

Logic Gates - II

Comprehensive Course on Digital Logic Design 2023/2024



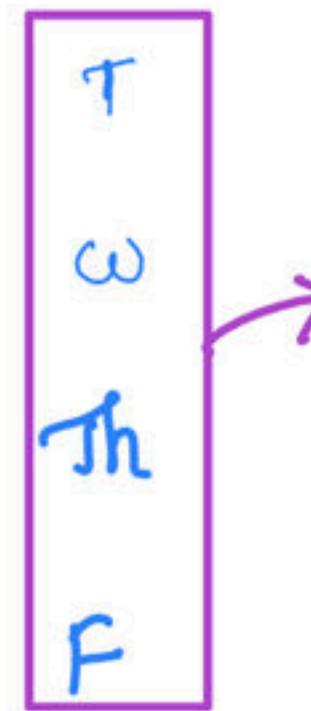
DIGITAL LOGIC DISIGN

(CS IT)

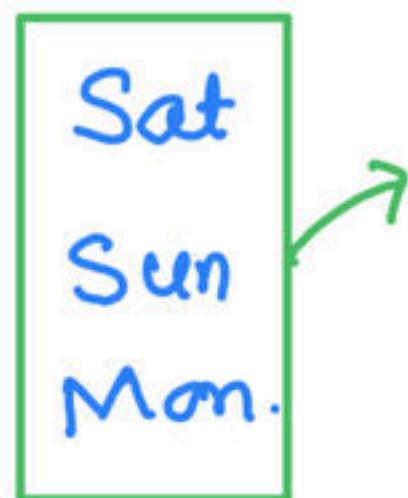
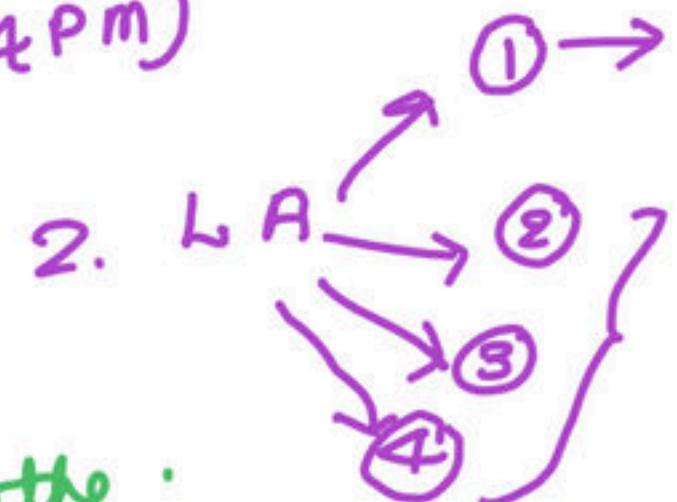


$$12 \times 2 = 24 \text{ hrs}$$

Engg. mathty



Digital
(2pm-4pm)



Engg. mathty
(8am-4pm)



Logic Gates

Logic gates are basic building blocks of digital circuits

Basic Gates

AND GATE

OR GATE

NOT GATE

Universal Gates

NAND GATE

NOR GATE

Derived Gates

EX- OR GATE

EX-NOR GATE

NOT Gate

Symbol

Truth table

Boolean expression

Switching circuit

A	Y
OFF	
ON	

Timing Diagram



NOTE:

1. The number of not gates present in the feedback only decide the nature of the logic circuit.
2. If the number of inverters in the feedback is even then ---->
3. If the number of inverters in the feedback is odd then --->

Time period (T) =

AND Gate

Symbol

Truth table

A	B	Y

If any one of the input is '0' then the output is '0'

Switching Circuit

A	B	Y
Off	Off	
Off	On	
On	Off	
On	On	

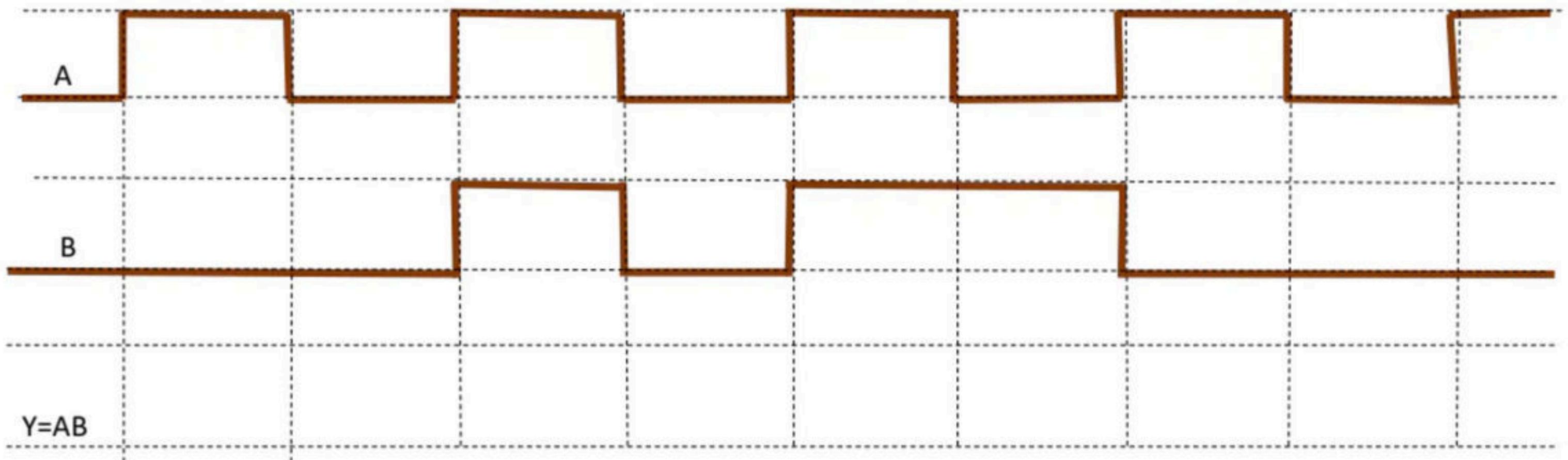
Enable input and Disable input

Commutative Law

Associative Law

Unused input in AND Gate

Timing Diagram



OR Gate

Symbol

Truth table

A	B	Y

If any one of the input is ‘1’ then the output is ‘1’

Switching Circuit

A	B	Y
Off	Off	
Off	On	
On	Off	
On	On	

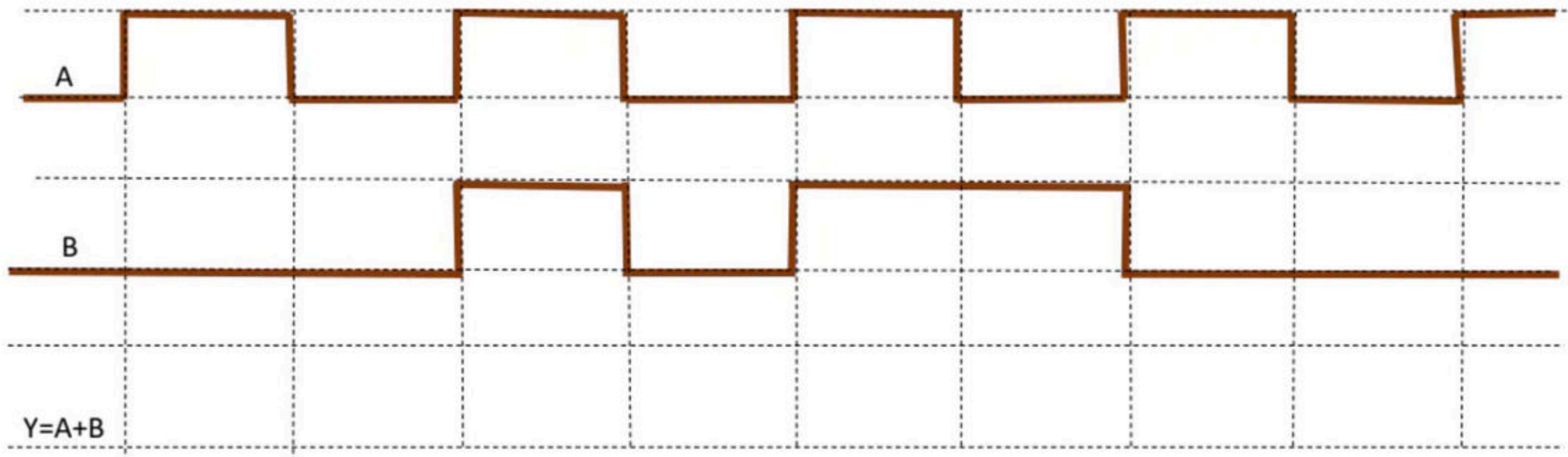
Enable input and Disable input

Commutative Law

Associative Law

Unused input in OR Gate

Timing Diagram

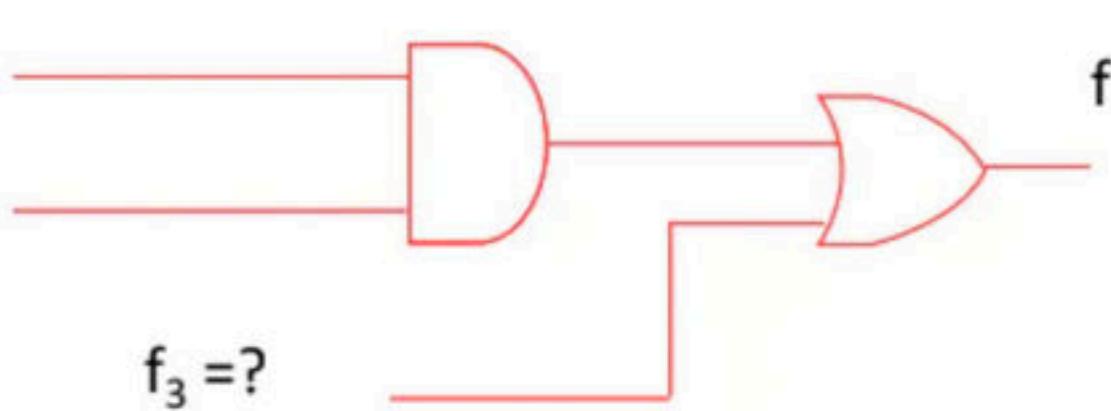


Q. Consider the logic circuit shown in the figure below. The function f_1 , f_2 and f (In canonical sum of products form in decimal notation) are

$$f_1(w, x, y, z) = \sum m(8, 9, 10)$$

$$f_2(w, x, y, z) = \sum m(7, 8, 12, 13, 14, 15)$$

$$f(w, x, y, z) = \sum m(8, 9)$$



The function f_3 is

(A) $\sum m(9, 10)$

(B) $\sum m(9)$

(C) $\sum m(1, 8, 9)$

(D) $\sum m(8, 10, 15)$

NAND Gate

Symbol

Truth table

A	B	Y

If any one of the input is ‘0’ then the output is ‘1’

Switching Circuit

A	B	Y
Off	Off	
Off	On	
On	Off	
On	On	

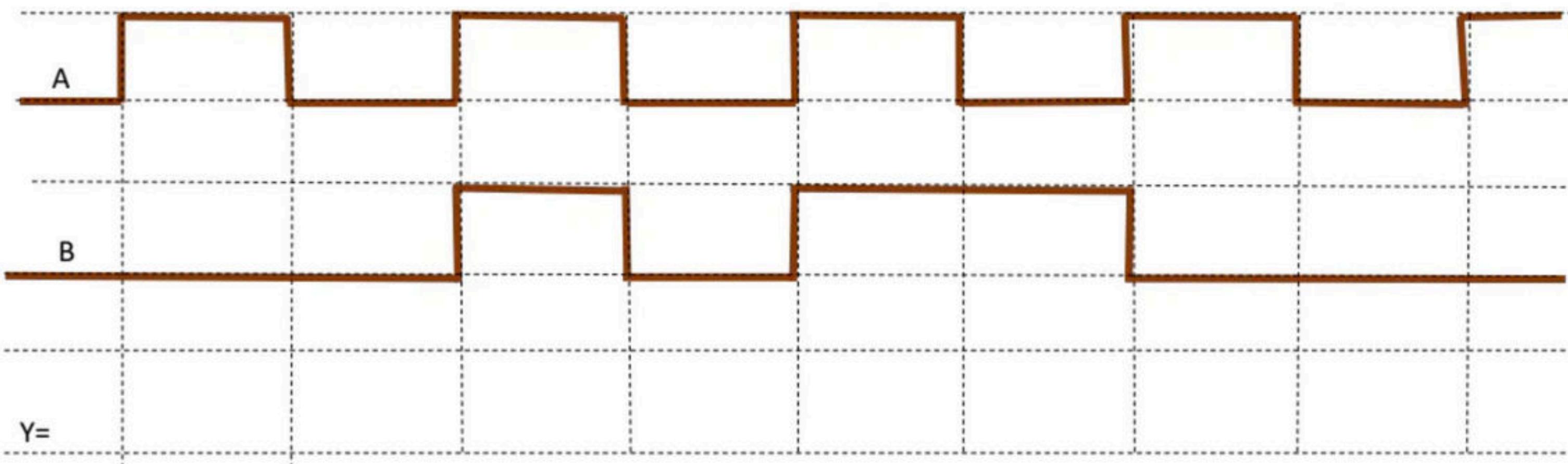
Enable input and Disable input

Commutative Law

Associative Law

Alternative Logic

Timing Diagram



NOR Gate

Symbol

Truth table

A	B	Y

If any one of the input is ‘1’ then the output is ‘0’

Switching Circuit

A	B	Y
Off	Off	
Off	On	
On	Off	
On	On	

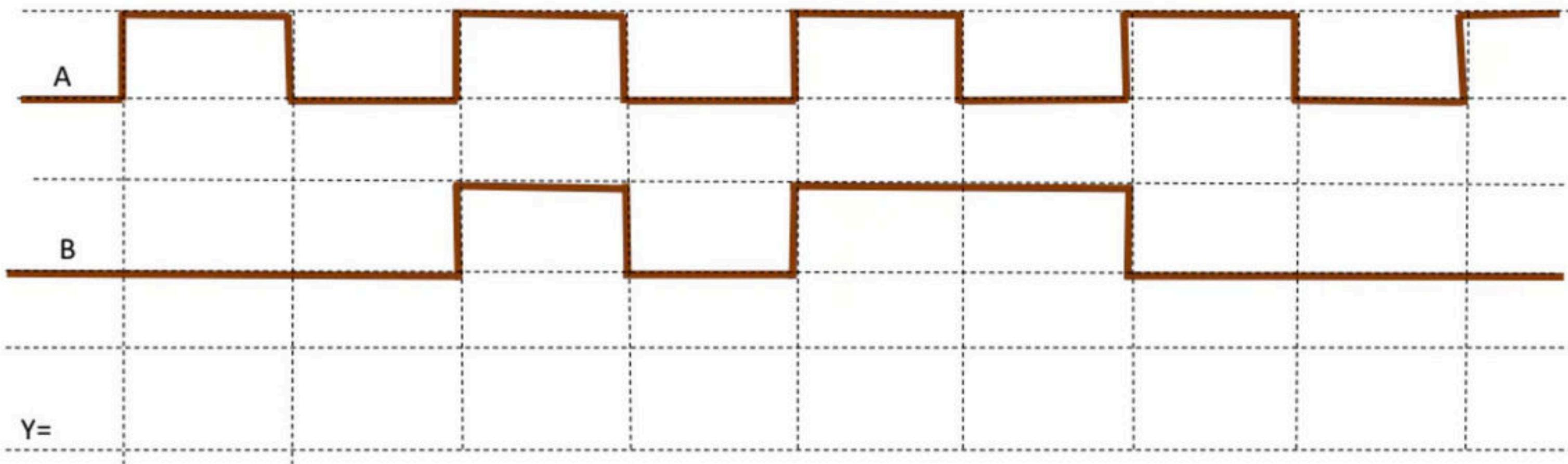
Enable input and Disable input

Commutative Law

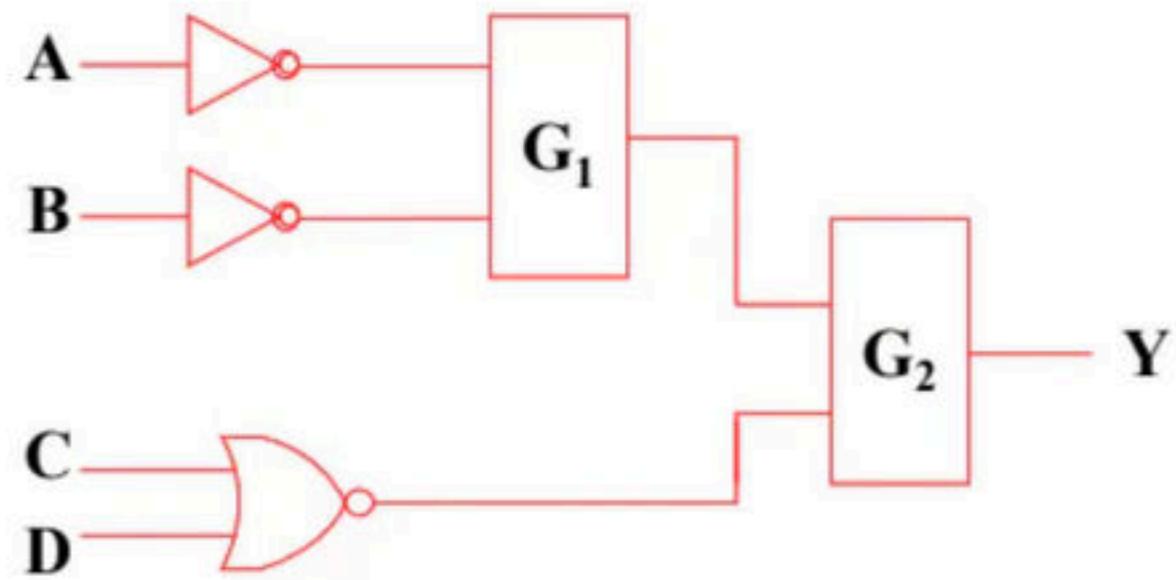
Associative Law

Alternative Logic

Timing Diagram



Q. In the figure shown, the output Y is required to be $Y = AB + \overline{C} \overline{D}$. The gates G₁ and G₂ must be, respectively,



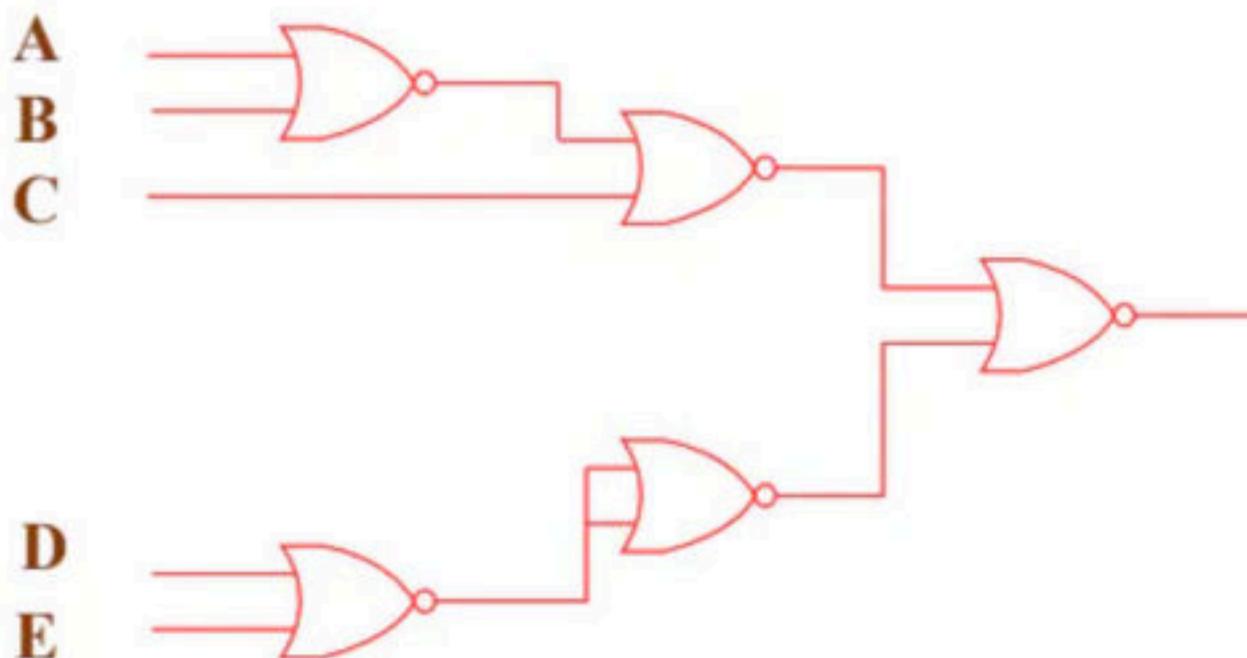
Q. The circuit shown in the figure realizes the function:

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

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

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

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



Q) The binary operator # is defined as $X \# Y = \bar{X} + \bar{Y}$, then which of the following is true

$$S_1 : P \# Q \# R = P \# (Q \# R)$$

$$S_2 : Q \# R = R \# Q$$

EX- OR Gate

Symbol

Truth table

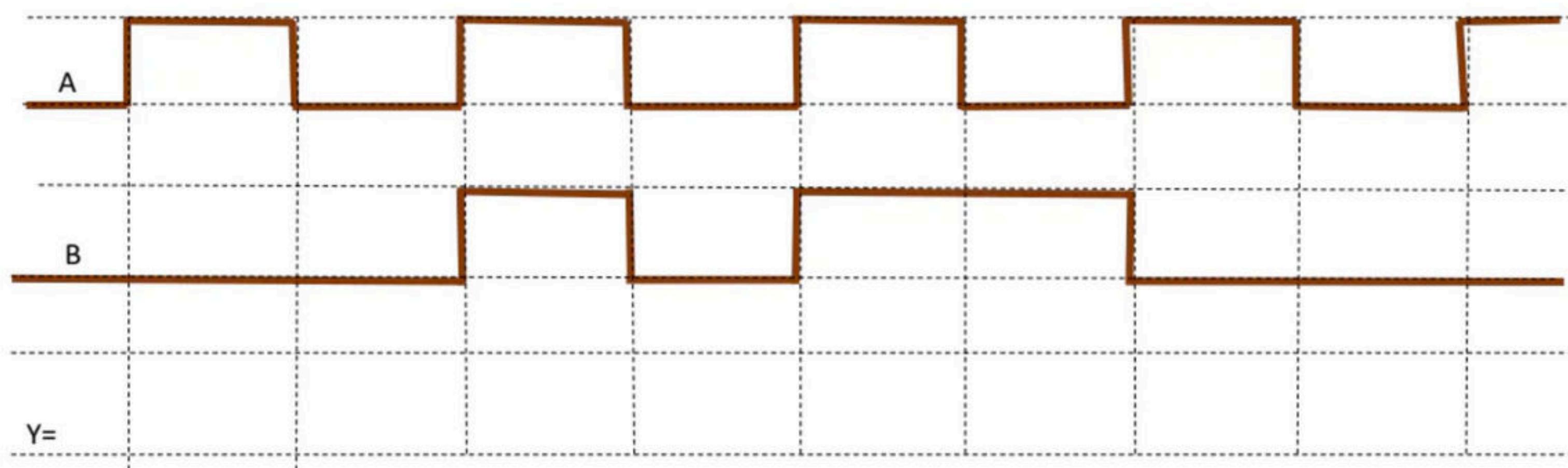
A	B	Y

If odd number of one's present then the output is '1'

Switching Circuit

A	B	Y
Off	Off	
Off	On	
On	Off	
On	On	

Timing Diagram



Commutative Law

Associative Law

EX- OR Gate as Buffer

EX- OR Gate as Inverter

Properties of EX- OR Gate

$$1. A \oplus 0 =$$

$$2. A \oplus 1 =$$

$$3. A \oplus A =$$

$$4. A \oplus \bar{A} =$$

5. $A \oplus A \oplus A \oplus A \dots \dots \dots \text{n - times} =$, **n – odd**

=, **n – even**

$$\mathbf{6. A} \oplus \overline{A}B =$$

$$7. \mathbf{AB} \oplus BC =$$

Q) Simplify the following

$$F = x \oplus y \oplus xy$$

Q) Simplify the following

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

EX-NOR Gate

Symbol

Truth table

A	B	Y

If even number of one's present then the output is '1'

