

Question No. 2(a)

computer system level hierarchy is combination of majorly 7 levels.

level 0 Digital logic

level 1 Control

level 2 Machine

level 3 System software

level 4 Assembly language

level 5 High level language

level 6 User

level 0  $\Rightarrow$  Digital Logic

It is related to digital logic. It is the basic for digital computing and provides a fundamental understanding of how circuits and hardware communicate within a computer. It consists of various logical circuits and gates etc.

level 1  $\Rightarrow$  Control

It is the level where microcode is used in the system. control units are included in this level of the computer system.

level-2  $\Rightarrow$  Machines.

Different types of hardware are used in the computer system to perform different types of activities. It contains instruction set architecture.

level-3  $\Rightarrow$  System Software

These softwares mainly helps in operating the process and it establishes the connection between hardware and user interface.  
It consists of operating system.

level-4  $\Rightarrow$  Assembly language

Machine ~~only~~ understands only the assembly language and hence all high level languages are changed to assembly language.

level-5  $\Rightarrow$  High level language

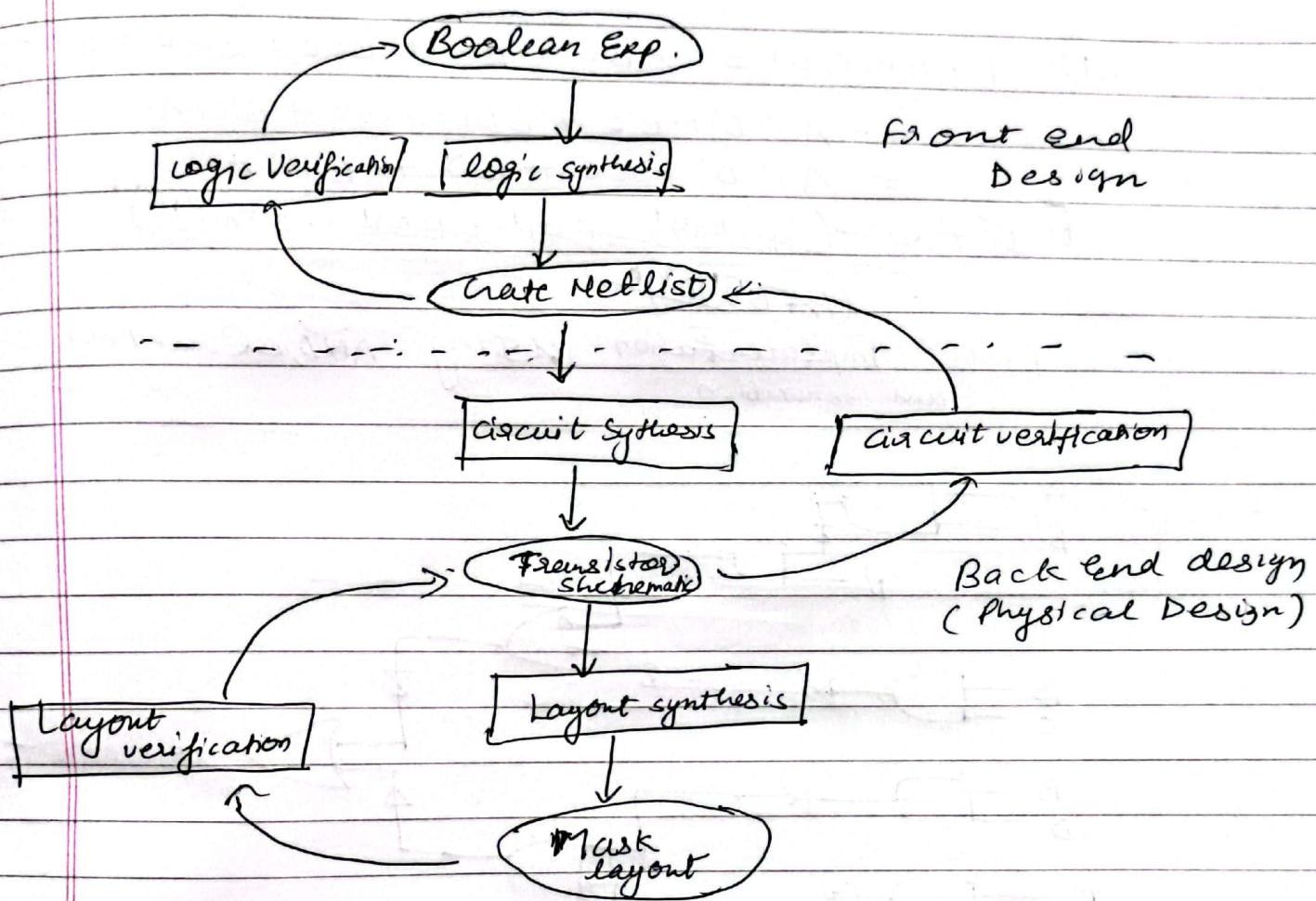
High level languages are the ones in which user gives the command i.e. C++, Java, etc.

