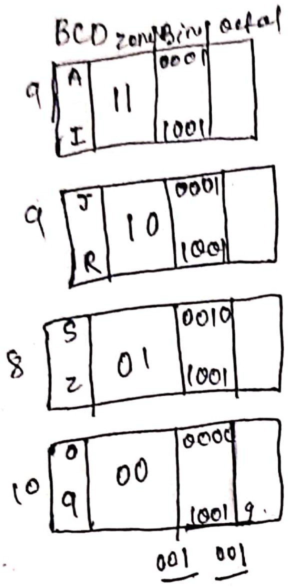
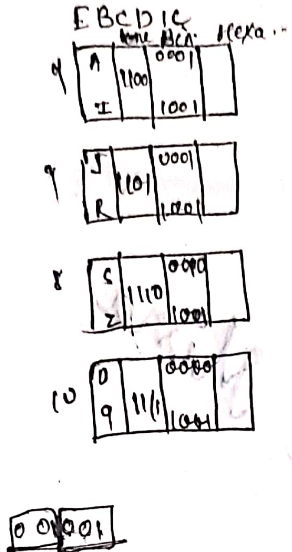
4 marks - 4x1 = 4

1 marks - 1x

1.5 marks -





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

001 001 = 8 + 1 = 9

Digital Logic :- UNIT-1

Character codes

Decimal system

Binary system

Decimal to Binary convertion

Hexadecimal notations

Boolean Algebra

Basic logic functions

Electronic logic gates

synthesis of logic functions

Minimisation of logic expressions

Minimisation using Karnaugh Maps

synthesis with NAND and NOR gates

Tri-state Buffers

Character Codes :-

→ Digital computers have made possible many scientific, industrial and commercial advances that became benefited for users of the computers.

→ Generally computers are used in scientific calculations, commercial and business data processing, air traffic control, space guidance, educational field and many other areas.

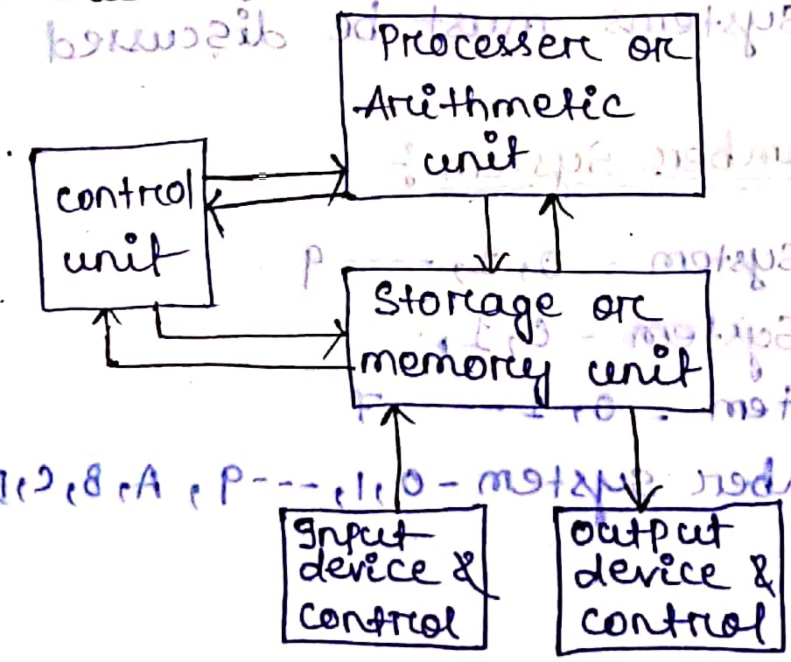
→ The generality of digital computers make all work operated perfectly.

→ The digital computers follow a sequence of instructions called as program to operate on given data.

→ The digital systems can represent information inside the computer system in some physical quantities called signals (Electronic signal).

→ The signals inside the electronic digital systems have only two discrete values like 0 and 1 are referred as Binary.

→ The block diagram of digital system is represented like this -



, Control unit supervises the program instructions.

→ Processer manipulates the data as specified by the program.

→ Memory unit stores both program and data.

→ Input devices transfers the program and data to the memory.

→ Output devices transfers the results from the computer to the user.

The digital computers work according to the above diagram and manipulates the descrete elements of information provided to it by representing them in the binary form.

→ The operands used for calculation are expressed in binary number system.

→ The decimal digits inside the system are represented in binary codes.

→ The data processing is carried out by means of binary logic elements using binary signals.

→ The quantities are stored in binary storage elements.

→ Hence the number system followed by the operator and the digital systems must be discussed in detail.

<u>Various types of Number System</u> :-

i) Decimal Number system - 0, 1, ---, 9

ii) Binary Number System - 0, 1.

iii) Octal Number system - 0, 1 --- 7

iv) Hexadecimal Number system - 0, 1, --- 9, A, B, C, D, E, F

The digital computer can receive any type of data like alpha - numeric data (a, b, c --- z, 0, 1, 2 --- 9 etc) and special symbols (+, -, *, /, space, =, —, \, ', ;, :, etc.

To represent these symbols in binary coding a several group of bits can be used.

→ A sequence of or group of bits can be refered as a byte (8 bits).

Data Represent Formats :-

BCD — 6 bits (Binary coded decimal)

EBCDIC — 8 bits (Extended Binary coded Decimal Instruction code)

ASCII-7 — 7 bits

ASCII-8 — 8 bits (American standard code for Information Interchange

	Octal	Hexadecimal
0	000	0000
1	001	0001
2	010	0010
3	011	0011
4	100	0100
5	10ϕ	0101
6	110	0110
7	111	0111
8		1000
9		1001
10		1010 (A)
11		1011 (B)
12		1100 (C)
13		1101 (D)
14		1110 (E)
15		1111 (F)

i) Convertion of Decimal to Binary and Binary to Decimal :-

$(\quad)_{10} = (\quad)_2$

1. $(256)_{10} = (100000000)_2$

$= 1\times2^8 + 0\times2^7 + 0\times2^6 + 0\times2^5 +$
$0\times2^4 + 0\times2^3 + 0\times2^2 + 0\times2^1 + 0\times2^0$
$= (256)_{10}$

$2\,|\,256 - 0$
$2\,|\,128 - 0$
$2\,|\,64 - 0$
$2\,|\,32 - 0$
$2\,|\,16 - 0$
$2\,|\,8 - 0$
$2\,|\,4 - 0$
$2\,|\,2 - 0$
$2\,|\,1 - 1$
0

2. $(459)_{10} = (111001011)_2$

$= 1\times2^8 + 1\times2^7 + 1\times2^6 + 0\times2^5 +$
$0\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 +$
1×2^0
$= 256 + 128 + 64 + 8 + 2 + 1$
$= (459)_{10}$

$2\,|\,459 - 1$
$2\,|\,229 - 1$
$2\,|\,114 - 0$
$2\,|\,57 - 1$
$2\,|\,28 - 0$
$2\,|\,14 - 0$
$2\,|\,7 - 1$
$2\,|\,3 - 1$
$2\,|\,1 - 1$
0

3. $(2341)_{10} = (100100100101)_2$

$= 1\times2^{11} + 0\times2^{10} + 0\times2^9 + 1\times2^8 +$
$0\times2^7 + 0\times2^6 + 1\times2^5 + 0\times2^4 +$
$0\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0$
$= 2048 + 256 + 32 + 4 + 1$
$= (2341)_{10}$

$2\,|\,2341 - 1$
$2\,|\,1170 - 0$
$2\,|\,585 - 1$
$2\,|\,292 - 0$
$2\,|\,146 - 0$
$2\,|\,73 - 1$
$2\,|\,36 - 0$
$2\,|\,18 - 0$
$2\,|\,9 - 1$
$2\,|\,4 - 0$
$2\,|\,2 - 0$
$2\,|\,1 - 1$
0

4. $(1375)_{10} = (10101011111)_2$

$= 1\times2^{10} + 0\times(2^9) + 1\times2^8 +$
$0\times2^7 + 1\times2^6 + 0\times2^5 +$
$1\times2^4 + 1\times2^3 + 1\times2^2 +$
$1\times2^1 + 1\times2^0$
$= 1024 + 256 + 64 + 16 + 8 + 4 + 2 + 1$
$= (1375)_{10}$

$2\,|\,1375 - 1$
$2\,|\,687 - 1$
$2\,|\,343 - 1$
$2\,|\,171 - 1$
$2\,|\,85 - 1$
$2\,|\,42 - 0$
$2\,|\,21 - 1$
$2\,|\,10 - 0$
$2\,|\,5 - 1$
$2\,|\,2 - 0$
$2\,|\,1 - 1$
0

ii) Convertion of Octal to Decimal :-

1. $(153)_8 = 1\times8^2 + 5\times8^1 + 3\times8^0$
$= 64 + 40 + 3$
$= (107)_{10}$

2. $(2754)_8 = 2\times8^3 + 7\times8^2 + 5\times8^1 + 4\times8^0$
$= 1024 + 448 + 40 + 4$
$= (1516)_{10}$

3. $(3106)_8 = 3\times8^3 + 1\times8^2 + 0\times8^1 + 6\times8^0$
$= 1536 + 64 + 6$
$= (1606)_{10}$

4. $(500)_8 = 5\times8^2 + 0\times8^1 + 0\times8^0$
$= (320)_{10}$

iii) Convertion of Decimal to Octal :-

1. $(107)_{10} = (153)_8$

$$\begin{array}{r|l l} 8 & 107 & -3 \\ 8 & 13 & -5 \\ 8 & 1 & -1 \\ & 0 & \end{array}$$

2. $(1516)_{10} = (2754)_8$

$$\begin{array}{r|l l} 8 & 1516 & -4 \\ 8 & 189 & -5 \\ 8 & 23 & -7 \\ 8 & 2 & -2 \\ & 0 & \end{array}$$

3. $(1606)_{10} = (3106)_8$

$$\begin{array}{r|l l} 8 & 1606 & -6 \\ 8 & 200 & -0 \\ 8 & 25 & -1 \\ 8 & 3 & -3 \\ & 0 & \end{array}$$

4. $(320)_{10} = (500)_8$

$$\begin{array}{r|l l} 8 & 320 & -0 \\ 8 & 40 & -0 \\ 8 & 5 & -5 \\ & 0 & \end{array}$$

iv) Conversion of Octal to Binary

Octal → decimal → Binary

1. $(256)_8 = 2\times8^2 + 5\times8^1 + 6\times8^0$

$= 128 + 40 + 6$

$= (174)_{10}$

2	174	0
2	87	1
2	43	1
2	21	1
2	10	0
2	5	1
2	2	0
2	1	1
	0	

Ans = $(10101110)_2$

$1\times2^7 + 0\times2^6 + 1\times2^5 + 0\times2^4 + 1\times2^3 +$
$1\times2^2 + 1\times2^1 + 0\times2^0$

$= 128 + 32 + 8 + 4 + 2$

$= (174)_{10}$

8	174	6
8	21	5
8	2	2
	0	

$(256)_8$

2. $(743)_8 = 7\times8^2 + 4\times8^1 + 3\times8^0$

$= 448 + 32 + 3$

$= (483)_{10}$

2	483	1
2	241	1
2	120	0
2	60	0
2	30	0
2	15	1
2	7	1
2	3	1
2	1	1
	0	

Ans = $(111100011)_2$

$1\times2^8 + 1\times2^7 + 1\times2^6 + 1\times2^5 + 0\times2^4 + 0\times2^3 +$
$0\times2^2 + 1\times2^1 + 1\times2^0$

$= 256 + 128 + 64 + 32 + 2 + 1$

$= (483)_{10}$

8	483	3
8	60	4
8	7	7
	0	

$(743)_8$

3. $(312)_8 = 3\times8^2 + 1\times8^1 + 2\times8^0$

$= 192 + 8 + 2$

$= (202)_{10}$

2	202	0
2	101	1
2	50	0
2	25	1
2	12	0
2	6	0
2	3	1
2	1	1
	0	

Ans = $(11001010)_2$

$1\times2^7 + 1\times2^6 + 0\times2^5 + 0\times2^4 + 1\times2^3 +$
$0\times2^2 + 1\times2^1 + 0\times2^0$

$= 128 + 64 + 8 + 2$

$= (202)_{10}$

8	202	2
8	25	1
8	3	3
	0	

$(312)_8$

v) Conversion of Binary to Octal :

Binary → decimal → Octal

1. $(10101110)_2 = 1\times2^7 + 0\times2^6 + 1\times2^5 + 0\times2^4 + 1\times2^3 + 1\times2^2 + 1\times2^1 + 0\times2^0$

$= (174)_{10}$

8 | 174 — 6
8 | 21 — 5
8 | 2 — 2
0

Ans = $(256)_8$

2. $(111100011)_2 = 1\times2^8 + 1\times2^7 + 1\times2^6 + 1\times2^5 + 0\times2^4 + 0\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0$

$= 256 + 128 + 64 + 32 + 2 + 1$

$= (483)_{10}$

8 | 483 — 3
8 | 60 — 4
8 | 7 — 7
0

Ans = $(743)_8$

3. $(11001010)_2 = 1\times2^7 + 1\times2^6 + 0\times2^5 + 0\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0$

$= 128 + 64 + 8 + 2$

$= (202)_{10}$

8 | 202 — 2
8 | 25 — 1
8 | 3 — 3
0

Ans = $(312)_8$

4. $(1110010)_2 = 1\times2^6 + 1\times2^5 + 1\times2^4 + 0\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0$

$= 64 + 32 + 16 + 2$

$= (114)_{10}$

8 | 114 — 4
8 | 14 — 6
8 | 1 — 1
0

Ans = $(164)_8$

vi) Converction of Hexadecimal to Decimal:

1. $(39A)_{16} = 3\times16^2 + 9\times16^1 + A\times16^0$

$= 3\times256 + 144 + 10\times1$

$= 768 + 154$

$= (922)_{10}$

2. $(B65F)_{16} = B\times16^3 + 6\times16^2 + 5\times16^1 + F\times16^0$

$= 11\times4093 + 6\times256 + 5\times16 + 15\times1$

$= 45056 + 1536 + 95$

$= (46687)_{10}$

3. $(7FE)_{10} = 7\times16^2 + F\times16^1 + E\times16^0$

$= 7\times256 + 15\times16 + 14\times1$

$= 1792 + 240 + 14$

$= (2046)_{10}$

vii) Converction of Decimal to Hexadecimal :-

1. $(922)_{10} = (\cancel{11\text{-}65\text{-}15})_{16}$

$= (3\,9\text{-}10)_{16}$

$= (39A)_{16}$

16 | 922 — 15
16 | 2917 — 5
16 | 182 — 6
16 | 11 — 11
0

16 | 922 — 10
16 | 57 — 9
16 | 3 — 3
0

2. $(46687)_{10} = (11\text{-}65\text{-}15)_{16}$

$= (B65F)_{16}$

16 | 46687 — 15
16 | 2917 — 5
16 | 182 — 6
16 | 11 — 11
0

3. $(\cancel{7FE})_{10}$ $(2046)_{10} = (7\text{-}15\text{-}14)_{16}$

$= (7FE)_{16}$

16 | 2046 — 14
16 | 127 — 15
16 | 7 — 7
0

viii) Conversion of Hexadecimal to Binary :-

Hexadecimal → decimal → Binary

1. $(B65F)_{16}$ $= B\times16^3 + 6\times16^2 + 5\times16^1 + F\times16^0$

$= 11\times4096 + 6\times256 + 5\times16 + 15\times1$

$= 45056 + 1536 + 80 + 15$

$= (46687)_{10}$

2 | 46687	1
2 | 23343	1
2 | 11671	1
2 | 5835	1
2 | 2917	1
2 | 1458	0
2 | 729	1
2 | 364	0
2 | 182	0
2 | 91	1
2 | 45	1
2 | 22	0
2 | 11	1
2 | 5	1
2 | 2	0
2 | 1	1
0	

Ans = $(1011011001011111)_2$

$1\times2^{15} + 0\times2^{14} + 1\times2^{13} + 1\times2^{12} + 0\times2^{11} + 1\times2^{10} +$
$1\times2^{9} + 0\times2^{8} + 0\times2^{7} + 1\times2^{6} + 0\times2^{5} +$
$1\times2^{4} + 1\times2^{3} + 1\times2^{2} + 1\times2^{1} + 1\times2^{0}$

$= 32768 + 8192 + 4096 + 1024 + 512 +$
$64 + 16 + 8 + 4 + 2 + 1$

$= (46687)_{10}$

2. $(7FE)_{16}$ $= 7\times16^2 + F\times16^1 + E\times16^0$

$= 7\times256 + 15\times16 + 14\times1$

$= 1792 + 240 + 14$

$= (2046)_{10}$

2 | 2046	0
2 | 1023	1
2 | 511	1
2 | 255	1
2 | 127	1
2 | 63	1
2 | 31	1
2 | 15	
2 | 7	
2 | 3	
2 | 1	
0	

Ans = $(11111111110)_2$

$= 1\times2^{10} + 1\times2^{9} + 1\times2^{8} + 1\times2^{7} + 1\times2^{6} +$
$1\times2^{5} + 1\times2^{4} + 1\times2^{3} + 1\times2^{2} + 1\times2^{1} +$
0×2^{0}

$= 1024 + 512 + 256 + 128 + 64 + 32 + 16 +$
$8 + 4 + 2$

$= (2046)_{10}$

Binary → decimal → Hexadecimal

1. $(1011011001011111)_2$

$1\times2^{15} + 0\times2^{14} + 1\times2^{13} + 1\times2^{12} + 0\times2^{11} + 1\times2^{10} + 1\times2^{9} +$
$0\times2^{8} + 0\times2^{7} + 1\times2^{6} + 0\times2^{5} + 1\times2^{4} + 1\times2^{3} + 1\times2^{2} +$
$1\times2^{1} + 1\times2^{0}$

$= 32768 + 8192 + 4096 + 1024 + 512 + 64 + 16 + 8 + 2 + 1$

$= (46687)_{10}$

16	46687	15
16	2917	5
16	182	6
16	11	11
	0	

Ans: $(11\text{-}5\ 6\text{-}15)_{16}$

$= (B56F)_{16}$

2. $(11111111110)_2$

$= 1\times2^{10} + 1\times2^{9} + 1\times2^{8} + 1\times2^{7} + 1\times2^{6} + 1\times2^{5} +$
$1\times2^{4} + 1\times2^{3} + 1\times2^{2} + 1\times2^{1} \times 0\times2^{0}$

$= (2046)_{10}$

16	2046	14
16	127	15
16	7	7
	0	

Ans: $(7\text{-}15\text{-}14)_{16}$

$= (7FE)_{16}$

Convertion on floating point numbers :-

* Binary to Decimal :-

1. $(110.101)_2 = (\quad)_{10}$

$110.101 = 1\times2^2 + 1\times2^1 + 0\times2^0 + (1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3})$

$= 4 + 2 + (1\times\frac{1}{2} + 0 + 1\times\frac{1}{2^3})$

$= 6 + (\frac{1}{2} + \frac{1}{8})$

$= 6 + 0.5 + 0.125$

$= (6.625)_{10}$

2. $(101.1101)_2 = (\quad)_{10}$

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

$= 4 + 1 + (1\times 1/2 + 1\times 1/2^2 + 1\times 1/2^4)$

$= 5 + (\frac{1}{2} + \frac{1}{4} + \frac{1}{16})$

$= 5 + (0.5 + 0.25 + 0.0625)$

$= (5.8125)_{10}$

* Decimal to Binary :-

1. $(5.8125)_{10} =$

$$\begin{array}{r|l} 2 & 5 - 1 \\ 2 & 2 - 0 \\ 2 & 1 - 1 \\ & 0 \end{array}$$

$.8125 \times 2 = 1.6250 \rightarrow 1$

$.6250 \times 2 = 1.2500 \rightarrow 1$

$.2500 \times 2 = 0.5000 \rightarrow 0$

$.5000 \times 2 = 1.0000 \rightarrow 1$

$(101.1101)_2$

2. $(6.625)_{10} =$

$$\begin{array}{r|l} 2 & 6 - 0 \\ 2 & 3 - 1 \\ 2 & 1 - 1 \\ & 0 \end{array}$$

$625 \times 2 = 1.250 \rightarrow 1$

$250 \times 2 = 0.500 \rightarrow 0$

$500 \times 2 = 1.000 \rightarrow 1$

$(110.101)_2$

3) $(4.47)_{10}$

step 1

$$\begin{array}{r|l} 2 & 4 \quad 0 \\ 2 & 2 \quad 0 \\ 2 & 1 \quad 1 \\ & 0 \end{array}$$

step 2

$47 \times 2 = 0.94 \rightarrow 0$
$94 \times 2 = 1.88 \rightarrow 1$
$88 \times 2 = 1.76 \rightarrow 1$
$76 \times 2 = 1.52 \rightarrow 1$
$52 \times 2 = 1.04 \rightarrow 1$

$k = 5$

step 1 = 100
step 2 = 01111

combining step 1 & 2
$= (100.01111)_2$

Questions to Practise :

