

## Introduction to Digital Logic Circuit Simulation with Logisim

This week will be the third (and last) lab designed to introduce you to software tools that will be used for the rest of the semester. This week, you will learn to use Logisim which provides a platform for constructing and testing digital circuit designs.

The author of Logisim was nice enough to write a short, tutorial introduction to the program. Your first goal for this lab will be to complete this tutorial (with one or two little modifications described below). You should be able to do this in a fraction of the lab period.

Unfortunately, the author's tutorial does not cover several features of the program that will be important to us in future labs. So, your next goal will be to complete the tutorial component of this handout which picks up where the Logisim tutorial leaves off. This will probably take most of the remainder of the lab period.

Armed with the knowledge of Logisim the combination of these two tutorials should give you, I then want you to design circuits for two functions. The first will be an implementation of a version of the comparison operation used as an example in class. The second will be what is called a priority encoder.

The bad thing about digital circuits is that all the useful ones are already done. In particular, all of the functions for which you will design circuits for this lab are already available in the Logisim circuit library. We are asking you to reinvent the wheel! Alas, if we worked to find examples that were not part of the library, they would either be unnatural or too complicated. So please just trust us that rebuilding these common circuits will help solidify your ability to use Logisim and your knowledge of digital logic.

### The Logisim Tutorial

First, you should launch Logisim by either clicking on “Dash Home” (the icon at the top of the launcher), searching for “logisim”, dragging the logisim icon to your launcher, and finally clicking on the icon in your launcher to start Logisim or simply typing the command “logisim &” in a terminal window. The second choice may sound simpler, but the first option will make it very simple to launch Logisim in the future.

When the program starts up, it tends to take over your whole screen like Mars. Worse yet, at least on my account, the height of the window it creates exceeds that of the screen so you cannot grab the lower right corner of the window to resize it. You can get around this by holding the meta key while depressing the middle mouse button with the mouse near the lower right corner of the window.<sup>1</sup>

Once you have Logisim running, go to the Help menu and select “Tutorial”. Follow the instructions until you are about to start Step 1. Before you start “Step 1: Adding Gates”:

- Select “Add Circuit” from the Project menu.
- Name the new circuit “tutorialXor”.
- Now, complete the rest of the tutorial.

### Beyond the Logisim Tutorial

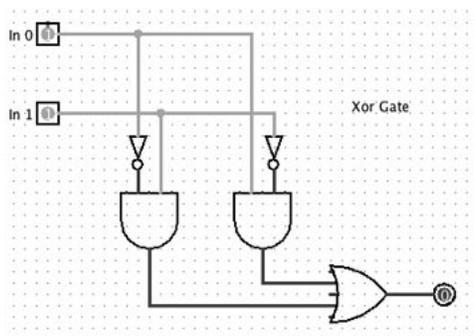
Practice makes perfect. Right?

---

<sup>1</sup>This and other useful windowing system shortcuts can be seen by pressing the key with the flag icon right next to the meta key.

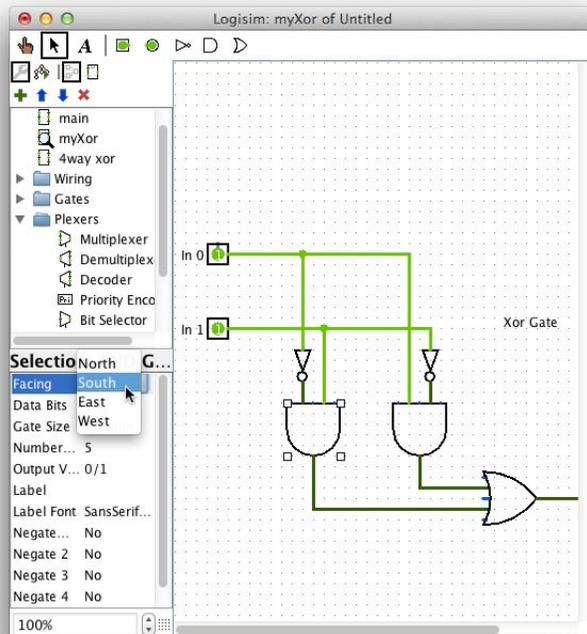
The first thing we want you to do beyond the built-in tutorial, is do it again, but with a slight twist.

First, when you get to step 1, add a new circuit as described above but this time name it “xor2”. Next, in Step 1 of the tutorial, the author shows a diagram of the circuit he hopes you will construct and then follows the layout of that diagram throughout the tutorial. We want you to try a different layout this time. What we have in mind is shown below:



In the abstract, the layout of a circuit is not important as long as the correct components and connections are included. In practice, layouts that involve connections that make multiple turns on the way from source to destination tend to be difficult to read. It is often cleaner to arrange components so that all signal lines make one left or right turn on the way from one layer of gates to the next. The text refers to this as PLA style. We think you will find the remainder of this lab easier if you follow this style. The alternate layout shown above uses it.

