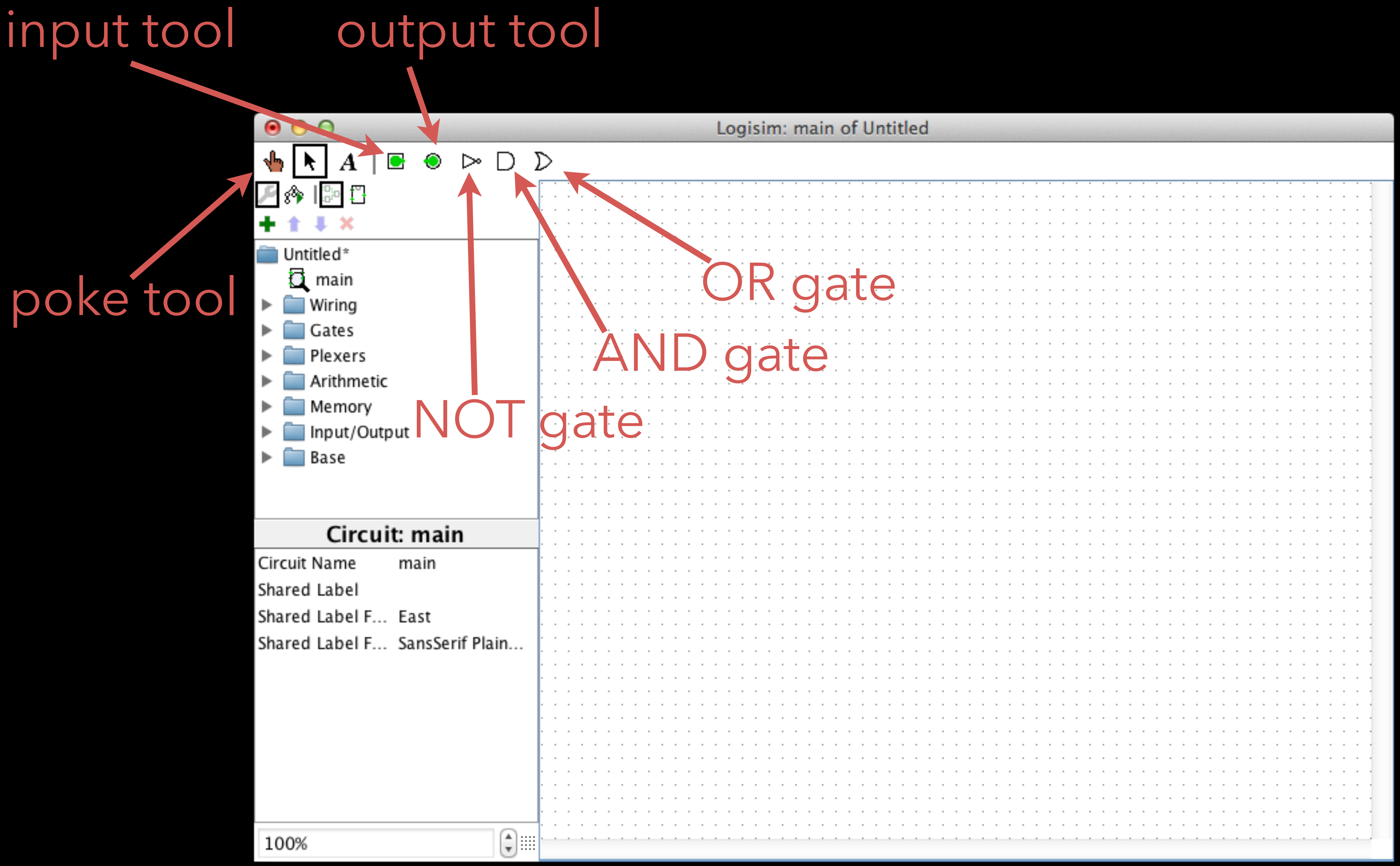
Meet Logisim



input tool

output tool

poke tool

OR gate

AND gate

NOT gate

Logisim: main of Untitled

Untitled*

main

Wiring

Gates

Plexers

Arithmetic

Memory

Input/Output

Base

Circuit: main

Circuit Name main

Shared Label

Shared Label F... East

Shared Label F... SansSerif Plain...

