

**PART-1**

*Number System and its Arithmetic, Signed Binary Numbers.*

**Que 1.1.** Define number system. Give classification of number system. Also define signed and unsigned binary number.

**Answer**

- A. **Number system :** It is a language of digital systems consisting of a set of symbols called digits with rules defined for their addition, multiplication and other mathematical operations.
- B. **Classification of number system :** The classification of number system is as follows :
  - 1. **Decimal number system :** It has 10 symbols, so the base or radix of this number system is 10. The 10 symbols are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
  - 2. **Binary number system :** It is a base 2 number system. The two binary digits are 1 and 0.
  - 3. **Octal number system :** It has a base of 8, it has eight possible digits 0, 1, 2, 3, 4, 5, 6 and 7.
  - 4. **Hexadecimal number system :** It is a base 16 number system. It has digits from 0 to 9, A, B, C, D, E and F.
- C. **Signed binary number :** Binary number that carry identification as to their polarity is called signed binary number. Plus (+) and minus (-) signs for positive and negative numbers can be represented in a digital format. The three major signed binary notations are: sign magnitude notation, 1's complement notation and 2's complement notation.
- D. **Unsigned binary number :** In these type of numbers we do not consider the (+) or (-) sign and concentrate only on the magnitude (absolute value) of numbers.

**Que 1.2.** Convert the following,

1.  $(5162)_{10} = ()_2$
2.  $(11011001)_2 = ()_{10}$
3.  $(6273)_{10} = ()_8$
4.  $(7860)_{10} = ()_{16}$
5.  $(A23B8)_{16} = ()_{10}$

**AKTU 2022-23 (Sem-3), Marks 10**

**Answer**

1.  $(5162)_{10} = ()_2$

- |   |        |
|---|--------|
| 2 | 5162   |
| 2 | 2581 0 |
| 2 | 1290 1 |
| 2 | 645 0  |
| 2 | 322 1  |
| 2 | 161 0  |
| 2 | 80 1   |
| 2 | 40 0   |
| 2 | 20 0   |
| 2 | 10 0   |
| 2 | 5 0    |
| 2 | 2 1    |
| 2 | 1 0    |
- $(5162)_{10} = (1010000101010)_2$
- $(11011001)_2 = ()_{10}$
- |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
|---|---|---|---|---|---|---|---|
- $1 \times 2^0 = 1$   
 $0 \times 2^1 = 0$   
 $0 \times 2^2 = 0$   
 $1 \times 2^3 = 8$   
 $1 \times 2^4 = 16$   
 $0 \times 2^5 = 0$   
 $1 \times 2^6 = 64$   
 $1 \times 2^7 = 128$
- $\underline{217}$
- $(11011001)_2 = (217)_{10}$
- $(6273)_{10} = ()_8$
- |   |       |
|---|-------|
| 8 | 6273  |
| 8 | 784 1 |
| 8 | 98 0  |
| 8 | 12 2  |
| 8 | 1 4   |
|   | 0 1   |
- $(6273)_{10} = (14201)_8$
- $(7860)_{10} = ()_{16}$
- |    |             |
|----|-------------|
| 16 | 7860        |
| 16 | 491 4       |
| 16 | 30 11 = (B) |
| 16 | 1 14 = (E)  |
|    | 0 1         |
- $(7860)_{10} = (1EB4)_{16}$
- $(A23B8)_{16} = ()_{10}$

$$\begin{array}{ccccccc}
 & A & 2 & 3 & B & 8 \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
 & 8 \times 16^0 = 8 & & & & & \\
 & & B \times 16^1 = 176 & & & & \\
 & & 3 \times 16^2 = 768 & & & & \\
 & & 2 \times 16^3 = 8192 & & & & \\
 & & A \times 16^4 = 655360 & & & & \\
 & & & & & & \\
 & & & & & & \underline{\underline{664504}} \\
 & & & & & & \\
 & (A23B8)_{16} = (664504)_{10} & & & & & 
 \end{array}$$

**Que 1.3.** Convert the following :

- i.  $(62.7)_8 = (\ )_{16} = (\ )_2$   
ii.  $(BC\ 6)_{16} = (\ )_{10} = (\ )_2$

**Answer**

i.  $(62.7)_8 = (\ )_{16} = (\ )_2$   
 $\begin{array}{r} 110 \\ 6 \end{array} \quad \begin{array}{r} 010 \\ 2 \end{array} \quad \begin{array}{r} .111 \\ 7 \end{array} \rightarrow \begin{array}{r} 0011 \\ 110 \\ .0010 \\ .1110 \end{array} \Rightarrow (32.E)_{16}$

ii.  $(BC6)_{16} = (\ )_{10} = (\ )_2$   
 $(BC6)_{16} = 11 \times 16^2 + 12 \times 16^1 + 6 \times 16^0 = (3014)_{10}$   
 $(3014)_{10} = (10111000110)_2$

**Que 1.4.** Add the following numbers :

- i.  $(ABC)_{16} + (CDE)_{16}$   
ii.  $(77)_8 + (107)_8$

**Answer**

i.  $(ABC)_{16} = 1010 \ 1011 \ 1100$  (in binary)  
 $(CDE)_{16} = 1100 \ 1101 \ 1110$   
 $(ABC)_{16} + (CDE)_{16} = \begin{array}{r} 10111 \\ 1001 \\ 1010 \end{array} = (179A)_{16}$

ii.  $(77)_{16} = 000 \ 111 \ 111$  (in binary)  
 $(107)_{16} = 001 \ 000 \ 111$   
 $(77)_{16} + (107)_{16} = \begin{array}{r} 010 \\ 000 \\ 111 \end{array} = (206)_8$

**Que 1.5.** Perform the following subtraction using 2's complement

method :

- i.  $01000 - 01001$   
ii.  $0011.1001 - 0001.1110$

**Answer**

i. Assuming,  $X = 01000, Y = 01001$   
1's complement :  $Y = 10110$

2's complement :  $\begin{array}{r} +1 \\ Y = 10111 \\ X = 01000 \\ - Y = 10111 \\ \hline \text{Difference} = 11111 \end{array}$

There is no end carry.

$\therefore X - Y = -(2\text{'s complement of } 11111)$   
 $= -(00001)$

ii. Assuming,  $X = 0011.1001, Y = 0001.1110$   
1's complement :  $Y = 1110.0001$

2's complement :  $\begin{array}{r} +1 \\ Y = 1110.0010 \\ X = 0011.1001 \\ - Y = 1110.0010 \\ \hline \text{Difference} = \boxed{1} \ 0001.1011 \\ \rightarrow \text{discard carry} \end{array}$

Since, carry is generated, answer will remain same, i.e., 0001.1011

## PART-2

Logic Simplification and Combinational Logic Design :  
Binary Codes.

**Que 1.6.** Describe the binary codes. Show the classification of binary codes in tabular format.

**Answer**

- 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.
- 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 :

- Weighted codes
- Non-weighted codes
- Reflective codes
- 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:

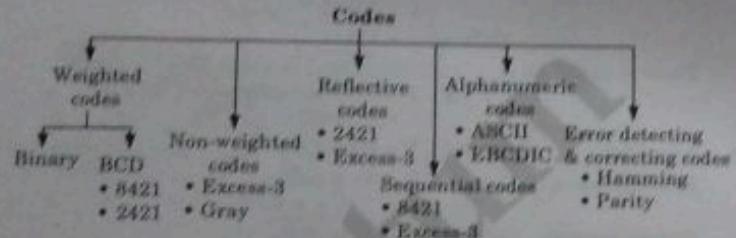


Fig. 1.6.1.

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's 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 1.7.** Represent the decimal number 6 in (i) excess-3 code, (ii)

BCD code, (iii) Gray code, (iv) 8421 code and (v) 2421 codes.

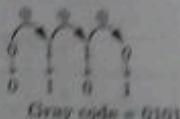
#### Answer

##### I. Excess-3 code :

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

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

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



Gray code = 0101

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

$$(6)_{8421} = 1010$$

v. 2421 code :

$$(6)_{2421} = 1100$$

#### PART-3

##### Code Conversion.

**Que 1.8.** Design Binary code to Gray code converter.

AKTU 2018-19 (Sem-3), Marks 07

OR

Design a binary code to gray code converter. Also show its truth table, boolean expression and logic diagram.

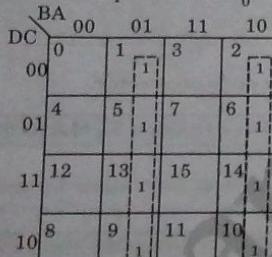
AKTU 2021-22 (Sem-4), Marks 10

#### Answer

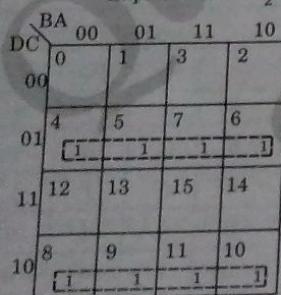
##### Truth table :

D	Binary code				Gray code			
	C	B	A	G <sub>1</sub>	G <sub>2</sub>	G <sub>3</sub>	G <sub>4</sub>	
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	1
0	0	1	1	0	0	0	1	0
0	1	0	0	0	0	1	1	0
0	1	0	1	0	0	1	1	1
0	1	1	0	0	0	1	0	1
0	1	1	1	0	0	1	0	0
1	0	0	0	1	1	1	0	0
1	0	0	1	1	1	0	1	1

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

**K-map simplification :**Expression for  $G_0$ 

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

Expression for  $G_2$ 

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

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

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

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

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

Logic diagram :

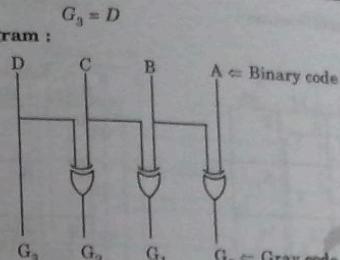


Fig. 1.8.1.

Que 1.9. 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	0	1	0
0	1	0	1	0	0	0	1	1
0	1	1	0	0	0	1	0	0
0	1	1	1	0	0	1	1	1
1	0	0	0	0	1	0	0	0
1	0	0	1	0	0	1	0	1
1	0	1	0	0	1	0	0	0
1	0	1	1	1	0	0	0	1
1	1	0	0	1	0	0	0	0
1	1	0	1	1	0	0	0	1
1	1	1	0	0	1	0	0	0
1	1	1	1	1	0	0	1	1

K-map simplification :

Expression for $B_0$				
BA	00	01	11	
DC	0	1	3	2
00	0	1	1	1
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

$B_0 = A$

Expression for $B_1$				
BA	00	01	11	
DC	0	1	3	2
00	0	1	1	1
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

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

Expression for $B_2$				
BA	00	01	11	
DC	0	1	3	2
00	0	1	1	1
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

$B_2 = \bar{D}C + CB$

Expression for $B_3$				
BA	00	01	11	
DC	0	1	3	2
00	0	1	1	1
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

$B_3 = DC\bar{B}$

Expression for $B_4$				
BA	00	01	11	
DC	0	1	3	2
00	0	1	1	1
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

$B_4 = DC + DB$

Logic diagram :

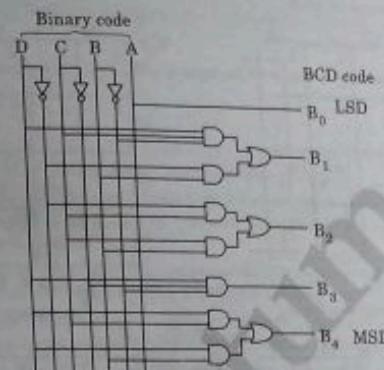


Fig. 1.9.1. Logic circuit for binary to BCD converter.

Que 1.10. Design an excess-3 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	0
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 × (Don't care condition) for the corresponding codes.

K-map simplification :

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

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

Expression for  $B_1$ 

		Expression for $B_1$				
		00	01	11	10	
		00	x	1	3	2
E <sub>3</sub>	E <sub>2</sub>	00	1	5	7	6
0	1	4	1			1
1	12	11	13	15	14	x
10	8	9	11	10	1	1

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

Logic diagram :

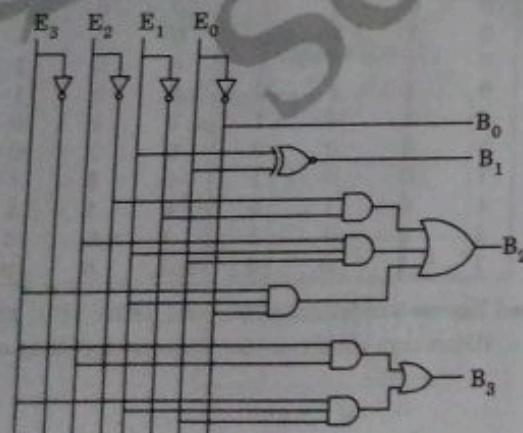


Fig. 1.10.1.

## PART-4

Review of Boolean Algebra and DeMorgan's Theorem.

Que 1.11. Discuss about the concept of Boolean algebra.

## Answer

Boolean algebra is a fundamental concept that deals with the manipulation of binary values, represented as 0 and 1.

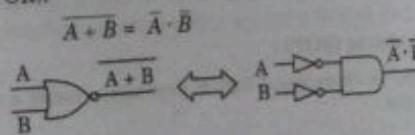
- Basic concepts :** Boolean algebra operates on variables that can only take on two values: 0 (false) or 1 (true). These variables are combined using three basic operations :
  - AND (A) :** The AND operation represents the logical conjunction. The result of  $A \wedge B$  is 1 only if both A and B are 1. Otherwise, the result is 0.
  - OR (V) :** The OR operation represents the logical disjunction. The result of  $A \vee B$  is 1 if either A or B is 1. Otherwise, the result is 0.
  - NOT ( $\neg$ ) :** The NOT operation represents the logical negation. The result of  $\neg A$  is the opposite of A. If A is 1, then  $\neg A$  is 0. Conversely, if A is 0, then  $\neg A$  is 1.
- Truth tables :** Truth tables are a convenient way to represent the results of Boolean operations for all possible combinations of input values. Each row in a truth table represents a different combination of input values, and each column represents the output value for a specific operation.
- Boolean expressions :** Boolean expressions are combinations of variables and Boolean operators. They can be written in various forms, such as conjunctive normal form or disjunctive normal form. Simplifying Boolean expressions can reduce the complexity of digital circuits and improve their performance.

Que 1.12. Define the De Morgan's theorem of logic simplification for SOP & POS forms.

AKTU 2022-23 (Sem-3), Marks 10

## Answer

**First De Morgan's 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).



Inputs		Intermediate outputs			Outputs	
A	B	$A + B$	$\bar{A}$	$\bar{B}$	$A + \bar{B}$	$\bar{A} \bar{B}$
0	0	0	1	1	1	1
0	1	1	1	0	0	0
1	0	1	0	1	0	0
1	1	1	0	0	0	0

**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).

$$\overline{A \cdot B} = \bar{A} + \bar{B}$$



Inputs		Intermediate outputs			Outputs	
A	B	$AB$	$\bar{A}$	$\bar{B}$	$\overline{AB}$	$\bar{A} + \bar{B}$
0	0	0	1	1	1	1
0	1	0	1	0	1	1
1	0	0	0	1	1	1
1	1	1	0	0	0	0

$$\begin{aligned} \text{Example : } (AB)(CD)(EFG) &= (\overline{AB}) + (\overline{CD}) + (\overline{EFG}) \\ &= (\bar{A} + \bar{B}) + (\bar{C} + \bar{D}) + (\bar{E} + \bar{F} + \bar{G}) \end{aligned}$$

### PART-5

#### SOP and POS Forms, Canonical Forms.

**Que 1.13.** Explain sum of products (SOP) and product of sums (POS) with example.

#### Answer

##### Sum of products (SOP) :

1. SOP is a way to represent a logical expression using a sum of several product terms.

2. In SOP, each product term is a combination of input variables (either complemented or uncomplemented) connected by the logical AND operation. These product terms are then connected by the logical OR operation.
3. SOP is also known as the "Minterm" representation because each product term represents a unique combination of input variables for which the expression evaluates to 1.
4. SOP expressions are used to implement digital functions using AND and OR gates. It is often used for simplifying and analyzing logic circuits.

**Example SOP expression :**  $F(A, B, C) = ABC + \bar{A}BC + ABC + \bar{A}\bar{B}C$

##### Product of sums (POS) :

1. POS is another way to represent a logical expression, but it uses a product of several sum terms.
2. In POS, each sum term is a combination of input variables connected by the logical OR operation, and these sum terms are then connected by the logical AND operation.
3. POS is also known as the "Maxterm" representation because each sum term represents a unique combination of input variables for which the expression evaluates to 0.
4. POS expressions are used to implement digital functions using OR and AND gates. It is an alternative representation to SOP and can be useful in certain situations, especially when dealing with simplification or implementing certain types of circuits.

**Example POS expression :**

$$F(A, B, C) = (A + B + C)(A + B + C')(A + B' + C)(A' + B + C)$$

**Que 1.14.** Discuss about canonical forms in digital circuits.

#### Answer

##### Canonical form :

1. The canonical form refers to the representation of a logical expression in its most basic and unaltered form. It can be either a canonical SOP or a canonical POS form.
2. Canonical SOP is when an expression is represented as the sum of all possible minterms for a given set of input variables, and canonical POS is when it is represented as the product of all possible maxterms.
3. Canonical forms are used for clarity, and they provide a unique representation of a function.
4. They are often used in theory and analysis rather than practical circuit implementation because they are not efficient for larger circuits.

**PART-6**

*Karnaugh Maps Method up to Five Variables, Don't Care Conditions, POS Simplification.*

**Que 1.15.** Write a short note on Karnaugh map. Also show the reduction of boolean expression and how to mark pairs.

**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 :

A \ B	$\bar{B}$	B
$\bar{A}$	$\bar{A}\bar{B}$	$\bar{A}B$
A	$A\bar{B}$	AB

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

Fig. 1.15.1.

**3. For three variables :**

- i. There are eight minterms for three binary variables. Hence the K-map consists of eight squares.

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	$m_0$	$m_1$	$m_3$	$m_2$
A	$m_4$	$m_5$	$m_7$	$m_6$

Fig. 1.15.2.

- ii. 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.
- iii. 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.

**4. For four variables :**

- i. The map for boolean function of four binary variables require sixteen minterms, hence the map consists of sixteen squares.

**Digital Electronics****1-17 G (OEC-Sem-3 & 4)**

- i. The listed terms are from 0 to 15, i.e., 16 minterms. The map shows the relationship with the four variables.
- ii. 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	$C\bar{D}$
$\bar{A}\bar{B}$	$m_0$	$m_1$	$m_3$	$m_2$
$\bar{A}B$	$m_4$	$m_5$	$m_7$	$m_6$
AB	$m_{12}$	$m_{13}$	$m_{15}$	$m_{14}$
$A\bar{B}$	$m_8$	$m_9$	$m_{11}$	$m_{10}$

CD	00	01	11	10
AB	0000 0	0001 1	0011 3	0010 2
OR	0100 4	0101 5	0111 7	0110 6
	1100 12	1101 13	1111 15	1110 14
	1000 8	1001 9	1011 11	1010 10

Fig. 1.15.3.

**Que 1.16.** 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 (Sem-3), Marks 07

**Answer**

CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	1	1
$\bar{A}B$	1	1	1	1
AB	1	1	1	1
$A\bar{B}$	1	1	1	1

Fig. 1.16.1.

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

**Que 1.17.** 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 (Sem-3), Marks 07

**Answer**

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

**K-map :**

		CDE	000	001	011	010	110	111	101	100
		AB	00	01	12	8	1	1	28	1
		01	1	1	5	13	1	1	21	25
		11	1	1	3	7	15	11	19	23
		10	2	1	6	14	10	1	18	22
									30	1
										26

Fig. 1.17.1.

$$Y = B\bar{C}\bar{D} + \bar{B}CD + \bar{A}\bar{B}\bar{D}\bar{E} + A\bar{C}\bar{D}E + B\bar{C}\bar{D}\bar{E} + ABD\bar{E} + \bar{A}CDE$$

$$Y = (B + AE)\bar{C}\bar{D} + (\bar{B} + \bar{A}E)CD + (\bar{A} + C)\bar{B}\bar{D}\bar{E} + (\bar{C} + A)BDE$$

**Que 1.18.** Simplify the following Boolean function using K-map and also draw the simplified logic circuit using basic gates.

$$f(A, B, C, D) = \Sigma m(0, 1, 5, 6, 12, 13, 14) + d(2, 4)$$

AKTU 2021-22 (Sem-4), Marks 10

**Answer**

**K-map :**

		CD	00	01	11	10	
		AB	00	1	1	3	x
		01	x	1	5	6	7
		11	1	1	12	13	14
		10	8	9	11	10	

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

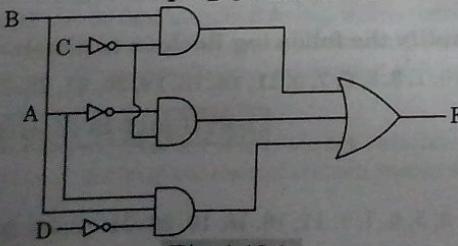


Fig. 1.18.1.

**Que 1.19.** Simplify the logic function using K-map

$$Y = \Sigma m(0, 2, 3, 4, 6, 7, 9, 11, 16, 18, 19, 20, 22, 23, 25, 27)$$

**Answer**

$$Y = \Sigma m(0, 2, 3, 4, 6, 7, 9, 11, 16, 18, 19, 20, 22, 23, 25, 27)$$

The given function can be solved by using 5-variable K-map

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

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

$$Y = \bar{B}\bar{D}\bar{E} + \bar{B}D + B\bar{C}E$$

**K-map :**

		DE	00	01	11	10	
		ABC	00	1	0	1	3
		001	1	4	5	1	7
		011	12		13	15	14
		010	8	1	9	1	11
		110	24	1	25	1	27
		111	28		29	31	30
		101	1	20	21	1	23
		100	1	16	17	1	19

Fig. 1.19.1

**Que 1.20.** Simplify  $Y = \Sigma m(3, 6, 7, 8, 10, 12, 14) + d(0, 1, 6, 15)$  using K-map method and implement the simplified circuit using logic gates.

AKTU 2022-23 (Sem-4), Marks 10

**Answer**

Given,

$$Y = \Sigma m(3, 6, 7, 8, 10, 12, 14) + d(0, 1, 6, 15)$$

K-map :

	CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	x	x	1	3	2
$\bar{A}B$	4	5	1	7	6
$A\bar{B}$	1	12	13	15	14
$AB$	8	9	11	10	1

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

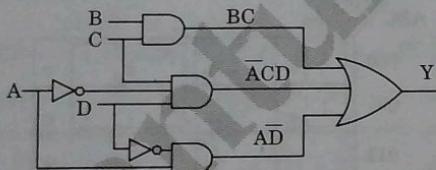


Fig. 1.20.1.

**Que 1.21.** Simplify the following expression into product of sum (POS) form

- $ABC + A\bar{B}D + BCD$
- $AC\bar{D} + \bar{C}D + A\bar{B} + ABCD$

**Answer**

- $ABC + A\bar{B}D + BCD$

- Let  $Y = ABC + A\bar{B}D + BCD$   
 $= ABC(D + \bar{D}) + A\bar{B}(C + \bar{C})D + (A + \bar{A})BCD$   
 $= AB\bar{C}D + AB\bar{C}\bar{D} + A\bar{B}CD + A\bar{B}\bar{C}D + ABCD + A\bar{B}CD$

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

- Now for POS form we will take complement function,  
 $\bar{Y} = \Pi M(0, 1, 2, 3, 4, 5, 6, 8, 10, 14)$
- Minimization through K-map is shown in Fig. 1.21.1.

	CD	$C + D$	$C + \bar{D}$	$\bar{C} + \bar{D}$	$\bar{C} + D$
$A + B$	0	0	0	0	2
$A + \bar{B}$	0	0	1	3	0
$\bar{A} + \bar{B}$	4	5	7	6	0
$\bar{A} + B$	12	13	15	14	0
$\bar{A} + B$	0	8	9	11	10

Fig. 1.21.1.

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

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

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

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

- Now for POS form, we have to take complement function,

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

- Minimization through K-map is shown in Fig. 1.21.2.

	CD	$C + D$	$C + \bar{D}$	$\bar{C} + \bar{D}$	$\bar{C} + D$
$A + B$	0	0	1	0	2
$A + \bar{B}$	0	0	0	0	0
$\bar{A} + \bar{B}$	4	5	7	6	0
$\bar{A} + B$	12	13	15	14	0
$\bar{A} + B$	0	8	9	11	10

Fig. 1.21.2.

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

**PART-7**

NAND and NOR Implementation.

**Que 1.22.** Implement the Boolean function  $F(x, y, z) = (1, 2, 3, 4, 6, 7)$  using NAND gates.

AKTU 2022-23 (Sem-4), Marks 10

**Answer**

- The K-map simplification is shown in Fig. 1.22.1.

x \ y\z	00	01	11	10
0	0	1	1	1
1	1	4	5	6

Fig. 1.22.1.

- Hence, the simplified function is

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

- Implementation using NAND gate is shown in Fig. 1.22.2

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

Using De Morgan's theorem,

$$F(x, y, z) = \overline{\overline{y} + \bar{x}z + x\bar{z}}$$

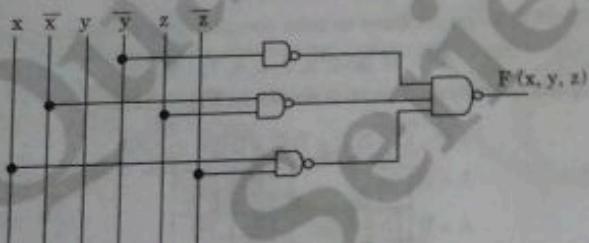


Fig. 1.22.2.

**Que 1.23.** Design an XOR gate by using NAND gate implementation.

AKTU 2022-23 (Sem-3), Marks 10

**Answer**

The NAND gate is a universal logic gate hence, using which we may implement any other logic gate.

**Logic diagram :**

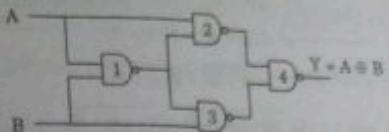


Fig. 1.23.1.

The output of 1<sup>st</sup> NAND gate is

$$Y_1 = \overline{AB}$$

Similarly outputs of second and third NAND gates are

$$Y_2 = \overline{A \cdot \overline{AB}}$$

$$Y_3 = \overline{B \cdot \overline{AB}}$$

As these two outputs  $Y_2$  and  $Y_3$  are connected to the fourth NAND gate. This NAND gate will produce an output.

$$\begin{aligned} Y &= \overline{A \cdot \overline{AB} \cdot \overline{B \cdot \overline{AB}}} \\ &= \overline{A \cdot \overline{AB}} + \overline{B \cdot \overline{AB}} \\ &= A(\overline{A} + \overline{B}) + B(\overline{A} + \overline{B}) \\ &= AA + A\overline{B} + B\overline{A} + BB \\ &= A\overline{B} + B\overline{A} \\ &= A \oplus B \end{aligned}$$

**Que 1.24.** Express the design of EX-OR gate with the help of

- NAND gates only and
- NOR gate only.

AKTU 2021-22 (Sem-4), Marks 10

**Answer**

- Design of EX-OR (XOR) gate using NAND gates : Refer Q. 1.23, Page 1-22G, Unit-1.
- Design of EX-OR gate using NOR gates :

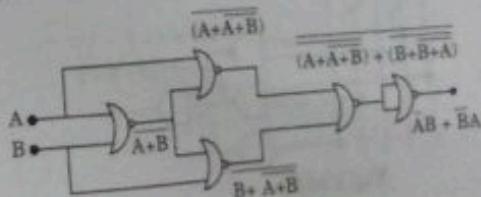


Fig. 1.24.1.

$$\begin{aligned}
 f &= \bar{A}B + \bar{B}A = \bar{A}\bar{B} + \bar{B}A + A\bar{A} + B\bar{B} \\
 &= \bar{A}(B + A) + \bar{B}(A + B) \\
 &= \bar{A}(\overline{A + B}) + \bar{B}(\overline{A + B}) \\
 &= \overline{A + (A + B)} + \overline{B + (A + B)} \\
 &= \overline{(A + (A + B))} + \overline{(B + (A + B))}
 \end{aligned}$$

**Que 1.25.** Implement the following boolean function with NAND gates.

$$F(x, y, z) = \sum m(1, 2, 3, 4, 5, 7)$$

**Answer**

1. The K-map simplification is shown in Fig. 1.25.1.

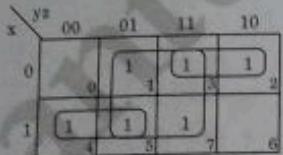


Fig. 1.25.1.

2. Hence, the simplified function is

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

3. Implementation using NAND gates is shown in Fig. 1.25.2.

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

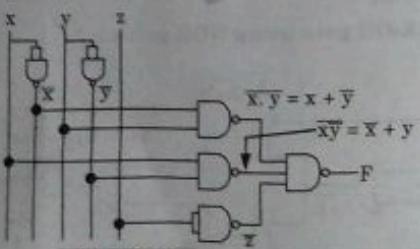


Fig. 1.25.2.

**PART-B**

Quine McClusky Method (Tabular Method).

**Que 1.26.** Minimize the following using Quine-McCluskey method :

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

AKTU 2018-19 (Sem-3), Marks 07

**Answer**

1. Arrange minterms according to categories of 1's as shown in table 1.26.1.

Table 1.26.1.

No. of 1's	Min-terms	Binary					Minterms (2 cell)					Binary					Minterms (4 cell)					Binary				
		A	B	C	D	E	A	B	C	D	E	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	-	0	-	-	
1	$m_1$	0	0	0	0	1	0	8*	*	0	-	0	0	0	-	-	-	-	-	-	-	-	-	-	-	
	$dm_8$	0	1	0	0	0	1	9	*	0	-	0	0	1	-	-	-	-	-	-	-	-	-	-	-	
2	$m_9$	0	1	0	0	1	8*	9	*	0	1	0	0	-	-	-	-	-	-	-	-	-	-	-	-	
	$m_{24}$	1	1	0	0	0	0	8*	24	-	1	0	0	0	-	-	-	-	-	-	-	-	-	-	-	
3	$dm_{11}$	0	1	0	1	1	9, 11*	11	0	1	0	-	1	-	-	-	-	-	-	-	-	-	-	-	-	
	$m_{35}$	0	1	1	1	1	11*	15	0	1	-	1	1	1	-	-	-	-	-	-	-	-	-	-	-	
4	$m_{29}$	1	1	1	0	1	15, 31*	-	1	1	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	
	$m_{30}$	1	1	1	1	0	29, 31*	1	1	1	1	-	1	-	-	-	-	-	-	-	-	-	-	-	-	
5	$dm_{31}$	1	1	1	1	1	30, 31*	1	1	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	

2. List of prime implicants:

Table 1.26.2.

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 1.26.3.

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

$$Y = BCDE + ABCE + ABCD + ACD$$

Que 1.27. Minimize the following Boolean function using tabulation method :

$$F(a, b, c, d, e) = \Sigma m(0, 4, 12, 16, 19, 24, 27, 28, 29, 31)$$

AKTU 2022-23 (Sem-4), Marks 10

### Answer

$$\text{Given } f(a, b, c, d, e) = \Sigma m(0, 4, 12, 16, 19, 24, 27, 28, 29, 31)$$

Min-term	a b c d e	Gro-up	Min-term	Step 1		Step 2		Step 3	
				a b c d e	Gro-up	Min-term	a b c d e	Gro-up	Min-term
$m_0$	0 0 0 0 0	0	$m_0$	0 0 0 0 0	0	0, 4	0 0 _ 0 0		
$m_4$	0 0 1 0 0	1	$m_4$	0 0 1 0 0	0, 16	0 0 0 0 0			
$m_{12}$	0 1 1 0 0		$m_{16}$	1 0 0 0 0	1	4, 12	0 _ 1 0 0		
$m_{16}$	1 0 0 0 0	2	$m_{12}$	0 1 1 0 0	16, 24	1 _ 0 0 0			
$m_{19}$	1 0 0 1 1		$m_{24}$	1 1 0 0 0	2	12, 28	_ 1 1 0 0		
$m_{24}$	1 1 0 0 0	3	$m_{19}$	1 0 0 1 1	24, 28	1 1 _ 0 0			
$m_{27}$	1 1 0 1 1		$m_{28}$	1 1 1 0 0	3	19, 27	1 _ 0 1 1		
$m_{28}$	1 1 1 0 0	4	$m_{27}$	1 1 0 1 1	28, 29	1 1 1 0 _			
$m_{29}$	1 1 1 0 1		$m_{29}$	1 1 1 0 1	4	27, 31	1 1 _ 1 1		
$m_{31}$	1 1 1 1 1	5	$m_{31}$	1 1 1 1 1	29, 31	1 1 1 1 _			

No further reduction therefore

Thus select minimum number of prime implicant

	$m_0$	$m_4$	$m_{12}$	$m_{16}$	$m_{19}$	$m_{24}$	$m_{27}$	$m_{28}$	$m_{29}$	$m_{31}$
0, 4	$\bar{a} \bar{b} \bar{d} \bar{e}$	x	x							
0, 16	$\bar{b} \bar{c} \bar{d} \bar{e}$	x			x					
4, 12	$\bar{a} \bar{c} \bar{d} \bar{e}$		x	x						
16, 24	$a \bar{c} \bar{d} \bar{e}$				x	x				
12, 28	$b \bar{c} \bar{d} \bar{e}$		x					x		
24, 28	$a \bar{b} \bar{d} \bar{e}$					x	x	x		
19, 27	$a \bar{c} de$					x	x			
28, 29	$abc \bar{d}$						x	x	x	
27, 31	$\bar{a} bde$						x		x	x
29, 31	$abce$							x	x	x

$$Y = a \bar{c} de + \bar{a} \bar{b} \bar{d} \bar{e} + \bar{a} c \bar{d} \bar{e} + bc \bar{d} \bar{e} \\ + a \bar{c} \bar{d} \bar{e} + ab \bar{d} \bar{e} + abde + abc \bar{d} + abc e$$

Que 1.28. Use Quine-Mc Cluskey (QM) method to solve the following function :

Table 1.26.2.

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 1.26.3.

Prime implicants	Minterm							
	✓	✓	0	1	9	15	✓	✓
						24	29	30
8*, 24					○			
9, 11*			○					
11*, 15				○				
15, 31*				○				
29, 31*	✓					○		
30, 31*	✓						○	
0, 1, 8*, 9	○	○	○					

$$Y = BCDE + ABCE + ABCD + ACD$$

**Que 1.27.** Minimize the following Boolean function using tabulation method :

$$F(a, b, c, d, e) = \Sigma m(0, 4, 12, 16, 19, 24, 27, 28, 29, 31)$$

AKTU 2022-23 (Sem-4), Marks 10

### Answer

$$\text{Given } f(a, b, c, d, e) = \Sigma m(0, 4, 12, 16, 19, 24, 27, 28, 29, 31)$$

Min-term	a	b	c	d	e	Gro-up	Min-term	a	b	c	d	e	Gro-up	Min-term	a	b	c	d	e	
$m_0$	0	0	0	0	0	0	$m_0$	0	0	0	0	0	0	0.4	0	0	0	0	0	
$m_4$	0	0	1	0	0	1	$m_4$	0	0	1	0	0	0	0.16	0	0	0	0	0	
$m_{12}$	0	1	1	0	0		$m_{16}$	1	0	0	0	0		1	4.12	0	1	0	0	
$m_{16}$	1	0	0	0	0	2	$m_{12}$	0	1	1	0	0			16, 24	1	0	0	0	
$m_{19}$	1	0	0	1	1		$m_{24}$	1	1	0	0	0		2	12, 28	-	1	1	0	
$m_{24}$	1	1	0	0	0	3	$m_{19}$	1	0	0	1	1		3	19, 27	1	0	1	1	
$m_{27}$	1	1	0	1	1		$m_{28}$	1	1	1	0	0			28, 29	1	1	1	0	
$m_{28}$	1	1	1	0	0	4	$m_{27}$	1	1	0	1	1		4	27, 31	1	1	-	1	
$m_{29}$	1	1	1	0	1		$m_{29}$	1	1	1	0	1			29, 31	1	1	1	-	1
$m_{31}$	1	1	1	1	1	5	$m_{31}$	1	1	1	1	1								

No further reduction therefore

Thus select minimum number of prime implicant

		$m_0$	$m_4$	$m_{12}$	$m_{16}$	$m_{19}$	$m_{24}$	$m_{27}$	$m_{28}$	$m_{29}$	$m_{31}$
0, 4	$\bar{a} \bar{b} d \bar{e}$	×	×								
0, 16	$\bar{b} \bar{c} \bar{d} \bar{e}$		×								
4, 12	$a \bar{c} \bar{d} \bar{e}$			×	×						
16, 24	$a \bar{c} \bar{d} \bar{e}$					×		×			
12, 28	$b \bar{c} \bar{d} \bar{e}$					×					×
24, 28	$ab \bar{d} \bar{e}$								×	×	
19, 27	$a \bar{c} de$								×	×	
28, 29	$abc \bar{d}$									×	×
27, 31	$\bar{a} bde$									×	×
29, 31	$abce$										×

$$Y = a \bar{c} de + \bar{a} \bar{b} \bar{d} \bar{e} + \bar{a} c \bar{d} \bar{e} + bc \bar{d} \bar{e} \\ + a \bar{c} \bar{d} \bar{e} + ab \bar{d} \bar{e} + abc \bar{d} + abce$$

**Que 1.28.** Use Quine-Mc Cluskey (QM) method to solve the following function :

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

**Answer**

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

1.

