COS 116
The Computational Universe
Laboratory 7: Digital Logic I

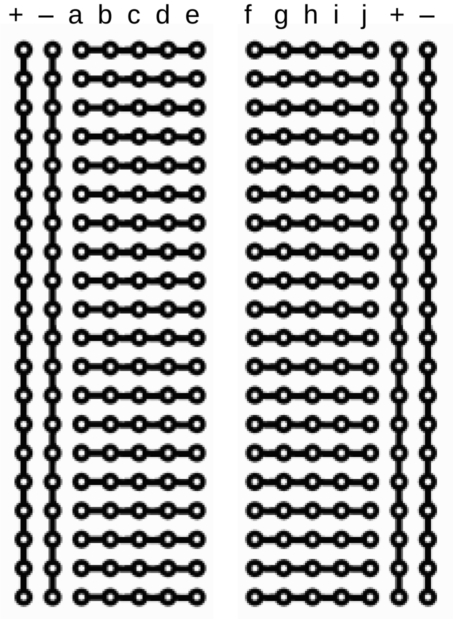
In this lab you'll construct simple combinational circuits in software, using a simulator, and also in hardware, with a "breadboard" and silicon chips. You'll use these circuits to learn about digital logic.

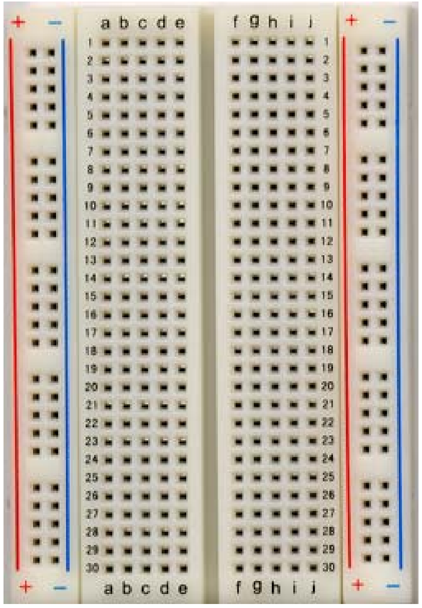
If you get stuck at any point, feel free to discuss the problem with another student or a TA. However, you are not allowed to copy another student's answers.

Hand in your lab report at the beginning of lecture on Tuesday, April 3. Make sure you answer the questions and complete the deliverables printed in bold. (Number them by Part and Step.) Include a picture of the Majority-of-3 circuit that you design in Logisim. Also, when you assemble the Majority-of-3 circuit on the breadboard, show it to your TA. Otherwise you will not get credit for the lab.

Also, at the end of the lab you must remove any wires or components that you added to the breadboard and place them back in the plastic box you received. Return this plastic box to the TA before leaving the lab.

Breadboards are used to build temporary circuits during prototyping. Their name derives from little holes into which you insert electronic components and connecting wires. You can quickly assemble and disassemble circuits. No soldering required! Here is a picture.





Some of these holes are wired together on the inside, as shown in the diagram on the left. For example, if you insert two wires into the holes labeled “a” and “e” in the same row, they will be electrically connected.

The breadboard has plugs at the top to accept power. **Do not connect the power supply until you reach the appropriate time in the experiment. Also, your breadboard will come with some wires and switches inserted into it already; please do not remove them at any time.**

Our silicon chips use a voltage of 5 volts to denote the binary value 1 and a voltage of zero volts to denote 0. These two voltages are also called “power” and “ground” respectively. Notice that the breadboard has two columns on each side marked with the red and blue stripes (and the symbols + and –). All holes in the + columns are connected to the power (red/+) terminal on top and all holes in the – columns are connected to the ground (blue) terminal of your power supply.

Here are the parts that you will need for today. Your lab TA will make them available to you.

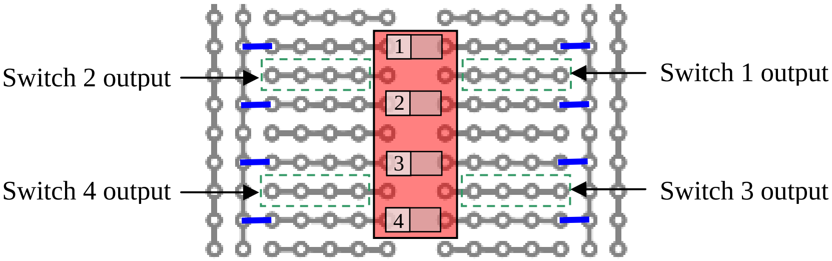
- 1 power supply (share with a neighbor)
- 1 set of power supply connectors
- 1 breadboard (with pre-wired switches)
- 1 set of breadboard jumper wires, multiple colors
- 1 LED
- 1 resistor
- 1 74LS04 chip (6 “Not” gates)
- 1 74LS08 chip (4 “And” gates)
- 1 74LS32 chip (4 “Or” gates)
- 1 chip extractor tool