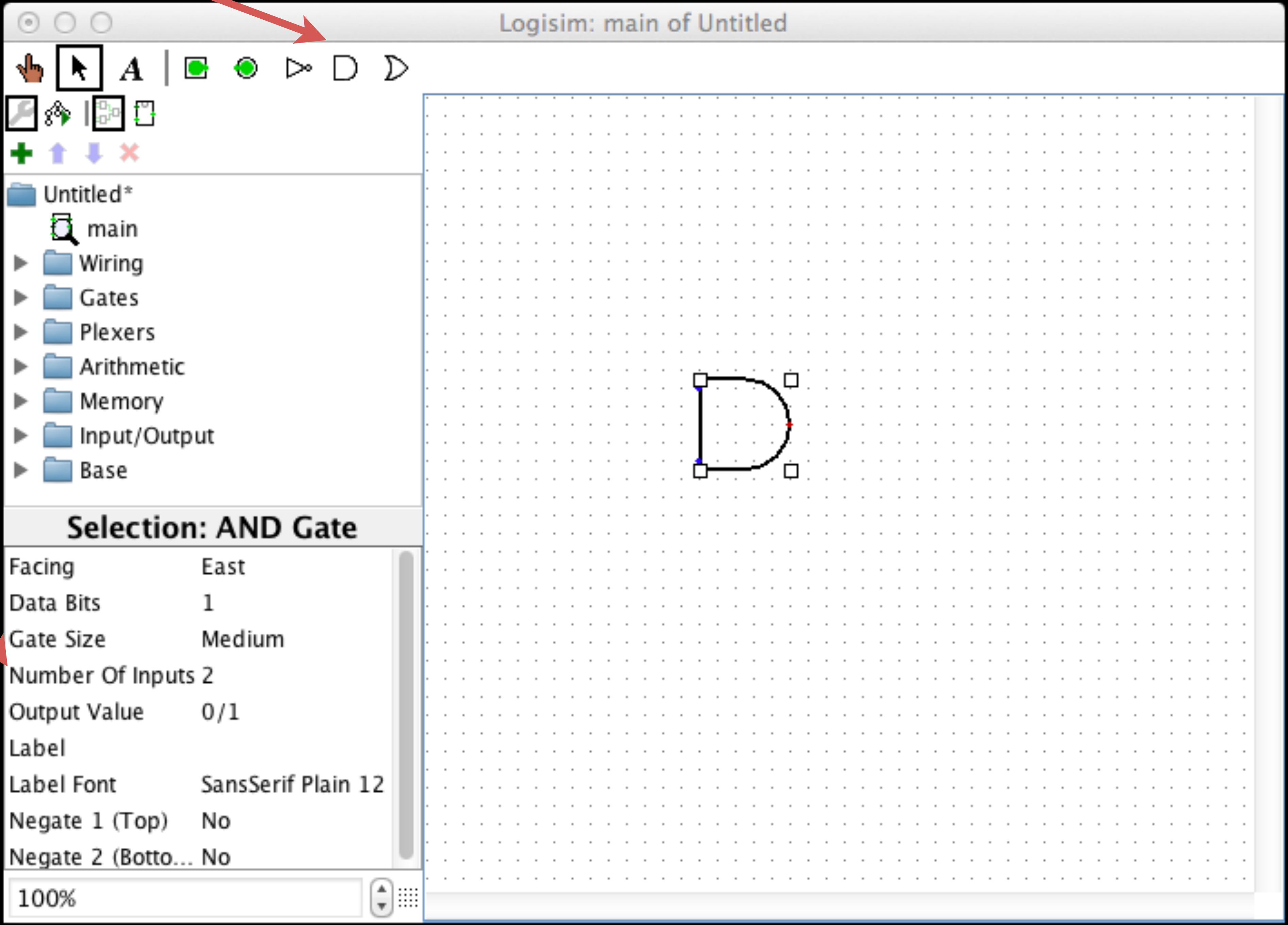
100%

Add a gate

Click on the AND gate

Click on the workspace to place it

Set the number of inputs to 2



Selection: AND Gate

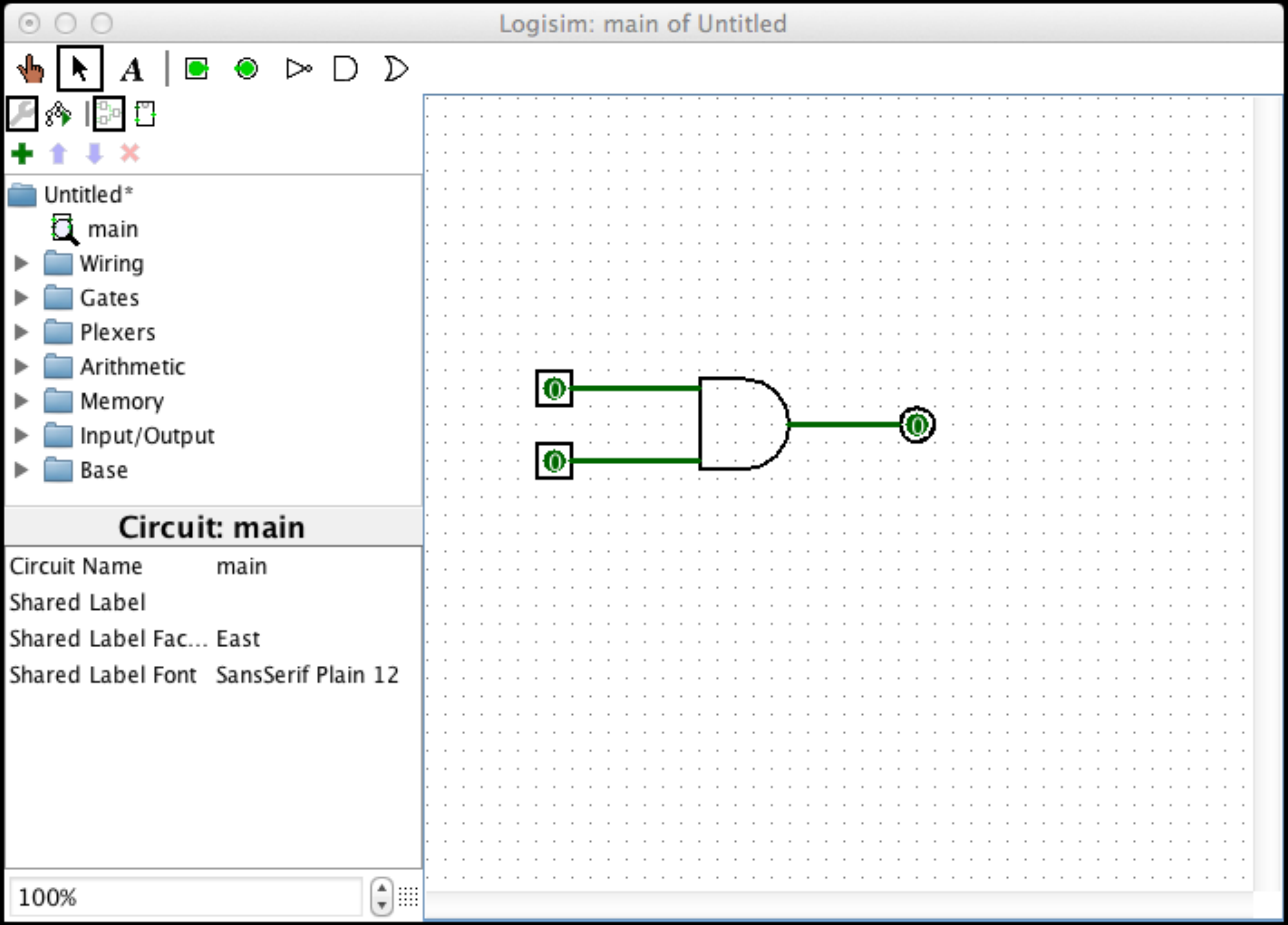
Facing	East
Data Bits	1
Gate Size	Medium
Number Of Inputs	2
Output Value	0/1
Label	
Label Font	SansSerif Plain 12
Negate 1 (Top)	No
Negate 2 (Botto...	No

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)