Switching Circuit

A	B	Y
Off	Off	
Off	On	
On	Off	
On	On	

A	B	C	$Y = A \odot B \odot C$	$Y = A \oplus B \oplus C$	$Y = (A \odot B)$	$Y = (A \odot B) \odot C$	$Y = (A \odot C)$	$Y = (A \odot C) \odot B$
0	0	0						
0	0	1						
0	1	0						
0	1	1						
1	0	0						
1	0	1						
1	1	0						
1	1	1						

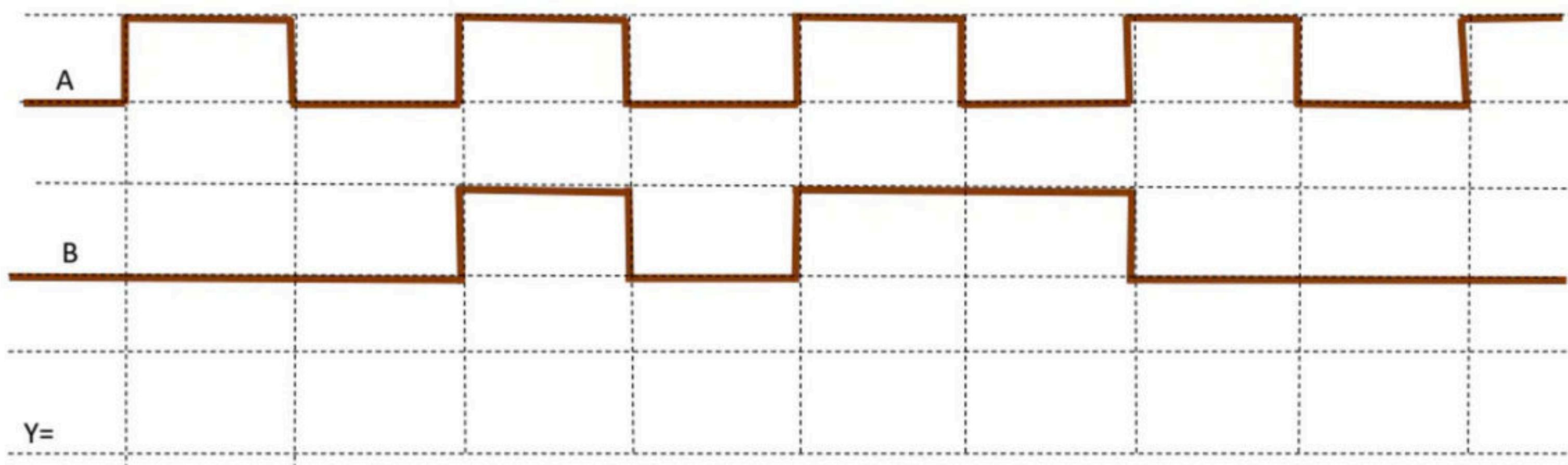
Commutative Law

Associative Law

EX-NOR Gate as Buffer

EX-NOR Gate as Inverter

Timing Diagram



Properties of EX- NOR Gate

$$1. A \odot 0 =$$

$$2. A \odot 1 =$$

$$3. A \odot A =$$

$$4. A \odot \bar{A} =$$

$$5. A \odot A \odot A \odot A \dots\dots n -\text{times} = \begin{cases} n -\text{even} \\ n -\text{odd} \end{cases}$$

$$6. \overline{\mathbf{A} \odot \mathbf{B}} =$$

$$7. \mathbf{A} \oplus \overline{\mathbf{B}} =$$

$$8. \overline{\mathbf{A}} \oplus \mathbf{B} =$$

$$9. \overline{\mathbf{A}} \oplus \overline{\mathbf{B}} =$$

$$10. \bar{A} \odot \bar{B} =$$

11. $A \odot \bar{B} =$

12. $\bar{A} \odot B =$

13. $\bar{A} \oplus \bar{B}$

$$14. \overline{A \odot B \odot C} =$$

$$15. \bar{A} \odot \bar{B} \odot \bar{C}$$

16. $A \odot B \odot C$

t

$$17. [A \oplus B] \odot C =$$

$$18. A \odot [B \oplus C] =$$

EX- OR Gate

OUTPUT = 1

For odd number of 1's

Odd number of 1's detector

Inequality detector

Anti coincident Gate

EX-NOR Gate

OUTPUT = 1

For even number of 1's

Even number of 1's detector

Equality detector

Coincident Gate

Q) Simplify the following

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

Q) Simplify the following

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

Q) Simplify the following

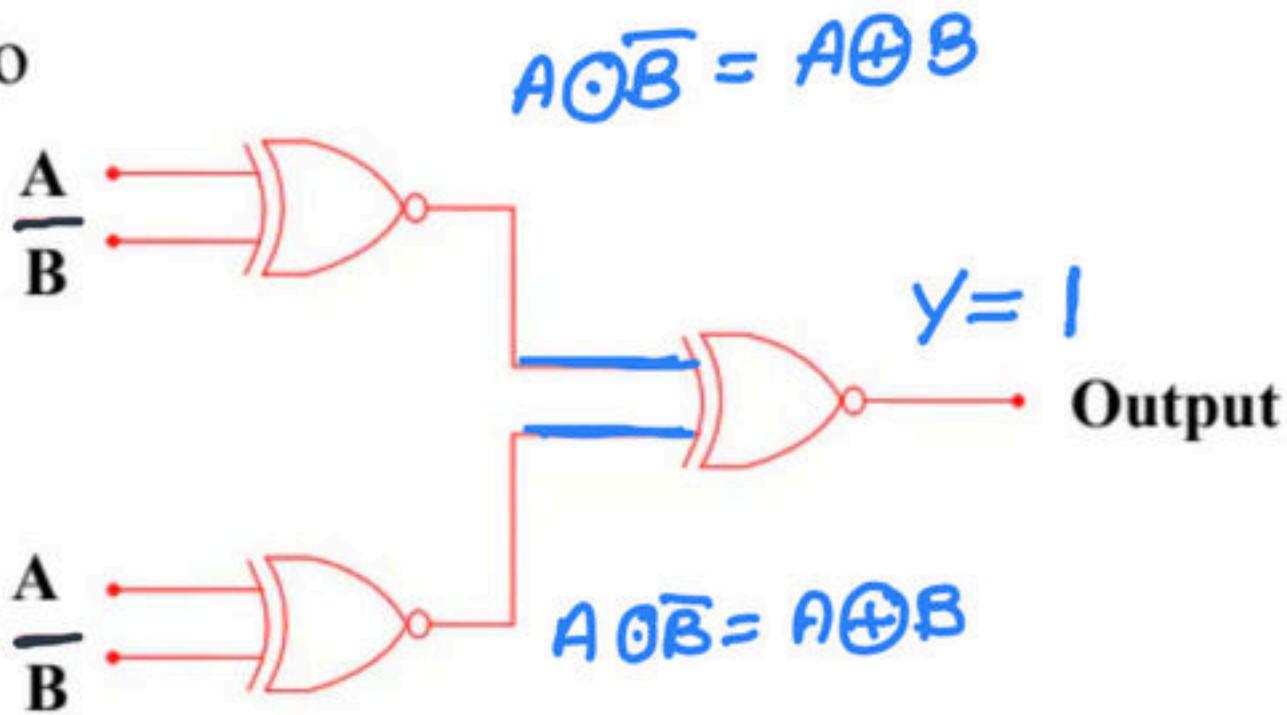
$$F = (A \oplus B) + (A \oplus \bar{B})$$

Q) Simplify the following

$$[(1 \oplus P) \oplus (P \oplus Q)] \oplus [(P \oplus Q) \oplus (Q \oplus 0)]$$

Q. The output of the circuit shown (in figure) is equal to

- (a) 0
- (b) 1**
- (c) $\overline{A}B + A\overline{B}$
- (d) $(\overline{A} * \overline{B}) * (\overline{A} * B)$



XNOR

Q. The output of the combinational circuit given below is,

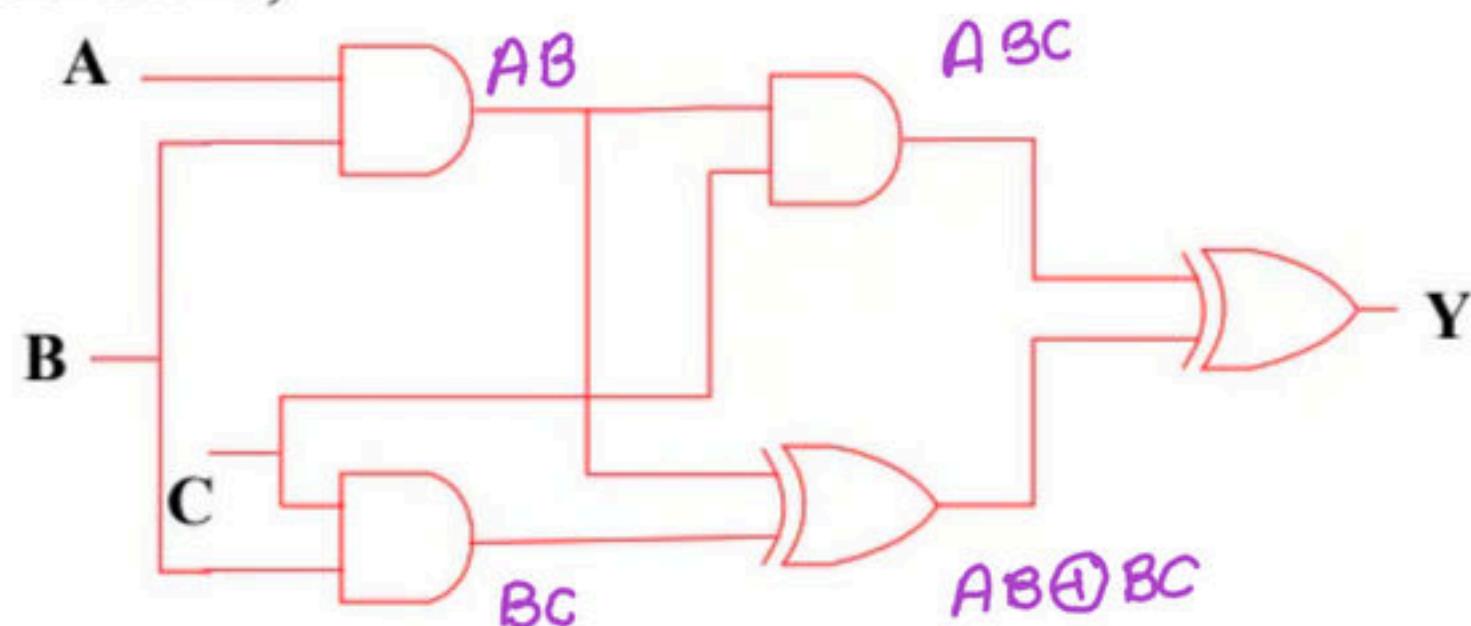
- (a) $A + B + C$
- (b) ~~$A(B + C)$~~
- (c) $B(C + A)$
- (d) $C(A + B)$

$$Y = ABC \oplus B(A \oplus C)$$

$$= ABC \oplus AB \oplus AC$$

$$= AB [C \oplus B] \oplus AC$$

$$= AB\bar{C} \oplus AC$$



$$Y = A[C \oplus BC\bar{C}]$$

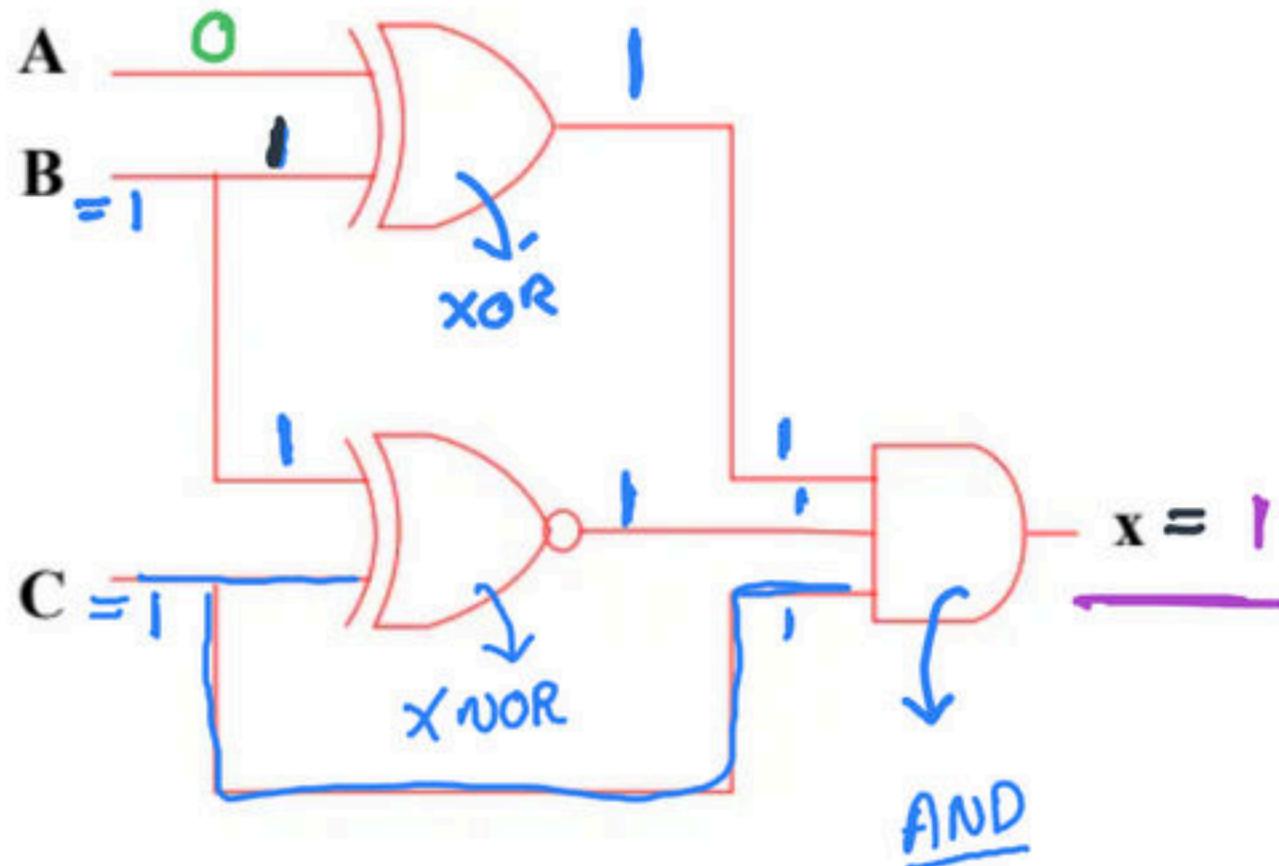
$$Y = A[C + BC\bar{C}]$$

$$Y = A[C + B]$$

Q. For the logic circuit shown in the given figure, the required input condition (A, B, C) to make the output (X)=1 is

- (a) 1, 0, 1
- (b) 0, 0, 1
- (c) 1, 1, 1
- (d) 0, 1, 1

A = 0
B = 1
C = 1

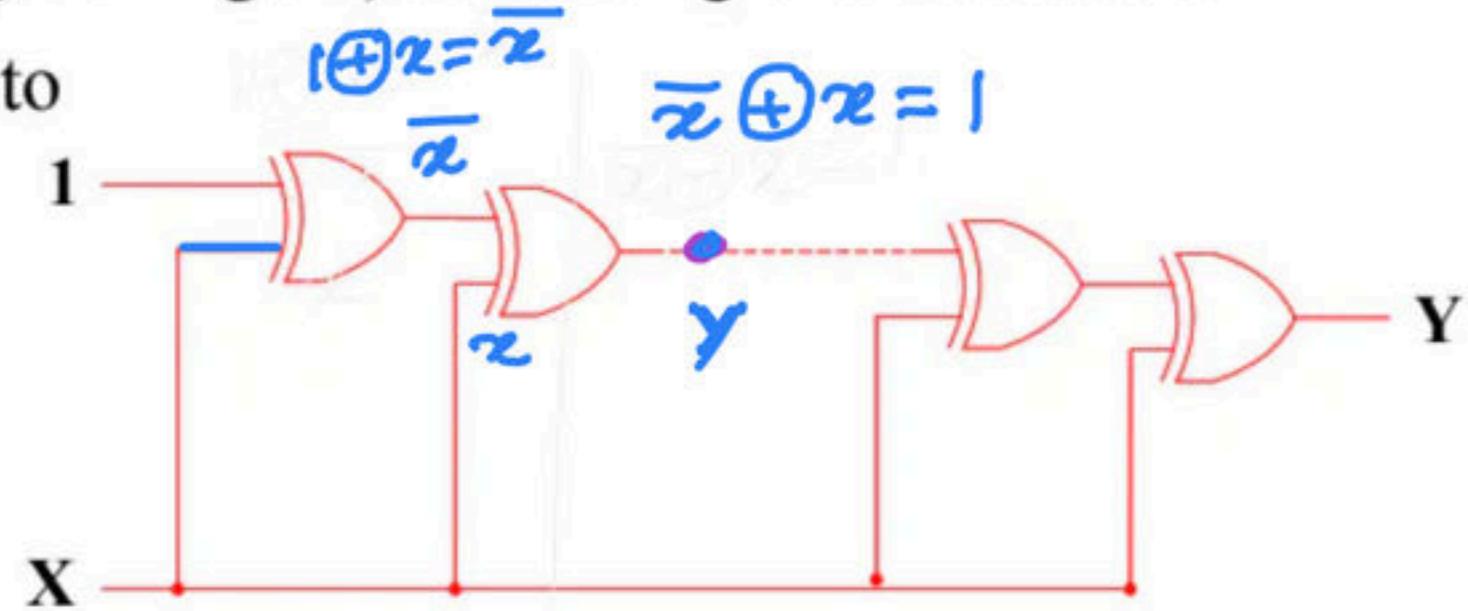


$$Y = ABC$$

Q. If the input to the digital circuit (shown in the given figure) consisting of a cascade of 20 XOR-gates is X, then the output Y is equal to

- (a) 0
- (b) 1
- (c) \bar{X}
- (d) X

$$\left. \begin{array}{l} n_{\text{even}} \\ n = 2^0 \end{array} \right| \quad n = 2$$



Q) Find the minterms of 3 variable EX-OR and EX-NOR gate

$$y(A, B, C) = A \oplus B \oplus C = \sum m(1, 2, 4, 7)$$

$$y(A, B, C) = A \ominus B \ominus C = \underline{\sum m(0, 3, 5, 6)}$$

Q) Find the minterms of 4 variable EX-OR and EX-NOR gate

$$y(A, B, C, D) = A \oplus B \oplus C \oplus D = \sum m(1, 2, 4, 7, 8, 11, 13, 14)$$

$$y(A, B, C, D) = A \ominus B \ominus C \ominus D = \sum m(0, 3, 5, 6, 9, 10, 12, 15)$$

Q. The output F in the digital logic circuit shown in the figure is

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

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

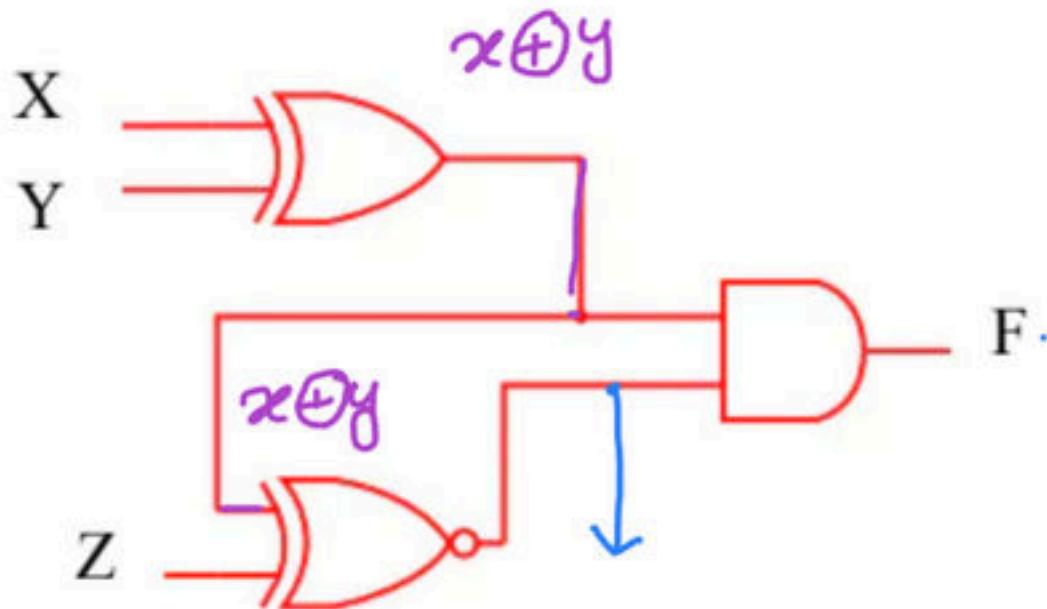
(c) $F = \overline{XYZ} + XYZ$

(d) $F = \overline{XYZ} + XYZ$

$$F = (\bar{x} \oplus y) ((x \oplus y) \odot z)$$

$$F = (\bar{x} \oplus y) \left[(\bar{x} \oplus y) \bar{z} + (\bar{x} \oplus y) z \right]$$

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



$$(x \oplus y) \odot z$$

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

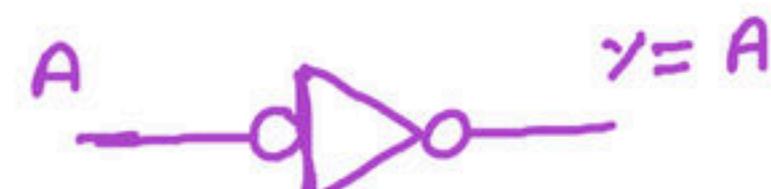
S.No

Logic gate

Alternative logic

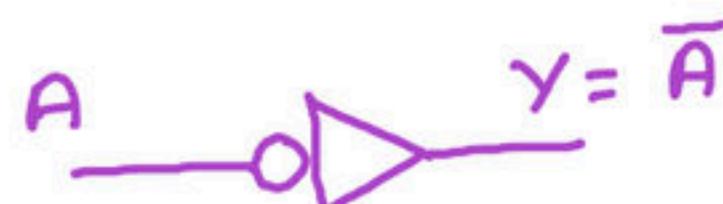
1.

