B.TECH DEGREE EXAMINATION,JANUARY 2023

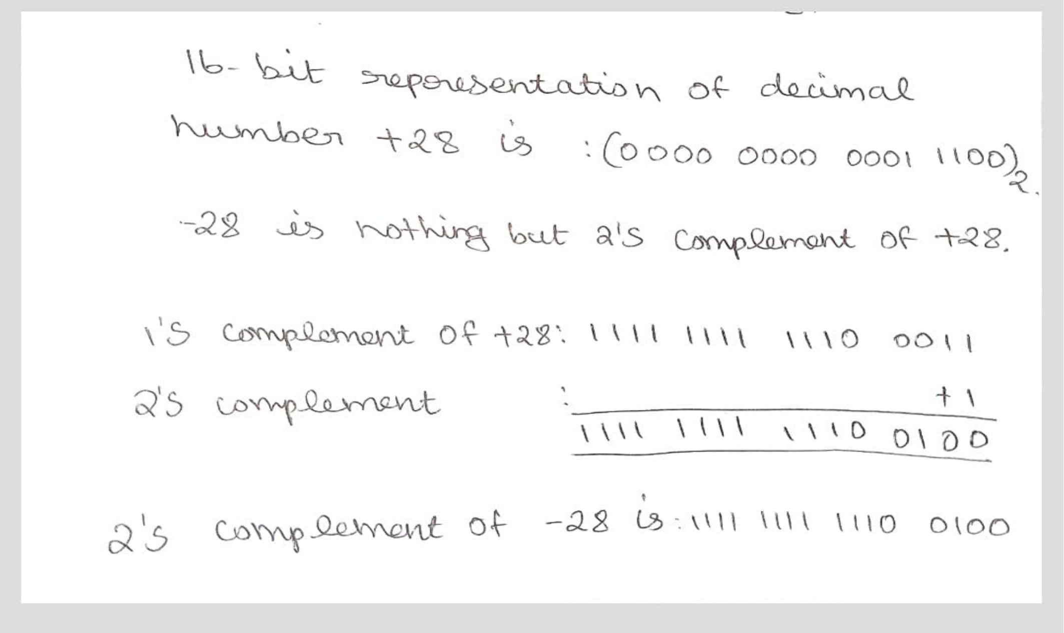
THIRD SEMESTER

INFORMATION TECHNOLOGY

DIGITAL SYSTEM DESIGN

PART-A(10X2=20 MARKS)

1.Write 16-bit 2'scomplement representation for the decimal number -28.



16- bit reporesentation of decimal number +28 is :$(0000\ 0000\ 0001\ 1100)_2$.

-28 is nothing but 2's complement of +28.

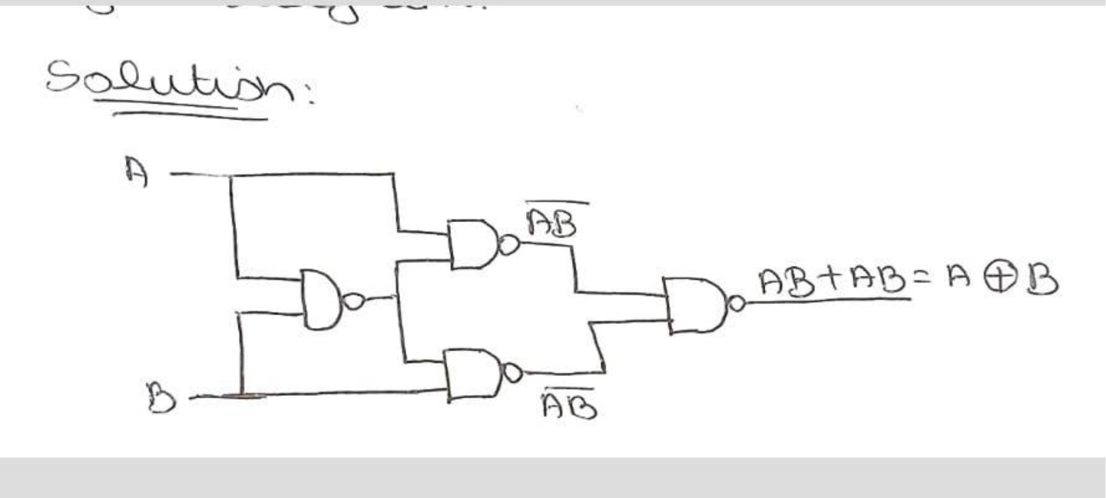
1's complement of +28: 1111 1111 1110 0011

2's complement : +1

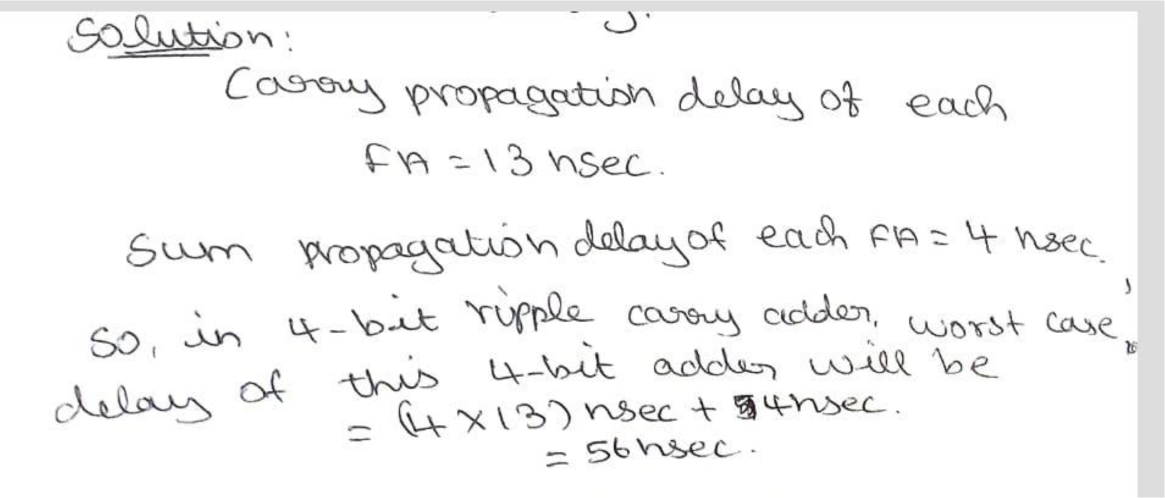
1111 1111 1110 0100

2's complement of -28 is: 1111 1111 1110 0100

2.Find the minimum number of 2-input NAND gates required to implement a 2 input XOR gates and draw the logic diagram.



3.A four bit ripple carry adder is realized using 4 identical full adder.The carry propagation delay of each (FA) is 13ns.Calculate the worst case delay.



Solution:

Carry propagation delay of each FA = 13 nsec.

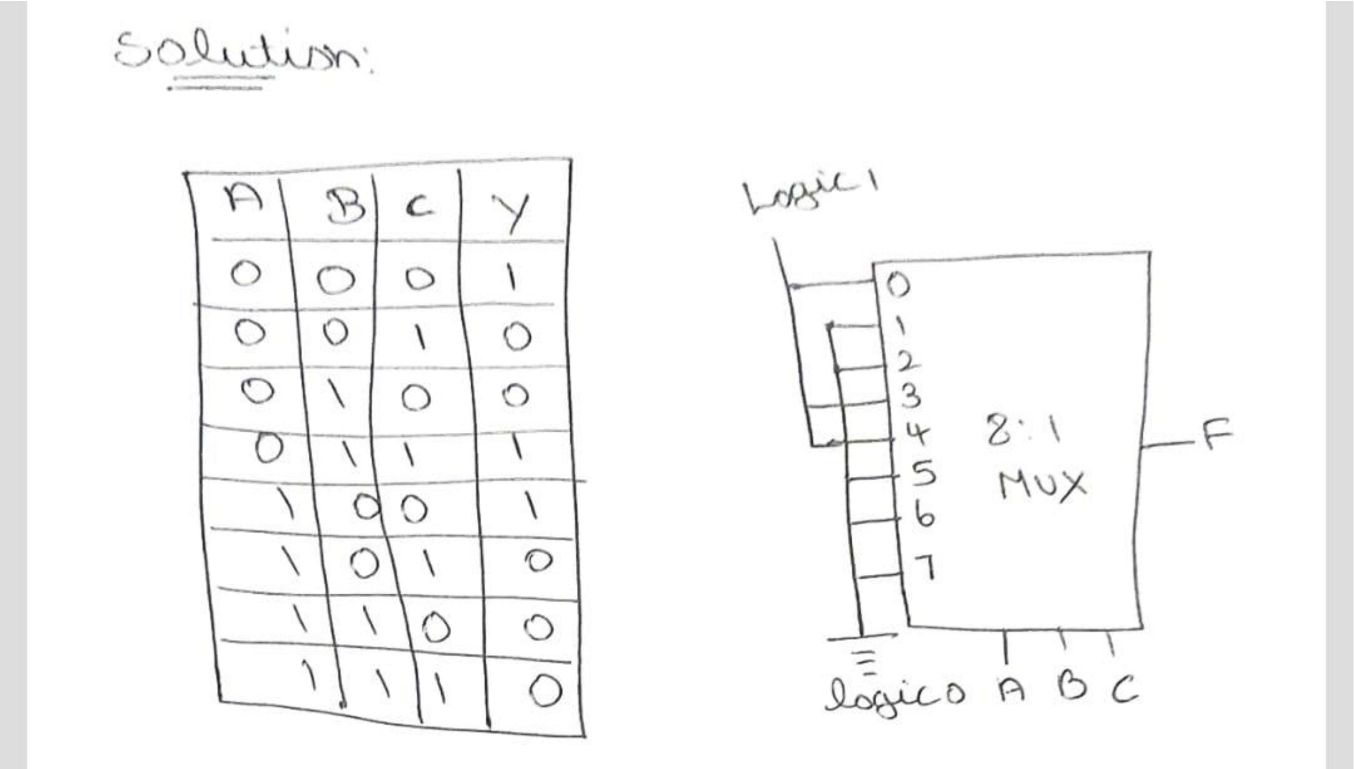
Sum propagation delay of each FA = 4 nsec.

