Date

Chapter no. 1:

Only 'one short question' expected.

graph and its terminologies (slide 10)

Reasons to shift from Analog and digital?

What are digital wave form?

Converting analog signals to digital we use 'Sampling and Quantization'.

Sampling → vertical (lines)

Quantization → horizontal (line)

Analog signals ;
- have high accuracy and handles.
- more detailed ∴ requires more power.
- more prone to noise.
- requires more space.

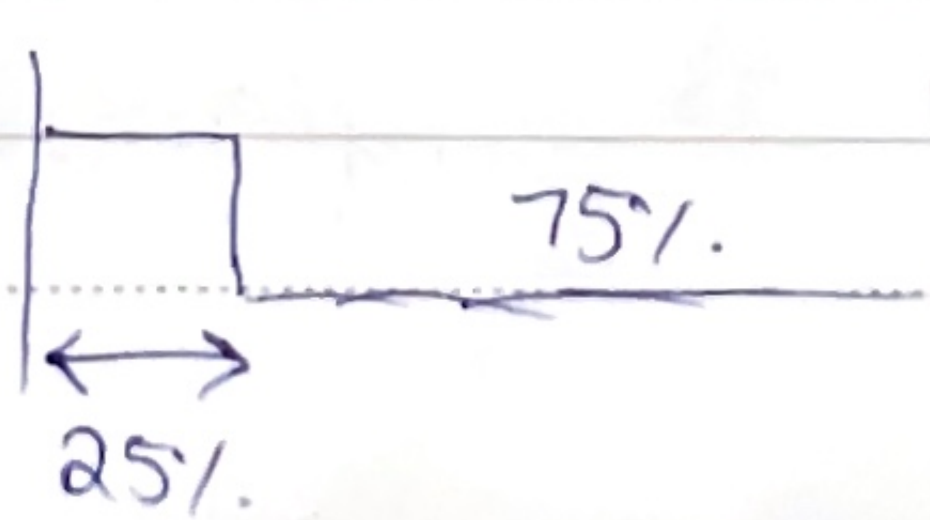
⇒ all characters; viceversa for digital signals.

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

(consecutive crest & truoghs =?

TTL Logic
↳ 0V - 2V low voltage. (would not allow follow of I)
3V - 5V - 10V high voltage (would allow follow of I)

Duty cycle, represent the stage of waveform in which the state is 'high'



∴ duty cycle = 25%.

Date INTRODUCTORY SLIDE 14:- 5

Q. What is the importance of studying 'LOGIC DESIGN'

Q. Coordination of many levels of abstraction. (", Slide 15)

~ x ~

① Binary to Decimal. (Multiply accordingly with the base of 2)

② Decimal to Binary (Multiple division Method) → if any other base is required divide by that no. in this method.

③ Decimal equivalent of a Binary No.

④ BINARY ADDITION:-

Basic Rules of Binary addition.

$1 + 0 = 1$ Sum part = 1, carry part = 0

$0 + 1 = 1$ " "

$0 + 0 = 0$ Sumpart = 0, carrypart = 0

$1 + 1 = 10$ Sum part = 0, carry part = 1.

3 Bit situation

$1 + 0 + 0 = 01$ Sum = 1, carry = 0

$1 + 1 + 0 = 10$ Sum = 0, carry = 1

$1 + 0 + 1 = 10$ Sum = 0 carry = 1

$1 + 1 + 1 = 11$ Sum = 1, carry 1

⑤ Binary substraction → Conventional Method
→ 2's complement Method.

$$\begin{array}{c|l} 2 & 4 \\ 2 & 2 - 0 \\ & 1 - 0 \end{array}$$

Date 10/09/24.

Chapter no. 2

DECIMAL NO. SYSTEM

$10^3, 10^2, 10^1, 10^0$ → (a) :- assigning accurate bases of 10.

1928

$$\begin{array}{cccc} 1 & 9 & 2 & 8 \\ 10^3 & 10^2 & 10^1 & 10^0 \\ \hline 1000 & +900 & +20 & +8 \end{array} \Rightarrow 1928.$$

(a) :- how a no. is formed using the base 10 terms.

$$\begin{array}{ccccccc} 1 & 9 & 2 & 8 & .\ 5 & 0 & 1 \\ 10^3, & 10^2, & 10^1, & 10^0 & 10^{-1}, & 10^{-2}, & 10^{-3} \end{array}$$

(b) :- assigning bases 10 in decimal terms.

$$\begin{array}{ccccccc} 1 & 9 & 2 & 8 & .\ 5 & 0 & 1 \\ 10^3 & \times 10^2 & \times 10^1 & \times 10^0 & 10^{-1} & 10^{-2} & 10^{-3} \\ \hline 1000 & +900 & +20 & +8 & .\ 5 & +0 & +0.001 \end{array}$$

$\Rightarrow 1928.501$.

BINARY NO. SYSTEM.

has the Base 2, i.e all natural numbers to be represented in 0s & 1s.

DECIMAL	BINARY
0	0
1	1
2	10
3	11
4	100

Date

BINARY TO DECIMAL :-

① 1 0 0 1 0

$2^4\ 2^3\ 2^2\ 2^1\ 2^0$

16 + 0 + 0 + 2 + 0 ⇒ (18)

② 1 0 1 0 . 1 0

$2^3\ 2^2\ 2^1\ 2^0 \mid 2^{-1}\ 2^0$

10 + 0.5 + 0 ⇒ (10.5)

DECIMAL TO BINARY

▷ a whole no. is simply found by multiple division method.

