

Digital Systems

Prepared by Dr. Alaa Eldeen Sayed Ahmed

IT131
Boolean Algebra Part I

Lecture 2



Boolean Algebra

- Binary Logic and Gates
- Why study Boolean Algebra?
- Boolean Algebra
- Simplifications

Boolean Algebra

Binary Logic and Logic Gates

- Our objective is to learn how to design digital circuits.
- These circuits use binary systems and Deals with binary variables and with operations that assume logical meaning.
- Manipulation of binary information is done using circuits called ***logic gates***.

Boolean Algebra

Logic Gates

- The building blocks used to create digital circuits are logic gates.
- There are three **elementary** logic gates and a range of other **simple** gates.
- Each gate has its own **logic symbol** which allows complex functions to be represented by logic diagrams.
- The function of each gate can be represented by a **truth table** or using **Boolean function**.

Boolean Algebra

- The table used to represent the Boolean expression of a logic gate function is commonly called a **Truth Table**.
- A logic gate truth table shows each possible input combination to the gate or circuit with the resultant output depending upon the combination of these input(s).

Boolean Algebra

- For example, consider a single 2-input logic circuit with input variables labelled as A and B.
- There are “four” possible input combinations or 2^2 of “0” and “1” for the two inputs.

A	B
0	0
0	1
1	0
1	1

Boolean Algebra

- a 3-input logic circuit would have 8 possible input combinations or 2^3 and a 4-input logic circuit would have 16 or 2^4 , and so on as the number of inputs increases.

A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Boolean Algebra

AND Gate

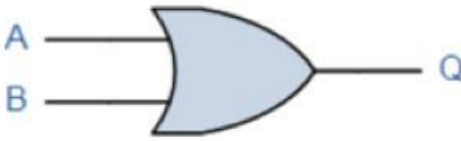
- Boolean Algebra Truth Tables For A 2-input AND Gate
- Note that the Boolean Expression for a two input AND gate can be written as: $A.B$ or just simply AB without the decimal point.

Symbol	Truth Table		
 2-input AND Gate	A	B	Q
	0	0	0
	0	1	0
	1	0	0
	1	1	1
Boolean Expression $Q = A.B$	Read as A AND B gives Q		

Boolean Algebra

OR Gate

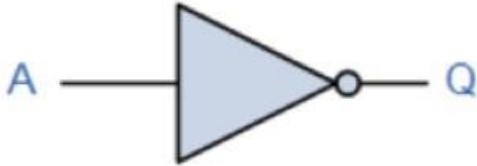
- Boolean Algebra Truth Tables For A 2-input OR Gate
- Note that the Boolean Expression for a two input OR gate can be written as: $A+B$.

Symbol	Truth Table		
 2-input OR Gate	A	B	Q
	0	0	0
	0	1	1
	1	0	1
	1	1	1
Boolean Expression $Q = A+B$	Read as A OR B gives Q		

Boolean Algebra

Inverter (Not Gate)

- For a single input NOT (Inverter) gate, the output Q is ONLY true when the input is “NOT” true, the output is the inverse or complement of the input giving the Boolean Expression of: ($Q = \text{NOT } A$).

Symbol	Truth Table	
 Inverter or NOT Gate	A	Q
	0	1
	1	0
Boolean Expression $Q = \text{NOT } A$ or $\bar{A}$	Read as inversion of A gives Q	

Boolean Algebra

NAND (Not AND) Gate

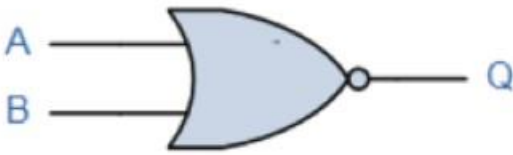
- For a 2-input NAND gate, the output Q is NOT true if BOTH input A and input B are true, giving the Boolean Expression of: ($Q = \text{not}(A \text{ AND } B)$).

