Unit - II

Logical Operations , Logic Gates & Boolean Algebra

Logic Gates

- **Standard Logic Gates**

1. AND
2. OR
3. NOT

- **Exclusive Logic Gates**

1. XOR
2. XNOR

- **Universal Logic Gates**

1. NAND
2. NOR

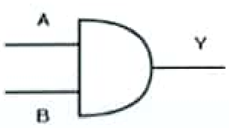
Standard Logic Gates

1. AND Gate:

- It is a basic type of digital circuit. It has two or more inputs and one output.
- The AND gate is an electronic circuit that gives a **high** output (1) only if **all** its inputs are high i.e. 1.
- A dot (.) is used to show the AND operation i.e. A.B
- Sometimes this dot is omitted and it is written as AB
- The AND operation for the output is defined as "Y equals to A **AND** B"

$$Y = A.B$$

Truth Table and Diagram of AND Gate:



Truth Table		
A	B	A . B
1	1	1
1	0	0
0	1	0
0	0	0

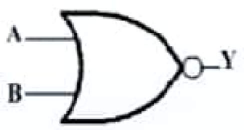
2. OR Gate:

- It is also basic type of digital circuit. It has two or more inputs and one output.
- The OR gate is an electronic circuit that gives a **High** output (1) if either of one input is high (1).
- A plus (+) is used to show the OR operation i.e. A + B
- The OR operation for the output is defined as "Y equals to A **OR** B"

$$Y = A + B$$

6

Truth Table and Diagram of OR Gate:



inputs		output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

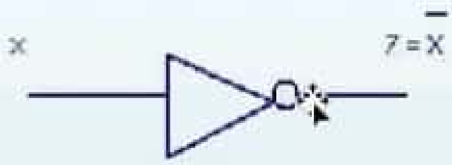
3. NOT Gate:

- The NOT gate is an electronic circuit that produces an inverted version of the input at its output.
- The NOT logic operation returns true if its input is false, and false if its input is true.
- It is also known as an *inverter*.
- If the input variable is A, the inverted output is known as NOT A. This is also shown as A', or A with a bar over the top.

$$Y = A'$$

Truth Table and Diagram of NOT Gate:

NOT Gate



TRUTH TABLE

INPUT	OUTPUT
X	Z
0	1
1	0

Exclusive Logic Gates

1. XOR Gate:

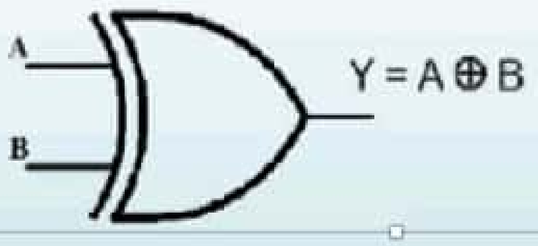
- The XOR logic operation (which stands for "Exclusive OR" returns true if either of its inputs differ, and false if they are all the same.
- In other words, if its inputs are a combination of true and false, the output of XOR is true. If its inputs are all true or all false, the output of XOR is false (0).
- In Boolean algebra, the XOR value of two inputs A and B can be written as **A⊕B** (the XOR symbol, ⊕, resembles a plus sign inside a circle).

$$Y = A \oplus B$$

OR

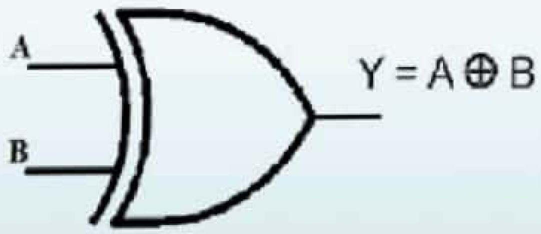
$$Y = \overline{A}B + A\overline{B}$$

Truth Table and Diagram of XOR Gate:



INPUT		OUTPUT
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Truth Table and Diagram of XOR Gate:



INPUT		OUTPUT
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

2. XNOR Gate:

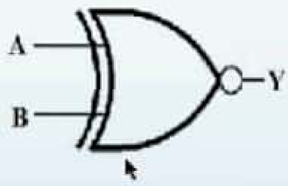
- The XNOR logic operation (which stands for "Exclusive NOT OR" returns true (1) if both the inputs are same, and false (0) if inputs are different.
- In other words, if its inputs are a combination of true(1) and false(0), then the output of XNOR is false(0). If its inputs are all true or all false, the output of XNOR is true(1).
- In Boolean algebra, the XNOR value of two inputs A and B can be written as $\overline{A \oplus B}$

$$Y = \overline{A \oplus B}$$

OR

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

Truth Table and Diagram of XNOR Gate:



inputs		output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Universal Logic Gates

1. NAND Gate:

- This is a NOT-AND gate which is equal to an AND gate followed by a NOT gate.
- The outputs of all NAND gates are high(1) if **any** of the inputs are low(0).
- The symbol is an AND gate with a small circle on the output. The small circle represents inversion.
- In Boolean **algebra**, the NAND value of two inputs A and B can be written as $\overline{AB}$ (AB with an overscore)
- NAND Gate is also called **"Universal Logic gate"** because any other logic operation can be created by using only NAND gates.

$$Y = \overline{AB}$$

Truth Table and Diagram of NAND Gate:

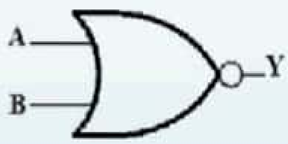
Inputs		output
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

2. NOR Gate:

- This is a NOT-OR gate which is equal to an OR gate followed by a NOT gate.
- The outputs of all NOR gates are low(0) if **any** of the inputs are high(1).
- The symbol is an OR gate with a small circle on the output. The small circle represents inversion.
- In Boolean algebra, the NOR value of two inputs A and B can be written as $\overline{A + B}$ (A+B with an overscore).
- NOR Gate is also called **"Universal Logic gate"** because any other logic operation can be created by using only NOR gates.

$$Y = \overline{A + B}$$

Truth Table and Diagram of NOR Gate:



inputs		output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

17

Boolean Algebraic Properties

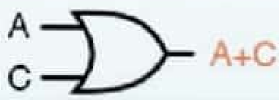
S.No.	Theorems/ Identities	Name (if any)
1	$x + \overline{x} = 1$	Additive Identities
2	$x . \overline{x} = 0$	Multiplicative Identities
3	$x + 0 = x$	Additive Identities
4	$x . 0 = x$	Multiplicative Identities
5	$x + y = y + x$	Commutative Property
6	$x . y = y . x$	Commutative Property
7	$x + (y + z) = (x + y) + z$	Associative Property
8	$x(yz) = (xy)z$	Associative Property
9	$x(y + z) = xy + xz$	Distributive Property

Realizing Circuits From Boolean Expressions

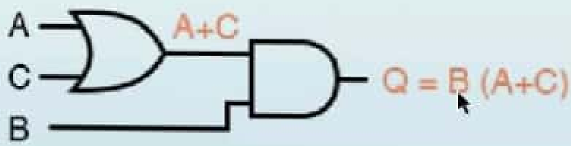
Example : 1

- Evaluate the expression "B(A + C)"