1) $(6.986)_{10} = (\quad)_2$

$k = 8$

step 1

$$\begin{array}{r|l} 2 & 6 \quad 0 \\ 2 & 3 \quad 1 \\ 2 & 1 \quad 1 \\ & 0 \end{array}$$

step 2

$986 \times 2 = 1.972 \rightarrow 1$
$972 \times 2 = 1.944 \rightarrow 1$
$944 \times 2 = 1.888 \rightarrow 1$
$888 \times 2 = 1.776 \rightarrow 1$
$776 \times 2 = 1.552 \rightarrow 1$
$552 \times 2 = 1.104 \rightarrow 1$
$104 \times 2 = 0.208 \rightarrow 0$
$208 \times 2 = 0.416 \rightarrow 0$

step-1 = 110
step 2 = 11111100
combining step 1 and 2

$(110.11111100)_2$

$= 1\times2^2 + 1\times2^1 + 0\times2^0 + (1\times2^{-1} + 1\times2^{-2} + 1\times2^{-3} + 1\times2^{-4} + 1\times2^{-5} + 1\times2^{-6} + 0\times2^{-7} + 0\times2^{-8})$

$= 4+2+(0.5 + 0.25 + 0.125 + 0.0625 + 0.03125 + 0.015625)$

$= 6 + 0.984375$
$= 6.984375$

* Octal to Decimal :

1. $(2.275)_8 = (\quad)_{10}$

$2 \times 8^0 + (2 \times 8^{-1} + 7 \times 8^{-2} + 5 \times 8^{-3})$

$= 2 + (2 \times \frac{1}{8} + 7 \times \frac{1}{8^2} + 5 \times \frac{1}{8^3})$

$= 2 + (\frac{1}{4} + 7 \times \frac{1}{64} + 5 \times \frac{1}{512})$

$= 2 + (2 \times 0.125 + 7 \times (0.015625) + 5 \times (0.001953125))$

$= 2 + (0.25 + 0.109375 + 0.009765625)$

$= (2.369140625)_{10}$

* Decimal to Octal :

1. $(2.369140625)_{10} = (\quad)_8$

Step-1

$8 \underline{| 2} - 2 \uparrow$
$\;\; 0$

Step-2

$.369140625 \times 8 = 2.953125 \rightarrow 2$

$.953125 \times 8 = 7.625000 \rightarrow 7$

$.625000 \times 8 = 5.000000 \rightarrow 5$

Step.1 = 2 Step2 = 275

Combining Step1 and 2

$(2.275)_8$

Q. $(7643.32)_8 = (\quad)_{10}$

$= 7 \times 8^3 + 6 \times 8^2 + 4 \times 8^1 + 3 \times 8^0 + (3 \times 8^{-1} + 2 \times 8^{-2})$

$= 7 \times 512 + 6 \times 64 + 32 + 3 + (3 \times \frac{1}{8} + 2 \times \frac{1}{8^2})$

$= 3584 + 384 + 35 + (3 \times 0.125 + 2 \times 0.015625)$

$= 4003 + (0.375 + 0.031250)$

$= 4003 + 0.406250$

$= (4003.406250)_{10}$

Step 1

8 | 4003 — 3
8 | 500 — 4
8 | 62 — 6
8 | 7 — 7
0

Step 2

.406250 × 8 = 3.250000 → 3
.250000 × 8 = 2.000000 → 2

combining step 1 & 2

$(7643.32)_8$

2. $(124.327)_8 = (\quad)_{10}$

$1\times8^2 + 2\times8^1 + 4\times8^0 + (3\times8^{-1} + 2\times8^{-2} + 7\times8^{-3})$

$= 64 + 16 + 4 + \left(3\times\frac{1}{8} + 2\times\frac{1}{64} + 7\times\frac{1}{8^3}\right)$

$= 84 + (3\times0.125 + 2\times0.015625 + 7\times0.0019$

$= 84 + 0.375 + 0.031250 + 0.01367 1875$

$= (84.4192875)_{10}$

Step-1

8 | 84 — 4
8 | 10 — 2
8 | 1 — 1
0

Step-2

.4192875 × 8 = 3.3543000 → 3
.3543000 × 8 = ~~2.3543000~~
2.87544000 → 2
.87544000 × 8 = 7.00352000 → 7

Step-1 = 124 Step-2 = 327

combining step 1 and step 2

$= (124.327)_8$

* Hexadecimal to Decimal :

$(A2B \cdot C4)_{16} = (\quad)_{10}$

$A \times 16^2 + 2 \times 16^1 + B \times 16^0 + (C \times 16^{-1} + 4 \times 16^{-2})$

$= 10 \times 256 + 2 \times 16 + 11 \times 1 + \left(12 \times \frac{1}{16} + 4 \times \frac{1}{16^2}\right)$

$= 2560 + 43 + (0 \cdot 75 + 0 \cdot 015625)$

$= 2603 + 0 \cdot 765625$

$= (2603 \cdot 765625)_{10}$

* Decimal to Hexadecimal :

$(2603 \cdot 765625)_{10} = (\quad)_{16}$

Step 1

$16 \,|\, 2603 \; - 11$
$16 \,|\, 162 \; - 2$
$16 \,|\, 10 \; - 10$
0

Step 2

$\cdot 765625 \times 16 = 12 \cdot 250000 \to 12$

$\cdot 250000 \times 16 = 4 \cdot 000000 \to 4$

Step 1 = 10_2_11
= A2B

Step 2 = 12_4
= C4

combining step 1 and Step 2

$(A2B \cdot C4)_{16}$

Q. $(EC8 \cdot 6D)_{16} = (\quad)_{10}$

$E \times 16^2 + C \times 16^1 + 8 \times 16^0 + (6 \times 16^{-1} + D \times 16^{-2})$

$= 14 \times 256 + 12 \times 16 + 8 + \left(6 \times \frac{1}{16} + 13 \times \frac{1}{256}\right)$

$= 3584 + 192 + 8 + (6 \times 0 \cdot 0625 + 13 \times 0 \cdot 00390625)$

$= 3784 + (0 \cdot 3750 + 0 \cdot 05078125)$

$= (3784 \cdot 42578125)_{10}$

Step-1

$16 \,|\, 3784 \; - 8$
$16 \,|\, 236 \; - 12$
$16 \,|\, 14 \; - 14$
0

14_12_8 = EC8

Step 2

$\cdot 42578125 \times 16 = 6 \cdot 81250000 \to 6$

$\cdot 81250000 \times 16 = 13 \cdot 00000000 \to 13$

6_13
= 6D

Step-1 = EC8 Step 2 = 6D

Combining step1 and 2

$(EC8.6D)_{10}$

Q. $(DF.19E)_{16} = (\quad)_{10}$

$D \times 16^1 + F \times 16^0 + (1 \times 16^{-1} + 9 \times 16^{-2} + E \times 16^{-3})$

$= 13 \times 16 + 15 + \left(\frac{1}{16} + 9 \times \frac{1}{16^2} + 14 \times \frac{1}{16^3}\right)$

$= 208 + 15 + (0.0625 + 9 \times 0.00390625 + 14 \times 0.000244140625)$

$= 225 + (0.0625 + 0.03515625 + 0.00341796$

$= 223 + (0.1010742187 5)$

$= (223.1010742187 5)_{10}$

Step 1

$16 \underline{| 223} - 15$

$16 \underline{| 13} - 13$

0

13_15

= DF

Step2

$.1010742187 5 \times 16 = 1.6171875 00$

$.6171875 0000 \times 16 = 9.875000000$

$.8750000000 \times 16 = 14.000000000$

1 19_14 = 19E

Step-1 = DF Step-2 = 19E

combining step1 and 2

$(DF.19E)_{16}$

A. Shortcut Methods to Convert :-

→ Octal to Binary
→ Binary to Octal
→ Hexadecimal to Binary
→ Binary to Hexadecimal

→

i) $(630.4)_8 = (\quad)_2$

$= (110\ 011\ 000 . 100)_2$

(ii) $(110011000.100)_2 = (\quad)_8$

$= (630.4)_8$

iii) $(10.7)_8 = (\quad)_2$

$(001\ 000 . 111)_2$

$= (1000 . 111)_2$

iv) $(1000.111)_2 = (\quad)_8$

$(001\ 000 . 111)_2$

$= (10.7)_8$

v) $(0.3475)_8 = (\quad)_2$

$= (0.011100111101)_2$

vi) $(0.011100111101)_2 = (\quad)_8$

$= (0.3475)_8$

vii) $(36.215)_8 = (\quad)_2$

$(011110 . 010101)_2$

viii) $(011110.010101)_2 = (\quad)_8$

$= (36.215)_8$

ix) $(732.154)_8 = (\quad)_2$

$(111\ 011\ 010 . 001101100)_2$

x) $(111011010.001101100)_2 = (\quad)_8$

$= (732.154)_8$

Conversion from any base to decimal :

$(\quad)_5 = (\quad)_{10}$

1. $(32.4)_5 = (\quad)_{10}$

$3 \times 5^1 + 2 \times 5^0 + (4 \times 5^{-1})$

$= 15 + 2 + (4 \times \frac{1}{5})$

$= 17 + 0.8$

$= 17.8$

2. $(345.8)_9 = (\quad)_{10}$

$3 \times 9^2 + 4 \times 9^1 + 5 \times 9^0 + 8 \times 9^{-1}$

$= 243 + 36 + 5 + (8 \times \frac{1}{9})$

$= 284 + 0.8888$

$= (284.8888)_{10}$

3. $(298.6)_{12} = (\quad)_{10}$

$2 \times 12^2 + 9 \times 12^1 + 8 \times 12^0 + 6 \times 12^{-1}$

$= 288 + 108 + 8 + 0.5$

$= (404.5)_{10}$

Conversion from Decimal to any base :

1. $(17.8)_{10} = (\quad)_5$

Step 1

$5 \,|\, 17 - 2$

$5 \,|\, 3 - 3$

0

Step 2

$.8 \times 5 = 4.0 \rightarrow 4$

combining step 1 and 2

$(32.4)_5$

2. $(284.8888)_{10} = (\quad)_9$

Step 1

$$\begin{array}{l|ll} 9 & 284 & -5 \\ 9 & 31 & -4 \\ 9 & 3 & -3 \\ & 0 & \end{array}$$

Step 2

$.8888 \times 9 = 7.9992 \rightarrow 7$

$.9992 \times 9 = 8.9928 \rightarrow 8$

$.9928 \times 9 = 8.9352 \rightarrow 8$

$.9352 \times 9 = 8.4168 \rightarrow 8$

combining step 1 and 2 -

$(345.7888)_9$

3. $(404.5)_{10} = (\quad)_{12}$

Step 1

$$\begin{array}{l|ll} 12 & 404 & -8 \\ 12 & 33 & -9 \\ 12 & 2 & -2 \\ & 0 & \end{array}$$

Step 2

$0.5 \times 12 = 6.0$

combining step 1 and 2 -

$= (298.6)_{12}$

Scanned with OKEN Scanner

Binary Arithmatic :-

Binary Addition :-

→ Binary addition is similar to decimal addition as it has two digits 0 and 1 we have to fo 4 possible cases for addition.

Case 1 = $0+0=0$
Case 2 = $0+1=1$
Case 3 = $1+0=1$
Casy 4 = $1+1=10$, so we have to put 0 a the result and carry over to the previous digit.

1) Step 1

$$\begin{array}{r} 10111 \\ +\ 01101 \\ \hline 100100 \end{array}$$

Step 2

Check -

$2^4+2^2+2^1+2^0 = 16+4+2+1 = 23$

$2^3+2^2+2^1+2^0 = 8+4+1 = 13$

$$\begin{array}{r} 23 \\ 13 \\ \hline 36 \end{array}$$

$2^5+2^2 = 32+4 = 36$

2)

$$\begin{array}{r} 1001 \\ +\ 1110 \\ \hline 10111 \end{array}$$

$2^3+\cancel{2^2}+2^0 = 8+1 = 9$

$2^3+2^2+2^1 = 8+4+2 = 14$

$2^4+2^2+2^1+2^0 = 16+4+2+1 = 23$

$$\begin{array}{r} 9 \\ 14 \\ \hline 23 \end{array}$$

3)

$$\begin{array}{r} 100101 \\ 1101111 \\ \hline 10010100 \end{array}$$

$2^5+2^2+2^0 = 32+4+1 = 37$

$2^6+2^5+2^3+2^2+2^1+2^0 = 64+32+8+4+2+1$

$= 111$

$$\begin{array}{r} 111 \\ 37 \\ \hline 148 \end{array}$$

$2^7+2^4+2^2 = 128+16+4$

$= 148$

1)
$$\begin{array}{r} 11110011 \\ +\ 1100111 \\ \hline 101011010 \end{array}$$

$2^7+2^6+2^5+2^4+2^1+2^0 = 128+64+32+16+2+1 = 243$

$2^6+2^5+2^2+2^1+2^0 = 64+32+4+2+1 = 103$

$$\begin{array}{r} 103 \\ 243 \\ \hline 346 \end{array}$$

$2^8+2^6+2^4+2^3+2^1$
$= 256+64+16+8+2 = 346$

Binary Subtraction :-

Case 1 = 0 − 0 = 0
Case 2 = 1 − 0 = 1
case 3 = 1 − 1 = 0
case 4 = 0 − 1 = ~~0~~ 1, 1 is borrowed from previous digit and 0 becomes 10, Now subtract, put 1 in the result as 10 − 1 = 1

1) Step 1

$$\begin{array}{r} 1110001 \\ -\ 111001 \\ \hline 0111000 \end{array}$$

Step 2

$2^6+2^5+2^4+2^0 = 64+32+16+1 = 113$
$2^5+2^4+2^3+2^0 = 32+16+8+1 = 57$

$$\begin{array}{r} 113 \\ 57 \\ \hline 56 \end{array}$$

$2^5+2^4+2^3$
$= 32+16+8 = 56$

2)
$$\begin{array}{r} 1000 \\ -\ 0011 \\ \hline 0101 \end{array}$$

$2^3 = 8$
$2^1+2^0 = 2+1 = 3$

$$\begin{array}{r} 8 \\ -\ 3 \\ \hline 5 \end{array}$$

$2^2+2^0 = 4+1 = 5$

3)
$$\begin{array}{r} 110110 \\ -\ 011101 \\ \hline 011001 \end{array}$$

$2^5+2^4+2^2+2^1 = 32+16+4+2 = 54$
$2^4+2^3+2^2+2^0 = 16+8+4+1 = 29$

$$\begin{array}{r} 54 \\ -\ 29 \\ \hline 25 \end{array}$$

$2^4+2^3+2^0 = 16+8+1 = 25$

4) 11100111

 1111001

0 0 1 1 0 1 1 1 0

$\rightarrow 2^6 + 2^5 + 2^3 + 2^2 + 2^1$

$= 64 + 32 + 8 + 4 + 2 = 110$

$2^6 + 2^5 + 2^4 + 2^3 + 2^0 = 64 + 32 + 16 + 8 + 1 = 121$

Binary Multiplication :

case 1 $0 \times 0 = 0$

case 2 $0 \times 1 = 0$

case 3 $1 \times 0 = 0$

case 4 $1 \times 1 = 01$

1) 1) $(10111)_2 \times (101)_2 = (\quad)_2$

```
   10111
 x   101
 -------
   10111
  00000
 10111
 -------
(1110011)_2
```

$2^4 + 2^2 + 2^1 + 2^0 = 16 + 4 + 2 + 1 = 23$

$2^2 + 2^0 = 4 + 1 = 5$

$23 \times 5 = 115$

$2^6 + 2^5 + 2^4 + 2^1 + 2^0$

$= 64 + 32 + 16 + 2 + 1$

$= 115$

2) 2)

```
      1110011
 x       1111
 ------------
      1110011
     1110011
    1110011
   1110011
 ------------
(11010111101)_2
```

$2^6 + 2^5 + 2^4 + 2^1 + 2^0 = 64 + 32 + 16 + 2 + 1 =$

$2^3 + 2^2 + 2^1 + 2^0 = 8 + 4 + 2 + 1 =$

$115 \times 15 = 1725$

3)

$2^{10} + 2^9 + 2^7 + 2^5 + 2^4 + 2^3 + 2^2 + 2^0$

$= 1024 + 512 + 128 + 32 + 16 + 8 + 4 + 1$

$= 1725$

) 1100110
× 110

0000000
1100110
1100110

$(1001100100)_2$

$2^6+2^5+2^2+2^1 = 64+32+4+2 = 102$

$2^2+2^1 = 4+2 = 6$

$102 \times 6 = 612$

$2^9+2^6+2^5+2^2$
$= 512+64+32+4 = 612$

) 1111
× 1111

1111
1111
1111
1111

$(11100001)_2$

$2^3+2^2+2^1+2^0 = 8+4+2+1 = 15$
$2^3+2^2+2^1+2^0 = 15$

$15 \times 15 = 225$

$2^7+2^6+2^5+2^0 = 128+64+1+32$
$= 225$

Binary Division :

case1 - $0 \div 1 = 0$
case2 - $1 \div 1 = 1$

1) $(11000)_2 \div (1000)_2$

```
        11
1000 | 11000
       1000
       -----
        1000
        1000
       -----
           0
```

$11000 = 2^4+2^3 = 24$
$1000 = 2^3 = 8$
$11 = 2^1+2^0 = 2+1 = 3$

$24 \div 8 = 3$

2) $(1110100)_2 \div (111010)_2$

```
          10
111010 | 1110100
         111010
         -------
              00
               0
              --
               0
```

$1110100 = 2^6+2^5+2^4+2^2 = 64+32+16+4 = 116$

$111010 = 2^5+2^4+2^3+2^1 = 32+16+8+2 = 58$

$10 = 2^1 = 2$

$116 \div 58 = 2$

3) $(1111000)_2 \div (100)_2$

```
        11110
100 ) 1111000
      100
      ---
       111
       100
       ---
        110
        100
        ---
         100
         100
         ---
          00
           0
          --
           0
```

$1111000 = 2^6 + 2^5 + 2^4 + 2^3$
$= 64 + 32 + 16 + 8$

$100 = 2^2 = 4$

$11110 = 2^4 + 2^3 + 2^2 + 2^1$
$= 16 + 8 + 4 + 2$
$= 30$ //

Signed Magnitude Representation :-

→ Signed numbers in binary number system be represented in adding an extra bit at left most end or before the most significant bits (MSB).

→ The signed bit 0 is used for + sign.
1 is used for − sign.

e.g →

		signed bit
+7 →	0, 111	0 (+)
−7 →	1, 111	1 (−)
−6 →	1, 110	1 (−)
−25 →	1, 11001	1 (−)
+13 →	0, 1101	0 (+)

→ Some times the sum of positive and negative a number is not zero.

e.g → Take two 8 bit numbers -

+1 → 0, 0000001
−1 → +1, 0000001

$+1 + (-1) = 0$

1, 0000010 = −2 //

Hence it proves that the result is incorrect.

In reality $+1-1=0$. Hence the arithmatic operations of binary digits may not provide always correct results.
Hence those need to be converted to unsigned binary format and then performed arithmatic calculations.
For this purpose we can use complementary representation of a negative number.

1's complement Representation :-

- The 1's complement of a binary number can be obtained by changing each 0 to a 1 and each 1 to a 0.
- This complemented value represents the negative or reverse of the original number.
- In digital system 1's complement of a number can be generated easily using inverters.
- 1's complement of $+6 \rightarrow 0,110$

  $-6 \rightarrow 1,001$

- The relation to obtain the complement is

$$= 1, (2^n-1-x)$$

$x \rightarrow$ number
$n \rightarrow$ no. of digit

i) $+6 = 0,110 = 1, (2^3-1-110)$

$$= 1, (1000-1-110)$$
$$= 1 (111-110)$$
$$= 1, (001)$$
$$1, 001$$

ii) $+12 = 0, 1100$ $\quad$ $+12 = 0, 1100 = 1, (2^4-1-1100)$

$-12 = 1, 0011$ $\quad$ $= 1, (10000-1-1100)$

$$= 1, (0011)$$
$$= 1, 0011$$

Similarly, we can ...