① 512 :-

2	512
2	256 — 0
2	128 — 0
2	64 — 0
2	32 — 0
2	16 — 0
2	8 — 0
2	4 — 0
2	2 — 0
	1 — 0

$(1000000000)_2$.

②

294

2	294 — 0
2	147 — 1
2	73 — 1
2	36 — 0
2	18 — 0
2	9 — 1
2	4 — 0
2	2 — 0
	1

$(100100110)_2$

~ x ~

512.33:-

- a decimal point conversion is operated through. (calculated)
  - → multiple division for base/non-decimal part
  - & → multiple division for the decimal part. (multiply by 2 4 to 5 time atmost)

512.33
↓
found in previous example

$0.33 \times 2 = \underline{0}.66$ carry (save this zero)
$0.66 \times 2 = \underline{1}.32$

** $\boxed{0.32} \times 2 = \underline{0}.64$ carry (save this zero)
$0.64 \times 2 = \underline{1}.23$

$(1000000000.\underline{0101})_2$
↓
Deduced as described above.

'BINARY ADDITION':

Sum part | Carry Part

→ Rules given in slides.

Date

Check in Binary.

$$\begin{array}{r} 21 \\ +\ 7 \\ \hline 28 \end{array} \Rightarrow \begin{array}{ccccc} 1 & 0^{1} & 1^{1} & 0^{1} & 1 \\ 0 & 0 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 0 & 0 \end{array}$$

carry part above

sum part below

$2^4\ 2^3\ 2^2\ 2^1\ 2^0$

$16+8+4+0+0 \Rightarrow 28.$

$$\begin{array}{r} 1^{1}11 \\ +\ 11 \\ \hline 110 \end{array} \qquad \begin{array}{r} 100 \\ 10 \\ \hline 110 \end{array}$$

$$\begin{array}{r} 111 \\ 111 \\ 11 \\ \hline 11010 \end{array} \qquad \begin{array}{r} 110 \\ 100 \\ \hline 1010 \end{array}$$

BINARY SUBSTRACTION:-

$(1010)_2 - (0101)_2 = (0101)_2$

$(10 - 5 = 5)$ how?

[Conventional Method]

$$\begin{array}{cccc} \overset{0}{\cancel{1}} & \overset{10}{\cancel{0}} & \overset{0}{\cancel{1}} & \overset{10}{\cancel{0}} \\ 0 & 1 & 0 & 1 \\ \hline 0 & 1 & 0 & 1 \end{array} \Rightarrow 5.$$

using the Rules of substraction.

$1-0=1$

$0-0=0$

$1-1=0$

with the borrow of 1. $\boxed{10-1=1}$

[2's Complement Method]

Add the 2's Complement of the 2nd No. with the 1st Number.

1's complement is; 1010

2's Complement is (+1) ⇒ 1011

Now;

$$\begin{array}{rr} & 1010 \\ & 1011 \\ \hline 1 & 0111 \end{array}$$

* last carry is ignored

2 | 3 − 1
| 1

Date

Binary Operation
Binary Multiplication/substr.

Binary Multiplication.

Multiplication Rules:

① $0 \times 0 = 0$
$0 \times 1 = 0$
$1 \times 0 = 0$
$1 \times 1 = 1\cancel{0}$

15 multiply 15

$$\begin{array}{r} 15 \\ \times\ 15 \\ \hline 225 \end{array}$$

$$\begin{array}{r} 100 \\ \times\ 10 \\ \hline 000 \\ 100\times \\ \hline 1000 \end{array} \Rightarrow 1000$$

$$\begin{array}{r} 1111 \\ \times\ 1111 \\ \hline 1111 \\ 1111\times \\ 1111\times\times \\ +\ 1111\times\times\times \\ \hline 11100001 \end{array}$$

→ as no Rule for this point consider it in Decimal i.e $1+1+1+1=4$ and $4 = 100$

carry part

sum part

Binary Division:-

$14 \div 3$ [ie $1110 \div 0011$]

find 2's complement.

0011
inverse
(1's complement) → 1100
(add '1') → $\boxed{1101}$

$$\begin{array}{r} 1\cancel{0}0.1\cancel{00} \\ 41 \enclose{longdiv}{1101.1101} \\ 11 \\ \hline \times\ 011 \\ 11 \\ \hline \cancel{0}1 \end{array}$$

$$\begin{array}{r} 100.1001\cancel{0} \\ 11 \enclose{longdiv}{1101.1101} \\ 11 \\ \hline 0011 \\ 11 \\ \hline 0101/10\cancel{0} \end{array}$$

① ④
9 ÷ 2
A − B

~> representing in bit in every no. system

Date

check
~> Dealing these example with point(.)
~> concept of conversion.

HEXADECIMAL NUMBERS.
(4 Bit combination

▷ Converting 1600 Decimal Number into Hexacimal system.

$$\begin{array}{r|l} 16 & 1600 \\ 16 & 100 - 0 \\ & 6 - 4 \end{array}$$

$\therefore (1600)_{10} \rightarrow (640)_{16}$ $\therefore$ converted.

▷ Converting 640 Hexadecimal number into Decimal Number.

640.510
same ↩
↓ ↓ ↓
16^{-1} 16^{-2} 16^{-3}

6 4 0
↓ ↓ ↓
×
16^2 16^1 16^0

$\Rightarrow 6 \times 16^2 + 4 \times 16^1 + 0 \times 16^0$
$\Rightarrow (1600)_{10}$
$\therefore$ hence converted.

▷ Converting Hexacimal into Binary.

