Morris Mano
Anand Kumar

01. Number Systems and Codes

⇒ Review of number systems: Decimal systems, Binary, Octal & Hexa decimal Numbering systems,

⇒ Number base conversions, binary arithmatic, compliments

⇒ Signed binary numbers, Binary codes: weighted, Non-weighted error detecting and error correcting codes- hamming codes and other codes.

⇒ Boolean Algebra and logic Gates: Basic definition, Basic Logic operations, Basic thoerems and properties, Truth tables, Boolean functions, Representation of Boolean functions: Canonical and standard forms, Other logic operations Digital logic gates, Universal gates.

02. Gate Level Minimization

⇒ Algebraic simplification of Boolean expression, the map method, four variable K-map, five variable K-map Prime implicants, POS & SOP simplification,

⇒ Don't care conditions, Tabular Method - Simplification of Boolean function using tabulation Method, two level and multi-level NAND & NOR Implementation, Other two level Implementation circuits, Ex-OR function and properties.

03. Combinational Circuits

⇒ Combinational circuits, Analysis and Synthesis (Design)

⇒ procedure, Design using conventional logic gates:

⇒ Binary Adder - Subtractor - Decimal Adder - Binary multiplier

⇒ Magnitude comparator - Parity generators and checkers - Code Converters; Hazards; Decoder, Encoders, Multiplexers, Demultiplexer, Logic Design using MSI Components.

04. Synchronous Sequential Circuits

⇒ Sequential Circuits, Latches, flips-flops - Level and Edge triggering, Master - Slave flip-flops & Edge triggered flip-flops,

⇒ Characteristic table and equation, Excitation table, flip flop conversions

⇒ Analysis of synchronous sequential circuits, state reduction

⇒ and assignment, Design procedure of synchronous sequential circuits, Design examples; Serial binary adder - sequence detector.

05. Shift Registers and Counters

⇒ Registers, Shift Registers: SISO, SIPO, PISO, PIPO and universal shift registers, applications, Counters: Ripple Counters,

⇒ Synchronous counters, other counters and Application of counters

⇒ finite state Machines: Introduction - General Model of FSM, Classifications of FSM (Mealy & moore Models), Design of FSM, Design examples, Capabilities, and limitations of FSM.

06. Memory and Programmable Logic

⇒ Introduction, Random Access Memory, Memory Decoding

⇒ Read only memory, Programmable Logic Array

⇒ Programmable Array Logic, Sequential Programmable devices

⇒ Complex Programmable logic devices,

⇒ field Programmable Gate Arrays.

Introduction to Digital Electronics

The term 'digital' is derived from the way in which computers perform operations by counting

⇒ Digital electronics involves circuits and systems in which there are only two possible states which are typically represented by amplitude levels.

⇒ Digital is the way in which the computers performs the arithmatical operations.

⇒ Some digital signals are Keyboard signals

⇒ Sine wave is known as the fundamental or universal wave.

⇒ In digital systems two states are used to represent numbers, symbols, characters etc i.e 1 & 0.

* Difference between analog signal and Digital signal :-

(i) analog signal is defined as amplitude variables whose size is proportional to the quantity it represents.

(ii) Analog signal is continuous in nature.

(iii) Analog signal has infinite set of possible values.

(iv) examples:- Sound, Velocity, Wave, Pressure, Temperature

(01) A Digital signal is one which changes states between two discrete amplitude levels.

(02) Digital signals are discrete in nature.

(03) Here are 2 possible set of values 1 & 0.

(04) Examples are :- Telegraph signal, Keyboard signal

* Advantages of digital systems:-

01) Data accuracy and precision are greater in digital systems. & can be fabricated easily.

02) Noise is less in case of digital systems.

03) Digital systems are less time consuming.

04) Easier to understand, Easy to design

05) Complexity is less than analog systems.

information can be stored easily, Accurate & versatile

* Disadvantages of digital system:-

⇒ Most physical quantities are analog in nature. To take advantage of digital technique, the analog inputs are first to be converted to digital signal by the help of analog to digital converter (ADC). Then after processing converted back to analog form.

* Application of Digital Signals:-

01) Consumer electronics ex:- Television, Computers

02) Communication systems

03) Signal Processing, image & speech processing

Number systems and codes

Bit:- Binary digit '0' or 1.

1 Nibble - 4 bits;

1 byte - 8 bits;

1 word - 2 bytes; 16 bits

1 double word - 4 bytes; 32 bits.

1 quad word - 8 bytes; 64 bits.

1 Kilo byte - 2^{10} bytes; 1024 bytes.

1 Mega byte - $2^{10}.2^{10}$ bytes; 1024 Kilo bytes.

1 Giga byte - $2^{10}.2^{10}.2^{10}$ bytes; 1024 MB.

1 Tera byte - $2^{10}.2^{10}.2^{10}.2^{10}$ bytes; 1024 GB.

Number Symstems:- There are four number systems

01. Binary Number system:- $r=2$, '0' or '1'.
02. Octal number System:- $r=8$, 0, 1, 2, 3, 4, 5, 6, 7
03. Decimal number system:- $r=10$ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
04. Hexadecimal number system:- $r=16$ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 & A, B, C, D, E, F

Number system conversion:-

Binary to decimal:-

(i) $(110110) = 1\times2^5 + 1\times2^4 + 0\times2^3 + 1\times2^2 + 1\times2^1 + 0\times2^0$

$= 32 + 16 + 0 + 4 + 2 = (54)_{10}$

(ii) $(1101101110) = 1\times2^9 + 1\times2^8 + 0\times2^7 + 1\times2^6 + 1\times2^5 + 0\times2^4 + 1\times2^3 + 1\times2^2 + 1\times2^1 + 0\times2^0$

$= 512 + 256 + 64 + 32 + 8 + 4 + 2 = (878)_{10}$

Scanned by CamScanner

Decimal	Hexadecimal	Binary (16 8 4 2 1)	Octal
0	0	0 0 0 0 0	0
01	1	0 0 0 0 1	1
02	2	0 0 0 1 0	2
03	3	0 0 0 1 1	3
04	4	0 0 1 0 0	4
05	5	0 0 1 0 1	5
06	6	0 0 1 1 0	6
07	7	0 0 1 1 1	7
08	8	0 1 0 0 0	10
09	9	0 1 0 0 1	11
10	A	0 1 0 1 0	12
11	B	0 1 0 1 1	13
12	C	0 1 1 0 0	14
13	D	0 1 1 0 1	15
14	E	0 1 1 1 0	16
15	F	0 1 1 1 1	17
16	10	1 0 0 0 0	20
17	11	1 0 0 0 1	21
18	12	1 0 0 1 0	22
19	13	1 0 0 1 1	23
20	14	1 0 1 0 0	24
21	15	1 0 1 0 1	25
22	16	1 0 1 1 0	26
23	17	1 0 1 1 1	27
24	18	1 1 0 0 0	30
25	19	1 1 0 0 1	31
26	1A	1 1 0 1 0	32
27	1B	1 1 0 1 1	33
28	1C	1 1 1 0 0	34
29	1D	1 1 1 0 1	35
30	1E	1 1 1 1 0	36

$(11011.101)_2 = 1\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0$
$+ 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3}$

$= 16 + 8 + 0 + 2 + 1 + 0.5 + 0 + 0.125$

$= (27.625)_{10}.$

(iv) $(1010.1100)_2 = (10.75)_{10}$

02) Octal to Decimal :-

(i) $(36)_8 = 3\times8^1 + 6\times8^0 = 24 + 6 = (30)_{10}$

(ii) $(563)_8 = 5\times8^2 + 6\times8^1 + 3\times8^0 = 320 + 48 + 3 = (371)_{10}$

(iii) $(125.32) = 1\times8^2 + 2\times8^1 + 5\times8^0 + 3\times8^{-1} + 2\times8^{-2}$

$= 64 + 16 + 5 + 0.375 + 0.03125$

$= (85.40625)_{10}.$

03) Hexa decimal to decimal :-

(i) $(5AE)_{16} = 5\times16^2 + A\times16^1 + E\times16^0$

$= 256\times5 + (10\times16) + (14\times1)$

$= 1280 + 160 + 14 = (1454)_{10}$

(ii) $(35D.4F) = (\quad)_{10}$

$= 3\times16^2 + 5\times16^1 + D\times16^0 + 4\times16^{-1} + 15\times16^{-2}$

$= 3\times256 + 80 + 13\times1 + \frac{4}{16} + \frac{15}{256}$

$= (861.3085938)_{10}$

(iii) $(675.AC)_{16} = 6\times16^2 + 7\times16^1 + 5\times16^0 + \frac{10}{16} + \frac{12}{256}$

$= 1536 + 112 + 5 + 1.6 + 0.046875$

$= (1653.671875)_{10}$

04) Decimal to Binary :-

(i) $(565)_{10} = (1000110101)_2$

(ii) $(342.625)_{10} = (101010110.101)_2$

$0.625 \times 2 = 1.250$

$0.250 \times 2 = 0.500$

$0.5 \times 2 = 1.000$

(iii) $(125.123)_{10} = (1111101.00011)_2$

$0.123 \times 2 = 0.246$

$0.246 \times 2 = 0.492$

$0.492 \times 2 = 0.984$

$0.984 \times 2 = 1.968$

$1.968 \times 2 = 2.936$

05) Decimal to Octal :-

(i) $(365)_{10} = (555)_8$

(ii) $(286.052)_{10} = (436.032)_8$

$0.052 \times 8 = 0.416$

$0.416 \times 8 = 3.328$

$0.328 \times 8 = 2.624$

(iii) $(482.58)_{10} = (742.450)_8$

$0.58 \times 8 = 4.64$

$0.64 \times 8 = 5.12$

$0.12 \times 8 = 0.96$

(06) Decimal to Hexadecimal :-

(i) $(1532)_{10} = (5FC)_{16}$

(ii) $(356.49)_{10} = (164.707)_{16}$

$0.49 \times 16 = 7.84$

$0.84 \times 16 \quad 13.44$

(iii) $(2532.86)_{10} = (9E4.DC28)_{16}$

$0.86 \times 16 = 13.76$

$0.76 \times 16 = 12.16$

$0.16 \times 16 = 2.56$

$0.56 \times 16 = 8.96$

07. Octal to Binary :- radix = 8 ⇒ $8 = 2^{3}$

⇒ each number will be represented in 3 digits.

(i). $(563)_8 = (101\,110\,011)_2$

(ii). $(445.31)_8 = (100100101.011001)_2$

(iii). $(1234.56)_8 = (001\,010\,011\,100.101\,110)_2$

08) Binary to Octal :-

(i) $(110\,111\,0010)_2 = (1562)_8$ 001|101|110|010 → 1 5 6 2

(ii) $(011\,110\,000\,111.111\,100)_2 = (3607.74)_8$

(iii) $(11001101\,0101.00110)_2 = (6325.14)_8$

09) Hexadecimal to Binary :- $r = 16$, $16 = 2^{4}$, 4 bit

(i) $(ACF)_{16} = (1010\,1100\,1111)_2$

(ii) $(1AD.EC)_{16} = (0001\,1010\,1101.1110\,1100)_2$

(iii) $(25B.CD)_{16} = (0010\,0101\,1011.1100\,1101)_2$

(10) Binary to Hexadecimal :-

(i) $(11011100101)_2 = (6E5)_{16}$

(ii) $(111000110011.110110)_2 = (E33.D8)_{16}$

(iii) $(1010101011.110011)_2 = (2AB.CC)_{16}$

(11) Octal to Hexadecimal :- [Octal → Binary → Hexa]

(i) $(357)_8 = (EF)_{16}$

(ii) $(536.53)_8 = (15E.AC)_{16}$

(iii) $(135.67)_8 = (5D.DC)_{16}$

(12) Hexadecimal to Octal :-

(i) $(3AF.DE)_{16} = (1657.674)_8$

(ii) $(4BC.5A)_{16} = (2274.264)_8$

(iii) $(125.46)_{16} = (445.214)_8$

(iv) $(ABC.DE)_{16} = (5274.674)_8$

Problems :-

01. $(253.56)_8 = (\quad)_{10}$

02. $(123.4F8)_{16} = (\quad)_{10}$

03. $(536.32)_{10} = (\quad)_8$

04. $(1010101.010)_2 = (\quad)_{10}$

05. $(4326.95)_{10} = (\quad)_{16}$

06. $(1534.516)_{10} = (\quad)_2$

07. $(5316.72)_8 = (\quad)_{16}$

08. $(101011.110)_2 = (\quad)_8$

09. $(110110110111.110101) = (\quad)_{16}$

10. $(5FD.CA)_{16} = (\quad)_8$

11 $(3561.52)_8 = (\quad)_2$

12 $(4FA.CB)_{16} = (\quad)_{10}$

13 $(4AC.AB)_{16} = (\quad)_{10}$

14 $(952.58)_{10} = (\quad)_{16}$

15 $(101010.110)_2 = (\quad)_{10}$

01) $(253.56)_8 = (\quad\quad)_{10}$

Solution :- $= 2\times 8^2 + 5\times 8^1 + 3\times 8^0 + \frac{5}{8} + \frac{6}{8^2}$

$= 2\times 64 + 40 + 3 + 0.625 + 0.09375$

$= (171.71875)_{10}$

(02) $(123.4F8)_{16} = (\quad\quad)_{10}$

Solution :- $1\times 16^2 + 2\times 16 + 3\times 1 + \frac{4}{16} + \frac{15}{256} + \frac{8}{16^3}$

$= 256 + 32 + 3 + 0.25 + 0.05859375 + 0.00195$

$= (291.3105469)_{10}$

03. $(536.32)_{10} = (1030.2436)_8$

Solution :-

$0.32\times 8 = 2.56$

$0.56\times 8 = 4.48$

$0.48\times 8 = 3.84$

$0.84\times 8 = 6.72$

8 | 536
8 | 67 - 0
8 | 8 - 3
1 - 0

04) $(1010101.010)_2 = (\quad\quad)_{10}$

Solution :- $1\times 2^6 + 0\times 2^5 + 1\times 2^4 + 0\times 2^3 + 1\times 2^2 + 0\times 2^1 + 1\times 2^0$
$+ 0\times 2^{-1} + \frac{1}{2^2} + 0\times 2^{-3}$

$= 2^6 + 2^4 + 4 + 1 + \frac{1}{4} = 64 + 16 + 4 + 1 + 0.25 = (85.25)_{10}$

05) $(4326.95)_{10} = (\quad\quad)_{16}$

Solution :- $(10E6.F33)_{16}$

$0.95\times 16 = 15.2$

$0.2\times 16 = 3.2$

$0.2\times 16 = 3.2$

16 | 4326
16 | 270 - 6
16 | 16 - 14
1 - 0

(06) $(1534.516)_{10} = (10111111110.1000)_2$

$0.516 \times 2 = 1.032$

$0.032 \times 2 = 0.064$

$0.064 \times 2 = 0.128$

$0.128 \times 2 = 0.256$

2 | 1534
2 | 767 - 0
2 | 383 - 1
2 | 191 - 1
2 | 95 - 1
2 | 47 - 1
2 | 23 - 1
2 | 11 - 1
2 | 5 - 1
2 | 2 - 1
1 - 0

(07) $(5316.72)_8 = (ACE.E8)_{16}$

5316 ⇒ 1010|1100|1110 . 1110|1000|

A C E . E 8

(08) $(101011.110)_2 = (53.6)_8$

101|011|.|110|

5 3 . 6

(09) $(110110110111.110101)_2 = (1B6F.D4)_{16}$

1|1011|0110|1111|.|1101|0100|

1 B 6 F D 4

(10) $(5FD.CA)_{16} = (2775.624)_8$

8421 8421 8421 . 8421 8421

0101 1111 1101 . 1100 1010

010|111|111|101|.|110|010|10

2 7 7 5 : 6 2 4

(11) $(3561.72)_8 = (\quad)_2$

421 421 421 421 . 421 421

011 101 110 001 . 111 010

$(11101110001.11101)_2$

(12) $(4FA.CB)_{16} = (\quad)_{10}$

$= 4\times16^2 + 15\times16 + 10\times1 + \frac{12}{16} + \frac{13}{256}$

$= (1274.792969)_{10}$

$(4FA.CB)_{16} = (\quad)_2$

8421	8421	8421	.	8421	8421
0100	1111	1010	.	1100	1011

$(10011111010.11001011)_2$

(13) $(4AC.AB)_{16} = (\quad)_{10}$

Solution :- $4\times16^2 + 10\times16 + 12\times1 + \frac{10}{16} + \frac{11}{256}$

$= (1196.667969)_{10}$

(14) $(452.58)_{10} = (1C4.947)_{16}$

$0.58\times16 = 9.28$

$0.28\times16 = 4.48$

$0.48\times16 = 7.68$

$16 \,|\, 452$

$16 \,|\, 28 - 4$

$1 - 12$

(15) $(101010.110)_2 = (\quad)_{10}$

$1\times2^5 + 0\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0 + \frac{1}{2} + \frac{1}{4} +$

$= 32 + 8 + 2 . \frac{1}{2} + \frac{1}{4} = 42 + 0.75 = 42.75$

Conversion from any number system to any other number system:-

(i) $(5643.12)_7 = (\quad)_5$

Solution:- $5\times7^3 + 6\times7^2 + 4\times7 + 3 + \frac{1}{7} + \frac{2}{49}$

$5\times343 + 6\times49 + 28 + 3 + \frac{1}{7} + \frac{2}{49}$

$= (2040.184)$

$(2040.184)_{10} = (31130.043)_5$

$0.184 \times 5 = 0.92$

$0.92 \times 5 = 4.6$

$0.6 \times 5 = 3.$

$$\begin{array}{r|l} 5 & 2040 \\ 5 & 408 - 0 \\ 5 & 81 - 3 \\ 5 & 16 - 1 \\ 5 & 3 - 1 \\ & 0 - 3 \end{array}$$

(ii) $(1432)_5 = (\quad)_9$

$1 \times 5^3 + 4 \times 5^2 + 3 \times 5 + 2 \times 5^0 = 125 + 4 \times 25 + 15 + 2$

$= (242)_{10} = (288)_9$

$$\begin{array}{r|l} 9 & 242 \\ 9 & 26 - 8 \\ 9 & 2 - 8 \\ & 0 - 2 \end{array}$$

(iii) $(1453.52)_6 = (\quad)_8$

$= 1 \times 6^3 + 4 \times 6^2 + 5 \times 6^1 + 3 \times 6^0 + \frac{5}{6} + \frac{2}{36}$

$= 216 + 144 + 30 + 3 + 0.833 + \frac{2}{36}$

$= (393.8888889)_{10}$

$$\begin{array}{r|l} 8 & 393 \\ 8 & 49 - 1 \\ 8 & 6 - 1 \\ & 0 - 6 \end{array}$$

611

$0.8889 \times 8 = 7.1112$

$0.1112 \times 8 = 0.8896$

$0.8896 \times 8 = 7.1168$

$0.1168 \times 8 \quad 0.9344$

$= (393.8888889)_{10} = (611.7070)$

find the value of 'x' for the following

$(1x5.32)_8 = (85.40625)_{10}$

$(85.40625)_{10} = (125.32)_8$

$$\begin{array}{r|l} 8 & 85 \\ 8 & 10 - 5 \\ & 1 - 2 \end{array}$$

$x = 2$

Compliments :-

⇒ To simplify the subtraction operation

⇒ Logical calculations

They are of 2 types:-

01. Radix compliment (r's Compliment)

02. Diminished Radix compliment ($(r-1)$'s compliment)

Decimal Number System:-

10's Compliment

9's Compliment

⇒ $(r-1)$'s compliment can be calculated $\boxed{(r^n - 1) - N}$

where $r \rightarrow$ radix; $n \rightarrow$ digits no ; $N \rightarrow$ Given number

⇒ To calculate r's compliment $\boxed{r^n - N}$

Calculate the 9's & 10's compliment of the following.

		9's compliment		10's Compliment.
(i)	$(263.35)_{10}$ ⇒	$(736.64)_{10}$	=	$(736.65)_{10}$
(ii)	$(345.67)_{10}$ ⇒	$(654.32)_{10}$	=	$(654.33)_{10}$

Binary Number System:-

0+0=0
0+1=1
1+0=1
1+1=0 with carry 1

Calculate the 1's & 2's compliment of the following

		1's compliment		2's compliment
(i)	$(1011011)_2$ =	$(0100100)_2$	=	$(0100101)_2$
(ii)	$(11100101)_2$ =	$(00011010)_2$	=	$(00011011)_2$
(iii)	$(1010110)_2$ =	$(0101001)_2$	=	(0101010)

Calculate the 7's & 8's compliment of the following

		7's compliment		8's compliment
(1)	$(4536)_8$ =	$(3241)_8$	=	$(3242)_8$
(2)	$(1354)_8$ =	$(6423)_8$	=	$(6424)_8$

Calculate the 15's & 16's compliment of the following

	15's Compliment		16's compliment
(i) $(ABF)_{16}$ =	$(540)_{16}$	=	$(541)_{16}$
(ii) $(354C)_{16}$ =	$(CAB3)_{16}$	=	$(CAB4)_{16}$
	$(A3DF)_{16}$	=	$(A3E0)_{16}$

$$\begin{array}{r} 567 \\ +\;\;\;1 \\ \hline 570 \end{array} \qquad \begin{array}{r} A3DF \\ +1 \\ \hline A3E0 \end{array} \qquad \begin{array}{r} 5A3.4 \\ 33C.F \\ \hline 8E0.3 \end{array}$$

01. Write the 1st 10 numbers in radix 4,

Decimal	Radix 4
0	0
1	1
2	2
3	3
4	10
5	11
6	12
7	13
8	20
9	21

Careful imp

Subtraction using compliments.

01) Decimal Number System

Subtraction using 9's compliments.

Step 01:- Calculate the 9's compliment of the subtrahend

Step 02:- add it to the minuend.

Step 03:- If carry is present, the result is +ve.

Step 04:- If carry is not present, the result is -ve & again calculate 9's compliment for the answer.

Step 05:- and end around the carry

Perform the following subtraction using 9's compliment

(i) 534 - 236 = +298

534
763

1297
↳ +

298

(ii) 325 - 616 = -291

325 - 616 ⇒ 325
383 no carry
---
708 -ve ⇒ -291

M.S.B ⇒ Most significant Bit

M.S.D ⇒ Most significant digit

L.S.B ⇒ Least significant Bit

L.S.D ⇒ Least significant digit

Perform the following using 9's compliment method

(i) 536.4 - 329.6

536.4 ⇒ Minuend

+670.3 ⇒ 9's compliment of 329.6

1206.7

↳ 1 End around carry.

206.8.

(ii) 315.23 - 654.15

ignoring carry in 10's compliment

Solution :- 315.23

-345.84

661.07

As no carry is present, result is negative

∴ again calculating 9's compliment ⇒ - 338.92 [9's compliment of 661.07]

Perform the following using 10's compliment method

(i) 536.4 - 329.6

Solution :- 536.4 → Minuend

+670.4 → 10's compliment of 329.6

$\boxed{1}$206.8 → ignoring the carry.

206.8

536.4
329.6

206.8

(ii) 315.23 − 654.15

Solution :-

$$\begin{array}{r} 315.23 \rightarrow \text{Minuend} \\ 345.85 \rightarrow \text{10's complement of 654.15} \\ \hline 661.08 \end{array}$$

→ As no carry is present, result is −ve.

∴ Solution is −338.92 [10's complement of 661.08]

9's → 338.91

(iii) 123.61 − 429.54

Solution :-

$$\begin{array}{r} 123.61 \\ 570.46 \\ \hline 694.07 \end{array}$$

→ no carry therefore

Calculating 10's complement of 694.07 ⇒ [305.93]

= −305.93

Binary addition :-

(i) 0+0=0

(ii) 0+1=1

(iii) 1+0=1

(iv) 1+1=0 with carry is 1

(v) 1+1+1=1 with carry is 1

(a)

$$\begin{array}{r} 1000 \\ +\,0101 \\ \hline 1101 \end{array}$$

(b)

$$\begin{array}{rl} 1010111 & \rightarrow 87 \\ 0110111 & \rightarrow 55 \\ \hline 10001110 & \Rightarrow 142 \end{array}$$

> 142

(c)

$$\begin{array}{rl} 10101010 & \rightarrow 170 \\ 11101101 & \rightarrow 237 \\ \hline 110010111 & \Rightarrow 407 \end{array}$$

> 407

(d)

$$\begin{array}{r} 101011.101 \\ 011111.010 \\ \hline 1001010.111 \end{array}$$

Binary Subtraction:-

$0-0=0$

$1-0=1$

$1-1=0$

$0-1=1$ with borrow with 1

(a)
```
  1010
- 0101
------
  0101
```

(b)
```
  1000
  0110
------
  0010
```

(c)
```
  10101101 → 173
- 01010100 → 84
----------
  01011001 → 89
```

(d)
```
  100010101 → 277
  011101010 → 234
-----------
  000101011 → 43
```

M.S.B → 0 → +ve number
M.S.B → 1 → -ve number.

Subtraction using 1's complement method:-

Step 01:- Calculate the given 1's complement of subtrahend

Step 02:- Add it to the minuend.

Step 03:- If carry is present, the result is +ve and end around the carry

Step 04:- If carry is not present, the result is -ve & again calculate 9's complement for the answer.

(a) $x = 1000 \Rightarrow 8$

$y = 0101 \Rightarrow 5$

(i) $x-y \Rightarrow$
```
   1000 → Minuend
 + 1010 → 1's compliment of 0101
 ------
 1 0010 →
      1   end around carry
 ------
   0011 → final answer
```

(ii) $y - x \Rightarrow$ 0101 → Minuend

0111 → 1's complement of 1000

1100 → carry not present & result is -ve

∴ Solution is − 0011 ⇒ (1's complement of 1100)

(b) $x = 10101011$

$y = 01110111$

$x - y$ = 10101011 − 01110111 ⇒

```
   10101011
   10001000
 ----------
 1 00110011
          1
 ----------
   00110100
```

$y - x \Rightarrow$

```
  01110111
  01010100
 ----------
  11001011 ⇒ carry is absent
```

∴ Solution is − 00110100

(c) $x = 111001110$ → 462

$y = 101110110$ → 374

$x - y \Rightarrow$

```
   111001110
   010001001
 -----------
 1 001010111
           1
 -----------
   001011000 → 88
```

$y - x \Rightarrow$

```
  101110110
  000110001
 -----------
  110100111 ⇒ −001011000
```

374
88
462

Subtraction using 2's complement:

Step 01: Calculate 2's complement of subtrahend i.e add 1 to 1's complement

Step 02: Add it to minuend

Step 03: If carry is present, the result is +ve and ignore the carry

Step 04: If carry is absent, the result is -ve and again calculate 2's complement for answer.

(a) X = 1000

Y = 0101

(i) X-Y ⇒ 1000 → minuend

1011 → 2's complement of Y

ignoring carry ← $\boxed{1}$ 0011 → final answer

(ii) Y-X ⇒ 0101 → Minuend

no carry 1000 → 2's complement

M.S.B ← $\boxed{1}$101 → for answer, calculate 2's complement

i.e (0010+1) - 0011 → final answer.

1000 → 0111

1

1000

(b) X = 10101011

Y = 01110111

(i) X-Y ⇒ 10101011 → Minuend

10001001 → 2's complement

Ignore ← $\boxed{1}$ 00110100 → final answer

10001000

1

10001001

(ii) Y-X ⇒ 01110111 → Minuend

no carry & M.S.B is indicates a -ve number ← 01010101 → 2's complement,

$\boxed{1}$1001100 ⇒ 00110011

1

-00110100 } 2's complement of (11001100)

01010100

1

01010101

(c) $X = 11100 1110$

$Y = 101110110$

(i) $X - Y \Rightarrow$

```
  111001110
  010001010
ignore ← [1]1      111
  ---------
  001011000 → final answer
```

```
 010001001
+        1
 ---------
 010001010
```

(ii) $Y - X \Rightarrow$

```
    10111011 0
    00011001 0
no carry and M.S.B is 1 ← [1]1010 1000
```

```
 000110001
+        1
 ---------
 000110010
```

calculating 2's complement i.e [00101011 1 + 1]

i.e − 001011000

22nd January 2018

Signed Binary Numbers :- M.S.B determines the sign of number.

Decimal	Signed magnitude form	Signed 1's complement	Signed 2's complement
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
0	0000	0000	0000
−0	1000	1111	−
−1	1001	1110	1111
−2	1010	1101	1110
−3	1011	1100	1101
−4	1100	1011	1100
−5	1101	1010	1011
−6	1110	1001	1010
−7	1111		

1's complement subtraction for signed numbers

+6 → 00000110
-6 → 10000110
1's com -6 → 11111001

+13 → 0000 1101
-13 → 1000 1101
1's comp -13 → 1111 0010

+6 → 0 0000 110
+13 → 0000 1101
19 00010011

+13 → 0000,1101
-6 → 1111 1001
$\boxed{1}$ 00000110
→ 1
00000111

-13 → 11110010
+6 → 00000110
-7 $\boxed{1}$1111000 no carry

Calculating 1's complement
10000111

-13 11110010
-6 11111001
-19 1110101 1
→ 1
M.S.B is 1 -ve ← $\boxed{1}$1101100
16 8 4 2 1
∴ - 10010011

+38 → 00100110
-38 → 10100110
1's complement -38 → 11011001

+38 → 00100110
+25 → 00011001
63 00111111

+25 → 00011001
-25 → 10011001
1's comp -25 → 11100110

+38 → 00100110
-25 → 11100110
+13 $\boxed{1}$0 000,1100
→ 1
00001101 → +13

-38 → 11011001
+25 → 00011001
. no carry $\boxed{1}$1110010
- 10001101 → -13

-38 → ~~101 0010 110~~ 11011001

-25 → 11100110

$\boxed{1}$+

10111111

1

M.S.B is 1 → -ve ← $\boxed{1}$1000000

calculating 1's complement for result obtained

1011111

2's complement subtraction for signed numbers

+6 → 0000 0110

-6 → 1000 0110

1's com

-6 → 1111 1001

2's com

-6 → 11111 010

+13 → 0000 1101

-13 → 1000 1101

1's com

-13 → 1111 0010

2's com

-13 → 1111 0011

+13 → 0000 1101

-6 → 1111 1010

+7 ~~1~~ 00000111 neglecting carry.

+ve

-13 → 1111 0011

+6 → 0000 0110

-7 $\boxed{1}$1111001 no carry

-ve number

∴ calculating 2's complement

10000110

1

10000111 ⇒ -7

-13 → 11110011

-6 → 11111010

neglect ← $\boxed{1}$

$\boxed{1}$1101101

-ve number ⇒ 10010010 ⇒ -10010011

+38 → 00100110

-38 → 10100110

1's comp

-38 → 11011001

2's com

-38 → 11011010

+25 → 00011001

-25 → 10011001

1's comp

-25 → 11100110

2's comp

-25 → 11100111

$$\begin{array}{rr} +38 \rightarrow & 00100110 \\ +25 \rightarrow & 00011001 \\ \hline +63 & 00111111 \end{array}$$

$$\begin{array}{rr} +38 \rightarrow & 00100110 \\ -25 \rightarrow & 11100111 \\ \hline +13 & \boxed{1}00001101 \rightarrow +13 \end{array}$$

$$\begin{array}{rr} -38 \rightarrow & 11011010 \\ +25 \rightarrow & 00011001 \\ \hline -13 & \boxed{1}1110011 \end{array}$$ no carry.

-ve number

calculating 1's complement of 11110011 → 10001100

$$\begin{array}{r} 10001100 \\ 1 \\ \hline 10001101 \end{array} = -13$$

$$\begin{array}{rr} -38 \rightarrow & 11011010 \\ -25 \rightarrow & 11100111 \\ \hline -63 & \boxed{1}11000001 \end{array}$$

neglect

-ve

10111110 → 1's complement

$$\begin{array}{r} 10111110 \\ 1 \\ \hline 10111111 \end{array} \Rightarrow -63$$

2's complement

23rd January 2019

Binary Codes:

Binary codes

- Numeric codes
  - Weighted codes
    - 8421 code
    - 5421 code
    - 5211 code
    - 2421 code
    - 84-2-1 code
  - Non-weighted codes
    - Excess-3 code
    - Gray code
    - Five-bit B.C.D code
  - Self-complementing codes
    - Excess-3 code
    - 2421 code
    - 5211 code
    - 84-2-1 codes
  - Sequential code
    - BCD code → Binary coded decimal
    - Excess-3 code
  - Error detecting & correcting code
    - parity → detect
    - Hamming code → detect & correct
  - Cyclic codes → Gray code
  - Reflective code → Gray code
- Alpha numeric codes
  - ASCII
  - EBCDIC → Extended Binary Coded Decimal Interchange Code. (EBCDIC)
  - Hollerith

giving priority to L.S.B (decimal 0–4); giving priority to M.S.B (decimal 5–9)

Decimal number	8421 code	5421 Code	2421 Code	5211 Code	84-2-1 Code	Excess-3 Code (non weighted code)	
0	0000	0000	0000	0000	0000	0011	
1	0001	0001	0001	0001	0111	0100	4
2	0010	0010	0010	0011	0110	0101	5
3	0011	0011	0011	0101	0101	0110	6
4	0100	0100	0100	0111	0100	0111	7
5	0101	1000	1011	1000	1011	1000	8
6	0110	1001	1100	1010	1010	1001	9
7	0111	1010	1101	1100	1001	1011	10
8	1000	1011	1110	1110	1000	1011	11
9	1001	1100	1111	1111	1111	1100	12

Self complementing codes :-

⇒ 9 − 6 = 3 → 0110 ; 9 − 7 = 2 → 0101

1001 ← 1's complement → 1010

⇒ Self complementing codes are the codes in which 9 − N = x where x is the 1's complement of N.

ex:-
(i) Excess-3 code
(ii) 2421 code
(iii) 5211 Code
(iv) 84-2-1 Code.

Sequential Codes :-

Binary coded Decimal (B.C.D code) (8421):

⇒ In B.C.D code, each and every digit is represented with 4 bits.

⇒ In B.C.D 1010, 1011, 1100, 1101, 1110, 1111 are invalid codes.

* B.C.D addition:

⇒ The correction factor in B.C.D is +6.

⇒ The correction factor in Excess-3 code is +3.

⇒ Add 6 (0110) to the number if we get invalid code.

```
  25 → 0010 0101
+ 42 → 0100 0010
  ---------------
  67   0110 0111
         6    7
```

```
  54 → 0101 0100
  49 → 0100 1001
 ---------------
 103   1001 1101
          +0110
       ---------
      1010 0011
        10    3
```

```
  653 → 0110 0101 0011
+ 169 → 0001 0110 1001
  --------------------
  822   0111 1011 1100
          7   11   12
               Illegal code

0111 1011 1100
     0110 0110
--------------
1000 0010 0010
  8    2    2
```

* B.C.D Subtraction:

```
(a)  85 → 1000 0101
    -34 → 0011 0100
    ---------------
     51   0101 0001
            5    1
```

```
(b)  95 → 1001 0101
    -67 → 0110 0111
    ---------------
     28   0010 1110
            2   14 → Invalid
               -0110
          ---------
          0010 1000
            2    8
```

(c) 845 → 1000 0100 0101
-257 → 0010 0101 0111
588
0101 1110 1110
0110 0110
0101 1000 1000
$\underbrace{0101}_{5}\ \underbrace{1000}_{8}\ \underbrace{1000}_{8}$

(d) 756.4 → 0111 0101 0110 . 0100
-378.5 → 0011 0111 1000 . 0101
377.9
0011 1101 1101 . 1111
$\underbrace{0011}_{3}\ \underbrace{1101}_{13}\ \underbrace{1101}_{13}\ .\ \underbrace{1111}_{15}$

∴ 0011 1101 1101 . 1111
0110 0110 0110
0011 0111 0111 1001
$\underbrace{0011}_{3}\ \underbrace{0111}_{7}\ \underbrace{0111}_{7}\ .\ \underbrace{1001}_{9}$

Perform the following subtraction in 8421 BCD code using 9's complement method

(i) 56 → 56 → 0101 0110 → 56 in B.C.B
-32 → 67 → 0110 0111 → 9's complement of 32 in B.C.D
24
1011 1101
23
+1
24
+0110 +0110 (Adding 0110 for illegal codes)
0010 0011
→1 (End around the carry)
0010 0100 (Answer in B.C.D)
$\underbrace{0010}_{2}\ \underbrace{0100}_{4}$

(ii) 635.4 → 0110 0011 0101 . 0100   635.4
-427.7 → 0100 0010 0111 . 0111   572.2
207.7
1207.6
1
207.7

635.4 → 0110 0011 0101 . 0100
572.2 → 0101 0111 0010 . 0010

1011 1010 0111 . 0110
11 10 7 6
0110 0110

0010 0000 0111 . 0110
→ 1

0010 0000 0111 . 0111
2 0 7 . 7

(c) 256.5 → 256.5 → 0010 0101 0110 . 0101 → in
−429.7 → 570.2 → 0101 0111 0000 . 0010 → 9's
−173.2 826.7

0111 1100 0110 . 0111
−173.2 → 9's complement of 826.7 +0110 → adding 0110
1000 0010 0110 . 0111 no carry
8 2 6 . 7
1 7 3 2

(d) 123.6 → 123.6 → 0001 0010 0011 . 0110
−578.4 → 421.5 → ~~0100 0111 1000 . 1000~~
−454.8 545.1
0110 1001 1011 . 1010
454.8

123.6 → 0001 0010 0011 . 0110 → 123.6 in B.C.D
421.5 → 0100 0010 0001 . 0101 → 9's complement of 578

0101 0100 0100 . 1011
0110 → adding 0110

0101 0100 0101 . 0001
5 4 5 . 1

Perform the following subtracting in 8421 in B.C.D by using 10's complement.

