CMPE 12L Lab 1 - Fall 2014
Introduction to Digital Logic

Prof. Hans-Peter Dommel
Due: January 20, 2015 5pm
25 Points (10 Report, 15 Work)

Administrative Issues

- Please make sure you are attending the correct lab section.
- Please make sure you have a UCSC account because you will need it to submit your lab assignments.
- Your report and files must be uploaded to eCommons by the deadline above. Any demonstration to the TAs/Tutors must be done within a week of the deadline above.

Objective

In this first lab, you will learn how to use the MultiMedia Logic (MML) application in Windows to enter and simulate a logic schematic. MML is a free schematic entry and simulation tool. It is already installed on the lab machines, but if you would like to get the tool for home (you still have to come to lab), it is available at `http://users.soe.ucsc.edu/~mrg/mml.zip`.

You will use this program to build some simple circuits that give you a better intuition on how logic works. You are expected to finish part A of this lab during the first lab section and the remaining parts during the second lab section. Feel free to start the later sections early, though you might find them difficult.

Prerequisites

- Read through this **entire** lab assignment.
- Read the textbook chapter 3 up to section 3.4.
- Review the lecture notes on digital logic.

Part A: Using MultiMedia Logic

Do the tutorial for MultiMedia Logic (Help-Tutorial). If the help-tutorial doesn't work, go to `http://www.youtube.com/watch?v=hJq2gECXYWc&noredirect=1`.

This simple tutorial will walk you through building and simulating a simple circuit, save the result file as lab1_tutorial.lgi. Use Text tool to put the required comments on your schematic. Now, experiment with the logic gates by showing De Morgan's Laws in action. Save the resulting schematic as lab1_demorgans.lgi and submit it when you are done. We will not have covered this material in class yet, but it is fun to just jump right in and starting doing things. We will catch up quickly in class, trust that!

1. Connect two switches to an AND gate. Connect the output of the AND gate to an LED. See what happens on the output when you change the inputs. Figure 1 shows how to add a switch from the palette.

2. Now, add an inverter between the switch and the gate for each input. Refer to Figure 2.

3. Connect the same two inputs to an OR gate and corresponding LED. Change the OR gate to a NOR gate: Right-click on the OR gate, select Properties, and choose "Invert Output (NOR)." Refer to Figure 3.

4. Demonstrate De Morgan's Laws by running the circuit simulation and seeing that the two LEDs show the same output.

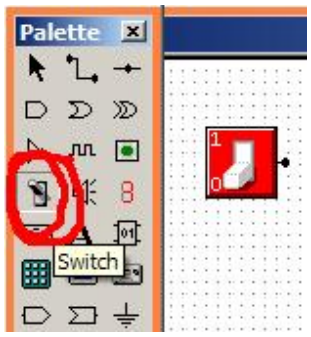

Figure 1: Adding a switch from the palette.

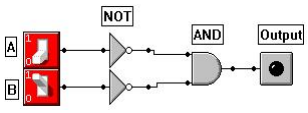

Figure 2: AND gate with inverted inputs.

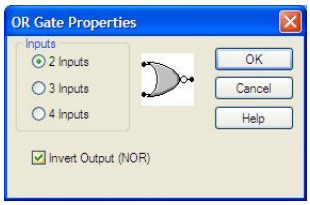

Figure 3: Inverting the output for a two-input OR gate to create a NOR gate.

Table 1: Implement this truth table

IN[2]	IN[1]	IN[0]	Output
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Part B: Implementing Functions as Sum of Products (SOP)

This part is pretty simple if you wait till after we cover this in class. Design two separate pieces of logic (all on one schematic) that implement the truth table in Table 1. The first implementation can use AND and OR gates, the second can use **ONLY** NAND gates. Do not try and minimize this logic. How many transistors does each implementation take?

Things to note for Part B:

- To create a NAND gate select the AND gate, use the gate properties like in Figure 3.
- Connect the inputs to the Switch tools and the output to an LED to verify your circuits work correctly; you can use the same inputs for both circuits but will need separate output LEDs.
- As in Part A put correct comments in your schematic.

Part C: Logic Minimization

Take your Sum of Product (SoP) expression from part B and use Boolean algebra to minimize it. Draw the final circuit on a new schematic and discuss the minimization steps you did in the write up.

