CS223 Laboratory Assignment 3

Modeling Decoders and MUXs in System Verilog

Lab dates and times:

- Section 1: Oct 16, Mon 08:30-12:20 in EA-Z04
- Section 2: Oct 17, Tue 08:30-12:20 in EA-Z04
- Section 3: Oct 18, Wed 08:30-12:20 in EA-Z04
- Section 4: Oct 16, Mon 13:30-17:20 in EA-Z04
- Section 5: Oct 20, Fri 08:30-12:20 in EA-Z04
- Section 6: Oct 17, Tue 13:30-17:20 in EA-Z04

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)

Today's lab needs considerable prior preparation. These prior designs and System Verilog models should be prepared in advance, and assembled neatly into a Preliminary Report with a printed cover page and printed pages for the schematics and System Verilog codes. Each page should have a proper heading. The contents of the report should be as follows:

a) A cover page which includes the following: course name and code number, the number of the lab, your name and student ID, date, <u>number of your trainer pack (remember lab policies. You must always use same pack number).</u>
b) **[5]** Behavioral SystemVerilog module for 2-to-4 decoder and a testbench for it.
c) **[5]** Behavioral SystemVerilog module for 2-to-1 multiplexer.
d) **[5]** Behavioral SystemVerilog module for 4-to-1 multiplexer by using three 2-to-1 multiplexer.
e) **[5]** 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.
f) **[10]** Schematic (block diagram) and SystemVerilog module for F(A,B,C,D,E)= $\sum$(2,4,5,9,10,11,13,18,23,24,26,27) function (note that there are 12 terms), using one (not two) 8-to-1 multiplexer and a 2-to-4 decoder. Do **not** use any other gates (AND, OR, XOR, etc.). Your design must contain **only** one 8-to-1 multiplexer and one 2-to-4 decoder. Complements of signals are available.

You can refer to the slides of chapter 4 of your textbook while preparing your modules and testbenches. 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!

Additional pre-lab work:

You MUST 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 by free webpack license.

- Suggestions for Lab Success with System Verilog, Vivado, and BASYS3.
- Basys3 Vivado Decoder Tutorial.
- Vivado Tutorial.
- Basys3™ FPGA Board Reference Manual (just take a look, and later use it as reference when needed).

You will need a copy of your designs and System Verilog programs with you in the lab to refer to or possibly correct and change it. The Preliminary Report will be turned in at the <u>start</u> of lab. Therefore, you must make a photocopy of it before you come to the lab, in order to use during lab.

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) <u>Write code</u>: 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) **[10]** <u>Simulate it</u>: 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) <u>Make FPGA project</u>: 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) **[15]** <u>Test it:</u> 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 (45 points)

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) <u>Write code</u>: Give dataflow System Verilog module for a 4-to-1 multiplexer.
    b) **[10]** <u>Simulate it</u>: Simulate your 4-to-1 multiplexer and show it to your TA.
    c) <u>Write code</u>: 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) **[10]** <u>Simulate it</u>: Simulate 8-to-1 multiplexer and show it to your TA.
    e) **[25]** <u>Test it</u>: Set up the circuit you designed in the preliminary work for $F(A,B,C,D,E)= \sum(2,4,5,9,10,11,13,18,23,24,26,27)$ in a new module, using a single 8-to-1 multiplexer and a 2-to-4 decoder. Using inverses of signals is allowed. Test your circuit using switches as inputs and a LED as output. Show your circuit to TA. Be prepared to answer questions that you may be asked.

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.

2) CONGRATULATIONS! You are finished with Lab #3 and are one step closer to becoming a computer engineer.

NOTES

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

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

LAB POLICIES

1. There are three computers in each row in the lab. <u>Don't use middle computers</u>, 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 <u>must</u> 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 30 minutes, you will lose that session completely.
8. When you are done, <u>DO NOT</u> 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.