No. of 1's	Minterms	Binary	Minterms (2 cell)	Binary	Minterm (4 cell)	Binary
1.	$m_5$	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	✓	✓	✓	✓	✓	✓	✓
$\bar{A}BD$ ✓	5, 7	○	○					
$BCD$ ✓	7, 15		○					○
$A\bar{B}$ ✓	8, 9, 10, 11			○	○	○	○	
$AC$ ✓	10, 11, 14, 15				○	○	○	○

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

3. Logic diagram :

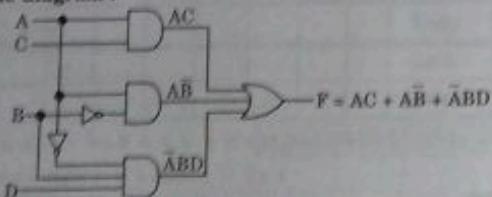


Fig. 1.28.1.

**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 binary codes. Show the classification of binary codes in tabular format.

ANS: Refer Q. 1.6, Unit-1.

- Q. 2. Design a binary code to gray code converter. Also show its truth table, boolean expression and logic diagram.

ANS: Refer Q. 1.8, Unit-1.

- Q. 3. Define the De Morgan's theorem of logic simplification for SOP & POS forms.

ANS: Refer Q. 1.12, Unit-1.

- Q. 4. 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. 1.16, 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. 1.17, Unit-1.

- Q. 6. Simplify
- $Y = \Sigma m(3, 6, 7, 8, 10, 12, 14) + d(0, 1, 8, 15)$
- using K-map method and implement the simplified circuit using logic gates.

ANS: Refer Q. 1.20, Unit-1.

- Q. 7. Implement the Boolean function
- $F(x, y, z) = (1, 2, 3, 4, 6, 7)$
- using NAND gates.

ANS: Refer Q. 1.22, Unit-1.

- Q. 8. Express the design of EX-OR gate with the help of
- 
- i. NAND gates only and
- 
- ii. NOR gate only.

ANS: Refer Q. 1.24, Unit-1.

- Q. 9. 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. 1.26, Unit-1.



# 2

UNIT

## Combinational Logic

### CONTENTS

Part-1 : MSI Devices Like Magnitude Comparator	2-2G to 2-8G
Part-2 : Multiplexers	2-9G to 2-15G
Part-3 : Demultiplexers	2-15G to 2-17G
Part-4 : Decoders	2-18G to 2-20G
Part-5 : Encoders, Multiplexed Display	2-20G to 2-22G
Part-6 : Half and Full Adders	2-22G to 2-25G
Part-7 : Subtractors	2-25G to 2-28G
Part-8 : Serial and Parallel Adders	2-28G to 2-29G
Part-9 : BCD Adder	2-30G to 2-32G

2-2 G (OEC-Sem-3 & 4)

Combinational Logic

#### PART-1

*MSI Devices Like Magnitude Comparator.*

**Que 2.1.** Describe combinational logic circuit with its block diagram.

#### Answer

1. Combinational logic circuits consist of an interconnection of logic gates in which the output at any time depends upon the combination of input signals present at that instant only, and does not depend on any past conditions.
2. In combinational circuit, the output does not depend on the past value of input or output. Hence combinational circuits do not require any memory.

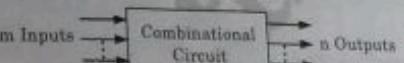


Fig. 2.1.1.

3. In a combinational circuit, for a change in the input the output appears immediately, except for the propagation delay through circuit gates.
4. For  $m$  input variables, there are  $2^m$  possible combinations of binary input values.

**Que 2.2.** 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.

AKTU 2017-18 (Sem-3), Marks 07

#### Answer

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).

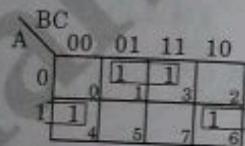
2-1 G (OEC-Sem-3 & 4)

**Circuit design :**

Let the three input be  $A$ ,  $B$  and  $C$ , so according to the question 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



$$Y = A\bar{C} + \bar{A}C = A \oplus C$$



**Que 2.3.** What is magnitude comparator? Design a single 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. 2.3.1.

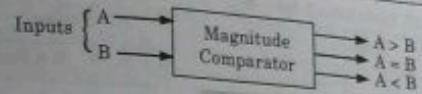


Fig. 2.3.1.

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. 2.3.2.
5. The circuit diagram and truth table of a single bit magnitude comparator is shown in Fig. 2.3.2.

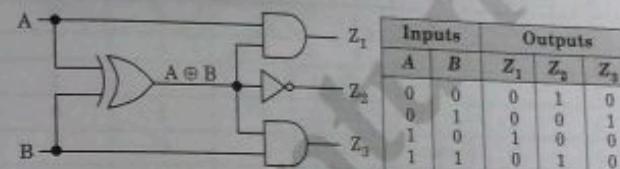


Fig. 2.3.2.

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

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

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

**Que 2.4.** Draw and explain 2-bit magnitude comparator. Also represent output with the help of logic diagram.

AKTU 2022-23 (Sem-4), Marks 10

**Answer**

1. A comparator used to compare two binary numbers each of two bits is called a 2-bit Magnitude comparator. It consists of four inputs and three outputs to generate less than, equal to and greater than between two binary numbers.

2. The truth table for a 2-bit comparator is given below:

Input				Output		
$A_1$	$A_0$	$B_1$	$B_0$	$A < B$	$A = B$	$A > B$
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

3. From the above truth table K-map for each output can be drawn as follows:

$B_1B_0$	A > B			
$A_1A_0$	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

$$F(A, B, C) = A_1B_1' + A_0B_1'B_0 + A_1A_0B_0'$$

$B_1B_0$	A = B			
$A_1A_0$	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1

$$\begin{aligned} F(A, B, C) &= A_1'A_0'B_1'B_0' + A_1'A_0B_1B_0 + A_1A_0B_1B_0 + A_1A_0B_1B_0' \\ &= A_1'B_1'(A_0B_0' + A_0B_0) + A_1B_1(A_0B_0 + A_0B_0) \\ &= (A_0B_0 + A_0B_0')(A_1B_1 + A_1'B_1) \\ &= (A_0 \oplus B_0)(A_1 \oplus B_1) \end{aligned}$$

$B_1B_0$	A < B			
$A_1A_0$	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	1	0

$$F(A, B, C) = A_1B_1 + A_0B_1B_0 + A_1A_0B_0$$

4. By using these Boolean expressions, we can implement a logic circuit for this comparator as given below:

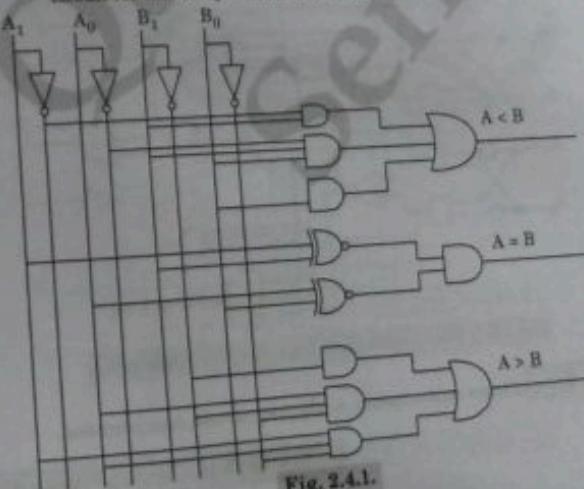


Fig. 2.4.1.

**Que 2.5.** Design a three bit comparator circuit using logic gates.

**Answer**

3-bit magnitude comparator :

$$A = A_2 A_1 A_0$$

$$B = B_2 B_1 B_0$$

Two numbers  $A$  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.  $(A < B) = \bar{A}_2 B_2 + x_2 \bar{A}_1 B_1 + x_2 x_1 \bar{A}_0 B_0$

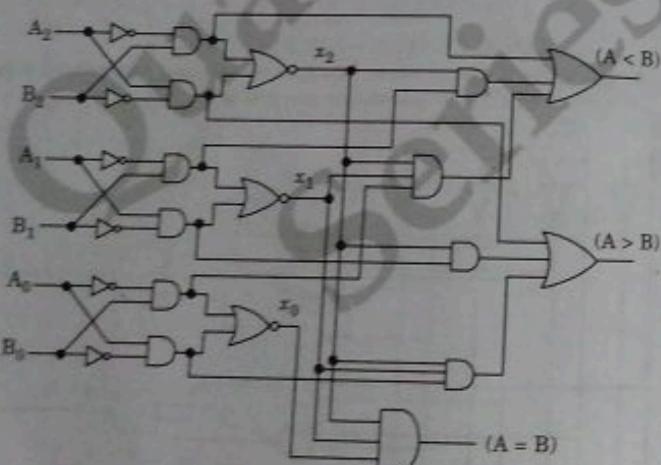


Fig. 2.5.1. 3-bit magnitude comparator using logic gates.

**Que 2.6.** Draw and explain 4-bit magnitude comparator.

AKTU 2017-18 (Sem-3), Marks 07

**2-8 G (OEC-Sem-3 & 4)**

**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.

$$x_i = A_i B_i + A'_i B'_i ; i = 0, 1, 2, 3$$

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  $A < B$ ,

$$(A > B) = A_3 B'_3 + x_3 A_2 B'_2 + x_3 x_2 A_1 B'_1 + x_3 x_2 x_1 A_0 B'_0$$

$$(A < B) = A'_3 B_3 + x_3 A'_2 B_2 + x_3 x_2 A'_1 B_1 + x_3 x_2 x_1 A'_0 B_0$$

4. The logical implementation is shown in Fig. 2.6.1.

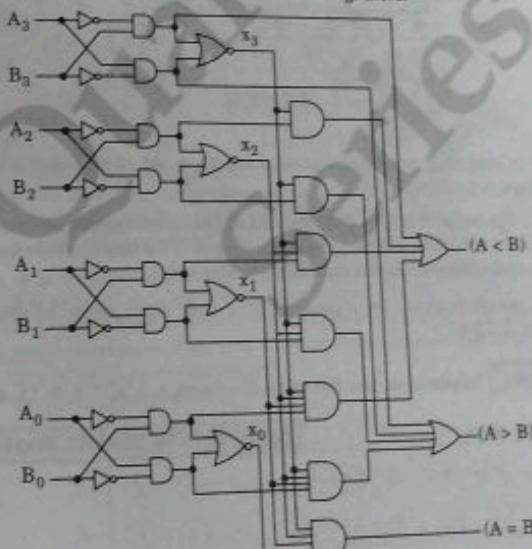


Fig. 2.6.1. 4-bit magnitude comparator using logic gates.

## PART-2

## Multiplexers.

**Que 2.7.** What is multiplexer in the digital electronics?

**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. 2.7.1(a).

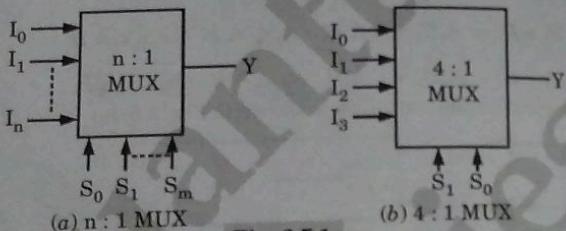


Fig. 2.7.1.

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. 2.7.1(b).

**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 (Sem-3), Marks 07

**Answer**

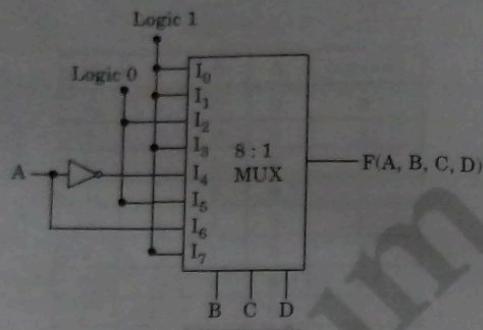
Decimal	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

1. 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.
2. A is assumed to be MUX input and B, C, D are used as selection lines.
  - i.  $\bar{A}$  is complement variable of A for the minterm 0 to 7.
  - ii. 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 :



Que 2.9. Implement the following Boolean function.

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

- i. 4 : 1 MUX
- ii. 2 : 1 MUX

AKTU 2018-19 (Sem-3), Marks 07

## Answer

- i. Implementation using 4 : 1 MUX : We have to use four variables as input of MUX and one variable as select line.

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

$$\begin{aligned} \text{First column } (I_0) \quad &= \bar{C}\bar{D} + \bar{C}D + CD \\ &= \bar{C}(\bar{D} + D) + CD = \bar{C} + CD \\ &= \bar{C} + D \end{aligned}$$

Second column ( $I_1$ )

$$\begin{aligned} &= \bar{C}\bar{D} + CD = C \oplus D \quad [\oplus \rightarrow \text{Ex-nor}] \end{aligned}$$

Third column ( $I_2$ )

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

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

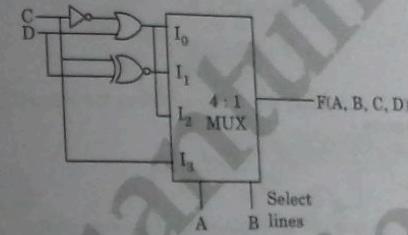
$$= \bar{C} + CD = \bar{C} + D$$

Fourth column ( $I_3$ )

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

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

Logic diagram :



- ii. Implementation using 2 : 1 MUX : We have to use three variables as input of MUX and one variable as select line.

Implementation table :

	$\bar{D}$	D
	$I_0$	$I_1$
$\bar{ABC}$	0	1
$\bar{ABC}$	2	3
$\bar{ABC}$	4	5
$\bar{ABC}$	6	7
$\bar{ABC}$	8	9
$\bar{ABC}$	10	11
$ABC$	12	13
$ABC$	14	15

First column ( $I_0$ )

$$\begin{aligned} &= \bar{A}BC + \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + ABC \\ &= \bar{A}\bar{C}(\bar{B} + \bar{B}) + A(\bar{B}\bar{C} + BC) \\ &= \bar{A}\bar{C} + A(B \odot C) \end{aligned}$$

Second column ( $I_1$ )

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

Logic diagram :

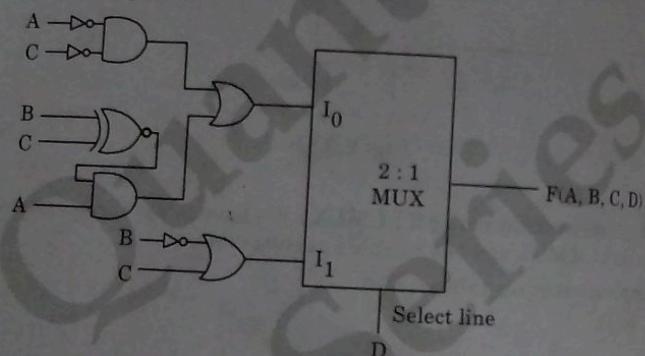


Fig. 2.9.2.

**Que 2.10.** Implement the function  $Y(A, B, C, D) = \sum m(0, 1, 2, 5, 8, 13, 14)$  using 8:1 multiplexer. Consider A, B, C as the select lines.

AKTU 2021-22 (Sem-4), Marks 10

**Answer**

The procedure is same as Q. 2.8, Page 2-9G, Unit-2.

Implementation :

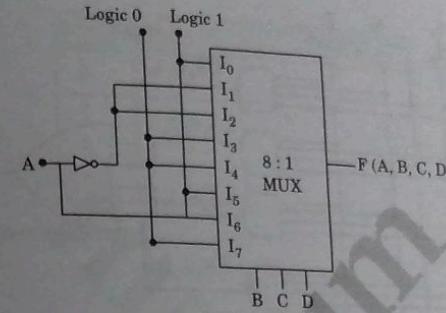


Fig. 2.10.1.

**Que 2.11.** Design 4:1 multiplexer using gates.

AKTU 2022-23 (Sem-3), Marks 10

**Answer**

4 : 1 multiplexer is a combinational circuit with four input lines  $I_0, I_1, I_2$  and  $I_3$  one output line 'O' and two selection lines  $S_0$  and  $S_1$ . On receiving active low enable input, multiplexer is able to perform function.

Truth table of 4 : 1 MUX is :

Input			Output
$\bar{E}$	$S_1$	$S_0$	O
1	x	x	0
0	0	0	$I_0$
0	0	1	$I_1$
0	1	0	$I_2$
0	1	1	$I_3$

The logic expression for 4 : 1 MUX is given as

$$O = \bar{E} \bar{S}_1 \bar{S}_0 I_0 + \bar{E} \bar{S}_1 S_0 I_1 + \bar{E} S_1 \bar{S}_0 I_2 + \bar{E} S_1 S_0 I_3$$

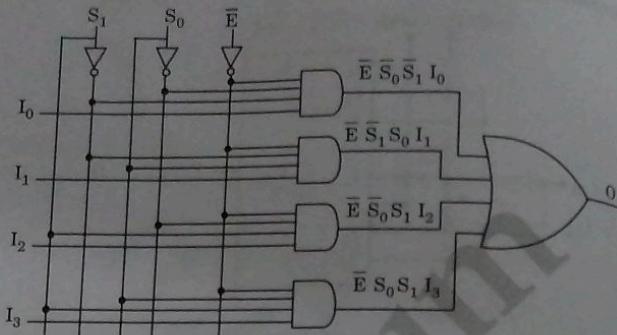


Fig. 2.11.1.

**PART-3****Demultiplexers.****Que 2.12.**

- What is demultiplexer ? Explain it.
- Can demultiplexer be used as a logic element ? If yes, what are its advantages over realization using gates ?

**Answer**

- A demultiplexer is a combinational circuit with one input line, two or more output lines and one or more selection lines. A demultiplexer, routes binary information present on the input line, to any one of the output lines. The output line that receives the information present on the input lines is selected by the bit status of the selection lines.
- Demultiplexers can be used as a logic element. It can be used to implement logic functions. Since it is an MSI circuit, it reduces IC package count and system count while implementing logic circuits than implementing the same using logic gates.

**Que 2.13. Design one to two-line demultiplexer.****Answer**

A one to two line De-MUX is a combinational circuit with one input line,  $I$ , two output lines  $O_0$  and  $O_1$  and selection line  $S_0$ .

**Truth table :**

Input		Output	
$\bar{E}$	$S_0$	$O_0$	$O_1$
1	x	0	0
0	0	$I$	0
0	1	0	$I$

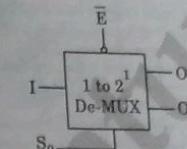
**Block diagram :**

Fig. 2.13.1.

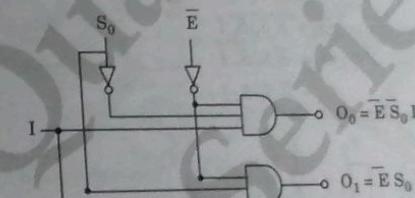
**Logic diagram :**

Fig. 2.13.2.

Boolean expressions for outputs are

$$O_0 = \bar{E} \bar{S}_0 I$$

$$O_1 = \bar{E} S_0 I$$

**Que 2.14. Design one to four line demultiplexer.****Answer**

Here, one input line  $I$ , four output lines  $O_0, O_1, O_2$  and  $O_3$ , 2 selection lines  $S_1$  and  $S_0$  are used.

Input			Output			
$\bar{E}$	$S_1$	$S_0$	$O_0$	$O_1$	$O_2$	$O_3$
1	*	*	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	1	1	0	0	0	1

Block diagram :

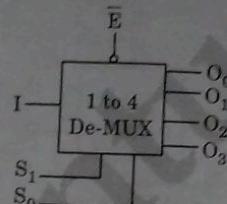


Fig. 2.14.1.

The boolean expressions are

$$O_0 = \bar{E} \bar{S}_1 \bar{S}_0 I$$

$$O_1 = \bar{E} \bar{S}_1 S_0 I$$

$$O_2 = \bar{E} S_1 \bar{S}_0 I$$

$$O_3 = \bar{E} S_1 S_0 I$$

Logic diagram :

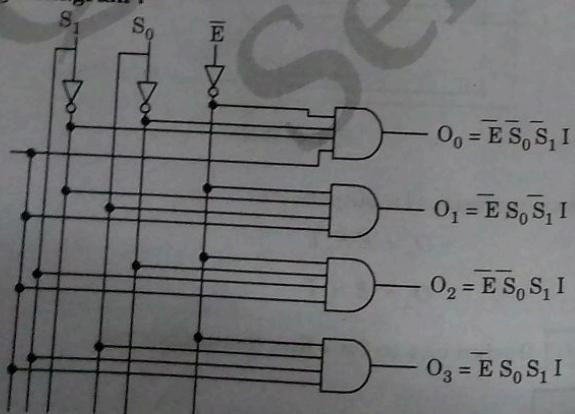


Fig. 2.14.2.

## PART-4

Decoders.

**Que 2.15.** Write a short note on decoder. Also draw the logic diagram for 2 to 4 decoder.

**Answer****A. Decoder :**

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.

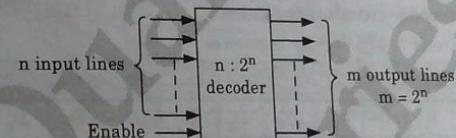


Fig. 2.15.1. Block diagram of a decoder.

**B. 2 to 4 binary decoder :**

Fig. 2.15.2 shows the 2 to 4 decoder. Here 2 represent the input lines and 4 represents output lines. Fig. 2.15.2 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. 2.15.2. Logic diagram of 2 to 4 decoder.

Que 2.16. Design a 3 : 8 decoder circuit using gates.

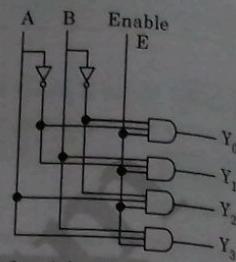
AKTU 2022-23 (Sem-3), Marks 10

**Answer**

3 : 8 decoder circuit using gates :

Truth table of 3 : 8 decoder is given as

Inputs			Outputs							
A	B	C	$D_0$	$D_1$	$D_2$	$D_3$	$D_4$	$D_5$	$D_6$	$D_7$
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1



2-20 G (OEC-Sem-3 &amp; 4)

Combinational Logic

Logic diagram :

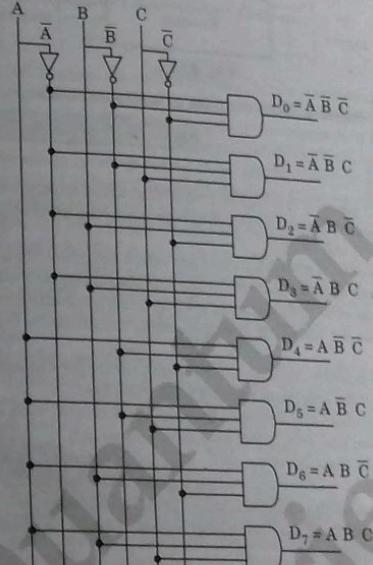


Fig. 2.16.1.

**PART-5**

Encoders, Multiplexed Display.

Que 2.17. 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. 2.17.1.

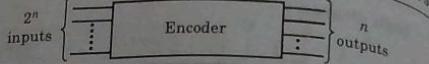


Fig. 2.17.1.

**Que 2.18.** Write a short note on priority encoder.

AKTU 2018-19 (Sem-3), Marks 10

**Answer**

1. A priority encoder is a digital circuit that takes multiple binary inputs and produces a binary output that represents the highest-priority input that is currently active.
2. In other words, if multiple inputs are active simultaneously, the priority encoder will output the code for the input with the highest priority.
3. Priority encoders are commonly used in interrupt controllers, which are responsible for handling multiple interrupt requests from different devices.
4. By using a priority encoder, the interrupt controller can ensure that the most important interrupt request is handled first.
5. Priority encoders are also used in other applications, such as analog-to-digital conversion and digital-to-analog conversion.
6. In analog-to-digital conversion, a priority encoder can be used to determine which of several analog signals is the largest.
7. In digital-to-analog conversion, a priority encoder can be used to determine which of several digital signals should be converted to an analog signal.
8. Priority encoders are relatively simple circuits, and they can be implemented using a variety of logic gates. The specific implementation of a priority encoder will depend on the number of inputs and the desired priority levels.

**Que 2.19.** Explain difference between encoder and decoder.

**Answer**

S. No.	Encoder	Decoder
1.	In this case, the applied signal is the active signal input.	Decoder accepts coded binary data as its input.
2.	The number of inputs accepted by an encoder is $2^n$ .	The number of input accepted by decoder is only $n$ inputs.
3.	Operation performed is simple.	Operation performed is complex.
4.	OR gate is the basic logic element used in it.	AND gate along with NOT gate is the basic logic element used in it.
5.	Output lines for an encoder is $n$ .	Output lines for decoder is $2^n$ .

**Que 2.20.** Discuss about multiplexed display.

**Answer**

1. Multiplexed displays are commonly used in combinational logic circuits to efficiently control and display information on multiple output devices, such as LEDs (Light Emitting Diodes) or 7-segment displays.
2. Here's how multiplexed displays work in combinational logic circuits:
  - i. **Multiple display elements :** Suppose you have multiple display elements (e.g., LEDs or 7-segment displays) that you want to control. Instead of dedicating individual pins or control lines for each display element, which can quickly become impractical, you use multiplexing.
  - ii. **Multiplexer (MUX) :** A multiplexer (MUX) is a combinational logic circuit that selects one of several input lines and routes it to the output based on control signals. In the context of multiplexed displays, the multiplexer is used to select which display element to illuminate at any given time.
  - iii. **Time division multiplexing :** The key idea in multiplexed displays is time-division multiplexing. In this method, you cycle through the display elements one at a time, rapidly switching between them. During each time slot, you display the desired information for that element.
  - iv. **Control logic :** Combinational logic circuits are used to generate the control signals for the multiplexer. These control signals determine which display element is active at any given moment.
  - v. **Synchronization :** To create the illusion of all display elements being active simultaneously, you need to synchronize the switching rate with the display's refresh rate. This typically involves using a clock signal to control the multiplexing process.

**PART-6**

Half and Full Adders.

**Que 2.21.** Describe half adder and full adder in brief. Implement the circuit using logic gates.

OR

Explain the design of a full adder, with its truth table and boolean expression.

AKTU 2021-22 (Sem-4), Marks 10

OR

Design a full adder using two half adders.

**Answer**

**Half adder :**

1. The block diagram of half adder is shown in Fig. 2.21.1.

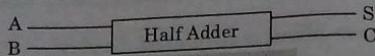


Fig. 2.21.1. Half adder.

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. 2.21.2.

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

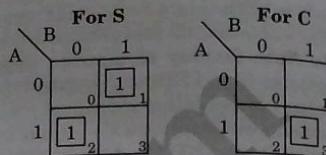


Fig. 2.21.2.

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. 2.21.3.

#### 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. 2.21.4.

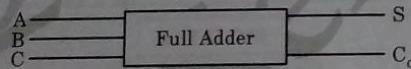


Fig. 2.21.4. Full Adder.

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

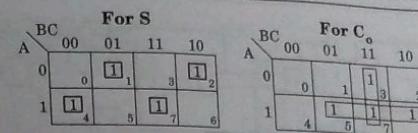


Fig. 2.21.5.

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

Carry :  $C_o = AB + AC + BC$

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

$$\begin{aligned} \text{Sum : } S &= ABC + A\bar{B}\bar{C} + \bar{A}BC + \bar{A}\bar{B}C \\ &= ABC + \bar{A}BC + A\bar{B}\bar{C} + \bar{A}\bar{B}C \\ &= C(AB + \bar{A}\bar{B}) + \bar{C}(A\bar{B} + \bar{A}B) \\ &= C(\bar{A}B + \bar{A}B) + \bar{C}(AB + \bar{A}B) \\ &= (A \oplus B) \oplus C \\ \text{Carry : } C_o &= AB + AC + BC \\ &= AB + C(A + B) \\ &= AB + C(A + \bar{A})(A + \bar{A})(B + \bar{B}) \\ &= AB + C[AB + A\bar{B} + \bar{A}B] \\ &= AB + ABC + C(\bar{A}B + \bar{A}B) \\ &= AB(1 + C) + C(A \oplus B) \\ &= AB + C(A \oplus B) \end{aligned}$$

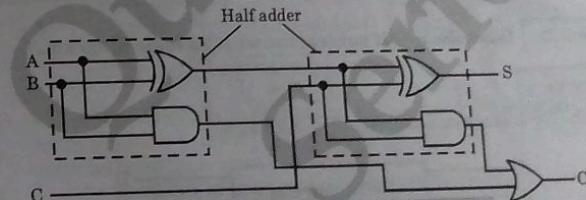


Fig. 2.21.6. Full adder circuit using 2 half adder.

- Que 2.22. Construct a full adder and implement the full adder with the help of half adders. Also implement the full adder with NAND gates only.

AKTU 2022-23 (Sem-4), Marks 10

#### Answer

- A. Full adder : Refer Q. 2.21, Page 2-22G, Unit-2.

- B. Full adder with NAND gates :

1. The full adder circuit can be realized using the NAND logic gates as shown in Fig. 2.22.1.

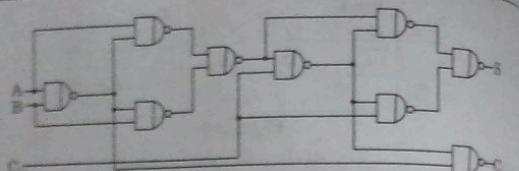


Fig. 2.22.1. Full adder with NAND gates.

2. Equation of the sum output for the full adder circuit with NAND gate is obtained as follows :

$$S = \overline{A \oplus B} \cdot \overline{(A \oplus B)C_n} \cdot C_n + \overline{(A \oplus B)C_n} \cdot A \oplus B = A \oplus B \oplus C_n$$

$$\text{where, } A \oplus B = \overline{A \cdot \overline{B}} + \overline{A} \cdot B$$

And equation of the carry output of the full adder circuit with NAND gate is given by,

$$C_{\text{out}} = \overline{C_n} \cdot \overline{(A \oplus B) \cdot AB} = AB + (A \oplus B)C_n$$

### PART-7

#### Subtractors.

**Que 2.23.** Describe a half subtractor with its logic diagram.

**Answer**

1. The block diagram is shown in Fig. 2.23.1.

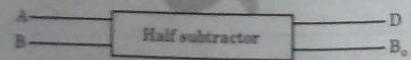


Fig. 2.23.1. Half subtractor.

2. It has two inputs, A (minuend) and B (subtrahend) and two outputs D (difference) and B\_out (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. 2.23.2.

Inputs		Outputs	
A	B	D	$B_{\text{out}}$
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

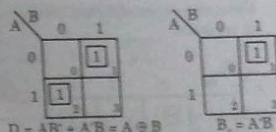


Fig. 2.23.2.

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

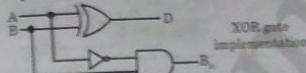


Fig. 2.23.3.

**Que 2.24.** Design a full subtractor circuit with three inputs  $x$ ,  $y$ ,  $B_{\text{in}}$  and two outputs 'Diff' and  $B_{\text{out}}$ . The circuit subtracts  $x - y - B_{\text{in}}$ , where,  $B_{\text{in}}$  is the input borrow,  $B_{\text{out}}$  is the output borrow and 'Diff' is the difference.

**Answer**

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

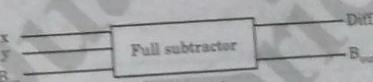


Fig. 2.24.1.

2. Fig. 2.24.1 shows the block diagram approach of full subtractor. It has three inputs  $x$ ,  $y$  and  $B_{\text{in}}$  and two outputs 'Diff' and  $B_{\text{out}}$  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	$B_{\text{in}}$	Diff	$B_{\text{out}}$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
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).

	x	y	$B_{in}$	00	01	11	10
0	0	0	0	1	1	3	1
1	1	1	1	4	5	7	6

	x	y	$B_{in}$	00	01	11	10
0	0	0	0	0	1	1	1
1	1	1	1	4	5	7	6

$$\text{Diff} = x'yB_{in} + x'y'B_{in}' + xy'B_{in}' + xyB_{in}$$

$$B_{out} = x'B_{in} + yB_{in} + x'y$$

Fig. 2.24.2.

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

$$\begin{aligned} \text{Diff} &= xyB_{in} + xy'B_{in}' + x'yB'_{in} + x'y'B_{in} \\ &= B_{in}(xy + x'y') + B'_{in}(xy' + x'y) \\ &= B_{in}(x \oplus y)' + B'_{in}(x \oplus y) = (x \oplus y) \oplus B_{in} \\ \text{and } B_{out} &= x'y + x'B_{in} + yB_{in} = x'y + B_{in}(x' + y) \\ &= x'y + B_{in}(x' + y)(x + x') (y + y') \\ &= x'y + B_{in}[x'y + xy + x'y'] = x'y + x'yB_{in} + B_{in}(xy + x'y) \\ &= x'y(B_{in} + 1) + B_{in}(x \oplus y)' = x'y + B_{in}(x \oplus y)' \end{aligned}$$

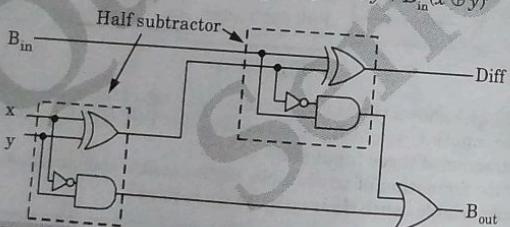


Fig. 2.24.3. Full subtractor circuit using 2 half subtractor.

Que 2.25. Draw a full subtractor circuit using NAND gate.

AKTU 2018-19 (Sem-3), Marks 3.5

Answer

Full subtractor using only NAND gates

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

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

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

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

By using the above expressions for D and  $B_{out}$ , the full subtractor is implemented using only NAND gates as shown in Fig. 2.25.1.

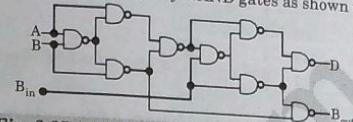


Fig. 2.25.1. Full subtractor using only NAND gate.

## PART-B

Serial and Parallel Adders.

Que 2.26. Explain the concept of serial adder with accumulators.

Answer

1. The block diagram of serial adder with accumulators is shown in Fig. 2.26.1.

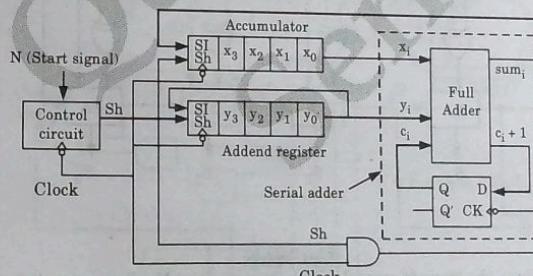


Fig. 2.26.1.

2. The full adder is used to perform bit by bit addition and D-Flip flop is used to store the carry output generated after addition.  
 3. This carry is used to carry input for the next addition. Initially the D Flip flop is cleared and addition starts with the least significant bits of both register.

4. After each clock pulse data within the right shift registers are shifted right 1-bit and we get from next digit and carry of previous addition new inputs for the full adder.

5. Truth table is shown as below :

	X	Y	$c_i$	sum <sub>i</sub>	$c_{i+1}$
t <sub>0</sub>	0101	0111	0	0	1
t <sub>1</sub>	0010	1011	1	0	1
t <sub>2</sub>	0001	1101	1	1	1
t <sub>3</sub>	1000	1110	1	1	0
t <sub>4</sub>	1100	0111	0	(1)	(0)

**Que 2.27.** Design a 4 bit parallel binary Adder/subtractor circuit  
OR

Design a 4-bit adder circuit using gates.

AKTU 2022-23 (Sem-3), Marks 10

#### Answer

The addition and subtraction operations can be combined into one circuit with one common binary adder by including an exclusive OR gate with full adder.

