UPPSC-AE 2025

Uttar Pradesh
Public Service Commission

Combined State Engineering Services Examination

Assistant Engineer

Electrical Engineering

Digital Electronics

Well Illustrated **Theory** *with*
Solved Examples and **Practice Questions**

MADE EASY PUBLICATIONS

Digital Electronics

Contents

CHAPTER 1

Boolean Algebra and Logic Gates

1.1 Introduction

- The binary operations performed by any digital circuit with the set of elements 0 and 1, are called logical operations or logic functions. The algebra used to symbolically represent the logic function is called Boolean algebra. It is a two state algebra invented by George Boole in 1854.
- Thus, a Boolean algebra is a system of mathematics logic for the analysis and designing of digital systems.
- A variable or function of variables in Boolean algebra can assume only two values, either a '0' or a '1'. Hence, (unlike another algebra) there are no fractions, no negative numbers, no square roots, no cube roots, no logarithms etc.

1.2 Logic Operations

- In Boolean algebra, all the algebraic functions performed is logical. These actually represent logical operations. The AND, OR and NOT are the basic operations that are performed in Boolean algebra.
- In addition to these operations, there are some derived operation such as NAND, NOR, EX-OR, EX-NOR that are also performed in Boolean algebra.

1.2.1 AND Operation

The AND operation in Boolean algebra is similar to the multiplication in ordinary algebra. It is a logical operation performed by AND gate.

$$A \cdot A = A$$
$$A \cdot 0 = 0 \rightarrow \text{Null law}$$
$$A \cdot 1 = A \rightarrow \text{Identity law}$$
$$A \cdot \bar{A} = 0$$

1.2.2 OR Operation

The OR operation in Boolean algebra is performed by OR-gate.

$$A + A = A$$
$$A + 0 = A \rightarrow \text{Null law}$$
$$A + 1 = 1 \rightarrow \text{Identity law}$$
$$A + \bar{A} = 1$$

1.2.3 NOT Operation

- The NOT operation in Boolean algebra is similar to the complementation or inversion in ordinary algebra. The NOT operation is indicated by a bar (—) or (') over the variable.
- $A \xrightarrow{NOT} \bar{A}$ or A' (complementation law)

  and $\bar{\bar{A}} = A \Rightarrow$ double complementation law.

1.2.4 NAND Operation

The NAND operation in Boolean algebra is performed by AND operation with NOT operation i.e. the negation of AND gate operation is performed by the NAND gate.

1.2.5 NOR Operation

The NOR operation in Boolean algebra is performed by OR operation with NOT operation. i.e. the negation of OR gate operation is performed by the NOR gate.

1.2.6 EX-OR Operation

Unlike basic operations of logic gates, this used for special purpose and is represented by symbol '⊕' where

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

1.3 Laws of Boolean Algebra

The Boolean algebra is governed by certain well developed rules and laws.

1.3.1 Commutative Laws

- The commutative law allows change in position of AND or OR variables. There are two commutative laws.
  - (*i*) $A + B = B + A$
    Thus, the order in which the variables are ORed is immaterial.
  - (*ii*) $A \cdot B = B \cdot A$
    Thus, the order in which the variables are ANDed is immaterial.
- This law can be extended to any number of variables.

1.3.2 Associative Laws

- The associative law allows grouping of variables. There are two associative laws
  - (*i*) $(A + B) + C = A + (B + C)$
    Thus, the way the variables are grouped and ORed is immaterial.
  - (*ii*) $(A \cdot B) \cdot C = A \cdot (B \cdot C)$
    Thus, the way the variables are grouped and ANDed is immaterial.
- This law can be extended to any number of variables.

1.3.3 Distributive Laws

- The distributive law allows factoring or multiplying out of expressions. There are two distributive laws.
  (*i*) $A(B + C) = AB + AC$ (*ii*) $A + BC = (A + B)(A + C)$
- This law is applicable for single variable as well as a combination of variables.

