Unit-4.

Number System Representation.

- **Number System**

Number System is a basis of various countings.

- **Radix or Base of a number system.**

Base or Radix of a number system is the number of symbols/variables used in its representation.

For example :- A decimal number system has 10 symbols/10 variables.

0, 1, 2, 3, 4, 5, 6, 7, 8, 9. → Its Base or radix is 10.

Similarly, Binary no. system has 2 symbols.

0, 1 → Its base or radix is 2.

And octal has 8 symbols, and its base or radix is 8 → 0, 1, 2, 3, 4, 5, 6, 7

→ Types of Number system.

1. Decimal Number System
2. Binary Number System
3. Octal Number System
4. Hexadecimal Number system.

1. Decimal Number System.

It has 10 symbols or digits

0, 1, 2, 3, 4, 5, 6, 7, 8, 9 Radix = 10

(1) $(\overset{2}{7}\overset{1}{2}\overset{0}{4})_{10}$

$\Rightarrow 7 \times 10^2 + 2 \times 10^1 + 4 \times 10^0$

(ii) $(\overset{2\ 1\ 0\ \cdot\ 1\ 2}{724.36})_{10}$

$\Rightarrow 7\times10^2+2\times10^1+4\times10^0\,(\cdot)\ 3\times10^{-1}+6\times10^{-2}$

2. Binary Number system.

It has 2 symbols or digits

0, 1 , Radix = 2

(i)

2^4	2^3	2^2	2^1	2^0
16	8	4	2	1

$\Rightarrow (\overset{2^2 2^1 2^0}{101})_2$

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

$= 4+0+1 = 5$

(ii) $(\overset{2^3\ 2^2\ 2^1\ 2^0}{1\ 1\ 0\ 1})_2$

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

$\Rightarrow 8+4+0+1 = 13$

3. Octal Number System.

It has 8 Symbols or digits.

0,1,2,3,4,5,6,7 , Radix = 8.

(i) $(\overset{2\ 1\ 0}{2\ 4\ 6})_8$

$\Rightarrow 2\times8^2+4\times8^1+6\times8^0$

$= 2\times64+32+6$

$= 128+32+6 = 160+6 = (166)_{10}$

4. Hexadecimal Number system

It has 16 Symbols or digits

0,1,2,3,4,5,6,7,8,9, A, B, C, D, E, F , Radix = 16

(v) $\overset{1\ 0}{(21)_{16}}$

$\Rightarrow 2\times16^1 + 1\times16^0$
$= 32+1 = 33$

Conversion of Number System.

1. Decimal to Binary
2. Decimal to octal
3. Decimal to hexadecimal
4. Binary to Decimal
5. Binary to octal
6. Binary to hexadecimal
~~7. Octal to Decimal~~
~~8. Octal to Binary~~
7. Octal to hexadecimal
8. Hexadecimal to ~~Decimal~~ Octal.

1. Decimal to binary number system.

$\rightarrow (18)_{10} \rightarrow (10010)_2$.

Decimal to any other number systems [divide the given number/digits with base of other number system].

2	18	0
2	9	1
2	4	0
2	2	0
	1	

$\Rightarrow (10010)_2$

→ $(18.18)_{10} = (10010.0010)_2$

	Real (integer)	fraction	Real ↓ fraction
$.18 \times 2$	0	.36	0.36
$.36 \times 2$	0	.72	0.72
$.72 \times 2$	1	.44	1.44
$.44 \times 2$	0	.88	0.88

(maximum 3 ya 4 tak lena है) in fraction (approx).

(ii) Decimal to octal :-
Divide decimal number with 8.

→ $(387)_{10} \rightarrow (603)_8$

8	387	3
8	48	0
	6	

→ $(783.38)_{10} \rightarrow (1417.3024)_8$

8	783	7
8	97	1
8	12	4
	1	

	Real	fraction
$.38 \times 8$	3	.04
$.04 \times 8$	0	.32
$.32 \times 8$	2	.56
$.56 \times 8$	4	.48

(iii) Decimal to Hexadecimal
Divide decimal number by 16

→ $(987)_{10} \longrightarrow (3DB)_{16}$

$$\begin{array}{r|l|l}
16 & 987 & 11 \uparrow (B) \\
16 & 61 & 13 \ (D) \\
 & 3 &
\end{array}$$

→ $(357.35)_{10} \rightarrow (165.599)_{16}$

$$\begin{array}{r|l|l}
16 & 357 & 5 \uparrow \\
16 & 22 & 6 \\
 & 1 &
\end{array}$$

	Real	fraction
$.35 \times 16$	5	.60
$.60 \times 16$	9	.60
$.60 \times 16$	9	.60

(iv) Binary to Decimal:
Every digit will be multiplied with 2^n.
$n = 0, 1, 2, 3, \ldots n$

→ $(11011)_2 = (27)_{10}$

$$\begin{array}{ccccc}
4 & 3 & 2 & 1 & 0 \\
1 & 1 & 0 & 1 & 1
\end{array}$$

$\Rightarrow 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$
$= 16 + 8 + 0 + 2 + 1$
$= 27$

→ $(10110.10)_2 = (\ 22.5\ )_{10}$

$\overleftarrow{10110}^{43210}.\overrightarrow{10}$

~~$1\times10^4 + 0\times10^3 + 1\times10^2 + 1\times10^1 + 0\times10^0$~~

$= 1\times2^4 + 0\times2^3 + 1\times2^2 + 1\times2^1 + 0\times2^0 (.) 1\times2^{-1} + 0\times2^{-2}$

$= 16 + 0 + 4 + 2 + 0 . 5$

$= 22.5$

~~$.10$~~

~~$= 1\times2^{-1} + 0\times2^{-2}$~~

~~$= 0.5 + 0$~~

~~$= 0.5$~~

(V) Binary to Octal :-

~~Every digit will be multip~~

Grouped all the binary number in group of 3 from LSB to MSB

least significant bit to most significant ~~bit~~ .

→ $(1001010)_2 = (\ 112\ )_8$

MSB ← LSB

$\underbrace{\overset{2^2}{0}\overset{2^1}{0}\overset{2^0}{1}}_{1}\ \underbrace{\overset{2^2}{0}\overset{2^1}{0}\overset{2^0}{1}}_{1}\ \underbrace{\overset{2^2}{0}\overset{2^1}{1}\overset{2^0}{0}}_{2}$

(VI) Binary to Hexadecimal.

Grouped all the binary number in group of 4 from LSB to MSB

→ $(1001100101)_2 \rightarrow (\ 265\ )_{16}$

$$\overset{2^3}{0}\overset{2^2}{0}\overset{2^1}{1}\overset{2^0}{0}\;\overset{2^3}{0}\overset{2^2}{1}\overset{2^1}{1}\overset{2^0}{0}\;\overset{2^3}{0}\overset{2^2}{1}\overset{2^1}{0}\overset{2^0}{1}$$

$$2 \qquad 4+2 \qquad 4+1$$

$$2 \qquad 6 \qquad 5$$

(VII) Octal to Decimal

In this case we multiply each bits by 8^n, $n = 0, 1, 2, 3, \ldots$

→ $(475.25)_8 \rightarrow (317.328125)_{10}$

→ $\overset{2}{4}\,\overset{1}{7}\,\overset{0}{5}\,.\,\overset{1}{2}\,\overset{2}{5}$

$4\times8^2 + 7\times8^1 + 5\times8^0 (+) 2\times8^{-1} + 5\times8^{-2}$

$= 256 + 56 + 5 + 0.25 + 0.078125$

$= (317.328125)_{10}$

(VIII) Hexadecimal to Decimal

Multiply each digits by 16^n, $n = 0, 1, 2, 3, \ldots$

→ $(9B2.1A)_{16} \rightarrow (2482.1015)_{10}$

$9\;B\;2\;.\;1\;A$

$\Rightarrow \overset{2}{9}\;\overset{1}{11}\;\overset{0}{2}\;.\;\overset{1}{1}\;\overset{2}{10}$

$\Rightarrow 9\times16^2 + 11\times16^1 + 2\times16^0 + 1\times16^{-1} + 10\times16^{-2}$

$\Rightarrow 2304 + 176 + 2 + 0.0625 + 0.0390$

$\Rightarrow (2482.1015)_{10}$

(IX) Octal to Hexadecimal. → first step → Octal to binary
→ second step → Binary to hexadecimal.

→ $(615)_8 \rightarrow (\ \ 18D\ \ )_{16}$.

Octal to binary :- 6 1 5
110 001 101

Binary to Hexadecimal

3 2 1 0

$\underbrace{0001}\ \underbrace{1000}\ \underbrace{1101}$

1 8 13
1 8 D

$(18D)_{16}$.

(X) Hexadecimal to octal. → First step → Hexa to binary
→ Second step → Binary to Octal.

$(25B)_H \rightarrow (\ \ 1133\ \ )_8$

2 5 B
0010 0101 1011

16 8 4 2 1

Binary to Octal

$\underbrace{001}\,\underbrace{001}\,\underbrace{011}\,\underbrace{011}$

1 1 3 3

$(1133)_8$

⇒ Any Base system to decimal : Multiply with $Base^n$.

⇒ Decimal to Any Base System : Divide by $Base^n$.

