LEARN AND FIX · TUTORIAL · CHECKED ON 5.0.0

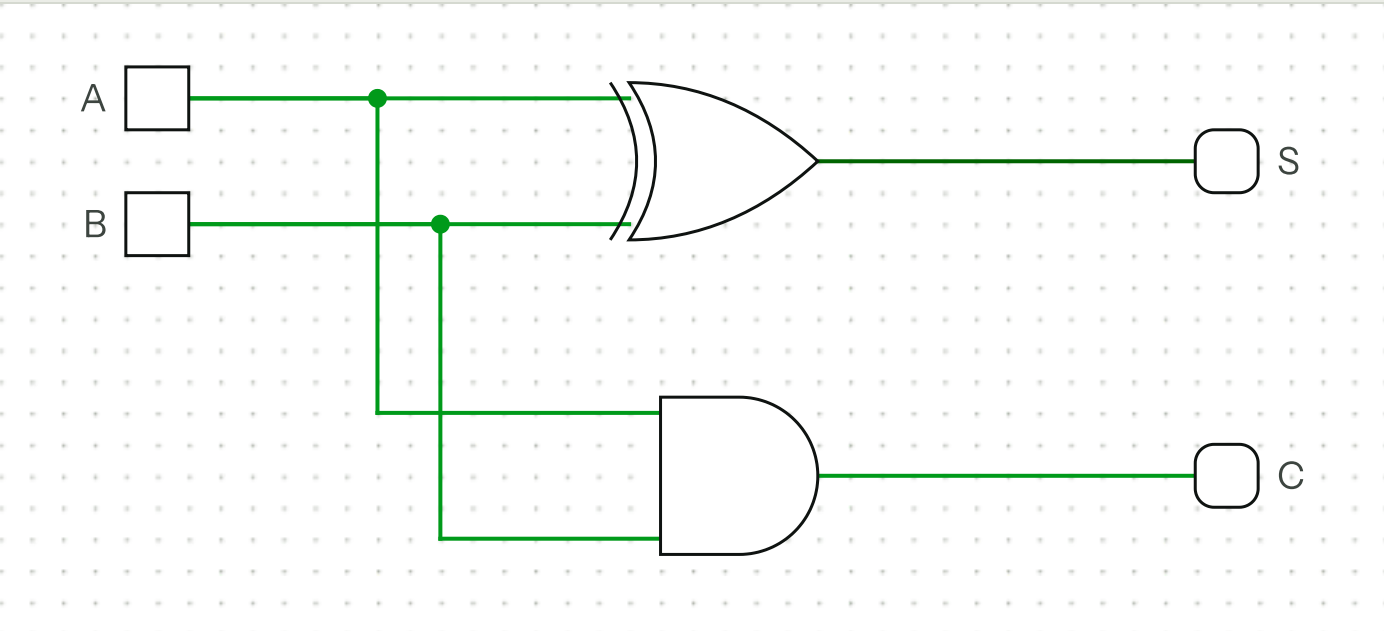
Your first Logisim circuit, step by step

1. Open Logisim-evolution, drag two input pins, one XOR gate, one AND gate and two output pins onto the canvas, and drag from port to port to wire them. Click the hand icon (the Poke tool) and click the inputs to switch them between 0 and 1: the outputs change at once, because the simulation is always running. Save with Ctrl+S; the file is a `.circ`. You can download our finished file to compare.

FOR Beginners, Logisim-evolution 5.0.0 CHECKED 11 Nov 2026

FIX TIME about 15 min

CIRC-TUTORIAL-HALF-ADDER · MAIN



SAVED WITH ■ 5.0.0 T-PROJ-TUTORIAL ↗ Open in .circ Viewer ↗

This is what you will have at the end: a half adder. With A = 1 and B = 1 the sum S is 0 and the carry C is 1.

CC BY 4.0 · SHA-256 CC7C23C201C8

DESCRIBE THIS CIRCUIT

Components: Input pin A; Input pin B; XOR Gate 1; AND Gate 1; Output pin S; Output pin C.

- Input pin A connects to XOR Gate 1, AND Gate 1 (net A).
- Input pin B connects to XOR Gate 1, AND Gate 1 (net B).
- XOR Gate 1 connects to Output pin S (net S).
- AND Gate 1 connects to Output pin C (net C).

≠ **Using classic 2.7.1?** 2.7.1

The steps are the same except where a version note says otherwise. See Logisim 2.7.1 ↗ and the differences at the end of this page.

1 **What we tested for this page**

We opened the finished circuit in Logisim-evolution 5.0.0 and used the Poke tool, the Analyze Circuit window and the save and reopen behaviour (F83, F45). We have not yet timed a first build by hand with the mouse on a developer machine, so the shots below show the finished circuit, not each building step.

