

Logic Gates

General things to know,

$+V_{cc} \rightarrow 5V \rightarrow \text{Logic 1}$

GND, Ground $\rightarrow \text{Logic 0}$

$-2.5V \rightarrow 1$

$+3.5V \rightarrow 1$

$-5.5V \rightarrow 0$

$+1.1V \rightarrow 0$

» Types of logic

(1) +ve logic \rightarrow Higher of two voltage is logic 1
and lower of two voltage is logic 0.

(2) -ve logic \rightarrow Lower of two voltages is logic 1
and higher of two voltage is logic 0.

$-2.5V \rightarrow 0$

$+3.5V \rightarrow 0$

$-5.5V \rightarrow 1$

$+1.1V \rightarrow 1$

NOTE:- +ve logic \rightarrow holes (in processors, practically)

-ve logic $\rightarrow e^-$

$U_e > U_h = -\text{ve logic is practically used.}$

To convert +ve logic into -ve logic, convert 0 by 1
and 1 by 0 in I/P and O/P of Truth table.

AND			OR		
a	b	y	a	b	y
0	0	0	0	0	0
0	1	0	0	1	1
1	0	0	1	0	1
1	1	1	1	1	1

$$Y = a \cdot b \xrightarrow{\text{dual}} a + b$$

} duality theorem
 • \leftrightarrow +
 0 \leftrightarrow 1
 concept of
 -ve logic

- * Dual is nothing but is -ve logic of any function from truth table, it is clear that +ve logic AND gate is equivalent to -ve logic OR gate

OR

Dual of AND gate is OR gate

- » Duality theorem / Dual of a function

Replace • \leftrightarrow +
 0 \leftrightarrow 1

AND $\xleftarrow[\text{Dual}]{\text{-ve logic}}$ OR

NOT $\xleftarrow[\text{Dual}]{\text{-ve logic}}$ NOT

NAND $\xleftarrow[\text{-ve logic}]{\text{Dual}}$ NOR

EX-OR $\xleftarrow[\text{Dual}]{\text{-ve logic}}$ EX-NOR $(a \oplus b)$
 $= ab + \bar{a}\bar{b}$

$$= ab + \bar{a}\bar{b}$$

» Logic Gates

Controls the flow of I/P, checks the condition of I/P and then gives O/P

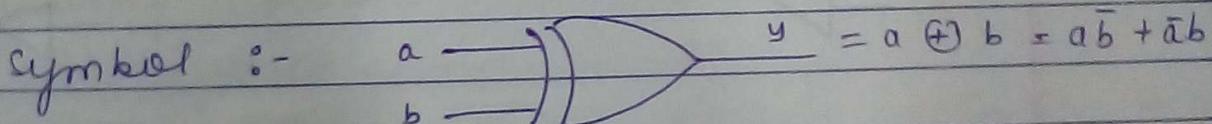
Gates control flow of input and it checks the value of input and depending on the condition, it gives output.

Logic gates are basic building blocks of any digital system

» Arithmetical Circuits

EX-OR and EX-NOR gates are the arithmetical gates

EX-OR GATE



Output of EX-OR gate becomes high when total no. of 1's in input are odd, otherwise low.

Truth Table :

a	b	y
0	0	0
0	1	1
1	0	1
1	1	0

$$= \bar{a}b + \bar{a}b \quad (\text{SOP})$$

$$= (a+b) \cdot (\bar{a}+\bar{b}) \quad (\text{POS})$$

Eg: $y = \sum m(1, 2, 4, 7)$

001 010 100 111

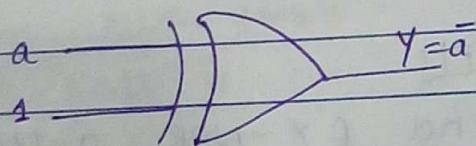
in all min term + All have odd no. of 1.

$$y = a \oplus b \oplus c$$

$$y = \sum m(0, 3, 5, 6) \rightarrow \text{EX-NOR}$$

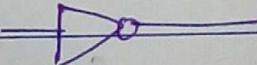
Practical Eg: Two-way switches (staircase wall)

implement EX-OR gate as NOT gate



Buffer

Inverter (NOT gate)



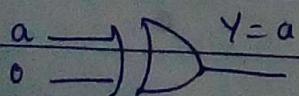
Q

How EX-OR gate works as buffer & inverter?

