



**easy-solutions**

**Mumbai University Paper Solutions**

Strictly as per the New Revised Syllabus (Rev - 2016) of  
**Mumbai University** w.e.f. academic year 2017-2018  
(As per Choice Based Credit and Grading System)



# Digital Circuit Design

Semester III - Electronics Engineering

**Chapterwise Paper Solution upto May 2019.**



**Tech Knowledge**  
Publications

## INDEX

- Chapter 1** : Number Systems and Codes
- Chapter 2** : Logic Gates and Boolean Algebra
- Chapter 3** : Logic Minimization and Reduction Techniques
- Chapter 4** : Arithmetic Circuits
- Chapter 5** : Multiplexer & Demultiplexer
- Chapter 6** : Logic Families
- Chapter 7** : Latches and Flip Flops
- Chapter 8** : Shift Registers
- Chapter 9** : Counters

## Table of Contents

• <b>Index</b>	
• <b>Syllabus</b>	
• <b>Dec. 2017</b>	<b>D(17)-1 to D(17)-19</b>
• <b>May 2018</b>	<b>M(18)-1 to M(18)-19</b>
• <b>Dec. 2018</b>	<b>D(18)-1 to D(18)-19</b>
• <b>May 2019</b>	<b>M(19)-1 to M(19)-13</b>
• <b>Question Papers</b>	<b>Q-1 to Q-4</b>

□□□

# **Syllabus**

## **Module 1**

### **Number Systems and Codes :**

Review of Number System, Binary Code, Binary Coded Decimal, Octal Code, Hexadecimal Code and their conversions, Binary Arithmetic: One's and two's complements, Excess-3 Code, Gray Code, Weighted code, Parity Code : Hamming Code.

## **Module 2**

### **Logic Gates and Boolean Algebra :**

Digital logic gates, Realization using NAND, NOR gates, Boolean Algebra, De Morgan's Theorem, SOP and POS representation, K Map up to four variables and Quine-McClusky method upto four variables.

## **Module 3**

### **Combinational Logic Circuits and Hazards :**

Arithmetic Circuits : Adders/Subtractors:Half adder, Full adder, Half Subtractor, Full Subtractor, Ripple carry adder, Carry Look ahead adder and BCD adder, Magnitude Comparator.

### **Multiplexer and De-multiplexer :**

Multiplexer, cascading of Multiplexer, Boolean Function implementation using single multiplexer and basic gates, De-multiplexer, encoder and decoder, Parity Circuits, ALU. Hazards : Timing hazards static and dynamic.

## **Module 4**

### **Logic Families :**

Basics of standard TTL (Two input NAND gate operation), CMOS (Inverter, Two input NAND gate, Two input NOR gate), Interfacing of TTL to CMOS and CMOS to TTL, ECL, Working and characteristics of logic families.

## **Module 5**

### **Sequential Logic Principles :**

Latches and Flip flops : Difference between latches and flip flops, RS, JK, Master slave flip flops, T and D flip flops with various triggering methods, Conversion of flip flops, Applications of latches and flip flops in switch debouncing, bus holder circuits, Flip flops timing considerations and Metastability.

## **Module 6**

### **Counters and Registers :**

Asynchronous and Synchronous, Up/Down, Johnson Counter, MOD N, BCD counter using Decade counter, Ring counters, Shift registers, Universal Shift Register.

□□□



## Digital Circuit Design

### Statistical Analysis

Chapter No.	Dec. 2017	May 2018	Dec. 2018	May 2019
Chapter 1	10 Marks	05 Marks	15 Marks	15 Marks
Chapter 2	15 Marks	-	-	10 Marks
Chapter 3	10 Marks	10 Marks	20 Marks	15 Marks
Chapter 4	20 Marks	20 Marks	15 Marks	15 Marks
Chapter 5	20 Marks	30 Marks	15 Marks	10 Marks
Chapter 6	10 Marks	15 Marks	15 Marks	05 Marks
Chapter 7	05 Marks	15 Marks	20 Marks	25 Marks
Chapter 8	10 Marks	-	-	10 Marks
Chapter 9	20 Marks	25 Marks	20 Marks	20 Marks
-	-	-	-	20 Marks

**Dec. 2017**

### Chapter 1 : Number Systems and Codes [Total Marks : 10]

**Q. 1(a) Explain the following decimals in gray code form.**

1.  $(42)_{10}$       2.  $(17)_{10}$

(5 Marks)

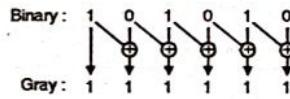
**Ans. :**

1.  $(42)_{10}$  to gray :

Step 1 : Convert  $(42)_{10}$  to binary :

$$(42)_{10} = (101010)_2$$

Step 2 : Binary to gray : (C-3048)



$$\therefore (42)_{10} = (111111)_{\text{gray}}$$

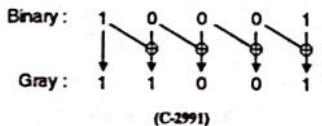
...Ans.

2.  $(17)_{10}$  to gray :

Step 1 : Convert  $(17)_{10}$  to binary :

$$(17)_{10} = (10001)_2$$

Step 2 : Binary to gray :



$$(17)_{10} = (11001)_{\text{gray}}$$

...Ans.



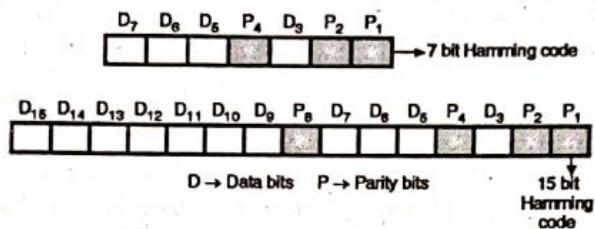
Q. 6(b) Write short note : Hamming code.

(5 Marks)

Ans. :

Hamming code :

Hamming code is basically a linear block code named after its inventor. It is an error correcting code. The parity bits are inserted in between the data bits as shown in Fig. 1- Q. 6(b). The 7-bit Hamming code is used commonly, but the concept can be extended to any number of bits.

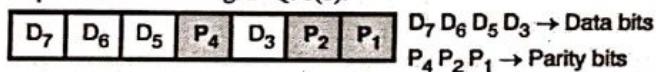


(G-1946) Fig. 1- Q. 6(b) : Hamming code words

Note that the parity bits are inserted at each  $2^n$  bit where  $n = 0, 1, 2, 3, \dots$ . Thus,  $P_1$  is at  $2^0 = 1$ , i.e. at first bit,  $P_2$  is at  $2^1 = 2$ ,  $P_4$  is at  $2^2 = 4$  and  $P_8$  is at  $2^3 = 8$  as shown in Fig. 1- Q. 6(b).

Bit Hamming Code :

A scientist named R.W. Hamming developed a coding system which was easy to implement. Assuming that four data bits are to be transmitted, he suggested a code word pattern shown in Fig. 2- Q. 6(b).



(G-1947) Fig. 2- Q. 6(b) : Code word pattern for Hamming code

The D bits in Fig. 2- Q. 6(b) are data bits, whereas P bits are parity bits. The parity bits  $P_1, P_2, P_4$  are adjusted in a particular way.

Minimum number of parity bits :

Following table gives a listing of minimum number of parity bits needed for various ranges of "m" information bits.

Number of parity bits to be used	
Number of information bits	Number of parity bits
2 to 4	3
5 to 11	4
12 to 26	5
27 to 57	6
58 to 120	7

## Chapter 2 : Logic Gates and Boolean Algebra [Total Marks : 15]

Q. 1(c) State and prove Demorgan's theorem.

(5 Marks)

Ans. :

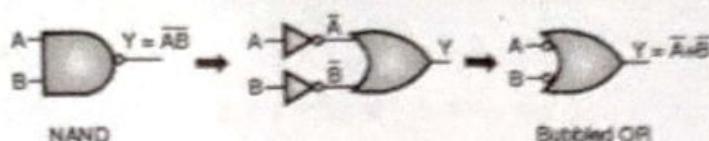
Demorgan's theorem :

Theorem 1 :  $\overline{AB} = \overline{A} + \overline{B}$  : NAND = Bubbled OR :

This theorem states that, the complement of a product is equal to addition of the complements. This rule is illustrated in Fig. 1- Q. 1(c). The Left Hand Side (LHS) of this theorem represents a NAND gate with inputs A and B whereas the Right Hand Side

(RHS) of the theorem represents an OR gate with inverted inputs. Such an OR gate is called as "Bubbled OR". Thus we can state De-Morgan's first theorem as a NAND operation is equivalent to a bubbled OR operation.

NAND = Bubbled OR.



(B-412) Fig. 1- Q. 1(c) : Illustration of De-Morgan's first theorem

This theorem can be verified by writing a truth table as shown in Fig. 2- Q. 1(c).

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

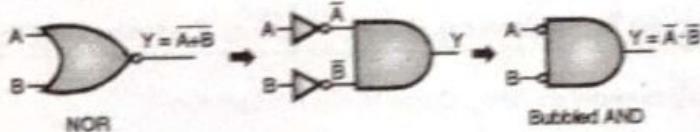
LHS  $\bar{AB} = \bar{A} + \bar{B}$  RHS

(C-413) Fig. 2- Q. 1(c) : Verification of the theorem  $\bar{AB} = \bar{A} + \bar{B}$

Theorem 2 :  $\overline{A + B} = \bar{A} \cdot \bar{B}$  : NOR = Bubbled AND :

This theorem is illustrated in Fig. 3- Q. 1(c). The LHS of this theorem represents a NOR gate with inputs A and B whereas the RHS represents an AND gate with inverted inputs. Such an AND gate is called as "Bubbled AND". Thus we can state De-Morgan's second theorem as a NOR function is equivalent to a bubbled AND function.

NOR = Bubbled AND



(B-414) Fig. 3- Q. 1(c) : Illustration of De-Morgan's second theorem

This theorem can be verified by writing a truth table for both the sides of the theorem statement. This truth table is shown in Fig. 4- Q. 1(c), which shows that LHS = RHS.

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

LHS  $\bar{A} + \bar{B} = \bar{A} \cdot \bar{B}$  RHS

(C-415) Fig. 4- Q. 1(c) : Truth table to verify De-Morgan's theorem

Q. 4(b) Using Boolean algebra and De-Morgan's theorem prove that

$$\bar{Y}\bar{Z} + \bar{W}\bar{X}\bar{Z} + \bar{W}XY\bar{Z} + WYZ\bar{Z} = \bar{Z}$$

Simplify the expression  $[\bar{A}\bar{B}(C + BD) + \bar{A}\bar{B}]C$  as much as possible.

(10 Marks)



Ans. :

$$1. \text{ LHS} = \bar{Y}\bar{Z} + \bar{W}\bar{X}\bar{Z} + \bar{W}XY\bar{Z} + WY\bar{Z}$$

$$\begin{aligned}
 &= \bar{Y}\bar{Z} + \bar{W}\bar{Z}(\bar{X} + XY) + WY\bar{Z} \\
 &= \bar{Y}\bar{Z} + \bar{W}\bar{Z}(\bar{X} + Y) + WY\bar{Z} \quad \dots (\because \bar{X} + XY = \bar{X} + Y) \\
 &= \bar{Y}\bar{Z} + \bar{X}\bar{W}\bar{Z} + \bar{W}Y\bar{Z} + WY\bar{Z} \\
 &= \bar{Y}\bar{Z} + \bar{X}\bar{W}\bar{Z} + Y\bar{Z}(\bar{W} + W) \\
 &= \bar{Y}\bar{Z} + \bar{X}\bar{W}\bar{Z} + Y\bar{Z} \quad \dots (\because W + \bar{W} = 1) \\
 &= \bar{Z}(Y + \bar{Y}) + \bar{X}\bar{W}\bar{Z} \\
 &= \bar{Z}(1 + \bar{X}\bar{W}) \quad \dots (\because Y + \bar{Y} = 1) \\
 &= \bar{Z} \quad \dots (\because 1 + \bar{X}\bar{W} = 1) \\
 &= \text{Proved}
 \end{aligned}$$

$$2. \text{ Y} = [A\bar{B}(C + BD) + \bar{A}\bar{B}]C$$

$$\begin{aligned}
 &= [\bar{A}\bar{B}C + A\bar{B}BD + \bar{A}\bar{B}]C \\
 &= [\bar{A}\bar{B}C + 0 + \bar{A}\bar{B}]C \quad \dots (\because BB = 0) \\
 &= \bar{A}\bar{B}CC + \bar{A}\bar{B}C \\
 &= \bar{A}\bar{B}C + \bar{A}\bar{B}C \quad \dots (\because CC = C) \\
 &= \bar{B}C(A + \bar{A}) \\
 &= \bar{B}C \quad \dots (\because A + \bar{A} = 1)
 \end{aligned}$$

### Chapter 3 : Logic Minimization and Reduction Techniques [Total Marks : 10]

Q. 3(b) Minimize the following expression using Quine McClusky technique.

$$F(A, B, C, D) = \sum M(0, 1, 2, 3, 5, 7, 9, 11)$$

(10 Marks)

Ans. :

Step 1 : Group the minterms according to number of 1's :

(C-6457) Table 1- Q. 3(b)

Group	Minterm pair	Binary representation				Prime Implicants
		A	B	C	D	
0	0	0	0	0	0	✓
1	1	0	0	0	1	✓
	2	0	0	1	0	✓
2	3	0	0	1	1	✓
	5	0	1	0	1	✓
	9	1	0	0	1	✓
3	7	0	1	1	1	✓
	11	1	0	1	1	✓



**Step 2 : Group the minterms into groups of two :**

(C-6458) Table 2- Q. 3(b)

Group	Minterm pair	Binary representation				Prime implicants
		A	B	C	D	
0	0, 1	0	0	0	-	✓
	0, 2	0	0	-	0	✓
1	1, 3	0	0	-	1	✓
	1, 5	0	-	0	1	✓
	1, 9	-	0	0	1	✓
	2, 3	0	0	1	-	✓
2	3, 7	0	-	1	1	✓
	3, 11	-	0	1	1	✓
	5, 7	0	1	-	1	✓
	9, 11	1	0	-	1	✓

**Step 3 : Group the minterms into groups of four :**

(C-6459) Table 3- Q. 3(b)

Group	Quad of minterms	Binary representation				Prime implicants
		A	B	C	D	
0	0, 1, 2, 3	0	0	-	-	$\bar{A}\bar{B}$
	0, 2, 1, 3	0	0	-	-	
1	1, 3, 5, 7	0	-	-	1	$\bar{A}D$
	1, 5, 3, 7	0	-	-	1	
1	1, 3, 9, 11	-	0	-	1	$\bar{B}D$
	1, 9, 3, 11	-	0	-	1	

**Step 4 : Prepare the table of prime implicants :**

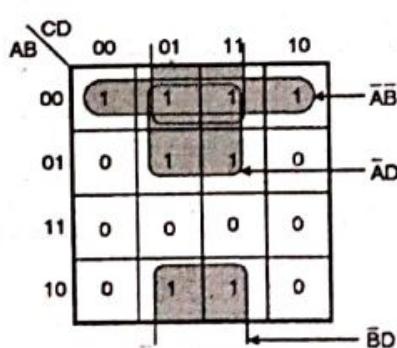
(C-6460) Table 4- Q. 3(b)

Prime implicants	Decimal numbers	Given minterms							
		0	1	2	3	5	7	9	11
$\bar{A}\bar{B}$	0, 1, 2, 3	(X)	X	(X)	X				
$\bar{A}D$	1, 3, 5, 7		X		X	(X)	(X)		
$\bar{B}D$	1, 3, 9, 11		X		X			(X)	(X)

The encircled crosses represent the EPIS. So the terms  $\bar{A}\bar{B}$ ,  $\bar{A}D$  and  $\bar{B}D$  are the EPIS which covers all the minterms.

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

**Step 5 : Crosscheck using K-map :**



(C-6461) Fig. 1- Q. 3(b) : K-map



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

Thus the results obtained by the two methods are exactly identical.

### Chapter 4 : Arithmetic Circuits [Total Marks : 20]

**Q. 4(a) Design a 2 bit comparator and implement using logic gates.**

(10 Marks)

**Ans. :**

**2 bit comparator :**

For a 2-bit comparator, each input word A and B is 2 bit long.

The truth table of a 2-bit comparator is shown in Table 1- Q. 4(a).

The K-maps for the three outputs and corresponding simplified expressions are shown in Figs. 1, 2 and 3- Q. 4(a). From these

K-maps we get the simplified expressions for the three outputs of comparator are as follows :

$$A < B = \bar{A}_1 \bar{A}_0 B_0 + \bar{A}_1 B_1 + \bar{A}_0 B_1 B_0$$

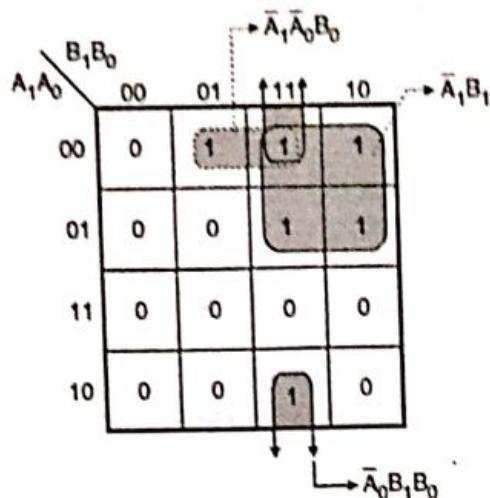
**Table 1- Q. 4(a) : Truth table for a 2-bit comparator**

Inputs				Outputs		
$A_1$	$A_0$	$B_1$	$B_0$	$A < B$	$A = B$	$A > B$
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

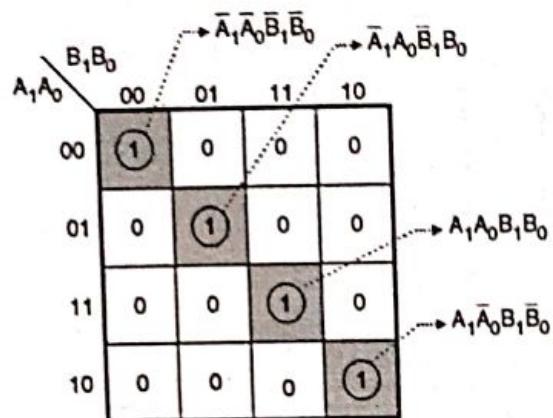


**K-maps :**

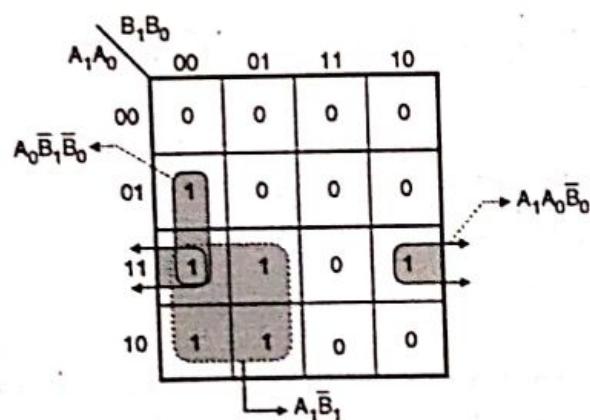
**For  $A < B$  :**

(C-406) Fig. 1- Q. 4(a) : K-map for output  $A < B$ 

**For  $A = B$  :**

(C-406) Fig. 2- Q. 4(a) : K-map for output  $A = B$ 

**For  $A > B$  :**

(C-407) Fig. 3- Q. 4(a) : K-map for output  $A > B$



**Simplified expression :**

$$A > B = A_0 \bar{B}_1 \bar{B}_0 + A_1 A_0 \bar{B}_0 + A_1 \bar{B}_1$$

**Simplification for output  $A = B$  :**

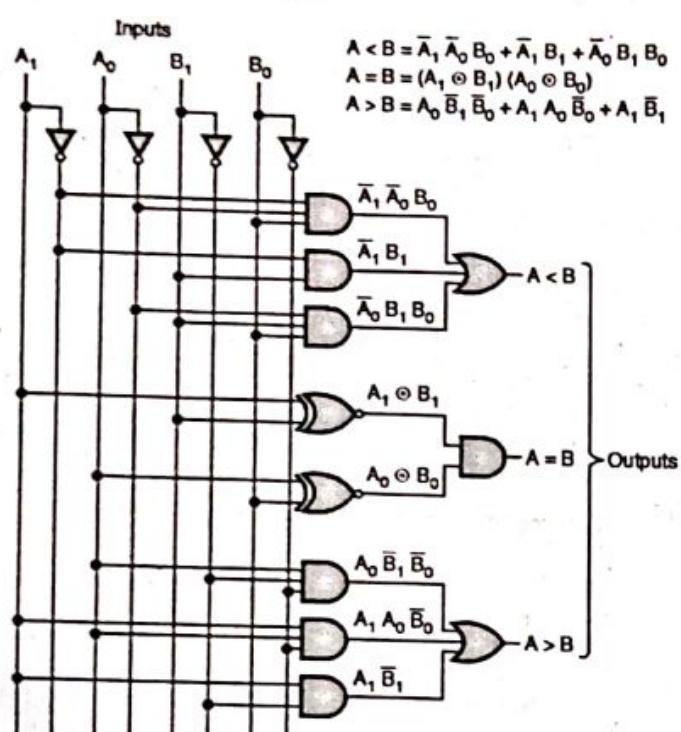
Refer the K-map for output  $A = B$  shown in Fig. 2- Q. 4(a).

The expression for  $A = B$  is given by,

$$\begin{aligned} (A = B) &= \bar{A}_1 \bar{A}_0 \bar{B}_1 \bar{B}_0 + \bar{A}_1 A_0 \bar{B}_1 B_0 \\ &\quad + A_1 A_0 B_1 B_0 + A_1 \bar{A}_0 B_1 \bar{B}_0 \\ &= \bar{A}_0 \bar{B}_0 (\bar{A}_1 \bar{B}_1 + A_1 B_1) + A_0 B_0 (\bar{A}_1 \bar{B}_1 + A_1 B_1) \\ &= (\bar{A}_1 \bar{B}_1 + A_1 B_1) (\bar{A}_0 \bar{B}_0 + A_0 B_0) \\ \therefore (A = B) &= (A_1 \oplus B_1) (A_0 \oplus B_0) \text{ where } \oplus = \text{EX-NOR} \end{aligned}$$

**Logic diagram for a two bit comparator :**

The logic diagram for the 2-bit digital comparator is shown in Fig. 4- Q. 4(a). This diagram is drawn by referring to the simplified expressions of the outputs.



(C-408) Fig. 4- Q. 4(a) : Logic diagram for 2-bit comparator

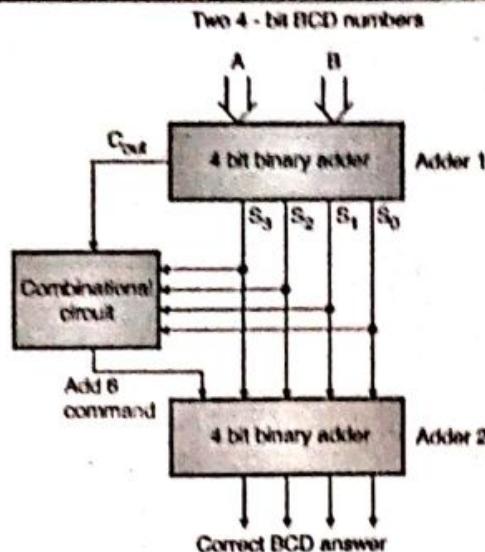
**Q. 5(b) Design BCD adder using the integrated circuit 4 bit binary adders.**

**(10 Marks)**

**Ans. :**

**BCD adder :**

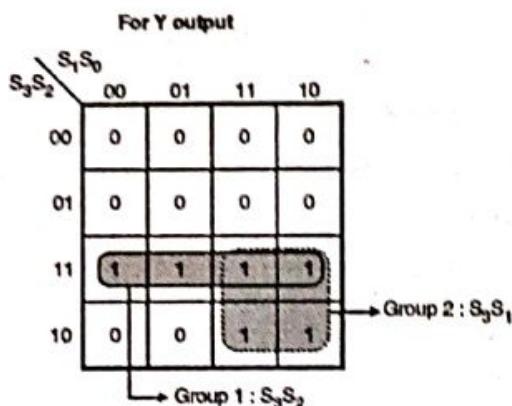
The block diagram of such a BCD adder is shown in Fig. 1- Q. 5(b). So we have to design the combinational circuit that finds out whether the sum is greater than 9 or carry = 1.



