



# QUANTUM Series

Semester - 3 Electronics & Allied Branches

## Digital System Design



- Topic-wise coverage of entire syllabus in Question-Answer form.
- Short Questions (2 Marks)

Session  
2019-20  
Odd Semester

Includes solution of following AKTU Question Papers

2014-15 • 2015-16 • 2016-17 • 2017-18 • 2018-19

## CONTENTS

### KEC 302 : Digital System Design

#### **UNIT-1 : LOGIC SIMPLIFICATION & COMBINATIONAL LOGIC DESIGN**

**(1-1 B to 1-32 B)**

Binary codes, code conversion, review of Boolean algebra and Demorgans theorem, SOP & POS forms, Canonical forms, Karnaugh maps up to 6 variables, tabulation method.

#### **UNIT-2 : MSI DEVICES**

**(2-1 B to 2-31 B)**

MSI devices like comparators, multiplexers, encoder, decoder, driver & multiplexed display, half and full adders, subtractors, serial and parallel adders, BCD adder, barrel shifter and ALU.

#### **UNIT-3 : SEQUENTIAL LOGIC DESIGN**

**(3-1 B to 3-40 B)**

Sequential logic design: Building blocks like S-R, JK and Master-Slave JK FF, edge triggered FF, state diagram, state reduction, design of sequential circuits, ripple and synchronous counters, shift registers, finite state machines, design of synchronous FSM, algorithmic state machines charts. Designing synchronous circuits like pulse train generator, pseudo random binary sequence generator, clock generation.

#### **UNIT-4 : LOGIC FAMILIES & MEMORIES**

**(4-1 B to 4-32 B)**

TTL NAND gate, specifications, noise margin, propagation delay, fan-in, fan-out, tristate TTL, ECL, CMOS families and their interfacing, memory elements, concept of programmable logic devices like FPGA, logic implementation using programmable devices.

#### **UNIT-5 : D/A AND A/D CONVERTER**

**(5-1 B to 5-21 B)**

Digital-to-Analog converters (DAC): Weighted resistor, R-2R ladder, resistor string etc. analog-to-digital converters (ADC): single slope, dual slope, successive approximation, flash etc. Switched capacitor circuits: Basic concept, practical configurations, application in amplifier, integrator, ADC etc.

#### **SHORT QUESTIONS**

**(SQ-1 B to SQ-17 B)**

#### **SOLVED PAPERS (2014-15 TO 2018-19)**

**(SP-1 B to SP-32 B)**



## Logic Simplification and Combinational Logic Design

### CONTENTS

<b>Part-1</b> : Binary Codes .....	<b>1-2B to 1-3B</b>
<b>Part-2</b> : Code Conversion .....	<b>1-3B to 1-12B</b>
<b>Part-3</b> : Review of Boolean Algebra and .....	<b>1-12B to 1-15B</b>
Demorgan's Theorem : SOP and	
POS Forms, Canonical Forms	
<b>Part-4</b> : Karnaugh Maps Upto 6 .....	<b>1-15B to 1-22B</b>
Variables	
<b>Part-5</b> : Tabulation (Quine-McCluskey) .....	<b>1-22B to 1-31B</b>
Method	

**PART-1***Binary Code.***Questions-Answers****Long Answer Type and Medium Answer Type Questions**

**Que 1.1.** Describe the binary codes. Show the classification of binary codes in tabular format.

**Answer**

1. Code is the representation of group of symbols, words, or letters. As the digital data is used as group of binary numbers, so, we call it as the binary codes.
2. These binary codes are used for the designing and analysis of digital circuit, computer applications, in digital communication. The codes are classified into certain following categories :
  - i. Weighted codes
  - ii. Non-weighted codes
  - iii. Reflective codes
  - iv. Sequential codes
  - v. Alphanumeric codes
  - vi. Error detecting and correcting codes.
3. Since, all these codes use only 0 and 1, so it is easier to implement. The binary codes can also be used for representing the numbers as well as the alphanumeric letters.
4. The classification of codes can be composed in tabular form which is as follows :

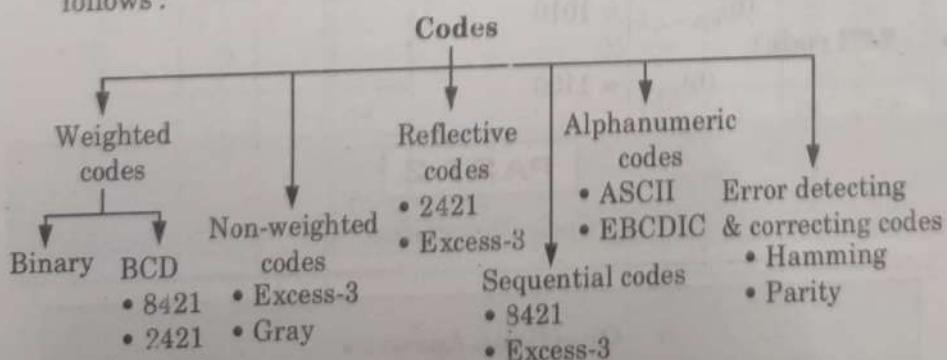


Fig. 1.1.1

5. Weighted binary codes are those which obey the positional weight for the number to represent.
6. In non-weighted codes, the positional weights are not assigned.
7. In reflective code, the reflectivity is desirable. For example, in  $9_8$ , complement subtraction, i.e., code for 9 is the complement for 0, code for 8 is complement of 1, 7 for 2, 6 for 3 and 5 for 4.
8. In sequential code, each succeeding code is one binary number greater than the preceding code.
9. The alphanumeric codes are designed to represent numbers as well as characters.
10. The error detecting and correcting codes are used to detect and correct the error like 0 may change to 1 or vice-versa by using some special codes which possess the capacity to detect and correct the error.

**Que 1.2.** Represent the decimal number 6 in (i) excess-3 code, (ii) BCD code, (iii) Gray code, (iv) 8421 code and (v) 2421 codes.

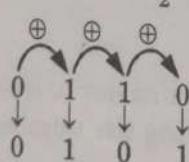
**Answer**

i. Excess-3 code :

$$\begin{array}{r}
 6 \text{ (in BCD)} = 0110 \\
 + 3 \qquad \qquad \qquad = 0011 \\
 \hline
 9 \qquad \qquad \qquad = 1001
 \end{array}$$

ii. BCD code :  $(6)_{10} = 0110$  (in BCD)

iii. Gray code :  $(6)_2 = 0110$



Gray code = 0101

iv. 8421 code :  $(6)_2 = 0110$

$$(6)_{8,4,-2,-1} = 1010$$

v. 2421 code :

$$(6)_{2,4,2,1} = 1100$$

**PART-2**

*Code Conversion.*

**Questions-Answers**

Long Answer Type and Medium Answer Type Questions

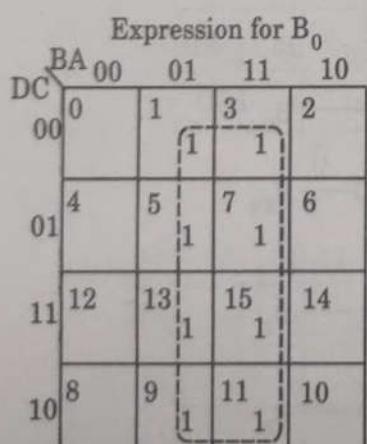
**Que 1.3.** Design Binary to BCD code converter.

**Answer**

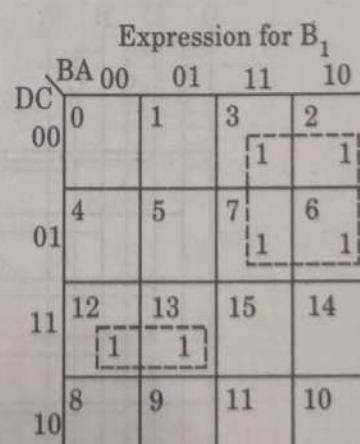
Truth table :

Binary code				BCD code				
D	C	B	A	$B_4$	$B_3$	$B_2$	$B_1$	$B_0$
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1	0
0	0	1	1	0	0	0	1	1
0	1	0	0	0	0	1	0	0
0	1	0	1	0	0	1	0	1
0	1	1	0	0	0	1	1	0
0	1	1	1	0	0	1	1	1
1	0	0	0	0	1	0	0	0
1	0	0	1	0	1	0	0	1
1	0	1	0	1	0	0	0	0
1	0	1	1	1	0	0	0	1
1	1	0	0	1	0	0	1	0
1	1	0	1	1	0	0	1	1
1	1	1	0	1	0	1	0	0
1	1	1	1	1	0	1	0	1

K-map simplification :



$$B_0 = A$$



$$B_1 = DC\bar{B} + \bar{D}\bar{B}$$

Expression for $B_2$					
DC	BA	00	01	11	10
00		1		3	2
01			5	7	6
11		1	1	1	1
10		12	13	15	14
				1	1
		8	9	11	10

$$B_2 = \overline{DC} + CB$$

Expression for $B_3$					
DC	BA	00	01	11	10
00		1		3	2
01			5	7	6
11		12	13	15	14
10		8	9	11	10
		1	1	1	1
				1	1

$$B_3 = DC\overline{B}$$

Expression for $B_4$					
DC	BA	00	01	11	10
00		1		3	2
01			5	7	6
11		12	13	15	14
10		8	9	11	10
		1	1	1	1
				1	1

$$B_4 = DC + DB$$

Logic diagram :

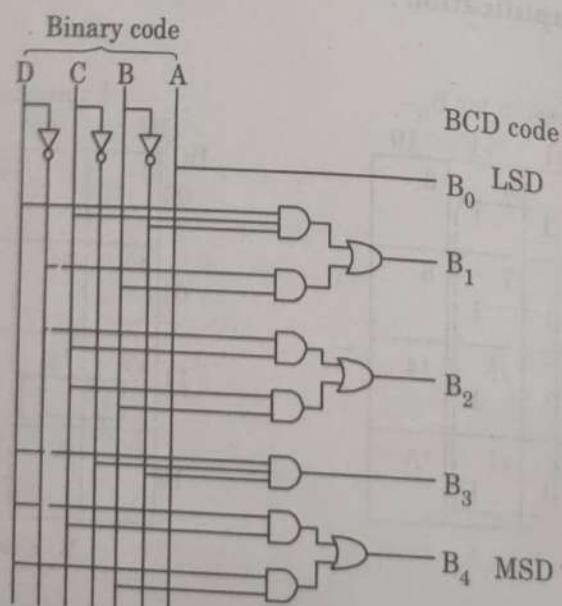


Fig. 1.3.1. Logic circuit for binary to BCD converter.

**Que 1.4.** Design a combinational circuit that converts a BCD code to excess-3 code.

[ARTU 2016-17, Marks 15]

**Answer**

Truth table :

Input BCD				Output excess-3 code			
A	B	C	D	w	x	y	z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

1. The maps in Fig. 1.4.1, are plotted to obtain simplified boolean functions for the outputs.
2. A two-level logic diagram of each output may be obtained directly from the boolean expressions derived from the maps.

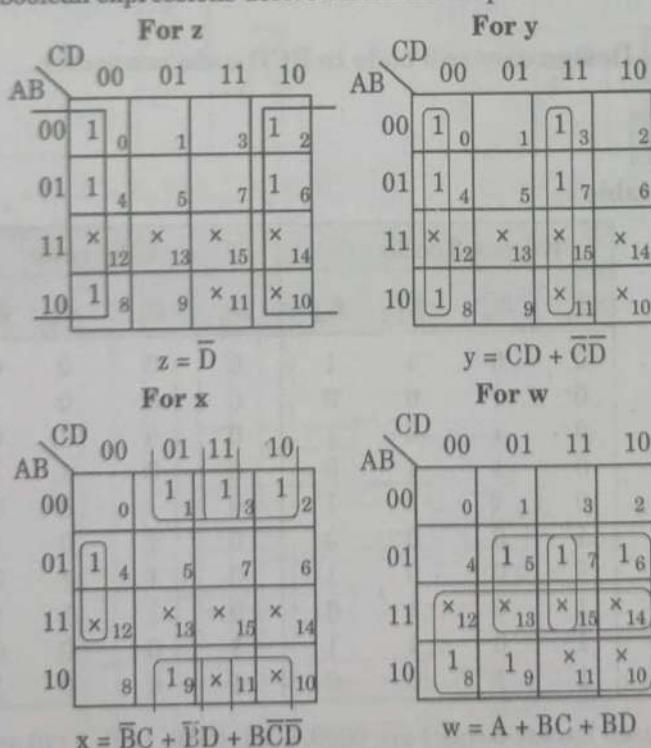


Fig. 1.4.1. Maps for BCD to excess-3 code converter.

3. The expressions derived from the maps are :

$$z = \bar{D}$$

$$y = CD + \bar{C}\bar{D} = CD + \overline{(C+D)}$$

$$\begin{aligned} x &= \bar{B}C + \bar{B}\bar{D} + B\bar{C}\bar{D} = \bar{B}(C+D) + B\bar{C}\bar{D} \\ &= \bar{B}(C+D) + B\overline{(C+D)} \end{aligned}$$

$$w = A + BC + BD = A + B(C+D)$$

4. The logic diagram that implements these expressions is shown in Fig. 1.4.2.

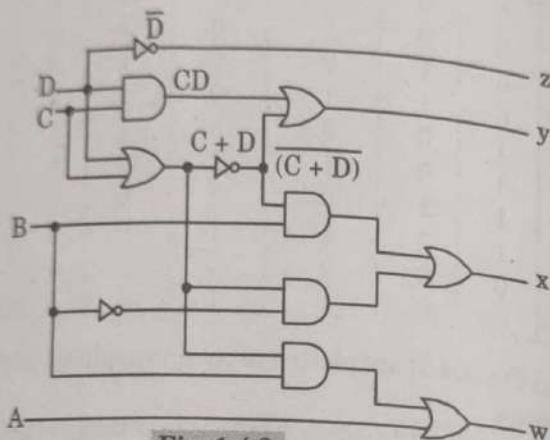


Fig. 1.4.2.

**Que 1.5.** Design excess-3 code to BCD code converter.

**Answer**

Truth table :

Excess-3 code				BCD code			
$E_3$	$E_2$	$E_1$	$E_0$	$B_3$	$B_2$	$B_1$	$B_0$
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	1
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	1
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	1

The unused Excess-3 codes are 0000, 0001, 0010, 1101, 1110 and 1111. So place X (Don't Care condition) for the corresponding codes.

**K-map simplification :**

		Expression for $B_0$				
		00	01	11	10	
		E <sub>1</sub> E <sub>0</sub>	00	01	11	10
E <sub>3</sub>	E <sub>2</sub>	00	x	1	3	2
00	00	00	x	x		x
01	01	01	4	5	7	6
11	11	11	12	13	15	14
10	10	10	8	9	11	10

$$B_0 = \bar{E}_0$$

		Expression for $B_1$				
		00	01	11	10	
		E <sub>1</sub> E <sub>2</sub>	00	01	11	10
E <sub>3</sub>	E <sub>2</sub>	00	x	1	3	2
00	00	00	x	x		x
01	01	01	4	5	7	6
11	11	11	12	13	15	14
10	10	10	8	9	11	10

$$B_1 = \bar{E}_1 E_0 + E_1 \bar{E}_0 = E_1 \oplus E_0$$

		Expression for $B_2$				
		00	01	11	10	
		E <sub>1</sub> E <sub>0</sub>	00	01	11	10
E <sub>3</sub>	E <sub>2</sub>	00	x	1	3	2
00	00	00	x	x		x
01	01	01	4	5	7	6
11	11	11	12	13	15	14
10	10	10	8	9	11	10

$$B_2 = \bar{E}_2 \bar{E}_1 + E_2 E_1 E_0 + E_3 E_1 \bar{E}_0$$

		Expression for $B_3$				
		00	01	11	10	
		E <sub>1</sub> E <sub>2</sub>	00	01	11	10
E <sub>3</sub>	E <sub>2</sub>	00	x	1	3	2
00	00	00	x	x		x
01	01	01	4	5	7	6
11	11	11	12	13	15	14
10	10	10	8	9	11	10

$$B_3 = E_3 E_2 + E_3 E_1 E_0$$

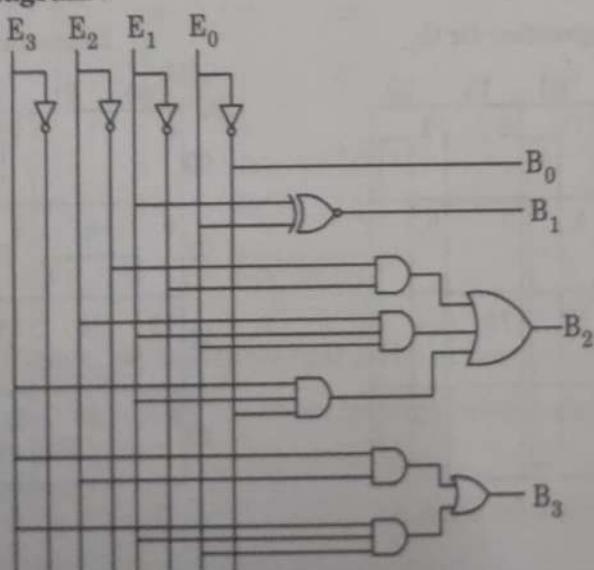
**Logic diagram :**

Fig. 1.5.1.

Que 1.6, Design Binary code to Gray code converter.

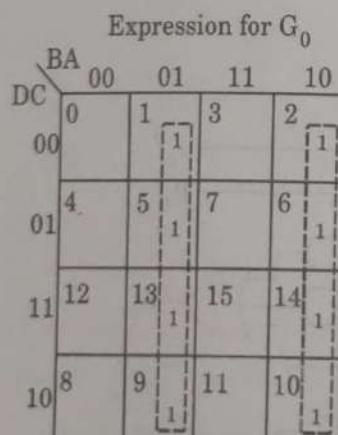
AKTU 2018-19, Marks 07

## Answer

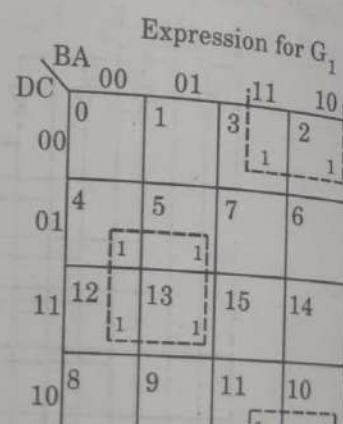
Truth table :

Binary code				Gray code			
D	C	B	A	G <sub>3</sub>	G <sub>2</sub>	G <sub>1</sub>	G <sub>0</sub>
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	1	1
0	1	0	0	0	0	1	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	0
0	1	1	1	0	1	0	1
1	0	0	0	1	1	0	1
1	0	0	1	1	1	0	0
1	0	1	0	1	1	0	0
1	0	1	1	1	1	1	1
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	0
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	1

K-map simplification :



$$G_0 = \bar{B}A + B\bar{A} = B \oplus A$$



$$G_1 = C\bar{B} + \bar{C}B = C \oplus B$$

		Expression for $G_2$				
		BA	00	01	11	10
DC	00	0	1	3	2	
	01	4	5	7	6	
DC	11	12	13	15	14	
	10	8	9	11	10	
		(1)	(1)	(1)	(1)	(1)

$$G_2 = \bar{D}\bar{C} + D\bar{C} = C \oplus D$$

		Expression for $G_3$				
		BA	00	01	11	10
DC	00	0	1	3	2	
	01	4	5	7	6	
DC	11	12	13	15	14	
	10	8	9	11	10	
		(1)	(1)	(1)	(1)	(1)

$$G_3 = D$$

We get the simplified boolean expression for the code converter of Binary to Gray code.

$$G_0 = B\bar{A} + \bar{B}A = B \oplus A$$

$$G_1 = C\bar{B} + \bar{C}B = C \oplus B$$

$$G_2 = D\bar{C} + \bar{D}C = C \oplus D$$

$$G_3 = D$$

Logic diagram :

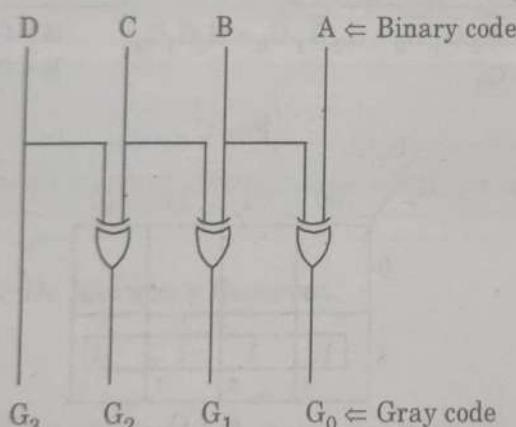
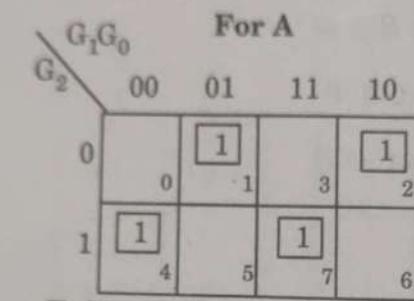


Fig. 1.6.1.

**Que 1.7.** Design a combinational circuit that converts a 3-bit Gray code to a 3-bit binary number. Implement the circuit with  
 i. Exclusive-OR gate  
 ii. NAND gate only.

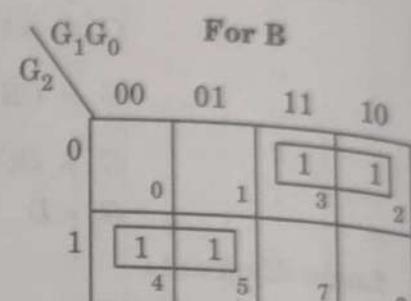
**Answer****Gray code to binary code converter :**

Gray code			Binary code		
$G_2$	$G_1$	$G_0$	$C$	$B$	$A$
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	1
1	0	1	1	1	0
1	1	0	1	0	0
1	1	1	1	0	1

**K-map simplification :**

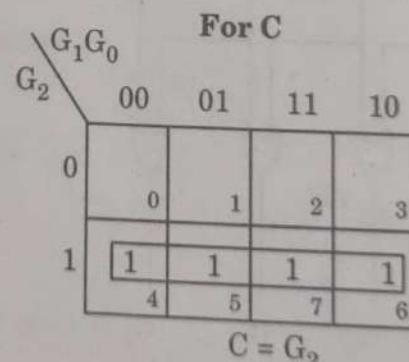
$$A = \bar{G}_2 \bar{G}_1 G_0 + \bar{G}_2 G_1 \bar{G}_0 + G_2 \bar{G}_1 \bar{G}_0 + G_2 G_1 G_0$$

$$A = G_2 \oplus G_1 \oplus G_0$$



$$B = G_2 \bar{G}_1 + \bar{G}_2 G_1$$

$$B = G_2 \oplus G_1$$



$$C = G_2$$

- Logic diagram :**  
i. Using XOR gates :

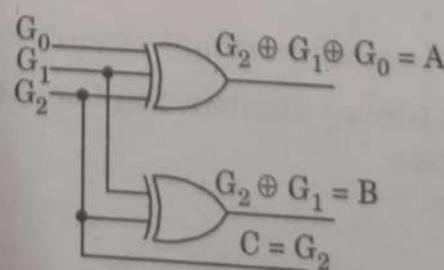


Fig. 1.7.1.

## ii. Using NAND gates :

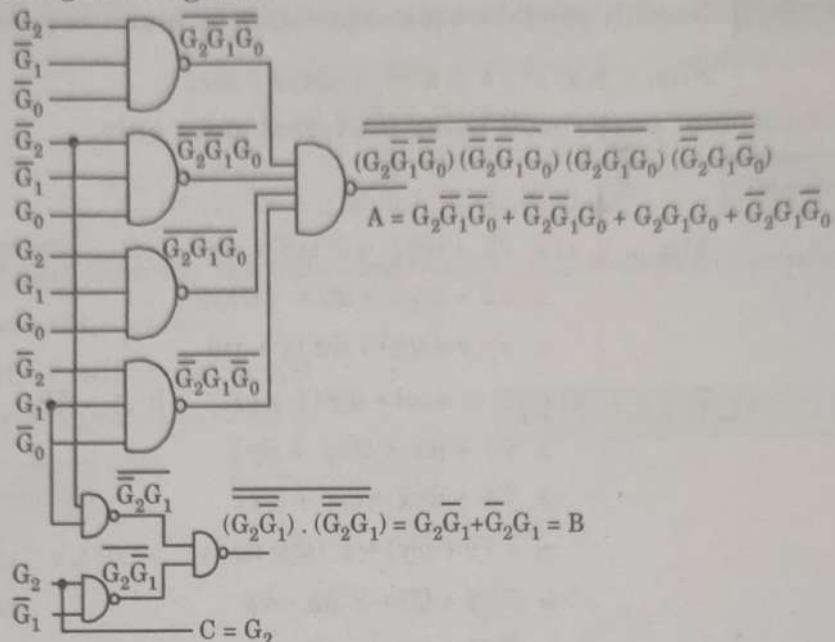


Fig. 1.7.2.

**PART-3**

*Review of Boolean Algebra and Demorgan's Theorem : SOP and POS Forms, Canonical Forms.*

**Questions-Answers**
**Long Answer Type and Medium Answer Type Questions**

**Que 1.8.** State De Morgan's theorem.

**Answer**

**First De Morgan theorem :** It states, complement of two or more variables and then AND operation on these is equivalent to NOR operation on these variables. (NOR means complement of two or more variables OR).

$$\overline{A_1 + A_2} = \overline{A_1} \cdot \overline{A_2}$$

**Second De Morgan theorem :** It states that complement of two or more variables and then OR operation on these is equivalent to a NAND operation on these variables (NAND means complement of two or more variables AND).

$$\overline{A_1 A_2 A_3} = \overline{A_1} + \overline{A_2} + \overline{A_3} \dots$$

**Que 1.9.** Simplify the following expression as much as possible :

$$F(w, x, y, z) = \bar{y}\bar{z} + \bar{w}\bar{x}\bar{z} + \bar{w}xy\bar{z} + wyz$$

and implement your result using universal gates only.

**Answer**

$$\begin{aligned}
 F(w, x, y, z) &= \bar{y}\bar{z} + \bar{w}\bar{x}\bar{z} + \bar{w}xy\bar{z} + wyz \\
 &= \bar{y}\bar{z} + wy\bar{z} + \bar{w}\bar{x}\bar{z} + \bar{w}xy\bar{z} \\
 &= \bar{z}(\bar{y} + wy) + \bar{w}\bar{z}(\bar{x} + xy) \\
 &= \bar{z}(\bar{y} + w) + \bar{w}\bar{z}(\bar{x} + y) \quad [\because \bar{A} + AB = \bar{A} + \bar{B}] \\
 &= \bar{y}\bar{z} + w\bar{z} + \bar{w}\bar{x}\bar{z} + \bar{w}y\bar{z} \\
 &= \bar{y}\bar{z} + \bar{w}y\bar{z} + w\bar{z} + \bar{w}\bar{x}\bar{z} \\
 &= \bar{z}(\bar{y} + \bar{w}y) + \bar{z}(w + \bar{w}\bar{x}) \\
 &= \bar{z}(\bar{y} + \bar{w}) + \bar{z}(w + \bar{x}) \\
 &= \bar{z}(\bar{y} + \bar{w} + w + \bar{x}) \\
 &= \bar{z}(\bar{y} + 1 + \bar{x}) \quad [\because 1 + A = 1] \\
 &= \bar{z}(\bar{y} + 1) \quad [\because 1 + \bar{A} = 1] \\
 &= \bar{z}
 \end{aligned}$$



Using NAND gate  
(a)



Using NOR gate  
(b)

Fig. 1.9.1.

**Que 1.10.** Simplify the following boolean equation :

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

**Answer**

$$\text{Given, } Y(A, B, C, D) = \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}\bar{C}\bar{D} = \bar{A}\bar{B}\bar{D}(C + \bar{C})$$

$$\text{since, } C + \bar{C} = 1$$

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

**Que 1.11.** Express the following boolean function  $F$  in a sum of minterms and a product of maxterms.

$$F(x, y, z) = (xy + z)(y + xz)$$

**Answer**

1. Given  $F(x, y, z) = (xy + z)(y + xz)$   
 $= xy.y + x.yxz + yz + xz.z$
2. By associative property,  $x.x = x$

$$\begin{aligned}
 F_m &= xy + xyz + yz + xz = xy(z + \bar{z}) + xyz + (x + \bar{x})yz + xz(y + \bar{y}) \\
 &= xyz + xy\bar{z} + xyz + xyz + \bar{x}yz + xyz + x\bar{y}z \\
 &= xyz + xy\bar{z} + \bar{x}yz + x\bar{y}z \quad (\because x + \bar{x} = x) \\
 F &= \Sigma m(3, 5, 6, 7) \\
 F_M &= \bar{F}_m = (\bar{x} + \bar{y} + \bar{z})(\bar{x} + \bar{y} + z)(\bar{x} + y + \bar{z})(x + \bar{y} + \bar{z}) \\
 F &= \prod M(0, 1, 2, 4)
 \end{aligned}$$

**Que 1.12.** Simplify the following boolean expression to a minimum number of literals.

- i.  $\bar{A}\bar{C} + ABC + A\bar{C} + A\bar{B}$
- ii.  $(\bar{x}\bar{y} + z) + z + xy + wz$

AKTU 2014-15, Marks 3.5

**Answer**

- i.  $\bar{A}\bar{C} + ABC + A\bar{C} + A\bar{B}$ :

$$\begin{aligned}
 \text{Let } Y &= \bar{A}\bar{C} + A\bar{C} + A\bar{B} + ABC \\
 &= \bar{C}(\bar{A} + A) + A(\bar{B} + BC) \\
 &= \bar{C} + A(\bar{B} + C) \quad [\because A + \bar{A} = 1] \\
 &= \bar{C} + AC + A\bar{B} \quad [\because \bar{C} + AC = (\bar{C} + A)(\bar{C} + C)] \\
 &= \bar{C} + A + A\bar{B} \\
 &= \bar{C} + A(1 + \bar{B}) \\
 &= A + \bar{C}
 \end{aligned}$$

- ii.  $(\bar{x}\bar{y} + z) + z + xy + wz$

$$\begin{aligned}
 \text{Let } Y &= (\bar{x}\bar{y} + z) + z + xy + wz \\
 &= \bar{x}\bar{y} + z + xy + wz \\
 &= \bar{x}\bar{y} + xy + z(1 + w) \\
 &= \bar{x}\bar{y} + xy + z = x \odot y + z \quad [\because 1 + w = 1]
 \end{aligned}$$

**Que 1.13.** Convert the given expression into canonical SOP form

$$Y = A + AB + BC$$

**Answer**

$$\begin{aligned}
 Y &= A(B + \bar{B})(C + \bar{C}) + AB(C + \bar{C}) + BC(A + \bar{A}) \\
 Y &= (AB + A\bar{B})(C + \bar{C}) + ABC + A\bar{B}\bar{C} + ABC + \bar{A}BC \\
 &= ABC + A\bar{B}\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + ABC + A\bar{B}\bar{C} + ABC + \bar{A}BC \\
 &= ABC + A\bar{B}\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + \bar{A}BC \\
 &\quad [\because A + A = A]
 \end{aligned}$$

**Que 1.14.** Convert the given expression into canonical POS form

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

**Answer**

$$\begin{aligned} Y &= (A + B\bar{B} + C\bar{C})(A + \bar{B} + C\bar{C})(A + B + \bar{C}) \\ &= (A + B\bar{B} + C)(A + B\bar{B} + \bar{C})(A + \bar{B} + C)(A + \bar{B} + \bar{C})(A + B + \bar{C}) \\ &= (A + B + C)(A + \bar{B} + C)(A + B + \bar{C})(A + \bar{B} + \bar{C}) \\ &\quad (A + \bar{B} + C)(A + \bar{B} + \bar{C})(A + B + \bar{C}) \\ Y &= (A + B + C)(A + \bar{B} + C)(A + B + \bar{C})(A + \bar{B} + \bar{C}) \end{aligned}$$

#### PART-4

Karnaugh Maps Upto 6 Variables.

#### Questions-Answers

#### Long Answer Type and Medium Answer Type Questions

**Que 1.15.** Write a short note on Karnaugh map. Also show the reduction of boolean expression and how to mark pairs. How gate-level minimization is implemented?

**Answer**

1. Karnaugh map is another way of presenting the information given by a truth table. These maps are also known by the name *K-map*. Let us consider the map for two variables. There may be four possible combinations within four squares.
2. Each square represents unique minterms as shown in Fig. 1.15.1 :

	B	$\bar{B}$	B
A	$\bar{A}\bar{B}$	$\bar{A}B$	$A\bar{B}$
	0	1	2
A	$A\bar{B}$	AB	
	3		

OR

	B	0	1
A	0	00	01
	0	0	1
1	10	11	3
	2		

Fig. 1.15.1.

For three variables :

1. There are eight minterms for three binary variables. Hence the *k-map* consists of eight squares.

2. The K-map drawn in Fig. 1.15.2, for three variables is marked with numbers in each row and each column to show the relationship between the squares and the three variables.

		BC	$\bar{B}C$	BC	BC	$B\bar{C}$	
		A	$m_0$	$m_1$	$m_3$	$m_2$	
		$\bar{A}$	$m_4$	$m_5$	$m_7$	$m_6$	
OR		A	BC	00	01	11	10
	0		000	001	011	010	02
	1		100	101	111	110	16

Fig. 1.15.2.

3. For example, the square assigned to  $m_5$ , which corresponds to row 1 and column 01. When these two numbers reconsidered, they give the binary number 101, whose decimal equivalent is 5.

For four variables :

- The map for boolean function of four binary variables require sixteen minterms, hence the map consists of sixteen squares.
- The listed terms are from 0 to 15, i.e., 16 minterms. The map shows the relationship with the four variables.
- In every square the numbers are written. The number denotes that this square corresponds to that number's minterm.

		CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$	
		AB	$m_0$	$m_1$	$m_3$	$m_2$	
		$\bar{A}\bar{B}$	$m_4$	$m_5$	$m_7$	$m_6$	
OR		AB	$m_{12}$	$m_{13}$	$m_{15}$	$m_{14}$	
	00	A $\bar{B}$	$m_8$	$m_9$	$m_{11}$	$m_{10}$	
	01		0000	0001	0011	0010	02
	11		0100	0101	0111	0110	06
	10		1100	1101	1111	1110	14

Fig. 1.15.3.

Que 1.16. Simplify the boolean function.

$$F(w, x, y, z) = \Sigma m(1, 3, 7, 11, 15)$$

which has the don't care conditions

$$d(w, x, y, z) = \Sigma d(0, 2, 5)$$

AKTU 2016-17, Marks 10

Answer

$$1. F(w, x, y, z) = \Sigma m(1, 3, 7, 11, 15)$$

and don't care conditions

$$d(w, x, y, z) = \Sigma d(0, 2, 5)$$

2. The minterms of  $F$  are the variable combinations that make the function equal to 1. The minterms of  $d$  are the don't care minterms that may be either 0 or 1.
3. The K-map simplification is shown in Fig. 1.16.1.

	yz 00	01	11	10
wx 00	x <sub>0</sub>	1 <sub>1</sub>	1 <sub>3</sub>	x <sub>2</sub>
01	0 <sub>4</sub>	x <sub>5</sub>	1 <sub>7</sub>	0 <sub>6</sub>
11	0 <sub>12</sub>	0 <sub>13</sub>	1 <sub>15</sub>	0 <sub>14</sub>
10	0 <sub>8</sub>	0 <sub>9</sub>	1 <sub>11</sub>	0 <sub>10</sub>

$$F = yz + \bar{w}x$$

(a)

	yz 00	01	11	10
wx 00	x <sub>0</sub>	1 <sub>1</sub>	1 <sub>3</sub>	x <sub>2</sub>
01	0 <sub>4</sub>	x <sub>5</sub>	1 <sub>7</sub>	0 <sub>6</sub>
11	0 <sub>12</sub>	0 <sub>13</sub>	1 <sub>15</sub>	0 <sub>14</sub>
10	0 <sub>8</sub>	0 <sub>9</sub>	1 <sub>11</sub>	0 <sub>10</sub>

$$F = yz + \bar{w}z$$

(b)

Fig. 1.16.1.

4. The minterms of  $F$  are marked by 1's, those of  $d$  are marked by x's and the remaining is filled with 0's.
5. To get the simplified expression in SOP form, we must include all five 1's in the map, but we may or may not include any of the x's, depending on the way the function is simplified.
6. In Fig. 1.16.1(a), don't care minterms 0 and 2 are included with the 1's, resulting as

$$F = yz + \bar{w}x$$

7. In Fig. 1.16.1(b), don't care minterm 5 is included with the 1's, resulting as

$$F = yz + \bar{w}z$$

8. The K-map in Fig. 1.16.1(b) is more feasible because, we have to use the minimum don't care.

**Que 1.17.** Simplify the following expression into product of sum (POS) form

- i.  $A\bar{B}\bar{C} + A\bar{B}D + BCD$   
ii.  $AC\bar{D} + \bar{C}D + A\bar{B} + ABCD$

AKTU 2014-15, Marks 3.5

**Answer**

- i.  $A\bar{B}\bar{C} + A\bar{B}D + BCD$

1. Let  $Y = A\bar{B}\bar{C} + A\bar{B}D + BCD$

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

$$= A\bar{B}\bar{C}D + A\bar{B}\bar{C}\bar{D} + A\bar{B}CD + A\bar{B}\bar{C}D + ABCD + \bar{A}BCD$$

$$Y = \Sigma m(7, 9, 11, 12, 13, 15)$$

2. Now for POS form we will take complement function,

$$\bar{Y} = \prod M(0, 1, 2, 3, 4, 5, 6, 8, 10, 14)$$

3. Minimization through K-map is shown in Fig. 1.17.1.

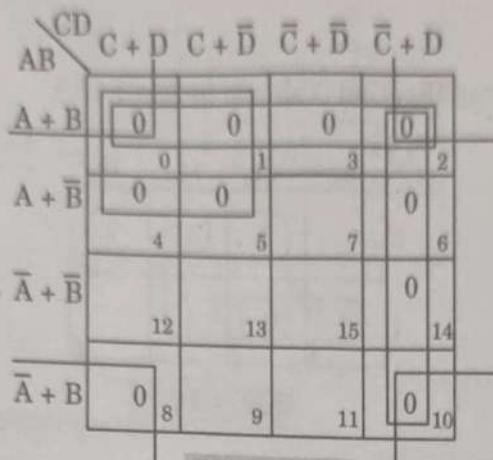


Fig. 1.17.1.

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

ii.  $AC\bar{D} + \bar{C}D + A\bar{B} + ABCD$

1. Let,  $Y = AC\bar{D} + \bar{C}D + A\bar{B} + ABCD$

$$\begin{aligned}
 &= A(B + \bar{B})C\bar{D} + (A + \bar{A})(B + \bar{B})\bar{C}D + A\bar{B}(C + \bar{C})(D + \bar{D}) + ABCD \\
 &= A(B + \bar{B})C\bar{D} + (A + \bar{A})(B + \bar{B})\bar{C}D + A\bar{B}(C + \bar{C})(D + \bar{D}) + ABCD \\
 &= ABC\bar{D} + A\bar{B}C\bar{D} + AB\bar{C}D + A\bar{B}\bar{C}D + \bar{A}B\bar{C}\bar{D} \\
 &\quad + \bar{A}\bar{B}\bar{C}D + A\bar{B}CD + A\bar{B}\bar{C}\bar{D} + ABCD
 \end{aligned}$$

$$Y = \sum m(1, 5, 8, 9, 10, 11, 13, 14, 15)$$

2. Now for POS form, we have to take complement function,

$$\bar{Y} = \prod M(0, 2, 3, 4, 6, 7, 12)$$

3. Minimization through K-map is shown in Fig. 1.17.2.

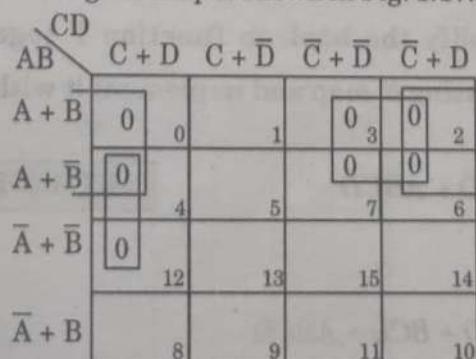


Fig. 1.17.2.

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

**Que 1.18.** Implement the following boolean function with NAND gates.  $F(x, y, z) = \Sigma m(1, 2, 3, 4, 5, 7)$

AKTU 2016-17, Marks 10

**Answer**

1. The K-map simplification is shown in Fig. 1.18.1.

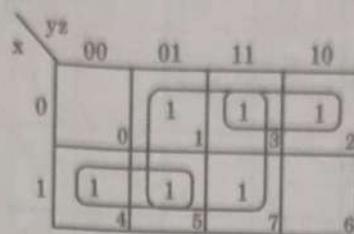


Fig. 1.18.1.

2. Hence, the simplified function is

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

3. Implementation using NAND gates is shown in Fig. 1.18.2.

$$\begin{aligned} F &= (\bar{x} + \bar{y}) \cdot (\bar{x} + y) \cdot (\bar{z}) \\ &= z + x\bar{y} + \bar{x} \cdot y \end{aligned}$$

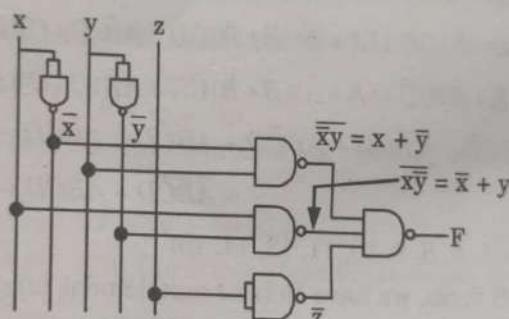


Fig. 1.18.2.

**Que 1.19.** Simplify the boolean function  $Y$  together with don't care condition  $d$  using K-map and implement it with two level NAND gate circuit.

$$Y = BD + BC\bar{D} + A\bar{B}CD$$

AKTU 2014-15, Marks 3.5

**Answer**

1. Given,  $Y = BD + BC\bar{D} + A\bar{B}CD$

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

$$= ABCD + \bar{A}BCD + AB\bar{C}D + \bar{A}B\bar{C}D + ABC\bar{D} + \bar{A}BC\bar{D} + A\bar{B}CD$$

$$Y = \Sigma m(5, 6, 7, 10, 13, 14, 15)$$

2. As there is not given any don't care condition so K-map is as shown in Fig. 1.19.1.

	CD \ AB	00	01	11	10
00	0	1	3	2	
01	4	1 5	1 7	1 6	
11	12	13	15	14	
10	8	9	11	10	

Fig. 1.19.1.

$$Y = BD + BC + ACD$$

3. NAND gate implementation :

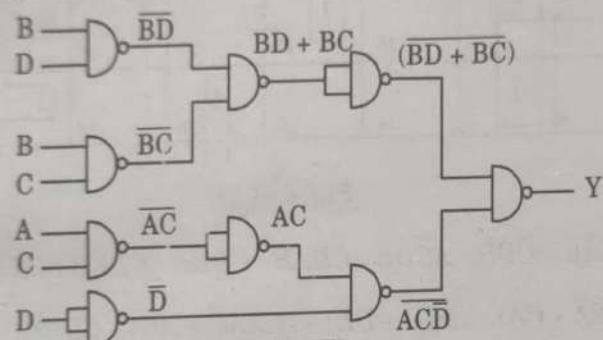


Fig. 1.19.2.

Que 1.20. Minimize the given boolean function using K-map.

$$F(A, B, C, D) = \Sigma m(3, 4, 5, 7, 9, 13, 14, 15)$$

AKTU 2018-19, Marks 07

Answer

	CD \ AB	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	$CD$
$\bar{A}\bar{B}$	0	1		1	3
$\bar{A}B$	1	1	1		6
$A\bar{B}$	12	1	13	15	14
$AB$	8	1	9	11	10

Fig. 1.20.1.

$$F = \bar{A}\bar{B}\bar{C} + \bar{A}CD + A\bar{C}D + ABC$$

**Que 1.21.** Simplify the following Boolean function using K-map  
 $Y = \Sigma m(0, 1, 3, 5, 6, 7, 9, 11, 16, 18, 19, 20, 21, 22, 24, 26)$

AKTU 2017-18, Marks 07

**Answer**

**K-map :**

$f(C, D, E, A, B)$

		CDE	000	001	011	010	110	111	101	100
		AB	00	01	11	10	11	10	11	10
C	D	0	1	0	4	12	8	1	24	28
		1	1	1	5	13	9	25	29	20
E	A	0	1	1	3	7	15	11	27	31
		1	2	1	6	14	10	1	26	30
		B	0	1	0	1	0	1	0	1

Fig. 1.21.1

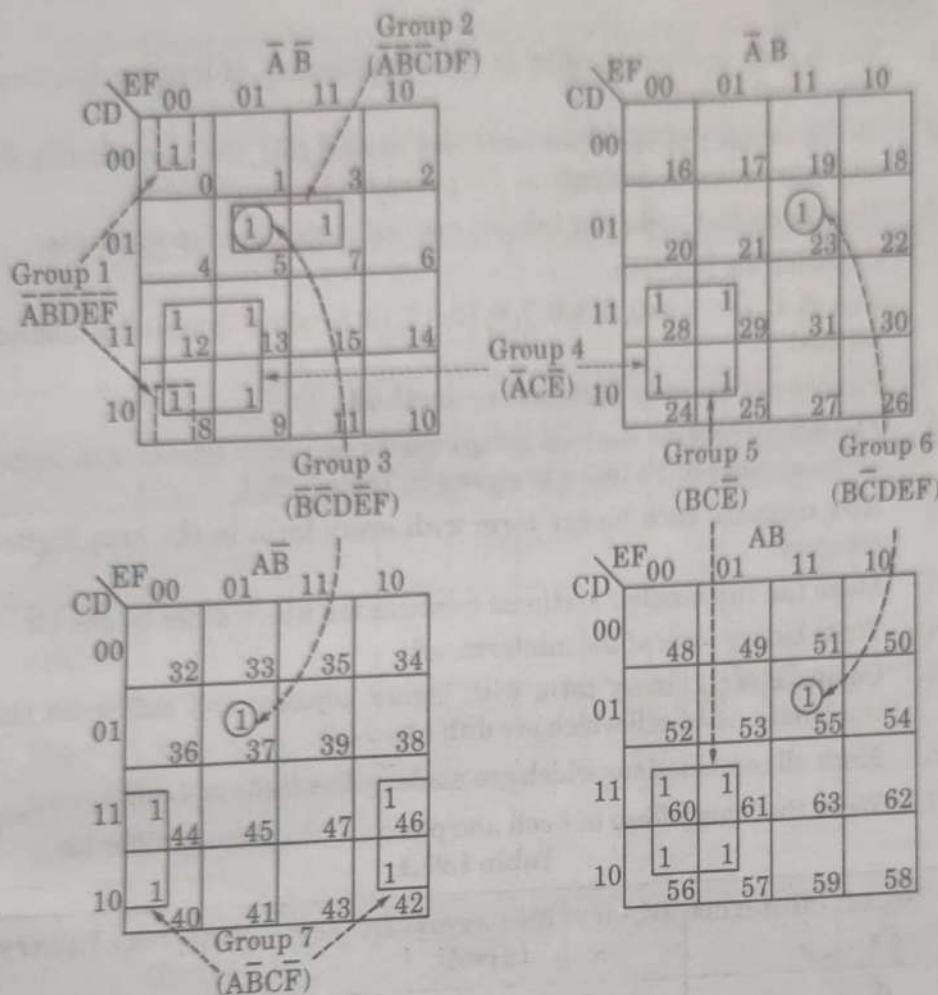
$$\begin{aligned}
 Y &= \bar{D}\bar{E}A\bar{B} + \bar{C}\bar{D}B + \bar{A}\bar{C}\bar{D}E + C\bar{D}A\bar{B} + \bar{C}D\bar{E}B + C\bar{D}\bar{E}\bar{B} + C\bar{D}E\bar{A} + C\bar{D}\bar{E}A \\
 &= \bar{D}\bar{B}(\bar{E}\bar{A} + CA) + \bar{C}B(\bar{D} + D\bar{E}) + \bar{D}E(A\bar{C} + \bar{A}C) + C\bar{E}(D\bar{B} + \bar{D}A) \\
 &= \bar{D}\bar{B}(\bar{E}\bar{A} + CA) + \bar{C}B(\bar{D} + \bar{E}) + \bar{D}E(A \oplus C) + C\bar{E}(D\bar{B} + \bar{D}A)
 \end{aligned}$$

**Que 1.22.** Simplify the Boolean function

$F(A, B, C, D, E, F) = \Sigma m(0, 5, 7, 8, 9, 12, 13, 23, 24, 25, 28, 29, 37, 40, 42, 44, 46, 55, 56, 57, 60, 61)$

**Answer**

1. Group 1 and group 2 are two pairs of 1's in the first 16-cell map.
2. Group 3 is formed by two isolated 1's from first 16-cell map and third 16-cell map.
3. Group 4 is a combination of two quads from first 16-cell and second 16-cell map.
4. Similarly group 5 is a combination of two quads from second 16-cell map and fourth 16-cell map.
5. Group 6 is again a combination of isolated 1's from second and fourth 16-cell maps.
6. Finally group 7 is a quad within the third 16-cell map.



The expression is,

$$F = \overline{ABDEF} + \overline{ABCDF} + \overline{BCDEF} \\ + \overline{ACE} + B\bar{C}\bar{E} + B\bar{C}DEF + A\bar{B}CF$$

### PART-5

Tabulation (Quine-McCluskey) Method.

#### Questions-Answers

Long Answer Type and Medium Answer Type Questions

**Que 1.23.** What is the significance of Quine-McCluskey method or tabular method?

**Answer**

1. The K-map method is suitable for simplification of boolean function upto 5 or 6 variables.
  2. As the number of variables increases beyond this, the visualization of adjacent square is difficult, as the geometry is more involved.
  3. The Quine-McCluskey or tabular method is employed in such cases.
- Consider the function,

$F(A, B, C, D) = \Sigma m(0, 2, 3, 6, 7, 8, 10, 12, 13)$  for simplifying using tabular method.

**Process of solving McCluskey method :**

1. The binary representations are grouped a section of numbers in terms of the number of 1's index as shown in Table 1.23.1.
2. Now compare each binary term with every term in the next higher category.
3. Make the two number sectional combination which differ by one bit.
4. Write binary form of the minterm cell.
5. Compare each binary term with higher adjacent cell and write the combination of 4 cell which are differ by 1-bit.
6. Mark all combinations which are made by the digits of 4 cell.
7. Write the binary form of 4 cell and place a ( ) in place of differ bit.

**Table 1.23.1.**

No. of 1's	Minterms	Binary	Minterms (2 cell)	Binary	Minterm (4 cell)	Binary
0	$m_0$	0 0 0 0	0, 2✓ 0, 8✓	0 0 _ 0 _ 0 0 0	0, 2, 8, 10	_ 0 _ 0
1	$m_2$	0 0 1 0	2, 3✓	0 0 1 _	2, 3, 6, 7	0 _ 1 _
			2, 6✓ 2, 10✓	0 _ 1 0 _ 0 1 0		
2	$m_6$	1 0 0 0	8, 10✓	1 0 _ 0		
			8, 12	1 _ 0 0		
3	$m_7$	0 1 1 1	3, 7✓	0 _ 1 1		
			6, 7✓	0 1 1 _		
			12, 13	1 1 0 _		
	$m_{13}$	1 1 0 1				

8. Apply same process to the resultant stage.
9. All the terms which remain unchecked are the PI's. Now prepare a PI chart to determine essential prime implicants.
10. All the PI's are represented in rows and each minterm of function in a column as shown in Table 1.23.2.
11. Put the  $\odot$  in each row to show the composition of minterms that make PI's.

Table 1.23.2.

Minterms	Prime implicant	$m_0$	$m_2$	$m_3$	$m_6$	$m_7$	$m_8$	$m_{10}$	$m_{12}$	$m_{13}$
$A\bar{C}\bar{D}$	8, 12						$\odot$		$\odot$	
$AB\bar{C} \checkmark$	12, 13								$\odot$	$\odot$
$\bar{B}\bar{D} \checkmark$	0, 2, 8, 10	$\odot$	$\odot$				$\odot$	$\odot$		
$\bar{A}C \checkmark$	2, 3, 6, 7		$\odot$	$\odot$	$\odot$	$\odot$				

13. The column that contains a single dot  $\odot$  is essential prime implicant.
14. A tick mark is put above each column which has only one  $\odot$  mark.
15. The sum of all EPI's gives the function in its minimal SOP form.

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

**Que 1.24.** Minimize the following switching function using Quine-McCluskey method.

$$F(x_1, x_2, x_3, x_4, x_5) = \Sigma m(0, 1, 2, 8, 9, 15, 17, 21, 24, 25, 27, 31)$$

AKTU 2016-17, Marks 15

## Answer

Table 1.24.1. For obtaining all the prime implicants.

No. of I's	Minterms	Binary	Minterms (2 cell)	Binary	Minterm (4 cell)	Binary
0	$m_0$	00000	0, 1 ✓	0000-	0, 1, 8, 9	0-00-C
1	$m_1$	00001	0, 2 H	000-0	1, 9, 17, 25	-001B
	$m_2$	00010	0, 8 ✓	0-000	8, 9, 24, 25	-100-A
	$m_8$	01000				
2	$m_9$	01001	1, 9 ✓	0-001		
	$m_{17}$	10001	1, 17 ✓	-0001		
	$m_{24}$	11000	8, 9 ✓	0100-		
			8, 24 ✓	-1000		
3	$m_{21}$	10101	9, 25 ✓	-1001		
	$m_{25}$	11001	17, 21 G	10-01		
4	$m_{15}$	01111	17, 25 ✓	1-001		
	$m_{27}$	11011	24, 25 ✓	1100-		
5	$m_{31}$	11111	25, 27 F	110-1		
			15, 31 E	-1111		
			27, 31 D	11-11		

Table 1.24.2. Prime-implicant

Minterm	$m_0$	$m_1$	✓ $m_2$	$m_8$	$m_9$	✓ $m_{15}$	$m_{17}$	✓ $m_{21}$	$m_{24}$	$m_{25}$	$m_{27}$	$m_{31}$
A				○	○				○	○		
B		○			○		○			○		
C	○	○		○	○							
D											○	○
E ✓						○						
F									○	○		
G ✓							○	○				
H ✓	○		○									

So the essential prime implicants are,

$$F(x_1, x_2, x_3, x_4, x_5) = E + G + A + H = x_2 \bar{x}_3 x_4 x_5 + x_1 \bar{x}_2 \bar{x}_4 x_5 + \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_5 + x_2 \bar{x}_3 \bar{x}_4$$

**Que 1.25.** Minimize the following using Quine-McCluskey method :

$$F(W, X, Y, Z) = \Sigma m(0, 3, 5, 6, 7, 10, 12, 13) + \Sigma d(2, 9, 15)$$

AKTU 2015-16, Marks 15

**Answer**

1.  $F(W, X, Y, Z) = \Sigma m(0, 3, 5, 6, 7, 10, 12, 13) + \Sigma d(2, 9, 15)$

First, we group the minterms according to the numbers of 1's.

No. of 1's	Minterms	Binary	Minterms (2 cell)	Binary	Minterm (4 cell)	Binary
0	$m_0$	0000	0, 2*	0 0 _ 0	2*, 3, 6, 7	0 _ 1 _
1	$dm_2$	0010	2*, 3✓	0 0 1 _	5, 7, 13, 15*	_ 1 _ 1
2	$m_3$ $m_5$ $m_6$ $m_{10}$ $m_{12}$ $dm_9$	0011 0101 0110 1010 1100 1001	2*, 6✓ 2*, 10 3, 7✓ 5, 7✓ 5, 13✓ 6, 7✓	0 _ 1 0 _ 0 1 0 0 _ 1 1 0 1 _ 1 _ 1 0 1 0 1 1 _		
3	$m_7$ $m_{13}$	0111 1101	12, 13 9*, 13	1 1 0 _ 1 _ 0 1		
4	$dm_{15}$	1111	7, 15*✓ 13, 15*✓	_ 1 1 1 1 1 _ 1		

2. Then, we prepare the table of prime implicants.

Minterm	Prime implicant	$m_0$	$m_3$	$m_5$	$m_6$	$m_7$	$m_{10}$	$m_{12}$	$m_{13}$
$\bar{W}\bar{X}\bar{Z} \checkmark$	0, 2*	○							
$\bar{X}Y\bar{Z} \checkmark$	2*, 10						○		
$WX\bar{Y} \checkmark$	12, 13							○	○
$W\bar{Y}Z$	9*, 13								○
$\bar{W}Y \checkmark$	2*, 3, 6, 7		○		○	○			
$XZ \checkmark$	5, 7, 13, 15*			○		○			○

$$F(W, X, Y, Z) = \bar{W}Y + XZ + \bar{W}\bar{X}\bar{Z} + \bar{X}Y\bar{Z} + WX\bar{Y}$$

**Que 1.26.** Use Quine-McCluskey (QM) method to solve the following function :

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

AKTU 2014-15, Marks 3.5

**Answer**

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

No. of 1's	Minterms	Binary	Minterms (2 cell)	Binary	Minterm (4 cell)	Binary
1.	$m_8$	1 0 0 0	8, 9✓	1 0 0 _	8, 9, 10, 11	10 _ _
2.	$m_5$	0 1 0 1	8, 10✓	1 0 _ 0	10, 11, 14, 15	1 _ 1 _
	$m_9$	1 0 0 1	5, 7	0 1 _ 1		
	$m_{10}$	1 0 1 0	9, 11✓	1 0 _ 1		
3.	$m_7$	0 1 1 1	10, 11✓	1 0 1 _		
	$m_{11}$	1 0 1 1	10, 14✓	1 _ 1 0		
	$m_{14}$	1 1 1 0	7, 15	_ 1 1 1		
4.	$m_{15}$	1 1 1 1	11, 15✓	1 _ 1 1		
			14, 15✓	1 1 1 _		

2. All the terms which are unchecked are prime implicants.

Now, we prepare a prime implicant chart to determine essential prime implicant is as follows :

Minterms	Prime implicants	✓ $m_5$	✓ $m_7$	✓ $m_8$	✓ $m_9$	✓ $m_{10}$	✓ $m_{11}$	✓ $m_{14}$	✓ $m_{15}$
$\bar{A}BD$ ✓	5, 7	○	○						
$BCD$ ✓	7, 15		○						○
$A\bar{B}$ ✓	8, 9, 10, 11			○	○	○	○		
$AC$ ✓	10, 11, 14, 15					○	○	○	○

Therefore,  $F(A, B, C, D) = AC + A\bar{B} + \bar{A}BD$

## 3. Logic diagram :

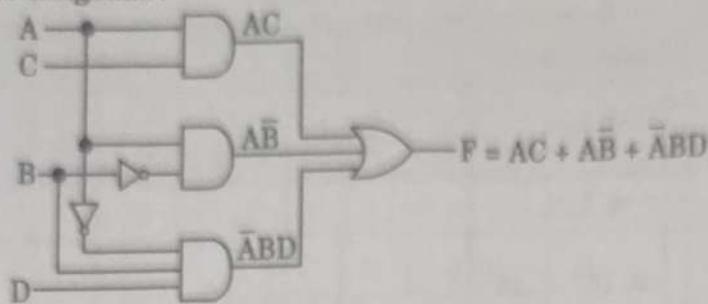


Fig. 1.26.1.

