

# CPE100 Hands-On Assignment 1

---

Adapted From: Fall 2017, Section 1001 (Dr. Harris)

## 1. Introduction

In this portion of this assignment, you will use the Quartus II Web Edition v13.1 software, a powerful commercial suite that will enable you to design, simulate, and test digital circuits.

You can use this software either:

1. Installed the software on your own laptop or computer (see Homework 2), or
2. Using the PCs located in the Electrical and Computer Engineering Laboratories, located in TBE B-350 or TBE B-311. For this, you must have an ACE account.

Skim through all of these instructions and carefully read the **What to Turn In** section (Section 3) before beginning this portion of the homework.

You will complete the following steps to:

- Step 1.** Design a Boolean function
- Step 2.** Sketch a gate-level implementation of the function.
- Step 3.** Enter your design in Quartus II
- Step 4.** Simulate your design using ModelSim

Here are instructions again for installing the Quartus II software on your own PC if you didn't do so last time but would still like to. **Note: It will likely be easier to use the ECE labs than to install software yourself.**

### Step 0. Optionally install the Quartus II software

If you would like to install the Quartus II v13.1 software on your own laptop or computer, follow these instructions. Otherwise skip to Step 1.

Note that a newer version of the software exists, however, you will install version 13.1 to be consistent with the version in the ECE labs.

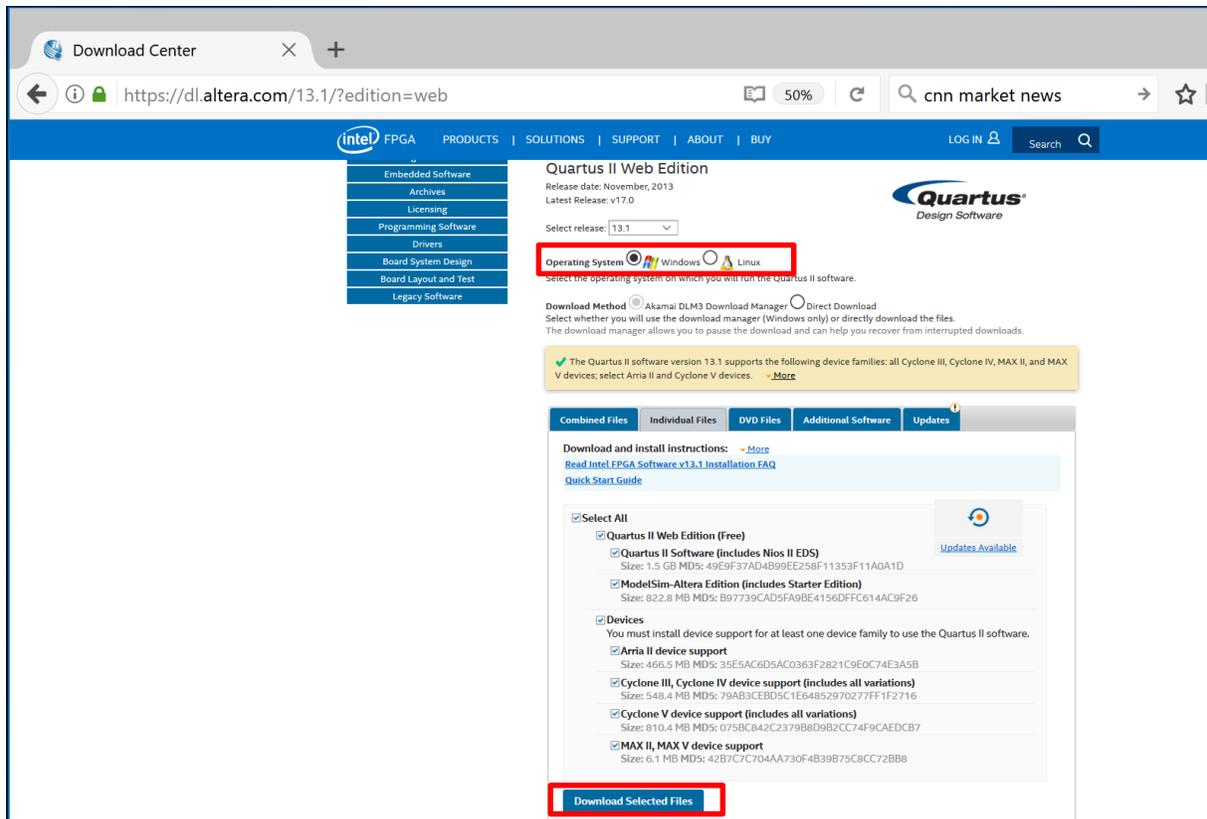
---

**[Fall 2019], the labs are transitioning to version 14.1.** You can find direct UNLV hosted links to the software on the lab equipment webpage at the bottom when you search for "Software Downloads".