Connect your breadboard to the power supply using the red and black power supply leads. Connect the red terminal of the power supply to the red post on the breadboard (and the black terminal to the black post). Each power supply can feed multiple breadboards since the jacks can stack on top of each other. Switch the power supply on, and verify that its voltage reading is 5 volts. If not, contact the TA.

Caution: For safety, before you insert or remove any components from the breadboard, always disconnect the power by unplugging the red power supply connector from the breadboard’s connector post.

Part 1: Understanding switches, LEDs, and the breadboard

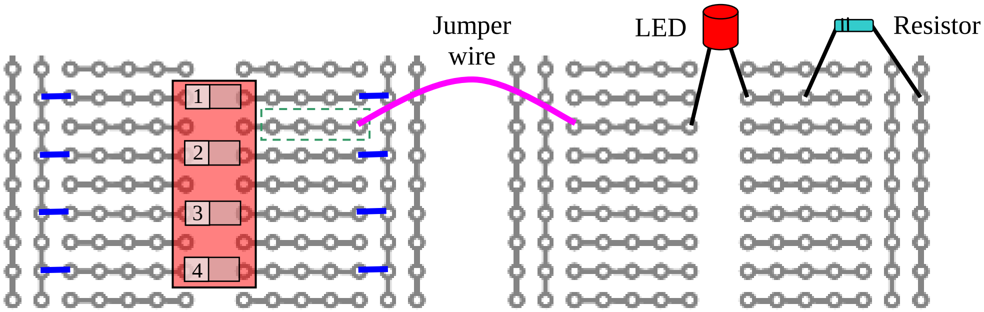
In this part you’ll implement a very simple circuit involving a single switch and a light bulb (LED): the switch is used to turn the LED on and off.



Switch position: Left = “1”, Right = “0”

Your breadboard is divided into two halves, each like the pictures on page 1. On the left half, we’ve wired some switches for you to use in all the experiments. There are four white switches in one red block. For each switch, flipping the switch to the right connects the output to ground, resulting in a binary “0” value; and flipping the switch to the left connects the output to power, resulting in a binary “1”. You can feed the output from one of the switches as an input to another component by using a jumper wire to connect to one of the switch’s output holes, shown below. Your kit has jumper wires of different lengths. You should always route your wires cleanly (laying them flat on the breadboard) as this will help avoid wiring errors.

As your first exercise with the breadboard, you’ll construct the following simple circuit involving a switch, LED, and resistor.



1. **Disconnect the power by unplugging the red power supply wire from the breadboard.**

2. **Insert the LED into the right half of the breadboard. Make sure the longer lead is on the left, as shown above!** The LED will not function if you connect it in reverse. Use a red LED if possible, as it tends to be the brightest.

3. **Use a short jumper wire to connect one of Switch 1’s output holes to a hole in the row where you inserted the LED’s longer leg.**

4. **Connect a resistor between the row where you inserted the LED’s shorter leg and one of the ground holes in the breadboard’s rightmost column. (Unlike the LED, the resistor can be connected in either direction.)** The voltage from

the power supply will damage the LED if you connect it directly to the switch. The resistor reduces the voltage and protects the LED.