**Que 1.27.** Use the Quine-McCluskey method to generate the set of prime implicants for the following function :

$$F(A, B, C, D) = \sum m(0, 1, 4, 5, 6, 7, 9, 11, 15) + \sum \phi(10, 14).$$

Also obtain all minimal expressions for the function. Draw a logic diagram using only NAND gates to implement your best solution obtained.

**Answer**

1. Given,  $F(A, B, C, D) = \sum m(0, 1, 4, 5, 6, 7, 9, 11, 15) + \sum \phi(10, 14)$

No. of 1's	Minterms	Binary	Minterms (2 cell)	Binary	Minterm (4 cell)	Binary
0	$m_0$	0 0 0 0	0, 1 ✓	0 0 0 _	0, 1, 4, 5	0 _ 0 _
1	$m_1$	0 0 0 1	0, 4 ✓	0 _ 0 0	4, 5, 6, 7	0 1 _ _
	$m_4$	0 1 0 0	1, 5 ✓	0 _ 0 1	6, 14*, 7, 15	_ 1 1 _
2	$m_5$	0 1 0 1	1, 9	_ 0 0 1		
	$m_6$	0 1 1 0	4, 5 ✓	0 1 0 _		
	$m_9$	1 0 0 1	4, 6 ✓	0 1 _ 0		
	$dm_{10}$	1 0 1 0	5, 7 ✓	0 1 _ 1		
3	$m_7$	0 1 1 1	6, 7 ✓	0 1 1 _		
	$m_{11}$	1 0 1 1	6, 14* ✓	_ 1 1 0		
	$dm_{14}$	1 1 1 0	9, 11	1 0 _ 1		
4	$m_{15}$	1 1 1 1	10*, 11	1 0 1 _		
			7, 15 ✓	_ 1 1 1		
			11, 15	1 _ 1 1		
			14*, 15	1 1 1 _		

Minterms	Prime implicants	$m_0$	$m_1$	$m_4$	$m_5$	$m_6$	$m_7$	$m_9$	$m_{11}$	$m_{13}$
$A'C' \checkmark$	0, 1, 4, 5	○	○	○	○					
$A'B$	4, 5, 6, 7				○	○	○	○		
$BC$	6, 14*, 7, 15						○	○		
$B'C'D$	1, 9			○					○	
$AB'C'$	9, 11								○	○
$AB'C'$	10*, 11								○	
$ACD$	11, 15								○	
$ABC$	14*, 15								○	○

The essential prime implicant,

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

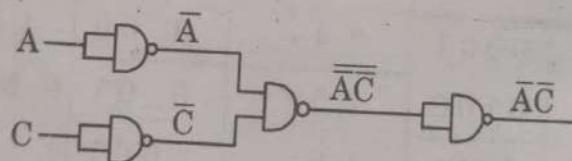


Fig. 1.27.1.

**Que 1.28.** Minimize the following using Quine-McCluskey method :

$$F(A, B, C, D) = \Sigma m(0, 1, 9, 15, 24, 29, 30) + \Sigma d(8, 11, 31)$$

AKTU 2018-19, Marks 07

**Answer**

- Arrange minterms according to categories of 1's as shown in table 1.28.1 :

Table 1.28.1.

No. of 1's	Min- terms	Binary					Minterms (2 cell)	Binary					Min- terms (4 cell)	Binary						
		A	B	C	D	E		A	B	C	D	E			A	B	C	D	E	
0	$m_0$	0	0	0	0	0	0, 1 ✓	0	0	0	0	-	0, 1, 8*, 9	0	-	0	0	-		
1	$m_1$	0	0	0	0	1	0, 8* ✓	0	-	0	0	0		0	-	0	0	-		
	$dm_8$	0	1	0	0	0	1, 9 ✓	0	-	0	0	1			0	-	0	0	-	
2	$m_9$	0	1	0	0	1	8*, 9 ✓	0	1	0	0	-			0	-	0	0	-	
	$m_{24}$	1	1	0	0	0	8*, 24	-	1	0	0	0				0	-	0	0	-
3	$dm_{11}$	0	1	0	1	1	9, 11*	0	1	0	-	1	0, 1, 8*, 9	0	-	0	0	-		
4	$m_{15}$	0	1	1	1	1	11*, 15	0	1	-	1	1			0	-	0	0	-	
	$m_{29}$	1	1	1	0	1	15, 31*	-	1	1	1	1				0	-	0	0	-
	$m_{30}$	1	1	1	1	0	29, 31*	1	1	1	-	1				0	-	0	0	-
5	$dm_{31}$	1	1	1	1	1	30, 31*	1	1	1	1	-				0	-	0	0	-

2. List of prime implicants :

Table 1.28.2.

Prime implicants	Binary representation				
	A	B	C	D	E
8*, 24	-	1	0	0	0
9, 11*	0	1	0	-	1
11*, 15	0	1	-	1	1
15, 31*	-	1	1	1	1
29, 31*	1	1	1	-	1
30, 31*	1	1	1	1	-
0, 1, 8*, 9	0	-	0	0	-

3. Select the minimum number of prime implicants which must cover all the minterms except don't care minterms.

Table 1.28.3.

Prime implicant	Minterm						
	✓ 0	✓ 1	9	15	✓ 24	✓ 29	✓ 30
8*, 24 ✓					◎		
9, 11*			◎				
11*, 15				◎			
15, 31*				◎			
29, 31* ✓					◎		
30, 31* ✓						◎	
0, 1, 8*, 9 ✓	◎	◎	◎				

$$Y = \overline{BCDE} + ABCE + ABCD + \overline{ACD}$$

### VERY IMPORTANT QUESTIONS

*Following questions are very important. These questions may be asked in your SESSIONALS as well as UNIVERSITY EXAMINATION.*

**Q. 1.** Design a combinational circuit that converts a BCD code to excess-3 code.

**Ans.** Refer Q. 1.4, Page 1-6B, Unit-1.

**Q. 2.** Simplify the following boolean expression to a minimum number of literals.

i.  $\bar{A}\bar{C} + ABC + A\bar{C} + A\bar{B}$

ii.  $(\bar{x}\bar{y} + z) + z + xy + wz$

**Ans.** Refer Q. 1.12, Page 1-14B, Unit-1.

**Q. 3.** Simplify the boolean function.

$$F(w, x, y, z) = \Sigma m(1, 3, 7, 11, 15)$$

which has the don't care conditions

$$d(w, x, y, z) = \Sigma d(0, 2, 5)$$

**Ans.** Refer Q. 1.16, Page 1-16B, Unit-1.

- Q. 4. Simplify the boolean function  $Y$  together with don't care condition  $d$  using K-map and implement it with two level NAND gate circuit.

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

Ans. Refer Q. 1.19, Page 1-19B, Unit-1.

- Q. 5. Simplify the following Boolean function using K-map

$$Y = \Sigma m(0, 1, 3, 5, 6, 7, 9, 11, 16, 18, 19, 20, 21, 22, 24, 26)$$

Ans. Refer Q. 1.21, Page 1-21B, Unit-1.

- Q. 6. Minimize the following switching function using Quine-McCluskey method.

$$F(x_1, x_2, x_3, x_4, x_5) = \Sigma m(0, 1, 2, 8, 9, 15, 17, 21, 24, 25, 27, 31)$$

Ans. Refer Q. 1.24, Page 1-24B, Unit-1.

- Q. 7. Use Quine-McCluskey (QM) method to solve the following function :

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

Ans. Refer Q. 1.26, Page 1-27B, Unit-1.



# 2

UNIT

MSI Devices

## CONTENTS

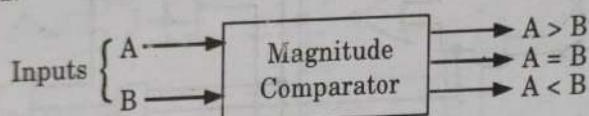
Part-1 :	MSI Devices Like Comparators .....	2-2B to 2-4B
Part-2 :	Multiplexers .....	2-5B to 2-11B
Part-3 :	Encoder .....	2-12B to 2-14B
Part-4 :	Decoder .....	2-14B to 2-19B
Part-5 :	Driver and Multiplexed Display .....	2-19B to 2-20B
Part-6 :	Half and Full Adders .....	2-20B to 2-23B
Part-7 :	Subtractors .....	2-23B to 2-26B
Part-8 :	Serial and Parallel Adders .....	2-26B to 2-27B
Part-9 :	BCD Adder .....	2-27B to 2-28B
Part-10 :	Barrel Shifter and ALU .....	2-29B to 2-30B

**PART-1***MSI Devices Like Comparators.***Questions-Answers****Long Answer Type and Medium Answer Type Questions**

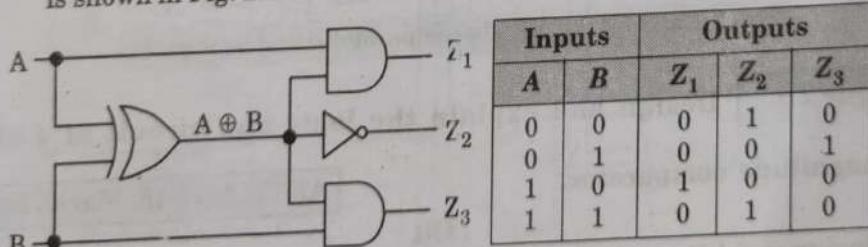
**Que 2.1.** What is magnitude comparator ? Design a three bit comparator circuit using logic gates.

**Answer**

1. A magnitude comparator is a combinational circuit designed primarily to compare the relative magnitude of the two binary numbers  $A$  and  $B$ .
2. Naturally, the result of this comparison is specified by three binary variables that indicate, whether  $A > B$ ,  $A = B$  or  $A < B$ .
3. The block diagram of a single bit magnitude comparator is shown in Fig. 2.1.1.

**Fig. 2.1.1.**

4. EX-OR and AND gate is used to implement the circuit. If the EX-OR gate and two AND gates are combined, the circuit will function as a single bit magnitude comparator as shown in Fig. 2.1.2.
5. The circuit diagram and truth table of a single bit magnitude comparator is shown in Fig. 2.1.2.

**Fig. 2.1.2.**

$Z_1$  is high when  $A > B$ ,

$Z_2$  is high when  $A = B$ ,

$Z_3$  is high when  $A < B$ .

The same concept is adopted to form an  $n$ -bit magnitude comparator  
3-bit magnitude comparator :

$$A = A_2 A_1 A_0$$

$$B = B_2 B_1 B_0$$

Two number  $A$  and  $B$  are equal, only if all the pairs of significant digits are equal. i.e.,

$$A_2 = B_2, A_1 = B_1, A_0 = B_0$$

When numbers are binary, then equality relation of each pair of bits can be expressed by the equivalent function as,

$$x_i = A_i B_i + \bar{A}_i \bar{B}_i, i = 0, 1, 2$$

**Design procedure :**

- i.  $(A = B) = x_2 \cdot x_1 \cdot x_0 = (A_2 \odot B_2)(A_1 \odot B_1)(A_0 \odot B_0)$
- ii.  $(A > B) = A_2 \bar{B}_2 + x_2 A_1 \bar{B}_1 + x_2 x_1 A_0 \bar{B}_0$
- iii.  $(A < B) = \bar{A}_2 B_2 + x_2 \bar{A}_1 B_1 + x_2 x_1 \bar{A}_0 B_0$

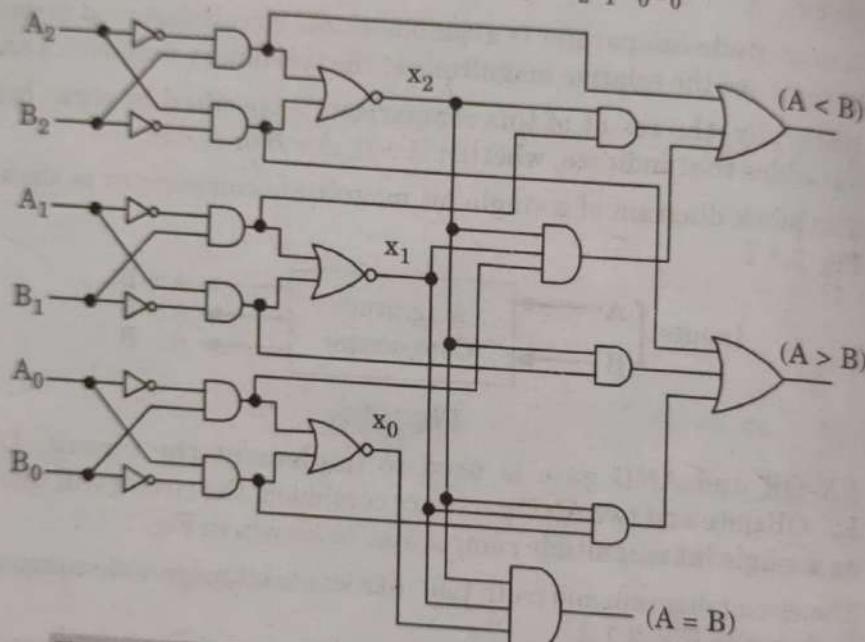


Fig. 2.1.3. 3-bit magnitude comparator using logic gates.

**Que 2.2.** Design and explain the logic and circuit of 4-bit magnitude comparator.

AKTU 2014-15, Marks 06

OR  
Design a 4-bit magnitude comparator using one bit comparator modules.

AKTU 2015-16, Marks 10

OR  
Draw and explain 4-bit magnitude comparator.

AKTU 2017-18, Marks 07

**Answer**

1. Let two numbers  $A$  and  $B$  with four digits each.

$$A = A_3 A_2 A_1 A_0$$

$$B = B_3 B_2 B_1 B_0$$

2. The two numbers are equal if all pairs of significant digits are equal, i.e., if  $A_3 = B_3, A_2 = B_2, A_1 = B_1$  and  $A_0 = B_0$ . Equality relation is generated by EX-NOR gate.

$$x_i = A_i B_i + \bar{A}_i \bar{B}_i ; i = 0, 1, 2, 3.$$

where  $x_i$  is equality of two numbers

$$x_i = 1, \text{ if } A = B$$

$$x_i = 0, \text{ otherwise,}$$

$$(A = B) = x_3 x_2 x_1 x_0 = 1, \text{ if all pairs are equal.}$$

3. To determine if  $A > B$  or  $A < B$ ,

$$(A > B) = A_3 \bar{B}_3 + x_3 A_3 \bar{B}_2 + x_3 x_2 A_1 \bar{B}_1 + x_3 x_2 x_1 A_0 \bar{B}_0$$

$$(A < B) = \bar{A}_3 B_3 + x_3 \bar{A}_2 B_2 + x_3 x_2 \bar{A}_1 B_1 + x_3 x_2 x_1 \bar{A}_0 B_0$$

4. The logical implementation is shown in Fig. 2.2.1.

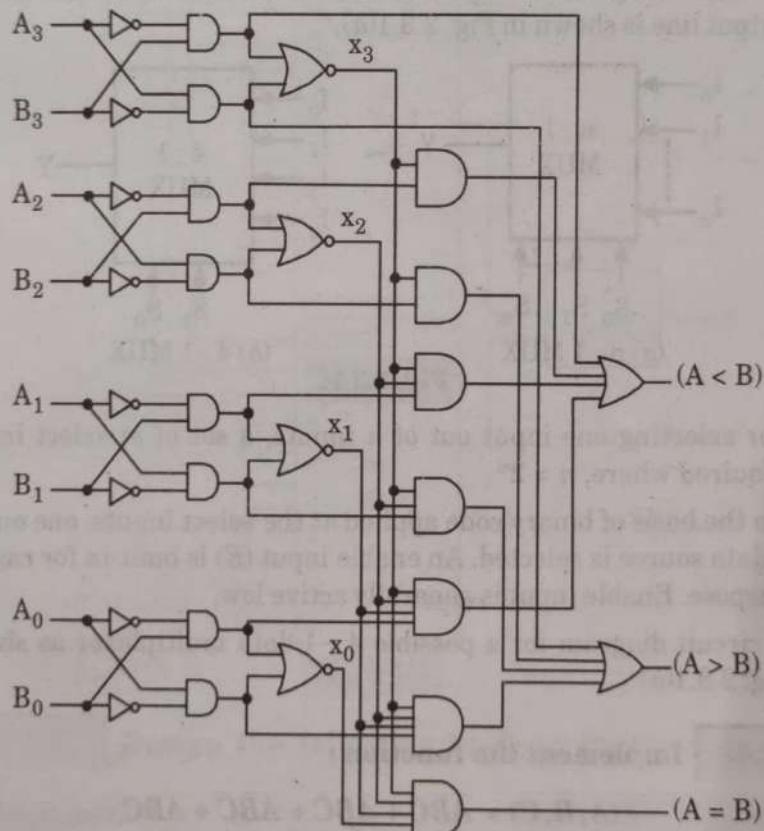


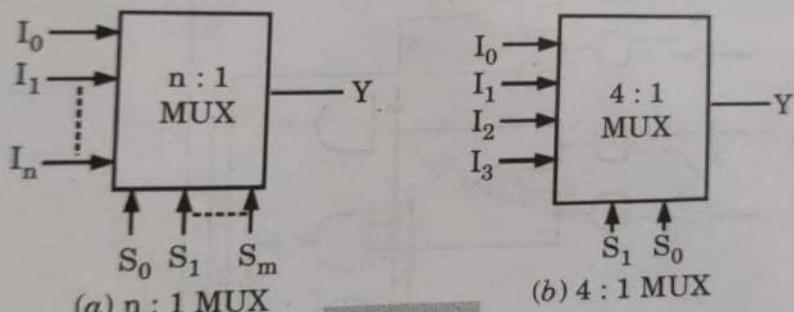
Fig. 2.2.1. 4-bit magnitude comparator using logic gates.

**PART-2***Multiplexers.***Questions-Answers****Long Answer Type and Medium Answer Type Questions**

**Que 2.3.** What is the role of multiplexer in the digital electronics? Explain the logic how it selects a one input among several inputs.

**Answer**

1. A multiplexer (MUX) is a combinational circuit that selects one input out of several inputs and directs it to a single output.
2. The particular input selection is controlled by a set of select inputs.
3. The block diagram of a digital multiplexer with  $n$  input lines and single output line is shown in Fig. 2.3.1(a).

**Fig. 2.3.1.**

4. For selecting one input out of  $n$  inputs, a set of  $m$  select inputs is required where,  $n = 2^m$ .
5. On the basis of binary code applied at the select inputs, one output of  $n$  data source is selected. An enable input ( $E$ ) is built-in for cascading purpose. Enable input is generally active low.
6. A circuit diagram for a possible 4 : 1 data multiplexer as shown in Fig. 2.3.1(b).

**Que 2.4.** Implement the function :

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

Using 4 : 1 multiplexer using  $B$  and  $C$  variables to the selection lines.

**Answer**

1. Given,  $F(A, B, C) = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$   
 2. Implementation using 4 : 1 MUX,  $F(A, B, C) = \Sigma m(1, 2, 4, 7)$

Input	$I_0$	$I_1$	$I_2$	$I_3$
$A'$	0	(1)	(2)	3
A	(4)	5	6	(7)

Output to multiplexer      A      A'      A'      A

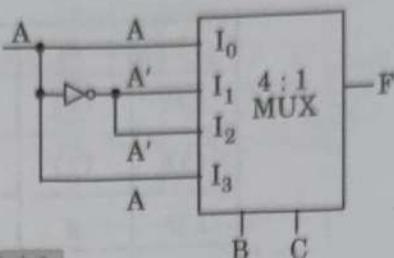


Fig. 2.4.1.

**Que 2.5.** Construct a  $16 \times 1$  multiplexer with two  $8 \times 1$  and one  $2 \times 1$  multiplexer. Use block diagrams.

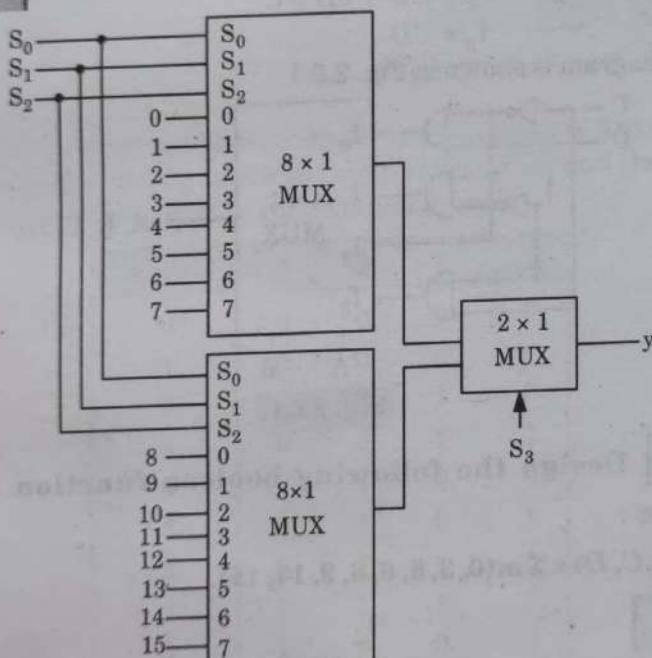
**Answer**

Fig. 2.5.1.

**Que 2.6.** Design the following boolean function using  $4 \times 1$  multiplexer.  
**AKTU 2014-15, Marks 06**

$$F(A, B, C, D) = \Sigma m(0, 1, 3, 4, 8, 9, 15)$$

**Answer**

1. Given,  $F(A, B, C, D) = \sum m(0, 1, 3, 4, 8, 9, 15)$
2. We have to design it using 4 : 1 multiplexer, so we can use two variable  $(A, B)$  for select lines and implementation table is as follows :

	$I_0$	$I_1$	$I_2$	$I_3$
$\bar{C}\bar{D}$	①	④	⑧	12
$\bar{C}D$	②	5	⑨	13
$C\bar{D}$	2	6	10	14
$CD$	③	7	11	⑯
	$\bar{C} + D$	$\bar{C}\bar{D}$	$\bar{C}$	$CD$

3. Now,

$$I_0 = \bar{C}\bar{D} + \bar{C}D + CD = \bar{C} + D$$

$$I_1 = \bar{C}\bar{D}$$

$$I_2 = \bar{C}\bar{D} + \bar{C}D = \bar{C}$$

$$I_3 = CD$$

4. Logic diagram is shown in Fig. 2.6.1

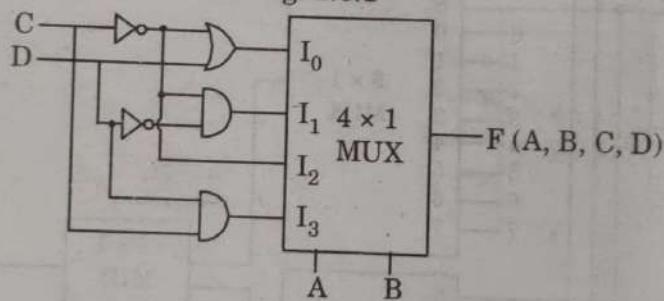


Fig. 2.6.1.

**Que 2.7.** Design the following boolean function using the multiplexer :

$$F(A, B, C, D) = \sum m(0, 3, 5, 6, 8, 9, 14, 15).$$

**Answer**

$$F(A, B, C, D) = \sum m(0, 3, 5, 6, 8, 9, 14, 15)$$

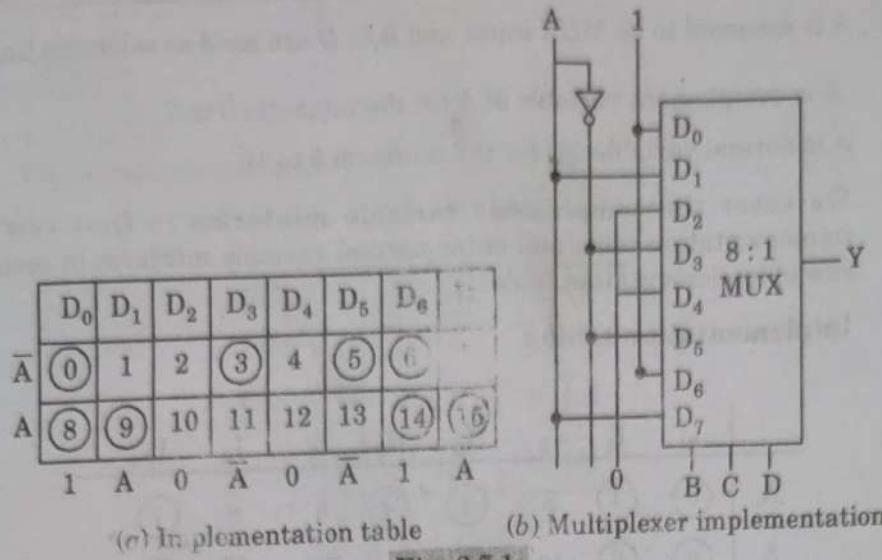


Fig. 2.7.1.

Que 2.8. Implement the function  $F = \Sigma m(0, 1, 3, 4, 7, 8, 9, 11, 14, 15)$

using 8:1 MUX.

AKTU 2017-18, Marks 07

Answer

- The given Boolean function is a four variable function. Any one variable of the function can be taken as input to the MUX and the remaining variables are connected to the selection lines.

Decimal	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

2. A is assumed to be MUX input and B,C,D are used as selection lines.
- $\bar{A}$  is complement variable of A for the minterm 0 to 7.
  - A is normal variable (A) for the minterm 8 to 15.
3. We enter the complement variable minterms in first row of implementation table and enter normal variable minterm in second row of implementation table.

Implementation table :

	I <sub>0</sub>	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>	I <sub>4</sub>	I <sub>5</sub>	I <sub>6</sub>	I <sub>7</sub>
$\bar{A}$	(0)	(1)	2	(3)	(4)	5	6	(7)
A	(8)	(9)	10	(11)	12	13	(14)	(15)
	1	1	0	1	$\bar{A}$	0	A	1

Logic diagram :

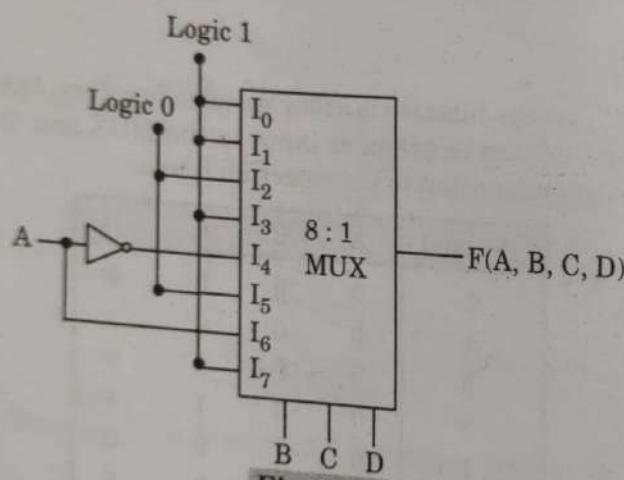


Fig. 2.8.1.

Que 2.9.

Implement the following Boolean function.

- 4 : 1 MUX
  - 2 : 1 MUX
- $$F(A, B, C, D) = \Sigma(0, 1, 3, 4, 7, 8, 9, 11, 14, 15)$$

AKTU 2018-19, Marks 07

**Answer****i. 4 : 1 MUX :****Implementation table :**

	$\bar{A}\bar{B}$	$\bar{A}B$	$A\bar{B}$	$AB$
	$I_0$	$I_1$	$I_2$	$I_3$
(00) $\bar{C}\bar{D}$	0	4	8	12
(01) $\bar{C}D$	1	5	9	13
(10) $C\bar{D}$	2	6	10	14
(11) $CD$	3	7	11	15

$$\text{First column } (I_0) = \bar{C}\bar{D} + \bar{C}D + CD$$

$$= \bar{C}(\bar{D} + D) + CD = \bar{C} + CD$$

$$= \bar{C} + D$$

$$\text{Second column } (I_1)$$

$$= \bar{C}\bar{D} + CD = C \odot D \quad [\odot \rightarrow \text{EX-NOR}]$$

$$\text{Third column } (I_2)$$

$$= \bar{C}\bar{D} + \bar{C}D + CD$$

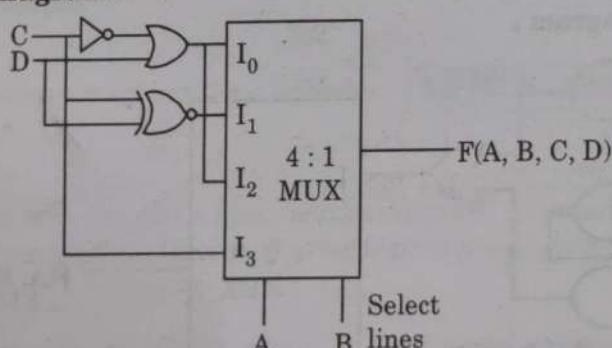
$$= \bar{C}(\bar{D} + D) + CD$$

$$= \bar{C} + CD = \bar{C} + D$$

$$\text{Fourth column } (I_3)$$

$$= C\bar{D} + CD$$

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

**Logic diagram :****Fig. 2.9.1.****ii. Implementation using 2 : 1 MUX :**

We have to use three variables as input of MUX and one variable as select line

**Implementation table :**

	$\bar{D}$ $I_0$	D $I_1$
$\bar{A}\bar{B}\bar{C}$	①	②
$\bar{A}\bar{B}C$	2	③
$\bar{A}\bar{B}\bar{C}$	④	5
$\bar{A}BC$	6	⑦
$\bar{A}\bar{B}\bar{C}$	⑧	⑨
$\bar{A}\bar{B}C$	10	⑪
$A\bar{B}\bar{C}$	12	13
$ABC$	⑭	⑮

First column ( $I_0$ )

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

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

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

Second column ( $I_1$ )

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

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

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

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

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

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

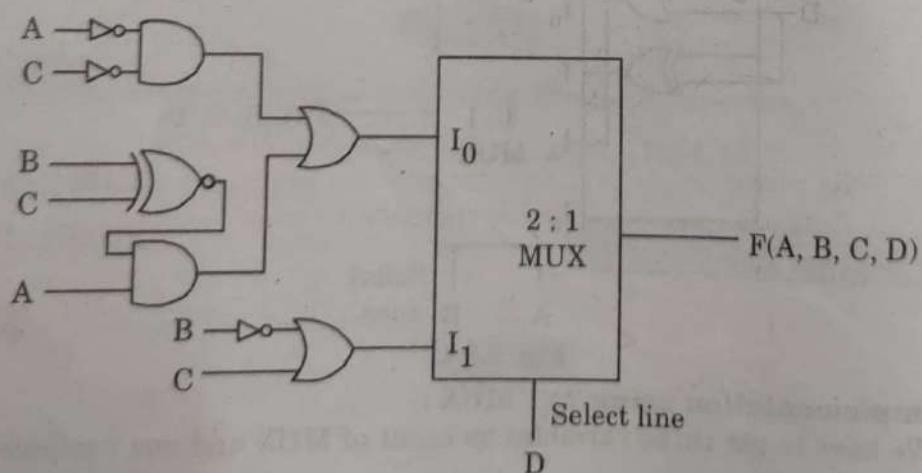
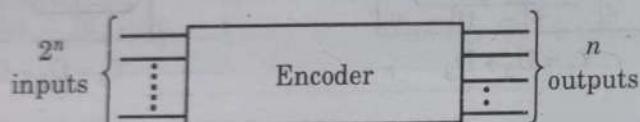
**Logic diagram :**

Fig. 2.9.2.

**PART-3***Encoder.***Questions-Answers****Long Answer Type and Medium Answer Type Questions****Que 2.10.** What do you mean by encoder ?**Answer****Encoder :**

1. The encoder is another example of combinational circuit that performs the inverse operation of a decoder. It is designed to generate a different output code for an input which becomes active.
2. In an encoder, the number of outputs is less than the number of inputs. There are  $2^n$  input lines and  $n$  output lines.
3. The block diagram of an encoder is shown in Fig. 2.10.1.

**Fig. 2.10.1.****Que 2.11.** What is priority encoder ? Explain with the help of suitable example.**OR**Write a short note on priority encoder. **AKTU 2018-19, Marks 07****Answer**

**Priority encoder :** In priority encoder if two or more inputs are equal to 1 at the same time, the input having highest priority will be considered.

**Example :**

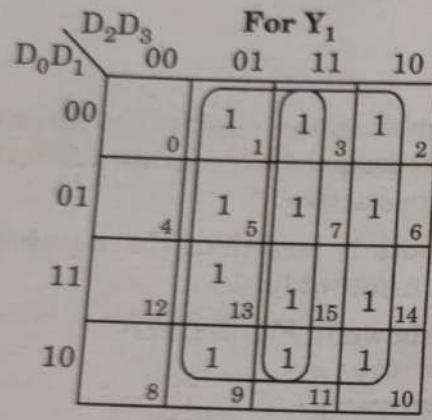
Four inputs  $D_0, D_1, D_2, D_3$  where  $D_3$  has highest priority and  $D_0$  has lowest priority.

$Y_0, Y_1$  : binary output

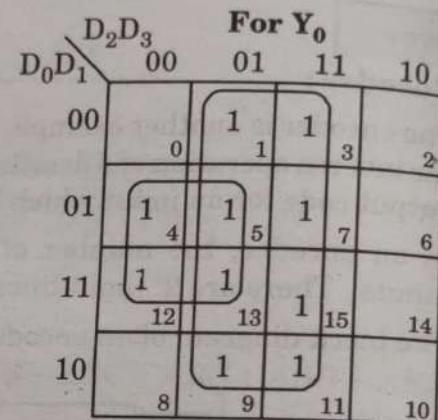
$V$  : validity of output

**Table 2.11.1** Truth table for 4-bit priority encoder.

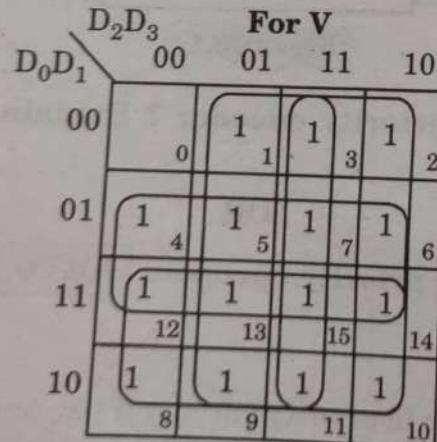
Input				Output			
$D_0$	$D_1$	$D_2$	$D_3$	$Y_1$	$Y_0$	$V$	
0	0	0	0	*	*	0	
1	0	0	0	0	0	1	
*	1	0	0	0	1	1	
*	*	1	0	1	0	1	
*	*	*	1	1	1	1	



$$Y_1 = D_2 + D_3$$



$$Y_0 = D_3 + D_1 \bar{D}_2$$



$$V = D_0 + D_1 + D_2 + D_3$$

**Fig. 2.11.1.**

**Logic diagram :**

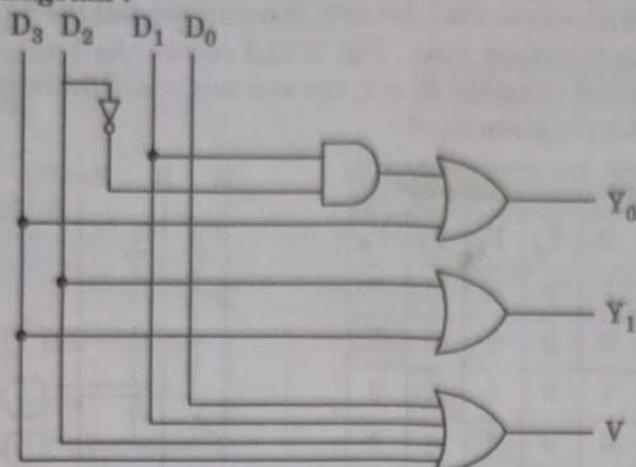


Fig. 2.11.2.

#### PART-4

*Decoder.*

#### Questions-Answers

##### Long Answer Type and Medium Answer Type Questions

**Que 2.12.** Write a short note on decoder.

#### Answer

1. A decoder is a combinational circuit that converts binary information from  $n$  input lines to a maximum of  $2^n$  unique output lines.
2. If the  $n$ -bit coded information has unused combinations, the decoder may have fewer than  $2^n$  outputs.
3. The decoders presented here are called  $n$  to  $m$  line decoders, where  $m \leq 2^n$ . Their purpose is to generate the  $2^n$  (or fewer) minterms of  $n$  input variables.

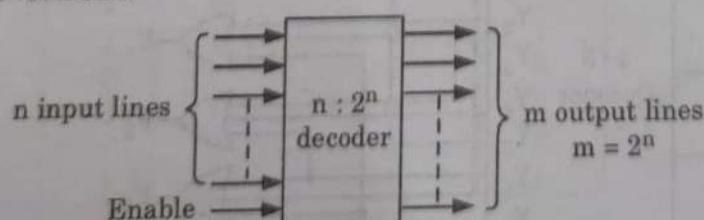


Fig. 2.12.1. Block diagram of a decoder.

**2 to 4 binary decoder :**

Fig. 2.12.2 shows the 2 to 4 decoder. Here 2 represent the input lines and 4 represents output lines. Fig. 2.12.2 shows the truth table for a 2 to 4 decoder. If enable (E) is 1, one and only one of the outputs  $Y_0$  to  $Y_3$  is active for the given input.

Inputs			Outputs			
E	A	B	$Y_3$	$Y_2$	$Y_1$	$Y_0$
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

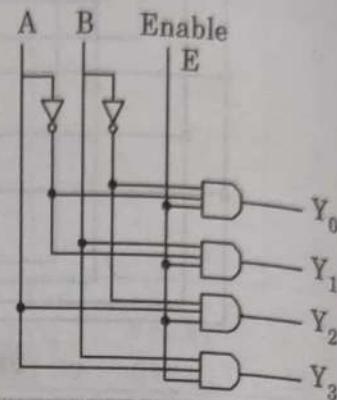


Fig. 2.12.2. Logic diagram of 2 to 4 decoder.

**Que 2.13.** Using a decoder and external gates, design the combinational circuit defined by the following three boolean functions :

$$F_1 = x'yz' + xz, F_2 = xy'z' + yz', F_3 = x'y'z' + xy$$

AKTU 2015-16, Marks 10

**Answer**

Let us consider 3 to 8 line decoder. The implementation of the given three functions using 3 to 8 line decoder and a few OR gates are shown as follows :

$$\begin{aligned} F_1 &= x'yz' + xz = x'yz' + xz(y + y') \\ &= x'yz' + xyz + xy'z = \Sigma m(2, 7, 5) \\ F_2 &= xy'z' + yz' = xy'z' + (x + x')yz' \\ &= xy'z' + xyz' + x'yz' = \Sigma m(2, 4, 6) \\ F_3 &= x'y'z' + xy = x'y'z' + xy(z + z') \\ &= x'y'z' + xyz + xyz' = \Sigma m(0, 6, 7) \end{aligned}$$

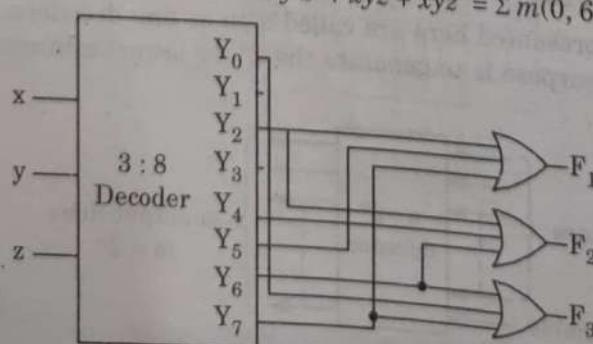


Fig. 2.13.1. Implementation of the given Boolean function using 3:8 decoder.

**Que 2.14.** Draw the logic diagram of a two to four line decoder using NOR gates only.

AKTU 2016-17, Marks 10

**Answer**

Truth table :

Enable	Input		Output			
	E	A	B	$Y_3$	$Y_2$	$Y_1$
0	x	x		0	0	0
1	0	0		0	0	1
1	0	1		0	0	0
1	1	0		0	1	0
1	1	1		1	0	0

Circuit using NOR gate :

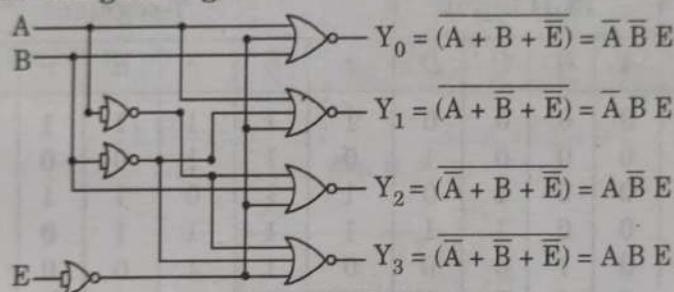


Fig. 2.14.1.

**Que 2.15.** Design a full subtractor circuit with a decoder and two OR gates.

**Answer**

Full subtractor using decoder :

Input			Output	
A	B	C	D	$B_0$
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

Difference :  $D = \Sigma m(1, 2, 4, 7)$

Borrow :  $B_0 = \Sigma m(1, 2, 3, 7)$

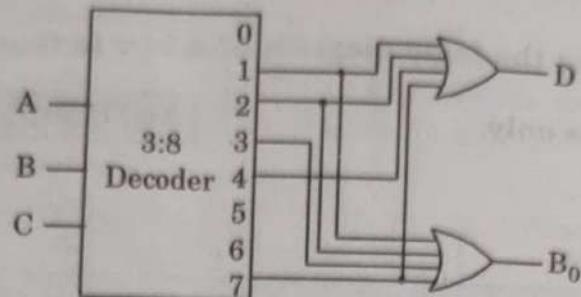


Fig. 2.15.1. Full subtractor using 3:8 decoder.

**Que 2.16.** Design a BCD to 7 segment decoder. Assume positive logic, minimize the function.

AKTU 2014-15, Marks 06

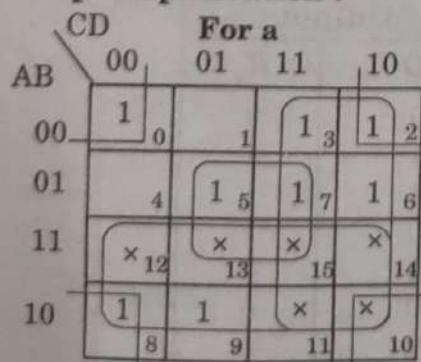
**Answer**

1. The truth table for a BCD to 7 segment decoder is as given as follows:

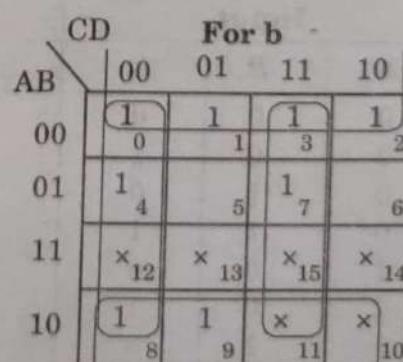
Digit	BCD input				7-segment						
	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

2. The unused BCD codes are 1010, 1011, 1100, 1101, 1110, and 1111. So place  $\times$  (don't care condition) for these corresponding cells.

**K-map simplification :**



$$a = A + C + BD + \bar{B}\bar{D}$$



$$b = \bar{B} + \bar{C}\bar{D} + CD$$

		For e					
		CD	00	01	11	10	
AB		00	1 0	(1 1)	1 3		
		01	1 4	1 5	1 7	1 6	
AB		11	x	x	x	x	
		10	12	13	15	14	
		00	1 1	x	x		
		01	8	9	11	10	

$$c = B + \bar{C} + D$$

		For e					
		CD	00	01	11	10	
AB		00	1 0		3	(1 2)	
		01	4	5	7	1 6	
AB		11	x	x	x	x	
		10	12	13	15	14	
		00	1 1	x	x		
		01	8	9	11	10	

$$e = \bar{B} \bar{D} + C \bar{D}$$

		For d					
		CD	00	01	11	10	
AB		00	1	0	1	2	
		01	4	(1 5)	7	1 6	
AB		11	x	x	x	x	
		10	12	13	15	14	
		00	1	1	x	x	
		01	8	9	11	10	

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

		For f					
		CD	00	01	11	10	
AB		00	1 0	1	3	2	
		01	1 4	(1 5)	7	1 6	
AB		11	x	x	x	x	
		10	12	13	15	14	
		00	1	1	x	x	
		01	8	9	11	10	

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

		For g					
		CD	00	01	11	10	
AB		00	0	1	1	2	
		01	1 4	(1 5)	7	1 6	
AB		11	x	x	x	x	
		10	12	13	15	14	
		00	1	1	x	x	
		01	8	9	11	10	

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

Fig. 2.16.1.

3. Fig. 2.16.2 shows the logic diagram of BCD to 7 segment display decoder.

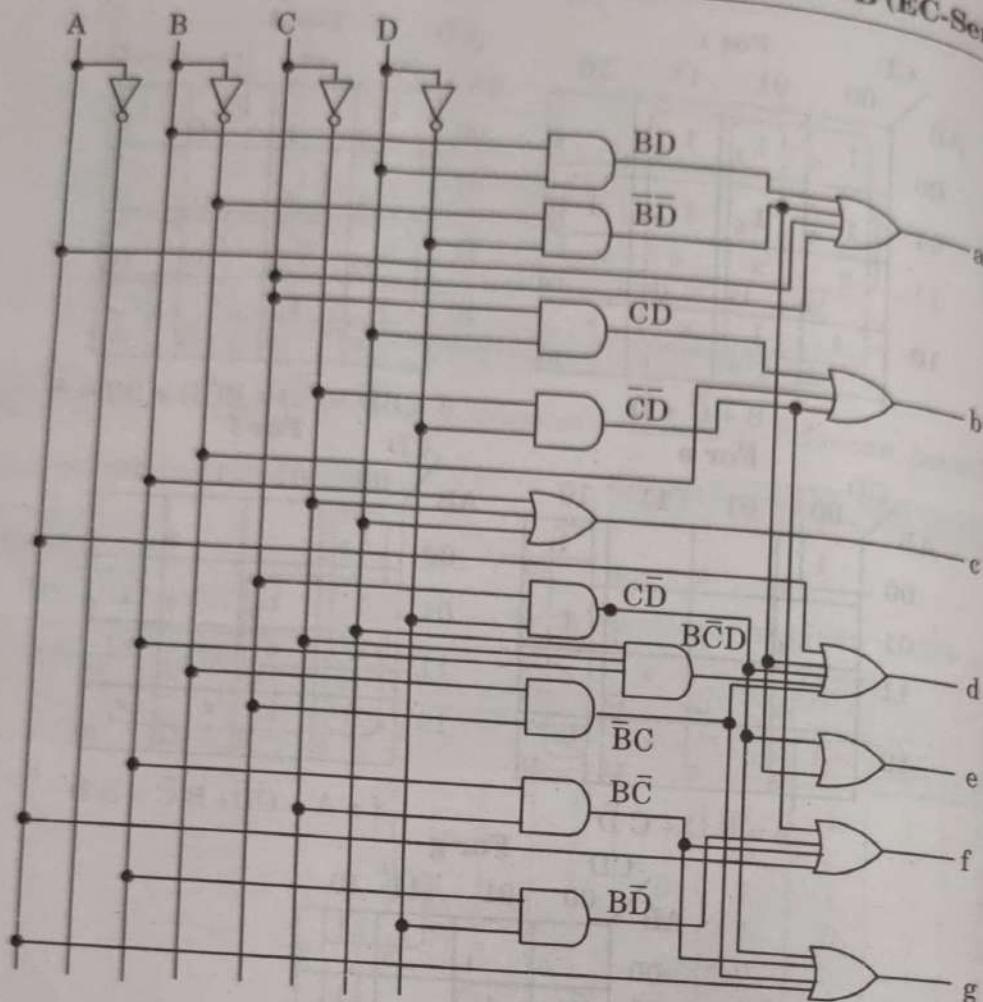


Fig. 2.16.2.

**PART-5***Driver and Multiplexed Display.***Questions-Answers****Long Answer Type and Medium Answer Type Questions**

**Que 2.17.** Explain multiplexed display.

**Answer**

1. In a multiplexed display, each seven-segment display is lit in turn with its appropriate value. This is done by outputting the appropriate lines with one part of the driver chip, while another part of the logic enables

the common cathode connection of the selected seven-segment display, (Fig. 2.17.1).

2. Then a small time later, the common cathode of the next display is activated and lit with its required number.
3. Thus if you could look at the display in slow motion, you would see each of the seven segment displays light then go out sequentially.
4. The trick is that if the displays are strobed sufficiently rapidly (say a few hundred times a second) the eye is deceived into thinking they are on all the time, just like a television set.
5. The advantages of multiplexed displays are as follows: only one set of current-limiting resistors is required, fewer wiring connections need to be made and power consumption is less since only one seven-segment display is lit at any given time.
6. Also, since multiplexed display chips are available from manufacturers, you do not have to design and build one yourself.

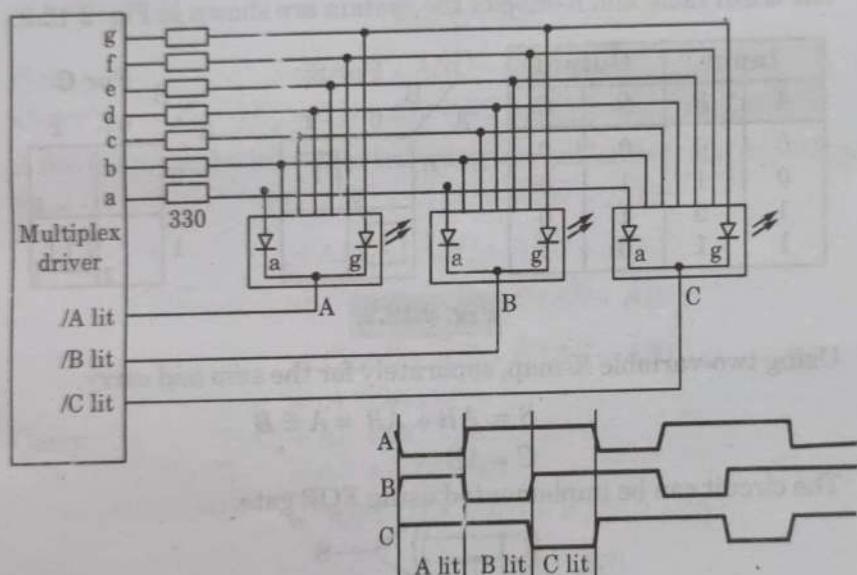


Fig. 2.17.1.

**PART-6***Half and Full Adders.***Questions-Answers****Long Answer Type and Medium Answer Type Questions**

**Que 2.18.** Describe half adder and full adder in brief. Implement the circuit using logic gates.

OR

Design a full adder using two half adders.

AKTU 2015-16, Marks 7.5

**Answer**

**Half adder :**

1. The block diagram of half adder is shown in Fig. 2.18.1.

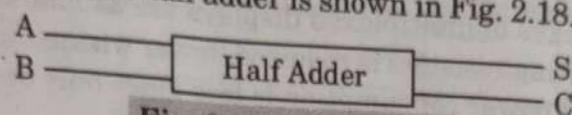


Fig. 2.18.1. Half adder.

where,  $A$  and  $B$  are the inputs and  $S$  and  $C$  are the outputs sum and carry respectively.

2. The truth table and K-map of the system are shown in Fig. 2.18.2.

Input		Output	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

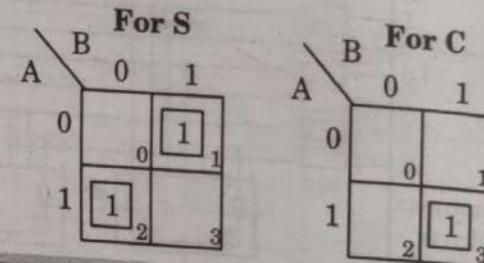


Fig. 2.18.2.

3. Using two-variable K-map, separately for the sum and carry.

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

$$C = AB$$

4. The circuit can be implemented using XOR gate.

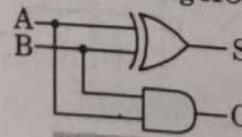


Fig. 2.18.3.

**Full adder :**

1. Full adder is a circuit that performs the addition of three binary digits. It has three inputs  $A$ ,  $B$  and  $C$  with two output  $S$  and  $C_o$ , where  $C_o$  is the previous carry. The block diagram is shown in Fig. 2.18.4.

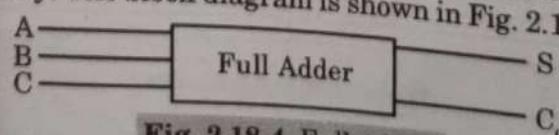


Fig. 2.18.4. Full Adder.

2. If there are three input variables the combinations are eight ( $2^3 = 8$ ). Now form the truth table of the full adder.

Inputs			Outputs	
A	B	C	S	$C_o$
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

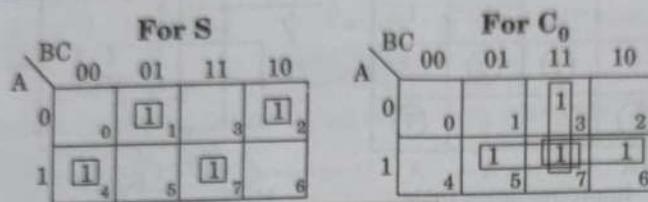


Fig. 2.18.5.

3. Sum :  $S = ABC + A\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}\bar{B}C$

Carry :  $C_o = AB + AC + BC$

4. A full adder can be implemented using two half adders and one OR gate.

Sum :  $S = ABC + A\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}\bar{B}C$

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

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

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

$$= (A \oplus B) \oplus C$$

Carry :  $C_o = AB + AC + BC$

$$= AB + C(A + B)$$

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

$$= AB + C[AB + A\bar{B} + \bar{A}B]$$

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

$$= AB(1 + C) + C(A \oplus B)$$

$$= AB + C(A \oplus B)$$

Half adder

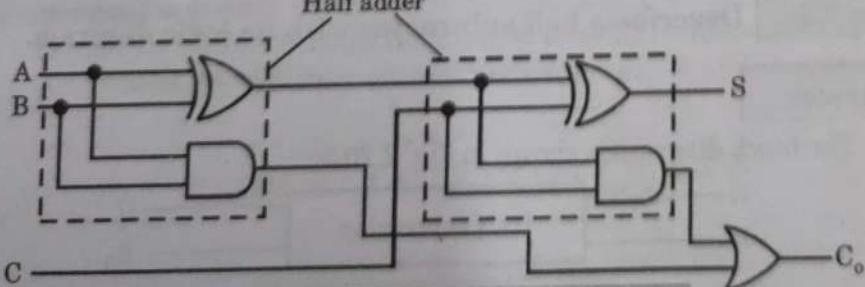


Fig. 2.18.6. Full adder circuit using 2 half adder.

**Que 2.19.** Implement a full adder circuit using  $4 \times 1$  multiplexer.

**Answer**

1. Canonical form of sum and carry for full adder :

$$\text{Sum} = \Sigma m(1, 2, 4, 7) = \bar{B}\bar{C}A + \bar{B}C\bar{A} + B\bar{C}\bar{A} + BCA$$

$$\begin{aligned}\text{Carry} &= \Sigma m(3, 5, 6, 7) = BCA + \bar{B}CA + B\bar{C}A + BCA \\ &= \bar{B}CA + B\bar{C}A + BC\end{aligned}$$

For sum :

	I <sub>0</sub>	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>
$\bar{A}$	0	(1)	(2)	3
A	(4)	5	6	(7)
	A	$\bar{A}$	$\bar{A}$	A

Implementation table

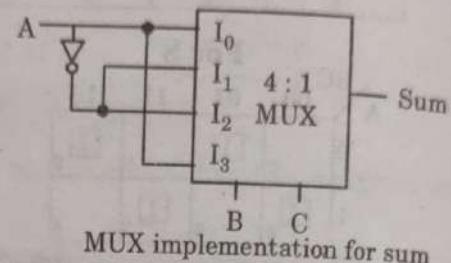


Fig. 2.19.1.

