unit-2 Boolean algebra and Theorems

postulates	a	b
postulate-2	$A+0=A$	$A\cdot 1=A$
postulate-3 (commutative law)	$A+B=B+A$	$AB=BA$
postulate-4 (Distributive law)	$A\cdot(B+C)=A\cdot B+A\cdot C$	$A+(B\cdot C)=(A+B)\cdot(A+C)$
postulate-5	$A+\bar{A}=1$	$A\cdot\bar{A}=0$
Theorems	(a)	(b)
theorem 1	$A+A=A$	$A\cdot A=A$
Theorem 2	$A+1=1$	$A\cdot 0=0$
Theorem 3	$\bar{\bar{A}}=A$	
Theorem 4	$A+AB=A$	$A(A+B)=A$
Theorem 5	$A+\bar{A}B=A+B$	$A(\bar{A}+B)=AB$
Theorem 6	$A+(B+C)=(A+B)+C$	$A(BC)=(AB)C$

denotion of Complement form is called literal

De-morgan's laws (or) Theorems

1) $\overline{AB}=\bar{A}+\bar{B}$

The Complement of the product is equal's to Sum of the Complements

Truth Table :-

A	B	$\bar{A}$	$\bar{B}$	AB	$\overline{AB}$	$\bar{A}+\bar{B}$
0	0	1	1	0	1	1
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	0	0

(ii) $\overline{A+B} = \bar{A}\cdot\bar{B}$ Complement of a Sum is equal to product of the Complements

A	B	$\bar{A}$	$\bar{B}$	$A+B$	$\overline{A+B}$	$\bar{A}\cdot\bar{B}$
0	0		1	0	1 one input is high then high	1
0	1	1	0	1	0 OR gate	0 And gate
1	0	0	1	1	0	0 one input is low then low
1	1	0	0	1	0	0

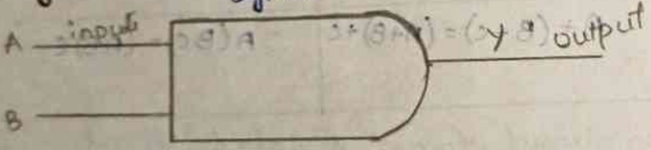
Logic gates = Basic element of digital System

→ AND, OR, NOT

→ NAND, NOR

→ Ex-OR, Ex-NOR

AND gate Symbol

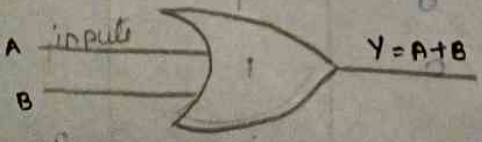


Boolean expression $Y = A\cdot B$

Truth table :- one input is low then the output is low

A	B	$Y = A\cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

OR gate



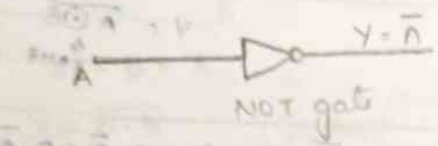
Boolean expression $= Y = A+B$

Truth table :- one input is high then the output is high

A	B	$Y = A+B$
0	0	0
0	1	1
1	0	1
1	1	1

<u>NOT</u> <u>Gate</u>

Symbol :-

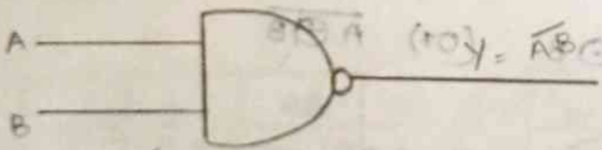


Truth table :-

A	$Y = \overline{A}$
0	1
1	0

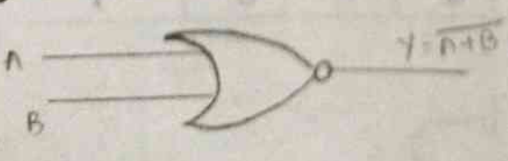
<u>NAND gate</u> NAND = AND + NOT

If any one input is low then output is high



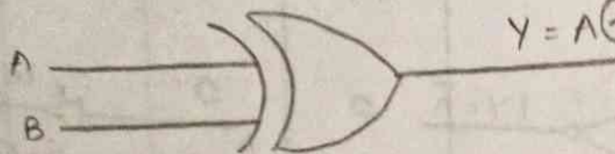
A	B	$Y = \overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