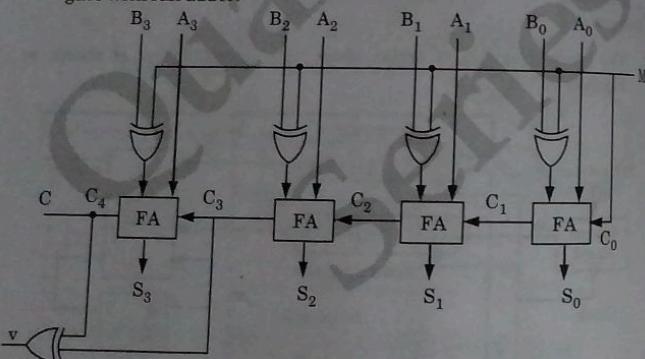


Fig. 2.27.1.

Here, the Mode  $M$  controls the operation when  $M = 0$ , the circuit is an adder whereas when  $M = 1$ , the circuit is subtractor.

Each exclusive - OR gate receives input  $M$  and one of the inputs of  $B$ . When  $M = 0$ , we have  $B \oplus 0 = B$ .

Thus, the full adder receives the value of  $B$ , the input carry is 0, and the circuit performs  $A$  plus  $B$ , when  $M = 1$ , we have  $B \oplus 1 = B'$  and  $C_0 = 1$ . The  $B$  inputs are all complemented and a 1 is added through the input carry.

#### PART-9

##### BCD Adder.

**Que 2.28.** Draw a BCD adder circuit and explain its working.

AKTU 2018-19 (Sem-3), Marks 3.5

OR

Draw a decimal adder to add BCD numbers.

AKTU 2017-18 (Sem-3), 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). A BCD adder circuit must be able to do the following and it is shown in Fig. 2.28.1.
  - Add two 4-bit BCD numbers using straight binary addition.
  - If 4-bit sum is equal to or less than 9, the sum is a valid BCD number and no correction is needed.
  - 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.

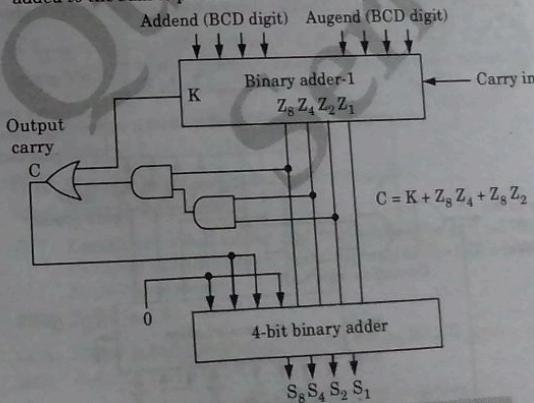


Fig. 2.28.1. Block diagram of a BCD adder.

4. After each clock pulse data within the right shift registers are shifted right 1-bit and we get from next digit and carry of previous addition new inputs for the full adder.
5. Truth table is shown as below :

	X	Y	$c_i$	sum <sub>i</sub>	$c_{i+1}$
$t_0$	0101	0111	0	0	1
$t_1$	0010	1011	1	0	1
$t_2$	0001	1101	1	1	1
$t_3$	1000	1110	1	1	0
$t_4$	1100	0111	0	(1)	(0)

Que 2.27. Design a 4 bit parallel binary Adder/subtractor circuit  
OR

Design a 4-bit adder circuit using gates.

AKTU 2022-23 (Sem-3), Marks 10

**Answer**

The addition and subtraction operations can be combined into one circuit with one common binary adder by including an exclusive OR gate with full adder.

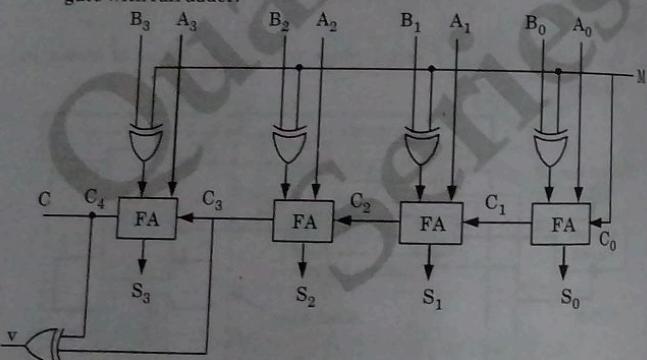


Fig. 2.27.1.

Here, the Mode  $M$  controls the operation when  $M = 0$ , the circuit is an adder whereas when  $M = 1$ , the circuit is subtractor.

Each exclusive - OR gate receives input  $M$  and one of the inputs of  $B$ . when  $M = 0$ , we have  $B \oplus 0 = B$ .

Thus, the full adder receives the value of  $B$ , the input carry is 0, and the circuit performs  $A$  plus  $B$ , when  $M = 1$ , we have  $B \oplus 1 = B'$  and  $C_0 = 1$ . The  $B$  inputs are all complemented and a 1 is added through the input carry.

**PART-9**

BCD Adder.

Que 2.28. Draw a BCD adder circuit and explain its working.

AKTU 2018-19 (Sem-3), Marks 3.5

OR

Draw a decimal adder to add BCD numbers.

AKTU 2017-18 (Sem-3), 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). A BCD adder circuit must be able to do the following and it is shown in Fig. 2.28.1.

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)<sub>2</sub> should be added to the sum to produce the valid BCD symbols.

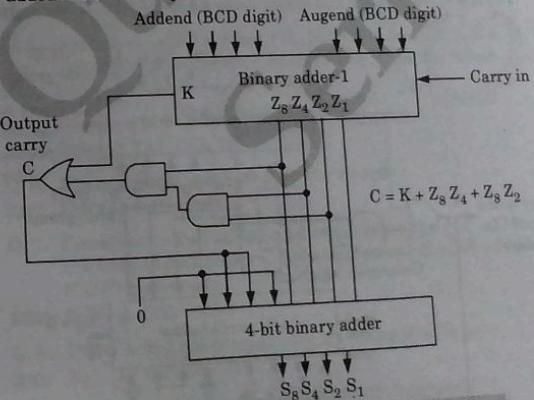


Fig. 2.28.1. Block diagram of a BCD adder.

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
1	0	0	0	0	1	0	1	1	7
1	0	0	1	0	0	1	0	0	8
1	0	1	0	1	0	0	0	1	9
1	0	1	1	1	0	0	0	0	10
									11

Que 2.29. Design a BCD adder using 4-bit parallel adder.

AKTU 2022-23 (Sem-4), Marks 10

### Answer

1. A BCD adder is a combinational circuit that adds two BCD numbers. It can be designed using a 4-bit parallel adder with some additional logic to correct the sum if it is greater than 10.
2. The block diagram of a BCD adder using a 4-bit parallel adder is shown in Fig. 2.29.1.

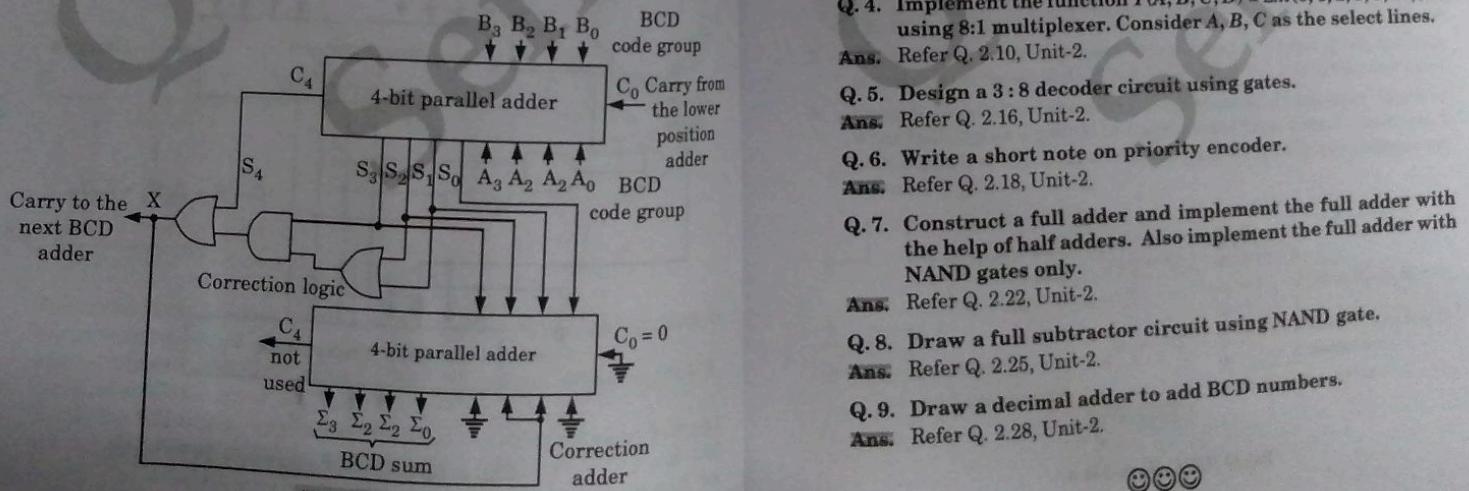


Fig. 2.29.1.

3. The 4-bit parallel adder adds the two input BCD numbers and produces a 4-bit binary sum. The output of the parallel adder is then fed into the correction logic.
4. The correction logic detects if the sum is greater than 9. If it is, then the correction logic adds 6 to the sum to produce a valid BCD sum.
5. Second BCD adder, the carry-out of the second BCD adder is BCD adder is connected as the carry-in of the third BCD adder and so on.

### VERY IMPORTANT QUESTIONS

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

- Q. 1. 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.  
Ans. Refer Q. 2.2, Unit-2.
- Q. 2. Draw and explain 2-bit magnitude comparator. Also represent output with the help of logic diagram.  
Ans. Refer Q. 2.4, Unit-2.
- Q. 3. Draw and explain 4-bit magnitude comparator.  
Ans. Refer Q. 2.6, Unit-2.
- Q. 4. Implement the function  $Y(A, B, C, D) = \Sigma m(0, 1, 2, 5, 8, 13, 14)$  using 8:1 multiplexer. Consider A, B, C as the select lines.  
Ans. Refer Q. 2.10, Unit-2.
- Q. 5. Design a 3 : 8 decoder circuit using gates.  
Ans. Refer Q. 2.16, Unit-2.
- Q. 6. Write a short note on priority encoder.  
Ans. Refer Q. 2.18, Unit-2.
- Q. 7. Construct a full adder and implement the full adder with the help of half adders. Also implement the full adder with NAND gates only.  
Ans. Refer Q. 2.22, Unit-2.
- Q. 8. Draw a full subtractor circuit using NAND gate.  
Ans. Refer Q. 2.25, Unit-2.
- Q. 9. Draw a decimal adder to add BCD numbers.  
Ans. Refer Q. 2.28, Unit-2.



# 3

UNIT

## Sequential Logic and its Applications

### CONTENTS

- Part-1 :** Storage Elements : ..... 3-2G to 3-9G  
Latches and Flip-Flops,  
Characteristic Equations of Flip-Flops
- Part-2 :** Flip-Flop Conversion ..... 3-9G to 3-14G
- Part-3 :** Shift Registers ..... 3-14G to 3-20G
- Part-4 :** Ripple Counters, ..... 3-20G to 3-32G  
Synchronous Counters
- Part-5 :** Other Counters : ..... 3-32G to 3-34G  
Johnson and Ring Counter

#### PART-1

*Storage Elements : Latches and Flip-Flops,  
Characteristic Equations of Flip-Flops.*

- Que 3.1.** What do you mean by storage elements and latches ?  
Describe the operation of SR latch.

#### Answer

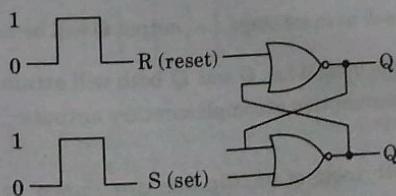
**Storage element :** A storage element in a digital circuit can maintain a binary state indefinitely (as long as power is delivered to the circuit), until directed by an input signal to switch states.

#### Latches :

1. Latches are said to be level sensitive devices, flip-flops are edge sensitive devices. Latches are the basic circuits from which all flip-flops are constructed.
2. Although latches are useful for storing binary information and for the design of asynchronous sequential circuits, they are not practical for use as storage elements in synchronous sequential circuits because they are the building blocks of flip-flop.

#### SR latch :

1. The SR latch is a circuit with two cross-coupled NOR gates or two-coupled NAND gates, and two inputs labeled S for 'set' and R for 'reset'. The SR latch constructed with two cross-coupled NOR gate is shown in Fig. 3.1.1.
2. The latch has two useful states. When output  $Q = 1$  and  $Q' = 0$ , the latch is said to be in the set state. When  $Q = 0$  and  $Q' = 1$ , it is in the reset state.
3. Outputs Q and Q' are normally the complement of each other. However when both inputs are equal to 1 at the same time, a condition in which both outputs are equal to 0 (rather than be mutually complementary) occurs.



(a) Logic diagram

S	R	Q	$Q'$
1	0	1	0
0	0	1	0 (after S = 1, R = 0)
0	1	0	1
0	0	0	1 (after S = 0, R = 1)
1	1	0	0 (forbidden)

(b) Function table

4. If both inputs are then switched to 0 simultaneously, the device will enter an unpredictable or undefined state or a metastable state. Consequently, in practical applications, setting both inputs to 1 is forbidden.

**Que 3.2.** What is flip-flop? Draw the logic diagram and give the characteristic table of SR flip-flop.

OR

Describe the working of the following flip-flops :

- i. SR      ii. JK      iii. T      iv. D

**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.

**i. SR flip-flop :**

1. The circuit diagram and truth table of SR flip-flop are shown in Fig. 3.2.1. 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  $\bar{Q}$ ).

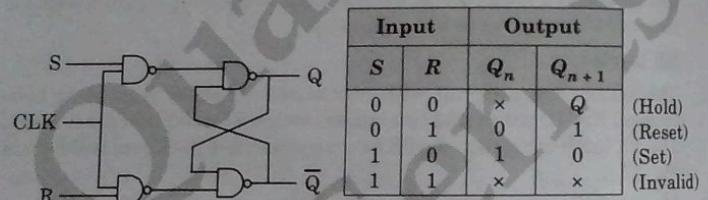
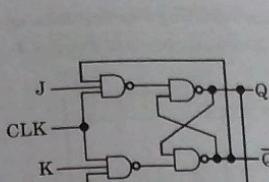


Fig. 3.2.1. SR Flip-flop.

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.
- ii. JK flip-flop :**
1. The circuit diagram and truth table of JK flip-flop are shown in Fig. 3.2.2.



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. 3.2.2. JK flip-flop.

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 :
  - a. Master Slave JK flip-flop.
  - b. Edge-triggered flip-flop.

**iii. T flip-flop :**

1. The T flip-flop is also known by the name of toggling flip-flop. The diagram for T flip-flop along with the truth table is shown in Fig. 3.2.3.

T	$Q_n$	$Q_{n+1}$
0	0	0
0	1	1
1	0	1
1	1	0

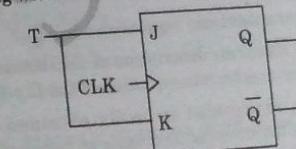


Fig. 3.2.3.

2. Toggling flip-flop means the output toggles between 0 and 1 when input is  $T = 1$  at  $CLK = 1$ . If  $J = K = 1$  in JK flip-flop the resulting flip-flop is known as T flip-flop.

**iv. D flip-flop :**

1. D flip flop is also known as delay flip-flop because the output follows the input after a clock pulse.

2. In JK flip-flop when  $J = \bar{K}$  then the resulting flip-flop acts as a flip-flop. The diagram along with truth table is shown in Fig. 3.2.4.

D	$Q_n$	$Q_{n+1}$
0	0	0
0	1	0
1	0	1
1	1	1

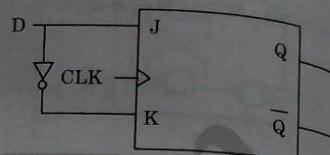


Fig. 3.2.4.

3. It is either used as a delay device or as latch to store 1-bit of binary information.

**Que 3.3.** Write the difference between latches and flip-flops.

**Answer**

S.No.	Latch	Flip-flop
1.	It is level triggered.	It is edge triggered.
2.	There is no clock pulse.	There is a clock pulse.
3.	It is used in asynchronous sequential logic circuit.	It is used in synchronous sequential logic circuit.

**Que 3.4.** Obtain the characteristic equation of SR flip-flop and D flip-flop.

**Answer**

**Characteristic equation :**

- The algebraic description of the next state ( $Q_{n+1}$ ) of a flip-flop is called the characteristic equation of the flip-flop.
- This expression is easily obtained by constructing the K-map simplification for next state in terms of the present state and flip-flop inputs.

**For SR flip-flop :**

The SR flip-flop has two inputs R and S. The truth table and K-map for SR flip-flop are shown in Fig. 3.4.1.

Truth table of SR flip-flop

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

For $Q_{n+1}$	
SQ <sub>n</sub>	R
00	1
01	1
11	3
10	2
0	0
1	1
4	5
5	7
7	x
6	x

Fig. 3.4.1.

The characteristic equation of SR flip-flop is

$$Q_{n+1} = S + \bar{R}Q_n$$

**For D flip-flop :** D flip-flop has one input and one output : The truth table and K-map are shown in Fig. 3.4.2.

Truth table of D flip-flop.

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

For $Q_{n+1}$	
D	$Q_n$
0	0
1	1

Fig. 3.4.2.

The characteristic equation of D flip-flop is

$$Q_{n+1} = D$$

**Que 3.5.** Obtain the characteristic equation of JK flip-flop and T flip-flop.

OR

Elaborate the characteristic equations of S-R and J-K flip-flops.

AKTU 2022-23 (Sem-3), Marks 10

OR

For the clocked JK flip-flop write the state table, state equation with state diagram.

AKTU 2022-23 (Sem-4), Marks 10

**Answer**

**Characteristic equation of SR flip-flop :** Refer Q. 3.4, Page 3-12, Unit-3.

**Clocked JK flip-flop and state table :** Refer Q. 3.2, Page 3-10, Unit-3.

**For JK flip-flop :**

The characteristic equation of JK flip-flop is obtained by using 3 variables K-map. The truth table and K-map for JK flip-flop are shown in Fig. 3.5.1.

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

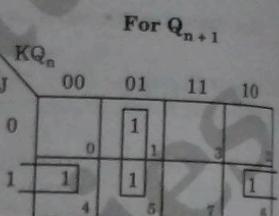


Fig. 3.5.1.

The characteristic equation (state equation) of JK flip-flop is:

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

**State diagram of JK flip-flop :**

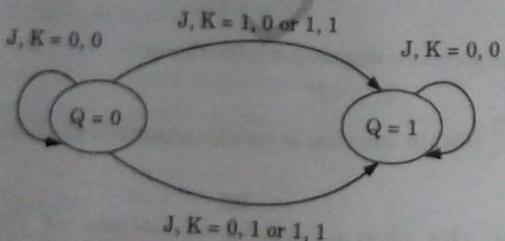


Fig. 3.5.2.

**For T flip-flop :** The truth table and K-map for T-flip-flop are shown in Fig. 3.5.3.

Truth table of T flip-flop.

Flip-flop inputs		Present state	Next state
T	Q <sub>n</sub>	Q <sub>n+1</sub>	
0	0	0	
0	1	1	
1	0	1	
1	1	0	

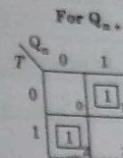


Fig. 3.5.3.

The characteristic equation of T flip-flop is

$$Q_{n+1} = \bar{T}Q_n + T\bar{Q}_n$$

**Que 3.6.** What is race around condition in JK flip-flop ?

**Answer**

Race around condition in JK flip-flop :

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.

i.e.,  $J = K = 1$   
and  $t_{pdFF} < t_{PW}$

**Que 3.7.** Discuss excitation table for SR, JK, T and D flip flop.

AKTU 2022-23 (Sem-4), Marks 10

**Answer**

Excitation table for SR flip-flop :

Present state of Q output	Next state of Q output	S input	R input
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

## Excitation table for JK flip-flop :

Q Output		Inputs	
Present state	Next state	J	K
Case I 0	0	0	x
Case II 0	1	1	x
Case III 1	0	x	1
Case IV 1	1	x	0

Don't care  
conditional  
result due  
to toggling

## Excitation table for T flip-flop :

Q output		Input
Present state	Next state	T
0	0	0
0	1	1
1	0	1
1	1	0

## Excitation table for D flip-flop :

Q output		Input
Present state	Next state	D
0	0	0
0	1	1
1	0	0
1	1	1

## PART-2

Flip-Flop Conversion.

**Que 3.8.** Analyze SR flip-flop using NAND-NAND logic and obtain its characteristic equation and excitation table. Explain how will you convert it in D flip-flop.

AKTU 2021-22 (Sem-4), Marks 10

OR

Explain how will you convert SR flip-flop into D flip-flop ?

## Answer

**SR flip-flop using NAND-NAND logic :** Refer Q. 3.2, Page 3-3G, Unit-3.

**Characteristic equation of SR flip-flop :** Refer Q. 3.4, Page 3-5G, Unit-3.

**SR flip-flop to D flip-flop :**

Truth table for D flip-flop

Excitation table for SR flip-flop

Inputs		States	
D	Q <sub>n</sub>	Q <sub>n+1</sub>	
0	0	0	
1	0	1	
0	1	0	
1	1	1	

Present state		Next state		Flip-flop Inputs	
Q <sub>n</sub>	Q <sub>n+1</sub>	S	R		
0	0	0	x		
0	1	1	0		
1	0	0	1		
1	1	x	0		

## Conversion table :

Inputs		Present state		Next state
D	Q <sub>n</sub>	Q <sub>n+1</sub>	S	R
0	0	0	0	x
0	1	0	0	1
1	0	1	1	0
1	1	1	x	0

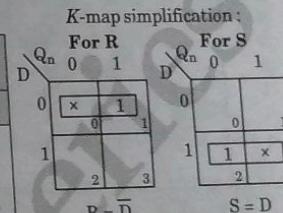


Fig. 3.8.1.

## D flip-flop using SR flip-flop :

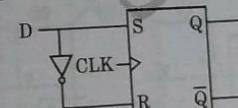


Fig. 3.8.2.

**Que 3.9.** Convert the SR flip-flop to JK flip-flop.

AKTU 2018-19 (Sem-3), Marks 3.5

## Answer

- The JK flip-flop is constructed by using SR flip-flop.

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		Next state	Flip-flop inputs	
$Q_n$	$Q_{n+1}$	R	S	
0	0	x	0	
1	1	0	0	
0	0	1	1	
1	1	0	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	0
1	0	0	1	0	1
1	0	1	1	0	x
1	1	0	1	0	1
1	1	1	0	1	0

3. K-map simplification :

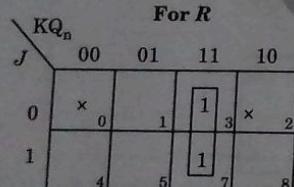


Fig. 3.9.1.

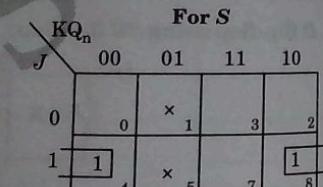


Fig. 3.9.2.

4. The obtained boolean expression :

$$S = J\bar{Q}_n \quad \text{and} \quad R = KQ_n$$

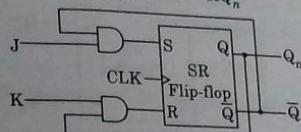


Fig. 3.9.3.

Que 3.10. Construct a D flip-flop using JK flip-flop.

**Answer**

1. The D flip-flop is constructed using JK flip-flop.

Given flip-flop is JK flip-flop and required flip-flop is D flip-flop

Truth table of D flip-flop

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

Excitation table of JK flip-flop

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

2. Form the conversion table

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

3. K-map simplification :

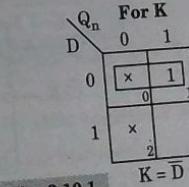
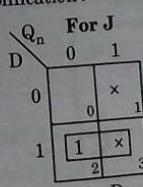


Fig. 3.10.1.

- Expressions are  $J = D$  and  $K = \bar{D}$   
 4. Logic diagram :

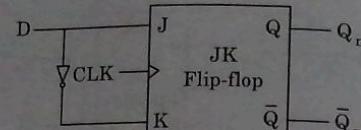


Fig. 3.10.2.

**Que 3.11.** Convert the given D flip-flop to T flip-flop.

**Answer**

1. The T flip-flop is constructed by using D flip-flop.

Truth table of T flip-flop

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

Excitation table of D flip-flop

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

2. Form the conversion table and K-map simplification.

Required flip-flop			Given flip-flop	
Flip-flop inputs	Present state	Next state	Flip-flop inputs	D
T	$Q_n$	$Q_{n+1}$		
0	0	0		0
0	1	1		1
1	0	1		1
1	1	0		0

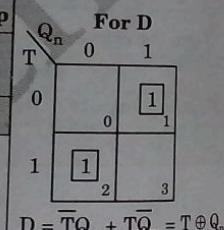


Fig. 3.11.1.

4. Logic diagram :

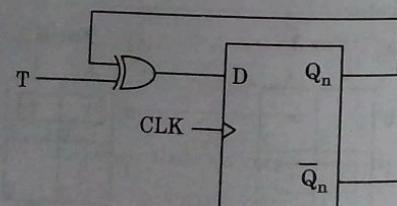


Fig. 3.11.2.

**Que 3.12.** Discuss the race around condition of JK flip-flop. How JK flip-flop can be used as T flip-flop, explain the design procedure.

AKTU 2021-22 (Sem-4), Marks 10

**Answer**

- A. **Race around condition of JK flip-flop :** Refer Q. 3.6, Page 3-8G, Unit-3.  
 B. **Conversion of JK into T flip-flop :**  
 1. T flip-flop is obtained by shorting J and K terminal with the same input.  
 2. Conversion table for the given JK flip-flop into T flip flop is

Input	Present state	Next state	Flip-flop inputs	
T	$Q_n$	$Q_{n+1}$	$J_A$	$K_A$
0	0	0	0	x
0	1	1	x	0
1	0	1	1	x
1	1	0	x	1

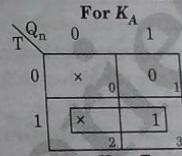
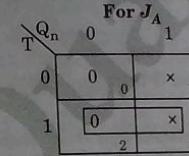


Fig. 3.12.1.

Logic diagram :

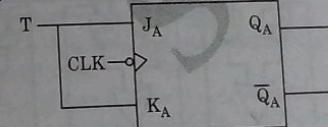


Fig. 3.12.2.

**PART-3**

Shift Registers.

**Que 3.13.** What do you mean by shift register? What is the need of shift register? Draw and explain bidirectional shift register.

AKTU 2018-19 (Sem-3), Marks 07

- Expressions are  
4. Logic diagram :  $J = D$  and  $K = \bar{D}$

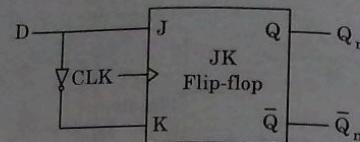


Fig. 3.10.2.

**Que 3.11.** Convert the given D flip-flop to T flip-flop.

**Answer**

1. The T flip-flop is constructed by using D flip-flop.

Truth table of T flip-flop

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

Excitation table of D flip-flop

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

2. Form the conversion table and K-map simplification.

Required flip-flop			Given flip-flop	
Flip-flop inputs	Present state	Next state	Flip-flop inputs	
T	$Q_n$	$Q_{n+1}$	D	
0	0	0	0	
0	1	1		1
1	0	1		1
1	1	0	0	

Fig. 3.11.1.

4. Logic diagram :

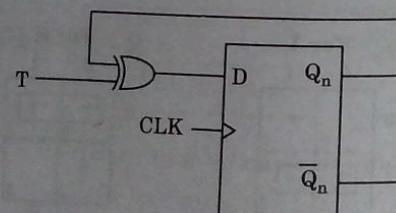


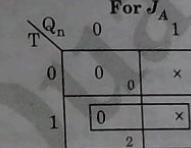
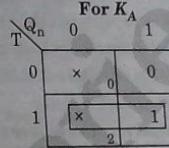
Fig. 3.11.2.

**Que 3.12.** Discuss the race around condition of JK flip-flop. How JK flip-flop can be used as T flip-flop, explain the design procedure.  
AKTU 2021-22 (Sem-4), Marks 10

**Answer**

- A. Race around condition of JK flip-flop : Refer Q. 3.6, Page 3-8G, Unit-3.
- B. Conversion of JK into T flip-flop :
1. T flip-flop is obtained by shorting J and K terminal with the same input.
  2. Conversion table for the given JK flip-flop into T flip flop is

Input	Present state	Next state	Flip-flop inputs
T	$Q_n$	$Q_{n+1}$	$J_A$ $K_A$
0	0	0	0      x
0	1	1	x      0
1	0	1	1      x
1	1	0	x      1

 $J_A = T$  $K_A = T$ 

Logic diagram :

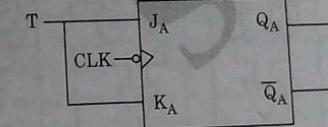


Fig. 3.12.2.

**PART-3**

Shift Registers.

**Que 3.13.** What do you mean by shift register ? What is the need of shift register ? Draw and explain bidirectional shift register.  
AKTU 2018-19 (Sem-3), Marks 07

OR

Write down the classification of shift registers.

**Answer****A. 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.

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

- Temporary storage.
- Shifting capability.

**C. Classification of shift registers :**

## 1. Classification based on the direction of data movement :

- Shift left register.
- Shift right register.
- Bidirectional shift registers.

**D. Classification based on the mode of input and output :**

- Serial in serial out shift register (SISO)
- Serial in parallel out shift register (SIPO)
- Parallel in serial out shift register (PISO)
- Parallel in parallel out shift register (PIPO)
- Universal shift register.

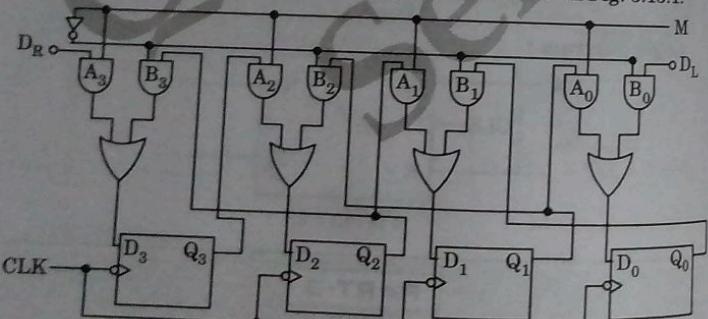
**E. Bidirectional shift register :** It consists of four D flip-flops, four OR gates, eight AND gates and one NOT gate as shown in Fig. 3.13.1.

Fig. 3.13.1.

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

OR

Draw and explain the PISO, PIPO register.

AKTU 2017-18 (Sem-3), Marks 07

OR

Elaborate the working and circuit of a Serial-in-Serial-Out shift register.

AKTU 2022-23 (Sem-3), Marks 10

**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. 3.14.1 shows the block diagram of serial in serial out shift register (SISO).

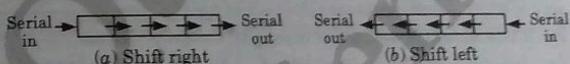


Fig. 3.14.1. 4-bit serial in serial out shift register.

**Shift right register :**

- In this register while accepting data serially, the group of bits is shifted towards the right side.
- Hence the serial data is entered onto the left side of register and it leaves from the right side serially. Fig. 3.14.2 shows the logic circuit for a 4-bit shift right register.

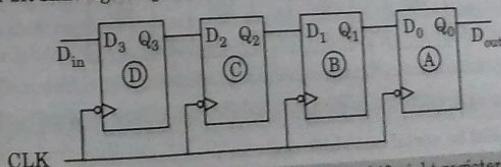


Fig. 3.14.2. Logic circuit for a 4-bit shift right register.

**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. 3.14.3 shows the logic circuit for a 4-bit shift left register.

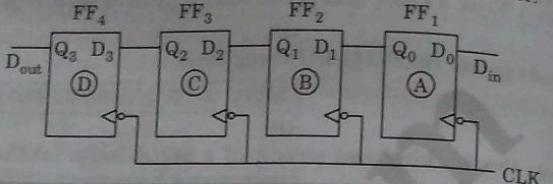


Fig. 3.14.3. Logic circuit for a 4-bit shift left register.

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. 3.14.4 shows the 4-bit serial in parallel out shift register.

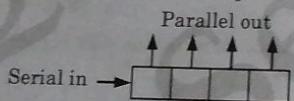


Fig. 3.14.4. Block diagram of 4-bit serial in parallel out shift register.

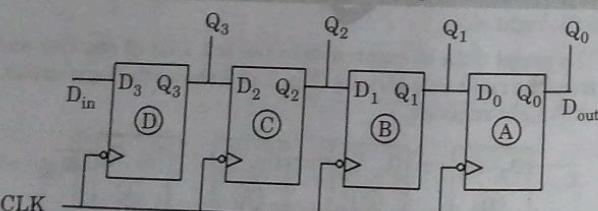


Fig. 3.14.5. Logic circuit for 4-bit serial in parallel out shift register.

**Parallel in serial out shift register :**

1. Fig. 3.14.6 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.

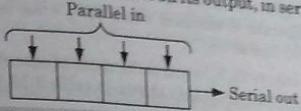


Fig. 3.14.6. Parallel in serial out shift register.

**Parallel in parallel out shift register :**

1. All the data appear simultaneously along with all the flip-flop inputs and outputs.
2. Fig. 3.14.7 shows the logic diagram for 4-bit parallel in parallel out shift register.

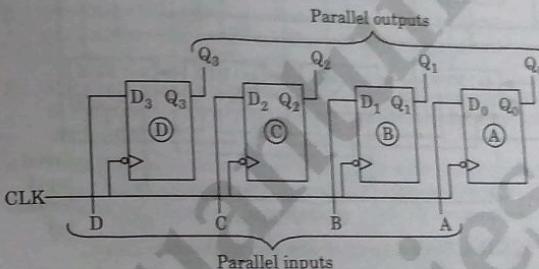


Fig. 3.14.7. Logic diagram for 4-bit parallel in parallel out shift register.

**Que 3.15.** With the help of diagram, explain the operation of universal shift register.

OR

Draw and explain 4-bit universal shift register.

OR

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

AKTU 2017-18 (Sem-3), 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. 3.15.1. 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  $I_1$  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 3.15.1.

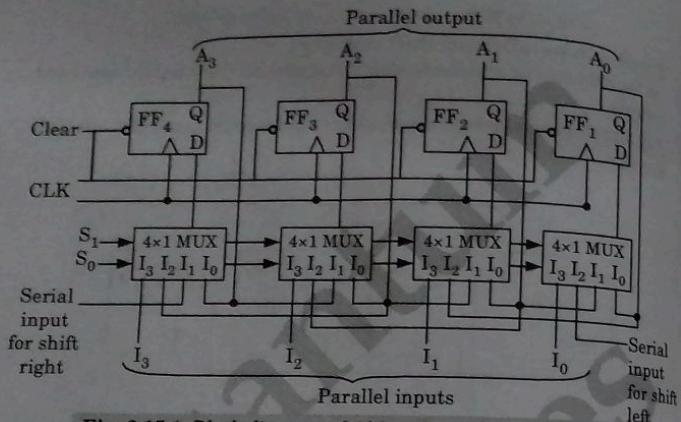


Fig. 3.15.1. Block diagram of 4-bit universal shift register.

Table 3.15.1. Function table

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

**Operation :**

- When  $S_1S_0 = 00$ , the present value of the register is applied to the  $D$  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-flops. The next clock edge transfers into each flip-flop the binary value it held previously and no change of state occurs.
- When  $S_1S_0 = 01$ , the input  $I_1$  of the multiplexer has a path to the  $D$  inputs of the flip-flops. This causes a shift right operation, with the serial input transferred in flip-flop  $FF_4$ .
- 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.

4. 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 3.16.** What are the applications of shift register ?

**Answer**

- Serial-to-parallel conversion :** Shift registers can be used to convert serial data into parallel data. This is useful when you have a serial data stream (one bit at a time) and need to process or output the data in parallel (multiple bits at a time).
- Parallel-to-serial conversion :** Conversely, shift registers can convert parallel data into serial data. This is helpful when you have parallel data but need to transmit it serially.
- Data storage :** Shift registers can be used for temporary data storage. Serial input data is shifted into the register and can be held there until it is needed.
- Serial data transmission :** Shift registers are commonly used in serial communication protocols such as SPI (Serial Peripheral Interface) and UART (Universal Asynchronous Receiver-Transmitter) to transmit data serially.

**PART-4**

Ripple Counters, Synchronous Counters.