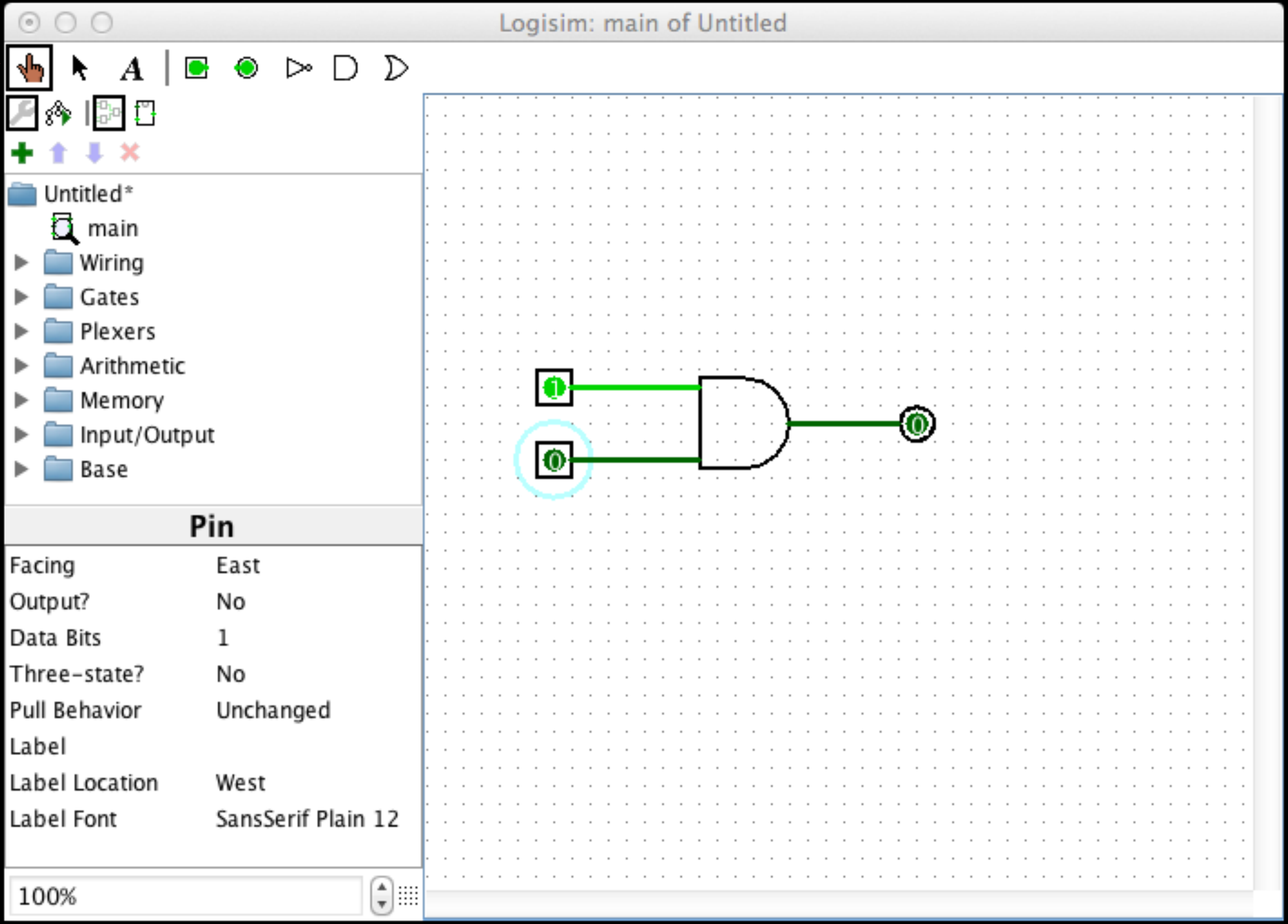
Circuit: main

Circuit Name	main
Shared Label	
Shared Label Fac...	East
Shared Label Font	SansSerif Plain 12

