

LOGIC GATES:

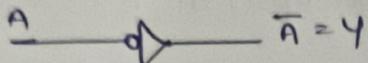
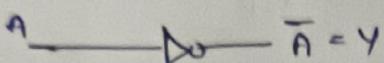
⇒ Basic Building Blocks.

NOT }
AND }
OR }

NAND }
NOT }

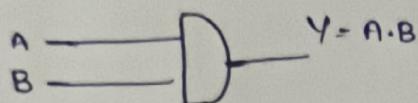
EXOR }
EXNOR }
Arithmetic ckt.
used for comparators, parity generator, checker, code converter
(Binary to gray, gray to Binary).

NOT:



A	Y
0	1
1	0

AND:

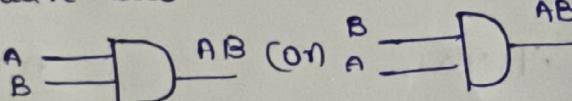


⇒ O/p is low, if any i/p is low
i.e logic '0'.

⇒ AND gate follows both commutative law and Associative law

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

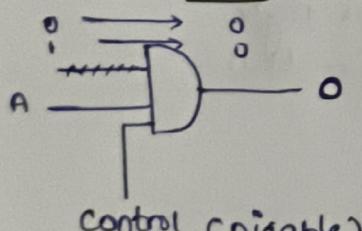
$$(1) AB = BA$$



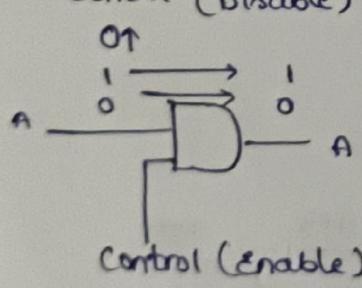
$$(2) ABC = (AB) \cdot C = A \cdot (BC)$$



⇒ Disable & Enable: depends on control pin



⇒ Thus O/p remain "0" due to control i/p disable. And gate is not in working state.



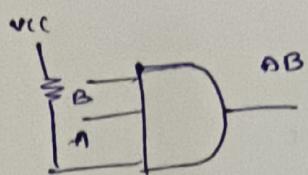
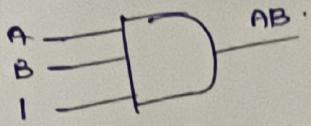
⇒ AND gate is in working state, o/p is changing in Enable state.

⇒ In TTL logic family, If any i/p is open and float then it will act as '1'

⇒ In ECL family, floating i/p will act as ≈ 0.10 .

→ Enable and disable:

unused I/P:

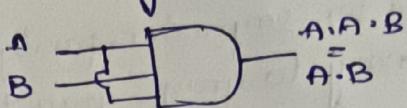
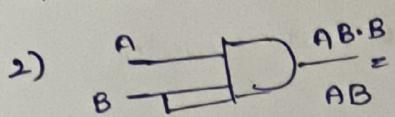


$$1 \cdot AB = AB$$

→ In multipin (TIP) AND gate unused I/P can be connected to logic "1" or

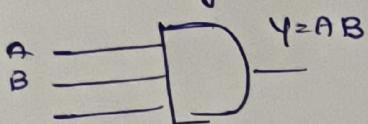
"pull up"

→ unused I/P can be connected to logic "0" or "pull down"



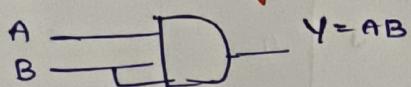
→ unused I/P can be connected to one of the used I/P.

3) If it is TTL logic family, then unused I/P can be open or floated
then it is logic '1'

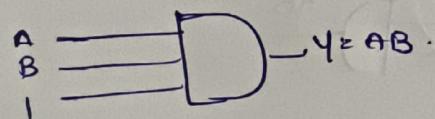


only for TTL).

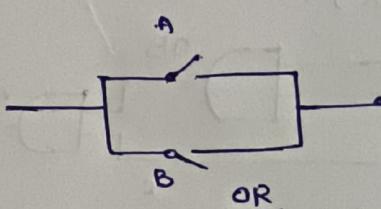
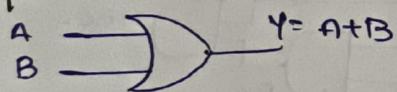
Note: Because of unnecessary I/P attached to B, In will go down



→ Best way to connecting unused pin (I/P) in AND gate is connecting to logic '1'.



