Objectives

- Digital logic circuits using a simple logic gate simulator.

References

Software needed:

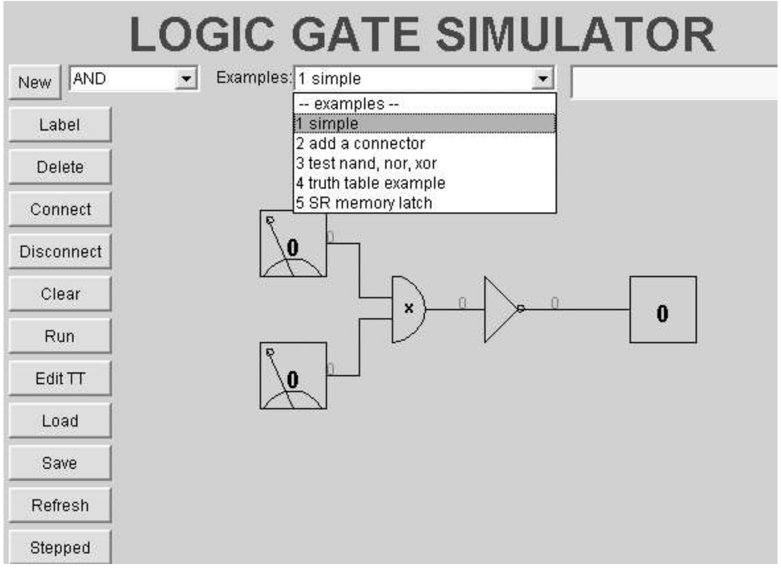
1) A web browser (Internet Explorer or Netscape)
2) Applet from the lab website:
   a) LogicGates applet

Activity

Digital logic circuits are the bedrock of computer design. CPU chips such as the Intel Pentium or AMD Athlon are silicon wafers in which millions of circuits are embedded. Everything your computer does depends upon them.

In this lab, we will learn how to use the "LogicGates" applet to draw and testdigital logic circuits. Chip manufacturers, such as Intel and AMD, use sophisticated (and extremely expensive) software to help them design and test their chips. But the basic idea is the same.

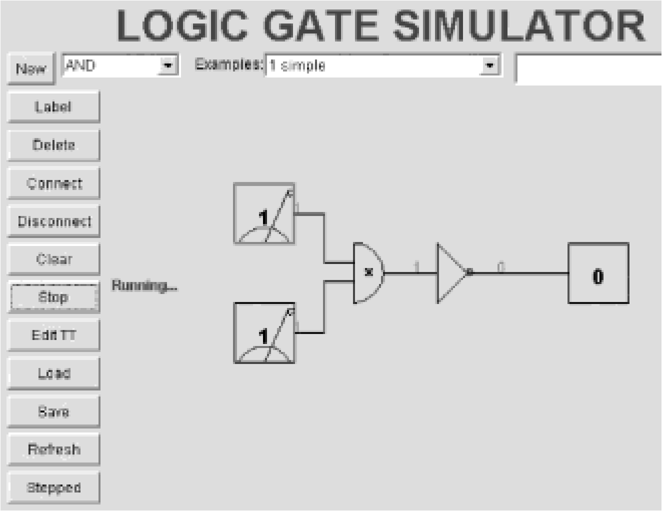
First, start the "LogicGates" applet. You can load a number of examples by pulling down the example choice pull-down menu. Load Example 1 "simple," as shown here:



Each logic gate has a distinctive shape. The wire coming out of the gate shows its current value. Gates are oriented so that their inputs come in from the left side and their outputs go out from the right side. This cannot be changed nor can the gates be rotated in this applet.

Two special boxes exist for input and output. The switch box has a lever in it along with 1 or 0. To flip the lever and change the value, click on the number right in the center. The other box has only 1 or 0 in it. This is the output box that accepts a wire from a gate and displays its value.

To propagate new values through the circuit, click on the *Run* button. It changes to say *Stop* so that when you click on it again, the simulator stops running. When the circuit is running, you can click on the switches and watch values change throughout the circuit.



