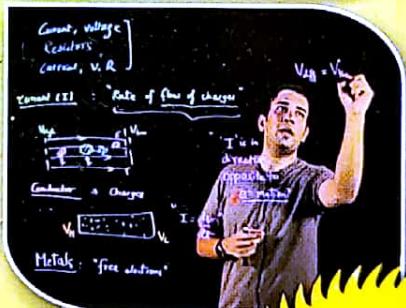
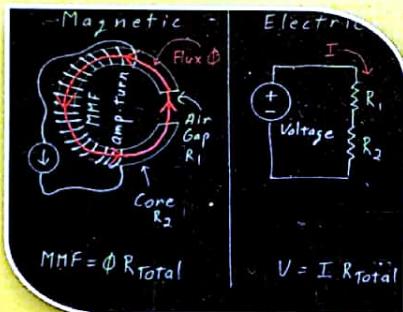


QUANTUM Series

Semester-4 Electrical & Electronics Engineering

Digital Electronics



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

Session
2019-20
Even Semester

Includes solution of following AKTU Question Papers:

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

PUBLISHED BY:

**Apram Singh
Quantum Page Pvt. Ltd.**
Plot No. 59/2/7, Site - 4, Industrial Area,
Sahibabad, Ghaziabad-201 010

Phone : 0120-4160479

Email : pagequantum@gmail.com Website : www.quantumpage.co.in
Delhi Office : 1/6590, East Rohtas Nagar, Sahadara, Delhi-110032

© ALL RIGHTS RESERVED

No part of this publication may be reproduced or transmitted,
in any form or by any means, without permission.

Information contained in this work is derived from sources
believed to be reliable. Every effort has been made to ensure
accuracy, however neither the publisher nor the authors
guarantee the accuracy or completeness of any information
published herein, and neither the publisher nor the authors
shall be responsible for any errors, omissions, or damages
arising out of use of this information.

Digital Electronics (EN : Sem-4)1st Edition : 2009-102nd Edition : 2010-113rd Edition : 2011-124th Edition : 2012-135th Edition : 2013-146th Edition : 2014-157th Edition : 2015-168th Edition : 2016-179th Edition : 2017-1810th Edition : 2018-1911th Edition : 2019-20**Price: Rs. 125/- only****Printed at : Mayank Enterprises, Delhi-110093.****CONTENTS****KEE 401 : Digital Electronics****UNIT-1 : DIGITAL SYSTEM & BINARY NUMBERS (1-1 A to 1-34 A)**

Number System and its arithmetic, Signed binary numbers, Binary codes, Cyclic codes, Hamming Code, the map method up to five variable, Don't care conditions, POS simplification, NAND and NOR implementation, Quine McClusky method (Tabular method).

UNIT-2 : COMBINATIONAL LOGIC (2-1 A to 2-33 A)

Combinational Circuits: Analysis Procedure, Design procedure, Binary adder-subtractor, Decimal adder, Binary multiplier, Magnitude comparator, Multiplexers, Demultiplexers, Decoders, Encoders.

UNIT-3 : SEQUENTIAL LOGIC (3-1 A to 3-37 A)

Storage elements: latches & flip flops, Characteristic Equations of Flip Flops, Flip Flop Conversion, Shift Registers, Ripple Counters, Synchronous Counters, Other Counters: Johnson & Ring Counter.

UNIT-4 : SYNCHRONOUS SEQUENTIAL CIRCUITS (4-1 A to 4-51 A)

Analysis of clocked sequential circuits with state machine designing, State reduction and assignments, Design procedure. Analysis procedure of Asynchronous sequential circuits, circuit with latches, design procedure, Reduction of state and flow table, Race-free state assignment, Hazards.

UNIT-5 : MEMORY & PLA (5-1 A to 5-45 A)

Digital Logic Families: DTL, DCTL, TTL, ECL & CMOS etc., Fan Out, Fan in, Noise Margin; RAM, ROM, PLA, PAL; Circuits of Logic Families, Interfacing of Digital Logic Families, Circuit Implementation using ROM, PLA and PAL; CPLD and FPGA.

SHORT QUESTIONS (SQ-1 A to SQ-15 A)**SOLVED PAPERS (2014-15 TO 2018-19) (SP-1 A to SP-26 A)**

1

UNIT

Digital System and Binary Numbers

Part-1 (1-2A to 1-16A)

- Number System and Its Arithmetic
- Signed Binary Numbers
- Binary Codes
- Cyclic Codes
- Hamming Code

A. Concept Outline : Part-1 1-2A
B. Long and Medium Answer Type Questions 1-2A

Part-2 (1-16A to 1-34A)

- The Map Method upto Five Variables
- Don't Care Conditions
- POS Simplification
- NAND and NOR Implementation
- Quine Mc-Cluskey Method (Tabular Method)

A. Concept Outline : Part-2 1-16A
B. Long and Medium Answer Type Questions 1-17A

1-1A (EC/CS/IT-Sem-3)

1-2 A (EC/CS/IT-Sem-3)

Digital System & Binary Numbers

PART-1

Number System and Its Arithmetic, Signed Binary Numbers, Binary Codes, Cyclic Codes, Hamming Code.

CONCEPT OUTLINE : PART-1

- The number system are as follows :
 - i. Decimal number system.
 - ii. Binary number system.
 - iii. Octal number system.
 - iv. Hexadecimal number system.
- **Complements** : These are used in digital computers to simplify the subtraction operation and for logical manipulation.
- One of the most common error correcting codes is the Hamming code.
- A code which has one bit change in successive code words is called cyclic code.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 1.1. Define number system and also define signed and unsigned binary number ?

Answer

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.

The classification of number system are 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.
5. **Complements** : These are used in digital computers to simplify the subtraction operation and for logical manipulation.

6. **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.
7. **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 decimal numbers to their binary equivalents :

- i. $(83)_{10}$
 ii. $(79.515)_{10}$
 iii. $(109.125)_{10}$
 iv. $(225.225)_{10}$

OR

Convert the decimal number 225.225 to binary number.

Answer

- i. $(83)_{10}$

2	83
2	41 1
2	20 1
2	10 0
2	5 0
2	2 1
1	0

$$\text{Thus, } (83)_{10} = (1010011)_2$$

- ii. $(79.515)_{10}$

2	79
2	39 1
2	19 1
2	9 1
2	4 1
2	2 0
1	0

$$\begin{array}{l} 0.515 \times 2 = 1.030 \rightarrow 1 \\ 0.030 \times 2 = 0.060 \rightarrow 0 \\ 0.060 \times 2 = 0.120 \rightarrow 0 \\ 0.120 \times 2 = 0.240 \rightarrow 0 \\ 0.240 \times 2 = 0.480 \rightarrow 0 \\ 0.480 \times 2 = 0.960 \rightarrow 1 \\ 0.960 \times 2 = 1.920 \end{array}$$

Thus,

$$\begin{aligned} (79)_{10} &= (1001111)_2 \\ (79.515)_{10} &= (1001111.100001)_2 \end{aligned}$$

1-4 A (EC/CS/IT-Sem-3)

III. $(109.125)_{10}$

2	109
2	54 1
2	27 0
2	13 1
2	6 1
2	3 0
1	1

$$\begin{array}{l} 0.125 \times 2 = 0.250 \rightarrow 0 \\ 0.250 \times 2 = 0.500 \rightarrow 0 \\ 0.500 \times 2 = 1.000 \rightarrow 1 \\ 0.000 \times 2 = 0.000 \end{array}$$

$$\begin{aligned} \text{Thus, } (109)_{10} &= (1101101)_2 \\ (109.125)_{10} &= (1101101.00010)_2 \end{aligned}$$

iv. $(225.225)_{10}$

2	225
2	112 1
2	56 0
2	28 0
2	14 0
2	7 0
2	3 1
2	1 1
0	1

$$\begin{array}{l} 0.225 \times 2 = 0.450 \rightarrow 0 \\ 0.45 \times 2 = 0.90 \rightarrow 0 \\ 0.90 \times 2 = 1.80 \rightarrow 1 \\ 0.80 \times 2 = 1.60 \rightarrow 1 \\ 0.60 \times 2 = 1.20 \rightarrow 0 \\ 0.20 \times 2 = 0.40 \rightarrow 0 \\ 0.40 \times 2 = 0.80 \rightarrow 1 \\ 0.80 \times 2 = 1.6 \rightarrow 1 \\ 0.6 \times 2 = 1.2 \end{array}$$

$$(225.225)_{10} \rightarrow (11100001.001110011)_2$$

Que 1.3. Convert the following :

- i. $(62.7)_8 = (\)_{16} = (\)_2$
 ii. $(BC6)_{16} = (\)_{10} = (\)_2$

AKTU 2012-13, Marks 05

Answer

i. $(62.7)_8 = (\)_{16} = (\)_2$

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

$$\therefore (62.7)_8 = (32.E)_{16} = (00110010.1110)_2$$

ii. $(BC6)_{16} = (\)_{10} = (\)_2$

$$\begin{aligned} (BC6)_{16} &= 11 \times 16^2 + 12 \times 16^1 + 6 \times 16^0 = (3014)_{10} \\ (3014)_{10} &= (101111000110)_2 \end{aligned}$$

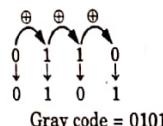
Que 1.4. Represent the decimal number 6 in (i) excess-3 code, (ii) BCD code, (iii) Gray code, (iv) 8421 code and (v) 2421 codes.

AKTU 2012-13, Marks 05

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: $(6)_2 = 0110$ 

$$\text{Gray code} = 0101$$

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

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

v. 2421 code: $(6)_{2,4,2,1} = 1100$

Que 1.5. Convert the following numbers into desired base :

$$\text{i. } (A6BF5)_{16} = (?)_2 = (?)_{\text{Gray}}$$

$$\text{ii. } (17 - 135) \text{ using 2's complement}$$

AKTU 2014-15, Marks 3.5

Answer

$$\text{i. } (A6BF5)_{16} = (1010011010111110101)_2 \\ = (1111010101111000001111)_{\text{Gray}}$$

$$\text{ii. } (17 - 135) \text{ in binary form} = (10001)_2$$

$$135 \text{ in binary form} = (10000111)_2$$

$$2\text{'s complement of } 135 = 01111001$$

$$\text{Now, adding } 17 \text{ in binary form} = + 00010001$$

As there is no carry, the answer is negative and is in the 2's complement form. The answer is,

$$-(2\text{'s complement of } 10001010) \\ = -(01110110)_2$$

Thus the answer is $-(01110110)_2$, i.e., $(-118)_{10}$.

Que 1.6. Add the following numbers :

$$\text{i. } (ABC)_{16} + (CDE)_{16}$$

$$\text{ii. } (77)_8 + (107)_8$$

AKTU 2011-12, Marks 5

Answer

$$\text{i. } (ABC)_{16} = 1010 \ 1011 \ 1100 \text{ (in binary)}$$

$$(CDE)_{16} = 1100 \ 1101 \ 1110$$

$$(ABC)_{16} + (CDE)_{16} = \underline{\underline{10111 \cdot 1001 \ 1010}}$$

$$= (179A)_{16}$$

$$\text{ii. } (77)_{16} = 000 \ 111 \ 111 \text{ (in binary)}$$

$$(107)_{16} = 001 \ 000 \ 111$$

$$(77)_{16} + (107)_{16} = \underline{\underline{010 \ 000 \ 111}}$$

$$= (206)_8$$

Que 1.7. Represent the unsigned decimal number 965 and 672 in BCD and then show the steps necessary to find their sum.

AKTU 2012-13, Marks 05

Answer

1. The BCD representations of the given number are as follows :

965	1001	0110	0101
672	0110	0111	0010
FD7	1111	1101	0111

2. Since, F > 9 and D > 9, i.e., invalid BCD, therefore, adding 6 (0110)₂ in F and D, we get

1111	1101	0111
0110	0110	0000
0001	0110	0011
1	6	3

Valid BCD
Equivalent decimal

Hence, the sum is 1637.

Que 1.8. The solution to the quadratic equation $x^2 - 11x + 22 = 0$ are $x = 3$ and $x = 6$. What is the base of the number system used ?

AKTU 2012-13, 2013-14; Marks 05

Answer1. Suppose the base of the number is b . The given quadratic equation is $(x^2 - 11x + 22)_b = 0$... (1.8.1)

2. The solution of quadratic equation is,

$$x = 3 \text{ and } x = 6$$

3. The quadratic equation formed with these roots is

$$(x - 3)(x - 6) = x^2 - (6 + 3)x + (6 \times 3) \quad \dots(1.8.2)$$

4. Comparing eq. (1.8.2) with the given quadratic eq. (1.8.1)

$$(9)_{10} = (11)_b$$

$$\begin{aligned} b^1 \times 1 + b^0 \times 1 &= 9 \\ b + 1 &= 9, \quad b = 8 \\ \text{also, } (18)_{10} &= (22)_b \end{aligned}$$

$$\begin{aligned} b^1 \times 2 + b^0 \times 2 &= 18 \\ 2(b + 1) &= 18, \quad b = 8 \end{aligned}$$

Hence, the base of the number system is 8.

Que 1.9. Perform the following subtraction using 2's complement

method :

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

AKTU 2011-12, Marks 05

Answer

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

$$\begin{array}{r} & +1 \\ Y = 10110 & \\ \hline X = 01000 & \\ - Y = 10111 & \\ \hline \text{Difference} = 11111 & \end{array}$$

There is no end carry,

$$\begin{aligned} X - Y &= -(2\text{'s complement of } 11111) \\ &= -(00001) \end{aligned}$$

- ii. Assuming, $X = 0011.1001, Y = 0001.1110$

1's complement : $Y = 1110.0001$

$$\begin{array}{r} & +1 \\ Y = 1110.0001 & \\ \hline X = 0011.1001 & \\ - Y = 1110.0010 & \\ \hline \text{Difference} = \boxed{1} 0001.1011 & \end{array}$$

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

Que 1.10. Describe the binary codes. Show the classification of binary codes in tabular format.

Answer

1. Code is the representation of group of symbols, words, or letters. As the digital data is used as group of binary numbers so, we call it as the binary codes.
2. These binary codes are used for the designing and analysis of digital circuit, computer applications, in digital communication. The codes are

classified into certain following categories :

- i. Weighted codes
- ii. Non-weighted codes
- iii. Reflective codes
- iv. Sequential codes
- v. Alphanumeric codes
- vi. Error detecting and correcting codes.
3. Since, all these codes use only 0 and 1, so it is easier to implement. The binary codes can also be used for representing the numbers as well as the alphanumeric letters.
4. The classification of codes can be composed in tabular form which is as follows :

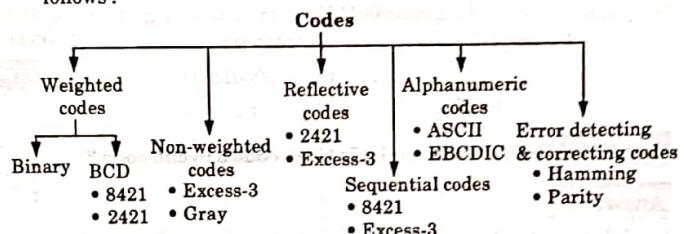


Fig. 1.10.1.

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 1.11. Represent decimal number “-13” in all three methods of negative binary number representation using eight bits.

AKTU 2011-12, Marks 05

AKTU 2013-14, Marks 2.5

Answer

Given, number is $(-13)_{10}$

i. Sign magnitude:

$$(-13)_{10} = (10001101)_2$$

ii. Using 2's complement:

$$(13)_{10} = (00001101)_2$$

$$2\text{'s complement} = (2)^8 - (00001101)_2$$

$$= (100000000)_2 - (00001101)_2$$

$$(-13)_{10} = (11110011)_2$$

iii. Using 1's complement:

$$(13)_{10} = (00001101)_2$$

$$1\text{'s complement} = (2)^8 - 1 - (00001101)_2$$

$$= (11111111)_2 - (00001101)_2$$

$$(-13)_{10} = (11110010)_2$$

Que 1.12. Define cyclic code. Is Gray code a cyclic code?

Answer

- When a bit pattern of two consecutive numbers differ in only one bit position, these codes are called cyclic codes.
- Cyclic code forms a subclass of linear block codes.
- An (n, k) linear block code C is called a cyclic code if it satisfies the following property:

If an n -tuple

$$v = (v_0, v_1, \dots, v_{n-1}) \quad \dots(1.12.1)$$

is a code-vector of C , then the n -tuple

$$v^{(1)} = (v_{n-1}, v_0, v_1, \dots, v_{n-2})$$

which is obtained by shifting v cyclically one place to the right, is also a code-vector of C .

$$v^{(i)} = (v_{n-i}, v_{n-i+1}, \dots, v_0, v_1, \dots, v_{n-i-1}) \quad \dots(1.12.2)$$

- The code word v can be represented by a code polynomial as

$$V(x) = v_0 + v_1x + v_2x^2 + \dots + v_{n-1}x^{n-1} \quad \dots(1.12.3)$$

Theorem: If $g(x)$ is a polynomial of the degree $(n-k)$ and is a factor of $x^n + 1$, then, $g(x)$ generates an (n, k) cyclic code in which the code polynomial $V(x)$ for a data vector $D = (d_0, d_1, d_2, \dots, d_{k-1})$ is generated by

$$V(x) = D(x)g(x) \quad \dots(1.12.4)$$

An example of cyclic code:



It can be seen that the code 1011, 1101, 1110, 0111 is obtained by a cyclic shift of n -tuple 1011 ($n = 4$). The code obtained by rearranging the four words is also a cyclic code. Thus 1011, 1101, 1110, 0111 are also cyclic codes. Yes, Gray code is a cyclic code as here consecutive bit changes only one bit position.

Que 1.13. Consider a $(7, 4)$ cyclic code. The generator polynomial for this code is given as $G(x) = 1 + x + x^3$. Find all the code words of this code.

Answer

- Given, $G(x) = 1 + x + x^3$
- t^{th} row of generator matrix is given by

$$x^{n-t} + R_t(x) = Q_t(x) G_t(x)$$

Here $n = 7$, $k = 4$ and $m = n - k = 7 - 4$, $m = 3$

$$x^{7-t} + R_t(x) = Q_t(x) G_t(x)$$

- For $t = 1$

$$x^6 + R_1(x) = Q_1(x) G_1(x)$$

$$G_1(x) = x^3 + x + 1$$

$$\frac{x^3 + x + 1}{x^3 + x + 1}$$

$$\overline{x^6 + x^4 + x^3} \quad [\text{By using Modulo-2 subtraction}]$$

$$\frac{x^4 + x^3}{x^4 + x^2 + x}$$

$$\frac{x^4 + x^2 + x}{x^3 + x^2 + x}$$

$$\frac{x^3 + x^2 + x}{x^3 + 0 + x + 1}$$

$$\frac{x^3 + 0 + x + 1}{x^2 + 1}$$

$$\frac{x^2 + 1}{x^2 + 1}$$

$$\frac{x^2 + 1}{x^2 + 1}$$

$$Q_1(x) = x^3 + x + 1$$

$$R_1(x) = x^2 + 1$$

$$x^6 + x^2 + 1 = (x^3 + x + 1)(x^3 + x + 1)$$

- 1st row polynomial

$$\Rightarrow x^6 + x^2 + 1$$

- 2nd row polynomial

$$\Rightarrow x^5 + x^2 + x + 1$$

- 3rd row polynomial

$$\Rightarrow x^4 + x^2 + x$$

- 4th row polynomial

$$\Rightarrow x^3 + x + 1$$

Hence $G = \begin{bmatrix} x^6 & x^5 & x^4 & x^3 & x^2 & x^1 & x^0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$

5. $C = DG$

$D = [0 \ 0 \ 0 \ 0]$

$C_1 = D_1 G = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$

$C_2 = [0 \ 0 \ 0 \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} = [0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1]$

Similarly calculating for other messages

Message block	Code word vector
0 0 0 0	0 0 0 0 0 0 0
0 0 0 1	0 0 0 1 0 1 1
0 0 1 0	0 0 1 0 1 1 0
0 0 1 1	0 0 1 1 1 0 1
0 1 0 0	0 1 0 0 1 1 1
0 1 0 1	0 1 0 1 1 0 0
0 1 1 0	0 1 1 0 0 0 1
0 1 1 1	0 1 1 1 0 1 0
1 0 0 0	1 0 0 0 0 1 0
1 0 0 1	1 0 0 0 1 1 0
1 0 1 0	1 0 0 1 0 0 1
1 0 1 1	1 0 1 1 0 0 0
1 1 0 0	1 1 0 0 0 0 1
1 1 0 1	1 1 0 1 0 0 0
1 1 1 0	1 1 1 0 0 1 0
1 1 1 1	1 1 1 1 1 0 1

Ques 1.4. Prepare Hamming code for the message "01001001010" assuming even parity. Also explain error detection and correction capabilities at the receiver by assuming an error in any one of the received bits.

AKTU 2015-16, Marks 10

1. 11-bit data word : 01001001010
We include 4 parity bits with the 11-bit word and arrange the

15-bits as follows :

Bit position :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P_1	P_2	0	P_4	1	0	0	P_8	1	0	0	1	0	1	0

2. Each parity bit is calculated as follows :

$$\begin{aligned} P_1 &= \text{XOR of bits } (3, 5, 7, 9, 11, 13, 15) \\ &= 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \\ &= 0 \end{aligned}$$

$$\begin{aligned} P_2 &= \text{XOR of bits } (3, 6, 7, 10, 11, 14, 15) \\ &= 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \\ &= 1 \end{aligned}$$

$$\begin{aligned} P_4 &= \text{XOR of bits } (5, 6, 7, 12, 13, 14, 15) \\ &= 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \\ &= 1 \end{aligned}$$

$$\begin{aligned} P_8 &= \text{XOR of bits } (9, 10, 11, 12, 13, 14, 15) \\ &= 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \\ &= 1 \end{aligned}$$

3. Substituting the 4 parity bits in their proper position, we obtain the 15-bit composite word stored in memory.

Bit position :

0	1	0	1	1	0	0	1	1	0	0	1	0	1	0
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

When the 15-bits are read from memory they are checked again for error, the parity is checked over the same combinations of bits, including the parity bit.

4. The 4 check bits are evaluated as follows :

$$C_1 = \text{XOR of bits } (1, 3, 5, 7, 9, 11, 13, 15) = 0$$

$$C_2 = \text{XOR of bits } (2, 3, 6, 7, 10, 11, 14, 15) = 0$$

$$C_4 = \text{XOR of bits } (4, 5, 6, 7, 12, 13, 14, 15) = 0$$

$$C_8 = \text{XOR of bits } (8, 9, 10, 11, 12, 13, 14, 15) = 0$$

5. A 0 check bit designates even parity over the checked bits and a 1 designates odd parity. Since the bits were stored with even parity, the result, $C = 0000$ indicates that no error has occurred. However, if $C \neq 0$, then the 4-bit binary number formed by the check bits gives the position of the erroneous bits.

6. For example, consider the following three cases :

Bit position :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Case I	0	1	0	1	1	0	0	1	1	0	0	1	0	1
Case II	1	1	0	1	1	0	0	1	1	0	0	1	0	1
Case III	0	1	0	1	0	0	0	1	1	0	0	1	0	1

7. Evaluating the XOR of the corresponding bits, we determine the 4 check bits to be as follows :

	C_8	C_4	C_2	C_1
For no error :	0	0	0	0
With error in bit 1 :	0	0	0	1
With error in bit 5 :	0	1	0	1

The error can be corrected by complementing the corresponding position of the bit in error.

Error detection and correction :

1. The dynamic physical interaction of the electrical signals affecting the data path of a memory unit may cause occasional errors in storing and retrieving the binary information.
2. The reliability of a memory unit may be improved by employing error-detecting and error-correcting codes. The most common error detection scheme is the parity bit.
3. In the Hamming code, k parity bits are added to an n -bit data word, forming a new word of $n + k$ bits.
4. The bit positions are numbered in sequence from 1 to $n + k$. Those positions numbered as a power of 2 are reserved for the parity bits. The remaining bits are the data bits. The code can be used with words of any length.
5. The Hamming code can detect and correct only a single error. By adding another parity bit to the coded word the Hamming code can be used to correct a single error and detect double errors.
6. If we include this additional parity bit, then the previous 15-bit coded word becomes 010110011001010 P_{16} , where P_{16} is evaluated from the EX-OR of the other 15 bits. This produces the 16-bit word 0101100110010101 (even parity).
7. When the 16-bit word is read from memory, the check bits are evaluated, as is the parity P over the entire 16-bits. If $P = 0$, the parity is correct (even parity), but if $P = 1$, then the parity over the 16-bits is incorrect (odd parity). The following four cases can arise :
 - i. If $C = 0$ and $P = 0$, no error occurred.
 - ii. If $C \neq 0$ and $P = 1$, a single error occurred that can be corrected.
 - iii. If $C \neq 0$ and $P = 0$, a double error occurred that is detected, but that cannot be corrected.
 - iv. If $C = 0$ and $P = 1$, an error occurred in the P_{16} bit.
8. This scheme may detect more than two errors, but is not guaranteed to detect all such errors.

Que 1.15. For the Hamming code 1001101001 received at the receiver end, correct this code for error if any ? [AKTU 2014-15, Marks 3.5]

Answer

1. Given, Hamming code received = 1001101001

P_1	P_2	D_3	P_4	D_5	D_6	D_7	P_8	D_9	D_{10}
1	0	0	1	1	0	1	0	0	1

2. Let code is sent and received on the basis of odd parity. Therefore, we can find the correction bits as follows :

$C_1 = \text{EX-OR } (P_1, D_3, D_5, D_7, D_9) = \text{EX-OR } (1, 0, 1, 1, 0) = 0$	$(1, 0, 1, 1, 0) = 0$
$C_2 = \text{EX-OR } (P_2, D_3, D_6, D_7, D_{10}) = \text{EX-OR } (0, 0, 0, 1, 1) = 1$	$(0, 0, 0, 1, 1) = 1$
$C_4 = \text{EX-OR } (P_4, D_5, D_6, D_7) = \text{EX-OR } (1, 1, 0, 1) = 0$	$(1, 1, 0, 1) = 0$
$C_8 = \text{EX-OR } (P_8, D_9, D_{10}, D_{11}) = \text{EX-OR } (0, 0, 1) = 0$	$(0, 0, 1) = 0$
3. Thus error bit location is $C_8 C_4 C_2 C_1 = 0010$
4. Thus error is in 2nd bit. Thus correct code is 1101101001.

Que 1.16. Detect and correct error (if any) in the following received even parity Hamming code word 00111101010. Also find out the correct message.

AKTU 2011-12, Marks 05

Answer

1. The received Hamming code word = 00111101010

2. The parity bit required is determined by

$$2^k \geq n + k + 1$$

where, n = number of data bits

k = number of parity bits

3. Since, the Hamming code word is of 11-bit, i.e.,

$$n + k = 11$$

so,

$$2^k \geq 12$$

4. Hence, number of parity bit required, $k = 4$, (P_1, P_2, P_3, P_4) .

Thus, in this received Hamming code, number of data bit is 7 and number of parity bit is 4.

