Computer System Architecture
COMP201TH
Lecture-2
Boolean Algebra

"The general laws of Nature are not, for the most part, immediate objects of perception….They are in all cases, and in the strictest sense of the term, probable conclusions."

…George Boole

Introduction:

George Boole, a 19th century English Mathematician, developed a system of logical algebra by which reasoning can be expressed mathematically. In 1854, Boole published a classic book, "An Investigation of the Laws of thought" on which he founded the Mathematical theories of Logic and Probabilities.

Boole's system of logical algebra, now called Boolean algebra, was investigated a tool for analyzing and designing relay switching circuits by Claude E. Shannon at the Massachusetts Institute of Technology in 1938. Shannon, a research assistant in the Electrical Engineering Department, wrote a thesis entitled "A symbolic analysis of Relay and Switching Circuits". As a result of his work, Boolean algebra is now, used extensively in the analysis and design of logical circuits.

Today, Boolean algebra is the backbone of computer circuit analysis.

- Boolean Algebra:
  - Boolean algebra is a form of algebra that deals with single digit binary values and variables.
  - Boolean algebra has only two mathematical operations:
    - Addition
    - Multiplication
      - These operations are associated with the OR gate and the AND gate, respectively.
  - The purpose of Boolean algebra is to facilitate the analysis and design of digital circuits. It provides a convenient tool to:
    - Express in algebraic form a truth table relationship between binary variables.
    - Express in algebraic form the input-output relationship of logic diagrams.
    - Find simpler circuits for the same function.

• Logic Symbols that you might see in Boolean expressions:

Basic logic symbols

Symbol	Name	Read as
¬ ˜ !	negation	not
∧ · &	**logical** conjunction	and
∨ + ∥	**logical** (inclusive) disjunction	or
⊕ ⊻	exclusive disjunction	xor

• Basic identities in Boolean Algebra:

- **Annulment Law:**
  A variable ANDed with 0 gives 0, while a variable ORed with 1 gives 1 i.e.

$$X + 1 = 1 \qquad X \cdot 0 = 0$$

- **Identity Law:**
  In this law, variable remains unchanged if its ORed with 0 or ANDed with 1 i.e.

$$X + 0 = X \qquad X \cdot 1 = X$$

Identity

- **Idempotent Law:**
  A variable remain unchanged when it is ORed or ANDed with itself i.e.

$$X + X = X \qquad X \cdot X = X$$

Idempotent Law

- **Complement Law:**
  If a complement is added (ORed) to a variable it gives one, if a variable is multiplied (ANDed) with its complement it results in 0 i.e.

$$X + X' = 1 \qquad X \cdot X' = 0$$

Complement

- **Involution Law:**
  A variable with two negation symbol gets cancelled out and original variable is obtained i.e.

$(X')' = X$ Involution Law

- **Commutative Law:**

A variable order does not matter in this law i.e.

$X + Y = Y + X$ $XY = YX$ Commutativity

- **Associative Law:**

The order of operation does not matter if the priority of variables is same.

$X + (Y + Z) = (X + Y) + Z$ $X(YZ) = (XY)Z$ Associativity

- **Distributive Law:**

This law governs opening up of brackets i.e.

$X(Y + Z) = XY + XZ$ $X + YZ = (X + Y)(X + Z)$ Distributivity

- **Absorption Law:**

This law involves absorbing the similar variables i.e.

$X + XY = X$ $X(X + Y) = X$ Absorption Law

- **Simplification Law:**

$X + X'Y = X + Y$ $X(X' + Y) = XY$ Simplification

X+X\`Y = (X+X\`) (X+Y) (Distributive law X+YZ = (X+Y) (X+Z)

= 1. (X+Y) (X+X\` =1)

