



Computer Systems

Week 1

Student Name: Lau Ngoc Quyen

Student ID: 104198996 / SWD00109

Overview

This laboratory session is designed to familiarise you with the Logisim Simulator and with the behaviours of simple gates, both individuals and in combination. You should spend time getting to know and trying the simulator's functions and features, and checking out the resources below.

Purpose: Demonstrate your capacity to Boolean Algebra and use Logisim to design, build and test simple digital circuits.

Task: Create and test simple digital circuits using simple gates using Logisim.

Time: This lab is due by the start of your week 2 lab.

Assessment: This lab is worth 1% (up to a maximum of 5%) of your assessment for this unit, and only if demonstrated to your lab demonstrator in the week it is due.

Resources:

- Swin tutorials:
 - [Logisim introduction](#)
- External links:
 - [Logisim Beginner's Tutorial](#)
 - [Logisim Video Tutorial](#)

Submission Details

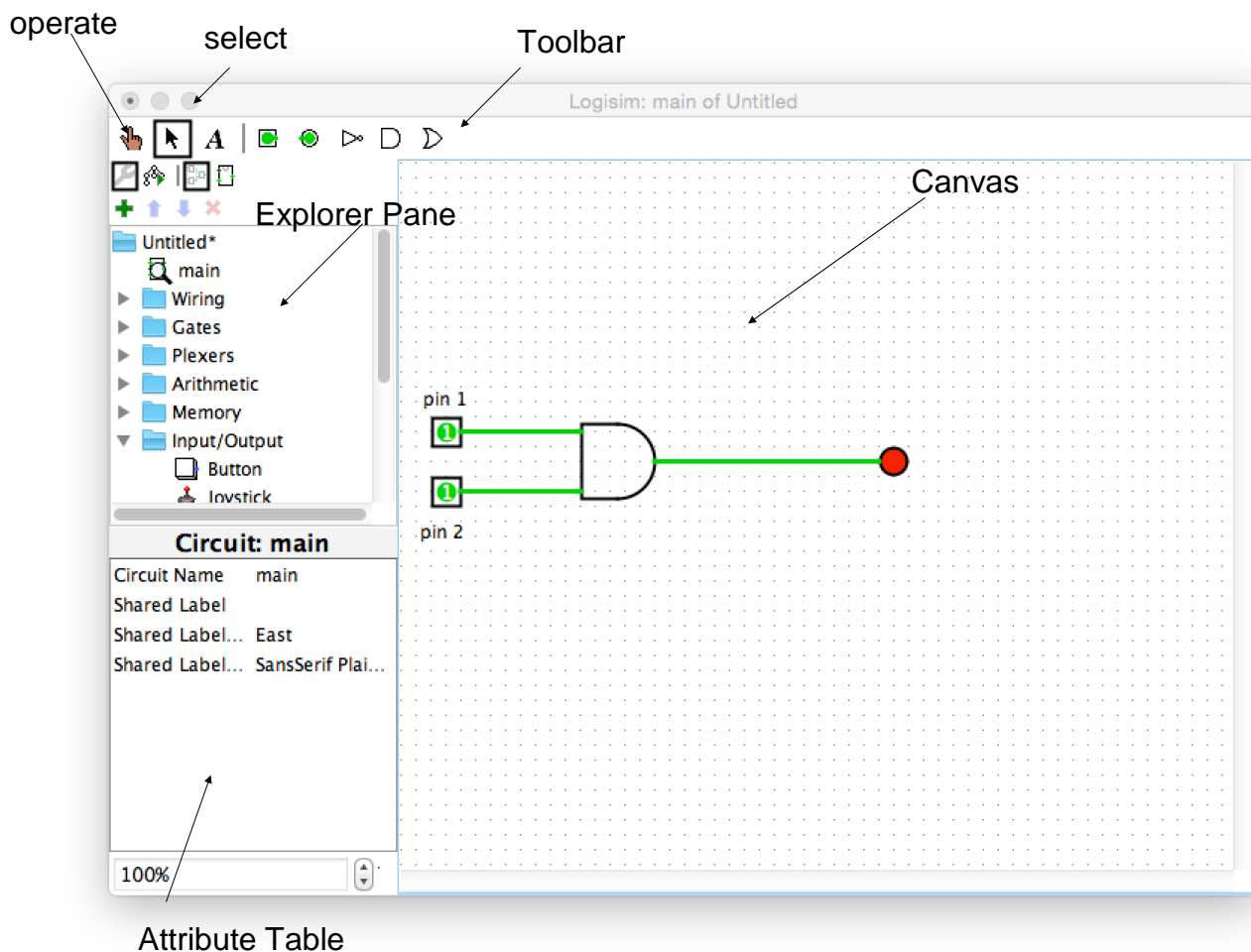
You must submit the following files to Canvas:

- A document containing all required work as described below.