Symbol	Truth Table		
 2-input NAND Gate	A	B	Q
	0	0	1
	0	1	1
	1	0	1
	1	1	0
Boolean Expression $Q = \overline{A \cdot B}$	Read as A AND B gives NOT-Q		

Boolean Algebra

NOR (Not OR) Gate

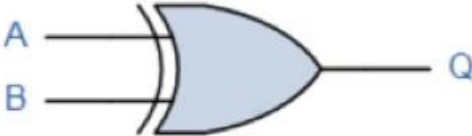
- For a 2-input NOR gate, the output Q is true if BOTH input A and input B are NOT true, giving the Boolean Expression of: ($Q = \text{not}(A \text{ OR } B)$).

Symbol	Truth Table		
 2-input NOR Gate	A	B	Q
	0	0	1
	0	1	0
	1	0	0
	1	1	0
Boolean Expression $Q = \overline{A+B}$	Read as A OR B gives NOT-Q		

Boolean Algebra

XOR (Exclusive OR) Gate

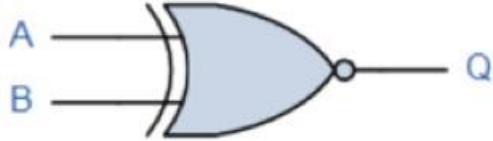
- For a 2-input Ex-OR gate, the output Q is true if EITHER input A or if input B is true, but NOT both giving the Boolean Expression of: ($Q = (A \text{ and NOT } B) \text{ or } (\text{NOT } A \text{ and } B)$).

Symbol	Truth Table		
 2-input Ex-OR Gate	A	B	Q
	0	0	0
	0	1	1
	1	0	1
	1	1	0
Boolean Expression $Q = A \oplus B$			

Boolean Algebra

XNOR (Exclusive NOR) Gate

- For a 2-input Ex-NOR gate, the output Q is true if BOTH input A and input B are the same, either true or false, giving the Boolean Expression of: ($Q = (A \text{ and } B) \text{ or } (\text{NOT } A \text{ and } \text{NOT } B)$).

Symbol	Truth Table		
 2-input Ex-NOR Gate	A	B	Q
	0	0	1
	0	1	0
	1	0	0
	1	1	1
Boolean Expression $Q = \overline{A \oplus B}$			

Boolean Algebra

- Boolean Algebra is a system of mathematics based on logic that has its own set of rules or laws which are used to:
 - define Boolean expressions.
 - reduce Boolean expressions.
 - analyse and simplify the digital (logic) circuits.
- The variables used in Boolean Algebra have only one of two possible values, a logic “0” and a logic “1”

Boolean Algebra

- Boolean Algebra is made up of
 1. Elements - variables or constants with value 1 or 0.
 2. Operators - AND, OR and NOT
 3. Axioms and Theorems.
- There are three ways to represent a Boolean Expression/Function.
 1. Algebraic Equation
 2. Truth Table
 3. Circuit Diagram

Boolean Algebra

- **Boolean function:** expression consists of binary variables, constants (0/1), and logic operations.
It can be equal 1 or 0.
- The output of any function can be equal to 0 or 1.
- **Example:** $F = x + y' z$;
- **Question:** When $F=1$ and $F=0$?

Boolean Algebra

- It can be presented by Truth Table, where number of rows is equal to 2^n where n is number of variables.
- The combinations are binary numbers from 0 to 2^n-1 .
- **Boolean variable:** a variable x is a Boolean variable if value of $x \in B$
- **Literal:** a variable in complemented or uncomplemented forms.

Boolean Algebra

- **Example:** $F = x + y' z$;

x	y	z	y'	y'.z	x+ y'.z
0	0	0	1	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	1	0	0	1

Boolean Algebra

Expression Simplification

- By manipulating a Boolean expression according to Boolean algebraic rules, it is often possible to obtain a simpler expression for the same function. This simpler expression reduces both the number of gates in the circuit and the numbers of inputs to the gates.
- To see how this is done, we must first study the basic rules of Boolean algebra.

