CS270 Recitation 11
"Designing an Incrementer in Logisim"

Goals

To understand how to design and debug "fat" combinatorial circuits using Logisim. Fat circuits are those that operate at the "word-level" rather than at the level of a single bit.

The Assignment

You are going to design, implement and test an ***8-bit incrementer***.

This is similar to the circuit that's inside the +1 box in the LC-3 datapath and produces the value PC+1 (the incrementer in the LC-3 uses 16 bits). It can easily be implemented as an adder, one of whose inputs is hardwired to the integer 1, but we want you to build it from scratch. There are a number of steps, and ***logisim*** is only the last one.

1. *First understand the algorithm that your circuit is to implement.*

   In order to add 1 to any (2's complement or unsigned) binary integer, we use the grade school algorithm: work our way from the least significant bit (LSB) to the left towards the MSB. Work out three examples on paper. Show them to your TA.

2. *Decompose it into a function for each bit in the input, and write their truth tables.*

   The input at any bit position is an input bit A, and a carry bit C_{in}, and the result of the addition at that position is an output bit S, and a carry bit C_{out}. So, our first step is to write a truth table that specifies these two outputs as a function of the two inputs. Since there are two inputs and two outputs, your truth table should have 4 rows and 4 columns. Show it to your TA.

   Do not proceed to step 3 before a TA has checked off your truth table.

3. *Now implement and test it in **logisim***

   a. Make a subdirectory called R11 for this recitation, where all files should reside.
   b. We want your design to be modular. So first implement a (sub) circuit called Incr1bit which is in charge of incrementing one single bit.
   c. Build your complete incrementer circuit by making eight copies of the Incr1bit sub-circuit and connecting them appropriately. Use splitters (look it up in the user guide) so that you specify an 8-bit bundle of wires as the input and a similar 8-bit bundle of wires as the output.
   d. What is the input to the LSB?
   e. What is the output from the MSB?
   f. Finally, note that the bit-0 sub-circuit can be simplified. Design another sub-circuit called IncrBit0 and use that instead of Incr1Bit for the LSB.