Name_______________________________________ Period ______ Logic and Computation

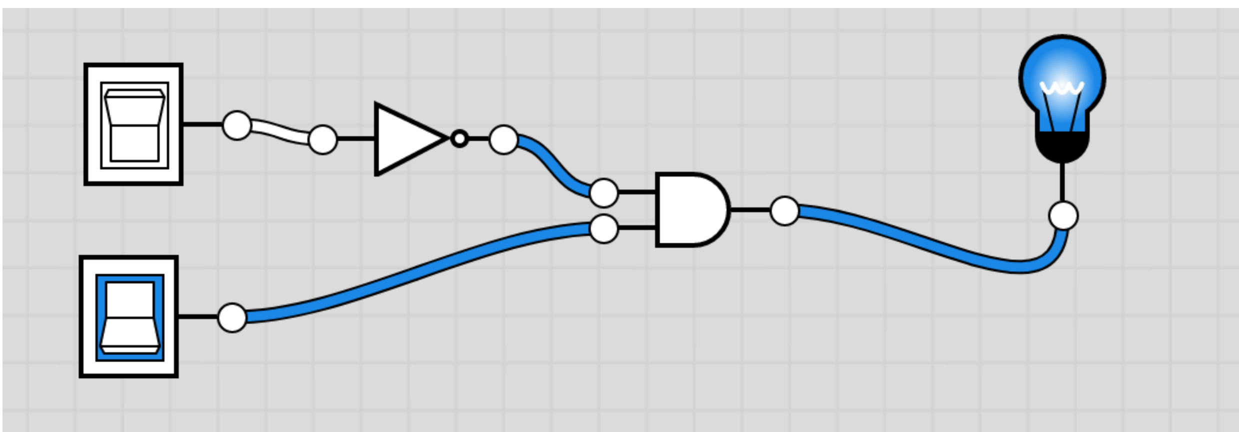
Circuits and Logic Gates Lab

Directions:

Open a new browser window and go to the website logic.ly/demo. (This is a free trial version, but there is no need to ever purchase anything.)

This program allows us to simulate the effects of different circuits built with switches, bulbs, and different logic gates. Clicking on a switch allows "electricity" to flow through its wire. If the wire is connected to a lightbulb, the bulb will turn on. Logic gates have input wires (on the left) and an output wire (on the right). Logic gates will affect the flow of electricity in the same way that the Boolean logic operators combine their inputs to output true or false.

Here is an example of a circuit made of logic gates, switches, and a bulb:



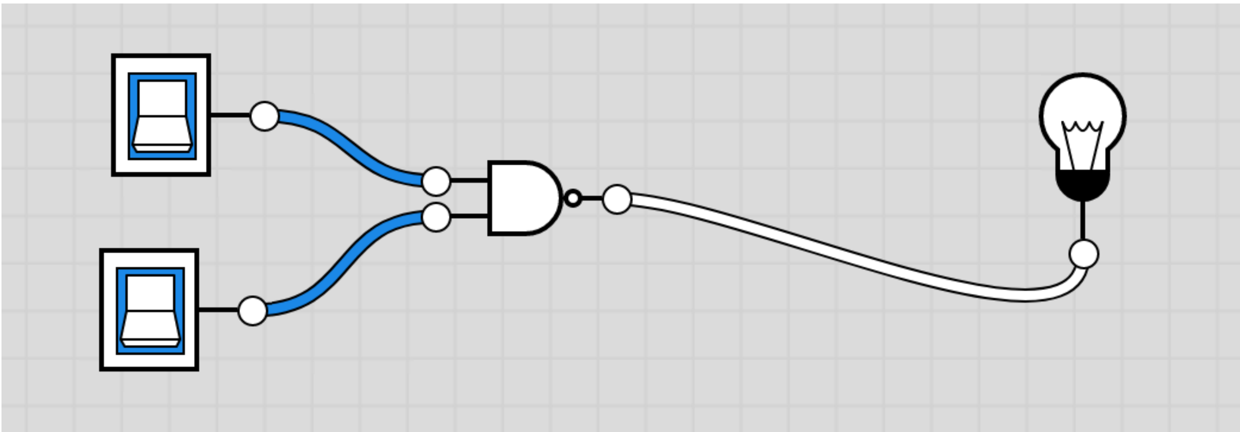
You will complete the activities below while experimenting with the effects of different logic gates to accomplish the challenges in this lab. You will turn in a google docs file with screenshots of your work. The instructions for each activity will say when you need to take a screenshot.

