Two-bit Adder Lab - Instructions

Overview

For this lab, you will create a logic circuit that performs binary addition. This circuit will be larger and have more wires and gates than the circuits you built in the previous lab, so we will connect and test the circuit in a simulation program (Tinkercad). Then, after verifying your circuit in simulation, you may optionally wire it up on your breadboard with actual wires and chips for extra credit (+10). There will also be the option of creating a subtractor circuit for extra credit (+10).

Two-bit Binary Addition

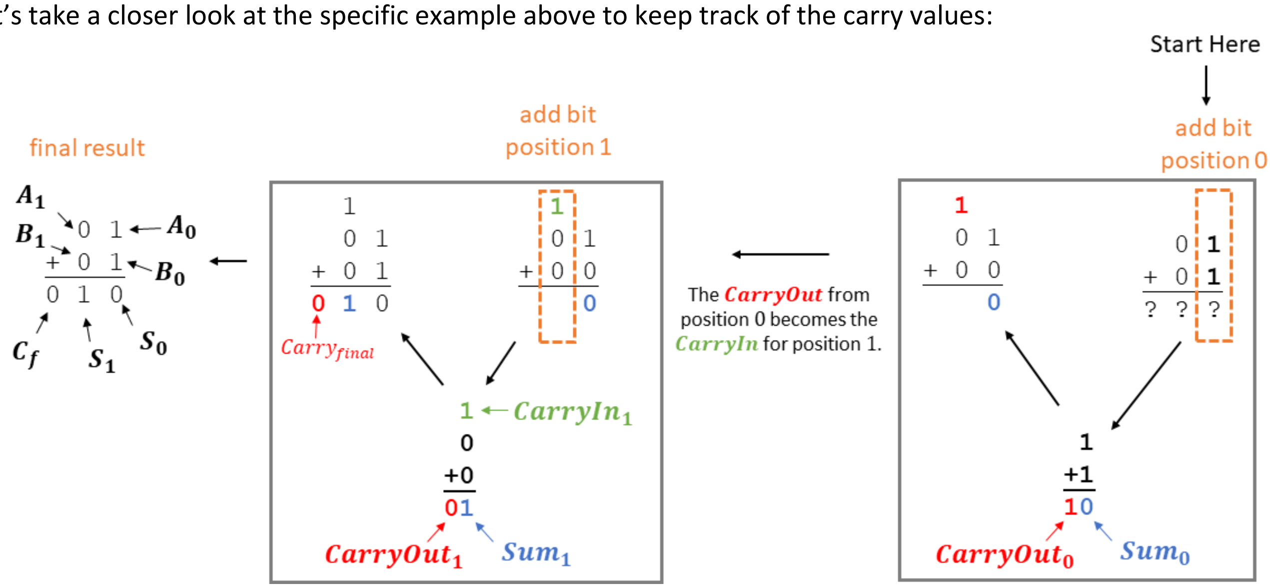
Because each bit of a binary number can be '1' or '0', we can represent each bit of a binary number with its own Boolean variable. We number each of the variables, starting with zero on the *least significant bit* (the bit farthest to the right). For example, if you have a 4-bit binary number A with the value 1011, you can write the number as $A_3A_2A_1A_0$, where $A_3 = 1, A_2 = 0, A_1 = 1$, and $A_0 = 1$.

Your circuit will add a *two-bit* binary number (A_1A_0) to another two-bit binary number (B_1B_0). The result of your addition will be a two-bit binary number Sum (S_1S_0), with a final carryout, $Carry_{final}$ (C_f). The addition your circuit performs will follow this format:

```
     A1 A0                      0 1
+    B1 B0   for example:     + 0 1
Cf   S1 S0                    0 1 0
```

Note: Do not confuse the '+' sign here with the OR operation. While they use the same mathematical symbol, here we are doing binary addition and not a Boolean OR operation.

Let's take a closer look at the specific example above to keep track of the carry values:



Here are more example addition problems your circuit will be able to perform:

```
   0 1        1 1        0 0        0 1        1 0
 + 0 0      + 1 0      + 1 1      + 1 1      + 1 0
 0 0 1      1 0 1      0 1 1      1 0 0      1 0 0

(1+0=1)    (3+2=5)    (0+3=3)    (1+3=4)    (2+2=4)
```

Truth Table

Arithmetic functions such as addition, can be defined using a truth table. Each row of the truth table is a different binary addition problem. Take a look at a few rows of the truth table below to make sure you understand how the output values are calculated.

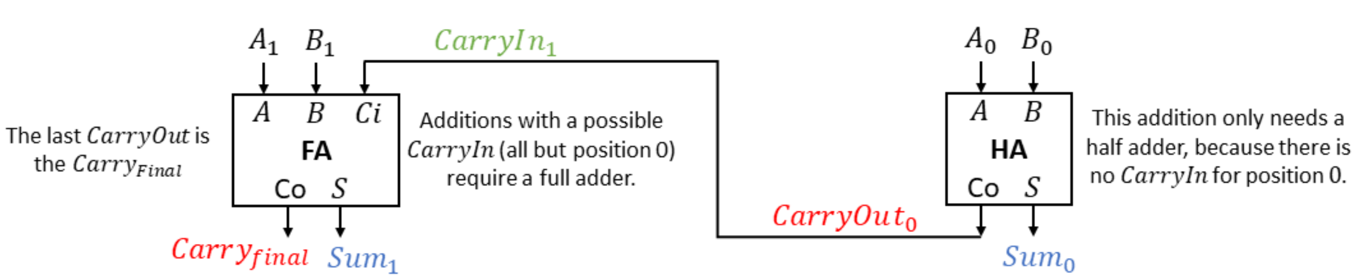
A_1	A_0	B_1	B_0		$Carry_{final}$	Sum_1	Sum_0
0	0	0	0		0	0	0
0	0	0	1		0	0	1
0	0	1	0		0	1	0
0	0	1	1		0	1	1
0	1	0	0		0	0	1
0	1	0	1		0	1	0
0	1	1	0		0	1	1
0	1	1	1		1	0	0
1	0	0	0		0	1	0
1	0	0	1		0	1	1
1	0	1	0		1	0	0
1	0	1	1		1	0	1
1	1	0	0		0	1	1
1	1	0	1		1	0	0
1	1	1	0		1	0	1
1	1	1	1		1	1	0

For example, the last row of the truth table is adding the binary number 11 (A_1A_0) to the binary number 11 (B_1B_0), which results in the binary number 110 ($C_fS_1S_0$) – which is 3+3=6 in base ten.

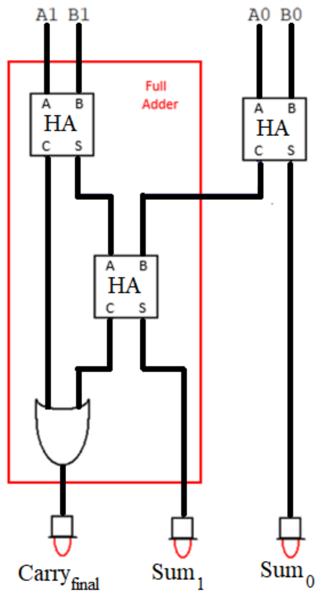
You will construct and test a 2-bit binary adder circuit that implements this truth table and verify its correct operation.

A Two-Bit Adder Circuit

For the least significant bit (bit position 0), you only need a half adder (HA). For bit position 1, you will need a full adder (FA) so that it can have a $CarryIn$ input.

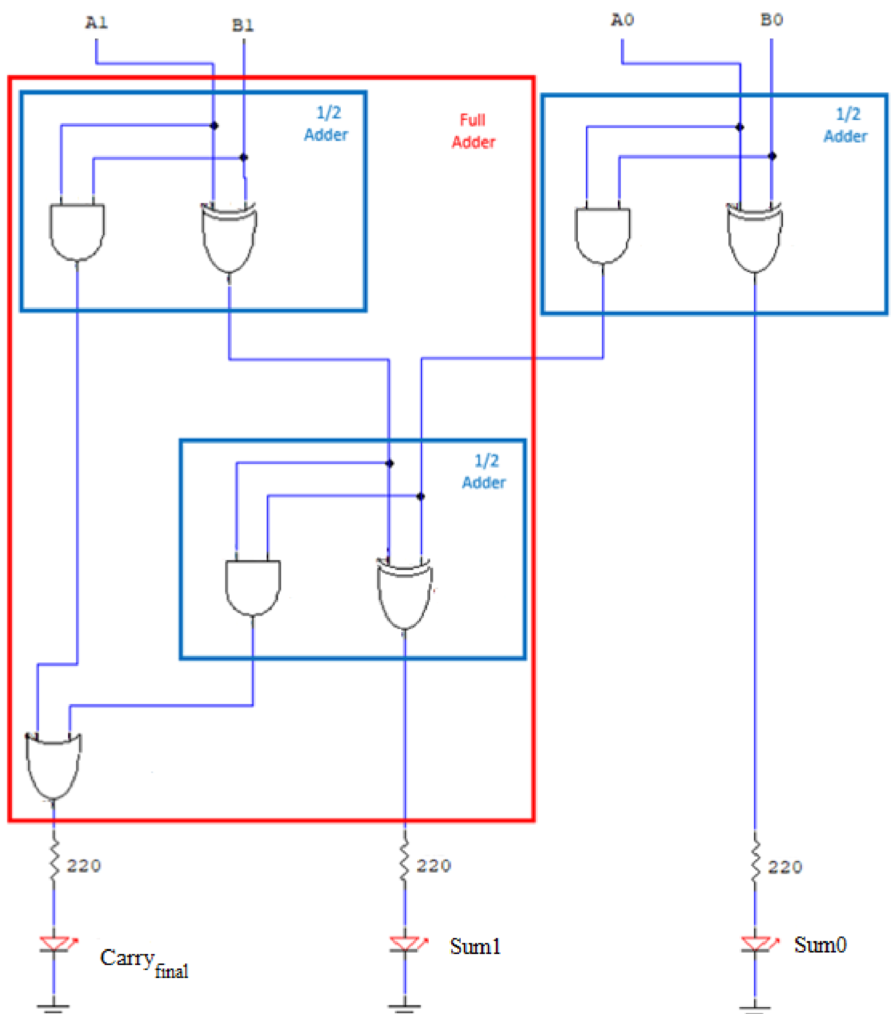


A block diagram for the circuit that implements the 2-bit adder is shown below.