So, in 4-bit ripple carry adder, worst case delay of this 4-bit adder will be

= (4 × 13) nsec + 4nsec.

= 56nsec.

4**.implement the given function using multiplexerF(A,B,C)=£$_m$(0,3,4).**



Solution:

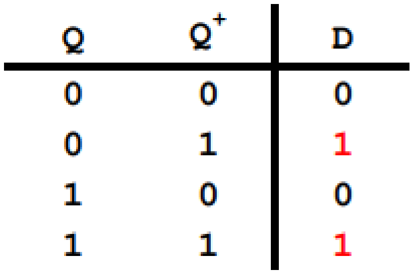
A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

5.what is sequential circuit?How does it differ from combinational circuit?

Parameters	**Combinational Circuit**	**Sequential Circuit**
Meaning and Definition	It is a type of circuit that generates an output by relying on the input it receives at that instant, and it stays independent of time.	It is a type of circuit in which the output does not only rely on the current input. It also relies on the previous ones.
Feedback	A Combinational Circuit requires no feedback for generating the next output. It is because its output has no dependency on the time instance.	The output of a Sequential Circuit, on the other hand, relies on both- the previous feedback and the current input. So, the output generated from the previous inputs gets transferred in the form of feedback. The circuit uses it (along with inputs) for generating the next output.

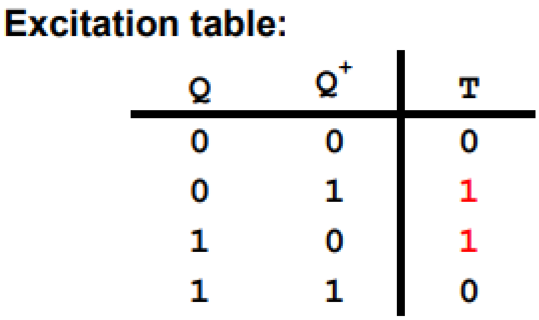
6.**Give the excitation table for D-FF AND T-FF.**

D-FLIP FLOP



Q	Q^+	D
0	0	0
0	1	1
1	0	0
1	1	1

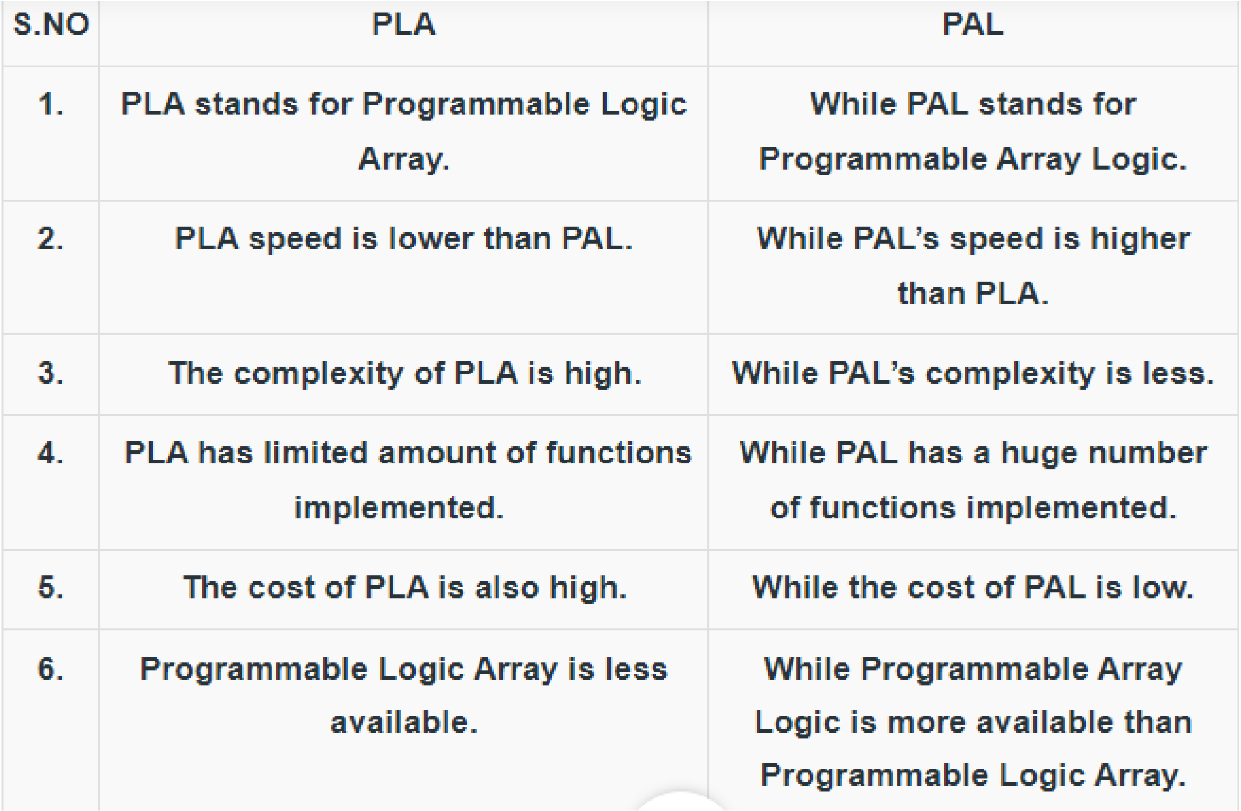
T-FLIP FLOP



Excitation table:

Q	Q^+	T
0	0	0
0	1	1
1	0	1
1	1	0

7.List the major difference between PAL AND PLA.



S.NO	PLA	PAL
1.	PLA stands for Programmable Logic Array.	While PAL stands for Programmable Array Logic.
2.	PLA speed is lower than PAL.	While PAL's speed is higher than PLA.
3.	The complexity of PLA is high.	While PAL's complexity is less.
4.	PLA has limited amount of functions implemented.	While PAL has a huge number of functions implemented.
5.	The cost of PLA is also high.	While the cost of PAL is low.
6.	Programmable Logic Array is less available.	While Programmable Array Logic is more available than Programmable Logic Array.

8.What are the steps for the design of asynchronous sequential circuit?

1. Create a state table or state diagram from the given problem statement.
2. Create a new reduced state table by removing all the redundant states.
3. Create the transition table.
4. Write the excitation and output Boolean equations and simplify them.
5. Draw the logic diagram.

9.Define gate-level modelling?

Gate level modeling is virtually the lowest level of abstraction because the switch-level abstraction is rarely used. Gate level modeling is **used to implement the lowest-level modules in a design, such as multiplexers, full-adder, etc**. Verilog has gate primitives for all basic gates.

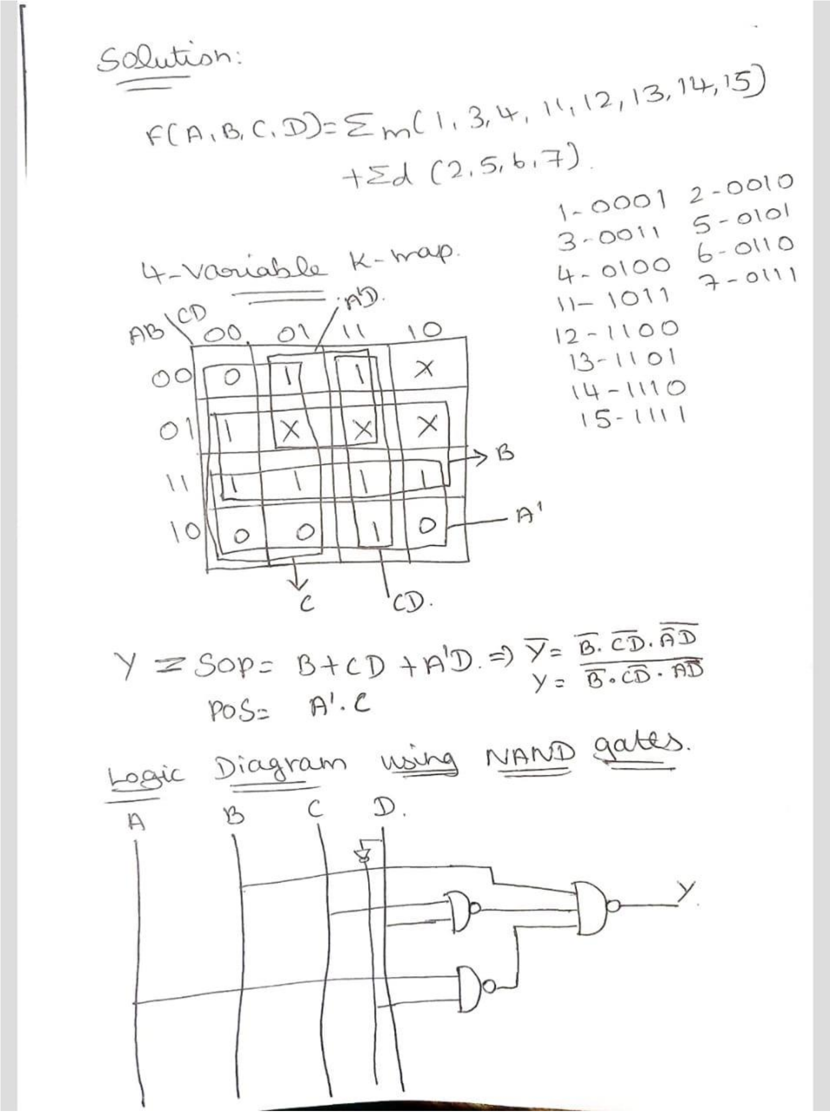
10.Write the syntax for even construct.

