

Logisim: A graphical system for logic circuit design and simulation

October 21, 2001

Abstract

Logisim facilitates the practice of designing logic circuits in introductory courses addressing computer architecture. The program's design emphasizes simplicity of use, with a secondary goal of enabling design of sophisticated circuits. This motivates a two-tiered system, where users can move to the second tier by selecting a menu option.

Users draw circuits of logic gates using a toolbox model popular in drawing programs. The circuit automatically propagates circuit values through the circuit; by selecting the appropriate tool, users can toggle switches to see how the circuit behaves in other situations.

In the advanced tier, the user can treat circuits as black boxes within larger circuits, enabling the simulation of hierarchical designs. The author has successfully drawn a simple eight-bit CPU in this system.

The program has proven useful in a variety of introductory courses, from a non-majors survey course to a sophomore-level Systems course. Students find Logisim simple to follow, and they find the laboratories designed around it useful in reinforcing the circuit concepts from class.

1 Motivation

Our department offers three classes in which logic circuits play a major role. Circuits are one of the first units in both of our survey courses (a non-major course and a first-semester majors course). And circuits make up a large fraction of a sophomore-level Systems course. All of these courses have laboratory sections, where students have structured assignments during scheduled meetings in computer laboratories. Because of this structure, and because of the significant role of circuits in the classes, it is important that students practice logic circuits in the laboratory.

Though an important part of the course, logic circuits are not so prominent in the course that we can afford for students to build circuits on breadboards (too steep a learning curve) or to purchase commercial circuit simulation software (too expensive for students). Thus, we need a free piece of software to support this portion of the laboratories. However, we could not locate a solid piece of software for our courses — and certainly not one for both Unix and Windows (two of the classes meet in a Windows laboratory, and one meets in a Unix laboratory).

Thus the author began his own, named Logisim. The emphasis of this project is to develop an easy-to-learn program for developing and simulating circuits of logic gates. The greatest emphasis of the project is on ease of learning, so that students in our introductory courses, who spend only two laboratory sessions with the program, meet as few problems learning the software as possible. For these students, sophisticated ability is not important.

But the Systems students do build relatively sophisticated circuits, so a secondary emphasis is on the power of the circuits built with the program. They need the capability to build more circuits illustrating the internals of a simple CPU. This requires the ability to “black-box” smaller circuits.

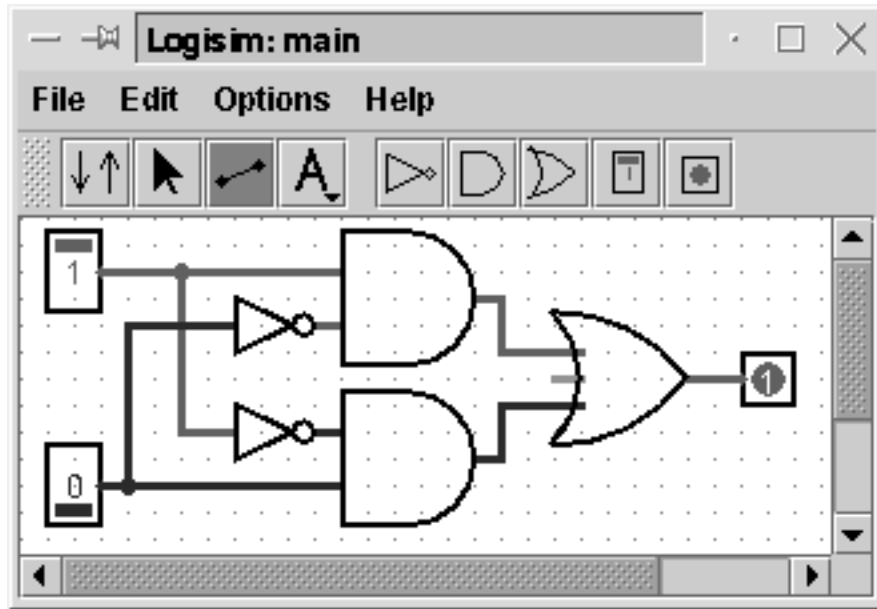


Figure 1: A simple circuit built in Logisim.

Logisim is written as a Java application, so that it can run on both Windows and Unix workstations (two of the above courses meet in Windows labs and another in a Unix laboratory). It fits within a larger suite of Java programs developed locally for other modules of our course (emphasizing binary numeric representation, assembly programming, and finite automata).

2 Beginning features

Figure 1 contains a screen shot illustrating what beginning Logisim users see. Logisim uses the toolbox concept common in drawing programs. Starting from the left of the toolbar, the tools available to the beginner are the simulation tool, the selection tool, the wiring tool, and the label tool. Continuing leftwards, there are tools for adding NOT, AND, and OR gates, as well as tools for adding input switches and output LEDs.

