Sanchit Jain

Legend

I am a computer science faculty, having a teaching experience of more than 8 years. I have a very sound experience in teaching students.

43M Watch mins | **3M** Watch mins (last 30 days) | **20K** Followers | **2K** Dedications

Educator highlights

- Works at **Knowledge Gate**
- Studied at **Delhi Technological University**
- Have qualified gate Have experience of more than 8 years Have a YouTube channel with 5 lakh follower
- Lives in **Ghaziabad, Uttar Pradesh, India**
- Unacademy Educator since **19th July, 2019**
- **844,388 live minutes** taught in last 30 days
- Knows **Punjabi, Hinglish, Hindi and English**

Months Wise Calendar for GATE-2022 (Unacademy Plus) (Sanchit Jain)

Month	Morning Class (7:00am – 9:00am)	Evening Class (7:00pm – 9:00pm)
Feb-2021	DS & Programming (15-Feb to 23-Mar) (56 Hours)	DM (22-Feb to 27-Mar) (50 Hours)
Mar-2021	Algo (24-Mar to 26-April) (38 Hours)	DBMS (29-Mar to 30-April) (50 Hours)
Apr-2021	CN(29-April to 15-June) (54 Hours)	DE (7-May to 1-June) (40 Hours)
May-2021	TOC (26-May to 05-July) (58 Hours)	COA (23-June to 20-July) (38 Hours)
Jun-2021	Compiler (7-July to 5-Aug) (34 Hours)	OS (21-July to 24-Aug) (58 Hours)
Jul-2021		
Aug-2021	DM (11-Aug to 20-Sept) (50 Hours)	**DS & Programming (25-Aug to 5-Oct) (56 Hours)**
Sep-2021	DBMS (22-Sept to 29-Oct) (50 Hours)	Algo (6-Oct to 1-Nov) (38 Hours)
Oct-2021	**DE (3-Nov to 23-Nov) (40 Hours)**	**CN(10-Nov to 5 Jan) (54 Hours)**
Nov-2021	COA (24-Nov to 13-Dec) (38 Hours)	Compiler (26-Jan to 15-Feb) (34 Hours)
Dec-2021	OS (15-Dec to 11-Jan) (58 Hours)	
Jan-2022	TOC (12-Jan to 12-Feb) (58 Hours)	
Feb-2022		

Date	Day	Subject-1	Topic	Time	Duration
03-11-2021	Wednesday	Digital Electronics	Boolean Algebra & Expression Part-1	7:00 am to 9:00 am	2
04-11-2021	Thursday	Digital Electronics	Boolean Algebra & Expression Part-2	7:00 am to 9:00 am	2
05-11-2021	Friday				
06-11-2021	Saturday				
07-11-2021	Sunday				
08-11-2021	Monday	Digital Electronics	Simplication of Boolean Algebra Part-1	7:00 am to 9:00 am	2
09-11-2021	Tuesday	Digital Electronics	Doubt Clearing Session	7:00 am to 9:00 am	2
10-11-2021	Wednesday	Digital Electronics	Simplication of Boolean Algebra Part-2	7:00 am to 9:00 am	2
11-11-2021	Thursday	Digital Electronics	Simplication of Boolean Algebra Part-3	7:00 am to 9:00 am	2
12-11-2021	Friday	Digital Electronics	Combinational Circuits Part-1	7:00 am to 9:00 am	2
13-11-2021	Saturday	Digital Electronics	Doubt Clearing Session	7:00 am to 9:00 am	2
14-11-2021	Sunday	Digital Electronics	Combinational Circuits Part-2	7:00 am to 9:00 am	2
15-11-2021	Monday	Digital Electronics	Combinational Circuits Part-3	7:00 am to 9:00 am	2
16-11-2021	Tuesday	Digital Electronics	Sequential Circuits Part-1	7:00 am to 9:00 am	2
17-11-2021	Wednesday	Digital Electronics	Doubt Clearing Session	7:00 am to 9:00 am	2
18-11-2021	Thursday	Digital Electronics	Sequential Circuits Part-2	7:00 am to 9:00 am	2
19-11-2021	Friday	Digital Electronics	Sequential Circuits Part-3	7:00 am to 9:00 am	2
20-11-2021	Saturday	Digital Electronics	Number System Part-1	7:00 am to 9:00 am	2
21-11-2021	Sunday	Digital Electronics	Doubt Clearing Session	7:00 am to 9:00 am	2
22-11-2021	Monday	Digital Electronics	Number System Part-2	7:00 am to 9:00 am	2
23-11-2021	Tuesday	Digital Electronics	Floating Point Repreentation	7:00 am to 9:00 am	2

Date	Day	Subject-2	Topic	Time	Duration
10-11-2021	Wednesday	CN	Basics of Computer Networks Part-1	6:00 pm to 8:00 pm	2
11-11-2021	Thursday	CN	Basics of Computer Networks Part-2	6:00 pm to 8:00 pm	2
15-11-2021	Monday	CN	Basics of Computer Networks Part-3	6:00 pm to 8:00 pm	2
16-11-2021	Tuesday	CN	Doubt Clearing Session	6:00 pm to 8:00 pm	2
17-11-2021	Wednesday	CN	Access Control Part-1	6:00 pm to 8:00 pm	2
18-11-2021	Thursday	CN	Access Control Part-2	6:00 pm to 8:00 pm	2
22-11-2021	Monday	CN	Flow Control Part-1	6:00 pm to 8:00 pm	2
23-11-2021	Tuesday	CN	Doubt Clearing Session	6:00 pm to 8:00 pm	2
24-11-2021	Wednesday	CN	Error Control Part-1	6:00 pm to 8:00 pm	2
25-11-2021	Thursday	CN	Error Control Part-2	6:00 pm to 8:00 pm	2
29-11-2021	Monday	CN	Framing and Ethernet	6:00 pm to 8:00 pm	2
30-11-2021	Tuesday	CN	Doubt Clearing Session	6:00 pm to 8:00 pm	2
01-12-2021	Wednesday	CN	Network Layer protocol Part-1	6:00 pm to 8:00 pm	2
02-12-2021	Thursday	CN	Network Layer protocol Part-2	6:00 pm to 8:00 pm	2
06-12-2021	Monday	CN	Network Layer protocol Part-3	6:00 pm to 8:00 pm	2
07-12-2021	Tuesday	CN	Doubt Clearing Session	6:00 pm to 8:00 pm	2
13-12-2021	Monday	CN	IP addressing and routing and Routing Part-1	6:00 pm to 8:00 pm	2
14-12-2021	Tuesday	CN	IP addressing and routing and Routing Part-2	6:00 pm to 8:00 pm	2
20-12-2021	Monday	CN	Transport Layer Part-1	6:00 pm to 8:00 pm	2
21-12-2021	Tuesday	CN	Doubt Clearing Session	6:00 pm to 8:00 pm	2
27-12-2021	Monday	CN	Transport Layer Part-2	6:00 pm to 8:00 pm	2
28-12-2021	Tuesday	CN	Transport Layer Part-3	6:00 pm to 8:00 pm	2
03-01-2022	Monday	CN	Application Layer	6:00 pm to 8:00 pm	2
04-01-2022	Tuesday	CN	Doubt Clearing Session	6:00 pm to 8:00 pm	2
05-01-2022	Wednesday	CN	crypto	6:00 pm to 8:00 pm	2

GATE & ESE | Plus | All courses | All languages | All topics

HINDI CS & IT

Complete Course on Digital Electronics for GATE

Lesson 3 • Today, 7:00 AM

Sanchit Jain

HINGLISH PRACTICE & STRATEGY

FAQ Handbook for GATE (CS & IT)

Lesson 7 • Nov 10, 2021 8:31 PM

Ravindrababu Ravula

HINDI CS & IT

Complete Course on Algorithms for GATE

Lesson 18 • Today, 6:05 PM

Sanchit Jain

HINGLISH CS & IT

Complete Course on Computer Networks for GATE

Starts on Nov 10, 2021 • 26 lessons

Sanchit Jain

HINGLISH CS & IT

Complete Course on DBMS for GATE

Ended on Oct 28, 2021 • 25 lessons

Sanchit Jain

HINDI CS & IT

Complete Course on Programming & Data Structures for GATE

Ended on Oct 12, 2021 • 28 lessons

Sanchit Jain

HINDI CS & IT

Complete Course on Discrete Mathematics for GATE

Ended on Oct 10, 2021 • 27 lessons

Sanchit Jain

HINDI CS & IT

Foundation Course on Operating Systems for GATE 2022

Ended on Aug 23, 2021 • 30 lessons

Sanchit Jain

HINDI CS & IT

Complete Course on Compiler Design for GATE

Ended on Aug 13, 2021 • 18 lessons

Sanchit Jain

HINDI CS & IT

Complete Course on Computer Architecture for GATE 2022

Ended on Jul 20, 2021 • 19 lessons

Sanchit Jain

HINGLISH CS & IT

Foundation Course on Theory of Computation for GATE 2022

Ended on Jul 5, 2021 • 29 lessons

Sanchit Jain

HINDI CS & IT

Foundation Course on Digital Electronics

Ended on Jun 1, 2021 • 20 lessons

Sanchit Jain

Simplification of Boolean Algebra - Part III

Complete Course on Digital Electronics for GATE

Sanchit Jain • Lesson 6 • Nov 11, 2021

Boolean Expressions

- Boolean expressions are the method using which we save the information about the Boolean function, that when we get value 1 and when we get value 0 as output.

- So, we convert the truth table of the function into an expression, reading which we can understand where the output is 1 and when it is zero.

Light	Day	Engine	Warning
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

- $W = a'b'c + ab'c' + abc' + abc$

- There are two popular approaches for writing these expressions.
  - Sum of Product (SOP) (which remember when we get 1)
  - Product of Sum (POS) (which remember when we get 0)

- Make a note of this as we are studying Boolean function remembering both 0 and 1 is not required, so we can either concentrate on 0 or 1.

Light	Day	Engine	Warning
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

- $W(L, D, E) = \sum_m (1, 4, 6, 7)$
- $W(L, D, E) = \prod_M (0, 2, 3, 5)$

- Power of SOP and POS bother are same, i.e. any Boolean functions can be represented using both SOP and POS.

