Basic Logic Gates

Object

To investigate the properties of the various types of logic gates, and construct some useful combinations of these gates. This lab is a modification of the labs available at computersystemsbook.com. The labs found there are "in person" and these are virtual labs that will use the platform Logisim to simulate combinational circuits. Logisim can be downloaded at: https://sourceforge.net/projects/circuit/ As a Java app, it can run on many platforms. Once it is up and running, view the tutorial under the help tab.

Study sections

Computer Systems, Fourth Edition, Jones and Bartlett Publishers: Section 10.1, Boolean Algebra and Logic Gates; Section 10.2, Combinational Analysis.

Procedure

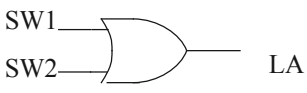
In the following procedure SW stands for switch and is used to set the input level. Also, L stands for light.

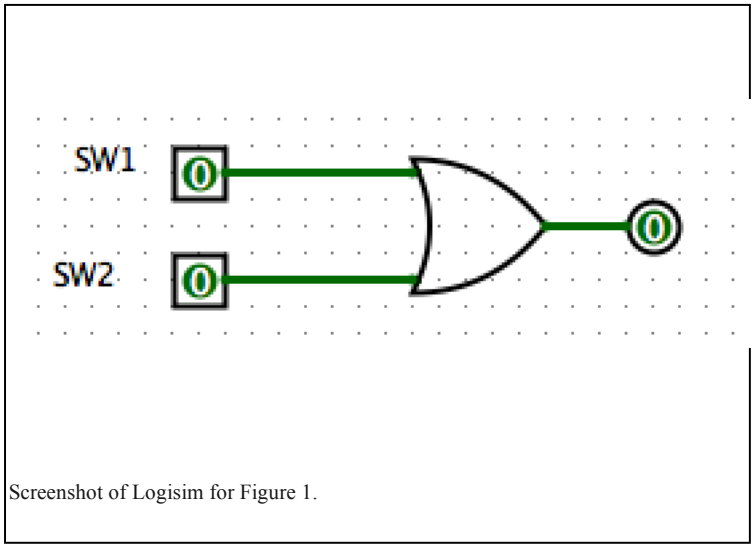
1. OR gate function (A + B)

In Logisim, build the circuit shown in Figure 1. Operate the switches and complete the truth table. Provide a screenshot in the space provided below.

Figure 1

SW1	SW2	LA
0	0	0
0	1	1
1	0	1
1	1	1





Screenshot of Logisim for Figure 1.

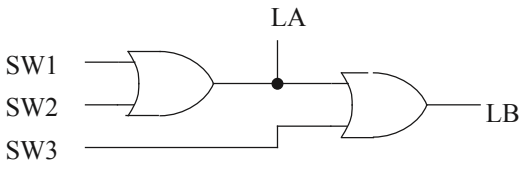
2. Two-level 3-input OR gate

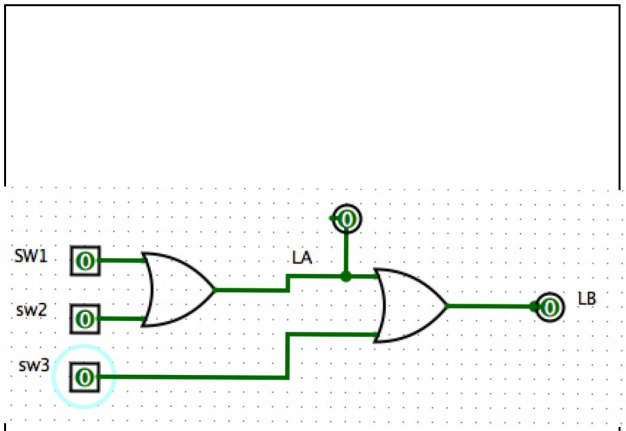
Create the circuit as shown in Figure 2 using Logisim. Set the switches as shown in the truth table of Figure 2 and record the light conditions. Provide a screenshot in the space provided.

Figure 2

SW1	SW2	SW3	LA	LB
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

S





Screenshot of Logisim for Figure 2.