(a) 95 → 95
−64 → 36 → 10's complement of 64.
31 31

95 → 1001 0101 → 95 in B.C.D
36 → 0011 0110 → 9's complement of 64 in BCD
31   1100 1011
0110 0110 → adding 0110 to illegal codes
0010 0001 → carry propagated and end carry ignored
0011 0001 → Answer is '31' in B.C.D

(b) 625.3 → 625.3 → 0110 0010 0101 . 0011
-456.4 → 543.6 → 0101 0100 0011 . 0110
1689   1011 0110 1000 . 1001
16 6 8 9

1011 0110 1000 . 1001
0110
0001 0110 1000 . 1001
1 6 8 . 9
→ Ignore carry

(c) 456.5 → 456.5 → 0100 0101 0110 . 0101 → 456.5 in BCD
-648.7 → 351.3 → 0011 0101 0001 . 0011 → 9's complement
807.8   0111 1010 0111 . 1000
-192.2   0110 → adding 0110
0111 0000 0111 . 1000
→ propagated
1000 0000 0111 . 1000 → no carry
8 0 7 . 8 →
- 1 9 2 . 2 → 10's complement

(d) 256.4 → 256.4
-625.8 → 374.2 → 10's complement of 625.8
630.6 → no carry ∴ result is -ve
-369.4 → 10's complement of 630.6

256.4 → 0010 0101 0110.0100 → 256.4 in B.C.D
374.2 → 0011 0111 0100.0010 → 10's complement of 625.8

630.6   0101 1100 1010.0110
-369.4   0110 0110   → adding 0110 to illegal code

0101 | 0010 | 0000.0110 → carry propagated

0110 0011 0000.0110
6 3 0 . 6 → Calculating 10's complement as no carry is present
-369.4

Excess-3 (XS-3) Code:

⇒ Excess-3 is non-weighted BCD code
⇒ It is sequential and self complementing.
⇒ Invalid codes are 0000(0), 0001(1), 0010(2), 1101(13), 1110(14), 1111(15)
⇒ The correction factor for Excess-3 code is +3
⇒ (0), (1), (2), (13), (14), (15) are invalid or illegal code in XS-3 code because, XS-3 code starts with 3 and ends at 12,

Excess-3 Addition:

→ If carry is generated, add 3(+0011)
→ If carry is not generated, subtract 3(-0011)

(i) 35 → 0110 1000 → 0110 1000 (35 in XS-3)
+46 → 0111 1001 → 0111 1001 (46 in XS-3)

81   1110 0001   1101 | 0001
1101 | 0001 (carry propogation)

1110 0001 (Add 0011 if carry
-0011 0011 generated subtract
1011 0100 0011 if carry not generated)
11 4

11 − 3   4 − 3

8   1 → Answer is 81 in XS-3

(ii) 546.32 → 1000 0111 1001.0110 0101

+354.16 → 0110 1000 0111.0100 1001

900.48   1110 1111 0000.1010 1110

(carry propogated)

1110 0000 0000.1010 1110

1111 0000 0000.1010 1110

−0011 0011 −0011 −0011 −0011

1100 0011 0011 .0111 1011 (Answer 900.48 in XS-3)

12   3   3   7   11

900.48

(iii) 236.54 → 0101 0110 1001.1000 0111 → 236.54 in XS-3

723.13 → 1010 0101 0110.0100 0110 → 723.13 in XS-3

959.67   1111 1011 1111.1100 1101 (No carry propogated)

−0011 −0011 −0011.0011 −0011 (Add 0011 if carry generated subtract 0011 generated)

1100 1000 1100.1001 1010

12   8   12   9   10

959.67

Excess-3 subtraction:-

→ if borrow is taken, subtract 3(0011)

→ if borrow is not taken Add 3(0011)

(i) 75 → 1010 1000 (75 in XS-3)

−39 → 0110 1100 (39 in XS-3)

36   0011 1100 (if borrow taken subtract 3 otherwise add 3)

+0011 −0011

0110 1001 (Answer 36 in XS-3)

6   9

36

11-3 4-3

8 1 → Answer is 81 in XS-3

(ii) 546.32 → 1000 0111 1001.0110 0101

+354.16 → 0110 1000 0111.0100 1001

900.48

1110 1111 0000.1010 1110

(carry propogated)

1110 0000 0000.1010 1110

1111 0000 0000 .1010 1110

-0011 0011 -0011 -0011 -0011

1100 0011 0011 .0111 1011 (Answer 900.48 in XS-3)

12 3 3 7 11

900.48

(iii) 236.54 → 0101 0110 1001.1000 0111 → 236.54 in XS-3

723.13 → 1010 0101 0110.0100 0110 → 723.13 in XS-3

959.67

1111 1011 1111.1100 1101 (No carry propogated)

-0011 -0011 -0011 .0011 -0011 (Add 0011 if carry generated subtract 0011 generated)

1100 1000 1100 .1001 1010

12 8 12 9 10

959.67

Excess-3 subtraction:

→ if borrow is taken, subtract 3(0011)

→ if borrow is not taken Add 3(0011)

(i) 75 → 1010 1000 (75 in XS-3)

-39 → 0110 1100 (39 in XS-3)

36

0011 1100 (if borrow taken subtract 3 otherwise add 3)

+0011 -0011

0110 1001 (Answer 36 in XS-3)

6 9

36

Scanned by CamScanner

(ii) 845.32 → 1011 0111 1000 . 0110 0101
-634.58 → 1001 0111 0111 . 1000 1011

210.74

0010 0001 0000 . 1101 1010
0011 0011 0011 . 0011 -0011

0101 0100 0011 . 1001 0111 Answer in XS-3
5 4 3 . 9 7

210.74

(iii) 736.21 → 1010 0110 1001 . 0101 0100
-498.35 → 0111 1100 1011 . 0110 1000

237.86

0010 1001 1001 . 1110 1100
0011 -0011 -0011 . 0011 -0011

0101 0110 1010 . 1011 1001

237.86 5 6 10 11 9

01. Perform the following subtraction in XS-3 using 9's complement method.

(i) 63.54 → 63.54 → 1001 0110 . 1000 0111 (63.54 in XS-3)
39.42 → 60.57 → 1001 0011 . 1000 1010 (9's complement of 39.42 in XS-3)

①24.11
24.12

0010 0001 0000 0000 (Carry propogate end around carry)

0010 1010 . 0001 0010
+0011 -0011 +0011 +0011 (add 0011 if carry generated subtract 0011 if carry not generated)

0101 0111 0100 0101
5 7 . 4 5

24.12

(ii) 845.32 → 845.32 → 1011 0111 1000 . 0110 0101
634.58 → 365.41 → 0110 1001 1000 . 0111 0100

①210.73

0001 0000 0000 . 1101 1001

210.74

0010 0001 0000 . 1101 1010
+0011 +0011 +0011 . -0011 -0011

0101 0100 0111 . 1010 0111
5 4 3 10 7

210.74

(iii) 328.56 → 328.56 (9's complement of 653.75)
-653.75 → 346.24 (As no carry present again
674.80 Calculate 9's complement)
325.19

328.56 → 0110 0101 1011 . 1000 1001
346.24 → 0110 0111 1001 . 0101 0111
1000 1100 0100 . 1101 0000
1000 1101 0100 . 1110 0000
-0011 -0011 +0011 -0011 +0011
1101 1010 0111 . 1011 0011
9 10 7 11 3
-3 -3 -3 -3 -3
6 7 4 . 8 0

325.19 [9's complement of 674.80]

(iv) 251.78 → 251.78
-648.43 → 351.56
603.34 → As no carry present again
i.e 396.65 calculate 9's complement

251.78 → 0101 1000 0100 . 1010 1011
351.56 → 0110 1000 0100 . 1000 1001
1011 0000 1000 0010 0100
1100 0000 1001 . 0011 0100
-0011 +0011 -0011 0011 +0011
1010 0011 0110 . 0110 0111
9 3 6 . 6 7

603.34 ⇒ 396.65 [9's complement of 603.34]

20/01/19

Perform the following subtraction in Excess-3 using 10's complement method

(i) 936
-521

(ii) 745
-368

(iii) 745
-368

(i) 936 ⇒ 936 → 1100 0110 1001 → 936 in Xs'3

− 521 ⇒ 479 → 0111 1010 1100 → 10's complement of 521

415 | 1 415 | 1 0011 1 0000 1 0101 → carry propagated

0100 0001 0101

478
+1
479

0011 0011 0011 Adding 0011 if carry generated

0111 0100 1000

7 4 8 Answer in Xs'3

4 1 5 Answer

(ii) 745 → 745 → 1010 0111 1000 → 745 in Xs'3

−368 → 632 → 1001 0110 0101 → 10's complement of 368

377 | 1 377 | 1 0011 1101 1101

+0011 −0011 −0011 → subtracting 0011 when no carry is present

0110 1010 1010

634
+1
632

6 10 10 Answer in Xs'3

3 7 7 Answer

(iii) 542 → 542 → 1000 0111 0101 → 542 in Xs'3

− 736 → 264 → 0101 1010 0111 → 10's complement of 736

−194 | 806 | 1101 1 0001 1100 → carry propagated

−0011 +0011 −0011 → adding & subtracting for presence of carry & absent

263 | 193
+1 | +1
264 | −194

1001 0100 1001

−11 8 9

8 0 6 → answer

1 9 3 → 9's complement of 806

+1

−1 9 4 ⇒ final answer.

(01) 346 → 346 → 0110 0100 1001 → 346 in XS'3

-531 → 469 → 0111 1001 1100 ⇒ 10's complement of 531

-185   815   1101 0000 0101 → carry propagated

0101

1100 0001

468   184   -0011 +0011 +0011 → adding +0011

+1   +1

469   -185   1011 0100 1000

11 4 8

8 1 5

1 8 4

+ 1 ⇒ calculating 10's comp

-1 8 5 ⇒ final answer

Error detecting and correcting codes.

⇒ Error detecting code → Parity

⇒ Error correction code → Hamming code.

⇒ Parity — Even parity → Even no. of 1's
— Odd Parity → Odd no. of 1's

Decimal	Binary Code	Even Parity	Odd parity	
0	0000	0	1	
1	0001	1	0	
2	0010	1	0	
3	0011	0	1	
4	0100	1	0	
5	0101	0	1	
6	0110	0	1	
7	0111	1	0	
8	1000	1	0	
9	1001	0	1	

In an even parity scheme, which of the following words contain an error

(a) 10110110 (b) 11011011 (c) 10101000

answer is a & c as they contain odd number of ones.

In an odd parity scheme which of the following words contain an error

(a) 11011101 (b) 11011010 (c) 111011000

answer is a, & c as they contain even number of ones.

Hamming Code:- it is a correction and detection code. It detects only one error.

These are of 3 types

01) 7-bit hamming code

02) 12-bit hamming code

03) 15-bit hamming code

7-bit hamming code:-

⇒ encode the message bit 1010 in an even parity hamming code

1	2	3	4	5	6	7
P_1	P_2	D_3	P_4	D_5	D_6	D_7
		1		0	1	0

P_1 is set to 0 or 1 to have even parity scheme.

$P_1 \rightarrow (P_1 D_3 D_5 D_7) = (P_1\ 1\ 0\ 0) = 1$

$P_2 \rightarrow (P_2 D_3 D_5 D_6) = (P_2\ 1\ 0\ 1) = 0$

$P_4 \rightarrow (P_4 D_5 D_6 D_7) = (P_4\ 0\ 1\ 0) = 1$

Transmitted code $\Rightarrow X = P_1\ P_2\ D_3\ P_4\ D_5\ D_6\ D_7$

$1\ 0\ 1\ 1\ 0\ 1\ 0$

Recieved code $\Rightarrow Y = 1\ 0\ \cancel{1}\ 0\ 1\ 0\ 1\ 0$

$P_1 P_2 D_3 P_4 D_5 D_6 D_7$

check sum:-

$c_1 = P_1 \oplus D_3 \oplus D_5 \oplus D_7 = 1 \oplus 0 \oplus 0 \oplus 0 \Rightarrow 1$

$c_2 = P_2 \oplus D_3 \oplus D_6 \oplus D_7 = 0 \oplus 0 \oplus 1 \oplus 0 \Rightarrow 1$ $\Rightarrow 3$

$c_3 = P_4 \oplus D_5 \oplus D_6 \oplus D_7 = 1 \oplus 0 \oplus 1 \oplus 0 \Rightarrow 0$

3rd bit has an error.

for 1011, 1110, 0101 generate a 7-bit hamming code using even parity.

$P_1\ P_2\ D_3\ P_4\ D_5\ D_6\ D_7$

$1\ \ 0\ 1\ 1$

$P_1 \to (P_1 D_3 D_5 D_7) \to (P_1\ 1\ 0\ 1) = 0$

$P_2 \to (P_2 D_3 D_6 D_7) \to (P_2\ 1\ 1\ 1) = 1$

$P_4 \to (P_4 D_5 D_6 D_7) \to (P_4\ 0\ 1\ 1) = 0$

7 bit hamming code is $P_1\ P_2\ D_3\ P_4\ D_5\ D_6\ D_7$

$0\ 1\ 1\ 0\ 0\ 1\ 1$

(02) $1110 \Rightarrow P_1\ P_2\ D_3\ P_4\ D_5\ D_6\ D_7$

$1\ \ 1\ 1\ 0$

$P_1 \to (P_1 D_3 D_5 D_7) \to (P_1\ 1\ 1\ 0) \Rightarrow 0$

$P_2 \to (P_2 D_3 D_6 D_7) \to (P_2\ 1\ 1\ 0) \Rightarrow 0$

$P_4 \to (P_4 D_5 D_6 D_7) \to (P_4\ 1\ 1\ 0) \Rightarrow 0$

7-bit hamming code is $P_1\ P_2\ D_3\ P_4\ D_5\ D_6\ D_7$

$0\ 0\ 1\ 0\ 1\ 1\ 0$

(iii) 0101 $\Rightarrow$ P_1 P_2 D_3 P_4 D_5 D_6 P_7

0 1 0 1

$P_1 \rightarrow (P_1\ D_3\ D_5\ D_7) \rightarrow (P_1\ 011) \rightarrow 0$

$P_2 \rightarrow (P_2\ D_3\ D_6\ D_7) \rightarrow (P_2\ 001) \rightarrow 1$

$P_4 \rightarrow (P_4\ D_5\ D_6\ D_7) \rightarrow (P_4\ 101) \rightarrow 0$

7-bit hamming code is P_1 P_2 D_3 P_4 D_5 D_6 D_7

0 1 0 0 1 0 1

In a communication system the recieved code words are 1011011, 1010101, 1110110 using 7 bit hamming code. Find transmitted code words.

P_1 P_2 D_3 D_4 D_5 P_6 D_7

1 0 1 1 0 1 1

$c_1 = (P_1\ D_3\ D_5\ D_7) \Rightarrow 1 \oplus 1 \oplus 0 \oplus 1 \Rightarrow 1$

$c_2 = P_2 \oplus D_3 \oplus D_6 \oplus D_7 \Rightarrow 0 \oplus 1 \oplus 1 \oplus 1 \Rightarrow 1$

$c_3 = P_4 \oplus D_5 \oplus D_6 \oplus D_7 \Rightarrow 1 \oplus 0 \oplus 1 \oplus 1 \Rightarrow 1$

$0\overset{4}{1}\overset{2}{1}\overset{1}{1} = 7$

The transmitted code 7 bit hamming code is = $\underline{\underline{1011010}}$

(ii) 1010 101

P_1 P_2 D_3 P_4 D_5 D_6 D_7

1 0 1 0 1 0 1

$c_1 = P_1 \oplus D_3 \oplus D_5 \oplus D_7 \Rightarrow 1 \oplus 1 \oplus 1 \oplus 1 \Rightarrow 0$

$c_2 = P_2 \oplus D_3 \oplus D_6 \oplus D_7 \Rightarrow 0 \oplus 1 \oplus 0 \oplus 1 \Rightarrow 0$

$c_3 = P_4 \oplus D_5 \oplus D_6 \oplus D_7 \Rightarrow 0 \oplus 1 \oplus 0 \oplus 1 \Rightarrow 0$

0

Transmitted code is 1010101,,

(iii) 1110110

P_1	P_2	D_3	P_4	D_5	D_6	D_7
1	1	1	0	1	1	0

$c_1 = P_1 \oplus D_3 \oplus D_5 \oplus D_7 \Rightarrow 1 \oplus 1 \oplus 1 \oplus 0 \Rightarrow 1$ 0011

$c_2 = P_2 \oplus D_3 \oplus D_6 \oplus D_7 \Rightarrow 1 \oplus 1 \oplus 1 \oplus 0 \Rightarrow 1$

$c_3 = P_4 \oplus D_5 \oplus D_6 \oplus D_7 \Rightarrow 0 \oplus 1 \oplus 1 \oplus 0 \Rightarrow 0$

error is in 3rd bit

transmitted code is 1100110

generate a 12 bit hamming code, using even parity scheme if the message bits are 10110110

Solution:-

P_1	P_2	D_3	P_4	D_5	D_6	D_7	P_8	D_9	D_{10}	D_{11}	D_{12}
		1		0	1	1		0	1	1	0

P_1 is is set to 0 or 1 using even parity.

$P_1 \rightarrow (P_1, D_3, D_5, D_7, D_9, D_{11}) = (P_1\ 10101) = 1$

P_2 is set to 0 or 1 using even parity

$P_2 \rightarrow (P_2\ D_3\ D_6\ D_7\ D_{10}\ D_{11}) \rightarrow (P_2\ 11111) = 1$

P_4 is set to 0 or 1 using even parity

$P_4 \rightarrow (P_4\ D_5\ D_6\ D_7\ D_{12}) \rightarrow (P_4\ 0110) = 0$

P_8 is set to 0 or 1 using even parity.

$P_8 \rightarrow (P_8\ D_9\ D_{10}\ D_{11}\ D_{12}) = (P_8\ 0110) = 0$

12- bit hamming code is 111001100110

In a 12-bit hamming code, if the received hamming code is 111001101110 when transmitted through a noisy channel, find the transmitted code word.

$$Y = \begin{array}{cccccccccccc} P_1 & P_2 & D_3 & P_4 & D_5 & D_6 & D_7 & P_8 & D_9 & D_{10} & D_{11} & D_{12} \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \end{array}$$

$C_1 = P_1 \oplus D_3 \oplus D_5 \oplus D_7 \oplus D_9 \oplus D_{11} = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 1$

$C_2 = P_2 \oplus D_3 \oplus D_6 \oplus D_7 \oplus D_{10} \oplus D_{11} = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 0$

$C_4 = P_4 \oplus D_5 \oplus D_6 \oplus D_7 \oplus D_{12} = 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 0$ 9

$C_8 = P_8 \oplus D_9 \oplus D_{10} \oplus D_{11} \oplus D_{12} = 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = 1$

The error is at 9^{th} bit

∴ The transmitted code is 111001100110

03. Generate a 15-bit haming code for the following message bits

(a) 10110110101 (b) 11101011110 (c) 11001100111

(a) 10110110101

$$Y = \begin{array}{ccccccccccccccc} P_1 & P_2 & D_3 & P_4 & D_5 & D_6 & D_7 & P_8 & D_9 & D_{10} & D_{11} & D_{12} & D_{13} & D_{14} & D_{15} \\ & & 1 & & 0 & 1 & 1 & & 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{array}$$

P_1 is set to zero or one using even parity.

$P_1 \rightarrow (P_1 \oplus D_3 \oplus D_5 \oplus D_7 \oplus D_9 \oplus D_{11} \oplus D_{13} \oplus D_{15})$

$= P_1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = (P_1\ 1010111) = 1$

$P_2 \rightarrow$ is set to zero or one using even parity.

$P_2 \rightarrow (P_2 D_3 D_6 D_7 D_{10} D_{11} D_{14} D_{15}) = (P_2\ 1\ 1\ 1\ 1\ 1\ 0\ 1) = 0$

P_4 is set to zero or one using even parity.

$P_4 = (P_4 D_5 D_6 D_7 D_{12} D_{13} D_{14} D_{15}) = (P_4\ 0\ 1\ 1\ 0\ 1\ 0\ 1) = 0$

$P_8 = (P_8 D_9 D_{10} D_{11} D_{12} D_{13} D_{14} D_{15}) = (P_8\ 0\ 1\ 1\ 0\ 1\ 0\ 1) = 0$

The 15-bit hamming code is 101001100110101

(b) 11101011110

$Y = P_1\ P_2\ D_3\ P_4\ D_5\ D_6\ D_7\ P_8\ D_9\ D_{10}\ D_{11}\ D_{12}\ D_{13}\ D_{14}\ D_{15}$

1 1 1 0 1 0 1 1 1 1 0

P_1 is set to zero or one using even parity.

$P_1 \rightarrow (P_1 D_3 D_5 D_7 D_9 D_{11} D_{13} D_{15}) \rightarrow (P_1\ 1\ 1\ 0\ 1\ 1\ 1\ 0) = 1$

P_2 is set to zero or one using even parity.

$P_2 \rightarrow (P_2 D_3 D_6 D_7 D_{10} D_{11} D_{14} D_{15}) \rightarrow (P_2\ 1\ 1\ 0\ 0\ 1\ 1\ 0) = 0$

P_4 is set to zero or one using even parity.

$P_4 \rightarrow (P_4 D_5 D_6 D_7 D_{12} D_{13} D_{14} D_{15}) \rightarrow (P_4\ 1\ 1\ 0\ 1\ 1\ 1\ 0) = 1$

P_8 is set to zero or one using even parity

$P_8 \rightarrow (P_8 D_9 D_{10} D_{11} D_{12} D_{13} D_{14} D_{15}) \rightarrow (P_8\ 1\ 0\ 1\ 1\ 1\ 1\ 0) = 1$

The transmitted code is 101111011011110

(c) 11001100111

$Y = P_1\ P_2\ D_3\ P_4\ D_5\ D_6\ D_7\ P_8\ D_9\ D_{10}\ D_{11}\ D_{12}\ D_{13}\ D_{14}\ D_{15}\ D_{16}$

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

$P_1 \rightarrow (P_1 D_3 D_5 D_7 D_9 D_{11} D_{13} D_{15}) = (P_1\ 1\ 1\ 0\ 1\ 0\ 1\ 1) = 1$

$P_2 \rightarrow (P_2 D_3 D_6 D_7 D_{10} D_{11} D_{14} D_{15}) = (P_2\ 1\ 0\ 0\ 1\ 0\ 1\ 1) = 0$

$P_4 \rightarrow (P_4 D_5 D_6 D_7 D_{12} D_{13} D_{14} D_{15}) = (P_4\ 1\ 0\ 0\ 0\ 1\ 1\ 1) = 0$

$P_8 \rightarrow (P_8 D_9 D_{10} D_{11} D_{12} D_{13} D_{14} D_{15}) = (P_8\ 1\ 1\ 0\ 0\ 1\ 1\ 1) = 1$

The transmitted code is 101010011100111

Gray Code :-

- ⇒ It is a non-weighted code
- ⇒ It is cyclic and Reflective code
- ⇒ Successive codewords differ by 1 bit position.
- ⇒ It is also called as unit-distance code.

1 bit	2 bit	3 bit
0	00	000
1	01	001
	11	011
	10	010
		110
		111
		101
		100

Calculate the 4-bit Gray code for the binary decimal numbers zero to 15.

Solution

Decimal numbers	Binary code	Gray code
0	0000	0000 - 0
1	0001	0001 - 1
2	0010	0011 - 3
3	0011	0010 - 2
4	0100	0110 - 6
5	0101	0111 - 7
6	0110	0101 - 5
7	0111	0100 - 4
8	1000	1100 - 12
9	1001	1101 - 13
10	1010	1111 - 15
11	1011	1110 - 14
12	1100	1010 - 10
13	1101	1011 - 11
14	1110	1001 - 9
15	1111	1000 - 8

calculate the Gray code for the following binary codes

(a) 10110110 (b) 11100010 (c) 11001010 (d) 11110000

11101101 10010011 10101111 10001000

Solution

(a) 10110110 ↓ 11101101

(b) 1110001 ↓ 10010010

(c) 11001010 ↓ 10101111

(d) 11110000

10001000

for the following Gray code, calculate the numbers in decimal.

(a) 11101101 → 10110110 = 128 + 32 + 16 + 4 + 2 = 182

(b) 10010010 → 11100010 = = 226

(c) 10101111 → 11001010 = = 202

(d) 10001000 → 11110000 = = 240

(d) 10001000 (c) 10101111 (b) 10010010 11101101

11110000 11001010 11100010 10110110

$\Rightarrow 10110110 \Rightarrow 128+32+16+4+2 = 182$

$\Rightarrow 11100010 = 128+64+32+2 = 226$

$\Rightarrow 11001010 = 128+64+8+2 = 202$

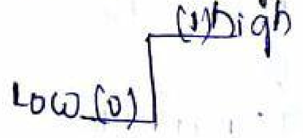
$\Rightarrow 11110000 = 128+64+32+16 = 240$

Logic Gates :-

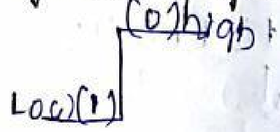
⇒ These are the basic building block of digital system

⇒ Logic Gate has the ability to make the decision.

Positive Logic



Negative Logic



Logic design :-

⇒ The interconnection of gates to perform a logical operation is known as 'Logic design'

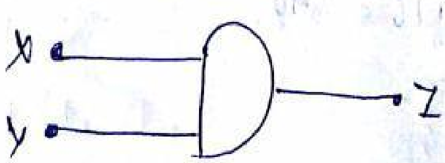
Truth Table :-

* The list of all possible combinations with inputs and outputs is known as truth table

There are 7 Logic Gates :-

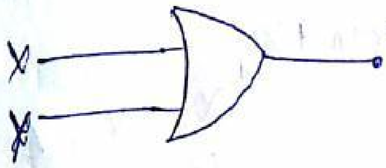
01. AND Gate
02. OR Gate } Basic Gates
03. NOT Gate
04. NAND Gate } Universal Gates
05. NOR Gate
07. Ex-OR Gate
08. Ex-NOR Gate

01) AND Gate :- $(X \cdot Y)$



X	Y	Z
0	0	0
1	0	0
0	1	0
1	1	1

OR Gate :- $(X+Y)$



$Z = X+Y$

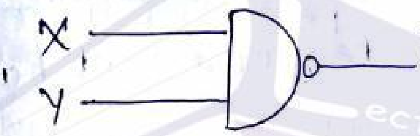
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

03. NOT Gate :- ( )

$Y = \bar{X}$

X	Y
0	1
1	0

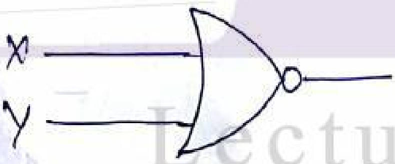
04. NAND Gate :-



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

X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

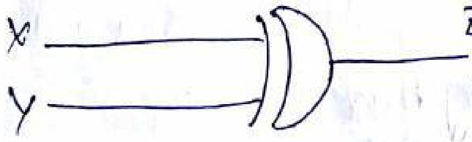
05. NOR Gate :-



$Z = \overline{X+Y}$

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

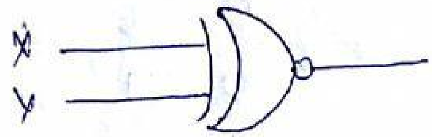
06. Ex-OR Gate :-



$Z = X \oplus Y = X\bar{Y} + \bar{X}Y$

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

07) Ex- NOR Gate :-



$Z = \overline{X \oplus Y} = \bar{X}\bar{Y} + XY$

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

30th January 2019 :-

Boolean Algebra :-

* Boolean Algebra is a system of mathematical logic, any complex function can be represented as leave, in logic design

Logic Operation :-

01) AND Operation
02) OR Operation
03) NOT Operation
04) NAND & NOR Operation
05) X-OR & X-NOR Operation

Axioms of Boolean Algebra

Axioms ⇒ No mathematical equation to prove

01) $0.0 = 0$
02) $0.1 = 0$
03) $1.0 = 0$
04) $1.1 = 1$
05) $0+0 = 0$

06). $0+1=1$

07). $1+0=1$

08). $1+1=1$

09). $\bar{0}=1$

10). $\bar{1}=0$

* Laws of Boolean Algebra:-

(01) Complementation Law:-

(i) $\bar{0}=1$

(ii) $\bar{1}=0$

(iii) if $A=0$ then $\bar{A}=1$

(iv) if $A=1$ then $\bar{A}=0$

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

(02) AND Law:-

(i) $A\cdot 0=0$

(ii) $A\cdot 1=A$

(iii) $A\cdot A=A$

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

(03) OR Law:-

(i) $A+0=A$

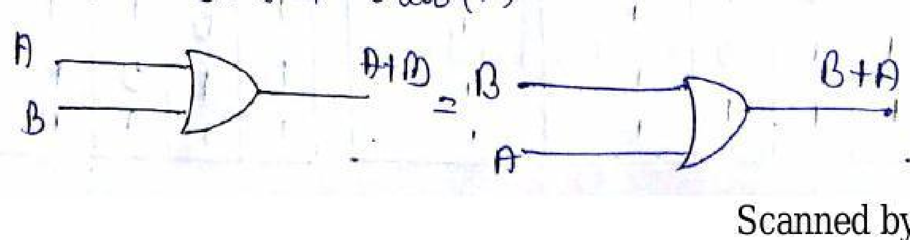
(ii) $A+1=1$

(iii) $A+A=A$

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

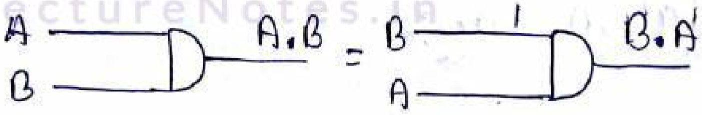
(04) Commutative Law:-

(i) $A+B=B+A \Rightarrow$ Law(1)



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

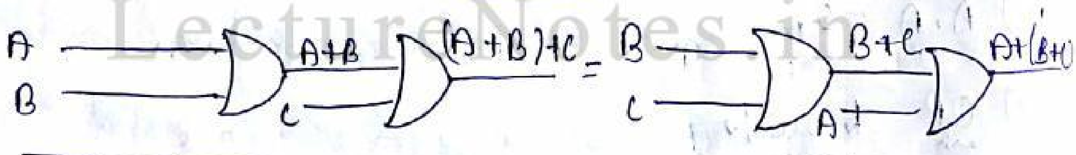
Law (ii) :- $A \cdot B = B \cdot A$



A	B	A.B		B	A	B.A
0	0	0		0	0	0
0	1	0		0	1	0
1	0	0		1	0	0
1	1	1		1	1	1

(05) Associative Law :-

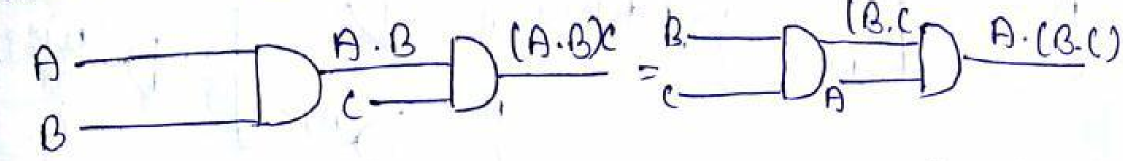
Law (i) :- $(A+B)+C = A+(B+C)$



A	B	C	A+B	(A+B)+C
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
1	0	0	1	1
0	1	1	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

A	B	C	B+C	A+(B+C)
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	0	0	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

Law (ii) :- $(A.B).C = A.(B.C)$

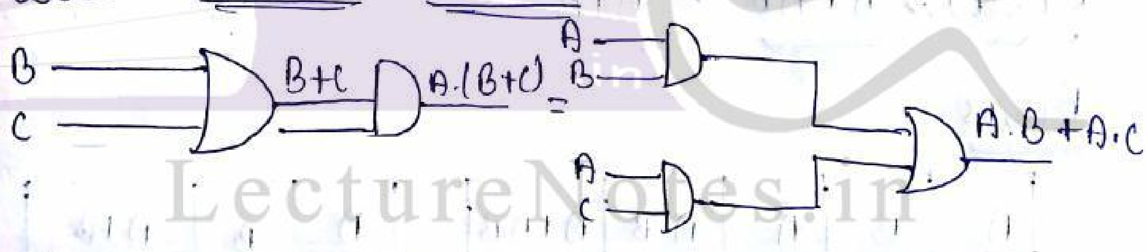


A	B	C	A.B	(A.B)C
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

A	B	C	(B.C)	A.(B.C)
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

(06) Distributive Law :-

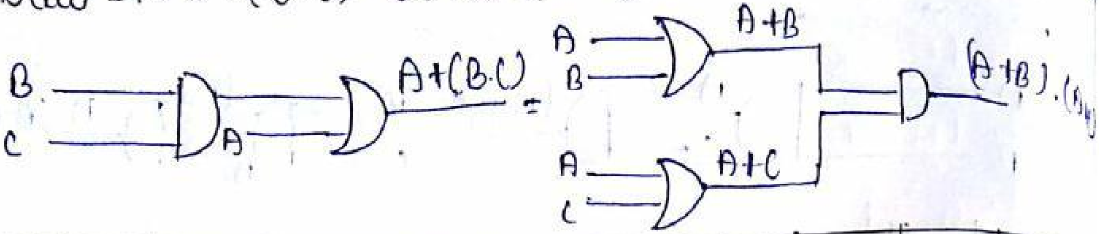
Law 01 :- $A.(B+C) = AB + AC$



A	B	C	B+C	A.(B+C)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

A	B	C	A.B	A.C	A.B + A.C
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

Law 2 :- $A+(B \cdot C) = (A+B) \cdot (A+C)$



A	B	C	B.C	A+C
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

A	B	C	A+B	A+C	(A+B)(A+C)
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	1	1	1
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

07. Redundant Literal Rule [RLR] :-

Law(i) :- $A + \bar{A}B = A+B$

Proofs

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

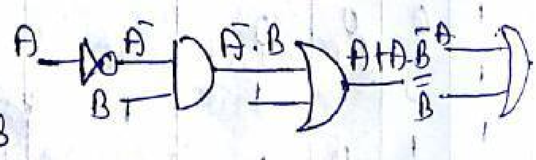
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

Proof :- L.H.S $= A + \bar{A}B$

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

$= 1 \cdot (A+B)$

$= (A+B)$



Law (ii) :- $A\cdot(\bar{A}+B) = A\cdot B$

A	B	$\bar{A}$	$\bar{A}+B$	$A\cdot(\bar{A}+B)$
0	0	1	1	0
0	1	1	1	0
1	0	0	0	0
1	1	0	1	1

A	B	$A\cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

$L.H.S = A\cdot(\bar{A}+B)$

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

$= A.B$ //

(8) Idempotente Law :-

Law (i) :- $A+A = A$

Proof : If $A=0$ then $A+A = 0+0 = 0 = A$

If $A=1$ then $A+A = 1+1 = 1 = A$

Law (ii) :- $A\cdot A = A$

Proof :- If $A=0$ then $A.A = 0.0 = 0 = A$

30th 2:pm If $A=1$ then $A.A = 1.1 = 1 = A$.

(9) Absorption Law :-

Law 1 :- $A + AB = A$

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

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

Law (ii) :- $A\cdot(\bar{A}+B) = A\cdot B$

A	B	$\bar{A}$	$\bar{A}+B$	$A\cdot(\bar{A}+B)$
0	0	1	1	0
0	1	1	1	0
1	0	0	0	0
1	1	0	1	1

A	B	$A\cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

$L.H.S = A\cdot(\bar{A}+B)$

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

$= A.B$ //

(8) Idempotence Law :-

Law (i) :- $A+A=A$

Proof : If $A=0$ then $A+A = 0+0 = 0 = A$

If $A=1$ then $A+A = 1+1 = 1 = A$

Law (ii) :- $A\cdot A = A$

Proof :- if $A=0$ then $A.A = 0.0 = 0 = A$

if $A=1$ then $A.A = 1.1 = 1 = A$

30th 2:pm

(9) Absorption Law :-

Law 1 :- $A+AB = A$

A+B

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

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

Law (ii) $A.(A+B) = A$

Proof :- $A.(A+B) = A.A + A.B$

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

$= A$

A	B	A + B	A.(A+B)
0	0	0	0
0	1	1	0
1	0	1	1
1	1	1	1

(10) Consensus Theorem :- [Included factor Theorem]

Law (i) :- $AB + \bar{A}C + BC = AB + \bar{A}C$

Proof :- $AB + \bar{A}C + BC = AB + \bar{A}C + BC(A+\bar{A})$

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

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

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

Law (ii) :- $(A+B)(\bar{A}+C).(B+C) = (A+B).(\bar{A}+C)$

Proof :-

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

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

L.H.S $= AC + \bar{A}B + BC(B+C) =$

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

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

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

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

11) Transposition Theorem :-

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

Proof :- $R.H.S = (\bar{A}+B)(A+C)$

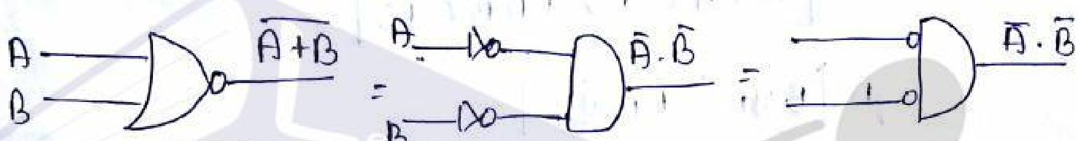
$= A \cdot \bar{A}+\bar{A} C+A B+B C$

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

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

(12) Demorgan's Theorem :-

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

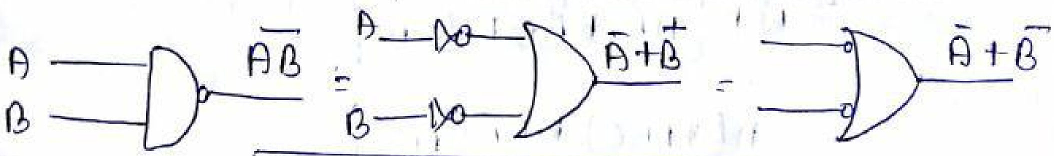


Bubbled And Gate = NOR Gate

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

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

Law (ii) :- $\overline{A \cdot B}=\bar{A}+\bar{B}$



Bubbled OR Gate = NAND Gate

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

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

Reduce the following expression using laws of Boolean Algebra:-

(a) $A[B+\bar{C}(\overline{\bar{A}B+A\bar{C}})]$

Solution:- $A\left[B+\bar{C}\left(\overline{\bar{A}B}.\overline{A\bar{C}}\right)\right]$

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

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

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

$= A\left[B+\bar{A}\bar{C}+0+\bar{A}\bar{B}\bar{C}+0\right]$

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

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

$= AB+0+0$

$= AB$

(b) $A+B[AC+(B+\bar{C})D]$

Solution:- $A+B[AC+BD+\bar{C}D]$

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

$= A(1+BC)+BD[1+\bar{C}]$

$= A+BD$

(c) $\left(\overline{A+\bar{B}\bar{C}}\right)(A\bar{B}+ABC)$

Solution:- $(\bar{A}\cdot\overline{\bar{B}\bar{C}})(A\bar{B}+ABC)$

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

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

$= 0$

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

Solution:- $(B+BC)(B+\bar{B}C)(B+D)$

$(B+BC)(B+D) = B + \underset{0}{B\bar{B}}C + BC + \bar{B}CB \; (B+D)$

$B(1+C)(B+D) = B + BC[B+D]$

$B(B+D) = B + BC + BCD$

$B+BD = B[1+D] = B$ $\quad = B + BC[1+D] = B+BC = B[1+C] = B$

Show that $AB + A\bar{B}C + B\bar{C} = AC + B\bar{C}$

Solution:- $A(B+\bar{B}C) + B\bar{C}$ → Redundant Literal Rule

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

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

$= AC + B\bar{C} + AB$ → By conses Theorem

$= AC + B\bar{C}$

Show that $A\bar{B}C + B + B\bar{D} + AB\bar{D} + \bar{A}C = B+C$,

Solution:- $A\bar{B}C + B + B\bar{D} + AB\bar{D} + \bar{A}C$

$A\bar{B}C + B(1+\bar{D}+AB)AB\bar{D} + \bar{A}C$

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

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

$= A\bar{B}C + B[1+\bar{D}+A\bar{D}] + \bar{A}C$

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

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

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

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

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

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

$= \bar{A}C + B + C = B + C[\bar{A}+1] = \underline{\underline{B+1}}$, Hence prooved.

Duality and Compliment :-

Find the duality of the following expressions

(a) $A.(A.B) = A.B$

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

(b) $\overline{\bar{A}B + \bar{A} + AB} = 0$

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

(c) $AB + \overline{AC} + A\bar{B}C[AB+C] = 1$

$(A+B) \cdot \overline{(A+C)} \cdot (A+\bar{B}+C) + [(A+B) \cdot C] = 0$

(d) $A + B(\overline{C+\bar{D}E}) = A + B\bar{C}DE$

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

(e) $\overline{A\bar{B}+ABC} + A(B+A\bar{B}) = 0$

$\overline{A+\bar{B} \cdot (A+B+C)} \cdot A + (B \cdot A + \bar{B}) = \underline{1}$

1st february 2019.

find the complement of the following expressions

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

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

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

(b) $\bar{B}\bar{C}D + (\overline{B+C+D}) + \bar{B}\bar{C}\bar{D}E = \overline{\bar{B}\bar{C}D + (\overline{B+C+D}) + \bar{B}\bar{C}\bar{D}E}$

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

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

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

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

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

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

(e) $(\overline{ABC})(\overline{A+B+C})$

Solution :- apply whole complement

$\overline{(\overline{ABC}) \cdot (\overline{A+B+C})}$

Replace 'and' with 'or'

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

Sum of products [S.O.P] and Product of sums [P.O.S] forms:-

Ex:- $\underbrace{\bar{A}B}_{\text{product}} + \underbrace{\bar{A}B\bar{C}}_{\text{product}} + \underbrace{ABC}_{\text{product}} + A$

⇒ S.O.P form (or) disjunctive normal form [DNF]

⇒ $ABC, \bar{A}B\bar{C}, \bar{A}BC$ ⇒ Standard SOP (or) Expanded SOP, disjunctive cannocial form (or) cannonical SOP

Ex:- $\underbrace{(\bar{A}+B)}_{\text{Sum}} \underbrace{(\bar{A}+B+\bar{C})}_{\text{Sum}} \cdot \underbrace{(B+\bar{C})}_{\text{Sum}}$

The above form is POS form (or) conjunctive normal form [CNF]

→ converts it into

standard POS (or) Expanded POS, conjunctive cannonical form (or) Cannonical POS

Minterm:-
Each product term is standard SOP form is known as Minterm.

Maxterm:- Each sum term in standard POS form is known as Maxterm.

X	Y	Z	Min term term	Designation	Max term 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

P₂) Convert the following expressions into standard SOP form.

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

Solution ⇒ $\bar{A}B + B(A+\bar{A})$

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

$= \bar{A}B + BA \Rightarrow 01 + 11 \Rightarrow m_1 + m_3 \Rightarrow \Sigma m(1,3)$

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

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

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

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

Scanned by CamScanner

$= 011\ 0 + 111 + 101 + 011 \Rightarrow \Sigma m(3,5,6,7)$
$\quad 6 \quad\quad 7 \quad\quad 5 \quad\quad 3$

(iii) $f(A,B,C) = A + AB + BCA$

Solution :- $A(B+\bar{B})(C+\bar{C}) + AB(C+\bar{C}) + BCA$

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

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

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

$= 111 + 110 + 101 + 100$
$\quad 7 \quad\quad 6 \quad\quad 5 \quad\quad 4$

$= m_7 + m_6 + m_5 + m_4$

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

Expand the following expressions into standard POS form

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

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

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

$= (\bar{A}+B)\cdot(A+B)\cdot(A+\bar{B})$
$\quad 1\ 0 \quad\quad 0\ 0 \quad\quad 0\ 1$

$= M_2\ M_0\ M_1$

$= \Pi M(0,1,2)$

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

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

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

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

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

$= (A+B+C)\ (A+B+\bar{C})(A+\bar{B}+C)(A+\bar{B}+\bar{C})(\bar{A}+B+C)\ (\bar{A}+B+\bar{C})\ (\bar{A}+B+\bar{C})$
$\quad 0\ 0\ 0 \quad\quad 0\ 0\ 1 \quad\quad 0\ 1\ 0 \quad\quad 0\ 1\ 1 \quad\quad 1\ 0\ 0 \quad\quad 1\ 0\ 1$

$= \Pi M(0,1,2,3,4,5)$

Expand the following into minterms and maxterms

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

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

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

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

$\quad 01 \quad 00 \quad 10$

$= 1 \quad 0 \quad 2$

$\Rightarrow \Sigma m(0,1,2) \rightarrow$ Minterm; $\Pi M(3) \rightarrow$ Maxterm

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

Solution: $A(B+\bar{B})(C+\bar{C})(D+\bar{D})+B\bar{C}(A+\bar{A})(D+\bar{D})$
$+AB\bar{D}(C+\bar{C})+ABCD$

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

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

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

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

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

$AB+A\bar{B}\cdot(CD+C\bar{D}+\bar{C}D+\bar{C}\bar{D})+B\bar{C}A+\bar{A}B\bar{C}(D+\bar{D})+ABC\bar{D}+AB\bar{C}\bar{D}$
$+ABCD$

$=ABCD+AB\bar{C}\bar{D}+AB\bar{C}D+AB\bar{C}\bar{D}+A\bar{B}CD+A\bar{B}C\bar{D}+A\bar{B}\bar{C}D$
$+A\bar{B}\bar{C}\bar{D}+B\bar{C}AD+B\bar{C}A\bar{D}+\bar{A}B\bar{C}D+\bar{A}B\bar{C}\bar{D}+ABC\bar{D}+AB\bar{C}\bar{D}$
$+ABCD$

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

Weekly Assesment Test - 2

(01) Each product sum term in standard Product of Sums is __________.