OR Gate: (Inclusive OR)

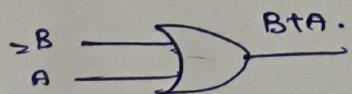
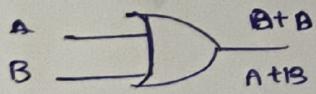


A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

⇒ when any of the I/P is high in OR gate then O/P is high

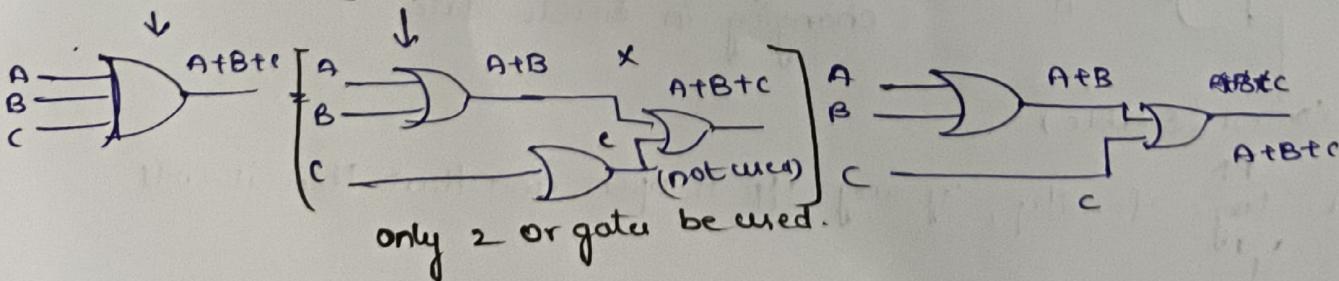
⇒ OR gate follows commutative law and associative law.

i) $A+B = B+A$.

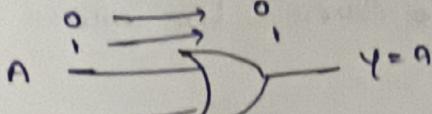


ii) Associative Law:

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

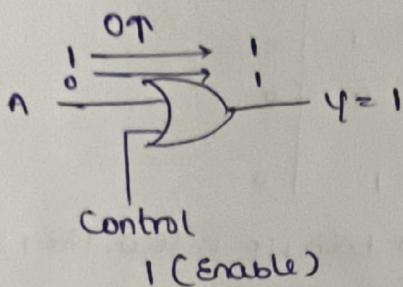


⇒ Enable and disable:



⇒ o/p is changing as I/p is changing or we say the gate is work state in enable state

Control (enable).



⇒ o/p is not changed, and it is said to be disable.

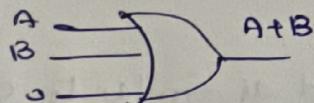
Unused I/p's:

1. In OR gate, unused I/p is connected to logic '0' or "pull down"

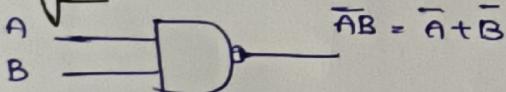
2. Connected to the one of the used I/p

3. It is the FCL then unused I/p can be open or floated.

⇒ In OR gate, Best way of connecting the unused I/p is connect to logic '0'.

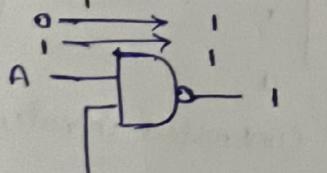


NAND gate:

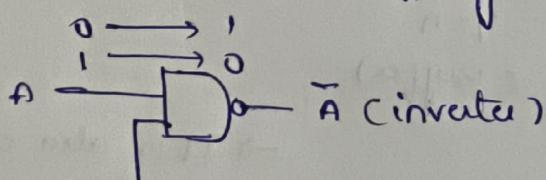


A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

⇒ Enable & disable:



⇒ when both I/p high the o/p is low



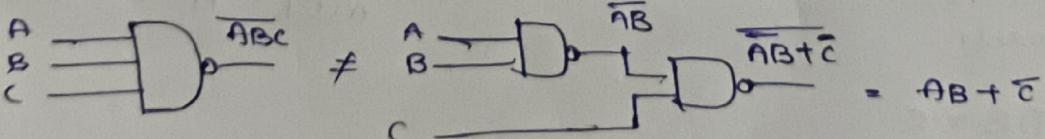
Control (enable)

0↑
not changing

Control (enable)

1↑

⇒ NAND gate follow commutative law but not follow associative law.