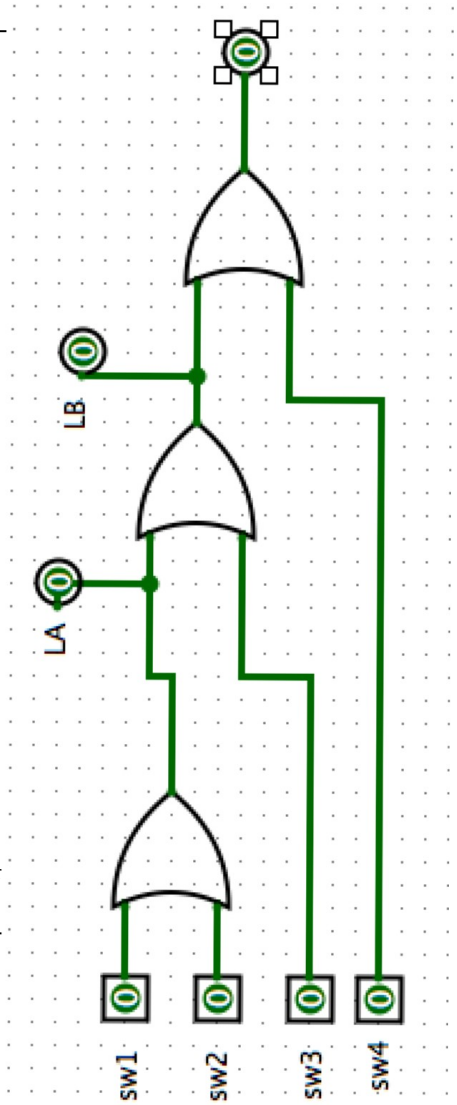
3. Three-level 4-input OR gate

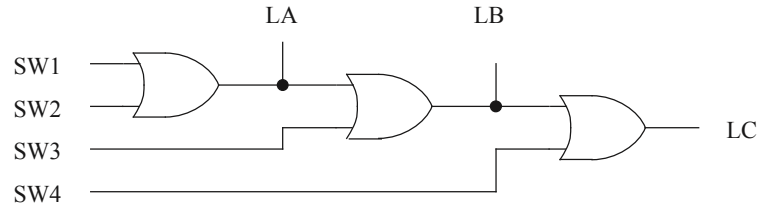
Create the circuit as shown in Figure 3 using Logisim. Set switches as indicated in the truth table of Figure 3 and record the light conditions. Provide a screenshot in the space provided.

Figure 3

SW1	SW2	SW3	SW4	LA	LB	LC
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	0	1	1	0	1	1
0	1	0	0	1	1	1
0	1	0	1	1	1	1
0	1	1	0	1	1	1
0	1	1	1	1	1	1
1	0	0	0	1	1	1
1	0	0	1	1	1	1
1	0	1	0	1	1	1
1	0	1	1	1	1	1
1	1	0	0	1	1	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1

