Binary addition

Date: / / Page no:

How to add two decimal no:

0 1				
547	5×10^2	4×10^2 +	7×10^0	12×10^0
25		2×10^1 +	5×10^0	is not a decimal Num.
572 ans.	5	7	2	

base(r) = 10

$7<9$	$12>9$ 0 to 9
$7 = 0+7$	$(10+2)\times10^0$
$0\times10^2+7\times10^1$	$10\times10^1+2\times10^0$

1> 1 1 0 → $1\times2^2+1\times2^1+0\times2^0$ base(r) = 2

1 0 1 → $1\times2^2+0\times2^1+1\times2^0$ 0 and 1

Sum 1 0 1 1 $1\times2^1+1\times2^0$ sum < 1

2×2^2 2>1

$2 = 2+0$

$2+0\times2^2$

$1\times2^3+0\times2^2$

	S	C
0+0	0	0
0+1	1	0
1+0	1	0
1+1	0	1

Binary Subtraction

1-0 = 1

1-1 = 0

(16 8 4 2 1)

17) 1 1 0 1 1 -27 $1\times2^4+1\times2^3+0\times2^2+1\times2^1+1\times2^0$

$1\times2^3 = 2\times2^2$

−1 0 1 1 0 22 $1\times2^4+0\times2^3+1\times2^2+1\times2^1+0\times2^0$

0 0 1 0 1 ans $0\times2^4+0\times2^3+1\times2^2+0\times2^1+1\times2^0$

borrow → 2

2) SCSE
3) 3CSE41 D-21

(11)

2 2 2

1 1 1 0 14

− 1 1 1 7

0 1 1 1 7 Ans

H.W.

$10 - 2+2^3$

$2 + 1\times2^3$

2 2 2

1 0 1 0 21

− 1 1 1 1 15

0 0 1 1 0 6 Ans

$2^1\ 2^0$

2 1

10→2

2 2 2

1 0 1 0

− 1 1 1 1

0 0 1 1 0

(1)

0 (10) 10

1 1 0 0 1

0 1 1 1 1

0 1 0 1 0

Minimization

Minimization
- Boolean Algebra
- K-map
- Tabulation

Boolean algebra is used when no. of variables are less and o/p may be either zero or one.
K-map is used when no. of variables are 2, 3, 4, 5 and o/p may be 0, 1 or X.
If no. of variables are very high then Tabulation Method is used.

Boolean Algebra

Invented in 1854 by George Boole.

Not operation :-

$A \rightarrow \overline{A}$ (or) A'

$\overline{\overline{A}} = A$

AND operation :-

$0 \cdot 0 = 0$
$0 \cdot 1 = 0$
$1 \cdot 0 = 0$
$1 \cdot 1 = 1$

$A \cdot A = A$
$A \cdot 0 = 0$ ✓
$A \cdot 1 = A$
$A \cdot \overline{A} = 0$ ✓

OR operation :-

$A + A = A$
$A + 0 = A$
$A + 1 = 1$ ✓

Minimization

Minimization
- Boolean Algebra
- K-map
- Tabulation

Boolean algebra is used when no. of variables are less and o/p may be either zero or one.
K-map is used when no. of variables are 2, 3, 4, 5 and o/p may be 0, 1 or X.
If no. of variables are very high then Tabulation Method is used.

Boolean Algebra

Invented in 1854 by George Boole.

Not operation :-

$A \rightarrow \overline{A}$ (or) A'

$\overline{\overline{A}} = A$

AND operation :-

$0 \cdot 0 = 0$
$0 \cdot 1 = 0$
$1 \cdot 0 = 0$
$1 \cdot 1 = 1$

$A \cdot A = A$
$A \cdot 0 = 0$ ✓
$A \cdot 1 = A$
$A \cdot \overline{A} = 0$ ✓

OR operation :-

$A + A = A$
$A + 0 = A$

Minimization

Minimization
- Boolean Algebra
- K-map
- Tabulation

Boolean algebra is used when no. of variables are less and o/p may be either zero or one.
K-map is used when no. of variables are 2, 3, 4, 5 and o/p may be 0, 1 or X.
If no. of variables are very high then Tabulation Method is used.

Boolean Algebra

Invented in 1854 by George Boole.

Not operation :-

$A \rightarrow \bar{A}$ (or) A'

$$\bar{\bar{A}} = A$$

AND operation :-

$0 \cdot 0 = 0$
$0 \cdot 1 = 0$
$1 \cdot 0 = 0$
$1 \cdot 1 = 1$

$A \cdot A = A$
$A \cdot 0 = 0$ ✓
$A \cdot 1 = A$
$A \cdot \bar{A} = 0$ ✓

OR operation :-

$A + A = A$
$A + 0 = A$
$A + 1 = 1$ ✓
$A + \bar{A} = 1$ ✓

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

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

$\Rightarrow A[B+\bar{B}] = A$ //

Advantages of minimization :-

(1) No. of logic gates decreases
(2) Speed is increased
(3) Power dissipation is decrease
(4) Fan in may be reduced.
(5) Complexicity will decrease.

---

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

Sol :-

$A + AC + AB + BC$

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

$= A + BC$

(1) $(A+B)(A+C) = A + BC$ Transposition Theorem.

(2) $(X+Y)(\bar{X}+Z) = \bar{X} + YZ$

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

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

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

$= A + B\bar{C}$

④ $(A+B)(A+\bar{B})(\bar{A}+B)(\bar{A}+\bar{B})$

$= (A)(\bar{A}) = 0$

⑤ $(x+y)(\bar{x}+\bar{y})(\bar{x}+y) = y(\bar{x}+\bar{y}) = \bar{x}y$

$\boxed{(A+B)(A+C) = A+BC}$ Transposition theorem

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

$① + ②③ = (① + ②)(① + ③)$

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

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

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

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

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

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

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

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

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

$= \bar{A} + B$

$AB + B\bar{C} + AC = B\bar{C} + AC$

$AB + \bar{B}C + AC = AB + \bar{B}C$

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

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

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

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

Demorgan's Theorems :-

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

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

Boolean algebra

1. Minimization
2. Sop — minimal, canonical
3. Pos — minimal, Canonical
4. Truth Tables
5. Dual
6. complements
7. Venn diagrams
8. Switching ckts
9. Statement.

