AKTU B.Tech II-Year

GATEWAY CLASSES

EC & EC Allied

DSD : Digital System Design

Unit-1 Lec-4



Avinash Sir

Today's target

- Code Conversions
- Review of Boolean Algebra and Laws
- De Morgan Theorem
- PYQ , DPP

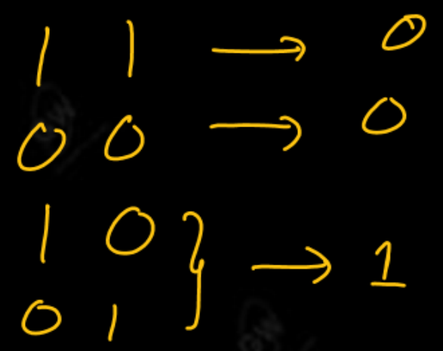
Gateway Classes

AKTU : *DIGITAL SYSTEM DESIGN (BEC-302)* Syllabus

UNIT-1 : Logic simplification and combinational logic design

- Number Systems
- Binary arithmetic
- Signed magnitude representation
- Binary codes, code conversion
- Review of Boolean algebra and Demorgans theorem
- SOP & POS forms
- Canonical forms
- Karnaugh maps up to 5 variables
- Tabulation method.

Gateway Classes



GRAY CODE CONVERSION

(i) Binary → Gray Code Conversion

Rule: 1. The most significant bit (MSB) of Gray code = MSB of Binary code.

2. Each next Gray code bit = XOR of previous Binary bit and current Binary bit.

Example: Convert Binary 1011 to Gray Code

MSB remains same → 1

2nd bit = 1 ⊕ 0 = 1

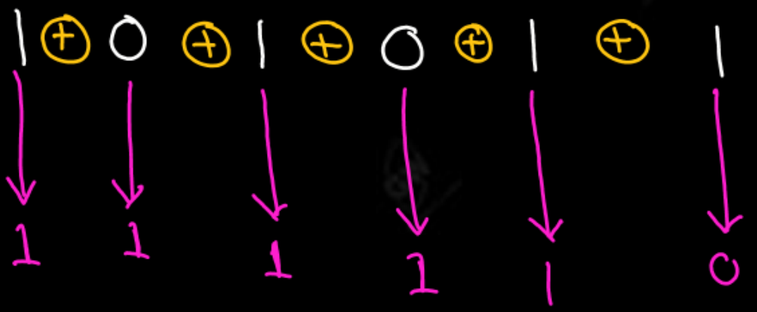
3rd bit = 0 ⊕ 1 = 1

4th bit = 1 ⊕ 1 = 0

Gray Code = 1110



Ex. Convert 101011 into Gray Code



(111110) Gray Code.

Gateway Classes

GRAY CODE CONVERSION

(ii) Gray Code → Binary Conversion

Rule:

The MSB of Binary = MSB of Gray.

Each next Binary bit = Previous Binary bit ⊕ Current Gray bit.

Example:

Convert Gray 1110 to Binary

MSB remains same → 1

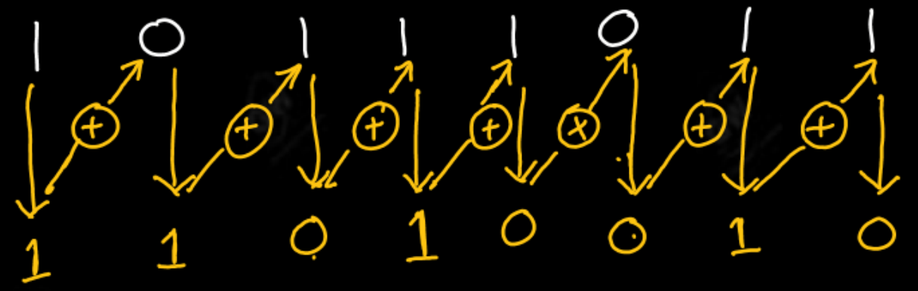
2nd bit = 1 ⊕ 1 = 0

3rd bit = 0 ⊕ 1 = 1

4th bit = 1 ⊕ 0 = 1

Binary = 1011





Binary = $(11010010)_2$

Ex. Convert 10111011

Gateway Classes

Binary ↔ Binary Coded Decimal (BCD)

(a) Binary → BCD

Binary → decimal → BCD

BCD represents each decimal digit using 4 bits.

0101 ← → 1001

Example: Decimal 59

Binary = 111011 (6 bits)

BCD = 0101 1001 (two groups of 4 bits → "5" and "9")

5 9

(b) BCD → Binary

BCD → Dec → Binary

Convert each 4-bit group into decimal, then into full binary.

Gateway Classes

Binary ↔ Excess-3 (XS-3) Code

Excess-3 is a self-complementary code, each decimal digit = BCD + 3.

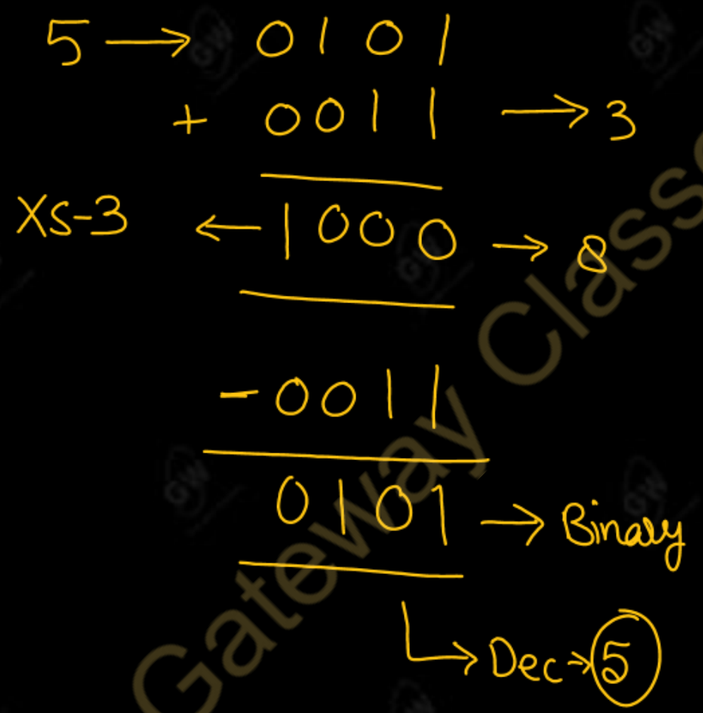
(a) Decimal → XS-3

Example: Decimal 5

BCD = 0101

Add 3 = 1000

XS-3 = 1000



(b) XS-3 → Decimal

Example: XS-3 = 1000

Subtract 3 → 0101 (5 in decimal)