Step - 1



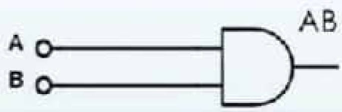
Step - 2



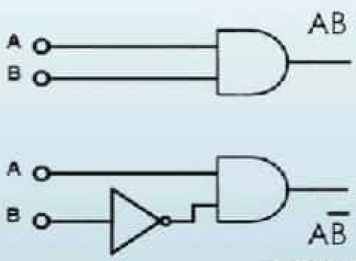
Example : 2

- Evaluate the expression $\overline{(AB + A\overline{B})}$

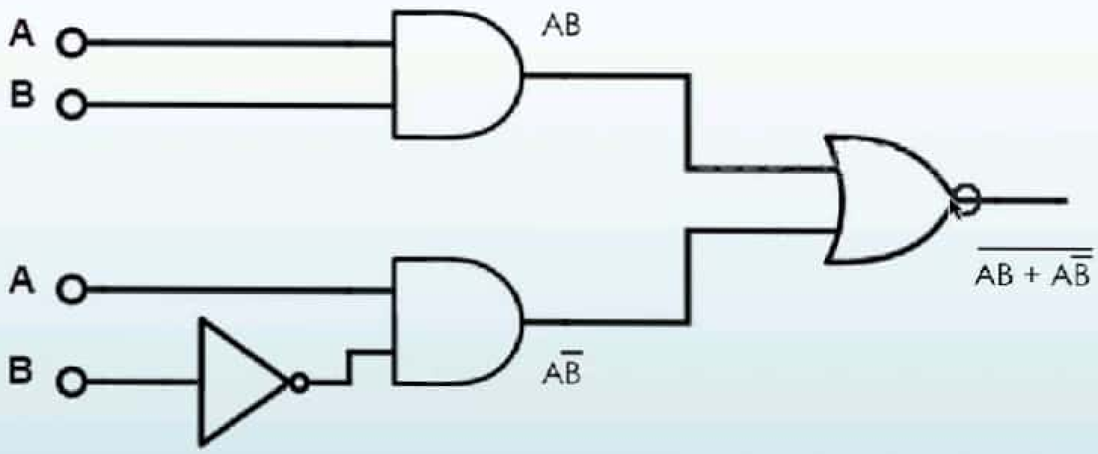
Step - 1



Step - 2



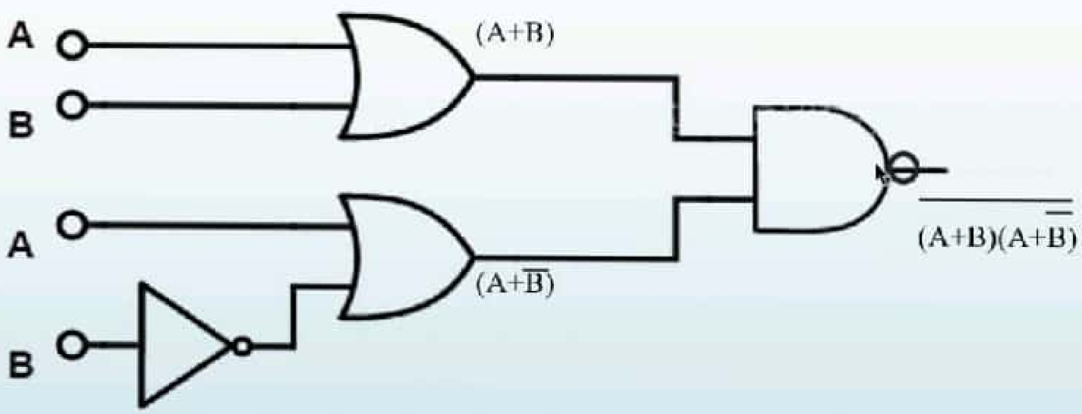
Step - 3



Example : 3

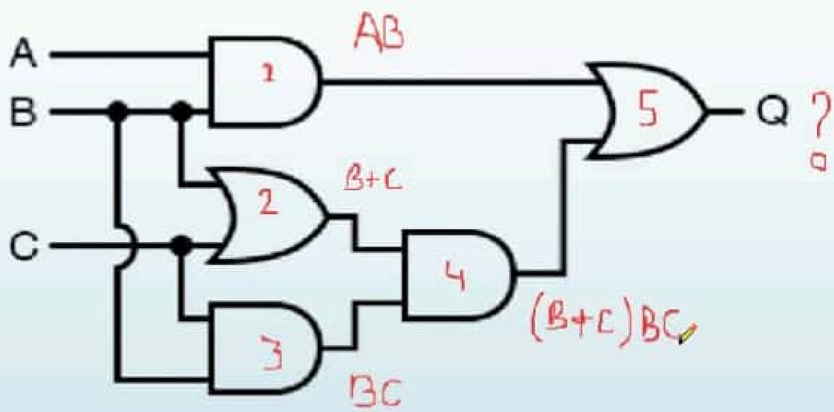
- Evaluate the expression $\overline{(A+B)(A+\overline{B})}$