→ SOP form will used when o/p of logic ckt is one.

→ SOP form each product terms is known as mine term (or) Implement.

$5 \longrightarrow 101$

$A\ \bar{B}\ C$

$9 \longrightarrow 1001$

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

Minterm of.

For given truth table minimize that

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

$\bar{A}\bar{B}$ ← (0,0) ; $A\bar{B}$ ← (1,0)

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

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

$= \bar{B}$

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

00 , 10

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

$\bar{B}\,[A+\bar{A}]$

$= \bar{B}$

for given expression.

$$Y(A,B) = \sum m(0,2,3)$$

Minimize the expression.

00, 10, 11

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

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

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

$$= A + B$$ // Minimal

$A + \bar{A}B$ =

Minimal → $A + B$

Canonical → $A \cdot 1 + \bar{A} \cdot B$

(Standard form) → $A[B+\bar{B}] + \bar{A}B$

⇒ $AB + A\bar{B} + \bar{A}B$

No. of minterms in Canonical SOP form

$A + B\bar{C} \cdot 1$

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

(a) 4
(b) 5
(c) 6
(d) 7

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

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

POS

→ POS form is used when o/p of logic ckt is zero.

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

max. term

→ In pos form each term is known as Max terms.

$5 \longrightarrow 101$

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

→ For given truth table minimize POS form.

A	B	Y
0	0	1
0	1	0 ✓
1	0	1
1	1	0 ✓

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

$= \pi M(1,3)$

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

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

$\Rightarrow \sum m(0,2)$

$\sum m(0,2) = \pi M(1,3)$

$\sum m(0,1,5,7) = \prod M(2,3,4,6)$

for given truth table minimized SOP expression & POS expression same.

Dual

→ Dual Expression is used to convert +ve logic to negative logic or negative logic to +ve logic.

+ve logic AND

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

-ve logic (OR)

1	1	1
1	0	1
0	1	1
0	0	0

+ve logic AND = -ve logic OR

+ve logic OR = -ve log AND

Dual

① AND ↔ OR

② . ↔ +

③ 1 ↔ 0

④ keep variable as it is.

$A\bar{B}C + \bar{A}BC + AB\bar{C}$ Dual.

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

→ for any logical expression if two times dual is used resulting same expression.

→ with n variable max. possible min terms as max. term → 2^n.

→ with n variable max. possible logical expression is 2^{2^n}

$$n \Rightarrow 2 \Rightarrow 2^{2^2} \Rightarrow 2^4 \Rightarrow 16$$

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

$$n \Rightarrow 4 = 2^{2^4} = 2^{16} = 2^6 * 2^{10} = 64k$$

$$64 \times 1024 = 65536$$

Self Dual →

$AB + BC + AC$

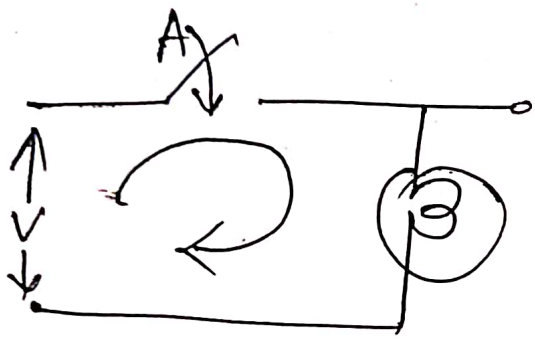
$(A+B)(B+C)(A+C)$

$\Rightarrow \{B+(AC)\}(A+C)$

$\Rightarrow AB + BC + AC$

Switching Ckts

(1) For given switching ckt o/p Y is.

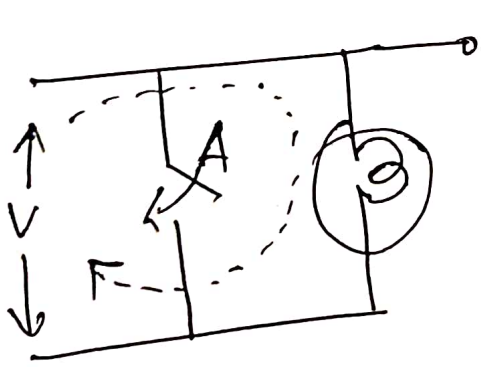


A	Y
0	0
1	1

Buffer

$Y = A$

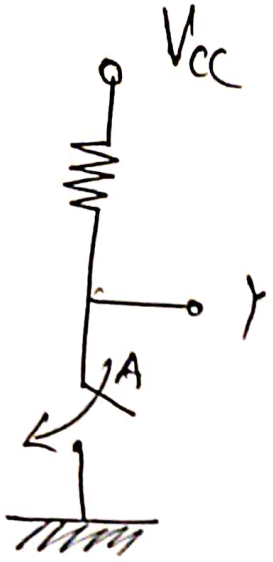
(2)