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



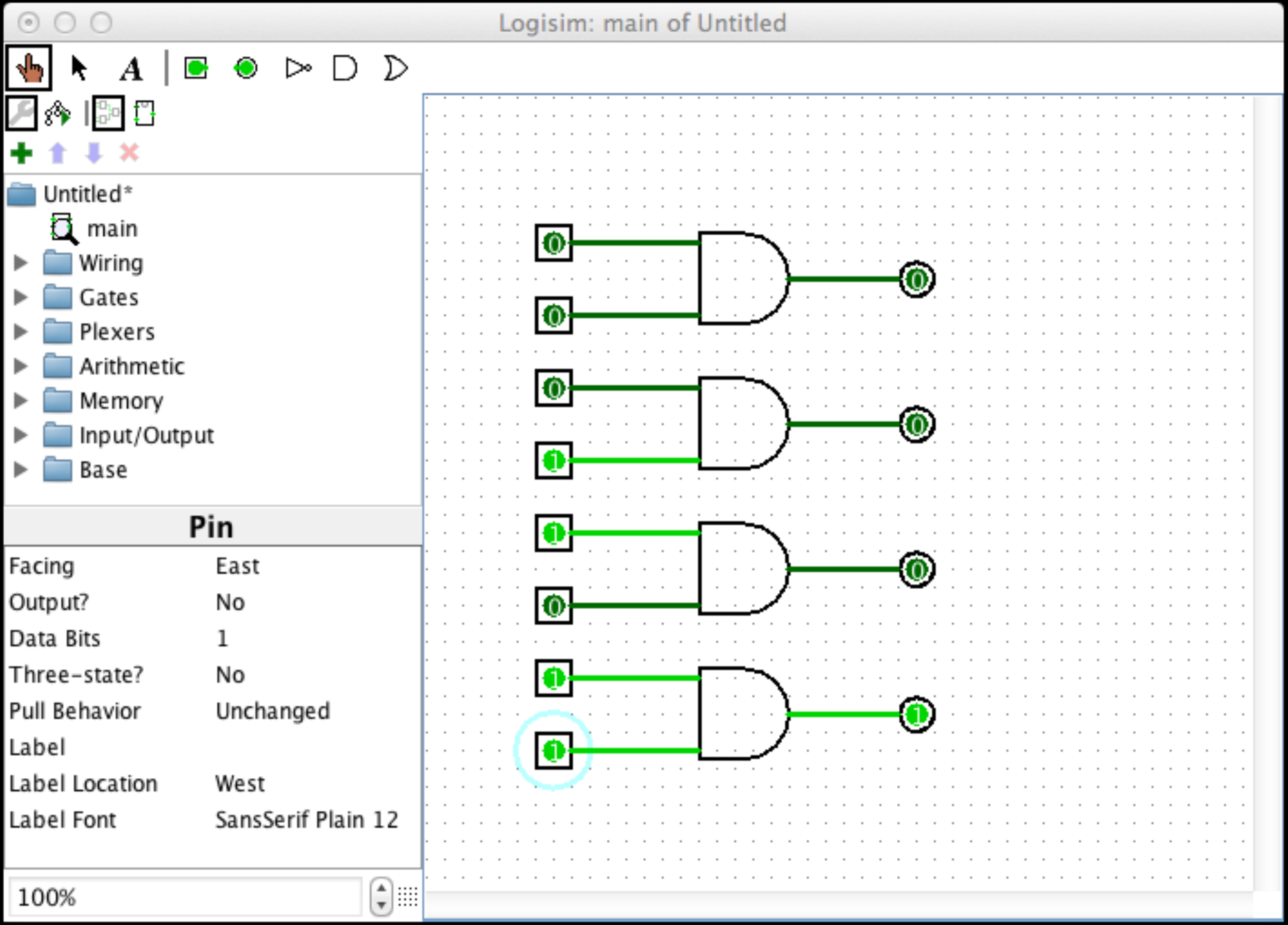
Logisim: main of Untitled

Pin

Facing	East
Output?	No
Data Bits	1
Three-state?	No
Pull Behavior	Unchanged
Label	
Label Location	West
Label Font	SansSerif Plain 12

Playing with AND

Here are the four possible combinations of our two inputs.



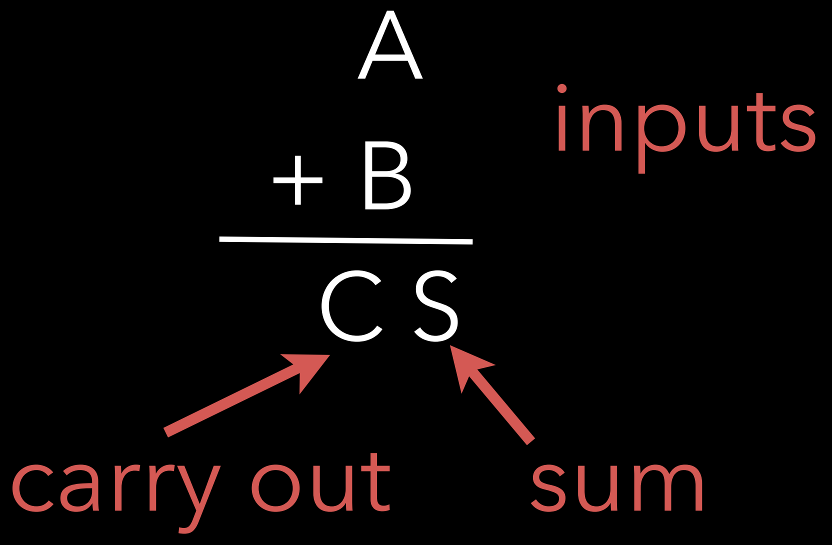
Pin

Facing	East
Output?	No
Data Bits	1
Three-state?	No
Pull Behavior	Unchanged
Label	
Label Location	West
Label Font	SansSerif Plain 12

Adding two 1 bit numbers

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

$$\begin{array}{r} 1 \\ +\ 1 \\ \hline 10 \end{array} \qquad \begin{array}{r} 1 \\ +\ 0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 0 \\ +\ 1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 0 \\ +\ 0 \\ \hline 0 \end{array}$$



$$\begin{array}{r} A \\ +\ B \\ \hline C\ S \end{array}$$

inputs

carry out

sum

Build a truth table

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

$$\begin{array}{r} 1 \\ +\ 1 \\ \hline 10 \end{array} \qquad \begin{array}{r} 1 \\ +\ 0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 0 \\ +\ 1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 0 \\ +\ 0 \\ \hline 0 \end{array}$$

A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$\begin{array}{r} A \\ +\ B \\ \hline C\ S \end{array}$$

Convert to equations

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

$$\begin{array}{r} 1 \\ +\ 1 \\ \hline 10 \end{array} \qquad \begin{array}{r} 1 \\ +\ 0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 0 \\ +\ 1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 0 \\ +\ 0 \\ \hline 0 \end{array}$$

A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$S = \overline{A}B + A\overline{B}$$

$$C = AB$$

Equation to gates

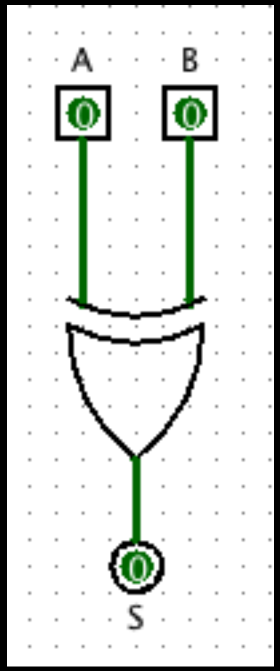
We will need two circuits to implement our two equations

$S = \overline{A}B + A\overline{B}$

$C = AB$

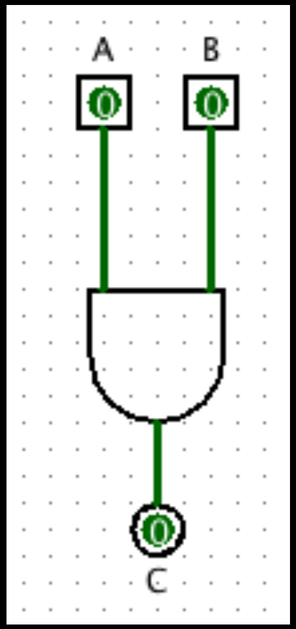
2 AND gates,
1 OR gate and
2 NOT gates

... or 1 XOR



XOR stands for exclusive OR. It is true when exactly one of the inputs is true.