Gateway Classes

Boolean Algebra

- Boolean Algebra is used to analyze and simplify digital (logic) circuits.
- Each variable in Boolean algebra has values either 0 (Low) or 1 (High).
- The basic operations used in Boolean algebra are:

Low → 0 → false

High → 1 → True

1 + 0 = 1 1 . 0 = 0 1 → 0

(a) Logical addition (b) Logical multiplication (c) Complementation

- Laws of Boolean Algebra

(1) Commutative Law (2) Associative Law (3) Distributive Law

(4) AND Laws (5) OR Laws (6) Inversion Law

(7) Absorption Law (8) De Morgan's Law

Gateway Classes

(1) Commutative Law

The commutative law states that changing the sequence of the variables does not have any effect on the output of a logic circuit.

(a) $A \cdot B = B \cdot A$ (b) $A + B = B + A$

(2) Associative Law

This law states that the order in which the logic operations are performed is irrelevant as their effect is the same.

(a) $(A \cdot B) \cdot C = A \cdot (B \cdot C)$ (b) $(A + B) + C = A + (B + C)$

(3) Distributive Law

$$A \cdot (B + C) = A \cdot B + A \cdot C$$

Gateway Classes



$\overline{\overline{A}} = A$

$A = 1$

$\overline{A} = 0$

$1 \cdot 0 \rightarrow 0$

(4) AND Laws

These laws use the AND operation. Therefore, they are called AND laws.

(a) $A \cdot 0 = 0$ (b) $A \cdot 1 = A$ (c) $A \cdot A = A$ (d) $A \cdot \bar{A} = 0$

Any var. ⟵ ⊤ ↳ always

(5) OR Laws

These laws use the OR operation. Therefore, they are called OR laws.

(a) $A + 0 = A$ (b) $A + 1 = 1$ (c) $A + A = A$ (d) $A + \bar{A} = 1$

$0+1=1$
$1+1=1$

$1+1=1$
$0+0=0$

Let $A = 1$, 0
$\bar{A} = 0$, 1
$1+0=1$

(6) Inversion Law

The inversion law states that double inversion of a variable results in the original variable itself

$$\overline{\overline{A}} = A$$

(7) Absorption Law

This Law enables a reduction in a complicated expression to a simple one by absorbing like terms

$$A + AB = A \Rightarrow A(1+B) = A$$

1

Gateway Classes

M.Imp

AKTU-2023-24

(8) De Morgan's Law

The two important laws have been suggested by De Morgan's.

(i) This law states that the complement of a product is equal to the addition of the complements.

$$\overline{AB} = \overline{A} + \overline{B} \qquad \overline{ABC} = \overline{A} + \overline{B} + \overline{C}$$

(ii) This law states that the complement of a sum is equal to the product of the complements.

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

Ex:

$$A + BC = (A+B)(A+C)$$

Proof:

$$(A+B)(A+C) = A \cdot A + A \cdot C + B \cdot A + B \cdot C$$

($A \cdot A = A$)

$$= A + A \cdot C + B \cdot A + B \cdot C$$

$$= A(1+C) + B \cdot A + B \cdot C$$

($1 + C = 1$)

$$= A + BA + BC$$

$$= A(1+B) + BC$$

($1 + B = 1$)

$$= A + BC$$

Gateway Classes

Example – 1 : Simplify the expression $A + \bar{A}B + AB$ using Boolean algebra.

(1) (3)

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

$= A + B$

Ans.

Rule :- $A + \bar{A} = 1$

$\bar{A}B + A(1 + B)$

$A + \bar{A}B$

Example – 2 : Simplify the Boolean function using Boolean algebra theorems

$$A'B'C' + A'BC' + AB'C' + ABC'$$

$= A'C'(B' + B) + AC'(B' + B)$

$= A'C' + AC'$

$= C'(A + A')$

$= C'$

Ans.

$\because B + B' = 1$

$\because A + A' = 1$

Gateway Classes

Example – 3 : Simplify the Boolean function F = XY + XY'Z + YZ' using Boolean algebra.

$= XY(Z+Z') + XY'Z + YZ'$ ← $XY[Z+\bar{Z})$

Dis Prop.

$= XYZ + XYZ' + XY'Z + YZ'$

$= XZ(Y+Y') + YZ'(X+1)$ ↑ ↑ ↳ 1

$\therefore Y+Y'=1$, $X+1=1$

$$F = XZ + YZ'$$

Ans.

Example – 4 : Simplify the following expression Y = (A+B)(A+C).

$Y = (A+B)(A+C)$

$= A \cdot A + A \cdot C + B \cdot A + B \cdot C$ ← A

$= A + AC + BA + BC$

$= A(1+C) + BA + BC$

$= A + AB + BC = AC(1+B) + BC$

$$Y = A + BC$$

Gateway Classes

SOP and POS Representation for Logic Expression

Sum of Product → Var → A, B, C

$A\bar{B}C + \bar{A}B\bar{C} + \bar{A}\bar{B}\bar{C}$

Any logic expression may be expressed in the following two forms

(i) Sum of Product form (SOP)

$f(A, B, C) = \underline{AB} + \underline{BC} + AC$ → Product

(AB: Cx, BC: Ax, AC: Bx)

Standard or Canonical SOP form

Each product term consists of all the variables in the complemented or un-complemented form.