A HDL is a domain specific language, specifically designed to support the description of digital logic circuits and clock driven sequential logic. As such a HDL contains **special constructs to enable the description of digital hardware and RTL elements**.

PART-B(5X11=55 MARKS)

UNIT 1

11.Simplify the following Boolean function using K-MAP and implement using only NAND gates F(A,B,C,D)=£M(1,3,4,11,12,13,14,15)+ £D((2,5,6,7).



Solution:

$F(A,B,C,D)=\Sigma m(1,3,4,11,12,13,14,15)$
$+\Sigma d(2,5,6,7)$.

1-0001	2-0010
3-0011	5-0101
4-0100	6-0110
11-1011	7-0111
12-1100	
13-1101	
14-1110	
15-1111	

4-Variable K-map.

AB \ CD	00	01	11	10
00	0	1	1	X
01	1	X	X	X
11	1	1	1	1
10	0	0	1	0

$Y = Sop= B+CD+A'D. \Rightarrow \overline{Y}= \overline{B}.\overline{CD}.\overline{\overline{A}D}$

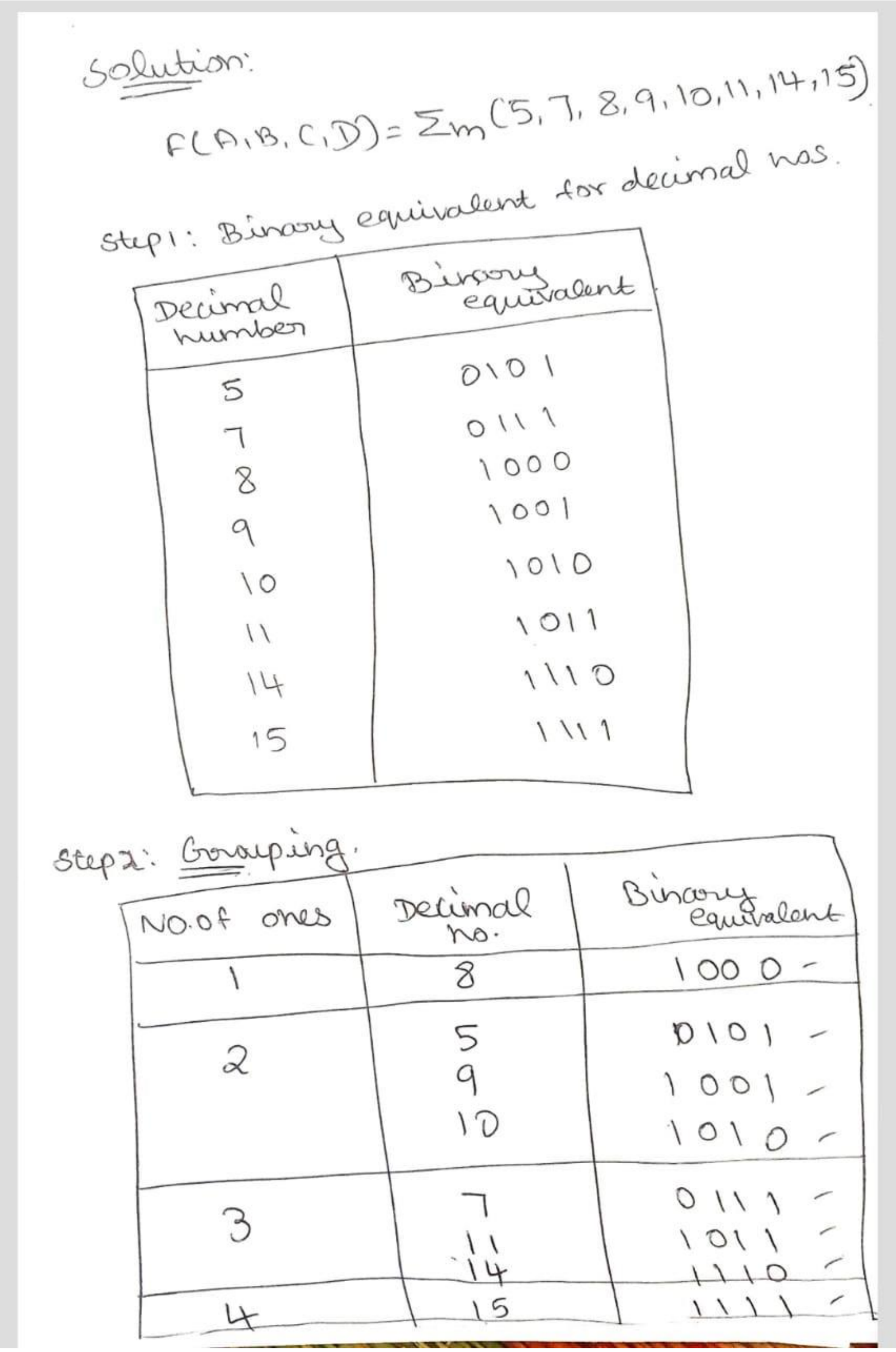
$Y= \overline{\overline{B}\cdot\overline{CD}\cdot\overline{\overline{A}D}}$

$PoS= A'.C$

Logic Diagram using NAND gates.

OR

12. Minimize the given Boolean function using Quine McCluskey F(A,B,C,D)= £M(5,7,8,9,10,11,14,15).



Solution:

$F(A,B,C,D) = \Sigma m(5,7,8,9,10,11,14,15)$

step1: Binary equivalent for decimal nos.

Decimal number	Binary equivalent
5	0101
7	0111
8	1000
9	1001
10	1010
11	1011
14	1110
15	1111

step2: Grouping.

No.of ones	Decimal no.	Binary equivalent
1	8	1000 ✓
2	5	0101 ✓
	9	1001 ✓
	10	1010 ✓
3	7	0111 ✓
	11	1011 ✓
	14	1110 ✓
4	15	1111 ✓

step3: 2-cell Grouping.

2-cell	Binary equivalent.
(8,9)	100- ✓
(8,10)	10-0 ✓
(5,7)	01-1 → Ⓐ
(9,11)	10-1 ✓
(10,11)	101- ✓
(10,14)	1-10 ✓
(7,15)	-111 → Ⓑ
(11,15)	1-11 ✓
(14,15)	111- ✓

Step4: 4-cell Grouping

4-cell	Binary equivalent.
(8,9,10,11)	10--
(8,10,9,11)	10--
(10,11,14,15)	1-1-
(10,14,11,15)	1-1-

Step 5: Remove Replicant.

4-cell.	Binary equivalent.
(8, 9, 10, 11)	10 - -
(10, 11, 14, 15).	1 - 1 -

Prime implicants are,

	ABCD	
(5, 7)	01 - 1	$A'BD$
(7, 15)	- 111	BCD
(8, 9, 10, 11)	10 - -	AB'
(10, 11, 14, 15)	1 - 1 -	AC.

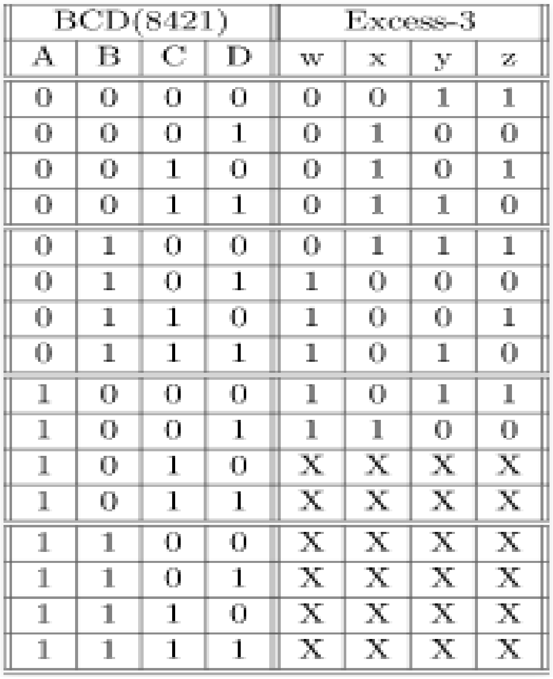
Prime Implicant Chart:

	5	7	8	9	10	11	14	15.
⊛ (5, 7)	⊛	*						
(7, 15)		*						*
⊛ (8, 9, 10, 11)			*	⊛	*	*		
⊛ (10, 11, 14, 15)					*	*	⊛	*

$$SOP = A'BD + AB' + AC.$$

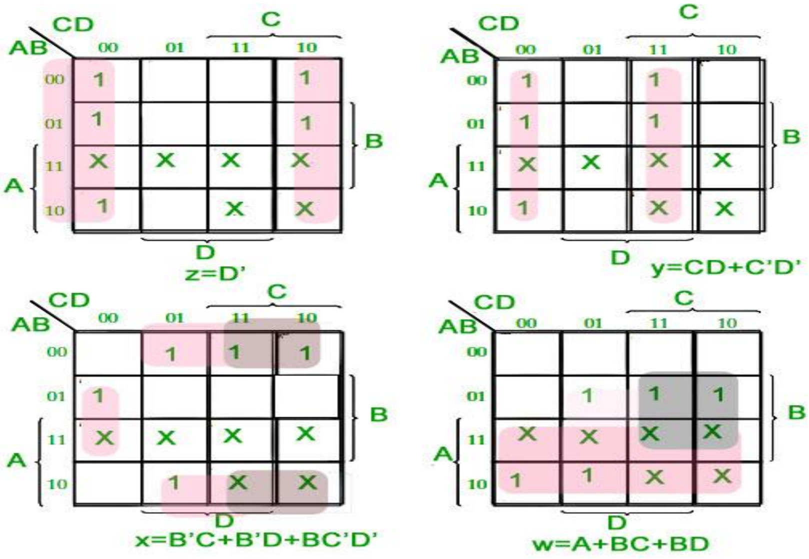
UNIT 2

13.Design a circuit that converts BCD code to Excess-3 code converter.



