

Assignment-1* Number Systems and Codes *

Ques.1. Write the first 20 decimal digits in base 3.

$$\text{i} > \begin{array}{|c|c|} \hline 3 & 1 \\ \hline 0 & 1 \\ \hline \end{array}$$

Ans.-1

$$\text{ii} > \begin{array}{|c|c|} \hline 3 & 2 \\ \hline 0 & 2 \\ \hline \end{array}$$

Ans.-2

$$\text{iii} > \begin{array}{|c|c|} \hline 3 & 3 \\ \hline 3 & 1 \\ \hline 0 & 1 \\ \hline \end{array}$$

Ans.-10

$$\text{iv} > \begin{array}{|c|c|} \hline 3 & 4 \\ \hline 3 & 1 \\ \hline 0 & 1 \\ \hline \end{array}$$

Ans.-11

$$\text{v} > \begin{array}{|c|c|} \hline 3 & 5 \\ \hline 3 & 1 \\ \hline 0 & 1 \\ \hline \end{array}$$

Ans.-12

$$\text{vi} > \begin{array}{|c|c|} \hline 3 & 6 \\ \hline 3 & 2 \\ \hline 0 & 2 \\ \hline \end{array}$$

Ans.-20

$$\text{vii} > \begin{array}{|c|c|} \hline 3 & 7 \\ \hline 3 & 2 \\ \hline 0 & 2 \\ \hline \end{array}$$

Ans.-21

$$\text{viii} > \begin{array}{|c|c|} \hline 3 & 8 \\ \hline 3 & 2 \\ \hline 0 & 2 \\ \hline \end{array}$$

Ans.-22

$$\text{ix} > \begin{array}{|c|c|} \hline 3 & 9 \\ \hline 3 & 3 \\ \hline 0 & 1 \\ \hline \end{array}$$

Ans.-100

$$\text{x} > \begin{array}{|c|c|} \hline 3 & 10 \\ \hline 3 & 3 \\ \hline 3 & 1 \\ \hline 0 & 1 \\ \hline \end{array}$$

Ans.-101

$$\text{xii} > \begin{array}{|c|c|} \hline 3 & 11 \\ \hline 3 & 3 \\ \hline 3 & 1 \\ \hline 0 & 1 \\ \hline \end{array}$$

Ans.-102

$$\text{xii} > \begin{array}{|c|c|} \hline 3 & 12 \\ \hline 3 & 4 \\ \hline 3 & 1 \\ \hline 0 & 1 \\ \hline \end{array}$$

Ans.-110

$$\text{xiii} > \begin{array}{|c|c|} \hline 3 & 13 \\ \hline 3 & 4 \\ \hline 3 & 1 \\ \hline 0 & 1 \\ \hline \end{array}$$

Ans.-111

$$\text{xiv} > \begin{array}{|c|c|} \hline 3 & 14 \\ \hline 3 & 4 \\ \hline 3 & 1 \\ \hline 0 & 1 \\ \hline \end{array}$$

Ans.-112

$$\text{xv} > \begin{array}{|c|c|} \hline 3 & 15 \\ \hline 3 & 5 \\ \hline 3 & 1 \\ \hline 0 & 1 \\ \hline \end{array}$$

Ans.-120

$$\text{xvi} > \begin{array}{|c|c|} \hline 3 & 16 \\ \hline 3 & 5 \\ \hline 3 & 1 \\ \hline 0 & 1 \\ \hline \end{array}$$

Ans.-121

$$\text{xvii} > \begin{array}{|c|c|} \hline 3 & 17 \\ \hline 3 & 5 \\ \hline 3 & 1 \\ \hline 0 & 1 \\ \hline \end{array}$$

Ans.-122

$$\text{xviii} > \begin{array}{|c|c|} \hline 3 & 18 \\ \hline 3 & 6 \\ \hline 3 & 2 \\ \hline 0 & 2 \\ \hline \end{array}$$

Ans.-200

xix > 3 | 19

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

Ans. - 201

xx > 3 | 20

$$\begin{array}{r} 3 \\ 6 \quad 2 \\ 3 \quad 2 \quad 0 \\ 0 \quad 2 \end{array}$$

Ans. - 202

Que. 2.

Convert the decimal no. 250.5 to base 3, base 4, base 4, base 7, base 8, base 16.

→

Base 3 :

3 | 250

$$\begin{array}{r} 3 \\ 83 \quad 1 \\ 3 \quad 27 \quad 2 \\ 3 \quad 9 \quad 0 \\ 3 \quad 3 \quad 0 \\ 3 \quad 1 \quad 0 \\ 0 \quad 1 \end{array}$$

$$0.5 * 3 = 1.5 = 1$$

$$0.5 * 3 = 1.5 = 1$$

$$0.5 * 3 = 1.5 = 1$$

$$\text{Ans. : } (100021.111)_3 = 250.5$$

Base 4 :

4 | 250

$$\begin{array}{r} 4 \\ 62 \quad 2 \\ 15 \quad 2 \\ 3 \quad 3 \\ 0 \quad 3 \end{array}$$

$$0.5 * 0.4 = 2.0$$

$$\text{Ans. - } 250.5 = (3322.2)_4$$

Base 7 :

7 | 250

$$\begin{array}{r} 7 \\ 35 \quad 5 \\ 5 \quad 0 \\ 0 \quad 5 \end{array}$$

$$0.5 * 7 = 3.5 \div 3$$

$$0.5 * 7 = 3.5 \div 3$$

$$\text{Ans. - } 250.5 = (505.33)_7$$

Base 8 :

8 250	$0.5 * 8 = 4.0 = 4$
8 31 2	
8 3 7	
0 3 1	Ans. - $250_{10} = (372)_8$

Base 16 :

16 250	$0.5 * 16 = 8.0 = 8$
16 15 10 = A	
0 15 = F	Ans. - $(F A.8)_{16} = 250_{10}$

Que.3. Convert the following decimal no.s to binary.

(1) 12.0625

2 12	$0.0625 * 2 = 1.250 = 1$
2 6 0	$0.25 * 2 = 0.5 = 0$
2 3 0	$0.5 * 2 = 1.0 = 1$
2 1 0	
0 1	Ans. - $(12.0625)_{10} = (1100.101)_2$

(2) $(10^4)_{10}$

2 10000	
2 5000 0	
2 2500 0	
2 1250 0	
2 625 0	
2 312 1	
2 158 0	
2 78 0	
2 39 1	
2 19 1	
2 9 1	
2 4 0	
2 2 0	
2 1 0	
0 1	

(3) 673.23

2	673	
2	336 1	$0.23 * 2 = 0.46 = 0$
2	165 0 ↑	$0.46 * 2 = 0.96 = 0$
2	94 0	$0.96 * 2 = 1.84 = 1$
2	42 0	$0.84 * 2 = 1.68 = 1$
2	21 0	$0.68 * 2 = 1.36 = 1$
2	10 1	$0.36 * 2 = 0.72 = 0$
2	5 0	
2	2 1	Ans. - $(673.23)_0$
2	1 0	$(101010001.001110)_0$
	0 1	

(4) 1998

2	1998	
2	999 0	
2	499 1 ↑	
2	249 1	Ans. - $(1998)_0 =$
2	124 1	
2	62 0	$(11111001110)_0$
2	31 0	
2	15 1	
2	7 1	
2	3 1	
2	1 1	
	0 1	

Que-4 Convert the following binary number to decimal.

(1) 10.10001

$$1 \times 2^1 = 2$$

$$0 \times 2^0 = 0$$

$$1 \times 2^{-1} = 0.5$$

$$0 \times 2^{-2} = 0$$

$$0 \times 2^{-3} = 0$$

$$0 \times 2^{-4} = 0$$

$$\text{Ans. } = (2.53125)_{10}$$

$$1 \times 2^{-5} = 0.03125$$

(2) 101110.0101

$$0 \times 2^0 = 0$$

$$1 \times 2^1 = 2$$

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

$$1 \times 2^3 = 8$$

$$0 \times 2^4 = 0$$

$$1 \times 2^5 = 32$$

$$0 \times 2^{-1} = 0$$

$$1 \times 2^{-2} = 0.25$$

$$0 \times 2^{-3} = 0$$

$$1 \times 2^{-4} = 0.0625$$

$$\text{Ans. } = (32+8+4+2) \cdot (0.25+0.0625)$$

$$= (46.3125)_{10}$$

(3) 1110101.110

$$1 \times 2^0 = 01$$

$$0 \times 2^1 = 0$$

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

$$0 \times 2^3 = 0$$

$$1 \times 2^4 = 16$$

$$1 \times 2^5 = 32$$

$$1 \times 2^6 = 64$$

$$1 \times 2^{-1} = 0.5$$

$$1 \times 2^{-2} = 0.25$$

$$0 \times 2^{-3} = 0$$

$$1 \times 2^{-4} = 0.0625$$

$$1 \times 2^{-5} = 0.03125$$

$$1 \times 2^{-6} = 0.015625$$

$$\text{Ans. } = (64+32+16+4+1) \cdot (0.5+0.25+0.0625+0.03125+0.015625)$$

$$= (117.75)_{10}$$

(4) 1101101.111

$$1 \times 2^0 = 1$$

$$0 \times 2^1 = 0$$

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

$$1 \times 2^3 = 8$$

$$0 \times 2^4 = 0$$

$$1 \times 2^5 = 32$$

$$1 \times 2^6 = 64$$

$$1 \times 2^{-1} = 0.5$$

$$1 \times 2^{-2} = 0.25$$

$$1 \times 2^{-3} = 0.125$$

$$\text{Ans.} = (64 + 32 + 8 + 4 + 1) \cdot (0.5 + 0.25 + 0.125)$$

$$= (109.875)_{10}$$

Que-5 Convert the following number from the given base to the bases indicated

(a) decimal 225.225 to binary, Octal & Hexadecimal

<i>

2	225	
2	112 1	$0.225 * 2 = 0.45 = 0$
2	56 0	$0.45 * 2 = 0.9 = 0$
2	28 0	$0.9 * 2 = 1.8 = 1$
2	14 0	$0.8 * 2 = 1.6 = 1$
2	7 0	$0.6 * 2 = 1.2 = 1$
2	3 1	
2	1 1	
0	1	

Ans. - $(11100001.00111)_2$

<i>

$$(225.225)_{10} = (\text{ })_8$$

8	225	
8	28 1	
8	3 4	
0	3	

$$0.225 * 8 = 1.8 = 1$$

$$0.8 * 8 = 6.4 = 6$$

$$0.4 * 8 = 3.2 = 3$$

$$0.2 * 8 = 1.6 = 1$$

Ans. - $(341.163)_8$

(iii) $(225.225)_{10} = (?)_{16}$

16	225	0.225 * 16 = 3.6 = 3
16	14	0.6 * 16 = 9.6 = 9
0	14 = E	0.6 * 16 = 9.6 = 9

Ans. - (E1.399)

(b) binary 11010111.110 to decimal, octal, & hexadecimal

(i) $(11010111.110)_2 = (?)_{10}$

$$= 1 \times 2^0 = 1 \quad 1 \times 2^{-1} = 0.5$$

$$1 \times 2^1 = 2 \quad 1 \times 2^{-2} = 0.25$$

$$1 \times 2^2 = 4 \quad 0 \times 2^{-3} = 0$$

$$0 \times 2^3 = 0$$

$$1 \times 2^4 = 16 \quad \text{Ans.} = (128 + 64 + 16 + 4 + 2 + 1) .$$

$$0 \times 2^5 = 0 \quad (0.5 + 0.25)$$

$$1 \times 2^6 = 64 \quad = (215.75)_{10}$$

$$1 \times 2^7 = 128$$

(ii) $(11010111.110)_2 = (?)_8$

$$(011 \quad 010 \quad 111 \quad . \quad 110)$$

$$\begin{array}{ccccccc} / & / & | & & | \\ 2+1 & 2 & 4+2+1 & & 4+2 = 6 \\ = 3 & & = 7 & & \end{array} \quad 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

$$= (327.6)_8$$

<iii> $(11010111.110)_2 = (?)_{16}$

$$\begin{array}{r}
 (\underline{1101} \quad \underline{0111} \quad \cdot \underline{1100}) \\
 8+4+1 \quad 4+2+1 \quad 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 \\
 = 13 = D \quad = 7 \quad = 8+4 = 12 = C \\
 \\ = (D7.C)_{16}
 \end{array}$$

(C) Octal - 623.77 to decimal, binary & hexadecimal

<i> $(623.77)_8 = (?)_{10}$

$$\begin{array}{ll}
 6 \times 8^2 = 384 & 7 \times 8^{-1} = 0.875 \\
 2 \times 8^1 = 16 & 7 \times 8^{-2} = 0.1093 \\
 3 \times 8^0 = 3 &
 \end{array}$$

$$\text{Ans.} = (403.9843)_{10}$$

<ii> $(623.77)_8 = (?)_2$

$$\begin{array}{ccccccccc}
 6 & 2 & 3 & . & 7 & 7 \\
 \hline
 110 & 010 & 011 & \underline{111} & 2 & 7 \\
 & & & & 2 & 3 & 1 \uparrow \\
 & & & & 2 & 1 & 1 \uparrow \\
 & & & & 0 & & \\
 \Rightarrow & (110010011.111)_2
 \end{array}$$

<iii> $(623.77)_8 = (?)_{16}$

$$\begin{array}{ccccccccc}
 (6 & 2 & 3 & . & 7 & 7) \\
 \hline
 110 & 010 & 011 & \underline{111} & 111 \\
 0001 & 1001 & 0011 & . & 1111 & 1100 \\
 \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\
 1 & 9 & 3 & & 15 & 12 \\
 & & & & = F & = C
 \end{array}$$

$$\text{Ans.} = (193.FC)_{16}$$

(d) Hexadecimal 2ACS.D to decimal, octal and binary.

(i) $(2ACS.D)_{16} = (\quad)_{10}$

$$5 \times 16^0 = 5$$

$$16^{-1} \times 13 = 0.8125$$

$$12 \times 16^1 = 192$$

$$10 \times 16^2 = 2560$$

$$2 \times 16^3 = 8192$$

$$\text{Ans.} = (10949.8125)_{10}$$

(ii) $(2ACS.D)_{16} = (\quad)_8$

$$\begin{aligned} &= \underline{\underline{010}} \underline{\underline{101}} \underline{\underline{011}} \underline{\underline{000}} \underline{\underline{101}} \cdot \underline{\underline{1101}} \\ &\quad \begin{array}{c} 2 \quad 5 \quad 3 \quad 0 \quad 5 \\ \downarrow \end{array} \quad \begin{array}{c} 110 \\ 6 \\ \downarrow \end{array} \quad \begin{array}{c} 100 \\ 4 \\ \downarrow \end{array} = 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 \\ &= (25305.64)_8 \end{aligned}$$

Ques-5 Convert the following nos. to

(iii) $(2ACS.D)_{16} = (\quad)_2$

$$\begin{aligned} &= (\begin{array}{ccccc} 10 & & & & \\ 2 & A & C^2 & S & D \end{array})_2 \\ &= (0010 \quad 1010 \quad 1100 \quad 0101 \quad 1101)_2 \end{aligned}$$

$$= (0010101011000101.1101)_2$$

Ques-6 Convert the following nos to decimal

a) $(1001001.011)_2$

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

$$= (64 + 8 + 1) \cdot (0.25 + 0.125)$$

$$= (73.375)_{10}$$

b) $(121210)_3$

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

$$= (81 + 54 + 9 + 6 + 1)$$

$$= (151)_{10}$$

c) $(1032.2)_4$

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

$$= (64 + 0 + 48 + 16) \cdot (0.5)$$

$$= (78.5)_{10}$$

d) $(4310)_5$

$$= (4 \times 5^3 + 3 \times 5^2 + 1 \times 5^1 + 0 \times 5^0)$$

$$= (500 + 75 + 5 + 0)$$

$$= (580)_{10}$$

e) $(0.342)_6$

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

$$= 0, (0.5 + 0.11 + 0.00925)$$

$$= (0.61925)_{10}$$

f) $(50)_7$

$$= (5 \times 7^1 + 0 \times 7^0)$$

$$= (35)_{10}$$

g) $(8.3)_9$

$$= (8 \times 9^0) \cdot (3 \times 9^{-1})$$

$$= (8.333)_{10}$$

h) $(198)_{12}$

$$= (1 \times 12^2 + 9 \times 12^1 + 8 \times 12^0)$$

$$= 144 + 108 + 8$$

$$= (260)_{10}$$

Que.7 Obtain 1's & 2's Complement of the following binary numbers.

1) 1010101

$$1\text{'s Complement} = 0101010$$

$$2\text{'s Complement} = 0101010 + 1$$

$$0101010 + 1 = 0101011$$

2) 011000

$$1\text{'s Complement} = 1000111$$

$$2\text{'s Complement} = 1000111 + 1$$

$$1000111 + 1 = 1000110$$

$$1000110 + 1 = 1000111$$

3) 0000001010 = 1100000 = from question

$$1\text{'s Complement} = 1111110$$

$$2\text{'s Complement} = 1111110 + 1$$

$$1111110 + 1 = 1111111$$

$$1111111 + 1 = 1111110$$

4) 10000

$$1\text{'s Complement} = 01111$$

$$2\text{'s Complement} = 01111 + 1$$

$$= 01110$$

5) 00000

$$1\text{'s Complement} = 11111$$

$$2\text{'s Complement} = 11111 + 1$$

$$= 11110$$

Que-8 Obtain 9's & 10's Complement of the following binary numbers.

1> 13579

$$\begin{array}{r} \text{9's Complement} = 9 \ 9 \ 9 \ 9 \ 9 \\ - 1 \ 3 \ 5 \ 7 \ 9 \\ \hline 8 \ 6 \ 4 \ 2 \ 0 \end{array}$$

$$10's \text{ Complement} = 86420 + 1 = 86421$$

2> 09900

$$\begin{array}{r} \text{9's Complement} = 9 \ 9 \ 9 \ 9 \ 9 \\ - 0 \ 9 \ 9 \ 0 \ 0 \\ \hline 9 \ 0 \ 0 \ 9 \ 9 \end{array}$$

$$10's \text{ Complement} = 90099 + 1 = 90100$$

3> 90090

$$\begin{array}{r} \text{9's Complement} = 9 \ 9 \ 9 \ 9 \ 9 \\ - 9 \ 0 \ 0 \ 9 \ 0 \\ \hline 0 \ 9 \ 9 \ 0 \ 9 \end{array}$$

$$10's \text{ Complement} = 09909 + 1 = 09910$$

4> 10000

$$\begin{array}{r} \text{9's Complement} = 9 \ 9 \ 9 \ 9 \ 9 \\ - 1 \ 0 \ 0 \ 0 \ 0 \\ \hline 8 \ 9 \ 9 \ 9 \ 9 \end{array}$$

$$10's \text{ Complement} = 89999 + 1$$

$$= 90000$$

5> 00000

$$\begin{array}{r} \text{9's Complement} = 9 \ 9 \ 9 \ 9 \ 9 \\ - 0 \ 0 \ 0 \ 0 \ 0 \\ \hline 9 \ 9 \ 9 \ 9 \ 9 \end{array}$$

$$10's \text{ Complement} = 99999 + 1 = 100000$$

Ques-9 Perform the Subtraction with the following decimal numbers using 1, 10's & 2.9's Complement.

$$a) 5250 - 321$$

$$1) 5250$$

$$2) 0321$$

(1) Smallest is 2nd no. So, take 9's Complement of 2nd.

$$0321 \rightarrow 9999$$

$$\begin{array}{r} 0321 \\ \hline 9678 \end{array}$$

Now, add (1) + (2) 9's Complement

$$5250$$

$$9678$$

$$\begin{array}{r} 5250 \\ 9678 \\ \hline 14928 \end{array}$$

1 is ignored

(2) take 2nd

10's Complement.

$$9999$$

$$- 0321$$

$$\begin{array}{r} 9999 \\ - 0321 \\ \hline 9678 \end{array}$$

$$1$$

$$+ 1$$

$$9679$$

Add (1) + (2) 10's Complement

$$5250$$

$$+ 9679$$

$$\begin{array}{r} 5250 \\ 9679 \\ \hline 14929 \end{array}$$

1 is ignored

$$Ans: 4929$$

b) $\overset{①}{3} \overset{②}{5} 7 0 - 2 1 0 0$

① take 2^{nd} - 9's Complement

$$\begin{array}{r} 9 9 9 9 \\ - 2 1 0 0 \\ \hline 7 8 9 9 \end{array}$$

Now, $① + ②^{\text{nd}}$ 9's Complement

$$3570$$

$$+ 7899$$

$$\hline 11469$$

$$1469 + 1 \Rightarrow 1470$$

② $2^{\text{nd}} \rightarrow 10$'s Complement

$$\begin{array}{r} 9 9 9 9 \\ - 2 1 0 0 \\ \hline 7 8 9 9 \\ + 1 \\ \hline 7 9 0 0 \end{array}$$

Now, $① + ②$ 10's Complement

$$3570$$

$$+ 7900$$

$$\hline 11470 \rightarrow 1470$$

take it
ignored

c)

$$753 - 864$$

① take 2^{nd} 9's Complement

$$\begin{array}{r} 9 9 9 \\ - 8 6 4 \\ \hline 1 3 5 \end{array}$$

Now, ① + ②nd g's Complement.

753

$$\begin{array}{r} 080 \\ + 135 \\ \hline 888 \end{array}$$

Take - (g's Complement of 888)

999

- 888

$$\hline 1110000 = 1110000$$

Now ② 10's Complement

999

- 864

$$\hline 135 + 1 \Rightarrow 136$$

Add ① + 136

$$753 + 136 \Rightarrow 889$$

there, no Carry So - (10's Complement of 889)

$$999 - 136 = 863$$

- 889

d) 20 - 1000

① ② g's Complement

9999

1000

8999

Now ① + ② g's Complement

0020

+ 8999

90119

→ No Carry

So - (9's Complement of 9019)

$$= - \left(\begin{array}{r} 9 & 9 & 9 & 9 \\ - 9 & 0 & 1 & 9 \\ \hline 0 & 9 & 8 & 0 \end{array} \right) \Rightarrow - 0980$$

② 2 10's Complement

$$\begin{array}{r} 9 & 9 & 9 & 9 \\ - 1 & 0 & 0 & 0 \\ \hline 8 & 9 & 9 & 9 + 1 \Rightarrow 900011 \end{array}$$

take ① + ② 10's Complement

$$\begin{array}{r} 0 & 0 & 2 & 0 \\ + 9 & 0 & 0 & 0 \\ \hline 9 & 0 & 2 & 0 \rightarrow \text{No carry} \end{array}$$

So - (10's Complement of 9020)

$$\begin{array}{r} = \left(\begin{array}{r} 9 & 9 & 9 & 9 \\ - 9 & 0 & 2 & 0 \\ \hline 0 & 9 & 7 & 9 + 1 \end{array} \right) \Rightarrow - 0980 \end{array}$$

Ques-10 Perform the Subtraction with the following binary no. using 1,2's or 2,1's Complement.

a) $11010 - 01101$
 (i) (ii)

① $2 \rightarrow 1$'s Complement

$$\begin{array}{r} 10010 \\ \text{Now } ① + 10010 \\ 11010 \\ 10010 \\ \hline 101100 \end{array} \rightarrow \begin{array}{r} 01100 \\ + 01101 \\ \hline 01101 \end{array}$$

(2) $2 \rightarrow 2's$ Complement $\Rightarrow 10010 + 1 \Rightarrow 10011000 - ?$

$$10010 + 1 \Rightarrow 10011000 - ?$$

Now, $\textcircled{1} + 10011$

$$\begin{array}{r} 11010 \\ + 10011 \\ \hline \end{array}$$

$$10011$$

$$101101$$

$101101 \rightarrow \text{Carry is ignored}$

$$\begin{array}{r} 1101 \\ + 101101 \\ \hline \end{array}$$

(i) (ii)

b) $11010 - 10000$

(1) $\textcircled{2} \rightarrow 1's$ Complement $\rightarrow 01111$

$$\begin{array}{r} 11010 \\ + 01111 \\ \hline \end{array}$$

$$+ 01111$$

$(101110001 \text{ result from } 11010 + 01111)$

So,

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

$$+ 1$$

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

(2) $\textcircled{2} \rightarrow 2's$ Complement $\rightarrow 01111 + 1 \Rightarrow 10000$

Now $\textcircled{1} + 10000$

$$\begin{array}{r} 11010 \\ + 10000 \\ \hline \end{array}$$

$$+ 10000$$

$$\begin{array}{r} 101010 \\ + 1 \\ \hline \end{array} \Rightarrow 01010$$

ignored

c) $10010 - 10011$

(1) (i) (ii)

$\textcircled{2} \rightarrow 1's$ Complement.

$$01100$$

add $\textcircled{1} + 01100$.

$$\begin{array}{r} 10010 \\ + 01100 \\ \hline \end{array}$$

$$+ 01100$$

$$\begin{array}{r} 11110 \\ + 1 \\ \hline \end{array}$$

Q. - (1's Comp. of 11110)

$$= -0001$$

(2)

② → 2's Comp.

$$= 01100 + 1 = 01101$$

Now ① + 01101

$$\begin{array}{r} 10010 \\ + 01101 \\ \hline 11111 \end{array}$$

No, Carry

Q. - (2's Complement of 11111).

$$= -(00000 + 1)$$

$$= -(00001)$$

d)

$$(i) \quad 100 - 110000$$

$$01010$$

(1)

② → 1's Complement

$$00111$$

Now ① + 00111

$$000100$$

$$001111$$

$$\hline 010011 \rightarrow \text{No Carry}$$

Q. - (1's Comp. of 010011)

$$= -(101100)$$

(2)

2 → 1's Comp.

$$001111 + 1 = 010000$$

Now ① + 010000

$$000100$$

$$+ 010000$$

$$\hline 010100 \rightarrow \text{No Carry}$$

$= - (2\text{'s Comp. of } 010100)$

$$101011 + 1 \approx 101100$$

$$\Rightarrow -(101100)$$

Ques-11 Represent the decimal no. 8620

a) BCD.

$$\begin{array}{cccc} 8 & 6 & 2 & 0 \\ 1000 & 0110 & 0010 & 0000 \end{array}$$

$$\Rightarrow (1000011000100000)_{BCD}$$

b) Excess-3 code

$$\begin{array}{cccc} 8 & 6 & 2 & 0 \\ 3 & 3 & 3 & 3 \\ \hline 11 & 9 & 5 & 3 \end{array}$$

$$\begin{array}{cccc} 1 & 1 & 9 & 5 \\ 1011 & 1001 & 0101 & 0011 \end{array}$$

$$\Rightarrow (1011100101010011)_{\text{Excess-3}}$$

c) 2,4,2,1

$$\begin{array}{cccc} 8 & 6 & 2 & 0 \\ 1110 & 1100 & 0010 & 0000 \end{array}$$

$$(1110110000100000)_{2421}$$

d) Binary no.:

$$2 \ 8620$$

$$2 \ 4310 \ 0$$

$$2 \ 2156 \ 0$$

$$2 \ 1077 \ 1$$

2 538

2 269 0 $\Rightarrow (8620)_0 =$

2 134 1

2 67 0 $(10000110101100)_2$

2 33 1

2 18 1

2 8 0

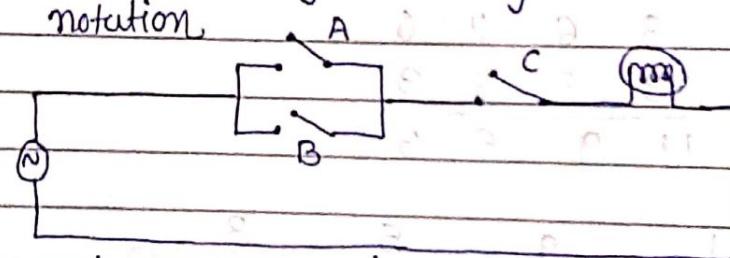
2 4 0

2 2 0

2 1 0

0 1

Que-12 Express the following Switching Circuit in binary Logic notation.