5. 11-bit Hamming code can be represented as

P_1	P_2	D_3	P_4	D_5	D_6	D_7	P_8	D_9	D_{10}	D_{11}
0	0	1	1	1	1	0	1	0	1	0

6. Now, the check bits can be determined as

$$\begin{aligned} C_1 &= P_1 \oplus D_3 \oplus D_5 \oplus D_7 \oplus D_9 \oplus D_{11} \\ &= 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 = 0 \end{aligned}$$

$$\begin{aligned}C_2 &= P_2 \oplus D_3 \oplus D_6 \oplus D_7 \oplus D_{10} \oplus D_{11} \\&= 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 1\end{aligned}$$

$$\begin{aligned}C_4 &= P_4 \oplus D_5 \oplus D_6 \oplus D_7 \\&= 1 \oplus 1 \oplus 0 = 1\end{aligned}$$

$$\begin{aligned}C_8 &= P_8 \oplus D_9 \oplus D_{10} \oplus D_{11} \\&= 1 \oplus 0 \oplus 1 \oplus 0 = 0\end{aligned}$$

7. Hence, the check bit is 0110, this indicates that the number at 6th location i.e., D_6 is an error. Hence, it is given 1 and it should be 0. Therefore, the correct message bit is 0011001010.

Que 1.17. Design a parity generator to generate an odd parity bit for a 4-bit word. Use EX-OR and EX-OR gate.

AKTU 2012-13, Marks 10

Answer

- The circuit that generates the parity bit in the transmitter is called a parity generator. The circuit that checks the parity in the receiver is called a parity checker.
- Consider a 4-bit data whose input bit is A, B, C and D . So, the truth table for an odd parity generator is

4-bit message				Parity bit
A	B	C	D	P
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

K-map and logic diagram :

		For F			
		00	01	11	10
AB	CD	00	1	1	2
		01	4	5	6
AB	CD	11	1	1	14
		10	8	9	10
		12	13	15	

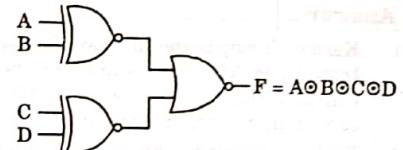


Fig. 1.17.1. 4-bit odd parity.

$$\begin{aligned}F &= A \odot B \odot C \odot D \\&= \text{EX-NOR}(A, B, C, D)\end{aligned}$$

PART-2

The Map Method upto Five Variables, Don't Care Conditions, POS Simplification, NAND and NOR Implementation, Quine Mc-Cluskey Method (Tabular Method).

CONCEPT OUTLINE : PART-2

• Basic theorems and properties of boolean algebra :

- a. $x + 0 = x$ b. $x \cdot 1 = x$
- c. $x + x' = 1$ d. $x \cdot x' = 0$
- e. $x + x = x$ f. $x \cdot x = x$
- g. $x + 1 = 1$ h. $x \cdot 0 = 0$
- Involution : $(x')' = x$
- Commutative : a. $x + y = y + x$ b. $xy = yx$
- Associative : a. $x + (y + z) = (x + y) + z$ b. $x(yz) = (xy)z$
- Distributive : a. $x(y + z) = xy + xz$ b. $x + yz = (x + y)(x + z)$
- De-Morgan's : a. $(x + y)' = x'y'$ b. $(xy)' = x' + y'$
- Absorption : a. $x + xy = x$ b. $x(x + y) = x$

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 1.18. Write a short note on Karnaugh map. Also show the reduction of boolean expression and how to mark pairs. How gate level minimization is implemented?

Answer

1. Karnaugh map is another way of presenting the information given by truth table. These maps are also known by the name *K-map*. Let's 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. 1.18.1;

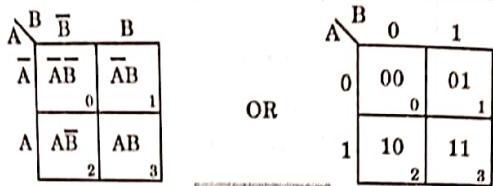


Fig. 1.18.1.

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

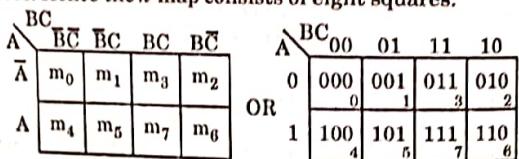


Fig. 1.18.2.

3. The K-map drawn in Fig. 1.18.2, for three variables is marked with numbers in each row and each column to show the relationship between the squares and the three variables.

4. For example, the square assigned to m_0 , which corresponds to row and column 01. When these two numbers are reconsidered, they give the binary number 101, whose decimal equivalent is 5.

For four variables :

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

AB	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}	

AB	CD	00	01	11	10	
00	0000	0	0001	1	0011	2
01	0100	4	0101	5	0111	7
11	1100	12	1101	12	1111	15
10	1000	8	1001	9	1011	11

OR

AB	CD	00	01	11	10	
00	0000	0	0001	1	0011	2
01	0100	4	0101	5	0111	6
11	1100	12	1101	12	1111	14
10	1000	8	1001	9	1011	10

Fig. 1.18.3.

Que 1.19. Minimize the given boolean function using K-map.
 $F(A, B, C, D) = \Sigma m(3, 4, 5, 7, 9, 13, 14, 15)$

Answer

AB	CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	1	2	
$\bar{A}B$	1	1	1	1	2
AB	12	1	1	13	14
$A\bar{B}$	8	1	1	9	10

Fig. 1.19.1.

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

Que 1.20. Explain with example using four variable map. How don't care conditions are implemented in K-map minimization ?

AKTU 2012-13, Marks 05

Simplify the boolean function.

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

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

AKTU 2016-17, Marks 05

Answer

1. Consider the example, which has boolean function $F(w, x, y, z) = \Sigma m(1, 3, 7, 11, 15)$ and don't care conditions

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

2. The minterms of F are the variable combinations that make the function equal to 1. The minterms of d are the don't care minterms that may be either 0 or 1.
3. The K-map simplification is shown in Fig. 1.20.1.

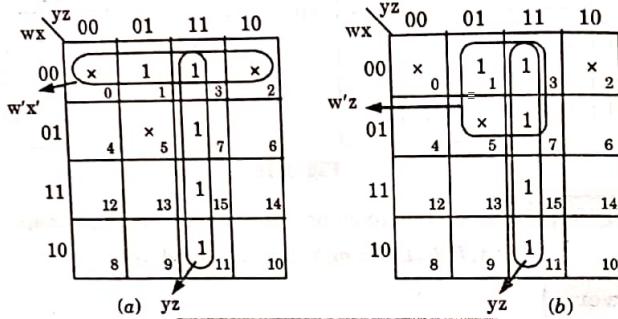


Fig. 1.20.1. K-map minimization.

4. The minterms of F are marked by 1's, those of d are marked by x's and the remaining is filled with 0's.
5. To get the simplified expression in SOP form, we must include all five 1's in the map, but we may or may not include any of the x's, depending on the way the function is simplified.
6. In Fig. 1.20.1(a), don't care minterms 0 and 2 are included with the 1's, resulting as
- $$F = yz + w'x'$$
7. In Fig. 1.20.1(b), don't care minterm 5 is included with the 1's, resulting as
- $$F = yz + w'z$$
8. The K-map in Fig. 1.20.1(b) is more feasible because, we have to use the minimum don't care.

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

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

AKTU 2014-15, Marks 3.5

Answer

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

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

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

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

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

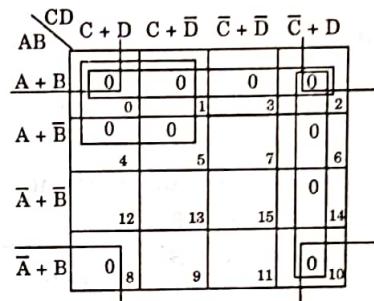


Fig. 1.21.1.

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

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

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

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

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

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

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

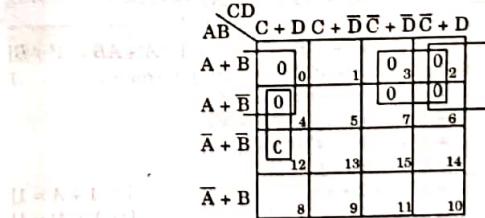


Fig. 1.21.2.

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

Que 1.22. Simplify the following boolean expression to a minimum number of literals.

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

AKTU 2014-15, Marks 3.5

Answer

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

Let

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

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

Let

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

Que 1.23. Simplify the following expression as much as possible:

$F(w, x, y, z) = y'z' + w'x'z' + w'xyz' + wyz'$
and implement your result using universal gates only.

AKTU 2013-14, Marks 2.5

Answer

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

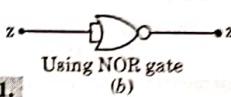
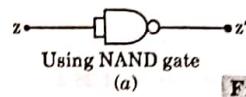


Fig. 1.23.1.

Que 1.24. Simplify the following boolean equation :

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

AKTU 2012-13, Marks 05

Answer

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

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

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

Que 1.25. Express the following boolean function F in a sum of minterms and a product of maxterms.

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

AKTU 2011-12, Marks 05

Answer

$$\begin{aligned} 1. \text{ Given } F(x, y, z) &= (xy + z)(y + xz) \\ &= xy.y + x.xyz + yz + xz.z \end{aligned}$$

$$2. \text{ By associative property, } x.x = x$$

$$\begin{aligned} F_m &= xy + xyz + yz + xz = xy(z + \bar{z}) + xyz + (x + \bar{x})yz + xz(y + \bar{y}) \\ &= xyz + xy\bar{z} + xyz + xyz + \bar{x}yz + xyz + x\bar{y}z \\ &= xyz + xy\bar{z} + \bar{x}yz + x\bar{y}z \quad (\because x + \bar{x} = x) \end{aligned}$$

$$F = \Sigma m(3, 5, 6, 7)$$

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

$$F = \prod M(0, 1, 2, 4)$$

Que 1.26. Implement the following boolean function with NAND gates. $F(x, y, z) = \Sigma m(1, 2, 3, 4, 5, 7)$ AKTU 2016-17, Marks 10

Answer

- The K-map simplification is shown in Fig. 1.26.1.

x \ yz	00	01	11	10
0	0	1	1	1
1	1	1	1	0

Fig. 1.26.1.

2. Hence, the simplified function is

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

3. Implementation using NAND gates is shown in Fig. 1.26.2.

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

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

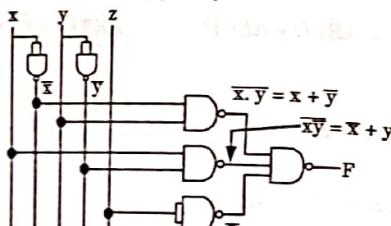


Fig. 1.26.2.

Que 1.27. Simplify the boolean function Y together with don't care condition 'd' using K-map and implement it with two level NAND gate circuit.

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

AKTU 2014-15, Marks 3.5

Answer

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

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

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

2. As there is not given any don't care condition so K-map is as shown in Fig. 1.27.1.

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

Fig. 1.27.1.

3. NAND gate implementation:

$$Y = BD + BC + AC\bar{D}$$

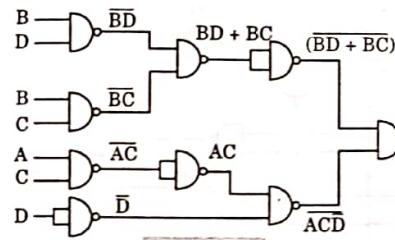


Fig. 1.27.2.

Que 1.28. Minimize the given boolean function using K-map and implement the simplified function using NAND gates only.

$$F(A, B, C, D) = \Sigma m(0, 1, 2, 9, 11, 15) + d(8, 10, 14)$$

AKTU 2011-12, Marks 05

Answer

Minimization using K-map :

1. Given, $F(A, B, C, D) = \Sigma m(0, 1, 2, 9, 11, 15) + d(8, 10, 14)$

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

Fig. 1.28.1.

$$F(A, B, C, D) = AC + \bar{B}\bar{C} + \bar{B}\bar{D} \quad \dots(1.28.1)$$

Implementation using NAND gates :

1. Taking double inversion of eq. (1.28.1)

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

2. Using De-Morgan's theorem :

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

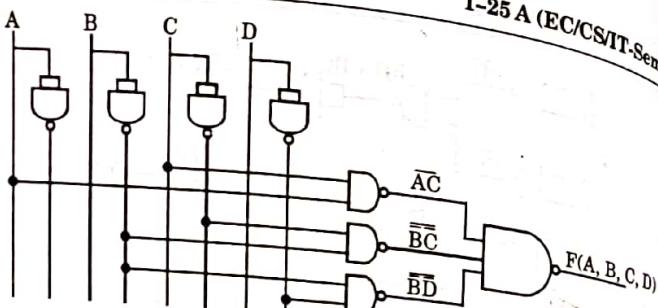


Fig. 1.28.2.

Que 1.29. What is the significance of Quine-Mc Cluskey method or tabular method?

Answer

1. The K-map method is suitable for simplification of boolean function upto 5 or 6 variables.
2. As the number of variables increases beyond this, the visualization of adjacent square is difficult, as the geometry is more involved.
3. The Quine-Mc Cluskey or tabular method is employed in such cases. Consider the function,
 $F(A, B, C, D) = \sum m(0, 2, 3, 6, 7, 8, 10, 12, 13)$ for simplifying using tabular method.

Process of solving Mc Cluskey method :

1. The binary representations are grouped a section of numbers in terms of the number of 1's index as shown in Table 1.29.1.
2. Now compare each binary term with every term in the next higher category.
3. Make the two number sectional combination which differ by one bit.
4. Write binary form of the minterm cell.
5. Compare each binary term with higher adjacent cell and write the combination of 4 cell which are differ by 1-bit.
6. Mark all combinations which are made by the digits of 4 cell.
7. Write the binary form of 4 cell and place a (.) in place of differ bit.

Table 1.29.1.

No. of 1's	Minterms	Binary	Minterms (2 cell)	Binary	Minterm (4 cell)	Binary
0	m_0	0 0 0 0	0, 2✓ 0, 8✓	0 0 _ 0 _ 0 0 0	0, 2, 8, 10	_ 0 _ 0
1	m_2	0 0 1 0	2, 3✓ 2, 6✓	0 0 1 _ 0 _ 1 0	2, 3, 6, 7	0 _ 1 _
	m_8	1 0 0 0	2, 10✓ 8, 10✓	_ 0 1 0 1 0 _ 0		
			8, 12	1 _ 0 0		
2	m_3	0 0 1 1	3, 7✓			
	m_6	0 1 1 0	6, 7✓	0 _ 1 1		
	m_{10}	1 0 1 0	12, 13	0 1 1 _		
	m_{12}	1 1 0 0	12, 13	1 1 0 _		
3	m_7	0 1 1 1				
	m_{13}	1 1 0 1				

8. Apply same process to the resultant stage.
9. All the terms which remain unchecked are the PI's. Now prepare a PI chart to determine essential prime implicants.
10. All the PI's are represented in rows and each minterm of function in a column as shown in Table 1.29.2.
11. Put the ✓ in each row to show the composition of minterms that make PI's.

Table 1.29.2.

Minterms	Prime implicant	✓	m_0	m_1	m_3	m_6	m_7	m_8	m_{10}	m_{12}	m_{13}
$A\bar{C}\bar{D}$	8, 12								○		○
$A\bar{B}\bar{C} \checkmark$	12, 13										○ ○
$\bar{B}\bar{D} \checkmark$	0, 2, 8, 10	○	○						○	○	
$\bar{A}C \checkmark$	2, 3, 6, 7		○	○	○	○	○				

13. The column that contains a single dot ○ is essential prime implicant.
14. A tick mark is put above each column which has only one ○ mark.

15. The sum of all EPI's gives the function in its minimal SOP form.
 $\therefore F(A, B, C, D) = ABC\bar{C} + \bar{B}\bar{D} + \bar{A}C$

Que 1.30. Minimize the following switching function using Quine-Mc Cluskey method.

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

AKTU 2016-17, Marks 15

Answer

Table 1.30.1. For obtaining all the prime implicants.

No. of 1's	Minterms	Binary	Minterms (2 cell)	Binary	Minterm (4 cell)	Binary
0	m_0	00000	0, 1 K	0000_	0, 1, 8, 9	0_00_C
1	m_1	00001	0, 2 J	000_0	1, 9, 17, 25	_001B
	m_2	00010	0, 8 ✓	0_000	8, 9, 24, 25	_100_A
	m_8	01000				
2	m_9	01001	1, 9 ✓	0_001		
	m_{17}	10001	1, 17 I	_0001		
	m_{24}	11000	8, 9 ✓	0100_		
3	m_{21}	10101	9, 25 ✓	_1001		
	m_{25}	11001	17, 21 H	10_01		
4	m_{15}	01111	17, 25 ✓	1_001		
	m_{27}	11011	24, 25 G	1100_		
5	m_{31}	11111	25, 27 F	110_1		
			15, 31 E	_1111		
			27, 31 D	11_11		

Table 1.30.2. Prime-implicant

Minterm	m_0	m_1	\checkmark m_2	m_3	m_4	\checkmark m_{15}	m_{17}	\checkmark m_{21}	m_{24}	m_{26}	m_{27}	m_{31}
A				○	○			○	○	○	○	
B		○			○		○		○	○	○	
C	○	○			○	○					○	○
D							○				○	○
E ✓							○					○
F									○	○	○	
G								○	○			
H ✓								○	○			
I		○					○	○				
J ✓	○		○									
K	○	○										

So the essential prime implicants are,

$$F(x_1, x_2, x_3, x_4, x_5) = E + H + J \\ = x_2x_3x_4x_5 + x_1\bar{x}_2\bar{x}_4x_5 + \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_5$$

Que 1.31. Minimize the following using Quine-Mc Cluskey method:

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

AKTU 2015-16, Marks 15

Answer

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

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

No. of 1's	Minterms	Binary	Minterms (2 cell)	Binary	Minterm (4 cell)	Binary
0	m_0	0 0 0 0	0, 2*	0 0 _ 0	2*, 3, 6, 7	0 _ 1
1	dm_2	0 0 1 0	2*, 3✓	0 0 1 _	5, 7, 13, 15*	0 _ 1
2	m_3 m_5 m_6 m_{10} m_{12}	0 0 1 1 0 1 0 1 0 1 1 0 1 0 1 0 1 1 0 0	2*, 6✓ 2*, 10	0 _ 1 0 _ 0 1 0	3, 7✓ 5, 7✓ 5, 13✓	0 _ 1 1 0 1 _ 1 _ 1 0 1
3	dm_9 m_7 m_{13}	1 0 0 1 0 1 1 1 1 1 0 1	6, 7✓ 12, 13	0 1 1 _ 1 1 0 _	9*, 13	1 _ 0 1
4	dm_{15}	1 1 1 1	7, 15*✓ 13, 15*✓	_ 1 1 1 1 1 _ 1		

No. of 1's	Minterms	Binary	Minterms (2 cell)	Binary	Minterm (4 cell)	Binary
1.	m_8	1 0 0 0	8, 9✓	1 0 0 _	8, 9, 10, 11	1 0 _ _
2.	m_6 m_9 m_{10}	0 1 0 1 1 0 0 1 1 0 1 0	8, 10✓ 5, 7 9, 11✓	1 0 _ 0 0 1 _ 1 1 0 _ 1	10, 11, 14, 15	1 _ 1 _
3.	m_7 m_{11} m_{14}	0 1 1 1 1 0 1 1 1 1 1 0	10, 11✓ 10, 14✓	1 0 1 _ 1 _ 1 0	7, 15	_ 1 1 1
4.	m_{15}	1 1 1 1	11, 15✓ 14, 15✓	1 _ 1 1 1 1 1 _		

2. All the terms which are unchecked are prime implicants.
Now, we prepare a prime implicant chart to determine essential prime implicant is as follows :

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

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

3. Logic diagram :

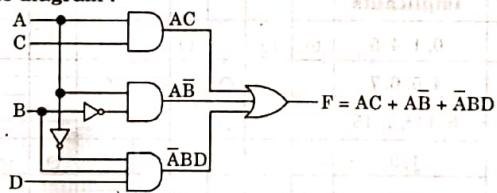


Fig. 1.32.1.

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

Minterm	Prime implicant	m_0	m_3	m_5	m_6	m_7	m_{10}	m_{11}	m_{12}
$\bar{W}\bar{X}\bar{Z}$ ✓	0, 2*	○							
$\bar{X}Y\bar{Z}$ ✓	2*, 10						○		
$WX\bar{Y}$ ✓	12, 13						○	○	
$W\bar{Y}Z$	9*, 13							○	
$\bar{W}Y$ ✓	2*, 3, 6, 7		○		○	○			
XZ ✓	5, 7, 13, 15*			○		○			○

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

Que 1.32. Use Quine-Mc Cluskey (QM) method to solve the following function :

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

AKTU 2014-15, Marks 3.5

Answer

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

Que 1.33. Use the Quine-Mc Cluskey method to generate the set of prime implicants for the following function :

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

Also obtain all minimal expressions for the function. Draw a logic

diagram using only NAND gates to implement your best solution obtained.

Answer

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

No. of 1's	Minterms	Binary	Minterms (2 cell)	Binary	Minterm (4 cell)	Binary
0	m_0	0000	0, 1 ✓	000-	0, 1, 4, 5	0_0
	m_1	0001	0, 4 ✓	0_-00	4, 5, 6, 7	01-
	m_4	0100	1, 5 ✓	0_-01	6, 14*, 7, 15	-11-
2	m_5	0101	1, 9	_-001		
	m_6	0110	4, 5 ✓	010-		
	m_9	1001	4, 6 ✓	01_-0		
	dm_{10}	1010	5, 7 ✓	01_-1		
3	m_7	0111	6, 7 ✓	011-		
	m_{11}	1011	6, 14* ✓	_110		
	dm_{14}	1110	9, 11	10_-1		
4	m_{15}	1111	10*, 11	101-		
			7, 15 ✓	_111		
			11, 15	1_-11		
			14*, 15	111-		

Minterms	Prime implicants	✓	m_0	m_1	m_4	✓	m_4	m_6	m_7	m_9	m_{10}	m_3
$A'C' \checkmark$	0, 1, 4, 5	○	○	○	○							
AB	4, 5, 6, 7											
BC	6, 14*, 7, 15			○		○	○					○
$B'C'D$	1, 9		○		○	○						
$AB'C'$	9, 11		○					○				
$AB'C'$	10*, 11							○				
ACD	11, 15							○	○			
ABC	14*, 15							○				○

The essential prime implicant,

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

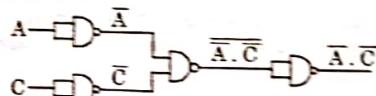


Fig. 1.33.1.

Que 1.34. Minimize the following function using Quine-McCluskey method:

$$F(A, B, C, D) = \sum m(0, 2, 3, 6, 7, 8, 9, 10, 13)$$

AKTU 2011-12, Marks 05

Answer

1. Given, $F(A, B, C, D) = \sum m(0, 2, 3, 6, 7, 8, 9, 10, 13)$

Table 1.34.1.

No. of 1's	Minterms	Binary	Minterms (2 cell)	Binary	Minterm (4 cell)	Binary
0	m_0	0000	0, 2 ✓	00_0	0, 2, 8, 10	_0_0
1	m_2	0010	0, 8 ✓	_000	2, 3, 6, 7	0_-1_-
	m_8	1000	2, 3 ✓	001_-		
2	m_1	0011	2, 6 ✓	0_-10		
	m_6	0110	2, 10 ✓	_010		
	m_9	1001	8, 9	100_-		
	m_{10}	1010	8, 10 ✓	10_-0		
3	m_7	0111	3, 7 ✓	0_-11		
	m_{13}	1101	6, 7 ✓	011_-		
			9, 13 ✓	1_-01		

Table 1.34.2.

Minterm	Prime implicants	✓	m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}	m_{11}	m_{12}	m_{13}
$B'D' \checkmark$	0, 2, 8, 10	○	○											○	○	
$A'C' \checkmark$	2, 3, 6, 7			○	○	○	○	○	○							
$AC'D' \checkmark$	8, 9												○	○		
$ABC' \checkmark$	9, 13												○	○		

Essential prime implicant:

Essential prime implicant is determined by searching of single dot columns and select the prime implicant corresponding to that dot by \checkmark . The check mark in front of it.

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

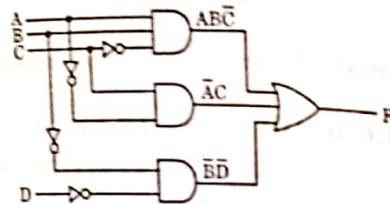


Fig. 1.34.1.

Que 1.35. Minimize the following function by tabular method & implement the result using NAND gate only :

$$F(w, x, y, z) = \sum m(1, 4, 8, 9, 13, 14, 15) + d(2, 3, 11, 12)$$

AKTU 2012-13, Marks 10

Answer

No. of 1's	Minterms	Binary	Minterms (2 cell)	Binary	Minterm (4 cell)	Binary
1.	m_1	0001	1, 3*	0 0 _ 1	1, 3*, 9, 11*	0 1
	dm_2	0010	1, 9*	_ 0 0 1	8, 9, 12*, 13	1 0
	m_4	0100	2*, 3*	0 0 1 _	9, 11*, 13, 15	1 _ 1
	m_5	1000	4, 12*	_ 1 0 0	12*, 13, 14, 15	1 1 _
2.	dm_3	0011	8, 9*	1 0 0 _		
	m_9	1001	8, 12*	1 _ 0 0		
	dm_{12}	1100	3, 11*	_ 0 1 1		
3.	dm_{11}	1011	9, 11*	1 0 _ 1		
	m_{13}	1101	9, 13*	1 _ 0 1		
	m_{14}	1110	12*, 13*	1 1 0 _		
4.	m_{15}	1111	12*, 14*	1 1 _ 0		
			11*, 15*	1 _ 1 1		
			13, 15*	1 1 _ 1		
			14, 15*	1 1 1 1		

Prime implicant selection chart :

	Prime implicants	✓ m_1	✓ m_4	✓ m_8	✓ m_9	✓ m_{13}	✓ m_{14}	✓ m_{15}
$\bar{x}z \checkmark$	1, 3*, 9, 11*	○			○			
$w\bar{y} \checkmark$	8, 9, 12*, 13			○	○	○		
wz	9, 11*, 13, 15			.	○	○		○
$wx \checkmark$	12*, 13, 14, 15					○	○	○
$x\bar{y}\bar{z} \checkmark$	4, 12*		○					

Thus, essential prime implicant are,

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

Implementation using NAND gates :

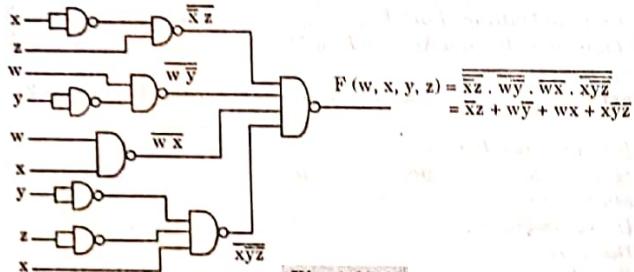


Fig. 1.35.1.



2

UNIT

Combinational Logic

Part-1 (2-2A to 2-16A)

- Combinational Circuits : Analysis Procedure
- Design Procedure
- Binary Adder-Subtractor
- Decimal Adder

A. Concept Outline : Part-1 2-2
B. Long and Medium Answer Type Questions 2-2

Part-2 (2-16A to 2-33A)

- Binary Multiplier
- Magnitude Comparator
- Multiplexers
- Demultiplexers
- Decoders
- Encoders

A. Concept Outline : Part-2 2-16
B. Long and Medium Answer Type Questions 2-17

2-2 A (EC/CS/IT-Sem-3)

Combinational Logic

PART- 1

Combinational Circuits : Analysis Procedure, Design Procedure, Binary Adder-Subtractor, Decimal Adder.