To take a screenshot (Mac computers): Press these keys together **Command Control Shift 4**. You will see a crosshairs cursor appear which you can use to select a rectangle on your screen to take a picture of. This will be copied to your clipboard. Press **Command v** to paste the screenshot into a google document.

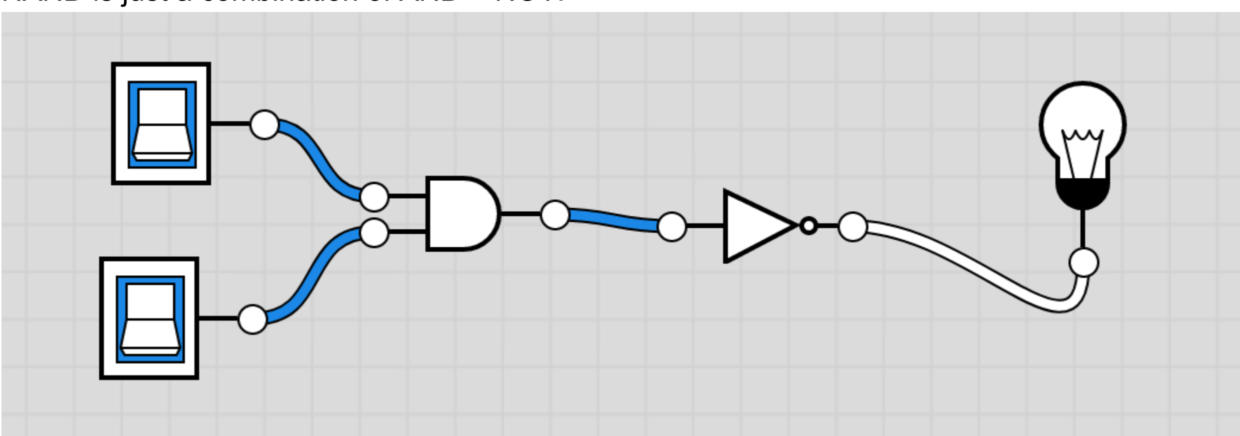
To take a screenshot (Windows computers): Press these keys together **WindowsKey Shift S**. You will see a crosshairs cursor appear which you can use to select a rectangle on your screen to take a picture of. This will be copied to your clipboard. Press **Control v** to paste the screenshot into a google document. *Another option is to use the Snipping Tool. Open the start menu and type "snip" in the search bar to find the program.*

Activity 1: Not-AND (NAND) Gate

Step 1. Connect two switches to a NAND gate and wire the output to a lightbulb. Experiment with pressing different combinations of switches. Notice that it lets electricity pass through the gate whenever both switches are not turned on at the same time. This is the opposite of an AND gate.



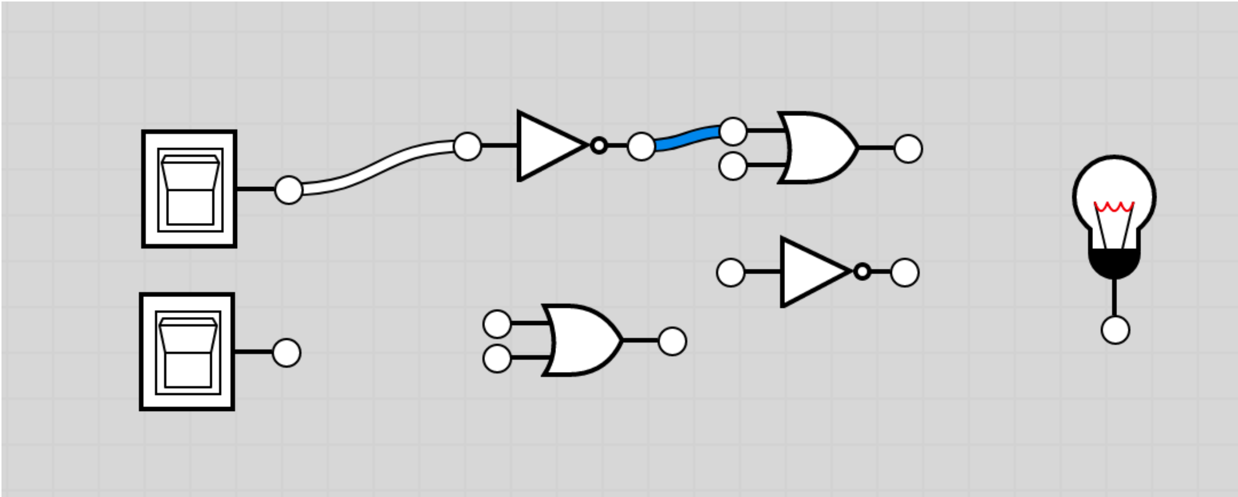
Step 2. Now delete the NAND gate and rewire the circuit with an AND gate followed by a NOT gate. Experiment with the switches and see that the effect is the same. This shows that the NAND is just a combination of AND + NOT.