Instructions

1. Start Logisim and explore its graphical interface and components Install the tools you need to get started.
 - You may also want to install Logisim on your own machine, which you can download for Windows, Mac OS and Linux from:
<https://github.com/reds-heig/logisim-evolution>
2. Lets take a look at Logisim's important controls:



3. Draw the circuit above in Logisim, including the labels. Notice the following:
 - You can select pins, gates and other components from the toolbar, or the Explorer Pane.
 - Once selected, you can click in the canvas to position them.
 - The red dot is an output LED.
 - When you select the AND gate, notice the Attribute Table which shows the current settings for the gate. You can change the number of inputs from the default 5 to 2 (as above) by changing the “Number of Inputs” attribute (4th from the top).
4. Operate the circuit to verify its correctness. To do this, click the operate (finger) icon in the top left corner of the GUI, and then click the pins to see how changing their states effects the LED.

5. Test your circuit for all possible input combinations using a truth table like the one below (where the LED being red represents 1).

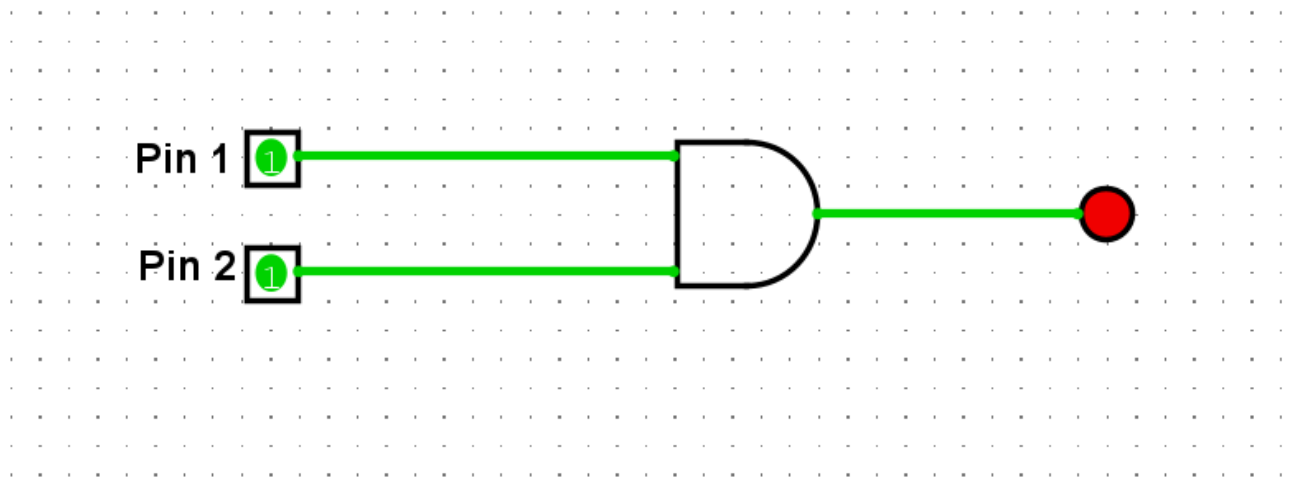


Image 1: AND circuit

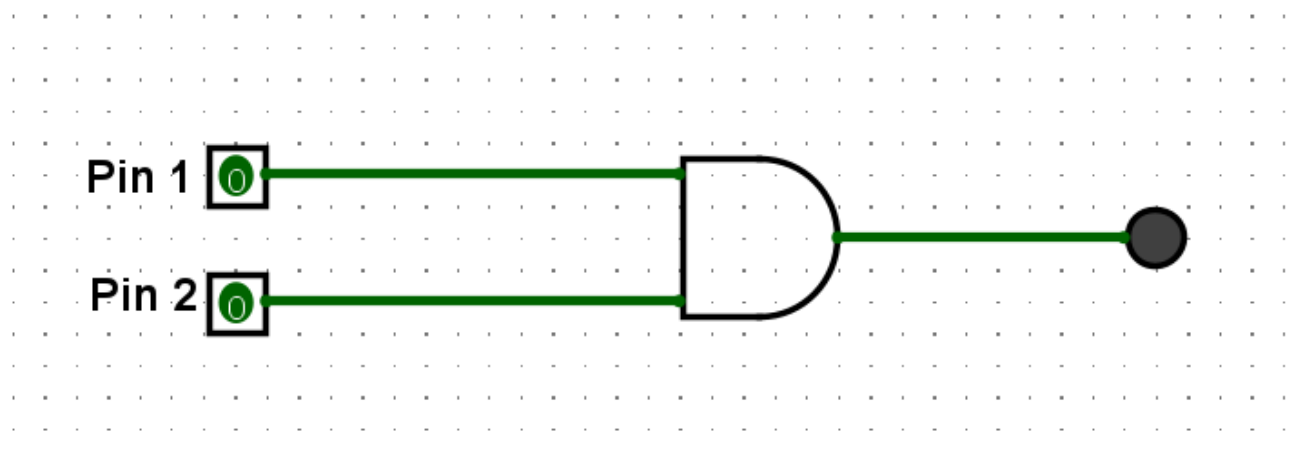


Image 2: Pin 1 = 0; Pin 2 = 0

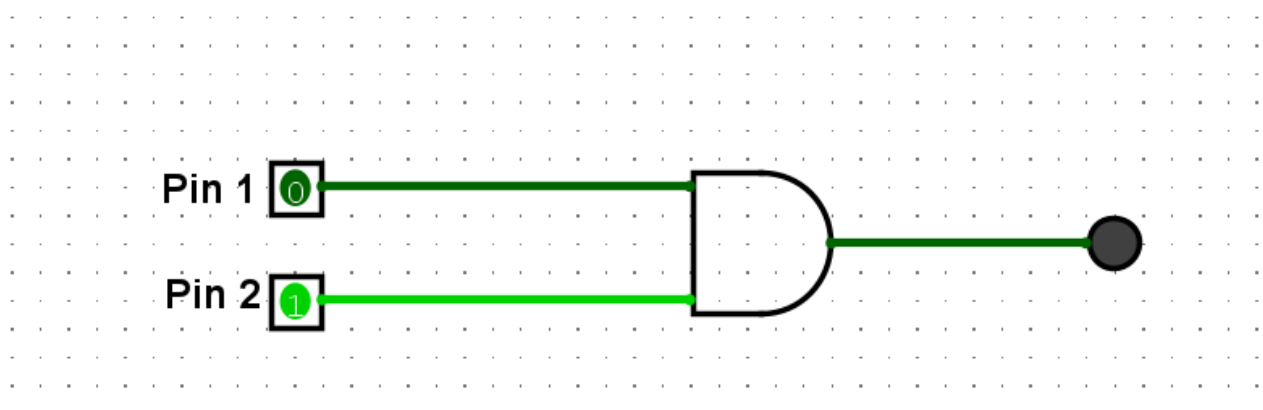
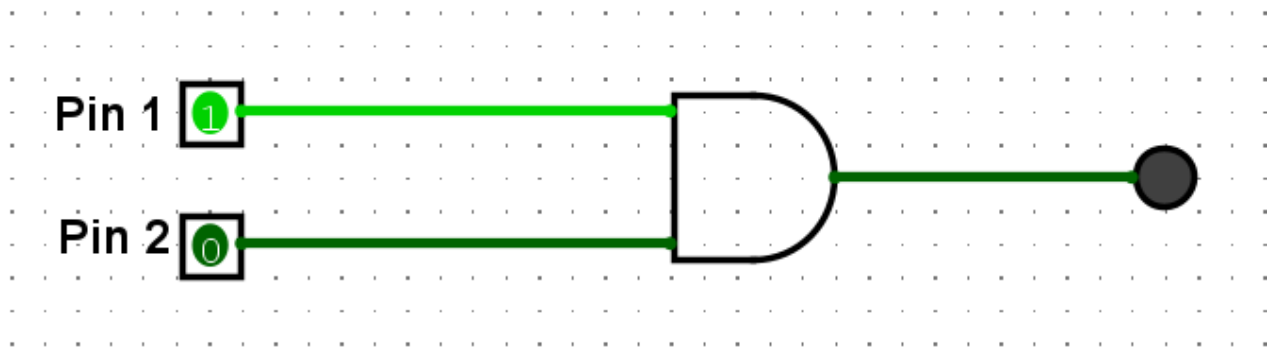
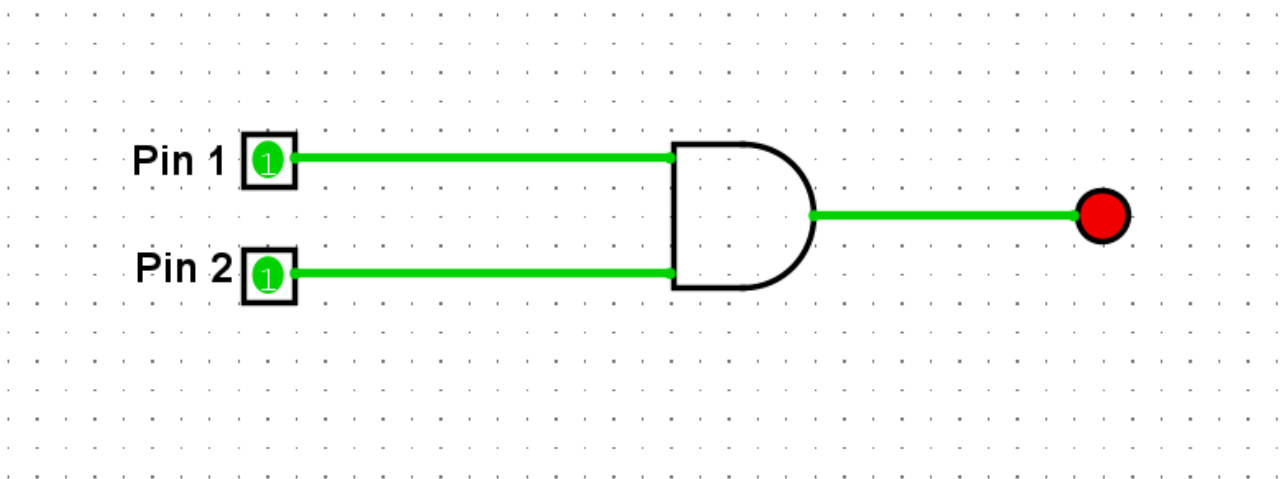


