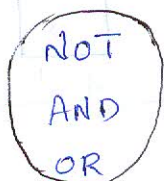


Logic Gates

→ Basic Building blocks of any logic ckt.

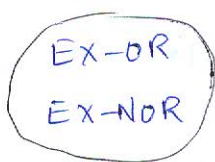
→



→ Basic gates



→ universal gates



→ Arithmetic cks
comparators
code converters

⇒ Not gate (Inverter) :-



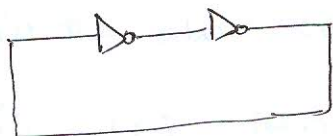
0 → 1

1 → 0

→

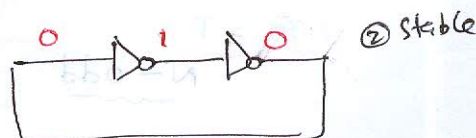
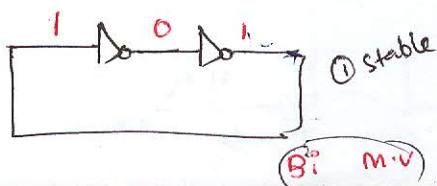


Q

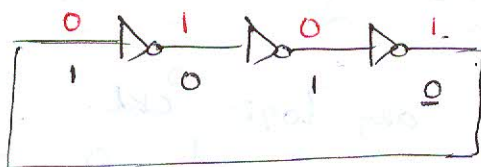


the ckt shown in figure acts as

- ① Buffer ② Astable MV ③ Basic memory element (or) Bistable MV.
④ oscillator



Q3



Astable multi vibrator.

(or)

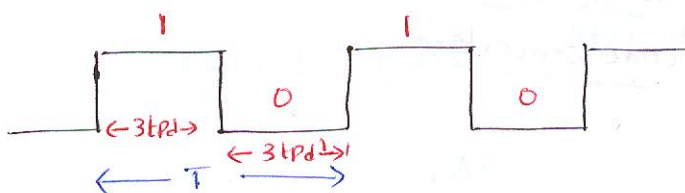
Square wave generator.

(or)

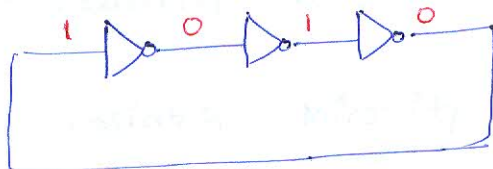
clock generator

(or)

Ring oscillator



$$T = 6 t_{pd} = 2 \times 3 \times t_{pd}$$



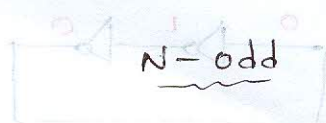
→ for getting 1, we require 6 t_{pd} ✓ Time delay

→ In ring oscillator the Time period of generator Square wave is given by

$$T = 2N t_{pd}$$

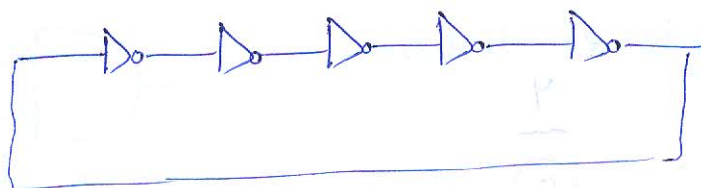
→ N - No. of inverters in feedback

→ t_{pd} - Propagation delay of Each inverter



Q3 The Ring oscillator shown in fig Propagation delay of Each inverter is 100 psec. then frequency of generated square wave is.

Sol:-



$$N=5$$

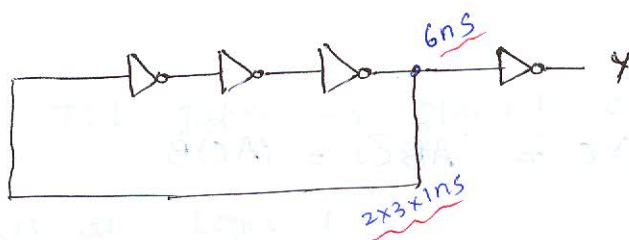
$$T = 2N \times t_{pd}$$

$$= 2 \times 5 \times 100 \text{ ps}$$

$$T = 1000 \text{ ps}$$

$$f = \frac{1}{T} = \frac{1}{1000 \text{ ps}} = \frac{10^{12}}{1000} = 10^9 = 1 \text{ GHz}$$

Q4

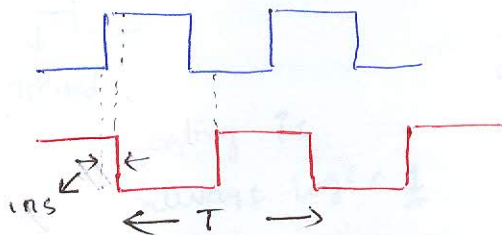


$Y = \text{Astable M.V}$

in the above ckt shown in fig the P.d of each inverter is 1ns then time period of generator is

- ✓ a) 6ns b) 7ns c) 8ns d) 4ns

Sol:-

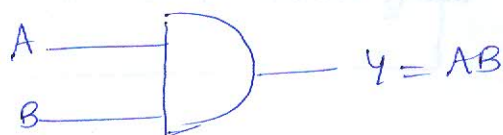


$$T = 2 \times 3 \times 1 \text{ ns}$$

$$T = 6 \text{ ns}$$

⇒ And gate :-

→



→

<u>A</u>	<u>B</u>	<u>Y</u>
0	0	0
0	1	0
1	0	0
1	1	1 ✓

1) Commutative Law :-

$$AB = BA$$

→ And gate is Commutative Law

2) Associate Law :-

$$ABC = (AB)C = A(BC) = (AC)B$$

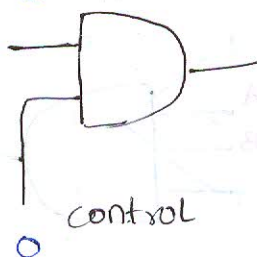
→ And gate follows both Commutative and

Associative Laws

→ For AND gate 1 is acting as Enable
0 is acting as disable.

→

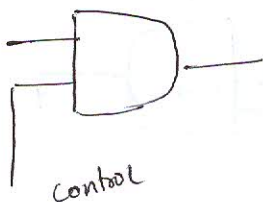
A → 0
 1 → 0
 0 → 0



output is not changed

↳ disable

A → A
 1 → 1
 0 → 0

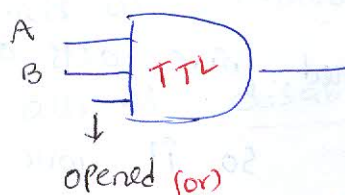


output is changed.