Step 3. Now delete both gates. Leave the switches and the light bulb.

Your challenge is to build a circuit that works the exact same way as NAND, but uses only a combination of NOT gates and OR gates. You may use any number of these gates, but you may not use any different gates than these.

The picture below shows an example of getting started on the challenge. (You might use more or fewer gates than these and the wiring might be different than what is shown below.)

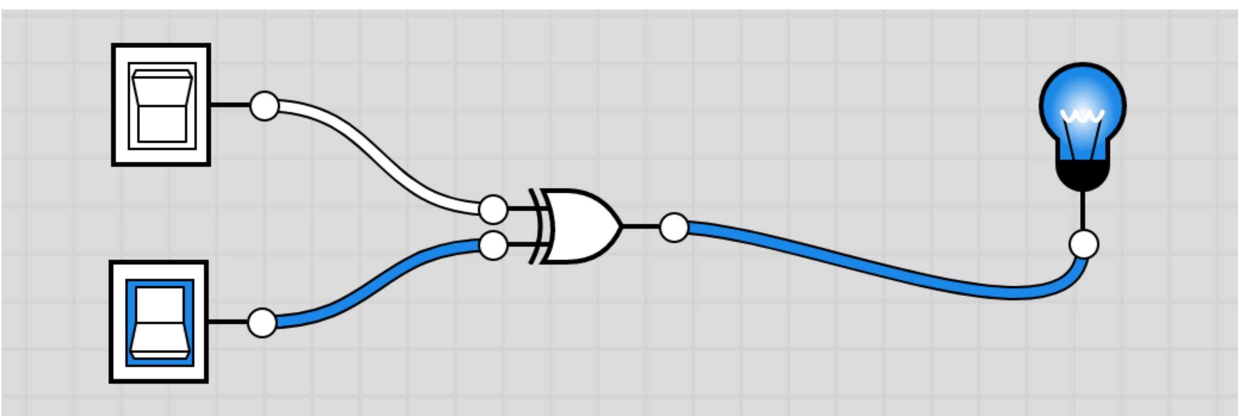


Challenge 1: Simulate a NAND gate using only NOTs and ORs. The configuration and number of gates you use will be different than this picture!

* * * **Take a screenshot** of your completed circuit that solves the challenge above and paste it into the google document that you will use to turn in your lab. * * *