- Mathematically speaking we should choose (0 or 1), whatever is less in number because then it will be easy to represent

A	B	$f_1 =$	$f_2 =$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

Q The truth table represents the Boolean function **(GATE-2012) (1 Marks)**

(A) X **(B)** X+Y **(C)** $X \oplus Y$ **(D)** Y

X	Y	F (X, Y)
0	0	0
0	1	0
1	0	1
1	1	1

Break

SOP (sum of product)

- A sum of product form expression contains product terms (AND terms) which are sum (OR) together, that's why called sum of product.

- Each product terms (AND terms) consists of one or more literals (variables) appearing either in complements or uncomplemented form. E.g. a'b + b'c' + abc

- A product term which contains all the literals (variables) either in complemented or uncomplemented form is called minterm.

- In a n variable function, there will be 2^n minterms.

0 ⟷ 7

1 ⟷ 1

2 ⟷ 5

3 ⟷ 4

Binary Representation	Sequence	Minterm	Designation
000	0	a'b'c'	m_0
001	1	a'b'c	m_1
010	2	a'bc'	m_2
011	3	a'bc	m_3
100	4	ab'c'	m_4
101	5	ab'c	m_5
110	6	abc'	m_6
111	7	Abc	m_7

- The result of a product term must always be 1, If a literal is having value 1 then it is ok, but if not, then we complement those which is 0, it to make it 1.

- There is only 1 input sequence of variables for any minterm on which the output is 1, so it represents information.

- Then the sum of all product term(minterm) to from a function, and functions will have value 1, if at least one of the product term(minterm) is 1.

Binary Representation	Sequence	Minterm	Designation
000	0	a'b'c'	m_0
001	1	a'b'c	m_1
010	2	a'bc'	m_2
011	3	a'bc	m_3
100	4	ab'c'	m_4
101	5	ab'c	m_5
110	6	abc'	m_6
111	7	Abc	m_7

Canonical logic forms

- Either in POS or SOP form it is not essential that all product or sum terms contains all the literals.

- **Canonical SOP form**: - In a sum of product form expression, if each AND term (product term) consists all the literals(variables) appearing either in complements or uncomplemented form. E.g. a'bc + ab'c' + abc. Then the form is said to be canonical SOP.

Q The minterm expansion of f(P, Q, R) = PQ + QR' + PR' is **(GATE-2010) (2 Marks)**

(A) $m_2 + m_4 + m_6 + m_7$

(B) $m_0 + m_1 + m_3 + m_5$

(C) $m_0 + m_1 + m_6 + m_7$

(D) $m_2 + m_3 + m_4 + m_5$

Q Consider a function and find no of minterms $F(a, b, c) = a + a'b + abc + bc'$?

Break

POS (Product of Sum)

- A product of sum (POS) form of expression contains OR (sum) terms which are AND (product) together, that's why called product of sum expression.

- Each OR term (sum term) consists of one or more literals(variables) appearing either in complemented or uncomplemented form. (a' + b). (b' + c'). (a + c)

- A OR (sum) term which contains all the literals(variables) either in complemented or uncomplemented form is called maxterm. In a n variable function, there will be 2^n maxterms.

Binary Representation	Sequence	Maxterm	Designation
000	0	a + b + c	M_0
001	1	a + b + c'	M_1
010	2	a + b' + c	M_2
011	3	a + b' + c'	M_3
100	4	a' + b + c	M_4
101	5	a' + b + c'	M_5
110	6	a' + b' + c	M_6
111	7	a' + b' + c'	M_7

- The result of the OR (sum) term must be 0, so If a variable is having value 0 then it is ok, but if not then we complement the variable it to make it 0.

- There is only 1 input sequence for which the output of a Maxterm is 0. Because, it requires all values as zero.

- Then the product of all sum term (maxterm), from a function, and functions will have a value 0, if any of the sum term(maxterm) is 0.

Binary Representation	Sequence	Maxterm	Designation
000	0	a + b + c	M_0
001	1	a + b + c'	M_1
010	2	a + b' + c	M_2
011	3	a + b' + c'	M_3
100	4	a' + b + c	M_4
101	5	a' + b + c'	M_5
110	6	a' + b' + c	M_6
111	7	a' + b' + c'	M_7

Canonical logic forms

- **Canonical POS form**: - In a product of sum form expression, if each OR term (sum term) consists all the literals(variables) appearing either in complements or uncomplemented form. E.g. (a' + b + c). (a + b' + c'). (a + b + c). Then the form is said to be Canonical POS form.

Binary Representation	Sequence	Maxterm	Designation
000	0	a + b + c	M_0
001	1	a + b + c'	M_1
010	2	a + b' + c	M_2
011	3	a + b' + c'	M_3
100	4	a' + b + c	M_4
101	5	a' + b + c'	M_5
110	6	a' + b' + c	M_6
111	7	a' + b' + c'	M_7

Q Given the function F = P'+QR, where F is a function in three Boolean variables P, Q and R and P' =! P, consider the following statements. **(GATE-2015) (2 Marks)**

S1: F = Σ (4, 5, 6)

S2: F = Σ (0, 1, 2, 3, 7)

S3: F = Π (4, 5, 6)

S4: F = Π (0, 1, 2, 3, 7)

Which of the following is true?

(A) S1-False, S2-True, S3-True, S4-False

(B) S1-True, S2-False, S3-False, S4-True

(C) S1-False, S2-False, S3-True, S4-True

(D) S1-True, S2-True, S3-False, S4-False

Break

No of functions possible

Q With n-Boolean variables how many different Boolean functions are possible?

No of functions possible

Q With n-Boolean variables how many different Boolean functions are possible?

Solution: with n binary variable we can generate 2^n combinations. And as we know that a Boolean function can also have only 2 possible values either 0 or 1, so total number of different functions possible will be $2^{(2^n)}$.

Similarly, we can generalize this idea as $x^{(y^z)}$ are the total number of functions possible, x is the nature of the function, y is the nature of the variable and z is the number of variables.

Q How many different Boolean functions of degree 4 are there? **(NET-JUNE-2013)**

(A) 2^4 **(B)** 2^8 **(C)** 2^{12} **(D)** 2^{16}

Q What is the maximum number of different Boolean functions involving n Boolean variables? **(GATE-2007) (1 Marks)**

a) n^2 **b)** 2^n **c)** 2^{2^n} **d)** 2^{n^2}

Break

Complementation

- Let us consider a function f(a, b, c, d,……, 0, 1, +, $\cdot$), then the complement of the function is defined as f'(a', b', c',……, z', 1, 0, ., +).

- i.e. When all the variables are replaced by their compliments, 0→1,1→0, or →and, and → or, then we will be called them compliment of a function.

- We can directly put a bar of the entire Boolean expression to find complement of the function.

- We can also directly deal in minterms and maxterm to write complement function.

OR	→	NOR
AND	→	NAND
EX-OR	→ Complement	EX-NOR

Break

Duality

- Let us consider a function f(a, b, c, d,……,z,0,1,+,·), then the dual of the function is defined as f^d(a, b, c,……,z,1,0,·.,+).

- When the nature of variable remains same but 0 → 1, 1 → 0, or → and, and → or, then they are called dual functions.

a	b		
L 0 H	L 0 H	L 0 H	L 0 H
L 0 H	H 1 L	H 1 L	L 0 H
H 1 L	L 0 H	H 1 L	L 0 H
H 1 L	H 1 L	H 1 L	H 1 L

- When we take dual of a function, then the functionality of a function remains the same but a positive logic system is transformed to negative logic system.

- If a function works corrects in positive logic system, then it must also work correct in negative logic system as it do not depends on magnitude.

OR	→	AND
NOR	→	NAND
EX-OR	→	EX-NOR

dual

Q The dual of a Boolean expression is obtained by interchanging **(NET-DEC-2013)**

(A) Boolean sums and Boolean products

(B) Boolean sums and Boolean products or interchanging 0's and 1's

(C) Boolean sums and Boolean products and interchanging 0's & 1's

(D) Interchanging 0's and 1's

Q The dual of the switching function $x + yz$ is: **(NET-DEC-2007) (NET-DEC-2008)**

(A) $x + yz$ **(B)** $x' + y'z'$ **(C)** $x(y + z)$ **(D)** $x'(y' + z')$

Break

Neutral Function

- A function f is said to be neutral if, it has equal number of minterms and maxterms.

Self-Dual

- A function f is said to be self-dual if both the functions and its dual are same.
- $f = f^D$
- $F(a, b, c) = ab + bc + ca$

Q which of the following functions are self-dual?

~~f (a, b, c) = $\sum_m$ (0, 3)~~

F (a, b, c) = $\sum_m$ (0, 1, 6, 7) ✓

F (a, b, c) = $\sum_m$ (0, 1, 2, 4) ✗

F (a, b, c) = $\sum_m$ (3, 5, 6, 7) ✗

0 — 7

1 — 6

2 — 5

3 — 4

$n \longrightarrow 2^n \longrightarrow \frac{2^n}{2} = 2^{n-1} \longrightarrow \frac{2^{n-1}}{2} = 2^{n-2}$

$$\boxed{{}^{2^{n-1}}C_{2^{n-2}}}$$

$${}^nC_r = \frac{n!}{r!\,(n-r)!}$$

Q how to check weather a function is self-dual or not?

1) check whether function is neutral or not i.e. (minterm = maxterm)

2) A self-dual function does not have any mutually exclusive terms. So in every pair of mutual exclusion term, we can pick only 1 minterm.

3) for n variable functions total 2^n minterms are possible, so we will have 2^{n-1} pair of mutually exclusive minterms, in every pair of mutually exclusive minterms we have two choice so, For n variable function total number of $2^{(2^{(n-1)})}$ self-dual functions are possible.