LINUX LAB · LOGISIM-EVOLUTION 5.0.0 · 11 NOV 2026 · F83, F45

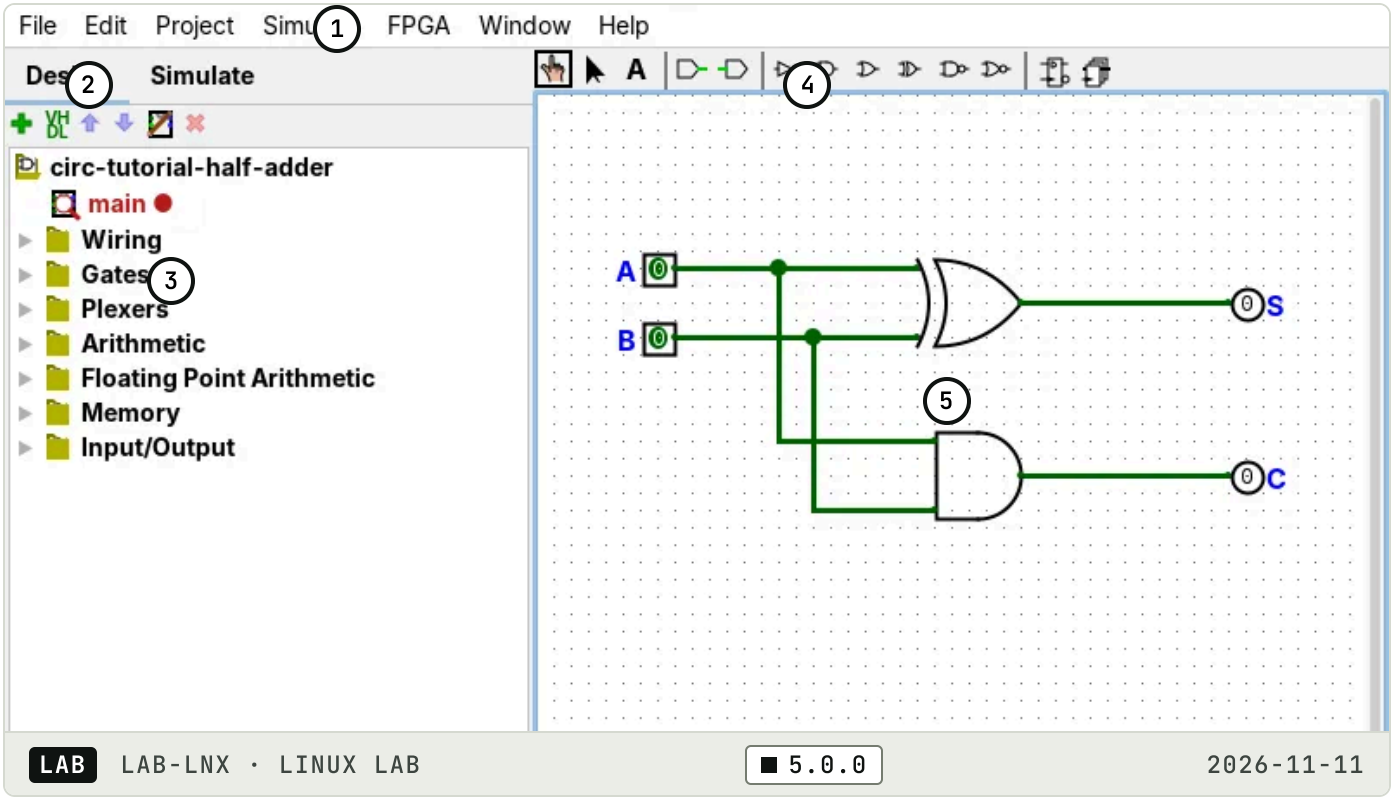
Before you start

YOU NEED 0 / 3 DONE

- [ ] Logisim-evolution 5.0.0 installed. Installers include Java, so you do not need to install Java yourself.
- [ ] To know which version your course asks for. If it names 2.7.1, use that and read the differences at the end.
- [ ] About 15 minutes.

If you do not have it yet, download Logisim safely ↗. If it will not open, see Logisim will not open ↗. Not sure which version? See Logisim vs Logisim-evolution ↗.

A 60-second tour of the window



The window with the finished half adder open. Linux lab machine, 5.0.0.

`LAB-LNX-LE500-TUTORIAL-WINDOW-01`

1. **Menu bar:** File, Edit, Project, Simulate, FPGA, Window, Help.
2. **Design and Simulate tabs** above the component tree.
3. **Component tree:** the libraries Wiring, Gates, Plexers, Arithmetic, Floating Point Arithmetic, Memory and Input/Output hold the parts you can place.
4. **Toolbar:** Poke (the hand), Edit (the arrow), Text, then input and output pins, the basic gates, a flip-flop and a register.
5. **Canvas** with a dot grid: this is where you draw. The Properties panel under the tree shows the settings of whatever is selected.