CONCEPT OUTLINE : PART- 1

- **Combinational circuits :** It consists of input variables, logic gates, and output variables. The logic gates accept signals from the input variables and generate output signals. This process transforms binary information from the given input data to the required output data.
- **Half adder :** It needs two binary inputs, augend and addend bits and two binary outputs, sum and carry.
- **Full adder :** It performs the arithmetic sum of three input bits. It consists of three inputs and two outputs.
- **Half subtractor :** It is a combinational circuit that subtracts two bits and produces their difference. It also has an output to specify if a 1 has been borrowed.
- **Full subtractor :** It performs a subtraction between two bits, taking into account borrow of the lower significant stage. The circuit has three inputs and two outputs.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

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.

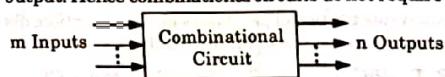


Fig. 2.1.1.

2-1 A (EC/CS/IT-Sem-3)

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. Explain the analysis procedure for combinational logic circuits.

Answer

- The analysis of a combinational circuit is that we determine the function that the circuit implements. In this analysis, the logic diagram is given and culminates with a set of boolean function, a truth table or possible explanation of the circuit operation.
- The important note that the analysis is to make is whether the given circuit is combinational or sequential logic circuit. The diagram of a combinational logic circuit has logic gates with no feedback paths or memory elements.

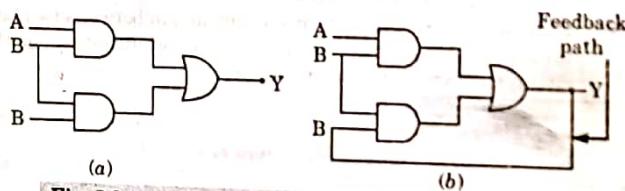


Fig. 2.2.1. (a) Example of combinational logic circuit and
 (b) Not a combinational logic circuit.

Boolean expression from logic diagram :

- Once the logic diagram is verified as a combinational circuit, then we can obtain the boolean function.
 - Label all gate outputs that are a function of input with arbitrary symbols. Determine the boolean functions for each gate output.
 - Label the gates that are a function of input variables and previously labeled gates with other arbitrary symbol. Find the boolean functions for these gates.
 - Repeat the step (ii) until the outputs of the circuits are obtained.
 - By repeated substitution of previously defined function, obtain the output boolean function in terms of input variables.

$$T_1 = AB, T_2 = BC, Y = T_1 + T_2 = AB + BC = B(A + C)$$

A	B	C	T ₁	T ₂	y
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	1	1
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	1	0	1
1	1	1	1	1	1

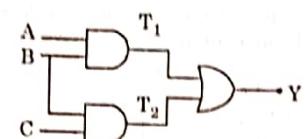


Fig. 2.2.2.

Que 2.3. Explain the design procedure for combinational logic circuits.

Answer

- From the specifications of the circuits, determine the required number of inputs and outputs and assign a symbol to each.
- Derive the truth table that defines the required relationship between inputs and outputs.
- Obtain the simplified boolean functions for each output as a function of the input variables.
- Draw the logic diagram and verify the correctness of the design (manually or by simulation).

Que 2.4. Construct a BCD to excess-3 code converter with a 4-bit adder. What must be done to change the circuit to excess-3 to BCD code converter?

AKTU 2011-12, Marks 10

OR

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

AKTU 2016-17, Marks 15

Answer

Truth table :

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

- The maps in Fig. 2.4.1, are plotted to obtain simplified boolean function for the outputs.
- A two-level logic diagram of each output may be obtained directly from the boolean expressions derived from the maps.

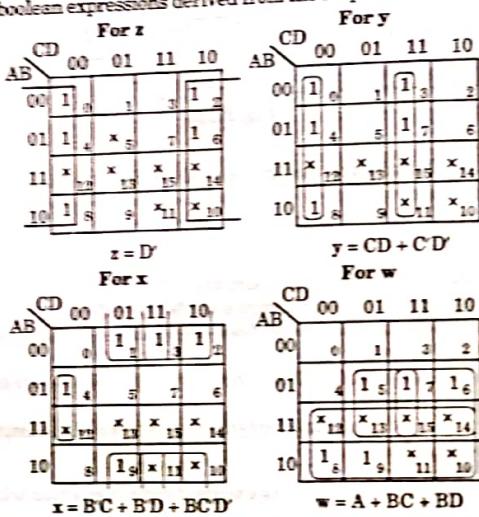


Fig. 2.4.1. Maps for BCD to excess-3 code converter.

- The expressions derived from the maps are :

$$\begin{aligned} z &= D \\ y &= CD + C'D' = CD + (C + D)' \\ x &= BC + BD + BCD' = B(C + D) + BCD' \\ &= B(C + D) + B(C + D)' \\ w &= A + BC + BD = A + B(C + D) \end{aligned}$$

- The logic diagram that implements these expressions is shown in Fig. 2.4.2.

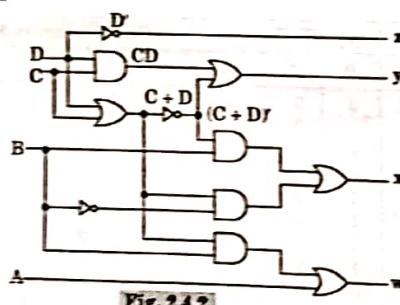


Fig. 2.4.2.

- To change the circuit to an excess-3 code to BCD code, the excess-3 code should be provided as input of combinational circuit and BCD number should be generated at output.

Que 2.5. Design a combinational circuit that converts a 3-bit Gray code to a 3-bit binary number. Implement the circuit with

- Exclusive-OR gate

- NAND gate only.

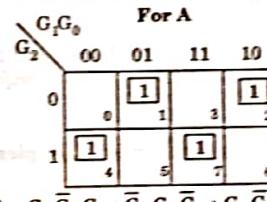
AKTU 2012-13, 2013-14; Marks 10

Answer

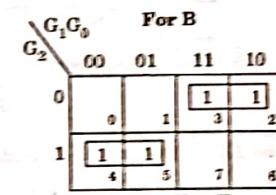
Gray code to binary code converter :

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

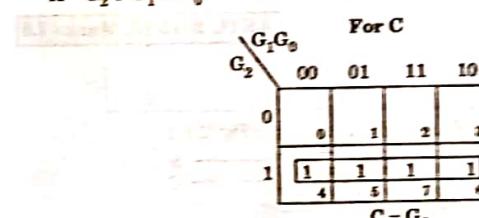
K-map simplification :



$$\begin{aligned} A &= G_2 \bar{G}_1 G_0 + \bar{G}_2 G_1 \bar{G}_0 + G_2 \bar{G}_1 \bar{G}_0 + G_2 G_1 G_0 \\ A &= G_2 \oplus G_1 \oplus G_0 \end{aligned}$$



$$\begin{aligned} B &= G_2 \bar{G}_1 + \bar{G}_2 G_1 \\ B &= G_2 \oplus G_1 \end{aligned}$$



$$C = G_2$$

Fig. 2.5.1.

Logic diagram :
i. Using XOR gates:

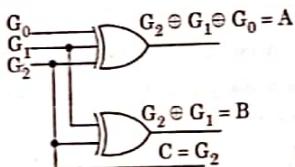


Fig. 2.5.2.

ii. Using NAND gates:

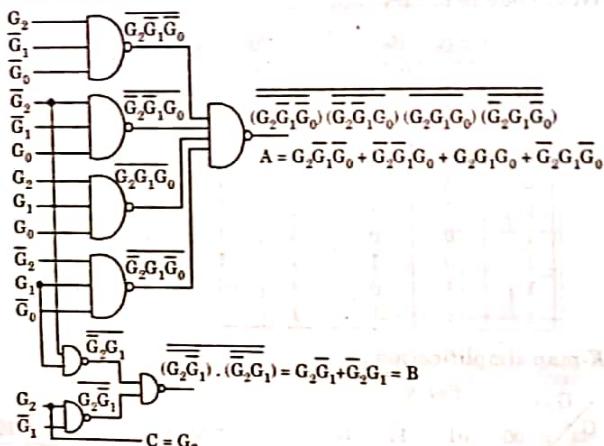


Fig. 2.5.3.

Que 2.6. Describe half adder and full adder in brief. Implement the circuit using logic gates.

OR

Design a full adder using two half adders.

AKTU 2015-16, Marks 7.5

Answer

Half adder:

1. The block diagram of half adder is shown in Fig. 2.6.1.

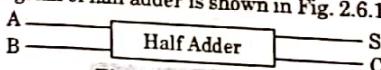


Fig. 2.6.1. Half adder.

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. 2.6.2.

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

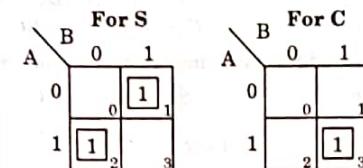
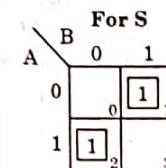


Fig. 2.6.2.

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. 2.6.3.

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 is the previous carry. The block diagram is shown in Fig. 2.6.4.

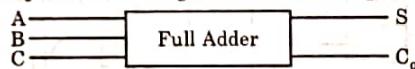


Fig. 2.6.4. Full Adder.

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

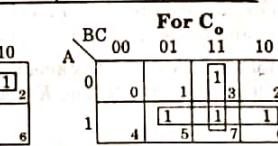
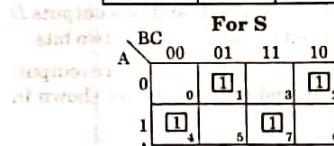


Fig. 2.6.5.

3. Sum : $S = ABC + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C$
 Carry : $C_o = AB + AC + BC$
4. A full adder can be implemented using two half adders and one OR gate.
 Sum : $S = ABC + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C$
 $= ABC + \bar{A}\bar{B}C + A\bar{B}\bar{C} + \bar{A}\bar{B}C$
 $= C(AB + \bar{A}\bar{B}) + \bar{C}(A\bar{B} + \bar{A}B)$
 $= C(A\bar{B} + \bar{A}B)' + C'(A\bar{B} + \bar{A}B)$
 $= (A \oplus B) \oplus C$
 Carry : $C_o = AB + AC + BC$
 $= AB + C(A + B)$
 $= AB + C(A + \bar{A})(A + \bar{B})(B + \bar{B})$
 $= AB + C[AB + A\bar{B} + \bar{A}B]$
 $= AB + ABC + C(A\bar{B} + \bar{A}B)$
 $= AB(1 + C) + C(A \oplus B)$
 $= AB + C(A \oplus B)$

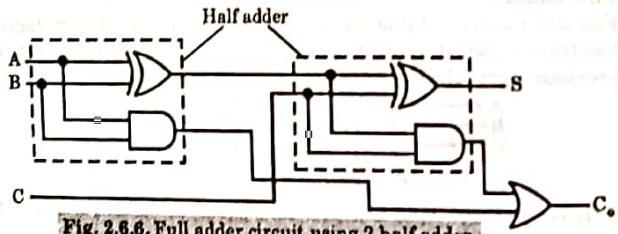


Fig. 2.6.6. Full adder circuit using 2 half adder.

Ques 2.7. Describe a half subtractor with its logic diagram.

Answer

1. The block diagram is shown in Fig. 2.7.1.

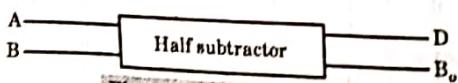


Fig. 2.7.1. Half subtractor.

2. It has two inputs, A (minuend) and B (subtrahend) and two outputs D (difference) and Bo_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 ≠ B. The K-map and truth table are shown in Fig. 2.7.2.

Inputs		Outputs	
A	B	D	B _{out}
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

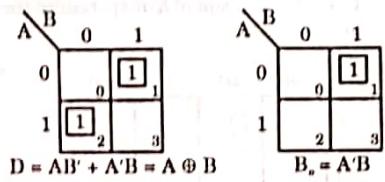


Fig. 2.7.2.

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

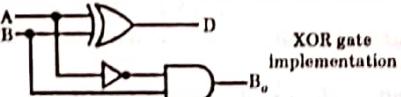


Fig. 2.7.3.

Ques 2.8. Design a full subtractor circuit with three inputs x, y, and two outputs 'Diff' and B_{out}. The circuit subtracts x - y - B_{in}, where, B_{in} is the input borrow, B_{out} is the output borrow and 'Diff' is the difference.

AKTU 2016-17, Marks 10

Answer

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

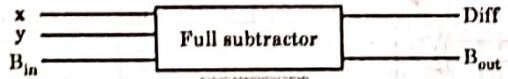


Fig. 2.8.1.

2. Fig. 2.8.1 shows the block diagram approach of full subtractor. It has three inputs x, y and B_{in} and two outputs 'Diff' and B_{out} produced by subtraction of three input bits.
 3. For the formation of truth table, eight possible combinations of three input variables with their outputs are required.

Inputs			Outputs	
x	y	B _{in}	Diff	B _{out}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

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

	yB_{in}	00	01	11	10
x		0	1	3	2
0		0	1	3	2
1		4	5	7	6

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

	yB_{in}	00	01	11	10
x		0	1	1	1
0		0	1	1	1
1		4	5	7	6

Fig. 2.8.2.

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

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

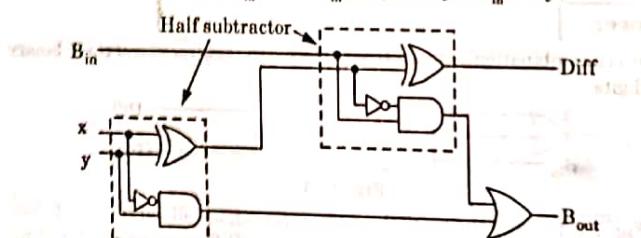


Fig. 2.8.3. Full subtractor circuit using 2 half subtractor.

Que 2.9. Write a short note on binary adder.

Answer

1. A single full-adder is capable of adding three one 1-bit binary numbers or two 1-bit numbers and one previous carry, in order to add binary numbers, with more than one bit, additional full adders must be employed.
2. A 4-bit parallel adder can be constructed using four full adders as shown in Fig. 2.9.1. These four full adders are connected in cascade, i.e., the carry input of the next higher-order adder. An n -bit parallel adder is constructed using ' n ' number of full adders.

Fig. 2.9.1 shows the interconnection of four full adders to provide a 4-bit binary parallel adder.

4. The augend bits of A and the addend bits of B are designated by subscript numbers from right to left, with subscript 0 denoting the low order bit.
5. The carries are connected in chain through the full adders. The input carry is denoted by C_{in} and output is C_{out} . The S output generates the required sum bits.

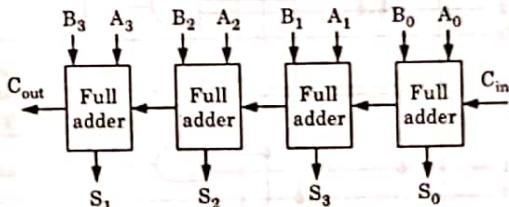
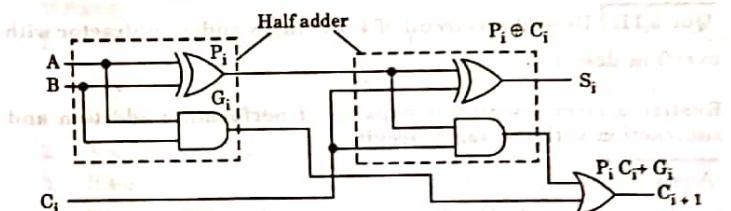


Fig. 2.9.1. Block diagram of parallel adder.

Que 2.10. Describe carry look ahead adder.

Answer

1. The addition of two binary numbers in parallel implies that all the bits of the augend and addend are available for computation at the same time.
2. The carry propagation time is an important attribute of the adder because it limits the speed with which two numbers are added.
3. An obvious solution is to increase the complexity of the equipment in such a way that the carry delay time is reduced.



4. Consider the circuit of the full adder shown in Fig. 2.10.1. If we define two new binary variables

$$P_i = A_i \oplus B_i, G_i = A_i B_i$$

5. The output sum and carry can respectively be expressed as
 $S_i = P_i \oplus C_i, C_{i+1} = G_i + P_i C_i$

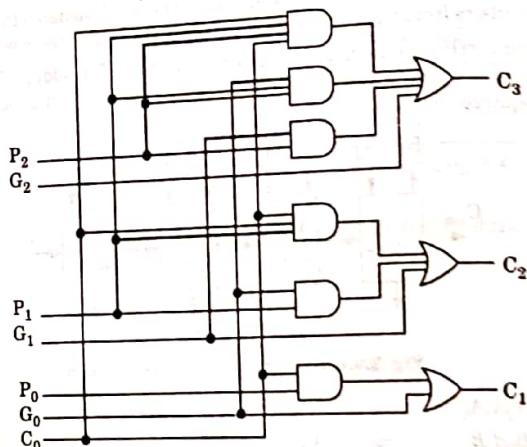


Fig. 2.10.2. Logic diagram of carry look ahead generator.

6. G_i is called a carry generate, and it produces a carry of 1 when both A_i and B_i are 1, regardless of the input carry C_i . P_i is called a carry propagate because it determines whether a carry into stage i will propagate to stage $i+1$.

$$C_0 = \text{input carry}$$

$$C_1 = G_0 + P_0 C_0$$

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 (G_0 + P_0 C_0) \\ = G_1 + P_1 G_0 + P_0 P_1 C_0$$

$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

Que 2.11. Describe a circuit of 4-bit binary adder-subtractor with overflow detection.

OR

Realize a circuit which is capable of performing addition & subtraction with the same circuit.

Answer

- The addition and subtraction operations can be combined into one circuit with one common binary adder by including an exclusive-OR gate with each full adder.
- A 4-bit adder-subtractor circuit is shown in Fig. 2.11.1. The mode input M controls the operation.

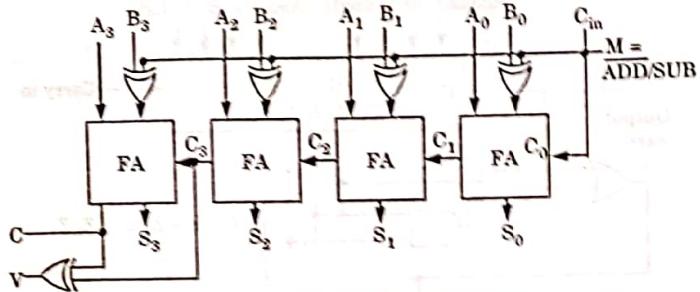


Fig. 2.11.1. 4-bit adder-subtractor (with overflow detection).

- When $M = 0$, the circuit is an adder, and when $M = 1$, the circuit becomes a subtractor, each exclusive-OR gate receives input M and one of the inputs of B .
- When $M = 0$, we have $B \oplus 0 = B$. The full adders receive 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.
- The circuit performs the operation A plus the 2's complement of B . (The exclusive-OR with output V is for detecting an overflow).
- When two numbers with n digits each are added and the sum is a number occupying $n+1$ digits, we say that an overflow occurred.
- Overflow is a problem in digital computers because the number of bits that hold the number is finite and a result that contains $n+1$ bits cannot be accommodated by an n -bit word.

Que 2.12. Discuss BCD adder with its block diagram.

Answer

- BCD adder is a 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. 2.12.1.
 - Add two 4-bit BCD numbers using straight binary addition.
 - If 4-bit sum is equal to or less than 9, the sum is 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 (01100)₂ should be added to the sum to produce the valid BCD symbols.

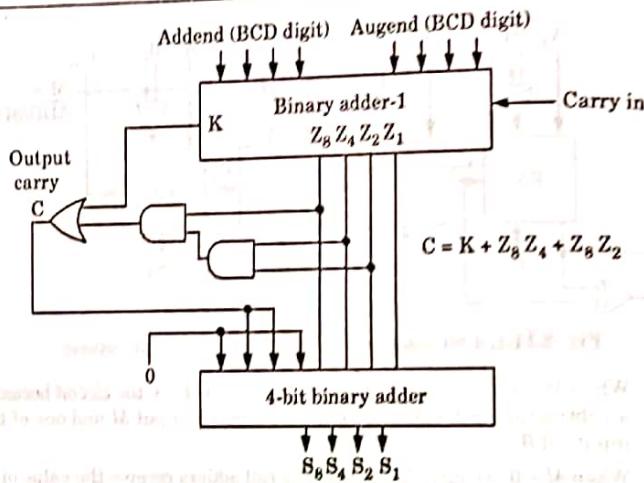


Fig. 2.12.1. Block diagram of a BCD adder.

Binary Sum				BCD Sum					Decimal
Z ₈	Z ₄	Z ₂	Z ₁	C	S ₈	S ₄	S ₂	S ₁	
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	0	1	0	4
0	1	0	1	0	0	1	0	0	5
0	1	1	0	0	0	1	0	1	6
1	0	0	1	0	1	0	1	1	7
1	0	0	1	0	1	0	0	0	8
1	0	1	0	1	0	0	0	1	9
1	0	1	1	1	0	0	0	0	10
								1	11

Que 2.13. Discuss excess-3 adder with its block diagram.

Answer

- Excess-3 code is obtained by adding, 3, to a BCD code. For example, the excess-3 code of 3 is 0011.
- The excess-3 adder performs the addition of two excess-3 number. The following steps are followed to perform excess-3 code addition. Fig. 2.13.1 shows 4-bit excess-3 adder.

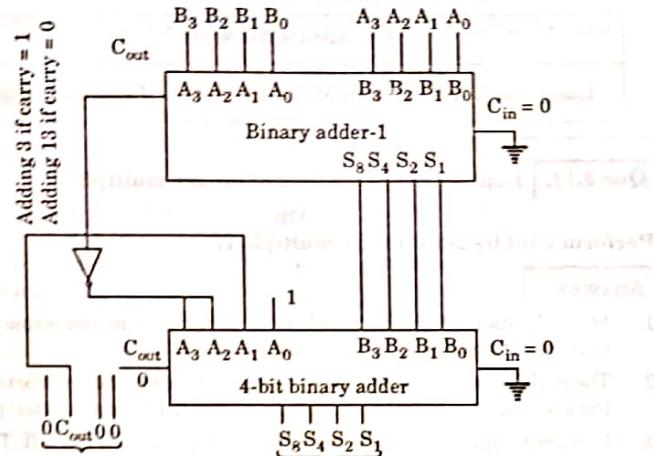


Fig. 2.13.1. 4-bit excess-3 adder.

- Add two excess-3 number.
- If the above sum produces a carry, add 3 (0011) to the sum of two digits. If the above sum (step 1) does not produce a carry, subtract 3 from the above sum (or) add 13 to the sum of two digits.

PART-2

Binary Multiplier, Magnitude Comparator, Multiplexers, Demultiplexers, Decoders, Encoders.

CONCEPT OUTLINE : PART-2

- Binary multiplier:** It is a combinational circuit which performs the multiplication of binary numbers in the same way as multiplication of decimal numbers.
- Magnitude comparator:** It is a combinational circuit that compares two numbers A and B and determines their relative magnitudes. The outcome of the comparison is specified by three binary variables that indicate whether $A > B$, $A = B$ or $A < B$.
- Decoders:** It is a combinational circuit that converts binary information from n input lines to a maximum of 2^n output lines.
- Encoders:** An encoder is a digital circuit that performs the inverse operation of a decoder. An encoder has 2^n input lines and n output lines.

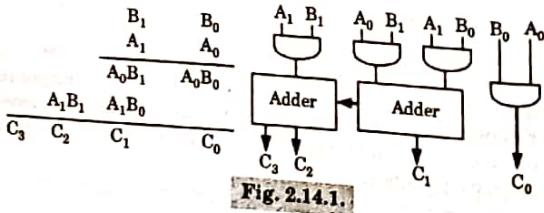
Questions-Answers**Long Answer Type and Medium Answer Type Questions****Que 2.14.** Explain the procedure of binary multiplier.

OR

Perform 2-bit by 2-bit binary multiplier.

Answer

- Multiplication of binary numbers is performed in the same way as 4. multiplication of decimal numbers.
- The multiplicand is multiplied by each bit of the multiplier, starting from the least significant bit. Each such multiplication forms a partial product.
- Successive partial products are shifted one position to the left. The final product is obtained from the sum of the partial product.
- To see how a binary multiplier can be implemented with a combinational circuit, consider the multiplication of two 2-bit numbers as shown in Fig. 2.14.1.
- The multiplicand bits are B_1 and B_0 , the multiplier bits are A_1 and A_0 , and the product is $C_3 C_2 C_1 C_0$. The first partial product is formed by multiplying $B_1 B_0$ by A_0 .
- The multiplication of two bits such as A_0 and B_0 produces 1 if both bits are 1; otherwise, it produces 0. This is identical to an AND operation. Therefore, the partial product can be implemented with AND gates as shown in the Fig. 2.14.1.
- The second partial product is formed by multiplying $B_1 B_0$ by A_1 and shifting one position to the left.
- The two partial products are added with two half adder (HA) circuits. Usually, there are more bits in the partial products and it is necessary to use full adders to produce the sum of the partial products.
- Note that the least significant bit of the product does not have to go through an adder, since it is formed by the output of the first AND gate.

**Que 2.15.** Design a 4-bit binary multiplier with 3-bit binary multiplier.**Answer**

- Consider a multiplier circuit that multiplies a binary number of 4-bits by a number of 3-bits.
- Let the multiplicand represented by $B_3 B_2 B_1 B_0$ and the multiplier by $A_2 A_1 A_0$.
- Since $K = 4$ and $J = 3$, we need 12 AND gates and two 4-bit adders to produce a product of seven bits.

The logic diagram of 4-bit by 3-bit multiplier is shown in Fig. 2.15.1.

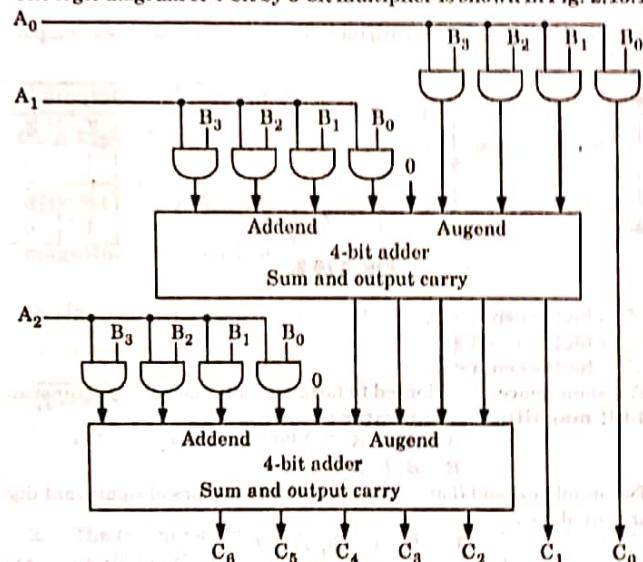


Fig. 2.15.1. 4-bit by 3-bit binary multiplier.

Que 2.16. Design a single bit magnitude comparator.

OR

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

AKTU 2011-12, Marks 10

Answer

- A magnitude comparator is a combinational circuit designed primarily to compare the relative magnitude of the two binary numbers A and B .

Digital Logic Design

2-19 A (EC/CS/IT-Sem-3)

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. 2.16.1.



Fig. 2.16.1.

4. EX-NOR and AND gate is used to implement the circuit. If the 1 NOR gate and two AND gates are combined, the circuit will function as a single bit magnitude comparator as shown in Fig. 2.16.2.
5. The circuit diagram and truth table of a single bit magnitude comparator is shown in Fig. 2.16.1.

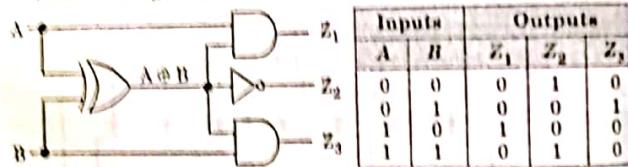


Fig. 2.16.2.