0←→7
1←→6
2←→5
3←→4
e.g. of Mutual exclusive min terms

Q The dual of a Boolean function F (X_1, X_2, ..., X_n, +, *, '), written as F^D, is the same expression as that of F with + and * swapped. F is said to be self-dual if F = F^D. The number of self-dual functions with n Boolean variables is. **(GATE-2014) (1 Marks)**

a) 2^n 8

b) $2^{(n-1)}$ 3

c) $2^{(2014^n)}$ 3

d) $2^{(2^{(n-1)})}$ 58

$$\frac{2^n}{2} = 2^{n-1} \longrightarrow 2^{2^{n-1}}$$

Q How many Boolean functions of degree n are self-dual? **(NET-JUNE-2014)**

(A) 2^n **(B)** $(2)^{2n}$ **(C)** 2^{n^2} **(D)** $(2)^{\wedge}(2^{n-1})$

Break

Orthogonal

- A function f is said to be orthogonal if the compliment and dual of the function are same.

- $f' = f^d$

- $f(a, b, c) = a'b'c' + abc + a'bc' + ab'c$

- A function can never be both orthogonal and self-dual because it will imply the function is same to its compliment, which is never possible.

Q how to check weather a function is orthogonal or not?

1) check whether function is neutral or not i.e. (minterm = maxterm)

2) we can choose any pair of mutual exclusive terms, but the minterms and its complement, both must be present.

3) If there is a function f of n-variables, so we will have 2^{n-1} pair of mutually exclusive minterms, out of which we have to select exactly half of the pairs (2^{n-2}), then $(2^{n-1})C(2^{n-2})$ different orthogonal functions are possible.

0←→7
1←→6
2←→5
3←→4

e.g. of Mutual exclusive min terms

Q.39 Consider a Boolean function $f(w, x, y, z)$ such that **(GATE- 2021)**

$$f(w, 0, 0, z) = 1$$
$$f(1, x, 1, z) = x + z$$
$$f(w, 1, y, z) = wz + y$$

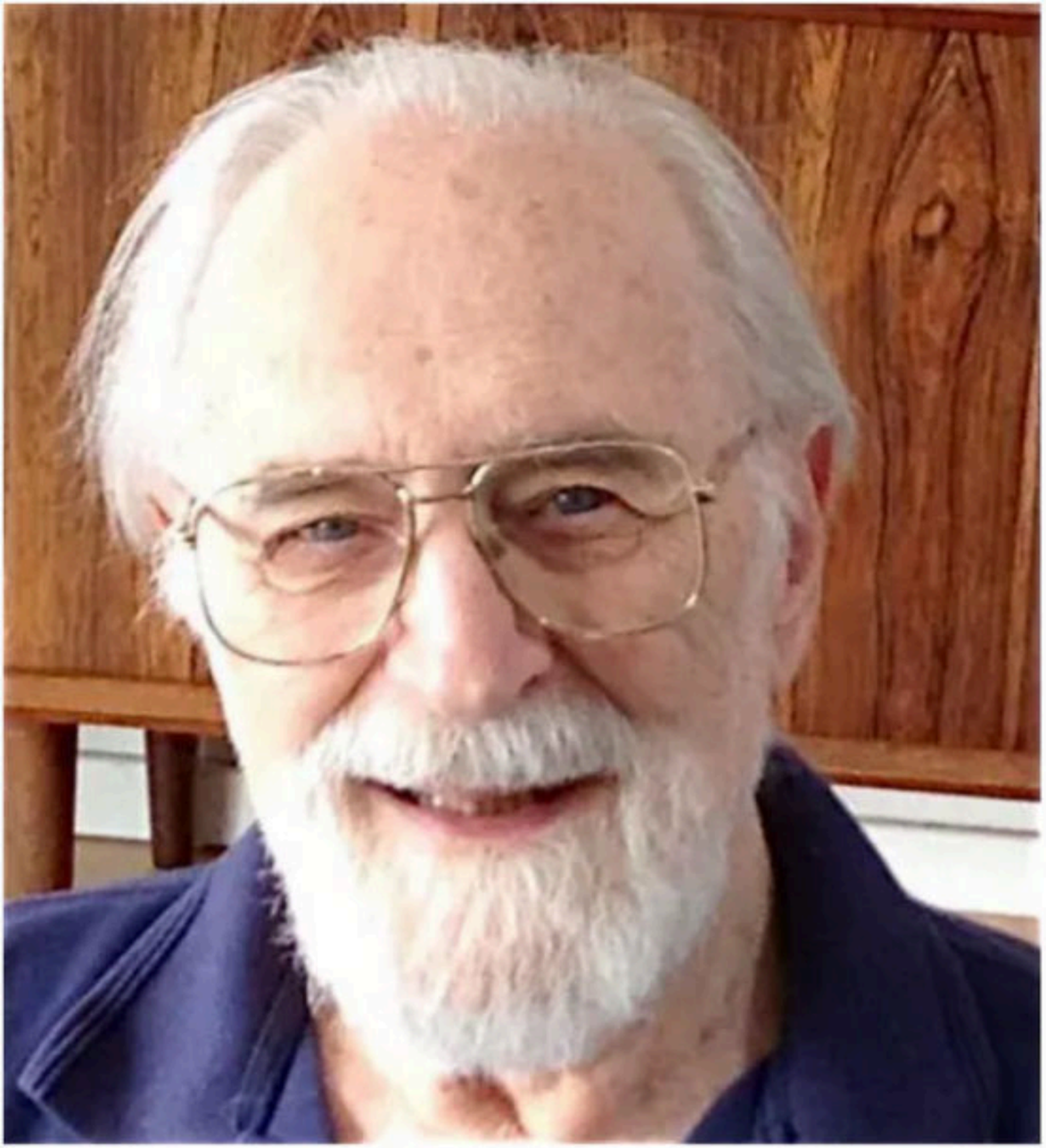
The number of literals in the minimal sum-of-products expression of f is _____.

Simplification of Boolean expression

- After deriving a Boolean expression from the truth table next important step is to minimise it, so that the cost of implementation into hardware and delay because of additional gates can be reduced. There are following methods for simplifying a Boolean expression

1) Using Karnaugh-map

2) Using Algebraic method (Boolean Laws)



- Maurice Karnaugh (born 4 October 1924) is an American physicist, mathematician and inventor known for the Karnaugh map used in Boolean algebra.

- Age: 96

Karnaugh Map

- **Problem with other methods: -** The algebraic procedure using Boolean laws and rules for minimising Boolean expression becomes difficult when a function becomes complex, because we have to identify where is the scope of minimization based on some Boolean laws and It is difficult to understand where we reach saturation point or not.

$$a + a'b + a'b'c + a'b'c'd'$$

- Karnaugh map is one of the most extensively used tool, it is a graphical representation, represents truth table by pictorial form, provides a systematic method for simplifying or minimizing a Boolean expression. For a n-variable k-map, there will be 2^n cells addressed by a gray code. Each cell corresponds to one minterm or maxterm.

$f(a, b, c, d)$

	a	b	c	d	F
0	0	0	0	0	
1	0	0	0	1	
2	0	0	1	0	
3	0	0	1	1	
4	0	1	0	0	1
5	0	1	0	1	
6	0	1	1	0	1
7	0	1	1	1	
8	1	0	0	0	
9	1	0	0	1	1
10	1	0	1	0	
11	1	0	1	1	
12	1	1	0	0	
13	1	1	0	1	
14	1	1	1	0	
15	1	1	1	1	

	ab	$\bar{a}\bar{b}$	$\bar{a}b$	ab	$a\bar{b}$
cd		00	01	11	10
$\bar{c}\bar{d}$	00	0	4	12	8
$\bar{c}d$	01	1	5	13	9
cd	11	3	7	15	11
$c\bar{d}$	10	2	6	14	10

4x4

	ab	a'b'	a'b	ab	ab'
cd		00	01	11	10
c'd'	00	0	4	12	8
c'd	01	1	5	13	9
cd	11	3	7	15	11
cd'	10	2	6	14	10

L

	cd	$\bar{c}\bar{d}$	$\bar{c}d$	cd	$c\bar{d}$
ab		00	01	11	10
$\bar{a}\bar{b}$	00	0	1	3	2
$\bar{a}b$	01	4	5	7	6
ab	11	12	13	15	14
$a\bar{b}$	10	8	9	11	10

R

$a\bar{c} + \bar{a}c$ = 4 ×

$\Rightarrow$ 4

$ab + ac$

$\rightarrow$ ~~$a(b+c)$~~ ×

	ab	a'b'	a'b	ab	ab'
cd		00	01	11	10
c'd'	00	0	4	12	8
c'd	01	1	5	13	9
cd	11	3	7	15	11
cd'	10	2	6	14	10

	cd	c'd'	c'd	cd	cd'
ab		00	01	11	10
a'b'	00	0	1	3	2
a'b	01	4	5	7	6
ab	11	12	13	15	14
ab'	10	8	9	11	10

	ad	$\bar{a}\bar{d}$	$\bar{a}d$	ad	$a\bar{d}$
bc		00	01	11	10
$\bar{b}\bar{c}$	00	0	1	9	8
$\bar{b}c$	01	2	3	11	10
bc	11	6	7	15	14
$b\bar{c}$	10	4	5	13	12

$\bar{a}\bar{c}$

	db	$\bar{d}\bar{b}$	$\bar{d}b$	db	$d\bar{b}$
ca		00	01	11	10
$\bar{c}\bar{a}$	00	0	4	5	1
$\bar{c}a$	01	8	12	13	9
ca	11	6	14	15	11
$c\bar{a}$	10	X_2	X_6	X_7	X_3