<u>NOR gate</u> NOR = OR + NOT



A	B	$Y = \overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Exclusive-OR gate



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

(or)

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

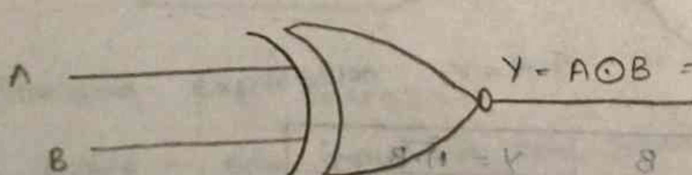
X-NOR

different inputs high Same inputs are low

A	B	$\overline{A}$	$\overline{B}$	$Y = A\cdot\overline{B} + B\cdot\overline{A}$
0	0	1	1	0 + 0 = 0
0	1	1	0	0 + 1 = 1
1	0	0	1	1 + 0 = 1
1	1	0	0	0 + 0 = 0

Exclusive - NOR gate

$Y = A \odot B$ (or) $\overline{A \oplus B}$



$Y = A \odot B = A\cdot B + \overline{A}\cdot\overline{B}$

Same inputs ay the high lekop the low

A	B	$\overline{A}$	$\overline{B}$	$A\cdot B$	$\overline{A}\cdot\overline{B}$	$Y = A\cdot B + \overline{A}\cdot\overline{B}$
0	0	1	1	0	1	1
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	1	0	0	1	0	1

Realization of Basic gates using universal gates

using NAND gate (NAND gate logic we chesi migilinavi 2 ravali)

NOT :

NAND Truth table

A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

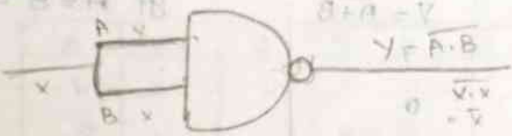
A not gate can be made from a NAND gate by connecting all the inputs together and creating a single common input for a two input NAND gate

if $A = B = X$

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

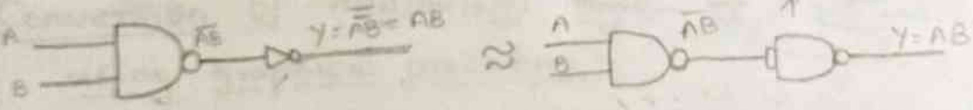
$= \overline{X \cdot X}$

$= \bar{X}$



[fix one input the input gate kane & result vasturdhi]

AND function using NAND = AND + NOT



A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

≈

A	B	$\overline{AB}$	$\overline{\overline{AB}}$
0	0	1	0
0	1	1	0
1	0	1	0
1	1	0	1

OR function :

$Y = A + B$

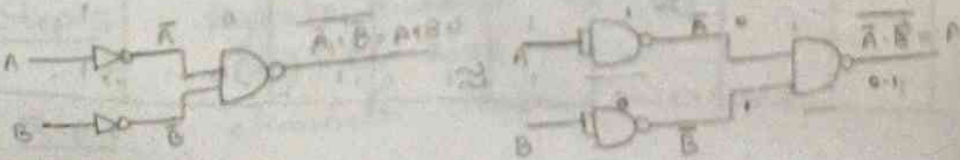
$= \overline{\overline{A + B}}$

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

[Demorgan's Law]

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

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



Truth table :-

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

A	B	$\bar{A}$	$\bar{B}$	$\overline{\bar{A}\cdot\bar{B}} = A+B$
0	0	1	1	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	1