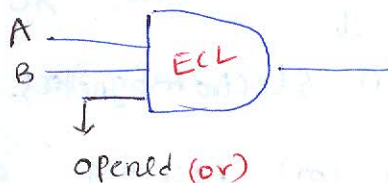
↳ enable

→ In TTL gates → opened or floating input will
 acts as Logic 1

→ where as in ECL logic family floating input
 will acts as Logic 0.



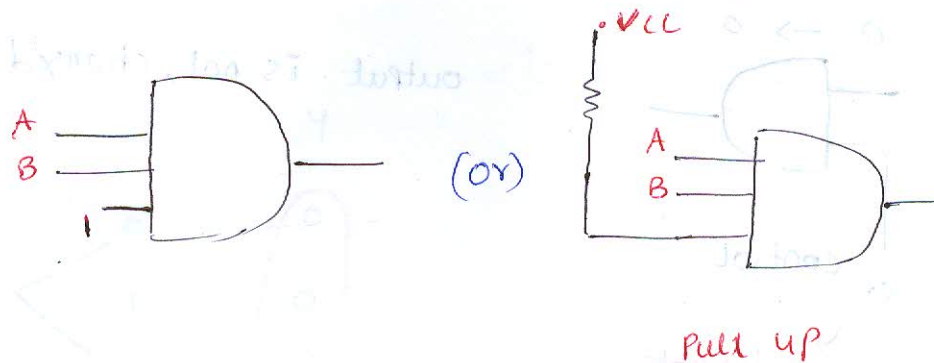
Floating is
 always logic 1



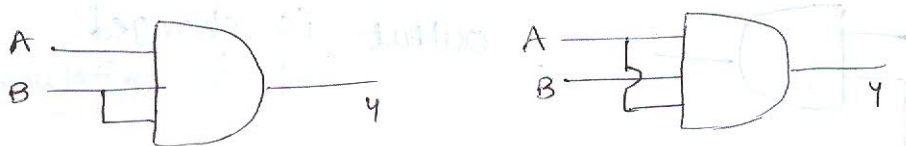
Floating is
 always logic 0

→ In multi input And gate unused input can be

1) connected to Logic 1 or pull up.



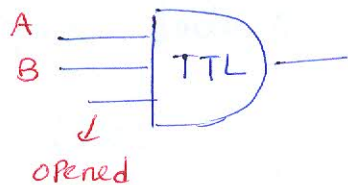
2) Connect to one of used input.



③ if it is TTL then unused ip can be opened or Floating

→ among these best way of connecting unused ip is connect them to logic 1 or Vcc

sol:



→ in Electromagnetics opened wire acts as an Transmitting (or) Receiving antenna so it will take an necessary signal so noise margin is increased it is not at all best way.

⇒ OR gate :-

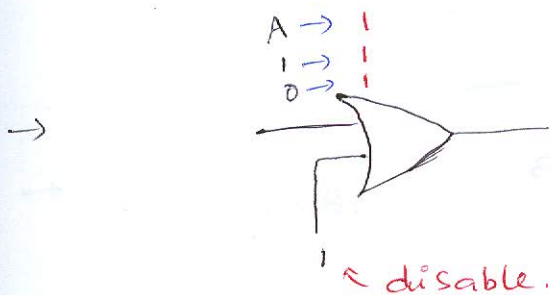
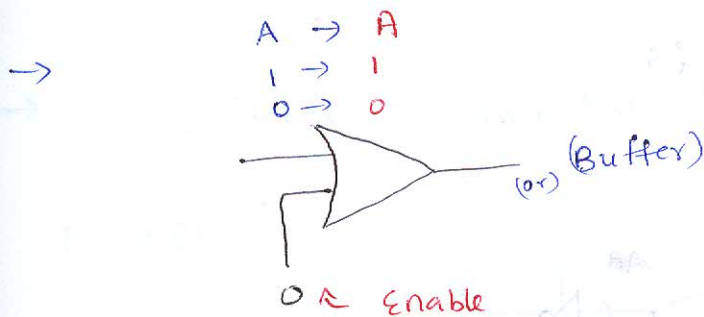
<u>A</u>	<u>B</u>	<u>Y</u>
0	0	0
0	1	1
1	0	1
1	1	1

→ OR gate follows both Commutative and Associative

laws

$$A+B = B+A$$

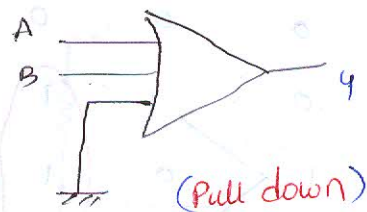
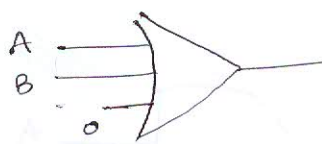
$$A+B+C = (A+B)+C = A+(B+C)$$



→ 0 acts as Enable in OR gate
1 acts as disable in OR gate.

→ In multi i/p OR gate unused i/p can be

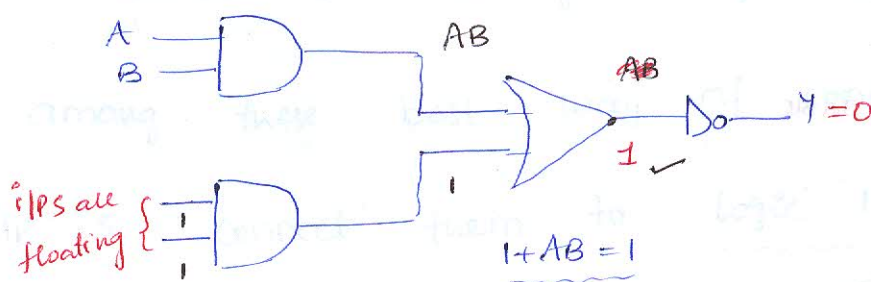
✓ 1) connected to Logic 0 (or) pull down



② Connect to one of used input

③ If it is ECL then unused i/p can be
opened (or) float

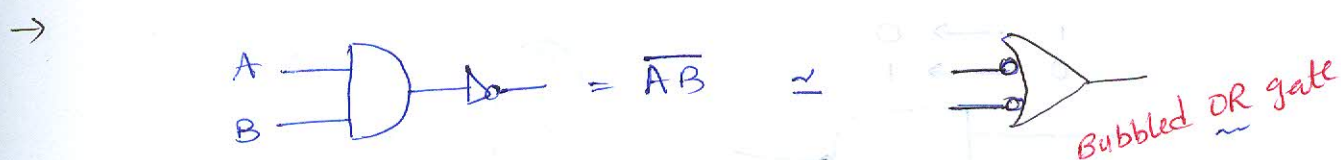
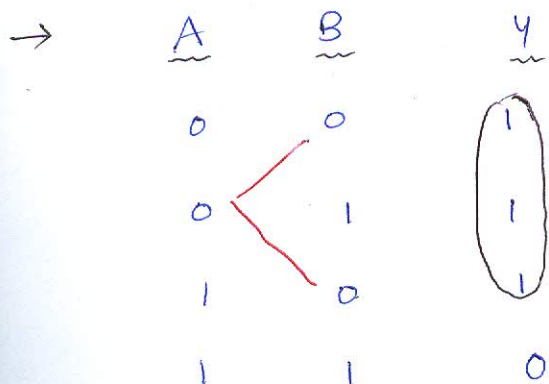
Q. The ckt shown in fig is TTL And-OR-Inverter ckt
for the given i/p's o/p y is



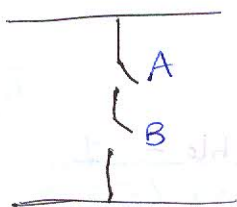
- a) 0 b) 1 c) $\overline{AB}$ d) AB

15/06/11.