- [http://eelabs.faculty.unlv.edu/index.html?navi=main\\_labequipment](http://eelabs.faculty.unlv.edu/index.html?navi=main_labequipment)

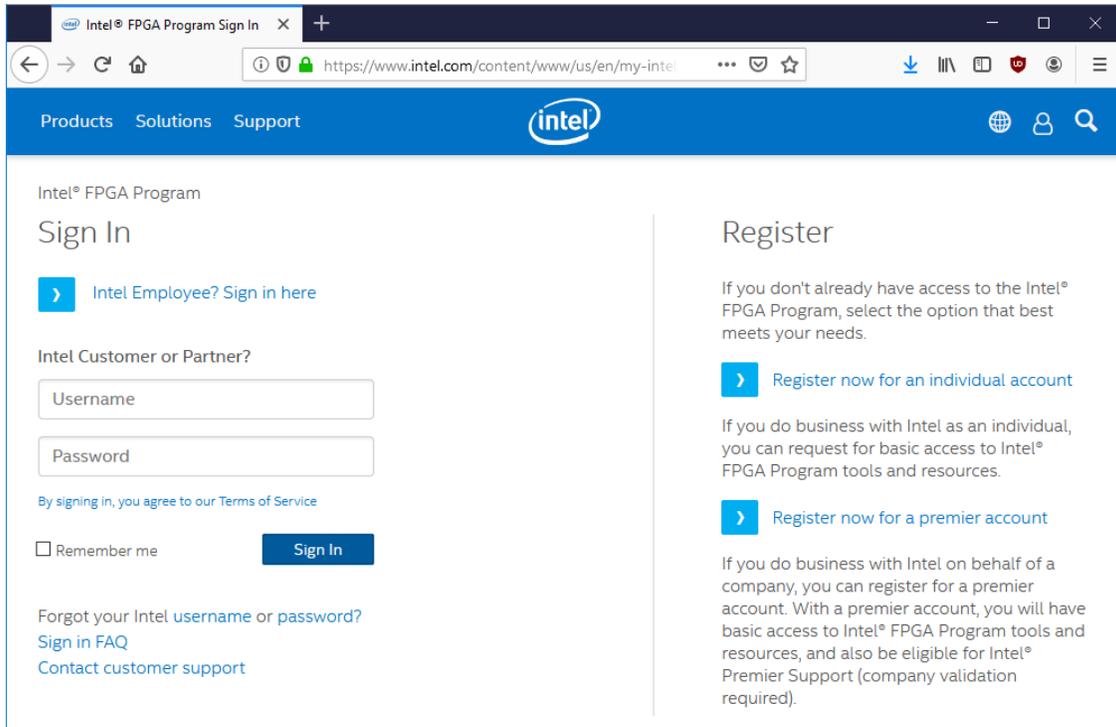
---

Browse to: <http://fpgasoftware.intel.com/13.1/?edition=web>. Select the operating system of choice (Windows or Linux) and leave the **Select All** box selected, as shown in Figure 1. Then click on **Download Selected Files**. You will have to register an Individual Account using UNLV as your business and email.



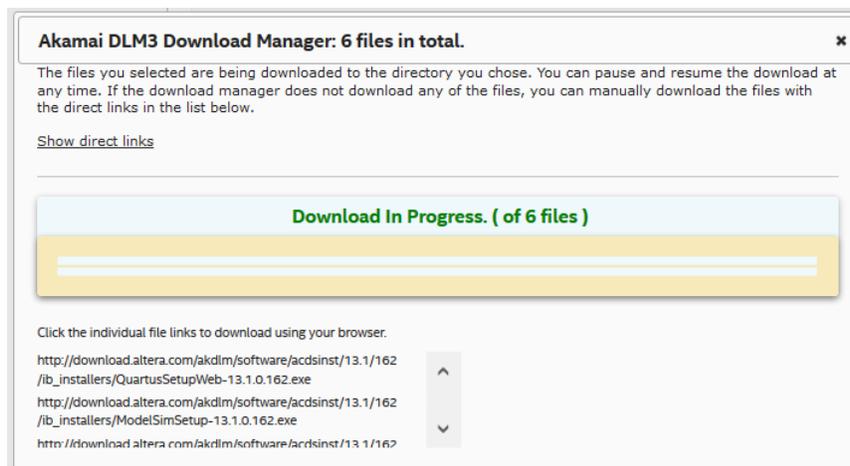
**Figure 1. Quartus II Web Edition v13.1 download window**

You will now be directed to the Intel FPGA Program login page. Select **Register now for an individual account**, as shown in Figure 2 to create a free account. Enter your personal information using your UNLV email address and use UNLV for your business (~3000 employees) and click on **Create Account**.