1.3.4 Idempotence Laws

Idempotence means the same value. There are two Idempotence laws

(*i*) $A \cdot A = A$

i.e. ANDing of a variable with itself is equal to that variable only.

(*ii*) $A + A = A$

i.e. ORing of a variable with itself is equal to that variable only.

1.3.5 Absorption Laws

There are two absorption laws

(*i*) $A + AB = A(1 + B) = A$ (*ii*) $A(A + B) = A$

1.3.6 Involutionary Law

This law states that, for any variable 'A'

$$\bar{\bar{A}} = (A')' = A$$

1.4 Boolean Algebraic Theorems

1.4.1 De Morgan's Theorem

- These are very useful in simplifying expressions in which a product or sum of variables is inverted.
- De Morgan's theorem represents two of the most important rules of Boolean algebra.

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

Thus, the complement of the product of variables is equal to the sum of their individual complements.

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

Thus, the complement of a sum of variables is equal to the product of their individual complements.

- The above two laws can be extend for 'n' variables as

$$\overline{A_1 \cdot A_2 \cdot A_3 \cdots A_n} = \bar{A}_1 + \bar{A}_2 + \cdots + \bar{A}_n$$

and $$\overline{A_1 + A_2 + \cdots + A_n} = \bar{A}_1 \cdot \bar{A}_2 \cdot \bar{A}_3 \cdot \bar{A}_4 \cdots \bar{A}_n$$

1.4.2 Transposition Theorem

The transposition theorem states that $(AB + \bar{A}C) = (A + C)(\bar{A} + B)$

1.4.3 Consensus Theorem/Redundancy Theorem

- This theorem is used to eliminate redundant term.
- A variable is associated with some variable and its compliment is associated with some other variable and the next term is formed by the left over variables, then the term becomes redundant.
- It is applicable only if a Boolean function,

(*i*) Contains 3-variables.

(*ii*) Each variable used two times.

(*iii*) Only one variable is in complemented or uncomplemented form.

Then, the related terms to that complemented and uncomplemented variable is the answer.

- Consensus theorem can be extended to any number of variables.

  e.g. $AB + \bar{A}C + BC = AB + \bar{A}C$

Example - 1.1 The Boolean expression $(A + B)(\bar{B} + C)(C + A) = (A + B)(\bar{B} + C)$ can be simplified as

(a) $(A + B)(\bar{B} + C)$ (b) $(\bar{A} + B)(\bar{B} + C)$

(c) $(A + \bar{B})(\bar{B} + C)$ (d) $(A + \bar{B})(B + \bar{C})$

Solution : (a)

Proof:

$$\begin{aligned}\text{LHS} &= (A + B)(\bar{B} + C)(C + A)\\ &= (A\bar{B} + AC + BC)(C + A)\\ &= A\bar{B}C + AC + BC + A\bar{B} + AC + ABC\\ &= AC + BC + A\bar{B}\\ \text{RHS} &= (A + B)(\bar{B} + C)\\ &= A\bar{B} + AC + BC = \text{LHS}\end{aligned}$$

Example - 1.2 $\overline{\bar{A}\,\bar{B}\,\bar{C}}$ is equal to

(a) $\bar{A} + \bar{B} + \bar{C}$ (b) $\overline{ABC}$

(c) $A + B + C$ (d) $A \cdot B \cdot C$ [UPPSC]

Solution: (c)

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

1.4.4 Duality Theorem

It is one of the elegant theorems proved in advance mathematics.

"Dual expression" is equivalent to write a negative logic of the given Boolean relation. For this we have to

(i) change each **OR** sign by an **AND** sign and vice-versa.

(ii) complement any **'0'** or **'1'** appearing in expression.

(iii) keep literals/variables as it is.

Example - 1.3 The self dual expression of Boolean relation, $\bar{A}BC + AB\bar{C} + A\bar{B}\bar{C}$ is

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

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

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

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

Solution: (c)

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

Its DUAL

$$\Downarrow$$

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

NOTE

- For any logical expression, if we take two times Dual, we get same given expression as previous.
- For 1-*time Dual*, if we get same function or expression it is called "*Self Dual Expression*".
- With *N-variables*, maximum possible *Self-Dual Function* = $(2)^{2^{n-1}} = 2^{(2^n/2)}$.
- Remember that with *N-variables, maximum possible distinct logic functions* = 2^{2^n}.

