STUDY MATERIAL

SUBJECT : Digital Logic Design (DLD)

[Module - I]

SEMESTER : 3^{RD}

BRANCH : COMPUTER SCIENCE AND ENGINEERING (CSE)

FACULTY : Mrs B Mohapatra

Department Of Electronics and Telecommunication

Scanned with CamScanner

D-24-07-2020

Subj-DLD

Number System in digital electronics :-

→ In digital electronics, the number system is used for representing the information.

→ Each number system (N.S) is expressed by different base or radix which is defined as the total number of the digit used in the N.S.

→ For example — if the N.S representing the digit from 0-9 then the base of the system is 10.

Types of Number System :-

1. Binary Number System
   ⇒ Base-2
   ⇒ Numbers (0,1)
   Ex:- $(10101)_2$
2. Decimal Number System
   ⇒ Base-10
   ⇒ Numbers (0-9)
   Ex:- $(98)_{10}$
3. Octal Number System
   ⇒ Base-8
   ⇒ Numbers (0-7)
   Ex:- $(672)_8$
4. Hexadecimal Number System
   ⇒ Base-16
   ⇒ Numbers (0-9, A-F)
   Ex:- $(A2)_{16}$

~~Conversion of Binary to octal :-~~

2020/9/9 15

Scanned with CamScanner

* Table shown below the decimal, binary, octal and hexadecimal numbers from 0 to 15 :

Decimal	Binary	octal (0-7) 8	Hexadecimal (0-15) 16
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

16 = 4 × 2

Conversion of Number system :-

Decimal to Binary NS :-

Step1 :- Divide the no. by 2 and find quotient and remainder.

Step2 :- Divide the quotient by 2 and find the next quotient and remainder.

Step3 :- Repeat the step-2 until the quotient becomes 0 or 1.

Step4 :- Write the remainders from bottom to top to find the answer.

Scanned with CamScanner

Ex:- $(27)_{10} = (?)_2$

2	27	
2	13	1
2	6	1
2	3	0
2	1	1
	0	1

$(27)_{10} = (11011)_2$

Ex. $(0.8125)_{10} = (?)_2$

$= (0.1101)_2$

$$
\begin{array}{r}
0.8125 \\
\times \quad 2 \\
\hline
1.6250 \\
\times \quad 2 \\
\hline
1.2500 \\
\times \quad 2 \\
\hline
0.5000 \\
\times \quad 2 \\
\hline
1.0000
\end{array}
$$

Ex $(0.432)_{10} = (?)_2$

$= (0.011)_2$

$$
\begin{array}{r}
0.432 \\
\times \quad 2 \\
\hline
0.864 \\
\times \quad 2 \\
\hline
1.728 \\
\times \quad 2 \\
\hline
1.456
\end{array}
$$

2020/9/9 15:45

Scanned with CamScanner

H.T

Q $(46.345)_{10} = (\ ?\ )_2$ $(101110.01011)_2$

$2 \lfloor 46$ $\qquad 0.345 \times 2 =$

Binary to Decimal conversion :-

Ex:- $(\overset{3}{1}\overset{2}{1}\overset{1}{0}\overset{0}{1})_2 = (\ ?\ )_{10}$

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

$= 8 + 4 + 0 + 1$

$= (13)_{10}$ (Ans)

Ex $(0.\overset{-1}{1}\overset{-2}{0})_2 = (\ ?\ )_{10}$

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

$= 1\times\frac{1}{2^1} + 0\times\frac{1}{2^2}$

$= 0.5 + 0$

$= (0.5)_{10}$ (Ans).

H.T $(\overset{4}{1}\overset{3}{0}\overset{2}{1}\overset{1}{0}\overset{0}{1}.\overset{-1}{1}\overset{-2}{1})_2 = (\ ?\ )_{10}$

$= (21.75)_{10}$ ✓. $\qquad 46.3$

D.25.07-2020

Decimal to octal conversion :-

Ex $(112)_{10} = (\ ?\ )_8$

8	112	
8	14	0
8	1	6
	0	1

$(112)_{10} = (160)_8$

2020/9/9 15:46

Scanned with CamScanner

Ex:- $(0.53)_{10} = (?)_8$

$$= \begin{array}{r} 0.53 \\ \times \quad 8 \\ \hline 4.24 \\ \times \quad 8 \\ \hline 1.92 \\ \times \quad 8 \\ \hline 7.36 \\ \times \quad 8 \\ \hline 2.88 \end{array}$$

$= (0.4172)_8$

H.T:- $(22.62)_{10} = (?)_8$

Octal to Decimal :-

Ex:- $(127.4)_8 = (?)_{10}$

$= 1\times8^2 + 2\times8^1 + 7\times8^0 + 4\times8^{-1}$

$= 64 + 16 + 7 + 4\times\frac{1}{8}$

$= (87.5)_{10}$

$4\times\frac{1}{8} = 0.5$

HT:- ~~$(64.08)_8 = (?)_{10}$~~

$(64.24)_8 = (?)_{10}$

Decimal to Hexadecimal :-

Ex:- $(402.3)_{10} = (?)_{16} = (192.4C)_{16}$

$$\begin{array}{r|l|l} 16 & 402 & \\ \hline 16 & 25 & 2 \\ \hline 16 & 1 & 9 \\ \hline & 0 & 1 \end{array}$$

$(402)_{10} = (192)_{16}$

$(0.3)_{10} = (0.4C)_{16}$

$$\begin{array}{r} 0.3 \\ \times \quad 16 \\ \hline 4.8 \\ \times \quad 16 \\ \hline 12.8 \end{array}$$

Hexadecimal to Decimal :-

Ex :- $(\overset{3}{B}\overset{2}{6}\overset{1}{5}\overset{0}{F})_{16} = (?)_{10}$

$= B\times16^{3} + 6\times16^{2} + 5\times16^{1} + F\times16^{0}$

$= 11\times16^{3} + 6\times16^{2} + 5\times16^{1} + 15\times16^{0}$

$= (46687)_{10}$

HT :- $(A25.16)_{16} = (?)_{10}$

Binary to octal :-

Ex $(001\ 111\ 000\ 111\ 001)_2 = (?)_8$

$1 \quad 7 \quad 0 \quad 7 \quad 1$

$2^{3} \leftarrow$ no. of bits $= 8$

$= (17071)_8$

Ex $(001\ 011\ 100\ 101\ 110)_2 = (?)_8$

$1 \quad 3 \quad 4 \quad 5 \quad 6$

$(13456)_8$

		8	16
000	0	000	0000
001	1 → $8 = 2^3$	001	0001
010	2	010	0010
011	3	011	0011
100	4	100	0100
101	5	101	0101
110	6	110	0110
111	7	111	0111
1000	8		

Octal to Binary

$(273)_8 = (?)_2$

$(010\ 111\ 011)_2$

H.T $(75671)_8 = (?)_2$

Binary to Hexadecimal

$(1100\ 1101\ 0011)_2 = (?)_{16}$

C D 3

$(CD3)_{16}$

H.T. $(011111\ 11000010)_2 = (?)_{16}$

Hexa to Binary:

H.T $(A52)_{16} = (?)_2$

Scanned with CamScanner

Solution

H.T. Q. $(46.345)_{10}$

Sol^n:

$$
\begin{array}{r|l|l}
2 & 46 & \\
2 & 23 & 0 \\
2 & 11 & 1 \\
2 & 5 & 1 \\
2 & 2 & 1 \\
2 & 1 & 0 \\
 & 0 & 1
\end{array}
$$

$$
\begin{array}{r}
0.345 \\
\times\ \ \ \ 2 \\
\hline
0.690 \\
\times\ \ \ \ 2 \\
\hline
1.380 \\
\times\ \ \ \ 2 \\
\hline
0.760 \\
\times\ \ \ \ 2 \\
\hline
1.520
\end{array}
$$

$\cancel{(46)_{10} = (111000)_2}$

$(46)_{10} = (101110)_2$ $\qquad$ $(0.345)_{10} = (0.0101)_2$

$\therefore (46.345)_{10} = (\overset{101110}{\cancel{111000}}.0101)_2$ Ans

H.T. Q $(\overset{4\,3\,2\,1\,0}{10101}.\overset{-1\,-2}{11})_2 = (\ ?\ )_{10}$

Sol^n

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

$= 16 + 0 + 4 + 0 + 1 + 0.5 + 0.25$

$= 21 + 0.75$

$= (21.75)_{10}$ Ans

H.T. Q $(22.62)_{10} = (\ ?\ )_8$

Sol^n

$$
\begin{array}{r|l|l}
8 & 22 & \\
8 & 2 & 6 \\
 & 0 & 2
\end{array}
$$

$(22)_{10} = (26)_8$

$$
\begin{array}{r}
0.62 \\
\times\ \ \ 8 \\
\hline
4.96 \\
\times\ \ \ 8 \\
\hline
7.68 \\
\times\ \ \ 8 \\
\hline
5.44
\end{array}
$$

$(0.62)_{10} = (0.475)_8$

$\therefore (22.62)_{10} = (26.475)_8$ Ans

Scanned with CamScanner

H.T. Q $(64.24)_8 = (?)_{10}$

Solⁿ

$= 6\times8^1 + 4\times8^0 + 2\times\frac{1}{8} + 4\times\frac{1}{8^2}$

$= 48 + 4 + \frac{1}{4} + 4\times\frac{1}{8\times8}$

$= 48 + 4 + 0.25 + 0.625$

$= 52 + 0.875$

$= (52.875)_{10}$ Ans

H.T. Q $(A25.16)_{16} = (?)_{10}$

Solⁿ

$= A\times16^2 + 2\times16^1 + 5\times16^0 + 1\times\frac{1}{16} + 6\times\frac{1}{16\times16}$