Screenshot of Logisim for

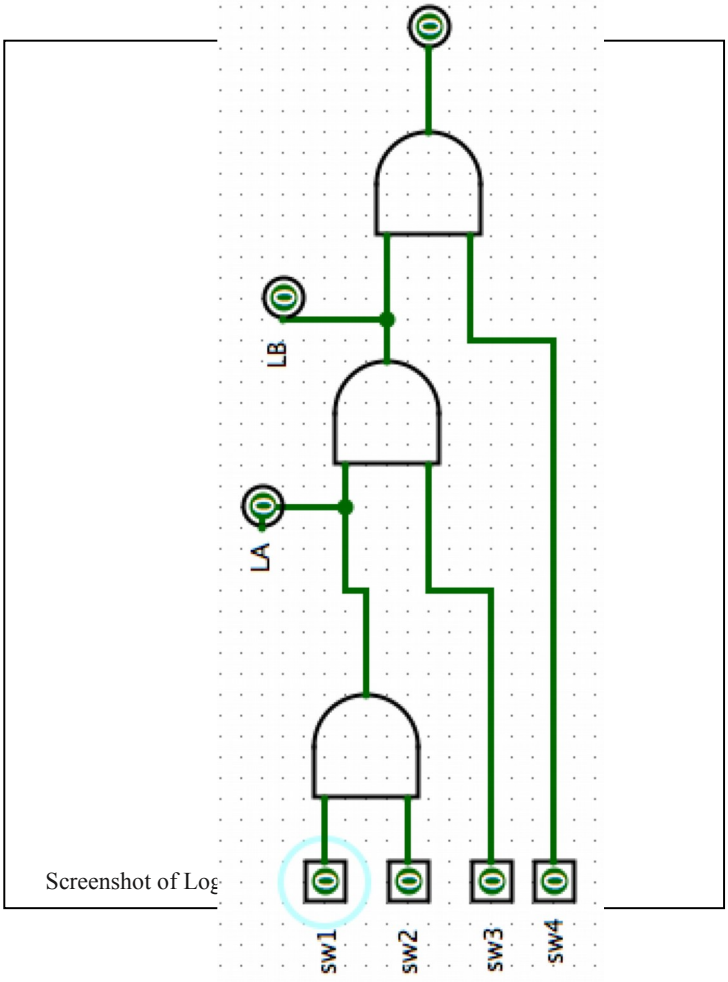




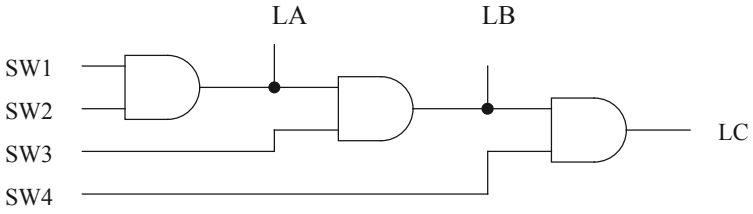
4a. AND gate function (AB)

Create the circuit as shown in Figure 4 using Logisim. Set the switches as indicated in the truth table of Figure 4 and record the light conditions.

SW1	SW2	SW3	SW4	LA	LB	LC
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	0	0	0
1	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	0	0	0
1	0	1	1	0	0	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	1	0
1	1	1	1	1	1	1

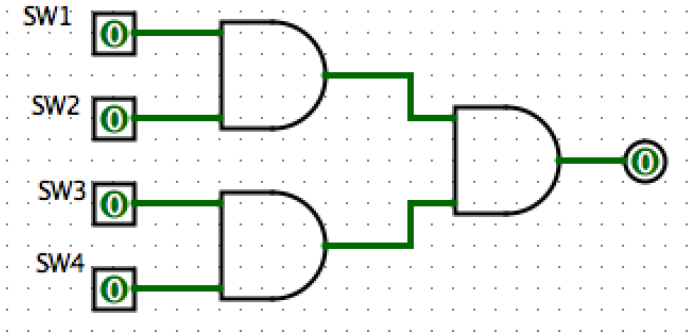


Screenshot of Log



4b Equivalent two-level 4-input circuit

Propagation delay is the time between application of a change to the input and the appearance of the resulting change on the output. While the time is insignificant in human terms (on the order of 10-15 nanoseconds), it often becomes critical in high speed circuits. The above circuit contains three propagation delays, hence the name three-level circuit. However, it may be reduced to a two level circuit using the same number of gates.
Draw the equivalent two-level 4-input circuit using three 2-input AND gates in the space below. Only the output (LC) column of the truth table need be satisfied. You do not need to construct the circuit in Logisim.



The output of the LC column would be the same as in 4a.

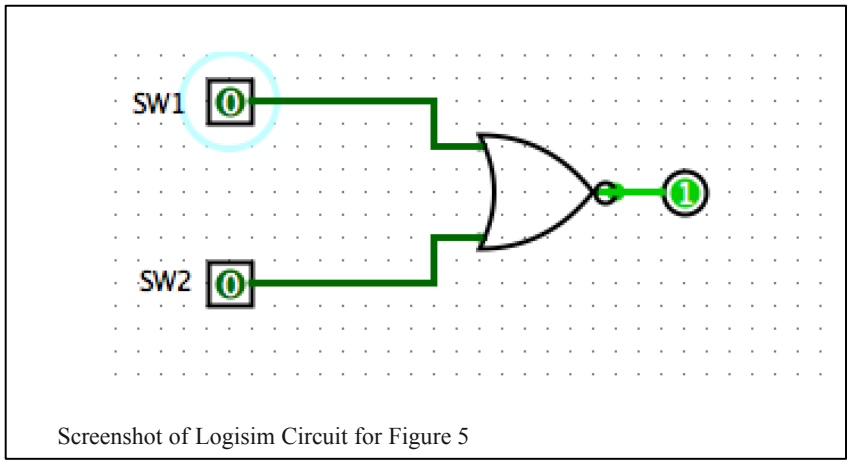
Equivalent two-level 4-input circuit

5. NOR gate function $(A + B)_{!}$

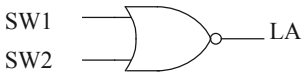
Create the circuit as shown in Figure 5 in Logisim. Set the switches as indicated in the truth table of Figure 5 and record the light indications. Note that the truth table is identical but reversed to that of the OR gate. Thus the NOR gate is identical to an OR gate followed by an inverter.