(02) Illegal codes in X_s-3 are __________

(03) ______ Binary code is Reflective, Cyclic and Non-weighted code

(04) Add the octal numbers 273.56 + 425.07

273.56 → 010 111 011 . 101 110
425.07 → 100 010 101 . 000 111
111 010 000 . 110 101
7 2 0 . 65

273.56 →
425.07 →

(05) Subtract the hexadecimal 67F2.6E − 4A0E.A9

(06) Complement of $(\overline{ABC})(\overline{A+B+C})$ is

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

(08) find the error in 1010100

(09) 463.12 - 875.34 in excess - 3 using 10's complement method

(10) State & proove Demorgan's Laws.

5th February 2019

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

Solution:- $f = A(\bar{B}+A)B$

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

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

0 0 0 1 1 0

$= \Pi M(0,1,2)$

$= m(3)$

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

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

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

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

000 001 010 011 100 101

$= \Pi M(0,1,2,3,4,5)$

$= \Sigma m(6,7)$

(e) Write the algebraic terms of a variable expression having the following minterms.

(a) m_0

(b) m_5

(c) m_9

(d) m_{14}

Given minterm	M_0	m_5	m_9	m_{14}
Binary form	0000	0101	1001	1110
Algebraic form	$\bar{A}\bar{B}\bar{C}\bar{D}$	$\bar{A}B\bar{C}D$	$A\bar{B}\bar{C}D$	$ABC\bar{D}$

(f) Write the algebraic term of a four variable expression having the following maxterms.

(a) M_3 (b) M_9 (c) M_{11} (D) M_{14}.

Given max term	M_3	M_9	M_{11}	M_{14}
Binary Form	0011	1001	1011	1110
Algebraic form	$A+B+\bar{C}+\bar{D}$	$\bar{A}+B+C+\bar{D}$	$\bar{A}+B+\bar{C}+\bar{D}$	$\bar{A}+\bar{B}+\bar{C}+D$

02. Boolean Algebra

K maps:-

⇒ 'K' stands for "Karnaugh Maps"

⇒ A K-map is a chart or graph composed of arrangement of adjacent cells, each representing a sum term (or) product term

⇒ A "n-variable K-map will have 2^n possible combinations (or) cells (or) squares.

⇒ 2- variable K-map → 4 squares

⇒ 3- Variable K- map → 8 squares

⇒ 4- variable K- map → 16 squares

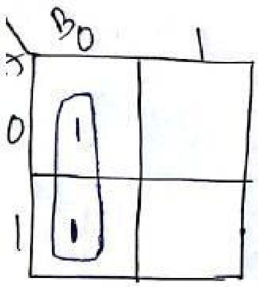
⇒ 5- Variable K- map → 32 squares

⇒ 6 - Variable K- map → 64 squares

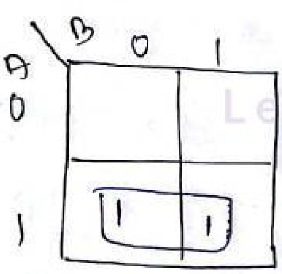
2- Variable K-map:-

$m_0 = \bar{A}\bar{B}$ $\quad m_1 = \bar{A}B$ $\quad m_2 = A\bar{B}$ $\quad m_3 = AB$

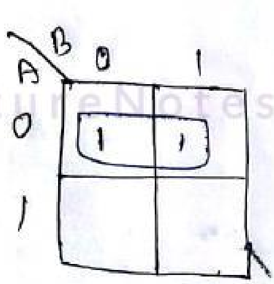
A \ B	0	1
0	$\bar{A}\bar{B}$ (0)	$\bar{A}B$ (1)
1	$A\bar{B}$ (2)	AB (3)



$f = \bar{B}$



$f = A$



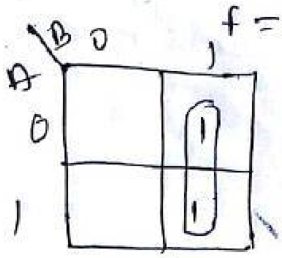
$f = \bar{A}$

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

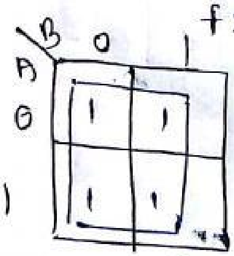
$\bar{B}(A+\bar{A})$

$\bar{B}\cdot 1$

$= \bar{B}$



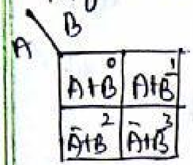
$f = B$

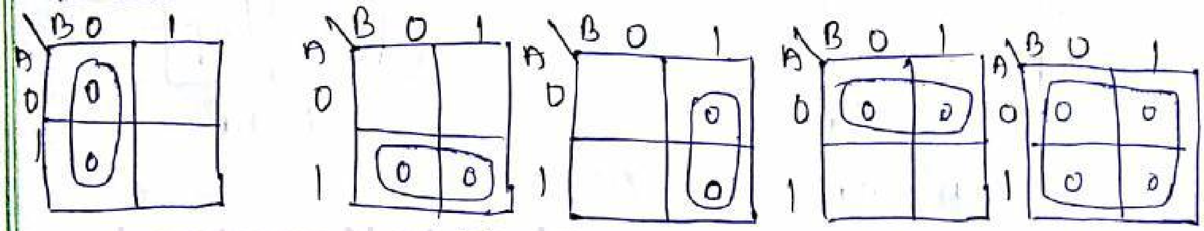


$f = 1$

mapping of P.O.S Expressions:-

$M_0 = A + B \quad M_1 = A + \bar{B} \quad M_2 = \bar{A} + B \quad M_3 = \bar{A} + \bar{B}$





$F = B \qquad F = \bar{A} \qquad F = \bar{B} \qquad F = A \qquad F = 0$

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

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

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

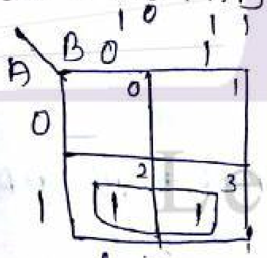
$B(A+\bar{A}) + B$

$B + B$

$= B$

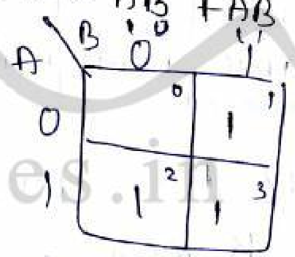
Reduce the following expressions by using K-map.

(i) $F = A\bar{B} + AB \Rightarrow S.O.P$ $\Sigma m(2,3)$



$f = A$

(ii) $f = A\bar{B} + AB + \bar{A}B \Rightarrow S.O.P$ $\Sigma m(1,2,3)$



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

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

$= A + \bar{A}B$

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

$= \underline{A+B}$

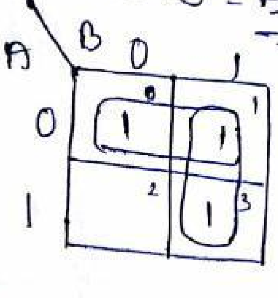
(iii) $F = \bar{A}\bar{B} + \bar{A}B + AB \Rightarrow S.O.P$

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

$= \bar{A} + AB$

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

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



Reduce the following expression using K-map.

(i) $(A+B)(\bar{A}+B)(\bar{A}+\bar{B})$ $\quad$ 0 0, 1 0, 1 1

$\Pi M(0,2,3)$

A \ B	0	1
0	0 (0)	(1)
1	0 (2)	0 (3)

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

(ii) $f = (A+B)(A+\bar{B})(\bar{A}+\bar{B})$ $\quad$ 0 0, 0 1, 1 1

$\Pi M(0,1,3)$

A \ B	0	1
0	0 (0)	0 (1)
1	(2)	0 (3)

$f = A\bar{B}$

3- Variable K-map :-

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

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

Gray code

000 → 000 → 0
001 → 001 → 1
010 → 011 → 3
011 → 010 → 2
100 → 110 → 6
101 → 111 → 7
110 → 101 → 5
111 → 100 → 4

Reduce the expression $f = \bar{A}\bar{B}C + A\bar{B}C + \bar{A}B\bar{C} + AB\bar{C} + ABC$ using K-map.

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

001 $\quad$ 101 $\quad$ 010 $\quad$ 110 $\quad$ 111 $= \Sigma m(1,2,5,6,7)$

A \ BC	00	01	11	10
0	(0)	1 (1)	(3)	1 (2)
1	(4)	1 (5)	1 (7)	1 (6)

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

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

$\quad\ \ 0\ 0\ 0 \qquad\ 1\ 0\ 1 \qquad\quad 1\ 1\ 1 \qquad\quad 0\ 1\ 1 \qquad\quad 1\ 1\ 0$

Solution:- $\Pi M(0,5,7,3,6)$

A \ BC	00	01	11	10
0	0^{0}	^{1}	0^{3}	^{2}
1	^{4}	0^{5}	0^{7}	0^{6}

If we combine only one term → we get 3 literals.

If we combine only 2 terms → we get 2 literals.

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

If we combine only 4 terms → we get only 1 literals.

Implement the following functions in NAND Gate

(i) $f(A,B,C) = \Sigma(1,2)$ (ii) $F(A,B,C) = \Sigma(0,6)$

(iii) $F(A,B) = \Sigma(1,2)$

(i) $f(A,B,C) = \Sigma(1,2)$ ⇒ taking 3-variable K-map.

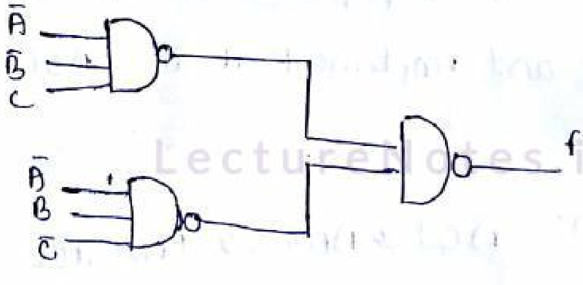
A \ BC	00	01	11	10
0	^{0}	1^{1}	^{3}	1^{2}
1	^{4}	^{5}	^{7}	^{6}

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

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

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

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

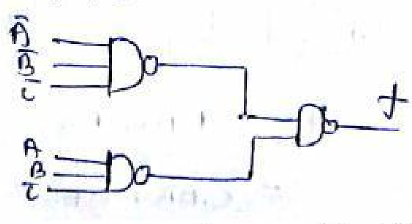


(ii) $f(A,B,C) = \Sigma(0,6)$ ⇒ Considering 3 variable K-map

A \ BC	00	01	11	10
0	1^{0}	^{1}	^{3}	^{2}
1	^{4}	^{5}	^{7}	1^{6}

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

$f = \overline{\overline{\bar{A}\bar{B}\bar{C} + AB\bar{C}}} \Rightarrow \overline{\overline{\bar{A}\bar{B}\bar{C}} . \overline{AB\bar{C}}}$



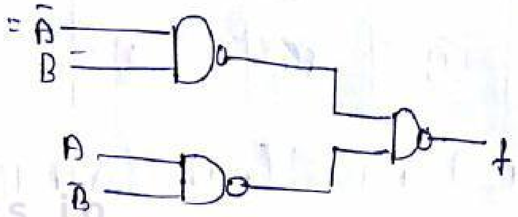
Scanned by CamScanner

(iii) $f(A,B) = \Sigma(1,2) \Rightarrow$ Considering 2 variable K-map

A \ B	0	1
0	(0)	1 (1)
1	1 (2)	(3)

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

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



Simplify using K-map and implement the same using NAND Gate

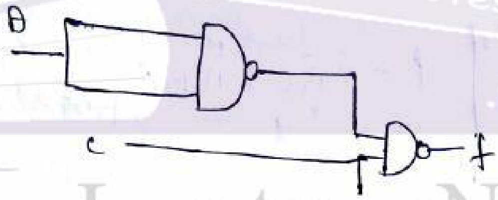
(i) $f(A,B,C) = \Sigma(0,2,4,5,6,7)$

A \ BC	00	01	11	10
0	1 (0)	(1)	(3)	1 (2)
1	1 (4)	1 (5)	1 (7)	1 (6)

Group 4,5,6,7 so that we will get 1 variable

Group 0,4,2,6

$f = A + \bar{C} \Rightarrow \overline{\overline{A + \bar{C}}} = \overline{\bar{A} \cdot C}$



Reduce the expression $f = \Sigma m(0,2,3,4,5,6)$ using mapping and implement it in AOI logic and NAND logic.

AOI $\Rightarrow$ And OR Inverter

Solution: AOI $\Rightarrow$ And or Inverter

$f = \Sigma m(0,2,3,4,5,6)$

A \ BC	00	01	11	10
0	1 (0)	(1)	1 (3)	1 (2)
1	1 (4)	1 (5)	(7)	1 (6)

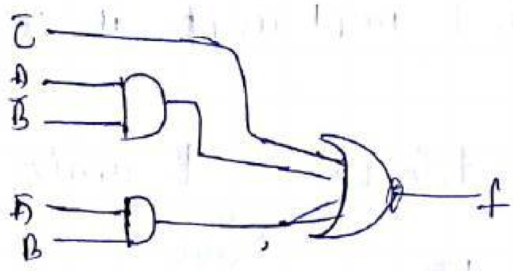
Grouping 0,4,2,6 gives 1 variable.

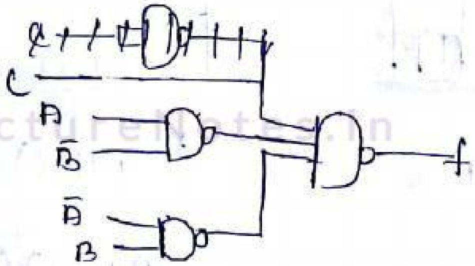
3 & 5 are grouped to themselves

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

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

AOI logic





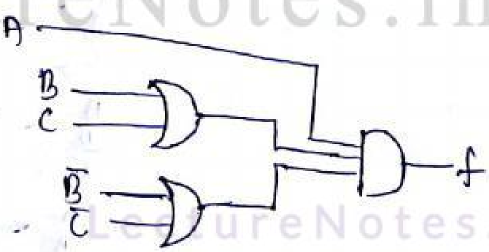
Reduce the expression $f = \Pi M(0,1,2,3,4,7)$ using mapping and implement it in OAI logic and NOR Logic

OAI → Or And Inverter

Solution:-

A \ BC	00	01	11	10
0	0 (0)	0 (1)	0 (3)	0 (2)
1	0 (4)	(5)	0 (7)	(6)

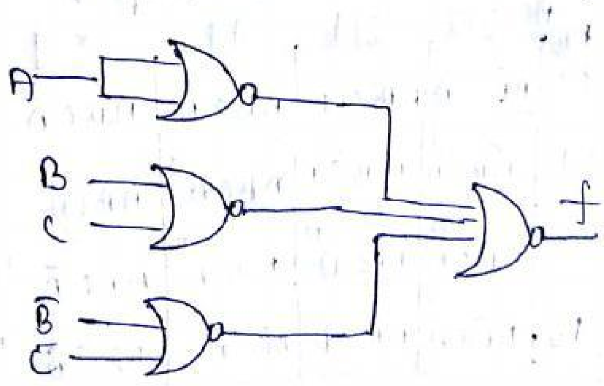
$f = A(B+C)(\bar{B}+\bar{C})$



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

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

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



Obtain the real minimal expression for $f = \Sigma m(1,2,4,6,7)$ and implement it using Universal gates.

Solution $f = \Sigma m(1,2,4,6,7)$ $\qquad$ $f = \Pi M(0,3,5)$

A \ BC	00	01	11	10
0	[0]	1 [1]	[3]	1 [2]
1	1 [4]	[5]	1 [7]	1 [6]

A \ BC	00	01	11	10
0	0 [0]	[1]	0 [3]	[2]
1	[4]	0 [5]	[7]	[6]

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

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

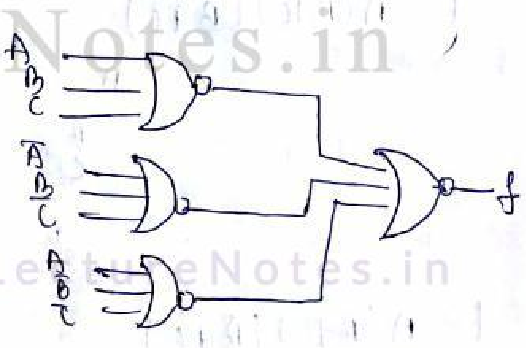
$= \overline{\overline{(A+B+C)(\bar{A}+B+\bar{C})(A+\bar{B}+\bar{C})}}$

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

As real minimal expression is considered we need to take any one of the above through which we can minimize the no. of gates.

$f = \Sigma m(1,2,4,6,7)$ needs 5 gates and $f = \Pi M(0,3,5)$ needs only 4 gates.

⇒ And if we get same no. of gates we should go for NAND.



4-Variable K-map:

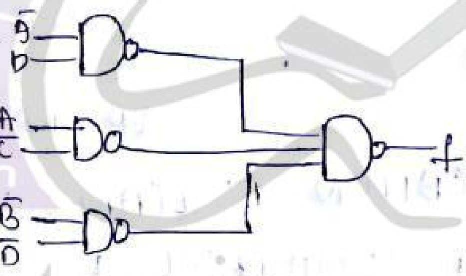
AB \ CD	00	01	11	10
00	$\bar{A}\bar{B}\bar{C}\bar{D}$ [0]	$\bar{A}\bar{B}\bar{C}D$ [1]	$\bar{A}\bar{B}CD$ [3]	$\bar{A}\bar{B}C\bar{D}$ [2]
01	$\bar{A}B\bar{C}\bar{D}$ [4]	$\bar{A}B\bar{C}D$ [5]	$\bar{A}BCD$ [7]	$\bar{A}BC\bar{D}$ [6]
11	$AB\bar{C}\bar{D}$ [12]	$AB\bar{C}D$ [13]	$ABCD$ [15]	$ABC\bar{D}$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

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

Reduce the expression $f = \Sigma m(0,1,2,3,5,7,8,9,10,12,13)$ and implement the expression by using universal logic

Solution :- Draw a 4-variable K-map.

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



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

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

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

Reduce the expression $\Pi M(2,8,9,10,11,12,14)$ and implement the expression by using universal logic

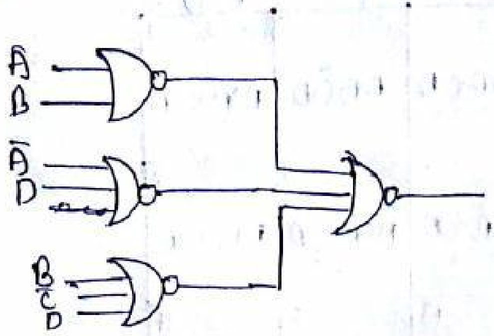
AB \ CD	00	01	11	10
00	(0)	(1)	(3)	0 (2)
01	(4)	(5)	(7)	(6)
11	0 (12)	(13)	(15)	0 (14)
10	0 (8)	0 (9)	0 (11)	0 (10)

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

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

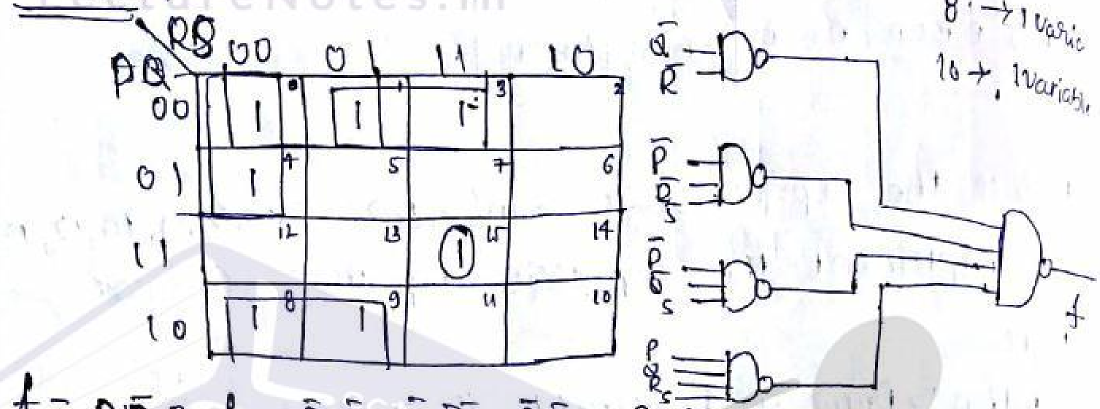
$= \overline{\overline{(\bar{A}+D)} + \overline{(B+\bar{C}+D)} + \overline{(\bar{A}+B)}}$

Using NOR Gate:-



$f(P,Q,R,S) = \Sigma m(0,1,3,4,8,9,15)$

Solution:-



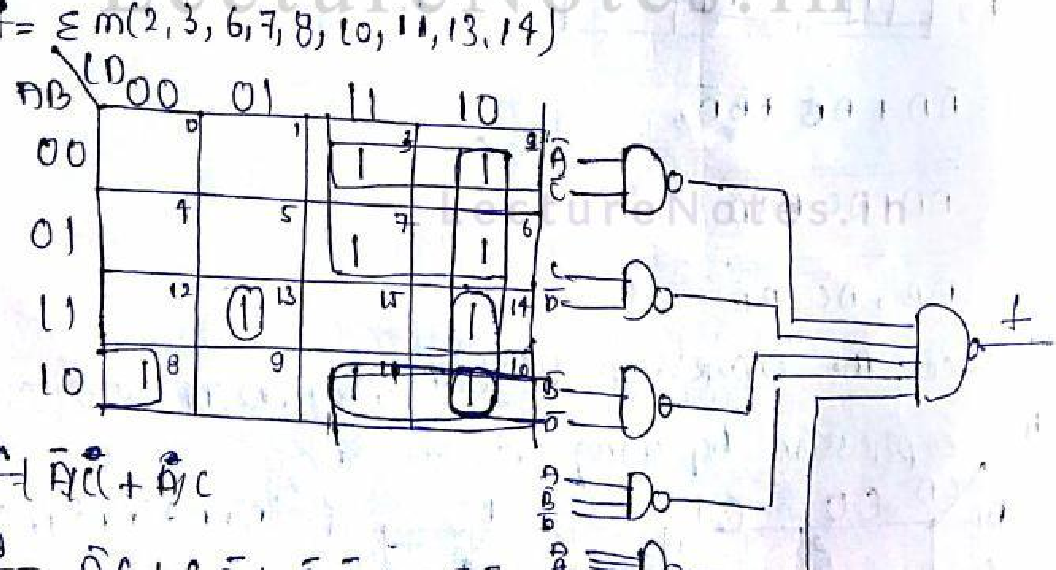
$f = \bar{Q}\bar{R} + \bar{P}\bar{R}\bar{S} + \bar{P}\bar{Q}S + PQRS$

$= \overline{\overline{\bar{Q}\bar{R} + \bar{P}\bar{R}\bar{S} + \bar{P}\bar{Q}S + PQRS}}$

$= \overline{\overline{\bar{Q}\bar{R}} \cdot \overline{\bar{P}\bar{R}\bar{S}} \cdot \overline{\bar{P}\bar{Q}S} \cdot \overline{PQRS}}$

OBTAIN BOOLEAN EXPRESSION AND IMPLEMENT IT IN UNIVERSAL LOGIC

$f = \Sigma m(2,3,6,7,8,10,11,13,14)$



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

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

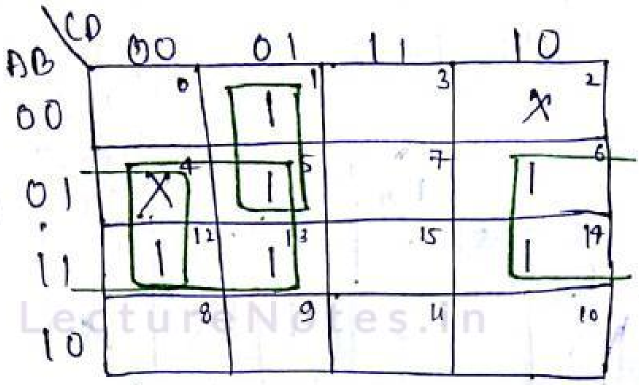
$= \overline{\overline{\bar{A}C + C\bar{D} + \bar{B}\bar{C} + A\bar{B}D + AB\bar{C}D}}$

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

K-MAP SIMPLIFICATION USING DON'T CARES:

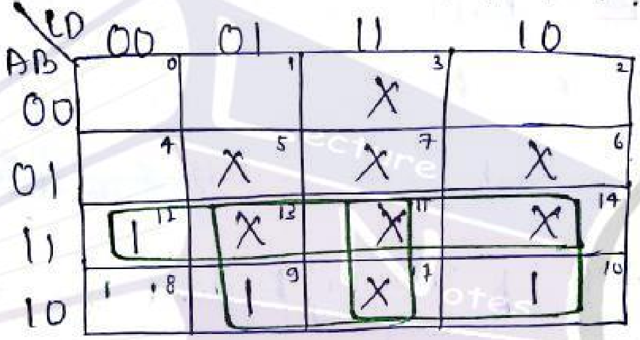
Reduce the following expression using K-Maps

$f = \Sigma m(1,5,6,12,13,14) + d(2,4)$



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

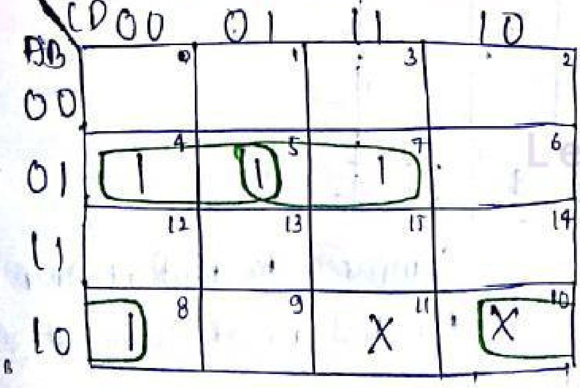
$f = \Sigma m(9,10,12) + d(3,5,6,7,11,13,14,15)$



$f = AB + AD + AC$

Reduce the following expression using K-map and implement it with universal logic.

$f = \Sigma m(4,5,7,8) + d(10,11)$

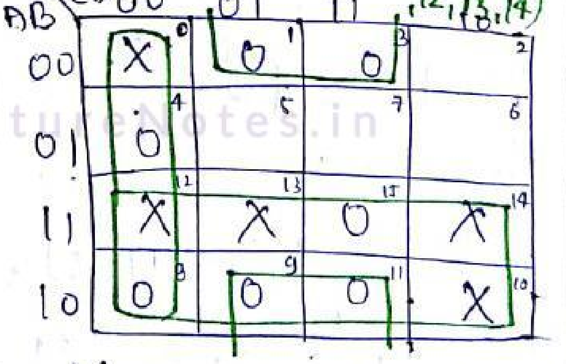


$f = \bar{A}B\bar{C} + A\bar{B}\bar{D} + \bar{A}BD$

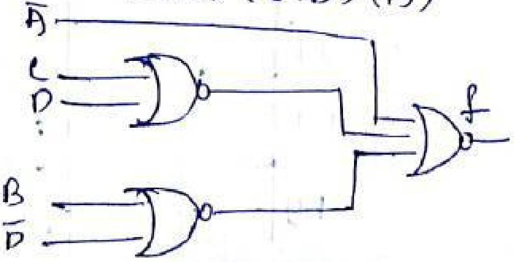
$= \overline{\overline{\bar{A}B\bar{C} + A\bar{B}\bar{D} + \bar{A}BD}}$

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

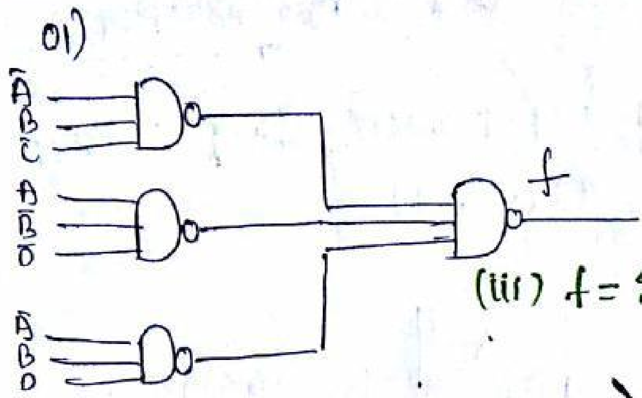
$f = \Pi M(1,3,4,8,9,11,15) + d(0,10,12,13,14)$



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



01)



(iii) $f = \Sigma m(1,6,10,11,12,13,15) + d(4,5,7,8,14)$

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

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

(iv) $f = \Sigma m(1,3,7,11,15) + d(0,2,5)$

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

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

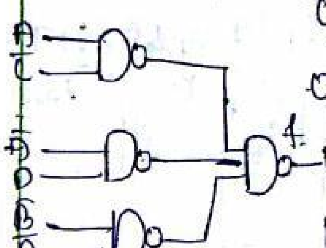
Reduce using maping and implement the real minimal expression in universal logic

$f = \Sigma m(0,1,2,3,5,7,8,9,10,12,13)$

Solution:-

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

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



Compared to NOR, NAND has less power consumption

$f = \Pi M(4,6,11,14,15)$

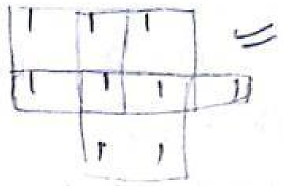
AB \ CD	00	01	11	10
00				
01	0			0
11			0	0
10			0	

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

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

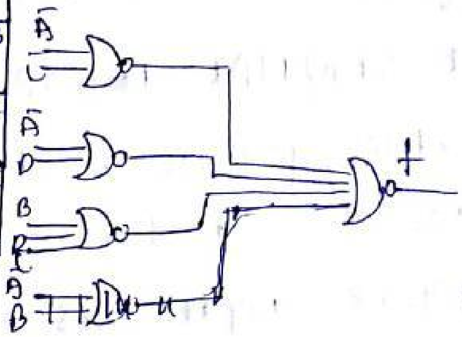
(ii) $f = \Pi M(2,8,9,10,11,12,14)$

$f = \Sigma m(0,1,3,4,5,6,7,13,15)$



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

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



(i) $f = \Sigma m(0,2,4,6,7,8,10,12,13,15)$

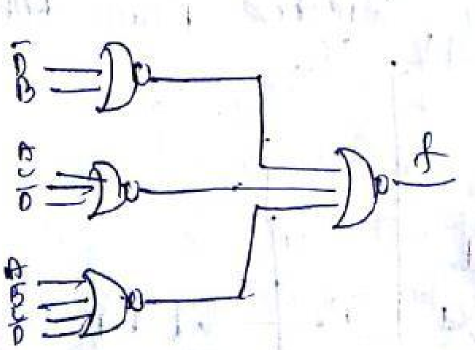
AB \ CD	00	01	11	10
00	1			1
01	1		1	1
11	1	1	1	
10	1			1

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

$f = \Pi M(1,3,5,9,11,14)$

AB \ CD	00	01	11	10
00		0	0	
01		0		
11				0
10		0	0	

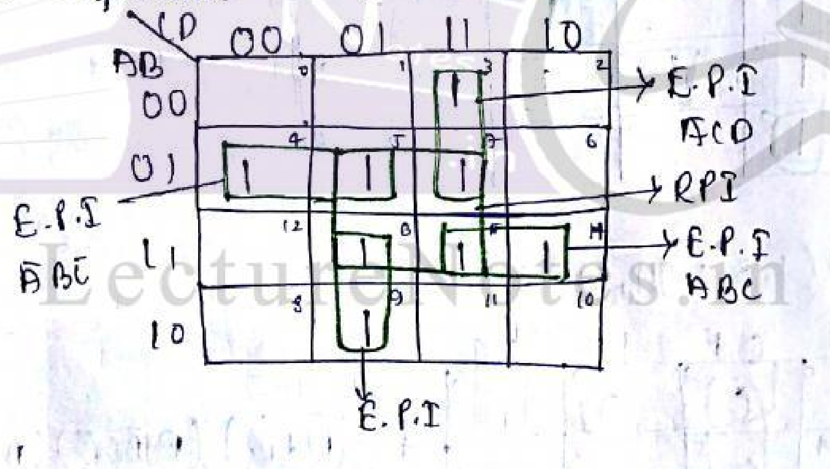
$f = (\bar{D}+B)(A+\bar{B}+\bar{C}+\bar{D})(A+C+\bar{D})$



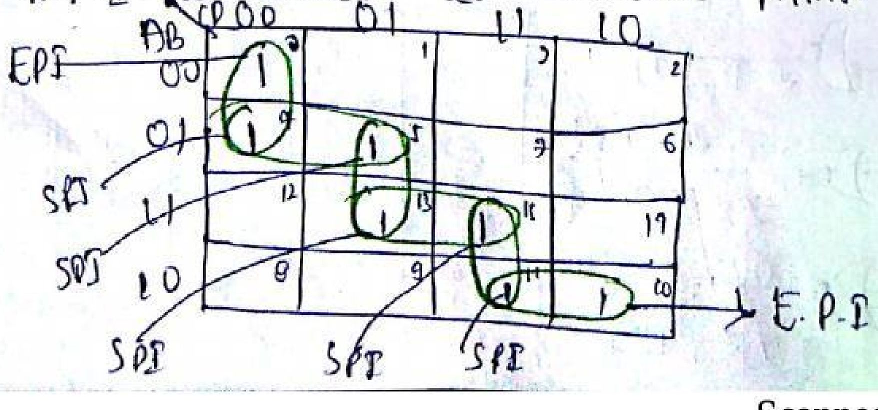
* **PRIME IMPLICANT:~** Each square or rectangle is made up of adjacent minterms called sub-cube.
→ Each sub cube is known as a prime implicant.
→

* **ESSENTIAL PRIME IMPLICANT:~ (EPI)**
→ The prime implicant which contains atleast one '1' which can't be covered by other prime implicants is known as Essential Prime Implicant

* **REDUNDANT PRIME IMPLICANT:- (RPI)**
→ The prime implicant whose each '1' is covered by other E.P.I is known as Redundant Prime Implicant



* **SELECTIVE PRIME IMPLICANT:~**
The prime implicant which is neither E.P.I nor R.P.I is known as Selective prime implicant



*FALSE PRIME IMPLICANTS:-

⇒ The maxterms are called False minterms.

⇒ The prime-implicants obtained by using max terms are known as False-Prime Implicants

*FALSE ESSENTIAL PRIME IMPLICANTS:-

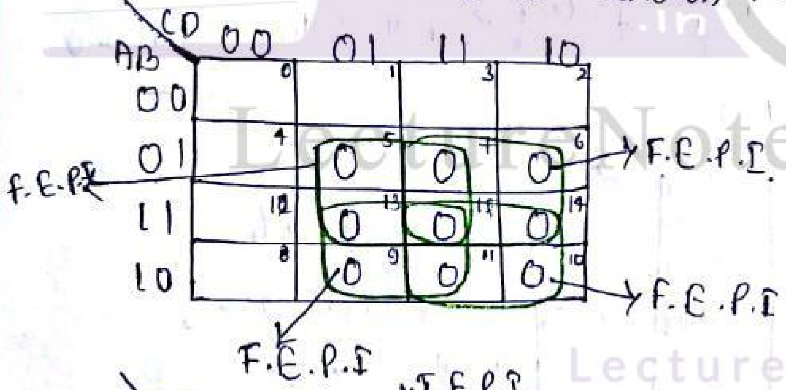
⇒ The False prime Implicant which contains atleast one zero which can't be covered by other F.P.I is known as F.E.P.I

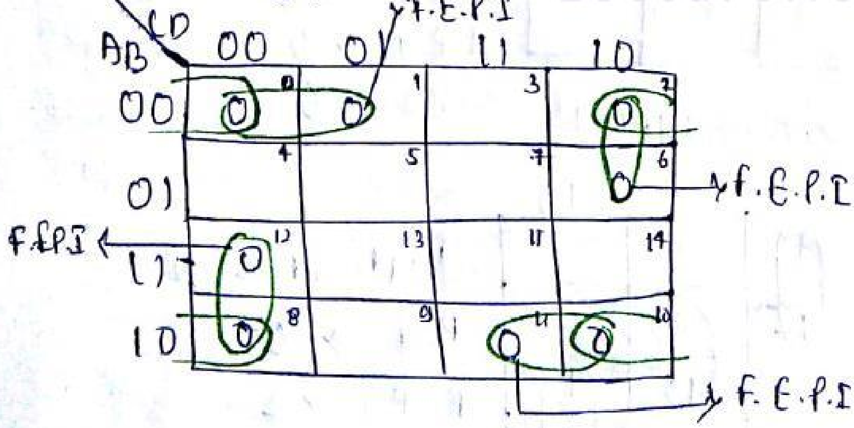
*FALSE REDUNDANT PRIME IMPLICANTS:-

⇒ The False Prime Implicant whose each zero is covered by other F.E.P.I is known as False Redundant Prime Implicant

*FALSE SELECTIVE PRIME IMPLICANT:-

⇒ The False Prime Implicant which is neither F.E.P.I nor F.R.P.I is known F.S.P.I

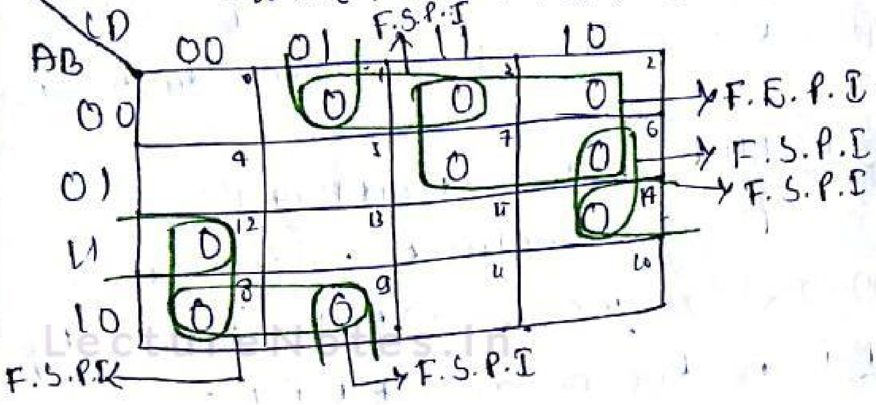




Calculate the F.P.I, F.E.P.I, F.R.P.I, F.S.P.I

$f(A,B,C,D) = \Sigma m(0,4,5,10,11,13,15)$

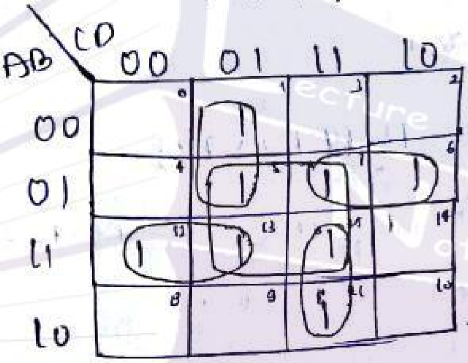
$= \Pi M(1,2,3,6,7,8,9,12,14)$



13/2/19

find the number of implicants, P.I, E.P.I, R.P.I, S.P.I for the following functions

(i) $F = \Sigma(1,5,6,7,11,12,13,15)$



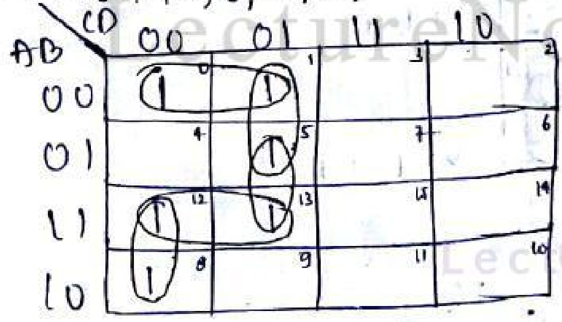
implicants → 08

Prime Implicants → 05

Essential Prime Implicant → 04

Selective prime Implicant → 0

(ii) $F = \Sigma(0,1,5,8,12,13)$



Implicant → 6

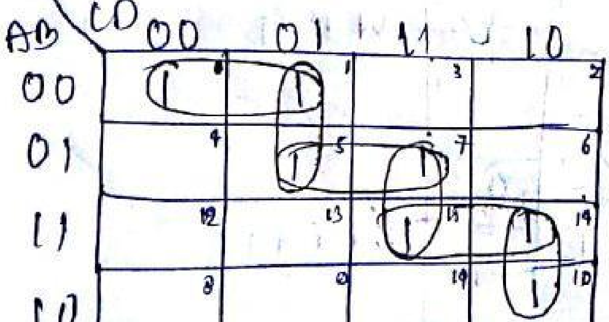
Prime implicant → 6

E.P.I → 0

R.P.I → 0

S.P.I → 6.