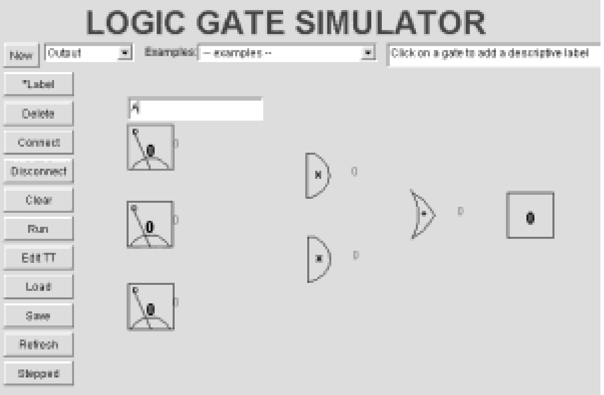
To practice using the simulator, we'll build the logic circuit. First, if your simple example circuit is still running, press the Stop button. Now, clear the screen by clicking on the *Clear* button. To build the circuit, you'll need three switches, two AND gates, one OR gate, and one output box. To create them, click on the *New* button, pull down the gate menu and select the gate or other item you want. Then click once where you want the switch, gate, or output box to sit. You can reposition gates, switches, and outputs by dragging them to a new location.

Notice that when you are adding gates, the *New* button changes to say **New*. Most of the buttons in this applet work like this: they show they are still active by displaying an asterisk in front of the label on the button. Because the button stays active, you can click on different gates without having to re-click the *New* button over and over.

Click on the *New* button again to stop adding components. The asterisk will go away.

Let's also attach labels to the boxes to mimic the circuit diagram. However, in the Logic Gate simulator, labels attach to boxes.

First click on the *Label* button, which allows us to label gates and switches. Then click on the top switch. A textfield opens up right above the switch. Type "A" into it and press Return.



Output

Examples: -- examples --

Click on a gate to add a descriptive label

A

Here’s what you should see:

Label the middle switch as B and the bottom switch as C. Also label the output X. If you want to be complete, label the top AND gate as D and the bottom AND gate as E. When done labeling, click on the *Label* button so that the asterisk doesn't appear.

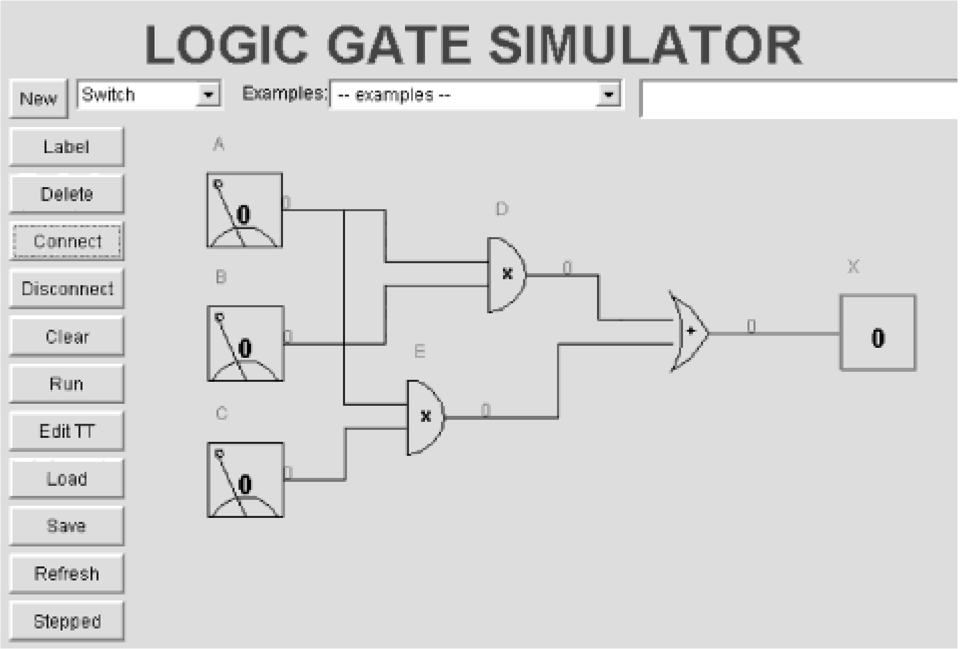
Now it's time to connect the gates, so click on the *Connect* button. Click once on switch "A." A red line flashes into view, anchored in the center of the switch and following the mouse pointer around the screen. Click on the top AND gate. Repeat this process and connect all the boxes so they match the screenshot below. If you make a mistake, click on the *Disconnect* button, then click on a gate. All lines between that gate and all other components will be removed.