Ø

level-6  $\Rightarrow$  user

This consists of user and executable programs.

## Question No. 1 (c)



In Front end design we create gate netlist using logic of Boolean expression. Then we do logic verification for the gate Netlist.

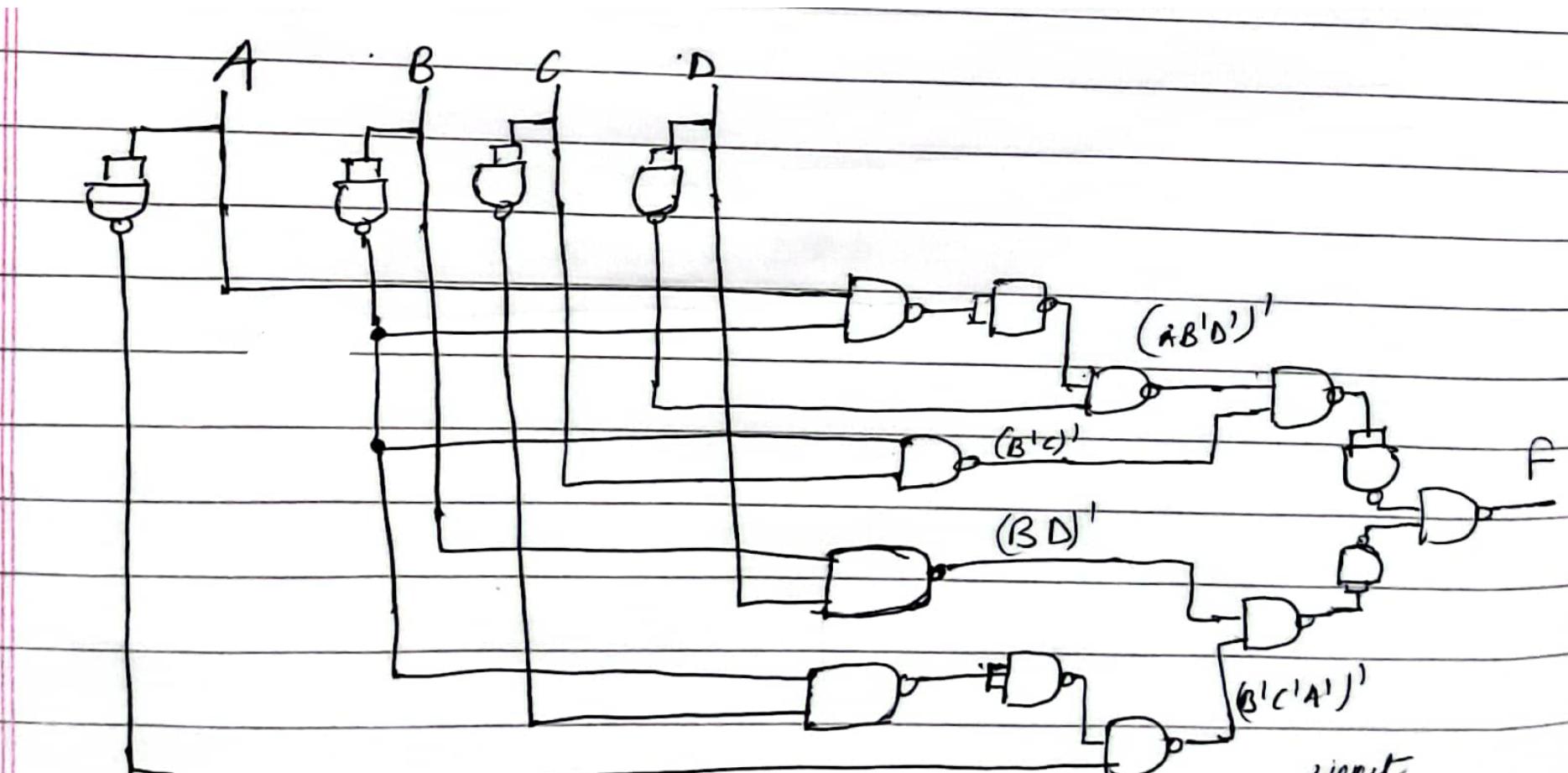
After that we got circuit synthesis in which we create circuit in which gates are implemented using transistors. Again in circuit verification we check transistor implementation is same as the gate netlist.

In layout synthesis we implement the transistor schematic on chip. And again we do layout verification in which we test for implementation of transistor schematic.