Image 3: Pin 1= 0; Pin 2 = 1**Image 4: Pin 1= 1; Pin 2 = 0****Image 5: Pin 1= 1; Pin 2 = 1**

You should include a table like this in your submission document, along with a screen shot of your circuit.

Pin 1	Pin 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

6. If you don't already have one, create a folder to store your logisim circuits (e.g., *Documents/Logisim/Lab1*). On a Swinburne computer you may wish to use a directory on your student drive or a USB storage device.
7. Save the current circuit and create a new canvas by selecting File | New from the menu bar.
8. Connect up an inverter (NOT gate), a pin and an LED to the output. **Check its correctness by filling out a truth table like the following. Add the circuit screen shot and the table to your submission document:**

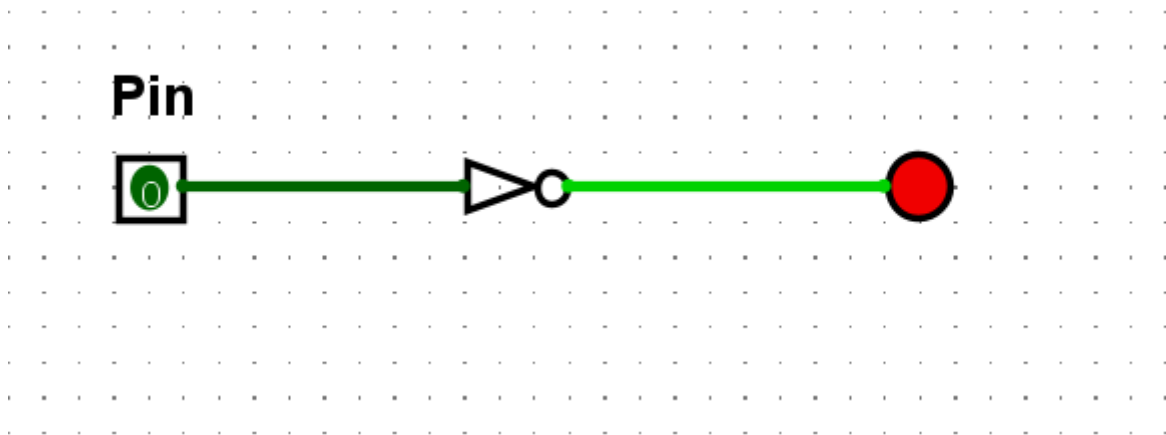


Image 6: NOT gates, Pin = 0

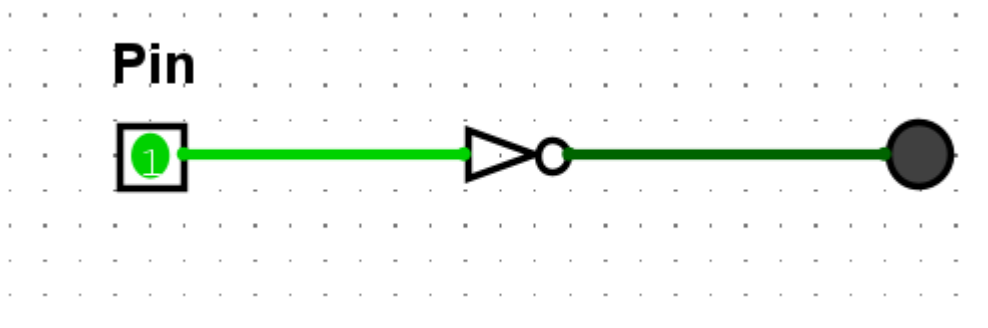


Image 7: NOT gates, Pin = 1

Pin	Output
0	1
1	0

9. Save the current circuit and create a new canvas.
10. Connect up a 2-input XOR gate, connect a pin to each input and an LED to the output. **Check its correctness by filling out a truth table like the following. Add the circuit screen shot and the table to your submission document:**