A gate's output may have any number of wires coming out of it so that it connects to more than one other gate. Merely select the same gate as the left end of a wire more than once.

Automatic elbows break lines up into horizontal and vertical segments. To see a version that uses diagonal lines, click on the *Stepped* button. However, in most circuit fabrication technologies, all wires must lie on a grid so diagonal wires are either impossible or expensive. To see the elbowed lines again, click on the bottom button, which has changed from *Stepped* to *Diagonal*.

Sometimes the elbows make it hard to see overlapped wires, and that plagues this circuit. So move the bottom AND gate a little to the left until the wires show more clearly.

Here's the final picture:



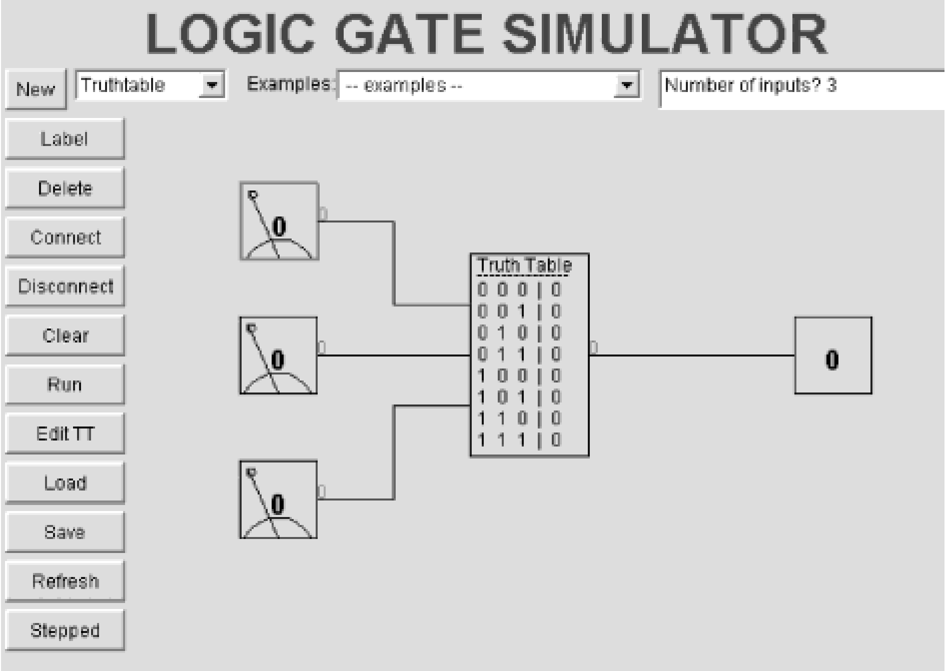
Now run the circuit by clicking the *Run* button. Click on the switches to change their values. Record the results in a truth table and see if the circuit gives you the same results. They had better agree! (They will, don't worry.)

Imagine what a terrible thing it would be if a major chip manufacturer's simulation software didn't give the right values and they sent a chip into production with flaws in it. Well, actually, don't bother imagining: It's already happened! In the early 1990s, Intel recalled their new Pentium processor when scientists and engineers pointed out that in certain calculations the chip gave the wrong answer! Analysts have estimated that this bug in the chip's division algorithm cost the company about $500 million. (Too bad they weren't using our simulator to check their circuit design!)

Our simulator provides you with six types of gates: AND, OR, NOT, NAND, NOR, and XOR. Another type of "gate" available in our circuit simulation is the *Truth table*, which allows you to avoid building really complicated circuits with lots of gates. Instead, you fill in the truth table values and the simulator treats it as a gate. A truth table "gate" can have two, three, or more inputs.

Let's build a circuit using truth tables. Clear the screen and add three new switches and one output.