The full tour is on Interface and shortcuts ↗.

Build a half adder

A half adder adds two bits. The sum is A XOR B and the carry is A AND B.

1 Name your circuit and save it first

Choose File, Save As… (`Ctrl` + `Shift` + `S` (`⌘` + `⇧` + `S` on macOS)) and name it `half-adder.circ`. Saving first means you never lose work to a crash. Logisim-evolution also has autosave, but save by hand anyway.

1 **Check:** A file named half-adder.circ exists.

2 Place two input pins

Click the Input pin tool in the toolbar, then click the canvas twice, one pin above the other. Give them labels now: select a pin with the Edit tool and type `A` and `B` in the Label field of the Properties panel.

> **Label the pins now**
> Test vectors and autograders compare labels exactly, including capital letters. A and B are different from a and b.

1 **Check:** Two input pins are on the canvas.

3 Place an XOR gate and an AND gate

Click the XOR Gate tool in the toolbar and click the canvas. Do the same with the AND Gate tool, below the first. Both have two inputs by default; the Number of Inputs attribute changes that later (see Logic gates ↗). The small dots on a gate are its ports: wires connect there.

1 **Check:** An XOR gate and an AND gate are on the canvas, to the right of the pins.

4 Place two output pins

Click the Output pin tool and click the canvas twice. Label them `S` for sum and `C` for carry.

1 **Check:** Two output pins are to the right of the gates, labelled S and C.

5 Wire it

Drag from a port to another port. A wire is straight, so a bend needs two drags. Wire A and B each to one input of the XOR and one input of the AND; wire the XOR output to S and the AND output to C. A dot appears where three wires meet.

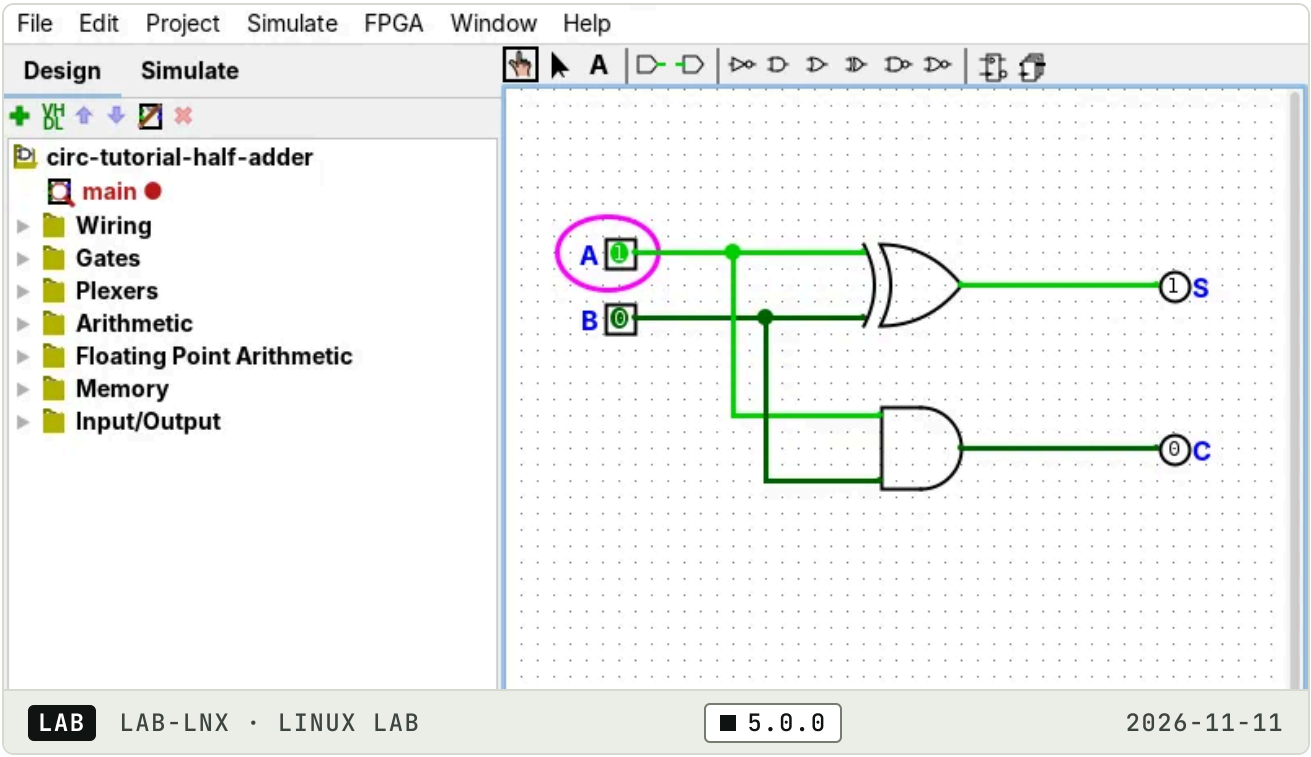
> **! A wire that stops short is not connected**
>
> A wire that ends near a port but not on it stays unconnected: the output reads `U` and the wire is blue. See what blue means ↗.