Figure 5

SW1	SW2	LA
0	0	1
0	1	0
1	0	0
1	1	0



Screenshot of Logisim Circuit for Figure 5

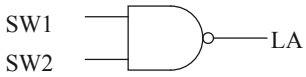


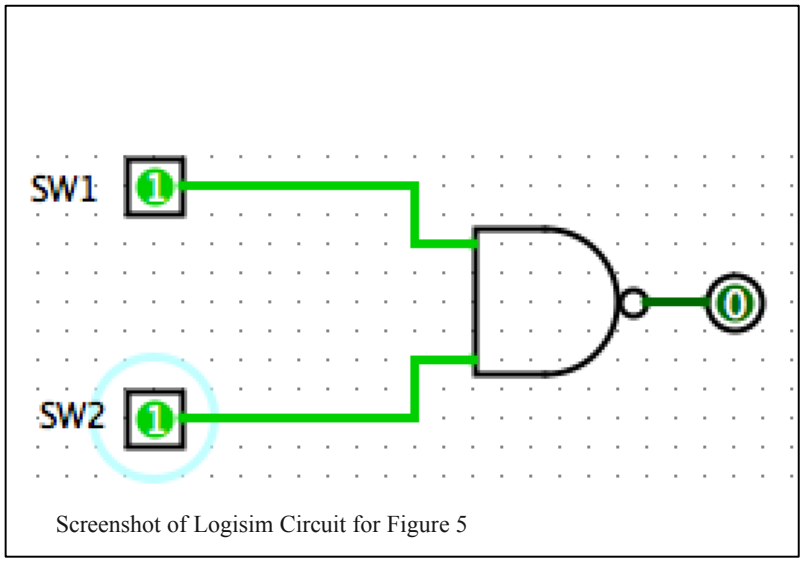
6. NAND gate function $(AB)_!$

Create the circuit in Figure 6 using Logisim. Set the switches as indicated in the truth table of Figure 6 and record the light conditions. Note that the NAND function is to the AND as the NOR is to the OR.

Figure6

SW1	SW2	LA
0	0	1
0	1	1
1	0	1
1	1	0





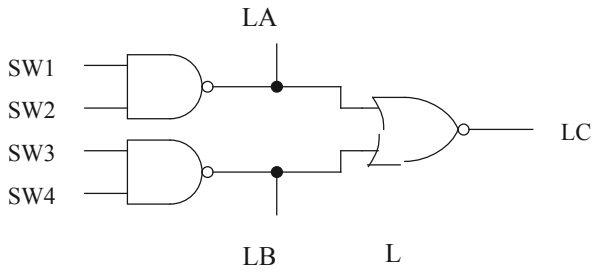
Screenshot of Logisim Circuit for Figure 5

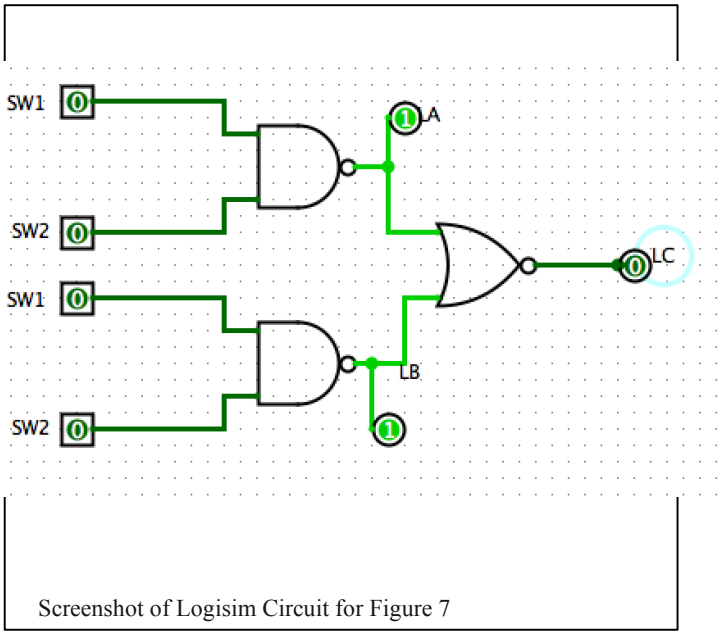
7. The negated-input OR and the negated-input AND concept