Activity 2: Exclusive OR (XOR) Gate

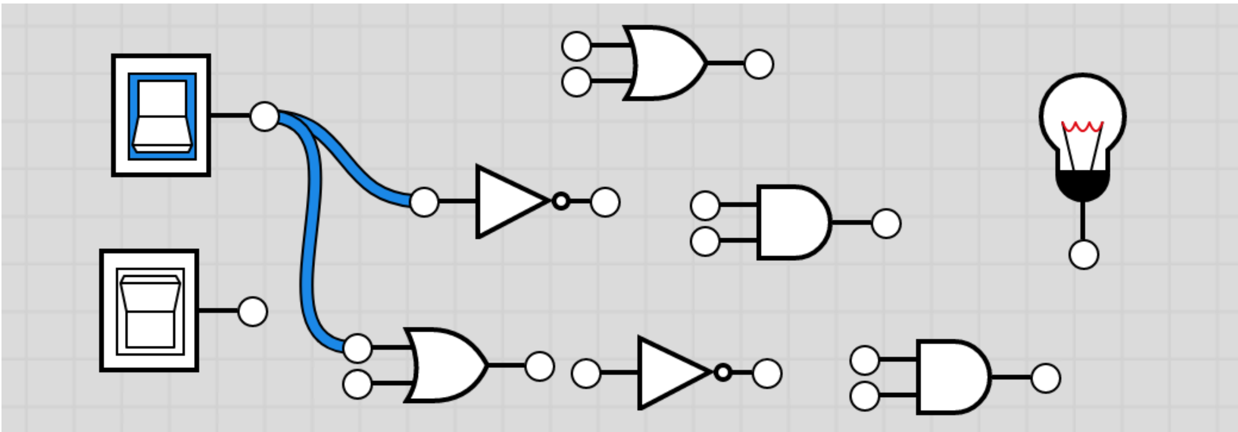
Step 1. Connect two switches to an XOR gate and wire the output to a lightbulb. Experiment with pressing different combinations of switches. Notice that it only lets electricity pass through the gate when exactly one switch is turned on.



XOR is different than the standard OR gate since the OR gate also turns the light on when both switches are on at the same time.

Step 2. Now delete the XOR gate. **Your challenge is to build a circuit using only a combination of AND, OR, and NOT gates that has the same effect as the XOR gate.** You may use any number of these gates, but you may not use any different gates than these.

The picture below shows an example of getting started on the challenge. (You might use more or fewer gates than these and the wiring might be different than what is shown below.)

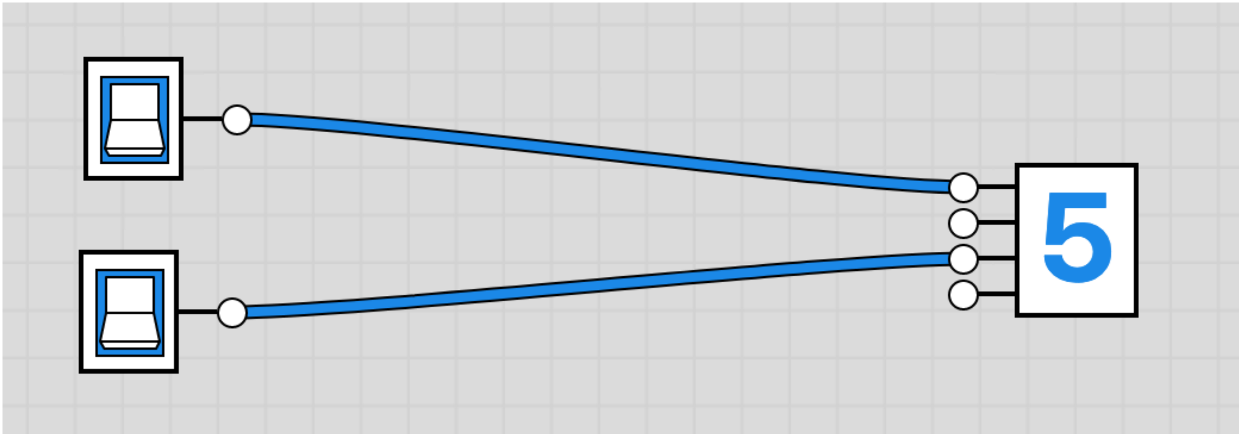


Challenge 2: Simulate an XOR gate using only AND, OR, and NOT gates. The configuration and number of gates you use will be different than this picture!

* * * **Take a screenshot** of your completed circuit that solves the challenge above and paste it into the google document that you will use to turn in your lab. * * *

Activity 3: Counting switches.