Q. Determine the value of x if ;

(i) $(211)_x = (152)_8$

(ii) $(193)_x = (623)_8$ Ans → 16

(i) $(211)_x = (152)_8$

$2 \times x^2 + 1 \times x^1 + 1 \times x^0 = 1 \times 8^2 + 5 \times 8^1 + 2 \times 8^0$

$2x^2 + x + 1 = 64 + 40 + 2$

$2x^2 + x = 106$

$2x^2 + x - 106 + 1 = 0$ ⇒ $2x^2 + x - 105 = 0$

$\boxed{x = 7}$

(ii) $(193)_x = (623)_8$

$1 \times x^2 + 9 \times x^1 + 3 \times x^0 = 6 \times 8^2 + 2 \times 8^1 + 3 \times 8^0$

$x^2 + 9x + 3 = 384 + 16 + 3$

$x^2 + 9x - 400 = 0$

$\boxed{x = 16}$

Binary Arithmetic's

(I) → Binary Addition

		Addition	Carry
0 + 0	=	0	0
0 + 1	=	1	0
1 + 0	=	1	0
1 + 1	=	0	1

(II) → Subtraction

		Sub	Borrow
0 − 0	=	0	0
0 − 1	=	1	1
1 − 0	=	1	0
1 − 1	=	0	0

(i) Add $(1010)_2$ and $(0011)_2$.

Binary

```
    1
  1 0 1 0
+ 0 0 1 1
---------
  1 1 0 1
```

Decimal = 10
+3
13

(ii) Add 28 and 15 in binary

→ Convert 28 into binary.

11100

32 16 8 4 2 1
1 1 1 0 0
1 0 1 0 1 1

15 into binary

1111

Decimal = 43

```
    1 1 0 0
⇒   1 1 1 0 0   Binary.
  + 0 1 1 1 1
  -----------
  1 0 1 0 1 1
```

→ Subtract $(11011)_2$ and $(10110)_2$

$$\begin{array}{r} 11011 \\ -\ 10110 \\ \hline 000101 \end{array}$$

$$\begin{array}{r} 27 \\ 22 \\ \hline 5 \end{array}$$

→ Subtract 38 and 29 in binary.

38 in binary = 100110
29 in binary = 11101

$$\begin{array}{r} 100110 \\ -\ 011101 \\ \hline 001001 \end{array}$$

$$\begin{array}{r} 38 \\ 29 \\ \hline 9 \end{array}$$

Complements and Binary subtraction using 1's and 2's.

Complements :-

1. There are two types of complements :
(i) r's complement
(ii) (r-1)'s complement

(1) r's complement

$r^n - N$

r = radix or base
N = given number.
n = no. of ~~dt~~ integer digits

Q→ Find the 10's complement of given number.
(a) $(37218)_{10}$
(b) $(0.12345)_{10}$

(a) $N = 37218$

$n = 5$

$r = 10$

$\rightarrow r^n - N$

$= 10^5 - 37218$

$= 62782$

(b) $N = 0.12345$

$n = 0$

$r = 10$

$\rightarrow r^n - N$

$= 10^0 - 0.12345$

$= 1 - 0.12345$

$= 0.87655$

Q. Find 2's complement

(a) $(1101.01)_2$

(b) $(0.0011)_2$

(a) $N = 1101.01$

$n = 4$

$r = 2$

$\rightarrow r^n - N$

$= 2^4 - 1101.01$

$= 16 - 1101.01$

$= 10000 - 1101.01$

$$\begin{array}{r} 10000.00 \\ -\ 1101.01 \\ \hline 00010.11 \\ \hline \end{array}$$

$= 10.11$

(b) $N = 0.0011$, $n = 0$

$r = 2$

$\rightarrow r^n - N$

$= 2^0 - 0.0011$

$= 1 - 0.0011$

$= 1.0000 - 0.0011$

$$\begin{array}{r} 1.0000 \\ -\ 0.0011 \\ \hline 0.1101 \end{array}$$

$= 0.1101$

(2) (r-1)'s complement

$(r^n - 1) - N \rightarrow$ without fraction

$(r^n - r^{-m}) - N \rightarrow$ m is fraction digits

Q Find 9's complement of
(a) $(37218)_{10}$ (b) $(0.12345)_{10}$

(a) $N = 37218$
$n = 5$
$r = 10$

9's complement $= (r^n - 1) - N$
$= (10^5 - 1) - 37218$
$= 99999 - 37218$
$= 62781$

(b) $N = 0.12345$
$n = 0$
$r = 10$
$m = 5$
9's complement $= (r^n - r^{-m}) - N$
$= 10^0 - 10^{-5} - 0.12345$
$= (1 - 10^{-5}) - 0.12345$
$= 0.99999 - 0.12345$
$= 0.87654$

Q. Find 1's complement
(a) $(10110)_2$ (b) $(0.0110)_2$

(a) $N = 10110$

$n = 5$

$r = 2$

1's complement $= (r^n - 1) - N$

$= (2^5 - 1) - 10110$

$= (32 - 1) - 10110$

$= 31 - 10110$

$= 11111 - 10110$

$= 01001$

$$\begin{array}{r} 11111 \\ -\ 10110 \\ \hline 01001 \end{array}$$

Shortcut.

(1) 1's complement.

invert binary digits ($1 \rightarrow 0$ or $0 \rightarrow 1$)

$10110 \rightarrow 01001$ Ans//.

(2) 2's complement. (1's + 1).

$10110 \rightarrow$ 1's $\Rightarrow$ $$\begin{array}{r} 01001 \\ +\ \ \ \ \ 1 \\ \hline 01010 \end{array}$$ Ans.

(3) 9's complement :-

Subtract all digits from 9.

$37218 \rightarrow$ $$\begin{array}{r} 99999 \\ 37218 \\ \hline 62781 \end{array}$$

(4) 10's Complement (9's + 1).

37218 → 9's ⇒ 62781

$$
\begin{array}{r}
62781 \\
+ \quad\quad 1 \\
\hline
62782
\end{array}
$$

62782 // Ans.

Binary Subtraction using 1's and 2's complement.

(a) Using 1's complement.

→ If a binary digit $(B)_2$ is subtracted from $(A)_2$, then the following ~~operation~~ steps to be followed.

(i) 1st obtain the 1's complement of the number to be subtracted $(B)_2$

(ii) Add A with 1's complement of B, using binary operation

(iii) If final carry (1) is generated, then the result will be positive and carry will be added into the result.

(iv) If no carry generated (Final carry (0)), the result will be negative and it ~~should~~ will be inverted (1's complement)

Q. Subtract $(1011)_2$ from $(0100)_2$ using 1's complement.

→ A = 0100 B = 1011

⇒ B' = 1's complement of B = 0100

⇒ Add A with B'

```
  0100
+ 0100
  1000
```

final carry (0) → Result is negative

→ Find 1's complement = 0111 (−7)

Q. Subtract $(4)_{10}$ from $(9)_{10}$ using 1's complement.

→ A = 9 ⇒ binary = 1001

B = 4 ⇒ binary = 0100

→ B' → 1's complement of B = 1011

Add A and B',

$$\begin{array}{r} \scriptstyle 0\,1\,1 \\ 1001 \\ 1011 \\ \hline \boxed{1}0100 \end{array}$$

final carry (1) → Result positive.

$$\begin{array}{r} 0100 \\ +\ 1 \\ \hline 0101 \end{array}$$ Ans.

(b) Binary Subtraction using 2's complement.

→ If a number $(B)_2$ is subtracted from $(A)_2$ then the following steps to be followed to perform subtraction using 2's complement.

(i) Take 2's complement of number to be subtracted $(B)_2$

(ii) Add $(A)_2$ with 2's complement of $(B)_2$ using binary addition,

(iii) If final carry is (1) then the result will be positive, the final carry will be discarded.

(iv) If final carry is (0), then the result will be negative, and take 2's complement.

Q Perform $(9)_{10} - (5)_{10}$ using 2's complement.

→ A = 9 → Binary = 1001
B = 5 → Binary = 0101

→ B' = 2's complement of B ⇒ 1010

$$\begin{array}{r} 1010 \\ +\ \ 1 \\ \hline 1011 \end{array}$$

→ Add A and B'

$$\begin{array}{r} 1001 \\ +\ 1011 \\ \hline \boxed{1}0100 \end{array}$$

final carry → 1 , Result positive.

→ final result is <u>0100</u>.

Q Perform $(20)_{10} - (30)_{10}$ using 2's complement.

A = 20 → Binary ⇒ 10100
B = 30 → Binary ⇒ 11110

Take B' → 2's complement of B ⇒

$$\begin{array}{r} 00001 \\ +\ \ \ \ 1 \\ \hline 00010 \end{array}$$

→ Add A and B'

$$\begin{array}{r} 10100 \\ +\ 00010 \\ \hline 10110 \end{array}$$

final carry → 0
Result → negative.

2's complement of result (10110) ⇒

$$\begin{array}{r} 01001 \\ +\ \ \ \ 1 \\ \hline -\ 01010 \end{array}$$

// Final answer.

LOGIC GATES.

→ The electronics or digital circuits which have ability to make logical decisions (logical operations) by producing output under certain input conditions are commonly known as Logic gates.

→ Logic operators.

→ To solve or simplify the logical expression or logical operation/ binary operation using digital circuits is called logic operators.

→ There are mainly three logic operators -

(i) NOT operator

(ii) AND operator

(iii) OR operator.