1.4.5 Complementary Theorem

For obtaining complement expression we have to

(i) change each **OR** sign by **AND** sign and vice-versa.

(ii) complement any **'0'** or **'1'** appearing in expression.

(iii) complement the individual literals/variables.

Example - 1.4 The compliment of the function $f = A\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C}$ is equal to

(a) $(\bar{A}+B+\bar{C})(A+\bar{B}+C)(\bar{A}+B+C)$ (b) $(\bar{A}+\bar{B}+C)(A+\bar{B}+\bar{C})(\bar{A}+B+C)$

(c) $(\bar{A}+\bar{B}+C)(A+\bar{B}+C)(\bar{A}+\bar{B}+C)$ (d) $(\bar{A}+B+\bar{C})(A+\bar{B}+C)(\bar{A}+\bar{B}+\bar{C})$

Solution : (a)

$$\bar{f} = \text{Complement of f}$$

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

Example - 1.5 The Boolean expression $A(A + B)$ is equal to

(a) 1 (b) B

(c) A (d) A + B [UPPSC]

Solution: (c)

$$\begin{aligned} Y &= A(A + B) \\ &= A \cdot A + A \cdot B \\ &= A + AB \\ &= A(1 + B) \\ &= A \end{aligned}$$

Example - 1.6 The Boolean function $(x + y)(\bar{x} + z)(y + z)$ is equal to which one of the following expressions?

(a) $(x + y)(y + z)$ (b) $(\bar{x} + z)(y + z)$

(c) $(x + y)(\bar{x} + z)$ (d) $(x + y)(x + \bar{z})$

Solution: (c)

By using consensus theorem,

$$(x+y)(\bar{x}+z)(y+z) = (x+y)(\bar{x}+z)$$

Example - 1.7 The minimized form of the logical expression $(\bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC + AB\bar{C})$ is

(a) $\bar{A}\bar{C} + B\bar{C} + \bar{A}B$ (b) $A\bar{C} + \bar{B}C + \bar{A}B$

(c) $\bar{A}C + \bar{B}C + \bar{A}B$ (d) $A\bar{C} + \bar{B}C + A\bar{B}$

Solution: (a)

Given,

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

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

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

Example - 1.8 If X and Y are Boolean variables, which one of the following is the equivalent of X ⊕ Y ⊕ XY?

(a) $X + \bar{Y}$ (b) $X + Y$

(c) 0 (d) 1

Solution: (b)

Let,

$$Z = X \oplus Y \oplus XY$$

$$Z = X \oplus \left[\bar{Y}(XY) + Y(\overline{XY})\right]$$

$$= X \oplus \left[Y(\bar{X} + \bar{Y})\right] = X \oplus \left[(\bar{X}Y)\right]$$

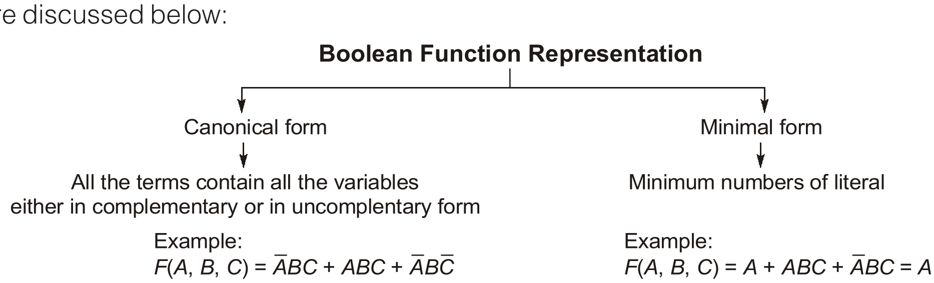
$$= \bar{X}(\bar{X}Y) + X(\overline{\bar{X}Y}) = \bar{X}Y + X(X + \bar{Y})$$

$$= \bar{X}Y + X + X\bar{Y} = \bar{X}Y + X(1 + \bar{Y})$$

$$= \bar{X}Y + X = (X + \bar{X})(X + Y) = (X + Y)$$

1.5 Representation of Boolean Functions

- A function of 'n' Boolean variables denoted by $f(A_1, A_2, \cdots A_n)$ is another variable of algebra and takes one of the two possible values either 0 or 1. The various ways of representing a given function are discussed below:



- The term 'literal' means a binary variable either in complementary or in uncomplimentary form.

1.5.1 Minterms and Maxterms

- n-binary variables have 2^n possible combinations.
- Minterm is a product term, it contains all the variables either complementary or uncomplimentary form for that combination the function output must be '1'.
- Maxterm is a sum term, it contains all the variables either complementary or uncomplimentary form for that combination the function output must be '0'.

NOTE

- In "Minterms" we assign '1' to each uncomplemented variable and '0' to each complemented variable.
- In "Maxterms" we assign '0' to each uncomplemented variable and '1' to each complemented variable.

1.5.2 Sum of Product (SOP) Form

- The SOP expression usually takes the form of two or more variables ANDed together. Each product term may be minterm or implicant.

$$Y = \bar{A}BC + A\bar{B} + AC$$

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

- This form is also called the "disjunctive normal form".
- The SOP expression is used most often because it tends itself nicely to the development of truth tables and timing diagrams.
- SOP circuits can also be constructed easily by using a special combinational logic gates called the "AND-OR-INVERTER" gate.
- SOP forms are used to write logical expression for the output becoming Logic '1'.

Input (3-Variables)			Output (*Y*)
A	*B*	*C*	*Y*
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$\therefore$ Notation of SOP expression is,

$$f(A, B, C) = \Sigma m(3, 5, 6, 7)$$

$\therefore$ $$Y = m_3 + m_5 + m_6 + m_7$$

also, $$Y = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

1.5.3 Product of Sum (POS) Form

- The POS expression usually takes the form of two or more order variables within parentheses, ANDed with two or more such terms.

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

- This form is also called the "Conjunctive normal form".
- Each individual term in standard *POS* form is called *Maxterm*.
- *POS* forms are used to write logical expression for output be coming *Logic* '0'.

From the above truth table, we get

$$F(A, B, C) = \Pi M(0, 1, 2, 4)$$

$\therefore$ $$Y = M_0 \times M_1 \times M_2 \times M_4$$

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

- We also conclude that From the above truth table, and from above equations

  if, $Y = \Sigma m(3, 5, 6, 7)$

  then, $Y = \Pi M(0, 1, 2, 4)$

1.5.4 Standard Sum of Product Form

- In this form the function is the sum of number of product terms where each product term contains all the variables of the function, either in complemented or uncomplemented form.
- It is also called canonical SOP form or expanded SOP form.
- The function $[Y = A + B\bar{C}]$ can be represented in canonical form as:

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

1.5.5 Standard Product of Sum Form

- This form is also called canonical POS form or expanded POS form.

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

- Then, the canonical form of the given function

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

1.5.6 Truth Table Form

A truth table is a tabular form representation of all possible combinations of given function.

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

Then,

	A	B	C	Y
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	0

Example - 1.9 Simplify the expression for the following truth table.

A	B	Y
0	0	1
0	1	0
1	0	1
1	1	0

is equal to

(a) $\bar{A}$ (b) $\bar{A} + B$

(c) $\bar{B}$ (d) $\bar{A} + \bar{B}$

Solution : (c)

$$\Rightarrow \quad Y = \bar{A}\,\bar{B} + A\bar{B} \quad \Rightarrow Y = \bar{B}$$

Example - 1.10 Simplify the expression $Y(A, B) = \Pi M(1, 3)$.