Boolean Algebra

Basic identities of Boolean algebra

- T1 : Properties of 0
 - (a) $0 + X = X$
 - (b) $0.X = 0$
- T2 : Properties of 1
 - (a) $1 + X = 1$
 - (b) $1.X = X$

$$0 + X = X$$

0	X	R
0	0	0
0	1	1

$$0 . X = 0$$

0	X	R
0	0	0
0	1	0

$$1 + X = 1$$

1	X	R
1	0	1
1	1	1

$$1 . X = X$$

1	X	R
1	0	0
1	1	1

Boolean Algebra

Basic identities of Boolean algebra

- T3 : Commutative Law
 - (a) $X + Y = Y + X$
 - (b) $X.Y = Y.X$

X	Y	$X+Y$	$Y+X$
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	1

X	Y	$X.Y$	$Y.X$
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

Boolean Algebra

Basic identities of Boolean algebra

- T4 : Associative Law
- (a) $(X.Y).Z = X.(Y.Z)$

(b) $(X + Y) + Z = X + (Y + Z)$

X	Y	Z	Y . Z	X . Y	X . (Y . Z)	(X . Y) . Z
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	1	0	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	1	0	0
1	1	1	1	1	1	1

X	Y	Z	Y + Z	X + Y	X + (Y + Z)	(X + Y) + Z
0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	0	1	1	1
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

Boolean Algebra

Basic identities of Boolean algebra

- T5 : Distributive Law
- (a) $X(Y + Z) = X.Y + X.Z$

(b) $X + (Y.Z) = (X + Y)(X + Z)$

X	Y	Z	Y+Z	X.Y	X.Z	X.(Y+Z)	X.Y+X.Z
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	0	1	1	1
1	1	0	1	1	0	1	1
1	1	1	1	1	1	1	1

X	Y	Z	Y.Z	X+Y	X+Z	X+Y.Z	(X+Y).(X+Z)
0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0
0	1	0	0	1	0	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