1 **Check:** Every pin and gate port is joined by a wire.

Wire colours: blue, red, orange ↗

6 Test it with the Poke tool

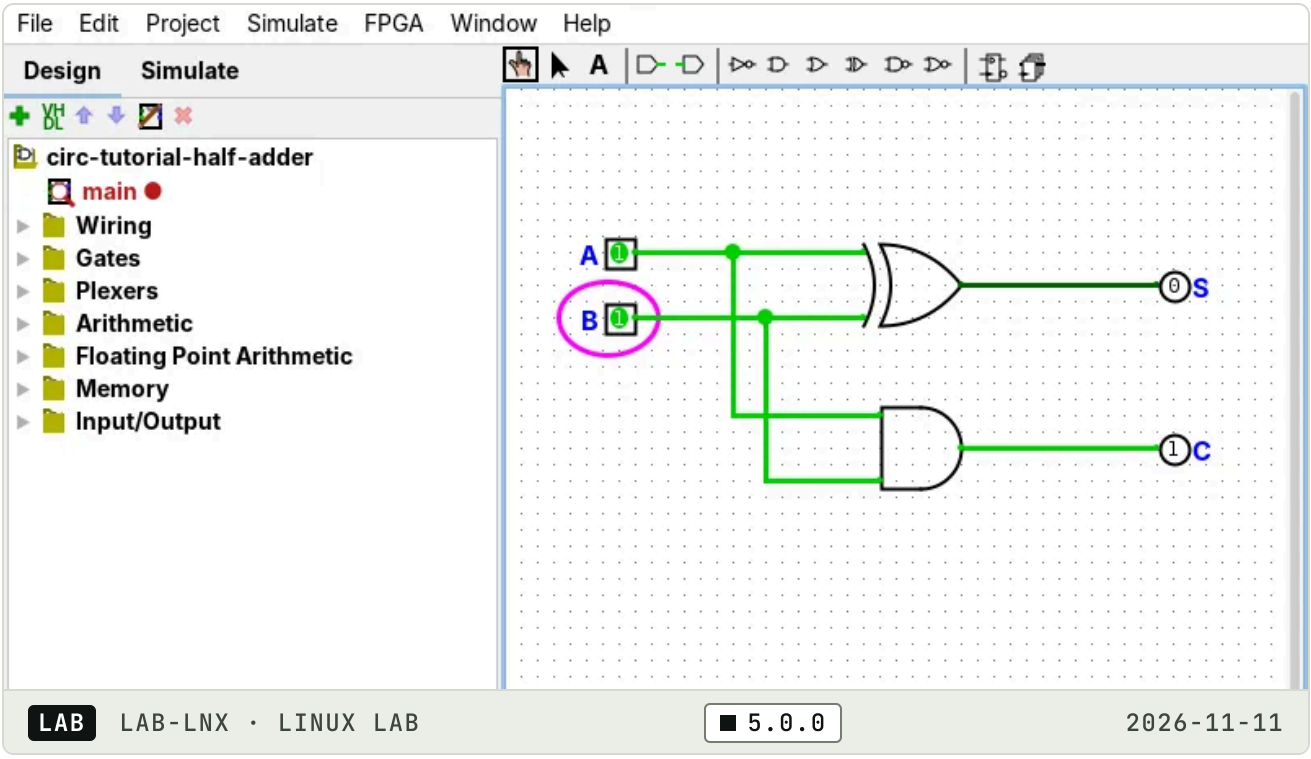
Click the hand icon (the Poke tool), then click input A. In our test one click on A changed it from 0 to 1, and S followed at once. Logisim draws a magenta ring where you clicked.



One Poke click on A: A = 1, S = 1, C = 0 (F83).

`LAB-LNX-LE500-TUTORIAL-POKE-A-01`

Click B as well: S goes back to 0 and C becomes 1.



Both inputs 1: S = 0, C = 1 (F83).

