Lab 1: Intro to Logic Simulation

Due Friday, 12 October 2018, 11:59 PM

Minimum Submission Requirements

- Create a Lab1 folder (note the capitalization convention, include no extra characters in the directory name) that contains the following files:
    - Lab1.lgi
    - README.txt
- Commit and Push your repository
- Tag the commit that you would like to be graded

Lab Objective

This lab will introduce you to a schematic entry logic simulation program, Multimedia Logic. In this lab you will practice creating truth tables and implementing logic based on those truth tables.

Tutorial

Before starting the lab assignment, follow the tutorial listed in the Help menu.

Specification

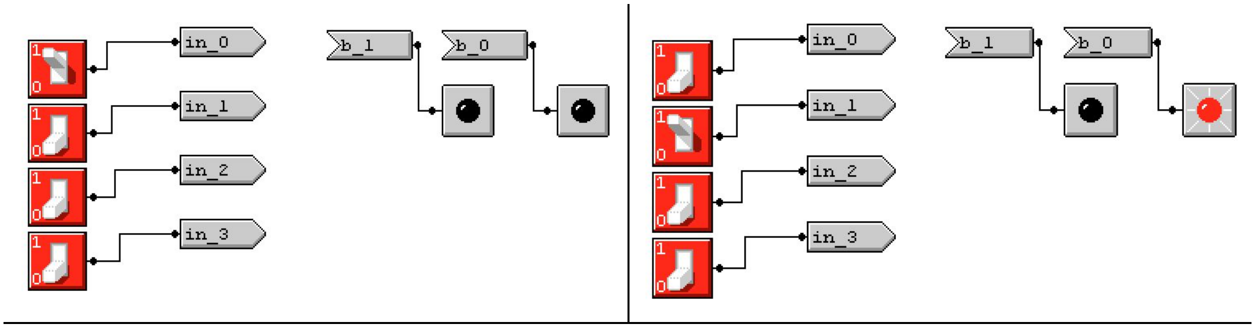
Part A

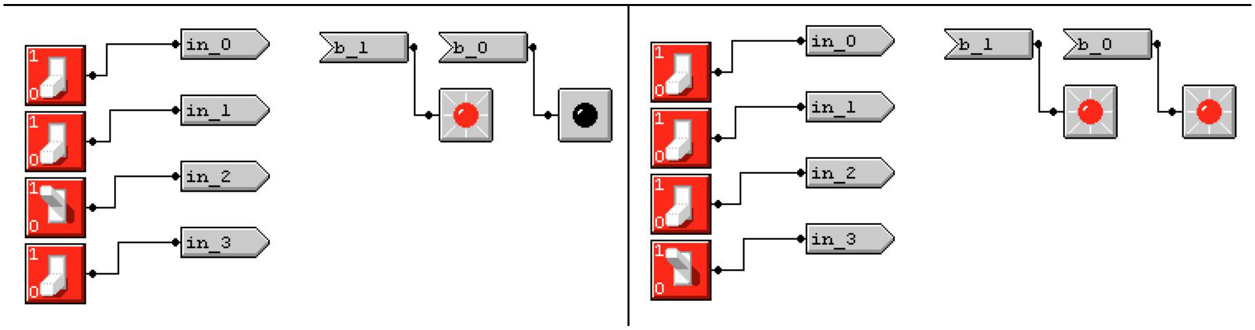
Connect the wires from the user input switches to the 7 segment display component provided by Multimedia Logic.

Part B

For this part, assume only one input switch will be on at once.

Create a truth table and implement the logic required to light the binary value of the switch number. For example, if only in_0 is on, then you should display a binary zero on the LEDs (both b_1 and b_0 are off). If only in_1 is on, then display a binary one on the LEDs (b_1 is off and b_0 is on). See the pictures below for examples.





Part C

Implement the truth table below using either the Sum of Products (SOP) or Product of Sums (POS) method. Then, implement the same truth table using only NAND gates. For extra credit, implement it once more using only NOR gates. Use an LED to display the result of your logic. Assume an on LED represents “1” and an off LED represents “0.”

in_3	in_2	in_1	in_0	c_0
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1

Template

The components on the first page of your schematic file should match the template provided on Canvas. You are permitted to change only the text fields - your name, CruzID, and descriptions of the outputs. Additional wires and logic circuits shall be drawn on subsequent pages of your Multimedia Logic schematic. Remember to rename the template file to Lab1.lgi.

