Lab 5: Half-adder + Half-adder = Full-Adder!

Introduction

Using half-adders as a building blocks, we can effectively combine two of these to form a full adder.
If we remember from last time:

"*Essentially, the adding [two binary numbers] is the most basic arithmetic operation. This simple addition consists of four possible elementary operations of 0 + 0 = 0, 0 + 1 = 1, 1 + 0 =*

1, and lastly 1 + 1 = 10. These operations produce a sum that is one digit in length except for the last one, 1 + 1 = 10. In this case, the result includes a carry bit."

However this time with full adders, the result forms the arithmetic sum of three bits. The full adder essentially consists of three inputs and two outputs. The first two inputs (x and y) represent the two significant bits to be added together. The final bit (z), is the carry from the previous lower significant position. With this, it is necessary to have two outputs denoted by S and C for solution and carry.

(40 points) Full adder design

Design a full adder that takes two 1 bit inputs (x and y) to represent the two significant bits to be added and another input (z) to represent the carry from the previous lower significant position. The outputs will have two bits (C and S).
Include the following in your lab report:

- Truth table by hand.
- K-maps by hand (there will be 2, one for output S and one for output C).
- Simplified logical expressions.

(60 points) Full-adder implementation, part 1: Breadboard

Using your design from above, implement a full adder by doing the following: Essentially, we will be using two half adders and an OR gate to implement out full adders. This time, we will implement half adders using XOR and AND gates (for S and C respectively). We will combine two of these with an OR gate to form our full adder. Include the following in your report:

- Logic circuit using Logisim. We will be using the 7486 (XOR), 7432 (OR), and 7408 (AND) chips.
- On the breadboard, implement the full adder. Demo this part to the TA.