Boolean Algebra

Basic identities of Boolean algebra

- T6 : Idempotence Law

- (a) $X + X = X$

- (b) $X.X = X$

X	X	R
0	0	0
1	1	1

X	X	R
0	0	0
1	1	1

Boolean Algebra

Basic identities of Boolean algebra

- T7 : Absorption (Redundance) Law
- (a) $X + X.Y = X$ (b) $X.(X + Y) = X$

X	Y	X.Y	X+X.Y
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

X	Y	X + Y	X.(X+Y)
0	0	0	0
0	1	1	0
1	0	1	1
1	1	1	1

- $$\begin{aligned} X + X.Y &= X.(1+Y) \\ &= X.(1) \\ &= X \end{aligned}$$

$$\begin{aligned} &X.(X+Y) \\ &X.X + X.Y = \\ &X + X.Y = X \end{aligned}$$

Boolean Algebra

Basic identities of Boolean algebra

- T8 : Complementary Law
 - (a) $X + X' = 1$

X	X'	R
0	1	1
1	0	1

(b) $X \cdot X' = 0$

X	X'	R
0	1	0
1	0	0

Boolean Algebra

Basic identities of Boolean algebra

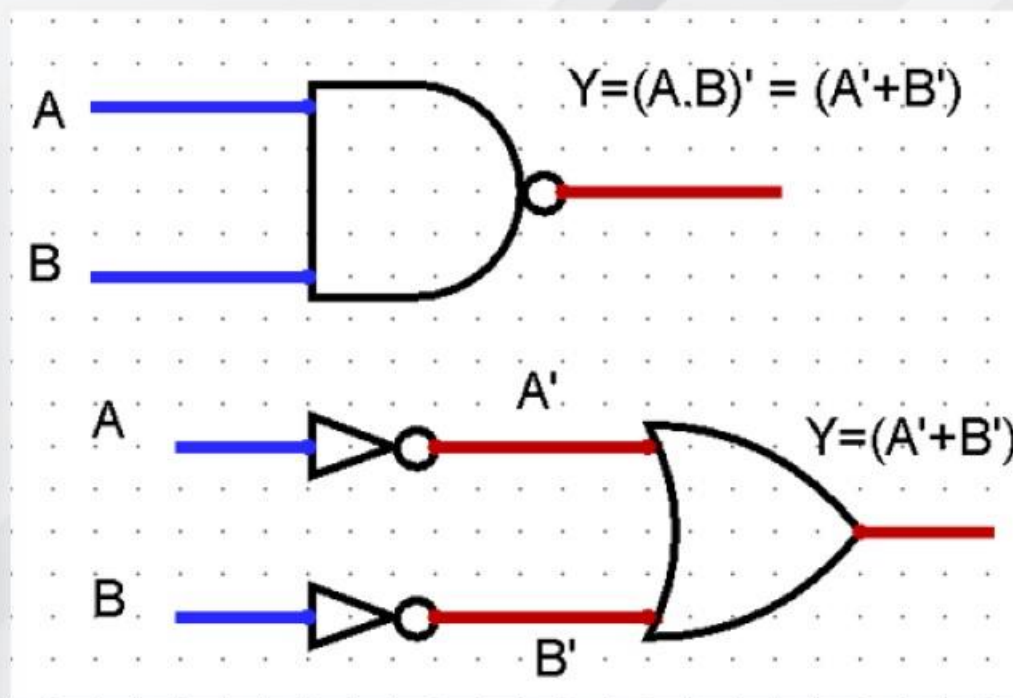
- T9 : Involution
 - (a) $x'' = x$
- T10 : De Morgan's Theorem
 - (a) $(X+Y)' = X'.Y'$
 - (b) $(X.Y)' = X'+Y'$

Boolean Algebra

De Morgan's Theorem

- $(A \cdot B)' = (A' + B')$

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

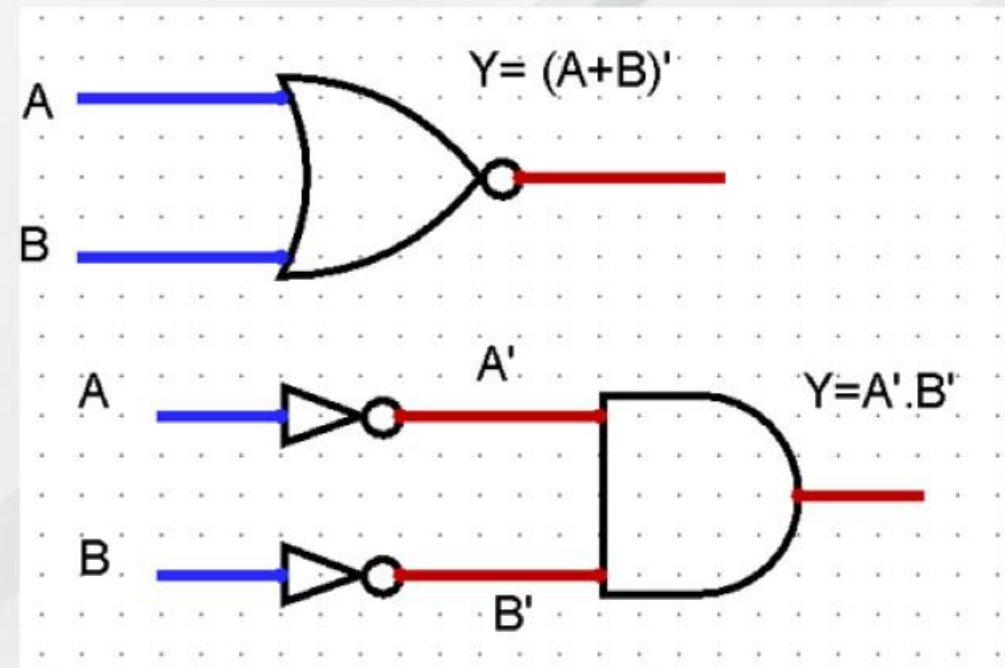


Boolean Algebra

De Morgan's Theorem

- $(A + B)' = A' \cdot B'$

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



Boolean Algebra

Expression Simplification

- When a Boolean expression is implemented with logic gates, each term requires a gate, and each variable\literals designates an input to gate.
- Simplify to contain the smallest number of literals (complemented and un-complemented variables) and terms

