on the blank schematic page to bring up the Symbol dialog window. Double-click on the libraries to view the various components available.

    - Enter **XOR** as the name and click **OK**. Place the component in the upper middle portion of your schematic.

    - Using the same procedure, add another **XOR**, three **AND2** gates, one **OR3** gate, three **INPUT** components, and two **OUTPUT** components to your schematic. You can also use Copy (Ctrl-C) and Paste (Ctrl-V) to add additional same components.

    - As Figure 3 shows, connect the inputs and outputs of the gates together. Connection can be made by selecting the Orthogonal Node Tool from the menu bar. Click, hold, and drag from one of the endpoints to the other end point to make a connection. **Tip:** Moving the mouse pointer over the tool icons will display their respective names. The Orthogonal Node Tool icon looks like this: ⌐.

    - Note: another way to make signal connections is to simply move the arrow cursor (selection tool) to hover over the end of a pin. The arrow will change to a '+' and show the Orthogonal Node Tool icon. (If this doesn't happen, make sure nothing is currently selected by clicking on empty space in the schematic and then moving the arrow over the pin end.) Once you have the '+' cursor, click, hold and drag the mouse button to the other endpoint in order to make a connection. The advantage of this method is that you can automatically go between Selection mode and Orthogonal Node Tool mode without manually switching back and forth.

    - Connect the output components. Once the output component is on the schematic, select and drag it until the connection point just touches the output pin of the XOR and OR3 gates and then release the mouse button. Select the output component again and move it to the right. You will notice that a wire will appear between the two points, maintaining the connection. (This is known as "rubberbanding" and assumes that "rubberbanding" or ⧗ is enabled, which should be default. This is one of the icons on the menu bar.)

    - As shown in Figure 3, name the input components as SW[0], SW[1], SW[2], and name the output components as LEDR[0] and LEDR[1]. You can double-click on the pin to bring up the Pin Properties dialog box or just click on the pin name, right-click and select Properties.

    At this point, your schematic should look like the one shown below in Figure 3.
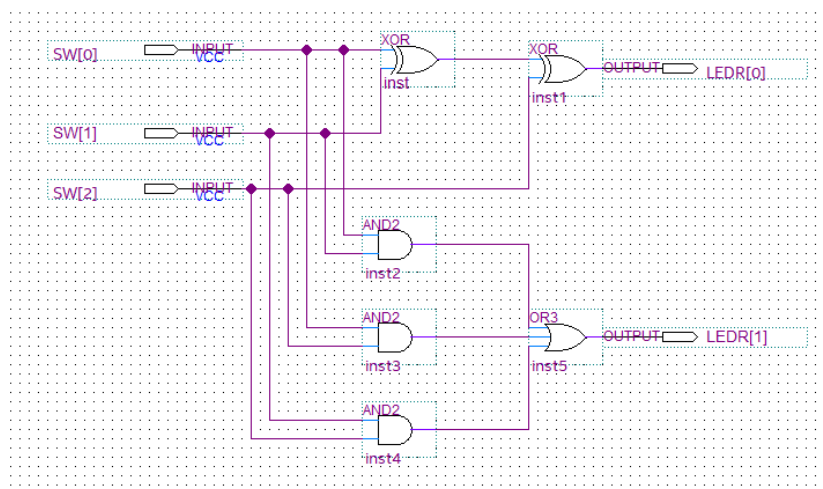

Figure 3. Schematic of a Full Adder.

5. Make pin assignments for the DE10-Lite board.

   - There are several ways to make pin assignments. One way is to select **Assignments > Assignment Editor** and manually enter the pin assignment for each input and output node. An easier way is to use the DE10-Lite System Builder tool to generate a .qsf file containing the pin assignments. However, to use System Builder to create pin assignments, you must name the input and output nodes using the default names used by System Builder. (These are the names used in Figure 3). These names are specified in the DE10-Lite User's Manual as well as in the System Builder output files.