1 AND gate



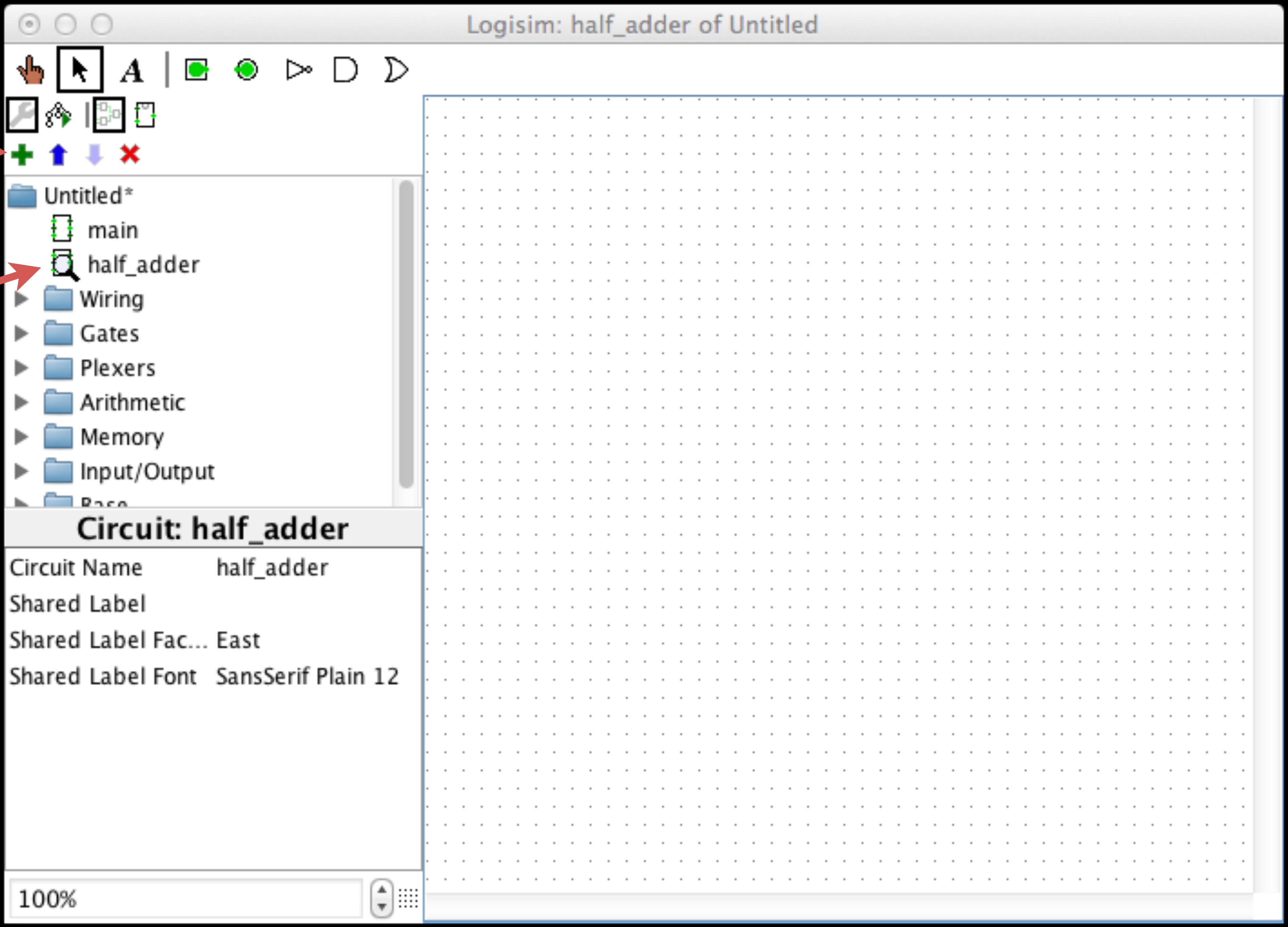
Build the circuit in Logisim

Create a circuit called half_adder

click the green plus sign

click here

new circuit will show up here



Logisim: half_adder of Untitled

Untitled*
- main
- half_adder
- Wiring
- Gates
- Plexers
- Arithmetic
- Memory
- Input/Output
- Base

Circuit: half_adder

Circuit Name	half_adder
Shared Label	
Shared Label Fac...	East
Shared Label Font	SansSerif Plain 12

100%

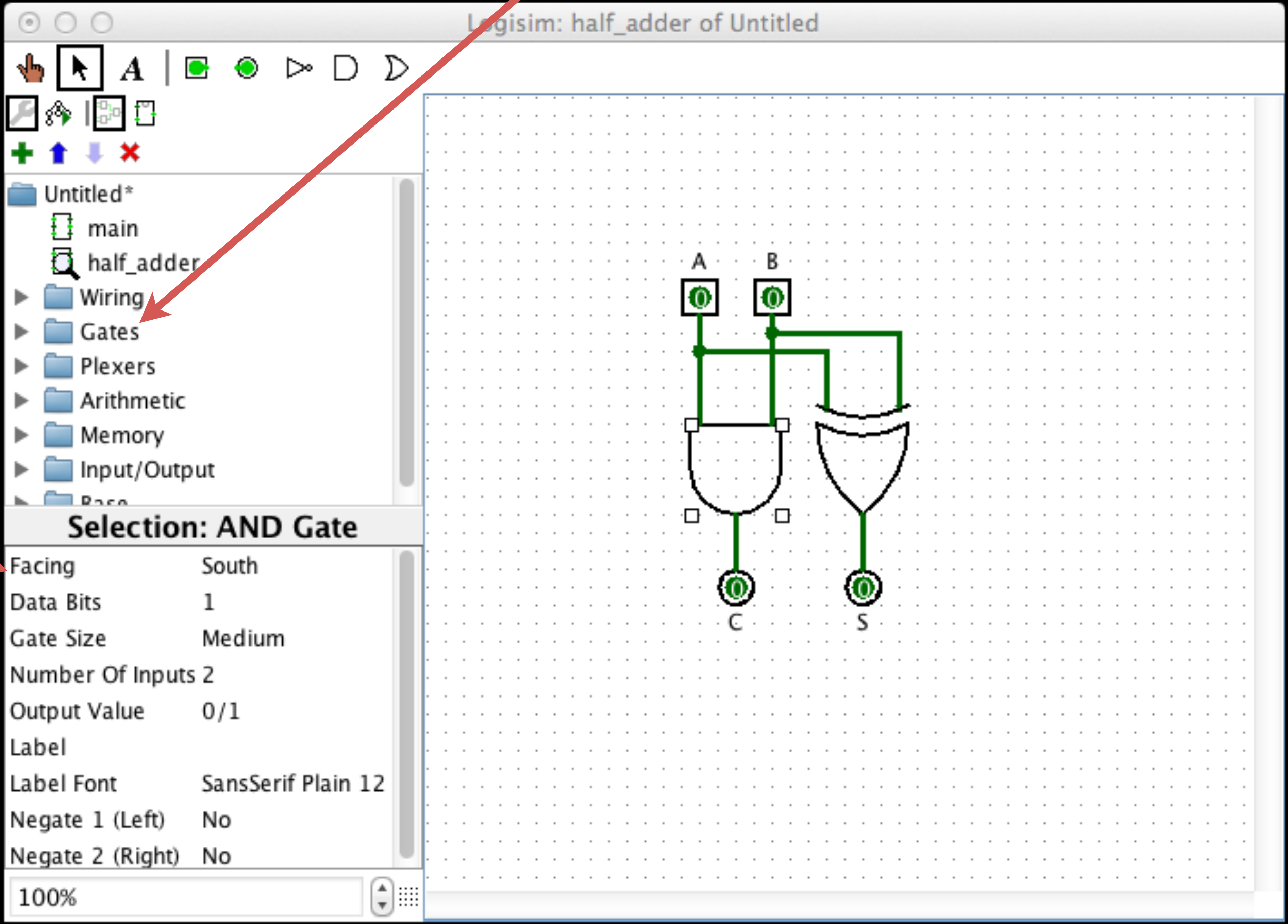
Make sure that you are now in the circuit -- it should have a magnifying glass on it