Decimal	1's complement in binary format	Decimal	Complement
+0	0,000	-0	1,111
+1	0,001	-1	1,110
+2	0,010	-2	1,101
+3	0,011	-3	1,100
+4	0,100	-4	1,011
+5	0,101	-5	1,010
+6	0,110	-6	1,001
+7	0,111	-7	1,000

Addition and Subtraction using 1's Complement

i) Addition :-

1) Add +4 and +9

Represent the numbers in 8 bit

$$\begin{array}{rl} +4 & = 0,0000100 \\ +9 & = 0,0001001 \\ \hline +13 & \ 0,0001101 = +13 \end{array}$$

2) Add +9 and +10

Represent the numbers in 8 bit

$$\begin{array}{rl} +9 & = 0,0001001 \\ +10 & = 0,0001010 \\ \hline +19 & \ 0,0010011 = 2^4 + 2^1 + 2^0 \end{array}$$

$= 16 + 2 + 1$

$= 19$

3) Add +25 and +38

+25	=	0,0011001
+38	=	0,0100110
+63		0,0111111

$= 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0$

$= 32 + 16 + 8 + 4 + 2 + 1$

$= 63$

2|38 − 0
2|19 − 1
2|9 − 1
2|4 − 0
2|2 − 0
2|1 − 1
0

2|25 − 1
2|12 − 0
2|6 − 0
2|3 − 1
2|1 − 1
0

ii) Subtraction :-

1) Subtract 7 from 21

= 21 − 7
= 21 + (−7)

+7 = 0,0000111
−7 = 1,1111000

2|21 − 1
2|10 − 0
2|5 − 1
2|2 − 0
2|1 − 1
0

Now . +21 = 0,0010101
−7 = 1,1111000
+ [1]0000 1101
→ 1
0,0001110

+21
−7
+14

$2^3 + 2^2 + 2^1$
$= 8 + 4 + 2 = 14$

2) Subtract 15 from 35

35 − 15
35 + (−15)

+35 = 0,0100011
+15 = 0,0001111
−15 = 1,1110000

2|15 − 1
2|7 − 1
2|3 − 1
2|1 − 1
0

2|35
17

+35 = 0,0100011
−15 = 1,1110000
+20
[1]0001 0011
→ 1
00010100

$2^4 + 2^2 = 20$

3) Subtract 13 - 9

13 - 9

= 13 + (-9)

$$
\begin{array}{r|l}
2 & 13 \quad 1 \\
2 & 6 \quad 0 \\
2 & 3 \quad 1 \\
2 & 1 \quad 1 \\
 & 0
\end{array}
$$

+13 = 0,0001101

+9 = 0,0001001

-9 = 1,1110110

$$
\begin{array}{rrl}
+13 & = & 0,0001101 \\
-9 & = & 1,1110110 \\
\hline
4 & & \boxed{1}\,0\,0\,0\,0\,0\,0\,1\,1 \\
 & & \qquad\qquad\quad\;\; 1 \\
\hline
2^2 = 4_{10} & & 0,0000100
\end{array}
$$

→ If sum of two numbers doesn't exceed 15 we can use 4 bit operand. If it doesn't exceeds 255 then we can use 8 bits operands.

4) perform 13 − 14 (one positive number & other negative large number)
or
subtract 14 from 13.

13 − 14

13 + (−14)

+13 = 0,0001101

+14 = 0,0001110

−14 = 1,1110001

$$
\begin{array}{r|l}
2 & 13 \quad 1 \\
2 & 6 \quad 0 \\
2 & 3 \quad 1 \\
2 & 1 \quad 1 \\
 & 0
\end{array}
\qquad
\begin{array}{r|l}
2 & 14 \\
2 & 7 \\
2 & 3 \\
2 & 1 \\
 & 0
\end{array}
$$

$$
\begin{array}{rrl}
+13 & = & 0,0001101 \\
-14 & = & 1,1110001 \\
\hline
-1 & & 1,1111110
\end{array}
$$

Now find complement of the result and then put a −ve sign.

$$
\begin{array}{l}
1\,1\,1\,1\,1\,1\,1\,0 \\
0\,0\,0\,0\,0\,0\,0\,1 \\
= -1
\end{array}
$$

) subtract -9 and 4

$-9-4$

$= (-9)+(-4)$

$+9 = 1001$

$-9 = 0110$

$+4 = 0100$

$-4 = 1011$

$$\begin{array}{rr} -9 = & 0110 \\ +\ -4 = & 1011 \\ \hline -13 & \boxed{1}0001 \\ & \to 1 \\ \hline & 0010 \end{array}$$

Now find complement of the result & put $-$ve sign -

$= 1101$

$= -13$

6) Subtract -41 and 24

$-41-24$

$= (-41)+(-24)$

$+41 = 0,0101001$

$-41 = 1,1010110$

$+24 = 0,0011000$

$-24 = 1,1100111$

2	41	1
2	20	0
2	10	0
2	5	1
2	2	0
2	1	1
	0	

2	24	
2	12	
2	6	
2	3	
2	1	
	0	

$$\begin{array}{rr} -41 = & 1,1010110 \\ -24 = & 1,1100111 \\ \hline -65 & \boxed{1}10111101 \\ & \to 1 \\ \hline & 10111110 \end{array}$$

write the complement - $01000001 = 2^6 + 2^0$

$= 64+1$

$= -65$

6. Subtract 19 and -49

19 - (-49)

= 19 + 49

2	19	1
2	9	1
2	4	0
2	2	0
2	1	1
	0	

19 = 0,0010011

49 = 0,0110001

---

68   0,1000100

$2^6 + 2^2 = 64 + 4$

$= 68$ //

2's Complement :-

→ 2's complement of a number can be generated by finding the 1's complement of a number and add 1 to the LSB of the result.

e.g.→ 1101 find 2's complement.

Now 1101

Complement 0010

+ 1

0011

Find it's 2's complement using 5 bit register.

01101

complement = 10010

+ 1

10011

Addition and Subtraction Using 2's Complement :

Step-1

First find the 2's complement of the given negative number.

Step-2

Sum of the number with the given positive number.

Step-3

a) If sum generates a carry bit 1 then the number is a positive number and the carry bit will be discarded and remaining bits are the final result.

b) If the positive number is added with a large negative then the result will be negative.

Hence we have to find the 2's complement of the final result and put a -ve sign.

c) If two negative numbers are added then the carry bit will be discarded and then the 2's complement of the bits with a negative sign will be final result.

1) a) 1101 − 1001

13 = 1101 ⟶ Minuend

−9 = −1001 ⟶ Subtraend

1101 + (−1001)

Step-1 find 2's complement of -ve number.

−1001

complement 0110

$$\begin{array}{r} 0110 \\ +\quad 1 \\ \hline 0111 \end{array}$$

Step-2 Add the 2's complement result with the minuend

$$\begin{array}{r} 1101 \\ 0111 \\ \hline 10100 \end{array}$$

Step-3 As the result is +ve, discard the ca bit and note down the result.

= 0100

Check : $\begin{array}{r} 1101 \\ -1001 \\ \hline 0100 \end{array} = \begin{array}{r} 13 \\ -9 \\ \hline 4 \end{array}$

b) $\begin{array}{r} 10101 \\ -00111 \\ \hline \end{array}$

21 = 10101

−7 = −00111

Step-1 find complement of negative number

−00111

= 11000

$\begin{array}{r} 11000 \\ +\quad 1 \\ \hline \end{array}$

2's complement 11001

Step-2 $\begin{array}{r} 11001 \\ +10101 \\ \hline \boxed{1}01110 \end{array}$

Step-3 01110

Check $\begin{array}{r} 10101 \\ -00111 \\ \hline 01110 \end{array}$ $\begin{array}{r} 21 \\ -7 \\ \hline 14 \end{array}$

c) $\begin{array}{rr} 110110 & -54 \\ -011010 & -26 \end{array}$

110110 − 011010

= 110110 + (− 011010)

Step-1 Find complement of negative number

011010

complement 100101

$$\begin{array}{r} 100101 \\ + \quad\quad 1 \\ \hline 100110 \end{array}$$ 2's complement

Step-2 Add complement with Minuend.

$$\begin{array}{r} 100110 \\ + \ 110110 \\ \hline \boxed{1}011100 \end{array}$$

Step-3 As the result is positive discard the carry bit & note down the result.

011100

check :

$$\begin{array}{r} 110110 \\ - \ 011010 \\ \hline 011100 \end{array}$$

2. Addition of a positive number with a large negative number.

a) 1101 − 1110

1101 + (−1110)

1101 = 13

1110 = −14

Step-1 Find complement of −ve number.

1110

compl : 0001

$$\begin{array}{r} 0001 \\ + \quad 1 \\ \hline 0010 \end{array}$$

Step-2 Add the complement with minuend.

$$\begin{array}{r} 0010 \\ + \ 1101 \\ \hline 1111 \end{array}$$

Step-3 As the result is -ve we have to find 2's complement of the result and put negative sign.

Num = 1111

compl = 0000

$$\begin{array}{r} 0000 \\ +\quad 1 \\ \hline 0001 \end{array}$$

Ans = - 0001

check :

$$\begin{array}{r} 1110 \\ -1101 \\ \hline 0001 \end{array} \qquad \begin{array}{r} -14 \\ 13 \\ \hline -1 \end{array}$$

b)
$$\begin{array}{r} 10101 \\ -\ 10111 \end{array} \qquad \begin{array}{r} = 21 \\ = -23 \\ \hline -2 \end{array}$$

Step-1 find 2's complement of -ve no.

10111

compl = 01000

$$\begin{array}{r} 01000 \\ +\quad 1 \\ \hline 01001 \end{array}$$ 2's complement.

Step-2 Add the complement with minuend.

$$\begin{array}{r} 01001 \\ +\ 10101 \\ \hline 11110 \end{array}$$

Step-3 As the result is -ve we have to find 2's complement of the result and put a negative sign.

Num = 11110

compl = 00001

$$\begin{array}{r} 00001 \\ +\quad 1 \\ \hline 00010 \end{array} \qquad \begin{array}{r} 21 \\ -23 \\ \hline -2 \end{array}$$

check :

$$\begin{array}{r} 10111 \\ 10101 \\ \hline 00010 \end{array}$$

) 11101 = 29 11101 − 11111

− 11111 : −31 11101 + (−11111)

Step-1 Find 2's complement of −ve no.

11111

Comp. 00000

$$\begin{array}{r} 00000 \\ + \quad 1 \\ \hline 00001 \end{array}$$

Step-2 Add the complement with minuend.

$$\begin{array}{r} 00001 \\ + \ 11101 \\ \hline 11110 \end{array}$$

Step-3 As the result is −ve we have to find 2's complement of the result and put a negative sign.

$$\begin{array}{r} 11110 \\ \text{Compl. } 00001 \\ + \quad 1 \\ \hline 00010 \end{array}$$

Check

$$\begin{array}{r} 11111 \\ -11101 \\ \hline 00010 \end{array} \qquad \begin{array}{r} 29 \\ -31 \\ \hline -2 \end{array}$$

3. Add two negative numbers using 2's compl.

(a) −1101 using a 5 bit register.
−1110

−01101 + 01110

Step-1 Find 2's complement of negative numbers.

	−01101		−01110
Compl =	10010	compl =	10001
+	1	+	1
	10011		10010

Step-2 Add the complements −

$$
\begin{array}{r}
10011 \\
+\ 10010 \\
\hline
\boxed{1}00101
\end{array}
$$

Step-3 Discared the carry bit, find 2's complem of the result and put a −ve sign.

$$
\begin{array}{rr}
 & 00101 \\
\text{comp} = & 11010 \\
+ & 1 \\
\hline
 & 11011
\end{array}
$$

Ans = −11011

Check

$$
\begin{array}{r}
-1110 \\
-1101 \\
\hline
11011
\end{array}
\qquad
\begin{array}{r}
= -13 \\
= -14 \\
\hline
-27
\end{array}
$$

(b) − 110101
− 101101

− 0110101
− 0101101

Step-1 Find complement (2's) of −ve number.

	−0110101		−0101101
Compl =	1001010	compl =	1010010
+	1	+	1
	1001011		1010011

Step-2 Add two ...

$$\begin{array}{r} 1001011 \\ +\ 1010011 \\ \hline \boxed{1}0011110 \end{array}$$

Step-3 Discard the carry bit and find 2's complement of the result & put a −ve sign.

$$\begin{array}{rr} & 0011110 \\ \text{compl.} & 1100001 \\ + & 1 \\ \hline & 1100010 \end{array}$$

Check = $\begin{array}{r} -110101 \\ -101101 \\ \hline 1100010 \end{array}$ $\begin{array}{r} = -53 \\ = -45 \\ \hline -98 \end{array}$

c) $\begin{array}{r} -11110 \\ -10111 \end{array}$ $\begin{array}{r} = -30 \\ = -23 \end{array}$

$$\begin{array}{r} -011110 \\ -010111 \end{array}$$

Step-1 Find 2's complement of −ve number.

$$\begin{array}{rr} & -011110 \\ \text{compl} = & 100001 \\ + & 1 \\ \hline & 100010 \end{array} \qquad \begin{array}{rr} & -010111 \\ \text{compl} = & 101000 \\ + & 1 \\ \hline & 101001 \end{array}$$

Step-2 Add the complements

$$\begin{array}{r} 100010 \\ +\ 101001 \\ \hline \boxed{1}001011 \end{array}$$

Step-3 Find 2's complements of the result & put a −ve sign.

$$\begin{array}{rr} & 001011 \\ \text{comp} = & 110100 \\ + & 1 \\ \hline & 110101 \end{array}$$

Check = $\begin{array}{r} -11110 \\ -10111 \\ \hline -110101 \end{array}$ $\begin{array}{r} = -30 \\ = -23 \\ \hline -53 \end{array}$

Binary Coded Decimal :-

Binary coded decimal is a system of writing numerals that assigns a four digit binary code to each digit 0 through 9 in a decimal $( )_{10}$ number.

BCD is a way to convert decimal numbers into their binary equivalents.

It is not same as binary representation.

Each digit in a decimal $( )_{10}$ number is represented as a group of 4 binary digits or bits.

- Each digit is encoded separately.
- The full number is first segregated into individual digits & then their 4 bit binary
- coded decimal is represented.
- The BCD representation can be done using either 4 bit or 8 bit equivalence to perform their requirements of arithmatic operations.

Decimal	Binary	BCD
0	0000	0000
⋮		
9	1001	1001
10		0001 0000
11		0001 0001
⋮		
100		00010000 0000
⋮		
1000		0001 0000 0000 0000

Q. 97

1) $(238)_{10}$

BCD = $\underbrace{0010}_{2}$ $\underbrace{0011}_{3}$ $\underbrace{100}_{8}$

2) $(1764)_{10}$

BCD = $\underbrace{0001}_{1}$ $\underbrace{0111}_{7}$ $\underbrace{0110}_{6}$ $\underbrace{0100}_{4}$

3) $(95032)_{10}$

BCD = $\underbrace{1001}_{9}$ $\underbrace{0101}_{5}$ $\underbrace{0000}_{0}$ $\underbrace{0011}_{3}$ $\underbrace{0010}_{2}$

Advantages of BCD Representation :

→ It solves size limitations of integer arithmetic

→ It enables easy conversion between machine readable and human readable numerals.

→ Compared to binary system BCD is easy to code and decode.

→ It provides a fast and efficient way to convert decimal numbers into binary.

→ It is useful in digital display.

→ BCD is also used in some currency applications where floating point representations are not completely accurate.

o

0100
4

2 0011 0010
3 2

esentation :-

s of integer arithmatic
ion between machine
eadable numerals.
tem BCD is easy to

d efficient way to
into binary.
displays.
me currency applicat-
nt representations
ccurate.

character	BCD code Zone	BCD code Digit	octal equivalent
A	11	0001	61
B	11	0010	62
C	11	0011	63
D	11	0100	64
E	11	0101	65
F	11	0110	66
G	11	0111	67
H	11	1000	70
I	11	1001	71
J	10	0001	41
K	10	0010	42
L	10	0011	43
M	10	0100	44
N	10	0101	45
O	10	0110	46
P	10	0111	47
Q	10	1000	50
R	10	1001	51
S	01	0010	22
T	01	0011	23
U	01	0100	24
V	01	0101	25
W	01	0110	26
X	01	0111	27
Y	01	1000	30
Z	01	1001	31

character	Z
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

eg→ write th
word B

In BCD b

B = 1100
A = 110
S = 010
E = 110

Hence,

11001
B

will r

eg→ wi
for

In B

D =
I
G
Hence
octe

Character	BCD code		Octal equivalent
	Zone	Digit	
0	00	0000	00
1	00	0001	01
2	00	0010	02
3	00	0011	03
4	00	0100	04
5	00	0101	05
6	00	0110	06
7	00	0111	07
8	00	1000	10
9	00	1001	11

eg→ write the binary digits used to record the word BASE in BCD.

In BCD binary notation (refer to figure)

B = 110010
A = 110001
S = 010010
E = 110101

Hence, the Binary digits

$\underset{B}{\underline{110010}}$ $\underset{A}{\underline{110001}}$ $\underset{S}{\underline{010010}}$ $\underset{E}{\underline{110101}}$

will record the word ~~Ba~~ BASE in BCD.

eg→ Using octal notation, write the BCD coding for the word DIGIT.

In BCD octal notation (refer to figure)

D = 64   I = 71
I = 71   T = 27
G = 67

Hence, the BCD coding for the word DIGIT in octal notation will be = $\frac{64}{D}$ $\frac{71}{I}$ $\frac{67}{G}$ $\frac{71}{I}$ $\frac{23}{T}$

EBCDIC :

The major problem with BCD code is that it can represent only 64 (2^6) different characters. This is not sufficient for providing decimal numbers (10), lowercase (small) letters (26), uppercase (capital) letters (26), and a large no. of other special characters (28+).
Developed by IBM, most IBM models use EBCDIC code, many other computers also use EBCDIC code.

Character	EBCDIC code Zone	EBCDIC code Digit	Hexadecimal equivalent
A	1100	0001	C1
B	1100	0010	C2
C	1100	0011	C3
D	1100	0100	C4
E	1100	0101	C5
F	1100	0110	C6
G	1100	0111	C7
H	1100	1000	C8
I	1100	1001	C9
J	1101	0001	D1
K	1101	0010	D2
L	1101	0011	D3
M	1101	0100	D4
N	1101	0101	D5
O	1101	0110	D6
P	1101	0111	D7
Q	1101	1000	D8
R	1101	1001	D9

Character	EBC[DIC] Zone
S	1110
T	1110
U	1110
V	1110
W	1110
X	111[0]
Y	111[0]
Z	111
0	111
1	111
2	111
3	111
4	11
5	11
6	1
7	1
8	1
9	1

eg :-

① BIT =
$\underbrace{1100}_{\text{A}} \underbrace{0010}_{\text{B}}$ 11[...]

② BAKE =
$\underbrace{1100}_{\text{A}} \underbrace{0010}_{\text{B}}$ 11[...]

③ HOUSE =
$\underbrace{1100\ 1000}_{\text{H}}$ 11[...]

Character	EBCDIC Code Zone	Digit	Hexadecimal equivalent
S	1110	0010	E2
T	1110	0011	E3
U	1110	0100	E4
V	1110	0101	E5
W	1110	0110	E6
X	1110	0111	E7
Y	1110	1000	E8
Z	1110	1001	E9
0	1111	0000	F0
1	1111	0001	F1
2	1111	0010	F2
3	1111	0011	F3
4	1111	0100	F4
5	1111	0101	F5
6	1111	0110	F6
7	1111	0111	F7
8	1111	1000	F8
9	1111	1001	F9

e.g.

① BIT =

$\underbrace{1100\ 0010}_{B}$ $\underbrace{1100\ 1001}_{I}$ $\underbrace{1110\ 0011}_{T}$

② BAKE =

$\underbrace{1100\ 0010}_{B}$ $\underbrace{1100\ 0001}_{A}$ $\underbrace{1101\ 0010}_{K}$ $\underbrace{1100\ 0101}_{E}$

③ HOUSE =

$\underbrace{1100\ 1000}_{H}$ $\underbrace{1101\ 0110}_{O}$ $\underbrace{1110\ 0100}_{U}$ $\underbrace{1110\ 0010}_{S}$ $\underbrace{1100\ 0101}_{E}$

Scanned with OKEN Scanner

ASCII:

→ ASCII stands for American standard code for Information Interchange.

→ Several American computer manufacturers adopt ASCII for their internal code of computers.

→ This code is popular for data communication, internal data representation in microcomputers for high-end computer configurations.

→ ASCII has two types ASCII-7 and ASCII-8 where 7 bit or 8 bit code representation can be possible.

