Logic gates —

NOT, AND, OR → Basic gate

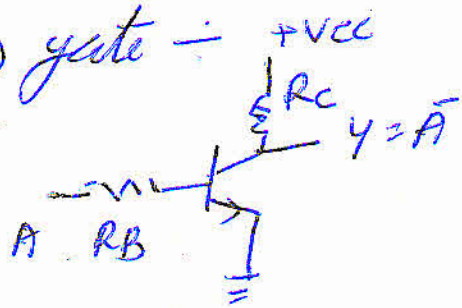
NAND, NOR → Universal gate

Buffer → 

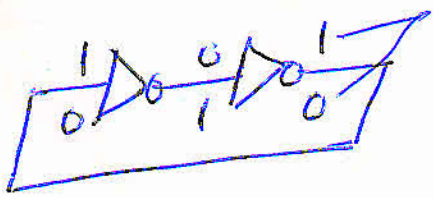
Buffer produces transfer function and does not produce any logical operation. This ckt is mainly used for power amplification of signal and is equivalent to two inverter in cascade.

* All gets except Buffer and Inverter have multiple Inputs

⇒ Not (Inverter) gate —

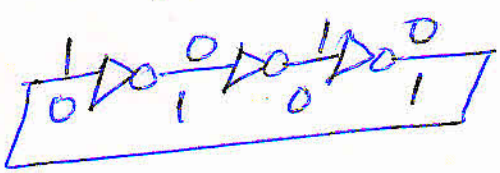


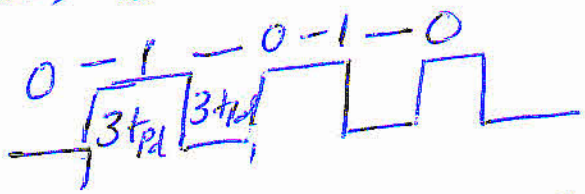
A	F
0	1
1	0



Two stable State
Basic memory element
Bistable multivibrator

* Even No. of NOT GATE → BMV.





* ODD No of NOT GATE — AMV

$= 6 t_{pd}$

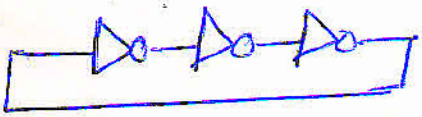
The generated Square wave time period $= 2n\, t_{pd}$

$= 2 \times n \times t_{pd}$

n = No of Inverter

t_{pd} = propagation delay of each Inverter

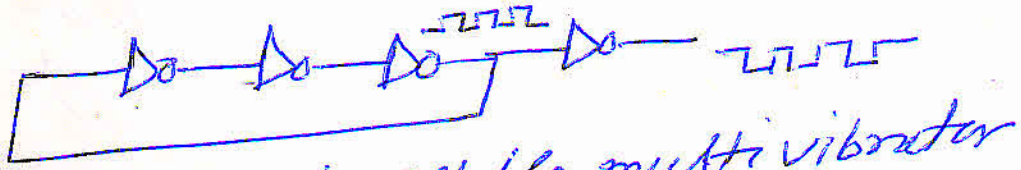
i) AMV
ii) Sq wave gen
iii) Ring oscillator
iv) clock generator

⇒  $P_{id} = 25ns$

time period of generated waveform

$$T = 2 \times 3 \times 25 \times 10^{-9} = 150ns$$

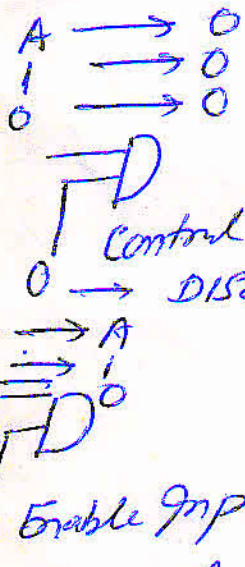
$$f_{req.} = \frac{1}{T} = \frac{10^9}{150} = 6.6\,MHz$$