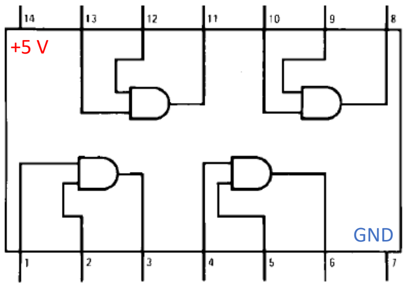
This diagram shows the outputs going to LEDs. The full-adder can be constructed with two half-adders and an OR gate.

The schematic for the circuit with all the logic gates is shown below. The instructions will periodically refer to this schematic to help you understand how you are connecting the logic gate chips that have the gates inside.

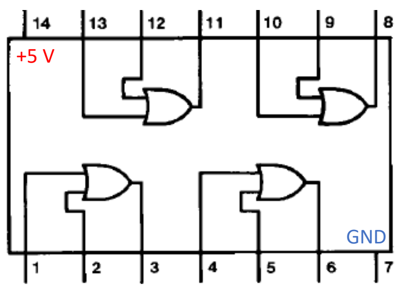


Logic Chips

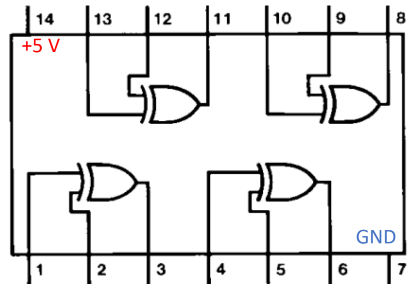
Your circuit will require three AND gates, three XOR gates, and one OR gate, as shown in the previous figure. In the previous lab, you used the integrated circuit (IC) chips from your kit that each contain multiple logic gates inside. In this lab, you will use simulated versions of the same chips. The following links take you to datasheets that show the pin connections of each chip: 74xx08 (four 2-input AND gates), 74xx32 (four 2-input OR gates), and 74xx86 (four 2-input XOR gates). These datasheets help you know which pins to connect your wires to.



74xx08 AND chip



74xx32 OR chip



74xx86 XOR chip

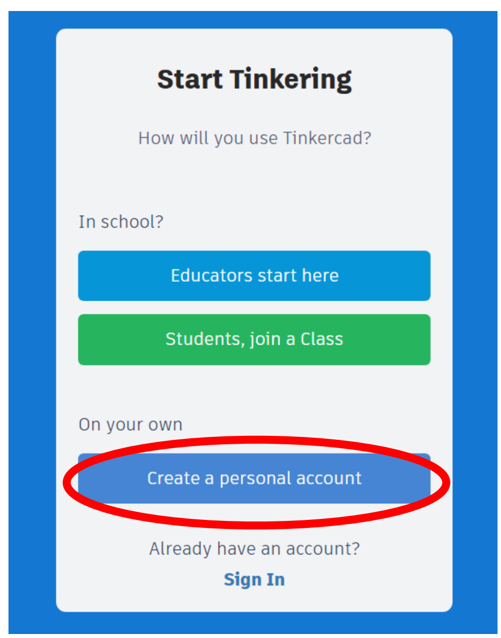
When you use this chips in Tinkercad, they will look like the chips in the image below.

Creating a Tinkercad Account

(If you already have a Tinkercad account, you may skip this section, log into your existing account, and jump to the next section: *Creating a New Tinkercad Circuit Simulation*.)

Tinkercad is a free online simulation tool that, among other things, allows you to design and simulate circuits. It has simulated LEDs, wires, logic gate IC chips, breadboards, Arduinos, resistors, etc., that you can connect together and observe the results.

1. Begin by going to https://www.tinkercad.com/.
2. Click "Sign Up."
3. Choose "Create a Personal Account" (see image to the right)



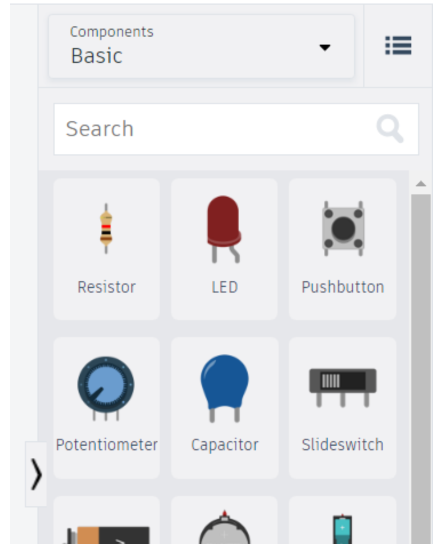
Note: Do not choose "Students, Join a Class" unless your instructor has told you otherwise.

4. Continue with all the prompts to finish creating your account.

Creating a New Tinkercad Circuit Simulation

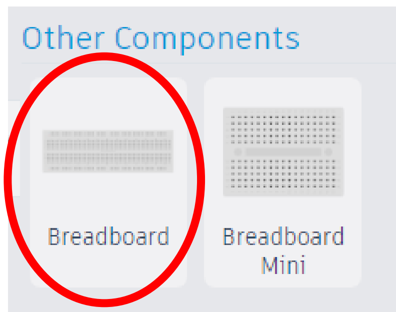
After logging in, click on "Create your first Circuits design." If you can't see the link to create a Circuits design, try clicking on your user profile picture in the top right, then "New Design," and then "Circuit."

Adding a Breadboard



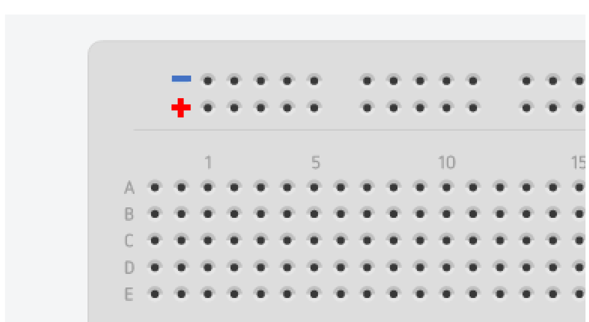
On the right of the screen, you will find the Components Panel (see image to the right), which you can hide or show with the arrow.

In the component Search bar, type "breadboard." Then choose the "Breadboard" component (not "Breadboard Small" or "Breadboard Mini") as shown below.



After clicking on the Breadboard component, click in the middle of the screen to place the breadboard.

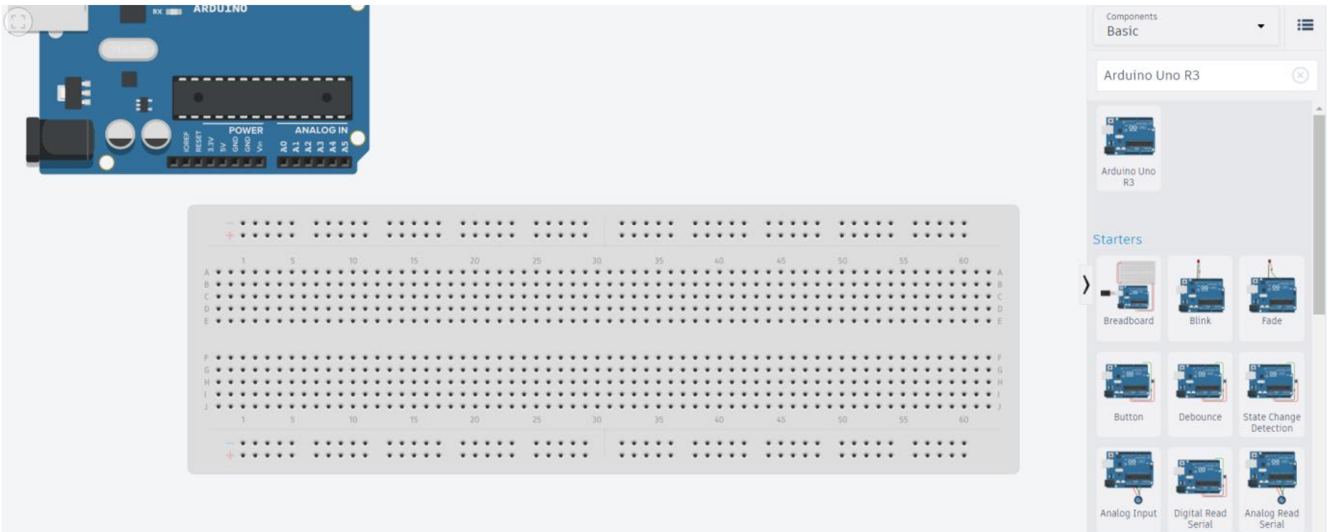
Notice the + (plus) and – (minus) signs next to the buses on the breadboard. The default orientation of the breadboard positions the LOW voltage bus (+) above the HIGH voltage bus (–).

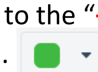


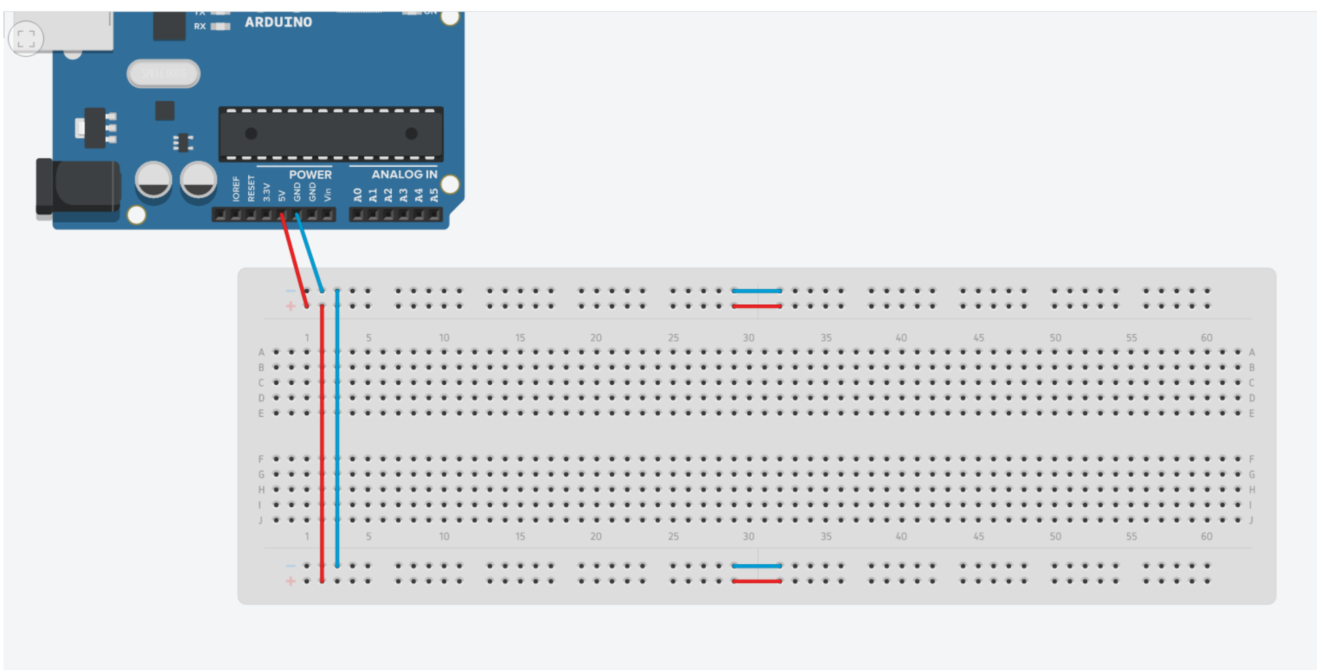
Connecting HIGH and LOW voltage to the Breadboard

There are multiple methods in Tinkercad to get a HIGH and LOW voltage supply. For this lab, you will add an Arduino UNO to access its 5V (HIGH) and GND (LOW) pins. Even though we don't need to run any code on the Arduino, we will use this method so that it matches closely with the components you have in your kit.

