Laboratory 1

Introduction to Verilog

1.1 Outcomes and Objective

The outcome of this lab is to introduce you to the Quartus II software, design entry using Verilog and circuit simulation. Through this process you will achieve the following learning objectives.

- Elementary Logical Functions
- Conversion between two different representations of a logic function
- Writing concurrent signal assignment statements for a logic function
- Writing a Verilog statement using primitive logic operations
- Creating a simulation timing diagram for a module

1.2 FPGA: Creating a project in Quartus and running a testbench

In this section you will apply inputs to a 2-input AND gate and observing the output on a timing diagram. Since all this activity will take place in the memory of a computer and not on actual hardware, it is called a simulation. To start this process, you will first have to create a project and add files to it.

1. Select an appropriate working directory for your project. I would recommend selecting your network drive.
   a. Create a new folder *lab1*,
   b. Create another folder within *lab1* called *andgate2*,
   c. Download *andgate2.v* and *andgate2_tb.v* from Canvas,
   d. Save these files in andgate2 directory.
2. Start Quartus II.
   a. If you are prompted by a License Setup choose the free option. You may need to restart Quartus if this happens.
3. Select *File -> New Project Wizard.*
4. In the **Directory, Name, Top-Level Entity** page of the New Project Wizard pop-up:
   a. To the right of the "What is the working directory" box click the ... button,

   b. In the Select Folder pop-up, navigate so you can see the andgate2 directory created in step 1,
   c. Select the andgate2 folder, click Select Folder,
   d. In the "What is the name of this project" field type *andgate2*
   e. click *Next*.

5. In the **Project Type** page of the New Project Wizard pop-up:

   a. Select the *Empty project* radio button,
   b. click *Next*.

6. In the **Add Files** page of the New Project Wizard pop-up:

   a. Click the ... button to the right of File name,
   b. In the Select File pop-up, navigate to, and select, *andgate2.v* and *andgate2_tb.v*, click Open,
   c. The file should appear in the window below,
   d. Click *Next*

7. In the **Family & Device Settings** page of the New Project Wizard pop-up:

   a. Device family, Family: Cyclone V
   b. Package: FBGA
   c. Pin Count: 672
   d. Speed Grade: 7_H6
   e. Select Specific device selected in 'Available devices' list
   f. From the list of available devices, select: 5CGXFC5C6F27C7
   g. Click Next

8. In the **EDA Tool Settings** page of the New Project Wizard pop-up:

   a. In the Simulation row

      i. Tool Name column: ModelSim-Altera
      ii. Formats column: Verilog HDL

   b. Leave other defaults alone
   c. Click Next

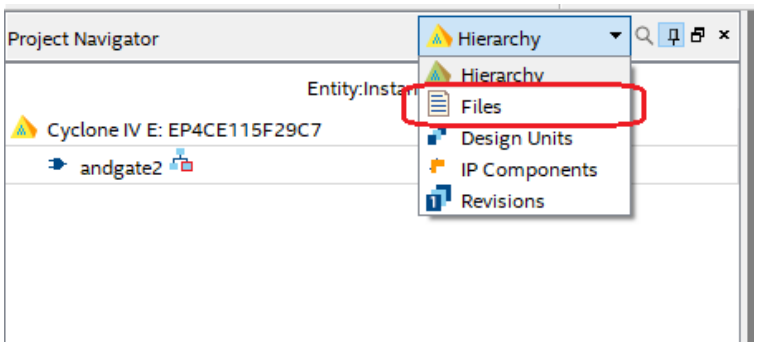
9. In the **Summary** page of the New Project Wizard pop-up:

   a. Review information,
   b. Click Finish.

