In a previous lesson, we saw how a half adder can be used to determine the sum and carry of two input bits. What if we have three input bits—X, Y, and Cᵢ, where Cᵢ is a carry-in that represents the carry-out from the previous less significant bit addition? In this situation, we have what is known as a FULL ADDER—a circuit that adds three one-bit values. These values are the addends X and Y, and carry-in Cᵢ.

When three single-bit values are added, the highest possible result would be 1 + 1 + 1 = 11, which is the binary representation of the decimal number 3. The entire truth table for the FULL ADDER would look like this:

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>CARRY IN Cᵢ</th>
<th>CARRY OUT Cₒ</th>
<th>SUM S</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

CₒS taken together, represents the binary sum X+Y+Cᵢ. For example, when CₒS is 11 (corresponding to decimal 3), then X+Y+Cᵢ = 1+1+1.

One way to build a FULL ADDER is to use two half adders as shown in this circuit diagram:

The half adder on the left computes the sum and carry for the addends X and Y. This sum and the carry-in are then added by the half-adder on the right, producing a final sum and a carry bit. There is a Cₒ carry out if either or both of the two carry bits are ON—explaining the use of the OR gate on the far upper right of the circuit diagram.

The FULL ADDER (FA for short) circuit can be represented in a way that hides its inner-workings:

The FULL ADDER can then be assembled into a cascade of full adders to add two binary numbers. For example the diagram below shows how one could add two 4-bit binary numbers X₃X₂X₁X₀ and Y₃Y₂Y₁Y₀ to obtain the sum S₃S₂S₁S₀ with a final carry-out C₄.

This circuit is commonly called a ripple-carry adder since the carry bit ripples from one FA to the next FA on its left. This rippling causes this adder design to be somewhat slow when compared to other more complex designs.

The initial carry-in can be hard-wired to zero, making it unnecessary for a half adder on the far right.
**ACTIVITY 1:** Construct a FULL ADDER

The above picture shows one way to construct a FULL ADDER from two half adders. The half adder on the left is essentially the half adder from the lesson on half adders. So if you still have that constructed, you can begin from that point. Inputs and outputs have been labeled in the picture to correspond to the FULL ADDER as discussed on the previous page.

Once you have completed the construction of the FULL ADDER, you should test it out to make sure that it agrees with the truth table. With X, Y, and Cᵢ all OFF, both outputs should be OFF. If any single input X, Y, or Cᵢ is ON, then S should be ON and C₀ should be off. If any pair of the three inputs is ON, then the output C₀ only should be ON. If all three inputs are ON, then both outputs should be ON.

**ACTIVITY 2:** Construct a Ripple-Carry Adder

If you are in a classroom setting, and each lab group of students has constructed a FULL ADDER, you might find it interesting and fun to connect your FULL ADDER creations into a single n-bit ripple-carry adder and then test it out by adding two n-bit binary numbers. (n is the number of FULL ADDERS that your class has connected together.)

**YOU NEED:**

1 power
1 fork
3 buttons
3 LEDs (to display input values)
4 split
2 AND
2 XOR
2 RGB LEDs (to display output values)
1 OR
3 wires (more wires may be needed if your class constructs a ripple-carry adder)