0. Pre-requisite

To complete this lab, you must have completed Lab 0. If you have not, please refer back to Lab 0 and complete it.

Note: You can store any of your lab work on a second storage device or in the cloud so that you can work from the latest copy at your convenience.

1. Lab 1 Assignment

1.1 Part 1 – Adding an `AND` Gate

Expand the `Gates` option in the ***Explorer Pane*** by clicking on the icon next to the word ***Gates***. Note that the icon may be different than what is shown dependent on your operating system. You should see several different gates appear. (See Figure 1 – [1])

1. To add a gate to your workbench canvas, simply click on the gate you want and drag the gate to the canvas (work area) with your mouse. You are not limited to a single type of gate, just select the gates you need and drag them to the workbench. In this case, you should select the `AND` gate.

2. Notice that the ***Attributes*** table for this gate has automatically expanded. If it has not, just click on your gate (on the workbench canvas) and the table will appear to the bottom left. You can easily adjust the number of inputs of the gate by changing the value for ***Number of Inputs*** in the ***Attributes*** table.

3. Change the number of inputs to 5 and observe what happens to your `AND` gate. Keep in mind that in general inputs are on the left side of the gate and the output is on the right side of the gate. Change the number of inputs to 2. This lab will be using a 2-input `AND` gate.

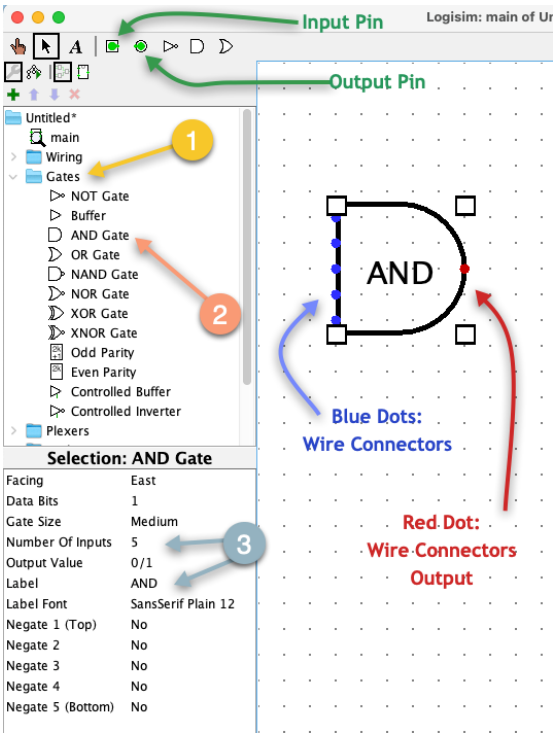

Figure 1: Logisim Workbench

4. Add a Label to your gate in the ***Attributes*** table. Find the Label attribute in the ***Attributes*** table. You can use any text/name to label the gate but since this is an `AND` gate, label it "AND1" (see Figure 1 – [3]). Labeling your gates becomes very useful when there are many gates in your design.

5. Now we need to add wires, a source (aka input source or input), and an output. To add wires, drag your mouse from the little blue dots on the left of the gate (see Figure 1). Then draw your wires an inch or so long, by dragging your mouse away from the gate.

6. Now to apply input pins (the source) to any free end of your wires. In the ***Tool Bar***, you should see .  These are the input and output pins. The pins left to right are input and output consecutively.

7. Apply a wire and an output pin to the right side of your AND gate. Note that you can simply attach the pin to the gate and then drag the pin to create a wire.

8. The last thing you need to do is to label your input and output pins. Click on one of them, then in the associated ***Attribute*** Table, enter a label. The top input pin is "x", the bottom one "y", and the output "xy". (Figure 2) Logisim Evolution is finicky about what is entered in the label. You can always use the text option to label your components. (Figure 3) All components of your logic circuits should be labeled.

9. **Congratulations!! You have just successfully added an AND gate logic circuit**. It should look like the one shown in Figure 2 below.

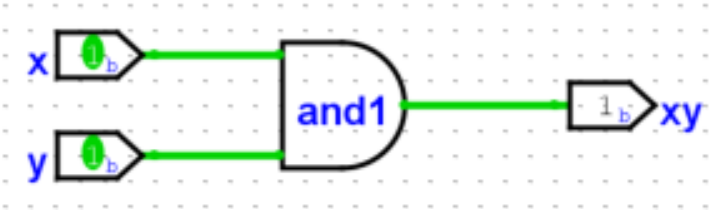

Figure 2: x AND y circuit

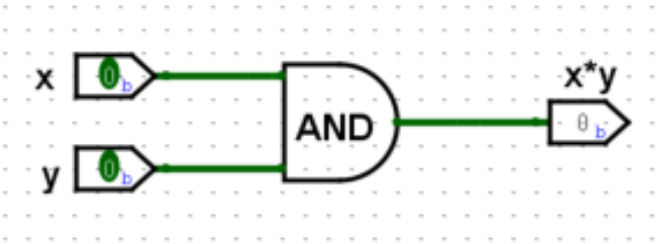

Figure 3: Circuit labeled with text tool

1.2 Part 2 – Simulate your gate

1. You may use the "poke" tool to simulate the circuit. The "poke" tool looks like a little hand and is in the left corner of the Tool Bar.
2. Before simulating your circuit, you should understand how the circuit will behave and know what the outputs will be. A good way to accomplish this is to keep a truth table, which accounts for all possible inputs and respective outputs.

