Digital Eletronics

Minimization & logical exp. < Boolean Algebra / K map

Logic gates

Combinational ckts → Digital logic Design, Moris Mono

Sequential ckts

ADC & DAC → ADC & DAC, Tomb & Schilling

Logic families

Memories.

Number system, Data interpretation

Boolean & K-Map

1) Boolean
2) K-map

Boolean algebra is used when number of variables are less & output may be 0 or 1. K-map is used when no. of variables are 2,3,4,5 and output may be 0, 1 or X (don't care).

Boolean Algebra (1854)

Switches < on, 1 / off, 0

1) Not ($\bar{A}$ or A')

$\bar{\bar{A}} \rightarrow A$

$\overline{\overline{ABC}} \rightarrow ABC$

2) AND

$A.A = A$

$A.1 = A$

$A.0 = 0$

$A.\bar{A} = 0$

3) OR

$A + A = A$

$A + 0 = A$

$A + 1 = 1$

$A + \bar{A} = 1$

Q 3

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

$AB + A\bar{B}(C+\bar{C})$

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

A

Advantage of minimization

1. Number of logic gates will decrease
2. speed of operation increases
3. Power dissipation is less.
4. circuit complexity reduced.
5. fan-in may be reduces.

Ques

Transposition theorem

(i) $\boxed{(A+B)(A+C) = A \cdot A + BC = A + BC}$

$A + AC + AB + BC$

$A[1 + C + B] + BC$

$A[1] + BC$

$A + BC$

(ii) $\boxed{[(A+B)(A+\bar{B})][(\bar{A}+B)(\bar{A}+\bar{B})]}$

$[A+0][\bar{A}+0]$

0

(iii) $\boxed{[(P+Q+R)(P+\bar{Q}+R)](P+\bar{Q}+\bar{R})}$

$\cancel{P.A\ B(Q+R)(\bar{Q}+R)}$

$(\underbrace{P+R}_{A}+Q)(\underline{P+R}+\bar{Q})(P+\bar{Q}+\bar{R})$

$[P+R][P+\bar{Q}+\bar{R}]$

$P + R\cdot\bar{Q}$

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

$① + ②.③ = (①+②)(①+③)$

Ques

$A + \bar{A}B$

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

$A+B$

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

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

$(A+\bar{B})$

$\bar{A} + AB$

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

$(\bar{A}+B)$

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

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

Ques $AB + \bar{A}B + \bar{A}\bar{B}$

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

$\bar{A} + AB$

$\bar{A} + B$

(II) $A \cdot B + \bar{A}\bar{B} + A\bar{B}$

$A.B + \bar{B}$

$A + \bar{B}$

(III) $AB\bar{C} + ABC + \bar{A}BC$

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

$= AB + \bar{A}BC$

$= B[A + \bar{A}C]$

$= B[A+C]$

(IV) $ABC + \bar{A}BC + A\bar{B}C + AB\bar{C}$

BC, AC, AB

$BC + AC + AB$

(V) $AB + \bar{A}C + BC$ → Redundent term

$AB + \bar{A}C + BC(A+\bar{A})$

$AB + \bar{A}C + ABC + \bar{A}BC$

AB, $\bar{A}C$

$AB(1+C) + \bar{A}C[1+B]$

$AB + \bar{A}C$

CONCENCUS THEOREM

$AB + \bar{A}C + BC$ or Rudendency theorem

AB
BC
CA

(I) Three variable

(II) Twice each variable used twice.

(III) only one variable is complimented/uncomplimented i.e. A

related terms of A

$AB + \bar{A}C$

Ex 5

(i) $AB + B\bar{C} + AC = B\bar{C} + AC$

(ii) $AB + \bar{B}C + AC = AB + \bar{B}C$

(iii) $A\bar{C} + BC + AC = A + BC$]× not applicable is this case

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

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

Application for POS

(vi) $\bar{A}\bar{B} + A\bar{C} + \bar{B}\bar{C}$ /uncomplimented

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

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

(viii) $(\bar{A}+\bar{B})(\bar{B}+\bar{C})(A+\bar{C})$

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

De-Morgan theorems

(i) $\overline{A+B+C} = \bar{A}.\bar{B}.\bar{C}$

(ii) $\overline{ABC} = \bar{A}+\bar{B}+\bar{C}$

Boolean

(i) minimisation
(ii) SOP — minimal, canonical
(iii) POS.
(iv) Dual.
(v) Compliment.
(vi) Truth table.
(vii) Venn diagram.
(viii) Switching ckt.
(ix) Statements.