A	B	C	out
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

A/B	00	01	11	10	01
0	0	1	5	2	4
1	4	5	7	6	3

$$= BC + AC$$

$$= (A+B)C$$

Assignment-2* Boolean Algebra & Logic Gates *

Ques-1 Simplify the following boolean functions to a minimum number of literals.

$$(a) xy + xy'$$

$$= x(y+y') \quad (\because (y+y')=1)$$

$$= x$$

$$(b) (x+y)(x+y')$$

$$= (xx + xy^2 + yx + yy')$$

$$= x + xy' + xy + 0$$

$$= x + x(y+y')$$

$$= x$$

$$(c) xyz + x'y + xyz'$$

$$= (xyz + xyz') + x'y \Rightarrow xy(z+z') + x'y$$

$$= xy + x'y \Rightarrow y(x+x') \Rightarrow y$$

$$(d) zx + zx'y$$

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

$$= z(x+y) \Rightarrow xz + yz \Rightarrow z(x+y)$$

$$(e) (A+B')(A'+B')'$$

$$= (A+B')(A+B) \rightarrow (AA + AB + AB' + BB')$$

$$= A + AB + AB' + 0 \Rightarrow A + A(B+B')$$

$$= A$$

$$(f) y(wz' + wz) + xy$$

$$= y(w(z+z')) + xy = y(w) + xy$$

$$= yw + xy \Rightarrow y(w+x)$$

$$\text{G) } (A+B)'(A'+B')' \\ = (A'B')(A'+B')' \\ = 0 \quad (AB) = 0$$

Que-2 Reduce the following boolean expressions to the required number of literals.

a) $ABC + A'B'C' + A'BC + ABC' + A'B'C'$ to five literals
 $= AB(C+C') + A'B'(C+C') + A'BC$
 $= AB + A'B' + A'BC$
 $= AB + A'(B'+BC) = AB + A'B' + A'BC$

b) $BC + AB' + AB + BCD$ to four literals
 $= BC + BCD + A(C'+B)$
 $= BC(C+D) + AC' + AB$
 $= BC + A(C'+B)$

c) $[(CD') + A]' + A + CD + AB$ to three literals
 $= [C'D + (A'+A)] + CD + AB$
 $= D[C + C'] + AB$
 $= AB + D$

Que-3 Find the complement of the following boolean functions & reduce them to a minimum no. of literals.

a) $F = (BC' + A'D)(AB' + CD')$ $(xy)' = x' + y'$
 $f' = ((BC' + A'D) \cdot (AB' + CD'))'$ $(x+y)' = \bar{x} \cdot \bar{y}$

$$\begin{aligned} F' &= ((BC' + A'D)' + (AB' + CD')') \\ &= (BC')' \cdot (A'D)' + (AB')' \cdot (CD')' \\ &= (B' + C) \cdot (A + D') + (A' + B) \cdot (C' + D) \end{aligned}$$

$$\begin{aligned}
 &= B'D + B'D' + CA + CD' + A'C' + A'D + BC' + BD \\
 &= B'(A+D') + C(A+D') + A'(C'+D) + B(C'+D)
 \end{aligned}$$

b) $F = B'D + A'BC' + ACD + A'BC$

$$\begin{aligned}
 F' &= ((B'D) \cdot (A'BC') \cdot (ACD) \cdot (A'BC))' \\
 &= (B'+D)' \cdot (A'+B+C')' \cdot (A'+C'+D)' \cdot (A+B'+C')' \\
 &= (B+D) \cdot (A+B'+C) \cdot (A'+C'+D) \cdot (A+B'+C')
 \end{aligned}$$

Que-4 Obtain the truth table of the function

$$\begin{aligned}
 f = xy + xy' + y'z &\Rightarrow xy(z+z') + xy'(z+z') + (x+x')y'z \\
 &= xyz + xyz' + xy'z + xy'z' + \\
 &\quad xy'z + x'y'z
 \end{aligned}$$

x	y	z	y'	xy	xy'	$y'z$	$xyz + xy' + y'z$
0	0	0	1	0	0	0	0
0	0	1	1	0	0	1	1
0	1	0	0	0	0	0	0
0	1	1	0	0	0	0	0
1	0	0	1	0	1	0	1
1	0	1	1	0	1	1	1
1	1	0	0	1	0	0	1
1	1	1	0	1	0	0	1

$$f(x,y,z) = xy + xy' + y'z = \Sigma(1,4,5,6,7)$$

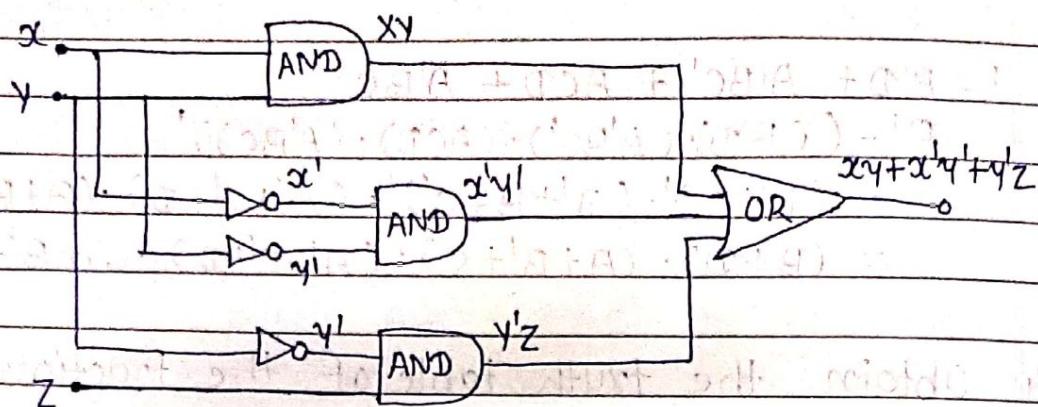
Truth Table:

	x	y	z	00	01	11	10	
0	0	0	0	0	0	0	1	in terms of SOM
1	1	1	0	1	1	1	0	for POM = \pi(0,2,3)

Que-5 Given the boolean function :

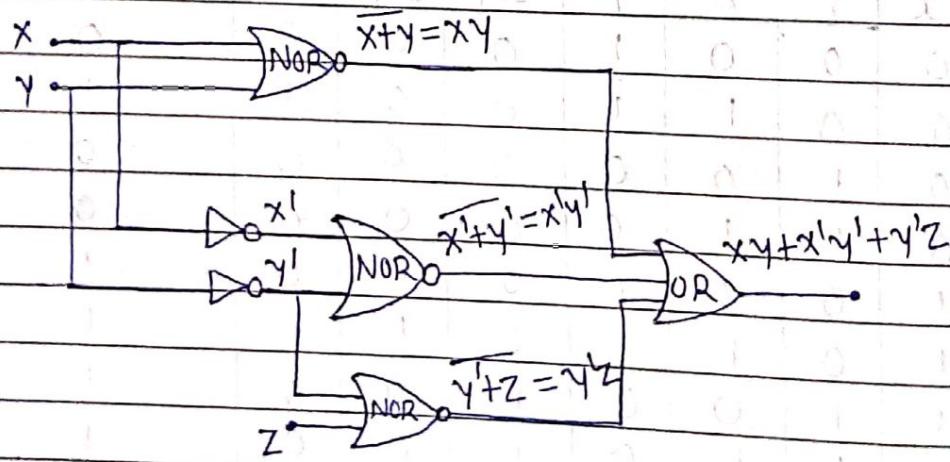
$$f = xy + x'y' + y'z$$

a) Implement it with AND, OR & NOT gates.



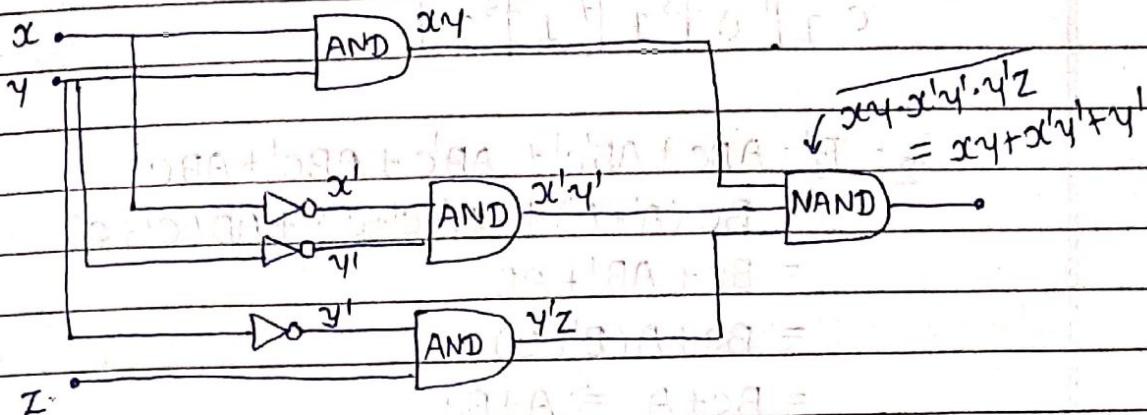
b) Implement it with only OR & NOT gates.

$$f = xy + x'y' + y'z$$



c) Implement it with only AND & NOT gates.

$$f = xy + x'y' + y'z$$



Que. 6 Simplify the functions T_1 & T_2 to a minimum no. of literals.

A	B	C	T_1	T_2
0	0	0	1	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

	AB	$A'B'$	$A'B$	AB	AB'
C	00	01	11	10	
C'	0	1	1	0	0
C	1	1	0	0	0

$$\begin{aligned}
 T_1 &= A'B'C' + A'B'C + A'BC' \\
 &= A'B'(C' + C) + A'BC' \\
 &= A'B' + A'BC' \Rightarrow (A'B' + A'B)(A'B + C') \\
 &= A'(B' + B)(A'B + C') \\
 &= A'(A'B + C') = A'A'B + A'c' \\
 &= A'B + A'c' \Rightarrow A'(B + c')
 \end{aligned}$$

C	AB	A'B'	A'B	AB	AB'
00	00	01	11	10	10
For T ₂ :	C' 0	0 0	2 0	6 1	4 1
	C 1	1 0	3 1	7 1	5 1

$$\begin{aligned}
 T_2 &= T_1' = A'BC + AB'C' + AB'C + ABC' + ABC \\
 &= BC(A' + A) + AB'(C' + C) + AB(C' + C) \\
 &= BC + AB' + AB \\
 &= BC + A(B' + B) \\
 &= BC + A \Rightarrow A + BC
 \end{aligned}$$

Que-7 Express the following functions in a sum of minterms & product of maxterms.

a) $f(A, B, C, D) = D(A' + B) + B'D$

* Method-1

A	B	B'	C	D	A'	DA'	DB	DB'	DA' + DB + BD
0	0	1	0	0	1	0	0	0	0
0	0	1	0	1	1	1	0	1	1
0	0	1	1	0	1	0	0	0	0
0	0	1	1	1	1	1	0	1	0
0	1	0	0	0	1	0	0	0	1
0	1	0	0	1	1	1	1	0	0
0	1	0	1	0	1	0	0	0	1
0	1	0	1	1	1	1	1	1	0
1	0	1	0	0	0	0	0	0	1
1	0	1	0	1	0	1	0	1	0
1	0	1	1	0	0	0	0	0	1
1	0	1	1	1	0	1	0	1	0
1	1	0	0	0	0	0	0	0	1
1	1	0	0	1	0	1	0	0	0
1	1	0	1	0	0	0	1	0	1
1	1	0	1	1	0	1	1	1	0
	AB		00	01	11	10			

CD	00	01	11	10
	0	4	12	8

For SOM: $\Sigma(1, 3, 5, 7, 9, 11, 13, 15)$

00	0	0	0	0
----	---	---	---	---

OR

01	1	1	1	1
----	---	---	---	---

$$= A'B'C'D + A'B'C'D + A'B'C'D +$$

$$A'B'C'D + A'B'C'D + A'B'C'D +$$

$$A'B'C'D + A'B'C'D$$

11	1	1	1	1
----	---	---	---	---

for POM: $\Pi(0, 2, 4, 6, 8, 10, 12, 14)$

10	2	6	14	10
	0	0	0	0

b) $f(w,x,y,z) = y'z + wxy' + wxz' + w'x'z$

* Method - 2

$$f(w,x,y,z) = (y'z)(x+x') (w+w') + wxy'(z+z')$$

$$+ wxz'(x+x') + w'x'z'(y+y')$$

$$= \underset{1101}{wx'y'z} + \underset{0001}{w'x'y'z} + \underset{1101}{wx'y'z} + \underset{1100}{wxy'z'} + \underset{1110}{wx'yz'} +$$

$$\underset{1100}{wx'y'z'} + \underset{0010}{w'x'yz'} + \underset{0000}{w'x'y'z'}$$

$$= \begin{matrix} 1101 & 0001 \\ \downarrow & \downarrow \\ 8401 \end{matrix}$$

$$\text{for } \text{SOM} = f(w,x,y,z) = \Sigma (13, 1, 12, 14, 2, 0)$$

$$\text{OR } \Sigma (0, 1, 2, 12, 14, 13)$$

$$wx'y'z + w'x'y'z + wx'y'z' + wx'yz' + \\ w'x'yz + w'x'y'z'$$

$$\text{for } \text{POM} = f(w,x,y,z) = \Pi (3, 4, 5, 6, 7, 8, 9, 10, 11, 15)$$

c) $f(x,y,z) = \overline{x} + \overline{y} \cdot \overline{z}$

$$\text{for } \text{SOM} = (x,y,z) = \Sigma (0, 1, 2, 3, 4, 5, 6, 7)$$

$$\text{for } \text{POM} = f = (x,y,z) = \Pi$$

Ques.-8 Convert the following to the Order Connociel term

(a) $f(x,y,z) = \Sigma (1, 3, 7)$

$\rightarrow f(x,y,z) = \Pi m (0, 2, 4, 5, 6)$

$$(b) f(x,y,z) = \Sigma_m(0,2,6,11,13,14)$$

$$f(x,y,z) = \Pi_m(1,3,4,5,7)$$

$$(c) f(x,y,z) = \Pi(0,3,6,7)$$

$$f(x,y,z) = \Sigma_m(1,2,4,5)$$

Que-9. Obtain the Simplified in sum of products for the following boolean functions.

$$(a) f(x,y,z) = \Sigma(2,3,6,7)$$

	xy	00	01	11	10
z	0	0	1	1	0
	1	0	1	1	0

$$= \Sigma(2,3,6,7) \Rightarrow x'y'z + x'y'z + xyz' + xy'z$$

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

$$(b) f(A,B,C,D) = \Sigma(7,13,14,15)$$

AB	00	01	11	10	
D	0	4	12	8	
CD	00				
	1	5	13	9	
01			1		
	3	7	5	11	
11		1	1		
	2	6	4	10	
10			1		

$$\Rightarrow ABC'D + A'BCD +$$

$$ABC'D' + ABCD$$

$$f = \Rightarrow DAB + BCD + CAB$$

Ques. 10

Obtain the Simplified expression in SOP for the following boolean f^n .

$$a) xy + x'y'z' + x'yz'$$

→ method: 1

x	y	z	x'	y'	z'	xy	$x'y'z'$	$x'yz'$	$xy + x'y'z' + x'yz'$
0	0	0	1	1	1	0	1	0	1
0	0	1	1	1	0	0	0	0	0
0	1	0	1	0	1	0	0	1	1
0	1	1	1	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0	0
1	0	1	0	1	0	0	0	0	0
1	1	0	0	0	1	1	0	0	1
1	1	1	0	0	0	1	0	0	1

$$(x,y,z) = (0,0,0) \Rightarrow (0,1,1) = 7$$

x	00	01	11	10
z	0	1	1	1
y	0	1	1	0
	1	3	7	5

$$f = (x,y,z) = \Sigma m(0,2,6,7)$$

→ method: 2

$$\begin{array}{ccccccc}
 f & = & xyz' & + & xyz & + & x'y'z' & + & x'yz' \\
 & & 110 & & 111 & & 000 & & 010 \\
 & & 6 & & 7 & & 0 & & 2
 \end{array}$$

$$= \Sigma m(0,2,6,7)$$

$$b) A'B + Bc' + B'c'$$

$$\Rightarrow A'B(c+c') + (A+A')Bc' + B'c'(A+A')$$

$$\begin{array}{ccccccc}
 & & A'Bc & + & A'Bc' & + & ABC' \\
 & & 3 & & 2 & & 6
 \end{array}$$

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

A	B	C	A'	B'	C'	$A'B$	BC'	$B'C'$	$A'B + BC' + B'C'$
0	0	0	1	1	1	0	0	0	1
0	0	1	1	1	0	0	0	0	0
0	1	0	1	0	1	1	1	0	1
0	1	1	1	0	0	1	1	0	1
1	0	0	0	1	1	0	0	1	1
1	0	1	0	1	0	0	0	0	0
1	1	0	0	0	1	0	0	0	0
1	1	1	0	0	0	0	0	0	0

	$A'B'$	$A'B$	AB	AB'
c'	0	1	2	3
c	0	1	2	3
	0	1	2	3

$$\Rightarrow c' + A'B$$

$$f = (A, B, C) = \Sigma_m(0, 3, 2, 4, 6)$$

Ques-11 Obtain the Simplified expressions in sop for the following boolean functions.

(a) $D(A'+B) + B'(C+AD)$

	$A'B'$	$A'B$	AB	AB'
$c'D$	0	4	12	8
$c'D'00$	0	1	2	3
$c'D'01$	1	1	1	1
$c'D'11$	1	1	1	1
$c'D'10$	2	6	14	10

$$\therefore F(A, B, C, D) = D + CB'$$

$$(b) ABD + A'C'D' + A'B + ACD' + AB'D$$

A	B	C	D	A'	D'	AD	AB'	$A'D$	$AD + A'D' + A'B$
0	0	0	0	1	1	0	1	0	1
0	0	1	0	1	0	0	0	0	0
0	0	0	1	1	1	0	1	0	1
0	0	1	1	1	0	0	0	0	0
0	1	0	0	1	1	0	1	1	1
0	1	1	0	1	0	0	0	1	1
0	1	0	1	1	1	0	1	1	1
0	1	1	1	1	0	0	0	1	1
1	0	0	0	0	1	0	0	0	0
1	0	1	0	0	0	1	0	0	1
1	0	0	1	0	1	0	0	0	0
1	0	1	1	0	0	1	0	0	1
1	1	0	0	0	1	0	0	0	0
1	1	1	0	0	0	1	0	0	1
1	1	0	1	0	1	0	0	0	0
1	1	1	1	0	0	1	0	0	1

		AB	00	01	11	10
		CD	00	01	11	10
00	0	0	1	1		
00	1	1	1	1		
01	0	1	1	1	1	1
01	1	1	1	1	1	1
11	0	1	1	1	1	1
11	1	1	1	1	1	1
10	0	2	6	1	14	10
10	1	1	1	1		

$$f(A, B, C, D) \neq AD + A'B + AB'$$

Ques-12 Obtain the Simplified expressions in product of sum,

(a) $f(x,y,z) = \prod(0,1,4,5) \rightarrow$ Sum of maxterm.

	00	01	11	10	010	110	001	101	111	011	100
z	0	2	6	4	010	110	001	101	111	011	100
z'	1	3	7	5	110	010	100	000	111	010	101
	0	1	7	5	0	1	0	1	1	0	1

$$f(x,y,z) \Rightarrow y$$

(b) $f(A,B,C,D) = \prod(0,1,2,3,4,10,11)$

	AB	AB'	A'B'	A'B	0D	01	11	10	101	110	111
CD	00	4	12	8	0	1	2	3	10	11	111
	00	0	0	0	1	0	0	0	0	0	0
	01	1	5	13	9	0	1	0	1	1	1
	11	0	7	15	11	0	0	0	0	0	0
	10	2	6	14	10	0	1	1	0	0	0

$$f(A,B,C,D) = (A+B)(A+C+D)(C'+B)$$

Ques-13

Simplify the boolean function f in SOP using the don't care conditions:

$$(a) f = y' + x'z' \quad d = yz + xy$$

x	y	y'	z	$x'z'$	$y' + x'z'$	yz	xy	$yz + xy$
0	0	1	0	1	1	0	0	0
0	0	1	1	0	1	0	0	0
0	1	0	0	1	1	0	0	1
0	1	0	1	0	0	1	0	1
1	0	1	0	0	1	0	0	0
1	0	1	1	0	1	0	0	0
1	1	0	0	0	0	0	1	1
1	1	0	1	0	0	1	1	1

	$x'y$	00	01	11	10
0	0	1	2	6	4
1	1	1	3	7	5

$$F(x,y,z) = 1$$

$$(b) f = B'C'D' + BCD' + ABCD'$$

$$d = B'CD + A'B'C'D'$$

$$F = B'C'D' + BC'D + ABCD' \quad d = B'C'D + A'B'C'D$$

A	B	C	D	$B'C'D'$	BCD'	$ABC'D'$	F	$B'C'D$	$A'B'C'D$	d
0	0	0	0	1	0	0	1	0	0	0
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	1	0	X
0	1	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	X
0	1	1	0	0	1	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0
1	0	0	0	1	0	0	1	0	0	0
1	0	0	1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0
1	1	1	0	0	1	1	1	0	0	0
1	1	1	1	0	0	0	0	0	0	0

AB'	$A'B'$	AB	$A'B$	$AB' + AB$
CD	00	01	11	10
$C'D'$	00	1	12	8
$C'D$	01	1	5	9
CD	11	3	9	15
CD'	10	2	6	14

$$F = (ABC'D) = C'D'B' + BCD'$$

Assignment-3

Page No.

DATE: / /

Que.-1

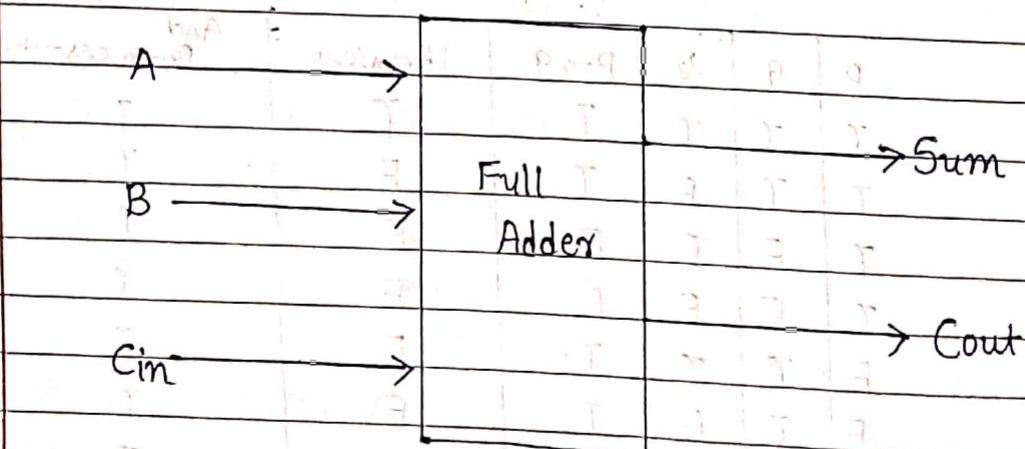
Explain Full Adder in details, with circuit diagram

→ Full Adder is the adder which adds three inputs and produces two outputs.

Inputs → first two inputs are A and B third input is Cin [input carry]

Outputs → Output Carry Cout
Sum (Sum Output)

- A full Adder logic is designed in such a manner that it can take eight inputs together to create a byte-wide adder and cascade the carry bit from one adder to the another.