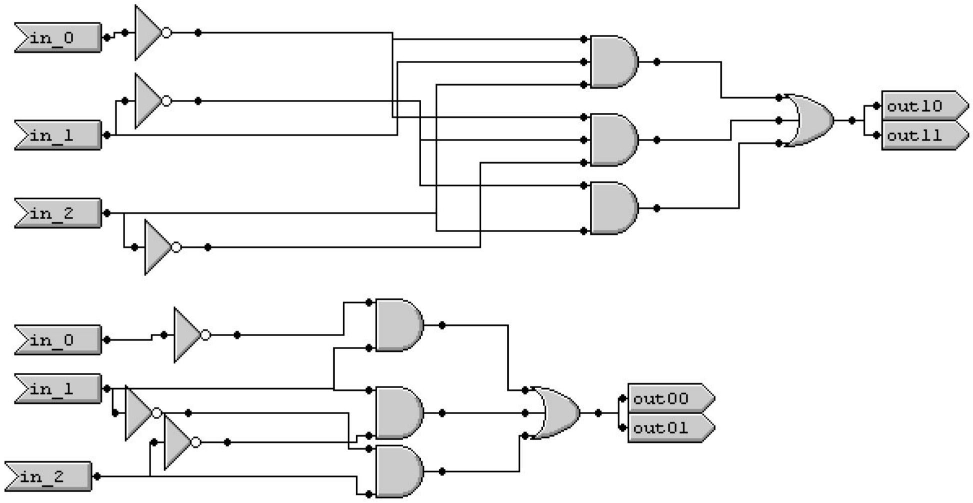
Comments

Each page of your Multimedia Logic schematic should be labeled with your last name, first name, and CruzID (the name used in your UCSC email address). Label each circuit with a description of the functionality and the part of the lab that it is for.

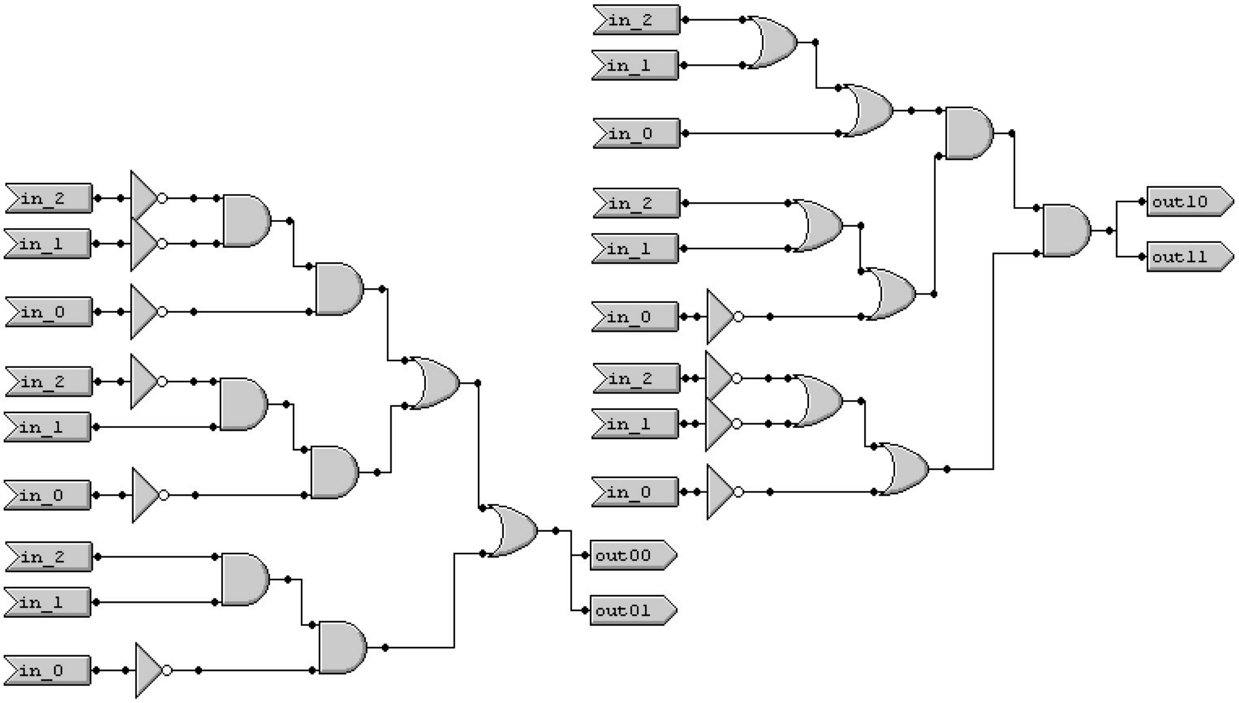
Visual Structure

Presentation of information is an important part of deliverables. Clean documentation is easy to comprehend and looks professional. Your circuits should be structured in an organized method that is easy to read and interpret. Using the “Snap to Grid” setting under the View menu makes it easy to line up components. A clean circuit uses many senders and receivers with meaningful names, and has no wires crossing over each other. Note that there may be multiple receivers for one sender. See below for examples of messy and clean circuits.