$= 10\times256 + 32 + 5 + 0.0625 + 0.0234375$

$= 2560 + 37 + 0.0859375$

$= (2597.0859375)_{10}$ Ans

H.T. Q $(75671)_8 = (?)_2$

Solⁿ

$(111\ 101\ 110\ 111\ 001)_2$ Ans

H.T. Q $(0111111100 0010)_2 = (?)_{16}$

Solⁿ

$(0001\ 1111\ 1100\ 0010)_2$

$= (1FC2)_2$ Ans

H.T. Q $(A52)_{16} = (?)_2$

Solⁿ $(1010\ 0101\ 0010)_2$ Ans

2020/9/9 15:46

Scanned with CamScanner

D. 27.07.2020

Octal to Hexadecimal

Ex $(27)_8 = (\ )_2 = (17)_{16}$

27

$(\underbrace{0\,0}_{}1\underbrace{0\,111}_{})_2 \to (\ )_{16}$

$(\underbrace{0001}\,\underbrace{0111})_2$

1 7

Ex $(356)_8 = (\ )_{16} = (0EE)_{16}$

$\to (\ )_2$

H.T

~~0011~~ $(011\,101\,110)_2 \to (0000\ 1110\ 1110)$ — 0 E E

Ans

Hexadecimal to octal :-

Ex $(589)_{16} = (\ )_8 = (2611)_8$

$\to (\ )_2$

$= (010\,110\,001\,001)_2$

2 6 1 1

H.T :- $(A77)_{16} = (\ )_8 = (5167)_8$

$(101\,001\,110\,111)_2$

5 1 6 7

Ans

Binary Arithmatic :-

Binary Addition :-

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 10

carry — magnitude value

Ex

```
   1 1
   1 1 1 1  → 15
 + 1 0 1 0  → 10
 ----------
 1 1 0 0 1  → 25
```

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

= 16 + 8 + 0 + 0 + 1

Binary subtraction :-

(0, 1) (00, 01, 10, 11)

$2^2 = 4$

00, 01, 10, 11

0 − 0 = 0
1 − 0 = 1
1 − 1 = 0
0 − 1 = 1 ⟹ 10 − 1 = 1

Ex :-

```
   1 1 0 1
 − 1 0 0 1
 ---------
   0 1 0 0
```

Ex

```
   1 0 0 1
 − 0 1 1 1
 ---------
   0 0 1 0
```

B. Multiplication :-

0 × 0 = 0
0 × 1 = 0
1 × 0 = 0
1 × 1 = 1

B. Division :-

0 ÷ 1 = 0
1 ÷ 1 = 1
0 ÷ 0 = not possible
1 ÷ 0 = not possible

Ex

```
   1 0 1 0
 − 0 0 1 1
```

2020/9/9 15

Scanned with CamScanner

29/07/20

Find Radix of a no. :- (Base)

Q $(20)_6 * (22)_7 = (300)_x$. Find x ;

Sol^n :- $(2\times 6^1 + 0\times 6^0) \times (2\times 7^1 + 2\times 7^0)$

$= 3\times x^2 + 0\times x^1 + 0\times x^0$

$\Rightarrow (12+0)\times(14+2) = 3x^2 + 0 + 0$

$\Rightarrow 12\times 16 = 3x^2$

$\Rightarrow x = \sqrt{\frac{12\times 16}{3}} = \sqrt{\frac{\cancel{192}^{64}}{\cancel{3}}} = \sqrt{64}$

$\Rightarrow \boxed{x = 8}$

Q $(\sqrt{41})_x = (5)_x$. Find x ?

Sol^n :- $\sqrt{4\times x^1 + 1\times x^0} = 5\times x^0$

$\Rightarrow \sqrt{4x+1} = 5$

$\Rightarrow 4x+1 = 5^2$

$\Rightarrow 4x+1 = 25$

$\Rightarrow 4x = 25-1 = 24$

$\Rightarrow x = \frac{\cancel{24}^{6}}{4}$

$\Rightarrow \boxed{x = 6}$

H.T Q_1 — $(42)_x = (50)_{x-1}$

Q_2 — $\left(\frac{42}{5}\right)_x = (10)_x$

2020/9/9 15:47

Scanned with CamScanner

Complement of a number :-

-ve

+5 − 2

→ 1's complement

→ 2's complement

9's
10's

1's complement :-

Steps 1 - Convert each binary bit to its alternate bit.

i.e. 0 → 1
1 → 0

Ex :- $(011010)_2$
↓ 1's
$(100101)_2$

Ex :- $(1111000)_2$
↓ 1's
$(0000111)_2$

2's complement :-

Steps - 1 → Find out 1's complement of the given binary no.

2 → Add 1 to this 1's complement number.

Ex :- $(010100)_2 \xrightarrow{2's} (101100)$
↓ 1's
$(101011)_2$
+ 1
$(101100)_2$ → 2's complement

2020/9/9 15

Scanned with CamScanner

Ex :- $(1110101)_2 \xrightarrow{2's} (0001011)$

↓ 1's complement

$(0001010)_2$

$+ \quad 1$

$(0001011)_2$ (Ans)

Ex :- $(011010)_2$

↓ 2's

$(100110)_2$

1's complement subtraction :-

Steps

1 — Determine 1's complement of the given no. (–ve form). (Subtrahend)

2 — Add this no. with the other number (minuend).

3 — If a carry is generated then add this carry.

4 — If no carry is generated then find the 1's complent with a –ve sign.

Ex :- Subtract $(1010)_2$ from $(1111)_2$ using 1's complement

Sol^n

1111 → minuend

– 1010 → subtrahend

2020/9/9 15:47

Scanned with CamScanner

$$\begin{array}{r} 1111 \\ -1010 \\ \hline \end{array}$$

1's complement of $(1010) = (0101)_2$

$$\begin{array}{r} 1111 \\ +0101 \\ \hline 1,0100 \end{array}$$

carry

$$\begin{array}{r} 0100 \\ +\quad 1 \\ \hline (0101)_2 \end{array}$$ Ans

Ex $(1010)_2$ from $(1000)_2$

Sol^n $(1000)_2 - (1010)_2$

$\quad 8 \qquad\quad 10$

$$\begin{array}{r} 8 \\ -10 \\ \hline -2 \end{array}$$

1's complement of $(1010)_2 = (0101)_2$

Add

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

1's complement of $(1101)_2 = -(0010)_2$

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

$= 0 + 0 + 2 + 0$

$= -2$

Ans

Ex

H.T Q $(0001)_2 - (1011)_2 \to$ 1's

2020/9/9 15

Scanned with CamScanner

D. 31.07.2020

2's complement subtraction :-

Step 1 – Determine the 2's complement of the minuend.

2 – Add this with the other n

3 – If carry generated then omit the carry.

4 – If no carry generated then the result is in 2's complement form.

– To get the true form take 2's complement again with a –ve sign.

Ex :- $(1010)_2$ from $(1000)_2$ 2's

Soln :-

```
  1000 → 8  subtrahend
– 1010 ← minuend       10
```

2's complement of $(1010)_2$

$= (0110)_2$

$8 - 10 = -2$

```
  1000
+ 0110
------
 (1110)
```

true result/form — 2's compl. of $(1110)_2 = -(0010)_2$

$\downarrow$

$1 \times 2^1 = -2$

H.T :- $\cancel{(0011)_2}$

$(1011)_2 - (0011)_2$

$11 - 3 = +8$

Sol^n $(1011) - (0011)$
MSB LSB MSB LSB

2's $\rightarrow (1101)_2$

Add $\rightarrow$

$$\begin{array}{r} 1011 \\ +1101 \\ \hline \cancel{1}\,1000 \end{array}$$

carry

so. the answer is $(1000)_2$ by omitting the carry.

H.W.

Q1 $(1100)_2 - (0101)_2$

Q2 $(0010)_2 - (1.0001)_2$

2's.

Signed Binary Representation

−ve numbers

$\boxed{1 \text{ byte} = 8 \text{ bits}}$

If a binary no. is represent in 8 bit format then we have two parts —

1. Sign part.
2. Magnitude part.

MSB	7
sign	Magnitude

MSB Most significant bit

LSB (least significant bit)

ex:- +3 → 0,0000011
sign Mag

(+1,)

$-4 \rightarrow (1,0000100)_2$

$\downarrow$

(100)

Rule :- $+ve \rightarrow$ sign bit '0'

$-ve \rightarrow$ " '1'

Addition in 2's complement

~~ter~~

$+\ -$

$-\ +$

$+\ +$

$-\ -$

Case 1 :- Two no.s are +ve

Ex :- +19 and +22

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

$19 = (10011)_2$

2	22	
2	11	0
2	5	1
2	2	1
	1	0

$(22) = (10110)_2$

$+19 = 0,0010011$

$+22 = 0,0010110$

$+41 \quad \underset{\text{sign bit}}{0}, \underbrace{0101001}_{\text{mag.}}$

$\rightarrow 1\times2^5 + 1\times2^3 + 1\times2^0$

$= 32 + 8 + 1$

$= +41$

Case 2 :- +ve and -ve.