(iii) $F = \Sigma(0,1,5,7,10,14,15)$



Implicant → 7

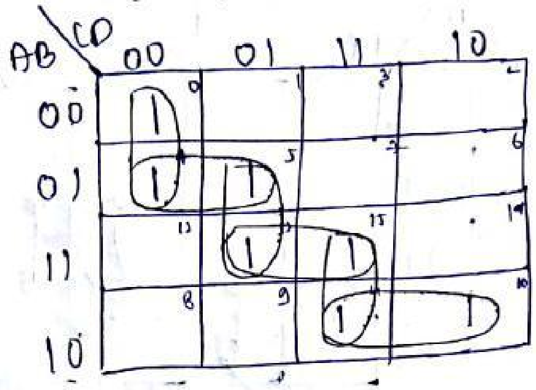
P.I → 6

E.P.I → 2

R.P.I → 0

S.P.I → 4

04) $f = \Sigma(0,4,5,10,11,13,15)$



Implicant → 07

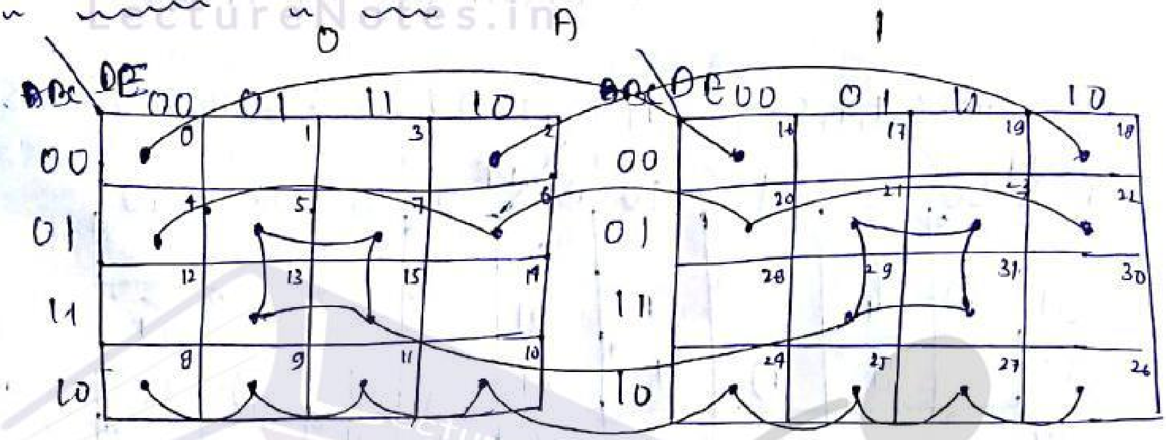
P.I → 6

E.P.I → 2

R.P.I → 0

S.P.I → 4.

5- Variable K- map :-



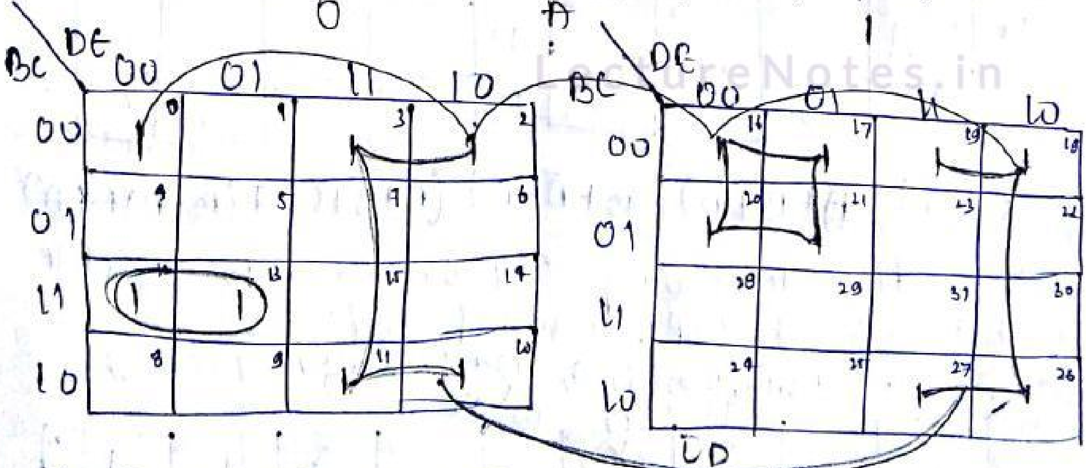
2 terms → 4 literals   4 → 3 literals.

8 terms → 2 literals   16 terms → 1 literal.

32 terms → 0 literals & 1 will be the answer.

Reduce the following expressions in S.O.P and P.O.S forms using mapping.

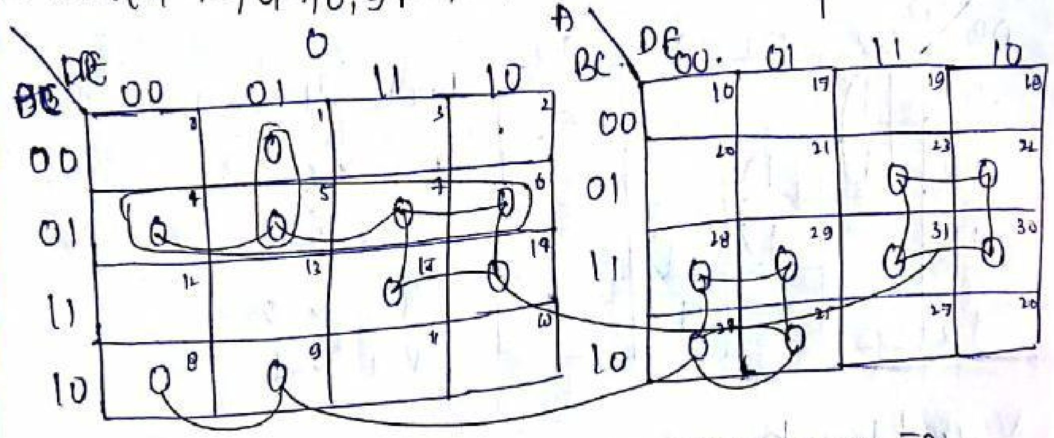
$f = \Sigma m(0,2,3,10,11,12,13,16,17,18,19,20,21,26,27)$



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

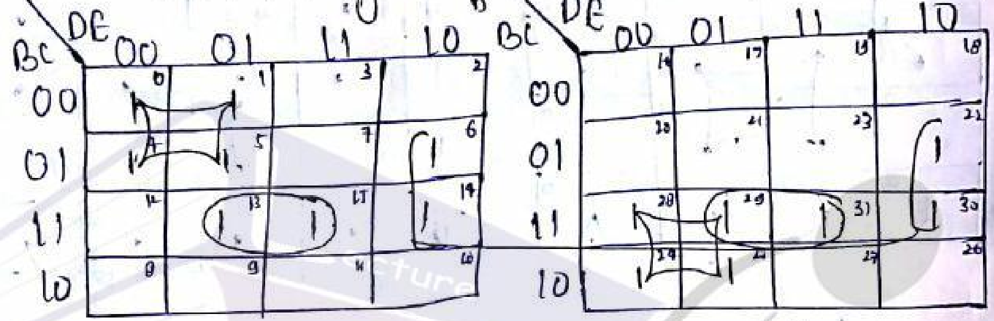
K-map → reduce product terms

$f = \Pi M(1,4,5,6,7,8,9,14,15,22,23,24,25,28,29,30,31)$



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

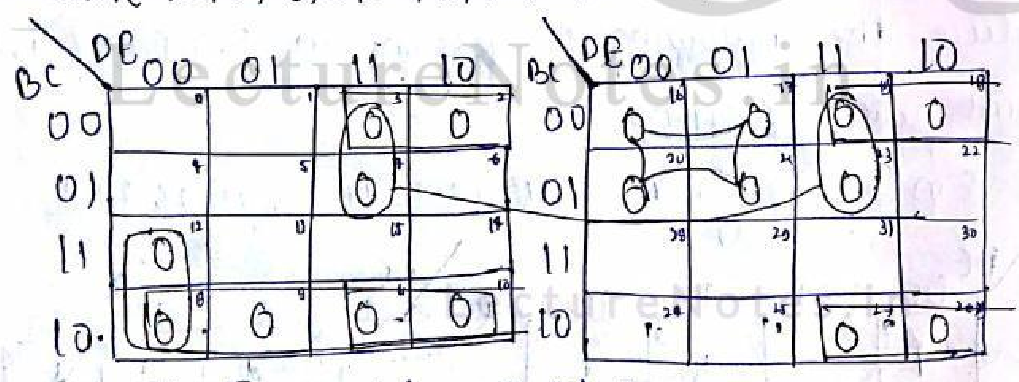
$f = \Sigma m(0,1,4,5,6,13,14,15,22,24,25,28,29,30,31)$ SOP & POS



$f = \bar{A}\bar{B}D + CD\bar{E} + BCE + AB\bar{D}$

P.O.S

$f = \Sigma m(2,3,7,8,9,10,11,12,16,17,18,19,20,21,23,26,27)$

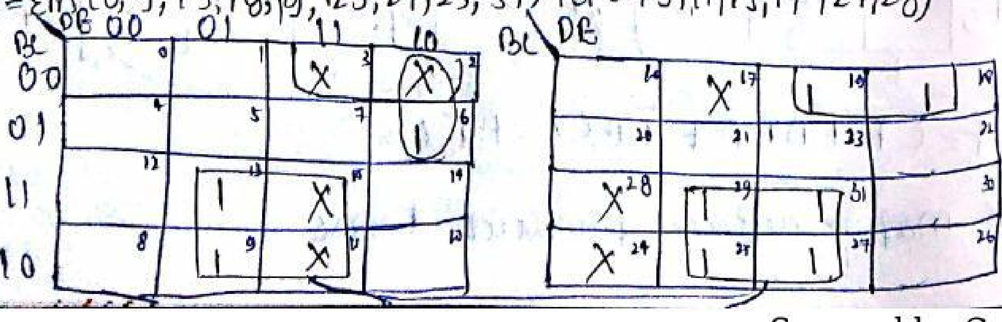


4/2/19

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

Reduce the following expressions into simplest possible P.O.S & S.O.P forms $f = \Sigma m($

$f = \Sigma m(6,9,13,18,19,25,27,29,31) + d(2,3,11,15,17,24,28)$



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

$P.O.S = \prod M(0,1,4,5,7,8,10,12,14,16,20,21,22,23,26,30) + d(2,3,11,15,17,24,28)$

A = 0

BC \ DE	00	01	11	10
00	0	0	X	X
01	0	0	0	
11	0		X	0
10	0		X	0

A = 1

BC \ DE	00	01	11	10
00	0	X		
01	0	0	0	0
11	X			0
10	X			0

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

Tabular Method:- (or) Quine Mc-cluskey Method:-

P.) Obtain the minimal expression for $f = \Sigma m(1,2,3,5,6,7,8,9,12,13,15)$

Step 01:-

Index 1 :- 1 → 0001
2 → 0010
8 → 1000

Index 2 :- 3 → 0011
5 → 0101
6 → 0110
9 → 1001
12 → 1100

Index 3 :- 7 → 0111
13 → 1101

Index 4 :- 15 → 1111

Step 02:-

(1,3) (2) ⇒ 00-1
(1,5) (4) ⇒ 0-01
(1,9) (8) ⇒ -001
(2,3) (1) ⇒ 001-
(2,6) (4) ⇒ 0-10
(8,9) (1) ⇒ 100-
(8,12) (4) ⇒ 1-00

(3,7) (4) ⇒ 0-11
(5,7) (2) ⇒ 01-1
(5,13) (8) ⇒ -101
(6,7) (1) ⇒ 011-
(9,13) (4) ⇒ 1-01
(12,13) (1) ⇒ 110-

(7,15) (8) ⇒ -111
(13,15) (2) ⇒ 11-1

Step 03:-

(1,3,5,7) (2,4) ⇒ 0--1
(1,5,9,13) (4,8) ⇒ --01
(2,3,6,7) (1,4) ⇒ 0-1-
(8,9,12,13) (1,4) ⇒ 1-0-

(5,7,13,15) (2,8) ⇒ -1-1

Step 04: Prime Implicant Chart

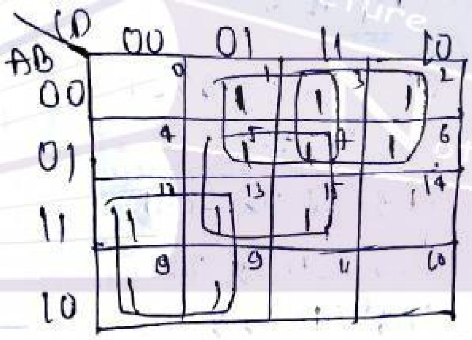
PI/minterm	1	2	3	5	6	7	8	9	12	13	15
P(1,3,5,7)	X		X	X		X					
Q(1,5,9,13)	X			X				X		X	
R(2,3,6,7)		⊗	X		⊗	X					
S(8,9,12,13)							⊗	X	⊗	X	
T(5,7,13,15)				X		X				X	⊗

f = R+S+T+P (or) f = R+S+T+Q (or) f = R+S+T+Q

f = 0-1- + 1-0- + -1-1 + 0--1 = $\bar{A}C + A\bar{C} + BD + \bar{C}D$

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

Checking:



f = (w,x,y,z) = Σm(0,1,5,7,8,10,14,15)

Step 01:

Index 0:
0000-0

Index 1:
0001-1
1000-8

Index 2:
0101-5
1010-10

Index 3:
0111-7
1110-14

Index 4:
1111-15

Step 02:

(1,0) (1) ⇒ 000-
(0,8) (8) ⇒ -000

(1,5) (4) ⇒ 0-01
(8,10) (2) ⇒ 10-0

(5,7) (2) ⇒ 01-1
(10,14) (4) ⇒ 1-10

(7,15) (8) ⇒ -111
(14,15) (1) ⇒ 111-

Step 03:

Step 03:- Prime Implicant Chart

PI / minterm	0	1	5	7	8	10	14	15
A (0,1)*	X	X						
B (0,8)	X				X			
C (1,5)		X	X		X	X		
D (8,10)*			X	X				
E (5,7)*			X					
F (10,14)				X		X	X	
G (7,15)				X				X
H (14,15)*							X	X

$f = A + D + E + H$

$= 000\_ + 10\_0 + 01\_1 + 111\_$

$= \bar{w}\bar{x}\bar{y} + w\bar{x}\bar{z} + \bar{w}xz + wxy$

$f = \Sigma m(6,7,8,9) + d(10,11,12,13,14,15)$

Step 01

Index 1

8 → 1000

Index 2

6 → 0110

9 → 1001

10 → 1010

12 → 1100

Index 3

7 → 0111

11 → 1011

13 → 1101

14 → 1100

Index 04

15 → 1111

Step 2

(8,9) (1) 100_

(8,10) (2) 10_0

(8,12) (4) 1_00

(6,7) (1) 011_

(6,14) (8) _110

(9,11) (2) 10_1

(9,13) (4) 1_01

(10,11) (1) 101_

(10,14) (4) 1_10

(12,13) (1) 110_

(12,14) (2) 11_0

(7,15) (8) _111

(11,15) (4) 1_11

(13,15) (2) 11_1

(14,15) (1) 111_

Step 03

Q (8,9,10,11) (1,2) 10__

R (8,9,12,13) (1,4) 1_0_

(8,10,12,14) (2,4) 1__0 ✓

S (6,7,14,15) (1,8) _11_

don't × (10,11,14,15) (1,4) 1_1_

× (12,13,14,15) (1,2) 11__

(9,11,13,15) (2,4) 1__1 ✓

Step 04

P (8,9,10,11,12,13,14,15)

(1,2,4) 1___

PI/minterm	6	7	8	9
P(8,9,10,11,12,13,14,15)			X	X
Q(8,9,10,11)	·		X	X
R(8,9,12,13)			X	X
* S(6,7,14,15)	X	X		

$f = P + S$

$1--- + -11- = A + BC$

Check K-map:

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

$f = A + BC$

find the minimal expression for $f = \Pi M(2,3,8,12,13)\, d(10,14)$ using Tabular method

Step 01

Index 01:-
0010 - 2 -
1000 - 8 -

Index 02:-
3 → 0011 -
10 → 1010 -
12 → 1100 -

Index 03:-
13 → 1101 -
14 → 1110 -

Step 02:-

(2,3)(1) 001_ A
(2,10)(8) -010 B
(8,10)(2) 10-0 -
(8,12)(4) 1-00 -

(10,14)(4) 1-10 -
(12,13)(1) 110- C
(12,14)(2) 11-0 -

Step 03:-

(8,10,12,14)(2,4) 1--0

Step 04:-

$f_1 = A.C.D$

$= (001-).(110-)(1--0)$

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

Prime Implicant chart	2	3	8	10	12	13	14
A(2,3)	X	X					
B(2,10)	X			X			
C(12,13)					X	X	
D(8,10,12,14)			X	X	X		X

Scanned by CamScanner

AB \ CD	00	01	11	10
00			0	0
01				
11	0	0		X
10	0			X

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

NAND- NAND IMPLEMENTATION:-

* Bubbled and gate is NOR Gate
* Bubbled or gate is NAND Gate

Step 01:-

→ for AND gate keep the bubble at the end.

→ for OR gate keep the bubble at the starting point of the gate.

Step 02:-

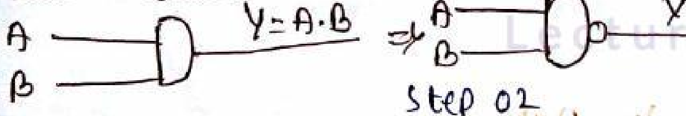
→ Wherever bubble was kept, keep a NOT Gate.

Step 03:-

→ Replace NOT gate with NAND Gate.

01. NOT GATE:-

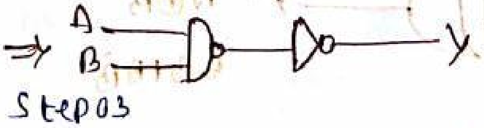
02. AND GATE :-



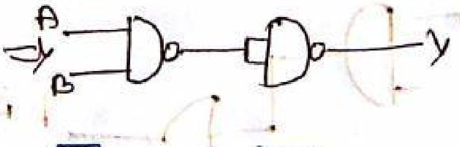
Step01

2 → NAND are required.

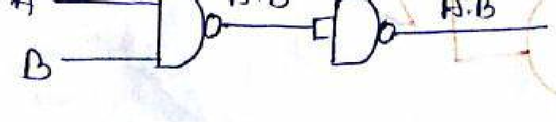
Step 02



Step03



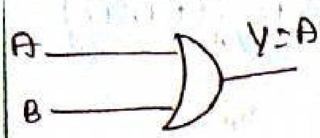
CHECK:-



$= A.B$ → AND

(03) OR Gate :-

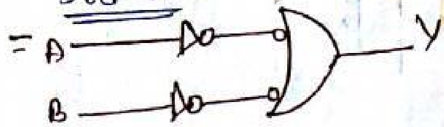
3-NAND Gate are required



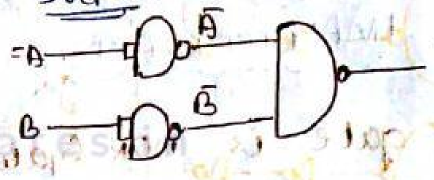
=

Step 01 :-

Step 02 :-



Step 03 :-



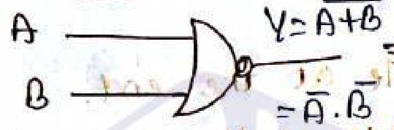
$Y = \overline{\bar{A} \cdot \bar{B}}$

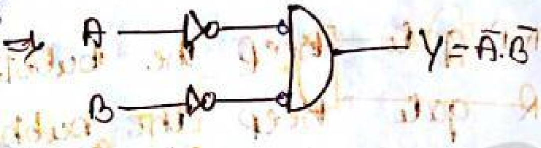
$= \bar{\bar{A}} + \bar{\bar{B}}$

$= A+B$

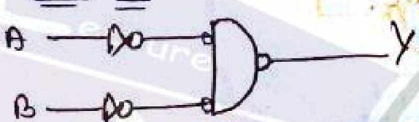
04) NOR Gate :-

4-NAND Gate are required

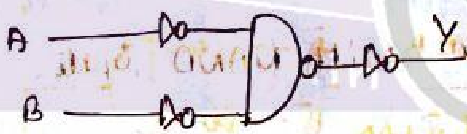




Step 01 :-

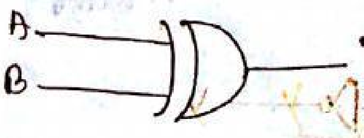


Step 02 :-



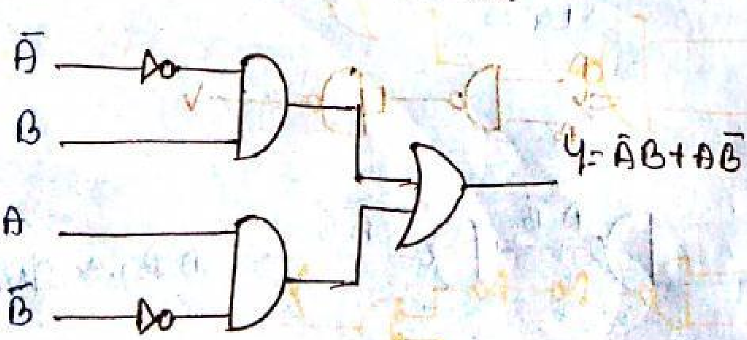
Step 03 :-

(05) EX-OR Gate :-

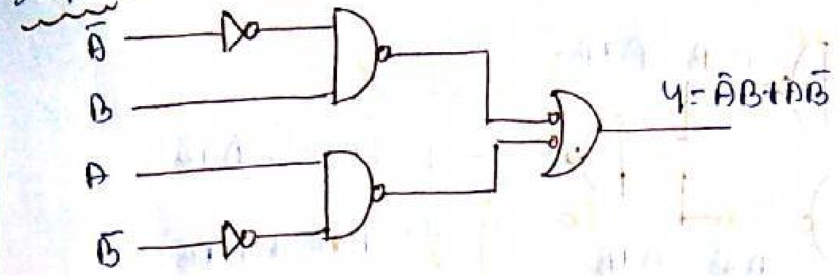


$Y = A \oplus B$

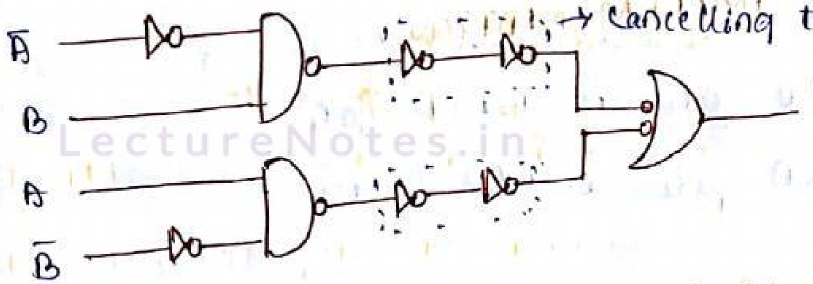
$= \bar{A}B + A\bar{B}$



Step 01:-



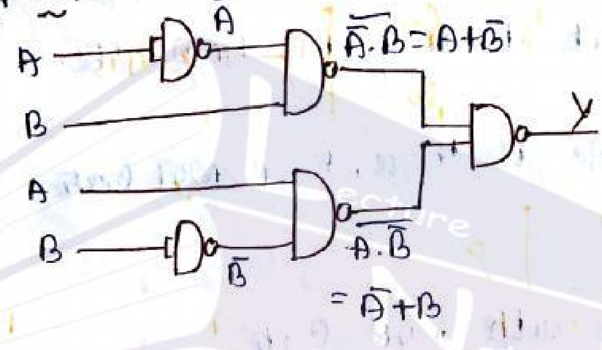
Step 02:-



→ Cancelling two NOT Gates $(\bar{\bar A} = A)$

5 NAND Gates are required.

Step 03:-

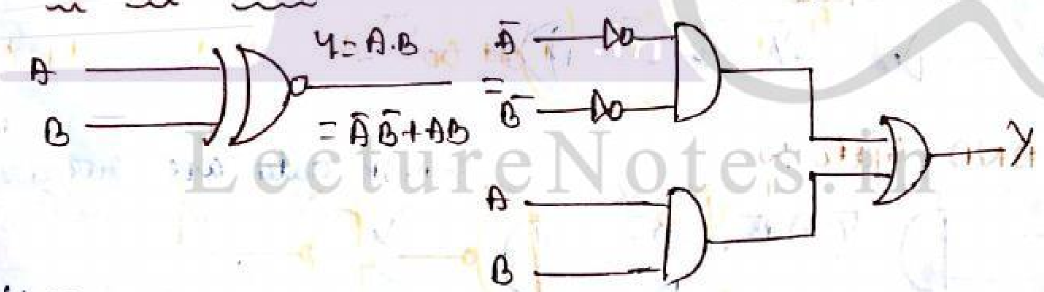


$Y = \overline{(A+\bar B)(\bar A + B)}$

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

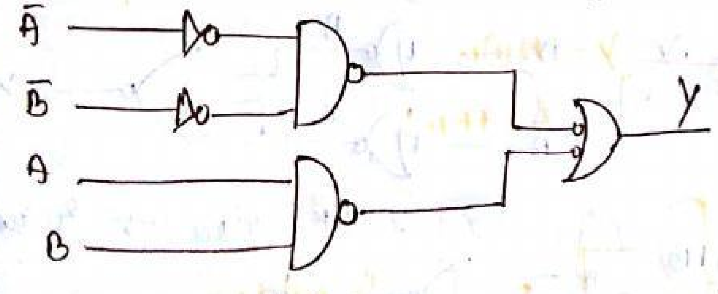
$= \bar A \cdot B + A \cdot \bar B$

(06) EX-NOR Gate :-

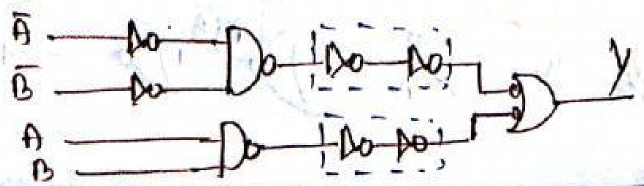


Step 01:- AND Gate → End bubble

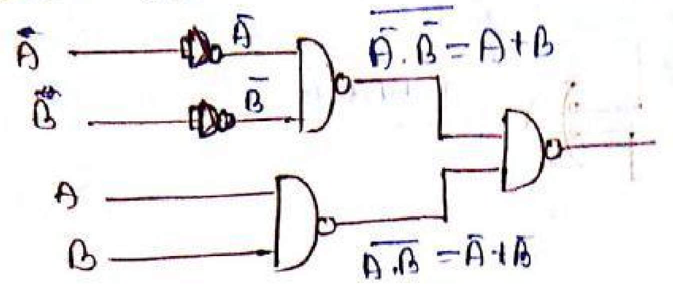
OR Gate → Starting point bubble.



Step 02:- Keeping NOT Gate near bubble.



Step 03:-



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

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

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

NOR- NOR IMPLEMENTATION:-

Bubbled OR gate is NAND Gate

Bubbled AND gate is NOR Gate.

Step 1:-

→ Keep the bubble at the end for OR Gate

→ keep the bubble at starting for AND Gate

Step 02:-

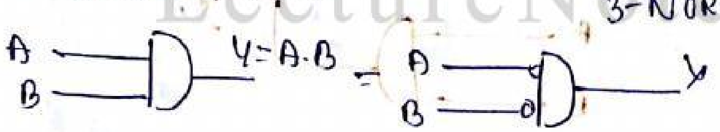
⇒ Wherever the bubble is there, keep NOT Gate

Step 03:-

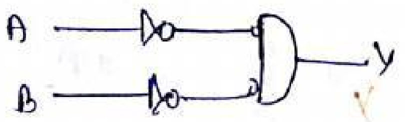
⇒ Replace NOT gate with NOR Gate

01. NOT Gate:-

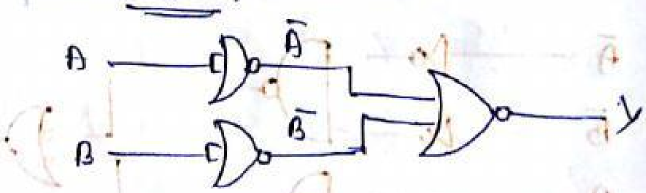
02. AND Gate:- 3-NOR gates are required.



Step 02:-

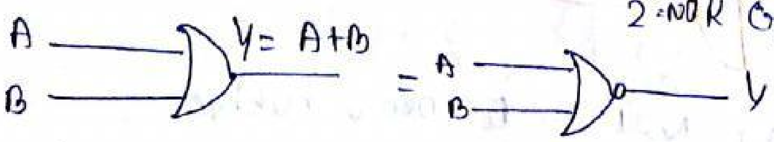


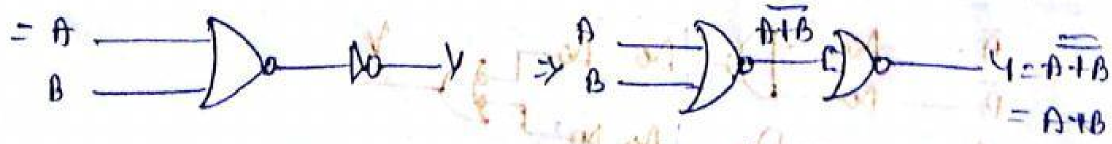
Step 03:-



2-NOR Gates are required.

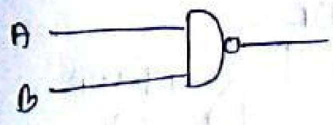
03. OR Gate:-





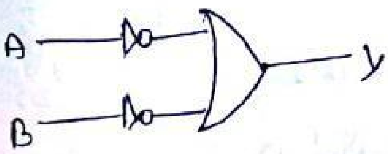
04. NAND Gate :-

4 - NOR gates are required

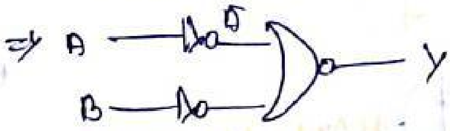


$Y = \overline{A \cdot B} = \bar{A} + \bar{B}$

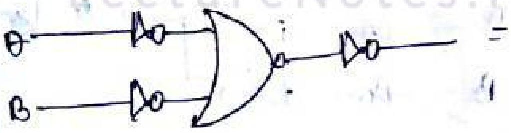
Step 01:-



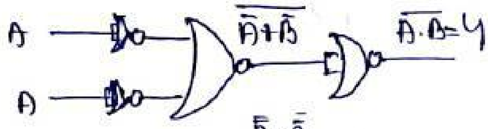
Step 07:-



Step 02:-



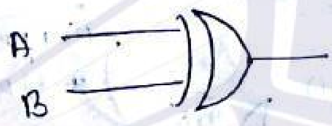
Step 03:-



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

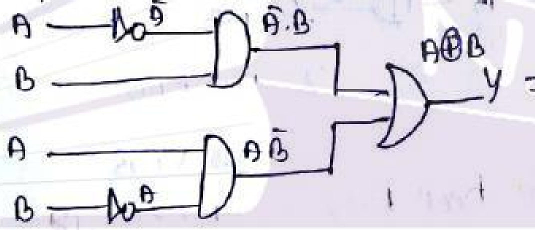
$= A \cdot B$

05. EX-OR Gate :-

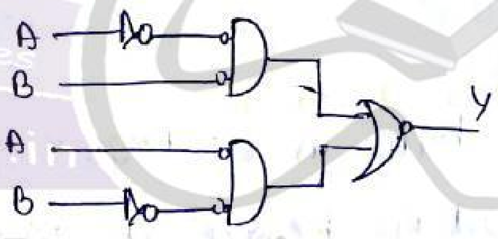


$Y = A \oplus B$

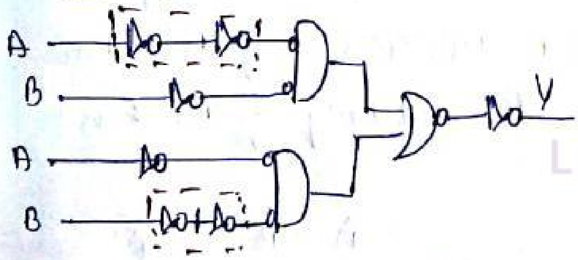
$= \bar{A}B + A\bar{B}$



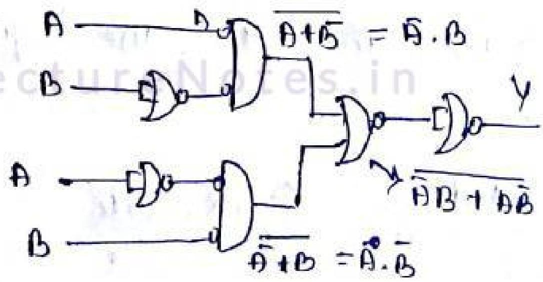
Step 01

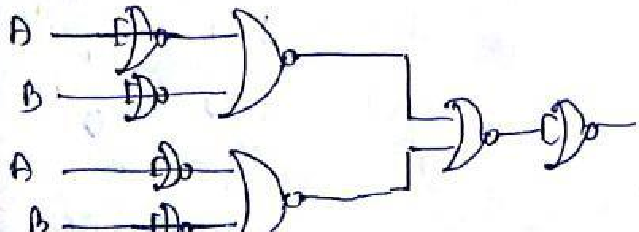


Step 02:-



Step 03:-

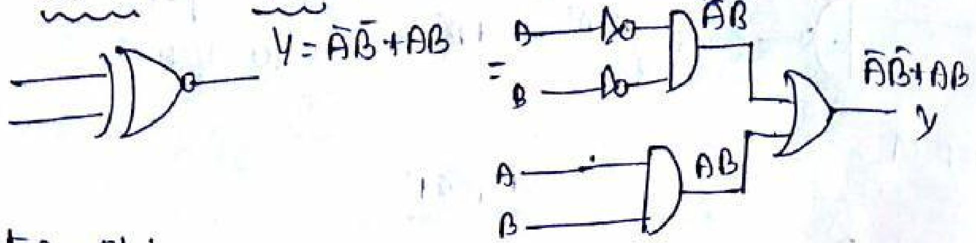




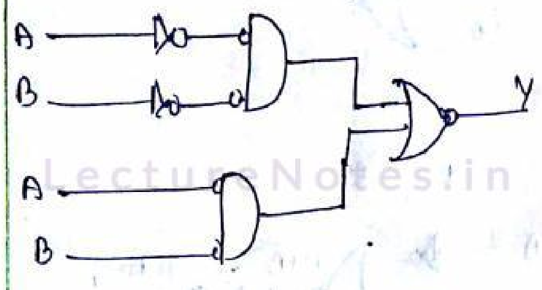
$Y = \overline{\overline{A\bar{B} + \bar{A}B}}$

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

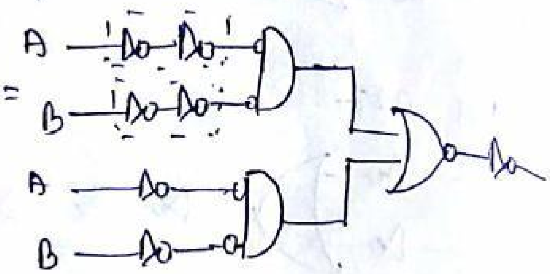
EX-NOR Gate :-



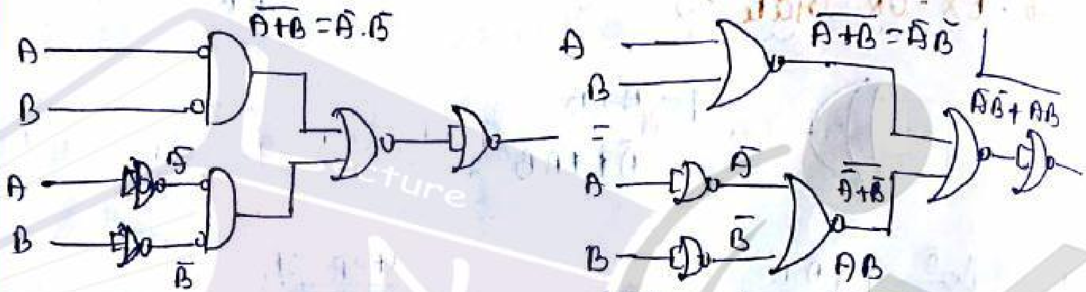
Step -01 :-



Step 02:-



Step -03 :-



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

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

A bulb in a stair case has two switches. One switch being at ground floor and other on first floor. The bubl can be turned ON and off by and one of the switches irrespective of the state of the other state. The Logic of switching of the bulb resembles

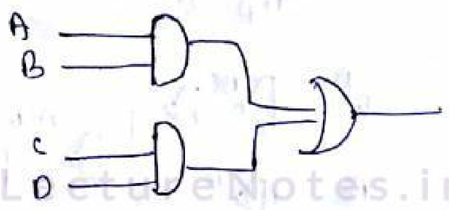
Solution:- X-OR Gate

A	B	XOR
0	0	0
0	1	1
1	1	1
1	0	0

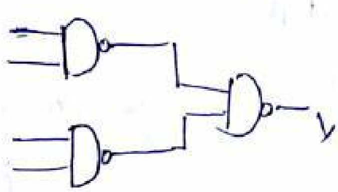
The Boolen function $Y = AB + CD$ is to be realised using 2 input NAND Gates. The minimum number of gates required is

Solution :- Draw the logic gates for the given expression

$Y = AB + CD \Rightarrow$



$\Rightarrow Y = \overline{\overline{AB + CD}} = \overline{\overline{AB} \cdot \overline{CD}}$



What is the minimum number of 2-input NAND Gates required to implement the following Boolean functions

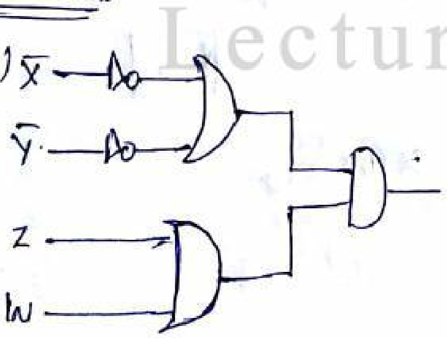
(a) $f = (\bar{X} + \bar{Y})(Z + W)$

(b) $Z = A\bar{B}C$

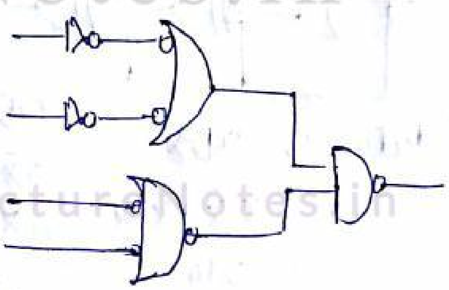
(c) $f = A + A\bar{B} + A\bar{B}C$

Solution

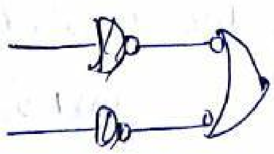
(a)



$\Rightarrow$



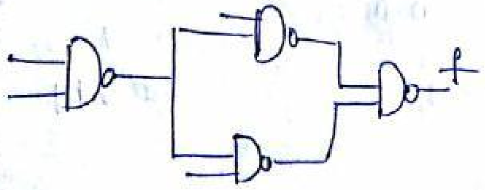
$\Rightarrow$



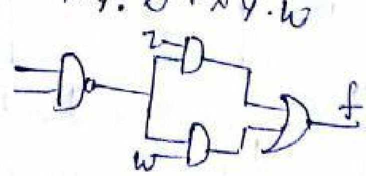
① $f = (\bar{X} + \bar{Y})(Z + W)$

$= \overline{XY}\,(Z + W)$

$= \overline{XY} \cdot Z + \overline{XY} \cdot W$

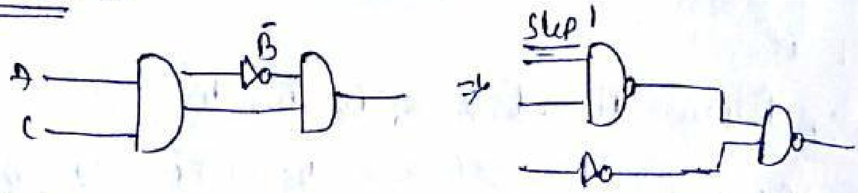


$\Leftarrow$

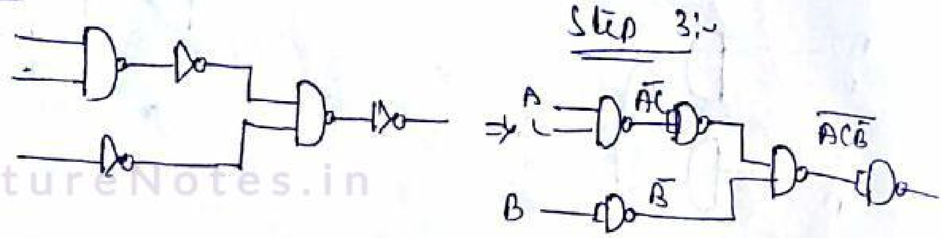


(b) $Z = A\bar{B}C$

Solution:



Step 2:

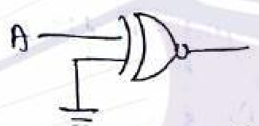


(i) $f = A + A\bar{B} + A\bar{B}C$

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

Zero NAND gates are required to realise

03) The o/p of the logic gate in the fig is

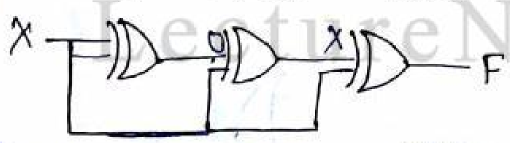


(a) 0
(b) 1
(c) $\bar{A}$
(d) A

A . 0 $\quad A \cdot B = \bar{A}\bar{B} + AB$

$f = A.0 + \bar{A}.1 \Rightarrow 0 + \bar{A} = \bar{A}$

04) for the circuit shown below, the o/p 'f' is given by



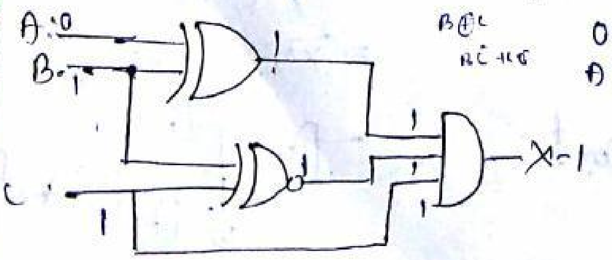
$\boxed{f=0}$

Solution:
$X \oplus X$
$X.\bar{X} + \bar{X}.X$
$0+0$
$=0$

$0 \oplus X$
$0\bar{X} + \bar{0}X$
$\bar{X} + X.1$
$= X$

$X \oplus X = 0$

05) for the logic circuit shown in fig, the required i/p conditions to make the o/p $x=1$ is



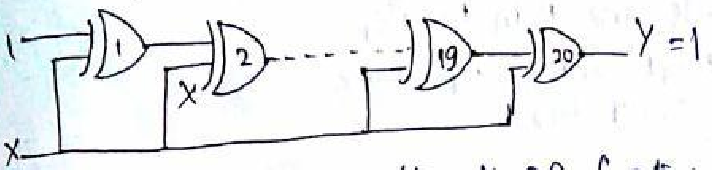
$B \oplus C$
$B\bar{C} + \bar{B}C$

0, 1, 1
A B C

(A) 1, 0, 1
(b) 0, 0, 1
(c) 1, 1, 1
~~(d)~~ 0, 1, 1

If the i/p to a digital circuit shown consisting of cascaded 20 X-OR Gates is X, the o/p Y is equal to

$1 \oplus X \Rightarrow 1\bar{X} + X\bar{1}$
$\bar{X}$



<u>Solution</u>:- O/P at 1^{st} X-OR Gate:-

$1 \oplus X \Rightarrow 1.\bar{X} + X.0 = \bar{X}$

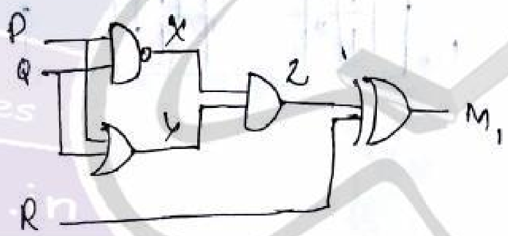
O/P at 2^{nd} X-OR Gate:-

$\bar{X} \oplus X \Rightarrow \bar{X}.\bar{X} + X.X \Rightarrow \bar{X} + X = 1.$

O/P at 19^{th} X-OR $\Rightarrow \bar{X}$

O/P at 20^{th} X-OR $\Rightarrow 1$

Which of the following boolean expression correctly represent the relation b/w PQR & M_1



<u>Solution</u>:- (a) $M_1 = (P \text{ OR } Q) \text{ XOR } R$

(b) $M_1 = (P \text{ AND } Q) \text{ XOR } R$

(c) $M_1 = (P \text{ NOR } Q) \text{ XOR } R$

~~(d)~~ $M_1 = (P \text{ XOR } Q) \text{ XOR } R$

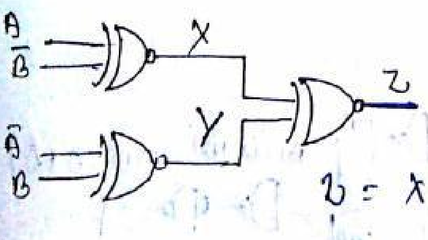
$X = \overline{P.Q}$; $Y = P+Q$

$Z = X.Y = \overline{P.Q}.(P+Q) \Rightarrow (\bar{P}+\bar{Q})(P+Q) = (\underbrace{(\bar{P}+\bar{Q})P}_{0})(\bar{P}Q+\bar{Q}P)$

$= (P \text{ XOR } Q)$

$M_1 = Z \text{ XOR } R \Rightarrow (P \text{ XOR } Q) \text{ XOR } R$

The o/p of the circuit show is equal to



$A \odot \bar{B} \Rightarrow \bar{A}.\bar{B} + B.\bar{\bar{B}} \Rightarrow \bar{A}\bar{B} + AB \Rightarrow A \oplus B$

$\bar{A} \odot B \Rightarrow \bar{A}.\bar{B} + BA \Rightarrow A \oplus B$

$Z = X \odot Y \Rightarrow (A \oplus B) \odot (A \oplus B)$

$\Rightarrow (A \oplus B)(A \oplus B) + \overline{A \oplus B}\,\overline{(A \oplus B)}$

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

Implement the following functions using NAND gates

(b) $f(A,B,C,D) = \Sigma m(0,3,4,8,\ldots)$

(a) $f = \overline{XY + XZ}$

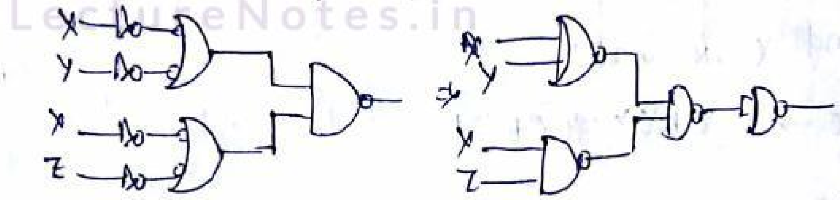
(c) $f = (CD+\bar{D})(AB)$

(d) $f = \bar{C}\bar{D} + \bar{B}CD + ACD$

} assume both true & complemented i/p are available

(a) $\Rightarrow \overline{XY+XZ} \Rightarrow \overline{XY}\cdot\overline{XZ} \Rightarrow (\bar{X}+\bar{Y})\cdot(\bar{X}+\bar{Z})$

4-NAND Gates required.