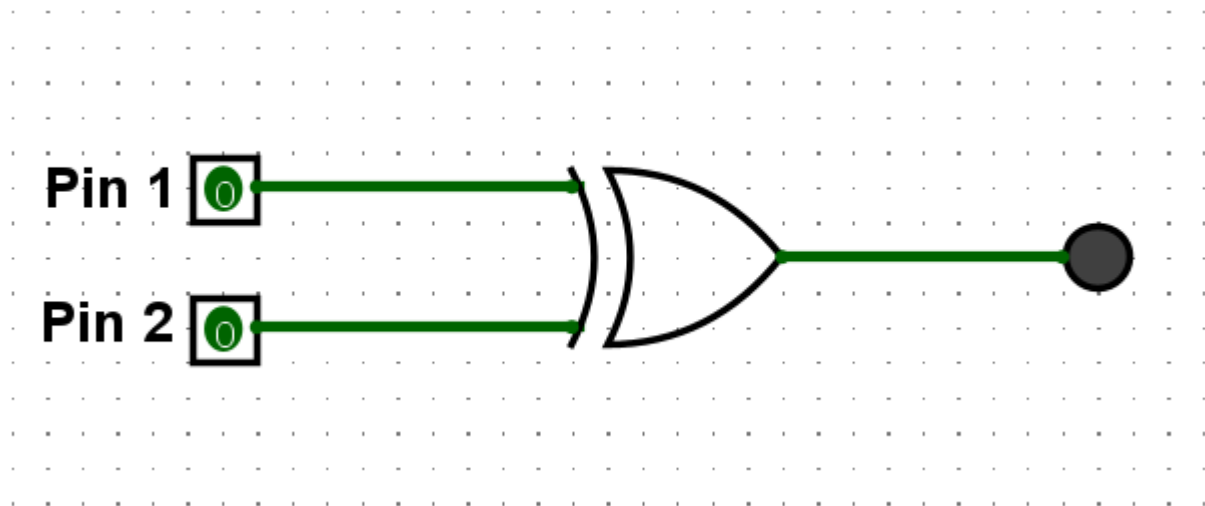


Image 8: XOR gates, Pin = 0; Pin 2 = 0

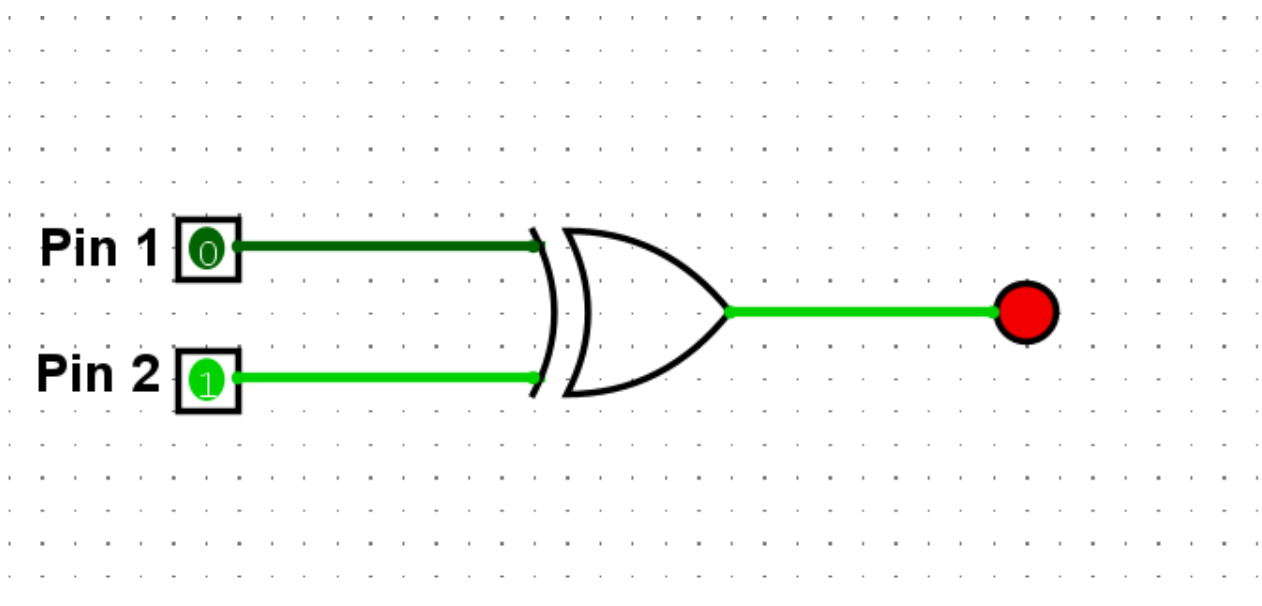