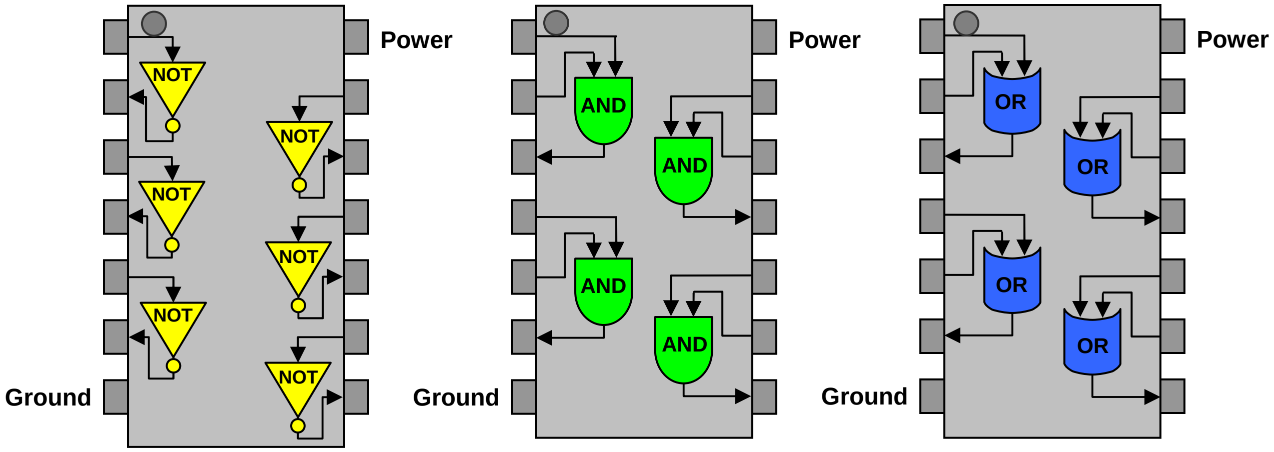
5. **Reconnect the power supply to the breadboard. Toggle Switch 1, and verify that the LED is on when the switch is to the left and off when the switch is to the right.**

6. **Rewire your circuit to use Switch 2 instead of switch 1, and verify that it works. Repeat for Switch 3 and Switch 4.** If any of the switches do not function correctly, ask your TA for assistance.

Part 2: Understanding Logic Gates and Truth Tables

Now we understand the gates contained inside our silicon chips, and build a circuit with a single gate. Each chip used today packages several identical gates:

74LS04 chip **74LS08 chip** **74LS32 chip**



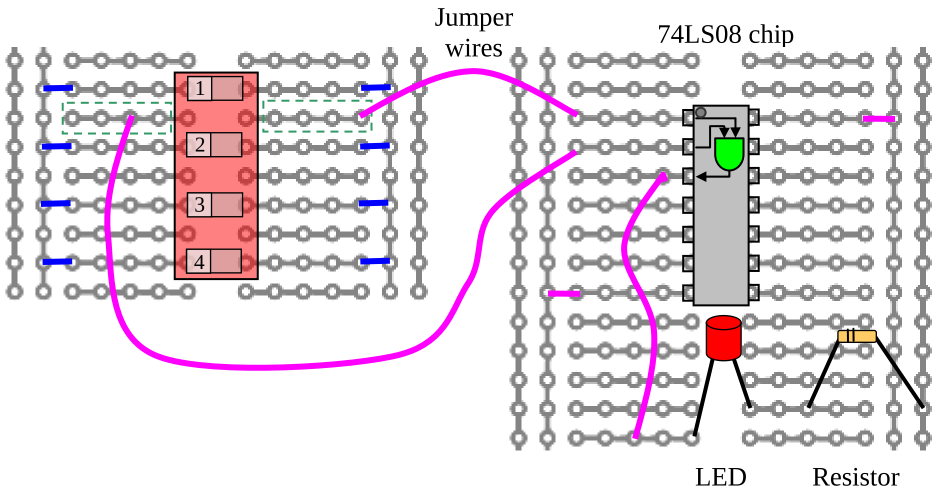
6 NOT gates **4 AND gates** **4 OR gates**

Notice the small imprinted dot at the top of each chip—you'll use this to orient the chip correctly on the breadboard. The upper-right pin on each chip should be connected to power (wire this to the + column on the breadboard), and the lower-left pin should be connected to ground (wire this to the – column). The diagram above shows how the remaining pins are assigned to the inputs and outputs of the gates.

1. **Disconnect the power by unplugging the red power supply wire from the breadboard, and remove all the components (except the switch) which you inserted in the previous experiment.**

2. **Insert the LED, resistor, and 74LS08 (AND gate) chip into the right half of the breadboard, as shown in the picture below. Make sure the notch in the**

chip is on the top, and make sure the longer lead of the LED is on the left. The chip may be damaged if you insert it upside down or wire it incorrectly.



a. **Connect the outputs for switches 1 and 2 to the inputs of the AND gate.**

b. **Connect the output from the AND gate to the LED.**

c. **Connect the chip's power and ground inputs to the + and – columns of the breadboard, respectively.**

3. **Reconnect the power supply to the breadboard.**

4. **Complete the following truth table. Does it match the correct behavior for an AND gate?**

Switch 1	**Switch 2**	**LED**
Right (Off)	Right (Off)	
Right (Off)	Left (On)	
Left (On)	Right (Off)	
Left (On)	Left (On)	

5. **Disconnect the power from the breadboard again.**

6. **Carefully remove the 74LS08 chip using the chip extractor tool. (Ask your TA if you need help.)**

7. **Insert the 74LS32 chip (OR gates) in the same breadboard location. Again, make sure the notch on the chip is on the top.**

8. **Reconnect the power.**

9. **Complete another truth table for the modified configuration. Does your new circuit match the correct behavior for an OR gate?**

10. **Repeat steps 5–9 with the 74LS04 chip (NOT gates). This time, you'll need to rewire the breadboard, since a NOT gate has only one input instead of two like AND and OR gates.** Connect a single switch to the input of the NOT gate, and connect the NOT gate's output to the LED.