(a) $\bar{A}$ (b) $\bar{B}$

(c) $\bar{A} + B$ (d) $\bar{A} + \bar{B}$

Solution : (b)

$$(1)_{10} = (01)_2 = \bar{A}B\,(SOP) = A + \bar{B}\,(\text{POS})$$

$$(3)_{10} = (11)_2 = AB \cdot (SOP) = \bar{A} + \bar{B}\,(\text{POS})$$

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

1.5.7 Dual Form

- Dual form is used to convert positive logic to the negative logic and vice-versa.
- In positive logic system, higher voltage is taken as logic '1' and in negative logic system, higher voltage is taken as logic '0'. For example.

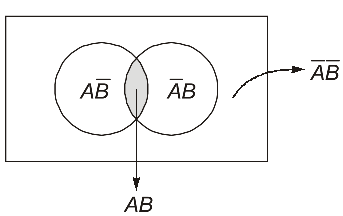
  (i) For (positive) logic

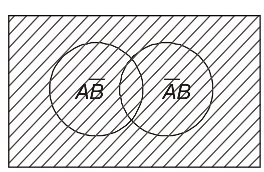
  Logic '1' = 0 V; Logic '0' = –5 V

  (ii) For (negative) logic

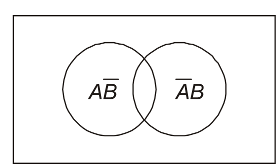
  Logic '1' = –0.8 V; Logic '0' = –1.7 V

1.5.8 Venn Diagram Form

- A Boolean algebra can be represented by a Venn diagram in which each variable is considered as a set.
- The AND operation is considered as an intersection and the OR operation is considered as a union.
- 

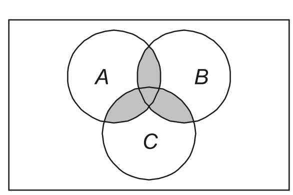


(1) $Y = 1$



(2) $Y = 0$

Example - 1.11 For the given Venn diagram the minimize the expression for the shade area is represented



(a) $\bar{A}B + BC + \overline{AC}$ (b) $\bar{A}B + \bar{B}C + AC$

(c) $\bar{A}B + \bar{B}C + \bar{A}C$ (d) $AB + BC + AC$

Solution : (d)

The shaded area includes

$AB\bar{C}, A\bar{B}C, \bar{A}BC, ABC$

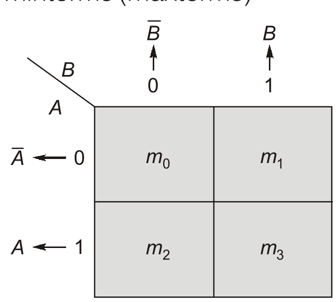
$$\therefore \quad Y = ABC + AB\bar{C} + A\bar{B}C + \bar{A}BC$$
$$= AB + AC(B + \bar{B}) + BC(A + \bar{A})$$
$$= AB + BC + AC$$

1.6 Karnaugh Map

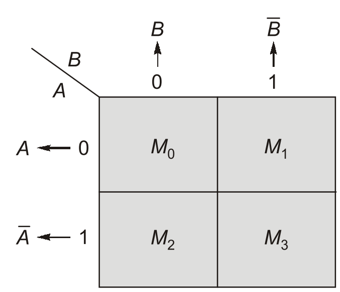
- The "Karnaugh map" is a graphical method which provides a systematic method for simplifying and manipulating the Boolean expressions or to convert a truth table to its corresponding logic circuit in a simple, orderly process.
- In this technique, the information contained in a truth table or available in SOP or POS form is represented on K-map.
- Although this technique may be used for any number of variables, it is generally used up to 6-variables beyond which it becomes very cumbersome.
- In n-variable *K*-map there are 2^n cells.
- "Gray code" has been used for the identification of cells.