has three terms and eight literals

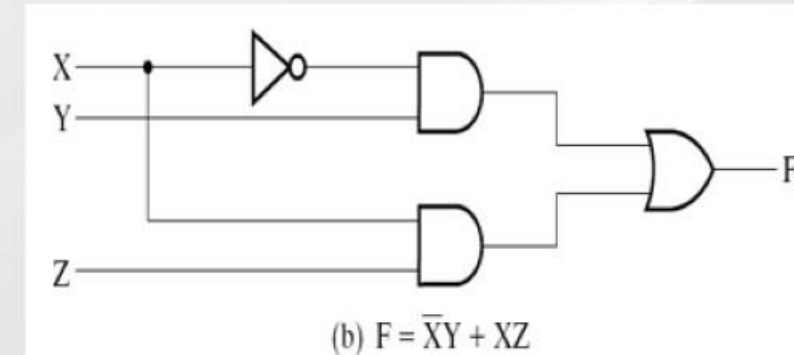
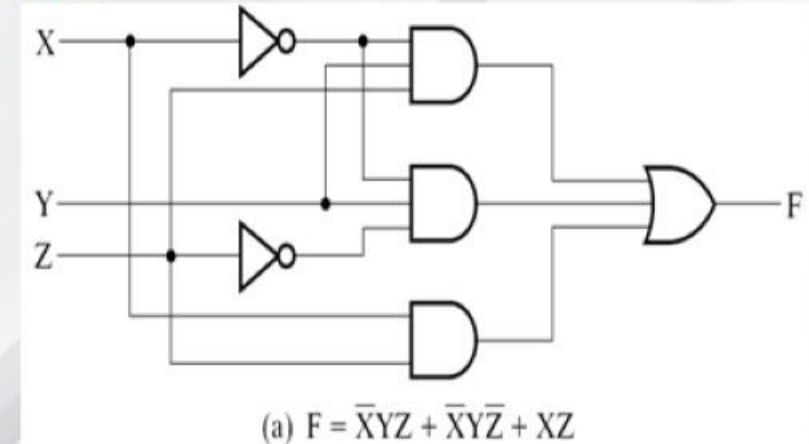
$$F = X'YZ + X'YZ' + XZ$$

$$F = X'Y(Z + Z') + XZ$$

has two terms and four literals

$$F = X'Y.1 + XZ$$

$$F = X'Y + XZ$$



It is highly desirable to find the **simplest circuit** implementation with the **smallest number of gates or wires**.

Boolean Algebra

Example 1

- For the given truth table, devise the Boolean equation.

- Solution

- $F = X'Y'Z + XY'Z' + XY'Z + XYZ' + XYZ$

- Simplify:

- $F = X'Y'Z + XY'(Z' + Z) + XY(Z' + Z)$

- $F = X'Y'Z + XY' + XY$

- $F = X'Y'Z + X(Y' + Y)$

- $F = X'Y'Z + X$

Truth Table

X Y Z	F
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	0
1 0 0	1
1 0 1	1
1 1 0	1
1 1 1	1

Boolean Algebra

Complementing Functions

- The complement (inverse) a function $F = F'$ is obtained from interchanging 0's to 1's in the value of F in the truth table.
- A Boolean equation for F' can be derived algebraically by using DeMorgan's Theorem:
 - interchange AND and OR operators
 - complement each constant value and literal
 - Examples:

$$\begin{aligned} F &= x'yz' + xy'z' \\ F' &= (x+y'+z)(x'+y+z) \end{aligned}$$

$$\begin{aligned} G &= (a'+bc)d' + e \\ G' &= (a(b'+c')+d)e' \end{aligned}$$

Boolean Algebra

- Exercises:
- - Simplify the following Boolean expressions to a minimum number of literals:
- a. $F(x,y) = xy + xy'$
 - $F = x(y + y') = x.1 = x$
- b. $F(x,y) = (x + y)(x + y')$
 - $F = xx + xy' + yx + yy' = x + xy' + xy + 0 = x(1 + y' + y) = x.1 = x$