## Question No. 2

$$\begin{aligned}(a) \quad f(A, B, C, D) &= AB'D' + B'C + BCD + BC'D + B'C'A' \\ &= AB'D' + B'C + BD(C+C') + B'C'A' \\ &= AB'D' + B'C + BD + B'C'A'\end{aligned}$$

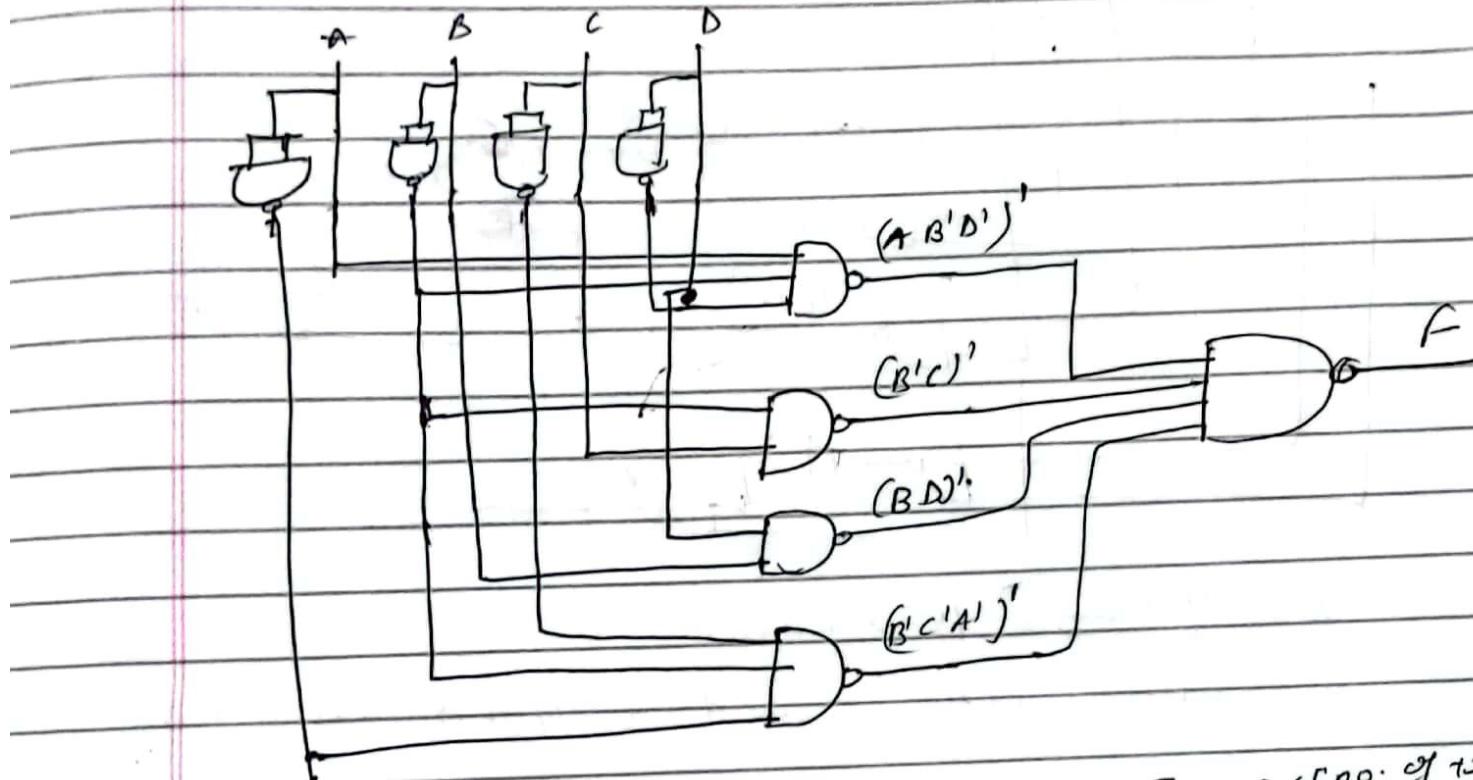
Now we use NAND alternative of two input AND or OR and NOT and extra NOT gates.