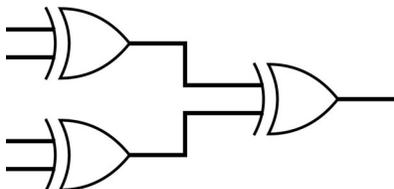
To complete a layout in this style, you will need to tell Logisim to rotate some of the gates 90 degrees from their default orientation. You can do this as soon as you place a new device or whenever it is selected. The left margin of the Logisim window is divided into two areas called the “Explorer Pane” and “Attribute Table”. The attribute table appears below the explorer pane. When a gate is selected, the first item in the attribute table should be “Facing”. To rotate the selected gate, depress the mouse on the current orientation and select a new orientation from the menu that appears as shown below.



## Using a Subcircuit

One of the good features of Logisim is that you can use a circuit you have already built as a building block in another, more complex circuit. This is why we had you select “Add Circuit” each time you started one of the two versions of the exclusive or circuit. Now, you will use the xor2 circuit you just created to build a 4-input exclusive or circuit.

On pages 70 and 71, our text uses the exclusive-or as an example of how circuits can be constructed using multiple levels of logic rather than by simply building a sum of products implementation based on the truth table. When applied to multiple inputs, an exclusive or circuit should output one when the number of input signals that are one is odd. Clearly, a circuit for this function could be constructed in sum-of-products form. It is also possible, however, to simply use a tree of two-input exclusive or gates. For example, the circuit shown below implements exclusive or of four inputs.



We want you to build such a circuit, but instead of using Logisim’s built-in implementation of exclusive or, you should use your own xor2 circuit.

To do this:

- First select “Add Circuit” from the Project menu and name the new circuit “xor4”.
- The names of all of the circuits you have created, “tutorialXor”, “xor2”, and now “xor4”, should be listed under the Logisim “Explorer Pane”. Click once (ONLY ONCE) on the item

labeled “xor2”. Now click once somewhere in the blank canvas associated with your new circuit. Logisim should place a little icon that looks like this in the circuit:



This icon represents a copy of your “xor2” circuit.

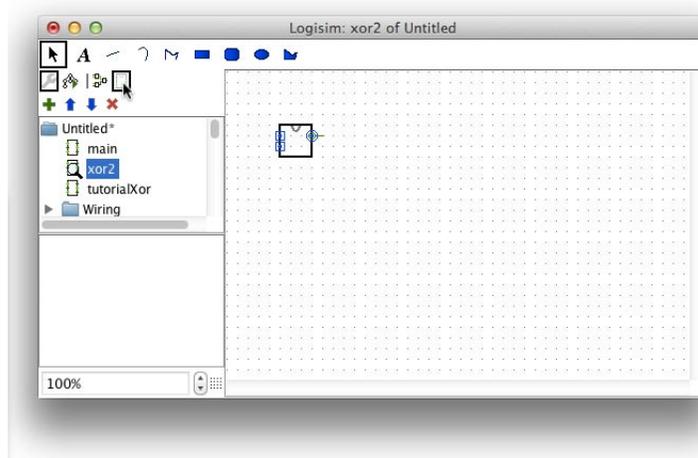
Basically, just as the Logisim tutorial taught you to place simple gates and pins by clicking on the buttons at the top of the window, you can place a copy of any circuit found in the explorer pane in the current circuit by clicking once on the desired subcircuit in the explorer pane and then clicking on the current circuit’s canvas.

Right now, you only have one copy of your xor2 circuit in your xor4 circuit. The circuit shown above uses 3 2-input exclusive or gates to build one 4-input exclusive or. So, to build a similar circuit, you should:

- Add two more copies of your xor2 circuit in roughly the same orientation shown in the circuit above.

There is nothing about the little icons that represent copies of your xor2 circuit that makes it clear what they are. We can fix this by adding a label to the icon.

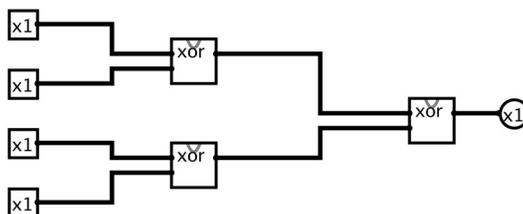
- First, double-click on the item for “xor2” in the explorer pane. This tells Logisim that you want to edit this circuit. The contents of the canvas should be replaced by the circuit diagram you constructed earlier for xor2. If you wanted to, you could revise your xor2 circuit in any way you wanted at this point. Don’t!
- Instead, click on the rightmost icon on the row of four buttons just below the row of buttons where you find the icons for and gates, or gates, and pins. The contents of the canvas should be replaced by an editable version of the icon for xor2 as shown below:



Now, the only thing you can change is the way this icon will appear in other circuits that use xor2.

