

# **Boolean algebra. Logic operations. Formula and their conversion.**

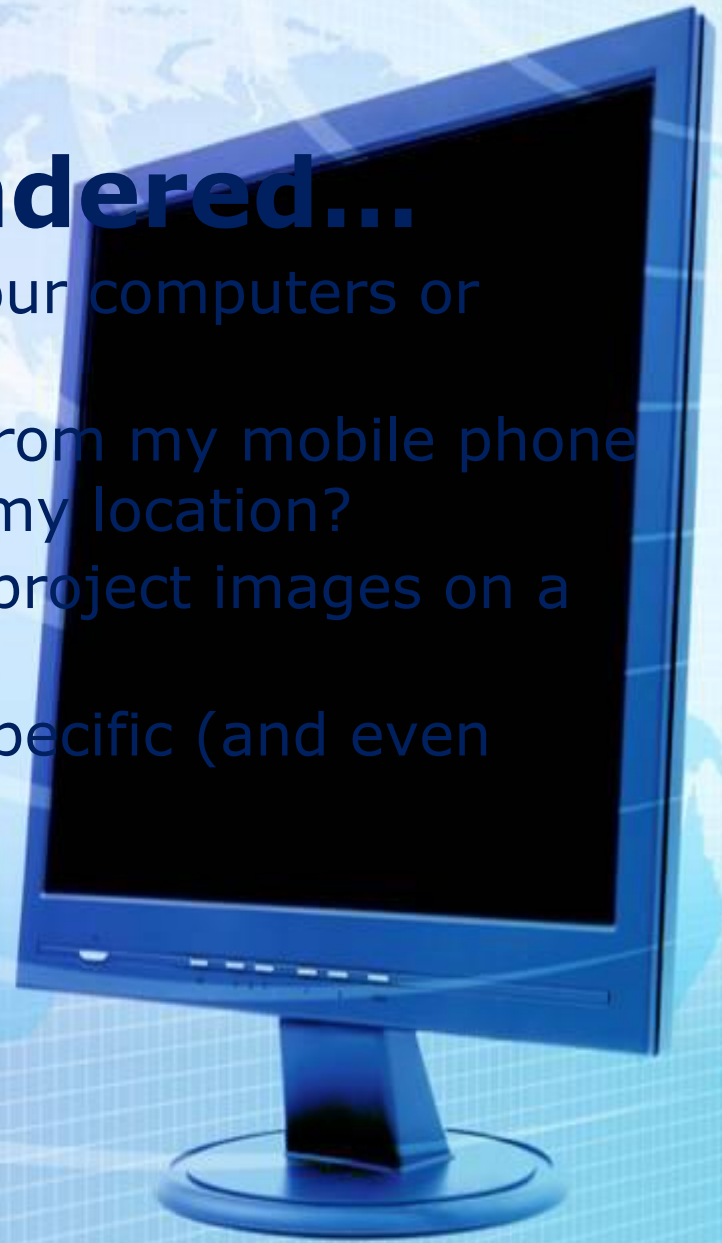
Author: Snassapin Temirlan  
T-201.

# CONTENTS

- Introduction to Boolean Algebra
- Basic Definitions and Axioms in Boolean Algebra
- Basic Theorems
- Product-of-sums and Sum-of-products
- Minimal Boolean Expressions and Prime Implicants
- Applications and other means of simplification:
  - Logic gate and circuits
  - Truth tables and Boolean functions
  - Karnaugh map (K-map)

# Have you ever wondered...

- How can we communicate with our computers or laptops?
- How is it possible that my SMS from my mobile phone be sent hundreds of miles from my location?
- How does televisions be able to project images on a screen?
- Why does robots be able to do specific (and even complicated) tasks?



# An introduction

- A statement is true if it agrees with reality, false if it doesn't.
- Two-state logic assumes that each statement is either true or false.
- The Greeks, especially Aristotle, worked out the theory of two-state logic in great detail.
- In 1854, George Boole came up with symbolic logic, better known as the Boolean Algebra. Boolean algebra uses letters and symbols to represent statements and their logical connections.
- Each variable in Boolean algebra has either of two values: true or false. (this is why it is called a two-state or binary algebra)
- Boolean algebra was a far-out subject until 1938, when Claude Shannon used it to analyze and design telephone switching circuits.
- "He let the variables represents closed and open relays.
- Boolean algebra has become one of the major design tools of digital and computer electronics

# When to use Boolean Algebra?

- At least one (1) or more inputs of either logic 1 (true) or logic 0 (false) and a single desired output (either a 1 or a 0, depending on the inputs)
- Examples:
  - $F = a + b$
  - $F = a * b$
  - $F = (a + b) * c'$
  - $F = abc' + (bd)' + ab + a'cd$
- Note that inputs a, b, c, and d should have a value either a logic 1 or logic 0 and the output F should acquire a value either 1 and 0.