$$\text{No. of transistors} = 17 \times 4 = 68$$

<sup>input</sup>  
 :- 1<sup>st</sup> AND layer  
 4 - transistors

for optimizing the circuit  
we can use three input & 4 input NAND

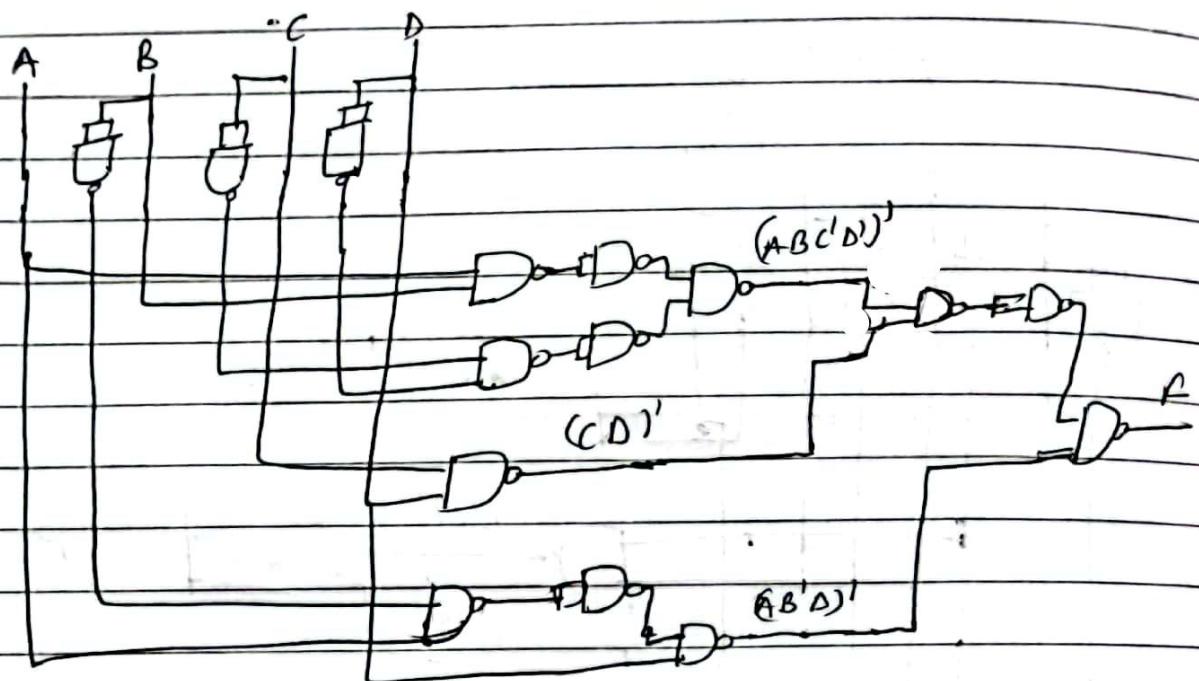


$$\text{No. of transistors} = 6 \times [\text{no. of transistors in 4 input NAND}] + 2 \times [\text{no. of transistors in 2 input AND}] + 1 \times [\text{no. of transistors in 2 input AND}]$$

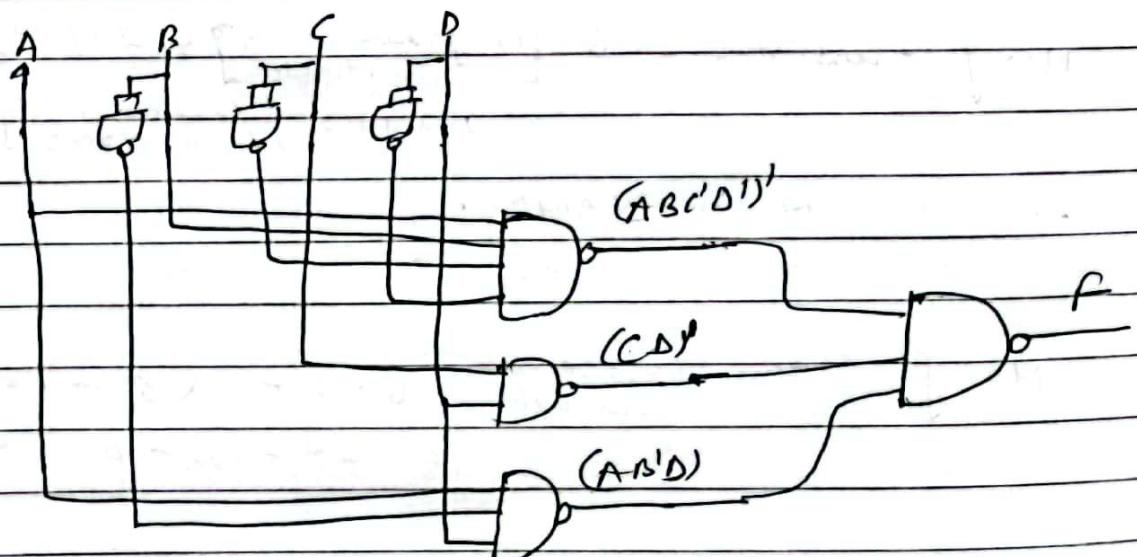
$\therefore$  9n N in NAND  
there are 24 transistors.

$$\begin{aligned}\text{No. of transistors in circuits} &= 6 \times 9 + 2 \times 6 + 1 \times 8 \\ &= 27 + 12 + 8 = 47 \text{ transistors}\end{aligned}$$