Double click to change circuits

Put together the circuit

The XOR gate is hiding in here

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

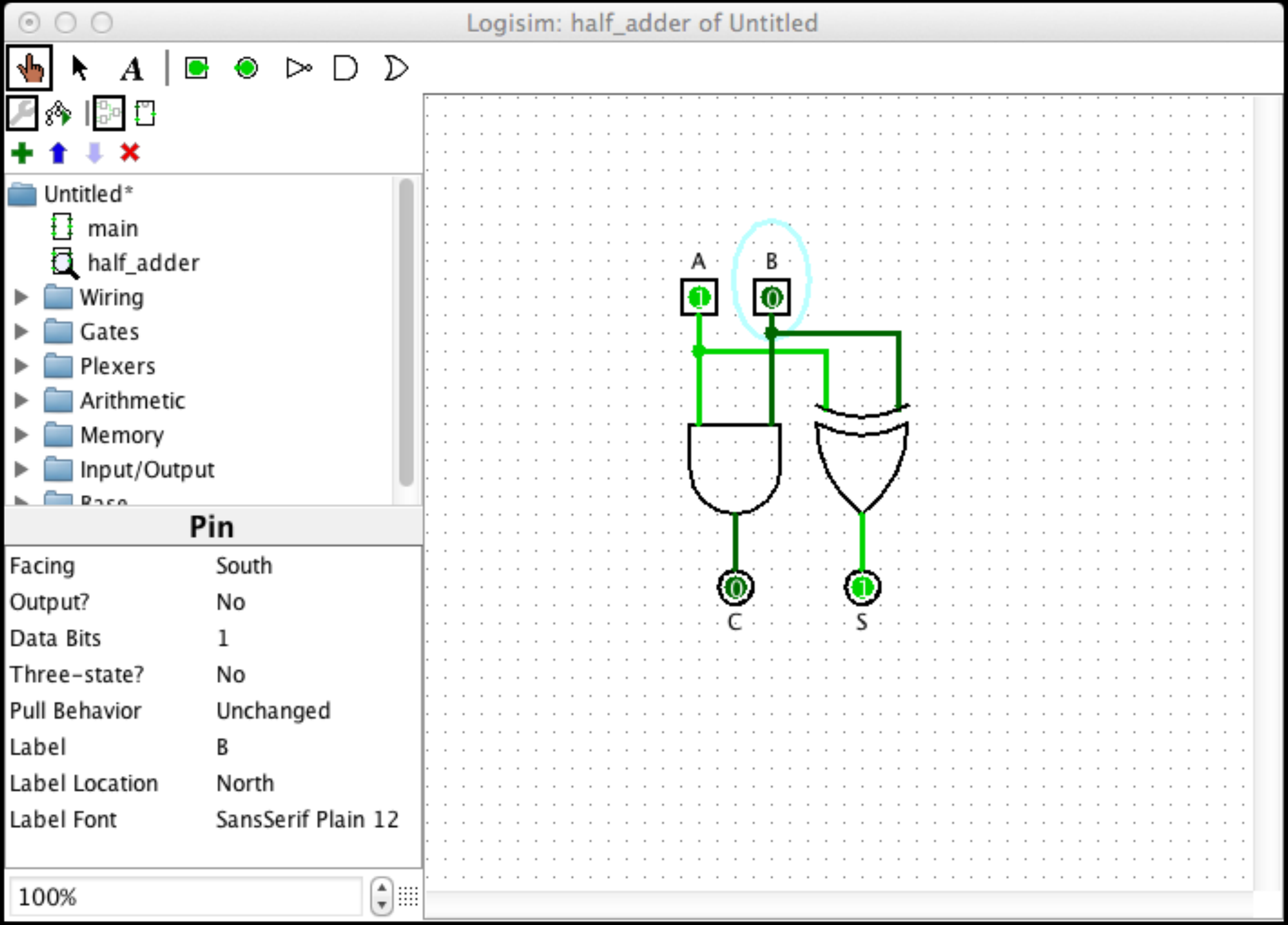


Selection: AND Gate

Facing	South
Data Bits	1
Gate Size	Medium
Number Of Inputs	2
Output Value	0/1
Label	
Label Font	SansSerif Plain 12
Negate 1 (Left)	No
Negate 2 (Right)	No

Build the circuit in Logisim

Make sure to test your circuit



Pin

Facing	South
Output?	No
Data Bits	1
Three-state?	No
Pull Behavior	Unchanged
Label	B
Label Location	North
Label Font	SansSerif Plain 12