**Que 3.17.** Design a 3-bit up/down ripple counter.

AKTU 2017-18 (Sem-3), Marks 07

**Answer**

- 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.
  - 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.
  - The control signal  $M$  is used to select the direction of count sequence. Fig. 3.17.1 shows the 3-bit ripple.
- $M = 1$ ; counter acts as up-counter  
 $M = 0$ ; counter acts as down counter

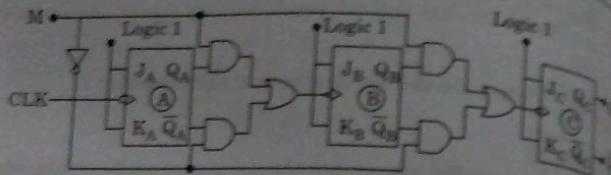


Fig. 3.17.1. 3-bit up/down ripple counter.

**Que 3.18.** 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 (Sem-3), Marks 12

**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  $Q_4$  must be externally connected to clock input of flip-flop.
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  $Q_0, Q_1, Q_2, Q_3$  outputs.

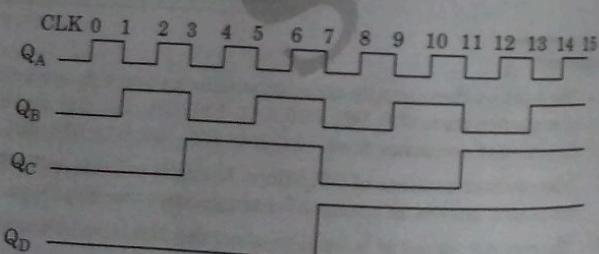
**Timing diagram :**

Fig. 3.18.1.

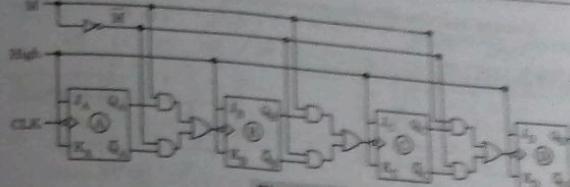
**Logic diagram :**

Fig. 3.18.2.

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

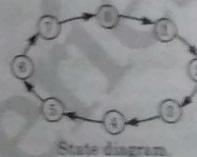
**Que 3.19.** Design a 3-bit synchronous counter using JK flip-flop.

**Answer****3-bit synchronous counter :**

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

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

Table 3.19.1 : 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

Fig. 3.19.1.

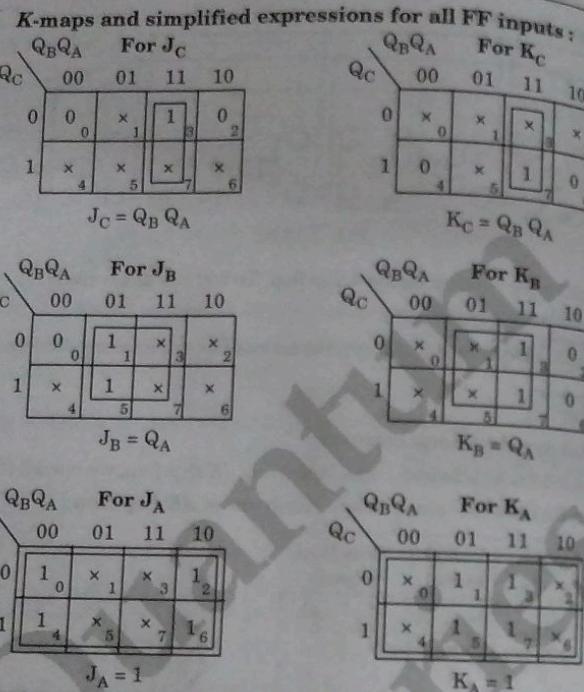


Fig. 3.19.2. K-maps and simplified equations for different FF inputs.  
Thus the simplified equations are :

$$J_C = Q_B Q_A \quad K_C = Q_B Q_A$$

$$J_B = Q_A \quad K_B = Q_A$$

$$J_A = 1 \quad K_A = 1$$

**Logic diagram :** Fig. 3.19.3 shows the logic diagram of a 3-bit synchronous counter using JK flip-flops.

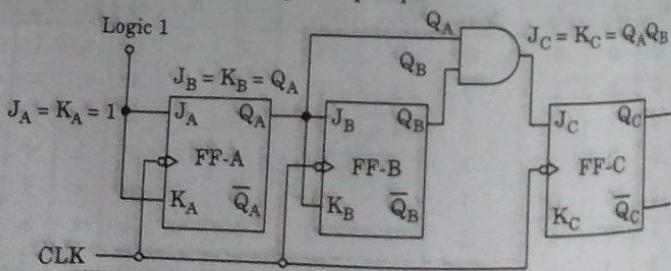


Fig. 3.19.3. 3-bit synchronous counter using JK FFs.

**Que 3.20.** Design and implement MOD-10 synchronous counter.

AKTU 2022-23 (Sem-4), Marks 10

**Answer**

**Step 1 :** Here 10 indicate total number of states, the count sequences are from 0 to 9.

**Step 2 :** The number of flip-flops required can be determined by using the following equation,

$$M \leq 2^N$$

Where

$$M = \text{MOD number is equal to 10}$$

$$10 \leq 2^N$$

$$\Rightarrow N = 4$$

Therefore, to design a MOD-10 counter, 4 flip-flops are required.

**Step 3 :** Excitation table for T flip-flop:

Previous state ( $Q_n$ )	Next stage ( $Q_{n+1}$ )	T
0	0	0
0	1	1
1	0	1
1	1	0

**Step 4 : State table:**

Present state	Next state				Flip-flop inputs							
	$Q_3$	$Q_2$	$Q_1$	$Q_0$	$Q'_3$	$Q'_2$	$Q'_1$	$Q'_0$	$T_3$	$T_2$	$T_1$	$T_0$
0 0 0 0	0	0	0	0	0	0	1	0	0	0	1	1
0 0 0 1	0	0	1	0	0	1	0	0	0	0	1	1
0 0 1 0	0	0	1	0	0	0	1	1	0	0	0	1
0 0 1 1	0	0	1	1	0	1	0	0	0	1	1	1
0 1 0 0	0	1	0	0	1	0	1	0	1	0	0	1
0 1 0 1	0	1	0	1	1	1	0	0	0	1	1	1
0 1 1 0	0	1	1	0	1	1	1	0	0	0	0	1
0 1 1 1	0	1	1	1	1	0	0	0	1	1	1	1
1 0 0 1	1	0	0	1	0	0	0	1	0	0	0	1
1 0 0 0	1	0	0	0	0	0	0	1	0	0	0	1

## Step 5 : K-map simplification

		Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	00	01	11	10	
		Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	00	01	11	10	
		Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	00	0	0	0	0
		01	Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	0	0	1	0
		11	Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	x	x	x	x
		10	Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	0	1	x	x

$$T_3 = Q_3Q_0 + Q_2Q_1Q_0$$

		Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	00	01	11	10	
		Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	00	0	0	1	0
		Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	00	0	0	1	0
		01	Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	x	x	x	x
		11	Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	0	0	x	x
		10	Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	0	0	x	x

$$T_2 = Q_1Q_0$$

		Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	00	01	11	10	
		Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	00	0	1	1	0
		Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	00	0	1	1	0
		01	Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	x	x	x	x
		11	Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	0	0	x	x
		10	Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	0	0	x	x

$$T_1 = Q_3Q_0$$

		Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	00	01	11	10	
		Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	00	1	1	1	1
		Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	00	1	1	1	1
		01	Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	1	1	1	1
		11	Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	1	1	1	1
		10	Q <sub>3</sub> Q <sub>2</sub>	Q <sub>1</sub> Q <sub>0</sub>	1	1	1	1

$$T_0 = 1$$

## Step 6 : Logic diagram :

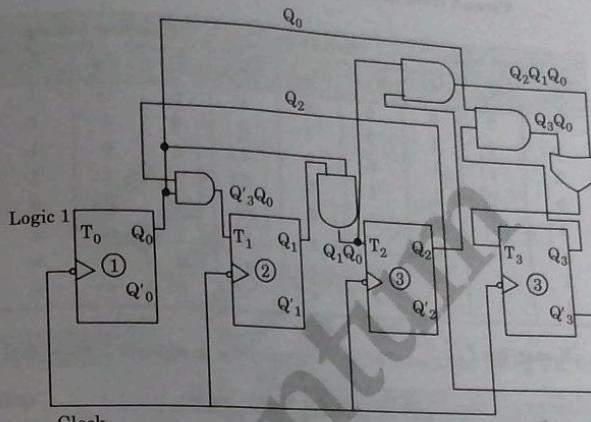


Fig. 3.20.1.

Que 3.21. Explain the working and circuit of a modulo-5 counter using gates.

AKTU 2022-23 (Sem-3), Marks 10

## Answer

Working and circuit of a modulo-5 counter using gates :

The number of flip flops required can be determined by using the following equation,

$$M \leq 2^N$$

where M = MOD number is equal to 5.

$$5 \leq 2^N$$

$$\Rightarrow N = 3$$

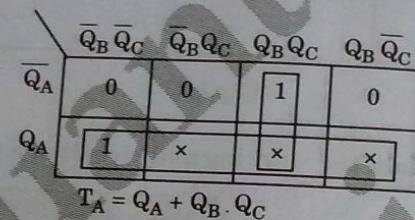
Therefore, to design a MOD 5 counter, 3 flip flops are required.

Excitation table of T flip flop

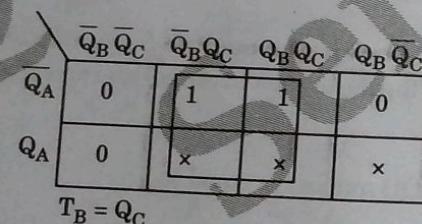
Present state Q <sub>n</sub>	Next state Q <sub>n+1</sub>	T
0	0	0
0	1	1
1	0	1
1	1	0

Circuit state table by using excitation table :

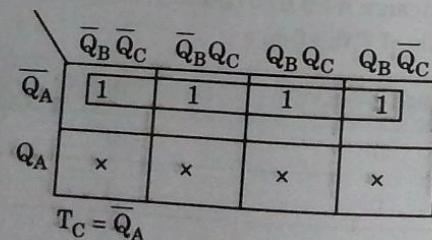
Present state			Next state			Flip-flop inputs		
$Q_A$	$Q_B$	$Q_C$	$Q_{A+1}$	$Q_{B+1}$	$Q_{C+1}$	$T_A$	$T_B$	$T_C$
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	1
1	0	0	0	0	0	1	1	1
1	0	1	x	x	x	x	x	0
1	1	0	x	x	x	x	x	x
1	1	1	x	x	x	x	x	x

K-map for  $T_A$  :

$$T_A = Q_A + Q_B \cdot Q_C$$

K-map for  $T_B$  :

$$T_B = Q_C$$

K-map for  $T_C$  :

$$T_C = \bar{Q}_A$$

Logic diagram :

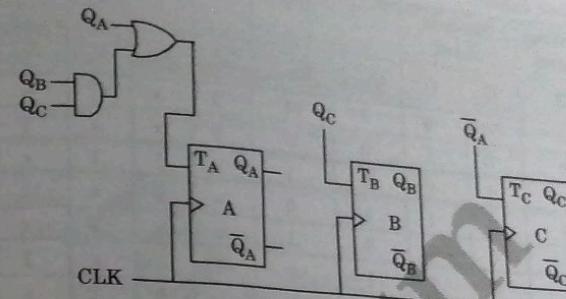


Fig. 3.21.1.

**Que 3.22.** Design a modulo-4 UP/DOWN counter using JK flip-flop.

AKTU 2018-19 (Sem-3), Marks 3.5

**Answer**

1. The count sequences of a modulo-4 up counter are 00, 01, 10 and 11. The count sequence of a modulo-4 down counter is 11, 10, 01 and 00.
2. Two flip flops are required to design modulo-4 UP/DOWN counter.

State diagram :

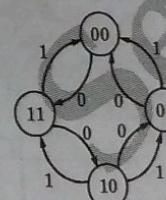


Fig. 3.22.1.

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	Present state		Next state		Flip-flop input				
	X	Q <sub>A</sub>	Q <sub>B</sub>	Q <sub>A+1</sub>	Q <sub>B+1</sub>	J <sub>A</sub>	K <sub>A</sub>	J <sub>B</sub>	K <sub>B</sub>
0	0	0	1	1	1	x		1	x
0	0	1	0	0	0	x		x	
0	1	0	0	1	x	1		1	x
0	1	1	1	0	x	0		x	
1	0	0	0	1	0	x		1	x
1	0	1	1	0	1	x		x	
1	1	0	1	1	x	0		1	x
1	1	1	0	0	x	1		x	1

K-map simplification :

Expression for J<sub>A</sub>

X\Q <sub>A</sub> Q <sub>B</sub>	00	01	11	10
0	0	1	x	2
1	4	5	7	6

$$J_A = \overline{X} \overline{Q}_B + XQ_B = X \oplus Q_B$$

Expression for K<sub>A</sub>

X\Q <sub>A</sub> Q <sub>B</sub>	00	01	11	10
0	0	1	x	2
1	4	5	7	6

$$K_A = \overline{X} \overline{Q}_B + XQ_B = X \oplus Q_B$$

Expression for J<sub>B</sub>

X\Q <sub>A</sub> Q <sub>B</sub>	00	01	11	10
0	0	1	x	1
1	4	5	7	6

$$J_B = 1$$

Expression for K<sub>B</sub>

X\Q <sub>A</sub> Q <sub>B</sub>	00	01	11	10
0	0	1	x	1
1	4	5	7	6

$$K_B = 1$$

Logic diagram :

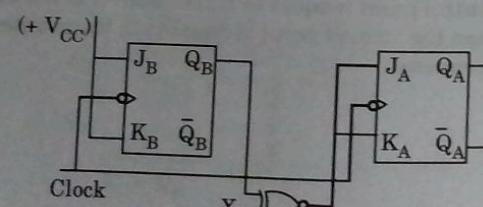


Fig. 3.22.2. Modulo-4 binary UP/DOWN counter.

Que 3.23. Design a 2 bit synchronous up counter.

Answer

Step 1 : To design a synchronous up counter, first we need to know the number of flip flops are required, we can find out by considering a number of bits mentioned in the question. So, in this, we required to make 2 bit counter so the number of flip flops required is 2.

Step 2 : After that, we need to construct a state table with an excitation table.

T flip flop excitation table :

Table 3.23.1:

Present state	Next state	T
0	0	0
0	1	1
1	0	1
1	1	0

Table 3.23.2 is created as follow : When Q<sub>2</sub> = 0 which is the present state and Q'<sub>2</sub> = 0 which is the next state then T<sub>2</sub> becomes 0. Similarly, if Q<sub>2</sub> is 0 and Q'<sub>2</sub> is 1 then T<sub>2</sub> becomes 1. In a similar way, it goes on.

State table with excitation table :

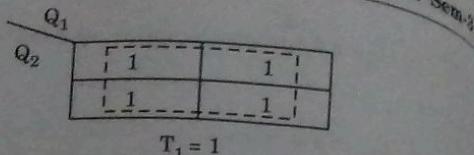
Table 3.23.2:

Present state		Next state		Flip flop	
Q <sub>2</sub>	Q <sub>1</sub>	Q' <sub>2</sub>	Q' <sub>1</sub>	T <sub>2</sub>	T <sub>1</sub>
0	0	0	1	0	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	0	0	1	1

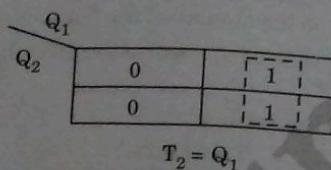
Step 3 : After that, we determine the equation from the boolean algebra or K map for the design of the counter. So for T<sub>1</sub> and T<sub>2</sub> we got 1 and Q<sub>1</sub>.

K-map :

For T<sub>1</sub> flip flop,



For  $T_2$  flip flop,



**Step 4:** According to the equation got from K map create the design of 2 bit synchronous up counter.

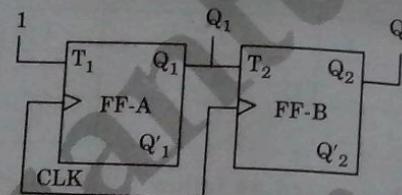


Fig. 3.23.1.

In the above design,  $T_1$  is getting high input and  $T_2$  is getting input from the output of the  $T_1$  flip-flop.

**Que 3.24.** Differentiate between synchronous and asynchronous counters. Design a 2 bit synchronous up counter.

AKTU 2021-22 (Sem-4), Marks 10

### Answer

**A. Difference between synchronous and asynchronous counters:**

S. No.	Synchronous counter	Asynchronous counter
1.	In synchronous counter, all flip flops are triggered with same clock simultaneously.	In asynchronous counter, different flip flops are triggered with different clock, not simultaneously.
2.	Synchronous counter is faster than asynchronous counter in operation.	Asynchronous counter is slower than synchronous counter in operation.

- |    |   |   |
|----|---|---|
| 3. | Synchronous counter does not produce any decoding errors. | Asynchronous counter produces decoding error.             |
| 4. | Synchronous counter is also called parallel counter.      | Asynchronous counter is also called serial counter.       |
| 5. | In synchronous counter, propagation delay is less.        | In asynchronous counter, there is high propagation delay. |

**B. Design of 2 bit synchronous up counter :** Refer Q. 3.23,  
Page 3-30G, Unit-3.

### PART-5

Other Counters : Johnson and Ring Counter.

**Que 3.25.** Explain 4 bit Johnson counter with circuit diagram and waveforms.

### Answer

- Fig. 3.25.1 shows a 4-bit Johnson counter. The last flip flop  $Q_3$  is connected to the input of first flip flop  $D_0$ . Since  $Q_3$  is 1, so at the rising pulse of the first clock cycle  $Q_0$  will change to 1.
- The output state will change to  $Q_3 Q_2 Q_1 Q_0 = 0001$  after first clock pulse, then at the rising pulse of the second clock cycle the output state will change  $Q_3 Q_2 Q_1 Q_0 = 0011$  and so on till at the rising pulse of the fourth clock pulse the output state changes to 1111.

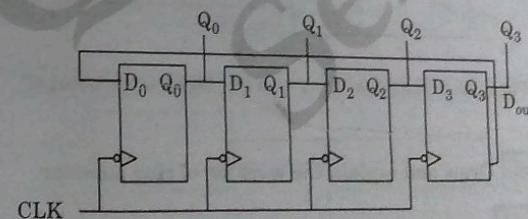


Fig. 3.25.1. Logic circuit for 4-bit Johnson counter.

- At the first rising pulse the state then becomes 1110, at the sixth pulse the state becomes 1100 and so on. At the eighth pulse it becomes 0000.
- Table 6 shows the truth table of Johnson counter.

Table 3.25.1.

Clock	$Q_0$	$Q_1$	$Q_2$	$Q_3$
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
8	0	0	0	0

5. Fig. 3.25.2 gives the timing sequence for a four-bit Johnson counter.

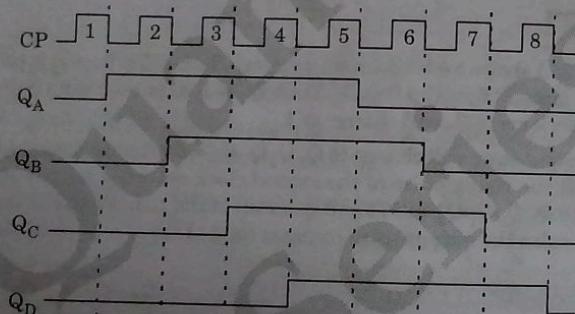


Fig. 3.25.2. Timing sequence for a four-bit Johnson counter.

**Que 3.26.** Draw and explain the working of ring counter.

**Answer**

1. Ring counter is a type of counter composed of flip flops connected into a shift register, with the output of the last flip flop fed to the input of the first, making a "circular" or "ring" structure.
2. The logic diagram of ring counter is shown in Fig. 3.26.1.
3. Here, the  $Q$  output of each stage is connected to the  $D$  input of the next stage and the output of last stage is fed back to the input of first stage to '1' and remaining outputs are zero, i.e.,  $Q_A$  is one and  $Q_B, Q_C, Q_D$  are zero.

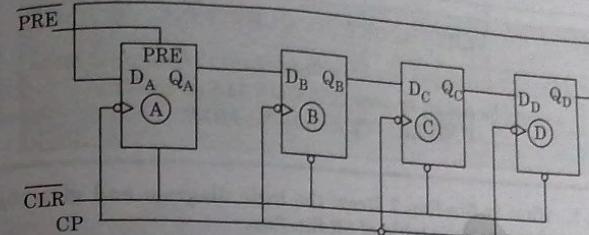


Fig. 3.26.1.

Clock Pulse	$Q_A$	$Q_B$	$Q_C$	$Q_D$
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	0	0	0

4. The first clock pulse produces  $Q_B = 1$  and remaining outputs are zero. According to the clock pulses applied at the clock input  $CP$ , a sequence of four states is produced.
5. The timing sequence for ring counter is shown in Fig. 3.26.2.

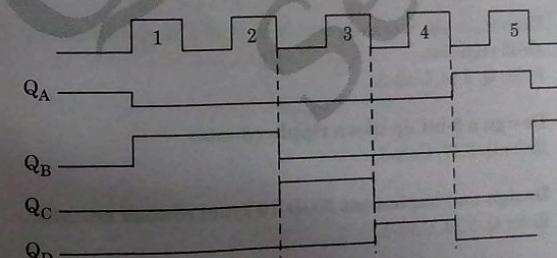


Fig. 3.26.2.

**VERY IMPORTANT QUESTIONS**

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

**Q. 1.** What is flip-flop ? Draw the logic diagram and give the characteristic table of SR flip-flop.

**Ans.** Refer Q. 3.2, Unit-3.

**Q. 2.** For the clocked JK flip-flop write the state table, state equation with state diagram.

**Ans.** Refer Q. 3.5, Unit-3.

**Q. 3.** Analyze SR flip-flop using NAND-NAND logic and obtain its characteristic equation and excitation table. Explain how will you convert it in D flip-flop.

**Ans.** Refer Q. 3.8, Unit-3.

**Q. 4.** Convert the SR flip-flop to JK flip-flop.

**Ans.** Refer Q. 3.9, Unit-3.

**Q. 5.** What do you mean by shift register ? What is the need of shift register ? Draw and explain bidirectional shift register.

**Ans.** Refer Q. 3.13, Unit-3.

**Q. 6.** Elaborate the working and circuit of a Serial-in-Serial-Out shift register.

**Ans.** Refer Q. 3.14, Unit-3.

**Q. 7.** Design a 3-bit up/down ripple counter.

**Ans.** Refer Q. 3.17, Unit-3.

**Q. 8.** Design and implement MOD-10 synchronous counter.

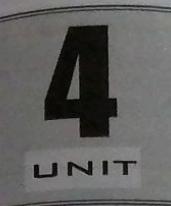
**Ans.** Refer Q. 3.20, Unit-3.

**Q. 9.** Design a modulo-4 UP/DOWN counter using JK flip-flop.

**Ans.** Refer Q. 3.22, Unit-3.

**Q. 10.** Differentiate between synchronous and asynchronous counters. Design a 2 bit synchronous up counter.

**Ans.** Refer Q. 3.24, Unit-3.



## Synchronous and Asynchronous Sequential Circuits

### CONTENTS

- Part-1 :** Analysis of Clocked Sequential Circuits with State Machine Designing ..... 4-2G to 4-10G
- Part-2 :** State Reduction and Assignments, Design Procedure ..... 4-11G to 4-14G
- Part-3 :** Analysis Procedure of Asynchronous Sequential Circuits ..... 4-15G to 4-21G
- Part-4 :** Circuit with Latches, Design Procedure ..... 4-21G to 4-24G
- Part-5 :** Reduction of State and Flow Table ..... 4-25G to 4-27G
- Part-6 :** Race Free State Assignment, Hazards ..... 4-27G to 4-32G

**PART - 1***Analysis of Clocked Sequential Circuits with State Machine Designing.*

**Que 4.1.** Discuss the analysis of clocked sequential circuits.

**Answer**

1. For the analysis of sequential circuit, we start with the logic diagram.
2. The excitation equation or boolean expression of each flip-flop is derived from this logic diagram.
3. To obtain the next state equation, we insert the excitation equation into the characteristic equations.
4. The output equations can be derived from the schematic. We can generate the state table using output and next state equations.

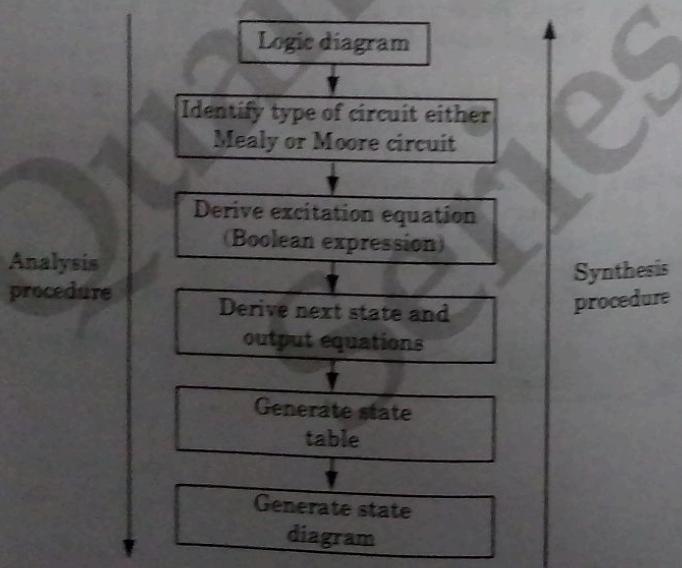


Fig. 4.1.1. Analysis and synthesis procedure of sequential circuits.

**Que 4.2.** 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 function of the present state and inputs a set of state equations for the circuit :

$$A(t+1) = A(t)x(t) + B(t)y(t)$$

$$B(t+1) = A(t)x(t)$$

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

$$y(t) = A(t) + B(t)x(t)$$

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

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

**State table :**

1. 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. 4.2.1 is shown in Table 4.2.1.
2. The Table 4.2.1 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  $t$ .
3. 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  $(t+1)$ .
4. The output section gives the value of  $y$  at time  $t$  for each present state and input condition.
5. The next state of flip-flop  $A$  must satisfy the state equation

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

Table 4.2.1.

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

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

$$B(t+1) = Ax$$

7. The output column is derived from the output equation

$$y = Ax' + Bx'$$

**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. 4.2.2.

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

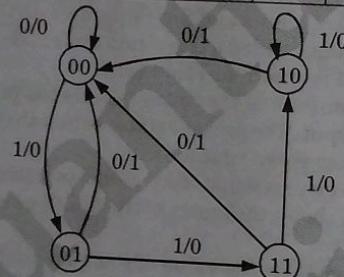


Fig. 4.2.1. State diagram of the circuit.

3. The state diagram provides the same information as the state table and is obtained directly from Table 4.2.1 or Table 4.2.2.

**Que 4.3.** 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.

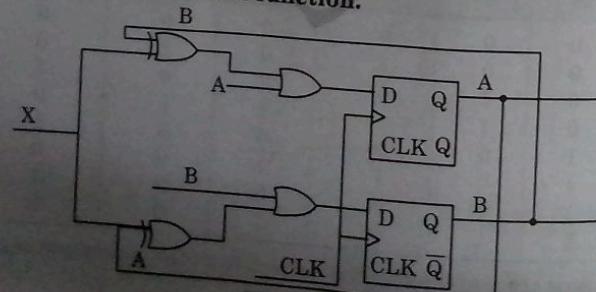


Fig. 4.3.1.

**Answer**

1. From the circuit shown in Fig. 4.3.1, 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 Table 4.3.1.

Table 4.3.1.

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

3. State diagram :

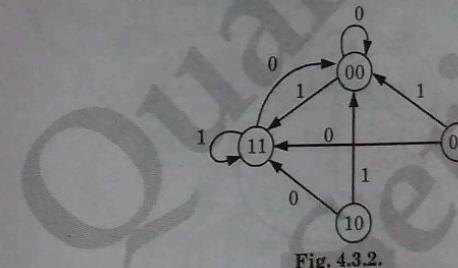
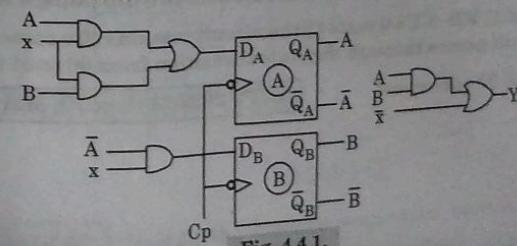


Fig. 4.3.2.

- Que 4.4.** Derive the state table and state diagram for the sequential circuit is shown in Fig. 4.4.1.



**Answer**

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

$$Y = AB + \bar{x} \quad \dots(4.4.1)$$

$$D_A = Ax + Bx \quad \dots(4.4.2)$$

$$D_B = \bar{A}x \quad \dots(4.4.3)$$

2. From eq. (4.4.1), (4.4.2) and (4.4.3) then state table will be

Table 4.4.1.

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

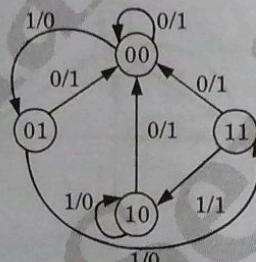


Fig. 4.4.2.

**Que 4.5.** Design a sequential circuit with two flip flops, A & B and one input X. When  $X = 0$  state of the circuit remains the same, when  $X = 1$  circuit passes through the state transition from 00 to 01 to 11 to 10 back to 00 and repeat.

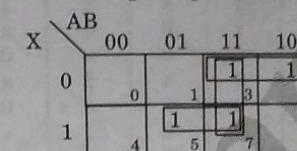
AKTU 2022-23 (Sem-4), Marks 10

**Answer**

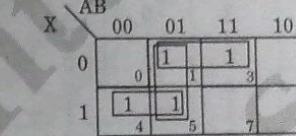
State table :

X	Present state		Next state		Flip flops	
	A	B	A	B	$D_A$	$D_B$
0	0	0	0	0	0	0
0	0	1	0	1	0	1
0	1	0	1	0	1	0
0	1	1	1	1	1	0
1	0	0	0	1	0	1
1	0	1	1	1	1	1
1	1	1	1	0	1	0
1	1	0	0	0	0	0

K-map simplification :



$$D_A = X B + \bar{X} A$$



$$D_B = \bar{X} A + X B$$

Sequential circuit diagram :

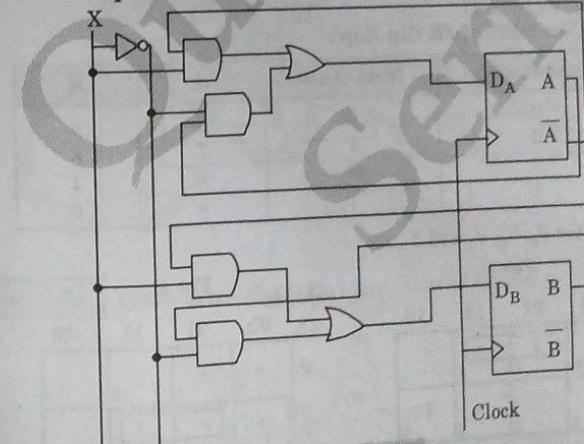


Fig. 4.5.1.

**Que 4.6.** Design the clocked sequential circuit whose state diagram is given in Fig. 4.6.1 using JK flip-flop.

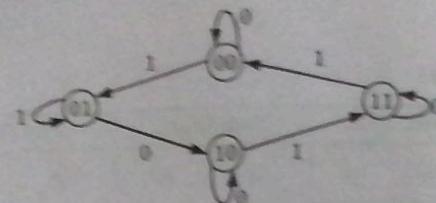


Fig. 4.6.1.

**Answer**

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

Input	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	x	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	$J$	$K$
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

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

For $J_1$				
$Q_1 Q_0$	00	01	11	10
0		1	x	x
1	4	5	x	x

$J_1 = \overline{x} Q_0$

For $K_1$				
$Q_1 Q_0$	00	01	11	10
0	x	x	1	s
1	x	x	1	s

$K_1 = x Q_0$

**Digital Electronics**

For $J_0$				
$Q_1 Q_0$	00	01	11	10
0	0	x	1	s
1	1	x	x	1

$J_0 = x$

For $K_0$				
$Q_1 Q_0$	00	01	11	10
0	x	1	x	s
1	x	1	x	0

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

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

$$J_1 = \overline{x} \bar{Q}_1 + x Q_1 = x \oplus Q_1 = x \odot Q_1$$

$$J_0 = x$$

$$K_1 = x Q_0$$

$$J_0 = \overline{x} Q_0$$

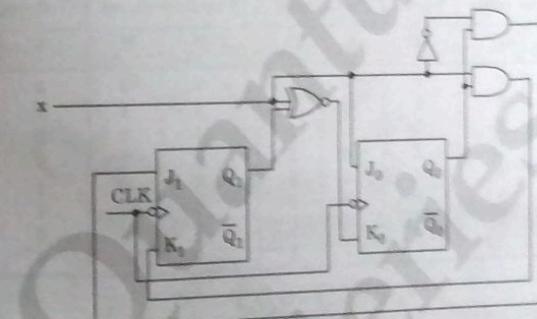


Fig. 4.6.3.

**Que 4.7.** Implement the following state diagram.

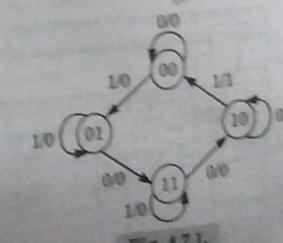


Fig. 4.7.1.

**Answer**

State Table 4.7.1 :

Present state		Next state		Output	
		$x = 0$	$x = 1$	$x = 0$	$x = 1$
$Q_A$	$Q_B$	$Q_A Q_B$	$Q_A Q_B$	$Y$	$Y$
0	0	00	01	0	0
0	1	11	01	0	0
1	0	10	00	1	1
1	1	10	11	0	0

Circuit excitation Table 4.7.2 :

Present state		Input	Next state		Flip flop input		Output
$Q_A$	$Q_B$	$X$	$Q_A$	$Q_B$	$T_A$	$T_B$	$Y$
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	1	1	0	0
0	1	1	0	1	0	0	0
1	0	0	1	0	0	0	1
1	0	1	0	0	1	0	1
1	1	0	1	0	0	1	0
1	1	1	1	1	0	0	0

$$T_A = \overline{Q}_A Q_B \overline{X} + Q_A \overline{Q}_B X$$

$$T_B = \overline{Q}_A \overline{Q}_B X + Q_A Q_B \overline{X}$$

$$\begin{aligned} Y &= Q_A \overline{Q}_B \overline{X} + Q_A \overline{Q}_B X \\ &= \overline{Q}_A \overline{Q}_B (\overline{X} + X) = \overline{Q}_A \overline{Q}_B \end{aligned}$$

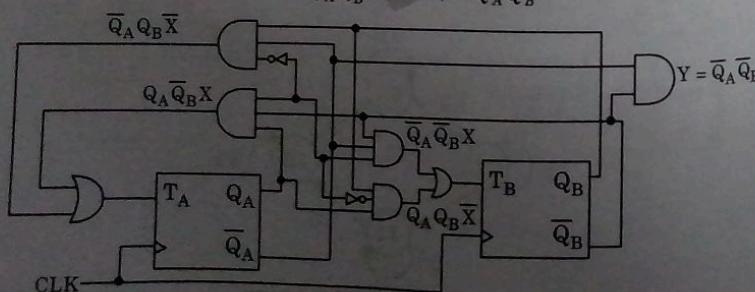


Fig. 4.7.2.

**PART-2**

State Reduction and Assignments, Design Procedure.

Que 4.8. What do you understand by state reduction?

OR

Draw the reduced state table and reduced state diagram for the state table given in Fig. 4.8.1.

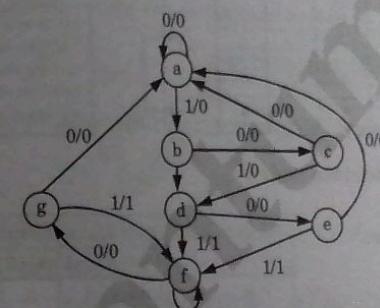


Fig. 4.8.1.

AKTU 2018-19 (Sem-3), Marks 07

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

- 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.
- 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.
- 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.
- When two states are equivalent one of them can be removed without altering input-output relationship.

**Numerical :**

- The given Fig. 4.8.2 has seven states, one input and one output. The given state diagram is converted to state table.

