CS223 Lab Assignment 1 & 2

Preliminary Work Submission Deadline:

All Sections: 24th Apr, Mon 12:00

Lab dates and times:

Section 1: 27th Apr, Thu 13:30-17:20 in EA-Z04
Section 2: 24th Apr, Mon 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

Points Distributions:
Lab 1:

- Preliminary Work: 40 pts
- Implementation:
  - Part 1: 25 pts
  - Part 2: 35 pts

Lab 2:

- Preliminary Work: 30 pts
- Implementation: 70 pts

Preliminary Work

You must do this part before coming to lab. You are going to upload the preliminary reports for both labs separately to Moodle.

Lab 1

Physical gates are built out of transistors, and require physical signals that use correct voltage levels for inputs and produce physical signals with correct voltage levels for outputs. To work correctly, the transistor circuits that comprise a gate must have connections to a voltage supply and to ground. For example, in the case of 74-series logic circuits used in this lab, the supply voltage (Vcc) must be 5 volts. In these integrated circuit packages, several gates are contained. Search Google specifying that gate number and its function (e.g. "7486 XOR gate") for pin connection diagrams, such as the pin diagram shown in Figure 1. You must have the pin diagrams for each gate you want to use, in order to do the following tasks. You can find pinout of rest of 74-series gates here: http://www.qsl.net/on7pc/datasheet/ttl7400/7400family.pdf. Other gates you need are: 7408 quad 2-input AND, 7432 quad 2-input OR and an INVERTER.

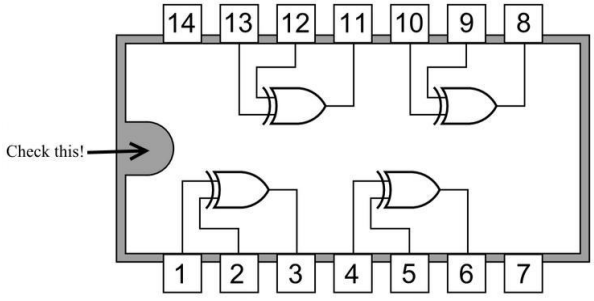

Fig 1: 7486 Quad 2-input XOR gates

1. First, read the document posted in Moodle for CS223 labs: “Circuit Schematic versus Logic Diagram”. Then, using the logic diagrams in Figure 2 & 3 (below) as a starting point, draw a *circuit schematic* of each digital circuit "you will build. This should include **pin numbers** marked on the inputs and outputs of all the gates, **part numbers** (IC’s code) of the IC package marked on each gate, plus **power and ground connections** marked on the side of the drawing.