1. In the Components Panel, search for "Arduino Uno R3."
2. Click on the "Arduino Uno R3" and place it above the breadboard.

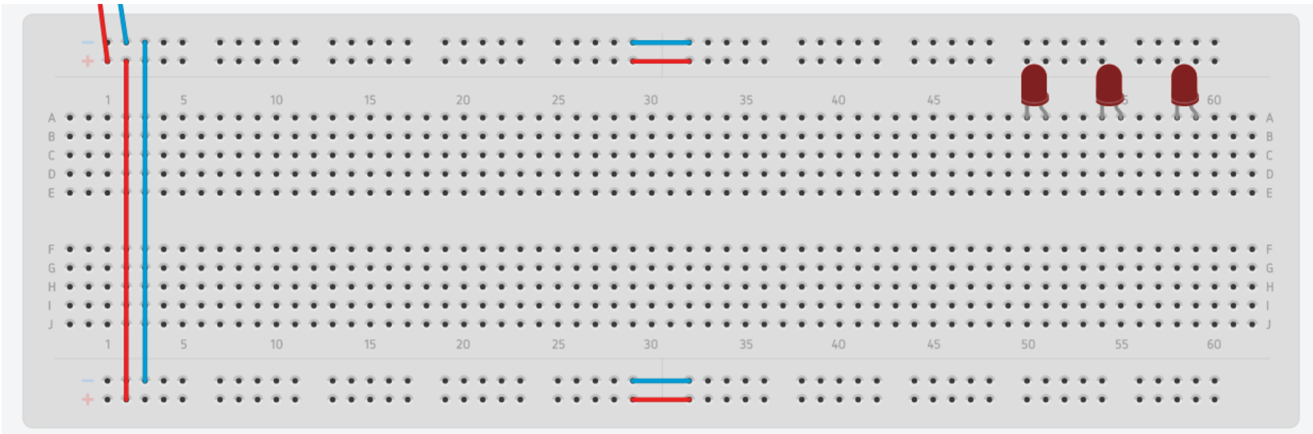


3. Click the 5V pin of the Arduino and connect it to the HIGH Voltage bus (a pin next to the "+" sign).
4. Change the color of the wire to red with the menu option at the top of the screen. 
5. Click one of the GND Arduino pins and connect it to the LOW Voltage bus (a pin next to the "–" sign).
6. Change the color of the GND wire to blue.
7. Connect the left buses to the right buses and the top buses to the bottom buses as in the previous lab.
8. Change each wire to red or blue appropriately. Your circuit should now look like the image below:



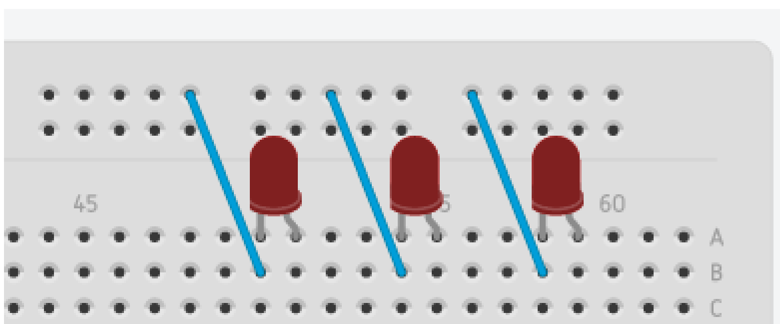
Connect and Test the Output LEDs

1. In the Components Panel, search for LED. Click on the red LED and place it on the breadboard positioned as in the image below. Place two more LEDs next to it.

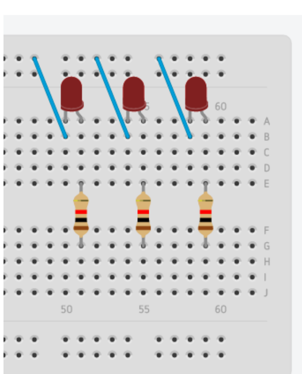


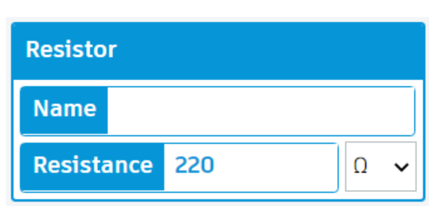
The bent leg of each LED is its longer leg, which is the positive side. The straight leg is the shorter leg and is the negative side. For an LED to turn on, it needs a higher voltage on the positive side than the negative side.

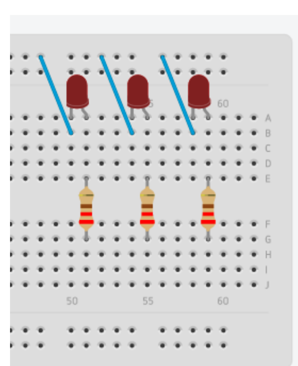
2. Connect the negative side of each LED to Ground by using wires to connect a pin from the same column as the negative side of each LED to a pin on the Ground bus row, as shown below.



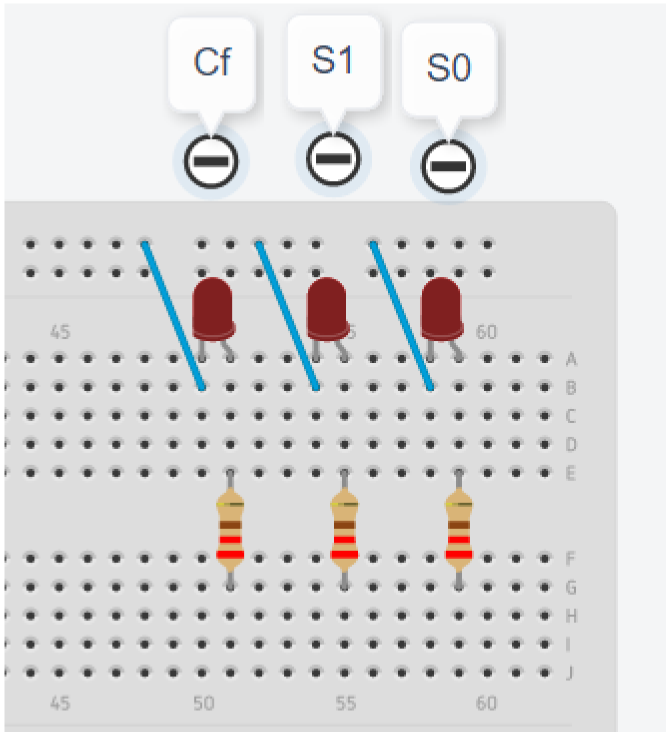
3. LEDs require *current-limiting* resistors to not burn out. In the Components Panel, search for “Resistor.” Place a resistor for each LED on the breadboard across the middle gap in the same column as the positive side of each LED, as shown below (left).
4. We need 220 Ohm (220 Ω) resistors. Click on each resistor and change the resistance to 220 Ω. (Be sure to change k Ω to Ω). That will change the colors to Red, Red, Brown, Gold (bottom to top). Note that your kit labels 220 Ω as Red, Red, Black, Black, Brown–both color patterns mean 220 Ω.



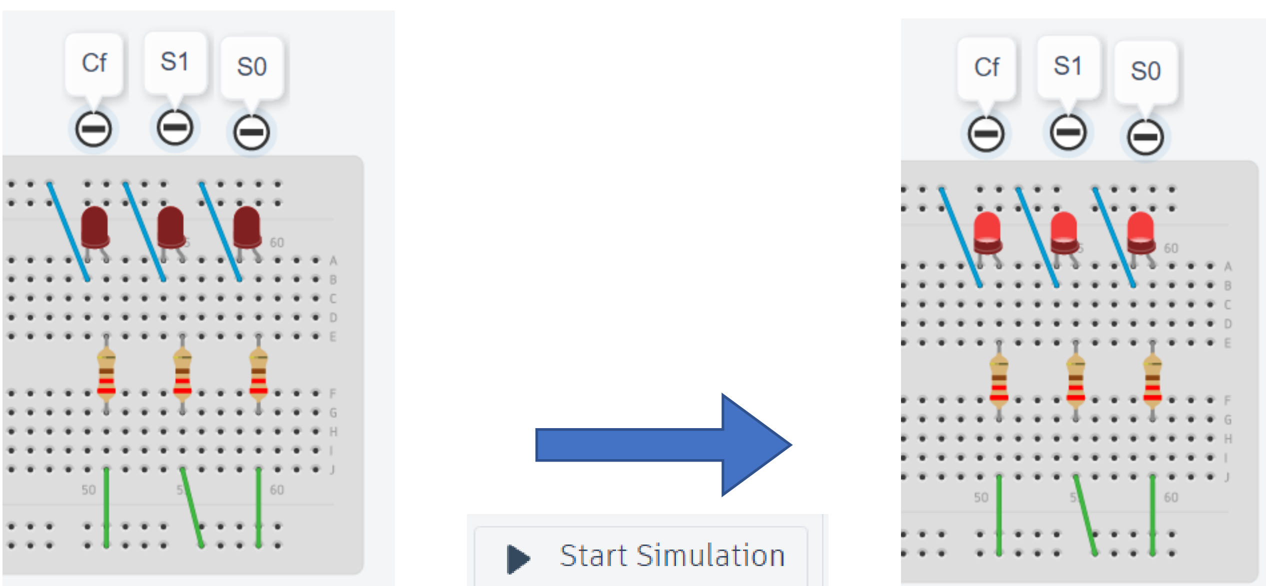




5. To keep track of what each LED is showing, click on the “Notes” tool in the top menu (shown to the right), and add a note above each LED, labeling them as **Cf** ($Carry_{final}$), **S1** (Sum_1), and **S0** (Sum_0).