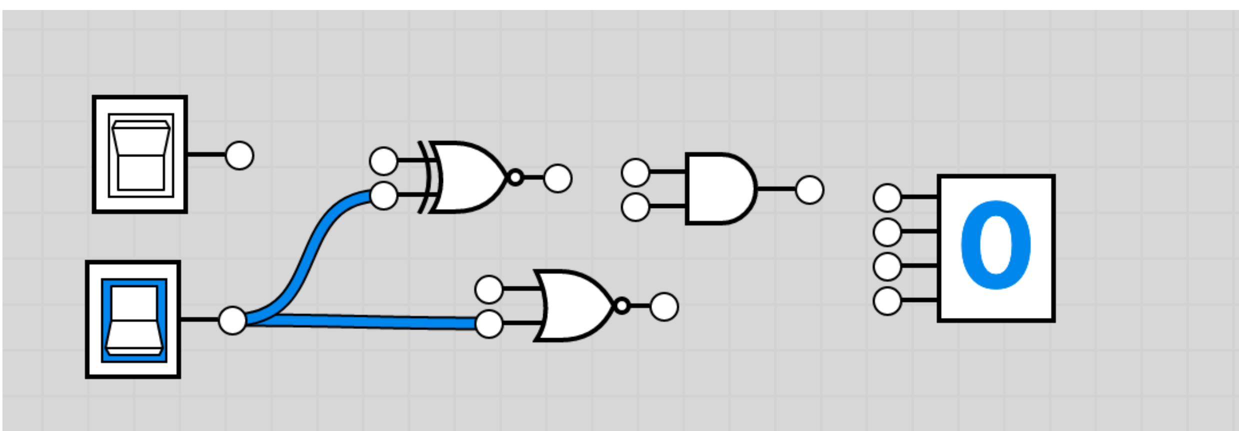
Step 1. The 4-Bit digit component has four input wires that add a different number onto the counter. The top input wire adds 1 to the current number, the second adds 2, the third adds 4, and the bottom wire adds 8. Recreate the simple circuit below that adds a 1 and a 4 to the component.



Step 2. Surprisingly, this component has a way of counting up to the number 15, but it uses letters for numbers above 9. From 0 to 15, the 4-bit digit displays these digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. The purpose of counting this way will become clear when we learn about different number base systems like binary and hexadecimal. With 4 switches (and no logic gates), can you make the component show each of the 16 possible digits from 0 to F?

Step 3. Delete all but two switches. **Your challenge is to add logic gates to the circuit to make the 4-Bit digit correctly count the number of activated switches when there are only two switches total. There is no restriction on which gates you may use.** In other words, we should see a 2 appear when both switches are on, a 1 appear when only one switch is on, and 0 appear when no switches are on.

The picture below shows an example of getting started on the challenge. (You might use more, fewer, or different gates than these and the wiring might be different than what is shown below.)



Challenge 3: Make the 4-Bit Digit count the number of active switches (out of two total switches). Any gates may be used. The configuration and number of gates you use will be different than this picture!

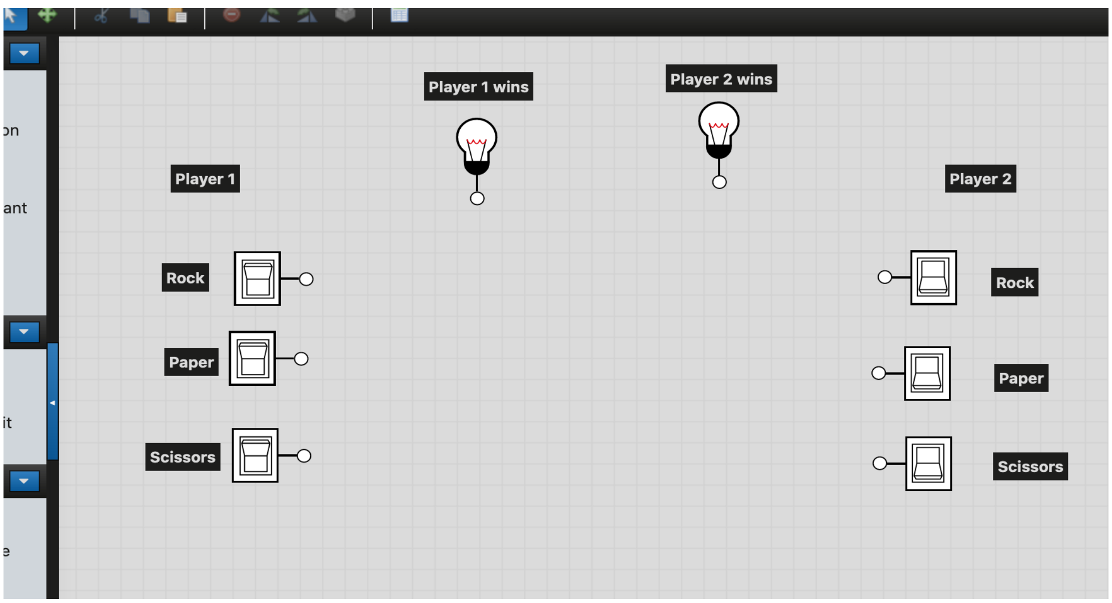
For an added challenge, get it to correctly count the number of switches turned on when there are three switches total. (This will require much more wiring!)

