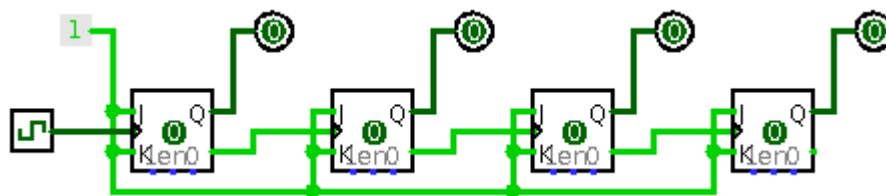


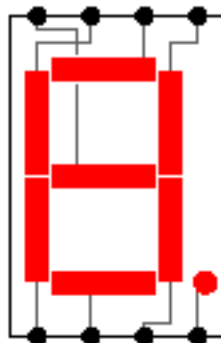
In this lab you will use Logisim to construct a 4-bit counter that is linked to the clock tool and uses JK flip flops. Then, also using Logisim, you will design a circuit that will translate the 4-bit binary counter to output the hexadecimal counter value and display it on the 7-segment display tool. When it is complete you will turn on the tick simulation to view the 7-segment display cycle from 0 to F.

1. First we construct a 4-bit counter from JK flip flops. The diagram below shows the construction. Build the circuit in Logisim and use the poke tool to increment the counter by turning the clock pulse on and off. The 1 in the upper left of the circuit is the constant tool, which is under Wiring. The default setting for the constant is a single bit with a 1 output, which is what you need for this circuit.

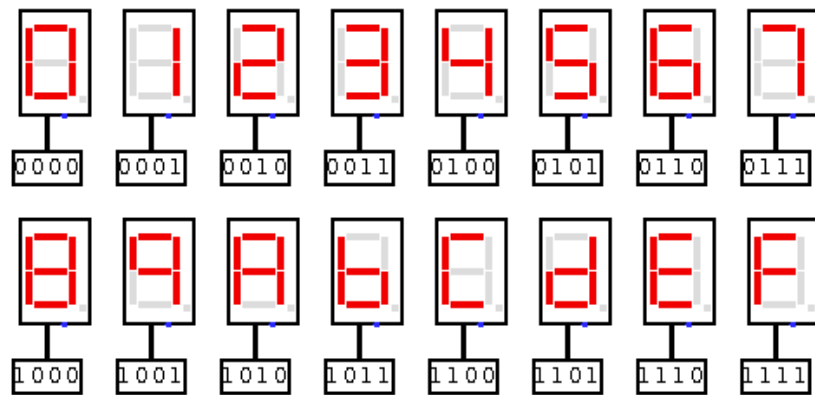
The logic of the circuit is fairly simple, trace through it and notice that we are using the 11 toggle input and allowing Q' to take care of the binary carries. If you select Simulate > Ticks Enabled (or Ctrl+K), this will turn on the clock simulator and you can watch the binary number increase to 1111 and then reset to 0000 to begin cycling around. Note that the least significant bit in this circuit is on the far left. You can also change the speed of the clock ticks by selecting Simulate > Tick Frequency > 4 Hz (or whatever speed you would like).



2. The 7-segment display tool (under Input/Output) has 8 pins to it, 4 on top and 4 on the bottom. The one in the lower right is for the decimal point and is not needed in this lab. The way it is wired is shown in the diagram below. Create a new circuit board to use (Project > Add Circuit), hook up 7 input pins to the top 4 and bottom left three input pins of the display and then use the poke tool to get a feel for how this display works.

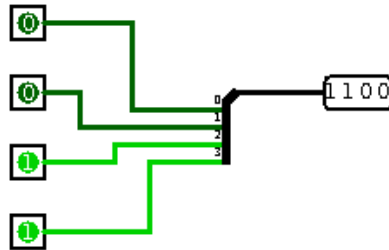


3. Although Logisim has a hexadecimal display tool you must use the 7-segment display tool for this lab. Create a new circuit board to use and design a circuit that translates a 4-bit input to its hexadecimal character on the 7-segment display. This sounds harder than it really is. Think of the 4-bit input as 4 separate binary digit inputs and the position of segment to display as the output. So this circuit will have 4 inputs and 7 outputs. For example, the top bar is the third pin from the left on the top of the display. The inputs of, 0, 2, 3, 5, 6, 7, 8, 9, A, C, E, and F must all turn on this segment. So each of these inputs must have that pin as an output. Use the Analyse Circuit dialog to create the truth table for this circuit for all 16 possible combinations of inputs and the 7 outputs. Then let Logisim build the translation circuit. You may use multiple input gates (no need to restrict to two). We could also construct this circuit using plexers later in the lab, but we will let Logisim use the standard AND, OR, and NOT gates at this point.



4. Create another new circuit board and copy both the counter and translation circuit to the new board. Add in a 7-segment display tool. Hook up the outputs of the counter to the inputs of the translator, be careful that the bit correspondence is correct. Then hook up the outputs of the translator to the 7-segment display tool. Using the poke tool or ticks simulator, test the circuit. When you get it working, show me your circuit and test it for me.
5. Now we are going to design the same circuit using a decoder. Create another new circuit board, add in a decoder, 7 OR gates, and a 7-segment display. Set the decoder to have 4 select bits, this will produce 16 output contacts numbered 0 – 15, just enough to do hexadecimal. Link each of the OR gates to one of the bars in the 7-segment display. Then take each output of the decoder and split it into multiple lines that feed into the OR gates that will produce the correct image on the 7-segment display. This decoder needs a 4 bit selection input, so add in an input tool, change its Data Bits to 4 and feed this into the decoder. Test that each binary number is converted to the correct hexadecimal image.
6. Now we would like to link this up to the 4-bit counter from number 1. The problem here is that the counter has four 1-bit outputs and the decoder requires one 4-bit input. So we need to create a circuit that will do this translation, fortunately Logisim has a

tool for this, the Splitter (which is really a splitter and joiner). Add a Splitter tool, set both the Fan Out and the Bit Width In to 4. Now, just for testing, add in 4 1-bit inputs and one 4-bit output, as pictured below. Use the poke tool to get a feel for the bits being updated, the LSB is on top and the MSB is on the bottom.



7. We are now ready to hook the whole thing up. Create another new circuit board and copy both the counter and the decoder to 7-segment display circuit to the new board leave space between the two for the converter we just built. Now copy the four 1-bit to one 4-bit converter to the new board. Link the counter to the converter and the converter to the decoder. Using the poke tool or ticks simulator, test the circuit. When you get it working, show me your circuit and test it for me.

Turn In Your Work:

Use the text tool to put the names of the members of your group on the circuit boards. Finally, upload the circ file to the MyClasses page for this lab. The circ file should contain all of the boards you created in this lab.