BCD(8421)				Excess-3			
A	B	C	D	w	x	y	z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

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.



z=D'

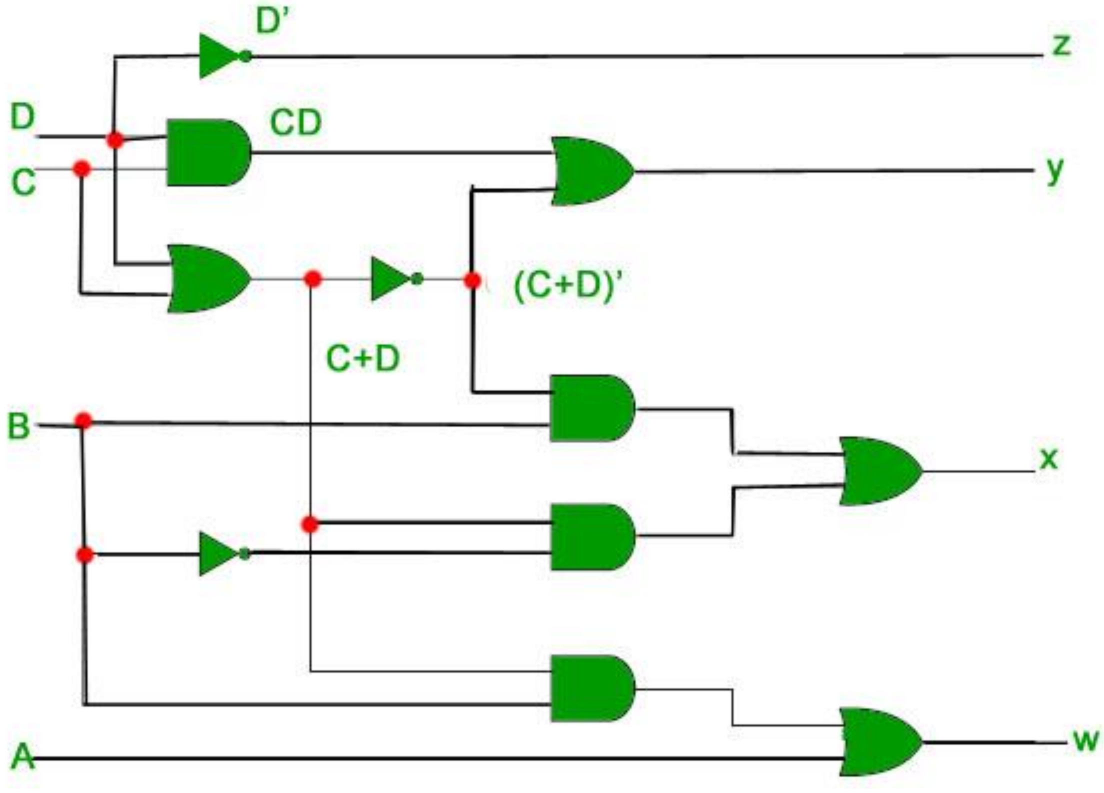
y=CD+C'D'

x=B'C+B'D+BC'D'

w=A+BC+BD

Corresponding minimized Boolean expressions for Excess-3 code bits –

The corresponding digital circuit-



OR

14.Implement the switching function F=£M(3,6,7,8,10,12,13,14,15)using an 8 input multiplexer.

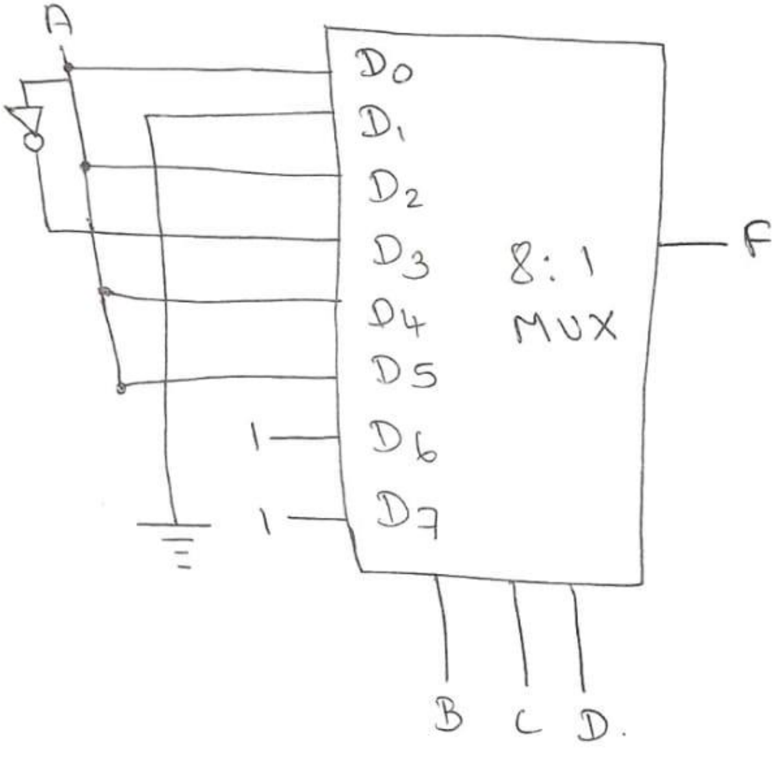
In the given boolean expression, there are 4 variables. We should use 2^4 : 1 = 16 : 1 multiplexer. But as per the question, it is to be implemented with 8 : 1 mux.

For 8 : 1 multiplexer, there should be 3 selection lines. So from the given 4 variables, the 3 least significant variables(B, C, D) are used as selection line inputs.

The 8 inputs are derived using the implementation table shown below

	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7.
$\bar{A}$	0	1	2	[✓ 3]	4	5	[✓ 6]	[✓ 7]
A	[✓ 8]	9	[✓ 10]	11	[✓ 12]	[✓ 13]	[✓ 14]	[✓ 15]
	A	0	A	$\bar{A}$	A	A	1	1

from the derived input 8:1 multiplexer, can be drawn as below.



UNIT 3

15.A sequential circuit has two flip flops A and B,two inputs x and y and one output z.the flip flop input equations and circuit output equations are:

J_A=Bx+B’y’

K_A=B’xy’

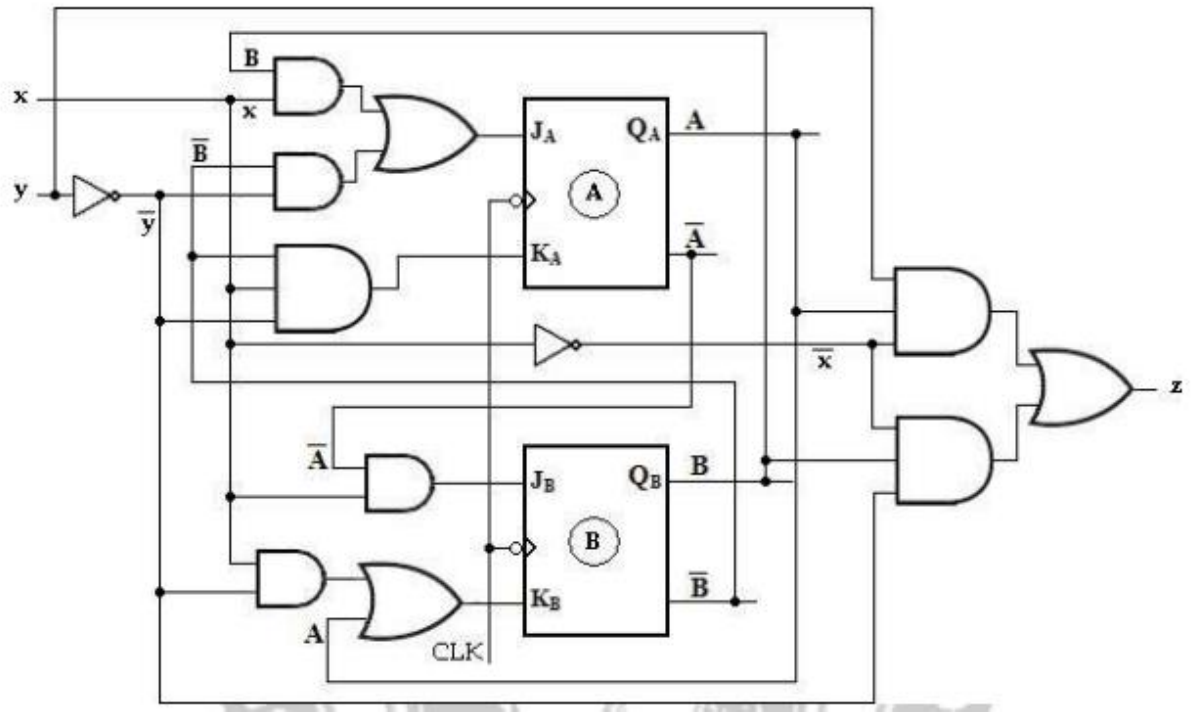
J_B=A’x

K_B=A+xy’

Z=Ax’y’+Bx’y’

Draw the logic diagram and tabulate the state table.