`LAB-LNX-LE500-TUTORIAL-POKE-AB-01`

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

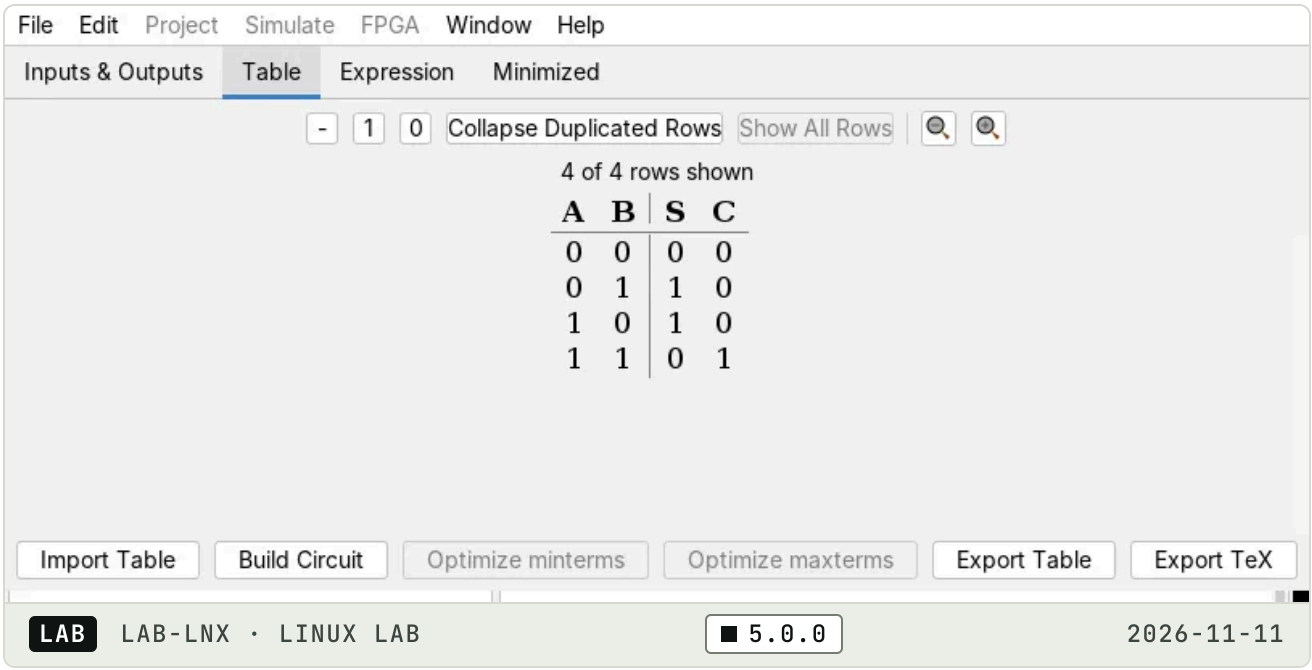
CHECK EACH ROW 0 / 4 DONE

- [ ] A = 0, B = 0 gives S = 0, C = 0
- [ ] A = 0, B = 1 gives S = 1, C = 0
- [ ] A = 1, B = 0 gives S = 1, C = 0
- [ ] A = 1, B = 1 gives S = 0, C = 1

1 **Check:** The outputs match the four rows below.

7 Check with the automatic truth table

Choose Project, Analyze Circuit. A second window opens with the tabs Inputs & Outputs, Table, Expression and Minimized. The Table tab lists the four rows without you poking anything; the Expression tab shows S = A⊕B and C = A·B (F83). See Combinational Analysis ↗.



Project, Analyze Circuit, Table tab: the same four rows (F83).