Create the circuit shown in Figure 7 in Logisim. Set the switches as indicated in the truth table of Figure 7 and record the light indications. Compare the truth table of Figure 7 with that of the 4-input AND gate of Figure 4. Note that the output columns (LC) are identical. Therefore the circuit of Figure 7 is performing the 4-input AND function, although the second level gate is a NOR gate.

Figure 7

SW1	SW2	SW3	SW4	LA	LB	LC
0	0	0	0	1	1	0
0	0	0	1	1	1	0
0	0	1	0	1	1	0
0	0	1	1	1	0	0
0	1	0	0	1	1	0
0	1	0	1	1	1	0
0	1	1	0	1	1	0
0	1	1	1	1	0	0
1	0	0	0	1	1	0
1	0	0	1	1	1	0
1	0	1	0	1	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	1	0
1	1	1	0	0	0	1
1	1	1	1	0	0	1





Screenshot of Logisim Circuit for Figure 7

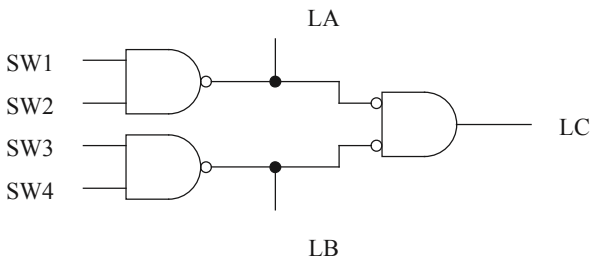
Inspection of the circuit in Figure 7 is misleading as to its actual function, due to the NOR symbol, especially if it were part of a complex logic diagram which was being analyzed. Therefore, it has become conventional to draw the negated input AND function of the NOR gate as in Figure 8.

Figure 8

NOR gate equivalent to negated-input AND gate

Figure 9 shows the equivalent circuit of Figure 7, redrawn so that its function becomes readily apparent. In your mind you can simply cancel the effect of two inverters in a row.

Figure 9

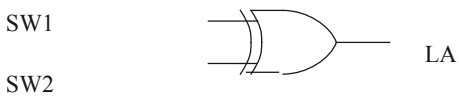


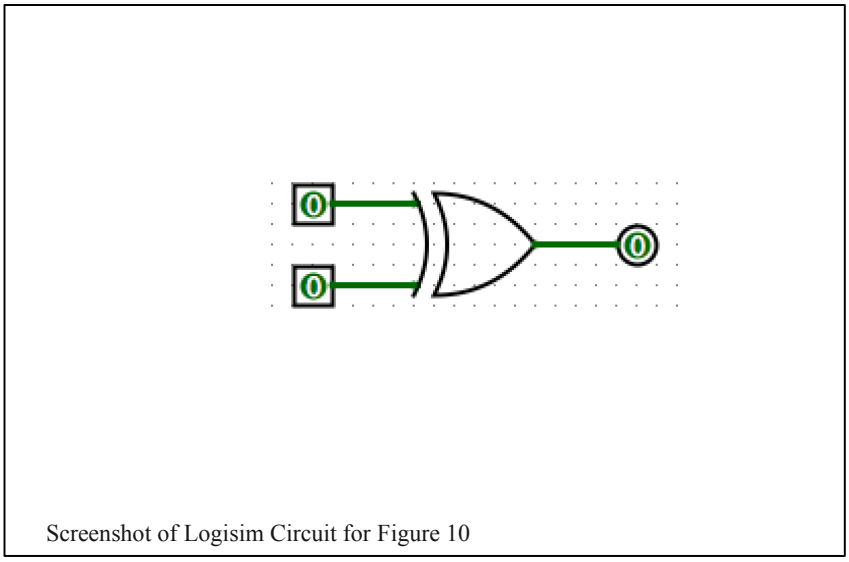
8. The XOR function

Create the circuit shown in Figure 10 in Logism. Set the switches as shown in the truth table of Figure 10 and record the light indications.

