Meet Logisim

- Input tool
- Output tool
- Poke tool
- AND gate
- OR gate
- NOT gate
Add a gate

Click on the AND gate

Click on the workspace to place it

Set the number of inputs to 2
Connect the gate to some pins

Add two input pins behind the gate

Add an output pin in front of the gate

Draw the wires in (click and drag -- if you look closely, you will see little dots where wires can connect)
Use the Poke tool

Use the poke tool to change the values on the input pins

Notice that the value changed to a 1 and the wire lit up
Playing with AND

Here are the four possible combinations of our two inputs.
Adding two 1 bit numbers

We would like to build a circuit that can add two 1-bit numbers together

1  
+ 1  
----  
10

1  
+ 0  
----  
1

0  
+ 1  
----  
1

0  
+ 0  
----  
0

A  
+ B  
----  
CS

inputs

carry out  
sum
Build a truth table

We can express this as a **truth table**

\[
\begin{array}{c|c|c|c|c}
A & B & C & S \\
\hline
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 \\
\end{array}
\]
Convert to equations

Now, we can extract the minterms and write two equations, one for each output.

\[
\begin{array}{c|c|c|c|c|}
 & & & & \\
 A & B & C & S \\
\hline
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 \\
\end{array}
\]

\[
S = \overline{A}B + AB \\
C = AB
\]
Equation to gates

We will need two circuits to implement our two equations

\[ S = AB + AB \]

\[ C = AB \]

2 AND gates, 1 OR gate and 2 NOT gates

... or 1 XOR

XOR stands for exclusive OR. It is true when exactly one of the inputs is true.
Build the circuit in Logisim

Create a circuit called half_adder
click the green plus sign

Make sure that you are now in the circuit -- it should have a magnifying glass on it

Double click to change circuits
Put together the circuit

To change the orientation and add labels to the pins, use the attribute panel.

The XOR gate is hiding in here.
Build the circuit in Logisim

Make sure to test your circuit
What if the numbers have more than 1 bit?

If we add another bit to each, we have four times the number of possible equations.

```
01 01 00 00
+ 01 +00 +01 +00
---- ---- ---- ----
010 001 001 000

11 11 10 10
+ 01 +00 +01 +00
---- ---- ---- ----
100 011 011 010
```

```
01 01 00 00
+ 11 +10 +11 +10
---- ---- ---- ----
100 011 011 010

11 11 10 10
+ 11 +10 +11 +10
---- ---- ---- ----
110 101 101 100
```
Looking at the second column

The second column is not the same -- it has to add three numbers instead of two

just looking at the second column

addition is associative

... and we already know how to add two 1 bit numbers together
Making a full adder

The Carry_out is just the OR of the two carries from the half adders because it will never be the case that both half adders produce one (check it yourself)
Make a full adder in Logisim

Create a new circuit and call it full_adder

Add two half_adders to the circuit
    just click the half_adder once like it was a gate and then click in your workspace

Add an OR gate

Hook it all together
Putting it all together

With a half adder and a couple of full adders, we can make something called **ripple-carry adder** called that because any carries generated in the first column can ripple up to the last one.
Build a 4-bit adder

• Double click on the main circuit

• Recreate the 4-bit adder from the previous page using three full adders and one half adder

• Add the 8 input pins and five output pins and label them AO, A1, A2, A3, B0, B1, B2, B3, S0, S1, S2, S3, carry_out
  • [note - the wires may not connect in exactly the same place as shown in the diagram]

• Test, test, test