Z_1 is high when $A > B$,

Z_2 is high when $A = B$,

Z_3 is high when $A < B$.

The same concept is adopted to form an n -bit magnitude comparator.

3-bit magnitude comparator :

$$A = A_2 A_1 A_0$$

$$B = B_2 B_1 B_0$$

Two 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 :

$$(A = B) = x_0 x_1 x_0 = (A_2 \odot B_2)(A_1 \odot B_1)(A_0 \odot B_0)$$

$$(A > B) = A_2 \bar{B}_2 + x_2 A_1 \bar{B}_1 + x_2 x_1 A_0 \bar{B}_0$$

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

2-20 A (EC/CS/IT-Sem-3)

Combinational Logic

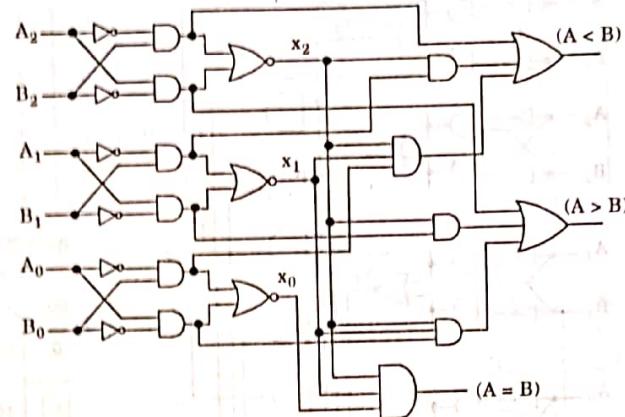


Fig. 2.16.3. 3-bit magnitude comparator using logic gates.

- Que 2.17. Design and explain the logic and circuit of 4-bit magnitude comparator.**

AKTU 2014-15, Marks 06

OR

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

AKTU 2015-16, Marks 10

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

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

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

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

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

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

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

4. The logical implementation is shown in Fig. 2.17.1.

Digital Logic Design

2-21 A (EC/CS/IT-Sem-3)

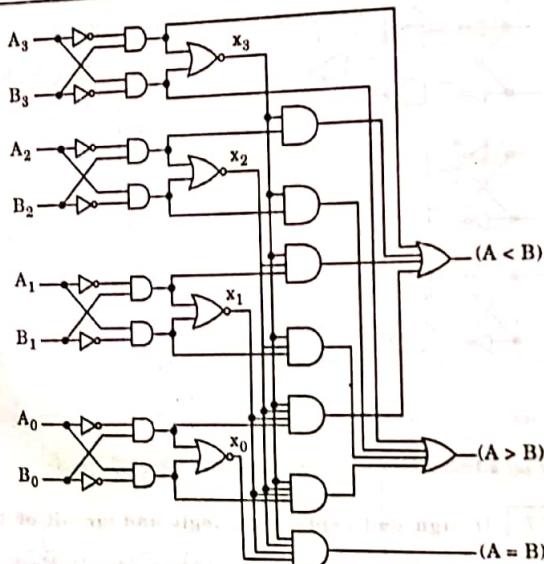


Fig. 2.17.1. 4-bit magnitude comparator using logic gates.

Que 2.18. Design a combinational circuit that compare two 2-bit number A and B to check if they are equal. Use NAND gates only or NOR gates only to implement your design.

AKTU 2013-14, Marks 05

Answer

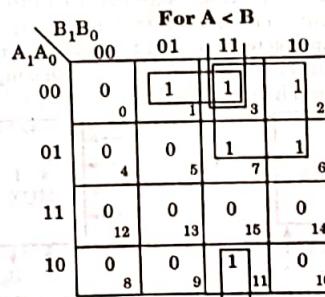
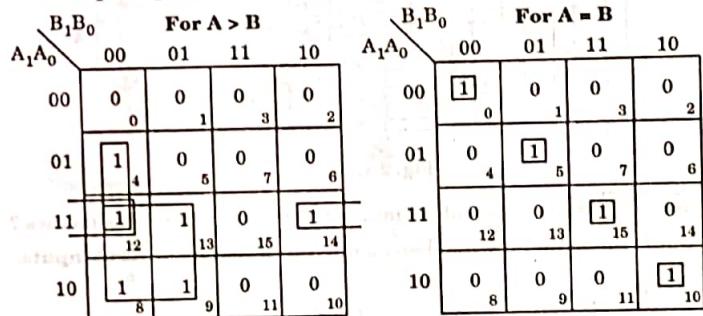
Inputs				Outputs		
A_1	A_0	B_1	B_0	$A > B$	$A = B$	$A < B$
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	0	0	1
0	1	0	1	1	0	0
0	1	1	0	0	1	0
0	1	1	1	0	0	1
1	0	0	0	1	0	0

2-22 A (EC/CS/IT-Sem-3)

Combinational Logic

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

K-map simplification :



$$\begin{aligned} (A = B) &= \bar{A}_1\bar{A}_0\bar{B}_1\bar{B}_0 + \bar{A}_1\bar{A}_0\bar{B}_1B_0 + A_1A_0B_1B_0 + A_1\bar{A}_0B_1\bar{B}_0 \\ &= \bar{A}_1\bar{B}_1(\bar{A}_0\bar{B}_0 + A_0B_0) + A_1B_1(A_0B_0 + \bar{A}_0\bar{B}_0) \\ &= (A_0 \odot B_0)(A_1 \odot B_1) \\ (A < B) &= \bar{A}_1\bar{A}_0B_0 + \bar{A}_0B_1B_0 + \bar{A}_1B_1 \end{aligned}$$

Fig. 2.18.1.

Logic diagram :

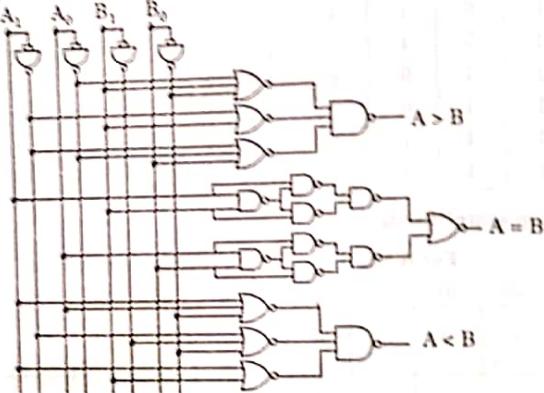


Fig. 2.18.2.

Que 2.19. What is the role of multiplexer in the digital electronics? Explain the logic how it selects a one input among several inputs?

Answer

1. A multiplexer (MUX) is a combinational circuit that selects one input out of several inputs and directs it to a single output.
2. The particular input selection is controlled by a set of select inputs.
3. The block diagram of a digital multiplexer with n input lines and single output line is shown in Fig. 2.19.1(a).

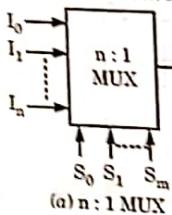
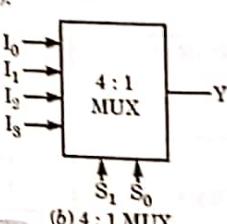


Fig. 2.19.1.



(b) 4 : 1 MUX

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 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. 2.19.1(b).

Que 2.20. Implement a full adder circuit using 4×1 multiplexers.

Answer**Implementation of full adder circuit using 4×1 MUX :**

Canonical form of sum and carry for full adder :

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

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

For sum :

	I ₀	I ₁	I ₂	I ₃
\bar{A}	0	(1)	(2)	3
A	(4)	5	6	(7)
	A	\bar{A}	\bar{A}	A

Implementation table

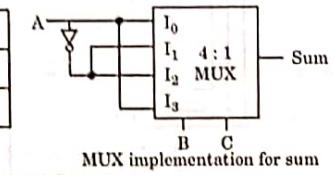


Fig. 2.20.1.

For carry :

	I ₀	I ₁	I ₂	I ₃
\bar{A}	0	1	2	(3)
A	4	(5)	(6)	(7)
	0	A	A	1

Implementation table

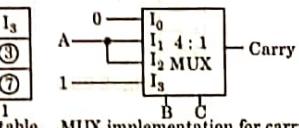


Fig. 2.20.2.

Que 2.21. Implement the function :

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

Using 4 : 1 multiplexer using B and C variables to the selection lines.**AKTU 2011-12, Marks 05****Answer**

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

Input	I ₀	I ₁	I ₂	I ₃
A'	0	(1)	(2)	3
A	(4)	5	6	(7)

Output to multiplexer

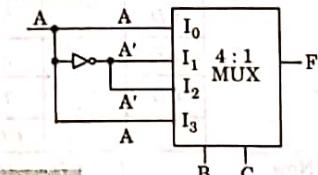


Fig. 2.21.1.

Que 2.22. Construct a 16×1 multiplexer with two 8×1 and one 2×1 multiplexer. Use block diagrams.

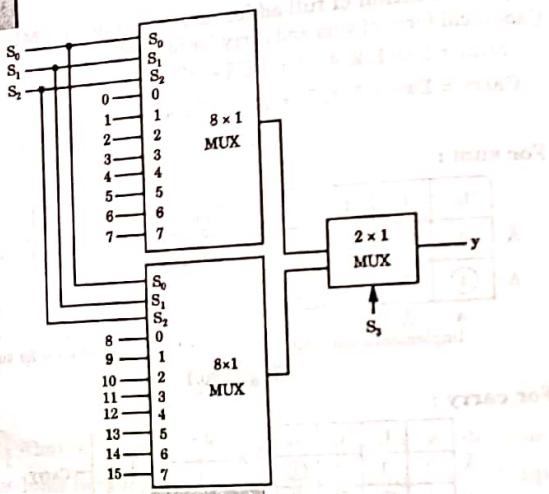
Answer

Fig. 2.22.1.

Que 2.23. Design the following boolean function using 4:1 multiplexer.

AKTU 2014-15, Marks 1

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

Answer

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

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

3. Now,

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

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

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

$$I_3 = CD$$

4. Logic diagram is shown in Fig. 2.23.1

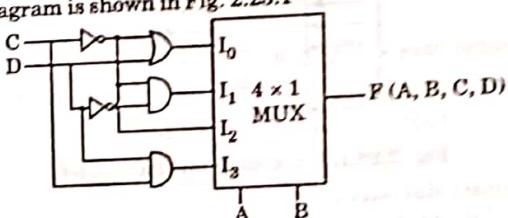


Fig. 2.23.1.

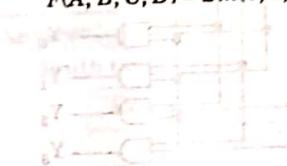
Que 2.24. Design the following boolean function using the multiplexer:

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

AKTU 2013-14, Marks 05

Answer

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



(a) Implementation table

(b) Multiplexer implementation

Fig. 2.24.1.

Que 2.25. Write a short note on decoder.

Answer

- A decoder is a combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines.
- If the n -bit coded information has unused combinations, the decoder may have fewer than 2^n outputs.
- 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.

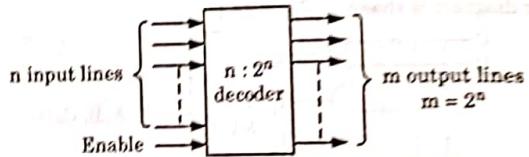


Fig. 2.25.1. Block diagram of a decoder.

2 to 4 binary decoder:

Fig. 2.25.2 shows the 2 to 4 decoder. Here 2 represent the input lines and 4 represents output lines. Fig. 2.25.2 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

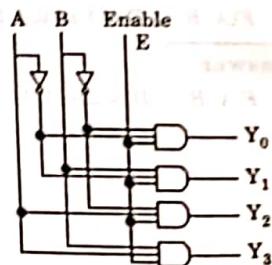


Fig. 2.25.2. Logic diagram of 2 to 4 decoder.

Que 2.26. Using a decoder and external gates, design the combinational circuit defined by the following three boolean functions :

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

AKTU 2015-16, Marks 10

Answer

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

$$F_1 = x'y'z' + xz = x'y'z' + xz(y + y') \\ = x'y'z' + xyz + xy'z = \Sigma m(2, 7, 5)$$

$$F_2 = xy'z' + yz' = xy'z' + (x + x')y, z' \\ = xy'z' + xyz' + x'y'z' = \Sigma m(2, 4, 6)$$

$$F_3 = x'y'z' + xy = x'y'z' + xy(z + z') \\ = x'y'z' + xyz + xyz' = \Sigma m(0, 6, 7)$$

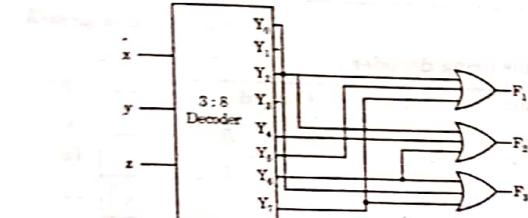


Fig. 2.26.1. Implementation of the given Boolean function using 3:8 decoder.

Que 2.27. Draw the logic diagram of a two to four line decoder using NOR gates only.

AKTU 2016-17, Marks 10

OR

Design a logic circuit diagram of a 2-to-4 line decoder with an enable input using only NOR gates.

AKTU 2013-14, Marks 10

Answer

Truth table :

Enable	Input		Output				
	E	A	B	Y_3	Y_2	Y_1	Y_0
0	x	x	x	0	0	0	0
1	0	0	0	0	0	0	1
1	0	1	0	0	0	1	0
1	1	0	0	0	1	0	0
1	1	1	1	0	0	0	0

Circuit using NOR gate :

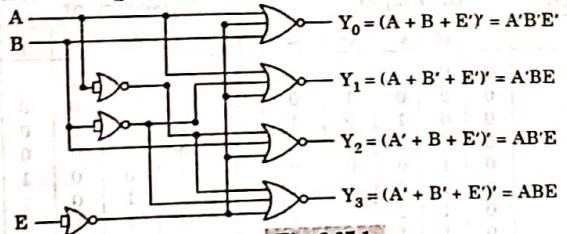


Fig. 2.27.1.

Que 2.28. Design a full subtractor circuit with a decoder and two OR gates.

AKTU 2011-12, Marks 10

Answer

Full subtractor using decoder :

Input			Output	
A	B	C	D	B_0
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Difference :

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

Borrow :

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

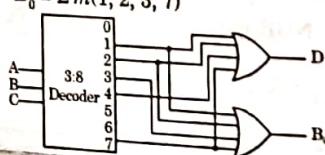


Fig. 2.28.1. Full subtractor using 3:8 decoder.

Que 2.29. Design a BCD to 7 segment decoder. Assume positive logic, minimize the function.

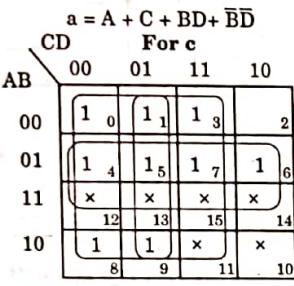
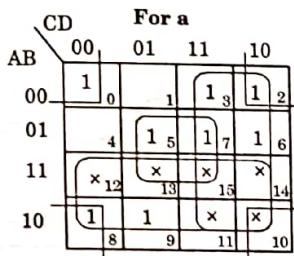
AKTU 2014-15, Marks 06

Answer

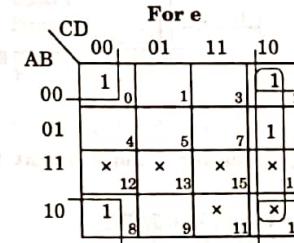
Digit	BCD input				7-segment						
	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	1	0	1	1	1
3	0	0	1	1	1	1	0	1	1	0	1
4	0	1	0	0	1	1	1	1	0	0	0
5	0	1	0	0	0	1	1	1	0	0	1
6	0	1	0	1	0	1	1	1	0	1	1
7	0	1	1	0	1	0	1	1	1	0	1
8	1	0	1	1	1	0	1	1	1	1	1
9	1	0	0	0	1	1	1	1	0	1	0

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

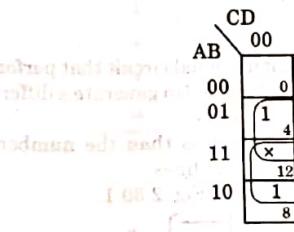
K-map simplification :



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

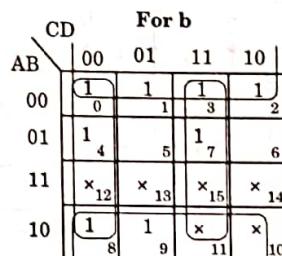


$$e = \bar{B} \bar{D} + C \bar{D}$$

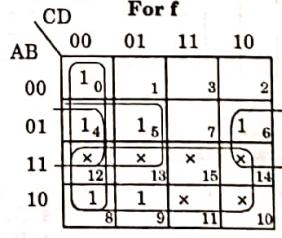


$$g = A + B \bar{C} + \bar{B} C + C \bar{D}$$

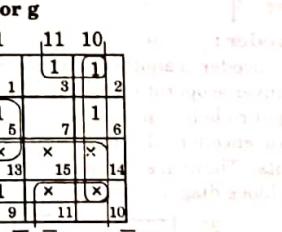
Fig. 2.29.1.



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



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



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

Fig. 2.29.1.

3. Fig. 2.29.2 shows the logic diagram of BCD to 7-segment display decoder.

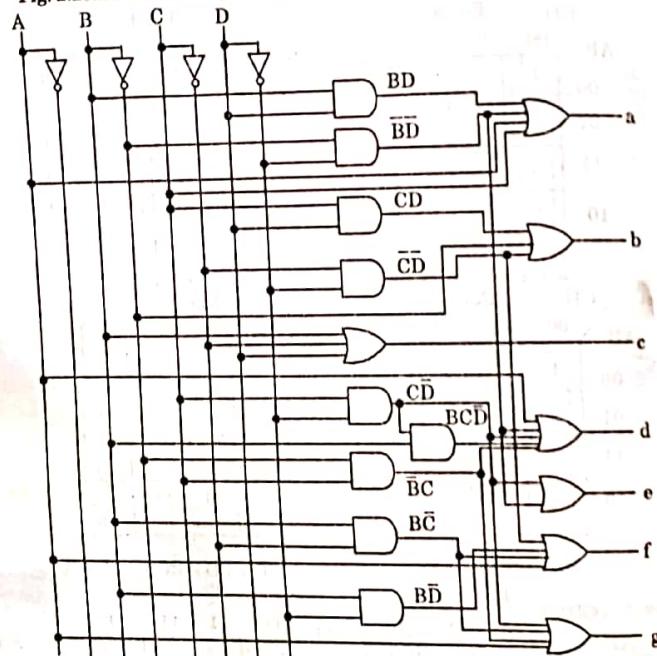


Fig. 2.29.2.

Que 2.30. What do you mean by encoder? Implement the following functions using 3 : 8 decoder.

$$F_1(A, B, C) = \Sigma m(0, 1, 3, 5, 6)$$

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. 3.20.1

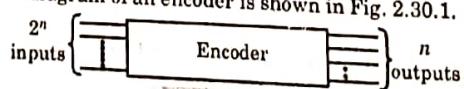


Fig. 2.30.1.

Implementation of given function using 3×8 decoder :

$$F_1(A, B, C) = \Sigma m(0, 1, 3, 5, 6)$$

$$F_2(A, B, C) = \Sigma m(0, 2, 4, 7)$$

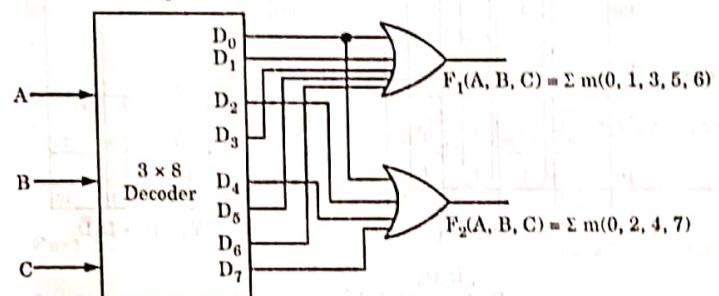


Fig. 2.30.2.

Que 2.31. What is priority encoder? Explain with the help of suitable example.

Answer

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

Example :

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

Y_0, Y_1 : binary output

V: validity of output

Table 2.31.1 Truth table for 4-bit priority encoder.

Input				Output			V
D ₀	D ₁	D ₂	D ₃	Y ₁	Y ₀		
0	0	0	0	x	x		0
1	0	0	0	0	0		1
x	1	0	0	0	1		1
x	x	1	0	1	0		1
x	x	x	1	1	1		1

		For Y_1				
		00	01	11	10	
D ₀	D ₁	00	1 0	1 1	1 3	1 2
00	01	1 4	1 5	1 7	1 6	
01	11	1 12	1 13	1 15	1 14	
11	10	1 8	1 9	1 11	1 10	

$$Y_1 = D_2 + D_3$$

		For Y_0				
		00	01	11	10	
D ₀	D ₁	00	0 0	1 1	1 3	2
00	01	1 1	1 4	1 5	1 7	6
01	11	1 1	1 12	1 13	1 15	14
11	10	1 8	1 9	1 11	1 10	

$$\begin{matrix} 3 \\ \text{UNIT} \end{matrix}$$

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

		For V				
		00	01	11	10	
D ₀	D ₁	00	1 0	1 1	1 3	2
00	01	1 1	1 4	1 5	1 7	6
01	11	1 1	1 12	1 13	1 15	14
11	10	1 8	1 9	1 11	1 10	

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

Fig. 2.31.1.

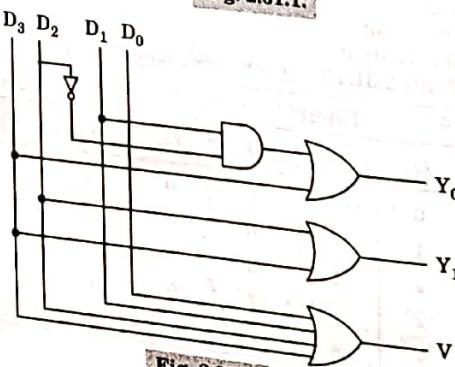


Fig. 2.31.2.



Sequential Logic and its Applications

Part-1 (3-2A to 3-19A)

- Storage Elements : Latches and Flip-Flops
- Characteristic Equations of Flip-Flops
- Flip-Flop Conversion

A. Concept Outline : Part-1 3-2A
B. Long and Medium Answer Type Questions 3-2A

Part-2 (3-19A to 3-37A)

- Shift Registers
- Ripple Counters
- Synchronous Counters
- Other Counters : Johnson and Ring Counter

A. Concept Outline : Part-2 3-19A
B. Long and Medium Answer Type Questions 3-20A

PART-1

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

CONCEPT OUTLINE : PART-1

- Sequential circuits :** It consists of a combinational circuit to which storage elements are connected to form a feedback path.
- Latches :** Storage elements that operate with signal levels (rather than signal transitions) are referred to as latches. Latches are said to be level sensitive (triggered) devices.
- Flip-flops :** Storage elements that are controlled by a clock transition are referred to as flip-flops. It is a binary storage device capable of storing one bit of information. Flip-flops are edge sensitive (triggered) devices.
- Characteristic equations of flip-flops :**

$$1. SR \text{ flip-flop} : Q_{n+1} = S + \bar{R} Q_n$$

$$2. D \text{ flip-flop} : Q_{n+1} = D$$

$$3. T \text{ flip-flop} : Q_{n+1} = \bar{T}Q_n + T\bar{Q}_n$$

$$4. JK \text{ flip-flop} : Q_{n+1} = J\bar{Q}_n + \bar{K}Q_n$$

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

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.
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.

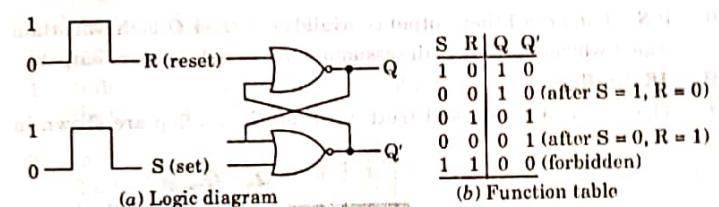


Fig. 3.1.1

Que 3.2. Explain the mechanism that a flip-flop holds a one bit of information. Describe the operation and working of the following flip-flops:

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

OR

What is flip-flop? Draw the logic diagram and give the characteristic table of JK flip-flop.

Answer

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 *Q'*).

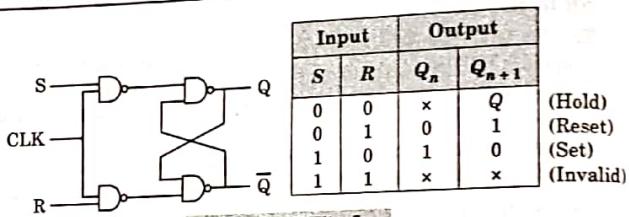


Fig. 3.2.1. SR Flip-flop.

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.

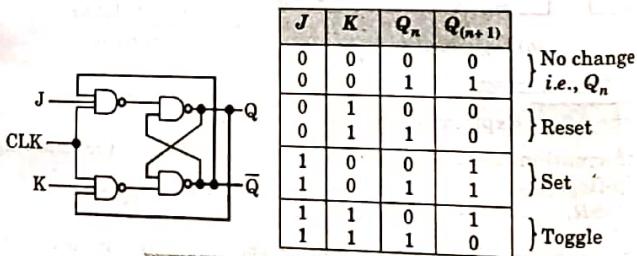


Fig. 3.2.2. JK flip-flop.

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 input.
5. After a time interval Δt equal to propagation delay through two NAND gates in series. The output will oscillate between 0 and 1.
6. At the end of CLK the output is uncertain and the condition is race-around condition. There are two methods to avoid race-around condition by using :

- Master Slave JK flip-flop.
- Edge-triggered flip-flop.

iii. T flip-flop:

- 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

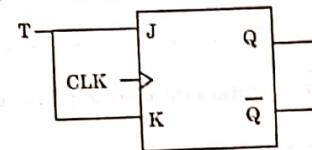


Fig. 3.2.3.

- 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:

- D flip flop is also known as delay flip-flop because the output follows the input after a clock pulse.
- In JK flip-flop when $J = \bar{K}$ then the resulting flip-flop acts as D 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

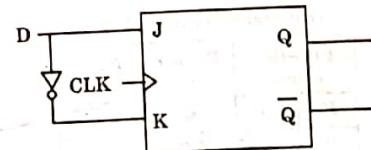


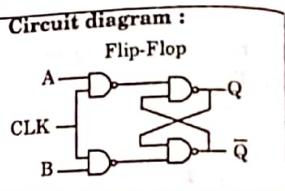
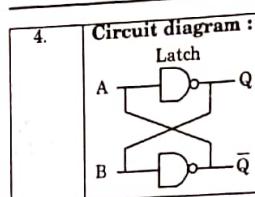
Fig. 3.2.4.

- 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.	Storage element that operate with signal levels.	Storage elements that are controlled by clock transitions.
2.	It is level triggered.	It is edge triggered.
3.	There is no clock pulse.	There is a clock pulse.



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

Answer

Characteristic equation :

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

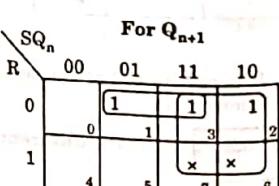
For SR flip-flop :

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

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

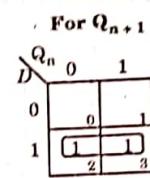


Fig. 3.4.2.

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.