For carry :

	I <sub>0</sub>	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>
$\bar{A}$	0	1	2	(3)
A	4	(5)	(6)	(7)
	0	A	A	1

Implementation table

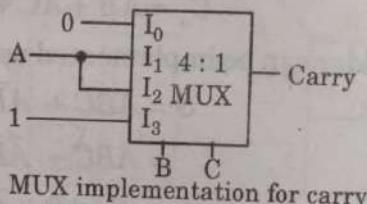


Fig. 2.19.2.

## PART-7

Subtractors.

### Questions-Answers

#### Long Answer Type and Medium Answer Type Questions

**Que 2.20.** Describe a half subtractor with its logic diagram.

**Answer**

1. The block diagram is shown in Fig. 2.20.1.

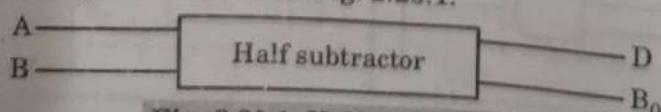


Fig. 2.20.1. Half subtractor.

2. It has two inputs,  $A$  (minuend) and  $B$  (subtrahend) and two outputs  $D$  (difference) and  $B_o$  (borrow) are produced by subtraction of two bits.
3. The truth table can be formed by keeping in mind that difference (output) is 0 if  $A = B$  and 1 if  $A \neq B$ . The K-map and truth table are shown in Fig. 2.20.2.

Inputs		Outputs	
$A$	$B$	$D$	$B_o$
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

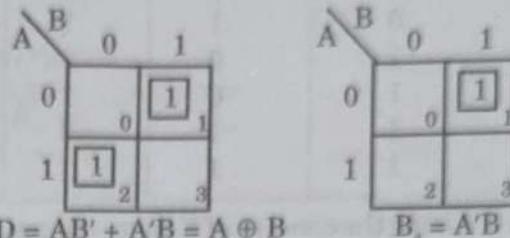


Fig. 2.20.2.

4. The logical implementation using basic logic gates and XOR gate :

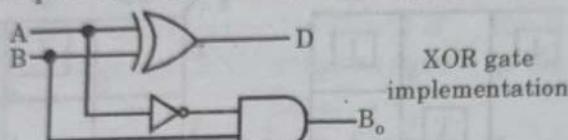


Fig. 2.20.3.

**Que 2.21.** Design a full subtractor circuit with three inputs  $x, y$ ,  $B_{in}$  and two outputs Diff and  $B_{out}$ . The circuit subtracts  $x - y - B_{in}$ , where,  $B_{in}$  is the input borrow,  $B_{out}$  is the output borrow and Diff is the difference.

AKTU 2016-17, Marks 10

**Answer**

1. It is a combinational circuit that performs the subtraction of three binary digits.

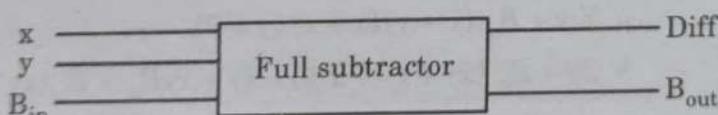
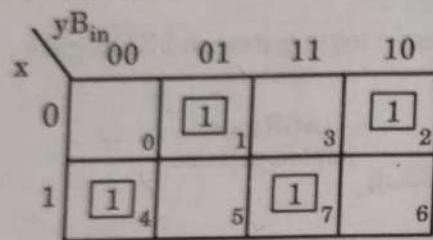


Fig. 2.21.1.

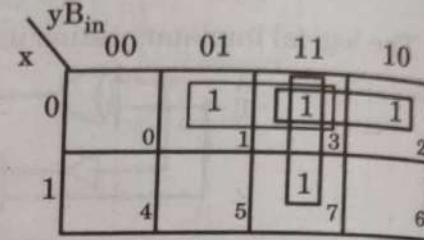
2. Fig. 2.21.1 shows the block diagram approach of full subtractor. It has three inputs  $x, y$  and  $B_{in}$  and two outputs 'Diff' and  $B_{out}$  produced by subtraction of three input bits.
3. For the formation of truth table, eight possible combinations of three input variables with their outputs are required.

Inputs			Outputs	
$x$	$y$	$B_{in}$	Diff	$B_{out}$
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

4. Using the concept of K-map, reduce the truth table to a function (algebraic or boolean).



$$\text{Diff} = \bar{x} \bar{y} B_{in} + \bar{x} y \bar{B}_{in} + xy B_{in} + x \bar{y} \bar{B}_{in}$$



$$B_{out} = \bar{x} B_{in} + y B_{in} + \bar{x} \bar{y}$$

Fig. 2.21.2.

5. A full subtractor can also be implemented using two half subtractors and an OR gate.

$$\begin{aligned} \text{Diff} &= xy B_{in} + x \bar{y} \bar{B}_{in} + \bar{x} y \bar{B}_{in} + \bar{x} \bar{y} B_{in} \\ &= B_{in} (xy + \bar{x} \bar{y}) + \bar{B}_{in} (x \bar{y} + \bar{x} y) \\ &= B_{in} (\overline{x \oplus y}) + \bar{B}_{in} (x \oplus y) = (x \oplus y) \oplus B_{in} \end{aligned}$$

$$\begin{aligned} \text{and } B_{out} &= \bar{x} y + \bar{x} B_{in} + y B_{in} = \bar{x} y + B_{in} (\bar{x} + y) \\ &= \bar{x} y + B_{in} (\bar{x} + y)(x + \bar{x})(y + \bar{y}) \\ &= \bar{x} y + B_{in} (\bar{x}y + xy + \bar{x} \bar{y}) = \bar{x} y + \bar{x} y B_{in} + B_{in} (xy + \bar{x} \bar{y}) \\ &= \bar{x} y (B_{in} + 1) + B_{in} (\overline{x \oplus y}) = \bar{x} y + B_{in} (\overline{x \oplus y}) \end{aligned}$$

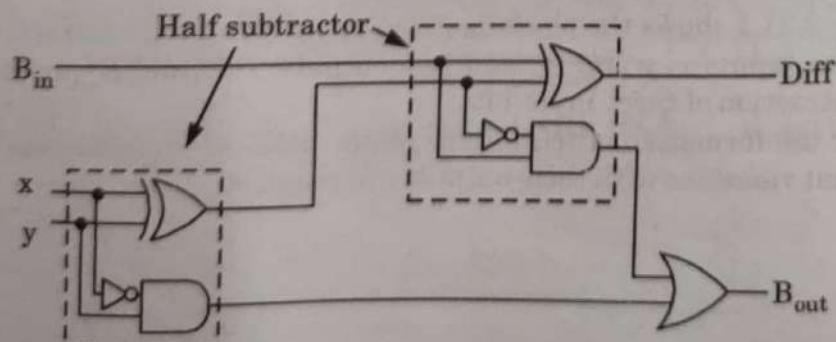


Fig. 2.21.3. Full subtractor circuit using 2 half subtractor.

**Que 2.22.** Draw a full subtractor circuit using NAND gate.

**AKTU 2018-19, Marks 3.5**

**Answer**

Full subtractor using only NAND gates

$$= A \oplus B \oplus B_{in} = \overline{(A \oplus B)(A \oplus B)B_{in}} \quad B_{in}(\overline{A \oplus B})\overline{B_{in}}$$

$$B_{out} = \bar{A}B + B_{in}\overline{(A \oplus B)} = \overline{\bar{A}B + B_{in}(A \oplus B)}$$

$$= \overline{\bar{A}B} \cdot \overline{B_{in}(A \oplus B)} = \overline{B(\bar{A} + \bar{B})} \quad B_{in}[\overline{B_{in}} + \overline{(A \oplus B)}]$$

$$B_{out} = \overline{B \overline{AB} B_{in}[B_{in}(A \oplus B)]}$$

By using the above expressions for  $D$  and  $B_{out}$ , the full subtractor is implemented using only NAND gates as shown in Fig. 2.22.1.

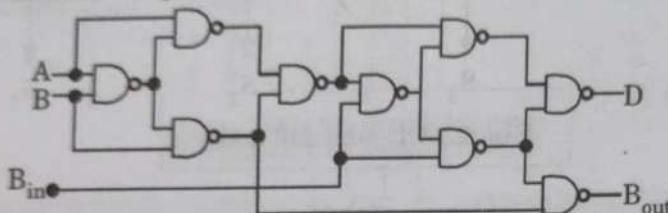


Fig. 2.22.1. Full subtractor using only NAND gate.

**PART-8**

*Serial and Parallel Adders.*

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

**Que 2.23.** Design a 4-bit binary parallel adder.

**Answer**

1. The sum of two  $n$ -bit binary numbers,  $A$  and  $B$ , can be generated in two ways: either in a serial fashion or in parallel.
2. The series addition method uses only one full-adder circuit and a storage device to hold the generated output carry.
3. The pair of bits in  $A$  and  $B$  are transferred serially, one at a time, through the single full-adder to produce a string of output bits for the sum.

4. The stored output carry from one pair of bits is used as an input carry for the next pair of bits.
5. The parallel method uses  $n$  full-adder circuits, and all bits of  $A$  and  $B$  are applied simultaneously.
6. A binary parallel adder is a digital function that produces the arithmetic sum of two binary numbers in parallel.
7. It consists of full-adders connected in cascade, with the output carry from one full-adder connected to the input carry of the next full-adder.

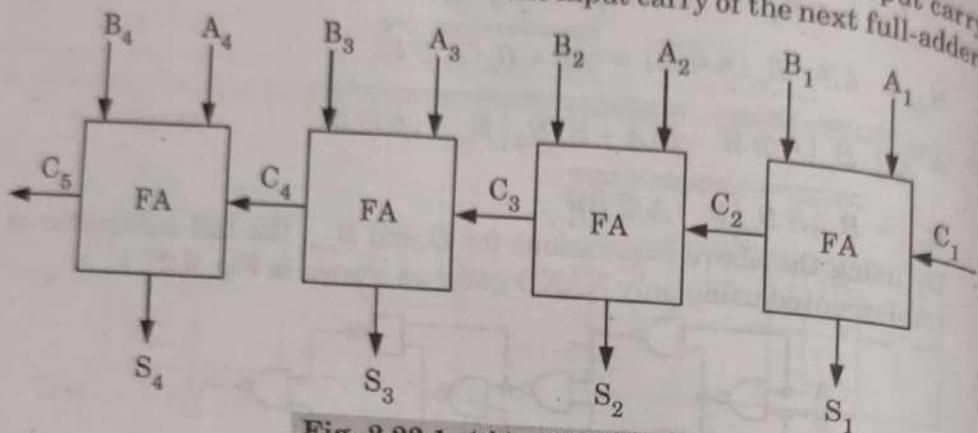


Fig. 2.23.1. 4-bit full adders.

**PART-9***BCD Adder.***Questions-Answers****Long Answer Type and Medium Answer Type Questions**

**Que 2.24.** Draw a BCD adder circuit and explain its working.

**AKTU 2018-19, Marks 07****OR**

Draw a decimal adder to add BCD number.

**AKTU 2017-18, Marks 07****Answer**

1. BCD adder is circuit that adds two BCD digits in parallel and produces a sum digit which is also BCD. BCD numbers use 10 symbols (group of 4 bits 0000 to 1001). BCD adder circuit must be able to do the following and it is shown in Fig. 2.24.1.
2. Add two 4-bit BCD numbers using straight binary addition.

3. If 4-bit sum is equal to or less than 9, the sum is a valid BCD number and no correction is needed.
4. If the 4-bit sum is greater than 9 or if a carry is generated from the sum, the sum is invalid BCD number. Then the digit 6 ( $0110_2$ ) should be added to the sum to produce the valid BCD symbols.

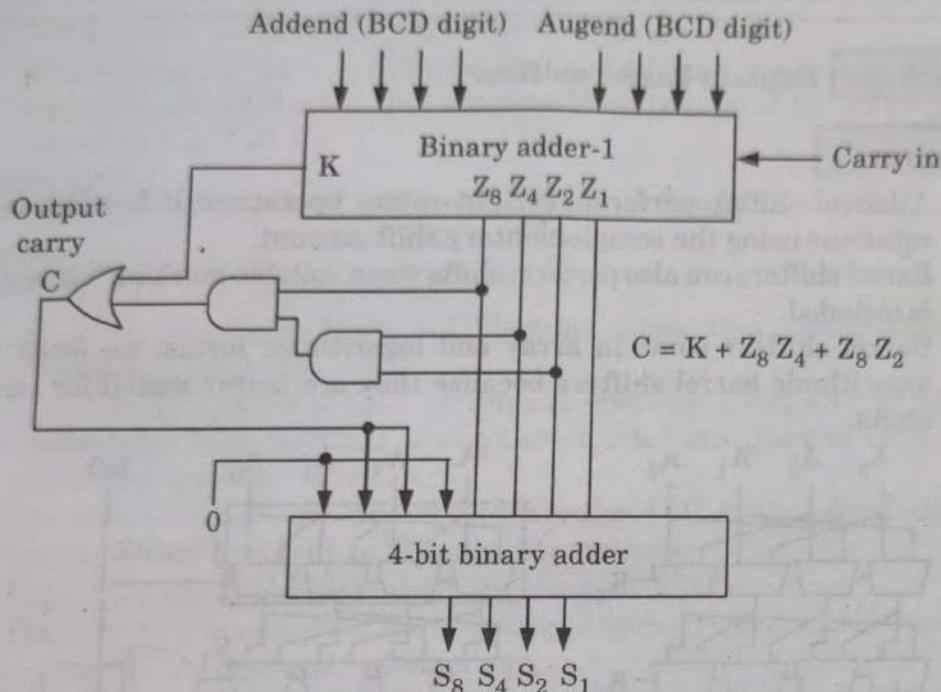


Fig. 2.24.1. Block diagram of a BCD adder.

Binary Sum				BCD Sum					Decimal
Z <sub>8</sub>	Z <sub>4</sub>	Z <sub>2</sub>	Z <sub>1</sub>	C	S <sub>8</sub>	S <sub>4</sub>	S <sub>2</sub>	S <sub>1</sub>	
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1	1
0	0	1	0	0	0	0	1	0	2
0	0	1	1	0	0	0	1	1	3
0	1	0	0	0	0	1	0	0	4
0	1	0	1	0	0	1	0	1	5
0	1	1	0	0	0	1	1	0	6
0	1	1	1	0	0	1	1	1	7
1	0	0	0	0	1	0	0	0	8
1	0	0	1	0	1	0	0	1	9
1	0	1	0	1	0	0	0	0	10
1	0	1	1	1	0	0	0	1	11

PART-1 □

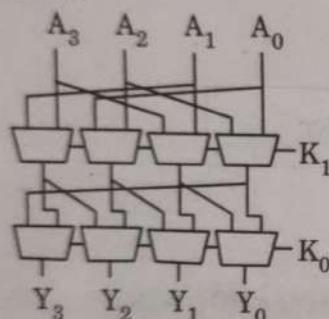
Barrel Shifter and ALU.

**Questions-Answers****Long Answer Type and Medium Answer Type Questions**

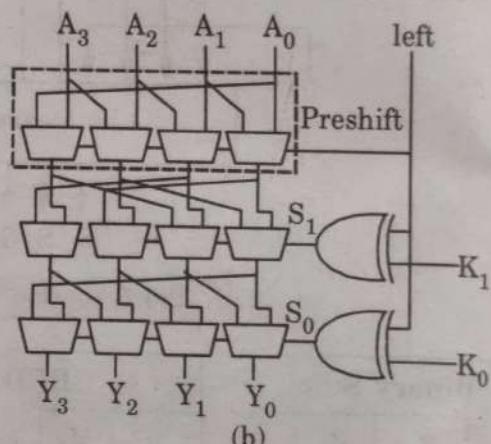
**Que 2.25.** Explain barrel shifter.

**Answer**

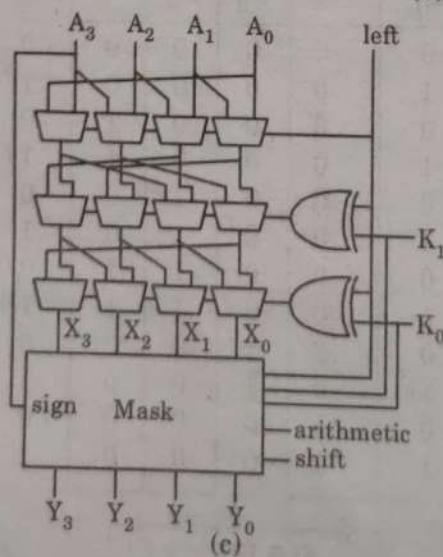
1. A barrel shifter performs a right rotate operation. It handles left rotations using the complementary shift amount.
2. Barrel shifters can also perform shifts when suitable masking hardware is included.
3. Barrel shifters come in array and logarithmic forms; we focus on logarithmic barrel shifters because they are better suited for large shifts.



(a)



(b)

**Fig. 2.25.1.**

4. Fig. 2.25.1(a) shows a simple-4 bits barrel shifter that performs right rotations. However, unlike funnel shifters, barrel shifters contain long wrap-around wires.
5. In a large shifter, it is beneficial to upsize or buffer the drivers for these wires. Fig. 2.25.1(b), shows an enhanced version that can rotate left by prerotating right by 1, then rotating right by  $\bar{k}$ .
6. Performing logical or arithmetic shifts on a barrel shifter requires a way to mask out the bits that are rotated off the end of the shifter, as shown in Fig. 2.25.1(c).

**Que 2.26.** Write a short note on Arithmetic Logic Unit (ALU).

**Answer**

1. An Arithmetic Logic Unit (ALU) is a multioperation, combinational logic digital function.
2. It can perform a set of basic arithmetic operations and a set of logic operations. The ALU has a number of selection lines to select a particular operation in the unit.
3. The selection lines are decoded within the ALU so that  $k$  selection variables can specify up to  $2^k$  distinct operations.
4. Fig. 2.26.1 shows the block diagram of a 4-bit ALU.
5. The four data inputs from A are combined with the four inputs from B to generate an operation at the F output.
6. The mode-select input  $s_2$  distinguishes between arithmetic and logic operations.
7. The two function-select inputs  $s_1$  and  $s_0$  specify the particular arithmetic or logic operation to be generated. With three selection variables, it is possible to specify four arithmetic operations (with  $s_2$  in one state) and four logic operations (with  $s_2$  in the other state).
8. The input and output carries have meaning only during an arithmetic operation.
9. The input carry in the least significant position of an ALU is quite often used as fourth selection variable that can double the number of arithmetic operations. In this way, it is possible to generate four more operations, for a total of eight arithmetic operations.

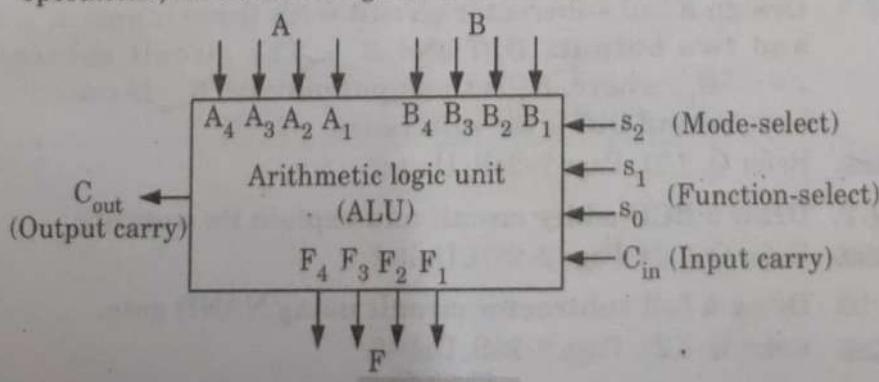
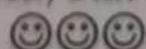


Fig. 2.26.1.

**VERY IMPORTANT QUESTIONS**

*Following questions are very important. These questions may be asked in your SESSIONALS as well as UNIVERSITY EXAMINATION.*

- Q. 1.** Draw and explain 4-bit magnitude comparator.  
**Ans.** Refer Q. 2.2, Page 2-3B, Unit-2.
- Q. 2.** Design the following boolean function using 4-to-1 multiplexer.  
 $F(A, B, C, D) = \Sigma m(0, 1, 3, 4, 8, 9, 15)$   
**Ans.** Refer Q. 2.6, Page 2-6B, Unit-2.
- Q. 3.** Implement the function  $F = \Sigma m(0, 1, 3, 4, 7, 8, 9, 11, 14, 15)$  using 8:1 mux.  
**Ans.** Refer Q. 2.8, Page 2-8B, Unit-2.
- Q. 4.** Using a decoder and external gates, design the combinational circuit defined by the following three boolean functions :  
 $F_1 = x'yz' + xz, F_2 = xy'z' + yz', F_3 = x'y'z' + xy$   
**Ans.** Refer Q. 2.13, Page 2-15B, Unit-2.
- Q. 5.** Draw the logic diagram of a two to four line decoder using NOR gates only.  
**Ans.** Refer Q. 2.14, Page 2-16B, Unit-2.
- Q. 6.** Implement the following Boolean function.  
 $F(A, B, C, D) = \Sigma(0, 1, 3, 4, 7, 8, 9, 11, 14, 15)$   
 i. 4:1 MUX  
 ii. 2:1 MUX  
**Ans.** Refer Q. 2.9, Page 2-9B, Unit-2.
- Q. 7.** Design a full adder using two half adders.  
**Ans.** Refer Q. 2.18, Page 2-21B, Unit-2.
- Q. 8.** Design a full subtractor circuit with three inputs  $x, y, B_{in}$  and two outputs Diff and  $B_{out}$ . The circuit subtracts  $x - y - B_{in}$ , where,  $B_{in}$  is the input borrow,  $B_{out}$  is the output borrow and Diff is the difference.  
**Ans.** Refer Q. 2.21, Page 2-24B, Unit-2.
- Q. 9.** Draw a BCD adder circuit and explain its working.  
**Ans.** Refer Q. 2.24, Page 2-27B, Unit-2.
- Q. 10.** Draw a full subtractor circuit using NAND gate.  
**Ans.** Refer Q. 2.22, Page 2-26B, Unit-2.





## Sequential Logic Design

### CONTENTS

- |                |   |                       |
|----------------|---|-----------------------|
| <b>Part-1</b>  | : Building Block Like S-R Flip Flop ..... | <b>3-2B to 3-3B</b>   |
| <b>Part-2</b>  | : JK and Master-Slave JK FF .....         | <b>3-3B to 3-5B</b>   |
| <b>Part-3</b>  | : Edge Triggered FF .....                 | <b>3-5B to 3-6B</b>   |
| <b>Part-4</b>  | : State Diagram, State Reduction .....    | <b>3-6B to 3-16B</b>  |
|                | Design of Sequential Circuits             |                       |
| <b>Part-5</b>  | : Ripple and Synchronous Counters .....   | <b>3-16B to 3-24B</b> |
| <b>Part-6</b>  | : Shift Registers .....                   | <b>3-24B to 3-31B</b> |
| <b>Part-7</b>  | : Finite State Machines (FSM), .....      | <b>3-31B to 3-33B</b> |
|                | Design of Synchronous FSM                 |                       |
| <b>Part-8</b>  | : Algorithmic State Machines .....        | <b>3-33B to 3-35B</b> |
|                | (ASM) Charts                              |                       |
| <b>Part-9</b>  | : Designing Synchronous Circuits .....    | <b>3-36B to 3-38B</b> |
|                | Like Pulse Train Generator                |                       |
| <b>Part-10</b> | : Pseudo Random Binary Sequence .....     | <b>3-38B to 3-39B</b> |
|                | Generator, Clock Generation               |                       |

**PART-1***Building Block Like S-R Flip Flop.***Questions-Answers****Long Answer Type and Medium Answer Type Questions**

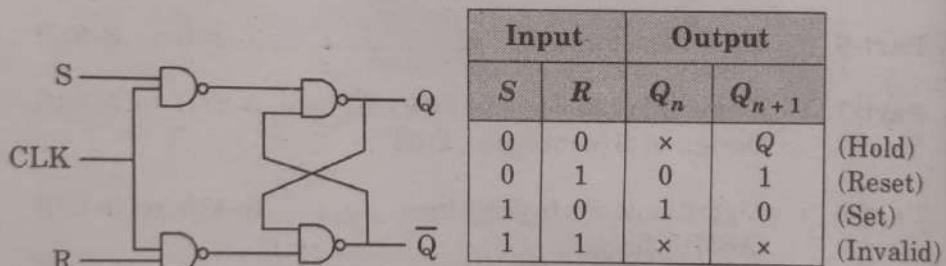
**Que 3.1.** What do you mean by flip flop ? Explain SR flip-flop.

**Answer****Flip-flops :**

Flip-flops are binary cells capable of storing one bit of information. A flip-flop circuit has two outputs, one for the normal value and one for the complement value of the bit stored in it.

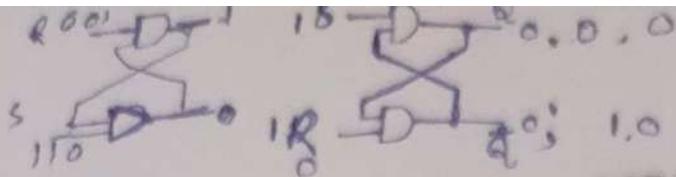
**SR flip-flop :**

1. The circuit diagram and truth table of SR flip-flop are shown in fig. 3.1.1. This is also known as clocked set-reset flip-flop.
2. The circuit functions when clock pulse is active, i.e., 1 otherwise it will hold its output values ( $Q$  and  $\bar{Q}$ ).

**Fig. 3.1.1. SR Flip-flop.**

3. It can be observed from the truth-table that if  $S = R = 0$  and CLK is active then the output is same as previous.
4. If  $S = 0$  and  $R = 1$ , the flip-flop will be in reset stage, i.e., output  $Q$  will be 0.
5. If  $S = 1$  and  $R = 0$ , the flip-flop will be in set stage, i.e., output  $Q$  will be 1.
6. If  $S = 1$  and  $R = 1$  then output is invalid i.e.,  $Q$  and  $\bar{Q}$  both will attain logic 1 which contradicts the assumption of complementary outputs.

**Que 3.2.** Write the difference between latches and flip-flops.



**Answer**

S. No.	Latch	Flip-flop
1.	Storage element that operate with signal levels.	Storage elements that are controlled by clock transitions.
2.	It is level triggered.	It is edge triggered.
3.	There is no clock pulse.	There is a clock pulse.
4.	<b>Circuit diagram :</b>  Latch 	<b>Circuit diagram :</b>  Flip-Flop 

**PART-2**

*JK and Master-Slave JK FF.*

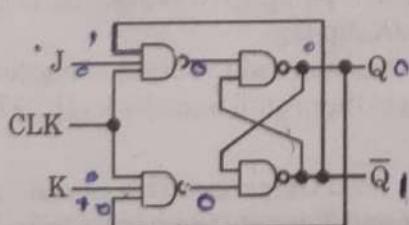
**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

**Que 3.3.** Discuss JK flip-flop with its circuit diagram.

**Answer**

1. The circuit diagram and truth table of JK flip-flop is shown in Fig. 3.3.1.



J	K	$Q_n$	$Q_{(n+1)}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

} No change  
i.e.,  $Q_n$   
} Reset  
} Set  
} Toggle

Fig. 3.3.1. JK flip-flop.

J K - 1 - 0

2. The previous problem that  $S = R = 1$  is invalid in  $SR$  flip-flop has been overcome by  $JK$  flip-flop.
3. The working of  $JK$  flip-flop is similar to  $SR$  flip-flop except that when  $J = K = 1$ , the output exists, i.e., when  $J = K = 1$ , the output is 1 when its previous output is 0 and 0 if its previous output is 1.
4. The condition  $J = K = 1$  causes a major problem, i.e., race-around condition. Consider  $J = K = 1$  and  $Q = 0$  and a pulse is applied at  $CLK$  input.
5. After a time interval  $\Delta t$  equal to propagation delay through two  $NAND$  gates in series. The output will oscillate between 0 and 1.
6. At the end of  $CLK$  the output is uncertain and the condition is race-around condition. There are two methods to avoid race-around condition by using :
  - Master Slave  $JK$  flip-flop.
  - Edge-triggered flip-flop.

**Que 3.4.** Write a short note on master-slave  $JK$  flip-flop.

**Answer**

1. Master-slave combination can be constructed for any type of flip-flop by adding a gated  $SR$  flip-flop. Fig. 3.4.1 explains the master-slave operation of  $JK$  flip-flop.

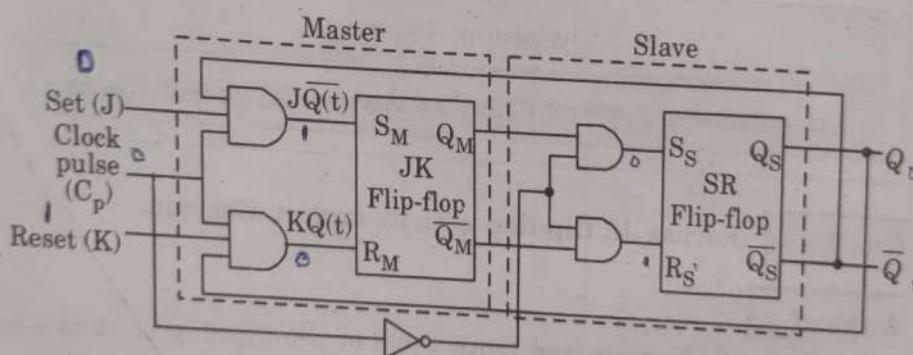


Fig. 3.4.1. Master-slave  $JK$  flip-flop.

2. It requires two flip-flops where one is gated  $JK$  acts as master and other  $SR$  flip-flop acts as slave. Output of slave flip-flop is fed as input to AND gate of master flip-flop which acts as  $JK$  flip-flop.
3. Information present at  $JK$  flip-flop is transmitted to output of master flip-flop on positive level clock and is held there until negative level clock pulse appears.
4. The truth table 3.4.1 follows negative level triggering. This cascading avoids race around condition when  $J$  and  $K$  inputs are having logic-1 information.

Table 3.4.1. Characteristics of master and slave JK flip-flop.

Input		CP	Outputs	Remarks
J (Set)	K (Reset)			
0	0		Q(t)	Previous state
0	1		0	Set
1	0		1	Clear (Reset)
1	1		$\bar{Q}(t)$	Toggle state

5. Let both inputs  $J$  and  $K$  are 1. Previous outputs of master and slave flip-flops are  $Q_M = 0$  and  $Q_S = 0$ , respectively.
6. During high level clock pulse, information is transmitted to master flip-flop because  $C_p \rightarrow 1$  for master and slave flip-flop holds previous output because  $C_p \rightarrow 0$ . So output of master flip-flop are  $Q_M \rightarrow 1$  and  $\bar{Q}_M \rightarrow 0$  and outputs of slave flip-flop are  $Q_S \rightarrow 0$  and  $\bar{Q}_S \rightarrow 1$ .
7. During low level clock pulse, information is transmitted to slave flip-flop because  $C_p \rightarrow 1$  for slave and master flip-flop holds previous output because  $C_p \rightarrow 0$ . So outputs of master flip-flop are  $Q_M \rightarrow 1$  and  $\bar{Q}_M \rightarrow 0$  and outputs of slave flip-flop are  $Q_S \rightarrow 1$  and  $\bar{Q}_S \rightarrow 0$ .
8. During next high level clock pulse, information is transmitted to master flip-flop because  $C_p \rightarrow 1$  for master and slave flip-flop holds previous output because  $C_p \rightarrow 0$  for slave. Input of master flip-flop is  $S_M = JC_p\bar{Q}_S \Rightarrow 1.1.0 \Rightarrow 0$  and  $R_M = KC_pQ_S \Rightarrow 1.1.1 \Rightarrow 1$  will reset the master flip-flop.
9. During low level clock pulse, information is transmitted to slave flip-flop and master flip-flop holds previous output. Slave flip-flop copies master's output. Hence, race around situation is avoided.

**PART-3**

Edge Triggered FF.

**Questions-Answers****Long Answer Type and Medium Answer Type Questions**

**Que 3.5.** Explain edge triggered flip-flop.

**Answer**

1. Edge triggered flip-flop synchronizes state change with clock pulse.

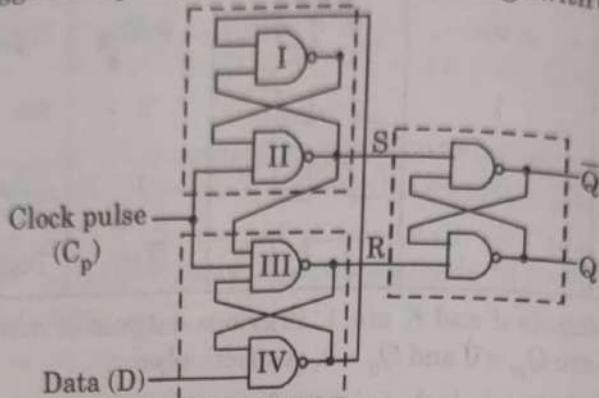


Fig. 3.5.1. D type positive edge triggered flip-flop.

2. Output is transmitted at a specific level of clock pulse otherwise inputs are locked and flip-flop does not perform operation.
3. The operation is resumed when the clock pulse level return to zero and high level pulse occurs.
4. Flip-flop shown in Fig. 3.5.1 is positive edge-triggered flip-flop. Flip-flop can be negative edge-triggered also.
5. Let  $D$  be 0 and clock pulse,  $C_p \rightarrow 0$ . NAND-II and NAND-III gates give outputs  $R \rightarrow 1$  and  $S \rightarrow 1$  as one input;  $C_p$  is 0. Outputs of NAND-I and NAND-IV gates are 0 and 1, respectively. The output will not change any state. So, the inputs are locked.

## PART-4

*State Diagram, State Reduction, Design of Sequential Circuits.*

### Questions-Answers

#### Long Answer Type and Medium Answer Type Questions

**Que 3.6.** Discuss the concept of state equation, state table and state diagram in clocked sequential circuit.

**Answer**

**State equations :**

1. The behavior of a clocked sequential circuit can be described algebraically by means of state equations.

2. A state equation (also called a transition equation) specifies the next state as a function of the present state and inputs a set of state equations for the circuit :

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

$$B(t+1) = \overline{A(t)}x(t)$$

3. The present state value of the output can be expressed algebraically

$$y(t) = [A(t) + B(t)] \overline{x(t)}$$

4. By removing the symbol ( $t$ ) for the present state, we obtain the output boolean equation :

$$y = (A + B) \overline{x}$$

#### State table :

- The time sequence of inputs and flip-flop states can be enumerated in a state table (sometimes called a transition table). The state table for the circuit of Fig. 3.6.1 is shown in Table 3.6.1.
- The Table 3.6.1 consists of four sections labeled present state, input, next state, and output. The present state section shows the states of flip-flops  $A$  and  $B$  at any given time  $t$ .
- The input section gives a value of  $x$  for each possible present state. The next state section shows the states of the flip-flops one clock cycle later, at time  $(t+1)$ .
- The output section gives the value of  $y$  at time  $t$  for each present state and input condition.
- The next state of flip-flop  $A$  must satisfy the state equation

$$A(t+1) = Ax + Bx$$

Table 3.6.1.

Present state		Input	Next state		Output
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

6. The next state of flip-flop  $B$  is derived from the state equation

$$B(t+1) = \overline{Ax}$$

7. The output column is derived from the output equation

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

**State diagram :**

1. The information available in a state table can be represented graphically in the form of a state diagram.
2. In this type of diagram, a state is represented by a circle, and the (clock-triggered) transitions between states are indicated by directed lines connecting the circles.

Table 3.6.2.

Present state		Next state				Output	
		x = 0		x = 1		x = 0	x = 1
A	B	A	B	A	B	y	y
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

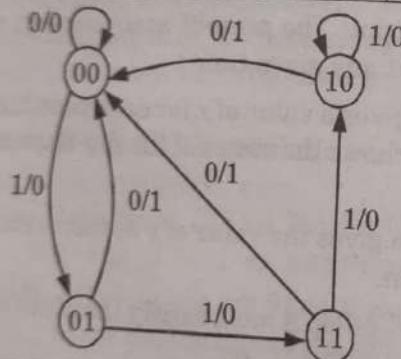


Fig. 3.6.1. State diagram of the circuit.

3. The state diagram provides the same information as the state table and is obtained directly from Table 3.6.1 or Table 3.6.2.

**Que 3.7.** Design the clocked sequential circuit for the following state diagram using JK flip-flop.

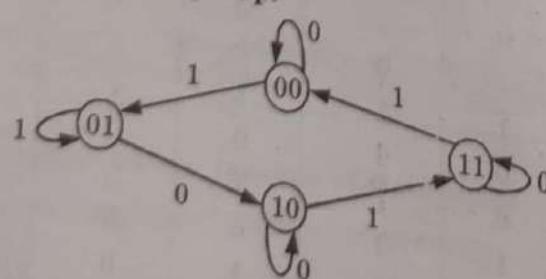


Fig. 3.7.1.

**Answer**

The state table for the given state diagram is (Moore model) :

Input <i>x</i>	Present state		Next state		Flip-flop inputs			
	$Q_1$	$Q_0$	$Q_1$	$Q_0$	$J_1$	$K_1$	$J_0$	$K_0$
0	0	0	0	0	0	x	0	x
0	0	1	1	0	1	x	x	1
0	1	0	1	0	x	0	0	x
0	1	1	1	1	x	0	x	0
1	0	0	0	1	0	x	1	x
1	0	1	0	1	0	x	0	0
1	1	0	1	1	x	0	1	x
1	1	1	0	0	x	1	x	1

Columns of  $J_1, K_1, J_0, K_0$  are filled by the help of excitation table of JK flip-flop.

Excitation table of JK flip-flop :

Present state	Next state	<i>J</i>	<i>K</i>
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

K-maps for  $J_0, K_0, J_1$  and  $K_1$  are :

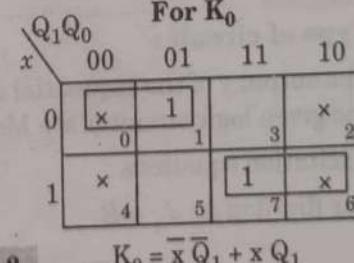
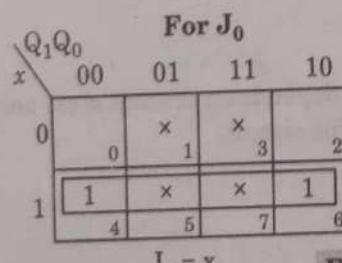
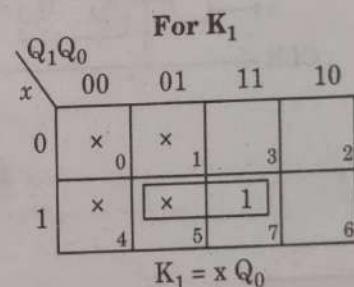
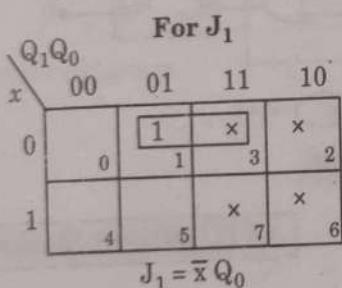


Fig. 3.7.2.

The boolean expressions for  $J_1, K_1, J_0$  and  $K_0$  are :

$$K_0 = \bar{x} \bar{Q}_1 + x Q_1 = \overline{x \oplus Q_1} = x \odot Q_1$$

$$J_0 = x$$

$$K_1 = xQ_0$$

$$J_1 = \bar{x}Q_0$$

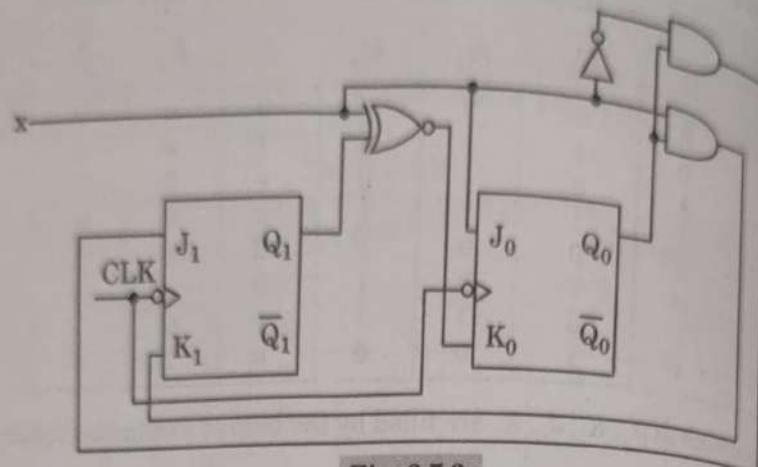


Fig. 3.7.3.

**Que 3.8.** Derive the state table and state diagram for the sequential circuit shown in Fig. 3.8.1.

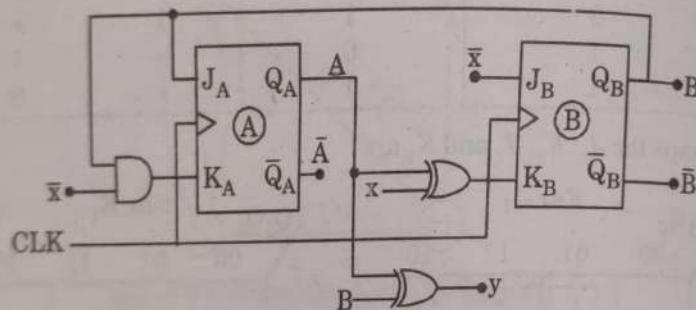


Fig. 3.8.1.

AKTU 2015-16, Marks 10

### Answer

#### 1. Type of circuit :

The output  $y$  of the sequential circuit depends on present state only, so the given logic circuit is the Moore type circuit.

#### 2. Excitation equations :

For flip-flop A :  $J_A = B$

$$K_A = B\bar{x}$$

For flip-flop B :  $J_B = \bar{x}$

$$K_B = A \oplus x$$

For output  $y$ :  $y = A \oplus B$

3. We know that characteristics equation of  $JK$  flip-flop :

$$A_{n+1} = J\bar{Q}_n + \bar{K}Q_n$$

4. State equation for flip-flop  $A$  :

$$\begin{aligned} A_{n+1} &= B\bar{A} + (\bar{B}\bar{x})A && (\because Q_n = A) \\ &= B\bar{A} + A(\bar{B} + x) \\ &= B\bar{A} + A\bar{B} + xA \\ A_{n+1} &= (A \oplus B) + xA \end{aligned}$$

5. State equation for flip-flop  $B$  :

$$\begin{aligned} B_{n+1} &= \bar{x}\bar{B} + (\bar{A} \oplus x)B && (\because Q_n = B) \\ &= \bar{x}\bar{B} + (Ax + \bar{A}\bar{x})B \\ B_{n+1} &= \bar{x}\bar{B} + Ax B + \bar{A}\bar{x}B \end{aligned}$$

State table :

Present state	Next state		Output
	$x = 0$	$x = 1$	
$AB$	$AB$	$AB$	
00	01	00	0
01	11	10	1
10	11	10	1
11	00	11	0

State diagram :

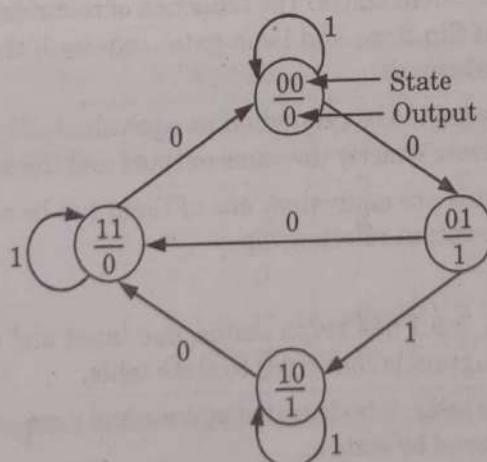


Fig. 3.8.2.

Que 3.9. What do you understand by state reduction ? Reduce the state diagram shown in Fig. 3.9.1.

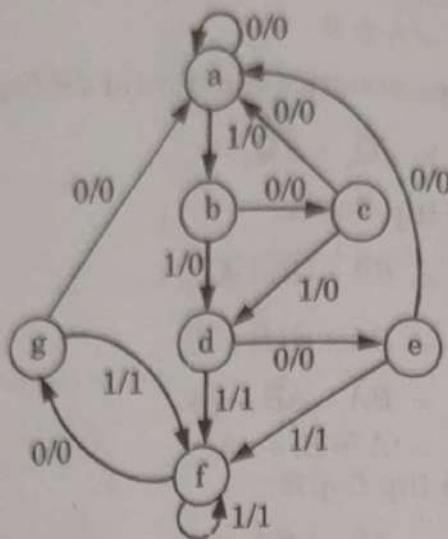


Fig. 3.9.1.

AKTU 2015-16, Marks 10

OR

Draw the reduced state table and reduced state diagram for the state table given in Fig. 3.9.1.

AKTU 2018-19, Marks 07

**Answer****State reduction :**

1. Any logic design process must consider the problem of minimizing the cost of the final circuit. One way to reduce the cost is, by reducing the number of flip-flops, i.e., by reducing the number of states.
2. The state reduction technique basically avoids the introduction of redundant equivalent states. The reduction of redundant states reduces the number of flip-flops and logic gates required, thus reducing the cost of the final circuit.
3. Two states are said to be redundant or equivalent, if every possible set of inputs generate exactly the same outputs and the same next states.
4. When two states are equivalent one of them can be removed without altering input-output relationship.

**Numerical :**

1. The given Fig. 3.9.1 has seven states, one input and one output. The given state diagram is converted to state table.
2. From the state table, it is clear that states *e* and *g* are equivalent. So the state *g* is replaced by state *e*.

State table :

Present state	Next state		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
$a$	$a$	$b$	0	0
$b$	$c$	$d$	0	0
$c$	$a$	$d$	0	0
$d$	$e$	$f$	0	1
$e$	$a$	$f$	0	1
$f$	$g$	$f$	0	1
$g$	$a$	$f$	0	1

Both are equivalent states because of state  $e$  and  $g$  having same next state and same output.

Reducing the state table :

Present state	Next state		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
$a$	$a$	$b$	0	0
$b$	$c$	$d$	0	0
$c$	$a$	$d$	0	0
$d$	$e$	$f$	0	1
$e$	$a$	$f$	0	1
$f$	$e$	$f$	0	1

Both are equivalent states

3. From the reduced table, states  $d$  and  $f$  are equivalent, hence  $f$  can be replaced by  $d$  and it can be removed.

Reduced table :

Present State	Next state		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
$a$	$a$	$b$	0	0
$b$	$c$	$d$	0	0
$c$	$a$	$d$	0	0
$d$	$e$	$d$	0	1
$e$	$a$	$d$	0	1

4. The state diagram of the reduced state table is shown in Fig. 3.9.2.

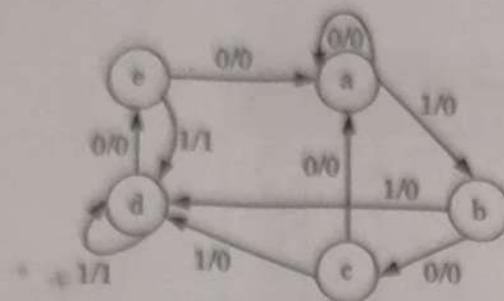


Fig. 3.9.2. State diagram.

**Que 3.10.** Derive the state table and state diagram for the sequential circuit is shown in Fig. 3.10.1.

AKTU 2018-19, Marks 07

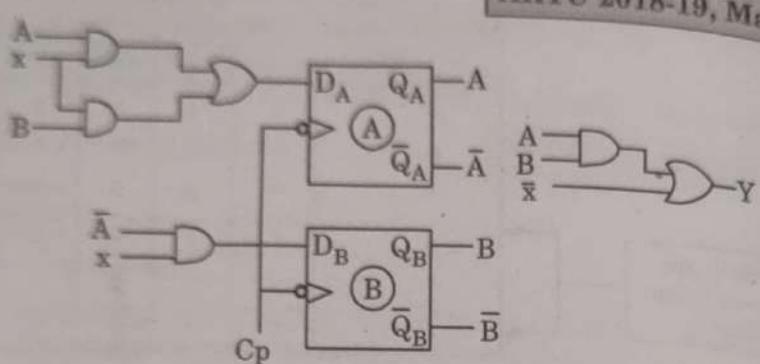


Fig. 3.10.1.

### Answer

- The behaviour of circuit is determined by the following Boolean expression,

$$Y = AB + \bar{x} \quad \dots(3.10.1)$$

$$D_A = Ax + Bx \quad \dots(3.10.2)$$

$$D_B = \bar{A}x \quad \dots(3.10.3)$$

- From eq. (3.10.1), (3.10.2) and (3.10.3) then state table will be

Table 3.10.1.

Present State		Next state		Output	
A	B	$x = 0$	$x = 1$	$x = 0$	$x = 1$
0	0	0 0	0 1	1	0
0	1	0 0	1 1	1	0
1	0	0 0	1 0	1	0
1	1	0 0	1 0	1	1

3. We draw state diagram with the help of state table

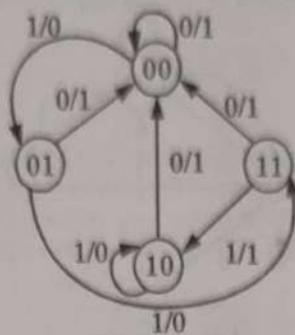


Fig. 3.10.2.

**Que 3.11.** Derive the state table and state diagram of the synchronous sequential circuit shown below ( $X$  is an input to the circuit). Explain the circuit function. AKTU 2017-18, Marks 07

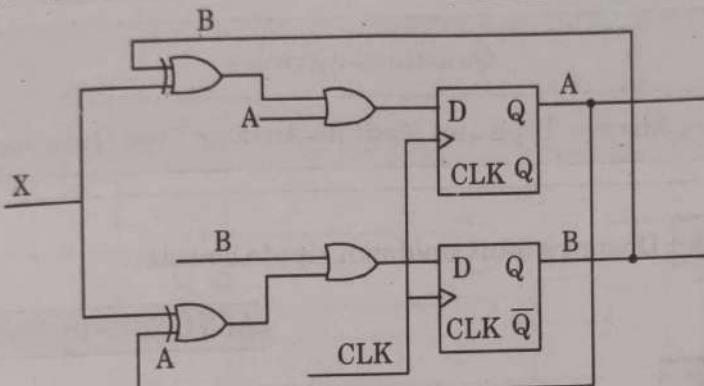


Fig. 3.11.1.

**Answer**

1. From the circuit shown in Fig. 3.11.1, the output equation can be obtained as,

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

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

2. The state table for the circuit shown in Fig. 3.11.1.

Present state		Input	Next state	
A	B	X	A(t+1)	B(t+1)
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

## 3. State diagram :

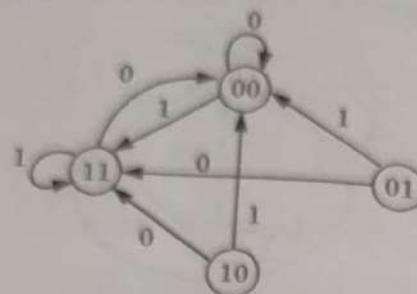


Fig. 3.11.2.

## PART-5

## Ripple and Synchronous Counters.

## Questions-Answers

## Long Answer Type and Medium Answer Type Questions

**Que 3.12.** Design a 3-bit up/down ripple counter.

**AKTU 2017-18, Marks 07**

**Answer**

1. The 3-bit up/down ripple counter, which can count in upward direction of sequence from 000, 001, 010, 011, 100, 101, 110, 111 and downward direction of sequence from 111, 110, 101, 100, 011, 010, 001, 000.
2. 3-bit counter consists of 3 flip-flops. In ripple counter, a flip-flop output transition serves as a source for triggering other flip-flops.
3. The control signal  $M$  is used to select the direction of count sequence. Fig. 3.12.1 shows the 3-bit ripple.

$M = 1$ ; counter acts as up-counter

$M = 0$ ; counter acts as down counter

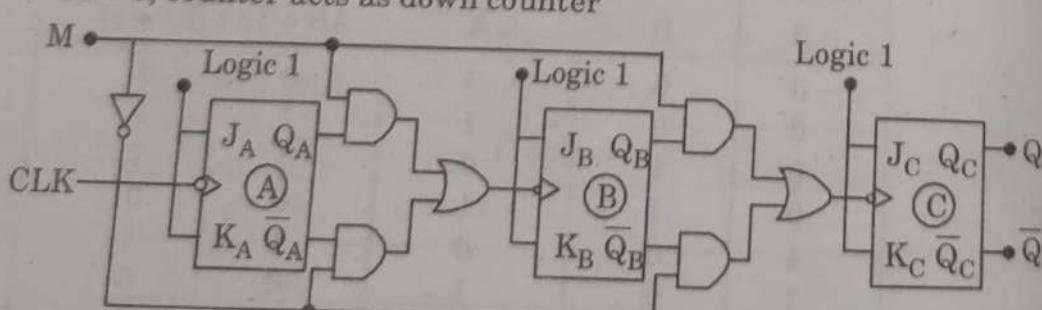


Fig. 3.12.1. 3-bit ripple up-down.

**Que 3.13.** Draw the logic diagram of a 4-bit binary counter with parallel load.

**Answer**

1. The operation of the counter is summarized in Table 3.13.1. The four control inputs—Clear, CLK, load, and count, determine the next state.
2. The clear input is asynchronous and when equal to 0, causes the counter to be cleared regardless of the presence of clock pulses or other inputs.

Table 3.13.1.

Clear	CLK	Load	Count	Function
0	x	x	x	Clear to 0
1	1	1	x	Load inputs
1	1	0	1	Count next binary state
1	1	0	0	No change

3. The logic diagram for 4-bit binary counter with parallel load is shown in Fig. 3.13.1.

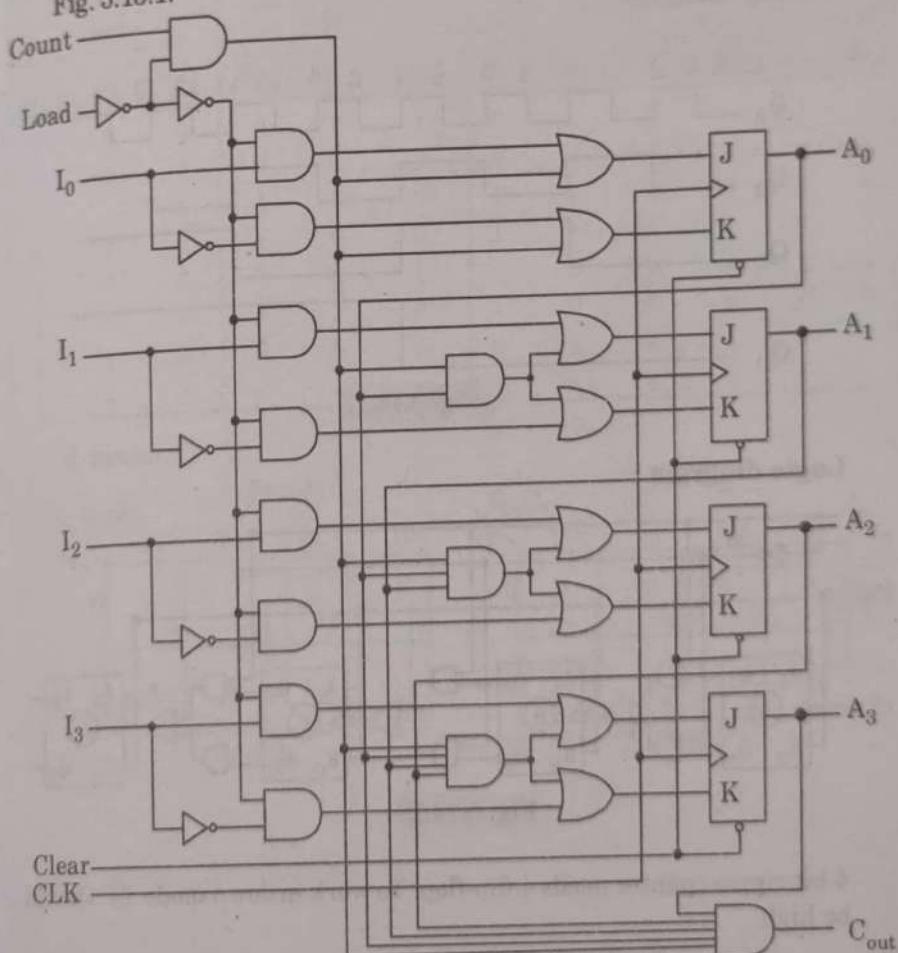


Fig. 3.13.1.

**Que 3.14.** Draw diagram of a 4-bit binary ripple down counter using flip-flops that trigger on negative edge transition. Also draw a timing diagram of the counter.

**OR**

Design a ripple decade counter using JK flip-flop.

AKTU 2018-19, Marks 3.5

**Answer**

**4-bit binary ripple down counter :**

1. The 4-bit asynchronous counter is constructed by using JK flip-flop (asynchronous counter are also called ripple counter).
2. The output  $Q_A$  must be externally connected to clock input of flip-flop  $B$ .
3. The input count pulses are applied to clock input of flip-flop  $A$ . Simultaneous divisions of 2, 4, 8, and 16 are performed at the  $Q_A, Q_B, Q_C, Q_D$  outputs.

