

## 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)** \_\_\_\_\_

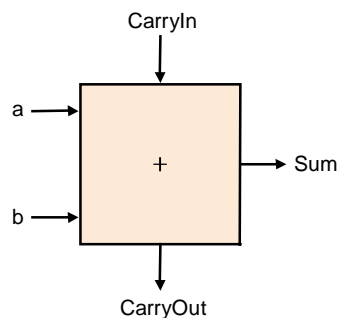
**Part 3 (25pts)** \_\_\_\_\_

**Part 4 (25pts)** \_\_\_\_\_

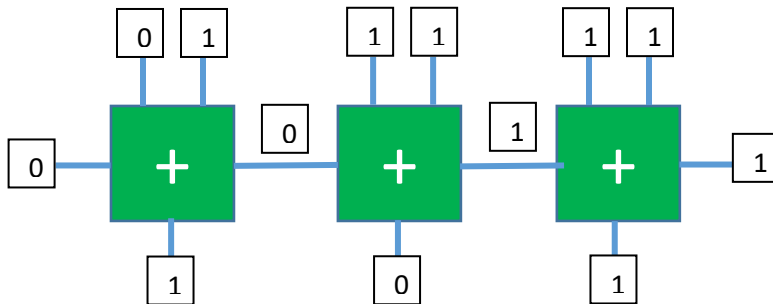
Instructions: For this exercise you are encouraged to work in groups of two or three so that you can discuss the problems, help each other when you get stuck and check your partners work. There are four problems relating to Boolean Algebra and Logic Circuits. The objective of the lab is to review the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

The 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:  $a + b + \text{CarryIn} = 2 \times \text{CarryOut} + \text{Sum}$ . 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  $i$ -th 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. In the diagram, the full adders are rotated and flipped so that the CarryIn is on the right and the inputs  $a$  and  $b$  are on the top. 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. Derive a truth table for the output bits (Sum and CarryOut) of a full adder
2. Using the truth table derive a sum of products expression for Sum and CarryOut. Draw a circuit for this expression. You may use multi-input or and and gates (see the slides on Logic Design for the notation)
3. Express Sum and CarryOut using the parity and majority functions
  - a.  $\text{parity}(x_1, \dots, x_n) = 1$  if an odd number of  $x_i$ s are 1
  - b. Write the parity function using xor. Check your results for  $n=3$
  - c.  $\text{majority}(x_1, \dots, x_n) = 1$  if a majority of  $x_i$ s are 1
4. Using properties of Boolean algebra simplify your expressions. Use xor for Sum. Draw the simplified circuits.