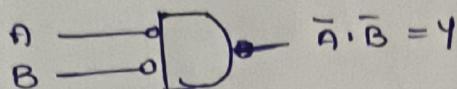
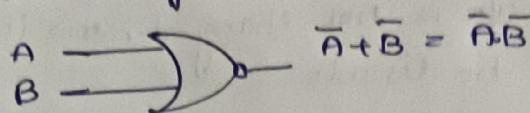
⇒ The only two gates all not follow associative law i.e universal gate (NAND and NOR gate)

unused i/p's:

⇒ unused i/p in NAND gate can be connected to same as the unused i/p in AND gate (logic "1")

NOR gate: (Bubbled AND)

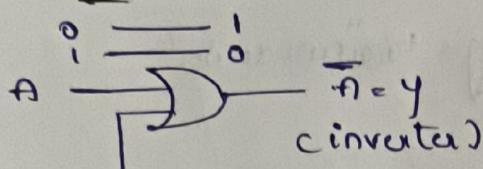
⇒ OR gate followed by NOT gate



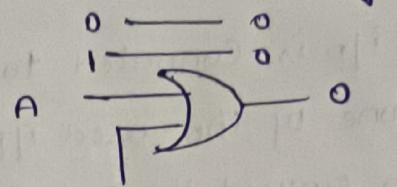
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

⇒ when both i/p is low, then o/p is high.

Enable & disable:



control (0) disable



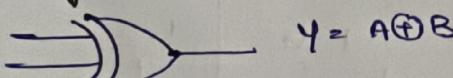
Control (1) Enable

$$(i) \bar{A} + \bar{B} = \bar{B} + \bar{A} \rightarrow \text{commutative law}$$

$$(ii) \bar{A} + \bar{B} + C \neq (\bar{A} + \bar{B}) + C$$

⇒ unused i/p in NOR gate can be connected as similar to OR gate (logic "0")

EXOR - XOR gate: (Exclusive OR gate).

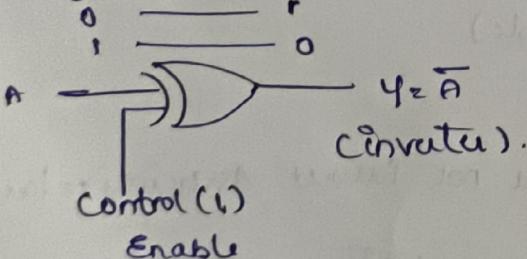


Enable & disable:



control (0)

enable



control (1)

enable

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

if $A = B$, low ("0")
 $A \neq B = \text{high ("1")}$

⇒ It is also called as controlled inverter.

Note:

$$A \oplus A = 0$$

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

$$A \oplus 0 = A \text{ (buffer)}$$

$$A \oplus 1 = \bar{A} \text{ (inverter)}$$

$$\rightarrow \text{if } A \oplus B = C$$

$$(i) A \oplus C = B$$

$$(ii) A \oplus B = C$$

$$(iii) A \oplus B \oplus C = 0$$

Since

$$A \oplus A = 0$$

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

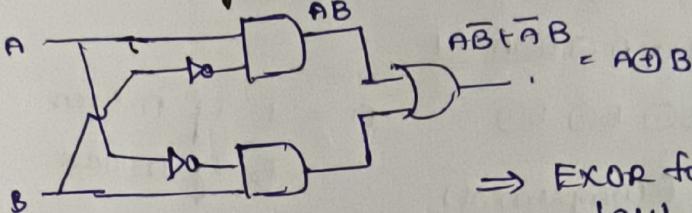
$$A \oplus A \oplus A \oplus A = 0$$

Then we say
 add no. of same ilp give same o/p and
 Even no. of same ilp give zero

$$\Rightarrow B \oplus B \oplus B \oplus \dots n = B, \text{ if } n \text{ is odd}$$

