CS223 Laboratory Assignment 3

Modeling Decoders and Multiplexers in SystemVerilog

Lab dates and times:

Section 1:	23.03.2020 Monday	08:40-12:25
Section 2:	25.03.2020 Wednesday	08:40-12:25
Section 3:	10.03.2020 Tuesday	08:40-12:25
Section 4:	12.03.2020 Thursday	08:40-12:25

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 pts)

Today's lab needs considerable advance preparation. Circuit schematics, SystemVerilog models and testbenches should be prepared in advance and assembled neatly into a Preliminary Report with a cover page and pages for the schematics and SystemVerilog codes. Each page should have a proper heading. The content of the report will be as follows:

(a) A cover page including: course code, course name and section, the number of the lab, your name-surname, student ID, date.

(b) Write a *Behavioral* SystemVerilog module for the 3:8 decoder and a testbench for it.

(c) Write a *Behavioral* SystemVerilog module for the 4:1 multiplexer.

(d) Draw a circuit schematic (block diagram) for the 8:1 multiplexer by using two 4:1 multiplexers, an INVERTER, two AND gates, and an OR gate.

(e) Write a *Structural* SystemVerilog module for the 8:1 multiplexer you designed in part (d) and a testbench for it.

(f) Draw a circuit schematic (block diagram) to implement the Boolean function $F(A,B,C,D) = \sum(1, 2, 7, 8, 9, 10, 12, 13)$ by using only one 8:1 multiplexer and one INVERTER.

(g) Write a *Structural* SystemVerilog module for the circuit you designed in part (f) and a testbench for it.

Note that behavioral model describes the function of a module using Boolean equations and continuous assignment statements, whereas structural modeling refers to using and combining simpler pieces of modules (it is an application of hierarchy). You can refer to the slides of Chapter 4 of your textbook in Unilica while preparing your SystemVerilog modules and testbenches.

The Preliminary Report must be uploaded to Unilica before the start of lab. You will need a copy of your designs and SystemVerilog programs with you to use it in the lab. Therefore, you should get a copy of it before you come to the lab, for your own use.

Additional pre-lab work:

You should read the following documents (available on Unilica) to be familiar with steps of design flow (Simulation, Synthesis, Implementation, Bitstream Generation, 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.
- Basys 3 Vivado Decoder Tutorial.
- Vivado Tutorial.
- Basys 3 FPGA Board Reference Manual.

Implementation on FPGA

In this step, you will implement your modules on FPGA board. 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 to test your designs.

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

Part A: Decoder (20 pts)

Decoders are widely used in digital design, as a building block. They can be built from logic gates and their function is often described (and modeled in System Verilog) with Boolean equations. A 3:8 decoder decodes a 3-bit input binary number by setting only one of the 8 outputs to 1. The outputs of a decoder are called one-hot, because exactly one is HIGH at a given time. Decoder outputs are mutually exclusive and in fact each output represents a single minterm.

(a) Simulation: Using the SystemVerilog module for the 3:8 decoder and testbench code you wrote in preliminary part (b), verify in simulation that your circuit works correctly.

(b) Program the FPGA: Now, follow the Xilinx Vivado design flow to synthesize, implement, generate bitstream file, and download your 3:8 decoder to Basys 3 FPGA board.

(c) Test your design: Using the switches and LEDs (on Basys 3) that you have assigned in constraint file (.xdc), test your decoder. When you are convinced that it works correctly, show the physical implementation results to the TA. Be prepared to answer questions that you may be asked.

Part B: Multiplexer and Boolean function (50 pts)

Multiplexers are among the most commonly used combinational circuits. They choose an output from among several possible inputs based on the value of a select signal. A multiplexer (MUX) is a building block and it can be used to build more complex systems in digital design. In addition, multiplexers can be used as lookup tables to implement logic functions. In general, a $\mathbf{2^N}$ input multiplexer can be used to perform any N-input logic function.

A $\mathbf{2^N}$**:1** multiplexer has $\mathbf{2^N}$ data inputs and 1 output. Using N-bit select signal (control input), it selects between one of data inputs to connect to output. Wider multiplexers can be built from smaller multiplexers as you will see in this part of the lab.

(a) Simulation: Using the SystemVerilog module for the 8:1 multiplexer and testbench code you wrote in preliminary part (e), verify in simulation that your circuit works correctly.

(b) Program the FPGA: Now, follow the Xilinx Vivado design flow to synthesize, implement, generate bitstream file, and download your 8:1 multiplexer to Basys 3 FPGA board.

(c) Test your design: Using the switches and LEDs (on Basys 3) that you have assigned in constraint file (.xdc), test your multiplexer. When you are convinced that it works correctly, show the physical implementation results to the TA.

(d) Simulation: Using the SystemVerilog module for the Boolean function and testbench code you wrote in preliminary part (g), verify in simulation that your circuit works correctly.

(e) Program the FPGA: Now, follow the Xilinx Vivado design flow to synthesize, implement, generate bitstream file, and download your circuit to Basys 3 FPGA board.

(f) Test your design: Using the switches and LEDs (on Basys 3) that you have assigned in constraint file (.xdc), test your circuit. When you are convinced that it works correctly, show the physical implementation results to the TA.

Submit your code for MOSS similarity testing

Finally, when you are done and before leaving the lab, you need to upload the file *StudentID_SectionNumber.txt* created in the Implementation on FPGA part. Be sure that the file contains exactly and only the codes which are specifically detailed above. If you have multiple files, just copy and paste them in order, one after another inside text file. Check the specifications! Even if you didn't finish or didn't get the SystemVerilog part working, you must submit your code to the Unilica 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 the code that you actually wrote yourself! All students must upload their code to the 'Unilica Assignment' specific for their sections. 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#3 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 Unilica.

LAB POLICIES

1. There are three computers in each row in the lab. Don't use middle computers, unless you are allowed by lab coordinator.
2. You borrow a lab-board containing the development board, connectors, etc. in the beginning. The lab coordinator takes your signature. When you are done, return it to his/her, otherwise you will be responsible and lose points.
3. Each lab-board has a number. You must always use the same board throughout the semester.
4. You must be in the lab, working on the lab, from the time lab starts until you finish and leave. (bathroom and snack breaks are the exception to this rule). Absence from the lab, at any time, is counted as absence from the whole lab that day.
5. No cell phone usage during lab. Tell friends not to call during the lab hours--you are busy learning how digital circuits work !
6. Internet usage is permitted only to lab-related technical sites. No Facebook, Twitter, email, news, video games, etc--you are busy learning how digital circuits work !
7. You need to be in lab on time and turn in your preliminary report at the start of the lab.