6. Test that the LEDs can turn on by temporarily adding three test wires to connect the other end of each resistor to the high voltage bus (see image below). Then press “Start Simulation.”

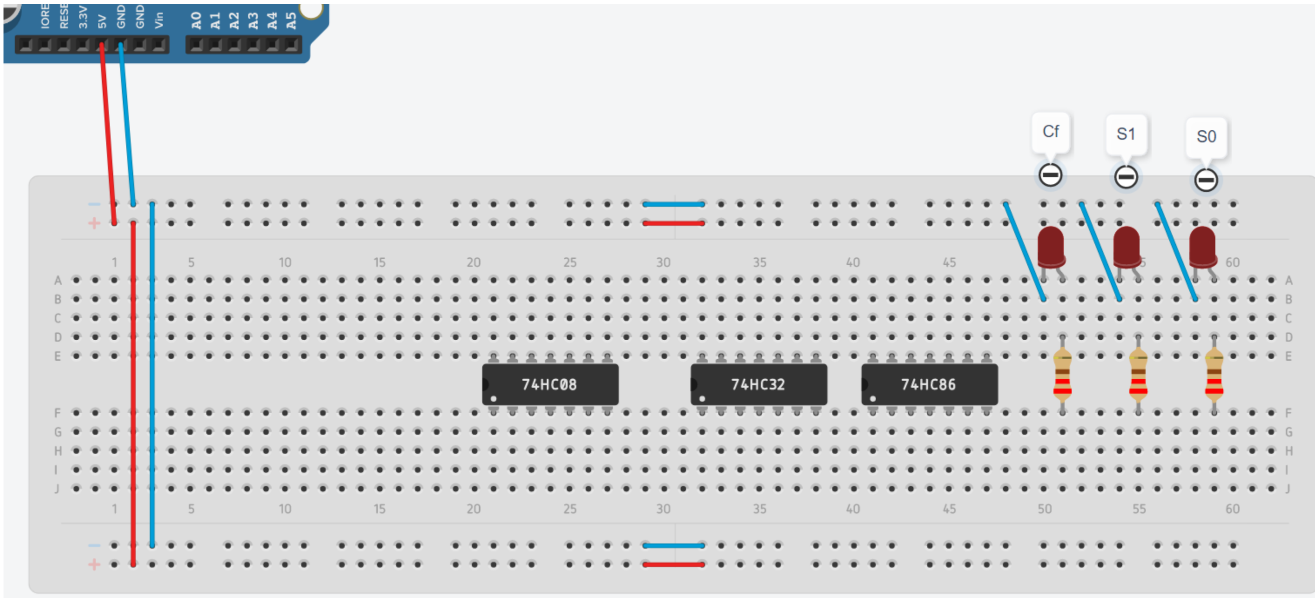


If the LEDs do not turn on, double check the wiring. You can always ask your instructor for help.

7. End the simulation by pressing “Stop Simulation.”
8. **Remove the three green test wires.**

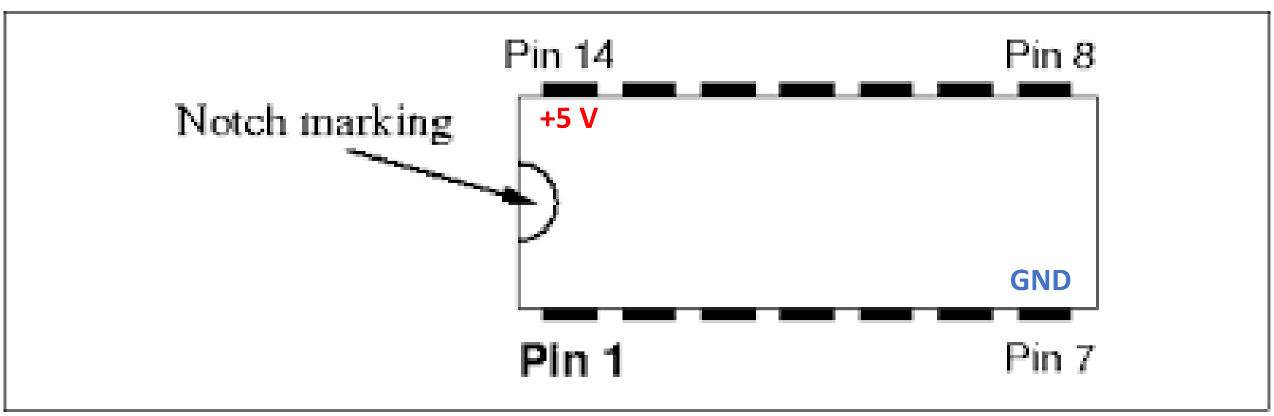
Add the Logic Gate Chips

1. In the Components Panel, search for “7408” and select the “Quad AND gate” chip.
2. Add it to the breadboard, straddling the middle gap.
3. Repeat with “7432” (the “Quad OR gate” chip) and with “7486” (the “Quad XOR gate” chip). See the image below.

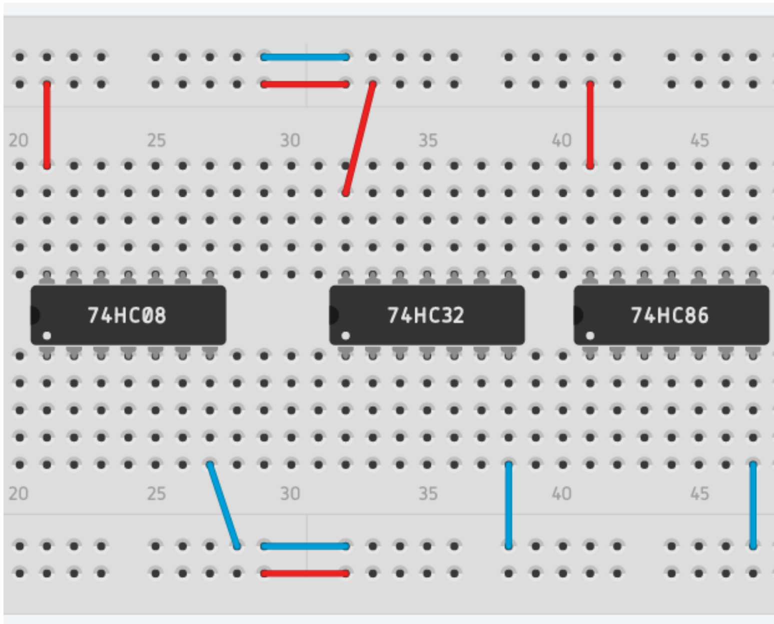


Note: The spacing between chips shown above is to make room for wires we will be adding later. Your chips may go on any columns, as long as the wires you connect to them are in the same column as the pin on the chip they are meant to connect to. However, you may find it easier to follow the remaining instructions if you place the chips in the same positions shown above. The columns are numbered.

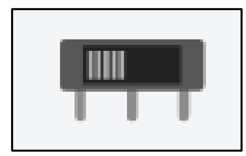
Recall from the pin diagrams in the datasheets that the top left pin (Pin 14) of each chip should be connected to 5V and the bottom right pin (Pin 7) of each chip should be connected to Ground.



4. Connect pin 14 of each chip to 5V (the HIGH voltage bus) and pin 7 of each chip to Ground (the LOW voltage bus) as shown in the next image.

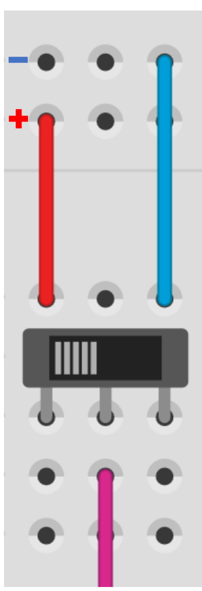


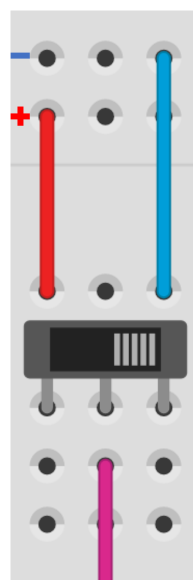
Add the Slider Switches as Input Signals



In a previous lab you moved an input wire back and forth between the HIGH voltage bus and the LOW voltage bus to change the input signal back and forth between a TRUE (1) and a False (0). You may have also used push buttons in a lab. Tinkercad has slider switches that you will use for the input signals.

A slider switch works by connecting one side to a HIGH voltage and the other side to a LOW voltage. There is a third middle pin that connects to the left or right side based on if the slider is slid to the left or to the right:

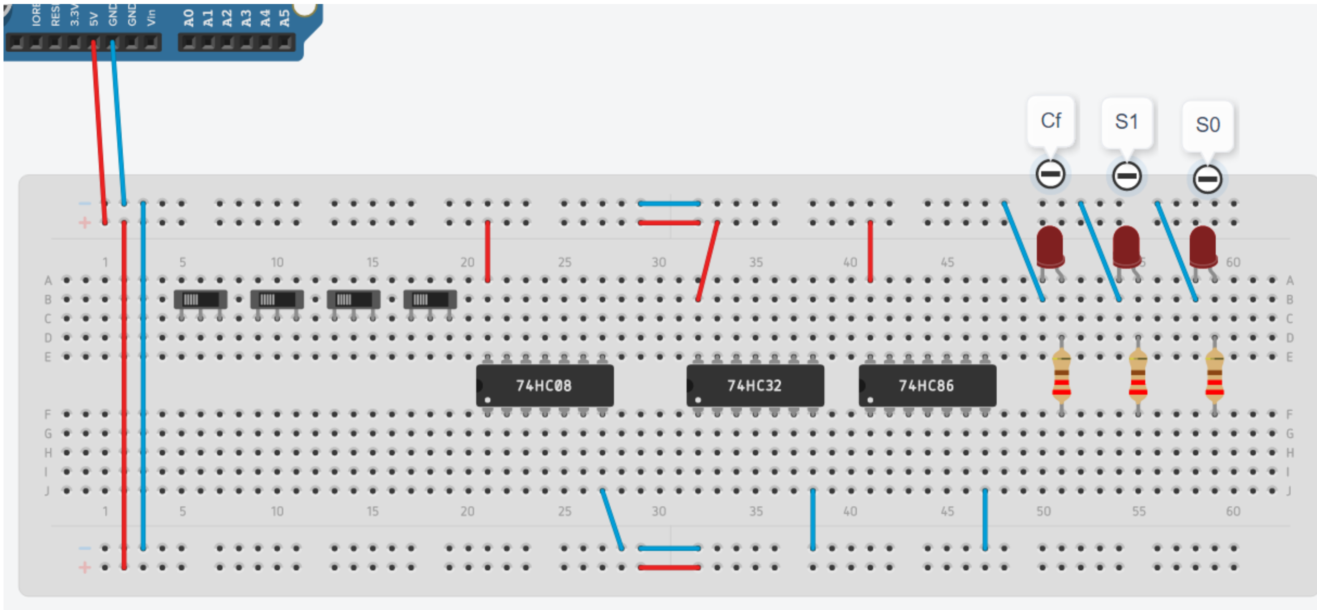




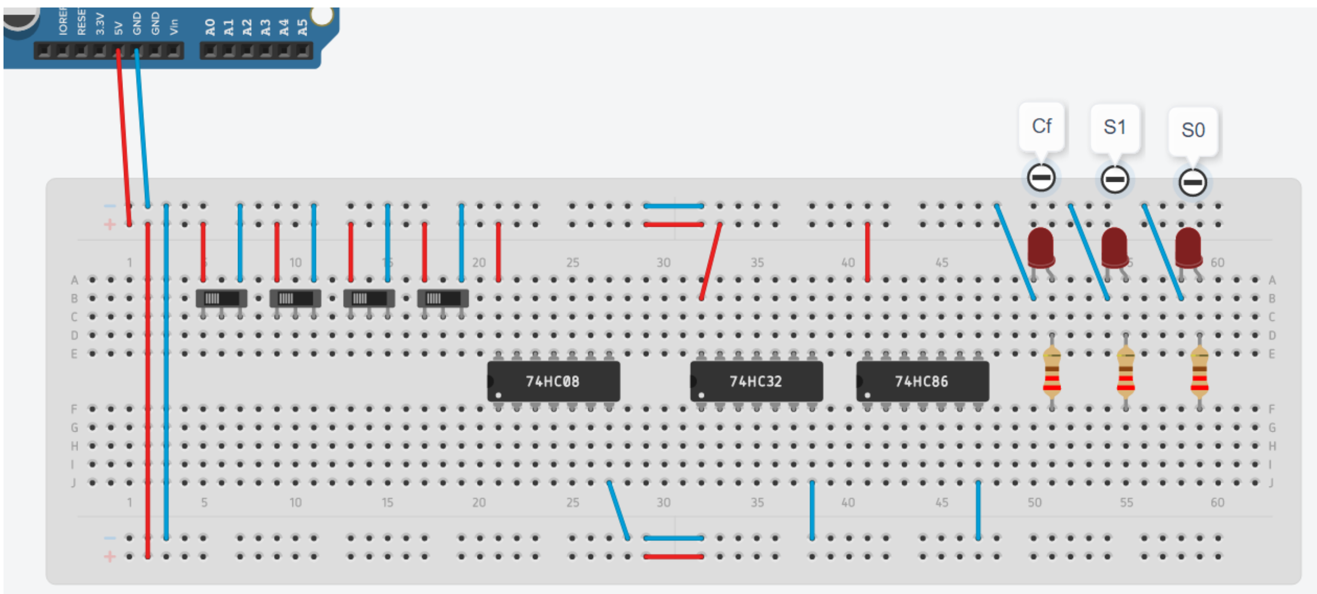
With the slider switched to the **left side**, the pink wire on the middle pin is connected to the red wire on the left pin. In this case, since the left pin is connected to **HIGH** voltage, the pink wire also has a HIGH voltage and can send an input of ‘**1**’ to a gate.

With the slider switched to the **right side**, the pink wire on the middle pin is connected to the blue wire on the right pin. In this case, since the right pin is connected to **LOW** voltage, the pink wire also has a Low voltage and can send an input of ‘**0**’ to a gate.

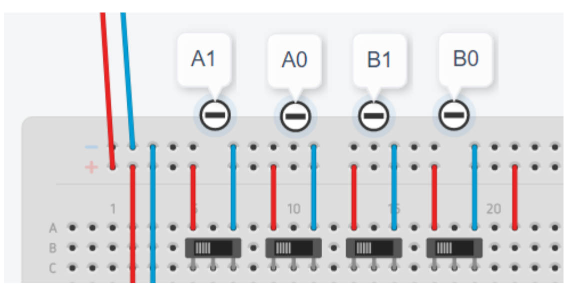
1. In the Components Panel, search for “Slideswitch.” Add four slider switches to the left side of the breadboard, like in the image below:



2. For each slider switch, connect the left pin to the HIGH voltage bus and the right pin to the LOW voltage bus. Your circuit should now look like the image below:

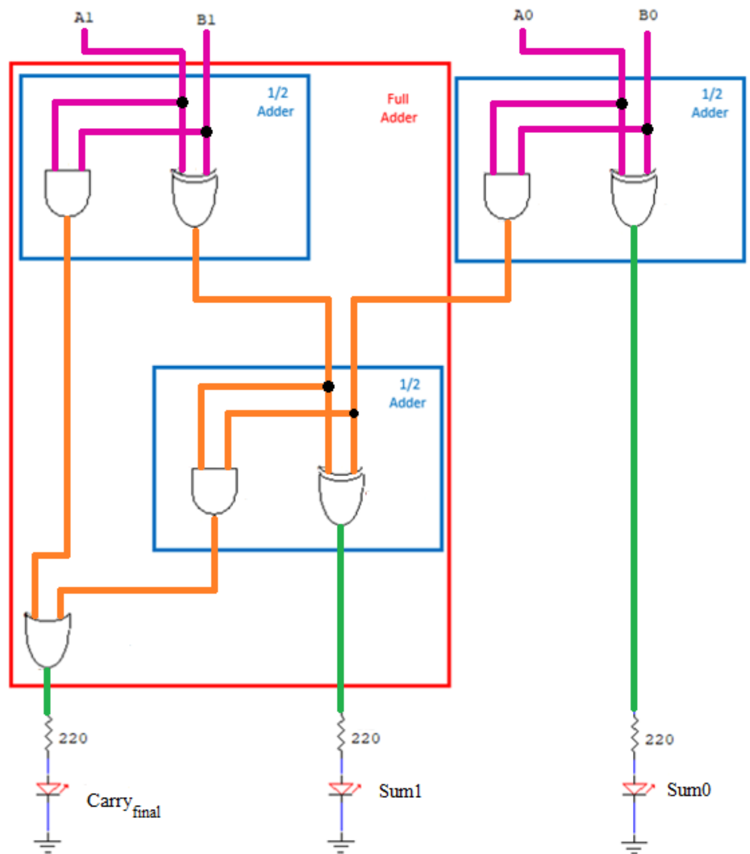


3. Using the Notes tool, add labels above each of the slider switches, labeling them as A1, A0, B1, and B0:



Add the Input Wires

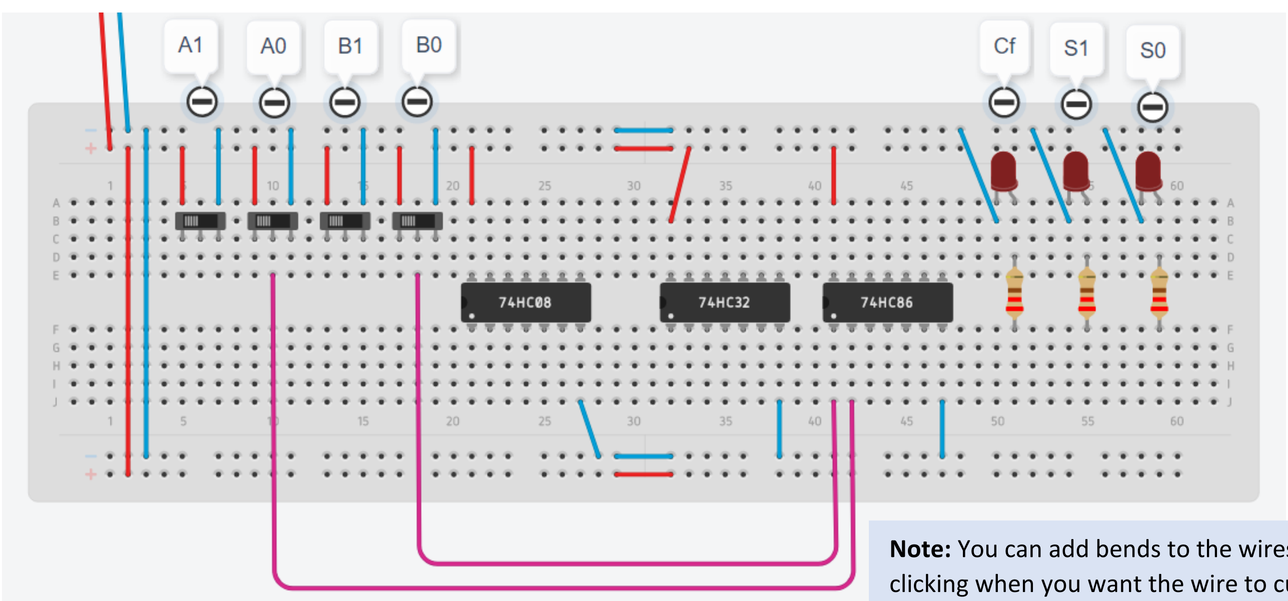
You may use any color wire you wish. These instructions use pink/purple for the input wires from the sliders, green for the output wires to the LEDs, and orange for the internal gate-to-gate wires, as in this schematic:



Note: Each input signal (A1, B1, A0, B0) goes into a half adder. In each half adder is an XOR gate (for the sum) and an AND gate (for the carry). Each input signal goes to both the XOR gate *and* the AND gate in the half adder. A0 and B0 are added together for the bit 0 position addition, and A1 and B1 are added together for the bit 1 addition.

This section will instruct you on how to connect the input wires (pink) coming from A1, A0, B1, and B0.