→ Types of Logic gates

Logic gates

- Basic gate
  - (i) NOT
  - (ii) AND
  - (iii) OR
- universal gates
  - (i) NAND
  - (ii) NOR
- Special purpose gates
  - (i) EX-OR / XOR
  - (ii) EX-NOR / XNOR

(i) Basic Gates :

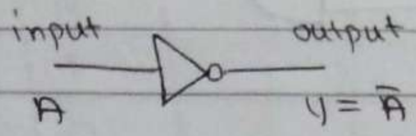
(1) NOT Gate :

→ The logical circuits which perform NOT operation.

→ The NOT operator represents a logical inversion or complementing.

→ The NOT operator is denoted by a bar ($\bar{}$) over the variable to be inverted.

Symbol :-



Truth table : A table that lists all the possible combinations of input variables and corresponding output is known as Truth table.

Input A	Output y
0	1
1	0

Boolean expression :

The Boolean expression of ~~not~~ NOT gate is represented as

$\boxed{y = \bar{A}}$ y = output

A = input

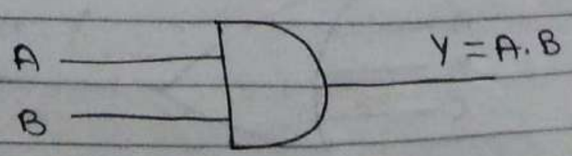
(2) AND Gate.

→ It performs AND operation, where AND operation represents logical multiplication.

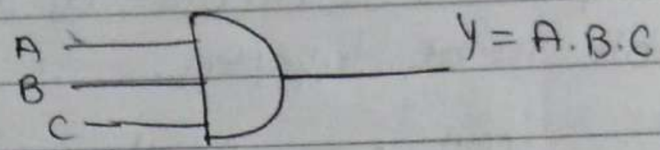
→ It is denoted by dot (•) between variables to be multiplied.

$$A \cdot B \Rightarrow A \text{ and } B$$

Symbol :-



For 3 inputs:-



Truth table:

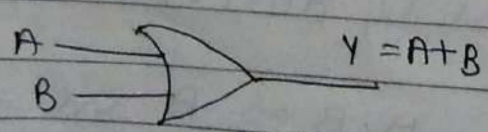
Inputs		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

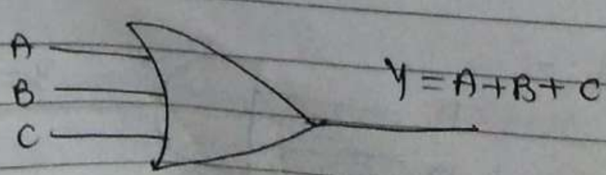
Boolean Expression:
The boolean expression of AND gate is given by

$Y = A.B$ → Y = output
A and B = inputs.

(3) OR Gate.
→ It performs OR operation, that is logical addition.
→ It is denoted by (+) sign between variables to be added.

Symbol:-





Truth table.

Inputs A	B	Output Y
0	0	0
0	1	1
1	0	1
1	1	1

Boolean Expression:-

The boolean expression of OR Gate is given by

$Y = A + B$ => Y = output
A & B = inputs

Universal Gate.

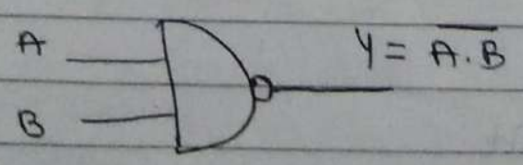
(1) NAND Gate.

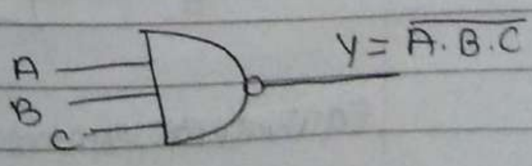
→ The term NAND can be splitted as NOT-AND which means that the NAND operation can be implemented with the combination of AND gate and NOT gate.

→ Thus a NAND gate is equivalent to an AND gate followed by NOT gate.

→ A NAND is also known as Universal gate because all the gates can be implemented using NAND gate.

Symbols





Truth table:

Inputs		Output
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Boolean Expression.

The Boolean Expression of NAND is given by

$Y = \overline{A \cdot B}$, Y = output
A & B = input.

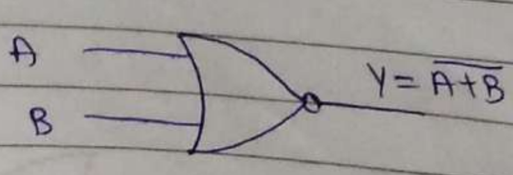
(II) NOR Gate.

→ The term NOR can be splitted as NOT and OR Gate which means that NOR operation can be implemented with the combination of OR Gate and NOT Gate.

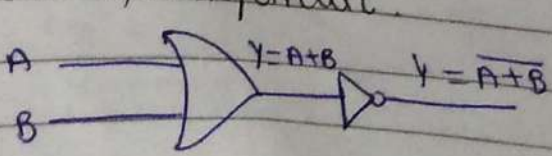
→ Thus, a NOR gate is equivalent to an OR Gate followed by NOT gate.

→ This also a type of universal gate.

Symbols.



Equivalent symbols/circuit.



Page No. | Date

Truth table:

Input		Output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Boolean Expression.

The Boolean Expression can be given as

$Y = \overline{A+B}$ $\Rightarrow y =$ output

$A \& B =$ input.

Special purpose gate

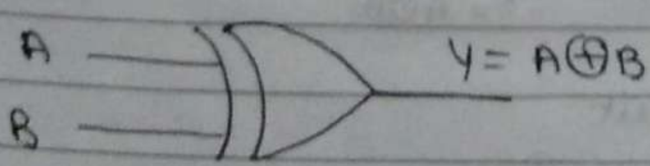
(1) Ex-OR Gate.

→ The exclusive-OR gate is abbreviated as EX-OR gate or X-OR gate.

→ It performs EX-OR operation which is denoted by $\oplus$ between input variables.

→ When both the inputs are same $(A = B)$, then output is low otherwise high.

Symbols:



Truth Table :

Input		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	0	0

Boolean Expression:

The Boolean expression of EX-OR Gate is given by

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

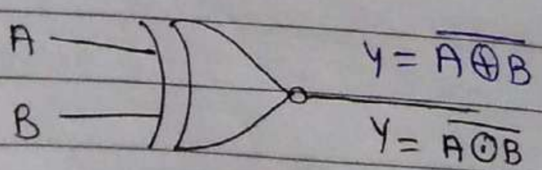
(2) EX-NOR Gate.

→ The Exclusive-NOR gate is abbreviated as EX-NOR or X-NOR Gate.

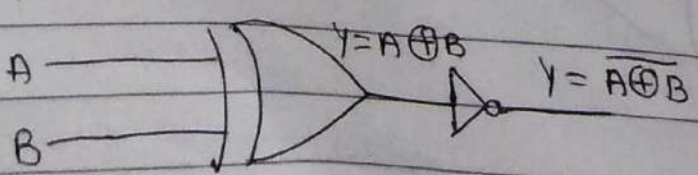
→ EX-NOR gate is equivalent to EX-OR gate followed by NOT Gate.

→ When both the inputs are same (A=B), The output will be high, otherwise low.

Symbols :



→ Equivalent circuit.



Truth table.

Inputs		Outputs
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Boolean Expression.

The Boolean Expression of EX-NOR Gate is given by

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

Imp * NAND Gate as an Universal Gate.

→ We know that the NAND Gate is a universal gate using which all other gates can be implemented

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

1. NOT gate using NAND gate.

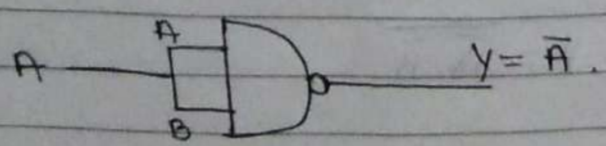
→ The Boolean expression of NAND gate is

$$Y = \overline{A.B} \quad \text{— (1)}$$

if A=B, then eq(1) becomes

$$Y = \overline{A.A}$$

Hence, $\boxed{Y = \bar{A}}$ But, $A.A = A$



(2) AND gate using NAND gate.

→ Boolean expression of NAND gate is

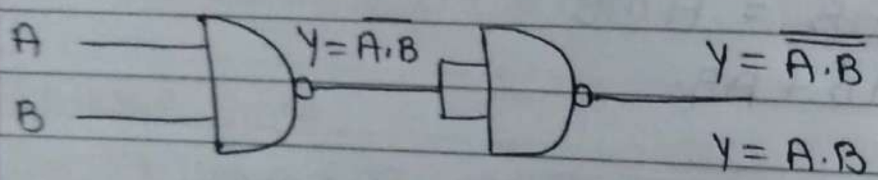
$$Y = \overline{A \cdot B} \quad \text{— (1)}$$

Boolean expression of AND gate is

$$Y = A . B \quad \text{— (2)}$$

Take the double inversion of RHS of eq (2)

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



(3) OR Gate using NAND gate.

→ Boolean expression of OR gate is

$$Y = A + B \quad \text{— (1)}$$

Take double inversion of RHS of eq (1)

$$Y = \overline{\overline{A + B}} \quad \text{— (2)}$$

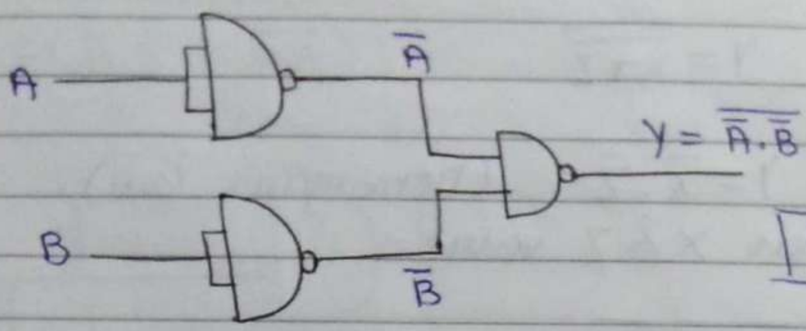
• From De-Morgan's theorem

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

eq (2) becomes

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

Hint :- double inversion in conversion of gate.