   - Run the System Builder program. Specify the project name, such as lab1, and the I/O devices that you will use in your project. An example screenshot is shown in Figure 4. There is no problem if you select I/O devices that are not used in your schematic. Those pin assignments will just be ignored. For example, in Figure 4, the CLOCK, LEDx10, Buttonx2, 7-Segmentx6 and Switchx10 are all selected, even though they might not all be needed in this design.
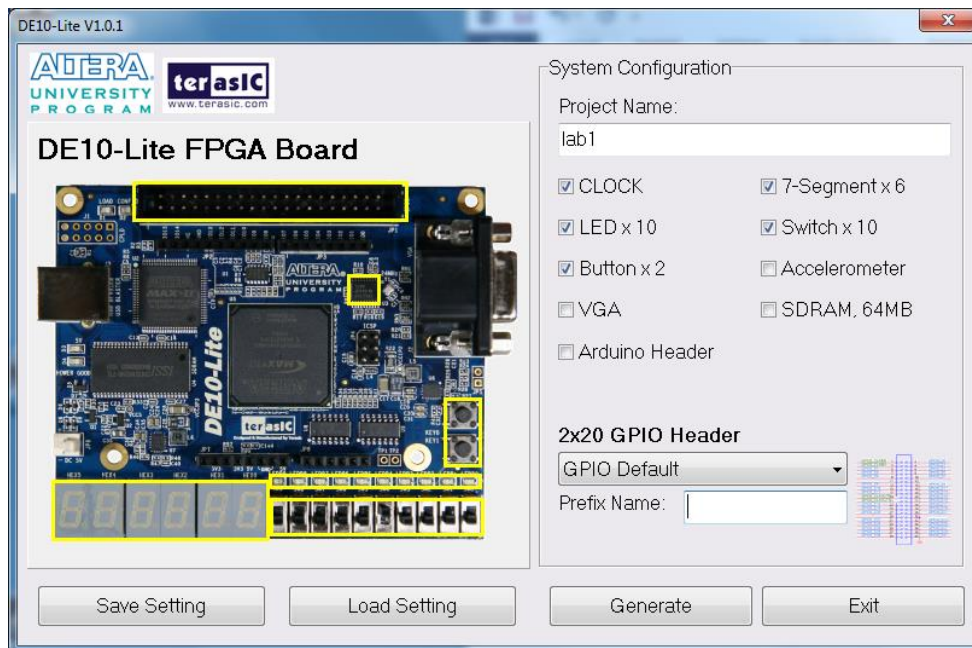

Figure 4. Generate Pin Assignments Using System Builder Program

   - Click **Generate** and navigate to your Quartus project directory. <u>Note that System Builder will **not** use your Quartus project directory by default.</u> In a location that is common with your projects, right click and select New > Folder and name the folder systembuilder. Double-click on this folder and save the System Builder project in this directory. Ensure that the 'Project Name' is set to 'eec18_imports', since this file will be used throughout the quarter. Make sure that the name of the saved file is eec18_imports.qpf. **You do not want to overwrite the project file, lab1.qpf, which you created earlier.** You can check the systembuilder\eec18_imports.v file to see the default names used by System Builder for the I/O ports.

   - In Quartus, click on **Assignments > Import Assignments** and select the file systembuilder\eec18_imports.qsf. This will import the pin assignments generated by System Builder into your project.

6. Compile your design by clicking on the **Start Compilation** icon, ▶, on the menu toolbar. (You can also select **Processing > Start Compilation**.) Your design should compile without errors. You can ignore any warnings.

Note: If you get an **Error 275062**, it means two components have the same instance name. You can right-click one of the components and select **Properties** and give it a unique instance name.

7. Click on **Assignments > Assignment Editor** to verify that the proper pin assignments have been made. There should be a green check next to each port that has been assigned, shown by the word Ok in Figure 5. The FPGA pin number in the Assignment Editor should match the location specified in Tables 3-6 and 3-3 in the DE10_Lite_User_Manual, as shown in Figures 6 and 7.