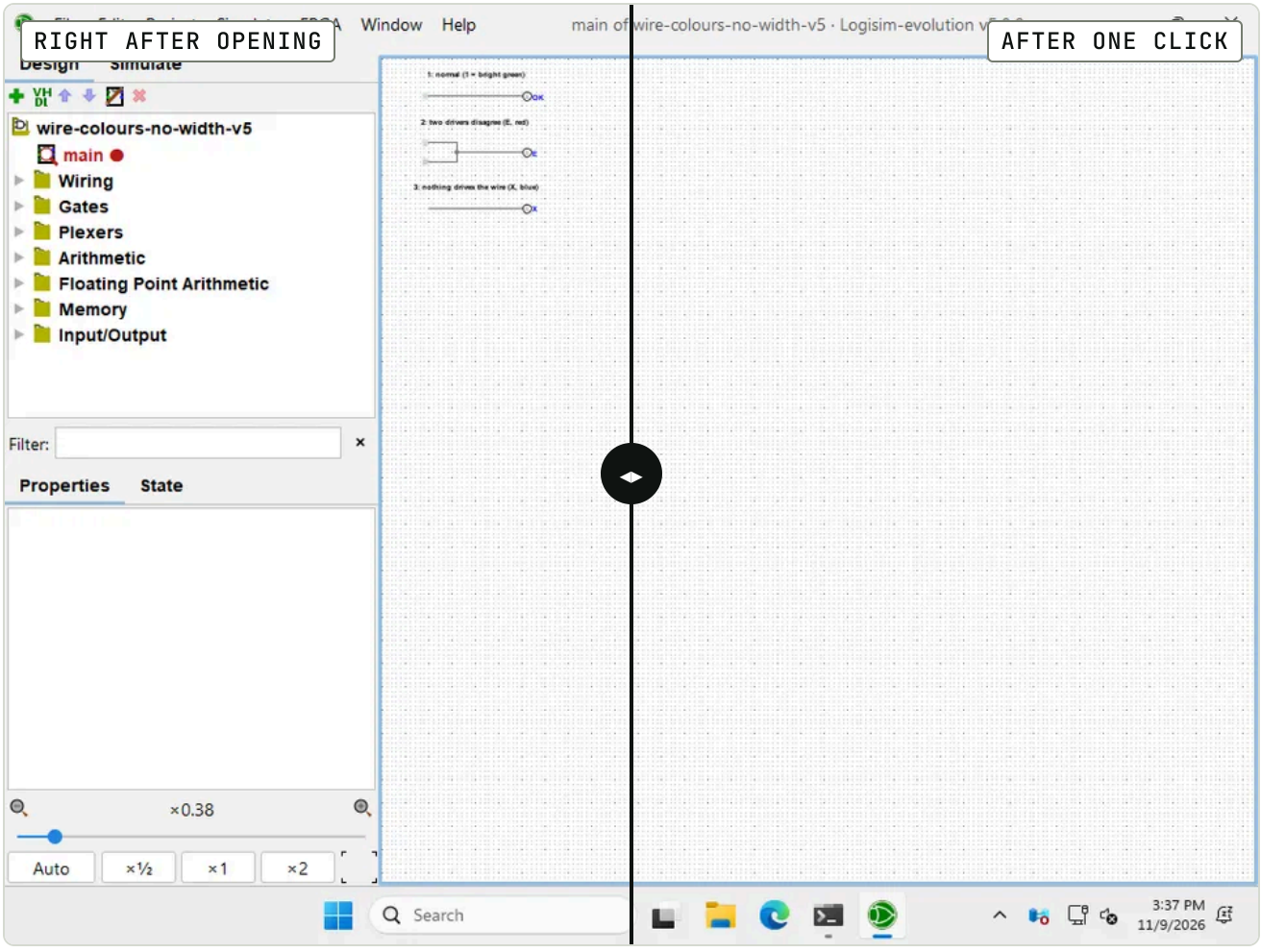
`LAB-LNX-LE500-ANALYZE-TABLE-01`

1 **Check:** The Table tab shows the same four rows.

8 Save and reopen

Save with Ctrl + S (⌘ + S on macOS), close the file and open it again from File, Open Recent.

> **Grey wires after opening**
>
> When the file opens, the wires are grey and the pins show 0. Click once on the canvas or press Ctrl + R (⌘ + R on macOS) (Simulate, Reset Simulation) and the colours come back. This happens with files saved by 5.0.0 itself, on Linux, on a Windows desktop and on macOS (F45, F49, F64). Moving the mouse does not fix it. See grey wires ↗.