F 4 0
// equal to 15
↓ ↓ ↓
$(1111\ 0100\ 0000)_2$ $\therefore$ converted

~> // just put individual Binary codes.

Date

7- SEGMENT DISPLAY :-

↳ can display from 0–9. (one)

BCD
(encode for decimal into Binary)

↓

Basically used to convert Binary of no. greater than 10 to show in 7-segment display by addition of '6' i.e $(110)_2$

Consider $(1010)_2 = (10)$

It cannot be shown in one 7 segment display ∴ we take BCD

$$\begin{array}{r} 1010 \\ 110 \\ \hline 0001\,0000 \end{array}$$

$\boxed{1}$ $\boxed{0}$

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

Date 19/09/24.

'GRAY CODE'

CONVERSION FROM BINARY TO GRAY CODE :-

Consider Binary code of 15 (in 5 Bit) into gray Code

1. Bring the MSB down as it is.
2. Take XOR of each Bit with theselves and bring down.

$010 + 5 = 15$

$\oplus\oplus\oplus\oplus$
01111
$\downarrow\downarrow\downarrow\downarrow\downarrow$
01000 ⇒ This is the gray code for 15.

CONVERSION FROM GRAYCODE TO BINARY :-

1. Bring the MSB down as it is.
2. Now take 1 Bit from below and take XOR with 1 from above.

$(1010)_G$ is the gray code for 12.

1010
$\downarrow$
$(1100)_2$ ⇒ The Binary Code.

Date

Chapter no. 3: 'DIGITAL FUNDAMENTALS'

XOR GATE (X)

TRUTH TABLE

A	B	$A \oplus B$ (X)
0	0	0
0	1	1
1	0	1
1	1	0

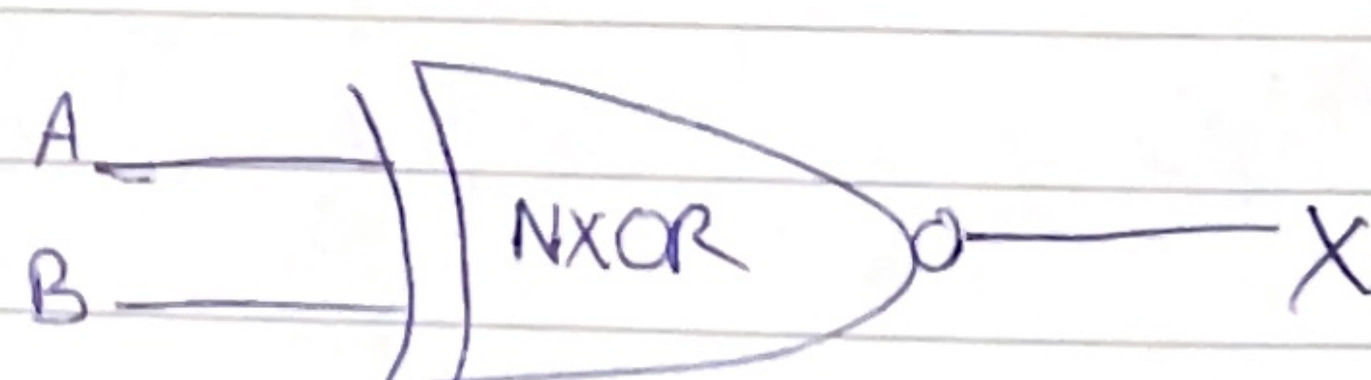
BOOLEAN ALGERA

$$\bar{A}.B + A\bar{B}$$
$$\bar{0}.1 + 0.\bar{1}$$
$$\cancel{0}1 + 0$$

XNOR GATE : $(\overline{A \oplus B})$

TRUTH TABLE (NOT OF XOR)

