# Addition Circuit

This week you will design several circuits. In particular, you will learn to use a digital circuit design tool called Logisim, and you'll use Logisim to build a key component of a modern digital computer: An adder!

- Begin by opening Logisim on your computer. You can obtain Logisim for your own computer by going to the Logisim web site. Logisim is written in Java and should work "right out of the box" on your computer.

- First, download the `circuits.circ` file from Canvas that we've provided to help you get started.

- Start up Logisim -- as long as it's installed, you can probably start it by double-clicking the `circuits.circ` file you unzipped in the step above. Go to the "Help" menu and select "Tutorial". Read through the beginner's tutorial up to and including Step 4, *testing your circuit*. As you do so, feel free to create the XOR circuit the tutorial is describing in the "MYXOR" tab of the `circuits.circ` Logisim file. This is pretty short reading, and don't worry if you don't digest every last drop as you read it! At least you'll know what's in the tutorial so that you can go back and find the details later if you need them.

## Circuit design!

- This `circuits.circ` file is where you will write and save all of your circuits for this assignment. If your `circuits.circ` file is not yet open, open it by double-clicking on it, or open it from within Logisim by going to the Logisim "File" menu, selecting "Open", and loading in `circuits.circ`.

  **Notice** that in the Logisim explorer pane (the upper left part of the Logisim screen) there are icons labelled `MyXOR`, `full adder`, and `4-bit ripple carry adder`. Those are all of the parts ("subcircuits") of this week's lab.

## Part 1: Tutorial and XOR

- Double-click on the `MyXOR` icon in the explorer pane. For the moment, you'll be working on that subcircuit. Your job here is to build a circuit to compute the XOR function using **only** `AND`, `OR`, and `NOT` gates. This is the goal of the tutorial you read -- if you already followed that through and completed the XOR circuit, great!! Go ahead onto the full adder circuit in Part 2.

- If you have not yet built the XOR circuit, use the tutorial and the minterm expansion principle, construct a two-input circuit that outputs a 1 if and only if ***exactly one*** of the two inputs is a 1.

Label your two inputs `X` and `Y` and your output `Output`. Recall that you can label inputs and outputs by selecting the arrow from the menu bar, clicking on the input or output, and then editing the label in the attribute table—the pane in the lower left of the Logisim screen. **Do not** use the text tool to label your inputs and outputs. If you do, those labels won't show up later when you'll need them most!

Test your circuit by changing the inputs with the "poke tool" (the little hand in the upper left of the menu bar). Check to see if you are getting the right outputs for your inputs. Save the file by going to the "File" menu and clicking on "Save".

## Part 2: Designing and building a Full Adder

- Next, you'll build a full adder (FA). Recall that a full adder is a device that takes **three** inputs: two bits to be added, and the "carry in" from the previous column of addition. The FA computes **two** outputs: the "sum" bit, and a "carry out" bit to be sent or "propagated" to the next column of addition. Your first task is to either look up or write down the full truth table for a FA (full adder). You don't need to turn this in, but you'll need it to design your circuit. Then use the minterm expansion principle to find expressions for the "sum" bit and also the "carry out" bit. Note that each output requires a separate application of the minterm expansion principle.

  Don't try to simplify your circuit and please don't use gates other than `AND`, `OR`, and `NOT`. The purpose of this problem is to practice using the minterm expansion principle!

- ***Double-click*** on the **FA** subcircuit tab in your `circuits.circ` file...

- Now, go to the `FA` subcircuit by double-clicking on that icon in the explorer pane. You'll notice that we've provided the labelled inputs and outputs. Inputs `X` and `Y` are the two bits to be added, and `CarryIn` is the carry from the previous column. Similarly, the two outputs are already placed at the bottom. You can move these if you need to, but please keep their relative positions the same so that we can easily test your circuit.

  Notice that we have rotated the input and output pins from their normal default orientation. This was done by clicking on the item and selecting the desired rotation in the attribute pane in the lower left of the screen. Test your full adder by changing inputs with the "poke" tool. Now is a good time to save your file again, too!

## Part 3: Building a 4-bit ripple-carry Adder

- ***Double-click*** on the ripple-carry adder subcircuit in your `circuits.circ` file...

- Now that you have a full adder (FA), you will build a 4-bit ripple-carry adder. Recall that this device takes two 4-bit numbers and adds them up. Let's call the digits of the first number X3, X2, X1, X0 where X0 is the least significant bit (the rightmost bit) and X3 is the most significant bit. Similarly, let's call the digits of the second number Y3, Y2, Y1, Y0. We wish to compute:

- ```
    X3 X2 X1 X0
  ```

- ```
  + Y3 Y2 Y1 Y0
  ```

  ```
  ---------------
  ```

Notice that although there are only 4 bits in each of the two numbers, the sum can have 5 bits due to a carry! Double-click on the "4-bit Ripple-Carry Adder" icon in the explorer pane. You can "plop" full-adders into your 4-bit ripple-carry adder by *single-clicking* on the "FA" icon from the explorer pane (since you've already designed the FA). This will create a "box" that represents your FA. After all, now that you've designed the FA, you don't particularly want to see all of its details each time you use it! If you move your mouse ("hover") over this box, you'll see that it says "FA". If you "hover" over the "pins" (the little input and output markers on the box), it will show you what those pins are. This is because we labelled those pins when we designed the FA.

You can move the inputs and outputs that we've provided, but please keep their relative positions so that we can find them easily.

One last thing! You'll need to have a carry-in of 0 on the far right of your ripple-carry adder. One way to do this is to add an extra input and set its value to be 0. This is not a great solution, since we really don't want the user of our ripple-carry adder to be able to change that carry-in value. A better choice is to go the explorer pane, click on the "Gates" folder, and then select the first item that you see there. It's labelled "1 Constant". Plop one of these down in your circuit. Then click on it and you can alter the value of that constant in the attributes pane. This is now a constant value that your circuit can use!