Start a new project called 140LTutorial2. In this tutorial, we will learn how to create a simple finite state machine using VHDL. The finite state machine will detects odd number of ones. There will be 3 inputs (data_in, rst, clk) and one output (odd_ind).
Enter the project name 140LTutorial2. Be sure to select HDL as the Top-Level Source Type. Click Next to continue.
Leave everything as default… Your simulator selection should have been selected as Modelsim-XE Verilog in the previous labs already. Click Next to continue.
Leave everything as default. You can always add new source files later.
Click Next to continue.
Since you don’t have existing sources to add, you can skip this step also. Click Next to continue.
A quick summary… click Finish
Now, let’s create a new VHDL source.
Make sure you select VHDL Module on the left pane and enter the name “my_first_fsm”
Click Next to continue.
Enter three inputs (data_in, rst, clk) and one output (odd_ind) to indicate an even or odd number of ones detected from the data_in every clock cycle. Click Next to continue.
A quick summary… click Finish.
Notice the 3 libraries already included as default.
Scroll down and you’ll see the skeleton code built for you. Entity is the definition and architecture is the actual logic implementation.
Add the code for the finite state machine.
Now, add the test case for the state machine by right clicking on the file and choose “New Source” as you did in previous tutorial.
Choose the file you just created to be tested.
Click Next to continue.
A new window will pop up. Be sure to select the Single Clock and select the “clk” as the clock line. Leave everything else as default.
Toggle the data_in line and set rst='0'.
Before running the simulation, you need to open the Simulation Behavioral Model’s “Properties…” window and make sure “Testbench Model Target Language” is VHDL. Once you verify this, run the simulation.
If you typed everything in correctly, you will see the simulation result as expected.
Let’s force a mistake by commenting out the ‘END CASE’ statement and see what happens…
Notice the Modelsim will not run if there is an error.
After making the correction, let’s see if we can see what the circuit will look like when we generate it.
Select the Sources for “Synthesis/Implementation” in the source upper left window.
In the Process window, drill down the “Design Utilities” and run the “Create Schematic Symbol”.
Also, right click on the “Implement Design” and click “Run”. This will translate your code to actual hardware.
If all goes well, you can see all green check marks when you drill down the “Implementation Design”.
You can see the actual logic circuit by running the “View RTL Schematic”.
This is the top level view. Double click on the circuit itself to drill down the detail.
You can see the circuit actually created. Double click on the block next to the inverter.
You now see the logic gates which implemented the logic of this circuit.
Close this window…
You can also see the entire design by running the “View Technology Schematic”.
After double click on the top level circuit, you can see the entire design. Now you are ready to do lab3.part3. Good luck!