A	B	X
0	0	1
0	1	0
1	0	0
1	1	1



BOOLEAN ALGERA

XNOR = $AB + \bar{A}\bar{B}$. $(A \odot B)$

Number of variables

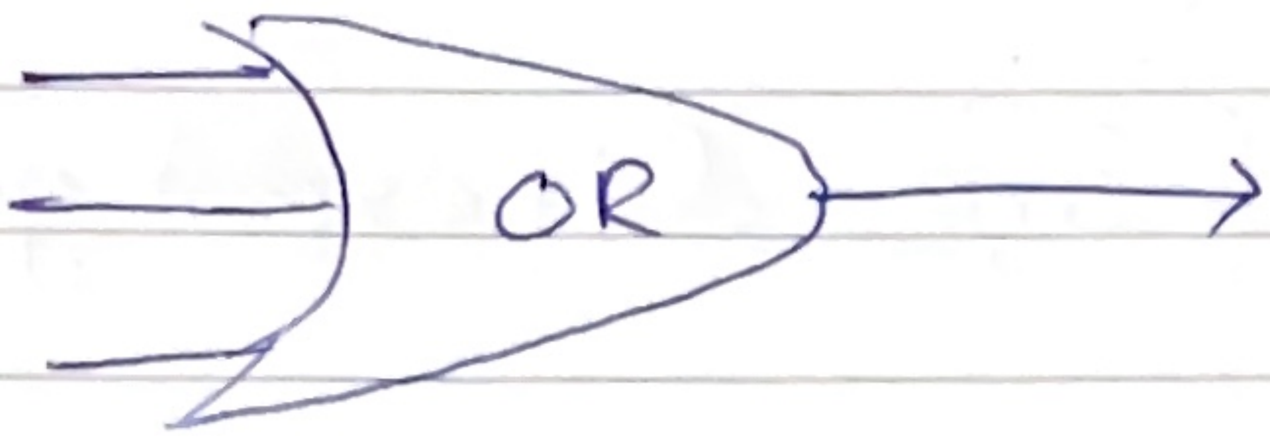
construct literals

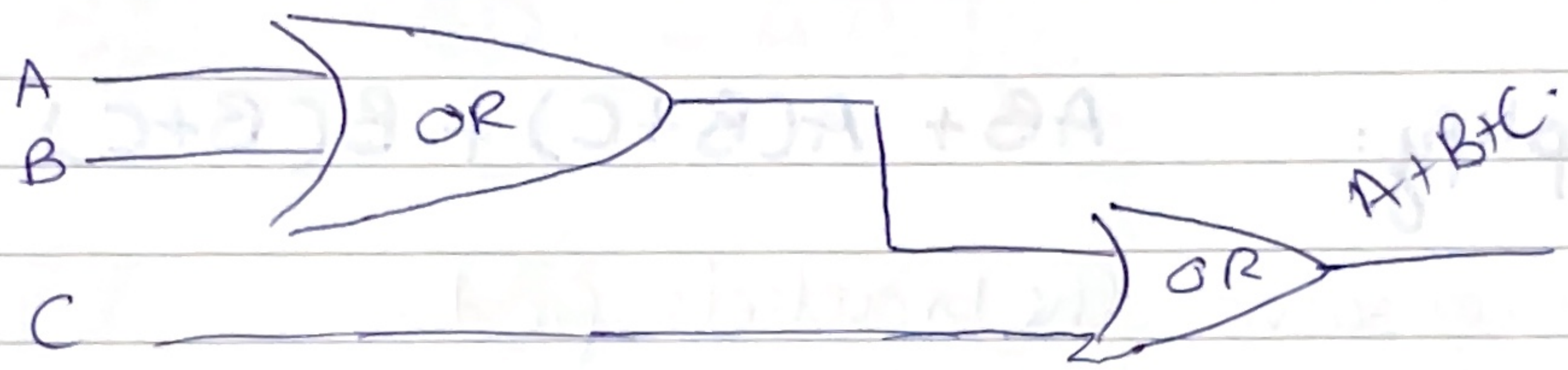
Date

Boolean Addition (equivalent to:)

OR :-

Obey's Associative law i.e $A+B+C = A+C+B$ etc...





A boolean OR is same as the OR logic operation.

Demorgan's law can be applied even with the increase in the no. of literals.

DEMORGAN'S LAW ↕ (2 basic laws)

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

imp* $\overline{ABCDE}\ \overline{FG} = (\bar{A} + \bar{B} + \bar{C} + \bar{D} + \bar{E})(\bar{F} + \bar{G})$

* 'NAND' & negative and are different gate.

Simplify: $AB + A(B+C) + B(B+C)$

alway salve the brackets first.

$AB + AB + \overset{A}{C} + BB + BC$ ⇒ By $A + A = A$

$AB + AC + BB + \overset{B}{C}$ ⇒ By $BB = B$

$AB + AC + B + \overset{B}{C}$.

taking common.

$B(1 + A + C) + AC$ ⇒ OR with 1 is alway 1.

$B(1) + AC$

$B + AC$ Ans

Create the result's truth table.

⇒ Boolean simplifications will include 4 inputs atmost.

Date

A	B	C	AC	B+AC
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	1
1	1	0	0	1
1	1	1	1	1

$$[A\bar{B}\cancel{C}(C+BD)+\bar{A}\bar{B}]C.$$

$$[A\bar{B}\cancel{C}C + \underbrace{A\bar{B}BCD}_{0} + \bar{A}\bar{B}]C.$$

$\bar{B}B = 0$

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

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

$C.C = C$

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

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

$$\bar{B}C(1)$$ Ans!

Date

min terms of SOP are represented by output(1).

Check logic diagram (SOP)

$(AB + AC)' + A'B'C$.

DeMorgan's law.

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

→ This form is also called as Minterms.

SUM OF PRODUCT :- (important).

STD SOP FORM.

either in complement or not
each term of SOP form must contain all the literals of our Domain.
① Domain {A,B,C}
$A + BC$ → wrong
$ABC + ABC$ → correct.

SOP - FORM (SUM OF PRODUCT) :

↳ standard / Canonical SOP? and its representation
↳ Mapping of SOP form to Truth table.

Examples:-

also called as minterms

→ 0's are represented by their complements

1 represents original.

	A	B	C	SOP FORM
m_0	0	0	0	$\bar{A}\bar{B}\bar{C}$
m_1	0	0	1	$\bar{A}\bar{B}C$
m_2	0	1	0	$\bar{A}B\bar{C}$
m_3	0	1	1	$\bar{A}BC$
m_4	1	0	0	$A\bar{B}\bar{C}$
m_5	1	0	1	$A\bar{B}C$
m_6	1	1	0	$AB\bar{C}$
m_7	1	1	1	$\bar{A}\bar{B}\bar{C}$

Date

Convert $A+BC$ into std. SOP Form.

$A+BC$ Domain $\{A, B, C\}$.

$A(\overbrace{B+\bar{B}}^{1}) + BC(\overbrace{A+\bar{A}}^{1})$ as multiply of 1 is always same.

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

$AB(\overbrace{C+\bar{C}}^{1}) + A\bar{B}(\overbrace{C+\bar{C}}^{1}) + ABC + \bar{A}BC$

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

Thus now in std state.

111 110 101 100 011

// now every term in std form will be one in TT and elsewise its 0.