Realization of basic gates using NOR

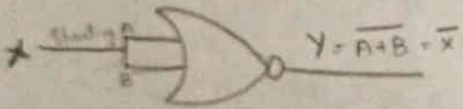
Not function :-

A	B	$Y = \overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

If $A = B = x$ then $Y = \overline{A+B}$

$= \overline{x+x}$

$= \bar{x}$



OHALU OR gate Kanuka A+B
Not undh Kanuka
$\overline{A+B}$

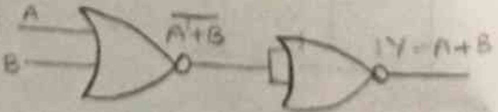
OR function

NOR $Y = \overline{A+B}$

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

$= A+B$

≈



A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

≈

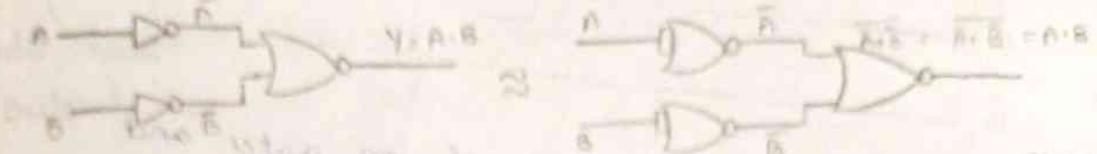
A	B	A+B	$\overline{A+B}$	$Y = \overline{\overline{A+B}} = A+B$
0	0	0	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	0	1

AND function

And gate Boolean expression

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

[from De morgans Law $\overline{A+B} = \overline{A} \cdot \overline{B}$]



Truth table

A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

≈

A	B	$\overline{A}$	$\overline{B}$	$\overline{A}+\overline{B}$	$Y = \overline{\overline{A}+\overline{B}} = AB$
0	0	1	1	1	0
0	1	1	0	1	0
1	0	0	1	1	0
1	1	0	0	0	1

Conversion of AND/OR/NOT logic to NAND/NOR using graphical procedure :-

Steps :-

Step-1 :- Draw and or AND/OR/NOT Logic

Step-2 :- If NAND has been choosen add bubbles on the output of each AND gate and bubbles on input side of OR gate

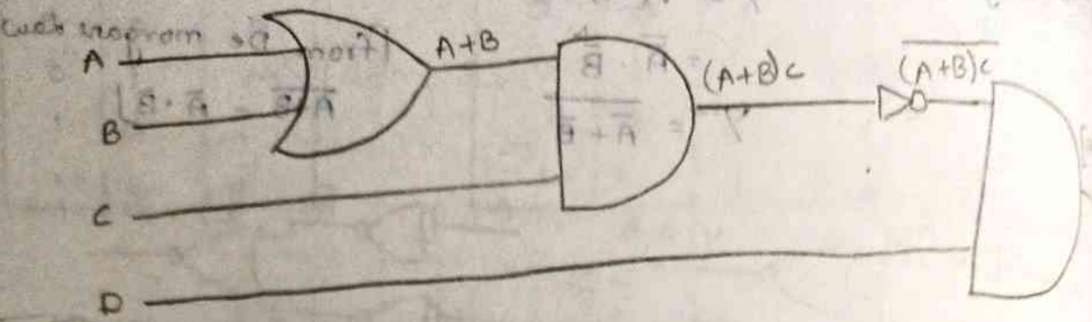
Step-3 :- If NOR has been choosen add bubbles and bubble on on output of each OR gate and bubble on input of each AND gate

Step-4 :- Add or substract an invertor (NOT gate) on each line that received a bubble in Step2 or step-3.