(C-397) Fig. 1- Q. 5(b) : Block diagram of BCD adder

**Design of Combinational Circuit :**

The output of combinational circuit should be 1 if the sum produced by adder 1 is greater than 9 i.e. 1001. The truth table is as follows :

**K-map :**

(C-398) Fig. 2- Q. 5(b) : K-map for Y output

(C-4180) Table 1- Q. 5(b) : Truth table for combinational circuit design

Sum bits of adder - 1 →	Inputs				Output Y
	$S_3$	$S_2$	$S_1$	$S_0$	
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	0
0	1	0	0	0	0
0	1	0	0	1	0
0	1	1	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	0	0	1	0
1	0	1	0	0	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	0	0	1	1
1	1	1	0	1	1
1	1	1	1	1	1

Sum is valid BCD number  
∴ Y = 0

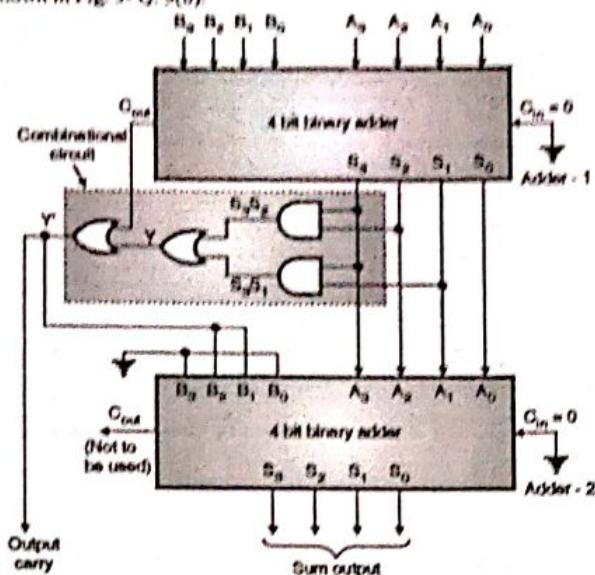
Sum is Invalid BCD number  
∴ Y = 1



The Boolean expression is,

$$Y = S_3S_2 + S_3S_1$$

The complete BCD adder is shown in Fig. 3- Q. 5(b).



(C-399) Fig. 3- Q. 5(b) : 4-bit BCD adder

The output of the combinational circuit should be 1 if  $C_{out}$  of adder-1 is high or if the output of adder-1 is greater than 9. Therefore  $Y$  is ORed with  $C_{out}$  of adder-1 as shown in Fig. 3- Q. 5(b). The output of combinational circuit is connected to  $B_1B_2$  inputs of adder-2 and  $B_3 = B_1 = 0$  as they are connected to ground permanently. This makes  $B_3B_2B_1B_0 = 0110$  if  $Y' = 1$ .

The sum outputs of adder-1 are applied to  $A_3A_2A_1A_0$  of adder-2. The output of combinational circuit is to be used as final output carry and the carry output of adder-2 is to be ignored.

### Chapter 5 : Multiplexer & Demultiplexer [Total Marks : 20]

Q. 2(b) Implement the following Boolean function using 8 : 1 multiplexer.

$$F(A, B, C, D) = \sum M(0, 1, 4, 5, 6, 8, 10, 12, 13)$$

(10 Marks)

Ans. :

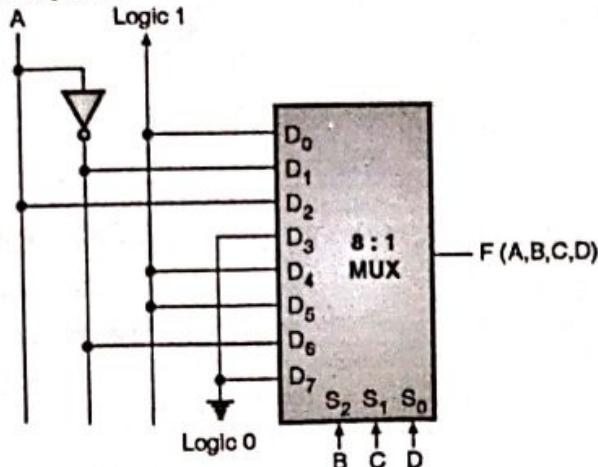
Step 1 : Write the design table :

(C-4455) Table 1- Q. 2(b) : Design table

Inputs	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>
$\bar{A}$	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15
Input to MUX	1	$\bar{A}$	A	0	1	1	$\bar{A}$	0



**Step 2 : Implementation using 8 : 1 multiplexer :**



(C-6456) Fig. 1- Q. 2(b) : Implementation using 8 : 1 MUX

**Q. 6 Write short note on**

(10 Marks)

- (a) Hazards.
- (c) Encoder and decoder.

**Ans. :**

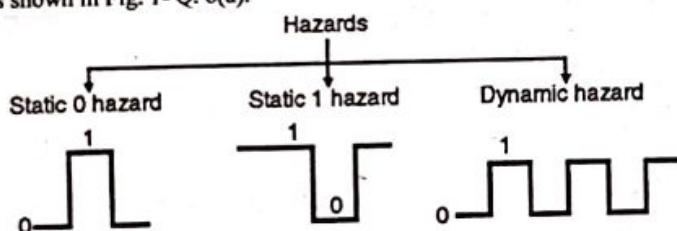
**(a) Hazards :**

The term hazards is used to specify the unwanted switching transients or false outputs or glitches which appear at the output of a circuit. These transient false outputs are produced due to finite propagation delay times of the components along different paths within the network. Difference between the propagation delays corresponding to different signal paths result in hazards.

We can define the hazard as the actual or potential malfunction of a logic network while making the transition between two input states when a single variable changes. Malfunctioning means anything other than the intended response. In the combinational circuits, the hazards will result in a false output value. But if such combinational circuits are used as a building block of an asynchronous sequential circuit then the hazards will result in a transition to incorrect stable state.

**Types of Hazards :**

The hazards are of 3-types as shown in Fig. 1- Q. 6(a).



(C-1270) Fig. 1- Q. 6(a) : Types of hazards

**Static 1 hazard :** If the output is supposed to be 1 (HIGH) but due to difference in propagation delay if it becomes 0 (LOW) for a short time, then a static 1 hazard is said to have taken place.

**Static 0 hazard :** A static 0 hazard is said to have taken place if the circuit output produces a short duration HIGH pulse when actually it is supposed to produce a LOW (0) output.



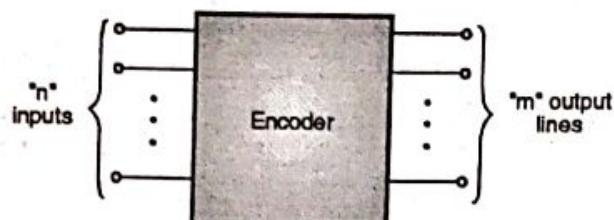
**Dynamic hazard :** In the static hazards the output changes twice. But as shown in Fig. 1-Q. 6(a). If the output changes for 3 or more times when it is supposed to change only once from 0 to 1 or from 1 to 0, then the dynamic hazard is said to have taken place.

**(c) Encoder and decoder :**

**Encoder :**

Encoder is a combinational circuit which is designed to perform the inverse operation of the decoder. An encoder has "n" number of input lines and "m" number of output lines.

An encoder produces an m bit binary code corresponding to the n bit digital number, applied at its input. Block diagram of an encoder is shown in Fig. 1-Q. 6(c).

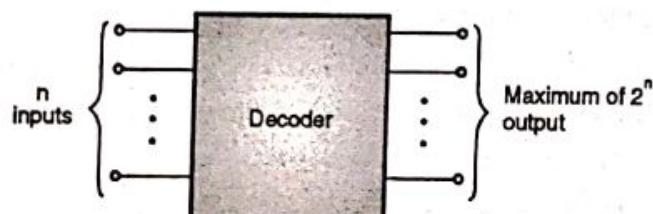


(C-466) Fig. 1-Q. 6(c) : Block diagram of an encoder

The encoder accepts an n input digital word and converts it into an m bit another digital word. For example a BCD number applied at the input can be converted into a binary number at the output. The internal combinational circuit of the encoder is designed accordingly.

**Decoder :**

A decoder is a combinational circuit. Fig. 2-Q. 6(c) shows the block diagram of a decoder. It has "n" inputs and to a maximum  $2^n$  outputs. Decoder is identical to a demultiplexer without any data input. It performs operations which are exactly opposite to those of an encoder.



(C-479) Fig. 2-Q. 6(c) : Block diagram of a decoder

## Chapter 6 : Logic Families [Total Marks : 10]

**Q. 1(b) Explain characteristics of logic families.**

**(5 Marks)**

**Ans. :**

**Fan-In :**

Fan-in is defined as the number of inputs a gate has. For example, a two input gate will have a fan-in equal to 2.

**Fan-out :**

Fan-out is defined as the maximum number of inputs of the same IC family that a gate can drive without falling outside the specified output voltage limits. Higher fan out indicates higher the current supplying capacity of a gate. For example, a fan out of 5 indicates that the gate can drive (supply current to) at the most 5 inputs of the same IC family.

**Noise Margin :**

Noise is an unwanted electrical disturbance which may induce some voltage in the connecting wires used between two gates or from a gate output to load.

Noise immunity is defined as the ability of a logic circuit to tolerate the noise without causing the output to change undesirably.

**Propagation Delay (Speed of Operation) :**

The output of a logic gate does not change its state instantaneously when the state of its input is changed. There is a time delay between these two time instants, which is called as the propagation delay. Thus propagation delay is defined as time delay between the instant of application of an input pulse and the instant of occurrence of the corresponding output pulse. Ideally propagation delay should be zero and practically it should be as short as possible.

**Power Dissipation :**

Due to applied voltage and currents flowing through the logic ICs, some power will be dissipated in it, in the form of heat. This power is in mill watts. Care should be taken to reduce the power dissipation taking place in the logic IC in order to protect the IC against damage due to excessive temperature, to reduce the loading on power supplies etc.

Another importance of power dissipation is that the product of power dissipation and propagation time is always constant. Therefore if we reduce the power dissipation may lead then the propagation delay will increase in order to keep their product constant.

**Q. 6(d) Compare TTL and CMOS logic families.**

(5 Marks)

**Ans. :****Table 1- Q. 6(d) : Comparison of CMOS and TTL**

Sr. No.	Parameter	CMOS	TTL
1.	Device used	N-channel MOSFET and P-channel MOSFET	Bipolar junction transistor
2.	$V_{IH}(\text{min})$	$3.5 \text{ V } (V_{DD} = 5 \text{ V})$	2 V
3.	$V_{IL}(\text{max})$	1.5 V	0.8 V
4.	$V_{OH}(\text{min})$	4.95 V	2.7 V
5.	$V_{OL}(\text{max})$	0.05 V	0.4 V
6.	High level noise margin	$V_{NH} = 1.45 \text{ V}$	0.4 V
7.	Low level noise margin	$V_{NL} = 1.45 \text{ V}$	0.4 V
8.	Noise immunity	Better than TTL	Less than CMOS
9.	Propagation delay	105 nS (Metal gate CMOS)	10 nS. (Standard TTL)
10.	Switching speed	Less than TTL.	Faster than CMOS.

**Chapter 7 : Latches and Flip Flops [Total Marks : 05]****Q. 1(d) Convert JK flip flop to T flip flop.**

(5 Marks)

**Ans. :****Conversion of JK flip flop to T flip flop:****Step 1 : Write the truth table for conversion :**

The required truth table is obtained from the excitation tables of JK and T flip flops as follows :

**(C-638) Table 1- Q. 1(d) : Truth table for conversion from JK to T FF**

T	Inputs		Outputs	
	Present state of Q	Next state of Q	J	K
0	0	0	0	X
1	0	1	1	X
1	1	0	X	1
0	1	1	X	0

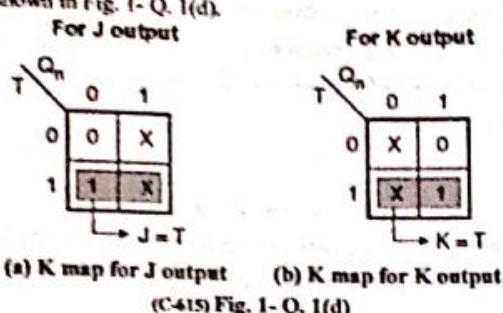
←———— Excitation table of T FF —————→

←———— Excitation table of JK FF —————→



## Step 2 : K maps and simplification :

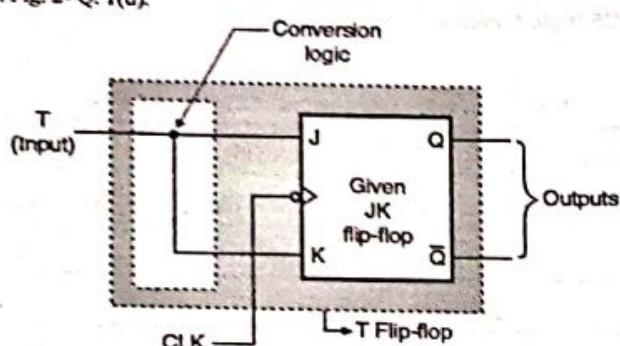
The K maps for outputs J and K are shown in Fig. 1- Q. 1(d).



(C-415) Fig. 1- Q. 1(d)

## Step 3 : Draw the logic diagram :

The logic diagram is shown in Fig. 2- Q. 1(d).



(C-416) Fig. 2- Q. 1(d) : Logic diagram for conversion of JK FF to T FF

Chapter 8 : Shift Registers [Total Marks : 10]

**Q. 2(a)** What is shift register ? Explain any one type of shift register. Give its applications. (10 Marks)

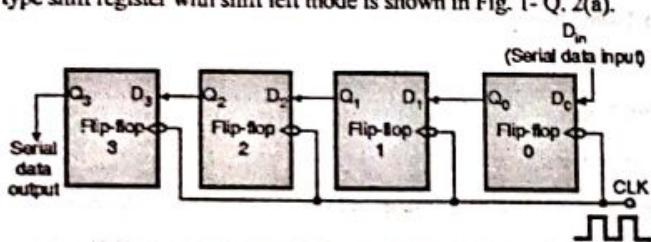
**Ans. :**

**Definition of Shift Registers :**

The binary data in a register can be moved within the register from one flip-flop to the other or outside it with application of clock pulses. The registers that allow such data transfers are called as shift registers.

**Serial Input Serial Output (Shift Left Mode) :**

The serial input serial output type shift register with shift left mode is shown in Fig. 1- Q. 2(a).



(C-732) Fig. 1- Q. 2(a) : Serial shift left register

Let all the flip-flops be initially in the reset condition i.e.  $Q_3 = Q_2 = Q_1 = Q_0 = 0$ . We are going to illustrate the entry of a four bit binary number 1 1 1 1 into the register.

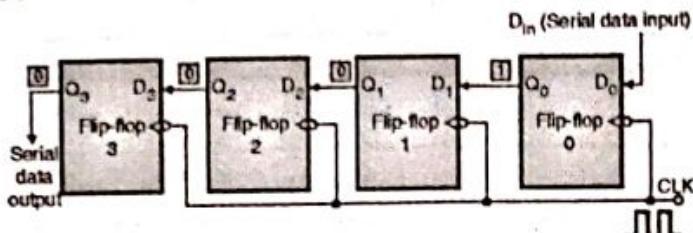
When this is to be done, this number should be applied to " $D_{in}$ " bit by bit with the MSB bit applied first. The D input of FF-0 i.e.  $D_0$  is connected to serial data input ( $D_{in}$ ). Output of FF-0 i.e.  $Q_0$  is connected to the input of the next flip-flop i.e.  $D_1$  and so on.

**Operation :**

Before application of clock signal let  $Q_3 Q_2 Q_1 Q_0 = 0000$  and apply MSB bit of the number to be entered to  $D_{in}$ . So  $D_{in} = D_0 = 1$ .

Apply the clock. On the first falling edge of clock, the FF-0 is set and the stored word in the register is,

$$Q_3 Q_2 Q_1 Q_0 = 0001$$

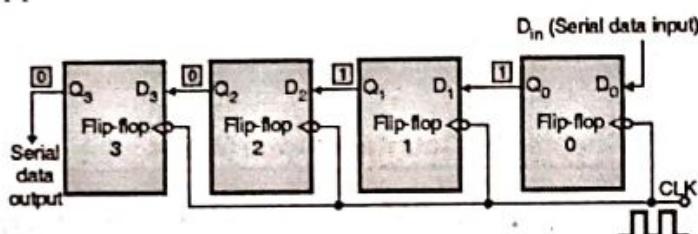


(C-733) Fig. 2- Q. 2(a) : Shift register status after first falling clock edge

Apply the next bit to  $D_{in}$ . So  $D_{in} = 1$ .

As soon as the next negative edge of the clock hits, FF-1 will set and the stored word changes to,

$$Q_3 Q_2 Q_1 Q_0 = 0011$$



(C-734) Fig. 3- Q. 2(a) : Shift register status after the second falling edge of clock

Apply the next bit to be stored i.e. 1 to  $D_{in}$ .

Apply the clock pulse. As soon as the third negative clock edge hits, FF-2 will be set and the output get modified to,

$$Q_3 Q_2 Q_1 Q_0 = 0111.$$

Similarly with  $D_{in} = 1$ , and with the fourth negative clock edge arriving, the stored word in the register is,

$$Q_3 Q_2 Q_1 Q_0 = 1111.$$

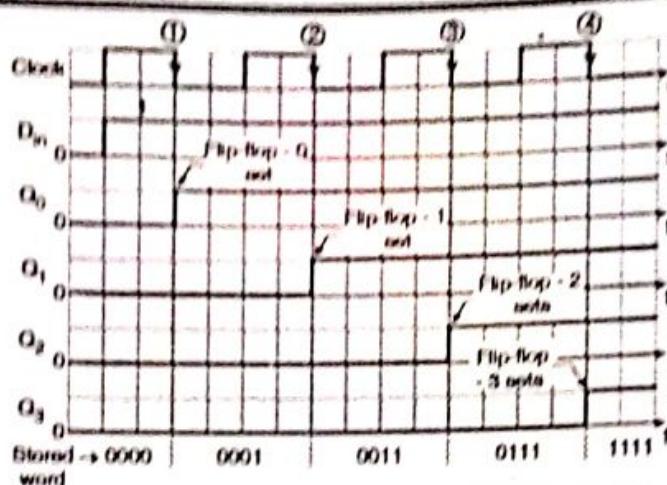
(C-737) Table 1- Q. 2(a) : Summary of shift left operation

	CLK	$Q_3$	$Q_2 = D_3$	$Q_1 = D_2$	$Q_0 = D_1$	Serial Input $D_{in} = D_0$
Initially		0	0	0	0	
1 <sup>st</sup>	↓	0	0	0	1	1
2 <sup>nd</sup>	↓	0	0	1	1	1
3 <sup>rd</sup>	↓	0	1	1	1	1
4 <sup>th</sup>	↓	1	1	1	1	1

Direction of data travel ←

**Waveforms for shift left operation :**

The waveforms for the shift left operation are shown in Fig. 4- Q. 2(a).



(C-738) Fig. 4-Q. 2(a) : Waveforms for the shift left operation

**Applications :**

1. For temporary data storage.
2. For multiplication and division.
3. As a delay line.
4. Serial to parallel converter.

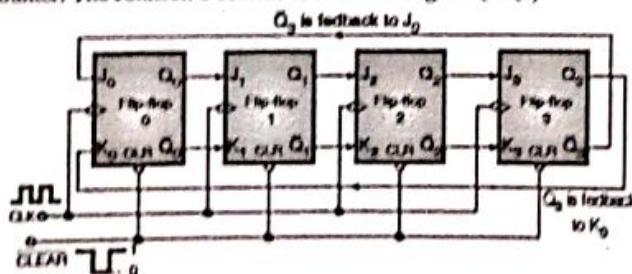
**Chapter 9 : Counters [Total Marks : 20]**

**Q. 3(a)** Explain the Johnson's counter. Design for initial state 0110. From initial state explain and draw all possible states. (10 Marks)

**Ans. :****Johnson's counter :**

In the ring counter the outputs of FF-3 were connected directly to the inputs of FF-0 i.e.  $Q_3$  to  $J_0$   $\bar{Q}_3$  to  $K_0$ .

Instead if the outputs are cross coupled to the inputs i.e. if  $Q_3$  is connected to  $K_0$  and  $\bar{Q}_3$  is connected to  $J_0$  then the circuit is called as twisted ring counter or Johnson's counter. The Johnson's counter is shown in Fig. 1-Q. 3(a).



(C-1342) Fig. 1-Q. 3(a) : Twisted ring counter or Johnson counter

All the flip-flops are negative edge triggered, and clock pulses are applied to all of them simultaneously. The clear inputs of all the flip-flops are connected together and connected to an external clear signal. Note that all these clear inputs are active low inputs.

**Operation :**

Initially a short negative going pulse is applied to the  $\overline{\text{clear}}$  input of all the flip-flops. This will reset all the flip-flops. Hence initially the outputs are,  $Q_3 Q_2 Q_1 Q_0 = 0 0 0 0$ . But  $\bar{Q}_3 = 1$  and since it is coupled to  $J_0$  it is also equal to 1.



$$\therefore J_0 = 1 \text{ and } K_0 = 0$$

....Initially

### On the first falling edge of clock pulse :

As soon as the first negative edge of clock arrives, FF-0 will be set. Hence  $Q_0$  will become 1.

But there is no change in the status of any other flip-flop.

Hence after the first negative going edge of the clock the flip-flop outputs are,

$$Q_3 Q_2 Q_1 Q_0 = 0001$$

### On the second negative going clock edge :

Before the second negative going clock edge,  $Q_3 = 0$  and  $\overline{Q}_3 = 1$ . Hence  $J_0 = 1$  and  $K_0 = 1$ . Also  $Q_0 = 1$ . Hence  $J_1 = 1$ . Hence as soon as the second falling clock edge arrives, FF-0 continues to be in the set mode and FF-1 will now set. Hence  $Q_1$  will become 1 and  $\overline{Q}_1 = 0$ .

There is no change in the status of any other flip-flop. Hence after the second clock edge the outputs are,

$$Q_3 Q_2 Q_1 Q_0 = 0011.$$

Similarly after the third clock pulse, the outputs are,

$$Q_3 Q_2 Q_1 Q_0 = 0111.$$

And after the fourth clock pulse, the outputs are,

$$Q_3 Q_2 Q_1 Q_0 = 1111.$$

Note that now  $\overline{Q}_3 = 0$  i.e.  $J_0 = 0$  and  $K_0 = 1$ .

Hence as soon as the fifth negative going clock pulse strikes, FF-0 will reset. But the outputs of the other flip-flops will remain unchanged. So after the fifth clock pulse, the outputs are,

$$Q_3 Q_2 Q_1 Q_0 = 1110$$

.....after the 5<sup>th</sup> clock pulse

This operation will continue till we reach the all zero output state. (i.e.  $Q_3 Q_2 Q_1 Q_0 = 0000$ ).

The operation of Johnson's counter is summarized in Table 1- Q. 3(a).

(C-6295) Table 1- Q. 3(a) : Summary of operation of Johnson's counter

CLEAR	CLK	$Q_3$	$Q_2$	$Q_1$	$Q_0$	State number	Decimal equivalent
	Initially	0	0	0	0	1	0
1	↓	0	0	0	1	2	1
1	↓	0	0	1	1	3	3
1	↓	0	1	1	1	4	7
1	↓	1	1	1	1	5	15
1	↓	1	1	1	0	6	14
1	↓	1	1	0	0	7	12
1	↓	1	0	0	0	8	8
1	↓	0	0	0	0	1	0

Note that there are 8 distinct states of output.

In general we can say that the number of states of a Johnson's counter is twice the number of flip-flops used. Therefore for a 4-flip-flop Johnson's counter, there are 8-distinct output states.

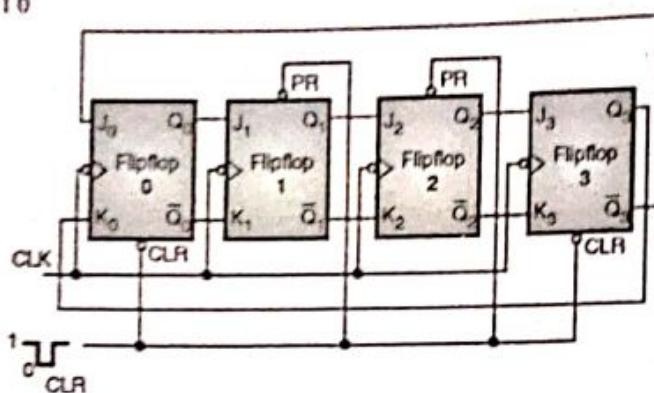
**Solution to problem :**

The required Johnson's counter is shown in Fig. 2- Q. 3(a).

Counters 0 and 3 are reset to 0 while counters 1 and 2 are preset to 1 initially.

$$\therefore Q_3 Q_2 Q_1 Q_0 = 0110$$

...initially



(C-3564) Fig. 2- Q. 3(a) : Required Johnson's counter

The other possible states of this Johnson's counter are listed in Table 2-Q. 3(a).

