

CS223 Laboratory Assignment 3

Modeling Decoders and MUXs in System Verilog

Lab dates and times:

Section 1:	24.10.2020 Saturday	Even (08:30-10:00), Odd (10:00-11:30)
Section 2:	24.10.2020 Saturday	Even (15:30-17:00), Odd (17:00-18:30)
Section 3:	25.10.2020 Sunday	Odd (15:30-17:00), Even (17:00-18:30)
Section 4:	23.10.2020 Friday	Even (08:30-10:00), Odd (10:00-11:30)
Section 5:	19.10.2020 Monday	Odd (13:30-15:00), Even (15:00-16:30)
Section 6:	20.10.2020 Tuesday	Even (13:30-15:00), Odd (15:00-16:30)

Location: EA Z04 (in the EA building, straight ahead past the elevators)

Each student should attend the time block depending on their student id. A student with student id 22050977(Odd) from section 4 should attend lab at 10:00 am on Friday for example.

Preliminary Report (30 points)

Today's lab needs considerable advance preparation. These advance designs and SystemVerilog models should be prepared in advance, and assembled neatly into a Preliminary Report with a cover page and pages for the SystemVerilog codes. Each page should have a proper heading. The report should be uploaded on Moodle as a pdf file before the start of the lab. The content of the report will be as follows:

- A cover page including: course code, course name and section, the number of the lab, your name-surname, student ID, date.
- Behavioral SystemVerilog module for 2-to-4 decoder and a testbench for it.
- Behavioral SystemVerilog module for 4-to-1 multiplexer.
- Schematic (block diagram) and structural System Verilog module of 8-to-1 MUX by using two 4-to-1 MUX modules, two AND gates, an INVERTER, and an OR gate. Prepare a test bench for it.
- Schematic (block diagram) and SystemVerilog module for $F(A,B,C,D)=\sum(0,1,3,4,7,8,10,11,15)$ function, using one (not two) 8-to-1 multiplexer and an inverter.

You can refer to the slides of chapter 4 of your textbook while preparing your modules and test benches. Don't forget that you have to hand in your reports at the start of the labs and penalties may apply if you fail to do so!

Part A: Decoders (25 points)

Decoders are widely used in digital design, as a building block. Although they themselves can be built with logic gates, their function is often described (and modeled in System Verilog) rather than their structure. As you will see, decoders can be composed into larger decoders.

A 2-to-4 decoder decodes a 2-bit input binary number by setting exactly one of the decoder's 4 outputs to 1. Unless it has an enable signal, one and only one output of a decoder will ever be 1 at the same time, corresponding to the current value of the inputs. With an enable signal, it is possible to make all the outputs be 0, when the decoder is disabled. When enabled, it behaves as described above. Decoder outputs are mutually exclusive, and in fact are the minterms of the inputs.

- *Create a new Xilinx Vivado Project to do a), b), c) and d). 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) Write code: Give the System Verilog code which models a 2-to-4 decoder in behavioral style. (This means modeling with Boolean equations, using continuous assignment statements.)
 - b) Simulate it: Using the System Verilog testbench code, verify in simulation that your 2-to-4 decoder with enable is working correctly. (Be sure to compare the order of the ports in your module with the order of the ports in the instantiation of your decoder in the testbench, to make sure they match 1-to-1.)
 - c) Make FPGA project: Now, follow the Xilinx design flow to synthesize, create programming file, and download your 2-to-4 decoder to your BASYS-3 FPGA board.
 - d) Test it: Using the switches and LEDs on BASYS-3 that you have assigned now, 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: Multiplexers and Boolean function implementation (45points)

A multiplexer ("MUX" for short) is another higher-level building block, used widely in digital design. A M-to-1 multiplexer has M data inputs and 1 data output, and allows only one input to pass through to the output. A set of select inputs determines which input to pass through. MUXs can be composed into larger MUXs, as you will see in this part of the lab.

- a) Write code: Give dataflow System Verilog module for a 4-to-1 multiplexer.
- b) Simulate it: Simulate your 4-to-1 multiplexer and show it to your TA.
- c) Write code: Write structural System Verilog code for an 8-to-1 multiplexer using two 4-to-1 multiplexers, two AND gates, an OR gate and an inverter.
- d) Simulate it: Simulate 8-to-1 multiplexer and show it to your TA.

- e) Test it: Set up the circuit you designed for $F(A,B,C,D)=\sum(0,1,3,4,7,8,10,11,15)$ in preliminary work in a new module, using a single 8-to-1 multiplexer. Using inverses of signals is allowed. Test your circuit using switches as input and a LED as output. Show your circuit to 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 copy all the SystemVerilog codes you wrote to a txt file named StudentID_name.txt and upload it to Moodle. 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 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 indication of plagiarism). So be sure that the code you submit is the code that you actually wrote yourself!

Part 3: 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.

LAB POLICIES

1. There are three computers in each row in the lab. Don't use middle computers, unless you are allowed by lab supervisor.
2. You borrow a Lab-board containing the development board, connectors, etc. in the beginning. The lab supervisor takes your signature. When you are done, return it to her, otherwise you will be responsible and lose points.
3. Each Lab-board has a number. You must always use the same trainer board pack 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. If you come to lab later than 20 minutes, you will lose that session completely.
8. When you are done, DO NOT return IC parts into the IC boxes, where you've taken them first. Just put them inside your Lab-board box. Lab coordinator will check and return them later.