- Click on the text tool (the big “A” in the second position of the top row of buttons).

- Click in the middle of the image of the icon for your subcircuit. A text entry box should appear.
- Type “xor” into the box.
- Now, double-click on the item for xor4 in the explorer pane. This takes you back to the canvas where you can edit xor4. You should find that all three copies of the icon representing your subcircuit now contain the label “xor”.
- Use the features you learned earlier to complete the circuit by adding wires and pins as shown below:



- Test it! The output should be 1 whenever the number of 1s in the input bits is odd.

## Buses and Splitters

So far, you have built a two-input exclusive or gate and a four-input exclusive or gate. Care to guess what comes next? Yes, at this point you should:

- Select “Add Circuit” from the project menu once again and name the new circuit “xor8”.

The circuit you are about to build will take 8 input signals. We could provide a separate pin and separate wire for each of these 8 signals, but it should be clear that as the number of inputs to a circuit increases, this becomes unattractive. As an alternative, Logisim lets us create a *bus* to carry a group of related signals from a source to a destination. A bus is basically a collection of wires that are routed in parallel.

Logisim automatically creates a bus when you try to drag a wire between two components that input and output multi-bit signals. Therefore, we need to start by including two such components in xor8.

- Start by adding a pin near the left edge of the blank canvas for the xor8 circuit.
- Next, add two copies of your xor4 circuit directly above/below one another about three inches to the right of the pin.
- Finally, connect your single pin to the top input on the top xor4 gate.

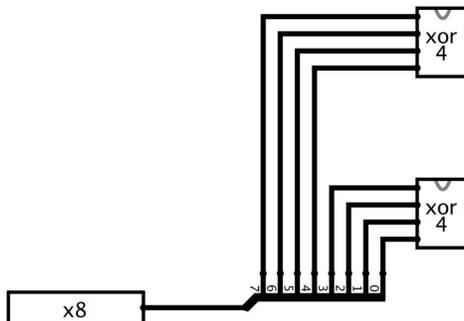
All of this should work fine since the the pin you have created produces a single output signal. Let us change this (even though it will temporarily break the circuit).

- Click once on the pin so that its attributes are displayed in the bottom left corner of the Logisim window (the attribute pane). The pin’s third attribute should be “Data bits” and it should initially be set to 1.
- Change the pin’s data bits to 8.

The pin's icon in the canvas will now change to a box in which you can set eight different bit values to serve as input to an eight-input device. This is what we want, but it makes Logisim very upset because the pin is currently connected by a wire to another device that expects four separate one bit inputs instead of an eight bit input. To fix this:

- Delete the wire connecting the pin to the xor4 gate.
- Add a “Splitter” between the pin and the two xor4 gates. The splitter can be found within the Wiring folder in the explorer pane. Once you find it, just click once on its icon and then click once in the canvas to place it.
- Two of the attributes for this new Splitter will be “Fan in” and “Fan out”. By default, they will both be 2. You should set them both to 8. Setting fan in to 8 indicates that this splitter expects its input (which will be connected to the lower left corner of its icon) to be an 8-bit bus. Setting the fan out to 8 indicates that the splitter should divide this input up into 8 outputs so each output will be a separate signal.
- You may also want to change the splitter so that it is right-facing and oriented northward.
- Drag a wire from the 8-bit pin to the input of the splitter. Since the splitter expects an 8-bit input, Logisim will be happy with this.
- Drag wires from each of the outputs of the splitter to the inputs of your two xor4 gates.

At this point, your circuit should look something like:



Now, to finish things off:

- Add an xor2 component to the right of your xor4 gates. Connect the outputs of the xor4 gates to the inputs of the xor2 gate. Connect an output pin to the output of the xor2 gate.
- Switch to probe mode and test your circuit.

Finally, let's make a “main” circuit that tests your xor4.

- Double-click on main in the explorer pane to begin editing the main circuit. This is the circuit you would have been editing right at the beginning of the Logisim tutorial if we had not kept telling you to add subcircuits.
- Add an input probe, an xor8 gate, and an output probe to the canvas.
- Change the input probe's data bits attribute to 8.

- Connect the input probe to the input of your xor8 and the output of your xor8 to the output pin.
- Test this circuit.