(C-3565) Table 2- Q. 3(a) : All possible states of a Johnson's counter

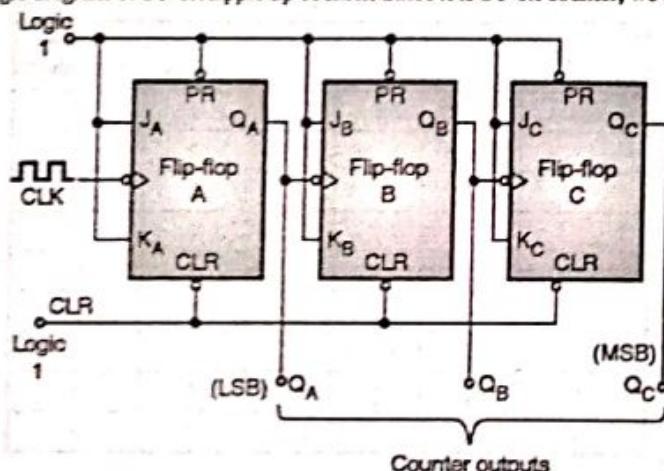
Clear	CLK	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>
1	X	0	1	1	0
1	↓	1	1	0	1
1	↓	1	0	1	0
1	↓	0	1	0	0
1	↓	1	0	0	1

**Q. 5(a)** Explain the working of 3 bit asynchronous counter with proper timing diagram. (10 Marks)

**Ans. :**

**3 bit asynchronous counter :**

Fig. 1- Q. 5(a) shows the logic diagram of a 3-bit ripple up counter. Since it is a 3-bit counter, we need to use 3-flip-flops.



(C-775) Fig. 1- Q. 5(a) : 3-bit ripple up counter



Table 1- Q. 5(a) summarizes the operation of the 3-bit asynchronous up counter.

Table 1- Q. 5(a) : Summary of operation of a 3-bit ripple up counter

Clock	Flip-flop outputs			State	Decimal equivalent
	$Q_C$ (MSB)	$Q_B$	$Q_A$ (LSB)		
Initially	0	0	0	1	0
1 <sup>st</sup> (↓)	0	0	1	2	1
2 <sup>nd</sup> (↓)	0	1	0	3	2
3 <sup>rd</sup> (↓)	0	1	1	4	3
4 <sup>th</sup> (↓)	1	0	0	5	4
5 <sup>th</sup> (↓)	1	0	1	6	5
6 <sup>th</sup> (↓)	1	1	0	7	6
7 <sup>th</sup> (↓)	1	1	1	8	7
8 <sup>th</sup> (↓)	0	0	0	1	0

Note that the asynchronous preset and clear terminals are also being used. Both of them are active low inputs. Hence for the normal output of the counter preset and clear terminals should be connected to logic 1.

#### Number of states :

$$\text{Number of states} = 2^n = 2^3 = 8.$$

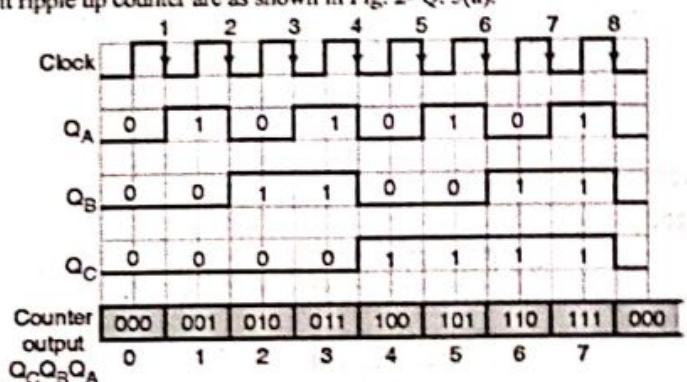
The 3 bit ripple up counter can have 8 distinct states i.e.  $Q_C\ Q_B\ Q_A$  can take up values from 000, 001, 010, .....110, 111.

#### Maximum count :

Maximum count =  $2^n - 1 = 8 - 1 = 7$ . Refer Table 1-Q. 5(a) the maximum count is  $Q_C\ Q_B\ Q_A = 1\ 1\ 1$  i.e. decimal 7. Note that  $Q_C$  is treated as MSB and  $Q_A$  as LSB.

#### Timing diagram :

The timing diagram of a 3-bit ripple up counter are as shown in Fig. 2- Q. 5(a).



(C-776) Fig. 2- Q. 5(a) : Timing diagram for a 3-bit ripple up counter

May 2018

Chapter 1 : Number Systems and Codes [Total Marks : 05]

Q. 1(a) Convert the following numbers as mentioned against them :

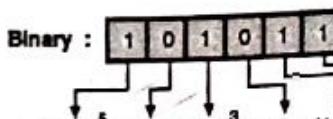
(5 Marks)

- (101011)<sub>2</sub> convert to decimal number.
- Convert (129.625)<sub>10</sub> hexadecimal form.
- Write (-20)<sub>10</sub> in two's complement form.

Ans. :

1.  $(101011)_2 = (?)_{10}$

Step 1 : Convert binary to decimal :

Binary :   
 Decimal (N) :  $(1 \times 2^5) + 0 + (1 \times 2^3) + 0 + (1 \times 2^1) + (1 \times 2^0)$   
 $N = 32 + 8 + 2 + 1 = (43)_{10}$   
 (C-7229)

$\therefore (101011)_2 = (43)_{10}$

2.  $(129.625)_{10} = (?)_{16}$

Step 1 : Convert the integer part :

16	129	1	LSB
16	8	8	
0			MSB

$\therefore (129)_{10} = (81)_{16}$

(C-7230)

Step 2 : Convert the fractional part :

Decimal	Base	Product	Carry	Hex
0.625	$\times 16$	10.0	10	A MSB
0.0	$\times 16$	0.0	0	0 LSB

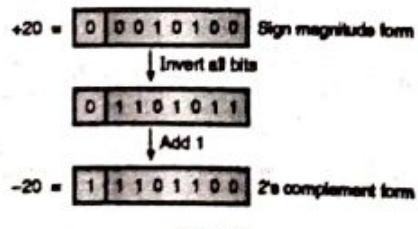
$\therefore (0.625)_{10} = (0.A0)_{16}$

(C-7231)

$\therefore (129.625)_{10} = (81.A0)_{16}$

...Ans

3. Two's complement of  $(-20)_{10}$  :



Chapter 3 : Logic Minimization and Reduction Techniques [Total Marks : 10]

Q. 2(a) Simplify using Quine McCluskey method and draw the logic diagram using basic gates for the following function :

$Y = F(A, B, C, D) = \sum m(5, 11, 13, 14, 15) + \sum d(4, 6, 7)$ .

(10 Marks)



Ans. :

Step 1 : Group the minterms according to number of 1's :

(C-7233) Table 1- Q. 2(a)(a)

Group	Minterms	Binary representation			
		A	B	C	D
1	4*	0	1	0	0
2	5	0	1	0	1
	6*	0	1	1	0
3	7*	0	1	1	1
	11	1	0	1	1
	13	1	1	0	1
	14	1	1	1	0
4	15	1	1	1	1

\* represents the don't care terms

Step 2 : Group the minterms of form pairs :

(C-7234) Table 1- Q. 2(a)(b)

Group	Minterm pair	Binary representation				Prime Implicants
		A	B	C	D	
1	4, 5	0	1	0	-	✓
	4, 6	0	1	-	0	✓
2	5, 7	0	1	-	1	✓
	6, 7	0	1	1	-	✓
	5, 13	-	1	0	1	✓
	6, 14	-	1	1	0	✓
3	7, 15	-	1	1	1	✓
	11, 15	1	-	1	1	ACD
	13, 15	1	1	-	1	✓
	14, 15	1	1	1	-	✓

Step 3 : Group the minterms to form quads :

(C-7235) Table 1- Q. 2(a)(c)

Group	Minterm quads	Binary representation				Prime Implicants
		A	B	C	D	
1	4, 5, 6, 7	0	1	-	-	AB
	4, 6, 5, 7	0	1	-	-	
2	5, 7, 13, 15	-	1	-	1	BD
	5, 13, 7, 15	-	1	-	1	
	6, 7, 14, 15	-	1	1	-	BC
	6, 14, 7, 15	-	1	1	-	



**Step 4 : Prepare the table of prime implicants :**

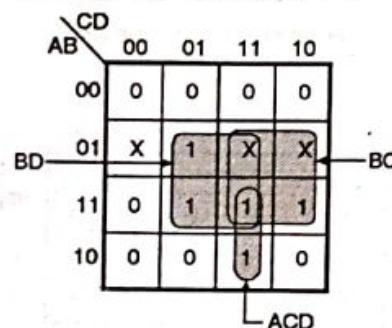
(C-7236) Table 1- Q. 2(a)(d)

Prime implicants	Decimal numbers	Given minterms				
		5	11	13	14	15
ACD	11, 15		(X)			X
$\bar{A}B$	4, 5, 6, 7	X				
BD	5, 7, 13, 15	X		(X)		X
BC	6, 7, 14, 15				(X)	X

The encircled crosses represents the EPIs. So the terms ACD, BD and BC are the EPIs. They cover all minterms.

$$\therefore F(A, B, C, D) = ACD + BD + BC$$

**Step 5 : Crosscheck using K-map :**

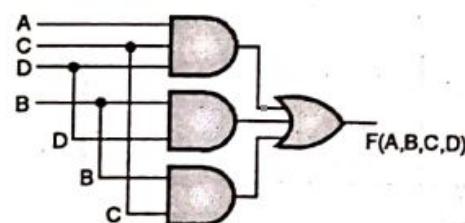


(C-7237) Fig. 1- Q. 2(a) : K-map

$$\therefore F(A, B, C, D) = ACD + BD + BC$$

**We get the same simplified expression using K-map**

**Step 6 : Logic diagram :**



(C-7238) Fig. 2- Q. 2(a) : Logic diagram

#### Chapter 4 : Arithmetic Circuits [Total Marks : 20]

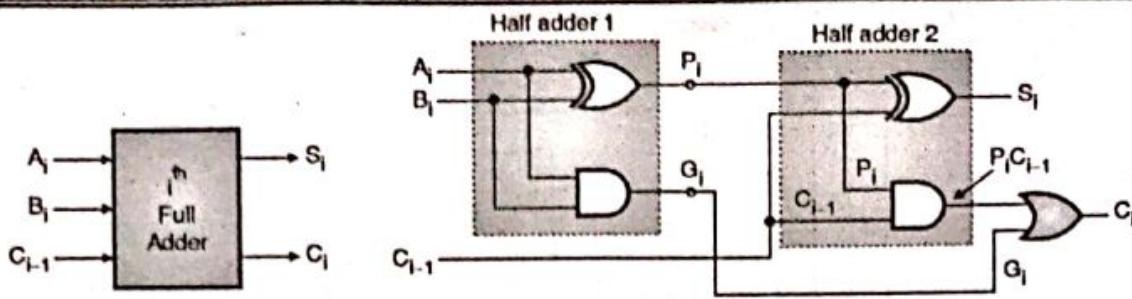
**Q. 3(b) Design 3 bit look ahead carry generator circuit.**

(10 Marks)

**Ans. :**

**3 bit look ahead carry generator :**

Consider the block diagram of full adder Fig. 1- Q. 3(b)(a) and its AND-OR-EXOR realization shown in Fig. 1- Q. 3(b)(b).


 (a) Block diagram of  $i^{\text{th}}$  full adder

(b) Realization using AND-OR-EXOR gates

(C-389) Fig. 1- Q. 3(b)

Refer Fig. 1- Q. 3(b)(b) to write,

$$P_i = A_i \oplus B_i \quad \dots(1)$$

$$\text{and } G_i = A_i B_i \quad \dots(2)$$

$$\text{Also } S_i = P_i \oplus C_{i-1} = A_i \oplus B_i \oplus C_{i-1} \quad \dots(3)$$

$$\text{and } C_i = G_i + P_i C_{i-1} \quad \dots(4)$$

The carry output  $G_i$  of the first half adder is equal to 1 if  $A_i = B_i = 1$  and a carry is generated at the  $i^{\text{th}}$  stage of the parallel adder. That means  $C_i = 1$ . This variable  $G_i$  is known as Carry Generate and its value does not depend on the input carry i.e.  $C_{i-1}$ . The variable  $P_i$  is called as Carry Propagate because this term is associated with the propagation of carry from  $C_{i-1}$  to  $C_i$ . Now consider Equation (4) i.e.  $C_i = G_i + P_i C_{i-1}$ . Using this equation, we can write the expression for the carry output of each stage in a 4 bit parallel adder as follows :

$$P_i = A_i \oplus B_i$$

**Stage expression for carry output :**

$$0 \quad C_0 = G_0 + P_0 C_{-1} \quad \dots(5)$$

$$\begin{aligned} 1 \quad C_1 &= G_1 + P_1 C_0 \\ &= G_1 + P_1 (G_0 + P_0 C_{-1}) \\ \therefore C_1 &= G_1 + P_1 G_0 + P_0 P_1 C_{-1} \end{aligned} \quad \dots(6)$$

**Stage expression for carry output :**

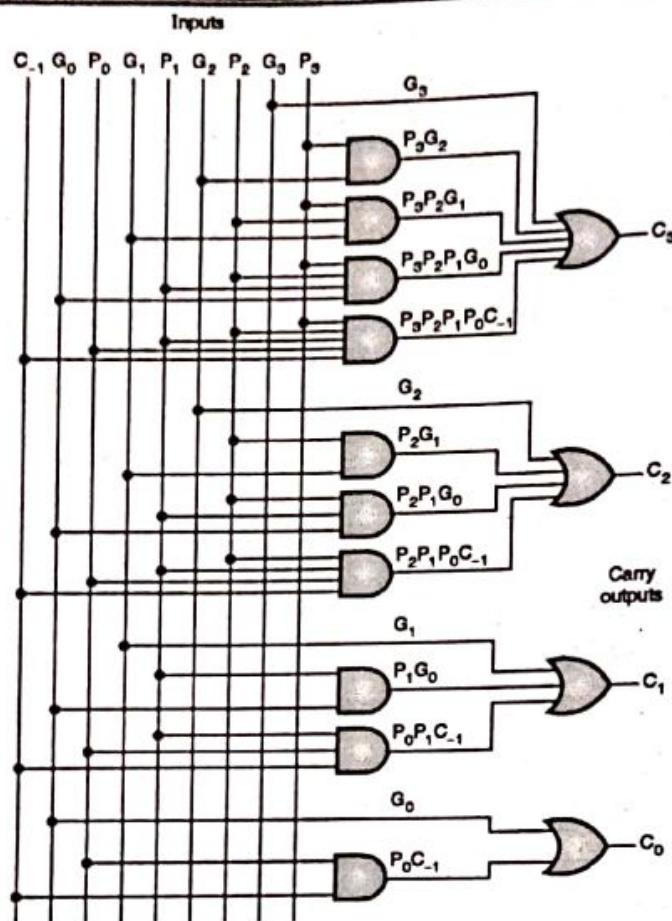
$$\begin{aligned} 2 \quad C_2 &= G_2 + P_2 C_1 \\ &= G_2 + P_2 (G_1 + P_1 G_0 + P_0 P_1 C_{-1}) \\ \therefore C_2 &= G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_{-1} \end{aligned} \quad \dots(7)$$

$$3 \quad C_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_{-1} \quad \dots(8)$$

In the expressions stated above, the variables involved are  $G_0$ ,  $G_1$ ,  $G_2$ ,  $G_3$ ,  $P_0$ ,  $P_1$ ,  $P_2$ ,  $P_3$  and  $C_{-1}$ . Out of them the  $G$  variables are generated from the  $A$  and  $B$  inputs using AND gates (as illustrated in Equation (2)). And the  $P$  variables are obtained again directly from  $A$  and  $B$  inputs using EX-OR gates (as illustrated in Equation (1)). If the  $G$ ,  $P$  and  $C_{-1}$  are at a time available, then it is possible to produce the carry outputs  $C_0$ ,  $C_1$ ,  $C_2$  and  $C_3$  by using 2-level realization (AND-OR or NAND-NAND) etc.

These carry outputs are then connected to the carry inputs of the succeeding stages. This eliminates the problem of carry getting propagated like ripples.

The logic circuit of a look ahead carry generator is shown in Fig. 2-Q. 3(b) and it is based on Equations (1) through (8).



(C-39) Fig. 2- Q. 3(b) : Logic diagram of the look ahead carry generator

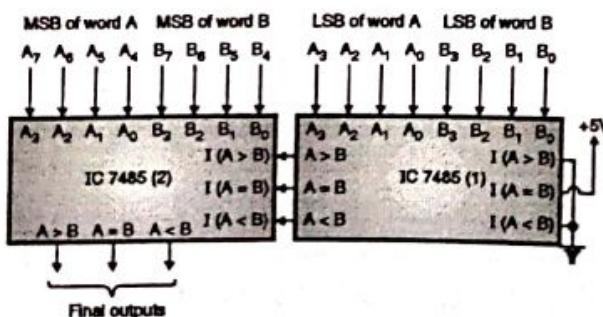
**Q. 6(b) Design 8 bit comparator using 4 bit comparator IC 7485 and explain its operation.**

(10 Marks)

**Ans. :**

**8 bit comparator using 4 bit comparator IC 7485 :**

IC 7485 is a four bit magnitude comparator. If we want to compare the words that are longer than 4 bit then the cascading is required to be done. An 8-bit comparator using IC 7485 is shown in Fig. 1- Q. 6(b). The outputs of IC 7485-1 are applied to the cascading inputs of IC 7485 - 2. The two 8-bit words to be compared are A ( $A_7 - A_0$ ) and B ( $B_7 - B_0$ ). These are divided in two 4-bit words each and then applied to the two comparator ICs.



(C-410) Fig. 1- Q. 6(b) : 8-bit magnitude comparator using IC 7485

**Operation :**

The MSB bits  $A_7$  and  $B_7$  are compared first. If  $A_7 = B_7$ , then the next bits  $A_6$  and  $B_6$  are compared. This will continue up to  $A_0$  and  $B_0$ . If  $A_7, A_6, A_5, A_4 = B_7, B_6, B_5, B_4$ , then IC 7485 (2) will check the cascading inputs. IC 7485 (1) will continue the comparison from  $A_3 - B_3$  to  $A_0 - B_0$ . If  $A_3, A_2, A_1, A_0 = B_3, B_2, B_1, B_0$ , then IC 7485 (1) will check its cascading inputs.

**Chapter 5 : Multiplexer & Demultiplexer [Total Marks : 30]**

**Q. 3(a)** Implement the following function using only one 4 : 1 multiplexer and gates.

$$Y = F(A, B, C, D) = \sum m(2, 3, 5, 7, 10, 11, 12, 13)$$

(10 Marks)

**Ans. :**

**Step 1 : Write the design table :**

	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
AB	0	1	(2)	(3)
AB	4	(5)	6	(7)
A <bar>B</bar>	8	9	(10)	(11)
AB	(12)	(13)	14	15
AB				

(C-7239) Fig. 1- Q. 3(a)

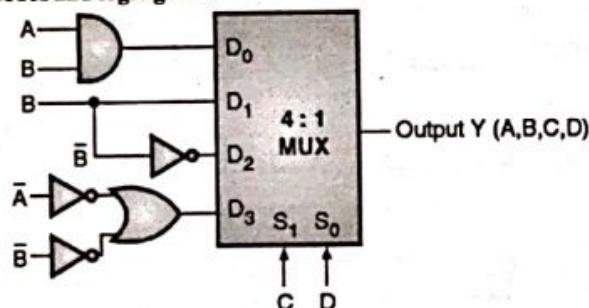
$$D_0 = AB$$

$$D_1 = \bar{A}\bar{B} + A\bar{B} = B(\bar{A} + A) = B$$

$$D_2 = \bar{A}\bar{B} + A\bar{B} = \bar{B}(\bar{A} + A) = \bar{B}$$

$$D_3 = \bar{A}\bar{B} + \bar{A}\bar{B} + A\bar{B} = \bar{A}(B + \bar{B}) + A\bar{B} = \bar{A} + A\bar{B} = \bar{A} + \bar{B}$$

**Step 2 : Implementation using 4 : 1 MUX and logic gates :**



(C-7240) Fig. 2- Q. 3(a)

**Q. 4(b)** Implement full adder using decoder having active low outputs and gates with fan in 2.

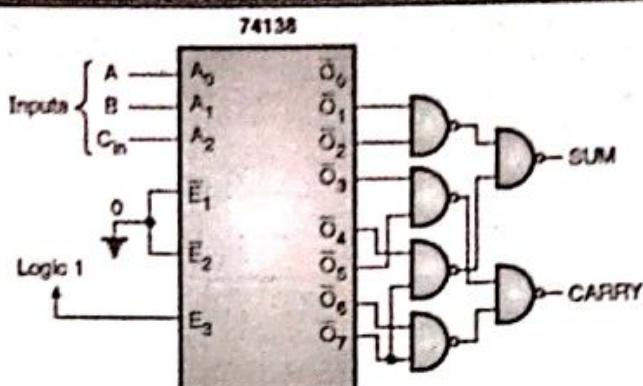
(10 Marks)

**Ans. :**

The truth table of a full adder is shown in Table 1- Q. 4(b).

Table 1- Q. 4(b) : Truth table of full adder

Inputs			Outputs	
A	B	C <sub>m</sub>	Sum	Carry
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



(C-724) Fig. 1- Q. 4(b) : Full adder implementation using IC 74138 and 2 fan in gates

The  $A$ ,  $B$  and  $C_{in}$  inputs are applied to the  $A_0$ ,  $A_1$  and  $A_2$  inputs of the decoder. The outputs of 74138 are active low. Hence we should apply  $\bar{O}_1$ ,  $\bar{O}_2$ ,  $\bar{O}_4$  and  $\bar{O}_5$  to two input NAND gates as shown in Fig. 1- Q. 4(b) to obtain the sum output.

Similarly outputs  $\bar{O}_3$ ,  $\bar{O}_6$ ,  $\bar{O}_7$  to the other set of NAND gates to obtain the carry output. Implementation of full adder is shown in Fig. 1- Q. 4(b). All the enable terminals are connected to their respective active levels to enable the IC.

**Q. 5(b) Explain parity circuits.**

(10 Marks)

**Ans. :**

#### Parity generators / checkers :

A parity bit is an additional bit which is added to a binary word in order to make the number of 1's in the new word formed, even (even parity) or odd (odd parity). The even and odd parity has been illustrated in Fig. 1- Q. 5(b).

Given word : 

1	0	0	1	0	1	0
---	---	---	---	---	---	---

Word with even parity : 

1	1	0	0	1	0	1	0
---	---	---	---	---	---	---	---

 → Number of 1's = 4  
→ Parity bit

Word with odd parity : 

0	1	0	0	1	0	1	0
---	---	---	---	---	---	---	---

 → Number of 1's = 3  
→ Parity bit

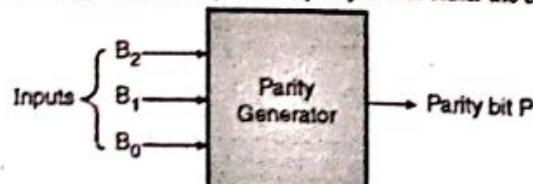
(C-503) Fig. 1- Q. 5(b) : Illustration of even and odd parity

Parity bits are added in order to detect any error occurred in the process of transmission. The words with parity bits are transmitted by the transmitter. At the receiver, these words are received and checked for errors. The receiver checks parity of the received word and compares it with the parity of the transmitted word. If they are same then there is no error. But if the parity of received word is different from that of transmitted word, then the receiver understands that an error is present.

In the even parity system, the added parity bit will make the total number of 1's an even number. In the odd parity system, the added parity bit will make the total number of 1's an odd number.

#### Parity generator :

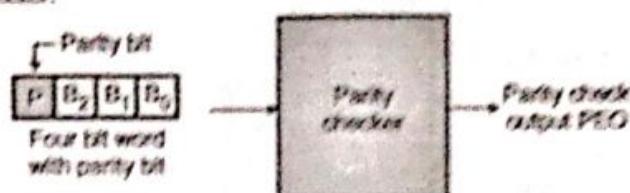
Parity generator is a logic circuit which generates the parity bits for even parity or odd parity. Block diagram of the parity generator is shown in Fig. 2- Q. 5(b). The digital word is applied at the input and a parity bit for either the even parity or odd parity.



(C-504) Fig. 2- Q. 5(b) : Block diagram of parity generator

**Parity Checker :**

The block diagram of parity checker is shown in Fig. 3- Q. 5(b). The parity checker circuit is at the receiver. The received word with a parity bit is applied to the parity checker.



(a.m) Fig. 3- Q. 5(b) : Block diagram of parity checker

The parity checker will check the parity of the received word and produce its output. Since the parity of transmitted words is "odd", the parity checker will treat the received words with odd parity as the correct words and those with even parity as the incorrect words. The output of parity checker circuit is denoted by parity error output PEO.

$PEO = 1$  .... If error is present i.e. if the received four bit word has an even parity.

$PEO = 0$  .... If there is no error i.e. if the received four bit word has an odd parity.