Full Adder : Truth Table

Inputs			Outputs	
A	B	Cin	Sum	Cout
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

• Simplified expressions :

→ K-map for Cout

Cout

Cin \ AB	AB'	A'B	AB	AB'	
Cin	0	2	1	6	4
Cin	1	1	3	1	7

$$\text{Cout} = BC_{\text{in}} + AB + AC_{\text{in}}$$

→ K-map for Sum

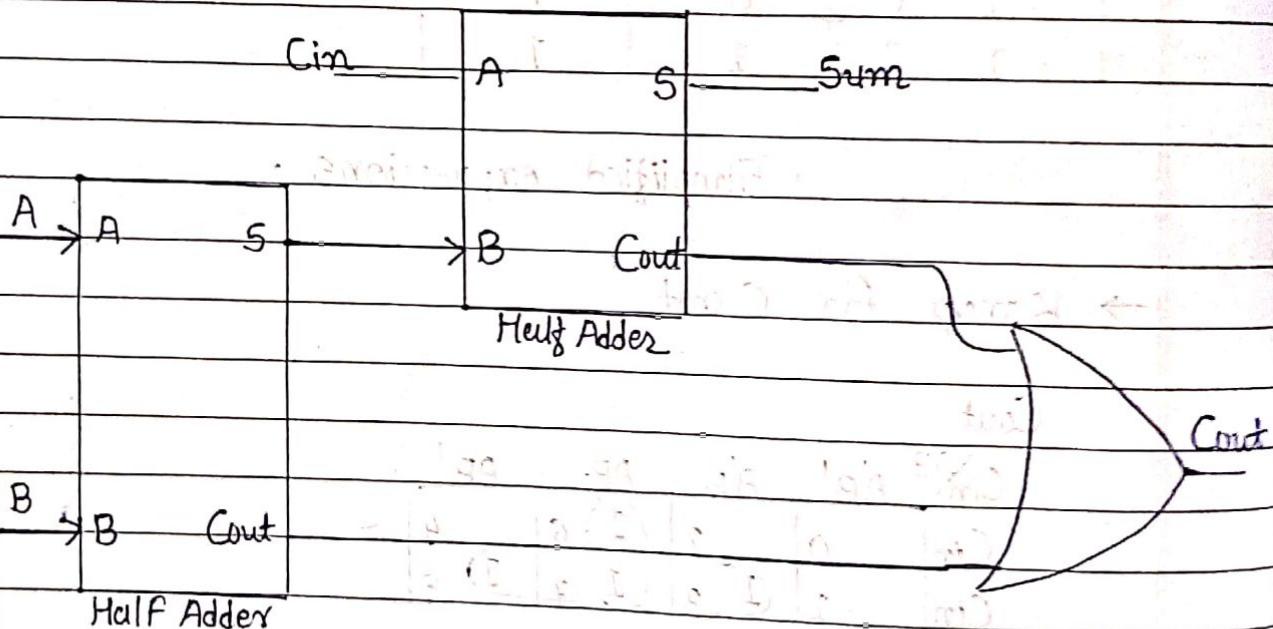
Cin \ AB	AB'	A'B	AB	AB'
Cin	0	2	6	4
Cin	1	3	7	5

50,

$$\begin{aligned}
 \text{sum} &= A'B'C_{in} + A'BC'_{in} + ABC_{in} + ABC'_{in} \\
 &= C_{in}(A'B^1 + AB) + C_{in}(A'B + AB') \\
 &= C_{in} \cdot X\text{-OR}(A \cdot X\text{-OR} B) \\
 &= A \oplus B \oplus C
 \end{aligned}$$

→ Implementation using Half Adders

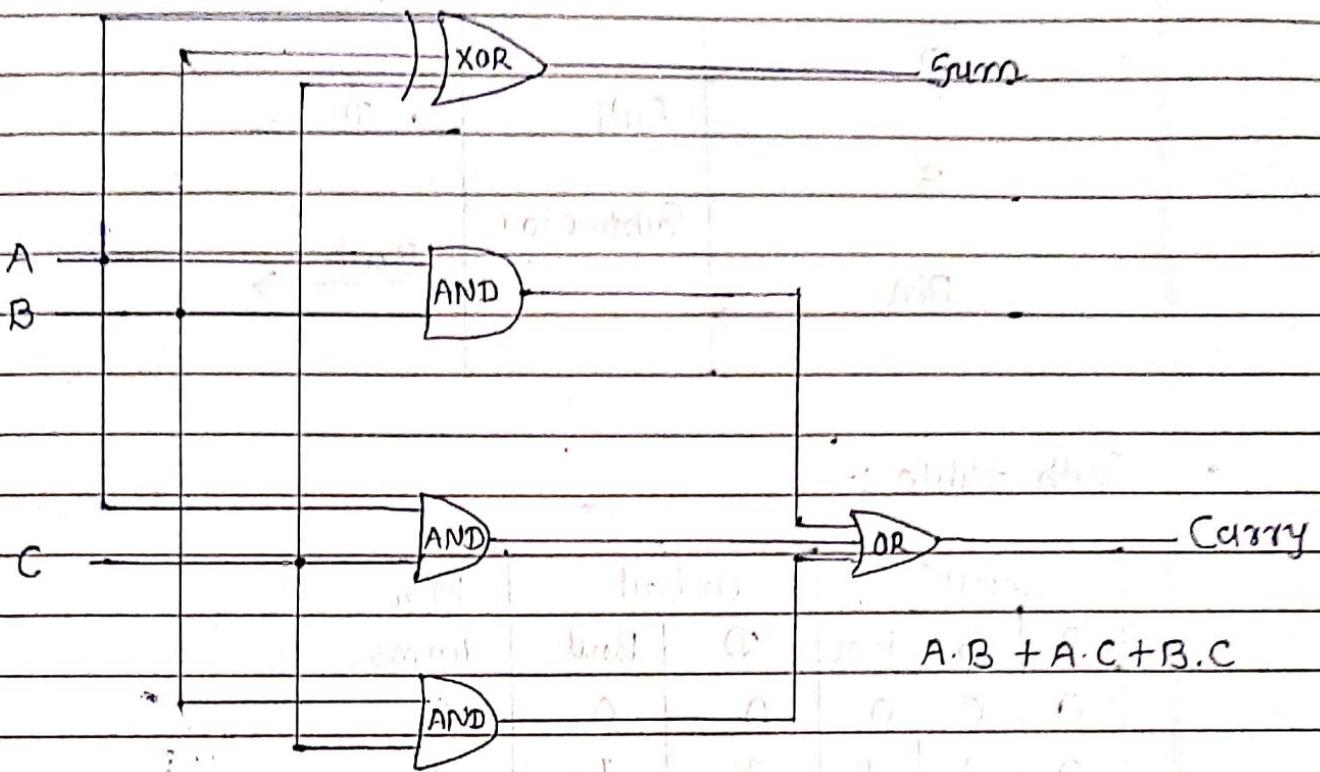
Required : 2 Half Adders
OR gate (1)



→ With this logic circuit, two bits can be added together, taking a carry from the next lower order of magnitude

→ Implementation using Ex-OR gates.

Required : there input Ex-OR gate (1) for sum
(2) input AND (3) for carry.
(4) input OR (5) for carry.



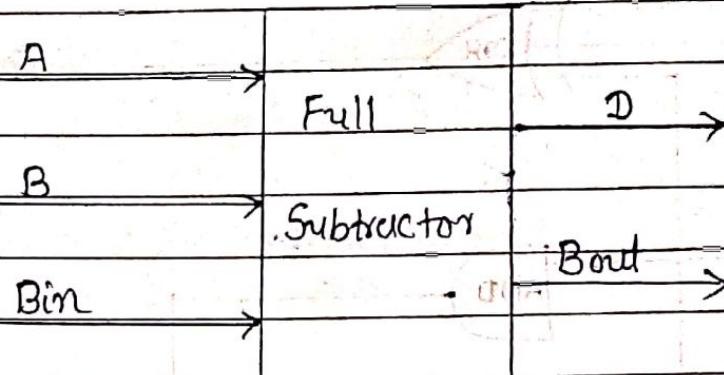
Ques-2 Explain Full Subtractor in details.

→ A full Subtractor is a Combinational Circuit that performs subtraction of two bits. One is minuend and other is subtrahend, taking into account borrow of the previous adjacent lower minuend bit.

→ This circuit has three inputs and two outputs.

inputs → A, B, Bin (Borrow in)

Output → D, Bout



• Truth table :-

	Input			Output		Min terms
	A	B	Bin	D	Bout	
1	0	0	0	0	0	0
2	0	0	1	1	1	1
3	0	1	0	1	1	2
4	0	1	1	0	1	3
5	1	0	0	1	0	4
6	1	0	1	0	0	5
7	1	1	0	0	0	6
8	1	1	1	1	1	7

→ K-map for Bout :-

		Bout					
		AB					
		Bin ²	0	1	2	3	4
		Bin	0	1	2	3	4
		1	1	1	1	1	1
		0	0	0	0	0	0

$$\begin{aligned}
 \text{So } \text{Bout} &= A'B' + A'B + BB' \\
 &= \text{Bin} (A \times \text{NOR } B) + A'B \\
 &= \text{Bin} (A \times \text{OR } B)' + A'B
 \end{aligned}$$

→ K map for difference :-

	AB	$A'B'$	$A'B$	AB	$B'A'$	
$B'm$	0	1	2	6	14	
$B'm$	1	1	3	7	5	

$$D = A'B'B'm + A'BB'm + ABB'm + AB'B'm$$

$$= B'm(A'B' + AB) + B'm'(AB' + A'B)$$

$$= B'm(A \oplus B) + B'm'(A \oplus B)$$

$$= B'm(A \oplus B)' + B'm'(A \oplus B)$$

$$= B'm \oplus (A \oplus B)$$

$$D = A \oplus B \oplus B'm$$

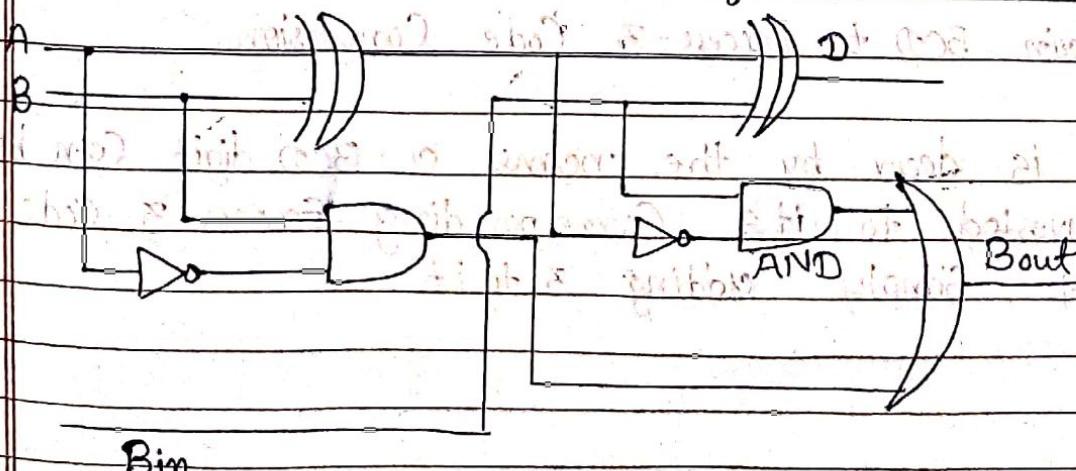
→ Implementation using two half Subtractor

Required : 2 Half Subtractor

OR gate (1)

Half Subtractor

Half Subtractor

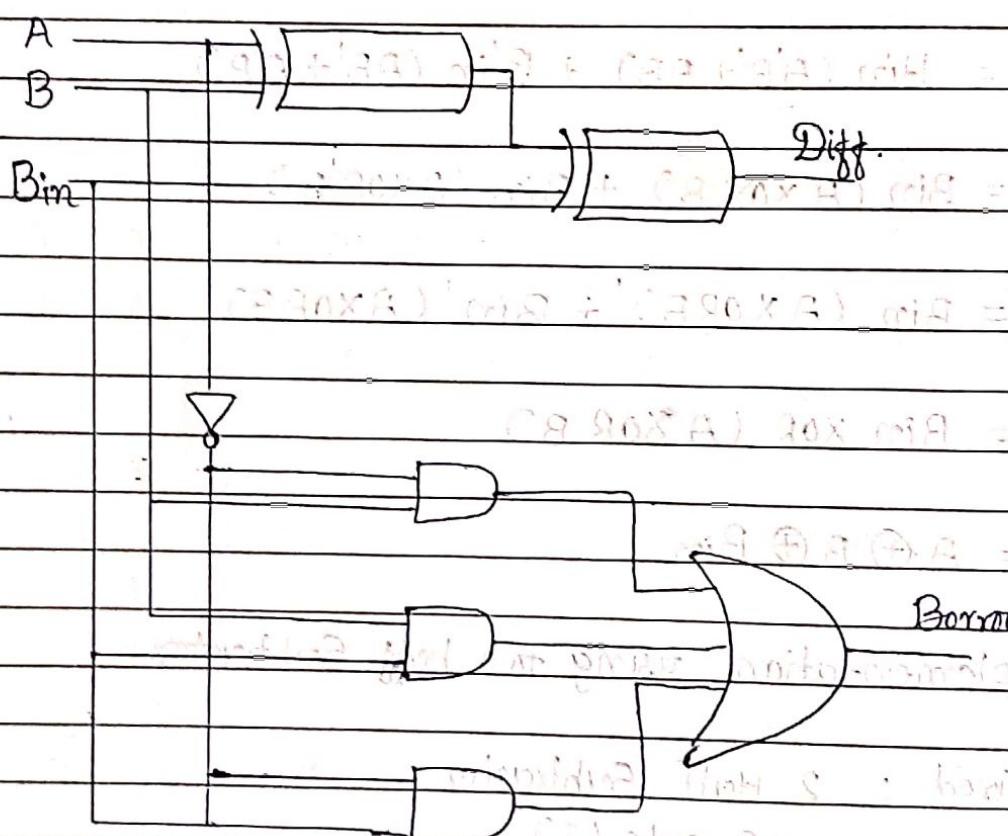


→ Implementation Using Ex-OR gate.

Required : 2 input AND gate (3)

NOT gate (1)

3 input OR gate (1)



Que-3 Explain BCD to Excess-3 Code Conversion.

→ As is clear by the name, a BCD digit can be converted to its corresponding Excess-3 code by simply adding 3 to it.

- Let w, x, y, z be the bits representing the binary numbering where w is the LSB and z is the MSB. and let A, B, C, D be the bits representing the gray code of the binary numbers where A is the LSB and D is the MSB.
- The truth table for the conversion is given below.
- The X's mark don't care conditions.
- To find the corresponding digital circuit, we will use the K-map technique for each of the Excess-3 code bits as output with all of the bits of the BCD number as input.

BCD	E-3	Minterms						
A	B	C	D	w	x	y	z	
0	0	0	0	0	0	1	1	0
0	0	0	1	0	1	0	0	1
0	0	1	0	0	1	0	1	2
0	0	1	1	0	1	1	0	3
0	1	0	0	0	1	1	1	4
0	1	0	1	1	0	0	0	5
0	1	1	0	1	0	0	1	6
0	1	1	1	1	0	1	0	7
1	0	0	0	1	0	1	1	8
1	0	0	1	1	1	0	0	9
1	0	1	0	X	X	X	X	10
1	0	1	1	X	X	X	X	11
1	1	0	0	X	X	X	X	12
1	1	0	1	X	X	X	X	13
1	1	1	0	X	X	X	X	14
1	1	1	0	X	X	X	X	15

K map for X.

$$\text{Min term } CD \text{ with value 1 is highlighted. So, } X = BC'D' + BC + B'C'D$$

AB	CD	$C'D'$	CD	CD'
$A'B'$	0	1 ₄	1 ₃	1 ₂
$A'B$	1 ₄	5	7	6
AB	X			
AB'		1	X	X

K map for Y.

$$\text{Min terms } CD \text{ and } C'D' \text{ with value 1 are highlighted. So, } Y = CD + C'D'$$

AB	CD	$C'D'$	CD	CD'
$A'B'$	1 ₀	1 ₃	1 ₂	
$A'B$	1 ₄	5	7	6
AB	X ₂	X ₃	X ₅	X ₄
AB'	1 ₈	X ₉	X ₁₁	X ₁₀

K map for Z.

$$\text{Min term } D \text{ with value 1 is highlighted. So, } Z = D'$$

AB	CD	$C'D'$	CD	CD'
$A'B'$	1 ₀	1	1 ₃	1 ₂
$A'B$	1 ₄	5	7	1 ₆
AB	X ₁₂	X ₁₃	X ₁₅	X ₁₄
AB'	1 ₈	9	X ₁₁	X ₁₀

K map for W.

$$\text{Min terms } A, BC, BD \text{ with value 1 are highlighted. So, } W = A + BC + BD$$

AB	CD	0	1	2	3
$A'B'$	0	1	1 ₂	3	
$A'B$	4	2 ₅	1 ₇	1 ₆	
AB	X ₁₂	X ₁₃	X ₁₅	X ₁₄	
AB'	1 ₈	2 ₉	X ₁₁	X ₁₀	

PC of this logic to binary conversion

D

CD

C

A

B

C+D

X

W

A truth table is made out to get the logic for adder and it is as follows:

Ques-4 Short note On BCD Addressing and Yields

- BCD Stand for binary Coded decimal.
- Suppose, We have two 4 bit Numbers A & B.
- The Value of A and B can varies from 0 (0000 in binary) to 9 (1001 in binary) because we are considering decimal numbers.
- The output will varies from 0 to 18. If we are not considering the carry from the previous sum.

- But If we are considering the Carry, then the maximum value of output will be 19.
 $(9+9+1=19)$
- When we are simply adding A and B then we get the binary sum.
- Here to get the output in BCD form then we will use BCD Adder.



→ If the sum of two numbers is less than or equal to 9, then the value of BCD sum and binary sum will be same otherwise they will differ by 6 (0110) in binary.

∴

Truth table :-

Binary Sum BCD Sum

	C'	S_3'	S_2'	S_1'	S_0'	C	S_3	S_2	S_1	S_0
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1
2	0	0	0	1	0	0	0	0	1	0
3	0	0	0	1	1	0	0	0	1	1
4	0	0	1	0	0	0	0	1	0	0
5	0	0	1	0	1	0	0	1	0	1
6	0	0	1	1	0	0	0	1	1	0
7	0	0	1	1	1	0	0	1	1	1
8	0	1	0	0	0	1	0	0	0	0
9	0	1	0	0	1	0	1	0	0	1

10	0	1	0	1	0	0	0	0	0
11	0	1	0	1	1	1	0	0	1
12	0	1	1	0	0	1	0	0	1
13	0	1	1	0	1	1	0	0	1
14	0	1	1	1	0	1	0	1	0
15	0	1	1	1	1	1	0	1	0
16	1	0	0	0	0	1	0	1	0
17	1	0	0	0	1	1	0	1	1
18	1	0	0	1	0	1	1	0	0
19	1	0	0	1	1	1	1	0	0

→ We are adding '0110' ($c=6$) only to the Second half of the table.

→ The Conditions are,

1. if $C'=1$ (Satisfies 16-19)

2. if $S_3', S_2' = 1$ (Satisfies 12-15)

3. if $S_3', S_1' = 1$ (Satisfies 10 and 12)

So our logic is,

$$C' + S_3', S_2' + S_3', S_1' = 1$$

Input

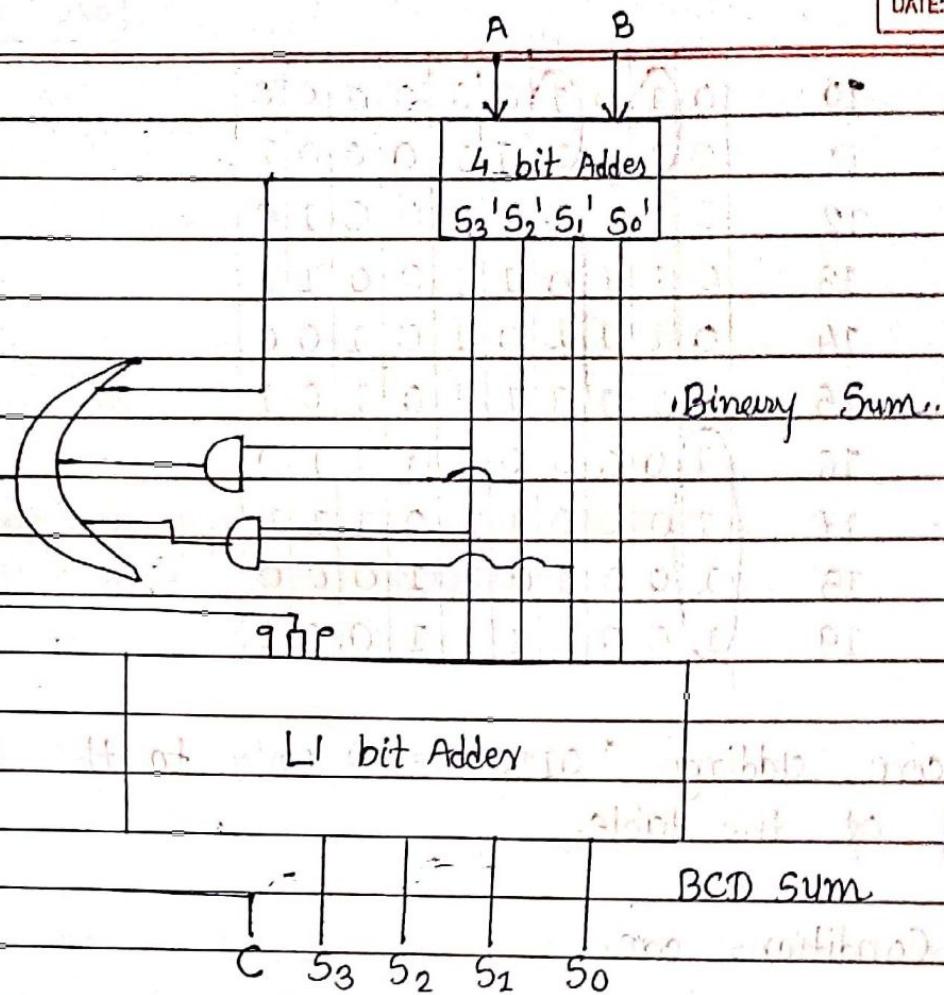
Condition

Output

Condition

Output

Output



Ques-5 Draw Circuit of 4-bit Comparator.

Ques-6 Comparisons between encoders & decoders.

PARAMETER

ENCODER

DECODER

Input applied

Active input Signal
[Original message
Signal]

Coded binary
input

Output generated

Coded binary
Output

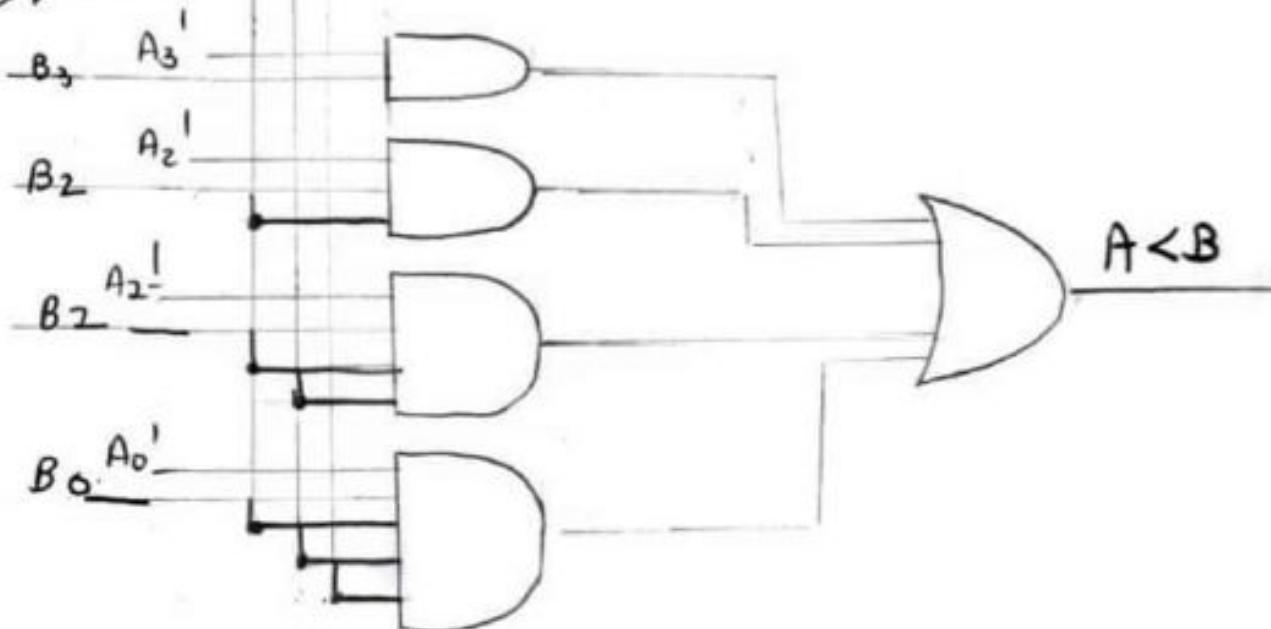
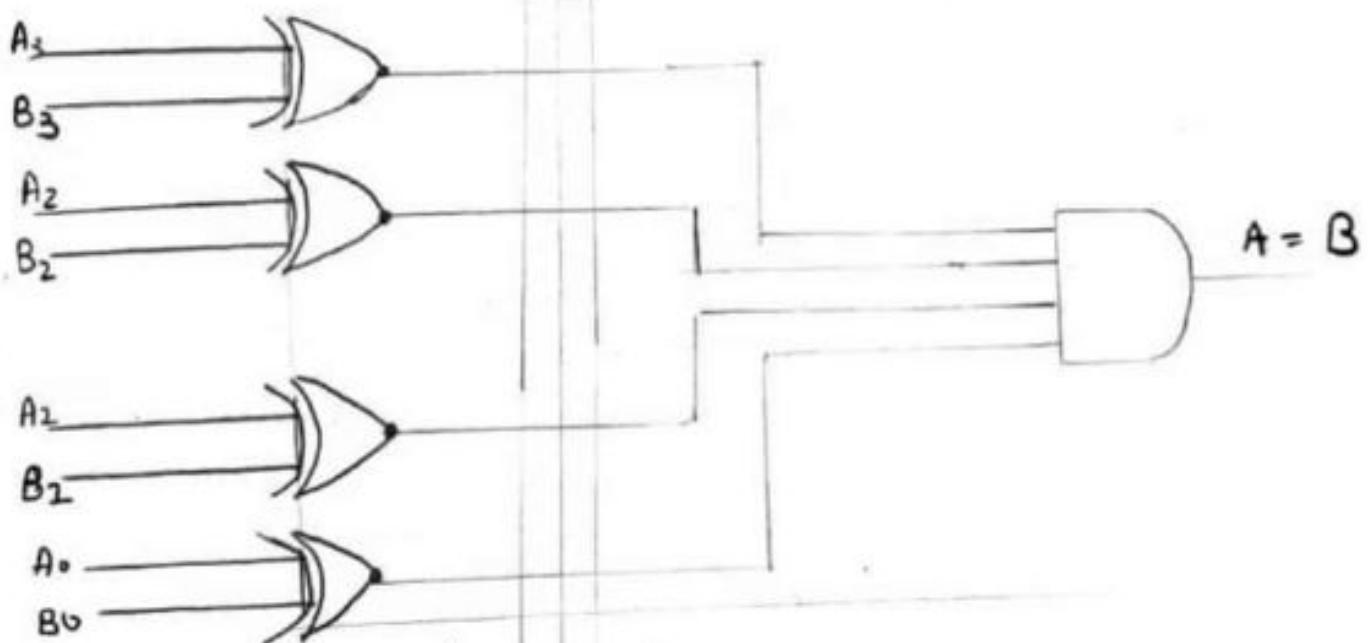
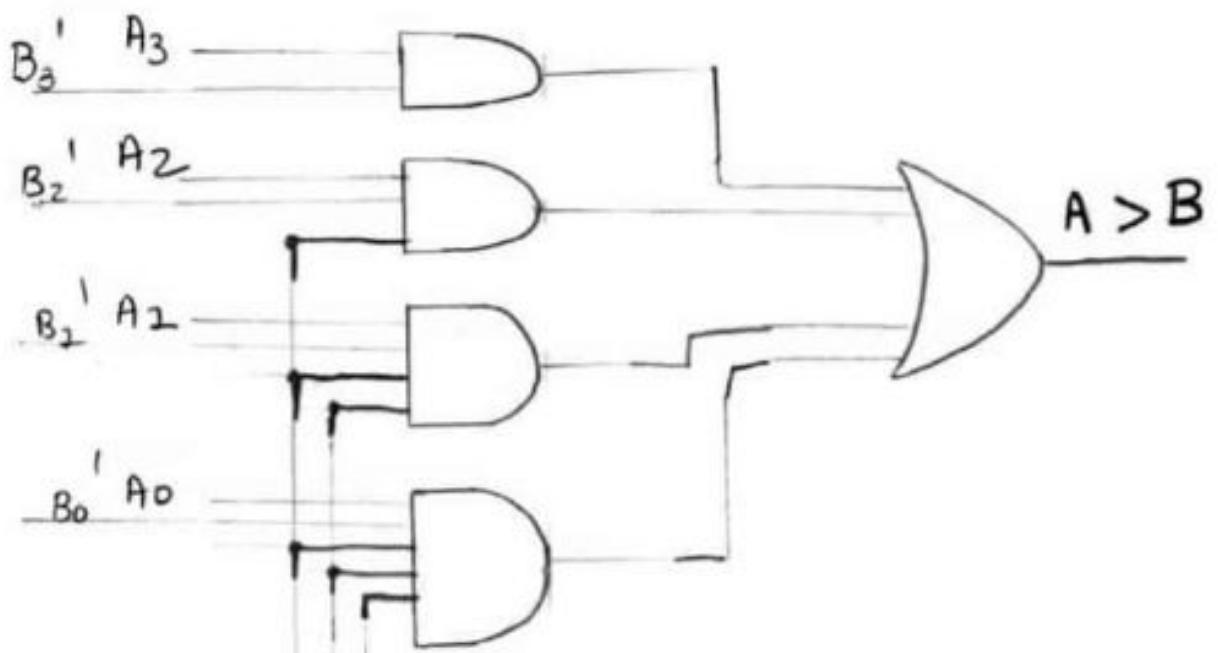
Active Output
Signal (original msg.)