Character	ASCII code	Hexadecimal	Character	ASCII code	Hexadecimal
NUL	0000 0000	0	DLE	0001 0000	10
SOH	0000 0001	1	DC1	0001 0001	11
STX	0000 0010	2	DC2	0001 0010	12
ETX	0000 0011	3	DC3	0001 0011	13
EOT	0000 0100	4	DC4	0001 0100	14
ENQ	0000 0101	5	NAK	0001 0101	15
ACK	0000 0110	6	SYN	0001 0110	16
BEL	0000 0111	7	ETB	0001 0111	17
BS	0000 1000	8	CAN	0001 1000	18
HT	0000 1001	9	EM	0001 1001	19
LF	0000 1010	0A	SUB	0001 1010	1A
VT	0000 1011	0B	ESC	0001 1011	1B
FF	0000 1100	0C	FS	0001 1100	1C
CR	0000 1101	0D	GS	0001 1101	1D
SO	0000 1110	0E	RS	0001 1110	1E
SI	0000 1111	0F	US	0001 1111	1F

Character	ASCII code	Hexadecimal	Character
SP	0010 0000	20	@
!	0010 0001	21	A
"	0010 0010	22	B
#	0010 0011	23	C
$	0010 0100	24	D
%	0010 0101	25	E
&	0010 0110	26	F
'	0010 0111	27	G
(	0010 1000	28	H
)	0010 1001	29	I
*	0010 1010	2A	J
+	0010 1011	2B	K
,	0010 1100	2C	L
-	0010 1101	2D	
.	0010 1110	2E	
/	0010 1111	2F	
0	0011 0000	30	
1	0011 0001	31	
2	0011 0010	32	
3	0011 0011	33	
4	0011 0100	34	
5	0011 0101	35	
6	0011 0110	36	
7	0011 0111	37	
8	0011 1000	38	
9	0011 1001	39	
:	0011 1010	3A	
;	0011 1011	3B	
<	0011 1100	3C	
=	0011 1101	3D	
>	0011 1110	3E	
?	0011 1111	3F	

Character	ASCII code	Hexadecimal	Character	ASCII code	Hexadecimal
SP	0010 0000	20	@	0100 0000	40
!	0010 0001	21	A	0100 0001	41
"	0010 0010	22	B	0100 0010	42
#	0010 0011	23	C	0100 0011	43
$	0010 0100	24	D	0100 0100	44
%	0010 0101	25	E	0100 0101	45
&	0010 0110	26	f	0100 0110	46
'	0010 0111	27	G	0100 0111	47
(	0010 1000	28	H	0100 1000	48
)	0010 1001	29	I	0100 1001	49
*	0010 1010	2A	J	0100 1010	4A
+	0010 1011	2B	K	0100 1011	4B
,	0010 1100	2C	L	0100 1100	4C
-	0010 1101	2D	M	0100 1101	4D
.	0010 1110	2E	N	0100 1110	4E
/	0010 1111	2F	O	0100 1111	4F
0	0011 0000	30	P	0101 0000	50
1	0011 0001	31	Q	0101 0001	51
2	0011 0010	32	R	0101 0010	52
3	0011 0011	33	S	0101 0011	53
4	0011 0100	34	T	0101 0100	54
5	0011 0101	35	U	0101 0101	55
6	0011 0110	36	V	0101 0110	56
7	0011 0111	37	W	0101 0111	57
8	0011 1000	38	X	0101 1000	58
9	0011 1001	39	Y	0101 1001	59
:	0011 1010	3A	Z	0101 1010	5A
;	0011 1011	3B	[	0101 1011	5B
<	0011 1100	3C	\	0101 1100	5C
=	0011 1101	3D	]	0101 1101	5D
>	0011 1110	3E	^	0101 1110	5E
?	0011 1111	3F	_	0101 1111	5F

Character	ASCII code	Hexadecimal	Character	ASCII code	Hexadecimal
`	0110 0000	60	p	0111 0000	70
a	0110 0001	61	q	0111 0001	71
b	0110 0010	62	r	0111 0010	72
c	0110 0011	63	s	0111 0011	73
d	0110 0100	64	t	0111 0100	74
e	0110 0101	65	u	0111 0101	75
f	0110 0110	66	v	0111 0110	76
g	0110 0111	67	w	0111 0111	77
h	0110 1000	68	x	0111 1000	78
i	0110 1001	69	y	0111 1001	79
j	0110 1010	6A	z	0111 1010	7A
k	0110 1011	6B	{	0111 1011	7B
l	0110 1100	6C	|	0111 1100	7C
m	0110 1101	6D	}	0111 1110	7D
n	0110 1110	6E	~	0111 1110	7E
o	0110 1111	6F	DEL	0111 1111	7F

e.g. ASCII-7 coded representation for GIRL.

G = 100 0111
I = 100 1001
R = 101 0010
L = 100 1100

GIRL = 1000111 1001001 1010010 100 1100

ASCII-8 coded representation for GIRL.

G = 0100 0111
I = 0100 1001
R = 0101 0010
L = 0100 1100

$\underbrace{0100\ 0111}_{G}$ $\underbrace{0100\ 1001}_{I}$ $\underbrace{0101\ 0010}_{R}$ $\underbrace{0100\ 1100}_{L}$

ASCII-8 coded representation for John.

		Hexadecimal
J	0100 1010	4A
o	0110 1111	6F
h	0110 1000	68
n	0110 1110	6E

John = 0100 1010 0110 1111 0110 1000 0110 1110

Boolean Algebra

→ For simplifying th
→ of prepositional
was developed
→ Later in 1938
design and ana
represent act
circuits.
→ It can help to

Concepts of

Logical Additi

Input		Ou
0	0	
0	1	
1	0	
1	1	

Logical Multip

Input	
0	0
0	1
1	0
1	1

Logical Comp

Input
0
1

ASCII code	Hexadecimal
0111 0000	70
0111 0001	71
0111 0010	72
0111 0011	73
0111 0100	74
0111 0101	75
0111 0110	76
0111 0111	77
0111 1000	78
0111 1001	79
0111 1010	7A
0111 1011	7B
0111 1100	7C
0111 1110	7D
0111 1110	7E
0111 1111	7F

...tation for GIRL:

...0 100 1100

... for GIRL.

...11 0100 1001 0101 0010 0100 1100

I R L

...ion for John:

...exadecimal-

...0 1000 0110 1110

Boolean Algebra:

→ For simplifying the representation and manipulation of prepositional logic we can use algebra which was developed by George Bolle in 1815.

→ Later in 1938 Boolean algebra developed per design and analysis of switching circuits. To represent action relays in modern electronic circuits.

→ It can help to solve algebric expression.

Concepts of Boolean Algebra:

Logical Addition (OR):

Input		Output
0	0	0
0	1	1
1	0	1
1	1	1

Logical Multiplication (AND):

Input		Output
0	0	0
0	1	0
1	0	0
1	1	1

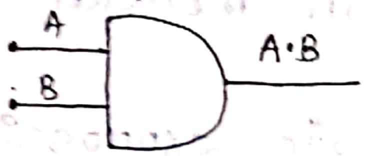
Logical complementation (NOT):

Input	Output
0	1
1	0

Logical AND :-

→ The boolean expression logical AND performs the logical multiplication or boolean product. where the boolean expression of A and B will be A * B or (A·B). If either of the input values is 0 then the output is zero.

→ The logic gate for AND operation :



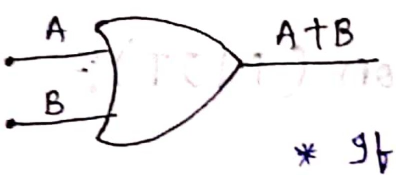
Truth Table :-

INPUT		OUTPUT
A	B	A·B
0	0	0
0	1	0
1	0	0
1	1	1

Logical OR :-

→ The boolean expression logical OR performs the logical addition or boolean sum, where the boolean expression of a A and B will be A + B.

→ The logic gate for OR operation :



Truth Table :-

INPUT		OUTPUT
A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

* If one or more inputs value are 1 then the output is 1.
- If all the inputs are zero then the output is zero.

→ The logical OR function compares two or more input conditions and generates true, if either one of the conditions is true.
→ It performs parallel operations.

Logical NOT :

→ Logical NOT operation is simply an inversion or complementary action of a boolean value.
→ Its output state is NOT.
→ The boolean expression of A will be $\bar{A}$ where bar (−) specify the NOT action. e.g. if $A = 0$, $\bar{A} = 1$
→ The logic gate for NOT operation :

Truth Table

INPUT	OUTPUT
0	1
1	0

Rules of Boolean Algebra :

Rule-1 : A = 0, if and only if A is not equal to 1.
A = 1, if and only if A is not equal to 0.

Rule-2 : $X + 0 = X$
$X * 1 = X$

Rule-3 : commutative Law :

$X + Y = Y + X$
$X \cdot Y = Y \cdot X$

Rule4 : Associative Law :

$X + (Y + Z) = (X + Y) + Z$
$X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z$

Rule 5 : Distributive Law :-

$$X\cdot(Y+Z) = X\cdot Y + X\cdot Z$$

Rule 6 : Inversion Law :-

$$X + \overline{X} = 1$$
$$X \cdot \overline{X} = 0$$

Rule 7 : Absorption Law :-

$$X\cdot(X+Y) = X$$
$$X + X\cdot Y = X$$

Rule 8 : DeMorgan's Law :-

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

complementation.

Rule 1 :-

$A = 0$, $A = 1$

$A = 1$, $A = 0$

If $A = 0$, $\overline{A} = 1$

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

e.g.→ If $X = 0$

$X\cdot\overline{X} = 0\cdot 1 = 0.$

$X + \overline{X} = 0 + 1 = 1$

Rule-2 :-

$X + 0 = X$

$X \cdot 1 = X$

e.g.→ for $X = 0$

then, $X + 0 = X$

for $X = 1$, $X\cdot 1 = X$

Scanned with OKEN Scanner

Rule-3 :-

$x + y = y + x$
$x \cdot y = y \cdot x$

e.g:-

x	y	$x+y$	$y+x$	$x \cdot y$	$y \cdot x$
0	0	0	0	0	0
0	1	1	1	0	0
1	0	1	1	0	0
1	1	1	1	1	1

Hence proved -
$x + y = y + x$
and $x \cdot y = y \cdot x$

Rule-4 :-

$x + (y+z) = (x+y) + z$
$x \cdot (y \cdot z) = (x \cdot y) \cdot z$

x	y	z	$(y+z)$	$(x+y)$	$(y \cdot z)$	$(x \cdot y)$	$(x+(y+z))$	$(x+y)+z$	$x \cdot (y \cdot z)$	$(x \cdot y) \cdot z$
0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	1	0	0
0	1	0	1	1	0	0	1	1	0	0
0	1	1	1	1	1	0	1	1	0	0
1	0	0	0	1	0	0	1	1	0	0
1	0	1	1	1	0	0	1	1	0	0
1	1	0	1	1	0	1	1	1	0	0
1	1	1	1	1	1	1	1	1	1	1

Hence, proved - $x + (y+z) = (x+y) + z$
$x \cdot (y \cdot z) = (x \cdot y) \cdot z$

Rule-5 :-

$x \cdot (y+z) = x \cdot y + x \cdot z$

e.g →

X	Y	Z	$Y+Z$	$X \cdot Y$	$X \cdot Z$	$X \cdot (Y+Z)$	$X \cdot Y + X \cdot Z$
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	0	1	1	1
1	1	0	1	1	0	1	1
1	1	1	1	1	1	1	1

Hence, proved -

$$X \cdot (Y+Z) = X \cdot Y + X \cdot Z$$

Rule-7 :-

$$X \cdot (X + Y) = X$$
$$X + X \cdot Y = X$$

e.g →

X	Y	$(X+Y)$	$(X \cdot Y)$	$X \cdot (X+Y)$	$X + X \cdot Y$
0	0	0	0	0	0
0	1	1	0	0	0
1	0	1	0	1	1
1	1	1	1	1	1

Hence, proved -

$$X \cdot (X+Y) = X$$
$$X \cdot X \cdot Y = X$$

Rule-8 :-

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

x	y	$\overline{x}$	$\overline{y}$	$x \cdot y$	$x + y$	$\overline{x \cdot y}$	$\overline{x} + \overline{y}$	$\overline{x + y}$	$\overline{x} \cdot \overline{y}$
0	0	1	1	0	0	1	1	1	1
0	1	1	0	0	1	1	1	0	0
1	0	0	1	0	1	1	1	0	0
1	1	0	0	1	1	0	0	0	0

Hence, proved -

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

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

Summary of Boolean Algebra laws :-

1. $\overline{0} = 1$
2. $\overline{1} = 0$
3. $\overline{\overline{X}} = X$
4. $X \cdot 0 = 0$
5. $X \cdot 1 = X$
6. $X \cdot X = X$
7. $X \cdot \overline{X} = 0$
8. $X + 0 = X$
9. $X + 1 = 1$
10. $X + X = X$
11. $X + \overline{X} = 1$
12. $X + Y = Y + X$
13. $X \cdot Y = Y \cdot X$
14. $X + (Y + Z) = (X + Y) + Z$
15. $X \cdot (Y \cdot Z) = (X \cdot Y) Z$
16. $X(Y + Z) = X \cdot Y + XZ$
17. $X + (Y \cdot Z) = (X + Y) \cdot (X + Z)$ (Distributive law)
18. $X + (\overline{X} \cdot Y) = X + Y$
19. $\overline{X + Y} = \overline{X} \cdot \overline{Y}$
20. $\overline{XY} = \overline{X} + \overline{Y}$

Solving using Boolean algebric Laws :-

1. $X \cdot Y \cdot Y \cdot Z \cdot Z \cdot Z$

$= X \cdot (Y \cdot Y) \cdot Z (Z \cdot Z)$ $\quad [\because X \cdot X = X]$

$= X \cdot Y \cdot Z \cdot Z$

$= X \cdot Y (Z \cdot Z)$

$= X \cdot Y \cdot Z$

2) $x \cdot x \cdot y + x \cdot y \cdot y + y \cdot \bar{y} \cdot x \cdot x$ $[\because (y \cdot \bar{y} = 0]$

$= (x \cdot x) y + x(y \cdot y) + 0 \cdot x$ $[\because x \cdot x = x]$

$= x \cdot y + x \cdot y + 0$ $[\because x + x = x]$

$= xy + xy$

3) $x(y + z(\overline{x \cdot y + x \cdot z}))$

$= x(y + z(\overline{x \cdot y} \cdot \overline{xz}))$

$= x(y + z\overline{xy} \cdot \overline{xz})$

$= xy + xz \cdot \overline{xy} \cdot \overline{xz}$

$= xy + \overline{xy}(xz \cdot \overline{xz})$ $[\because x \cdot \bar{x} = 0]$

$= xy + 0 = xy$

4) $x\bar{y} + \bar{x}y + xy + \bar{x} \cdot \bar{y}$

$= x\bar{y} + xy + \bar{x}y + \bar{x} \cdot \bar{y}$

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

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

$= x(1) + \bar{x}(1)$

$= x + \bar{x}$

$= 1$

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

$= x\bar{y}z \cdot (\bar{x} + y)$

$= \bar{x} \cdot x\bar{y}z + y \cdot x\bar{y}z$ $[\because \bar{x} \cdot x = 0]$

$= 0 \cdot \bar{y}z + 0 \cdot xz$

$= 0 + 0$

$= 0$

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

$= A \cdot A\bar{B}\bar{C} + 0 + A \cdot A\bar{B}C + 0 + A \cdot ABC + B \cdot BAC + A \cdot AB\bar{C} + B \cdot BA\bar{C}$

$= A\bar{B}\bar{C} + 0 + A\bar{B}C + ABC + BAC + AB\bar{C} + BA\bar{C}$

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

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

Scanned with OKEN Scanner

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

$= AB \cdot 1 + A\bar{B} \cdot 1$

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

$= A \cdot 1$

$= A$

Exclusive OR / XOR : $A \oplus B$

→ The boolean expression exclusive OR (XOR) is written as $A \oplus B$.

→ In exclusive OR if we have odd number of inputs are 1 then the output is 1, & even number of inputs are 1 then the output is zero.

output $A \oplus B = \bar{A} \cdot B + A \cdot \bar{B}$

→ The logic gate for $A \oplus B$

Truth table :

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

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

NAND Gate :

→ The boolean expression NAND performs the inversion on AND gate.

→ If the output of AND gate is zero then the result is zero.

→ If the output of AND gate is one then the result is zero.

→ Output $\overline{A \cdot B}$

The logic gate for $\overline{A \cdot B}$

Truth Table :

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

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

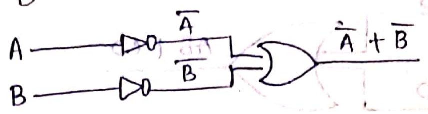
NAND performs the inversion

gate is zero then the result

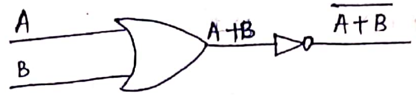
gate is one then the result

Construct the logic Diagram :

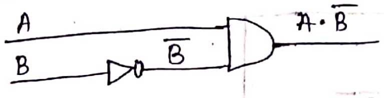
① $\overline{A} + \overline{B}$



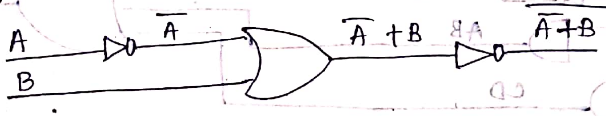
② $\overline{A+B}$



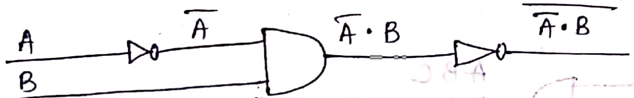
③ $A \cdot \overline{B}$



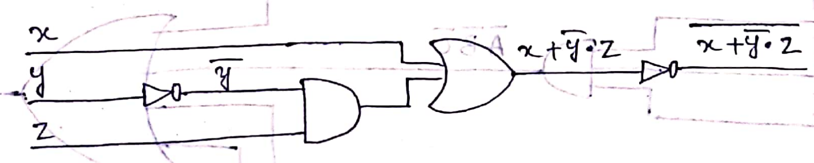
④ $\overline{\overline{A}+B}$



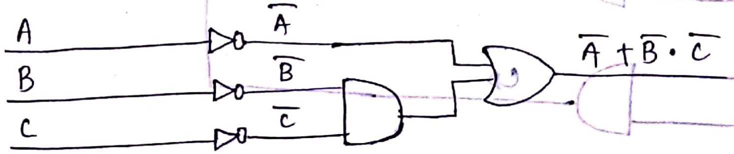
⑤ $\overline{\overline{A} \cdot B}$



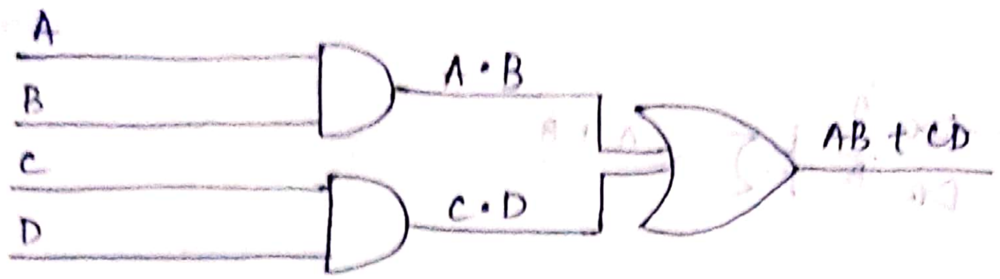
⑥ $\overline{x + \overline{y} \cdot z}$



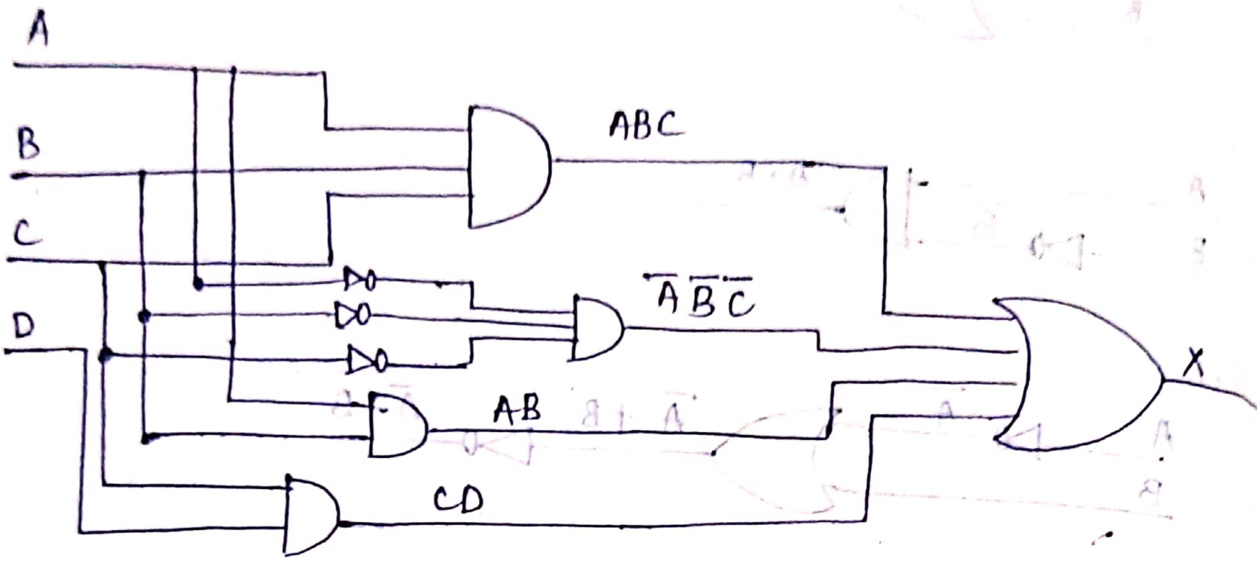
⑦ $\overline{A} + \overline{B} \cdot \overline{C}$