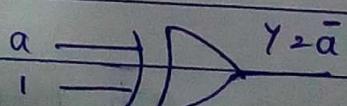
$$\text{EX-OR} \rightarrow \bar{a}b + \bar{b}a$$

$$\text{Buffer} \rightarrow y = a \Rightarrow a \cdot 1 + 0 \cdot \bar{a} = a \Rightarrow \text{Put } b=0 \text{ in EX-OR}$$

$$\text{Inverter} \rightarrow y = \bar{a} \rightarrow a \cdot 0 + 1 \cdot \bar{a} = \bar{a} \rightarrow \text{Put } b=1 \text{ in EX-OR}$$



Buffer



Inverter

Properties of EX-OR gate

$$1 \oplus a = \bar{a}$$

$$0 \oplus a = a$$

$$a \oplus 0 = a$$

$$a \oplus \bar{a} = 1$$

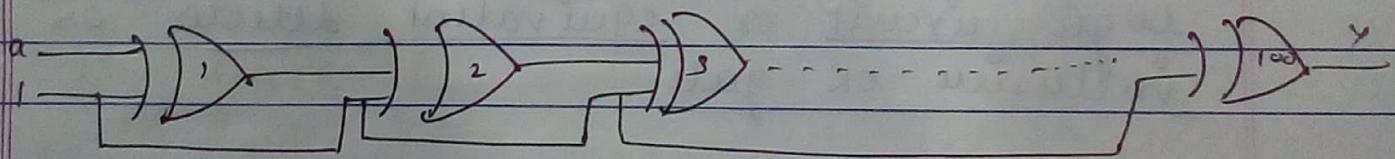
$$a \oplus a \oplus a = a$$

$$a \oplus a \oplus a \oplus a = 0$$

$$a \oplus @ \oplus a + \dots n = a \quad \left\{ \begin{array}{l} \text{if } n = \text{odd} \\ \text{or } n = \text{even} \end{array} \right\}$$

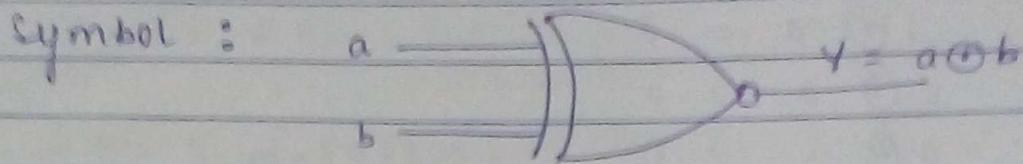
NOTE: Complement of EX-OR gate is EX-NOR gate
 if number of inputs is even and if number
 of inputs is odd then $\text{EX-OR} = \text{EX-NOR}$.

Q what will be output of 100th & 85th gate in
 following circuit.

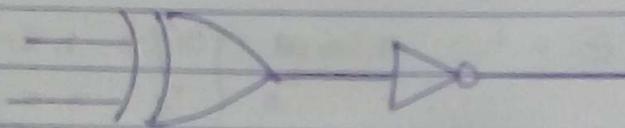


At 100th gate = \bar{a} { even output }
 85th gate = a { odd output }

EX- NOR GATE



Equivalent to



Truth Table :

a	b	y
0	0	1
0	1	0
1	0	0
1	1	1

EX-NOR gate is also called coincidence logic circuit or equivalent detector or inclusive OR gate

$$y = \bar{a}\bar{b} + ab \quad (\text{SOP})$$

OR

$$= (a + \bar{b})(\bar{a} + b) \quad (\text{POS})$$

$$= \overline{\bar{a}\bar{b} + b\bar{a}}$$

Properties of EX-NOR gate

$$a \oplus 0 = \bar{a} \quad (\text{NOT gate})$$

$$a \oplus 1 = a \quad (\text{buffer})$$

$$a \oplus a = 1$$

$$a \oplus \bar{a} = 0$$

$$a \oplus a \oplus a \dots n = 1 \begin{cases} \text{if } n \text{ is even} \\ a \quad \text{if } n \text{ is odd} \end{cases}$$

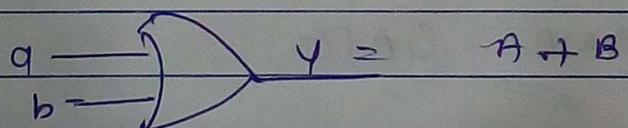
» Basic Gates

with the help of basic logic gates we can implement any digital circuit.