Part D: Guesing Game

Now you will create a fun guessing game in logic! Create a design that allows the user to play "guess the number," where the secret number is between 0 and 3. Here are the steps you to follow:

- Use the random number generator circuit element (see Figure 4). You only need to connect two of the outputs (any two will do), the outputs are the pins on the right side of the box.
- Use a push-button switch to drive the random number generation. Buttons send a 1 when pressed and 0 otherwise. Connect the switch to "C+".
- Use two switches for user input, this will be the guess.
- Use combinational logic to test for equality; basically, is the output of the random number the same as that of the switches? Hint: Think logic AND.
- Use an LED to indicate whether the user's guess was correct or not.

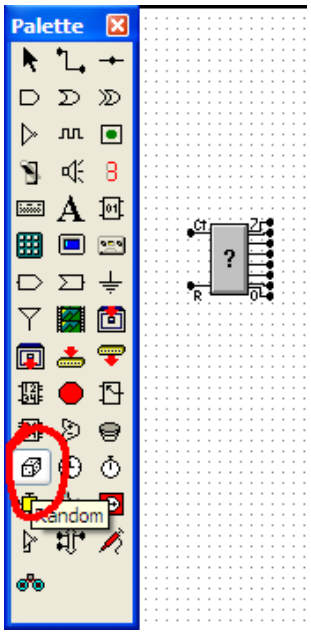

Figure 4: Random number schematic element that outputs a random 1 or 0 on each output

Lab write-up requirements

In the lab write-up, we will be looking for the following things. The lab report is worth 10 points. We do not break down the point values; instead, we will assess the lab report as a whole while looking for the following content in the report.

- Discuss the original design. How did you get to the logic design from the truth table?

- How many transistors in the original design from part B?
- Discuss the changes you made to your design.
- How many transistors in the improved design from part B?
- Why do AND and OR gates have more transistors than NAND and NOR?
- Discuss how you reduced the circuit in part B to the final circuit in part C. How many transistors in the final circuit?
- Make some sort of guess on how that random number generator works? How can things be really random in a computer with logic gates being deterministic.

The TAs/tutors will go over the basics of this lab with you and will help you out as needed. We will spend more time in lecture going over circuits.

Lab Submission

Your lab will be submitted via your eCommons account. Please log in to eCommons using your UCSC account and attach the following files to your "Lab1" assignment submission:

- lab1A_tutorial_[ucsc username].lgi
- lab1A_demorgans_[ucsc username].lgi
- lab1B_[ucsc username].lgi
- lab1C_[ucsc username].lgi
- lab1D_[ucsc username].lgi
- lab1_report_[ucsc username].pdf

Note that the final report must be submitted in PDF format.

Information on submitting assignments is available at

`http://its.ucsc.edu/ecommons/documentation/student/assignments-tool.html`.

Make sure to confirm that your assignment is SAVED and SUBMITTED before the deadline. You may resubmit your assignment an unlimited number of times up until the due date. Please enusre that your name is on all documents that you submit.

Check-off

For this lab, as with most labs, you will need to demonstrate your lab when it is finished to a TA/tutor and get it signed off. After the check off, you must still submit your files incuding any schematics and code as well as your report in the eCommons Dropbox.

If you end up needing a late check-off, we will do this (in person) using the files you submit in eCommons. You are only charged late days based on the submission time of these files and can get checked off in your next lab section.

Grading template

There are two parts to each lab grade, the work done such as creating a circuit or program and the report. For this lab the report is worth 10 points. circuits are worth 15 points. Here is the break down for the circuits, which you will get signed off by a tutor/TA:

Did you...?

- ☐ (1 pts) Have the schematics labeled (name, lab number, title, date, etc)?
- ☐ (1 pts) Part A: Do the simple tutorial?
- ☐ (2 pts) Part A: Correctly implement the logic for DeMorgans?
- ☐ (2 pts) Part B: Correctly implement the logic with AND/OR gates?
- ☐ (2 pts) Part B: Correctly implement the logic with only NAND gates?
- ☐ (2 pts) Part B: Correctly count the number of transistors in each implementation?
- ☐ (2 pts) Part C: Correct reduce the circuit down?
- ☐ (3 pts) Part D: Does the game work?

Happy Designing!!