Answer

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

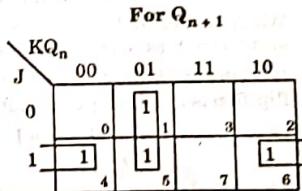


Fig. 3.5.1.

The characteristic equation of JK flip-flop is :

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

For T flip-flop : The truth table and K-map for T-flip-flop are shown in Fig. 3.5.2.

The excitation table of all the above flip-flops is as follows :

Present State (Q_n)	Next State (Q_{n+1})	S	R	J	K	T	D
0	0	0	x	0	x	0	0
0	1	1	0	1	x	1	1
1	0	0	1	x	1	1	0
1	1	x	0	x	0	0	1

Que 3.8. Explain how will you convert SR flip-flop into D flip-flop?

Answer

SR flip-flop to D flip-flop :

Truth table for D flip-flop

Inputs		States	
D	Q_n	Q_{n+1}	
0	0	0	
1	0	1	
0	1	0	
1	1	1	

Excitation table for SR flip-flop

Present state		Next state		Flip-flop Inputs	
Q_n	Q_{n+1}	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	Flip-flop inputs
D	Q_n	Q_{n+1}	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 :

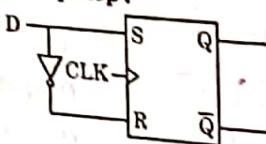


Fig. 3.8.2.

Que 3.9. Explain JK flip-flop with preset and clear. Also draw logic circuit of SR flip-flop using T flip-flop.

OR
Explain race around condition and its remedy in brief. Realise T flip-flop to SR flip-flop.

AKTU 2014-15, Marks 06

Answer

Race around condition : Refer Q. 3.6, Page 3-8A, Unit-3.

JK flip-flop with preset and clear :

Preset and clear inputs are called asynchronous or direct inputs. These inputs are connected directly into the latch portion of the flip-flop so that they override the effect of the synchronous inputs J, K and clock (CLK).

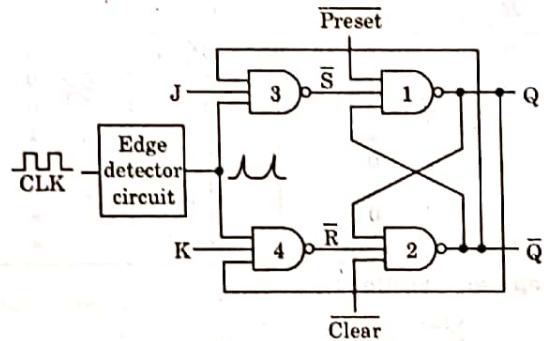


Fig. 3.9.1. JK flip-flop with clear and preset

- If preset and clear inputs are 1, circuit operates as JK flip-flop. If preset = 0 and clear = 1, output of NAND gate 1 will certainly be 1. Consequently, all the three inputs of NAND gate 2 will be 1 which will make $\bar{Q} = 0$. Hence, preset = 0 sets the flip-flop. Preset is active low signal. Similarly, low (0) on the clear input resets the flip-flop making $\bar{Q} = 1$.
- T flip-flop to SR flip-flop :

Step 1 : Truth table for SR flip flop :

S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	x

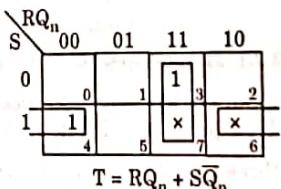
Step 2 : Excitation table for T flip-flop :

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

Step 3 : Conversion table :

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

K-map simplification :



$$T = RQ_n + S \bar{Q}_n$$

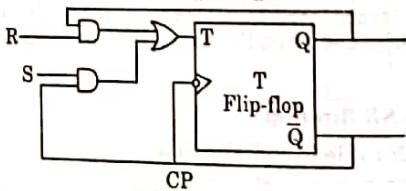


Fig. 3.9.2.

Que 3.10. Draw JK flip-flop and write the characteristic table & characteristic equation for it. Explain how will you convert it in T flip-flop ?

AKTU 2012-13, Marks 1

Answer

JK flip-flop : Refer Q. 3.2, Page 3-3A, Unit-3.

Conversion 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

For J_A	
0	0
1	x

For K_A	
0	0
1	x

Logic diagram :

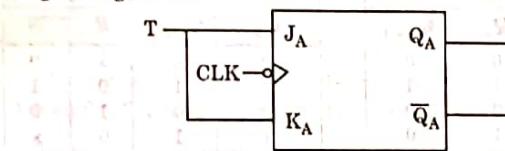


Fig. 3.10.2.

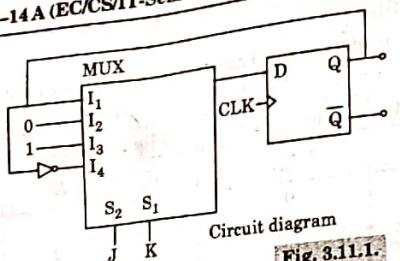
Que 3.11. Construct a JK flip-flop, using a D flip-flop, a two to four one line multiplexer and an inverter.

AKTU 2016-17, Marks 10

Answer

Note : Since a two to four one line multiplexer is not possible hence the solution is given by assuming four to one line multiplexer.

1. The circuit diagram of a JK flip-flop constructed with a D flip-flop and gates is shown in Fig. 3.11.1.
2. The J input sets the flip-flop to 1, the K input resets it to 0, and when both inputs are enabled, the output is complemented.
3. When $J = 0, K = 0, D = Q = Q_n$
4. When $J = 0, K = 1, 0$ line is selected, the output $Q = 0$.



Truth table		
J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	\bar{Q}_n

Fig. 3.11.1.

5. When $J = 1, K = 0, 1$ is selected, the output $Q = 1$.
 6. When $J = 1, K = 1, D = \bar{Q}_n$.

Que 3.12 Convert the SR flip-flop to JK flip-flop.

Answer

1. 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	R	S
0	0	x	0	0	0
1	1	0	1	1	1
0	0	1	0	0	1
1	1	0	x	1	x

2. Form the conversion table

Required flip-flop			Given flip-flop		
Flip-flop inputs	Present state	Next state	Flip-flop inputs	Present state	Next state
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	0	0
0	1	1	0	x	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:

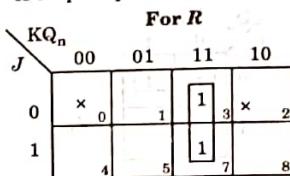


Fig. 3.12.1.

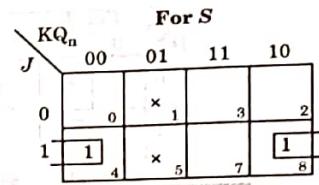


Fig. 3.12.2.

4. The obtained boolean expression:

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

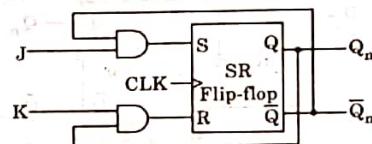


Fig. 3.12.3.

Que 3.13 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

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

2. Form the conversion table

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

3. K-map simplification:

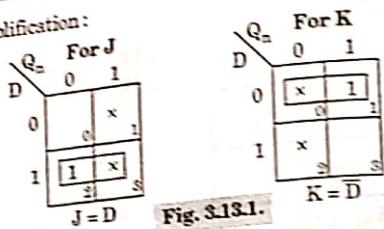


Fig. 3.13.1.

$$J = D \text{ and } K = \bar{D}$$

Expressions are

4. Logic diagram:

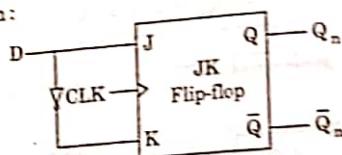


Fig. 3.13.2.

Que 3.14. 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

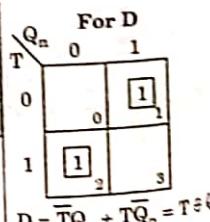
Excitation table of D flip-flop

Flip-flop inputs	Present state	Next state
D	Q _n	Q _{n+1}
0	0	0
0	1	1
1	0	0
1	1	1

2. From 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. 3.14.1.



$$D = \overline{T}Q_n + T\overline{Q}_n = T \oplus Q_n$$

4. Logic diagram:

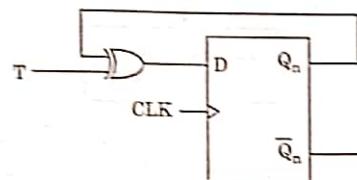


Fig. 3.14.2.

Que 3.15. What is ASM chart? Explain. Draw the state diagram, state table and ASM chart for a D flip-flop.

AKTU 2013-14, Marks 10

Answer

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

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

General description entry

State name ↓ Binary code

Register operation or unconditional list

Exit

Fig. 3.15.1.

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

Entry

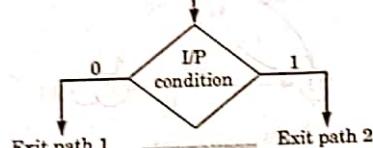
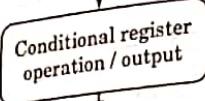


Fig. 3.15.2.

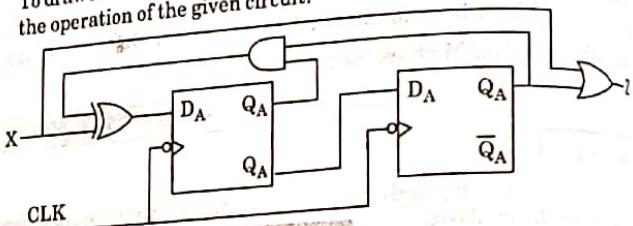
Conditional box : It is a unique box of ASM chart. The area shape of the conditional box is shown in Fig. 3.15.3. The round corners differentiate it from the state box.

Entry path from decision box
↓



Exit
Fig. 3.15.3

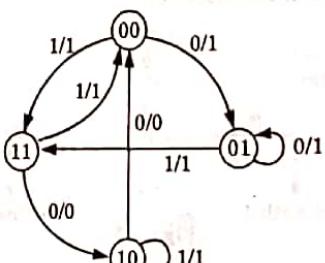
To draw ASM chart, first we form a state table for easy understanding the operation of the given circuit.

**Fig. 3.15.4.**

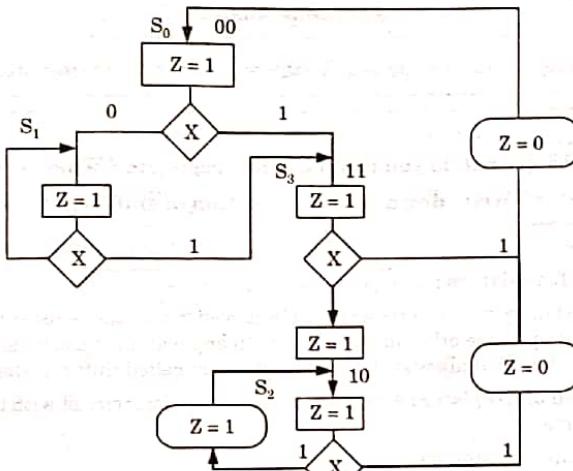
State table :

Present state	Next state, output						
	X = 0		X = 1				
Q _A	Q _B	Q _{A+1}	Q _{B+1}	Z	Q _{A+1}	Q _{B+1}	Z
0	0	0	1	1	1	1	1
0	1	0	1	1	1	1	1
1	0	0	0	0	1	0	1
1	1	1	0	0	0	0	1

State diagram :

**Fig. 3.15.5.**

ASM chart :

**Fig. 3.15.6.****PART-2**

Shift Register, Ripple Counter, Synchronous Counter, Other Counters : Johnson and Ring Counter.

CONCEPT OUTLINE : PART-2

- **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.
- **Modes of operation of a shift register :**
 1. Serial in serial out (SISO)
 2. Serial in parallel out (SIPO)
 3. Parallel in serial out (PISO)
 4. Parallel in parallel out (PIPO)
- **Ripple counter (asynchronous counter) :** For these counters the external clock signal is applied to one flip-flop and then output of preceding flip-flops is connected to the clock of next flip-flop.
- **Synchronous counter :** In these counters all the flip-flops receive the external clock pulse simultaneously.

Questions-Answers
Long Answer Type and Medium Answer Type Questions

Que 3.16. What do you mean by shift registers? What is the need of register? Write down the classification of shift registers.

Answer**Shift registers :**

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

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

- Temporary storage.
- Shifting capability.

Classification of shift registers :

- Classification based on the direction of data movement :
 - Shift left register.
 - Shift right register.
 - Bidirectional shift registers.
- 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.

Que 3.17. Write a short note on different types of shift register.

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

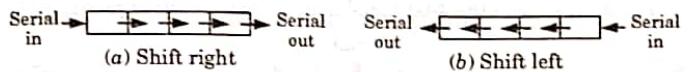


Fig. 3.17.1. 4-bit serial in serial out shift register.

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. 3.17.2 shows the logic circuit for a 4-bit shift right register.

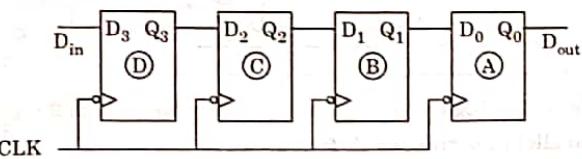


Fig. 3.17.2. Logic circuit for a 4-bit shift right register.

Shift left register :

- 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.
- Fig. 3.17.3 shows the logic circuit for a 4-bit shift left register.

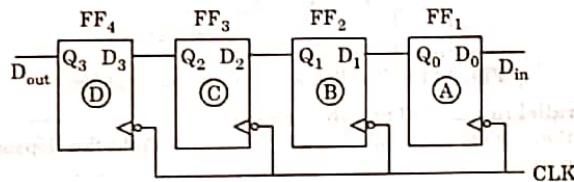


Fig. 3.17.3. Logic circuit for a 4-bit shift left register.

- The binary data is entered into right most flip-flop (FF4) and output is taken from the left most flip-flop (FF1) in serial form.

Serial in parallel out shift register :

- This is one type of shift register in which the data is entered in serial form and output is in parallel form.
- Hence, it is necessary to have all the data bits available as outputs at the same time.
- This type of shift register operation is same as the serial in serial out shift register.
- 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. 3.17.4 shows the 4-bit serial-to-parallel converter.

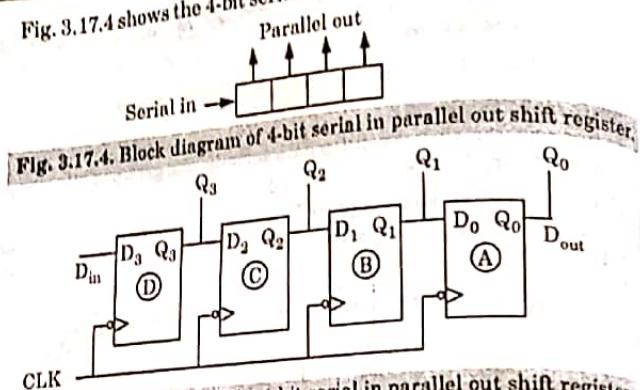


Fig. 9.17.5. Logic circuit for 4-bit serial in parallel out shift register.

Parallel in serial out shift register:

- Fig. 3.17.6 shows the block diagram of a parallel in serial out shift register. In this type, the bits are entered in parallel, i.e., simultaneously in their respective stages on a parallel line.
 - It produces the stored information on its output, in serial form.

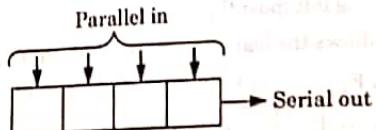


Fig. 3.17.6. Parallel in serial out shift register.

Parallel in parallel out shift register :

1. All the data appear simultaneously along with all the flip-flop inputs outputs.
 2. Fig. 3.17.7 shows the logic diagram for 4-bit parallel in parallel out register.

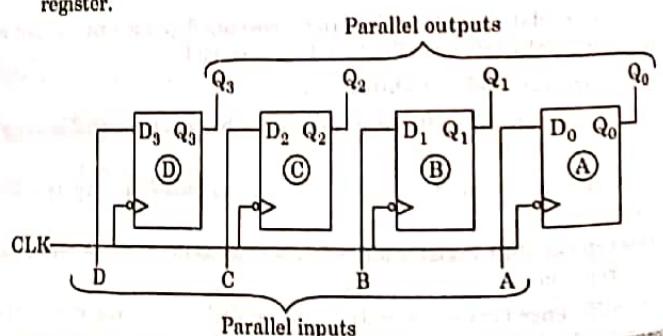


Fig. 3.17.7. Logic diagram for 4-bit parallel in parallel out shift register

Digital Logic Design

- Ques 3.18.** With the help of diagram, explain the operation of universal shift register.

AKTU 2014-15 Marks 06

OR
Draw and explain 4-bit universal shift register.

AKTU 2015-16 Marks 10

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. 3.18.1. It consists of four D flip-flop and four $4 : 1$ multiplexers.
 4. The four multiplexers have two common select lines S_1 and S_0 . Input I_0 in each multiplexer is selected when $S_1S_0 = 00$, input I_1 is selected when $S_1S_0 = 01$ and so on.
 5. The selection inputs (S_1S_0) control the mode of operation of the register according to the function table shown in table 3.18.1.

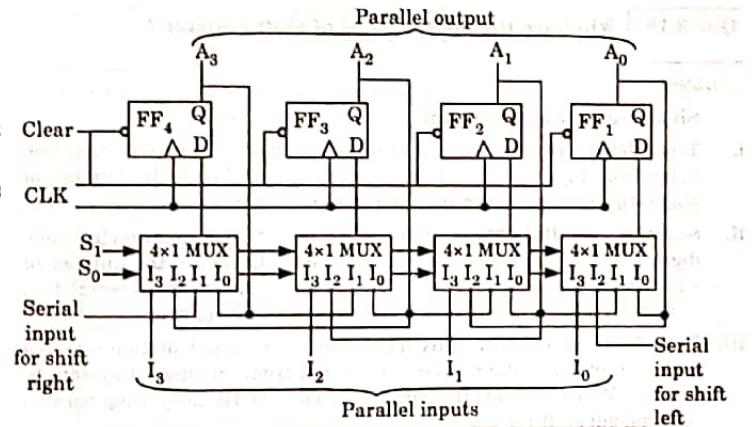


Fig. 3.18.1. Block diagram of 4-bit universal shift register.

Table 3.18.1. 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 inputs of the flip-flops. This condition forms a path from the outputs of each flip-flop to the binary value input of the same flip-flops. The 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 inputs of the flip-flops. This causes a shift right operation, with the serial input transferred in flip-flop FF_1 .
- When $S_1S_0 = 10$, a shift left operation results, with the other 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 the side for every clock.
- 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 3.19. What are the applications of shift register ?

Answer**Shift register applications :**

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

Que 3.20. Design a 4-bit serial in serial out shift register using D flip-flop.

Answer

Serial in-serial out shift register using D flip-flop :

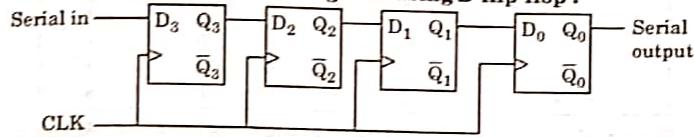
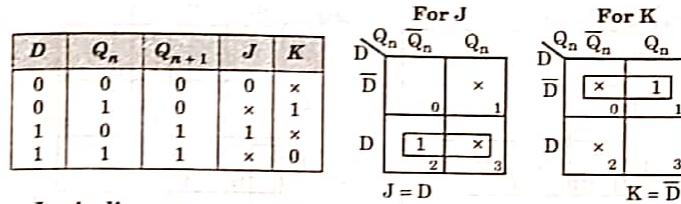


Fig. 3.20.1.

Conversion table and K-map :



Logic diagram :

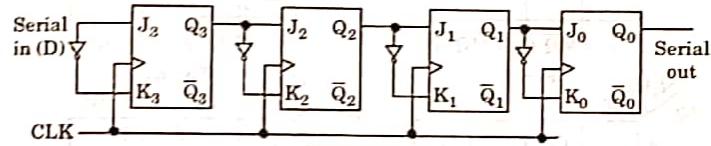


Fig. 3.20.2.

Que 3.21. Design a 3-bit binary up/down counter with a direction control M. Use JK flip-flops.

Answer

- The counter sequence for 3-bit binary up/down counter is :

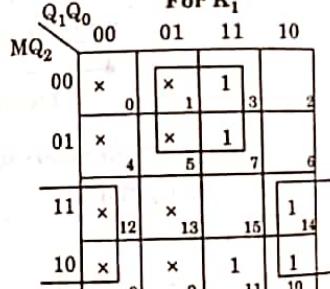
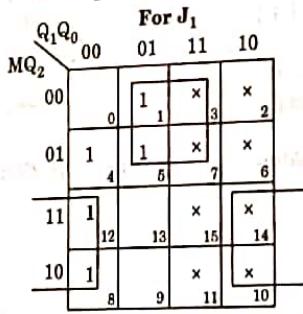
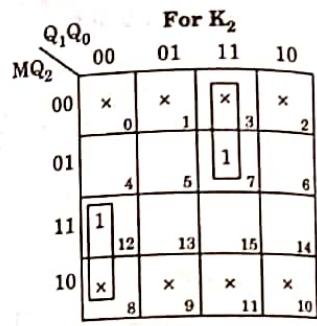
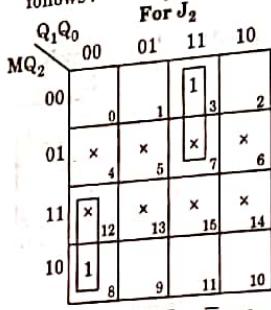
Direction control	Counter state			Flip-flop inputs									
	J_2	K_2	J_1	K_1	J_0	K_0	J_2	K_2	J_1	K_1	J_0	K_0	
0	0	0	0	0	0	x	0	x	0	x	1	x	
0	0	0	1	0	0	x	0	x	1	x	x	1	
0	0	1	0	0	0	x	0	x	x	0	1	x	
0	0	1	1	1	1	x	x	x	x	1	x	1	
0	1	0	0	0	x	0	0	0	x	1	x	x	
0	1	0	1	0	x	0	1	x	0	x	x	1	
0	1	1	0	0	x	0	0	x	0	1	x	1	
0	1	1	1	1	x	1	x	1	x	1	x	1	
1	0	0	0	0	1	x	1	x	1	x	1	x	

Sequential Logic & its Applications

3-26 A (EC/CS/IT-Sem-3)

	0	0	1	0	x	0	x	x	1
1	0	1	0	0	x	1	1	x	1
1	0	1	1	0	x	0	x	1	x
1	0	1	0	x	1	1	x	1	x
1	1	0	1	x	0	0	x	x	1
1	1	0	0	x	0	x	1	1	x
1	1	1	0	x	0	x	1	1	x
1	1	1	1	x	0	x	0	x	1

2. From table, we obtain, $J_0 = K_0 = 1$. K-maps, for J_1, K_1, J_2, K_2 are follows:



3. The minimized expressions are :

$$J_1 = K_1 = Q_0\bar{M} + \bar{Q}_0M$$

$$J_2 = K_2 = \bar{M}Q_1Q_0 + M\bar{Q}_1\bar{Q}_0$$

Que 3.22. Design a 3-bit asynchronous up-down counter using T flip-flop.

AKTU 2015-16, Marks 7

Digital Logic Design

3-27 A (EC/CS/IT-Sem-3)

Answer

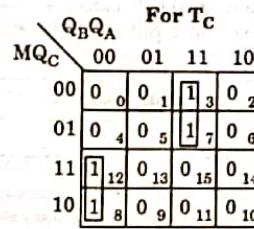
1. The number of flip-flop to be used is three. We shall use three toggle flip-flop. Let, up counting takes place with $M = 0$ and down counting take place for $M = 1$.

2. First, we write the circuit excitation table.

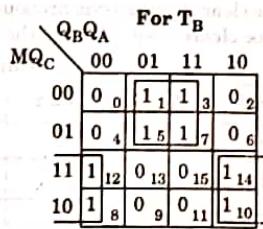
Table 3.22.1 : Excitation table for a 3-bit up/down synchronous counter

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

3. Next, let us obtain the K-maps and simplified expressions. The K-maps and simplified expressions for T_C, T_B and T_A have shown in Fig. 3.22.1.



$$T_C = \bar{M}Q_B Q_A + MQ_B \bar{Q}_A$$



$$T_B = \bar{M}Q_A + \bar{Q}_A M$$

$$= M \oplus Q_A$$

		For T_A			
		00	01	11	10
MQC	00	1 ₀	1 ₁	1 ₃	1 ₂
	01	1 ₄	1 ₅	1 ₇	1 ₆
	11	1 ₁₂	1 ₁₃	1 ₁₅	1 ₁₄
	10	1 ₈	1 ₉	1 ₁₁	1 ₁₀

Fig. 3.22.1. K-maps and simplified expressions.

4. Finally, let us draw the logic diagram.

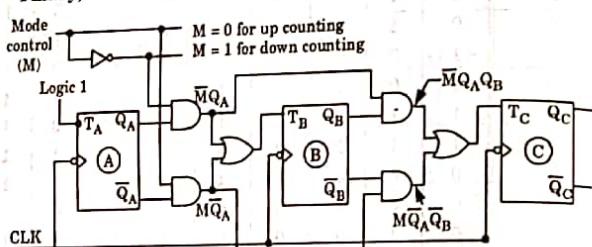


Fig. 3.22.2. Logic diagram of a 3-bit synchronous up/down counter.

Que 3.23. Draw the logic diagram of a 4-bit binary counter with parallel load.

Answer

- The operation of the counter is summarized in Table 3.23.1. The four control inputs—Clear, CLK, load, and count, determine the next state.
- The clear input is asynchronous and when equal to 0, causes the counter to be cleared regardless of the presence of clock pulses or other inputs.

Table 3.23.1.

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

- The logic diagram for 4-bit binary counter with parallel load is shown in Fig. 3.23.1.

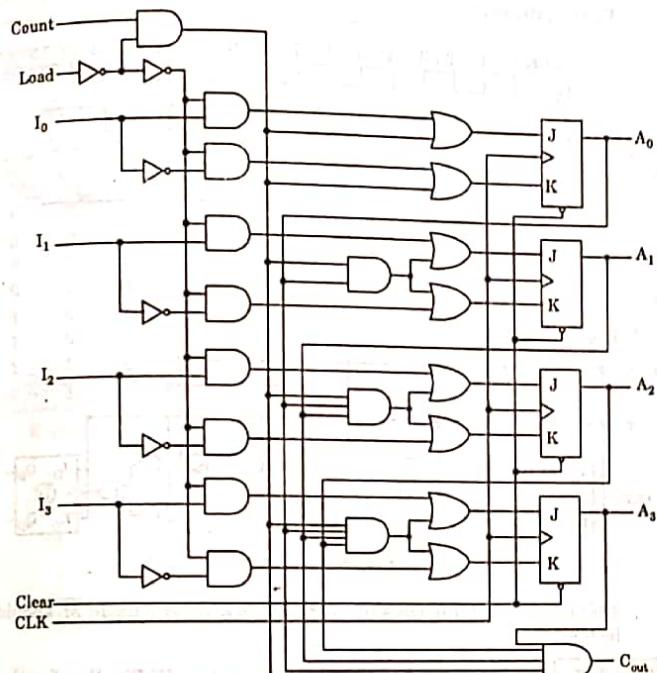


Fig. 3.23.1.

Que 3.24. 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.