**Truth table of parity checker :**

According to the conditions stated above, the truth table is prepared as follows :

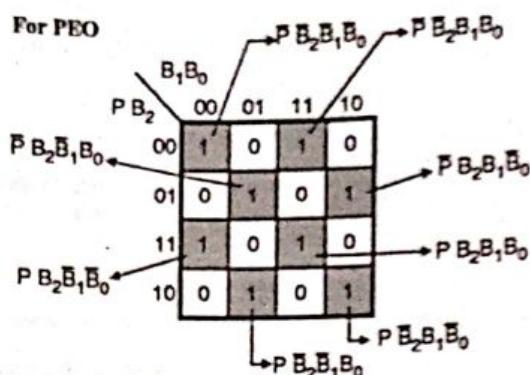
Table 1- Q. 5(b) : Truth table of a parity checker

Inputs				Output
P	B <sub>4</sub>	B <sub>3</sub>	B <sub>2</sub>	PEO
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

1 = error (even parity)  
0 = No error (odd parity)



## K-map and simplifications :



(C-508) Fig. 4- Q. 5(b) : K-map for a parity checker output

## Simplification :

$$\begin{aligned}
 \text{PEO} &= \bar{P} \bar{B}_2 \bar{B}_1 \bar{B}_0 + \bar{P} \bar{B}_2 B_1 B_0 + \bar{P} B_2 \bar{B}_1 B_0 + \bar{P} B_2 B_1 \bar{B}_0 + P B_2 \bar{B}_1 \bar{B}_0 + P B_2 B_1 B_0 + P \bar{B}_2 \bar{B}_1 B_0 + P \bar{B}_2 B_1 \bar{B}_0 \\
 &= \bar{B}_1 \bar{B}_0 (\bar{P} \bar{B}_2 + P B_2) + B_1 B_0 (\bar{P} \bar{B}_2 + P B_2) + \bar{B}_1 B_0 (\bar{P} B_2 + P \bar{B}_2) + B_1 \bar{B}_0 (\bar{P} B_2 + P \bar{B}_2) \\
 \therefore \text{PEO} &= (\bar{P} \bar{B}_2 + P B_2) (\bar{B}_1 \bar{B}_0 + B_1 B_0) + (\bar{P} B_2 + P \bar{B}_2) (\bar{B}_1 B_0 + B_1 \bar{B}_0) \\
 &= (P \odot B_2) (B_1 \odot B_0) + (P \oplus B_2) (B_1 \oplus B_0) \\
 &= (\overline{P \oplus B_2}) (\overline{B_1 \oplus B_0}) + (P \oplus B_2) (B_1 \oplus B_0)
 \end{aligned}$$

Let  $X = P \oplus B_2$  and  $Y = (B_1 \oplus B_0)$ 

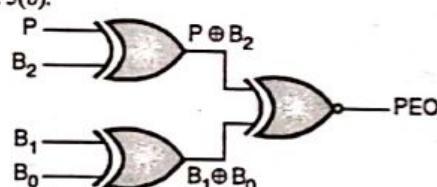
$$\therefore \text{PEO} = \bar{X} \bar{Y} + X Y = X \odot Y = \overline{X \oplus Y}$$

Substituting the values we get,

$$\text{PEO} = \overline{(P \oplus B_2) \oplus (B_1 \oplus B_0)}$$

## Logic diagram :

Logic diagram is as shown in Fig. 5- Q. 5(b).



(C-509) Fig. 5- Q. 5(b)

Chapter 6 : Logic Families [Total Marks : 15]

Q. 1(d) Explain current and voltage parameters of logic families.

(5 Marks)

Ans. :

## Voltage parameters (Threshold levels) :

Ideally the input voltage levels of 0 V and + 5 V (for TTL) are called as logic 0 and 1 levels respectively. However practically we won't always observe or obtain the voltage levels matching exactly to these values. Therefore it is necessary to define the worst case input voltages.

1.  $V_{L(\max)}$  - Worst case low level input voltage :

This is the maximum value of input voltage which is to be considered as a logic 0 level. If the input voltage is higher than  $V_{L(\max)}$  then it will not be treated as a low (0) input level.



**2.  $V_{IH(\min)}$  - Worst case high level input voltage :**

This is the minimum value of the input voltage which is to be considered as a logic 1 level. If the input voltage is lower than  $V_{IH(\min)}$  then it will not be treated as a High (1) input.

**3.  $V_{OH(\min)}$  - Worst case high level output voltage :**

This is the minimum value of the output voltage which will be considered as a logic HIGH (1) level. If the output voltage is lower than this level then it won't be treated as a HIGH (1) output.

**4.  $V_{OL(\max)}$  - Worst case low level output voltage :**

This is the maximum value of the output voltage which will be considered as a logic LOW (0) level. If the output voltage is higher than this value then it won't be treated as a LOW (0) output.

**Current parameters :**

- $I_{IL}$  - Low level input current :** It is the current that flows into the input terminals when a low level input voltage in the specified range is applied.
- $I_{IH}$  - High level input current :** It is the current that flows into the input terminals when a high level input voltage in the specified range is applied.
- $I_{OL}$  - Low level output current :** This is the current that flows out of the output when the output voltage happens to be in the specified low (0) voltage range and a specified load is applied.
- $I_{OH}$  - High level output current :** This is the current flowing from the output when the output voltage happens to be in the specified HIGH (1) voltage range and a specified load is applied.

**Q. 4(a) Draw circuit diagram of 2 input TTL NAND gate and explain its operation.**

(10 Marks)

Ans.:

**2 input TTL NAND gate :**

A two input TTL-NAND gate is shown in Fig. 1- Q. 4(a). A and B are the two inputs while Y is the output terminal of this NAND gate.

**Operation :**

In order to understand the operation of this circuit, now replace transistor  $Q_3$  by its equivalent circuit as shown in Fig. 2- Q. 4(a).

1. A and B are the input terminals. The input voltages A and B can be either LOW (zero Volt ideally) or HIGH ( $+V_{CC}$  ideally).
2. **A and B both LOW :** If A and B both are connected to ground, then both the B-E junctions of transistor  $Q_1$  are forward biased.

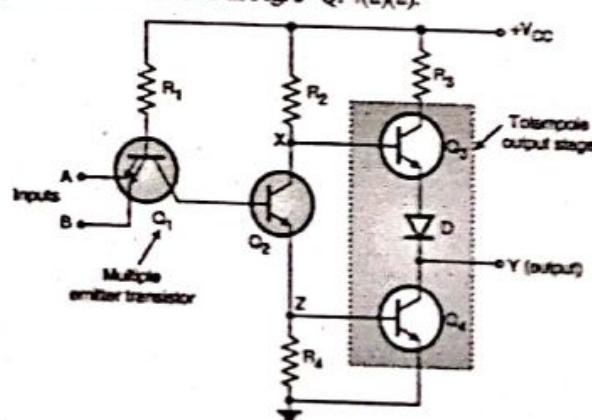
Hence diodes  $D_1$  and  $D_2$  in Fig. 2- Q. 4(a) will conduct to force the voltage at point C in Fig. 2- Q. 4(a) to 0.7 V. This voltage is insufficient to forward bias base-emitter junction of  $Q_2$  due to the presence of  $D_3$ . Hence  $Q_2$  will remain OFF. Therefore its collector voltage  $V_X$  rises to  $V_{CC}$ .

As transistor  $Q_3$  is operating in the emitter follower mode, output Y will be pulled up to high voltage.

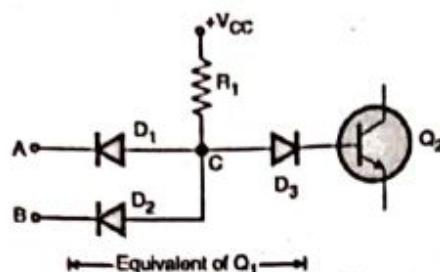
$$\therefore Y = 1 \text{ (HIGH)}$$

...For  $A = B = 0$  (LOW)

The equivalent circuit for this input condition is shown in Fig. 3- Q. 4(a)(a).



(C-1012) Fig. 1- Q. 4(a) : Two input TTL NAND gate

(C-1012) Fig. 2-Q. 4(a) : Transistor  $Q_1$  is replaced by its equivalent

3. Either A or B LOW : If any one input (A or B) is connected to ground with the other terminal left open or connected to  $+V_{CC}$ , then the corresponding diode ( $D_1$  or  $D_2$ ) will conduct.

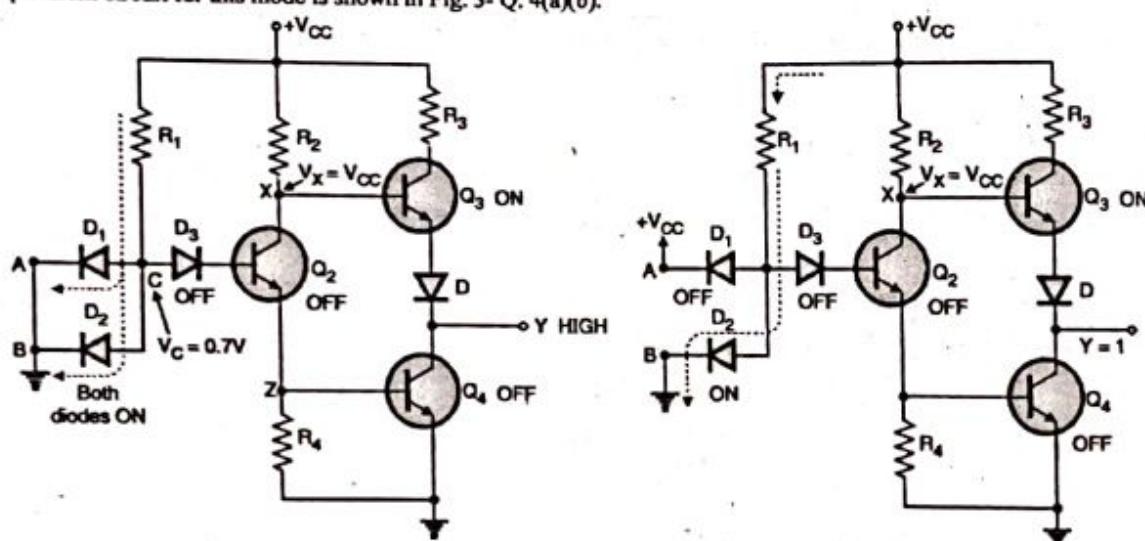
This will pull down the voltage at "C" to 0.7 V. (Fig. 3-Q. 4(a)). This voltage is insufficient to turn ON  $D_3$  and  $Q_2$ . So it remains OFF. So collector voltage  $V_X$  of  $Q_2$  will be equal to  $V_{CC}$ . This voltage acts as base voltage for  $Q_3$ .

As  $Q_3$  acts as an emitter follower, output Y will be pulled to  $V_{CC}$ .

$$\therefore Y = 1 \begin{cases} \text{if } A = 0 \text{ and } B = 1 \\ \text{if } A = 1 \text{ and } B = 0 \end{cases}$$

(C-6373)

The equivalent circuit for this mode is shown in Fig. 3-Q. 4(a)(b).

(a) Equivalent circuit for  $A = B = 0$ (b) Equivalent circuit for  $A = 1, B = 0$ 

(C-1013) Fig. 3-Q. 4(a)

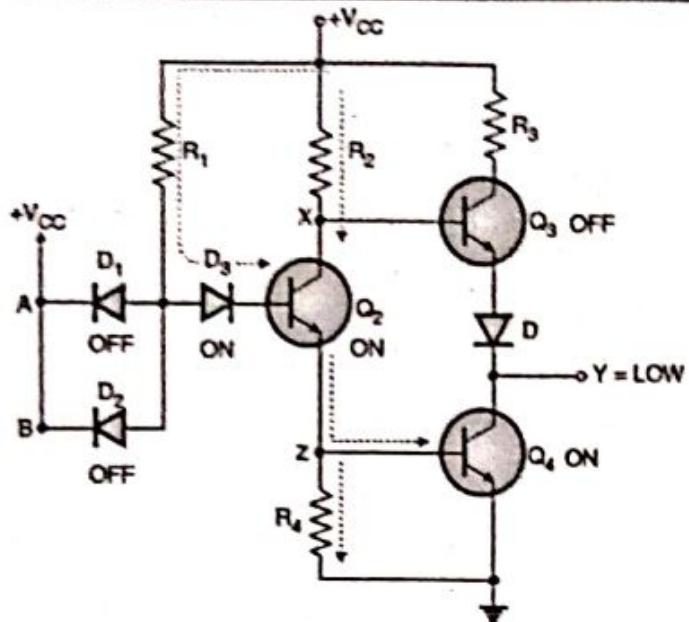
#### 4. A and B both HIGH :

If A and B both are connected to  $+V_{CC}$ , then both the diodes  $D_1$  and  $D_2$  will be reverse biased and do not conduct. Therefore voltage at point "C" i.e. at the anode of  $D_3$  increases to a sufficiently high value.

Therefore diode  $D_3$  is forward biased and base current is supplied to transistor  $Q_2$  via  $R_1$  and  $D_3$ , as shown in Fig. 4-Q. 4(a)(c). As  $Q_2$  conducts, the voltage at X will drop down and  $Q_3$  will be OFF, whereas voltage at Z (across  $R_3$ ) will increase to a sufficient level to turn ON  $Q_4$ . As  $Q_4$  goes into saturation, the output voltage Y will be pulled down to a low voltage.

$$\therefore Y = 0 \dots \text{For } A = B = 1$$

The equivalent circuit for this mode of operation is shown in Fig. 4-Q. 4(a)(c).



(C-1014) Fig. 4- Q. 4(a) : Equivalent circuit for  $A = B = 1$

### Chapter 7 : Latches and Flip Flops [Total Marks : 15]

Q. 1(c) Explain use of latch as a switch debouncer.

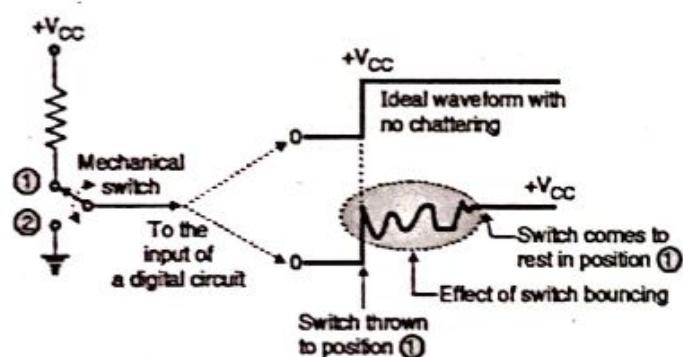
(5 Marks)

Ans. :

Latch as a switch debouncer :

Sometimes we need to use mechanical switches as input devices in order to enter digital information (0 or 1) into a digital system. Switch bouncing or chattering is a very serious problem associated with these switches. When the pole of a switch (also called as the arm) is thrown from say position 2 to 1. (see Fig. 1- Q. 1(c)(a)), it chatters or bounces a number of times before finally coming to rest in position 1.

The effect of this unwanted bouncing has been illustrated in Fig. 1- Q. 1(c)(b). The input to the logic circuit will oscillate due to switch chattering.

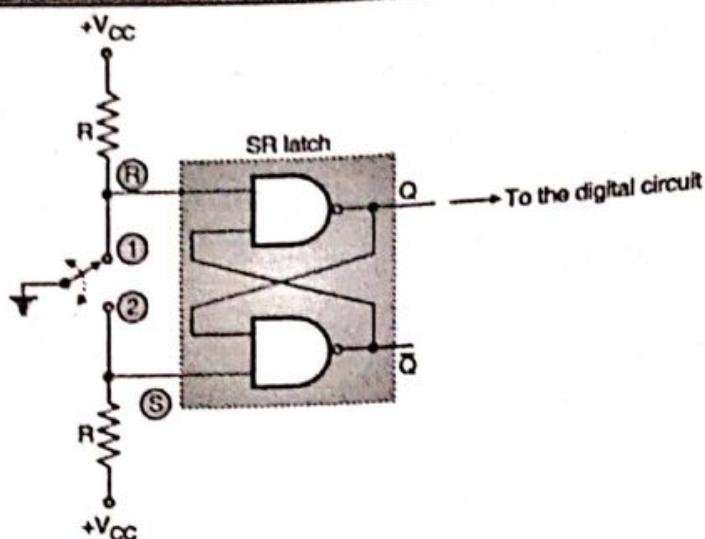


(a) Mechanical switch

(b) Effect of switch bouncing

(C-641) Fig. 1- Q. 1(c)

This produces an oscillatory output from a sequential circuit and creates problems in the operation of the system. This problem can be sorted out by using bounce elimination switch as shown in Fig. 1- Q. (c)(a). Fig. 2- Q. 1(c) shows how we can use the SR latch for creating a bounce elimination switch.

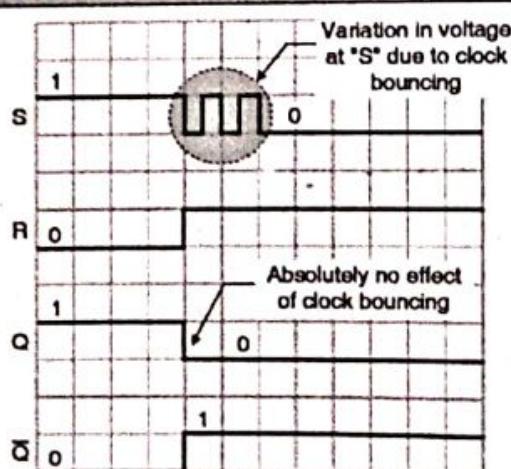


(C-642) Fig. 2- Q. 1(c) : A bounce elimination switch

**Operation :**

(C-643)

- Initially let the switch be in position 1
- ↓
- Therefore  $S = 1$  and  $R = 0$ . Hence  $Q = 1$
- ↓
- Now the switch is thrown to position 2
- ↓
- Therefore  $S = 0$  and  $R = 1$  as the switch first  
 $\therefore Q = 0$
- ↓
- Now if the switch bounces off the contact 2, then  $S = R = 1$
- ↓
- Hence  $Q$  output will remain unchanged. Hence  $Q = 0$
- ↓
- After some bounces, the switch comes to rest at position 2



(C-644) Fig. 3- Q. 1(c) : Waveforms of bounce elimination clock

Thus at the Q output of SR latch we get a clean waveform without any oscillations as shown in Fig. 3- Q. 1(c).

**Q. 6(a) Convert the flip flop**

- JKMS to D flip flop
- SR to T flip flop.

(10 Marks)

**Ans.:**

### Conversion of JKMS Flip Flop to D Flip Flop Conversion :

**Step 1 : Write the truth table for JK to D conversion :**

The truth table is as follows :

(C-6390) Table 1- Q. 6(a) : Truth table for JK to D flip flop conversion

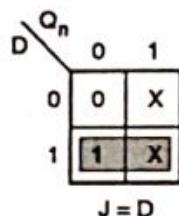
Inputs			Outputs	
D	Previous state $Q_n$	Next state $Q_{n+1}$	J	K
0	0	0	0	X
1	0	1	1	X
0	1	0	X	1
1	1	1	X	0

← Excitation table of D FF →

← Excitation table of JK FF →

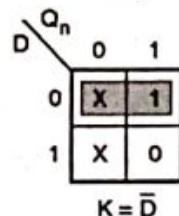
**Step 2 : K maps and simplification :**

For J output



(a) K map for J output

For K output

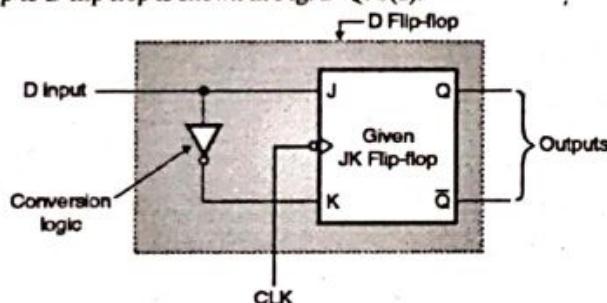


(b) K map for K output

(C-623) Fig. 1- Q. 6(a)

**Step 3 : Draw the logic diagram :**

The logic diagram for JK flip flop to D flip flop is shown in Fig. 2- Q. 6(a).



(C-624) Fig. 2- Q. 6(a) : Logic diagram for conversion from JK FF to D FF

### Conversion of SR Flip Flop to T Flip Flop :

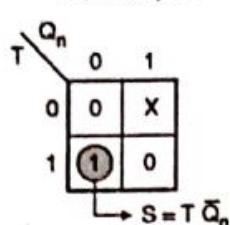
**Step 1 : Write the truth table :**

Table 2- Q. 6(a) : Truth table for SR FF to T FF

Inputs			Outputs	
T	Present state $Q_n$	Next state $Q_{n+1}$	S	R
0	0	0	0	X
1	0	1	1	0
1	1	0	0	1
0	1	1	X	0

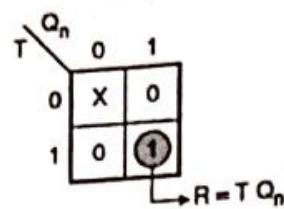
**Step 2 :** Write the K maps and obtain the expressions for S and R :

### For S output



(a) K map for S

### For R output

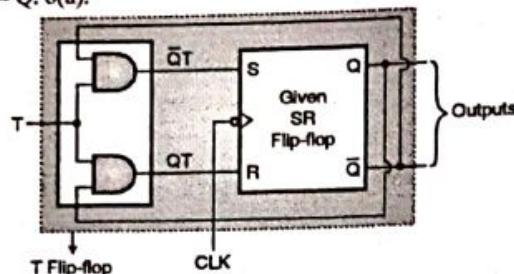


(b) K map for R

(C-617) Fig. 3- Q. 6(a)

### **Step 3 : Draw the logic diagram :**

The logic diagram is shown in Fig. 4-Q-6(a).



(C-618) Fig. 4-Q. 6(a) : Conversion from SR flip flop to T flip flop

## **Chapter 9 : Counters [Total Marks : 25]**

(5 Marks)

Q 1(b) Write differences between synchronous and asynchronous counters.

Ans. 2

#### **Comparison of synchronous and asynchronous counters:**

Sr. No.	Parameter	Asynchronous counter	Synchronous counter
1.	Circuit complexity	Logic circuit is simple	With increase in number of states, the logic circuit becomes complicated.
2.	Connection pattern	Output of the preceding FF, is connected to clock of the next FF.	There is no connection between output of preceding FF and CLK of next one.
3.	Clock input	All the FFs are not clocked simultaneously.	All FFs receive clock signal simultaneously.
4.	Propagation delay	$P.D. = n \times (t_d)$ where n is number of FFs and $t_d$ is p.d. per FF.	$P.D. = (t_d)_{FF} + (t_d)_{gate}$ It is much shorter than that of asynchronous counter.
5.	Maximum frequency of operation	Low because of the long propagation delay.	High due to shorter propagation delay.

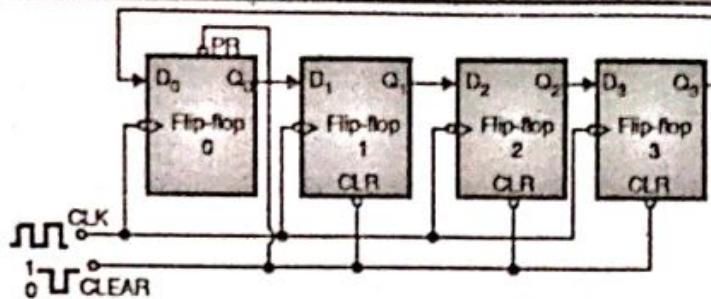
**Q. 3(b)** Draw four bit ring counter and explain its operation.

(10 Marks)

100

### **Four bit ring counter:**

Fig. 1. Q. 2(b) shows a typical application of shift registers called Ring Counter.



(C-471) Fig. 1- Q. 2(b) : A four bit ring counter

The connections reveal that they are similar to the connections for shift right operation, except for one change. Output of FF-3 is connected to data input  $D_0$  of FF-0. Ring counter is a special type of shift register.

#### Operation :

Initially a low clear (CLR) pulse is applied to all the flip-flops. Hence FF-3, FF-2 and FF-1 will reset but FF-0 will be preset. So the outputs of the shift register are :

$$Q_3 Q_2 Q_1 Q_0 = 0001.$$

Now the clear terminal is made inactive by applying a high level to it. The clock signal is then applied to all the flip-flops simultaneously. Note that all the flip-flops are negative edge triggered.

#### On the first negative going CLK edge :

As soon as the first falling edge of the clock hits, only FF-1 will be set because  $Q_0 = D_1 = 1$ . The FF-0 will reset because  $D_0 = Q_3 = 0$  and there is no change in the status of FF-2 and FF-3. Hence after the first clock pulse the outputs are

$$Q_3 Q_2 Q_1 Q_0 = 0010.$$

#### On the second falling edge of clock :

At the second falling edge of the clock, only FF-2 will be set because  $D_2 = Q_1 = 1$ . FF-1 will reset since  $D_1 = Q_0 = 0$ . There is no change in status of FF-3 and FF-0. So after the second clock pulse the outputs are,

$$Q_3 Q_2 Q_1 Q_0 = 0100.$$

Similarly after the third clock pulse the outputs are,

$$Q_3 Q_2 Q_1 Q_0 = 1000.$$

And after the fourth one the outputs are,

$$Q_3 Q_2 Q_1 Q_0 = 0001.$$

These are the outputs from where we started. Hence the operation repeats from this point onwards.