AND, OR, NOT gates are basic gates because with the help of combination of these gates we can implement any boolean function on digital circuit.

OR GATE

Symbol:



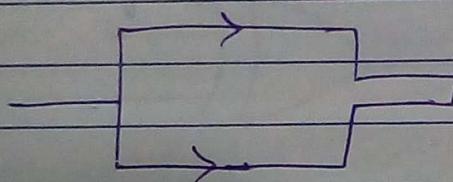
Truth Table :

a	b	y
0	0	0
0	1	1
1	0	1
1	1	1

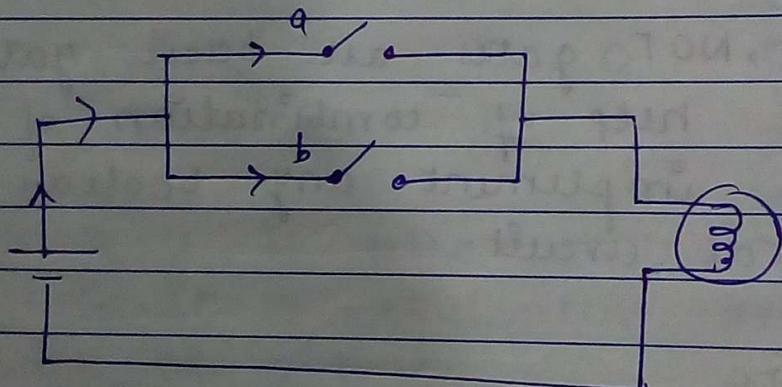
If any one input is high, the output will be high otherwise low.

NOTE: This is not arithmetic circuit coz we cannot implement mathematical op.

OR is parallel path.



implement OR gate by using switch



AND GATE

If all inputs are high output will be high otherwise low.

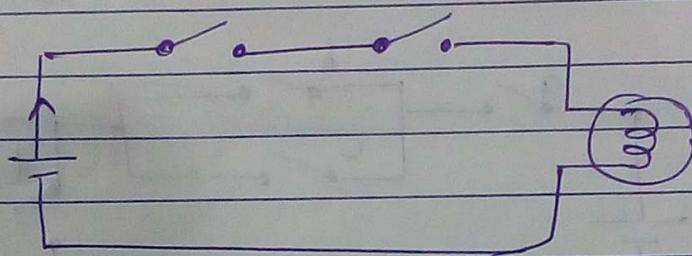
symbol :



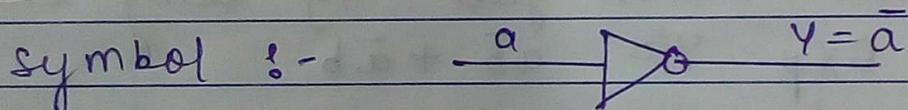
Truth Table :-

a	b	y
0	0	0
0	1	0
1	0	0
1	1	1

AND is series combination / connection



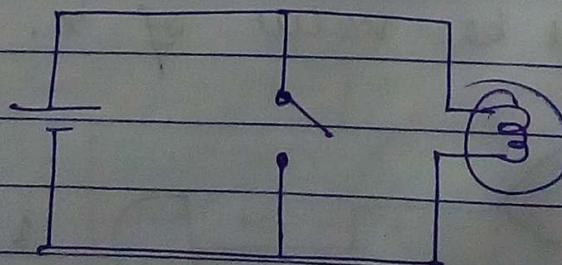
NOT GATE (Inverter)