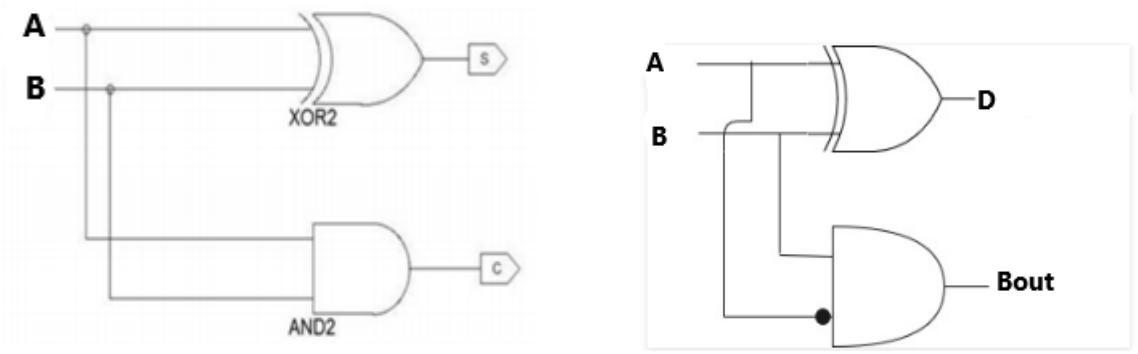

Figure 2. Half Adder & Subtractor

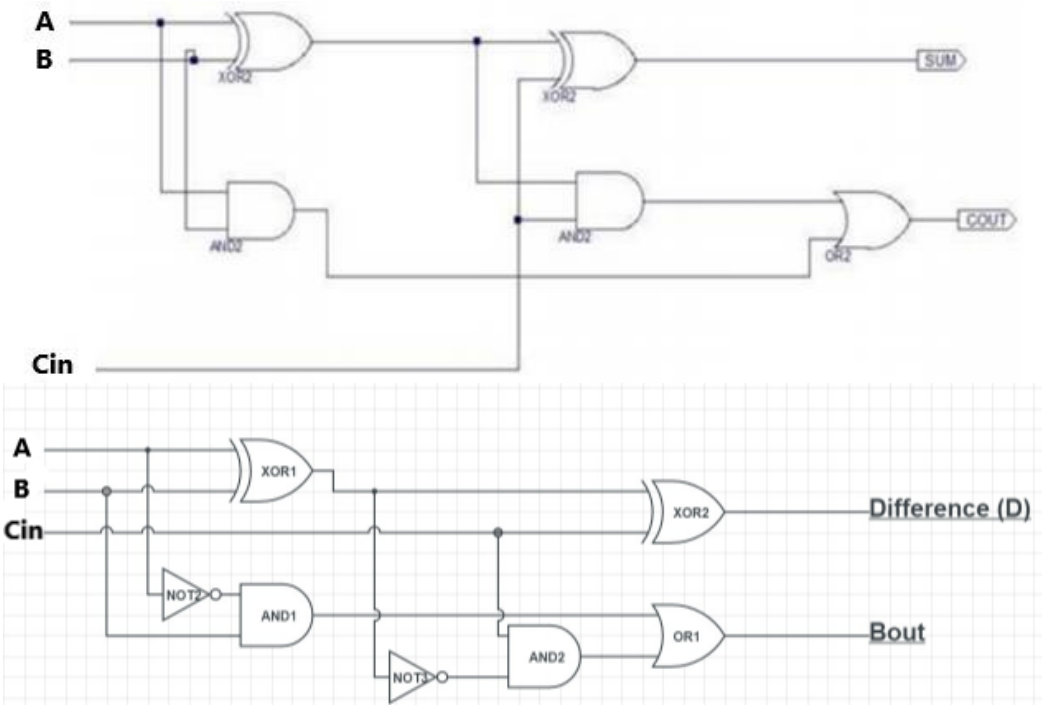

Figure 3.Full Adder & Subtractor

2. Draw a schematic for three input XOR gates using two input XOR gates.

Lab 2

For this lab, you will implement the same circuits, but this time on the **FPGA**. This lab needs considerably advanced preparation. You need to learn how to work with Xilinx's design toolset before attending the lab. In addition, SystemVerilog models and test benches 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 content of the report will be as follows:

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

(b) Behavioral SystemVerilog module for the half adder and a testbench for it.

(c) Behavioral SystemVerilog module for the half subtractor and a testbench for it.

(d) Structural SystemVerilog module for the full adder and a testbench for it. Use the half adder module you wrote in part (b).

(e) Structural SystemVerilog module for the full subtractor and a testbench for it. Use the half subtractor module you wrote in part (c).

Note that the 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 while preparing your modules and test benches.

Additional pre-lab work:

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

- Suggestions for Lab Success.
- Basys 3 Vivado Decoder Tutorial.
- Vivado Tutorial.
- Basys 3 FPGA Board Reference Manual.

Implementation

Lab 1

Part 1: Understanding and Building the Half Adder & Subtractor

1) Ask the TA or Tutor to come and check your first schematic. *Do not proceed to the next step until* you have verified that your circuit schematic is correct, and the TA or Tutor has approved it.