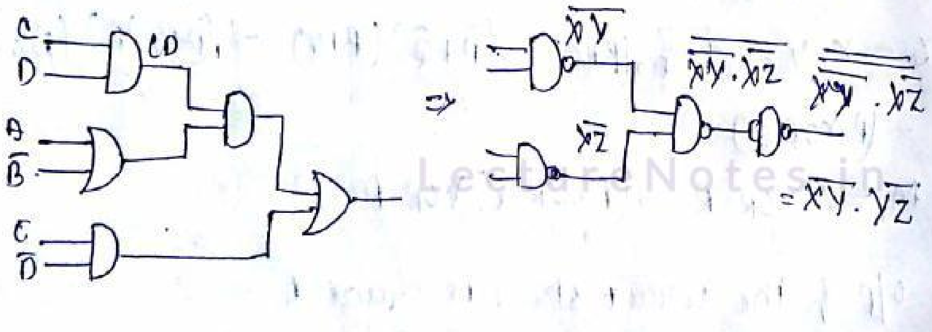


(b) $f(A,B,C,D) = \Sigma m(0,3,4,8,11,12,15)$

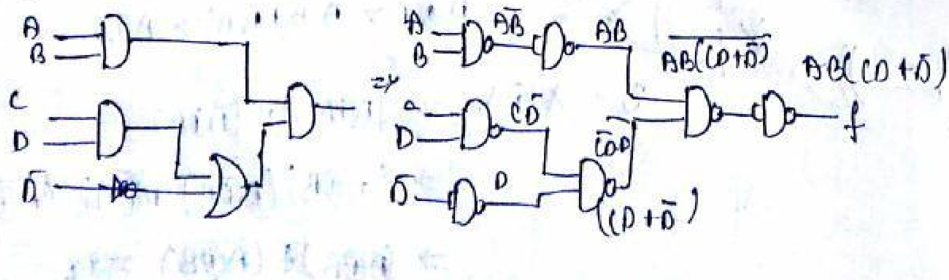
AB \ CD	00	01	11	10
00	1 ^{0}	^{1}	1 ^{3}	^{2}
01	1 ^{4}	^{5}	^{7}	^{6}
11	1 ^{12}	^{13}	1 ^{15}	^{14}
10	1 ^{8}	^{9}	1 ^{11}	^{10}

$f = \bar{C}\bar{D} + ACD + \bar{B}CD$

(d) $f = \bar{C}\bar{D} + \bar{B}CD + ACD \Rightarrow \bar{C}\bar{D} + CD(A+\bar{B})$



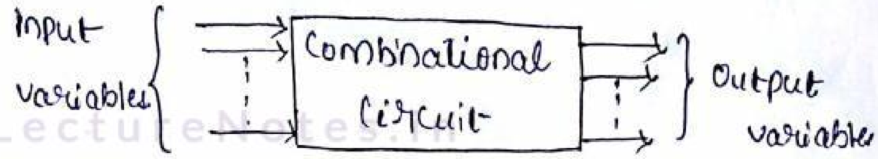
(c) $f = (CD+\bar{D})(AB)$



03. Combinational Circuits.

⇒ There are 2 types of circuits ① Sequential circuit ② Combina

⇒ In a digital circuit if the o/p at any instant of time depends only on the present i/p, then the circuit is said to be a combinational circuit.



Ex:- Adders, Subtracters, Multipliers, Multiplexer, Encoder, Decoders.

* Analysis and design of Combinational Circuits:-

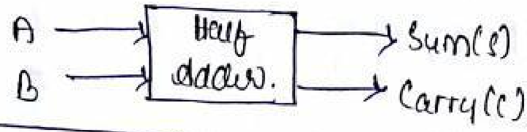
⇒ Identify the number of input and output variables

⇒ Express the output function interms of input variables

⇒ Truth table and logic diagram.

⇒ An Adder is a combinational circuit, which performs the addition of 2 bits [Half adder] and addition of 3 bits [full adder]

* Half adder:-



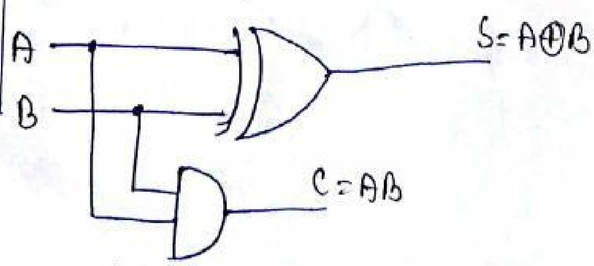
Inputs		Outputs	
A	B	C	D
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

(a) Truth Table.

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

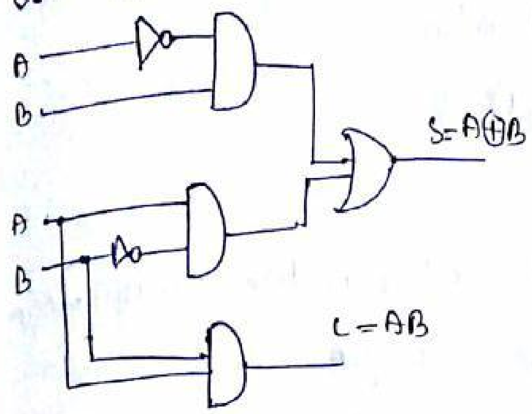
$\boxed{S = A \oplus B}$

$\boxed{C = AB}$



(b) Logic Diagram.

AOI Logic :-



$S = A\oplus B$

$S = \bar A B + A\bar B$

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

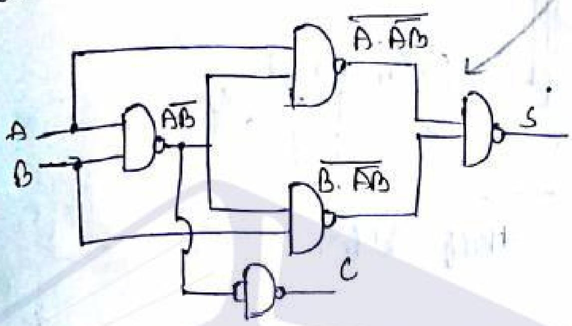
$= A(\bar A+\bar B) + B(\bar A+\bar B)$

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

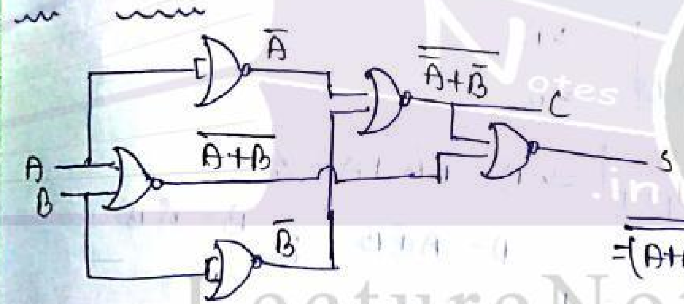
$= \overline{\overline{A\cdot\overline{AB} + B\cdot\overline{AB}}}$

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

* NAND Logic :-



NOR Logic :-



$S = \bar A B + A\bar B$

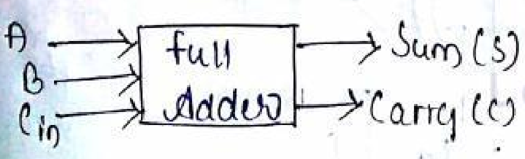
$= \bar A B + A\bar B + A\cdot\bar A + B\cdot\bar B$

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

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

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

Full Adder :-



$S = \bar A\bar B C_{in} + \bar A B\bar C_{in} + A\bar B\bar C_{in} + ABC_{in}$

$= \bar A[\bar B C_{in} + B\bar C_{in}] + A[\overline{\bar B\bar C_{in} \oplus BC_{in}}]$

$= \bar A(B\oplus C_{in}) + A(B\oplus C_{in})$

$\boxed{S = A\oplus B\oplus C_{in}}$ ⇒ Expression for sum

Inputs			Outputs	
A	B	C_{in}	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

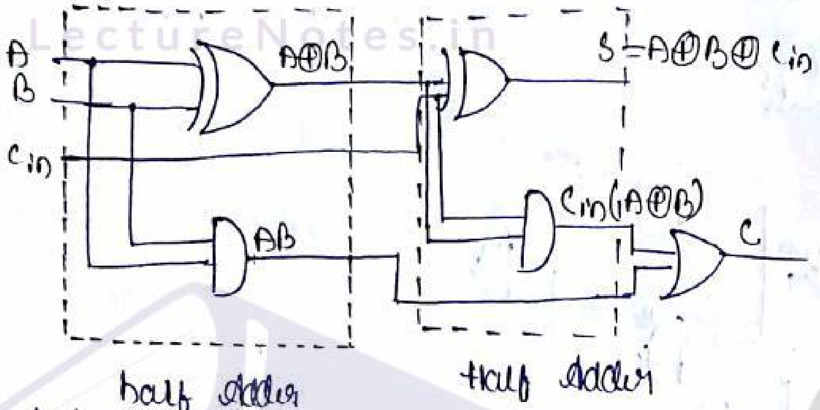
Expression for complementary :-

$$C = \bar{A}BC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in}$$

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

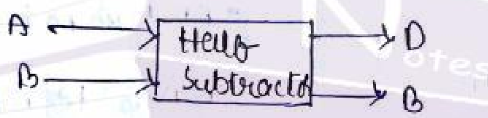
$$= \boxed{C = C_{in}(A \oplus B) + AB}$$

* Show that one full adder is the combination 2 half adder plus one OR gate



20/2/19 → Wednesday.

Half Subtractor :-

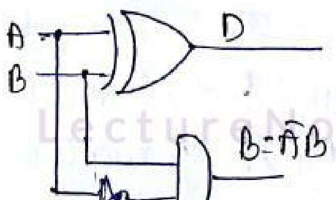


Inputs		Output	
A	B	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

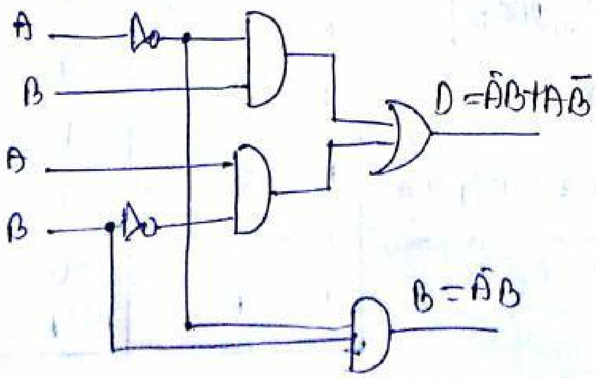
$\Rightarrow D = A\bar{B} + \bar{A}B$;

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

Logic Diagram :-

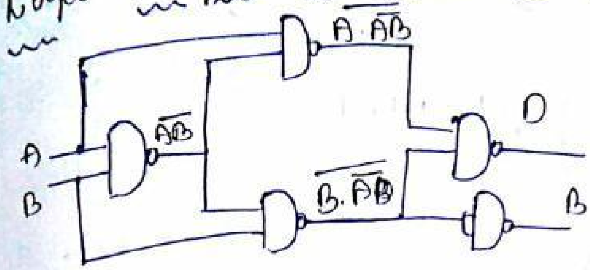


Logic Diagram in AOI Logic :-



Scanned by CamScanner

Logic Diagram in NAND Logic:



$D = \bar A B + A\bar B + A\cdot\bar A + B\bar B$

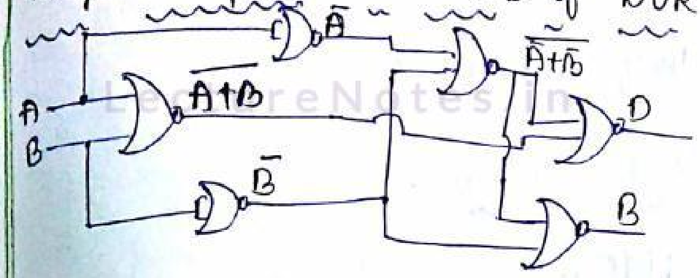
$= B(\bar A+\bar B) + A(\bar A+\bar B)$

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

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

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

Logic Diagram in terms of NOR Logic:



$D = A\bar B + \bar A B + A\cdot\bar A + B\cdot\bar B$

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

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

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

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

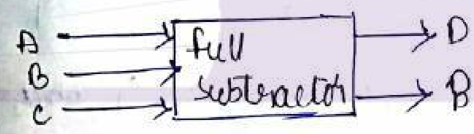
$B = \bar A B + A\cdot\bar A$

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

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

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

full subtractor:



Inputs			Outputs	
A	B	C	D	B
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$D = \bar A\bar B C + \bar A B\bar C + A\bar B\bar C + ABC$

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

$= \bar A(B\oplus C) + A(B\odot C)$

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

$\boxed{D = A\oplus B\oplus C}$

$B = \bar A\bar B C + \bar A B\bar C + \bar A BC + ABC$

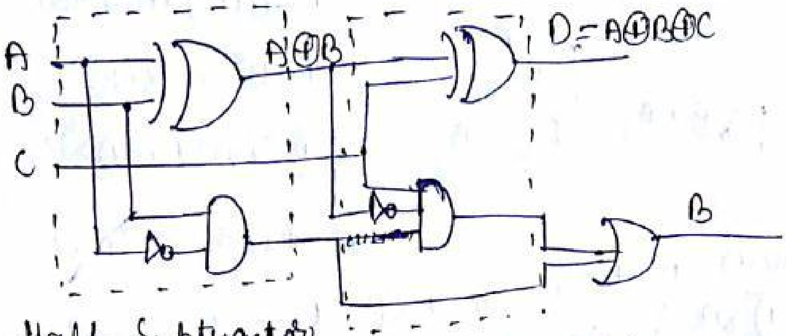
$= \bar A(\bar B C + B\bar C + BC) + ABC$

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

$= C(A\odot B) + \bar A B$

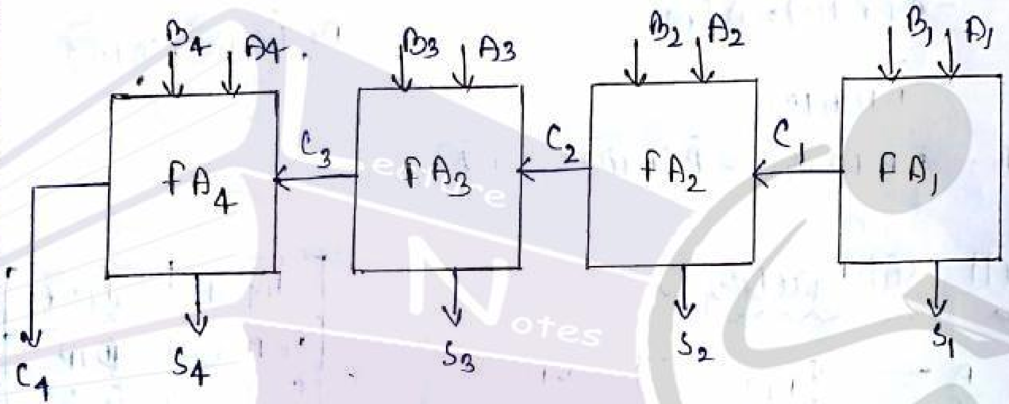
$\boxed{B = C(\overline{A\oplus B}) + \bar A B}$

Logic Diagram for full Subtractor:~



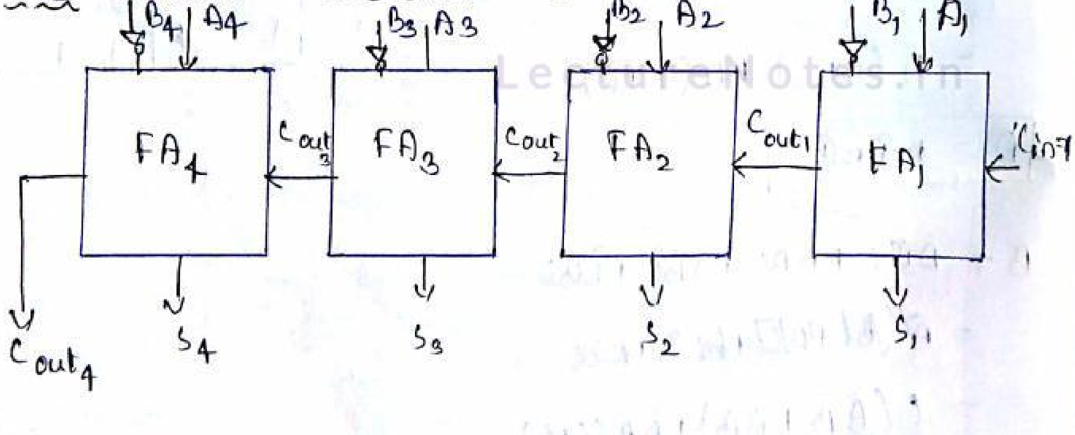
Applications of Full Adder:~

* Binary Parallel Adder:~



Disadvantage of the circuit is S_3 & S_4 depends on C_1 which results in the delaying of o/p.

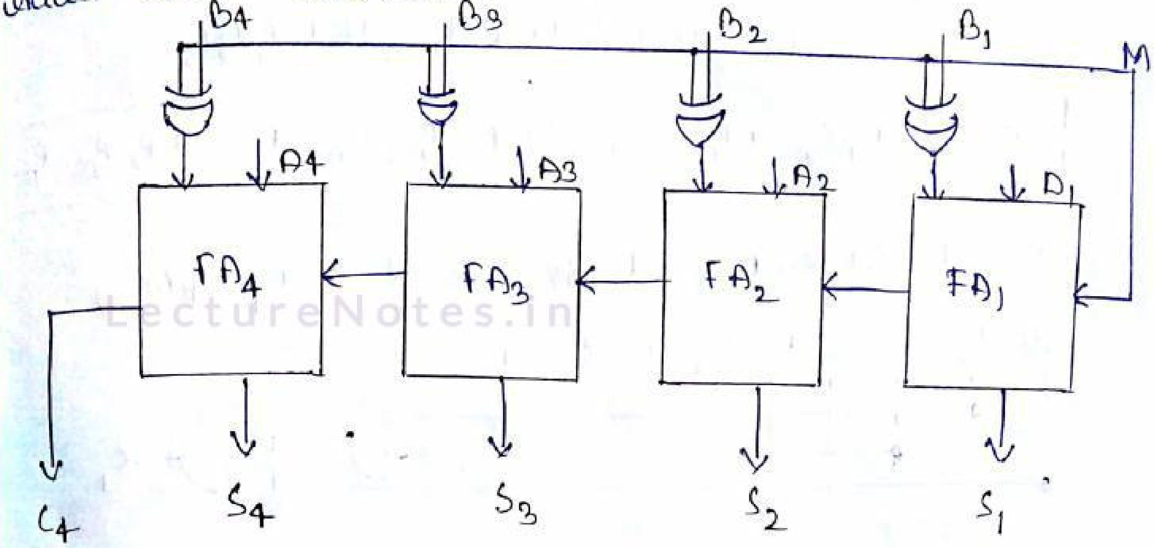
Binary Parallel Subtractor:~ [2's complement method]



Nor gates are used to calculate complement as we are doing 2's complement subtraction when we add $C_{in}=1$ we will find 2's complement

Binary parallel Adder and Subtractor:~

⇒ Out of the 7-logic gates X-OR acts as both adder and Subtractor



when $M=0$ ⇒ acts as Adder

when $M=1$ ⇒ acts as Subtractor

XOR gate output

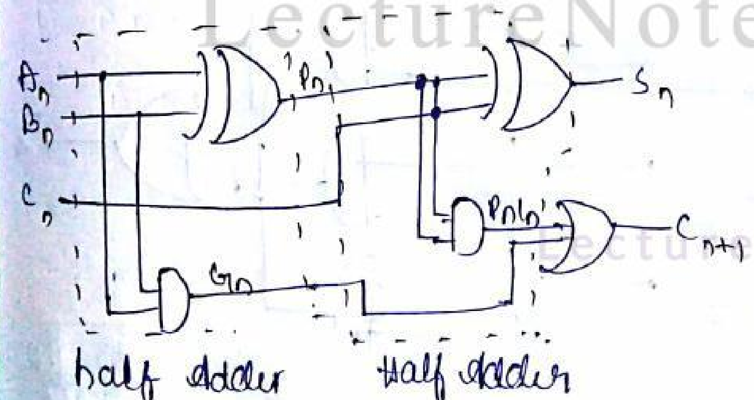
$M=0 \Rightarrow 0 \oplus B = 0\bar{B} + 1.B = B$

$M=1 \Rightarrow 1 \oplus B = 0.B + 1.\bar{B} = \bar{B}$

⇒ X-OR has more applications when compared to X-NOR gate.

20/2/19

Look Ahead Carry Adder:~



$P_n = A_n \oplus B_n$

$S_n = P_n \oplus C_n$

$C_{n+1} = G_n + P_n C_n$; when $n=0 \Rightarrow C_1 = G_0 + P_0 C_0$

when $n=1 \Rightarrow C_2 = G_1 + P_1 C_1 \Rightarrow G_1 + P_1(G_0 + P_0 C_0) = G_1 + P_1 G_0 + P_1 P_0 C_0$

when $n=2 \Rightarrow C_3 = G_2 + P_2 C_2 = G_2 + P_2(G_1 + P_1 G_0 + P_1 P_0 C_0) = G_2 + P_2 G_1 + P_2 P_1 G_0$

$n=2 \Rightarrow C_3 = G_2 + P_2 C_2 = G_2 + P_2(G_1 + P_1 G_0 + P_1 P_0 C_0)$

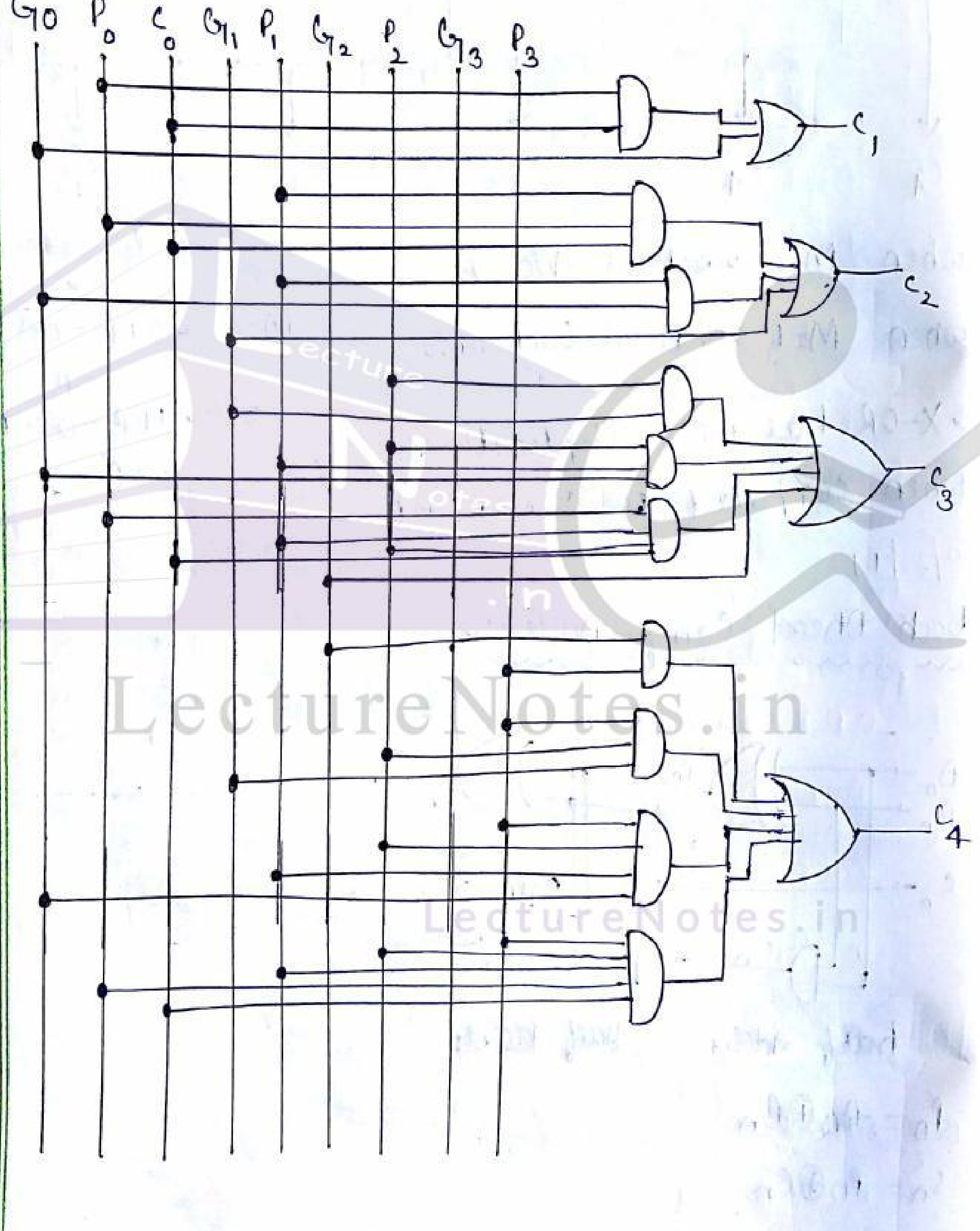
$= G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$

$n=3 \Rightarrow C_4 = G_3 + P_3 C_3$

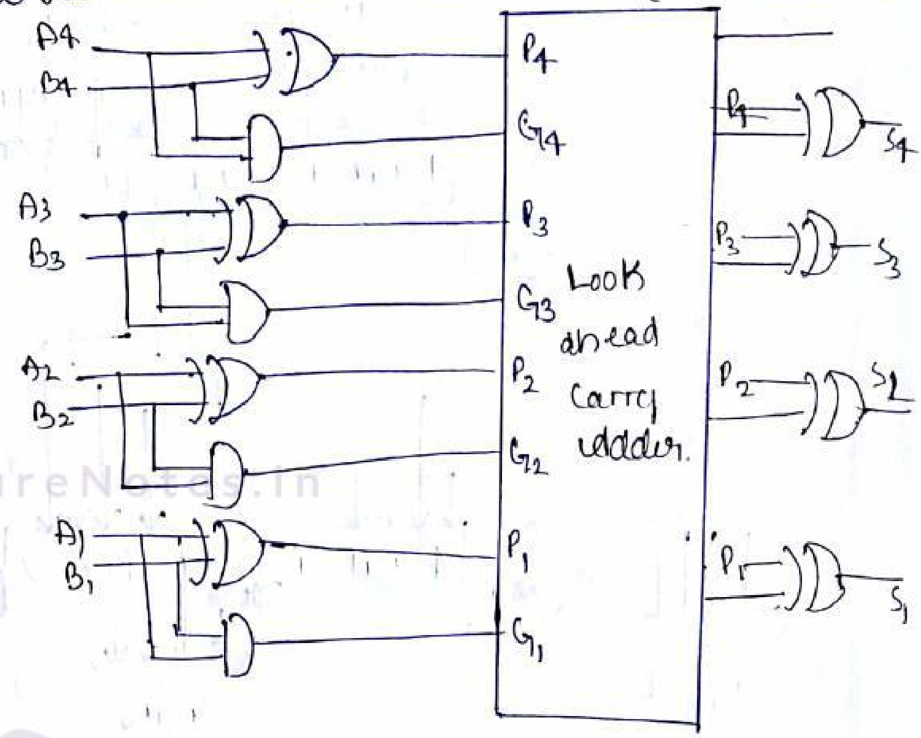
$= G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0$

$$\boxed{C_{n+1} = G_n + P_n G_{n-1} + P_n P_{n-1} G_{n-2} + P_n P_{n-1} P_{n-2} G_{n-3} + \dots + P_n P_{n-1} P_{n-2} \cdots P_0 C_0}$$

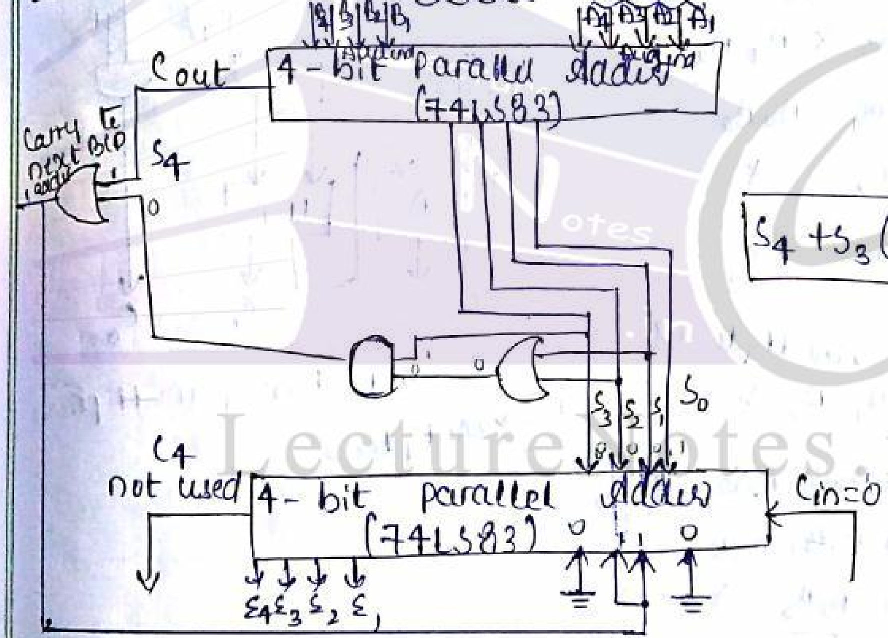
Internal circuit of Look Ahead carry Adder.



Block Diagram of Look Ahead Carry Generator



B.C.D Adder (or) Decimal Adder



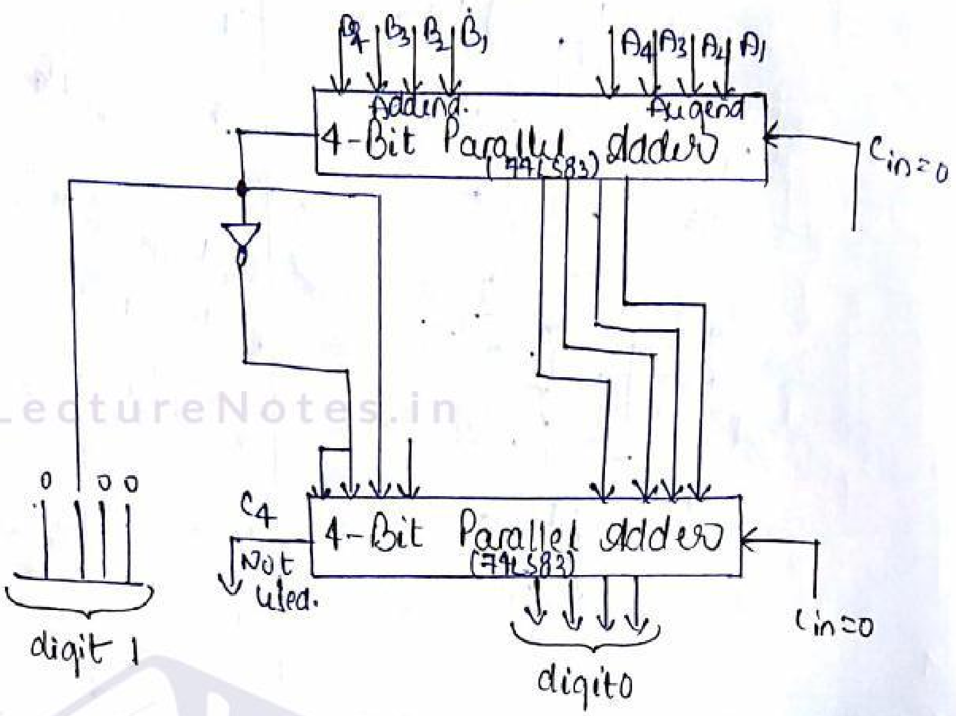
$S_4 + S_3(S_2+S_1)$

9 → 1001
8 → 1000
1 0001
0110
0111

Excess-3 Adder:-

if carry = 1 → add 3

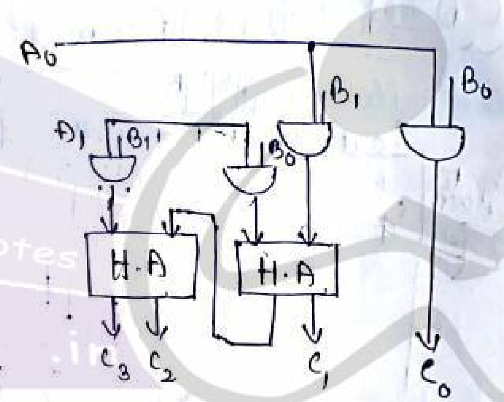
if carry = 0 → add 13



Binary Multiplier:-

⇒ $B_1\ B_0 \times A_1\ A_0$

		A_0B_1	A_0B_0
	A_1B_1	A_1B_0	x
C_3	C_2	C_1	C_0

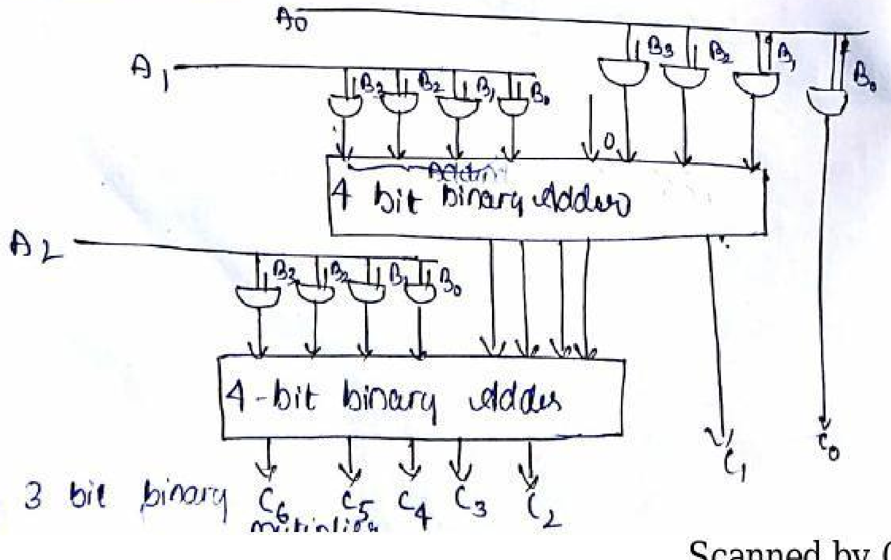


2-bit binary multiplier

⇒ $B_3\ B_2\ B_1\ B_0 \times A_2\ A_1\ A_0$ → Multiplier

→ Multiplicand

			A_0B_3	A_0B_2	A_0B_1	A_0B_0
		A_1B_3	A_1B_2	A_1B_1	A_1B_0	x
	A_2B_3	A_2B_2	A_2B_1	A_2B_0	x	x
C_6	C_5	C_4	C_3	C_2	C_1	C_0



4 bit by 3 bit binary multiplier

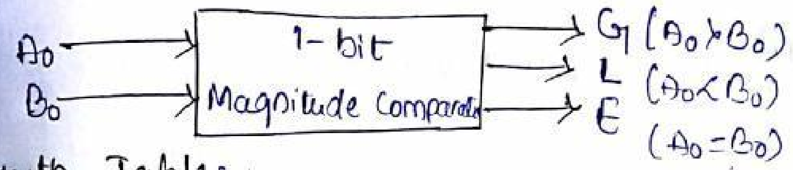
J × K AND gates are required ; K → Multiplicand

(J−1) K bit adders ; J → Multiplier

product term J+K bits

Magnitude Comparator:-

⇒ A magnitude comparator is a ~~mag~~ combinational circuit which is used to compare the magnitude of numbers represented in Binary.