Note that since the input pin in your xor8 circuit expects 8 input bits, you can/must connect an 8-bit bus to the input pin of this subcircuit when it is used in another circuit.

## Exercises

Now, you are ready to get some experience building circuits with Logisim on your own. When you are done, you will submit these circuits as your work for this week's lab. To prepare things so that you can submit them together in one file:

- Go to the File menu and select "New".
- Go the the File menu again and select "Save as". Save your work in a Lab3 subdirectory and give it the name Lab3Circuits.

When your work is complete, this project should contain several subcircuits implementing the functionality described below and one "main" circuit containing two simple circuits designed to test the two top-level subcircuits you have constructed. By next Monday, you will have to submit this project file and a file containing a printout of the diagrams of all of your subcircuits. We explain how to submit these files at the end of the handout.

## Comparators

In class, we have talked about circuits that determine whether one  $n$ -bit binary number is less than (or greater than) another. For  $n = 2$ , we have seen how to construct a circuit for this task by deriving a sum or products expression from the truth table. As your first task beyond the tutorial component of this week's lab, we would like you to construct a comparator by nesting circuits, just as we did for the multi-input exclusive or gate.

The first step is even simpler than what we did in class. Rather than compare two 2-bit binary numbers, we want you to construct a subcircuit that compares two 1-bit binary numbers. Call it "compare1". Your compare1 circuit should have two outputs. The first output should be true or false depending on whether its first input (treated as a 1-digit binary number) is less than its second input. The second output should be true or false depending on whether its first input is equal to its second input.

Next, construct a "compare2" circuit. This circuit will take 4 inputs which it will view as two 2-digit binary numbers. Like compare1, the compare2 circuit should have one output that indicates whether its first pair of inputs is less than its second pair of inputs and another output that indicates whether the pairs represent equal numbers. The compare2 circuit should be constructed using two compare1 circuits and whatever other simple gates (ands, ors, nots, and exclusive-ors) you need to do the job.

Finally, construct a "compare4" circuit. You should find the construction very similar to compare2. This time, you will use two compare2 circuits as building blocks (together with some other simple gates). When you build compare4, however, use multi-input pins so that instead of expecting 8 separate input lines compare 4 expects two 4-bit buses as inputs.

In addition, in the "main" circuit of your project, connect input and output pins to a copy of your compare4 circuit in such a way that it can be tested using just your main project.

## Priority Encoder

Finally, we would like you to use Logisim to implement the circuit described in exercise 2.36 in the text. We expect you to follow the same multi-level approach illustrated in other parts of this lab. That is, you should first build a “priority2” circuit that takes two input signals and produces a DONE signal and a 1-digit number as its output. Then use this to build a “priority4” circuit which takes 4 inputs and produces both DONE and a 2-bit number as output. Finally, use priority4 to construct the circuit described in the problem from the text.

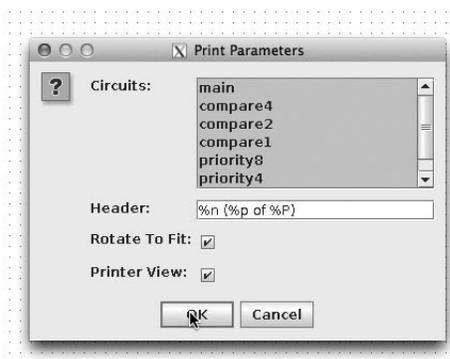
As you did for compare4, in the “main” circuit of your project, you should connect input and output pins to a copy of your priority8 circuit in such a way that it can be tested using just your main project.

## Submitting Your Work

First, make sure you select the save item from the Logisim File menu to make sure the latest version of all of your subcircuits is saved. Logisim saves this information in a file whose name ends in .circ. If you followed our instructions above, your circuits should be saved in a file whose full name is Lab3Circuits.circ.

In addition to this file, we want you to submit a file containing drawings of all of your circuits so that we can print these drawings easily. To produce this file:

- Select the “Print” item from the Logisim File menu. A dialog that looks like:



should appear.

- Shift click or drag to select the names of all of your circuits in the list in this window.
- Edit the “Header” field in the window by adding your name.
- Click OK. A new dialog box should appear.
- Make sure the Print to File check box is selected. Then click Print.
- Use the next dialog box to navigate into the folder where your .circ file is saved and save the file containing drawings of your circuits as out.ps in this folder.

Within a terminal window, use the cd command to make the directory containing your .circ and .ps files your current working directory. Then type the command:

```
turnin -c 237 out.ps Lab3Circuits.circ
```

Respond to the prompts appropriately and your work should be submitted.