   In this case, x and y are the inputs and F is the output where F is denoted by $x \cdot y$, or xy

x	y	F = xy
0	0	
0	1	
1	0	
1	1	

Figure 4: Truth table for AND gate (incomplete)

3. For this assignment, use the ***Poke*** tool to change the input values and simulate the circuit. Once the output updates for each pair of inputs, complete the corresponding row of the truth table.
4. Then use the test tool, the letter $\boldsymbol{A}$ in the tool bar, to label the circuit with your name. Take a screen shot of the circuit and save it using the class file name convention (Lab 0). You are required to submit files in pdf form. Once you have the file saved, you can then print to pdf.

Notes:

- In windows, there is a snipping tool that is extremely helpful for grabbing screenshots. On Mac you can use the screen capture (Ctrl + Shift + Cmd + 4) and then paste it on your document.
- On windows, if you don't have the option to print to pdf in your printer menu, you can download "CutePDF" writer pass and use that. Once installer, it shows up as a printer and will create a pdf for you. Get it here: https://www.cutepdf.com/products/cutepdf/writer.asp
- On Mac, you have the option to "Save as pdf" natively when you select print to any document.

1.3 Part 3 – Simulate the OR, NAND and NOR gates.

Now, let's do the same for the other 3 gates - OR, NAND, and NOR gates. As you complete each one, be sure to fill out the corresponding truth table (one for each logic gate), also take a screen shot of each logic circuit and save it as a PDF file with proper file name.

1. Repeat steps in part 1 and 2 (sections 2.1 and 2.2 above) for an OR gate.
2. Compare an Inverted-output AND to a NAND
   a. Create a circuit (inverted AND gate) consisting of a 2-input AND gate with a NOT gate on the output before the output pin.
   b. Just below this circuit, add a NAND gate. Figure 6.
   c. Simulate both circuits using the ***Poke*** tool and save the output results in a truth table for each output pin. Comparing the outputs in truth tables, what conclusion can you derive from the 2 logic circuits?

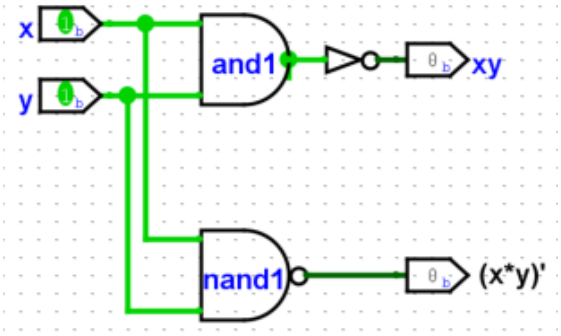

Figure 5: Inverted AND vs NAND gates

3. Repeat the step 2 above for a 2-input OR gate with a NOT on the output and a 2–input NOR gate. Does your conclusion above still hold true?

Save all the designs (logic circuit drawings) and truth tables of all parts in separate PDF files.

1.4 Part 4 – Activity

All software logic can be expressed with hardware logic gate designs. When you design a circuit, you need to know the inputs and desired outputs. With that information, you can build truth tables and derive the necessary logic.

With the logic gates we have learned so far, build a daily activity expert program, which determines the activity to do during weekends. You have the following information:

1.4.1 Inputs

There are 2 input variables:

- `IsWeekend` : values (`0`: weekday, `1`: weekend)
- `IsSunny`: values (`0`: not sunny, `1`: sunny)

1.4.2 Logic

- If today's weekend and sunny, then go hiking.
- Else if weather is sunny and today is not weekend, play tennis.
- Else if weather is not sunny and today's is weekend, play Rocketball.
- Else if weather is not sunny and today's is weekday, stay home and get some rest.

1.4.3 Outputs

- Go hiking
- Play tennis
- Play rocketball
- Stay home

2. Deliverables

1. **Part 1 and Part 2:**
   - One PDF file containing the logic circuit drawing of an `AND` gate with a truth table.
2. **Part 3:**
   - 3 PDF files (one for each `OR`, `NAND`, and `NOR` gates).
   - Each file must contain both logic circuit diagram and corresponding truth table(s).
   - Provide your answers to the questions in Part 2.3.2 and 2.3.3 in the corresponding file. i.e., in the `NAND` pdf the answer for 2.3.2 and in the `NOR` pdf for 2.3.3.
3. **Part 4:**
   - a single PDF file containing the logic circuit of the daily activity expert with the corresponding truth table with input values.

OR

4. Put all of drawings and truth tables of all parts in one PDF file but each PART should be in a separate page.
5. Submit your file(s) to the Canvas assignment.