2) Using your circuit schematics, build the half adder & subtractor circuits step-by-step, following the Digital Circuit Suggestions given on Moodle. Connect the inputs to switches on the logic board. Connect the outputs of your logic circuit to LEDs on the logic board. Don't forget to connect +VCC power and GND ground to the VCC and GND pins on both IC packages.

3) Make a test probe by connecting another LED on the logic board to one end of a long wire, whose other end will be used to touch circuit points and "see" the logic values. A full voltage level ~5 V will cause the LED to shine brightly; a 0 V level will not light up the LED. A low light output from the LED means that the voltage being sensed is in between logic 0 and logic 1, meaning something is WRONG with your circuit.

4) Now draw the truth table for the 2-input 2-output logic circuits that you have made, and fill in the left-hand (input) side in standard binary counting order. For each row, apply the input combinations by adjusting the switches, and measure the output. Use this information to complete the truth table, filling in the right-hand (output) side.

5) Compare your measured truth table that you just obtained from the circuit, with the below one. If there are no discrepancies, then it means that your logic circuit has worked as predicted. Ask the TA or Tutor to come and verify this. When the TA or Tutor has checked your circuit, you are done with this part.

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

A	B	D	Bout
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Figure 4. Half Adder & Subtractor Truth Tables

Part 2: Understanding and Building the Full Adder & Subtractor

1) In part 1 you implemented half adder & subtractor. Here you will do the same for full-adder & subtractor, the circuits for two bits and carry-in/borrow-in bit. As you did in part 1, ask the TA or Tutor to come and check your second schematic. *Do not proceed to the next step until* you have verified that your circuit schematic is correct, and the TA or Tutor has approved it.

2) Using your circuit schematic, build the circuit, following the Digital Circuit Suggestions given on Moodle. Connect the inputs to switches on the logic board. Connect the outputs of your logic circuit to LEDs on the logic board. Don't forget to connect +VCC power and GND ground to the VCC and GND pins on all the IC packages.

Lab 2

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 computeris 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 theproject in a directory where you can find it later and erase it (at the end of the lab).*

(a) Simulation: Implement the half adder & subtractor modules in behavioral style (preliminary part (b) & (c)). Then, using the SystemVerilog testbench codes you wrote, verify the simulations that your circuits work correctly.

(b) Simulation: Implement the full adder & subtractor modules in structural style (preliminary part (d) & (e)). Then, using the SystemVerilog testbench codes you wrote, verify the simulations that your circuits work correctly.

(c) When you are convinced that your codes work correctly, show the simulation results to yourTA. Be prepared to answer questions that you may be asked.

(d) Program the FPGA: Now, follow the Xilinx Vivado design flow to synthesize, implement, generate a bitstream file, and program full adder & subtractor to Basys 3 FPGA board.

(e) Test your design: Using the switches and LEDs (on Basys 3) that you have assigned in the constraint file (.xdc), 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 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! Evenif 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 the code that you actually wrote yourself! All students must upload their code to the 'Moodle Assignment' on the Moodle page 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 the computer. Because students of other sections will work with your system too.

Recommendations

In CS223 labs, you build circuits by ICs and by FPGA. It is better to obey some simple rules to avoid damaging electronic parts or confusing yourself with debugging your circuit.

- Avoid touching IC or FPGA pins directly by your hand. Static electricity of your body can damage them permanently.
- The white board which you setup your circuit on it, is called "breadboard". Search in internet and find out how its pins are connected internally.
- Postpone connecting power pins (Vcc and ground) to last step. Check circuit connections and if everything seems ok then connect power pins.
- For easier debugging of circuits, always follow a wire color convention. For example, always use black or white wire for ground and red wire for Vcc.
- If LED's light is weak, or if the IC's package is very hot (you can touch plastic part) you have a problem in the power pin connections (short circuit, connecting Vcc wire to ground pin,...).

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 #1 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.

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. 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.