⇒ 

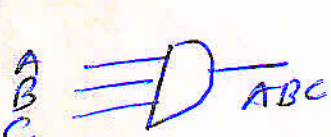
The circuit is astable multivibrator

AND GATE →

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



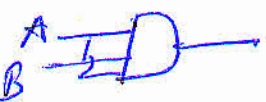
* AND GATE follow → Commutative & Associative law



pull up

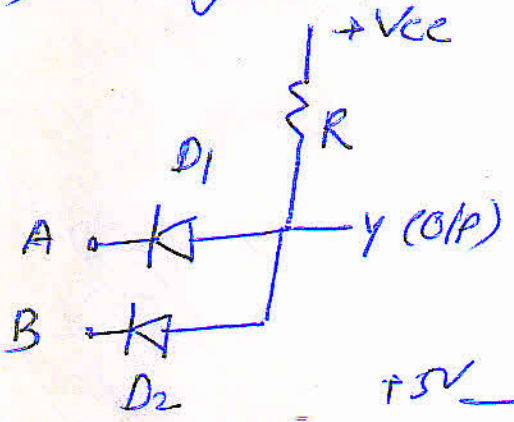
↑ Unused input — Best method

power loss & Draw More current



Unused I/P Can be connected to Anyone of used I/P

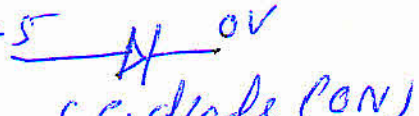
⇒ AND gate operation by diode gate



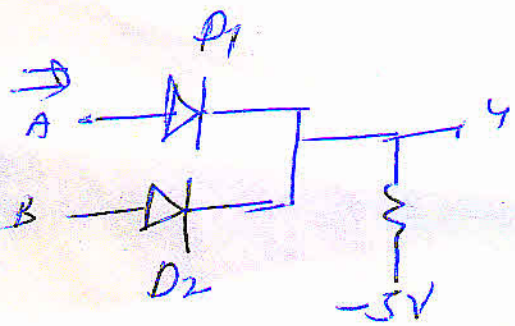
0 → 0V
1 → 5V

A	B	D_1	D_2	Y
0	0	on	on	0
0	1	on	off	0
1	0	off	on	0
1	1	off	off	1

O.C. diode
off



S.C. diode (ON)



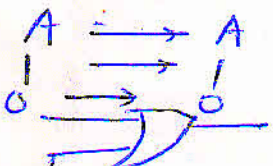
If Negative logic is used the diode GATE will Represent.