Truth Table:-

A_0	B_0	G	L	E
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

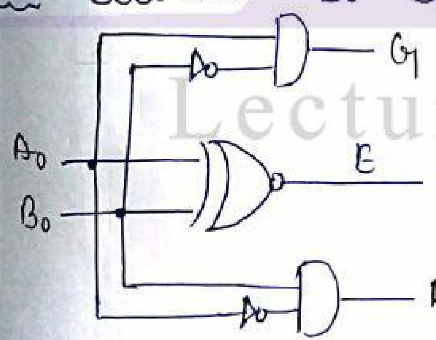
$\boxed{G \Rightarrow A_0 \bar{B_0}}$

$\boxed{L \Rightarrow \bar{A_0} B_0}$

$E \Rightarrow \bar{A_0}\bar{B_0} + A_0 B_0 \Rightarrow A_0 \odot B_0$

$\boxed{E \Rightarrow A_0 \odot B_0}$

Logic Diagram of Magnitude Comparator:-



* 2-bit Magnitude Comparator:-

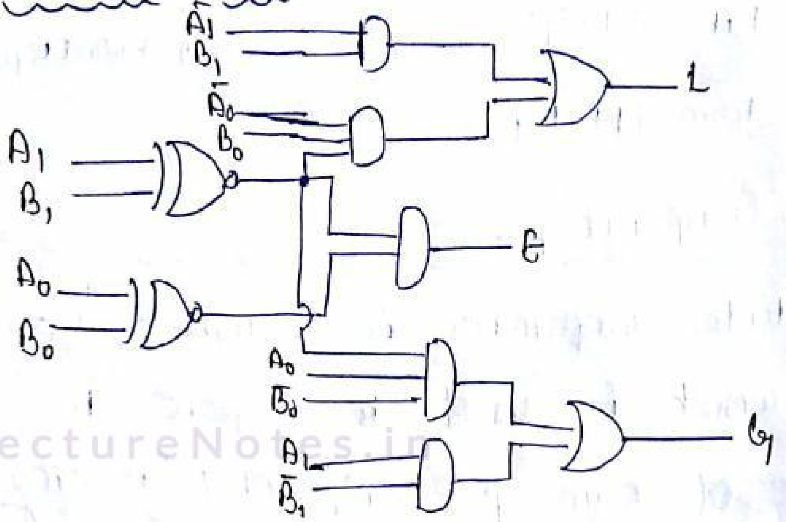
Equations are as follows:-

$G = A_1\bar{B_1} + (A_1 \odot B_1) A_0 \bar{B_0}$

$L = \bar{A_1} B_1 + (A_1 \odot B_1) \bar{A_0} B_0$

$E = (A_1 \odot B_1) \cdot (A_0 \odot B_0)$

Logic Diagram for 2-bit Magnitude Comparator



4-bit Magnitude Comparator:-

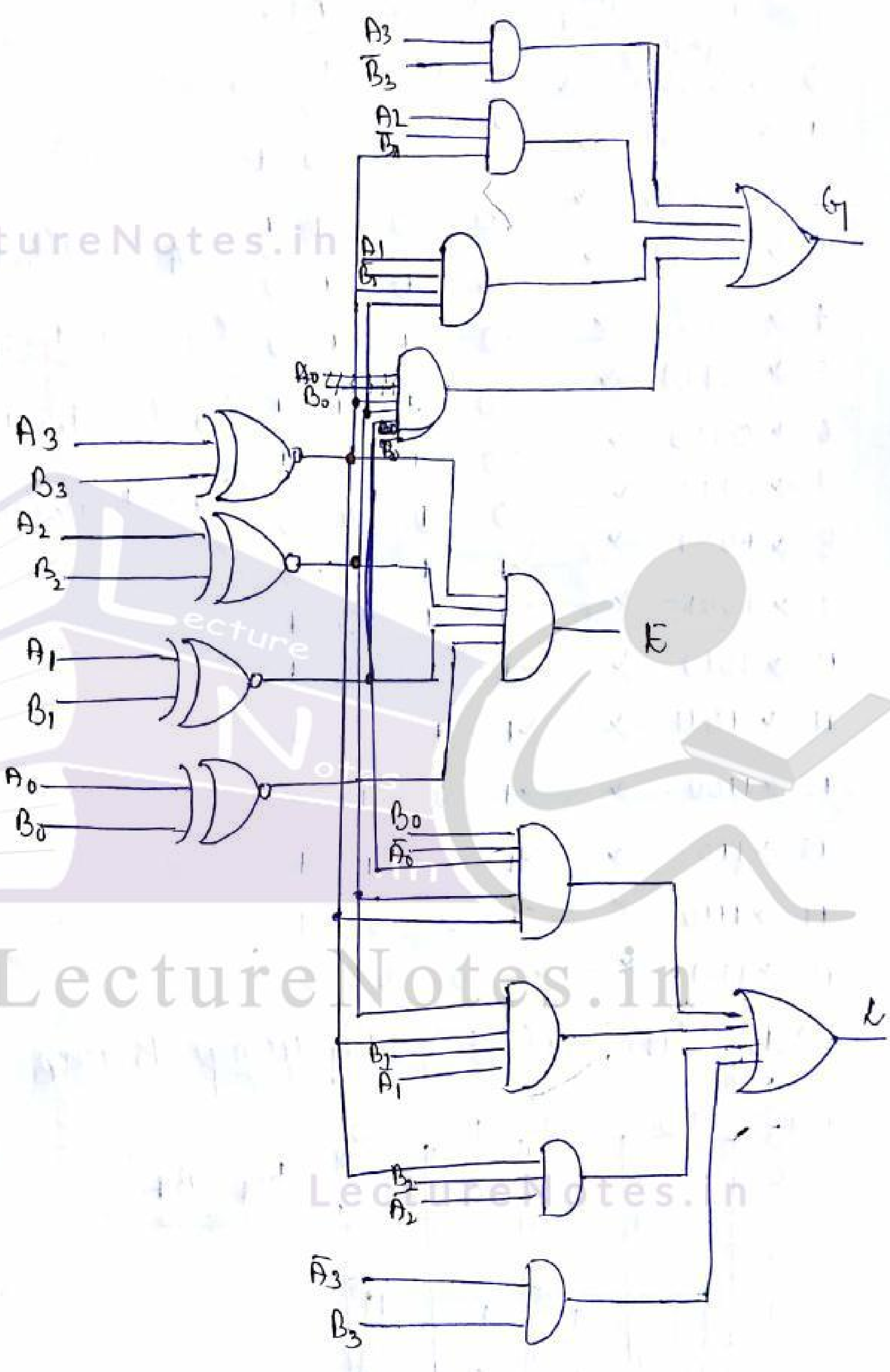
Equations:-

$$G = A_3\bar{B}_3 + (A_3 \odot B_3) A_2 \bar{B}_2 + (A_3 \odot B_3)(A_2 \odot B_2) A_1 \bar{B}_1 + (A_3 \odot B_3)(A_2 \odot B_2)(A_1 \odot B_1) A_0 \bar{B}_0$$

$$L = \bar{A}_3 B_3 + (A_3 \odot B_3)\bar{A}_2 B_2 + (A_3 \odot B_3)(A_2 \odot B_2)\bar{A}_1 B_1 + (A_3 \odot B_3)(A_2 \odot B_2)(A_1 \odot B_1)\bar{A}_0 B_0$$

$$E = (A_3 \odot B_3)(A_2 \odot B_2)(A_1 \odot B_1)(A_0 \odot B_0)$$

Logic Diagram for 4-bit Magnitude Comparator:-



Binary to Gray Code Converter:-

	$B_4B_3B_2B_1$	G_4	G_3	G_2	G_1
0 →	0000 →	0	0	0	0
1 →	0001 →	0	0	0	1
2 →	0010 →	0	0	1	1
3 →	0011 →	0	0	1	0
4 →	0100 →	0	1	1	0
5 →	0101 →	0	1	1	1
6 →	0110 →	0	1	0	1
7 →	0111 →	0	1	0	0
8 →	1000 →	1	1	0	0
9 →	1001 →	1	1	0	1
10 →	1010 →	1	1	1	1
11 →	1011 →	1	1	1	0
12 →	1100 →	1	0	1	0
13 →	1101 →	1	0	1	1
14 →	1110 →	1	0	0	1
15 →	1111 →	1	0	0	0

$G_4 = \Sigma m(8,9,10,11,12,13,14,15)$

$G_3 = \Sigma m(4,5,6,7,8,9,10,11)$

$G_2 = \Sigma m(2,3,4,5,10,11,12,13)$

$G_1 = (1,2,5,6,9,10,13,14)$

Solve G_4, G_3, G_2 & G_1 By using K-map.

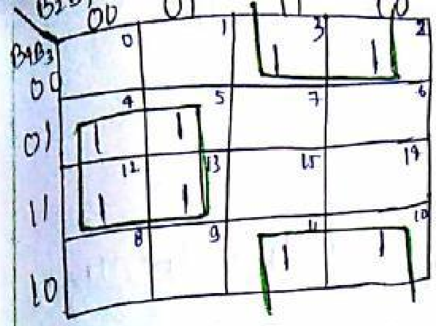
$B_4B_3 \backslash B_2B_1$	00	01	11	10
00				
01				
11	1	1	1	1
10	1	1	1	1

$G_4 = B_4$

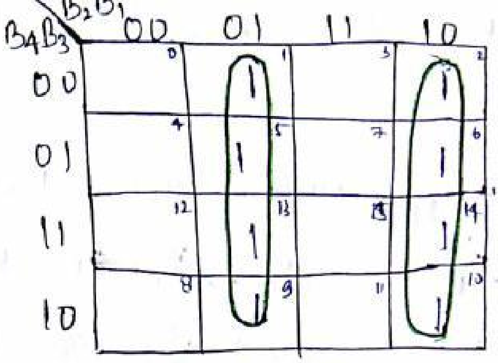
For G_3

$B_4B_3 \backslash B_2B_1$	00	01	11	10
00				
01	1	1	1	1
11				
10	1	1	1	1

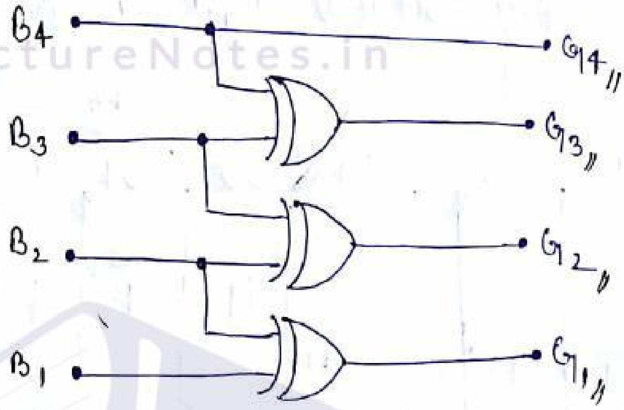
$G_3 = \bar{B}_4 B_3 + \bar{B}_3 B_4 = B_3 \oplus B_4$



$$G_2 = B_3\bar{B}_2 + B_2\bar{B}_3$$
$$= B_3 \oplus B_2$$



$$G_1 = \bar{B}_2 B_1 + B_2 \bar{B}_1$$
$$= B_2 \oplus B_1$$



Gray Code to Binary code Convertor :-

Input → Gray Code ; output → Binary code.

	G_4	G_3	G_2	G_1		B_4	B_3	B_2	B_1	
0 ←	0	0	0	0	→	0	0	0	0	→ 0
1 ←	0	0	0	1	→	0	0	0	1	→ 1
3 ←	0	0	1	1	→	0	0	1	0	→ 2
2 ←	0	0	1	0	→	0	0	1	1	→ 3
6 ←	0	1	1	0	→	0	1	0	0	→ 4
7 ←	0	1	1	1	→	0	1	0	1	→ 5
5 ←	0	1	0	1	→	0	1	1	0	→ 6
4 ←	0	1	0	0	→	0	1	1	1	→ 7
12 ←	1	1	0	0	→	1	0	0	0	→ 8
13 ←	1	1	0	1	→	1	0	0	1	→ 9
15 ←	1	1	1	1	→	1	0	1	0	→ 10
14 ←	1	1	1	0	→	1	0	1	1	→ 11
10 ←	1	0	1	0	→	1	1	0	0	→ 12
11 ←	1	0	1	1	→	1	1	0	1	→ 13

9| 1 0 0 1 → 1 1 1 0

9| 1 0 0 0 → 1 1 1 1

$B_4 = \Sigma m(8,9,10,11,12,13,14,15)$

$B_3 = \Sigma m(4,5,6,7,12,13,14,15) \Rightarrow B_3 = \Sigma m(4,5,6,7,8,9,10,11)$

$B_2 = \Sigma m(2,3,6,7,10,11,14,15) \Rightarrow B_2 = \Sigma m(2,3,4,5,8,9,14,15)$

$B_1 = \Sigma m(1,3,5,7,9,11,13,14) \Rightarrow B_1 = \Sigma m(1,2,4,7,8,11,13,14)$

Solving the expression for B_4, B_3, B_2, B_1 by using K-map

B_4

$G_4G_3 \backslash G_2G_1$	00	01	11	10
00				
01				
11	1	1	1	1
10	1	1	1	1

$B_4 = G_4$

B_3

$G_4G_3 \backslash G_2G_1$	00	01	11	10
00				
01	1	1	1	1
11				
10	1	1	1	1

$B_3 = \bar{G_4}G_3 + G_4\bar{G_3} = G_4 \oplus G_3$

$G_4G_3 \backslash G_2G_1$	00	01	11	10
00			1	1
01	1	1		
11			1	1
10	1	1		

$G_4G_3 \backslash G_2G_1$	00	01	11	10
00		1		1
01	1		1	
11		1		1
10	1		1	

$B_2 = \bar{G_4}\bar{G_3}G_2 + \bar{G_4}G_3\bar{G_2} + G_4\bar{G_3}\bar{G_2} + G_4G_3G_2$

$= \bar{G_4}(\bar{G_3}G_2 + G_3\bar{G_2}) + G_4(\bar{G_3}\bar{G_2} + G_3G_2)$

$= \bar{G_4}(G_3 \oplus G_2) + G_4(\overline{G_3 \oplus G_2})$

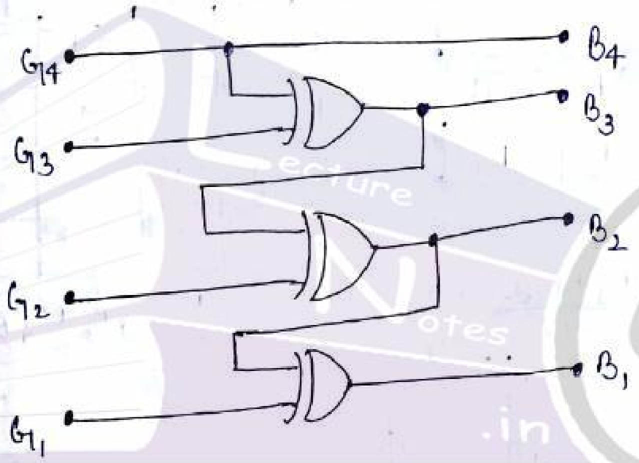
$= G_4 \oplus G_3 \oplus G_2$

$B_2 = B_3 \oplus G_2$

Scanned by CamScanner

$$B_1 = \overline{G_4}\,\overline{G_3}\,\overline{G_2}\,G_1 + \overline{G_4}\,\overline{G_3}\,G_2\,\overline{G_1} + \overline{G_4}\,G_3\,\overline{G_2}\,\overline{G_1} + \overline{G_4}\,G_3\,G_2\,G_1$$
$$+ G_4\,G_3\,\overline{G_2}\,G_1 + G_4\,G_3\,G_2\,\overline{G_1} + G_4\,\overline{G_3}\,\overline{G_2}\,\overline{G_1} + G_4\,\overline{G_3}\,G_2\,G_1$$
$$\Rightarrow \overline{G_4}\,\overline{G_3}(\overline{G_2}G_1 + G_2\overline{G_1}) + \overline{G_4}\,G_3(\overline{G_2}\,\overline{G_1} + G_2G_1) + G_4G_3(\overline{G_2}G_1 + G_2\overline{G_1})$$
$$+ G_4\overline{G_3}(\overline{G_2}\,\overline{G_1} + G_2G_1)$$
$$\Rightarrow \overline{G_4}\,\overline{G_3}(G_2 \oplus G_1) + \overline{G_4}\,G_3(\overline{G_2 \oplus G_1}) + G_4G_3(G_2 \oplus G_1) + G_4\overline{G_3}(\overline{G_2 \oplus G_1})$$
$$\Rightarrow (G_2 \oplus G_1)(\overline{G_4 \oplus G_3}) + (\overline{G_2 \oplus G_1})(G_4 \oplus G_3) = G_4 \oplus G_3 \oplus G_2 \oplus G_1$$
$$= B_3 \oplus G_2 \oplus G_1$$
$$\boxed{B_1 = B_2 \oplus G_1}$$

LOGIC DIAGRAM FOR GRAY TO BINARY CONVERTOR:



4-Bit Binary to B-C-D Code Convertor:

	Binary Code B_4	B_3	B_2	B_1		BCD Code A	B	C	D	E
0 →	0	0	0	0	→	0	0	0	0	0
1 →	0	0	0	1	→	0	0	0	0	1
2 →	0	0	1	0	→	0	0	0	1	0
3 →	0	0	1	1	→	0	0	0	1	1
4 →	0	1	0	0	→	0	0	1	0	0
5 →	0	1	0	1	→	0	0	1	0	1
6 →	0	1	1	0	→	0	0	1	1	0
7 →	0	1	1	1	→	0	0	1	1	1
8 →	1	0	0	0	→	0	1	0	0	0
9 →	1	0	0	1	→	0	1	0	0	1
10 →	1	0	1	0	→	1	0	0	0	0
11 →	1	0	1	1	→	1	0	0	0	1
12 →	1	1	0	0	→	1	0	0	1	0
13 →	1	1	0	1	→	1	0	0	1	1
14 →	1	1	1	0	→	1	0	1	0	0
15 →	1	1	1	1	→	1	0	1	0	1

$A = \Sigma m(10, 11, 12, 13, 14, 15)$

$B = \Sigma m(8, 9)$

$C = \Sigma m(4, 5, 6, 7, 14, 15)$

$D = \Sigma m(2, 3, 6, 7, 12, 13)$

$E = \Sigma m(1, 3, 5, 7, 9, 11, 13, 15)$

$A = B_4 B_3 + B_4 B_2$; $B = B_4 \bar{B}_3 \bar{B}_2$; $C = \bar{B}_4 B_3 + B_3 B_2$

$D = \bar{B}_4 B_2 + B_4 B_3 \bar{B}_2$ $E = B_1$

A

$B_4B_3 \backslash B_2B_1$	00	01	11	10
00				
01				
11	1	1	1	1
10			1	1

$A = B_4 B_3 + B_4 B_2$

B

$B_4B_3 \backslash B_2B_1$	00	01	11	10
00				
01				
11				
10	1	1		

$B = B_4 \bar{B}_3 \bar{B}_2$

C

$B_4B_3 \backslash B_2B_1$	00	01	11	10
00				
01	1	1	1	1
11			1	1
10				

$C = \bar{B}_4 B_3 + B_3 B_2$

D

$B_4B_3 \backslash B_2B_1$	00	01	11	10
00			1	1
01			1	1
11	1	1		
10				

$D = \bar{B}_4 B_2 + B_4 B_3 \bar{B}_2$

$B_4B_3 \backslash B_2B_1$	00	01	11	10
00		1	1	
01		1	1	
11		1	1	
10		1	1	

$E = B_1$

25/2/19

4-Bit B.C.D to Excess-3 Code Convertor:-

4-Bit B.C.D $B_4\ B_3\ B_2\ B_1$	XS-3 Code $X_4\ X_3\ X_2\ X_1$
0 0 0 0	0 0 1 1
0 0 0 1	0 1 0 0
0 0 1 0	0 1 0 1
0 0 1 1	0 1 1 0
0 1 0 0	0 1 1 1
0 1 0 1	1 0 0 0
0 1 1 0	1 0 0 1
0 1 1 1	1 0 1 0

1000 → 1011

1001 → 1100

$X_4 = \Sigma m(5,6,7,8,9) + d(10,11,12,13,14,15)$

$X_3 = \Sigma m(1,2,3,4,9) + d(10,11,12,13,14,15)$

$X_2 = \Sigma m(0,3,4,7,8) + d(10,11,12,13,14,15)$

$X_1 = \Sigma m(0,2,4,6,8) + d(10,11,12,13,14,15)$

$B_4B_3 \backslash B_2B_1$	00	01	11	10
00				
01		1	1	1
11	X	X	X	X
10	1	1	X	X

$X_4 = B_4 + B_3B_1 + B_3B_2$

$B_4B_3 \backslash B_2B_1$	00	01	11	10
00		1	1	1
01	1			
11	X	X	X	X
10		1	X	X

$X_3 = \bar{B}_3B_1 + \bar{B}_3B_2 + B_3\bar{B}_2\bar{B}_1$

$B_4B_3 \backslash B_2B_1$	00	10	11	10
00	1		1	
10	1		1	
11	X	X	X	X
10	1		X	X

$X_2 = \bar{B}_2\bar{B}_1 + B_2B_1$

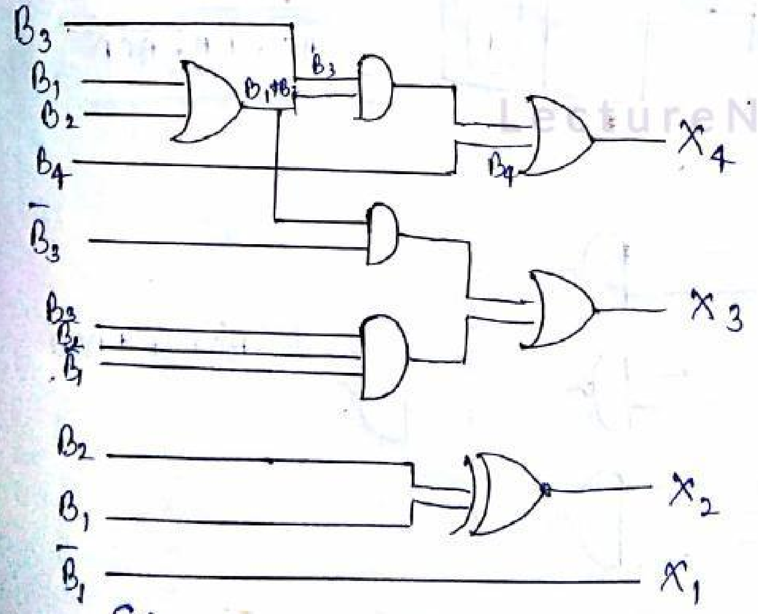
$= B_2 \odot B_1$

$B_4B_3 \backslash B_2B_1$	00	01	11	10
00	1			1
01	1			1
11	X	X	X	X
10	1		X	X

$X_1 = \bar{B}_2\bar{B}_1 + B_2\bar{B}_1$

$= \bar{B}_1(\bar{B}_2 + B_2)$

$= \bar{B}_1$



Logic Diagram of 4-bit B.C.D to XS-3

Design an S.O.P. circuit to detect the decimal number 5 through 12 in a 4-bit Gray Code Input

Solution:-

Decimal	Binary	Gray Code	Output 'f'
0	0000	0000	0
1	0001	0001	0
2	0010	0011	0
3	0011	0010	0
4	0100	0110	0
5	0101	0111	1
6	0110	0101	1
7	0111	0100	1
8	1000	1100	1
9	1001	1101	1
10	1010	1111	1
11	1011	1110	1
12	1100	1010	1
13	1101	1011	0
14	1110	1001	0
15	1111	1000	0

$f = \Sigma m(5, 6, 7, 8, 9, 10, 11, 12)$

$f = \Sigma m(7, 5, 4, 12, 13, 15, 14, 10) = \Sigma m(5, 4, 7, 10, 12, 13, 14, 15)$

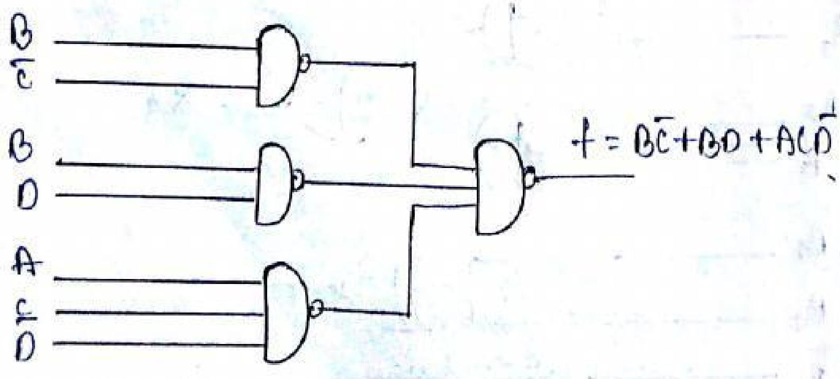
AB \ CD	00	01	11	10
00				
01	1	1	1	
11	1	1	1	1
10				1

$f = B\bar{C} + BD + AC\bar{D}$

$f = \overline{\overline{B\bar{C} + BD + AC\bar{D}}}$

$= \overline{\overline{B\bar{C}} \cdot \overline{BD} \cdot \overline{AC\bar{D}}}$

$=$



4-Bit B.C.D to Gray Code Convertor:-

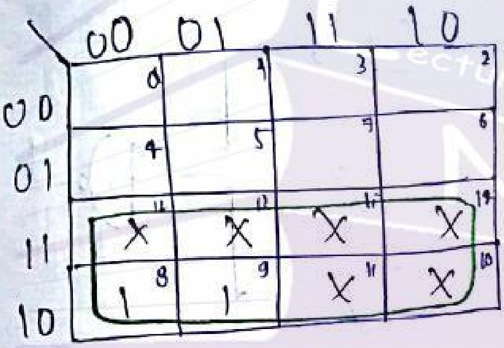
4-Bit B.C.D				Gray Code			
B_4	B_3	B_2	B_1	G_4	G_3	G_2	G_1
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1

$G_4 = \Sigma m(8,9) + d(10,11,12,13,14,15)$

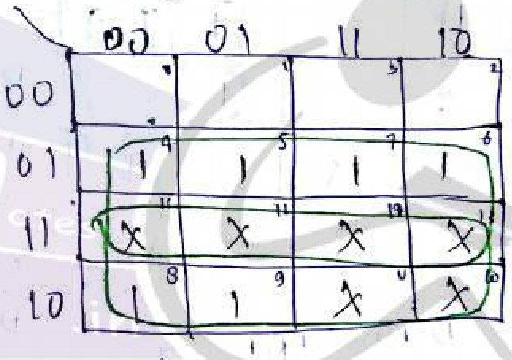
$G_3 = \Sigma m(4,5,6,7,8,9) + d(10,11,12,13,14,15)$

$G_2 = \Sigma m(2,3,4,5) + d(10,11,12,13,14,5)$

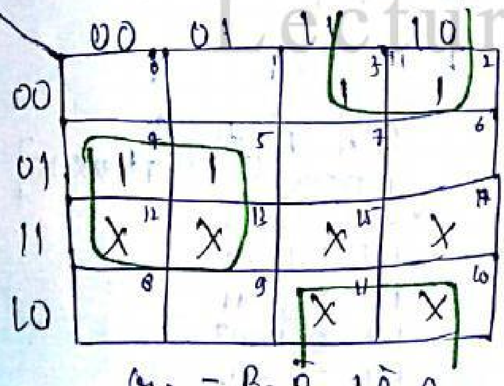
$G_1 = \Sigma m(1,2,5,6,9) + d(10,11,12,13,14,15)$



$G_4 = B_4$

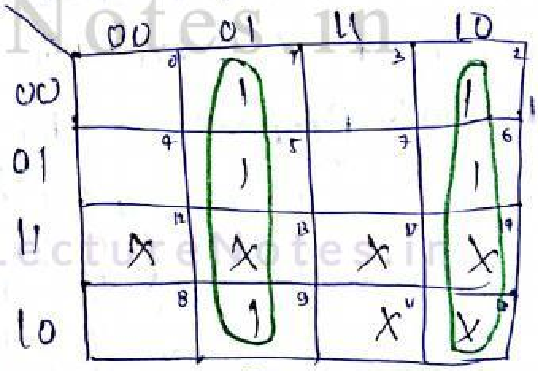


$G_3 = B_3 + B_4$



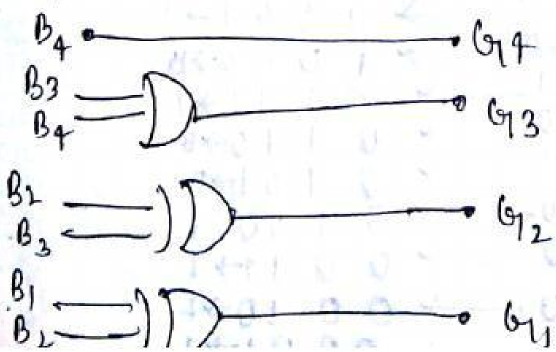
$G_2 = B_3\bar{B}_2 + \bar{B}_3 B_2$

$G_2 = B_3 \oplus B_2$



$G_1 = \bar{B}_2 B_1 + B_2 \bar{B}_1$

$G_1 = B_1 \oplus B_2$



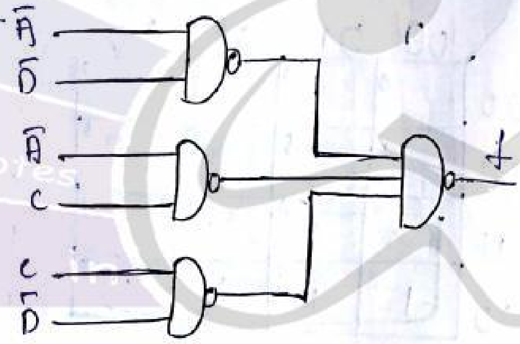
Design an S.O.P circuit to detect the decimal numbers 0, 2, 4, 6, 8 in a 5211 B.C.D code input

Solution:-

Decimal	Input	Output
0	0 0 0 0 →	1
1	0 0 0 1 →	0
2	0 0 1 1 →	1
3	0 1 0 1 →	0
4	0 1 1 1 →	1
5	1 0 0 0 →	0
6	1 0 1 0 →	1
7	1 1 0 0 →	0
8	1 1 1 0 →	1
9	1 1 1 1 →	0

$f = \Sigma m(0, 3, 7, 10, 14) + d(2, 4, 6, 9, 11, 13)$

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



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

Design a Combinational circuit to produce 2's complement of 4-bit Binary number.

Solution:-

Decimal	Input	Output [2's complement] $T_4 T_3 T_2 T_1$
0 →	0 0 0 0 →	0 0 0 0 → 0
1 →	0 0 0 1 →	1 1 1 1 → 15
2 →	0 0 1 0 →	1 1 1 0 → 14
3 →	0 0 1 1 →	1 1 0 1 → 13
4 →	0 1 0 0 →	1 1 0 0 → 12
5 →	0 1 0 1 →	1 0 1 1 → 11
6 →	0 1 1 0 →	1 0 1 0 → 10
7 →	0 1 1 1 →	1 0 0 1 → 9
8 →	1 0 0 0 →	1 0 0 0 → 8
9 →	1 0 0 1 →	0 1 1 1 → 7
10 →	1 0 1 0 →	0 1 1 0 → 6
11 →	1 0 1 1 →	0 1 0 1 → 5
12 →	1 1 0 0 →	0 1 0 0 → 4
13 →	1 0 1 0 →	0 0 1 1 → 3
14 →	1 1 1 0 →	0 0 1 0 → 2
15 →	1 1 1 1 →	0 0 0 1 → 1

$T_4 = \Sigma m(1,2,3,4,5,6,7,8)$

$T_3 = \Sigma m(1,2,3,4,9,10,11,12)$

$T_2 = \Sigma m(1,2,5,6,9,10,13,14)$

$T_1 = \Sigma m(1,3,5,7,9,11,13,15)$

$B_4B_3 \backslash B_2B_1$	00	01	11	10
00		1	1	1
01	1	1	1	1
11				
10	1			

$T_4 = B_4\bar{B}_3\bar{B}_2\bar{B}_1 + \bar{B}_4B_3 + \bar{B}_4B_1 + \bar{B}_4B_2$

$B_4(B_2+B_1) + \bar{B}_4B_3$

$B_4B_3 \backslash B_2B_1$	00	01	11	10
00		1	1	1
01	1			
11	1			
10		1	1	1

$T_3 = B_3\bar{B}_2\bar{B}_1 + \bar{B}_3B_1 + \bar{B}_3B_2$

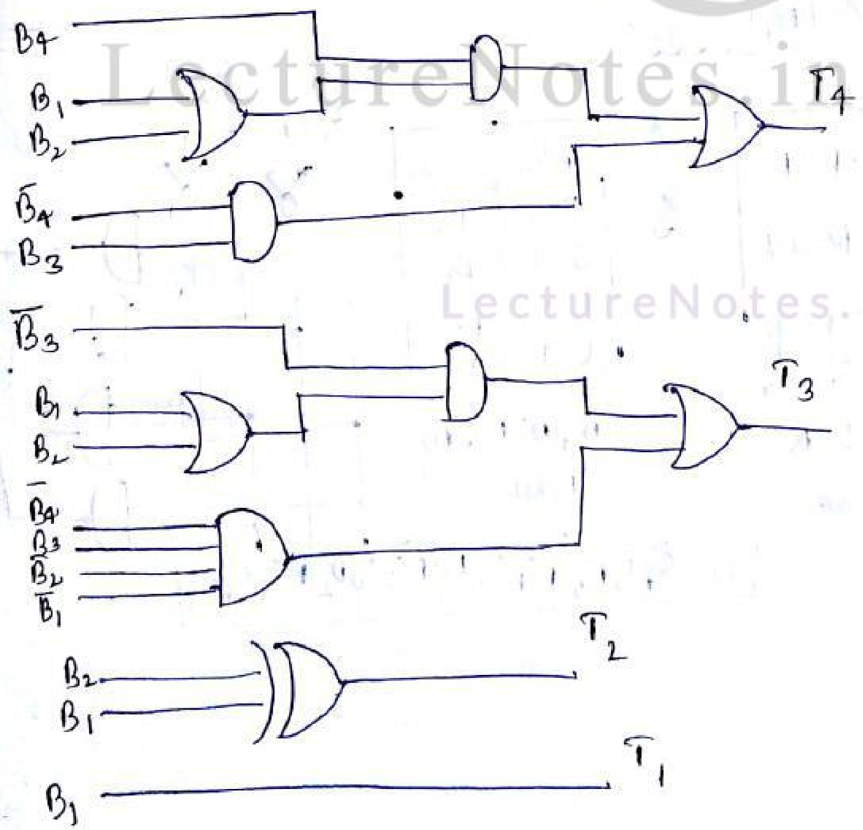
$B_4B_3 \backslash B_2B_1$	00	01	11	10
00		1		1
01		1		1
11		1		1
10		1		1

$T_2 = \bar{B}_2B_1 + B_2\bar{B}_1$

$= B_2 \oplus B_1$

$B_4B_3 \backslash B_2B_1$	00	01	11	10
00		1	1	
01		1	1	
11		1	1	
10		1	1	

$T_1 = B_1$

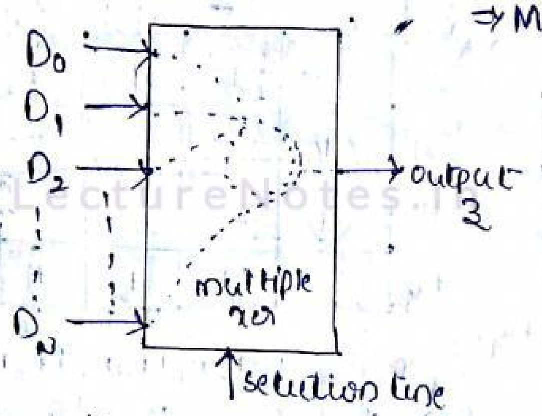


Multiplexers:-

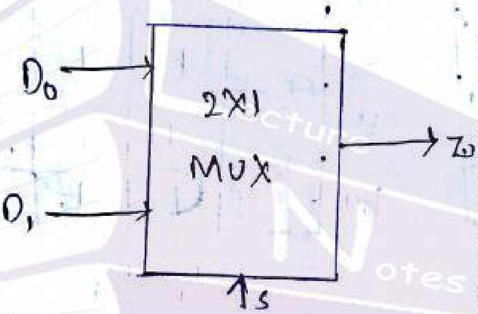
→ A multiplexer is a combinational circuit that accepts several data inputs and select only one of them as the output depending upon the selection lines

⇒ 2^n i/p lines; n selection lines, 1 o/p line.

⇒ Multiplexer is data selector



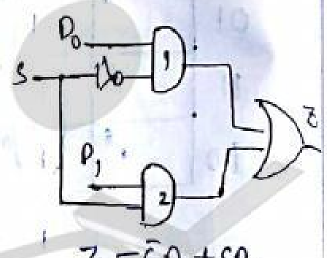
2X1 Multiplexer:-



fig(a) Block Diagram

S	Z
0	D_0
1	D_1

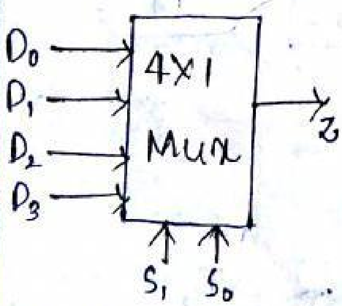
fig(b) Truth Table



$Z = \bar{S}D_0 + SD_1$

Logic Diagram

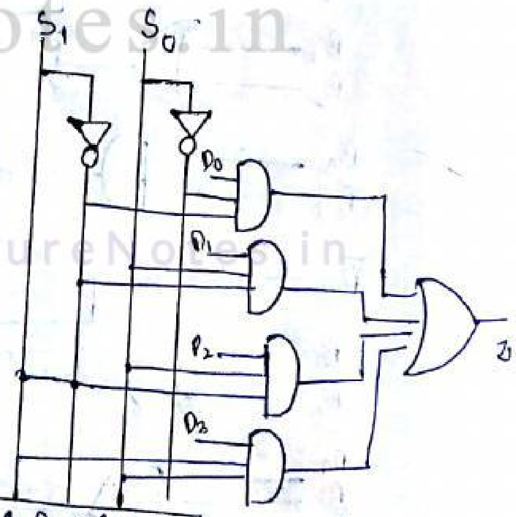
4X1 Multiplexer:-



fig(a) Block Diagram

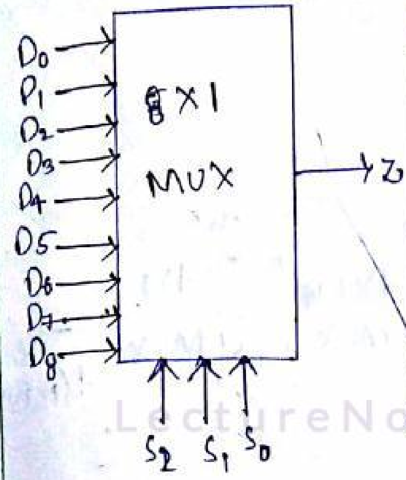
S_1 S_0	Z
0 0	D_0
0 1	D_1
1 0	D_2
1 1	D_3

fig(b) Truth Table.

