CS 270 Lab 4 (Boolean Algebra and Logic Circuits)

Week 4 - Oct. 16 – Oct. 20, 2017.

Name 1:____

Drexel Username 1:____

Name 2:____

Drexel Username 2:____

Name 3:____

Drexel Username 3:____

Grading:

Part 1 (25pts) ____

Part 2 (25pts) ____

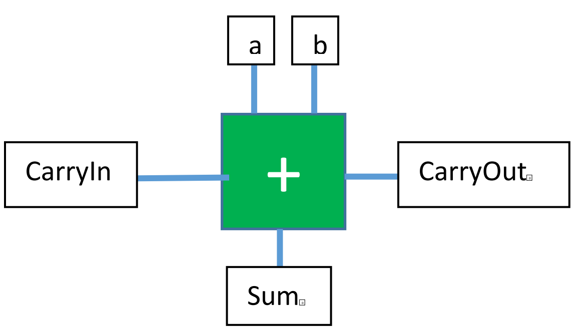
Part 3 (25pts) ____

Part 4 (25pts) ____

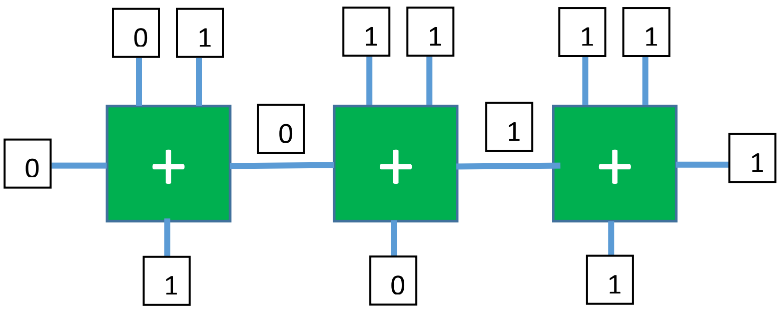
Instructions: For this exercise you are encouraged to work in groups of two or three so that you can discuss the problems, help each other when you get stuck and check your partners work. There are four problems relating to Boolean Algebra and Logic Circuits. The objective of the lab is to review the laws of Boolean algebra and disjunctive normal form (sum of products) and apply this knowledge to the design and simplification of logic circuits. The lab must be completed by the end of class on the following Wed.

The material for this lab is based on the lecture slides on Boolean Algebra and Simplification. Additional material on logic circuits can be found in the Foundations of Computer Science text. The questions in the lab concern a logic circuit called a Full Adder.

A full adder has 3 binary inputs (CarryIn, a and b) and two binary outputs (Sum and CarryOut).



The outputs are defined by the equations: a + b + CarryIn = 2 × CarryOut + Sum. A chain of full adders can be connected to perform binary addition. The CarryOut of one full adder is connected to the CarryIn of the next full adder. The CarryIn of the first full adder is set to zero and the CarryOut of the last full adder is the Carry of the binary addition. The individual a and b inputs of the i-th full adder are the ith bits of the input and the i-th Sum output is the i-th bit of the output. Such a chain is called a carry ripple adder. For example a chain of 3 full adders can be used to perform 3-bit binary addition. The low order bits are on the left and the initial CarryIn is set to 0.



In this example, the 3 bit numbers A = 110, which is 6, and B = 111, which is 7, are added to get the sum 1011, which is 13.

A full adder can be implemented using logic gates for the Boolean functions and, or, and xor. In the following questions you are to implement and simplify a full adder using Boolean functions and Boolean algebra to perform simplification.

1. Construct truth tables for the output bits (Sum and CarryOut) of a full adder.

2. Using the truth table derive a sum of products expression (disjunctive normal form) for Sum and CarryOut. Draw a circuit for this expression. You may use multi-input or and and gates (see the slides on Logic Circuits for the notation).

3. Consider the functions

   parity($x_1,\ldots,x_n$) = 1 if an odd number of x_is are 1 and 0 otherwise
   majority($x_1,\ldots,x_n$) = 1 if a majority of x_is are 1 and 0 otherwise

   a. Show how to compute the parity function using xor. Check your results for n=3
   b. Show how to compute the Sum and CarryOut functions using the parity and majority functions.

4. Using properties of Boolean algebra simplify your expressions from part 2. You should use xor for Sum. Draw the simplified circuits.