Ok	LEDR0[0]	Location	PIN_A8
Ok	LEDR1[1]	Location	PIN_A9
Ok	SW[0]	Location	PIN_C10
Ok	SW[1]	Location	PIN_C11
Ok	SW[2]	Location	PIN_D12

Figure 5. Verify Pin Assignments in the Assignment Editor

Table 3-6 Pin Assignment of 7-segment Displays

Signal Name	FPGA Pin No.	Description	I/O Standard
HEX00	PIN_C14	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX01	PIN_E15	Seven Segment Digit 0[1]	3.3-V LVTTL
HEX02	PIN_C15	Seven Segment Digit 0[2]	3.3-V LVTTL
HEX03	PIN_C16	Seven Segment Digit 0[3]	3.3-V LVTTL
HEX04	PIN_E16	Seven Segment Digit 0[4]	3.3-V LVTTL
HEX05	PIN_D17	Seven Segment Digit 0[5]	3.3-V LVTTL
HEX06	PIN_C17	Seven Segment Digit 0[6]	3.3-V LVTTL

Figure 6. Pin Assignments for DE10-Lite board (see p. 28 in User Manual)

Table 3-3 Pin Assignment of Push-buttons

Signal Name	FPGA Pin No.	Description	I/O Standard
KEY0	PIN_B8	Push-button[0]	3.3 V SCHMITT TRIGGER"
KEY1	PIN_A7	Push-button[1]	3.3 V SCHMITT TRIGGER"

Figure 7. Pin Assignments for DE10-Lite board (see p. 25 in User Manual)

8. Plug the USB cable into the USB Blaster Port on the DE10-LITE board and connect the other end of the USB cable to your PC. The USB cable will supply power to your DE10-Lite board. The USB-Blaster driver should be installed on your system to allow communication between Quartus and the DE10-LITE board.

9. Click on the **Programmer** icon, 🖑, on the menu tool bar. Make sure the Hardware Setup is specified as USB-Blaster [USB-0]. Click the Hardware Setup button, select the Hardware Settings Tab, and verify that USB-Blaster [USB-0] is selected in the Currently Selected Hardware field. If USB-Blaster is unavailable, that means Windows did not recognize the device. Make sure your DE10-Lite Board is connected to the PC. If the board is plugged in to the PC and Windows does not recognize it, you will need to install the Blaster driver before proceeding. To install the Blaster device driver manually, follow the Blaster Driver Installation Procedure in the Software Installation Guide handout. After the Blaster Driver is installed, continue with the following instructions.

   - Click on the gray chip box labeled 10M50DAF484, then select **Delete** to remove the old .sof file

- Click the **Add File…** button and select the output file from the output_files folder in your project folder. The correct output file will have the .sof extension (i.e. lab1.sof). Your screen should look like Figure 8.
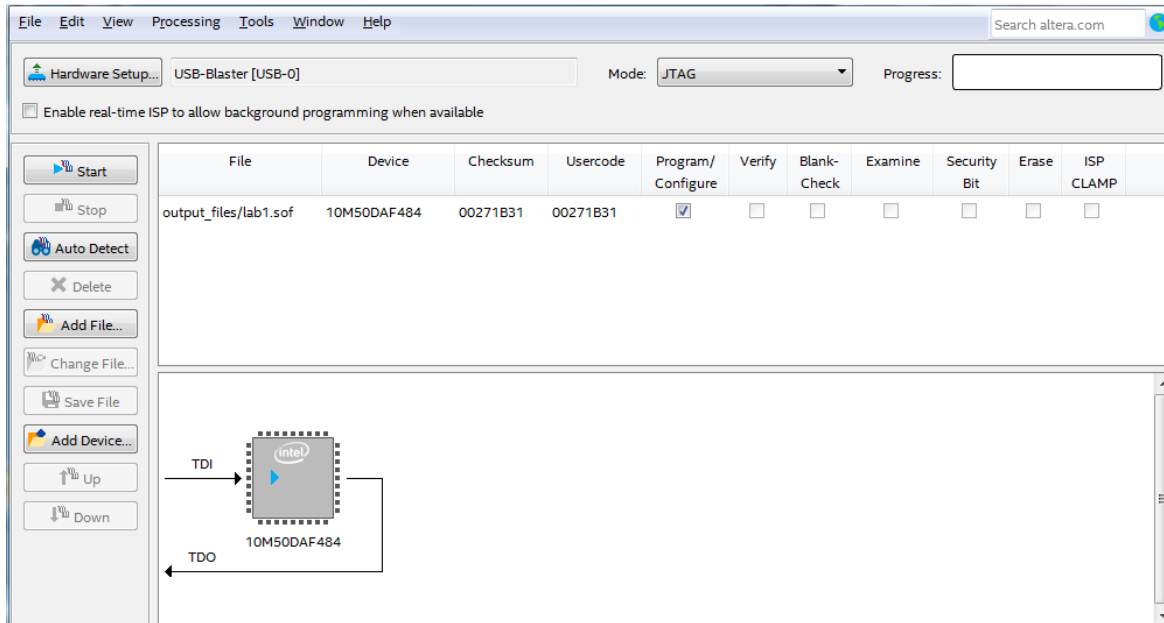