Buffer



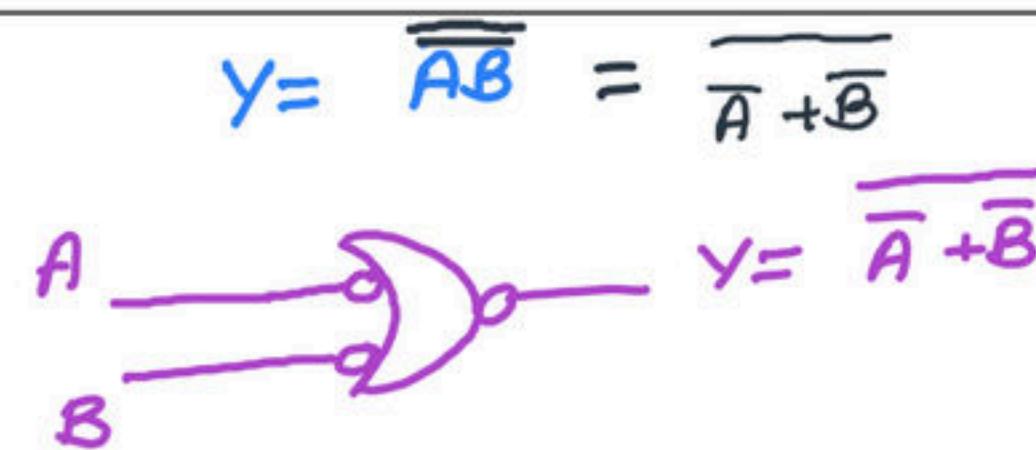
2

Not



3

AND



S.No

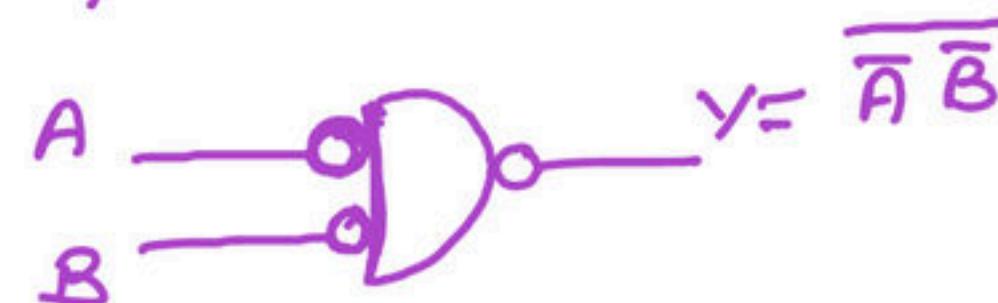
Logic gate

Alternative logic

4

OR

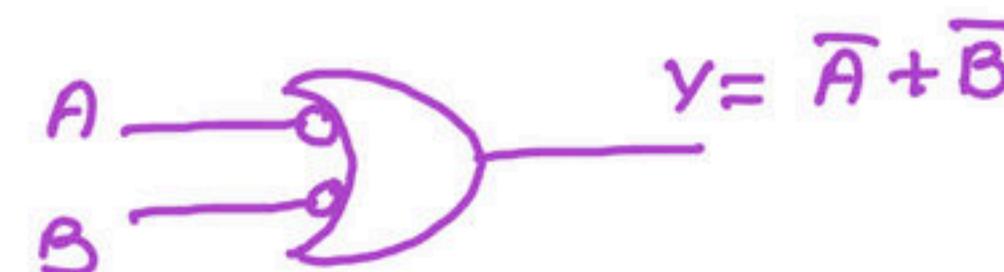
$$y = \overline{A+B} = \overline{A} \cdot \overline{B}$$



5

NAND

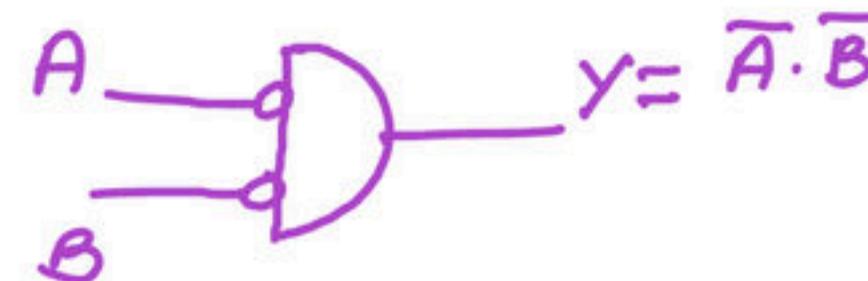
$$y = \overline{AB} = \overline{A} + \overline{B}$$



6

NOR

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



S.No

Logic gate

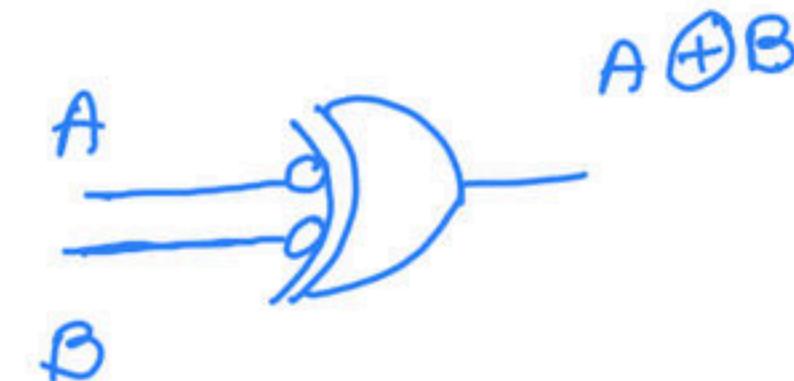
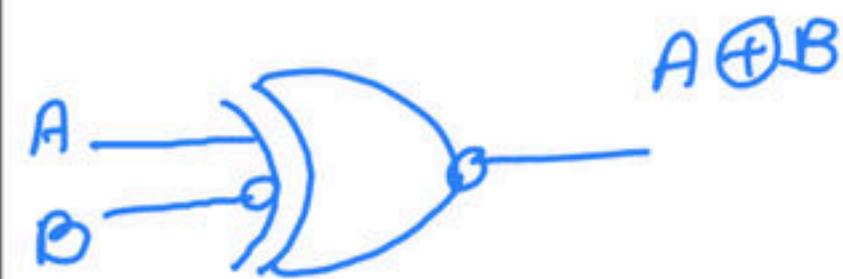
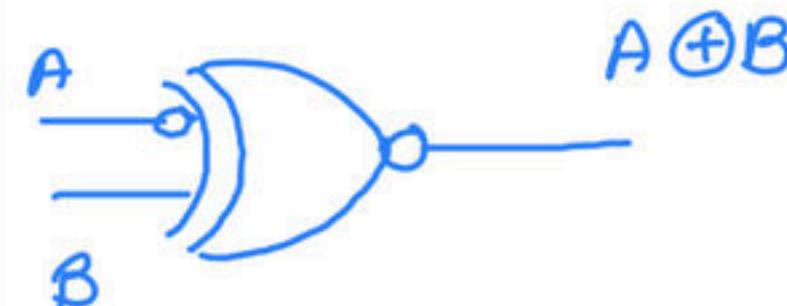
7

EX-OR

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

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

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



S.No

Logic gate

8

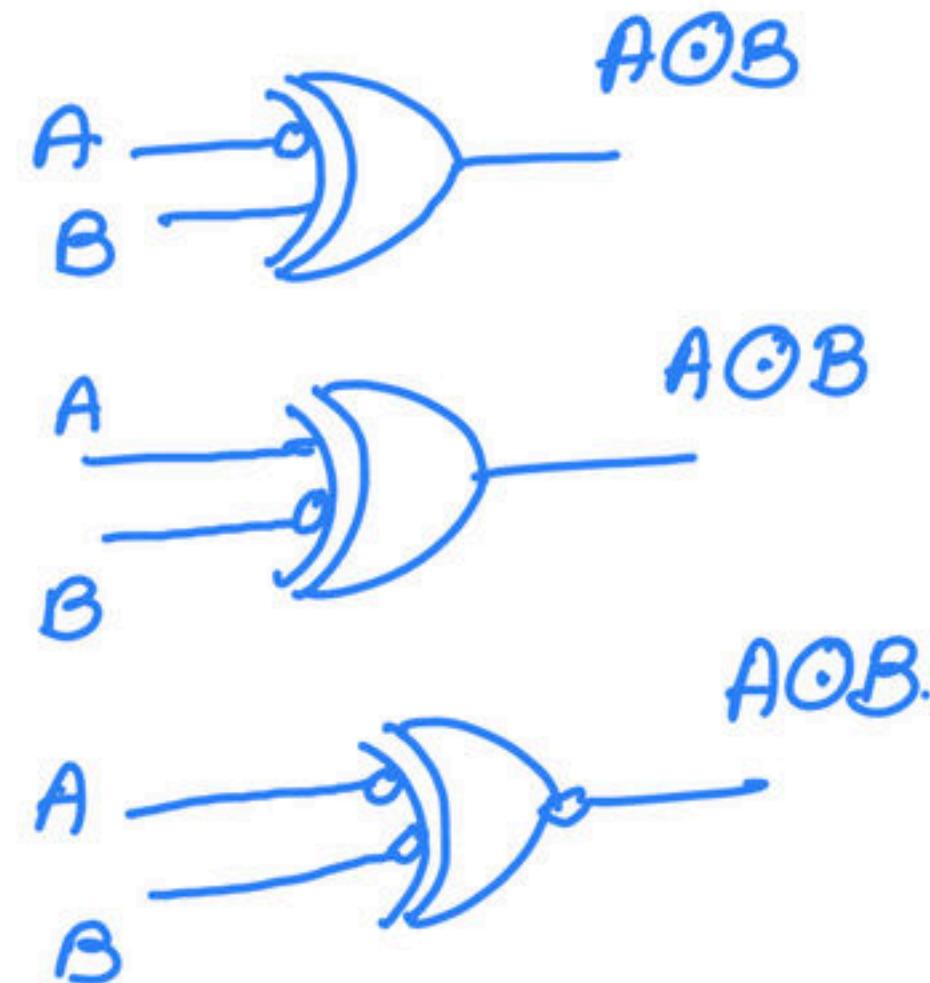
EX-NOR

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

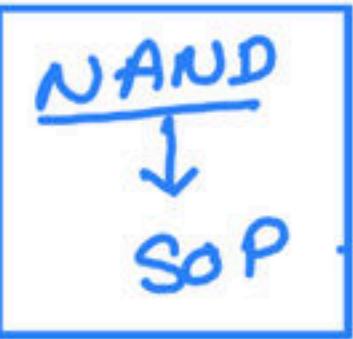
$$A \oplus \overline{B} = A \odot B$$

$$\overline{A} \odot \overline{B} = A \odot B$$

Alternative logic



Q) Implement using NAND gates $Y=AC+BC+AB$

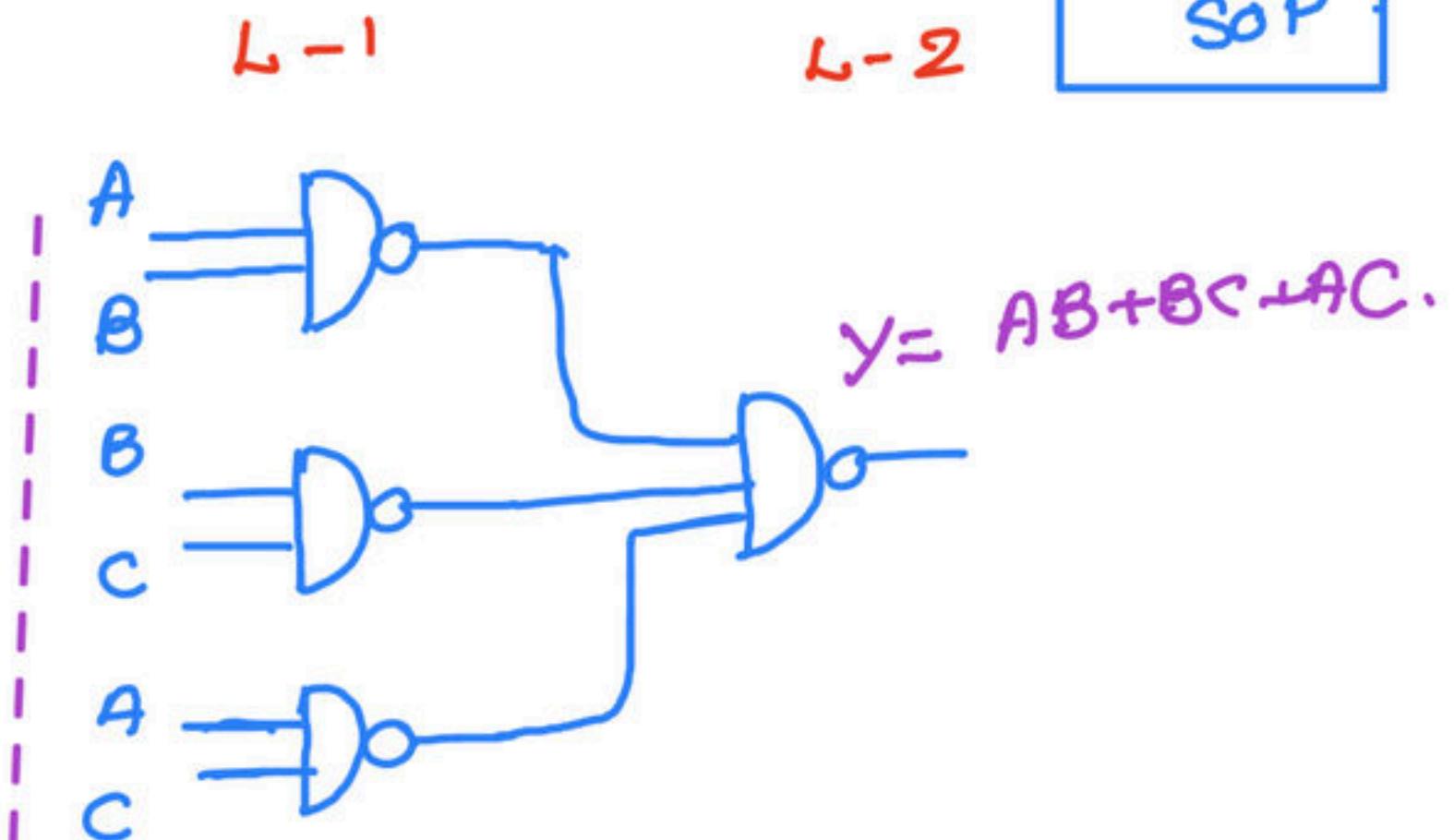
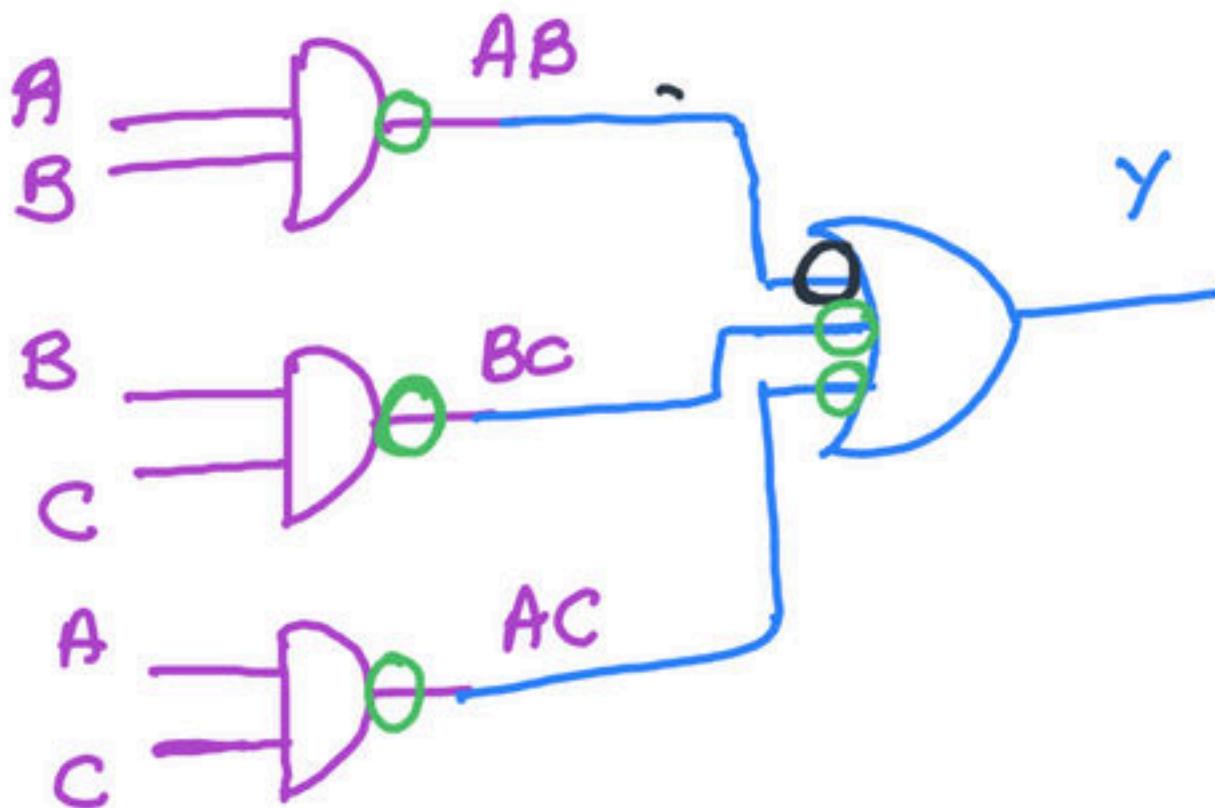


Method -1

$$y = AB + BC + AC$$

L-1

L-2



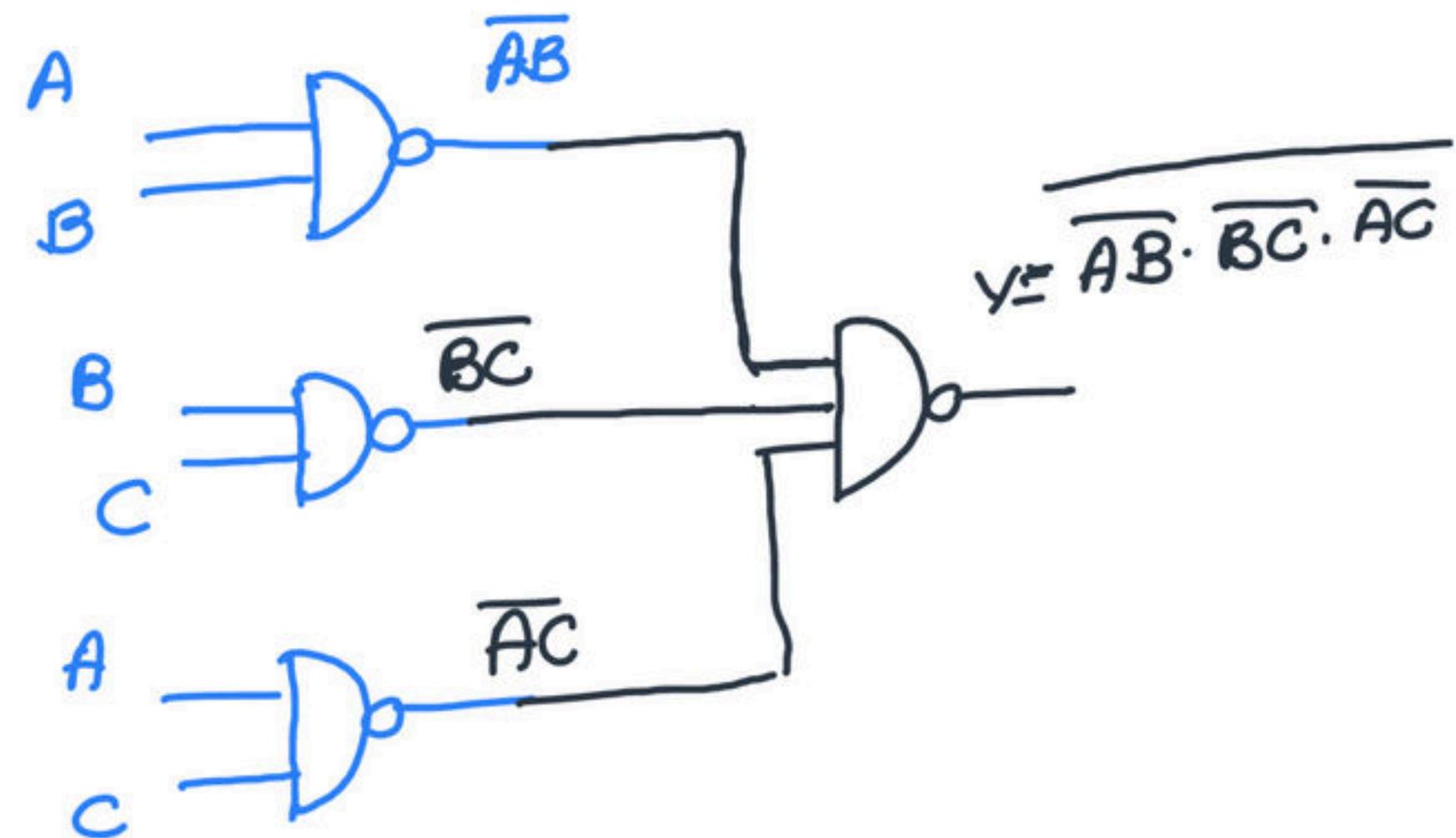
2-Level AND-OR logic \equiv 2-Level
NAND-NAND logic.

Method-2

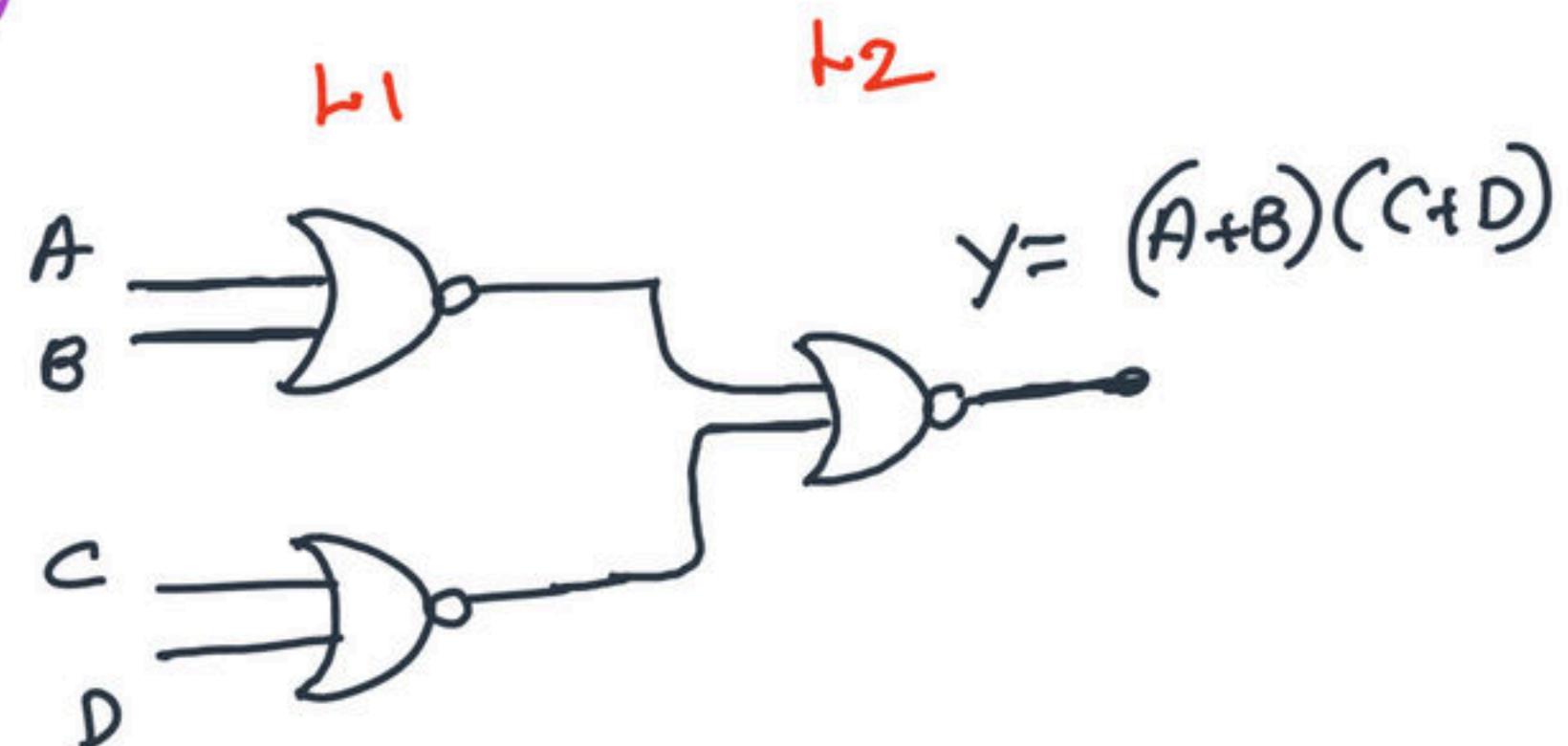
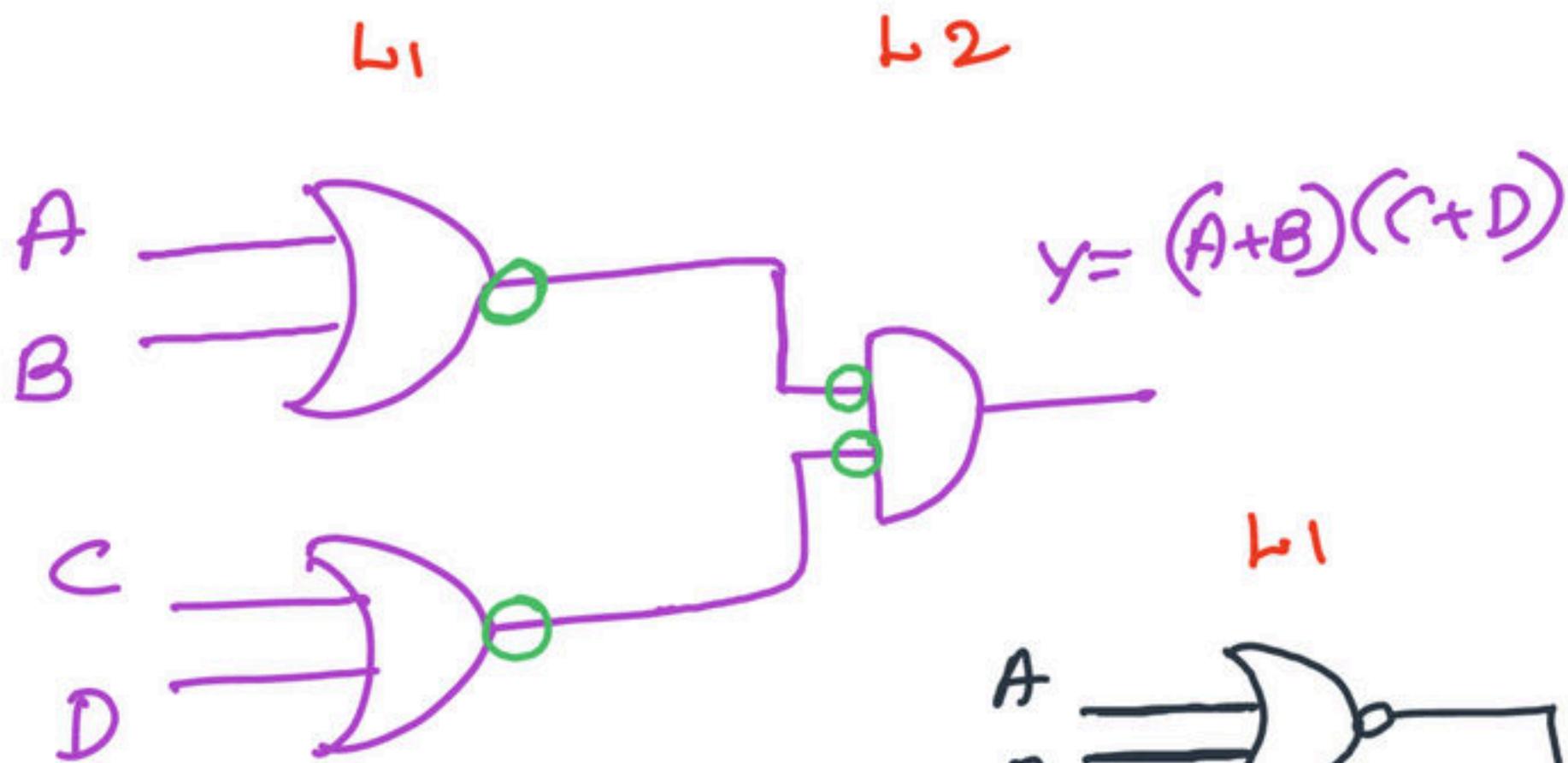
$$y = AB + BC + AC$$