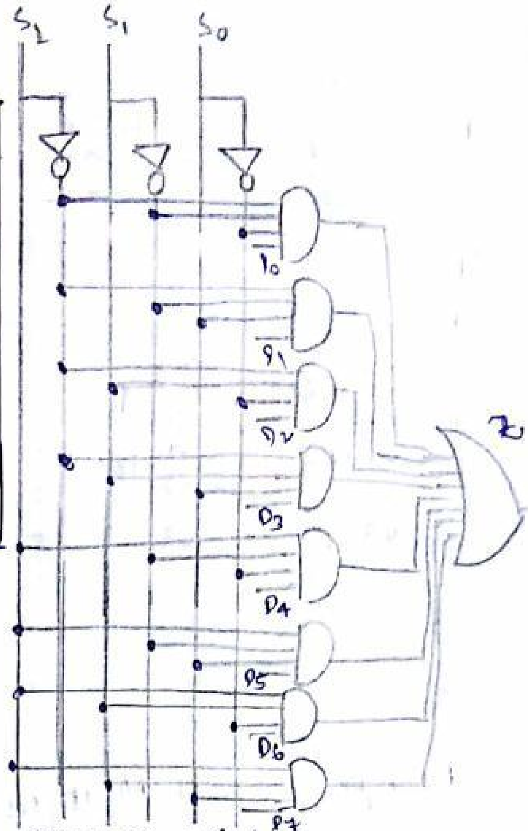


$$\boxed{Z = \bar{S}_1\bar{S}_0D_0 + S_1\bar{S}_0D_1 + \bar{S}_1S_0D_2 + S_1S_0D_3}$$

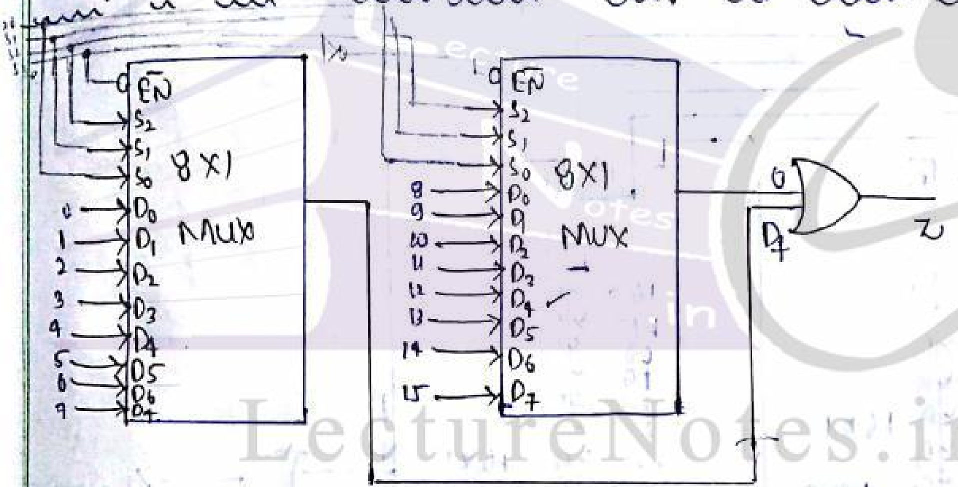
8X1 Multiplexers:-



$S_2 S_1 S_0$	Z
0 0 0	D_0
0 0 1	D_1
0 1 0	D_2
0 1 1	D_3
1 0 0	D_4
1 0 1	D_5
1 1 0	D_6
1 1 1	D_7



Design of 16X1 Multiplexer using 8X1 Multiplexers:-



Design of 32X1 Multiplexer using 16X1 multiplexer

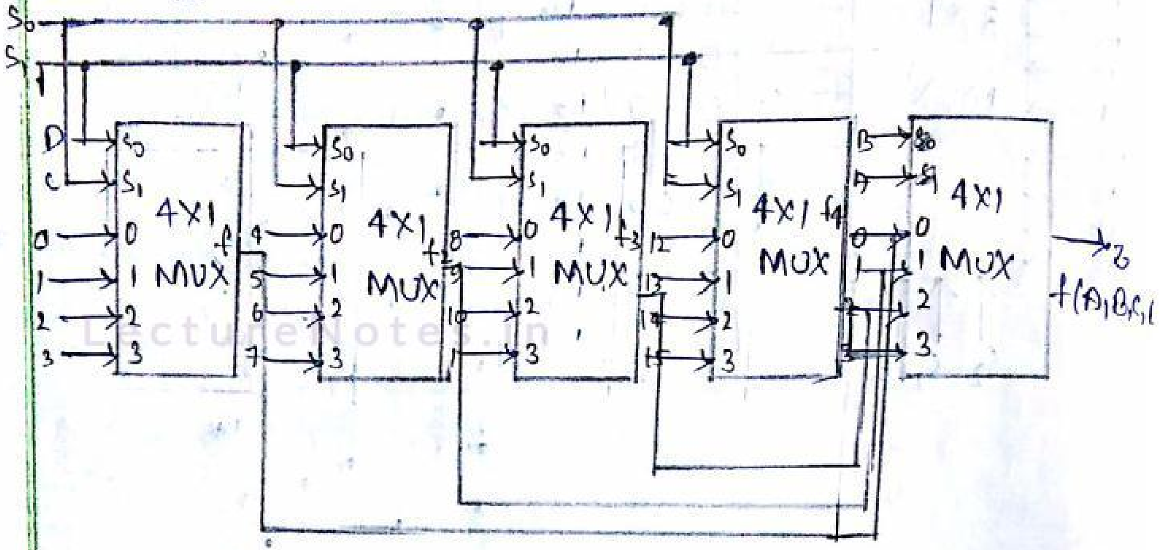
100

0110

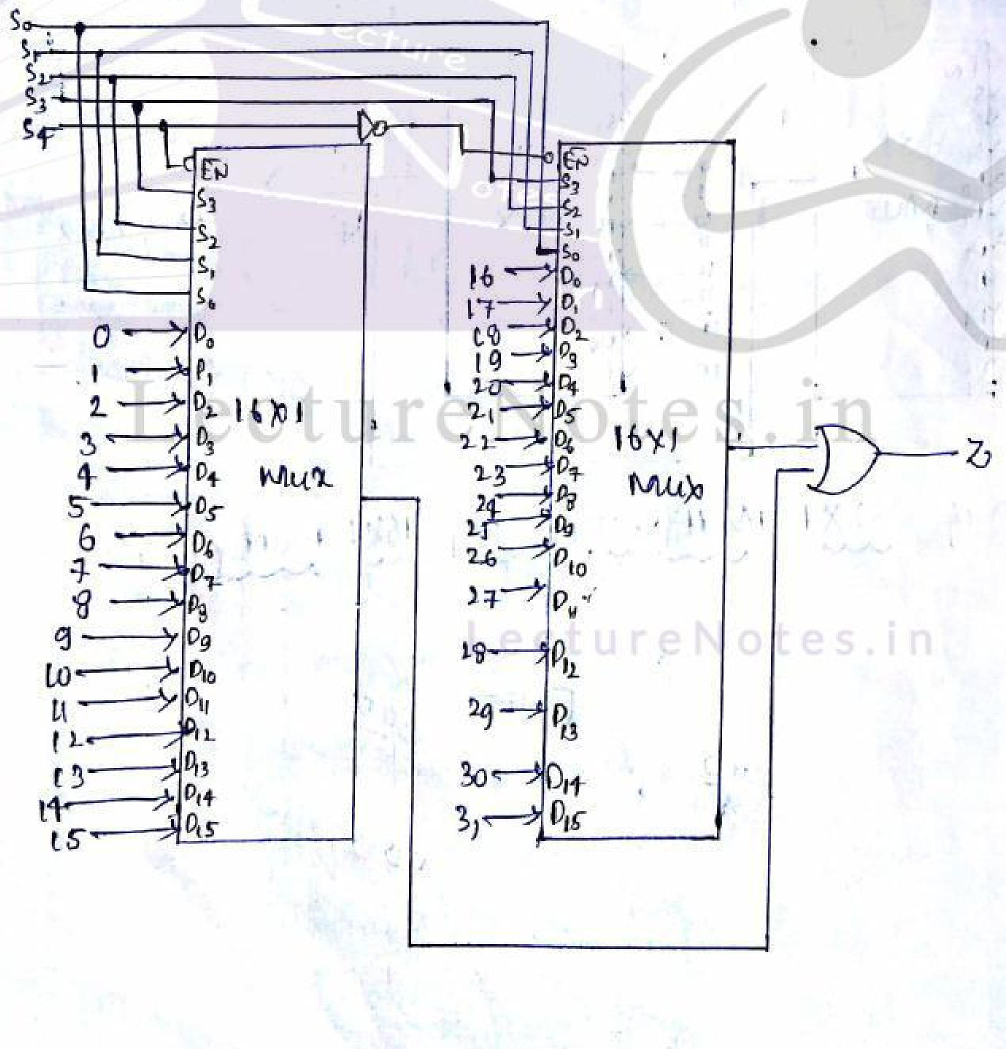
100
$S_3 S_2 S_1 S_0$

1100

Design of 16X1 MUX using 4X1 MUX:-



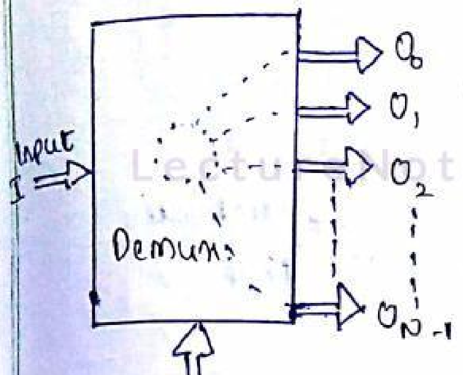
Design of 32X1 Multiplier using 16X1 Multiplier:-



Scanned by CamScanner

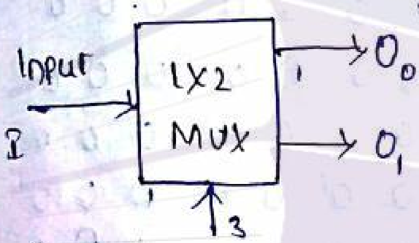
* Demultiplexer (Data distributor):

- ⇒ Demultiplexer is also known as Data Distributor
- ⇒ It is a reverse operation of multiplexer.
- ⇒ 1 input line and 2^n output lines.
- ⇒ n selection lines.



selection lines

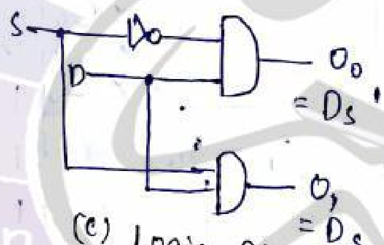
1 line to 2-line Demultiplexer:



Block diagram

S	O
0	O_0
1	O_1

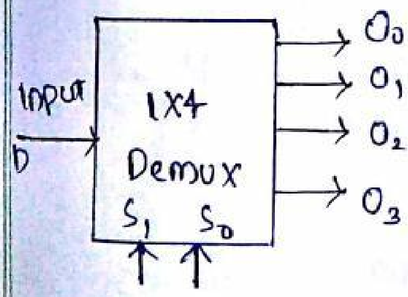
(b) Truth Table



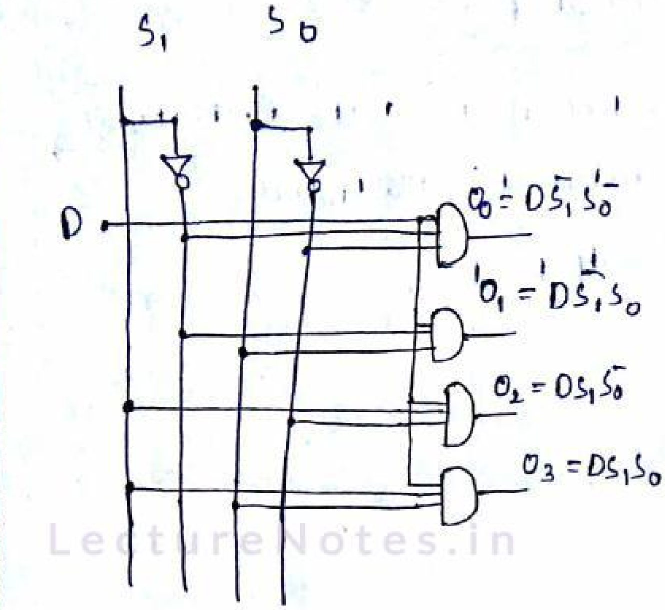
(c) Logic Diagram

1 line to 4-line Demultiplexer:

2 → select
4 → O/P



select i/p		outputs			
S_1	S_0	O_3	O_2	O_1	O_0
0	0	0	0	0	D
0	1	0	0	D	0
1	0	0	0	0	0
1	1	D	0	0	0

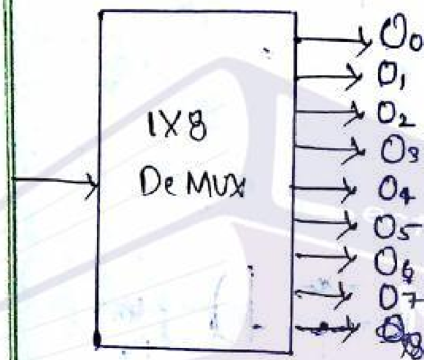


(c) Logic Diagram

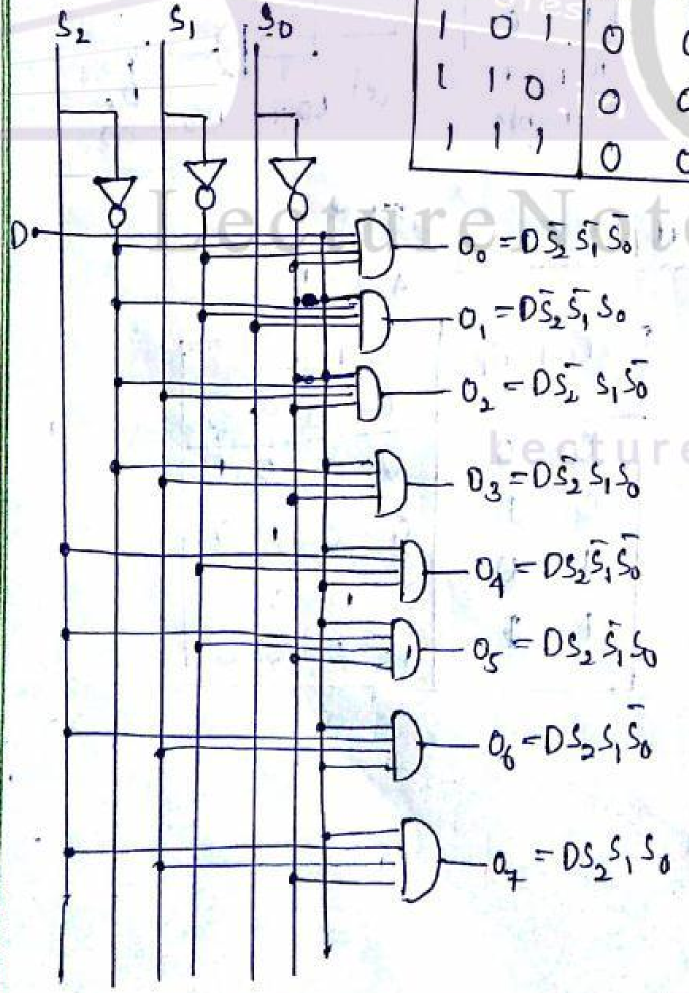
1- Line to 8 Line De-Multiplexer:-

$8 = 2^3$ → selection lines

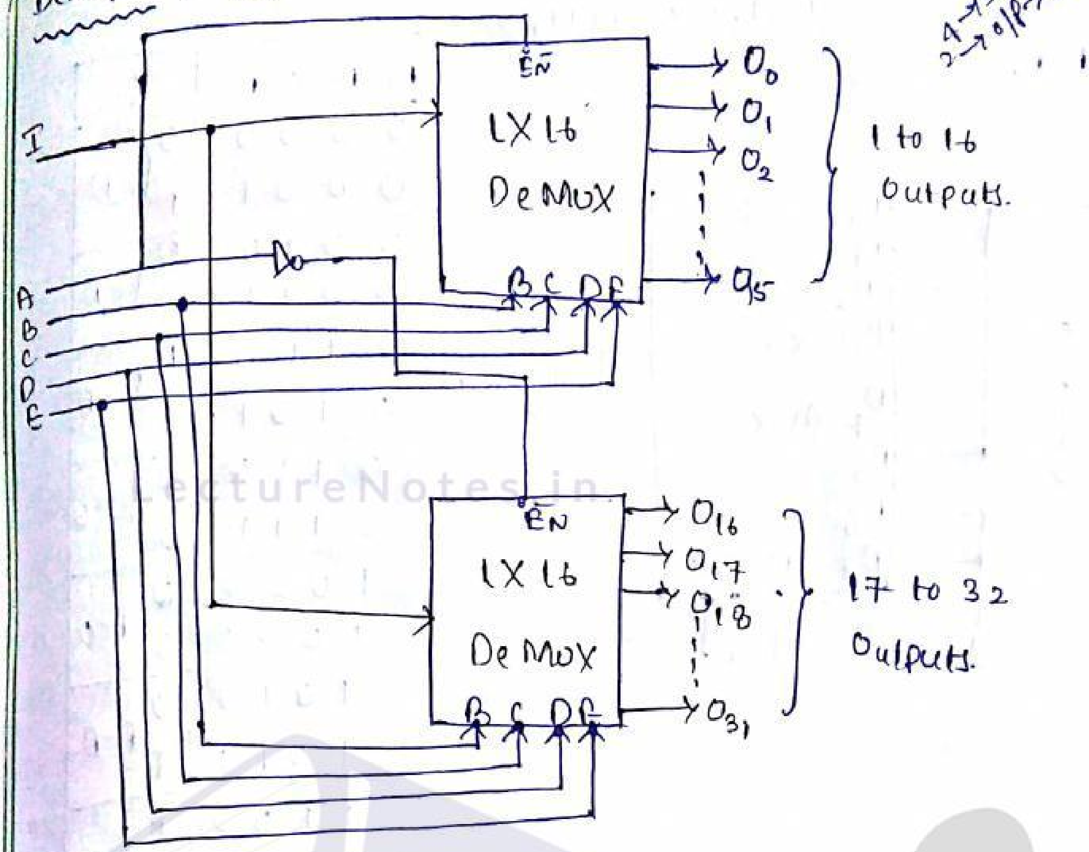
8 o/p



S_2	S_1	S_0	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7
0	0	0	0	0	0	0	0	0	0	0
0	0	1	D	0	0	0	0	0	0	0
0	1	0	0	D	0	0	0	0	0	0
0	1	1	0	0	0	D	0	0	0	0
1	0	0	0	0	0	0	D	0	0	0
1	0	1	0	0	0	0	0	D	0	0
1	1	0	0	0	0	0	0	0	D	0
1	1	1	0	0	0	0	0	0	0	D

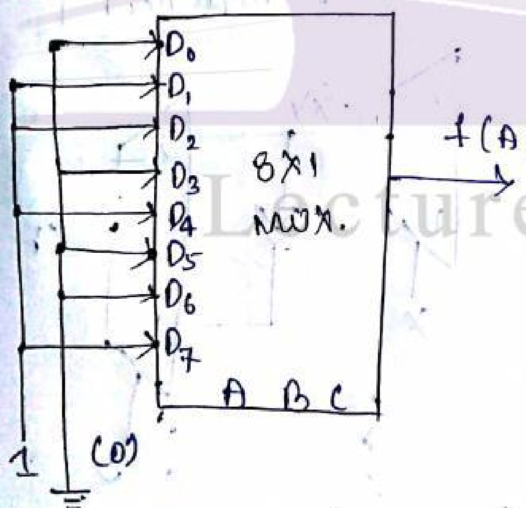


Designing 1X32 De MUX using 1X16 Demux :-



Q) Use a Multiplexer to implement a logic function $f = A \oplus B \oplus C$

Solution :-

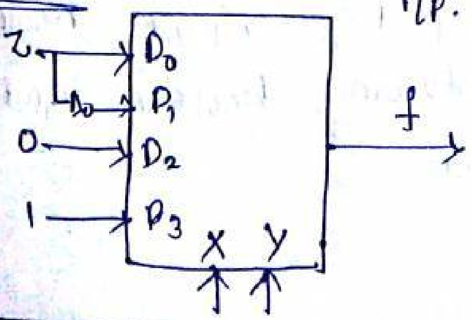


A B C	$f = A \oplus B \oplus C$
0 0 0	0 D_0
0 0 1	1 D_1
0 1 0	1 D_2
0 1 1	0 D_3
1 0 0	1 D_4
1 0 1	0 D_5
1 1 0	0 D_6
1 1 1	1 D_7

Q) Implement the boolean function $f(x,y,z) = \Sigma(1,2,6,7)$ using 4X1 MUX

Solution :-

z is external i/p.

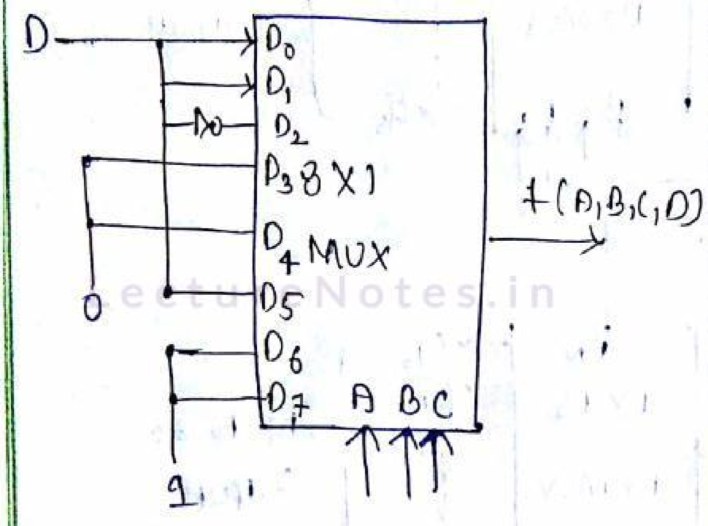


x y z	f	
0 0 0	0	$f = z$
0 0 1	1	
0 1 0	1	$f = \bar{z}$
0 1 1	0	
1 0 0	0	$f = 0$
1 0 1	0	
1 1 0	1	$f = 1$
1 1 1	1	

Implement the Boolean function using 8×1 MUX

$f(A,B,C) = \Sigma(1,3,4,11,12,13,14,15)$

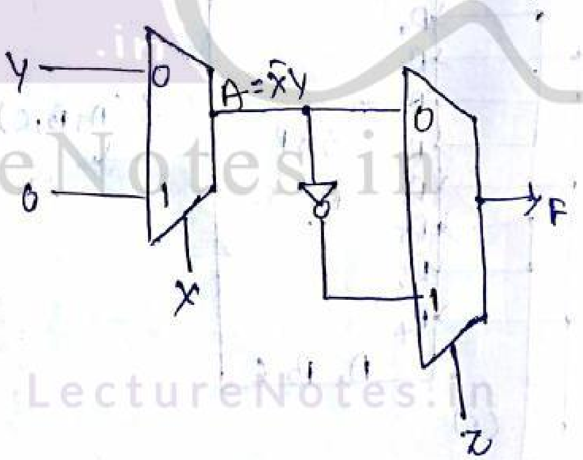
Solution:-



A B C D	f	
0 0 0 0	0	$f=D$
0 0 0 1	1	
0 0 1 0	0	$f=D$
0 0 1 1	1	
0 1 0 0	1	$f=\bar{D}$
0 1 0 1	0	
0 1 1 0	0	$f=0$
0 1 1 1	0	
1 0 0 0	0	$f=0$
1 0 0 1	0	
1 0 1 0	0	$f=D$
1 0 1 1	1	
1 1 0 0	1	$f=1$
1 1 0 1	1	
1 1 1 0	1	$f=1$
1 1 1 1	1	

1) Consider the circuit shown in the fig, The Boolean expression f implemented by the circuit is

Solution:-



$A = \bar{x}y + x.0$

$= \bar{x}y$

$f = \bar{z}.A + z.\bar{A}$

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

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

$= \bar{x}y\bar{z} + xz + \bar{y}z$

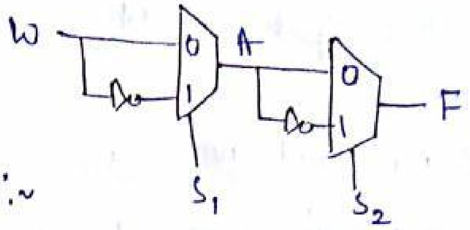
Consider a multiplexer based logic circuit shown in the fig which of the following boolean functions is realized by the circuit

(a) $F = W\overline{S_1 S_2}$

(b) $F = WS_1 + WS_2 + S_1 S_2$

(c) $WS_1 + WS_2 + S_1 S_2$

~~(d)~~ $W \oplus S_1 \oplus S_2$



Solution :-

$A = \bar{S}_1 W + S_1 \bar{W} = S_1 \oplus W$

$f = \bar{S}_1 (S_1 \oplus W) + S_2(\overline{S_1 \oplus W})$

$= W \oplus S_1 \oplus S_2$

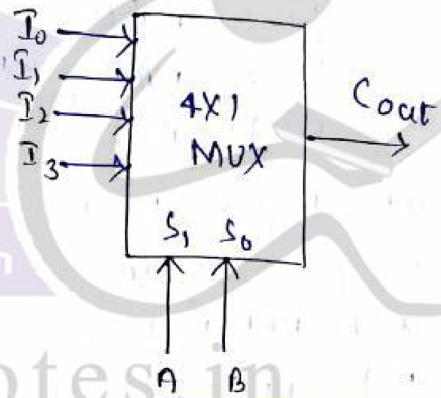
P.) A 4X1 MUX is to be used for generating the output carry of a full adder. A & B are the bits to be added while C_{in} is the i/p carry and C_{out} is the output carry. A & B are used as the select bits with A being the Most significant select bit. Which one of the following statements correctly describes the choice of the signals to be connected to the i/p I_0, I_1, I_2, I_3, so that the o/p is C_o.

~~(a)~~ $I_0 = 0, I_1 = C_{in}, I_2 = C_{in}$ & $I_3 = 1$

(b) $I_0 = 1, I_1 = C_{in}, I_2 = C_{in}$ & $I_3 = 1$

(c) $I_0 = C_{in}, I_1 = 0, I_2 = 1$ & $I_3 = C_{in}$

(d) $I_0 = 0, I_1 = C_{in}, I_2 = 1$ & $I_3 = C_{in}$



Solution :-

$I_0 = 0;\ I_1 = C_{in};\ I_2 = C_{in}$ & $I_3 = 1$

A	B	C_{in}	S	C	
0	0	0	0	0	$I_0 = 0$
0	0	1	1	0	
0	1	0	1	0	$I_1 = C_{in}$
0	1	1	0	1	
1	0	0	1	0	$I_2 = C_{in}$
1	0	1	0	1	
1	1	0	0	1	$I_3 = 1$
1	1	1	1	1	

Scanned by CamScanner

1/3/19 Some gate problems on Multiplexers and Combinational Circuits

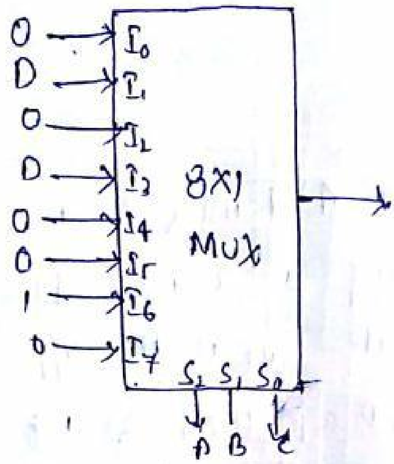
Q1) An 8X1 MUX is used to implement a logical function Y as shown in the fig. The o/p Y is given by

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

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

(c) $Y = AB\bar{C} + \bar{A}CD$

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



000 $\bar{A}\bar{B}\bar{C}$

001 $\bar{A}\bar{B}C \rightarrow I_1$

010 $\bar{A}B\bar{C}$

011 $\bar{A}BC \rightarrow I_3$

100 $A\bar{B}\bar{C}$

101 $A\bar{B}C$

110 $AB\bar{C} \rightarrow I_6$

1.11 ABC

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

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

$= A$

(2) The logical function implemented by the circuit below is [ground implies a logic '0']

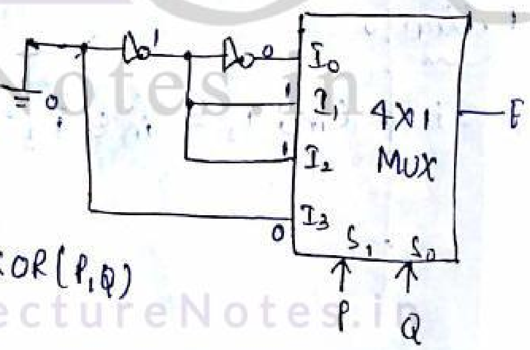
(a) $F = AND(P,Q)$

(b) $F = OR(P,Q)$

(c) $F = XNOR(P,Q)$

(d) $F = XOR(P,Q)$

$\bar{P}Q + P\bar{Q} \Rightarrow XOR(P,Q)$



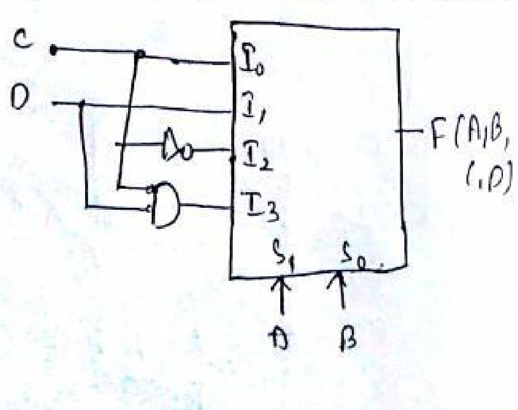
(Q3) The boolean function realized by the logic circuit shown is

(a) $F = \Sigma m(0,1,3,5,9,10,14)$

(b) $F = \Sigma m(2,3,5,7,8,12,13)$

(c) $F = \Sigma m(1,2,4,5,11,14,15)$

(d) $F = \Sigma m(2,3,5,7,8,9,12)$



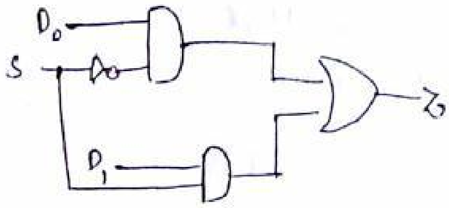
Q4) What are the min. no. of 2X1 multiplexers required to generate a 2-i/p AND gate & a 2 i/p Ex-OR gate?

~~(a)~~ 1 & 2

(b) 1 & 3

(c) 1 & 1

(d) 2 & 2

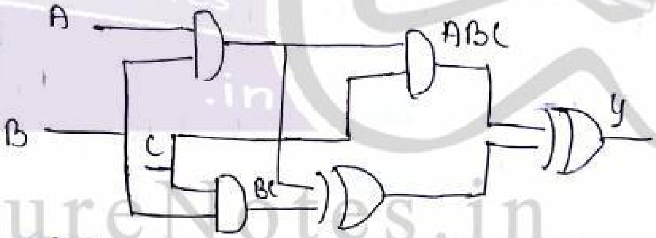


Q5) The o/p Y of a 2-bit comparator is logic 1 whenever the 2-bit i/p A is greater than the 2-bit i/p B. The no. of combination for which the o/p is logic 1 is

(a) 4 ~~(b)~~ 6 (c) 8 (d) 10.

A_1	A_0	B_1	B_0	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Q6) The o/p of the combinational circuit given below is (a) A+B+C (b) A(B+C) ~~(c)~~ B(C+A) (d) C(A+B)



Solution:-

$Y = (\overline{\overline{AB}\,BC + \overline{BC}\,AB})(ABC) + (\overline{ABC})(\overline{AB}\,BC + \overline{BC}\,AB)$

$= (\overline{\overline{AB}\,BC} \cdot \overline{\overline{BC}\,AB})(ABC) + (\overline{ABC})(\overline{AB}\,BC + \overline{BC}AB)$

$= (AB + \overline{BC})(BC + \overline{AB})(ABC) + (\overline{AB}\,C)(\overline{AB}\,BC + \overline{BC}\,AB)$

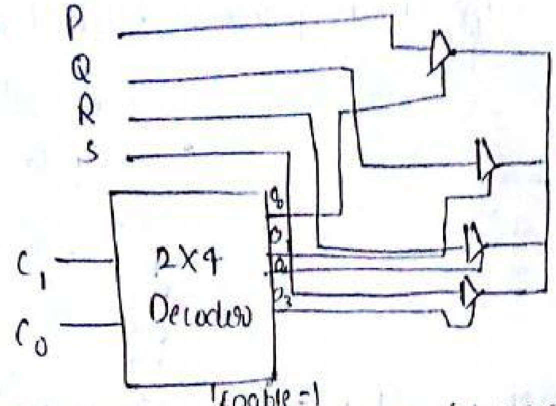
07) The functionality implemented by the circuit below is:

(a) 2 to 1 multiplexer

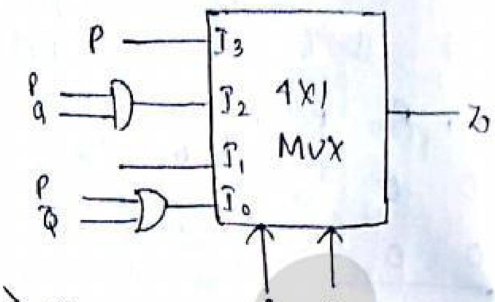
~~(b)~~ 4 to 1 multiplexer

(c) 7 to 1 multiplexer

(d) 6 to 1 multiplexer



08) For the circuit shown in the following figure $I_0 - I_3$ are i/p to the 4X1 multiplexer R(MSB) & S are control bits the o/p Z can be represented by



(a) $PQ + P\bar{Q}S + \bar{Q}R\bar{S}$

(b) $P\bar{Q}\bar{R} + \bar{P}Q\bar{R} + \bar{P}QRS + \bar{Q}R\bar{S}$

(c) $PQ\bar{R} + PQR\bar{S} + P\bar{Q}RS + \bar{Q}R\bar{S}$

(d) $P\bar{Q} + PQ\bar{R} + \bar{P}\bar{Q}\bar{S}$

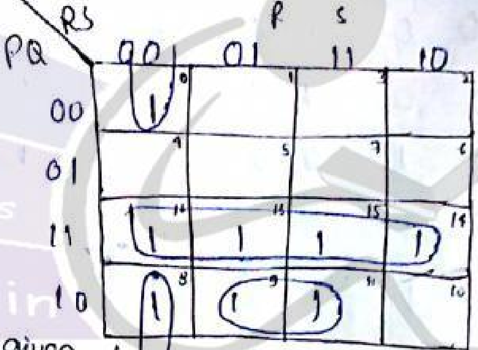
$PRS + PQR\bar{S} + (P+\bar{Q})\bar{R}\bar{S} + P\bar{R}S\bar{Q}$

(D11) 1110 1000 1001

$PRS + PQR\bar{S} + P\bar{R}\bar{S} + \bar{Q}\bar{R}\bar{S} + P\bar{R}S$

1011 1110 1000 0000 1001

1111 1100 1000 1101



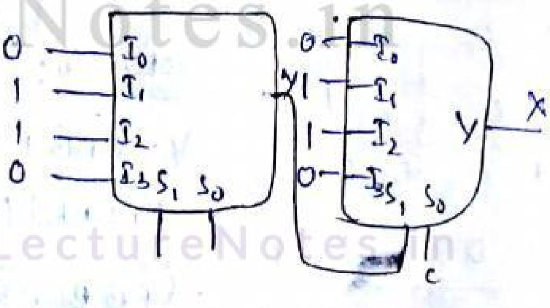
09) In the following circuit, X is given by

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

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

(c) $X = AB + BC + AC$

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



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

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

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

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

10) In the circuit shown, W & V are MSBs of the control i/p. The o/p F is given by 4:1 Mux.

(a) $F = W\bar{X} + \bar{W}\bar{X} + \bar{V}\bar{Z}$

(b) $F = W\bar{X} + \bar{W}X + \bar{V}Z$

~~(c)~~ $F = W\bar{X}\bar{Y} + \bar{W}X\bar{V}$

(d) $F = (\bar{W} + \bar{X})\bar{Y}\bar{Z}$

Solution:-

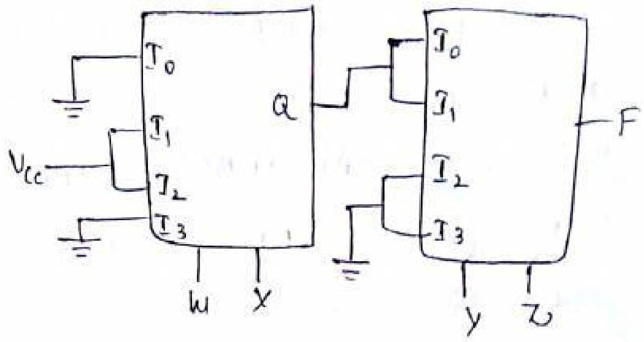
$Q = (w \oplus x)$

$F = \bar{Q}\bar{y}\bar{z} + yz\bar{Q}$

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

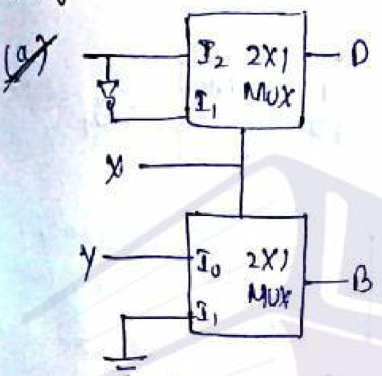
$= Q(\bar{y}(z + \bar{z}))$

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

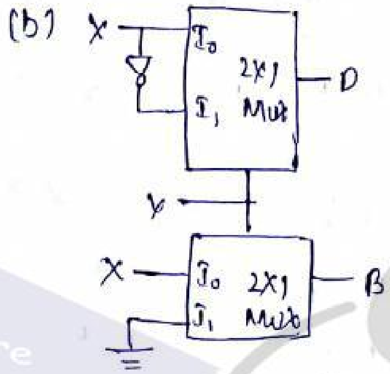


11. If X & Y are i/p & the difference ($D = X - Y$) & the borrow (B) are the O/P, which one of the following diagrams implement a half-subtractor

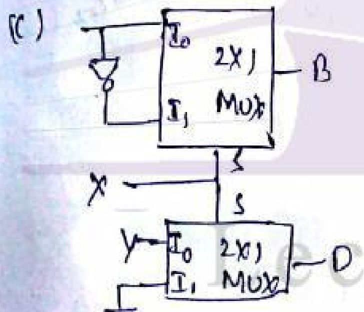
(a)



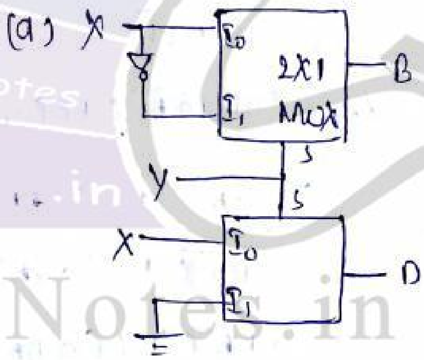
(b)



(c)



(d)



12) The minimum no. of 2 to 1 multiplexers required to realize a 4 to 1 multiplexer is

(a) 1 (b) 2 ~~(c)~~ 3 (d) 4

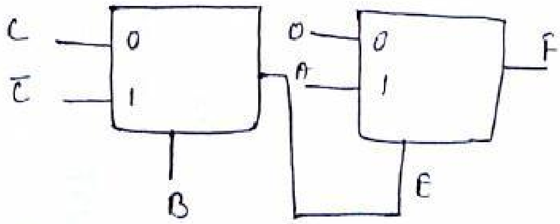
13) The boolean function of implemented in the figure using two i/p multiplexer is

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

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

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

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



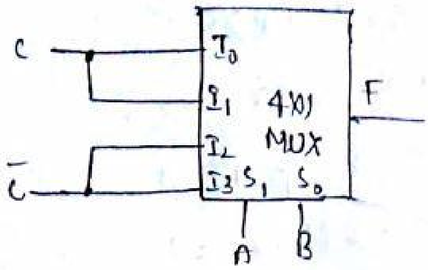
$E = \bar{B}C + B\bar{C}$

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

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

Scanned by CamScanner

(14) The logic realized by the circuit shown in the fig is



(a) $F = A\cdot C$

(b) $F = A\oplus C$ $\qquad \bar A\bar B C+\bar A B C+A\bar B\bar C+AB\bar C$

(c) $F = B\cdot C$ $\qquad \bar A C(B+\bar B)+A\bar C(B+\bar B)$

(d) $F = B+C$ $\qquad =\bar A C+A\bar C$

$= A\oplus C$

(15) A MUX network is shown in figure (i) $Z_1 = ?$

(a) $a+b+c$ (b) $ab+ac+bc$ (c) $a\cup b\cup c$ ~~(d)~~ $a\oplus b\oplus c$

(ii) $Z_2 = ?$

~~(a)~~ $ab+bc+ca$ (b) $a+b+c$ (c) abc (d) $a\cup b\cup c$

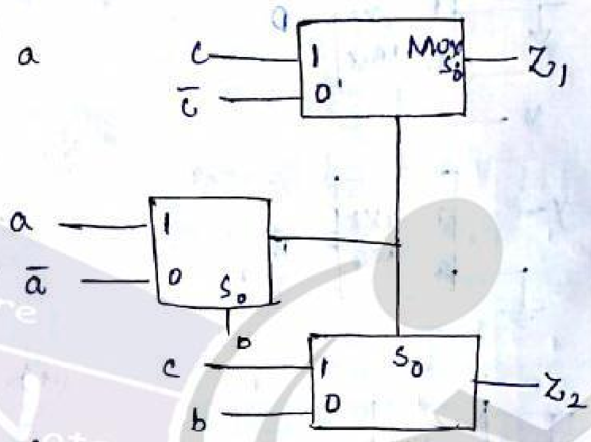
(iii) This circuit acts as a

~~(a)~~ Full adder

(b) Half Adder

(c) Full Subtractor

(d) half subtractor



$Z_0 = ab+\bar a\bar b$

$Z_1 = \overline{(ab+\bar a\bar b)}\bar c + c(ab+\bar a\bar b)$

$= \overline{(a\odot b)}\bar c + c(a\odot b)$

$= (a\oplus b)\bar c + c(a\odot b)$

$= (a\oplus b)\bar c + c(\overline{a\oplus b}) = a\oplus b\oplus c$

$Z_2 = (ab+\bar a\bar b)c + b(\overline{ab+\bar a\bar b})$

$= abc+\bar a\bar b c + b(\overline{a\odot b})$

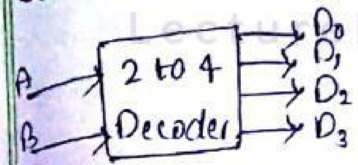
$= abc+\bar a\bar b c + b(a\bar b+b\bar a) = abc+\bar a\bar b c+b\bar a$

5/03/19

Decoders:-

→ A decoder is a combinational circuit that converts n-bit binary data into 2^n output lines. Only one line is activated at a time for each possible combination of the input data.

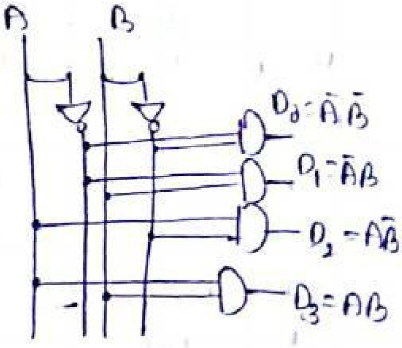
2-4 Decoder:-



Block Diagram

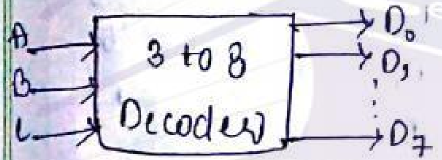
i/p		o/p			
A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Truth Table

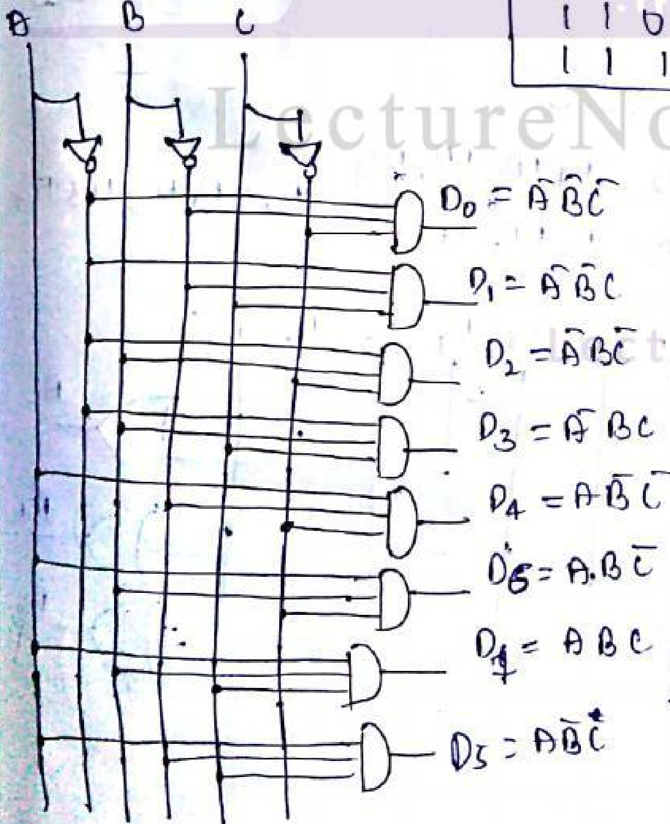


Logic Diagram

3 to 8 Decoder:-

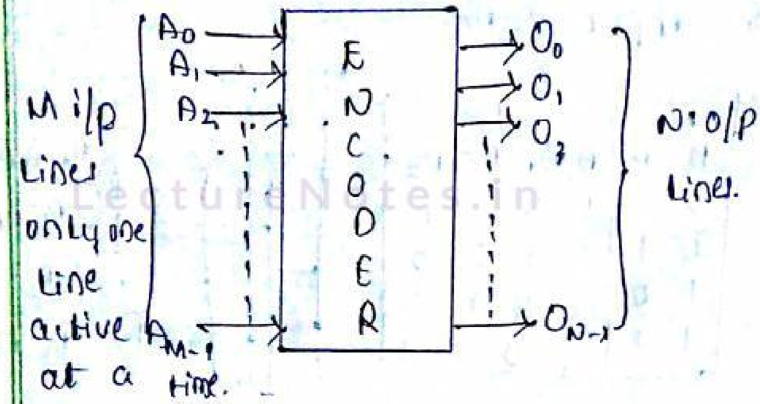


i/p			o/p							
A	B	C	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

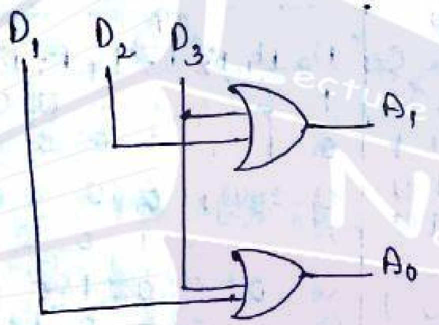


Encoder:

⇒ An encoder is a combinational circuit whose inputs are decimal digits and/or alphabetical characters and whose outputs are coded representations of those inputs.