TRUTH TABLE

	A	B	C	X
m0	0	0	0	0
m1	0	0	1	0
m2	0	1	0	0
m3	0	1	1	1
m4	1	0	0	1
m5	1	0	1	1
m6	1	1	0	1
m7	1	1	1	1

& Canonical form

Ans. $F(A,B,C) = \sum m(3,4,5,6,7)$

Check logic Diagram (POS Form)

This Form is also called as max terms

POS of max term represented by M → output (0)

PRODUCT OF SUM (POS FORM).

↳ When two or more sum terms are multiplied.

Example:

(i) $(A'+B)(A+B'+C)$

(ii) $A\ (A+B)$ (important case)

(iii) $(B+C)D$ (important case).

etc.

Convert $(A+B)(B+C)$ ' Domain $\{A, B, C\}$ ' into std ~~SOP~~ POS FORM.

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

let $C\bar{C}$ which is equal to zero add in first term

& $A\bar{A}$ which is " " " " 2nd "

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

By Associative law

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

∴ let

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

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

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

in POS form
$A \Rightarrow 0$ & $\bar{A}$ would be $\Rightarrow 1$.

Date

Now write the repeating terms first/once.

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

0 0 1   1 0 0   0 0 0 —— (a)

(Now in POS form the Outputs i.e (a) will be 0 in truth table and else wise 1. —— (b)

numbering starts from 0.

follow (b)

A	B	C	POS	output
0	0	0	$(M_0)\ A+B+C$	0
0	0	1	$(M_1)\ A+B+\bar{C}$	0
0	1	0	$(M_2)\ A+\bar{B}+C$	1
0	1	1	$(M_3)\ A+\bar{B}+\bar{C}$	1
1	0	0	$(M_4)\ \bar{A}+B+C$	0
1	0	1	$(M_5)\ \bar{A}+B+\bar{C}$	1
1	1	0	$(M_6)\ \bar{A}+\bar{B}+C$	1
1	1	1	$(M_7)\ \bar{A}+\bar{B}+\bar{C}$	1

Convert $(A+B)(C+D)$ into std form POS & SOP.
Domain : $\{A, B, C, D\}$.

let

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

Now by Associative law.

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

let

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

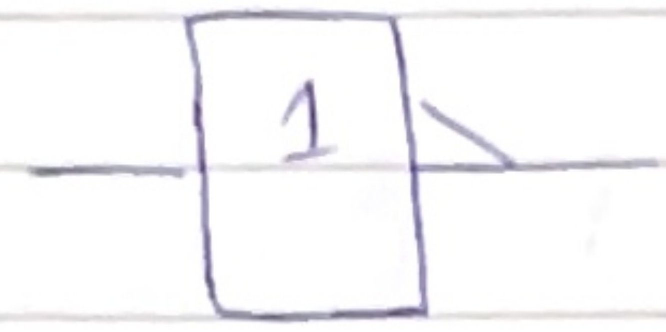
let.