A student building a circuit typically first lays down the components of the circuit. The student selects the tool for the gate to add. As the mouse moves across the canvas area, a ghost appears where the gate would be placed when the student clicks the mouse. All gates are snapped to a strict grid.

The second step is adding wires. The simplicity of Logisim's wiring tool may be its strongest feature. Students can drag anywhere on the canvas to place a wire. All wires must be horizontal or vertical, and they must fit exactly onto the grid. The program automatically detects and draws the connections as each wire is added: Anything the new wire's endpoints hits are connected, as are any existing wire's endpoints crossed by the new wire. (It may sound confusing, but the behavior is very intuitive.) In attaching to AND and OR gates, all five grid points on the left side are open for attachments. (For circuits with many gates, Logisim includes a "Small Gates" option. When selected, the gate tools add smaller gates allowing only three connections.)

The final step is the simulation. Even while the student adds wires, they can see the values flowing through the wires. (A bright green indicates a true value, while a dark green indicates a false value. Wires that are not yet connected to a known source are a pale blue.) Once the student selects the simulation tool, a

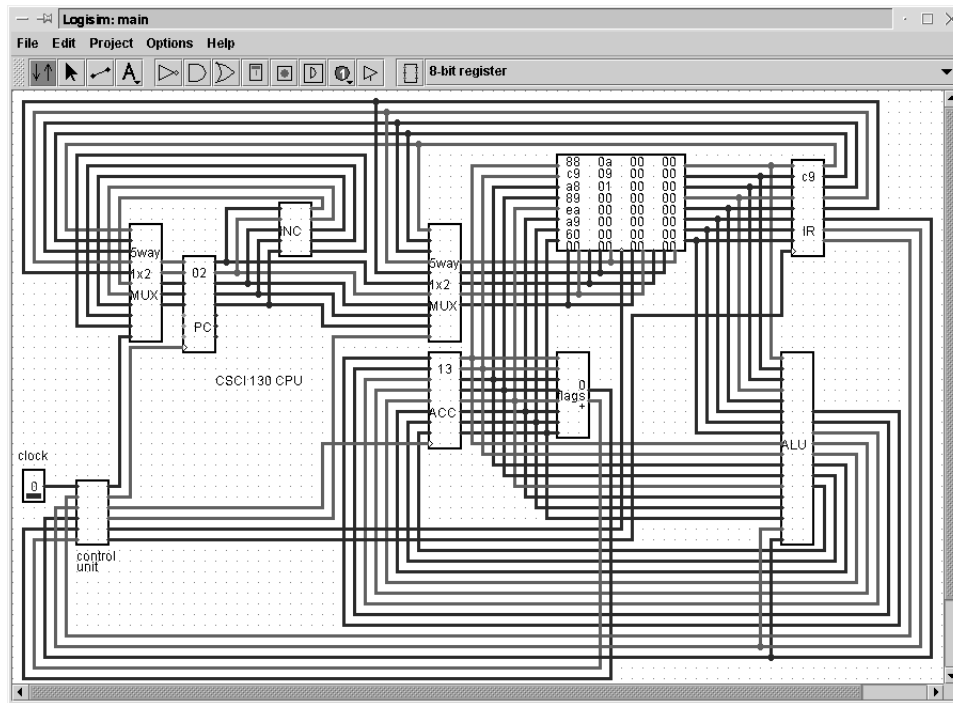


Figure 2: A more complex circuit in Logisim.

click on any of the switches will toggle it, with immediate propagation through the circuit.

When the student adds something mistakenly, a right-click (or command-click) on the circuit component will bring a pop-up menu, including a Delete option. Also, the selection tool permits selecting a region of the circuit, after which Cut from the Edit menu will remove all the components and wires lying within the rectangle.

The File menu gives students the ability to save their circuits for future editing and to print their circuits for attachment to their laboratory reports.

3 Advanced features

For non-majors, who represent about half of the users of Logisim at our colleges, the above is all of Logisim that they see. But other students, when they want to develop sequential and hierarchical circuits, will select the Advanced Tools option in the Options menu. This enables Logisim's more complex tools.

Figure 2 contains a screen shot illustrating a more complex circuit built using Logisim. In fact, this screen shot is of an implementation of the simple three-register, eight-bit CPU with 32 bytes of memory that our non-majors learn to program. It illustrates the practical limits of the current version of Logisim, but it goes well beyond what our students do in their classwork.