**Answer**

State Table 4.7.1 :

Present state		Next state		Output	
		$x = 0$	$x = 1$	$x = 0$	$x = 1$
$Q_A$	$Q_B$	$Q_A Q_B$	$Q_A Q_B$	$Y$	$Y$
0	0	00	01	0	0
0	1	11	01	0	0
1	0	10	00	1	1
1	1	10	11	0	0

Circuit excitation Table 4.7.2 :

Present state		Input	Next state		Flip flop input		Output
$Q_A$	$Q_B$	$X$	$Q_A$	$Q_B$	$T_A$	$T_B$	$Y$
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	1	1	0	0
0	1	1	0	1	0	0	0
1	0	0	1	0	0	0	1
1	0	1	0	0	1	0	1
1	1	0	1	0	0	1	0
1	1	1	1	1	0	0	0

$$T_A = \overline{Q}_A Q_B \bar{X} + Q_A \overline{Q}_B X$$

$$T_B = \overline{Q}_A \overline{Q}_B X + Q_A Q_B \bar{X}$$

$$Y = Q_A \overline{Q}_B \bar{X} + Q_A \overline{Q}_B X \\ = \overline{Q}_A \overline{Q}_B (\bar{X} + X) = \overline{Q}_A \overline{Q}_B$$

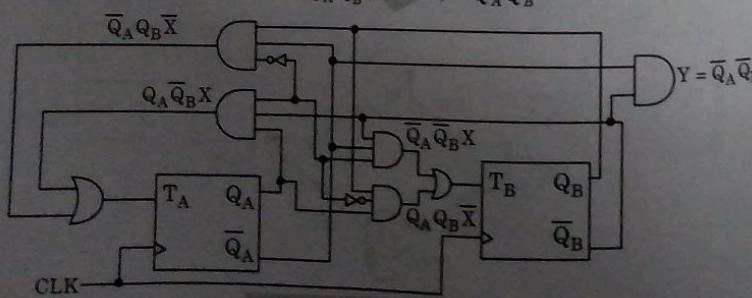


Fig. 4.7.2.

**PART-2**

State Reduction and Assignments, Design Procedure.

Que 4.8. | What do you understand by state reduction ?

OR

Draw the reduced state table and reduced state diagram for the state table given in Fig. 4.8.1.

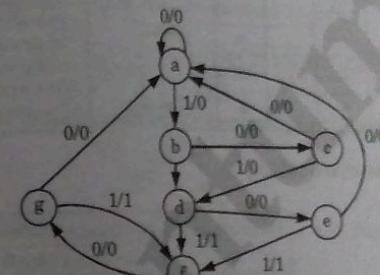


Fig. 4.8.1.

AKTU 2018-19 (Sem-3), Marks 07

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

- 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.
- 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.
- 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.
- When two states are equivalent one of them can be removed without altering input-output relationship.

**Numerical :**

- The given Fig. 4.8.2 has seven states, one input and one output. The given state diagram is converted to state table.

**Answer**

State Table 4.7.1 :

Present state		Next state		Output	
		$x = 0$	$x = 1$	$x = 0$	$x = 1$
$Q_A$	$Q_B$	$Q_A Q_B$	$Q_A Q_B$	$Y$	$Y$
0	0	00	01	0	0
0	1	11	01	0	0
1	0	10	00	1	1
1	1	10	11	0	0

Circuit excitation Table 4.7.2 :

Present state		Input	Next state		Flip flop input		Output
$Q_A$	$Q_B$	$X$	$Q_A$	$Q_B$	$T_A$	$T_B$	$Y$
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	1	1	0	0
0	1	1	0	1	0	0	0
1	0	0	1	0	0	0	1
1	0	1	0	0	1	0	1
1	1	0	1	0	0	1	0
1	1	1	1	1	0	0	0

$$T_A = \overline{Q}_A Q_B \overline{X} + Q_A \overline{Q}_B X$$

$$T_B = \overline{Q}_A \overline{Q}_B X + Q_A Q_B \overline{X}$$

$$\begin{aligned} Y &= Q_A \overline{Q}_B \overline{X} + Q_A \overline{Q}_B X \\ &= \overline{Q}_A \overline{Q}_B (\overline{X} + X) = \overline{Q}_A \overline{Q}_B \end{aligned}$$

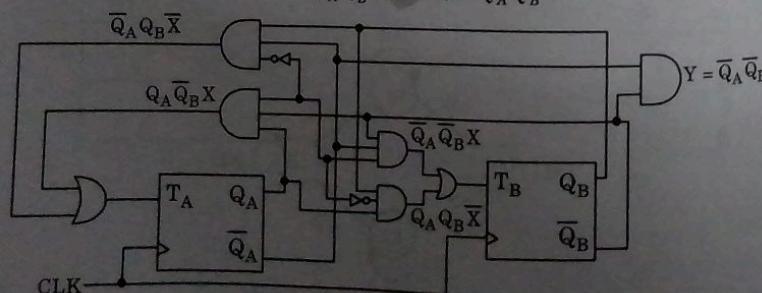


Fig. 4.7.2.

**PART-2**

*State Reduction and Assignments, Design Procedure.*

Que 4.8. | What do you understand by state reduction ?

OR

Draw the reduced state table and reduced state diagram for the state table given in Fig. 4.8.1.

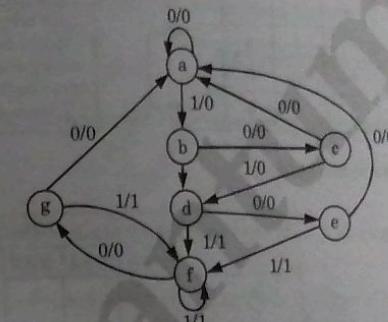


Fig. 4.8.1.

AKTU 2018-19 (Sem-3), Marks 07

**Answer**

**State reduction :**

- 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.
- 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.
- 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.
- When two states are equivalent one of them can be removed without altering input-output relationship.

**Numerical :**

- The given Fig. 4.8.2 has seven states, one input and one output. The given state diagram is converted to state table.

**Answer**

State Table 4.7.1 :

Present state		Next state		Output	
		$x = 0$	$x = 1$	$x = 0$	$x = 1$
$Q_A$	$Q_B$	$Q_A Q_B$	$Q_A Q_B$	$Y$	$Y$
0	0	00	01	0	0
0	1	11	01	0	0
1	0	10	00	1	1
1	1	10	11	0	0

Circuit excitation Table 4.7.2 :

Present state		Input	Next state		Flip flop input		Output
$Q_A$	$Q_B$	$X$	$Q_A$	$Q_B$	$T_A$	$T_B$	$Y$
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	1	1	0	0
0	1	1	0	1	0	0	0
1	0	0	1	0	0	0	1
1	0	1	0	0	1	0	1
1	1	0	1	0	0	1	0
1	1	1	1	1	0	0	0

$$T_A = \bar{Q}_A Q_B \bar{X} + Q_A \bar{Q}_B X$$

$$T_B = \bar{Q}_A \bar{Q}_B X + Q_A Q_B \bar{X}$$

$$\begin{aligned} Y &= Q_A \bar{Q}_B \bar{X} + Q_A Q_B \bar{X} \\ &= \bar{Q}_A \bar{Q}_B (\bar{X} + X) = \bar{Q}_A \bar{Q}_B \end{aligned}$$

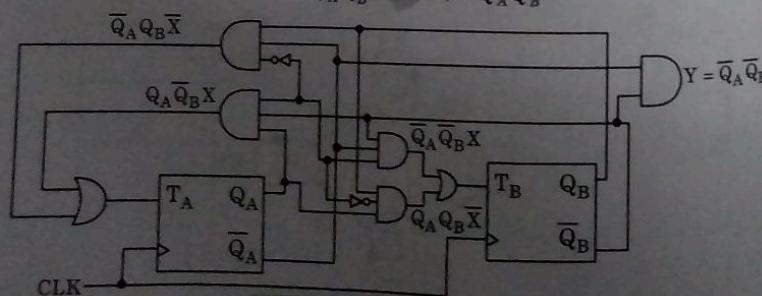


Fig. 4.7.2.

**PART-2**

State Reduction and Assignments, Design Procedure.

Que 4.8. | What do you understand by state reduction ?

OR

Draw the reduced state table and reduced state diagram for the state table given in Fig. 4.8.1.

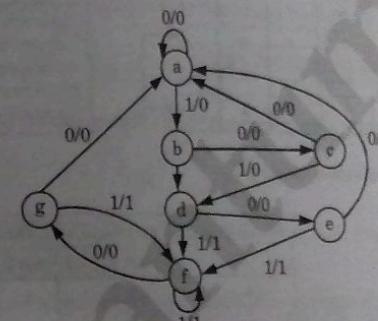


Fig. 4.8.1.

AKTU 2018-19 (Sem-3), Marks 07

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

- 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.
- 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.
- 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.
- When two states are equivalent one of them can be removed without altering input-output relationship.

**Numerical :**

- The given Fig. 4.8.2 has seven states, one input and one output. The given state diagram is converted to state table.

**Answer**

State Table 4.7.1:

Present state		Next state		Output	
		$x = 0$	$x = 1$	$x = 0$	$x = 1$
$Q_A$	$Q_B$	$Q_A Q_B$	$Q_A Q_B$	$Y$	$Y$
0	0	00	01	0	0
0	1	11	01	0	0
1	0	10	00	1	1
1	1	10	11	0	0

Circuit excitation Table 4.7.2:

Present state		Input	Next state		Flip flop input		Output
$Q_A$	$Q_B$	$X$	$Q_A$	$Q_B$	$T_A$	$T_B$	$Y$
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	1	1	0	0
0	1	1	0	1	0	0	0
1	0	0	1	0	0	0	1
1	0	1	0	0	1	0	1
1	1	0	1	0	0	1	0
1	1	1	1	1	0	0	0

$$T_A = \bar{Q}_A Q_B \bar{X} + Q_A \bar{Q}_B X$$

$$T_B = \bar{Q}_A \bar{Q}_B X + Q_A Q_B \bar{X}$$

$$\begin{aligned} Y &= Q_A \bar{Q}_B \bar{X} + Q_A \bar{Q}_B X \\ &= \bar{Q}_A \bar{Q}_B (\bar{X} + X) = \bar{Q}_A \bar{Q}_B \end{aligned}$$

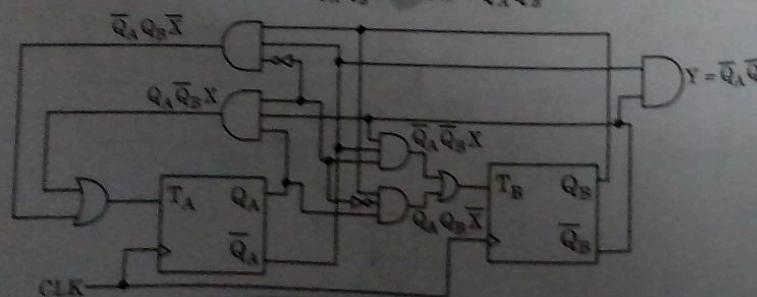


Fig. 4.7.2.

**PART-2***State Reduction and Assignments, Design Procedure.*

Que 4.8. | What do you understand by state reduction?

OR

Draw the reduced state table and reduced state diagram for the state table given in Fig. 4.8.1.

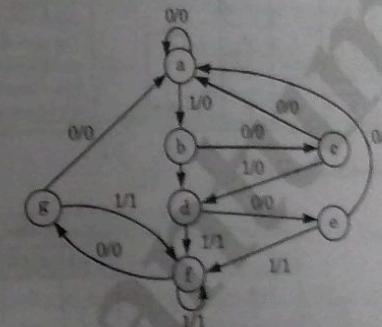


Fig. 4.8.1.

AKTU 2018-19 (Sem-3), Marks 07

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

- 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.
- 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.
- 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.
- When two states are equivalent one of them can be removed without altering input-output relationship.

**Numerical :**

- The given Fig. 4.8.2 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. 4.8.2.

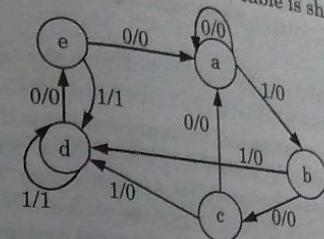


Fig. 4.8.2. State diagram.

- Que 4.9.** Explain state reduction and assignment with suitable example.

AKTU 2022-23 (Sem-4), Marks 10  
OR

Describe state assignment rules for assigning states for sequential circuits.

#### Answer

**State reduction with example :** Refer Q. 4.8, Page 4-11G, Unit-4.

#### State assignment :

- In sequential circuits we know that the behaviour of the circuit is defined in terms of its inputs, present state, next state and outputs.
- To generate the desired next state at particular present state and inputs, it is necessary to have specific flip-flop inputs. These flip-flop inputs are described by a set of boolean functions called flip-flop input functions.
- To determine the flip-flop input function, it is necessary to represent states in the state diagram using binary values instead of alphabets. This procedure is known as state assignment.

**State assignment rules :** The following rules are used in state assignment.

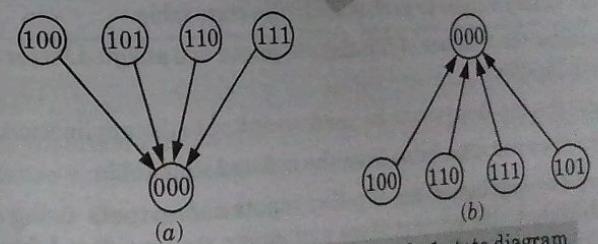


Fig. 4.9.1. (a) State assignment rule 1 state diagram.  
(b) State assignment rule 2 state diagram.

**Rule 1 :** States having the same next states for a given input condition should have assignments which can be grouped into logically adjacent cells in  $K$ -map.

**Rule 2 :** States having different next states should have assignment which can be grouped into logically adjacent cells in  $K$ -map.

**State assignment example :** State assignment is the process of giving unique binary codes to states in a finite state machine (FSM). For example, if you have a 3-state FSM, you can assign "00" to State A, "01" to State B, and "10" to State C, ensuring each state has a unique code. This simplifies circuit design and helps avoid ambiguity.

**Que 4.10.** Write the design procedure for clocked sequential circuits and implement the following state diagram.

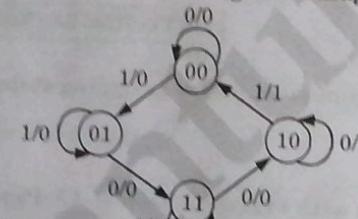


Fig. 4.10.1.

AKTU 2021-22 (Sem-4), Marks 10

**Answer**

**Design procedure :** The following steps are followed to design the clocked sequential logic circuit :

1. Obtain the state table from the given circuit information such as a state diagram, a timing diagram or description.
2. The number of states may be reduced by state reduction technique.
3. Assign binary values to each state in the state table.
4. Determine the number of flip-flops required and assign a letter symbol to each flip-flop.
5. Choose the flip-flop type to be used according to the application.
6. Derive the excitation table from the reduced state table.
7. Derive the expression for flip-flop inputs and outputs using  $K$ -map simplification (the present state and inputs are considered for  $K$ -map simplification) and draw logic circuit using flip-flops and gates.

**Numerical :** Refer Q. 4.7, Page 4-4G, Unit-4.

**PART-3***Analysis Procedure of Asynchronous Sequential Circuits.*

**Que 4.11.** Illustrate the working and applications of asynchronous sequential circuits.

AKTU 2022-23 (Sem-3), Marks 10

**Answer****A. Asynchronous sequential circuit :**

1. The asynchronous sequential logic circuit model is same as the synchronous sequential logic circuit. The only difference between two sequential logic circuits is that the memory element is replaced by a delay element in asynchronous sequential circuit.

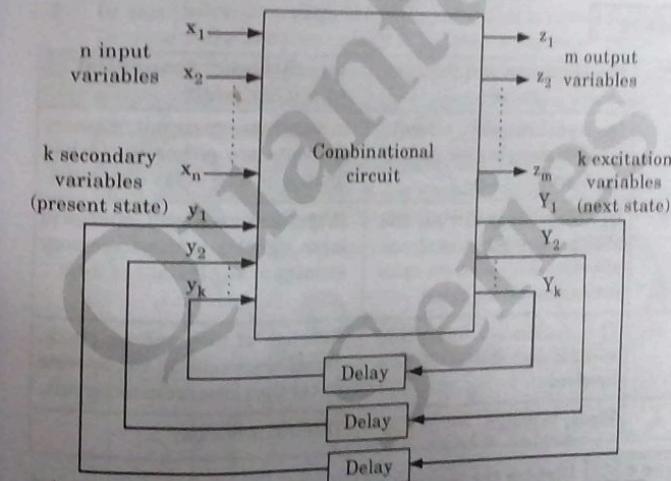


Fig. 4.11.1. Block diagram of an asynchronous sequential circuit.

2. In asynchronous sequential logic circuit, operation is not synchronized because we are not using any synchronous clock.
3. The block diagram of an asynchronous sequential circuit consists of a combinational circuit and delay elements connected to form feedback loops.
4. There are  $n$  input variables,  $m$  output variables and  $k$  internal states. The delay element provides a short-term memory for the sequential circuit.

**4-16 G (OEC-Sem-3 & 4) Synchronous & Asynchronous Sequential Circuits**

5. The present state and next state variables are called as secondary variable and excitation variable respectively.
  6. When an input variable changes in value  $y$ , the secondary variable does not change instantaneously. It takes a certain amount of time for the signal to propagate from the input terminals through the combinational circuit to the  $y$  excitation variables, which generate new values for the next state.
  7. These values propagate through the delay element and become the new present state for the secondary variables.
- B. Applications :** Asynchronous circuits are used in low-power and high-speed operations such as simple microprocessor, digital signal processing units and in timed circuits and delay elements.

**Que 4.12.** Differentiate between synchronous and asynchronous sequential circuit.

**Answer**

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.	Difficult to design.

**Que 4.13.** Discuss the concept of transition table, flow table and primitive flow table to design fundamental mode asynchronous sequential circuit.

**Answer**

**Transition table :**

1. In the transition table specific state variable values are assigned to each state. Assignment of values to state variables is called state assignment.
2. Like state table, transition table also represents relationship between input, output and flip-flop states. The Table 4.13.1 shows the transition table.

**Digital Electronics**

**4-17 G (OEC-Sem-3 & 4)**

Table. 4.13.1.

Present state		Next state		Output	
		X = 0	X = 1	X = 0	X = 1
A	B	AB	AB	Y	Y
0	0	00	10	0	1
0	1	11	00	0	0
1	0	10	01	1	0
1	1	00	10	1	0

4. Here,  $AB$  is the state variable. The  $AB = 00$  represents one state,  $AB = 01$  represents second state and so on.

**Flow table :**

1. The flow table in the asynchronous sequential circuit is same as that of state table in the synchronous sequential circuit.
2. In asynchronous sequential circuits state table is known as flow table because of the behaviour of the asynchronous sequential circuit.
2. The state change occur independent of a clock, based on the logic propagation delay and causes the states to flow from one to another.

**Primitive flow table :**

1. A primitive flow table is a special case of flow table. It is defined as a flow table which has exactly one stable state for each row in the table. The design process begins with the construction of primitive flow table.
2. The four states are designated by the letters  $a$ ,  $b$ ,  $c$  and  $d$ .

x	0	1
a	(a)	b
b	c	(b)
c	(c)	d
d	a	(d)

(a) Four states with one input

$x_1x_2$	00	01	11	10
a	(a), 0	(a), 0	(a), 0	b, 0
b	a, 0	a, 0	b, 1	(b), 0

(b) Two states with two inputs and 1 output

Fig. 4.13.1.

3. The table shown in Fig. 4.13.1(a) is called primitive flow table because it has only one state in each row and the table shown in Fig. 4.13.1(b) shows a flow table with more than one stable state in same row.
4. It has two states  $a$  and  $b$ , two inputs  $x_1$  and  $x_2$  with one output  $z$ .
5. If  $x_1 = 0$ , the circuit is in state  $a$  and if  $x_1$  goes to 1 while  $x_2$  is 0, the circuit goes to state  $b$ .

6. In order to obtain the circuit described by a flow table, it is necessary to assign to each state a distinct binary value. We assign binary 0 to state  $a$  and binary 1 to state  $b$ .

	$x_1x_2$	00	01	11	10
y	0	① <sub>0</sub>	① <sub>1</sub>	① <sub>3</sub>	1 <sub>2</sub>
	1	0 <sub>4</sub>	0 <sub>5</sub>	① <sub>7</sub>	① <sub>6</sub>

Transition table  $Y = x_1x_2' + x_1y$

	$x_1x_2$	00	01	11	10
y	0	0	0	0	0
	1	0 <sub>4</sub>	0 <sub>5</sub>	1 <sub>7</sub>	0 <sub>6</sub>

Map for output  $z = x_1x_2y$

Fig. 4.13.2.

7. Now the logic diagram is as follows :

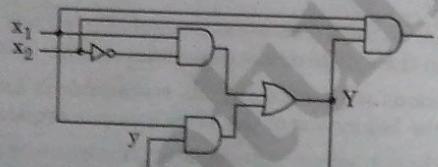


Fig. 4.13.3.

Que 4.14. An asynchronous sequential circuit with two excitation function with two feedback loop is given as :

$$Y_1 = xy_1 + \bar{x}y_2$$

$$Y_2 = x\bar{y}_1 + \bar{x}\bar{y}_2$$

- i. Draw the logic diagram of the circuit.  
ii. Derive the transition table and obtain the flow table.

AKTU 2021-22 (Sem-4), Marks 10

#### Answer

- i. Logic diagram :

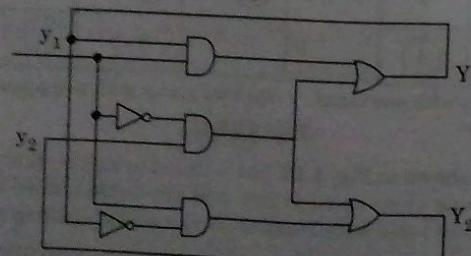


Fig. 4.14.1.

- i. Logic diagram : The logic diagram for  $Y_1$  and  $Y_2$  is shown in Fig. 4.14.2.

x	0	1
y <sub>1</sub> y <sub>2</sub>	00	00
	01	10
	11	11
	10	01

(a)  $Y_1 = xy_1 + \bar{x}y_2$

x	0	1
y <sub>1</sub> y <sub>2</sub>	00	00
	01	11
	11	10
	10	00

(b)  $Y_2 = x\bar{y}_1 + \bar{x}\bar{y}_2$

Fig. 4.14.2.

- ii. Transition table :

- a. The transition table is shown in Fig. 4.14.3, is obtained from the maps by combining the binary values in corresponding squares.
- b. The transition table shows the values of  $Y = Y_1Y_2$  inside each square.
- c. The first bit of  $Y$  is obtained from the value of  $Y_1$ , and the second bit is obtained from the value of  $Y_2$  in the same square position.
- d. For a state to be stable, the secondary variable must match the excitation variables i.e., the value of  $Y$  must be the same as that of  $y = y_1y_2$ .
- e. Those entries in the transition table where  $Y = y$  are circled to indicate a stable condition. An encircled entry represents an unstable state.

x	0	1
y <sub>1</sub> y <sub>2</sub>	00	01
	01	11
	11	10
	10	00

Fig. 4.14.3. Transition table.

- iii. Flow table :

- a. A flow table is similar to a transition table except that the internal states are symbolized with letters rather than binary numbers.
- b. The flow table also includes the output values of the circuit for each stable state.
- c. Flow table is shown in Fig. 4.14.4. The four states represent by the letters  $a$ ,  $b$ ,  $c$ , and  $d$ , it reduces to the transition table of Fig. 4.14.4 if we assign the following binary values to the states :  $a = 00$ ,  $b = 01$ ,  $c = 11$ , and  $d = 10$ .

4-20 G (OEC-Sem-3 & 4) Synchronous & Asynchronous Sequential Circuits

- d. The table of Fig. 4.14.4 is called a primitive flow table because it has only stable state in each row.

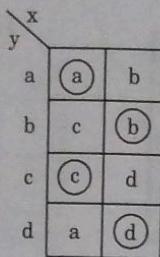


Fig. 4.14.4. Four state with one input.

**Que 4.15.** An asynchronous sequential logic circuit is described by the following excitation and output function

$$y = X_1 X_2 + (X_1 + X_2)Y$$

$$Z = y$$

Draw the logic diagram of the circuit. Also derive the transition table and output map.

AKTU 2017-18 (Sem-3), Marks 07

OR

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.

AKTU 2018-19 (Sem-3), Marks 07

**Answer**

State table :

Present state			Next state			Stable state	Output Z
$X_1$	$X_2$	$Y$	$X_1$	$X_2$	$Y$		
0	0	0	0	0	0	Yes	0
0	0	1	0	0	0	No	0
0	1	0	0	1	0	Yes	0
0	1	1	0	1	1	Yes	1
1	0	0	1	0	0	Yes	0
1	0	1	1	0	1	Yes	1
1	1	0	1	1	1	No	1
1	1	1	1	1	1	Yes	1

Digital Electronics

4-21 G (OEC-Sem-3 & 4)

- i. Logic diagram :

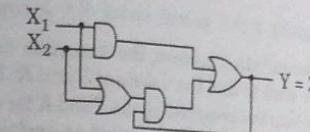


Fig. 4.15.1.

- ii. Transition table :

		00	01	11	10	Unstable state
		0	0	1	0	Stable state
Y	0	0	0	-	0	
	1	-	1	1	1	

Fig. 4.15.2.

Output map :

		00	01	11	10
		0	0	-	0
Y	0	0	0	-	0
	1	-	1	1	1

Fig. 4.15.3.

- iii. The circuit gives carry output of the full adder circuit.

**PART-4**

Circuit with Latches, Design Procedure.

**Que 4.16.** Derive a latch circuit from a transition table.

OR

Implement the circuit defined by the following transition table with a NOR SR Latch. Also show the implementation with NAND SR latch.

		00	01	11	10
		0	0	0	1
y	0	0	0	0	1
	1	0	0	1	1

Fig. 4.16.1.

AKTU 2021-22 (Sem-4), Marks 10

**Answer**

- During the implementation process, the transition table of the circuit is available and we wish to find the values of  $S$  and  $R$ . For this reason, we need a table that lists the required inputs  $S$  and  $R$  for each of the possible transitions from  $y$  to  $Y$ . Such a list is called an excitation table.
- The excitation table of the SR latch is shown in Fig. 4.16.2(b). The first two columns list the four possible transitions from  $y$  to  $Y$ . The next two columns specify the required input values that will result in the specified transition.
- For example, in order to provide a transition from  $y = 0$  to  $y = 1$ , it is necessary to ensure that input  $R = 0$ . This is shown in the second row of the transition table.
- The required input conditions for each of the latch transition table of Fig. 4.16.2(d) after removing the unstable condition  $SR = 11$ .
- For example, the transition table shows that in order to change from  $y = 0$  to  $Y = 0$ ,  $SR$  can be either 00 or 01. This means that  $S$  must be 1 and  $R$  may be either 0 or 1.
- Therefore, the first row in the excitation table shows  $S = 0$  and  $R = X$ , where  $X$  is a don't-care condition signifying either a 0 or a 1.

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

(a) Transition table

$$y = x_1 x'_2 + x_1 y$$

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

(b) Excitation table

$$y = x_1 x'_2 + x_1 y$$

$y$	00	01	11	10
0	0	0	0	1
1	0	0	x	x

$$(c) Map for S = x_1 x'_2$$

$y$	00	01	11	10
0	x	x	x	1
1	1	1	0	0

(d) Map for R = x'\_1

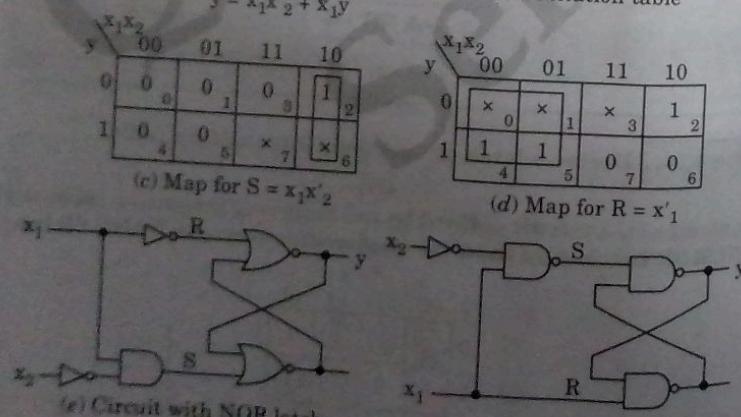


Fig. 4.16.1. Derivation of a latch circuit from a transition table.

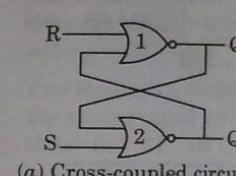
**Que 4.17.** Design a SR latch using NOR gates as well as NAND gates.

**Answer**

**SR latch :**

- The output  $Q$  is equivalent to the excitation variable  $Y$  and the secondary variable  $y$ . The boolean function for the output is  

$$Y = [(S + y)' + R]' = (S + y)R' = SR' + R'y$$
- Plotting  $Y$  as in Fig. 4.17.1, we obtain the transition table for the circuit.



(a) Cross-coupled circuit

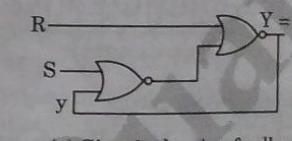
$S$	$R$	$Q$	$Q'$
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1
1	1	0	0

(After  $S = 1, R = 0$ )

$S$	$R$	$Q$	$Q'$
0	0	1	0
0	1	0	1
0	0	0	1
1	1	0	0

(After  $S = 0, R = 0$ )

(Forbidden)



(c) Circuit showing feedback

$SR$	$y$	00	01	11	10
0	0	0	0	0	1
1	1	1	0	1	1

(d) Transition table

$$Y = SR' + R'y$$

$$Y = S + R'y \text{ when } SR = 0$$

Fig. 4.17.1. SR latch with NOR gates.

- We can now investigate the behavior of the SR latch from the transition table. The state with  $SR = 10$  is a stable state because  $Y = y = 1$ ; likewise, the state with  $SR = 01$  is a stable state, because  $Y = y = 0$ . With  $SR = 10$ , the output  $Q = y = 1$  and the latch is said to be set. Changing  $S$  to 0 leaves the circuit in the set state. With  $SR = 01$ , the output  $Q = y = 0$  and the latch is said to be reset. A change of  $R$  back to 0 leaves the circuit in the reset state. These conditions are also listed in the truth table.
- When we OR the boolean expression  $SR'$  with  $SR$ , the result is the single variable  $S$ .

$$SR' + SR = S(R' + R) = S$$

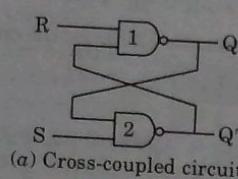
From this, we infer that  $SR' = S$  when  $SR = 0$ .

- However, if it is found that both  $S$  and  $R$  can be equal to 1 at the same time then it is necessary to use the original excitation function.

**4-24 G (OEC-Sem-3 & 4)** Synchronous & Asynchronous Sequential Circuits

6. The analysis of the SR latch with NAND gates is carried out in Fig. 4.17.2. The NAND latch operates with both inputs normally at 1, unless the state of the latch has to be changed. The condition to be avoided here is that both S and R not be 0 simultaneously. This condition is satisfied when  $S'R' = 0$ . The excitation function for the circuit in Fig. 4.17.2 is

$$Y [S(Ry)']' = S' + Ry$$

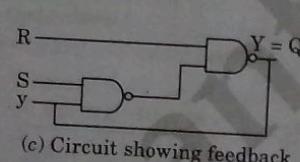


(a) Cross-coupled circuit

S	R	Q	Q'
1	0	0	1
0	0	0	1
0	1	1	0
0	0	1	0
1	1	1	1

(After SR = 10)  
(After SR = 01)

(b) Truth table



(c) Circuit showing feedback

SR	y			
	00	01	11	10
0	1	1	0	0
1	0	1	1	0

(d) Transition table

Fig. 4.17.2. SR latch with NAND gates.

7. Comparing this with the excitation function of the NOR latch, we note the S has been replaced with  $S'$  and  $R'$  with R.  
8. Hence, the input variables for the NAND latch require the complemented values of those used in the NOR latch. For this reason, the NAND latch is sometimes referred to as an  $S'R'$  latch (or  $\bar{S}\bar{R}$ ) latch.

**Que 4.18.** Explain the design procedure for asynchronous sequential circuit.

**Answer**

The steps involved in designing an asynchronous sequential circuit are:

1. Construct a primitive flow table from the problem statement. An intermediate step may include the development of a state diagram.
2. Primitive flow table is reduced by eliminating redundant states by using state reduction techniques.
3. State assignment is made.
4. The primitive flow table is realized using appropriate logic elements.

Digital Electronics

**4-25 G (OEC-Sem-3 & 4)**

**PART-5**

Reduction of State and Flow Table.

**Que 4.19.** Illustrate the state reduction technique for digital circuits.

AKTU 2022-23 (Sem-3), Marks 10

**Answer**

1. The state-reduction procedure for completely specified state tables is based on an algorithm that combines two states in state table into one, as long as they can be shown to be equivalent.
2. Two states are equivalent if, for each possible input, they give exactly the same output and go to the same next states or to equivalent next states.
3. The present states  $a$  and  $b$  have the same output for the same input. Their next states are  $c$  and  $d$  for  $x = 0$ , and  $b$  and  $a$  for  $x = 1$ . If we can show that the pair of states  $(c, d)$  are equivalent, then the pair of states  $(a, b)$  will also be equivalent, because they will have the same or equivalent next states.
4. When this relationship exists, we say that  $(a, b)$  imply  $(c, d)$  in the sense that if  $a$  and  $b$  are equivalent then  $c$  and  $d$  have to be equivalent. Similarly, the pair of states  $(c, d)$  implies the pair of states  $(a, b)$ , i.e.,  $a$  and  $b$  are equivalent, and so are  $c$  and  $d$ .

Table : 4.19.1. State table to demonstrate equivalent states

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

Table : 4.19.2. State table to be reduced

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

6. Two states having different outputs for the same input are not equivalent.
7. Two states that are not equivalent are marked with a cross (x) in the corresponding square, whereas their equivalence is recorded with a check mark (✓).

a	d, e✓				
c	x	x			
d	x	x	x		
e	x	x	x	x	
f	c, d x	c, e x a, b	x	x	x
g	x	x	x	d, e✓	d, e✓
a					x
b					
c					
d					
e					
f					

Fig. 4.19.1. Implication table.

8. The step-by-step procedure of filling in the squares is as follows : First, we place a cross in any square corresponding to a pair of states whose outputs are not equal for every input.
9. In this case, state c has a different output than any other state, so a cross is placed in the two squares of row c and the four squares of column c. There are nine other squares in this category in the implications table.
10. From the state table, we see that pair (a, b) implies (d, e), so (d, e) is recorded in the square defined by column a and row b. We proceed in this manner until the entire table is completed.
11. The next step is to make successive passes through the table to determine whether any additional squares should be marked with a cross.

12. A square in the table is crossed out if it contains at least one implied pair that is not equivalent.
13. For example, the square defined by a and f is marked with a cross next to (c, d) because the pair (c, d) defines a square that contains a cross. This procedure is repeated until no additional squares can be crossed out. In this example, the equivalent states are (a, b) (d, e) (d, g) (e, g)
14. We now combine pairs of states into larger groups of equivalent states. The last three pairs can be combined into a set of three equivalent states (d, e, g) because each one of the states in the group is equivalent to the other two. This group consists of (a, b) (c) (d, e, g) (f)
15. Thus, table can be reduced from seven states to four, one for each member of the preceding partition. The reduced state table is obtained by replacing state b by a and states e and g by d and is shown in Table 4.19.3.

Table 4.19.3 : Reduced state table

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

PART-6

Race Free State Assignment, Hazards.

Que 4.20. | What is the significance of state assignment ? List the different technique used for state assignment.

Answer

State assignment :

1. In synchronous circuit, the state assignments are made with the objective of circuit reduction. In asynchronous circuits, the objective of state assignment is to avoid critical races.
2. The following methods can be used to avoid critical race condition :
  - i. Shared row state assignment :
  - ii. Shared column state assignment :
  - iii. Shared row and column state assignment :
3. The state variables are assigned with binary numbers in such a way that only one state variable can change at one time when a state transition occurs.

#### 4-28 G (OEC-Sem-3 & 4) Synchronous & Asynchronous Sequential Circuits

For example, 101 and 111 are adjacent because they differ only in the second bit.

2. The state A is assigned binary value 00 and state C is assigned binary value 11.
3. The transition from A to C will go through D, which satisfy the condition that only one binary variable changes during each state transition avoiding the critical race.

##### ii. One hot state assignment :

1. In this type of state assignment, only one variable is active or hot for each row in the original flow table. Here the number of bits used for state variable is exactly equal to the number of states used.
2. Let us consider the Table 4.20.1. In the above task, between any two states there are two variable changes, this should be converted into single variable change by shared row state assignment.

Table 4.20.1.