Answer

4-bit binary ripple down counter:

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

Timing diagram :

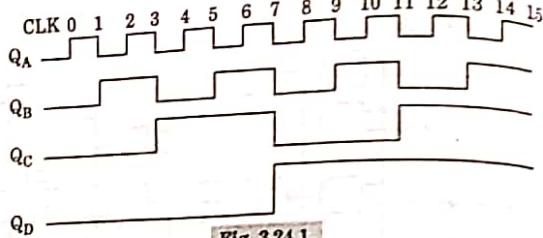


Fig. 3.24.1.

Logic diagram :

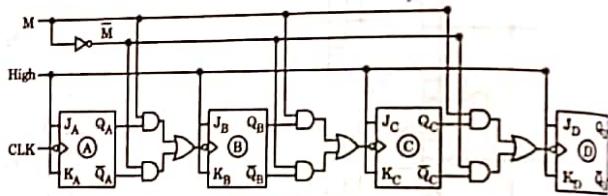


Fig. 3.24.2.

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

Que 3.25. Design a synchronous counter using JK flip-flop for the following input sequences :

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



Fig. 3.25.1.

Answer

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

K-maps:

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

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

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

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

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

Fig. 3.25.1.

Flip-flop required are : $2^n \geq N$ Here $N = 6$ So, $n = 3$, i.e., three flip-flops are required.

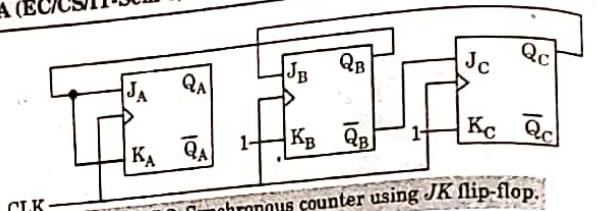


Fig. 3.25.2. Synchronous counter using JK flip-flop.

Que 3.26. Design a 4-bit binary synchronous counter with T flip-flop.

Answer

Design of 4-bit binary synchronous counter with T flip-flop :

Present state				Next state				Flip-flop inputs		
A	B	C	D	A	B	C	D	T _A	T _B	T _C
0	0	0	0	0	0	0	0	1	0	0
0	0	0	1	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1	0	0	0
0	0	1	1	0	1	0	0	0	0	1
0	1	0	0	0	1	0	1	0	0	0
0	1	0	1	0	1	1	0	0	0	1
0	1	1	0	0	1	1	1	0	0	0
0	1	1	1	1	0	0	0	1	1	1
1	0	0	0	1	0	0	1	0	0	0
1	0	0	1	1	0	1	0	0	0	1
1	0	1	0	1	0	1	1	0	0	0
1	0	1	1	1	1	0	0	0	1	1
1	1	0	0	1	1	1	0	0	0	0
1	1	0	1	1	1	1	1	0	0	1
1	1	1	0	1	1	1	1	0	0	0
1	1	1	1	0	0	0	0	1	1	1

K-map simplification :

For T _A				
CD	00	01	11	10
AB	00	01	11	2
00	0	1	3	2
01	4	5	1	7
11	12	13	15	14
10	8	9	11	10

For T _B				
CD	00	01	11	10
AB	00	01	11	2
00	0	1	1	3
01	4	5	1	7
11	12	13	15	14
10	8	9	11	10

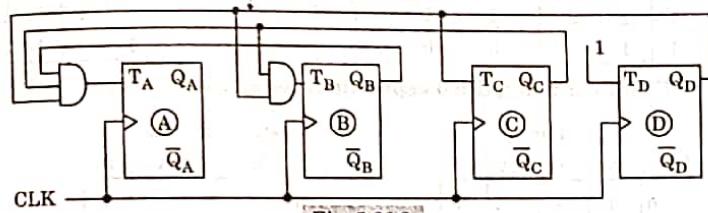
Fig. 3.26.1.

T_A = BCDT_B = CD

For T _C				
CD	00	01	11	10
AB	00	11	13	2
00	0	1	1	3
01	4	5	1	7
11	12	13	15	14
10	8	9	11	10

For T _D				
CD	00	01	11	10
AB	1	1	1	2
00	1	0	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

Fig. 3.26.2.

T_D = 1

Que 3.27. Design a 3-bit synchronous counter using JK flip-flops.

AKTU 2012-13, Marks 10

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

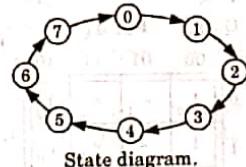


Fig. 3.27.1.

Excitation table for 3-bit synchronous counter :

Table 3.27.2: 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	0	0	x	1
1	0	0	1	1	0	x	0	1	x	x	1
1	0	1	1	1	1	x	0	x	0	1	x
1	1	0	1	1	0	x	1	x	1	x	1
1	1	1	0	0	0	x	1	x	1	x	1

K-maps and simplified expressions for all FF inputs :

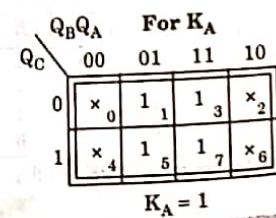
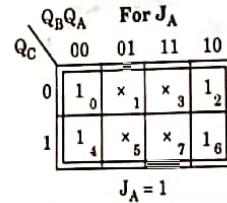
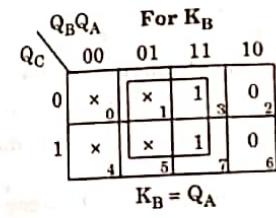
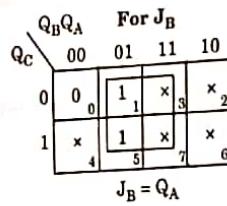
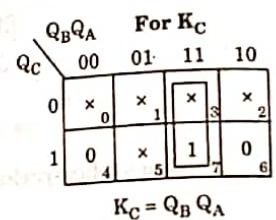
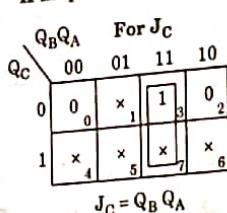


Fig. 3.27.2. K-maps and simplified equations for different FF inputs.

Thus the simplified equations are :

$$\begin{aligned} J_C &= Q_B Q_A & K_C &= Q_B Q_A \\ J_B &= Q_A & K_B &= Q_A \\ J_A &= 1 & K_A &= 1 \end{aligned}$$

Logic diagram :

Fig. 3.27.3 shows the logic diagram of a 3-bit synchronous counter using JK flip-flops.

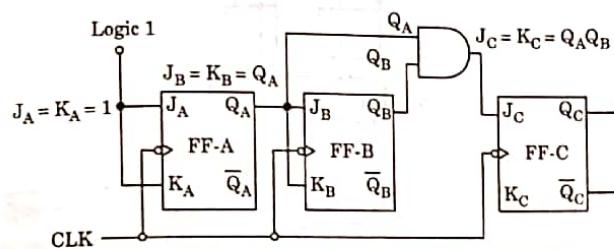


Fig. 3.27.3. 3-bit synchronous counter using JK FFs.

Que 3.28. Describe the operation of four bit synchronous binary counter with neat sketch.

AKTU 2016-17, Marks 10

Answer

4-bit synchronous binary counter :

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



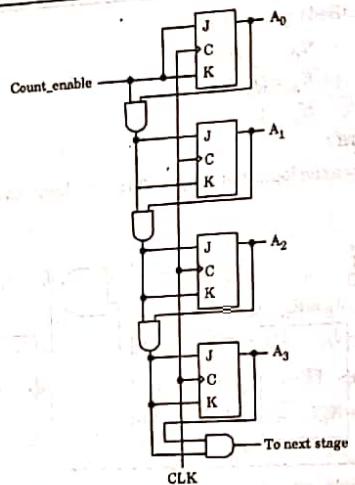
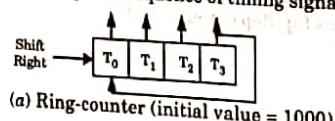


Fig. 3.28.1. Four-bit synchronous binary counter.

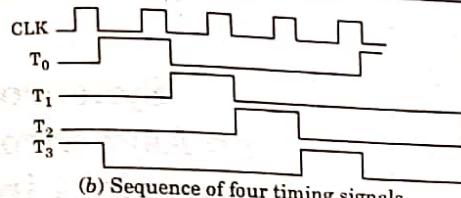
Que 3.29. Explain briefly ring counter.

Answer

1. A ring counter is a circular shift register with only one flip-flop being set at any particular time, all others are cleared.
2. The single bit is shifted from one flip-flop to the next to produce the sequence of timing signals. Fig. 3.29.1(a) shows a four-bit shift register connected as a ring counter.
3. The initial value of the register is 1000 and requires preset/clear flip-flops. The single bit is shifted right with every clock pulse and circulates back from T_3 to T_0 .
4. Each flip-flop is in the 1 state once every four clock cycles and produces one of the four timing signals shown in Fig. 3.29.1(b).
5. Each output becomes a 1 after the negative edge transition of a clock pulse and remains 1 during the next clock cycle.
6. The timing signals can also be generated by two bit counter that goes through four distinct states.
7. The decoder shown in Fig. 3.29.1(c) decodes the four states of the counter and generates the required sequence of timing signals.



(a) Ring-counter (initial value = 1000)



(b) Sequence of four timing signals

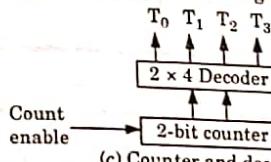


Fig. 3.29.1. Generation of timing signals.

Que 3.30. Write a short note on Johnson counter.

Answer

1. Johnson counter is similar to the ring counter except that the input of the first flip-flop, i.e., D is driven by E' output of the last flip-flop.
2. This feedback arrangement produces a sequence of states as shown in Fig. 3.30.1.
3. The 4-bit sequence has total 8-states. The n -bit counter (Johnson counter) will have modulo of 2^n .

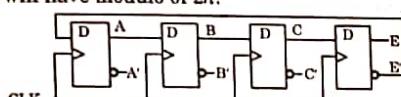


Fig. 3.30.1. Four-stage switch-tail ring counter/Johnson counter.

Sequence number	Flip-flop outputs			
	A	B	C	E
1	0	0	0	0
2	1	0	0	0
3	1	1	0	0
4	1	1	1	0
5	1	1	1	1
6	0	1	1	1
7	0	0	1	1
8	0	0	0	1

10-ans-1 ☺☺☺

4

UNIT

Synchronous and Asynchronous Sequential Circuits

Part-1 (4-2A to 4-18A)

- Analysis of Clocked Sequential Circuits with State Machine Designing
- State Reduction and Assignments
- Design Procedure

A. Concept Outline : Part-1 4-2A
B. Long and Medium Answer Type Questions 4-2A

Part-2 (4-18A to 4-51A)

- Analysis Procedure of Asynchronous Sequential Circuits
- Circuit with Latches
- Design Procedure
- Reduction of State and Flow Table
- Race Free State Assignment
- Hazards

A. Concept Outline : Part-2 4-18A
B. Long and Medium Answer Type Questions 4-19A

4-1A (EC/CS/IT-Sem-3)

4-2A (EC/CS/IT-Sem-3)

Synchronous & Asynchronous Circuits

PART-1

Analysis of Clocked Sequential Circuit with State Machine Designing, State Reduction and Assignment, Design Procedure.

CONCEPT OUTLINE : PART-1

- Sequential circuits : It consists of a combinational circuit to which storage elements are connected to form a feedback path.
- State reduction : The reduction in the number of flip-flops in a sequential circuit is referred to as the state reduction.
- State assignment : In order to design a sequential circuit with physical components, it is necessary to assign unique coded binary values to the states.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 4.1. Discuss the analysis of clocked sequential circuits.

Answer

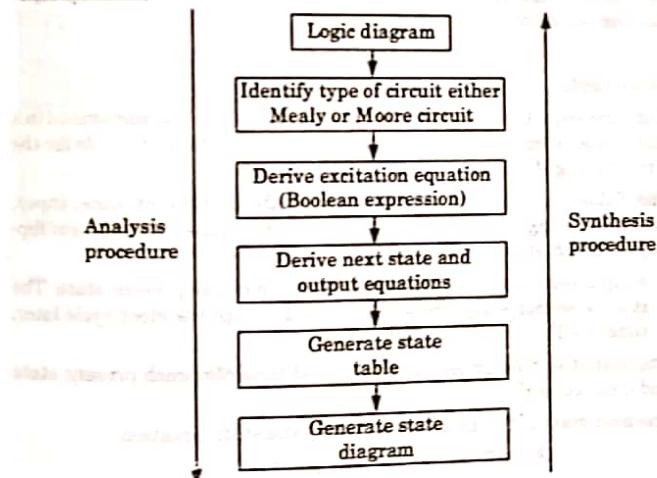


Fig. 4.1.1. Analysis and synthesis procedure of sequential circuits.

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

Que 4.2. Discuss the concept of state equation, state table and state diagram in clocked sequential circuit.

Answer

State equations :

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

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

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

- 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 :

- The time sequence of inputs and flip-flop states can be enumerated in 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.
- 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 .
- The input section gives a value of x for each possible present state. The next state section shows the states of the flip-flops one clock cycle later at time $(t+1)$.
- The output section gives the value of y at time t for each present state and input condition.
- The next state of flip-flop A must satisfy the state equation

$$A(t+1) = Ax + Bx'$$

Table. 4.2.1.

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

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

$$B(t+1) = Ax'$$

- The output column is derived from the output equation

$$y = Ax' + Bx$$

State diagram :

- The information available in a state table can be represented graphically in the form of a state diagram.
- 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	y	y
0	0	0	0	0	0
0	1	0	0	1	0
1	0	0	0	1	0
1	1	0	0	1	0

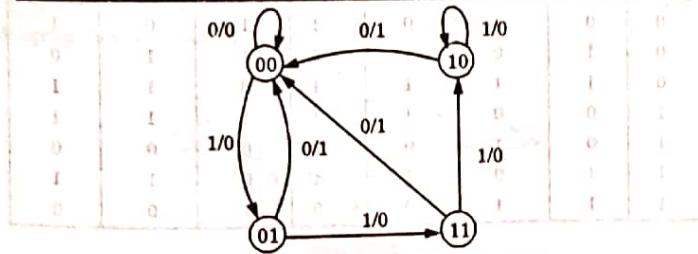


Fig. 4.2.1. State diagram of the circuit.

Digital Logic Design

- 4-5 A (EC/CS/IT-Sem-3)
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.** Design one input and one output sequential circuit using D flip-flop for the state diagram shown in Fig. 4.3.1.

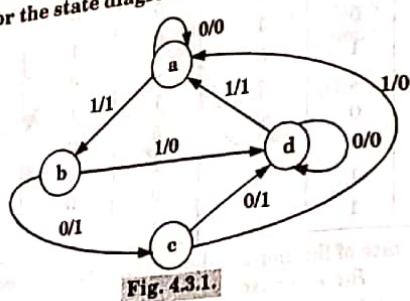


Fig. 4.3.1.

Answer

Present state	Next state		Output	
	X = 0	X = 1	X = 0	X = 1
a	a	b	0	1
b	c	d	1	0
c	d	a	1	0
d	d	a	0	1

Let $a = 00$, $b = 01$, $c = 10$, $d = 11$

Excitation table :

Present state	Input	Next state		Output	Flip-flop input	
		A	B		D _A	D _B
00	0	0	0	0	0	0
00	1	0	1	1	0	1
01	0	1	0	1	1	0
01	1	1	1	0	1	1
10	0	0	1	1	1	1
10	1	0	0	0	0	0
11	0	1	1	0	1	1
11	1	0	0	1	0	0

4-5 A (EC/CS/IT-Sem-3)

4-6 A (EC/CS/IT-Sem-3)

Synchronous & Asynchronous Circuits

BX	For D _A			
	00	01	11	10
0	0	1	1	2
1	1	5	7	6

$$D_A = \overline{AX} + \overline{AB}$$

BX	For D _B			
	00	01	11	10
0	0	1	1	2
1	1	5	7	6

$$D_B = \overline{AX} + \overline{AX} = A \oplus X$$

Fig. 4.3.2.

For output Y :

BX	Y			
	00	01	11	10
0	0	1	3	1
1	1	4	5	6

$$Y = \overline{ABX} + \overline{ABX} + ABX + ABX = A \oplus B \oplus X$$

Fig. 4.3.3.

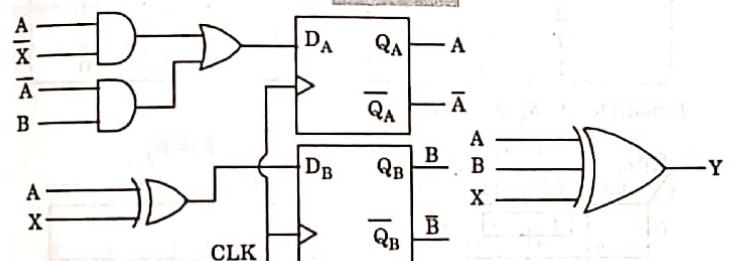


Fig. 4.3.4.

- Que 4.4.** Design the clocked sequential circuit whose state diagram is given in Fig. 4.4.1 using JK flip-flop.

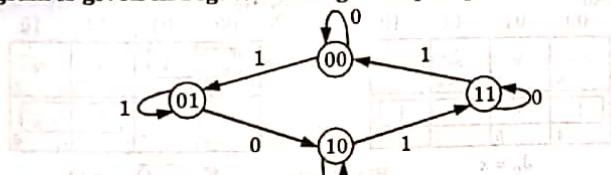


Fig. 4.4.1.

AKTU 2013-14, Marks 10

AKTU 2015-16, Marks 15

4-7 A (EC/CS/IT-Sem-3)

Answer

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

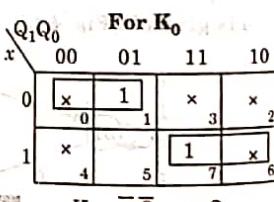
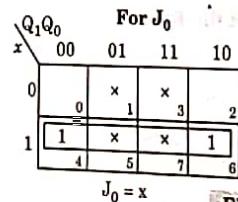
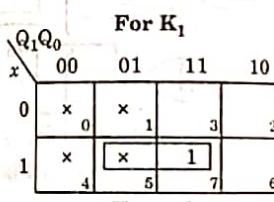
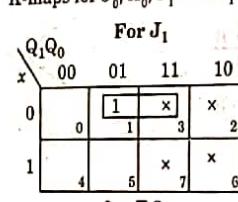
Input	Present state		Next state		Flip-flop inputs			
	Q_1	Q_0	Q_1	Q_0	J_1	K_1	J_0	K_0
0	0	0	0	0	0	x	0	x
0	0	1	1	0	1	x	0	1
0	1	0	1	1	x	0	x	0
0	1	1	0	1	0	x	1	x
1	0	0	1	0	x	0	1	0
1	0	1	0	1	x	1	x	0
1	1	0	1	1	x	1	x	1
1	1	1	0	0	x	1	0	0

Columns of J_1, K_1, J_0, K_0 are filled by the help of excitation table of 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_0, K_0, J_1 and K_1 are :



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

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

4-8 A (EC/CS/IT-Sem-3)

Synchronous & Asynchronous Circuits

$$J_0 = x$$

$$K_1 = x Q_0$$

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

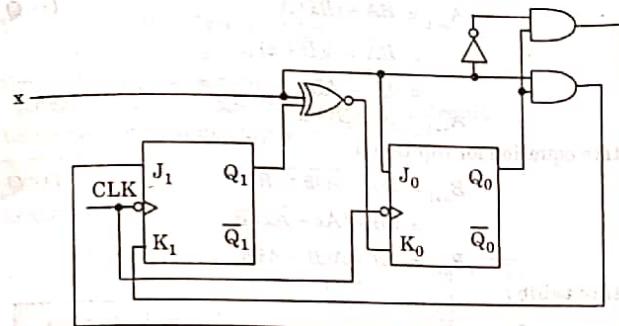


Fig. 4.4.3.

Que 4.5. Derive the state table and state diagram for the sequential circuit shown in Fig. 4.5.1.

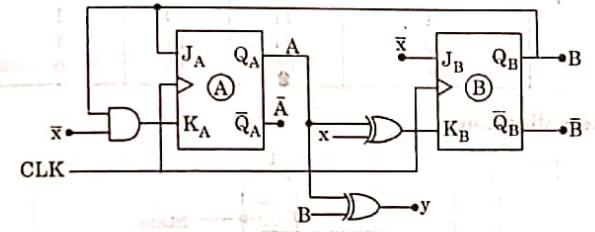


Fig. 4.5.1.

AKTU 2015-16, Marks 10

Answer

- Type of circuit :

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

- Excitation equations :

For flip-flop A : $J_A = B$

$K_A = B\bar{x}$

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

$K_B = A \oplus x$

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

Digital Logic Design

4-9 A (EC/CS/IT-Sem-3)

3. We know that characteristics equation of JK flip-flop :
 $A_{n+1} = J\bar{Q}_n + \bar{K}Q_n$

4. State equation for flip-flop A : $(\because Q_n = A)$

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

5. State equation for flip-flop B : $(\because Q_n = B)$

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

State table :

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

State diagram :

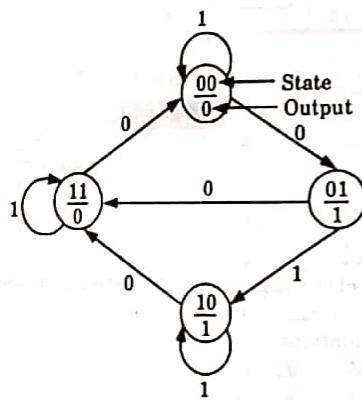


Fig. 4.5.2.

The state diagram for Moore machine is different from Mealy machine.
 Here each circle is coded with state binary number/output.

4-10 A (EC/CS/IT-Sem-3)

Synchronous & Asynchronous Circuits

Que 4.6. A sequential circuit with two D flip-flops A and B, two inputs x and y , and one output z , is specified by the following next state and output equations.

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

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

$$z = B$$

- Draw the logic diagram of the circuit.
- List the state table for the sequential circuit.
- Draw the corresponding state diagram.

Answer

i. Logic diagram :

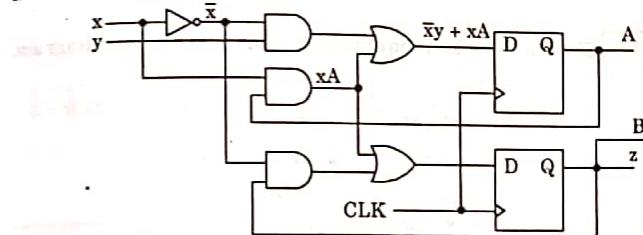


Fig. 4.6.1.

ii. State table :

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

iii. State diagram:

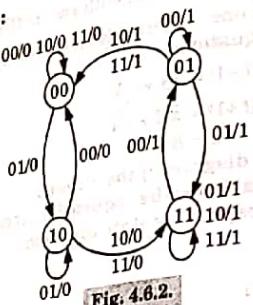


Fig. 4.6.2.

Que 4.7. Design a clocked sequential circuit for the state diagram.

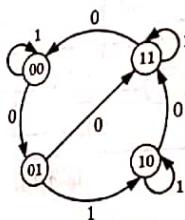


Fig. 4.7.1.

AKTU 2012-13, Marks 10

Answer

1. From the Fig. 4.7.1 we have

Present state		Input	Next state		Output	
A	B	X	A	B	D_A	D_B
0	0	0	0	1	0	1
0	0	1	0	0	0	0
0	1	0	1	1	1	1
0	1	1	1	0	1	0
1	0	0	1	1	1	1
1	0	1	1	0	1	0
1	1	0	0	0	0	0
1	1	1	1	1	1	1

2. K-maps:

For D_A

BX	00	01	11	10
A	0	0	1	1
1	1	1	1	0
$D_A = BX + \bar{A}\bar{B} + A\bar{B}$	0	1	1	2
$D_A = BX + A \oplus B$	4	5	7	6

For D_B

BX	00	01	11	10
A	0	1	0	1
1	1	1	1	0
$D_B = \bar{B}\bar{X} + \bar{A}\bar{X} + ABX$	1	0	3	2
$D_B = \bar{B}\bar{X} + \bar{A}\bar{X} + ABX$	4	5	7	6

Fig. 4.7.2.

3. Circuit diagram:

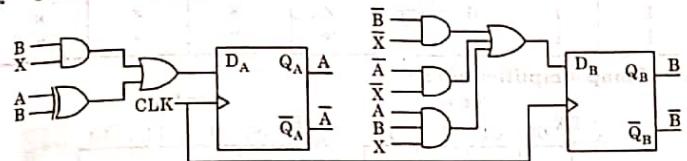


Fig. 4.7.3. Circuit diagram.

Que 4.8. Design a circuit that implements the state diagram.

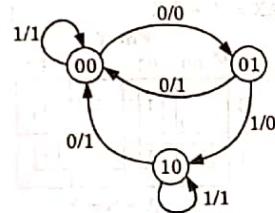


Fig. 4.8.1.

AKTU 2011-12, Marks 10

Answer

1. State table : From the given state diagram in Fig. 4.8.1.

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

As seen from the state table there is no equivalent state. Therefore there is no reduction in state diagram. To design the circuit we need two flip-flops.

2. Excitation table :

Present state	Input	Next state		Flip-flop input		Output
		A	B	D _A	D _B	
0	0	0	0	1	0	1
0	0	1	0	0	0	1
0	1	0	0	0	1	0
0	1	1	0	0	0	1
1	0	0	0	1	0	1
1	0	1	1	0	x	x
1	1	0	x	x	x	x
1	1	1	x	x	x	x

3. K-map simplifications :

$$\text{For } D_A \\ \begin{array}{c} \text{A} \\ \text{BX} \\ \begin{array}{cccc} 00 & 01 & 11 & 10 \end{array} \end{array} \quad \begin{array}{c} \text{A} \\ \text{BX} \\ \begin{array}{cccc} 00 & 01 & 11 & 10 \end{array} \end{array}$$

0	0	1	3	2
1	4	5	x	6
D _A = BX + AX = (A + BX)	0	1	3	2

$$\text{For } D_B \\ \begin{array}{c} \text{A} \\ \text{BX} \\ \begin{array}{cccc} 00 & 01 & 11 & 10 \end{array} \end{array}$$

0	1			
1	4	5	7	6
D _B = $\bar{A}\bar{B}X$	0	1	x	x

$$\text{For } Y \\ \begin{array}{c} \text{A} \\ \text{BX} \\ \begin{array}{cccc} 00 & 01 & 11 & 10 \end{array} \end{array}$$

0	0	1	3	2
1	1	x	x	6
Y = A + B \oplus X	0	1	3	2

Fig. 4.8.2.

4. Logic diagram :

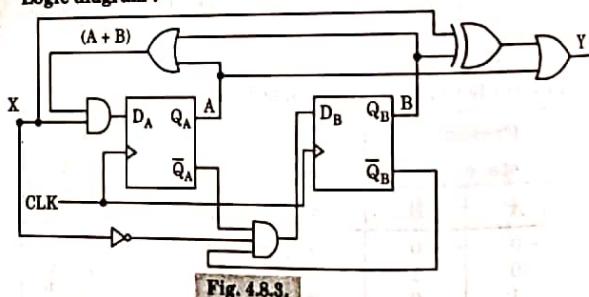


Fig. 4.8.3.

Que 4.9. Analyse the synchronous sequential circuit shown in Fig. 4.9.1 and draw the state diagram for it.

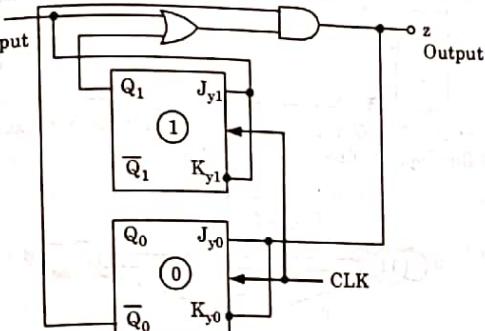


Fig. 4.9.1.