$$y = \overline{AB + BC + AC}$$

$$y = \overline{AB} \cdot \overline{BC} \cdot \overline{AC}$$



Q) Implement using NOR gates $Y = (A+B)(C+D)$



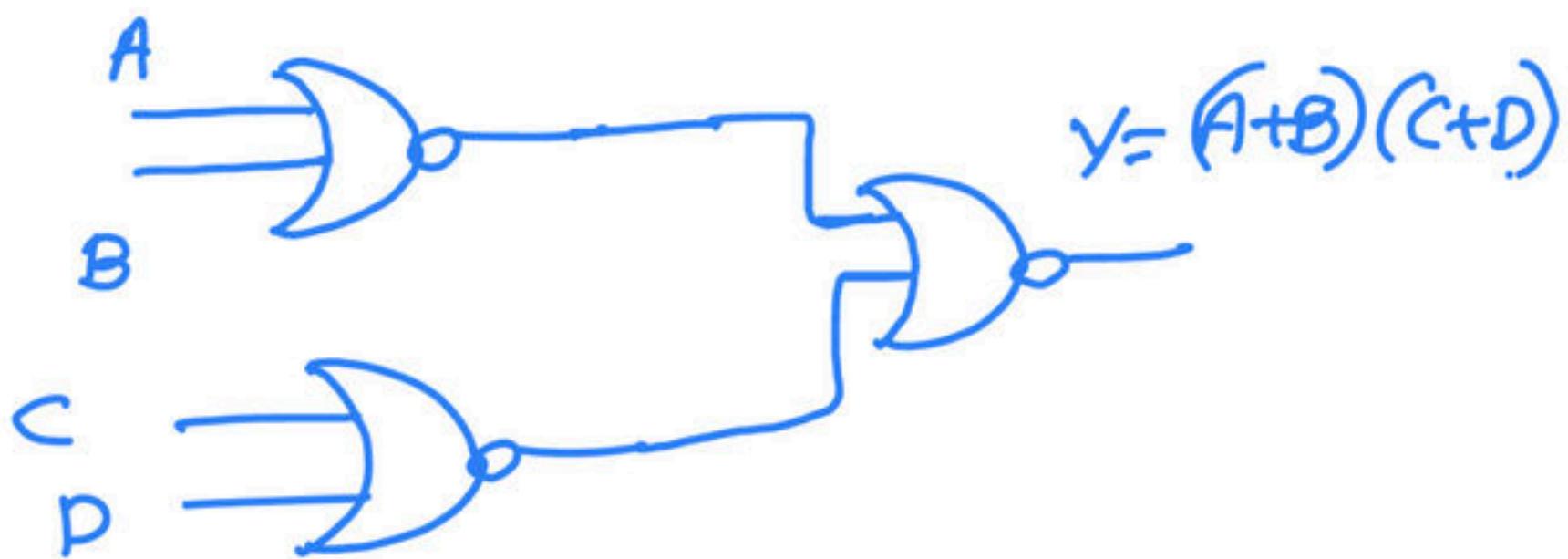
2-Level OR-AND logic
≡ 2-Level NOR-NOR logic

$$y = \overline{(A+B)(C+D)}$$

NOR
↓
POS.

$$y = \overline{\overline{(A+B)(C+D)}}$$

$$y = \overline{\overline{(A+B)}} + \overline{\overline{(C+D)}}$$



Note :

- 2- level **AND-OR** logic \equiv 2- level **NAND -NAND** logic
- 2- level **OR- AND** logic \equiv 2- level **NOR- NOR** logic

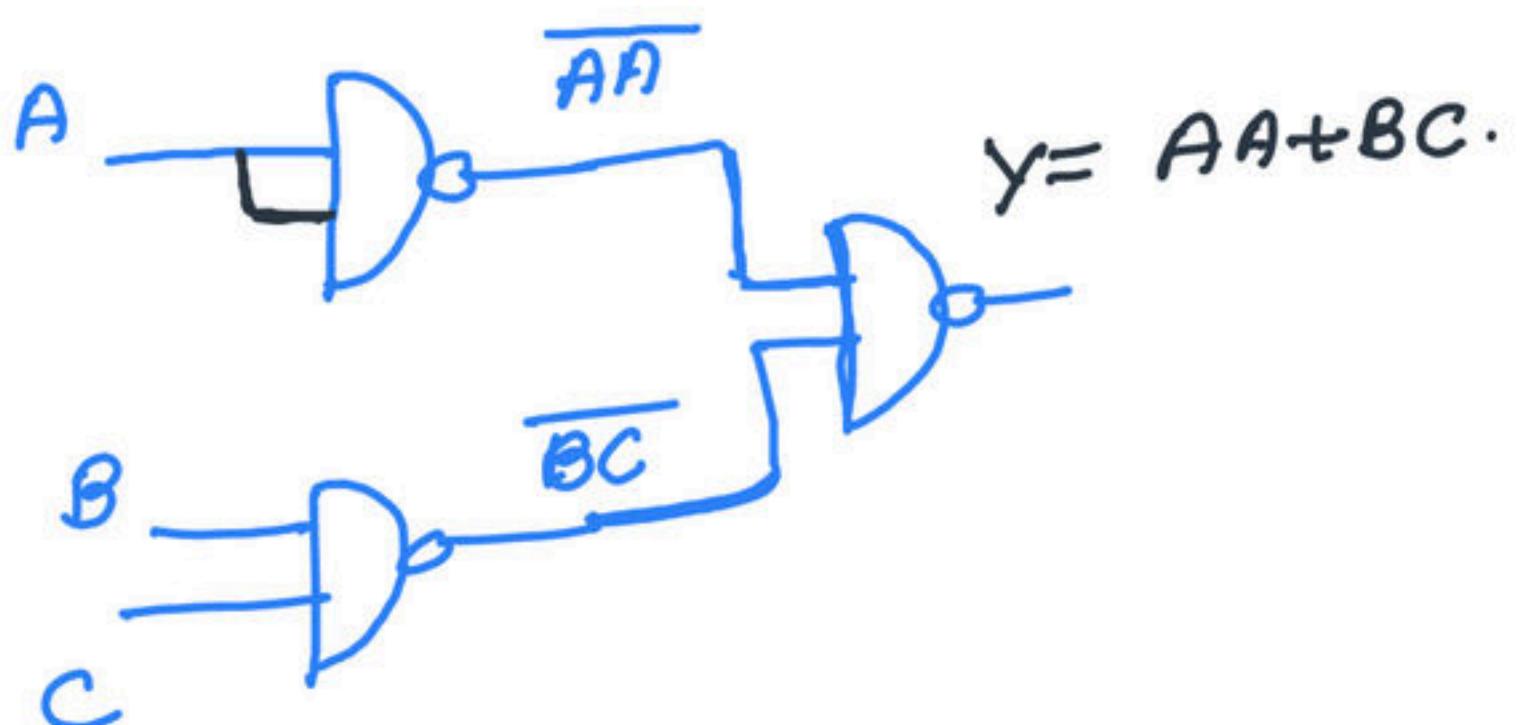
Q) $Y = A + BC$ implement using NAND gates

NAND
↓
SOP

$$Y = AA + BC$$

$$Y = \overline{\overline{AA} + BC}$$

$$Y = \overline{\overline{AA}} \cdot \overline{\overline{BC}}$$



③

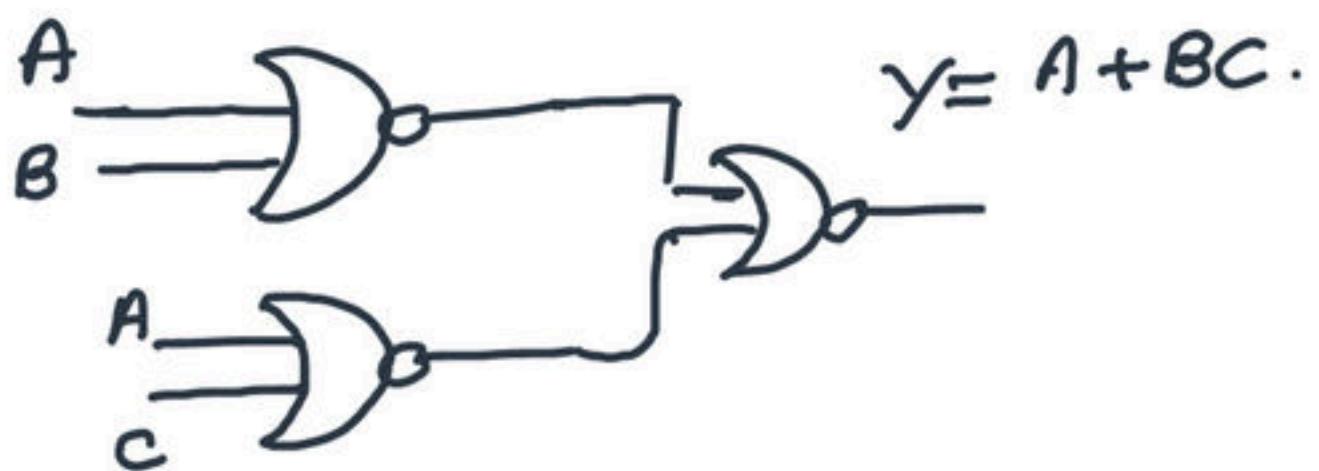
Q) $Y = A + BC$ implement using NOR gates

$$Y = \overline{(A+B)(A+C)} \rightarrow \text{pos.}$$

NOR
↓
pos

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

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



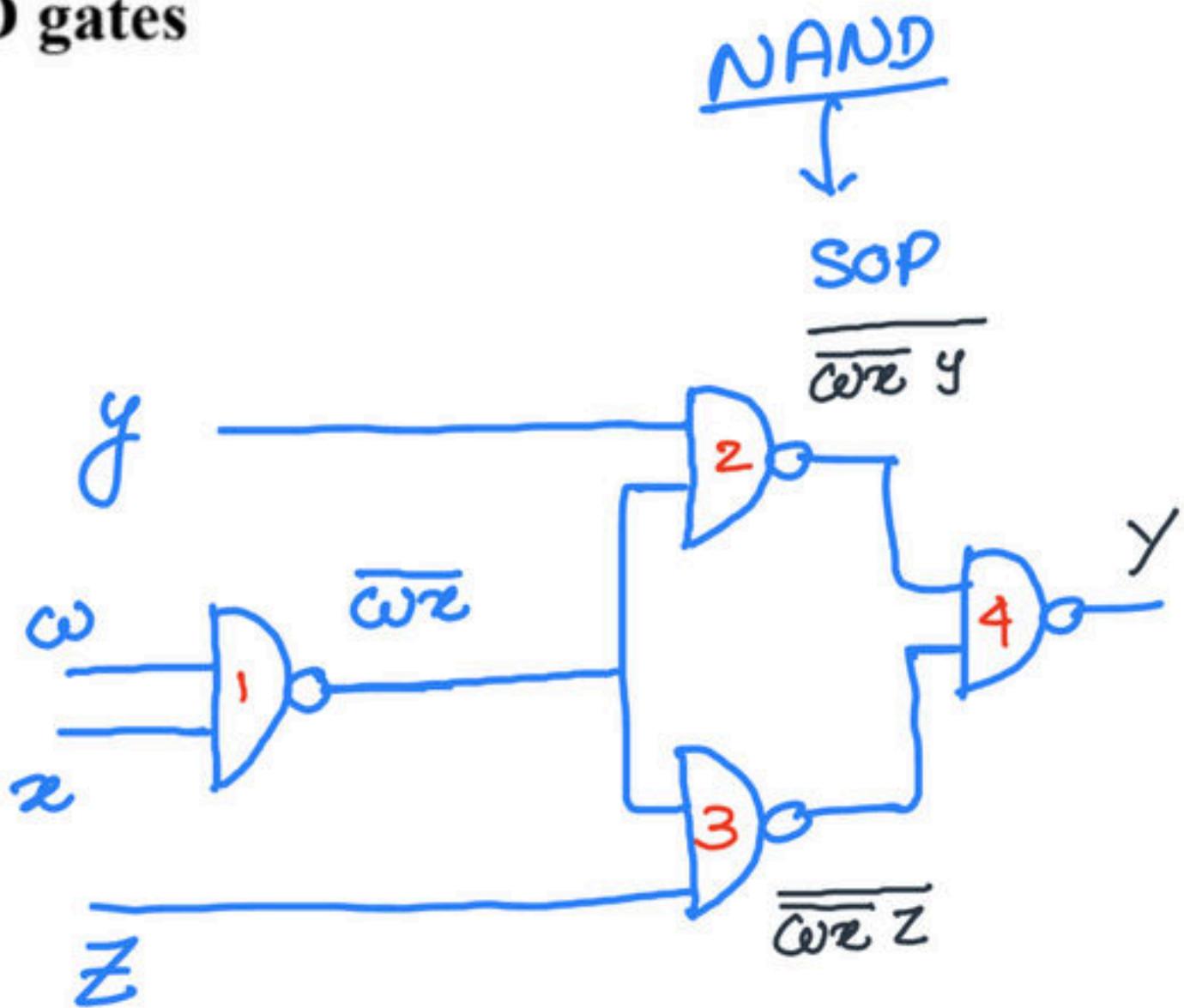
Q) $Y = (\bar{W} + \bar{X})(Y + Z)$ implement using NAND gates

$$Y = \overline{\omega x} Y + \overline{\omega x} Z$$

$$Y = \overline{\overline{\omega x}} Y + \overline{\overline{\omega x}} Z$$

$$Y = \overline{\overline{\omega x}} Y \cdot \overline{\overline{\omega x}} Z$$

④



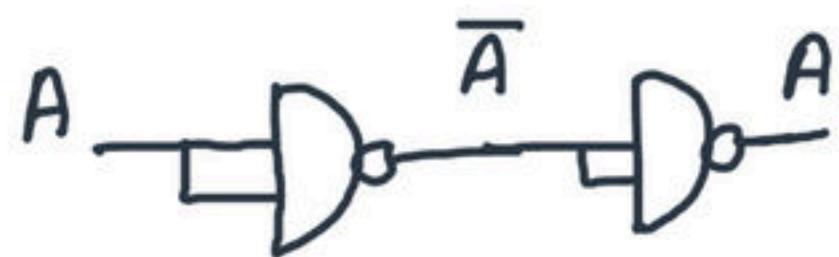
$$Y = \overline{\overline{\omega x}} Y \cdot \overline{\overline{\omega x}} Z$$

Universal Gates

- ❖ NAND and NOR gates are called as universal gates , because by using NAND and NOR gates , we can implement any Boolean expression .

NAND Gate as Universal Gate

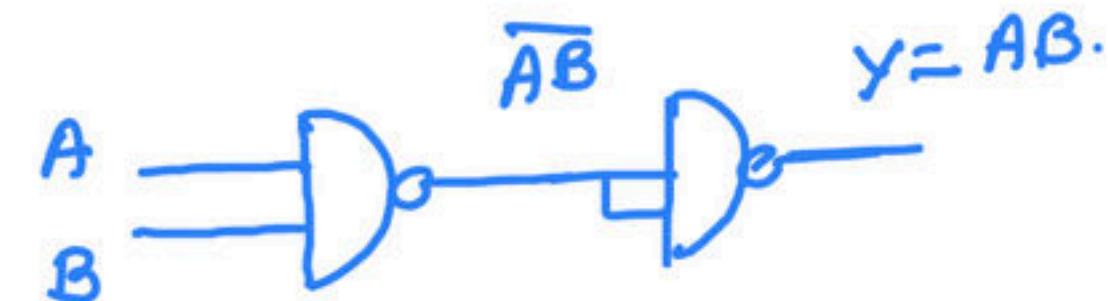
1. BUFFER GATE



2. NOT GATE



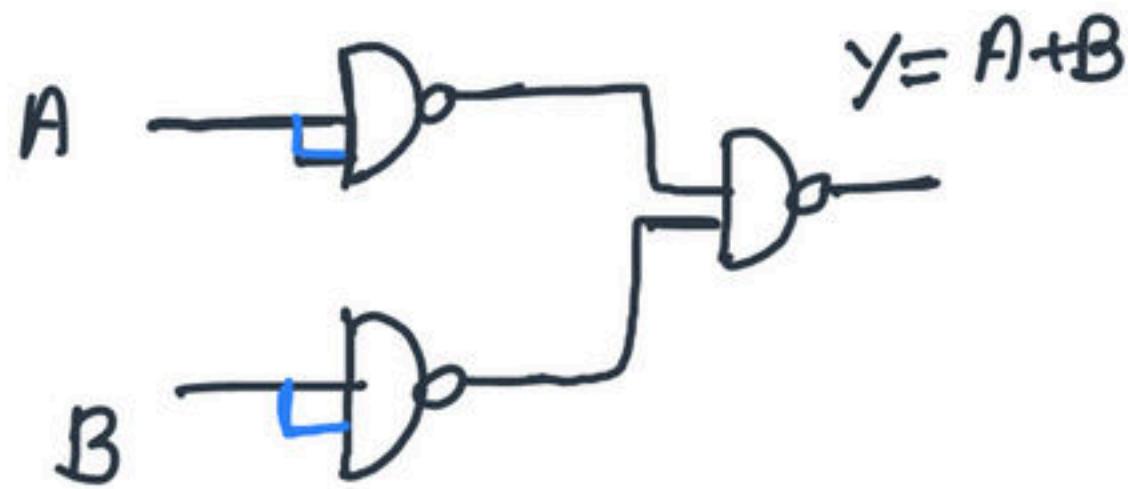
3. AND GATE



4. OR GATE

$$y = A + B$$

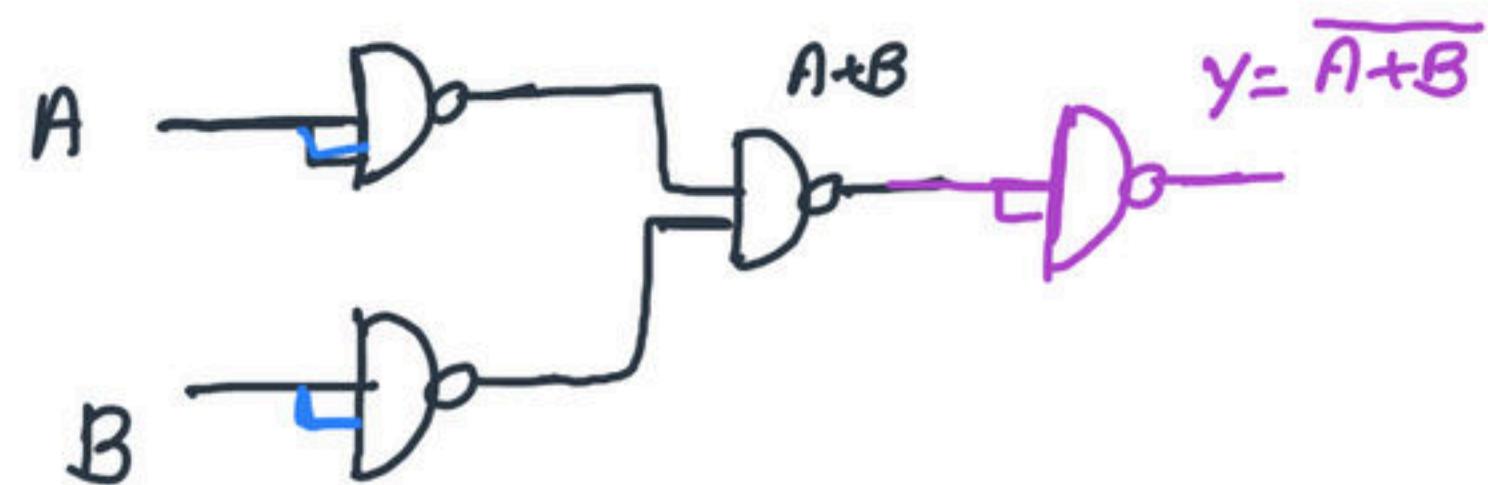
$$y = \overline{\overline{A+B}} = \overline{\overline{A} \cdot \overline{B}}$$