Ex :- +39 and -22

2	39	
2	19	1
2	9	1
2	4	1
2	2	0
	1	0

2	22	
2	11	0
2	5	1
2	2	1
	1	0

$(39) = (100111)_2$

$(22) = (10110)_2$

$+39 = (0,0100111)$

$+22 = (0,0010110)_2$

$\downarrow$ 2's

$-22 = (1,1101010)_2$

$$
\begin{array}{r l}
 & 0,0100111 \longrightarrow +39 \\
+ & 1,1101010 \longrightarrow -22 \\
\hline
1, & 0,\underbrace{0010001}_{\text{Mag}} \quad\quad +17
\end{array}
$$

carry sign

Ans :- $\begin{array}{c} \quad\ \ 6\,5\,4\,3\,2\,1\,0 \\ (0,0010001) \end{array}$

$\downarrow$

$1\times2^4 + 1\times2^0$

$= 16+1$

$= +17$

2020/9/9 15:4

Scanned with CamScanner

01/08/2020

DLD

case-3 :- −ve & +ve

Ex:- −47 & +29

2	47	
2	23	1
2	11	1
2	5	1
2	2	1
	1	0

2	29	
2	14	1
2	7	0
2	3	1
	1	1

47 = (101111)

29 = (11101)

+47 = 0,0101111

+29 = 0,0011101

−47 = 1,1010001

+29 = 0,0011101

−18 = 1,$\underbrace{1101110}_{\text{magnitude}}$

1 ← sign

2's

$\overset{}{1},\overset{6}{0}\overset{5}{0}\overset{4}{1}\overset{3}{0}\overset{2}{0}\overset{1}{1}\overset{0}{0} \rightarrow 1\times2^4 + 1\times2^1 = 16 + 2 = 18$

Case 4 -ve & -ve.

Ex :- -32 & -44

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

2	44	
2	22	0
2	11	0
2	5	1
2	2	1
	1	0

$32 = (100000)_2$

$44 = (101100)_2$

$+32 = (0,0100000)$

$+44 = (0,0101100)$

2's, 32 = -32 = 1,1100000

2's, 44 = -44 = + 1,1010100

$$\begin{array}{r} -32 \\ -44 \\ \hline -76 \end{array}$$

1, 1, 0110100

carry (omit)

sign

magnitude

2's

6 5 4 3 2 1 0

1001100

$-1\times 2^6 + 1\times 2^3 + 1\times 2^2 = -76$

Scanned with CamScanner

Ex:-

$+29, -33, = -4$

$+29 = (11101)_2 = (0,0011101)$

$+33 = (100001)_2 = (0,0100001)$

2's compl $33 = -33 = (1,1011111)_2$

$$\begin{array}{r} 0,0011101 \\ 1,1011111 \\ \hline 1,1111100 \end{array}$$

sign: mag

$\downarrow$ 2's

00000100

$\rightarrow 2^2 \times 1 = -4$

Ex:

HT -25, -27

Coding:-

→ Used for Security of data transmission.

Tx — Rx

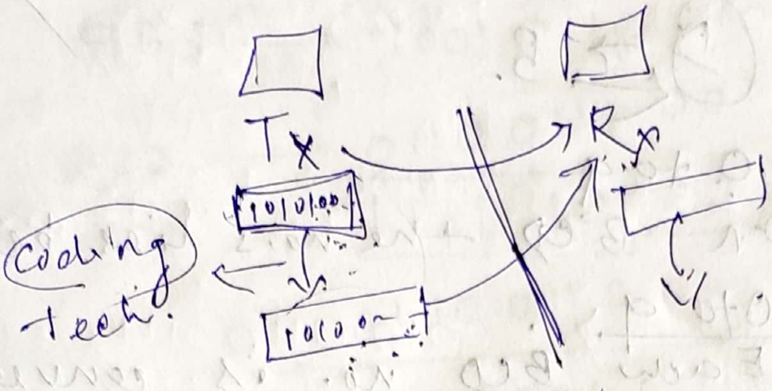
9M

01,1101

Algorithm diff form.

D. 05. 08. 2020

Coding :-



defⁿ :- process in dig. comᵐ where the original informatⁿ is coded to a different format with the help of different codes.

Advantage :-

→ Restrict unauthorised access of the user.

→ Reliable o/p i.e. it maintain both privacy & security of code.

8421

Types :-

1) → Weighted code
- → BCD code
- → 8421 code
- → 84-2-1 code
- → 2421 code

2) → Non-weighted code
- → Excess -3
- → Gray code
- → Binary code

3) → Alphanumeric code
- → ASCII code
- → EBCDIC

4) → Error correction and Detection code.

Scanned with CamScanner

BCD code :—

(Binary Coded Decimal) code

— D → B.

— 0 to 9

— In BCD the no.s lies betn 0 to 9.

→ Each BCD no. is converted to Binary no.s of 4 bit each.

Ex:— $16 = 2^4$ (16) 0 – 15

0101 10

$(52)_{10}$ Find its BCD code.

$(0101\ 0010)_{BCD}$

Ex:— $(843)_{10}$ (88.24)

$(1000\ 0100\ 0011)_{BCD}$

BCD Addition :—

Steps:— 1. Find the BCD of the two nos.

+ Augend
Addend

2. Add these two nos.

3. If the sum is ≤ 9 (Lesstran equals) then it is a valid no.

4. If the sum is > 9 then it is an invalid no. So, to correct it add 6 (0110) to it.

2020/9/9 15
Scanned with CamScanner

Ex: $(965)_{10} + (672)_{10}$ Add in BCD.

Solⁿ:–

$8 \to 1000$
$7 \to 0111$
$10 \to 1010$
0110

965	$\to$	1001 0110 0101
672	$\to$	0110 0111 0010
1637	15	$\underbrace{1111}_{15}\ \underbrace{1101}_{13}\ \underbrace{0111}_{7}$
	+	0110 0110
		$\boxed{1}\ \underbrace{0110}_{6}\ \underbrace{0011}_{3}\ \underbrace{0111}_{7}$

carry.

1 6 3 7

Ans: – $(965)_{10} + (672)_{10} = (1637)_{BCD}$

Ex: – HT $(238)_{10} + (195)_{10} = (?)_{BCD}$

BCD Subtraction :–

Step: – 1. Find BCD of the two nos.
2. Subtract the two nos.
3. If the difference result is ≤ 9, then valid no.
4. If > 9, then invalid no. So, to correct it, subtract 6.

Ex: – $(38)_{10} - (15)_{10} = (?)$ BCD. $= (23)_{BCD}$

Solⁿ: –

38	$\to$	0011 1000
− 15	$\to$	0001 0101
23		$\underbrace{0010}_{2}\ \underbrace{0011}_{3}$

2020/9/9 15:49

Scanned with CamScanner

Ex:- H.W. $(1952)_{10} + (2352)_{10}$

Ex:- H.W. $(88)_{10} - (92)_{10}$ -1

Ex 1 $(267)_{10} - (175)_{10} = (92)_{BCD}$.

Sol^n

$$
\begin{array}{rrl}
 & 267 \rightarrow & 0010\ 0110\ 0111 \\
- & 175 \rightarrow & -\ 0001\ 0111\ 0101 \\
\hline
 & (92)^{*} & \underbrace{0\ 000}_{0}\ \underbrace{1111}_{15}\ \underbrace{0010}_{2} \\
 & & \qquad\quad 0110 \\
\hline
 & & \underbrace{0\ 000}_{0}\ \underbrace{1001}_{9}\ \underbrace{0010}_{2}
\end{array}
$$

Scanned with CamScanner

07.08.2020

Excess-3 code :-

$(8)_{10} + 3 = (11)_{XS-3}$

BCD (0-9)

→ defⁿ :- To obtain the XS-3 code of any given no. then add 3 to that no. (to each digit of the no.)

Ex :- $(5\ 3)_{10}$
$+3\ +3$
$(8\ 6)$

Ex :- $(6\ 2\ 4)$
$+3\ +3\ +3$
$(9\ 5\ 7)_{XS-3}$

6 3 4
+3 +3 +3
9 A.

XS-3 Addition :-

Steps :-
1. Add the two nos by finding of XS-3 code binary format.
2. If result (grouping of 4 bits) generate carry then add 3 to it.
3. If no carry then subtract 3 to it.

Ex :- $(37)_{10} + (28)_{10}$ - Add in XS-3.

$\underset{+3\ +3}{37} \longrightarrow 6\ 10$

$\underset{+3\ +3}{28} \longrightarrow 5\ 11$

Scanned with CamScanner

$37 \rightarrow 6\ \ 10 \rightarrow$ 0110 1010

$28 \rightarrow 5\ \ 11 \rightarrow$ + 0101 1011

65 +3

+3

~~3 → 0011~~

98

no carry	1100	carry 0101
	− 0011	+ 0011
	1001	1000
	9	8

HW :- $(55)_{10} + (43)_{10}$? Add Xs-3

Xs-3 Subtraction :-

Step-1 1. Subtract two nos. by finding its XS-3, 4 bit binary format.

2. If ~~carry~~ borrow is present then subtract 3.

3. If no ~~carry~~ borrow is present then add 3.

Ex :- $(57)_{10} - (27)_{10}$ · Subtract Xs-3

Solⁿ :-

$57 \rightarrow 8\ \ 10 \rightarrow$ 1000 1010

+3 +3

− $27 \rightarrow 5\ \ 10 \rightarrow$ − 0101 1010

+3 +3

30

+3 +3

63

no	0011	0000 no
	+ 0011	+ 0011
	0110	0011
	6	3

Gray Code :—

→ Binary code

↳ 8 1000

7 0111

def^n ⇒ Special type of code in which only one bit is changed moving from one step to another.

1010001

Gray to Binary conversion :—

steps :— 1. Find the XOR or addition (without carry) of the previous result bit with the next bit.

2. Write the M.S.B. as usual & do the operation till the last bit.

XOR

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

Ex :— MSB LSB

$(1010)_G \rightarrow (\quad)_B$

$(1100)_B$

G → B

B → G

Ex :— $(001101)_G \rightarrow (\quad)_B$

$(001001)_B$

A.T :— $(0101101)_G \longrightarrow (?)_B$

Binary to Gray conversion :-

steps – 1. Write the MSB as-usual.

