CS223 Laboratory Assignment 2

2-bit adder on FPGA

Lab dates and times:

Section 1:	25.02.2021 Thursday	13:30-17:20
Section 2:	23.02.2021 Tuesday	08:30-12:20
Section 3:	24.02.2021 Wednesday	08:30-12:20
Section 4:	25.02.2021 Thursday	08:30-12:20

Location: EA Z04 (in the EA building, straight ahead past the elevators)
Groups: Each student will do the lab individually. Group size = 1

Preliminary Report (30 points)

In the previous lab, you implemented a 1-bit half-adder and full-adder using gates on the bread board. In this lab you will implement very similar circuits, but this time on the FPGA. Today's lab needs considerable advance preparation. You need to learn how to work with Xilinx's design tool set before attending the lab. Besides, Systemverilog models and testbenches should be prepared in advance, and assembled neatly into a Preliminary Report with a printed cover page and printed pages for the Systemverilog codes. Each page should have a proper heading. The content of the report is as follows:

a) A cover page which includes the following: course name, number and section, the number of the lab, your name, student ID, date.
b) Schematic for a 1-bit fulladder, made from 2-input XOR, AND and OR gates (you can refer to figures in Lab 1).
c) Schematic for a 2-bit adder made from two 1-bit fulladders(as ready black-boxes, you don't need to draw the same things again)
d) Dataflow Systemverilog module for the 1-bit fulladder.
e) Structural Systemverilog module for the 1-bit fulladder, and a testbench for it.
f) Structural Systemverilog module for the 2-bit adder, and a testbench for it. Use the 1-bit fulladder module you wrote inside.

Note that dataflow means modeling with Boolean equations, using continuous assignment statements, whereas structural refers to using and combining simpler pieces of modules. You can refer to the slides of chapter 4 of your textbook in Moodle while preparing your modules and testbenches.

Additional pre-lab work:

You should study the following documents (available on Moodle) to be familiar with steps of design flow (Simulation, Synthesis, Implementation, Generation of Programming File, Downloading to FPGA board), using Xilinx Vivado tool. You can download, install and practice working with Xilinx Vivado on your own computer with free webpack license.

- Suggestions for Lab Success.
- Basys3 Vivado Decoder Tutorial.
- Vivado Tutorial.
- BASYS-3 FPGA Board Reference Manual (just take a look, and later use it as reference when needed).

The Preliminary Report must be uploaded to Moodle before the start of lab for each section.

Implementation on FPGA (70 points)

In this step, you implement your modules on FPGA. You don't need to connect your BASYS-3 board to the Beti board. Working with standalone BASYS-3 and having it connected to your computer is enough for this lab. There are some switches and LEDs available on BASYS-3 which you can use them.

- *Create a new Xilinx Vivado Project. Use appropriate names for files and folders, keeping the project in a directory where you can find it later and erase it (at the end of lab).*

a) Simulation: Implement the 1-bit fulladder module in dataflow style (preliminary part-d.) Then, using the SystemVerilog testbench code you wrote, verify in simulation that your circuit works correctly.
b) Simulation: Implement the 1-bit fulladder module in structural style (preliminary part-e.) Then, using the SystemVerilog testbench code you wrote, verify in simulation that your circuit works correctly.
c) Simulation: Implement the 2-bit adder module using two 1-bit fulladers you wrote(preliminary part-g.) Then, using the SystemVerilog testbench code you wrote, verify in simulation that your circuit works correctly.
d) When you are convinced that your codes work correctly, show the simulation results to your TA. Be prepared to answer questions that you may be asked.
e) Program the FPGA: Now, follow the Xilinx design flow to synthesize, create programming file, and program all three modules to BASYS-3 FPGA board.
f) Test your design: Using the switches and LEDs (on BASYS-3) that you have assigned in constraint file, test your designs. When you are convinced that they work correctly, show the physical implementation results to the TA. Be prepared to answer questions that you may be asked.

Submit your code for MOSS similarity testing

Finally, when you are done and before leaving the lab, you need to combine all the design and simulation sources you wrote in a single file named *StudentID_SectionNumber.txt*. This will include all the Systemverilog codes you wrote. You don't need to add the constraints file(*.xdc file). If you have multiple files, just copy and paste them in order, one after another inside text file. Even if you didn't finish, or didn't get the Systemverilog part working, you must submit your code to the Moodle Assignment for similarity checking. Your codes will be compared against all the other codes in all sections of the class, by the MOSS program, to determine how similar it is (as an indication of plagiarism). So be sure that the code you submit is code that you actually wrote yourself! All students must upload their code to the

‘Moodle>Assignment’ specific for your section. Check submission time and don’t miss it before leaving the lab. After taking a backup of your work, don’t forget to delete it from computer. Because students of other sections will work with your system too.

Clean Up

1) Clean up your lab station, and return all the parts, wires, the Beti trainer board, etc. Leave your lab workstation for others the way you would like to find it.

2) CONGRATULATIONS! You are finished with Lab #2 and are one step closer to becoming a computer engineer.

NOTES

--Advance work on this lab, and all labs, is strongly suggested.

--Be sure to read and follow the Policies for CS223 labs, posted in Moodle.