Figure 8. Device Programmer Configuration

- Click **Start** to load the program into the DE10-LITE board. The rightmost two LEDs (LEDR[0] and LEDR[1]) should be **not** light up on the DE10-LITE board.

10. Push the rightmost three switches (from right to left, they are SW[0], SW[1], and SW[2]) on the DE10-LITE board to test your full adder circuit. The switches have value of '1' if they are pushed up, and value of '0' if not pushed. LEDR light up if the value is '1' and off if the value is '0'. Your full adder circuit should be able to perform the operation:

$$\{LEDR[1], LEDR[0]\} = SW[0] + SW[1] + SW[2]$$

Where LEDR[1] is the carry out bit, LEDR[0] is the sum bit, SW[2..0] are the three bits to be added.

For example, if all three switches are not pushed (i.e. all 0s), both LEDRs should be off (since 0+0+0 = 0); if you push all three switches (i.e., all 1s), both LEDRs should be on (since 1+1+1 = 3 (11 in binary format)).

11. Demonstrate your circuit to your TA and have him or her sign your cover sheet.

II. MODELSIM SIMULATION

Once your schematic has been drawn in Quartus, it is important to verify the functionality of the design. For small scale designs, it might be simpler to directly implement the design rather than simulate its functionality. However, in real-world designs, logic circuits often encompass tens of millions of logic gates making the design impossible to cost effectively implement and test.

For this section, reuse the full adder from Part I, Figure 3. Since we specified ModelSim-Altera as the simulation tool when we created the project, the project should be configured to run ModelSim-Altera.

1. From the Quartus pull-down menu, select Tools > Options.

2. Under General > EDA Tool Options, define the file path to ModelSim-Altera. (By default, it is 'C:\intelFPGA_lite\19.1\modelsim_ase\win32aloem')

