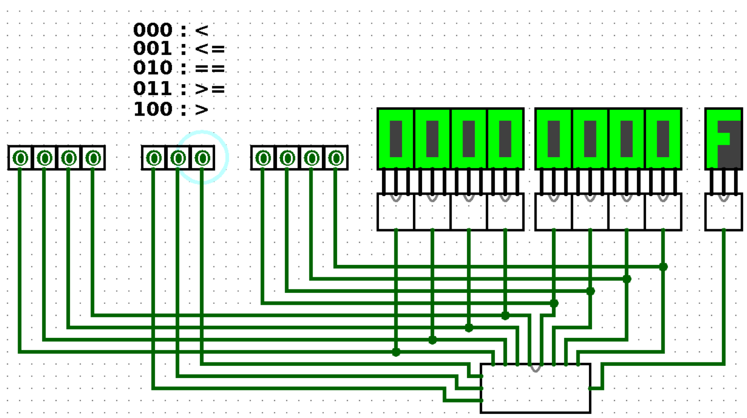
Instructions

Your second project with Logisim is to create a specialized circuit implementing all five basic Boolean comparison operators from the set

$$\{<, \leq, =, \geq, >\}$$

for two unsigned binary operands. You must display the input and result with two sets of LED matrices displaying inputs in [0000, 1111] and one LED matrix displaying the comparison's result (**F** or **T**) respectively. The comparison will be selected via a binary input.



The task requires nine 3 x 5 LED matrices—four for each binary input and one for the character output. You must also provide 11 1-bit two-state binary input—four for each unsigned 4-bit binary input and three to select the comparison operation as enumerated below. You will likely also require a 1-bit LED matrix driver for each of your two sets 0, 1 and F, T, to be used to control your LED matrices.

In addition to the necessary subcircuits described above, a set of comparators (1-bit and 4-bit) building up to the described comparator circuit will make the problem more manageable and separable. You do not necessarily have to build a specialized subcircuit. If you build a general-purpose comparator, your logic may be outside that circuit.

A helpful circuit for this project will likely the be the multiplexor. Using it to map the selected output of your binary comparator to the {F, T} binary driver.

I recommend the following strategy:

1. Build a full 1-bit binary comparator—that is a circuit accepting 5 input bits: a and b for an $a = b$ comparison, for example, and eq_{in}, gt_{in}, and lt_{in}, where

   - eq_{out} : $a = b$ **AND** eq_{in},
   - gt_{out} : $a > b$ **OR** ($a = b$ **AND** gt_{in}), and
   - lt_{out} : $a < b$ **OR** ($a = b$ **AND** lt_{in})

   Represent the input from a previous full 1-bit binary comparator.

2. Build a 4-bit binary comparator from four 1-bit full binary comparators. Each 1-bit comparator
   (a) Accepts one pair $a_i b_i$ of the four bits of the 4-bit input,
   (b) Compares that pair along with the result of any previous comparisons $a_{i-1}b_{i-1}$ using the output of some $i-1^{\text{th}}$ input, e.g.,
      - $eq_{out} := a_i = b_i$ AND $a_{i-1} = b_{i-1}$
      - $gt_{out} := a_i > b_i$ OR ($a_i = b_i$ AND $a_{i-1} > b_{i-1}$)
      - $lt_{out} := a_i < b_i$ OR ($a_i = b_i$ AND $a_{i-1} < b_{i-1}$)
   (c) See Page 309. Solved Problem 4 for an example drawing. It is an 8-bit built from two 4-bit, but the idea is the same.
3. Build a 1-bit binary LCD matrix driver.
4. Build a 1-bit Boolean {F, T} LCD matrix driver.
5. Use an eight way multiplexor to map the output of your 4-bit binary comparator using the mapping provided below.
6. Add the required 11 input pins and map them as shown in the illustration.

Operator Mapping:

000: $<$ 001: $\leq$

010: $=$

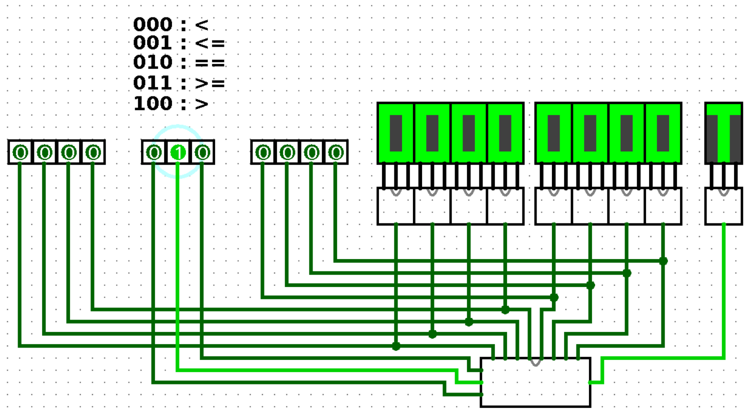
011: $\geq$ 100: $>$

Points

You must mark the file you want graded as "main." Right-click on the file you want to make main and set it. You may use no external libraries, i.e. arithmetic. Any group doing so will receive a zero for this project.

There are thee points available in this project. Partial credit will only be awarded for wholly completed parts below.

1. Program a binary LED matrix driver to display the input bits as directed above and illustrated in the provided graphic(s). **1.0 Points**
2. Program a 4-bit binary comparator with one output bit, three select input bits, comparing two 4-bit unsigned binary integers. Without part three, I will look at the output bit of this circuit to evaluate its correctness. Ensure you correctly label and/or document your output bit if you do not complete all three steps. **1.0 Points**
3. Program an LED matrix driver to display F for 0 and T for 1 and attach it to the output of the comparator above. Without part 2, you must provide an input pin for evaluation of the driver's behavior for credit. **1.0 Points**



The End.