Input lines

2^n

n

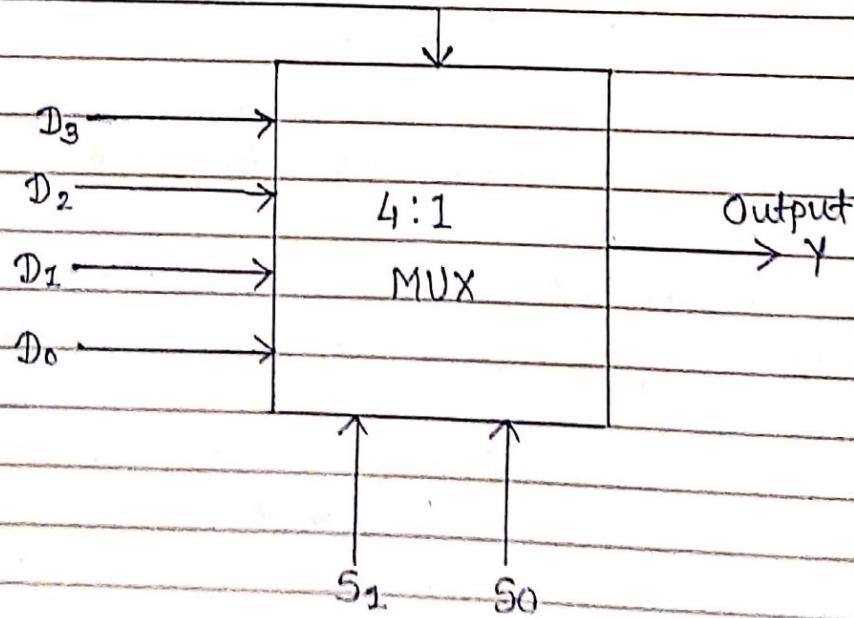
Output Lines	Maximum no. of output lines	2^n
Operations	Simple	Complex
Basic Logic element	OR gate	AND gate along with NOT gate.



Que.-7 Explain 4×1 Multiplexer in detail.

- A 4×1 multiplexer consists four data input lines D_0 to D_3 ; two Select Lines as S_1 and S_0 and a Single output Line Y .
- The Select Lines S_1 and S_0 Select one of the four Input Lines to Connect the Output Line.
- The particular input Combination on Select Lines Selects one of input (D_0 through D_3) to the Output.
-

Enable E



→ The truth table of a 4×1 Multiplexed.

Select Delexer input	Output	
S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

1 If $S_1 = 0$ and $S_0 = 0$ then $y = D_0$

Therefore $y = D_0 \cdot (\bar{S}_1) \cdot (\bar{S}_0)$

2 if $S_1 = 0$ and $S_0 = 1$ then $y = D_1$

Therefore $y = D_1 (\bar{S}_1) \cdot (S_0)$

3 if $S_1 = 1$ and $S_0 = 0$ then $y = D_2$

Therefore $y = D_2 S_1 (\bar{S}_0)$

4 if $S_1 = 1$ and $S_0 = 1$ then $y = D_3$

Therefore $y = D_3 \cdot S_1 \cdot S_0$

→ To get the total data output from the Multiplexers all these products terms are to be summed

$$y = D_0 \bar{S}_1 \bar{S}_0 + D_1 \bar{S}_1 S_0 + D_2 S_1 \bar{S}_0 + D_3 S_1 S_0$$

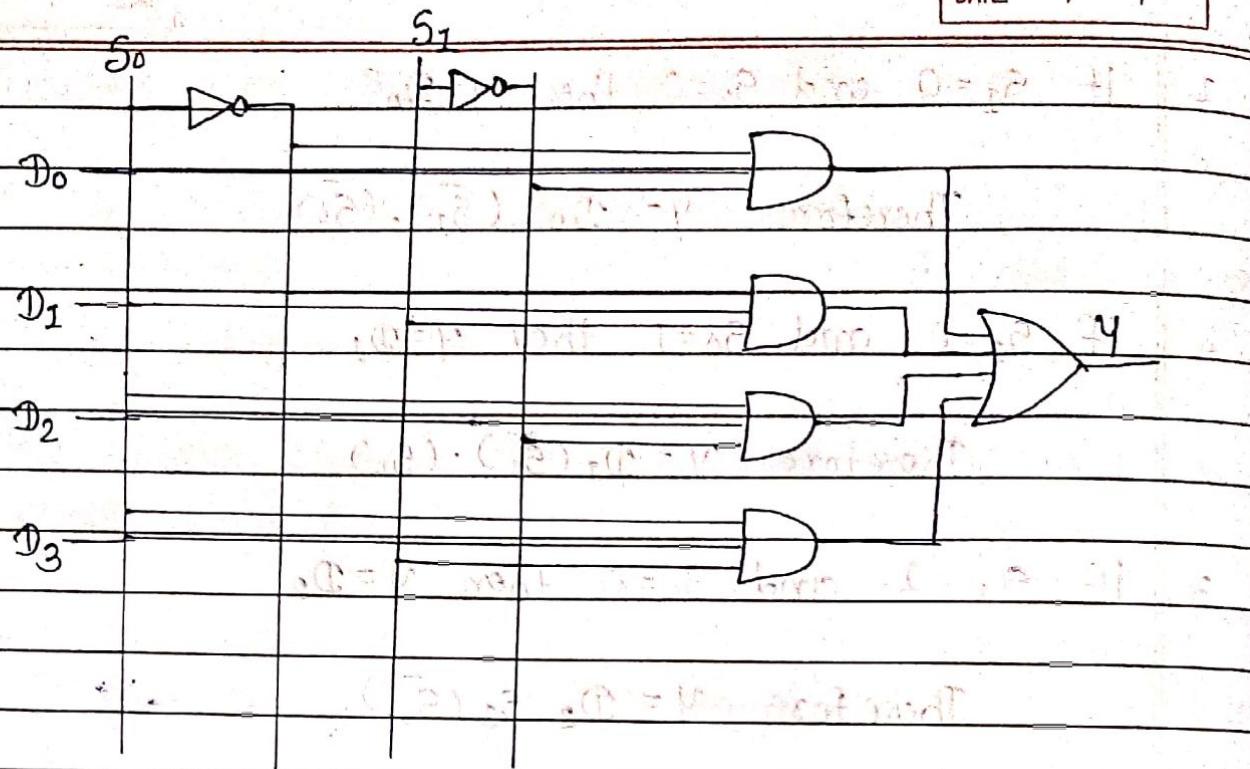
→ Implementations

Required : 3 input AND gates (4)

1 input NOT gates (1)

4 input OR gates (7)

⇒ Each data input line is connected as input to one AND gate and two Select Line are connected as other two inputs to it.



Q4:- Multiplexer.

- Multiplexer processes the digital information from Various Sources into a Single Source.
- It is known as digital Selector.
- Multiplexer is a digital Switch.
- It follows combinational logic type.
- It has common data input.
- It has single data output.
- It works on many to one Operational principle.
- In time division Multiplexing, Multiplexer is used at the transmitter end.

DE Multiplexer.

- Demultiplexer receives digital information from a Single Source and Converts it into Several Sources.
- It is known as Data distributor.
- Demultiplexer is a digital circuit.
- It also follows combinational logic type.
- It has single data input.
- It has n data outputs.
- It works on one to many Operational principle.
- In time division Multiplexing, demultiplexer is used at the receiver end.

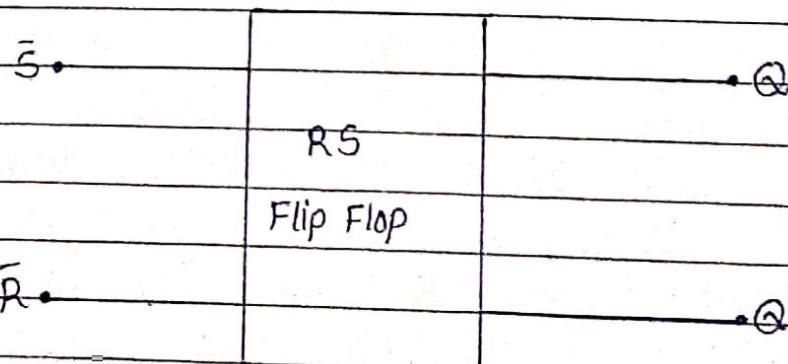
Assignment - 4

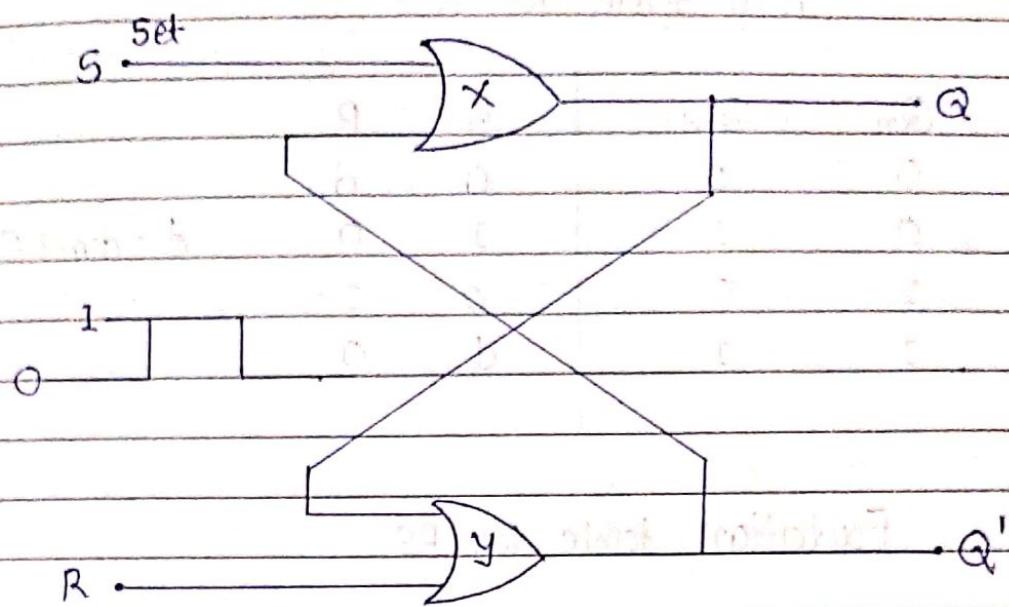
Page No.

DATE: / /

Que-1 Explain RS latch Flip Flop with NOR gates. also write down its truth table and execution table.

- The Simplest type of flip-flop is called an R-S latch.
- It has two Outputs labelled Q and Q' and two input labelled S and R.
- The state of the latch corresponds to the Level of Q (HIGH or Low, 1 or 0) and Q' is the complement of that State.
- It can be constructed using either two Cross-coupled NAND gates or two Cross-Coupled NOR gates.
- Using two NOR gates an active-HIGH S-R latch can be constructed and using two NAND gates an active low S-R latch can be constructed.
- RS Stands for Set/Reset.





RS using NOR gates

S	R	Q	\bar{Q}
0	0	No Range	No Range
0	1	0	1
1	0	1	0
1	1	0	0

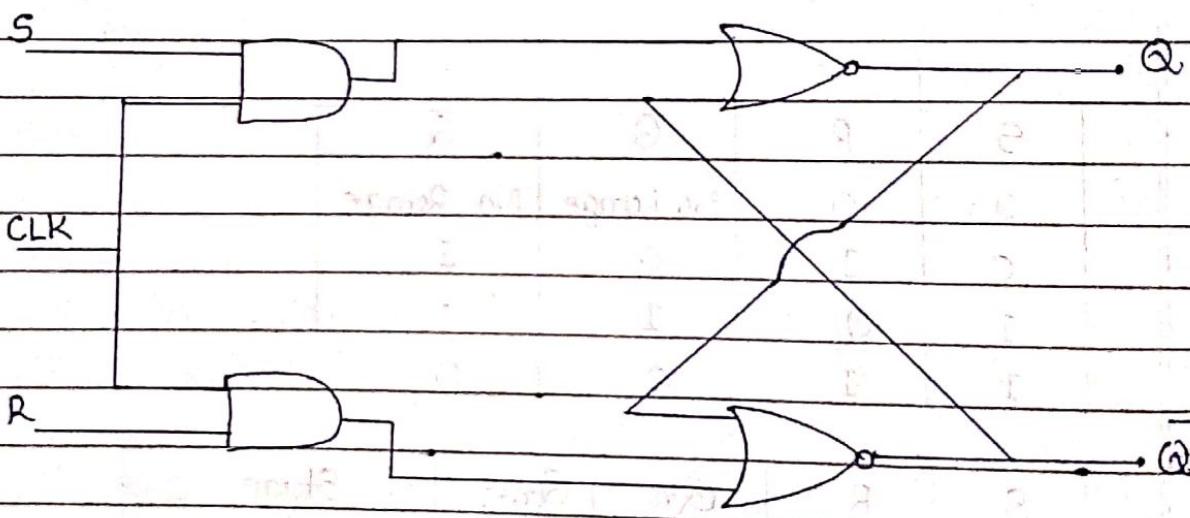
S	R	Q_n	Q_{n+1}	Stage
0	0	0	0	No Change
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	
1	0	0	1	SET
1	0	1	1	
1	1	0	X	Indeterminate (invalid)
1	1	1	X	

Truth table for R-S

Q_n	Q_{n+1}	S	R
0	0	0	d
0	1	1	0
1	0	0	1
1	1	d	0

Excitation table for RS

Que-2 Explain clocked RS Flip Flop With ZAND and 2NOR gates.



→ The above circuit shows the Clocked RS Flip flop with NOR gates.

→ The Operation of the Circuit is same as the RS flip flop with NOR gates when the Clock is high but When the Clock is low the Output State will be "No change state."

1) When the clock is low i.e. '0', the outputs of two input AND gates will be '0' for any input of S & R. In this case the output Q_{n+1} will be '0'.

- The 0-0 input to the RS flip-flop with NOR gates results in "No Change State" at the output.
- This ensures for the clock, zero or low conclusion the output will remain in the Same State.

2) When the clock is high the input AND gates acts as a input follower of the clock.

i.e. When the other input is 0 output will be Zero
or if the input is 1 output will be 1.

→ In this case the RS flip-flop acts according to the input S & R and change its state according to the input i.e. here Q_{n+1} = Q_n

clock	S	R	Q _{n+1}
0	-	-	No Change
1	0	0	No Change
1	0	1	0
1	1	0	1
1	1	1	Invalid

Ques-3 Explain D Flip Flop with NAND gates.

- D flip flop are used as a part of memory storage elements and digital processors as well.
 - The major applications of D flip flop are to introduce delay in timing circuit, as buffer Scrambling data at specific intervals.
 - Whenever the clock signal is low the input is never going to affect the output state.
 - The clock has to be high for the inputs to get active.
 - Thus, D flip flop is controlled by bi-stable latch where the clock signal is the control signal.
 - This gate is divided into positive edge triggered D flip flop and negative edge triggered D flip flop.

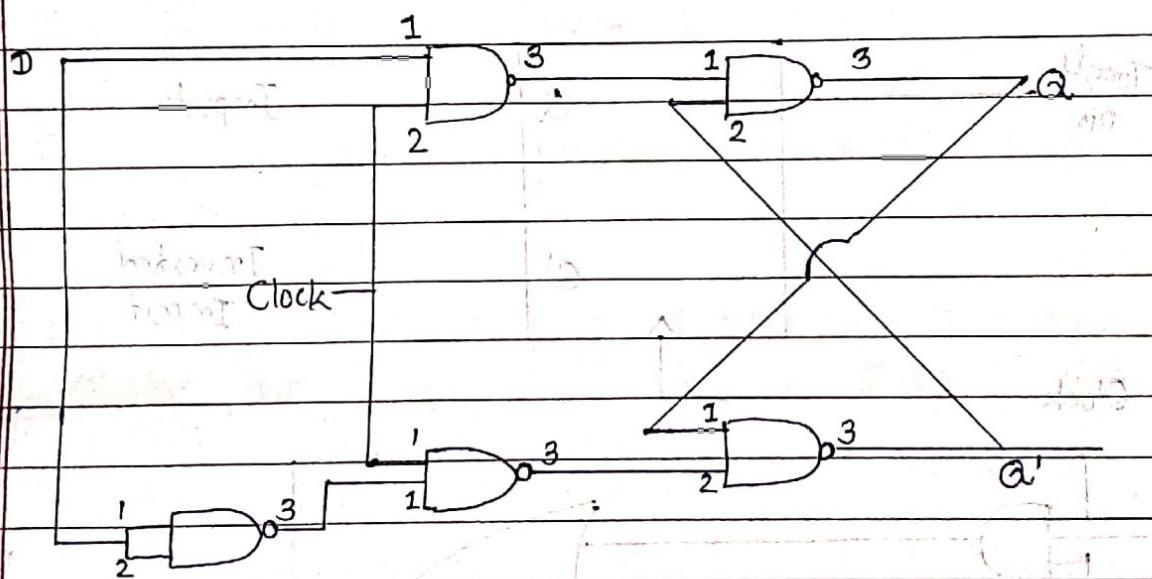
Hand-drawn logic diagram of a D flip-flop:

- Inputs:**
 - DATA PIN (labeled D and Q)
 - Clock
 - CLEAR
- Outputs:**
 - OUTPUT (labeled Q and \bar{Q})

• Truth table of D flip-flop

Write truth table of D flip-flop

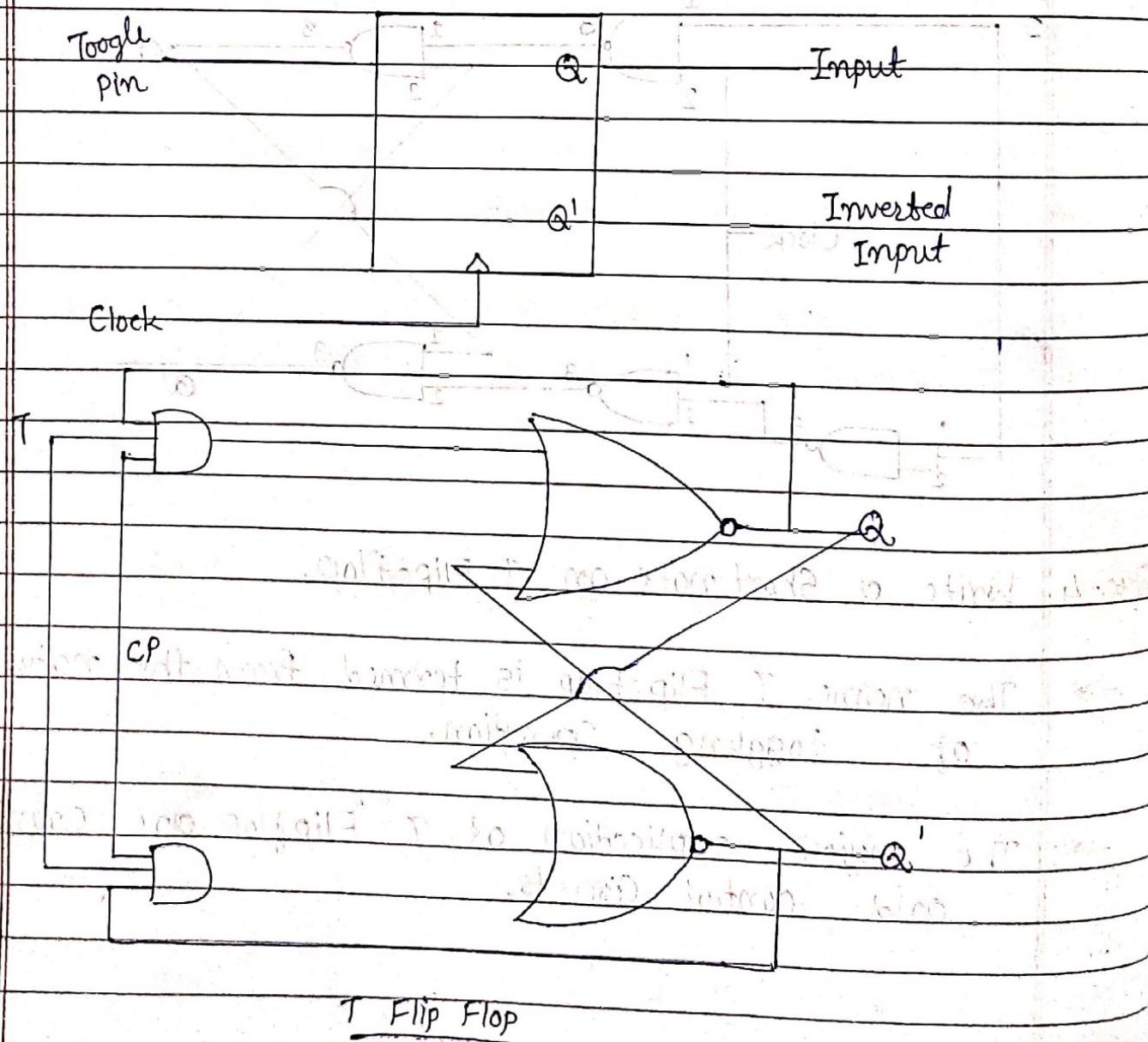
Clock	Input D	Output Q Q'
LOW	X	0 1
HIGH	0	0 1
HIGH	1	1 0



Ques-4 Write a short note on T Flip-flop.

- The name T Flip-flop is derived from the nature of toggling operation.
- The major applications of T flip-flop are counters and control circuits.

- Whenever the Clock Signal is low the input is never going to affect the Output State.
- The Clock has to be high for the inputs to get active.
- Thus, T flipflop is a controlled Bi-Stable latch where the clock signal is the Control Signal.
- Thus, the output has two Stable States based on the inputs.



Clock	INPUT	Output		
	Reset	T	Q	Q'
X	LOW	X	0	1
HIGH	HIGH	0	No Change	
HIGH	HIGH	1	Toggle	
LOW	HIGH	X	No Change	

• Truth Table •

Ques-5 Explain J-K Flip Flop.

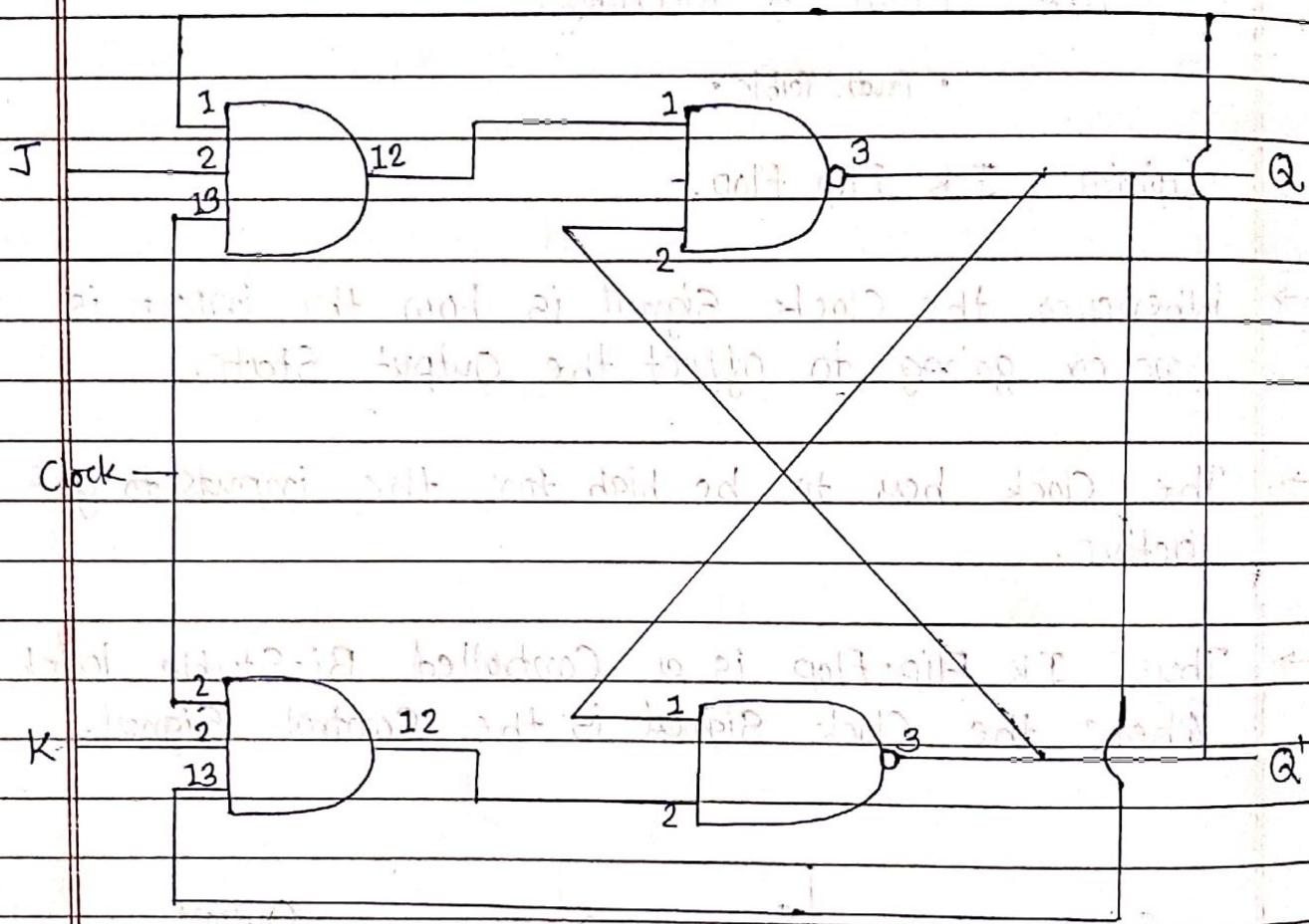
- Whenever the Clock Signal is Low the input is never going to affect the Output State.
- The Clock has to be high for the inputs to get active.
- Thus, JK Flip-Flop is a Controlled Bi-Stable latch where the Clock Signal is the Control Signal.