Truth Table :-

a	y
0	1
1	0

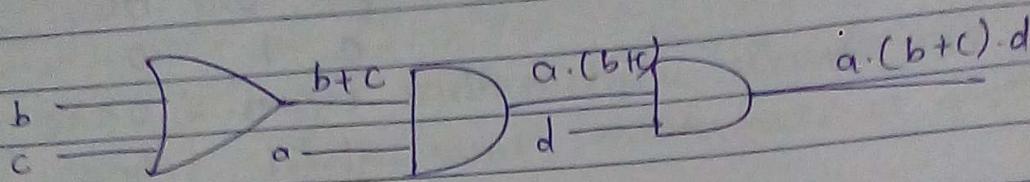
Circuit Representation



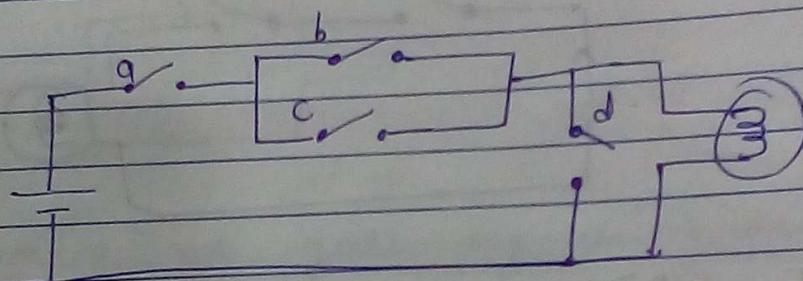
Q

implement

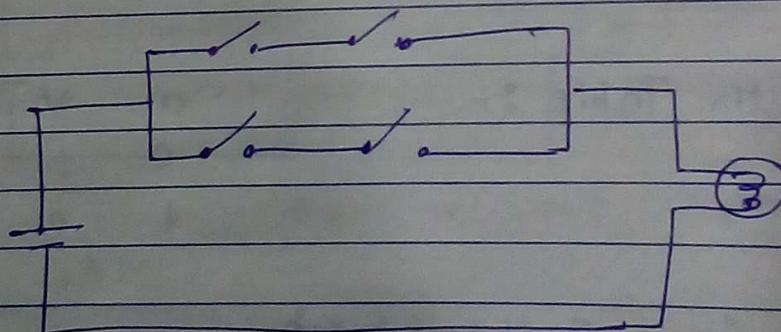
$$Y = a \cdot (b+c) \cdot \bar{d}$$



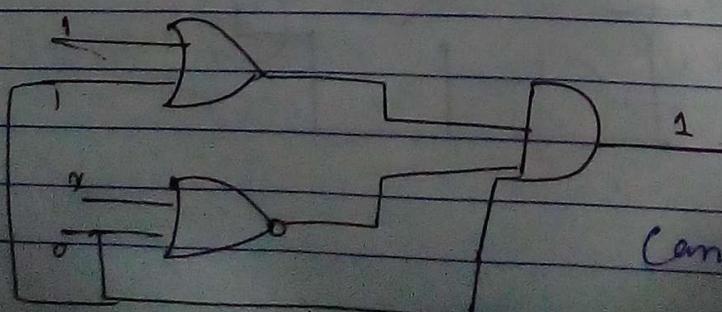
using switch , $Y = a \cdot (b+c) \cdot \bar{d}$
 $= (ab + ac)\bar{d}$



Q implement EX-OR gate by using switch
 Exp $Y = a\bar{b} + \bar{a}b$



Q what will be value of x .

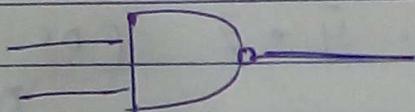


Cannot be
determined.

» Universal Gates

NAND, NOR gates are called universal gates because with the help of either NAND gate or NOR gate we can implement any Boolean function or digital circuit.

NAND \rightarrow AND + NOT



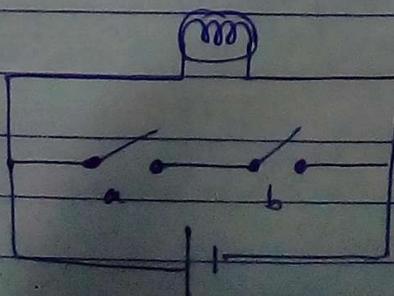
Truth Table :

a	b	y
0	0	1
0	1	1
1	0	1
1	1	0

$$y = \overline{a \cdot b}$$

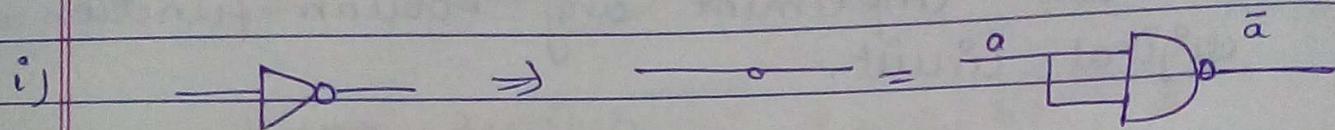
If any 1 input is low output will be high otherwise low.