$f(A, B, C) = \underline{ABC} + \underline{A\bar{B}\bar{C}} + \underline{\bar{A}BC}$

Product of Sum

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

(ii) Product of Sum form (POS)

$f(A, B, C) = (A+B).(B+C).(A+C)$

(A+B: Cy, B+C: Ax, A+C: By)

Standard or Canonical POS form

Each sum term consists of all the variables in the complemented or un-complemented form.

$f(A, B, C) = (A + B + C) . (A + \bar{B} + \bar{C}) . (\bar{A} + B + C) . (\bar{A}+\bar{B}+\bar{C})$

Gateway Classes

Conversion of a logical expression to standard SOP or POS form

Example – 1 : Convert the expression $Y = A\,B + \bar{A}\,C + B\,C$ into the standard SOP form.

$$Y = AB(C+\bar{C}) + \bar{A}C(B+\bar{B}) + BC(A+\bar{A})$$

$$= \underline{ABC} + AB\bar{C} + \underline{\bar{A}BC} + \bar{A}\bar{B}C + \underline{ABC} + \underline{\bar{A}BC}$$

$$= \underbrace{ABC} + \underbrace{AB\bar{C}} + \underbrace{\bar{A}BC} + \underbrace{\bar{A}\bar{B}C} \rightarrow \text{minterm}$$

$A + A = A$

$A \cdot B \rightarrow C \rightarrow (C+\bar{C}) \rightarrow 1$

$+ \bar{A}C \rightarrow B \rightarrow (B+\bar{B})$

$+ BC \rightarrow A \rightarrow (A+\bar{A})$

$0 \leftarrow C\cdot\bar{C} \quad B\cdot\bar{B} \quad A\cdot\bar{A}$

$C \quad B \quad A$

Ans

$\therefore A + BC = (A+B)(A+C)$

Example – 2 : Convert the expression $Y = (A+B)(A+C)(B+\bar{C})$ to standard POS form.

$$Y = (\overbrace{A+B}^{x} + C\cdot\bar{C})(A+C+B\cdot\bar{B})(B+\bar{C}+A\cdot\bar{A})$$

$$= (A+B+C)(A+B+\bar{C})(A+B+C)(A+\bar{B}+C)(A+B+\bar{C})(\bar{A}+B+\bar{C})$$

$$\boxed{Y = (A+B+C)(A+B+\bar{C})(A+\bar{B}+C)(\bar{A}+B+\bar{C})}$$

Maxterm

Ans

Gateway Classes

SOP

Minterm and Maxterm

Minterm

- A product term which contains all the function variables either in complemented form or in un-complemented form is called a Minterm. ABC , ABC̄ ,
- Each individual term in the standard SOP form is called a Minterm

Maxterm

- A sum term which contains all the function variables either in complemented form or in un-complemented form is called a Maxterm.
- Each individual term in the standard POS form is called a Maxterm

Minterm { 0 → Comp, 1 → UnComp | Max. 0 → UnComp, 1 → Comp

Maximum number of minterms = 2^n

Maximum number of maxterms = 2^n

Where n = number of variables

Variables		Minterms	Maxterms
A	B	m_i	M_i
0	0	$m_0 = \bar{A}\,\bar{B}$	$M_0 = A + B$
0	1	$m_1 = \bar{A}\,B$	$M_1 = A + \bar{B}$
1	0	$m_2 = A\,\bar{B}$	$M_2 = \bar{A} + B$
1	1	$m_3 = AB$	$M_3 = \bar{A} + \bar{B}$

Gateway Classes

Logic Gates

- The most basic element of digital circuits are the logic gates.
- Boolean functions may be practically implemented by using Logic gates.
- Electronic gates require a power supply.

LOGIC GATES

fundamental

Basic gates

(i) NOT gate

(ii) AND gate

(iii) OR gate

Universal gates

(i) NAND gate NOT + AND

(ii) NOR gate NOT + OR

Special purpose gate

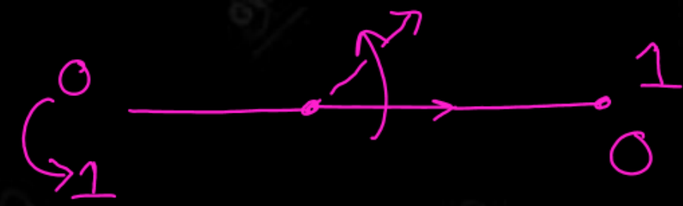
(i) EXOR gate

(ii) EXNOR gate

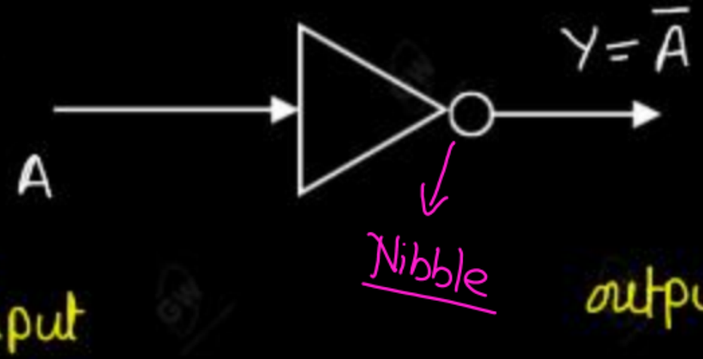
Gateway Classes

(1) NOT Gate

- The NOT gate is also known as an inverter.
- Output is the inverted version or complement of its input.
- NOT gate has only one input and one output.