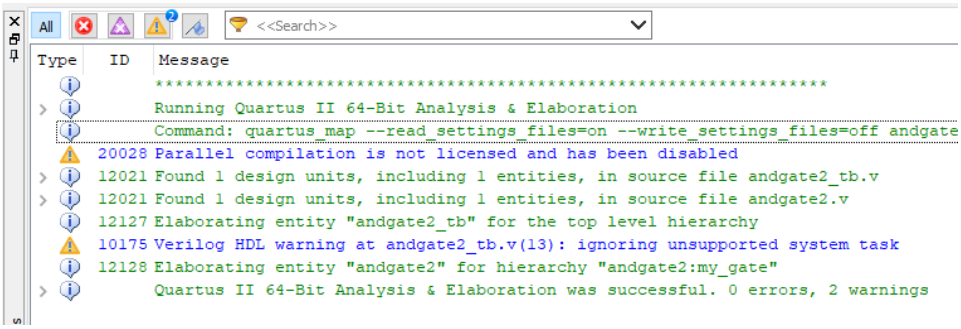
10. Back in the main Quartus II window, Click *Tools -> Options...*
11. In the Options pop-up:

    a. Select *EDA Tool Options* from the Category menu,
    b. If the last row, "ModelSim-Altera" is blank, click on the ... button at right and navigate to the *C:\intelFPGA_lite\18.1\modelsim_ase*, select the *win32aloem* folder, the click Select Folder. Note the software version in these instructions is 18.1 The version installed on your computer may be different. If so, the path should be the same with the exception of the version number.
    c. Click Ok.

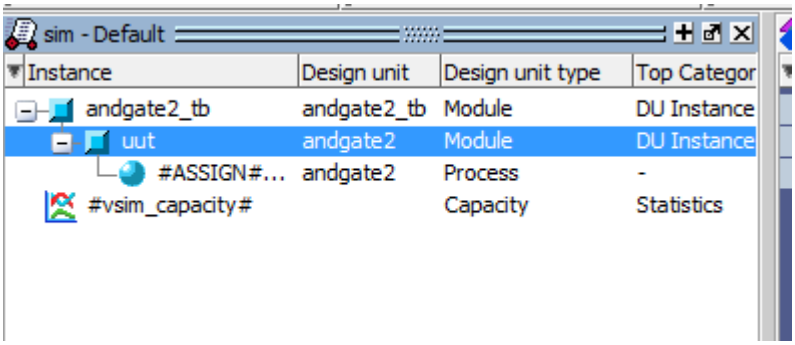
12. Click on the Files tab in the *Project Navigator* pane.



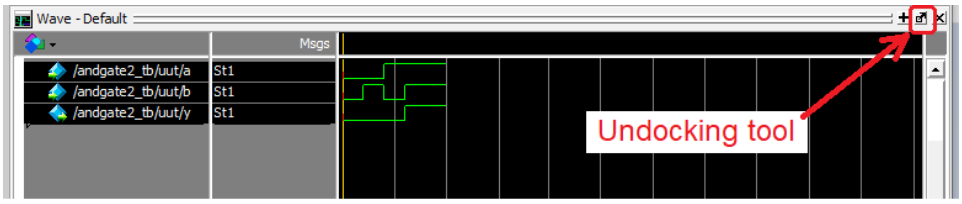
13. Right click on *andgate2_tb* in the *Project Navigator* pane and select Set as Top-Level entity.
14. Double click on *andgate2*.
15. If you added the Verilog file in the correct directory and included it in the project, a Verilog file should pop up on the right.
16. In the main Quartus II window, click on *Processing -> Start -> Start Analysis & Elaboration.* This may take some time, so be patient.
17. If you did everything correctly you should
    a. Notice that andgate_tb is the new top-level entity in the Hierarchy pane. Expand the andgate2_tb by clicking on the ">" arrow to see the entities inside it.
    b. You should see the following messages in the console area, the bottom pane.



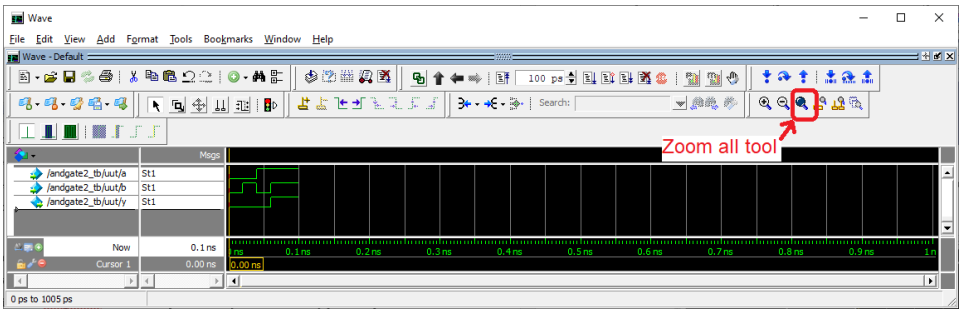
18. In the main Quartus II window, click *Tools -> Run Simulation Tool -> RTL Simulation.* The ModelSim program will launch. This may take a few moments, be patient. If you get a pop-up Nativelink Error window, then go back, check and fix the directory in step 11.
19. In ModelSim, click Simulate -> Start Simulation
20. In the Start Simulation pop-up, expand the *work* library by clicking on the "+" at left. click on *andgate2_tb* and click *Ok.*
21. In the sim pane, right mouse click on uut and select *Add Wave.*