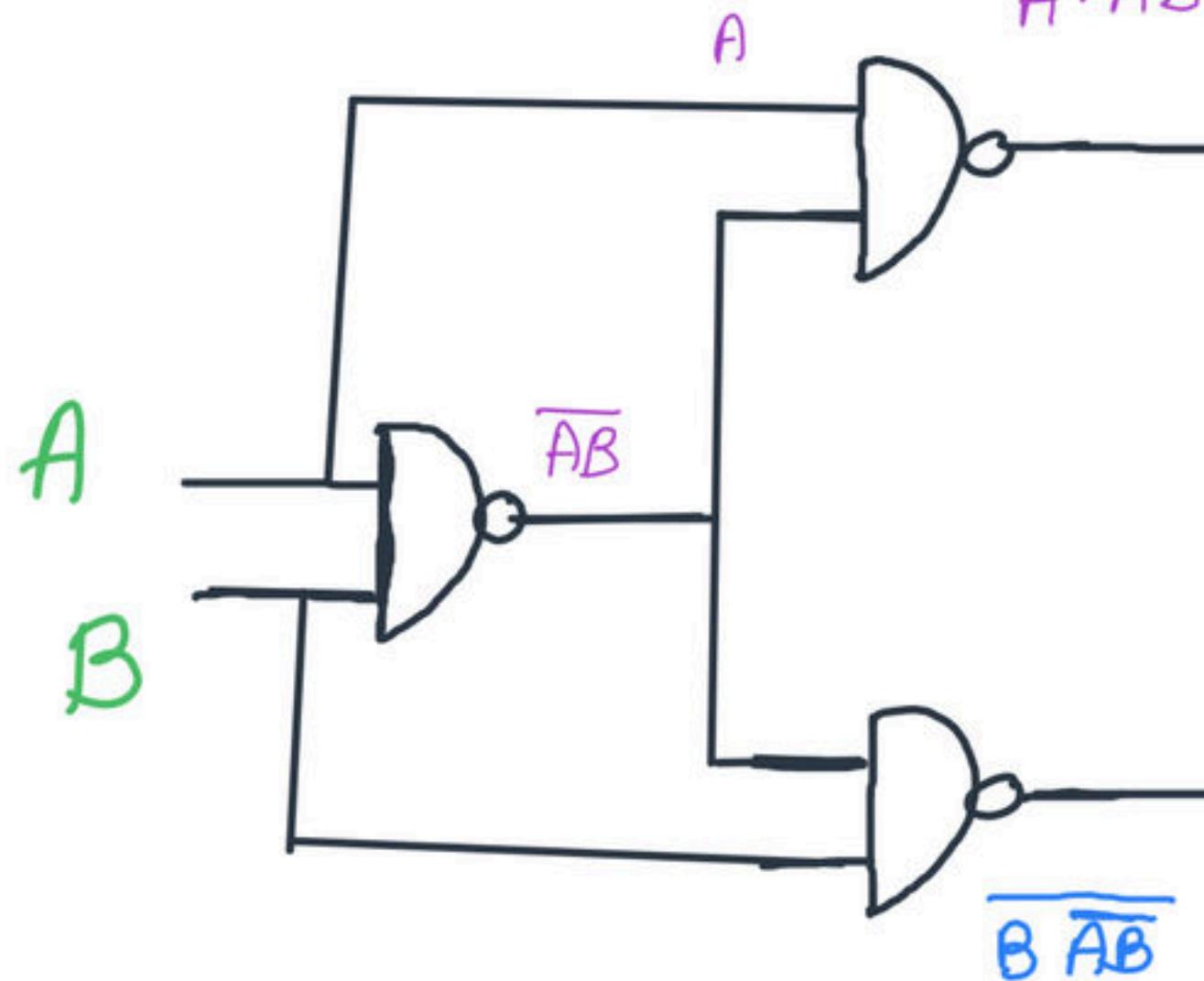
③

5. NOR GATE

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



6. EX-OR GATE



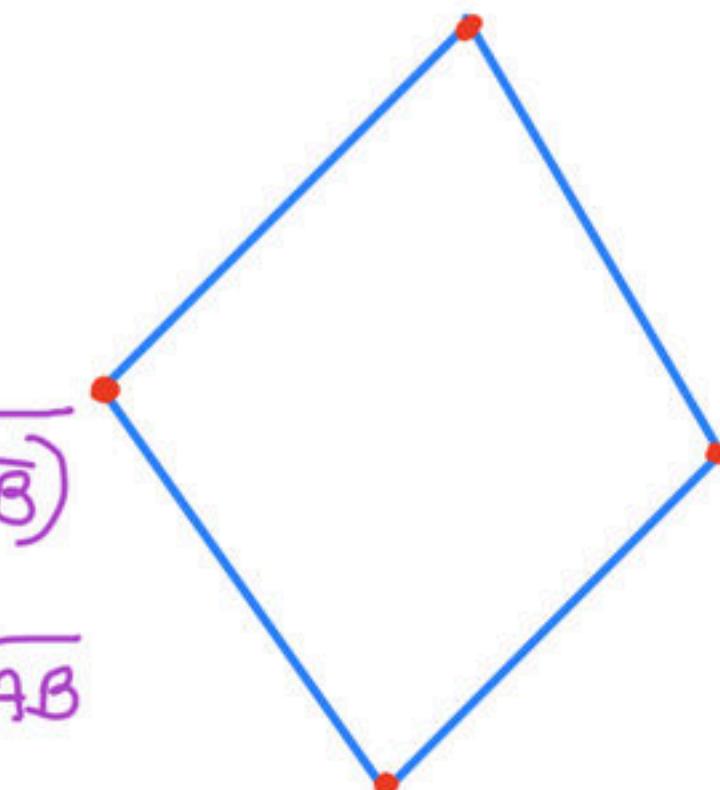
$$\overline{A \cdot \overline{AB}} = \overline{\overline{A}} + AB = \overline{A} + B$$

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

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

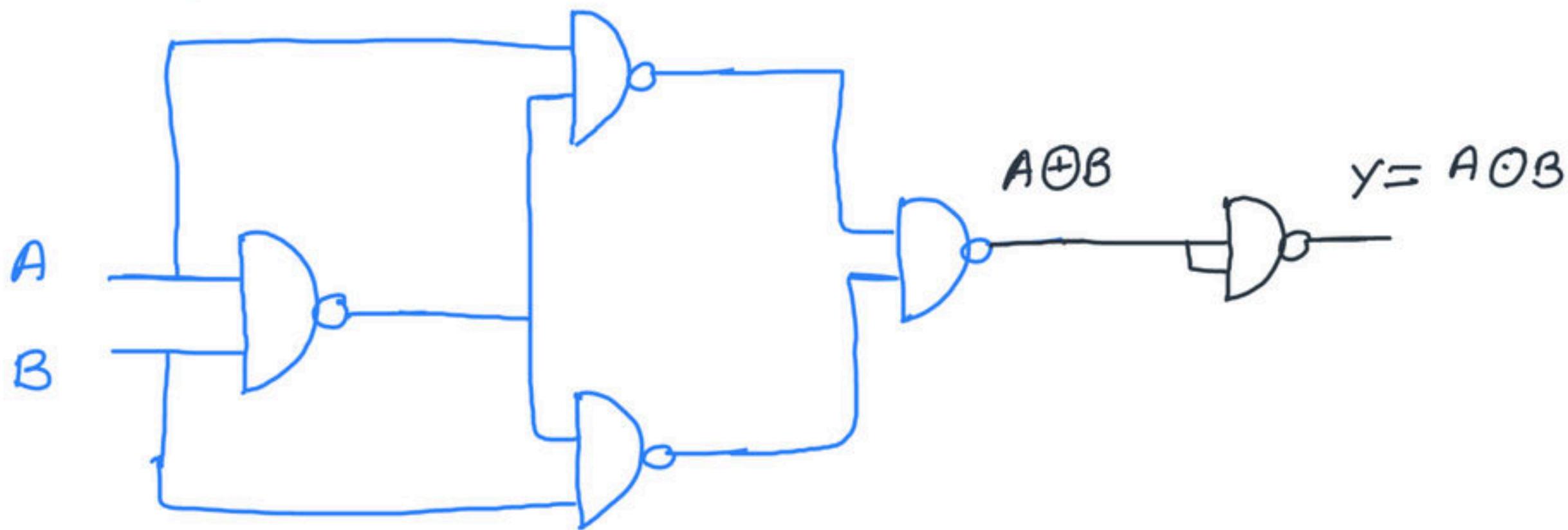
$$Y = \overline{A \oplus B}$$

$$Y = A \oplus B$$



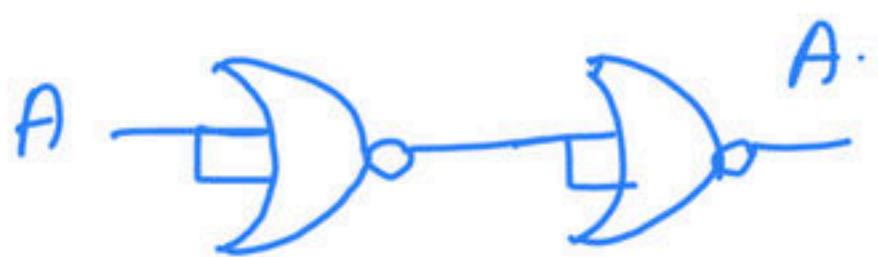
Kaju Barfi

7. EX-NOR GATE



NOR Gate as Universal Gate

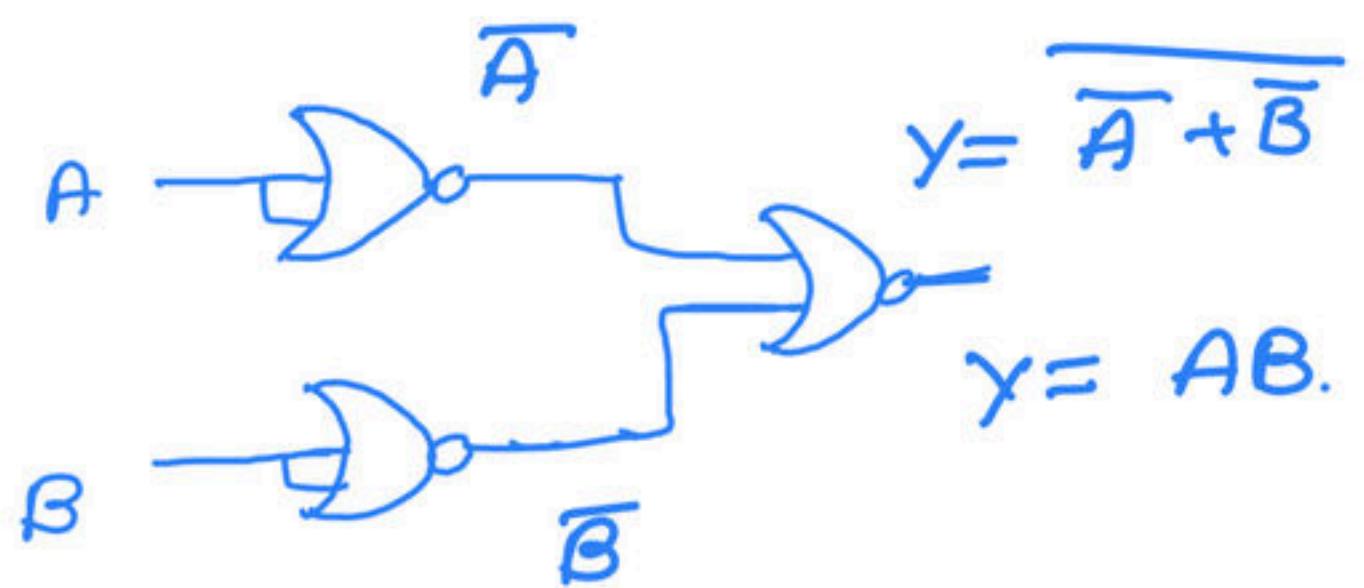
1. BUFFER GATE



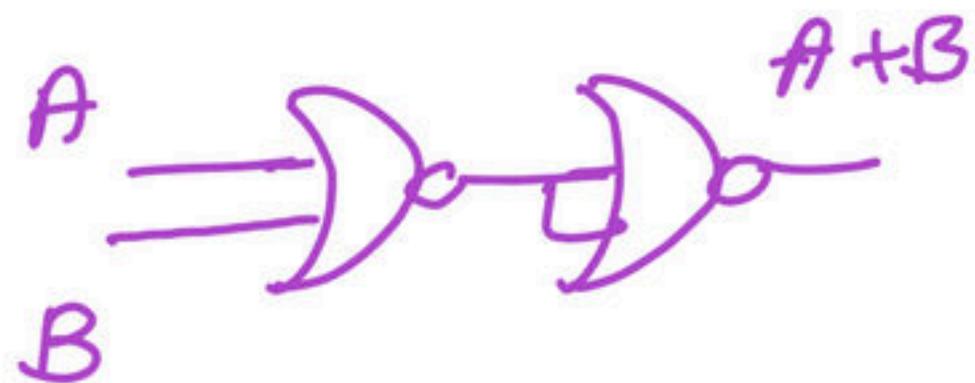
2. NOT GATE



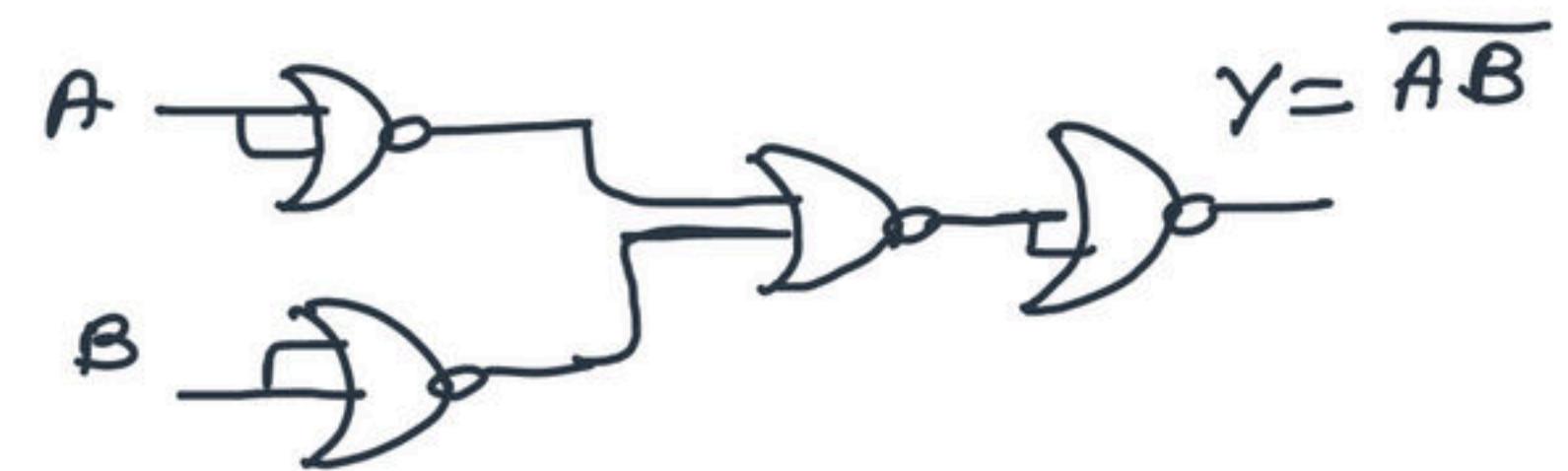
3. AND GATE



4. OR GATE

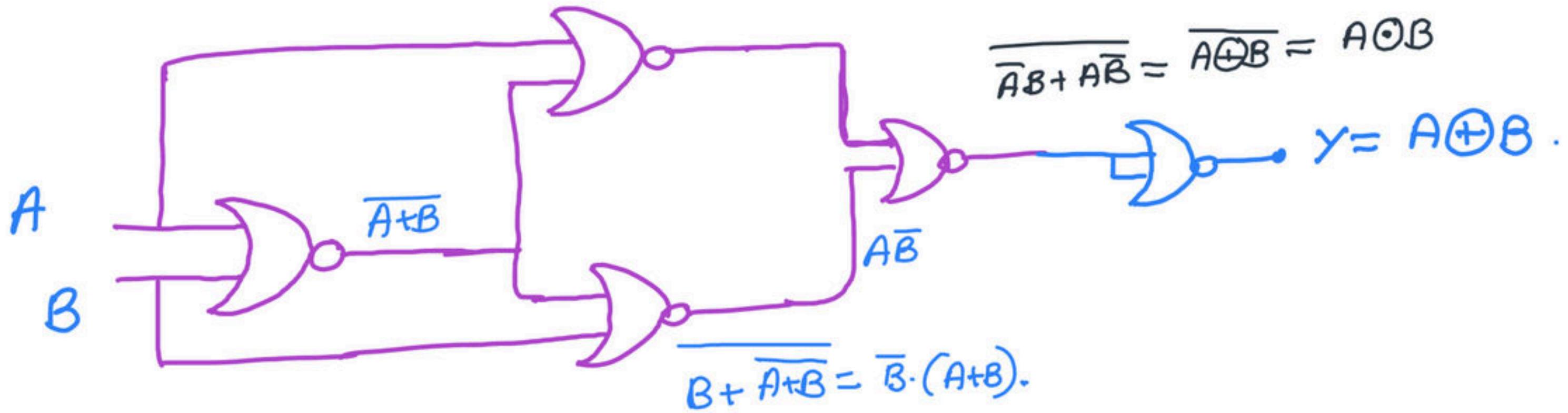


5. NAND GATE



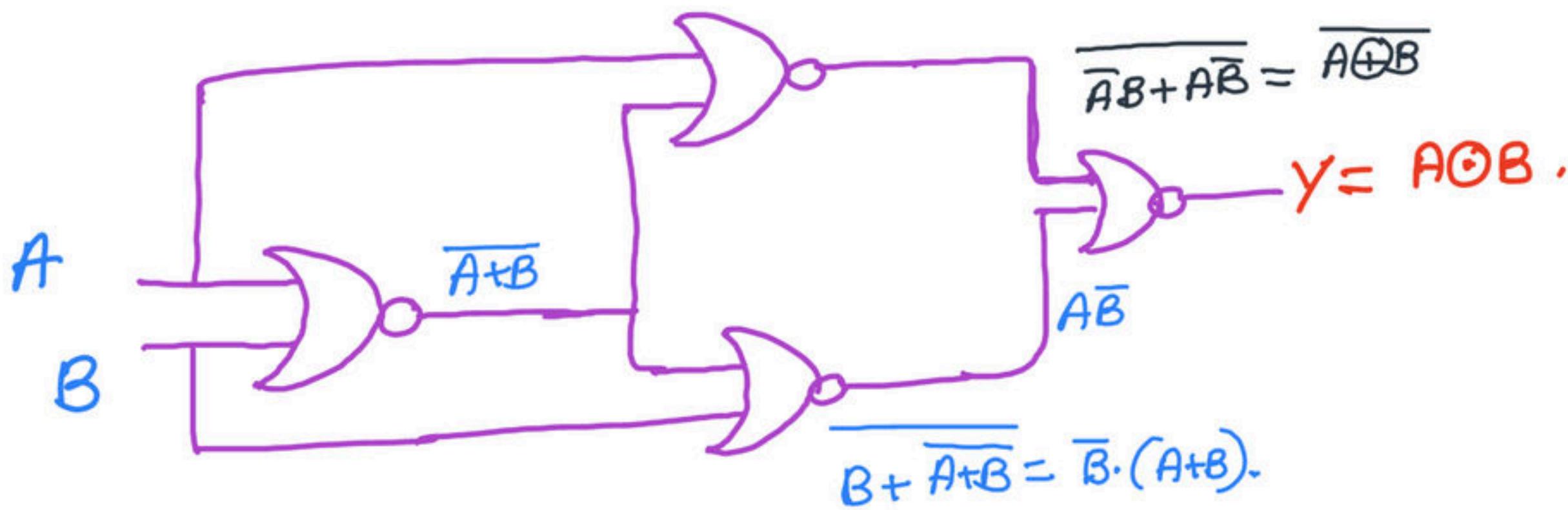
6. EX-OR GATE

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



7. EX-NOR GATE

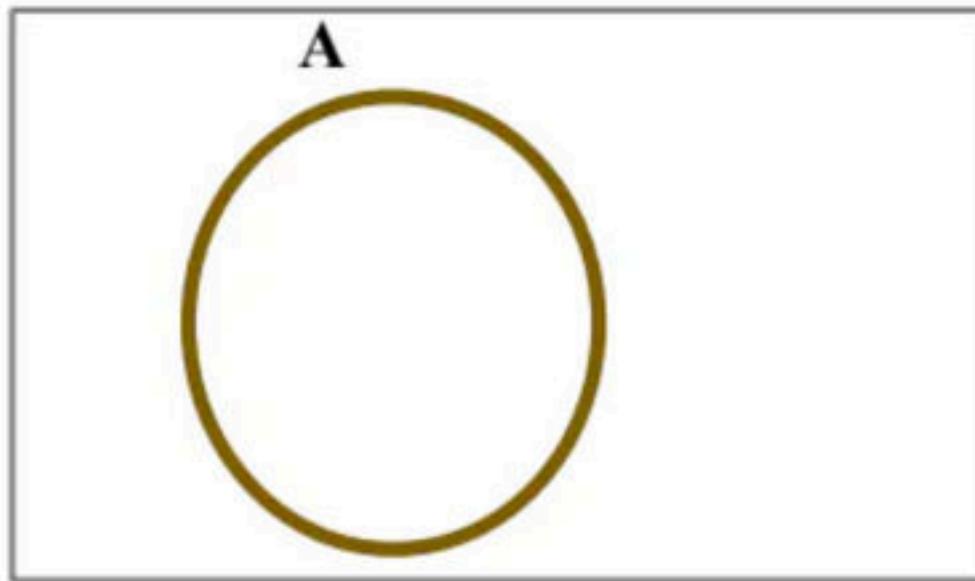
$$\overline{A + \overline{A+B}} = \overline{A} \cdot (A+B) = A \oplus B$$



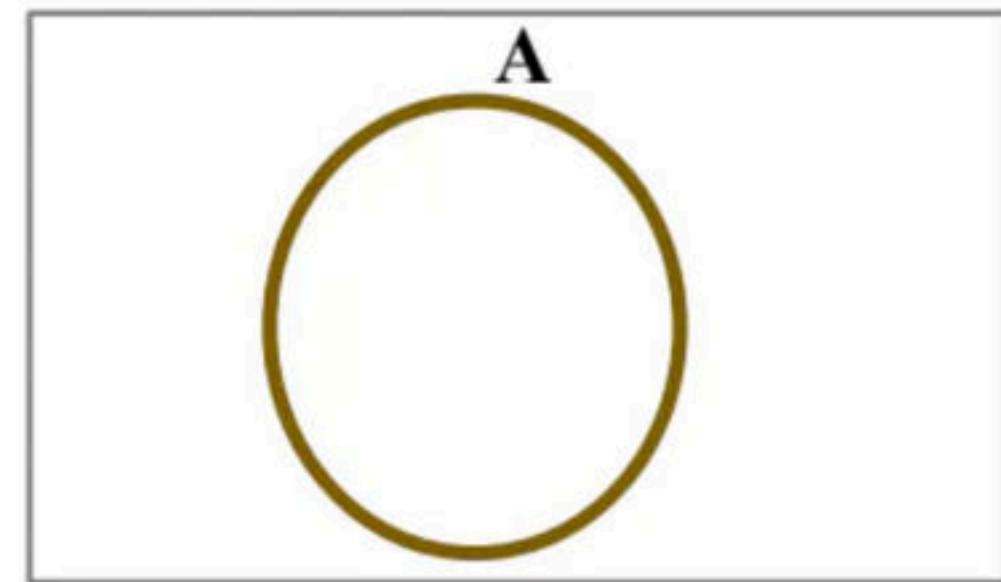
	Number of NAND GATES	Number of NOR GATES
BUFFER		
NOT		
AND		
OR		
EX-OR		
EX-NOR		
NAND		
NOR		

