CS 2110 Lab 4 Assignment: Gates, Logic, and More!

The TAs :)

February 1, 2021

Contents

1 CircuitSim

For this lab, and the rest of the class, you will need to familiarize yourself with CircuitSim. To do so, you can follow the following tutorial and implement it in the `tutorial.sim` file using CircuitSim:

- `https://ra4king.github.io/CircuitSim/tutorial/tut-3-tunnels-splitters`

Implement the logic as described in the above link in the corresponding circuit. Note that this will **not** be corrected by the autograder and is for your practice only.

2 OR Gates

2.1 What is an OR gate?

- OR

A	B	$A\|B$
0	0	0
0	1	1
1	0	1
1	1	1

- NOR

A	B	$\sim(A\|B)$
0	0	1
0	1	0
1	0	0
1	1	0

- NOT

A	$\sim A$
0	1
1	0

2.2 Building an OR gate

- Now that you know what the truth table of an OR gate is, let's try to build one in the **transistors** subcircuit of the `lab.sim` file
- Since we are using CMOS gate design, we will need a pair of p-type transistors and a pair of n-type transistors to build a NOR gate.
- We will then need one of each of the p-type transistors and n-type transistors to build a NOT gate which we will add to the end of our NOR gate
- Remember that due to some laws of physics, our p-type transistors will need to be connected to V_{dd} and our n-type transistors need to be connected to ground
- Since we want the output to be low for our NOR gate when A or B or both are high, we will need to put our p-type transistor in series so that power does not flow to the output when either is on
- Because we are using CMOS logic, we know that to complement our p-type transistors being in series, we need to connect our n-type transistors in parallel to ground
- Play around with the settings for the transistors until you have a working NOR gate
- Now you can use one p-type and one n-type transistor in series to negate the output of our NOR gate, by passing the output to both the transistors and getting the final output form the wire connecting them, making it an OR gate
- Note: You are only allowed to use the elements in the wiring tab of CircuitSim for this subcircuit. Do **not** use elements in the gates tab of CircuitSim.

3 Synthesizing Logic

3.1 ANDs, ORs, and DeMorgan's Law

In this section you will be synthesizing a logical statement into gates for testing. This circuit must be made in the `deMorgan` subcircuit of the `lab.sim` file. Feel free to use the pre-made gates in the "Gates" tab of CircuitSim.

- Logic: $\sim(\sim A\&B)|\sim(\sim B|C)$

You do not have to simplify the expression above; you can just implement it directly. See the corresponding circuits and implement the logic as described above. Do **not** change the names of the inputs or outputs. If you do, our autograder will cry!

4 Autograder

Once complete, run the autograder at a command prompt in the same directory as all your `.sim` files.

```
java -jar lab4-tester.jar
```

After you're done, take the short quiz on Canvas to get attendance credit! Ask your lab TAs if you get stuck.