⇒ NAND gate:-

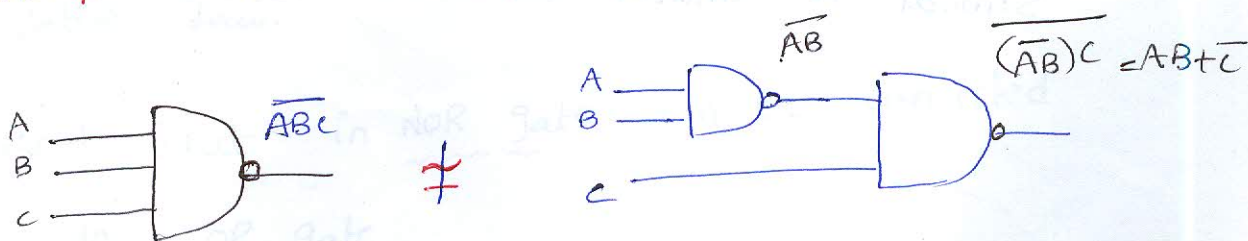


→ switches connected in series but they are in parallel.

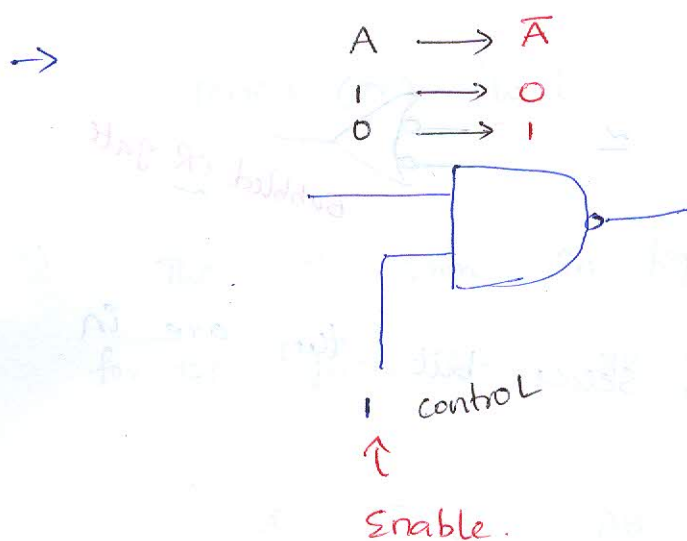
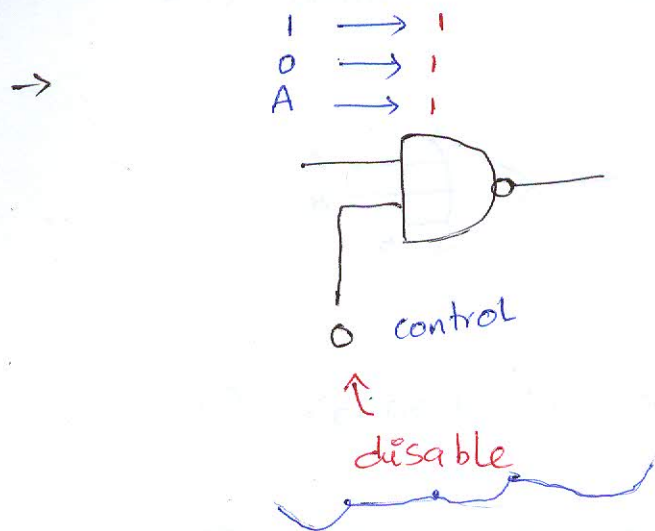


→ $\overline{AB} = \overline{BA}$

→ $\overline{ABC} \neq \overline{(\overline{AB})C}$



→ Nand gate follows Commutative law but not
Associative law.

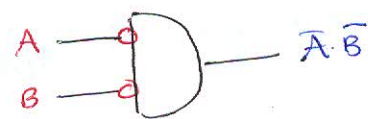
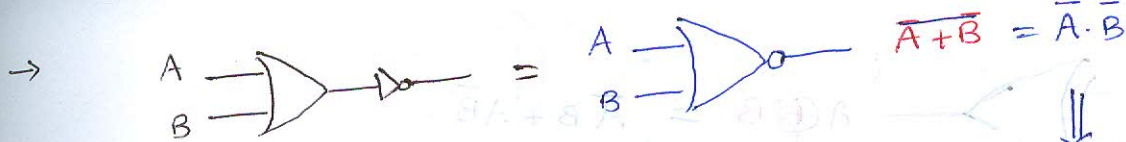


→ For Nand gate } enable - 1
 And gate } disable - 0

→ unused input in Nand gate can be connected
 Similar to unused input in And gate.



⇒ NOR gate :-



Bubbled and gate

→

<u>A</u>	<u>B</u>	<u>Y</u>
0	0	1
0	1	0
1	0	0
1	1	0

→

OR } enable → 0
 NOR } disable → 1

→ $\overline{A+B} = \overline{B+A}$

→ $\overline{A+B+C} \neq (\overline{A+B}) + C$

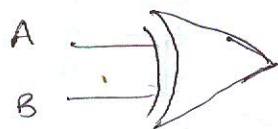
→ NOR gate follows commutative law but not
 Associative law.

→ un used input in NOR gate can be connected
 similar to OR gate.