A	Y
0	1
1	0

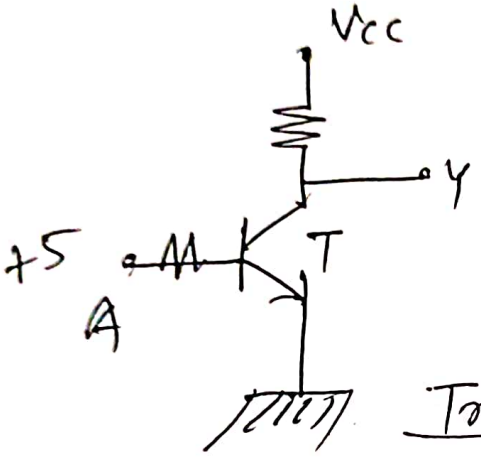
Not

$Y = \bar{A}$



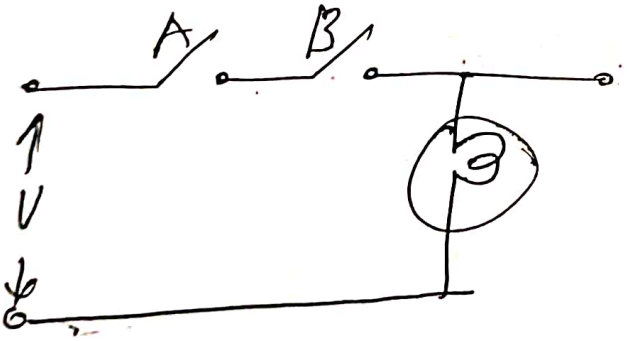
A	Y
0	1
1	0

$Y = \bar{A}$



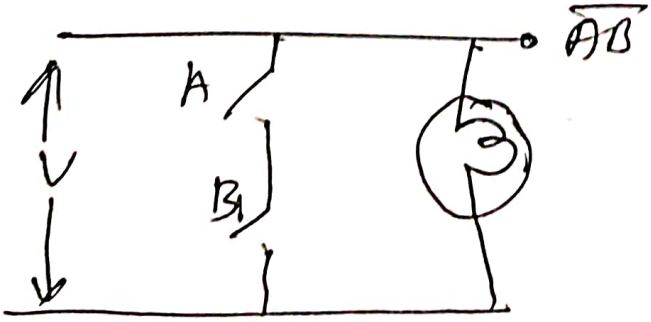
A	T	Y
0	off	1
1	on	0

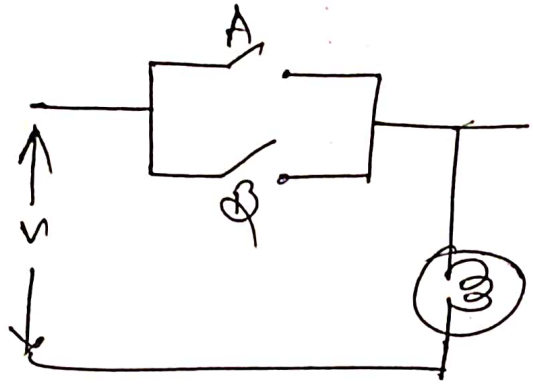
Transistor Not gate

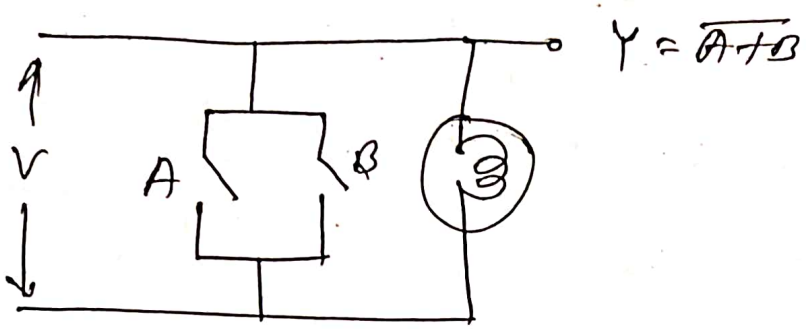


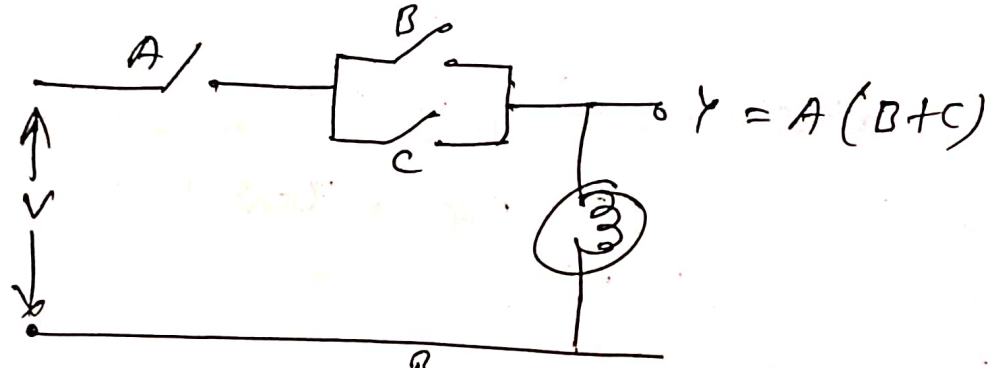
$Y = AB$.

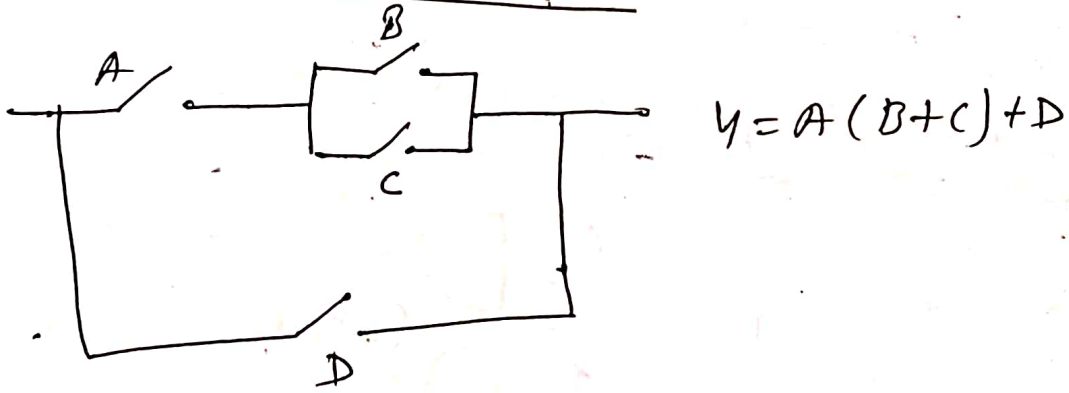
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1











→ A logic ckt have 3-i/p A & B & C & o/p is Y, Y=1, for the following combination.

(1) B and C are false → $\overline{B}\,\overline{C}$

(2) A and C are true. → AC

Logic Gates

Basic building blocks of digital ckt.