$\boxed{Y = A + B}$

4.# NOR Gate using NAND Gate.

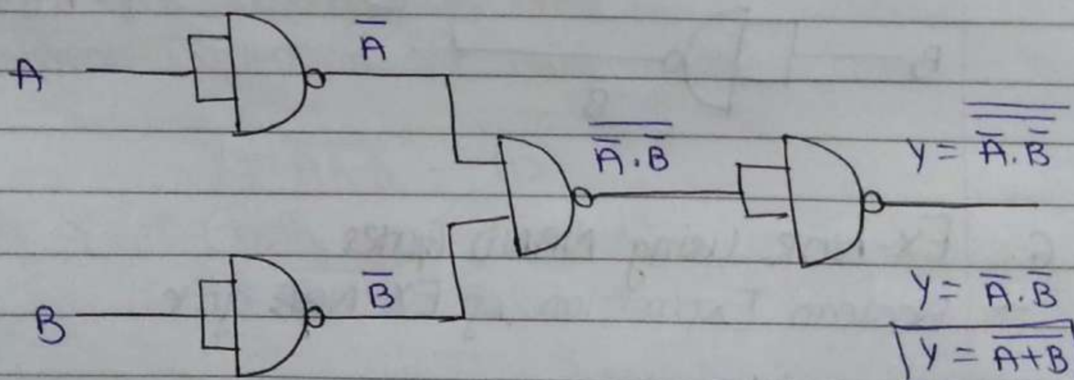
→ Boolean expression of NOR Gate is

$$Y = \overline{A+B} \quad \text{— (1)}$$

→ Take the double inversion of RHS of eq (1)

$$Y = \overline{\overline{\overline{A+B}}} \quad \text{— (2)}$$

$$Y = \overline{\overline{\bar{A} \cdot \bar{B}}} \quad \text{— (3)}$$



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

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

5. EX-OR using NAND Gate.

→ Boolean Expression for EX-OR Gate is

$Y = A \oplus B = \bar{A}B + A\bar{B}$ — (1)

$Y = X + Z$ — (2) ($X = \bar{A}B$, $Z = A\bar{B}$)

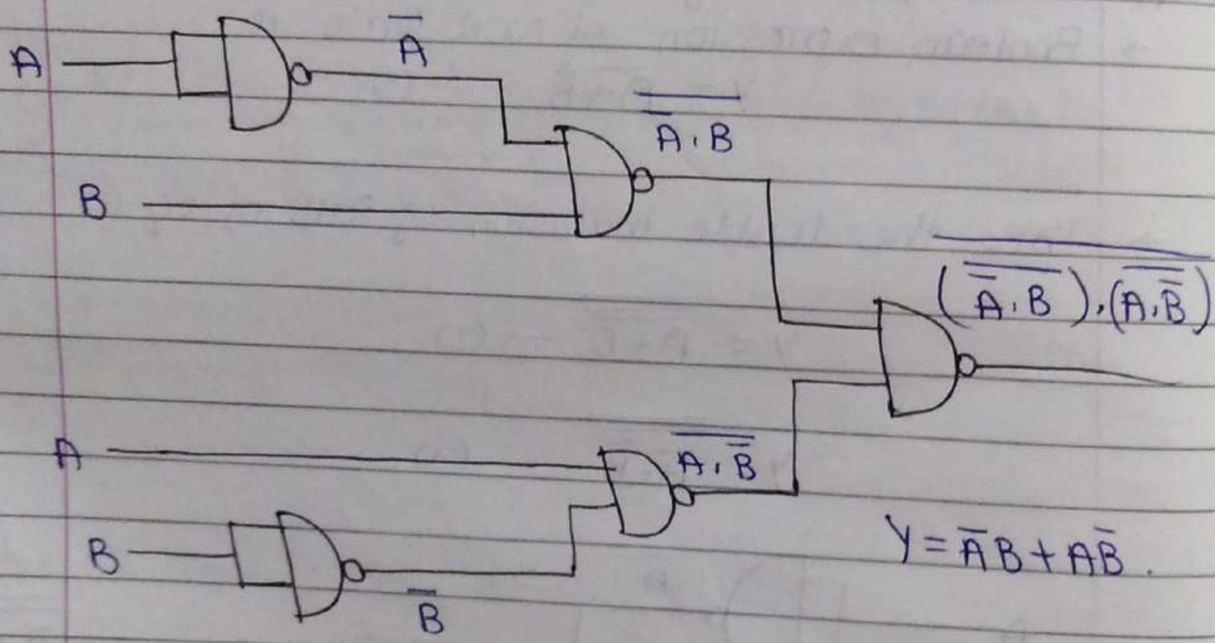
→ Take double inversion of RHS of eq(2)

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

$$Y = \overline{\overline{X} \cdot \overline{Z}} \quad \text{(Demorgan's law)}.$$

put X & Z value

$$Y = \overline{\overline{\overline{A}B} \cdot \overline{A\overline{B}}} \quad \text{—— (3)}$$



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

6. EX-NOR using NAND Gates.

→ Boolean Expression of EX-NOR gate.

$$Y = \overline{A}\overline{B} + AB \quad \text{—— (1)}$$

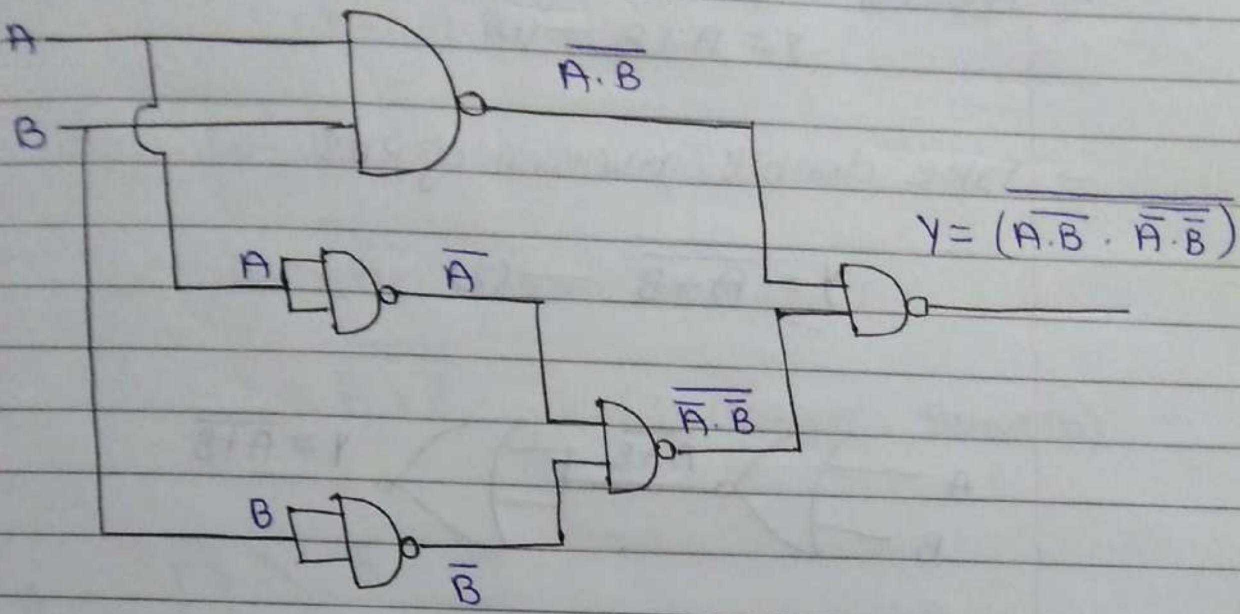
$$Y = X + Z \text{ — (2)} \quad \left(\begin{array}{l} X = \overline{A} \cdot \overline{B} \\ Z = AB \end{array} \right)$$

Take double inversion of RHS in eq (2)

$$Y = \overline{\overline{X+Z}} \quad \text{— (3)}$$

$$Y = \overline{\overline{X} \cdot \overline{Z}} \quad \text{(De-Morgan's theorem)}.$$

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



NOR as Universal Gate.

1. NOT Gate using NOR gate.

→ Boolean Expression for NOR Gate.

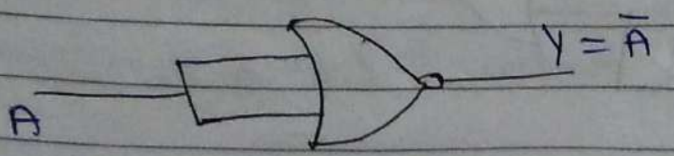
$$Y = \overline{A+B} \quad — (1)$$

if A = B (both input sorted)

$$Y = \overline{A+A} \quad — (2)$$

We know that $A+A = A$, eq^n (2) becomes

$$\boxed{Y = \bar{A}}$$



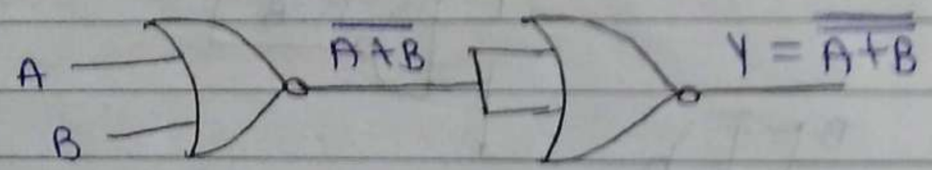
2. OR Gate using NOR Gate.