Implement NAND gate using switch



Advantages of universal gates

1. Less number of ICs are required. Hence cost will reduce



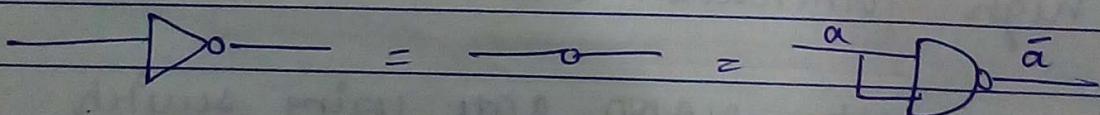
Representation of NOT gate

ii) $\overline{x+y+z} = \bar{x} \cdot \bar{y} \cdot \bar{z}$ (De Morgan's 1st theorem)

iii) $\bar{\bar{x}} = x$

iv)

Step 1: To implement any boolean function by using NAND gate only convert '14' into '1.' by applying De Morgan's 1st thm and $\bar{\bar{x}} = x$ theorem and use



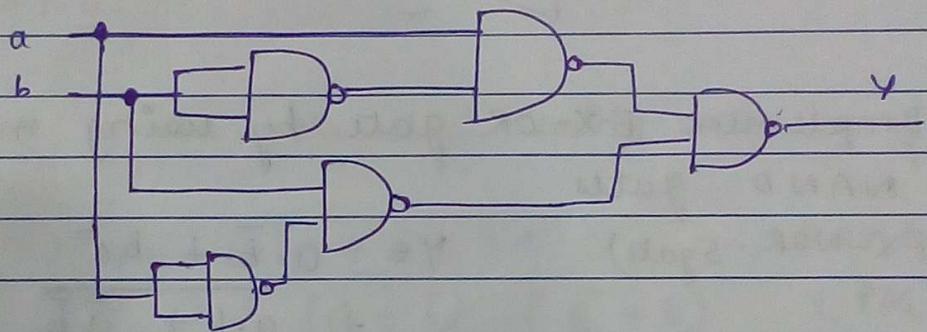
if it is required

Q) Implement EX-OR gate by using universal gate

$$Y = a\bar{b} + \bar{a}b$$

$$= \overline{\overline{a}\bar{b}} + \overline{b}\bar{a}$$

$$= \overline{\overline{a}\bar{b}} \cdot \overline{b}\bar{a}$$



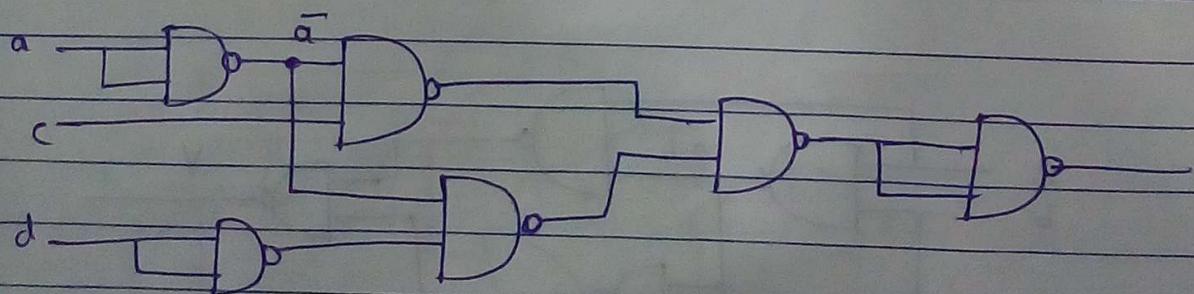
Q

$$Y = (a+\bar{c})(a+d)$$

$$\overline{(a+\bar{c})} \cdot \overline{(a+d)}$$

$$\Rightarrow \overline{a \cdot \bar{c}} \cdot \overline{a \cdot d}$$

$\Rightarrow \overline{\overline{a}\bar{c}} \cdot \overline{\overline{a}\bar{d}}$ \leftarrow double bar used
to implement NAND gate