Symbol



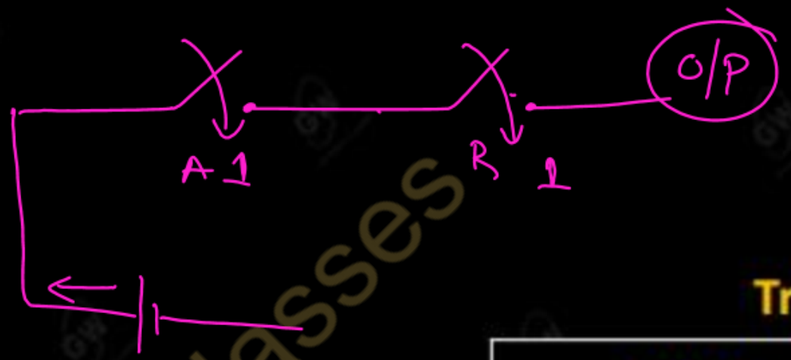
Truth Table

Input	Output
A	$Y = \bar{A}$
0	1
1	0

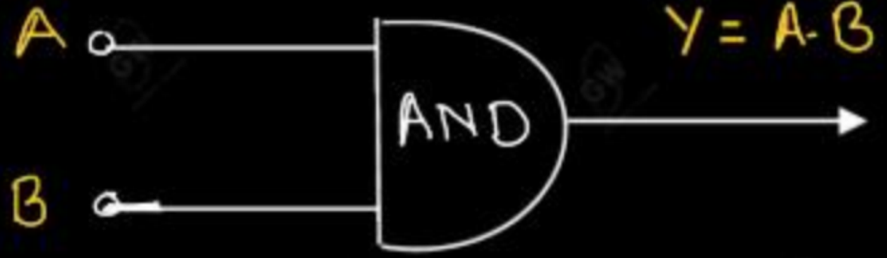
Gateway Classes

(2) AND gate

- The AND gate is an electronic circuit that gives a high output (1) only if all its inputs are high.
- It performs logical multiplication on its inputs.



Symbol



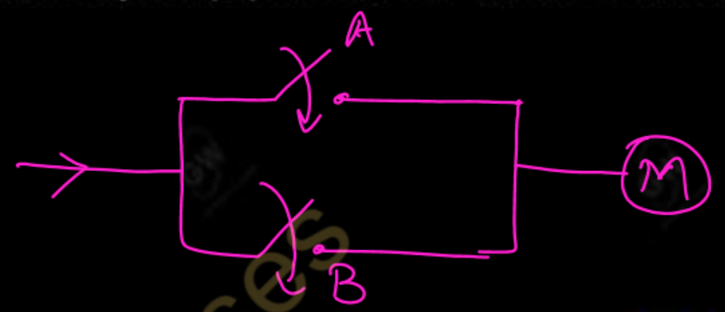
Truth Table

Input		Output
A	B	Y = A . B
0	0	0
0	1	0
1	0	0
1	1	1

Gateway Classes

(3) OR gate

- The OR gate is an electronic circuit that gives a high output (1) if one or more of its inputs are high.
- It performs logical addition on its inputs.



Symbol



Truth Table

Input		Output
A	B	Y = A + B
0	0	0
0	1	1
1	0	1
1	1	1

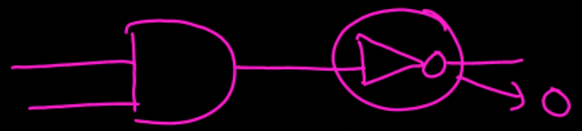
Gateway Classes

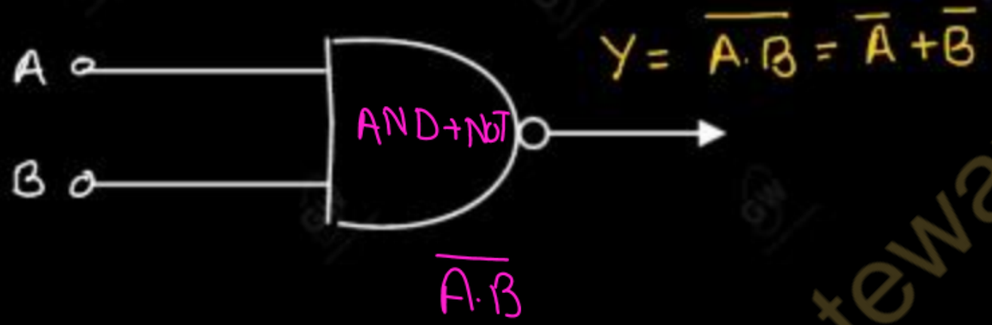
(4) NAND gate

Universal

- NOT + AND gate
- The input of all NAND gates are high if any of the input are low
- The small circle represents inversion

Symbol





Truth Table

Input		Output	
A	B	A.B	$Y = \overline{A.B}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Gateway Classes

(5) NOR gate

OR + NOT

- Not + OR Gate
- The outputs of all NOR gates are low if any of the inputs are high.

A+B

Symbol



Gateway Classes

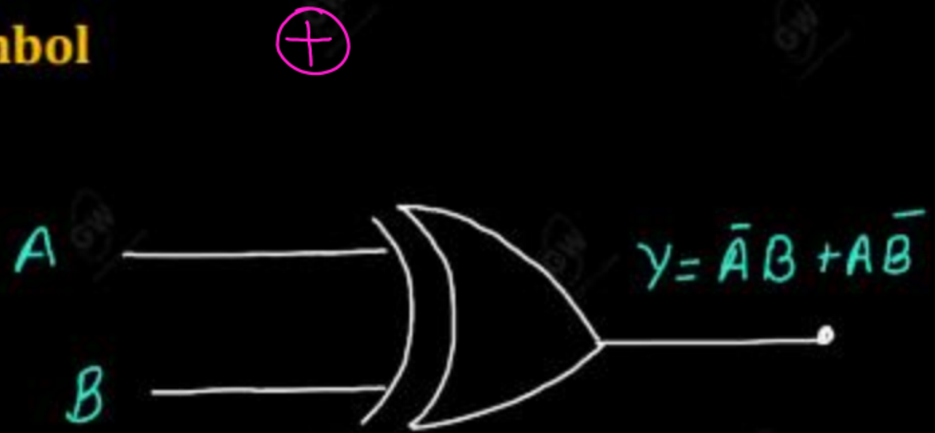
Truth Table

Input		Output	
A	B	OR	$Y = \overline{A + B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

(6) EXOR gate

- The EXOR gate is a circuit which will give a high output if both the input are different
- $\oplus$ is used to show EXOR operation

Symbol



Gateway Classes

Truth Table

Input		Output
A	A	$Y = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

(7) EX – NOR gate

- The EX-NOR gate is a circuit which will give a high output if both the input are same.
- $\odot$ is used to show EX-NOR gate.