#### Number of output states :

The number of output states for a ring counter will always be equal to the number of flip-flops. So for a 4-bit ring counter the number of states is equal to 4. The operation of a four bit ring counter is summarized in Table 1- Q. 2(b).

(C-471(a)) Table 1- Q. 2(b) : Summary of operation of a ring counter

CLR	CLK	$Q_0$	$Q_1$	$Q_2$	$Q_3$
U	X	1	0	0	0
1	↓	0	1	0	0
1	↓	0	0	1	0
1	↓	0	0	0	1
1	↓	1	0	0	0

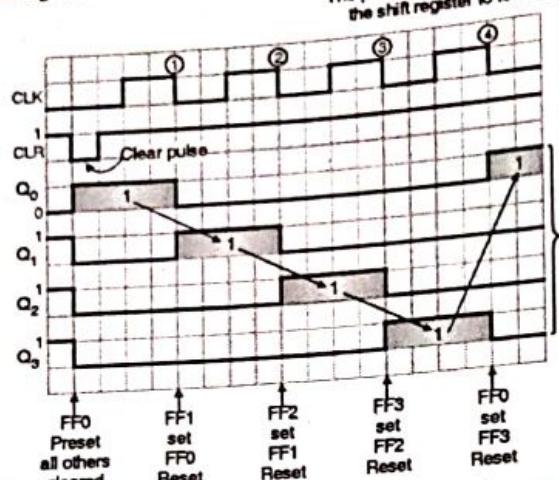
← FF-0 Preset,  
others cleared

The presetted 1  
follows a circular  
path to form a  
ring

**Waveforms for the ring counter :**

The waveforms for the 4-bit ring counter are as shown in Fig. 2- Q. 2(b). These waveforms clearly show that the presetted "1" shifts one bit per clock cycle and forms a ring. Hence the name ring counter.

The presetted "1" circulates through the shift register to form a ring



(C-1360) Fig. 2- Q. 2(b) : Waveforms of a four bit ring counter

(10 Marks)

**Q. 5(a) Design lockout free mod 10 up synchronous counter using JKMS flip flops.**

**Ans. :**

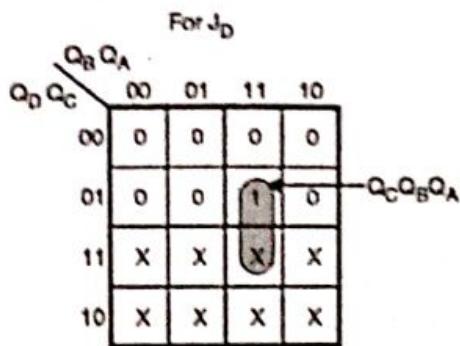
**Mod 10 up synchronous counter :**

**Step 1 : Write the circuit excitation table :**

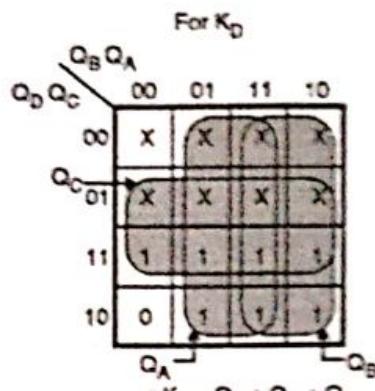
Present state				Next state				Flip flop inputs							
$Q_D$	$Q_C$	$Q_B$	$Q_A$	$Q_{D+1}$	$Q_{C+1}$	$Q_{B+1}$	$Q_{A+1}$	$J_D$	$K_D$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
0	0	0	0	0	0	0	1	0	x	0	x	0	x	1	x
0	0	0	1	0	0	1	0	0	x	0	x	1	x	x	1
0	0	1	0	0	0	1	1	0	x	0	x	x	0	1	x
0	0	1	1	0	1	0	0	0	x	1	x	x	1	x	1
0	1	0	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	0	0	0	0	x	1	0	x	0	x	x	1
1	0	1	0	0	0	0	0	x	1	0	x	x	1	0	x
1	0	1	1	0	0	0	0	x	1	0	x	x	1	x	1
1	1	0	0	0	0	0	0	x	1	x	1	0	x	0	x
1	1	0	1	0	0	0	0	x	1	x	1	0	x	x	1
1	1	1	0	0	0	0	0	x	1	x	1	x	1	0	x
1	1	1	1	0	0	0	0	x	1	x	1	x	1	0	x



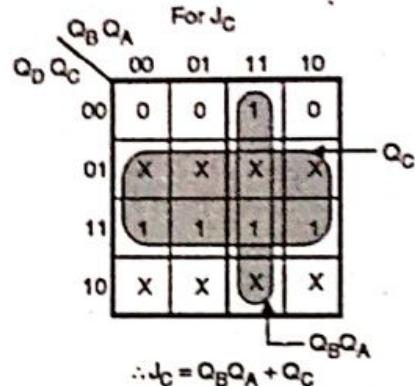
Step 2 : K-maps and simplification :



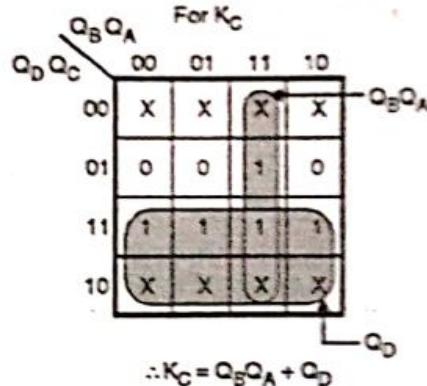
$$\therefore J_D = Q_C Q_B Q_A$$



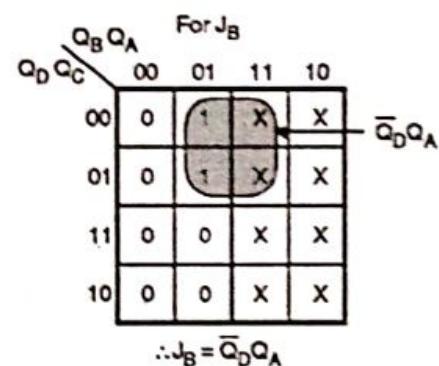
$$\therefore K_D = Q_A + Q_B + Q_C$$



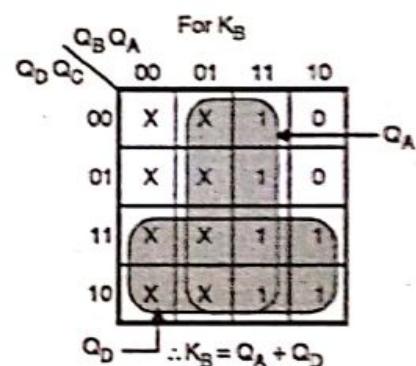
$$\therefore J_C = Q_B Q_A + Q_C$$



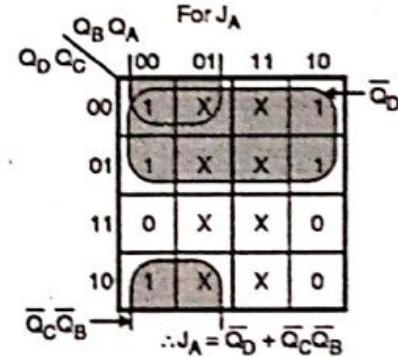
$$\therefore K_C = Q_B Q_A + Q_D$$



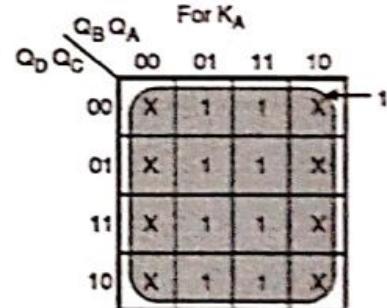
$$\therefore J_B = \bar{Q}_D Q_A$$



$$\therefore K_B = Q_A + Q_C$$



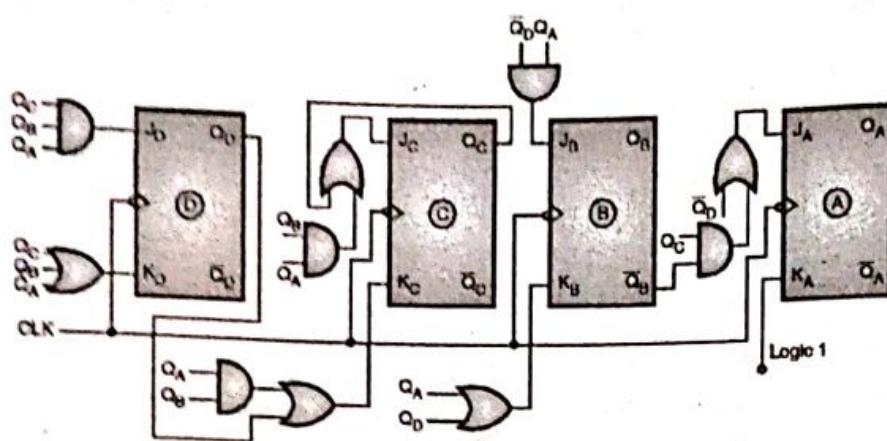
$$\therefore J_A = \bar{Q}_D + \bar{Q}_C \bar{Q}_B$$



$$\therefore K_A = 1$$

(C-3839) Fig. 1- Q. 5(a)

**Step 3 : Draw the logic circuit :**



(C-3840) Fig. 2- Q. 5(a)



Dec. 2018

**Chapter 1 : Number Systems & Codes [Total Marks - 15]**

**Q. 1(d) Evaluate following operation in BCD :**

I.  $(56)_{10} + (23)_{10}$       II.  $(48)_{10} + (26)_{10}$

(5 Marks)

**Ans. :**

I.  $(56)_{10} + (23)_{10}$

Decimal	BCD
56	0101 0110
+ 23	+ 0010 0011
	11
	<hr/>
	0111 1001
	Valid      Valid
	BCD      BCD

(C-7475) Fig. 1-Q. 1(d)

$\therefore (56)_{10} + (23)_{10} = (79)_{10}$

...Ans.

II.  $(48)_{10} + (26)_{10}$

Decimal	BCD
48	0100 1000
+ 26	+ 0010 0110
	<hr/>
	0110 1110
	Valid      Invalid
	BCD      BCD and Carry = 0

(C-7476) Fig. 2-Q. 1(d)(a)

Add 6 to the invalid BCD

0110 1110	
+	0000 0110 Add 6 for correction
	<hr/>
1 11	
	<hr/>
0111 0100	Correct result
7	4

(C-7477) Fig. 2-Q. 1(d)(b)

$\therefore (48)_{10} + (26)_{10} = (74)_{10}$

...Ans.

**Q. 2(a) Convert  $(27)_{10}$  and  $(42)_{10}$  into binary, octal, hexadecimal, excess-3 code and gray code.**

(10 Marks)

**Ans. :**

1. Conversion to binary :

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

(C-7478)

2	42	
2	21	0
2	10	1
2	5	0
2	2	1
2	1	0
	0	1

(C-7479)

Fig. 1-Q. 2(a)

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

$$\therefore (42)_{10} = (101010)_2$$

2. Conversion to octal :

8   27   LSD	8   42   LSD
8   3   3	8   5   2
0   3   MSD	0   5   MSD

(C-nan Fig. 1-Q. 2(a)(b))

$$\therefore (27)_{10} = (33)_8$$

$$\therefore (42)_{10} = (52)_8$$

3. Conversion to hex :

16   27   LSD	16   42   LSD
16   1   8	16   2   A
0   1   MSD	0   2   MSD

(C-nan Fig. 1-Q. 2(a)(c))

$$\therefore (27)_{10} = (1B)_{16}$$

$$\therefore (42)_{10} = (2A)_{16}$$

4. Conversion to Ex - 3 :

(i)  $N = (27)_{10}$ :

$$\begin{array}{r} \text{Convert to BCD:} & 0010\ 0111 \\ \text{Add } (3)_{10}: & + 0011\ 0011 \\ & \hline 1 & 111 \\ \text{XS-3 No:} & 0101\ 1010 \end{array}$$

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

(C-nan Fig. 1-Q. 2(a)(d))

(ii)  $N = (42)_{10}$ :

$$\begin{array}{r} \text{Convert to BCD:} & 0100\ 0010 \\ \text{Add } (3)_{10}: & + 0011\ 0011 \\ & \hline 1 \\ \text{XS-3 No:} & 0111\ 0101 \end{array}$$

(C-nan Fig. 1-Q. 2(a)(e))

$$(42)_{10} = (101010)_2$$

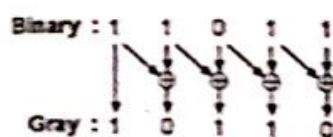
5. Conversion to gray :

(i)  $N = (27)_{10}$

Step 1 : Conversion to binary :

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

Step 2 : Binary to gray :



(C-nan Fig. 1-Q. 2(a)(f))

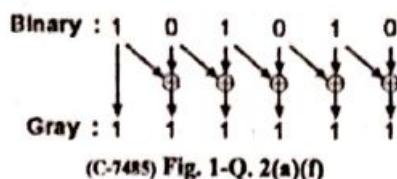
$$\therefore (27)_{10} = (10110)_{\text{gray}}$$

(ii)  $N = (42)_{10}$ :

Step 1 : Decimal to binary :

$$(42)_{10} = (101010)_2$$

**Step 2 : Binary to gray :**



$$\therefore (42)_{10} = (111111)_\text{gray}$$

...Ans.

### Chapter 3 : Logic Minimization and Reduction Techniques [Total Marks - 20]

**Q. 4(a) Simplify the expression in POS form for given function and realize it with basic gates.**

$$F(A, B, C, D) = \sum m(0, 4, 6, 7, 10, 12, 14) + d(2, 13)$$

(10 Marks)

**Ans. :**

**Given :**  $F(A, B, C, D) = \sum m(0, 4, 6, 7, 10, 12, 14) + d(2, 13)$

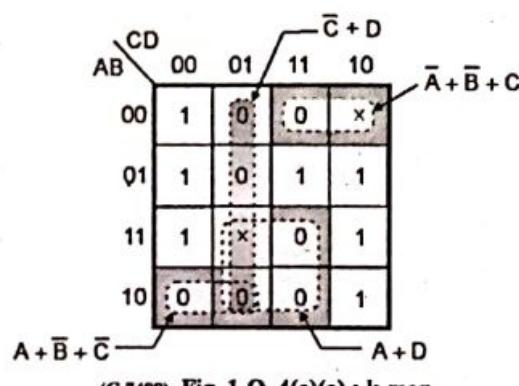
**To do :** Realization in POS.

**1. SOP to POS conversion :**

The given SOP expression can be converted into equivalent POS expression as follows :

$$F(A, B, C, D) = \sum \pi(1, 3, 5, 8, 9, 11, 15) + d(2, 13)$$

**2. k-map and simplification :**

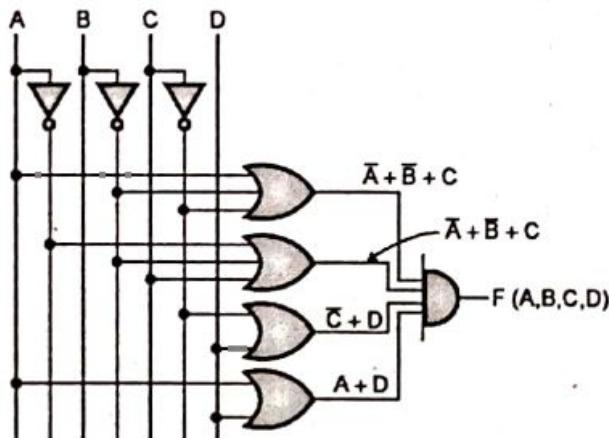


(C-7489) Fig. 1-Q. 4(a)(a) : k-map

Simplified expression is,

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

**3. Logic diagram :**



(C-7490) Fig. 1-Q. 4(a)(b) : Logic diagram

**Q. 5(b) Simplify the following four variable Boolean function using Quine-McCluskey technique.**

$$F(A, B, C, D) = \sum m(0, 2, 3, 6, 7, 8, 10, 12, 13)$$

(10 Marks)

**Ans. :**

Given :  $F(A, B, C, D) = \sum m(0, 2, 3, 6, 7, 8, 10, 12, 13)$

Step 1 : Group the minterms according to number of 1's :

Group	Minterm	Binary representation				Prime implicant
		A	B	C	D	
0	0	0	0	0	0	
1	2	0	0	1	0	
	8	1	0	0	0	
2	3	0	0	1	1	
	6	0	1	1	0	
	10	1	0	1	0	
	12	1	1	0	0	
3	7	0	1	1	1	
	13	1	1	0	1	

Step 2 : Group the minterms into groups of two :

Group	Minterm pair	Binary representation				Prime implicant
		A	B	C	D	
0	0-2	0	0	-	0	✓
	0-8	-	0	0	0	✓
1	2-3	0	0	1	-	✓
	2-6	0	-	1	0	✓
	2-10	-	0	1	0	$\bar{B}CD$
	8-10	1	0	-	0	$A\bar{B}D$
	8-12	1	-	0	0	$A\bar{C}D$
2	3-7	0	-	1	1	✓
	6-7	0	1	1	-	✓
	12-13	1	1	0	-	$A\bar{B}\bar{C}$

Step 3 : Group the minterms into groups of four :

Group	Minterm quads	Binary representation				Prime implicant
		A	B	C	D	
0	0-2-8-10	-	0	-	0	$\bar{B}\bar{D}$
	0-8-2-10	-	0	-	0	
1	2-3-6-7	0	-	1	-	$\bar{A}C$
	2-6-3-7	0	-	1	-	

Step 4 : Prepare the table of prime implicants :

(C-7496)

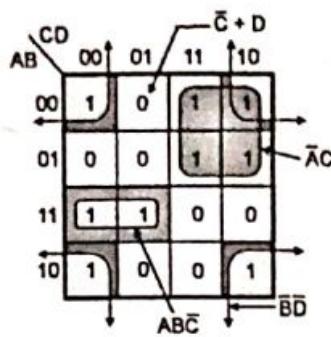
Prime Implicants	Decimal number	Given minterms								
		0	2	3	6	7	8	10	12	13
$\bar{B}CD$	2,10	x						x		
$A\bar{B}D$	8,10						x	x		
$A\bar{C}D$	8,12						x		x	
$A\bar{B}\bar{C}$	12,13								x	x
$\bar{B}\bar{D}$	0,2,8,10	x	x				x	x		x
$\bar{A}C$	2,3,6,7		x	x	x	x				



- The encircled crosses represent the EPIS. They cover all the given minterms.

$$\therefore F(A, B, C, D) = ABC + \bar{B}\bar{D} + \bar{A}C$$

**Step 5 : Croscheck using K-map :**



(C-7497) Fig. 1-Q. 5(b)(d)

$$\therefore F(A, B, C, D) = \bar{A}C + \bar{B}\bar{D} + ABC$$

- This is same as the result obtained with Quine Mc-Cluskey method.

### Chapter 4 : Arithmetic Circuits [Total Marks - 15]

**Q. 1(a) Design and implement full subtractor using logic gates.**

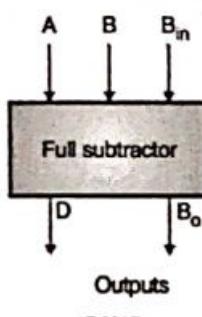
(5 Marks)

**Ans. :**

**Full Subtractor :**

- The full subtractor is a combinational circuit with three inputs A, B and  $B_{in}$  and two outputs D and  $B_o$ .

Inputs



Outputs

(C-3515)

- A is the minuend, B is subtrahend,  $B_{in}$  is the borrow produced by the previous stage, D is the difference output and  $B_o$  is the borrow output.

**Truth table :**

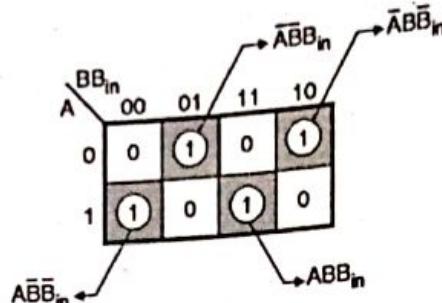
- The truth table for full subtractor is shown in Table 1-Q. 1(a).

Table 1-Q. 1(a) : Truth table for a full subtractor

Inputs			Outputs	
A (Minuend)	B (Subtrahend)	$B_{in}$ Previous borrow	$(A - B - B_{in})$	$B_o$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

**K-maps and simplifications :**

- K-maps for D and  $B_o$  outputs are shown in Figs. 1-Q. 1(a)(a) and (b).

**For difference output :**

(a) K-map for D

(C-360) Fig. 1-Q. 1(a)

$$\therefore D = \overline{A} \overline{B} B_{in} + \overline{A} B \overline{B}_{in} + A \overline{B} B_{in} + A B \overline{B}_{in}$$

**Simplification for difference output :**

From Fig. 1-Q. 1(a)(a),

$$\begin{aligned} D &= \overline{A} \overline{B} B_{in} + \overline{A} B \overline{B}_{in} + A \overline{B} B_{in} + A B \overline{B}_{in} \\ &= B_{in} \underbrace{(\overline{A} \overline{B} + A B)}_{\text{EX-NOR}} + \overline{B}_{in} \underbrace{(\overline{A} B + A \overline{B})}_{\text{EX-OR}} \end{aligned} \quad (\text{C-6376})$$

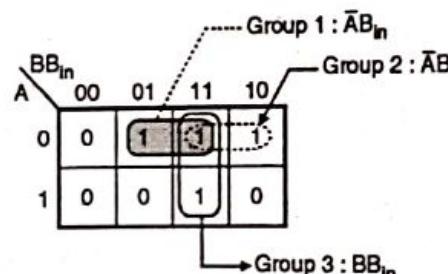
$$\therefore D = B_{in} (\overline{A} \oplus B) + \overline{B}_{in} (A \oplus B)$$

Let  $A \oplus B = C$ ,

$$\therefore D = B_{in} \overline{C} + \overline{B}_{in} C = B_{in} \oplus C$$

$$\therefore D = B_{in} \oplus A \oplus B$$

...(1)

**For borrow output :**(C-360) Fig. 1-Q. 1(a) : K-map for B<sub>o</sub>

$$\therefore B_o = \overline{A} B_{in} + \overline{A} B + B B_{in}$$

**Simplification for borrow output :**

From Fig. 1-Q. 1(a)(b),

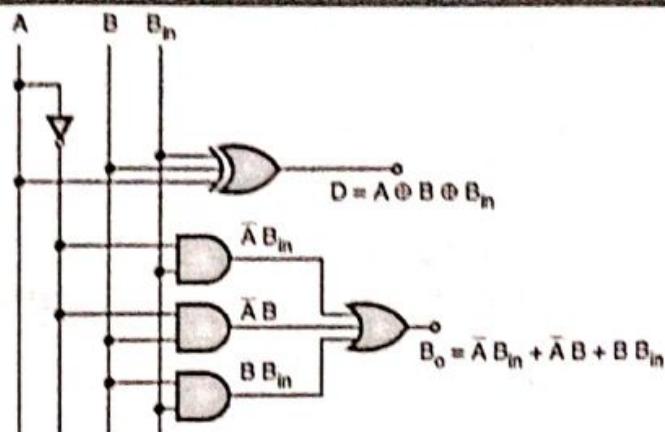
$$B_o = \overline{A} B_{in} + \overline{A} B + B B_{in}$$

...(2)

- No further simplification is possible.

**Logic diagram for full subtractor :**

- Logic diagram for the full subtractor is shown in Fig. 2-Q. 1(a). This has been drawn by using the Boolean equations of (1) and (2).



(C-361) Fig. 2-Q. 1(a) : Logic diagram for a full subtractor

**Q. 3(a)** Design a combinational logic circuit with four input variables that will produce logic 1 output when input is greater than 9. (10 Marks)

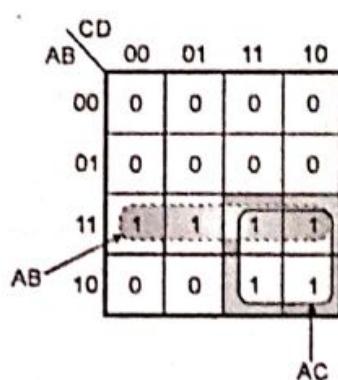
**Ans. :**

**Step 1 : Write truth table :**

(C-7486)

Inputs				Output Y
A	B	C	D	
0	0	0	0	0
0	0	0	1	0
⋮	⋮	⋮	⋮	⋮
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

**Step 2 : Write a K-map and simplify :**

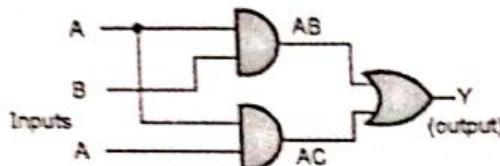


(C-7487) Fig. 1-Q. 3(a)(a) : K-map

∴ Simplified equation is

$$Y = AB + AC$$

Step 3 : Logic diagram :



(C-7488) Fig. 1-Q. 3(a)(b) : Logic diagram

### Chapter 5 : Multiplexer and Demultiplexer [Total Marks - 15]

Q. 1(c) Design a circuit using 2 : 1 MUX to implement 2 input NAND gate.

(5 Marks)

Ans. :

Given : A 2 : 1 MUX.