22. Choose *Simulate -> Run -> Run 100*. You should see inputs and output from andgate2. If you see only a small green portion of the waveform on the left margin of the timing diagram, you will need to zoom in on the waveform as follows. First click somewhere in the timing diagram (area under "Undocking tool" in the image below) and then click on the "Zoom all tool" shown in following image.
23. Save this waveform as an image as follows:
    a. Undock the Wave pane by clicking the undocking tool icon.



    b. Resize the undocked Wave window vertically by grabbing its top edge and dragging down. Make the window tall enough to fit all the waves with a little room to spare.



    c. Click the Zoom all tool to file the available horizontal space with the waveform.
    d. Click File -> Export -> Image

    If this does not work, you can take a screen shot of the window by pressing Alt-Print Screen. The "Alt" captures the currently active window into the graphics buffer.

    e. Navigate to your project directory, provide a File name, then click Save
    f. Exit Modelsim using File -> Quit. Do not save wave commands.
24. Back in Quartus, close your current project using File -> Close Project. Save if needed.

1.3 FPGA: Symbolic to Verilog , Timing Diagram, Truth Table

Write Verilog code to realize the function *f02 = a' + bc'* Note that this symbolic expression is written using the notation used in class. This is not a valid Verilog expression.

1. Create **a new project** folder within your *lab1* directory called *function02.*
2. Download *function02.v* and *function02_tb.v* from Canvas to the project directory.
3. Create a project for these two files using the steps above.
4. Modify the line of code that starts with *assign* to realize the function *f02* shown above.
5. Modify *function02_tb.v* so that *f02* is run through every combination of inputs. Assert the inputs in increasing binary numbering order starting from 0,0,0 and going to 1,1,1.
6. Perform simulation using the given testbench as described in previous steps. You will need to "run 100" twice as the simulation is over 100ns long.
7. Save this waveform as an image as done in the previous section. If the waveform is missing, you can add it back in using View -> Waveform.
8. From the information in the timing diagram, produce a truth table for *f02*. Remember that a truth table is an enumeration of every possible input and the associated output. Please look at Chapter 2 in the textbook for some examples if you are unclear about how to setup a truth table.

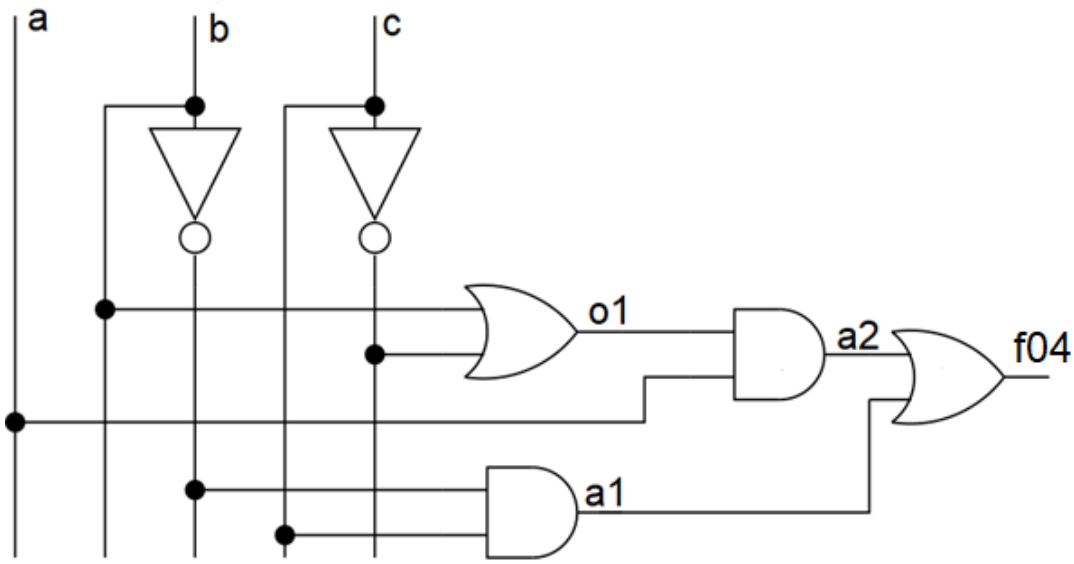
1.4 FPGA: Verilog to Symbolic, Truth Table, Circuit Diagram

