UNIVERSITY OF CALIFORNIA, SANTA CRUZ
BOARD OF STUDIES IN COMPUTER ENGINEERING

CMPE12/L: COMPUTER SYSTEMS AND ASSEMBLY LANGUAGE

Lab 1: Intro to Logic with Multimedia Logic
Worth 30 points (25 lab + 5 report)

Lab Objectives:

In this first lab, you will learn how to use the Multimedia Logic application in Windows to do schematic entry and simulation. MML is a free schematic entry and simulation tool. If you would like to get the tool for home (you still have to come to lab though) here is a link to it http://www.softronix.com/logic.html. You will use this program to build some simple circuits to give you a better intuition on how logic works.

Lab1 Preparation:

1. Read through this Lab1 assignment.
2. Read the textbook chapter 3 up to section 3.4.
3. Review the lecture notes on Digital Logic.

Schematic Comments Requirement

- Please put minimal block comments at the top of all schematics (pages) including your name and UCSC email address on the top one. You should also include more information such as Due Date, Lab number (Lab1), lab title, your section and TA/tutor.

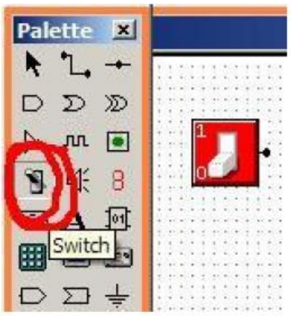
- Please comment your schematics (pages) neatly and succinctly. At a minimum label the page and the various input and output

Part A: Warm Up

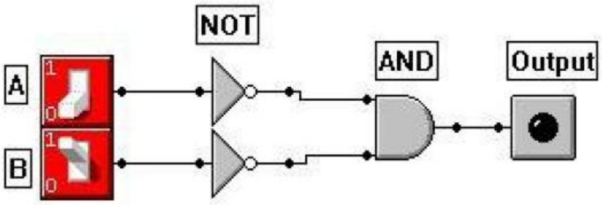
Do the tutorial for Multimedia Logic (Help->Tutorial) or look at the MML_Tutorial.pdf provided. This simple tutorial will walk you through building and simulating simple circuit, save the resulting file as Lab1.t.utorial.lgi, Use the "Text" tool to put the required comments on your schematic.

Now, experiment with the logic gates by showing De Morgan's Laws in action. Save the resulting schematic as lab1.lgi and submit when you are done, label the schematic (page) **PART A**.

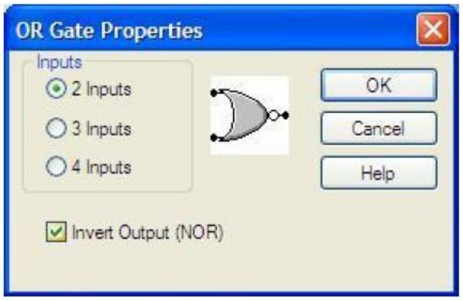
1. Connect two switches to an AND gate. Connect the output of the AND gate to an LED. See what happens on the output when you change the inputs.



2. Now, add an inverter between the switch and the AND gate for each input.



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)."



OR Gate Properties

Inputs

● 2 Inputs
○ 3 Inputs
○ 4 Inputs

☑ Invert Output (NOR)

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

Part B: Sum of Products

Start a new schematic page and label it PART B. Design two separate pieces of logic (all on the one page) that implement the following truth table. 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?

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	0

Things to note for Part B:

- To create a NAND gate select the AND gate, place it in your schematic, then right click on it and select **properties** and check the box that says "Invert Output(NAND)".
- Connect the inputs to "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 Optimization

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 page (labeled **PART C**) and discuss the minimization steps you did in the lab write up.

Part D: Guessing Game

Now you will create a fun guessing game in logic! Create a design on a new page labeled PART D 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. 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". 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.

Lab Write-up Requirements

In the lab write-up (worth 5 points), we will be looking for the following things. 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.

- Your name, email, ID, lab assignment, and section number.
- 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?
- Discussion 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 so, well, logical?

To alleviate file format issues we want lab reports in plain text. Feel free to use a word processor if you like to type it up but please submit a plain text file.

Files to Submit on Ecommons

- Lab1_tutorial.lgi
- Lab1.lgi
- Lab1_report.txt

Check-off

For this lab, as with most labs, you will need to demonstrate your lab when it is finished to the TA or tutor and get it signed off.