# Axioms in Boolean Algebra

- $a+b = b+a$
- $a*b = b*a$

COMMUTATIVE  
LAW

- $a+0 = a$
- $a*1 = a$

IDENTITY LAW

DISTRIBUTIVE  
LAW

- $a+(b*c) = (a+b)*(a+c)$
- $a*(b+c) = (a*b)+(a*c)$

COMPLEMENT  
LAW

- $a+a' = 1$
- $a*a' = 0$



# What is the output if we have \*?

Let  $a$  be the values on the column side and  $b$  be the values on the row side.

*	0	1
0	0	0
1	0	1

# What is the result of the unary operation?

Simply change the input 0 to an output of 1 and the input 1 to an output 0!!

\	0	1
0	1	0
1	0	1



# Other special theorems

Let  $a$  be any element of a Boolean algebra  $B$ .

Theorem 6: Uniqueness of compliment:

If  $a+x = 1$  and  $a*x = 0$ , then  $x = a'$

Theorem 7: Involution Law:

$$(a')' = a$$

Theorem 8: \*Inversion Law:

$$0' = 1$$

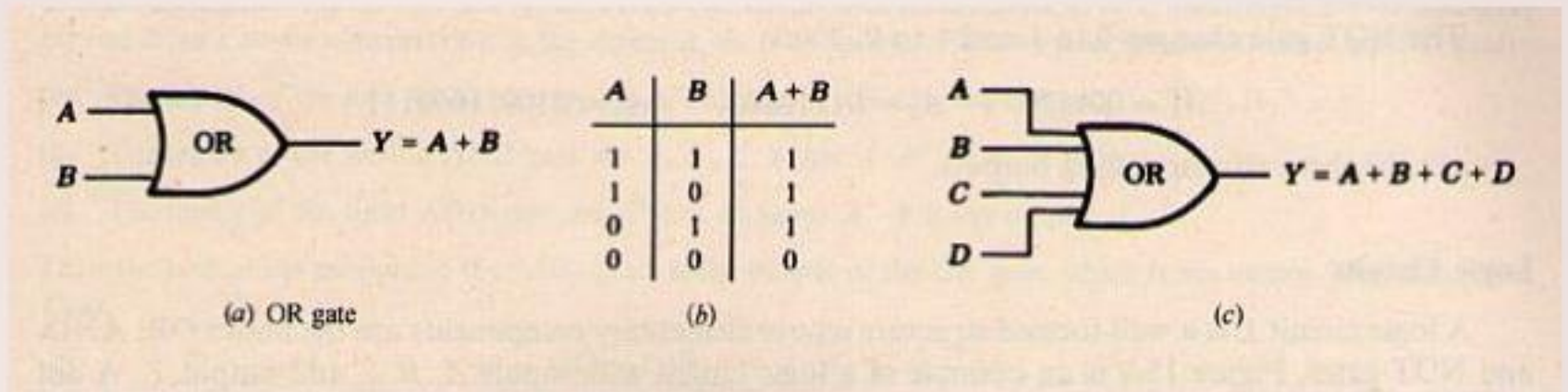
$$1' = 0$$

Theorem 9: DeMorgan's Laws:

$$(a+b)' = a' * b'$$

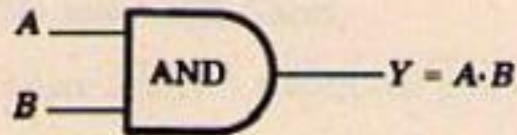
$$(a*b)' = a' + b'$$

# OR Gate



If at least one of the inputs has a value of logic high (1), the output is logic high (1). Else, the output is logic low (0)