Venn Diagrams

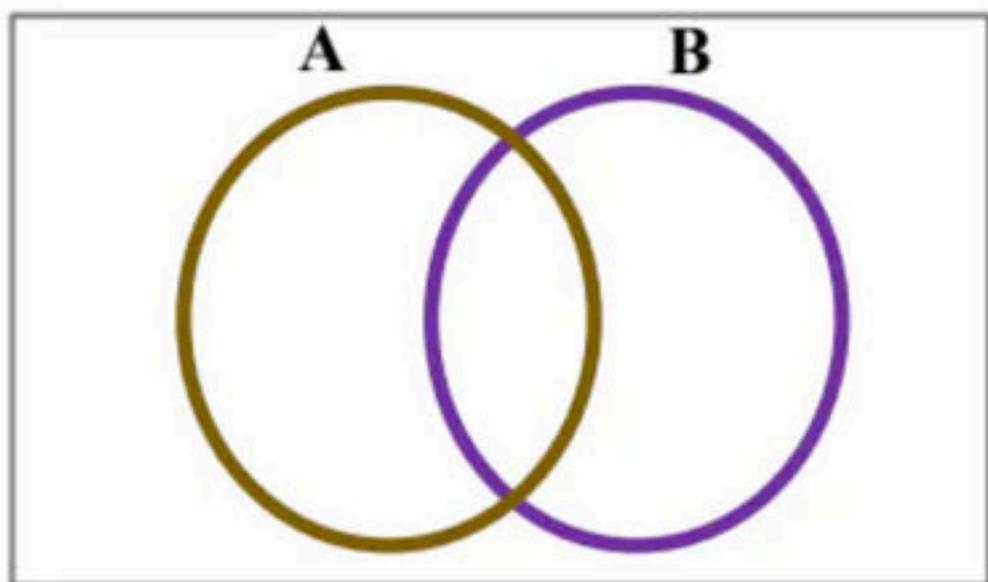
Buffer – Gate



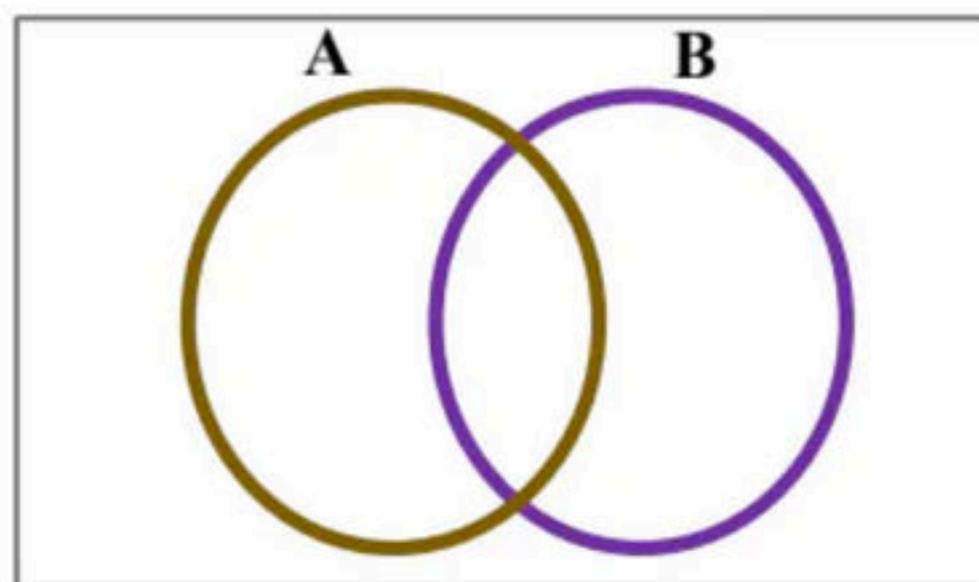
NOT – Gate



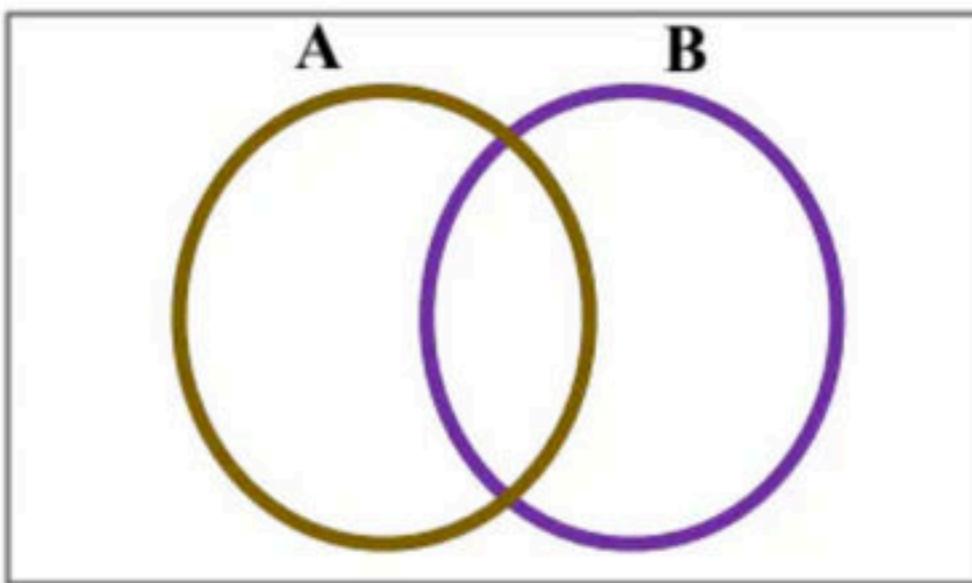
AND – Gate



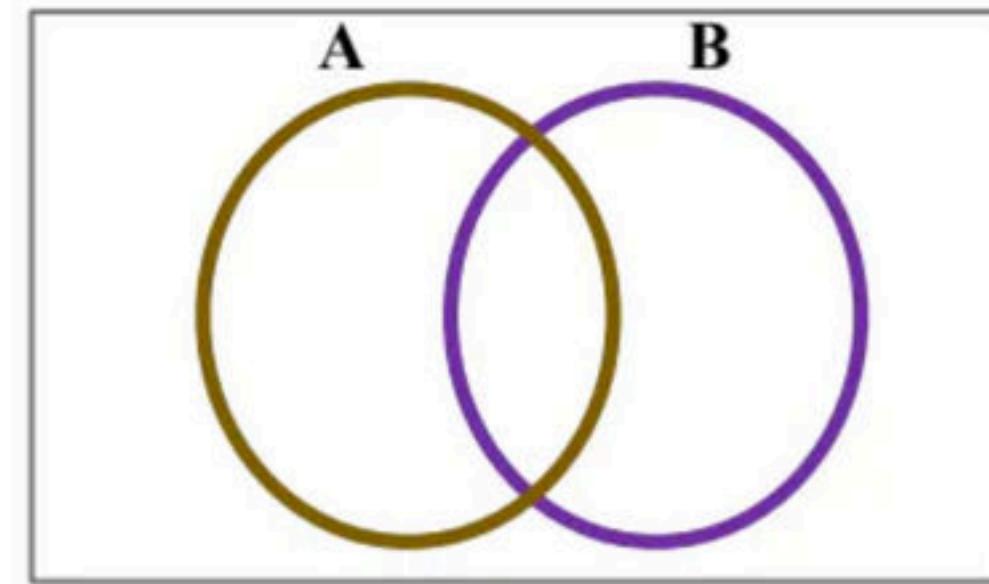
OR – Gate



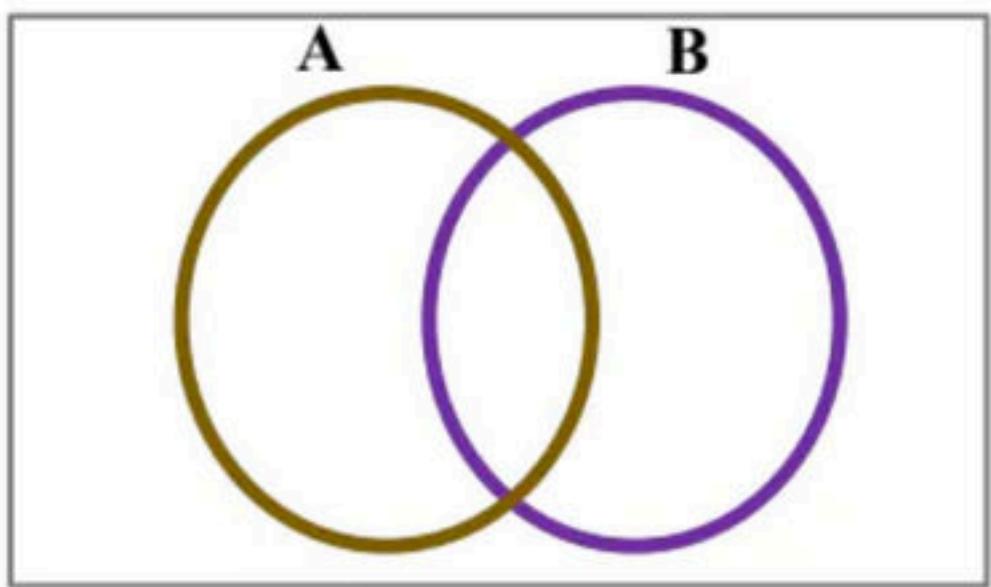
NAND – Gate



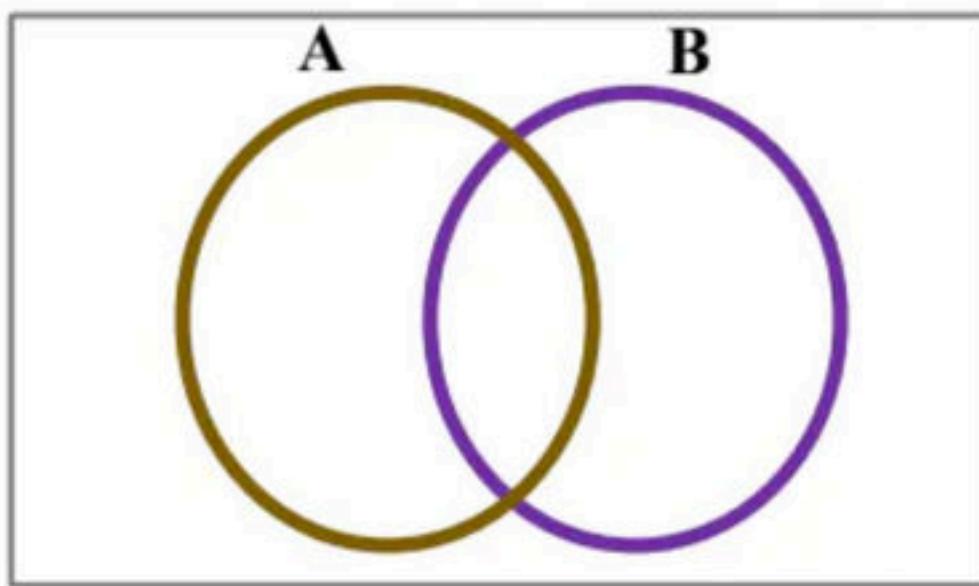
NOR – Gate



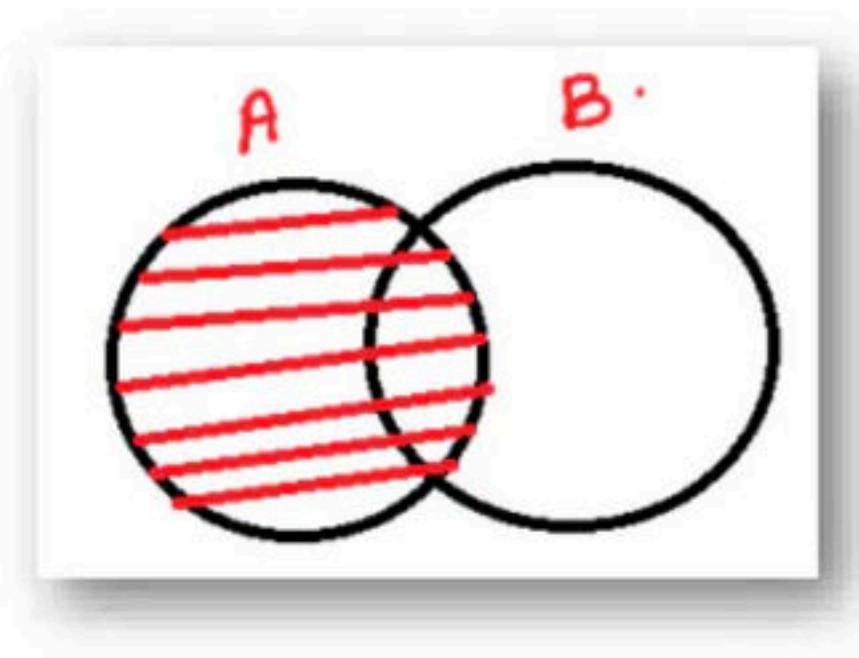
EX OR – Gate



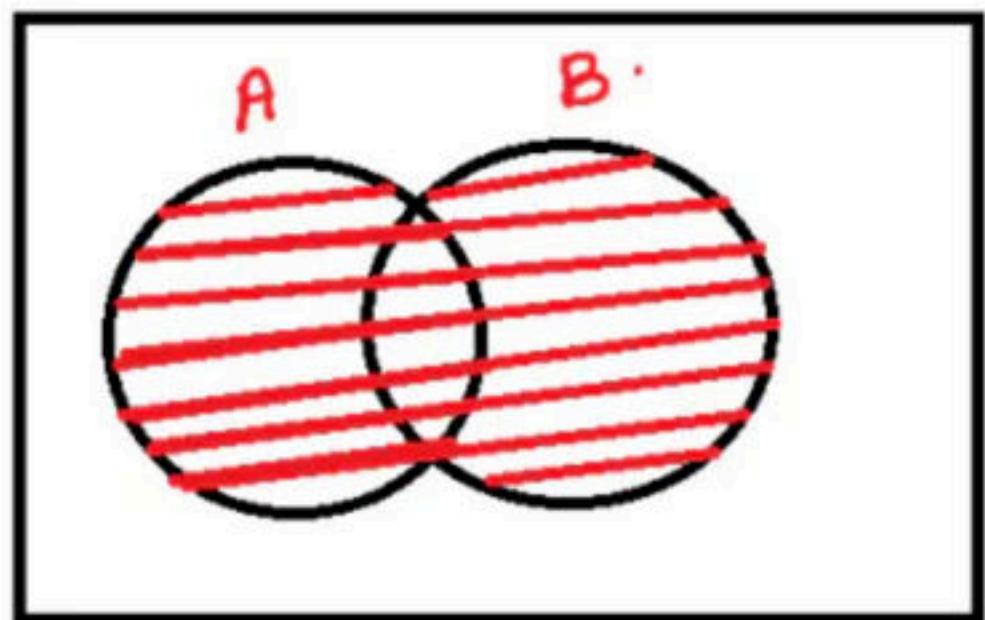
EX NOR – Gate



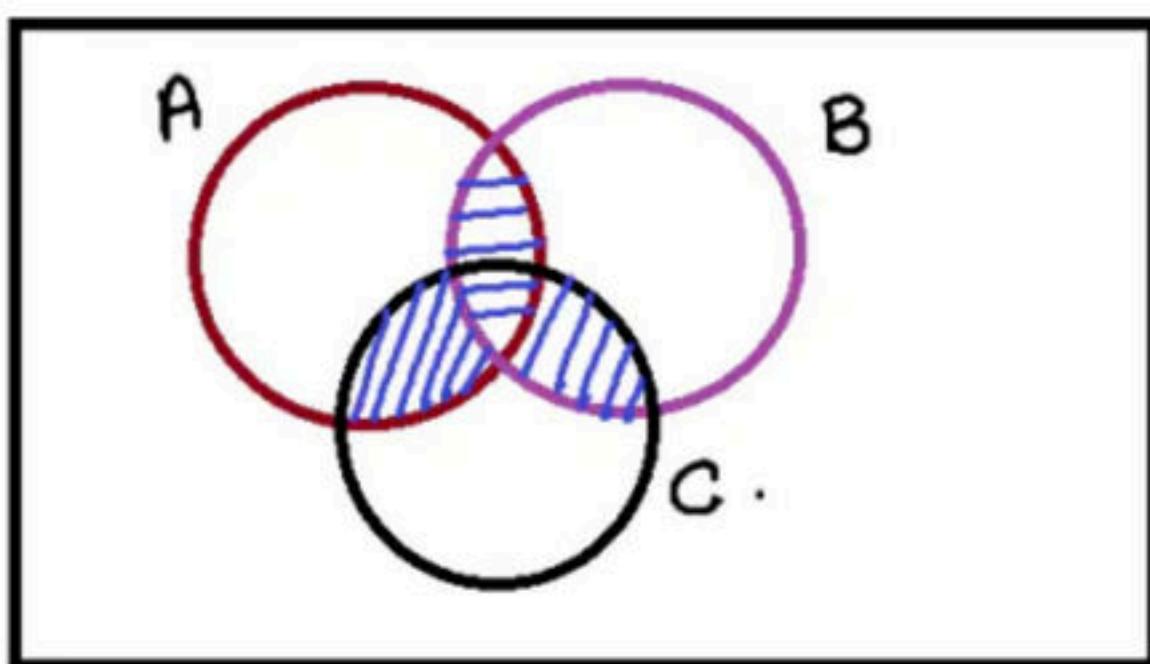
Q) For the given venn diagrams , find the minimized logical expression



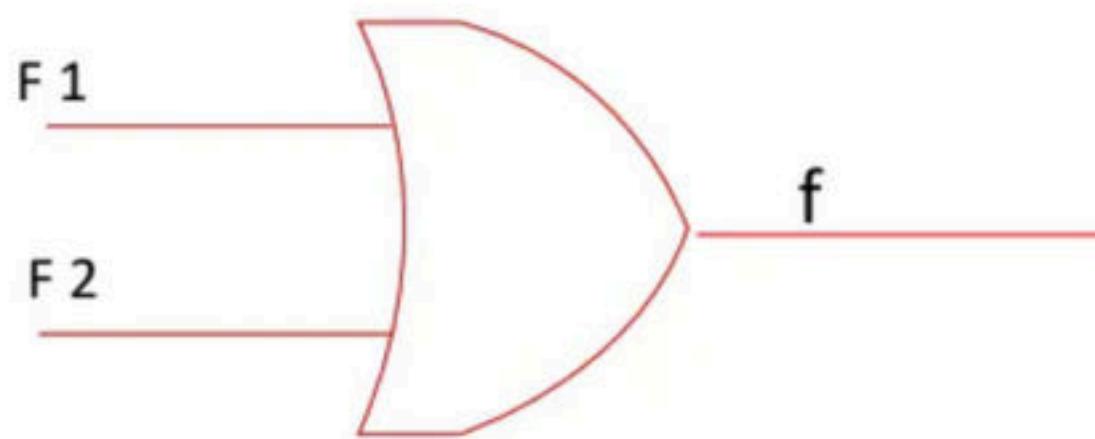
Q) For the given venn diagrams , find the minimized logical expression



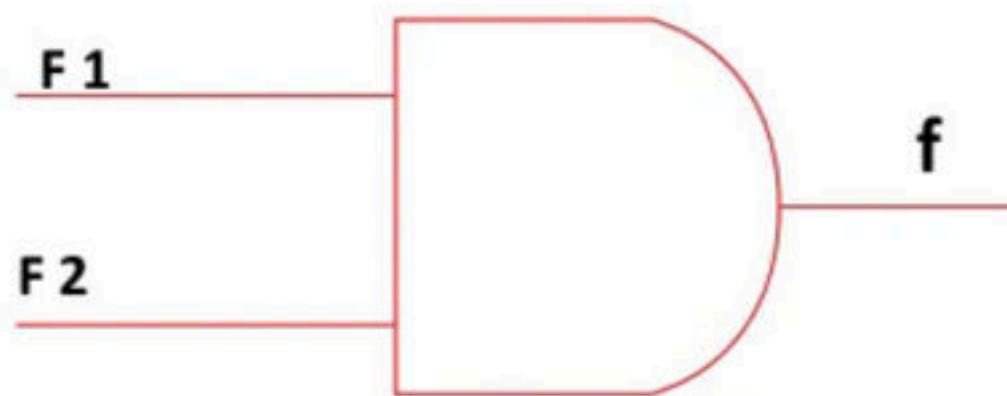
Q) For the given venn diagrams , find the minimized logical expression



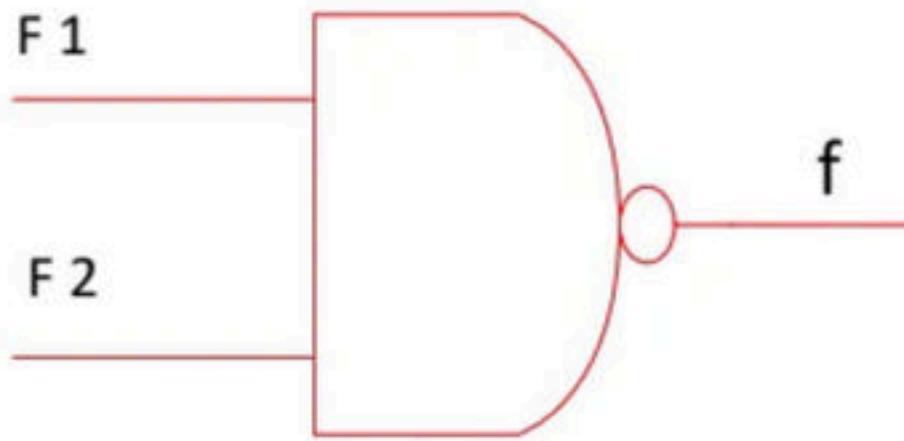
Q) If $F1 = \sum m(2,4,5,8,10)$ and $F2 = \sum m(0,1,2,8,14,15)$, then find f



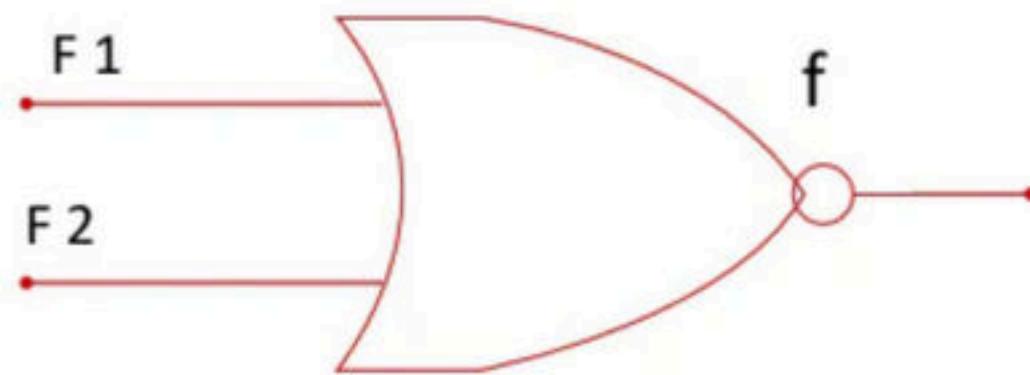
Q) If $F_1 = \sum m(2,4,5,8,10)$ and $F_2 = \sum m(0,1,2,8,14,15)$, then find f



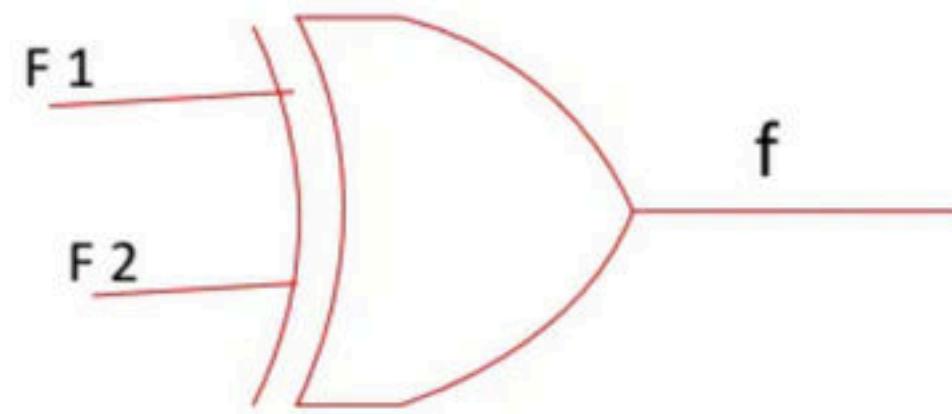
Q) If $F_1 = \sum m(2,4,5,8,10)$ and $F_2 = \sum m(0,1,2,8,14,15)$, then find f



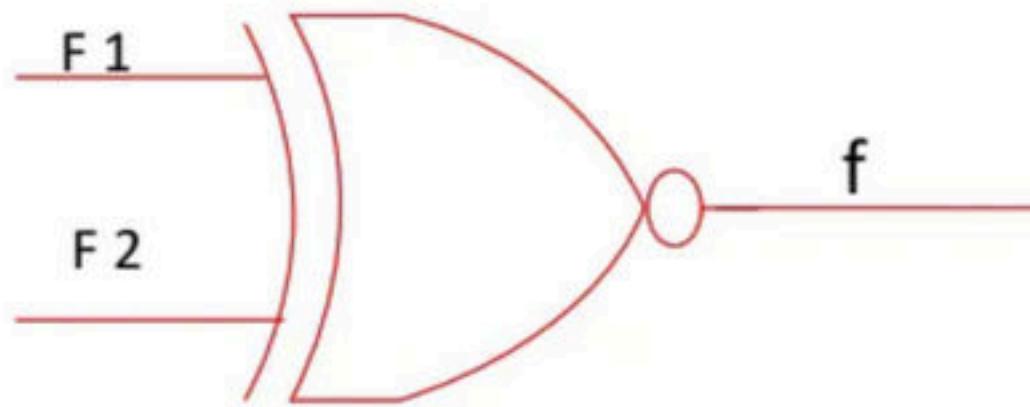
Q) If $F_1 = \sum m(2,4,5,8,10)$ and $F_2 = \sum m(0,1,2,8,14,15)$, then find f