**Figure 2. Create an Intel FPGA Program account**

After you have successfully created an account and logged in, the download will begin, as shown in Figure 3. If for some reason the download does not begin, you can click on **Show direct links** to download the individual files manually. Depending on your internet connection speed, downloading the files can take a while – so if you decide to install the software on your own computer, **don't do it at the last minute! The full download size is ~10GB.**



**Figure 3. Downloading Quartus II installation files**

After completing the download of all of the files, click on the QuartusSetupWeb-13.1.0.162.exe file to install the software.

### Step 1. Design a Boolean function

Now you will design a digital circuit that implements the following function of three inputs, A, B, and C.

The output **Y** is TRUE if (and only if):

- A and B are TRUE or
- B and C are TRUE or
- A and B are TRUE and C is FALSE or
- A and B are FALSE and C is TRUE

Write a minimized Boolean equation for Y in terms of the inputs A, B, and C in SOP form. Additionally, complete a full truth table for the function Y.

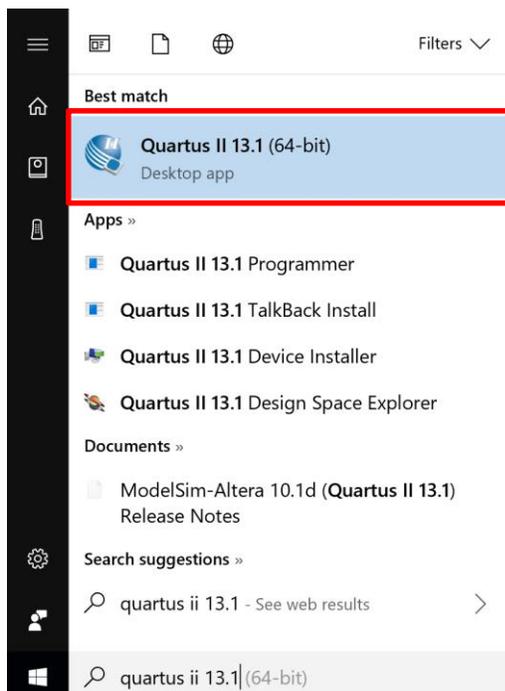
### Step 2. Sketch a gate-level implementation of the function.

You will now turn the expression (minimized SOP form) into a circuit consisting of AND, OR, and NOT gates. Clearly label the inputs and output on your schematic.

### Step 3. Enter your design in Quartus II

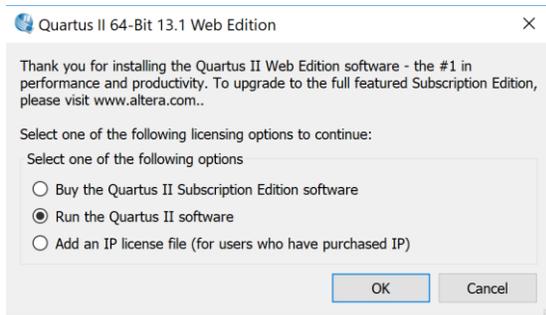
Now that you know how to produce the Y output using simple logic gates, you will enter your schematic into the Quartus II software and test it in simulation to see if it works the way you expected it to. The Quartus II software is a powerful and popular commercial suite of applications used by hardware designers.

First, you will learn how to start a new project. Start the Quartus II v13.1 software from the Start menu. Choose Quartus II (click on the Start button, then type in “Quartus II 13.1” in the search box to find the application), as shown in Figure 4.