0 = 0V
1 = -5V

		D_1	D_2	Y
0 0	0 0	ON	ON	0V → 0
0 1	0 -5	ON	OFF	0V → 0
1 0	-5 0	OFF	ON	0V → 0
1 1	-5 -5	OFF	OFF	-5V → 1

OR GATE →

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



0 → Enable Input

Buffer

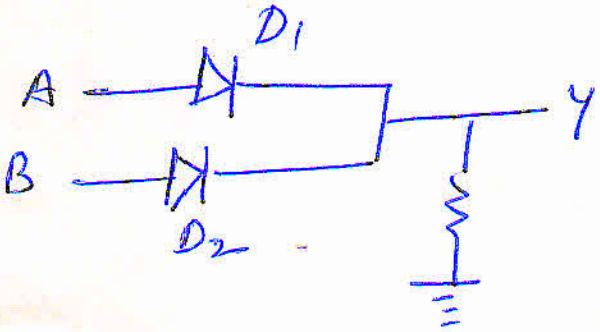
1 → Disable Input

* OR GATE follow associative and commelave law

A+B = B+A, (A+B)+C = A+B+C

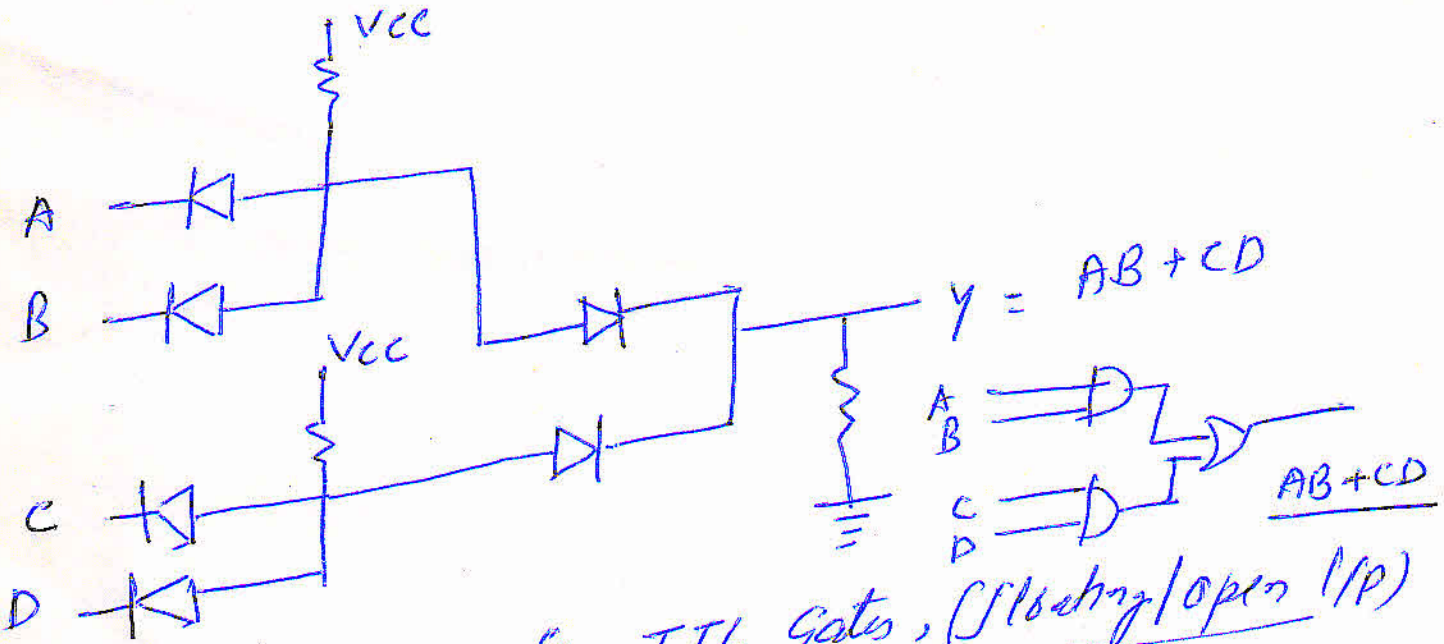
⇒ The unused I/P in the OR gate can be connected to logic 0 (ground) → pull down.

⇒ connected to any one of the used I/P



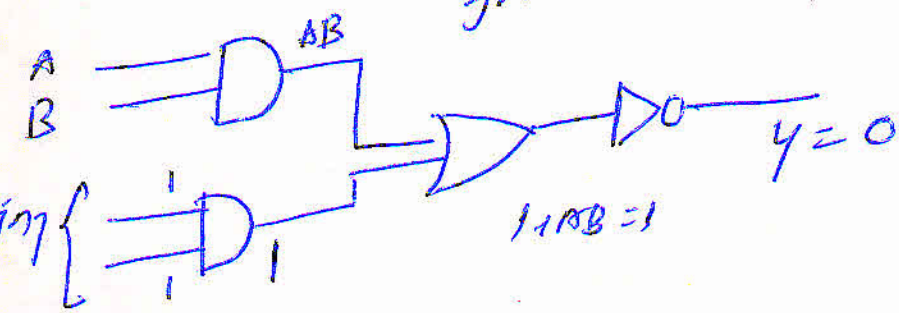
A	B	D_1	D_2	Y
0	0	off	off	0
0	1	off	on	1
1	0	on	off	1
1	1	on	on	1