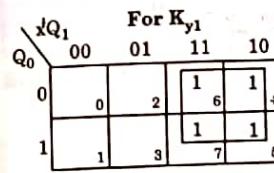
Answer

1. The boolean equations for the given circuit are :

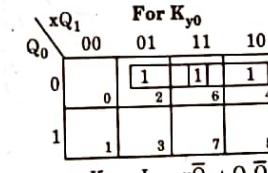
$$K_{y0} = J_{y0} = z = (x + Q_1)\bar{Q}_0 = x\bar{Q}_0 + Q_1\bar{Q}_0$$

$$K_{y1} = J_{y1} = x$$

2. The respective K-maps are :



$$K_{y1} = J_{y1} = x$$



$$K_{y0} = J_{y0} = x\bar{Q}_0 + Q_1\bar{Q}_0$$

Fig. 4.9.2.

3. State table :

Input	Present state		Next state		Flip-flop inputs				
	x	Q ₁	Q ₀	Q ₁	Q ₀	J _{y1}	K _{y1}	J _{y0}	K _{y0}
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	1
0	1	0	0	1	1	0	0	0	0
0	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1	0	0
1	0	1	1	1	1	1	1	1	1
1	1	1	0	0	1	1	1	0	0
1	1	1	1	0	1	1	0	1	0

Digital Logic Design

4-15 A (EC/CSIT-Sem-3)

4. Excitation table :

Q_n	Q_{n+1}	J	K
0	0	1	x
0	1	x	1
1	0	x	0
1	1	x	0

The entries of Q_1, Q_0 column are filled with the help of excitation table of JK flip-flop. The final state diagram is :

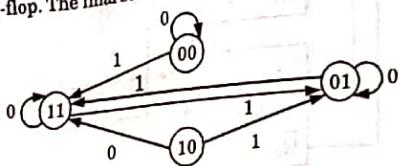


Fig. 4.9.3.

Que 4.10. What do you understand by state reduction ? Reduce the following state diagram.

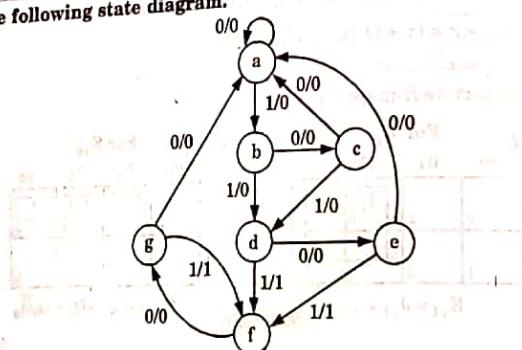


Fig. 4.10.1.

AKTU 2015-16, Marks 10

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.

4-16 A (EC/CSIT-Sem-3) Synchronous & Asynchronous Circuits

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

- 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	1
d	e	d	0	1
e	a	d	0	1

4. The state diagram of the reduced state table is shown in Fig. 4.10.2.

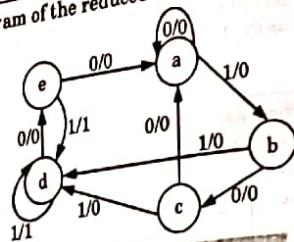


Fig. 4.10.2. State diagram.

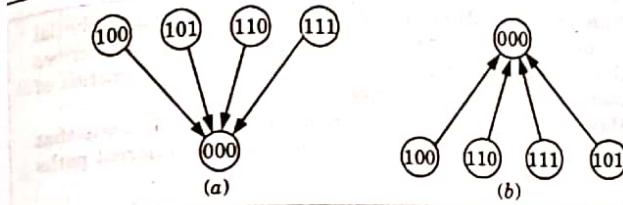
Que 4.11. Describe state assignment rules for assigning states for sequential circuits.

Answer

- 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.
- The following rules are used in state assignment.

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 assignments which can be grouped into logically adjacent cells in K-map.

Fig. 4.11.1. (a) State assignment rule 1 state diagram
(b) State assignment rule 2 state diagram.

- Que 4.12.** Describe the design procedure to design the clocked sequential logic circuit.

Answer

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

- Obtain the state table from the given circuit information such as a state diagram, a timing diagram or description.
- The number of states may be reduced by state reduction technique.
- Assign binary values to each state in the state table.
- Determine the number of flip-flops required and assign a letter symbol to each flip-flop.
- Choose the flip-flop type to be used according to the application.
- Derive the excitation table from the reduced state table.
- 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.

PART-2

*Analysis Procedure of Asynchronous Sequential Circuits,
Circuits with Latches, Design Procedure, Reduction of State and
Flow Table, Race Free State Assignment, Hazards.*

CONCEPT OUTLINE : PART-2

- Asynchronous sequential circuit :** It consists of a combinational circuit and delay elements connected to form feedback loops. There are n input variables, m output variables, and k internal states.

- Analysis procedure:** The analysis of asynchronous sequential circuits consists of obtaining a table or a diagram that describes the sequence of internal states and outputs as a function of changes in the input variables.
- Hazards:** Hazards are unwanted switching transients that may appear at the output of a circuit because different paths exhibit different propagation delays.
- Essential hazard:** This is caused by unequal delays along two or more paths that originate from the same input.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 4.13. Draw the block diagram of asynchronous sequential circuit and explain its working.

Answer

- 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 delay element in asynchronous sequential circuit.

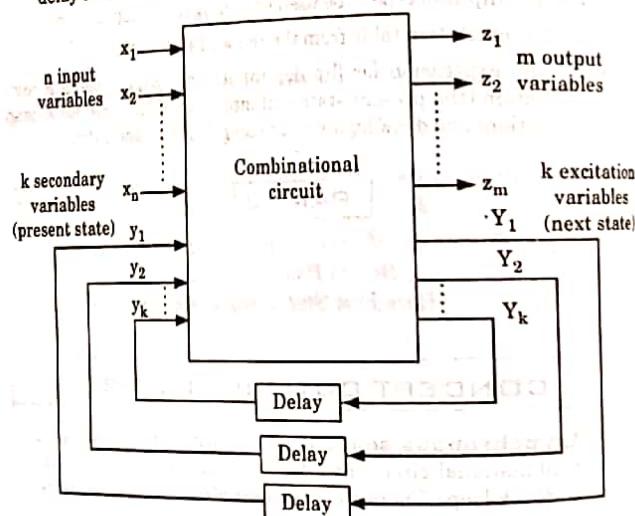


Fig. 4.13.1. Block diagram of an asynchronous sequential circuit.

- In asynchronous sequential logic circuit, operation is not synchronized because we are not using any synchronous clock.
- The block diagram of an asynchronous sequential circuit consists of a combinational circuit and delay elements connected to form feedback loops.
- There are n input variables, m output variables and k internal states. The delay element provides a short-term memory for the sequential circuit.
- The present state and next state variables are called as secondary variable and excitation variable respectively.
- 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.
- These values propagate through the delay element and become the new present state for the secondary variables.

Que 4.14. Differentiate between asynchronous and synchronous sequential circuit. Define fundamental mode of operation.

AKTU 2011-12, Marks 10

OR

State the differences between synchronous and asynchronous circuits with suitable examples.

AKTU 2013-14, Marks 10

Answer

S.No.	Synchronous sequential circuit	Asynchronous sequential circuit
1.	In synchronous circuit, memory elements are either unclocked 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.

- Fundamental mode asynchronous sequential circuit :**
1. In fundamental mode operation, the external inputs can change at any time and a transition from one state to another state occurs only when changes in input occurs.
 2. The input and output are represented by voltage levels rather than pulses.
 3. Delay lines are used as memory elements.

Que 4.15. Define fundamental mode operation for an asynchronous sequential circuit. What is the difference between an internal state and total state? Explain with example.

AKTU 2013-14, Marks 10

Answer

Fundamental mode operation : Refer Q. 4.14, Page 4-20A, Unit-4

- Difference between internal state and total state :
1. An internal state (or row state) is defined independent of input combinations by only the feedback loop state. The total state is defined by an internal state and the input combination.
 2. In an asynchronous circuit, the internal state can change immediately after a change in input. Because of this rapid change, it is sometimes convenient to combine the internal state with the input value together which is total state of the circuit.

$$\text{For example: } Y_1 = xy_1 + x'y_2$$

$$Y_2 = xy_1' + x'y_2$$

The next step is to plot the Y_1 and Y_2 functions in a map, as shown in Fig. 4.13.1(a) and (b).

x y_1y_2	0	1
00	0	0
01	1	0
11	1	1
10	0	1

(a) Map for
 $Y_1 = xy_1 + x'y_2$

x y_1y_2	0	1
00	0	1
01	1	2
11	1	3
10	0	2

(b) Map for
 $Y_2 = xy_1' + x'y_2$

x y_1y_2	0	1
00	0	0
01	1	1
11	1	0
10	0	0

(c) Transition table

Fig. 4.15.1.

3. The transition table shown in Fig. 4.15.1(c) is obtained from the map by combining the binary values in corresponding squares. The transition table shows the value of $Y = Y_1Y_2$ inside each square.
4. The circuit whose transition table is shown in Fig. 4.15.1(c) has four stable total state : $y_1y_2x = 000, 011, 110$, and 101 and four unstable total states : $001, 010, 111$, and 100 .

Que 4.16. Describe briefly the excitation table and output table of a fundamental mode asynchronous sequential circuit.

AKTU 2011-12, Marks 10

Answer

1. To describe excitation table and output table of fundamental mode asynchronous circuit we analyse the circuit.
2. Analysis of asynchronous sequential circuit comprises of obtaining a table or diagram that describes sequence of internal state and output as function of changes in the input variable.
3. Let us consider an asynchronous sequential circuit that have feedback path without employing flip-flop.

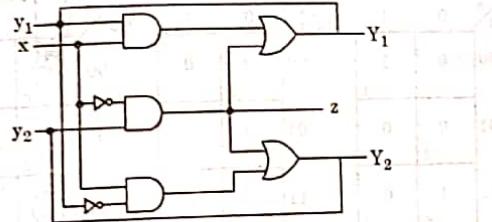


Fig. 4.16.1.

4. Fig. 4.16.1 shows sequential circuit without using any unclocked flip-flop. There are two feedback paths in the circuit.
5. The internal state has two excitation variables Y_1 and Y_2 and two secondary variables y_1 and y_2 . x is input variable to circuit.
6. The analysis of circuit starts by writing the boolean expression for excitation variables as a function of input and secondary variable.
7. These equations are obtained from logic diagram shown in Fig. 4.16.1.

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

$$Y_2 = x\bar{y}_1 + \bar{x}y_2$$

$$z = \bar{x}y_2$$

8. For the fundamental mode, only one input variable should change its values. We use Gray code format for this purpose.

Digital Logic Design

4-23 A (EC/CS/IT-Sem-3)

9. Table 4.16.1 shows present total state, next total state and the output variable z . It helps in understanding whether the circuit is stable or unstable.
10. When input change occurs, present state = next state \Rightarrow stable state.

Table 4.16.1.

Present state			Next state			Stable state	Output
y_2	y_1	x	y_2	y_1	x	Yes/No	z
0	0	0	0	0	0	Yes	0
0	0	1	1	0	1	No	0
0	1	0	0	0	1	No	0
0	1	1	0	1	0	Yes	0
1	0	0	1	1	0	No	1
1	0	1	1	0	1	Yes	0
1	1	0	1	1	0	Yes	1
1	1	1	0	1	1	No	0

Now, K-map for Y_1 , Y_2 in terms of y_1 , y_2 and x .

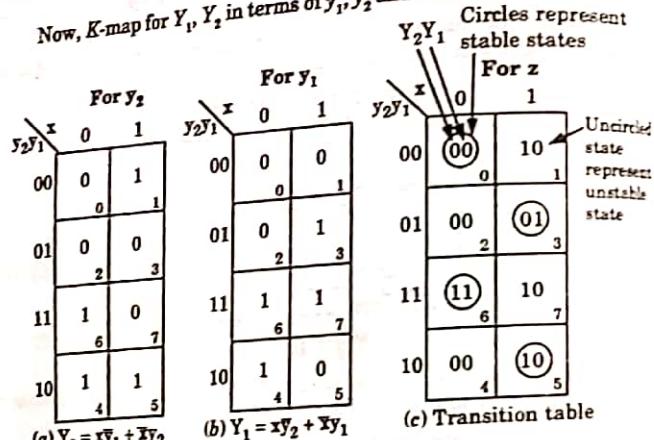


Fig. 4.16.2.

Que 4.17. Discuss the concept of transition table, flow table and primitive flow table to design fundamental mode asynchronous sequential circuit.

AKTU 2012-13, Marks 10

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.

4-24 A (EC/CS/IT-Sem-3)

Synchronous & Asynchronous Circuits

2. Like state table, transition table also represents relationship between input, output and flip-flop states. The Table 4.17.1 shows the transition table.

Table. 4.17.1.

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	01	00	0	0
1	0	10	01	1	0
1	1	00	10	1	0

4. Here, AB are the state variables. 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. 4.17.1.

3. The table shown in Fig. 4.17.1(a) is called primitive flow table because it has only one state in each row and the table shown in Fig. 4.17.1(b) shows a flow table with more than one stable state in same row.

Digital Logic Design

4-25 A (EC/CS/IT-Sem-3)

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 .

$x_1 x_2$	00	01	11	10
y	0 0	0 1	0 3	0 2
0	0 0	0 1	0 3	0 2
1	0 4	0 5	0 7	0 6

Transition table $Y = \bar{x}_1 x_2' + x_1 y$ Map for output $z = x_1 x_2'$

Fig. 4.17.2.

7. Now the logic diagram is as follows:

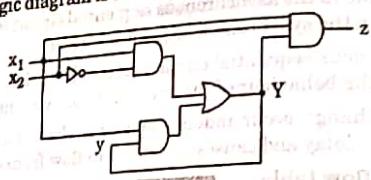


Fig. 4.17.3.

Que 4.18. An asynchronous sequential circuit has two internal states and one output. The excitation functions and output function of the circuit are as follows :

$$Y_1 = \bar{x}_1 x_2 + x_2 y_1$$

$$Y_2 = x_1 y_2 + x_2$$

and output function

$$Z = x_1 + y_2$$

i. Draw the logic diagram of the circuit.

ii. Obtain the transition table and output map.

AKTU 2012-13, Marks 10

Answer

i. Logic diagrams : $Y_1 = \bar{x}_1 x_2 + x_2 y_1$, $Y_2 = x_1 y_2 + x_2$ and output $Z = x_1 + y_2$

4-26 A (EC/CS/IT-Sem-3)

Synchronous & Asynchronous Circuits

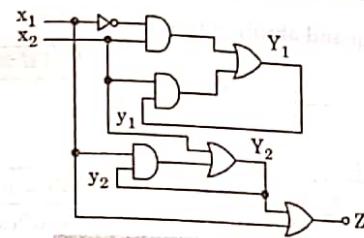


Fig. 4.18.1. Logic diagram.

ii. Transition table and output map :

		For Y_1			
$y_1 y_2$	$x_1 x_2$	00	01	11	10
00	00	0 0	1 1	0 3	0 2
01	01	0 4	1 5	0 7	0 6
11	11	0 12	1 13	1 15	0 14
10	10	0 6	1 9	1 11	0 10

$$Y_1 = \bar{x}_1 x_2 + x_2 y_1$$

$$Y_2 = x_1 y_2 + x_2$$

		For Z			
$y_1 y_2$	$x_1 x_2$	00	01	11	10
00	00	0 0	0 1	1 3	1 2
01	01	1 4	1 5	1 7	1 6
11	11	1 12	1 13	1 15	1 14
10	10	0 8	0 9	1 11	1 10

$$Z = x_1 + y_2$$

		Transition table			
$y_1 y_2$	$x_1 x_2$	00	01	11	10
00	00	(00, 0)	11, 0	01, 1	(00, 1)
01	01	00, 1	11, 1	(01, 1)	(01, 1)
11	00	00, 1	(11, 1)	(11, 1)	01, 1
10	00	00, 0	11, 0	11, 1	00, 1

Fig. 4.18.2.

Que 4.19. An asynchronous sequential circuit described by the following excitation and output functions.

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

where, X_1 and X_2 = input variables

$$Y = \text{excitation function}$$

$$Z = \text{output function}$$

i. Draw the logic diagram of the circuit.

ii. Derive transition table.

Digital Logic Design

iii. Output map and obtain a flow table.

4-27 A (EC/CS/IT-Sem-3)

AKTU 2014-15, Marks 06

Answer

State table :

Present state		Next state			Stable state	Output Z
X_1	X_2	Y	X_1'	X_2'	Y'	Yes/No
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

i. Logic diagram :

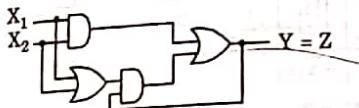


Fig. 4.19.1.

ii. Transition table :

X_1X_2		00	01	11	10	Unstable state
Y		0	0	1	0	Stable state
X_1	X_2	0	0	1	0	
1	0	0	1	1	1	

Fig. 4.19.2.

iii. Output map :

X_1X_2		00	01	11	10
Y		0	0	-	0
X_1	X_2	0	0	1	1
1	0	-	1	1	1

Fig. 4.19.3.

4-28 A (EC/CS/IT-Sem-3)

Synchronous & Asynchronous Circuits

Flow table : Let $S_0 = 0$ and $S_1 = 1$

x_1x_2		00	01	11	10
y	S_0	$S_0/0$	$S_0/0$	$S_1/1$	$S_0/0$
	S_1	$S_0/0$	$S_1/1$	$S_1/1$	$S_1/1$

Fig. 4.19.4.

Que 4.20. Define critical race and non-critical race. Discuss the concept of non-critical race with examples.

AKTU 2011-12, Marks 10

What are critical race and non-critical race ? How can they be avoided ?

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.
2. When unequal delays are encountered, a race condition may cause the state variables to change in an unpredictable manner.

x_1x_2		00	01	11	10
y	0	(0) 0	(0) 1	(0) 3	1 2
	1	0 4	0 5	(1) 7	(1) 6

Transition table $Y = x_1x_2' + x_1'y$

x_1x_2		00	01	11	10
y	0	0 0	0 1	0 3	0 2
	1	0 4	0 5	1 7	0 6

Map for output $z = x_1x_2y$

Fig. 4.20.1.

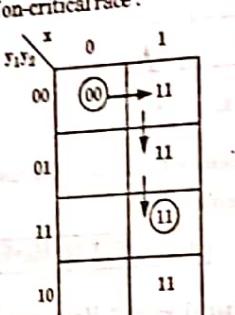
3. For example, if state variable changes from 00 to 11, the difference in delays may cause the first variable to change faster than the second, therefore the state variable change in sequence from 00 to 10 and then to 11. If second variable changes faster than the first then state variable changes from 00 to 01 and then to 11.

Critical race and non-critical race :

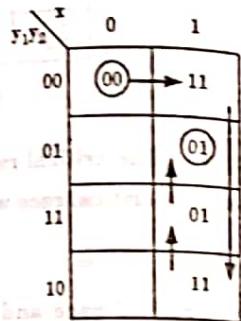
1. If the final stable state that the circuit reaches does not depend on the order in which the state variable changes, the race is called a non-critical race.

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 :



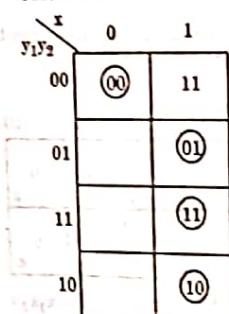
- (a) Possible transitions
 $00 \rightarrow 11$
 $00 \rightarrow 01 \rightarrow 11$
 $00 \rightarrow 10 \rightarrow 11$



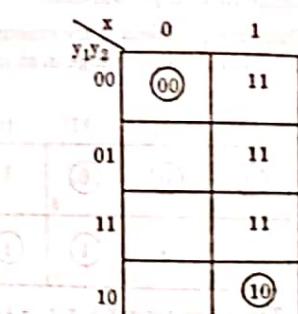
- (b) Possible transitions
 $00 \rightarrow 11 \rightarrow 01$
 $00 \rightarrow 01$
 $00 \rightarrow 10 \rightarrow 11 \rightarrow 01$

Fig. 4.20.2.

Critical race :



- (a) Possible transitions
 $00 \rightarrow 11$
 $00 \rightarrow 01$
 $00 \rightarrow 10$



- (b) Possible transitions
 $00 \rightarrow 11$
 $00 \rightarrow 01 \rightarrow 11$
 $00 \rightarrow 10 \rightarrow 10$

Fig. 4.20.3.

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.

Ques 4.21. Differentiate between :

- i. Stable state and unstable state.
- ii. Critical race and non-critical race.

AKTU 2012-13, Marks 10

Answer

S.No.	Stable state	Unstable state
1.	For a given input, the system is stable if the circuit reaches a steady state condition with $y_i = Y_i$.	For a given input, the unstable system gives the random values without reaching to the steady state.
2.	A circuit system is called a stable system if for a given input signal, there must be an output.	In unstable system, the output can not be predicted at any instant of time.
3.	A system is stable, if the input is bounded then the output should be bounded.	A system is unstable if the output is not limited or unbounded for the given input.

- ii. Critical race and non-critical race : Refer Q. 4.20, Page 4-28A, Unit-4.

Ques 4.22. Design a SR latch using NOR gates as well as NAND gates.

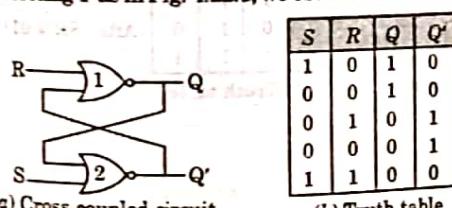
Answer

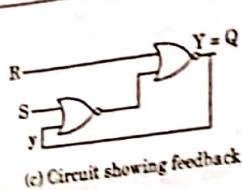
SR latch :

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

$$Y = [(S + y)' + R]' = (S + y)R' = SR' + R'y$$

2. Plotting Y as in Fig. 4.22.1, we obtain the transition table for the circuit.





(c) Circuit showing feedback

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

(d) Transition table

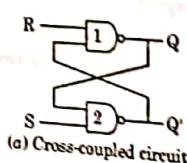
$$Y = SR' + R'y$$

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

Fig. 4.22.1. SR latch with NOR gates.

3. 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 $SR = 11$, 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.
4. 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 $SR = 0$,
5. 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.
6. The analysis of the SR latch with NAND gates is carried out in Fig. 4.22.2. 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 $SR' = 0$. The excitation function for the circuit is Fig. 4.22.2 is

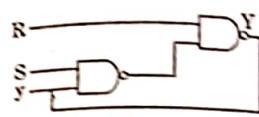
$$Y[S(Ry)]' = S' + Ry$$



(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

(b) Truth table



(c) Circuit showing feedback

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

(d) Transition table

Fig. 4.22.2. 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.

Ques 4.23. Derive a latch circuit from a transition table.

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. 4.23.1(d). 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. 4.23.1(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.

	x_1x_2	00	01	11	10
y	0	0 ₀	0 ₁	0 ₃	1 ₂
	1	0 ₄	0 ₅	1 ₇	1 ₆

(a) Transition table

$$y = x_1x_2' + x_1y$$

	x_1x_2	00	01	11	10
y	0	0 ₀	0 ₁	0 ₃	1 ₂
	1	0 ₄	0 ₅	1 ₇	1 ₆

(b) Excitation table

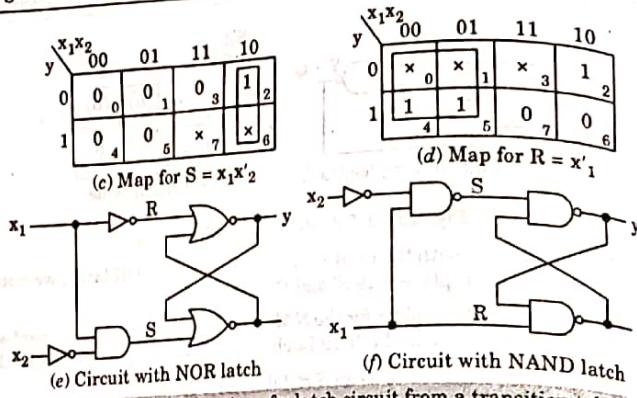


Fig. 4.23.1. Derivation of a latch circuit from a transition table.

Que 4.24. Write short notes on :

- Fundamental mode asynchronous sequential circuit.
- Pulse mode asynchronous sequential circuit.

Answer

- Fundamental mode asynchronous sequential circuit :** Refer Q. 4.14, Page 4-20A, Unit-4.
- Pulse mode asynchronous sequential circuit :**
 - We have seen that pulse mode asynchronous sequential circuit rely on input pulse rather than levels.
 - In this mode pulses may arrive at any time :
 - No two pulses should arrive at input simultaneously.
 - Width of the pulse must be long enough for the circuit to respond to the input.
 - Pulse width must not be so long that it is still present even after the new state is reached and cause a faulty change of state.
 - Flip-flops are commonly used as memory element. Memory element transitions are initiated only by input pulses.
 - Input variables are used only in the uncomplemented or the complemented forms.

Que 4.25. Explain the design procedure for asynchronous sequential circuit.**Answer**

The steps involved in designing an asynchronous sequential circuit are:

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

Design example : By using the boolean expression for S and R inputs, we can draw the logic diagram as shown in Fig. 4.25.1.

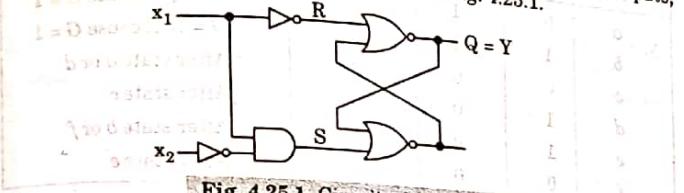


Fig. 4.25.1. Circuit with NOR latch.

Design procedure for implementing a circuit with SR latch :

- Given a transition table that specifies the excitation function $Y = Y_i Y_{i+1} \dots Y_K$, derive a pair of maps for S_i and R_i for each $i = 1, 2, \dots, K$. This is done by choosing the conditions specified in the latch excitation table.
- Derive the simplified boolean function for each S_i and R_i . Care must be taken not to make S_i and R_i equal to 1 in the same minterm square.
- Draw the logic diagram using k latches together with the gates required to generate the S and R boolean function. For NOR latches, use the S and R boolean obtained in step 2. For NAND latches, use the complemented values of those obtained in step 2.

Que 4.26. Discuss the concept of primitive flow table to design fundamental mode asynchronous sequential circuit.**Answer**

- A primitive flow table is a flow table with only one stable total state in each row. Remember that a total state consists of the internal state combined with the input.
- The derivation of the primitive flow table can be facilitated if we first form a table with all possible total states in the system. This is shown in Table 4.26.1 for the gated latch.
- Each row in the table specifies a total state, which consists of a letter designation for the internal state and a possible input combination for D and G .
- The output Q is also shown for each total state. From the design specifications, we know that $Q = 0$ if $DG = 01$ and $Q = 1$ if $DG = 11$, because D must be equal to Q when $G = 1$.

5. We assign these conditions to states a and b . When G goes to 0, the output depends on the last value of D . Thus, if the transition of DG is from 01 to 00 to 10, then Q must remain 0 because D is 0 at the time of the transition from 1 to 0 in G .