State variable				State	Input $X_1 X_2$			
$S_4$	$S_3$	$S_2$	$S_1$		00	01	11	10
0	0	0	1	A	(A)	B	C	D
0	0	1	0	B	A	(B)	C	D
0	1	0	0	C	A	B	(C)	(C)
1	0	0	0	D	(D)	B	C	D

**Que 4.21.** What are critical race and non-critical race ? How can they be avoided ?

AKTU 2018-19 (Sem-3), Marks 3.5

##### Answer

###### Race condition :

1. A race 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.

$x_1 x_2$	00	01	11	10	$x_1 x_2$	00	01	11	10
y	0	0	0	1	y	0	0	0	1
0	0	0	0	1	3	1	3	2	2
1	4	5	7	6	5	4	7	6	0

Transition table  $Y = x_1 x_2' + x_1 y$

Map for output  $z = x_1 x_2 y$

Fig. 4.21.1.

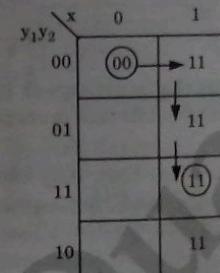
##### Digital Electronics

#### 4-29 G (OEC-Sem-3 & 4)

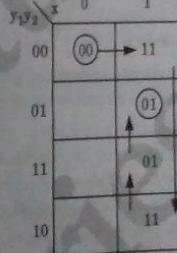
2. When unequal delays are encountered, a race condition may cause the state variables to change in an unpredictable manner.
3. For example, if state variable changes from 00 to 11, the difference in delays may cause the first variable to change faster than the second, therefore the state variable change in sequence from 00 to 10 and then to 11. If second variable changes faster than the first then state variable changes from 00 to 01 and then to 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 :



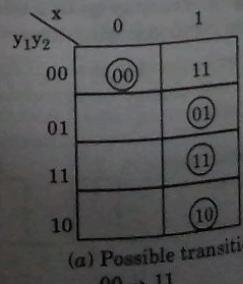
(a) Possible transitions  
 $00 \rightarrow 11$   
 $00 \rightarrow 01 \rightarrow 11$   
 $00 \rightarrow 10 \rightarrow 11$



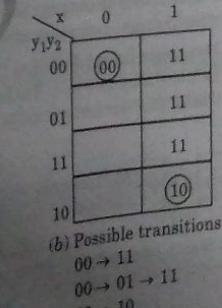
(b) Possible transitions  
 $00 \rightarrow 11 \rightarrow 01$   
 $00 \rightarrow 01$   
 $00 \rightarrow 10 \rightarrow 11 \rightarrow 01$

Fig. 4.21.2.

###### Critical race :



(a) Possible transitions  
 $00 \rightarrow 11$   
 $00 \rightarrow 01$   
 $00 \rightarrow 10$



(b) Possible transitions  
 $00 \rightarrow 11$   
 $00 \rightarrow 01 \rightarrow 11$   
 $00 \rightarrow 10$

Fig. 4.21.3.

**Methods of avoidance :**

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.

**Que 4.22.** Explain the term Hazard. Define different types of Hazards along with detection and reduction of Hazards.

AKTU 2022-23 (Sem-3), Marks 10

**Answer****Hazards :**

1. Hazards are unwanted switching transient that may appear at the output of a circuit because different paths exhibit different propagation delays.
2. 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.

**Various types of hazards in digital circuits are :**

- 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.

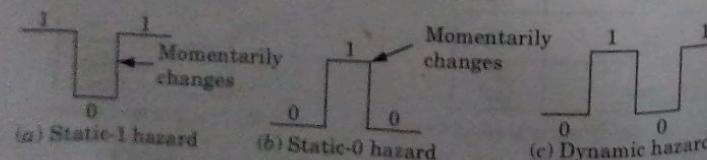


Fig. 4.22.1.

- iv. **Essential hazard :** 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(s) of a logic circuit as shown in Fig. 4.22.2.
3. We find that there are two output paths that contain combinational logic gates and sequential logic circuit.
4. It may so happen that certain paths may produce more delay than the other.

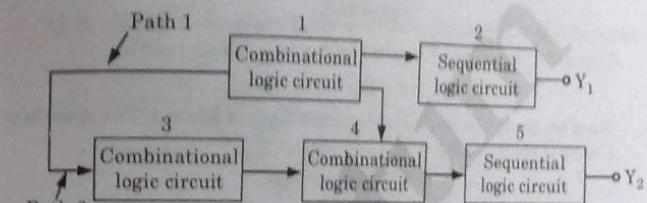


Fig. 4.22.2. Schematic circuit to show essential hazard.

**Hazard detection :** Following are some common approaches used to detect hazards :

1. **Simulation and timing analysis :** By simulating the circuit's behavior and analyzing its timing characteristics, hazards can be identified. The simulation results can highlight any unexpected glitches or momentary changes in the output signals.
2. **Static timing analysis :** It analyzes the circuit's timing paths, gate delays, and arrival times of signals to identify potential hazards. It can detect timing violations that may lead to hazards.

**Reduction of hazards :** 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.

**Que 4.23.** Describe the hazards in digital circuits. How are these removed? Design a hazards free circuit of the following Boolean function :

$$F(A, B, C) = \Sigma m(1, 2, 3, 5)$$

AKTU 2018-19 (Sem-3), Marks 3.5

**Answer**

**Hazards and its reduction :** Refer Q. 4.22, Page 4-30G, Unit-4.

**Numerical :**

1. The K-map simplification and logic diagram for Boolean function  $F$  is shown in Fig. 4.23.1(a) and Fig. 4.23.1(b).

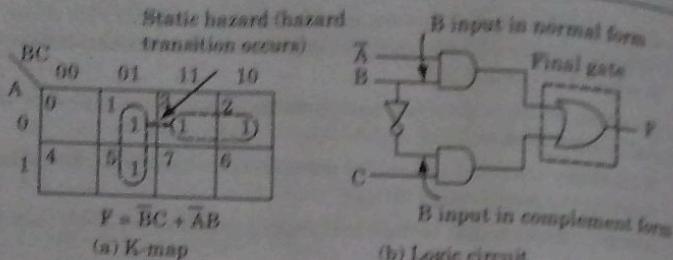


Fig. 4.23.1.

2. Expression for output  $F$  is given by

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

3. Then we group the two 1's to eliminate static-1 hazard. The modified K-map is shown in Fig. 4.23.2.

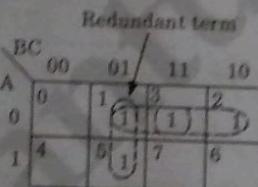


Fig. 4.23.2. K-map with redundant term.

4. The new expression for output will be

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

5. So, the hazard free circuit is shown in Fig. 4.23.3. Note that static-1 hazard has been eliminated due to the additional AND gate.

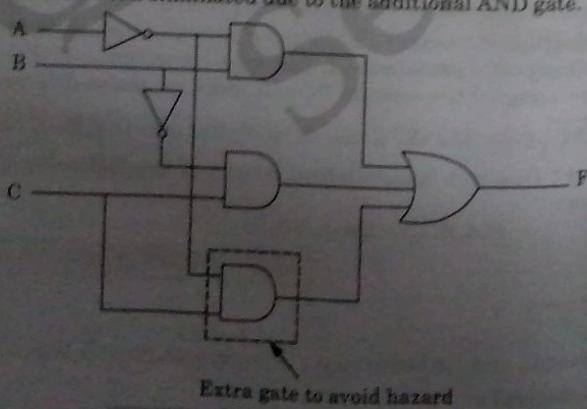


Fig. 4.23.3. Hazard free logic circuit.

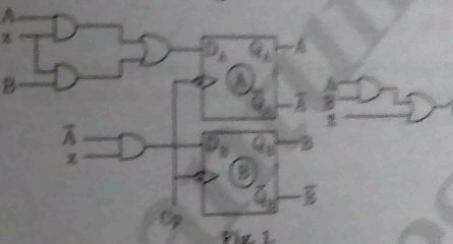
**VERY IMPORTANT QUESTIONS**

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

- Q. 1. Discuss the analysis of clocked sequential circuits.

Ans: Refer Q. 4.1, Unit-4.

- Q. 2. Derive the state table and state diagram for the sequential circuit is shown in Fig. 4.4.1.



Ans: Refer Q. 4.4, Unit-4.

- Q. 3. Explain state reduction and assignment with suitable example.

Ans: Refer Q. 4.9, Unit-4.

- Q. 4. Differentiate between synchronous and asynchronous sequential circuit.

Ans: Refer Q. 4.12, Unit-4.

- Q. 5. An asynchronous sequential logic circuit is described by the following excitation and output function

$$Y = X_1X_2 + (X_1 + X_2)Y$$

$$Z = Y$$

- Draw the logic diagram of the circuit.
- Derive the transition table and output map.
- Describe the behavior of the circuit.

Ans: Refer Q. 4.15, Unit-4.

- Q. 6. Implement the circuit defined by the following transition table with a NOR SR Latch. Also show the implementation with NAND SR latch.

	$x_1 x_2$			
$y$	00	01	11	10
0	0	0	0	1
1	0	0	1	1

Fig. 2.

Ans: Refer Q. 4.18, Unit-4.

Q. 7. Illustrate the state reduction technique for digital circuits.  
Ans: Refer Q. 4.19, Unit-4.

Q. 8. What are critical race and non-critical race ? How can they be avoided ?

Ans: Refer Q. 4.21, Unit-4.

Q. 9. Explain the term Hazard. Define different types of Hazards along with detection and reduction of Hazards.  
Ans: Refer Q. 4.22, Unit-4.

## 5 UNIT

## Memory and Programmable Logic Devices

### CONTENTS

- Part-1 : Digital Logic Families : ..... 5-2G to 5-17G  
DTL, DCTL, TTL, ECL and CMOS etc.
- Part-2 : Fan-out, Fan-in ..... 5-17G to 5-18G
- Part-3 : Noise Margin ..... 5-18G to 5-19G
- Part-4 : RAM, ROM ..... 5-19G to 5-25G
- Part-5 : PLA, PAL ..... 5-25G to 5-30G
- Part-6 : Circuits of Logic Families, ..... 5-30G to 5-32G  
Interfacing of Digital Logic Families
- Part-7 : Circuit Implementation Using ..... 5-32G to 5-39G  
ROM, PLA and PAL

**PART - 1****Digital Logic Families : DTL, DCTL, TTL, ECL and CMOS etc.**

**Que 5.1.** Discuss in brief about digital logic family.

**Answer**

1. The set of compatible ICs with the same logic levels and same supply voltages have been fabricated to perform the various logic functions known as logic family.
2. Based on the fabrication technology, logic families are classified into two types :
  - A. **Bipolar logic family :** Transistors and diodes are bipolar devices, in which the current flows because of both electrons and holes being charge carriers. On the basis of operations of transistors in ICs, bipolar logic families are further classified as :
    - i. Saturated bipolar logic families : In saturated bipolar logic families, transistors operate in saturation region. The speed of saturated bipolar logic family is low. Saturated bipolar logic families are listed below :
      - i. Resistor-transistor logic (RTL)
      - ii. Direct-coupled transistor logic (DCTL)
      - iii. Diode-transistor logic (DTL)
      - iv. High-threshold logic (HTL)
      - v. Transistor-transistor logic (TTL)
      - vi. Integrated injection logic (IIL or I<sup>2</sup>L)
    - b. Unsaturated bipolar logic families : In unsaturated bipolar logic families, transistors operate in active region. The speed of unsaturated bipolar logic families is high as compared to saturated logic families. Unsaturated bipolar logic families are listed below :
      1. Schottky transistor-transistor logic
      2. Emitter-coupled logic (ECL)
  - B. **Unipolar logic family :**
    1. The unipolar families include p-channel metal-oxide semiconductor field-effect transistor (PMOS), n-channel metal-oxide semiconductor field-effect transistor NMOS and CMOS.
    2. RTL, DCTL, DTL, and HTL families are obsolete. The logic families TTL, ECL, IIL, MOS, and CMOS are currently in use.

**Que 5.2.** What are the features of logic family ?

**Answer**

- The main features of a logic family are given below :
1. **High fan-out :** High fan-out is advantageous as it reduces the need of additional drivers to drive more gates.
  2. **High noise immunity :** The ability of circuits to operate reliably in noisy environments is important in many applications. Therefore, it should be highly immune to noise.
  3. **Low-power dissipation and high speed :** To achieve low-power dissipation and high speed of operation, the digital circuit must have a minimum number of gates between input and output.
  4. **Small size :** The size of IC should be small as much as possible.
  5. **Flexibilities available :** It must be considered while selecting a logic family for a particular application for the purpose of interfacing with other logic family devices. It should have good interfacing.

**Que 5.3.** Explain diode transistor logic (DTL).

**Answer**

1. It is a technology for designing and fabricating digital circuits where in logic gates employ both diodes and transistors.
2. The three-input DTL NAND gate is shown in Fig. 5.3.1.
3. The purpose of the input resistors in Fig. 5.3.1(a) is to isolate the inputs from one another.

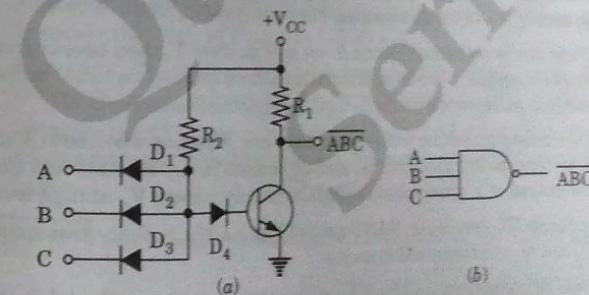


Fig. 5.3.1. DTL NAND gate; (a) Schematic, (b) Symbol.

4. The circuit shown is essentially a diode and circuit with a transistor inverter to provide the NAND function. The diodes provide much better input isolation than resistors.
5. Although the gates in the DTL family can switch states faster than the gates in the RTL family, the diodes place a limit on the switching speed due to the charge stored in their junctions.

6. A logic family designed to get around this problem is TTL (transistor-transistor logic) family.
7. The DTL family is probably the easiest logic to use, and is also the cheapest form of logic available for medium-speed applications.

**Que 5.4.** Explain direct coupled transistor logic (DCTL) of logic family.

**Answer**

1. Fig. 5.4.1(a) shows the circuit for DCTL-NOR gate and Fig. 5.4.1(b) shows the circuit for DCTL-NAND gates.

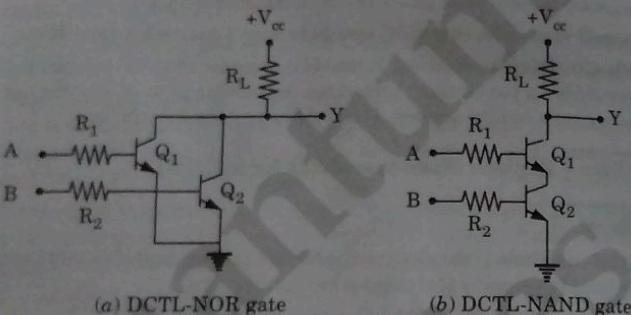


Fig. 5.4.1.

2. In DCTL - two input NOR gate of Fig. 5.4.1(a), there are two transistors Q<sub>1</sub> and Q<sub>2</sub> which share a common load resistor R<sub>L</sub>.
3. If any one or both inputs A and B are high at level '1', then base current will be supplied to one or both the transistors, causing them to conduct and thus collector current will flow through load resistor R<sub>L</sub>.
4. This makes the output voltage at Y to go low. Whereas when both A and B are low, both transistors Q<sub>1</sub> and Q<sub>2</sub> remain in cut-off state and thus the output at Y approaches that of the supply voltage V<sub>cc</sub> at high level.
5. In the DCTL - two input NAND gate of Fig. 5.4.1(b), both the transistors Q<sub>1</sub> and Q<sub>2</sub> are connected in series.
6. Only when both inputs A and B are high, both transistors Q<sub>1</sub> and Q<sub>2</sub> will conduct, as they are in series. This causes output voltage at Y to go low.
7. When any one or both inputs A and B are low, both transistors Q<sub>1</sub> and Q<sub>2</sub> cannot conduct. Therefore the output voltage at Y remains high.
8. The transistor with lower input impedance draws more current than the other transistors.

9. The transistors which draw less current will not turn on properly and give rise to malfunctioning of the circuit. This phenomenon is known as current hogging or more precisely as base current hogging.

**Que 5.5.** Define the TTL (Transistor-Transistor Logic) logic family used for digital circuits. AKTU 2022-23 (Sem-3), Marks 10

**Answer**

1. TTL (Transistor-Transistor Logic) is a popular logic family widely used in digital circuits.
2. It is based on bipolar junction transistors and was one of the earliest and most commonly used logic families.
3. TTL logic family offers a good balance between performance, simplicity, and cost-effectiveness.
4. Following are some key aspects of TTL:
  - i. **Technology :** TTL logic family is based on bipolar transistors, specifically NPN and PNP transistors.
  - ii. **Voltage levels :** TTL operates with a power supply voltage typically around 5 V, making it compatible with many digital systems.
  - iii. **Noise margins :** TTL provides relatively large noise margins, meaning it is less susceptible to noise or voltage fluctuations.
  - iv. **Speed and performance :** TTL offers fast switching speeds, making it suitable for applications that require high-speed operation.
  - v. **Power consumption :** TTL logic family consumes more power compared to some other logic families like CMOS.
  - vi. **Fan-out :** TTL gates have limited fan-out capability. Additional buffering or line drivers may be needed for larger fan-outs.
  - vii. **Interfacing :** TTL logic signals may require level shifting or buffering when interfacing with other logic families that operate at different voltage levels.

**Que 5.6.** Draw and explain the operation of a TTL NAND gate. AKTU 2017-18 (Sem-3), Marks 07

**Answer**

1. The circuit of the two-input TTL NAND gate is shown in Fig. 5.6.1. The input transistor, Q<sub>1</sub> is a multiple emitter transistor.

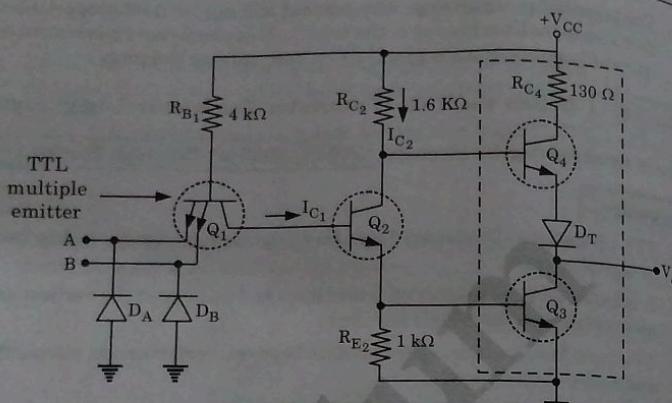


Fig. 5.6.1. TTL NAND gates.

2. Transistor  $Q_2$  is called the phase splitter. Emitter of transistor  $Q_2$  is connected to collector of transistor  $Q_4$  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 :

Table 5.6.1 explains the operation of TTL NAND gate with totem-pole having two inputs.

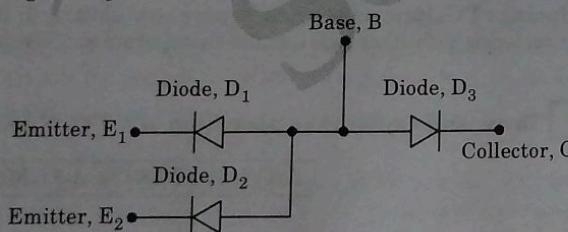


Fig. 5.6.2. Diode equivalent of TTL multiple emitters.

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  $Q_1$ .

Table 5.6.1. Operation of TTL NAND gate

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

Que 5.7. Describe the construction and operation of TTL NOR gate.

#### Answer

##### TTL NOR gate :

1. The circuit of the two-input TTL NOR gate is shown in Fig. 5.7.1. Two input transistors  $Q_A$  and  $Q_B$  are emitter transistors.

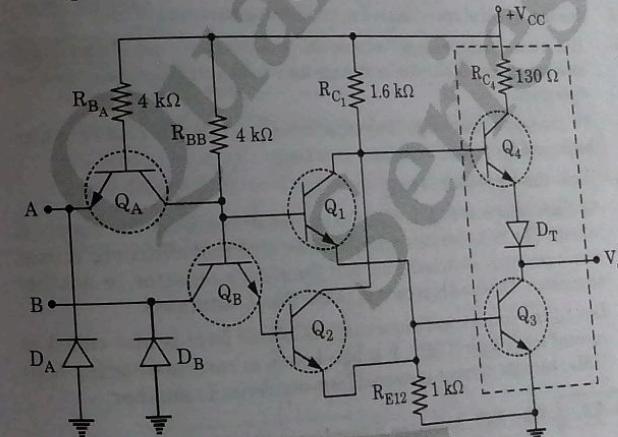


Fig. 5.7.1. TTL NOR gate.

2. Transistor  $Q_1$  and  $Q_2$  are called the phase splitters. Emitter of transistor  $Q_3$  is connected to collector of transistor  $Q_4$  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.

3. When negative spikes appear at the input terminals, the diodes conduct and bypass the spikes to ground.
4. Diode  $D_T$  ensures that transistors  $Q_3$  and  $Q_4$  do not conduct simultaneously. Transistor  $Q_3$  acts as an emitter follower.

**Operation :****Table 5.7.1.** Operation of TTL NOR gate.

Inputs		Transistors				Output $V_o$
A	B	$Q_A$	$Q_B$	$Q_1$	$Q_2$	
0	0	Forward bias (ON)	Forward bias (ON)	OFF	OFF	OFF
0	1	Forward bias (ON)	Reverse bias (OFF)	OFF	ON	ON
1	0	Reverse bias (OFF)	Forward bias (ON)	ON	OFF	OFF
1	1	Reverse bias (OFF)	Reverse bias (OFF)	ON	ON	ON

**Que 5.8.** Discuss about emitter coupled logic (ECL).**Answer**

1. ECL prevents transistor saturation, thereby increasing overall switching speed. The ECL is formed when BJTs are coupled at their emitters. It is also called current-mode logic or current-steering logic.
2. Its fast speed of operation is due to following reasons :
  - i. It is a non-saturated logic, because the transistors are not allowed to go into saturation. It eliminates storage time delays and increases speed of operation.
  - ii. To charge and discharge stray capacitances quickly, currents are kept high, and the output impedance is so low.
  - iii. It has limited voltages swing.
3. The advantage of ECL is that the current drawn from the supply is steady. The ECL operates on the principle of current switching. A fixed bias current is switched from one transistor's collector to another transistor. The fixed bias current is less than  $I_C$  saturated.
4. Due to the current mode operation, ECL logic form is also called as current mode logic (CML). It is also known as current-steering logic (CSL), because current is steered from one device to another.

**Que 5.9.** Why ECL is better ? Implement NAND gate with DTL and TTL.

**AKTU 2022-23 (Sem-4), Marks 10****Answer****A. Reasons why ECL is better :**

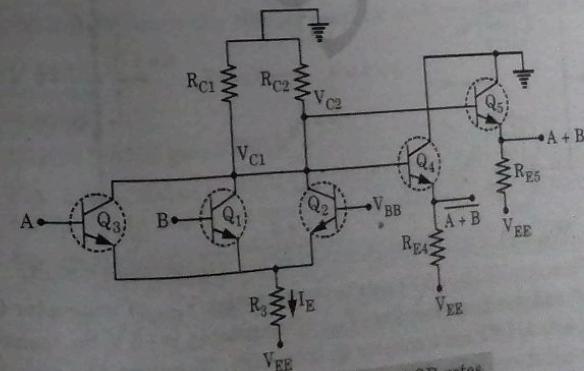
1. **High-speed operation :** ECL is known for its fast switching speeds and low propagation delay. It is often used in high-speed applications where timing precision is critical, such as in telecommunications and high-performance computing.
2. **Excellent noise margin :** ECL provides a superior noise margin, important in demanding environments.
3. **Differential signaling :** ECL is often implemented as differential logic, providing inherent noise immunity and making it suitable for high-speed data transmission.
4. **Well-suited for high-frequency applications :** ECL's ability to operate at high frequencies makes it suitable for applications like microwave and millimeter-wave circuitry.

**B. NAND gate with DTL and TTL :** Refer Q. 5.3, Page 5-3G and Q. 5.6, Page 5-5G, Unit-5.

**Que 5.10.** Describe the construction and operation of ECL OR/NOR gate.

**Answer****ECL OR/NOR gate :**

1. A two-input ECL OR/NOR gate is shown in Fig. 5.10.1.
2. It has two outputs which are complements of each other. Transistors  $Q_2$  and  $Q_1$  form a differential amplifier. Transistors  $Q_1$  and  $Q_3$  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 V base to emitter drops).

**Fig. 5.10.1. ECL NOR and OR gates.**

3. When negative spikes appear at the input terminals, the diodes conduct and bypass the spikes to ground.
4. Diode  $D_T$  ensures that transistors  $Q_3$  and  $Q_4$  do not conduct simultaneously. Transistor  $Q_3$  acts as an emitter follower.

**Operation :**

Table 5.7.1. Operation of TTL NOR gate.

Inputs		Transistors		Output			
A	B	$Q_A$	$Q_B$	$Q_1$	$Q_2$	$Q_3$	$Q_4$
0	0	Forward bias (ON)	Forward bias (ON)	OFF	OFF	OFF	ON
0	1	Forward bias (ON)	Reverse bias (OFF)	OFF	ON	ON	OFF
1	0	Reverse bias (OFF)	Forward bias (ON)	ON	OFF	ON	OFF
1	1	Reverse bias (OFF)	Reverse bias (OFF)	ON	ON	ON	OFF

**Que 5.8.** Discuss about emitter coupled logic (ECL).

**Answer**

1. ECL prevents transistor saturation, thereby increasing overall switching speed. The ECL is formed when BJTs are coupled at their emitters. It is also called current-mode logic or current-steering logic.
2. Its fast speed of operation is due to following reasons :
  - i. It is a non-saturated logic, because the transistors are not allowed to go into saturation. It eliminates storage time delays and increases speed of operation.
  - ii. To charge and discharge stray capacitances quickly, currents are kept high, and the output impedance is so low.
  - iii. It has limited voltages swing.
3. The advantage of ECL is that the current drawn from the supply is steady. The ECL operates on the principle of current switching. A fixed bias current is switched from one transistor's collector to another transistor. The fixed bias current is less than  $I_C$  saturated.
4. Due to the current mode operation, ECL logic form is also called as current mode logic (CML). It is also known as current-steering logic (CSL), because current is steered from one device to another.

**Que 5.9.** Why ECL is better ? Implement NAND gate with DTL and TTL.

AKTU 2022-23 (Sem-4), Marks 10

**Answer**

**A. Reasons why ECL is better :**

1. **High-speed operation :** ECL is known for its fast switching speeds and low propagation delay. It is often used in high-speed applications where timing precision is critical, such as in telecommunications and high-performance computing.
2. **Excellent noise margin :** ECL provides a superior noise margin, important in demanding environments.
3. **Differential signaling :** ECL is often implemented as differential logic, providing inherent noise immunity and making it suitable for high-speed data transmission.
4. **Well-suited for high-frequency applications :** ECL's ability to operate at high frequencies makes it suitable for applications like microwave and millimeter-wave circuitry.

**B. NAND gate with DTL and TTL :** Refer Q. 5.3, Page 5-3G and Q. 5.6, Page 5-5G, Unit-5.

**Que 5.10.** Describe the construction and operation of ECL OR/NOR gate.

**Answer**

**ECL OR/NOR gate:**

1. A two-input ECL OR/NOR gate is shown in Fig. 5.10.1.
2. It has two outputs which are complements of each other. Transistors  $Q_2$  and  $Q_1$  form a differential amplifier. Transistors  $Q_1$  and  $Q_3$  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 V base to emitter drops).

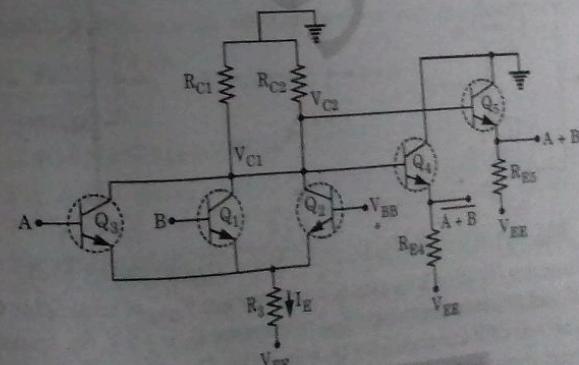
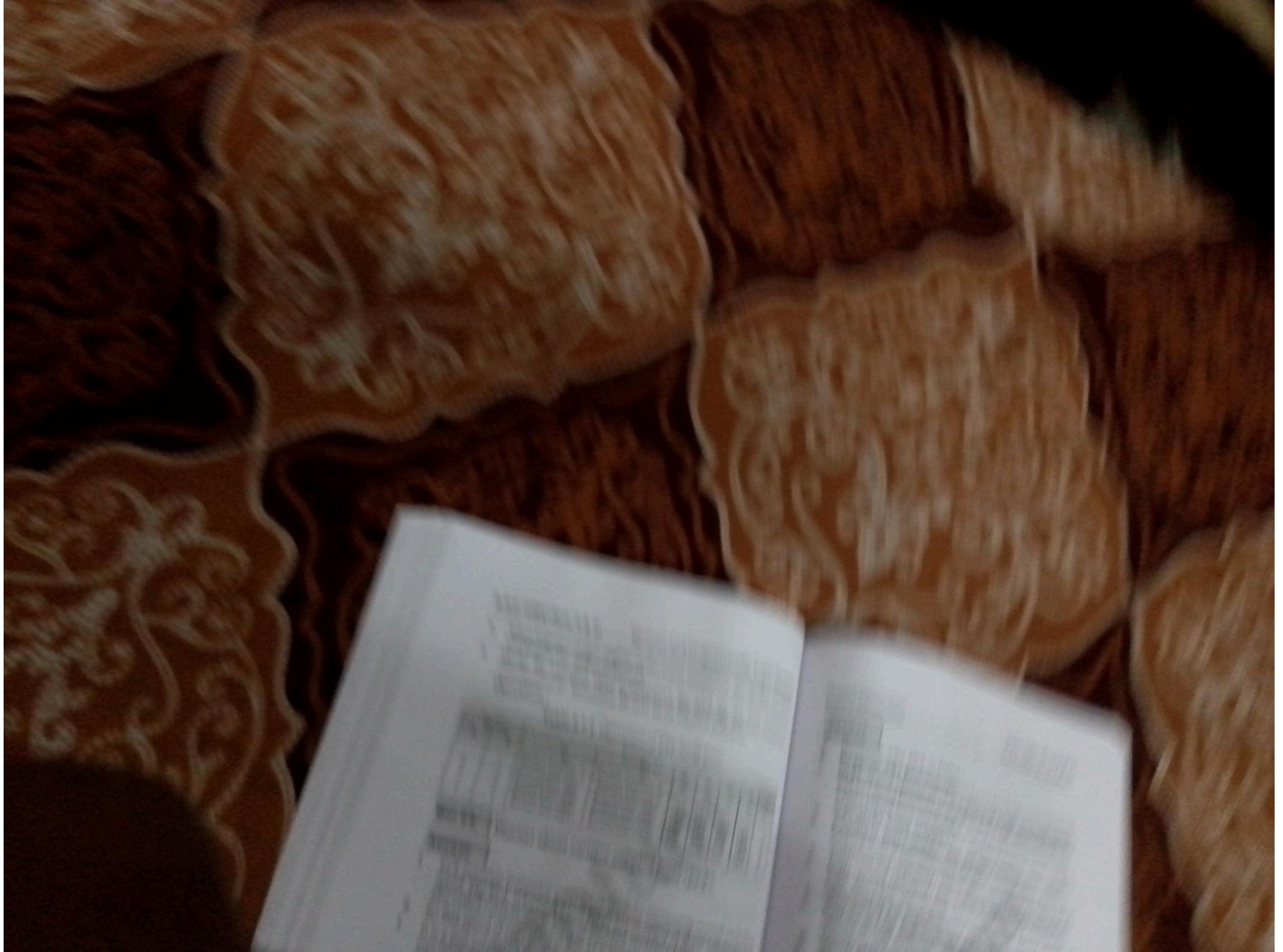


Fig. 5.10.1. ECL NOR and OR gates.



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

Table 5.10.1. Operation of ECL OR/NOR gate.

Inputs		Transistors					Output	
A	B	$Q_3$	$Q_1$	$Q_2$	$Q_4$	$Q_5$	$A + B$	$\bar{A} + \bar{B}$
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

Que 5.11. Describe the circuit and performance of CMOS inverter and state the characteristics of CMOS.

AKTU 2018-19 (Sem-3), Marks 3.5

### Answer

CMOS logic family uses both *p*- and *n*-channel MOSFET in the same circuit to have advantage over the PMOS and NMOS logic families.

#### CMOS inverter :

- 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.
- When  $A$  is LOW ( $0\text{ V}$ ). 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. 5.11.1(b) results in  $V_o$  as logic HIGH that is  $+5\text{ V}$ .

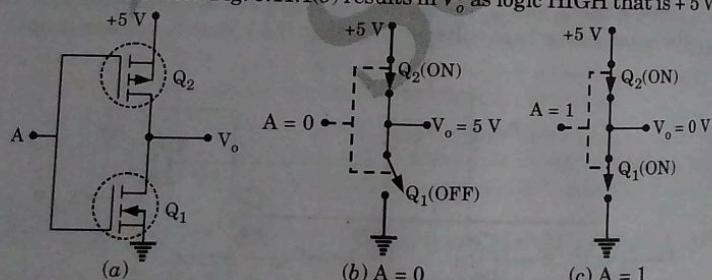


Fig. 5.11.1. (a) CMOS as inverter, (b) and (c) equivalent circuit.

- 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. 5.11.1(c) results with  $V_o$  as logic LOW that is  $0\text{ V}$ .

Table 5.11.1. Operation of CMOS inverter.

Input, A	<i>p</i> -channel MOSFET, $Q_1$	<i>n</i> -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 :

**Supply voltage :** The 4000 and 74C series can operate with  $V_{DD}$  values ranging from 3 to  $15\text{ V}$ . The 74HC and 74HCT series can operate with  $V_{DD}$  values ranging from 2 to  $6\text{ V}$ .

**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.

**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  $2.5\text{nW}$  per gate when  $V_{DD} = 5\text{ V}$ , and it increases to  $10\text{nW}$  per gate when  $V_{DD} = 10\text{ V}$ .

**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 ( $0\text{ V}$  or  $V_{DD}$ ) or to another input.

Que 5.12. Discuss the circuit diagram and operation of CMOS NAND gate.

### Answer

#### CMOS NAND gate :

- Fig. 5.12.1 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 ( $0\text{ V}$ ) and  $B$  is also LOW ( $0\text{ V}$ ), *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.,  $+5\text{ V}$ .

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

Table 5.10.1. 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

Que 5.11. Describe the circuit and performance of CMOS inverter and state the characteristics of CMOS.

AKTU 2018-19 (Sem-3), Marks 3.5

### Answer

CMOS logic family uses both *p*- and *n*-channel MOSFET in the same circuit to have advantage over the PMOS and NMOS logic families.

#### CMOS inverter :

- 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.
- When  $A$  is LOW ( $0\text{ V}$ ). 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. 5.11.1(b) results in  $V_o$  as logic HIGH that is  $+5\text{ V}$ .

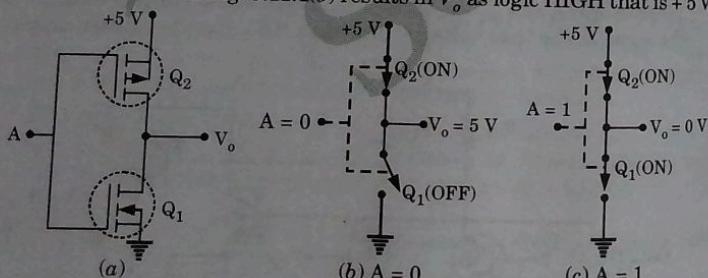


Fig. 5.11.1. (a) CMOS as inverter, (b) and (c) equivalent circuit.

- 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. 5.11.1(c) results with  $V_o$  as logic LOW that is  $0\text{ V}$ .

Table 5.11.1. Operation of CMOS inverter.

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

Truth table :

A	$V_o$
0	1
1	0

#### Characteristics of CMOS :

**Supply voltage :** The 4000 and 74C series can operate with  $V_{DD}$  values ranging from  $3$  to  $15\text{ V}$ . The 74HC and 74HCT series can operate with  $V_{DD}$  values ranging from  $2$  to  $6\text{ V}$ .

**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.

**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  $2.5\text{ nW}$  per gate when  $V_{DD} = 5\text{ V}$ , and it increases to  $10\text{ nW}$  per gate when  $V_{DD} = 10\text{ V}$ .

**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 ( $0\text{ V}$  or  $V_{DD}$ ) or to another input.

Que 5.12. Discuss the circuit diagram and operation of CMOS NAND gate.