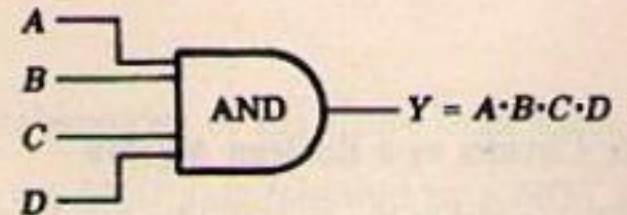
# AND Gate



(a) AND gate

A	B	A · B
1	1	1
1	0	0
0	1	0
0	0	0

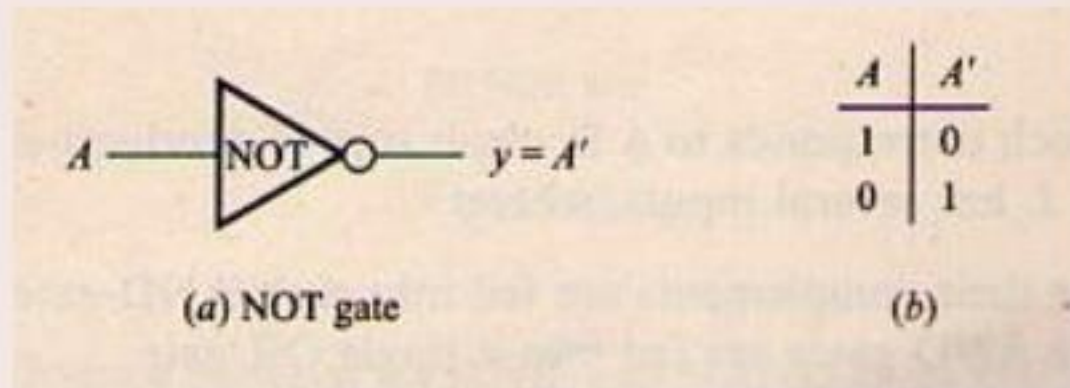
(b)



(c)

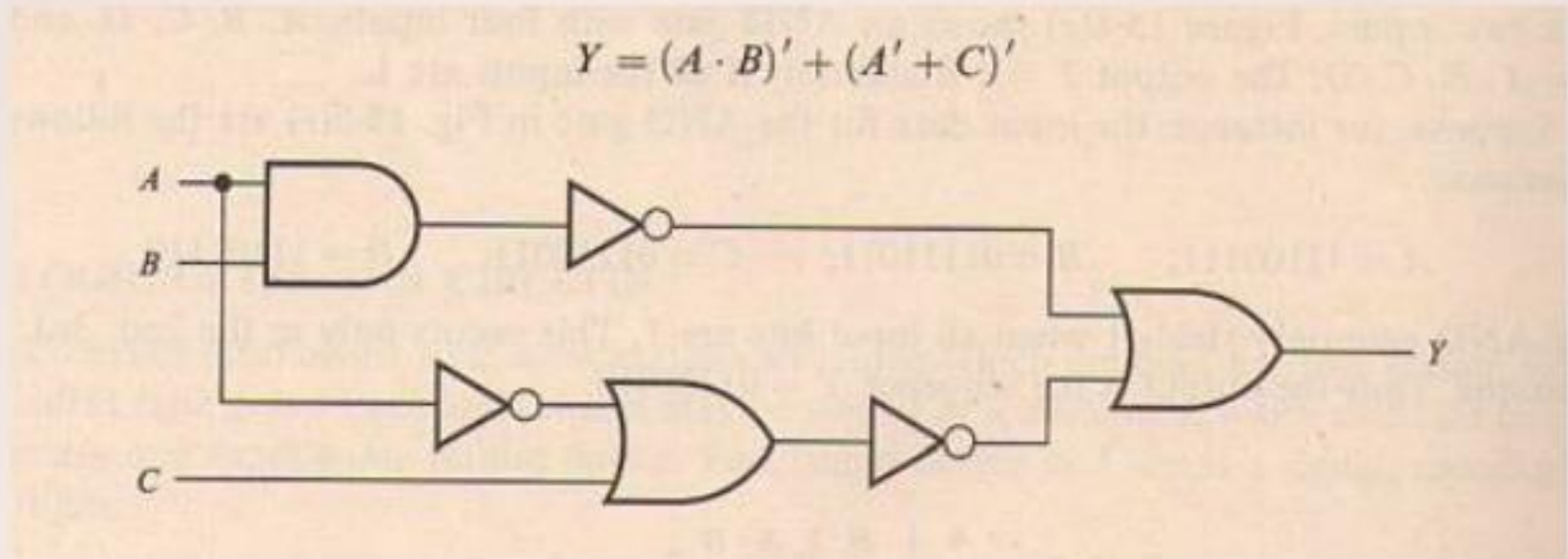
If all of the inputs has a value of logic high (1) , the output is logic high (1). Else, the output is logic low (0)

# NOT Gate



If the input is logic high (1), the output is logic low (0). And, vice versa.

# Logic Circuit



Integration of AND, OR, and NOT gates.

# Let's try this one...

Problem: Design a three input minimal AND-OR circuit L with the following table:

$$T = [A, B, C; L] = [00001111, 00110011, 01010101; 11001101]$$