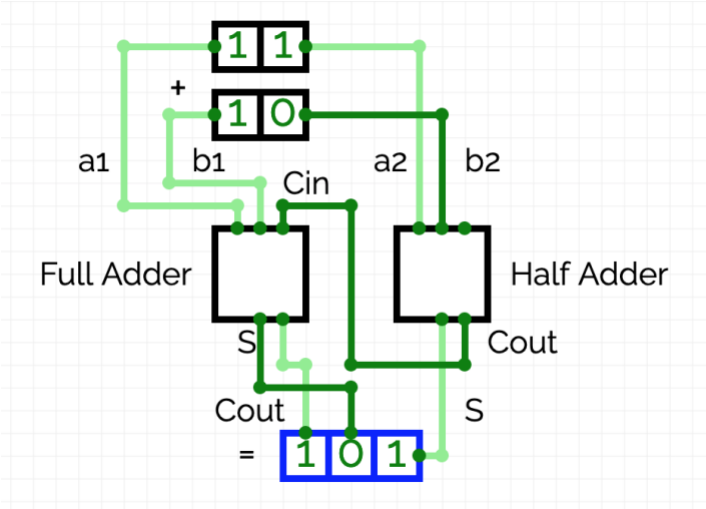
CSC 180, Lab #6
Fall 2024

Directions: Turn in a hard copy of this assignment, with your answers written on this or another sheet of paper. Circuits created in CircuitVerse should be printed out as described in class.

1. A circuit that outputs the Boolean value of $a > b$, where a and b are each 1 digit, is available here: https://circuitverse.org/users/89029/projects/a-gt-b-1-bit. Fork the circuit and use this circuit as a sub-circuit to create a circuit that outputs the value of $a > b$ when a and b are each 2 bits. If we denote $a = a_1a_2$ and $b = b_1b_2$, then the circuit is equivalent to: ($a_1 > b_1$) OR (NOT $a_1 > b_1$ AND $a_2 > b_2$). Use separate inputs for a_1, a_2, b_1, and b_2 and label the inputs. (Note: this problem is identical to the problem we did in class, on page 3 of the notes). [10 points]

2. An adder circuit is shown below.



Answer the questions corresponding to this adder circuit. [25 points]

   a. For the half-adder, specify the following values:

      a2 = sum (S) =
      b2 = carry out (Cout) =

   b. For the full-adder, specify the following values:

      a1 = sum (S) =
      b2 = carry out (Cout) =
carry in (Cin) =

c. What are the decimal values of the two numbers that are added together, and what is the decimal value of the sum (show your work by specifying powers of 2).

d. Complete the truth table below for the *Cout* output of the half-adder. What is the logical expression for obtaining *Cout* from *a* and *b*?

a	b	Cout
1	1	
1	0	
0	1	
0	0	

3. Create a two's complement circuit that finds the two's complement of a 3 bit number. You should use the following circuit as a starting point: https://circuitverse.org/users/89029/projects/two-s-complement-to-be-completed

   This starting point contains a 3 bit number, with lines connected to a *splitter* that will merge the 3 bits into a single output with a *BitWidth* of 3. Note that the *BitWidth* property allows you to control the number of bits used by various elements. We merge the 3 bits so that we can use a single adder circuit to find the sum (see the steps below). Recall that you can find the two's complement of a number by inverting each digit and then adding one. With this in mind, follow these steps to create a two's complement circuit. [15 points]

   1. Delete the lines from the inputs to the splitter (the splitter will be used at a later step).
   2. Invert each bit by adding appropriate NOT gates.
   3. Use the *splitter* to combine the 3 inverted bits into a single binary value.
   4. Add an *Adder* element, which is under the Miscellaneous (*Misc*) section. Change the *BitWidth* of the adder to 3.
   5. Use the adder to add 1 to the inverted input. In order to add 1 you will need to add an Input element, and change its *BitWidth* to 3, because this is what the adder is expecting. The value of this input should be 1 (denoted as 001).
   6. Use an Output element, with a *BitWidth* of 3, to display the output of the Sum from the adder. (The carry, or Cout, is ignored when finding the two's complement).
   7. Change the inputs to demonstrate that the two's complement of 010 is 110.