*** * * <u>Take a screenshot</u>** of your completed circuit that solves the challenge above and paste it into the google document that you will use to turn in your lab. *** * ***

Activity 4: Rock, Paper, Scissors game.

The ultimate challenge! You will wire up the game Rock, Paper, Scissors. The inputs for player 1 are on the left and those for player 2 are on the right. The lightbulb on the left should light whenever player 1 beats player 2. The right lightbulb should light up when player 2 beats player 1.

Arrange the six switches and two lightbulbs similar to the example shown below. (If you want, you can drag labels to name each item.) Connect logic gates to make the winning conditions for player 1 and player 2.



Challenge 4: Create a circuit that can determine the winner of a Rock, Paper, Scissors game.

As you know, rock beats scissors, scissors beats paper, and paper beats rock. You should decide what happens in the case of a tie (do no lightbulbs light up? Do both light up?)

Extra challenge: Add a third lightbulb at the bottom. This is to signal an error. The error light should turn on whenever any player tries to cheat by turning on more than 1 switch. If both players have all switches off or only 1 switch turned on, the error light should be off.

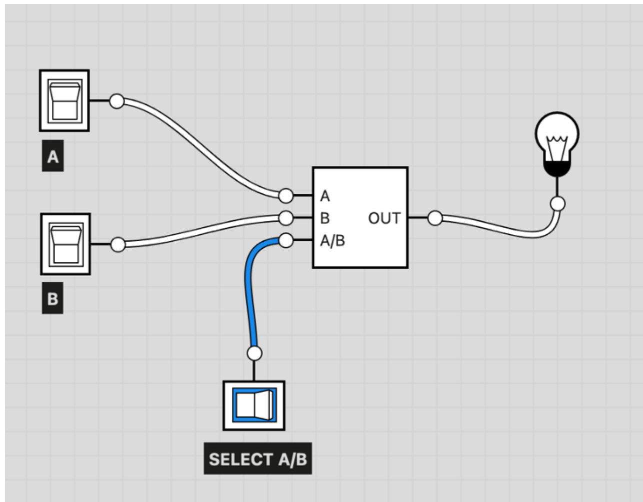
*** * * <u>Take a screenshot</u>** of your completed circuit that solves the challenge above and paste it into the google document that you will use to turn in your lab.
Please also demonstrate the working circuit to me so I can manually check that it works. *** * ***

Bonus Problems:

1. Label one switch P and another Q. Create a circuit equivalent to the logical function $P \rightarrow Q$.

2. It is interesting to know that we only need one type of logic gate to perform any computational task. It turns out that we only need NAND gates! Given any circuit made from any combination of logic gates, you can build a functionally equivalent circuit using just NAND gates. We can prove that this is true by building each of the basic logic gates out of NANDs. **Your challenge is to build circuits only out of NAND gates that are equivalent to each of the following: NOT gate, AND gate, OR gate, XOR gate.**

3. Create an integrated circuit named SELECT. When the SELECT A/B switch is turned on, the value of A appears on the bulb. When the SELECT A/B switch is off, the value of B appears on the bulb.



4. Create an integrated circuit named SELECT4. When the SELECT A/B switch is turned on and the SELECT x/y switch is turned on, the value of Ax appears on the bulb—and similarly for all other combinations of A/B and x/y.

5. Create SELECT8 in a very efficient way. (Use your previously-made integrated circuits as components.)