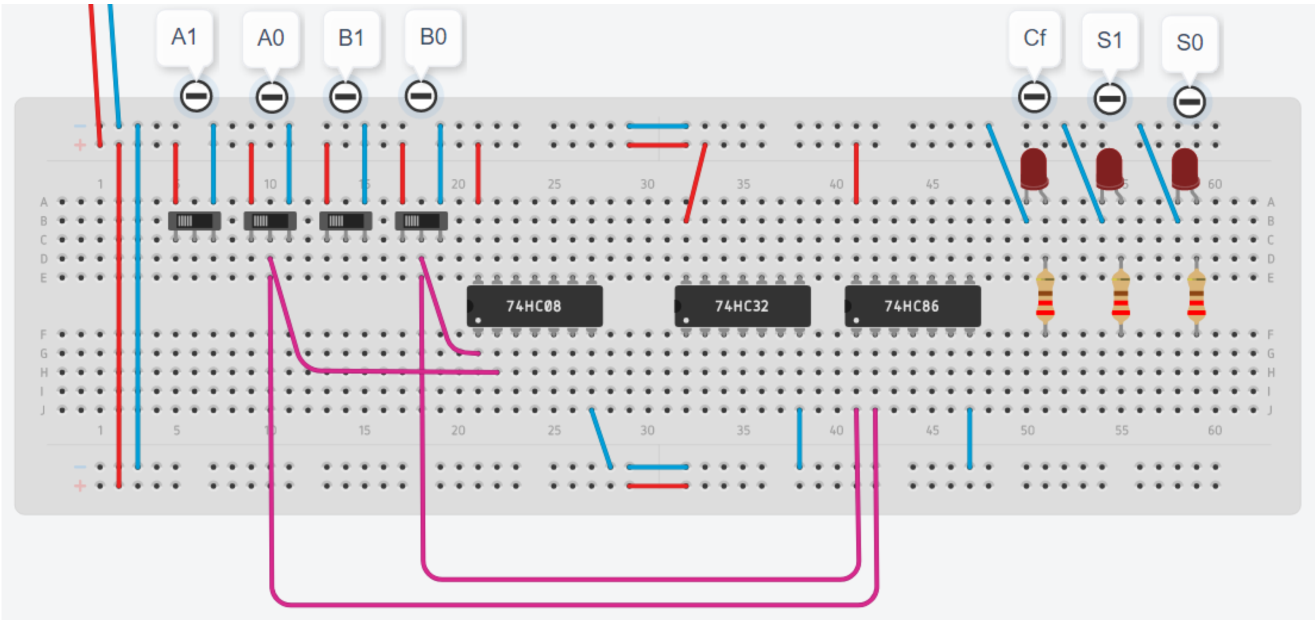
1. Connect A0 and B0 (the middle pins of their respective sliders) to the first XOR gate by connecting B0 to pin 1 of the 74xx86 chip and A0 to pin 2 of the same chip:



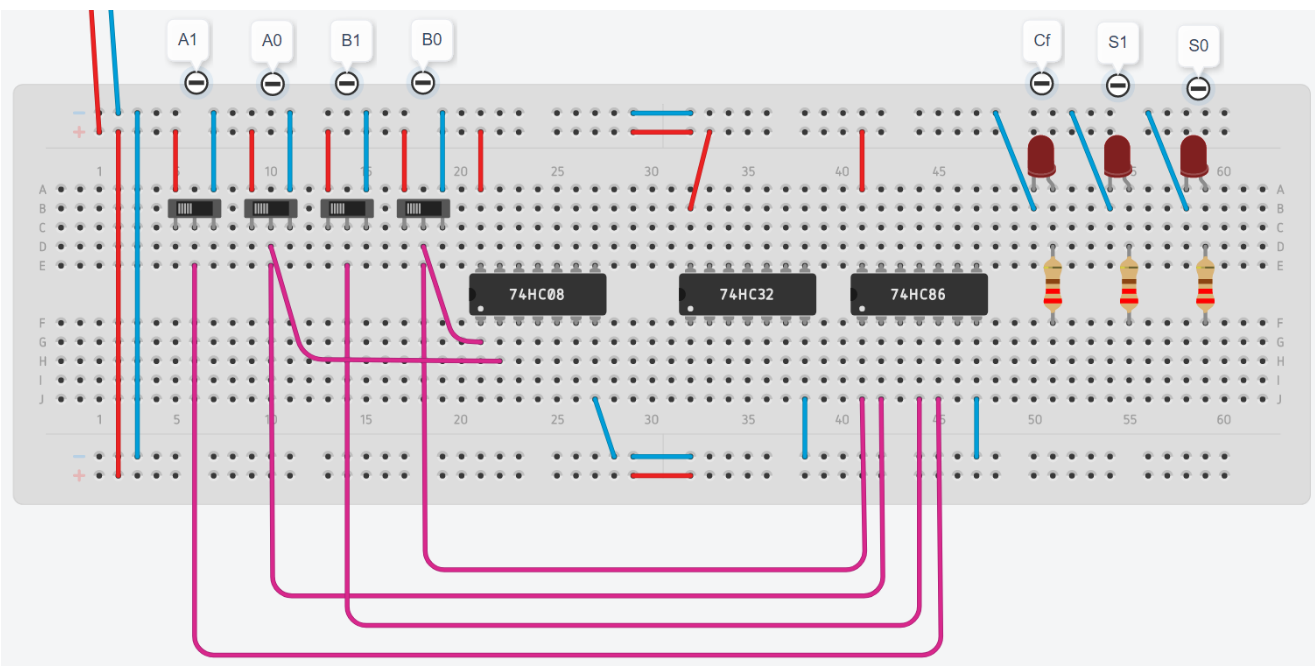
Note: You can add bends to the wires by clicking when you want the wire to curve.

Inputs A0 and B0 also need to go to an AND gate. You can have an input wire go to multiple gates by connecting another wire coming from the same column.

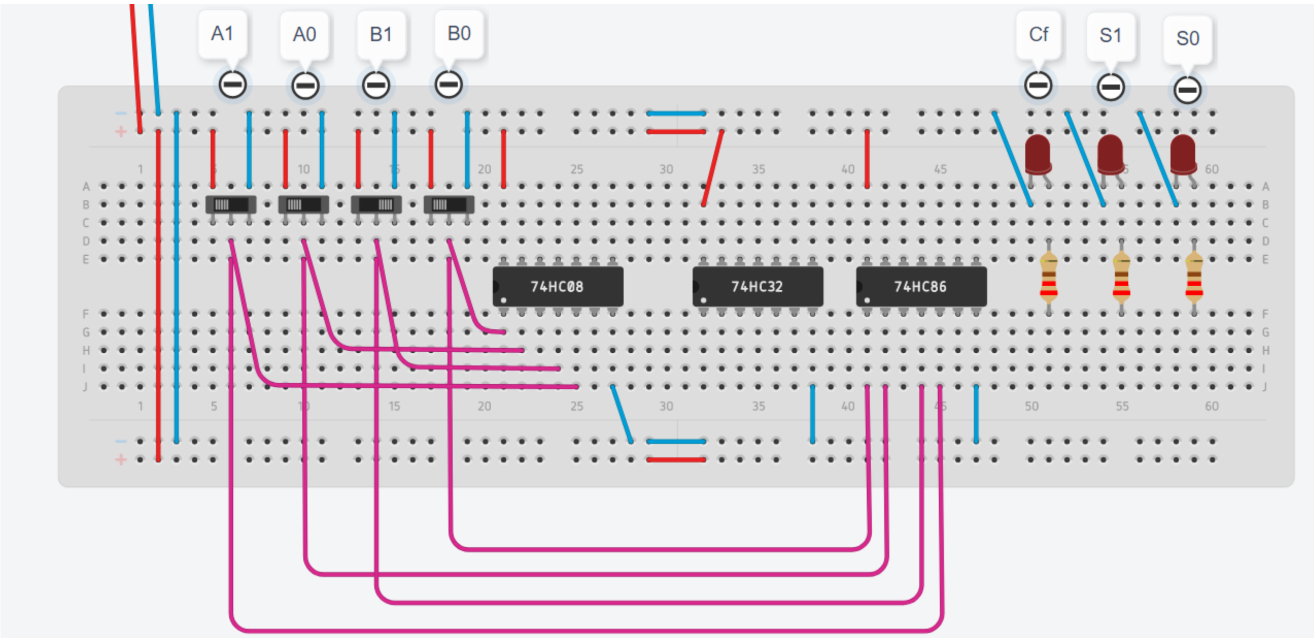
2. Connect A0 and B0 (the middle pins of their respective sliders) to the first AND gate by connecting B0 to pin 1 of the 74xx08 chip and A0 to pin 2 of the same chip:



3. Now, moving on to bit position 1, connect A1 and B1 to the next XOR gate by connecting B1 to pin 4 of the 74xx86 chip and A1 to pin 5 of the same chip:



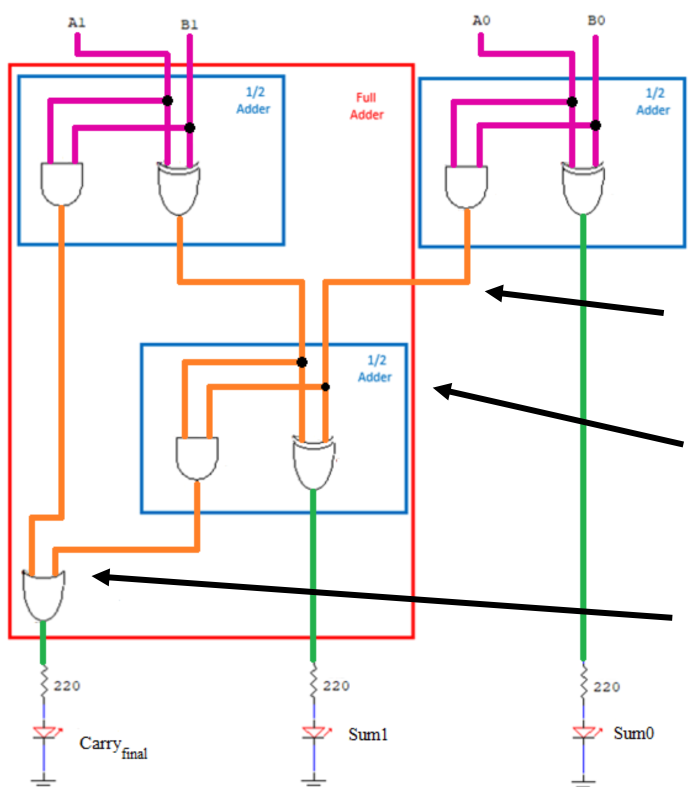
4. Connect A1 and B1 also to the next AND gate by connecting B1 to pin 4 of the 74xx08 chip and A1 to pin 5 of the same chip. Your circuit should now look like the image below:



Your input connects are now complete. Next are the gate-to-gate wires.

Add the Gate-to-Gate Wires

In this section, you will add the gate-to-gate wires shown below in orange. These wires include the connections that add the carry signal from the position 0 half adder with the sum signal of the position 1 half adder to turn it into a full adder. Also, the two carry signals from the two half adders in the full adder get ORed together.