Step-5 :- Replace bubbled OR by NAND and bubbled AND by NOR

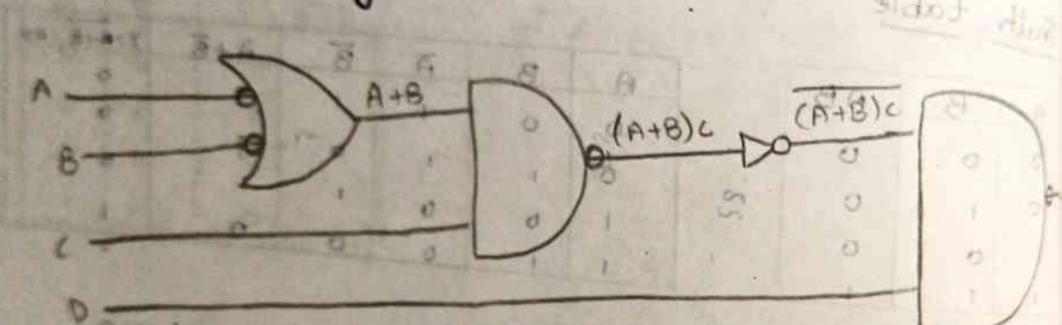
Step-6 :- eliminate double inversions

Expression :- $\left[\overline{(A+B)C}\right]D$
Bracket

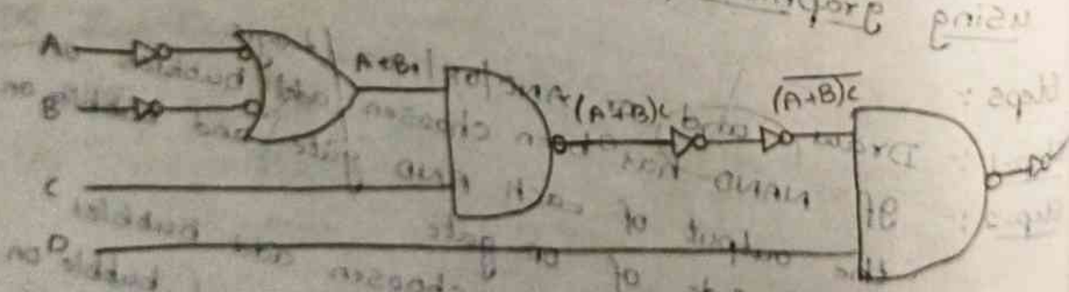


NAND

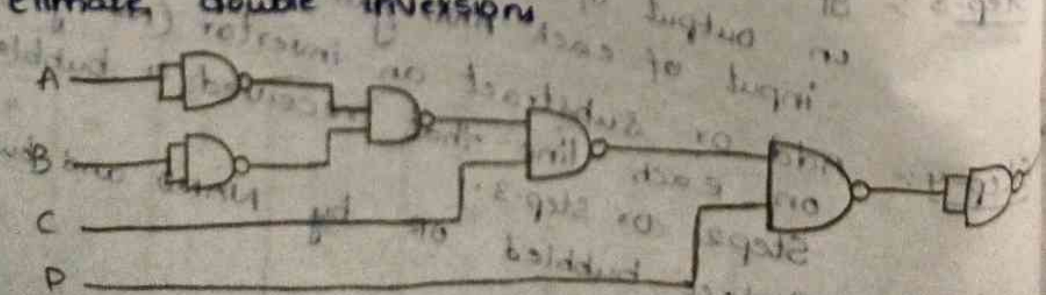
Add bubbles on the inputs of OR gates and output of AND gate



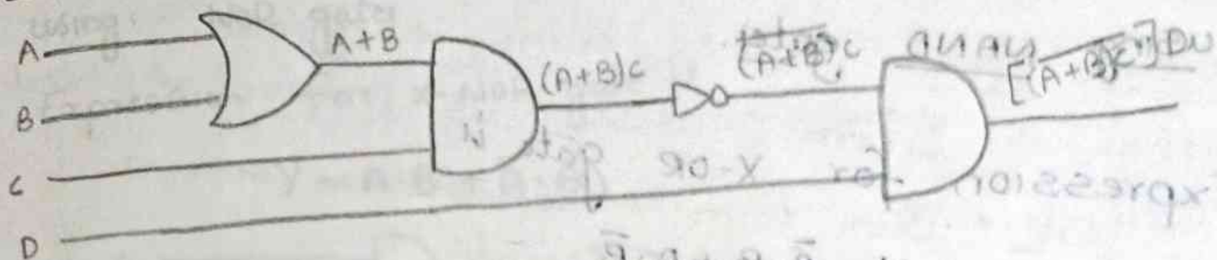
Add ~~cool~~ invertor (NOT gate) on each ~~bubble~~ line that received a bubble



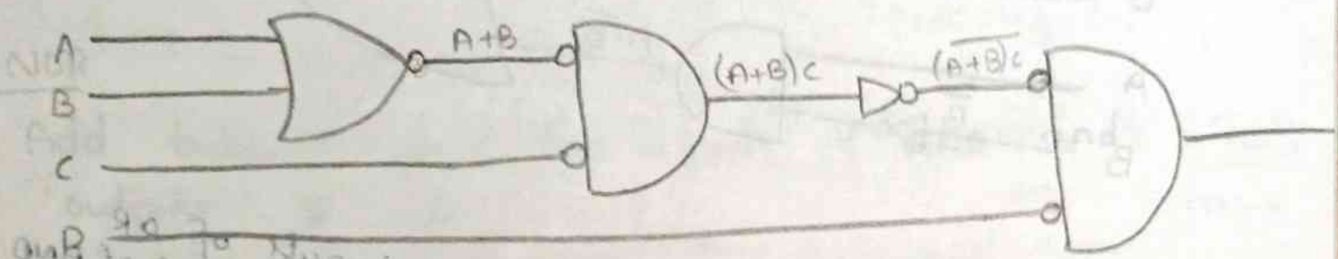
Replace ~~double~~ bubble OR by NAND and elimate double inversions