1. NOT
2. AND
3. OR

} Basic Gates

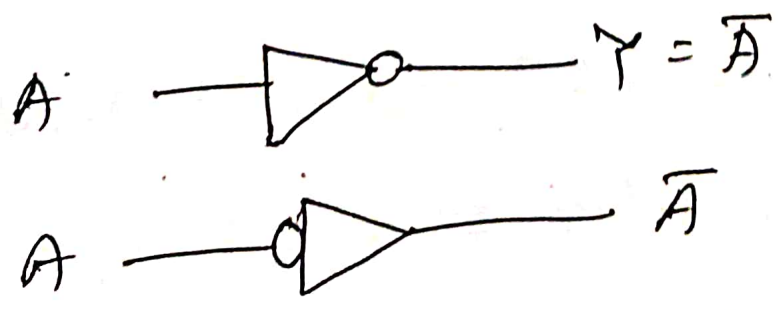
4. NAND
5. NOR

} universal Gates

6. EXOR
7. EXNOR

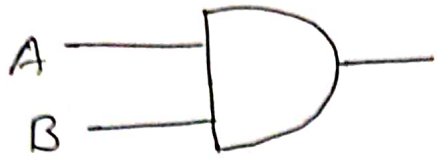
} Arithmetics.
Comparators
parity Generation/checking.
code Converters.

NOT



A	Y
0	1
1	0

AND Gate



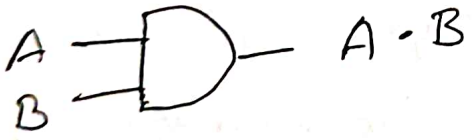
$Y = A B$

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

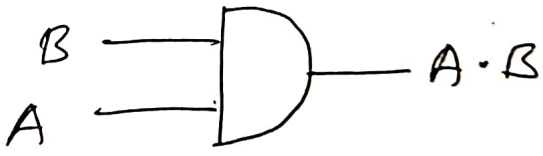
Series Switch

Property

① Commutative



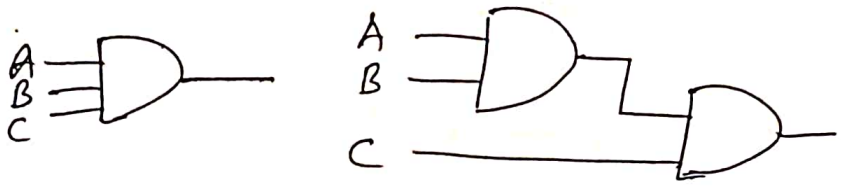
$A \cdot B$



$A \cdot B$

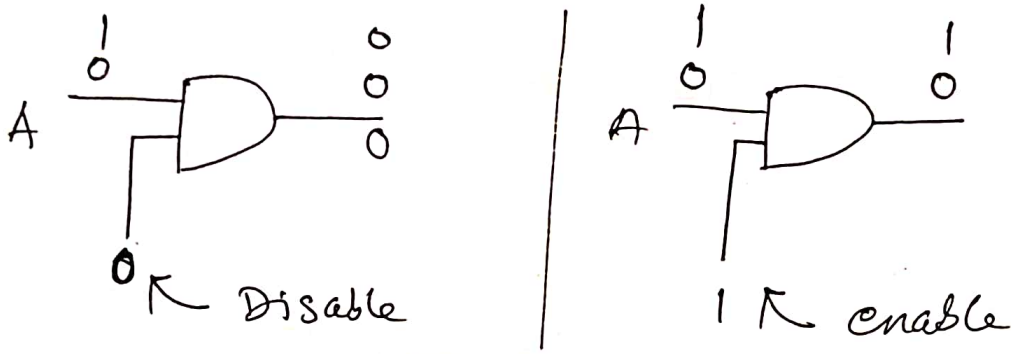
$$A B = B A$$

② Associative

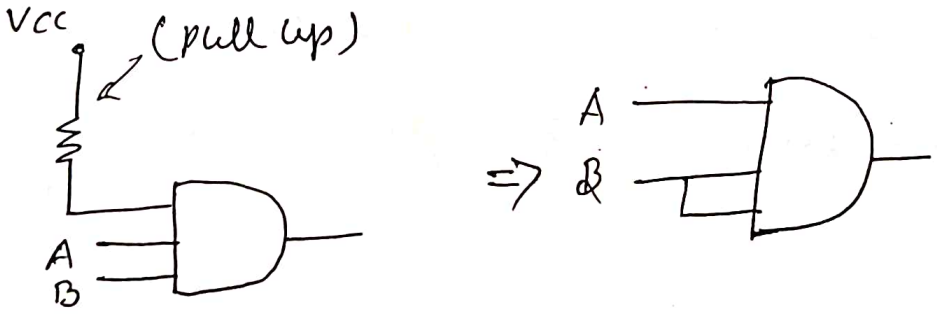


$$(A \cdot B)\, C = A\,(B \cdot C)$$

⇒ And Gate follows both commutative & associative law.



And	→ 0	→ Disable	→ O/p does not Change
And	→ 1	→ Enable	→ O/p ~~does not~~ Change

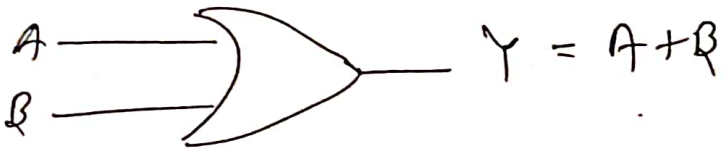


⇒ In TTL logic family any i/p is open or floating then it will acts as logic one (1).

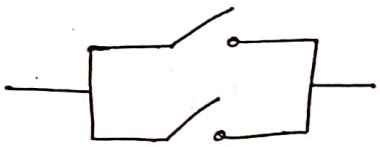
⇒ where as in ECL logic family floating i/p is acts as logic zero (0).

OR Gate

Symbol :-



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



Property —

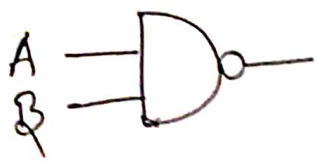
(1) It follows Commutative law.

$$A + B = B + A$$

(2) It follows associative law.