$$0, \text{ if } n \text{ is even.}$$

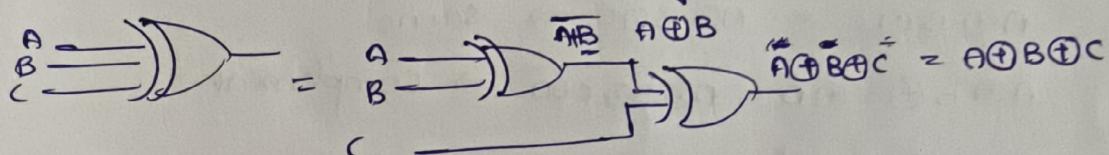
Internal diagram of EXOR gate:



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

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

\Rightarrow EXOR follows both commutative & associative law
 \Rightarrow EXOR only 2 ilp



Truth table

A	B	C	Y
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

which place in 1 present, the only odd no.

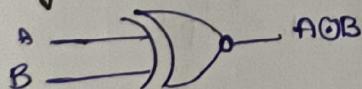
$$\bar{A}\bar{B}C + \bar{A}\bar{B}C + A\bar{B}\bar{C} + ABC$$

$$= 001 \quad 010 \quad 100 \quad 111$$

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

for reduced this expression
 to get
 $A \oplus B \oplus C$

EXNOR gate:



$$SOP : \bar{A}\bar{B} + AB$$

$$POS : (A + \bar{B})(\bar{A} + B)$$

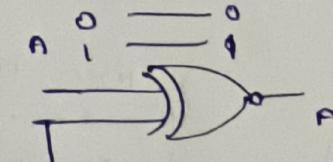
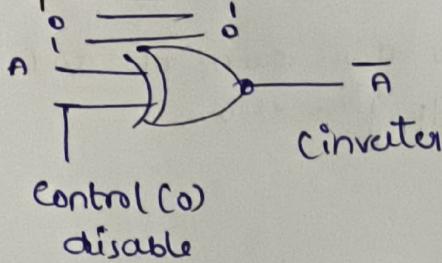
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

If $A=B$, high
 $A \neq B$, low.

$$\begin{array}{l} EXOR : \bar{A}B + A\bar{B} \\ \hline \end{array} \xleftarrow[SOP]{POS} \begin{array}{l} 11 \\ \bar{A}\bar{B} + A\bar{B} \\ (A+B)(A+\bar{B}) \end{array}$$

\Rightarrow also called as equivalent detector

Enable & disable:



control (C1)

Enable (Buffer)

Note:

$$A \oplus A = 1$$

since

$$A \oplus A = 1$$

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

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

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

$$A \oplus A \oplus A \oplus A = 1$$

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

$$B \oplus B \oplus B \oplus B \oplus \dots \quad n = 1 \text{ if } n = \text{Even}$$

$$B \text{ if } n = \text{odd}$$

ExOR & ExNOR are not equal (Complement)

one is Even or other is odd then ExOR, ExNOR become same.

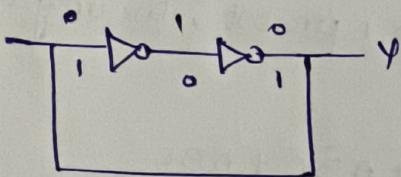
$$A \oplus B \oplus C = A \oplus B \oplus C \Rightarrow \text{Same}$$

$$A \oplus B \oplus C \oplus D = \overline{A \oplus B \oplus C \oplus D} \Rightarrow \text{Complement}$$

Problems:

NOT gate:

①



circuit shown in the figure are

(a) Buffer

(b) Astable m.v

(c) Bistable m.v

(d) Square wave generator.

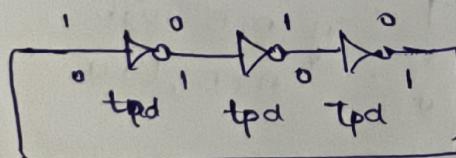
Sol: If there is no feedback,

Buffer. same i/p = same o/p.

But here, there is a feedback. o/p is stable, i/p as "1", then o/p is also "1".

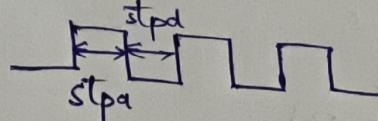
⇒ Hence it is system in Bistable multivibrator.

