HOMEWORK 2

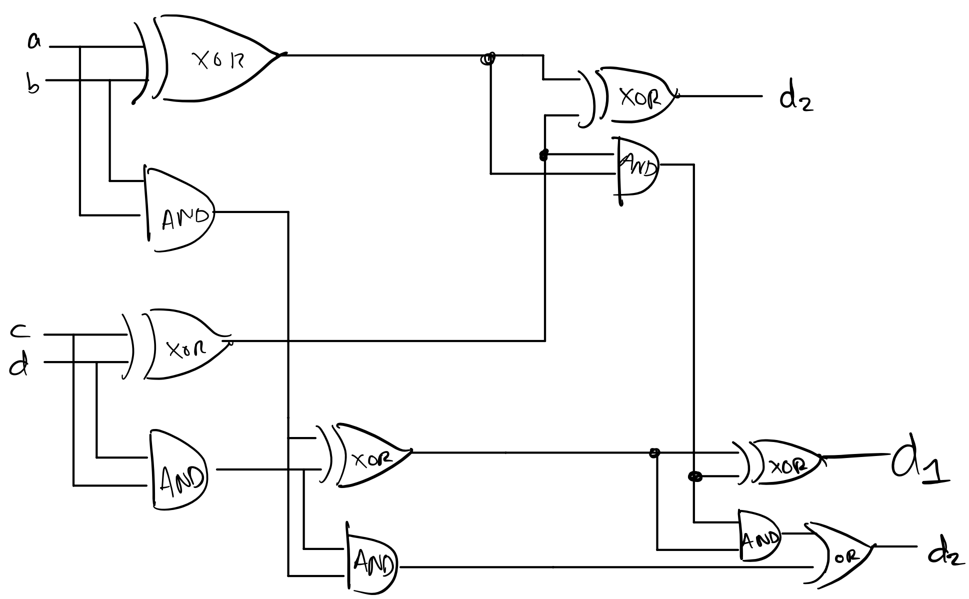
Name: Matthew Galitz **Computing ID:** ykk6rh

Collaborators: NONE !

Collaboration Policy: For this homework only! You may collaborate with other students in this class. As an **exception** to the usual collaboration policy, you do not need to tell us about casual interactions of the "I got X, what did you get?" variety. But **do** cite any close collaboration or major corrections; for example if the answer to the above hypothetical was "I think X is wrong, here's why" and then you change your answer, add a note like "mst3k suggested this answer" next to your answer. However, we expect that everyone will work on the assignment to better understand circuits, so **you may not directly copy another student's answer**.

PROBLEM 1 *4-input adder*

We have discussed both a 2-input and 3-input adders for single-bit values as we were building our ripple carry adder. Draw a 4-input adder for single-bit values: that is, a set of logic gates with 4 input wires (no need to name them) each representing a number between 0 and 1 and a multi-bit output z, composed of wires z_0 through $z_{\ldots}$ (where z_0 is the low-order bit, z_1 the next, etc., up to the number of wires needed for this task). The gates should ensure that $z =$ the sum of all four inputs.

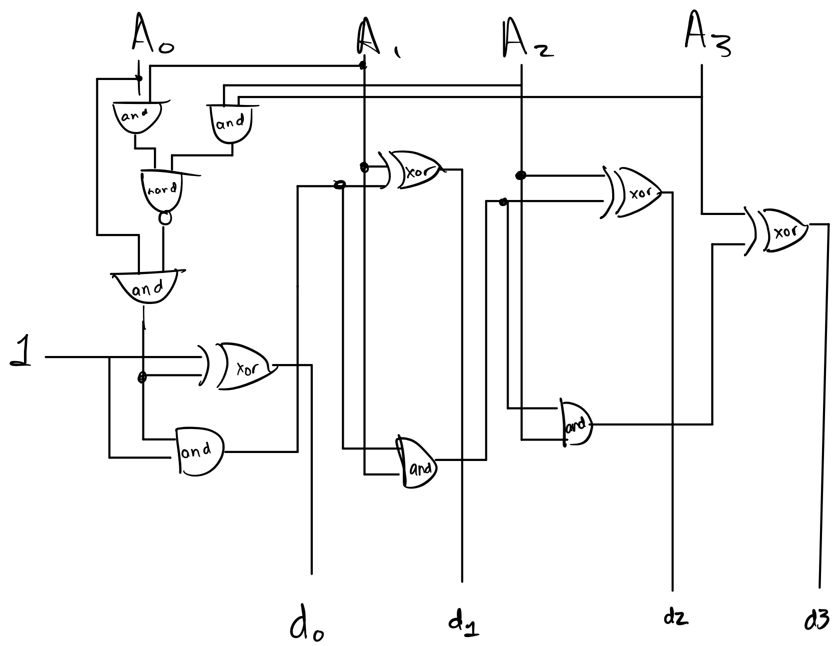


PROBLEM 2 *4-bit increment to 15*

In class we considered an increment circuit that adds 1 to its input value. How can we change our circuit to "stop" at `x = 0b1111`? That is, if `x` is not all 1s, then increment by 1. If `x` is all 1s, then increment by 0, i.e., `z = 0b1111`. Draw a circuit that does not use more than twice the number of gates in the original.

0 → least significant

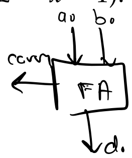
3 → most significant



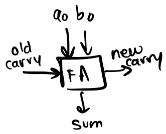
the output should be reflected

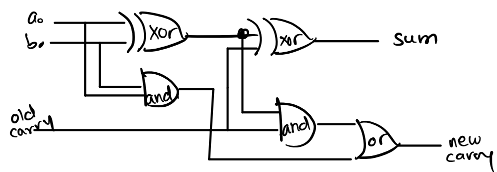
PROBLEM 3 *4-bit decrement*

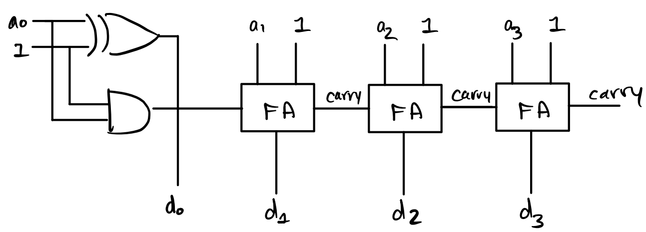
Now, rather than our 4-bit increment circuit that adds 1 to its input value, we want a circuit that subtracts 1 (i.e., $z = x - 1$). Draw a 4-bit decrement circuit that does **not** use not (~) gates.

I will use 

as shorthand for a full-adder.



is equivalent to




PROBLEM 4 *Fancy adder*

Given two 4-bit inputs x and y, draw a circuit that output the value z such that $z = x + x + y$. As a special property of **this circuit only**, we do **not** want overflow, so we have decided that z may have more than 4 bits to represent its value. Draw the corresponding circuit; label the output bits of z and state the number of bits needed in the output. *Hint: is there a fast way to calculate $x + x$ without using many gates?*