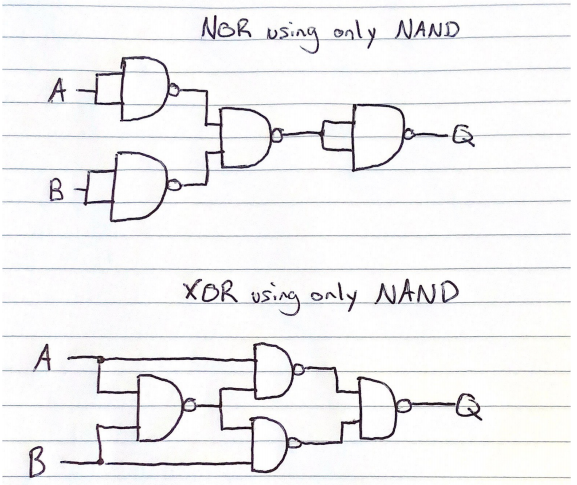
CPSC3300 – Computer Systems Organization  
Homework #2 – Boolean Algebra and Adders

Due: 11:59PM Monday, February 8

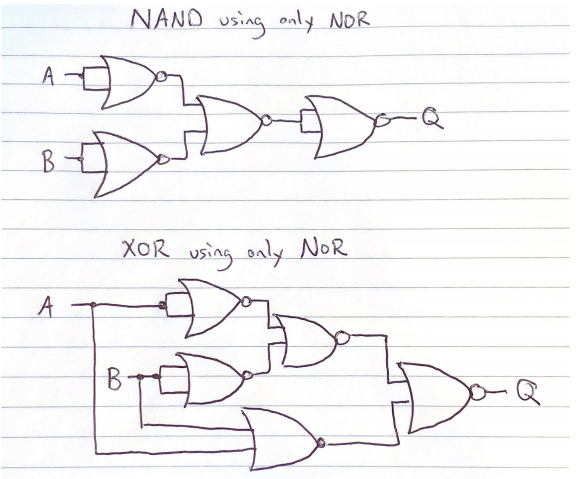
Submit to Canvas

Total 100pts

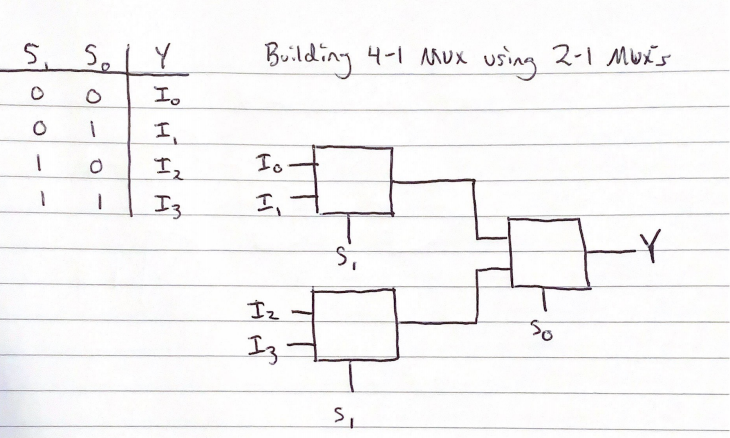
1. [20pts] Logical completeness
   1. Show that you can use only two-input NAND gates to implement each of the following two-input logic functions, and draw the used NAND gates and wiring.
      1. NOR function
      2. XOR function



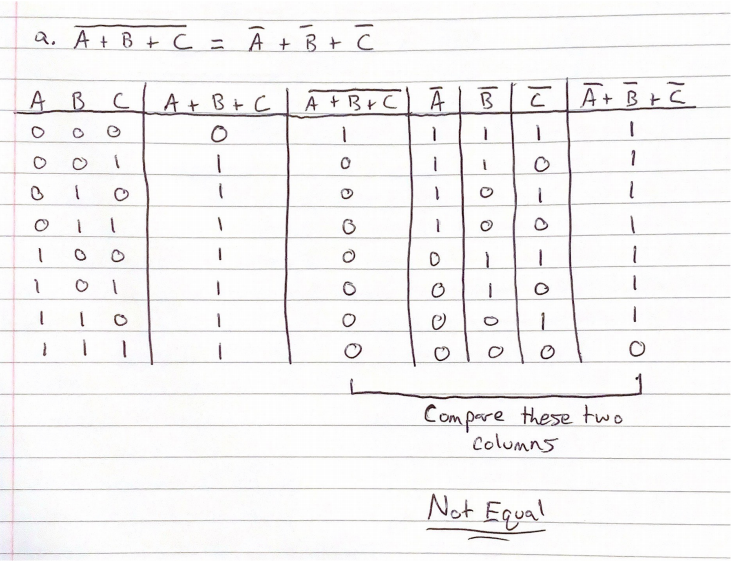
* 1. Show that you can use only two-input NOR gates to implement each of the following two-input logic functions, and draw the used NOR gates and wiring.
     1. NAND function
     2. XOR function

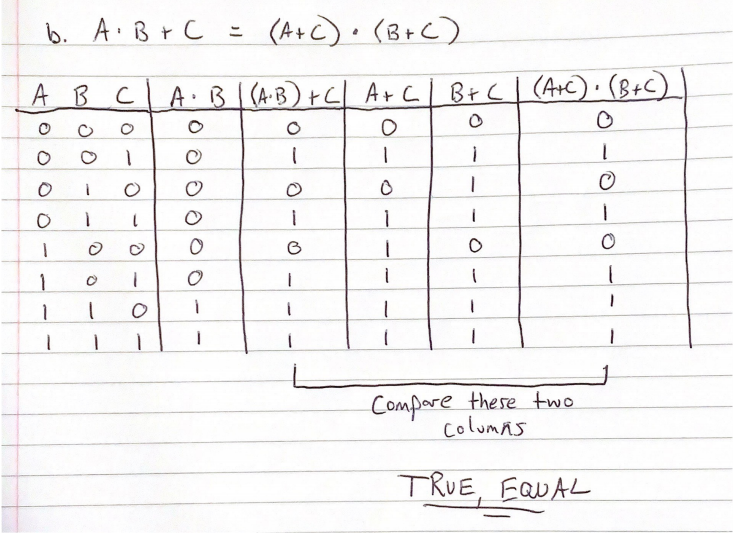


1. [10pts] Show how to use 2-1 Muxes to build a 4-1 Mux. Draw the used 2-1 Muxes and the wiring, and mark the 4 inputs and 1 output for the resulting 4-1 Mux.

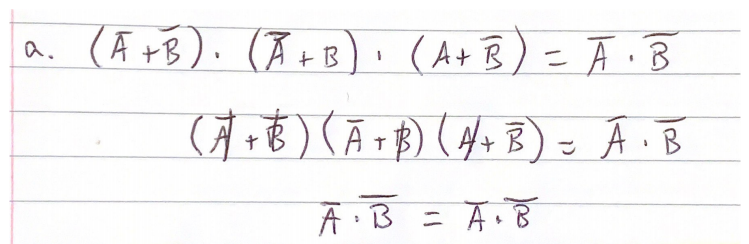


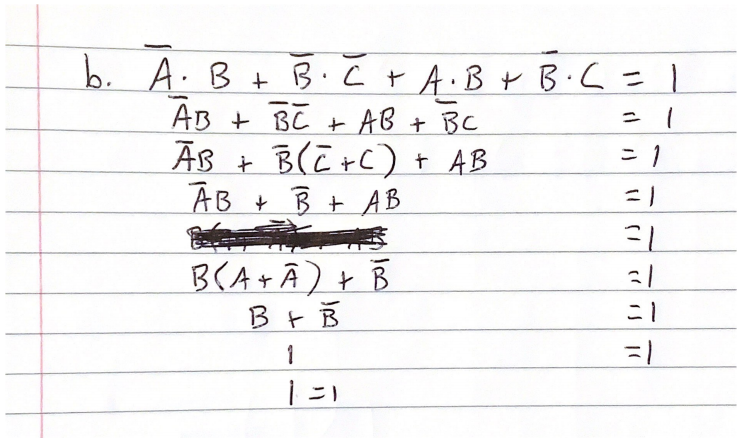
1. [10pts] Demonstrate by means of truth tables whether the following identities are valid or not:





1. [20pts] Prove the identity of each of the following Boolean equations, using algebraic manipulation:

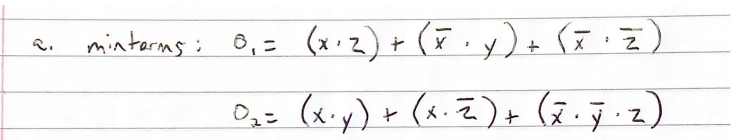




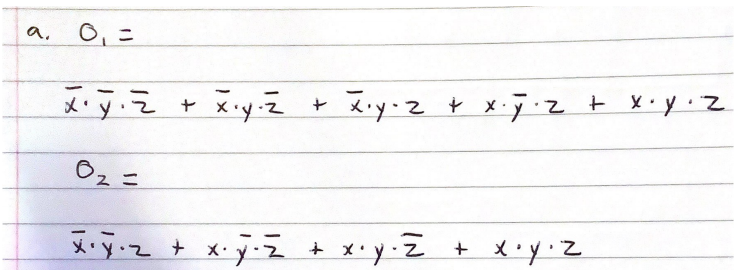
1. [20pts] For the Boolean function O1 and O2, as given in the following truth table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | | | **Output** | |
| **x** | **y** | **z** | **O1** | **O2** |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

* 1. List the minterms for a three-variable function with variables x, y, and z.



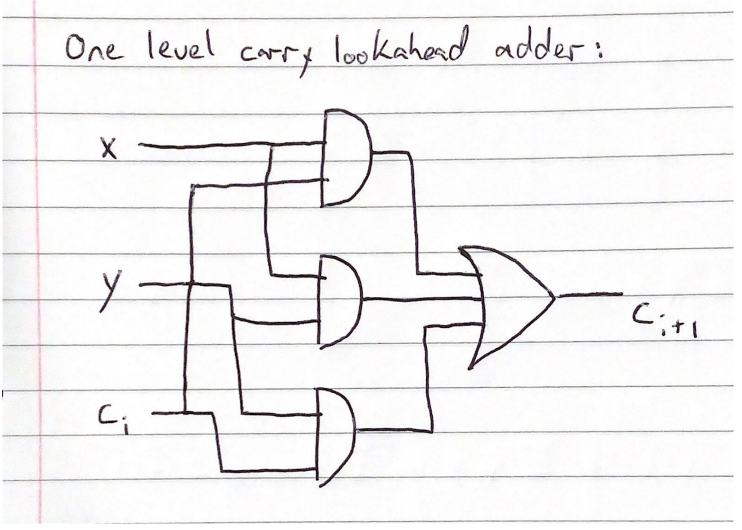
* 1. Express O1 and O2 in sum-of-product algebraic form.



1. [20pts] In class, we learned the implementation for a 4-bit carry lookahead adder. We can use the same idea and extend to build a 16-bit carry lookahead adder. Denote this implementation as a one-level carry lookahead adder.

In the textbook, Figure 8.6.3 shows a two-level implementation of a 16-bit carry lookahead adder. This adder uses 4-bit carry lookahead adders at the lower level, and uses a carry lookahead unit at the higher level.

Compare these two implementations and provide your explanation why the two-level implementation could be preferred.



A one-level carry lookahead adder can compute 4 bits across gates with n/4 blocks and a high number of gate delays. Information can take longer to process using just one level carry lookahead adders, but with the advanced computing speed when you move to a two-level carry lookahead adder that can implement with 16 bits, the speed increases tremendously. For example, a one-level carry lookahead adder for 4-bits would have a total of 6 gate delays, while a basic two-level adder processing 64 bits would have a total of 14 gate delays while processing much more information.