Step - 3



Realizing Boolean Expressions from Circuits

Example : 1



4. Now that we have a Boolean expression to work with, we need to apply the rules of Boolean algebra to reduce the expression to its simplest form

$AB + BC(B + C)$

↓ Distributing terms

$AB + BBC + BCC$

↓ Applying identity $AA = A$ to 2nd and 3rd terms

$AB + BC + BC$

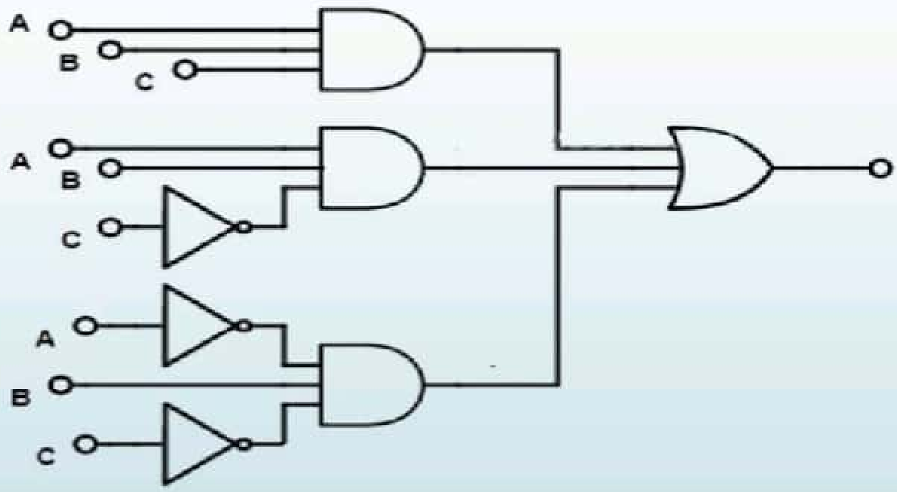
↓ Applying identity $A + A = A$ to 2nd and 3rd terms