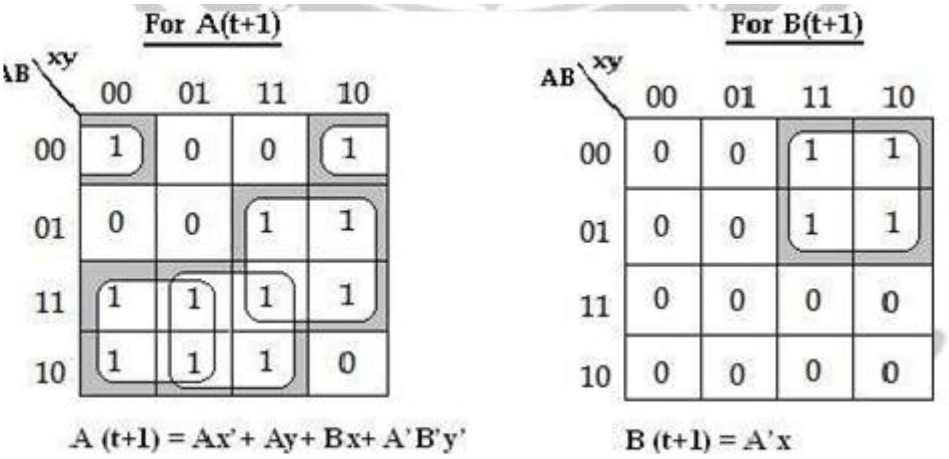
State diagram:



STATE TABLE:

Present state		Input		Flip-Flop Inputs				Next state		Output
A	**B**	**x**	**y**	**J_A= Bx+B’y’**	**K_A= B’xy’**	**J_B= A’x**	**K_B= A+xy’**	**A(t+1)**	**B(t+1)**	**z**
0	0	0	0	1	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	1	1	1	1	1	0
0	0	1	1	0	0	1	0	0	1	0
0	1	0	0	0	0	0	0	0	0	1
0	1	0	1	0	0	0	0	0	0	0
0	1	1	0	1	0	1	1	1	1	0
0	1	1	1	1	0	1	0	1	1	0
1	0	0	0	1	0	0	1	1	0	1
1	0	0	1	0	0	0	1	1	0	0
1	0	1	0	1	1	0	1	0	0	0
1	0	1	1	0	0	0	1	1	0	0
1	1	0	0	0	0	0	1	1	0	1
1	1	0	1	0	0	0	1	1	0	0
1	1	1	0	1	0	0	1	1	0	0
1	1	1	1	1	0	0	1	1	0	0

STATE EQUATION:



For A(t+1)

AB \ xy	00	01	11	10
00	1	0	0	1
01	0	0	1	1
11	1	1	1	1
10	1	1	1	0

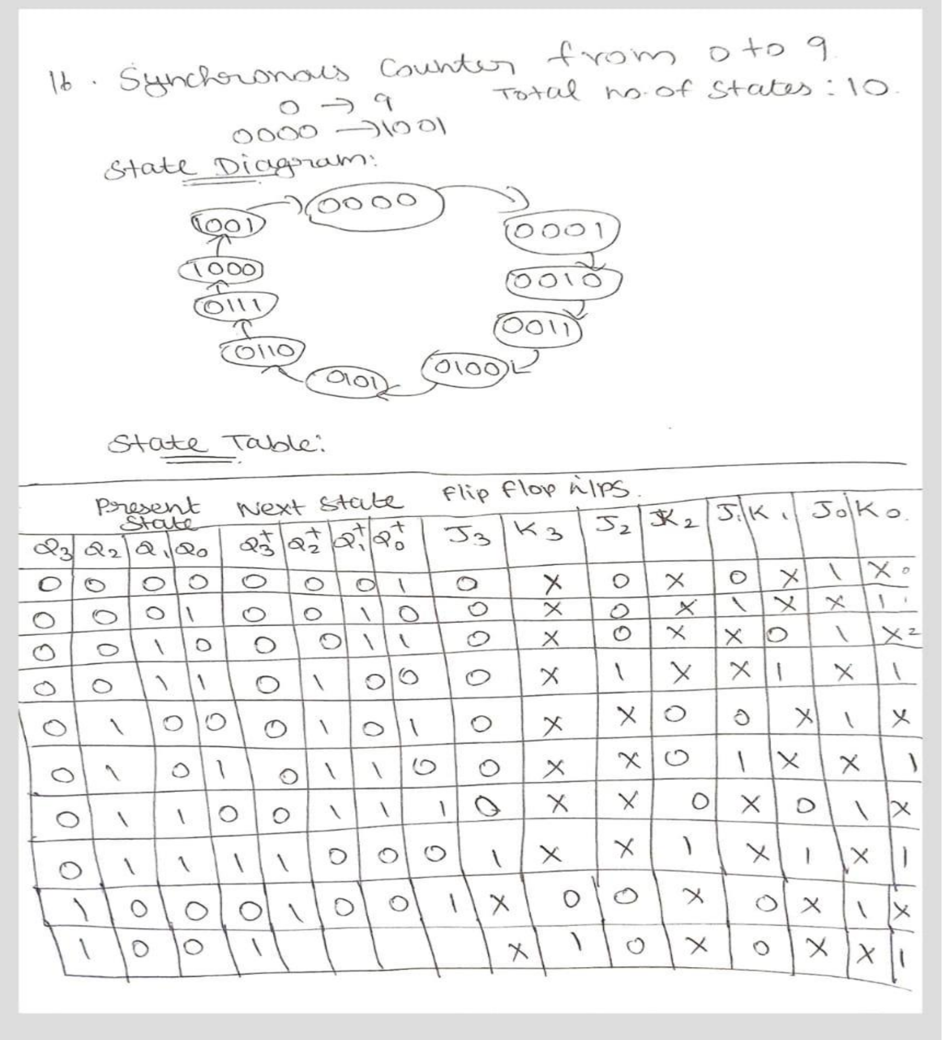
$A(t+1) = Ax' + Ay + Bx + A'B'y'$

For B(t+1)

AB \ xy	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	0	0

$B(t+1) = A'x$

16.Design a BCD synchronous counter using JK flip flop.



16 · Synchronous Counter from 0 to 9

Total no. of States : 10.

$0 \rightarrow 9$

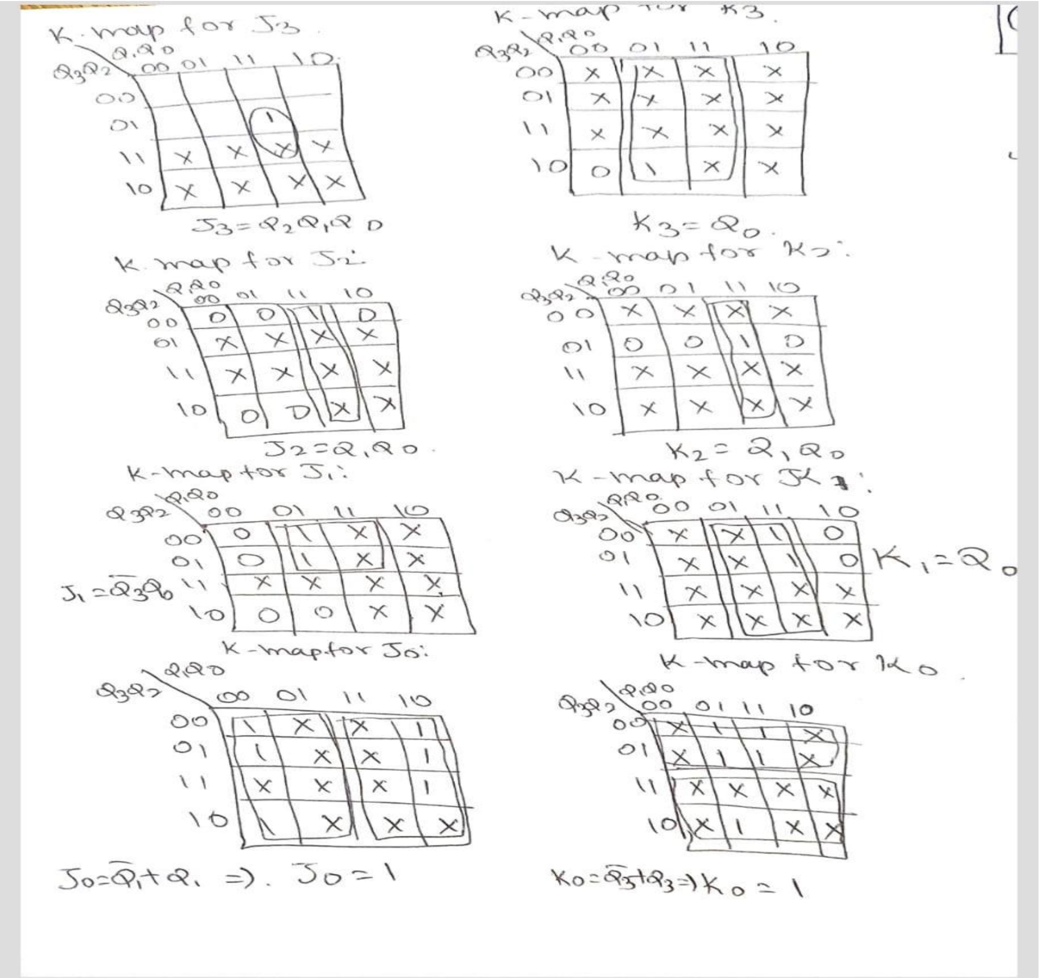
$0000 \rightarrow 1001$

State Diagram:

0000
0001
0010
0011
0100
0101
0110
0111
1000
1001

State Table:

Present State				Next State				Flip flop i/ps							
Q_3	Q_2	Q_1	Q_0	Q_3^+	Q_2^+	Q_1^+	Q_0^+	J_3	K_3	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	0	0	1	0	X	0	X	0	X	1	X
0	0	0	1	0	0	1	0	0	X	0	X	1	X	X	1
0	0	1	0	0	0	1	1	0	X	0	X	X	0	1	X
0	0	1	1	0	1	0	0	0	X	1	X	X	1	X	1
0	1	0	0	0	1	0	1	0	X	X	0	0	X	1	X
0	1	0	1	0	1	1	0	0	X	X	0	1	X	X	1
0	1	1	0	0	1	1	1	0	X	X	0	X	0	1	X
0	1	1	1	1	0	0	0	1	X	X	1	X	1	X	1
1	0	0	0	1	0	0	1	X	0	0	X	0	X	1	X
1	0	0	1					X	1	0	X	0	X	X	1



K.map for J_3.