**Timing diagram :**

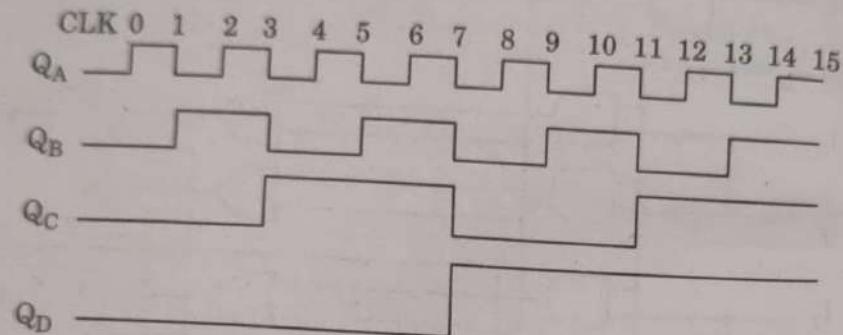


Fig. 3.14.1.

**Logic diagram :**

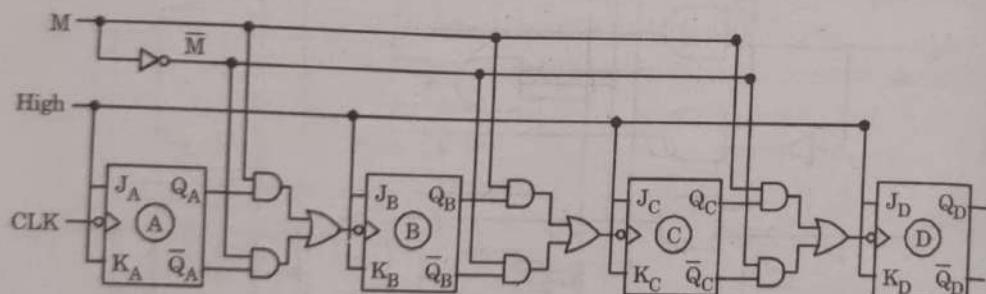


Fig. 3.14.2.

4-bit ripple counter needs 4 flip-flop. To work in down mode  $\bar{M}$  should be high.

**Que 3.15.** Design a synchronous counter using JK flip-flop for the following input sequences :

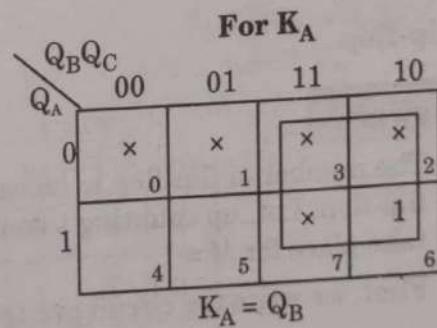
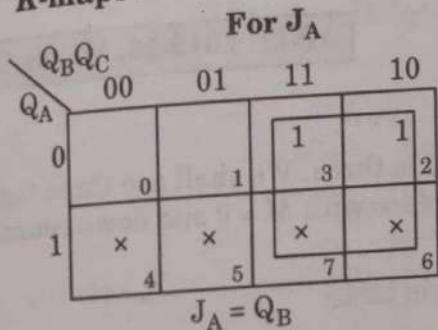
A	B	C
0	0	0
0	0	1
0	1	0
1	0	0
1	0	1
1	1	0
0	0	0



Fig. 3.15.1.

**Answer**

Present state			Next state			Flip-Flop inputs					
$Q_A$	$Q_B$	$Q_C$	$Q_{A+1}$	$Q_{B+1}$	$Q_{C+1}$	$J_A$	$K_A$	$J_B$	$K_B$	$J_C$	$K_C$
0	0	0	0	0	1	0	x	0	x	1	x
0	0	1	0	1	0	0	x	1	x	x	1
0	1	0	1	0	0	1	x	x	1	0	x
0	1	1	1	0	0	1	x	x	1	x	1
0	1	0	1	0	1	x	0	0	x	1	x
1	0	0	1	1	0	x	0	1	x	x	1
1	0	1	1	1	0	x	1	x	1	0	x
1	1	0	0	0	0	x	x	x	x	x	x
1	1	1	x	x	x						

**K-maps:**

		For $J_B$						For $K_B$							
		Q <sub>B</sub> Q <sub>C</sub>	00	01	11	10			Q <sub>B</sub> Q <sub>C</sub>	00	01	11	10		
		Q <sub>A</sub>	0	1	x	x			Q <sub>A</sub>	0	x	1	1		
		0	0	1	3	2			0	0	1	3	1		
		1	4	5	7	6			1	x	x	5	7		

$J_B = Q_C$

		For $J_C$						For $K_C$							
		Q <sub>B</sub> Q <sub>C</sub>	00	01	11	10			Q <sub>B</sub> Q <sub>C</sub>	00	01	11	10		
		Q <sub>A</sub>	0	1	x	x			Q <sub>A</sub>	0	1	1	x		
		0	0	1	3	2			0	0	1	3	2		
		1	4	5	7	6			1	x	1	x	x		

$J_C = \bar{Q}_B$

Flip-flop required are :  $2^n \geq N$

Here  $N = 6$

So,  $n = 3$ , i.e., three flip-flops are required.

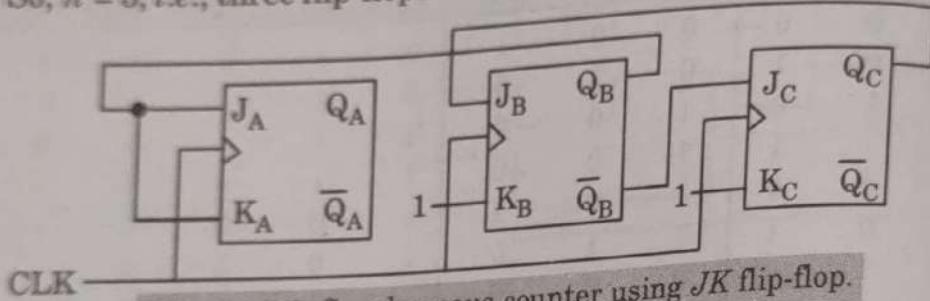


Fig. 3.15.2. Synchronous counter using JK flip-flop.

**Que 3.16.** Design a 3-bit asynchronous up-down counter using T flip-flop.

AKTU 2015-16, Marks 7.5

### Answer

- The number of flip-flop to be used is three. We shall use three toggle flip-flop. Let, up counting takes place with  $M = 0$  and down counting take place for  $M = 1$ .
- First, we write the circuit excitation table.

Table 3.16.1 : Excitation table for a 3-bit up/down synchronous counter

Mode control $M$	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	0	1	0	0	1
0	0	0	1	0	1	0	0	1	1
0	0	1	0	0	1	1	0	0	1
0	0	1	1	1	0	0	1	1	1
0	1	0	0	1	0	1	0	0	1
0	1	0	1	1	1	0	0	1	1
0	1	1	0	1	1	1	0	0	1
0	1	1	1	0	0	0	1	1	1
0	0	0	0	1	1	1	1	1	1
1	0	0	1	0	0	0	0	0	1
1	0	1	0	0	0	1	0	1	1
1	0	1	1	0	1	0	0	0	1
1	1	0	0	0	1	1	1	1	1
1	1	0	1	1	0	0	0	1	1
1	1	1	0	1	0	1	0	1	1
1	1	1	1	1	1	0	0	0	1

		For $T_C$				
		$Q_B Q_A$	00	01	11	10
$M Q_C$	$Q_B Q_A$	00	0 0	0 1	1 3	0 2
		01	0 4	0 5	1 7	0 6
$M Q_C$	$Q_B Q_A$	11	1 12	0 13	0 15	0 14
		10	1 8	0 9	0 11	0 10

$$T_C = \overline{M} Q_B Q_A + M \overline{Q}_B \overline{Q}_A$$

		For $T_B$				
		$Q_B Q_A$	00	01	11	10
$M Q_C$	$Q_B Q_A$	00	0 0	1 1	1 3	0 2
		01	0 4	1 5	1 7	0 6
$M Q_C$	$Q_B Q_A$	11	1 12	0 13	0 15	1 14
		10	1 8	0 9	0 11	1 10

$$T_B = \overline{M} Q_A + \overline{Q}_A M \\ = M \oplus Q_A$$

		For $T_A$				
		$Q_B Q_A$	00	01	11	10
$M Q_C$	$Q_B Q_A$	00	1 0	1 1	1 3	1 2
		01	1 4	1 5	1 7	1 6
$M Q_C$	$Q_B Q_A$	11	1 12	1 13	1 15	1 14
		10	1 8	1 9	1 11	1 10

$$T_A = 1$$

4. Finally, let us draw the logic diagram.

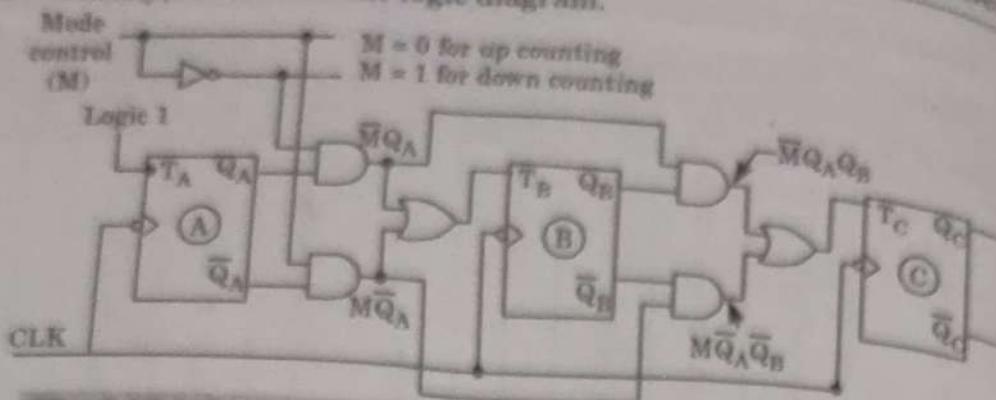


Fig. 3.16.1. Logic diagram of a 3-bit synchronous up/down counter.

**Que 3.17.** Design a 3-bit synchronous counter using JK flip-flops.

**Answer**

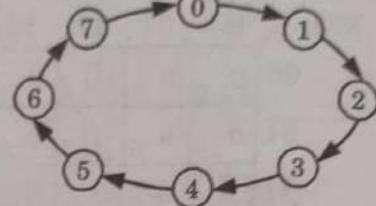
**3-bit synchronous counter :**

For a 3-bit synchronous counter using JK flip-flop, we need 3 flip-flops.

**Excitation table and state diagram of JK flip-flop :**

Present state	Next state	Flip-flop inputs	
$Q_n$	$Q_{n+1}$	$J$	$K$
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

Excitation table of JK flip-flop



State diagram.

Fig. 3.17.1.

**Excitation table for 3-bit synchronous counter :**

Table 3.17.1: Circuit excitation table

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

Digital System Design  
K-maps and simplified expressions for all flip-flop inputs :

		For $J_C$	
		$Q_B Q_A$	
		00	01
		11	10
$Q_C$			
0	0	0	1
1	1	1	x
		x	0
		5	7
		x	6

$$J_C = Q_B Q_A$$

		For $K_C$	
		$Q_B Q_A$	
		00	01
		11	10
$Q_C$			
0	x	x	x
1	0	1	0
	4	5	7
	x	1	0
	2	6	8

$$K_C = Q_B Q_A$$

		For $J_B$	
		$Q_B Q_A$	
		00	01
		11	10
$Q_C$			
0	0	1	1
1	1	1	x
	4	5	7
	x	3	2
	6	7	8

$$J_B = Q_A$$

		For $K_B$	
		$Q_B Q_A$	
		00	01
		11	10
$Q_C$			
0	x	1	1
1	x	1	0
	4	5	7
	x	1	0
	6	7	8

$$K_B = Q_A$$

		For $J_A$	
		$Q_B Q_A$	
		00	01
		11	10
$Q_C$			
0	1	0	x
1	1	x	1
	4	5	x
	x	7	6
	2	1	0

$$J_A = 1$$

Thus the simplified equations are :

$$J_C = Q_B Q_A$$

$$K_C = Q_B Q_A$$

$$J_B = Q_A$$

$$K_B = Q_A$$

$$J_A = 1$$

$$K_A = 1$$

Logic diagram :

Fig. 3.17.2 shows the logic diagram of a 3-bit synchronous counter using JK flip-flops.

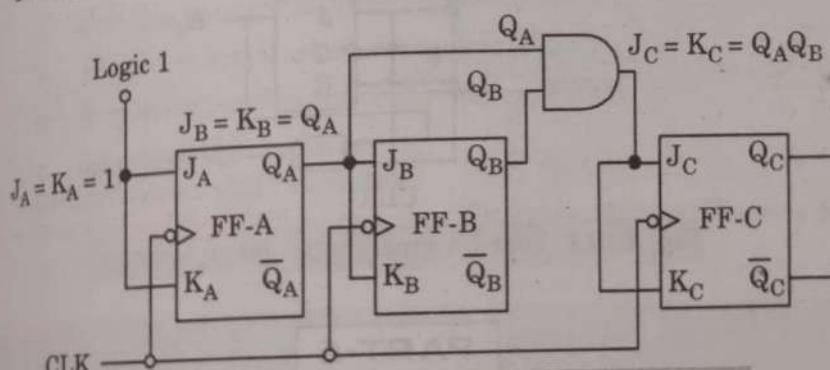


Fig. 3.17.2. 3-bit synchronous counter using JK FFs.

**Que 3.18.** Describe the operation of four bit synchronous binary counter with neat sketch.

**Answer**

1. The  $C$  inputs of all flip-flops are connected to a common clock. The counter is enabled by count enable.
2. If the enable input is 0, all  $J$  and  $K$  inputs are equal to 0 and the clock does not change the state of the counter.
3. The first stage  $A_0$  has its  $J$  and  $K$  equal to 1 if the counter is enabled. The other  $J$  and  $K$  inputs are equal to 1 if all previous least significant stages are equal to 1 and the count is enabled.
4. The chain of AND gates generates the required logic for the  $J$  and  $K$  inputs in each stage.
5. The counter can be extended to any number of stages, with each stage having an additional flip-flop and an AND gate that gives an output of 1 if all previous flip-flop outputs are 1.

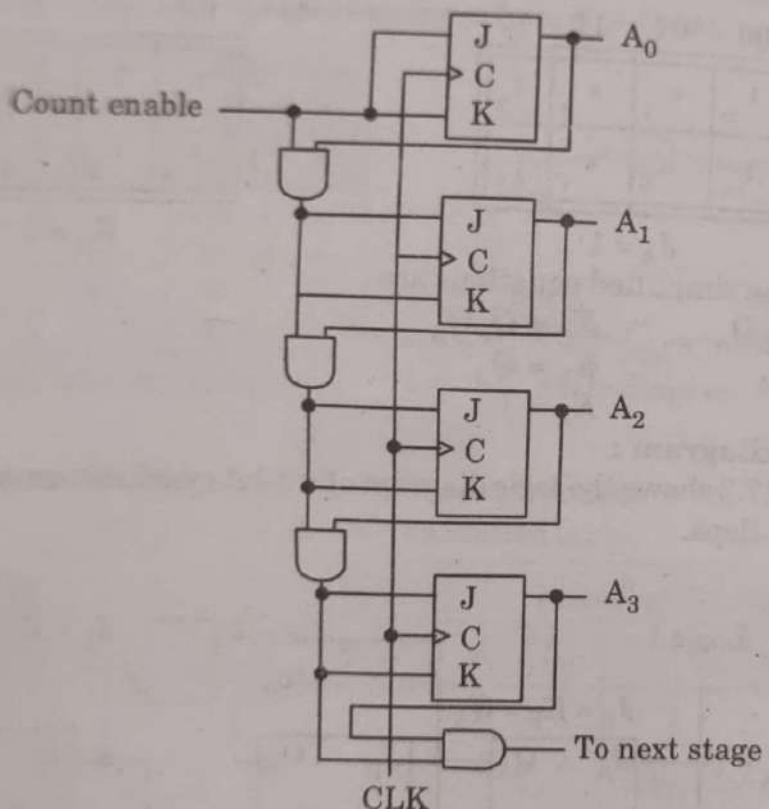


Fig. 3.18.1. Four-bit synchronous binary counter.

## Questions-Answers

## Long Answer Type and Medium Answer Type Questions

Que 3.19. Write down the classification of shift registers.

OR

What do you mean by shift register ? What is the need of shift register ? Draw and explain bidirectional shift register.

AKTU 2018-19, Marks 07

**Answer****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 shift registers.

**Need of a register :** A register is a sequential logic circuit with two basic functions :

- i. Temporary storage.
- ii. Shifting capability.

**Classification of shift registers :**

1. Classification based on the direction of data movement :

- i. Shift left register.
- ii. Shift right register.
- iii. Bidirectional shift registers.

2. Classification based on the mode of input and output :

- i. Serial in serial out shift register (SISO)
- ii. Serial in parallel out shift register (SIPO)
- iii. Parallel in serial out shift register (PISO)
- iv. Parallel in parallel out shift register (PIPO)
- v. Universal shift register.

**Bidirectional shift register :**

1. It consists of four  $D$  flip-flops, four OR gates, eight AND gates and one NOT gate as shown in Fig. 3.19.1.

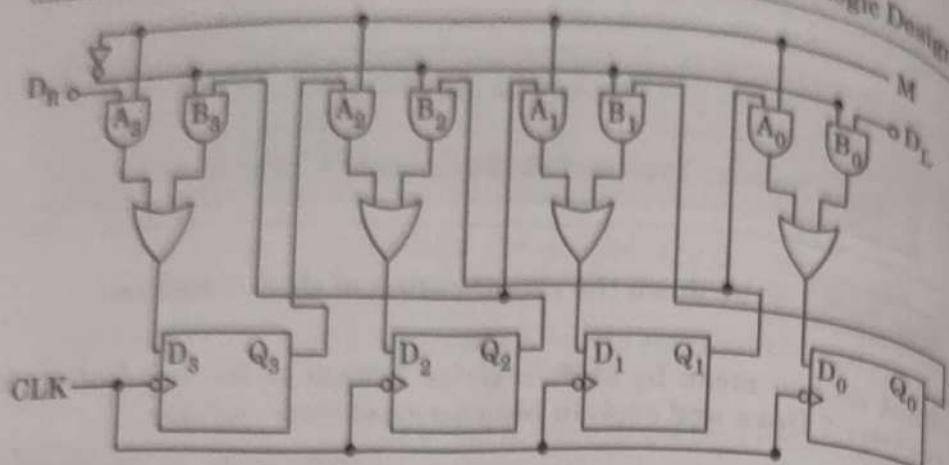


Fig. 3.19.1.

**Operation :**

- When mode control  $M = 1$ , all the A AND gates ( $A_3, A_2, A_1, A_0$ ) are enabled and the data at  $D_R$  is shifted to the right when clock pulses are applied.
- When  $M = 0$ , all A gates are disabled and all B gates are enabled. These enabled B gates allow data  $D_L$  to be shifted to left.
- $M$  should be changed only when  $CLK = 0$ , otherwise the data stored in the register may be changed.

**Que 3.20.** Write a short note on different types of shift register.

**OR**

Draw and explain the PISO, PIPO register.

**AKTU 2017-18, Marks 07**

**Answer**

**Serial in serial out shift register (SISO) :**

- The serial in serial out shift register accepts the data serially on a single input line.
- It also produces the stored information on its output in serial form. We can shift the data from left side or right side.
- Based on the shifting of data, the register is called shift left or shift right register. Fig. 3.20.1 shows the block diagram of serial in serial out shift register (SISO).

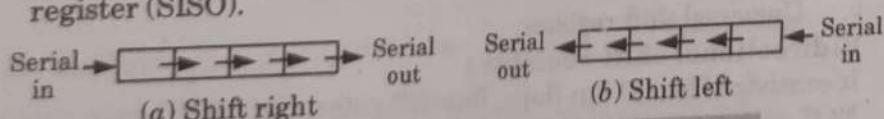


Fig. 3.20.1. 4-bit serial in serial out shift register.

**Shift right register :**

- In this register while accepting data serially, the group of bits is shifted towards the right side.

2. Hence the serial data is entered onto the left side of register and it leaves from the right side serially. Fig. 3.20.2 shows the logic circuit for a 4-bit shift right register.

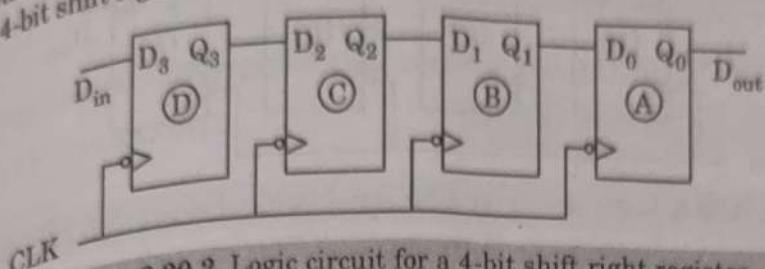


Fig. 3.20.2. Logic circuit for a 4-bit shift right register.

#### Shift left register :

1. The group of bits is shifted towards the left side in serial form. Hence the serial data is entered from right and the binary data at the output is taken from the left most flip-flop.
2. Fig. 3.20.3 shows the logic circuit for a 4-bit shift left register.

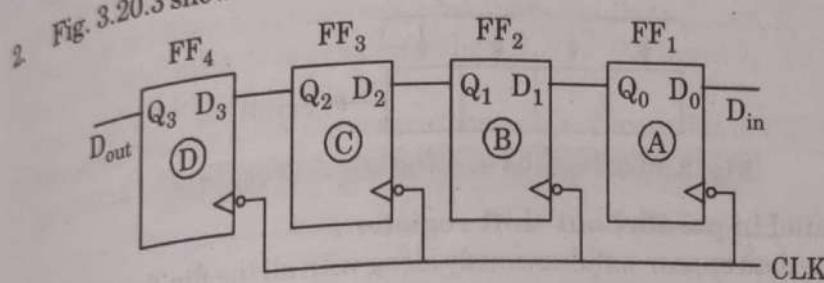


Fig. 3.20.3. Logic circuit for a 4-bit shift left register.

3. The binary data is entered into right most flip-flop ( $FF_1$ ) and output is taken from the left most flip-flop ( $FF_4$ ) in serial form.

#### Serial in parallel out shift register :

1. This is one type of shift register in which the data is entered in serial form and output is in parallel form.
2. Hence, it is necessary to have all the data bits available as outputs at the same time.
3. This type of shift register operation is same as the serial in serial out shift register.
4. The difference between serial out and parallel out shift registers is the way in which the data bits are taken out of the register.
5. Fig. 3.20.4 shows the block diagram of 4-bit serial in parallel out shift register.

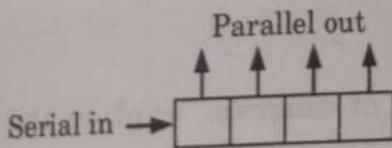
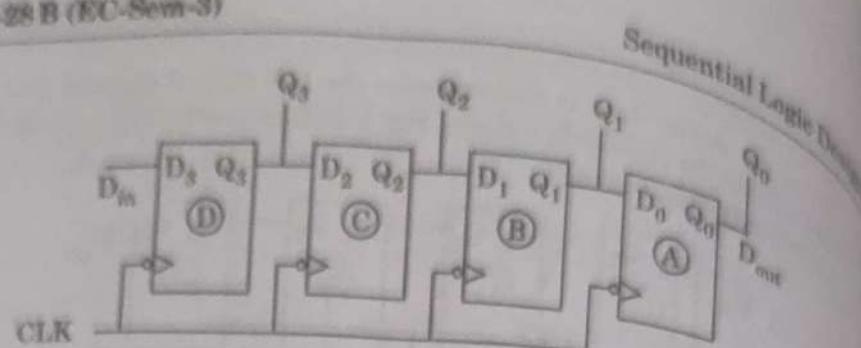


Fig. 3.20.4. Block diagram of 4-bit serial in parallel out shift register.

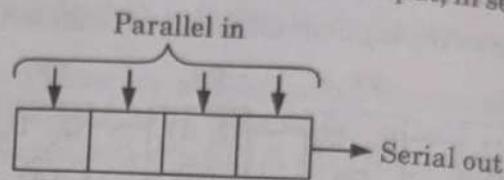
5. Fig. 3.20.5 shows the logic circuit for 4-bit serial-in parallel out shift register.



**Fig. 3.20.5.** Logic circuit for 4-bit serial in parallel out shift register.

#### Parallel in serial out shift register :

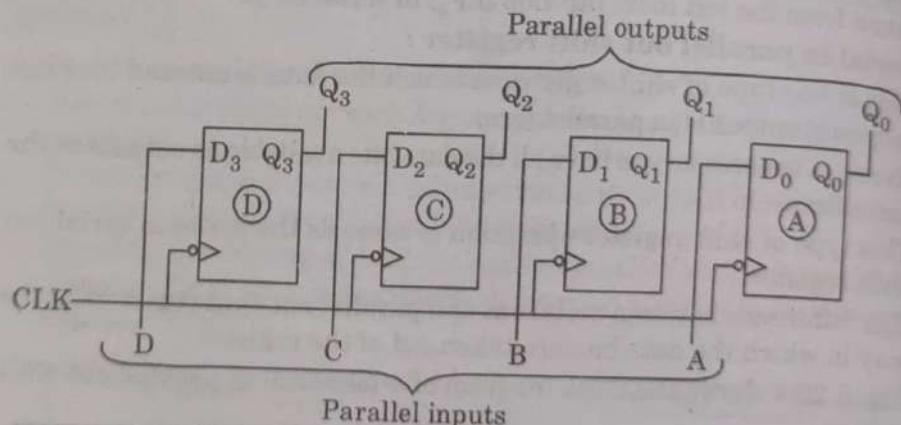
1. Fig. 3.20.6 shows the block diagram of a parallel in serial out shift register. In this type, the bits are entered in parallel, i.e., simultaneously into their respective stages on a parallel line.
2. It produces the stored information on its output, in serial form.



**Fig. 3.20.6.** Parallel in serial out shift register.

#### Parallel in parallel out shift register :

1. All the data appear simultaneously along with all the flip-flop inputs and outputs.
2. Fig. 3.20.7 shows the logic diagram for 4-bit parallel in parallel out shift register.



**Fig. 3.20.7.** Logic diagram for 4-bit parallel in parallel out shift register.

**Que 3.21.** With the help of diagram, explain the operation of universal shift register.

AKTU 2014-15, Marks 06

OR

Draw and explain 4-bit universal shift register.

AKTU 2015-16, Marks 10

OR

Design a universal shift register that performs HOLD, SHIFT RIGHT, SHIFT LEFT, and LOAD.

AKTU 2017-18, Marks 07

**Answer**

1. A shift register that can shift the data in both the directions (shift right or left) as well as load it parallelly, it is called as a universal shift register.
2. This shift register is capable of performing the following operations :
  - i. Parallel loading (parallel input parallel output).
  - ii. Left shifting.      iii. Right shifting.
3. The block diagram of a 4-bit universal shift register is shown in Fig. 3.21.1. It consists of four D flip-flop and four 4 : 1 multiplexers.
4. The four multiplexers have two common select lines  $S_1$  and  $S_0$ . Input  $I_0$  in each multiplexer is selected when  $S_1S_0 = 00$ , input  $I_1$  is selected when  $S_1S_0 = 01$  and so on.
5. The selection inputs ( $S_1S_0$ ) control the mode of operation of the register according to the function table shown in table 3.21.1.

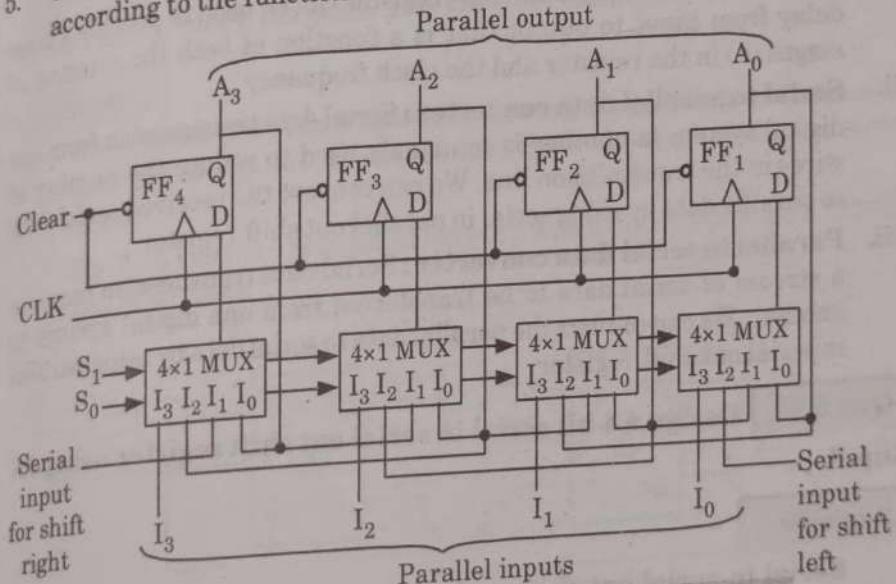


Fig. 3.21.1. Block diagram of 4-bit universal shift register.

Table 3.21.1. Function table

$S_1$	$S_0$	Function
0	0	Hold
0	1	Shift right
1	0	Shift left
1	1	Load

**Operation :**

1. When  $S_1S_0 = 00$ , the present value of the register is applied to the inputs of the flip-flops. This condition forms a path from the output of each flip-flop to the binary value input of the same flip-flop.
2. The next clock edge transfers into each flip-flop the binary value it held previously and no change of state occurs.
3. When  $S_1S_0 = 01$ , the input  $I_1$  of the multiplexer has a path to the inputs of the flip-flops. This causes a shift right operation, with the serial input transferred in flip-flop  $FF_4$ .
4. When  $S_1S_0 = 10$ , a shift left operation results, with the other serial input going into flip-flop  $FF_1$ . In this case, Input  $I_2$  of each multiplexer is connected to the output of each flip-flop. The data bit is shifted to left side for every clock.
5. When  $S_1S_0 = 11$ , the binary information on the parallel input lines is transferred into the register simultaneously during the next clock edge.

**Que 3.22.** What are the applications of shift register ?

**Answer**

- i. **Time delay :** The serial out shift register can be used to provide a time delay from input to output that is a function of both the number of stages ( $n$ ) in the register and the clock frequency.
- ii. **Serial to parallel data converter :** Serial data transmission from one digital system to another is commonly used to reduce the number of wires in the transmission line. We can convert the received serial data to parallel data by using serial in parallel out shift register.
- iii. **Parallel to serial data converter :** Serial data transmission requires a stream of serial data to be transferred from one digital system to another. We can convert the parallel data to serial data by using parallel in serial out shift register.

**Que 3.23.** Design a 4-bit serial in serial out shift register using JK flip-flop.

**Answer**

**Serial in-serial out shift register using JK flip-flop :**

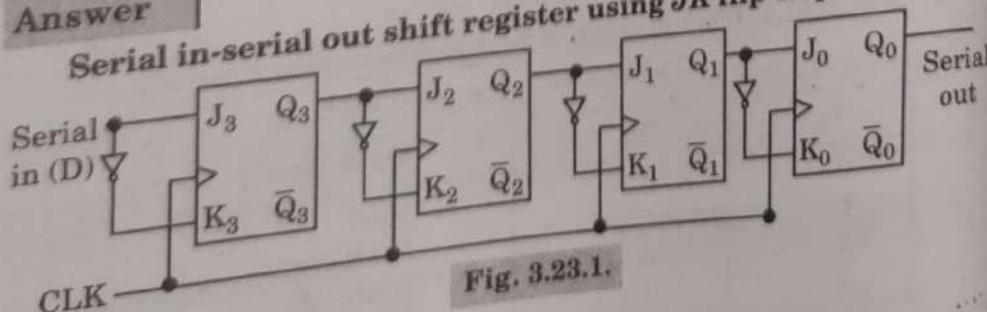


Fig. 3.23.1.

$D$	$Q_n$	$Q_{n+1}$	$J$	$K$
0	0	0	0	x
0	1	0	x	1
0	0	1	1	x
1	1	1	x	0
1	1	0	2	3

$D$	$Q_n$	$\bar{Q}_n$	$Q_n$
D			x
$\bar{D}$		0	1
D	1		x
$\bar{D}$	2		3

$$J = D$$

$D$	$Q_n$	$\bar{Q}_n$	$Q_n$
D		x	1
$\bar{D}$	0		1
D	x		2
$\bar{D}$	2		3

$$K = \bar{D}$$

### PART-7

Finite State Machines (FSM), Design of Synchronous FSM.

#### Questions-Answers

#### Long Answer Type and Medium Answer Type Questions

Que 3.24. Explain Moore type of synchronous sequential machine (FSM) using block diagram and suitable example.

#### Answer

- When the output of the sequential network depends only on the present state of the flip-flop, the sequential network is referred to as Moore model.
- Fig. 3.24.1 shows a sequential network which consists of two JK flip-flop and AND gates.

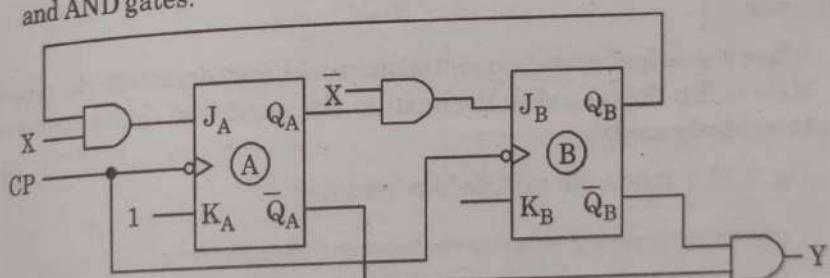
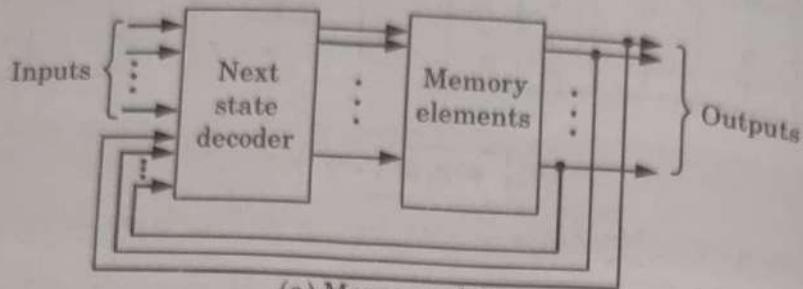


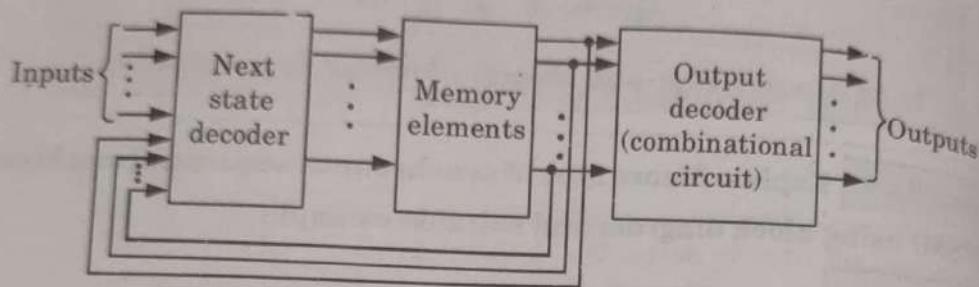
Fig. 3.24.1. Example of Moore model.

- The network has one input  $X$  and one output  $Y$ .
- As shown in the Fig. 3.24.2, input is used to determine the inputs of the flip-flops.
- It is not used to determine the output.
- The output is derived using only present states of the flip-flops or combination of it (in this case  $\bar{Q}_A \bar{Q}_B$ ).

7. In general form the Moore model can be represented with its block schematic as shown in Fig. 3.24.2 (a) and (b).
8. In the Moore model, as output depends only on present state of the flip-flops, it appears only after the clock pulse is applied, i.e., it varies in synchronism with the clock input.



(a) Moore model.



(b) Moore circuit model with an output decoder.

Fig. 3.24.2.

**Que 3.25.** Explain Mealy model.

**Answer**

- When the output of the sequential network depends on both the present state of flip-flop(s) and on the input(s), the sequential circuit is referred to as Mealy model.
- Fig. 3.25.1 shows the sample Mealy model.

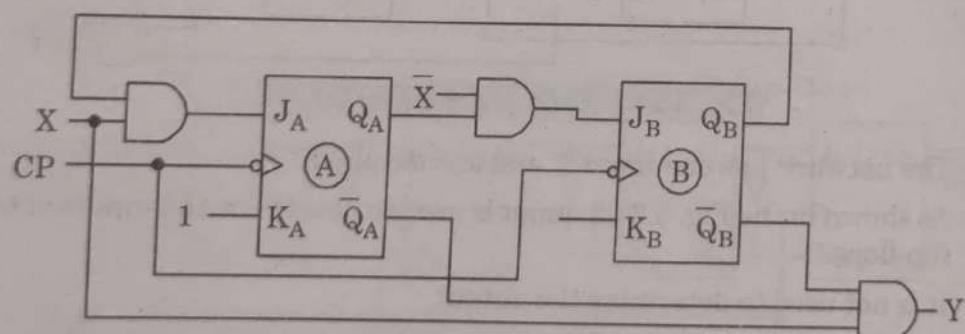


Fig. 3.25.1. Example of Mealy model.

3. As shown in the Fig. 3.25.1, the output of the circuit is derived from the combination of present state of flip-flops and input( $X$ ) of the circuit.
4. Looking at Fig. 3.25.1, we can easily realize that, changes in the input within the clock pulses cannot affect the state of the flip-flop. However, they can affect the output of the circuit.
5. If the input variations are not synchronized with the clock, the derived output will also not be synchronized with the clock and we get false output (as it is synchronous sequential network).
6. The false outputs can be eliminated by allowing input to change only at the active transition of the clock (in our example HIGH-to-LOW).
7. In general form the Mealy model can be represented with its block schematic as shown in Fig. 3.25.2.

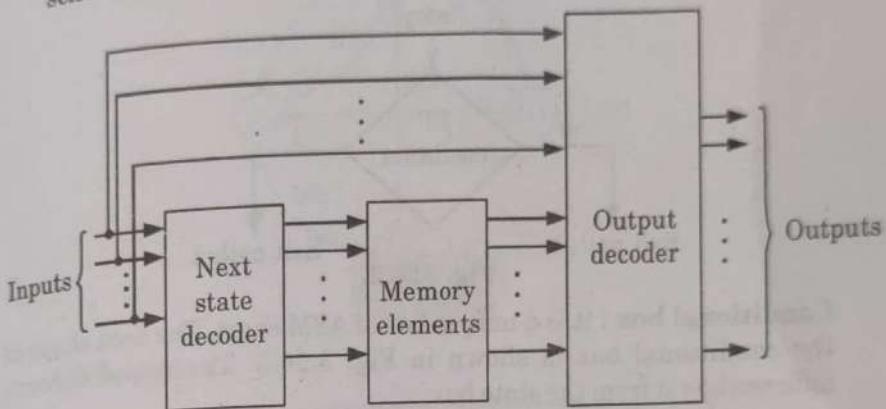


Fig. 3.25.2. Mealy circuit model.

**PART-B****Algorithmic State Machines (ASM) Charts.****Questions-Answers****Long Answer Type and Medium Answer Type Questions**

**Que 3.26.** What is ASM chart ? Explain. Draw the state diagram, state table and ASM chart for a D flip-flop.

**Answer**

ASM chart is composed of three basic elements : State box, decision box and conditional box.

**State box :** The state of the system is indicated by a state box. The shape of the state box is a rectangle.

General description entry

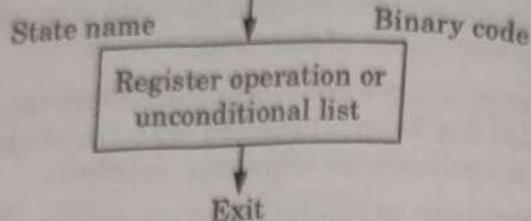


Fig. 3.26.1.

**Decision box :** It is a diamond shaped box used to describe the effect of an input on the control subsystem.

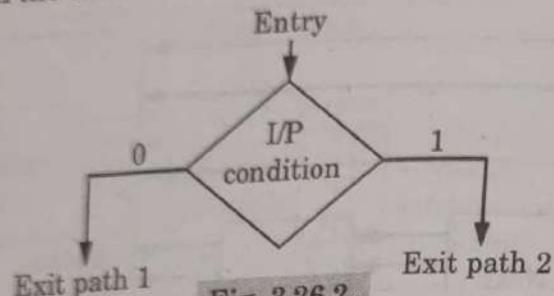


Fig. 3.26.2.

**Conditional box :** It is a unique box of ASM chart. The area shape of the conditional box is shown in Fig. 3.26.3. The round corners differentiate it from the state box.

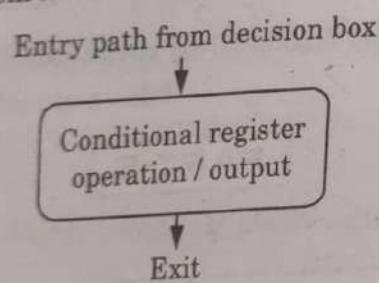


Fig. 3.26.3.

To draw ASM chart, first we form a state table for easy understanding of the operation of the given circuit.

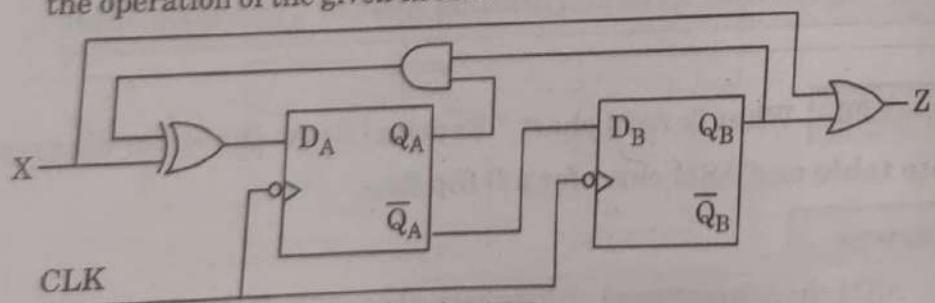


Fig. 3.26.4.

state table :

Present state		Next state, output					
		X = 0		X = 1		Z	
$Q_A$	$Q_B$	$Q_{A+1}$	$Q_{B+1}$	Z	$Q_{A+1}$	$Q_{B+1}$	Z
0	0	0	1	1	1	1	1
0	1	0	1	1	1	1	1
1	0	0	0	0	1	0	1
1	1	1	0	0	0	0	1

state diagram :

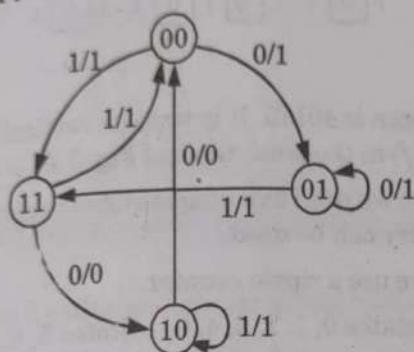


Fig. 3.26.5.

ASM chart :

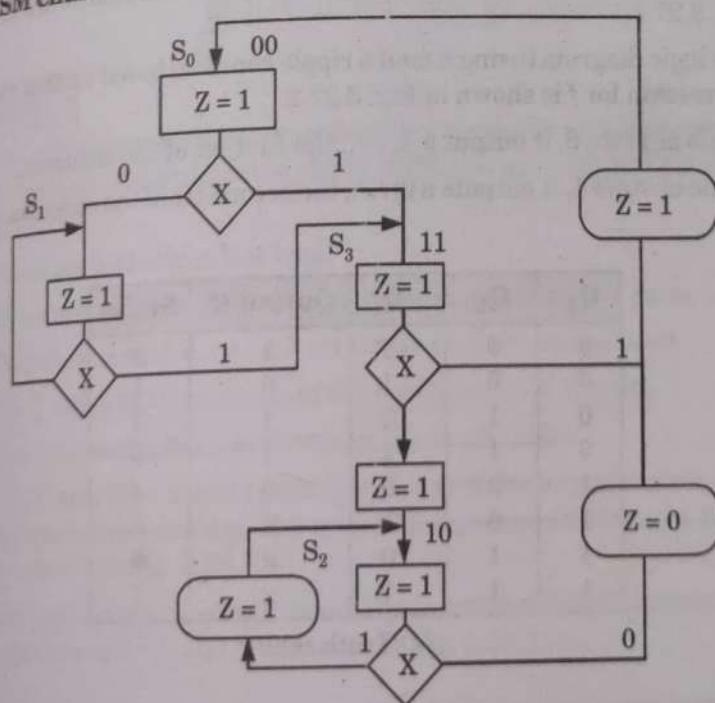


Fig. 3.26.6.

**PART-9****Questions-Answers****Long Answer Type and Medium Answer Type Questions**

**Que 3.27.** Generate the following pulse train using indirect logic.

1 0 1 1 0 1 0 1 1 0 1

**Answer**

1. The given sequence is 10110. It is written vertically under the column heading output ( $f$ ) in the truth table of Fig. 3.27.1(a).
2. It is 5 bits long, so we need five unique states to generate pulse train so, any mod-5 counter can be used.
3. For simplicity, we use a ripple counter.
4. It goes through states 0, 1, 2, 3, 4, 0 ... States 5, 6, 7 are invalid, so the corresponding outputs are don't cares.
5. The K-map for the output 1 in terms of the outputs of the flip-flops, its minimization, and the minimal expression obtained from it are shown in Fig. 3.27.1.
6. The logic diagram (using a mod 5 ripple counter) based on that minimal expression for  $f$  is shown in Fig. 3.27.2.
7. While at state 0, it output a 1, i.e., the first bit of the sequence.
8. While at state 1, it outputs a 0; i.e., the second bit of the sequence, and so on.

$Q_3$	$Q_2$	$Q_1$	Output ( $f$ )	States
0	0	0	1	0
0	0	1	0	1
0	1	0	1	2
0	1	1	1	3
1	0	0	0	4
1	0	1	x	5
1	1	0	x	6
1	1	1	x	7

(a) Truth table.

	$Q_2 Q_1$	00	01	11	10
$Q_3$	0	0	1	1	1
	1	4	x	5	x
				7	6

Output,  $f = Q_2 + \bar{Q}_3 \bar{Q}_1$   
 (b) K-map

Fig. 3.27.1. Pulse train generator.

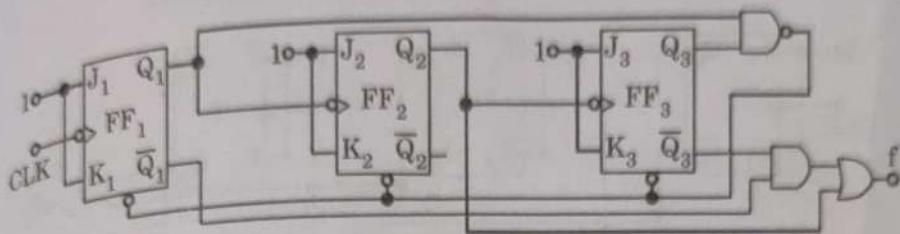


Fig. 3.27.2. Logic diagram of the pulse train generator.

Que 3.28. Design a pulse generator using indirect logic to produce the following waveforms.

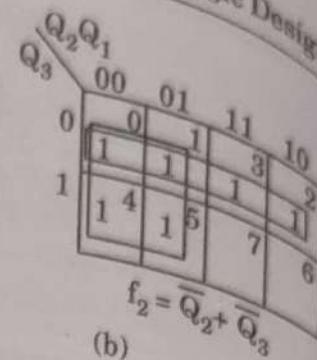
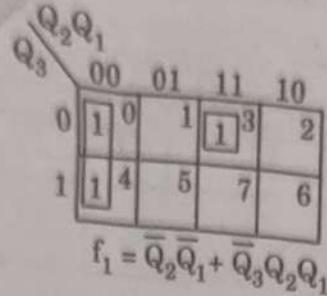
0	1	0	0	1	1	0	0	0
0	1	1	1	1	1	1	0	0

**Answer**

1. The pulse trains to be generated written vertically under column heading  $f_1$  and  $f_2$  in the truth table of Fig. 3.28.1(a) are : (a) 100 11000 and (b) 11111100.
2. These are both eight bits long.
3. So we need eight unique states to generate those two pulse trains.
4. Therefore, a mod-8, i.e., a 3-bit ripple counter can be used.
5. Let  $f_1$  and  $f_2$  be the outputs of the combinational circuits.
6. The state assignment is shown in the truth table.
7. The K-map for outputs  $f_1$  and  $f_2$  in terms of the outputs of the flip-flops, their minimization and the minimal expressions obtained from them are shown in Fig. 3.28.1(b).
8. The logic diagram (using a mod-8 ripple counter) based on those minimal expressions for  $f_1$  and  $f_2$  is shown in Fig. 3.28.2.

States	$Q_3$	$Q_2$	$Q_1$	$f_1$	$f_2$
0	0	0	0	1	1
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	1
4	1	0	0	1	1
5	1	0	1	0	1
6	1	1	0	0	0
7	1	1	1	0	0

(a)



(b)

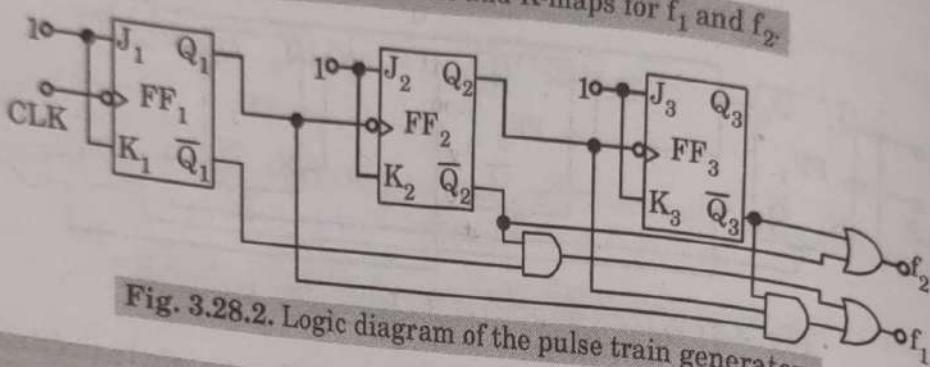
Fig. 3.28.1. Truth table and K-maps for  $f_1$  and  $f_2$ .

Fig. 3.28.2. Logic diagram of the pulse train generator.

**PART-10***Pseudo Random Binary Sequence (PRBS) Generator, Clock Generation.***Questions-Answers****Long Answer Type and Medium Answer Type Questions****Que 3.29.**

Write a short note on PSBR generator.

**Answer**

- Another important application of shift register is a Pseudo-Random Binary Sequence (PSBR) generator. Here, suitable feedback is used to generate pseudo-random sequence.
- The term random here means that the outputs do not cycle through a normal binary count sequence.
- The term pseudo here refers to the fact that the sequence is not truly random because it does cycle through all possible combinations once every  $2^n - 1$  clock cycles, where  $n$  represents the number of shift register stages (number of flip-flops).

**Que 3.30.** Explain clock generator.

**Answer**

1. The clock generator is a circuit that produces a timing signal for synchronization of the circuit's operation.
2. Examples of clock generators used in microprocessor systems include 8284 and 82284. 8284 generates the system clock for the 8086 and 8088 processors.
3. It requires a crystal or a TTL signal source for producing clock waveforms. It provides local READY and MULTIBUS READY synchronization.
4. 82284 is a clock generator/driver that provides clock signals for the 80286 processor and support components.
5. It also contains logic to supply READY to the CPU from either asynchronous or synchronous sources and synchronous RESET from an asynchronous input with hysteresis.
6. The 82284 is packaged in 18-pin DIP and contains a crystal-controlled oscillator, an MOS clock generator, a peripheral clock generator, multibus ready synchronization logic and system reset generation logic.

**VERY IMPORTANT QUESTIONS**

*Following questions are very important. These questions may be asked in your SESSIONALS as well as UNIVERSITY EXAMINATION.*

**Q.1. Design the clocked sequential circuit for the following state diagram using JK flip-flop.**

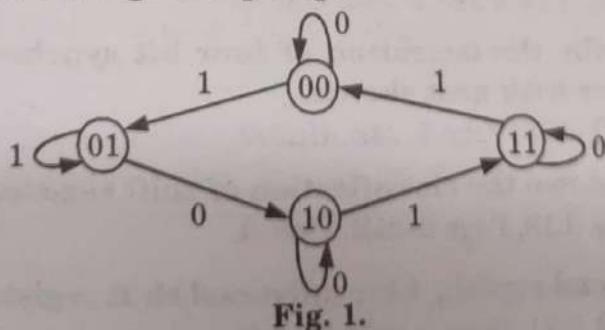


Fig. 1.

**Ans:** Refer Q. 3.7, Page 3-8B, Unit-3.

**Q.2. Derive the state table and state diagram for the sequential circuit is shown in Fig. 2.**

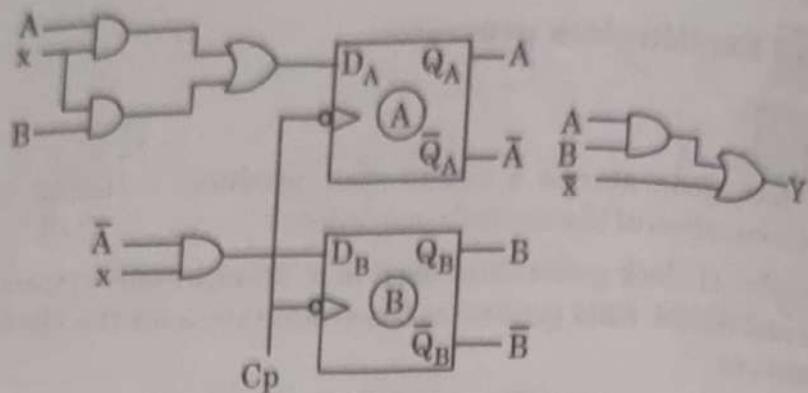


Fig. 2.

**Ans.** Refer Q. 3.10, Page 3-14B, Unit-3.

**Q. 3.** Draw the reduced state table and reduced state diagram for the state table given in Fig. 3.

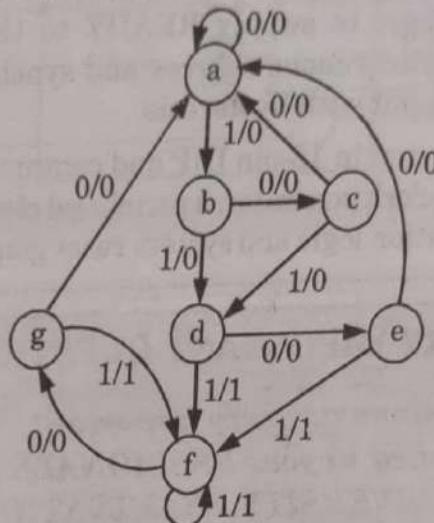


Fig. 3.

**Ans.** Refer Q. 3.9, Page 3-11B, Unit-3.

**Q. 4.** Design a ripple decade counter using JK flip-flop.

**Ans.** Refer Q. 3.14, Page 3-18B, Unit-3.

**Q. 5.** Describe the operation of four bit synchronous binary counter with neat sketch.

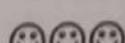
**Ans.** Refer Q. 3.18, Page 3-24B, Unit-3.

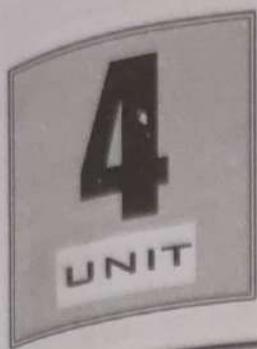
**Q. 6.** Write down the classification of shift registers.

**Ans.** Refer Q. 3.19, Page 3-25B, Unit-3.

**Q. 7.** Draw and explain 4-bit universal shft register.

**Ans.** Refer Q. 3.21, Page 3-28B, Unit-3.





## Logic Families and Semiconductor Memories

### CONTENTS

- |          |  |                |
|----------|--|----------------|
| Part-1 : | TTL NAND Gate, Specifications .....                    | 4-2B to 4-3B   |
| Part-2 : | Noise Margin .....                                     | 4-3B to 4-4B   |
| Part-3 : | Propagation Delay .....                                | 4-4B to 4-5B   |
| Part-4 : | Fan-in, Fan-out .....                                  | 4-5B to 4-6B   |
| Part-5 : | Tristate TTL .....                                     | 4-6B to 4-9B   |
| Part-6 : | ECL .....  | 4-9B to 4-11B  |
| Part-7 : | CMOS Families and .....                                | 4-11B to 4-17B |
| Part-8 : | Their Interfacing<br>Memory Elements, Concept of ..... | 4-18B to 4-20B |
| Part-9 : | Programmable Logic Devices<br>Like FPGA .....          | 4-20B to 4-21B |
|          | Logic Implementation using .....                       | 4-20B to 4-31B |
|          | Programmable Devices                                   |                |

**PART-1****TTL NAND Gate, Specifications.****Questions-Answers****Long Answer Type and Medium Answer Type Questions**

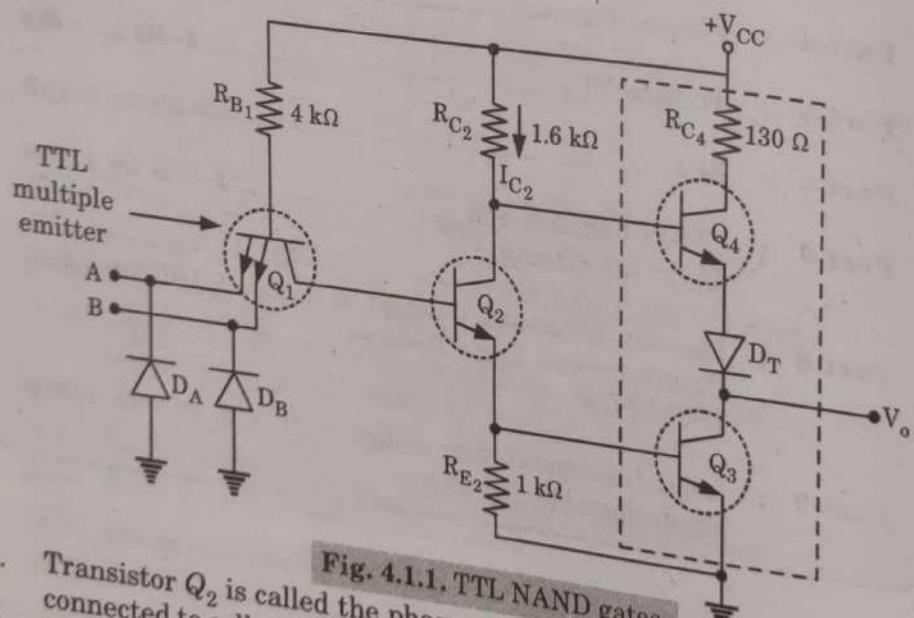
**Que 4.1.** Describe the construction and operation of TTL NAND gate.