⇒ EX-OR :-

(XOR)

→



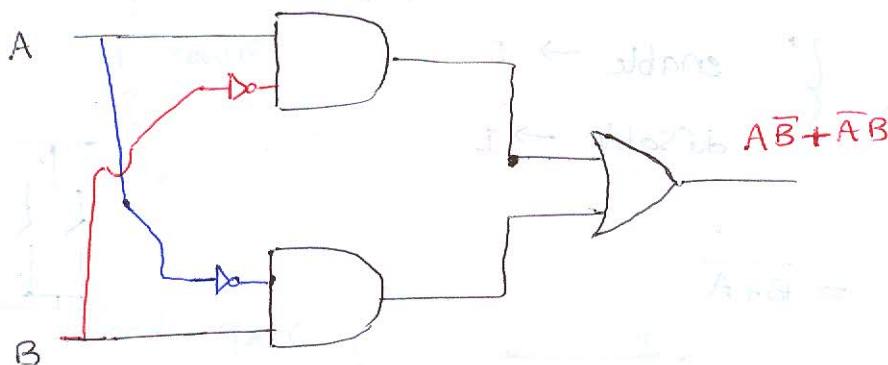
$$A \oplus B = \bar{A}B + A\bar{B}$$

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

→

	<u>A</u>	<u>B</u>	<u>Y</u>
<u>A+B</u> ←	0	0	0
	0	1	1 ✓
	1	0	1 ✓
<u>$\bar{A}+\bar{B}$</u> ←	1	1	0

→



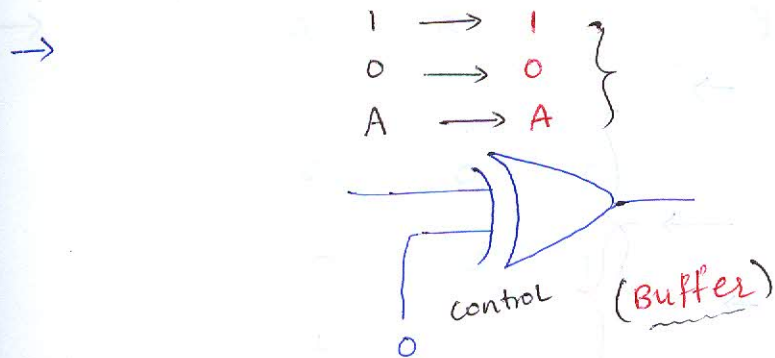
→

XOR gate is available with only two inputs
(XNOR)

→

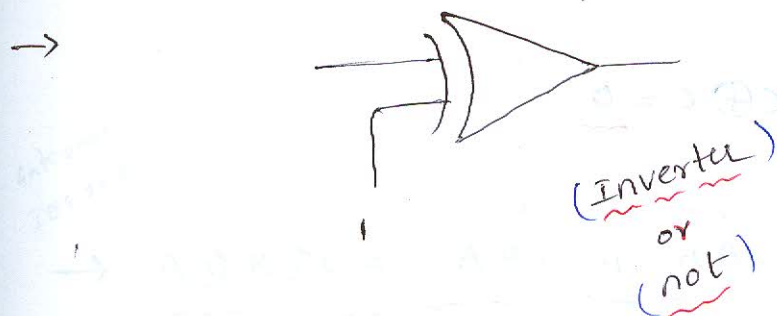
$$A \neq B \Rightarrow \underline{Y=1}$$

$$A=B \Rightarrow \underline{Y=0}$$



→ in Ex-OR gate if one of the input is Zero then it will act as an Buffer

A	→	$\bar{A}$	}
1	→	0	
0	→	1	



→ in Ex-OR gate if one of input is one then it will act as an inverter (or) Not gate

→ Ex-OR gate is known as Mod 2 ckt

→ $0 \oplus 0 = 0$

$0 \oplus 1 = 1$

$1 \oplus 0 = 1$

$1 \oplus 1 = 0$

discarding Carry is known

as modulo operation

2	1	0
x		
Carry	Sum	

→

$$\begin{aligned}
 A \oplus A &= 0 \rightarrow \\
 A \oplus \bar{A} &= 1 \rightarrow \\
 A \oplus 0 &= A \rightarrow \\
 A \oplus 1 &= \bar{A} \rightarrow
 \end{aligned}$$

Q3 if $A \oplus B = C$ Then

$$B \oplus C = A$$

$$A \oplus C = B$$

$$A \oplus B \oplus C = C \oplus C = \underline{0}$$

Q3 $A \oplus A = 0$ then

$$A \oplus A \oplus A = A \checkmark$$

$$A \oplus A \oplus A \oplus A = 0 \checkmark$$

→

$$A \oplus A \oplus A \oplus \dots \oplus A = n =$$

$$= 0 \rightarrow n \text{ is even}$$

$$= A \rightarrow n \text{ is odd.}$$

→ In EX-OR o/p is 1 when no. of 1's at the input is odd number.

→ EX-OR gate is called (odd no. of 1's detector)

→

$$Y = A \oplus B \oplus C$$

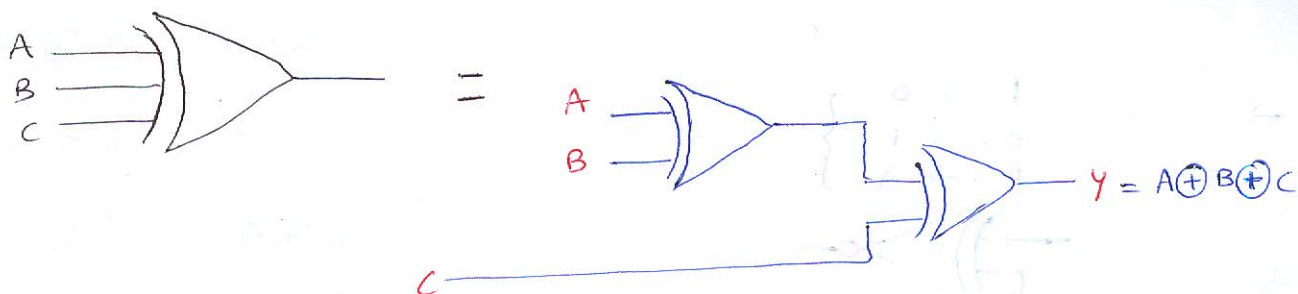
<u>A</u>	<u>B</u>	<u>C</u>	<u>Y</u>
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Gate 2007
IES 2010.

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

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

→



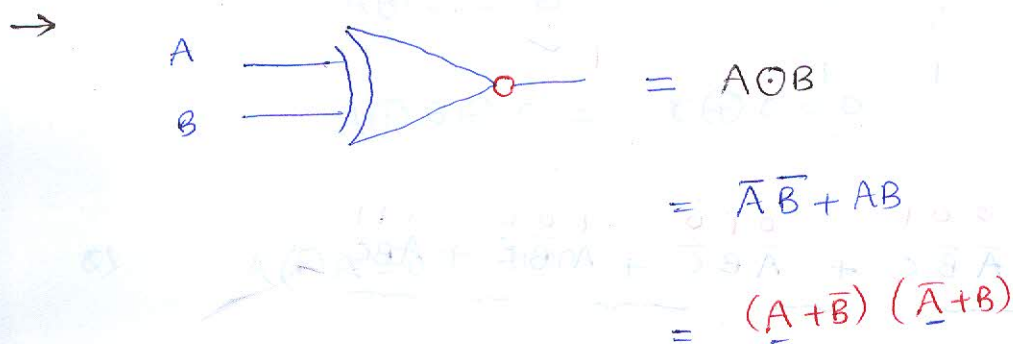
→ Ex-OR gate is an three variable Parity generator.

