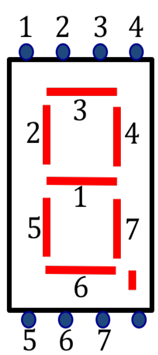
Seven Segment Circuit Design

Design a circuit that correctly shows the numbers 0-7 on a seven segment display given a 3-bit binary input.

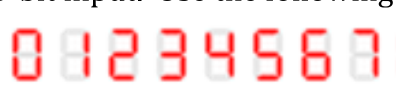
deliverables: You will turn in **2** items. The 1st file should be named sevenSegment.circ and contain your circuit implemented in logisim; submit this file to Moodle. The 2nd part should contain your truth table, karnaugh maps, and boolean expressions from steps 1-3 below. You may submit this as an electronic file or turn it in on paper at the beginning of class Wednesday (or put it in my mailbox anytime before that.

A seven-segment display is a familiar output device that is used to display numbers and a few letters. Simple alarm clocks, old calculators, and lots of other devices use seven-segment displays. Logism has a seven-segment display widget in its Input/Output section. Your job is to create a Logisim circuit taking three input lines representing an integer between 0 and 7 (each input is 1 bit of a 3-bit binary number), and produce output on a seven-segment display widget showing the corresponding digit. You may use AND, OR, and NOT gates, plus necessary pins and wires. The diagram below shows how the inputs to the Logisim seven-segment display correspond to the 7 segments. If an input is 1, the corresponding segment will be turned on. The 8th input connects to an optional decimal point that can light up, but we will ignore that.



You will complete this in the following steps:

1. Make a truth table showing the value of each segment (1 meaning on and 0 meaning off) for each 3-bit input. Use the following format for each number.



For example, the first row would look like

x_0	x_1	x_2		seg1	seg2	seg3	seg4	seg5	seg6	seg7
0	0	0		0	1	1	1	1	1	1

2. Create a karnaugh map for each segment, and circle or somehow mark the groups of 1's. Note that the segments are completely independent, you are basically creating 7 different circuits using the same 3 inputs, 1 for each segment. The completed truth table for seg1 is

x_0	x_1	x_2		seg1
0	0	0		0
0	0	1		0
0	1	0		1
0	1	1		1
1	0	0		1
1	0	1		1
1	1	0		1
1	1	1		0

which gives the karnaugh map

		x_1x_2			
		00	**01**	**11**	**10**
x_0	**0**	0	0	1	1
	1	1	1	0	1

3. Use the karnaugh map to write out a simplified boolean algebraic expression for each segment. For the above map I get the following expression representing seg1

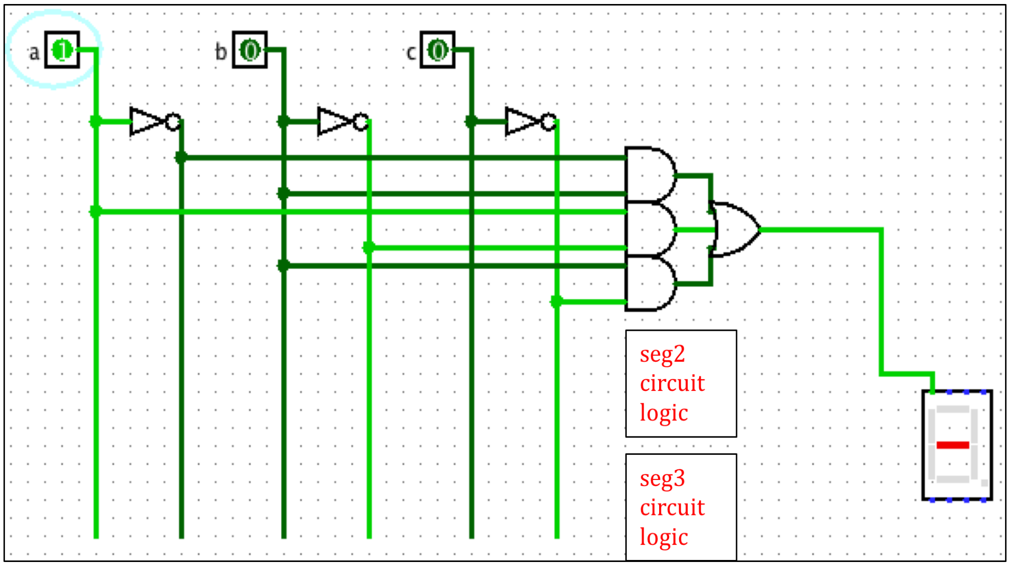
$$x_0 \bullet \overline{x_1} + \overline{x_0} \bullet x_1 + x_1 \bullet \overline{x_2}$$

Note that there may be multiple equivalent expressions for a segment that are equally simple. For example on the above map I could have circled the 1's in the lower left & right corners instead of the last column, giving the expression

$$x_0 \bullet \overline{x_1} + \overline{x_0} \bullet x_1 + x_0 \bullet \overline{x_3}$$

Any one of these equivalent expressions are acceptable.

4. Finally use your 7 expressions to create your circuit in logisim. Your final circuit should be clean and easy to interpret. Note that you can choose between 3 sizes of each gate in logisim to help make this easier. Below is a circuit for seg1, you should add the other 6 circuits in a column below seg1 as shown in the diagram, using the same inputs (you will need to extend the input lines further down).


a
1
b
0
c
0
seg2
circuit
logic
seg3
circuit
logic