Date

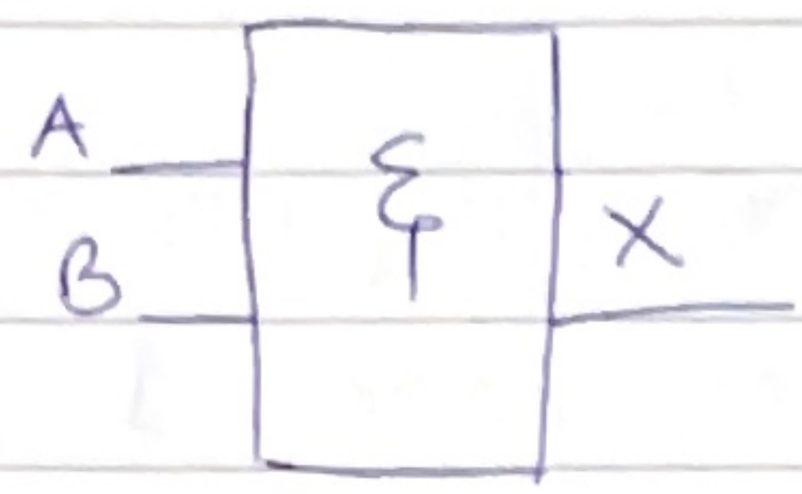
NOTE IMP. INFORMATION

CIRCUIT DIAGRAM:-

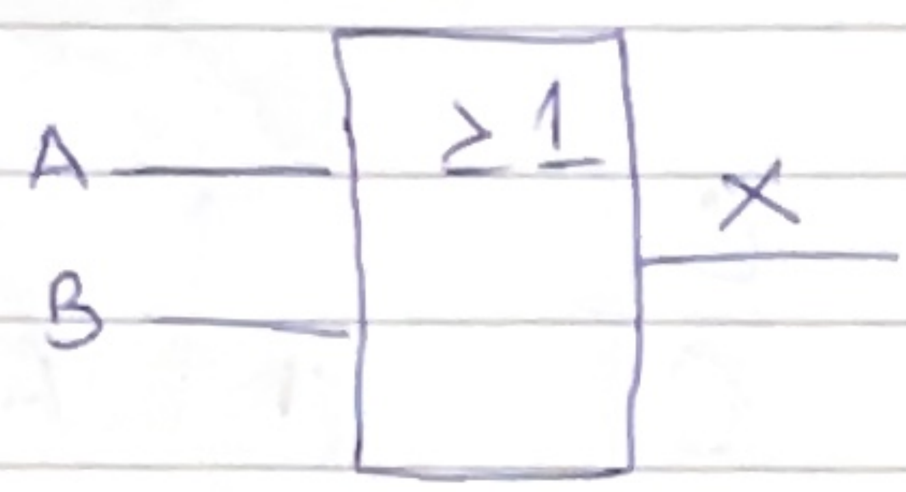
(i) NOT gate



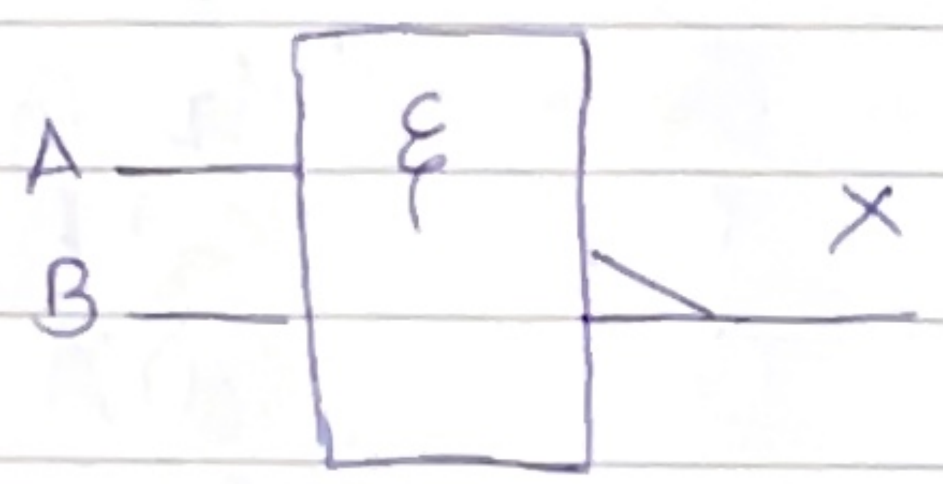
(ii) AND gate



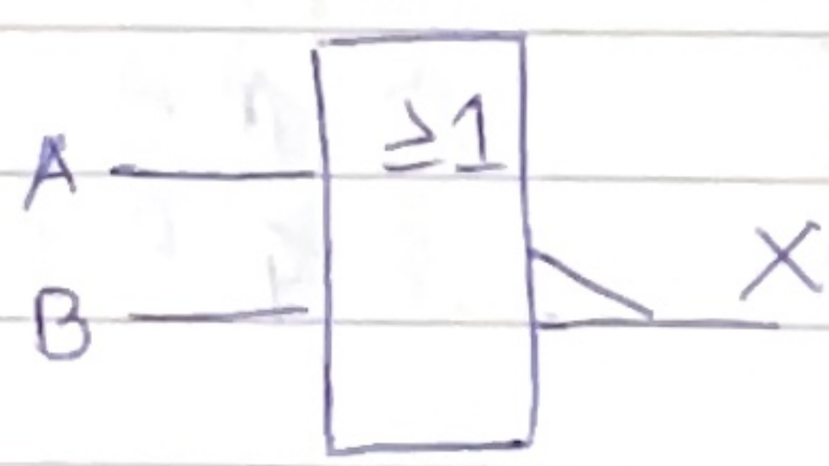
(iii) OR gate



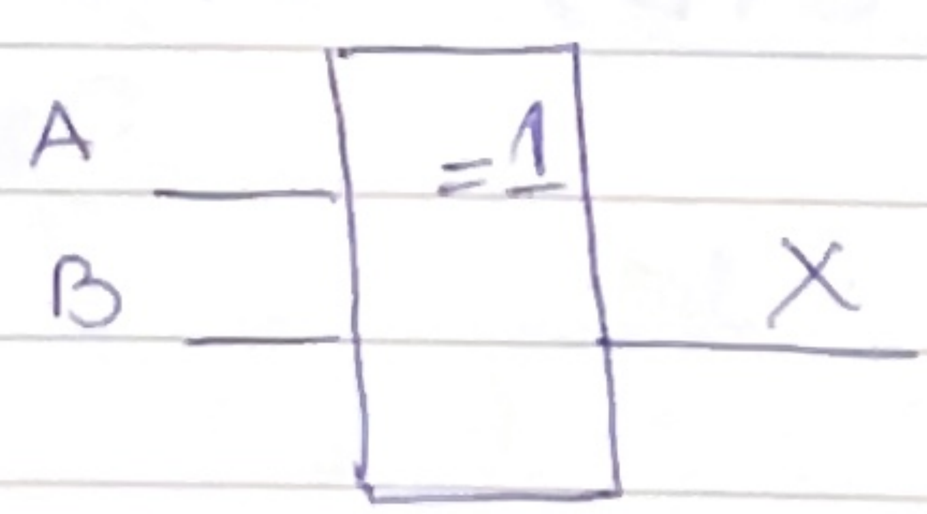
(iv) NAND gate



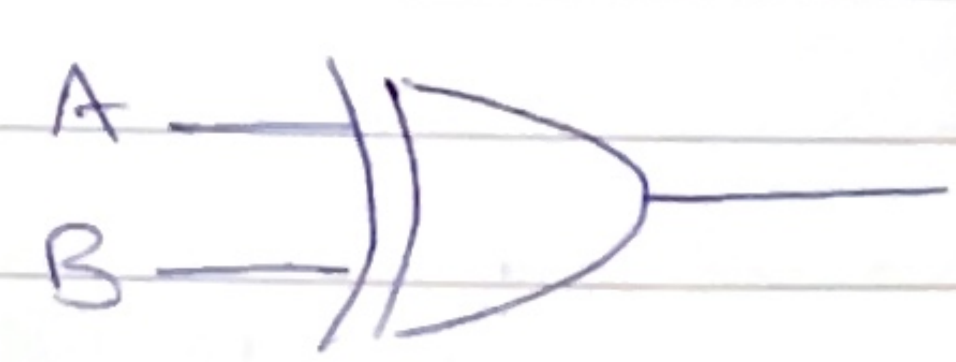
(v) NOR gate



(vi) XOR gate

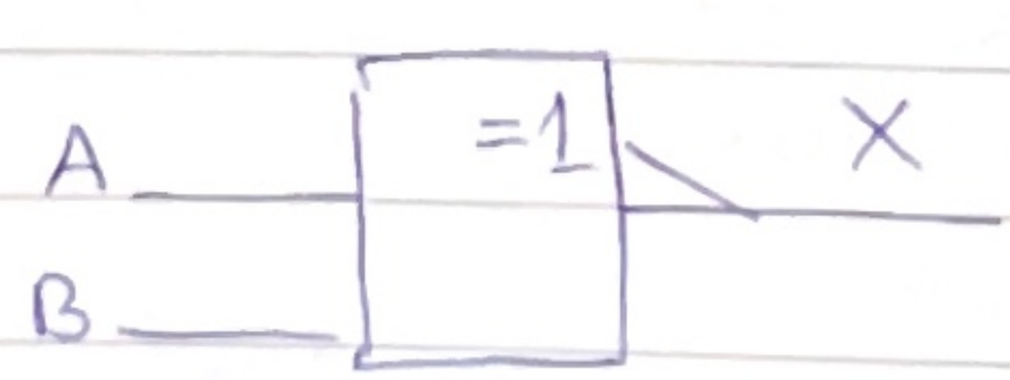


LOGIC DIAGRAM

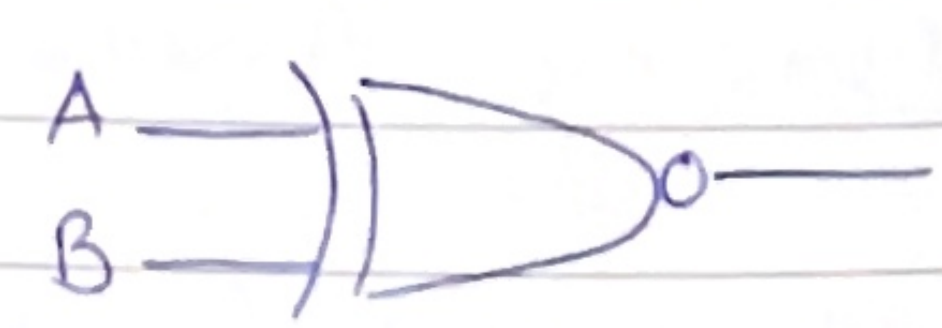


(vii) XNOR gate

(Bi-Conditional)



LOGIC DIAGRAM



Date

POS & SOP CONVERSION:-

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

Convert POS of $(A+B)(\bar{B}+C)$ in SOP form.

$\hookrightarrow$ (i)

(i) Convert given statement into std

(i)