### Answer

#### CMOS NAND gate :

- Fig. 5.12.1 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 ( $0\text{ V}$ ) and  $B$  is also LOW ( $0\text{ V}$ ), *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.,  $+5\text{ V}$ .

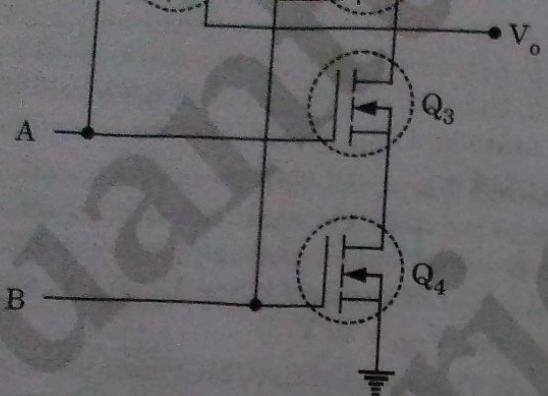


Fig. 5.12.1. CMOS as NAND gate.

### 5.12.1. Switching operation of CMOS NAND gate

p-channel MOSFET	n-channel MOSFET	Output		
$Q_1$	$Q_2$	$Q_3$	$Q_4$	$X$
ON	ON	OFF	OFF	1
ON	OFF	OFF	ON	1
OFF	ON	ON	OFF	1
OFF	OFF	ON	ON	0

Discuss the circuit diagram and operation of CMOS NOR

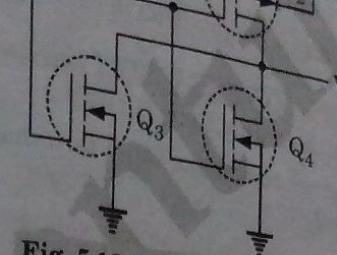


Fig. 5.13.1. CMOS as NOR.

Table 5.13.1. Switching operation of CMOS NOR gate

Inputs		<i>p</i> -channel MOSFET		<i>n</i> -channel MOSFET		Output
A	B	$Q_1$	$Q_2$	$Q_3$	$Q_4$	$X$
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 5.14. Construct the following logic gates from NMOS and PMOS logic families

i. NAND

ii. NOR

AKTU 2021-22 (Sem-4), Marks 10

### Answer

#### i. NMOS NAND gate:

1. The Fig. 5.14 shown is the circuit of the 2-input NMOS NAND gate.
2. It consists of three N-channel MOSFETs, in which  $Q_1$  acts as the load resistance, whereas  $Q_2$  and  $Q_3$  act as the switching MOSFETs. The two inputs A and B are given to MOSFET  $Q_2$  and  $Q_3$  respectively.

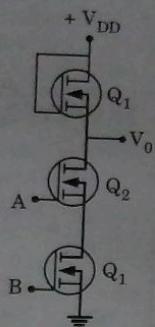


Fig. 5.14.1. 2-input NMOS NAND gate.

3. When either  $A$  or  $B$  is LOW, the MOSFET with low input will be turned OFF, thus making the output to be HIGH.
4. If HIGH input is given to both the input terminals, it will turn ON the MOSFETs  $Q_2$  and  $Q_3$ .
5. Therefore, the current due to the supply voltage  $V_{DD}$  will flow towards the ground making the output as LOW.

#### ii. NMOS NOR gate :

1. The following circuit shows the circuit of the 2-input NMOS NOR gate.
2. It has three N-channel MOSFETs, in which  $Q_1$  acts as the load resistance, MOSFETs  $Q_2$  and  $Q_3$  act as switching devices.
3. If Low input is given at both the inputs, it will turn OFF both the MOSFETs  $Q_2$  and  $Q_3$ , thereby making the output to be HIGH.

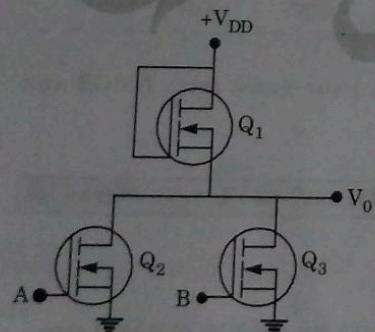


Fig. 5.14.2. 2-input NMOS NOR gate.

Table 5.14.2 :

A	B	$Q_2$	$Q_3$	$V_0$
0	0	OFF	OFF	1 (High)
0	1	OFF	ON	0 (Low)
1	0	ON	OFF	0 (Low)
1	1	ON	ON	0 (Low)

#### iii. PMOS NAND gates :

1. PMOS NAND gate is shown in Fig. 5.14.3. It is similar to its NMOS counterpart except for the fact that the supply voltage is negative. It employs negative logic.  $Q_1$  is ON as its gate is tied to  $-10\text{ V}$ .

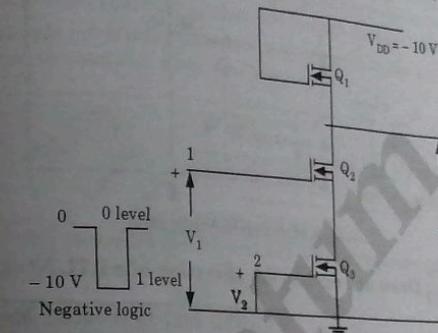


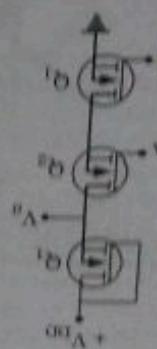
Fig. 5.14.3. PMOS NAND gate.

2. When both the inputs  $V_1$  and  $V_2$  are 1 ( $= -10\text{ V}$ ),  $Q_2$  and  $Q_3$  are ON,  $V_0 = 0 (= 0\text{ V})$ .
3. When both the inputs  $V_1$  and  $V_2$  are 0 ( $= 0\text{ V}$ ),  $Q_2$  and  $Q_3$  are OFF,  $V_0 = 1 (= -10\text{ V})$ .
4. When  $V_1 = 1 (= -10\text{ V})$  and  $V_2 = 0 (= 0\text{ V})$ ,  $Q_2$  is ON and  $Q_3$  is OFF,  $V_0 = 1 (= -10\text{ V})$ .
5. When  $V_1 = 0 (= 0\text{ V})$  and  $V_2 = 1 (= -10\text{ V})$ ,  $Q_2$  is OFF and  $Q_3$  is ON,  $V_0 = 1 (= -10\text{ V})$ .

#### iv. PMOS NOR Gates :

1. PMOS NOR gate is shown in Fig. 5.14.4.  $Q_1$  is always ON as its gate is tied to  $-10\text{ V}$ .
2. When both the inputs  $V_1$  and  $V_2$  are 1 ( $= -10\text{ V}$ ),  $Q_2$  and  $Q_3$  are ON,  $V_0 = 0 (= 0\text{ V})$ .
3. When both the inputs  $V_1$  and  $V_2$  are 0 ( $= 0\text{ V}$ ),  $Q_2$  and  $Q_3$  are OFF,  $V_0 = 1 (= -10\text{ V})$ .
4. When  $V_1 = 1 (= -10\text{ V})$  and  $V_2 = 0 (= 0\text{ V})$ ,  $Q_2$  is ON and  $Q_3$  is OFF,  $V_0 = 0 (= 0\text{ V})$ .
5. When  $V_1 = 0 (= 0\text{ V})$  and  $V_2 = 1 (= -10\text{ V})$ ,  $Q_2$  is OFF and  $Q_3$  is ON,  $V_0 = 1 (= 0\text{ V})$ .

Fig. 6.14.1. 2-input NMOS NAND gate.



A	B	$Q_1$	$Q_2$	$V_o$
1	1	ON	ON	0 (Low)
1	0	ON	OFF	0 (Low)
0	1	OFF	ON	1 (High)
0	0	OFF	OFF	1 (High)

Table 6.14.1

1. PMOS NAND gate is shown in Fig. 6.14.2. It is similar to its NMOS counterpart except for the fact that the supply voltage is negative. It employs a negative logic.  $Q_1$  is ON as its gate is tied to  $-10\text{ V}$ .
2. PMOS NAND gate is shown in Fig. 6.14.3. It is similar to its NMOS counterpart except for the fact that the supply voltage is negative. It employs a negative logic.  $Q_1$  is ON as its gate is tied to  $-10\text{ V}$ .
3. PMOS NAND gate is shown in Fig. 6.14.4. It is similar to its NMOS counterpart except for the fact that the supply voltage is negative. It employs a negative logic.  $Q_1$  is ON as its gate is tied to  $-10\text{ V}$ .

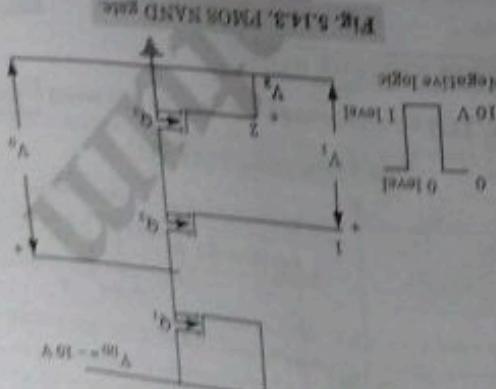


Fig. 6.14.2. 2-input NMOS NOR gate.

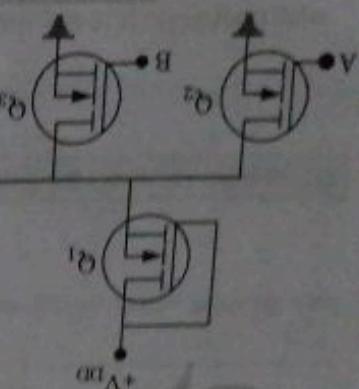


Table 6.14.2

A	B	$Q_1$	$Q_2$	$V_o$
1	1	ON	ON	0 (Low)
1	0	ON	OFF	0 (Low)
0	1	OFF	ON	1 (High)
0	0	OFF	OFF	1 (High)

Fig. 6.14.2. 2-input NMOS NOR gate.

1. PMOS NOR gate is shown in Fig. 6.14.4.  $Q_1$  is always ON as its gate is tied to  $-10\text{ V}$ .
2. When both the inputs  $V_1$  and  $V_2$  are  $1 (-10\text{ V})$ ,  $Q_1$  and  $Q_2$  are OFF,  $V_o = 0 (= 0\text{ V})$ .
3. When both the inputs  $V_1$  and  $V_2$  are  $0 (= 0\text{ V})$ ,  $Q_1$  and  $Q_2$  are OFF,  $V_o = 1 (= -10\text{ V})$ .
4. When  $V_1 = 1 (= -10\text{ V})$  and  $V_2 = 0 (= 0\text{ V})$ ,  $Q_1$  is ON and  $Q_2$  is OFF,  $V_o = 0 (= 0\text{ V})$ .
5. When  $V_1 = 0 (= 0\text{ V})$  and  $V_2 = 1 (= -10\text{ V})$ ,  $Q_1$  is OFF and  $Q_2$  is ON,  $V_o = 1 (= -10\text{ V})$ .
6. When  $V_1 = 0 (= 0\text{ V})$  and  $V_2 = 0 (= 0\text{ V})$ ,  $Q_1$  is OFF and  $Q_2$  is ON,  $V_o = 1 (= -10\text{ V})$ .

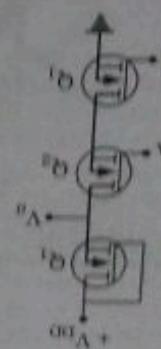
#### PMOS NOR Gates:

1. The following circuit shows the supply voltage  $V_{DD}$  will flow towards the ground making the output as LOW.
2. Therefore, the current due to the supply voltage  $V_{DD}$  will flow towards MOSFETs  $Q_1$  and  $Q_2$ .
3. When both the inputs  $V_1$  and  $V_2$  are  $0 (= 0\text{ V})$ ,  $Q_1$  and  $Q_2$  are OFF,  $V_o = 1 (= -10\text{ V})$ .
4. When  $V_1 = 1 (= -10\text{ V})$  and  $V_2 = 0 (= 0\text{ V})$ ,  $Q_1$  is ON and  $Q_2$  is OFF,  $V_o = 1 (= -10\text{ V})$ .
5. When  $V_1 = 0 (= 0\text{ V})$  and  $V_2 = 1 (= -10\text{ V})$ ,  $Q_1$  is OFF and  $Q_2$  is ON,  $V_o = 1 (= -10\text{ V})$ .
6. When both the inputs  $V_1$  and  $V_2$  are  $1 (= -10\text{ V})$ ,  $Q_1$  and  $Q_2$  are ON,  $V_o = 0 (= 0\text{ V})$ .

Fig. 6.14.3. PMOS NAND gate.

1. NMOS NOR gate:
2. The following circuit shows the current of the 2-input NMOS NOR gate.
3. If Low input is given at both the inputs, it will turn OFF both the MOSFETs  $Q_1$  and  $Q_2$ , thereby making the output to be HIGH.
4. If High input is given to both the input terminals, it will turn ON the MOSFETs  $Q_1$  and  $Q_2$  act as switching devices.
5. It has three N-channel MOSFETs, in which  $Q_1$  acts as the load resistance.
6. The ground making the output as LOW.
7. Therefore, the current due to the supply voltage  $V_{DD}$  will flow towards MOSFETs  $Q_1$  and  $Q_2$ .
8. When either A or B is LOW, the MOSFET with low input will be turned OFF, thus making the output to be HIGH.
9. When either A or B is HIGH, the MOSFET with high input will be turned ON to both the input terminals, it will turn ON the MOSFETs  $Q_1$  and  $Q_2$ .

Fig. 6.14.4. 2-input PMOS NAND gate.



A	B	$Q_1$	$Q_2$	$V_o$
1	1	ON	ON	1 (High)
1	0	ON	OFF	1 (High)
0	1	OFF	ON	1 (High)
0	0	OFF	OFF	1 (High)

Table 6.14.3

- iii) PMOS NAND Gates:
- Fig. 5.143 shows a 2-input PMOS NAND gate circuit. It consists of two PMOS transistors with their drains connected together. The source of one transistor is connected to the output  $V_o$ , and its gate is connected to input  $V_1$ . The source of the second transistor is also connected to the output  $V_o$ , and its gate is connected to input  $V_2$ . The drain of the second transistor is connected to the drain of the first transistor. The common drain connection is connected to ground. The supply voltage  $V_{DD}$  is connected to the common source connection of both transistors.
- The circuit implements negative logic. When either input  $V_1$  or  $V_2$  is high (1 level), the corresponding PMOS transistor turns off, pulling the output  $V_o$  high. When both inputs  $V_1$  and  $V_2$  are low (0 level), both transistors turn on, connecting the output  $V_o$  to ground.

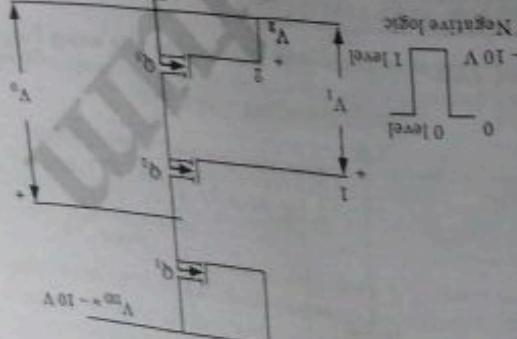


Fig. 5.143. PMOS NAND gate.

- ii) NMOS NOR Gates:
- Fig. 5.144 shows a 2-input NMOS NOR gate circuit. It consists of two NMOS transistors with their drains connected together. The source of one transistor is connected to the output  $V_o$ , and its gate is connected to input  $V_1$ . The source of the second transistor is also connected to the output  $V_o$ , and its gate is connected to input  $V_2$ . The drain of the second transistor is connected to the drain of the first transistor. The common drain connection is connected to ground. The supply voltage  $V_{DD}$  is connected to the common source connection of both NMOS transistors.
- The circuit implements negative logic. When either input  $V_1$  or  $V_2$  is high (1 level), the corresponding NMOS transistor turns off, pulling the output  $V_o$  high. When both inputs  $V_1$  and  $V_2$  are low (0 level), both transistors turn on, connecting the output  $V_o$  to ground.

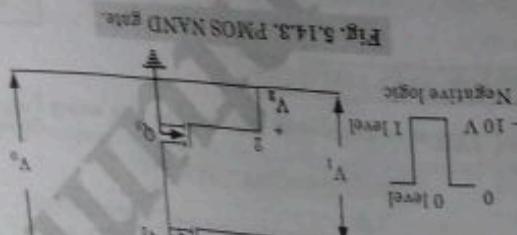


Fig. 5.144. NMOS NOR gate.

- i) PMOS NOR Gates:
- Fig. 5.145 shows a 2-input PMOS NOR gate circuit. It consists of two PMOS transistors with their drains connected together. The source of one transistor is connected to the output  $V_o$ , and its gate is connected to input  $V_1$ . The source of the second transistor is also connected to the output  $V_o$ , and its gate is connected to input  $V_2$ . The drain of the second transistor is connected to the drain of the first transistor. The common drain connection is connected to ground. The supply voltage  $V_{DD}$  is connected to the common source connection of both PMOS transistors.
- The circuit implements negative logic. When either input  $V_1$  or  $V_2$  is high (1 level), the corresponding PMOS transistor turns off, pulling the output  $V_o$  high. When both inputs  $V_1$  and  $V_2$  are low (0 level), both transistors turn on, connecting the output  $V_o$  to ground.

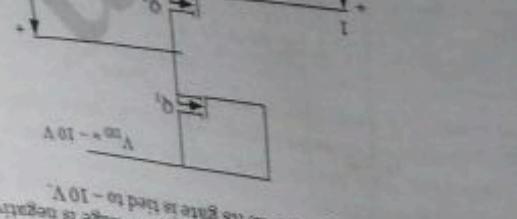


Fig. 5.145. PMOS NOR gate.

Fig. 5.142. 2-input NMOS NOR gate.

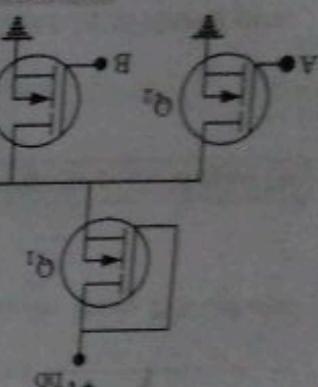


Table 5.142:

A	B	$Q_1$	$Q_2$	$Q_3$	$V_o$
1	1	1	ON	ON	0 (Low)
1	0	0	ON	OFF	0 (Low)
0	1	OFF	ON	1 (High)	
0	0	OFF	OFF	1 (High)	

Table 5.141:

A	B	$Q_1$	$Q_2$	$Q_3$	$V_o$
1	1	1	ON	ON	0 (Low)
1	0	0	ON	OFF	1 (High)
0	1	OFF	ON	1 (High)	
0	0	OFF	OFF	1 (High)	

Table 5.141:

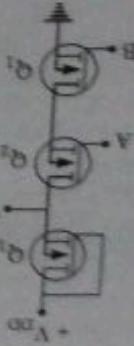


Fig. 5.141. 2-input PMOS NOR gate.

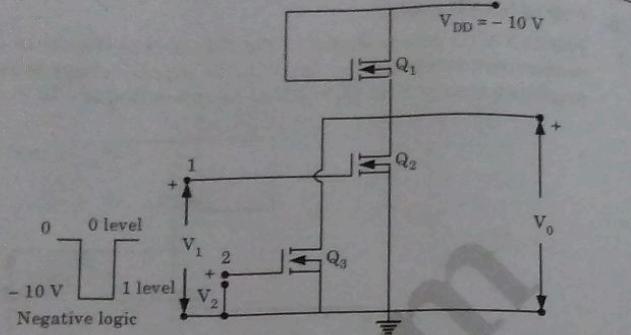


Fig. 5.14.4. PMOS NOR gate.

**Que 5.15.** Draw and explain the operation of a RTL NOR gate.

AKTU 2017-18 (Sem-3), Marks 07

**Answer**

- The basic circuit of the RTL digital logic family is the NOR gate shown in Fig. 5.15.1. Each input is associated with one resistor and one transistor.

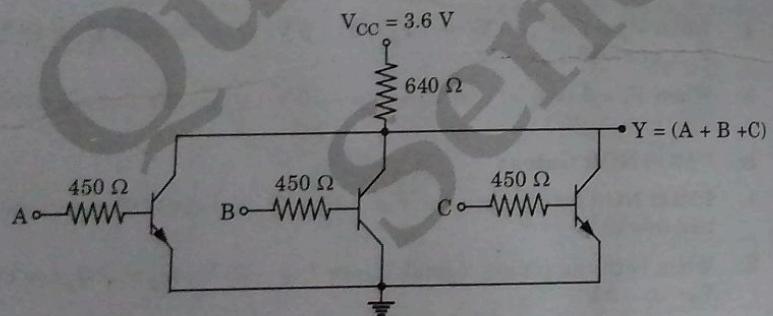


Fig. 5.15.1. Basic RTL NOR gate.

- 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.
- 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.

- 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  $V_{CC}$ .
- The fan-out of the RTL gate is limited by a high output voltage. As the output is loaded with inputs of other gates, more current is consumed by the load. This current must flow through the  $640 \Omega$  resistor.
- A simple calculation shows that if  $h_{FE}$  drops to 20, the output voltage drops to about 1 V when the fan-out is 5.

**PART-2**  
Fan-out, Fan-in.

**Que 5.16.** Describe the fan-out and fan-in condition of the digital logic gate.

**Answer**

**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.
- 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.
- High state fan-out is given by

$$\text{HIGH state fan-out} = \frac{I_{OH}}{I_{IH}} \quad \dots(5.16.1)$$

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.

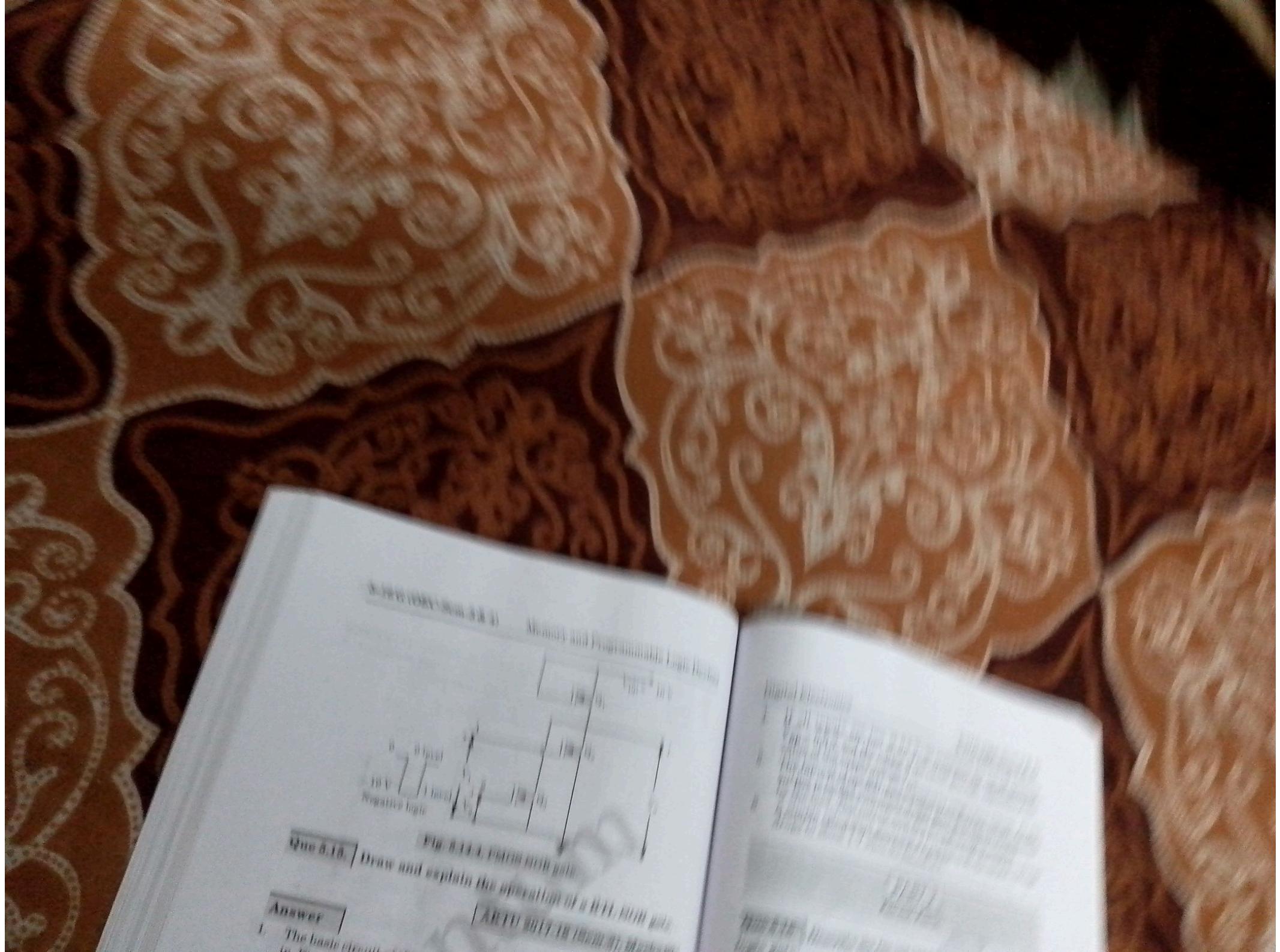
- Similarly, low state fan-out is given by

$$\text{LOW state fan-out} = \frac{I_{OL}}{I_{IL}} \quad \dots(5.16.2)$$

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.

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

$$\text{Fan-out} = \text{minimum of } \left\{ \frac{I_{OH}}{I_{IH}}, \frac{I_{OL}}{I_{IL}} \right\} \quad \dots(5.16.3)$$



**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-3****Noise Margin.**

**Que 5.17.** What do you understand by noise margin of logic circuit ? Explain with an example.

OR

Define noise margin, Fan-in, Fan-out as characteristics of logic families. Implement NAND gate with CMOS.

AKTU 2022-23 (Sem-4), Marks 10

**Answer****Noise margin with example :**

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  $V_{OH} = 2.4$  V,  $V_{OL} = 0.4$  V,  $V_{IH} = 2$  V and  $V_{IL} = 0.8$  V. The noise introduced in the signal ( $V_{NH}$  or  $V_{NL}$ ) is shown in Fig. 5.17.1.

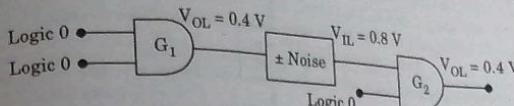


Fig. 5.17.1.

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  $V_{NL} = V_{IL} - V_{OL}$

7. Let the inputs of gate,  $G_1$  cause output as logic 1 in Fig. 5.17.2. This output acts as input for gate,  $G_2$ . Due to noise, actual input given to gate,  $G_2$  is

$$V_{NH} = V_{OH} - V_{IH}$$

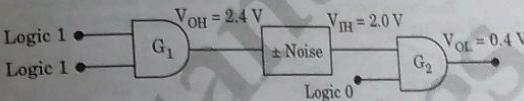


Fig. 5.17.2.

$V_{IL}$  in above equation that acts as input to gate  $G_2$ . Minimum high level noise level is

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

Fan-in and Fan-out : Refer Q. 5.16, Page 5-17G, Unit-5.

NAND gate with CMOS : Refer Q. 5.12, Page 5-11G, Unit-5.

**PART-4****RAM, ROM.**

**Que 5.18.** What is memory ? Write down the classification of semiconductor memories.

**Answer****Memory :**

1. Memories are made up of registers. Each register in the memory is one storage location also called memory location is identified by an address.

2. The number of storage locations can vary from a few in some memories to hundreds of thousands in others.



Fig. 5.18.1.

3. Each location can accommodate one or more bits. Generally, the total number of bits that a memory can store is its capacity.

#### Classification of semiconductor memories :

Fig. 5.18.2 shows an overview of semiconductor memory types.

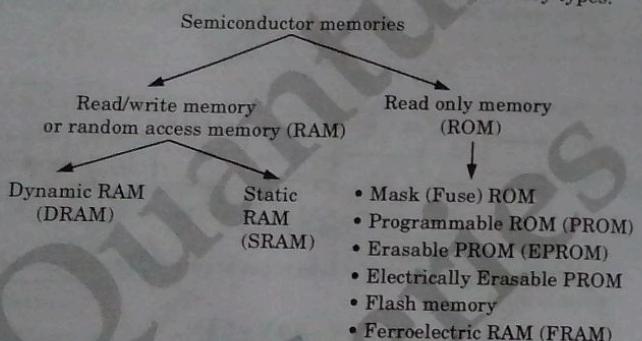


Fig. 5.18.2. Classification of semiconductor memories.

#### Que 5.19. What is RAM ? Distinguish between SRAM and DRAM.

#### Answer

##### Random access memory (RAM) :

- 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.
- The stored data is volatile, i.e., the stored data is lost when the power supply voltage is turned off.

#### Difference between SRAM and DRAM:

S. No.	Static RAM	Dynamic RAM
1.	Static RAM contains less memory cells per unit area.	Dynamic RAM contains more memory cells as compared to static RAM per unit area.
2.	It can be used as cache memory.	It can be used as main memory.
3.	Refreshing circuitry is not required.	Refreshing circuitry is required to maintain the charge on the capacitors after every few milliseconds.
4.	Static RAM consists of number of flip-flops. Each flip-flop stores one bit.	Dynamic RAM stores the data as charge on capacitor. It consists of MOSFET and capacitor for each cell.
5.	Faster.	Slower.

Que 5.20. Define the SRAM cell with working and circuit diagram

along with applications.

AKTU 2022-23 (Sem-3), Marks 10

#### Answer

**SRAM cell:** Static RAM cells are basically flip flops which can stay in a given state (i.e., store a bit) as long as power to the circuit is not interrupted.

##### Static RAM cell diagram :

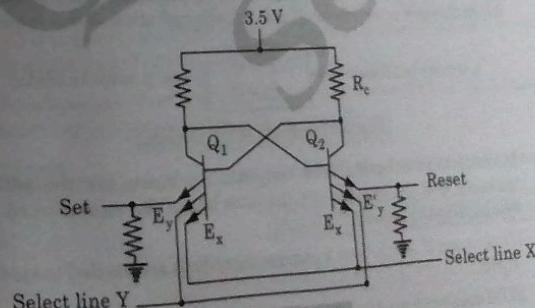


Fig. 5.20.1.

**Working :** In the given circuit multiple emitter transistors are used to form a flip flop with the set/reset facility and also the facility for select line connections. This cell is capable of storing only 1 bit.

- information  
es not permit  
eration.  
not lost even  
required.
- ata)
- address of  
word that is  
specify the  
e operation.
1. In this, data can be written any number of times. It can be reconstructed to the initial state even though it has been programmed previously.
  2. The data in the EPROM can be erased by exposing it to ultraviolet rays for a particular time period.
  3. After erasure, the EPROM returns to its initial state and can be reprogrammed to a new set of values.
- iv. **Electrically erasable PROMs (EEPROM) :**
1. This is similar to the EPROM, but the data can be erased with an electrical signal instead of ultraviolet light.
  2. The advantage of using EEPROM is that the device can be erased without removing it from its socket.

**Que 5.22. | Differentiate RAM and ROM. Explain various types of ROM.**

**AKTU 2021-22 (Sem-4), Marks 10**

**Answer****A. Difference between RAM and ROM :**

S.No.	Basis	RAM	ROM
1.	Data retention	RAM is a volatile memory which could store the data as long as the power supplied.	ROM is a non-volatile memory which could retain the data even when power is turned OFF.
2.	Speed	It is a high-speed memory.	It is much slower than the RAM.
3.	Size and capacity	Small size with less capacity.	Large size with higher capacity.
4.	Used as/in	CPU Cache, Primary memory.	Firmware, Micro-controllers.
5.	Accessibility	The data stored is easily accessible.	The data stored is not as easily accessible as in RAM
6.	Cost	Costlier	Cheaper than RAM.

**B. Types of ROM :** Refer Q. 5.21, Page 5-22G, Unit-5.

**Que 5.23.** Design  $8K \times 8$  RAM memory system, using  $1K \times 8$  memory ICs.

AKTU 2022-23 (Sem-4), Marks 10

**Answer****1. Number of chips required :**

Number of chips required

$$= \frac{\text{Desired RAM size}}{\text{Basic RAM size}}$$

$$= \frac{8K \times 8}{1K \times 8} = 8 \text{ chips}$$

**2. Address and data lines :**

Required size is  $8K \times 8 = 2^3 \times 2^{10} \times 8 = 2^{13} \times 8$

Number of address lines for  $8K \times 8 = 13$

Number of data lines for  $8K \times 8 = 8$

For  $1K \times 8 = 2^{10} \times 8$

Number of address lines for  $1K \times 8 = 10$

Data lines for  $1K \times 8 = 8$

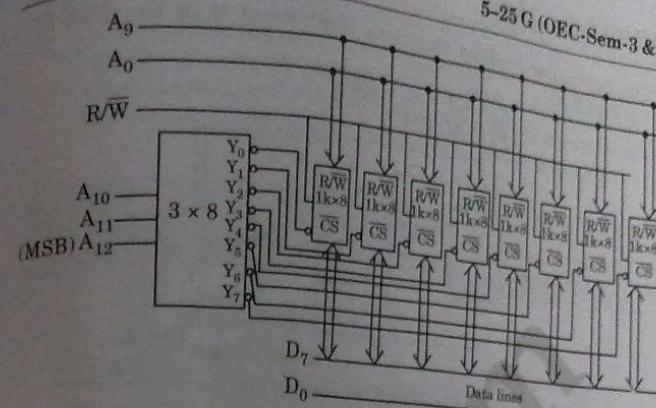


Fig. 5.23.1.

**PART-5**

PLA, PAL

**Que 5.24.** Write short notes on RAM and PLA

AKTU 2017-18 (Sem-3), Marks 07

OR

Elaborate the PLA (Programmable Logic Array) along with working and applications.

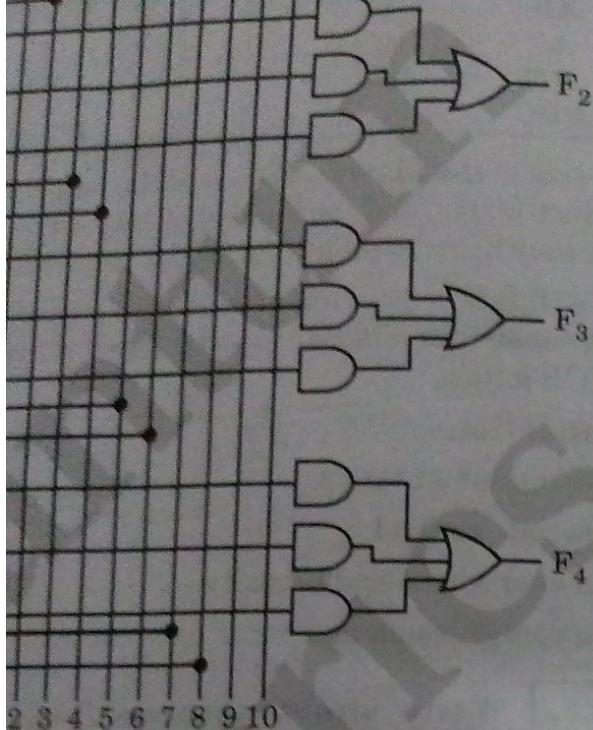
AKTU 2022-23 (Sem-3), Marks 10

**Answer**

**RAM :** Refer Q. 5.19, Page 5-20G, Unit-5.

**Programmable logic array (PLA) :**

- 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.
- 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.
- PLA can be implemented in several forms, i.e., NOR-NOR, NAND-NAND, NAND-NOR.
- The structure of PLA is shown in Fig. 5.24.1, and its internal logic with three inputs and two outputs is shown in Fig. 5.24.2.



four inputs, four outputs and a three wire AND-OR structure.

, the boolean function must be simplified to fit the situation with a PLA, a product term cannot have more OR gates. Therefore, each function can be realized without regarding common product terms.