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

NAND GATE

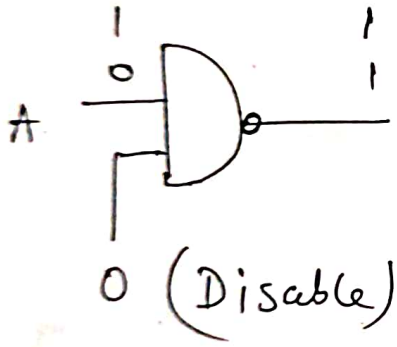


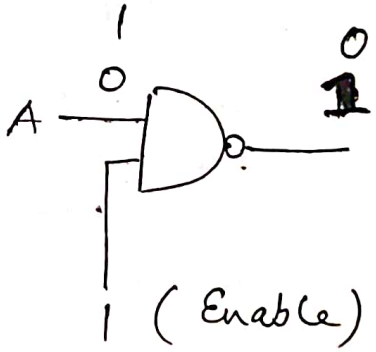
$Y = \overline{A \cdot B} = \overline{A} + \overline{B}$

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

Bubbled OR Gate

⇒ If Any i/p is zero o/p is one. and. both the i/p is one o/p is zero.



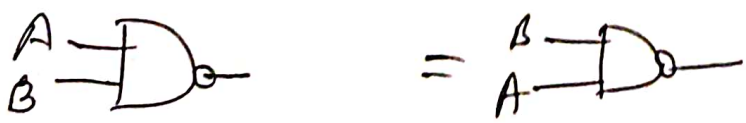


NAND → Enable → 1 → o/p changes
NAND → Disable → 0 → o/p doesn't change

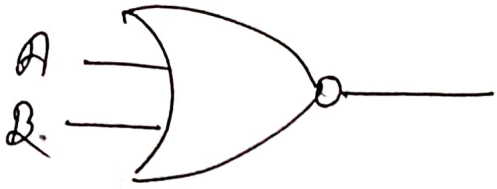
Property

① It follows Commutative law.

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

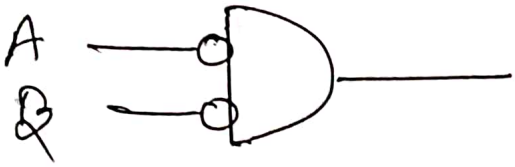


NOR Gate



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

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

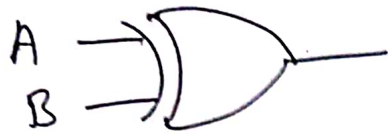


$\overline{A} \cdot \overline{B}$

Bubbled AND gate

⇒ Any i/p is one ~~opt~~ o/p is zero & if both i/p is zero o/p is one.

EXOR (OR) XOR (Exclusive OR)



$Y = A \oplus B$

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

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

Note: **

OR gate = Inclusive OR

**

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

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

$A = B \Rightarrow Y = 0$

$A \neq B \Rightarrow Y = 1$

3 Variable XOR gate. (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

Logical Expression :-

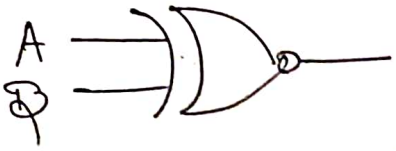
**

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

***

⇒ XOR gate is available with only two I/P. If we used multi i/p XOR gate we will go for Associative law. This Cause delay on O/P.

EXNOR Gate



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

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

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

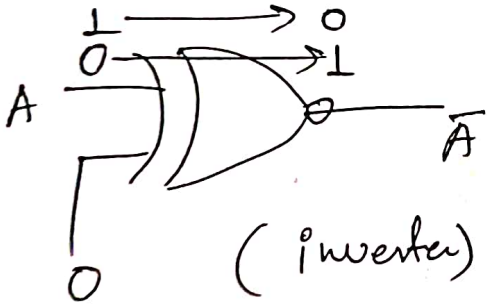
Synonams –

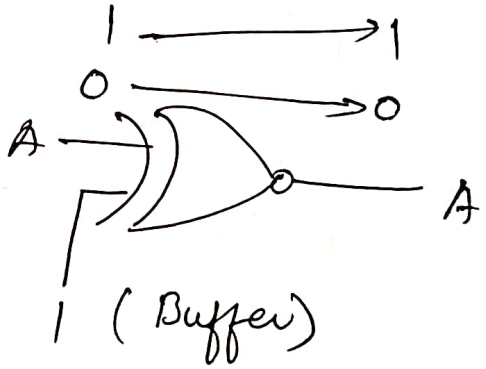
1. Equivalence gate
2. Coincedence logic gate.

Property –

$A = B \Rightarrow 1$

$A \neq B \Rightarrow 0$





Conclusion :-

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

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

$A \odot 1 = A$

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

Relations

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

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

(*) $$\boxed{A \oplus B \oplus C = A \odot B \odot C}$$

Let LHS : $\bar{A}B\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

RHS $A \odot B \odot C = (\overline{AB + \bar{A}\bar{B}}) \odot C$

$= X \odot C = XC + \bar{X}\bar{C}$

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

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

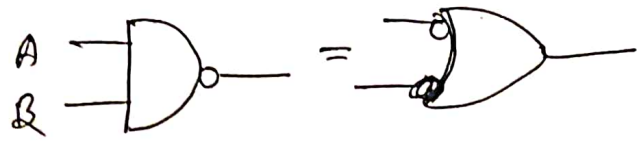
$= ABC + \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C}$

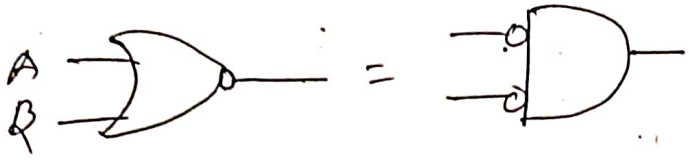
$=$ LHS

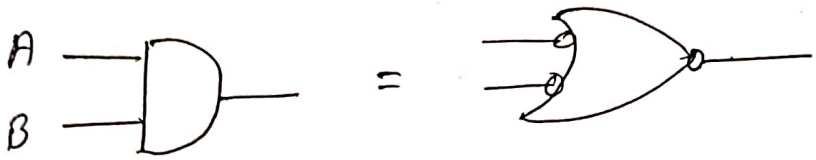
Hence, $A \oplus B \oplus C = A \odot B \odot C$ (Proved)