$Q_3Q_2 \backslash Q_1Q_0$	00	01	11	10
00				
01			1	
11	X	X	X	X
10	X	X	X	X

$J_3 = Q_2Q_1Q_0$

K-map for K_3.

$Q_3Q_2 \backslash Q_1Q_0$	00	01	11	10
00	X	X	X	X
01	X	X	X	X
11	X	X	X	X
10	0	1	X	X

$K_3 = Q_0$.

K. map for J_2:

$Q_3Q_2 \backslash Q_1Q_0$	00	01	11	10
00	0	0	1	0
01	X	X	X	X
11	X	X	X	X
10	0	0	X	X

$J_2 = Q_1Q_0$.

K-map for K_2:

$Q_3Q_2 \backslash Q_1Q_0$	00	01	11	10
00	X	X	X	X
01	0	0	1	0
11	X	X	X	X
10	X	X	X	X

$K_2 = Q_1Q_0$

K-map for J_1:

$Q_3Q_2 \backslash Q_1Q_0$	00	01	11	10
00	0	1	X	X
01	0	1	X	X
11	X	X	X	X
10	0	0	X	X

$J_1 = \bar{Q}_3Q_0$

K-map for JK_1:

$Q_3Q_2 \backslash Q_1Q_0$	00	01	11	10
00	X	X	1	0
01	X	X	1	0
11	X	X	X	X
10	X	X	X	X

$K_1 = Q_0$

K-map for J_0:

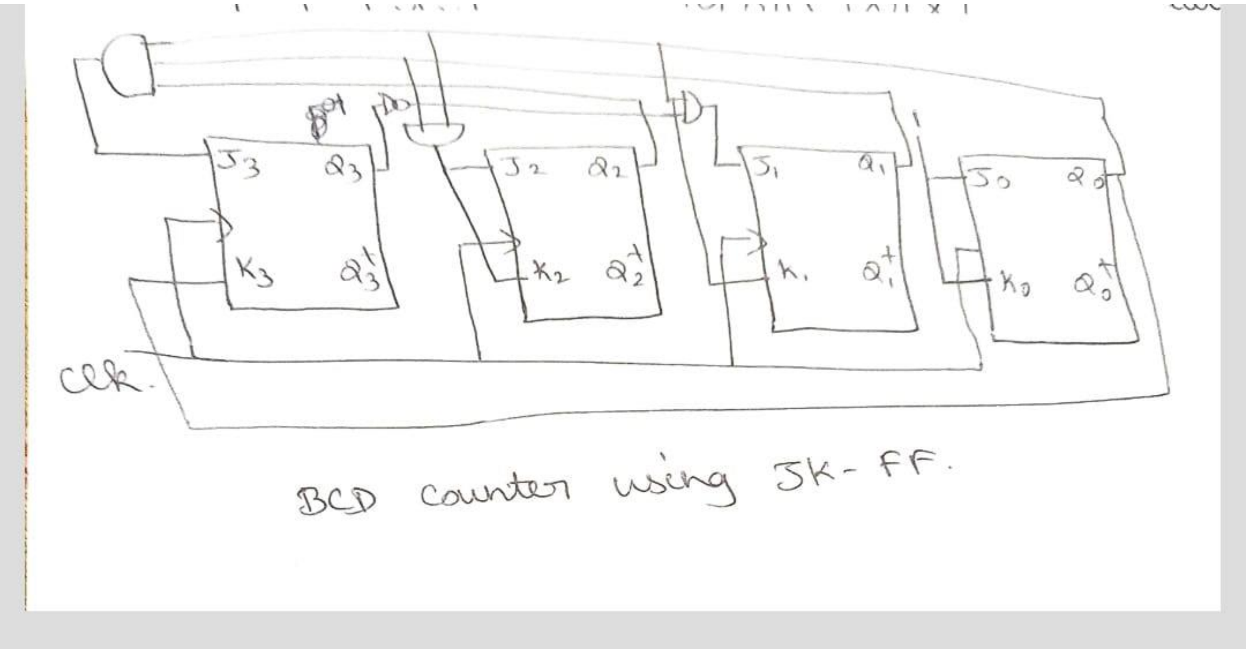
$Q_3Q_2 \backslash Q_1Q_0$	00	01	11	10
00	1	X	X	1
01	1	X	X	1
11	X	X	X	1
10	1	X	X	X

$J_0 = \bar{Q}_1 + Q_1 \Rightarrow J_0 = 1$

K-map for K_0.

$Q_3Q_2 \backslash Q_1Q_0$	00	01	11	10
00	X	1	1	X
01	X	1	1	X
11	X	X	X	X
10	X	1	X	X

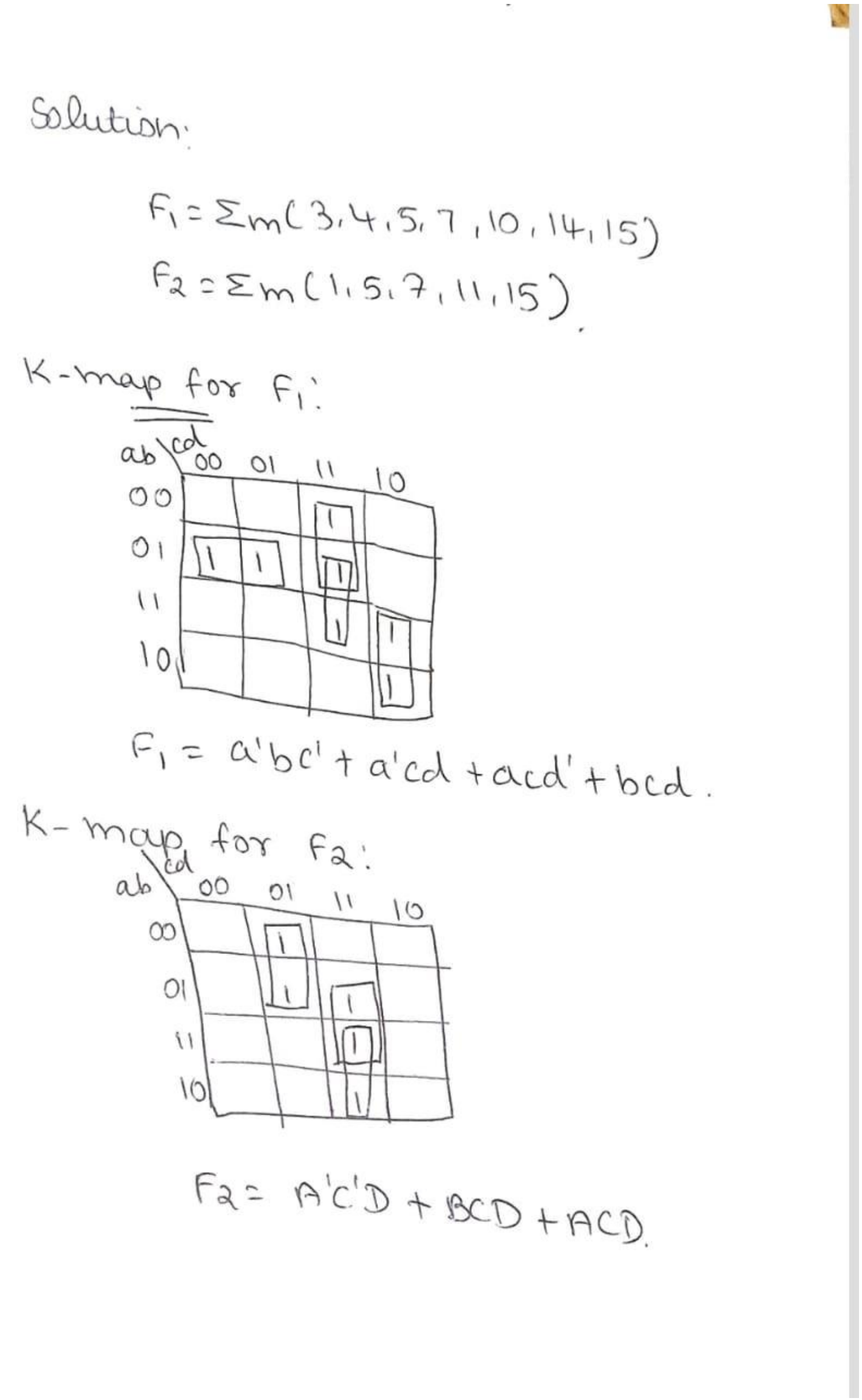
$K_0 = \bar{Q}_3 + Q_3 \Rightarrow K_0 = 1$



BCD counter using JK-FF.