⇒



for TTL Gates, (floating/open I/P) → logic 1

⇒



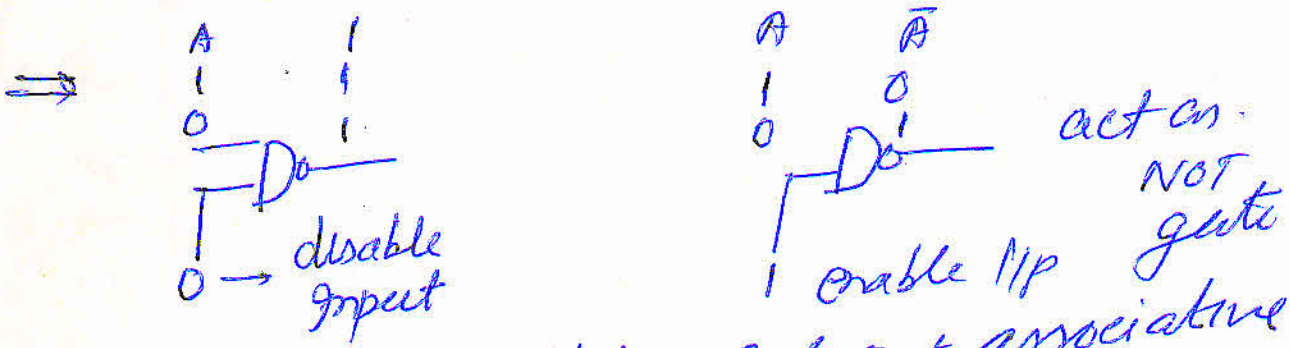
↓ mean

1

Note → for ECL Gate floating input is at logic '0' (open I/P)

NAND Gate → AND followed by NOT

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



* NAND follow commutative but not associative
* Unused I/p in NAND gate will function as AND gate

NOR Gate → OR Gate followed by NOT

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

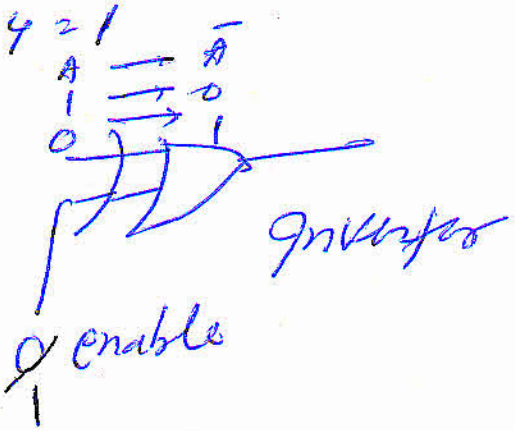
Enable = 0
Disable = 1

* Don't follow Associate law
* Unused I/p in NOR gate will function to OR gate.

EX-OR Gate :-

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

for two Input
A = B ⇒ Y = 0
A ≠ B ⇒ Y = 1

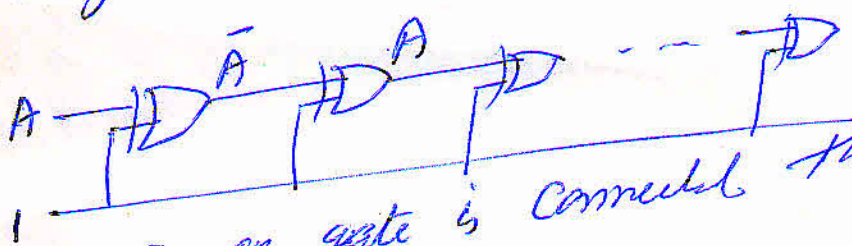


$\Rightarrow$ $A \oplus A = 0$
$A \oplus \bar{A} = 1$
$A \oplus 0 = A$
$A \oplus 1 = \bar{A}$ ← Self Inverse

$\Rightarrow$ If $A \oplus A \oplus A$ — n term.
$= 0$ If n is even
$= A$ If n is odd

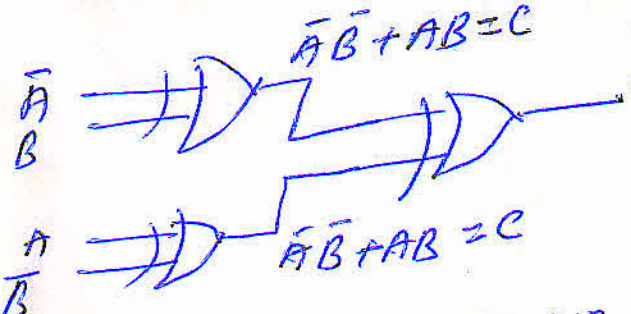
* EX-OR is Commutative and Associative
* EX-OR is called odd 1's detector. because its O/P is 1 only when NO. of 1's in the Inputs are odd.
* EX-OR is also used in parity generator or parity checkers.