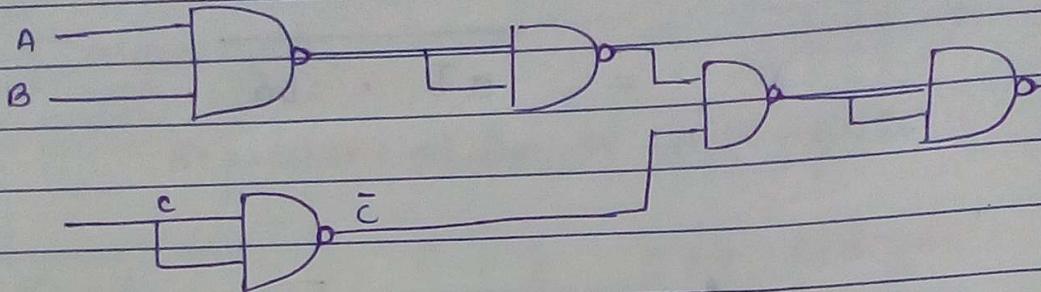


Q

Implement NAND gate $y = \overline{ab\bar{c}}$ by using two input

NAND gate

$$y = \overline{(A \cdot B) \cdot \bar{C}}$$



Q

Implement EX-OR gate by using minimum no. of
NAND gates

(EX-NOR gate)

$$y = \overline{a \cdot \bar{b} + b \cdot \bar{a}}$$

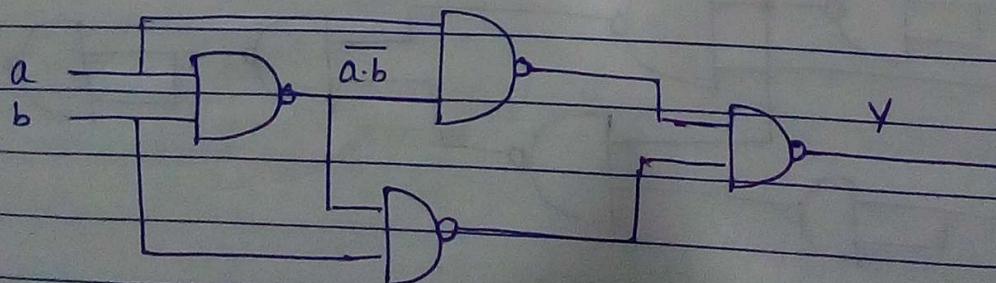
$$= \overline{a \cdot b + \bar{a} \cdot \bar{b}}$$

$$\text{EX-OR} \rightarrow y = \overline{\overline{(a+b)} \cdot \overline{(\bar{a}+b)}} \quad (\text{Pos}) \approx (\text{Inv})$$

$$= a \cdot \overline{(\bar{a}+b)} + b \cdot \overline{(\bar{a}+b)}$$

$$= a \cdot \overline{(\bar{a} \cdot b)} + b \cdot \overline{(\bar{a} \cdot b)}$$

$$= \overline{a \cdot (\bar{a} \cdot b)} \cdot \overline{b \cdot (\bar{a} \cdot b)}$$



Minimum no. of NAND gate req. = 4

Gate	Min. no of NAND gate req.	Min. no of NOR gate req.
NOT	1	1
AND	2	3
OR	3	2
NOR	4	-
\Rightarrow X-OR	4	5
X-NOR	5	4
NAND	-	4

Q Implement X-NOR gate by using min. no. of NOR gate.

$$\psi = \bar{a} \cdot \bar{b} + a \cdot b \quad (\text{SOP}) \quad (\text{used})$$

$$Y = (a + \bar{b}) \cdot (\bar{a} + b) \quad (\text{POS})$$

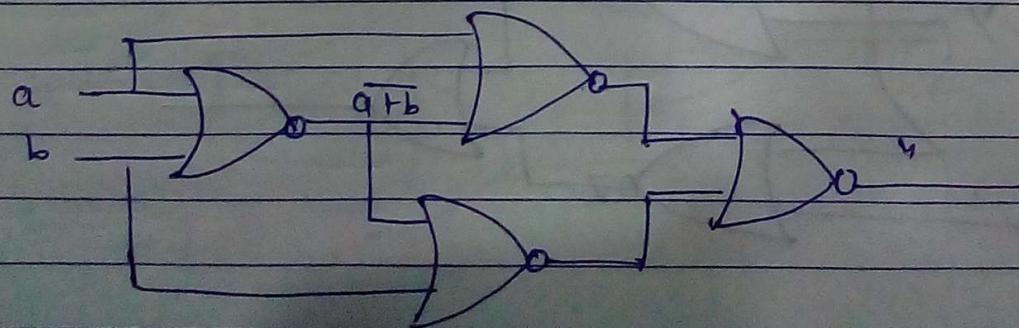
$$= (a + \bar{a}\bar{b}) \cdot (\bar{b} + \bar{a}\bar{b})$$

$$\left\{ \begin{array}{l} x + yz = (x+y)(x+z) \\ \text{OR} \quad x(y+z) = (x \cdot y) + (x \cdot z) \end{array} \right.$$

$$= (\overline{a + \bar{a}\bar{b}}) \cdot (\overline{\bar{b} + \bar{a}\bar{b}})$$

