Sitthiphol Yuwanaboon

John O’Neal

CS 130

23 February 2021

LAB #3

A.1 [10] <COD §A.2> In addition to the basic laws we discussed in this section, there are two important theorems, called DeMorgan's theorems:

$$\overline{A+B}=\overline{A}\cdot\overline{B} \text{ and } \overline{A\cdot B}=\overline{A}+\overline{B}$$

Prove DeMorgan's theorems with a truth table of the form

A	B	$\overline{A}$	$\overline{B}$	$\overline{A+B}$	$\overline{A}\cdot\overline{B}$	$\overline{A\cdot B}$	$\overline{A}+\overline{B}$
0	0	1	1	1	1	1	1
0	1	1	0	0	0	1	1
1	0	0	1	0	0	1	1
1	1	0	0	0	0	0	0

A.2 [15] <COD §A.2> Prove that the two equations for E are equivalent by using DeMorgan's theorems and the axioms shown COD Section A.2 (Gates, truth tables, and logical equations).

$$E=((A\cdot B)+(A\cdot C)+(B\cdot C))\cdot(\overline{A\cdot B\cdot C})$$

$$E=(A\cdot B\cdot\overline{C})+(A\cdot C\cdot\overline{B})+(B\cdot C\cdot\overline{A})$$

A.3 [10] <COD §A.2> Show that there are $2n$ entries in a truth table for a function with n inputs.

A.4 [10] <COD §A.2> One logic function that is used for a variety of purposes (including within adders and to compute parity) is *exclusive OR*. The output of a two-input exclusive OR function is true only if exactly one of the inputs is true. Show the truth table for a two-input exclusive OR function and implement this function using AND gates, OR gates, and inverters.

A.5 [15] <COD §A.2> Prove that the NOR gate is universal by showing how to build the AND, OR, and NOT functions using a two-input NOR gate.

A.6 [15] <COD §A.2> Prove that the NAND gate is universal by showing how to build the AND, OR, and NOT functions using a two-input NAND gate.

A.1 from definition A NAND gate is equivalent to an inversion followed by a and a Nor gate is equivalent to inversion followed by an AND

$$\overline{AB} \equiv \overline{A} + \overline{B}$$

$$\overline{A + B} \equiv \overline{A}\,\overline{B}$$

from Demorgan's theorem

from true table, we compare true value between these input and the output it is same through the gate. it is true that we have the same as the definition

(A2)

$$E = ((A.B + A.C + (B.C))(A.B.C)'$$

$$E = (A.B.C') + (A.C.B' + (B.C.A')$$

$$E(AB + AC + BC)(\overline{A} + \overline{B} + \overline{C})$$

$$E = (\overline{A}BC + A\overline{B}C + AB\overline{C})$$

AS above since $A\bar{A} = 0$

$$B.\bar{B} = 0$$

$$C.\bar{C} = 0$$

(A3)

the combination of input has value of true and false, nonetheless at How many input will produce 2^n power since n equal to number input

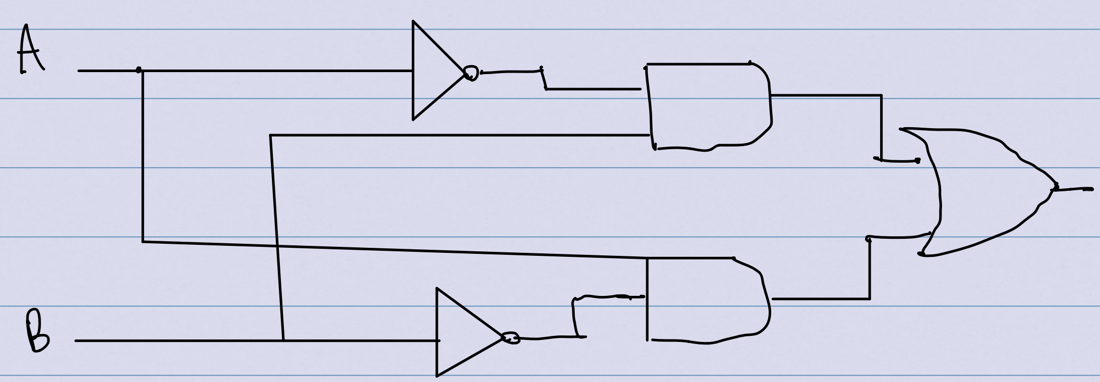
2 input = 4 row of true tab

(A.4)