6

3. From the Quartus pull-down menu, select Tools > Run Simulation Tool > Gate Level Simulation. Close any previous ModelSim windows.
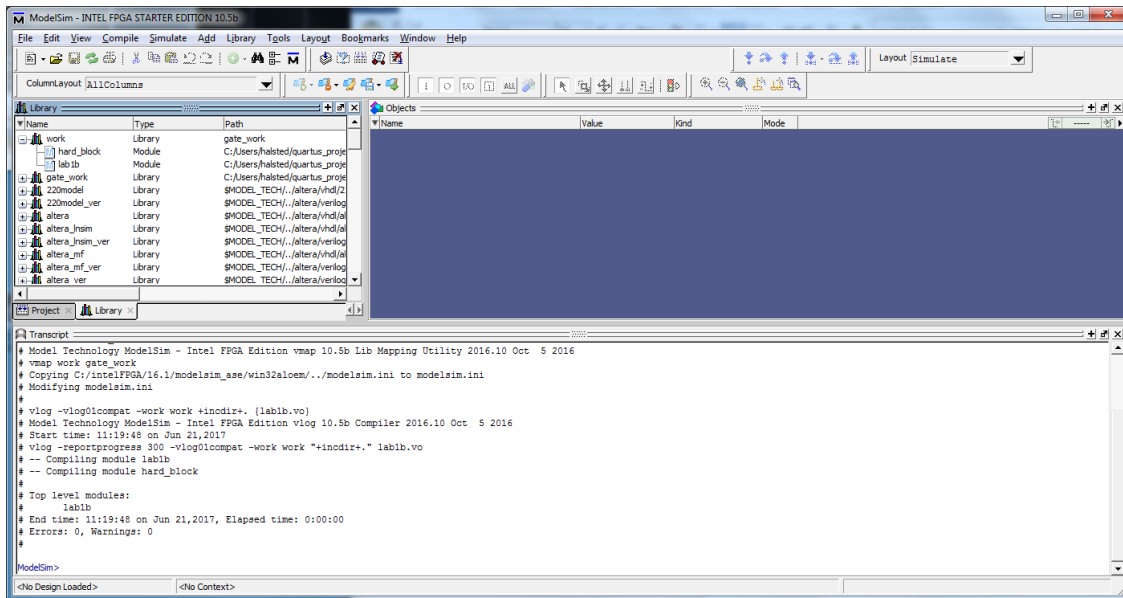

Figure 10. ModelSim-Altera

4. ModelSim will open and your design will be automatically transferred to ModelSim. Verify this by looking at **"work"** folder in the Library list window. If the Library window is not open, select **View > Library** to open it. Click on the + to expand the work library. You should see your lab1 project name, which is a Verilog netlist output file, in the **work** Library, as shown in Figure 10.

5. To simulate your design, select **Simulate > Start Simulation…** from the toolbar menu. Click on the **Design** tab and select your design file in the work library. In Figure 11, the Verilog netlist file, lab1, has been selected.
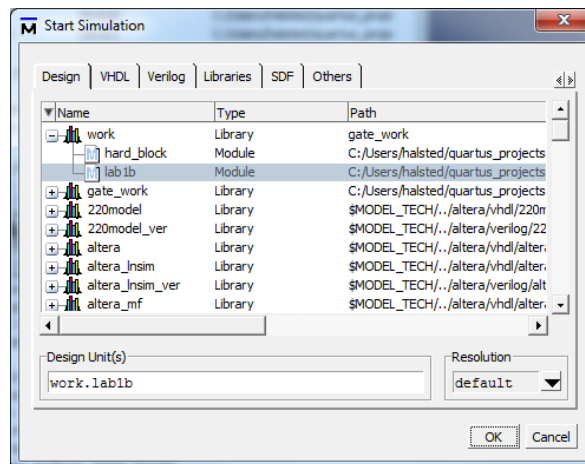

Figure 11. Selecting your Verilog netlist output file to simulate

6. Next, click the **Libraries** tab in the *same* Start Simulation dialog box. Click the **Add…** button to the left of the Search Libraries (-L) pane. A Select Library dialog box will pop up. Select the down arrow ▼ and scroll down through the list of libraries. Select **altera_ver** and **fiftyfivenm_ver**, as shown in Fig. 12.
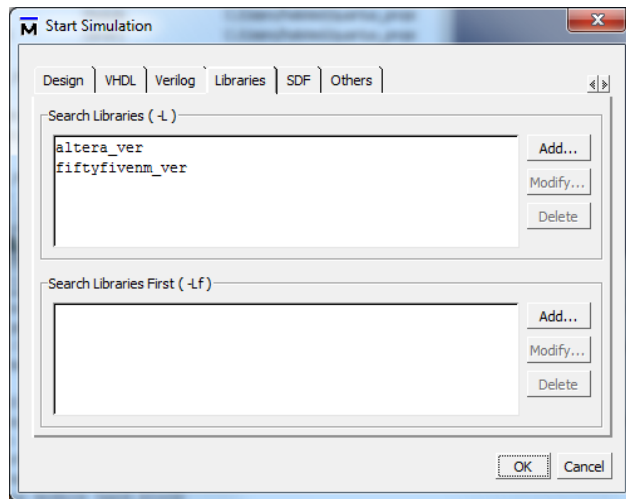
Figure 12. Selecting the libraries