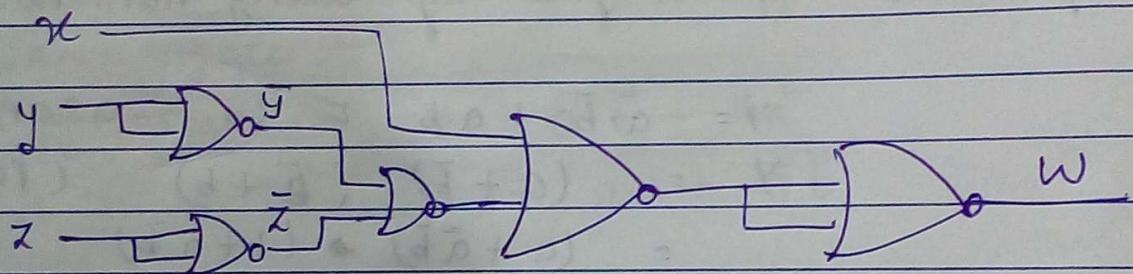
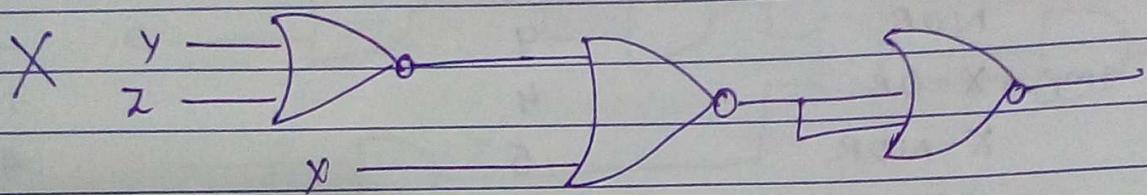
$$= (\overline{a + \bar{a} + b}) \cdot (\overline{b + \bar{a} + b})$$

$$= (\overline{a + \bar{a} + b}) + (\overline{b + \bar{a} + b})$$



Q Implement $x + yz$ by using NOR gate

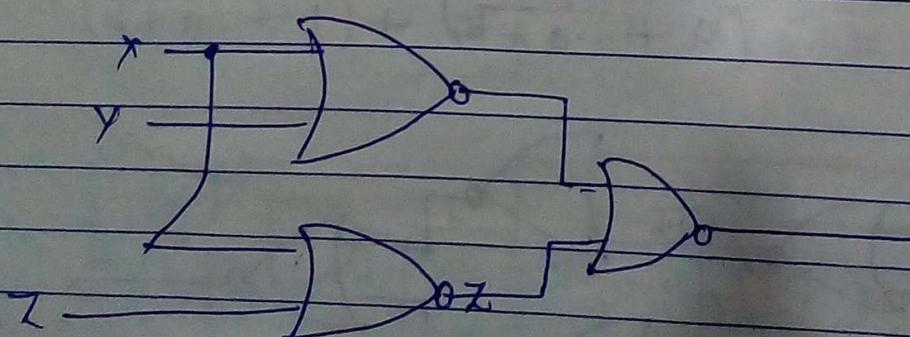
$$W \Rightarrow x + yz = \overline{\overline{x + \overline{yz}}} = \overline{\overline{x} + \overline{\overline{y}} + \overline{z}} \Rightarrow \overline{x + \overline{y} + \overline{z}}$$



Q Implement $x + yz$ by using min. no of NOR gate.