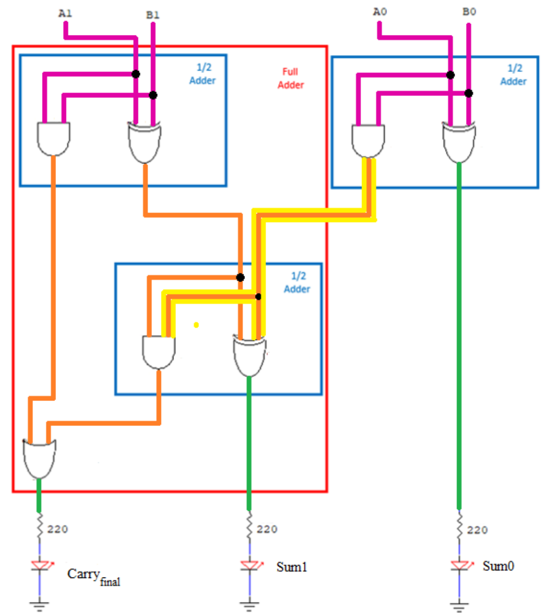


Note: The $CarryOut$ from the bit 0 position becomes the $CarryIn$ for the bit 1 position.

Note: Because bit position 1 has a $CarryIn$ signals, it needs another half adder to become a full adder.

Note: The carry signals from the two half adders in a full adder get ORed together.

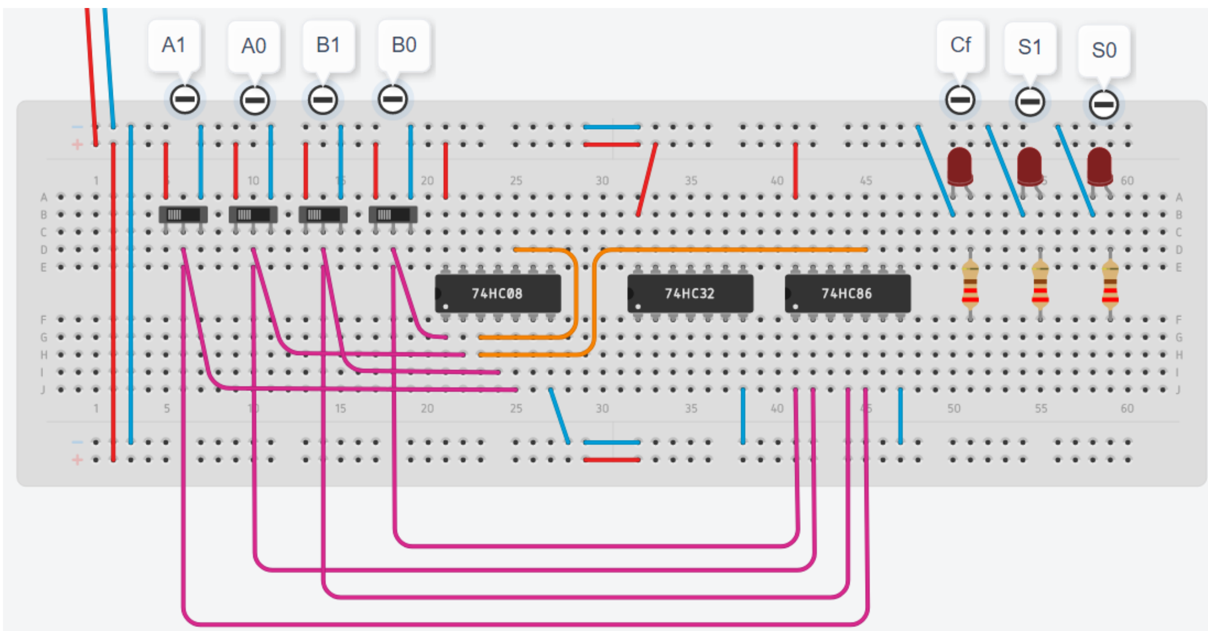
The *CarryOut* for the addition on bit position 0 is the orange wire highlighted in yellow in the image to the right. That signal comes from pin 3 of the 74xx08 and becomes the *CarryIn* to the full adder for the addition on bit position 1 by going to both the XOR gate and the AND gate in the last half adder.



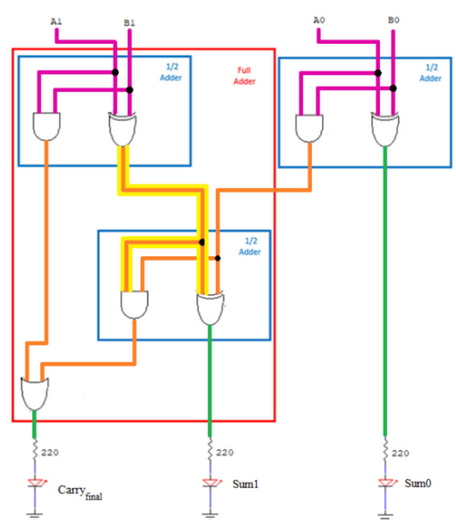
Connect this signal by doing the following two steps:

1. Connect the *CarryOut* from bit position 0 to the last XOR gate by connecting pin 3 of the 74xx08 chip to pin 10 of the 74xx86 chip.
2. Connect that same signal to the last AND gate by connecting pin 3 of the 74xx08 chip to pin 10 of the 74xx08 chip.

See the two wires just added below in orange:

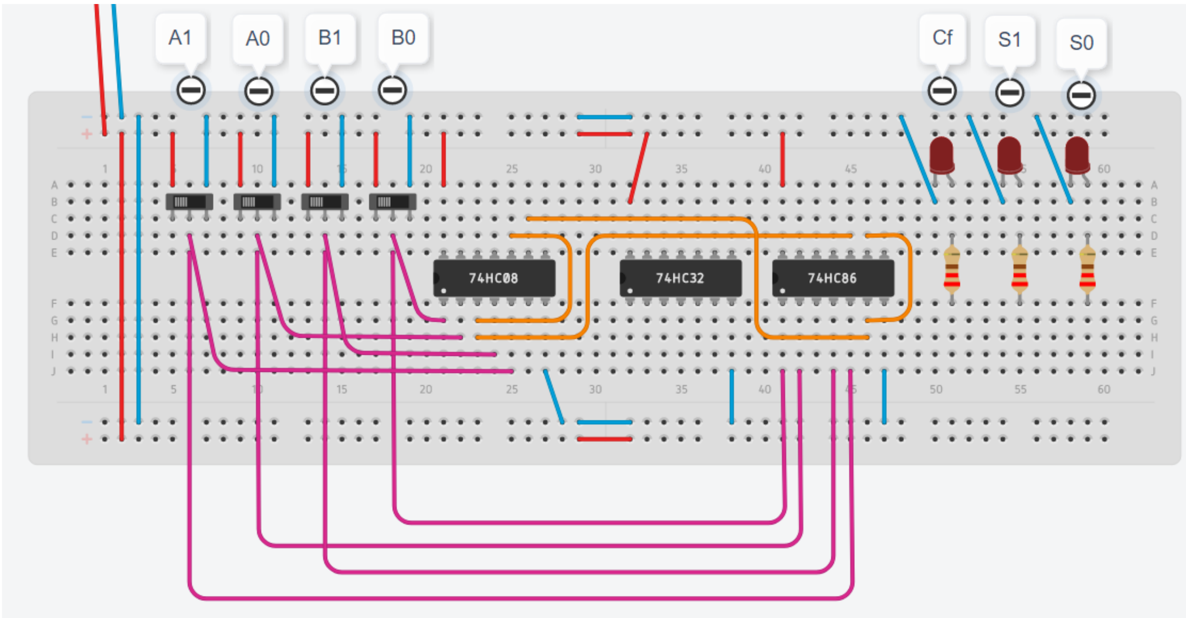


Next, the signal added to that $CarryIn$ value is the sum from bit position 1's first half adder, shown in the orange wire highlighted in yellow to the right. It comes from pin 6 of the 74xx86 chip.

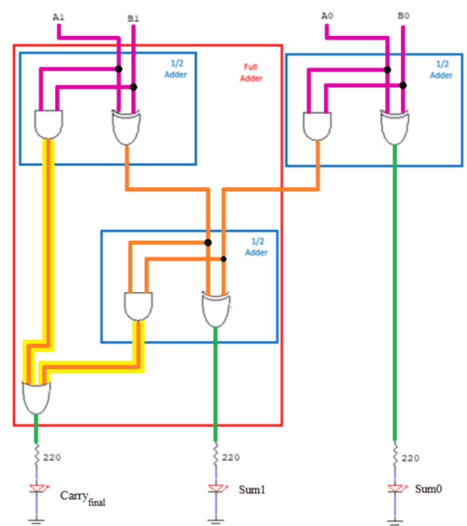


3. Connect pin 6 of the 74xx86 chip to pin 9 of the 74xx86 chip.
4. Also connect pin 6 of the 74x86 chip to pin9 of the 74xx08 chip.

After completing these steps, your circuit should look like the image on the next page, with the two additional orange wires.

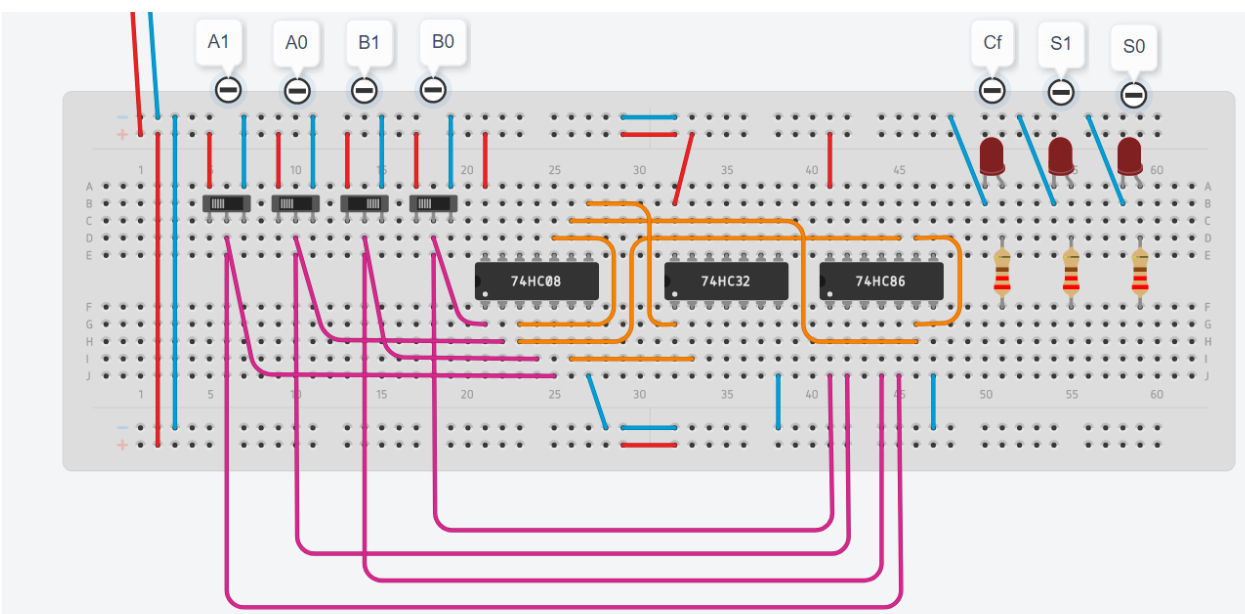


The final gate-to-gate connections are to take the two carry values from the two half adders in the full adder and to OR them together. These wires are the orange wires highlighted in yellow in the schematic to the right.