$\Rightarrow$



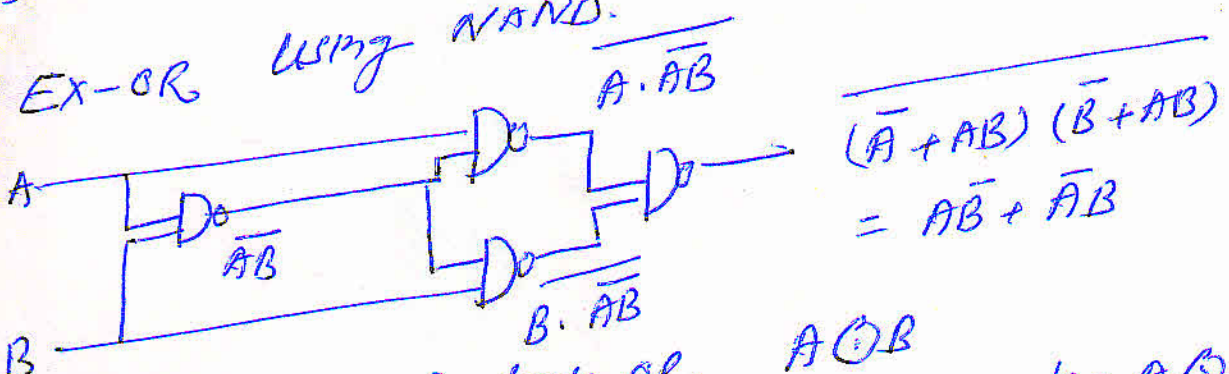
If 10 EX-OR gate is connected the O/P is
Hence $Y = A$

$\Rightarrow$

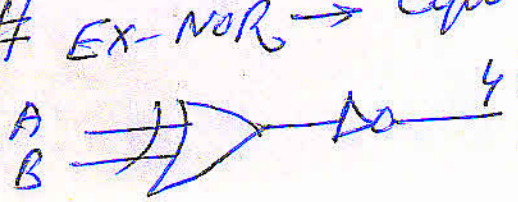


Same I/P
$Y = 0$

$\Rightarrow$ EX-OR using NAND.

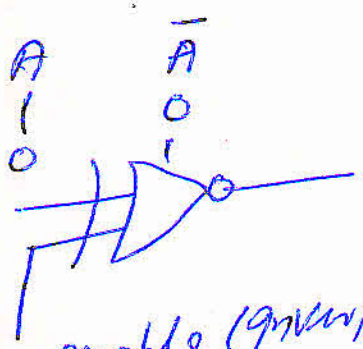


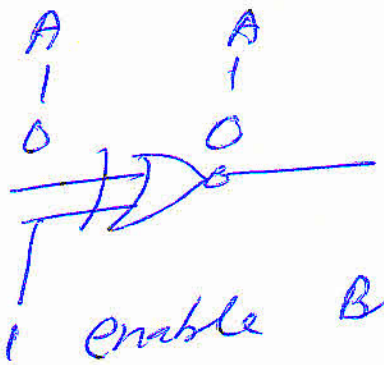
EX-NOR → equivalence OR $A \odot B$



If $A = B$ $Y = 1$
$A \neq B$ $Y = 0$

⇒





* It is used as equality detector, Also known as Coincidence logic
* EX-NOR is used even 1's detector O/P is 1 when No. of 1's in I/P is even.
* Ex-NOR follows Associative and Commutative law

⇒ $A \odot A = 1$ ⇒ $A \odot A \odot A \cdots$ n-term
$A \odot \bar{A} = 0$ $= 1$ if n is even.
$A \odot 0 = \bar{A}$ $= A$ if n is odd
$A \odot 1 = A$

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

* used in parity generator / parity checker
* $A \oplus B = \overline{A \odot B}$ ⇒ for two variable Ex-OR - X-NOR are complementary.
* $A \oplus B \oplus C = A \odot B \odot C$ ⇒ for three variable