**Figure 4. Start Quartus II 13.1**

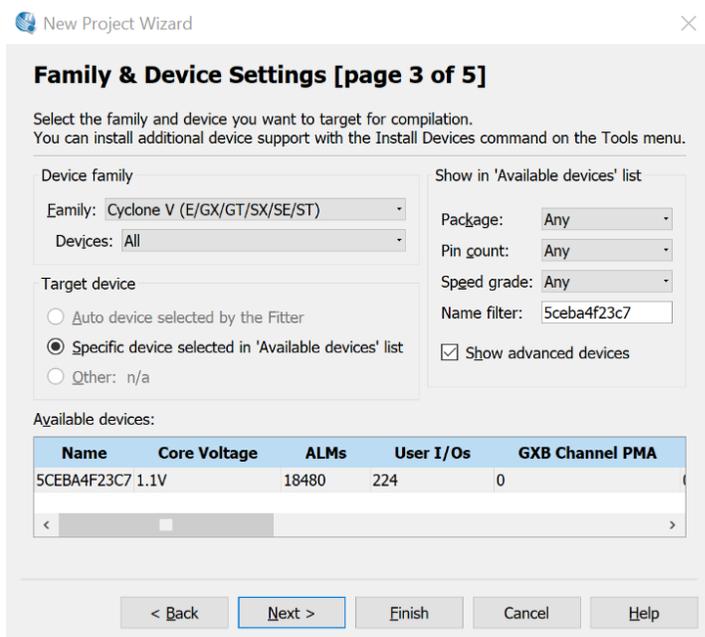
A window will pop up asking if you want to buy the software, simply click on **Run the Quartus II software**, as shown in Figure 5. Then click OK.



**Figure 5. Run the Quartus II software**

In the Quartus II Window, click on **File** → **New Project Wizard**. In the New Project Wizard, click Next. Then set the working directory to a convenient place (that you will be able to find again). Name the project **cpe100\_comp1\_xx**, where “xx” are your initials. Make sure there are no spaces or unusual characters in the path or file name; the tools may complain or silently misbehave if it has trouble with the file name.

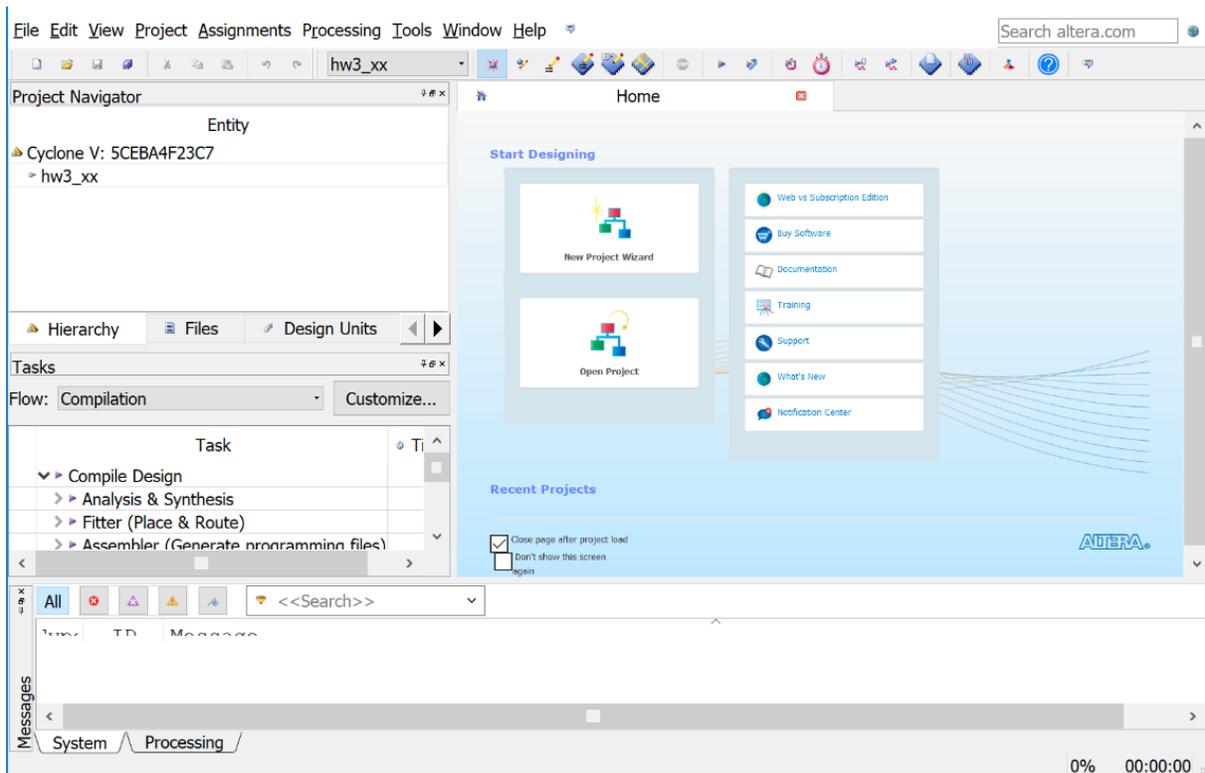
You are now on the **Family & Device Settings** page to select a chip. You will be using the Altera DE0-CV development board, which contains a Cyclone V 5CEBA4F23C7 FPGA. Set the family to **Cyclone V**, as shown in Figure 6. Scroll down and select the device (**5CEBA4F23C7**) from the list of Available Devices. (You can also type that value in the **Name filter** box to make it easier to find.) Highlight the device under Available devices, and click **Next**. Then click **Finish** to create your new project.