Messy Circuit Example



Clean Circuit Example

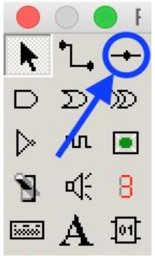


README.txt

This file should be a plain text document and contain your last name, first name, and CruzID. Use the template shown below (also on Canvas). Answer the questions listed using a minimum of eight sentences total. Put effort into your answers.

```
------------------------
LAB 1: Intro to Logic Simulation
CMPE 012 Fall 2018

Last Name, First Name
CruzID
-------------------------

What did you learn in this lab?
Write the answer here.

What worked well? Did you encounter any issues?
Write the answer here.

How would you redesign this lab to make it better?
Write the answer here.
```

Missing Wire Best Practices



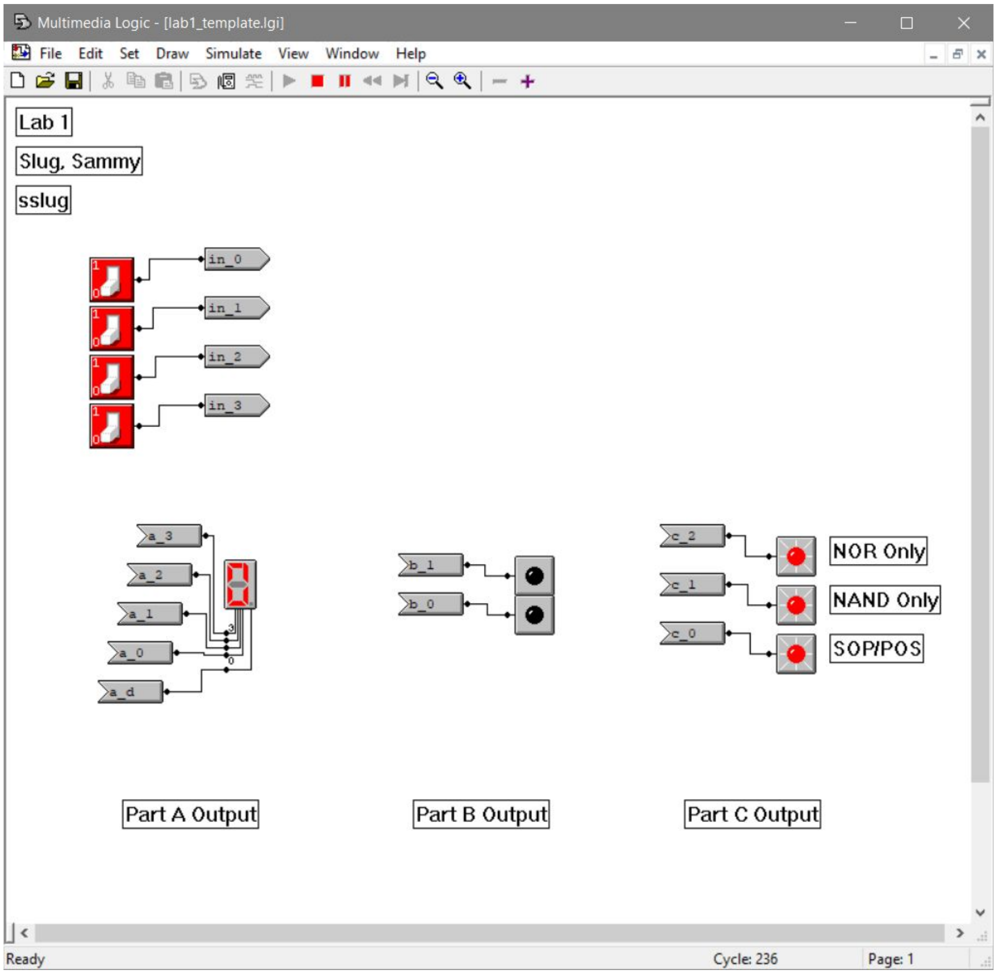
MML has a known bug which causes some wires to disappear during the git commit process. To reduce the likelihood of this occurring, DO NOT use the “Node” tool (it’s a tiny black dot located at the top-right of the tool palette). This tool is particularly vulnerable to the bug.

If this bug occurs, the grader will attempt to repair the missing wire in your file. This is only possible if your circuit is very readable. Make sure that wires do not cross whenever possible. Wire paths should be short and direct. Use receivers liberally.

Grading Rubric

```
6 pts   simulates without errors
9 pts   output matches the specification (credit for output only if circuit simulates)
        1 pts   part A
        4 pts   part B
        4 pts   part C
1 pts   complete header comments on every page of schematic and README
1 pts   useful & sufficient comments
1 pts   clean visual structure / use of white space
2 pts   README file complete with at least eight thoughtful, complete sentences
2 pts   EXTRA CREDIT for part C. Implement logic using NOR gates only.
```

Sample Output


Multimedia Logic - [lab1_template.lgi]
File Edit Set Draw Simulate View Window Help
Lab 1
Slug, Sammy
sslug
in_0
in_1
in_2
in_3
a_3
a_2
a_1
a_0
a_d
b_1
b_0
c_2
c_1
c_0
NOR Only
NAND Only
SOP/POS
Part A Output
Part B Output
Part C Output
Ready
Cycle: 236
Page: 1