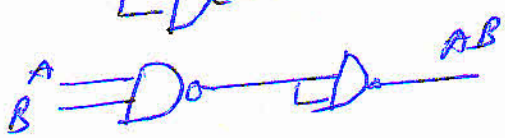
Universal logic gates → X-OR : X-NOR
NAND - NOR

Adv: * In case of shortage of any gate one may use universal gate
* Universal gate may be produced in bulk so as to reduce the cost

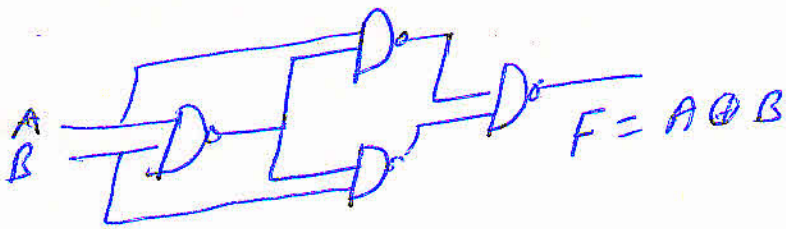
Disadv * Larger No. of gate may be required occupying more space
* Larger No. of gate slows down the speed of operation.

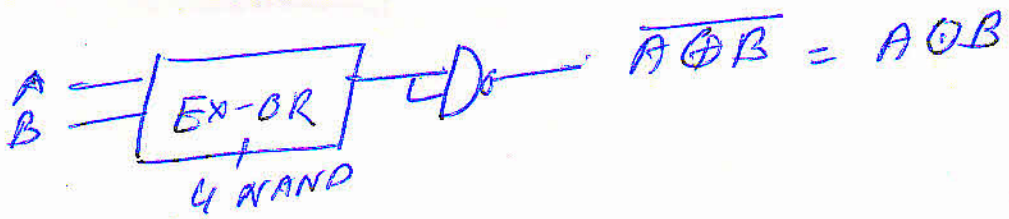
⇒ NAND as Universal gate

NOT → 

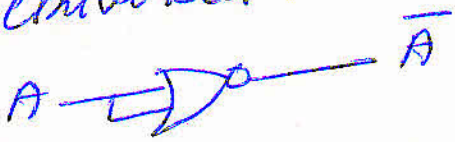
AND → 

OR → 

⇒ Ex-OR → 

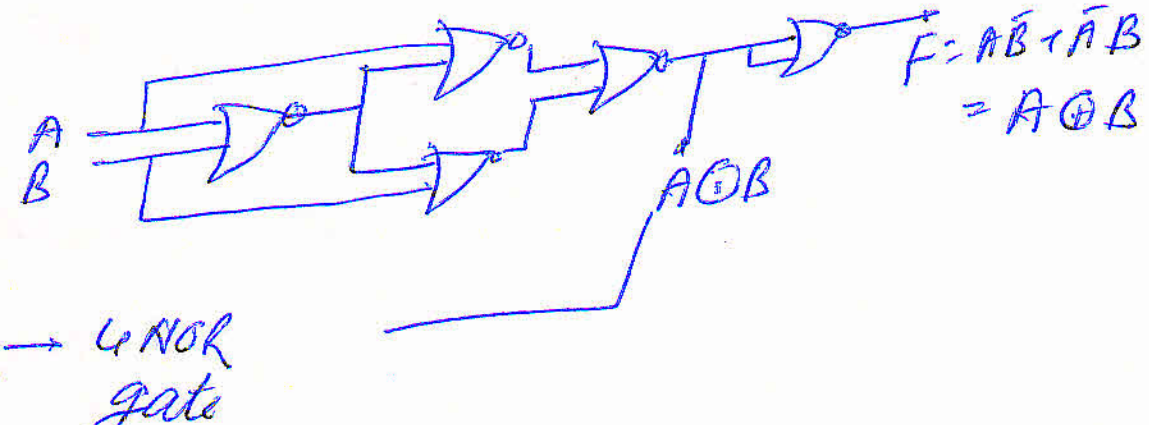
⇒ Ex-NOR → 

NOR as Universal.