Operator importance

1) [ ] ② NOT ③ AND ④ OR

Scanned with CamScanner

Ques ① 6 $\underset{A}{XY} + \underset{\overline{A}}{\overline{XY}}\ \underset{B}{WZ} = A + \overline{A}B = A+B$

$XY + WZ$

⑪ Let $F(A,B) = \overline{A} + B$, then the value of

$f[f[(x+y, y), z]]$

$\overline{x+y} + y$

$= \overline{\overline{x}}.\overline{y} + \overline{\overline{y}}$

$= y + \overline{x}$

Now, $\overline{\overline{x}+y} + z$

$x.\overline{y} + z$

⑪ Let $X * Y$ is $\overline{X}+Y$, and $Z = X * Y$, then the value of $Z * X$?

Sol^n :- $X * Y * X$

$\overline{X} + Y * X$

$\overline{\overline{X}+Y+\overline{X}+Y} + X$

$\overline{X} + Y + X.\overline{Y} + X$

$\overline{X} + Y + X(1+\overline{Y})$

$(X + \overline{X} + Y) = 1 + Y = 1$

$Z * X$

$= \overline{Z} + X$

$= \overline{\overline{X+Y}} + X$

$= X.\overline{Y} + X$

$= X$

SOP

$A\overline{B}C + AB\overline{C} + ABC$

minterm (or) implicant

In SOP form each product term is called min term or implicant

SOP are used when output of logic ckt is 1

Scanned with CamScanner

Ex 7 Write min term for

(i) 5 → 101 (ii) 9 → 1001

$A\bar{B}C$ $A\bar{B}\bar{C}D$

For given table minimize

	A	B	Y
$\bar{A}\bar{B}$	0	0	1
	0	1	0
$A\bar{B}$	1	0	1
	1	1	0

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

$$Y(A,B) = \Sigma m(0,2)$$

(iv) $Y(A,B) = \Sigma m(1,2,3)$

01 10 11

$\bar{A}B$ $A\bar{B}$ AB

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

$A + \bar{A}B$ → minimal → $A+B$

→ Canonical → $A(B+\bar{B}) + \bar{A}B$

Standard form $AB + A\bar{B} + \bar{A}B$

In canonical or standard SOP form each minterm will contain all variables.

Ques No of min term present Canonical expression of

$A + \bar{B}C$

(a) 5
(b) 6
(c) 7
(d) 3

$$A.(B+\bar{B})(C+\bar{C}) + \bar{B}C(A+\bar{A})$$

$$A[BC+B\bar{C}+\bar{B}C+\bar{B}\bar{C}] + \bar{B}CA + \bar{B}C\bar{A}$$

$$ABC + AB\bar{C} + \underline{A\bar{B}C} + A\bar{B}\bar{C} + \underline{\bar{B}CA} + \bar{B}C\bar{A}$$

same

POS 8

Ex $\underbrace{(A+B+\overline{C})}_{\text{max term}}(\overline{A}+B+\overline{C})\ (A+B+C)$

n POS form each term is max term. It is used when output is logic 0.

$5 \rightarrow 1\ 0\ 1$

$\overline{A}+B+\overline{C}$

AB	$A+B$	$A\overline{A}$	$B\cdot 1$
$\overline{A}B$	$\overline{A}+B$	$B\overline{B}$	$A\cdot A$
$A\overline{B}$	$A+\overline{B}$	$A\cdot 1$	$B+1$
$\overline{A}\,\overline{B}$	$\overline{A}+\overline{B}$	$B\cdot B$	$A+1$

for truth table

A	B	Y
0	0	1
0	1	0 ✓
1	0	1
1	1	0 ✓

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

$= \overline{B}$ product of maxterms

$Y(A,B) = \prod M(1,3) = \overline{B}$

$Y(A,B) = \sum m(0,2) = \overline{B}$

$\Rightarrow \boxed{\sum m(0,2) = \prod M(1,3)}$

2^3

→ $f(A,B,C) = \sum m(0,1,5,7) = \prod M(2,3,4,6)$

→ When n variables, max^m possible minterms or max terms are 2^n.

