Computer Science 281
Lab 1: Logisim Basics

The purpose of this lab is to introduce you to *Logisim*, a software package that will allow us to build digital logic circuits. The program is a simulator in which we can compose circuits out of logic gates, wires, and simple hardware devices.

Logisim is a platform-independent package, written in Java and distributed as a jar file that allows it to be executed on any system with a Java runtime installed. The current version of the program is 2.1.6 and the application distribution is available from `http://ozark.hendrix.edu/~burch/logisim/` if you wish to install it on your own system. The instructions below assume you are in the Linux lab in Olin 219.

1. Launch Logisim from a command-line prompt in a Terminal window as follows:

   ```
   $ java -jar /usr/local/bin/logisim-2.1.6.jar &
   ```

   The resulting screen has tool buttons at the top whose functions are as follows:

    **(probe)** allows changing of values in the circuit.

    **(select)** selects components to move or edit in the circuit.

    **(wire)** adds a wire to the circuit.

    **(text)** adds text (a label) to either a component in the circuit or on the circuit background.

    **(input pin)** adds an input to the circuit.

    **(output pin)** adds an output to the circuit.

    **(not)** adds a NOT gate to the circuit.

    **(and)** adds an AND gate to the circuit.

    **(or)** adds an OR gate to the circuit.

   Below the tool buttons, the upper left pane contains the aggregate circuit components in use as well as additional components that may be added to the circuit. The lower left pane shows properties (some editable) of the currently selected (or last selected) component in the circuit (if any), and the largest area is the "canvas" in which we add and wire together various components.

2. Start by adding an input pin to the canvas by clicking the input pin tool button and then clicking in the canvas towards the left edge. Next click the text tool button and click on the input pin and type an 'a' to label the input. Finally, click the probe tool and use the cursor on the canvas by clicking the input pin to see it change value.

3. Next add a NOT gate to the right of the input pin, leaving a short gap of 3 or 4 grid lines between the two components. Follow this with an output pin to the right of the NOT gate. Items are added in the same way that you added the input pin. When you add the output pin, include a label 'q'

for the pin. Finally wire together the input pin to the NOT gate input and the output of the NOT gate to the output pin. This is accomplished by using the wire tool and dragging a wire from the input pin to the NOT gate and another from the NOT gate to the output pin. You have just created a digital circuit realization of the boolean equation $q = \overline{a}$. Test the circuit works as you expect by toggling the input pin from 0 to 1 and back again.

4. Now add an AND gate to your circuit (independent of the NOT circuit you just completed). Do this by clicking the and gate tool button and then, *before you place the AND gate on the canvas*, click on the value of the fourth property in the lower left pane, labeled "Number Of Inputs". Change this from a 5 to a 2. Once you have changed the property, you can now bring the cursor over the canvas and place the AND gate. Put it below your NOT circuit, and leave some room to the left of the gate for input pins. Add the AND gate, two input pins labeled a and b to the left of the gate, and a single output pin to the right of the gate, labeled q. Wire them together to complete the circuit realization of the boolean equation $q = a \cdot b$.

5. Give a circuit realization of $q = a + b$ the same way you did for AND.

6. Give circuit realizations and (in your lab notebook) truth tables for the following boolean equations:

   (a) $q = (a \cdot \overline{b}) + (\overline{a} \cdot b)$

   (b) $q = (a \cdot b) + (\overline{a} \cdot \overline{b})$

   (c) $q = \overline{a \cdot b}$

   (d) $q = \overline{a} + \overline{b}$