**Figure 6. Set FPGA target**

The Quartus II window will open in a moment. You may wish to maximize the window. You will see three main panes, as shown in Figure 5 (and can bring them up from the View → Utility Windows menu if you accidentally close one):

- **Project Navigator:** Lists the current project's sources file and the chip in use.
- **Tasks:** Lists the processes to perform on the source selected in the Sources pane. For example, we will use this pane later to download a circuit design onto the Cyclone V FPGA.
- **Messages:** Lists the output of current processes, errors, and warning at the bottom of the screen. Keep an eye on these messages; important warnings appear here.

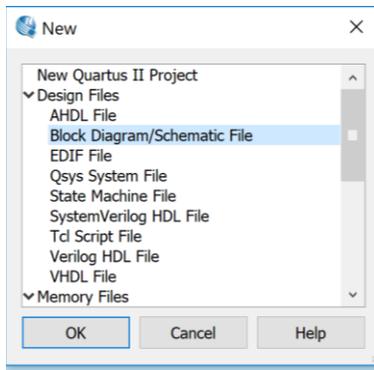


**Figure 7. Quartus II window**

We will describe some of the options for using these resources, but we also recommend exploring these resources on your own to become familiar with Quartus' capabilities. Use the Help menu for additional information.

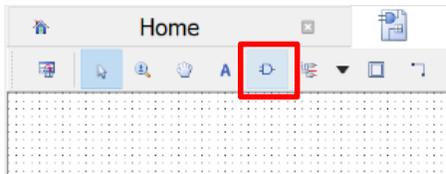
Quartus has a basic and strikingly ugly schematic editor that we will use. It is not particularly sophisticated because designers today primarily use hardware description languages (HDLs) instead of schematics. However, understanding schematics is an important first step to mastering HDLs.

Create a new schematic by choosing **File → New** and selecting **Block Diagram / Schematic File** (as shown in Figure 8), and click OK. A new schematic window named Block1.bdf will appear.



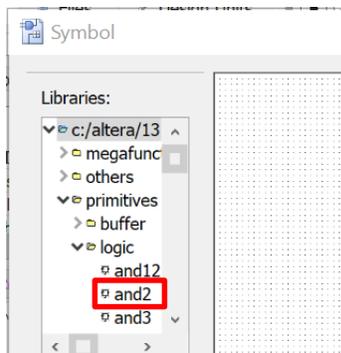
**Figure 8. New Block Diagram/Schematic File**

First, you will place the logic gates from your design. **Click on the Symbol Tool** icon (the button shaped like an AND gate) (see Figure 9).