5. Connect pin 8 of the 74xx08 chip to pin 1 of the 74xx32 chip.
6. Connect pin 6 of the 74xx08 chip to pin 2 of the 74xx32 chip.

Your circuit should now look like the image below, with the final two orange wired added:

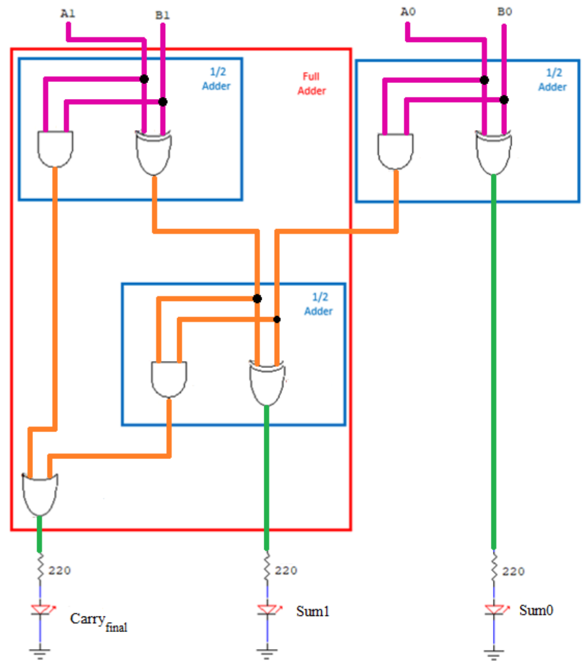


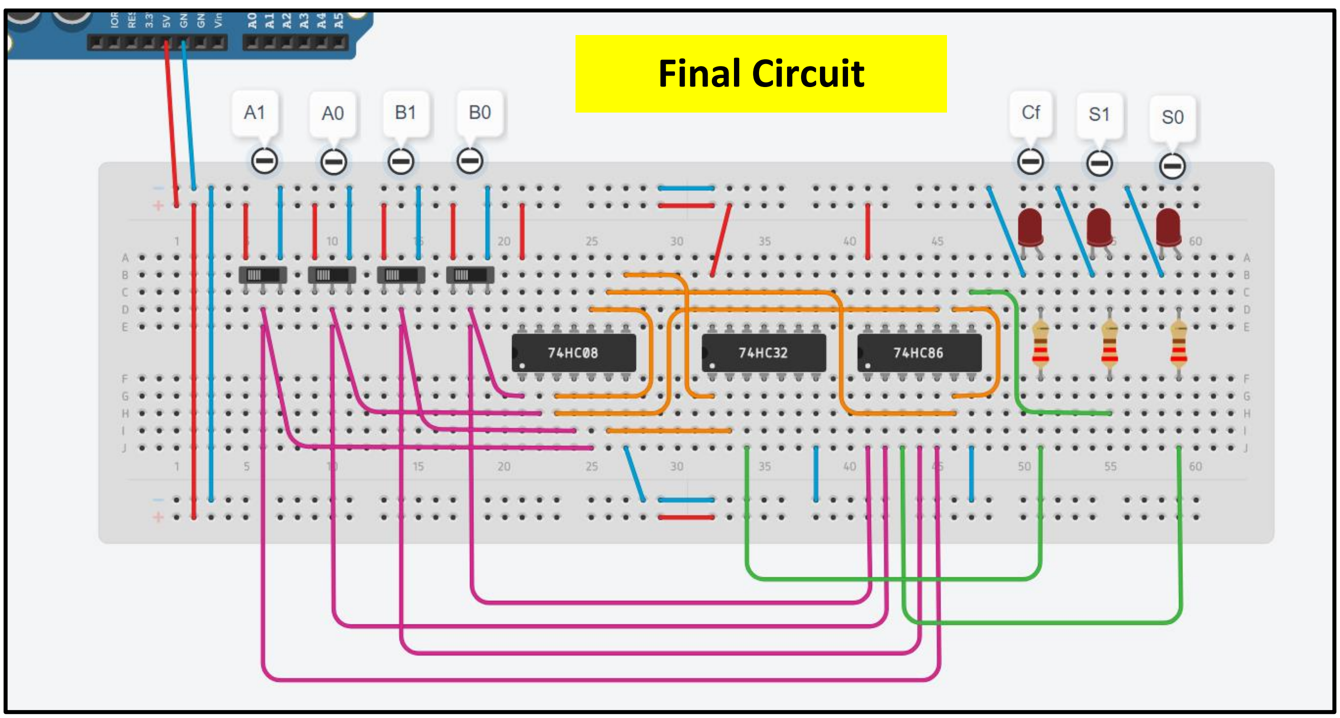
Add the Output Wires to the LEDs

The final wires to add to your circuit are the output wires going to the LEDs. These three wires are shown in green in the schematic to the right.

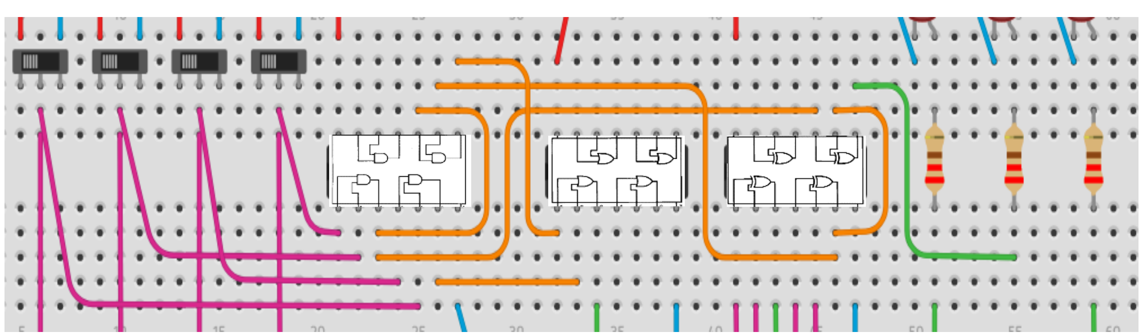
1. Connect the Sum_0 output signal to the Sum_0 LED by connecting pin 3 of the 74xx86 chip to the bottom side of the resistor for that LED.
2. Connect the Sum_1 output signal to the Sum_1 LED by connecting pin 8 of the 74xx86 chip to the bottom side of the resistor for that LED.
3. Connect the final carryout signal ($Carry_{final}$) to the $Carry_{final}$ LED by connecting pin 3 of the 74xx32 chip to the bottom side of the resistor for that LED.

Your final circuit should look like the image below.





(To visualize how the gates are connected, the image below overlays the datasheet diagrams onto the chips.)



Test the Final Two-bit Adder Circuit

1. Start the simulation by pressing "Start Simulation."
2. Open the I-Learn Lab Report quiz to record your results as you test the circuit in the Truth Table for Question 1.
3. Move each slider switch to the right (the '0' position) to make $A_1 = 0$, $A_0 = 0$, $B_1 = 0$, and $B_0 = 0$. This is for the first row of the truth table. You should see $C_f = 0$, $S_1 = 0$, and $S_0 = 0$ (all lights off). This is performing the binary addition problem 00 + 00 = 000. Record the results of the first row.
4. Move the B_0 slider switch to the left (the '1' position) to test the next row of the truth table (00 + 01). You should see that $C_f = 0$, $S_1 = 0$, and $S_0 = 1$, indicating that 00 + 01 = 001. Record the result.
5. Move the B_0 slider switch back to the right (the '0' position) and the B_1 slider switch to the left (the '1' position) to test the next row of the truth table (00 + 10). You should see that $C_f = 0$, $S_1 = 1$, and $S_0 = 0$, indicating that 00 + 10 = 010. Record the result.
6. Continue by testing each of the subsequent rows of the truth table. Verify that each input combination you test correctly performs that binary addition problem. Fill out the entire truth table in Question 1 in I-Learn.

If your circuit is producing incorrect results, do not simply put the correct results into I-Learn. The values you report in I-Learn need to be the actual results you observed from your circuit. So, if your circuit is producing incorrect results, work to fix your circuit before submitting. You may stop the simulation and compare your circuit with the final wiring diagram in the previous section to find any missing or incorrect connections. You may always ask your instructor for help. If you submit your report to I-learn and only then realized that your circuit produced the incorrect results, you may fix your circuit and resubmit.

Take a screenshot of your circuit in Tinkercad for Question 2 in the I-Learn Lab Report quiz.

You have completed the required portion of the lab. For extra credit, you may complete either or both of the following two challenges:

Extra Credit Challenge – Physical Adder (10 Extra Points)

For extra credit, do the lab physically on your breadboard with the parts from your kit. For the inputs, instead of the three-way slider switches, just move the input wires back and forth between the HIGH voltage bus and the LOW voltage bus. For the extra credit, submit a photo and truth table to the I-Learn Lab Report quiz.

Extra Credit Challenge – Simulated Subtractor (10 Extra Points)

You may also optionally convert your Tinkercad simulation of a 2-bit adder into a 2-bit subtractor ($A - B$) by adding to the circuit so that it will turn the second number (B_1B_0) into a negative number using two's complement before B goes into the adder. That will require flipping the bits of B and turning the half adder into a full adder to allow for a $CarryIn$ for the +1. For the extra credit, submit a screenshot and truth table.

Note on Extra Credit: For the extra credit truth tables in I-Learn, I-Learn will let you know if the values you entered are correct, but it will not automatically give you the extra credit. If after submitting this assignment, I-Learn indicates that you successfully entered the correct values for the extra credit Truth Tables, please message your instructor to get the extra credit points.