11. **Test your NOT gate circuit until it behaves correctly. Draw a picture of the working design in your notes.**

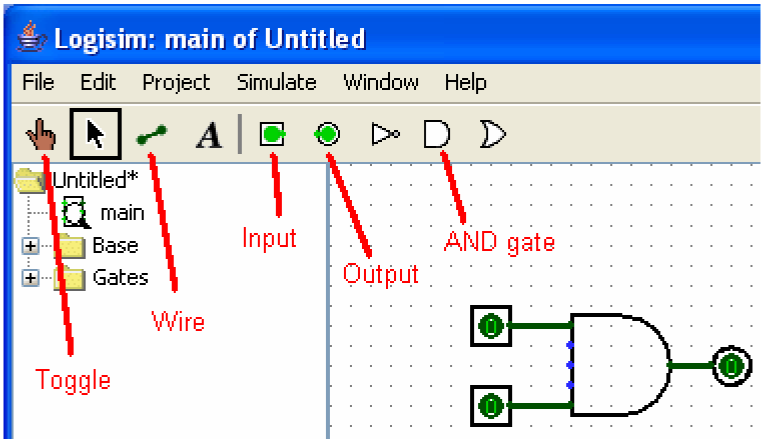
Part 3: Understanding Logisim

Logisim ("logic simulator") is a software program for designing combinational circuits. We will use Logism in this lab and the next lab to help us design complex circuits. The purpose of this section is to help you become comfortable with Logisim. **You do not have to submit your notes for this section as part of your lab report.**

1. **Install Logisim by clicking this link:**

   http://www.cs.princeton.edu/courses/archive/spring07/cos116/lab7_files/logisim.exe

   When prompted, save the file to the Desktop. After the download completes, double-click logisim.exe. (If you are warned that this program may be unsafe, click "Run" and "Ok" as needed.)



2. **Select the AND gate from the toolbar (see figure above), and place the gate anywhere in the dotted field.**

3. **Select the Input tool, and place two inputs to the left of the AND gate.**

4. **Select the Output tool, and place one output to the right of the AND gate.**

5. **Select the Wire tool, and draw a wire connecting the two inputs to the input pins of the AND gate. Also connect the output to the output pin of the AND gate. Your diagram should look similar to the circuit in the figure above.**

6. **Use the Toggle tool to change the values of the inputs while observing the output. Confirm that this circuit behaves like the first circuit you built in Part 2.**

7. **If you like, also construct the other circuits from Part 2 (using the OR gate and the NOT gate).**

Part 4: Boolean Formulas and Truth Tables

As you saw in the lecture, every Boolean function has three equivalent representations: a Boolean formula, a Boolean circuit, and a truth table. One simple example of a Boolean function is the majority function. Given *n* inputs, the majority function is true if more than half the inputs are true; otherwise it is false. Here is the Boolean formula for the three-input version of the majority function:

$$(A \cdot B) + (B \cdot C) + (A \cdot C)$$

Recall that the "·" symbol means AND, and the "+" symbol means OR. The parentheses are used to specify the order of operations, just like in arithmetic.

Confirm that the Boolean formula given above is correct by completing the truth table below. Make sure that the final column matches the behavior of the majority function.

A	B	C	(A · B)	(B · C)	(A · C)	(A · B) + (B · C) + (A · C)
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

Part 5: Building Combinational Circuits in Logisim

1. **Construct the majority-of-three circuit in Logisim. Your circuit should have three inputs and one output. Use the Boolean formula above to help you.** If you are having trouble getting started, first determine how many gates your circuit will need.

   Important: The gates on your chips have only two inputs. However, in Logisim, gates can have several inputs. Since you will implement your circuit design on the breadboard in Part 6, make sure that all the gates in your Logisim circuit use only two of their input pins.