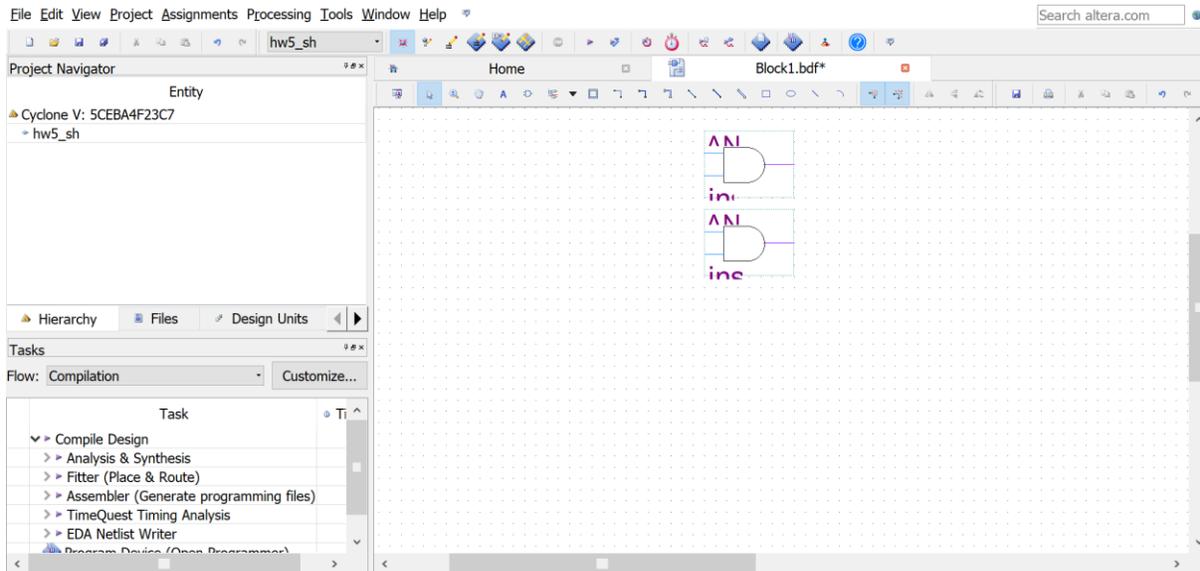
**Figure 9. Add gates to schematic**

In the small window that opens, expand the Altera library by clicking the arrow > next to it. Then look under primitives → logic and choose **and2**, as shown in Figure 10.



**Figure 10. Adding gates from the Altera library**

Click OK, then click again on the schematic window to place multiple 2-input AND gates, as shown in Figure 11. Leave some room between the gates to draw a wire later. Press the Esc key or right click and choose Cancel to get out of the placement mode. Place a single multiple-input OR gate as needed as well using the same method. Add other gates (NOT, etc.) as needed.



**Figure 11. Schematic entry**

Click on the Symbol Tool again and choose **primitives** → **pin** → **input**. Place three input pins on the left side. Leave some space between the pins and the gates so that you can wire them together later. Then choose an output pin and place it on the right. Double click on one of the input pins and change its name to **A**. Leave the default value unchanged at **VCC**. Rename the other inputs to **B** and **C**. Rename the output to **Y**.

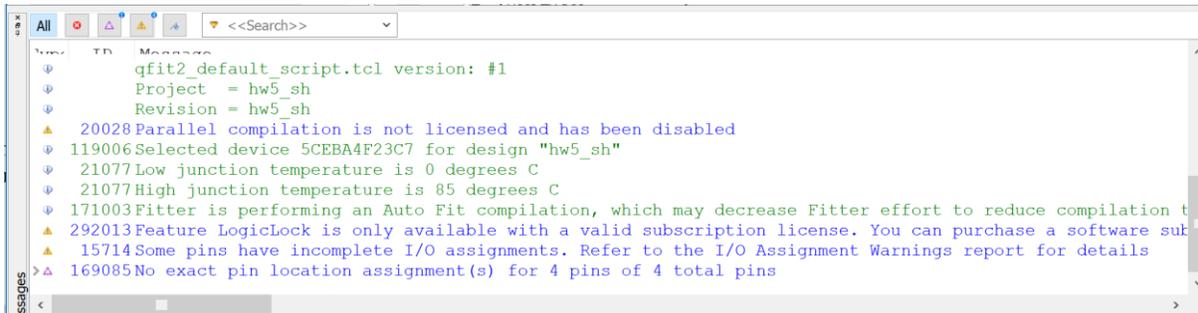
Use the Orthogonal Node Tool (  ) to wire the gates together. Click and drag to connect the pins to gates and the two gates together. As you get close to a pin or gate input/output, it will highlight its port.

If you need to make corrections, use the Selection Tool to grab and move gates or wires. Zoom in and out by using the View menu or holding the Ctrl key while turning the mouse wheel. Use delete and undo (Ctrl-z) as necessary.

Choose **File** → **Save** and save your schematic as **cpe100\_comp1\_xx.bdf**, where “xx” are your initials.

Select the Files tab in the Project Navigator pane to see a list of files of the project (presently just cpe100\_comp1\_xx.bdf). If you need to reopen the file later, double-click on it here.

To check your design, click on Start Compilation (  ) in the Task pane (or, alternatively, choose: **Processing** → **Start Compilation**). You’ll see a compilation report indicating 4 pins (i.e., 3 inputs and 1 output) and 1 logic element. Review the warnings and errors shown in the messages pane (see Figure 12).



**Figure 12. Messages pane**

You may get the following warnings that are harmless:

Warning (292013): Feature LogicLock is only available with a valid subscription license. You can purchase a software subscription to gain full access to this feature.

Warning (15714): Some pins have incomplete I/O assignments. Refer to the I/O Assignment Warnings report for details

Critical Warning (169085): No exact pin location assignment(s) for 4 pins of 4 total pins

etc.

Browse the warnings to make sure there are no mistakes. For example, the following warnings indicate that one of the gates is missing an input connection.