② ckt shown in



tpd = proportion delay.

0 = s tpd ; 1 = ̄ tpd :



⇒ It is square wave generator

⇒ Ring oscillator

⇒ clock generator

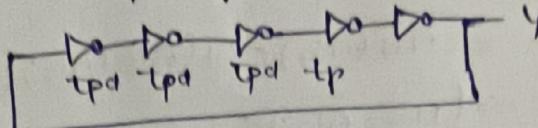
⇒ o/p is not stable, sometime 1, and 0 (astable multivibrator)

$$T = 4 tpd$$

$$T = 2 N tpd$$

No. of inverters

③ In the shown in fig, the propagation delay for each NOT gate is 100ps/sec, then frequency of square wave generator is



- (a) 100GHz (b) 1GHz (c) 100MHz (d) 10 MHz

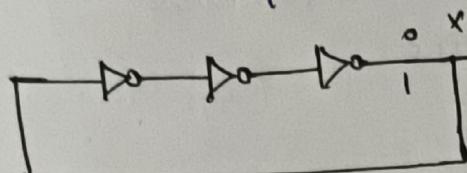
$$T = 2NT_{pd}$$

$$= 2(5) \times 100 \times 10^{-12}$$

$$= 1000 \text{ psec}$$

$$f = \frac{1}{1000 \times 10^{-12}} = 10^9 \text{ Hz}$$

④



The circuit shown in the fig. The propagation delay of each not gate is 2ns/sec. A square wave

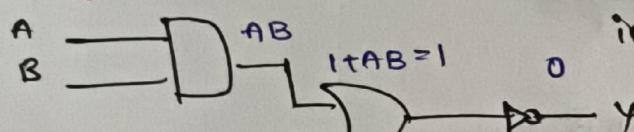
- (a) 6ns (b) 12ns (c) 14ns (d) 18ns

$$T = 2NT_{pd}$$

$$= 2 \times 3 \times 2 = 12 \text{ ns}$$

OR gate problem:

⑤



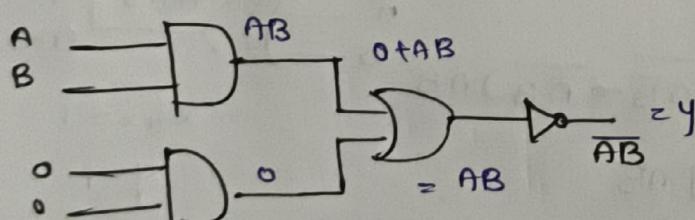
In the circuit shown in fig in TTL, AND or inverter. circuit for the given o/p in

floated
input {
for 0 b) 1 (c) AB (d) \bar{AB}

$$\begin{aligned} 0 &\rightarrow A \\ + &\rightarrow I \\ A+AB &= \\ A(I+B) &= \\ A(I) &= A \end{aligned}$$

tips:	In TTL	AND	OR
		open/float	float
In ECL	open/float	I	0
	float	0	open/float '0'

⑥

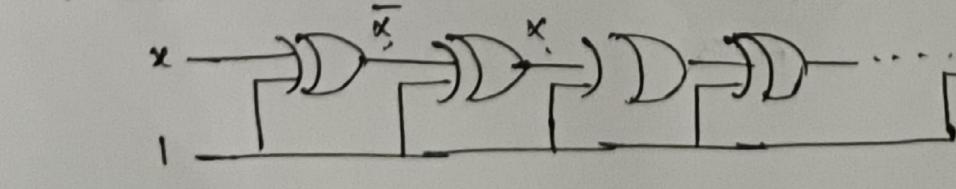


- (a) 0 (b) 1 (c) AB (d) \bar{AB}

$$\begin{aligned} x &= 1 \\ 1 &= 0 \\ 0 &= 1 \\ 1 &= 1 \end{aligned}$$

Exor problem: $(n+1)(2n+1) = 19$

⑦



$$\begin{aligned} B \oplus B \oplus B &= B \\ \text{if } n = \text{odd} & \\ \text{condition.} & \end{aligned}$$