Q) If $F_1 = \sum m(2,4,5,8,10)$ and $F_2 = \sum m(0,1,2,8,14,15)$, then find f

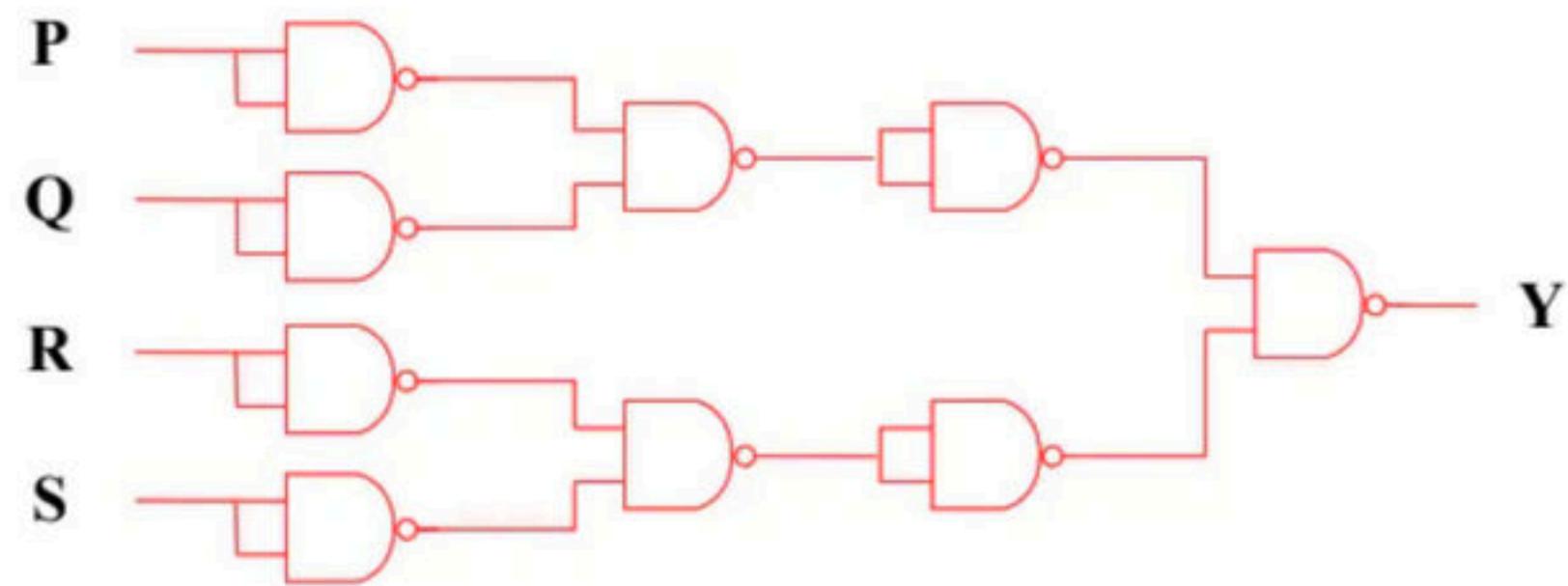


Q) If $F1 = \sum m(2,4,5,8,10)$ and $F2 = \sum m(0,1,2,8,14,15)$, then find f



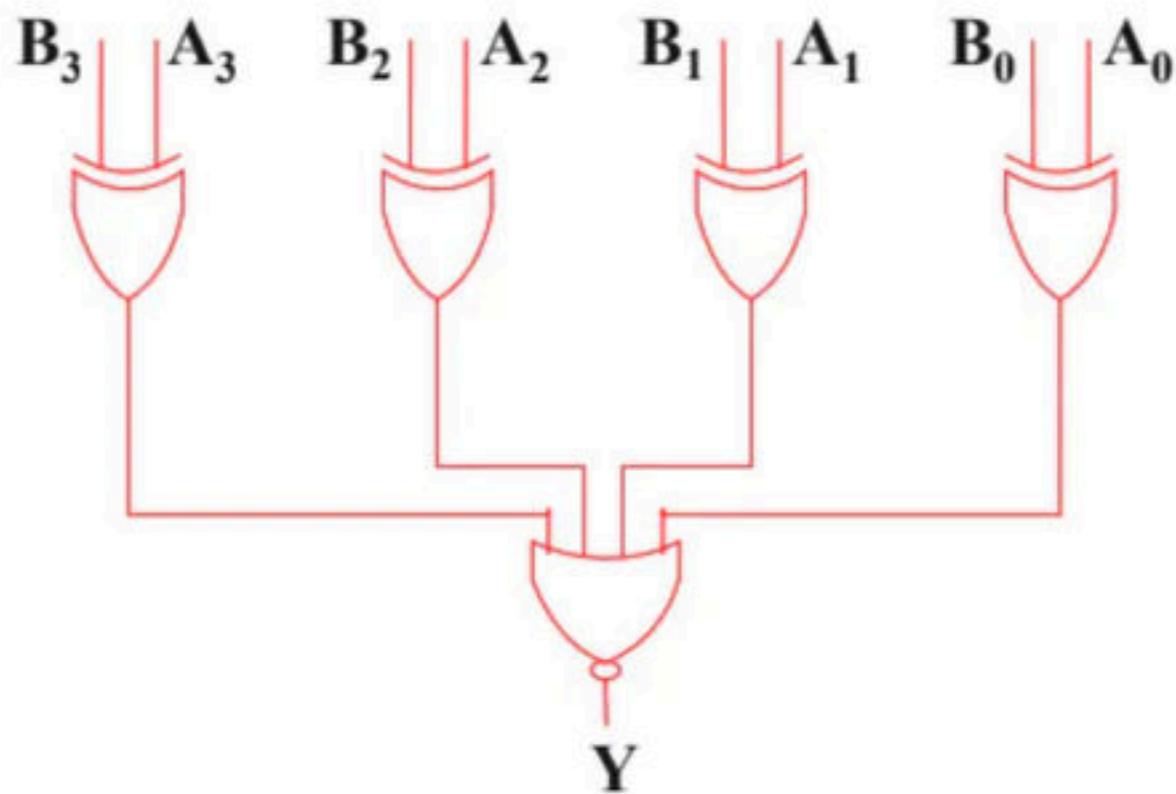
Q. For the circuit shown in figure, the Boolean expression for the output Y in terms of inputs P, Q, R and S is

- (a) $\bar{P} + \bar{Q} + \bar{R} + \bar{S}$
- (b) $P + Q + R + S$
- (c) $(\bar{P} + \bar{Q})(\bar{R} + \bar{S})$
- (d) $(P + Q)(R + S)$

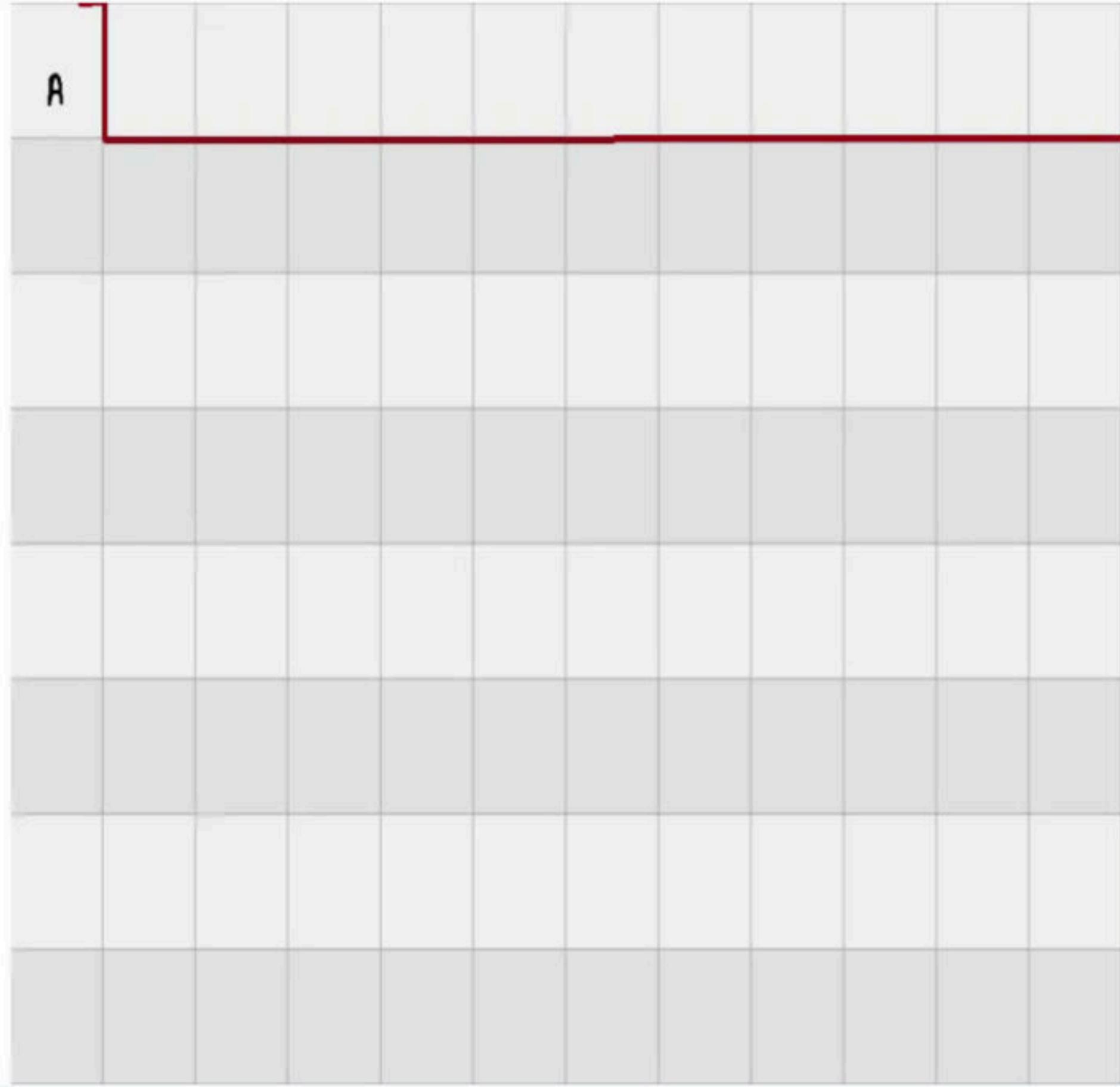
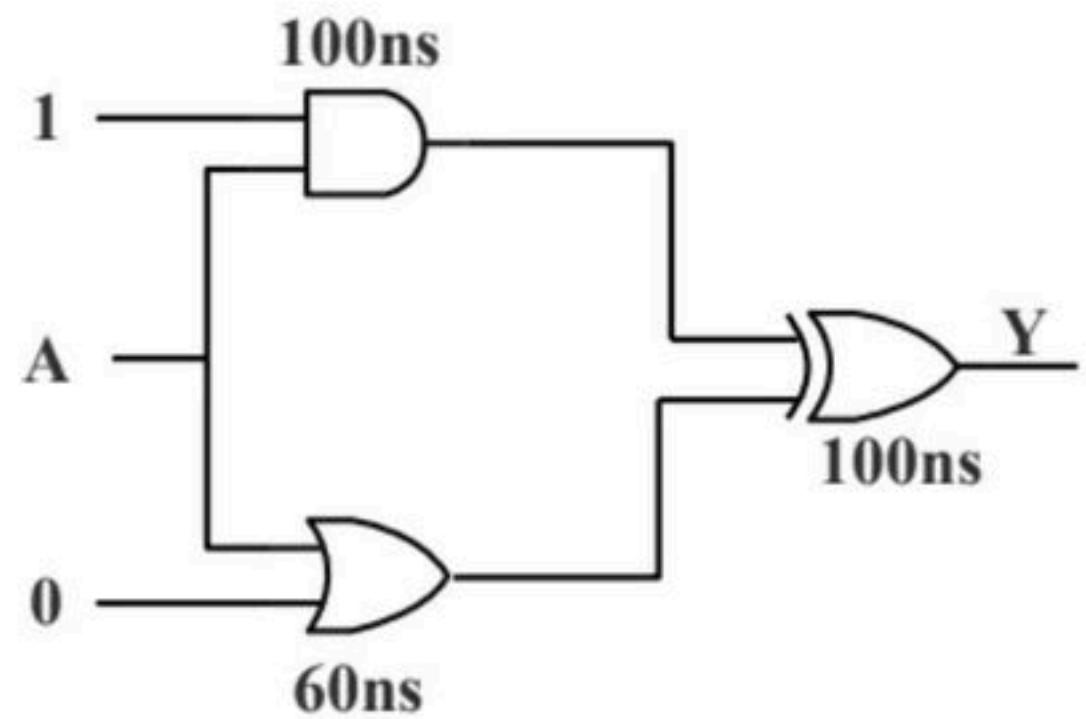


Q. A digital circuit, which compares two numbers, A_3, A_2, A_1, A_0 , B_3, B_2, B_1, B_0 is shown in figure. To get output $Y = 0$, choose one pair of correct input numbers.

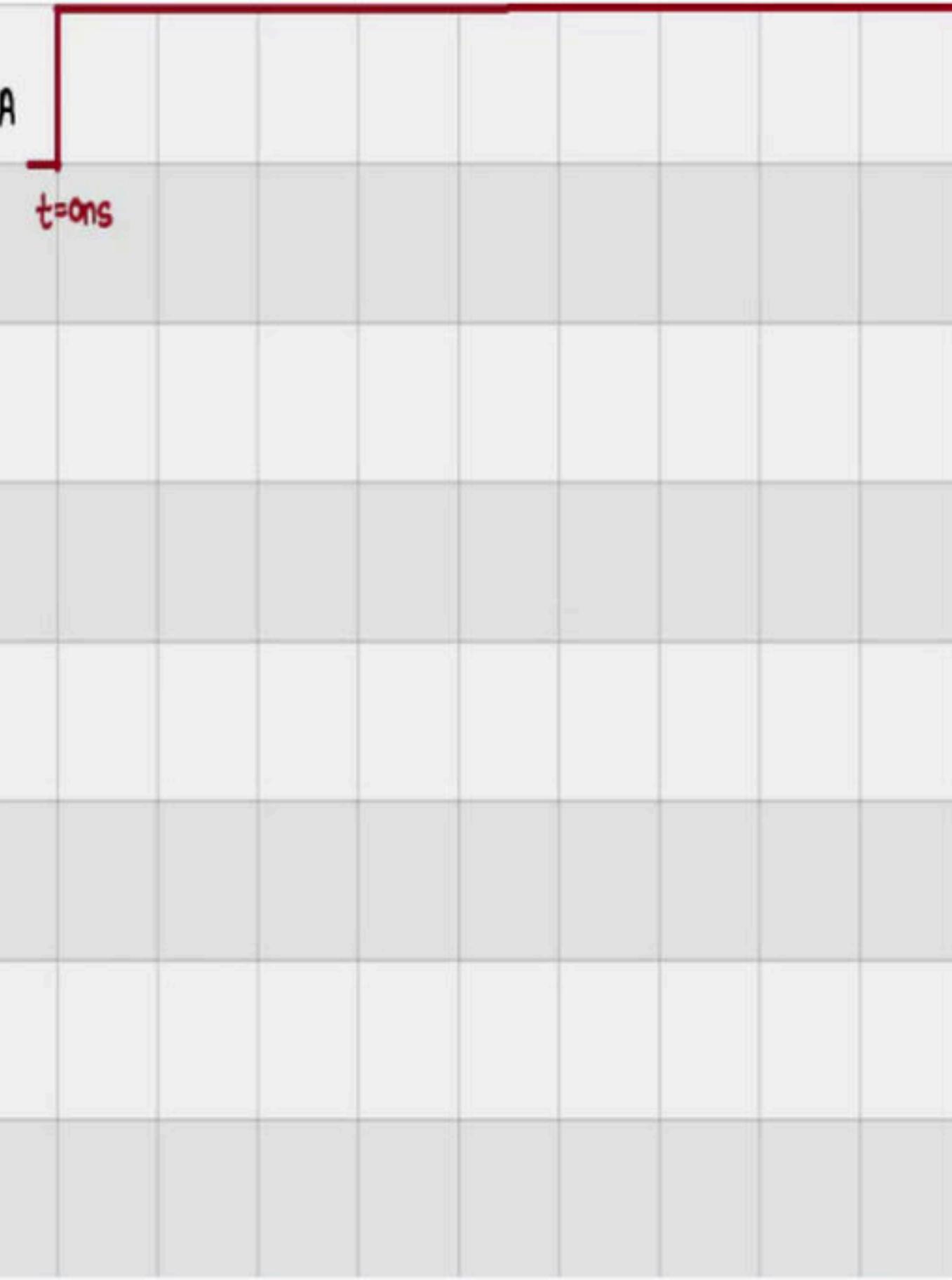
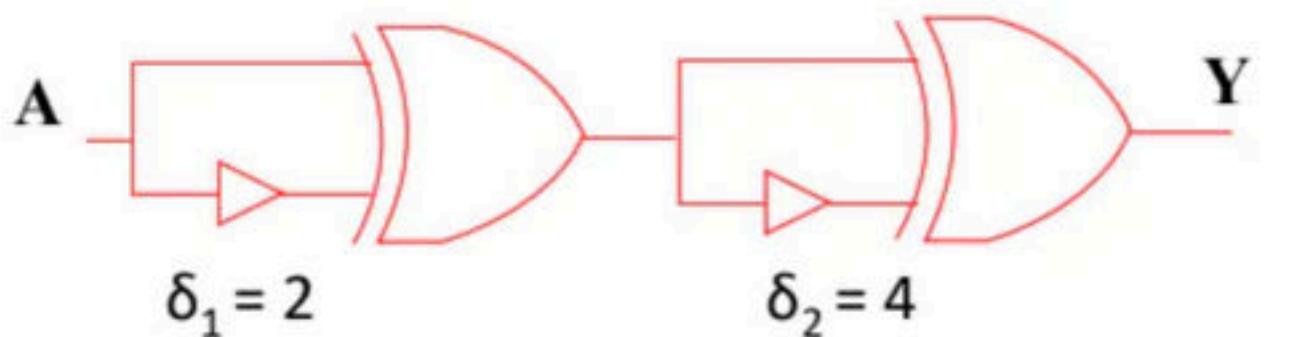
- (a) 1010, 1010
- (b) 0101, 0101
- (c) 0010, 0010
- (d) 0010, 1011



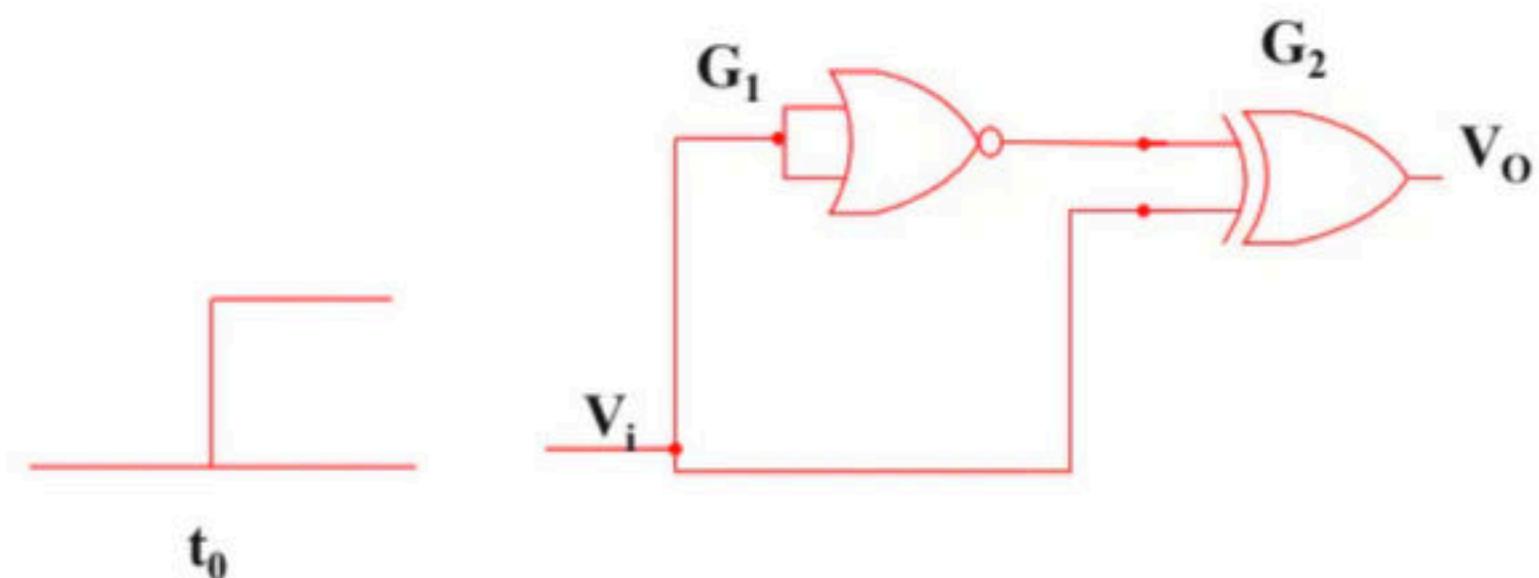
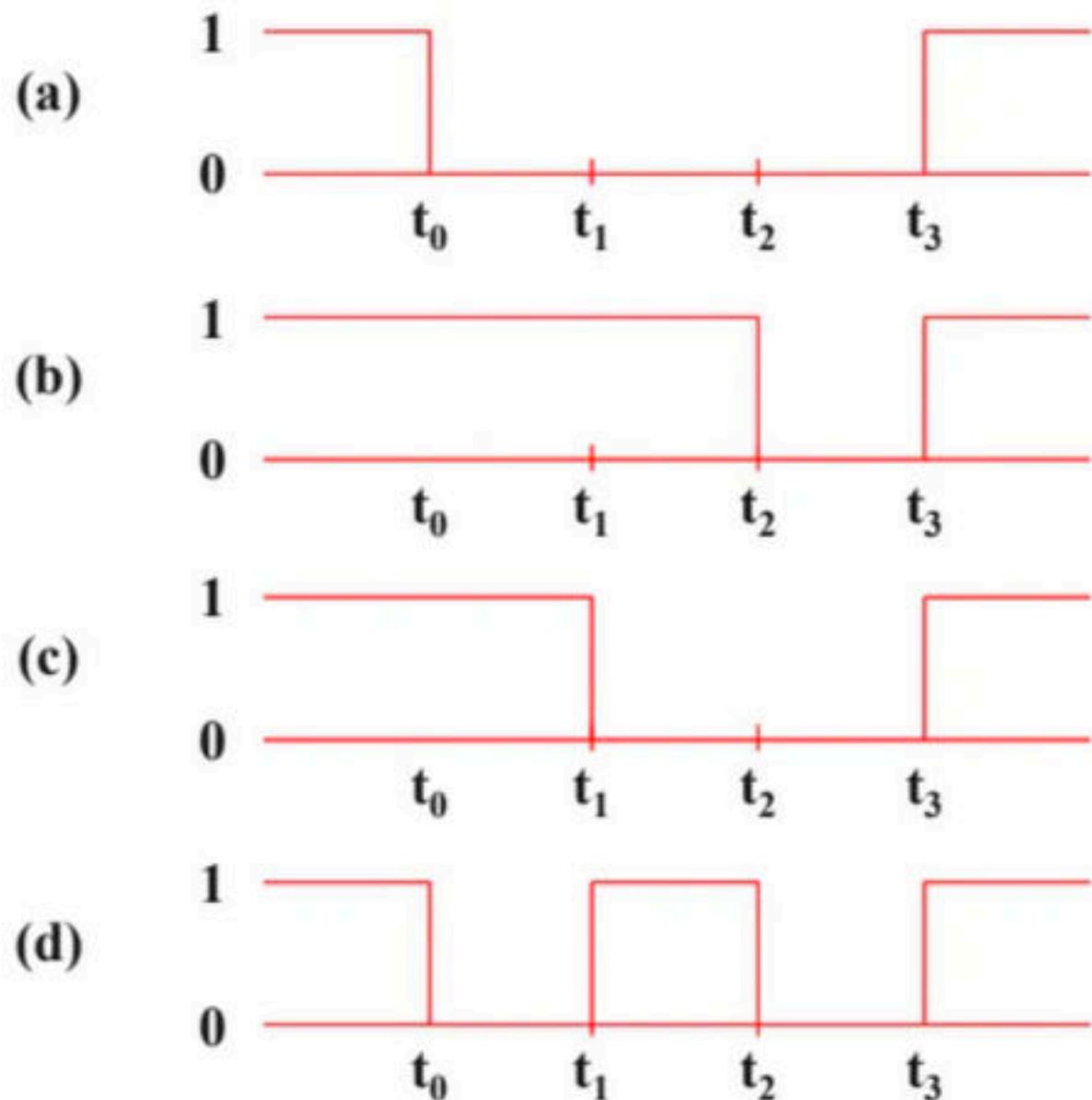
Q) Draw the output wave (Y)