→ Ex-OR gate is an stair case switch

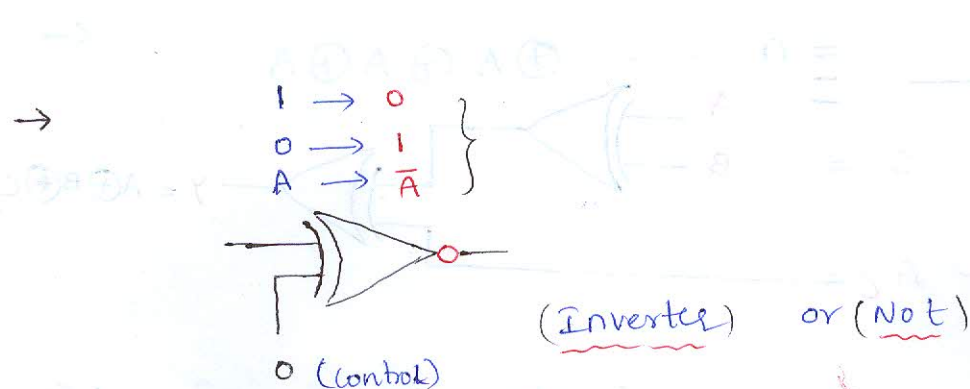
⇒ EX-NOR :-
(XNOR)

→

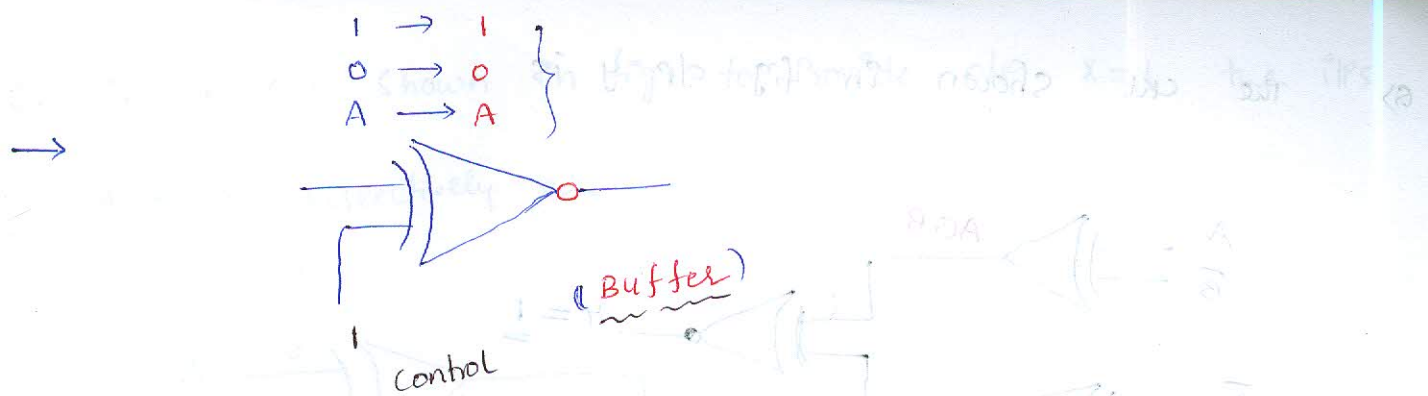
<u>A</u>	<u>B</u>	<u>Y</u>	
0	0	1	✓
0	1	0	
1	0	0	> Pos
1	1	1	✓



→ $A=B \Rightarrow Y=1$ [Equivalence kkt] or (Coincidence logic kkt) ✓
 $A \neq B \Rightarrow Y=0$



→ EX-NOR when input control is 0 then it will
 acts as an inverter or Not gate



→ in EX NOR when input is 1 then it will act as an Buffer

→

$$\begin{aligned}
 A \odot A &= 1 \\
 A \odot \bar{A} &= 0 \\
 A \odot 0 &= \bar{A} \\
 A \odot 1 &= A
 \end{aligned}$$