→ With n variables max^m possible logical expressions are 2^{2^n}.

9
$n = 2 \rightarrow 2^{2^2} = 2^4 = 16$

$n = 3 \rightarrow 2^{2^3} = 2^8 = 256$

$n = 4 \rightarrow 2^{2^4} = 2^{16}$

$= 2^6 \times 2^{10} = 64K$

$64 \times (1024) = 65536$

Dual / DUALITY

1) $+ \xleftrightarrow{\text{Replace or}} \cdot$

2) $1 \longleftrightarrow 0$

$A + 1 = 1 \xleftrightarrow{\text{Dual}} A \cdot 0 = 0$

AND $\xleftrightarrow{\text{Dual}}$ OR

NAND $\xleftrightarrow{\text{Dual}}$ NOR

EXOR $\xleftrightarrow{\text{Dual}}$ EXNOR

NOT $\longleftrightarrow$ NOT

+ve logic

logic 0 → 0V

logic 1 → +5V

-ve logic

logic 0 → 5V

logic 1 → 0V

higher voltage is logic 1 is +ve logic, ~~lower~~ higher voltage is logic 0 then -ve logic.

→ Dual is used to convert +ve logic into ~~+e~~ -ve logic and -ve logic to +ve logic.

→ And gate

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

OR gate

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

→ -ve logic OR gate

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

$\boxed{\text{+ve logic And gate = -ve logic OR gate}}$

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

INTER CHANGE

(I) And ↔ OR

$\cdot \longleftrightarrow +$

(II) $1 \longleftrightarrow 0$

(III) Keep variables as it is

$A \cdot 1 = A$

$A + 0 = A$

$A + 1 = 1$

$A \cdot 0 = 0$

Dual is $(A+B+\overline{C}).(\overline{A}+B+C).(A+B+C)$

→ for any logical expression if two time dual is used result in same expression

<u>Self Dual</u>

$AB+BC+AC$

$\downarrow$ Dual

$(A+B).(B+C).(A+C)$

$(B+AC)(A+C)$

$AB+BC+AC$

In self dual expressions if one time dual is used, result in same expression

No. of max^m Self dual expressions

$n \longrightarrow$ Self dual expressions $\longrightarrow 2^{2^{(n-1)}}$

→ $A \xleftrightarrow{Dual} A$

→ $\overline{A} \xleftrightarrow{Dual} \overline{A}$

→ $AB+A\overline{B}C + 1$

Dual

$(A+B)(A+\overline{B}C).0$

0

<u>Complement</u>

$Y = ABC + \overline{A}BC + AB\overline{C}$

(i) AND ⟷ OR

(ii) 1 ⟷ 0

(iii) Complement each variable also ✓

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

→ No Self Complement logic.

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1 ←
1	0	0	0
1	0	1	1 ←
1	1	0	1 ←
1	1	1	1 ←

$Y = \cancel{\bar{A}BC + A\bar{B}C + AB\bar{C} + ABC}$

$= \cancel{\bar{A}BC + A\bar{B}C + AB\bar{C} + ABC}$

$Y = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$

$= BC + AC + AB$

LOGIC GATES

→ Basic building blocks

NOT, AND, OR } Basic gates

NAND, NOR } Universal gates

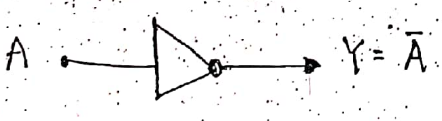
EXOR, EXNOR } Airthmatic ckts, comparators, parity generation/checkers, Code converters.