Error (275044): Port "IN" of type NOT of instance "inst3" is missing source signal

Error (12153): Can't elaborate top-level user hierarchy

Error: Quartus II 64-Bit Analysis & Synthesis was unsuccessful. 2 errors, 1 warning

Error (293001): Quartus II Full Compilation was unsuccessful. 4 errors, 1 warning

After you have compiled your design without errors, you are ready to test your circuit in simulation.

#### **Step 4. Simulate your design using ModelSim**

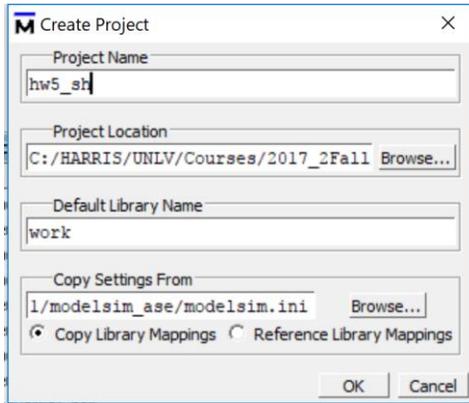
One motivation for drawing your full adder schematic in Quartus II is that you can now use software to simulate the operation of the circuit. This allows us to verify the correctness of your design before actually building the circuit in hardware. In this part of the assignment, you will simulate the design using **ModelSim-Altera 10.1d**. This software comes with the Quartus II 13.1 software.

ModelSim expects a description of a circuit in a hardware description language (HDL) such as Verilog. To convert your schematic to Verilog, open the schematic (available in the Project Navigator pane under the Files tab) and choose **File** → **Create / Update** → **Create HDL Design File for Current File**. Choose **Verilog HDL**, and click **OK**. Your file should be written to `cpe100_comp1_xx.v`. Watch for and correct any warnings or errors that arise.

Now fire up **ModelSim-Altera 10.1d**. Open ModelSim (from the Windows start menu: Start → ModelSim-Altera 10.1d to find the application). Maximize the ModelSim window when it

opens. If prompted, you may wish to associate file types with ModelSim but do not want to use Jumpstart.

Choose **File** → **New** → **Project**. Name the project `cpe100_comp1_xx` and put it in the directory where you are working (e.g. `cpe100_comp1_xx`), as shown in **Figure 13**. Accept the default library name of “work.” Click **OK**.



**Figure 13. Create ModelSim Project**

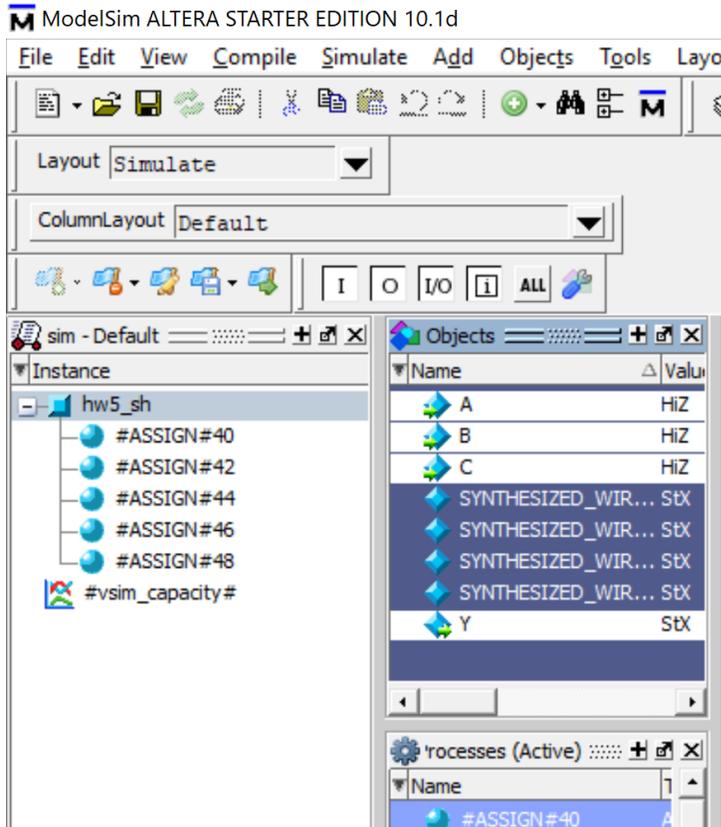
Then click the **Add Existing File** icon and add `cpe100_comp1_xx.v` (browse to your Quartus II project to find it). Click **OK**, then **Close**.