$\bar{a}c$

	ad	a'd'	a'd	ad	ad'
bc		00	01	11	10
b'c'	00	0	1	9	8
b'c	01	2	3	11	10
bc	11	6	7	15	14
bc'	10	4	5	13	12

	db	d'b'	d'b	db	db'
ca		00	01	11	10
c'a'	00	0	4	5	1
c'a	01	8	12	13	9
ca	11	10	14	15	11
ca'	10	2	6	7	3

a

	ab	$\bar{a}\bar{b}$	$\bar{a}b$	ab	$a\bar{b}$
c		00	01	11	10
$\bar{c}$	0	0	2	6	4
c	1	1	3	7	5

b

	bc	$\bar{b}\bar{c}$	$\bar{b}c$	bc	$b\bar{c}$
a		00	01	11	10
$\bar{a}$	0	0	1	3	2
a	1	4	5	7	6

c

	a	$\bar{a}$	a
bc		0	1
$\bar{b}\bar{c}$	00	0	4
$\bar{b}c$	01	1	5
bc	11	3	7
$b\bar{c}$	10	2	6

d)

	ab	a'b'	a'b	ab	ab'
c		00	01	11	10
c'	0	0	2	6	4
c	1	1	3	7	5

	bc	b'c'	b'c	bc	bc'
a		00	01	11	10
a'	0	0	1	3	2
a	1	4	5	7	6

	a	a'	a
bc		0	1
b'c'	00	0	4
b'c	01	1	5
bc	11	3	7
bc'	10	2	6

	a	$\bar{a}$	a
b		0	1
$\bar{b}$	0	0	2
b	1	1	3

	b	$\bar{b}$	b
a		0	1
$\bar{a}$	0	0	1
a	1	2	3

	a	a’	a
b		0	1
b’	0	0	2
b	1	1	3

	a	a’	a
b		0	1
b’	0	0	2
b	1	1	3

Break

K-Map for POS

	a+b	$a+b$	$a+\overline{b}$	$\overline{a}+\overline{b}$	$\overline{a}+b$
c+d		0+0	0+1	1+1	1+0
$c+d$	0+0	0	4	12	8
$c+\overline{d}$	0+1	1	5	13	9
$\overline{c}+\overline{d}$	1+1	3	7	15	11
$\overline{c}+d$	1+0	2	6	14	10

K-Map for POS

	a+b	a+b	a+b’	a’+b’	a’+b
c+d		0+0	0+1	1+1	1+0
c+d	0+0	0	4	12	8
c+d’	0+1	1	5	13	9
c’+d’	1+1	3	7	15	11
c’+d	1+0	2	6	14	10

Break

- **<u>Minimal function</u>**: - A function is said to be minimal if it is representing the function expression, if it is using minimal no of literals to represent the function.

- Rules of grouping: -

1. Every minterm must be covered
2. Group must have contiguous Cells(circular)
3. Group must be in horizontal or vertical fashion(circular)
4. Number of cells in a group must be in power of 2
5. Will Try to make largest group possible, so that number of literals in the expression can be reduced
6. Can also take don't care if it helps in creating the larger groups, other wise don't care.
7. Will consider new implicant if it is covering some new minterm.

	ab	a'b'	a'b	ab	ab'
cd		00	01	11	10
c'd'	00	0	4	12	8
c'd	01	1	5	13	9
cd	11	3	d 7	d 15	11
cd'	10	2	d 6	1 14	d 10

Don’t care condition

- Don’t care cases are those cases which can never occur logically in that function.
- It is not essential to cover don’t care conditions, but if don’t care are helping to generate bigger prime implicants, then we can use them.

Break

	ab	a'b'	a'b	ab	ab'
cd		00	01	11	10
c'd'	00			1	
c'd	01	D	D	1	D
cd	11		1	1	
cd'	10		1	1	

$bc + ab = y$

	ab	a'b'	a'b	ab	ab'
cd		00	01	11	10
c'd'	00		1		
c'd	01		1	1	1
cd	11	1	1	1	
cd'	10			1	

2ω

$\bar{a}b\bar{c} + \bar{a}cd + abc + a\bar{c}d$

=12

First try to cover those minterms which do not have an option.

Break

	ab	a'b'	a'b	ab	ab'
cd		00	01	11	10
c'd'	00		D		D
c'd	01	1	1	D	
cd	11		D	1	1
cd'	10		1	D	

For some functions more than one minimal Boolean expression are possible

Break

- **Implicants**: - Collection of adjacent minterms are called implicants.

- **Prime Implicant (PI)**: - Implicant is called Prime implicant (PI) if it is not subset of any other implicant(group). (overlapping is allowed).
  - We will always try to find Prime Implicant

$f(a, b, c, d) = \sum_m \{4,5,6,7,8,9,10,11,13,14\}$

	ab	a'b'	a'b	ab	ab'
cd		00	01	11	10
c'd'	00	0	4	12	8
c'd	01	1	5	13	9
cd	11	3	7	15	11
cd'	10	2	6	14	10

Essential Prime Implicant (EPI)

- If a prime implicant (PI) have some unique minterm which no other prime implicant covers then, it is called essential prime implicant (EPI).

- Essential prime implicant (EPI) must always be present in the minimal Boolean expression.

Break

Q $f(a, b, c) = \sum_m \{1,2,3,4,5\}$, find Number of PI, Number of EPI, Number of literals are there in minimal expression, Number of different minimal expression possible?

	ab	a'b'	a'b	ab	ab'
c		00	01	11	10
c'	0	0	2	6	4
c	1	1	3	7	5

1. Number of PI
2. Number of EPI
3. Number of different minimal expression possible
4. Number of literals are there in minimal expression

Q Consider the minterm list form of a Boolean function F given below.

F (P, Q, R, S) = ∑m (0, 2, 5, 7, 9, 11) + d (3, 8, 10, 12, 14)

Here, m denotes a minterm and d denotes a don't care term. The number of essential prime implicants of the function F is _____. **(GATE-2018) (2 Marks)**

	ab	a'b'	a'b	ab	ab'
cd		00	01	11	10
c'd'	00	0	4	12	8
c'd	01	1	5	13	9
cd	11	3	7	15	11
cd'	10	2	6	14	10

Q The total number of prime implicants of the function f(w, x, y, z) = Σ(0, 2, 4, 5, 6, 10) is _____. **(GATE-2015) (1 Marks)**

(A) 2 **(B)** 3 **(C)** 4 **(D)** 5

	ab	a'b'	a'b	ab	ab'
cd		00	01	11	10
c'd'	00	0	4	12	8
c'd	01	1	5	13	9
cd	11	3	7	15	11
cd'	10	2	6	14	10

Q Which are the essential prime implicants of the following Boolean function?

(GATE-2004) (1 Marks)

f (a, b, c) = a'c + ac' + b'c

(A) a'c and ac'

(B) a'c and b'c

(C) a'c only

(D) ac' and bc'

	ab	a'b'	a'b	ab	ab'
c		00	01	11	10
c'	0	0	2	6	4
c	1	1	3	7	5

Break

Q Consider the functions given below ?

	ab	a'b'	a'b	ab	ab'
c		00	01	11	10
c'	0	1	1		1
c	1		1	1	1

1. Number of PI

2. Number of EPI

3. Number of different minimal expression possible

4. Number of literals are there in minimal expression

Q Consider the functions given below ?

1. Number of PI
2. Number of EPI
3. Number of different minimal expression possible
4. Number of literals are there in minimal expression

	ab	a'b'	a'b	ab	ab'
cd		00	01	11	10
c'd'	00	1			1
c'd	01	1	1		
cd	11		1	1	
cd'	10			1	1

Break

	ab	a'b'	a'b	ab	ab'
cd		00	01	11	10
c'd'	00	1	D	D	1
c'd	01	D			
cd	11				
cd'	10	1			D

	ab	a'b'	a'b	ab	ab'
cd		00	01	11	10
c'd'	00		1	1	
c'd	01	D			1
cd	11	D			1
cd'	10		1	1	D

	ab	a'b'	a'b	ab	ab'
cd		00	01	11	10
c'd'	00	D	1		1
c'd	01		1	D	
cd	11	1	D	D	
cd'	10	D			D

	ab	a'b'	a'b	ab	ab'
cd		00	01	11	10
c'd'	00			D	1
c'd	01		1	D	1
cd	11		1	D	D
cd'	10		1	D	D

Q Simplify the following using K-map:

F (A, B, C, D) = Σ (0, 1, 2, 8, 9, 12, 13)

d (A, B, C, D) = Σ (10, 11, 14, 15) d stands for don't care condition. **(NET-JULY-2018)**

(a) A+ B' D' + BC **(b)** A+ B' D' + B' C' **(c)** A'+ B' C' **(d)** A' +B' C'+ B' D'

	ab	a'b'	a'b	ab	ab'
cd		00	01	11	10
c'd'	00	0	4	12	8
c'd	01	1	5	13	9
cd	11	3	7	15	11
cd'	10	2	6	14	10

Q The Boolean function with the Karnaugh map **(NET-NOV-2017)**

(a) (A+C).D+B **(b)** (A+B).C+D **(c)** (A+D).C+B **(d)** (A+C).B+D

CD \ AB	00	01	11	10
00	0	1	1	0
01	0	1	1	1
11	1	1	1	1
10	0	1	1	0

Q Given f (w, x, y, z) = Σ_m (0,1,2,3,7,8,10) + Σ_d (5,6,11,15), where d represents the don't-care condition in Karnaugh maps. Which of the following is a minimum product-of-sums (POS) form of f (w, x, y, z)? **(GATE-2017) (2 Marks)**

a) f = (w' + z') (x' + z) **b)** f = (w' + z) (x + z) **c)** f = (w + z) (x' + z) **d)** f = (w + z') (x' + z)

	wx	w'x'	w'x	wx	wx'
yz		00	01	11	10
y'z'	00	0	4	12	8
y'z	01	1	5	13	9
yz	11	3	7	15	11
yz'	10	2	6	14	10

	w+x	w+x	w+x'	w'+x'	w'+x
y+z		0+0	0+1	1+1	1+0
y+z	0+0	0	4	12	8
y+z'	0+1	1	5	13	9
y'+z'	1+1	3	7	15	11
y'+z	1+0	2	6	14	10

Q The Karnaugh map for a Boolean function is given as **(NET-AUG-2016)**

The simplified Boolean equation for the above Karnaugh Map is

(a) AB + CD + AB' + AD

(b) AB + AC + AD + BCD

(c) AB + AD + BC + ACD

(d) AB + AC + BC + BCD

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	0	0
$\bar{A}B$	0	0	1	0
AB	1	1	1	1
$A\bar{B}$	0	1	1	1

Q Consider the following minterm expression for F: **(GATE-2014) (2 Marks)**

F(P, Q, R, S) = Σ (0, 2, 5, 7, 8, 10, 13, 15)

The minterm 2, 7, 8, 13 are 'do not care' term.