→ Boolean Expression of OR Gate.

$$Y = A + B \quad —(1)$$

→ Take double inversion of RHS.

$$Y = \overline{\overline{A+B}} \quad —(2)$$



3. AND Gate using NOR gate.

→ Boolean expression of AND gate.

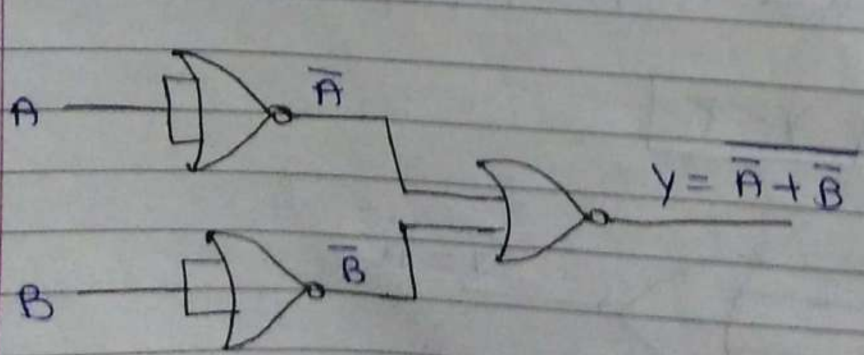
$$Y = A.B \quad —(1)$$

→ Take double inversion of RHS.

$$Y = \overline{\overline{A.B}} \quad —(2)$$

From De-Morgan's theorem $\overline{A.B} = \overline{A} + \overline{B}$.

$$Y = \overline{\overline{A} + \overline{B}} \quad —(3)$$



4. NAND gate using only NOR.

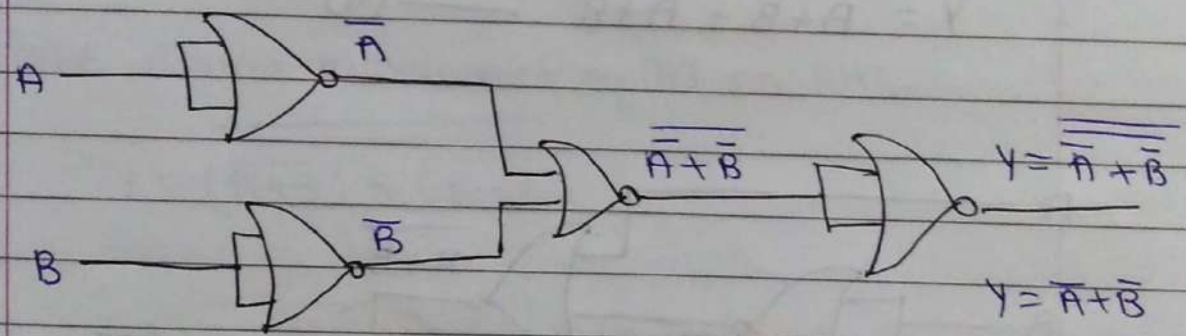
→ ~~Express~~ Boolean Expression for NAND gate.

$$Y = \overline{A \cdot B} \quad -(1)$$

→ Take double inversion of RHS.

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

$$Y = \overline{\overline{\bar{A} + \bar{B}}} \quad \text{(Demorgan's theorem).}$$



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

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

5. EX-OR Gate using NOR Gate.

→ Boolean Expression of EX-OR Gate.

$$Y = \bar{A}B + A\bar{B} \quad -(1)$$

$$Y = X + Z \quad -(2) \qquad (X = \bar{A}B, Z = A\bar{B})$$

→ Take double inversion of RHS of eq(2)

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

$$Y = \overline{\bar{X} \cdot \bar{Z}} \quad \text{(From Demorgan's theorem).}$$

put X and Z value.

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

$$Y = \overline{(\overline{\overline{A}} + \overline{B}) \cdot (\overline{A} + \overline{\overline{B}})} \quad \text{(using Demorgan's law)}$$

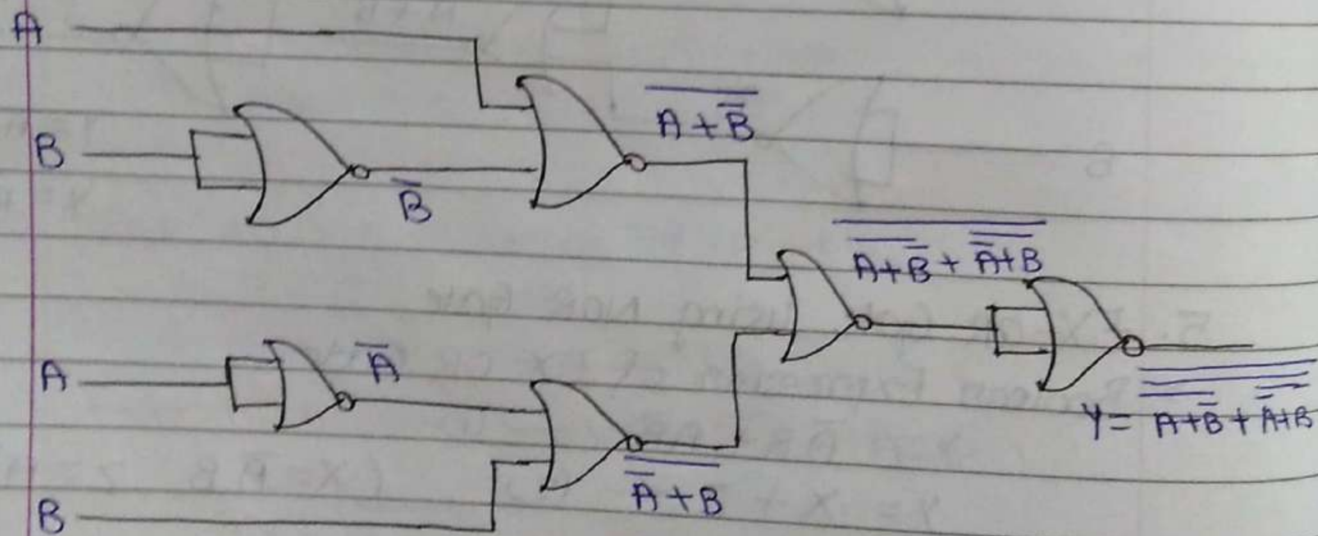
$$Y = \overline{(A + \overline{B}) \cdot (\overline{A} + B)}$$

Again using demorgan's law.

$$Y = \overline{A + \overline{B}} + \overline{\overline{A} + B} \quad \text{——— (3)}$$

→ Take double inversion of eq(3) on RHS

$$Y = \overline{\overline{\overline{A + \overline{B}} + \overline{\overline{A} + B}}} \quad \text{——— (4)}$$



6. EX-NOR Gate using only NOR Gate.

→ Boolean Expression of EX-NOR is.

$$Y = \overline{A}\,\overline{B} + AB \quad \text{— (1)}$$

$$Y = X + Z \quad \text{—— (2)} \qquad (X = \overline{A} \cdot \overline{B} \ , Z = AB)$$

→ Take double inversion of eq (2)

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

Scanned with CamScanner

$$Y = \overline{\overline{X}\cdot\overline{Z}} \quad — (3)$$

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

Using Demorgan's law.

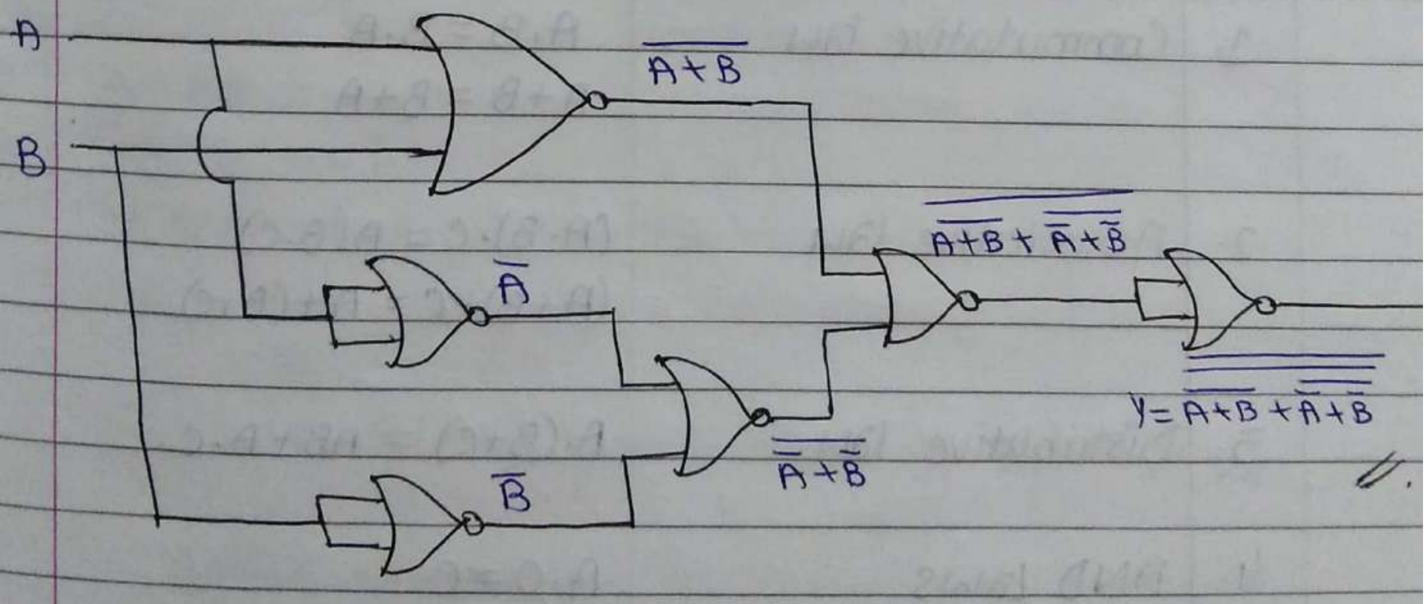
$$y = \overline{(A+B)\cdot(\overline{A}+\overline{B})}$$

Again using Demorgan's law

$$y = (\overline{A+B}) + (\overline{\overline{A}+\overline{B}}) \quad — (4)$$

→ Take double inversion of eq (4) on RHS

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



//.