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

### Full adder circuit using the PAL.

**L** : There are two functions used for the full adder :

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

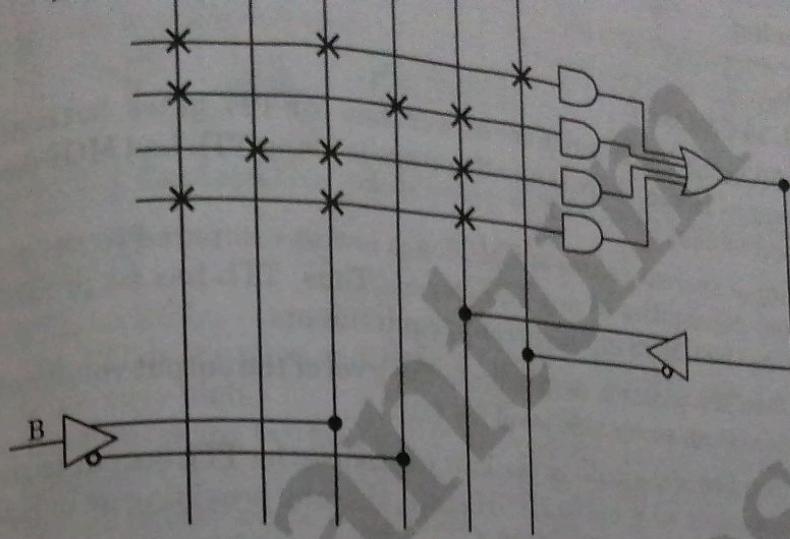


Fig. 5.26.1.

**Que 5.27.** Differentiate between PLA and PAL. Realize the full adder circuit using PAL.

AKTU 2018-19 (Sem-3), Marks 3.5

### Answer

#### 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.

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. 5.29.2 shows the ECL gate driving a TTL gate.

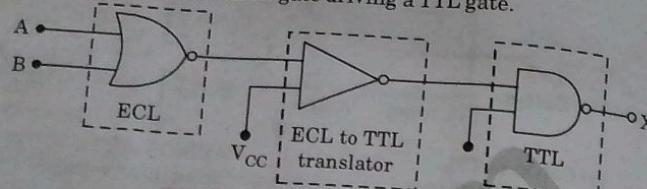


Fig. 5.29.2. ECL driving TTL.

### PART-7

Circuit Implementation Using ROM, PLA and PAL.

**Que 5.30.** Design a combinational circuit using a ROM that accepts a 3-bit number and generates an output binary number equal to the square of the input number.

**Answer**

Input			Output						
$B_2$	$B_1$	$B_0$	$D_5$	$D_4$	$D_3$	$D_2$	$D_1$	$D_0$	
0	0	0	0	0	0	0	0	0	
0	0	1	0	0	0	0	0	0	
0	1	0	0	0	0	0	0	1	
0	1	1	0	0	1	0	0	0	
1	0	0	0	1	0	0	0	1	
1	0	1	0	1	0	0	0	0	
1	1	0	1	0	0	1	0	1	
1	1	1	1	1	0	0	0	0	

For $D_0$				
$B_2$	$\bar{B}_1$	$\bar{B}_0$	$B_1$	$B_0$
0	1	1	1	3
4	1	5	1	7

$$D_0 = B_0$$

For $D_1$				
$B_2$	$\bar{B}_1$	$\bar{B}_0$	$B_1$	$B_0$
0	0	1	1	3
4	5	7	6	

$$D_1 = 0$$

For  $D_2$

$B_2$	$\bar{B}_1$	$\bar{B}_0$	$B_1$	$B_0$	$B_1\bar{B}_0$
$\bar{B}_2$	0	1	3	1	2
$B_2$	4	5	7	1	6

$$D_2 = B_1\bar{B}_0$$

For  $D_3$

$B_2$	$\bar{B}_1$	$\bar{B}_0$	$B_1$	$B_0$	$B_1\bar{B}_0$
$\bar{B}_2$	0	1	1	3	2
$B_2$	4	1	5	7	6

$$D_3 = \bar{B}_2B_1B_0 + B_2\bar{B}_1B_0$$

For  $D_4$

$B_2$	$\bar{B}_1$	$\bar{B}_0$	$B_1$	$B_0$	$B_1\bar{B}_0$
$\bar{B}_2$	0	1	3	2	
$B_2$	1	1	1	7	6

$$D_4 = B_2\bar{B}_1 + B_2B_0$$

For  $D_5$

$B_2$	$\bar{B}_1$	$\bar{B}_0$	$B_1$	$B_0$	$B_1\bar{B}_0$
$\bar{B}_2$	0	1	3	2	
$B_2$	4	5	1	1	6

$$D_5 = B_1B_2$$

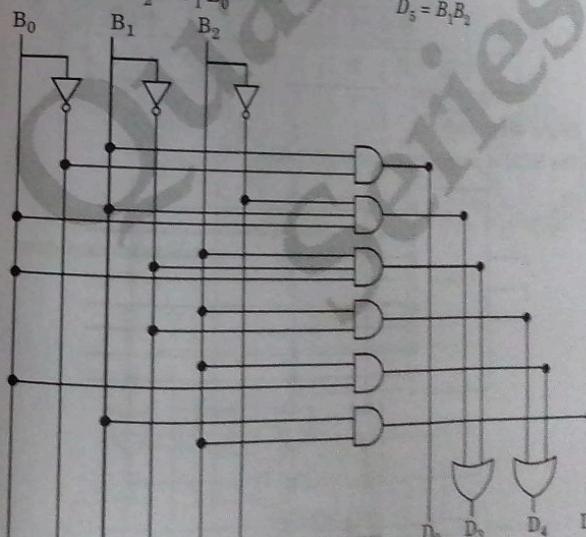


Fig. 5.30.1.

**Que 5.31.** Tabulate the truth table for  $8 \times 4$  ROM that implements the Boolean function :

**Que 5.33.** Explain PLA and PAL implement the given boolean function with a PLA.

$$Y_1(A, B, C) = \Sigma m(4, 5, 7) : Y_2(A, B, C) = \Sigma m(3, 5, 7)$$

AKTU 2021-22 (Sem-4), Marks 10

**Answer**

- A. Programmable Logic Array (PLA) : Refer Q. 5.24, Page 5-25G, Unit-5.
- B. Programmable Array Logic (PAL) : Refer Q. 5.25, Page 5-27G, Unit-5.
- C. Numerical : The procedure is same as Refer Q. 5.32, Page 5-35G, Unit-5.

**Implementation :**

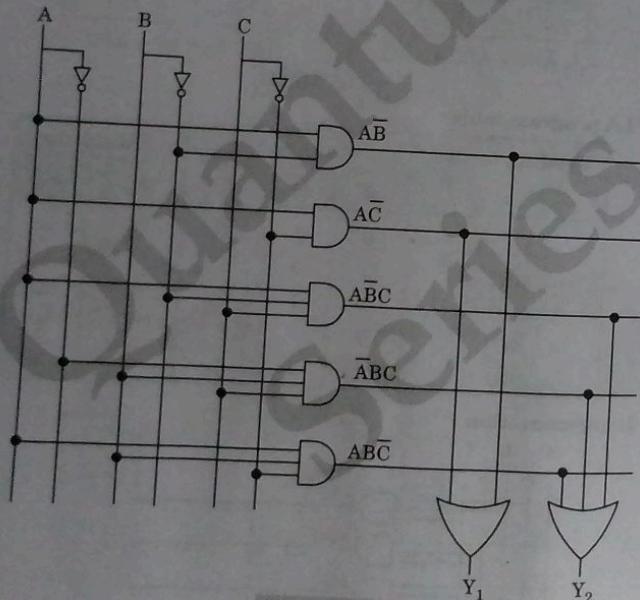


Fig. 5.33.1.

**Que 5.34.** Implement the following four boolean functions with a PAL.

$$W(A, B, C, D) = \Sigma(2, 12, 13)$$

$$X(A, B, C, D) = \Sigma(7, 8, 9, 10, 11, 12, 13, 14, 15)$$

$$Y(A, B, C, D) = \Sigma(0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15)$$

**Answer**

- Simplifying the four functions to a minimum number of terms results in the following boolean functions :

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

For X					
AB	CD	00	01	11	10
00	00	0	1	3	2
01	01	4	5	7	6
11	11	1	1	12	13
10	10	1	1	1	1

For Y					
AB	CD	00	01	11	10
00	00	1	0	1	1
01	01	1	1	1	1
11	11	4	5	7	6
10	10	12	13	15	14

For Z					
AB	CD	00	01	11	10
00	00	1	0	1	1
01	01	4	5	7	6
11	11	1	1	12	13
10	10	1	1	11	10

Fig. 5.34.1.

$$W = ABC' + A'B'CD'$$

$$X = A + BCD$$

$$Y = A'B + CD + B'D'$$

$$\begin{aligned} Z &= ABC'' + A'B'CD' + AC'D' + A'BC'D \\ &= W + AC'D' + A'BCD \end{aligned}$$

Input		Outputs
D	W	
—	—	$W = ABC' + A'B'CD'$
0	—	$X = A + BCD$
1	—	
1	—	
0	1	$Y = A'B + CD + B'D'$
0	0	
1	—	$Z = W + AC'D' + A'B'C'D$

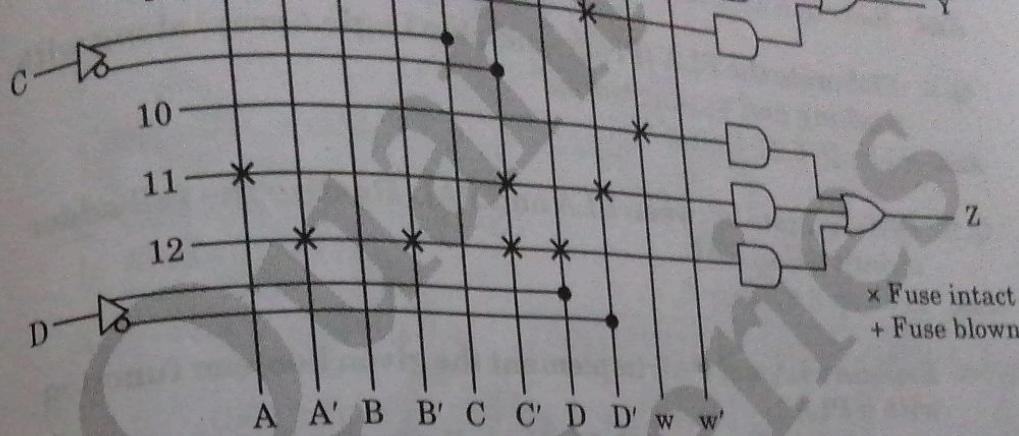


Fig. 5.34.2. Fuse map for PAL as specified in Table. 5.34.1.

### VERY IMPORTANT QUESTIONS

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

- Q. 1. Define the TTL (Transistor-Transistor Logic) logic family used for digital circuits.**

**Ans.** Refer Q. 5.5, Unit-5.

- Q. 2. Draw and explain the operation of a TTL NAND gate.**

**Ans.** Refer Q. 5.6, Unit-5.

**Q. 3.** Why ECL is better? Implement NAND gate with DTL and TTL.

**Ans.** Refer Q. 5.9, Unit-5.

**Q. 4.** Describe the circuit and performance of CMOS inverter and state the characteristics of CMOS.

**Ans.** Refer Q. 5.11, Unit-5.

**Q. 5.** Define noise margin, Fan-in, Fan-out as characteristics of logic families. Implement NAND gate with CMOS.

**Ans.** Refer Q. 5.17, Unit-5.

**Q. 6.** Define the SRAM cell with working and circuit diagram along with applications.

**Ans.** Refer Q. 5.20, Unit-5.

**Q. 7.** Differentiate RAM and ROM. Explain various types of ROM.

**Ans.** Refer Q. 5.22, Unit-5.

**Q. 8.** Elaborate the PLA (Programmable Logic Array) along with working and applications.

**Ans.** Refer Q. 5.24, Unit-5.

**Q. 9.** Differentiate between PLA and PAL. Realize the full adder circuit using PAL.

**Ans.** Refer Q. 5.27, Unit-5.

**Q. 10.** Explain PLA and PAL implement the given boolean function with a PLA.

$$Y_1(A, B, C) = \Sigma m(4, 5, 7) : Y_2(A, B, C) = \Sigma m(3, 5, 7)$$

**Ans.** Refer Q. 5.33, Unit-5.



1.1. What is (33)

**Ans.**

Now,

So, (33)

1.2. Convert

i. (562.13)

ii. (467.34)

**Ans.**

i. (562.13)

ii. (467.34)

1.3. Expl

# 1 UNIT

## Digital System and Binary Numbers (2 Marks Questions)

i.i. What is  $(33)_6 + (45)_6$

$$\begin{array}{r} (33)_6 = 3 \times 6^1 + 3 \times 6^0 = (21)_{10} \\ + (45)_6 = 4 \times 6^1 + 5 \times 6^0 = (29)_{10} \\ \hline \end{array}$$

Now,

$$\begin{array}{r} 6 \mid 50 \\ 6 \quad 8 \quad 2 \\ \hline 6 \quad 1 \quad 2 \\ \hline 0 \quad 1 \end{array}$$

$$\text{So, } (33)_6 + (45)_6 = (50)_{10} = (122)_6$$

i.ii. Convert the following :

i.  $(562.13)_7 = (?)_{10}$

**AKTU 2017-18 (Sem-3), Marks 02**

$$\begin{aligned} (562.13)_7 &= 5 \times 7^2 + 6 \times 7^1 + 2 \times 7^0 + 1 \times 7^{-1} + 3 \times 7^{-2} \\ &= (289.20)_{10} \\ \text{i. } (467.342)_8 &= 4 \times 8^2 + 6 \times 8^1 + 7 \times 8^0 + 3 \times 8^{-1} + 4 \times 8^{-2} + 2 \times 8^{-3} \\ &= (311.441)_{10} \end{aligned}$$

i.iii. Explain the signed binary number.

**AKTU 2022-23 (Sem-3), Marks 02**

Binary numbers that carry identification as to their polarity is called signed binary number. Plus (+) and minus (-) sign for positive and negative numbers respectively can be represented in digital format.

i.iv. Define cyclic codes.

When a bit pattern of two consecutive numbers differ by only one bit position, these codes are called cyclic codes.

**AKTU 2018-19 (Sem-3), Marks 02**

**SQ-2 G (OEC-Sem-3 & 4)**

**15. Write the advantage of Gray code over the straight binary number sequence.**

**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.

**16. Determine the value of base  $x$ , if  $(193)_x = (623)_8$ .**

**AKTU 2018-19 (Sem-3), Marks 02**

**Ans.**

$$(193)_x = (623)_8$$

Converting octal into decimal :  $6 \times 8^2 + 2 \times 8 + 3 = (403)_{10} = (623)_8$

$$(193)_x = 1 \times x^2 + 9 \times x^1 + 3 \times x^0 = (403)_{10}$$

$$x^2 + 9x + 3 = 403$$

$$x = 16 \quad \text{or} \quad x = -25$$

$\therefore$  Since, negative is not applicable, hence

$$x = 16$$

$$(193)_{16} = (623)_8$$

In H

**17. Identify the value of  $x$  in the expression  $(56.1A)_{16} = (x)_8$ .**

**AKTU 2021-22 (Sem-4), Marks 02**

**Ans.**

$$(56.1A)_{16} = (01010110.00011010)_2 = (126.064)_8$$

**18. Perform the subtraction  $(101101-100110)_2$  using 2's complement method.**

**AKTU 2021-22 (Sem-4), Marks 02**

**Ans.** Assuming,  $X = 101101$ ,  $Y = 100110$

1's complement :  $Y = 011001$

2's complement :  $Y = \underline{\underline{011010}}^{+1}$

$$\begin{array}{r} X = 101101 \\ Y = 011010 \\ \hline \text{Difference} = 110111 \end{array}$$

**19. Define the term universal gates and their applications.**

**AKTU 2022-23 (Sem-3), Marks 02**

**Ans.**

**A.**

Universal logic gate

1. Design
2. Implementation
3. Fault diagnosis
4. Application

**1.10. Define**

**Ans.** BCD digit 0 to 9

**B.** Con

**C.** A 5

**D.** A

- a.** Universal gates :- NOR gate and NOT gates are universal gates. These are called universal because it is possible to implement any logic gate or expression using only NOR gate or only NOT gates.
- b.** Applications of universal gates :-
- Simplification of circuit design.
  - Design optimisation.
  - Integrated circuit design.
  - Fault tolerant circuits.

**c.** Define BCD codes and convert  $(1538)_8$  into BCD number.

AKTU 2022-23 (Sem-3), Marks 02

- d.** BCD codes :- BCD is a coding scheme that represents decimal digits using a 4-bit binary code. In BCD, each decimal digit from 0 to 9 is represented by a unique 4-bit binary pattern.
- e.** Convert  $(42.56)_B$  into BCD number :

$$\begin{array}{r}
 \text{a. } S \quad D \quad I \\
 \downarrow \quad \downarrow \quad \downarrow \\
 \text{b. } 3 \times 10^0 = 3 \\
 \downarrow \\
 D = 10^2 = 10 = 10 = 208. \text{ [In hexadecimal } D = 13] \\
 \downarrow \\
 \text{c. } 5 \times 10^1 = 1250 \\
 \downarrow \\
 A = 10^3 = 10 = 10^2 = 40960 \\
 \downarrow \\
 (A, D, S)_{10} = (42.56)_B
 \end{array}$$

In BCD form  $42.56 = 0100\ 0010\ 0100\ 0101\ 0110$

**f.** Interpret the binary number  $(1011)_2$  into (i) Gray code (ii) Excess-3 code.

AKTU 2022-23 (Sem-4), Marks 02

**Ex-Grey code :-**



Hence, Gray code is 0110.

**g. Excess-3 code :-**

Excess-3 code of binary number  $(1011)_2 = 01000100$

**h.** Estimate  $(1011)_2 - (1101)_2$  using 1's and 2's complement method.

AKTU 2022-21 (Sem-4), Marks 02

**i.** Assuming  $X = 1011$ ,  $Y = 1101$

**SQ-4 G (OEC-Sem-3 & 4)**

$$\begin{array}{r} \text{1's complement : } Y = 0010 \\ +1 \\ \hline \end{array}$$

$$\begin{array}{r} \text{2's complement : } Y = 0011 \\ X = 1011 \\ Y = 0011 \\ \hline \text{Difference} & 1000 \end{array}$$

- 1.13. Implement the expression  $Y = ABC' + BD + E$  using NAND gate only.**

AKTU 2017-18 (Sem-3), Marks 02

**Ans.** There are five variables, A, B, C, D and E

$$Y = \overline{\overline{Y}} = \overline{\overline{ABC'} + BD + E} = \overline{\overline{ABC'}} \cdot \overline{\overline{BD}} \cdot \overline{\overline{E}}$$

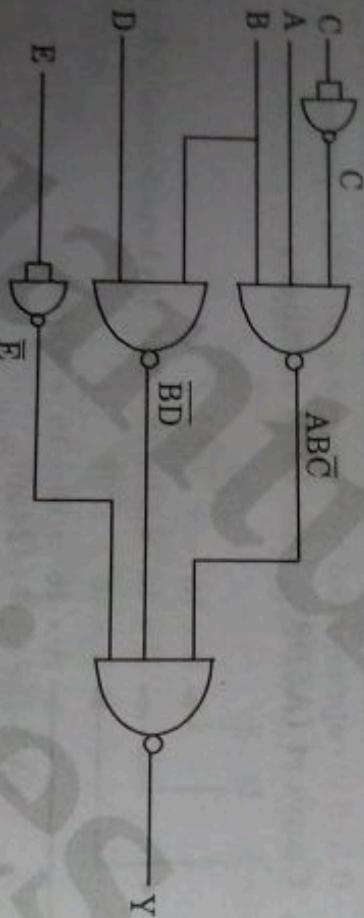
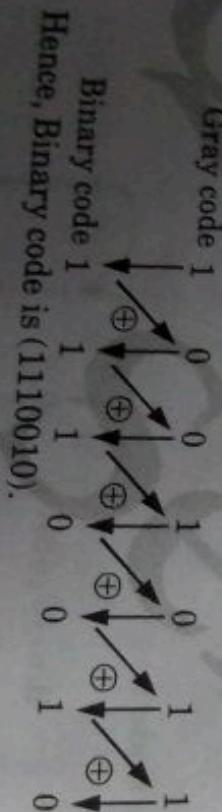


Fig. 1.13.1.

- 1.14. Convert gray code 1001011 to binary.**

**Ans.**



# 2

## Combinational Logic (2 Marks Questions)

**Ques.** Elaborate the term combinational circuits.

**AKTU 2022-23 (Sem-3), Marks 02**

- Ans.**
1. Combinational logic circuits consist of an interconnection of logic gates in which the output at any time depends upon the combination of input signals present at that instant only, and does not depend on any past conditions.
  2. In combinational circuit, the output does not depend on the past value of input or output. Hence combinational circuits do not require any memory.

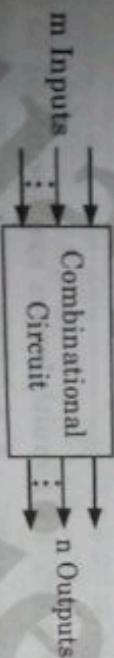


Fig. 2.1.1.

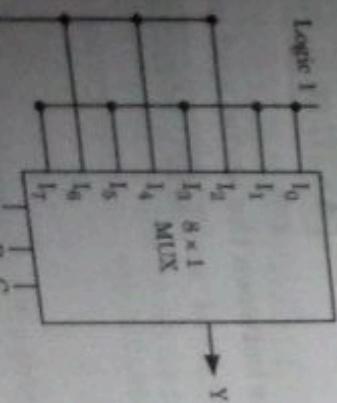
**Ques.** What is magnitude comparator?

**Ans.** A magnitude comparator is a combinational circuit designed primarily to compare the relative magnitude of the two binary numbers A and B.

**Ques.** Implement the following function using  $8 \times 1$  MUX.  

$$Y(A, B, C) = \sum m(0, 1, 3, 5, 7).$$

**Ans.**



Logic 0

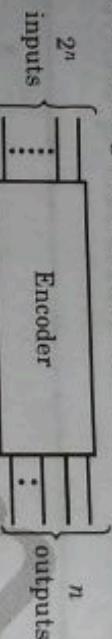
Fig. 2.3.1.

**SQ-6 G (OEC-Sem-3 & 4)**

2 Marks Questions

**2.4. What do you mean by encoder ?****Ans.**

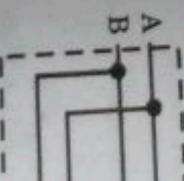
1. The encoder is another example of combinational circuit that performs the inverse operation of a decoder. It is designed to generate a different output code for an input which becomes active.
2. In an encoder, the number of outputs is less than the number of inputs. There are  $2^n$  input lines and  $n$  output lines.
3. The block diagram of an encoder is shown in Fig. 2.4.1.

**Fig. 2.4.1.****2.5. What is the difference between Multiplexer and Encoder ?****Ans.**

S.No.	Feature	Multiplexer	Encoder
1.	Input	Multiple data inputs.	Individual data inputs.
2.	Output	Single output.	Multiple output lines.
3.	Example	4-to-1 multiplexer.	4-to-2 encoder.

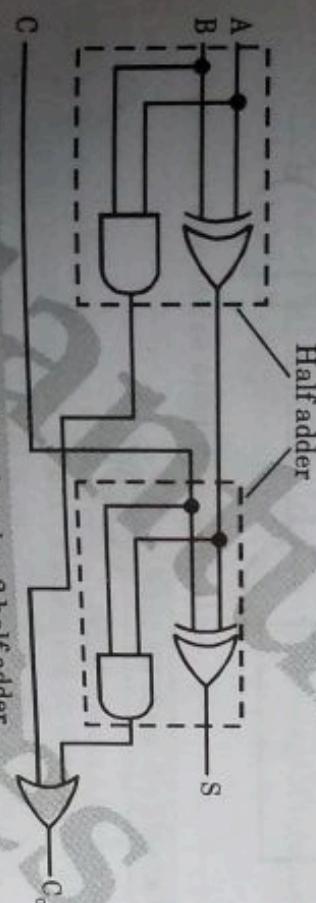
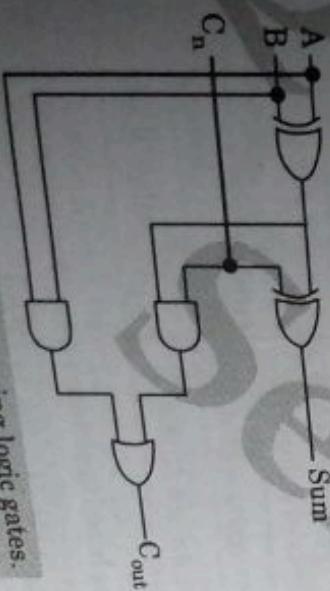
**2.6. How many  $4 \times 1$  multiplexers are required to implement****64  $\times$  1 multiplexer ? AKTU 2022-23 (Sem-4), Marks 02****Ans:** To implement a  $64 \times 1$  multiplexer using  $4 \times 1$  multiplexers, you will need a total of 21 multiplexers.**2.7. Define carry look ahead adder.****Ans:** It is an additional technique which speeds up the addition process by eliminating the problem due to inter-state carry delay.**2.8. Compare serial adder and parallel adder.****AKTU 2021-22 (Sem-4), Marks 02****OR****Differentiate between the serial and parallel adder.****AKTU 2022-23 (Sem-4), Marks 02****Ans.**

S. No.	Parameter
1.	Addition
2.	Type of
3.	Required

**Ans.****2.9. Draw****Ans.****Fig.****2.10. Cons****Ans.**

**Ans.**

S.No.	Parameters	Serial adder	Parallel adder
1.	Addition manner	It is used to add two binary numbers in serial form.	It is used to add two binary numbers in parallel form.
2.	Type of registers	A serial adder uses shift registers.	A parallel adder uses registers with parallel loads.
3.	Requirement	It requires a single full adder.	It requires multiple full adders.

**2.9. Draw a full adder using two half adders.****Ans.****Fig. 2.9.1.** Full adder circuit using 2 half adder.**2.10. Construct full adder using logic gates.****Ans.****Fig. 2.10.1.** Full adder using logic gates.**2.11. What is the role of subtractor in digital electronics ?****Ans.** Subtractor is an electronic logic circuit for calculating the difference between two binary numbers, the minuend and the number to be subtracted.**2.12. Draw the logic diagram of half subtractor.**

SQ-8 G (OEC-Sem-3 & 4)

2 Marks Questions

Ans.

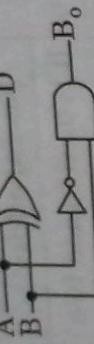


Fig. 2.12.1. Half subtractor.

2.13. Construct half subtractor using NAND gates.

Ans.

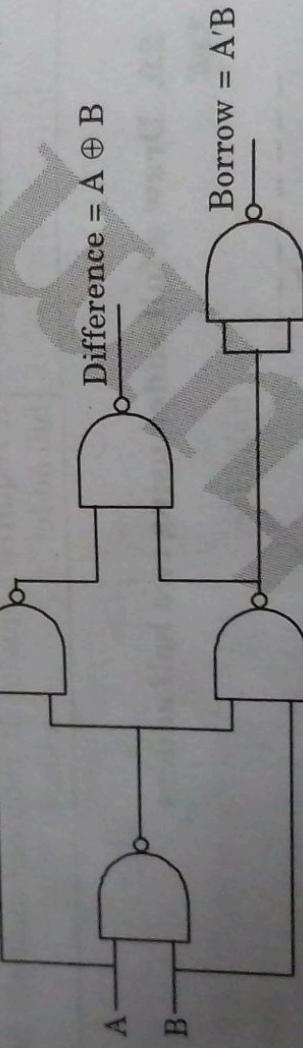


Fig. 2.13.1.



# 3

## Sequential Logic and its Applications (2 Marks Questions)

- 3.1. Illustrate the term sequential logic.

**AKTU 2022-23 (Sem-3), Marks 02**

**Ans.** Sequential logic refers to a type of digital logic circuit in which the output is not only determined by the current input but also by the previous states of the circuit. It involves the storage of information and the ability to remember past inputs or events.

- 3.2. Explain the term storage elements.

**AKTU 2022-23 (Sem-3), Marks 02**

**Ans.** A storage element in a digital circuit can maintain a binary state indefinitely (as long as power is delivered to the circuit), until directed by an input signal to switch states.

- 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. Write the difference between combinational and sequential circuits.

**AKTU 2017-18 (Sem-3), Marks 02**

**OR**

- What is difference between combinational and sequential circuits ?

**AKTU 2021-22 (Sem-4), Marks 02**

**OR**

- Differentiate between combinational and sequential circuits.

**AKTU 2022-23 (Sem-4), Marks 02**

- circuits.

## SQ-10 G (OEC-Sem-3 &amp; 4)

2 Marks Questions

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.

- 3.5. Write the excitation table and characteristic equation of JK flip-flop.

AKTU 2017-18 (Sem-3), Marks 02

- Ans.** Excitation table : The excitation table of JK flip-flops is as follows :

Present State ( $Q_n$ )	Next State ( $Q_{n+1}$ )	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

- B. Characteristic equation of JK flip-flop :  $Q_{n+1} = J\bar{Q}_n + \bar{K}Q_n$ .

- 3.6. What is race around condition ?

AKTU 2018-19 (Sem-3), Marks 02

AKTU 2017-18 (Sem-3), 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.

- 3.7. If in an edge triggered JK flip flop,  $J = 1$ ,  $K = 1$  and  $Q = 1$ , when the clock pulse goes high, what would be the next state of Q.

AKTU 2021-22 (Sem-4), Marks 02

- Ans.** The next state of Q is zero (toggle flip-flop output).

- 3.8. The content of 4 bit register is initially 1101. The register is shifted six times to right with the serial input being 101101. What is the content of the register after sixth shift ?

AKTU 2021-22 (Sem-4), Marks 02

Ans.

CLK	D
Initially	↓
1 <sup>st</sup>	↓
2 <sup>nd</sup>	↓
3 <sup>rd</sup>	↓
4 <sup>th</sup>	↓
5 <sup>th</sup>	↓
6 <sup>th</sup>	↓

- 3.9. Define race

- Ans.** In JK flip flop, if the output of flip flop changes before the next clock pulse, then Q output is said to have race around.

- 3.10. What is the table ?

Ans.

S.No.	Aspect
1.	Purpose
2.	Input Condition

- 3.11. What is r

- Ans.** A number to startin

- 3.12. How many counter

Ans.

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

3.9. Define race around condition in JK flip flop.

AKTU 2022-23 (Sem-4), Marks 02

**Ans.** In JK flip flop, if  $J = K = 1$  and if Clk = 1 for a long period of time, then Q output will toggle as long as Clk is high, which makes the output of flip flop unstable or uncertain. This problem is called race around condition in JK flip flop.

3.10. What is the difference between characteristic and excitation table ?

AKTU 2022-23 (Sem-4), Marks 02

-  $\bar{K}Q_n$

Ans.

S.No.	Aspect	Characteristic table	Excitation table
1.	Purpose	Describes the output behavior given the present state and input conditions.	Specifies the input conditions needed to achieve specific state transitions.
2.	Input Conditions	Current state and inputs.	Present state and desired next state.

3.11. What is modulus of a counter ?

AKTU 2018-19 (Sem-3), Marks 02

**Ans.** A number of states through which counter passes before returning to starting state or initial state is called modulus counter.

3.12. How many flip-flops are required to design Mod-5 ring counter and Mod-5 Johnson counter ?

AKTU 2018-19 (Sem-3), Marks 02

**SQ-12 G (OEC-Sem-3 & 4)**

**Ans:** Mod-5 Ring counter - 5 Flip-flop.  
Mod-5 Johnson counter - 3 Flip-flop.

**3.13. Define shift registers.**

**Ans:** 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.

**3.14. 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)

**3.15. Define the use of clock in digital circuits.**

**AKTU 2022-23 (Sem-3), Marks 02**

**Ans.**

In digital circuits, a clock is a timing signal that provides synchronization and coordination for the operation of various components. It acts as a reference signal, controlling the timing of events and ensuring that operations occur in a precise and synchronized manner. The clock signal helps maintain the integrity and reliability of digital systems.



# 4

## UNIT

### Synchronous and Asynchronous Sequential Circuits (2 Marks Questions)

- 4.1. Differentiate synchronous and asynchronous sequential circuits.

AKTU 2021-22 (Sem-4), Marks 02

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.

- 4.2. Explain the term synchronous circuits.

AKTU 2022-23 (Sem-3), Marks 02

**Ans.** Synchronous circuits are a type of digital circuit where the state changes occur only at specific times synchronized by a common clock signal. In synchronous circuits, all flip-flops or registers within the circuit are triggered by the same clock signal, ensuring that all state transitions and operations occur simultaneously and in a coordinated manner.

- 4.3. What are the types of asynchronous sequential circuits ?

**Ans.** There are two types of asynchronous circuits and

1. Fundamental mode circuits
2. Pulse mode circuits.

- 4.4. What do you mean by fundamental mode operation ?

**Ans.** In the fundamental mode of operation, the external inputs can be changed at any time and a transition from one state to another state occurs only when change in input occur.

**4.5. Define flow table.**

**Ans.** During the design of asynchronous sequential circuits, it is more convenient to name the states by letter symbols without making specific reference to their binary values. Such a table is called a flow table.

**4.6. Define critical race and non-critical race conditions.**

**AKTU 2021-22 (Sem-4), Marks 02**

**Ans.**

- 1. Critical race:** 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.
- 2. Non-critical race:** 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.

**4.7. Define the term hazards.**

**Ans.** Hazards are unwanted switching transient that may appear at the output of a circuit because different paths exhibit different propagation delays.

**4.8. Give the classification of hazards.**

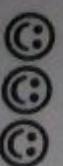
1. Static hazard : (i) Static-1 hazard (ii) Static-0 hazard
2. Dynamic hazard
3. Essential hazard.

**4.9. What is the method of prevention of hazard in logic gates?**

**Ans.** Static and dynamic hazards can be prevented by adding extra gates in the circuit as the redundant term.

**4.10. Define state assignment.**

**Ans.** In order to design a sequential circuit with physical components, it is necessary to assign unique coded binary values to the states is called state assignment.



# 5

## Programmable Logic Devices (2 Marks Questions)

more  
aking  
lled a  
erent  
iable  
ches  
nges,

es ?  
ates  
t the  
rent  
s, it  
s is

5.1. Illustrate the use of logic families in digital circuits.

**AKTU 2022-23 (Sem-3), Marks 02**

**Ans:** Logic families are used in the design and implementation of digital circuits. Each logic family has its own characteristics, such as voltage levels, power consumption, speed, noise immunity, and cost. The choice of logic family depends on the specific requirements of the digital circuit.

5.2. Write the advantage and disadvantages of TTL and CMOS logic family.

**AKTU 2021-22 (Sem-4), Marks 02**

**Ans:** A. Advantages and disadvantages of TTL:

1. **Advantages :**

- It has strong drive capability.
- It has noise immunity better than ECL but lesser than CMOS.

2. **Disadvantages :** Consumes more power compares to CMOS family and hence it is not suitable for battery powered devices.

B. Advantages and disadvantages of CMOS:

1. **Advantages :**

- Extremely large fan-out capability.
- Lowest power dissipation of all gates.
- Very high noise-immunity and noise-margin typically.

2. **Disadvantages :**

- Increased cost due to additional processing steps.
- The main disadvantage of the CMOS logic family is their slow speed of operation.

5.3. Explain fan-in and fan-out in logic families.

**AKTU 2021-22 (Sem-4), Marks 02**

OR

**SQ-16 G (OEC-Sem-3 & 4)**

**Elaborate the term Fan-in in digital circuits.**

**AKTU 2022-23 (Sem-3), Marks 02**

**OR**

**What do you mean by fan-out and fan-in ?**

**AKTU 2018-19 (Sem-3), 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.

$$\text{Fan-out} = \text{Minimum of } \left\{ \frac{I_{OH}}{I_H}, \frac{I_{OL}}{I_L} \right\}$$

**Fan-in :** The fan-in of a logic gate refers to the number of inputs that the gate is designed to handle.

**5.4. Define term propagation delay.**

**AKTU 2022-23 (Sem-4), Marks 02**

**Ans.** 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.

**5.5. 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.

**5.6. 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.

**5.7. How many address lines are needed to represent 8K meaning ?**

**AKTU 2022-23 (Sem-4), Marks 02**

**Ans.** To represent 8K memory locations, we need 13 address lines. This is because 8K is equal to 8192, and  $2^{13} = 8192$ .

**5.8. 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.

## Digital Electronics

SQ-17 G(OEC.Sem-3 & 4)

**5.9. 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.

**5.10. How many address lines and data I/O lines are required for a  $16K \times 12$  memory?**

$$16K \times 12 = 2^4 \times 2^{10} \times 12$$

Address lines = 14

Data lines = 12

**5.11. Differentiate between RAM and ROM.**

**Ans.**

S.No.	RAM (Random access memory)	ROM (Read only memory)
1.	It is a read/write memory.	It is a read only memory.
2.	Data stored in RAM is volatile.	ROMs are non-volatile.

**5.12. Give the difference between PAL and PLA.**

AKTU 2022-23 (Sem-4), Marks 02

**Ans.**

	PAL	PLA
S.No.	Feature	
1.	Flexibility	Less flexible
2.	Speed	Faster



transferred  
needed  
oring a

nt 8K  
ks 02  
This