Degenerative form: If a two level logic gate system final o/p is expressed with a single logic gate Then the two level logic gate system ~~can be~~ is known as Degenerative form.

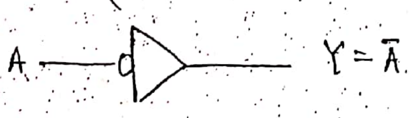
Non Degenerate form

AND - OR	OR - AND
NAND - NAND	NOR - NOR
NOR - OR	NAND - AND
OR - NAND	AND - NOR

Not

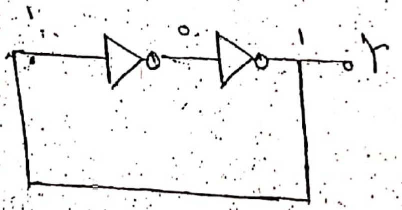


(or)



A	Y
0	1
0	1

14. The ckt in figure is



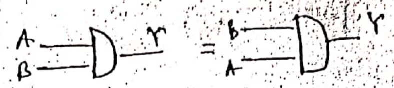
(a) Buffer
(b) Astable MV
(c) Bistable MV
(d) Square wave Generator

Scanned with CamScanner

AND gate 16

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

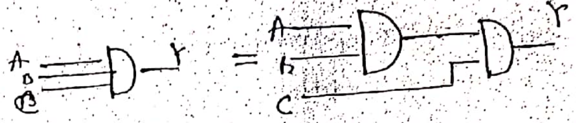
Series switches



→ Commutative }; $AB = BA$

→ ~~Distributive~~ }; $ABC = (AB)C$

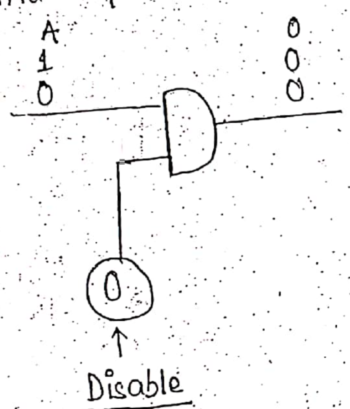
→ Associative } $= A(BC)$

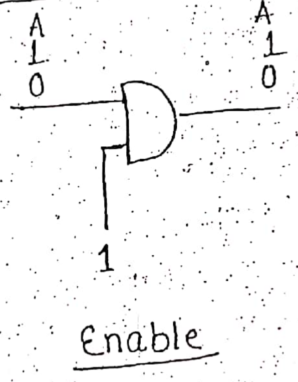


→ Distributive → $A(B+C) = AB + AC$

And Gate follow Commutative, Associative, & distributive law

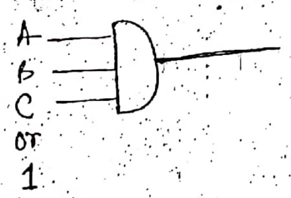
→



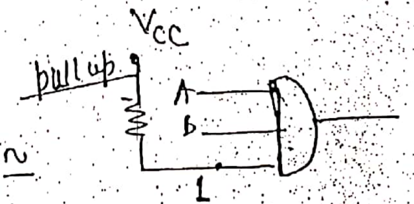


for And gate logic 0 is disable and 1 is enable

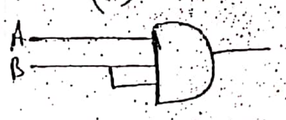
→



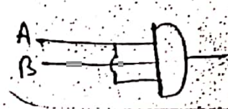
≈



(or)



(or)



→ In TTL Logic family open or floating input will acces logic 1 whereas in ECL logic gates open or floating input act as logic 0 for TTL

In AND gate unused input can be

(i) Connected to logic 1 (or) pull up

(ii) Connect to one of used input.