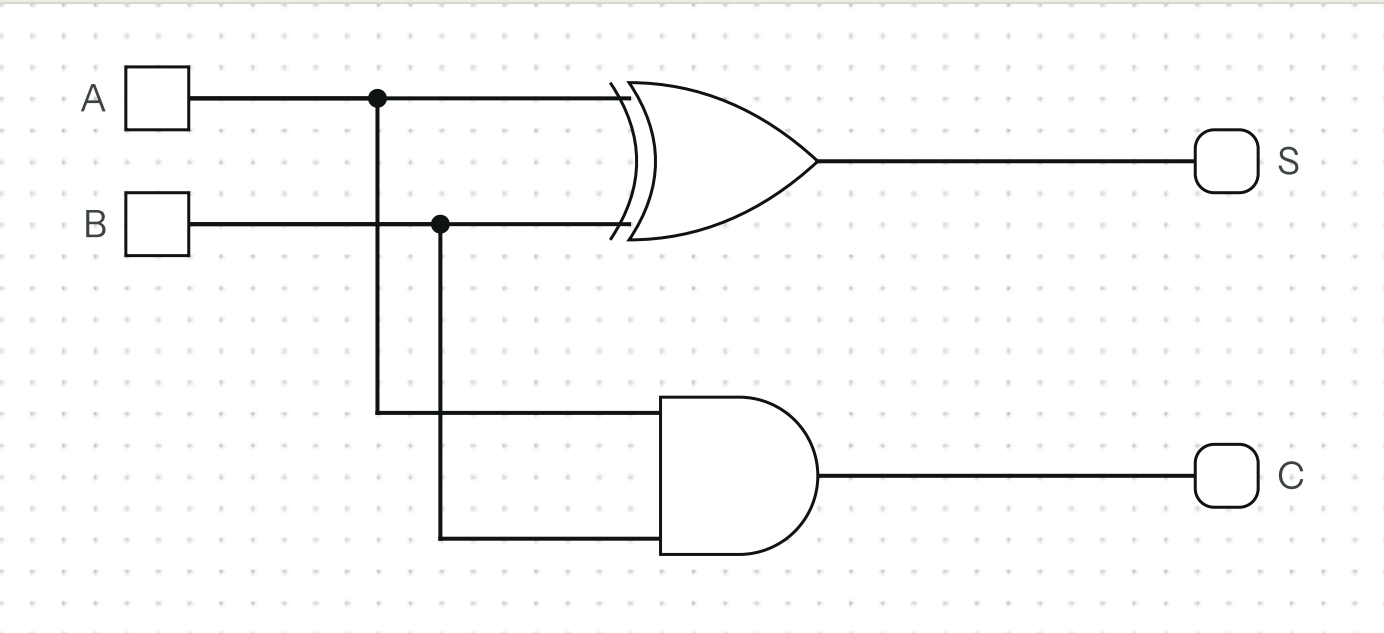


Windows Server 2025 desktop on a GitHub runner, 5.0.0 from the MSI.
`CI-W25-LE500-NATIVE-V5-BEFORE-CLICK-01 · CI-W25-LE500-NATIVE-V5-AFTER-CLICK-01`

1 **Check:** After one click on the canvas the wires show their colours again.

You can compare your file with ours, which passes a four-row check in 5.0.0:

CIRC-TUTORIAL-HALF-ADDER · MAIN



SAVED WITH ■ 5.0.0 T-PROJ-TUTORIAL ↗ Open in .circ Viewer ↗

Our half adder: A and B into an XOR for the sum and an AND for the carry.

CC BY 4.0 · SHA-256 CC7C23C201C8

DESCRIBE THIS CIRCUIT

Components: Input pin A; Input pin B; XOR Gate 1; AND Gate 1; Output pin S; Output pin C.

- Input pin A connects to XOR Gate 1, AND Gate 1 (net A).
- Input pin B connects to XOR Gate 1, AND Gate 1 (net B).
- XOR Gate 1 connects to Output pin S (net S).
- AND Gate 1 connects to Output pin C (net C).

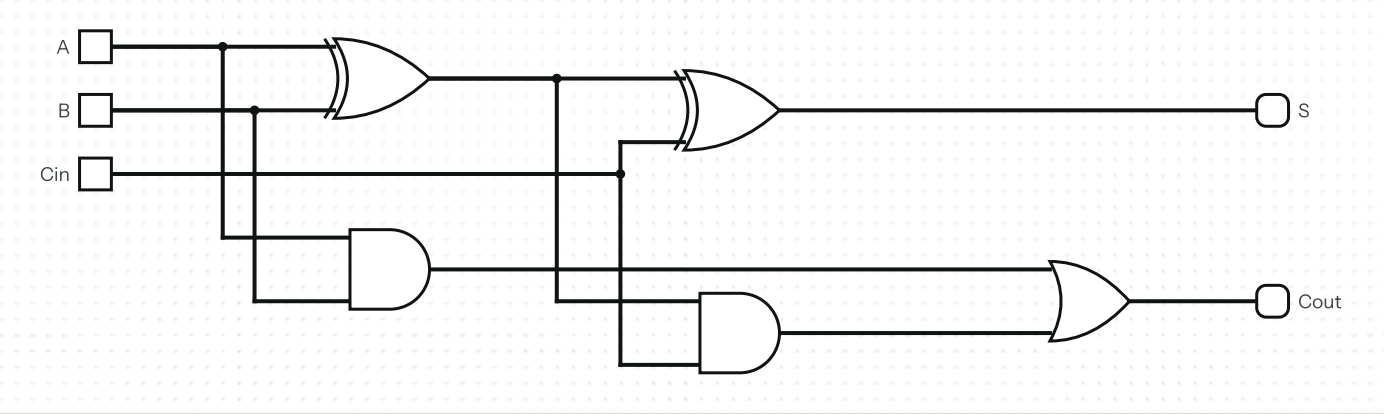
If something looks wrong

WHAT YOU SEE	CAUSE	WHERE TO LOOK
A wire is blue, the output reads U	A wire end is not on a port	Blue wire ↗
A wire is red, an output reads E	Two outputs drive one wire	Red wire ↗
A wire is orange, “Incompatible widths”	Two bit widths meet on one wire	Orange wire ↗
Everything is grey after opening any file, yours included	Values are not shown yet	Click the canvas or press Ctrl+R: grey wires ↗
Clicking an input does nothing	You are in the Edit tool, which selects instead of toggling	Click the hand (Poke) in the toolbar

Make it a full adder

A full adder also takes a carry in. Add a third input `Cin`, a second XOR and AND, and an OR gate: the first XOR gives A⊕B, the second gives the sum S, the two ANDs make the partial carries and the OR combines them into Cout.

CIRC-TUTORIAL-FULL-ADDER · MAIN



SAVED WITH ■ 5.0.0 T-PROJ-ADDER ↗ Open in .circ Viewer ↗

The full adder: two XOR, two AND and one OR gate. Eight rows check it.