NAND As Universal

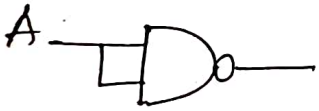
***



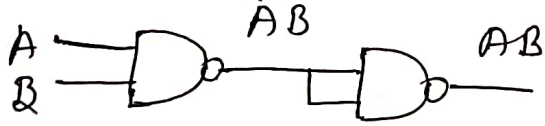




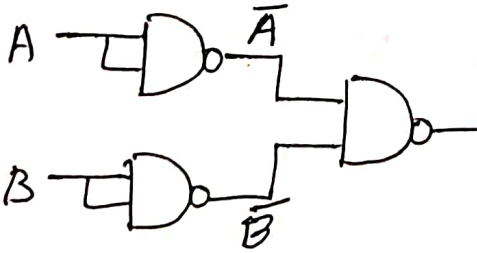
As NOT gate :-



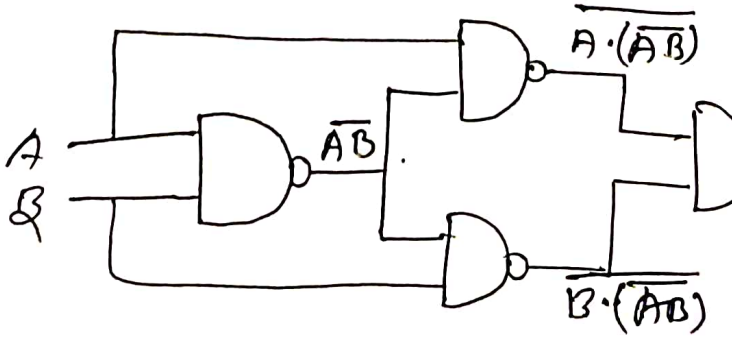
$\overline{A \cdot A} = \overline{A}$ ⟶ (1)

AND → 

⟶ (2)

OR → 

$\overline{\overline{A} \cdot \overline{B}} = A + B$ ⟶ (3)

XOR → 

⟶ (4)

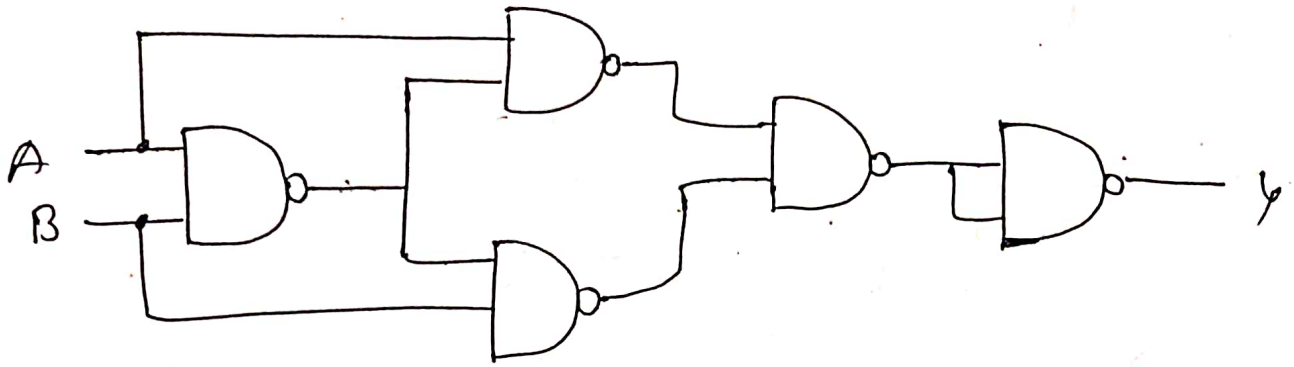
$\overline{(A \cdot (\overline{AB}))(B (\overline{AB}))}$

$= (A \cdot (\overline{AB}))' + (B (\overline{AB}))'$

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

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

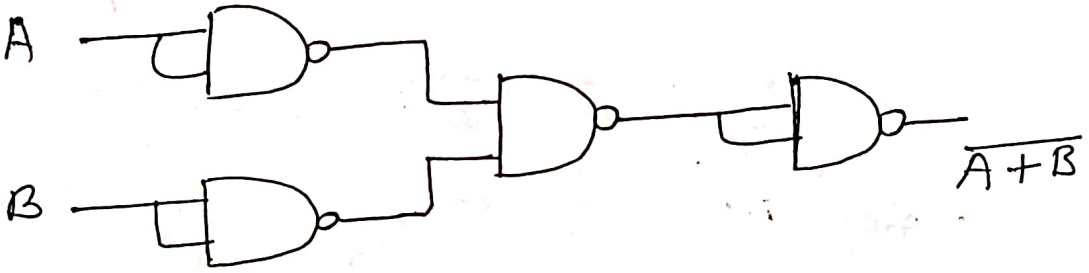
Ex NOR → → (5) gate.



~~NAND~~

NOR:

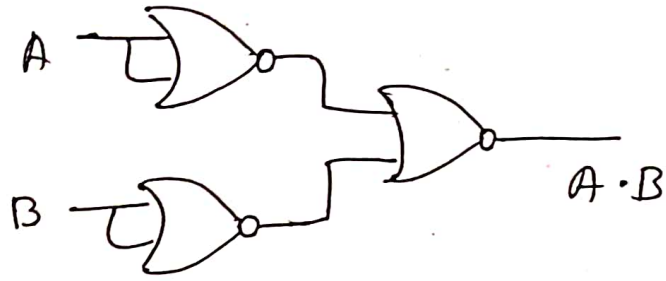
→ (4)