Set pin	J	Q	Output
Reset Pin	K	Q'	Inverted Output

Clock

- The J and K are the input States for the JK flip flops.

- The Q and Q' represents the Output States of the flipflops.
- The main advantage is that this has toggling function..



Clock	INPUT	OUTPUT
-------	-------	--------

X	Reset	J K	$Q \quad Q'$
---	-------	-----	--------------

High	Low	0 0	No change
------	-----	-----	-----------

High	High	0 1	0 1
------	------	-----	-----

High	High	1 0	1 0
------	------	-----	-----

High	High	1 1	Toggle
------	------	-----	--------

Low	High	x x	No Change
-----	------	-----	-----------

High	High	x x	No Change
------	------	-----	-----------

High	High	x x	No Change
------	------	-----	-----------

Ques-6 Explain Master Slave Flip flop.

[1] RS Flip flop

→ It is built from two gated SR latches.

1. Master
2. Slave

→ Master takes the flip-flop inputs.

S (Set)

R (Reset)

C (clock)

→ The clock input is fed to the latch's gate input.

→ The Slave takes the Master's output as inputs

(Q to S and Q to R) and the complement of the flip-flop's clock input.

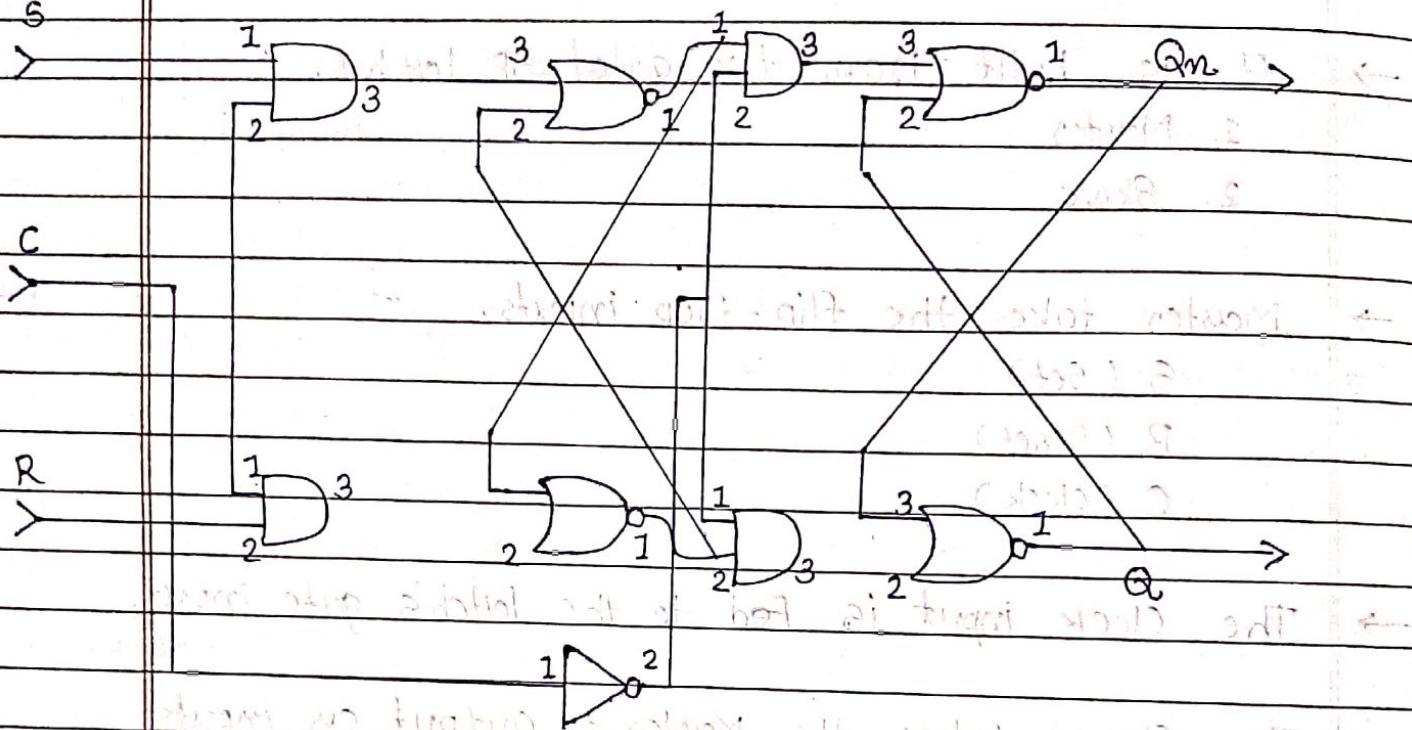
→ The Slave's output are the flip-flop's outputs.

→ This difference in clock inputs between the two latches disconnects them and eliminates the transparency between the flip-flop's inputs and outputs.

→ The Schematic below shows a Master Slave SR flip flop.

→ The two inputs S and R are used to Set and reset the data respectively.

→ The Clock input C is used to control both the master and slave latches making sure only one of the latches can set its data at given time.



→ When C has the value 1, the master latch can set its data and the slave latch can not.

→ When C has the value 0, the slave can set its data and the master cannot.

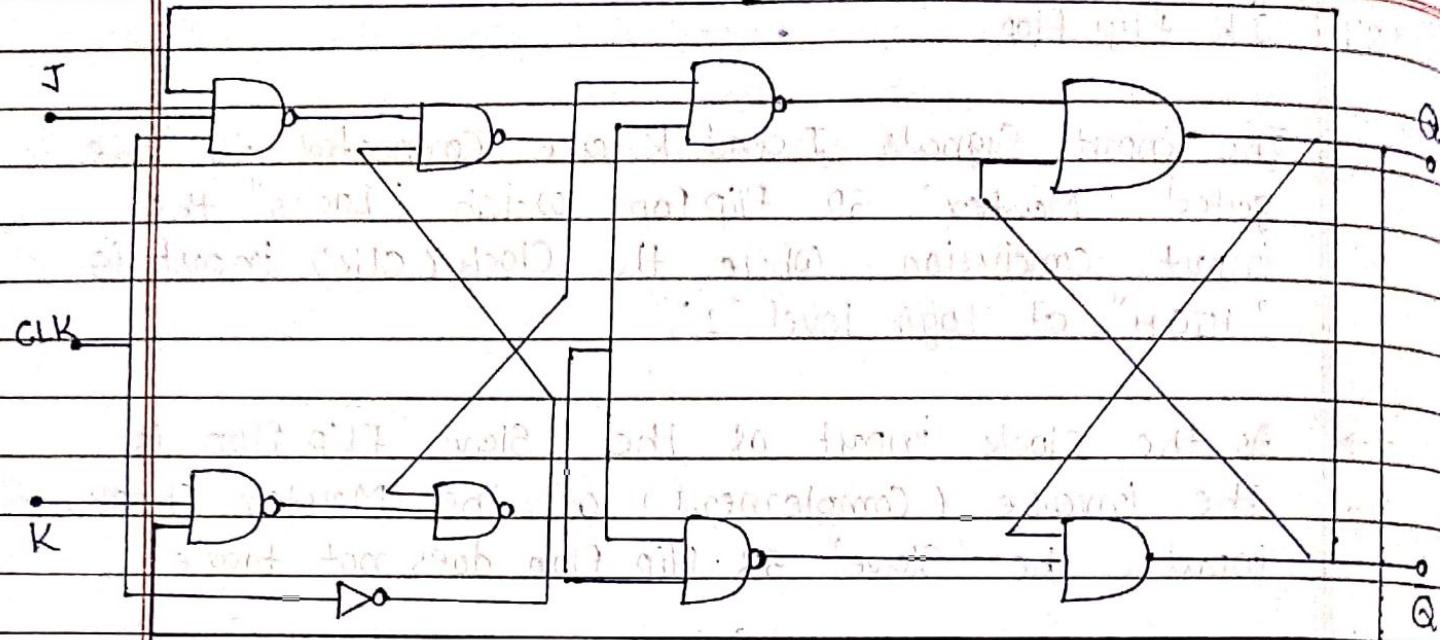
→ The outputs Q and Qm are the flip-flops stored data and their complements of the flip-flops stored data respectively.

Q(t+1) R Q(t+1) Meaning

0	0	Q	Qm	Hold
0	1	?	?	Reset
1	0	?	0	Set
1	1	?	?	undefined

[2] JK flip flop

- The input Signals J and K are connected to the gated "Master" SR flip flop which "Locks" the input conclusion while the Clock (CLK) input is "HIGH" at Logic level "1".
- As the clock input of the "Slave" flip flop is the inverse (Complement) of the Master Clock input, the "Slave" SR flip flop does not toggle.
- The Output from the "Master" flip flop are only "Seen" by the gated "Slave" flip flop When the Clock input goes "Low" to Logic Level "0".
- When the clock is "LOW" the Outputs from the Master flip flop are latched and any additional changes to its inputs are ignored.
- Then on the "Low-to-HIGH" transition of the clock pulse the inputs of the "Master" flip flop are fed through to the gated input of the Slave flip flop.
- On the "High to Low" transition the same inputs are reflected on the output of the Slave, making this type of flip flop a edge or pulse-triggered.
- Then the circuit accepts input data when the clock signal is "HIGH" and passes the clock to the output on the falling edge of the clock signal.



Que-7 Give & Classification of Switching Circuits.

→ Input signal at "Input" and output at "Output".

→ Mainly two types

→ [1] Combinational

→ [2] Sequential

[1]

Combinational

The term combinational Logic-Circuit is used for such circuits.

which gives output to final output directly.

Arithmetic and
Logic Functions

Data
Transmission

Converters

Address Multiplexors, Demultiplexors, Binary

Subtractors with demultiplication, BCD

Comparators

Encoders

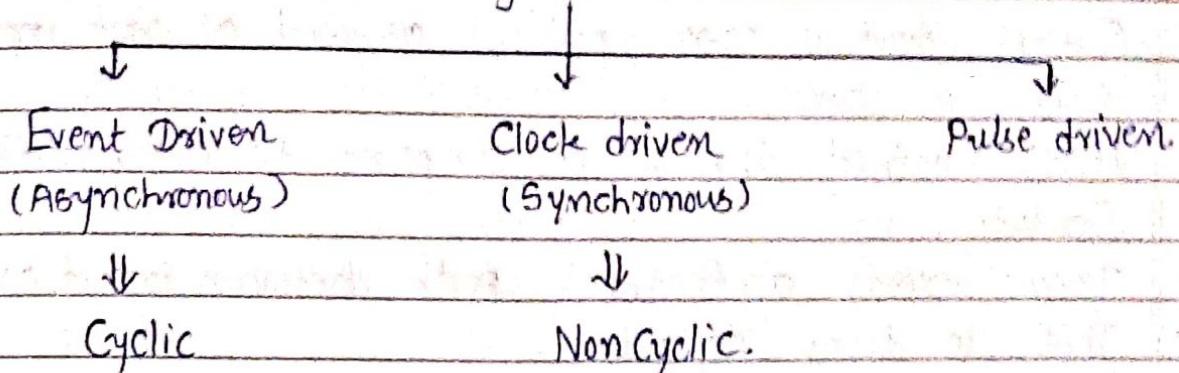
7 Segment

PLD's.

Decoders

[2]

Sequential Logic Circuit.



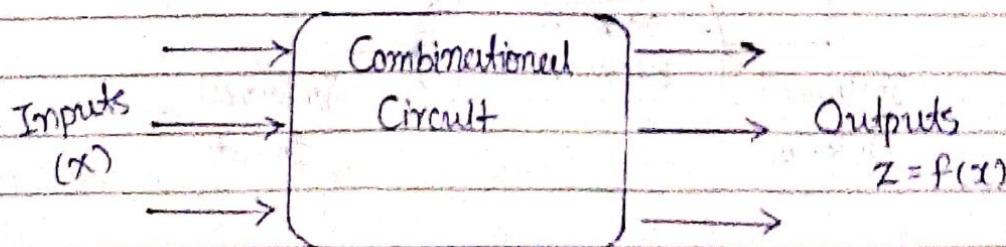
Que - 8

- Combinational Switching Circuit
- Output depends only upon present input
- Speed is fast
- It is designed easy.
- There is no feedback between Input and Output.
- This is time independent
- Elementary building blocks : Logic gates
- Used for arithmetic as well as boolean expression.
- Don't have capability to store any state.
- Don't have clock, they don't require trigger.
- don't have any memory element.
- Easy to use and handle.

Example:

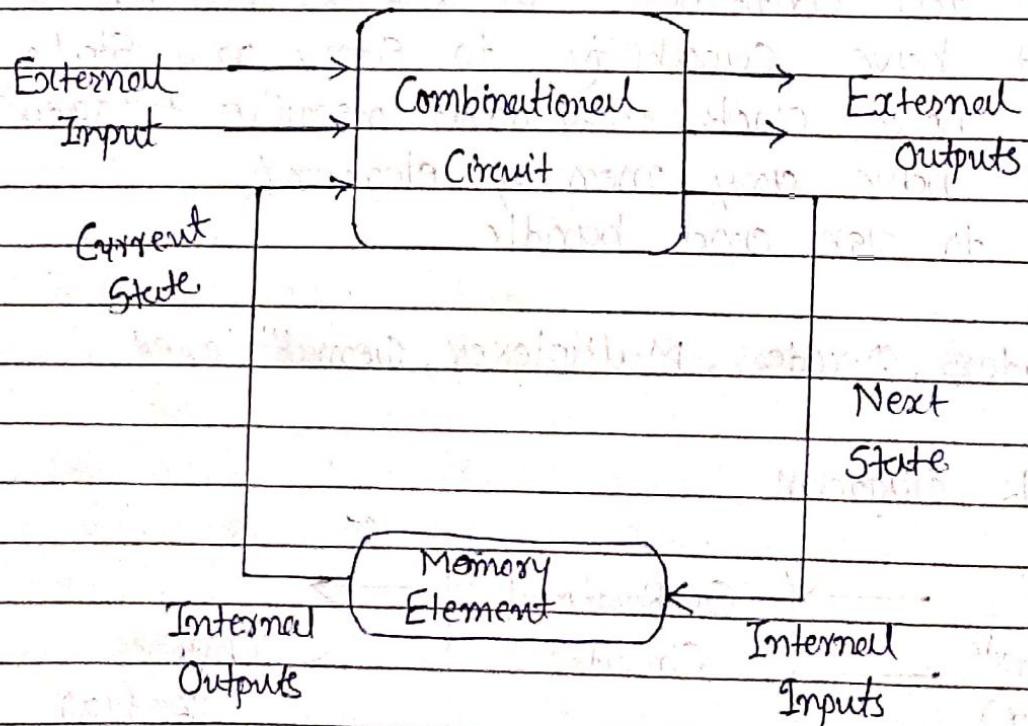
Encoders, Decoders, Multiplexer, Demultiplexer

Block diagram



• Sequential Switching Circuit

- Output depends upon present as well as past input
- Speed is slow.
- It is designed through as compared to combination circuit.
- There exists a feedback path between input and o/p
- This is time dependent.
- Elementary building blocks : Flip - Flops
- Mainly used for storing data,
- have capability to store any state or to retain earlier state.
- Clock dependent to and they need triggering
- have memory element
- It is not easy to use and handle.
- flipflops, Counter



Synchronous Sequential Circuits

- The feedback to the input for next Output generation is governed by Clock Signals.
- The memory unit which is being get used for governance is clocked flip flop.
- The States of Synchronous Sequential Circuits are always predictable and this reliable.
- It is easy to design Synchronous Sequential Circuit.
- Due to the propagation delay of Clock Signal in switching cell elements of the Circuit there is Slower in its Operation Speed.
- Used in counters, Shift registers, Memory Units.

Asynchronous Sequential Circuits

- The feedback to the input for next Output generation is not governed by Clock Signals.
- Unclocked flip flop or time delay is used as memory element in case of this circuits.
- There are chances of for the circuits to enter into wrong state because of the time difference between the arrivals of inputs.

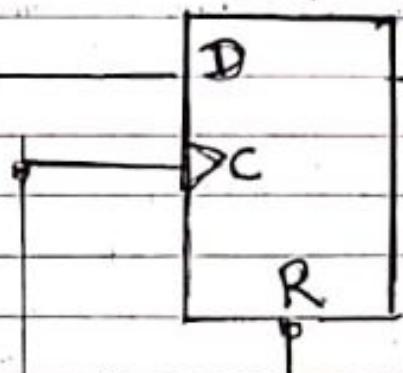
- The presence of feedback among logic gates causes instability issues making difficult design.
- There is no clock signal delay these are fast compared to the Synchronous Sequential Circuits.
- Used in low power and high speed operations such as Simple Microprocessors, digital Signal processing.

Que-9

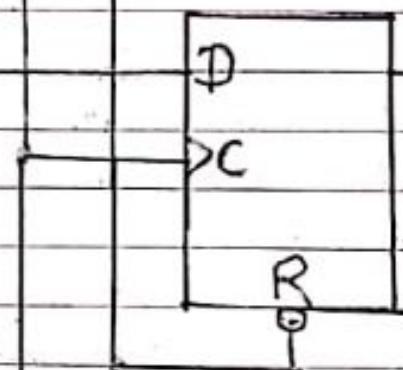
Explain 4-bit register.

- A register consists of a group of flip-flops and gates that effect their transition.
- The flip flop hold the binary information.
- The gates determine how the information is transferred into the registers.
- figure shows a register constructed with four D type flip flops.
- "clock" triggers all flip flops on the positive edge of each pulse.
- "clear" is useful for clearing the register to all 0's prior to its clocked operation.
- An n-bit register consists of a group of n flip flops capable of storing n bits of binary information.

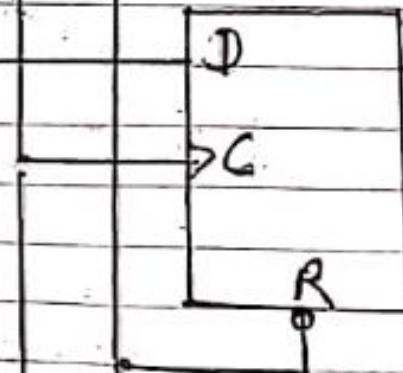
I_0 ————— | D | A_0



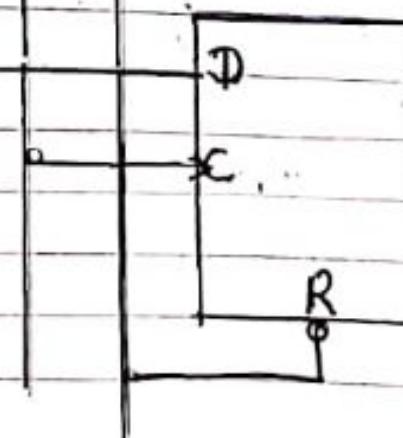
I_1 ————— | D | A_1



I_2 ————— | D | A_2



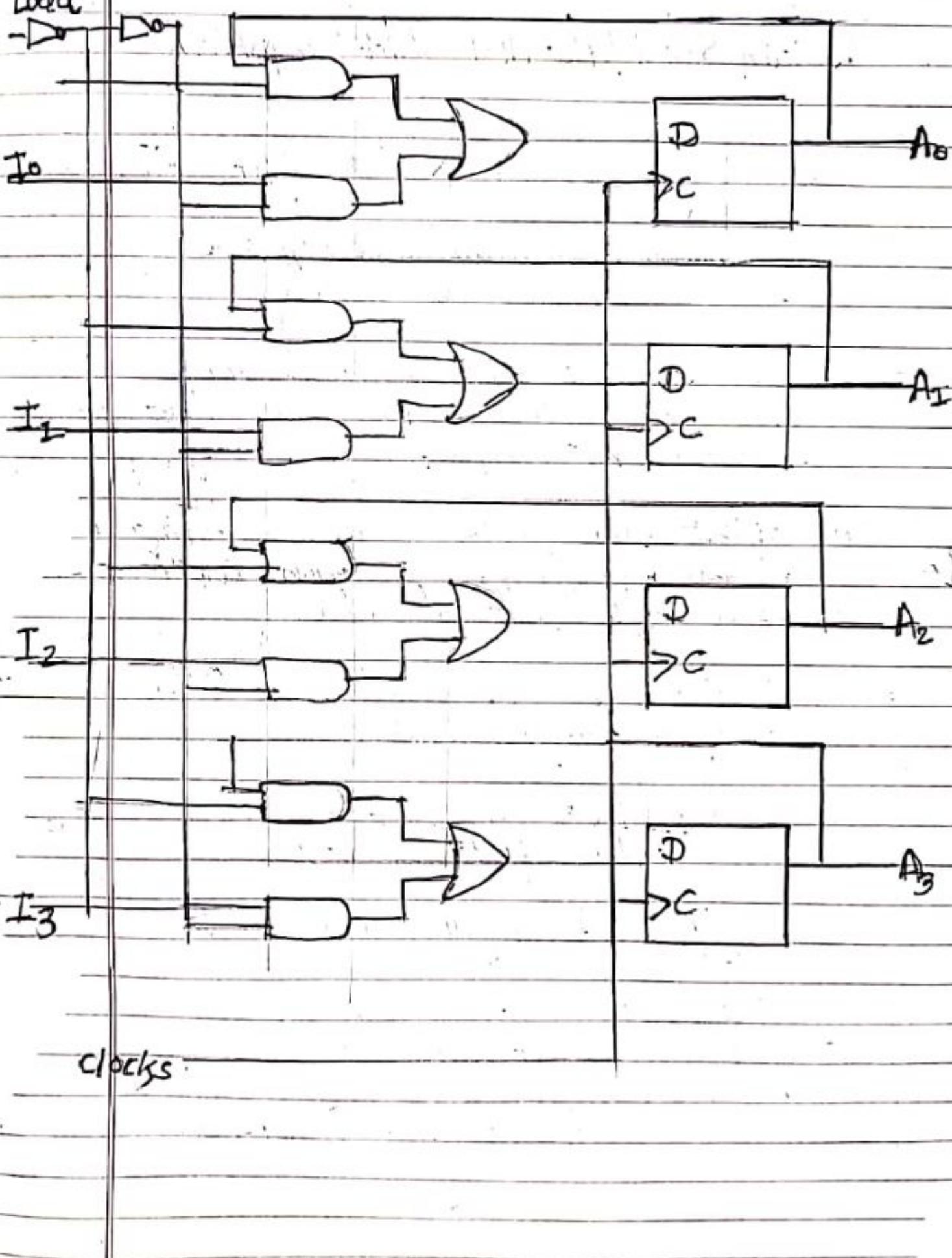
I_3 ————— | D | A_3



Ques-10 Explain 4-bit registers with Parallel Load.

- A Clock edge applied to the C inputs of the registers will load all four inputs in parallel.
- For Synchronism, it is advisable to control the operation of the register with the D inputs rather than controlling the clock in the C inputs of the flip flops.
- A 4-bit register with a Load Control input that is directed through gated and into the D inputs of the flip flops.
- When the Load input is 1, the data in the four inputs are transferred into the register with next positive edge of the clock.
- When the Load input is 0 the outputs of the flip flops are connected to their respective inputs.
- The feedback connection from output to input is necessary because the D flip flops does not have "no change" condition.

Load



clocks

Que.-12 Explain Serial / Parallel Shift registers.

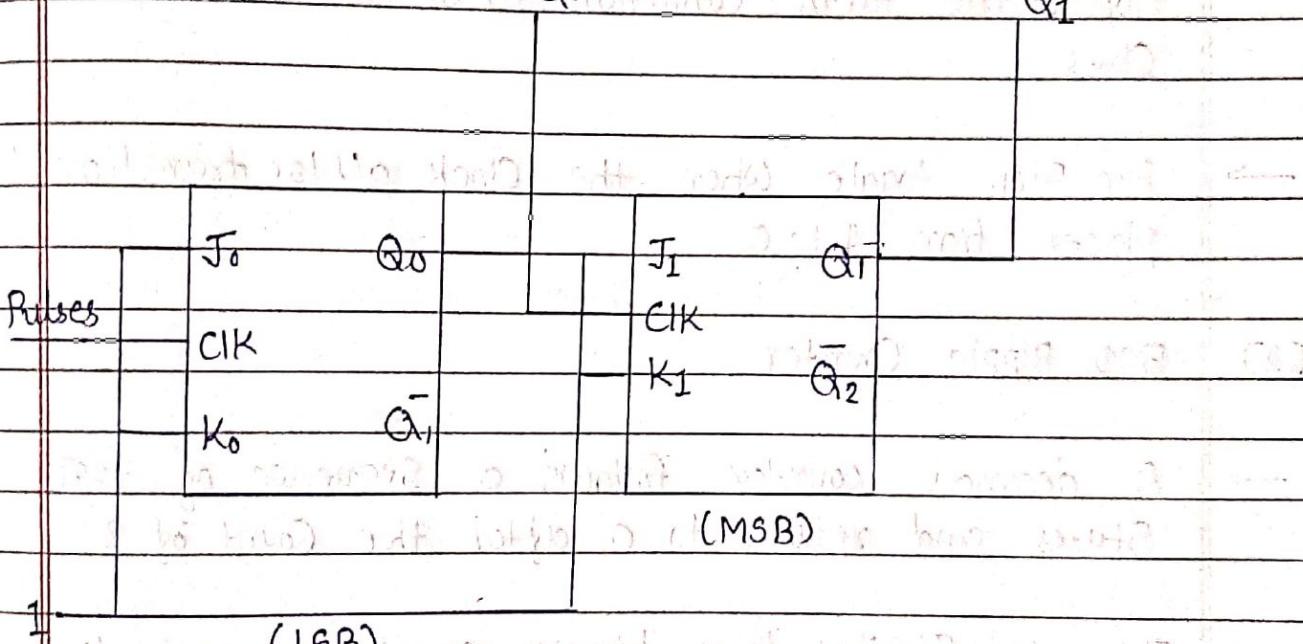
- The information stored within registers can be transferred with the help of Shift registers.
- Shift registered is a group of flip flops used to store multiple bits of data.
- The bits stored in such registered can be made to move within the registers and in/out of the registered by coupling Clock pulses.
- An n-bit Shift registers can be formed by connecting n flip flops where each flip flop stores a single bit of data.
- The registers which will shift the bits to right are called "Shift right registers."
- Shift registers are basically of 4 types.
 1. Serial in Serial out Shift registers.
 2. Serial in Parallel out Shift registers.
 3. Parallel in Serial out Shift registers.
 4. Parallel in parallel out Shift registers.