2. Add (without carry) or XORed, the next bit with the previous bit

3. Do the step-2 till the LSB.

Ex :- $(1101)_B \rightarrow (\ )_G$

$(1011)_G$

Ex :- $(101110)_B$

$(111001)_G$

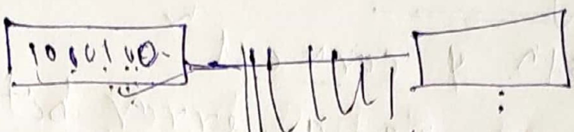
H.T :- $(0011111)_B \rightarrow (\ )_G$.

Scanned with CamScanner

08.08.2020

DLD

Error detection and Correction Code:-



1 1001101

Error detection :-

- parity bit - Extra bit added with the message bit during transmission
  1) Even parity
  2) Odd parity
- detect how many 1's are present in the msg.

Even parity :-

→ If the no. of 1's are even then even parity bit (P_e) is 0

→ If " " " odd then odd parity bit (P_e) is 1

Odd parity :-

Vice Versa of even parity.

Ex :-

msg.		P_e		P_o
1010	→	0	→	1
0001	→	1	→	0 → 00010
1110	→	1	→	0
0000	→	0	→	1

Disadvantages

→ Only single error bit identified.

Scanned with CamScanner

Error detection & error correction :-

Hamming code

Advantages
1. Detⁿ & correctⁿ
2. more no. of error bits are identified.

— 7 bit code word.

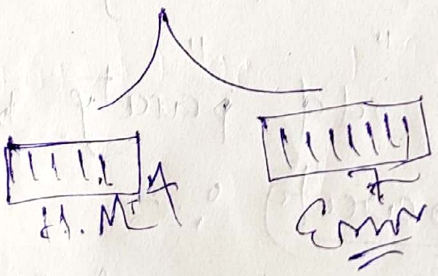
7	6	5	4 (2^2)	3	2 (2^1)	1 (2^0)
D_7	D_6	D_5	P_4	D_3	P_2	P_1

— 3 groups —

$P_1 = x = D_3 \oplus D_5 \oplus D_7$

$P_2 = y = D_3 \oplus D_6 \oplus D_7$

$P_4 = z = D_5 \oplus D_6 \oplus D_7$



Ex:- find 7 bit H.C where the data bits are (1110).

Solⁿ:- we know 7 bit H.C be -

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

Scanned with CamScanner

Now, we have to find out the x, y, z values -

$P_1 = x = D_3 \oplus D_5 \oplus D_7$

$= 0 \oplus 1 \oplus 0$

$\cancel{P_1 = 0}$ $\boxed{P_1 = 0}$

XOR

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

$P_2 = y = D_3 \oplus D_6 \oplus D_7$

$= 0 \oplus 1 \oplus 1$

$\boxed{P_2 = 0}$

$P_4 = z = D_5 \oplus D_6 \oplus D_7$

$= 1 \oplus 1 \oplus 1$

$\boxed{P_4 = 1}$

Now, the 7 bit H.C is

1 1 1 P_4 0 P_2 P_1

= (1 1 1 1 0 0 0) Ans.

Ex :- Given 7 bit H.C (0110110) Find the error bit & then correct it.

Soln :- Given

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

$x = P_1 \oplus D_3 \oplus D_5 \oplus D_7$

$= 0 \oplus 1 \oplus 1 \oplus 0$

$\boxed{x = 0}$

$y = P_2 \oplus D_3 \oplus D_6 \oplus D_7$

$= 1 \oplus 1 \oplus 1 \oplus 0$

$\boxed{y = 1}$

$z = P_4 \oplus D_5 \oplus D_6 \oplus D_7$

$= 0 \oplus 1 \oplus 1 \oplus 0$

$\boxed{z = 0}$

$x = 0, y = 1, z = 0$

So, the error bit is

$(z\,y\,x) = (0\,1\,0) = 2^{nd}$ bit (Ans)

Thus, the correct H.C. is (0110100) Ans

HW:

Q1 $(0101) \rightarrow$ Find 7 bit H.C

Q2 (0110101) - HC $\rightarrow$ Find error bit & correct.

Scanned with CamScanner

D. 10. 08. 2020

12 bit HC

				2^3				2^2		2^1	2^0
12	11	10	9	8	7	6	5	4	3	2	1
D_{12}	D_{11}	D_{10}	D_9	P_8	D_7	D_6	D_5	P_4	D_3	P_2	P_1

x =

y =

z =

(z y x)

	3	2	1	0
1 →	0	0	0	1
2 →	0	0	1	0
3 →	0	0	1	1
4 →	0	1	0	0
5 →	0	1	0	1
6 →	0	1	1	0
7 →	0	1	1	1
8 →	1	0	0	0
9 →	1	0	0	1
10 →	1	0	1	0
11 →	1	0	1	1
12 →	1	1	0	0

$P_1 = x = 3 \oplus 5 \oplus 7 \oplus 9 \oplus 11.$

$P_2 = y = 3 \oplus 6 \oplus 7 \oplus 10 \oplus 11$

$P_4 = z = 5 \oplus 6 \oplus 7 \oplus 12$

$P_8 = K = 9 \oplus 10 \oplus 11 \oplus 12$

KZYX = (1100)

Ex 12 bit H.C is given i.e. (000011101010) Find the error bit & corrected code.

Soln:-

D_{12}	D_{11}	D_{10}	D_9	P_8	D_7	D_6	D_5	P_4	D_3	P_2	P_1
12	11	10	9	8	7	6	5	4	3	2	1
0	0	0	0	1	1	1	0	1	0	1	0

XOR

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

$x = 1 \oplus 3 \oplus 5 \oplus 7 \oplus 9 \oplus 11$

$= 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0$

$= 1$

$y = 2 \oplus 3 \oplus 6 \oplus 7 \oplus 10 \oplus 11$

$= 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 1$

$x = 4 \oplus 5 \oplus 6 \oplus 7 \oplus 12$
$= 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0$
$= 1$

$\boxed{x = 1}$ $\boxed{y = 1}$ $\boxed{z = 1}$ $\boxed{K = 1}$

$K = 8 \oplus 9 \oplus 10 \oplus 11 \oplus 12$
$= 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0$

$\boxed{K = 1}$

$Kxyz = 1111 = \underline{15}$ th Bit
12 bit
No error bit present.

Ex: Given a 8-bit data word. 01011010, 12 bit H.C.

Sol:

12	11	10	9	8	7	6	5	4	3	2	1
0	1	0	1	P_8	1	0	1	P_4	0	P_2	P_1
				0				0		0	0

$P_1 = 3 \oplus 5 \oplus 7 \oplus 9 \oplus 11$
$= 0 \oplus 1 \oplus 1 \oplus 1 \oplus 1$

$\boxed{P_1 = 0}$

$P_2 = 3 \oplus 6 \oplus 7 \oplus 10 \oplus 11$
$= 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1$

$\boxed{P_2 = 0}$

$P_4 = 5 \oplus 6 \oplus 7 \oplus 12$
$= 1 \oplus 0 \oplus 1 \oplus 0$

$\boxed{P_4 = 0}$

$P_8 = 9 \oplus 10 \oplus 11 \oplus 12$
$= 1 \oplus 0 \oplus 1 \oplus 0 \rightarrow \boxed{P_8 = 0}$

Now, the 12 bit H.C as –
(010101010000) Ans

HT:– 12 bit H.C = (101000101011)
Find error bit & correct code.

ASCII code :–

- American Standard Code for Information Interchange.
- Used in micro computer systems.
- 7 bit code.
- 1967
- $2^7 = 128$ characters

128 characters:
- 26 lowercase alphabets (a, b ...)
- 26 uppercase (A, B ..)
- 33 special letters (@, $, # ..)
- 33 control letters (*, –, ÷, ;, /, ..)
- 10 (0–9) nums

$b_7\ b_6\ b_5$	$b_4\ b_3\ b_2\ b_1$
3	4
0 0 0	0 0 0 0
0 0 1	

EBCDIC :–

- Extended Binary coded Decimal Information code.
- Devl. IBM company.
- 8 bit code
- $2^8 = 256$ characteristics.
(A–Z, a–z, 0–9, %, #, +, –, $, ...)

D. 12.08.

Logic gates :-

Add, sub, mul
AND, OR

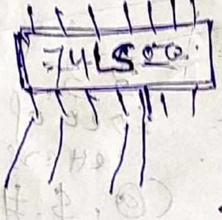
def :-

It will define the whole operation of a digital ckt and it is associated with its corresponding truth table.

Truth table :-

It is a predefined statement for each logic gate.

→ The logic gates are available as fabricated IC chip (Integrated chip).



