COM1006/COM1090 Devices and Networks (Autumn)
Tutorial Sheet #6: Implementing Arithmetic

1. Implement a 4-bit subtracter: For two 4-bit numbers X, Y that are expressed in two's complement representation your circuit should compute X-Y. Recall that X-Y can be written as X + (-Y), so you need to add X with the two's complement of Y. Use only elementary gates (AND, OR, NOT, NAND, NOR, EOR/XOR) and full adders wired like in a ripple-carry adder (find full adders as "Adder" in Logisim's "Arithmetic" folder). You may want to use the following 4-bit ripple-carry adder as a starting point:
   https://staffwww.dcs.shef.ac.uk/people/D.Sudholt/campus_only/misc/ripple-carry-adder-4.circ
   (you might have to remove the file ending ".xml")

   Hint: there is an easy way to add 1 when computing the two's complement of Y – think of the carry-in to the first bit. There is a solution using only 4 full adders.

2. Add a LED to your subtracter from exercise 1. which lights when an arithmetic overflow occurs.

   Hint: recall that you can look at the carry-in and carry-out of the MSB.

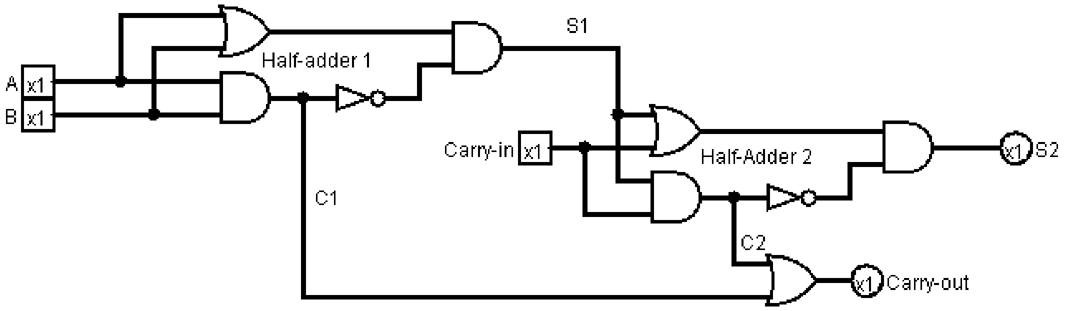
3. Download a stub for a 4-bit multiplication circuit from
   https://staffwww.dcs.shef.ac.uk/people/D.Sudholt/campus_only/misc/multiplier-4-stub.circ
   (remove the file ending ".xml" if necessary) and complete it as follows.

   a) To the right of the AND gates you find four splitters. Each splitter combines 8 input bits into a single multi-bit line. Complete the wiring to the splitters' inputs to create the partial products of multiplication as discussed in the lecture (e.g. 0000$a_3a_2a_1a_0$ AND $b_0b_0b_0b_0b_0b_0b_0b_0$). The LSB (least significant bit) is at the top and the MSB is at the bottom. Draw lines to the provided ground elements to set inputs to 0 (one such line can cover multiple inputs at once).

   b) Add those four partial products using Logisim's predefined multi-bit adders: find "Adder"s in the "Arithmetic" folder and make sure their data bits property is set to 8. Use a parallel arrangement of adders to reduce circuit depth.

   c) Output the result using an output pin with 8 data bits.

4. Consider the following full adder:



   a) What is the circuit depth of a single full adder?

   b) What is the number of gates the carry-in signal will traverse to get from the carry-in input to the carry-out output?
   Note that this number determines how long a signal needs to "ripple" from one full adder to the next. If you add a new full adder to a ripple-carry adder, the circuit depth increases by this number.

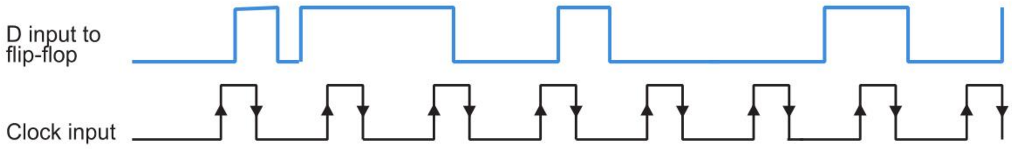
c) If A is your answer to a) and B is your answer to b), think of a 32-bit ripple-carry adder being constructed by taking one full adder (of depth A) and then adding 31 full adders (each increasing the depth by B). Derive the circuit depth of a 32-bit ripple-carry adder.

d) What is the circuit depth of a 32-bit carry look-ahead adder?
Hint: recall from Slide 15 that signals traverse 3 gates before reaching the *carry-in input* of any full adder.

e) How much faster is the 32-bit carry look-ahead adder compared to the 32-bit ripple-carry adder?

5. For the input and clock signals shown below, provide a timing diagram for the Q output of a D flip-flop. Assume that the flip-flop is:

    a) Level sensitive
    b) Positive edge triggered
    c) Negative edge triggered
    d) A master-slave flip-flop (assume that the master is triggered by positive edges and the slave is triggered by negative edges)



6. **[Optional exercise]** Implement an RS flip-flop in Logisim from cross-coupled NOR gates as in the lecture. Attach LEDs to both outputs. Label inputs R and S and outputs Q and NOT Q. Test whether your circuit correctly works as a memory element.