The minimum sum-of-products from for F is

a) QS' + Q'S

b) Q'S' + QS

c) Q'R'S' + Q'RS' + QR'S + QRS

d) P'Q'S' + P'QS + PQS + PQ'S'

	pq	p'q'	p'q	pq	pq'
rs		00	01	11	10
r's'	00	0	4	12	8
r's	01	1	5	13	9
rs	11	3	7	15	11
rs'	10	2	6	14	10

Q The simplified function in product of sums of Boolean function F(W, X, Y, Z) = Σ(0, 1, 2, 5, 8, 9, 10) is **(NET-JUNE-2013)**

(A) (W' + X') (Y' + Z') (X' + Z)

(B) (W' + X') (Y' + Z') (X' + Z')

(C) (W' + X') (Y' + Z) (X' + Z)

(D) (W' + X') (Y + Z') (X' + Z)

	wx	w'x'	w'x	wx	wx'
yz		00	01	11	10
y'z'	00	0	4	12	8
y'z	01	1	5	13	9
yz	11	3	7	15	11
yz'	10	2	6	14	10

	w+x	w+x	w+x'	w'+x'	w'+x
y+z		0+0	0+1	1+1	1+0
y+z	0+0	0	4	12	8
y+z'	0+1	1	5	13	9
y'+z'	1+1	3	7	15	11
y'+z	1+0	2	6	14	10

Q What is the minimal form of the Karnaugh map shown below? Assume that X denotes a don't care term. **(GATE-2012) (2 Marks)**

(A) b'd' **(B)** b'd' + b'c' **(C)** b'd' + a'b'c'd' **(D)** b'd' + b'c' + c'd'

	ab	a'b'	a'b	ab	ab'
cd		00	01	11	10
c'd'	00	1	D	D	1
c'd	01	D			1
cd	11				
cd'	10	1			D

Q (NET-JUNE-2010)

The function represented by the k-map given below is

A \ BC			
	1	0	0	1
	1	0	0	1

(A) $A \cdot B$

(B) $AB + BC + CA$

(C) $\overline{B \oplus C}$

(D) $A \cdot B \cdot C$

Q In the Karnaugh map shown below, X denotes a don't care term. What is the minimal form of the function represented by the Karnaugh map? **(GATE-2008) (2 Marks)**

A) b'd' + a'd' **B)** a'b' + b'd' + a'bd' **C)** b'd' + a'bd' **D)** a'b' + b'd' + a'd'

	ab	a'b'	a'b	ab	ab'
cd		00	01	11	10
c'd'	00	1	1		1
c'd	01	D			
cd	11	D			
cd'	10	1	1		D

Q Consider the following Boolean function of four variables: **(GATE-2007) (2 Marks)**

$f(w, x, y, z) = \sum(1,3,4,6,9,11,12,14)$

The function is:

(A) independent of one variable.

(B) independent of two variables.

(C) independent of three variables.

(D) dependent on all the variables.

	wx	w'x'	w'x	wx	wx'
yz		00	01	11	10
y'z'	00	0	4	12	8
y'z	01	1	5	13	9
yz	11	3	7	15	11
yz'	10	2	6	14	10

Q The switching expression corresponding to f(A, B, C, D) = Σ (1, 4, 5, 9, 11, 12) is

(GATE-2005) (2 Marks)

(A) BC'D' + A'C'D + AB'D

(B) ABC' + ACD + B'C'D

(C) ACD' + A'BC' + AC'D'

(D) A'BD + ACD' + BCD'

	ab	a'b'	a'b	ab	ab'
cd		00	01	11	10
c'd'	00	0	4	12	8
c'd	01	1	5	13	9
cd	11	3	7	15	11
cd'	10	2	6	14	10

Q The literal count of a Boolean expression is the sum of the number of times each literal appears in the expression. For example, the literal count of (xy + xz') is 4. What are the minimum possible literal counts of the product-of-sum and sum-of-product representations respectively of the function given by the following Karnaugh map? Here, X denotes "don't care" **(GATE-2003) (2 Marks)**

(A) (11, 9) **(B)** (9, 13) **(C)** (9, 10) **(D)** (11, 11)

	zw	z'w'	z'w	zw	zw'
xy		00	01	11	10
x'y'	00	X	1	0	1
x'y	01	0	1	X	0
xy	11	1	X	X	0
xy'	10	X	0	0	X

Q Minimum sum of product expression for f (w, x, y, z) shown in Karnaugh-map below is **(GATE-2002) (2 Marks)**

(A) xz + y'z **(B)** xz' + zx' **(C)** x'y + zx' **(D)** None of these

	ab	a'b'	a'b	ab	ab'
cd		00	01	11	10
c'd'	00	D	1	0	1
c'd	01	0	1	D	0
cd	11	1	D	D	0
cd'	10	D	0	0	D

Q Given the following Karnaugh map, which one of the following represents the minimal Sum-Of-Products of the map? **(GATE-2001) (2 Marks)**

(A) xy + y'z **(B)** wx'y' + xy + xz **(C)** w'x + y'z + xy **(D)** xz + y

	wx	w'x'	w'x	wx	wx'
yz		00	01	11	10
y'z'	00	0	D	0	D
y'z	01	D	1	D	1
yz	11	0	D	1	0
yz'	10	0	1	D	0

NOT

Q Let f (w, x, y, z) = ∑ (0, 4, 5, 7, 8, 9, 13, 15). Which of the following expressions are NOT equivalent to f? **(GATE-2007) (2 Marks)**

(A) x'y'z' + w'xy' + wy'z + xz

(B) w'y'z' + wx'y' + xz

(C) w'y'z' + wx'y' + xyz + xy'z

(D) x'y'z' + wx'y' + w'y

	wx	w'x'	w'x	wx	wx'
yz		00	01	11	10
y'z'	00	0	4	12	8
y'z	01	1	5	13	9
yz	11	3	7	15	11
yz'	10	2	6	14	10

Q Which function does NOT implement the Karnaugh map given below? **(GATE - 2000) (2 Marks)**

A) (W + X) Y

B) XY + YW

C) (W + X) (W' + Y) (X' + Y)

D) none of the above

	wz	w'z'	w'z	wz	wz'
xy		00	01	11	10
x'y'	00	0	X	0	0
x'y	01	0	X	1	1
xy	11	1	1	1	1
xy'	10	0	X	0	0

DATA

Q The simplified form of a Boolean equation (AB' + AB'C + AC) (A'C' + B') is: **(NET-JULY-2016)**

(A) A B'

(B) AB'C

(C) A'B

(D) ABC

	ab	a'b'	a'b	ab	ab'
c		00	01	11	10
c'	00	0	2	6	4
c	01	1	3	7	5

	a+b	a+b	a+b'	a'+b'	a'+b
C		0+0	0+1	1+1	1+0
C'	0	0	2	6	4
C	1	1	3	7	5

Q Consider the following Boolean expression for F: **(GATE-2014)(2 Marks)**

F(P, Q, R, S) = PQ + P'QR + P'QR'S

	pq	p'q'	p'q	pq	pq'
rs		00	01	11	10
r's'	00	0	4	12	8
r's	01	1	5	13	9
rs	11	3	7	15	11
rs'	10	2	6	14	10

the minimal sum-of-products from of F is

a) PQ + QR + QS

b) P + Q + R + S

c) P' + Q' + R' + S'

d) P'R + P'R'S + P

Q The sum of products expansion for the function **(NET-DEC-2013)** $F(x, y, z) = (x + y)z'$ is given as

(A) $x'y'z + xyz' + x'yz'$

(B) $xyz + xyz' + xy'z'$

(C) $xy'z' + x'y'z' + xyz'$

(D) $xyz' + xy'z' + x'yz'$

	xy	x'y'	x'y	xy	xy'
z		00	01	11	10
z'	00	0	2	6	4
z	01	1	3	7	5

Q The simplified SOP (Sum of Product) form of the Boolean expression (P + Q' + R'). (P + Q' + R). (P + Q + R') is **(GATE-2011)(1 Marks)**

(A) (P'.Q + R')

(B) (P + Q'.R')

(C) (P'.Q + R)

(D) (P.Q + R)

	pq	p'q'	p'q	pq	pq'
rs		00	01	11	10
r's'	00	0	2	6	4
r's	01	1	3	7	5

	p+q	p+q	p+q'	p'+q'	p'+q
r		0+0	0+1	1+1	1+0
r	0	0	2	6	4
r'	1	1	3	7	5

Q (NET-JUNE-2010)

$AB + \left(\overline{A + B}\right)$ is equivalent to

(A) $A \oplus B$

(B) $A \odot B$

(C) $(A \oplus B) \odot A$

(D) $(A \odot B) \oplus A$

	a	a'	a
b		0	1
b'	0	0	2
b	1	1	3

Q (NET-DEC-2009)

The Boolean expression $\bar{x}\,\bar{y}\,z + y\,z + x\,z$ is equivalent to :

(A) x (B) y (C) z (D) $x + y + z$

	xy	x'y'	x'y	xy	xy'
z		00	01	11	10
z'	00	0	2	6	4
z	01	1	3	7	5

Q The simplified form of the Boolean expression (X + Y + XY) (X + Z) is **(NET-DEC-2009)**

(A) X + Y + ZX + Y

(B) XY – YZ

(C) X+ YZ

(D) XZ + Y

	xy	x'y'	x'y	xy	xy'
z		00	01	11	10
z'	00	0	2	6	4
z	01	1	3	7	5

Q If P, Q, R are Boolean variables, then (P + Q')(PQ' + PR)(P'R' + Q') simplifies **(GATE-2008) (1 Marks)**

(A) PQ' **(B)** PR' **(C)** PQ' + R **(D)** PR' + Q

	pq	p'q'	p'q	pq	pq'
rs		00	01	11	10
r's'	00	0	2	6	4
r's	01	1	3	7	5

	p+q	p+q	p+q'	p'+q'	p'+q
r		0+0	0+1	1+1	1+0
r	0	0	2	6	4
r'	1	1	3	7	5

Q Simplified form of Boolean expression xy+(~x) z + yz is: **(NET-DEC-2007)**

(A) xy+(~x) z

(B) (~x) y + (~x) z

(C) (~x) y + xz

(D) xy + xz

	xy	x'y'	x'y	xy	xy'
z		00	01	11	10
z'	00	0	2	6	4
z	01	1	3	7	5

Q The logic expression x'yz' + x'yz + xyz' + xyz reduces to: **(NET-JUNE-2005)**

(A) x'z **(B)** xyz **(C)** y **(D)** y z

	xy	x'y'	x'y	xy	xy'
z		00	01	11	10
z'	00	0	2	6	4
z	01	1	3	7	5

Q The function $AB'C + A'BC + ABC' + A'B'C + AB'C'$ is equivalent to **(GATE-2004) (1 Marks)**