Types :-

Gates		IC No
NAND	—	7400
NOR	—	7402
NOT	—	7404 → Inverter
AND	—	7408
OR	—	7432
XOR	—	7486
XNOR	—	74135
BUFFER	—	74125 / 74126

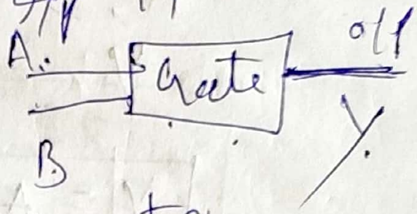
→ NAND & NOR ⇒ Universal gate

NOT, AND, OR, XOR, XNOR ⇒ Basic gates

AND gate :

0, 1. 00
01
10
11

A , B

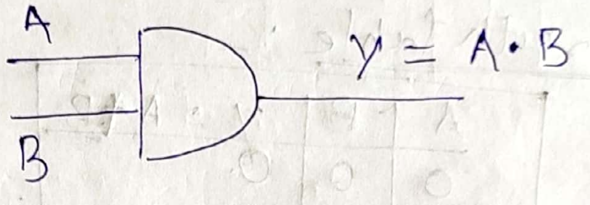


- A, B → two i/ps
  Y → o/p

- Represented by (•) operator
  i.e A • B

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

Symbol :-



Truth table :-

A	B	Y = A·B
0	0	0
0	1	0
1	0	0
1	1	1

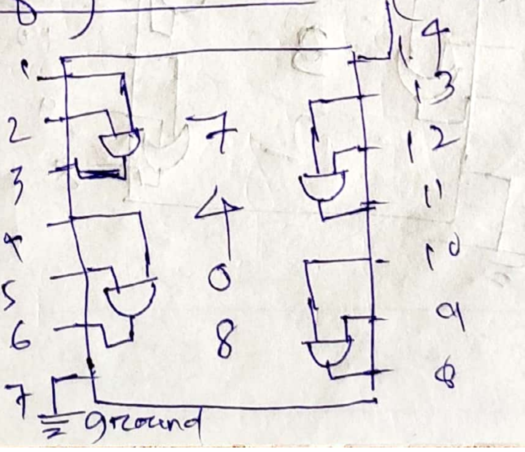
(LED off)
0 → Low
1 → High (LED glow)

Logic statement :-

The o/p is high (1), if both the i/ps are high (1),

or The o/p is low (0) if a single i/p is low (0)

Pin configuration :



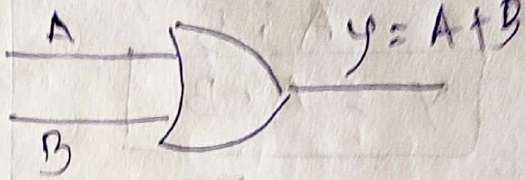
+ supply
− gnd

OR gate :-

A, B → two i/ps
Y → o/p.

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

Symbol :-



Truth table :-

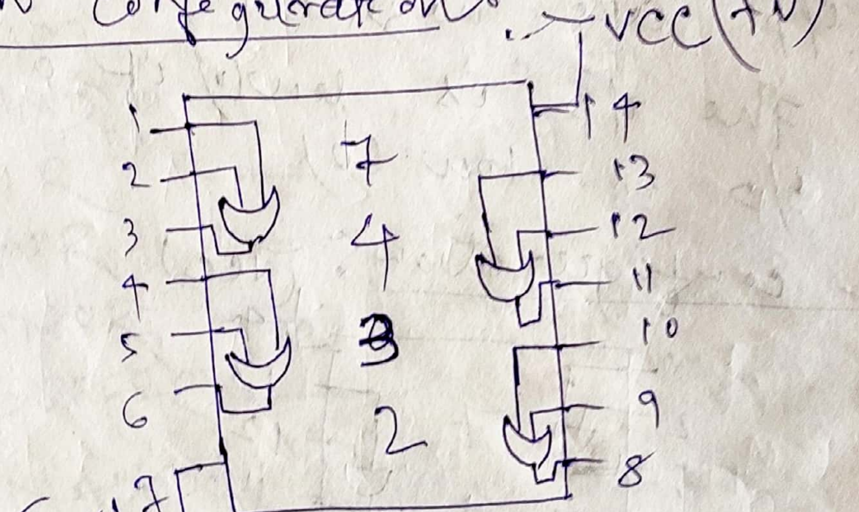
A	B	Y = A+B
0	0	0
0	1	1
1	0	1
1	1	1

logic statement :-

the o/p is low (0) if both the i/ps are low.

or the o/p is high (1) if a single i/p is high.

pin configuration :-



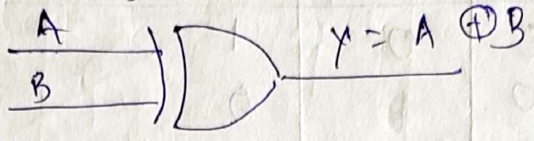
XOR :-

A, B → two i/ps

Y → o/p.

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

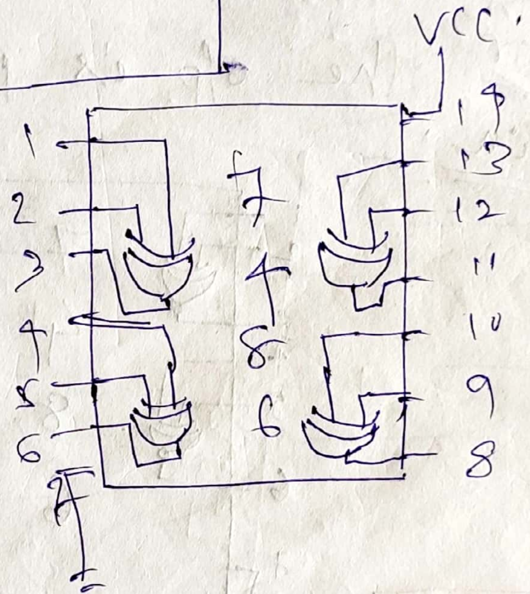
Symbol :-



Truth table:

A	B	Y = A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

Pin :-



Logic statement

The o/p is high for dissimilar i/ps.

or The o/p is low for similar i/ps.

XNOR :-

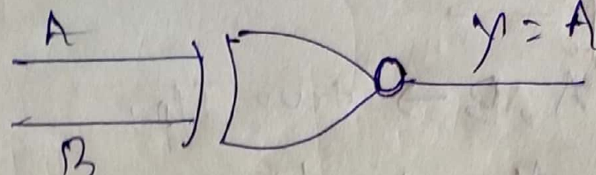
A, B → i/ps

Y → o/p.

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

Symbol:

XNOR



Truth table:

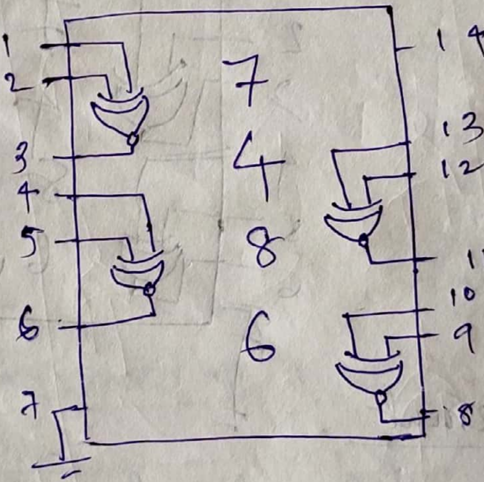
A	B	$Y = A \odot B = \overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

Logic statement:

The o/p is high (1) for similar i/ps.

or the o/p is low for dissimilar i/ps.

Pin:



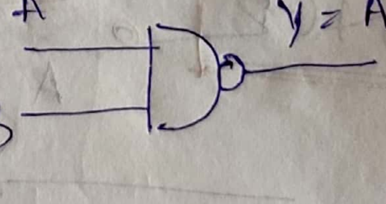
NAND

A, B → i/ps

Y → o/p

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

~~truth tabl~~ Symbol:-



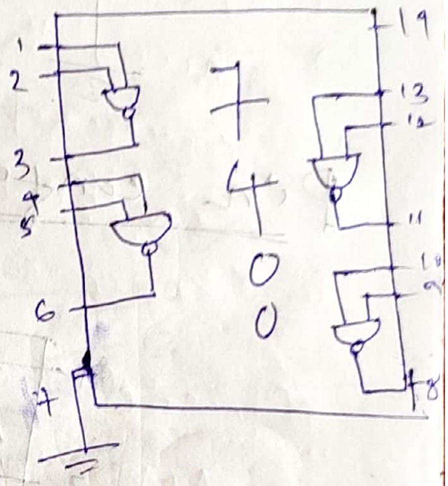
Truth table:

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

Logic statement:

The o/p is for a single i/p as low
or The o/p is low for both the i/ps are high.

Pin configuration:



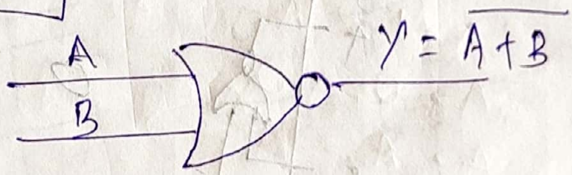
NOR gate:

A, B → i/ps

Y → o/p

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

Symbol:



Truth table:

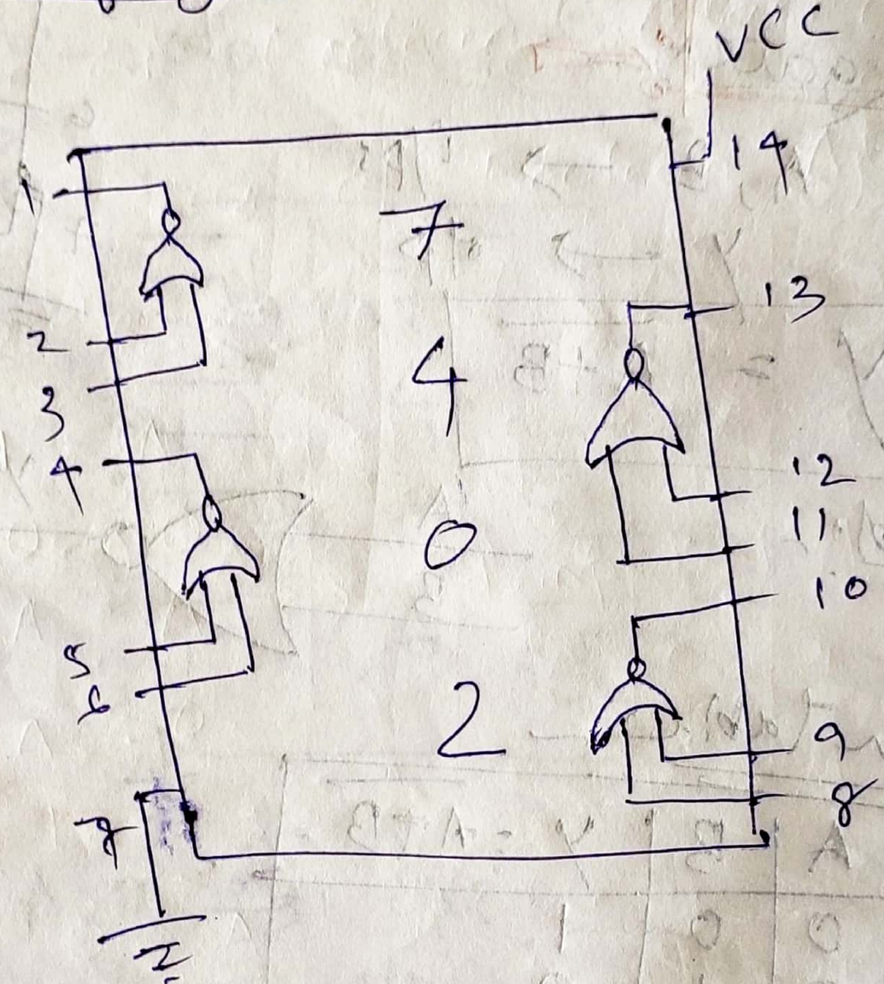
A	B	$Y = \overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Logic statement:

The o/p is high if both i/ps are low.
or The o/p is low if a single i/p is high.