CC BY 4.0 · SHA-256 137B2CF35D4E

DESCRIBE THIS CIRCUIT

Components: Input pin A; Input pin B; Input pin Cin; XOR Gate 1; AND Gate 1; XOR Gate 2; AND Gate 2; OR Gate 1; Output pin S; Output pin Cout.

- Input pin A connects to XOR Gate 1, AND Gate 1 (net A).
- Input pin B connects to XOR Gate 1, AND Gate 1 (net B).
- Input pin Cin connects to XOR Gate 2, AND Gate 2 (net Cin).
- XOR Gate 1 connects to XOR Gate 2, AND Gate 2.
- AND Gate 1 connects to OR Gate 1.
- XOR Gate 2 connects to Output pin S (net S).
- AND Gate 2 connects to OR Gate 1.
- OR Gate 1 connects to Output pin Cout (net Cout).

Download circ-tutorial-full-adder.circ ↗ and its 8-row test vector ↗. To run the vector, see Test vectors ↗. For the next step, build a 4-bit adder ↗.

Classic 2.7.1 differences you will notice

FEATURE	○ 2.7.1	■ 5.0.0
Components you can place	◧ PARTIAL 57 in 7 libraries	✓ YES 178 in 14 libraries (F23)
Opens a file from the other version	✕ NO cannot open 5.0.0 files (F04)	◧ PARTIAL opens 2.7.1 files with a compatibility warning (F02)
Menu names for the clock	⬚ UNTESTED Ticks Enabled, Tick Once	✓ YES Auto-Tick Enabled, Manual Tick (F46)
Wire colours after opening a file	⬚ UNTESTED	◧ PARTIAL grey until a click or Ctrl+R (F45)
Input pin Behavior attribute	⬚ UNTESTED	⬚ UNTESTED added in 4.0 according to upstream, not tested by us
Toolbar icons	⬚ UNTESTED	✓ YES as in the shots above

Next lessons

LEARN

Interface and shortcuts

Every part of the window and the keys you will use.

→ ↗

LEARN

Clock and simulation

Make a clock run and step through it.

→ ↗

LEARN

Subcircuits

Reuse a circuit as a part.

→
↗

LEARN

Combinational Analysis

Truth tables and expressions from a circuit.

→
↗

LEARN

Timing diagram

Watch signals change over time.

→
↗

LEARN

Test vectors

Check a circuit against a table of rows.

→
↗

FIX

Wire colours

What red, blue, orange and grey mean.

→
↗

FIX

Oscillation apparent

Fix a loop that never settles.

→
↗

BUILD

Adders

From one bit to four bits.

→
↗

BUILD

ALU

Combine adders and logic.

→
↗

BUILD

8-bit CPU

A processor from parts.

→
↗

Download this tutorial as a PDF

A print version of this page, made from the page itself: A4 PDF ↗ or US Letter PDF ↗. Logisim-evolution also ships its own tutorial under Help, Tutorial; we do not rehost it.

To see which version made a file and what opens where, read the .circ file page ↗.

Frequently asked questions

Is there a Logisim tutorial PDF?

Yes. This page has a printable version, generated from the page itself. Logisim-evolution also has its own tutorial under Help, Tutorial.

How do I change an input from 0 to 1?

Select the Poke tool (the hand in the toolbar) and click the input pin. In our test one click on A changed it from 0 to 1, and the outputs updated at once.

Why is my wire blue?

Nothing drives it. The most common cause is a wire end that stops just short of a port. See the wire colours page for the cause and the fix.

How do I rotate a gate?

Select the gate and change its Facing in the Properties panel on the left.

Does this tutorial work for Logisim 2.7.1?

Mostly. The steps are the same except where the version differences below say otherwise. A file saved by 5.0.0 does not open in 2.7.1.

Where is the Logisim-evolution documentation?

In the program, under Help: Tutorial, User's Guide and Library Reference.

Why are my wires grey after I open my file?

5.0.0 shows values only after one click on the canvas or a Reset Simulation (Ctrl+R). Moving the mouse does not help. We saw it on Linux, Windows and macOS.

SOURCES

1. Half adder walkthrough in the Logisim-evolution 5.0.0 window, logisim.dev lab data: `test-data/lab8-2026-11-11-tutorial/ (screenshots), test-data/lab3-2026-11-09-gui/ (menus)`

2. Verified circuits for this page, logisim.dev lab data: `src/content/circuits/circ-tutorial-half-adder*, circ-tutorial-full-adder*`

RELATED

LEARN AND FIX

Interface, shortcuts and preferences

The full interface tour

→ ↗

LEARN AND FIX

Clock, ticks and simulation controls

Clock and simulation controls

→ ↗

LEARN AND FIX

Subcircuits and custom appearance

Subcircuits

→ ↗

LEARN AND FIX

Combinational Analysis: truth table to circuit

The Analyze Circuit window

→ ↗

LEARN AND FIX

Wire colours and values

What wire colours mean

→

↗