University of California, Davis
Department of Electrical and Computer Engineering

EEC180B DIGITAL SYSTEMS II Spring Quarter 2014

LAB 4: Timing and Power Analysis Tutorial

Objective: In this lab, you will learn how to perform gate-level timing simulation using ModelSim and how to perform power analysis using Altera's QuartusII PowerPlay Power Analyzer. In this lab, you will analyze the timing and power consumption of your Booth multiplier design from Lab 3. In the next lab, you will use these tools and techniques to analyze and compare the timing and power consumption of several different multiplier designs in order to determine the optimal, energy-efficient design.

I. Gate-Level Timing Simulation using ModelSim

The procedure for performing gate-level timing simulation of a Quartus II design using ModelSim is described in detail in Chapter 2 of Volume 3: Verification of the Quartus II Handbook set. This chapter is available on-line at

http://www.altera.com/literature/hb/qts/qts_qii53001.pdf

The material in this section is adapted from this Altera document for our specific environment. Please refer to the on-line material if you desire more information.

Gate-level timing simulation is a ***post*** place-and-route simulation using (in our case) worst-case delays. In order to simulate the post place-and-route design, you will need to revise the testbench code given in Lab 3 so that it interfaces with your final synthesizable Verilog code for your Booth multiplier. The main changes will be the input and output signal names of your Verilog module. For example, the HEX3-0 signals are used to output the product rather than the Product signal and the Done signal will be output on one of the LED output signals.

1. Revise the testbench code so that it works with your synthesizable version of the Booth multiplier (i.e. the design that you actually download to the DE2 board). You should not change your Booth multiplier code at all; just make changes to the testbench.

2. To verify your new testbench code, you can perform a functional simulation of your final design.

3. In Quartus II, open your Booth multiplier project. Before you compile your design, you will set configuration options to produce a Verilog output file (**.vo**) and a Standard Delay Format Output file (**.sdo**), which will both be used in the gate-level simulation.

4. On the **Assignments** menu, click **EDA Tool Settings**. The **Settings** dialog box appears.

5. In the **Category** list, click the "+" icon to expand **EDA Tool Settings** and select **Simulation**. The **Simulation** page appears.

6. In the **Tool name** list, select **ModelSim**.

7. Under **EDA Netlist Writer options**, in the **Format for output netlist** box, select **Verilog**. By default, the post-synthesis netlist will be written to the <*project_directory*>/**simulation/modelsim** directory.

8. Perform a full compilation of your Booth multiplier design. On the **Processing** menu, click **Start Compilation**. Verify that your design compiles without errors and that the Verilog Output file (.vo) and the Standard Delay Format Output file (.sdo) are written to the <*project_directory*>/**simulation/modelsim** directory.

9. Exit Quartus II.

10. Copy your testbench program to the <*project_directory*>/**simulation/modelsim** directory. Run ModelSim from this directory and create a new project, adding the testbench file (.v) and the Verilog Output file (.vo) to the project.

11. Compile your design files (testbench and post synthesis netlist) in ModelSim.

12. Run the gate-level simulation by typing the following command at the ModelSim command prompt:

    vsim –t 1ps –L altera –L altera_mf –L lpm –L work work.tb_booth

    assuming the top level module of your testbench is named *tb_booth*.

13. Open a waveform file with the command
    add wave *
    and change the radix of the signals to Hexadecimal.

14. Simulate your design through all of the test vectors. Compare your simulation output with your functional simulation output. Determine the typical logic delays in various parts of your circuit.

Exercise
1. Record the delay from the rising edge of the clock to the "Done" signal and include it in your report.
2. Give one example of the difference between what you see in the waveforms produced during timing simulation and functional simulation.

II. QuartusII Classic Timing Analyzer

A much easier way to check the timing performance of your design is to run the QuartusII Classic Timing Analyzer tool.