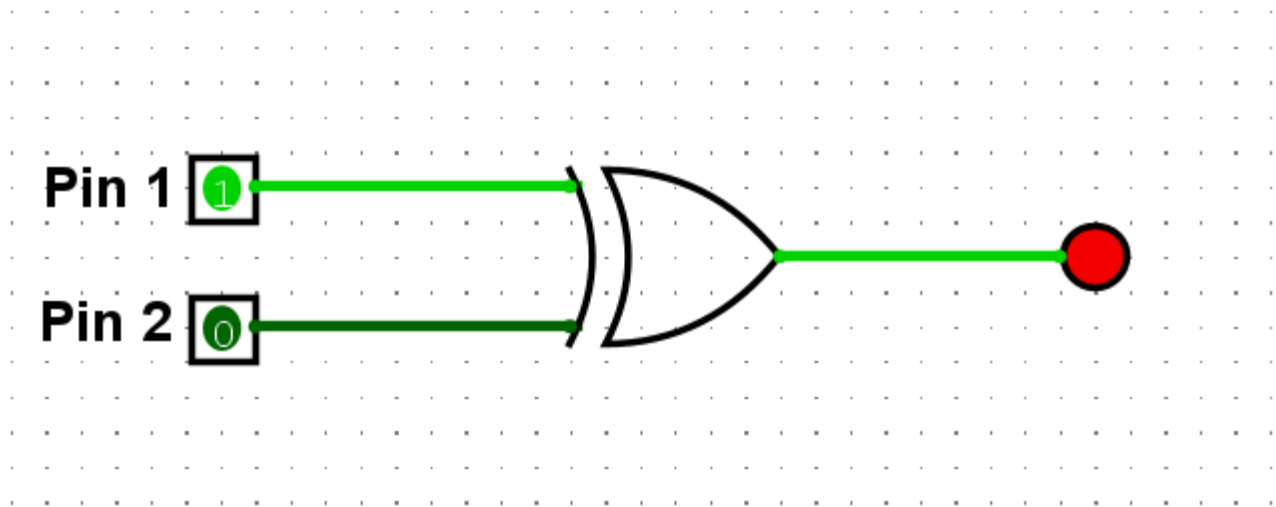
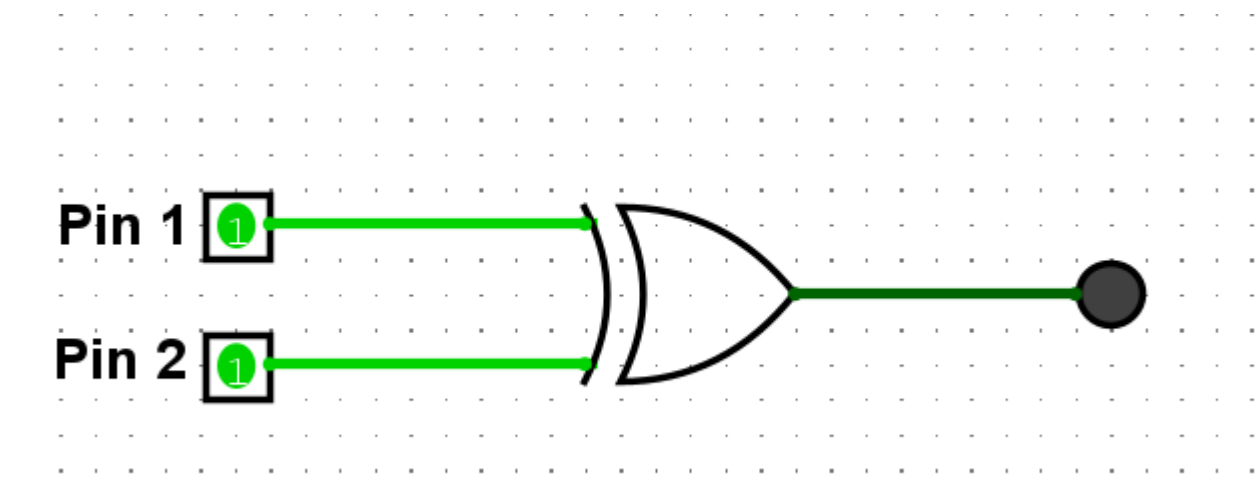


