

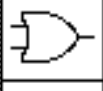
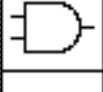


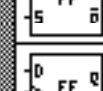
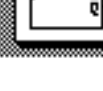


DigSim is a program for designing and experimenting with digital circuits. It is useful for both learning digital logic and for testing advanced circuit ideas.

When DigSim starts up, a palette of circuit elements and an untitled circuit window appear. You can drag any of the elements over to a circuit window to add them to the circuit.

The elements presently available are:

Input Switch		Constant Zero
Output Light		Constant One
Or Gate		Nor Gate
And Gate		Nand Gate
Xor Gate		Not Xor Gate
Inverter		3-Input Nor Gate
R-S Flip-Flop		3-Input Nand Gate
D Flip-Flop		J-K Flip-Flop

Any good digital logic book will explain the functions of these elements, but most of them are easy to figure out with some experimenting.

There are two tools for these elements: a soldering iron for connecting them and a pair of wire cutters for cutting connections. The Tools menu allows you to choose between them. When the soldering iron is selected, clicking on a lead out of an element and dragging to another lead will connect them with a wire as long as this does not connect two output leads together. Clicking on a wire where it comes out of the element (clicking in the middle of the wire will not work) will cut it out of the circuit.

Input switches and output lights are used to control the circuit after it is built. Clicking on an input switch will toggle its state from 0 to 1 or 1 to 0, and this value will be fed to the gates connected to it. Output lights will light up if they are connected to a 1 and stay dark if connected to a 0.

After you have built a circuit and saved it, its name will appear under the Sub-Circuits menu. You can now use that circuit as a sub-circuit inside any other circuit so that you can have much more complicated building blocks than are in the palette. When you have a circuit on the screen, select the sub-circuit you want and a box labeled with the sub-circuit name will appear in the middle of the window. This box will have one input lead for every input switch the original circuit had and an output lead for every output light. It can now be hooked up just like any circuit element. Double clicking on the box will open a circuit window showing what the box represents.

Changing the inputs and seeing what results is sometimes not enough to easily understand a circuit, especially if it contains a lot of memory elements like flip-flops. Running a simulation allows you to see how inputs and outputs are related in time. When you choose Run Simulation... under the Simulate menu, a dialog box fills the screen which has an entry for each input switch and output light in the front circuit window. To run the simulation, you draw in the waveform boxes for the inputs by clicking and dragging. Use the * to see more or less of the wave at a time. A vertical bar marks the end of wave, you can drag this back and forth to make the wave shorter or longer. The simulation is always 1024 time units long (a "time unit" is the time it takes for a signal to go from the input to the output of a single gate). If you only draw a wave 10 units long, it will either remain where you left it for the rest of the simulation or it will start again at the beginning if you click the repeat box for that wave. Once you have set up the inputs simply click the simulate box and it will calculate the outputs and then display them.

A few notes for advanced users: A "time unit" can be assumed to be about 10 nanoseconds for TTL gates. With a 512K Macintosh it is possible to simulate a circuit of 10,000 gates with a heavy use of sub-circuits. This allows students to simulate and explore the internals of a simple computer.

I welcome any suggestions for improvement for DigSim. If enough good suggestions come through, I will create a 3.0 version. If anyone is interested I will also create an analog version. Program ideas are also welcome. Remember this is shareware! I don't want to put myself through college by working at McDonalds!