**OR**  
Draw and explain the operation of a TTL NAND gate.

**AKTU 2017-18, Marks 07**

**Answer**

1. The circuit of the two-input TTL NAND gate is shown in Fig. 4.1.1. The input transistor,  $Q_1$  is a multiple emitter transistor.



**Fig. 4.1.1. TTL NAND gates.**

2. Transistor  $Q_2$  is called the phase splitter. Emitter of transistor,  $Q_4$  is connected to collector of transistor,  $Q_3$  through diode  $D_T$ .
3. Transistors  $Q_3$  and  $Q_4$  form a totem-pole arrangement. Diodes,  $D_A$  and  $D_B$  protect transistor,  $Q_1$  from being damaged by the negative spikes of voltages at the inputs.
4. When negative spikes appear at the input terminals, the diodes conduct and bypass the spikes to ground.

5. Diode  $D_T$  ensures that transistors,  $Q_3$  and  $Q_4$  do not conduct simultaneously. Transistor,  $Q_3$  acts as an emitter follower.

**Operation :**

1. A LOW voltage at either emitter  $E_1$  or emitter  $E_2$  forward-biases the corresponding diode  $D_1$  or  $D_2$  and reverse-biases diode  $D_3$  which is a base-collector junction of transistor  $Q_1$ . There is no flow of current from base to collector of transistor  $Q_1$ .
2. A LOW voltage on both emitters of transistor  $Q_1$  does the same action.
3. A HIGH voltage on both emitters reverse-biases both input diodes  $D_1$  and  $D_2$  and forward bias  $D_3$ . The current flows from base to collector of transistor  $Q_1$ .

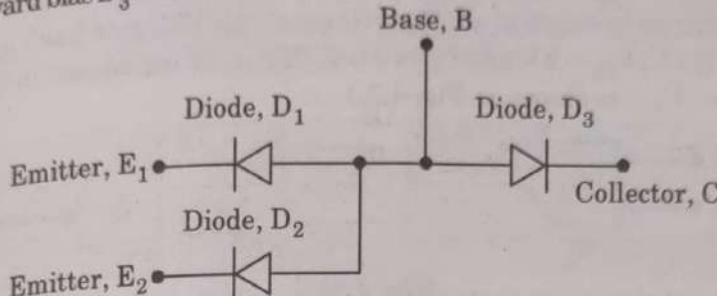


Fig. 4.1.2. Diode equivalent of TTL multiple emitters.

Table 4.1.1. Operation of TTL NAND gate

Inputs		Transistors				Output	
A	B	$Q_1$	$Q_1$	$Q_2$	$Q_3$	$Q_4$	$V_o$
		Emitter junction, A	Emitter junction, B				
0	0	Forward bias (ON)	Forward bias (ON)	OFF	OFF	ON	1
0	1	Forward bias (ON)	Reverse bias (OFF)	OFF	OFF	ON	1
1	0	Reverse bias (OFF)	Forward bias (ON)	OFF	OFF	ON	1
1	1	Reverse bias (OFF)	Reverse bias (OFF)	ON	ON	OFF	0

**PART-2***Noise Margin.***Questions-Answers****Long Answer Type and Medium Answer Type Questions**

**Que 4.2.** What do you understand by noise margin of logic circuit? Explain with an example.

**Answer**

1. The noise immunity of a logic circuit refers to the circuit's ability to tolerate noise voltages at its inputs. A quantitative measure of noise immunity is called noise margin.

2. Noise margin represents the maximum noise signal that can be added to the input signal of a digital circuit without causing an undesirable change in the circuit output.
3. Noise margin can be HIGH state noise margin or LOW state noise margin. HIGH state noise margin ( $NM_H$ ) is,  $V_{NH} = V_{OH} - V_{IH}$   
LOW state noise margin ( $NM_L$ ) is,  $V_{NL} = V_{IL} - V_{OL}$
4. High state noise margin is the difference between the lowest possible high output and the minimum input voltage required for a HIGH. Low state noise margin is the difference between the largest possible LOW output and the maximum input voltage for a LOW.
5. Consider an example of a TTL AND gate. The TTL gate has  $V_{OH} = 2.4\text{ V}$ ,  $V_{OL} = 0.4\text{ V}$ ,  $V_{IH} = 2\text{ V}$  and  $V_{IL} = 0.8\text{ V}$ . The noise introduced in the signal ( $V_{NH}$  or  $V_{NL}$ ) is shown in Fig. 4.2.1.

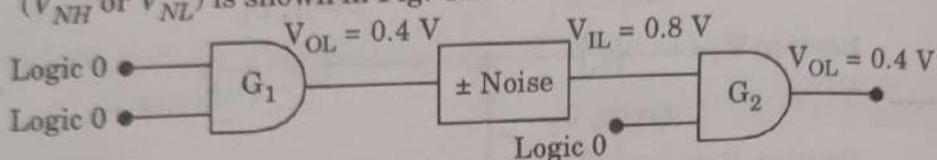


Fig. 4.2.1.

6. Let the inputs of gate,  $G_1$  cause output as logic 0. This output acts as input for gate,  $G_2$ . Due to noise, actual input given to gate,  $G_2$  is

$$V_{NL} = V_{IL} - V_{OL} \quad \dots(4.2.1)$$

7. Let the inputs of gate,  $G_1$  cause output as logic 1 in Fig. 4.2.2. This output acts as input for gate,  $G_2$ . Due to noise, actual input given to gate,  $G_2$  is

$$V_{NH} = V_{OH} - V_{IH} \quad \dots(4.2.2)$$

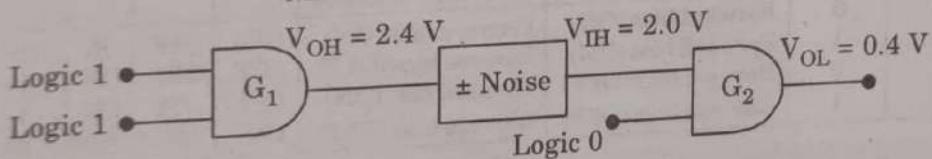


Fig. 4.2.2.

$V_{IH}$  in eq. (4.2.2) that acts as input to gate  $G_2$ . Minimum high level noise level is

$$V_{NH} = 2.4\text{ V} - 2.0\text{ V} = 0.4\text{ V}$$

### PART-3

Propagation Delay.

#### Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 4.3. Write a short note on propagation delay.

**Answer**

1. Propagation delay is defined as the time interval between changes in a defined logic level input and reflection of its effect at the output logic level.
2. The propagation delay for an integrated circuit (IC) logic gate may differ for each of the inputs. If all other factors are held constant, the average propagation delay in a logic gate IC increases as the complexity of the internal circuitry increases.

**PART-4***Fan-in, Fan-out.***Questions-Answers****Long Answer Type and Medium Answer Type Questions**

**Que 4.4.** Describe the fan-out and fan-in condition of the digital logic gate.

**Answer****Fan-out :**

1. The fan-out of a logic gate is defined as the maximum number of standard load that the output of the gate can drive without impairing its normal operation. Fan-out is also called the loading factor.
2. HIGH state fan-out is the fan-out of the gate when its output is logic 1. LOW state fan-out is the fan-out of the gate when its output is logic 0. The smaller of these two numbers is taken as the actual fan-out.
3. High state fan-out is given by

$$\text{HIGH state fan-out} = \frac{I_{OH}}{I_{IH}}$$

where,  $I_{OH}$  is the maximum current that the driver gate can source when it is in a 1 state.  $I_{IH}$  is the current drawn by each driven gate from the driver gate.

4. Similarly, low state fan-out is given by

$$\text{LOW state fan-out} = \frac{I_{OL}}{I_{IL}}$$

where,  $I_{OL}$  is the maximum current that the driver gate can sink when its output is a logic 0.  $I_{IL}$  is the current drawn from each driven gate by the driver gate.

5. The fan-out of a logic family can be calculated as

$$\text{Fan-out} = \text{minimum of } \left\{ \frac{I_{OH}}{I_{IH}}, \frac{I_{OL}}{I_{IL}} \right\}$$

**Fan-in :**

1. The fan-in of a digital logic gate refers to the number of inputs. For example, an inverter has a fan-in of 1, a 2-input NOR gate has a fan-in of 2, a 4-input NAND gate has a fan-in of 4 and so on.
2. A logic designer has to select the fan-in of the gate to accommodate the number of inputs.
3. At the hardware level, however, the fan-in provides information about the intrinsic speed of the gate itself.
4. In general, the propagation delay increases with the fan-in. This means that 2-input NAND gate is faster than the 4-input NAND if both are from same logic family.

**PART-5***Tristate TTL.***Questions-Answers****Long Answer Type and Medium Answer Type Questions**

**Que 4.5.** Describe the construction and operation of TTL inverter gate (NOT gate).

**Answer**

1. Fig. 4.5.1 shows a standard TTL circuit for an inverter. Transistor,  $Q_1$  is the input coupling transistor, and  $D_1$  is the input clamp diode. Transistor,  $Q_2$  is called a phase splitter, and the combination of  $Q_3$  and  $Q_4$  forms the output circuit often referred to as a totem-pole arrangement.

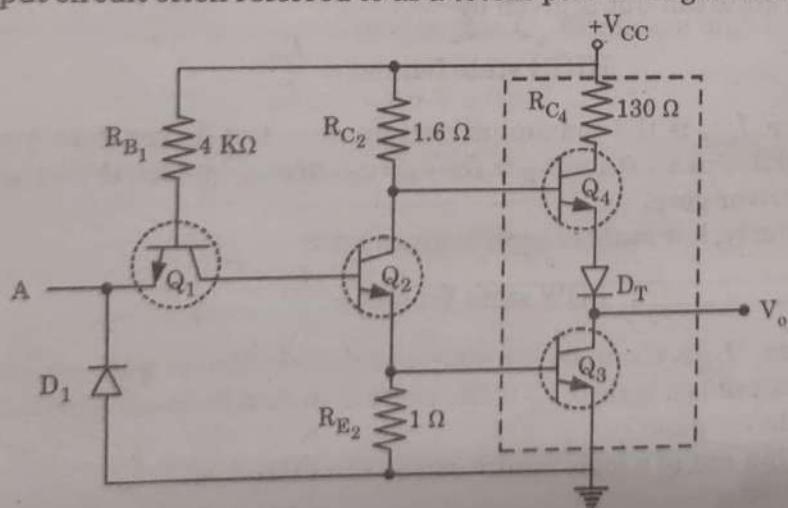


Fig. 4.5.1. TTL inverter gates.

2. When the input is HIGH, the base-emitter junction of transistor,  $Q_1$  is reverse-biased and the base-collector junction is forward-biased.
3. This condition permits current to flow through  $R_{B_1}$  and the base-collector junction of transistor,  $Q_1$  into the base of transistor,  $Q_2$ .
4. It drives transistor,  $Q_2$  into saturation. As a result, transistor,  $Q_3$  is turned ON due to ON state of transistor,  $Q_2$ , and its collector voltage, which is the output, is near to ground potential.
5. Therefore, a LOW output is produced for a HIGH input. At the same time, the collector of transistor,  $Q_2$  is at a sufficiently LOW voltage level to keep transistor,  $Q_4$  OFF.
6. When the input is LOW, the base-emitter junction of transistor,  $Q_1$  is forward-biased, and the base-collector junction is reverse-biased.
7. The current flows through resistor,  $R_{B_1}$  and the base-emitter junction of transistor,  $Q_1$ , to the LOW input. A LOW provides a path to ground for the current.
8. No current flows into the base of transistor,  $Q_2$ , so it is OFF. The collector of transistor,  $Q_2$  is HIGH, thus turning transistor,  $Q_4$  ON. A saturated transistor,  $Q_4$  provides a low-resistance path from  $V_{CC}$  to the output.
9. Therefore, a HIGH on the output is produced for a LOW on the input. At the same time, the emitter of transistor,  $Q_2$  is at ground potential, keeping transistor,  $Q_3$  OFF.

Table 4.5.1. Operation of TTL inverter.

Inputs	Transistors				Output $V_o$
	$Q_1$	$Q_2$	$Q_3$	$Q_4$	
A	Emitter junction, A				
Logic 0 Logic 1	Forward bias (ON) Reverse bias (OFF)	OFF ON	OFF ON	ON OFF	Logic 1 Logic 0

**Que 4.6.** Describe the construction and operation of TTL NOR gate.

**Answer**

1. The circuit of the two-input TTL NOR gate is shown in Fig. 4.6.1. Two input transistors  $Q_A$  and  $Q_B$  are emitter transistors.
2. Transistor  $Q_1$  and  $Q_2$  are called the phase splitters. Emitter of transistor  $Q_4$  is connected to collector of transistor  $Q_3$  through diode  $D_T$ .
3. Transistors  $Q_3$  and  $Q_4$  form a totem-pole arrangement. Diodes  $D_A$  and  $D_B$  protect transistor  $Q_A$  and  $Q_B$  from being damaged by the negative spikes of voltages at the inputs.
4. When negative spikes appear at the input terminals, the diodes conduct and bypass the spikes to ground.
5. Diode  $D_T$  ensures that transistors  $Q_3$  and  $Q_4$  do not conduct simultaneously. Transistor  $Q_3$  acts as an emitter follower.

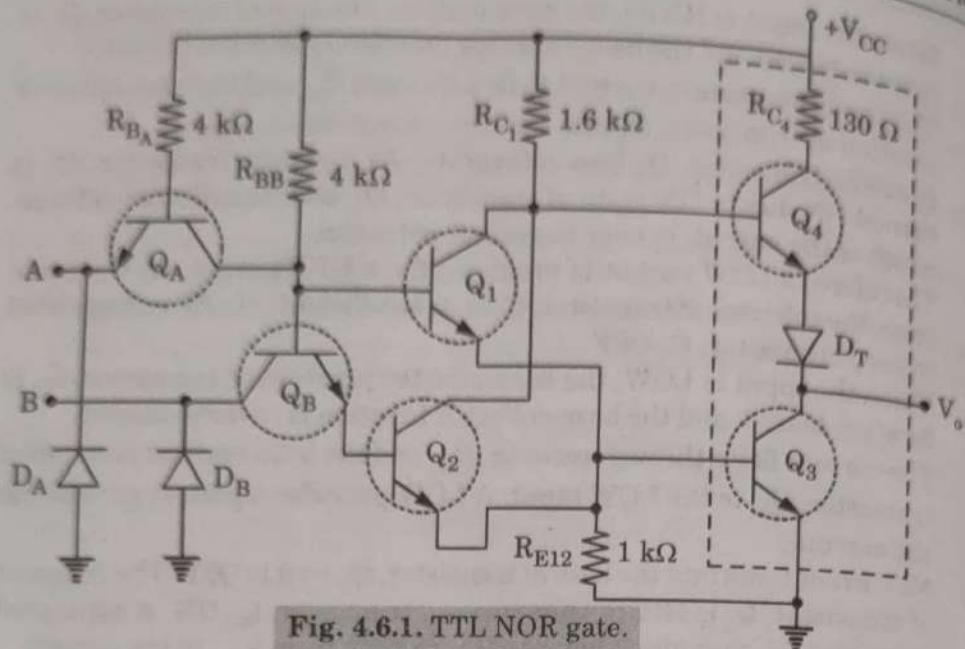


Fig. 4.6.1. TTL NOR gate.

**Operation :**

Table 4.6.1. Operation of TTL NOR gate.

Inputs		Transistors							Output
A	B	Q <sub>A</sub>	Q <sub>B</sub>	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	Q <sub>4</sub>	V <sub>o</sub>	
		Emitter junction, A							Emitter junction, B
0	0	Forward bias (ON)	Forward bias (ON)	OFF	OFF	OFF	ON	1	
0	1	Forward bias (ON)	Reverse bias (OFF)	OFF	ON	ON	OFF	0	
1	0	Reverse bias (OFF)	Forward bias (ON)	ON	OFF	ON	OFF	0	
1	1	Reverse bias (OFF)	Reverse bias (OFF)	ON	ON	ON	OFF	0	

**Que 4.7.** State various TTL parameters in brief.**Answer****i. Current sinking :**

1. A TTL circuit acts as a current sink in LOW state, as it receives current from the input of the gate by which it is driving.
2. Transistor is the current-sinking transistor or the pull-down transistor, because it brings the output voltage down to its LOW state.

**ii. Current sourcing :**

1. A TTL circuit acts as a current source in the HIGH state, as it supplies current to the gate by which it is driving.
2. Transistor is the current-sourcing transistor or the pull-up transistor, because it pulls up the output voltage to its HIGH state.

**iii. Floating inputs :**

- i. When a TTL input is HIGH (ideally + 5 V), the emitter current is approximately zero. When a TTL input is floating no emitter current is possible because of the open circuit.

- ii. Therefore, a floating TTL input is equivalent to a HIGH output. Because of this, unused TTL inputs are left unconnected : an open input allows the rest of the gate to function properly.
- iv. **TTL loading and fan-out :**
- 1. The TTL output has a limit,  $I_{OL}$  that gives the maximum current it can sink in LOW state and a limit,  $I_{OH}$ , gives the maximum current it can source in HIGH state.
- 2. To determine the fan-out, the drive capabilities of the output, i.e.,  $I_{OL}$  and  $I_{OH}$  and the current requirements of each input, i.e.,  $I_{IL}$  and  $I_{IH}$  are known.
- 3. So, HIGH and LOW state fan-outs are given by :

$$\text{HIGH state fan-out} = \frac{I_{OH}}{I_{IH}}$$

$$\text{LOW state fan-out} = \frac{I_{OL}}{I_{IL}}$$

- 4. The actual fan-out capability is equal to the smaller of the above two fan-out values and is given by

$$\text{Actual fan-out capability} = \min \left\{ \frac{I_{OH}}{I_{IH}}, \frac{I_{OL}}{I_{IL}} \right\}$$

v. **Unit load :**

Unit load means the current drawn or sourced back by similar gates.

**Example :** For 7400,

One unit load is  $40 \mu\text{A}$  in HIGH state that is known as  $I_{IH}$ .

One unit load is  $1.6 \text{ mA}$  in LOW state that is known as  $I_{IL}$ .

### PART-6

#### ECL.

#### Questions-Answers

#### Long Answer Type and Medium Answer Type Questions

**Que 4.8.** Explain the basic circuit and operation of emitter coupled logic (ECL). What are the functions of emitter follower ?

#### Answer

##### Basic ECL circuit :

- 1. The basic circuit for emitter-coupled logic is a differential amplifier configuration as shown in Fig. 4.8.1.
- 2. The  $V_{EE}$  supply produces a fixed current  $I_E$ , which remains around  $3 \text{ mA}$  during normal operation. This current is allowed to flow through either transistor  $Q_1$  or transistor  $Q_2$ , depending on the voltage level at  $V_{IN}$ .

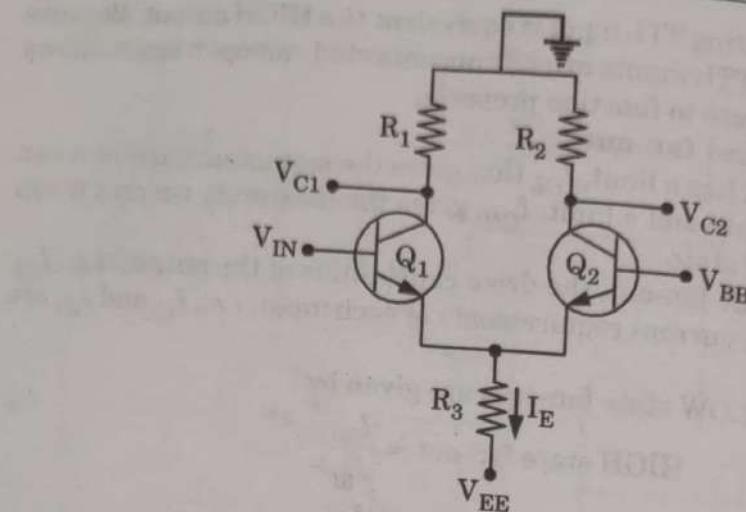


Fig. 4.8.1.

3. In other words, this current switches between collector of  $Q_1$  transistor and collector of  $Q_2$  transistor as  $V_{IN}$  switches between its two logic levels of  $-1.7\text{ V}$  (logical 0 for ECL) and  $-0.8\text{ V}$  (logical 1 for ECL).
4. Table 4.8.1 shows the resulting output voltages for these two conditions at  $V_{IN}$ .

Table 4.8.1. Operating states of ECL

$V_{IN}$ Voltage level	Binary logic	Outputs		Remarks
		$V_{C1}$	$V_{C2}$	
$-1.7\text{ V}$	Logic 0	0 V	$-0.9\text{ V}$	$Q_2$ conducts
$-0.8\text{ V}$	Logic 1	$-0.9\text{ V}$	0 V	$Q_1$ conducts

5. Two important points are noted :
  - i.  $V_{C1}$  and  $V_{C2}$  are the complements of each other, and
  - ii. The output voltage levels are not same as the input logic levels.
6. The emitter followers perform two functions :
  - i. Emitter followers subtract approximately  $0.8\text{ V}$  from  $V_{C1}$  and  $V_{C2}$  to shift the output levels to the correct ECL logic levels.
  - ii. Emitter followers provide very low output impedance (typically  $7\text{ }\Omega$ ), which provides for large fan-out and fast charging of load capacitance.

**Que 4.9.** Describe the construction and operation of ECL OR/NOR gate.

**Answer**

1. A two-input ECL OR/NOR gate is shown in Fig. 4.9.1.
2. It has two outputs which are complements of each other. Transistors  $Q_2$  and  $Q_1$  form a differential amplifier. Transistors  $Q_4$  and  $Q_5$  are in parallel.
3. Transistors  $Q_4$  and  $Q_5$  are emitter followers whose emitter voltages are the same as the base voltages (less than  $0.8\text{ V}$  base to emitter drops).

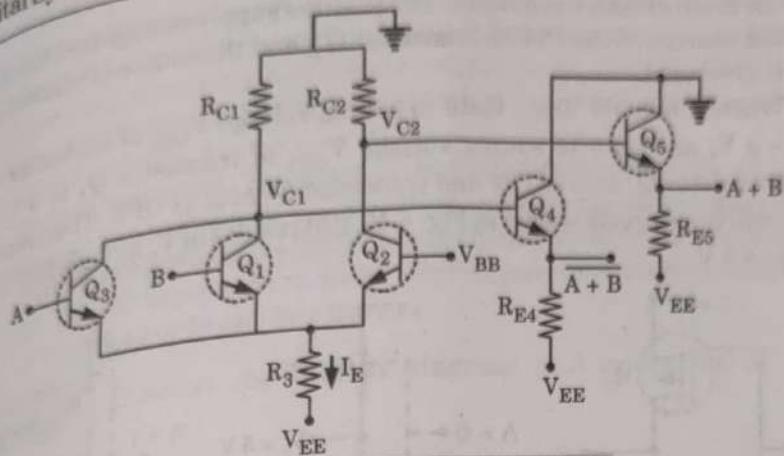


Fig. 4.9.1. ECL NOR and OR gates.

4. Inputs are applied to transistors  $Q_1$  and  $Q_3$ , and transistor  $Q_2$  is supplied with constant  $-1.3\text{ V}$ .

Table 4.9.1. Operation of ECL OR/NOR gate.

Inputs		Transistors					Output	
A	B	$Q_3$	$Q_1$	$Q_2$	$Q_4$	$Q_5$	$A + B$	$A + B$
0	0	OFF	OFF	ON	ON	OFF	0	1
0	1	OFF	ON	OFF	OFF	ON	1	0
1	0	ON	OFF	OFF	OFF	ON	1	0
1	1	ON	ON	OFF	OFF	ON	1	0

**PART-7***CMOS Families and their Interfacing.***Questions-Answers****Long Answer Type and Medium Answer Type Questions**

Que 4.10. Describe the circuit and performance of CMOS inverter and state the characteristics of CMOS. AKTU 2018-19, Marks 3.5

**Answer****CMOS inverter :**

1. It consists of an NMOS transistor  $Q_1$  and a PMOS transistor  $Q_2$ . The input is connected to the gates of both the devices and the output is at

the drain of both the devices. The positive supply voltage is connected to the sources of the PMOS transistor  $Q_2$ , and the source of transistor  $Q_1$  is grounded.

2. When  $A$  is LOW (0 V). Gate to source voltage  $V_{GS2}$  of transistor  $Q_2$  is  $-5\text{ V}$ , and gate to source voltage  $V_{GS1}$  of transistor  $Q_1$  is  $0\text{ V}$ . So, transistor  $Q_2$  acts as ON and transistor  $Q_1$  acts as OFF. Therefore, the switching circuit shown in Fig. 4.10.1(b) results in  $V_o$  as logic HIGH that is  $+5\text{ V}$ .

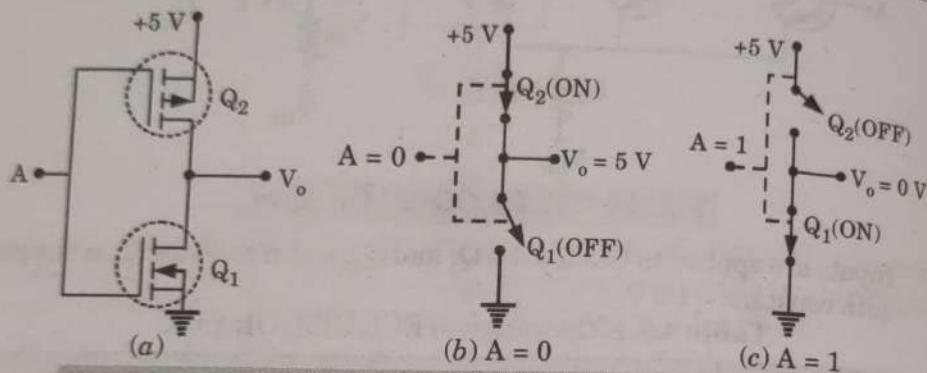


Fig. 4.10.1. (a) CMOS as inverter, (b) and (c) equivalent circuit.

3. When  $A$  is HIGH ( $+5\text{ V}$ ), gate to source voltage  $V_{GS2}$  of transistor  $Q_2$  is  $0\text{ V}$ , and gate to source voltage  $V_{GS1}$  of transistor  $Q_1$  is  $+5\text{ V}$ . So, transistor  $Q_2$  acts as OFF and transistor  $Q_1$  acts as ON. Therefore, the switching circuit shown in Fig. 5.13.1(c) results with  $V_o$  as logic LOW that is  $0\text{ V}$ .

Table 4.10.1. Operation of CMOS inverter.

Input, $A$	$p$ -channel MOSFET, $Q_1$	$n$ -Channel MOSFET, $Q_2$	Output, $V_o$
LOW ( $0\text{ V}$ )	ON	OFF	$+5\text{ V}$ (HIGH)
HIGH ( $5\text{ V}$ )	OFF	ON	$0\text{ V}$ (LOW)

Truth table :

$A$	$V_o$
0	1
1	0

#### Characteristics of CMOS :

- i. **Supply voltage :** The 4000 and 74C series can operate with  $V_{DD}$  values ranging from 3 to 15 V. The 74HC and 74HCT series can operate with  $V_{DD}$  values ranging from 2 to 6 V.
- ii. **Voltage levels :** When a CMOS output drives only a CMOS input and CMOS gate has an extremely high input resistance, the current drawn is almost zero and, therefore, the output voltage levels will be very close to zero for LOW state and  $V_{DD}$  for HIGH state.

iii. **Power dissipation**: When a CMOS circuit is in a static state, its power dissipation per gate is extremely small, but it increases with increase in operating frequency and supply voltage level. For DC, CMOS power dissipation is only 2.5 nW per gate when  $V_{DD} = 5$  V, and it increases to 10 nW per gate when  $V_{DD} = 10$  V.

iv. **Switching speed**: The speed of the CMOS gate increases with increase in  $V_{DD}$ . The increase in  $V_{DD}$  results in increase in power dissipation too. **Unused inputs**: The CMOS inputs should never be left disconnected. All CMOS inputs have to be tied either to a fixed voltage level (0 V or  $V_{DD}$ ) or to another input.

**Que 4.11.** Discuss the circuit diagram and operation of CMOS NAND gate.

**Answer**

- Fig. 4.11.1 shows a CMOS two-input NAND gate. Here, *p*-channel MOSFETs  $Q_1$  and  $Q_2$  are connected in parallel and *n*-channel MOSFETs  $Q_3$  and  $Q_4$  are connected in series.
- When  $A$  is LOW (0 V) and  $B$  is also LOW (0 V). *p*-channel MOSFET  $Q_1$  acts ON, *n*-channel MOSFET  $Q_3$  acts OFF, *p*-channel MOSFET  $Q_2$  acts ON and *n*-channel MOSFET  $Q_4$  acts OFF. Thus, the switching results  $V_o$  as logic HIGH i.e., +5 V.
- When  $A$  is LOW (0 V) and  $B$  is HIGH (5 V). *p*-channel MOSFET  $Q_1$  acts ON, *n*-channel MOSFET  $Q_3$  acts OFF, *p*-channel MOSFET  $Q_2$  acts OFF and *n*-channel MOSFET  $Q_4$  acts ON. Thus, the switching circuit results  $V_o$  as logic HIGH i.e., +5 V.
- When  $A$  is HIGH (+5 V) and  $B$  is LOW (0 V). *p*-channel MOSFET  $Q_1$  acts OFF, *n*-channel MOSFET  $Q_3$  acts ON, *p*-channel MOSFET  $Q_2$  acts ON and *n*-channel MOSFET  $Q_4$  acts OFF. Thus, the switching circuit results  $V_o$  as logic HIGH i.e., +5 V.
- When  $A$  is HIGH (+5 V) and  $B$  is also HIGH (+5 V). *p*-channel MOSFET  $Q_1$  acts OFF, *n*-channel MOSFET  $Q_3$  acts ON, *p*-channel MOSFET  $Q_2$  acts OFF and *n*-channel MOSFET  $Q_4$  acts ON. Thus, the switching circuit results  $V_o$  as logic LOW i.e., 0 V.

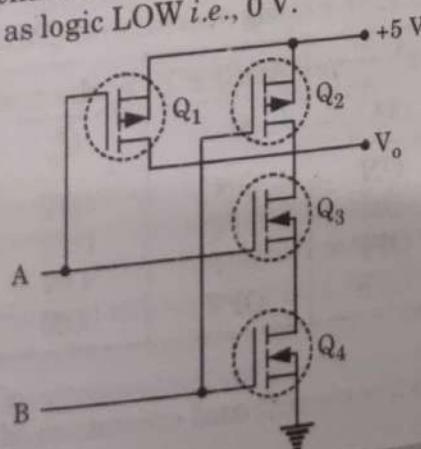


Fig. 4.11.1. CMOS as NAND gate.

Table 4.11.1. Switching operation of CMOS NAND gate

Inputs		<i>p</i> -channel MOSFET		<i>n</i> -channel MOSFET		Output
A	B	$Q_1$	$Q_2$	$Q_3$	$Q_4$	$V_o$
0	0	ON	ON	OFF	OFF	0
0	1	ON	OFF	OFF	ON	1
1	0	OFF	ON	ON	OFF	1
1	1	OFF	OFF	ON	ON	0

**Que 4.12.** Discuss the circuit diagram and operation of CMOS NOR gate.

**Answer**

- Fig. 4.12.1 shows a CMOS two-input NOR gate. Here, *p*-channel MOSFETs  $Q_1$  and  $Q_2$  are connected in series and *n*-channel MOSFETs  $Q_3$  and  $Q_4$  are connected in parallel.

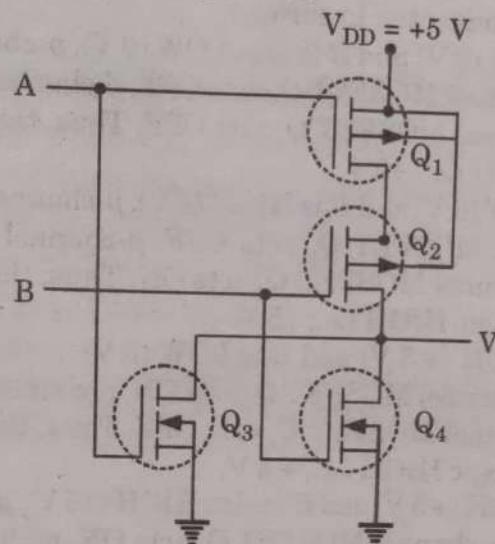


Fig. 4.12.1. CMOS as NOR.

Table 4.12.1. Switching operation of CMOS NOR gate

Inputs		<i>p</i> -channel MOSFET		<i>n</i> -channel MOSFET		Output
A	B	$Q_1$	$Q_2$	$Q_3$	$Q_4$	$V_o$
0	0	ON	ON	OFF	OFF	1
0	1	ON	OFF	OFF	ON	0
1	0	OFF	ON	ON	OFF	0
1	1	OFF	OFF	ON	ON	0

**Que 4.13.** Discuss the circuit and operation of CMOS transmission gate.

**Answer**

1. A transmission gate is simply a digitally controlled CMOS switch. When the switch is open (OFF), the impedance between its terminals is very large.
2. It is used to implement special logic functions. Since the CMOS gate can transmit signals in both directions, it is called a bilateral transmission gate or bilateral switch.
3. It is useful for digital and analog applications. The TTL and ECL gates are essentially unidirectional.
4. Fig. 4.13.1 shows the schematic diagram and logic symbols of a CMOS transmission gate. The *n*-channel MOS and *p*-channel MOS transistors are connected in parallel.
5. So, both polarities of input voltages can be switched. The control signal, *C* is connected to the *n*-channel MOSFET and its inverse is connected to the *p*-channel MOSFET.

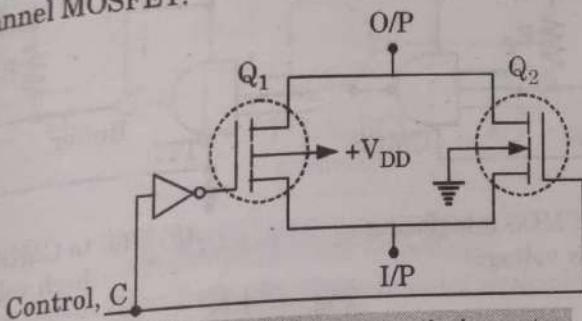


Fig. 4.13.1. CMOS transmission gate.

Table 4.13.1. Operation of CMOS transmission gate

Inputs		<i>p</i> -channel MOSFET		<i>n</i> -channel MOSFET		Action
<i>C</i>	<i>I/P</i>	$V_{G1}$	$Q_1$	$V_{G2}$	$Q_2$	
0	0	+ve	OFF	0 V	OFF	No transmission
0	1	0 V	OFF	-ve	OFF	No transmission
1	0	0 V	OFF	+ve	ON	Transmission
1	1	-ve	ON	0 V	OFF	Transmission

6. So, it can be concluded that when the control, *C* is HIGH, the circuit acts as a closed switch and allows the transmission of the signal from input to output.
7. When the control, *C* is LOW, the circuit acts as an open switch and blocks the transmission of the signal from input to output.
8. Since, the input and output terminals are interchangeable, the circuit can also transmit signals in the opposite direction. So, it acts as a bilateral switch.

**Que 4.14.** Explain TTL to CMOS interfacing and CMOS to TTL interfacing.

**Answer****TTL to CMOS :**

1. The MOS and CMOS gates are slower than the TTL gates, but consume less space. Hence, there is an advantage in using TTL and MOS devices in combination.
2. The input current values of CMOS are low as compared to the output current capabilities of any TTL series. Thus, TTL has no problem in meeting the CMOS input current requirements.
3. So, a level translator is used to raise the level of the output voltage of the TTL gate to an acceptable level for CMOS.
4. The presence of the pull-up resistor will cause the TTL output to rise to approximately + 5 V in the HIGH state, thereby providing an adequate CMOS input as shown in Fig. 4.14.1.

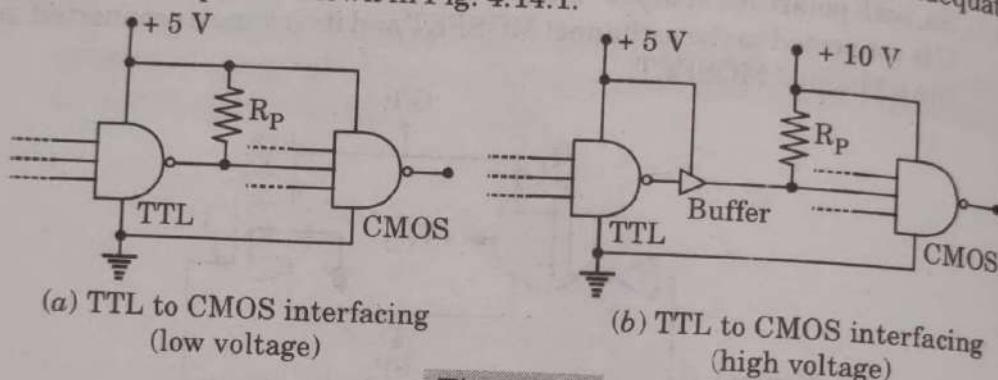


Fig. 4.14.1.

**CMOS to TTL :**

1. The CMOS output can supply enough voltage and current to satisfy the TTL input requirements in the HIGH state. Hence, no special consideration is required for the HIGH state.
2. But the TTL input current requirements at LOW state cannot be met directly.
3. Therefore, an interface circuit with a LOW input current requirement and a sufficiently high output current rating is required. The arrangement is shown in Fig. 4.14.2.
4. When a high voltage CMOS has to drive a TTL gate, a voltage level translator that converts the high voltage input to a + 5 V output is used between CMOS and TTL as shown in Fig. 4.14.2.

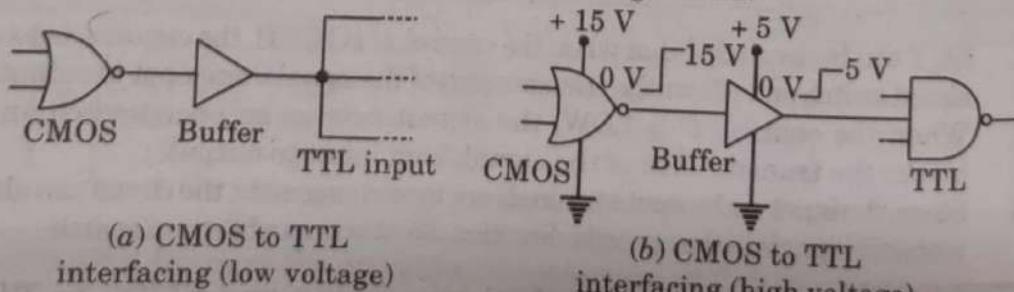


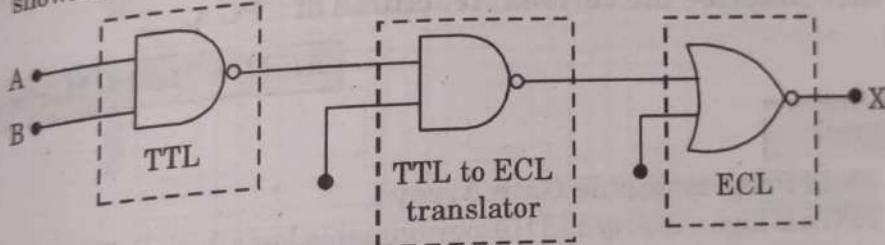
Fig. 4.14.2.

**Que 4.15.** Explain the interfacing of TTL to ECL and ECL to TTL.

**Answer**

**TTL to ECL :**

1. The TTL is the most widely used logic family, but its speed of operation is not very high.
2. The ECL is the fastest family. In some applications, the rate at which input data is to be handled may be much lower than the rate at which the output data is to be handled.
3. Therefore, it becomes necessary to interconnect the two different logic systems, such as TTL and ECL.
4. A TTL cannot interface directly with an ECL; it requires a translator as shown in Fig. 4.15.1.

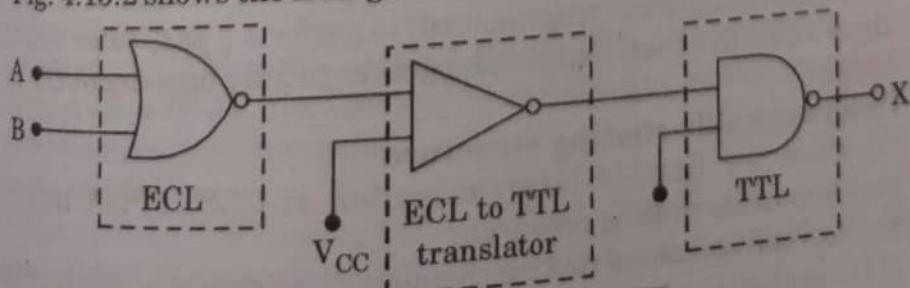


**Fig. 4.15.1. TTL driving ECL.**

5. One such application is in the time division multiplexing of  $n$  digital signals to form a single digital signal.
6. Although, the bit rate of each of the  $n$  signals may be handled using TTL, the bit rate of the composite signal is  $n$  times faster and may require ECL to process it.

**ECL to TTL :**

1. Sometimes, the input data is at a faster rate, but the output data is at a slower rate like in demultiplexers.
2. An ECL to TTL logic translator will be of use in such cases. It shows that the input logic levels of a translator are compatible with the output logic levels of ECL and the output logic levels of a translator are compatible with the input logic levels of a TTL.
3. Fig. 4.15.2 shows the ECL gate driving a TTL gate.



**Fig. 4.15.2. ECL driving TTL.**

**PART-B**

*Memory Elements, Concept of Programmable Logic Devices Like FPGA.*

**Questions-Answers****Long Answer Type and Medium Answer Type Questions**

**Que 4.16.** Discuss the concept of field programmable gate array (FPGA). Describe the various structures of FPGA.

AKTU 2018-19, Marks 3.5

**Answer****Field Programmable Gate Array :**

1. FPGA is high capacity PLD (programmable logic device). The gate array of FPGA has the ability to be programmed for a function by the user instead of the manufacturer of device.
2. FPGA consists of three configurable (programmable) logic modules (LMs) : configurable logic blocks (CLBs), input and output blocks and switching matrix for interconnection.
3. The CLB consists of a combinational logic array, data multiplexer (MUX) and flip-flops. The combinational array function is performed by look-up table (LUT).

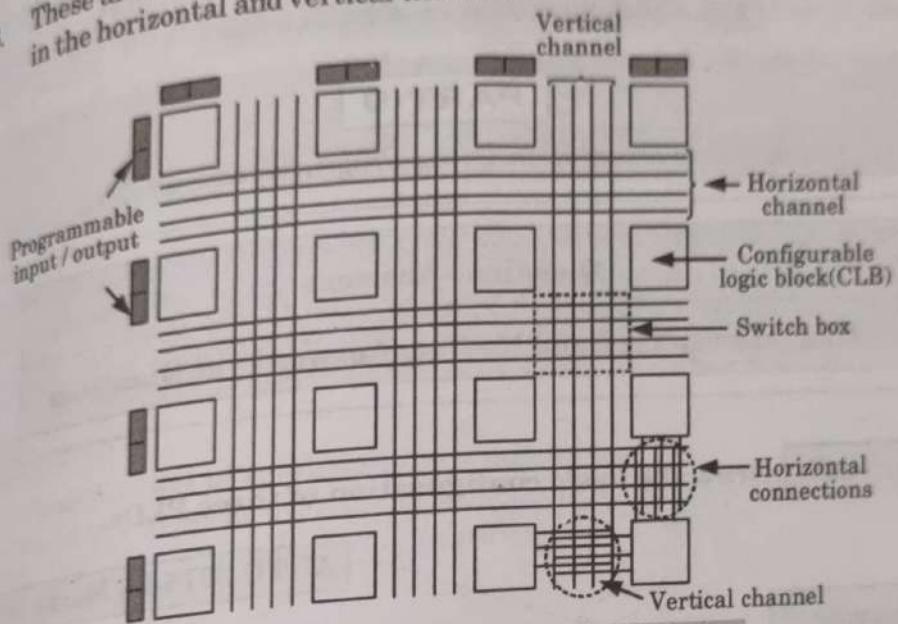
**Structures of FPGA :****i. Programmable logic structure :**

1. The programmable logic structure of FPGA consists of a two-dimensional array of CLBs.
2. Each CLB typically contains one or two flip-flops to allow implementation of sequential logic.
3. Large designs are partitioned and mapped to a number of CLBs with each CLB configured (programmed) to perform a particular function.
4. These CLBs are then connected together to fully implement the target design.

**ii. Programmable routing structure :**

1. To allow for flexible interconnection of CLB, FPGA has three programmable routing resources.
2. Vertical and horizontal routing channels which consist of different length of wires that can be connected together if needed.
3. These channels run vertically and horizontally between columns and rows of CLBs as shown in the Fig. 4.16.1.

4. Connection boxes, which are set of programmable links, can connect input and output pins of the CLBs to wires of the vertical or the horizontal routing channels.
5. Switch boxes are located at the intersection of the vertical and horizontal channels.
6. These are a set of programmable links that can connect wire segments in the horizontal and vertical channels.



**Fig. 4.16.1.** Programmable structure of field programmable logic array (FPGA).

### iii. Programmable input/output :

1. These are mainly buffers that can be configured either as input buffers or output buffers or input/output as shown in Fig. 4.16.1.
2. These allow the pins of the FPGA chip to function either as input pins or output pins or input/output pins.

### iv. Configurable logic blocks :

1. There are a number of CLBs in an FPGA organized as an array of rows and columns. The logic blocks are connected to the I/O blocks through common row / column programmable interconnects.
2. The common row / column interconnects are known as global interconnects.
3. A logic block consists of a number of LMs. The LMs are the basic logic elements in a FPGA. The LMs within a CLB are connected through local programmable interconnects.

### v. Logic module :

1. A logic module (LM) consists of a LUT, a D-type flip-flop and a MUX. Most of the FPGAs are based on 4-input LUT. Fig. 4.16.2 shows a block diagram of a LM with 4-input LUT.
2. Output of the LUT becomes the output of the LM either directly or through D-type flip-flop. Thus, the output can be configured for combinational or registered (i.e., through flip-flop).

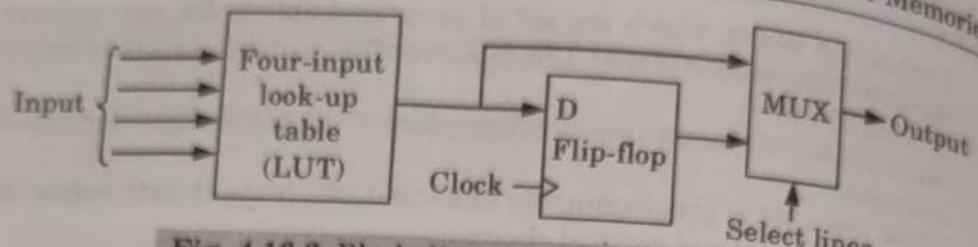


Fig. 4.16.2. Block diagram of logic module.

**PART-9***Logic Implementation Using Programmable Devices.***Questions-Answers****Long Answer Type and Medium Answer Type Questions**

**Que 4.17.** Draw the basic configuration of three PLDs.

**AKTU 2016-17, Marks 10**

**Answer**

1. The PROM is a combinational programmable logic device (PLD)-an integrated circuit with programmable gates divided into an AND array and an OR array to provide an AND-OR sum-of-product implementation. Fig. 4.17.1 shows the configuration of the three PLDs.

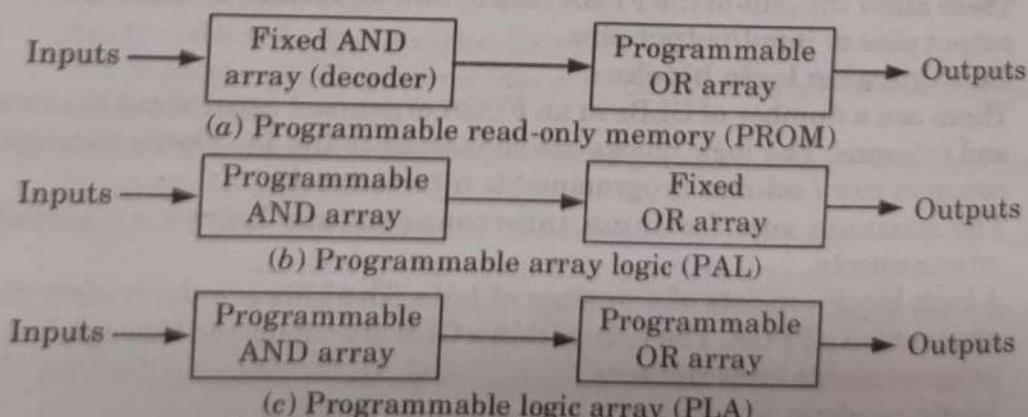


Fig. 4.17.1.

2. The PROM has a fixed AND array constructed as a decoder and a programmable OR array. The programmable OR gates implement the boolean functions in sum-of-minterms form.

3. The PAL has a programmable AND array and a fixed OR array. The AND gates are programmable to provide the product terms for the boolean functions which are logically summed in each OR gate.
4. The most flexible PLD is the PLA, in which both the AND and OR arrays can be programmed. The product terms in the AND array may be shared by any OR gate to provide the required sum of products implementation.

**Que 4.18.** What is the basic architecture of a PLA ? How is the capacity of a PLA specified ? How is it programmed ? Explain.  
OR

Write a short note on PLA.

AKTU 2017-18, Marks 3.5

OR

Write down the classification of semiconductor memories. Draw and explain the programmable logic array (PLA).

AKTU 2014-15, Marks 06

**Answer**

**Classification of semiconductor memories :**

Fig. 4.18.1 shows an overview of semiconductor memory types.

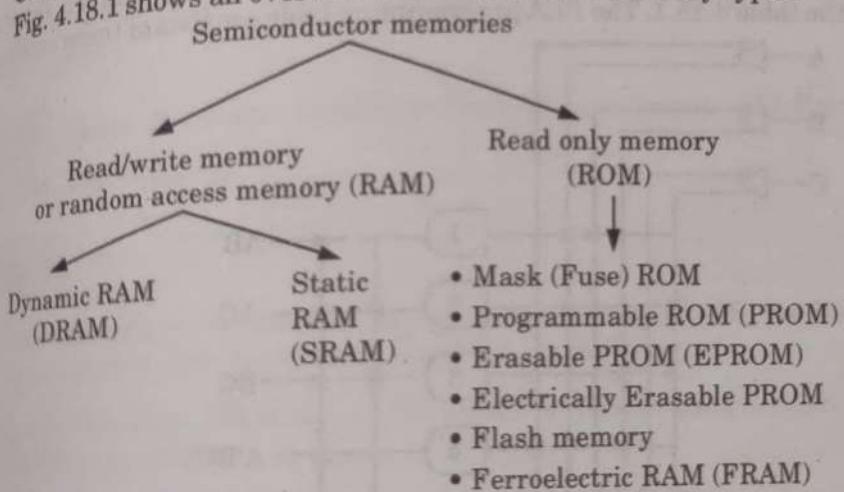


Fig. 4.18.1. Classification of semiconductor memories.

**Programmable logic array (PLA) :**

1. PLAs are used to map irregular combinational function onto regular structures. The PLA provides the designer with a systematic and regular way of implementing output functions of  $n$  variable in sum of product form.
2. PLA is one of the regular macro used in the implementation of FSM (finite state machine). PLA functions may be significantly changed without requiring major changes of either the design or layout. It is more compact in nature. Any of the logical function can be expressed in terms of SOP or POS.

3. PLA can be implemented in several forms, i.e., NOR-NOR, NAND-NAND, NAND-NOR.
4. The structure of PLA is shown in Fig. 4.18.2, and its internal logic with three inputs and two outputs is shown in Fig. 4.18.3.
5. The particular boolean functions implemented in the PLA of Fig. 4.18.3, are

$$F_1 = A\bar{B} + AC + \bar{A}\bar{B}\bar{C}$$

$$F_2 = \overline{(AC + BC)}$$

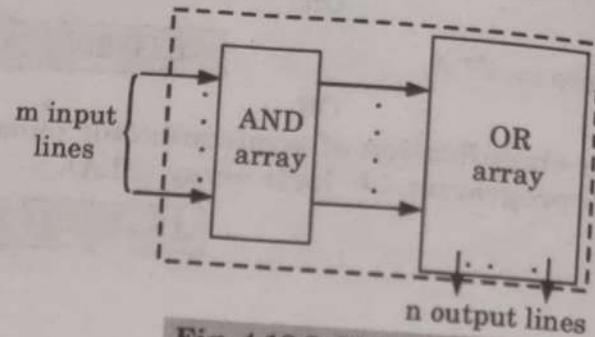


Fig. 4.18.2. PLA structure.

6. The programming table that specifies the PLA of Fig. 4.18.3 is listed in the table 4.18.1. The PLA programming table consists of three sections.

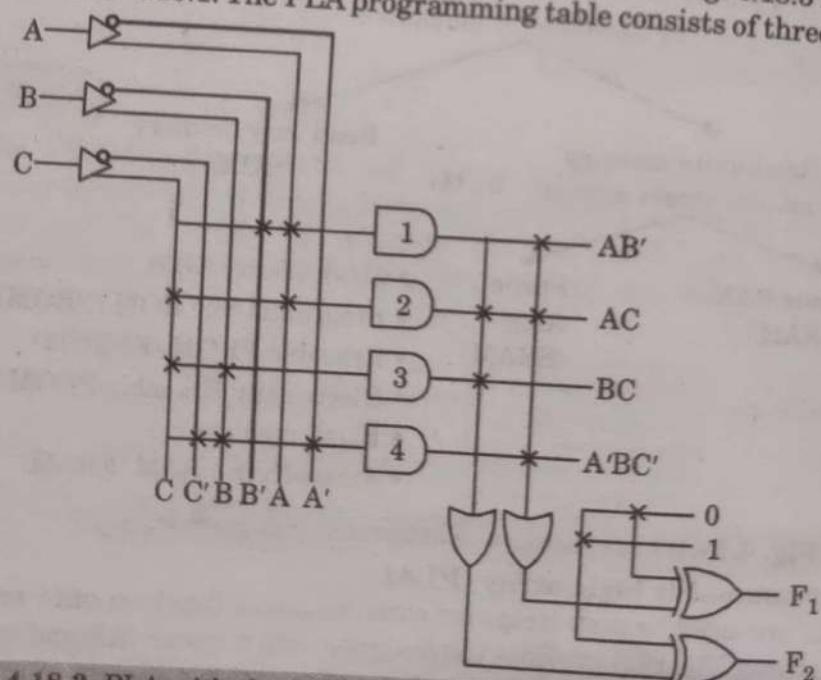


Fig. 4.18.3. PLA with three inputs, four product terms, and two outputs.

7. The first section lists the product term numerically. The second section specifies the required paths between input and AND gates. The third section specifies the path between the AND and OR gates.
8. For each output variable, we may have a T (true) or C (complement) for programming the XOR gate.
9. For each product term, the inputs are marked with 1, 0 or — (dash).

10. If the variable in the product term appears in the form in which it is true, the corresponding input variable is marked with a 1. If it appears complemented, the corresponding input variable is marked with a 0. If the variable is absent from the product term, it is marked with a dash.

Table 4.18.1. PLA programming

Product term	Input			Output	
	A	B	C	(T) $F_1$	(C) $F_2$
$A\bar{B}$	1	0	—	1	—
$AB$	2	1	—	1	1
$AC$	3	—	1	—	1
$BC$	4	0	1	0	1
$\bar{A}B\bar{C}$					

11. The size of the PLA is specified by the number of inputs, the number of product terms, and the number of outputs. A typical integrated circuit PLA may have 16 inputs, 48 product terms and eight outputs.
12. For  $n$  inputs,  $k$  product terms, and  $m$  outputs, the internal logic of the PLA consists of  $n$  buffer-inverter gates,  $k$  AND gates,  $m$  OR gates, and  $m$  XOR gates.