Boolean Algebra

→ Boolean Algebra is a system of mathematical logic.
→ Therefore, in order to deal mathematically with digital circuits, we need to use boolean Algebra.
→ Boolean Algebra may be defined as with the help of following aspects:-

(I) A set of elements
(II) A set of operators
(III) A number of unproved postulates or axions.

→ Boolean laws and postulates.

S.No	Name of law	Statements.
1.	Commutative law	$A.B = B.A$ $A+B = B+A$
2.	Associative law	$(A.B).C = A(B.C)$ $(A+B)+C = A+(B+C)$
3.	Distributive law	$A.(B+C) = AB+A.C$
4.	AND Laws	$A.0=0$ $A.1=A$ $A.A=A$ $A.\bar{A}=0$
5.	OR Laws	$A+0=A$ $A+1=1$ $A+A=A$

	$A+\bar{A}=1.$
6. Inversion Law	$\bar{\bar{A}}=A.$
7. Other Important laws.	$A+BC=(A+B)\cdot(A+C)$ $\bar{A}+AB=\bar{A}+B$ $\bar{A}+A\bar{B}=\bar{A}+\bar{B}$ $A+AB=A.$ $A+\bar{A}B=A+B$

Q1. Prove $A+AB=A$.

Ans- $LHS = A+AB$
$= A(1+B)$
$= A(1)$
$= A = RHS$

Hence proved.

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

Ans. $LHS = A+\bar{A}B$

We know that $A=A+AB$

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

Hence proved.

3. $(A+B)\cdot(A+C)=A+BC$

Ans - $LHS=(A+B)\cdot(A+C)$
$= A\cdot A+A\cdot C+B\cdot A+B\cdot C$
$= A+A\cdot C+A\cdot B+B\cdot C$
$= A+A\cdot C+B\cdot C$ $\quad (A=A+AB)$
$= A(1+C)+BC$

$= A + BC = RHS.$

Hence proved.

Duality principle

→ It is an important property of boolean algebra.

→ It states that in a two valued boolen algebra, the dual of an algebric expression can be obtained by interchanging OR and AND operators and ~~rept~~ by replacing 1's by 0's and 0's by 1's.

Duality Theorem.

→ According to the duality theorem, the following conversions are possible in a given boolean expression.

(i) We can change each AND to an OR operation

(ii) We can change each OR to an AND operation

(iii) We can complement any 1 or 0 appearing in the expression.

Ex:- 1. $A(B+C) = AB + AC$

Apply duality theorem.

$$A + BC = (A+B)\cdot(A+C).$$

2. Find the dual of the following.

(i) $A + AB = A$

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

(iii) $A + \bar{A} = 1$

(i) Apply duality theorem.

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

Date

(II) Apply duality theorem

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

(III) Apply duality theorem

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

Demorgan's theorems

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

verification:

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

LHS = RHS

Theorem 2: $\overline{A+B} = \bar{A} \cdot \bar{B}$

verification:

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

Q. Prove that $A + \bar{A}B + AB = A + B$

→ Proof :-

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

$$= A + B(1) \qquad (\because \bar{A} + A)$$

$$= A + B = RHS$$

Hence proved.

Q. Simplify the following expression.

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

From de-morgan's theorem $(\overline{\overline{A}\overline{B}} = \overline{\overline{A}} + \overline{\overline{B}})$

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

~~Using other important law~~

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

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

~~Using OR law~~ $\cancel{\overline{B} + B = 1}$.

$\cancel{Y = (\overline{2\overline{A} + 1})}$

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

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

$Y = \overline{\overline{A}} \cdot \overline{\overline{B}} \cdot \overline{AB}$ (De morgan's theorem)

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

$Y = A \cdot B (\overline{A} + \overline{B})$ (De morgan's theorem)

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

$Y = 0 + 0$

$\boxed{Y = 0}$

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

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

$Y = \overline{B}(A + \overline{A}) + B(\overline{A} + A)$ $(A + \overline{A} = 1)$

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

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

$Y = 1$

(III) $A\bar{B}C + \bar{A}BC + ABC = Y$.

$\Rightarrow AC(\bar{B}+B) + \bar{A}BC \qquad (\bar{B}+B=1)$

$= AC + \bar{A}BC$

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

$= C(A+B)$

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

Boolean Functions.

→ Boolean function is described by algebric expression called boolean expression which consist of binary variables and logic operation symbols.

For example:

$$\underbrace{F(A,B,C,D)}_{\text{Boolean function}} = \underbrace{A + B\bar{C} + ADC}_{\text{Boolean expression}}$$

It is of two types:-

(I) SOP form [Sum of Product].

(II) POS form [Product of Sum].

→ (1.) Sum of product form (SOP):

→ The SOP form expression contains two or more AND terms ORed together.

cd=ccd

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

or

$$Y = XY + \bar{X}Y + X\bar{Y} + \bar{X}\bar{Y}$$

→ (2.) Product of Sum form (POS)

→ The POS form expression contains two or more OR terms ANDed together.

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

or

$$Y = (X+Y)\cdot(\bar{X}+Y)\cdot(X+\bar{Y})\cdot(\bar{X}+\bar{Y}).$$

Canonical form or Standard form.

→ Generally the SOP or POS expression does not contains all the literals (variables).

Ex:- $ABC + B\bar{C}$

→ If each term of SOP or POS expression contains all literals (variables), then it is called Canonical or Standard form.

Ex:- $Y = ABC + A\bar{B}C + \bar{A}CB$
$Y = (A+B+C)\cdot(A+\bar{B}+C)\cdot(\bar{A}+C+B)$ — All variable A, B, C are present.

Ex:- $Y = AB + A\bar{C} + BC$, convert this into its canonical form.

→ Find the missing literals.

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

AB → C missing, $A\bar{C}$ → B missing, BC → A missing.

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

$A + \bar{A} = 1$
$B + \bar{B} = 1$
$C + \bar{C} = 1.$

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

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

2. $Y = (A+B)(A+C)(B+\bar{C})$, convert this into standard form.

2. Find the missing literals.

$$Y = \underbrace{(A+B)}_{C \text{ missing}} \cdot \underbrace{(A+C)}_{B \text{ missing}} \cdot \underbrace{(B+\bar{C})}_{A \text{ missing}}$$

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

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

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

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

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

3. Convert the following in its standard form.

(I) $Y = AB + AC + BC$

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

(III) $Y = A + BC + ABC$.

3.(I) $Y = AB + AC + BC$

$Y = AB + AC + BC$

$Y = AB(C+\bar{C}) + A(B+\bar{B})C + (A+\bar{A})BC$ | $\bar{A}+A=1$, $\bar{B}+B=1$, $\bar{C}+C=1$

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

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

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

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

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

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

Using

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

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

$C\cdot\bar{C} = 0$

~~Y = (A+B)~~

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

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

(III) $Y = A + BC + ABC$.

$Y = A + BC + ABC$

A → B,C missing

BC → A missing

Page No.
Date

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

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

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

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

M-Notations : Minterms and Maxterms.

→ Minterms

- Each individual term in the standard SOP form is called Minterm.
- In this case, complement term is represented with 0's and non-complement term is represented with 1's.

(0 → with complement ($\bar{A}$), 1 → without complement (A))

- It is denoted by 'm' and given by the expression

$$Y = \Sigma m(\ )$$

Example:-

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

$$Y = 111 + 100 + 011$$

$$Y = m_7 + m_4 + m_3$$

$$\boxed{Y = \Sigma m(3,4,7)}$$

→ Maxterms

- Each individual term in the standard POS form is called Maxterm.
- The complement is represented by 1 and without complement by 0.
- This is denoted by M and given by the expression :-

$$Y = \pi M(\ )$$

Scanned with CamScanner

8 — 15   15 — 31

Example:

$$Y = (A+\bar{B}+C)(A+B+C)(\bar{A}+\bar{B}+C)$$
$$Y = (010)(000)(110)$$
$$Y = M_2 M_0 M_6$$

$$\boxed{Y = \pi M(0,2,6)}$$

Note:

→ Minterm and Maxterm are complementary to each other.

Minterm $Y = \Sigma m(0,1,3,5,6,8,9,11,13)$

Maxterm $Y = \pi M(2,4,7,10,12,14,15)$

complementary.