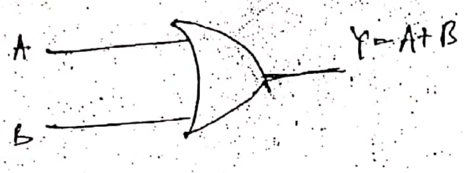
(iii) If TTL, then unused input can be open or float

→ The best way of connecting unused input is (i)

~~ADD~~ OR

OR Gate

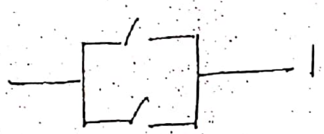
(i)



(ii)

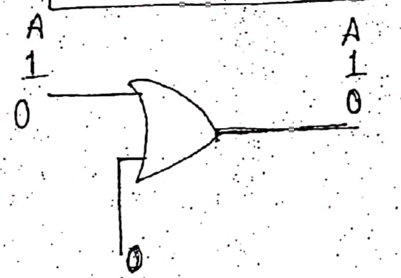
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

(iii)

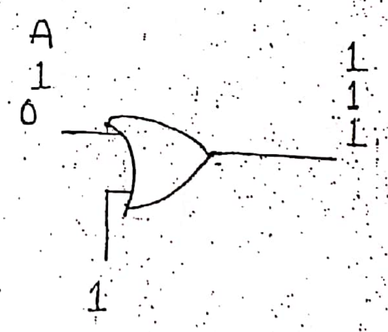


parallel switches

(iv)



enable



disable

(v) - Commutative law

$A+B = B+A$

Association

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

OR gate follow both commutative and associative law

(vi)

Best (i) Unused can be connected to logic 0

(ii) Connect to ground or pull down

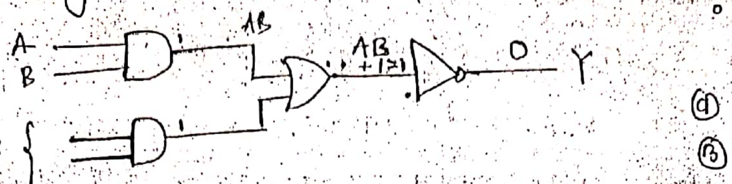
(iii) Connect to one of used inputs

(iv) In case of ECL it can be

Q: The ckt shown in figure is

TTL AND - OR - INVERTER

for the given input output Y is

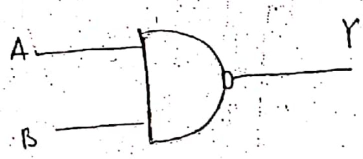


inputs are floating

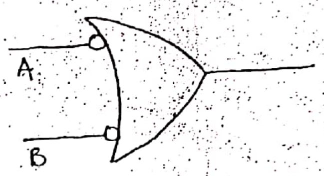
(a) AB
(b) $\overline{AB}$
(c) 1
(d) 0

Date
7/09/10

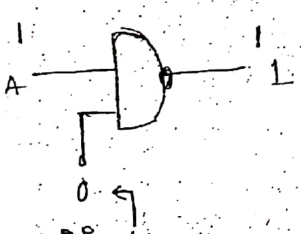
NAND GATE

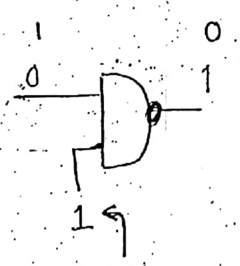


$Y = \overline{AB} = \bar{A} + \bar{B}$

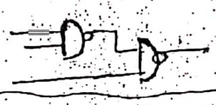


A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

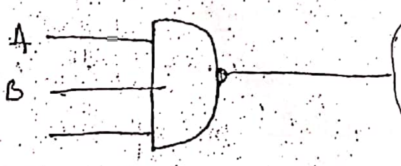




) Nand Gate follow commutative law.
NAND gate does not follow Associative law.

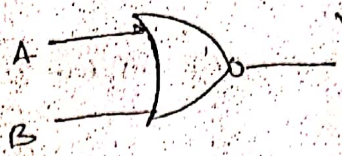


)



Unused input in NAND gate can be connected similar to unused inputs in AND gate.