Scanned with CamScanner

Pin configuration :-



Scanned with CamScanner

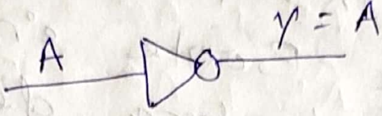
D.14

NOT gate :- or Inverter

A → i/p
Y → o/p

then $\boxed{Y = \overline{A}}$

Symbol :-

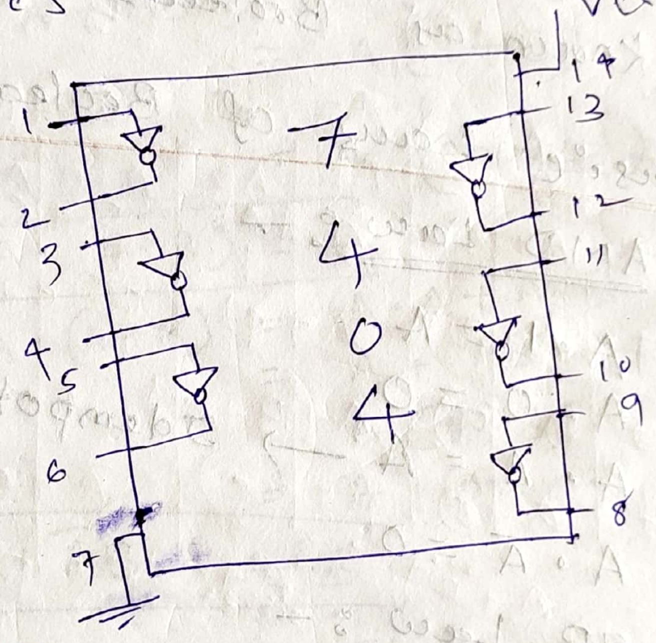


Truthtable :-

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

Logic statement :- The o/p is high(1) for low (0) i/p. or The o/p is low (0) for high (1) i/p.

Pin :-



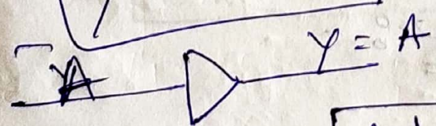
Buffer gate :-

→ Used to provide low impedance (resistance) path from i/p to o/p.

A → i/p
Y → o/p

$\boxed{Y = A}$

Symbol :-



Truth Table :-

A	Y
0	0
1	1

Boolean Algebra :-

→ It is used to simplify the design of logic circuits.

→ It is introduced by mathematician George Boole in 1854.

→ This algebra deals with the the rules by which the logical operations are carried out.

→ Thus, the mathematical operations carried out by digital ckts in binary mode (i.e. 0 & 1), is known as Booleen Algebra.

Basic Laws of Boolean Algebra :-

→ **AND Law :-**

$A \cdot 1 = A$

$A \cdot 0 = 0$

$A \cdot A = A$ → Idempotent law

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

→ **OR Law :-**

$A + 0 = A$

$A + 1 = 1$

$A + A = A$ → Idempotent law

$A + \bar{A} = 1$

→ **Inversion Law :-**

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

Scanned with CamScanner

→ **Commutative property** :-

$$A + B = B + A$$
$$A \cdot B = B \cdot A$$

→ **Associative property** :-

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

→ **Distributive property** :-

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

→ **Absorption Law** :-

(i) $\boxed{A + AB = A}$

Proof = $A + AB = A \cdot 1 + AB$
$= A(1 + B)$
$= A \cdot 1$
$= A$ (proved)

(ii) $\boxed{A \cdot (A + B) = A}$

Proof = $A \cdot (A + B) = A \cdot A + A \cdot B$
$= A + AB$
$= A(1 + B)$
$= A$ (proved)

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

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

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

Proof = $A \cdot (\bar{A} + B) = A \cdot \bar{A} + AB$
$= 0 + AB$
$= AB$ (proved)

→ **Consensus Law :-**

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

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

$= AB + \bar{A}C + BC \cdot 1$

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

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

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

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

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

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

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

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

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

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

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

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

D. 17.08.2020

* **Principle of Duality :-**

- According to the above laws, it is clear that one part of Boolean function may be obtained from the other part if the binary operators & the identity elements are interchanged.

- This is called duality principle

- Simply interchange OR-operator by AND-operator & vice-versa. Also, replace 1s by 0s and 0s by 1s.

Ex :- $(A+B)C + DB = 1$. Find its duality.

Ans :- $(AB + C)(D+B) = 0$.

D. 17. 08. 2020

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

Sol^n $= A\bar{B}.A + A\bar{B}C + AC.\bar{A} + A.C.\bar{B}$

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

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

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

$= A\bar{B}$ Ans

Q Complement the expression $\bar{A}B + C\bar{D}$

Sol^n - $\overline{\bar{A}B + C\bar{D}}$

$= (\overline{\bar{A}B}).(\overline{C\bar{D}})$ [∵ De-morgan's theorem]

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

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

De-morgan's theorem :-

Theorem-1 :- The complement of a product is equal to the sum of the complements of the variables.

i.e. - $\boxed{\overline{AB} = \overline{A} + \overline{B}}$

Theorem-2 :- The complement of a sum is equal to the product of the complements.

i.e. $\boxed{\overline{A+B} = \overline{A} \cdot \overline{B}}$

Proof —

theorem-1 LHS = RHS

Theorem-2 LHS = RHS

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

Q Simplify the expression

$$Y = \overline{A}B + ABD + A\overline{B}C\overline{D} + BC$$

Solⁿ

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

$$= B(\overline{A} + D) + C(B + A\overline{D}) \quad [\because A + \overline{A}B = A + B]$$

$$= \overline{A}B + BD + BC + AC\overline{D}$$

$$= \overline{A}B + BD + BC(A + \overline{A}) + AC\overline{D}$$

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

$= \bar{A}B(1+C) + BD + ABC + AC\bar{D}$

$= \bar{A}B + BD + ABC + AC\bar{D}$ Ans

Q. $Y = A\bar{B} + (\overline{\bar{A}+B})C$. solve

Sol^n

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

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

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

Representation of Boolean function:

- Standard form
  - Max term
  - Min term
- Canonical form
  - SOP (Sum of product)
  - POS (Product of sum)

Max term
→ Formed by 'OR' operator
→ Represented by πm_j where $j = 0, 1, 2 \dots$

Min term
→ Formed by 'AND' operator
→ Represented by Σm_j where $j = 0, 1, 2 \dots$

⇒ Min terms are standard product terms & max. terms are standard sum terms.

⇒ In Max term, 0 → non-complemented form
1 → complemented form

min term, 1 → non-complemented form
0 → complemented form

Scanned with CamScanner

Ex:- $F = A\bar{B}C + \bar{A}BC + AB\bar{C}$

min terms are —

$A\bar{B}C \rightarrow 101 \rightarrow 5$

$\bar{A}BC \rightarrow 011 \rightarrow 3$

$AB\bar{C} \rightarrow 110 \rightarrow 6$

$\therefore \sum m(3,5,6)$ min terms

$= \sum(m_3, m_5, m_6)$

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

max terms are —

$(A+\bar{B}+C) \rightarrow 010 \rightarrow 2$

$(\bar{A}+B+C) \rightarrow 100 \rightarrow 4$

$(A+B+C) \rightarrow 000 \rightarrow 0$

$\therefore \prod m(0,2,4)$ max terms

$= \prod(m_0, m_2, m_4)$

Scanned with CamScanner

D. 19.08.2020

POS form :- Product of sum

(~~Sum of Product~~)

Ex $F = \underbrace{(A+B+C)}_{\text{sum / max term}} \cdot (\bar{A}+B+C) \cdot (A+\bar{B}+C)$

POS $\leftrightarrow$ max term

SOP form :- (sum of product)

Ex - $F = \underbrace{\bar{A}BC}_{\text{Product / min}} + AB\bar{C} + \bar{A}\bar{B}\bar{C}$ — SOP

SOP $\Leftrightarrow$ min term

→ In POS form the expressions are written in max terms.

→ In SOP form the expressions are written in min terms.

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

Find the max term.

$A+B+C \rightarrow 000 \rightarrow 0$

$\bar{A}+B+C \rightarrow 100 \rightarrow 1$

$A+\bar{B}+C \rightarrow 010 \rightarrow 2$

Q Given $F = \prod M(0, 2, 3, 4)$

Write the POS form.

