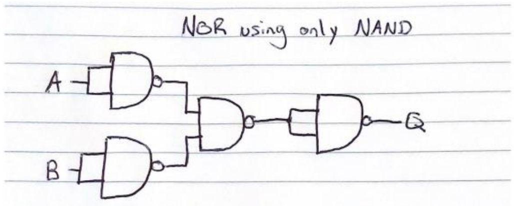
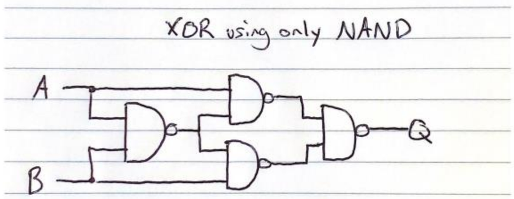
Patrick Woodrum - pwoodru

CPSC3300 – Computer Systems Organization
Homework #2 – Boolean Algebra and Adders
Due: 11:59PM Monday, February 8
Submit to Canvas

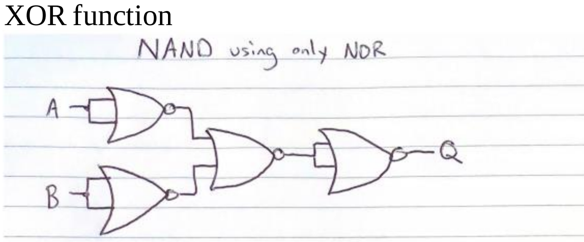
Total 100pts

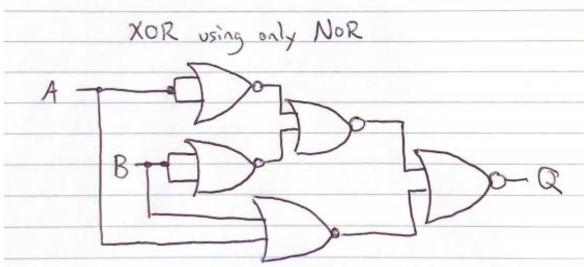
1. [20pts] Logical completeness
   a. Show that you can use only two-input NAND gates to implement each of the following two-input logic functions, and draw the used NAND gates and wiring.
      i. NOR function
      ii. XOR function





   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



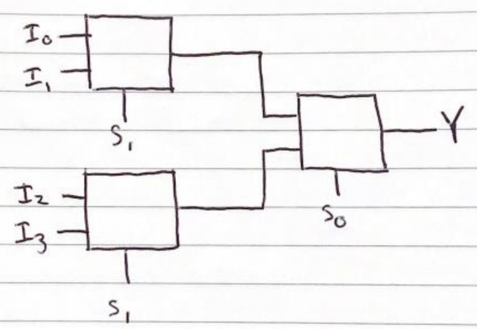


Patrick Woodrum - pwoodru

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.

S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Building 4-1 MUX using 2-1 MUXs



3. [10pts] Demonstrate by means of truth tables whether the following identities are valid or not:
   a. $\overline{A+B+C} = \bar{A} + \bar{B} + \bar{C}$

a. $\overline{A+B+C} = \bar{A} + \bar{B} + \bar{C}$

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

Compare these two columns

Not Equal

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

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

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

Compare these two columns

TRUE, EQUAL

Patrick Woodrum - pwoodru

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

a. $(\bar{A}+\bar{B})\cdot(\bar{A}+B)\cdot(A+\bar{B}) = \bar{A}\cdot\bar{B}$

$(\cancel{\bar{A}}+\cancel{\bar{B}})(\bar{A}+\cancel{B})(\cancel{A}+\bar{B}) = \bar{A}\cdot\bar{B}$

$\bar{A}\cdot\bar{B} = \bar{A}\cdot\bar{B}$

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

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

$\bar{A}B+\bar{B}\bar{C}+AB+\bar{B}C = 1$

$\bar{A}B+\bar{B}(\bar{C}+C)+AB = 1$

$\bar{A}B+\bar{B}+AB = 1$

~~[illegible]~~ $=1$

$B(A+\bar{A})+\bar{B} = 1$

$B+\bar{B} = 1$

$1 = 1$

$1 = 1$

5. [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

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

a. minterms: $O_1 = (x\cdot z)+(\bar{x}\cdot y)+(\bar{x}\cdot\bar{z})$

$O_2 = (x\cdot y)+(x\cdot\bar{z})+(\bar{x}\cdot\bar{y}\cdot z)$

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

$O_1 =$

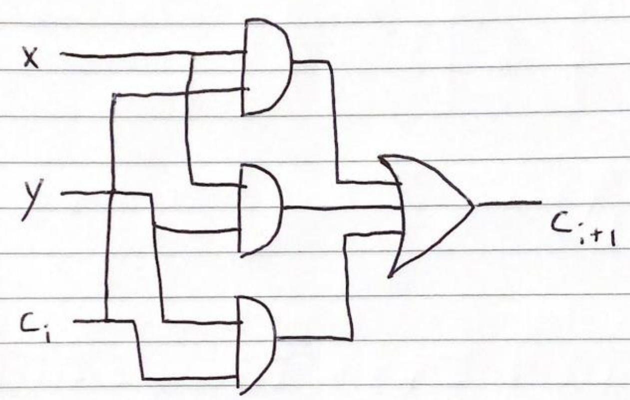
$\bar{x} \cdot \bar{y} \cdot \bar{z} + \bar{x} \cdot y \cdot \bar{z} + \bar{x} \cdot y \cdot z + x \cdot \bar{y} \cdot z + x \cdot y \cdot z$

$O_2 =$

$\bar{x} \cdot \bar{y} \cdot z + x \cdot \bar{y} \cdot \bar{z} + x \cdot y \cdot \bar{z} + x \cdot y \cdot z$

6. [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.

One level carry lookahead adder:



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.