7. Click **OK** to close the dialog box. You can ignore the warnings. Your screen will update and show you the "sim-Default" and the "Objects" windows.

8. You may or may not have the waveform viewer open. If not, click on **View** and select **Wave**. This will display the waveform viewer window.

9. You are now ready to simulate and view the values of the signals in your design. Let's add some signals we want to inspect into the Wave window. With your top-level design file highlighted in the sim-Default window, drag, and drop the **SW** and **LEDR** signals from the Objects window into the Wave window. (Another method is to select **SW** and **LEDR** in the Objects window, right-click and select **Add Wave**).

10. For the simulation to function, you need to set up the input stimulus on the input pins of your design. Right-click on the **SW** signal and select **Force...** The Force Selected Signal window will pop up. This window is used to force signals in your design to logic 0 or logic 1. Enter the digit **000** (zero) in the Value field and select **Drive** in the Kind field. If **Drive** does not work, try **Freeze**. Press **OK** to save changes and close the window.
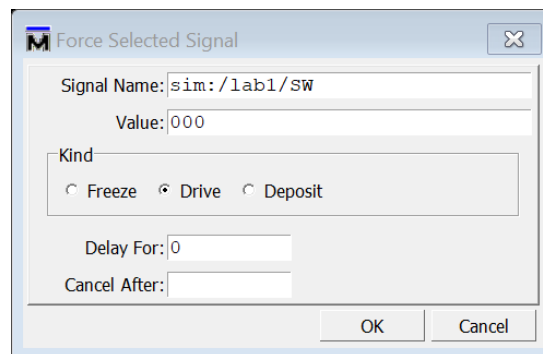

Figure 13. Forcing input values in Modelsim

You are now finished with setting up the input stimulus. Enter **20 ns** in the simulation time and click the **Run** icon, . You can also type in the command line **'run 20 ns'** *(without quotes)* and the simulation will run for 20 ns. Another option is to press the **F9** hotkey.
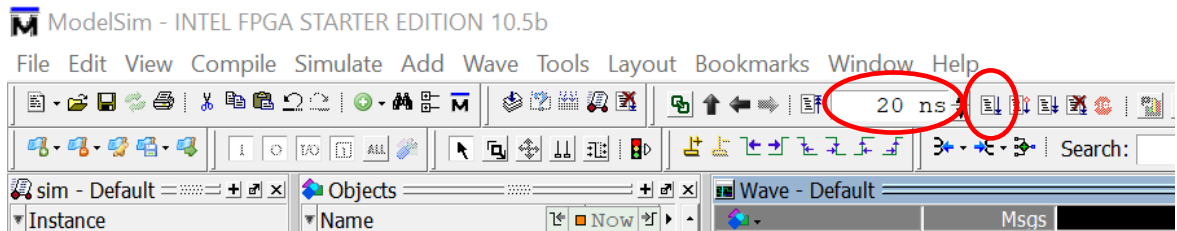
Figure 14. Set run time.

Repeat this step 7 times to forcing the values to: 001, 010, 011, 100, 101, 110, 111.

11. Select **View > Zoom > Zoom Full** to display the full 8 runs in the Wave window. It should look same as Figure 15.

Figure 15. ModelSim Wave Window

12. Does each waveform in the Modelsim simulation match the behavior of the corresponding signal when the circuit operates on the DE10-LITE board?

13. Demonstrate to your TA that you can simulate the circuit in ModelSim by reproducing Figure 15.