problems on EX-NOR

$$\begin{aligned}
 & \text{⑧ } \overline{A \oplus B} C \\
 \hookrightarrow & (\overline{A \bar{B}} + \overline{B \bar{A}}) C \Rightarrow (\overline{\bar{A} \bar{B} + B \bar{A}}) \bar{C} \Rightarrow (\overline{\bar{A} \bar{B} + A B}) \bar{C} + (\overline{A B + \bar{A} \bar{B}}) C \\
 & \Rightarrow (\overline{\bar{A} \bar{B} + A B}) C + (\overline{A B + \bar{A} \bar{B}}) C \\
 \text{since: } & \overline{\bar{A} \bar{B} + A B} = \overline{\bar{A} \bar{B} B} = \overline{A \oplus B} = \overline{\bar{A} \bar{B} + A B} \\
 \Rightarrow & (\overline{\bar{A} \bar{B} + A B}) \bar{C} + (\overline{A B + \bar{A} \bar{B}}) C \\
 & = \overline{A B C} + A \overline{B} \bar{C} + \overline{A} \bar{B} C + A B C
 \end{aligned}$$

A	B	C	y	$\Rightarrow A \oplus B \oplus C$
0	0	0	1	(A) $A \oplus B \oplus C$
0	0	1	0	(B) $A \oplus B \oplus C$
0	1	0	0	(C) $\overline{A \oplus B \oplus C} = A \oplus B \oplus C$
0	1	1	1	(D) $A B + B C + A C$
1	0	0	0	$\Rightarrow \overline{A \oplus B \oplus C}$
1	0	1	1	$A \oplus B \oplus C$
1	1	0	1	
1	1	1	0	

$$\begin{aligned}
 & \text{⑩ } \overline{A \oplus B} = A \oplus B \\
 & \text{put } x = \overline{A}; y = B \\
 & x \oplus y = \overline{x}y + x\overline{y} \\
 & = A B + \overline{A} \overline{B} \\
 & = A \oplus B
 \end{aligned}$$

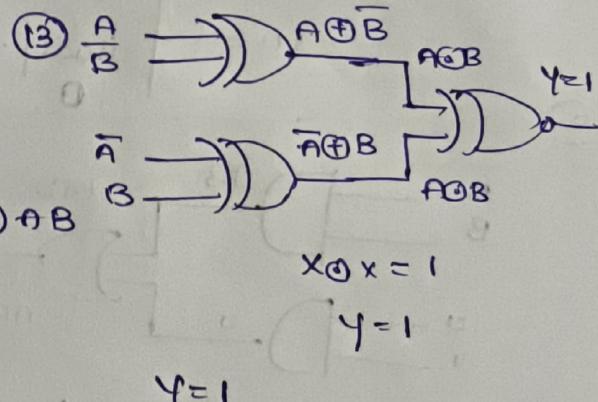
$$\begin{aligned}
 & \text{⑪ } \overline{A \oplus \bar{B}} = \\
 & \text{put } x = A; y = \overline{B} \\
 & x \oplus y = \overline{x}y + x\overline{y} \\
 & = \overline{A} \overline{B} + A B \\
 & = A \oplus B
 \end{aligned}$$

$$\begin{aligned}
 & \text{⑫ } A \oplus B \oplus A B \\
 & (A \bar{B} + \bar{A} B) \oplus A B \\
 & (A \bar{B} + \bar{A} B) \overline{A B} + (\overline{A \bar{B} + \bar{A} B}) A B \\
 & \Rightarrow A \bar{B} (\bar{A} + \bar{B}) + \bar{A} B (\bar{A} + \bar{B}) + (\overline{A \bar{B} + \bar{A} B}) A B \\
 & = A \bar{B} + \bar{A} B + [(\bar{A} + B)(A + \bar{B})] A B \\
 & = A \bar{B} + \bar{A} B + A B
 \end{aligned}$$