(A) AC'+AB+A'C **(B)** AB'+AC'+A'C **(C)** A'B+AC'+AB' **(D)** A'B+AC+AB'

	ab	a'b'	a'b	ab	ab'
c		00	01	11	10
c'	00	0	2	6	4
c	01	1	3	7	5

Q Which of the following expressions is equivalent to $(A \oplus B) \oplus C$ **(GATE-2004) (2 Marks)**

	ab	a'b'	a'b	ab	ab'
c		00	01	11	10
c'	00	0	2	6	4
c	01	1	3	7	5

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

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

(C) $ABC+A'(B \oplus C)+B'(A \oplus C)$

(D) None

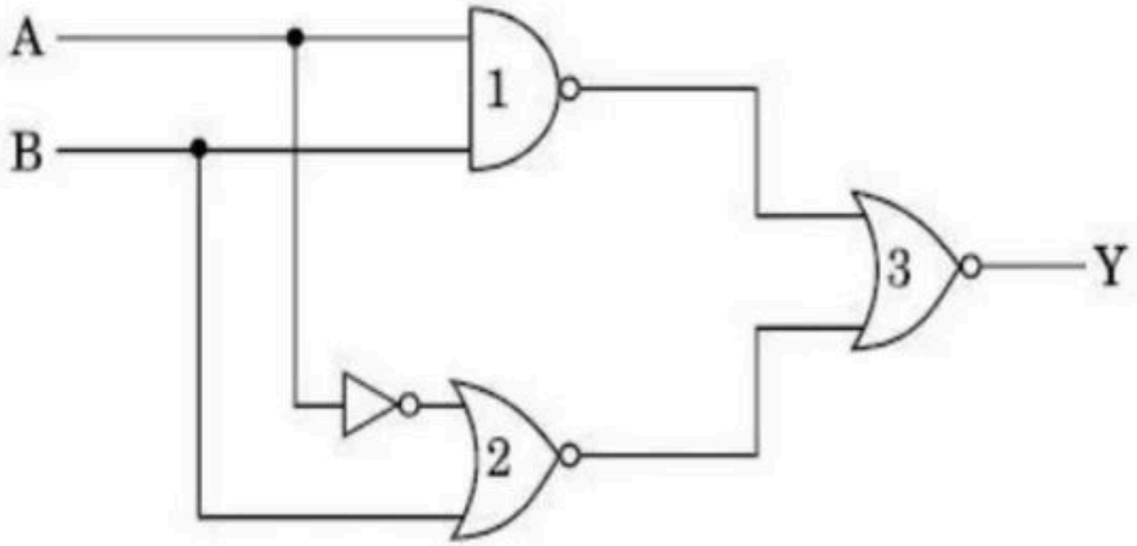
Q The Boolean function x'y' + xy + x'y is equivalent to **(GATE-2004) (1 Marks)**

(A) x' + y' **(B)** x + y **(C)** x + y' **(D)** x' + y

	x	x'	X
y		0	1
y'	0	0	2
y	1	1	3

Application

Q find the Boolean expression for the logic circuit shown below: **(NET-DEC-2018)**



a) ab’

b) a’b

c) ab

d) a’b

Q If *w, x, y, z* are Boolean variables, then which one of the following is INCORRECT? **(GATE-2017) (2 Marks)**

a) $wx + w(x + y) + x(x + y) = x + wy$

b) $(wx'(y + z'))' + w'x = w' + x + y'z$

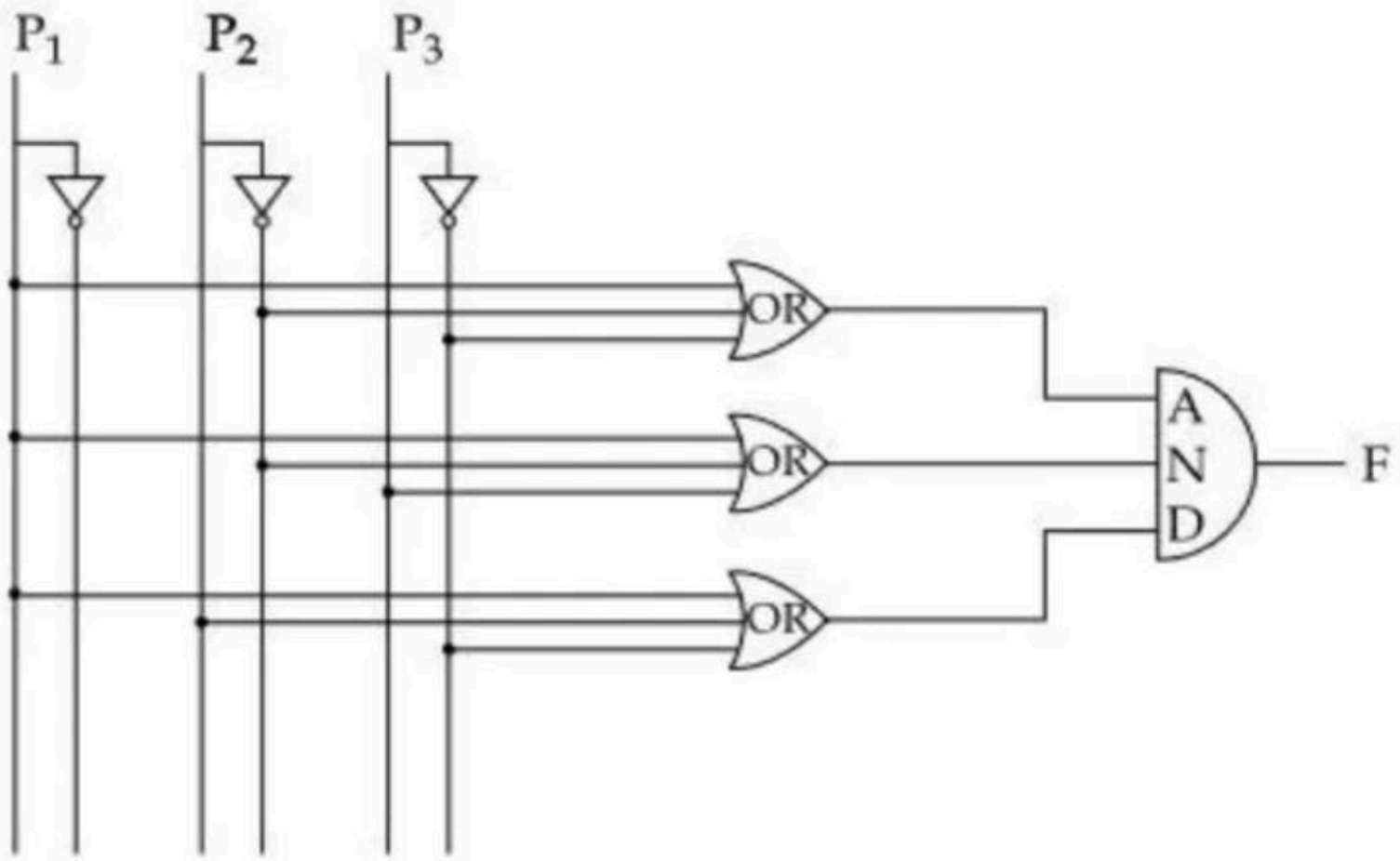
c) $(wx'(y + xz') + w'x')y = xy'$

d) $(w + y)(wxy + wyz) = wxy + wyz$

	wx	w'x'	w'x	wx	wx'
yz		00	01	11	10
y'z'	00	0	4	12	8
y'z	01	1	5	13	9
yz	11	3	7	15	11
yz'	10	2	6	14	10

Q The output of the following combinational circuit is F. **(NET-NOV-2017)**

The value of F is :



(a) $P_1 + P_2' P_3$

(b) $P_1 + P_2' P_3'$

(c) $P_1 + P_2 P_3'$

(d) $P_1' + P_2 P_3$

Q Simplified Boolean equation for the following truth table is: **(NET-JULY-2016)**

(a) F = yz' + y'z

(b) F = xy' + x'y

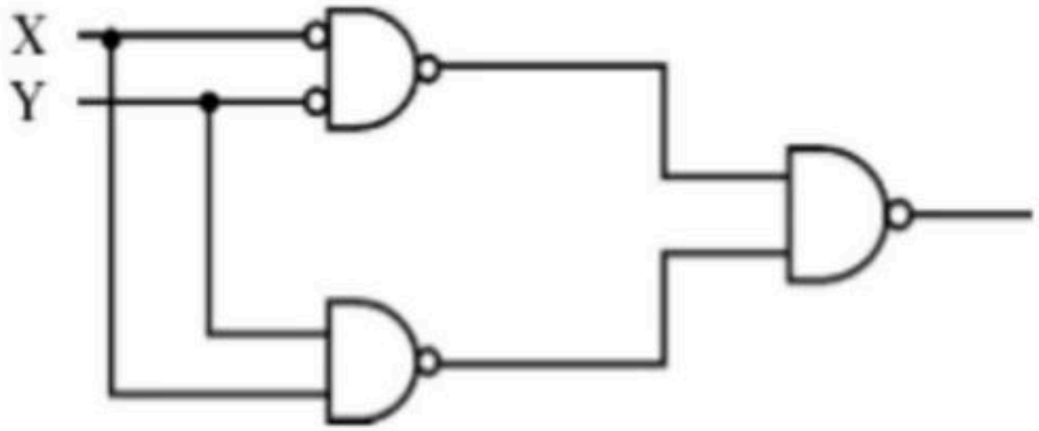
(c) F = x'z + xz'