Q

what $A \oplus \bar{B}$

↓

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

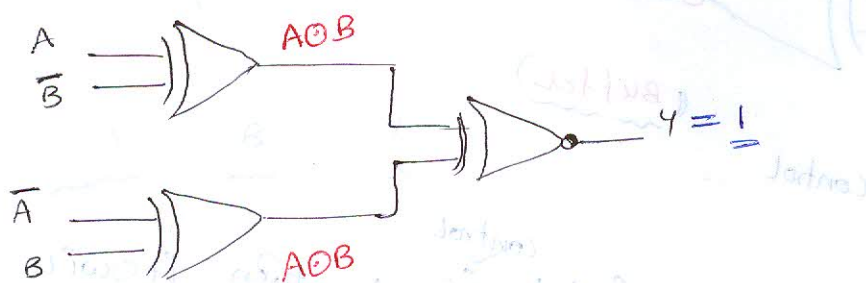
→ $A \oplus \bar{B} = A \odot B$

→ $\bar{A} \oplus B = A \odot B$

→ $A \odot \bar{B} = A \oplus B$

→ $\bar{A} \odot B = A \oplus B$

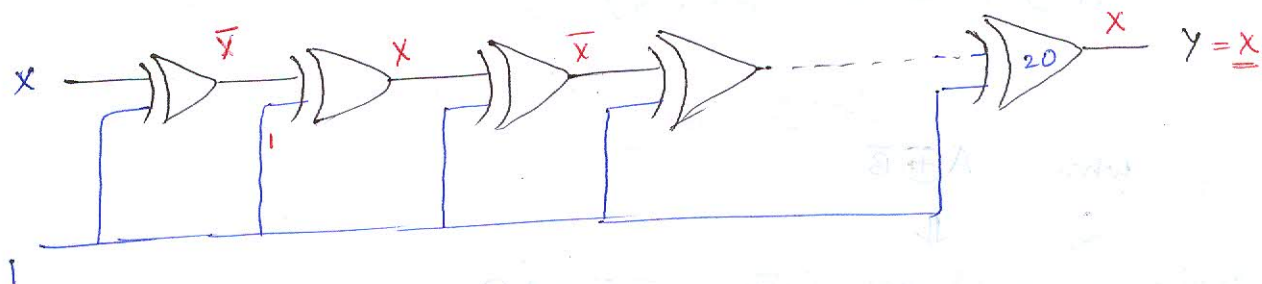
Q) The ckt shown in fig o/p y is



Sol:-

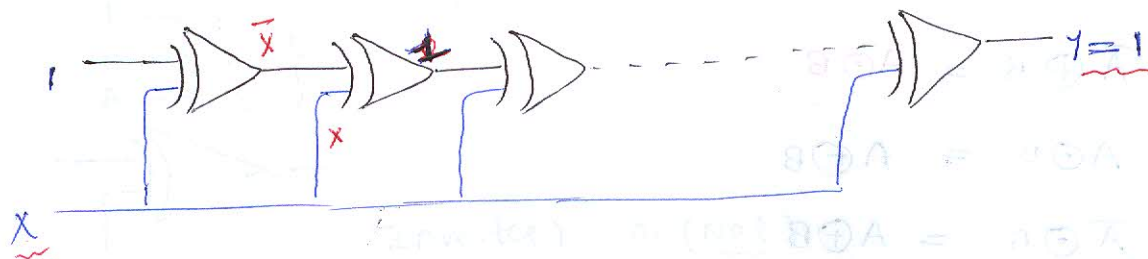


Q) The ckt shown in fig have 20 EX-OR gates then for the given i/p's o/p y is

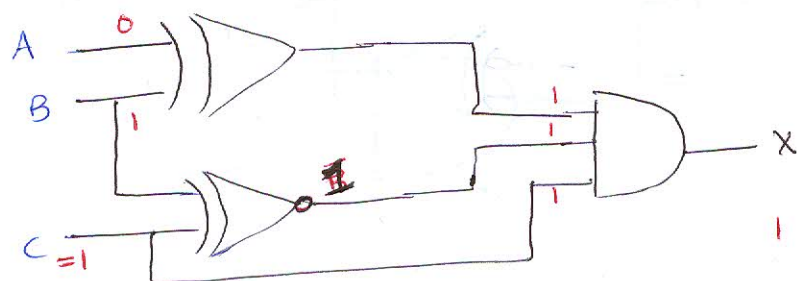


- a) 0 b) 1 c) x d) $\bar{x}$

Q)



Q) in the ckt shown in fig to provide o/p $X=1$ the i/ps A B c respectively are.

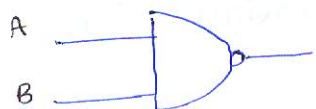


- a) 001 b) 101 c) 110 d) 011

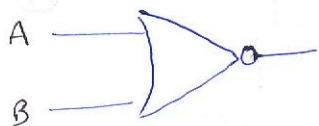
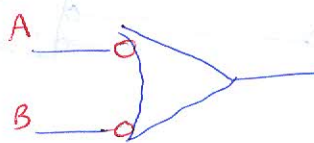
Q) $X \oplus Y \oplus XY = X + Y$ ✓

⇒ Alternative Symbols :-

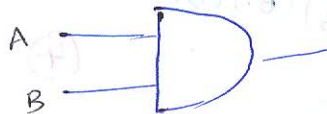
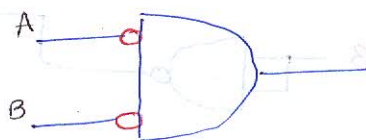
→



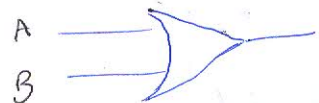
=



=

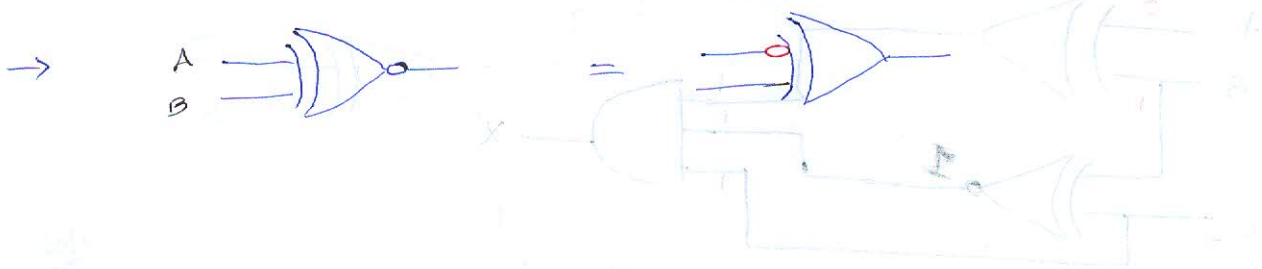
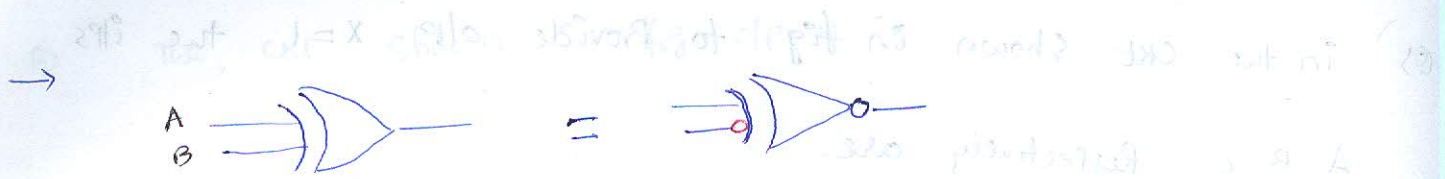