Figure 10

SW1	SW2	LA
0	0	0
0	1	1
1	0	1
1	1	0





Screenshot of Logisim Circuit for Figure 10

It can be seen from the truth table that the XOR gate is similar to the OR gate with the exception of the state in which both inputs are high. This feature makes the gate useful for binary adders, parity generators, etc.

You have now illustrated the function of each type of gate in actual operation, except for the inverter. As the inverter simply changes the input signal to the opposite level (1 to 0, 0 to 1) we will illustrate it only by its function as we use it in the next setups.

Normally there is a large preponderance of NAND and NOR gates in a logic design, with few or no AND and OR gates. As we have seen, the NAND and NOR gates invert the signal, and this low assertion is useful in a great number of instances. For example, the NAND and NOR gates may be used as inverts by applying the input signal to all of the inputs of the gate, or by tieing unused NAND inputs to VCC and unused NOR inputs to GND. Thus we may invert the outputs of the NAND and NOR gates by adding another gate as an inverter so as to obtain the AND and OR functions. However, as the AND and OR gates do not invert, it is not possible to derive the NAND and NOR functions from them.

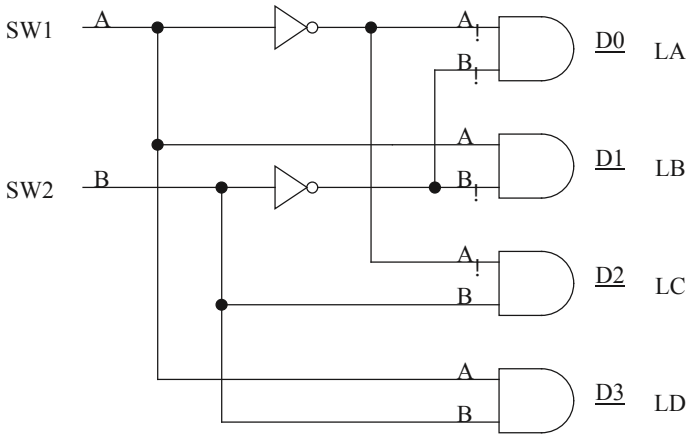
9a. A mystery circuit

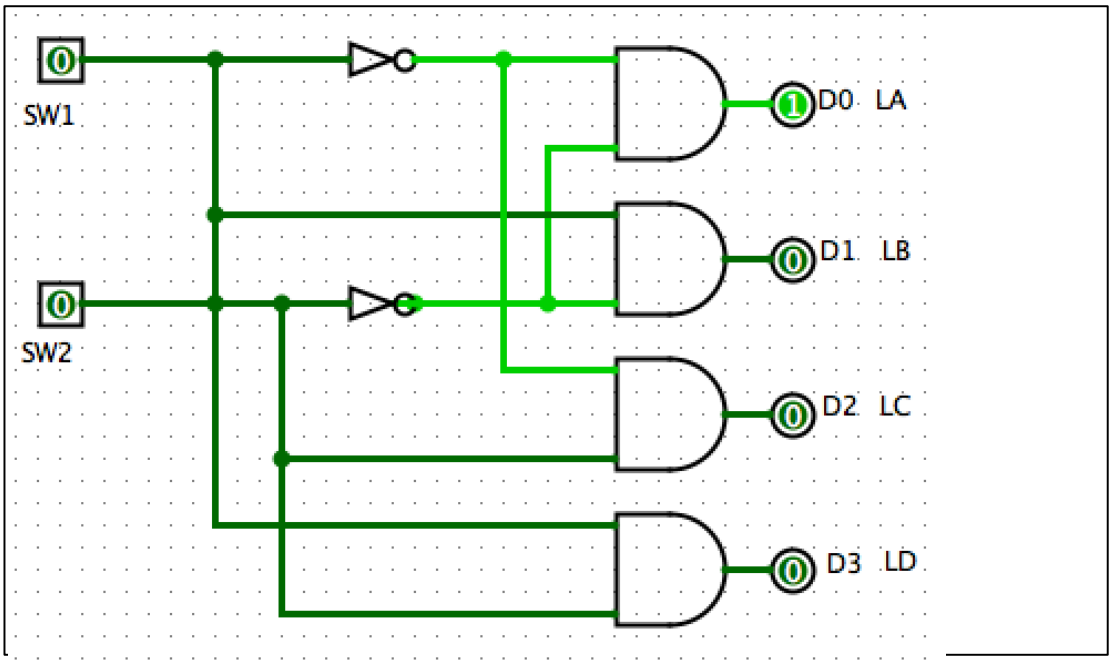
Create the circuit of Figure 11in Logisim. Set the switches as shown in the truth table of Figure 11. Set the switches as shown in the truth table of Figure 11 and record the light indications. If the inputs are considered as a two-bit binary word, with A being the least significant bit (LSB) and B being the most significant bit (MSB), what is the name of this circuit?