Symbol



Gateway Classes

Truth Table

Input		Output
A	A	$Y = A \odot B$
0	0	1
0	1	0
1	0	0
1	1	1

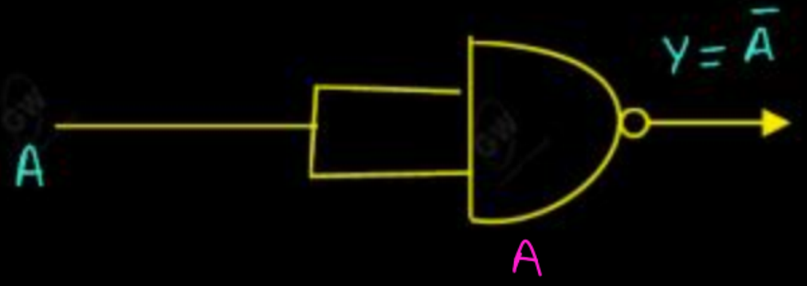
Universal Gates

- NAND and NOR gates are called as universal gates because it is possible to implement any Boolean expression with the help of only NAND gates or only NOR gates.
- A user can build any logical gates or combinational circuit with the help of only NAND gates or only NOR gates.
- NAND and NOR gates are economical and easier to fabricate and are the basic gates used in all IC digital logic families.