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

$$\begin{array}{r} 01 \\ +\ 01 \\ \hline 010 \end{array} \quad \begin{array}{r} 01 \\ +\ 00 \\ \hline 001 \end{array} \quad \begin{array}{r} 00 \\ +\ 01 \\ \hline 001 \end{array} \quad \begin{array}{r} 00 \\ +\ 00 \\ \hline 000 \end{array} \qquad \begin{array}{r} 11 \\ +\ 01 \\ \hline 100 \end{array} \quad \begin{array}{r} 11 \\ +\ 00 \\ \hline 011 \end{array} \quad \begin{array}{r} 10 \\ +\ 01 \\ \hline 011 \end{array} \quad \begin{array}{r} 10 \\ +\ 00 \\ \hline 010 \end{array}$$

$$\begin{array}{r} 01 \\ +\ 11 \\ \hline 100 \end{array} \quad \begin{array}{r} 01 \\ +\ 10 \\ \hline 011 \end{array} \quad \begin{array}{r} 00 \\ +\ 11 \\ \hline 011 \end{array} \quad \begin{array}{r} 00 \\ +\ 10 \\ \hline 010 \end{array} \qquad \begin{array}{r} 11 \\ +\ 11 \\ \hline 110 \end{array} \quad \begin{array}{r} 11 \\ +\ 10 \\ \hline 101 \end{array} \quad \begin{array}{r} 10 \\ +\ 11 \\ \hline 101 \end{array} \quad \begin{array}{r} 10 \\ +\ 10 \\ \hline 100 \end{array}$$

Looking at the second column

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

$$\begin{array}{r} 1 \\ 11 \\ +\ 11 \\ \hline 110 \end{array}$$

$$\begin{array}{r} 1 \\ 1 \\ +\ 1 \\ \hline 11 \end{array}$$

just looking at the second column

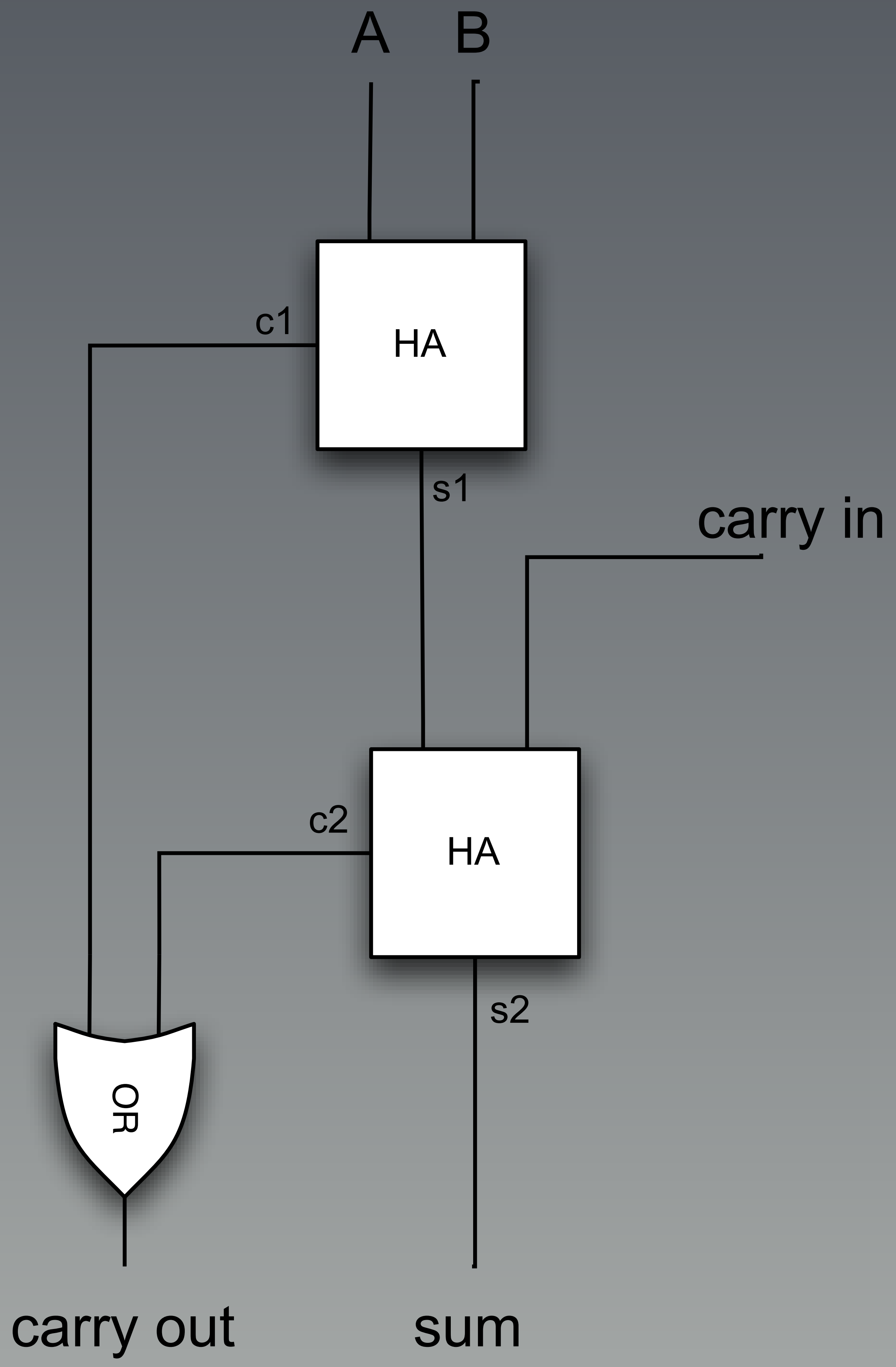
$$\begin{array}{r} 1 \\ +\ 1 \\ \hline 10 \\ +\ 1 \\ \hline 11 \end{array}$$