To do : Implement a 2 input NAND gate :

1. Write the truth table :

(C-7472) Table 1-Q. 1(c)(a) : Truth table of a 2 input NAND gate

Inputs A    B	Output Y
0    0	1
0    1	1
1    0	1
1    1	0

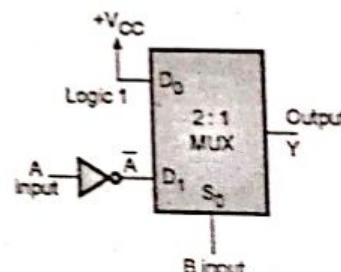
$$\therefore Y = \bar{A}\bar{B} + \bar{A}B + A\bar{B} = \sum m(0, 1, 2)$$

2. Write the design table :

(C-7473) Table 2-Q. 1(c)(b) : Design table

Inputs	D <sub>0</sub>	D <sub>1</sub>
$\bar{A}$	0	1
A	2	3
Input to MUX	1	$\bar{A}$

3. Implementation :



(C-7474) Fig. 1-Q. 1(c) : A 2 input NAND gate using 2 : 1 MUX

Q. 5(a) Implement the following expression using a single 8 : 1 multiplexer.

$$F(A, B, C, D) = \sum m(0, 2, 4, 6, 8, 10, 12, 14)$$

(10 Marks)

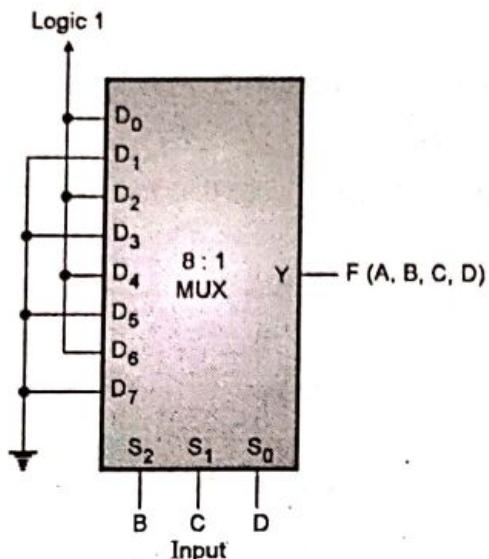
Ans. :

Step 1 : Write the design table :

(C-7491) Table 1-Q. 5(a) : Design table

Input	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>
$\bar{A}$	0	1	2	3	4	5	6	7
A	9	10	11	12	13	14	15	0
Input to MUX	1	0	1	0	1	0	1	0

easy-solutions

**Step 2 : Implementation :**


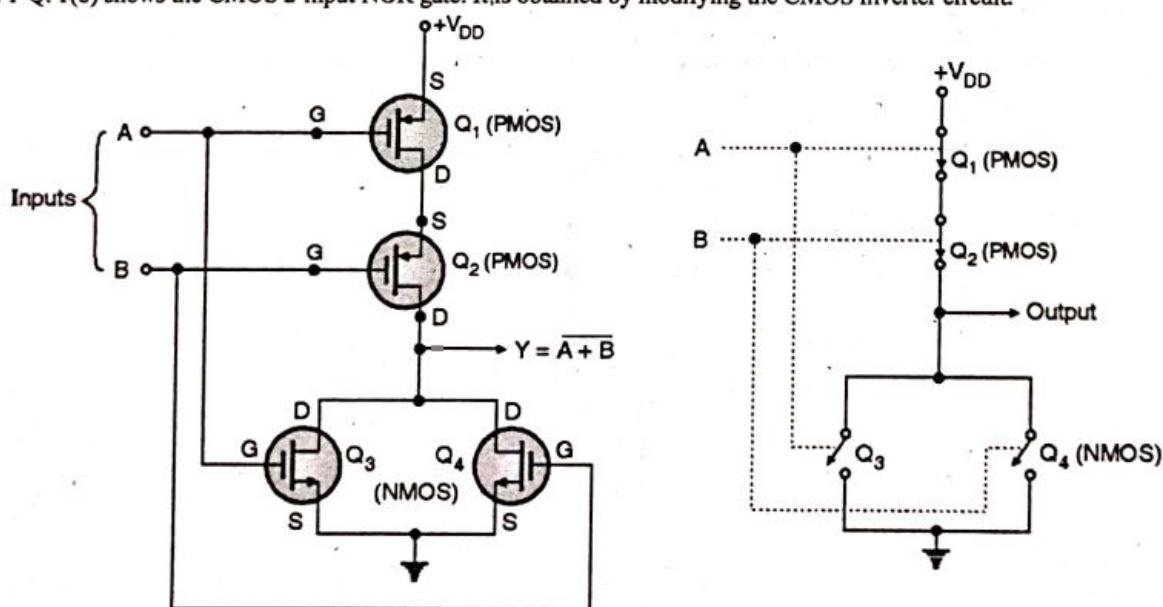
(C-7492) Fig. 1-Q. 5(a) : Implementation

**Chapter 6 : Logic Families [Total Marks - 15]**
**Q. 1(b) Explain the working of a two - inputs CMOS NOR gate with a neat diagram.**

(15 Marks)

**Ans. :**
**CMOS NOR Gate :**

- Fig. 1-Q. 1(b) shows the CMOS 2-input NOR gate. It is obtained by modifying the CMOS inverter circuit.



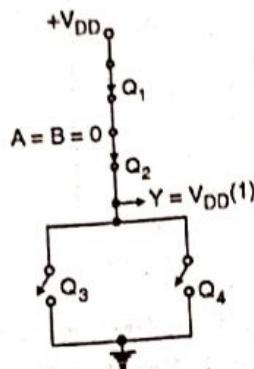
(C-1055) Fig. 1-Q. 1(b)

- $Q_1$  and  $Q_2$  are p-channel MOSFETs connected in series and  $Q_3, Q_4$  are n-channel MOSFETs connected in parallel with each other.
- Input A is connected to the gates of  $Q_1$  and  $Q_3$  while input B is connected to the gates of  $Q_2$  and  $Q_4$ .

Output  $Y = V_{DD} = 1$  if  $Q_1$  and  $Q_2$  both ON and  $Q_3, Q_4$  both OFF. And output  $Y = 0$  if  $Q_1$  or  $Q_2$  or both are OFF and  $Q_3$  or  $Q_4$  both are ON.

**Operation :****1. With  $A = B = 0$  :**

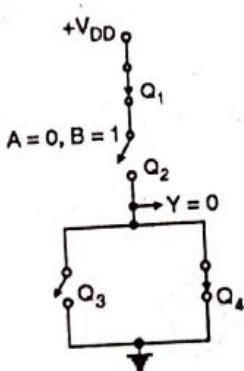
- $V_{GS1} = -V_{DD}$ ,  $V_{GS2} = -V_{DD}$ . Hence  $Q_1$  and  $Q_2$  will be ON.
- $V_{GS3} = 0$ ,  $V_{GS4} = 0$ . So  $Q_3$  and  $Q_4$  will be OFF.
- The equivalent circuit for this mode is shown in Fig. 2-Q. 1(b)(a) which shows that output  $Y = V_{DD}$  (logical 1).
- ∵ For  $A = 0$ ,  $B = 0$ , output  $Y = 1$ .

(a) For  $A = B = 0$ 

(C-1056) Fig. 2-Q. 1(b) : Equivalent circuits

**2. With  $A = 0$ ,  $B = 1$  :**

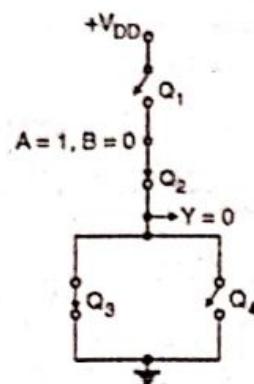
- If  $A = 0$  and  $B = 1$ , then  $Q_1$  will remain ON, but  $Q_2$  will turn OFF.
- $Q_3$  will remain OFF but  $Q_4$  will be ON.
- The equivalent circuit for this mode is shown in Fig. 2-Q. 1(b)(b) which shows that output  $Y = 0$  Volt (logical 0).
- Thus  $Y = 0$  for  $A = 0$  and  $B = 1$ .

(b) For  $A = 0$ ,  $B = 1$ 

(C-1056) Fig. 2-Q. 1(b) : Equivalent circuits

**3. With  $A = 1$ ,  $B = 0$  :**

- If  $A = 1$  and  $B = 0$  then  $Q_1$  will be OFF and  $Q_2$  will turn ON.
- $Q_3$  will be turned ON but  $Q_4$  will turn OFF.
- Equivalent circuit of this mode is shown in Fig. 2-Q. 1(b)(c) which shows that  $Y = 0$  Volt (logical 0).
- Thus  $Y = 0$  for  $A = 1$ ,  $B = 0$ .


 (c) For  $A = 1, B = 0$ 

(C-1056) Fig. 2-Q. 1(b) : Equivalent circuits

**4. For  $A = B = 1$ :**

- If A and B both are high (1), then  $Q_1$  and  $Q_2$  both will be OFF.
- $Q_3$  and  $Q_4$  both will be ON. Hence output  $Y = 0$ .
- The equivalent circuit for this mode is shown in Fig. 2-Q. 1(b)(d) and Table 1-Q. 1(b) summarizes the operation.

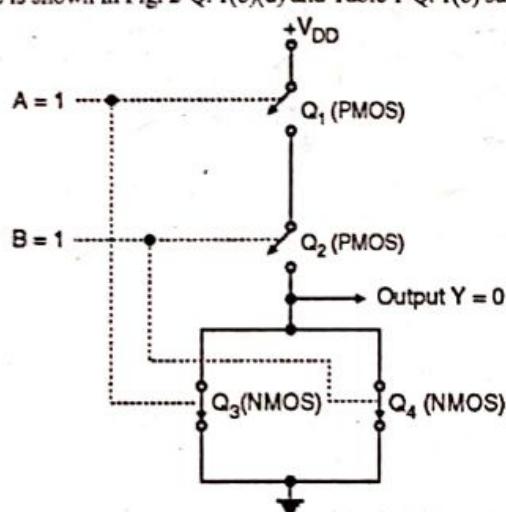

 (C-1057) Fig. 2-Q. 1(b)(d) : Equivalent circuit for  $A = B = 1$ 

Table 1-Q. 1(b): Summary of operation

Inputs		Transistors				Output $Y$
A	B	$Q_1$	$Q_2$	$Q_3$	$Q_4$	
0	0	ON	ON	OFF	OFF	$V_{DD}(1)$
0	1	ON	OFF	OFF	ON	0
1	0	OFF	ON	ON	OFF	0
1	1	OFF	OFF	ON	ON	0

Q. 6(b) Explain interfacing of a TTL gate driving CMOS gates and vice versa.

(10 Marks)

Ans. :

**TTL to CMOS Interfacing :**

- When we interface different types of ICs, it is necessary to check whether the driving IC is capable of meeting the current and voltage requirements of the load IC or not.
- For example, Table 1-Q. 6(b) reveals that the output current capability of TTL ICs is much higher than the input current values of CMOS ICs. Therefore there is no problem for a TTL IC to drive CMOS as far as current is concerned.

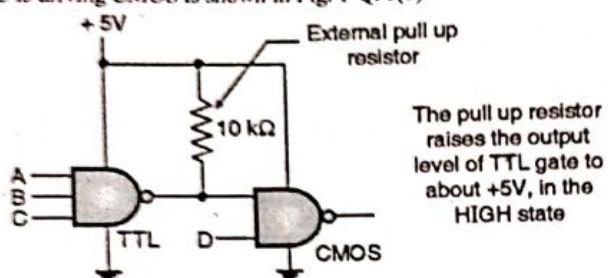


- But there is a problem when we compare the voltage levels of TTL and CMOS. Because  $V_{OH(min)}$  of a TTL series is very low as compared with  $V_{IH(max)}$  required for the CMOS series like 400B, 74HC or 74AC.
- In such situations, something has to be done to increase the level of TTL output voltage to an acceptable voltage level for CMOS.

(C-4253) Table 1-Q. 6(b) : Input output currents for standard devices with a supply voltage of 5 V

Parameter	CMOS		TTL		
	4000 B	74HC/HCT	74	74 LS	74 AB
$I_{IH(max)}$	1 $\mu$ A	1 $\mu$ A	40 $\mu$ A	20 $\mu$ A	20 $\mu$ A
$I_{IL(max)}$	1 $\mu$ A	1 $\mu$ A	1.6 mA	0.4 mA	0.5 mA
$I_{OH(max)}$	0.4 mA	4 mA	0.4 mA	0.4 mA	2 mA
$I_{OL(max)}$	0.4 mA	4 mA	16 mA	8 mA	20 mA

- The situation in which a TTL IC is driving CMOS is shown in Fig. 1-Q. 6(b).



(C-1065) Fig. 1-Q. 6(b) : TTL driving CMOS

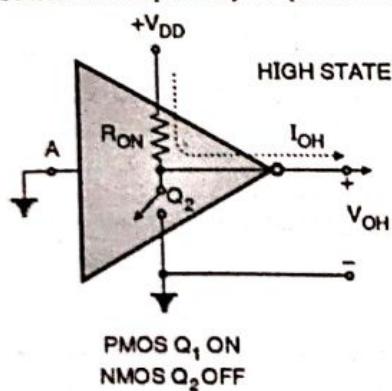
- Note the presence of pull up resistor at the output of the TTL gate. Due to this resistor the TTL output will rise to approximately +5 V in its HIGH state.
- This will provide the sufficient voltage level at the input of the CMOS gate.
- Such a pull up resistance is not required if the CMOS gate belongs to 74 HCT or 74 ACT family, because these families can accept the TTL outputs directly. In other words they are TTL compatible CMOS families.

#### CMOS to TTL Interface :

- Fig. 2-Q. 6(b)(a) shows the equivalent output circuit of a CMOS inverter in the HIGH state output, whereas Fig. 2-Q. 6(b)(b) shows its equivalent in the LOW state output.

#### Equivalent circuit in HIGH state :

- In the HIGH state of output, the P-channel MOSFET  $Q_1$  will be ON, so it has been replaced by the ON state resistance  $R_{ON}$ . Whereas the N-channel MOSFET  $Q_2$  will be OFF. So it has been replaced by an open switch as shown in Fig. 2-Q. 6(b)(a).



(a) Equivalent circuit of a CMOS inverter for HIGH state

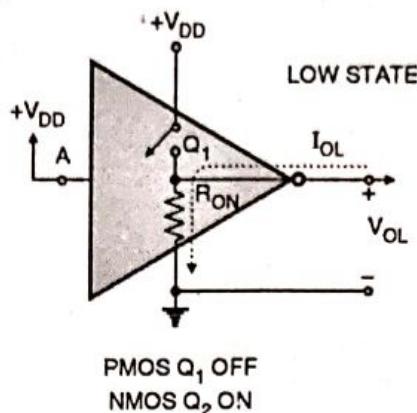
(C-1071) Fig. 2-Q. 6(b)



- Therefore the CMOS output is equivalent to a  $V_{DD}$  source with a source resistance of  $R_{ON}$ .

#### Equivalent circuit in LOW state :

- The equivalent circuit of a CMOS inverter in its low state of output is shown in Fig. 2-Q. 6(b)(b). In this state, the P-MOSFET  $Q_1$  is OFF and represented by an open switch.



(b) Equivalent circuit of a CMOS inverter for low state

(C-1071) Fig. 2-Q. 6(b)

- Whereas the N-channel MOSFET is ON. So it has been replaced by resistance  $R_{ON}$ . So the CMOS inverter now acts as a low resistance connected to ground and sinks current.
- Table 2-Q. 6(b) presents the various voltage and current levels for the standard CMOS (4000B) and TTL (74) series for comparison.

(C-6259) Table 2-Q. 6(b)

Sr. No.	Parameters for driving gate (CMOS) of 4000 B series	Parameters for the load gate (TTL) of 74 series
1.	$V_{OH(\min)} = 4.95V$	$V_{IH(\min)} = 2V$
2.	$V_{OL(\max)} = 0.05V$	$V_{IL(\max)} = 0.8V$
3.	$I_{OH(\max)} = 0.4mA$	$V_{IH(\max)} = 40\mu A$
4.	$I_{OL(\max)} = 0.4mA$	$I_{IL(\max)} = 1.6mA$

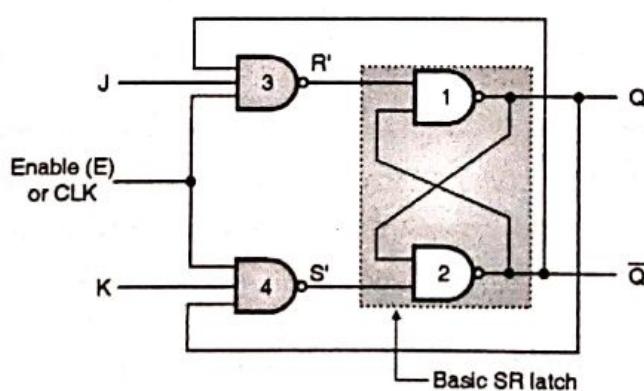
For low state output                                          For high state output

#### Chapter 7 : Latches and Flip Flops [Total Marks - 20]

Q. 3(b) Draw a circuit diagram of clocked J-K flip - flop using NAND gates with truth table. What is race around condition and how does it get eliminated ? (10 Marks)

Ans. :

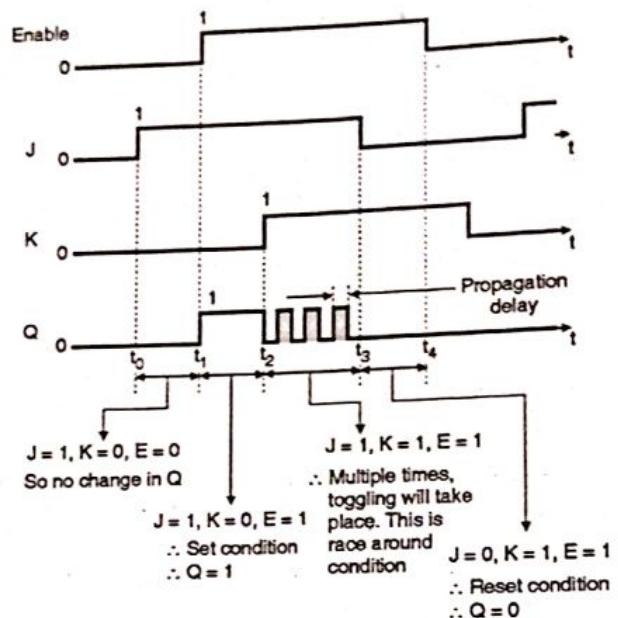
Clocked J-K flip:



(C-3420) Fig. 1-Q. 3(b) : Clocked J-K flip

**Race Around Condition in JK Latch :**

- The "Race Around Condition" that we are going to explain occurs when  $J = K = 1$  i.e. when the latch is in the toggle mode.
- Refer Fig. 2-Q. 3(b) which shows the waveforms for the various modes, when a rectangular waveform is applied to the "Enable" input.



(C-586) Fig. 2-Q. 3(b) : Waveforms for various modes of a JK latch

**Interval  $t_0 - t_1$** 

- During this interval  $J = 1, K = 0$  and  $E = 0$ .
- Hence the latch is disabled and there is no change in  $Q$ .

**Interval  $t_1 - t_2$** 

- During this interval  $J = 1, K = 0$  and  $E = 1$ .
- Hence this is a set condition and  $Q$  becomes 1.

**Interval  $t_2 - t_3$  : Race around**

- At instant  $t_2$ ,  $J = K = 1$  and  $E = 1$  Hence the JK latch is in the toggle mode and  $Q$  becomes low (0) and  $\bar{Q} = 1$ .
- These changed outputs get applied at the inputs of NAND gates 3 and 4 of the JK latch. Thus the new inputs to Gates 3 and 4 are:
  - NAND - 3 :  $J = 1, E = 1, \bar{Q} = 1$ .
  - NAND - 4 :  $K = 1, E = 1, Q = 0$ .
- Hence  $R'$  will become 0 and  $S'$  will become 1.
- Therefore after a time period corresponding to the propagation delay, the  $Q$  and  $\bar{Q}$  outputs will change to,  $Q = 1$  and  $\bar{Q} = 0$ .
- These changed output again get applied to the inputs of NAND-3 and 4 and the outputs will toggle again.
- Thus as long as  $J = K = 1$  and  $E = 1$ , the outputs will keep toggling indefinitely as shown in Fig. 2-Q. 3(b).
- This multiple toggling in the J-K latch is called as Race Around condition. It must be avoided.

**Interval  $t_3 - t_4$** 

- During this interval  $J = 0, K = 1$  and  $E = 1$ . Hence it is the reset condition.
- So  $Q$  becomes zero.

Q. 4(b) Convert the followings :

- SR flip flop to JK flip flop.
- JK flip-flop to D flip-flop

(10 Marks)

Ans.:

### SR Flip Flop to JK Flip Flop :

Step 1 : Write the truth table for SR to JK :

- The truth table for SR to JK flip flop conversion is shown in Table 1-Q. 4(b).

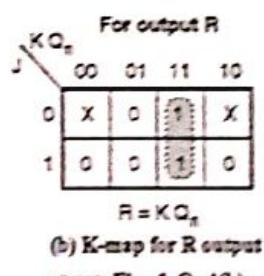
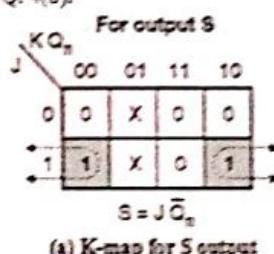
(C-419) Table 1-Q. 4(b) : Truth table for SR to JK FF conversion

		Inputs		Outputs	
J	K	Present state $Q_n$	Next state $Q_{n+1}$	S	R
0	0	0	0	0	X
0	1	0	0	0	X
1	0	0	1	1	0
1	1	0	1	1	0
0	1	1	0	0	1
1	1	1	0	0	1
0	0	1	1	X	0
1	0	1	1	X	0

↓ Excitation table of JK FF ↓  
↓ Excitation table of SR FF ↓

Step 2 : K maps and simplification :

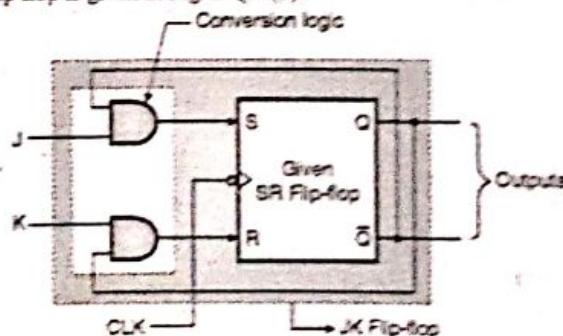
- K maps for S and R outputs are shown in Fig. 1-Q. 4(b).



(C-419) Fig. 1-Q. 4(b)

Step 3 : Logic diagram :

- The logic diagram of SR to JK flip flop is given in Fig. 2-Q. 4(b).



(C-420) Fig. 2-Q. 4(b) : SR to JK flip flop conversion



**JK Flip Flop to D Flip Flop Conversion :**

Step 1 : Write the truth table for JK to D conversion :

The truth table is as follows :

(C-639) Table 2-Q. 4(b) : Truth table for JK to D

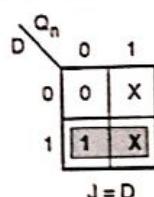
flip flop conversion			Outputs	
Inputs		Next state $Q_{n+1}$	J	K
D	Previous state $Q_n$			
0	0	0	0	X
1	0	1	1	X
0	1	0	X	1
1	1	1	X	0

↓ Excitation table of D FF

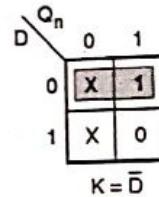
↓ Excitation table of JK FF

Step 2 : K maps and simplification :

For J output



For K output



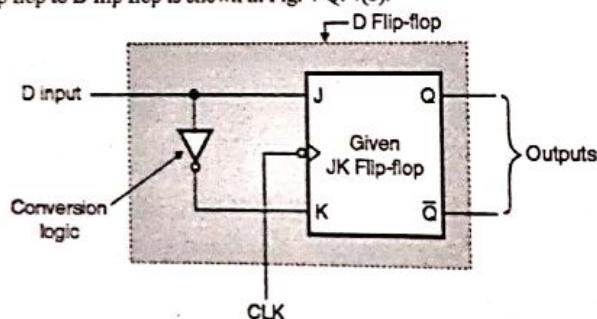
(a) K map for J output

(b) K map for K output

(C-623) Fig. 3-Q. 4(b)

Step 3 : Draw the logic diagram :

- The logic diagram for JK flip flop to D flip flop is shown in Fig. 4-Q. 4(b).



(C-624) Fig. 4-Q. 4(b): Logic diagram for conversion from JK FF to D FF

### Chapter 9 : Counters [Total Marks - 20]

**Q. 2(b)** Draw a neat circuit diagram of four bit twisted ring counter with initial state 0000 and relevant output waveforms.