Gateway Classes

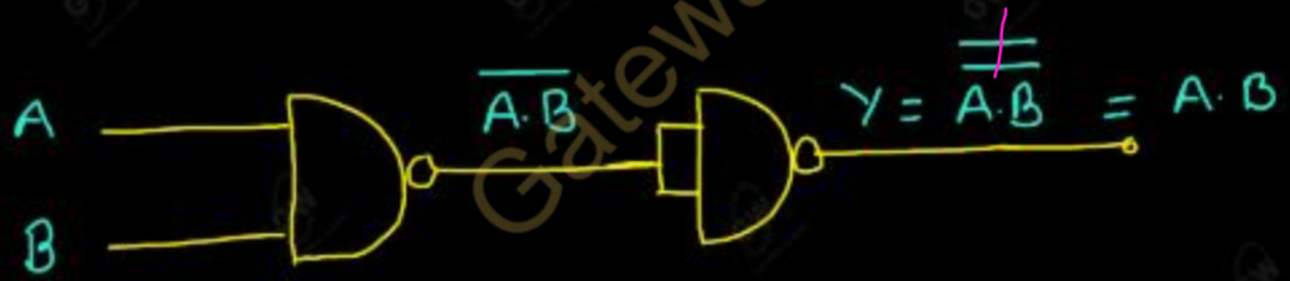
NAND Gate as a Universal Gate

(1) NOT gate using NAND gate



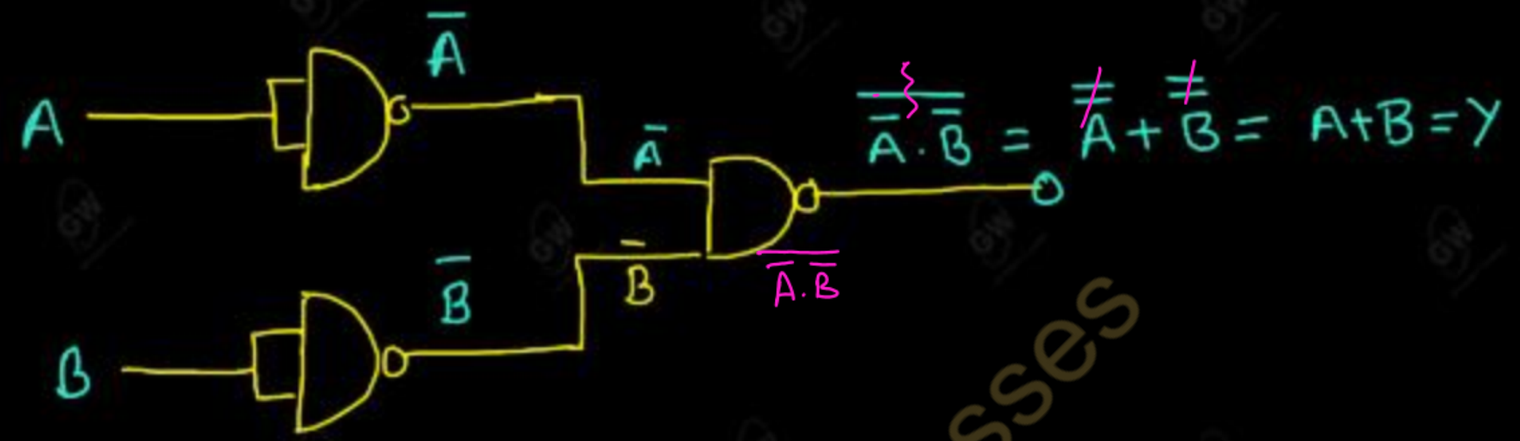
$A \cdot B$

(2) AND gate using NAND gate



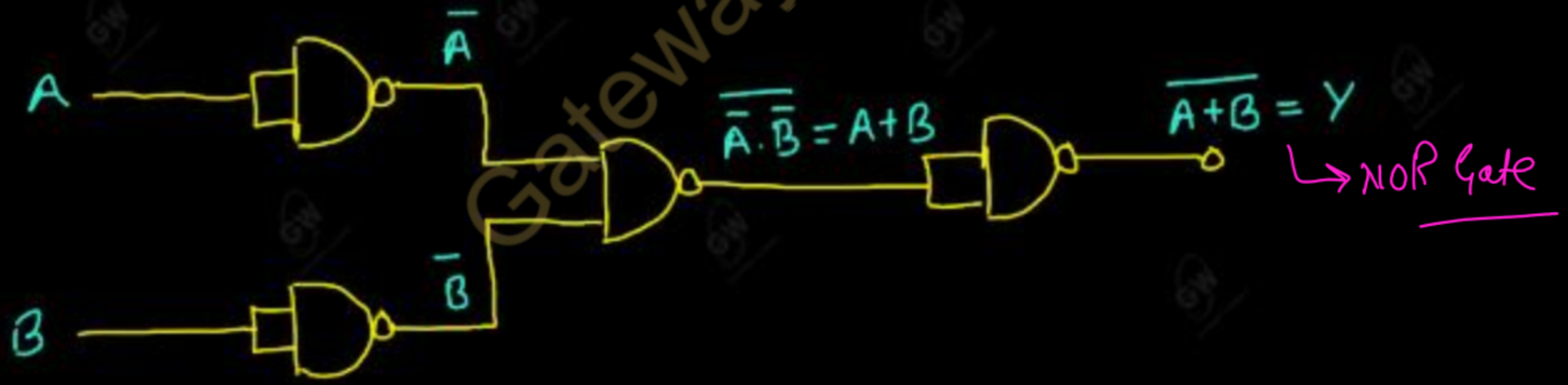
Gateway Classes

(3) OR gate using NAND gate



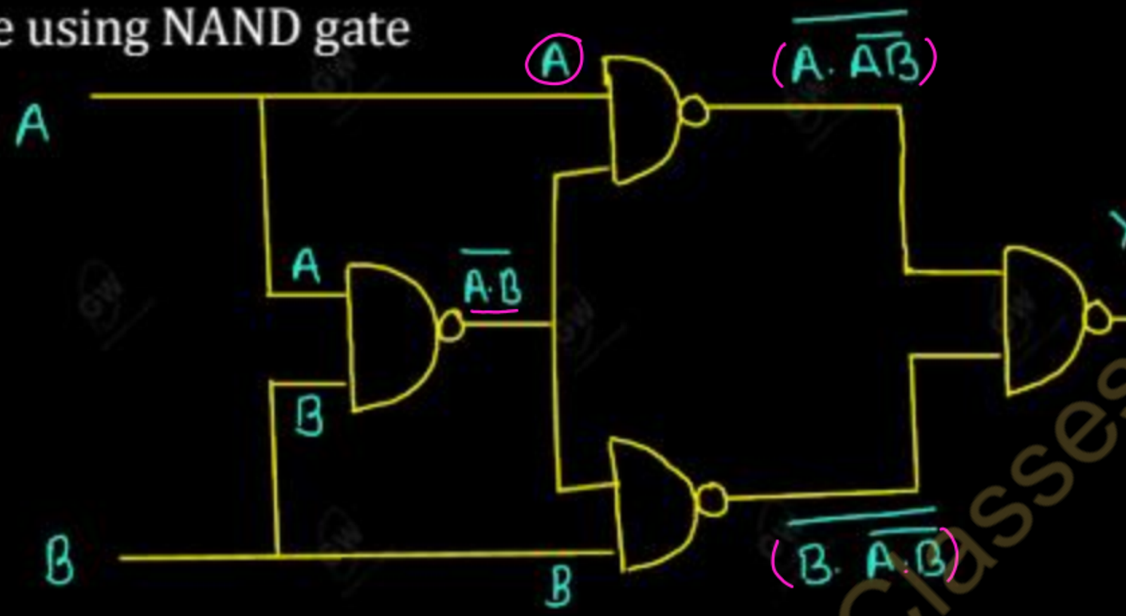
(4) NOR gate using NAND gate

AKTU-2024-25



Gateway Classes

(5) EXOR gate using NAND gate



$$Y = \overline{(A \cdot \overline{AB})} \cdot \overline{(B \cdot \overline{AB})} = \overline{\overline{A \cdot \overline{AB}}} + \overline{\overline{B \cdot \overline{AB}}}$$