The Advanced Tools option adds tools for a few more circuit components: D flip-flops, constant-value sources, and tri-state gates. But the most notable new feature that comes with Advanced Tools is the subcircuit tool and the Project menu, which enable hierarchical circuits.

Logisim views a circuit as a set of "subcircuits," each with a user-defined name. The Project menu lists all subcircuits in the currently open circuit. Selecting a subcircuit's menu item brings up that subcircuit for editing. The default subcircuit (which indeed is all the beginning user ever sees) is titled "main". But users

can add new subcircuits via Add New Subcircuit under the Project menu. This enables them to design useful pieces of a larger circuit, such as the multiplexers or the AL used in Figure 2.

To use a subcircuit as a black box within another subcircuit, the user would select the subcircuit tool on the far right side of the toolbar. The user selects the subcircuit to add from the list box in the toolbar (listing all subcircuits within the current project).

A subcircuit component is drawn as a rectangle with pins on the left and right sides. The pins on the left side correspond to the subcircuit's switches (in the same top-down order), while the pins on the right side correspond to the subcircuit's LEDs (also in the subcircuit's top-down order). This simple inferred structure from the subcircuit simplifies the circuit development process, though at the expense of less flexibility in how the subcircuit appears. It would be nice, for example, for a XOR gate subcircuit to appear as a XOR gate when it is used. But the added complexity this would entail to both the user and to the program conflicts with the design goals of Logisim.

When the circuit propagates values into a subcircuit component, it propagates those values through the subcircuit, and the values that that subcircuit's LEDs take on are the values propagated from the subcircuit component onward. (The subcircuit maintains values from previous propagations, so that a sequential subcircuit work well, even if it is used multiple times within the same circuit.)

4 Limitations

Logisim is designed for illustrating logic circuits in an introductory course. Using it in a full-semester architecture course is certainly possible, but the course would stretch the limits of the program, and some assignments could not be completed using Logisim. For example, Logisim provides minimal or no support for buses, circuit timing, and circuit layout.

One of the biggest limitations in building more realistic circuits with Logisim is the fact that it does not permit the bundling of wires. This makes the prospect of developing a circuit for dealing with wide units of data impractical. Thus, the author could certainly not recommend using Logisim to illustrate circuits working with even 16-bit data. The eight-bit CPU illustrated in Figure 2 is nearly as complex a circuit as Logisim can conveniently handle.

Also, the program's construction pays little attention to efficiency details. Some sluggishness can appear. For example, in running the CPU of Figure 2 on a recently purchased Sun workstation, Logisim delayed a full second after each toggle of the clock switch. For circuits of a moderate size, however, the delay is hardly perceptible. We have not noticed significant problems with efficiency in our classes.

Other limitations of Logisim derive from its use of the Java Virtual Machine (version 1.3). While this enables Logisim to easily work on both Windows and Unix platforms, it complicates the installation process. Locally, we have set it up simple scripts to make this easier in our laboratories.

5 Experience

Logisim was first developed for our introductory majors course in October 2000, though it was also clearly suitable for the non-majors course. Thus it began as a simple project, incorporating little more than the beginner's features. However, the professor of our Systems course recognized it as helpful for his course also, even without the more advanced features. But this usage motivated the addition of hierarchical circuits and other features. By the end of the first semester (Spring 2001), Logisim doubled in size, going from version 0.1 to version 0.5.

By the Fall 2001 semester, Logisim reached version 1.0, sporting project-management features to enable building and debugging larger circuits. Also added were the selection tool, a more complete help system,

and many minor enhancements. Logisim again had doubled in size, going to version 1.0. Bugs discovered during its use in the Fall 2001 semester has pushed Logisim to version 1.04.

Logisim has proven a boon for teaching logic circuits, allowing students to quickly get up to speed on the program and to concentrate on practicing how to build a circuit in an immediate-feedback environment. In the survey courses, students have found the program simple to use and helpful in illustrating the logic circuits taught during lecture. Prior to Logisim, they used a student project from many years ago that was more difficult to use, very unstable, and for which the source code had been lost. Logisim is proving a vast improvement. Additionally, its flip-flop capability has enabled the addition of a new laboratory assignment for small sequential circuits that we could not have students complete before (to simulate internal workings of a simple vending machine).

In our sophomore-level Systems course, students were working with breadboards prior to Logisim. But these were too difficult to use to expect students to build serious circuits illustrating many concepts taught in class. Now, students complete a single breadboard lab to get the idea of connecting electronics. But then they move to Logisim, with which they build and experiment with a small CPU.

Logisim is now a stable program in heavy use at our institutions; about 150 of our students study using the program each semester. Its simple sophistication has enabled students to practice more concepts from class. Logisim has proven a strong addition to a variety of classes.