NOT — 

AND — 

OR — 

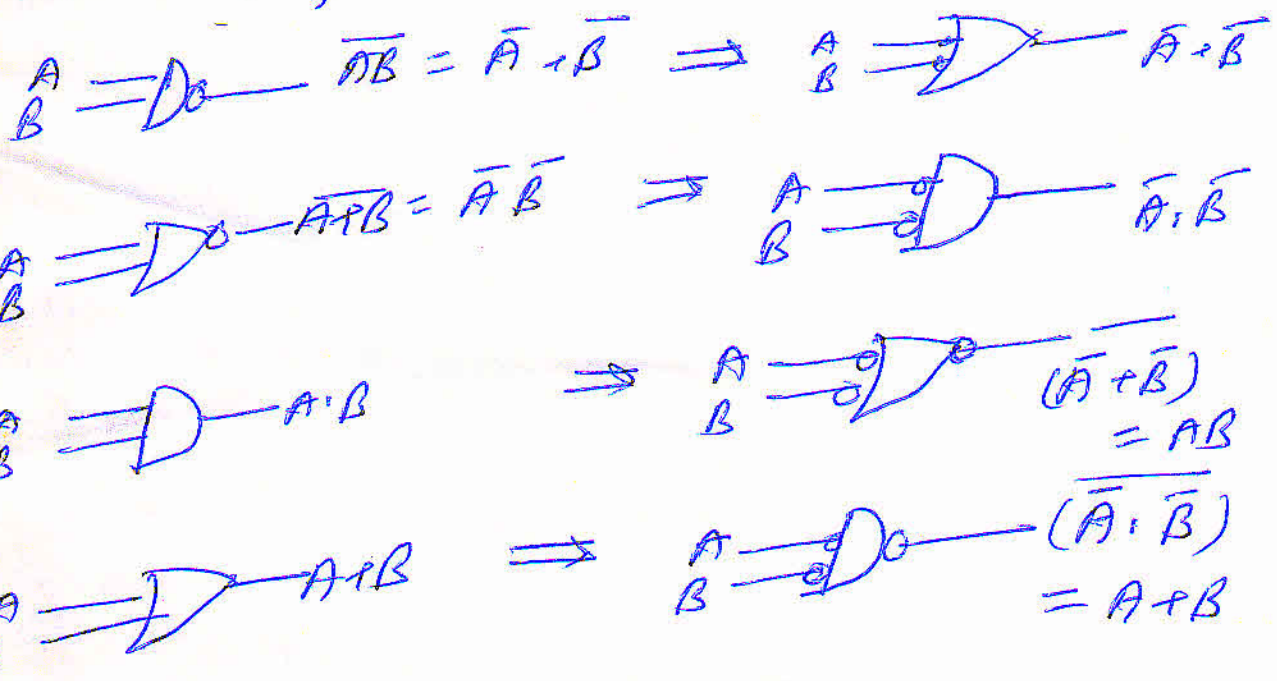
Ex-OR — 

Ex-NOR → 4 NOR gate

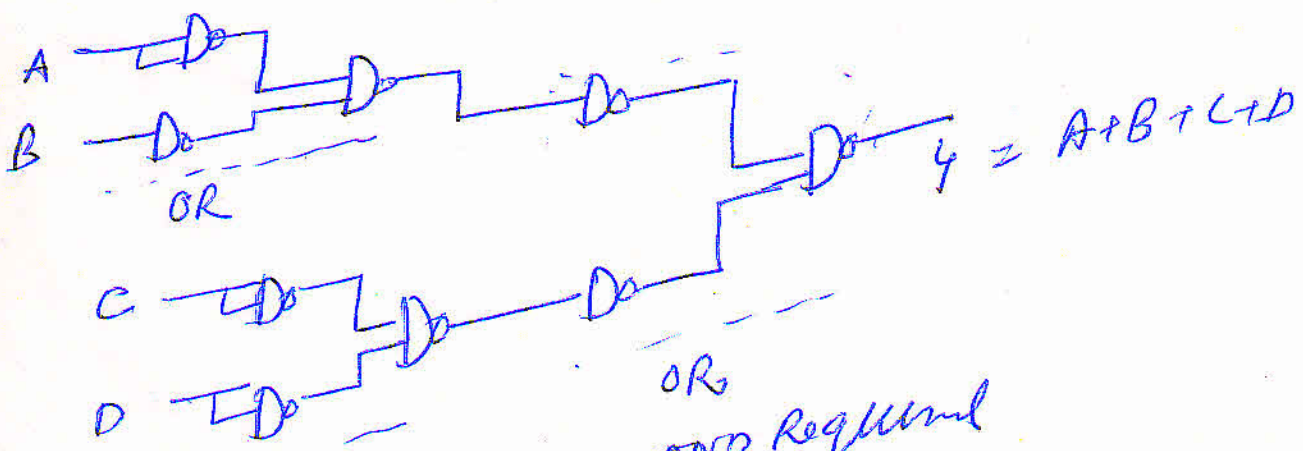
⇒

logic gate	No. of NAND	No. of NOR
NOT	1	1
AND	2	3
OR	3	2
Ex-OR	4	5
Ex-NOR	5	4

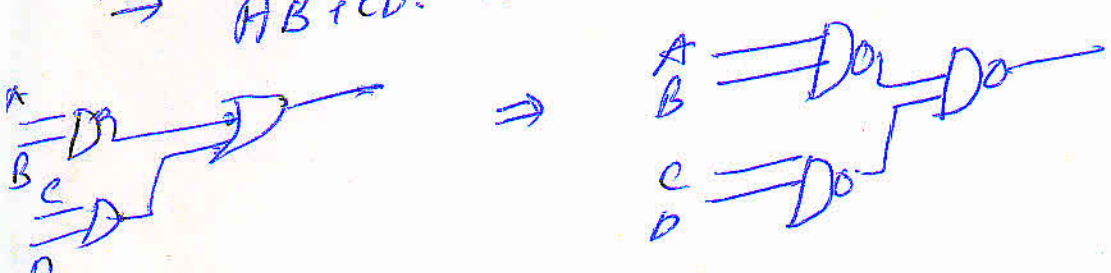
⇒ Bubbled Inputs.



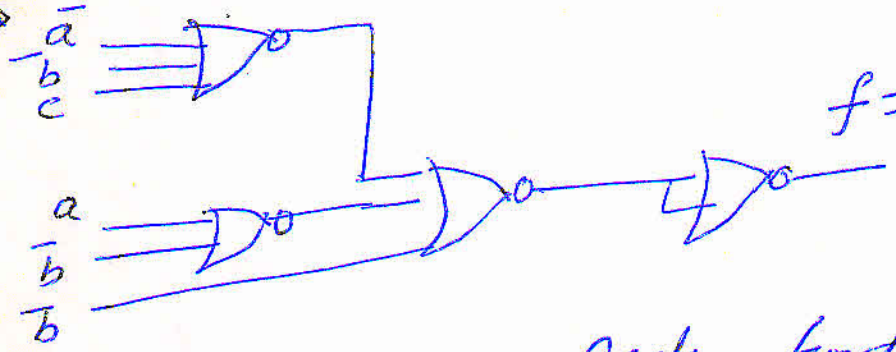
⇒



⇒ AB + CD. MIN. No of NAND Required



3 NAND.

⇒ 

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

Implementation of Boolean functn. by NAND and NOR gate —

SOP → NAND – NAND ⇒ AND-OR

POS → NOR – NOR ⇒ OR-AND

⇒ $F = AB + CD$

AND - OR

⇒

NAND – NAND

⇒ $F = (P+Q)(R+S)$

OR - AND

⇒

NOR – NOR

⇒ No of gate in one 14 PIN IC → different type

n → No. of input, m = No of gate of various type

$(n+1) \times m + 2 = 14$

* 2 I/P EXOR — $3 \times m + 2 = 14 \Leftarrow m = 4$
* 3 I/P NAND — $4 \times m + 2 = 14 \Leftarrow m = 3$
* 4 I/P NOR — $5 \times m + 2 = 14 \Leftarrow m = 2$
* 5 I/P OR — $6 \times m + 2 = 14 \Leftarrow m = 2$