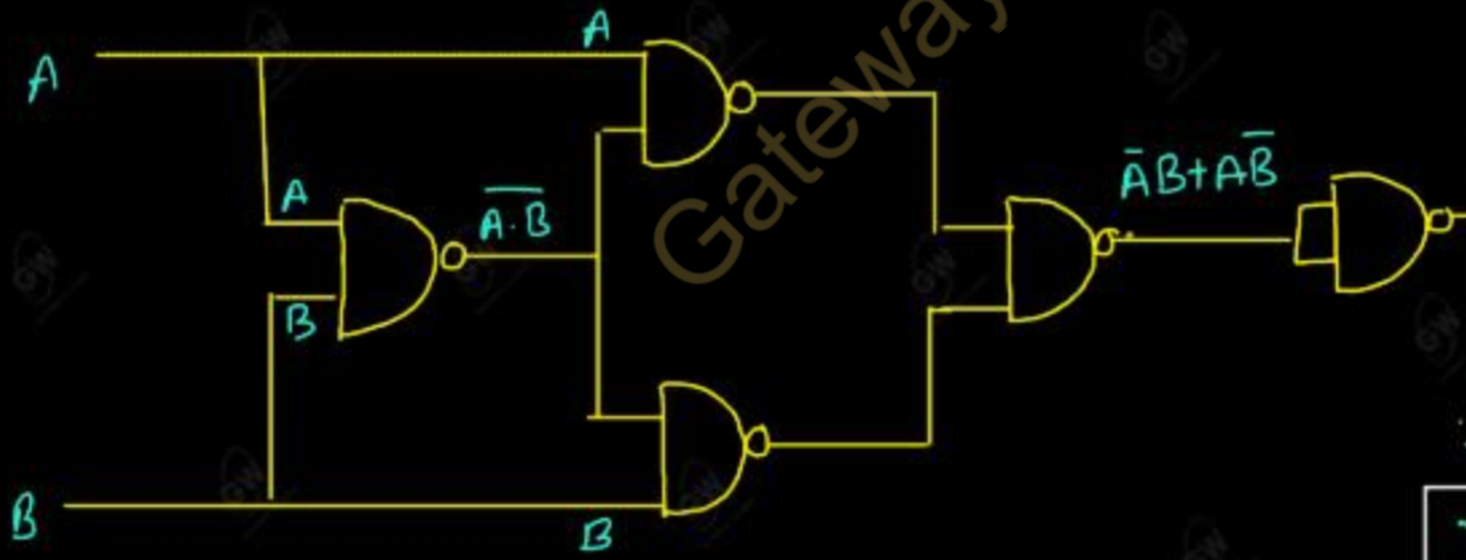
$$= A\overline{AB} + B\overline{AB}$$

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

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

$$Y = A\bar{B} + \bar{A}B \rightarrow \text{Ex-OR}$$

(6) EXNOR gate using NAND gate



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

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

$$Y = (A+\bar{B})(\bar{A}+B)$$

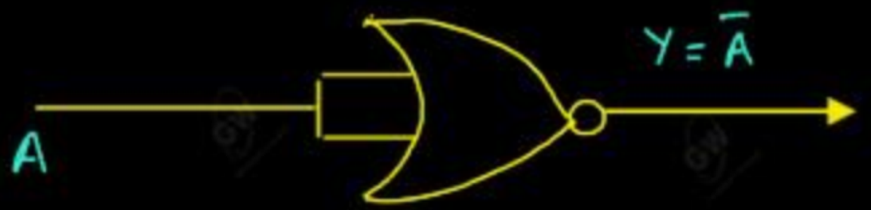
$$Y = A\bar{A} + AB + \bar{B}\bar{A} + B\bar{B} \rightarrow 0$$