Boolean Algebra

- c. $F(w,x,y,z) = xy + x(wz + wz')$
 - $F(w,x,y,z) = xy + xw(z+z') = xy + xw = x(y+w)$
- d. $F(A,B,C,D) = (BC' + A'D)(AB' + CD')$
 - $F(A,B,C,D) = AB'BC' + AB'A'D + CD'BC' + CD'A'D$
 - $= 0$

Boolean Algebra

- Reduce the following Boolean expressions to the indicated number of literals:

- $F(x,y,z) = x'z' + xyz + xz'$

to three literals

- $F = z' (x+x') + xyz$
 - $F = (z + z')(z' + xy)$
 - $F = xy + z'$

Boolean Algebra

- For the Boolean function do the following:
 - $F(x,y,z) = x'z' + xyz + xz'$
1. Obtain the truth table of F.
 2. Draw the logic diagram for the original Boolean expression.
 3. Use Boolean algebra to simplify the function to a minimum number of literals,
 4. Draw the logic diagram for the simplified Boolean expression.

Boolean Algebra

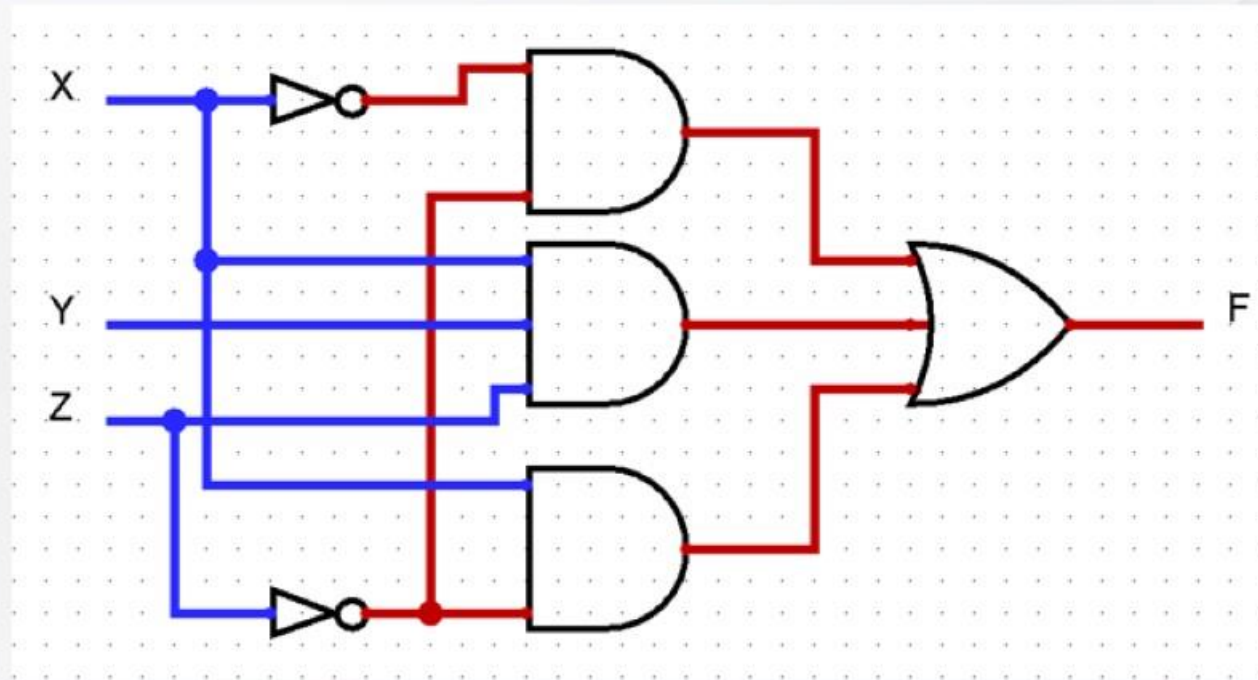
$$F(x,y,z) = x'z' + xyz + xz'$$

- Obtain the truth table of F.

