Department of Electrical and Computer Engineering

University of California, Davis

Lab 3: Simulating Using Testbenches

I. Prelab

Complete the following and submit your work at the beginning of your lab session.

1. Read the document: *A Guide for Using ModelSim* posted on the course web page
2. [15 pts] Complete a preliminary design of your three adders (draft circuit diagrams and verilog code)

Details not specified in this Lab should be chosen by you and plainly explained in your lab report.

II. Three 3-bit adder modules

Design and implement in verilog three 3-bit adders—meaning they have two 3-bit inputs, a *ci* carry input, and one 4-bit output. Begin by drawing detailed *circuit* diagrams of your add3.v and add3_error.v

1. add3.v — Build using three full-adder cells.
2. add3_error.v — Build using any method you like. The module adds two 3-bit numbers except for the two cases 7 + 4 + (*ci*=0) = 12, where 7 and 4 can be on either input: 7 + 4 + 0 = 12 and 4 + 7 + 0 = 12.
3. add3_ref.v — A Golden Reference design following the guidelines from posted handouts; for example, it must be written in a fundamentally different way than add3.v

III. Testbench #1

Design and implement in verilog a testbench that uses Approach #1 (one large `initial` block, however you do not need a *clock* for this lab) explained on slides 115–116 in the *Verilog 5: Testing* handout.

A) Test fifteen different input cases for add3.v and print inputs and outputs in text so it is clear your adder is functioning correctly. You must include these cases: 0+0, 0+1, 1+0, 5+5, 7+0, 0+7, 7+7.

B) Using the same testbench, generate waveforms that show any ten test cases clearly enough to verify correct operation.

IV. Testbench #2

Design and implement in verilog a testbench that uses the "Golden Reference" verification method explained in the *Verilog 5: Testing* handout. Use add3.v for the hardware design and use add3_ref.v for the golden reference. In addition to printing the input and output, it must also print "pass" or "fail" depending on whether the hardware design matches the reference.

A) Test all possible input cases.

B) Again test all possible input cases but use add3_error.v instead of add3.v to verify your testbench is checking correctly.

V. Submitted Work [100 pts total]

[15 pts] **Prelab**

[60 pts] **Lab Checkoff: ModelSim**

Demonstrate IV.A and IV.B operation to your TA.
Demonstrate your design compiling and operating. Your TA will ask you to generate and will record a `certutil` hash of your top-level files and this must match the hash of the files you upload to canvas so no file modifications are possible after your demo. The hash is calculated using the following commands:

1. in Quartus, double click a few key modules determined by your TA, into the Project Navigator Hierarchy pane to open its Verilog
2. right click the file's tab and select Copy Full Path
3. open a Command Prompt window (by typing "cmd" into the Windows search bar)
4. type "`certutil -hashfile` ***`filepath`***" where ***`filepath`*** is the file path that was copied earlier

[25 pts] **Lab Report**

Upload the following two files to Canvas by the end of your lab session—this is essential to receive any credit for the entire lab.

I. A single Zip file. Submit all Verilog hardware and testbench code that you wrote in a single zip file. Do not include any code that you did not write such as files generated by Quartus or IP components.

- Make a folder on your computer
- Copy all files to be submitted into the folder
- "zip" the folder into a single .zip file
- Log onto Canvas, click Assignments, find the correct lab number
- Upload the .zip file

II. A single PDF or Word file.

1. Detailed *circuit* diagrams of your add3.v, add3_error.v, and testbench #2
2. Text printout from III.A
3. Waveform printout from III.B
4. Text printout from IV.A
5. Text printout from IV.B

2021/04/01 Posted
2021/04/21 Fixed typo, "slides 115–116"