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

equivalent logic is equal to

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

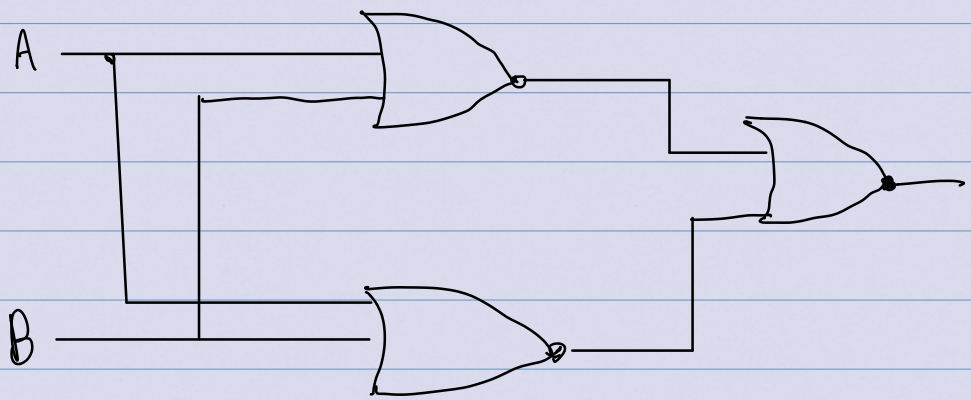


(A.5)

Notgate $\overline{A + A} = \bar{A}$

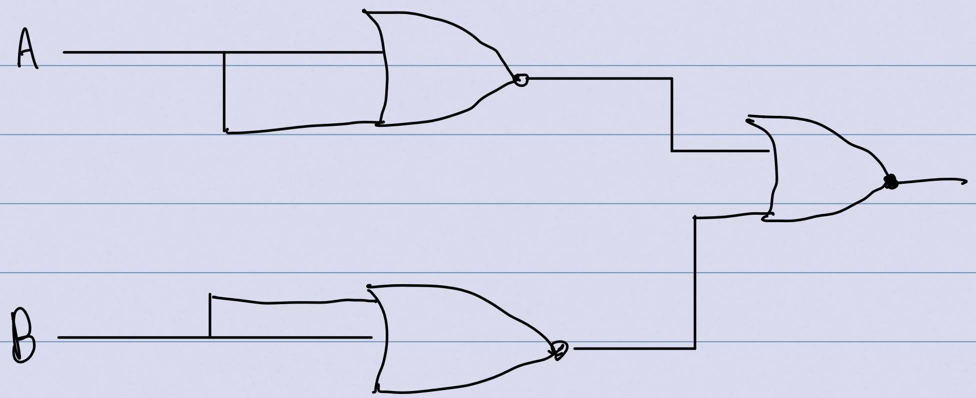
Or gate

$A + B = \overline{\overline{A + B}} = \overline{A \text{ NOR } B}$



AND gate $A \cdot B = \overline{\overline{A \cdot B}} = \overline{\bar{A} + \bar{B}}$

$\bar{A} \text{ NOR } \bar{B}$

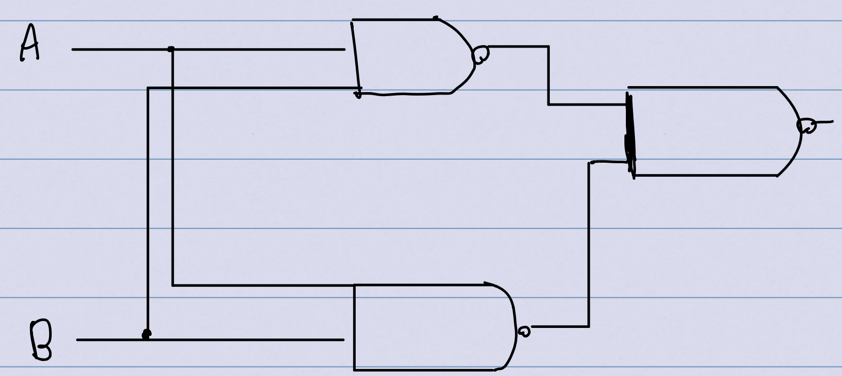


(A-b) Place NAND gate

NOT gate $= \overline{A \cdot A} = \overline{A}$

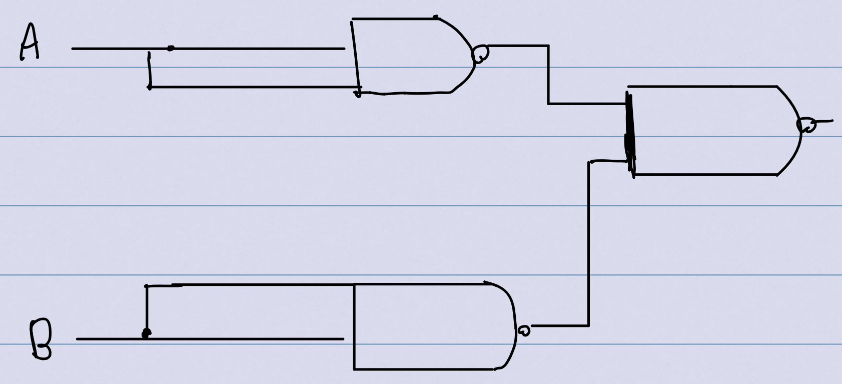
AND gate $A \cdot B = \overline{\overline{A \cdot B}} = \overline{A \text{ NAND } B}$