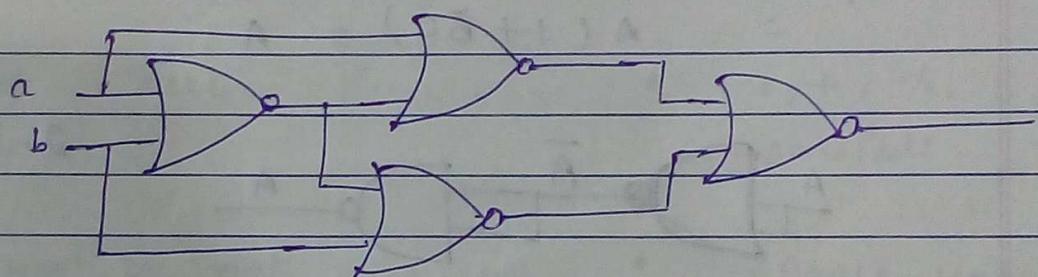
$$\begin{aligned} W &= \overline{\overline{x + yz}} = \overline{\overline{(x+y)} \cdot \overline{(x+z)}} \\ &= \overline{(x+y)} + \overline{(x+z)} \end{aligned}$$

=



Q Implement X-NOR gate by using min. no. of NOR gate

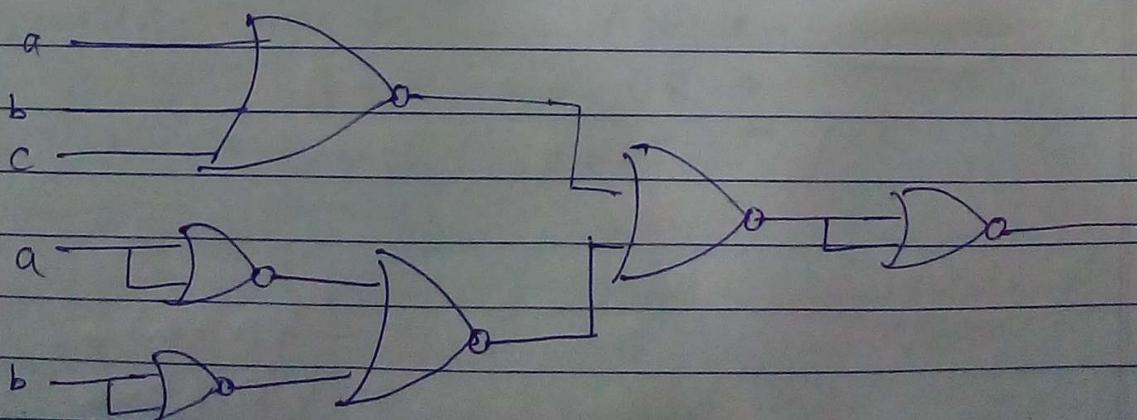
$$\begin{aligned}
 X-\text{NOR} \rightarrow Y &= \overline{\overline{a} \cdot \overline{b}} + ab \quad (\text{SOP}) \\
 &= (a + \overline{b}) \cdot (\overline{a} + b) \quad (\text{POS}) \\
 &= \underline{(a + \overline{a}\overline{b})(b + \overline{a}\overline{b})} \\
 &= \underline{(a + \overline{a+b})} \cdot \underline{(b + \overline{a+b})} \\
 &= \overline{a + \overline{a+b}} + \overline{b + \overline{a+b}}
 \end{aligned}$$



Q $y = \overline{abc} + ab$ using NOR gate

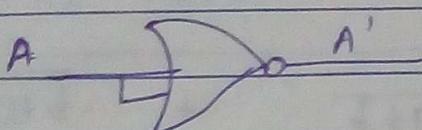
$$y = \overline{\overline{a}\overline{b}\overline{c}} + \overline{ab}$$

$$y = \overline{a + b + c} + \overline{\overline{a} + \overline{b}}$$



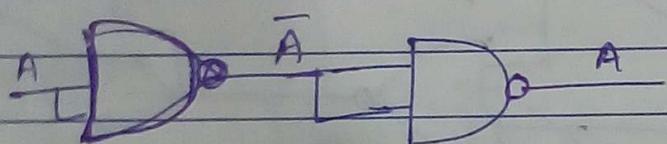
Q) determine min. no. of NOR gate required to implement following functions

$$f = A' + A'BC$$
$$A'(1+BC) = A'$$



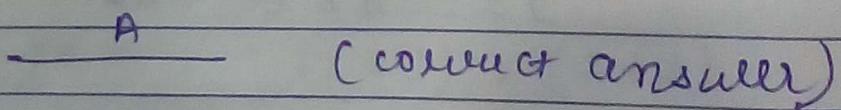
1-NOR gate

$$f = A + A\bar{B}\bar{C}$$
$$= A(1 + \bar{B}\bar{C}) = A$$



2- NAND

No gate required bcoz single variable is there and hence can be given through a wire.



(correct answer)