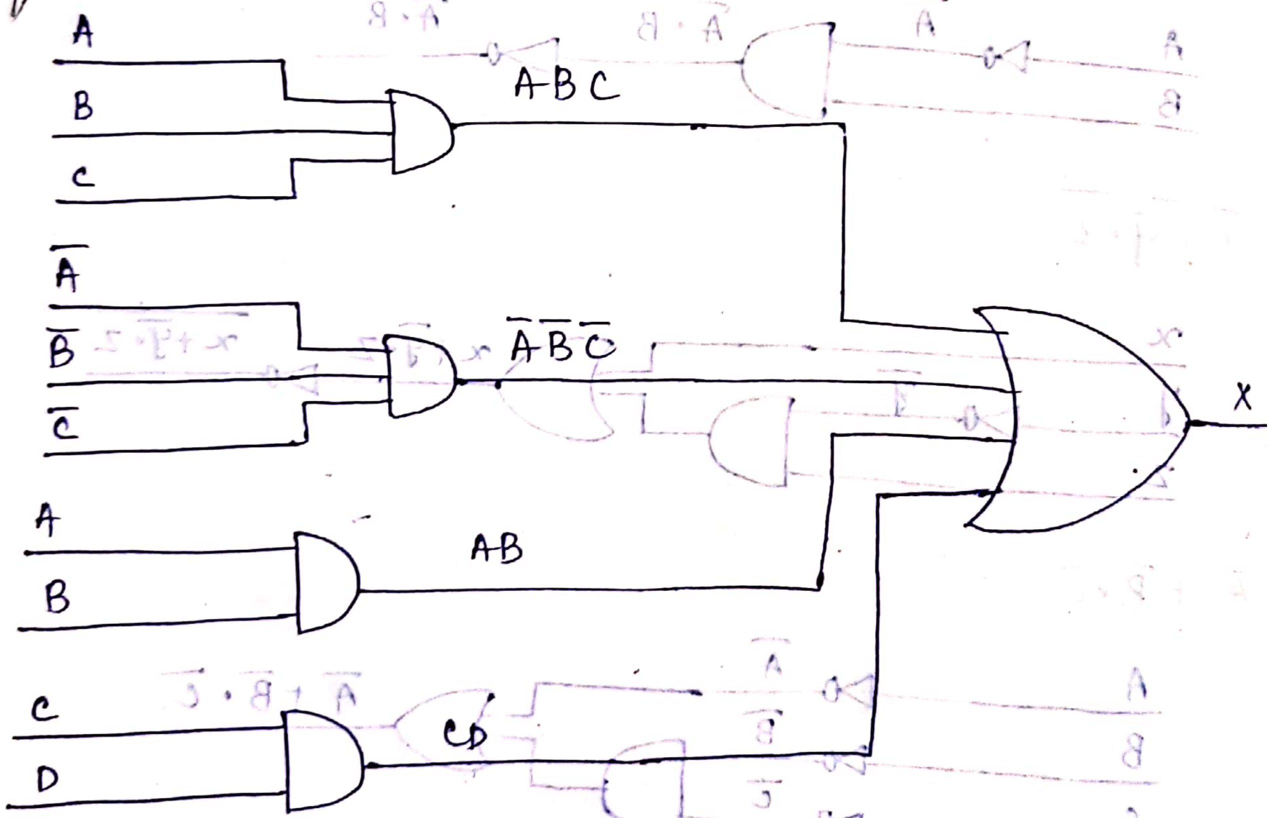
⑧ AB + CD



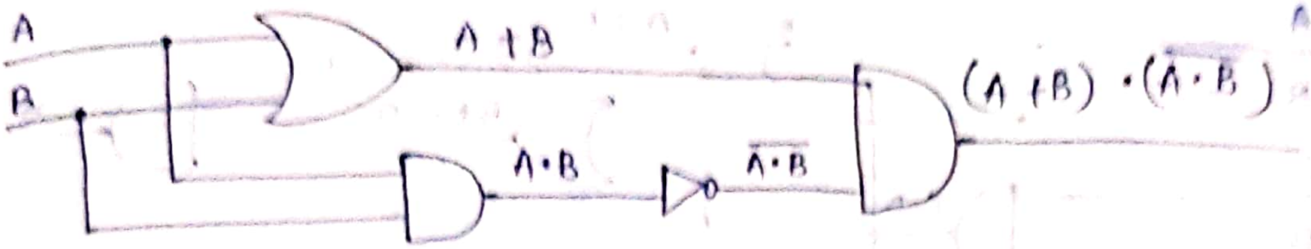
⑨ $X = ABC + \bar{A}\bar{B}\bar{C} + AB + CD$



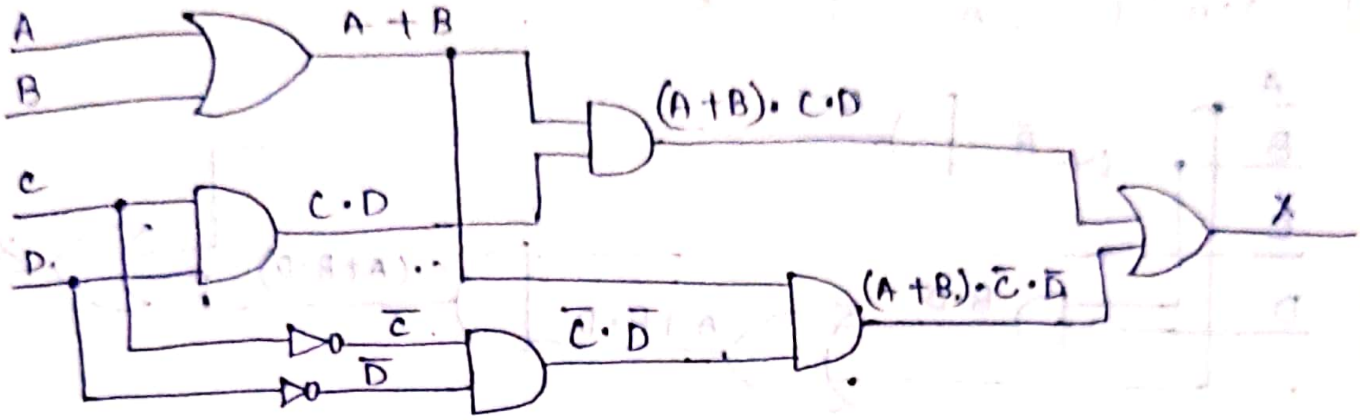
or



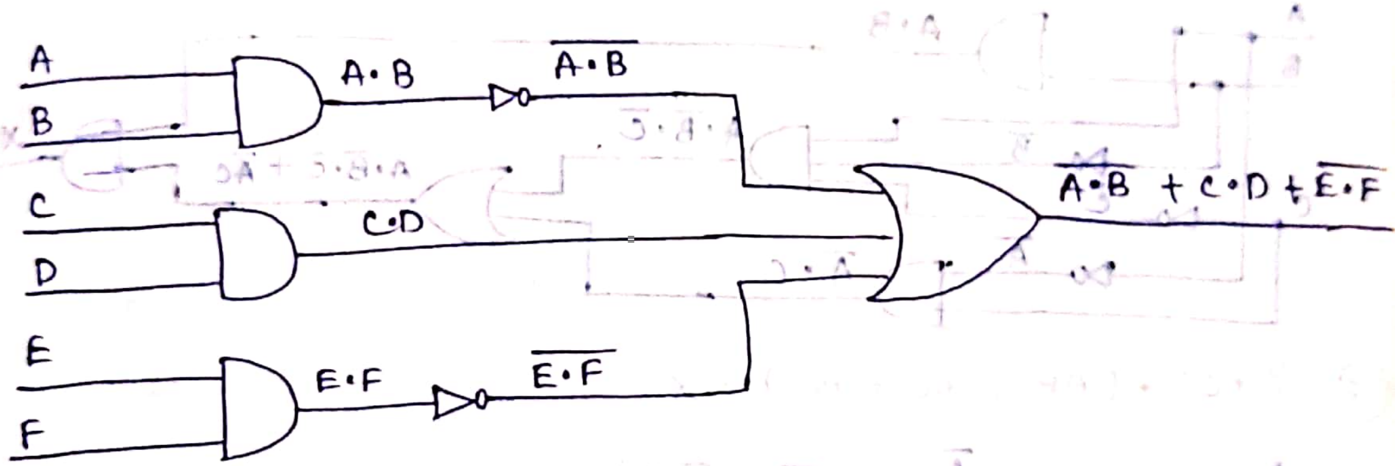
(10) $(A+B)\cdot(\overline{A\cdot B})$



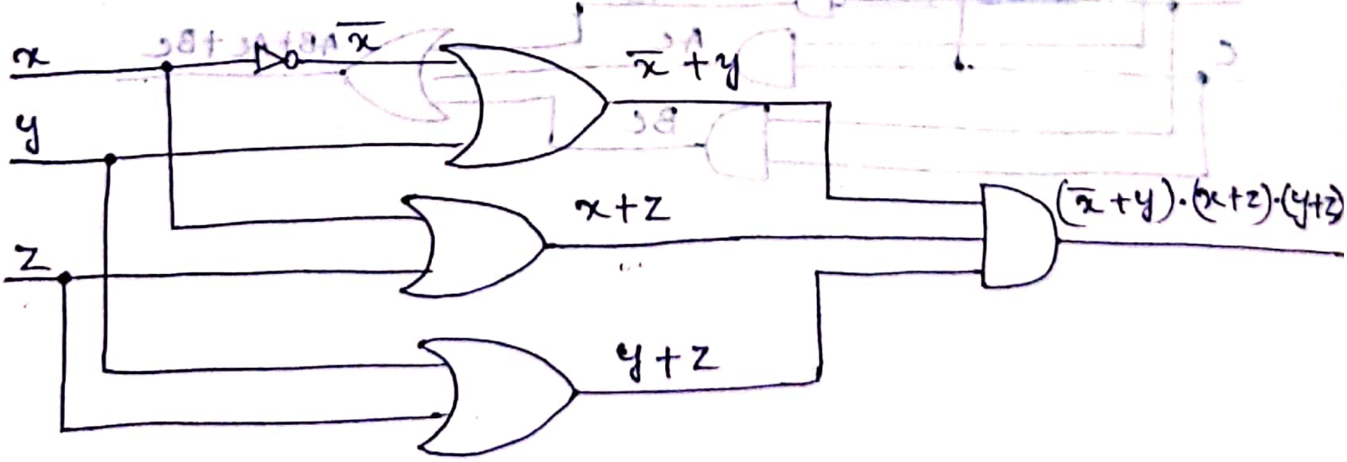
(11) $(A+B)\cdot C\cdot D+(A+B)\cdot\overline{C}\cdot\overline{D}=X$



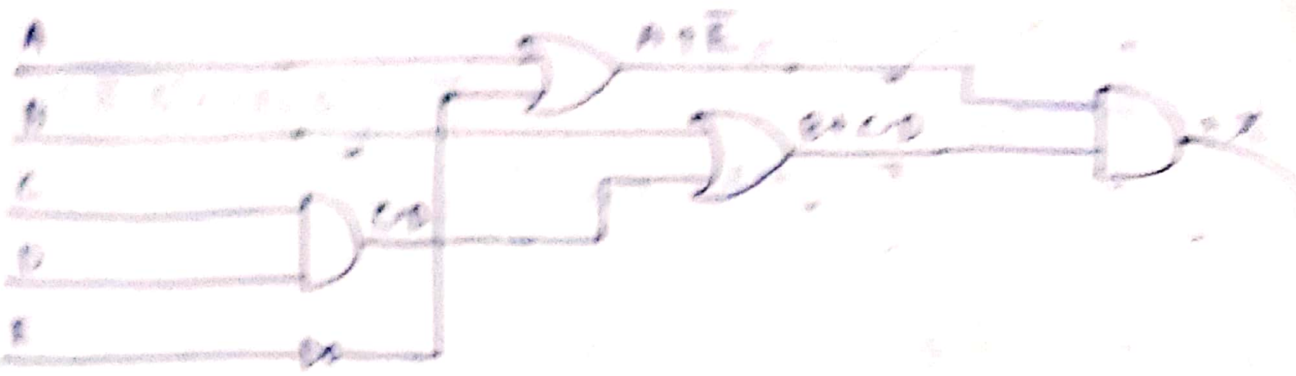
(12) $\overline{A\cdot B}+C\cdot D+\overline{E\cdot F}$



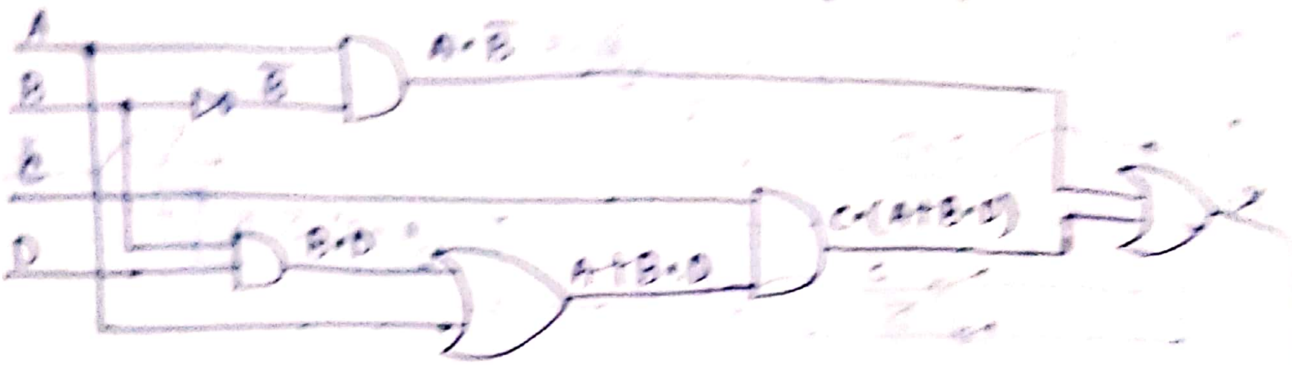
(13) $(\overline{x}+y)\cdot(x+z)\cdot(y+z)$



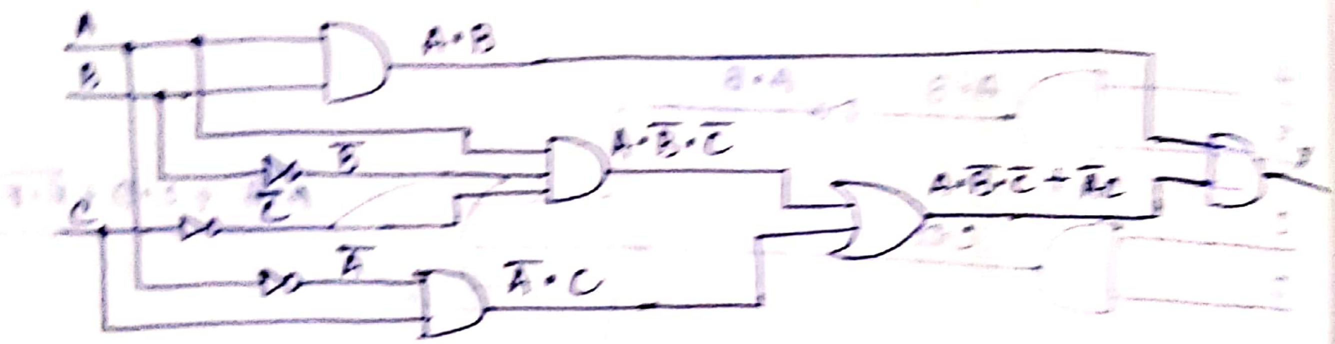
(14) $(A + \bar{E}) \cdot (B + C \cdot D) = X$



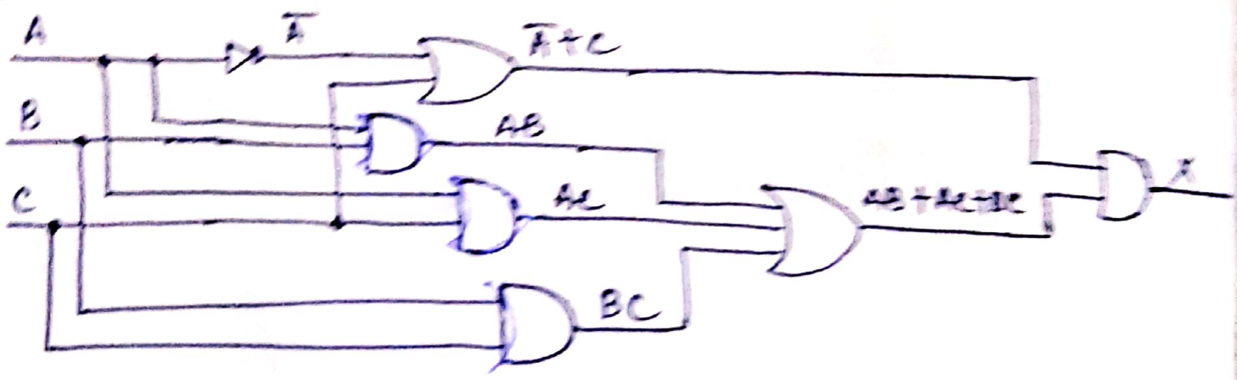
(15) $(A \cdot \bar{B}) + C \cdot (A + B \cdot D) = X$



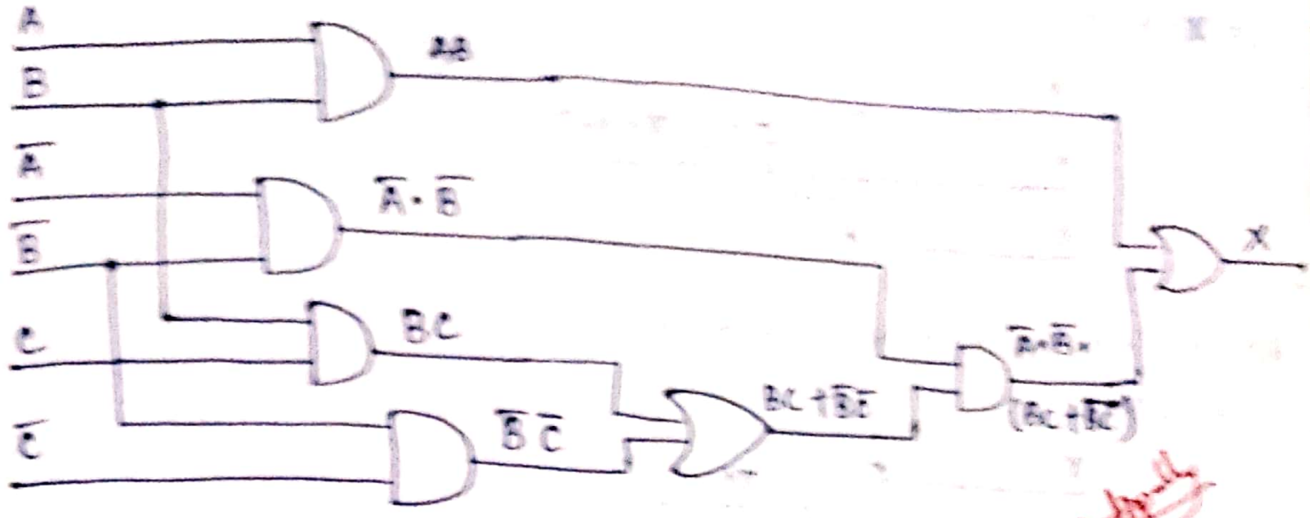
(16) $(A \cdot B) \cdot (A \cdot \bar{B} \cdot \bar{C} + \bar{A} C) = X$



(17) $(\bar{A} + C) \cdot (AB + AC + BC) = X$



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



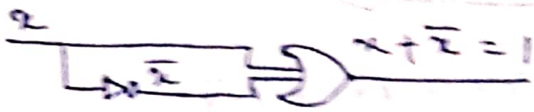
Minimise the term and draw the circuit diagram:

① $x(y+z(\overline{xy}+\overline{xz}))$

$= xy$

② $x\bar{y} + \bar{x}y + xy + \bar{x}\cdot\bar{y}$

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



③ $\overline{AB\cdot(\overline{\bar{C}+D})\cdot(AB)}$

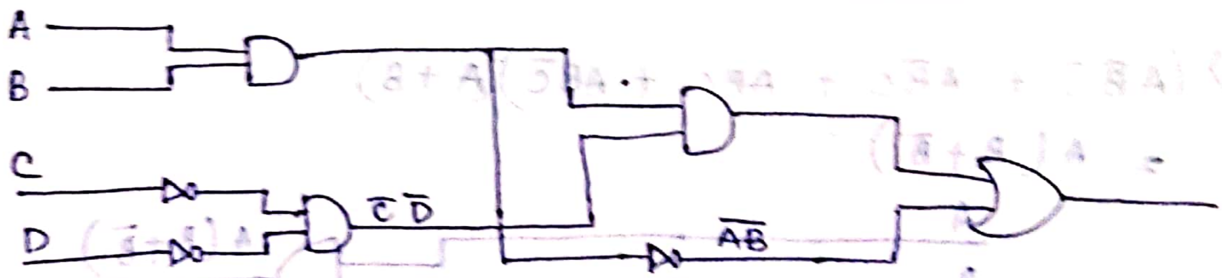
$= \overline{AB\cdot(\bar{\bar{C}}\cdot\bar{D})\cdot(AB)}$

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

$= (\overline{\overline{AB}} + \overline{\overline{\bar{C}\cdot\bar{D}}}) + \overline{AB}$

$= (\overline{\overline{AB}})\cdot(\overline{\overline{\bar{C}\cdot\bar{D}}}) + \overline{AB}$

$= ((AB)\cdot(\bar{C}\cdot\bar{D})) + \overline{AB}$



(14) 1) $x \cdot y \cdot y \cdot z \cdot z \cdot z$

$= x \cdot y \cdot z$

2) $x \cdot x \cdot y + x \cdot y \cdot y + y \cdot \bar{y} \; x \cdot x$

$= xy$

(15)

