

Advanced Logisim

Goals

You will be learning how to use the remaining essential parts of Logisim, in particular, splitters to take a subset of bits on a wire, and to rejoin them. The following exercises will introduce you to more advanced techniques/concepts in Logisim.

Setup

Start all Logisim circuits from scratch. Feel free to do each exercise as separate sub-circuits in the same Logisim file.

Exercise 1: Splitters

This is the last essential tool you will need in this class. To demonstrate its use, you will construct a circuit that manipulates an 8-bit number.

1. Create a new subcircuit and name it "Ex1".
2. Add an 8-bit input pin to your circuit and label it.
3. Add a 1-bit output pin labeled "Out1" and an 8-bit output pin labeled "Out2" to your circuit.
4. Go to the Wiring folder and select the Splitter circuit. This circuit will take a wire and split it into a smaller set of wires. Conversely, it can also take many sets of wires and combine them into a larger bus.
5. Before you place your circuit, change the "Bit Width In" property (bus width) to 8, and "Fan Out" property (# of branches) to 3. If you move your cursor over the



schematic, your cursor should look as follows:

6. Now, select which bits to send out to which part of your fan. The least significant bit is bit 0 and the most significant bit is bit 7. Change bits 1, 2, and 6 to be coming out on fan arm 1 (the middle one). FYI: the "None" option means that the selected bit will not come out on ANY of the fan arms.
7. Once you configure your splitter, you can place your splitter into your circuit.
8. Attach a 2-input AND gate to fan arms 0 and 2 and route the output of the AND gate to Out1.

9. Now we want Out2 to be the negative "sign and magnitude" value of the input. [Sign and magnitude](#) is an alternate way of representing signed values - like 2s complement, but simpler! The combinational logic should be straight-forward.
10. We will need another splitter to recombine the fans into a single 8-bit bus. Place another splitter with the proper properties (Bit Width In: 8, Fan Out: 3, correct fan widths). Play with the Facing and Appearance properties to make your final circuit as clean-looking as possible.

Checkoff

- *Show your Ex1 circuit to your TA.*
- *Assuming 2's complement representation, what inputs will produce Out1 = 1?*

Exercise 2: Rotate Right

With your knowledge of splitters and your knowledge and experience with multiplexers from the last lab, you are ready to implement a non-trivial combinational logic block: rotr, which stands for "Rotate Right". The idea is that rotr A,B will "rotate" the bit pattern of input A to the right by B bits. So, if A were 0b1011010101110011 and B were 0b0101 (5 in decimal), the output of the block would be 0b1001110110101011. Notice that the rightmost 5 bits were rotated off the right end of the value and back onto the left end. In RTL, the operation would be something like $R = A \gg B \mid A \ll (16 - B)$.

You must implement a subcircuit named "rotr" with the following inputs:

- **A**, 16 bits, the input to be rotated
- **B**, 4 bits, the rotation amount (Why 4 bits?)

The output should be A rotated right by B bit positions, as outlined above. You are **NOT** allowed to use Logisim shifters in your solution, though all other combinational logic (MUXes, constants, gates, adders, etc.) is allowed. Show off your rotr subcircuit in the main subcircuit.

Hint: Before you start wiring, you should think veeeery carefully about how you might decompose this problem into smaller ones and join them together. You should feel very free to use subcircuits when implementing rotr. If you don't, expect to regret it.

Hint, the second: Just because we gave you an RTL representation doesn't mean it's the best way to look at this problem. Think about the input bits of **B** and think about how to effectively use splitters!

Tip: If your wiring from a large splitter is getting messy, sometimes chaining splitters can keep things more localized and cleaner. For example, a 1 to 16 split can be achieved by a fan out of 4 connected to 4 more splitters of fan out 4.

Checkoff

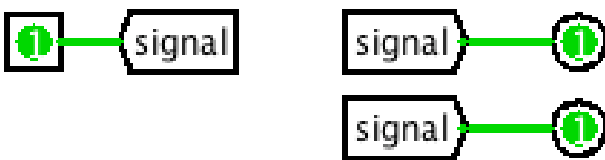
- Show your TA your rotr circuit and verify that it works.

Advanced Logisim Features

Here are two more Logisim features you will find useful.

Tunnels

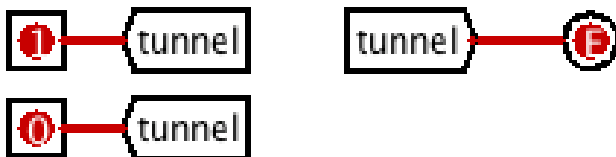
A [tunnel](#) allows you draw an "invisible wire" to bind two points together. Tunnels are grouped by case-sensitive labels give to a wire. They are used to connect wires like so:



Which has an effect such as the following:



Some care should be taken as to which wires are connected with tunnels to which other wires, such as in this case:



Which in turn has the following effect:



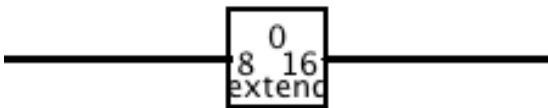
We strongly recommend you use tunnels with Logisim, because they make your circuits much cleaner looking, and therefore easier to debug.

Extenders

When changing the width of a wire, you should use a [bit extender](#) for clarity. For example, consider the following implementation of extending an 8-bit wire into a 16-bit wire:



Whereas the following is much simpler, easier to read, and less error-prone:



Additionally, consider the case of throwing out bits. In this example, an 8-bit wire is being converted into a 4-bit wire by throwing out the other bits:



Despite the implications of its name, a bit extender can also do this same operation:

