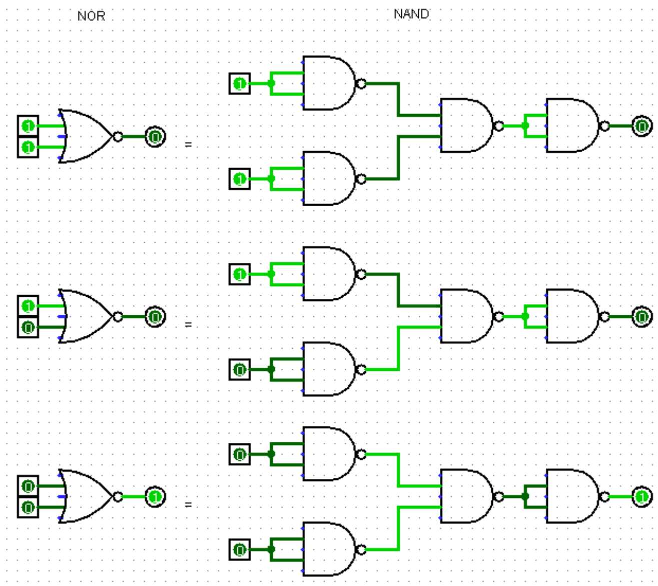
Eric Paulz
CPSC 3300-002

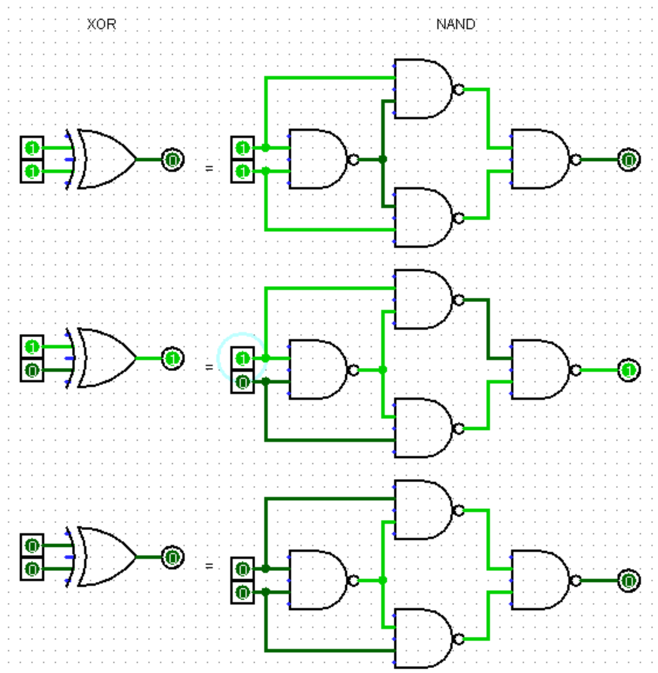
CPSC3300 – Computer Systems Organization
Homework #3 – Boolean Algebra and Adders
Due: 3:30 PM February 8
Submit to canvas

Total 100pts

1. [2pts] Logical completeness
   a. 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.
      i. NOR function



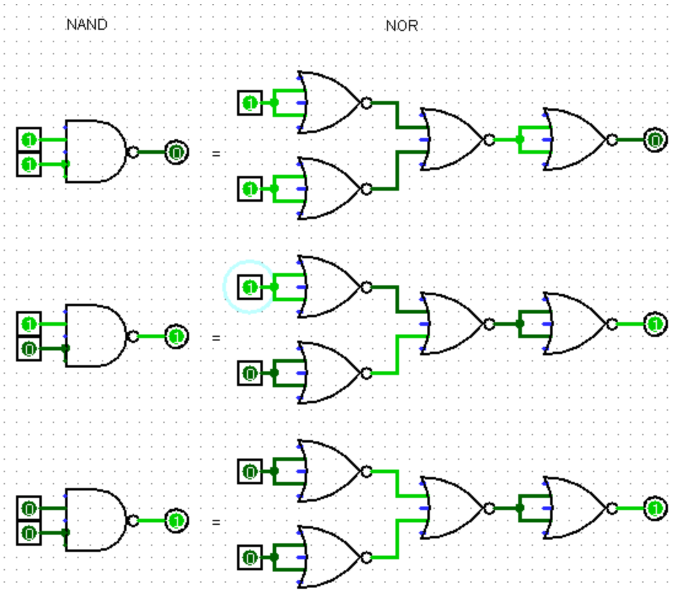
      ii. XOR function



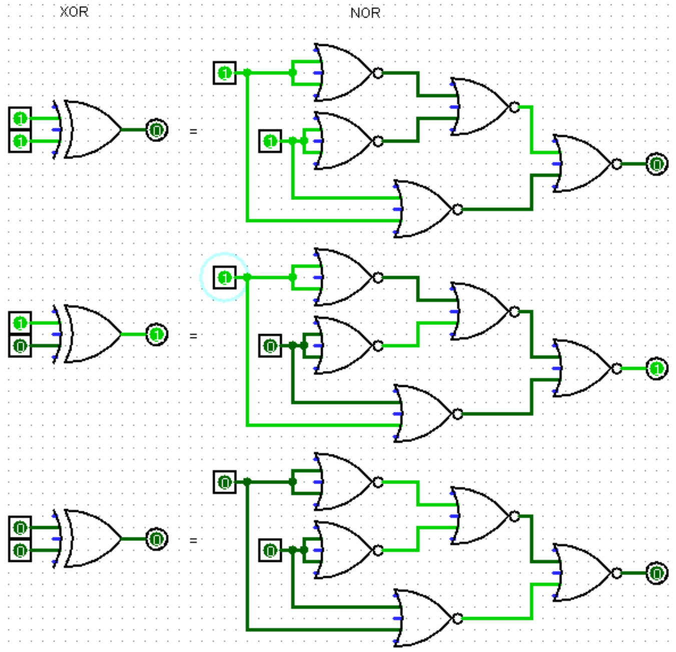
Eric Paulz
CPSC 3300-002

b. 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.

i. NAND function

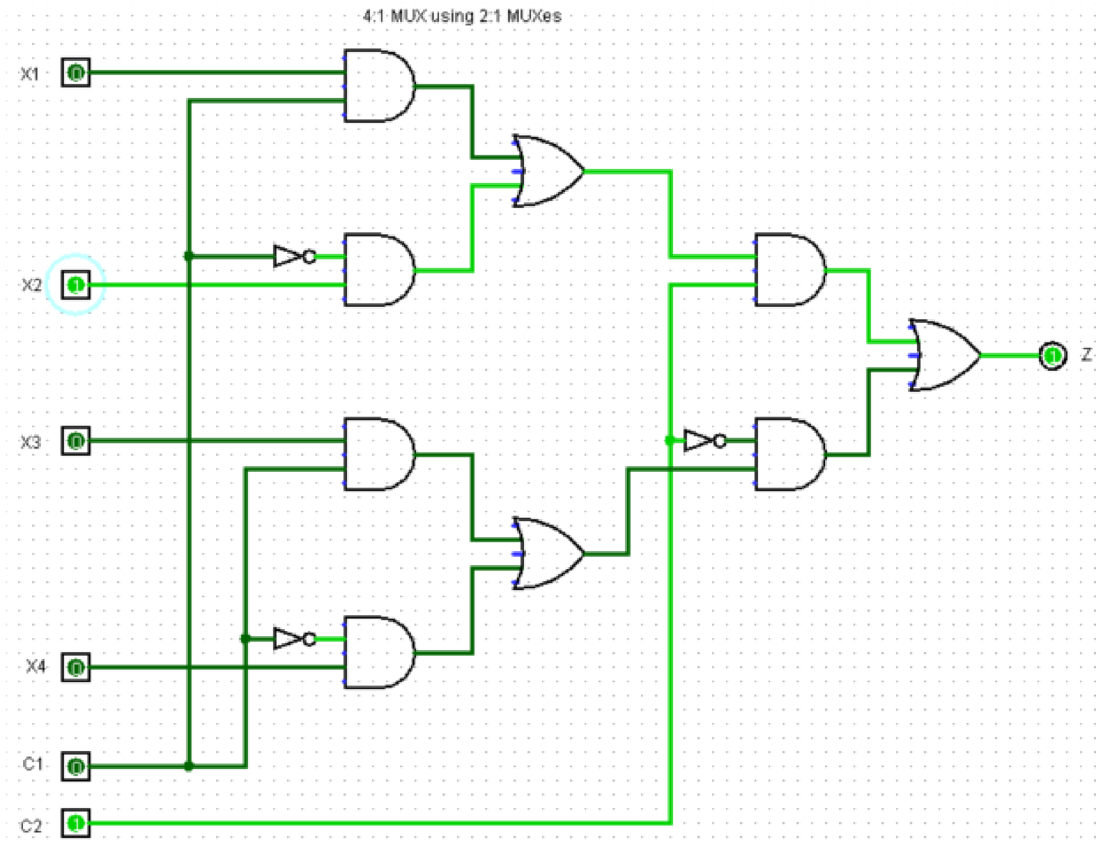


ii. XOR function



Eric Paulz
CPSC 3300-002

2. [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.



Eric Paulz
CPSC 3300-002

3. [10pts] Demonstrate by means of truth tables the validity of the following identities:
   a. $\overline{A \cdot B \cdot C} = \bar{A} \cdot \bar{B} \cdot \bar{C}$

A	B	C	$\overline{A \cdot B \cdot C}$	$\bar{A} \cdot \bar{B} \cdot \bar{C}$
0	0	0	1	1
0	0	1	1	0
0	1	0	1	0
1	0	0	1	0
0	1	1	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	0	0

False

   b. $A + B \cdot C = (A + B) \cdot (A + C)$

A	B	C	$A + B \cdot C$	$(A + B) \cdot (A + C)$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
1	0	0	1	1
0	1	1	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

True

4. [20pts] Prove the identity of each of the following Boolean equations, using algebraic manipulation:
   a. $\bar{A} \cdot \bar{B} + \bar{A} \cdot B + A \cdot B = \bar{A} + B$

$\bar{A}(\bar{B} + B) + A \cdot B = \bar{A} + B$ - distributive (T3)
$\bar{A}(1) + A \cdot B = \bar{A} + B$ - (T6)
$(\bar{A} + A) \cdot B = \bar{A} + B$ - commutative (T1)
$1 \cdot B = \bar{A} + B$ - (T6)
$B = B$

   b. $\bar{A} \cdot B + \bar{B} \cdot \bar{C} + A \cdot B + \bar{B} \cdot C = 1$

$\bar{A} \cdot B + A \cdot B + \bar{B} \cdot \bar{C} + \bar{B} \cdot C = 1$ - rearrange
$B(\bar{A} + A) + \bar{B}(\bar{C} + C) = 1$ - distributive law (T3)
$B(1) + \bar{B}(1) = 1$ - (T6)
$B + \bar{B} = 1$ - (T6)
$1 = 1$

Eric Paulz
CPSC 3300-002

5. [2pts] 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	1	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1

a. List the minterms for a three-variable function with variables x, y, and z.
   i. $\bar{x} \cdot \bar{y} \cdot \bar{z}$
   ii. $\bar{x} \cdot \bar{y} \cdot z$
   iii. $\bar{x} \cdot y \cdot \bar{z}$
   iv. $\bar{x} \cdot y \cdot z$
   v. $x \cdot \bar{y} \cdot \bar{z}$
   vi. $x \cdot \bar{y} \cdot z$
   vii. $x \cdot y \cdot \bar{z}$
   viii. $x \cdot y \cdot z$

b. Express O1 and O2 in sum-of-product algebraic form.
   i. O1 = $(\bar{x} \cdot \bar{y} \cdot \bar{z}) + (\bar{x} \cdot \bar{y} \cdot z) + (\bar{x} \cdot y \cdot z) + (x \cdot \bar{y} \cdot z) + (x \cdot y \cdot \bar{z})$
   ii. O2 = $(\bar{x} \cdot y \cdot \bar{z}) + (x \cdot \bar{y} \cdot \bar{z}) + (x \cdot y \cdot z)$

6. [20] 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 B.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.

The two-level implementation is generally more efficient because it requires the signals which anticipate a carry to travel through fewer gates. The reason that this is important is because by definition, a CLA "looks ahead" to see if a carry will be occurring, and the sooner this can be confirmed/denied, the faster the operation of the adder will be.