=

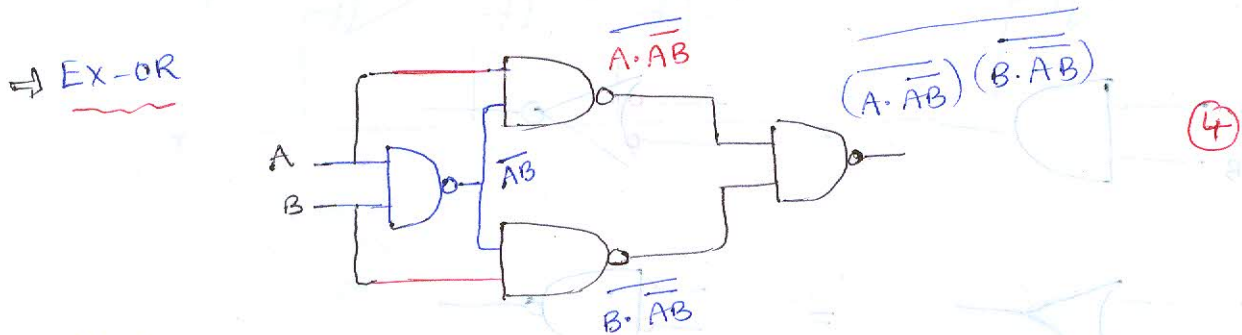
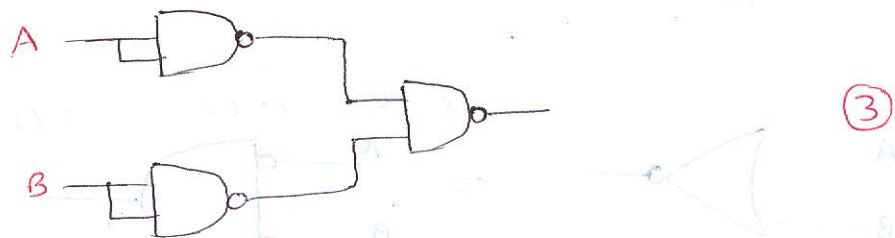
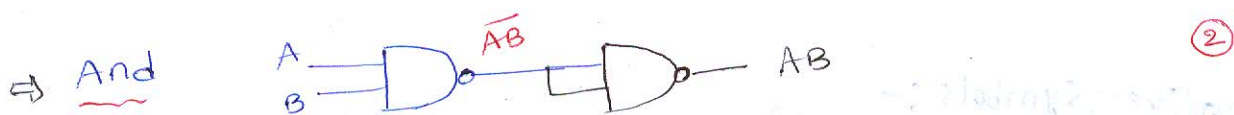
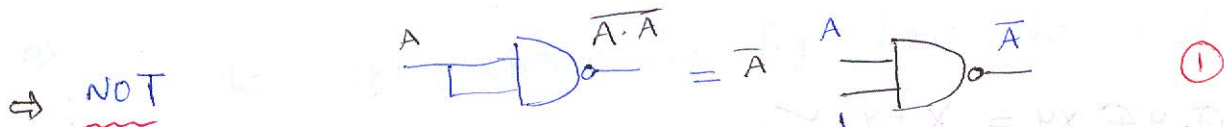