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

POS

$0 \rightarrow 000 \rightarrow A+B+C$

$2 \rightarrow 010 \rightarrow A+\bar{B}+C$

$3 \rightarrow 011 \rightarrow A+\bar{B}+\bar{C}$

$4 \rightarrow 100 \rightarrow \bar{A}+B+C$

Ans

$A+B+C$

$4 \rightarrow 100$ Max

ABC

$\boxed{\begin{array}{l} 0 \rightarrow A \\ 1 \rightarrow \bar{A} \end{array}}$

Q Given $F = \sum m(0, 1, 5, 7)$

write in SOP form.

Soln: Min terms are —

$0 \rightarrow 000 \rightarrow \bar{A}.\bar{B}.\bar{C}$

$1 \rightarrow 001 \rightarrow \bar{A}\bar{B}C$

$5 \rightarrow 101 \rightarrow A\bar{B}C$

$7 \rightarrow 111 \rightarrow ABC$

111
ABC

$0 \rightarrow \bar{A}$

$1 \rightarrow A$

SOP expression —

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

Procedures for deriving Minterms

steps Q $F = A + \bar{B}C$. Find the minterms. → SOP

Soln - Given $F = \underset{\text{1st}}{A} + \underset{\text{2nd term}}{\bar{B}C}$

$\Rightarrow F = A \cdot X \cdot X \cdot + X \cdot \bar{B}C$

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

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

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

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

$\Rightarrow F = ABC + AB\bar{C} + A\bar{B}C + A\bar{B}\bar{C}$
$+ \bar{A}\bar{B}C$ Ans.

SOP form.

Min terms —

$A.B.C \rightarrow 111 \rightarrow 7$

$AB\bar{C} \rightarrow 110 \rightarrow 6$

$A\bar{B}C \rightarrow 101 \rightarrow 5$

$A\bar{B}\bar{C} \rightarrow 100 \rightarrow 4$

$\bar{A}\bar{B}C \rightarrow 001 \rightarrow 1$

$F = \Sigma m(1, 4, 5, 6, 7)$ Ans.

Procedure for deriving Max term

Q $F = (A+B)(\bar{B}+\bar{C})$. find max terms.

2014 - $\cancel{F = (A+B+C.\bar{C})}$ (

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

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

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

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

POS form

max. term —

$A+B+C \rightarrow 000 \rightarrow 0$

$A+B+\bar{C} \rightarrow 001 \rightarrow 1$

$A+\bar{B}+\bar{C} \rightarrow 011 \rightarrow 3$

$\bar{A}+\bar{B}+\bar{C} \rightarrow 110 \rightarrow 7$.

$F = \Pi M(0, 1, 3, 7)$ Ans

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

(i) POS — max term

(ii) SOP — min terms

Take hint

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

D. 21.08.2020

H.W.

Sol^n - (i) $F = A + \bar{B}C$

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

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

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

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

Max terms —

$A+\bar{B}+C = 010 = 2$

$A+\bar{B}+\bar{C} = 011 = 3$

$A+B+C = 000 = 0$

$F = \Pi M(0,2,3)$ Ans

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

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

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

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

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

$+ \bar{A}\bar{B}C$ → SOP form

Min terms

$ABC = 111 = 7$

$AB\bar{C} = 110 = 6$

$A\bar{B}C = 101 = 5$

$A\bar{B}\bar{C} = 100 = 4$

$\bar{A}\bar{B}C = 001 = 1$

$F = \Sigma m(1,4,5,6,7)$ Ans

Scanned with CamScanner

Q $Y = D \longrightarrow$ POS form

Soln $Y = D + A\bar{A}$

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

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

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

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

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

Max term –

$\underbrace{0000}_{0}$, $\underbrace{0010}_{2}$, $\underbrace{0100}_{4}$, $\underbrace{0110}_{6}$,

$\underbrace{1000}_{8}$, $\underbrace{1010}_{10}$, $\underbrace{1100}_{12}$, $\underbrace{1110}_{14}$

$Y = \prod M (0, 2, 4, 6, 8, 10, 12, 14)$. Ans

Scanned with CamScanner

Gate level minimization Technique

Karnaugh Map (K-map) method

- It is used to simplify Boolean equations with less no. of variables. So, requirement of gates are also less.
- Here, for n-variable K-map, 2^n cells are required & each cell is a combination of variables.
- K-map can be done by –
  - i) 2-variables
  - ii) 3-variables
  - iii) 4-variables → in syllabus
  - iv) 5-variables

(i, ii, iii → in syllabus.)

2-variable K-map :— A, B → variables

S = no. of cells or sub squares.

$= 2^{\text{no. of variable}}$

$= 2^2 = 4$

A \ B	$\bar{B}$ 0	B 1
$\bar{A}$ 0	0 (00)	1 (01)
A 1	2 (10)	3 (11)

3-variable K-map : A, B, C → variables

$S = 2^3 = 8$

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

Scanned with CamScanner

4-Variable K-map

A, B, C, D → 4 variable

$S = 2^4 = 16$

AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	0 0000	1 0001	3 0011	2 0010
$\bar{A}B$ 01	4 0100	5 0101	7 0111	6 0110
AB 11	12 1100	13 1101	15 1111	14 1110
$A\bar{B}$ 10	8 1000	9 1001	11 1011	10 1010

D. 26.08.2020

Looping in K-map :-

(To eliminate variables)

- Self loop → No elimination
- pair loop → eliminate 1 variable
- Quad loop → eliminate 2 variables.
- Octet loop → eliminate 3 variables.

Restriction :-

- Adjacent squares should take part in elimination.
- Priority given in the order of octet loop > Quad loop > pair loop > self loop.

Q- Given, $F = \Sigma m(0, 2, 3)$.

Simplify using K-map

Solⁿ - It's a 3-variable K-map as 3 variables.

A, B, C

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

$L_1 = \bar{A}\bar{C}$

$L_2 = \bar{A}B$

$F = L_1 + L_2$

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

Scanned with CamScanner

Q Given $F_{(A,B,C)} = \Sigma m(1, 2, 3, 7)$

Sol^n -

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

L1, L2, L3

$L_1 = \bar{A}B$

$L_2 = \bar{A}C$

$L_3 = BC$

$F = L_1 + L_2 + L_3$

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

Q Simplify the expression

$Y = \Sigma m(3, 4, 5, 7, 9, 13, 14, 15)$

Sol^n - The K-map (4 variable) is -

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

L1, L2, L3, L4

$L_1 = \bar{A}CD$, $L_2 = ABC$, $L_3 = A\bar{C}D$, $L_4 = \bar{A}B\bar{C}$

$\therefore \boxed{Y = \bar{A}CD + ABC + A\bar{C}D + \bar{A}B\bar{C}}$ Ans

D. 28.08.2020

Q Simplify using K-map —

$$Y = \pi m(0, 2, 3, 4, 6, 8, 14, 15)$$

Sol^n — K-map is —

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

L_1, L_2, L_3, L_4

$$L_1 = A+D$$

$$L_2 = A+B+\bar{C}$$

$$L_3 = \bar{A}+\bar{B}+\bar{C}$$

$$L_4 = B+C+D$$

$$\therefore Y = L_1 \cdot L_2 \cdot L_3 \cdot L_4$$

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

Ans

Scanned with CamScanner

Q. $Y = \pi m(0, 2, 3, 4, 6, 7)$. Simplify using k-map?

max (POS)

111 ABC

Sol^n:

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

L_1, L_2

$L_1 = \bar{B}$

$L_2 = C$

Σ min
$Y = L_1 + L_2$
SOP

$\therefore Y = L_1 \cdot L_2$

$$\boxed{Y = \bar{B} \cdot C}$$ Ans

Q. $Y = \pi m(0, 1, 2, 3, 9, 11, 13, 15)$ max POS

1111

K-map

Sol^n

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

L_1, L_2

Σ AB

$L_1 = A+B$

$L_2 = \bar{A}+\bar{D}$

Scanned with CamScanner

POS

$Y = L_1 \cdot L_2$

$$\boxed{Y = (A+B)\cdot(\bar{A}+\bar{D})}$$ Ans

Q $Y = \Sigma m(0, 2, 4, 6, 8, 10, 12, 14, 15)$

Soln

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

L1

L2

$L_1 = \bar{D}$

$L_2 = ABC$

$$\boxed{Y = L_1 + L_2 = \bar{D} + ABC}$$ Ans

Q $Y = \Pi m(1, 3, 5, 7, 9, 11, 13, 15)$

K-map

Q $Y = \Sigma m(4, 5, 6, 7, 8, 9, 10, 11, 15)$

K-map

29/04/2020

Don't care condition:-

- min terms:-
- max terms-
- don't care bits-

- The bits which have no existence of 0 or 1 i.e. you don't know whether the don't care bits are 1 or 0 during transmission
- Transmitted along with minterms or max terms.
- They are represented as 'X' or 'd' symbols

Q. $Y = \Pi M(0,1,4,5) + d(2,7)$

Solve k-map. max terms, don't care bits.

