Digital Logic Design

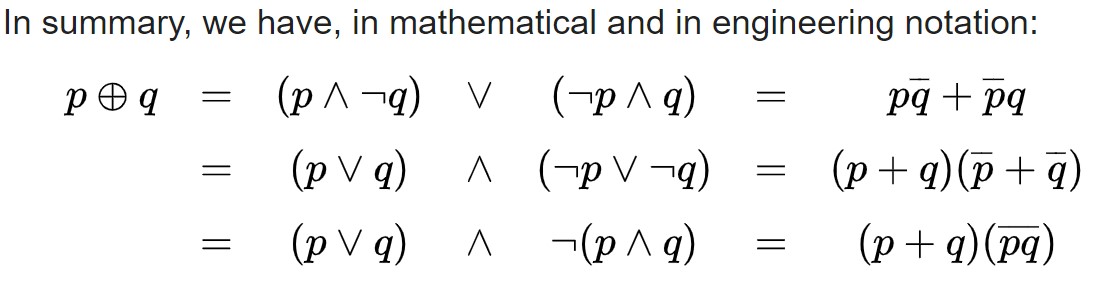
Summer STEM 2020

The Cooper Union

***Week 1 Day 3 ICE: TinkerCAD Circuits***

Problem 1:

The XOR operator can be defined in multiple forms; see this picture from Wikipedia:



Using TinkerCAD Circuits, verify that the statements on the right (ignore the mathematical notation in the middle) are equivalent by prototyping on a virtual breadboard. Use LEDs to check the outputs!

*Submit photos from TinkerCAD Circuits of at least 2 of the 3 – no need for Logisim.*

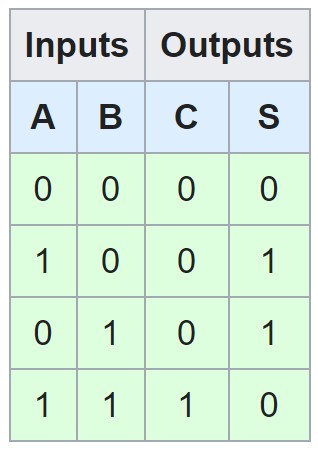
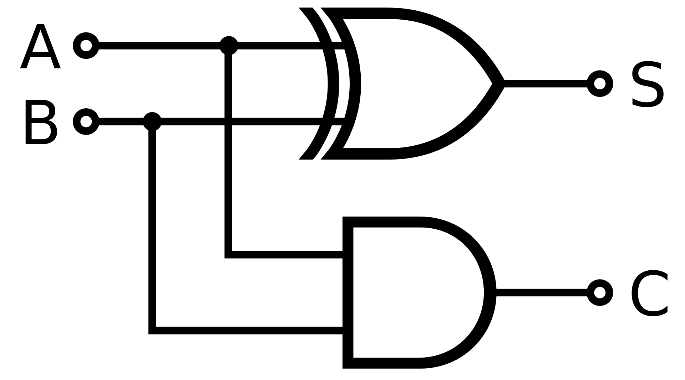
Problem 2:

In your DLD Summer STEM final projects, you may want to implement math operations

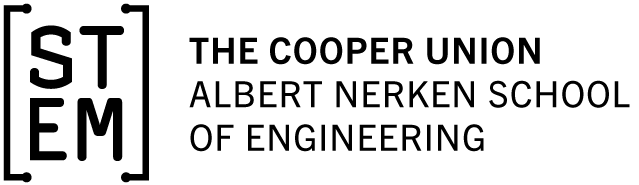
e.g. adding, subtracting etc. Computers use components called adders for adding any two n-bit binary numbers – in this exercise, you will learn how to construct an adder.

Half adders sum two single bits, *A* and *B*, and output the sum *S* and the carry bit *C*. Refer to this truth table and logic diagram from Wikipedia.

1. Why do half adders have two outputs, sum *S* and carry bit *C*? Hint: single bits.
2. Confirm using that the half adder logic works. *Upload a photo from TinkerCAD Circuits.*
3. Bonus: Explain how to derive the half adder logic.



1

Digital Logic Design

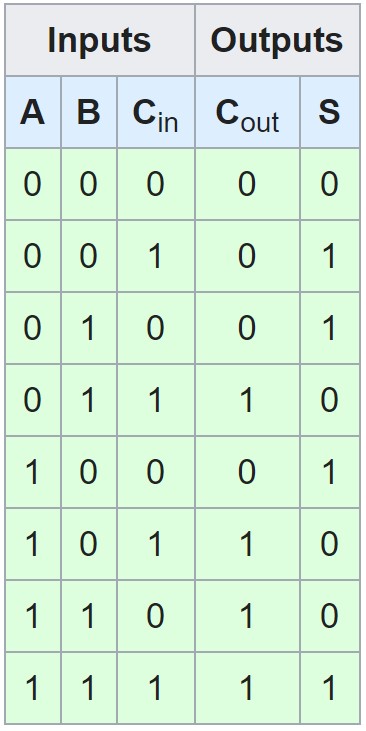
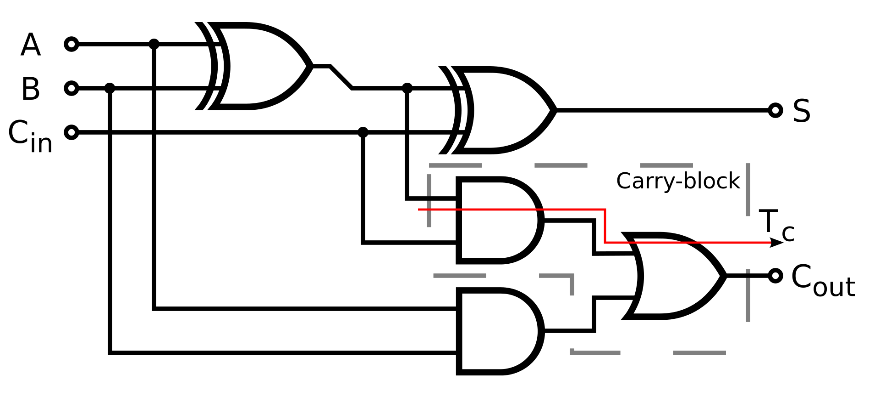
Summer STEM 2020

The Cooper Union

Problem 3:

Half adders are great… until you want to add more than one bit at a time. Full adders account for a carry-in bit *Cin* from a less-significant operation and can thus add larger numbers together; recall that half adders only add together two bits, and don’t account for carry-in bits. Refer to this truth table and logic diagram from Wikipedia.

1. Suppose we are adding 112 + 112; why won’t a half adder work? Write down the math on a piece of paper. Hint: look at the middle column.
2. Confirm that the full adder logic works. *Upload a photo from TinkerCAD Circuits*.
3. Bonus: Explain how to derive the full adder logic from half adders.

Tips:

* Submit the items in *italics* to Teams.
* Good work starts with showing your work!
  + Write down your Karnaugh maps and Boolean expressions on paper.
* Don’t be afraid to ask for help!
  + Consult your classmates and instructors if you get stuck.
  + Inform the instruction team if you have computer issues
* Follow good practices while breadboarding.
  + Good practices make it easy to debug your circuit. o Be consistent with wire colors – use red and black for power and ground only. o Keep wires running across only either rows or columns, never diagonally or curved. o Try to have only one wire per row.
  + Space components out properly – don’t make spaces between chips too wide or to narrow – 4 to 5 pins should be enough. For DIP switches, space 8 pins.
  + Don’t put more than one wire into one breadboard hole. o Use LEDs to show outputs. o Use SPST DIP switches to toggle inputs

2