The trickiest part is putting the right size truth table into the circuit. To do this, make sure the *New* button is activated (has an asterisk next to **New*.) Then pull down on the gates menu and bring *Truthtable* into view. In the yellow message box, Logic Gates will ask you how many inputs your truth table should have. The default is 2, but since we have 3 switches, replace the 2 with 3. Then click once in the middle of the area to create the truth table. Click off the New button and connect these gates as shown below:



Number of inputs? 3

Truth Table

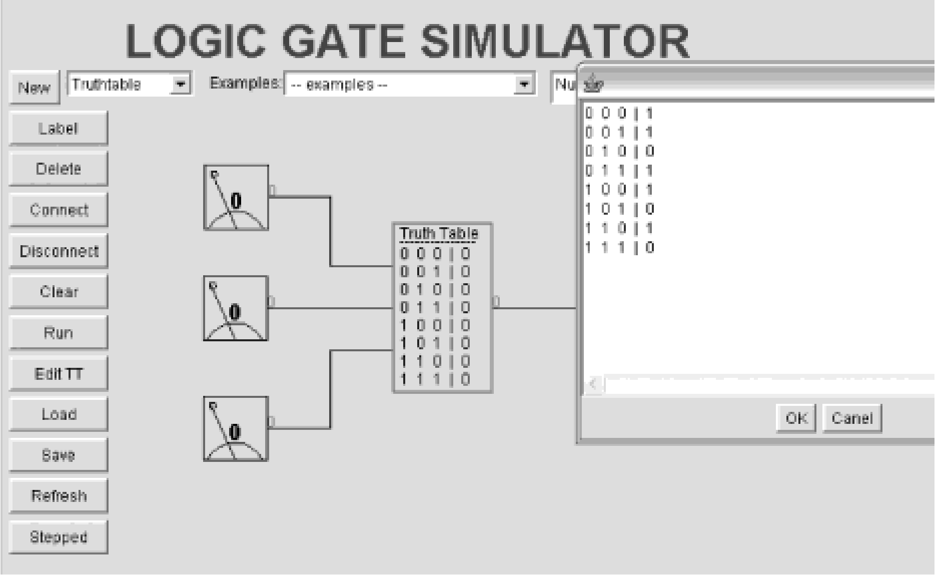
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Important note! Make sure that when you connect the switches, you connect the top one first, then the middle one, and finally the bottom one. This is necessary because the order of the incoming wires determines which value is associated with which column in the truth table.

The first wire connected will correspond to the first column in the table, the second wire will correspond to the second column, and so on. If you connect them in a different order, the wires going into the truth table will not correspond to the columns of your truth table.

New truth tables list all the input values in canonical order, but since the output is shown as always 0, we need to edit the truth table so it implements our desired output. To edit a truth table, click once on it (after making sure that all the buttons on the left are off, i.e., do not have asterisks in front of their names). This makes the truth table the top gate and its border will turn red. Now click on *EditTT*. A green window pops up, allowing you to edit this truth table.

The input values are separated from the output value by a vertical bar symbol (|). Go through the truth table and change some of the output values, as shown in the picture below.



```
Truth Table
0 0 0 | 0
0 0 1 | 0
0 1 0 | 0
0 1 1 | 0
1 0 0 | 0
1 0 1 | 0
1 1 0 | 0
1 1 1 | 0
```

```
0 0 0 | 1
0 0 1 | 1
0 1 0 | 0
0 1 1 | 1
1 0 0 | 1
1 0 1 | 0
1 1 0 | 1
1 1 1 | 0
```

Once you are done, click OK to dismiss the truth table editing window, and then click on *Refresh* or just drag the truth table around so that the simulator will repaint the screen.

Run the simulator and play with the switches, comparing the value in the output box to what you see in the truth table. Do they match?

Since the truth table "gate" bypasses digital logic gates altogether and implements the given truth table, it might seem like using it is cheating! Actually, the truth table is similar to SSI chips. Sometimes we are more interested in the function that the circuit computes (the pairing up of inputs and outputs) than in exactly which gates are needed to implement that function.

By the way, if there are only two inputs to the truth table, there will be two columns and four rows. If there are three inputs, there will be three columns and eight rows, as shown in the example. Four inputs would require four columns and 16 rows, and five inputs would require five columns and 32 rows. Do you see a pattern?

The input values in all the truth tables you see in this lab are shown in canonical (standard) order. This isn't necessary for the simulator to work properly, but canonical order helps us fallible, distracted humans keep track of our inputs and outputs.

Tip

LogicGates can be used as a standalone Java application. If you use it as an application (not an applet), you can load and save your programs. To run, navigate to the folder containing the LogicGates class files and double click on the **run_application.bat** file.