(10 Marks)

Ans. :

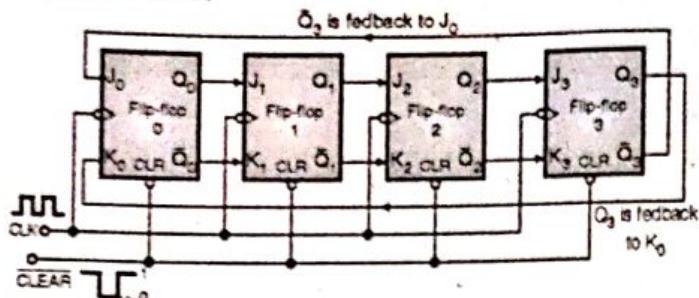
**Four bit twisted ring counter :**

- In the ring counter the outputs of FF-3 were connected directly to the inputs of FF-0 i.e.  $Q_3$  to  $J_0$ ,  $\bar{Q}_3$  to  $K_0$ .
- Instead if the outputs are cross coupled to the inputs i.e. if  $Q_3$  is connected to  $K_0$  and  $\bar{Q}_3$  is connected to  $J_0$  then the circuit is called as twisted ring counter or Johnson's counter.

**easy-solutions**



- The Johnson's counter is shown in Fig. 1-Q. 2(b).



(C-1362) Fig. 1-Q. 2(b) : Twisted ring counter or Johnson counter

- All the flip-flops are negative edge triggered, and clock pulses are applied to all of them simultaneously.
- The clear inputs of all the flip-flops are connected together and connected to an external clear signal. Note that all these clear inputs are active low inputs.

#### Operation :

- Initially a short negative going pulse is applied to the clear input of all the flip-flops. This will reset all the flip-flops. Hence initially the outputs are,  $Q_3 Q_2 Q_1 Q_0 = 0000$ .
- But  $\bar{Q}_3 = 1$  and since it is coupled to  $J_0$  it is also equal to 1.  
 $\therefore J_0 = 1$  and  $K_0 = 0$  .... Initially

#### On the first falling edge of clock pulse :

- As soon as the first negative edge of clock arrives, FF-0 will be set. Hence  $Q_0$  will become 1.
- But there is no change in the status of any other flip-flop.
- Hence after the first negative going edge of the clock the flip-flop outputs are,  
 $Q_3 Q_2 Q_1 Q_0 = 0001$

#### On the second negative going clock edge :

- Before the second negative going clock edge,  $Q_3 = 0$  and  $\bar{Q}_3 = 1$ . Hence  $J_0 = 1$  and  $K_0 = 1$ . Also  $Q_0 = 1$ . Hence  $J_1 = 1$ .
- Hence as soon as the second falling clock edge arrives, FF-0 continues to be in the set mode and FF-1 will now set. Hence  $Q_1$  will become 1 and  $\bar{Q}_1 = 0$ .
- There is no change in the status of any other flip-flop.
- Hence after the second clock edge the outputs are,  
 $Q_3 Q_2 Q_1 Q_0 = 0011$ .
- Similarly after the third clock pulse, the outputs are,  
 $Q_3 Q_2 Q_1 Q_0 = 0111$ .
- And after the fourth clock pulse, the outputs are,  
 $Q_3 Q_2 Q_1 Q_0 = 1111$ .

Note that now  $\bar{Q}_3 = 0$  i.e.  $J_0 = 0$  and  $K_0 = 1$ .

- Hence as soon as the fifth negative going clock pulse strikes, FF-0 will reset. But the outputs of the other flip-flops will remain unchanged. So after the fifth clock pulse, the outputs are,  
 $Q_3 Q_2 Q_1 Q_0 = 1110$  ..... after the 5<sup>th</sup> clock pulse
- This operation will continue till we reach the all zero output state. (i.e.  $Q_3 Q_2 Q_1 Q_0 = 0000$ ).
- The operation of Johnson's counter is summarised in Table 1-Q. 2(b).

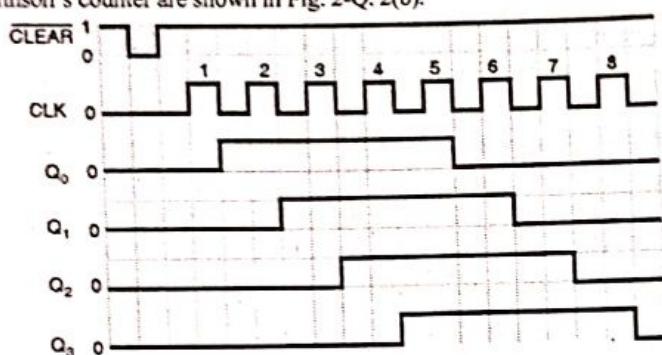


(C-6295) Table 1-Q. 2(b) : Summary of operation of Johnson's counter

CLEAR	CLK	$Q_3$	$Q_2$	$Q_1$	$Q_0$	State number	Decimal equivalent
Initially		0	0	0	0	1	0
1	↓	0	0	0	1	2	1
1	↓	0	0	1	1	3	3
1	↓	0	1	1	1	4	7
1	↓	1	1	1	1	5	15
1	↓	1	1	1	0	6	14
1	↓	1	1	0	0	7	12
1	↓	1	0	0	0	8	8
1	↓	0	0	0	0	1	0

**Waveforms for Johnson's counter :**

- The waveforms for a 4-bit Johnson's counter are shown in Fig. 2-Q. 2(b).



(C-875) Fig. 2-Q. 2(b): Waveforms of Johnson counter

**Q. 6(a) Design a Mod-5 synchronous up counter using T flip-flop. Design using minimal cost approach. (10 Marks)**

**Ans. :**

**Step 1 : Decide number of FFs :**

- Since the number of states is 5 we need to use 3 flip-flops.

**Step 2 : Excitation tables :**

- The excitation table of a T flip-flop is shown in Table 1-Q. 6(a).

Table 1-Q. 6(a) : Excitation table of a T flip-flop

Present state $Q_n$	Next state $Q_{n+1}$	T
0	0	0
0	1	1
1	0	1
1	1	0

**Circuit excitation table :**

- The circuit excitation table is shown in Table 1-Q. 6(a).
- Refer to the shaded portion of the circuit excitation table. This is nothing but the excitation table of FF-C. The  $T_C$  values are decided based on  $Q_B$  and  $Q_{B+1}$ .
- Similarly the entries for  $T_B$  are based on  $Q_B$  and  $Q_{B+1}$  whereas those for  $T_A$  are based on  $Q_A$  and  $Q_{A+1}$ .

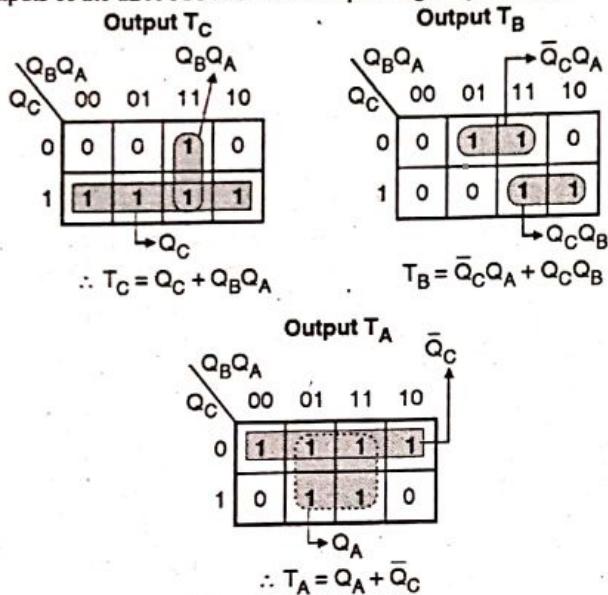


Table 2-Q. 6(a) : Circuit excitation table

Present state			Next state			Flip-flop inputs		
$Q_C$	$Q_B$	$Q_A$	$Q_{C+1}$	$Q_{B+1}$	$Q_{A+1}$	$T_C$	$T_B$	$T_A$
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	0	0	0	1	0	0
1	0	1	0	0	0	1	0	1
1	1	0	0	0	0	1	1	0
1	1	1	0	0	0	1	1	1

## Step 3 : K-maps and simplifications :

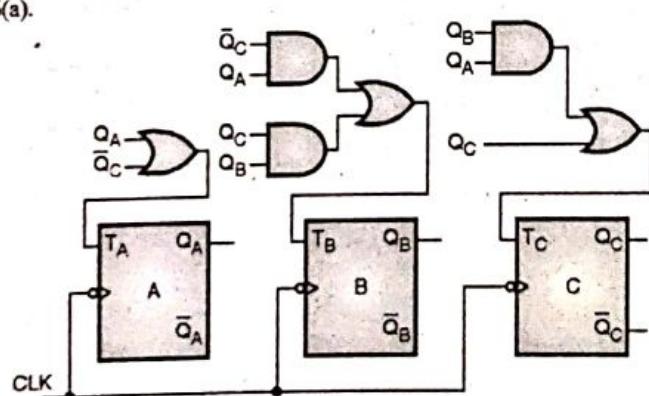
- K-maps for the  $T_A$ ,  $T_B$  and  $T_C$  inputs of the three FFs and the corresponding simplified equations are shown in Fig. 1-Q. 6(a).



(C-853) Fig. 1-Q. 6(a) : K-maps and simplifications

## Step 4 : Draw the logic diagram :

- The logic diagram of MOD-5 synchronous counter using T FFs is shown in Fig. 2-Q. 6(a).



(C-854) Fig. 2-Q. 6(a) : MOD-5 synchronous counter using T flip-flops



May, 2019

## Chapter 1 : Number Systems &amp; Codes [Total Marks - 15]

(5 Marks)

Q. 1(ii) Convert  $(73.301)_{10}$  into binary, octal, Hexadecimal and BCD equivalent.  
Ans. :

Step 1 : Convert decimal to binary :

Convert the Integer part :

2	73
2	36
2	18
2	9
2	4
2	2
2	1
0	1

$$\therefore (73)_{10} = (1001001)_2$$

Convert the fractional part :

Decimal Base Product Carry  
fraction

$0.301 \times 2 = 0.602$	0
$0.602 \times 2 = 1.204$	1
$0.204 \times 2 = 0.408$	0
$0.408 \times 2 = 0.816$	0
$0.816 \times 2 = 1.632$	1

$$\therefore (0.301)_{10} = (0.01001)_2$$

MSB

LSB

(C-7969) Fig. 1-Q. 1(ii)

...Ans.

$$\therefore (73.301)_{10} = (0.01001)_2$$

Step 2 : Convert decimal to octal :

Convert the Integer part :

8	73
8	9
8	1
0	1

$$\therefore (73)_{10} = (111)_8$$

Convert the fractional part :

Decimal Base Product Carry  
fraction

$0.301 \times 8 = 2.408$	2
$0.408 \times 8 = 3.264$	3
$0.264 \times 8 = 2.112$	2
$0.112 \times 8 = 0.896$	0
$0.896 \times 8 = 7.168$	7

$$\therefore (0.301)_{10} = (0.23207)_8$$

MSD

LSD

(C-7970) Fig. 1-Q. 1(ii)(a)

...Ans.

$$\therefore (73.301)_{10} = (111.23207)_8$$

Step 3 : Convert decimal to hexadecimal :

Convert the Integer :

16	73
16	4
0	4

$$\therefore (73)_{10} = (49)_{16}$$

Convert the fractional Part :

Decimal Base Product Carry Hex

$0.301 \times 16 = 4.816$	4	4	MSD
$0.816 \times 16 = 13.056$	13	C	
$0.056 \times 16 = 0.896$	0	0	
$0.896 \times 16 = 14.336$	14	E	
$0.336 \times 16 = 5.376$	5	5	LSD

$$\therefore (0.301)_{10} = (0.4C0E5)_{16}$$

(C-7971) Fig. 1-Q. 1(ii)(b)

...Ans.

$$\therefore (73.301)_{10} = (49.4C0E5)_{16}$$



Step 4 : Convert decimal to BCD equivalent :

Decimal	7	3	.	3	0	1
BCD equivalent	0111	0011	.	0011	0000	0001

(C-7972) Fig. 1-Q. 1(ii)(c)

$$\therefore (73.301)_{10} = (0111\ 0011.0011\ 0000\ 0001)_{BCD}$$

...Ans.

Q. 2(a) Perform following operation :

(10 Marks)

- (i) Addition  $24_{BCD} + 18_{BCD}$     (ii) Subtraction  $46_{10} - 22_{10}$  using 2's complement method.

Ans. :

- (i) Addition  $24_{BCD} + 18_{BCD}$  :

Decimal	BCD	
$(24)_{10}$ :	0 0 1 0   0 1 0 0	
$(18)_{10}$ :	0 0 0 1   1 0 0 0	
Carry		
	0 0 1 1   1 1 0 0	
	Valid BCD	Invalid BCD

(C-7973) Fig. 1-Q. 2(a)

Add 6 to the invalid BCD for correction.

+	0 0 1 1   1 1 0 0
+	0 0 0 0   0 1 1 0
Carry	1 1 1   1
	0 1 0 0   0 0 1 0
	↓   ↓

(C-7974) Fig. 1-Q. 2(a)(a)

$$\therefore (24)_{BCD} + (18)_{BCD} = (42)_{BCD}$$

...Ans.

- (ii) Subtraction  $46_{10} - 22_{10}$  using 2's complement method :

Ans. :

Step 1 : Obtain 2's complement of  $(22)_{10}$  :

$$(22)_{10} \longrightarrow (010110)_2 \longrightarrow (101001)_2 \xrightarrow{\text{add 1}} (101010)_2$$

(C-7975) Fig. 1-Q. 2(a)(b)

Step 2 : Add  $(46)_{10}$  and 2's complement of  $(22)_{10}$  :

$(46)_{10}$	1 0 1 1 1 0
2's complement of $(22)_{10}$	+ 1 0 1 0 1 0
Carry	1 1 1
Final carry	☒ 0 1 1 0 0 0 → (24)_{10}

(C-7976) Fig. 1-Q. 2(a)(c)

- Final carry 1 indicates that the answer is positive and in its true form.

$$\therefore (46)_{10} - (22)_{10} = (24)_{10}$$

...Ans.

## Chapter 2 : Logic Gates and Boolean Algebra [Total Marks - 10]

Q. 2(b) Reduce the given expression and Realize using NAND gate only.

$$Y = AB' + AC' + C + AD + AB'C + ABC$$

(10 Marks)

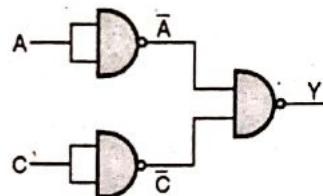
Ans. :

$$\begin{aligned} Y &= A\bar{B} + A\bar{C} + C + AD + A\bar{B}C + ABC \\ &= A\bar{B}(1+C) + A\bar{C} + C(1+AB) + AD \\ &= A\bar{B} + A\bar{C} + C + AD \quad [\because 1+C = 1 \text{ and } 1+AB = 1] \\ &= A\bar{B} + C + A + AD \quad [\because A + \bar{A}B = A + B] \\ &= A(\bar{B} + D + 1) + C = A + C \quad [\because 1 + \bar{B} + D = 1] \end{aligned}$$

Realization using NAND gates only :

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

$$Y = \overline{\overline{A \cdot C}}$$



(C-2993) Fig. 1-Q. 2(b)

### Chapter 3 : Logic Minimization and Reduction Techniques [Total Marks - 15]

Q. 1(iii) Simplify following three-variable expression using Boolean algebra πM (0, 1, 3, 4, 7). (5 Marks)

Ans. :

The given expression can be expressed in terms of max terms as,

$$Y = M_0 \cdot M_1 \cdot M_3 \cdot M_4 \cdot M_7$$

Step 1 : Bring the expression into SOP form :

$$\begin{aligned} Y &= (A+B+C)(A+B+\bar{C})(A+\bar{B}+\bar{C})(\bar{A}+B+C)(\bar{A}+\bar{B}+\bar{C}) \\ &= (A+B+C)(AA+A\bar{B}+A\bar{C}+AB+BB+BC+A\bar{C}+\bar{B}\bar{C}+\bar{C}\bar{C})(\bar{A}\bar{A}+\bar{A}\bar{B}+\bar{A}\bar{C}+\bar{A}B+BB+B\bar{C}+\bar{A}C+\bar{B}C+\bar{C}\bar{C}) \\ &= (A+B+C)(A+A\bar{B}+A\bar{C}+AB+0+B\bar{C}+A\bar{C}+\bar{B}\bar{C}+\bar{C})(\bar{A}+\bar{A}\bar{B}+\bar{A}\bar{C}+\bar{A}B+0+B\bar{C}+\bar{A}C+\bar{B}C+0) \\ &\quad (\because B\bar{B}=0, C\bar{C}=0, \bar{C}\bar{C}=\bar{C}) \\ &= (A+B+C)[A(1+\bar{B}+\bar{C}+B+\bar{C})+\bar{C}(B+\bar{B})+\bar{C}][\bar{A}(1+\bar{B}+\bar{C}+B+C)+B\bar{C}+\bar{B}C] \\ &= (A+B+C)[A+\bar{C}+\bar{C}][\bar{A}+B\bar{C}+\bar{B}C] \\ &= (A+B+C)(A+\bar{C})(\bar{A}+B\bar{C}+\bar{B}C) \\ &= (AA+A\bar{C}+BA+B\bar{C}+AC+C\bar{C})(\bar{A}+B\bar{C}+\bar{B}C) \\ &= [A+A(C+\bar{C})+AB+B\bar{C}+0][\bar{A}+B\bar{C}+\bar{B}C] \\ &= (A+A+AB+B\bar{C})(\bar{A}+B\bar{C}+\bar{B}C) \\ &= [A(1+1+B)+B\bar{C}][\bar{A}+B\bar{C}+\bar{B}C] \\ &= (A+B\bar{C})(\bar{A}+B\bar{C}+\bar{B}C) \\ &= AA+A\bar{B}\bar{C}+A\bar{B}C+A\bar{B}\bar{C}+B\bar{C}\bar{B}\bar{C}+B\bar{C}B\bar{C} \\ &= 0+A\bar{B}\bar{C}+A\bar{B}C+A\bar{B}\bar{C}+B\bar{C}+0 \\ &= B\bar{C}(A+\bar{A}+1)+ABC \\ Y &= B\bar{C}+ABC \end{aligned}$$

This is simplified expression.

a. 3(a) Simplify the following function using Quine-Mc Cluskey method.

$$\Sigma m(1, 2, 3, 5, 6, 12, 14, 15) + \Sigma d(4, 8, 11)$$

(10 Marks)

Ans:-

Step 1: Group the minterms according to number of 1's :

(C-NET) Table 1-Q. 3(a)

Group	Minterm	Binary representation			
		A	B	C	D
1	1	0	0	0	1
	2	0	0	1	0
	4	0	1	0	0
	5	1	0	0	0
2	3	0	0	1	1
	6	0	1	0	1
	8	1	0	0	1
	12	1	1	0	0
3	11	1	0	1	1
	14	1	1	1	0
4	15	1	1	1	1

Step 2 : Group the minterms to form pairs :

(C-NET) Table 1-Q. 3(a)(b)

Group	Pair of Minterm	Binary representation			
		A	B	C	D
1	1-3	0	0	-	1
	1-5	0	-	0	1
	1-9	-	0	0	1
	2-3	0	0	1	-
	4-5	0	1	0	-
	4-12	-	1	0	0
	5-9	1	0	0	-
	8-12	1	-	0	0
2	3-11	-	0	1	1
	9-11	1	0	-	1
3	11-15	1	-	1	1
	14-15	1	1	1	-

**Step 3 : Group the minterms to form quad :**

(C-7979) Table 1-Q. 3(a)(c)

Group	Minterm quad	Binary representation			
		A	B	C	D
1	1-3-9-11	-	0	-	1
	1-9-3-11	-	0	-	1

 $\bar{B}D$ 

**Step 4 : Prepare the table of prime implicants :**

(C-7980) Table 1-Q. 3(a)(d)

Prime Implicants	Decimal numbers	Given minterms						
		1	2	3	5	9	12	14
$\bar{A}BC$	2, 3	( $\otimes$ )	X					
$\bar{A}\bar{B}C$	4, 5			X				
$B\bar{C}D$	4, 12					X		
$A\bar{B}C$	8, 9				X			
$A\bar{C}D$	8, 12					X		
$ACD$	11, 15							X
$ABC$	14, 15						X	X
$BD$	1, 3, 9, 11			X		X		
$\bar{A}\bar{C}D$	1, 5	X		X				

Ans.

$$\therefore F(A, B, C, D) = \bar{B}D + \bar{A}\bar{B}C + \bar{A}\bar{C}D + A\bar{C}\bar{D} + ABC$$

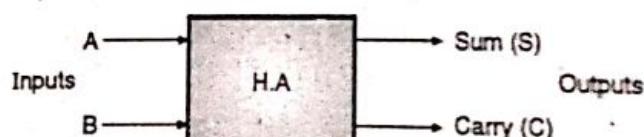
### Chapter 4 : Arithmetic Circuits [Total Marks - 15]

(5 Marks)

**Q. 1(iv) Design half adder circuit using basic gate.**

**Ans. : Half Adder :**

- Half adder is a combinational logic circuit with two inputs and two outputs. It is the basic building block for addition of two "single" bit numbers. This circuit has two outputs namely "carry" and "sum". The block diagram of half adder is as shown in Fig. 1-Q. 1(iv)(a).
- The half adder circuit is supposed to add two single bit binary numbers A and B. Therefore the truth table of a half adder is as shown in Fig. 1-Q. 1(iv)(b).



(a) Block diagram

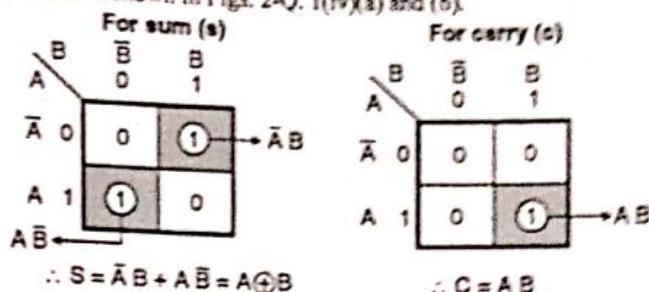
Inputs		Outputs	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

(b) Truth table

(C-344) Fig. 1-Q. 1(iv) : Half adder

**Karnaugh maps and simplified expressions for outputs :**

- K-maps for carry and sum outputs are as shown in Figs. 2-Q. 1(iv)(a) and (b).

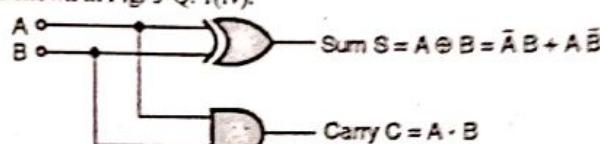


(C-345) Fig. 2-Q. 1(iv)

- Boolean expressions for the sum (S) and carry (C) output are obtained from the K-maps as follows :

$$\left. \begin{array}{l} S = \bar{A}B + A\bar{B} = A \oplus B \\ C = AB \end{array} \right\} \quad (C-374)$$

- The disadvantage of half adder is that addition of three bits is not possible to perform.
- Hence the half adder circuit is as shown in Fig. 3-Q. 1(iv).



(C-346) Fig. 3-Q. 1(iv) : Half adder circuit

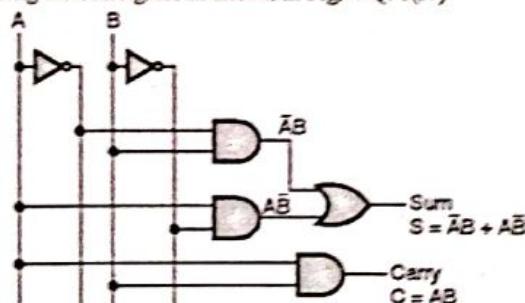
**Half adder using basic gates :**

- Refer Equation (1) which states that,

$$\text{Sum } S = \bar{A}B + A\bar{B}$$

$$\text{and Carry } C = AB$$

- These equations are implemented using the basic gates as shown in Fig. 4-Q. 1(iv)



(C-347) Fig. 4-Q. 1(iv) : Half adder using basic gates

**Q. 3(b) Design 2 Bit magnitude comparator using gates.**

(10 Marks)

**Ans. : Please refer Q. 4(a) of Dec. 2017.**
**Chapter 5 : Multiplexer and Demultiplexer [Total Marks - 10]**
**Q. 4(a) Implement the following Boolean function with 8 : 1 multiplexer.**

$$F(A, B, C, D) = \Sigma m(0, 2, 6, 10, 11, 12, 13) + \Sigma d(3, 8, 14)$$

(10 Marks)

**Ans. :**
**Step 1 : Write the design table :**

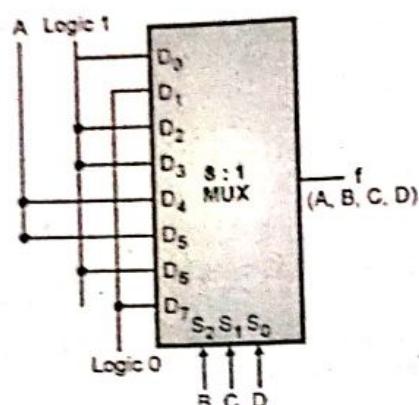
- The don't care conditions are assumed to be logic 1's. The design table is shown in Fig. 1-Q. 4(a)(a) and the corresponding logic diagram is shown in Fig. 1-Q. 4(a)(b).



