CSE P567 - Winter 2010

Lab 2 – Introduction to Simulation Using ModelSim

Overview

In this lab you will learn how to simulate, test and debug Verilog designs using the ModelSim simulator. The first task will be to go through a tutorial on ModelSim so that you understand how to do simple simulation tasks. Next you will use ModelSim to simulate a design that we will give you for a simple wavelet filter. This will make use of a "test fixture" aka "test harness" that automatically generates test data and checks the results. This will show that the design is in fact incorrect, so then you will then have to modify the design to get it to compute the correct values.

Step 0 – Install ModelSim

If you have not already done so, install ModelSim on your own computer so you can get started on the Lab at home. You will need ModelSim to do the next homework assignment as well. Instructions for installing ModelSim are on the course web page.

Step 1 – ModelSim Tutorial

Go through Chapters 3, 4, and 6 of the ModelSim Tutorial. Make sure you follow the steps for Verilog! The tutorial uses a simple Verilog counter design, which you have to copy from the ModelSim install folder. You should be able to figure out this counter design, but you can learn how to do simulation without understanding it. After going through the tutorial, you should know all the concepts that you need to know for this class.

The tutorial takes 1-2 hours to complete, so if you wait until class to go through the tutorial, you will probably not have enough time to finish the lab assignment.

Step 2 – Simulate the Wavelet Filter Design

Copy the wavelet filter Verilog files from the course folder called **simpleWavelet** into a new directory for Lab 2. Create a ModelSim project, and use ModelSim to simulate this design. The wavelet_tf.v file is a test fixture that tests the wavelet filter. You can just run the simulation as you did in the tutorial and look at the results to see if it is working correctly.

The 1D wavelet filter operates on blocks of 4 data values, translating input blocks into output blocks containing 4 different values. The data in an input block is labeled D0, D1, D2, D3 and the data in an output block is labeled Q0, Q1, Q2, Q3. The wavelet computation is as follows:

Q0 = D0 – D1
Q1 = D2 – D3
Q2 = D0 + D1 + D2 + D3
Q3 = D0 + D1 – D2 – D3

Step 3 – Fix the Wavelet Filter

The final step is to figure out why the wavelet filter is giving the wrong answers and fix the Verilog code to produce the correct results.

1. Take a look at this Verilog code and comment on how fast this design is (clock speed) assuming that adds and subtracts are relatively slow. Can you suggest ways to make it faster?

2. Comment on how much hardware it uses and suggest ways to make it smaller.

Step 4 – A more complicated Wavelet Filter (Optional)

The **wavelet** folder contains a different design for the wavelet filter. Simulate this design to see if it works and fix it if gives the wrong answers.

3. Compare the speed and cost of this design relative to the simple wavelet design. Are there general conclusions you can make about these two designs?