Therefore we must invert output of the NAND gate so we can use three NAND gates



OR gate

$A + B = \overline{\overline{A + B}} = \overline{\bar{A} \cdot \bar{B}} = \bar{A} \text{ NAND } \bar{B}$



Part A:

Methods to describe a combinational circuit

- True table
- Boolean algebraic expression
- Logic diagram

Ten properties of boolean algebra

- Commutative $x+y = y+x$, $x \cdot y = y \cdot x$
- Associative $(x+y)+z = x+(y+z)$, $(x \cdot y) \cdot z = x \cdot ($
- Distributive $x+(y \cdot z) = (x+y) \cdot (x+z)$, $x \cdot (y+z) = x \cdot y + x \cdot z$
- Identity $x+0 = x$, $x \cdot 1 = x$
- Complement $x+(x') = 1$, $x \cdot (x') = 0$

Precedence	Operator
Highest	Complement
	AND
Lowest	OR

Duality

Exchange + and ·

Exchange 1 and 0

Zero theorem

$x+1 = 1$

$x \cdot 0 = 0$

De Morgan's law

$(a \cdot b)' = a' + b'$

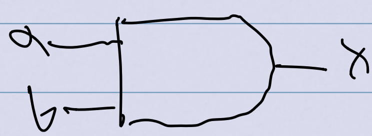
$(a+b)' = a' \cdot b'$

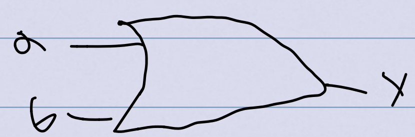
complement theorems

$(x')' = x$

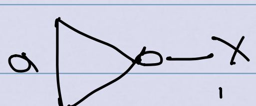
$1' = 0$

$0' = 1$



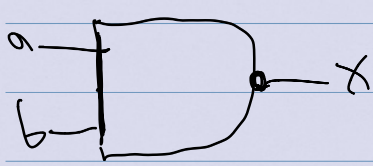


$x = a \cdot b$ AND Gate $x = a+b$ OR gate



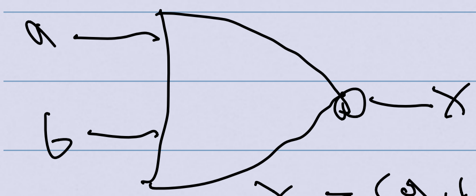
Inverter

$x = a'$



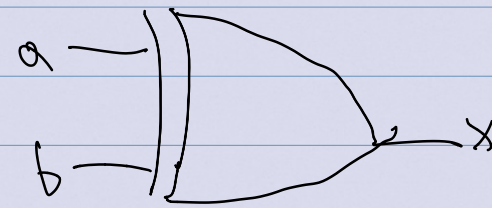
$x = (a \cdot b)'$

NAND Gate



$x = (a+b)'$

NOR GATE



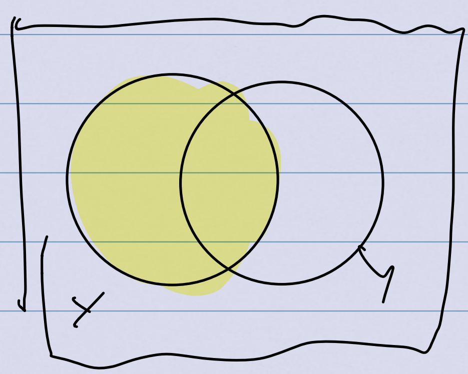
$x = a \oplus b$

XOR GATE

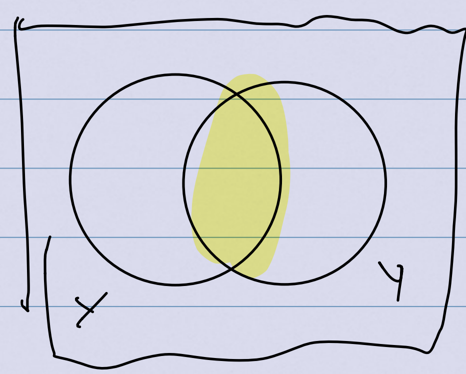
OR Gate is set union

AND Gate is set intersection

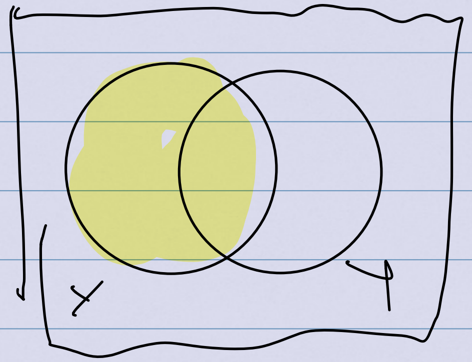
INVERTER is set of complement



x



$x \cdot y$



$$x + x'y$$