3) $x(y + z(\overline{x \cdot y + x \cdot z}))$

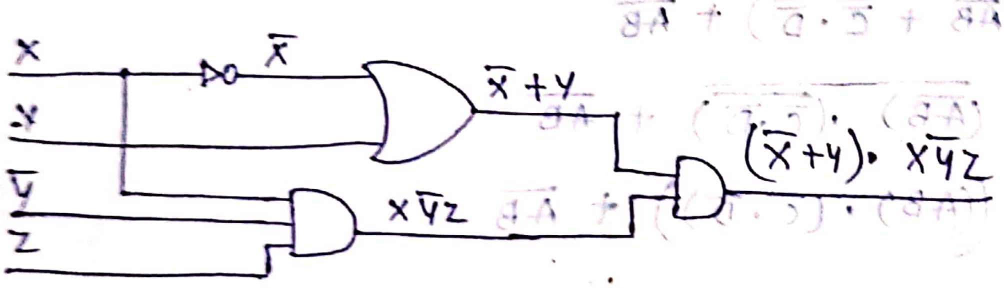
$= xy$

(16) 4) $x\bar{y} + \bar{x}y + xy + \bar{x} \cdot \bar{y}$

$= x + \bar{x}$

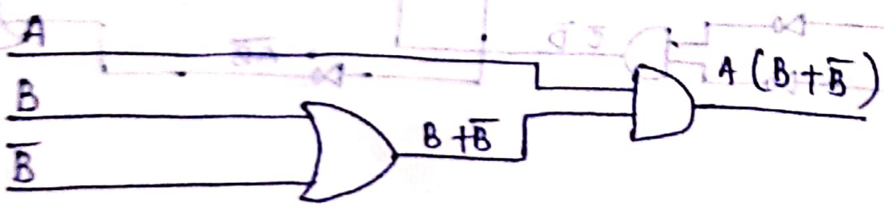
5) $(\bar{x} + y) \cdot x\bar{y}z$

(17) $= 0$



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

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

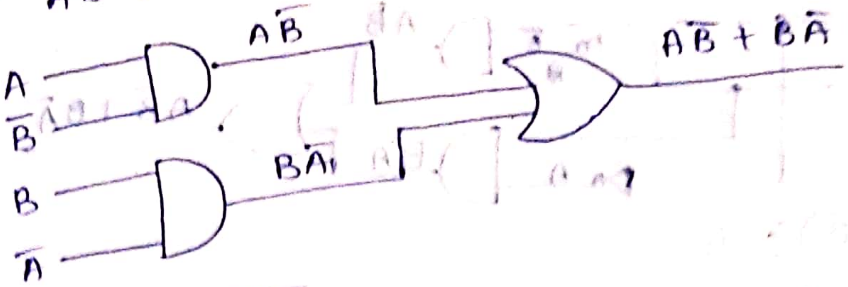


$(A+B)(\bar{A}+B)$

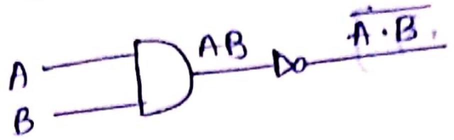
$= (A+B)(\bar{A}+\bar{B})$

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

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



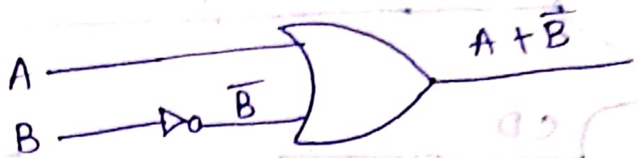
* $\bar{A}+\bar{B} = \overline{A\cdot B}$



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

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

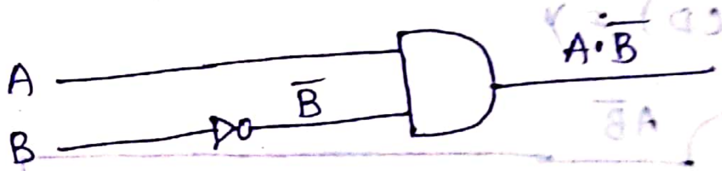
$= A + \bar{B}$



** $\overline{\bar{A}+B}$ $\qquad [\because \overline{X+Y} = \bar{X}\cdot\bar{Y}]$

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

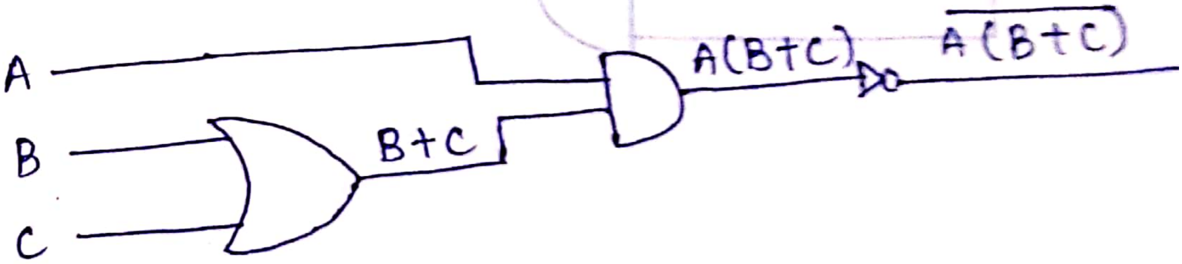
$= A\cdot\bar{B}$



* $\overline{\bar{A}+\bar{B}\cdot\bar{C}}$ $\qquad [\because \overline{X+Y} = \bar{X}\cdot\bar{Y}]$

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

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



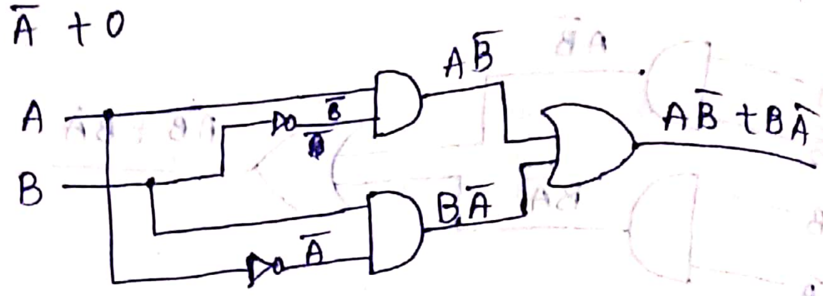
* $(A+B)(\overline{A\cdot B})$

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

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

$= 0 + A\bar{B} + B\bar{A} + 0$

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

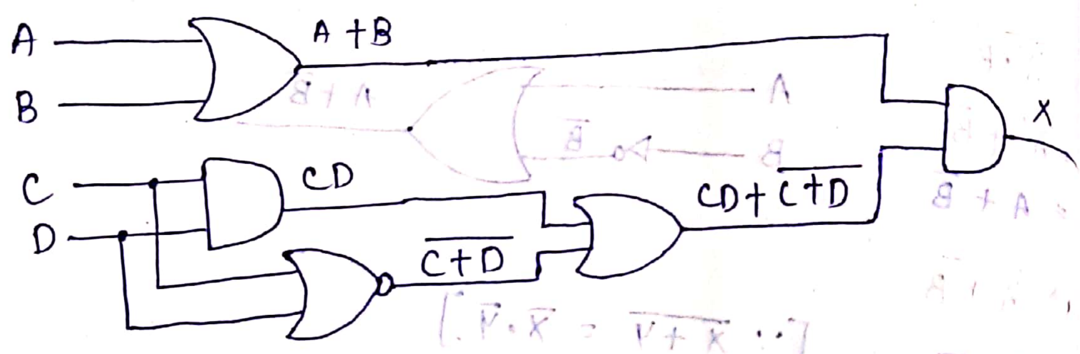


* $(A+B)C\cdot D + (A+B)\bar{C}\cdot\bar{D}$

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

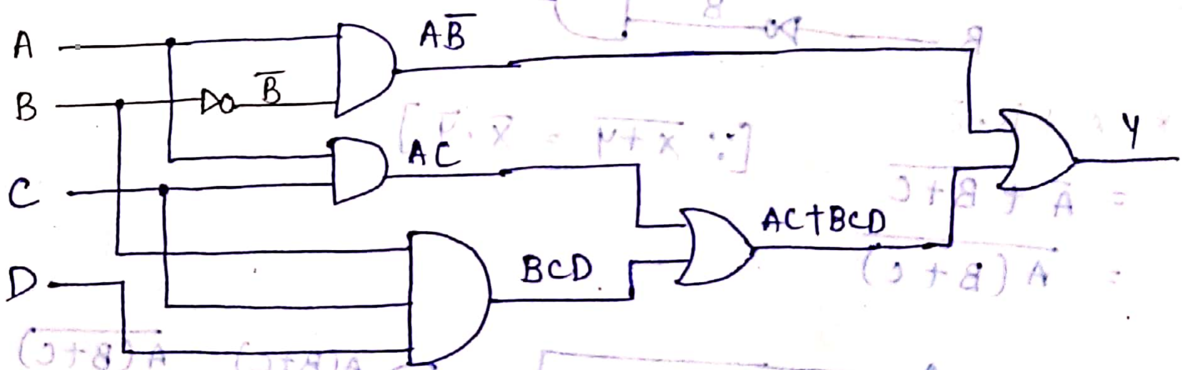
$= (A+B)(CD + \overline{C+D})$ $[\because \overline{X+Y} = \bar{X}\cdot\bar{Y}]$

$= X$



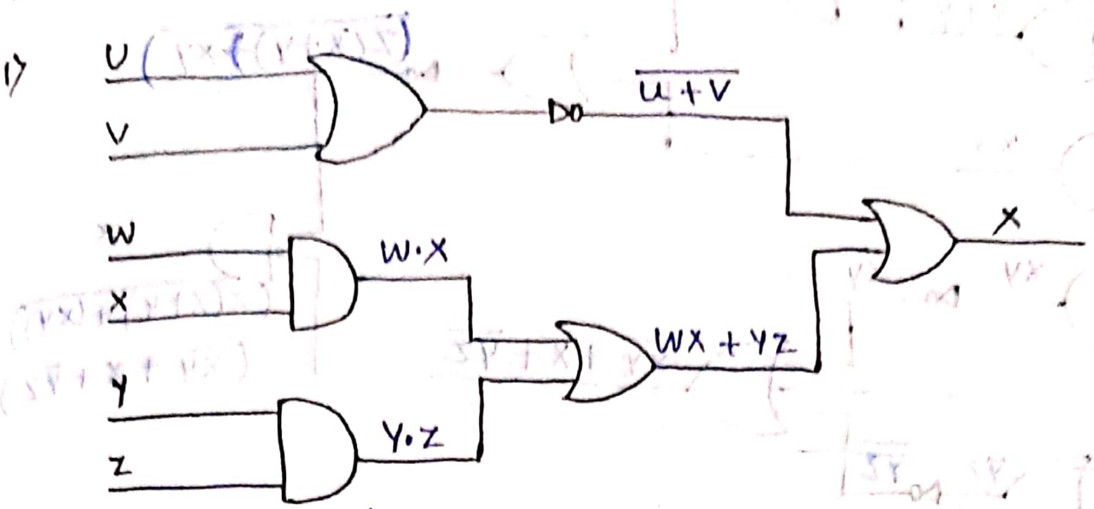
* $(A\cdot\bar{B}) + C(A+BD) = X$

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



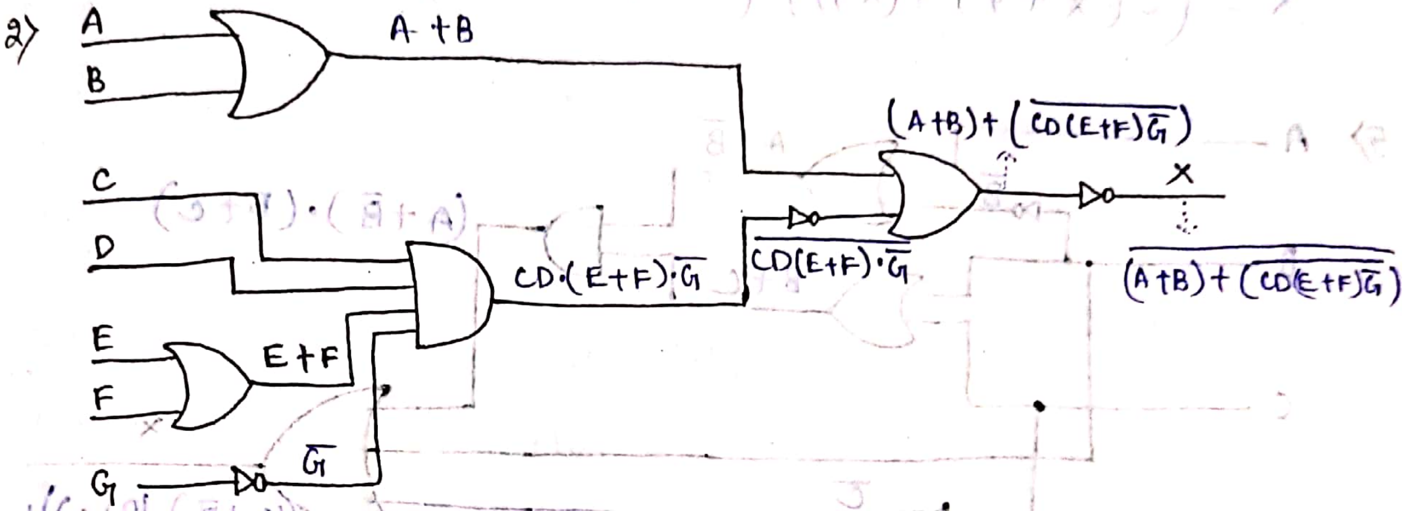
Write an expression for the given logic diagram :-

1)



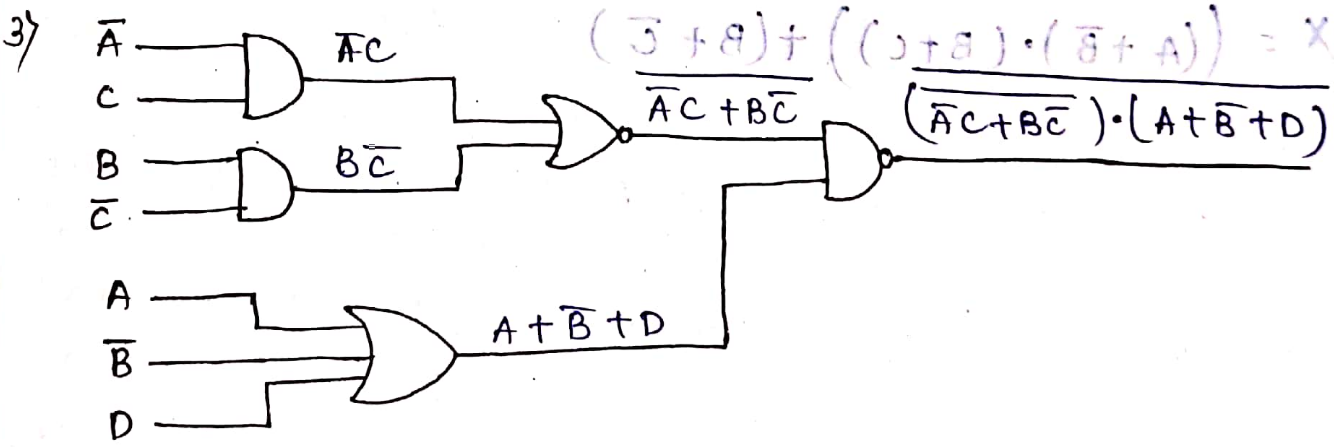
$X = (\overline{U+V}) + ((W \cdot X) + (Y \cdot Z))$

2)



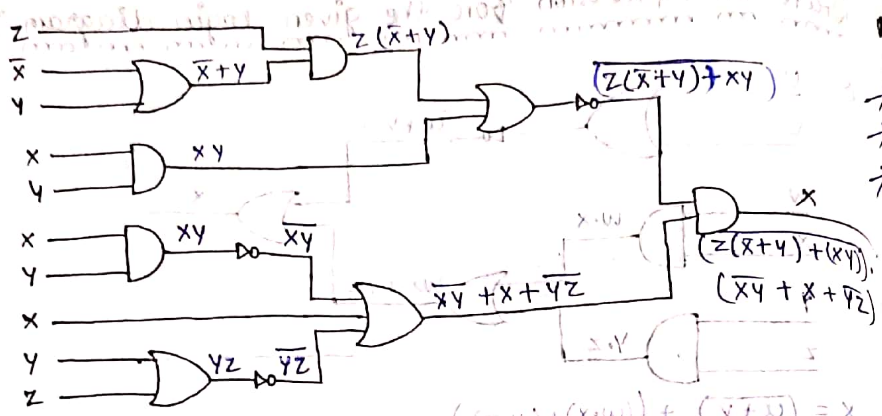
$X = \overline{(A+B) + (\overline{CD(E+F)\overline{G}})}$

3)



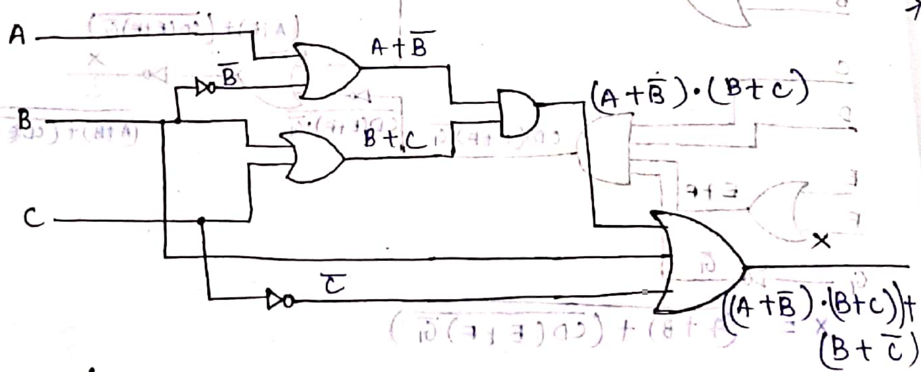
$X = \overline{(\overline{\overline{A}C + B\overline{C}}) \cdot (A + \overline{B} + D)}$

4)



$$X = \overline{(Z(\bar{X}+Y)+(XY))}\cdot(\overline{XY}+X+\overline{YZ})$$

5)



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

Minimisation Using K-map :-

→ The K-map is also known as Karnaugh
→ This map is used to simplify boolean
→ K-map are also used to simplify real requirements to be solved so that the implemented by using minimum number
→ In many digital circuit practical proble find expression with minimum variables
→ We can minimise the boolean expressio very easily using K-map without usi algebric theorem.
→ K-map can make two form i.e. Sum of product of sum (POS) according to problem.
→ K-map is represented in table but information than truth table.
→ Here each grid scales of K-map 0's and 1's then solve it by ma

Steps to solve an expression usi

i) Select K-map according to num
ii) Identify minterms as given in t to input variables.
iii) For SOP put 1's in the blocks to min-terms.
iv) For POS put 0's in the block to maxterm.
v) Create rectangular groups con Power of 2 like 2, 4, 6, 8, 16 cover as many elements as you

Minimisation Using K-map :-

- The K-map is also known as Karnaugh's maps.
- This map is used to simplify boolean functions.
- K-map are also used to simplify real world logical requirements to be solved so that they can be implemented by using minimum number of logic gates.
- In many digital circuit practical problems we need to find expression with minimum variables.
- We can minimise the boolean expression of 3, 4 variables very easily using K-map without using any boolean algebric theorem.
- K-map can make two form i.e. Sum of Product (SOP) and product of sum (POS) according to the need of the problem.
- K-map is represented in table but contains more information than truth table.
- Here each grid scales of K-map are filled with 0's and 1's then solve it by making groups.

Steps to solve an expression using K-map :-

i) Select K-Map according to number of variables.
i) Identify minterms as given in the problem according to input variables.
) For SOP put 1's in the blocks of K-Map respective to min-terms.
) For POS put 0's in the blocks of K-map respective to max term.
) Create rectangular groups containing total digits in Power of 2 like 2, 4, 6, 8, 16, 32 (except 1). Try to cover as many elements as you can in one group.

(ii) From the groups created the product terms need to be sum of for SOP term.

(iii) The summed up terms need to be multiplied for POS form.

Minterms and Maxterms :-

→ A binary variable may appear in its normal form as x or its complement form $\bar{x}$.

→ For example x and y are two variables and its possible options with AND operation are $\bar{x} \cdot \bar{y}$, $\bar{x} \cdot y$, $x \cdot \bar{y}$, $x \cdot y$.

→ Similarly we can have n variables, for which there are 2^n possible options.

Variables			Minterms		Maxterms	
x	y	z	Term	Designation	Term	Designation
0	0	0	$\bar{x}\bar{y}\bar{z}$	m_0	$X+Y+Z$	M_0
0	0	1	$\bar{x}\bar{y}z$	m_1	$X+Y+\bar{Z}$	M_1
0	1	0	$\bar{x}y\bar{z}$	m_2	$X+\bar{Y}+Z$	M_2
0	1	1	$\bar{x}yz$	m_3	$X+\bar{Y}+\bar{Z}$	M_3
1	0	0	$x\bar{y}\bar{z}$	m_4	$\bar{X}+Y+Z$	M_4
1	0	1	$x\bar{y}z$	m_5	$\bar{X}+Y+\bar{Z}$	M_5
1	1	0	$xy\bar{z}$	m_6	$\bar{X}+\bar{Y}+Z$	M_6
1	1	1	xyz	m_7	$\bar{X}+\bar{Y}+\bar{Z}$	M_7

Sum of Products (SOP):-

→ A SOP expression is a single product term (minterm) or several product terms (minterm) logically added togather.

e.g→ $x \cdot y$, $x \cdot \bar{y} \cdot z$, $x \cdot y + \bar{x} \cdot y$, $x\bar{y}z + \bar{x}y\bar{z}$, $x\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z}$.