$AB + BC$

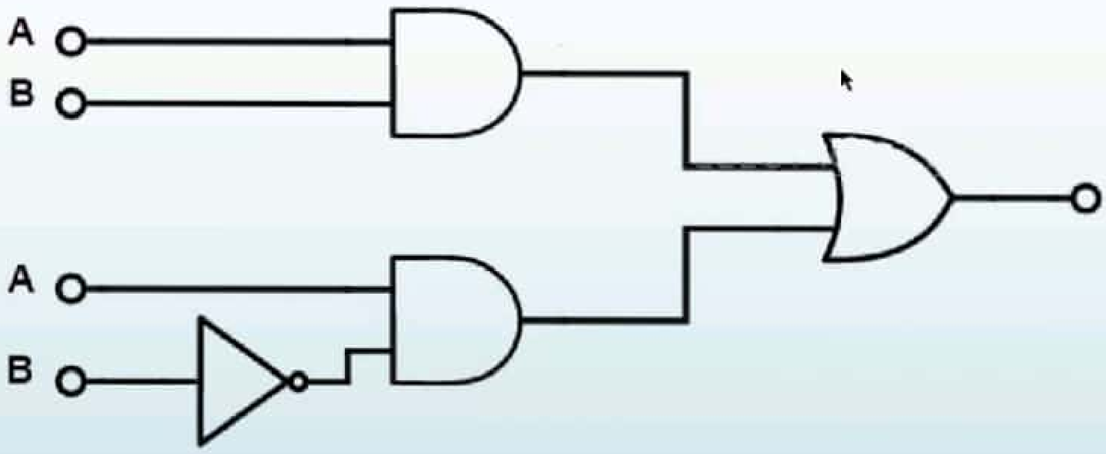
↓ Factoring B out of terms

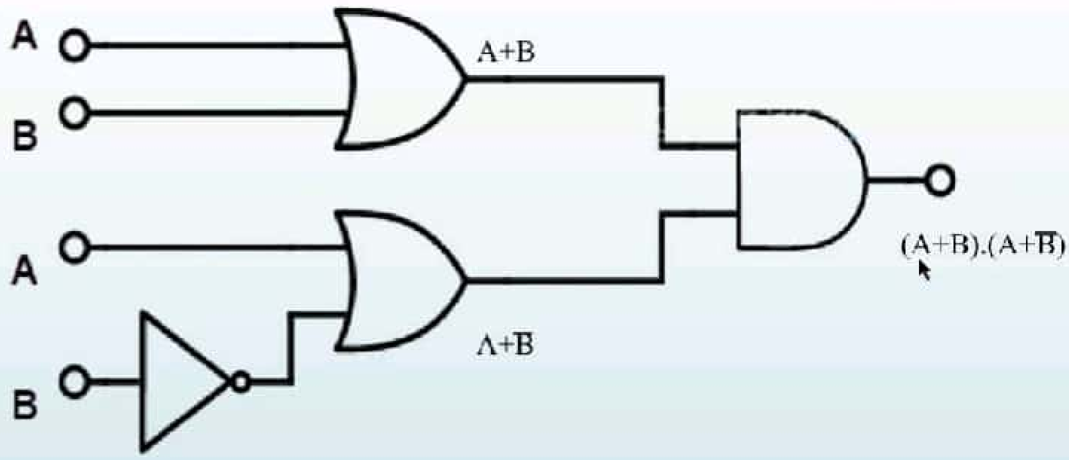
$B(A + C)$

Example : 2



Example : 3





Scanned with CamScanner

Postulates of Boolean Algebra

- Postulate 1:
  - a) A = 0, if and only if, A is not equal to 1
  - b) A = 1, if and only if, A is not equal to 0
- Postulate 2:
  - a) $x + 0 = x$
  - b) $x \cdot 1 = x$
- Postulate 3: Commutative Law
  - a) $x + y = y + x$
  - b) $x \cdot y = y \cdot x$

- Postulate 4: Associative Law

  a) $x + (y + z) = (x + y) + z$

  b) $x \cdot (y \cdot z) = (x \cdot y) \cdot z$

- Postulate 5: Distributive Law

  a) $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$

  b) $x + (y \cdot z) = (x + y) \cdot (x + z)$

- Postulate 6:

  a) $x + \overline{x} = 1$

  b) $x \cdot \overline{x} = 0$

Theorems of Boolean Algebra

The following two theorems are used in Boolean algebra.

- DeMorgan's theorem
- Duality theorem

37

DeMorgan's Theorem

- De Morgan has suggested two theorems which are extremely useful in Boolean Algebra.
- DeMorgan's Theorems are two additional simplification techniques that can be used to simplify Boolean expressions.

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

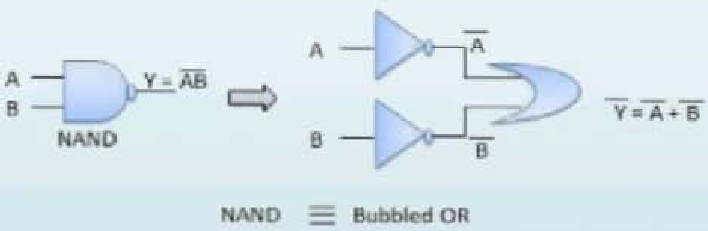
$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