$\Rightarrow (A+B+C) \quad (A+\bar{B}+C) \quad \cancel{+} \quad (\bar{A}+\bar{B}+\bar{C})$

↳ this the std form already.

(ii) FORM TRUTH Table

Sr no.	A	B	C	output	
0	0	0	0	0	→ POS
1	0	0	1	1	
2	0	1	0	0	→ POS
3	0	1	1	1	
4	1	0	0	1	
5	1	0	1	1	
6	1	1	0	1	
7	1	1	1	0	→ POS

(iii) Separate SOP & POS.

∵ the remaining are SOPs written.

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

(iv) Now writing the canonical form:-

POS $\quad F = \Pi M(0, 2, 7)$

SOP $\quad F = \Sigma m(1, 4, 3, 5, 6).$

$A.(B+C) \rightarrow POS$

$AB+AC \rightarrow SOP.$

Date

$1.A + AB \leftarrow SOP.$

DONT CARE CONDITION:-

It ignores the output irrespective of its state being 0 & 1.

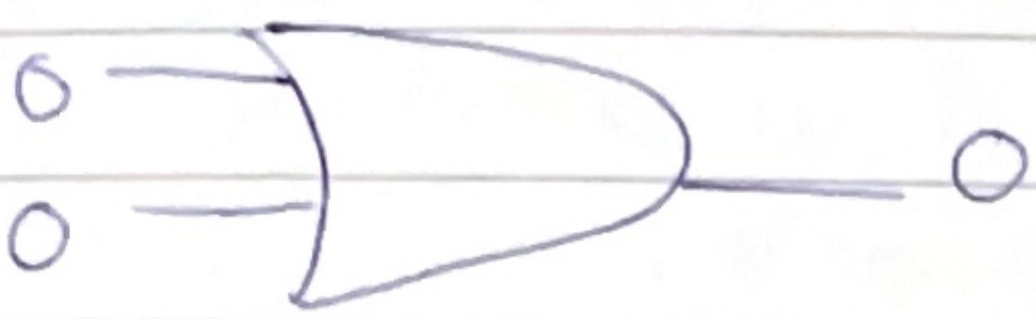
Example:-

(i) $F(A,B,C) = \Sigma m(1,2,3) + d(7,6)$

these term wouldn't be cared of ∴ their output is X.