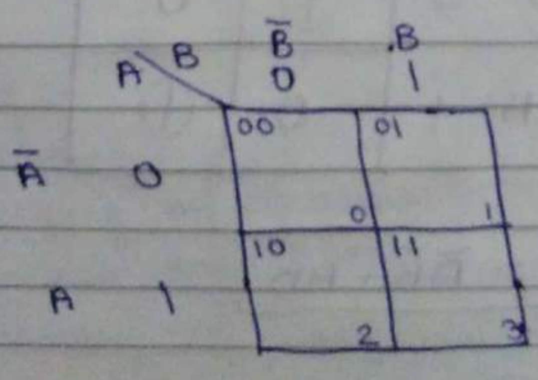
V.V.I:- <u>Karnaugh Map (K-Map)</u>

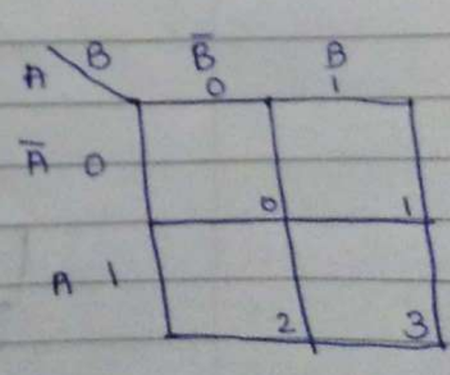
→ The K-Map is a graphical representation that provides a systematic methods or simplest method to simplify the boolean expression.

→ The requirement of number of cells is equal to 2^n, where $n = 1, 2, 3, \ldots$

① Two variable K-Map.

In 2-variable the number of cells will be $2^2 = 4$.





1. Verify the EX-NOR and EX-OR gate using K-Map.

1. EX-OR

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

EX-NOR

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

K-Map. = EX-OR

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

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

K-Map = EX-NOR

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

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

Scanned with CamScanner

(2) 3-Variable K-Map.

Number of cells $= 2^3 = 8$.

$A \backslash BC$	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0	000 (0)	001 (1)	011 (3)	010 (2)
A 1	100 (4)	101 (5)	111 (7)	110 (6)

Ex- Simplify $F(A,B,C) = \bar{A}BC + B\bar{C} + AB\bar{C} + A\bar{B}C$ in SOP and POS using K-Map.

→ $F = \bar{A}BC + B\bar{C} + AB\bar{C} + A\bar{B}C$

For SOP :- Min term.

$F = 011 + 010 + 110 + 101$

$F = m_3 + m_2 + m_6 + m_5$

$F = \Sigma m(2,3,5,6)$

$A \backslash BC$	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0	000 (0)	001 (1)	011 — 1 (3)	010 — 1 (2)
A 1	100 (4)	101 — 1 (5)	111 (7)	110 — 1 (6)

Note:- Grouping :-

→ 2 → Grouping
4 → Grouping
8 → Grouping
16 → Grouping

always with adjacent.

$$\boxed{F = A\bar{B}C + \bar{A}B + B\bar{C}.}$$

1st method

For POS :-

$F = 100 + 101 + 001 + 010$

$F = \pi M(0,1,4,7)$

A \ BC	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0	0 (0)	0 (1)	(3)	(2)
A 1	0 (4)	(5)	0 (7)	(6)

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

~~F = (A+B+C).(A+B).(B+C)~~

$F = \overline{\bar{B}\bar{C} + \bar{A}\bar{B} + ABC}$ (De-morgan's theorem).

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

2nd method

For POS

A \ BC	$(B+C)$ 00	$(B+\bar{C})$ 01	$(\bar{B}+\bar{C})$ 11	$(\bar{B}+C)$ 10
A 0	0 (0)	0 (1)	(3)	(2)
$\bar{A}$ 1	0 (4)	(5)	0 (7)	(6)

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

(3) 4-variable K-Map.

Number of cells $= 2^4 = 16$

AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	0	1	3	2
$\bar{A}B$ 01	4	5	7	6
AB 11	12	13	15	14
$A\bar{B}$ 10	8	9	11	10

Example-1.

AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	1			
$\bar{A}B$ 01	1		1	
AB 11			1	
$A\bar{B}$ 10	1			1

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

Example-2.

AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	1			1
$\bar{A}B$ 01		1	1	
AB 11		1	1	
$A\bar{B}$ 10	1			1

$Y = BD + \bar{B}\bar{D}$

Example-3

AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	1		1	1
$\bar{A}B$ 01	1		1	1
AB 11	1		1	1
$A\bar{B}$ 10	1		1	1

$Y = \bar{D} + C.$

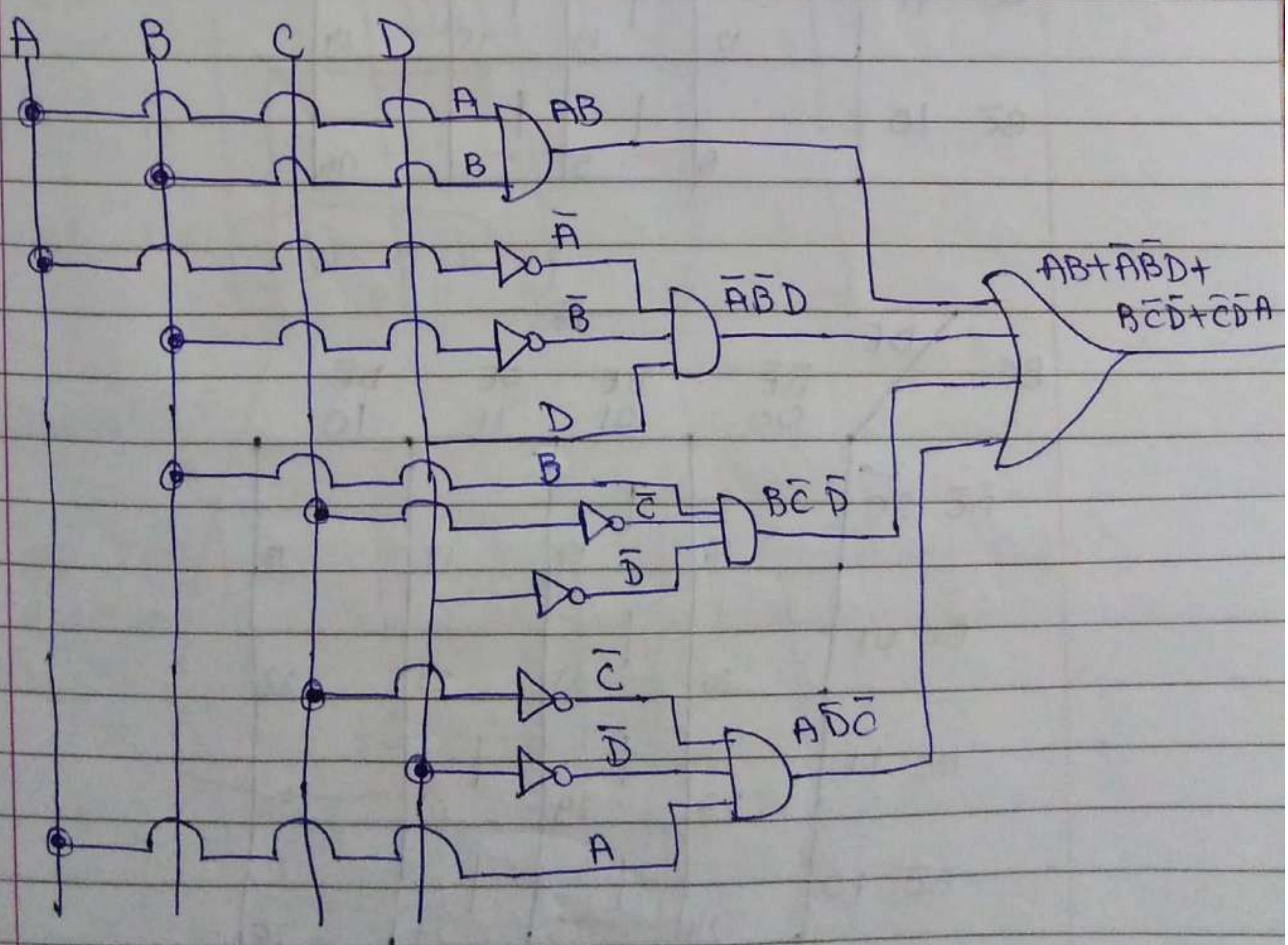
Ex- Minimise the SOP Expression with don't care (X) conditions.

$Y = \sum m(1,4,8,12,13,15) + d(3,14)$ ↗ don't

AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	$_{0}$	1_{1}	X_{3}	$_{2}$
$\bar{A}B$ 01	1_{4}	$_{5}$	$_{7}$	$_{6}$
AB 11	1_{12}	1_{13}	1_{15}	X_{14}
$A\bar{B}$ 10	1_{8}	$_{9}$	$_{11}$	$_{10}$

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

- Gate Realization.



(4) 5-Variable K-Map.

Number of cells $= 2^5 = 32$.

Q.→ Minimize the Boolean function, expression using K-Map.

$F(A,B,C,D,E) = \Sigma m(0,2,4,6,9,11,13,15,17,21,25,27,29,31)$.