**Que 4.19.** Draw the logic configuration of four input and four output PAL and explain.

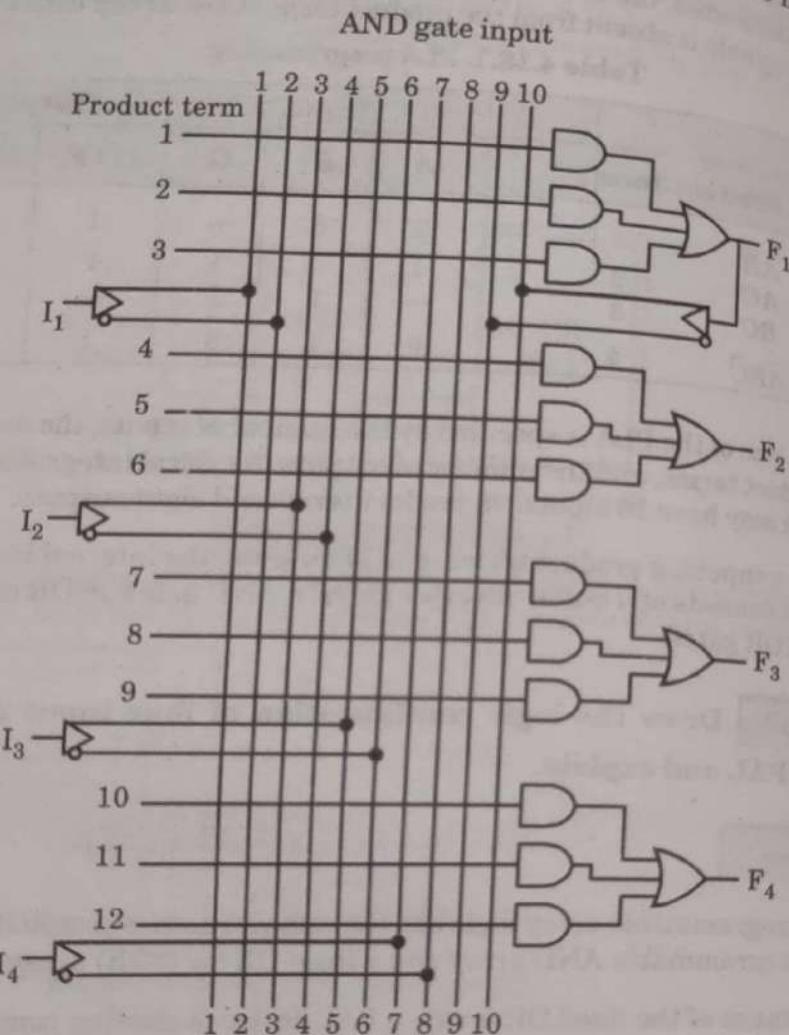
#### Answer

1. A programmable array logic has the same structure as a ROM, but has a programmable AND array and a fixed OR (or NOR) array.
2. Because of the fixed OR array, a PAL device is cheaper comparatively and easier to program.
3. However, the lack of shared rows with the column requires that each output function be simplified, with no common product term with others. It is easier to program, but is not flexible.

#### PAL with the four input and four output :

1. Each input has a buffer-inverter gate and each output is generated by fixed OR gate.
2. In designing with a PAL, the boolean function must be simplified to fit into each section unlike the situation with a PLA, a product term cannot be served among two or more OR gates. Therefore, each function can be simplified by itself, without regarding common product terms.

3. The number of product terms in each section is fixed, and if the number of terms in the function is too large, it may be necessary to have two sections to implement one boolean function.



**Fig. 4.19.1.** PAL with four inputs, four outputs and a three wire AND-OR structure.

**Que 4.20.** Realize the full adder circuit using the PAL.

**Answer**

**Full adder using PAL :** There are two functions used for the implementation of full adder :

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

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

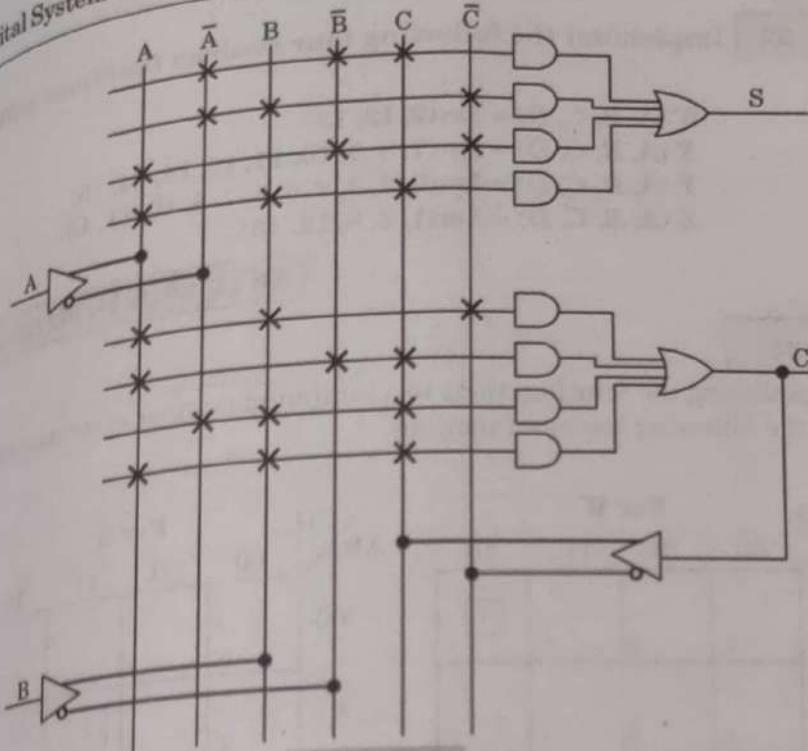


Fig. 4.20.1.

**Que 4.21.** Differentiate between PLA and PAL. Realize the full adder circuit using PAL.

AKTU 2018-19, Marks 3.5

**Answer**

A Difference :

S.No.	PAL	PLA
1.	It is moderately expensive and moderately complicated.	It is expensive than PAL and PROM and complicated to use.
2.	In this, only the AND array is programmable, OR array is fixed.	In this, both AND and OR arrays are programmable.
3.	It is easier to program because only the AND gates are programmable.	It is complicated to program because both the AND and OR gates are programmable.
4.	It is less flexible due to fixed OR gates.	It is more flexible than PAL.

1. Full adder using PAL : Refer Q. 4.20, Page 4-24B, Unit-4.

**Que 4.22.** Implement the following four boolean functions with a PAL.

$$W(A, B, C, D) = \Sigma m(2, 12, 13)$$

$$X(A, B, C, D) = \Sigma m(7, 8, 9, 10, 11, 12, 13, 14, 15)$$

$$Y(A, B, C, D) = \Sigma m(0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15)$$

$$Z(A, B, C, D) = \Sigma m(1, 2, 8, 12, 13)$$

AKTU 2016-17, Marks 15

**Answer**

1. Simplifying the four functions to a minimum number of terms results in the following boolean functions :

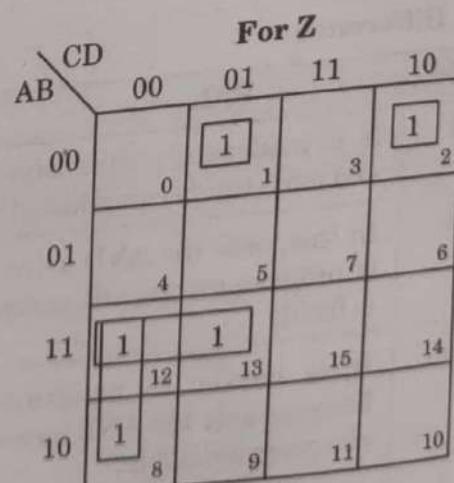
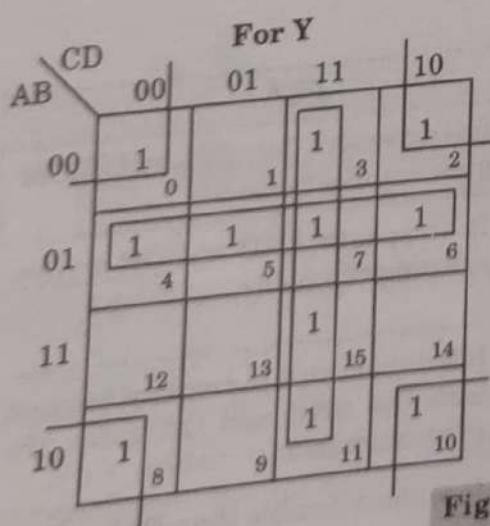
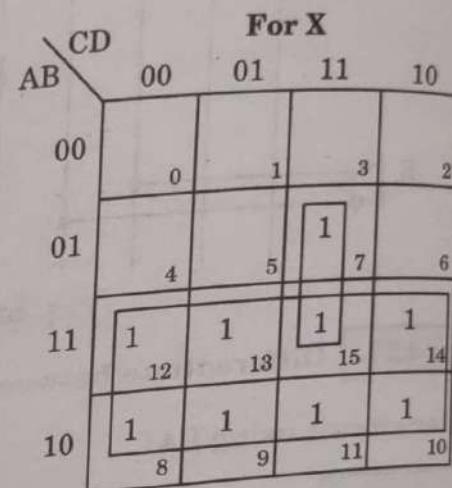
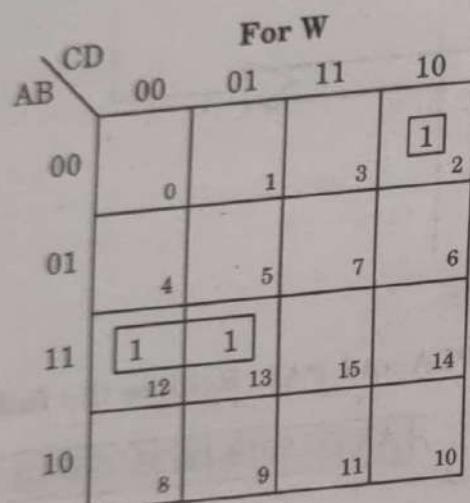


Fig. 4.22.1.

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

$$X = A + BCD$$

$$Y = \bar{A}B + CD + \bar{B}\bar{D}$$

$$Z = AB\bar{C} + \bar{A}\bar{B}CD + A\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D$$

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

2. Table 4.22.1 lists the PAL programming table for the four boolean functions. The table 4.22.1 is divided into four sections with three product terms in each section.
3. The first two sections need only two product terms to implement the boolean function. The last section for output Z needs four product terms. Using the output from W, we can reduce the function to three terms.
4. The fuse map for the PAL as specified in the programming Table 4.22.1 is shown in Fig. 4.22.2 for each 1 or 0 in the Table 4.22.1.
5. We mark the corresponding intersection in the diagram with the symbol for an intact fuse. For each dash, we mark the diagram with blown fuses in both the true and complement inputs.
6. If the AND gate is not used we leave all its input fuse intact. Since the corresponding input receives both the true value and the complement of each input variable, we have  $A\bar{A} = 0$  and the output of the AND gate is always 0.

Table 4.22.1. PAL programming

Product Term	AND Input					Outputs
	A	B	C	D	W	
1	1	1	0	—	—	$W = AB\bar{C} + \bar{A}\bar{B}CD$
2	0	0	1	0	—	
3	—	—	—	—	—	$X = A + BCD$
4	1	—	—	—	—	
5	—	1	1	1	—	
6	—	—	—	—	—	$Y = \bar{A}B + CD + \bar{B}\bar{D}$
7	0	1	—	—	—	
8	—	—	1	1	—	
9	—	0	—	0	—	
10	—	—	—	—	1	$Z = W + A\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D$
11	1	—	0	0	—	
12	0	0	0	1	—	

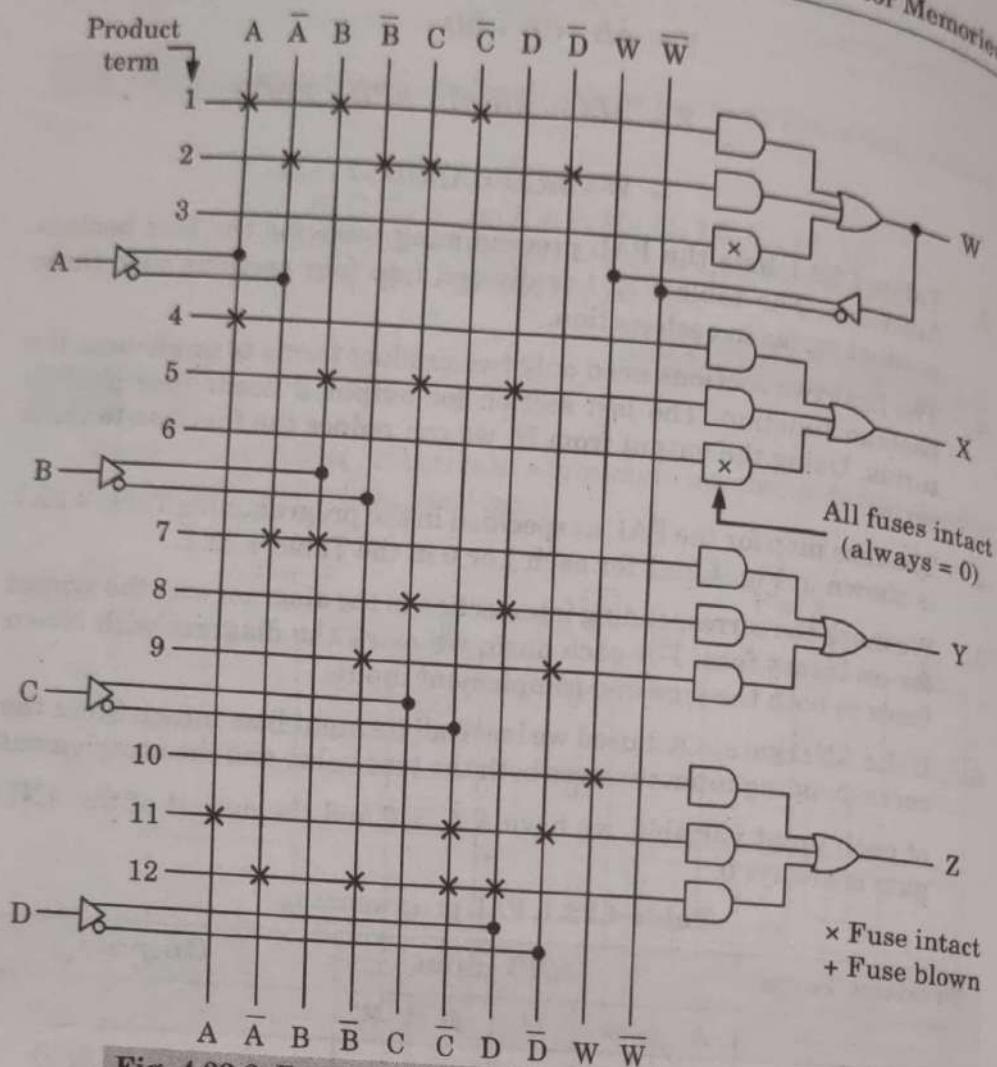


Fig. 4.22.2. Fuse map for PAL as specified in Table. 4.22.1.

**Que 4.23.** A combinational circuit is defined by the functions :

$$F_1(A, B, C) = \Sigma m(3, 5, 6)$$

$$F_2(A, B, C) = \Sigma m(0, 2, 7)$$

Implement the circuit with a PLA.

### Answer

Simplify the given boolean expression

$$F_1(A, B, C) = \Sigma m(3, 5, 6)$$

$$F_2(A, B, C) = \Sigma m(0, 2, 7)$$

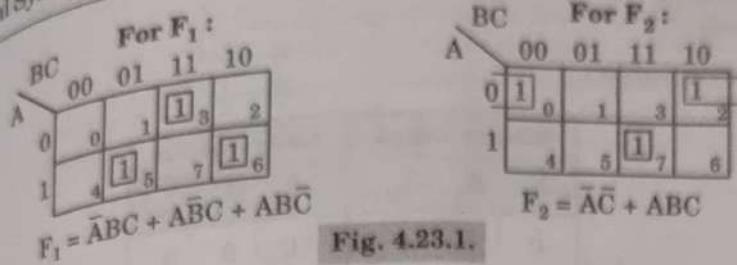


Fig. 4.23.1.

PLA program table :

Product term	Inputs			Outputs	
	A	B	C	$F_1$	$F_2$
$\bar{A}\bar{B}C$	0	1	1	1	-
$A\bar{B}C$	1	0	1	1	-
$A\bar{B}\bar{C}$	1	1	0	1	-
$\bar{A}\bar{C}$	0	-	0	-	1
$ABC$	1	1	1	-	1

Implementation :

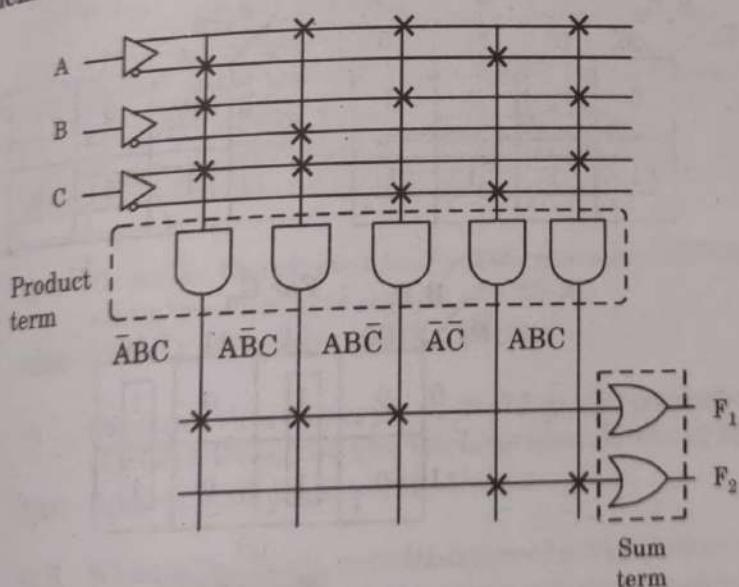


Fig. 4.23.2.

Que 4.24. Design a 3-bit binary to Gray code converter using PLA.

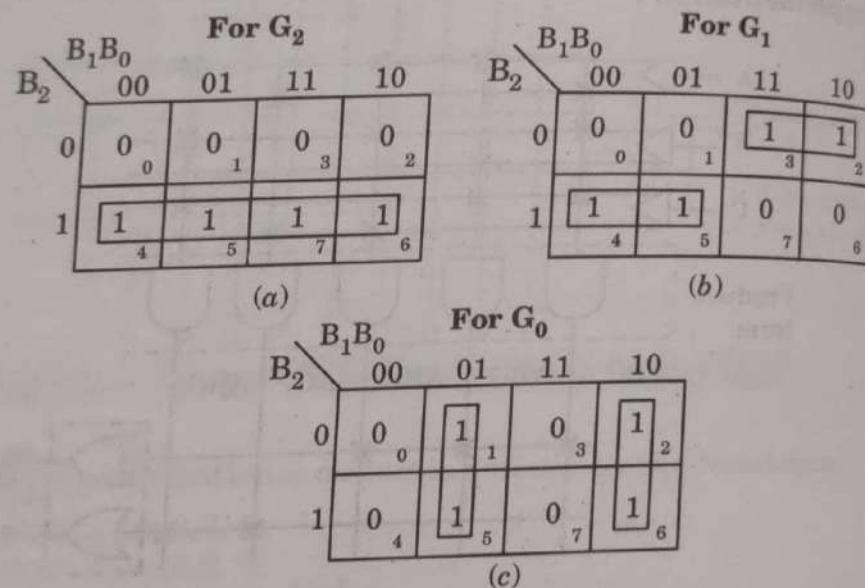
AKTU 2015-16, Marks 10

**Answer****Truth table :**

Binary input			Gray output		
$B_2$	$B_1$	$B_0$	$G_2$	$G_1$	$G_0$
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

**Simplification using K-map :**

The K-maps for the Gray outputs are as shown in Fig. 4.24.1.

**Fig. 4.24.1.**

$$G_2 = B_2$$

$$G_1 = \bar{B}_2 B_1 + B_2 \bar{B}_1$$

$$G_0 = \bar{B}_1 B_0 + B_1 \bar{B}_0$$

**Implementation :**

Fig. 4.24.2 shows the implementation using PLA.

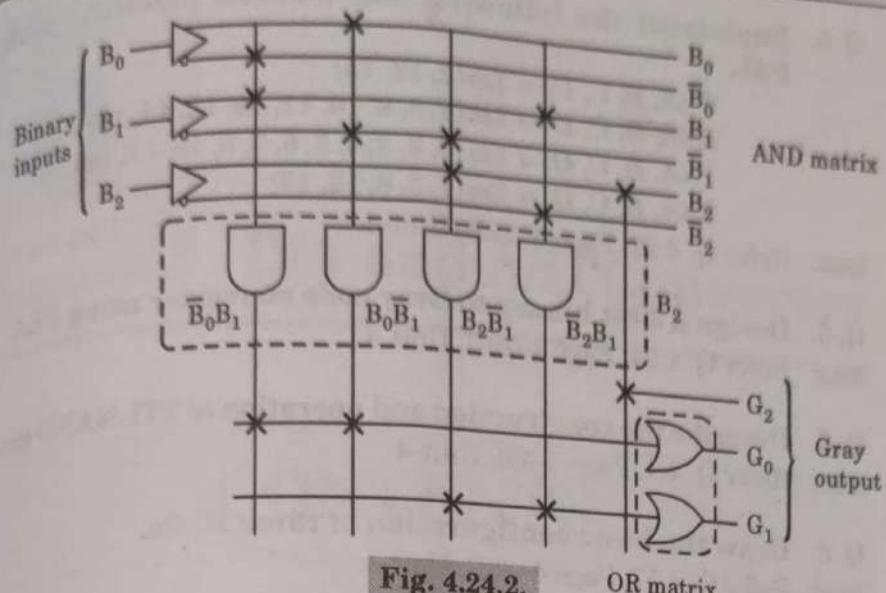


Fig. 4.24.2.

**VERY IMPORTANT QUESTIONS**

*Following questions are very important. These questions may be asked in your SESSIONALS as well as UNIVERSITY EXAMINATION.*

- Q. 1. Describe the circuit and performance of CMOS inverter and state the characteristics of CMOS.**

**Ans.** Refer Q. 4.10, Page 4-11B, Unit-4.

- Q. 2. Discuss the concept of field programmable gate array (FPGA). Describe the various structures of FPGA.**

**Ans.** Refer Q. 4.16, Page 4-18B, Unit-4.

- Q. 3. What is the basic architecture of a PLA ? How is the capacity of a PLA specified ? How is it programmed ? Explain.**

**Ans.** Refer Q. 4.18, Page 4-21B, Unit-4.

- Q. 4. Differentiate between PLA and PAL. Realize the full adder circuit using PAL.**

**Ans.** Refer Q. 4.21, Page 4-25B, Unit-4.

Q. 5. Implement the following four boolean functions with a PAL.

$$W(A, B, C, D) = \Sigma m(2, 12, 13)$$

$$X(A, B, C, D) = \Sigma m(7, 8, 9, 10, 11, 12, 13, 14, 15)$$

$$Y(A, B, C, D) = \Sigma m(0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15)$$

$$Z(A, B, C, D) = \Sigma m(1, 2, 8, 12, 13)$$

Ans. Refer Q. 4.22, Page 4-26B, Unit-4.

Q. 6. Design a 3-bit binary to Gray code converter using PLA.

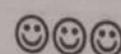
Ans. Refer Q. 4.24, Page 4-29B, Unit-4.

Q. 7. Describe the construction and operation of TTL NAND gate.

Ans. Refer Q. 4.1, Page 4-2B, Unit-4.

Q. 8. Draw the basic configuration of three PLDs.

Ans. Refer Q. 4.17, Page 4-20B, Unit-4.





## D/A and A/D Converter

### CONTENTS

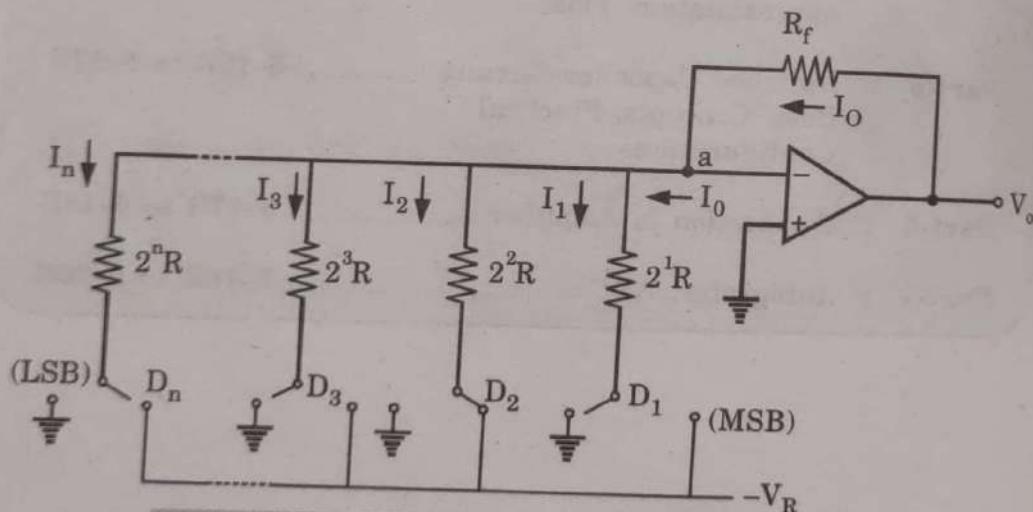
- |          |                                       |                |
|----------|---------------------------------------|----------------|
| Part-1 : | Weighted Resistor, R-2 R Ladder ..... | 5-2B to 5-7B   |
| Part-2 : | Resistor String .....                 | 5-7B to 5-8B   |
| Part-3 : | Analog to Digital Converters : .....  | 5-8B to 5-10B  |
|          | Single Slope                          |                |
| Part-4 : | Dual Slope, Successive .....          | 5-10B to 5-15B |
|          | Approximation, Flash                  |                |
| Part-5 : | Switched Capacitor Circuits : .....   | 5-15B to 5-17B |
|          | Basic Concepts, Practical             |                |
|          | Configurations                        |                |
| Part-6 : | Application in Amplifier .....        | 5-17B to 5-18B |
| Part-7 : | Integrator, ADC .....                 | 5-18B to 5-20B |

**PART-1***Weighted Resistor, R-2 R Ladder.***Questions-Answers****Long Answer Type and Medium Answer Type Questions**

**Que 5.1.** Explain working of weighted resistor D/A converter.

**Answer**

1. In a weighted resistor D/A converter every resistor has a definite weight assigned to it.
2. One of the simplest circuit shown in Fig. 5.1.1, is a summing amplifier with a binary weighted resistor network. It has  $n$ -electronic switches  $D_n, D_{n-1} \dots D_2, D_1$  controlled by binary input word.
3. If the binary input to a particular switch is 1, it connects the resistor to the reference voltage ( $-V_R$ ). And if the input bit is 0, the switch connects the resistor to the ground.



**Fig. 5.1.1.** A simple weighted resistor D/A converter.

4. From Fig. 5.1.1, the output current  $I_o$  for an ideal op-amp can be written as

$$\begin{aligned}
 I_o &= I_1 + I_2 + I_3 \dots + I_n \\
 &= \frac{V_R}{2R} D_1 + \frac{V_R}{2^2 R} D_2 + \frac{V_R}{2^3 R} D_3 + \dots + \frac{V_R}{2^n R} D_n \\
 &= \frac{V_R}{R} (D_1 2^{-1} + D_2 2^{-2} + D_3 2^{-3} + \dots + D_n 2^{-n})
 \end{aligned}$$

5. The output voltage,

$$V_o = I_o R_f$$

$$= V_R \frac{R_f}{R} (D_1 2^{-1} + D_2 2^{-2} + D_3 2^{-3} + \dots + D_n 2^{-n})$$

**Que 5.2.** Draw a 4-bit binary weighted D/A converter, find the value of step size if  $R = 10 K$  and  $R_f = 1.2 K$ . What is the output voltage when all binary inputs are at 5 V?

**Answer**

A Weighted resistor D/A converter :

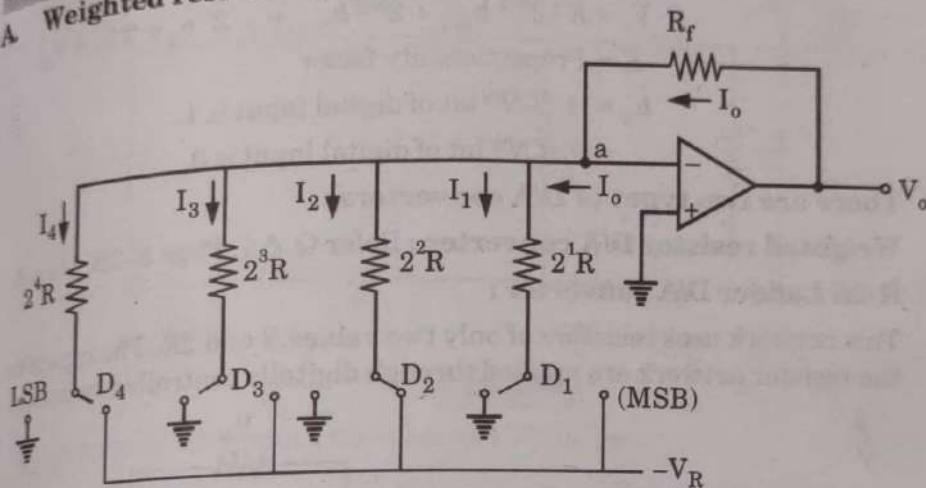


Fig. 5.2.1. 4-bit binary weighted resistor D/A converter.

B. Numerical :

Given :  $R = 10 K$ ,  $R_f = 1.2 K$ ,  $V_{in} = 5 V$ ,  $N = 4$

To Find : Step size, Output voltage.

$$1. \text{ Step size} = \frac{V_{in}}{2^N - 1} = \frac{5}{2^4 - 1} = \frac{5}{15} = 0.333 \text{ V}$$

$$2. \text{ Output voltage} = V_R \frac{R_f}{R} (2^{-1} + 2^{-2} + 2^{-3} + 2^{-4})$$

$$= \frac{5 \times 1.2}{10} \left( \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} \right)$$

$$= 0.5625 \text{ V}$$

**Que 5.3.** What is a DAC ? Describe the weighted resistor DAC.

Give mathematical expressions in support of your answer.

OR

Explain different types of DAC.

**Answer****DAC :**

- It is the process of taking a value represented in digital code and converting it into a voltage or current which is proportional to digital value. It is accomplished by the use of DAC or D/A converter.
- The analog voltage output  $V_o$  of an  $N$ -bit straight binary D/A converter is related to the digital input by the relation.

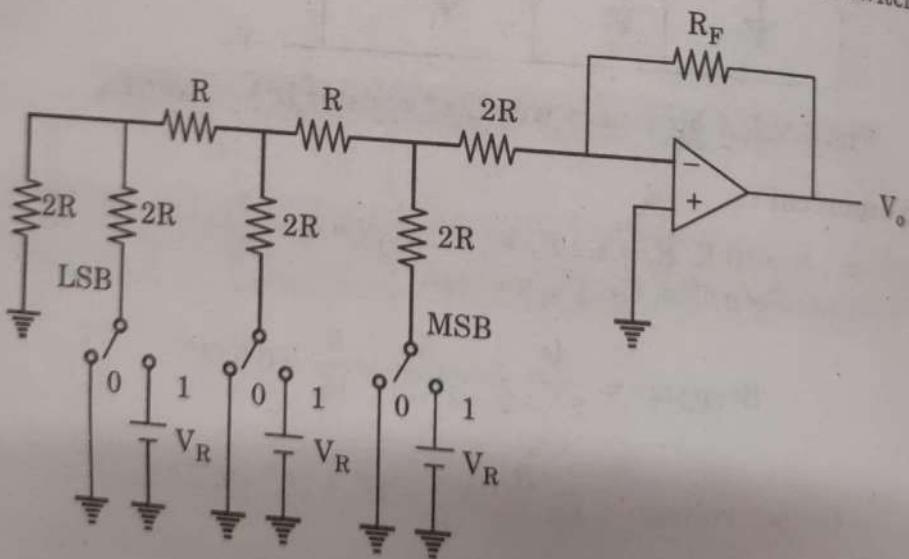
$$V_o = K (2^{N-1} b_{N-1} + 2^{N-2} b_{N-2} + \dots + 2^2 b_2 + 2 b_1 + b_0)$$

$K$  = Proportionality factor

$$b_N = \begin{cases} 1, & \text{if } N^{\text{th}} \text{ bit of digital input is 1} \\ 0, & \text{if } N^{\text{th}} \text{ bit of digital input is 0.} \end{cases}$$

**There are two types of D/A converters :**

- Weighted resistor D/A converter :** Refer Q. 5.1, Page 5-2B, Unit-5.
  - R-2R Ladder D/A converter :**
- This network uses resistors of only two values  $R$  and  $2R$ . The inputs to the resistor network are applied through digitally controlled switches.

**Fig. 5.3.1.**

- Consider a 3 bit R-2R Ladder D/A network. Let us assume a digital input of 001. The equivalent circuit becomes as shown in Fig. 5.3.2.

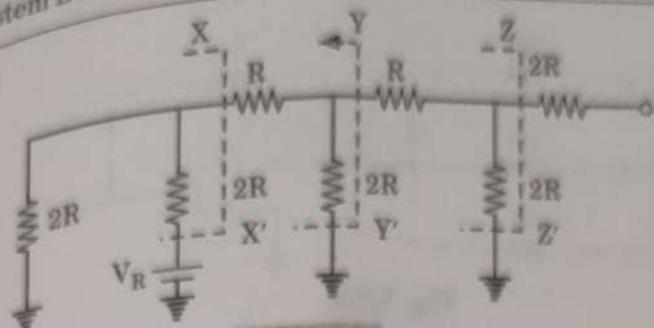


Fig. 5.3.2.

3. Applying Thevenin's theorem at point  $XX'$ , we get

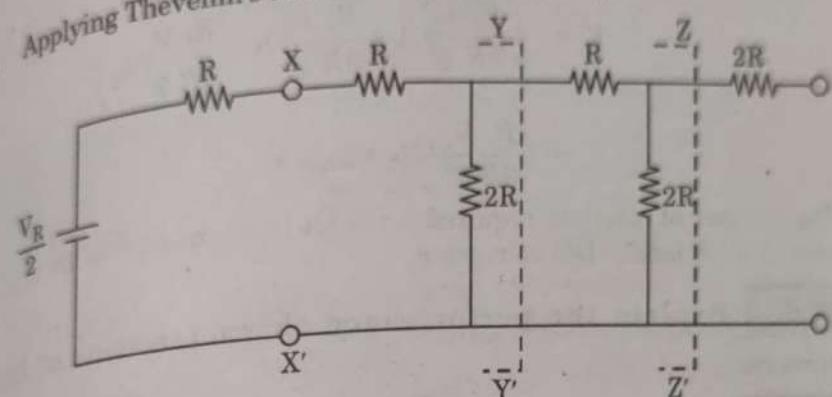


Fig. 5.3.3.

4. Applying Thevenin's theorem at  $YY'$ , we get

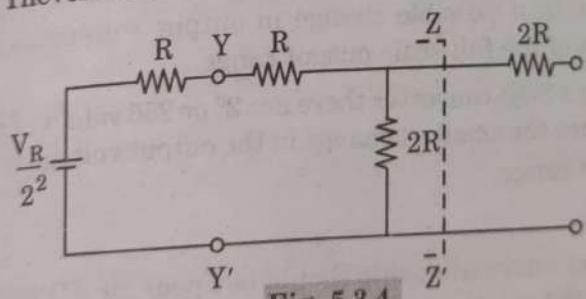


Fig. 5.3.4.

5. Applying Thevenin's theorem at  $ZZ'$ , we get

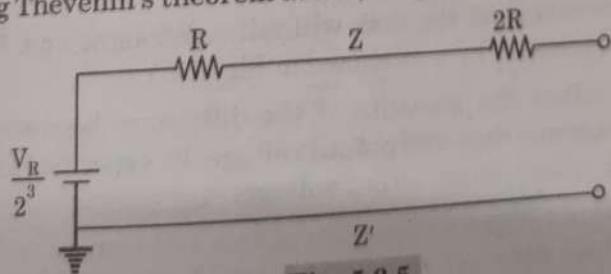


Fig. 5.3.5.

6. The equivalent resistance is  $3R$  in each case. The circuit reduces to

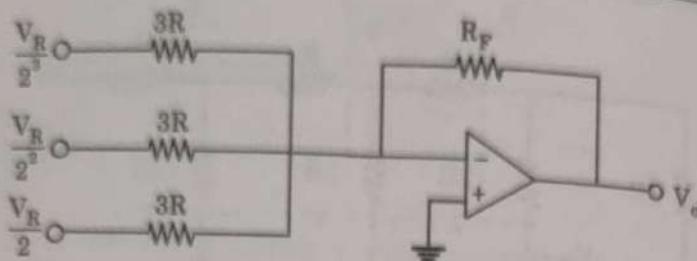


Fig. 5.3.6.

7. The output voltage is given as

$$\begin{aligned} V_o &= -\left( \frac{R_f}{3R} \frac{V_R}{2^3} b_0 + \frac{R_f}{3R} \frac{V_R}{2^2} b_1 + \frac{R_f}{3R} \frac{V_R}{2^1} b_2 \right) \\ &= -\frac{R_f}{3R} \frac{V_R}{2^3} [4b_2 + 2b_1 + b_0] \end{aligned}$$

8. The number of resistors required for  $N$ -bit D/A converter is  $2N$  in the case of  $R-2R$  ladder D/A converter.

**Que 5.4.** Explain the performance characteristics of D/A converters.

**Answer**

**1. Resolution :**

- i. It is the smallest possible change in output voltage as a fraction or percentages of the full-scale output range.
- ii. Example for a 8-bit converter there are  $2^8$  or 256 values of analog output voltage, hence the smallest change in the output voltage is  $1/255^{\text{th}}$  of full scale output range.

**2. Linearity :**

- i. The input-output relationship should be linear for a D/A converter. But sometimes the relation is non-linear. This is due to error in resistor values and voltage across the switches.
- ii. If converter was ideal the dots will fall on straight line. But if it has errors it is indicated by  $\epsilon$  as shown in Fig. 5.4.1.

**3. Accuracy :** It is the measure of the difference between the actual output voltage and expected output voltage. It is specified as percentage of full scale or maximum output voltage.

**4. Settling time :** When a digital input to a D/A converter changes, the output voltage does not change abruptly because of the delay in the circuit. The time required for the analog output voltage to settle within  $\pm (1/2)$  LSB of the final value after a change in digital input is called settling time.

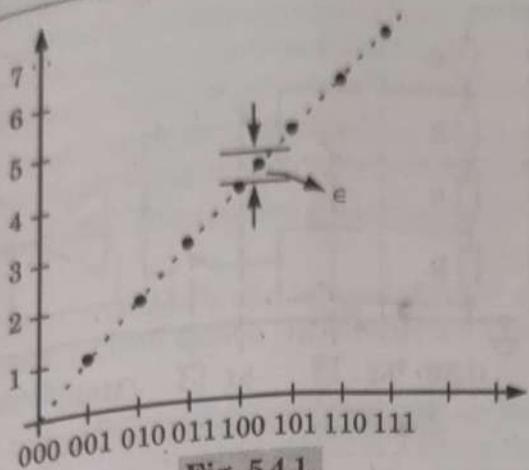


Fig. 5.4.1.

**PART-2***Resistor String.***Questions-Answers****Long Answer Type and Medium Answer Type Questions**

**Que 5.5.** Write a short note on resistor string.

**Answer**

1. A simple DAC that is inherently monotonic is the resistor string DAC, as shown in Fig. 5.5.1.
2. In this circuit, a reference voltage ( $V_{REF}$ ) is connected across a resistor string of equal value resistance.
3. For an  $n$ -bit DAC, this requires  $2^n$  resistors and the taps between the resistors are connected to switches controlled by the digital logic input values (and their complement).
4. At the output node, an output voltage ( $V_{OUT}$ ) is generated, the value set by the positions of the switches.
5. The output would need to be suitably buffered in order to prevent electrical loading of the converter output.

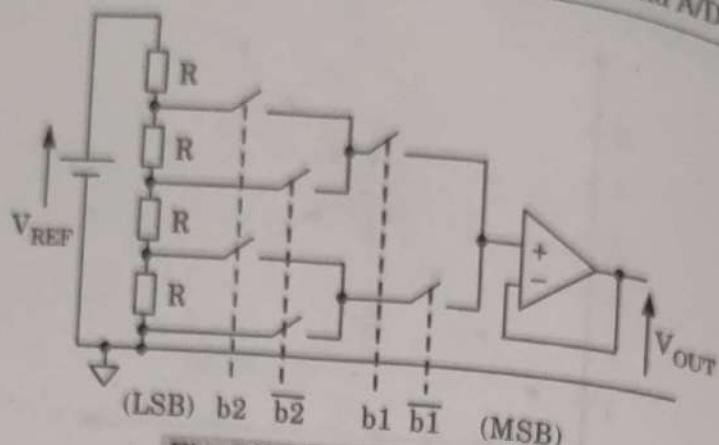


Fig. 5.5.1. Resistor string DAC.

**PART-3***Analog to Digital Converters : Single Slope.***Questions-Answers****Long Answer Type and Medium Answer Type Questions**

**Que 5.6.** Write short note on analog to digital converter (ADC).

**Answer**

1. The block schematic of ADC in Fig. 5.6.1 provides the function just opposite to that of a DAC.

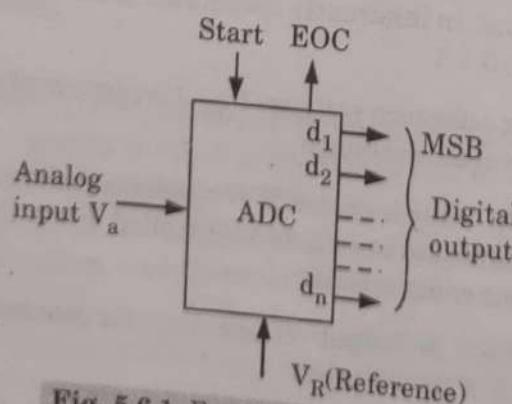


Fig. 5.6.1. Functional diagram of ADC.

2. It accepts an analog input  $V_a$  and produces an output binary word  $d_1, d_2, \dots, d_n$  of functional value  $D$ , so that

$$D = d_1 2^{-1} + d_2 2^{-2} + \dots + d_n 2^{-n} \quad \dots(5.6.1)$$

where  $d_1$  is the most significant bit and  $d_n$  is the least significant bit.

3. An ADC usually has two additional control lines, the START input to tell the ADC when to start the conversion and the EOC (end of conversion) output to announce when the conversion is complete.
4. Depending upon the type of application, ADCs are designed for microprocessor interfacing or to directly drive LCD or LED displays.
5. ADCs are classified broadly into two groups according to their conversion technique.
6. These are Direct type ADCs and Integrating type ADCs. Direct type ADCs compares a given analog signal with the internally generated equivalent signal.
7. Integrating type ADCs perform conversion in an indirect manner by first changing the analog input signal to a linear function of time or frequency and then to a digital code. The two most widely used integrating type converters are :
  - i. Charge balancing ADC.
  - ii. Dual slope ADC.
8. The most commonly used ADCs are successive approximation and the integrator type. The successive approximation ADCs are used in applications such as data loggers and instrumentation where conversion speed is important.
9. The successive approximation and comparator types are faster but generally less accurate than integrating type converters. The flash (comparator) type is expensive for high degree of accuracy.
10. The integrating type converter is used in applications such as digital meter, panel meter and monitoring systems where the conversion accuracy is critical.

**Que 5.7. Explain the working of single slope ADC.**

**Answer**

1. It consists of a ramp generator and BCD or binary counters. The Fig. 5.7.1 shows the single slope ADC.
2. At the start, the reset signal is provided to the ramp generator and the counters. Thus counters are reset to 0's.
3. The analog input voltage  $V_{in}$  is applied to the positive terminal of the comparator.
4. As this is more positive than the negative input, the comparator output goes high.
5. The output of ramp generator is applied to the negative terminal of the comparator.
6. The high output of the comparator enables the AND gate which allows clock to reach to the counters and also this high output starts the ramp.

### 5-10 B (EC-Sem-3)

D/A and A/D Converter

7. The ramp voltage goes positive until it exceeds the input voltage. When it exceeds  $V_{in}$ , comparator output goes low.
8. This disables AND gate which in turn stops the clock to the counters. The control circuitry provides the latch signal which is used to latch the counter data.
9. The reset signal resets the counters to 0's and also resets the ramp generator. The latched data is then displayed using decoder and a display device.

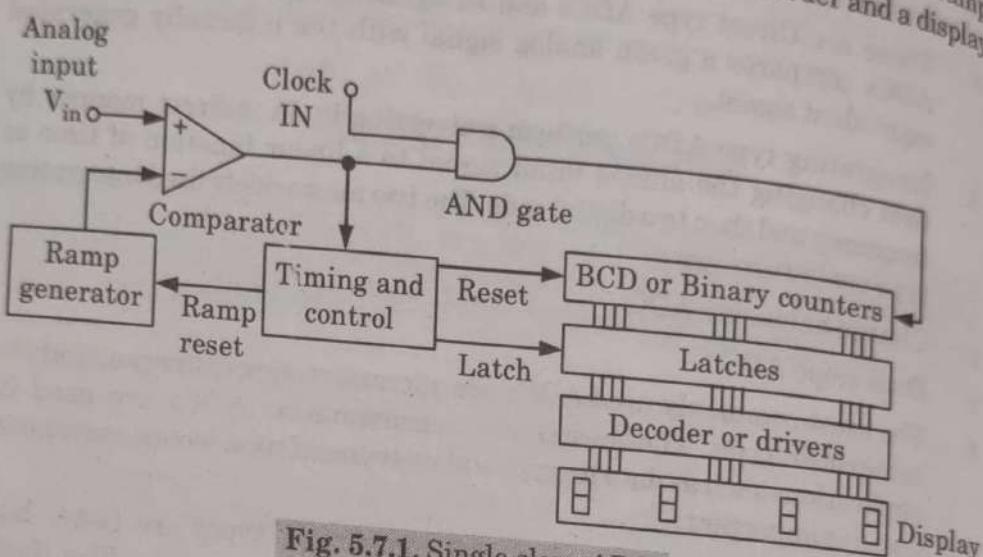


Fig. 5.7.1. Single slope ADC.

### PART-4

*Dual Slope, Successive Approximation, Flash.*

#### Questions-Answers

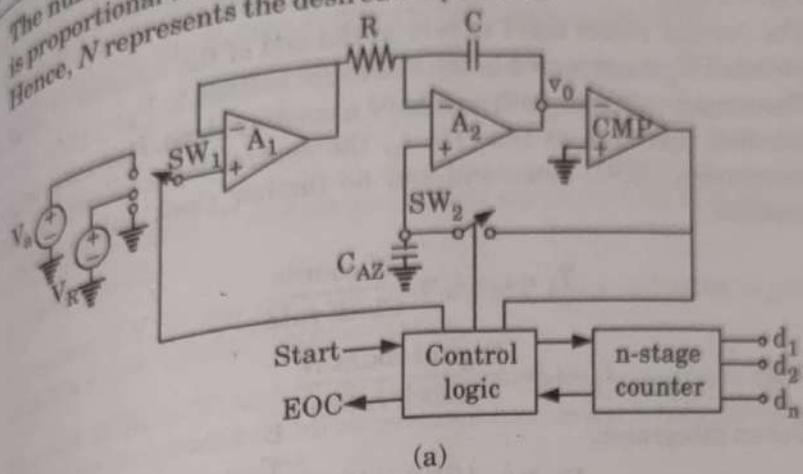
#### Long Answer Type and Medium Answer Type Questions

**Que 5.8.** Explain the working of dual slope integrating ADC with the help of circuit diagram.

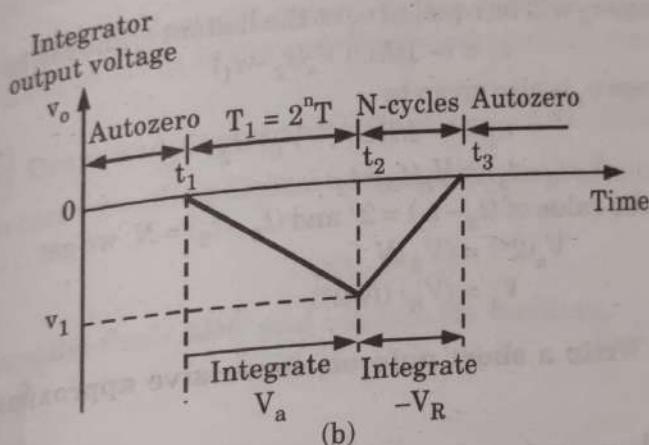
#### Answer

1. Fig. 5.8.1(a) shows the dual slop ADC functional diagram. The circuit consists of a high input impedance buffer  $A_1$ , precision integrator  $A_2$  and a voltage comparator.
2. The converter first integrates the analog input signal  $V_a$  for a fixed duration of  $2^n$  clock periods as shown in Fig. 5.8.1(b).
3. Then it integrates an internal reference voltage  $V_R$  of opposite polarity until the integrator output is zero.

- The number  $N$  of clock cycles required to return the integrator to zero is proportional to the value of  $V_a$  averaged over the integration period. Hence,  $N$  represents the desired output code.



(a)



(b)

Fig. 5.8.1. (a) Functional diagram of the dual slope ADC  
 (b) Integrated output waveform for the dual slope ADC.

#### Operation :

- Before the START command arrives, the switch  $SW_1$  is connected to ground and  $SW_2$  is closed.
- Any offset voltage present in the  $A_1, A_2$ , comparator loop after integration, appears across the capacitor  $C_{AZ}$  till the threshold of the comparator is achieved.
- The capacitor  $C_{AZ}$  thus provides automatic compensation for the input-offset voltages of all the three amplifiers.
- Later, when  $SW_2$  opens,  $C_{AZ}$  acts as a memory to hold the voltage required to keep the offset nulled.
- At the arrival of the START command at  $t = t_1$ , the control logic opens  $SW_2$  and connects  $SW_1$  to  $V_a$  and enables the counter starting from zero.
- The analog voltage  $V_a$  is integrated for a fixed number  $2^n$  counts of clock pulses after which the counter resets to zero.

7. If the clock period is  $T$ , the integration takes place for a time  $T_1 = 2^n \times T$  and the output is a ramp going downwards as shown in Fig. 5.8.1(b).
8. The counter resets itself to zero at the end of the interval  $T_1$  and the switch  $SW_1$  is connected to the reference voltage ( $-V_R$ ).
9. The output voltage  $v_o$  will now have a positive slope. However, when it becomes just zero at time  $t = t_3$ , the control logic issues an end of conversion (EOC) command and no further clock pulses enter the counter.

$$T_1 = t_2 - t_1 = \frac{2^n \text{ counts}}{\text{Clock rate}}$$

and  $t_3 - t_2 = \frac{\text{Digital count } N}{\text{Clock rate}}$

10. For an integrator,

$$\Delta v_o = (-1/RC) V(\Delta t)$$

11. The voltage  $v_o$  will be equal to  $v_1$  at the instant  $t_2$  and can be written as

$$v_1 = (-1/RC) V_a(t_2 - t_1)$$

12. The voltage  $v_1$  is also given by

$$v_1 = (-1/RC)(-V_R)(t_2 - t_3)$$

So,  $V_a(t_2 - t_1) = V_R(t_3 - t_2)$

Putting the value of  $(t_2 - t_1) = 2^n$  and  $(t_3 - t_2) = N$ , we get

$$V_a(2^n) = (V_R)N$$

or,  $V_a = (V_R)(N/2^n)$

**Que 5.9.** Write a short note on successive approximation A/D converter.

**Answer**

1. The successive approximation technique uses a very efficient code strategy to provide  $n$ -bit conversion in  $n$ -clock periods.

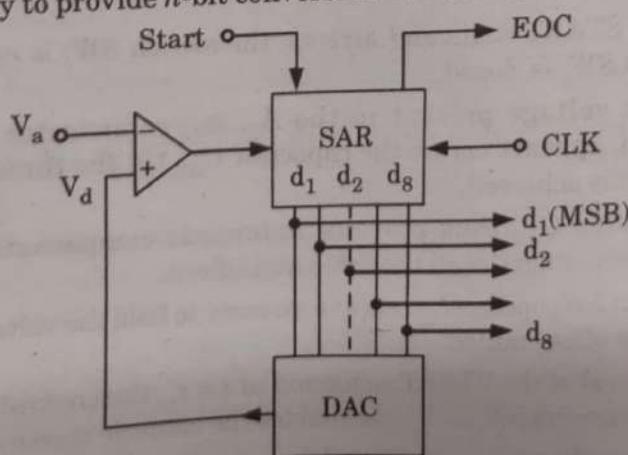


Fig. 5.9.1. Functional diagram of the successive approximation ADC.

**Working :**

1. With the arrival of the START command, the SAR sets the MSB  $d_1 = 1$  with all other bits to zero so that the trial code is 10000000.
2. The output  $V_d$  of the DAC is now compared with analog input  $V_a$ . If  $V_d$  is greater than the DAC output  $V_d$ , then 10000000 is less than the correct digital representation.
3. The MSB is left at '1' and the next lower significant bit is made '1' and further tested.
4. However, if  $V_d$  is less than the DAC output, then 10000000 is greater than the correct digital representation.
5. So reset MSB to '0' and go on to the next lower significant bit. This procedure is repeated for all subsequent bits, one at a time, until all bit positions have been tested.
6. Whenever the DAC output crosses  $V_a$ , the comparator changes state and this can be taken as the end of conversion (EOC) command.

**Que 5.10.** Draw and explain the flash type A/D converter. Also discuss the corresponding digital output with respect to input signal voltage.

**OR**

Design a parallel-flash ADC and explain its working.

**Answer**

The commonly used A/D converter is parallel or flash converter. A 3-bit parallel comparator A/D converter is shown in Fig. 5.10.1.

**Working :**

1. Let  $V_a$  is the input analog voltage to be converted into digital form. The reference voltage like  $V_{r1}, V_{r2}, \dots$  are generated using resistor network.
2.  $V_a$  is compared simultaneously with the reference voltage by using comparators.

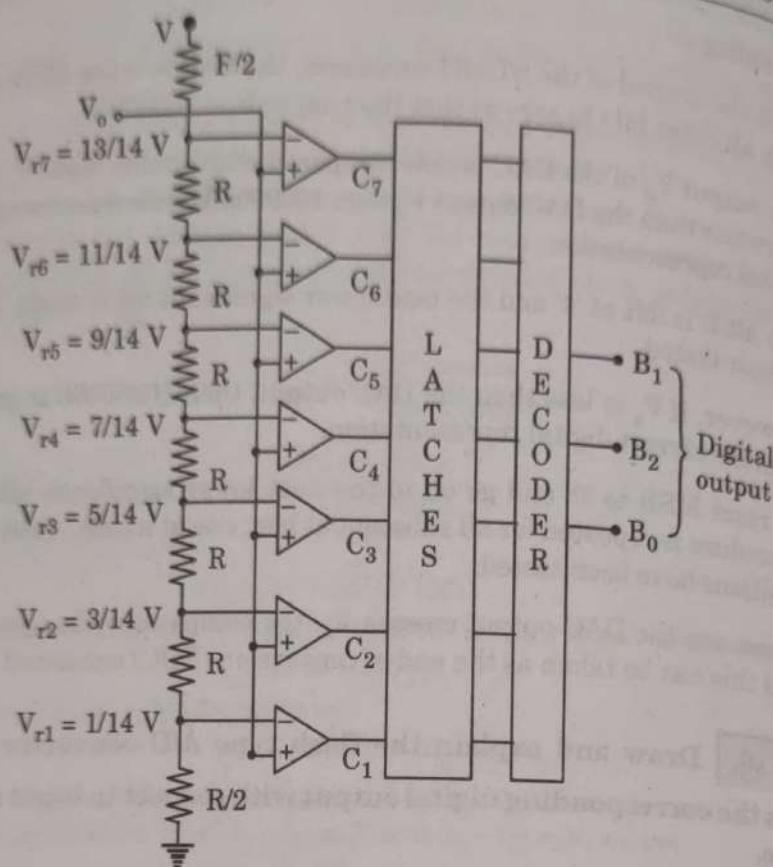


Fig. 5.10.1.

3. A 7-bit output is obtained from the comparator is converted to a 3-bit output using decoder circuit.
4. The process adopted here is the simplest and it works quite fast.
5. The demerits rapidly increase in the number of comparators with the number of bits and the corresponding complications of the decoder circuit.
6. The analog input, comparator outputs and digital output are shown in Table 5.10.1

Table 5.10.1.

Analog input $V_a$	Comparator outputs						Digital output			
	$C_7$	$C_6$	$C_5$	$C_4$	$C_3$	$C_2$	$C_1$	$B_2$	$B_1$	$B_0$
$0 \leq V_a < V_{r1}$	0	0	0	0	0	0	0	0	0	0
$V_{r1} < V_a < V_{r2}$	0	0	0	0	0	0	1	0	0	1
$V_{r2} < V_a < V_{r3}$	0	0	0	0	0	1	1	0	1	0
$V_{r3} < V_a < V_{r4}$	0	0	0	0	1	1	1	0	1	1
$V_{r4} < V_a < V_{r5}$	0	0	0	1	1	1	1	1	0	0

$V_{r5} < V_a < V_{r6}$   
 $V_{r6} < V_a < V_{r7}$   
 $V_{r7} < V_a \leq V$

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

**PART-5**

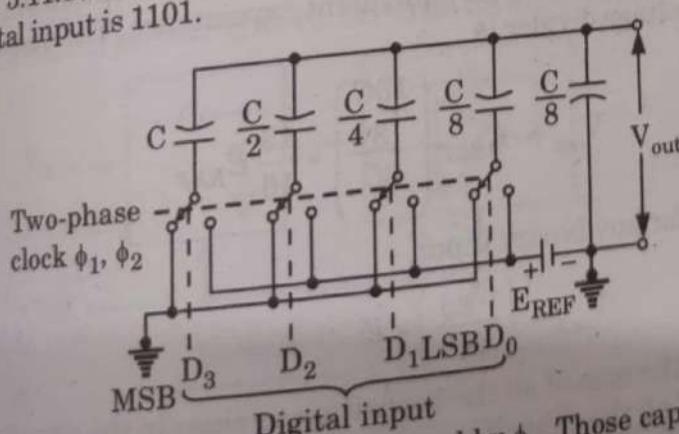
*Switched Capacitor Circuits : Basic Concept,  
Practical Configurations.*

**Questions-Answers****Long Answer Type and Medium Answer Type Questions**

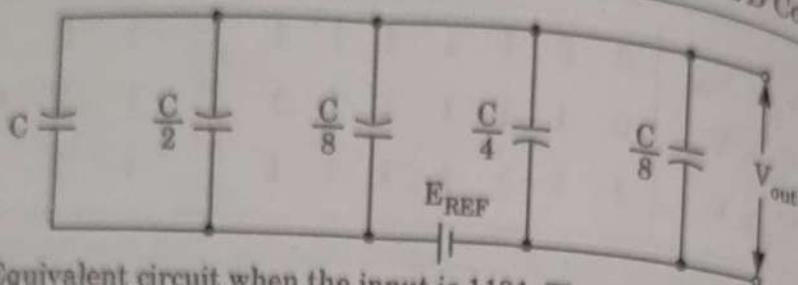
**Que 5.11.** Explain a 4-bit switched capacitor DAC.