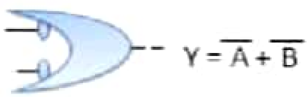
Theorem 1

$$\overline{A \,.\, B} = \overline{A} + \overline{B}$$

NAND = Bubbled OR

The left hand side (LHS) of this theorem represents a NAND gate with inputs A and B, whereas the right hand side (RHS) of the theorem represents an OR gate with inverted inputs. This OR gate is called as **Bubbled OR**





Bubbled OR

Table showing verification of the De Morgan's first theorem –

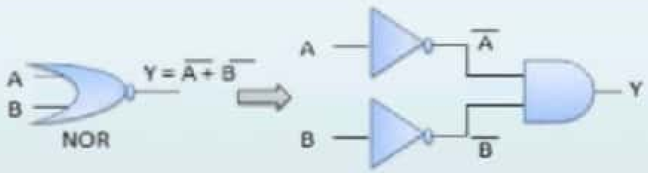
A	B	$\overline{AB}$	$\overline{A}$	$\overline{B}$	$\overline{A}+\overline{B}$
0	0	1	1	1	1
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	0	0	0

Theorem 2

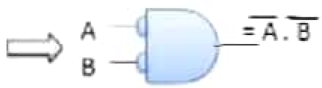
$$\overline{A+B} = \overline{A} \cdot \overline{B}$$

NOR = Bubbled AND

The LHS of this theorem represents a NOR gate with inputs A and B, whereas the RHS represents an AND gate with inverted inputs. This AND gate is called as **Bubbled AND**



NOR ≡ Bubbled AND



Bubbled AND

Table showing verification of the De Morgan's second theorem –

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

Duality Theorem

- This theorem states that the **dual** of the Boolean function is obtained by interchanging the logical AND operator with logical OR operator and zeros with ones.
- For every Boolean function, there will be a corresponding Dual function.

Theorems/ Identities
$x + x = x$
$x + 1 = 1$
$x + y = y + x$
$x + \overline{x} = 1$
$x + 0 = x$
$x . y + z = x . y + x . z$

Dual Theorems/ Identities
$x \cdot x = x$
$x \cdot 0 = 0$
$x . y = y . x$
$x . \overline{x} = 0$
$x . 1 = x$
$x + y . z = x + y . x + z$

Boolean & DeMorgan's Theorem

1) $X \cdot 0 = 0$
2) $X \cdot 1 = X$
3) $X \cdot X = X$
4) $X \cdot \overline{X} = 0$
5) $X + 0 = X$
6) $X + 1 = 1$
7) $X + X = X$
8) $X + \overline{X} = 1$
9) $\overline{\overline{X}} = X$

10A) $X \cdot Y = Y \cdot X$
10B) $X + Y = Y + X$
— Commutative Law

11A) $X(YZ) = (XY)Z$
11B) $X + (Y + Z) = (X + Y) + Z$
— Associative Law

12A) $X(Y + Z) = XY + XZ$
12B) $(X + Y)(W + Z) = XW + XZ + YW + YZ$
— Distributive Law

13A) $X + \overline{X}Y = X + Y$
13B) $\overline{X} + XY = \overline{X} + Y$
13C) $X + \overline{X}\overline{Y} = X + \overline{Y}$
13D) $\overline{X} + X\overline{Y} = \overline{X} + \overline{Y}$
— Consensus Theorem

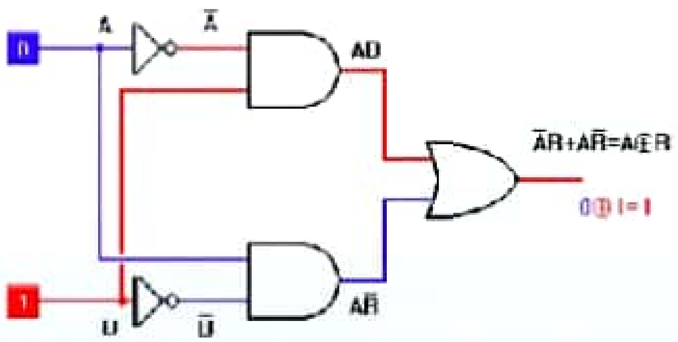
14A) $\overline{XY} = \overline{X} + \overline{Y}$
14B) $\overline{X + Y} = \overline{X}\,\overline{Y}$
— DeMorgan's