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

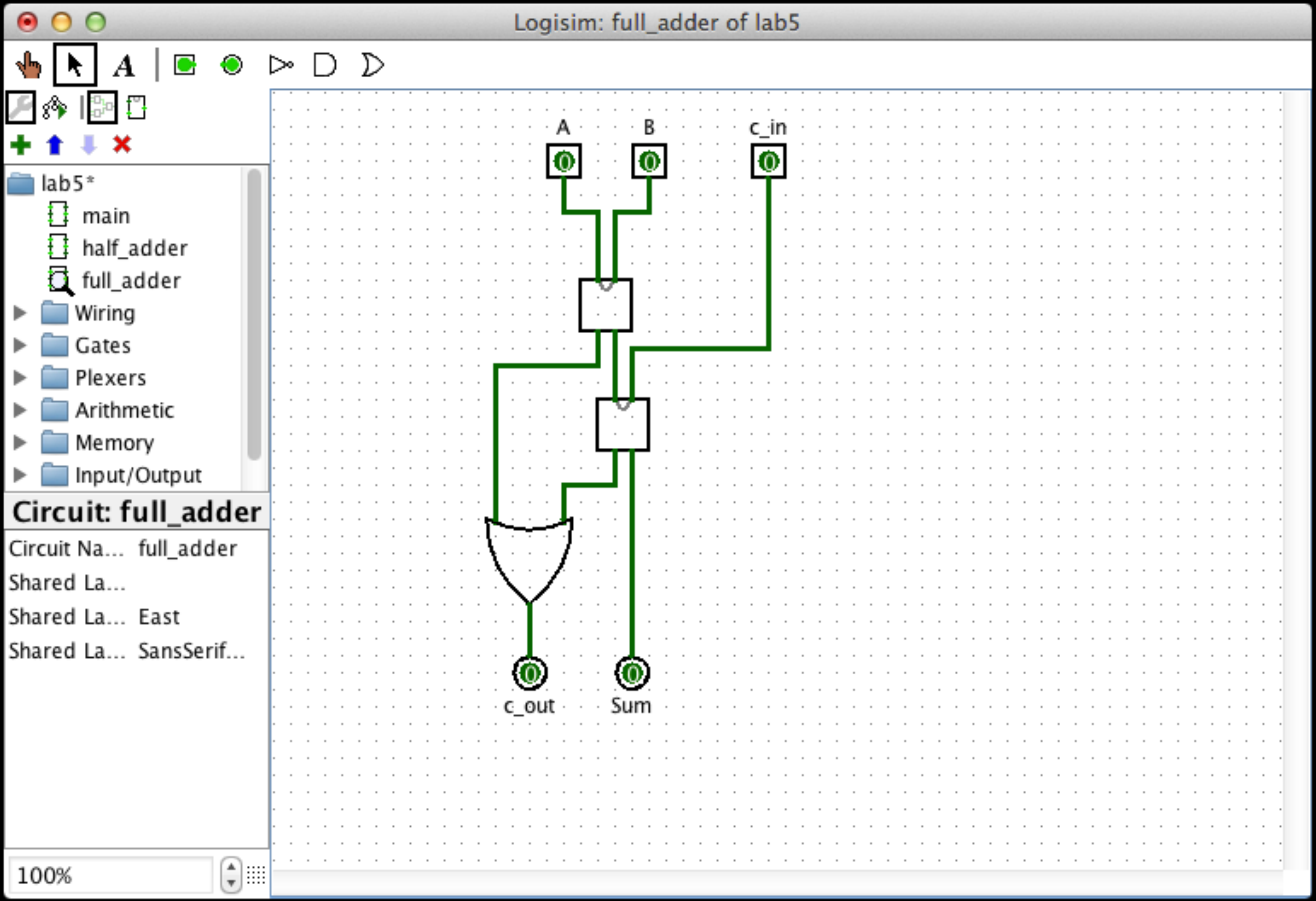
Create a new circuit and call it full_adder

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



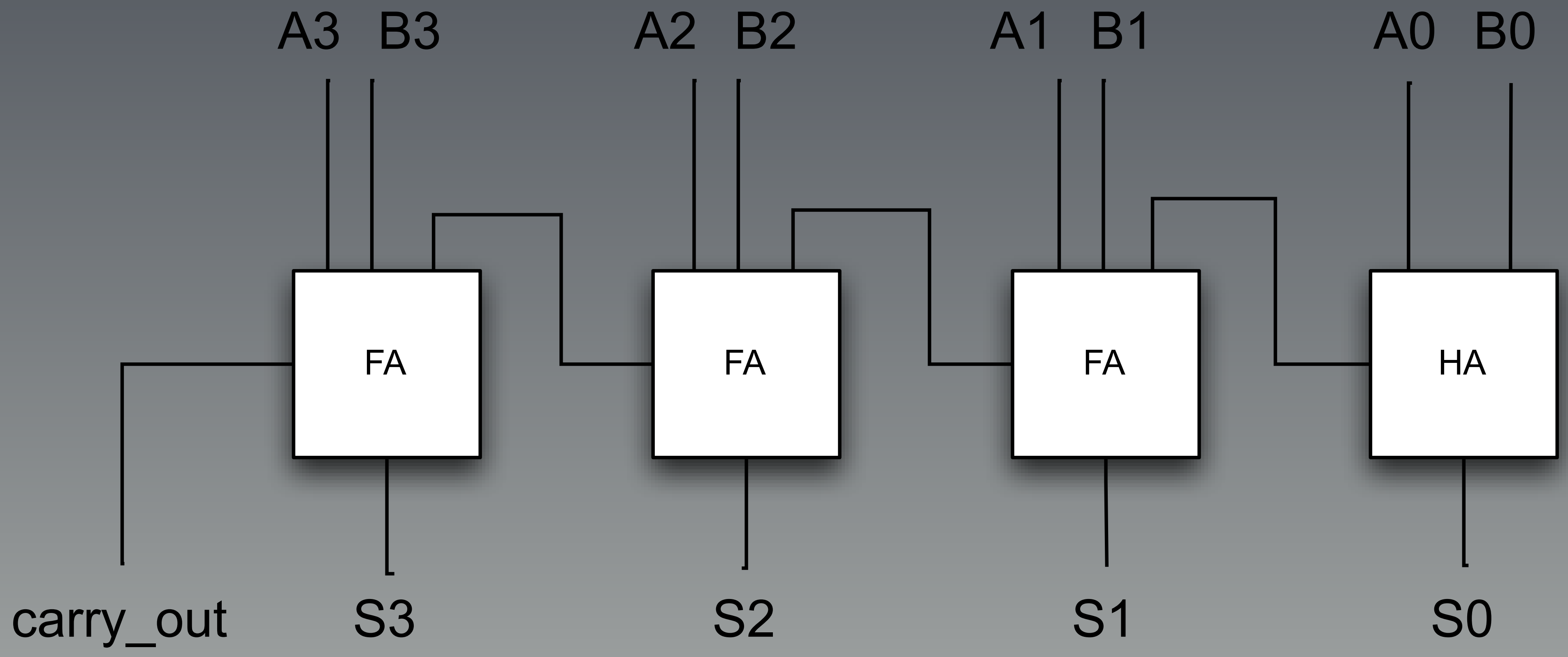
Circuit: full_adder

Circuit Na...	full_adder
Shared La...	
Shared La...	East
Shared La...	SansSerif...

Putting it all together

With a half adder an 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