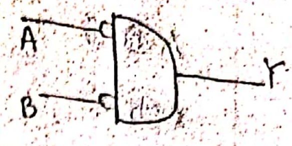
NOR GATE

(i)  

$Y = \overline{A+B}$

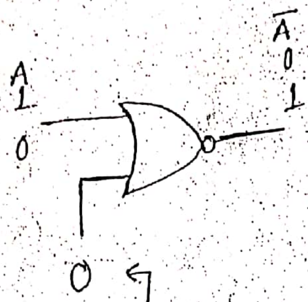
$= \bar{A}.\bar{B}$

(or)

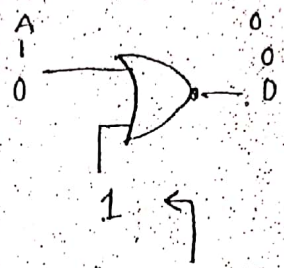


(ii) Truth table

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

(iii) 

Enable



Disable

(iv) NOR Gate follows Commutative law but not associative law

(v) $\boxed{\text{Unused input of NOR can be connected similar to OR gate.}}$

EX-OR Gate

Exclusive OR Gate

$Y = A \oplus B$

$x \oplus x = 0$

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

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

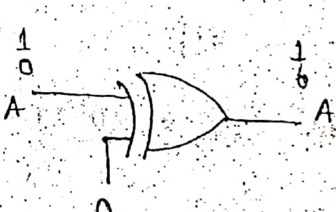
$x \oplus 0 = 0$

$\left.\begin{array}{l}\bar{x} \oplus y \\ x \oplus \bar{y}\end{array}\right\} \overline{(x \oplus y)}$

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

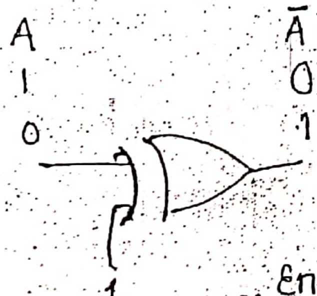
$Y = (A+B)(\bar{A}+\bar{B})$ .... POS

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

(iii) 

∴ Enable

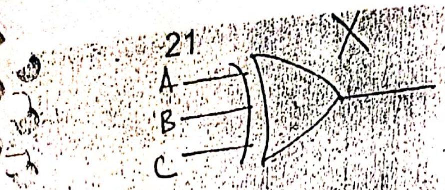
(Buffer)



∴ Enable

" Inverter "

21

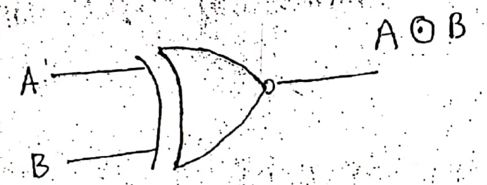


A	B	C	$Y = A \oplus B \oplus C$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$Y = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$ ...SOP

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

EX-NOR GATE



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

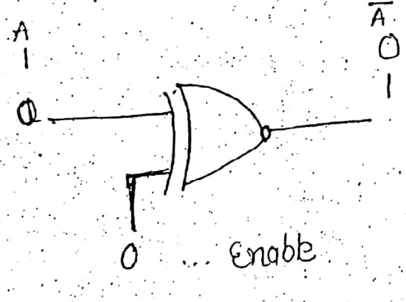
$Y = AB + \bar{A}\bar{B}$ ...SOP

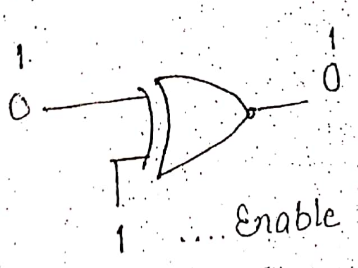
$Y = (\bar{A}+B)(A+\bar{B})$

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

(i) If $A = B$, output $= 1$ ....coincidence logic ckt
.....equilance ~~ckt~~ gate.