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

$$\Rightarrow (\bar{A} + \bar{B})(A + B)$$

$$= A + B.$$

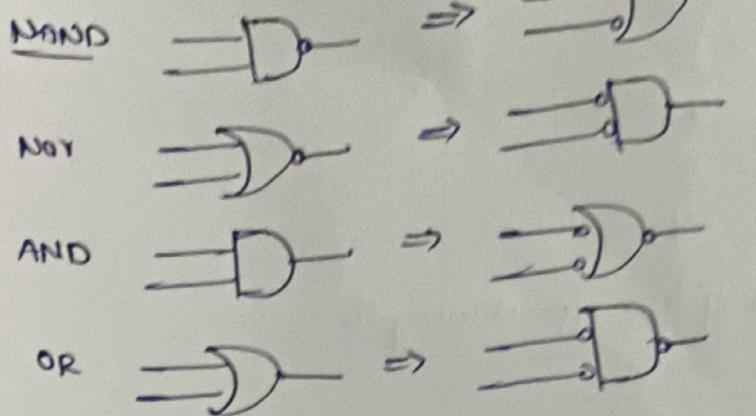


$$x \oplus x = 1$$

$$Y = 1$$

$$Y = 1$$

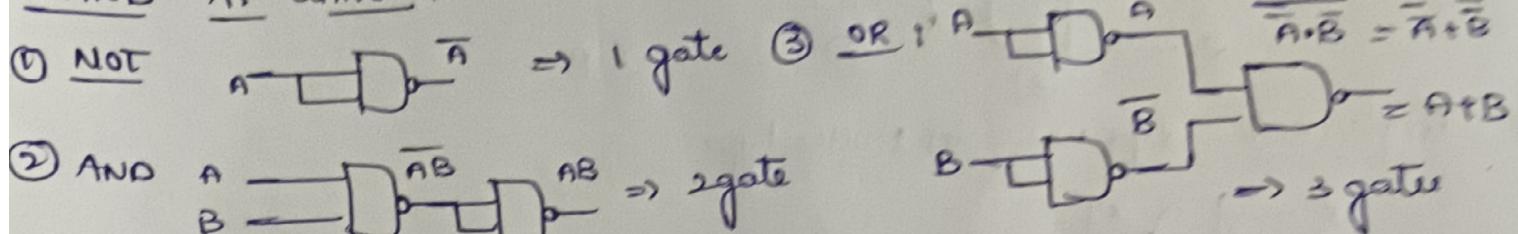
Symbols:



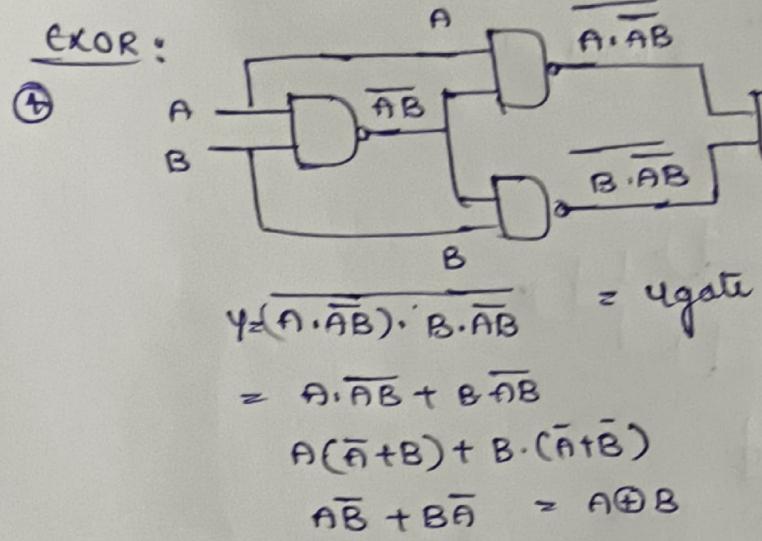
Bubbled or

Bubbled NAND

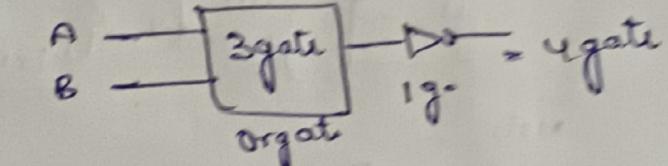
NAND as universal



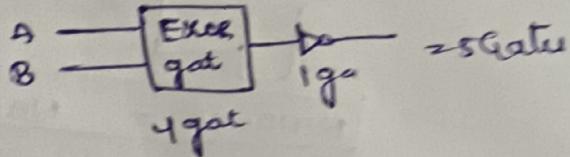
EXOR:



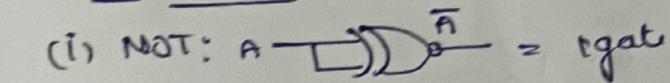
NOR:



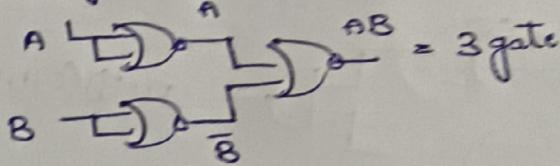
EXNOR:



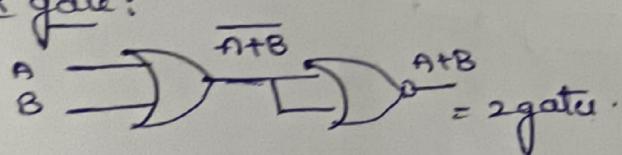
Nor as universal:



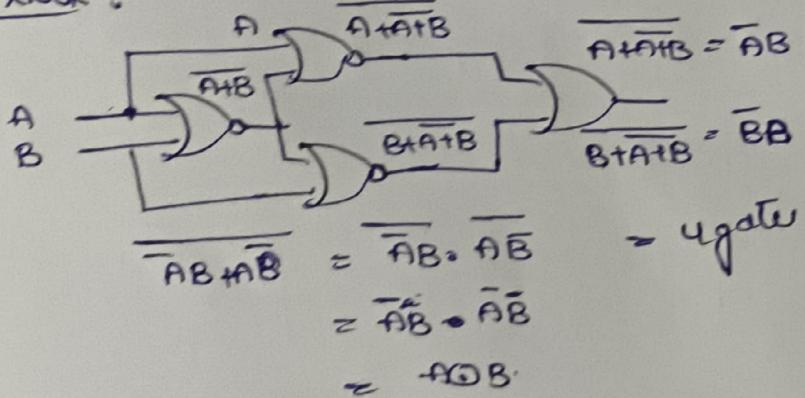
AND gate:



OR gate:



EXNOR:

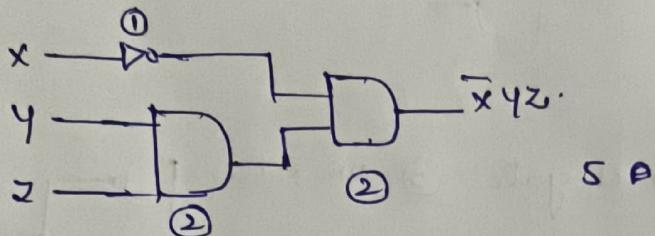


EXOR: 5 gates

NAND = 4 gates

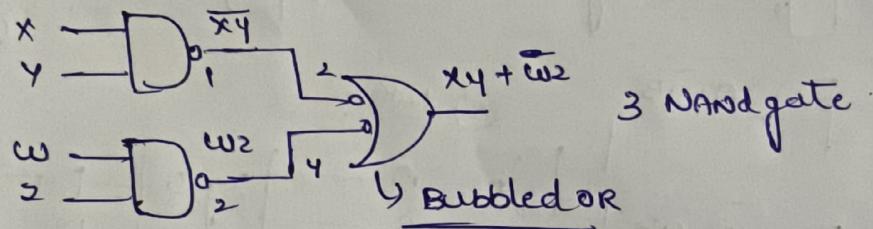
logic gate	No. of NAND	No. of NOR
NOT	1	1
ANP	2	3
OR	3	2
NOR	4	4
EXOR	4	5
EXNOR	5	4

(14) To implement $\bar{x}yz$. The min no. of two i/p NAND gate



5 p

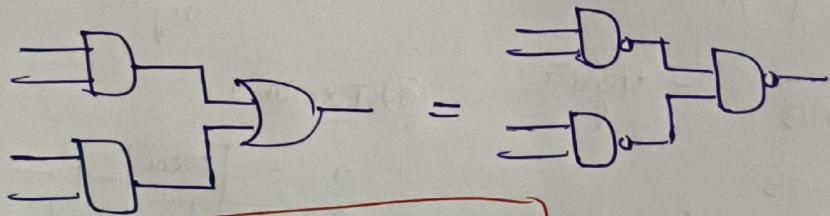
(15) Implement $xy + wz$ in NAND type



3 NAND gate

Bubbled OR

Note:



NAND-OR = NAND-NAND.

- SOP form, only NAND gate alone
- POS form, only NOR gate alone

(16) A
B
C
D

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

OR-AND = NOR-NOR