UNIT 4

17.combinational logic is defined by the functions F=£(3,4,5,7,10,14,15) AND £(1,5,7,11,15).IMPLEMENT the circuit using a programmable logic array with 4 inputs,6 product terms and 2 outputs.



Solution:

$F_1 = \Sigma m(3,4,5,7,10,14,15)$

$F_2 = \Sigma m(1,5,7,11,15)$.

K-map for F_1:

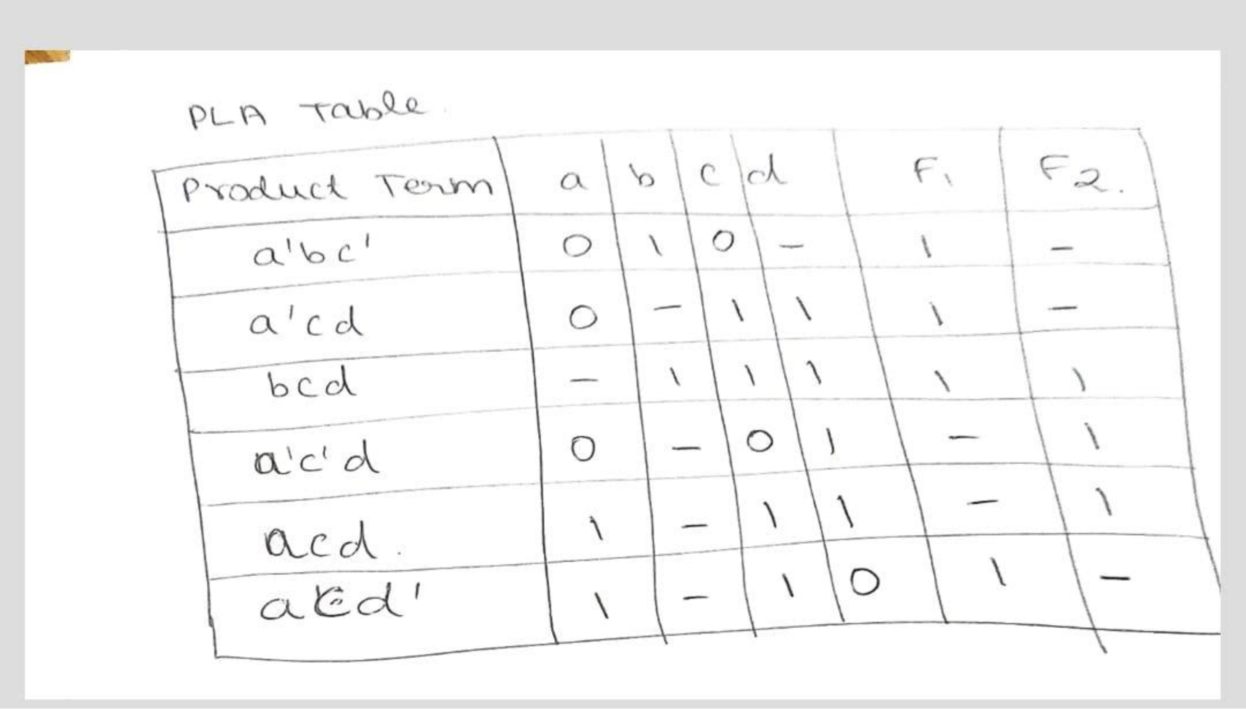
ab\cd	00	01	11	10
00			1	
01	1	1	1	
11			1	1
10				1

$F_1 = a'bc' + a'cd + acd' + bcd$.

K-map for F_2:

ab\cd	00	01	11	10
00		1		
01		1	1	
11			1	
10			1	

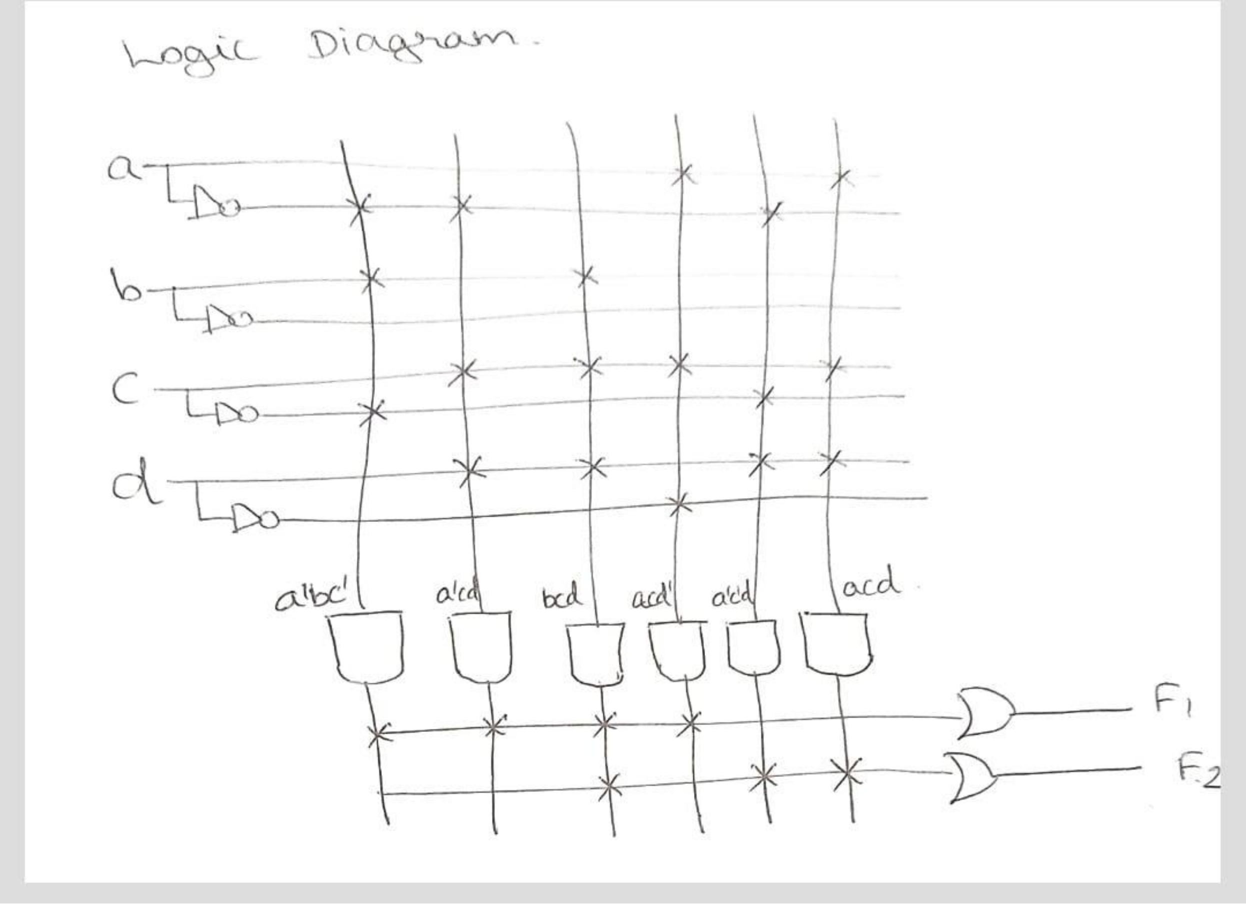
$F_2 = A'C'D + BCD + ACD$.



PLA Table.

Product Term	a	b	c	d	F₁	F₂.
a'bc'	0	1	0	–	1	–
a'cd	0	–	1	1	1	–
bcd	–	1	1	1	1	1
a'c'd	0	–	0	1	–	1
acd.	1	–	1	1	–	1
acd'	1	–	1	0	1	–

Logic Diagram.



OR

18.Tabulate the PAL programming table for the four Boolean functions listed below.

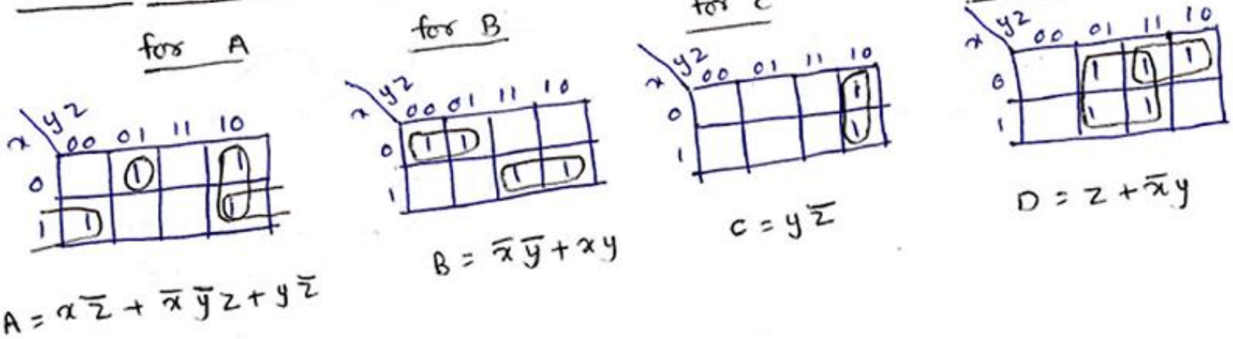
A(x,y,z)=£(1,2,4,6)

B(x,y,z)= £(0,1,6,7)

C(x,y,z)=£(2,6)

D(x,y,z)= £ (1,2,3,5,7).simplify using K-map.