You should see `cpe100_comp1_xx.v` in the ModelSim project pane. Double-click on it to view it. The file should list the inputs and outputs and the wires (using default names if you didn’t name them yourself). It should then have a series of “assign” statements describing the gates. `&` indicates AND. `|` indicates OR. `^` indicates XOR. `~` indicates NOT. Chapter 4 of the textbook describes SystemVerilog (the newest version of Verilog) if you’re interested to learn more about it.

Choose **Compile** → **Compile All** to compile the Verilog code into a form that ModelSim can simulate. Watch for and correct errors in the transcript pane. Then choose **Simulate** → **Start Simulation**. Click on **work** to expand the library, and choose `cpe100_comp1_xx` as your module to simulate. Click **OK**.

You may get a message: `Error: invalid command name ""`. Click **Skip Messages** to ignore it.

ModelSim will open more panes including **sim** and **Objects** panes that help you select signals for the waveform viewer. In the **sim** pane, be sure `cpe100_comp1_xx` is selected. In the **Objects** pane, you’ll see all the inputs, outputs, and internal wires. Select the inputs (A, B, and C) and the output Y using click and ctrl-click, as shown in Figure 14.



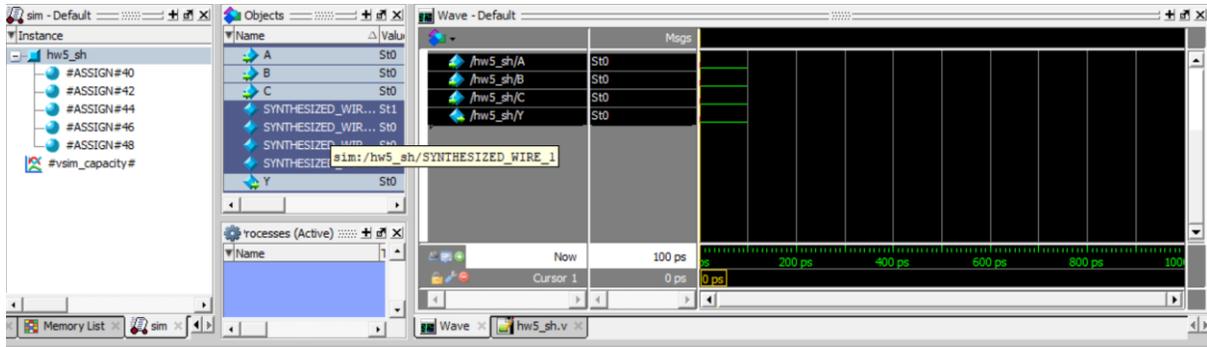
**Figure 14. Selecting signals**

Then right-click and choose **Add** → **To Wave** → **Selected Signals**. A waveform (“Wave”) pane will pop up with the signals.

Now it is time to apply the inputs. In the transcript pane at the bottom, type

```
force A 0
force B 0
force C 0
run 100
```

Press return after each line. This will set all three inputs to 0 and simulate for 100 ns. (Note: that Verilog is case-sensitive; “A” and “a” are different. **Don’t** type: force a 0, etc. or it will not work) You should see all the inputs and outputs with values of “0” (a low level) in the Wave pane, as shown in Figure 15.



**Figure 15. ModelSim simulation waveform**

Next, raise C:

```
force C 1
run 100
```

You'll see C rise. (This gives the  $ABC = 001$  combination). If your design is correct, Y will rise to 1.

Continue with the seven other patterns of inputs to check your truth table.

If you have errors, you may want to look at the internal nodes to track down the problem – i.e., add them to your waveform. Fix the schematic, then regenerate the Verilog file (all in Quartus). Then in ModelSim, recompile and restart the simulation.

If the waveform is not visible, choose **View** → **Wave** from the menu. Click the “Zoom Full” icon in the taskbar  to see the whole waveform of the simulation results. You can also use the “Zoom In” and “Zoom Out” icons:  . Check and see that the output value (Y) is correct. If not, go back and fix your schematic and resimulate. When the output values are correct, you have a working majority circuit!

Save an image of the waveform. Make sure the entire waveform is visible and that only the signals, A, B, C, and Y are visible (ordered from top to bottom). Also make sure you have tested the sequence in counting order (i.e., ABC go in this order: 000, 001, 010, 011, 100, 101, 110, 111). Select **File** → **Export** → **Image...**, and save the file. If needed, you can also print the waveform. Choose **File** → **Print** to print a copy of your waveforms to turn in. You can choose the start and end times in the bottom right of the print dialog box. Now close the ModelSim project.

If everything has gone well then you have now built your first digital logic circuit using a design editor.