1. Run QuartusII from your Quartus project directory. Open the project for the Booth multiplier.

2. On the **Processing** menu, select the Classic Timing Analyzer tool.

Exercise:

1. *What are tsu, tpd, tco and th?*
2. *What is the critical path of your circuit and the maximum frequency of operation?*
3. *Is the critical path consistent with your design?*
4. *Change the maximum clock frequency constrain to 80 MHz.*
5. *Recompile your design.*
6. *What is the critical path of your circuit? Explain if there is any difference.*

For more information on using the QuartusII Classic Timing Analyzer, refer to Chapter 8 of Volume 3: Verification of the Quartus II Handbook set. This material can be found on-line at:
http://www.altera.com/literature/hb/qts/qts_qii53004.pdf

III. QuartusII PowerPlay Power Analyzer

The PowerPlay power analysis tool is described in detail on Altera's website at:

http://www.altera.com/literature/hb/qts/qts_qii53013.pdf

As with the gate-level timing simulation, the PowerPlay Power Analyzer tool requires that your design is synthesized and placed and routed on the target device. In our case, we will use the ModelSim gate-level timing simulation results to generate signal activity and static probability information in a Value Change Dump file (**.vcd**).

To create a VCD file using the ModelSim simulator, perform the following steps:

1. On the **Assignments** menu, click **EDA Settings**. The **Settings** dialog box appears.

2. In the **Category** list, select **Simulation**. You should have already made several entries on this Simulation dialog box such as specifying ModelSim as the simulation tool and Verilog as the output file format.

3. Check the box **Generate Value Change Dump (VCD) file script**. This should also add checks in the boxes for **Map illegal HDL characters** and **Enable glitch filtering**. Also fill in the **Design Instance Name** with the instance name for your Booth multiplier given in the testbench file. For example, this could be the

instance name **UUT** if you followed the example testbench from Lab 3.Click **OK** to close the Simulation dialog box.

4. Compile the design with the QuartusII software. Quartus should now generate the post-synthesis netlist (.vo), the Standard Delay Format Output File (.sdo) as before as well as a script (.tcl) to cause ModelSim to produce a VCD file.

5. Open ModelSim and open the project for the gate-level timing simulation (probably in the *<project_directory>*/simulation/modelsim. Compile your testbench and multiplier circuit and load the testbench as before using the **vsim** command given in Part I. Open a waveform window and specify the radix of all the signals as Hexadecimal.

6. Run the tcl script using the **do** command as illustrated below:

   do booth_dump_all_vcd_nodes.tcl

   (Use the actual name of your script file if it doesn't match the example given above.)

7. Now run the testbench through all of the test vectors, as you did in the gate-level timing simulation. For example, you can use a command such as

   run 4000 ns

   to run the simulation for a fixed amount of time needed to check each test vector.

8. Quit the ModelSim simulator using the command **quit**. Verify that ModelSim has generated a VCD file (.vcd) in your directory.

9. Run QuartusII and open the QuartusII project.

10. On the **Assignments** menu, click **Settings**. The **Settings** dialog box appears.

11. In the **Category** list, select **PowerPlay Power Analyzer Settings**.

12. Select the setting **Use input file(s) to initialize toggle rates and static probabilities during power analysis**. Then click the **Add…** button and then browse to the correct directory to add the VCD file generated during your ModelSim timing simulation. Also, select the **Perform glitch filtering on VCD files**.

13. You should consult the on-line reference for the **PowerPlay Power Analyzer** tool to understand the options available to you and to decide on your configuration. Once you have determined the options, click **OK** to close the **Settings** dialog box.

14. On the **Processing** menu, click **PowerPlay Power Analyzer Tool**. Click **Start** to run the tool.

15. Once the **PowerPlay Power Analyzer Tool** completes, click **Report** to open the **PowerPlay Power Analyzer Summary** window. Analyze the results.

Exercise
Compare the power consumption of 8-bit and 16-bit Booth multiplier designs. This involves changing the parameter "n" in your design from 8 to 16 and repeating the steps.