X	Y	Z	X'	Z'	X'Z'	XYZ	XZ'	F
0	0	0	1	1	1	0	0	1
0	0	1	1	0	0	0	0	0
0	1	0	1	1	1	0	0	1
0	1	1	1	0	0	0	0	0
1	0	0	0	1	0	0	1	1
1	0	1	0	0	0	0	0	0
1	1	0	0	1	0	0	1	1
1	1	1	0	0	0	1	0	1

Boolean Algebra

- Draw the logic diagram for the original Boolean expression.
- $F(x,y,z) = x'z' + xyz + xz'$



Boolean Algebra

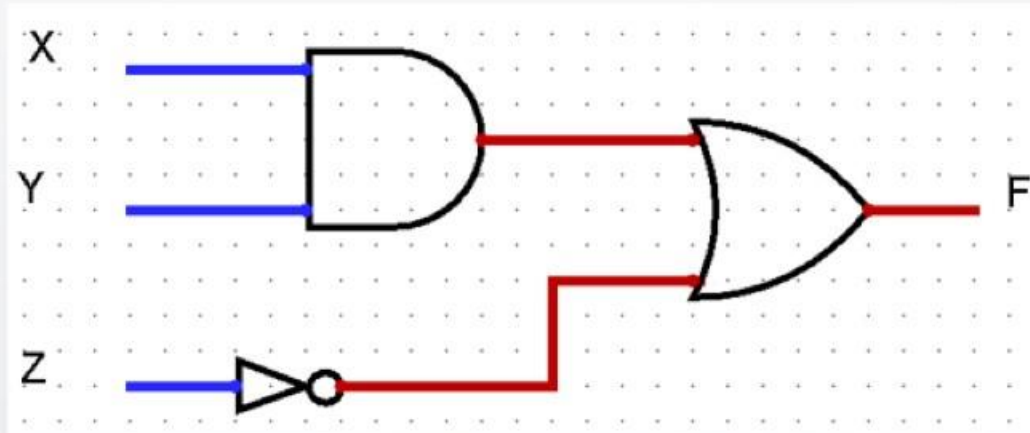
- Use Boolean algebra to simplify the function to a minimum number of literals,

- $F(x,y,z) = x'z' + xyz + xz'$

- $F = z' (x+x') + xyz$
 - $F = (z + z')(z' + xy)$
 - $F = xy + z'$

Boolean Algebra

- Draw the logic diagram for the simplified Boolean expression



X	Y	Z	XY	Z'	F
0	0	0	0	1	1
0	0	1	0	0	0
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	1	0	1

Boolean Algebra

- Reduce the following Boolean expressions to the indicated number of literals:

- $F(w,x,y,z) = w'x (z' + y'z) + x(w + w'y z)$

to one literal

- $F = x(w'z' + w'y'z + w + w'yz)$
- $F = x(w'z' + w + w'z(y + y'))$
- $F = x(w + w'(z' + z))$
- $F = x(w + w')$
- $F = x$

Boolean Algebra

- Reduce the following Boolean expressions to the indicated number of literals:

- $F(w,x,y,z) = wxy'z + w'xz + wxyz$

to two literal

- $F = xz(wy' + w' + wy)$
- $= xz(w(y + y') + w')$
- $= xz(w + w')$
- $= xz$

Boolean Algebra

- Simplify the following Boolean expressions to a minimum number of literals:

$$ABC + A'B + ABC'$$

- Solution
- $ABC + A'B + ABC' = AB + A'B = B$

Boolean Algebra

- Simplify the following Boolean expressions to a minimum number of literals:

$$(x + y)'(x' + y')$$

- **Solution:**
-
- $(x + y)'(x' + y') = x'y'(x' + y') = x'y'$

Boolean Algebra

- Simplify the following Boolean expressions to a minimum number of literals:

$$x'yz + xz$$

Solution:

- $x'yz + xz = (x'y + x)z$
- $= z(x + x')(x + y)$
- $= z(x + y)$



Thank You