Using Quartus and Qsim to simulate Verilog models

Dr. Tom Martin

ECE 2504, Spring 2015

This brief tutorial assumes that you are already familiar with using Quartus and Qsim to simulate schematics. The tutorial will walk you through the steps of creating a new project and adding Verilog files to it. You will then simulate the Verilog models with Qsim, in the same way that you simulated your schematics for projects 0 and 1.

1. Creating a new Quartus project:
2. Start Quartus.
3. Select **File** > **New Project Wizard**.
4. The New Project Wizard window will open. Select Next.
5. Choose a working directory. The default will be in the Altera installation directory, but you should choose a folder where you are storing your ECE 2504 materials.
6. Choose a name for the project. For this tutorial, we will use *verilog\_tutorial*.
7. Choose a name for the top-level entity for the project. This name must match exactly the name of the top-level module in your Verilog model. For this tutorial , we will use the default, which is the same as the name for the project, *verilog\_tutorial.*
8. Select Next.
9. On the Add Files page, select Next without adding any files. (We will create a Verilog file later. If you already had a Verilog file created, you would add it here.)
10. On the Family & Device Settings, set the family to “Cyclone IV E” and then under Available Devices select the EP4CE22F17C6. (Choose the Cyclone IV E under the Family pull down if necessary). The device does not matter for this homework, but when we start using the DE0 Nano board to implement our design later in the semester, the device setting will be important.
11. Select Next.
12. On the EDA Tool Settings page, select Next.
13. On the Summary page, select Finish.

You now have created a new Quartus project. The next set of steps will create a Verilog model in the project.

**Adding a Verilog model to the project**

1. Select File->New, and on the window that opens, select Verilog HDL File and then select OK.
2. A new Verilog file will be created with a default name (probably Verilog1.v) and open in the right Quartus panel. Select File->Save As, and then change the file name to verilog\_tutorial. Make sure the “Add file to current project” option is checked, then select Save.
3. Type the following model into the *verilog\_tutorial.v* file and then save it:

// My first Verilog model

module verilog\_tutorial(a, b, c, d, f);

input a, b, c, d;

output f;

wire g1, g2;

// For the built-in gate primitives, the output is the first port. The remaining ports are inputs.

and GATE1(g1, a, b);

and GATE2(g2, c, d);

or GATE3(f, g1, g2);

endmodule

1. Once you have saved the file, compile it by double-clicking on the “Analysis and Synthesis” button on the left or using the menus, Processing -> Start -> Start Analysis and Synthesis. This step is the same as what you did in projects 0 and 1 for your schematics. If you have any compilation errors, you should correct them before continuing to the remaining steps.

**Simulating the project in ModelSim**

1. You can now simulate your module using ModelSim using the same steps that you used with simulating schematics. If you have difficulty with the following steps, refer back to the more detailed simulation instructions from projects 0 and 1.
2. Open the verilog\_tutorial project in ModelSim.
3. Create the node finder files, and then create a new simulation input file.
4. Set up waveforms for inputs a, b, c, and d that use all possible combinations of values for them. (You may want to Group the inputs and then assign their Value to be a Count Value.) Save the waveform file.
5. Under the Simulation menu item, click Run Functional Simulation.
6. Verify that the output of the simulation behaves as you would expect from the netlist in the Verilog model. Your simulation results should be similar to the results shown below.