(ii) If $A \neq B$, output $= 0$





Inverter

→ $\boxed{\begin{array}{l} A \odot A = 1 \\ A \odot \bar{A} = 0 \\ A \odot 0 = \bar{A} \\ A \odot 1 = A \end{array}}$

(Buffer

→ $A \odot A = 1$

$A \odot 1 = A$

22

$A \oplus A = 0$ | $A \odot A = 1$

$A \oplus A \oplus A = A$ | $A \odot A \odot A = A$

$A \oplus A \oplus A \oplus A = 0$ | $A \odot A \odot A \odot A = 1$

**

$$\boxed{\begin{array}{l} \rightarrow A \oplus B = \overline{A \odot B} \\ \rightarrow A \oplus B \oplus C = A \odot B \odot C \end{array}}$$ ... Same in three variables.

$\rightarrow A \oplus B \oplus C \oplus D = \overline{A \odot B \odot C \odot D}$

- EX-NOR GATE acts as even number of 1 detector when number of input variables are even. Similary it act as odd number of 1 detector when number of input variable are odd.

les $\rightarrow$ $A \oplus \overline{B} = A \odot B$, $\rightarrow \overline{A} \oplus B = A \odot B$

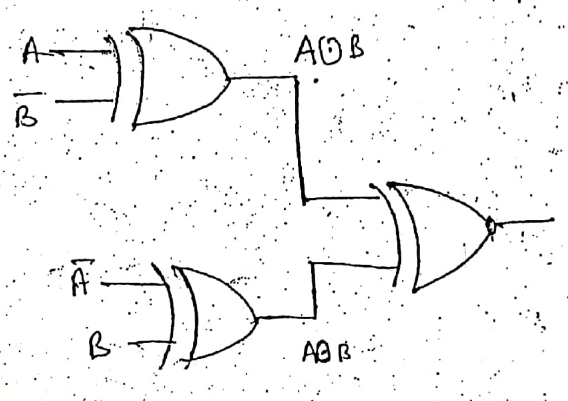
$\downarrow \quad \downarrow$

$x \quad y$ $\rightarrow A \odot \overline{B} = A \oplus B$

$\overline{x}y + x\overline{y}$ $\rightarrow \overline{A} \odot B = A \oplus B$

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

ues.