	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>
X	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15
Input to Mux	1	0	1	1	A	A	1	0

(C-788) Fig. 1-Q. 4(a)(a) : Design table

Step 2 : Implementation using 8 : 1 MUX :



(C-788) Fig. 1-Q. 4(a)(b) : Implementation using 8 : 1 MUX

Chapter 6 : Logic Families [Total Marks - 05]

Q. 1(i) Compare TTL and CMOS logic families with respect to :

- (i) Power dissipation (ii) Propagation delay
- (iii) Figure of merit (iv) Fan-out.

(5 Marks)

Ans. :

Comparison of TTL and CMOS logic families:

Sr. No.	Parameter	CMOS	TTL
1.	Propagation delay	105 nS (Metal gate CMOS)	10 nS. (Standard TTL)
2.	Power dissipation per gate.	P <sub>D</sub> = 0.1 mW. Hence used for battery backup applications	10 mW
3.	Speed power product (Figure of Merit)	10.5 pJ	100 pJ
4.	Fan-out	Typically 50.	10

Chapter 7 : Latches and Flip Flops [Total Marks - 25]

Q. 1(v) Explain different types of triggering methods used for flipflop.

(5 Marks)

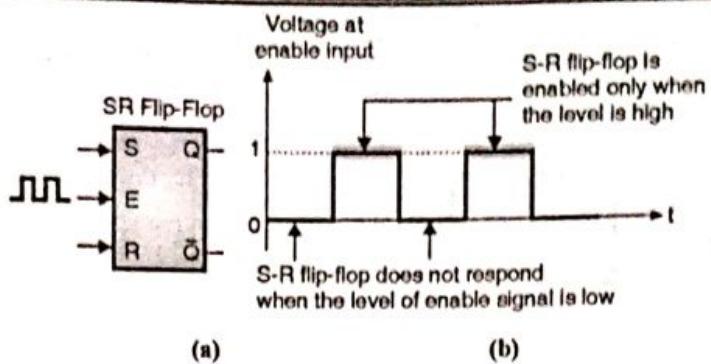
Ans. :

Triggering Methods :

1. Level triggered circuits
2. Edge triggered circuits

Concept of Level Triggering :

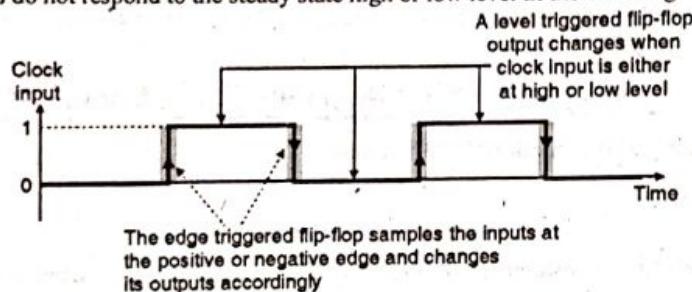
- The latch or flip-flop circuits which respond to change in their inputs, only if their enable input (E) held at an active level which may be either HIGH or LOW level are called as level triggered latches or flip-flops.
- Thus these circuits do not respond at the rising or falling edges of clock.
- They only respond to the steady HIGH or LOW levels of the clock signal.
- Fig. 1-Q. 1(v)(a) shows the symbol of a level triggered SR flip flop and Fig. 1-Q. 1(v)(b) shows the clock signal applied at its input.



(C-576) Fig. 1-Q. 1(v) : Concept of level triggering

**Concept of Edge Triggering :**

- The flipflops which change their outputs only corresponding to the positive (rising) or negative (falling) edge of the clock input are called as **edge triggered flipflops**.
- These flip-flops are therefore said to be edge sensitive or edge triggered rather than level triggered.
- The rectangular signal applied to the clock input of a flip-flop is shown in Fig. 2-Q. 1(v).
- If the same signal is applied as the clock signal to an edge triggered flip flop, then its outputs will change only at either rising (positive) edge or at the falling (negative) edge of the clock.
- The edge triggered flip-flops do not respond to the steady state high or low level in the clock signal at all.



(C-577) Fig. 2-Q. 1(v)

Q. 5(b) Convert JK flip flop into D and SR flip flop.

(10 Marks)

Ans. :

Conversion of JK flip flop into D flip flop : Please refer Q. 1(d) of Dec. 2017.

Conversion of JK Flip Flop to SR Flip Flop Conversion :

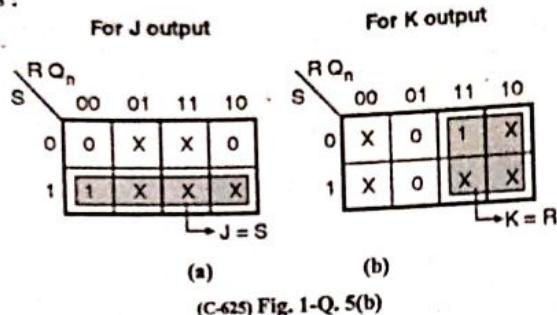
Step 1 : Write the truth table for JK to SR :

Table 1-Q. 5(b) : Truth table for JK to SR conversion

Inputs			Outputs		
S	R	Present state $Q_n$	Next state $Q_{n+1}$	J	K
0	0	0	0	0	x
0	1	0	0	0	x
1	0	0	1	1	x
1	0	0	1	1	x
0	1	1	0	x	1
0	1	1	0	x	1
0	0	1	1	x	0
1	0	1	1	x	0



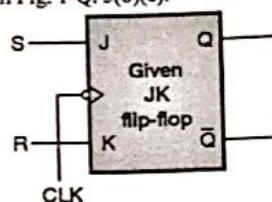
## Step 2 : K-maps and simplifications :



(C-625) Fig. 1-Q. 5(b)

## Step 3 : Logic diagram :

The logic diagram for JK to SR FF is shown in Fig. 1-Q. 5(b)(c).



(C-626) Fig. 1-Q. 5(b)(c) : JK to SR FF conversion

**Q. 6(b)** Explain race around condition in JK flipflop and discuss solution to avoid race around condition. (10 Marks)  
Ans. : Please refer Q. 3(b) of Dec. 2018.

Chapter 8 : Shift Registers [Total Marks - 10]

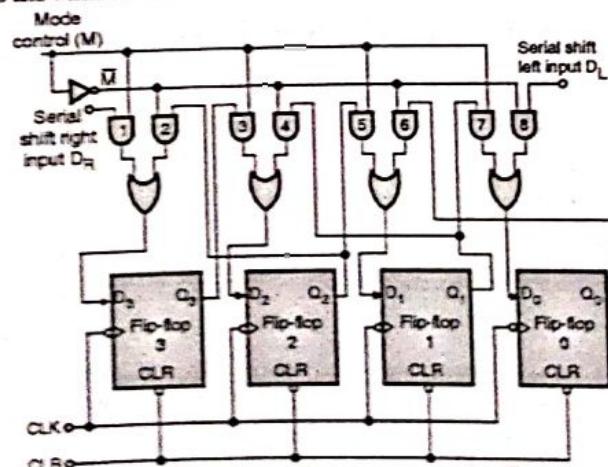
(10 Marks)

**Q. 4(b)** Draw and explain working of bidirectional shift register.

Ans. :

**Bidirectional Shift Register :**

- If a binary number is shifted left by one position then it is equivalent to multiplying the original number by 2. On the other hand if a binary number is shifted right by one position then it is equivalent to dividing the original number by 2. This is illustrated below.
- Let a four bit number  $Q_3 Q_2 Q_1 Q_0 = 0010 = (2)_{10}$  is existing in a shift register. Now with a 0 applied at the input, if we shift these contents by one position to left then we get  $Q_3 Q_2 Q_1 Q_0 = 0100 = (4)_{10}$ . Thus the shift left is equivalent to multiplying by 2. Now shift it right by one position to get  $Q_3 Q_2 Q_1 Q_0 = 0001 = (1)_{10}$ . Thus shifting right is equivalent to dividing by 2.
- Hence if we want to use the shift register to multiply and divide the given binary number, then we should be able to move the data in either left or right direction as and when we want.



(C-750) Fig. 1-Q. 4(b) : A 4-bit bi-directional shift register

- Such a register is called as a bi-directional register. A four bit bi-directional shift register is shown in Fig. 1-Q. 4(b).
- There are two serial inputs namely the serial right shift data input  $D_R$  and the serial left shift data input  $D_L$  alongwith a Mode control input (M).

**Operation :**

**With  $M = 1$  : Shift right operation**

- If  $M = 1$ , then the AND gates 1, 3, 5 and 7 are enabled whereas the remaining AND gates 2, 4, 6 and 8 will be disabled.
- Hence the data at  $D_R$  (shift right input) is shifted right bit by bit from FF-3 to FF-0 on the application of clock pulses.
- Thus with  $M = 1$  we get the serial right shift operation.

**With  $M = 0$  : Shift left operation**

- When the mode control M is connected to "0" then the AND gates 2, 4, 6 and 8 are enabled while 1, 3, 5 and 7 are disabled.
- Therefore the data at  $D_L$  (shift left input) is shifted left bit by bit from FF-0 to FF-3 on application of the clock pulses.
- Thus with  $M = 0$  we get the serial left shift operation.
- Note that M should be changed only when  $CLK = 0$ , otherwise the data stored in the register may get changed in an undesirable manner.

### Chapter 9 : Counters [Total Marks - 20]

**Q. 5(a) Design MOD 13 asynchronous up counter using JK Flip flop.**

**(10 Marks)**

**Ans. :**

**Step 1 : Write the truth table :**

(C-7913) Table 1-Q. 5(a)

Inputs				Output
A	B	C	D	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

↑  
Invalid  
states

**Step 2 : Draw the K-map :**

- The K-map is shown in Fig. 1-Q. 5(a).



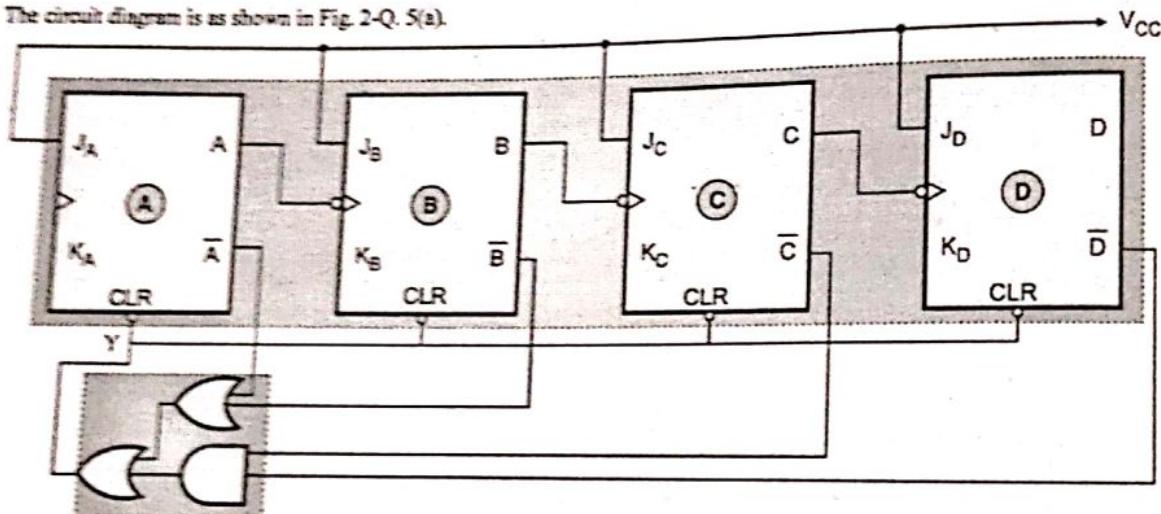
For Y					
CD	AB	00	01	11	10
		1	1	1	1
		1	1	0	0
		1	1	1	1

(C-7984) Fig. 1-Q. 5(a) : K-map

$$\therefore Y = \bar{A} + \bar{B} + \bar{C}\bar{D}$$

Step 3 : Draw the circuit diagram :

- The circuit diagram is as shown in Fig. 2-Q. 5(a).



(C-7985) Fig. 2-Q. 5(a) : Logic diagram of MOD-13 asynchronous up counter

Q. 6(a) Design 3 bit synchronous counter using T flip flop.

(10 Marks)

Ans. :

Design of the 3 Bit Synchronous Counter :

Step 1 : Decide number of FFs : A 3 bit counter goes through 8 states. So it needs three flip flops.

Step 2 : Excitation table of T FFs :

- Table 1-Q. 6(a) shows the excitation table of T FF.

Table 1-Q. 6(a) : Excitation table of a T FF

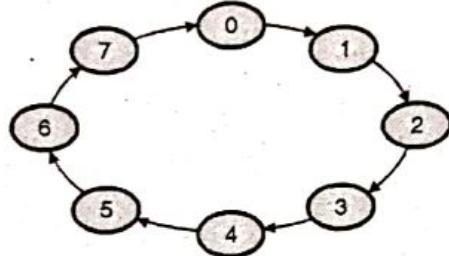
Present state $Q_s$	Next state $Q_{s+1}$	T
0	0	0
0	1	1
1	0	1
1	1	0

**Step 3 : State diagram and circuit excitation table :**

- Count sequence for a 3 bit up counter is given in Table 2-Q. 6(a) and Fig. 1-Q. 6(a) shows the corresponding state diagram.
- Table 1-Q. 6(a)(c) shows the circuit excitation table.

(C-6226) Table 2-Q. 6(a)

$Q_C$	$Q_B$	$Q_A$
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1



(C-819) Fig. 1-Q. 6(a) : State diagram

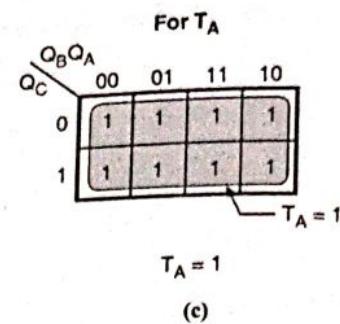
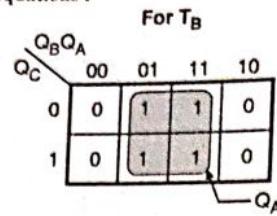
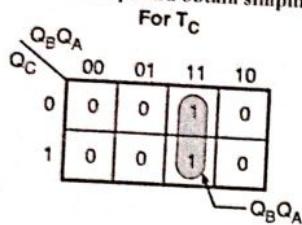
- Table 3-Q. 6(a)(c) has been written by referring to the present and next state of an output and the required value to input is written as per the excitation table of T FF.
- For example consider the shaded columns of Table 1-Q. 6(a)(c). i.e.  $Q_C$ ,  $Q_{C+1}$  and  $T_C$ .
- Consider the first row.  $Q_C = 0$ ,  $Q_{C+1} = 0$ . So as per the excitation table of a T FF  $T_C$  should be 0. Similarly all other entries are made.

Table 3-Q. 6(a) : Circuit excitation table

Present state			Next state			Flip flop input		
$Q_C$	$Q_B$	$Q_A$	$Q_{C+1}$	$Q_{B+1}$	$Q_{A+1}$	$T_C$	$T_B$	$T_A$
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1



**Step 4 : Write K maps and obtain simplified equations :**

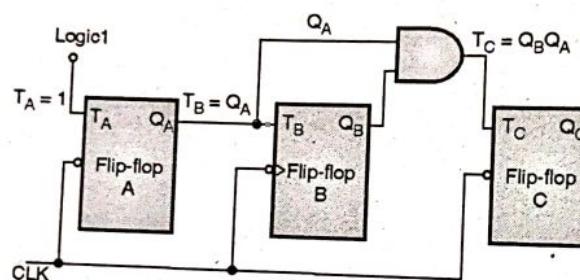


(C-1506) Fig. 2-Q. 6(a) : K-maps for different FF inputs

- Refer Figs. 2-Q. 6(a), (b), (c) for the K-maps corresponding to all the FF inputs. The simplified equations for  $T_A$ ,  $T_B$  and  $T_C$  also are shown.

**Step 5 : Draw the logic diagram :**

By using the simplified equations for  $T_A$ ,  $T_B$  and  $T_C$  we can draw the logic diagram for the 3 bit synchronous shown in Fig. 3-Q. 6(a).



(C-1507) Fig. 3-Q. 6(a) : Logic diagram of a 3-bit synchronous counter using T flip flop

---

## Question Papers

---

Dec. 2017

**Q. 1** Solve following

(20 Marks)

(a) Explain the following decimals in gray code form.

1.  $(42)_{10}$

2.  $(17)_{10}$

(b) Explain characteristics of logic families.

(c) State and prove Demorgan theorem.

(d) Convert JK flip flop to T flip flop.

**Q. 2 (a)** What is shift register ? Explain any one type of shift register. Give its applications.

(10 Marks)

(b) Implement the following Boolean function using 8 : 1 multiplexer.

$$F(A, B, C, D) = \sum M(0, 1, 4, 5, 6, 8, 10, 12, 13)$$

(10 Marks)

**Q. 3 (a)** Explain the Johnson's counter. Design for initial state 0110. From initial state explain and draw all possible states.

(10 Marks)

(b) Minimize the following expression using Quine McCluskey technique.

$$F(A, B, C, D) = \sum M(0, 1, 2, 3, 5, 7, 9, 11)$$

(10 Marks)

**Q. 4 (a)** Design a 2 bit comparator and implement using logic gates.

(10 Marks)

(b) Using Boolean algebra and De-Morgan's theorem prove that :

$$\bar{Y}\bar{Z} + \bar{W}\bar{X}\bar{Z} + \bar{W}XY\bar{Z} + WY\bar{Z} = \bar{Z}$$

Simplify the expression  $[A\bar{B}(C + BD) + \bar{A}\bar{B}]C$  as much as possible.

(10 Marks)

**Q. 5 (a)** Explain the working of 3 bit asynchronous counter with proper timing diagram.

(10 Marks)

(b) Design BCD adder using the integrated circuit 4 bit binary adders.

(10 Marks)

**Q. 6** Write short notes on following :

(20 Marks)

(a) Hazards.

(b) Hamming code.

(c) Encoder and decoder.

(d) Compare TTL and CMOS logic families.





**Q. 1 (a)** Convert the following numbers as mentioned against them :

(I)  $(101011)_2$  convert to decimal number.

(II) Convert  $(120.625)_{10}$  hexadecimal form.

(III) Write  $(-20)_{10}$  in two's complement form.

**Q. 1 (b)** Write differences between synchronous and asynchronous counters.

**Q. 1 (c)** Explain use of latch as a switch debouncer.

**Q. 1 (d)** Explain current and voltage parameters of logic families.

**Q. 2 (a)** Simplify using Quine McCluskey method and draw the logic diagram using basic gates for the following function

$$Y = F(A, B, C, D) = \sum m(5, 11, 13, 14, 15) + \sum d(4, 6, 7).$$

**Q. 2 (b)** Draw four bit ring counter and explain its operation.

**Q. 3 (a)** Implement the following function using only one 4 : 1 multiplexer and gates.

$$Y = F(A, B, C, D) = \sum m(2, 3, 5, 7, 10, 11, 12, 13)$$

**Q. 3 (b)** Design 3 bit look ahead carry generator circuit.

**Q. 4 (a)** Draw circuit diagram of 2 input TTL NAND gate and explain its operation.

**Q. 4 (b)** Implement full adder using decoder having active low outputs and gates with fan in 2.

**Q. 5 (a)** Design lockout free mod 10 up synchronous counter using JKMS flip flops.

**Q. 5 (b)** Explain parity circuits.

**Q. 6 (a)** Convert the flip flop

(I) JKMS to D flip flop

(II) SR to T flip flop.

**Q. 6 (b)** Design 8 bit comparator using 4 bit comparator IC 7485 and explain its operation.

Dec. 2018

**Q. 1 (a)** Design and implement full subtractor using logic gates.

**Q. 1 (b)** Explain the working of a two - inputs CMOS, NOR gate with a neat diagram.

**Q. 1 (c)** Design a circuit using 2 : 1 MUX to implement 2 input NAND gate.

(d) Evaluate following operation in BCD :

i.  $(56)_{10} + (23)_{10}$

ii.  $(48)_{10} + (26)_{10}$

Q-3

Q. 2 (a) Convert  $(27)_{10}$  and  $(42)_{10}$  into binary, octal, hexadecimal, excess-3 code and gray code. (5 Marks)

(b) Draw a neat circuit diagram of four bit twisted ring counter with initial state 0000 and relevant output waveforms. (10 Marks)

Q. 3 (a) Design a combinational logic circuit with four input variables that will produce logic 1 output when input is greater than 9. (10 Marks)

(b) Draw a circuit diagram of clocked J-K flip-flop using NAND gates with truth table. What is race around condition and how does it get eliminated? (10 Marks)

Q. 4 (a) Simplify the expression in POS form for given function and realize it with basic gates. (10 Marks)

$$F(A, B, C, D) = \Sigma m(0, 4, 6, 7, 10, 12, 14) + d(2, 13)$$

(b) Convert the followings : (10 Marks)

i. SR flip flop to JK flip flop.

ii. JK flip-flop to D flip-flop

Q. 5 (a) Implement the following expression using a single 8 : 1 multiplexer. (10 Marks)

$$F(A, B, C, D) = \Sigma m(0, 2, 4, 6, 8, 10, 12, 14)$$

(10 Marks)

(b) Simplify the following four variable Boolean function using Quine-McCluskey technique.

$$F(A, B, C, D) = \Sigma m(0, 2, 3, 6, 7, 8, 10, 12, 13)$$

(10 Marks)

Q. 6 (a) Design a Mod-5 synchronous up counter using T flip-flop. Design using minimal cost approach. (10 Marks)

(b) Explain interfacing of a TTL gate driving CMOS gates and vice versa. (10 Marks)

□□□

May 2019

Q. 1 Attempt any 4 questions from the following : (20 Marks)

(i) Compare TTL and CMOS logic families with respect to :

- (i) Power dissipation (ii) Propagation delay (iii) Figure of merit (iv) Fan-out.

(ii) Convert  $(73.301)_{10}$  into binary, octal, Hexadecimal and BCD equivalent.

(iii) Simplify following three-variable expression using Boolean algebra  $\pi M(0, 1, 3, 4, 7)$

(iv) Design half adder circuit using basic gate.

(v) Explain different types of triggering methods used for flipflop.



Q. 2 (a) Perform following operation :

- (i) Addition 24BCD + 18BCD      (ii) Subtraction 4610 - 2210 using 2's complement method.

(b) Reduce the given expression and Realize using NAND gate only.

$$Y = AB' + AC' + C + AD + AB'C + ABC$$

Q. 3 (a) Simplify the following function using Quine-Mc Cluskey method.

$$\Sigma m (1, 2, 3, 5, 9, 12, 14, 15) + \Sigma d (4, 8, 11)$$

(b) Design 2 Bit magnitude comparator using gates.

Q. 4 (a) Implement the following Boolean function with 8 : 1 multiplexer.

$$F(A, B, C, D) = \Sigma m (0, 2, 6, 10, 11, 12, 13) + \Sigma d (3, 8, 14)$$

(b) Draw and explain working of bidirectional shift register.

Q. 5 (a) Design MOD 13 asynchronous up counter using JK Flip flop.

(b) Convert JK flip flop into D and SR flip flop.

Q. 6 (a) Design 3 bit synchronous counter using T flip flop.

(b) Explain race around condition in JK flip flop and discuss solution to avoid race around condition.

(10 Marks)

□□□

**• Your Success is Our Goal**

**• Semester III - Electronics Engineering**

**• Digital Circuit Design**

**• Electronic Devices and Circuits - I**

**• Electronics Instruments and Measurement**

**• Electrical Network Analysis and Synthesis**



now with



**Paper Solutions Trusted by lakhs of students from more than 15 years**

## Distributors

### MUMBAI

**Student's Agencies (I) Pvt. Ltd.**

102, Konark Shram, Ground Floor, Behind Everest  
Building, 156 Tardeo Road, Mumbai.  
M : 91672 90777.

**Vidyaarthi Sales Agencies**

Shop. No. 5, Hendre Mansion, Khotachiwadi, 157/159,  
J.S.S Road, Girgaum, Mumbai. M : 98197 76110.

**Bharat Sales Agency**

Goregaonkar Lane, Behind Central Plaza Cinema,  
Charni Road, Mumbai. M : 86572 92797

**Ved Book Distributors - Mr. Sachin Waingade**  
*(For Library Orders)*

M : 80975 71421 / 92208 77214.  
E : mumbai@techknowledgebooks.com

**BOOKS ARE AVAILABLE AT ALL LEADING BOOKSELLERS !!**

**EMO34A Price ₹ 60/-**



**B-50**