$\bar{A}$

BC \ DE	$\bar{D}\bar{E}$ 00	$\bar{D}E$ 01	DE 11	$D\bar{E}$ 10
$\bar{B}\bar{C}$ 00	1 (0)	(1)	(3)	1 (2)
$\bar{B}C$ 01	1 (4)	(5)	(7)	1 (6)
BC 11	(12)	1 (13)	1 (15)	(14)
$B\bar{C}$ 10	(8)	1 (9)	1 (11)	(10)

A

BC \ DE	$\bar{D}\bar{E}$ 00	$\bar{D}E$ 01	DE 11	$D\bar{E}$ 10
$\bar{B}\bar{C}$ 00	(16)	1 (17)	(19)	(18)
$\bar{B}C$ 01	(20)	1 (21)	(23)	(22)
BC 11	(28)	1 (29)	1 (31)	(30)
$B\bar{C}$ 10	(24)	1 (25)	1 (27)	(26)

$\bar{A}+A=1$

overlap = 1

8-group

$Y = BE + \bar{A}\bar{B}\bar{E} + A\bar{D}E.$

(5.) 6-Variable K-Map.
Number of cells $= 2^6 = 64$

Q → Simplify the following six variable expression using K-Map.
$F(A,B,C,D,E,F) = \Sigma m(0,5,7,8,9,13,23,24,25,28,29,37,$
$40,41,42,43,55,56,57,60,61)$.

$\bar{B}$ — $\bar{A}$ — $\bar{A}\bar{B}$

CD \ EF	$\bar{E}\bar{F}$ 00	$\bar{E}F$ 01	EF 11	$E\bar{F}$ 10
$\bar{C}\bar{D}$ 00	1 (0)	(1)	(3)	(2)
$\bar{C}D$ 01	(4)	1 (5)	1 (7)	(6)
CD 11	(12)	1 (13)	(15)	(14)
$C\bar{D}$ 10	1 (8)	1 (9)	(11)	(10)

B — $\bar{A}$ — $\bar{A}B$

CD \ EF	$\bar{E}\bar{F}$ 00	$\bar{E}F$ 01	EF 11	$E\bar{F}$ 10
$\bar{C}\bar{D}$ 00	(16)	(17)	(19)	(18)
$\bar{C}D$ 01	(20)	(21)	1 (23)	(22)
CD 11	1 (28)	1 (29)	(31)	(30)
$C\bar{D}$ 10	1 (24)	1 (25)	(27)	(26)

$\bar{B}$

$CD \backslash EF$		$\bar E\bar F$ 00	$\bar E F$ 01	EF 11	$E\bar F$ 10	
	$\bar C\bar D$ 00	32	33	35	34	
A	$\bar C D$ 01	36	$\boxed{1}$ 37	39	38	$A\bar B$
	CD 11	44	45	47	46	
	$C\bar D$ 10	1 40	1 41	1 43	1 42	

B

$CD \backslash EF$		$\bar E\bar F$ 00	$\bar E F$ 01	EF 11	$E\bar F$ 10	
	$\bar C\bar D$ 00	48	49	51	50	
A	$\bar C D$ 01	52	53	1 55	54	AB
	CD 11	1 60	1 61	63	62	
	$C\bar D$ 10	1 56	1 57	59	58	

$\cancel{+ BC\bar E}$

$$Y = \bar A C\bar E\bar F + \bar B\bar C D\bar E F + A\bar B C\bar D + \bar A\bar B\bar E\bar F\bar D + BC\bar E + C\bar D\bar E + B\bar C DEF.$$

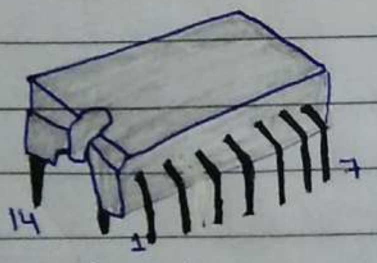
Integrated circuit (IC) technology.

Microchip or chip, an assembly of electronic components, fabricated as a single unit, in which miniaturized active device (eg - transistor, diodes) and passive device (eg - capacitor, resistor).

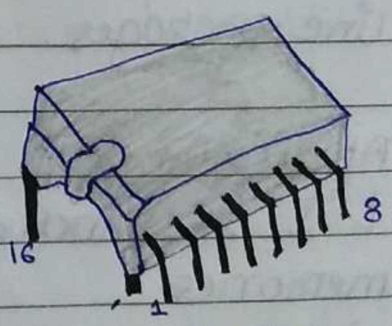
Types of integrated circuit.

1. Small-scale integration (SSI)
The simplest type of digital ICs are placed in the small scale integration (SSI) category.
These ICs have upto 10 equivalent gate circuits on a single chip.
No. of transistor - 1 to 10.
SSI ICs are fabricated in one of the two main configurations:-

(a) Dual-in line packages (DIP)



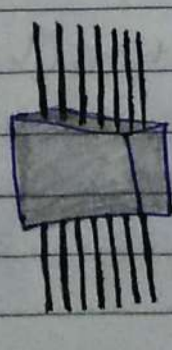
14-pin

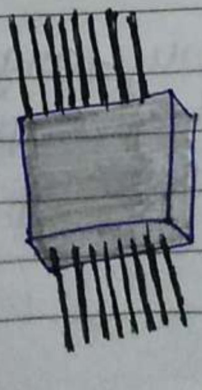


16-pin

(b) Flat pack

14-pin





16-pin

2. Medium-Scale integration (MSI).
The next larger commercially available ICs are called medium-scale integration (MSI) and contains the equivalent of about 10 to 99 gates.
Number of transistors - 10 to 500.

- Applications-
MSI circuits include the more complex logic functions such as in • flip-flops.
- encoders • counters • Demultiplexers
- decoders • Multiplexers • Registers
- Arithmetic circuits, small memories etc.

3. Large scale integration (LSI).
Large scale integration (LSI) ICs are still bigger and contains the equivalent of 100 to 9999 gates or more.
Most of the LSI ICs are fabricated in dual-in line packages.

- Applications.
- LSI ICs includes
→ memories → programmable logic devices
→ microprocessors → customised devices.

4. Very-Large scale integration (VLSI)
The complexity in VLSI is usually stated in terms of transistor count than gate count.

Any IC with over 1,000,000 transistor is VLSI IC.
No. of Logic gates - 10,000 to 99,999.

- Applications-
It includes advances microprocessor, large memories, larger programmable logic devices and customised devices.

⇒ Application of Small-Scale integration (SSI).
These includes basic gate functions
→ Logic gates - (AND, OR, NAND, NOR).

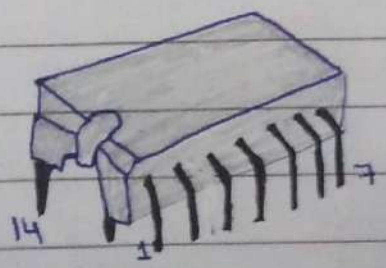
Integrated circuit (IC) technology.

Microchip or chip, an assembly of electronic components, fabricated as a single unit, in which miniaturized active device (eg - transistor, diodes) and passive device (eg - capacitor, resistor).

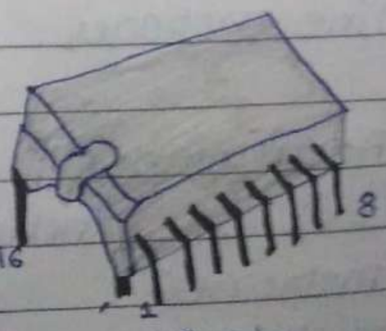
Types of integrated circuit.

1. Small-scale integration (SSI)
The simplest type of digital ICs are placed in the small scale integration (SSI) category.
These ICs have upto 10 equivalent gate circuits on a single chip.
No. of transistor - 1 to 10.
SSI ICs are fabricated in one of the two main configurations:-

(a) Dual-in line packages (DIP)

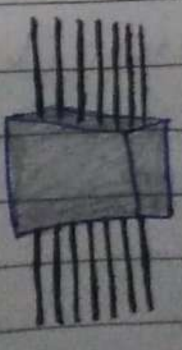


14-pin

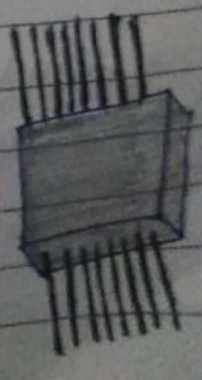


16-pin

(b) Flat pack



14-pin



16-pin

Scanned with CamScanner

2. Medium-Scale integration (MSI).
The next larger commercially available I.Cs are called medium-scale integration (MSI) and contains the equivalent of about 10 to 99 gates.
Number of transistors - 10 to 500.

- Applications-
MSI circuits include the more complex logic functions such as in • flip-flops.
- encoders • counters • Demultiplexers
- decoders • Multiplexers • Registers
- Arithmetic circuits, small memories etc.

3. Large scale integration (LSI).
Large scale integration (LSI) ICs are still bigger and contains the equivalent of 100 to 9999 gates or more.
Most of the LSI ICs are fabricated in dual-in line packages.

- Applications.
- LSI ICs includes
→ memories → programmable logic devices
→ microprocessors → customised devices.

4. Very-Large scale integration (VLSI)
The complexity in VLSI is usually stated in terms of transistor count than gate count.

Scanned with CamScanner

Any IC with over 1,000,000 transistor is VLSI IC.
No. of Logic gates - 10,000 to 99,999.

- Applications.

It includes advances microprocessor, large memories, larger programmable logic devices and customised devices.

⇒ Application of Small-Scale integration (SSI).
These includes basic gate functions
→ Logic gates - (AND, OR, NAND, NOR).