= X+Y

- **DeMorgan's Law:**

$(X + Y)' = X'Y'$ $(XY)' = X' + Y'$ DeMorgan's Law

$X + 0 = X$	$X \cdot 1 = X$	Identity
$X + 1 = 1$	$X \cdot 0 = 0$	
$X + X = X$	$X \cdot X = X$	Idempotent Law
$X + X' = 1$	$X \cdot X' = 0$	Complement
$(X')' = X$		Involution Law
$X + Y = Y + X$	$XY = YX$	Commutativity
$X + (Y + Z) = (X + Y) + Z$	$X(YZ) = (XY)Z$	Associativity
$X(Y + Z) = XY + XZ$	$X + YZ = (X + Y)(X + Z)$	Distributivity
$X + XY = X$	$X(X + Y) = X$	Absorption Law
$X + X'Y = X + Y$	$X(X' + Y) = XY$	Simplification
$(X + Y)' = X'Y'$	$(XY)' = X' + Y'$	DeMorgan's Law

- E.g. (X+Y) (X+Z) = X+YZ

  This rule can be proved as follows:

  (X+Y) (X+Z) = XX +XZ +XY+YZ (distributive law (X(Y+Z) = XY+XZ))
  = X+XZ+XY+YZ (X.X=X)
  = X(1+Z) +XY+YZ (X+1=1)
  = X.1+XY+YZ
  = X(1+Y)+YZ (1+Y=1)
  = X+YZ

DeMorgan's Theorems:

- **DeMorgan's first theorem is:**

The complement of a product of variables is equal to the sum of the complements of the variables,

Stated another way,

The complement of two or more ANDed variables is equivalent to the OR of the complements of the individual variables.

The formula for expressing this theorem for two variables is

$$\overline{XY} = \overline{X} + \overline{Y}$$

- **DeMorgan's Second theorem is:**

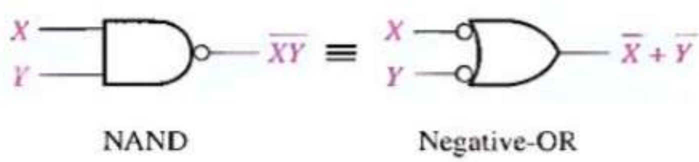
The complement of a sum of variables is equal to the product of the complements of the variables.

Stated another way,

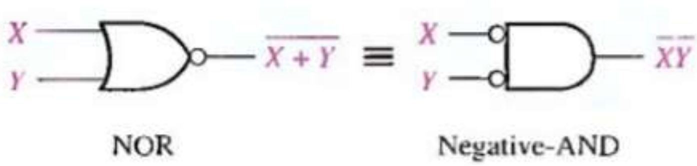
The complement of two or more ORed variables is equivalent to the AND of the complements of the individual variables,

The formula for expressing this theorem for two variables is

$$\overline{X + Y} = \overline{X}\,\overline{Y}$$



Inputs		Output	
X	Y	$\overline{XY}$	$\overline{X} + \overline{Y}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0



Inputs		Output	
X	Y	$\overline{X + Y}$	$\overline{X}\,\overline{Y}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

Fig. Gate equivalencies and the corresponding truth tables that illustrate DeMorgan's theorems and that NOR and NAND gates have two distinct graphic symbols.

Examples:

1) Simplify y = A+ A`. B

= A. (B+1) + A`.B	(Annulment Law: 1+B =1)
= A.B + A + A`.B	(Distributive Law)
= A.B + A`.B + A	(Commutative Law)
= (A+A`)B + A	(Distributive Law)
= B+A	

2) Simplify y = A` + B + A . B . C`

= A` (BC` +1) +B` + ABC`	(X+1 = 1 annulment law)
= A`BC` + A` + B` +ABC`	(distributive law)
=(A`+A) BC` + A` + B`	(Commutative and distributive law)
= BC` + A` + B`	

3) Simplify y = A`B`C` +A`BC` +A`BC+ AB`C`+ABC`+ABC

= A`C`(B`+B) + AC`((B`+B) + BC (A`+A)	
=A`C`+AC`+BC	
= (A`+A)C` +BC	
= C` + BC	
= (B+C`) (C+C`)	(distributive law)
= B+C`	