1.6.1 Two-variable K-Map

- Four cells
- Four minterms (maxterms)



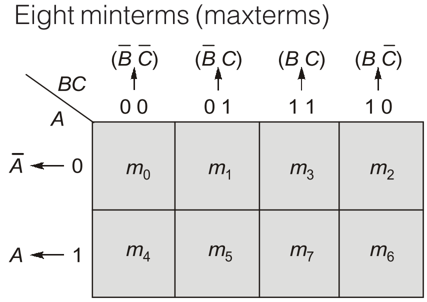
(For SOP)



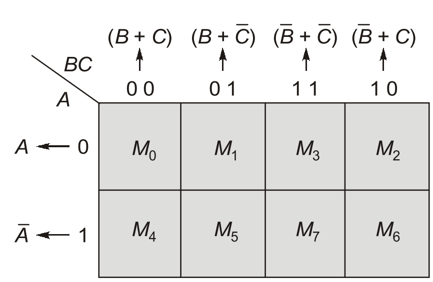
(For POS)

1.6.2 Three-variable K-map

- Eight cells
- Eight minterms (maxterms)



(For SOP)



(For POS)

1.6.3 Four-variable K-map

- Sixteen cells
- Sixteen minterms (maxterms)

Student's Assignments

Q.1 The logic expression $F = (\bar{A}+C)(B+C)(A+B)$ is equal to

(a) $(\bar{A}+B)(A+C)$ (b) $(A+B)(\bar{A}+C)$

(c) $(A+\bar{B})(\bar{A}+C)$ (d) $(A+\bar{B})(\bar{A}+\bar{C})$

Q.2 Which of the following statements is **not** correct?

(a) Disjunction of a variable with logic '0' results in the same variable.

(b) NOR function can be implemented by inverting the two inputs to an AND function.

(c) Conjunction of a variable with logic '1' results in the complement of variable.

(d) NAND function is commutative but not associative.

Q.3 The *K*-map of a function is given below:

X_1X_4 \ X_2X_3	00	01	11	10
00	1		x	x
01		1	x	1
11		x	1	
10	1	x		x

If "X" indicates don't care condition, then the minimized expression for the function is

(a) $X_3\bar{X}_1 + \bar{X}_1\bar{X}_3 + X_3X_2$

(b) $X_3X_2 + X_4\bar{X}_3 + X_2\bar{X}_1$

(c) $\bar{X}_4\bar{X}_3 + X_4X_3 + X_2\bar{X}_1$

(d) $\bar{X}_4\bar{X}_3 + X_4X_3 + X_3\bar{X}_1$

Q.4 In negative logic system

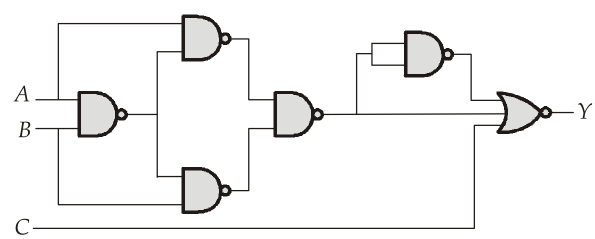
(a) The more negative of the two logic levels represents a logic '1' state.

(b) The more negative of the two logic levels represents a logic '0' state.

(c) All input and output voltage levels are negative.

(d) None of the above

Q.5 Consider the following circuit:



If the value of C is stuck at '0', then the output Y will be

(a) $A\bar{B}$ (b) 0

(c) 1 (d) $\bar{A}$

Q.6 The Boolean expression

$$X(P, Q, R) = \Pi(0, 5)$$

is to be realized using only two 2-input gates. Which are these gates?

(a) AND and OR (b) NAND and OR

(c) AND and XOR (d) OR and XOR

Q.7 In Boolean algebra if $F = (A + B)(\bar{A} + C)$, then

(a) $F = AB + \bar{A}C$ (b) $F = AB + \overline{AB}$

(c) $F = AC + \bar{A}B$ (d) $F = AA + \bar{A}B$

Q.8 The POS form of expression is suitable for circuit using:

(a) NOR (b) AND

(c) XOR (d) NAND

Q.9 Negative logic in a logic circuit is one in which

(a) logic 0 and 1 are represented by the negative and zero voltages respectively.

(b) logic 0 and 1 are represented by zero and positive voltages respectively.