(d) F = x'z + xz' + xyz

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

	xy	x'y'	x'y	xy	xy'
z		00	01	11	10
z'	0	0	2	6	4
z	1	1	3	7	5

Q The output of the following combinational circuit is: **(NET-AUG-2016)**



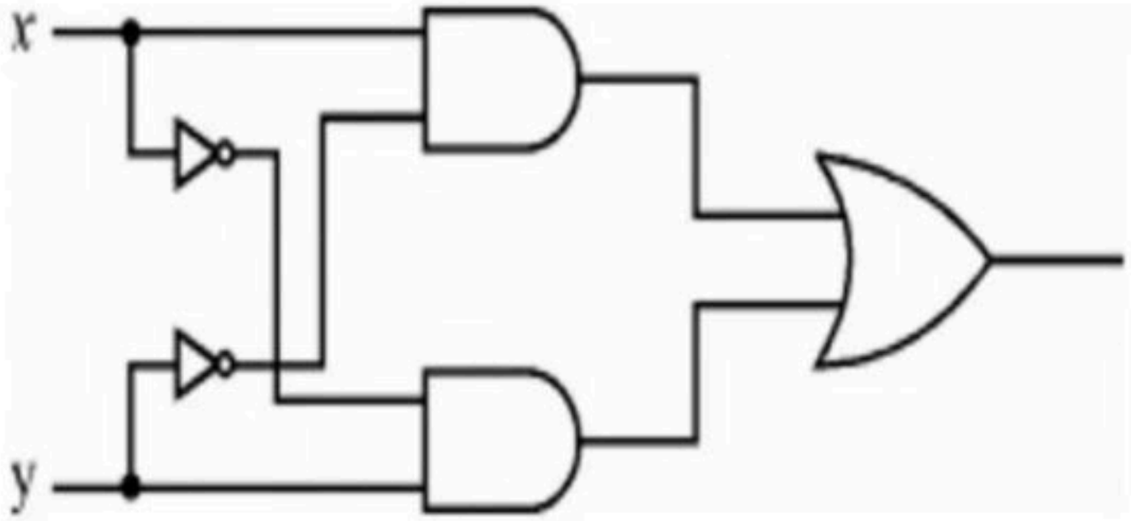
(a) X. Y

(b) X + Y

(c) X $\oplus$ Y

(d) (X $\oplus$ Y)'

Q What will be the output of the following logic diagram? **(NET-DEC-2013)**



	x	x'	X
y		0	1
y'	0	0	2
y	1	1	3

(A) x OR y

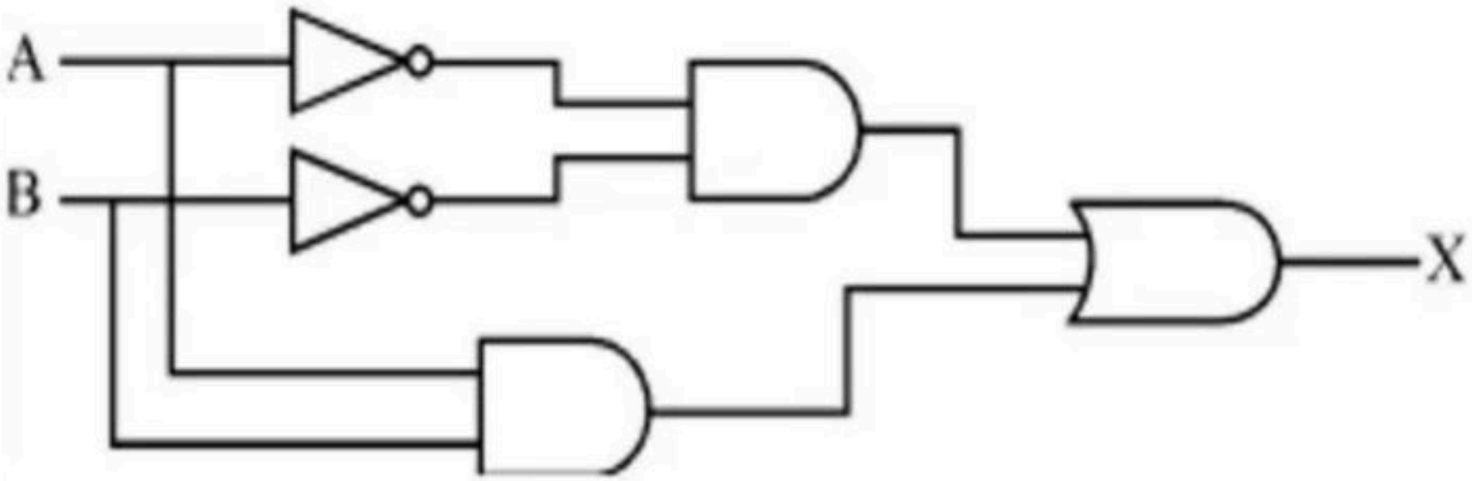
(B) x AND y

(C) x XOR y

(D) x XNOR y

Q What type of logic circuit is represented by the figure shown below? **(NET-SEPT-2013)**

(A) XOR **(B)** XNOR **(C)** XAND **(D)** XNAND



	a	a’	a
b		0	1
b’	0	0	2
b	1	1	3

Q (A + B) (AB)' is equivalent to **(NET-JUNE-2011)**

(A) $A \oplus B$

(B) A u B

(C) $(A \oplus B)$ u A

(D) (A u B) $\oplus$ A

Q Let f(A, B) = A’ + B. Simplified expression for function f(f(x + y, y), z) is : **(GATE-2002) (2 Marks)**

(A) x’ + z **(B)** xyz **(C)** xy’ + z **(D)** None of these

Use Referral Code KGYT for Unacademy Plus to Get minimum 10% Discount

Functions

Q Consider two functions f_1 and f_2?

$F_1 = \sum_m (1,2,5,6)$

$F_2 = \sum_m (2,3,4,5)$

Find $f_1.f_2$ and $f_1 + f_2$

Q Consider two functions f_1 and f_2?

$F_1 = \sum_m (1,2,5,6)$

$F_2 = \sum_m (2,3,4,5)$

Find $f_1.f_2$ and $f_1 + f_2$

	F_1	F_2	$f_1 + f_2$	$f_1 . f_2$
0	0	0		
1	1	0		
2	1	1		
3	0	1		
4	0	1		
5	1	1		
6	1	0		
7	0	0		

Q Consider f_1 & f_2 –

$F_1(a, b, c) = \sum_m (0,2,4) + d\,(3,5,7)$

$F_2(a, b, c) = \sum_m (2,3) + d\,(1,6,7)$

Find $f_1.f_2$ and $f_1 + f_2$

	F_1	F_2	$f_1 + f_2$	$f_1 . f_2$
0	1	0		
1	0	D		
2	1	1		
3	D	1		
4	1	0		
5	D	0		
6	0	D		
7	d	d		

Q Consider f_1 & f_2 –

$F_1(a, b, c, d) = \sum_m (1,3,4,5,9,10,11) + d\ (6, 8)$

$F_2(a, b, c, d) = \sum_m (0,2,4,7,8,15) + d\ (9, 12)$

Find $f_1.f_2$ and $f_1 + f_2$

	F_1	F_2	$f_1 + f_2$	$f_1 . f_2$
0	0	1		
1	1	0		
2	0	1		
3	1	0		
4	1	1		
5	1	0		
6	D	0		
7	0	1		
8	D	1		
9	1	D		
10	1	0		
11	1	0		
12	0	D		
13	0	0		
14	0	0		
15	0	1		

Q Consider a function f (a, b, c) = $\sum_m$ (3, 5, 6) is being minimized to A + BC. Predict what are the don't care conditions?

a) d(2,4) **b)** d(2,7) **c)** d(4,7) **d)** d (2,4,7)

	ab	a'b'	a'b	ab	ab'
c		00	01	11	10
c'	00	0	2	6	4
c	01	1	3	7	5

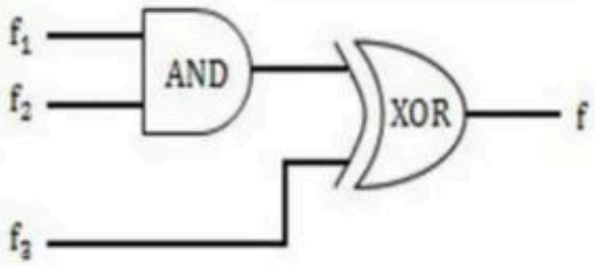
Q Consider three 4-variable functions f1, f2, and f3, which are expressed in sum-of-minterms as

$f_1 = (0, 2, 5, 8, 14)$

$f_2 = (2, 3, 6, 8, 14, 15)$

$f_3 = (2, 7, 11, 14)$

For the following circuit with one AND gate and one XOR gate, the output function f can be expressed as:



For the following circuit with one AND gate and one XOR gate, the output function f can be
expressed as: **(GATE-2019) (2 Marks)**

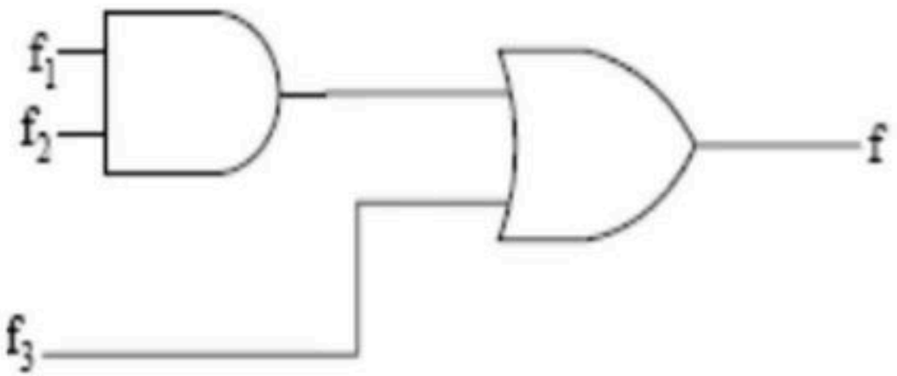
a) Σ(7, 8, 11)

b) Σ(2, 7, 8, 11, 14)

c) Σ(2, 14)

d) Σ(0, 2, 3, 5, 6, 7, 8, 11, 14, 15)

Q Given f_1, f_3 and f in canonical sum of products form (in decimal) for the circuit **(GATE-2008) (1 Marks)**