Name of circuit: _____binary decoder_____

Figure 11

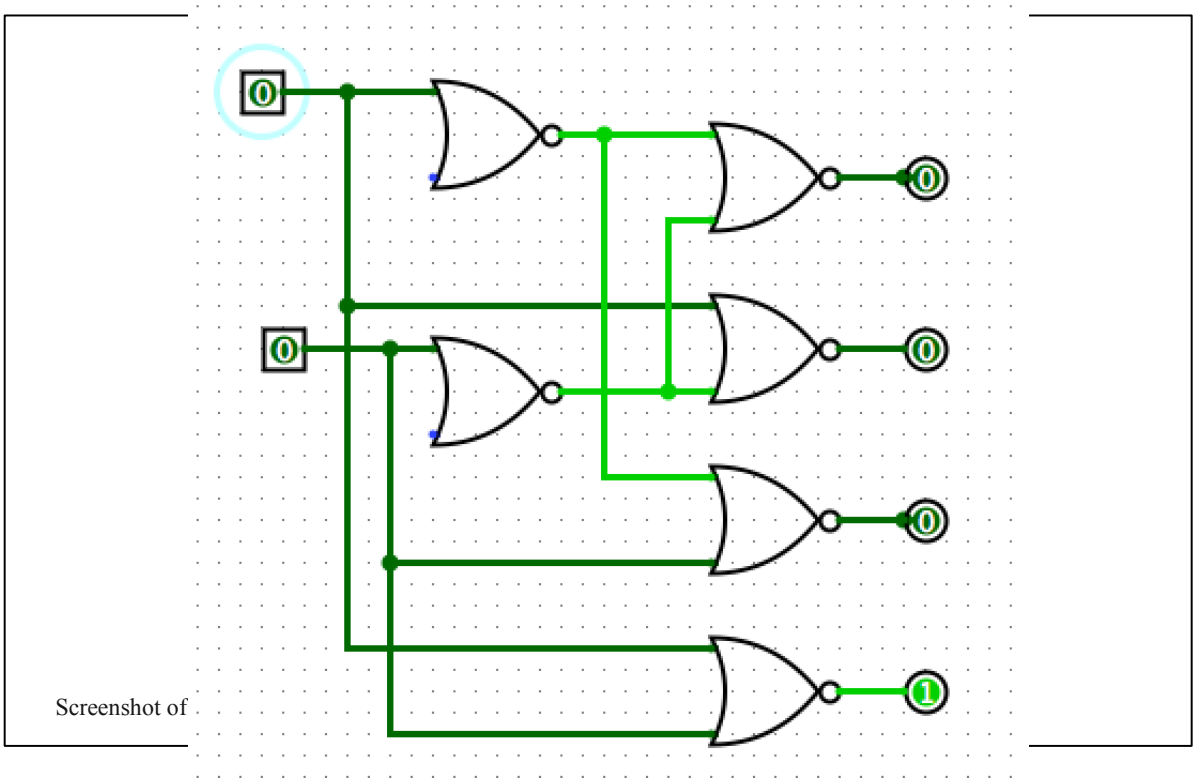
B SW2	A SW1	D0 LA	D1 LB	D2 LC	D3 LD
0	0	1	0	0	0
0	1	0	0	1	0
1	0	0	1	0	0
1	1	0	0	0	1





9b. Equivilent circuit with 6 NOR gates

Redesign the circuit to operate in exactly the same manner if the only components available were six NOR gates. Use two gates as inverters and draw the other four as negated-input AND gates. Show the circuit diagram below but do not construct the circuit in the lab.



Screenshot of