NOR

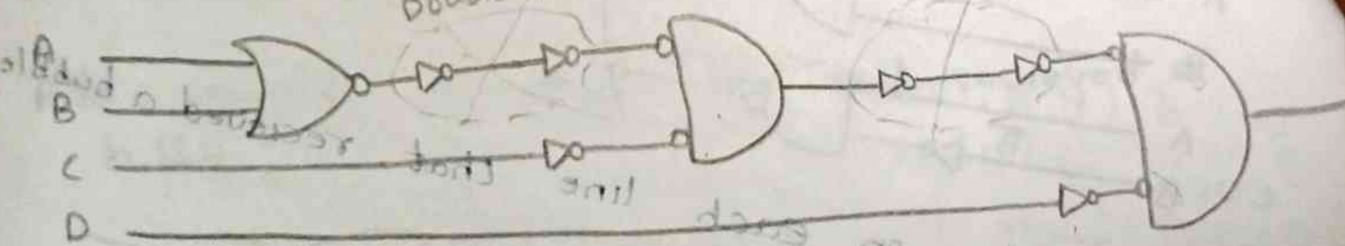


Add bubbles on the inputs of AND gate and output of OR gate

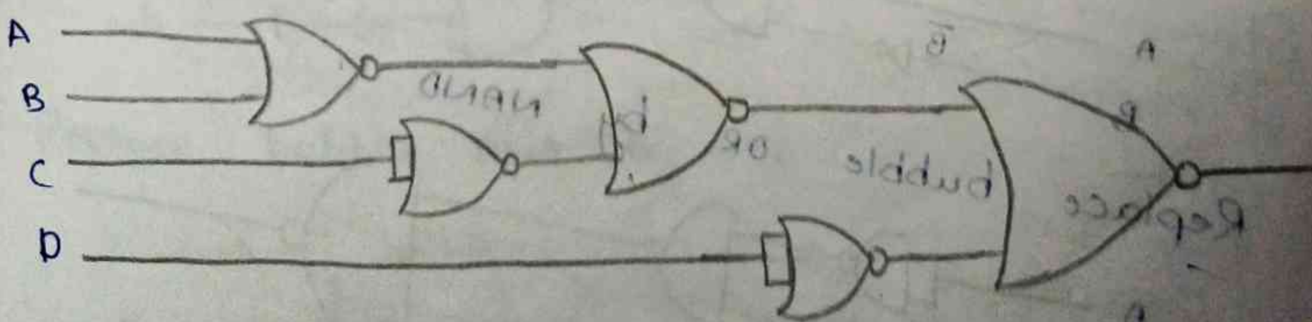


Add invertor (Not gate) on each line that received a bubble

Double inversions so cancel it



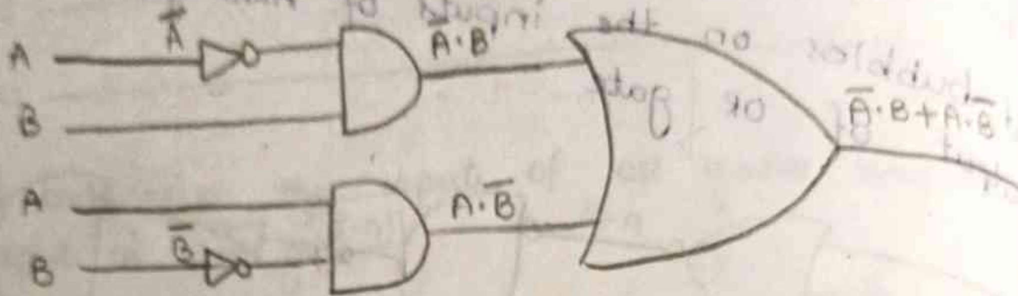
Replace bubble AND by NOR and eliminate double inversions



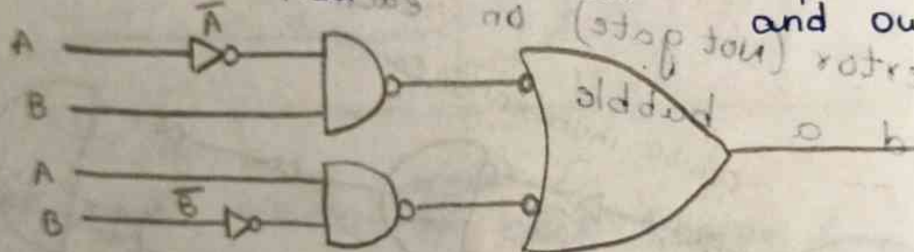
Implement boolean expression for X-OR gate using NAND gates