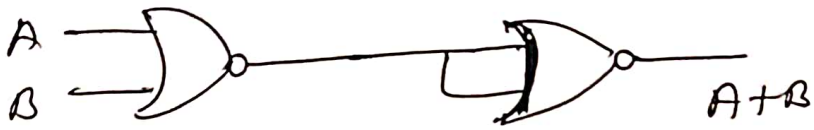
NOR as Universal

NOT → 

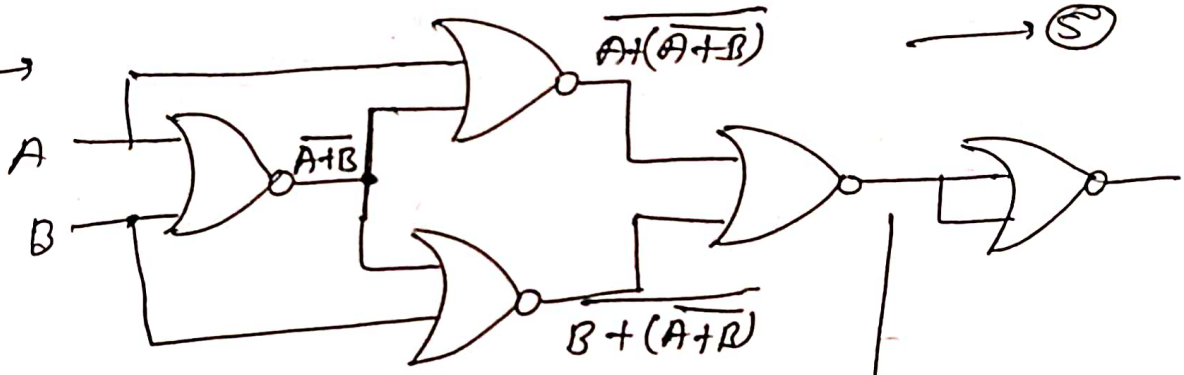
→ ①

AND → 

→ ③

OR → 

→ ②

EX-OR → 

→ ⑤

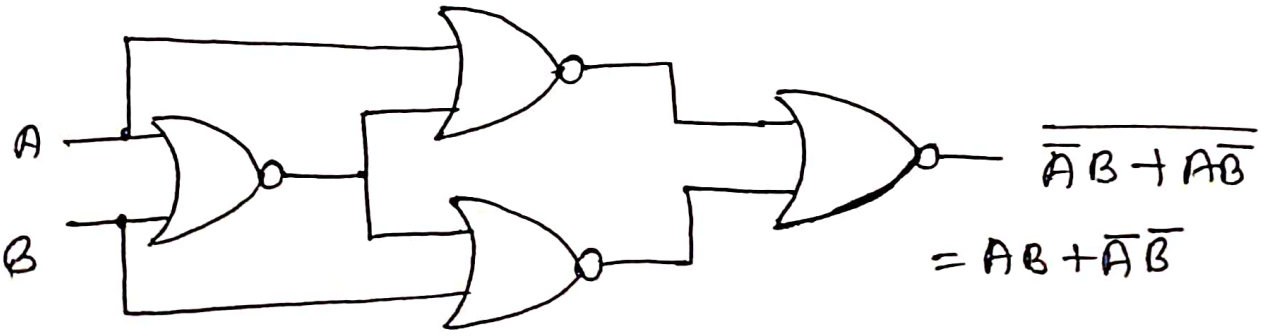
$$\overline{A+(\overline{A+B})} + \overline{B+(\overline{A+B})}$$

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

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

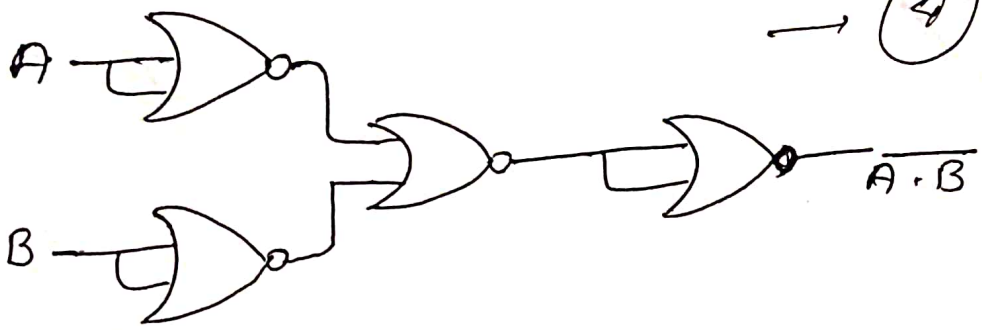
$$\Rightarrow \overline{\bar{A}B + A\bar{B}}$$

EX-NOR → → ④



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

NAND ⇒



→ (4)

Summary :-

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

Minterm and Maxterm :-

Minterms :- Each individual term in SSOP is called as minterm.

Maxterm :- Each individual ~~SPOS form~~ terms in SPOS is called as Maxterm.

For Two variable function →

F_1	Variable. A	B	Minterm. SSOP	Maxterm. SPOS
1	0	0	$\bar{A}\bar{B} \rightarrow m_0$	$A+B \rightarrow M_0$
0	0	1	$\bar{A}B \rightarrow m_1$	$A+\bar{B} \rightarrow M_1$
1	1	0	$A\bar{B} \rightarrow m_2$	$\bar{A}+B \rightarrow M_2$
1	1	1	$AB \rightarrow m_3$	$\bar{A}+\bar{B} \rightarrow M_3$

Ex $f(A,B) = AB + \bar{A}B$ - this is in SSOP form.
1 1 0 1

$\Sigma m(3, 1)$ Any where you saw like this sing its minterm in SSOP Form.

POS $(A+B)(\bar{A}+B)$
0 0 1 0
$M_0 \quad M_2 \Rightarrow \Pi M(0,2)$

Σ → denotes minterm in SSOP form.

what is maximum value is there, two variable.

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

$f = \Sigma m(0,1,3)$ $\Sigma m(0,2,3)$

Maxterm is complement to each other.

How can directly we can identify by Max. term and minterms each other.

Example:

$F_{ABC} = \Sigma_m(0,2,3,5) \rightarrow F(ABC)\ \Pi(1,4,6,7)$

↑ variable ⇒ There are three variable

How many minterms and maxterm.

Total minterms and maxterm. $2^n = 2^3 = 8$

$(0, 1, 2, 3, 4, 5, 6, 7, 8)$

* SOP to SSOP *

* steps *

1. Identify the missing variable in product term
2. multiply (variable + its complements)
3. Neglect the repeated term.

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

This is a three variable function.

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

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

$= \bar{A}BC + AB\bar{C} + \bar{A}BC,\ \Sigma m(3,6,7)$

$\quad 011 \quad 110 \quad 011$

This is Resultent of SSOP form. M3 M6 M7

Why we need conversion this SOP in SSOP For Example in your exams they asking by giving above o example what are the minterm of this expression, then we cant say what are the minterm for in your Expression we cant say by this SOP expression so definatly we need to SSOP expression to defined the minterm.