The Verilog code in the file function03 contains a complete circuit for *f03*. You will use the Quartus tools to get a timing diagram for the function and, by looking at the Verilog code, determine the symbolic form and circuit diagram.

1. Create **a new project** folder within your *lab1* directory called *function03.*
2. Download *function03.v* and *function03*_tb.v from Canvas to the project directory.
3. Create a project for these two files using the steps above.
4. Modify *function03_tb.v* so that *f03* is run through every combination of inputs. Assert the inputs in increasing binary numbering order starting from 0,0,0 and going to 1,1,1.
5. Perform simulation using this test bench as described in previous steps. You will need to "run 100" twice as the simulation is over 100ns long.
6. Save this waveform as an image, but with the following changes.
   a. Resize the area containing the names of the signals by grabbing the right vertical bar of the name area and moving it right.
   b. Re-order the waves so that f03 is lowest. Do this by grabbing the name "/function03_tb/uut/f03 and moving it below all the other signals.
   c. Color the intermediate signals (p1, p2, p4, p7) yellow by right clicking on them, selecting properties. In the View tab of the Wave Properties pop-up, click the Colors... button for Wave Color and choose Yellow, click Close, then OK.
   d. Change the color of *f03* to red.
7. From the information in the timing diagram, produce a truth table.
8. From the information in *function03.v* draw the circuit diagram for *f03*.
9. From the information in *function03.v* write down the symbolic form for *f03*.

1.5 FPGA: Circuit Diagram to Verilog, Symbolic, Truth Table

Write Verilog code to realize the function *f04* shown in the circuit diagram below.



1. Create **a new project** folder within your *lab1* directory called *function04*.
2. Download *function04.v* and *function04*_tb.v from Canvas to the project directory.
3. Create a project for these two files using the steps above.
4. Modify *function04.v* by writing an assignment statement for each of *o1*, *a1*, *a2*, and *f04*.
5. Modify *function04_tb.v* so that *f04* is run through every combination of inputs. Assert the inputs in increasing binary numbering order starting from 0,0,0 and going to 1,1,1.
6. Perform simulation using this test bench as described in previous steps. You will need to "run 100" twice as the simulation is over 100ns long.
7. Save this waveform as an image as done in a previous section. Color intermediate signals (01, a1, a2) yellow and output red.
8. From the information in the timing diagram, produce a truth table.

1.6 Turn in

Make a record of your response to numbered items below and turn them in a single copy as your team's solution on Canvas using the instructions posted there. Include the names of both team members at the top of your solutions. Use complete English sentences to introduce what each of the following listed items (below) is and how it was derived.

FPGA: Creating a project in Quartus and running a testbench

- Step 23 Timing diagram of AND gate

FPGA: Symbolic to Verilog , Timing Diagram, Truth Table

- Step 4 Verilog code for *f02*
- Step 7 Timing diagram of *f02*
- Step 8 Truth table of *f02*

FPGA: Verilog to Symbolic, Truth Table, Circuit Diagram

- Step 6 Timing diagram of *f03*
- Step 7 Truth table of *f03*
- Step 8 Circuit Diagram of *f03*
- Step 9 Symbolic form of *f03*

FPGA: Circuit Diagram to Verilog, Symbolic, Truth Table

- Step 4 Just the 4 Verilog assign statement for *o1*, *a1*, *a2*, and *f04*.
- Step 7 Timing diagram of *f04*
- Step 8 Truth table of *f04*