(c) logic 0 voltage level is lower than logic 1 voltage level.

(d) logic 0 voltage level is higher than logic 1 voltage level.

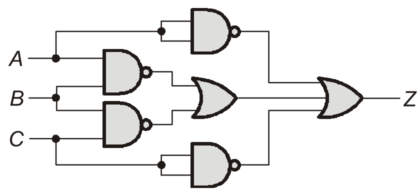
Q.10 The reduced form of Boolean expression

$$A\left[(B+C)\overline{(AB+AC)}\right] \text{ is}$$

(a) $\bar{A}B$ (b) $A\bar{B}$

(c) 0 (d) $AB + B\bar{C}$

Q.11 In the given combinational circuit, the output Z is



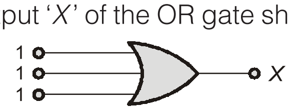
(a) $\overline{A}+\overline{B}+\overline{C}$ (b) $\overline{ABC}$
(c) $\overline{AB}+\overline{BC}+\overline{AC}$ (d) All of the above

Q.12 The number of terms in a logic function of three variables X, Y, Z are
(a) 3 (b) 4
(c) 8 (d) 16 **[UPPSC]**

Q.13 How many different Boolean expressions can be constructed by using 2 variables?
(a) 18 (b) 16
(c) 32 (d) 4

Q.14 With 4 Boolean variables, how many Boolean expressions can be formed?
(a) 16 (b) 256
(c) 1024 (1K) (d) 64 K(64×1024)

Q.15 The output 'X' of the OR gate shown in figure is



(a) 1 (b) 2
(c) 0 (d) 3 **[UPPSC]**

Q.16 The most suitable gate for comparing two bit is
(a) AND gate (b) OR gate
(c) NAND gate (d) EX-OR gate
[UPPSC]

Q.17 The minimum number of NOR gates required to implement $A(A + B)(A + B + C)$ is equal to
(a) 0 (b) 3
(c) 4 (d) 7 **[UPPSC]**

ANSWER KEY — STUDENT'S ASSIGNMENTS

1. (b)	**2.** (c)	**3.** (c)	**4.** (a)	**5.** (b)
6. (d)	**7.** (c)	**8.** (a)	**9.** (d)	**10.** (c)
11. (d)	**12.** (c)	**13.** (b)	**14.** (d)	**15.** (a)
16. (d)	**17.** (a)			

HINTS & SOLUTIONS — STUDENT'S ASSIGNMENTS

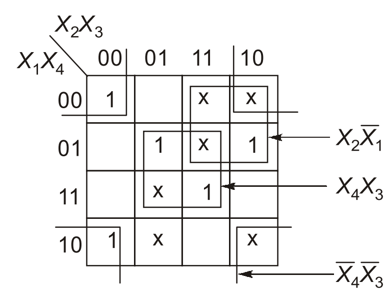
1. (b)

$$F = (\overline{A}+C)(B+C)(A+B)$$
$$= (\overline{A}+C)(A+B)$$
(By consensus theorem)

2. (c)

Conjunction of a variable with logic '1' results in the same variable.

3. (c)

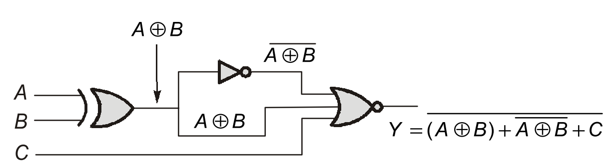


4. (a)

In negative logic system, the more negative of the two logic levels represents a logic '1' state.

5. (b)

The circuit can be redrawn as



$$Y = \overline{(A \oplus B) + \overline{A \oplus B} + C} = \overline{1 + C} = 0$$

6. (d)

The Boolean expression

$$X(P, Q, R) = \pi(0, 5)$$
$$= (P+Q+R)(\overline{P}+Q+\overline{R})$$
$$= Q + (\overline{P}R + P\overline{R})$$

8. (a)

POS form (Product of Sum form) of expression is suitable for circuit using NOR gates.