4 line to 2 line Encoder:



Inputs	Coded Output A_1	A_0
D_0	0	0
D_1	0	1
D_2	1	0
D_3	1	1

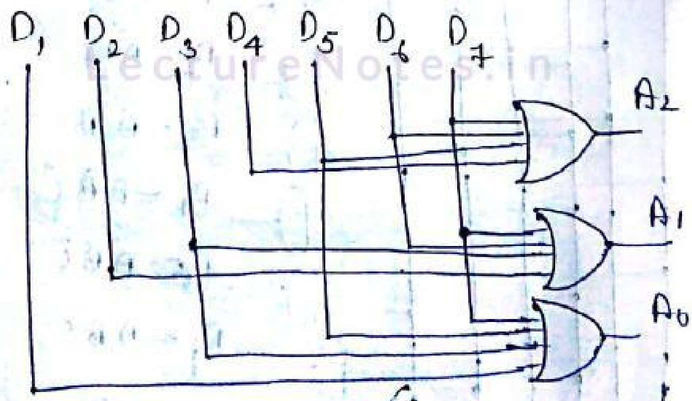
Octal to Binary Encoder:

octal i/p	Coded o/p A_2	A_1	A_0
D_0	0	0	0
D_1	0	0	1
D_2	0	1	0
D_3	0	1	1
D_4	1	0	0
D_5	1	0	1
D_6	1	1	0
D_7	1	1	1

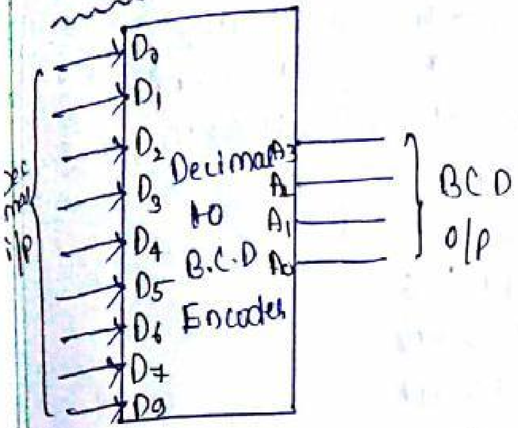
$A_2 = D_4 + D_5 + D_6 + D_7$

$A_1 = D_2 + D_3 + D_6 + D_7$

$A_0 = D_1 + D_3 + D_5 + D_7$



Decimal to B.C.D Encoder



Truth Table

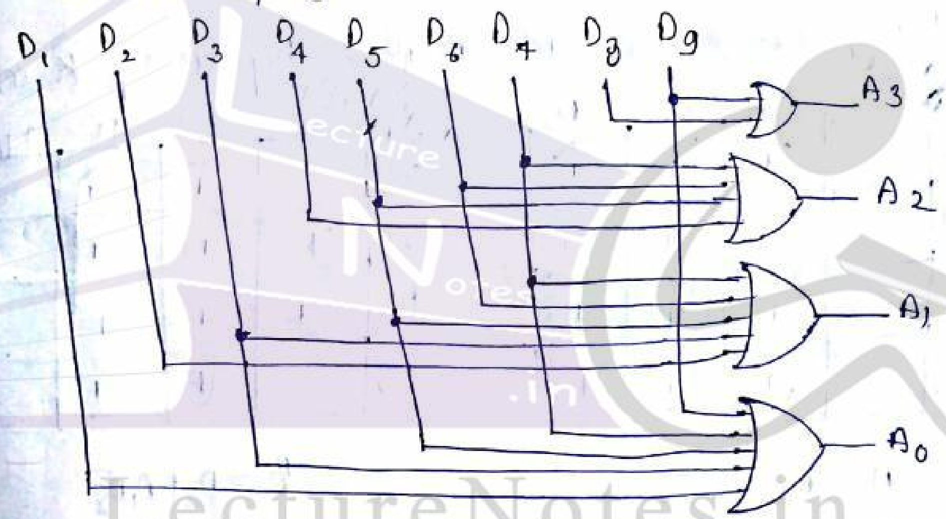
Decimal i/p	Coded o/p A_3 A_2 A_1 A_0
D_0	0 0 0 0
D_1	0 0 0 1
D_2	0 0 1 0
D_3	0 0 1 1
D_4	0 1 0 0
D_5	0 1 0 1
D_6	0 1 1 0
D_7	0 1 1 1
D_8	1 0 0 0
D_9	1 0 0 1

$A_3 = D_8 + D_9$

$A_2 = D_4 + D_5 + D_6 + D_7$

$A_1 = D_2 + D_3 + D_6 + D_7$

$A_0 = D_1 + D_3 + A_5 + D_7 + D_9$



4 bit priority Encoder:-

D_0	D_1	D_2	D_3	A	B	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

A → XX10
0010 → 2
0110 → 6
1010 − 10
1110 − 14
XXX1
0001 − 1
0011 − 3
0101 − 5
0111 − 7
1001 → 9
1010 → 10
1011 → 11
1101 → 13
1111 → 15

$A = \Sigma m(1, 2, 3, 5, 6, 7, 9, 10, 11, 13, 14, 15)$

$B \rightarrow X100$

$0100 \rightarrow 4$

$1100 \rightarrow 12$

$XXX1 \rightarrow$

$0001 \rightarrow 1$

$0011 \rightarrow 5$

$0101 \rightarrow 5$

$0111 \rightarrow 7$

$1001 \rightarrow 9$

$1011 \rightarrow 11$

$1101 \rightarrow 13$

$1111 \rightarrow 15$

$B \rightarrow 1000 \rightarrow 8$

$X100 \rightarrow 4$

$0100 \rightarrow 4$

$1100 \rightarrow 12$

$1001 \rightarrow 9$

$1011 \rightarrow 11$

$1101 \rightarrow 13$

$1111 \rightarrow 15$

$V \rightarrow XX10$

$0010 \rightarrow 2$

$0110 \rightarrow 6$

$1010 \rightarrow 10$

$1110 \rightarrow 14$

$V \rightarrow XXX1$

$0001 \rightarrow 1$

$0101 \rightarrow 5$

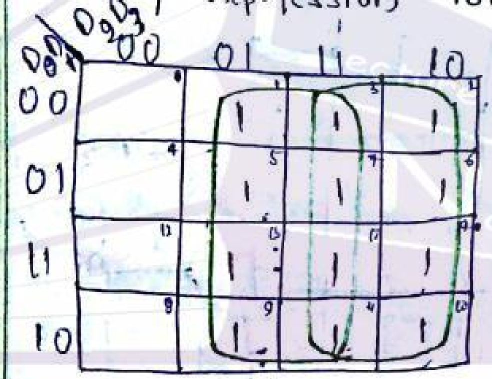
$0011 \rightarrow 3$

$0111 \rightarrow 7$

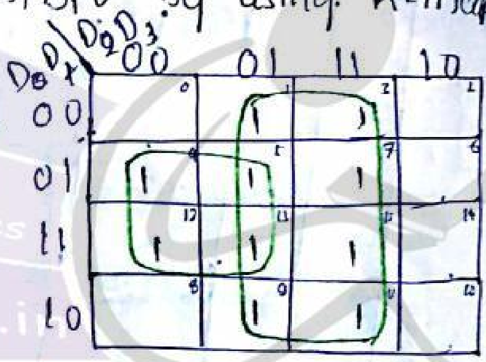
$B = \Sigma(1,3,4,5,7,9,11,13,15)$

$V = \Sigma m(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)$

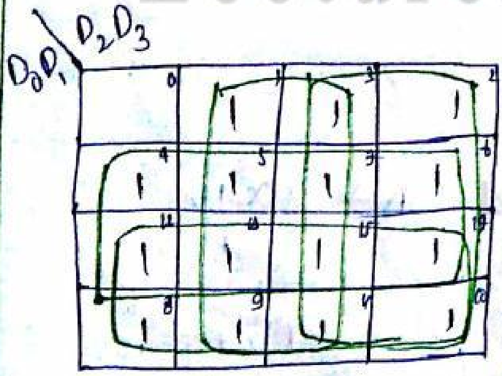
solving expression for A, B, V by using K-map

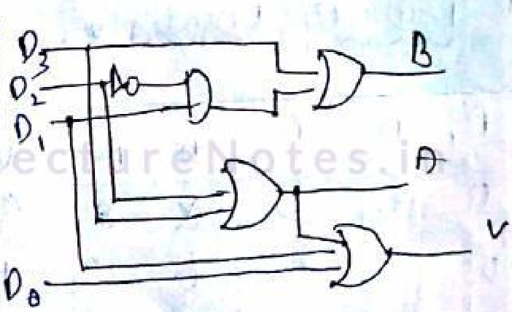


$A = D_2 + D_3$



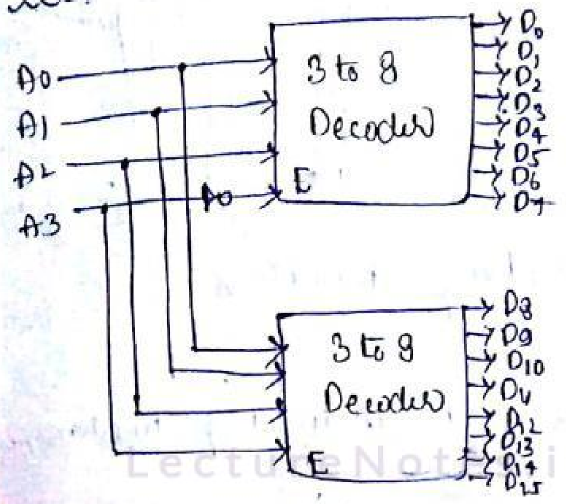
$B = D_3 + D_1\bar{D_2}$





$V = D_0 + D_1 + D_2 + D_3.$

5/03/19
4 to 16 Decoder using 3 to 8 Decoder:-



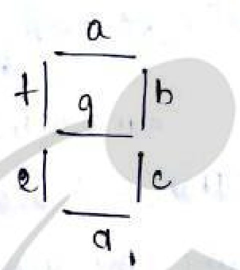
'NOT' gate is used to keep the decoder in high or low modes.

If we give 'NOT' gate to 2nd decoder, 1st decoder will be disabled.

If we give 'NOT' gate to 1st decoder, 2nd decoder will be disabled.

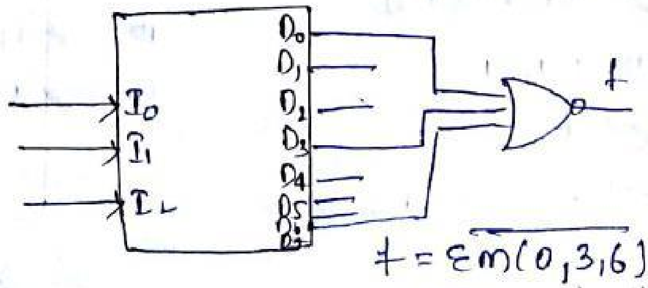
B.C.D to 7-Segment Display:-

B.C.D Data				Output						
A_3	A_2	A_1	A_0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	1
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1



Problems on Decoders:-

01. $f(x_2, x_1, x_0) = ?$



(a) $\pi(1,2,4,5,7)$

(b) $\Sigma(1,2,4,5,7)$

(c) $\Sigma(0,3,6)$

(d) None

$f = \overline{\Sigma m(0,3,6)}$

$= \Pi M(1,2,4,5,7)$

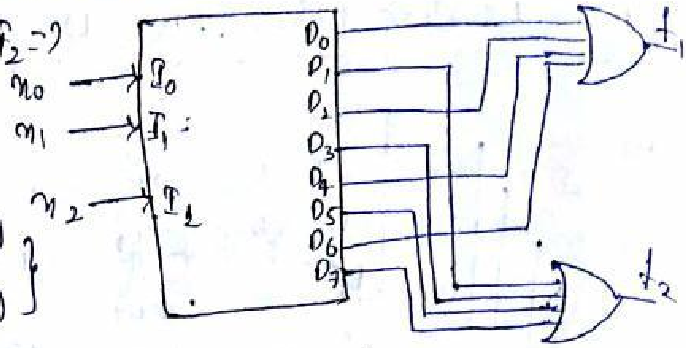
Q2) find f_1 & f_2 & $f_1 \times f_2$ = ?

(a) $x_0 x_1 x_2$

(b) $x_0 \oplus x_1 \oplus x_2$

(c) 1

~~(d)~~ 0



$f_1(0,2,4,6)$
$f_2(1,3,5,7)$ }

Product of complimentary functions are zero.

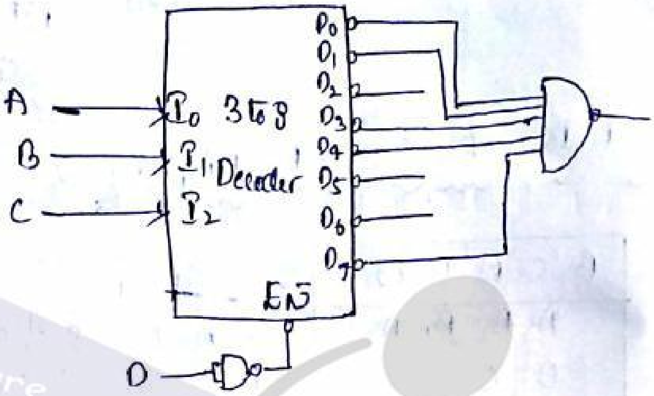
Q3) The logic circuit which is shown in fig implements

(a) $D[A \odot C + \bar{A}\bar{C}]$

(b) $D[B \oplus C + \bar{A}\bar{C}]$

(c) $D[B \oplus C + \bar{A}\bar{B}]$

~~(d)~~ $D[B \odot C + \bar{A}\bar{B}]$



$\overline{\Sigma m(0,1,3,4,7)}$

$= \Pi M(2,5,6,8)$

Solving by using 3-variable K-map

A \ BC	00	01	11	10
0	1 [0]	1 [1]	1 [3]	[2]
1	1 [4]	[5]	1 [7]	[6]

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

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

The building block shown in fig is a active high o/p decoder.

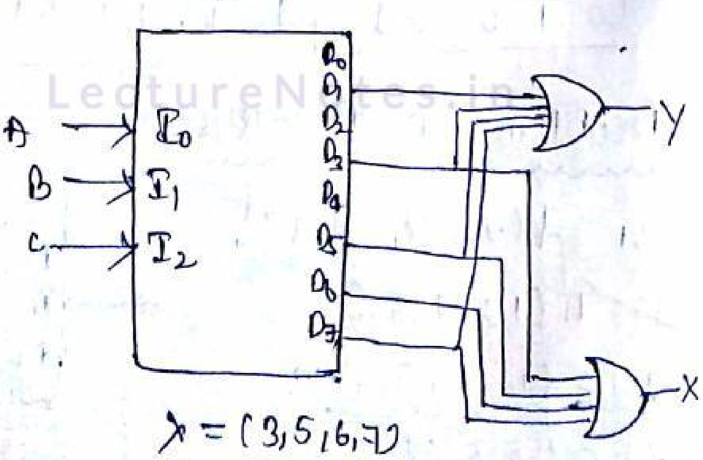
(i) The o/p x is

~~(a)~~ AB+BC+CA

(b) A+B+C

(c) ABC

(d) None of the above



$X = (3,5,6,7)$
$Y = (1,3,5,7)$

(ii) The o/p Y is

(a) A+B (b) B+C

(c) C+A ~~(d)~~ None of the above

Solving for X

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

$X = AC + BC + AB$

Solving for Y

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

$Y = C$ //

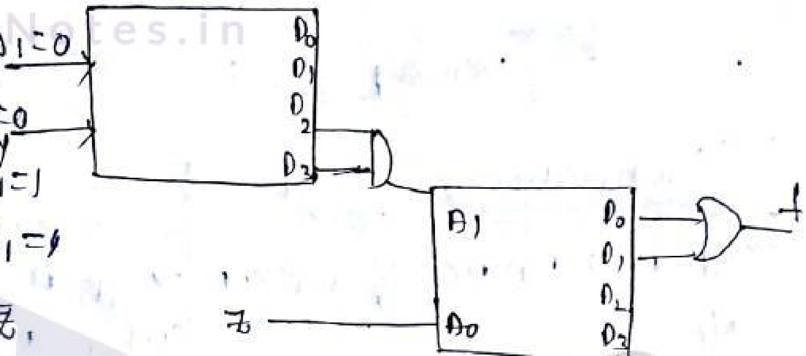
The logic circuit consists of 2 2 to 4 decoders as shown. The output of the decoder are

(i) $D_0 = 1$ when $A_0 = 0$ & $A_1 = 0$

(ii) $D_1 = 1$ when $A_0 = 1$ & $A_1 = 0$

(iii) $D_2 = 1$ when $A_0 = 0$, $A_1 = 1$

(iv) $D_3 = 1$ when $A_0 = 1$, $A_1 = 1$



$D_0 = \bar{A_0}\,\bar{A_1}$; $\Rightarrow$ 1, $\bar{Z}$,

$D_1 = A_0 \bar{A_1}$;

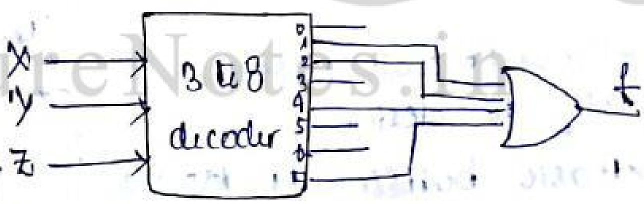
$D_2 = \bar{A_0} A_1$;

$D_3 = A_0 A_1$

The values of $f(X, Y, Z)$ are
(a) 0 (b) Z (c) $\bar{Z}$ ~~(d) 1~~

The circuit shown below is
(a) Full adder ~~(b)~~ 3 i/p X-OR gate (c) Half adder (d) 3 i/p X-NOR

Solution



$\Sigma m(1, 2, 4, 7)$

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

$f = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$
$= \bar{A}(\bar{B}C + B\bar{C}) + A(\bar{B}\bar{C} + BC)$
$= \bar{A}(B \oplus C) + A(\overline{B \oplus C})$
$= A \oplus B \oplus C \rightarrow$ sum of full adder

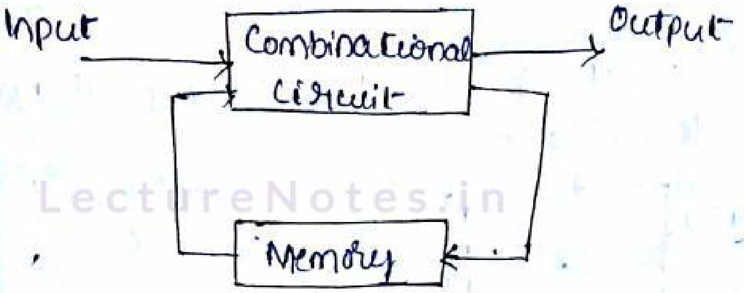
We have no information about carry, $\therefore$ We can't justify the circuit as full adder so our answer is 3 i/p X-OR

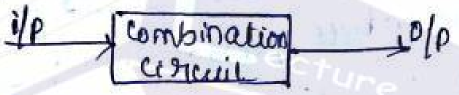
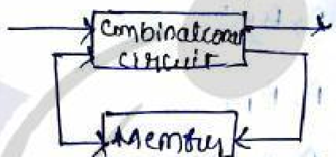
06/3/19

Synchronous Sequential Circuits.

Sequential Circuit:-

⇒ Sequential circuits are the circuits where the o/p is a function of present input and past i/p and o/p values



Combinational circuit	Sequential circuit
⇒ o/p is a function of present i/p	⇒ o/p is a function of present and past i/p & o/p.
⇒ 	⇒ 
⇒ Memory element is absent	⇒ Memory element is present
⇒ feedback does not exist	⇒ feedback exists.
⇒ combinational circuits are faster	⇒ sequential circuits are slower
⇒ Combinational circuits are easy to design	⇒ Difficult to design
⇒ Basic building blocks are logic gates	⇒ Basic buildings block are flip flop.
⇒ Combinational circuits are time independent	⇒ Sequential circuits are time dependent
⇒ Ex:- adders, Multiplexers, Demultiplexers, Encoders	⇒ Ex: flip flops, Counter, Registers

Sequential Circuits are classified as:-

01. Synchronous Sequential circuit → controlled by clock
02. Asynchronous Sequential circuit → does not use any clock.

* Latch and flipflop:~

⇒ flipflop is a basic memory element which is used to save the binary data.

⇒ Latch saves the data in asynchronous manner

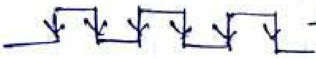
⇒ flipflop saves the data in synchronous manner.

⇒ flip flop has clock.

⇒ Latch doesnt have clock.

→ positive edge trigger

⇒ Latch is level triggered

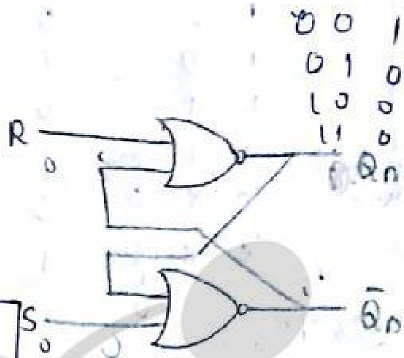


→ -ve edge trigger

⇒ flip-flop is edge triggered

* S-R Latch:~

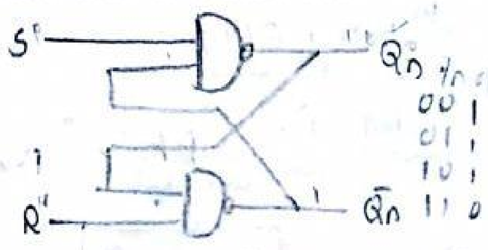
S → Set(1) R → Reset(0)



S-R latch using NOR gate

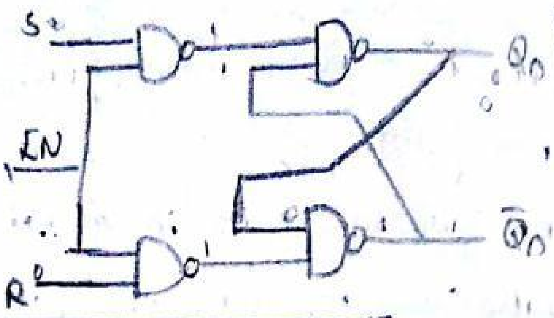
Inputs S R	Present state Q_n	Next stat Q_{n+1}	state
0 0	0	0	No change
0 0	1	1	
0 1	0	0	Reset
0 1	1	0	
1 0	0	1	Set
1 0	1	1	
1 1	0	X	Invalid
1 1	1	X	

S-R Latch using NAND Gates:~



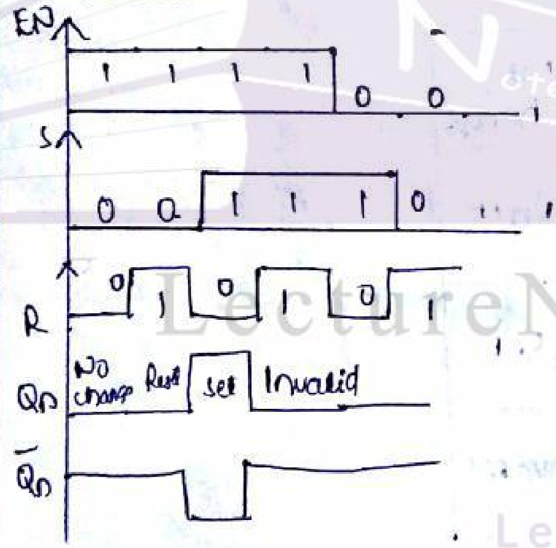
I/P S R	Prest Q_n	Next state Q_{n+1}	State
0 0	0	X	Invalid
0 0	1	X	
0 1	0	1	Set
0 1	1	1	
1 0	0	0	Reset
1 0	1	0	
1 1	0	0	No change
1 1	1	1	

* Gated S-R Latch:-



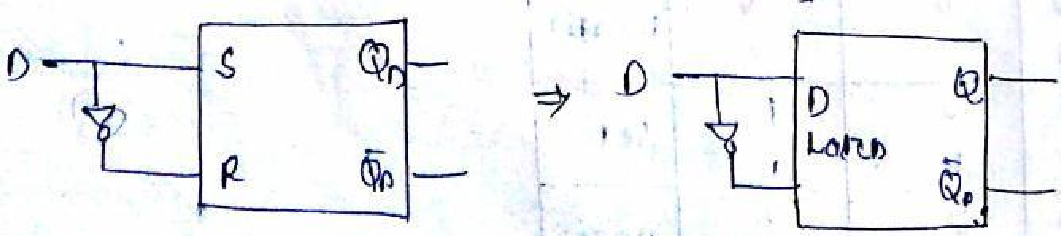
Enable EN	Inputs S R	Present state Q_n	Next state Q_{n+1}	state
1	0 0	0	0	No change
1	0 0	1	1	
1	0 1	0	0	Reset
1	0 1	1	0	
1	1 0	0	1	set
1	1 0	1	1	
1	1 1	0	X	Invalid
1	1 1	1	X	
0	X X	0	0	No change
0	X X	1	1	

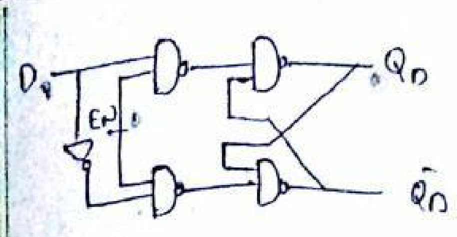
Waveforms.



8^{th} March 2019

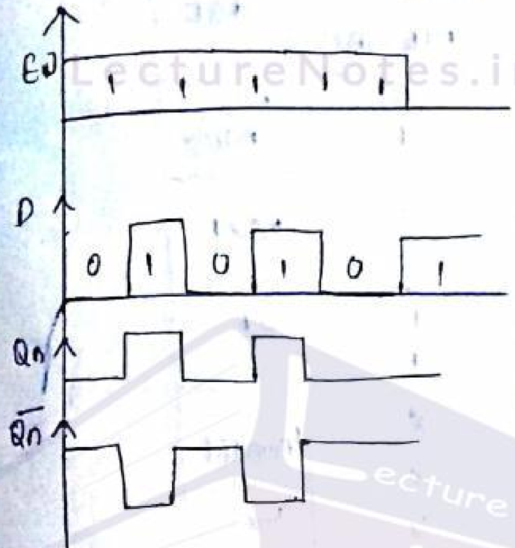
Gated D-Latch:- [D→ Delay, D→ Data]





Enable EN	Input D	Present state Q_n	Next state Q_{n+1}	State
1	0	0	0	Reset
1	0	1	0	
1	1	0	1	Set
1	1	1	1	
0	x	0	0	No change
0	x	1	1	

Waveforms:-



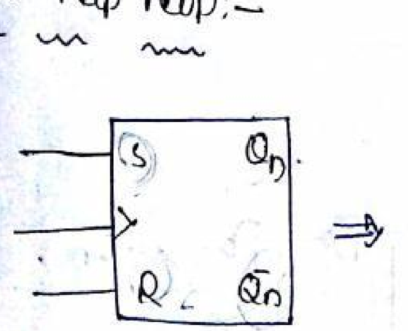
flip flop:- [JK flip flop → Jack Kilby]

There are 4 types of flip flops

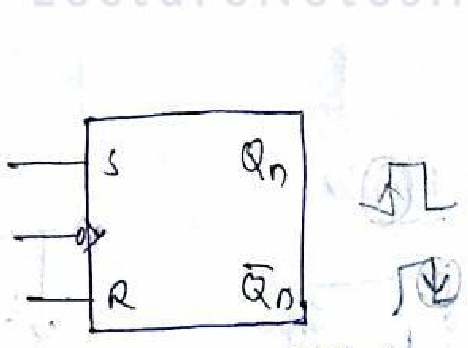
01) S R flip flop
02) D flip flop
03) JK flip flop
04) T flip flop. [T → Toggle (inversion of bits)]

01. Truth table
02. characteristic Equation
03. characteristic Table
04. Excitation Table.

01) SR flip flop:-

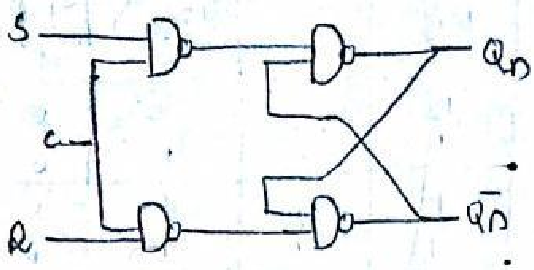


⇒



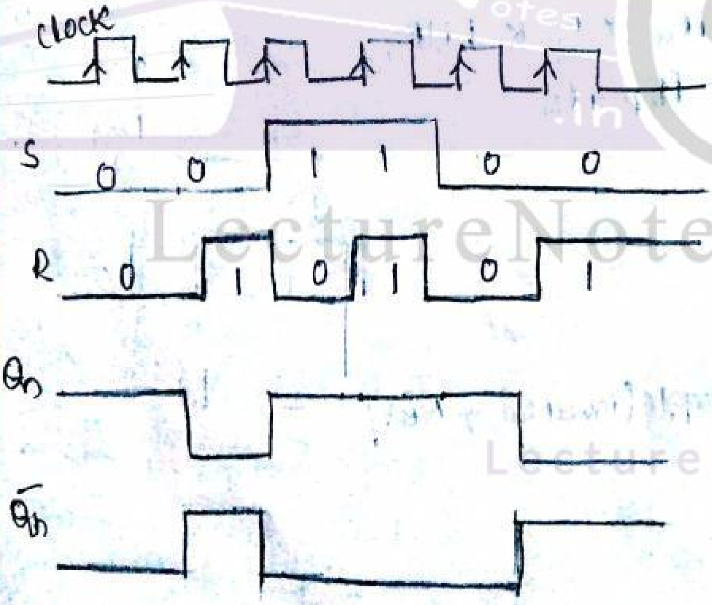
Positive edge triggered SR flip flop.

negative edge triggered SR flip flop

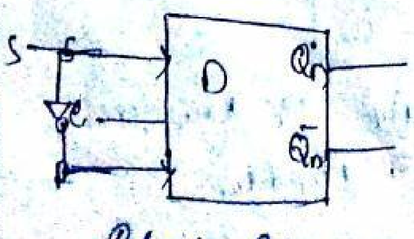


Circuit Diagram.

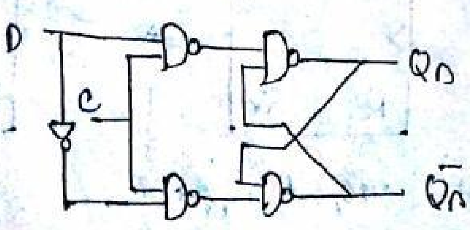
Clock C	Inputs S	R	Present state Q_n	Next state Q_{n+1}	State
↑	0	0	0	0	No change
↑	0	0	1	1	
↑	0	1	0	0	Reset
↑	0	1	1	0	
↑	1	0	0	1	Set
↑	1	0	1	1	
↑	1	1	0	X	Invalid
↑	1	1	1	X	
↓	X	X	0	0	No change
↓	X	X	1	1	



D- flip Flop:-



Block Diagram

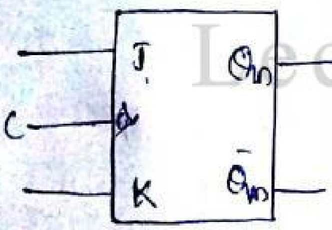


Logic Diagram.

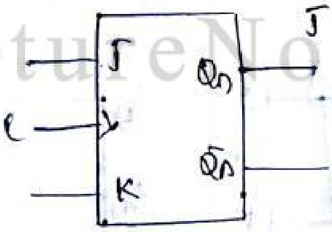
Clock C	Inputs S R	Present state Q_n	Next state Q_{n+1}	State
↑	0	0	0	Reset
↑	0	1	0	
↑	1	0	1	Set
↑	1	1	1	
↓	X	0	0	No change
↓	X	1	1	

13/03/19

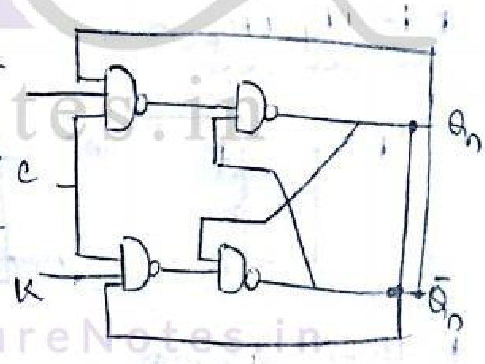
JK Flipflop:-



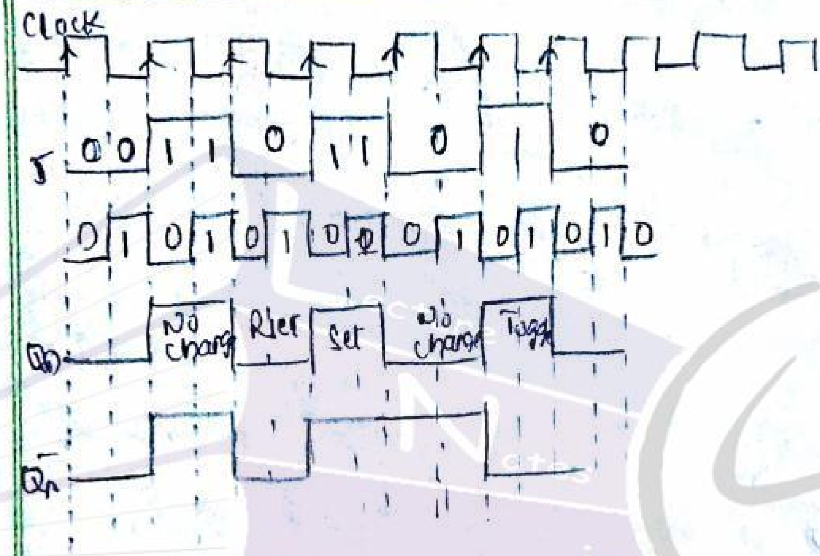
Negative Edge triggered



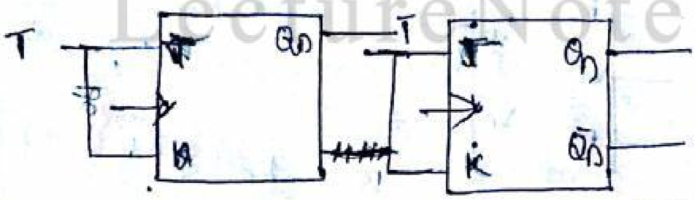
Positive Edge Triggered

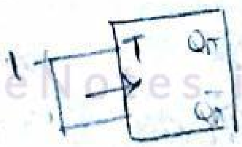


Clock C	Inputs J	Inputs K	Present state Q_n	Next state Q_{n+1}	state
↑	0	0	0	0	No change
↑	0	0	1	1	
↑	0	1	0	0	Reset
↑	0	1	1	0	
↑	1	0	0	1	Set
↑	1	0	1	1	
↑	1	1	0	1	Toggle
↑	1	1	1	0	
↓	X	X	0	0	No change
↓	X	X	1	1	

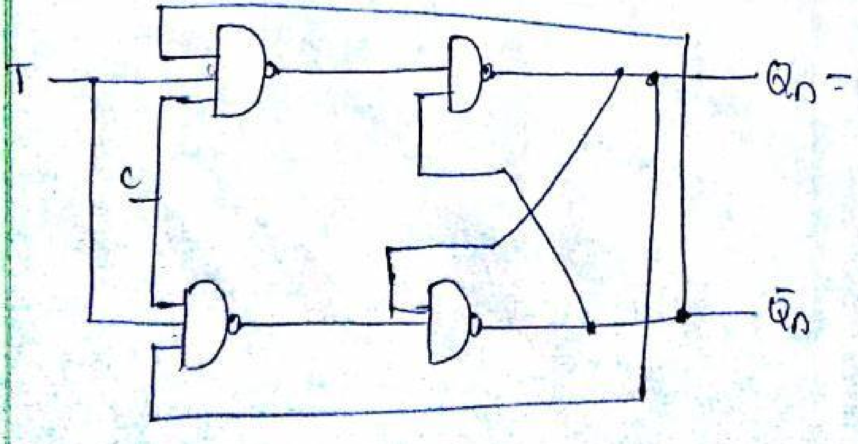


* T- Flip Flop:~





* Logic Diagram:~

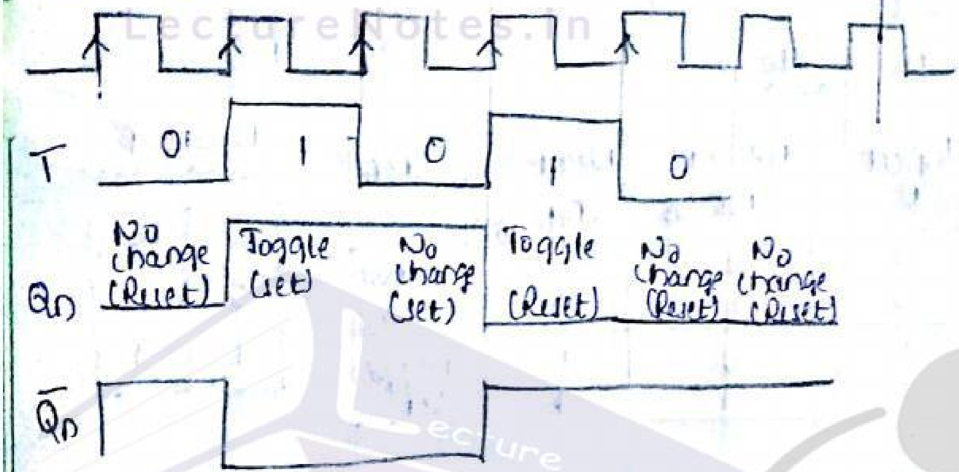


1 0 → Reset

0 1 - Set

Clock	Input	Present state Q_n	Next state Q_{n+1}	State
↑	0	0	0	No change
↑	0	1	1	
↑	1	0	1	Toggle
↑	1	1	0	
↓	X	0	0	No change.
↓	X	1	1	

Timing Diagrams



SR- Flipflop:-

Characteristic Table:-

Clock C	Inputs S R	Present state Q_n	Next state Q_{n+1}	State
↑	0 0	0	0	No change
↑	0 0	1	1	
↑	0 1	0	0	Reset
↑	0 1	1	0	
↑	1 0	0	1	set
↑	1 0	1	1	
↑	1 1	0	X	Invalid
↑	1 1	1	X	

Truth Table:-

S R	Q_{n+1}
0 0	Q_n
0 1	0
1 0	1
1 1	X

Characteristic Equation:-

S \ RQ_n	00	01	11	10
0		1		
1	1	1	X	X

$$\boxed{Q_{n+1} = S + \bar{R}Q_n}$$

Excitation Table:

Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Characteristic equation for JK flip flop

J \ KQ_n	00	01	11	10
0		1		
1	1	1		1

$$\boxed{Q_{n+1} = J\bar{Q_n} + \bar{K}Q_n}$$

D-Flip flop:

Characteristic Table:

Clock	Input D	Present state Q_n	Next state Q_{n+1}	state
↑	0	0	0	Reset / change
↑	0	1	0	
↑	1	0	1	Toggle set
↑	1	1	1	
↑	X			
↑	X			

Truth Table:

D	Q_{n+1}
0	0
1	1

Characteristic Equation:

D \ Q_n	0	1
0	0	0
1	1	1

$$\boxed{Q_{n+1} = D}$$

Excitation Table:

Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

00	0
01	1
10	1
11	0

JK - Flip Flop:

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

01. Characteristic Table

Clock	Input J	K	Present state Q_n	Next state Q_{n+1}	State
↑	0	0	0	0	No change
↑	0	0	1	1	
↑	0	1	0	0	Reset
↑	0	1	1	0	
↑	1	0	0	1	set
↑	1	0	1	1	
↑	1	1	0	1	Toggle
↑	1	1	1	0	

Truth Table:

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\bar{Q_n}$