**Answer**

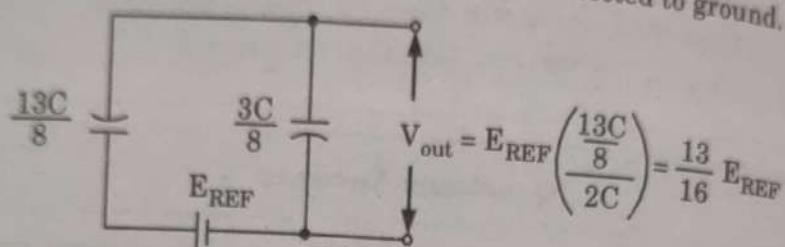
- Fig. 5.11.1(a) shows a 4-bit switched capacitor DAC, that the capacitance values have binary weights.
- A two-phase clock is used to control switching of the capacitors. When  $\phi_1$  goes HIGH, all capacitors are switched to ground and discharged.
- When  $\phi_2$  goes HIGH, those capacitors where the digital inputs are HIGH are switched to  $E_{REF}$ , whereas those inputs are LOW remain grounded.
- Fig. 5.11.1(b) shows the equivalent circuit when  $\phi_2$  is HIGH and the digital input is 1101.



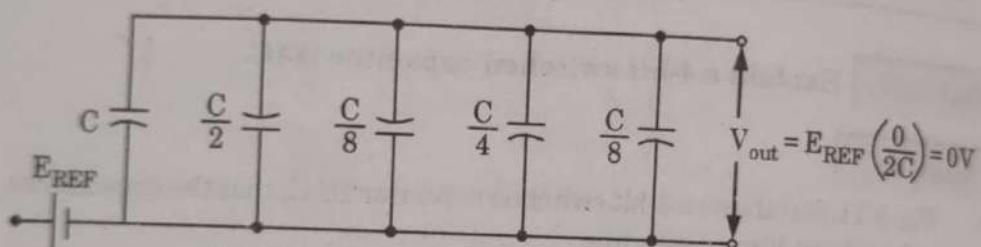
(a) All capacitors are switched to ground by  $\phi_2$ . Those capacitors whose digital inputs are HIGH are switched to  $E_{REF}$  by  $\phi_2$ .



(b) Equivalent circuit when the input is 1101. The capacitors switched to  $E_{REF}$  are in parallel as are the ones connected to ground.



(c) Circuit equivalent to (b). The output is determined by a capacitor voltage divider.



(d) Equivalent circuit when the input is 0000.

**Fig. 5.11.1.** The switched capacitor type DAC.

5. The capacitors whose digital inputs are 1, are in parallel and the capacitors whose digital inputs are 0, are in parallel with  $C/8$ .
6. The circuit as shown in Fig. 5.11.1(c), where each set of the parallel capacitors is replaced by its equivalent capacitance. The output of the capacitive voltage divider is

$$V_{out} = E_{REF} \left( \frac{\frac{13C}{8}}{2C} \right) = \frac{13}{16} E_{REF}$$

7. In general, for any binary input,

$$V_{out} = \left( \frac{C_{EQ}}{2C} \right) \times E$$

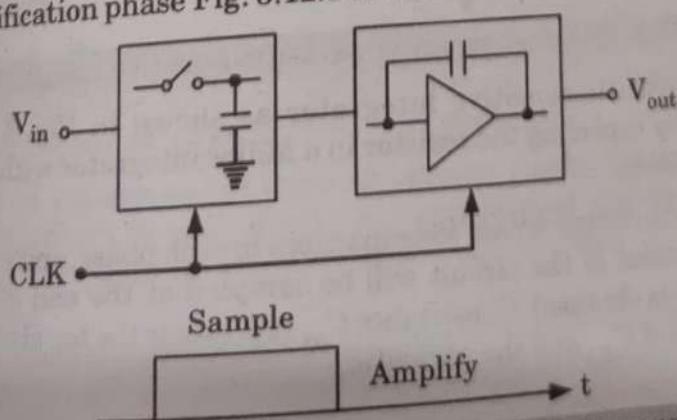
where  $2C$  is the sum of all the capacitance values in the circuit and  $C_{EQ}$  is the sum of all the capacitors whose digital inputs are HIGH.

8. The analog output is proportional to the digital input. When the input is 0000, the positive terminal of  $E_{REF}$  is effectively open-circuited as shown in Fig. 5.11.1(d) so, the output is 0 V.

9. Switched capacitor technology has been developed for implementing analog functions in integrated circuits, particularly MOS circuits. It is used to construct filters, amplifiers, and many other special devices.
10. The principal advantage of this technology is that, small capacitors of the order of a few picofarads can be constructed in the integrated circuits to perform the function of the much larger capacitors that are normally needed in low-frequency analog circuits.

**PART-6***Application in Amplifier.***Questions-Answers****Long Answer Type and Medium Answer Type Questions****Que 5.12.****Write a short note on switched capacitor amplifier.****Answer**

1. The switched capacitor amplifier of Fig. 5.12.1 lends itself to implementation in CMOS technology much more easily than in other technologies.
2. This is because discrete-time operations require switches to perform sampling as well as a high input impedance to sense the stored quantities with no corruption.
3. For example, if the op-amp of Fig. 5.12.1 incorporates bipolar transistors at its input, the base current drawn from the inverting input in the amplification phase Fig. 5.12.1 creates an error in the output voltage.

**Fig. 5.12.1. General view of switched-capacitor amplifier.**

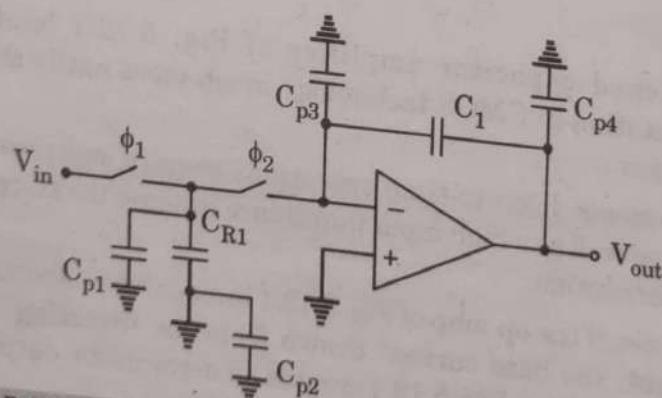
4. The existence of simple switches and high input impedance has made CMOS technology the dominant choice for sampled-data applications.

5. The foregoing discussion leads to the conceptual view illustrated in Fig. 5.12.1 for switched-capacitor amplifiers. In the simplest case, the operation takes place in two phases : sampling and amplification. Thus, in addition to the analog input,  $V_{in}$ , the circuit requires a clock to define each phase.

**PART-7***Integrator, ADC.***Questions-Answers****Long Answer Type and Medium Answer Type Questions**

**Que 5.13.** Discuss switched-capacitor (SC) parasitic-sensitive integrator with circuit diagram.

**Answer**



**Fig. 5.13.1.** Switched-capacitor parasitic-sensitive integrator.

1. The SC parasitic-sensitive integrator as shown in Fig. 5.13.1 was suggested by replacing the resistor in a Miller integrator with a parallel switch network.
2. Analyzing the charge across the capacitors in each phase, and considering that the output of the circuit will be sampled at the end of phase  $\phi_1$ , Table 5.13.1 is obtained. Capacitance  $C_{p1}$  represents the top plate parasitic capacitance of  $C_{R1}$  and the parasitic.

Table 5.13.1. Charge in the capacitors in each phase.

	$(n - 1)T$	$(n - 0.5)T$	$nT$
$Q_{CR1}$	$V_{in}[(n - 1)T]C_{R1}$	0	$V_{in}[nT]C_{R1}$
$Q_{C1}$	$-V_{out}[(n - 1)T]C_1$	$-V_{out}[(n - 0.5)T]C_1$	$-V_{out}[nT]C_1$
$Q_{Cp1}$	$V_{in}[(n - 1)T]C_{p1}$	0	$V_{in}[nT]C_{p1}$
$Q_{Cp2}$	0	0	0
$Q_{Cp3}$	0	0	0
$Q_{Cp4}$	$V_{out}[(n - 1)T]C_{p4}$	$V_{out}[(n - 0.5)T]C_{p4}$	$V_{out}[nT]C_{p4}$

- 3. Capacitances of both switches; capacitance  $C_{p2}$  represents the bottom plate parasitic capacitance of  $C_{R1}$ ; capacitance  $C_{p3}$  represents the top plate parasitic capacitance of  $C_1$ , the input capacitance of the op-amp, and the parasitic capacitance of switch  $\phi_2$ .
- 4. Capacitance  $C_{p4}$  represents the bottom plate parasitic capacitance of  $C_1$  and the input capacitance of the following stage.
- 5. Considering the transition  $(n - 1)T \rightarrow (n - 0.5)T (\phi_1 \rightarrow \phi_2)$  and the transition  $(n - 0.5)T \rightarrow (n)T (\phi_2 \rightarrow \phi_1)$ , Eq. 5.14.1 is obtained from adding all the capacitors that are connected to the virtual ground at the end of that transition ( $\phi_2$  in the first case and  $\phi_1$  in the second).
- 6. From the calculations it can also be concluded that the parasitic capacitance  $C_{p3}$  does not influence the performance of the circuit due to the virtual ground node in the negative node of the op-amp.

Que 5.14. Write a short note on parasitic-insensitive integrator.

#### Answer

##### Parasitic-Insensitive Integrator :

- 1. To overcome the nonlinear effect of the parasitic capacitance  $C_{p1}$ , new parasitic-insensitive structures were developed.
- 2. Fig. 5.14.1 shows one of these structures. Analyzing the charge across the capacitors in each phase, and considering that the output of the circuit will again be sampled at the end of phase  $\phi_1$ , Table 5.14.1 is obtained.
- 3. Capacitance  $C_{p1}$  represents the bottom plate parasitic capacitance of  $C_{R1}$  and the parasitic capacitances of the switches connected to the bottom plate of  $C_{R1}$ .
- 4. Capacitance  $C_{p2}$  represents the top plate parasitic capacitance of  $C_{R1}$  and the parasitic capacitances of the switches connected to the top plate of  $C_{R1}$ .

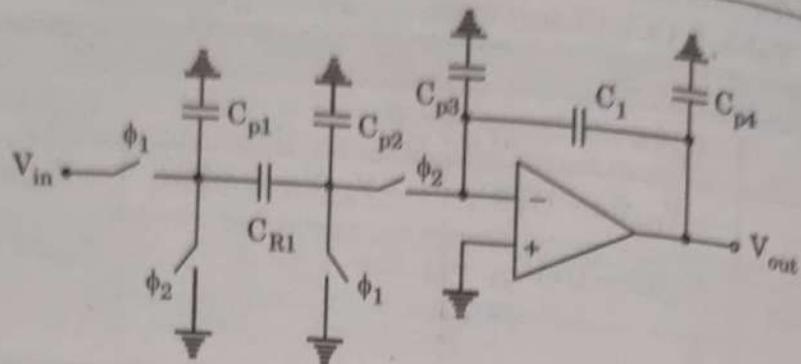


Fig. 5.14.1. Switched-capacitor parasitic-insensitive integrator.

Table 5.14.1. Charge in the capacitors in each phase.

	$(n - 1)T$	$(n - 0.5)T$	$nT$
$Q_{CR1}$	$-V_{in}[(n - 1)T]C_{R1}$	0	$-V_{in}[nT]C_{R1}$
$Q_{C_1}$	$-V_{out}[(n - 1)T]C_1$	$-V_{out}[(n - 0.5)T]C_1$	$-V_{out}[nT]C_1$
$Q_{C_{p1}}$	$V_{in}[(n - 1)T]C_{p1}$	0	$V_{in}[nT]C_{p1}$
$Q_{C_{p2}}$	0	0	0
$Q_{C_{p3}}$	0	0	0
$Q_{C_{p4}}$	$V_{out}[(n - 1)T]C_{p4}$	$V_{out}[(n - 0.5)T]C_{p4}$	$V_{out}[nT]C_{p4}$

5. Capacitance  $C_{p3}$  represents the top plate parasitic capacitance of  $C_1$ , the input capacitance of the op-amp, and the parasitic capacitance of switch connected to the top plate of  $C_1$ .
6. Capacitance  $C_{p4}$  represents the bottom plate parasitic capacitance of  $C_1$  and the input capacitance of the following stage.
7. Considering the transition  $(n - 1)T \rightarrow (n - 0.5)T$  ( $\phi_1 \rightarrow \phi_2$ ) and the transition  $(n - 0.5)T \rightarrow (n)T$  ( $\phi_2 \rightarrow \phi_1$ ), Eq. 5.14.1 is obtained from adding all the capacitors that are connected to the virtual ground at the end of that transition ( $\phi_2$  in the first case and  $\phi_1$  in the second).

#### VERY IMPORTANT QUESTIONS

*Following questions are very important. These questions may be asked in your SESSIONALS as well as UNIVERSITY EXAMINATION.*

**Q. 1. Explain working of weighted resistor D/A converter.**

**Ans.** Refer Q. 5.1, Page 5-2B, Unit-5.

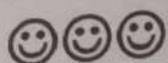
Q. 2. What is a DAC ? Describe the weighted resistor DAC. Give mathematical expressions in support of your answer.  
Ans. Refer Q. 5.3, Page 5-3B, Unit-5.

Q. 3. Write short note on analog to digital converter (ADC).  
Ans. Refer Q. 5.6, Page 5-8B, Unit-5.

Q. 4. Explain the working of single slope ADC.  
Ans. Refer Q. 5.7, Page 5-9B, Unit-5.

Q. 5. Explain the working of dual slope integrating ADC with the help of circuit diagram.  
Ans. Refer Q. 5.8, Page 5-10B, Unit-5.

Q. 6. Draw and explain the flash type A/D converter. Also discuss the corresponding digital output with respect to input signal voltage.  
Ans. Refer Q. 5.10, Page 5-13B, Unit-5.





## Logic Simplification and Combinational Logic Design (2 Marks Questions)

**1.1. Define combinational circuits.**

**Ans:** It consists of input variables, logic gates and output variables. Logic gates accept signal from input variable and generate output signals. This process transforms binary information from given input data to the required output data.

**1.2. Write the difference between combinational and sequential circuits.**

AKTU 2017-18, Marks 02

**Ans:**

S. No.	Combinational circuits	Sequential circuits.
1.	It consists of interconnection of logic gates only.	It consists of storage elements and logic gates.
2.	Output of combinational circuits depends only on the present value of input.	Output of sequential circuit depends on present and previous value of input and output.

**1.3. In how many ways the binary codes are classified ?**

**Ans:**

1. Weighted codes
2. Non-weighted codes
3. Reflective codes
4. Sequential codes
5. Alphanumeric codes
6. Error detecting and correcting codes.

**1.4. Define cyclic codes.**

AKTU 2018-19, Marks 02

**Ans:** When a bit pattern of two consecutive numbers differ by only one bit position, these codes are called cyclic codes.

1.5. Write the advantages of Gray code over the straight binary number sequence.

AKTU 2016-17, 2018-19; Marks 02

- Ans.**
1. The Gray code is used in applications in which the normal sequence of binary numbers generated by the hardware may produce an error or ambiguity during the transition from one number to the next. The Gray code eliminates this problem, since only one bit change its value during any transition between two numbers.
  2. Gray code represents analog data by a continuous change in the angular position of a shaft. Gray code eliminates ambiguity between the angle of the shaft and the value encoded by the sensor.

1.6. Convert  $(153.513)_{10}$  to an octal number ?

AKTU 2016-17, Marks 02

**Ans.**

$(153)_{10}$  to octal :       $(0.513)_{10}$  to octal :

8	153	1	0.513 × 8 = 4.104
8	19	3	0.104 × 8 = 0.832
8	2	2	0.832 × 8 = 6.656
	0		0.656 × 8 = 5.248
			0.248 × 8 = 1.984
			0.984 × 8 = 7.872

$$(153)_{10} = (231)_8, (0.513)_{10} = (0.406517\ldots)_8 \\ \text{So, } (153.513)_{10} = (231.406517)_8$$

1.7. Write four advantages of digital systems over analog system.

AKTU 2017-18, Marks 02

**Ans.**

1. Digital communication system is more robust than analog system because it can resist the corruption of signal much better in presence of channel noise.
2. Implementation of digital hardware in digital communication system is flexible and permits the use of microprocessor, digital switching etc.
3. In digital system, it is possible to multiplex several digital signals that offer more efficient use of available bandwidth.
4. In digital system, it is easy to store large quantities of information.

1.8. Convert the following expression into canonical POS form

$$Y = (A + B)(B + C)(A + C)$$

**Ans.** Given,

$$Y = (A + B)(B + C)(A + C)$$

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

Canonical form,

$$\begin{aligned}
 Y &= (A+B+C)(A+B+\bar{C})(A+B+C)(\bar{A}+B+C)(A+B+C)(A+\bar{B}+C) \\
 &= (A+B+C)(A+B+\bar{C})(\bar{A}+B+C)(A+\bar{B}+C)
 \end{aligned}$$

- 1.9.** Convert the binary number  $(110111)_2$  into Gray code.  
**Ans:**

Binary code	1	1	0	1	1	1
Gray code	1	0	1	1	0	0

Hence, Gray code is (101100).

- 1.10.** Convert gray code 1001011 to binary.  
**Ans:**

Gray code	1	0	0	1	0	1	1
Binary code	1	1	1	0	0	1	0

Hence, Binary code is (1110010).

- 1.11.** Convert the decimal number 32.57 in octal, binary, hexadecimal and Gray.

AKTU 2015-16, Marks 02

**Ans:**

For octal :

8	32	0
8	4	4
0		

$$32 = (40)_8$$

$$\text{and } .57 \times 8 = 4.56$$

$$.56 \times 8 = 4.48$$

$$.48 \times 8 = 3.84$$

$$\text{and } .57 \times 16 = 9.12$$

$$.12 \times 16 = 1.92$$

$$.92 \times 16 = 14.72$$

$$\text{Hence, } (32.57)_{10} = (40.443)_8$$

$$\text{Hence, } (32.57)_{10} = (20.91E)_{16}$$

For hexadecimal :

16	32	0
16	2	2
0		

$$32 = (20)_{16}$$

$$4$$

$$4$$

$$3$$

$$9$$

$$1$$

$$E$$

2018 (EC-Sem-3)

For Gray :

For binary :

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

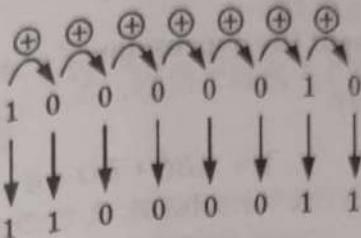
$$32 = (100000)_2$$

and  $.57 = .57 \times 2 = 1.14$

$$0.14 \times 2 = 0.28$$

$$\text{Hence, } (32.57)_{10} = (100000.10)_2$$

$$\text{Hence, } (32.57)_{10} = (110000.11)_{\text{Gray}}$$



- 1.12. Simplify the following boolean expression to a minimum number of literals :  $(x'y' + z')' + z + xy + wz$

AKTU 2015-16, Marks 02

Given,  $(x'y' + z')' + z + xy + wz$   
Using De-Morgan's theorem

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

$$\begin{aligned} (x'y' + z')' + z + xy + wz &= (x + y)z + z + xy + wz \\ &= z[1 + (x + y)] + xy + wz = z + xy + wz \\ &= z[1 + w] + xy = z + xy \end{aligned}$$

- 1.13. Convert the following :

i.  $(562.13)_7 = (?)_{10}$

AKTU 2017-18, Marks 02

ii.  $(467.342)_8 = (?)_{10}$

$$\begin{aligned} \text{i. } (562.13)_7 &= 5 \times 7^2 + 6 \times 7^1 + 2 \times 7^0 + 1 \times 7^{-1} + 3 \times 7^{-2} \\ &= (289.20)_{10} \end{aligned}$$

$$\begin{aligned} \text{ii. } (467.342)_8 &= 4 \times 8^2 + 6 \times 8^1 + 7 \times 8^0 + 3 \times 8^{-1} + 4 \times 8^{-2} + 2 \times 8^{-3} \\ &= (311.441)_{10} \end{aligned}$$

- 1.14. Determine the value of base  $x$ , if  $(193)_x = (623)_8$ .

AKTU 2018-19, Marks 02

Given  $(193)_x = (623)_8$

$$1 \times x^2 + 9 \times x + 3 \times x^0 = 6 \times 8^2 + 2 \times 8^1 + 3 \times 8^0$$

$$x^2 + 9x + 3 = 403$$

$$x^2 + 9x - 400 = 0$$

$$x^2 + 25x - 16x - 400 = 0$$

$$(x + 25)(x - 16) = 0$$

$$x \neq -25$$

$$x = 16$$

(∴ Base can't be negative)

1.15. Implement the expression  $Y = AB\bar{C} + BD + E$  using NAND gate only.

AKTU 2017-18, Marks 02

**Ans.**

There are five variables,  $A, B, C, D$  and  $E$

$$Y = \overline{\overline{Y}} = \overline{\overline{ABC} + BD + E} = \overline{\overline{ABC} \cdot \overline{BD} \cdot \overline{E}}$$

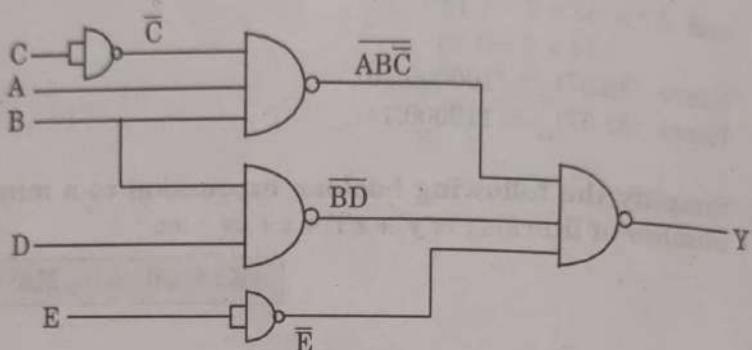


Fig. 1.15.1.

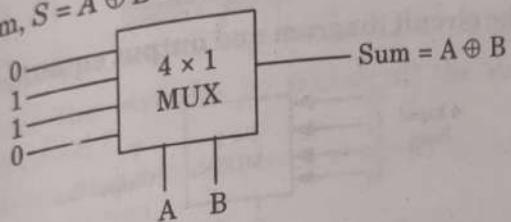


## MSI Devices (2 Marks Questions)

21. Design a half adder using multiplexer.

AKTU 2015-16, Marks 02

**Ans:** For sum,  $S = A \oplus B = \bar{A}B + A\bar{B}$



For carry,  $C = AB$

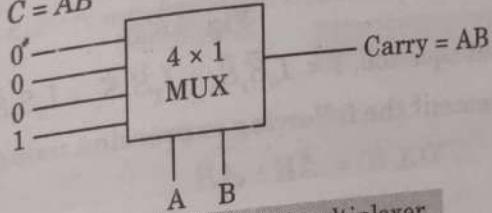


Fig. 2.1.1. Half adder using multiplexer.

22. Give the expression of full adder output.

**Ans:** Sum output,  $S = ABC + A\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} = A \oplus B \oplus C$   
Carry output,  $C_0 = AB + AC + BC$

23. Draw the logic diagram of half subtractor.

AKTU 2016-17, Marks 02

**Ans:**

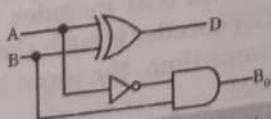


Fig. 2.3.1. Half subtractor.

24. Explain multiplexer circuit.

**Ans:** It is a combinational circuit that selects binary information from one of many lines and directs it to a single output line. The selection of particular input line is controlled by a set of selection lines. There

are  $2^n$  input lines and  $n$  select lines whose bit combination determines which input is selected.

- 2.5. Implement the following function using  $8 \times 1$  MUX,  
 $Y(A, B, C) = \sum m(0, 1, 3, 5, 7)$ .

Ans:

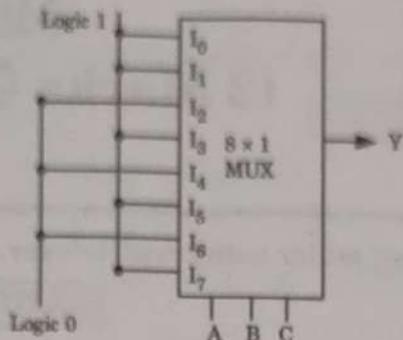


Fig. 2.5.1.

- 2.6. Give the circuit diagram and output equation for  $4 \times 1$  MUX.

Ans:

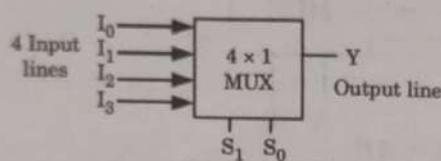


Fig. 2.6.1.

$$\text{Output equation, } Y = I_0\bar{S}_1\bar{S}_0 + I_1\bar{S}_1S_0 + I_2S_1\bar{S}_0 + I_3S_1S_0.$$

- 2.7. Implement the following expression using  $4 : 1$  MUX.

$$Y(A, B) = \bar{A}\bar{B} + A\bar{B}$$

Ans:

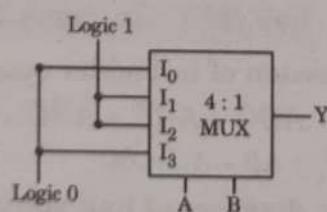


Fig. 2.7.1.

- 2.8. What do you mean by priority encoder?

Ans: It is an encoder circuit that includes the priority function. The operation of priority encoder is such that, if two or more inputs are equal to 1 at the same time, the input having the highest priority will take precedence.

- 2.9. Which circuit is used at input of communication system?

Ans: Encoder is used at input of communication system.

- 2.10. Specify the purpose of valid bit indicator in priority encoder.

AKTU 2016-17, Marks 02

Q.S.B (EC-Sem-3)

In priority encoder, valid bit indicator is set to 1, when one or more inputs are equal to 1. If all inputs are 0, there is no valid input and valid bit indicator is equal to 0. The other two outputs are not inspected when valid bit indicator equals to 0 and are specified as don't care conditions.

Table 2.10.1. Truth table of a priority encoder.

Inputs				Outputs		
$D_0$	$D_1$	$D_2$	$D_3$	$x$	$y$	$v$
0	0	0	0	x	x	0
1	0	0	0	0	0	1
x	1	0	0	0	1	1
x	x	1	0	1	0	1
x	x	x	1	1	1	1

2.11. What are the circuits by which all the expressions are implemented?

Multiplexer and decoder, both are used for implementing all boolean expression.

2.12. Write the difference between decoder and demultiplexer.

AKTU 2015-16, Marks 02

S. No.	Decoder	Demultiplexer
1.	It has $n$ input line.	It has 1 input line.
2.	Enable line is present.	There is no enable line.
3.	There is no select line.	Select line is present.

2.13. What is  $(33)_6 + (45)_6$

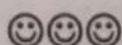
AKTU 2017-18, Marks 02

$$\begin{aligned}
 (33)_6 &= 3 \times 6^1 + 3 \times 6^0 = (21)_{10} \\
 + (45)_6 &= 4 \times 6^1 + 5 \times 6^0 = (29)_{10} \\
 &\hline
 && (50)_{10}
 \end{aligned}$$

Now,

$$\begin{array}{r|rr}
 6 & 50 \\
 \hline
 6 & 8 & 2 \\
 6 & 1 & 2 \\
 \hline
 0 & 1
 \end{array}$$

$$\text{So, } (33)_6 + (45)_6 = (50)_{10} = (122)_6$$





## Sequential Logic Design (2 Marks Questions)

**3.1. Define sequential circuits.**

**Ans:** Sequential circuits consist of a combinational circuit to which storage elements are connected to form a feedback path.

**3.2. What do you mean by flip-flops ?**

**Ans:** Storage elements that are controlled by a clock transition are flip-flops. It is a binary storage device capable of storing one bit of information. Flip-flops are edge triggered devices.

**3.3. Give the major differences between latch and flip-flop.**

**Ans.**

S.No.	Latch	Flip-flop
1.	Storage element that operates with signal levels.	Storage element that are controlled by clock transition.
2.	It is level triggered.	It is edge triggered.
3.	There is no clock pulse.	There is a clock pulse.

**3.4. Give the function table of SR latch.**

AKTU 2016-17, Marks 02

**Ans.**

$S$	$R$	$Q_{n+1}$
0	0	$Q_n$ (Preset state)
0	1	0
1	0	1
1	1	x

**3.5. Draw Master-Slave flip-flop.**

AKTU 2015-16, Marks 02

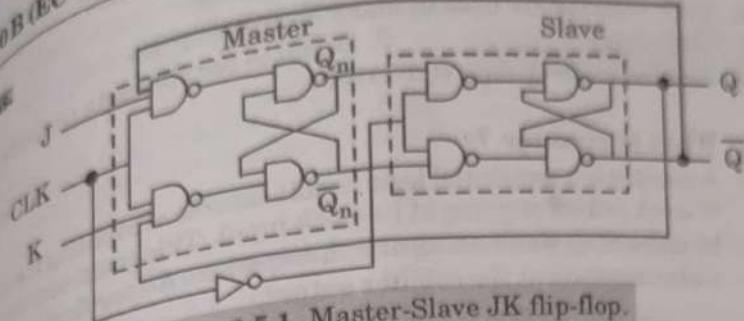


Fig. 3.5.1. Master-Slave JK flip-flop.

Ques. Express the characteristic equation for the JK flip-flop.

AKTU 2016-17, Marks 02

OR

Write the excitation table and characteristic equation of JK flip-flop.

AKTU 2017-18, Marks 02

~~JK flip-flop :  $Q_{n+1} = J\bar{Q}_n + \bar{K}Q_n$~~

~~Excitation table :~~

Flip-flop inputs		Present state	Next state
J	K	$Q_n$	$Q_{n+1}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Ans. 3.7. Define 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 shift registers.

Ans. 3.8. Give the classification of shift register on the basis of mode of operation.

~~Ans.~~

1. Serial in serial out shift register (SISO)
2. Serial in parallel out shift register (SIPO)
3. Parallel in serial out shift register (PISO)
4. Parallel in parallel out shift register (PIPO)

**3.9. Give the application of shift register.**

**Ans:** Shift register are used in many application as :

1. Time delay
2. Serial to parallel data converter
3. Parallel to serial data converter

**3.10. What is counter ?**

**Ans:** A counter is a sequential logic circuit capable of counting the number of clock pulses arriving at its clock input. This count sequence may be ascending, descending or non sequence. A specified sequence of states appears at the counter output.

**3.11. Define asynchronous counter or ripple counter.**

**Ans:** A binary asynchronous/ripple counter consists of a series connection of complementing flip-flops, with the output of each flip-flop connected to the clock input of the next higher-order flip-flop. The flip-flop holding the least significant bit receives the incoming clock pulses.

**3.12. Define synchronous counter.**

**Ans:** The synchronous counter is one in which all the flip-flops are triggered simultaneously by the clock pulse, so they are also called as parallel counters. Synchronous counter may reduce the delay occurred in asynchronous counter.

**3.13. What is modulus of a counter ?** AKTU 2018-19, Marks 02

**Ans:** The number of states passed by the counter before going to its initial state is called modulus of counter.

**3.14. The contents of a four bit register are initially 1011. The register is shifted six times to the right with serial input being 101111. What are the contents of the register after each shift ?** AKTU 2016-17, Marks 02

**Ans:** Given : Serial input = 101111, initial content = 1011.

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

3.IIB (EC-Sem-3)

Q.15. How many flip-flops are required to design Mod-5 ring counter and Mod-5 Johnson counter ?

AKTU 2018-19, Marks 02

ANS: Mod-5 Ring counter - 5 Flip-flop.  
Mod-5 Johnson counter - 3 Flip-flop.

Q.16. What are the required numbers of flip-flops in a MOD-16 asynchronous counter, MOD-16 synchronous counter, MOD-16 Johnson counter ?

AKTU 2015-16, Marks 02

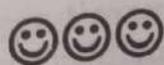
ANS: MOD-16 asynchronous counter  
MOD-16 synchronous counter  
MOD-16 Johnson counter

- : 4 flip-flops required.
- : 4 flip-flops required.
- : 8 flip-flops required.

Q.17. What is race around condition ?

AKTU 2017-18, 2018-19; Marks 02

ANS: A race around condition is said to exist in an asynchronous sequential circuit when two or more binary state variable change value in response to a change in input variable.  
It is eliminated by using master-slave JK flip-flop.



# 4

UNIT

## Logic Families and Semiconductor Memories (2 Marks Questions)

- 4.1. What do you mean by bipolar and unipolar logic families ?**

**Ans.** Bipolar logic family : In this logic family, the current flows because of both electrons and holes being charge carriers.

Unipolar logic family : It includes *p*-channel metal oxide semiconductor field-effect transistor (PMOS), *n*-channel metal oxide field effect transistor (NMOS) and CMOS. The current flows due to any one types of carrier.

- 4.2. What are the uses of ECL family ?**

**Ans.** ECL family is used in very high frequency application where its speed is superior. It is used in superfast computers and high-speed special purpose applications.

- 4.3. What are the difference between CMOS gate and TTL gate ?**

**Ans.** 1. CMOS gate can transmit signal in both direction but TTL and ECL gates are essentially unidirectional.  
2. CMOS consists of *n*-MOS and *p*-MOS but TTL and ECL uses transistors.

- 4.4. What do you mean by fan-out and fan-in ?**

**AKTU 2018-19, Marks 02**

**Ans.** Fan-out : The fan-out of a logic gate is defined as the maximum number of standard load that the output of the gate can drive without impairing its normal operation. Fan-out is also called the loading factor.

$$\text{Fan-out} = \text{Minimum of } \left\{ \frac{I_{OH}}{I_{IH}}, \frac{I_{OL}}{I_{IL}} \right\}$$

Fan-in : The fan-in of a logic gate refers to the number of inputs that the gate is designed to handle.

- 4.5. What do you understand by noise margin ?**

**Ans.** The noise immunity of a logic circuit refers to the circuit's ability to tolerate noise voltages at its inputs. A quantitative measure of noise immunity is called noise margin.

- 4.6. What is memory ?**

**Ans.** A memory unit is a device to which binary information is transferred for storage and from which information is retrieved when needed for processing. A memory is a collection of cells capable of storing a large quantity of binary information.

**Ans:** 4.7. What do you mean by FPGA ?

**Ans:** FPGA (Field Programmable Gate Array) is high capacity PLD. The gate array of FPGA has the ability to be programmed for a function by the user instead of the manufacturer of device.

**Ans:** 4.8. Write the steps that must be taken for the purpose of transferring a new word to be stored into memory.

**AKTU 2016-17, Marks 02**

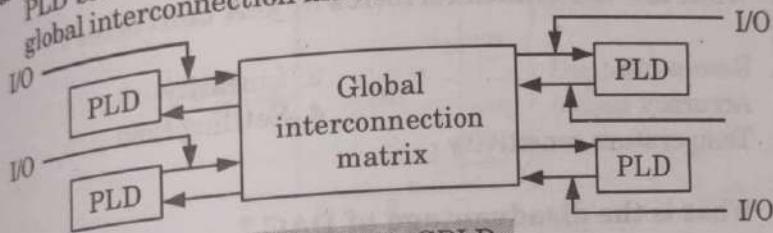
**Ans:** 4.9. The steps that must be taken for the purpose of transferring a new word to be stored into memory are as follows :

1. Apply the binary address of the desired word to the address lines.
2. Apply the data bits that must be stored in memory to the data input lines.

3. Activate the write input.

**Ans:** 4.10. What do you mean by CPLD ?

**Ans:** Complex-programmable logic device (CPLD) contains a bunch of PLD blocks whose input and output are connected together by a global interconnection matrix.



**Fig. 4.9.1. CPLD.**

**Ans:** 4.10. How many address lines and data I/O lines are required for a  $16K \times 12$  memory ?

**AKTU 2015-16, Marks 02**

**Ans:**  $16K \times 12 = 2^4 \times 2^{10} \times 12$   
Address lines = 14, Data lines = 12

**Ans:** 4.11. Define memory read and write operation.

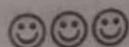
**Ans:** The process of storing new information into memory is referred to as a memory write operation and the process of fetching the stored information from the memory is referred to as a memory read operation.

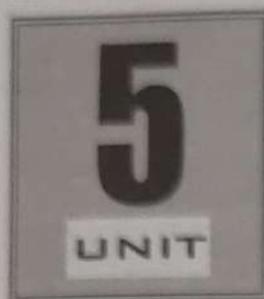
**Ans:** 4.12. What is ROM ?

**Ans:** Read only memory is a type of memory used in digital system which can perform only the read operation. This means that suitable binary information is already stored inside memory and can be retrieved or read anytime. However, that information cannot be altered by writing. ROM is non-volatile memory.

**Ans:** 4.13. Define the random access memory (RAM).

**Ans:** RAM can perform both read and write operations, the time it takes to transfer information to or from any desired random location is always the same, hence, the name random access memory. RAM is volatile memory.





## D/A and A/D Converter (2 Marks Questions)

**5.1. Define D/A conversion.**

**Ans:** It is the process of taking a value represented in digital code and converting it into a voltage or current which is proportional to digital value.

**5.2. What are the characteristics of D/A converter ?**

**Ans:**

- |                            |                  |
|----------------------------|------------------|
| 1. Resolution              | 2. Linearity     |
| 3. Accuracy                | 4. Settling time |
| 5. Temperature sensitivity |                  |

**5.3. What is the disadvantage of DAC ?**

**Ans:** The disadvantage of binary weighted type DAC is the wide range of resistor value required. It may be observe that for better resolution, the input binary word length has to be increased. Thus, as the number of bit increases, the range of resistance value increases.

**5.4. Draw the circuit diagram of A/D and D/A converter.**

**Ans:**

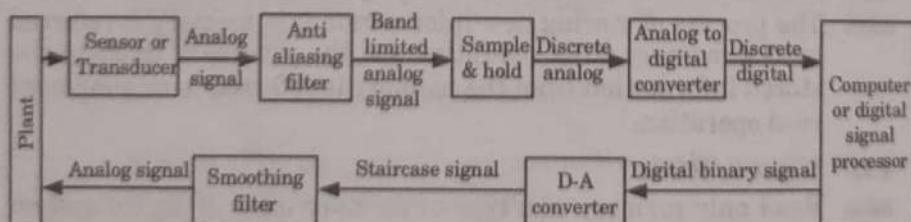


Fig. 5.4.1.

**5.5. Name the different types of ADCs used.**

**Ans:**

1. Flash (comparator) type converter
2. Counter type converter
3. Tracking or servo type converter
4. Successive approximation type comparator

Ques (EC-Sem-3)

56. Write the application in which integrating type converter is used?

- Ans.
1. Digital meter
  2. Panel meter
  3. Monitoring systems where the conversion accuracy is critical.
57. Draw the functional diagram of ADC and define its two additional control lines.

- Ans.
1. The START input to tell the ADC when to start the conversion.
  2. The EOC output to announce when the conversion is complete.

Functional diagram :

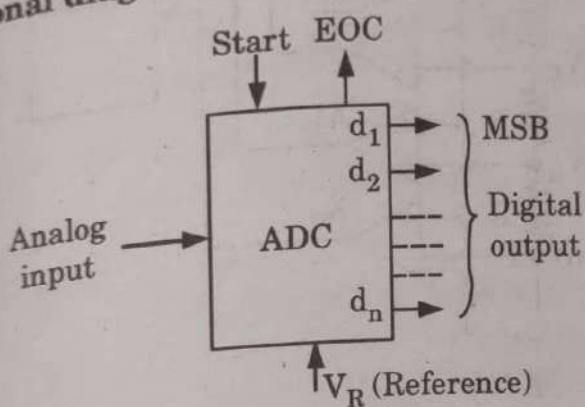


Fig. 5.7.1

58. The basic step of 9-bit DAC is 10.3 mV. If 000000000 represents 0 V, what output is produced if the input is 101101111?

Ans. The output voltage for input 101101111 is

$$10.3 \times 10^{-3} \times (1 \times 2^8 + 0 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0) = 10.3 \times 367 \times 10^{-3} = 3.78 \text{ V}$$

59. The basic step of a 8-bit DAC is 20 mV. If 00000000 represents 0 V, what is represented by the input 10110111?

Ans. The output voltage for input 10110111 is given by

$$20 \text{ mV} (1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0) = 20 \text{ mV} (128 + 32 + 16 + 4 + 2 + 1) = 20 \times 183 = 3660 \text{ mV} = 3.66 \text{ V}$$

- 5.10. Draw the circuit diagram of flash type A/D converter.

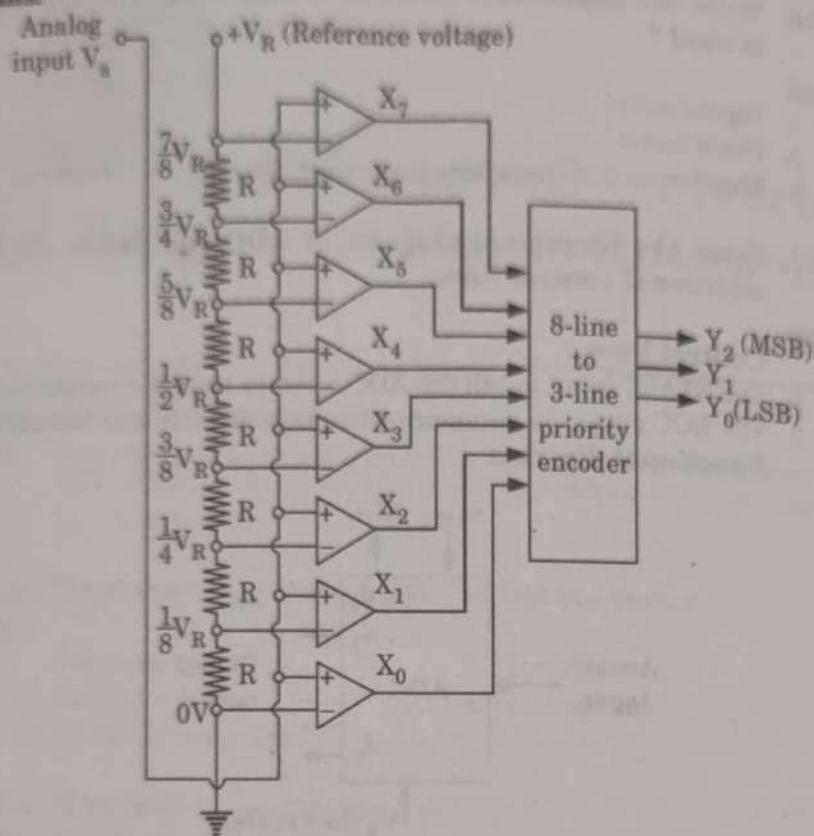
**Ans.**

Fig. 5.10.1.

- 5.11. For a dual slope ADC,  $t_1$  is 83.33 ms and the reference voltage is 100 mV. Calculate  $t_2$  if  $V_i$  is

- 100 mV and
- 200 mV.

**Ans.** Given :  $t_1 = 83.33 \text{ ms}$ ,  $V_R = 100 \text{ mV}$

We know that,

$$V_i = V_R \times \frac{t_2}{t_1} \quad \dots(5.11.1)$$

$$t_2 = \frac{V_i}{V_R} t_1$$

i.  $V_i = 100 \text{ mV}$

$$t_2 = \frac{100}{100} \times 83.33 = 83.33 \text{ ms}$$

ii.  $V_i = 200 \text{ mV}$

$$t_2 = \frac{200}{100} \times 83.33 = 166.66 \text{ ms}$$



B.Tech.

**(SEM. III) ODD SEMESTER THEORY  
EXAMINATION, 2014-15  
SWITCHING THEORY AND LOGIC DESIGN**

Time : 2 Hours

Max. Marks : 50

Notes : 1. Attempt all questions.  
2. All questions carry equal marks.

1. Attempt any four parts of the following : (3.5 × 4 = 14)  
 a. Convert the following numbers into desired base :

- i.  $(A6BF5)_{16} = (?)_2 = (?)_{\text{Gray}}$   
 ii.  $(17 - 135)$  using 2's complement

**Ans:** i.  $(A6BF5)_{16} = (1010011010111110101)_2$   
 $= (11110101111000001111)_{\text{Gray}}$

- ii.  $(17 - 135)$   
 17 in binary form =  $(10001)_2$

135 in binary form =  $(10000111)_2$

2's complement of 135 =  $01111001$

Now, adding 17 in binary form = +  $00010001$

10001010

As there is no carry, the answer is negative and is in the 2's complement form. The answer is,

- (2's complement of 10001010)

$= -(01110110)_2$

Thus the answer is  $-(01110110)_2$ , i.e.,  $(-118)_{10}$ .

- b. Simplify the following boolean expression to a minimum number of literals.

i.  $\bar{A}\bar{C} + ABC + A\bar{C} + A\bar{B}$

ii.  $(\bar{x}\bar{y} + z) + z + xy + wz$

**Ans:** Refer Q. 1.12, Page 1-14B, Unit-1.

- c. Simplify the following expression into product of sum (POS) form

i.  $A\bar{B} + A\bar{B}D + BCD$

ii.  $AC\bar{D} + \bar{C}D + A\bar{B} + ABCD$

**Ans:** Refer Q. 1.17, Page 1-17B, Unit-1.

- d. Use Quine-McCluskey (QM) method to solve the following function :

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

**Ans:** Refer Q. 1.26, Page 1-27B, Unit-1.

- e. Simplify the boolean function  $Y$  together with don't care condition  $d$  using K-map and implement it with two level NAND gate circuit.

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

**Ans:** Refer Q. 1.19, Page 1-19B, Unit-1.

- f. For the Hamming code 1001101001 received at the receiver end, correct this code for error if any ?

**Ans:**

1. Given, Hamming code received = 1001101001

$P_1$	$P_2$	$D_3$	$P_4$	$D_5$	$D_6$	$D_7$	$P_8$	$D_9$	$D_{10}$
1	0	0	1	1	0	1	0	0	1

2. Let code is sent and received on the basis of odd parity. Therefore, we can find the correction bits as follows :

$$C_1 = \text{bit } P_1, D_3, D_5, D_7, D_9 = 1, 0, 1, 1, 0 = 0 \text{ error does not exist.}$$

$$C_2 = \text{bit } P_2, D_3, D_6, D_7, D_{10} = 0, 0, 0, 1, 1 = 1 \text{ error exist.}$$

$$C_4 = \text{bit } P_4, D_5, D_6, D_7 = 1, 1, 0, 1 = 0 \text{ error does not exist.}$$

$$C_8 = \text{bit } P_8, D_9, D_{10}, D_{11} = 0, 0, 1 = 0 \text{ error does not exist.}$$

3. Thus error bit location is  $C_8C_4C_2C_1 = 0010$

4. Thus error is in 2<sup>nd</sup> bit. Thus correct code is 1101101001.

2. Attempt any two parts of the following :  $(6 \times 2 = 12)$

- a. Design a BCD to 7 segment decoder. Assume positive logic, minimize the function.

**Ans:** Refer Q. 2.16, Page 2-17B, Unit-2.

- b. Design the following boolean function using  $4 \times 1$  multiplexer.

$$F(A, B, C, D) = \Sigma m(0, 1, 3, 4, 8, 9, 15)$$

**Ans:** Refer Q. 2.6, Page 2-6B, Unit-2.

- c. Design and explain the logic and circuit of 4-bit magnitude comparator.

**Ans:** Refer Q. 2.2, Page 2-3B, Unit-2.

3. Attempt any two parts of the following :  $(6 \times 2 = 12)$

- a. Distinguish between synchronous and asynchronous digital sequential circuit. Design modulo-5 counter.

### Digital Logic Design

#### Difference :

S.No.	Synchronous sequential circuit	Asynchronous sequential circuit
1.	In synchronous circuit, memory elements are clocked flip-flops.	In asynchronous circuit, memory elements are either unclocked flip-flop or time delay elements.
2.	In synchronous circuit, the change in input signal can affect memory element upon activation of clock signal.	In synchronous circuit, change in input signal can affect memory element at any instant of time
3.	The maximum operating speed of clock depends on time involved.	Because of absence of clock, asynchronous circuit can operate faster than synchronous circuit.
4.	Easier to design.	More difficult to design.

#### Modulo - 5 counter :

Step 1 : Here 5 indicates total number of states, the count sequences are from 0 to 4.

Step 2 : The required number of flip-flops are

$$2^n \geq N$$

$$2^n \geq 5$$

$$n = 3$$

Step 3 : Desired sequence and state diagram :

Desired sequence

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

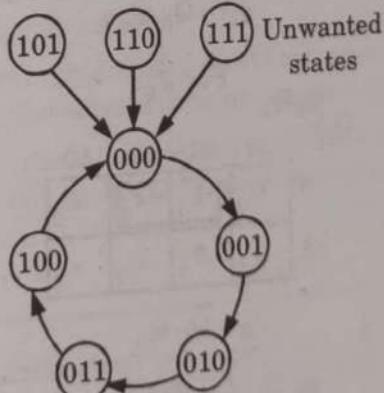


Fig. 1. State diagram.

Step 4 : Excitation table for JK flip-flop :

Present states $Q_n$	Next state $Q_{n+1}$	$J$	$K$
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

## Step 5 : State table :

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

## Step 6 : K-map simplification :

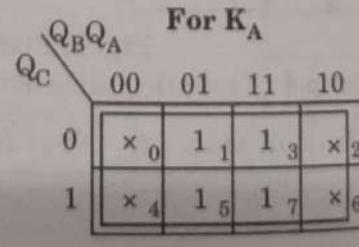
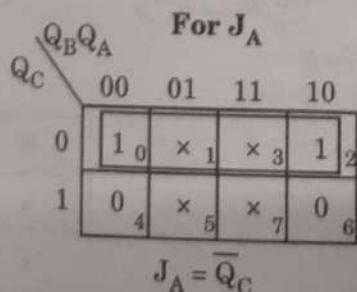
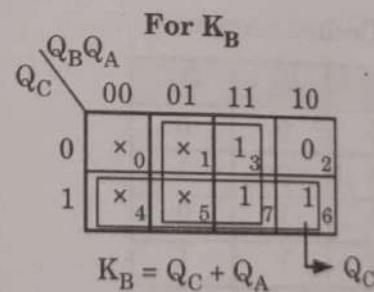
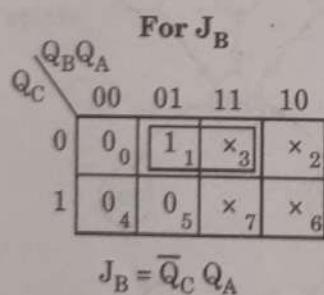
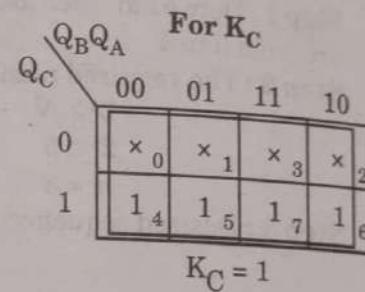
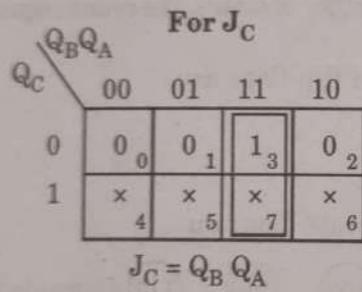


Fig. 2. K-maps and simplifications.

Step 7: The logic diagram of mod-5 counter is shown in Fig. 3.

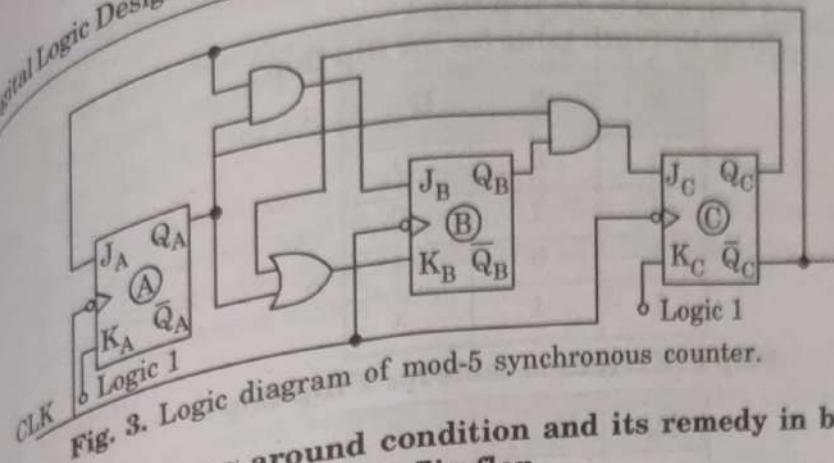


Fig. 3. Logic diagram of mod-5 synchronous counter.

b. Explain race around condition and its remedy in brief.  
Realise T flip-flop to SR flip-flop.

**Ans. Race around condition :**

1. A race around condition is said to exist in an asynchronous sequential circuit when two or more binary state variables change value in response to a change in an input variable.
2. When unequal delays are encountered, a race condition may cause the state variables to change in an unpredictable manner.
3. It occurs in JK flip-flop when the  $J = K = 1$  and propagation delay of the flip-flop is less than pulse width of clock.

$$J = K = 1 \\ \text{i.e.,} \\ \text{and} \quad t_{pdFF} < t_{PW}$$

**Elimination of race around condition using master-slave JK flip-flop :**

1. The master-slave combination consists of clocked JK flip-flop as a master and clocked SR flip-flop as a slave.
2. In this, master is positive level triggered but due to the presence of the inverter in the clock line, the slave will be at negative level.
3. Hence, when clock ( $CLK = 1$ ), the master is active and the slave is inactive whereas when clock ( $CLK = 0$ ), the slave is active and the master is inactive.

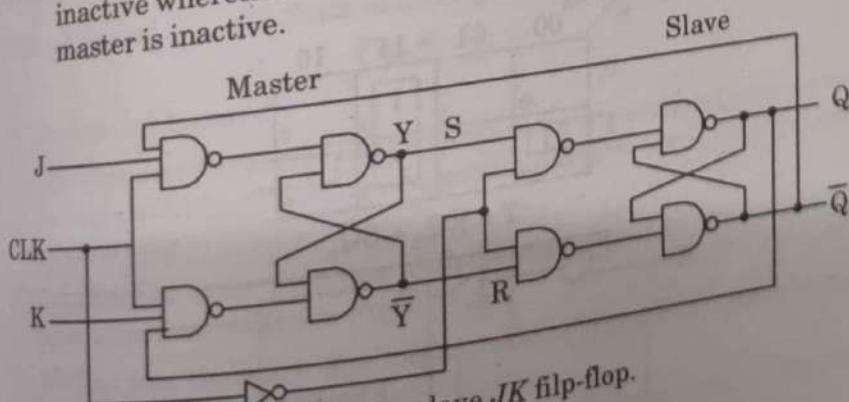


Fig. 4. Master-slave JK flip-flop.

**T flip-flop to SR flip-flop :**

**Step 1 : Truth table for SR flip flop :**

S	R	$Q_{n+1}$
0	0	$Q_n$
0	1	0
1	0	1
1	1	x

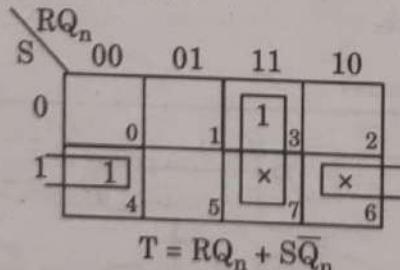
**Step 2 : Excitation table for T flip-flop :**

$Q_n$	$Q_{n+1}$	T
0	0	0
0	1	1
1	0	1
1	1	0

**Step 3 : Conversion table :**

Inputs		Present state	Next state	Flip-flop input
S	R	$Q_n$	$Q_{n+1}$	T
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	0
1	1	0	x	x
1	1	1	x	x

**K-map simplification :**



$$T = RQ_n + SQ_n$$

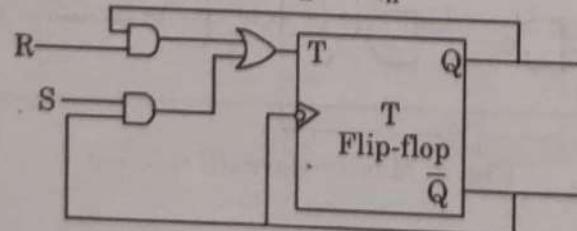


Fig. 5.

c. Write down the classification of semiconductor memories.  
d. Draw and explain the programmable logic array (PLA).  
Refer Q. 4.18, Page 4-21B, Unit-4.