Expression for X-OR gate is

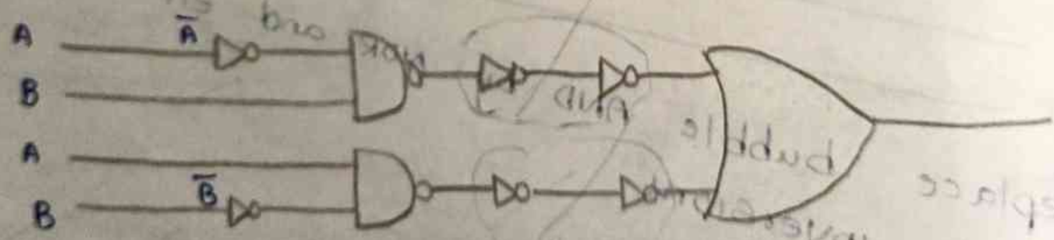
$$Y = \bar{A}\cdot B + A\cdot \bar{B}$$



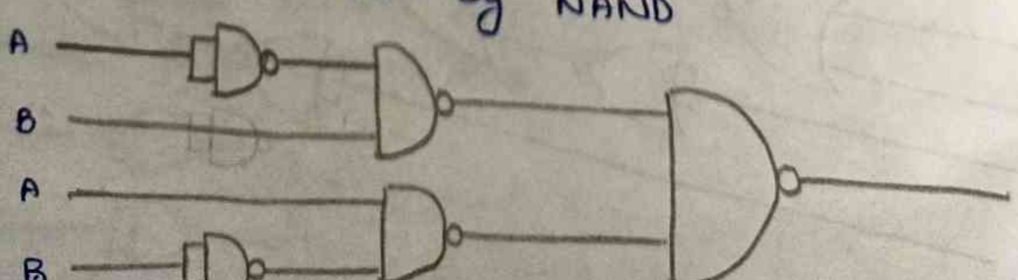
NAND Add bubbles on the inputs of OR and outputs of AND