Ques-13 Write a short note on.

[1] Binary Ripple Counter

→ A Binary Counter is a 2-Mod Counter which Counts up to 2 bit State values, i.e. $2^2 = 4$ values.

→ The flipflop having similar conditions for toggling like T and JK are used to construct the Ripple Counter.



→ In the circuit design of the binary ripple Counter, two JK flip flops are used.

→ The high voltage signal is passed to the inputs of both flip flops.

→ This high voltage input maintains the flip flops at a State 1.

→ The outputs Q_0 and Q_1 are the LSB and MSB bits.

J_n	K_n	Q_{n+1}
0	0	Q_m
1	0	1
0	1	0
1	1	Q_m

→ When the high voltage to the inputs of the flip flops, the forth condition is of the JK flip flop Clues.

→ Flip Flop toggle when the Clock pulse transition takes places from 1 to 0

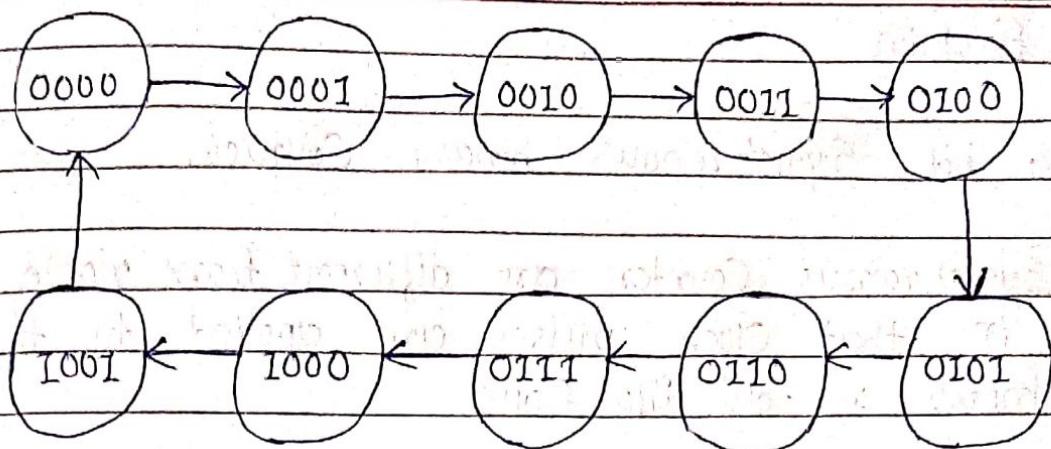
[2] BCD Ripple Counter

→ A decimal counter follows a sequence of ten states and returns to 0 after the count of 9.

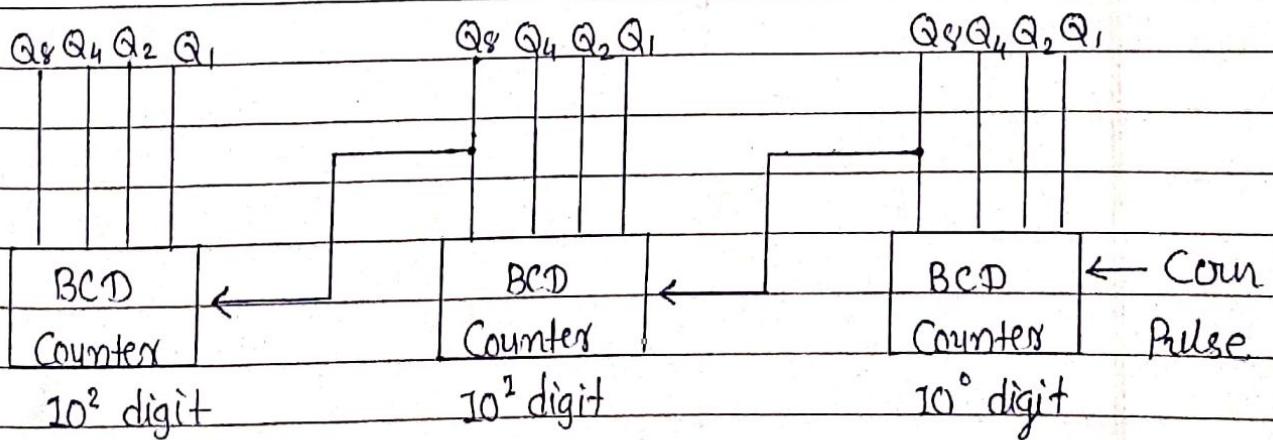
→ This is similar to a binary counter except that the state after 1001 is 0000.

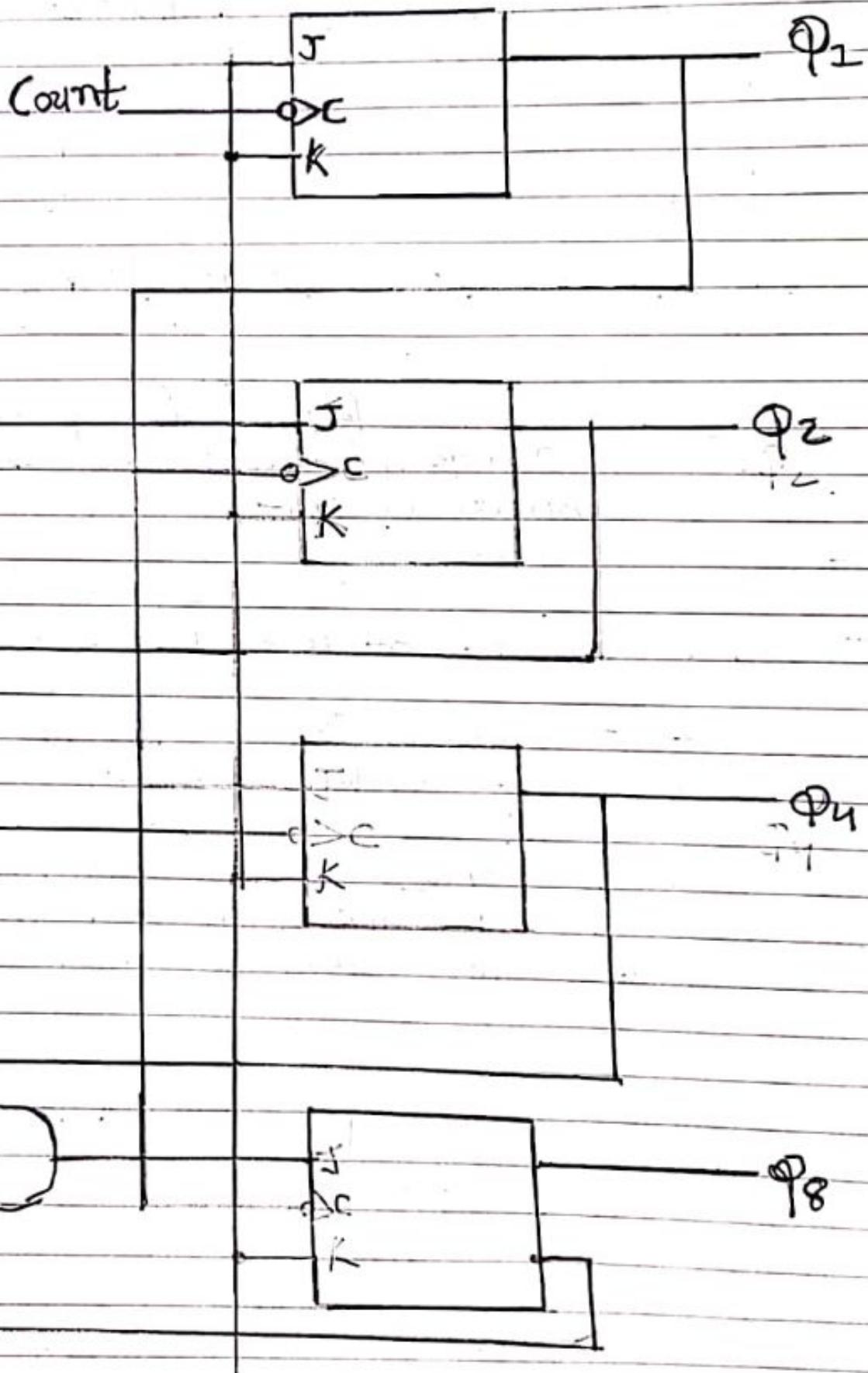
→ The operation of the Counter can be explained by a list of conditions for flip-flop transitions

→ The four outputs are designated by the letter symbol Q with a numeric subscript equal to the binary weight of the corresponding bit in the BCD code.



- The BCD Counter of is a decade Counter.
- To Count in decimal from 0 to 999. we need a three - decade Counter.
- The multiple decade Counters can be Constructed by Connectivity BCD Counters in Cascade, or for each decade.



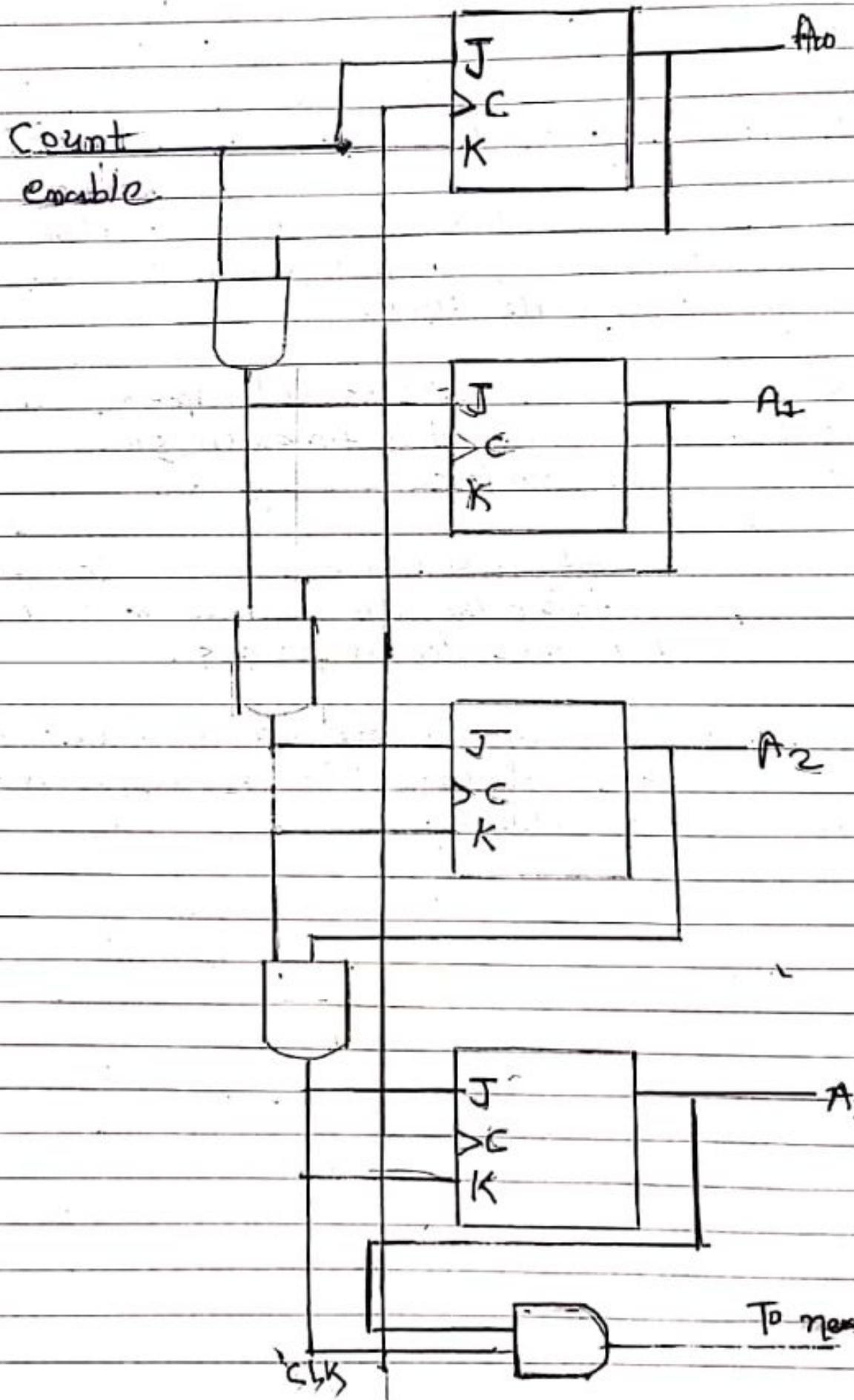


Que.-14

Explain.

1. 4 bit Synchronous binary Counter.

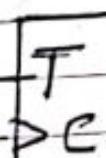
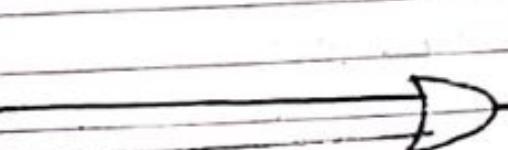
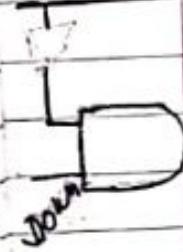
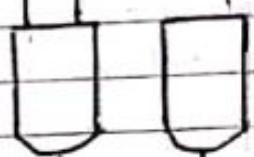
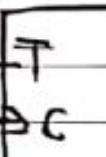
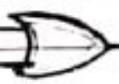
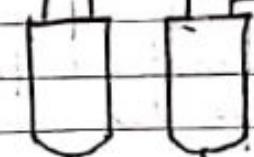
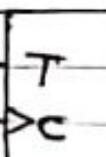
- Synchronous Counters are different from ripple Counter in that Clock pulses are applied to the inputs of all flip-flops
- A Common clock triggers all flip flops Simultaneously rather than one at a time in Succession as in a ripple Counter.
- The design of a Synchronous binary Counter is so simple that there is no need to go through a Sequential logic design process.
- Synchronous binary Counters have a regular pattern and can be constructed with Complementary flip-flop and gates.



2. 4 bit Up-Down Binary Counter

- These Counters Capable of Counting in either direction through any given Count Sequence.
- They can be reversed at any point within their Count Sequence by using an additional control input.
- Generally most bidirectional Counter Chips can be made to Change their Count direction either up or down at any point within their Count Sequence.
- This is achieved by using an additional input pin which determines the direction of the Count. either up or down.

Up

A₀A₁A₂A₃

clk

Assignment -5

Page No.

DATE: / /

Que-1 What do you mean by the term "State diagram"? What do the Vertices, the directed arcs; and the labels on the arcs of a State diagram represent?

→ A State diagram represents a Finite State machine (FSM).

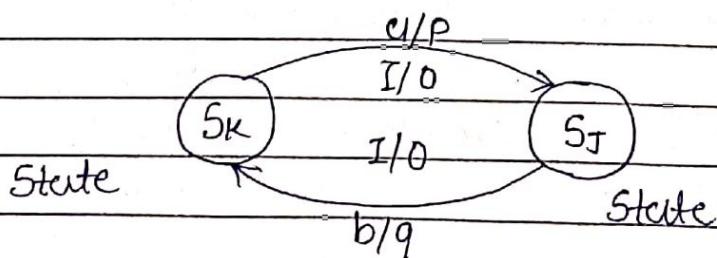
→ It contains Circles and Directed arcs.

[1] Circles : It represents the Directed arcs

→ Labelled with a binary encoded number (eg. s_0) or s_k reflecting State.

[2] Directed arcs : It represents the transitions Between States.

→ Labelled with a input/output for their State transition.



I (Input) : $x(t) \in \{a, b\}$

O (Output) : $y(t) \in \{p, q\}$

State : $S(t) \in \{S_k, S_j\}$

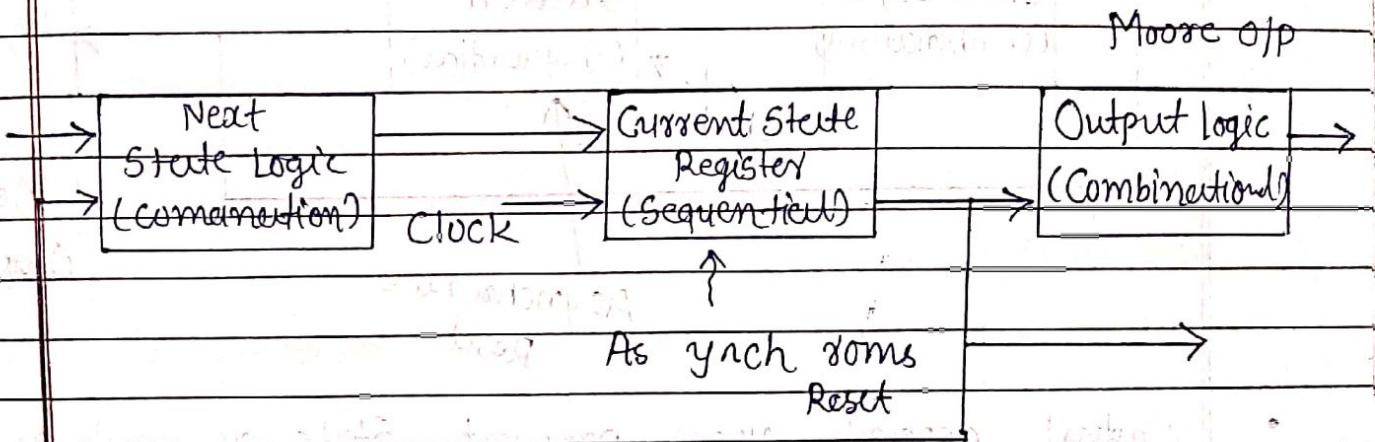
Initial State : $S(0) \in S_k$

Ques-2

What do you mean by the Moore model and the Mealy model of the State diagram?

[1] Moore Model

- A moore Model is defined as a Model in theory of Computation Whose Output Values are determined by only its Current State.

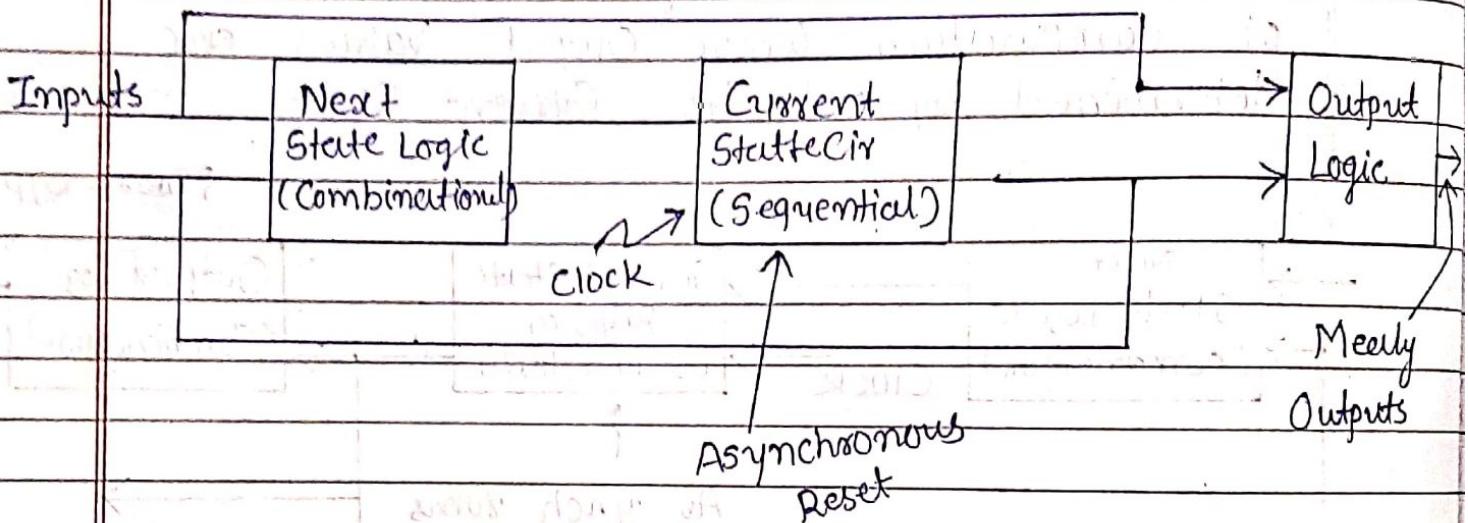


- Output depends only present State.
- If input changes, Output does Change.
- More numbers of States are required.
- There is Less hardware requirement for circuit implementation.
- They react slower to inputs (one clock cycle later)
- Synchronous output and state generation.
- Easy to design..

[2]

Mealy Machine

- A Mealy Machine is defined as a Machine in theory of Computation whose Output Values are determined by both its Current State and Current inputs.



- Output depends upon present State as well as present input.
- If input Changes, Output also Changes.
- Less Number of States are required.
- There is More hardware requirement for Circuit implementation.
- They react faster to inputs.
- Asynchronous Output generation
- It is difficult to design.

Que-3

What do you mean by the term 'State table'? What does each row, column and entry of the State table represent?

→ The State table is representation of a Sequential Circuit Consists of three Sections labeled present state, next state and output.

- Present State

→ The present state of the System, typically given in binary encoded form or with S_k .

→ So, a State of S_5 in our state diagram with 10 states would be represented as 0101 Since we require 4 bits.

- Each Row

→ Each Row Represent number of cases according to the number of States

if two States then $2^2 = 4$ Rows

if three States $2^3 = 8$

So in general 2^n rows

Where n is number of States.

- Input Column

→ Whatever external inputs used to cause the State transitions.

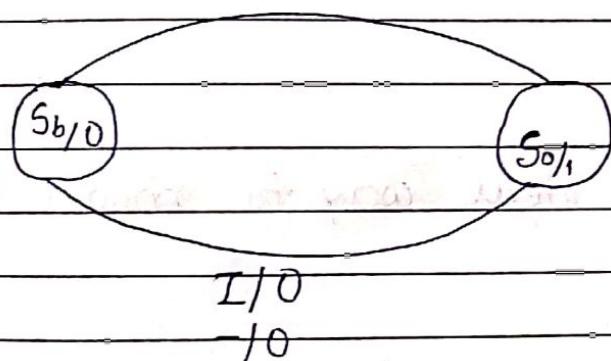
- Next Column (Next State)

→ The next state, generally in binary encoded form.

- Outputs

→ Whatever outputs, other than the state, for the system. Note that there would be no outputs in a Moore Machine

→ Example



→ State table

Present State	Input	Next State	O/P
S0	-	1	-
S1	-	0	1

Ques-4 What do you mean by the term 'excitation table'? What information does it give?

→ Excitation table shows the minimum input that are necessary to generate a particular next state.

- It defines the flip flop input Variable as function of the Current State and Next State.
- Excitation table is used for design of flip flops and counters.
- It also used to implement a) finite State Machine.

Ques-5

What are the advantages of (a) the State diagram and (b) the State table? Compare them.

- State diagram

- State diagram are ideal way to model finite State Machine.
- State diagram enable you to describe the behaviour of State Machine during their entire process.
- State diagrams make the System behaviour visible.

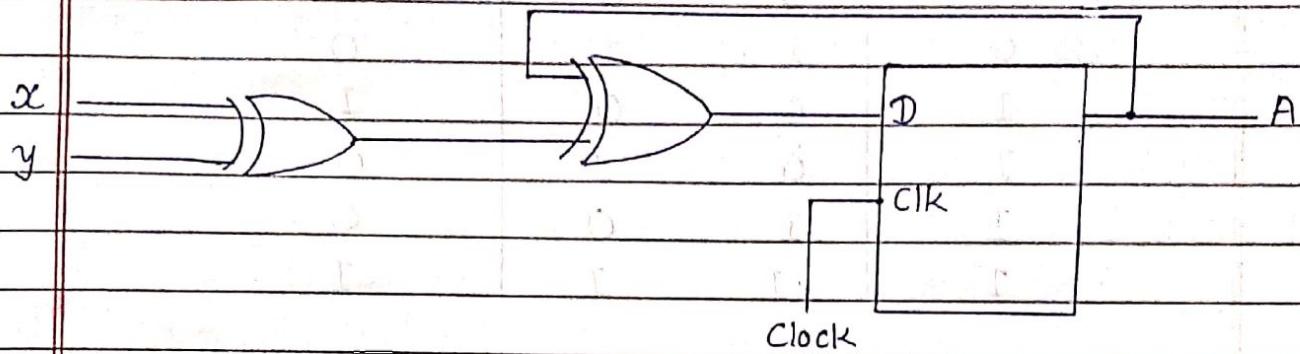
- State table

- State table is representation of different State in form of table.
- The main Advantage is that we can have all Possible cases in form of rows and we can easily Organize them.

Que-6 Draw the state diagram and State table for a

[1] D Flip flop

Step 1: Find State equation.



→ Input equation $D_A = A \oplus x \oplus y$

→ D_A Symbol implies a D flip-flop with Output A.

→ The x, y are the inputs to the Circuits.