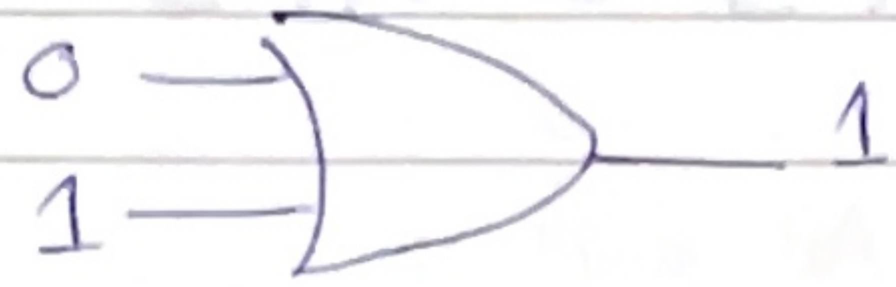
	A	B	C	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	~~0~~1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	X
7	1	1	1	X

} represents Dont care taken from canonical form.



(i)

$0+0=0$



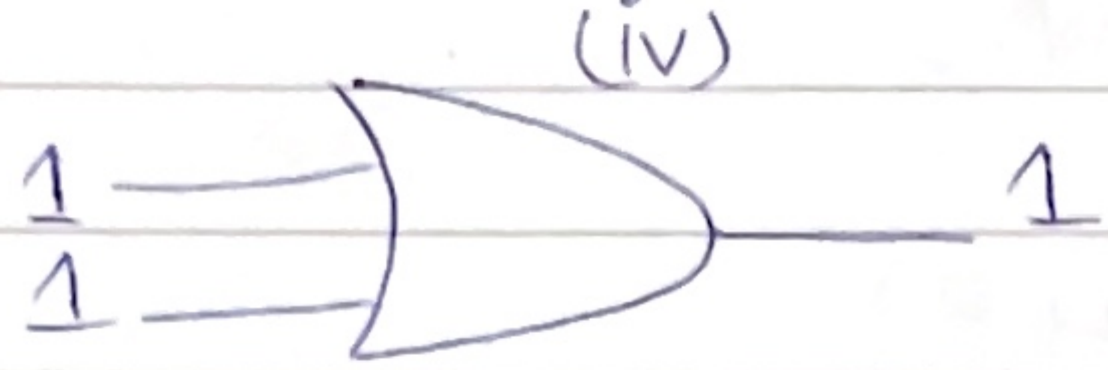
(ii)

$0+1=1$

(iii)

$1+0=1$.

(iv)



$1+1=1$.

LAWS & RULES OF BOOLEAN ALGEBRA.

(i) Commutative law.

$AB = BA$

$A+B = B+A$

(iii) Distributive law.

$AB+AC = A(B+C)$.

(ii) Associative law:

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

$A(BC) = (AB)C$

Rules:-

(i) $A+0 = A$
(ii) $A+1 = 1$
(iii) $A.0 = 0$
(iv) $A.1 = A$
(v) $A+A = A$
(vi) $A+\bar{A} = 1$
(vii) $A.A = A$
(viii) $\bar{\bar{A}} = A$
(ix) $A+AB = A$
(x) $A+\bar{A}B = A+B$
(xi) $A.\bar{A} = 0$
(xii) $(A+B)(\ +C) = A+BC$

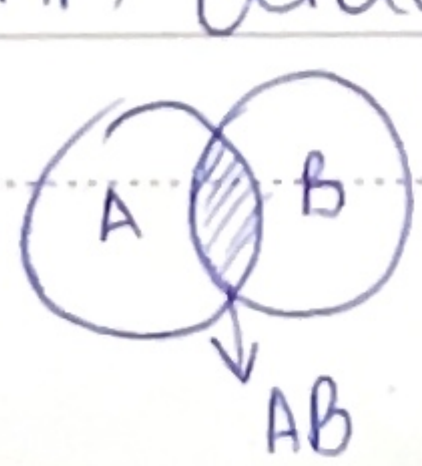
Proof $A+AB = A$ (draw both s logic diagram)

$\rightarrow$ taking common

$A(1+B) = A(1)$

$\Rightarrow A.$ $\hookrightarrow$ Rule 4

also can be proved using a truth table and venn diagram.



$\therefore A+AB = A$.

Date

Proof. $A + A'B = A + B$. (- prove by truth table and draw both logic diagrams)

By Rule : $A + AB = A$

AB

$\Rightarrow AA + \cancel{AB} + \bar{A}B \Rightarrow$ By Rule $A \cdot A = A$.
By Rule $A\bar{A} = 0$

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

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

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

↳ Rule $A + \bar{A} = 1$

$= 1 \cdot (A+B)$

↳ Rule $A \cdot 1 = 1?$

$= (A+B)$ proved

Proof. $(A+B)(A+C) = A + BC$. (" ")

$AA + AC + AB + CB$.

$A + AC + AB + BC$

→ $A \cdot 1 = A$. Rule

$A(1+C) + AB + BC$

↳ $1 + C = 1$. Rule.

$A + AB + BC$

$A(1+B) + BC$

} similar rules applied again.

$A + BC$. hence proved

Slide no. 29 contain imp questions regarding boolean proof (using B. Algebra).

Date

→ **Demorgan's law** (Draw T.T & logic Diagrams)

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

NAND Negative-OR (also $(XY)' = X' + Y'$)

2nd. $\overline{A+B} = \bar{A} . \bar{B}$ (Draw T.T & logic Diagram)

NOR Negative AND.

also $(X+Y)' = X' . Y'$

* combinational logical circuits are formed by combining multiple gates.

K-Maps:-

The complexity of the digital logic gates that implement a Boolean function is directly related to the complexity of the algebric expression from which the function is implemented.

∴ The mapping Method will minimize the boolean functions.

⇒ Regarded as the picterical form of the TT. known as the karnaugh-map / k-map.