K-map simplification



$A = x\bar{z} + \bar{x}\bar{y}z + y\bar{z}$

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

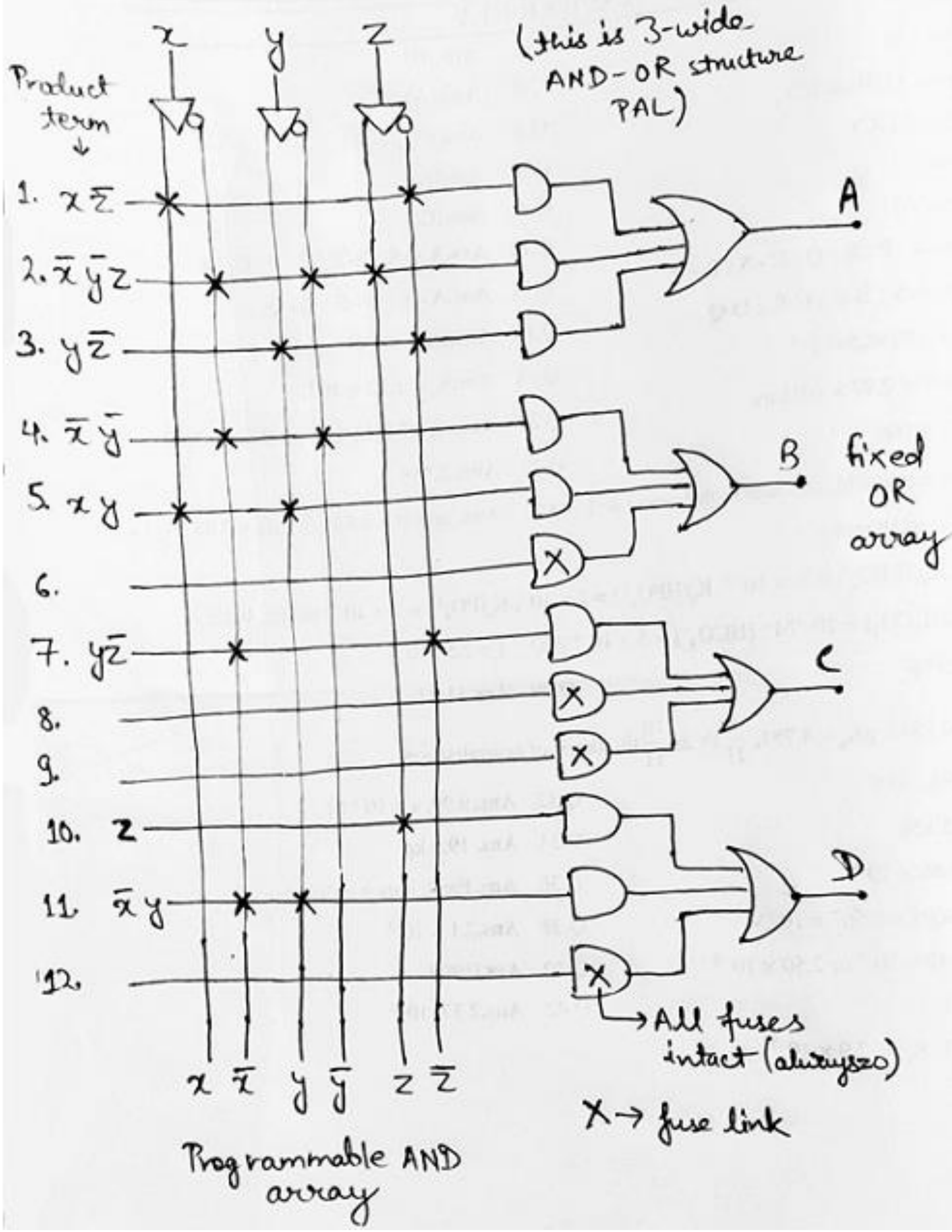
$C = y\bar{z}$

$D = z + \bar{x}y$

PAL Program Table

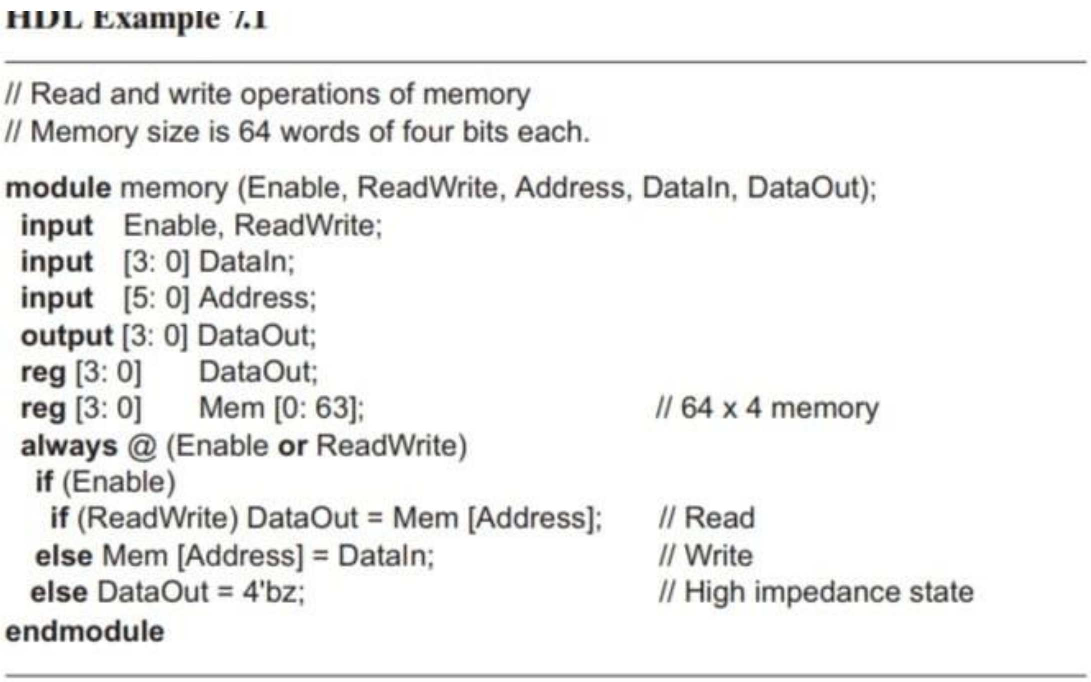
	Product terms	AND inputs x	y	z	outputs
1	$x\bar{z}$	1	—	0	$A = x\bar{z} + \bar{x}\bar{y}z + y\bar{z}$
2	$\bar{x}\bar{y}z$	0	0	1	
3	$y\bar{z}$	—	1	0	
4	$\bar{x}\bar{y}$	0	0	—	$B = \bar{x}\bar{y} + xy$
5	xy	1	1	—	
6	—	—	—	—	
7 8 9	$y\bar{z}$	—	1	0	$C = y\bar{z}$
10	z	—	—	1	$D = z + \bar{x}y$
11	$\bar{x}y$	0	1	—	
12	—	—	—	—	

Logic Diagram :-



UNIT 5

19.Interpret a Verilog HDL code to perform read and write operation of memory.



HDL Example 7.1

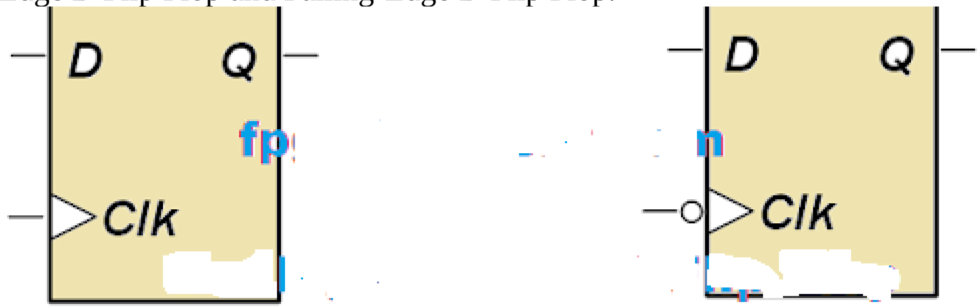
```
// Read and write operations of memory
// Memory size is 64 words of four bits each.

module memory (Enable, ReadWrite, Address, DataIn, DataOut);
  input   Enable, ReadWrite;
  input   [3: 0] DataIn;
  input   [5: 0] Address;
  output [3: 0] DataOut;
  reg [3: 0]      DataOut;
  reg [3: 0]      Mem [0: 63];                    // 64 x 4 memory
  always @ (Enable or ReadWrite)
   if (Enable)
    if (ReadWrite) DataOut = Mem [Address];       // Read
   else Mem [Address] = DataIn;                   // Write
   else DataOut = 4'bz;                           // High impedance state
endmodule
```

OR

20.Design Verilog module for an edge triggered D Flip flop in the data flow model.

D Flip-Flop is a fundamental component in digital logic circuits. Verilog code for D Flip Flop is presented in this project. There are two types of D Flip-Flops being implemented which are Rising-Edge D Flip Flop and Falling-Edge D Flip Flop.



Rising-Edge D Flip-Flop   Falling-Edge D Flip-Flop

```
module RisingEdge_DFlipFlop(D,clk,Q);
input D; // Data input
input clk; // clock input
output Q; // output Q
```

```
always @(posedge clk)
begin
 Q <= D;
end
endmodule


module FallingEdge_DFlipFlop(D,clk,Q);
input D; // Data input
input clk; // clock input
output reg Q; // output Q
always @(negedge clk)
begin
 Q <= D;
end
endmodule
```