Sol^n:-

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

$L_1 = B$

$\therefore \boxed{Y = L_1 = B}$ (Ans)

Scanned with CamScanner

Q- $Y = \Sigma m(0, 2, 3, 5, 7, 9, 11) + \Sigma d(1, 13, 15)$

Solⁿ — K-map

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

L_2

L_1

$L_1 = D$

$L_2 = \bar{A}\bar{B}$

$Y = L_1 + L_2$ SOP

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

Q- $Y = \Sigma m(2, 3, 6, 7, 11, 14) + \Sigma d(0, 1, 10)$

Solⁿ

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

$L_1 = \bar{A}C$

$L_3 = C\bar{D}$

$L_2 = \bar{B}C$

$\boxed{Y = L_1 + L_2 + L_3 = \bar{A}C + \bar{B}C + C\bar{D}}$ Ans

Scanned with CamScanner

Q. H.T. $Y = \prod m(0, 1, 4, 5, 8, 9, 13) + \prod d(12, 14, 15)$

D. 31.08.2020

NAND and NOR Implementation

Universal gate

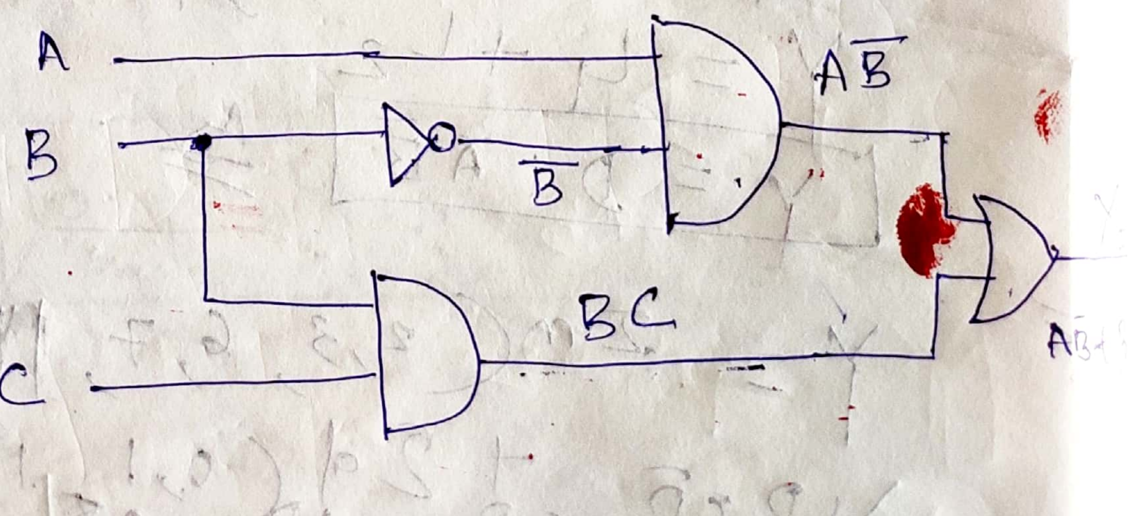
OR, AND, NOT, XOR, XNOR

Basic gates

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

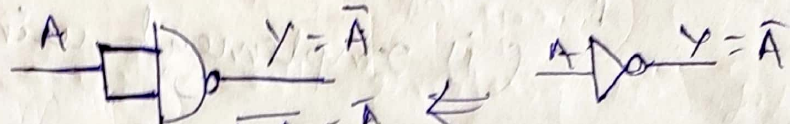
i) Implement using (Draw) Basic gates.

Ans



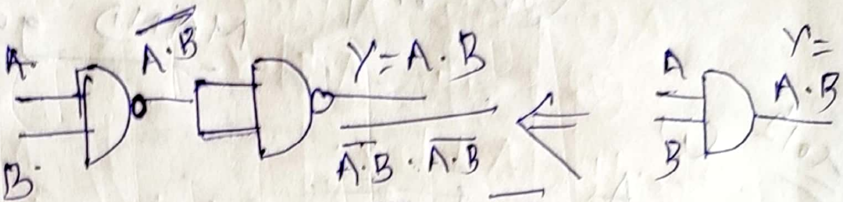
Using only NAND gates:

NOT →



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

AND →



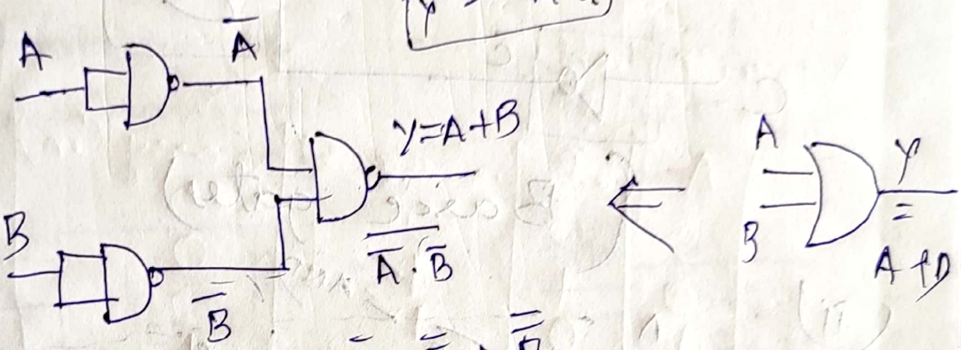
$\overline{\overline{A \cdot B} \cdot \overline{A \cdot B}}$

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

$= A \cdot B + A \cdot B$

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

OR →



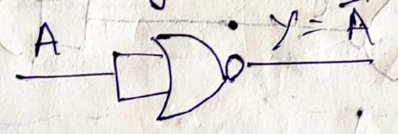
$\overline{\bar{A} \cdot \bar{B}}$

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

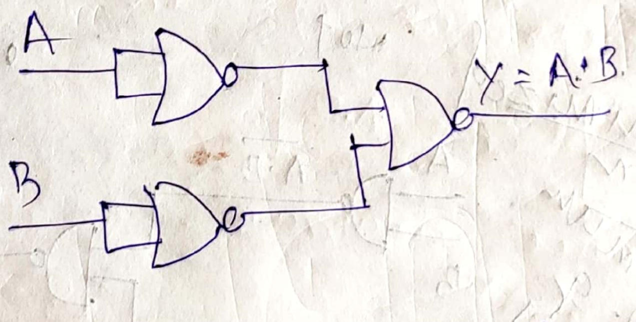
$\boxed{Y = A + B}$

Using only NOR gates:

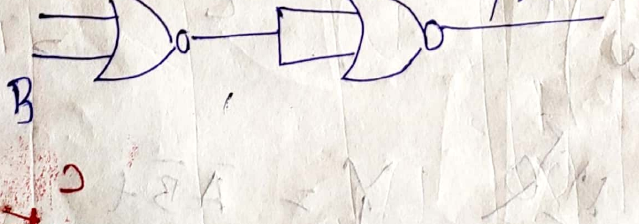
NOT →



AND →



OR →



Stop

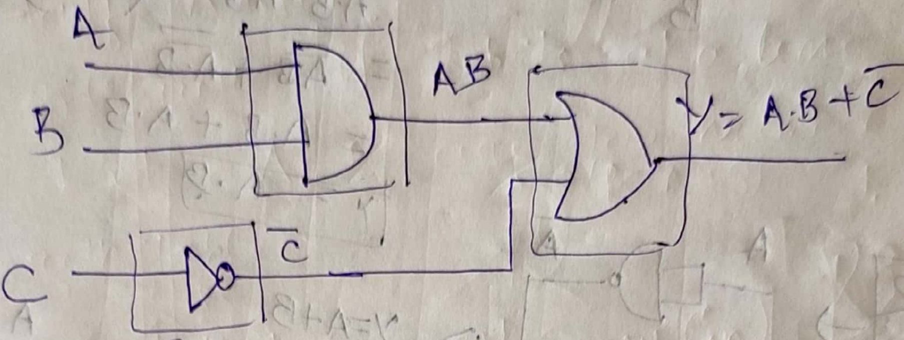
Q $Y = A.B + \overline{C}$

(i) Implement using Basic gates

(ii) Implement using NAND gates

Sol

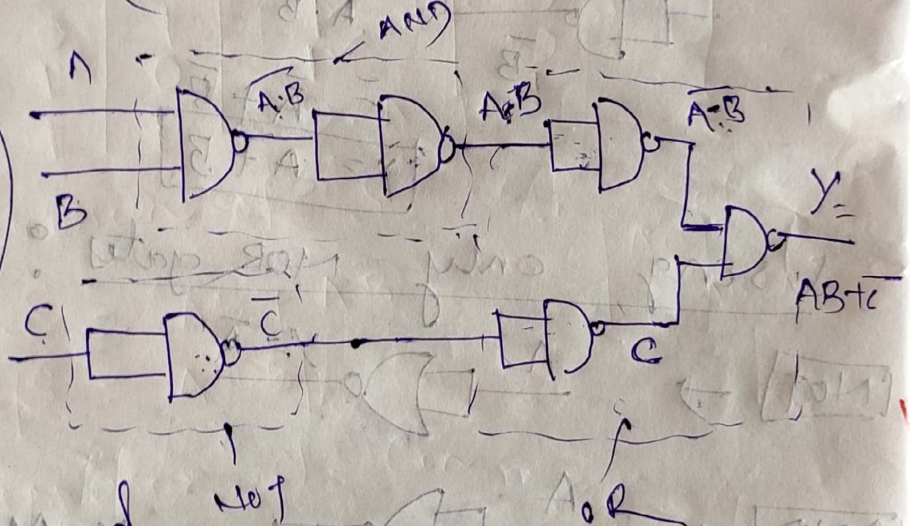
(i)



(Basic gates)

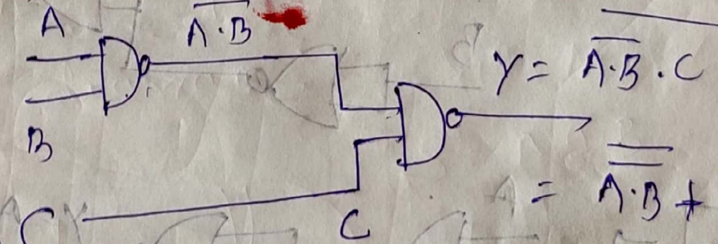
(ii)

(Using NAND gates)



minimized form

(minimum no. of NAND gates)



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

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

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

H.Q.

$Y = \overline{A}\,\overline{B} + C$

(1) Basic gate

(2) NAND

(3) NOR