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



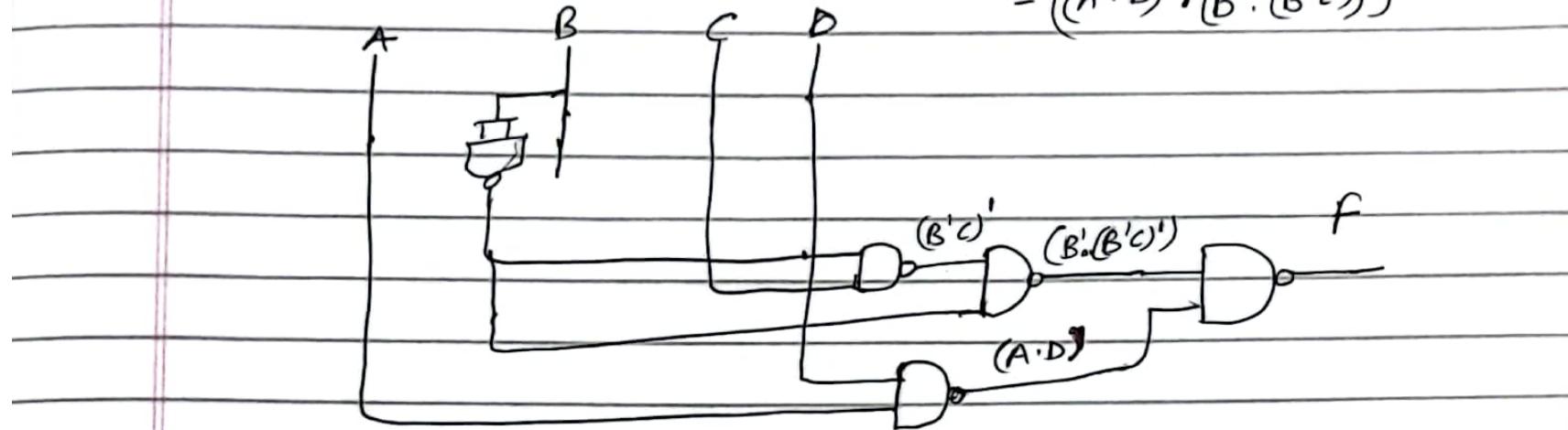
No. of terms =  $15 \times 4 = 60$   
 per optimization used 3 input, 4 input NAND



$$\text{No. of terms} = 4 \times 4 + 6 \times 2 + 8 \times 1 \\ = 16 + 12 + 8 = 36$$

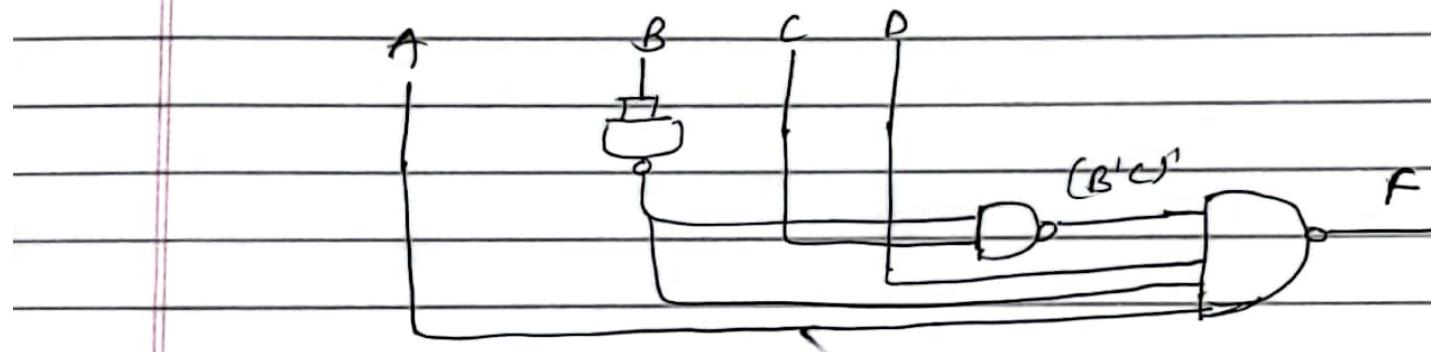
$$(4) \quad F(A, B, C, D) = A' + B + D' + B'C = (A \cdot D)' + (B \cdot (B'C))'$$