Image 9: XOR gates, Pin = 0; Pin 2 = 1**Image 10: XOR gates, Pin = 1; Pin 2 = 0****Image 11: XOR gates, Pin = 1; Pin 2 = 1**

Pin 1	Pin 2	Output
0	0	0
0	1	1
1	0	1
1	1	0

11. Name and save the current circuit and create a new canvas
12. *Using Boolean Algebra*, derive a logical expression that compares two binary inputs A and B. That is, it should evaluate to True if and only if both A and B are the same (i.e, output = 1 if inputs are both 0, or both 1).

Output= $A \cdot B + A' \cdot B'$

In this case:

- **$A \cdot B$** (**AND** gate) produces **1** when inputs are **1**
- **$A' \cdot B'$** (**AND** gate with **NOT** gate) produces **1** when inputs are **0**
- This is essentially the function of an **XNOR gate**, as it evaluates to **1** when both inputs are either the same (both **0** or both **1**).

Provide your answer in the submission document

13. Now implement your circuit from Step 12 in Logisim, and test it to ensure it works as described above.

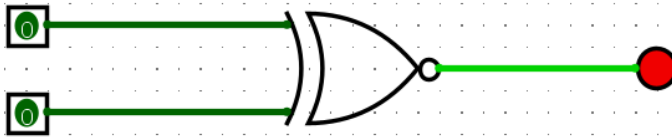
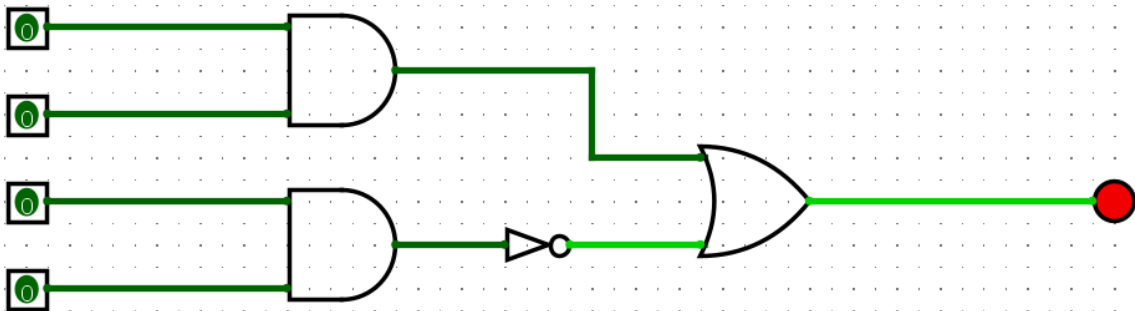
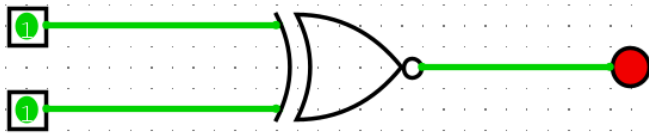
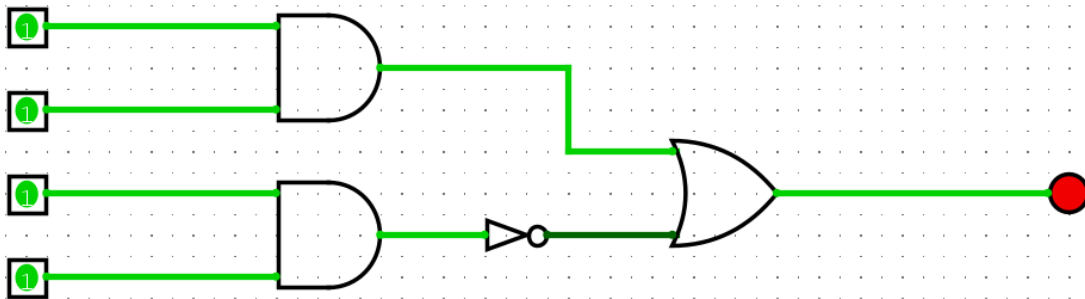
First circuit**Alternative circuit**

Image 12: Both input = 0

First circuit**Alternative circuit****Image 13: Both input = 1**

14. Save the current circuit.

15. Extend your circuit from Step 13 to do the same thing for three inputs. It should output 1 if all three input bits are either all 0, or all 1.