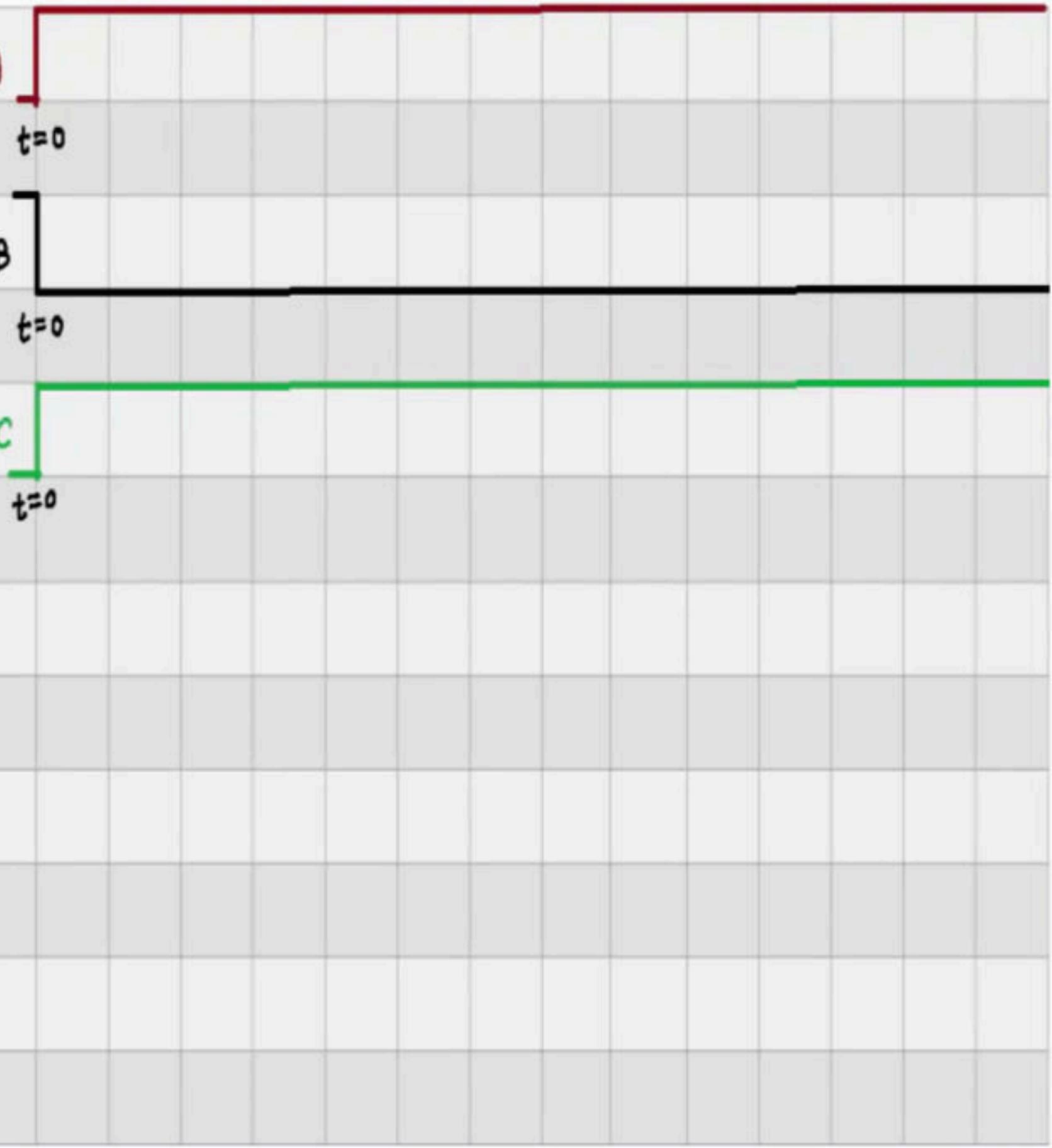
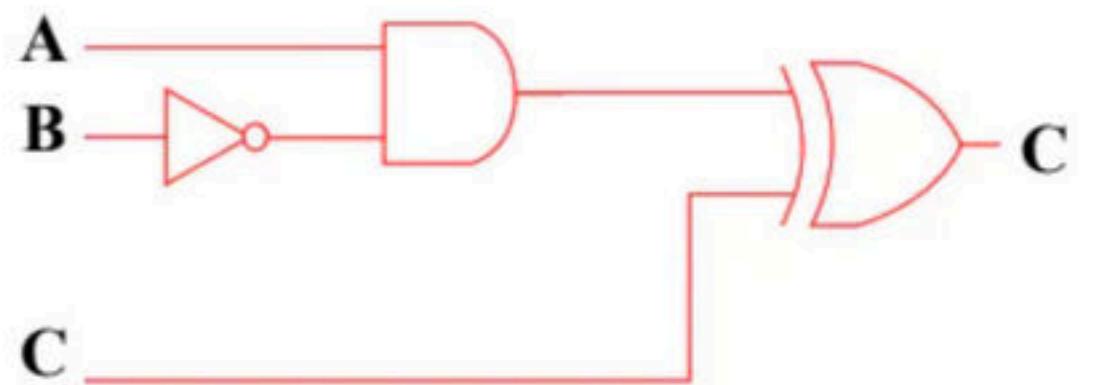
Q) How many transition's occurs in the output Y from 0 to 10 ns



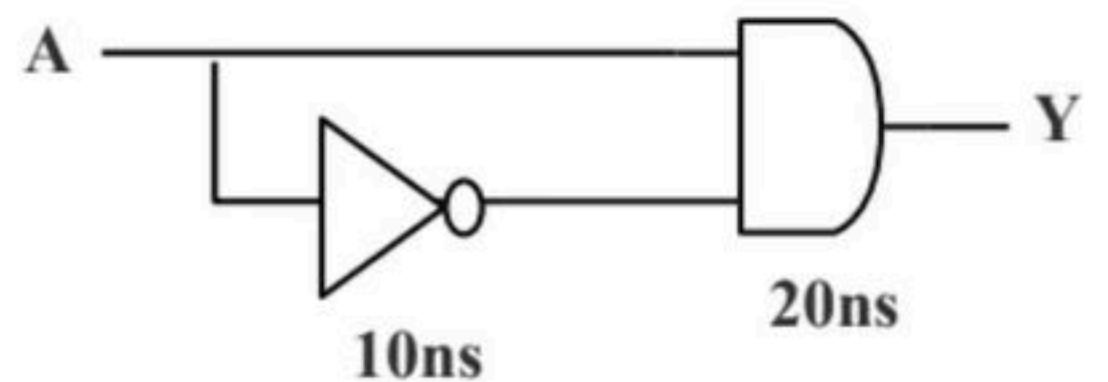
Q. The gates G_1 and G_2 in figure have propagation delays of 10nsec and 20nsec respectively. If the input V_i makes an abrupt change from logic 0 to 1 at time $t = t_0$ then the output waveform V_o is



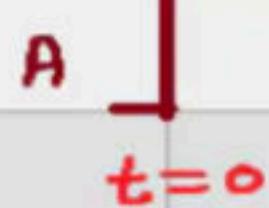
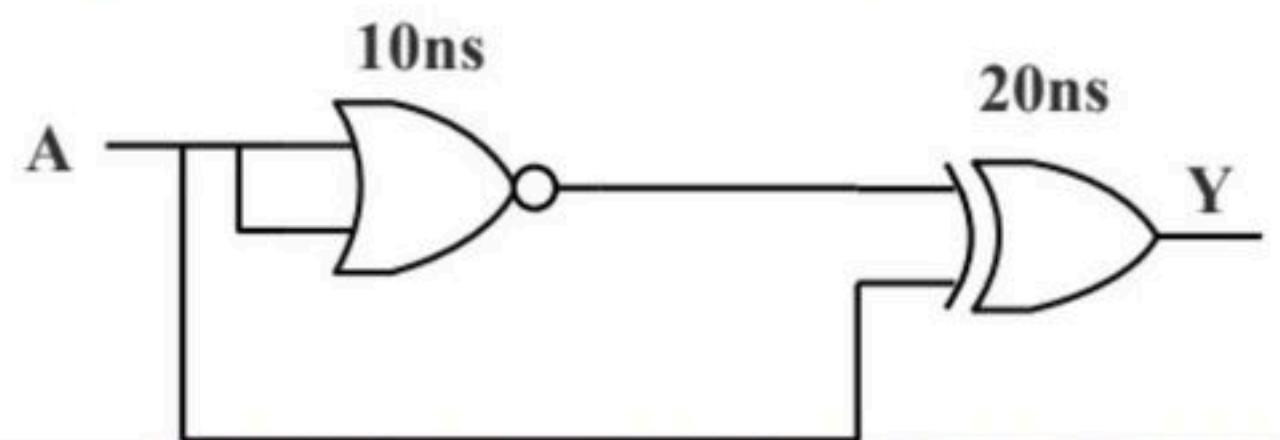
Q. All the logic gates shown in the figure have a propagation delay of 20 ns. Let $A = C = 0$ and $B=1$ until time $t = 0$. At $t= 0$, all the inputs flip (i.e., $A = C = 1$ and $B = 0$) and remain in that state. For $t > 0$, output $Z= 1$ for a duration (in ns) of



Q) Draw the output wave (Y)



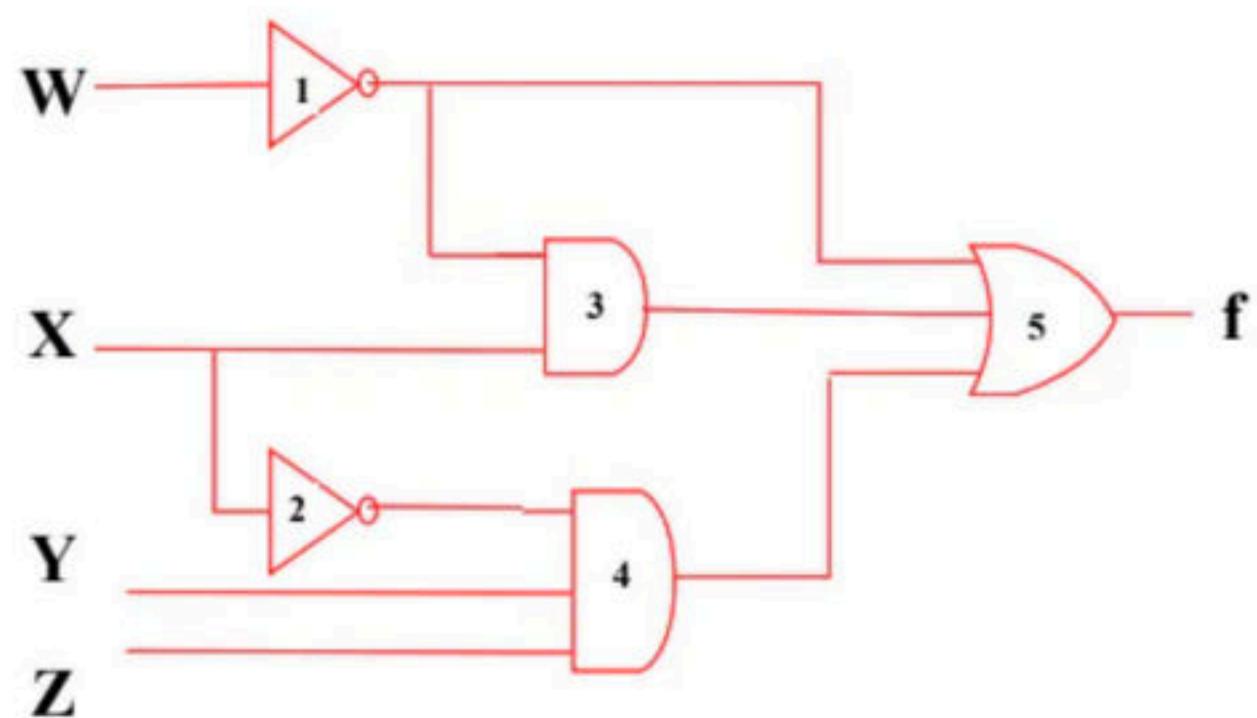
Q) Draw the output wave (Y)



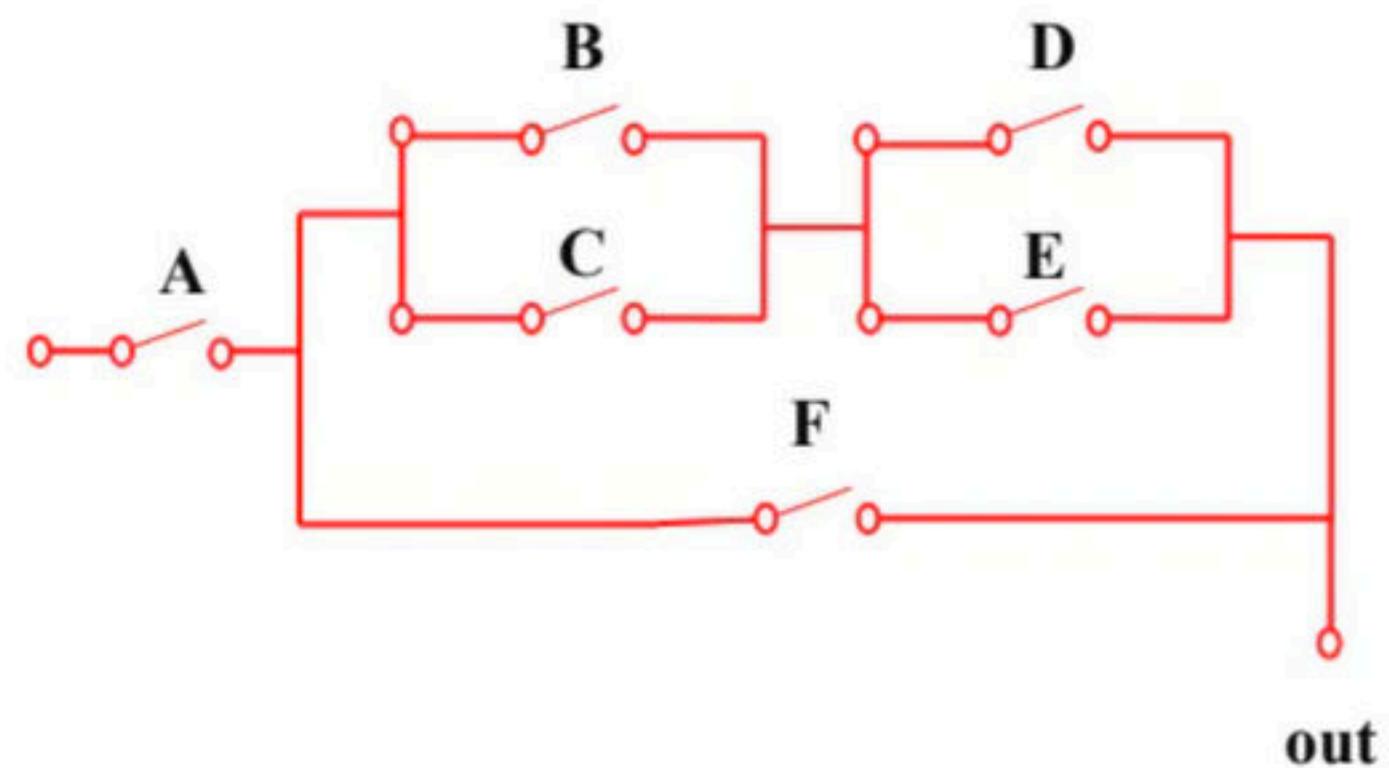
Q. Consider the following gate network:

Which one of the following gates is redundant?

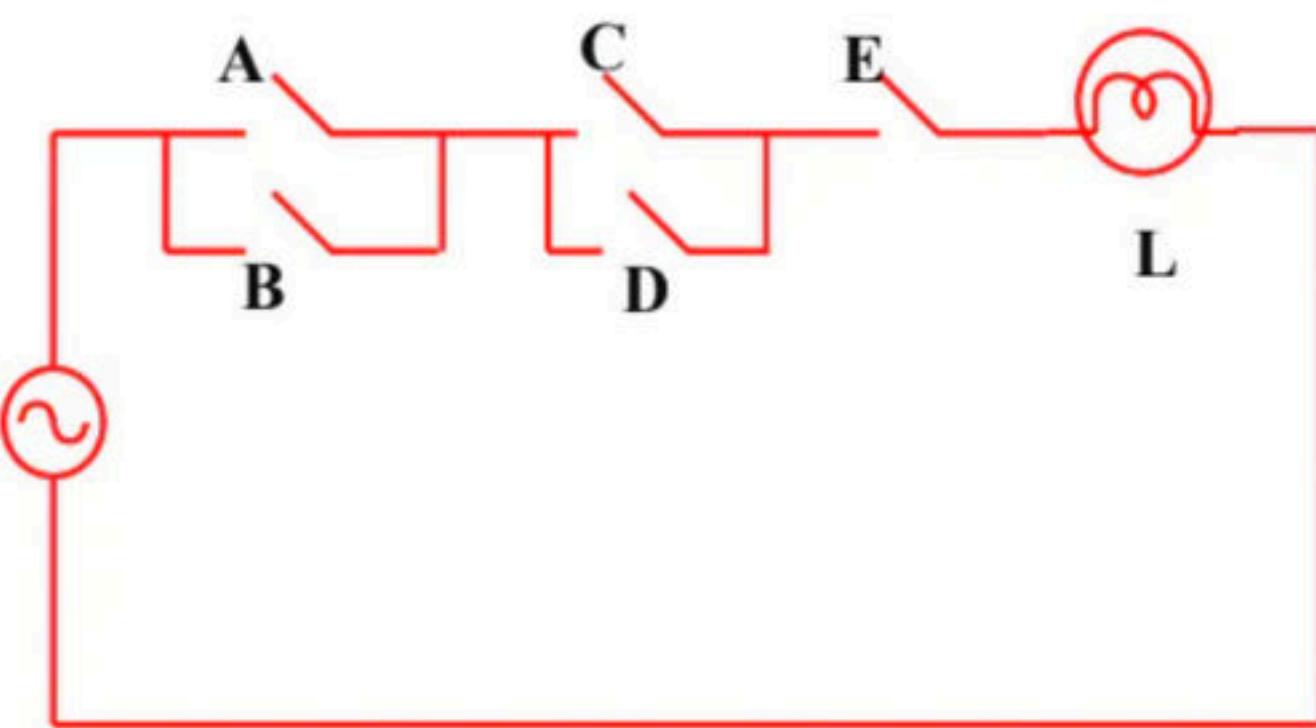
- (a) Gate No. 1
- (b) Gate No. 2
- (c) Gate No. 3
- (d) Gate No. 4



Q. What Boolean function does the following circuit represents:



Q. The switching circuit given in the figure can be expressed in binary logic notation as



Q) A 3 – input majority gate is defined by the logic function $M(a, b, c) = ab + bc + ca$, which one of the following gates is represented by the function $M(\overline{M(a, b, c)}, M(a, b, \bar{c}), c)$

- a) 3- input NAND gate
- b) 3- input EX-OR gate
- c) 3- input NOR gate
- d) 3- input XNOR gate

Q) The following expression was to be realized using 2 input AND , OR, NOT gates , however during fabrication all 2 input AND gates are mistakenly substituted by 2 input NAND gates

$(ab)c + (\bar{a}c)d + (bc)d + ad$ what is the function realised finally

- a) 1
- b) $\bar{a} + \bar{b} + \bar{c} + \bar{d}$
- c) $\bar{a} + b + \bar{c} + \bar{d}$
- d) $\bar{a} + \bar{b} + c + \bar{d}$

Functionally Complete

- By using the given set of Logic expression , if it is possible to implement all the Boolean functions , then the given set of logic expressions are called as functionally complete.
- All the Boolean functions are implemented by using the basic gates {AND , OR and NOT } so the set {AND , OR and NOT} is called as Functionally Complete set .

- NAND is always functionally complete since any given Boolean function can be implemented .
- NOR is always functionally complete since any given Boolean function can be implemented .
- The set {NOT, AND } , is a functionally complete
- The set {NOT, OR } is a functionally complete
- Functionally complete logic set is also called as universal logic gate

- For a given function to verify whether it is functionally complete or not then substitute A , 0, 1 in place of various Boolean variable's .

Q) Verify whether the function is functionally complete or not

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

Q) Verify whether the function is functionally complete or not

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

Q) Verify whether the function is functionally complete or not

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

Q) Verify whether the function is functionally complete or not

$$f(A, B) = A \oplus B$$

Q) Verify whether the function is functionally complete or not

$$f(A, B) = A \odot B$$

Q) Verify whether the function is functionally complete or not

$$f(x, y, z) = xyz + xy + \bar{y}\bar{z}$$

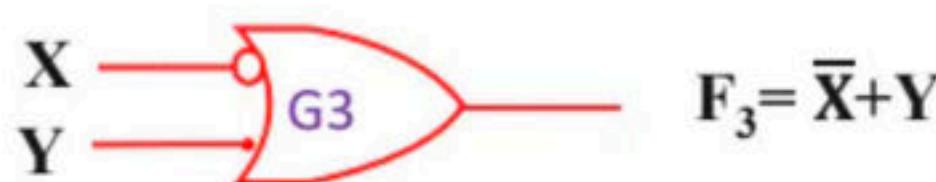
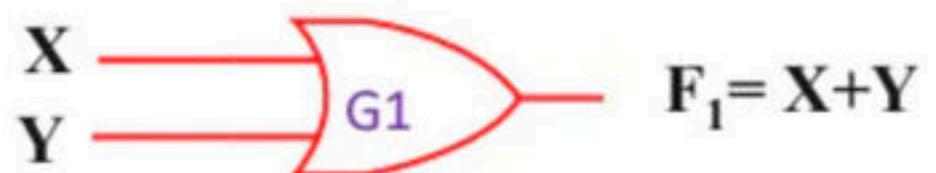
Q) Verify whether the function is functionally complete or not

$$f(x, y, z) = \bar{x}yz + \bar{x}y\bar{z} + xy$$

Q. A universal logic gate can implement any Boolean function by connecting sufficient number of them appropriately. Three gates are shown.

Which one of the following statements is TRUE?

- (a) Gate 1 is a universal gate
- (b) Gate 2 is a universal gate
- (c) Gate 3 is a universal gate
- (d) None of the gates shown is a universal gate



Inhibitor Logic

- If one of the input of AND gate (or) OR gate is inverted then it is called as Inhibitor logic

Represent the Boolean expression $F(A, B, C) = \Pi(0, 2, 4, 5)$ in standard POS Form.

Convert the following Boolean function into standard SOP and express it in terms of minterms.

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

Convert the following Boolean function into standard POS and express it in terms of maxterms.

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

Convert the following SOP expression to an equivalent POS expression.

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

For the Boolean function F given in the truth table, find the following:

- List the minterms of the function.
- List the minterms of F' .
- Express F in sum of minterms in algebraic form.
- Simplify the function to an expression with a minimum number of literals.

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

Express the following functions in sum of minterms and product of maxterms:

(a) $F(A, B, C, D) = B'D + A'D + BD$

(b) $F(x, y, z) = (xy + z)(xz + y)$

Express the complement of the following functions in sum of minterms:

(a) $F(A, B, C, D) = \Sigma(0, 2, 6, 11, 13, 14)$

(b) $F(x, y, z) = \Pi(0, 3, 6, 7)$

Convert the following to the other canonical form:

(a) $F(x, y, z) = \Sigma(1, 3, 7)$

(b) $F(A, B, C, D) = \Pi(0, 1, 2, 3, 4, 6, 12)$