Steps to be followed for expressing a function in its SOP form :-

Step 1 :- Construct a truth table for boolean function.

Step 2 :- Form a minterm for each combination of the variables that produces function.

Step 3 :- The desired expression is the min terms obtained in step-2

2 Variable K-Maps :-

XY

x \ y	0	1
0	m_0	m_1
1	m_2	m_3

or

00	01	11	10
m_0	m_1	m_3	m_2
(0)	(1)	(3)	(2)

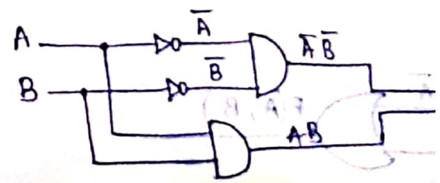
The possible combination of 2 min terms
$\{(m_0 m_1), (m_2 m_3), (m_0 m_2), (m_1 m_3)\}$

e.g→
(1) $F(A,B) = \Sigma(0,3)$

A \ B	0	1
0	1	
1		1

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

Logic gate :-



... SOP form.

... terms need to be multiplied for

... xterms :-

... may appear in its normal form ... ment form $\bar{x}$.

... are two variables and its ... AND operation are $\bar{x}\cdot\bar{y}$, $\bar{x}\cdot y$,

... are n variables, for which there ... ions.

...	Maxterms	
...ation	Term	Designation
...	$x+y+z$	M_0
...	$x+y+\bar{z}$	M_1
...	$x+\bar{y}+z$	M_2
...	$x+\bar{y}+\bar{z}$	M_3
...	$\bar{x}+y+z$	M_4
...	$\bar{x}+y+\bar{z}$	M_5
...	$\bar{x}+\bar{y}+z$	M_6
...	$\bar{x}+\bar{y}+\bar{z}$	M_7

... (SOP) :-

... a single product term (minterm) ... terms (minterm) logically

... $y + \bar{x}\cdot y$, $xyz + \bar{x}y\bar{z}$,

... $y\bar{z}$

Steps to be followed for expressing a boolean function in its SOP form :-

Step 1 :- Construct a truth table for the given boolean function.

Step 2 :- Form a minterm for each combination of the variables that produces a 1 in the function.

Step 3 :- The desired expression is the sum (OR) of all min terms obtained in step-2.

2 Variable K-Maps :-

XY

X \ Y	0	1
0	m_0	m_1
1	m_2	m_3

or

XY	00	01	11	10
	m_0	m_1	m_3	m_2
	(0)	(1)	(4)	(3)

The possible combination of 2 min terms are :-

$\{(m_0 m_1), (m_2 m_3), (m_0 \cdot m_2), (m_1 m_3)\}$

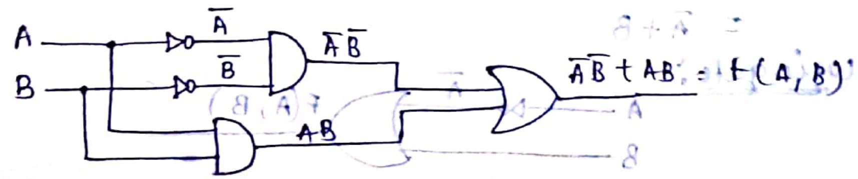
e.g →

(1) $F(A, B) = \Sigma(0, 3)$

A \ B	0	1
0	1	
1		1

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

Logic gate :-



(2) $F(A,B) = \Sigma(0,1,3) = SOP(\ldots$

A \ B	0	1
0	1	
1	1	1

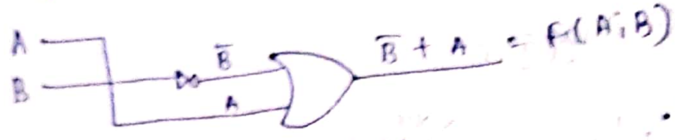
$f(A,B) = \bar{A}\bar{B} + A\bar{B} + A\bar{B} + AB$ $\quad (\because \bar{A} + A = 1)$

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

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

$= \bar{B} + A$

Logic gate:

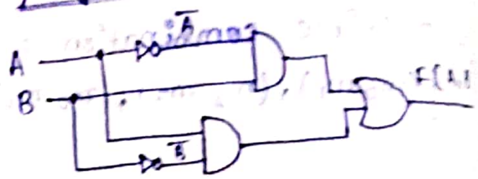


(3) $f(A,B) = \Sigma(1,2)$

A \ B	0	1
0		1
1	1	

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

Logic gate:



(4) $F(A,B) = \Sigma(0,1,3)$

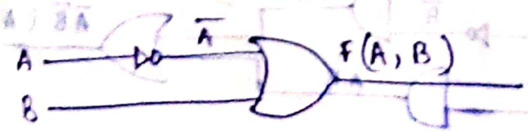
A \ B	0	1
0	1	1
1		1

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

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

$= \bar{A} + B$

Logic gate:



3-Variable K-map:

The number of cells in 3 variab[le]

$2^3 = 8$ (i.e. 8)

A \ BC	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

The possible grouping of adjacent m[interms]

$\{(m_0 m_1, m_3 m_2), (m_4 m_5 m_7 m_6)$

$(m_1 m_3 m_5 m_7), (m_3 m_2 m_7 m_6$

These are Quads.

e.g. (1) $f(A,B,C) = \Sigma(0,4)$

A \ BC	00	01	11	10
0	1			
1	1			

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

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

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

(2) $F(A,B,C) = \Sigma(0,1,2,3)$

A \ BC	00	01	11	10
0	1	1	1	1
1				

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

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

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

Logic gate:

3 - Variable K - map :

The number of cells in 3 variable K - map will be $2^3 = 8$ (i.e. - 8)

$A \backslash BC$	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

The possible grouping of adjucent minterms will be -

$\{(m_0 m_1 m_3 m_2), (m_4 m_5 m_7 m_6), (m_0 m_1 m_4 m_5),$
$(m_1 m_3 m_5 m_7), (m_3 m_2 m_7 m_6), (m_0 m_2 m_4 m_6)\}$

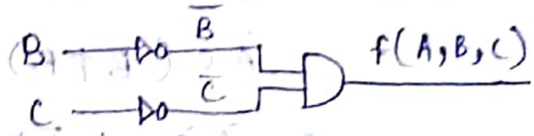
These are Quads.

e.g - (1) $f(A,B,C) = \Sigma(0,4)$

$A \backslash BC$	00	01	11	10
0	1			
1	1			

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

Logic gate :



(2) $F(A,B,C) = \Sigma(0,1,2,3)$

$A \backslash BC$	00	01	11	10
0	1	1	1	1
1				

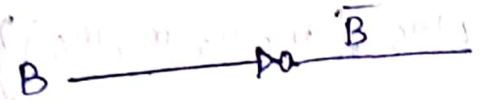
$f(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC + \bar{A}B\bar{C}$
$= \bar{A}\bar{B}(\bar{C} + C) + \bar{A}B(C + \bar{C})$
$= \bar{A}\bar{B} + \bar{A}B = \bar{A}(\bar{B} + B) = \bar{A}$

Logic gate : A —▷o— $\bar{A}$ = f(A,B,C)

(3) $f(A,B,C) = \Sigma(0,1,4,5)$

A \ BC	00	01	11	10
0	1	1		
1	1	1		

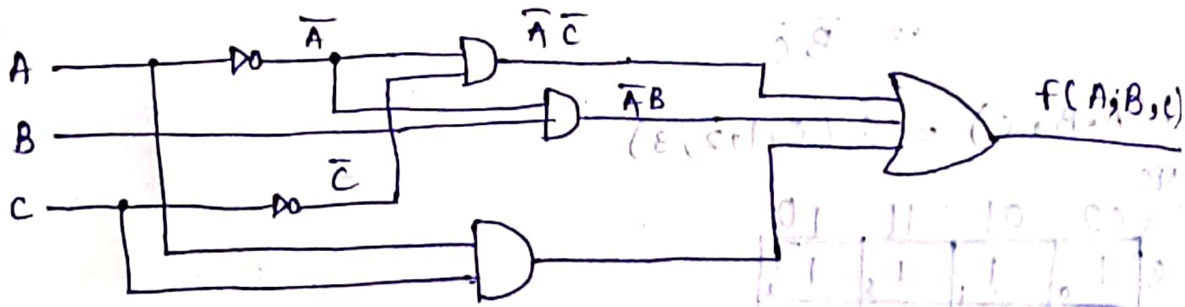
$$f(A,B,C) = \bar A\bar B\bar C + \bar A\bar B C + A\bar B\bar C + A\bar B C$$
$$= \bar A\bar B(\bar C + C) + A\bar B(\bar C + C)$$
$$= \bar A\bar B + A\bar B$$
$$= \bar B(\bar A + A)$$
$$= \bar B$$



(4) $f(A,B,C) = \Sigma(0,2,3,5,7)$

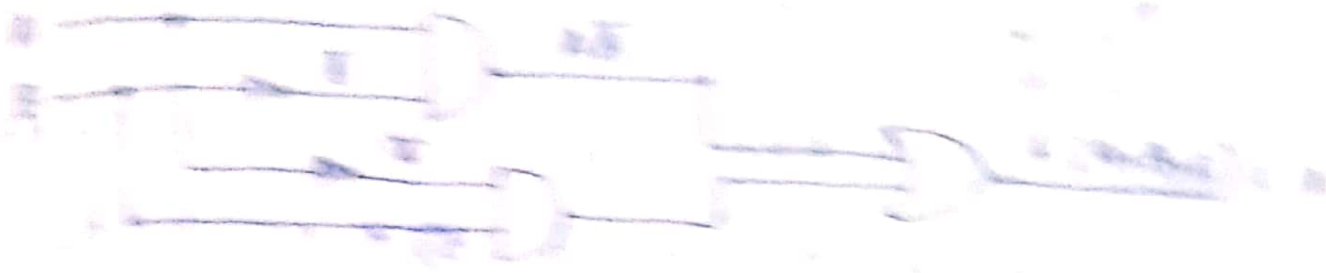
A \ BC	00	01	11	10
0	1		1	1
1		1	1	

$$f(A,B,C) = \bar A\bar B\bar C + \bar A B\bar C + \bar A BC + \bar A B\bar C + A\bar B C + ABC$$
$$= \bar A\bar C(\bar B + B) + \bar A B(C + \bar C) + AC(\bar B + B)$$
$$= \bar A\bar C + \bar A B + AC$$



(5) $f(A,B,C) = \Sigma(3,2,4,5)$

A \ BC	00	01	11	10
0			1	1
1	1	1		



1) $f(A,B,C) = \Sigma(0,2,4,5,7)$

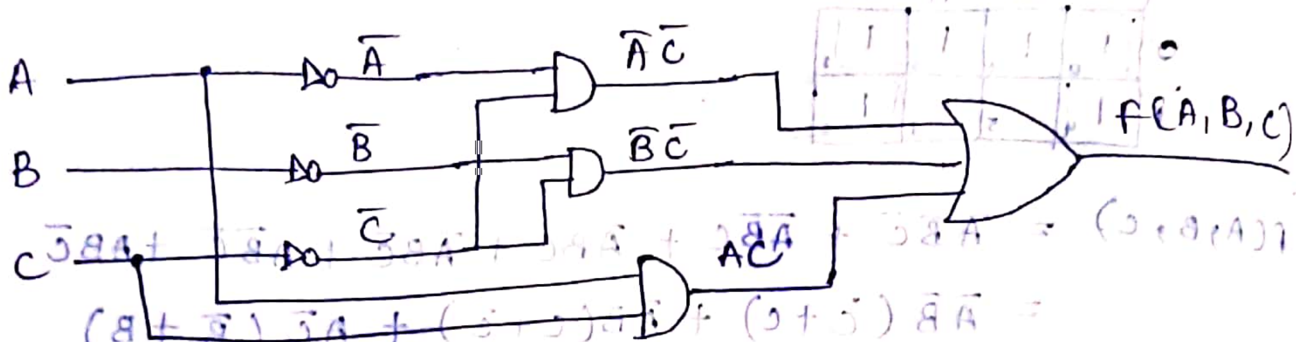
A \ BC	00	01	11	10
0	1			1
1	1	1	1	

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

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

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

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



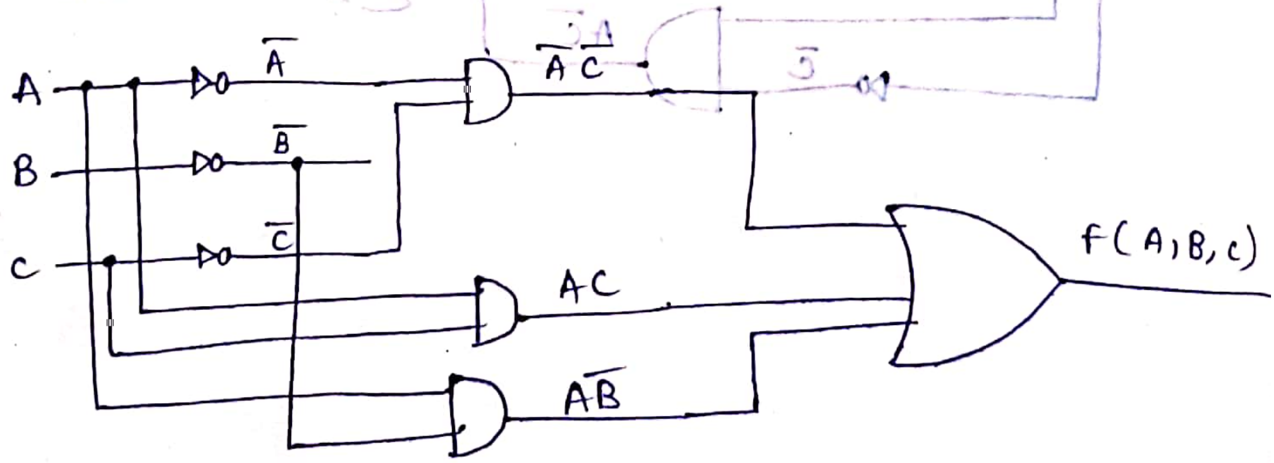
A \ BC	00	01	11	10
0	1			1
1	1	1	1	

or// $f(A,B,C)$

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

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

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



$F(A, B, C, D) = (0, 2, 4, 5, 7, 10, 12, 13, 15)$

AB \ CD	00	01	11	10
00	1 (0)	(1)	(3)	1 (2)
01	1 (4)	1 (5)	1 (7)	(6)
11	1 (12)	1 (13)	1 (15)	(14)
10	(8)	(9)	(11)	1 (10)

$\bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D + AB\bar{C}\bar{D} + AB\bar{C}D \rightarrow$ Quad-1

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

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

$\bar{A}B\bar{C}D + \bar{A}BCD + AB\bar{C}D + ABCD \rightarrow$ Quad-2

$= \bar{A}BD(\bar{C}+C) + ABD(\bar{C}+C)$

$= \bar{A}BD + ABD$

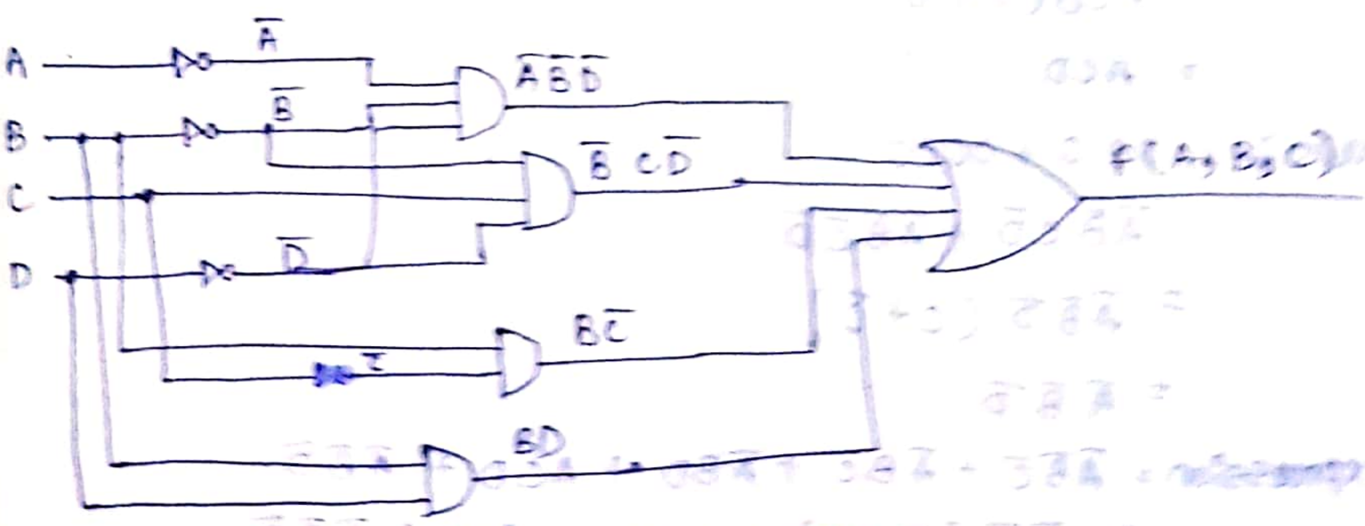
$\bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} = \bar{A}\bar{B}\bar{D}(\bar{C}+C) = \bar{A}\bar{B}\bar{D}$

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

$\bar{A}B\bar{C} + AB\bar{C} + \bar{A}BD + ABD + \bar{A}\bar{B}\bar{D} + \bar{B}C\bar{D}$

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

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



~~Max terms or Sum Terms:~~

→ ~~The term which are formed by 2 or more binary~~

3) $f(A,B,C,D) = (0,1,2,4,5,7,11,15)$

Step 1

AB \ CD	00	01	11	10
00	1	1		1
01	1	1	1	
11			1	
10			1	

Step 2

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

Step 3

Quad-1 = 0000 + 0001 + 0100 + 0101

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

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

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

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

~~Quad~~ Pair-1 = 0101 + 0111

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

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

$= \bar{A}BD$

Pair-2 = 1111 + 1011

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

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

$= ACD$

Pair-3 = 0010 + 0000

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

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

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

expression $= \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BD + ACD + \bar{A}\bar{B}\bar{D}$

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

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

by 2 or more binary

$(\ldots, 11, 15)$

Step-2

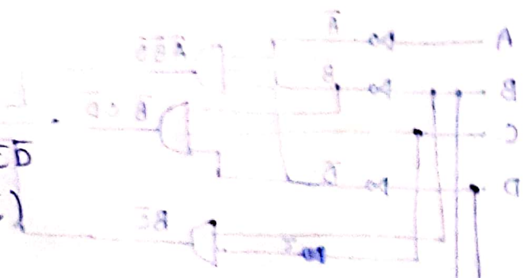
Decimal	A	B	C	D	Minterms
0	0	0	0	0	$\bar{A}\bar{B}\bar{C}\bar{D}$
1	0	0	0	1	$\bar{A}\bar{B}\bar{C}D$
2	0	0	1	0	$\bar{A}\bar{B}C\bar{D}$
3	0	0	1	1	
4	0	1	0	0	$\bar{A}B\bar{C}\bar{D}$
5	0	1	0	1	$\bar{A}B\bar{C}D$
6	0	1	1	0	
7	0	1	1	1	$\bar{A}BCD$
8	1	0	0	0	
9	1	0	0	1	
10	1	0	1	0	
11	1	0	1	1	$A\bar{B}CD$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

0101

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

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

$\bar{A}\bar{C}$



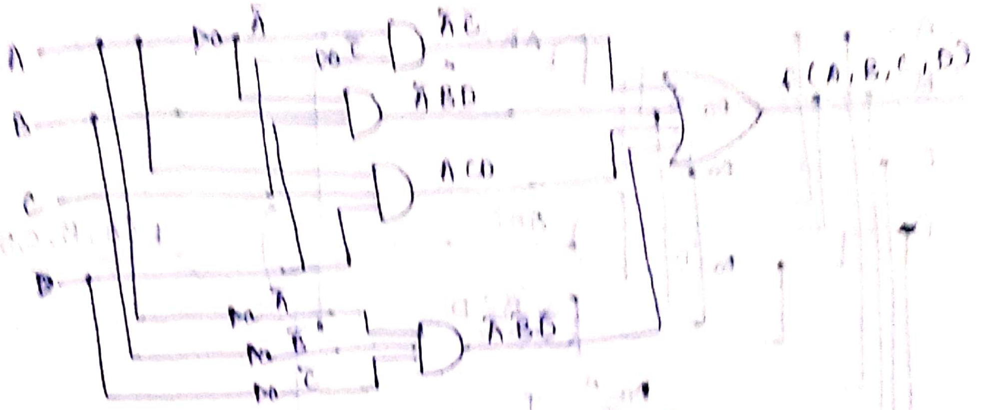
$\bar{C} + \bar{A}BD + ACD + \bar{A}\bar{B}\bar{D}$