Add inverter on each line that recieved a bubble



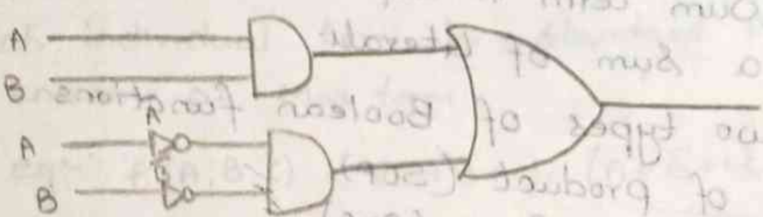
Replace bubble OR by NAND



Implement boolean expression for X-NOR gate using NOR gates

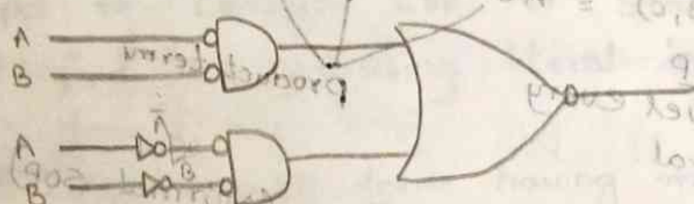
Expression for X-NOR gate is

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

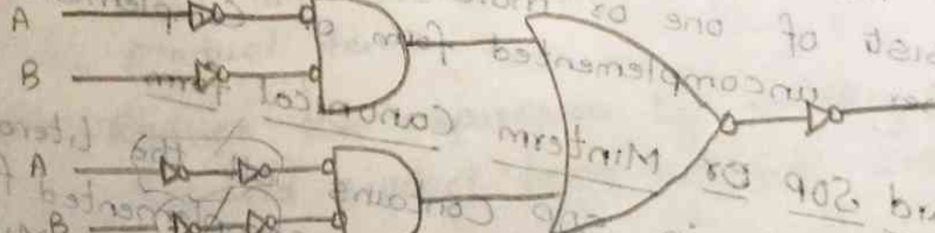


<u>NOR</u>

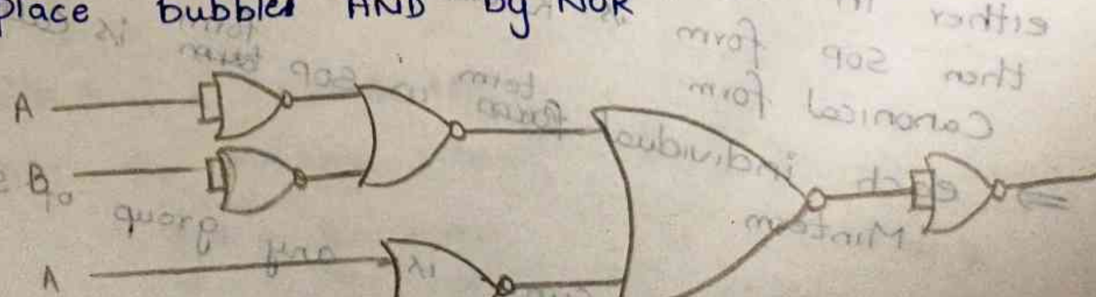
Add bubbles on the inputs of AND and outputs of OR



Add invertor on each line that recieved a bubble



Replace bubbles AND by NOR



SOP and POS

product term:- product term is defined as either a Literal (variable) or product of Literals

Sum term:- Sum term is defined as either a Literal or a sum of Literals.

they are two types of Boolean functions

⇒ Sum of product (SOP)

⇒ product of Sum (POS)

SOP:- A Sum of products is a group of product terms OR together (ORed)

sum terms

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

product terms

Standard SOP

which involves every Literal

(ii) $f(P,Q,R,S) = PQ + \bar{R}S + Q\bar{S}$ (Normal SOP)

each of these SOP expressions consists of 2 or more product terms and each product term Consist of one or more Literals appearing in either uncomplemented form or Complemented form

Standard SOP or Minterm Canonical form

If each term in SOP Contains all the literals either in Complemented or uncomplemented form then SOP form is known as standard or Minterm Canonical form.

⇒ each individual ~~form~~ term in SOP ~~term~~ form is called Minterm

POS:-

⇒ A product of Sums is any group of Sum terms ANDed

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

product

Standard POS term

⇒ If each term in POS contains all the literals either in complemented form or in uncomplemented form then the POS form is known as Standard POS or Max term. Canonical POS term

⇒ each individual term in standard POS term form is known as Max term.

eg:- $f(A,B,C) = (A+B+C) \cdot (A+\bar{B}+C) \cdot (A+\bar{B}+\bar{C})$

↓ Sum term or Max term

Converting expressions in Standard SOP or POS forms

1) steps to Convert SOP to Standard SOP

Step-1:- find the missing literal in each product term if any

Step-2:- "AND" each term having missing literal/literals with term/terms form by "OR"ing the literal and its complement

Step-3:- expand the terms by applying distributive law and reorder the literals in the product terms

Step-4:- Reduce the expression by omitting (Removing) Repeated product terms because $A+A=A$.

Eg-1 Convert the given expression in Standard SOP form

i) $f(A,B,C) = AB+BC+AC$

Write the missing literals

$$\underset{\downarrow\, C}{AB} + \underset{\downarrow\, A}{BC} + \underset{\downarrow\, B}{AC}$$

AND the missing literal by ORing its complement

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

$$ABC + AB\bar{C} + BCA + BC\bar{A} + ACB + AC\bar{B}$$

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

omitting the Repeating terms

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

Eg-2

$$f(A,B,C) = A + ABC$$

write the missing literals

$$A + ABC$$

$$\downarrow$$

$$BC$$

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

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

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

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

Steps to convert POS form to Standard POS

Step-1 :- find the missing literal in each sum term if any

Step-2 :- "OR" each sum term having missing literal / literals with term/terms form by AND the literal and its complement

Step-3 :- Expand the terms by applying distributive law and reorder the literals in the sum terms

Step-4 :- Reduce the ~~omitting~~ expression by omitting Repeated sum terms by $A \cdot A = A$

Eg-1 :- Convertt the given eqn Standard POS

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

write the missing literals

$$(A+B)\,(B+C)\,(C+A)$$

$$\downarrow C \qquad \downarrow A \qquad \downarrow B$$

$$= \left[(A+B) + C\cdot\bar{C}\right]\left[(B+C) + A\cdot\bar{A}\right]\left[(C+A) + \cancel{A\cdot A}\; B\cdot\bar{B}\right]$$

from distributive law

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

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

$A\cdot(A+B+C)$

$\downarrow$

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

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

K-Map :-

⇒ The basis of this method is a graphical chart is known as Karnaugh map (K-Map). It Contains boxes called, Cells.