Important Theorems of Boolean Algebra

S.No.	Theorems/ Identities	Dual Theorems/ Identities	Name (if any)
1	$x + x = x$	$x \cdot x = x$	Idempotent Law
2	$x + 1 = 1$	$x \cdot 1 = x$	
3	$x + x \cdot y = x$	$x \cdot (x + y) = x$	Absorption Law
4	$\overline{\overline{x}} = x$		Involution Law
5	$x \cdot (\overline{x} + y) = x \cdot y$	$x + (\overline{x} \cdot y) = x + y$	
6	$\overline{x + y} = \overline{x} \, . \, \overline{y}$	$\overline{x \, . \, y} = \overline{x} + \overline{y}$	De Morgan's Law

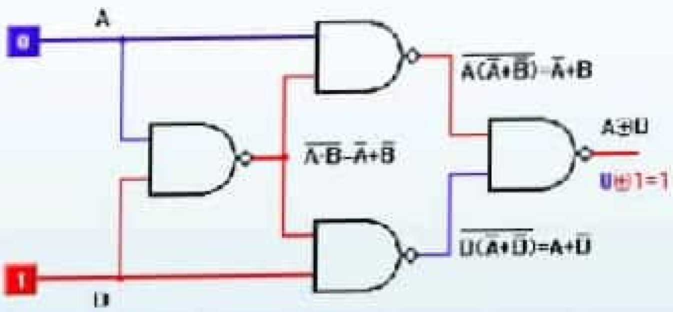
45

Deriving XOR Function



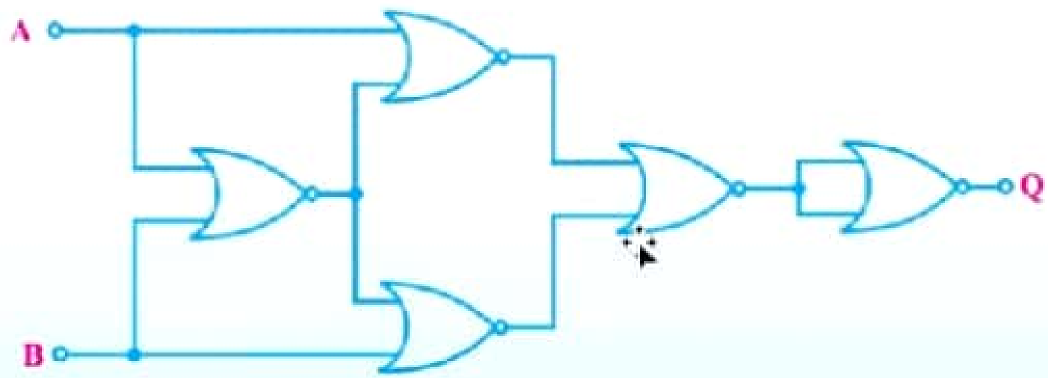
The practical problem with the circuit above is that it contains three different kinds of gates: AND, OR, and NOT.

Deriving XOR Function



This problem can be solved by using single quad two-input NAND gate.

Deriving XOR Function



This problem can also be solved by using single quad two-input NOR gate.

48

Assignment

- Make all three circuits for XNOR

Reducing Boolean Expression by Algebraic Reduction

Q-1

$$A\overline{B}D + A\,\overline{B}\,\overline{D} = \quad A\,\overline{B}\,(D + \overline{D})$$

$$A\,\overline{B}\,(1)$$

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

53

Assignment

Simplify: $C + \overline{BC}$

Simplify: $\overline{A}(A + B) + (B + AA)(A + \overline{B})$

Simplify: $\overline{AB}(\overline{A} + B)(\overline{B} + B)$

Simplify: $AB + A(B + C) + B(B + C)$