$B) + \bar{A}BD + ACD + \bar{A}\bar{B}\bar{D}$

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

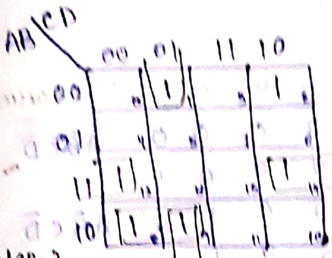
Step-4 Logic gate

$\bar{A}\bar{C} + \bar{A}BD + ACD + \bar{A}\bar{B}\bar{D}$



4) $f(A,B,C,D) = \Sigma(1,2,8,9,12,14)$

Step-1 K-map



Step-3

Pair-1 = 1000 + 1001

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

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

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

Pair-2 = 1100 + 1110

$= AB\bar{C}\bar{D} + ABC\bar{D}$

$= AB\bar{D}(\bar{C}+C)$

$= AB\bar{D}$

Pair-3 = 0001 + 1001

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

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

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

Single = 0010 = $\bar{A}\bar{B}C\bar{D}$

Step-2

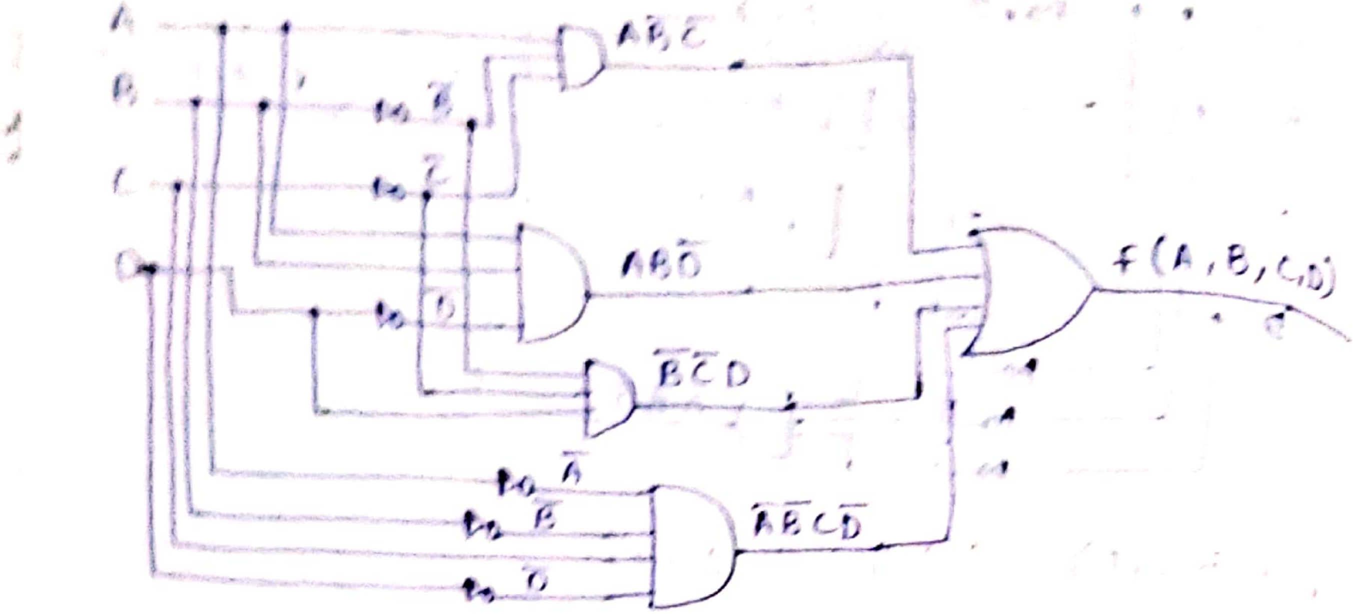
Decimal	A	B	C	D	Minterms
0	0	0	0	0	
1	0	0	0	1	$\bar{A}\bar{B}\bar{C}D$
2	0	0	1	0	$\bar{A}\bar{B}C\bar{D}$
3	0	0	1	1	
4	0	1	0	0	
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	
8	1	0	0	0	$A\bar{B}\bar{C}\bar{D}$
9	1	0	0	1	$A\bar{B}\bar{C}D$
10	1	0	1	0	
11	1	0	1	1	
12	1	1	0	0	$AB\bar{C}\bar{D}$
13	1	1	0	1	
14	1	1	1	0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

Scanned with OKEN Scanner

Answer

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

Logic gate



5) $f(A,B,C,D) = \Sigma(0,2,5,6,8,10,11,15)$

Step 1 K-map

AB \ CD	00	01	11	10
00	1			1
01		1		1
11			1	
10	1		1	1

Step 2

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

Step 3

Pair 1 = 0000 + 0010

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

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

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

Pair 2 = 0010 + 0110

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

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

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

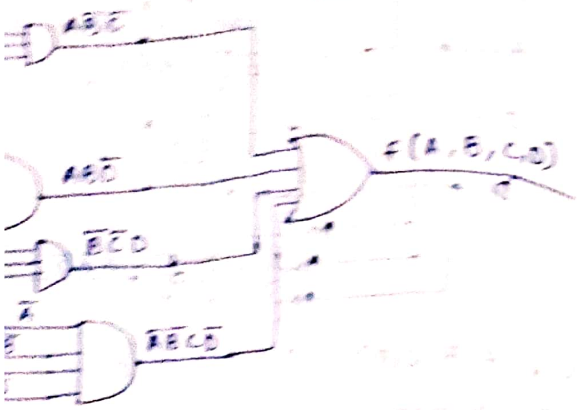
Pair 3 = 1000 + 1010

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

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

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

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



2, 5, 6, 8, 10, 11, 15)

step-2

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

pair $\Rightarrow$ 1111 + 1011

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

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

$= ACD$

single $=$ 0101

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

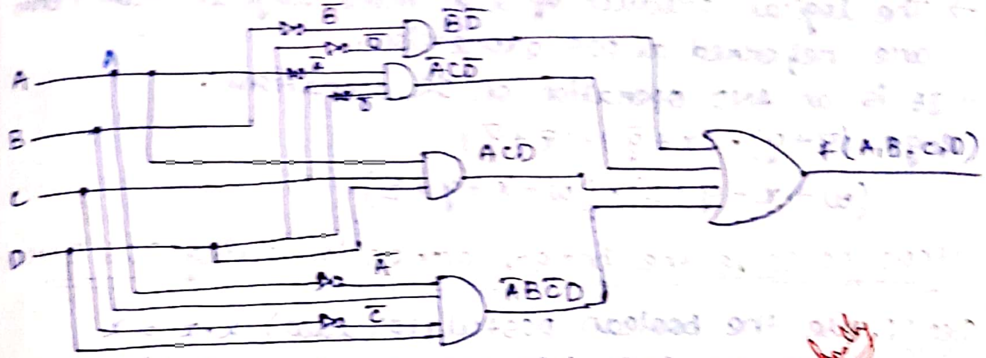
expression:

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

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

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

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



Max Terms or Sum Terms :-

→ The term which are formed by 2 or more binary variables combined with OR operation with each variable being prime or unprime is called as maxterm or sum term or standing term.

e.g→ $a+b$, $\bar{x}+y$, $A+\bar{B}+C$

→ Each max term can be represented with the binary number 0. The numbers can be 0, 1, 2, ------, n-1. Max terms can be $M_0, M_1, M_2 ---- M_{n-1}$

Product of Sum :-

→ The logical product of 2 or more logical sum term are referred as POS expression.

→ It is an AND operation on OR operations.

e.g→ $(\bar{x}+y)\cdot(x+\bar{y})\cdot(\bar{x}+\bar{y})$

$(w+x+y+z)\cdot(w+\bar{x}+y+\bar{z})$

steps to solve the boolean expression using POS form

Step 1: use the boolean postulate (law) $x\cdot x = x$
Hence we can use any one term 2 times

e.g→ $(p+q+r)$ can be written as $(p+q+r)\cdot(p+q+r)$

Suppose -

$(p+q+r)(p+q+\bar{r})\cdot(p+\bar{q}+r)\cdot(\bar{p}+q+r)$

$=(p+q+r)(p+q+r)(p+q+r)(p+q+\bar{r})(p+\bar{q}+r)(\bar{p}+q+r)$

1st 2nd 3rd 4th 5th 6th

Step 2: Use distributive law $x+(y\cdot z) = (x+y)\cdot(x+z)$
for 1st and 4th paranthesis, 2nd and 5th paranthesis, 3rd and 6th paranthesis.

$f = (q+r+p\bar{p})(p+q\bar{q}+r)(p+q+r\bar{r})$

Step 3: use the boolean postulate (law) $x\cdot\bar{x} = 0$
for simplifying the terms present in each paranthesis.

Sum Terms :-

are formed by 2 or more binary
ned with OR operation with each
prime or unprime is called as
m term or standing term.

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

can be represented with the binary
numbers can be $0, 1, 2, \dots, n-1$;
be $M_0, M_1, M_2, \dots, M_{n-1}$

:-

duct of 2 or more logical sum term
s POS expression.
eration on OR operations.

$\bar{y})\cdot(\bar{x}+\bar{y})$

$)\cdot(w+\bar{x}+y+\bar{z})$

boolean expression using POS form

ean postulate (law) $x\cdot x = x$
use any one term 2 times
be written as $(p+q+r)\cdot(p+q+r)$

$\cdot(p+\bar{q}+r)\cdot(\bar{p}+q+\bar{r})$

$q+r)(p+q+\bar{r})(p+\bar{q}+r)(\bar{p}+q+r)$

red 4th 5th 6th

ive law $x+(y\cdot z) = (x+y)\cdot(x+z)$
th parenthesis, 2nd and 5th
red and 6th parenthesis.

$+q\bar{q}+r)(p+q+r\bar{r})$

an postulate (law) $x\cdot\bar{x} = 0$
g the terms present in each

$f = (q+r+0)(p+0+r)(p+q+0)$

step 4: use the boolean postulate $x+0 = x$ for simplifying the terms.

$f = (q+r)(p+r)(p+q)$

$= (p+q)(q+r)(p+r)$

POS symbol = Π

POS of Max term ΠM

Solve :-

1) $f(A,B,C) = \Pi M(1,4,5)$

step-1: The selected sum terms are represented using -

A \ BC	00	01	11	10
0		0		
1	0	0		

Step 2

Decimal	A	B	C	Maxterms
0	0	0	0	
1	0	0	1	$A+B+\bar{C}$
2	0	1	0	
3	0	1	1	
4	1	0	0	$\bar{A}+B+C$
5	1	0	1	$\bar{A}+B+\bar{C}$
6	1	1	0	
7	1	1	1	

step 3

Expressions :-

pair 1 = $(1+0+0)\cdot(1+0+1)$

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

pair 2 = $(0+0+1)\cdot(1+0+1)$

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

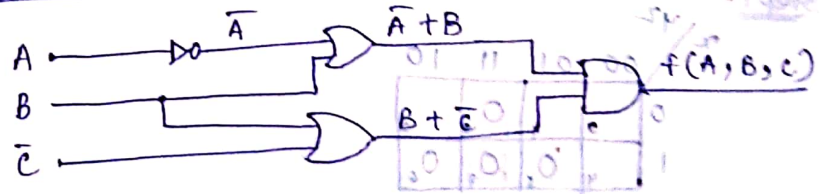
$f = (\bar{A}+B+C)(\bar{A}+B+\bar{C})(A+B+\bar{C})(\bar{A}+B+\bar{C})$

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

$= (\bar{A}+B+0)(0+B+\bar{C})$

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

Step-4 Logic gate :-



2) $f(A,B,C) = \Pi(0,1,2,4)$

Step-1

A \ BC	00	01	11	10
0	0	0		0
1	0			

Step-2

Dec	A	B	C	Maxterm
0	0	0	0	$A+B+C$
1	0	0	1	$A+B+\bar{C}$
2	0	1	0	$A+\bar{B}+C$
3	0	1	1	
4	1	0	0	$\bar{A}+B+C$
5	1	0	1	
6	1	1	0	
7	1	1	1	

Step-3

pair-1 $= (0+0+0)(0+0+1)$

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

pair 2 $= (0+0+0)(1+0+0)$

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

pair 3 $= (0+0+0)(0+1+0)$

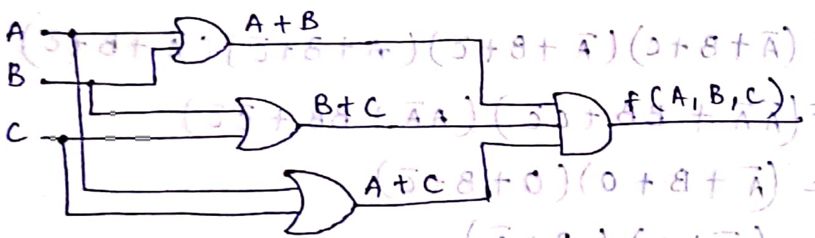
$= (A+B+C)(A+\bar{B}+C)$

Expression:

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

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

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



3) $f(x,y,z) = \Pi(3,5,6,7)$

Step 1 -

x \ yz	00	01	11	10
0			0	
1		0	0	0

Step-2

Dec	x	y
0	0	0
1	0	0
2	0	1
3	0	1
4	1	0
5	1	0
6	1	1
7	1	1

Step-3

pair 1 $= (0+1+1)(1+1+1)$

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

pair 2 $= (1+0+1)(1+1+1)$

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

pair 3 $= (1+1+1)(1+1+0)$

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

expression:

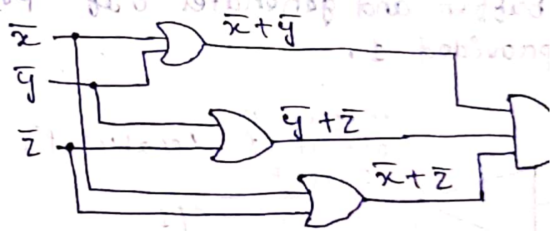
$f(x,y,z) = (x+\bar{y}+\bar{z})(\bar{x}+\bar{y}+\bar{z})(\bar{x}+y$ ... $(\bar{x}+\bar{y}+$ ...

$= (x\bar{x}+\bar{y}\bar{y}+\bar{z}\bar{z})(\bar{x}\bar{x}+y\bar{y}+\bar{z}$ ...

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

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

Step-4



Step 3

pair 1 = (0 + 1 + 1)(1 + 1 + 1)

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

pair 2 = (1 + 0 + 1)(1 + 1 + 1)

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

pair 3 = (1 + 1 + 1)(1 + 1 + 0)

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

Step 2

row	x	y	z	maxterm
0	0	0	0	
1	0	0	1	
2	0	1	0	
3	0	1	1	$x + \bar{y} + \bar{z}$
4	1	0	0	
5	1	0	1	$\bar{x} + y + \bar{z}$
6	1	1	0	
7	1	1	1	

expression:

$f(x, y, z) = (x + y + z)(\bar{x} + y + z)(x + \bar{y} + z)(x + y + \bar{z})$

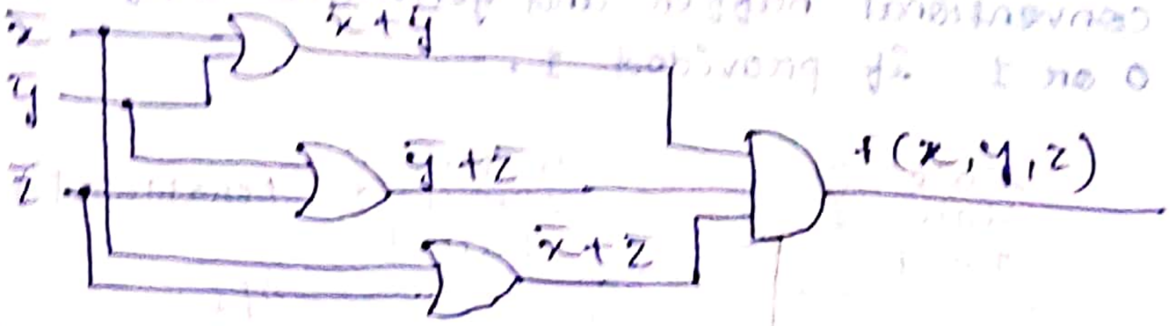
$(x + y + z)(x + y + z)$

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

$= (y + z)(x + z)(x + y)$

$= (x + y)(y + z)(x + z)$

Step 4



Tri State Buffer :-

→ usually any logic circuit has two states for input and output of binary information either 0 or 1.

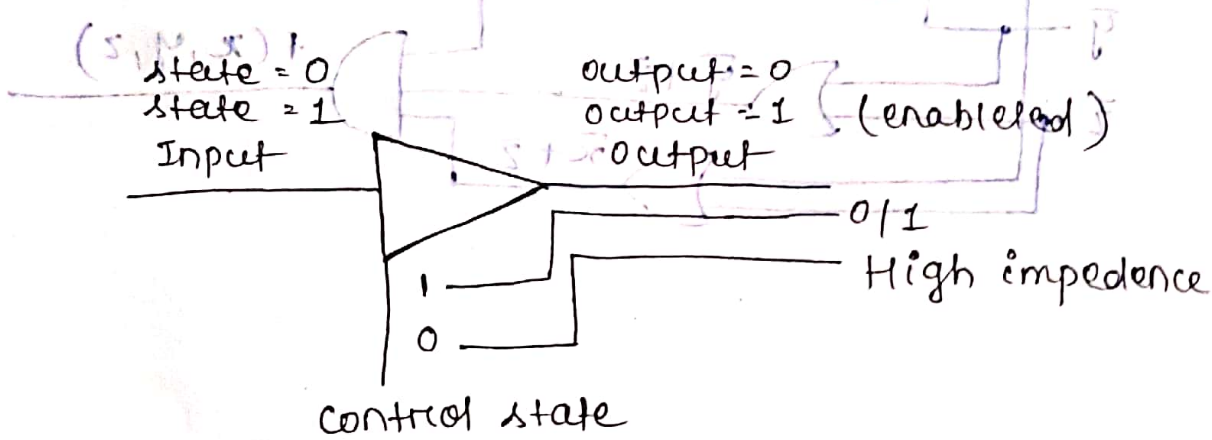
→ The tri state buffer includes 3 states.

→ It has 3 pins.

→ <u>Input</u> :- It accepts 0 which means disable or 1 which means enable.

→ <u>Output</u> :- The 3rd state is the control input if the three state control input is 0 then the output of it will be disabled and the gate will be in a high impedence state.

If the control input is 1 then the output will be enabled and the gate will behave as a conventional buffer and generates 0 if provided 0 or 1 if provided 1.



Description :-

→ As in a conventional gate 0 and 1 are two states.

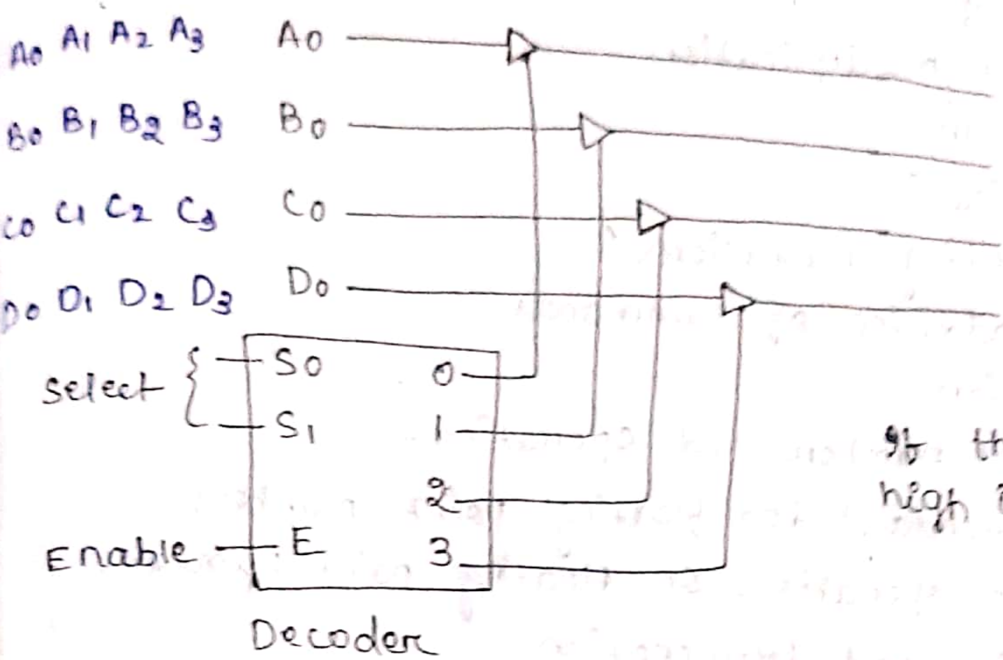
→ The 3rd state is a high impedence state.

→ The 3rd state behaves like an open circuit (open circuit means that the output is disconnected and doesn't have logic significance)

→ It may perform any type of conventional logic operations like AND, OR, NOT, NAND etc.

→ is differed than a normal buffer as it contains both normal input and control input. Here the control state is determined by the control input.

Implementation of Tristate Buffer :



If the decoder disables high impedence occurs.

Disable - High impedence occurs.

(Bus Line For Tristate Buffer)

→ To form a single bus line all the outputs of the 4 buffers are connected togather.

→ The control input will now decide which of the 4 normal inputs will communicate with the bus line.

→ The decoder is used to ensure that only one control input is active at a time.

→ From the above diagram the use of tristate buffer can be understood clearly.