$f_1 = \Sigma\, m\,(4, 5, 6, 7, 8)$

$f_3 = \Sigma\, m\,(1, 6, 15)$

$f = \Sigma\, m\,(1, 6, 8, 15)$

then f_2 is

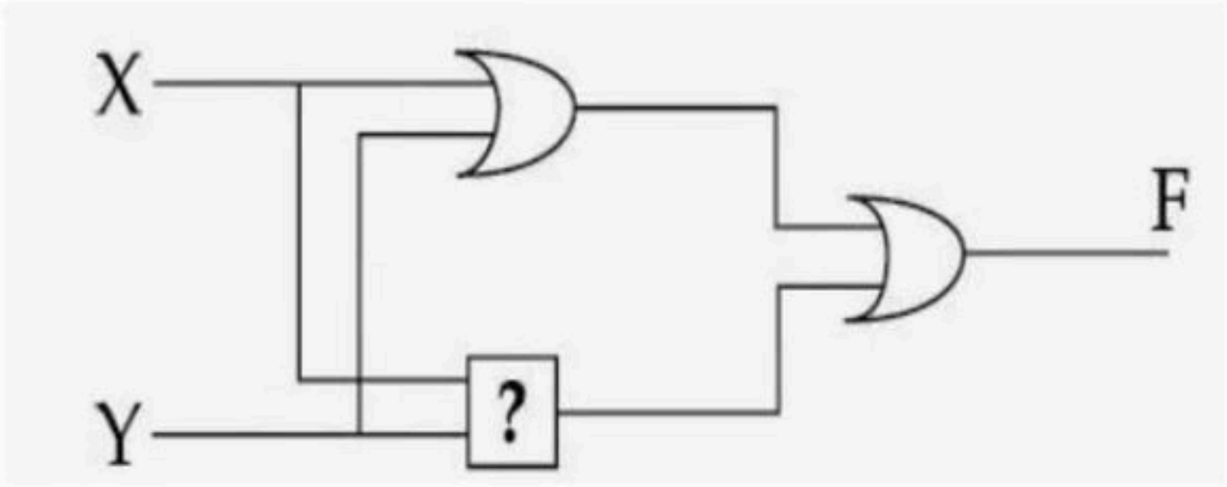
A) $\Sigma_m(4, 6)$

B) $\Sigma_m(4, 8)$

C) $\Sigma_m(6, 8)$

D) $\Sigma_m(4, 6, 8)$

Q Consider the following circuit: **(NET-DEC-2005)**



to make it a Tautology the [?] should be:

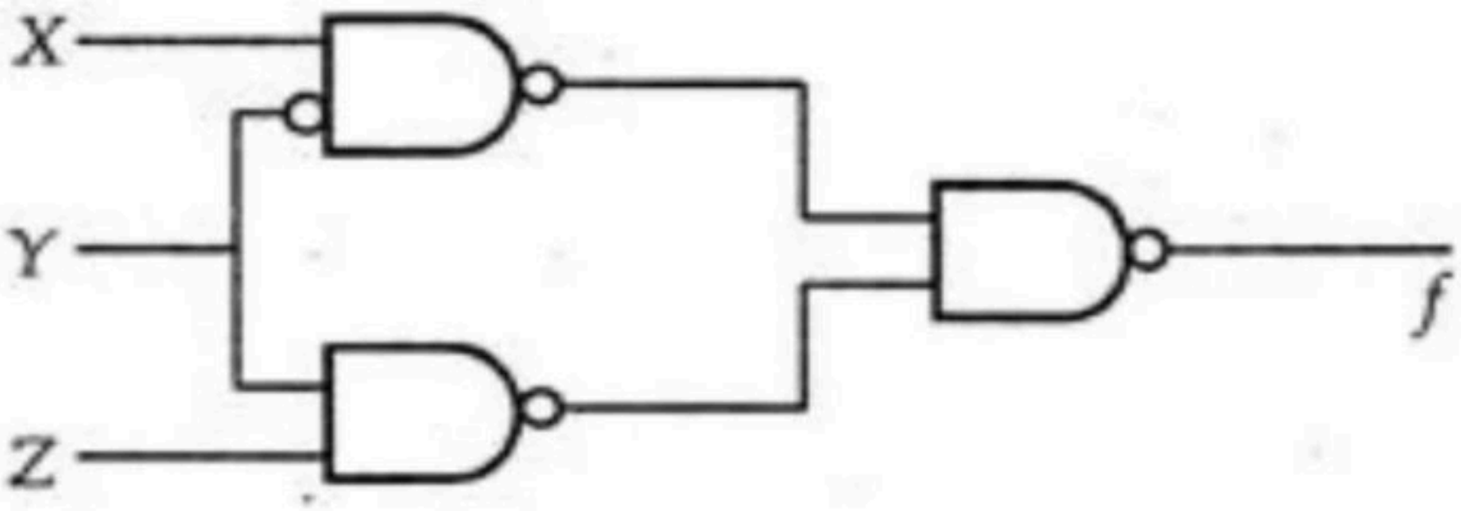
(A) NAND gate

(B) AND gate

(C) OR gate

(D) EX-OR gate

Q Consider the following circuit. **(GATE-2005) (1 Marks)**
Which one of the following is TRUE?



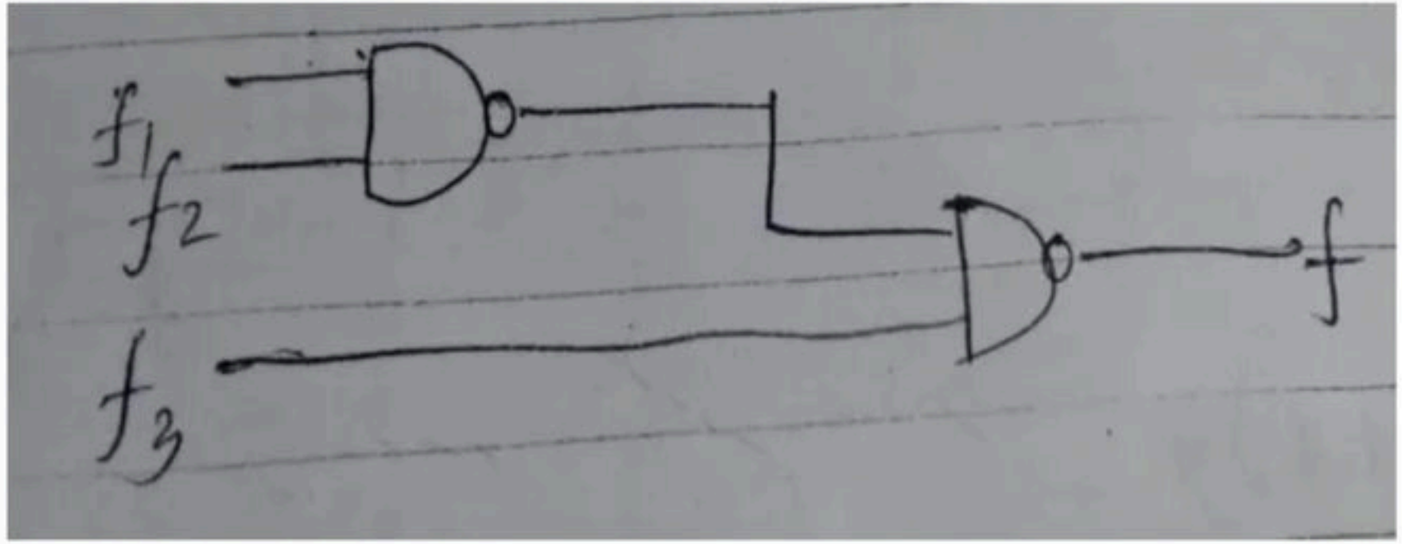
(A) f is independent of X
(B) f is independent of Y
(C) f is independent of Z
(D) None of X, Y, Z is redundant

$F_1(a, b, c, d) = \sum_m (0,2,5,6,8,10,15)$

$F_2(a, b, c, d) = \sum_m (0,5,8,12,14)$

$F(a, b, c, d) = \sum_m (0,5,6,7,8,12,13)$

$f_3(a, b, c, d) = ?$



Break

- **Irredundant function**: - A function f is said to be irredundant if no product term can be removed from the function without changing the functional capability of the function.

- **Minimal function**: - A function is said to be minimal if it is representing the function expression, if it is using minimal no of literals to represent the function.

- If a function is irredundant then, it may or may not be minimal, but if a function is minimal, then it is irredundant.

- F (x, y, z) = xz’ + y’z’ + yz + xz, this function is irredundant but not minimal
- F (x, y, z) = X + yz + y’z’
- Minimal ⊆ irredundant

	xy	x’y’	x’y	xy	xy’
z		00	01	11	10
z’	00	0	2	6	4
z	01	1	3	7	5

Disadvantage: -

- If we deal with large no of variables (more than 4), then it is difficult to use k-map.

Break

Absorption law

- $a + ab = a$
- $a.(a+b) = a$

Compensation theorem

- $ab + a'c + bc = ab + a'c$
- $(a+b)(a' + c)(b + c) = (a+b)(a' + c)$

- $a + a'b = a + b$
- $a.(a'+b) = a.\ b$

Q The absorption law in Boolean algebra say that **(NET-JUNE-2011)**

(A) $X + X = X$

(B) $X . X = X$

(C) $x + x . y = x$

(D) None of the above

Q The number of min-terms after minimizing the following Boolean expression is _____. **(GATE-2015) (1 Marks)**

$[D' + AB' + A'C + AC'D + A'C'D]'$

(A) 1 **(B)** 2 **(C)** 3 **(D)** 4

Q Consider an expression & minimize it

$a + a'b + a'b'c + a'b'c'd'$

Q Upto how many variables, can the Karnaugh map be used? **(NET-JUNE-2006)**

(A) 3 **(B)** 4 **(C)** 5 **(D)** 6

Q.54 Consider the following Boolean expression: **(GATE-2021)**

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

Which of the following Boolean expressions is/are equivalent to F?

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

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

(c) $(X + \bar{Z})(\bar{Y} + \bar{Z})$

(d) $X\bar{Y} + \bar{Z}$