$$= ((A \cdot D) \cdot (B \cdot (B'C)))'$$



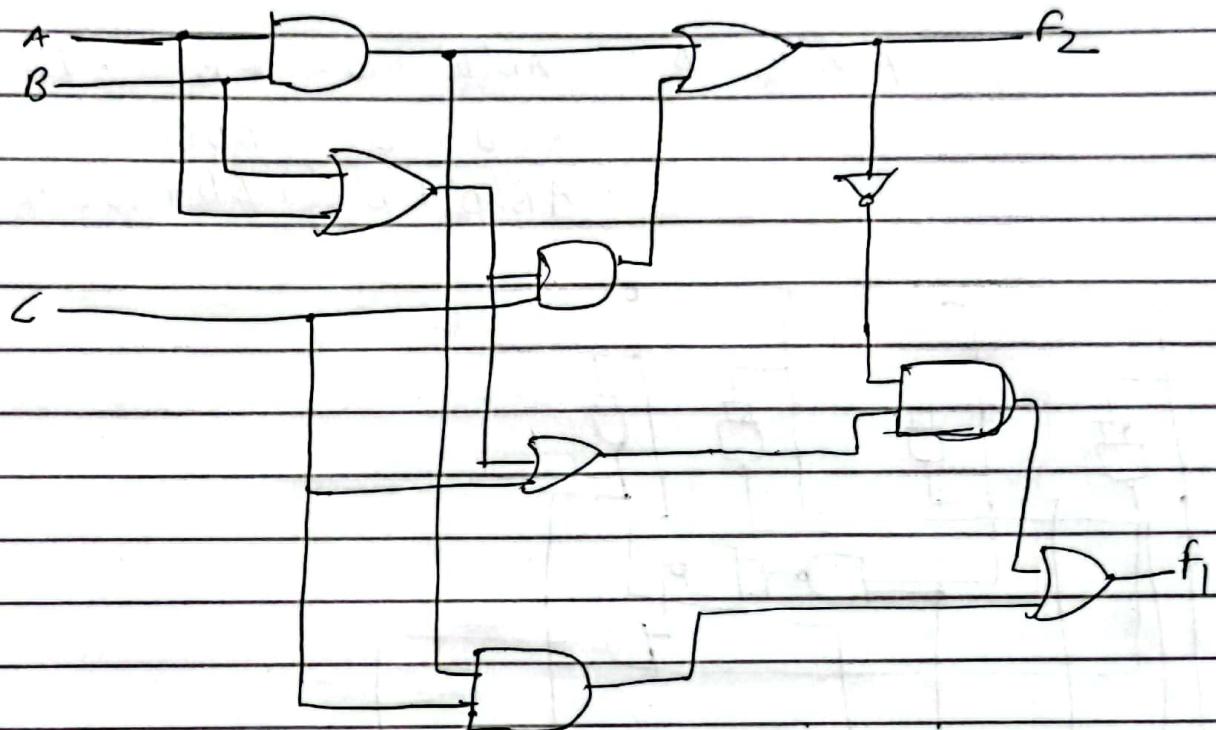
No. of transistors -:  $5 \times 2 = 20$

for optimizing, use four input NAND



No. of transistors  $\Rightarrow 4 \times 2 + 8 \times 1 = 16$

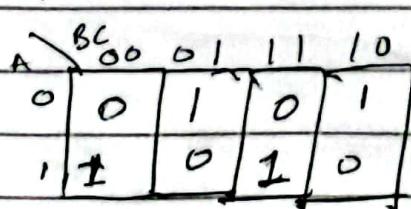
question no. 3



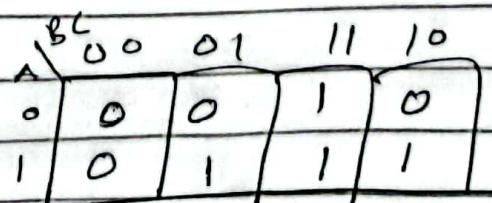
Truth Table

A	B	C	$F_1$	$F_2$
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

OK-map for  $f_1$



K-map  $f_2$



$$\Rightarrow f_1 = A \oplus B \oplus C$$

$$f_2 = BC + AC + AB$$

Question No. 5  
function table of JK flip-flop.