$A \odot B \odot A \odot B = 1$

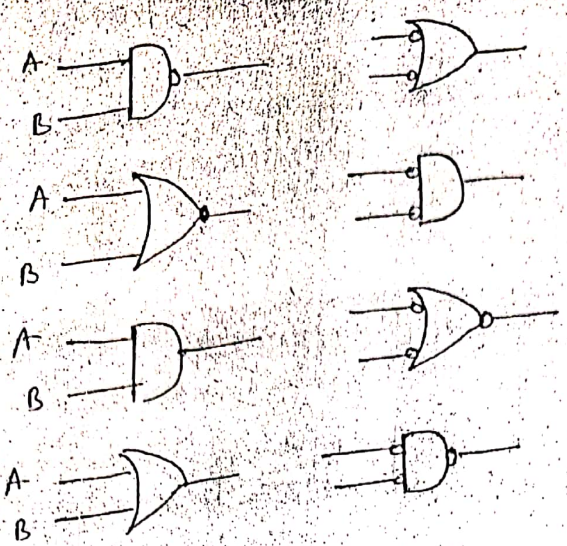
(a) 0

(b) 1 ✓

(c) $A \oplus B$

(d) $A \odot B$

Alternative Symbols



NAND $\xleftrightarrow{\text{Bubbled}}$ OR

NOR $\xleftrightarrow{\text{Bubbled}}$ AND

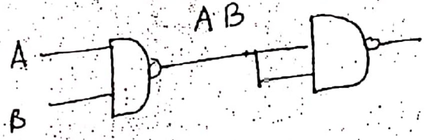
NAND as universal

Not

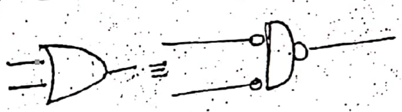
$\overline{AA} = \bar{A}$

AND

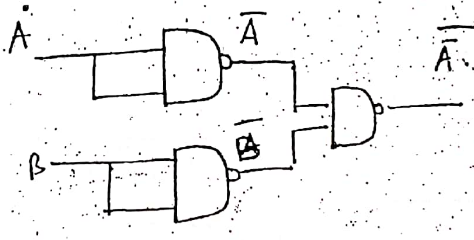
$\overline{\overline{AB}} = AB$



OR

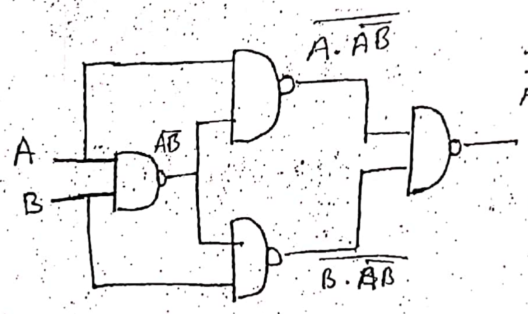


$\Rightarrow$



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

Ex OR Gate



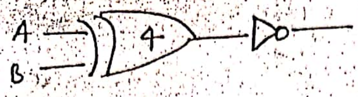
$\overline{\overline{A\cdot\overline{AB}}\cdot\overline{B\cdot\overline{AB}}} = \overline{\overline{A\cdot\overline{AB}}} + \overline{\overline{B\cdot\overline{AB}}}$

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

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

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

Ex NOR 24

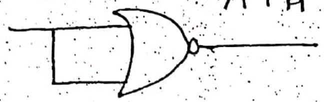


NOR

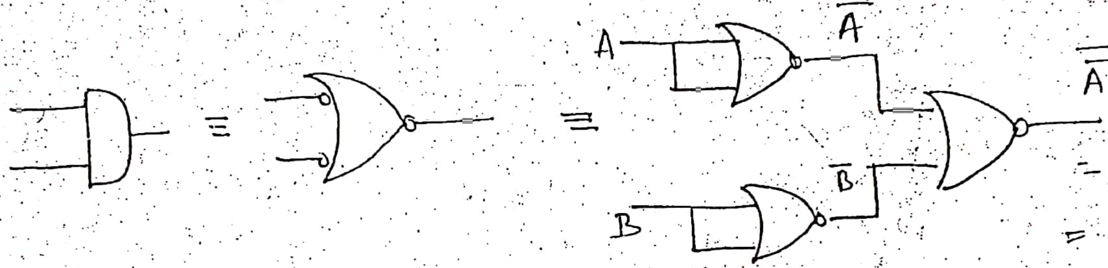
NOR is universal gate

1) NOT

$\overline{A+A} = \bar{A}$ — ①



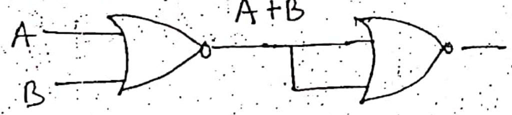
2) AND



$= \cancel{AB}$

$= A.B$

3) OR

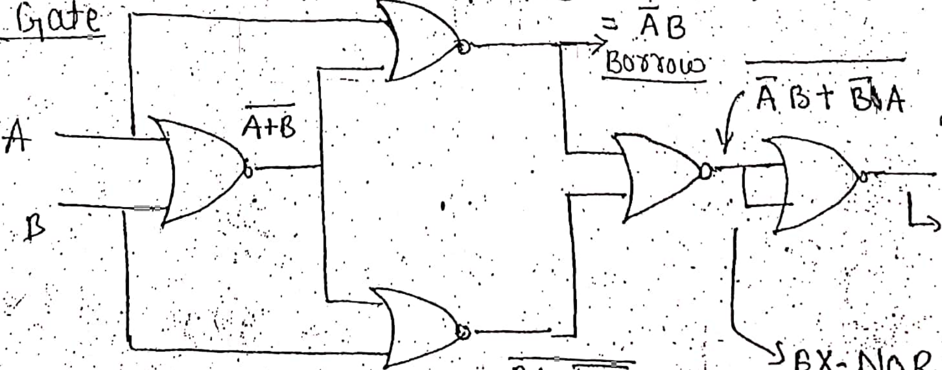


4) Ex OR Gate

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

$= \bar{A}B$

Borrow



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

↳ EX-OR ⑤

EX-NOR ④

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

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

$\bar{B}A$

5) NAND