Check its correctness by testing it and add a circuit screen shot to your submission document:

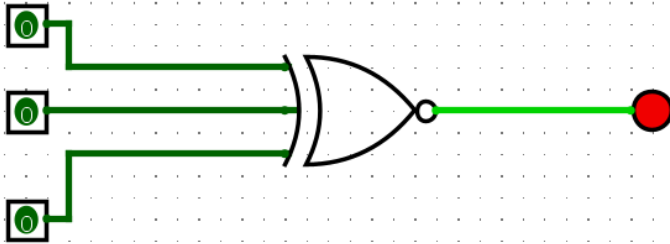
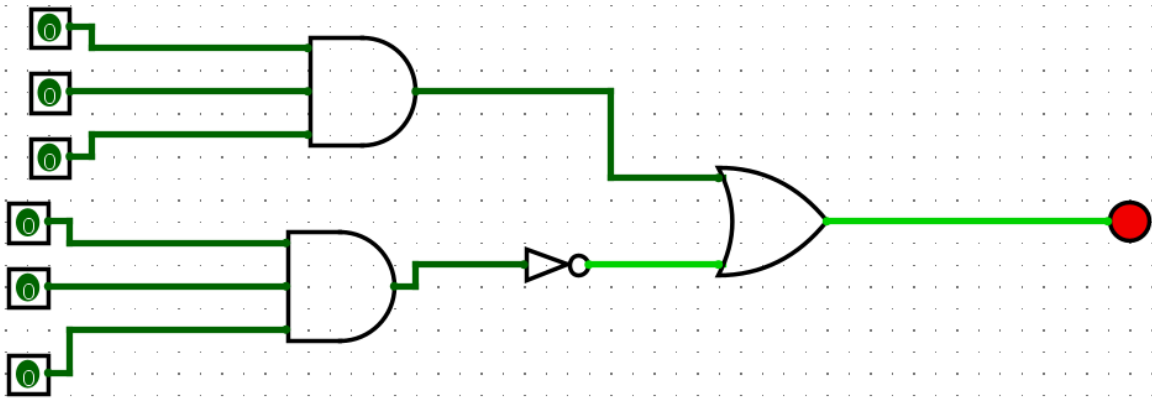
First circuit**Alternative circuit**

Image 14: All inputs pin = 0

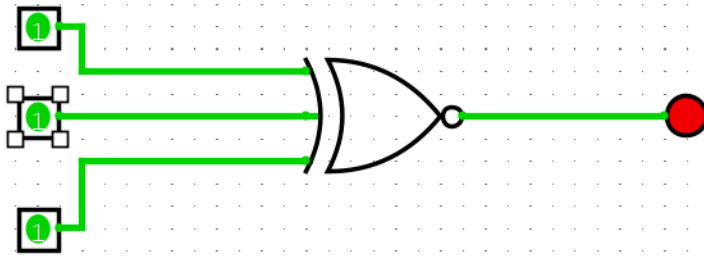
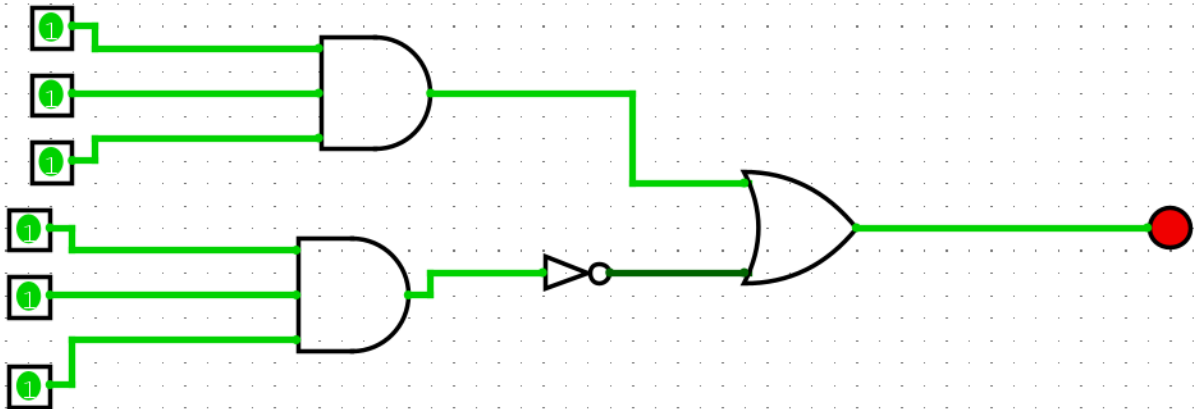
First circuit**Alternative circuit**

Image 15: All inputs pin = 1

When complete:

- Submit your answers (screen shots, etc) in a single document on Canvas
- Show your lab demonstrator your working circuits in class.