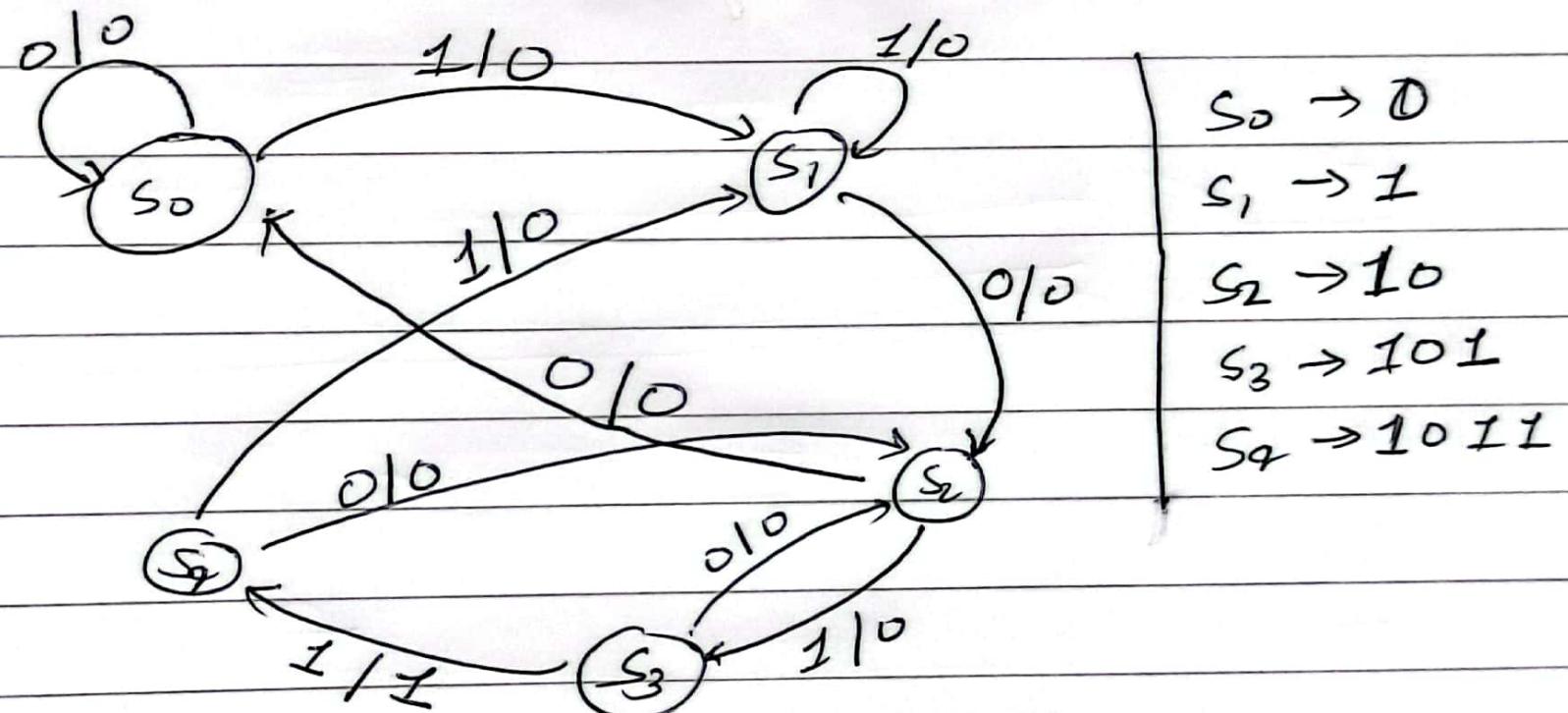
J	K	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$\bar{Q}(t)$