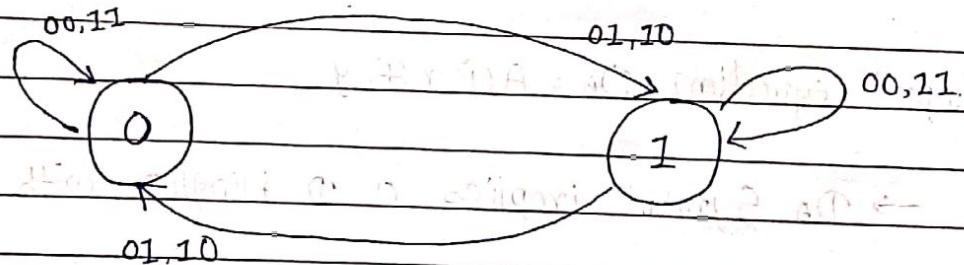
→ No Outputs equations are given, which implies that the Output Comes from the Output of the Flip flop.

So State equation.

$$A(t+1) = A \oplus x \oplus y$$

State Table

Present State	Inputs		Next State
A	x	y	A
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



State diagram of D Flipflop

[2] JK flip flop

Input equations

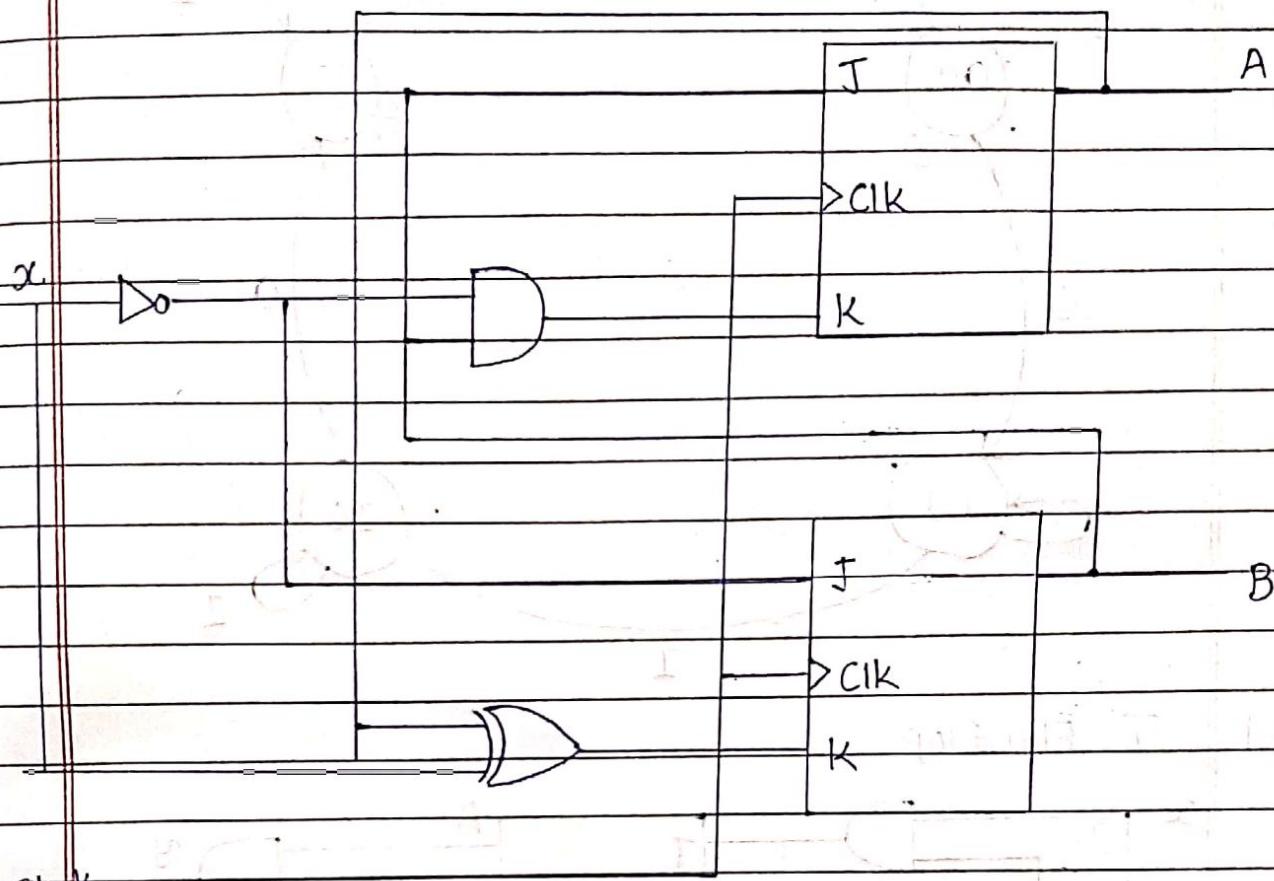
$$J_A = B \quad K_A = Bx'$$

$$J_B = x' \quad K_B = A'b + Ax' = A \oplus x$$

State equations

$$\begin{aligned} A \Rightarrow A^{(+)1} &= BA' + (Bx')'A \\ &= A'B + AB' + Ax \end{aligned}$$

$$\rightarrow \text{for } B \Rightarrow B(t+1) = x'B' + (A \oplus x)'B \\ = B'x' + ABx + A'B'x'$$

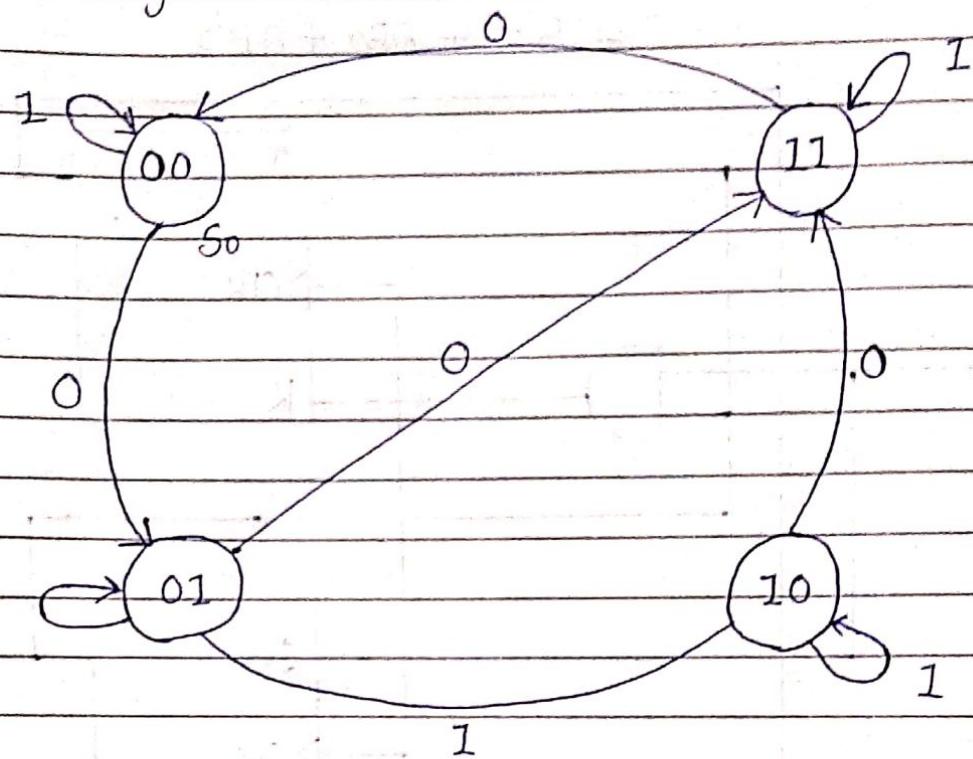


Clock

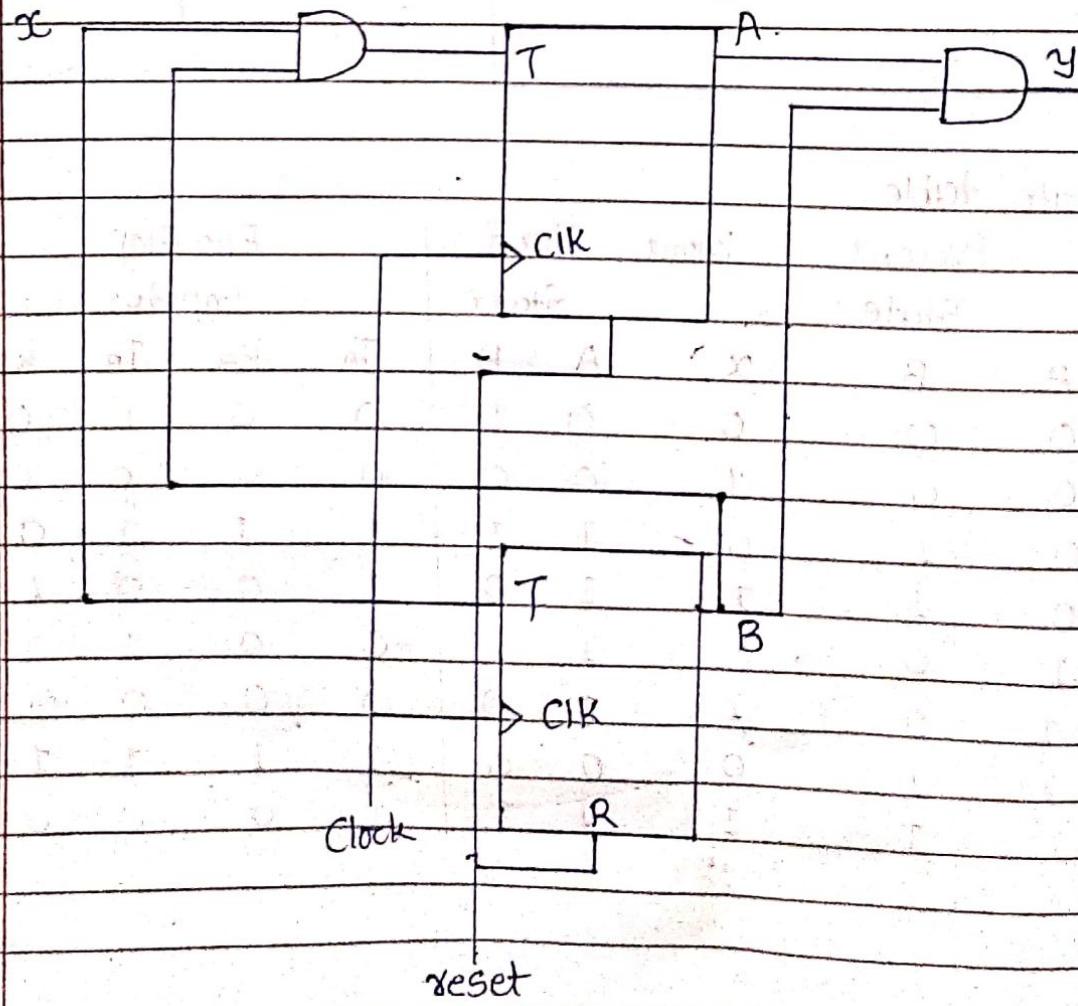
• State table.

Present State			input		Next State		Flip Flop Inputs			
A	B	x	A	B	J _A	K _A	J _B	K _B		
0	0	0	0	1	0	0	1	0		
0	0	1	0	0	0	0	0	1		
0	1	0	1	1	1	1	1	0		
0	1	1	1	0	1	0	0	1		
1	0	0	1	1	0	0	1	1		
1	0	1	1	0	0	0	0	0		
1	1	0	0	0	1	1	1	1		
1	1	1	1	1	1	0	0	0		

State diagram:



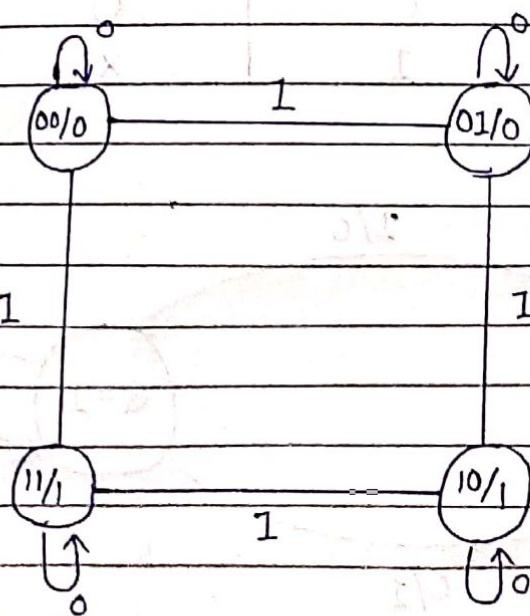
[3] T Flip flop



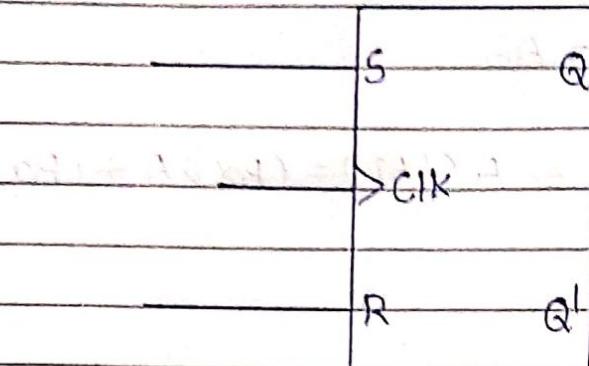
- Input equation : $\frac{T_A}{T_B} = Bx$
- Output equation : $y = AB$
- Characteristic eq : $A \Rightarrow A(t+1) = (Bx)'A + (Bx)A'$

→ State table

Present State	input		Next State	Output	
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	1



[4] S-R Flip flop

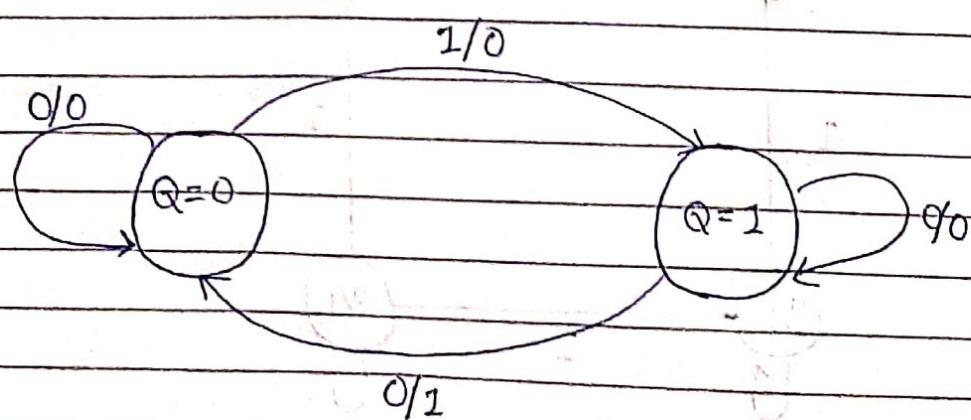


→ Characteristics equation

$$Q_{\text{next}} = S + R'Q$$

→ State table

Q	Q_{next}	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0



[5] 3-bit Gray Code Counter

1. State equations

$$J_0 = Q_2 Q_1 + \bar{Q}_2 Q_1 = \bar{Q}_2 + Q_1$$

$$K_0 = Q_2 \bar{Q}_1 - \bar{Q}_2 Q_1 = Q_2 + Q_1$$

$$J_1 = \bar{Q}_2 Q_0$$

$$K_1 = Q_2 Q_0$$

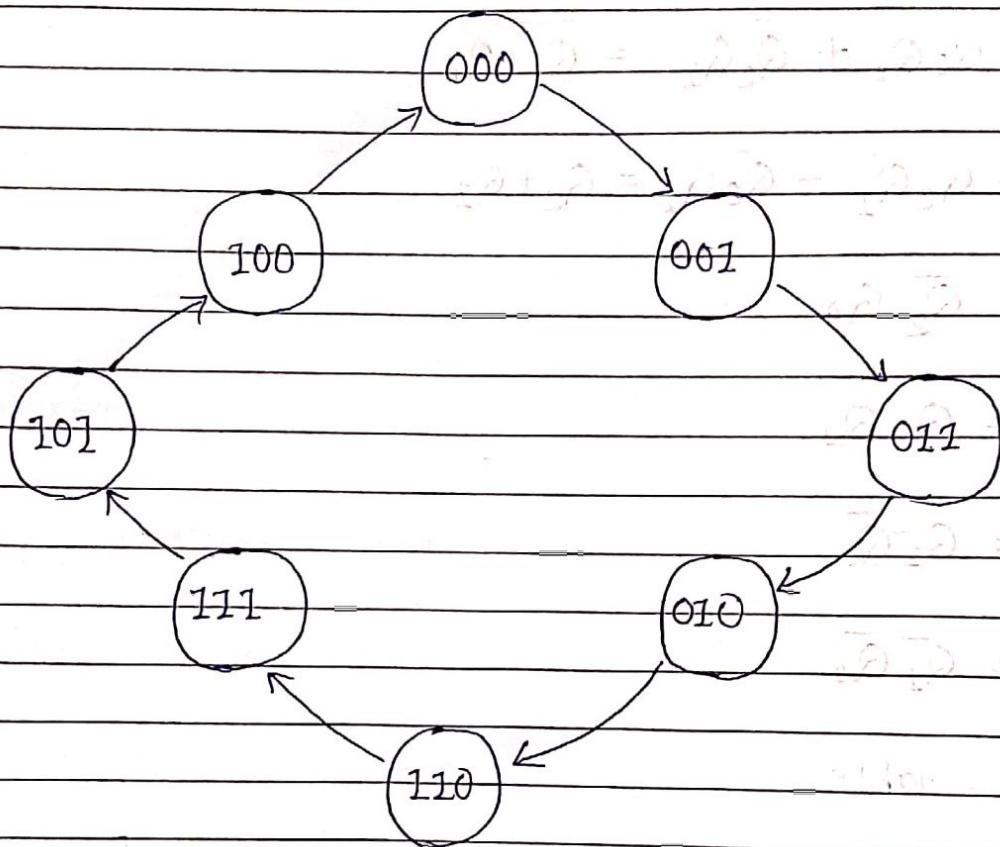
$$J_2 = Q_1 \bar{Q}_0$$

$$K_2 = \bar{Q}_1 \bar{Q}_0$$

2. State table

Present State			Next State			Output	
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	Q_N	Q_{N+1}
0	0	0	0	0	1	0	0
0	0	1	0	1	1	0	1
0	1	1	0	1	0	1	0
0	1	0	1	1	0	1	1
1	1	0	1	1	1		
1	1	1	1	0	1		
1	0	1	1	0	1		
1	0	1	1	0	0		
1	0	0	0	0	0		

3. State diagram



Assignment-6

Page No.

DATE: / /

Que.-1 Explain the types of Read-Only Memory (ROMs).

→ ROM Stands for Read Only Memory.

→ The Memory from which we can only read but cannot write on it.

• Types of ROMs:

[1] MROM (Masked ROM)

- The very first ROMs were hard-wired devices that contained a pre-programmed set of data as instructions.
- This kind of ROMs are known as Masked ROMs (which are in erasable).

[2] PROM (Programmable Read Only Memory)

- PROM is read only memory that can be modified only once by a user.
- The user buys a blank PROM and enters the desired contents using a PROM programmer.
- Inside the PROM chip, there are small fuses which are burnt open during programming.
- It can be programmed only once and is erasable.

[3] EPROM (Erasable and Programmable Read Only Memory)

- EPROM Can be erased by exposing it to ultra-violet light for a duration not up to 40 Minutes.
- Usually an EPROM eraser achieves this function.
- During Programming, an electrical Charge is trapped in an insulated gate region.
- The Charge is retained by for more than 10 years because the Charge has no leakage path.
- For erasing this Charge, ultra violet light is passed through a quartz Crystal Window.
- This exposure to ultra-violet dissipates the Charge.
- During normal use, the quartz lid is sealed with a sticker.

[4] EEPROM (Electrically Erasable and Program Read Only Memory)

- EEPROM is programmable and erased electrically.
- It can erased and reprogrammed about ten thousand times.

- Both erasing and Programming take about 4 to 10 Ms.
- In EEPROM any location can be Selectively erased and programmed.

Ques-2 Explain the Programmable Logic devices.

- Programmable Logic devices (PLDs) are the integrated Circuits.
- They Contain an array of AND gates & another arrays of OR gates.
- There are three Kinds of PLDs based on the type of arrays which has programmable feature.

1. Programmable Read Only Memory.

2. Programmable Array Logic.

3. Programmable Logic Array.

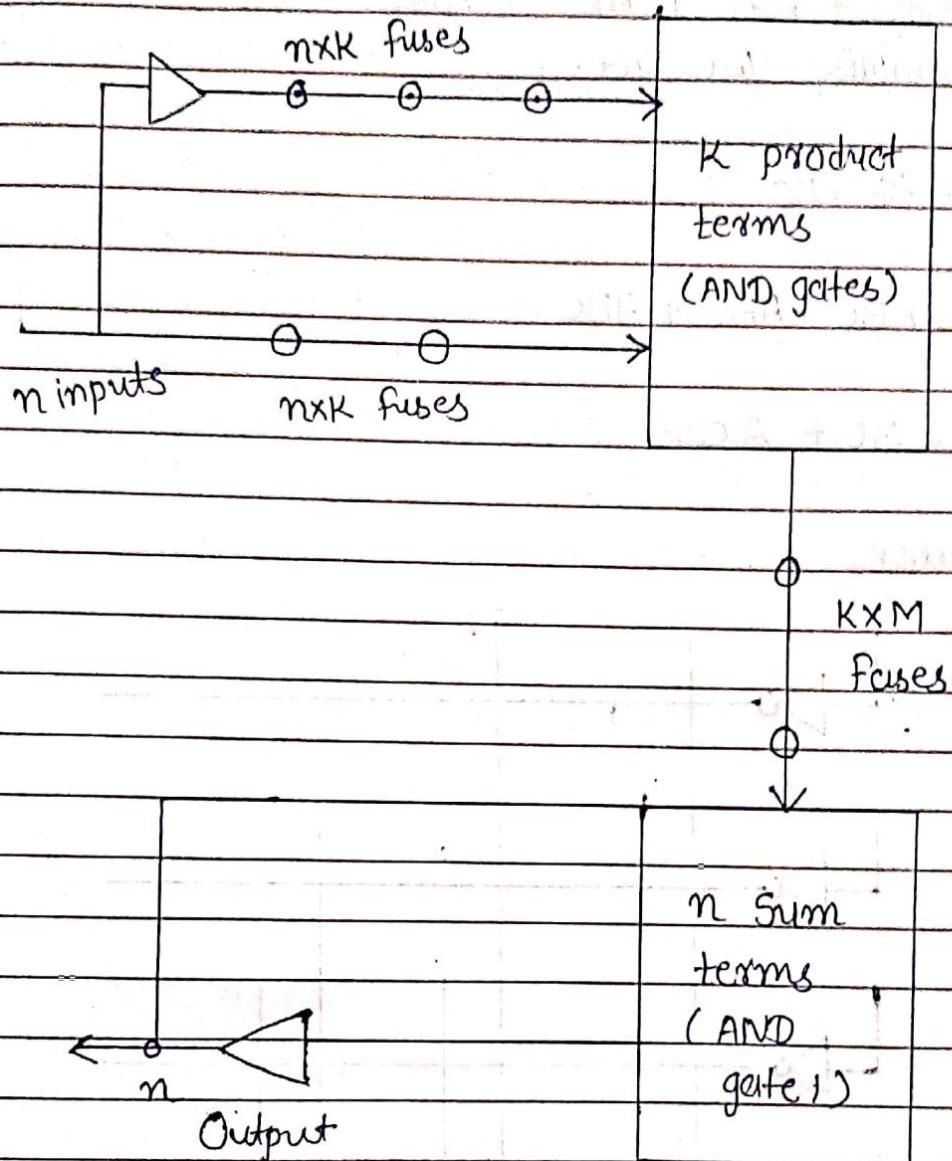
- The process of entering the information into these devices known as programming.
- Basically users can program these devices or ICs electrically in order to implement the Boolean functions based on the requirement.
- Here the term programming refers to hardware Programming but not Software programming.

Que-3

Short note On Programmable Logic arrays

- Programmable logic array is a fixed architecture logic device with Programmable AND gates followed by Programmable OR gates.
- PLA is basically a type of Programmable logic device used to build reconfigurable digital circuit.
- PLDs have undefined function at the time of Manufacturing but they are programmed before made into use.
- PLA is Combination of Memory and Logic.
- PLA is similar to ROM in concept however it does not provide full decoding of Variables and does not generate all minterm as in the Row.
- Through its name Consist of Non-Programmable it does not require any type of Programming like C and C++.

→ Basic Block diagram of PLA.



- Truth table.

A	B	C	F_1	F_2
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

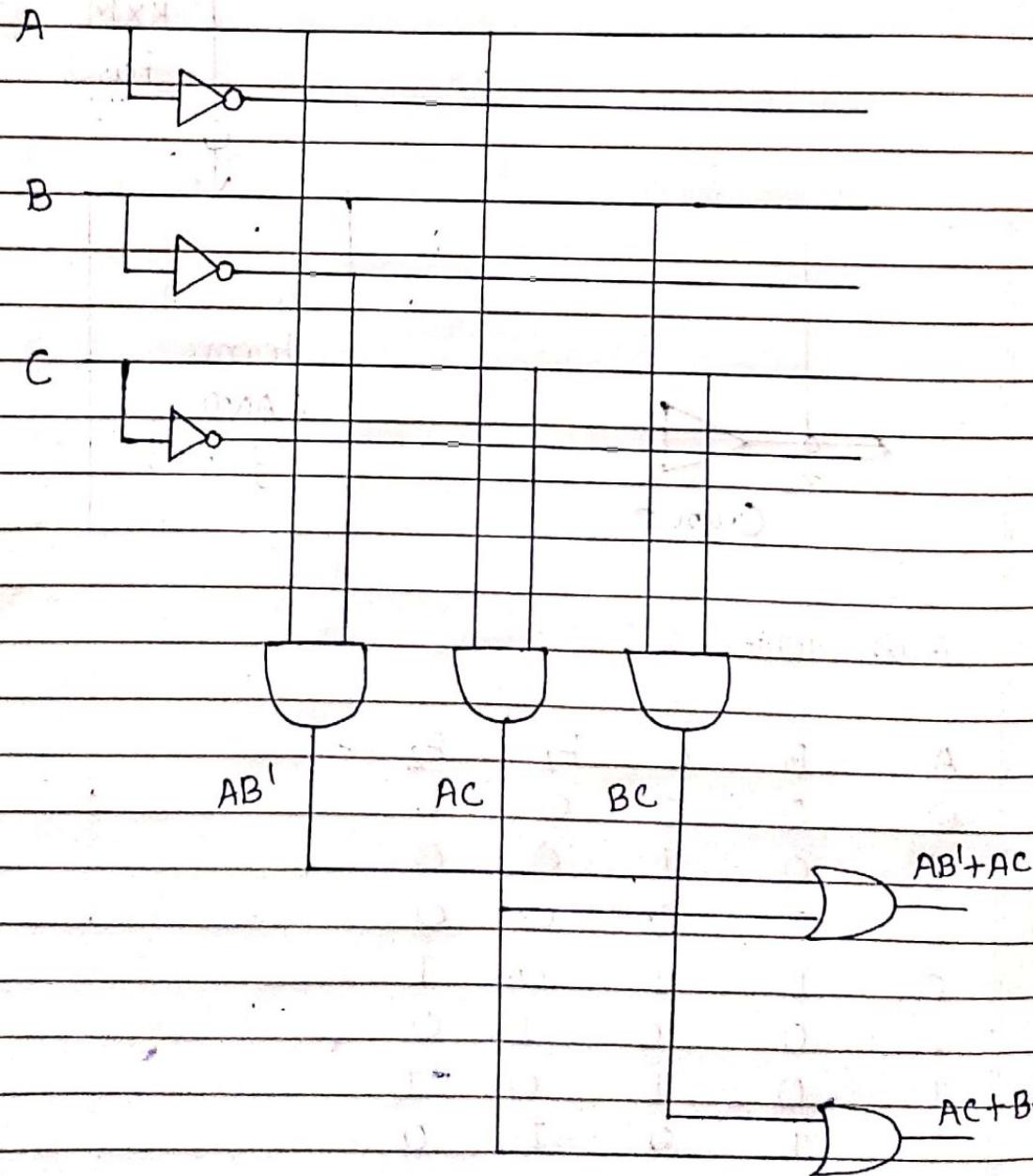
$F_1 = ABC' + AB'C + ABC' + ABC$ On
Simplify we get.

$$F_1 = AB + AC'$$

$$F_2 = A'BC + AB'C + ABC$$

$$F_2 = BC + AC$$

Realization.