4. Attempt any two parts of the following : (6 x 2 = 12)  
a. Explain hazard and its types. Define critical race and non-critical race. Also explain the elimination of hazards in asynchronous circuits.

- i. Hazards : Hazards are unwanted switching transient that may appear at the output of a circuit because different paths exhibit different propagation delays.  
ii. Hazards occur in combinational circuits, where they may cause a temporary false-output value. When this condition occurs in asynchronous sequential circuits, it may result in a transition to a wrong stable state.

#### Types of hazards :

- i. Static-1 hazard : In response to an input change and for some combination of propagation delays, a logic circuit may go to 0, when it should remain constant 1, this transient is called static-1 hazard.  
ii. Static-0 hazard : In response to an input change and for some combination of propagation delays, a logic circuit may go to 1 when it should remain constant at 0, this transient is called static-0 hazard.  
iii. Dynamic hazard : When the output of logic circuit is changed from 0 to 1 to 0 or 1 to 0 to 1. These two outputs may change more number of times, this transient is called dynamic hazard.

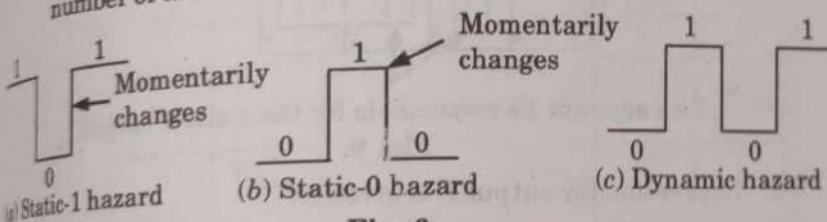


Fig. 6.

#### iv. Essential hazard :

1. The static and dynamic hazards can occur in combinational as well as sequential logic circuits. Essential hazards occur in sequential circuits only.
2. Let there be more than one path from the input to the output of a logic circuit as shown in Fig. 7.
3. We find that there are two output paths that contain combinational logic gates and sequential logic circuit.
4. It may happen that certain paths may produce more delay than the other.

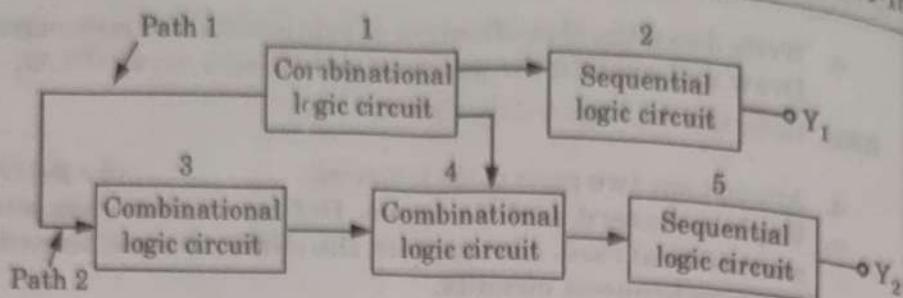


Fig. 7.

**Elimination of hazards :**

1. Static and dynamic hazard can be prevented by adding extra gates in the circuit as the redundant term. This is done by grouping the two adjacent 1's or 0's which are responsible for hazard.
2. Fig. 8 shows the logic circuit with hazard problem.

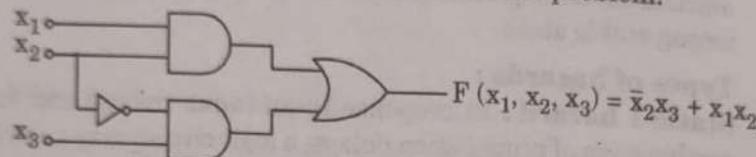
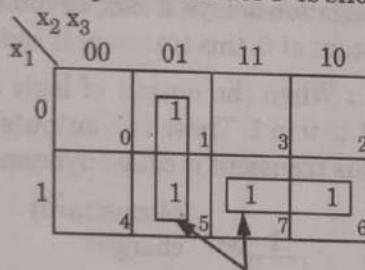


Fig. 8.

3. The K-map simplification for  $F$  is shown in Fig. 9.



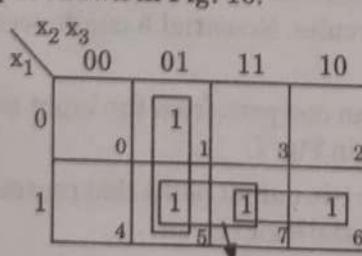
Two adjacent 1's responsible for the static-1 hazard.

Fig. 9.

4. Expression for output  $F$  is given by

$$F = \bar{x}_2x_3 + x_1x_2$$

5. Then we group the two 1's to eliminate static-1 hazard. The modified K-map is shown in Fig. 10.



new pair of 1's ( $x_1 x_3$ ) formed  
to eliminate static-1 hazard.

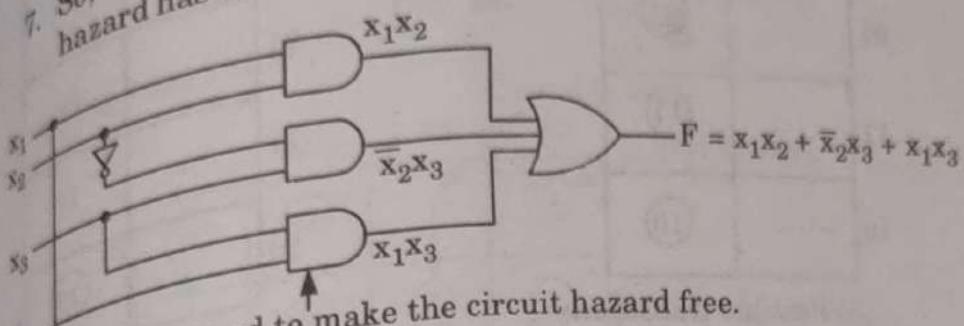
Fig. 10.

6. The new expression for output will be

$$F = \bar{x}_2x_3 + x_1x_2 + x_1x_3$$

new term

7. So, the hazard free circuit is shown in Fig. 11. Note that static-1 hazard has been eliminated due to the additional AND gate.



Additional gate used to make the circuit hazard free.

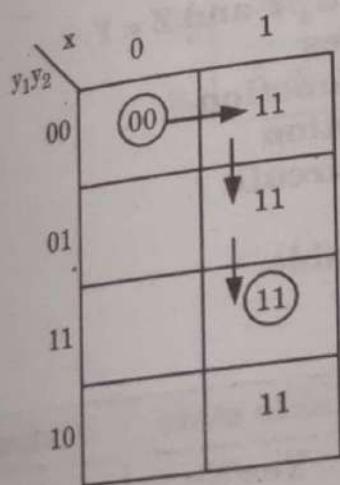
Fig. 11.

### Critical race and non-critical race :

1. If the final stable state that the circuit reaches does not depend on the order in which the state variable changes, the race is called a non-critical race.

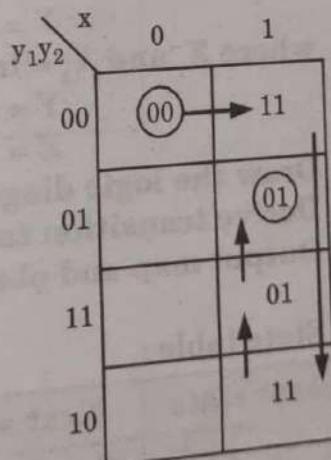
2. If it is possible to end up in two or more different stable states, depending on the order in which the state variable change, then it is a critical race. For proper operation, critical race must be avoided.

### 3. Non-critical race :



(a) Possible transitions

$$\begin{aligned} 00 &\rightarrow 11 \\ 00 &\rightarrow 11 \rightarrow 11 \\ 00 &\rightarrow 10 \rightarrow 11 \end{aligned}$$



(b) Possible transitions

$$\begin{aligned} 00 &\rightarrow 11 \rightarrow 01 \\ 00 &\rightarrow 01 \\ 00 &\rightarrow 10 \rightarrow 11 \rightarrow 01 \end{aligned}$$

Fig. 12.

#### Critical race t:

$y_1 y_2$	0	1
00	(00)	11
01		(01)
11		(11)
10		(10)

(a) Possible transitions

00 → 11

00 → 01

00 → 10

$y_1 y_2$	0	1
00	(00)	11
01		11
11		11
10		(10)

(b) Possible transitions

00 → 11

00 → 01 → 11

00 → 10

Fig. 13.

- b. With the help of diagram, explain the operation of universal shift register.

**Ans:** Refer Q. 3.21, Page 3-28B, Unit-3

- c. An asynchronous sequential circuit described by the following excitation and output functions.

$$Y = X_1 X_2 + (X_1 + X_2) Y \text{ and } Z = V$$

where  $X_1$  and  $X_2$  = input variables

$\gamma$  = excitation function

$Z$  = output function

- Draw the logic diagram of the circuit.
  - Derive transition table.
  - Output map and obtain a flow table.

378

### State table :

Digital Logic Design

i. Logic diagram :

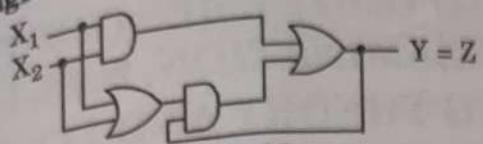


Fig. 14.

ii. Transition table :

	X <sub>1</sub> X <sub>2</sub>	00	01	11	10
Y	0	0	0	1	0
	1	0	1	1	1

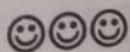
Unstable state  
Stable state

iii. Output map :

	X <sub>1</sub> X <sub>2</sub>	00	01	11	10
Y	0	0	0	1	0
	1	1	1	1	1

Flow table : Let S<sub>0</sub> = 0 and S<sub>1</sub> = 1

	X <sub>1</sub> X <sub>2</sub>	00	01	11	10
Y	S <sub>0</sub>	S <sub>0</sub> /0	S <sub>1</sub> /1	S <sub>0</sub> /0	
S <sub>0</sub>	S <sub>0</sub>	S <sub>0</sub> /0	S <sub>1</sub> /1	S <sub>1</sub> /1	S <sub>1</sub> /1
S <sub>1</sub>	S <sub>1</sub>	S <sub>1</sub> /1	S <sub>1</sub> /1	S <sub>1</sub> /1	S <sub>1</sub> /1



**B.Tech.**  
**(SEM. III) ODD SEMESTER THEORY  
EXAMINATION, 2015-16**

**SWITCHING THEORY AND LOGIC DESIGN**

Time : 3 Hours

Max. Marks : 100

**SECTION - A**

1. Attempt all parts. All parts carry equal marks. Write answer of each part in short :  $(2 \times 10 = 20)$

- a. Write the difference between decoder and demultiplexer.

**Ans.** Refer Q. 2.12, Page SQ-8B, 2 Marks Questions, Unit-2.

- b. Perform the subtraction using 2's complement  $46 - 23$ .

**Ans.**  $(46)_{10} \rightarrow (00101110)_2$   
 $(-23)_{10} \rightarrow$  2's complement of 23

$$\begin{array}{r} = 11101001 \\ 46 = 00101110 \\ -23 = +11101001 \\ \hline \end{array}$$

Neglect carry  $\leftarrow \boxed{1} 00010111$

$$46 - 23 = (00010111)_2 = 23$$

- c. Realize an EX-OR gate using NAND gates only.

**Ans.** EX-OR gate using NAND gates :

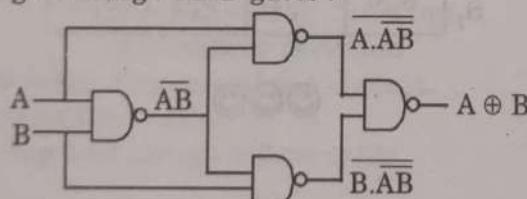


Fig. 1.

- d. How many address lines and data I/O lines are required for a  $16K \times 12$  memory ?

**Ans.** Refer Q. 4.10, Page SQ-14B, 2 Marks Questions, Unit-4.

- e. Simplify the following boolean expression to a minimum number of literals :  $(x'y' + z')' + z + xy + wz$

**Ans.** Refer Q. 1.12, Page SQ-4B, 2 Marks Questions, Unit-1.

- f. Differentiate between asynchronous and synchronous sequential circuits.

**Ans.** Refer Q. 3(a), Page SP-2B, Solved Paper 2014-15.

**a.** What are the required number of flip-flops in a MOD-16 asynchronous counter, MOD-16 Johnson counter ?  
Refer Q. 3.16, Page SQ-12B, 2 Marks Questions, Unit-3.

**b.** Design a half adder using multiplexer.  
Refer Q. 2.1, Page SQ-6B, 2 Marks Questions, Unit-2.

**c.** Convert the decimal number 32.57 in octal, binary, hexadecimal and Gray.  
Refer Q. 1.11, Page SQ-3B, 2 Marks Questions, Unit-1.

**d.** Draw Master-Slave flip-flop.  
Refer Q. 3.5, Page SQ-9B, 2 Marks Questions, Unit-3.

### SECTION - B

Attempt any five questions from this section : (10 × 5 = 50)  
**2.** Design a 4-bit magnitude comparator using one bit comparator modules.  
Refer Q. 2.2, Page 2-3B, Unit-2.

**3.** Prepare Hamming code for the message "01001001010" assuming even parity. Also explain error detection and correction capabilities at the receiver by assuming an error in any one of the received bits.

- Ans:**
- 11-bit data word : 01001001010
  - We include 4 parity bits with the 11-bit word and arrange the 15-bits as follows :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$P_1$	$P_2$	0	$P_4$	1	0	0	$P_8$	1	0	0	1	0	1	0

3. Each parity bit is calculated as follows :

$$\begin{aligned}
 P_1 &= \text{XOR of bits } (3, 5, 7, 9, 11, 13, 15) \\
 &= 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 P_2 &= \text{XOR of bits } (3, 6, 7, 10, 11, 14, 15) \\
 &= 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \\
 &= 1
 \end{aligned}$$

$$\begin{aligned}
 P_4 &= \text{XOR of bits } (5, 6, 7, 12, 13, 14, 15) \\
 &= 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \\
 &= 1
 \end{aligned}$$

$$\begin{aligned}
 P_8 &= \text{XOR of bits } (9, 10, 11, 12, 13, 14, 15) \\
 &= 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \\
 &= 1
 \end{aligned}$$

3. Substituting the 4 parity bits in their proper position, we obtain the 15-bit composite word stored in memory.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	0	1	1	0	0	1	1	0	0	1	0	1	0

When the 15-bits are read from memory they are checked again for error, the parity is checked over the same combinations of bits, including the parity bit.

5. The 4 check bits are evaluated as follows :

**Step 1 :** Analyze bits 1, 3, 5, 7, 9, 11, 13, 15

$$P_1 D_3 D_5 D_7 D_9 D_{11} D_{15} = 00101000 \rightarrow \text{even parity}$$

So no error,  $C_1 = 0$

**Step 2 :** Analyze bits 2, 3, 6, 7, 10, 11, 14, 15

$$P_2 D_3 D_6 D_7 D_{10} D_{11} D_{14} D_{15} = 10000010$$

So no error,  $C_2 = 0$

**Step 3 :** Analyze bits 4, 5, 6, 7, 12, 13, 14, 15

$$P_4 D_5 D_6 D_7 D_{12} D_{13} D_{14} D_{15} = 11001010 \rightarrow \text{even parity}$$

So no error,  $C_4 = 0$

**Step 4 :** Analyze bits 8, 9, 10, 11, 12, 13, 14, 15

$$P_8 D_9 D_{10} D_{11} D_{12} D_{13} D_{14} D_{15} = 11001010 \rightarrow \text{even parity}$$

So no error,  $C_8 = 0$

6. Since the bits were stored with even parity, the result,  $C = 0000$  indicates that no error has occurred.

7. However, if  $C \neq 0$ , then the 4-bit binary number formed by the check bits gives the position of the erroneous bits.

#### Error detection :

- The Hamming code can detect and correct only a single error. By adding another parity bit to the coded word the Hamming code can be used to correct a single error and detect double errors.
- Assume an error occurs at 15<sup>th</sup> bit, so the given coded word becomes 010110011001011.

The 4 check bits are calculated as follows :

**Step 1 :** Analyze bits 1, 3, 5, 7, 9, 11, 13, 15

$$P_1 D_3 D_5 D_7 D_9 D_{11} D_{15} = 00101001 \rightarrow \text{odd parity}$$

So error,  $C_1 = 1$

**Step 2 :** Analyze bits 2, 3, 6, 7, 10, 11, 14, 15

$$P_2 D_3 D_6 D_7 D_{10} D_{11} D_{14} D_{15} = 10000011 \rightarrow \text{odd parity}$$

So error,  $C_2 = 1$

**Step 3 :** Analyze bits 4, 5, 6, 7, 12, 13, 14, 15

$$P_4 D_5 D_6 D_7 D_{12} D_{13} D_{14} D_{15} = 11001011 \rightarrow \text{odd parity}$$

So error,  $C_4 = 1$

**Step 4 :** Analyze bits 8, 9, 10, 11, 12, 13, 14, 15

$$P_8 D_9 D_{10} D_{11} D_{12} D_{13} D_{14} D_{15} = 11001011 \rightarrow \text{odd parity}$$

So error,  $C_8 = 1$

$C = 1111 \Rightarrow$  Error is detected at 15<sup>th</sup> bit. This is the process of error detection in Hamming code.

**Error correction :**

1. The error in a code word is corrected by inverting the incorrect bit.

2. Since the error is detected at 15<sup>th</sup> bit. So we are inverting it from 1 to 0 to obtain correct code word as follows :  
Correct code word = [010110011001010]

4. Using a decoder and external gates, design the combinational circuit defined by the following three boolean functions :

$$F_1 = x'yz' + zx, F_2 = xy'z' + z'y, F_3 = x'y'z' + xy$$

Refer Q. 2.13, Page 2-15B, Unit-2.

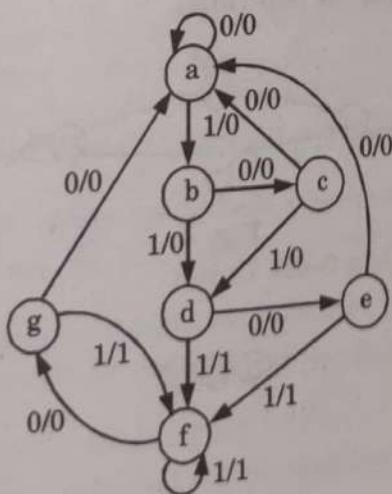
5. Design a 3-bit binary to Gray code converter using PLA.  
Refer Q. 4.24, Page 4-29B, Unit-4.

6. Draw and explain 4-bit universal shift register.  
Refer Q. 3.21, Page 3-28B, Unit-3.

7. Describe the hazards in digital circuits. How are these removed ?

Refer Q. 4(a), Page SP-7B, Solved Paper (2014-15),

8. What do you understand by state reduction ? Reduce the following state diagram.



**Fig. 2.**

Refer Q. 3.9, Page 3-11B, Unit-3.

9. Derive the state table and state diagram for the sequential circuit shown in Fig. 3.

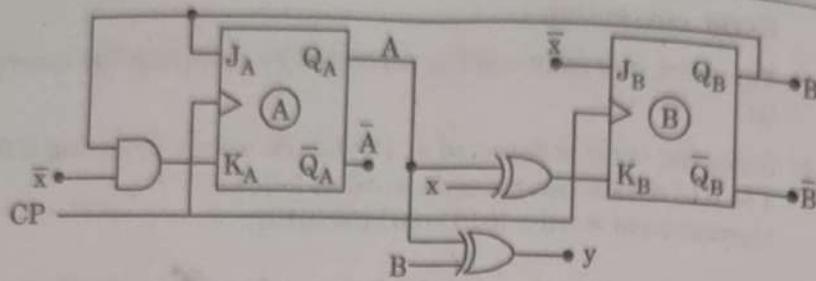


Fig. 3.

**Ans.** Refer Q. 3.8, Page 3-10B, Unit-3.

### SECTION - C

Attempt any two questions from this section.  $(15 \times 2 = 30)$

10. Minimize the following using Quine-McCluskey method :

$$F(W, X, Y, Z) = \Sigma m(0, 3, 5, 6, 7, 10, 12, 13) + \Sigma d(2, 9, 15)$$

**Ans.** Refer Q. 1.25, Page 1-26B, Unit-1.

11. a. Design a 3-bit asynchronous up-down counter using T flip-flop.

**Ans.** Refer Q. 3.16, Page 3-20B, Unit-3.

- b. Design a full adder using two half adders.

**Ans.** Refer Q. 2.18, Page 2-21B, Unit-2.

12. Design the clocked sequential circuit for the following state diagram using JK flip-flops.

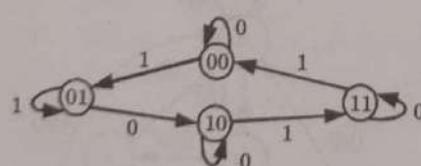
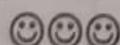


Fig. 4.

**Ans.** Refer Q. 3.7, Page 3-8B, Unit-3.



B.Tech.

(SEM. III) ODD SEMESTER THEORY  
EXAMINATION, 2016-17

SWITCHING THEORY AND LOGIC DESIGN

Time : 3 Hours

Max. Marks : 100

SECTION - A

1. Attempt all parts. All parts carry equal marks. Write answer of each part in short : (2 x 10 = 20)

a. Convert  $(153.513)_{10}$  to an octal number.  
Ans. Refer Q. 1.6, Page SQ-2B, 2 Marks Questions, Unit-1.

b. Write the advantages of Gray code over the straight binary number sequence.  
Ans. Refer Q. 1.5, Page SQ-2B, 2 Marks Questions, Unit-1.

c. Give the general procedure for converting a multilevel AND-OR diagram into an all NAND diagram.  
Ans.

1. Convert all AND gates to NAND gates with AND-invert graphic symbols.

2. Convert all OR gates to NAND gate with invert-OR graphic symbol.

3. Check all bubbles in the diagram. For every bubble that is not compensated by another small circle along the same line, insert an inverter (a one-input NAND gate) or complement the input literal.

d. Draw the logic diagram of half subtractor.

Ans. Refer Q. 2.3, Page SQ-6B, 2 Marks Questions, Unit-2.

e. Specify the purpose of valid bit indicator in priority encoder.

Ans. Refer Q. 2.10, Page SQ-7B, 2 Marks Questions, Unit-2.

f. Give the function table of SR latch.

Ans. Refer Q. 3.4, Page SQ-9B, 2 Marks Questions, Unit-3.

g. Express the characteristic equation for the JK flip-flop.

Ans. Refer Q. 3.6, Page SQ-10B, 2 Marks Questions, Unit-3.

h. Compare Mealy and Moore model of finite state machine.

**Ans.**

S. No.	Mealy model	Moore model
1.	Its output is a function of present input as well as present state.	Its output is the function of present state only.
2.	It requires less number of states for implementing same function.	It requires more number of states for implementing same function.

- i. The contents of a four bit register are initially 1011. The register is shifted six times to the right with serial input being 101111. What are the contents of the register after each shift?

**Ans.** Refer Q. 3.14, Page SQ-11B, 2 Marks Questions, Unit-3.

- j. Write the steps that must be taken for the purpose of transferring a new word to be stored into memory.

**Ans.** Refer Q. 4.8, Page SQ-14B, 2 Marks Questions, Unit-4

### SECTION - B

2. Attempt any five questions from this section :  $(10 \times 5 = 50)$

a Simplify the boolean function.

$$F(w, x, y, z) = \Sigma m(1, 3, 7, 11, 15)$$

which has the don't care conditions

$$d(w, x, y, z) = \Sigma d(0, 2, 5)$$

**Ans.** Refer Q. 1.16, Page 1-16B, Unit-1.

- b. Implement the following boolean function with NAND gate :  $F(x, y, z) = \Sigma(1, 2, 3, 4, 5, 7)$

**Ans.** Refer Q. 1.18, Page 1-19B, Unit-1.

- c. Design a full subtractor circuit with three inputs  $x, y, B_{in}$  and two outputs Diff and  $B_{out}$ . The circuit subtracts  $x-y-B_{in}$ , where  $B_{in}$  is the input borrow,  $B_{out}$  is the output borrow and Diff is the difference.

**Ans.** Refer Q. 2.21, Page 2-24B, Unit-2.

- d. Draw the logic diagram of a two to four line decoder using NOR gates only.

**Ans.** Refer Q. 2.14, Page 2-16B, Unit-2.

- e. Construct a JK flip-flop, using a D flip-flop, a two to four one line multiplexer and an inverter.

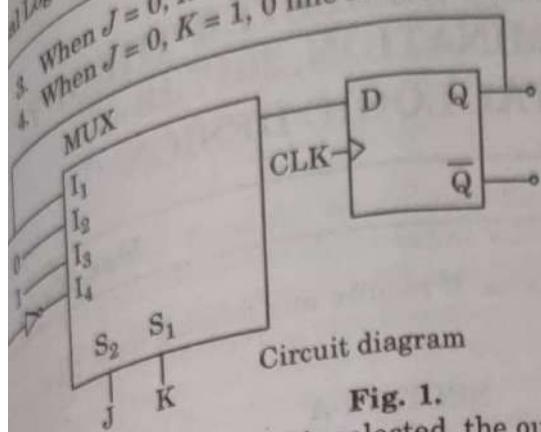
**Ans.** Since a two to four one line multiplexer is not possible hence the solution is given by assuming four to one line multiplexer.

1. The circuit diagram of a JK flip-flop constructed with a D flip-flop, 4:1 MUX and gates is shown in Fig. 1.

2. The J input sets the flip-flop to 1, the K input resets it to 0, and when both inputs are enabled, the output is complemented.

al Logic Design

3. When  $J = 0, K = 0, D = Q = Q_n$   
 4. When  $J = 0, K = 1, 0$  line is selected, the output  $Q = 0$ .



Truth table

$J$	$K$	$Q_{n+1}$
0	0	$Q_n$
0	1	0
1	0	1
1	1	$\bar{Q}_n$

Fig. 1.

5. When  $J = 1, K = 0, 1$  is selected, the output  $Q = 1$ .  
 6. When  $J = 1, K = 1, D = \bar{Q}_n$ .

f. Design a hazard free circuit of the following boolean function  $F(x_1, x_2, x_3) = \Sigma m(1, 5, 6, 7)$   
 Refer Q. 4(a), Page SP-7B, Solved Paper (2014-15).

g. Describe the operation of four bit synchronous binary counter with neat sketch.  
 Refer Q. 3.18, Page 3-24B, Unit-3.

h. Draw the basic configuration of three PLDs.  
 Refer Q. 4.17, Page 4-20B, Unit-4.

### SECTION - C

Note: Attempt any two questions from this section.  $(15 \times 2 = 30)$   
 3. Minimize the following switching function using Quine-McCluskey method.

$$F(x_1, x_2, x_3, x_4, x_5) = \Sigma m(0, 1, 2, 8, 9, 15, 17, 21, 24, 25, 27, 31)$$

Refer Q. 1.24, Page 1-24B, Unit-1.

4. Design a combinational circuit that converts a BCD code to excess-3 code.  
 Refer Q. 1.4, Page 1-6B, Unit-1.

5. Implement the following four boolean functions with a PAL.

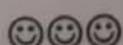
$$W(A, B, C, D) = \Sigma m(2, 12, 13)$$

$$X(A, B, C, D) = \Sigma m(7, 8, 9, 10, 11, 12, 13, 14, 15)$$

$$Y(A, B, C, D) = \Sigma m(0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15)$$

$$Z(A, B, C, D) = \Sigma m(1, 2, 8, 12, 13)$$

Refer Q. 4.22, Page 4-26B, Unit-4.



**B. Tech.**  
**(SEM. III) ODD SEMESTER THEORY  
EXAMINATION, 2017-18**  
**DIGITAL LOGIC DESIGN**

---

Time : 3 Hours

---

Max. Marks : 70

**Note :** Attempt all sections. If require any missing data; then choose suitably.

**SECTION-A**

1. Attempt all questions in brief:

$(2 \times 7 = 14)$

- a. Write four advantages of digital systems over analog system.

**Ans.** Refer Q. 1.7, Page SQ-2B, 2 Marks Questions, Unit-1.

- b. Write the excitation table and characteristic equation of JK flip-flop.

**Ans.** Refer Q. 3.6, Page SQ-10B, 2 Marks Questions Unit-3.

- c. Write the difference between combinational and sequential circuits.

**Ans.** Refer Q. 1.2, Page SQ-1B, 2 Marks Questions, Unit-1.

- d. What is  $(33)_6 + (45)_6$

**Ans.** Refer Q. 2.13, Page SQ-8B, 2 Marks Questions, Unit-2.

- e. Implement the expression  $Y = ABC\bar{C} + BD + E$  using NAND gate only.

**Ans.** Refer Q. 1.15, Page SQ-5B, 2 Marks Questions, Unit-1.

- f. Convert the following :

i.  $(562.13)_7 = (?)_{10}$

ii.  $(467.342)_8 = (?)_{10}$

**Ans.** Refer Q. 1.13, Page SQ-4B, 2 Marks Questions, Unit-1.

- g. What is race around condition ?

**Ans.** Refer Q. 3.17, Page SQ-12B, 2 Marks Questions, Unit-3.

## SECTION-B

- Ans.** 2. Attempt any three of the following :  
 b. Simplify the following Boolean function using K-map  

$$Y = \sum m(0, 1, 3, 5, 6, 7, 9, 11, 16, 18, 19, 20, 21, 22, 24, 26) \quad (7 \times 3 = 21)$$
 Refer Q. 1.21, Page 1-21B, Unit-1.

**Ans.** b. Write the steps for combinational circuit designing and design a circuit of three input which gives an high output whenever the sum of LSB and MSB bit is 1.

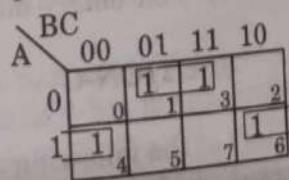
Step for designing combinational logic circuit :

1. From the specifications of the circuits, determine the required number of inputs and outputs and assign a symbol to each.
2. Derive the truth table that defines the required relationship between inputs and outputs.
3. Obtain the simplified boolean functions for each output as a function of the input variables.
4. Draw the logic diagram and verify the correctness of the design (manually or by simulation).

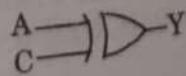
Let the three input be A, B and C, so according to the question, the truth table will be as follows :

LSB	MSB		
A	B	C	Output
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

So, K-map for output



$$Y = A\bar{C} + \bar{A}C = A \oplus C$$



- c. Implement the function  $F = \sum m(0, 1, 3, 4, 7, 8, 9, 11, 14, 15)$  using 8:1 mux.

**Ans.** Refer Q. 2.8, Page 2-8B, Unit-2.

d. Draw and explain the PISO, PIPO register.

**Ans:** Refer Q. 3.20, Page 3-26B, Unit-3.

e. Draw and explain 4-bit by 3-bit multiplier.

**Ans:**

1. Consider a multiplier circuit that multiplies a binary number of 4-bits by a number of 3-bits.
2. Let the multiplicand represented by  $B_3B_2B_1B_0$  and the multiplier by  $A_2A_1A_0$ .
3. Since  $K = 4$  and  $J = 3$ , we need 12 AND gates and two 4-bit adders to produce a product of seven bits.
4. The logic diagram of 4-bit by 3-bit multiplier is shown in Fig. 1.

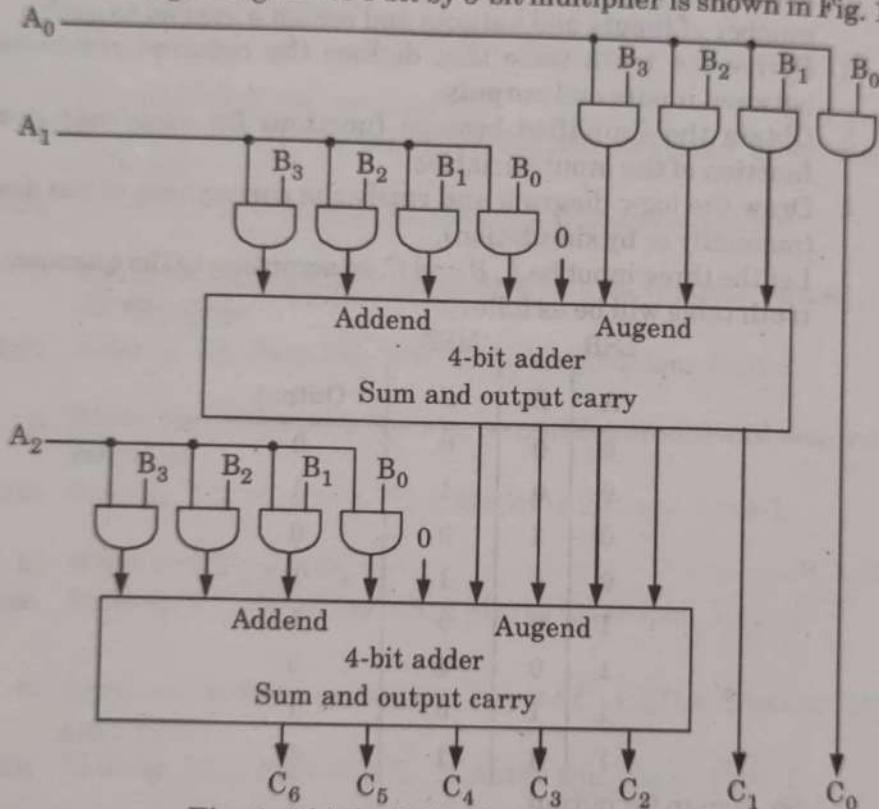


Fig. 1. 4-bit by 3-bit binary multiplier.

### SECTION-C

3. Attempt any one part of the following :  $(7 \times 1 = 7)$

a. Design a universal shift register that performs HOLD, SHIFT RIGHT, SHIFT LEFT, and LOAD.

**Ans:** Refer Q. 3.21, Page 3-28B, Unit-3.

b. Generate the Hamming code for the word 11011. Assume that a single error occurs while storing the generated Hamming code. Explain how this single error is detected.

- Ans.** 1. Given data : 11011, it is a 5-bit data word  
Required parity bit,  $k$  is given by  
 $n \leq 2^k - k - 1$   
 $5 \leq 2^4 - 4 - 1$   
 $k = 4$

2. Bit position  
Hence, 4 parity bits are required for 5-bit message.

1	2	3	4	5	6	7	8	9
$P_1$	$P_2$	1	$P_4$	1	0	1	$P_8$	1

3. Parity bit is calculated as follow :
- $$P_1 = 1 \oplus 1 \oplus 1 \oplus 1 = 0 \quad \text{(XOR of bits 3, 5, 7, 9)}$$
- $$P_2 = 1 \oplus 0 \oplus 1 = 0 \quad \text{(XOR of bits 3, 6, 7)}$$
- $$P_4 = 1 \oplus 0 \oplus 1 = 0 \quad \text{(XOR of bits 5, 6, 7)}$$
- $$P_8 = 1 \quad \text{(XOR of bits 9)}$$

3. So, Hamming code for the data word 11011 will be,  
Code : 001010111

4. Assume a single error generated while storing code, let received code be 001110111

5. **Error correction** : The check bits are determined as  
Step 1 : Analyze bits 1, 3, 5, 7, and 9

$$P_1 D_3 D_5 D_7 D_9 = 0 1 1 1 1 \rightarrow \text{even parity}$$

So no error,  $C_1 = 0$

- Step 2 : Analyze bits 2, 3, 6, and 7

$$P_2 D_3 D_6 D_7 = 0 1 0 1 \rightarrow \text{even parity}$$

So no error,  $C_2 = 0$

- Step 3 : Analyze bits 4, 5, 6, and 7

$$P_4 D_5 D_6 D_7 = 1 1 0 1 \rightarrow \text{odd parity}$$

So error exist,  $C_4 = 1$

- Step 4 : Analyze bits 8 and 9

$$P_8 D_9 = 1 1 \rightarrow \text{even parity}$$

So no error,  $C_8 = 0$

6. **Error bit location** :  $C_8 C_4 C_2 C_1 = 0100$   
7. Thus, there is error in 4th bit. So, correct code is 001010111.

4. Attempt any **one** part of the following :

( $7 \times 1 = 7$ )

- a. Draw and explain 4-bit magnitude comparator.

**Ans.** Refer Q. 2.2, Page 2-3B, Unit-2.

- b. Draw a decimal adder to add BCD numbers.

**Ans.** Refer Q. 2.24, Page 2-27B, Unit-2.

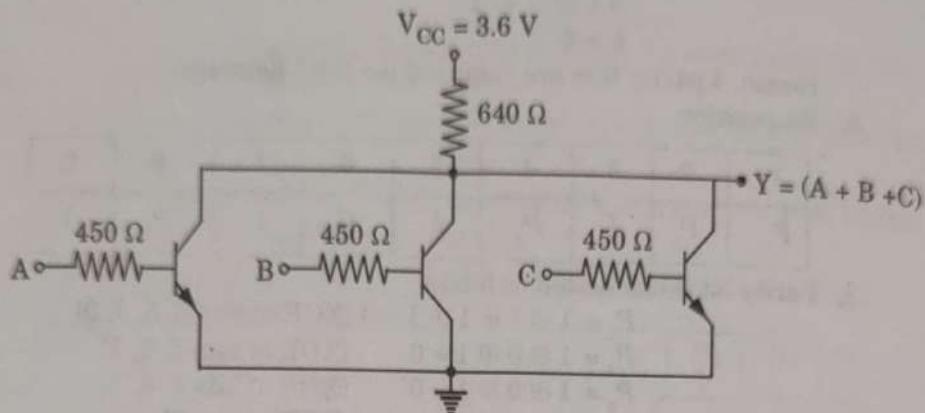
( $7 \times 1 = 7$ )

5. Attempt any **one** part of the following :

- a. Draw and explain the operation of a RTL NOR gate.

**Ans:**

1. The basic circuit of the RTL digital logic family is the NOR gate shown in Fig. 2. Each input is associated with one resistor and one transistor.



**Fig. 2. Basic RTL NOR gate.**

2. The collectors of the transistors are tied together at the output. The voltage levels for the circuit are 0.2 V for the low level and from 1 to 3.6 V for the high level.
3. If any input of the RTL gate is high, the corresponding transistor is driven into saturation and the output goes low, regardless of the states of the other transistors.
4. If all inputs are low at 0.2 V, all transistors are cut-off because  $V_{BE} < 0.6$  V and the output of the circuit goes high, approaching the value of the supply voltage  $V_{CC}$ .

- b. Draw and explain the operation of a TTL NAND gate.

**Ans:** Refer Q. 4.1, Page 4-2B, Unit-4.

6. Attempt any one part of the following :  $(7 \times 1 = 7)$

- a. An asynchronous sequential logic circuit is described by the following excitation and output function

$$\begin{aligned} y &= X_1 X_2 + (X_1 + X_2)Y \\ Z &= y \end{aligned}$$

Draw the logic diagram of the circuit. Also derive the transition table and output map.

**Ans:** Refer Q. 4(c), Page SP-10B, Solved Paper (2014-15).

- b. Design a 3-bit up/down ripple counter.

**Ans:** Refer Q. 3.12, Page 3-16B, Unit-3.

7. Attempt any one part of the following :  $(7 \times 1 = 7)$

- a. Write short notes on RAM and PLA

**Ans. Random access memory (RAM) :**

1. It is a read/write memory that permits the modification of data bits stored in the memory array as well as their retrieval on demand.
  2. The stored data is volatile, i.e., the stored data is lost when the power supply voltage is turned off.
  3. RAM's are classified into two main categories :
- a. **Dynamic RAM** : The DRAM cell consists of a capacitor to store binary information '1' (high voltage) or '0' (low voltage) and a transistor to access the capacitor. Due to the advantage of low cost and high density, DRAM is widely used for main memory.
- b. **Static RAM** : The SRAM cell consists of a latch therefore, the cell data is kept as long as the power is turned on and refresh operation is not required as in case of DRAM cells. SRAM is mainly used for the cache memory in applications.

b. Derive the state table and state diagram of the synchronous sequential circuit shown below (X is an input to the circuit). Explain the circuit function.

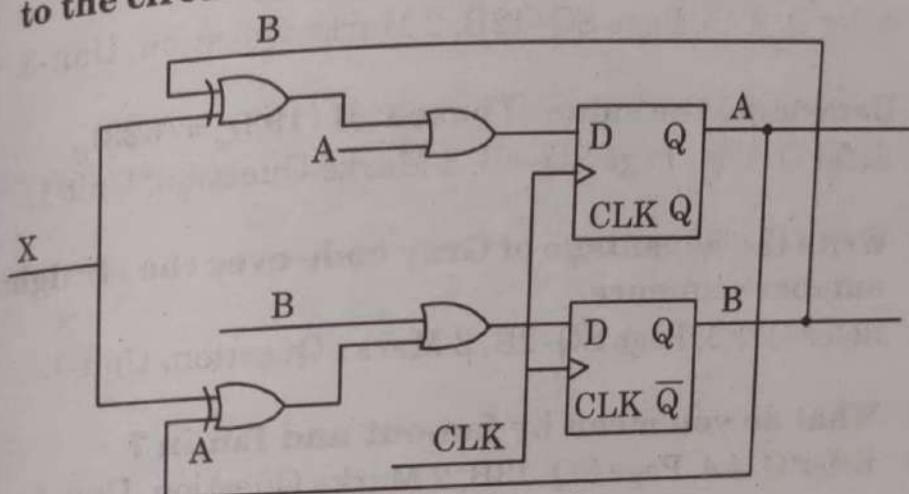
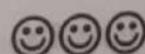


Fig. 3.

**Ans.** Refer Q. 3.11, Page 3-15B, Unit-3.



**B. Tech.**  
**(SEM. III) ODD SEMESTER THEORY**  
**EXAMINATION, 2018-19**  
**DIGITAL LOGIC DESIGN**

---

Time : 3 Hours

---

Max. Marks : 70

**Note :** Be precise in your answer. In case of numerical problem assume data wherever not provided.

**SECTION-A**

1. Attempt all of the following questions :  $(2 \times 7 = 14)$   
 a. What is modulus of a counter ?

**Ans.** Refer Q. 3.13, Page SQ-11B, 2 Marks Question, Unit-3.

- b. How many flip-flops are required to design Mod-5 ring counter and Mod-5 Johnson counter ?

**Ans.** Refer Q. 3.15, Page SQ-12B, 2 Marks Question, Unit-3.

- c. Determine the value of base  $x$ , if  $(193)_x = (623)_8$ .

**Ans.** Refer Q. 1.14, Page SQ-4B, 2 Marks Question, Unit-1.

- d. Write the advantage of Gray code over the straight binary number sequence.

**Ans.** Refer Q. 1.5, Page SQ-2B, 2 Marks Question, Unit-1.

- e. What do you mean by fan-out and fan-in ?

**Ans.** Refer Q. 4.4, Page SQ-13B, 2 Marks Question, Unit-4.

- f. Define cyclic codes.

**Ans.** Refer Q. 1.4, Page SQ-1B, 2 Marks Question, Unit-1.

- g. What is race around condition ?

**Ans.** Refer Q. 3.17, Page SQ-12B, 2 Marks Question, Unit-3.

**SECTION-B**

2. Attempt any three of the following questions :  $(7 \times 3 = 21)$

- a. Minimize the given boolean function using K-map.

$$F(A, B, C, D) = \Sigma m(3, 4, 5, 7, 9, 13, 14, 15)$$

**Ans.** Refer Q. 1.20, Page 1-20B, Unit-1.

- b. Minimize the following using Quine-McCluskey method :

$$F(A, B, C, D) = \Sigma m(0, 1, 9, 15, 24, 29, 30) + \Sigma d(8, 11, 31)$$

**Ans.** Refer Q. 1.28, Page 1-29B, Unit-1.

c. Write a short note on priority encoder.  
Ans. Refer Q. 2.11, Page 2-12B, Unit-2.

d. Implement the following Boolean function.  
Ans.  $F(A, B, C, D) = \Sigma m(0, 1, 3, 4, 7, 8, 9, 11, 14, 15)$

i. 4 : 1 MUX      ii. 2 : 1 MUX  
Ans. Refer Q. 2.9, Page 2-9B, Unit-2.

e. Design Binary code to Gray code converter.  
Ans. Refer Q. 1.6, Page 1-9B, Unit-1.

### SECTION-C

3. Attempt any one part of the following questions :  $(7 \times 1 = 7)$

a. i. Draw a BCD adder circuit and explain its working.

Ans. Refer Q. 2.24, Page 2-27B, Unit-2.

ii. Convert the SR flip-flop to JK flip-flop.

Ans. The JK flip-flop is constructed by using SR flip-flop.

1. Truth table of JK flip-flop

Flip-flop inputs		Present state	Next state
J	K	$Q_n$	$Q_{n+1}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Excitation table of SR flip-flop

Present state	$Q_n$	Flip-flop inputs	
		R	S
0	0	x	0
0	1	0	1
1	0	1	0
1	1	0	x

2. Form the conversion table

Required flip-flop			Given flip-flop		
Flip-flop inputs		Present state	Next state	Flip-flop inputs	
J	K	$Q_n$	$Q_{n+1}$	R	S
0	0	0	0	x	0
0	0	1	1	0	x
0	1	0	0	x	0
0	1	1	0	1	1
1	0	0	1	0	x
1	0	1	1	0	1
1	1	0	1	0	1
1	1	1	0	1	0

## 3. K-map simplification :

		For R			
		00	01	11	10
J	0	x	0	1	1
	1	4	5	1	7

		For S			
		00	01	11	10
J	0	0	x	1	3
	1	1	4	x	5

## 4. The obtained boolean expression :

$$S = J\bar{Q}_n \text{ and } R = KQ_n$$

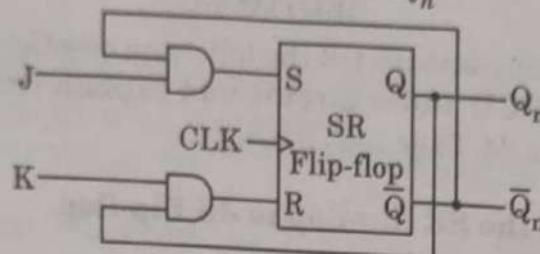


Fig. 1.

b. What do you mean by shift register ? What is the need of shift register ? Draw and explain bidirectional shift register.

**Ans:** Refer Q. 3.19, Page 3-25B, Unit-3.

4. Attempt any one part of the following questions : (7 × 1 = 7)  
a. i. Design a modulo-4 UP/DOWN counter using JK flip-flop.

**Ans:**

1. The count sequences of a modulo-4 up counter are 00, 01, 10 and 11.
  2. The count sequence of a modulo-4 down counter is 11, 10, 01 and 00.
- State diagram :

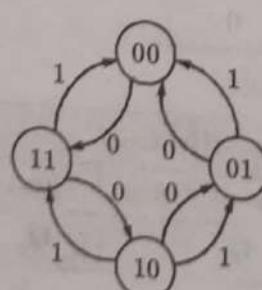


Fig. 2.

3. When the control input is equal to 1, the counter is working as an UP counter. When the control input is equal to 0, the counter is working as a DOWN counter.

State table :

Control input X	Present state		Next state		Flip-flop input			
	$Q_A$	$Q_B$	$Q_{A+1}$	$Q_{B+1}$	$J_A$	$K_A$	$J_B$	$K_B$
0	0	0	1	1	1	x	1	x
0	0	1	0	0	0	x	x	1
0	1	0	0	1	x	1	1	x
0	1	1	1	0	x	0	x	1
0	0	0	0	1	0	x	1	x
1	0	1	1	0	1	x	x	1
1	1	0	1	1	x	0	1	x
1	1	1	0	0	x	1	x	1

K-map simplification :

Expression for  $J_A$

$Q_A Q_B$	00	01	11	10
X	0	1	3	2
0	1	x	x	x
1	4	5	7	6

$$J_A = \overline{X} \overline{Q}_B + XQ_B = X \oplus Q_B$$

Expression for  $J_B$

$Q_A Q_B$	00	01	11	10
X	0	1	3	2
0	1	x	x	1
1	4	5	7	6

$$J_B = 1$$

Logic diagram :

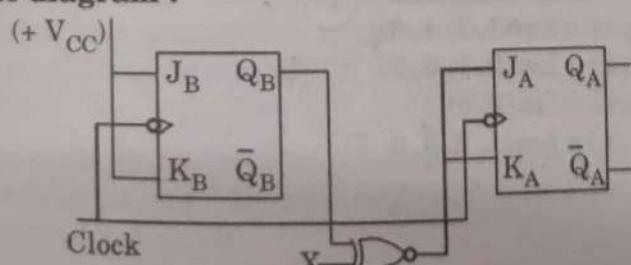


Fig. 3. Modulo-4 binary UP/DOWN counter.

Expression for  $K_A$

$Q_A Q_B$	00	01	11	10
X	0	1	3	2
0	x	1	x	1
1	x	1	x	1

$$K_A = \overline{X} \overline{Q}_B + XQ_B = X \oplus Q_B$$

Expression for  $K_B$

$Q_A Q_B$	00	01	11	10
X	0	1	3	2
0	1	x	x	1
1	1	x	x	1

ii. Design a ripple decade counter using JK flip-flop.

**Ans:** Refer Q. 3.14, Page 3-18B, Unit-3.

b. i. What is critical race and non-critical race ? How can they be avoided ?

**Ans:** Critical and non-critical condition : Refer Q. 4(a), Page SP-7B, Solved Paper (2014-15).

**Method of avoided :**

1. Race may be avoided by making a proper binary assignment to the state variable.
2. The state variable must be assigned binary numbers in such a way that only one state variable can change at any one time when a state transition occurs in the flow table.
3. Race may be avoided by directing the circuit through intermediate unstable state with a unique state variable change. Race around condition is an example of race.

ii. Describe the hazards in digital circuits. How are these removed ? Design a hazards free circuit of the following Boolean function :

$$F(A, B, C) = \Sigma m(1, 2, 3, 5)$$

**Ans:** Refer Q. 4(a), Page SP-7B, Solved paper (2014-15).

5. Attempt any one part of the following questions : (7 × 1 = 7)

a. i. Describe the circuit and performance of CMOS inverter and state the characteristics of CMOS.

**Ans:** Refer Q. 4.10, Page 4-11B, Unit-4.

ii. Differentiate between PLA and PAL. Realize the full adder circuit using PAL.

**Ans:** Refer Q. 4.21, Page 4-25B, Unit-4.

b. i. Discuss the concept of field programmable gate array (FPGA). Discuss the various structures of FPGA.

**Ans:** Refer Q. 4.16, Page 4-18B, Unit-4.

ii. Tabulate the truth table for 8 × 4 ROM that implements the Boolean function :

$$A(x, y, z) = \Sigma m(1, 2, 4, 6)$$

$$B(x, y, z) = \Sigma m(0, 1, 6, 7)$$

$$C(x, y, z) = \Sigma m(2, 6)$$

$$D(x, y, z) = \Sigma m(1, 2, 3, 5, 7)$$

**Ans.** Truth table for the given function :

Inputs			Outputs			
x	y	z	A	B	C	D
0	0	0	0	1	0	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	0	0	1
1	0	0	1	0	0	0
1	0	1	0	0	0	1
1	1	0	1	1	1	0
1	1	1	0	1	0	1

**Logic diagram :**

The ROM is programmed for truth table.

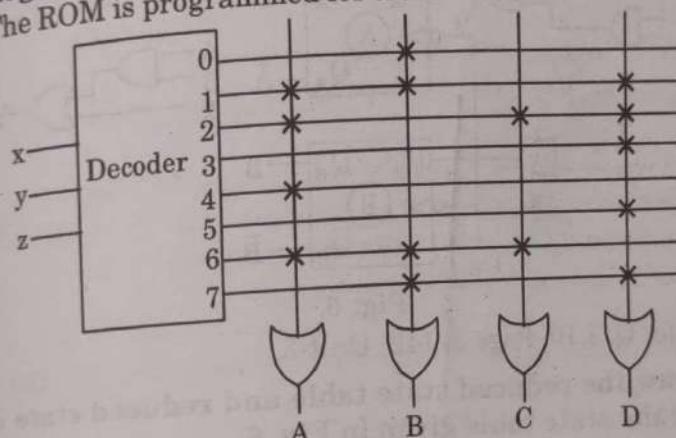


Fig. 4.

6. Attempt any one part of the following questions :  $(7 \times 1 = 7)$
- a. An asynchronous sequential logic circuit is described by the following excitation and output function

$$Y = X_1 X_2 + (X_1 + X_2)Y$$

$$Z = Y$$

- i. Draw the logic diagram of the circuit.
- ii. Derive the transition table and output map.
- iii. Describe the behavior of the circuit.

**Ans.** Refer Q. 4(c), Page SP-10B, Solved Paper (2014-15).

- b.i. The code 101101010 is received, correct any errors. There are four parity bits and odd parity is used.

**Ans.** Received codeword

D <sub>9</sub>	P <sub>8</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	P <sub>4</sub>	D <sub>3</sub>	P <sub>2</sub>	P <sub>1</sub>
1	0	1	1	0	1	0	1	0

**Step 1 :** Analyze bits 1, 3, 5, 7

$$P_1 D_3 D_5 D_7 = 0 0 0 1 \rightarrow \text{odd parity}$$

So there is no error

**Step 2 :** Analyze bits 2, 3, 6, 7

$$P_2 D_3 D_6 D_7 = 1 0 1 1 \rightarrow \text{odd parity}$$

So there is no error.

**Step 3 :** Analyze bits 4, 5, 6, 7

$$P_4 D_5 D_6 D_7 = 1 0 1 1 \rightarrow \text{odd parity}$$

So there is no error.

**Step 4 :** Analyze bits 8, 9, 10, 11, 12, 13, 14, 15

$$P_8 P_9 = 0 1 \rightarrow \text{odd parity}$$

So there is no error.

- ii. Draw a full subtractor circuit using NAND gate.

**Ans.** Refer Q. 2.22, Page 2-26B, Unit-2.

7. Attempt any one part of the following questions : (7 × 1 = 7)

- a. Derive the state table and state diagram for the sequential circuit is shown in Fig. 5.

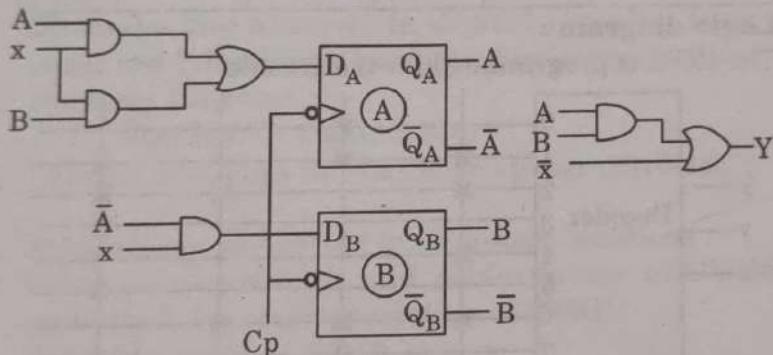


Fig. 5.

**Ans.** Refer Q. 3.10, Page 3-14B, Unit-3.

- b. Draw the reduced state table and reduced state diagram for the state table given in Fig. 6.

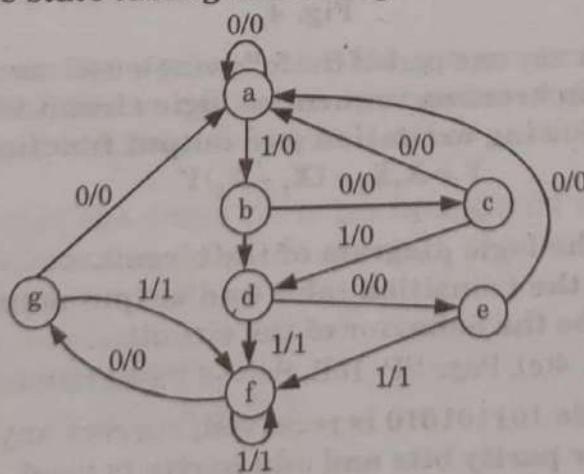
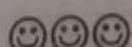


Fig. 6.

**Ans.** Refer Q. 3.9, Page 3-11B, Unit-3.





# QUANTUM Series

## Related titles in Quantum Series

### For Semester - 3 (Electronics & Allied Branches)

- Engineering Science Course / Mathematics - IV
- Technical Communication / Universal Human Values
- Electronic Devices
- Digital System Design
- Network Analysis and Synthesis

- Topic-wise coverage in Question-Answer form.
- Clears course fundamentals.
- Includes solved University Questions.

A comprehensive book to get the big picture without spending hours over lengthy text books.

**Quantum Series** is the complete one-stop solution for engineering student looking for a simple yet effective guidance system for core engineering subject. Based on the needs of students and catering to the requirements of the syllabi, this series uniquely addresses the way in which concepts are tested through university examinations. The easy to comprehend question-answer form adhered to by the books in this series is suitable and recommended for student. The students are able to effectively grasp the concepts and ideas discussed in their course books with the help of this series. The solved question papers of previous years act as a additional advantage for students to comprehend the paper pattern, and thus anticipate and prepare for examinations accordingly.

The coherent manner in which the books in this series present new ideas and concepts to students makes this series play an essential role in the preparation for university examinations. The detailed and comprehensive discussions, easy to understand examples, objective questions and ample exercises, all aid the students to understand everything in an all-inclusive manner.

- The perfect assistance for scoring good marks.
- Good for brush up before exams.
- Ideal for self-study.



## Quantum Publications®

(A Unit of Quantum Page Pvt. Ltd.)

Plot No. 59/2/7, Site-4, Industrial Area, Sahibabad,  
Ghaziabad, 201010, (U.P.) Phone: 0120-4160479  
E-mail: pagequantum@gmail.com Web: www.quantumpage.co.in



Find us on: [facebook.com/quantumseriesofficial](https://facebook.com/quantumseriesofficial)