$Q(t)$	J	K	00	01	11	10
0	0	0	0	0	1	1
1	1	1	1	0	0	1

$$Q(t+1) = J(Q(t))' + K' Q(t)$$

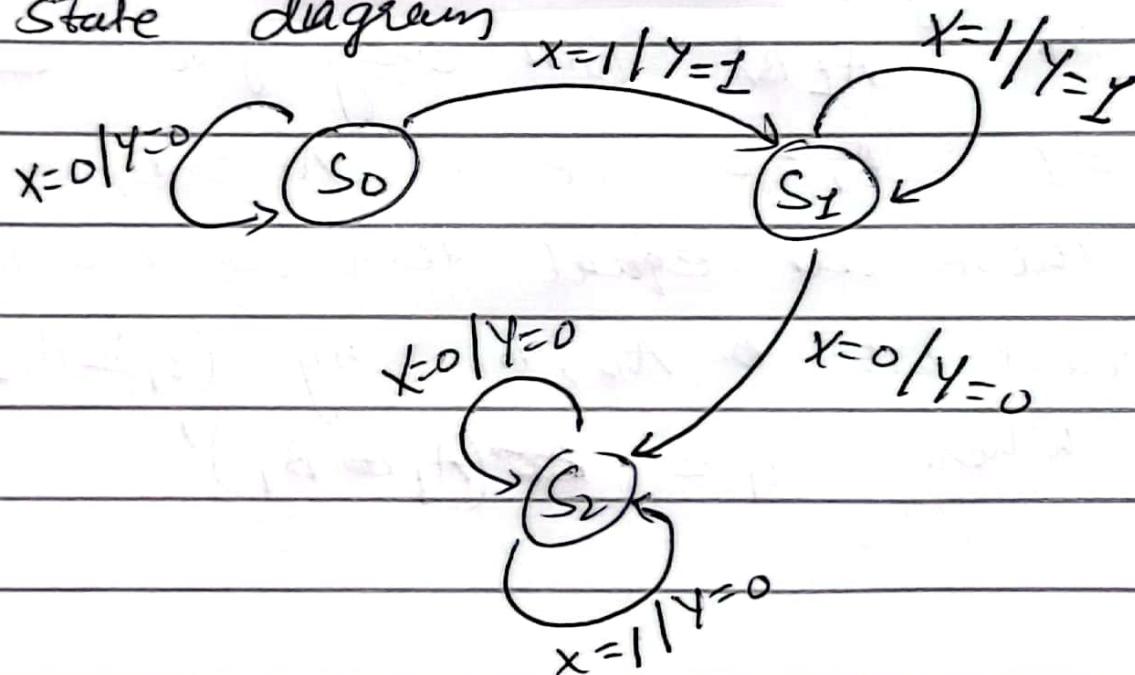
## Question No. 6

$s_0 \rightarrow$  state when it starts receiving bits



Question No. 7

State diagram



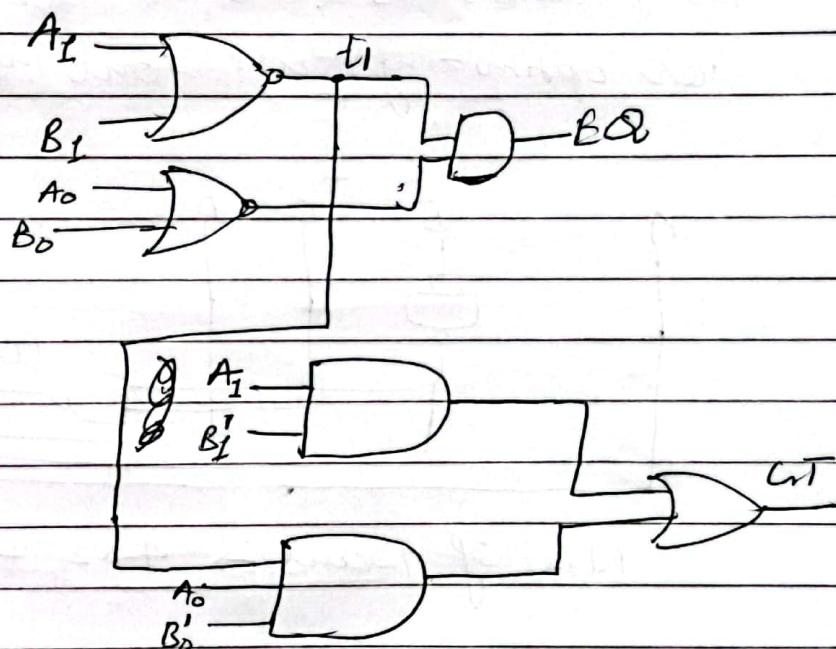
### Question No. 8

let  $A = A_1 A_0$        $B = B_1 B_0$

If MSB of  $A$  is greater than MSB of  $B$   
then  $A > B$ .

else if  $A_1 = B_1$  we go to check  $A_0 \& B_0$ .

hence to check equality we use XNOR gate



for checking greater than we use AND gate as  $A_1 B_1$  will only give 1 when

$$A_1 = 1, B_1 = 0 \text{ i.e. } \Rightarrow A_1 > B_1$$

when these are equal then we check next

significant bit  $\&$   $A_0, B_0$  by  $(t_1 \& A_0 \& B_0')$

$$\text{where } t_1 = (A_1 \oplus B_1)'$$

Question No. 9

Implementing function  $\pi(0, 1, 2, 4, 5, 6, 7, 11, 12, 13, 15)$

Let the four inputs as  $A, B, C, D$  and the function be  $F$

Truth Table

$A$	$B$	$C$	$D$	$F$
0	0	0	0	1 ] $\rightarrow 1$
0	0	0	1	1 ] $\rightarrow 0$
0	0	1	0	1 ] $\rightarrow 0$
0	0	1	1	0 ] $\rightarrow 0$
0	1	0	0	1 ] $\rightarrow 1$
0	1	0	1	1 ] $\rightarrow 1$
0	1	1	0	1 ] $\rightarrow 1$
0	1	1	1	1 ] $\rightarrow 1$
1	0	0	0	0 ] $\rightarrow 0$
1	0	0	1	0 ] $\rightarrow 0$
1	0	1	0	0 ] $\rightarrow 0$
1	0	1	1	1 ] $\rightarrow 0$
1	1	0	0	1 ] $\rightarrow 1$
1	1	0	1	1 ] $\rightarrow 1$
1	1	1	0	0 ] $\rightarrow 0$
1	1	1	1	1 ] $\rightarrow 0$

for implementing it using  $8 \times 1$  MUX take  $A, B, C$  as select line and  $D$  as input