- PLA is used for various Combinational circuit using buffer, AND gate and OR gate.
- In PLA all the minterms are not realized but only required minterms are implemented.
- As PLA has programmable AND gate array and programmable OR gate array it provides more flexibility.

⇒ Applications :

- PLA is used to provide Control Over datapath.
- PLA is used as counter.
- PLA is used as deorder demultiplexer.
- PLA is used as a Bus interface in program I/O.

Que.-4 Explain the Programmable Array Logic.

- Programmable Array Logic is a commonly used programmable logic device.
- It has programmable AND array and fixed OR array.
- Because only the AND array is programmable it is easier to use but not flexible as compared to programmable logic array.

- PAL's Only Limitation is Number of AND gates.
- PAL Consists of small programmable read only Memory and additional Output logic used to implement a particular desired logic function with limited Components.
- Main difference between PLA, PAL and ROM is their basic Structure.
- In PLA programmable AND gate is followed by programmable OR gate.
- In PAL programmable AND gate is followed by fixed OR gate.
- In ROM, fixed AND gate array is followed by programmable OR gate array.



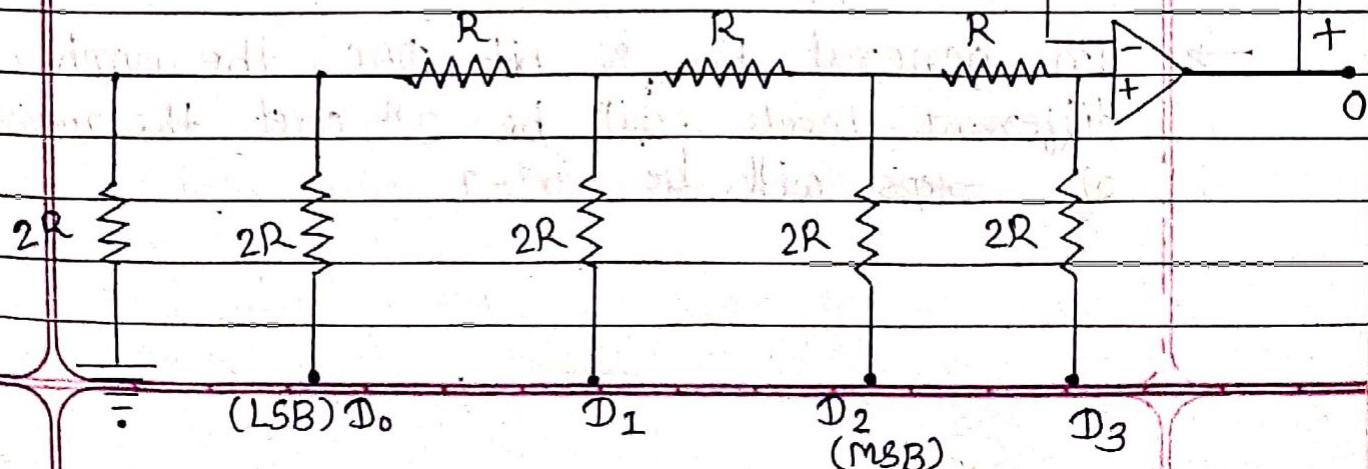
Advantage of PAL

- Highly efficient
- Low production Cost as Compared to PLA.
- Highly Secure
- High Reliability.
- Low power required for working.
- More flexible to design.

Assignment - 7

Que-1 Explain D/A Converter R-2R Ladder circuits with any one example.

- The R-2R Ladder type DAC is the most popular DAC.
- It uses a ladder network consisting Series parallel combinations of two resistors of values R and $2R$.
- The Operational amplifier Configured as Voltage follower is used to prevent Loading.
- Figure Shows the Circuit diagram of a R-2R Ladder type DAC having 4-bit digital input.
- When a digital Signal D_3, D_2, D_1, D_0 is applied at the input terminals of the DAC, an equivalent among Signal is produced at the output terminal.



Case 1 : When the input is 1000

Below figure

Resolution (Step Size)

The resolution of a DAC is defined as the smallest change that can occur in an analog output as a result of a change in digital input.

The resolution of a DAC is also defined as the reciprocal of the number of discrete steps in the fullscale output of the DAC.

$$\therefore \text{resolution} = \frac{\text{Step Size}}{\text{Full Scale}} \times 100\%$$

Since, full Scale = number of steps \times Step size

$$\therefore \text{resolution} = \frac{\text{total number of steps}}{2^N} \times 100\%$$

\rightarrow In general for N bit DAC, the number of different levels will be 2^N and the number of steps will be $2^N - 1$.

• Accuracy :

- The Accuracy of a DAC is usually Specified in terms of its full Scale errors and linearity error.
- The full Scale error is the Maximum Deviation of the DAC's Output from its expected Value.
- The linearity error is the Maximum Deviation of the analog output from the ideal output

• Settling time.

- The Operating Speed of a DAC is usually Specified by giving its Setting time.
- It is defined as the total time between the instant when the digital input changes and the time that the output enters a Specified error band.

• Offset Voltage.

- Ideally, the Output of a DAC Should be zero When the binary input is Zero.
- In practice, however, there is a very small Output Voltage under this Situation called the

Offset Voltage.

Monofonicity

- A DAC is said to be Monotonic if its Output increase as the binary input is incremented from one value to the next.
- This means that the Staircase Output will have no downward steps as the binary input.

Temperature Sensitivity

- The analog output voltage for any fixed digital input varies with temperature.

* Example

- An 8 bit DAC produces $V_{out} = 0.05 V$ for a digital input of 00000001. Find their full-scale output. What is the resolution?

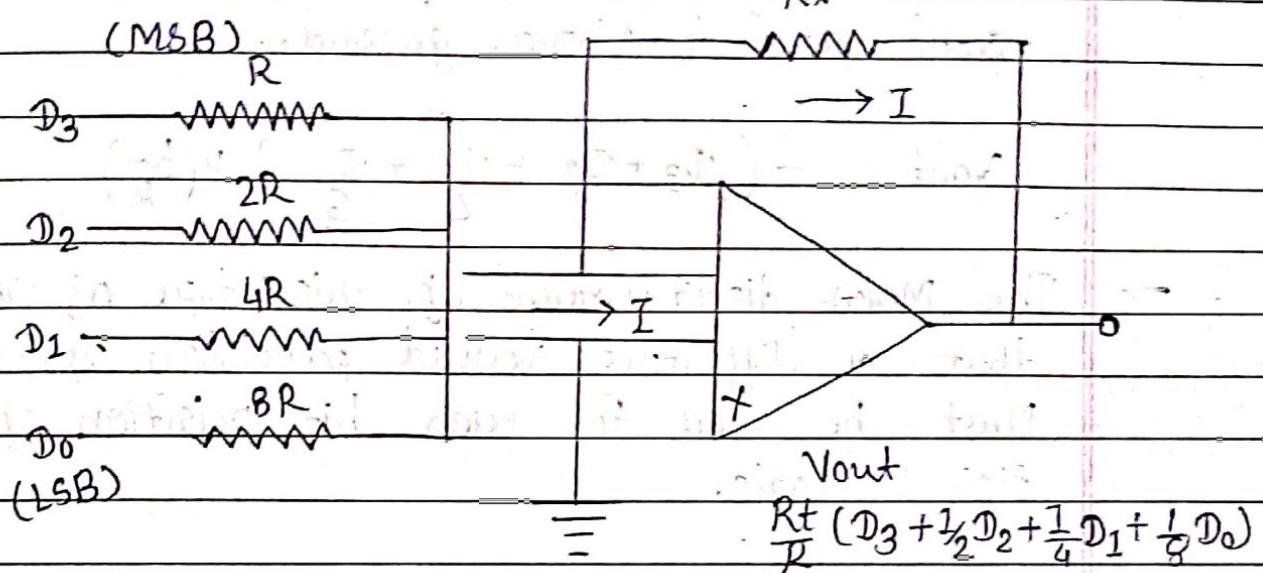
$$\text{Full Scale Output} = \text{Step Size} \times \text{No. of Steps}$$

$$\begin{aligned} &= 0.05 \times (2^8 - 1) = 0.05 \times 255 \\ &= 12.75 V \end{aligned}$$

$$\text{Resolution} = \frac{1}{255} \times 100\% = 0.392\%$$

V_{out} for an Input of 00101010 = 42×0.05
 $= 2.10 \text{ V}$

Ques-2 Explain D/A Converter binary weighted Resistor



- The diagram of the weighted-resistor DAC is shown in Figure.
- The Operational amplifier is used to produce a weighted sum of the digital inputs. Where the weights are proportional to the weights of the bit positions of inputs.
- Since the op-amp is connected as an inverting amplifier each input is amplified by a factor equal to the ratio of the feedback resistance.

→ The MSB D_3 is amplified by R_s/R_n . D_2 is amplified by $R_t/2R$, D_2 is also by $R_s/8R$ and D_0 the LSB is amplified by $R_s/8R$.

- The inverting terminal of the op-amp in figure acts as a virtual ground.

$$V_{out} = - \left(\frac{D_3}{2} + \frac{D_2}{4} + \frac{D_1}{8} + D_0 \right) \times \left(\frac{R_t}{R} \right)$$

→ The main disadvantage of this type of DAC is that a different valued precision resistor must be used for each bit position of the digital logic.

* Example

For the weighted resistor DAC, determine

(a) The weight of each input bit if the inputs are 0V and 5V.

(b) The full-scale output if $R_i = R_o = 1\text{ k}\Omega$.

* Solution.

(c) If MSB passes with a gain of 1
So its weight = 5V.

The Next bit passes with a gain of $1/2$

So its weight = 2.5 V

The following bit passes with a gain of $1/4$

So its weight = 1.25 V

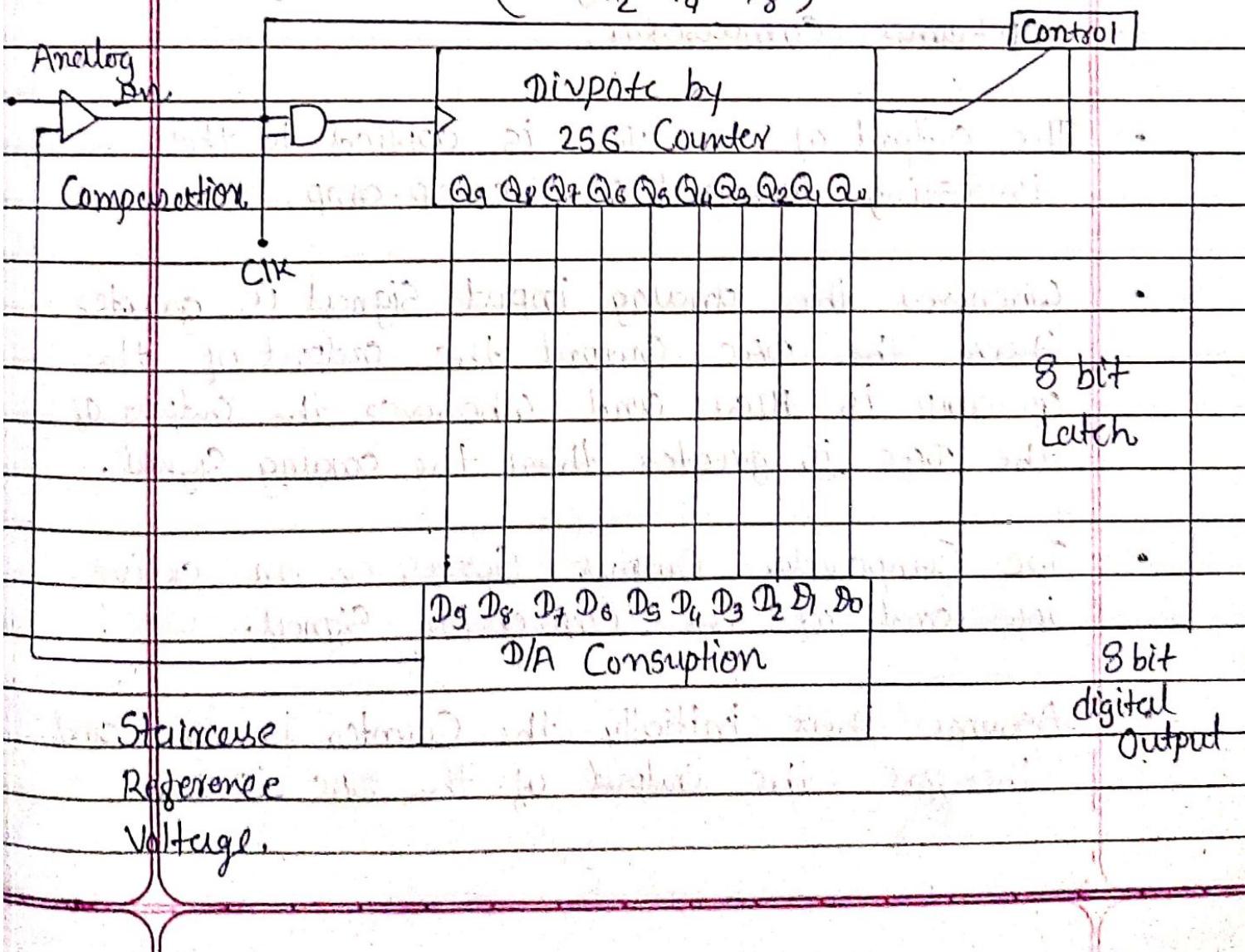
The LSB passes with a gain of $1/8$

So its weight = 0.625 V

(b) Therefore, the full scale output when

$$R_f = R = 1 \text{ k}\Omega$$

$$V_{out} = -(5 + \frac{s_1}{2} + \frac{s_2}{4} + \frac{s_3}{8}) = -9.375 \text{ V}$$



- This is a Simplest type of the A/D Converter.
- It employs a binary counter, a voltage comparator, a Control Circuit, an AND gate, latches and a D/A Converter.
- It is also called a digital ramp DAC because the waveform at the Output of the DAC is Step by Step ramp.
- The analog Signal to be Converted is applied to the non-inverting terminal of the op-amp Comparator.
- The Output of the DAC is applied to the inverting terminal of the op-amp.
- Whenever the analog input Signal is greater than the DAC Output the output of the op-amp is HIGH and whenever the output of the DAC is greater than the analog Signal.
- The Comparator Output Serves as an active low end of the Conversion Signal.
- Assume that initially the Counter is reset and therefore the Output of the DAC is zero.

These binary States are converted into reference analog voltage by the DAC.

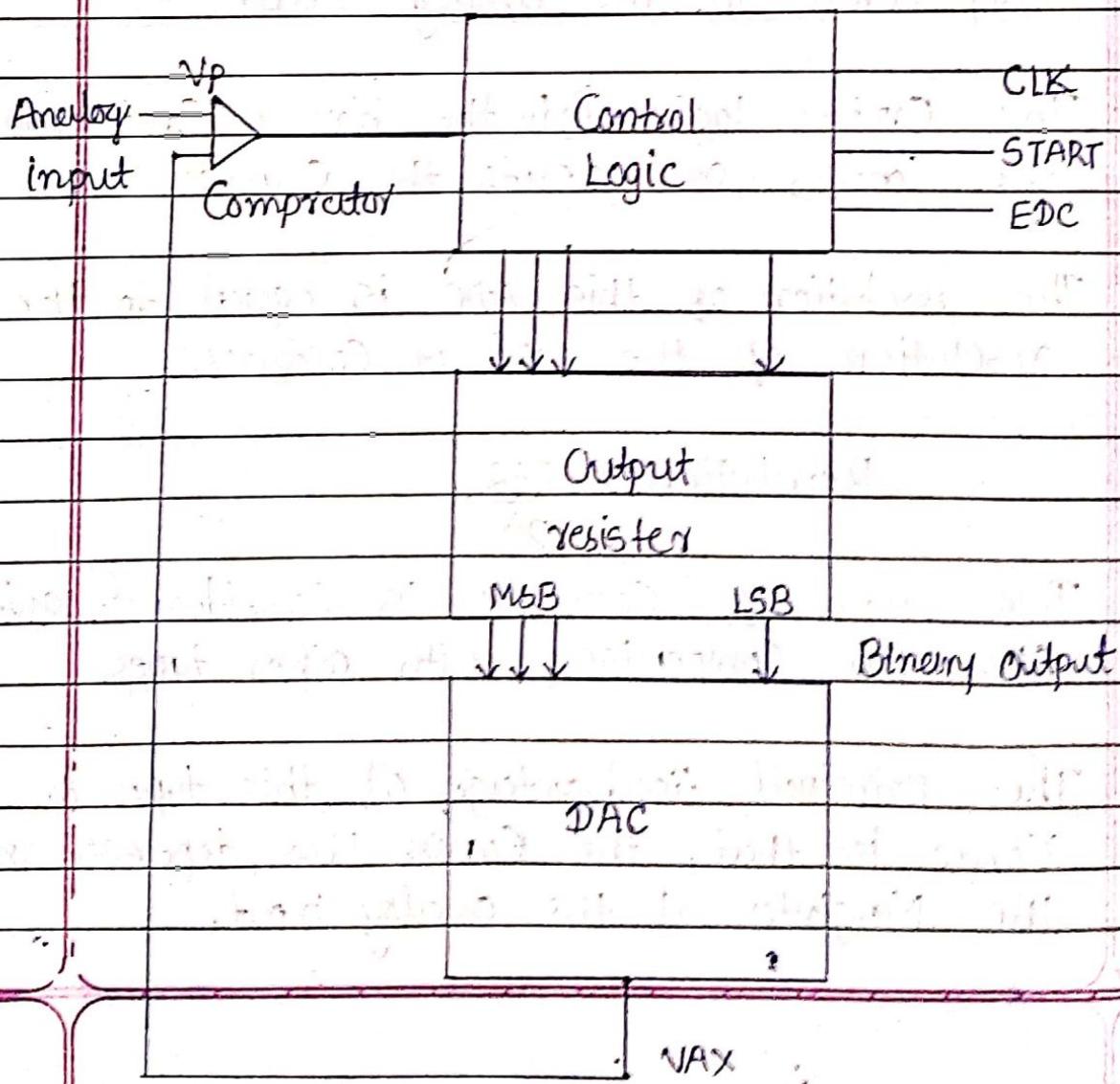
- The Counter Continues to advance from one State to the next produce successive larger steps in the reference Voltage.
- When the Staircase Output Voltage reaches the value of the analog Signal. The Compensator outputs a low and the AND gate is disabled.
- The Count it reached is the digital Output proportional to the analog input.
- The Control logic loads the binary Cout into the latches and resets the Counter.
- The resolution of this ADC is equal to the resolution of the DAC it contains.

$$\text{Resolution} = \frac{\text{FSR}}{2^N}$$

- This type of Converter is Considered quite slow in Comparison with other types.
- The Principal disadvantage of this type of Counter is that, the Count fine depends on the Magnitude of the analog input.

Que-4 Explain Successive approximation type ADC

- The Successive-approximation Converter is one of the most widely used types of ADC.
- It has a much shorter conversion time taken than the other types, with the exception of the parallel type.
- It also has a fixed conversion time which is not dependent on the value of the analog input.



- Figure Shows a basic block diagram of a 4-bit Successive approximation type ADC.

- It consists of a DAC, an Output register, a Comparator and Control Circuity or Logic.

- The basic operation is as follows.

→ The bits of DAC are enabled one at a time, starting with the MSB.

→ As each bit is enabled the comparator produces an output that indicates whether the analog input voltage is greater or less than the output of the DAC.

- If the D/A output is greater than the analog input, the Comparator output is low, causing the bit in the Control register to reset.

- The System enables the MSB first, then the next significant bit and so on.

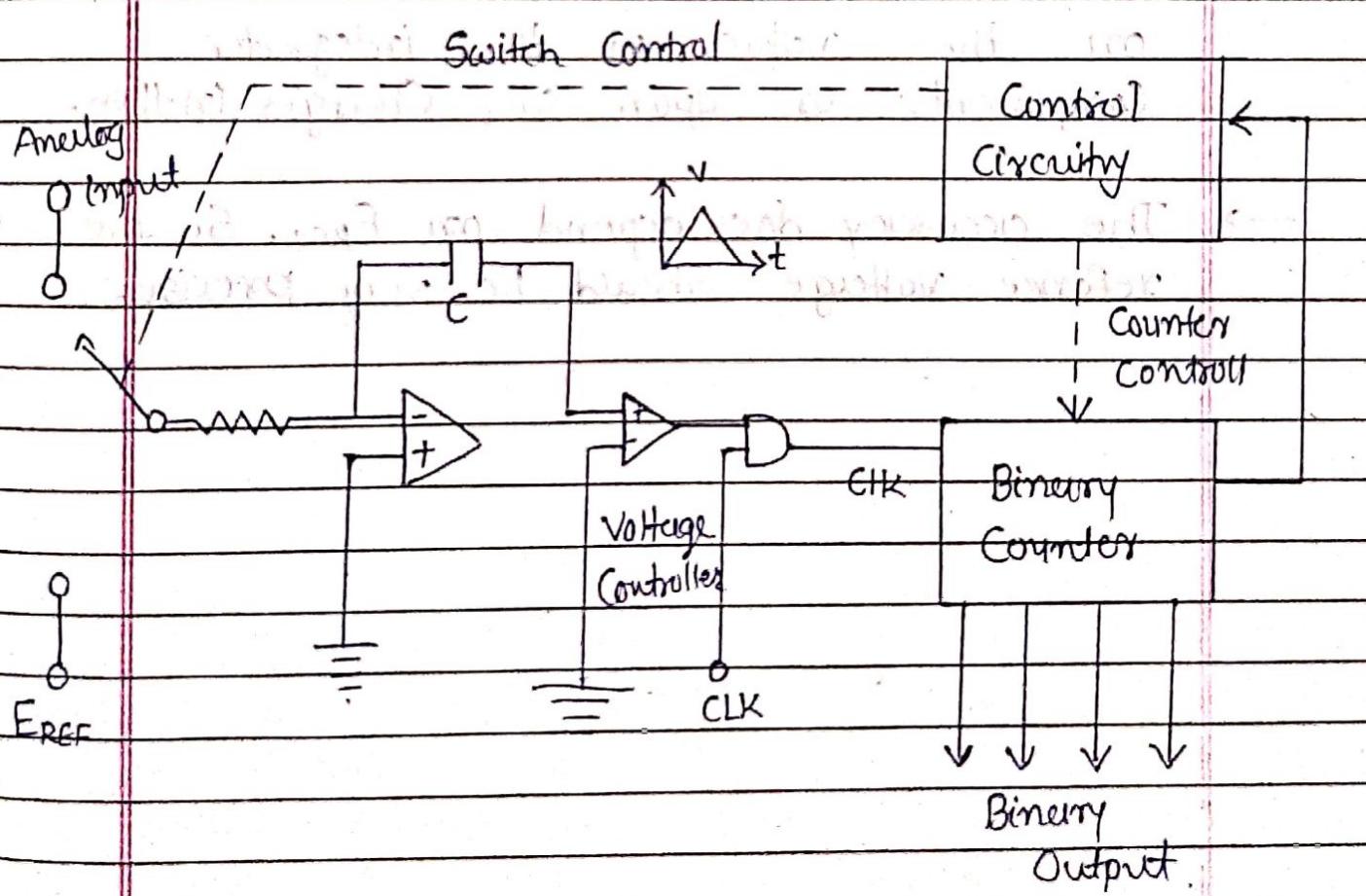
- After all the bits of the DAC have been tried, the conversion cycle is complete.

- The processing of each bit takes one clock cycle: so the total conversion time in N-bit SA-type ADC will be N clock cycle.

Que.-5 Explain Dual Slope type A/D Converter.

- The Dual-Slope Converter is one of the slowest conversion but is relatively inexpensive because it does not require precision Components such as DAC or VCO.
- Another Advantage of the dual-Slope ADC is its low Sensitivity of noise and variations in its component values caused by temperature changes.
- Because of its large Conversion time, the dual slope ADC is not used in any data acquisition applications.
- The Major Applications of this type of Converter are in digital Voltmeters, Multimeters etc. Where slow conversions are not a problem.
- Since it is not fast enough its use is restricted to Signals heavily low to Medium frequencies.

- A dual-Slope ADC used an Operational amplifier to integrate the analog input.
- The Output of the integrator is a ramp, whose slope is proportional to the input Signal. Fin. Since the Components R and C are fixed.
- If the ramp is allowed to continue for a fixed time, the Voltage it reaches in that time depends on the Slope of the ramp and therefore on the Value of Fin.



- The basic principle of the integrating ADC is that, the voltage reached by the ramp Controls the Length of time that the binary counter.
- In the dual-slope ADC, two integrations are performed
- Assume that the Counter is present and the Output of the integrator is zero.
- The accuracy of the Counter does not depend on the values of the integrator Components or upon any Charges in them.
- The accuracy does depend on E_{ref} . So the reference voltage Should be very precise.