=



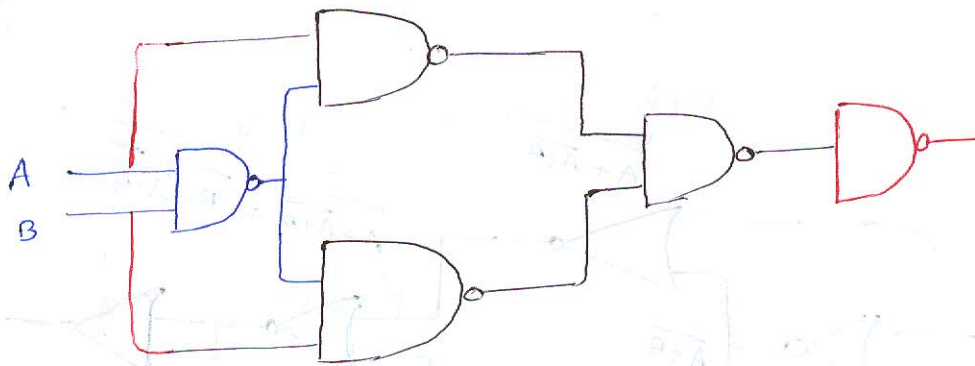


⇒ NAND as universal gate :-



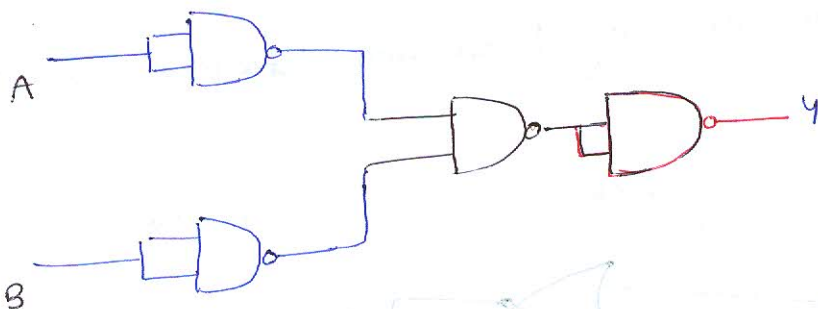
Ex-NOR

⇒ Ex-NOR



⑤

⇒ NOR



④

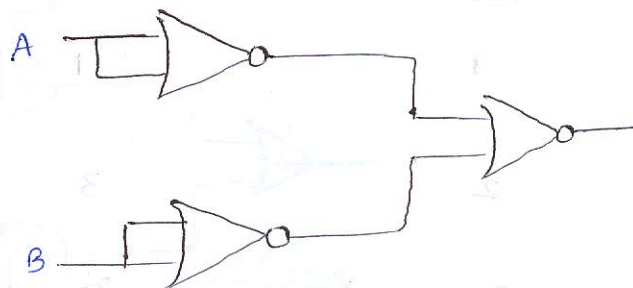
⇒ NOR as universal gate :-

⇒ NOT



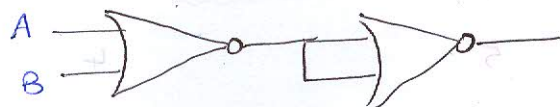
①

⇒ And



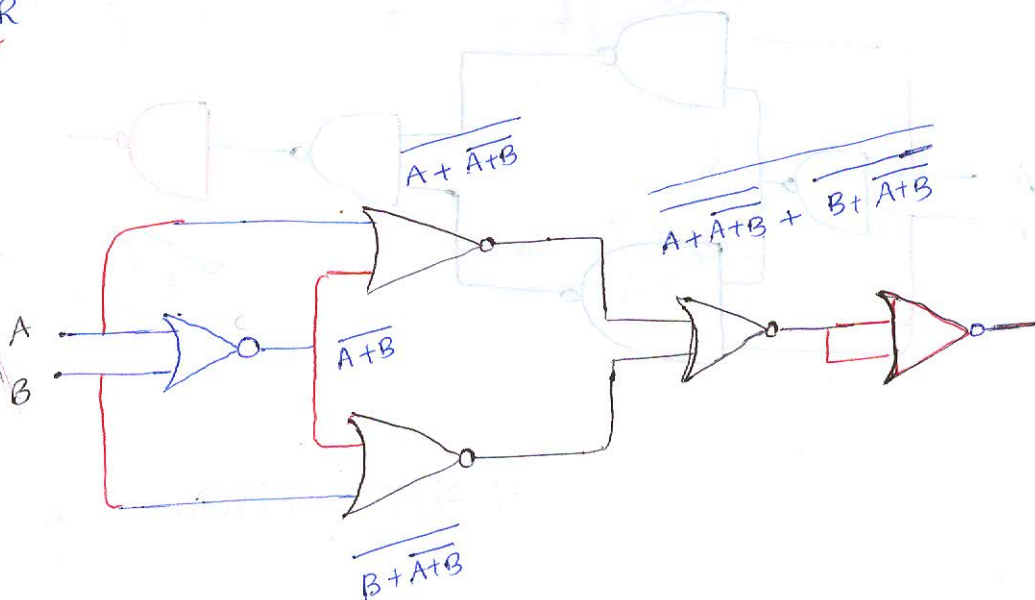
③

⇒ OR

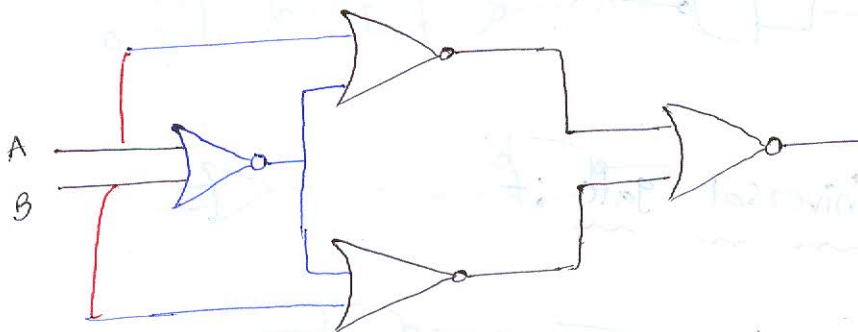


②

→ EX-OR



→ EX-NOR



→ Logic gate

No. of NAND

No. of NOR

NOT

1

1

AND

2

3

OR

3

2

EX OR

4

5

EXNOR

5

4

NOR

4

1

NAND

1

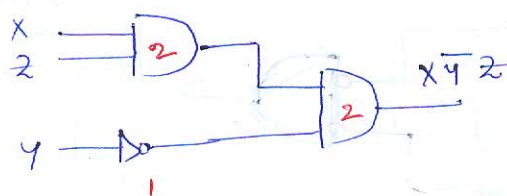
4

Q) TO implement $x\bar{y}z$ min no. of two i/p Nand gates required

- a) 3 b) 4 c) 5 d) 7

Sol

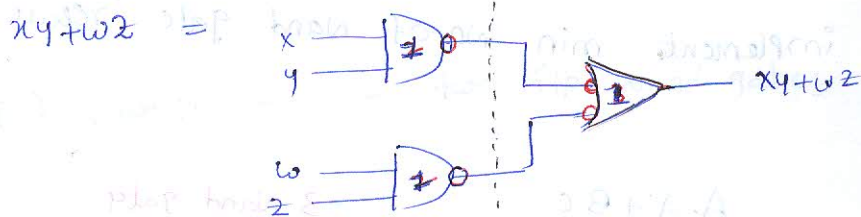
$$x\bar{y}z = (xz)\bar{y}$$



Q) TO implement $xy + wz$ min no. of two i/p Nand gates required.

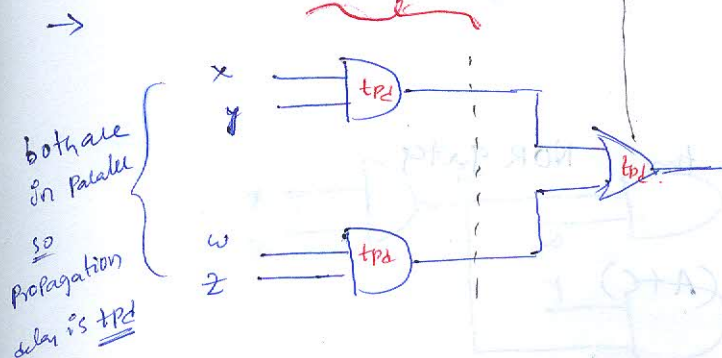
- a) 3 b) 4 c) 5 d) 7

Sol



Two level NAND-NAND

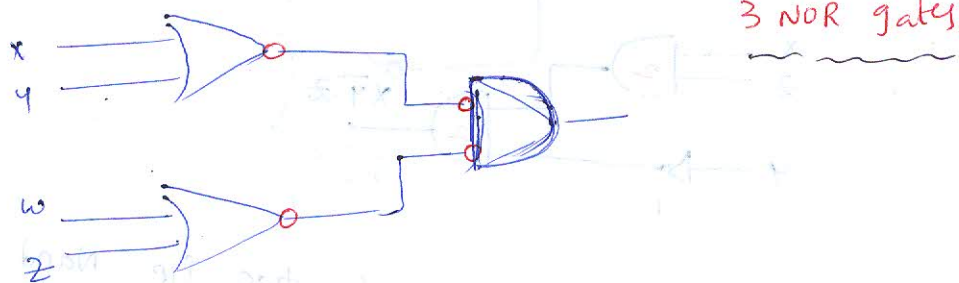
$$t_{pd} + t_{pd} = 2t_{pd} \quad \text{[Two level]}$$



two level And-OR = Two level NAND-NAND

→ SOP ckt's can be implemented with min no. of
Nand gates

→ $(x+y)(w+z)$



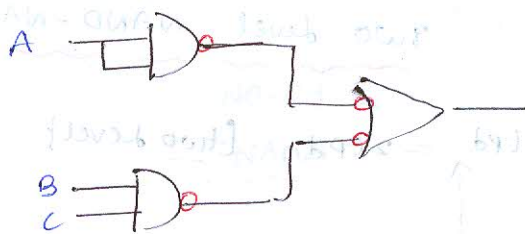
→ POS ckt's can be implemented with min no.
of NOR gates.

Q) To implement min no. of nand gate realise $A+BC$

sol

$$A \cdot A + BC$$

3 Nand gates



Q)

$$A+BC$$

implement

thr NOR gates

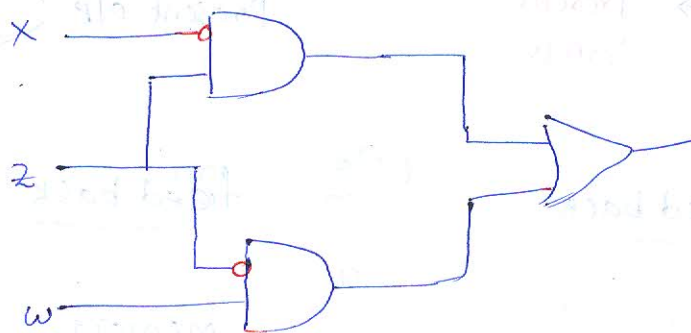
$$A+BC = (A+B)(A+C)$$

3 NOR gates

Q1 To implement $(\bar{x} + \bar{z})(w + z)$ $\longrightarrow$ min no. of two i/p nand gates required.

Sol

$$\bar{x}z + \bar{z}w$$



$$\bar{x}\bar{z}(w+z)$$

$\underline{A}(w+z) = \underline{Aw + Az} \rightarrow 4$ nand gates are required

Q2

$(\bar{x} + \bar{z})(w + y)$ $\longrightarrow$ two i/p nand gates

Sol

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

$$A(w+y)$$

$(Aw + Ay) \rightarrow$ min 4 nand gates