Date: / /

POS to SPOS

Steps

* Identify the missing variable.
* Add with that variable & its Complement seperatly
* Neglect the repeated terms.

Example :- $f(A,B,C) = A(A+C)$ (POS)

B, C are missed — B is missed.

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

$= \underset{0\,0\,0}{(A+B+C)}\ \underset{0\,1\,0}{(A+\bar{B}+C)}\ \underset{0\,1\,1}{(A+\bar{B}+\bar{C})}\ \underset{0\,0\,1}{(A+B+\bar{C})}$ SPOS

$\Pi_M [M_0, M_2, M_3, M_1]$

$\Pi_m [0, 1, 2, 3]$ ans

Converting SSOP to SPOS

$f(A,B,C) = \Sigma_m (0, 1, 3, 4, 7,)$

This is three variable.

minterms and maxterms. $2^n = 2^3 = 8$

what are the Associted Maxterm: Just take the complement. Ss Complement what ever. Min.terms are not present that numbe will come Maxterms

$f(A,BC), \Pi M[2, 5, 6]$

010 . 101 110

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

Ex Express the Function $Y = A + \bar{B}C$ in (a) canonical SOP and (b) canonical POS form.

Sol

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

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

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

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

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

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

$$ABC + AB\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + \bar{A}\bar{B}C$$
$$111 \quad 110 \quad 101 \quad 100 \quad 001$$
$$m_7 \quad m_6 \quad m_5 \quad m_4 \quad m_1$$

Therefore
$$Y = \Sigma m(1, 4, 5, 6, 7)$$

(b) canonical POS

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

$$= (A+\bar{B})(A+C) \quad [\because A + BC = (A+B)(A+C)]$$

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

$$= (A+\bar{B}+C)(A+\bar{B}+C)(A+\bar{B}+\bar{C})(A+B+C)$$
$$010 \quad 011 \quad M_0$$
$$M_2 \quad M_3$$

$$\therefore Y = \Pi[0, 2, 3]$$

SPOS → POSSOP

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

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

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

$= 0_{m0}\ 0_{m1} = \pi m[0,1] \Rightarrow \bar{P}$

$= \Sigma m[2,3]$ POS.

SSOP $A\bar{B}+AB$ POS.

* KARNAUGH MAP [K-map] *

Karnaugh map is also called as K-map
this is developed by scientist Karnaugh. by His name so this is developed in 1953

* Developed by Karnaugh in 1953
* used to ~~simplf~~ simplify boolean algebric expressions without using boolean theorems.

So when we are solving a boolean expression we use to consider boolean laws and theorems. ~~tha~~ so that may be somewhat ~~teppy~~ typical for us. So ~~that may~~ whenever you forget that laws and theorem..
* whenever you forget that laws and theorems you can make use of this K-map to simplify that Boolean expression

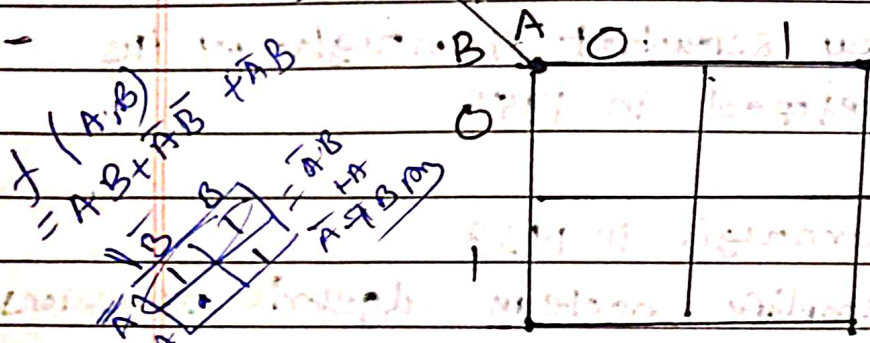
* Now will say how the K-map will be what is the structure of K-map and how we are having variables in that and that will see now.

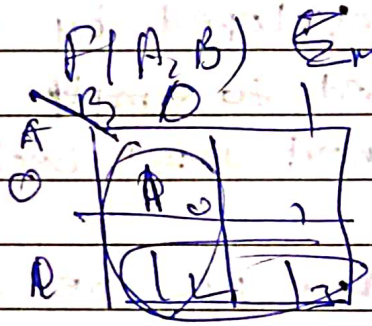
* K-map may be for 2 variable. we may have $2^2 = 4$ cells in K-map.

* K-map is nothing its a pictorial ~~representation~~ Representation or Pictorial solving method of Boolean expression without using boolean laws and theorems.

Example for 2 variable K-map.

this box always contained with the saperation and this ——— is for variables one is A and B. you can interchange the positions.





* number of cells $= 2^n$
$= 2^2$
$= 4$

n = number of variables

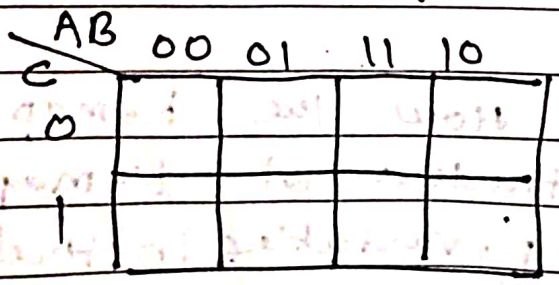
$\bar{B}$ + A

* Three variable K-map.

How to make structure of 3 variable K-map

* we can take vertically or Horizontaly

* what is the cell means the posibility of your variable combination.



n = 3

$2^n = 2^3 = 8$

* the reason behind this is for any state change For every state change you need change one 1 bit

3ESE4 - Dual of Boolean Expression.

* Dual is nothing but Interchanging OR with AND and AND with OR

Replace.

* OR with AND ($+ \longrightarrow \cdot$)
* AND with OR ($\cdot \longrightarrow +$)
* 1 with zero
* 0 with 1

Ex for the identity

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

The Dual form is

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

$$\Rightarrow A\bar{A} + AC + \bar{A}B + BC\,(B+C)$$

$$= 0 + ABC + \bar{A}B + BC + AC + \bar{A}BC + BC$$

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

$$= BC + \bar{A}B + \bar{B}C + AC = BC + \bar{A}B + AC$$

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