⇒ Each of the cell represents one of the 2^n possible products that can be formed from n variables

2 Variables Can Contains $2^2 = 4$ cells

3 Variables Can Contains $2^3 = 8$ Cells

4 Variables Can Contains $2^4 = 16$ Cells

2 Variable K-Map

A \ B	$\bar{B}$	B
0 $\bar{A}$	$\bar{A}\bar{B}$ 00 — 0	$\bar{A}B$ 01 — 1
1 A	$A\bar{B}$ 10 — 2	AB 11 — 3

3 Variable K-Map

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	$\bar{A}\bar{B}\bar{C}$ 000 — 0	$\bar{A}\bar{B}C$ 001 — 1	$\bar{A}BC$ 011 — 3	$\bar{A}B\bar{C}$ 010 — 2
A	$A\bar{B}\bar{C}$ 100 — 4	$A\bar{B}C$ 101 — 5	ABC 111 — 7	$AB\bar{C}$ 110 — 6

4 Variable K-Map

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	$\bar{A}\bar{B}\bar{C}\bar{D}$ 0000 0	$\bar{A}\bar{B}\bar{C}D$ 0001 1	$\bar{A}\bar{B}CD$ 0011 3	$\bar{A}\bar{B}C\bar{D}$ 0010 2
$\bar{A}B$	$\bar{A}B\bar{C}\bar{D}$ 0100 4	$\bar{A}B\bar{C}D$ 0101 5	$\bar{A}BCD$ 0111 7	$\bar{A}BC\bar{D}$ 0110 6
AB	$AB\bar{C}\bar{D}$ 1100 12	$AB\bar{C}D$ 1101 13	$ABCD$ 1111 15	$ABC\bar{D}$ 1110 14
$A\bar{B}$	$A\bar{B}\bar{C}\bar{D}$ 1000 8	$A\bar{B}\bar{C}D$ 1001 9	$A\bar{B}CD$ 1011 11	$A\bar{B}C\bar{D}$ 1010 10

2 cells grouping

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1 0	1 1	3	2
$\bar{A}B$	1 4	5	7	6
AB	12	13	15	14
$A\bar{B}$	1 8	1 9	11	1 10

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

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

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

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

1) Minimise the expression $Y = A\bar{B}C + \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C}$ using K-map method

$A \backslash BC$	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	1 0	1 1	1 3	2
A	1 4	1 5	7	6

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

2) Simplify the Boolean function $F(A, B, C, D) = \Sigma m(0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14)$

$AB \backslash CD$	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1 0	1 1	3	1 2
$\bar{A}B$	1 4	1 5	7	1 6
AB	1 12	1 13	15	1 14
$A\bar{B}$	1 8	1 9	11	10

common lekapothe (1) common ani teesukovali

$$Y = \bar{C} + AB\bar{D} + C\bar{D}\bar{A}$$
$$= AB\bar{D} + \bar{A}C\bar{D} + \bar{C}$$

3) $f(A, B, C, D) = \Sigma m(0, 1, 2, 3, 11, 12, 14, 15)$

$AB \backslash CD$	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1 0	1 1	1 3	1 2
$\bar{A}B$	4	5	7	6
AB	1 12	13	1 15	1 14
$A\bar{B}$	8	9	1 11	10

$$Y = \bar{A}\bar{B} + ACD + AB\bar{D}$$

POS

$\Pi m(0, 2, 3, 8, 9, 12, 13, 15)$

AB \ CD	$C+D$	$C+\bar{D}$	$\bar{C}+\bar{D}$	$\bar{C}+D$
$A+B$	0 0	1	0 3	0 2
$A+\bar{B}$	4	5	7	6
$\bar{A}+\bar{B}$	0 12	0 13	0 15	14
$\bar{A}+B$	0 8	0 9	11	10

quite opp to SOP

Bar Vunte SOP lo ikkada pos lo bar Radhu . ki badhulu

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