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

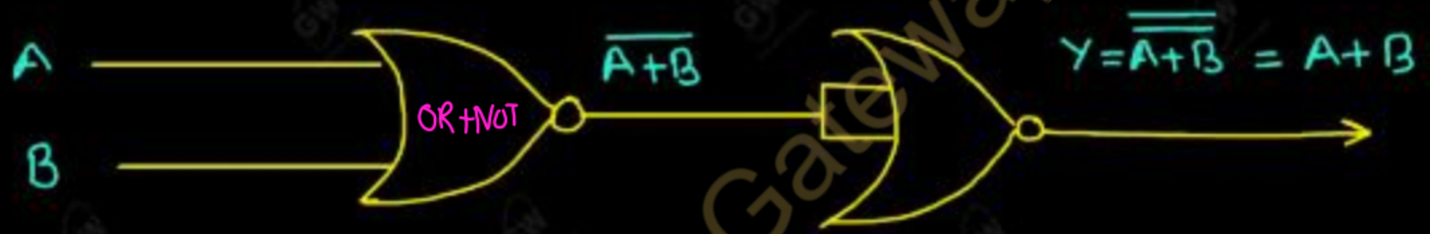
Gateway Classes

NOR gate as a universal gate

(1) NOT gate using NOR gate

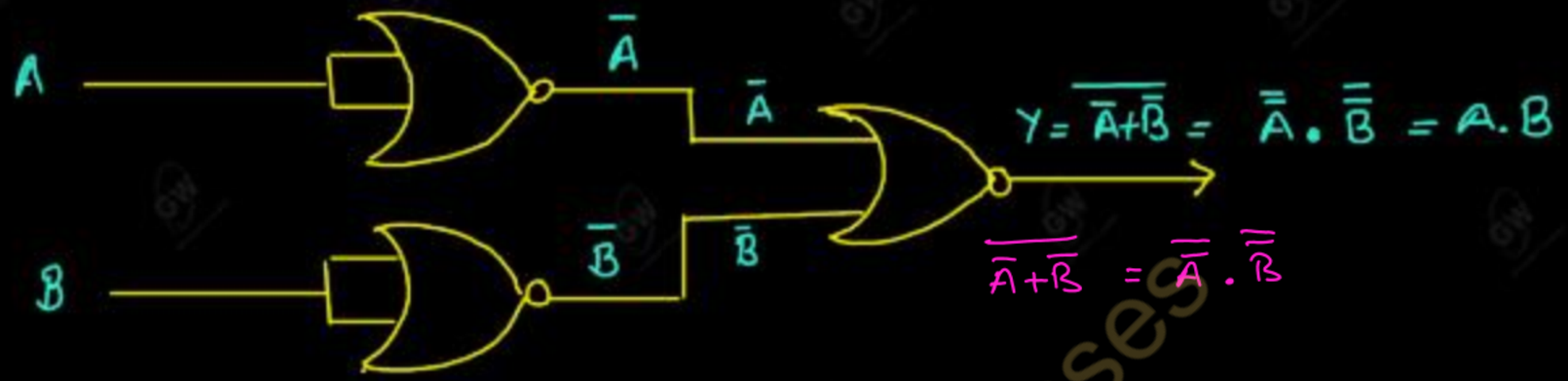


(2) OR gate using NOR gate



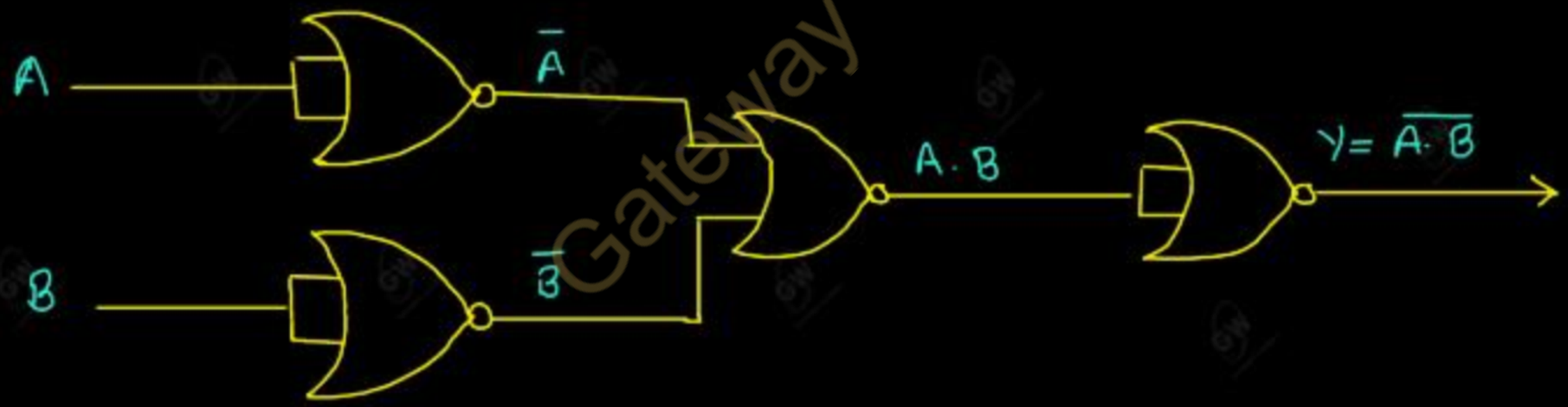
Gateway Classes

(3) AND gate using NOR gate



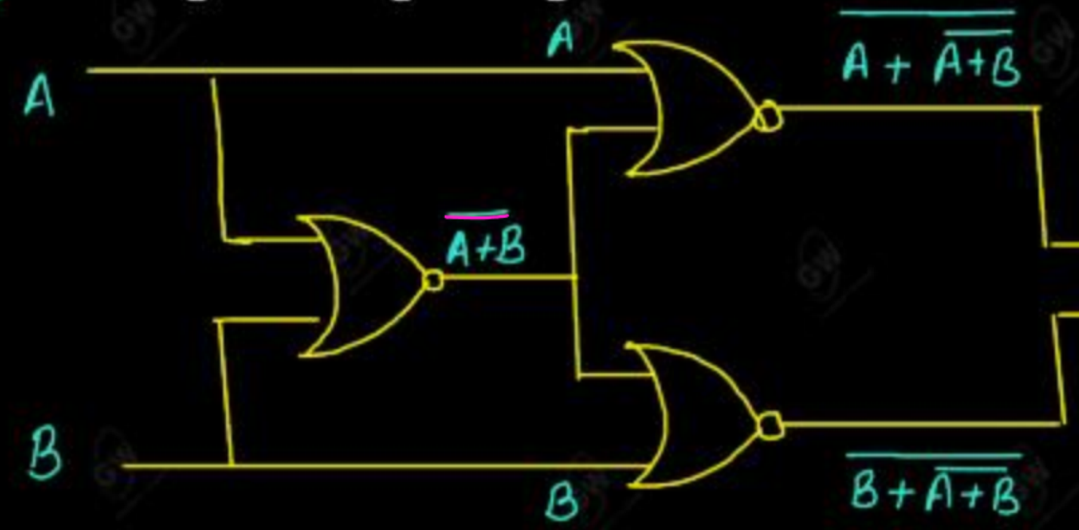
(4) NAND gate using NOR gate

$\overline{A \cdot B}$



Gateway Classes

(5) EX-OR gate using NOR gate



$$\overline{(A+\overline{A+B})+(B+\overline{A+B})}$$

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

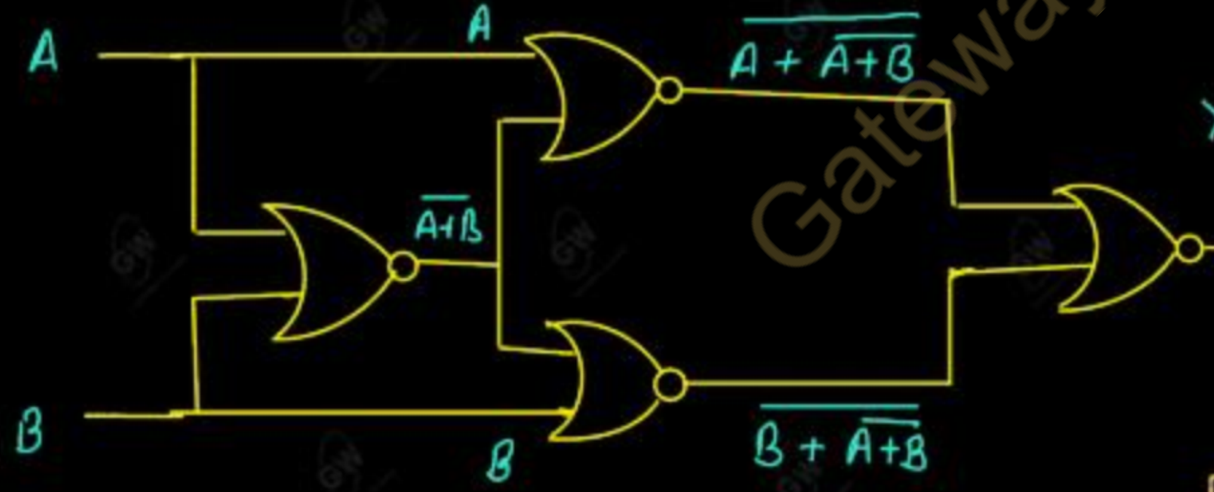
$$= \bar{A}\cdot\overline{A+B} + \bar{B}\cdot\overline{A+B}$$

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

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

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

(6) EXNOR gate using NOR gate



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

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

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

$$Y = (A+\bar{B})\cdot(\bar{A}+B) = A\bar{A} + AB + \bar{B}\bar{A} + B\bar{B}$$

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

Gateway Classes

Gateway Classes

Empowering Learners, Transforming Futures

GATEWAY CLASSES

Thank You

Gateway Classes

GET IT ON Google Play

Download on the App Store

Helpline No- 7819 0058 53