2. **Use the Toggle tool to compare the behavior of your circuit to the truth table you completed in Part 4. Confirm that your circuit correctly computes the majority function.**

3. **Save a picture of your circuit to hand in with your lab report.** There are a few ways to do this. Here is a simple method:

   a. Click anywhere in the window of the Logisim program, and press Alt+PrintScreen. This takes a "snapshot" of the active window.
   b. Open a new Word document, and press Ctrl+V. This copies your snapshot into the document.

4. **Show your circuit to the TA before proceeding to Part 6. This will ensure that any mistakes are caught early, and will save you time.**

Part 6: Building Combinational Circuits on the Breadboard

1. **Disconnect the power by unplugging the red power supply wire from the breadboard, and remove all the components you inserted in previous parts.**

2. **Implement the majority-of-3 circuit on the breadboard. Use the switches and chips you learned about in Part 2. Use your design from Part 5 as a blueprint.**

   Reminder: Every chip you place on the breadboard must be connected to power and ground in order to function properly. Refer to Part 2 for instructions.

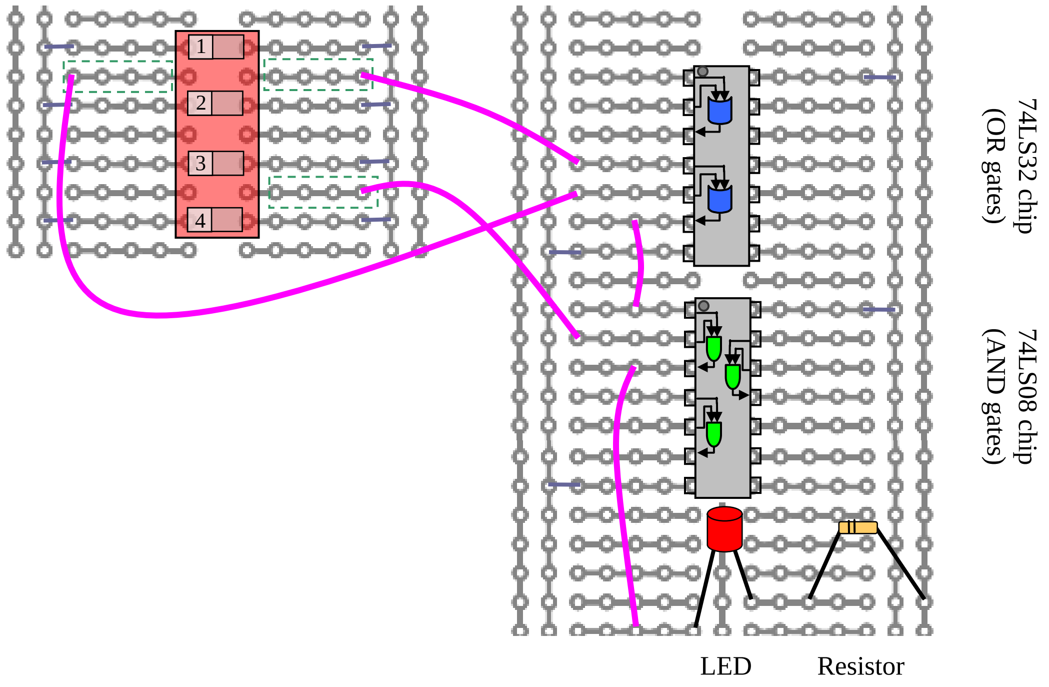
3. **Reconnect the power supply to the breadboard.**

4. **Test your circuit to make sure it matches the truth table for the majority function. If it doesn't, check whether the connections are correct. When your circuit is working properly, show it to your lab TA so that you can get credit for finishing the lab.**

Further Questions to answer in your report

1. **Write the truth table for the majority-of-4 function and represent it with a Boolean expression.**

2. **Determine the Boolean function that is represented by the following breadboard circuit. Denote each of the inputs and the output of the circuit by a variable, and write the Boolean expression of the function being computed.**



3. **Using the chips from this lab, approximately how many gates can fit on a single breadboard? (Ignore the space used for wiring.) Intel's Pentium 4 processor contains approximately 10 million gates. How many breadboards would it take to contain this many gates? Estimate the number of football fields this many breadboards would cover. (State any assumptions you make.)**

4. **You verified the correctness of the majority-of-3 circuit by checking that its behavior matches the truth table. This involved checking $2^3 = 8$ combinations of values by hand. How long would this technique take if you were checking a circuit with 100 inputs? (State any assumptions you make.) How long would it take you if you were as fast as the Pentium 4 processor and could check three billion combinations each second?**