Table 4.26.1 Gated latch total states

State	Inputs		Output Q	Comments
	D	G		
a	0	1	0	$D = Q$ because $G = 1$
b	1	1	1	$D = Q$ because $G = 1$
c	0	0	0	After state a or d
d	1	0	0	After state c
e	1	0	1	After state b or f
f	0	0	1	After state e

6. If the transition of DG is from 11 to 10 to 00, then Q must remain 1. This information results in six different total states, as shown in the Table 4.26.1. Note that simultaneous transitions of two input variables, such as from 01 to 10 or from 11 to 00, are not allowed in fundamental mode operation.
7. The primitive flow table for the gated latch is shown in Fig. 4.26.1. It has one row for each state and one column for each input combination.

States	Inputs DG			
	00	01	11	10
a	$c, -$	$(a), 0$	$b, -$	$-,-$
b	$-,-$	$a, -$	$(b), 1$	$e, 1$
c	$(c), 0$	$a, -$	$-,-$	$d, -$
d	$c, -$	$-,-$	$b, -$	$(d), 0$
e	$f, -$	$-,-$	$b, -$	$(e) -$
f	$(f), 1$	$a, 1$	$-,-$	$e, -$

Fig. 4.26.1. Primitive flow table.

8. The output is marked with a dash to indicate a don't care condition. All outputs associated with unstable states are marked with a dash to indicate don't care conditions.

Que 4.27. What do you mean by merging? Explain the reduction of primitive flow table.

Answer

1. The primitive flow table has only one stable state in each row. The table can be reduced to a smaller number of rows if two or more stable

- states are placed in the same row. The grouping of stable states from separate rows into one common row is called merging.
2. Merging a number of stable states in the same row means that the binary state variable ultimately assigned to the merged row will not change when the input variable changes.
3. Two or more rows in the primitive flow table can be merged into one row if there are non-conflicting states and outputs in each of the columns. Whenever one state symbol and don't-care entries are encountered in the same column, the state is listed in the merged row.
4. Moreover, if the state is circled in one of the rows, it is also circled in the merged row. The output value is included with each stable state in the merged row. Because the merged states have the same output, the state cannot be distinguished on the basis of the output.

States	DG			
	00	01	11	10
a	$c, -$	$(a), 0$	$b, -$	$-,-$
c	$(c), 0$	$a, -$	$-,-$	$d, -$
d	$c, -$	$-,-$	$b, -$	$(d), 0$

(a) States that are candidates for merging

States	DG			
	00	01	11	10
a, c, d	$(c), 0$	$(a), 0$	$b, -$	$(d), 0$
c, e, f	$(f), 1$	$a, -$	$(b), 1$	$(e), 1$

(b) Reduced table (two alternatives)

Fig. 4.27.1.

The primitive flow table is separated into two parts of three rows each, as shown in Fig. 4.27.1(a).

5. Each part shows three stable states that can be merged because there are no conflicting entries in each of the four columns.
6. The first column shows state c in all the rows and 0 or a dash for the output. Since a dash represents a don't-care condition, it can be associated with any state or output. The two dashes in the first column can be taken to be 0 output to make all three rows identical to a stable state c with a 0 output.
7. The second column shows that the dashes can be assigned to correspond to a stable state a with a 0 shows that the dashes can be assigned to correspond to a stable state a with a 0 output.
8. Note that if a state is circled in one of the rows, it is also circled in the merged row.

9. Similarly, the third column can be merged into an unstable state b with a don't-care output, and the fourth column can be merged into stable state d and a 0 output.
10. Thus, the three rows a, c , and d can be merged into one row with three stable states and one unstable state, as shown in the first row of Fig. 4.27.1.(b).

Que 4.28. Describe the state reduction technique and reduction of flow table. OR How to obtain a reduced state table from implication table.

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 4.28.1. State table to demonstrate equivalent states

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

Table 4.28.2. 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 (\times) in the corresponding square, whereas their equivalence is recorded with a check mark (\checkmark).
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.

a	d, e \checkmark					
c	\times	\times				
d	\times	\times	\times			
e	\times	\times	\times	\times		
f	$c, d \times$ a, b		\times	\times	\times	
g	\times	\times	\times	$d, e \checkmark$	$d, e \checkmark$	\times
	a	b	c	d	e	f

Fig. 4.28.1. Implication 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.

4-39 A (EC/CS/IT-Sem-3)

Digital Logic Design

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 4.28.3.

Table 4.28.3 : Reduced state table

Present 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

Que 4.29. Convert the flow table of Fig. 4.29.1. into a transition table by assigning the following binary values to the states : $a = 00$, $b = 11$ and $c = 01$.

- i. Assign values to the extra fourth state to avoid critical races.
- ii. Assign outputs to the don't care states to avoid momentary false outputs.

	00	01	11	10
a	(a), 0	b, -	c, -	(a), 1
b	a, -	(b), 0	(b), 0	c, -
c	a, -	b, -	(c), 1	(c), 0

Fig. 4.29.1.

4-40 A (EC/CS/IT-Sem-3)

Synchronous & Asynchronous Circuits

Answer

- i. A flow table with three states requires an assignment of 2 variables. we have : $a \rightarrow b, a \rightarrow c, b \rightarrow a, b \rightarrow c, c \rightarrow a$ and $c \rightarrow b$

a	00
b	11
c	01

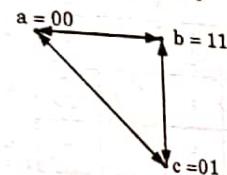


Fig. 4.29.2. Transition diagram.

2. This assignment will cause a critical race during the transition from a to b (two changes in the binary state) and also from b to a .
3. A race free assignment can be obtained by adding an extra row to the original flow table.

	00	01	11	10
a	(a)	d	c	(a)
b	d	(b)	(b)	c
c	a	b	(c)	(c)
d	a	b	-	-

	00	01	11	10
a	(a)	d	c	(a)
b	d	(b)	(b)	c
c	a	b	(c)	(c)
d	a	b	-	-

Flow table

Fig. 4.29.3.

4. The added row d is assigned the binary value (10), which is adjacent to both a and b .
5. The transition from a to b must go through d thus avoiding critical race.
6. The two squares with dashes in row d represent unspecified state (don't care).

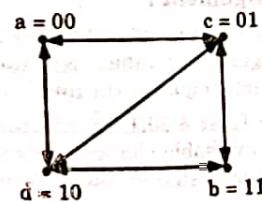


Fig. 4.29.4. Transition diagram.

Digital Logic Design

- 4-41 A (EC/CS/IT-Sem-3)
2. The new flow table is converted to a transition table to complete the design process.

		x_1x_2	00	01	11	10
		a = 00	(00)	10	01	(00)
x_1x_2	a = 00	b = 11	10	(11)	(11)	01
		c = 01	00	11	(01)	(01)
d = 10		00	11	-	-	-

Flow table Transition table

Fig. 4.29.5.

Que 4.30. What is the significance of state assignment? List the different techniques used for state assignment.

Answer

State assignment :

- 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.
- The following methods can be used to avoid critical race condition:

i. Shared row state assignment :

- 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.

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

- The state A is assigned binary value 00 and state C is assigned binary value 11.

- The transition from A to C will go through D, which satisfies the condition that only one binary variable changes during each state transition avoiding the critical race.

ii. One hot state assignment :

- 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.

- Let us consider the Table 4.30.1. 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.

4-42 A (EC/CS/IT-Sem-3)

Synchronous & Asynchronous Circuits

Table 4.30.1.

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	(B)	C	D	
0	1	0	0	C	(C)	B	(C)	(C)
1	0	0	0	D	(D)	B	C	(D)

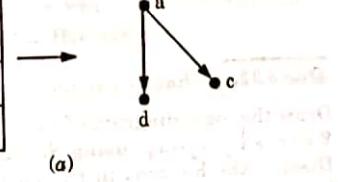
Que 4.31. Consider the reduced flow table shown in Fig. 4.31.1. Obtain the transition diagram for a race-free binary state assignment.

00 01 11 10				
a	(a)	c	(a)	d
b	a	(b)	c	(b)
c	(c)	(c)	(c)	d
d	(d)	b	a	(d)

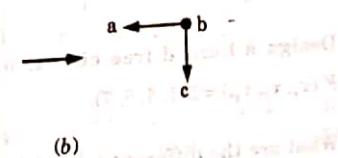
Fig. 4.31.1.

Answer

Flow table				
00		01	11	10
a	(a)	c	(a)	d
b	a	(b)	c	(b)
c	(c)	(c)	(c)	d
d	(d)	b	a	(d)



Flow table				
00		01	11	10
a	(a)	c	(a)	d
b	a	(b)	c	(b)
c	(c)	(c)	(c)	d
d	(d)	b	a	(d)



Digital Logic Design

Flow table				
	00	01	11	
s	a	c	a	d
b	a	b	c	b
c	c	c	c	d
d	d	b	a	d

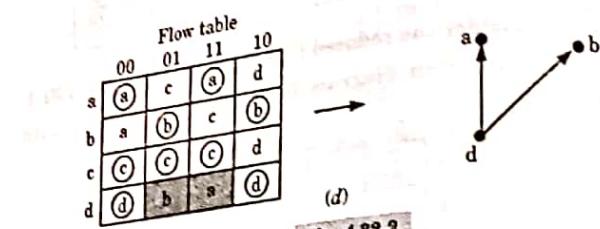


Fig. 4.31.2.

The final transition diagram can be obtained by joining all these four transition diagrams.

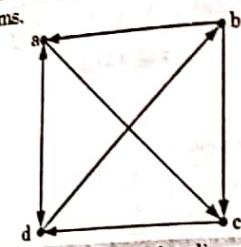


Fig. 4.31.3. Transition diagram.

Que 4.32. What is various types of hazards in digital circuit?

Draw the logic diagram of the product of sum expression:

$$Y = (x_1 + x_2')(x_2 + x_3)$$
, using NAND gates.

Discuss the hazards in the circuit and find a way to remove the hazard. Draw the hazard free realization of the circuit.

AKTU 2013-14, Marks 10

OR

Design a hazard free circuit of the following boolean function

$$F(x_1, x_2, x_3) = \Sigma(1, 5, 6, 7)$$
.

AKTU 2016-17, Marks 10

OR

What are the different types of hazards in asynchronous circuits?

4-43 A (EC/CS/IT-Sem-3)

4-44 A (EC/CS/IT-Sem-3) Synchronous & Asynchronous Circuits

Answer

- i. 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.

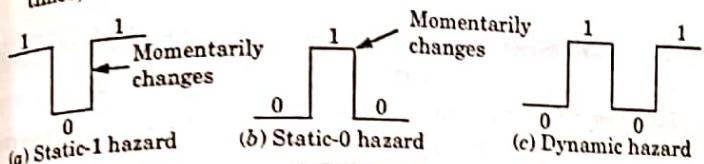


Fig. 4.32.1.

- 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. 4.32.2.
- 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.

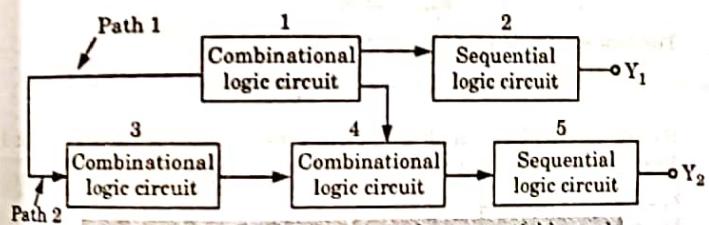


Fig. 4.32.2. Schematic circuit to show essential hazard.

Elimination of hazards :

1. Static and dynamic hazard can be prevented by adding extra gates in the circuit as the redundant term. This is done by grouping the two adjacent 1's or 0's which are responsible for hazard.
2. Fig. 4.32.3 shows the logic circuit with hazard problem.

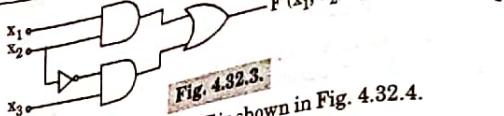
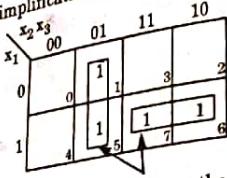


Fig. 4.32.3.

3. The K-map simplification for F is shown in Fig. 4.32.4.

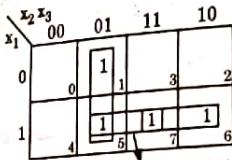


Two adjacent 1's responsible for the static-1 hazard.
Fig. 4.32.4.

4. Expression for output F is given by

$$F = \bar{x}_2x_3 + x_1x_2$$

5. Then we group the two 1's to eliminate static-1 hazard. The modified K-map is shown in Fig. 4.32.5.



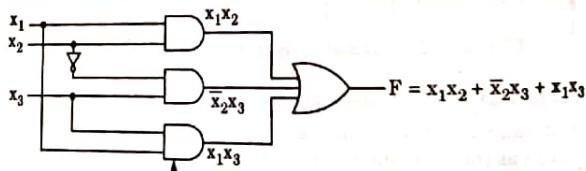
new pair of 1's (x_1x_3) formed
to eliminate static-1 hazard.

Fig. 4.32.5.

6. The new expression for output will be

$$F = \bar{x}_2x_3 + x_1x_2 + x_1x_3$$

7. So, the hazard free circuit is shown in Fig. 4.32.6. Note that static-1 hazard has been eliminated due to the additional AND gate.



Additional gate used to make the circuit hazard free.

Fig. 4.32.6.

$$\begin{aligned} Y &= (x_1 + \bar{x}_2)(x_2 + x_3) \\ Y &= \overline{\overline{Y}} = \overline{(x_1 + \bar{x}_2)(x_2 + x_3)} \\ &= \overline{(x_1 + \bar{x}_2)} + \overline{(x_2 + x_3)} \\ &= \bar{x}_1x_2 + \bar{x}_2\bar{x}_3 = (\bar{x}_1x_2) \times (\bar{x}_2\bar{x}_3) \end{aligned}$$

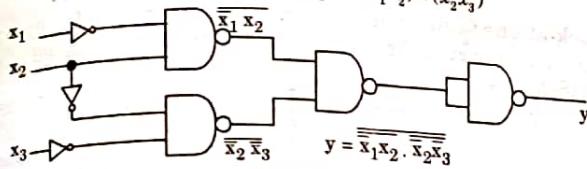


Fig. 4.32.7.

Que 4.33. Explain hazard and its types. Define critical race and non-critical race. Also explain the elimination of hazards in asynchronous circuits.

AKTU 2014-15, Marks 06

OR

Describe the hazards in digital circuits. How are these removed?

AKTU 2015-16, Marks 10

Answer

Hazards :

- Hazards are unwanted switching transient that may appear at the output of a circuit because different paths exhibit different propagation delays.
- Hazards occur in combinational circuits, where they may cause a temporary false-output value. When this condition occurs in asynchronous sequential circuits, it may result in a transition to a wrong stable state.

Types of hazards : Refer Q. 4.32, Page 4-43A, Unit-4.

Critical race and non-critical race : Refer Q. 4.20, Page 4-28A, Unit-4.

Elimination of hazards : Refer Q. 4.32, Page 4-43A, Unit-4.

Que 4.34. Explain the term :

i. Stuck-at-0 fault.

ii. Stuck-at-1 fault.

Answer

- Stuck-at-zero fault :** When an input line is grounded by a faulty connection so that the line is permanently at logic 0 is called stuck-at-zero (S-A-0) fault.

4-47 A (EC/CS/IT-Sem-3)

Digital Logic Design

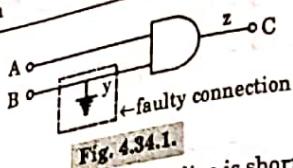


Fig. 4.34.1.

- ii. **Stuck-at-one fault:** When an input line is shorted to V_{CC} by a faulty connection so that the line is permanently at logic 1, is called stuck-at-one (S-A-1) fault.

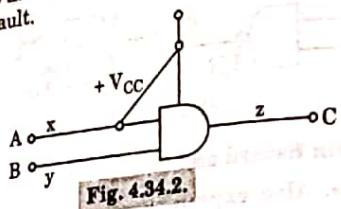


Fig. 4.34.2.

Que 4.35. What do you mean by spikes? Why do they occur? Explain static-0, static-1 and dynamic hazards.

Answer

- When an input to the combinational logic changes, unwanted switching transients may appear at the network output, these unwanted transients are called spikes.
- The transient occurs in a network because of different path delays from input of the network to the output of the network.
- Consider a network.

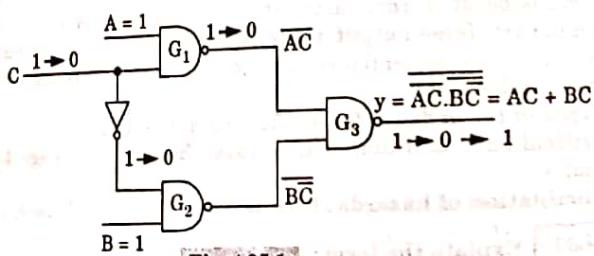


Fig. 4.35.1.

- Consider a situation where, $A = B = 1$ and C changes from 1 to 0. For this new input condition the network output should remain constant to logic-1.
- However, because of different propagation delays of different gates, the network output Y may momentarily go to logic 0. This would happen if the gate outputs switches in following sequence :

4-48 A (EC/CS/IT-Sem-3)

Synchronous & Asynchronous Circuits

i. Gate-1 output goes to 1

ii. Gate-3 output goes to 0, i.e., $y = 0$

iii. Inverter output goes to 1

iv. Gate-2 output goes to '0'

v. Gate-3 output (i.e., y) goes to '1'

Static-0, static-1 and dynamic hazard: Refer Q. 4.32, Page 4-43A, Unit-4.

Que 4.36. Find a circuit that has no static hazards and implement the boolean function : $F(A, B, C, D) = \Sigma m(0, 2, 6, 7, 8, 10, 12)$

Answer

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

1. K-map :

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

$$F(A, B, C, D) = B'D' + A'BC + AC'D'$$

Fig. 4.36.1.

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

$$F(A, B, C, D) = B'D' + A'BC + AC'D' + A'CD'$$

Fig. 4.36.2.

2. Circuit:

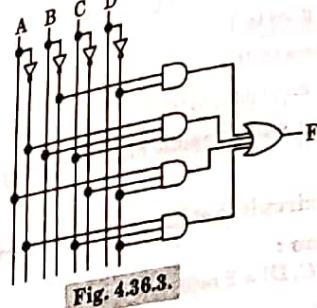


Fig. 4.36.3.

Que 4.37. Distinguish between synchronous and asynchronous digital sequential circuit. Design modulo-5 counter.

AKTU 2014-15, Marks 06

Answer

Difference between synchronous and asynchronous digital sequential circuit: Refer Q. 4.14, Page 4-20A, Unit-4.

Modulo - 5 counter :

Step 1 : Here 5 indicates total number of states, the count sequences are from 0 to 4.

Step 2 : The required number of flip-flops are

$$2^n \geq N$$

$$2^3 \geq 5$$

$$n = 3$$

Step 3 : Desired sequence and state diagram :

Desired sequence

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

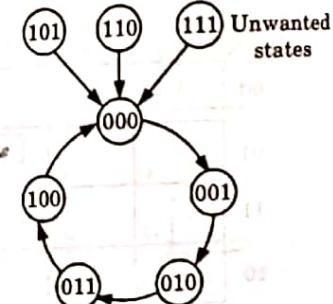


Fig. 4.37.1. State diagram.

Step 4 : Excitation table for JK flip-flop :

Present states Q_n	Next state Q_{n+1}	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

Step 5 : State table :

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

Step 6 : K-map simplification :

		For J_C			
Q_C	$Q_B Q_A$	00	01	11	10
0	00	0 ₀	0 ₁	1 ₃	0 ₂
1	x4	x5	x7	x6	

$$J_C = Q_B Q_A$$

		For K_C			
Q_C	$Q_B Q_A$	00	01	11	10
0	x0	x1	x3	x2	
1	14	15	17	16	

$$K_C = 1$$

		For J_B			
Q_C	$Q_B Q_A$	00	01	11	10
0	00	1 ₁	x ₃	x ₂	
1	04	0 ₅	x ₇	x ₆	

$$J_B = \overline{Q}_B Q_A$$

		For K_B			
Q_C	$Q_B Q_A$	00	01	11	10
0	x0	x1	x3	x2	
1	x4	x5	x7	x6	

$$K_B = Q_C + Q_A$$

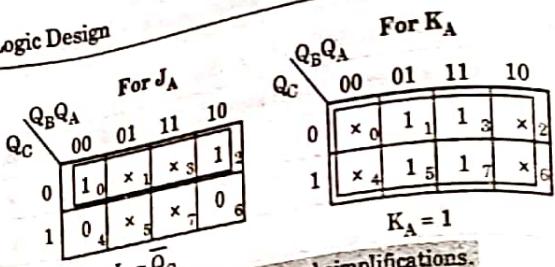


Fig. 4.37.2. K-maps and simplifications.

Step 7: The logic diagram of mod-5 counter is shown in Fig. 4.37.3.

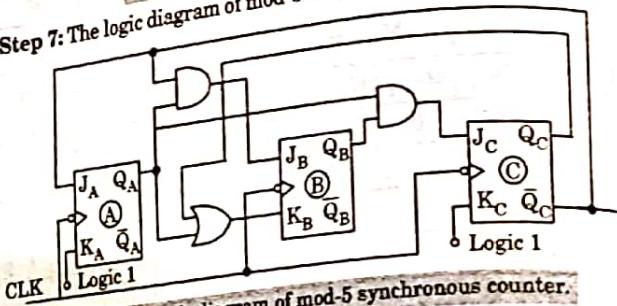


Fig. 4.37.3. Logic diagram of mod-5 synchronous counter.

Memory and Programmable Logic Devices

Part-1

- Digital Logic Families : DTL, DCTL, TTL, ECL and CMOS etc.
- Fan-out, Fan-in
- Noise Margin

A Concept Outline : Part-1

B Long and Medium Answer Type Questions 5-2A

(5-2A to 5-19A)

Part-2

- RAM
- ROM
- PLA
- PAL

A Concept Outline : Part-2 5-20A

B Long and Medium Answer Type Questions 5-20A

(5-20A to 5-38A)

Part-3

- Circuits of Logic Families
- Interfacing of Digital Logic Families
- Circuit Implementation Using ROM, PLA, PAL
- CPLD
- FPGA

A Concept Outline : Part-1 5-38A

B Long and Medium Answer Type Questions 5-38A

PART-1
Digital Logic Families : DTL, DCTL, TTL, ECL and CMOS etc.,
Fan-out, Fan-in, Noise Margin

CONCEPT OUTLINE : PART-1

- Digital circuits are constructed on a single chip, known as ICs. The set of compatible ICs with the same logic levels and same supply voltage have been fabricated to perform the various logic functions known as logic family.
- Diode-transistor logic (DTL) and direct-coupled transistor logic (DCTL) belong to saturated bipolar logic families. The speed of saturated bipolar logic family is low.
- Transistor-transistor logic (TTL) and CMOS are suitable forSSI and MSI.
- Emitter-coupled logic (ECL) is mainly used in superfast computers.
- Fan-out :** The maximum number of standard loads that the output of the gate can drive without impairing its normal operation. It is also called loading factor.
- Fan-in :** The number of inputs that the gate is designed to handle.
- Noise immunity :** 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.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 5.1. Discuss in brief about digital logic family.

Answer

- 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.
- Based on the fabrication technology, logic families are classified into two types :
 - Bipolar logic family.
 - Unipolar logic family.

i. 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 :

a. 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 :

- Resistor-transistor logic (RTL)
- Direct-coupled transistor logic (DCTL)
- Diode-transistor logic (DTL)
- High-threshold logic (HTL)
- Transistor-transistor logic (TTL)
- Integrated injection logic (IIL or I²L)

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 :

- Schottky transistor-transistor logic
- Emitter-coupled logic (ECL)
- ii. Unipolar logic family :

- The unipolar families include p-channel metal-oxide semiconductor field-effect transistor (PMOS), n-channel metal-oxide semiconductor field-effect transistor NMOS and CMOS.
- 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 :

- High fan-out :** High fan-out is advantageous as it reduces the need of additional drivers to drive more gates.
- 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.
- 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.

Memory & Programmable Logic Devices

- 5-4 A (EC/CS/IT-Sem-3)**
- Small size : The size of IC should be small as much as possible.
 - 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

- It is a technology for designing and fabricating digital circuits where logic gates employ both diodes and transistors.
- The three-input DTL NAND gate is shown in Fig. 5.3.1.
- The purpose of the input resistors in Fig. 5.3.1(a) is to isolate the inputs from one another.

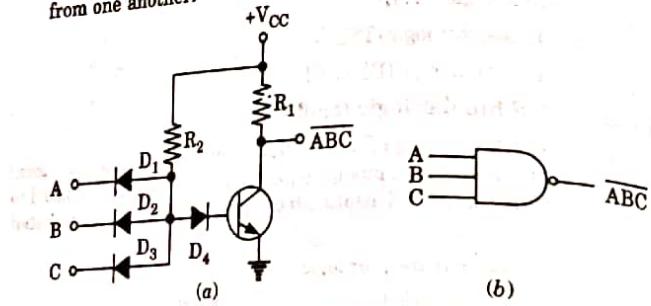


Fig. 5.3.1. DTL NAND gate: (a) Schematic, (b) Symbol.

- 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.
- 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.
- A logic family designed to get around this problem is TTL (transistor-transistor logic) family.
- 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

- 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.

Memory & Programmable Logic Devices

- 5-4 A (EC/CS/IT-Sem-3)**
- Small size : The size of IC should be small as much as possible.
 - 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

- It is a technology for designing and fabricating digital circuits where logic gates employ both diodes and transistors.
- The three-input DTL NAND gate is shown in Fig. 5.3.1.
- The purpose of the input resistors in Fig. 5.3.1(a) is to isolate the inputs from one another.

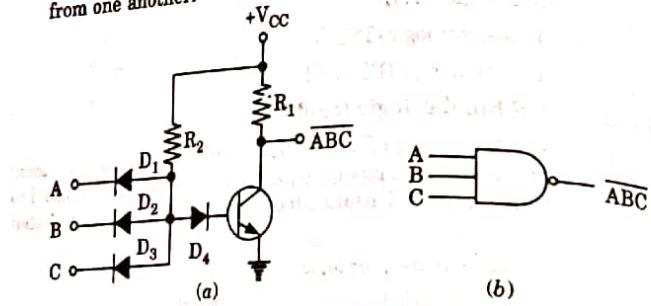


Fig. 5.3.1. DTL NAND gate: (a) Schematic, (b) Symbol.

- 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.
- 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.
- A logic family designed to get around this problem is TTL (transistor-transistor logic) family.
- 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

- 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.

Digital Logic Design

5-5 A (EC/CS/IT-Sem-3)

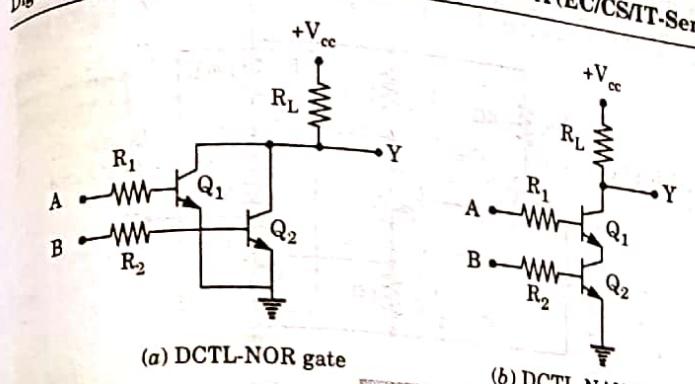


Fig. 5.4.1.

- In DCTL - two input NOR gate of Fig. 5.4.1(a), there are two transistors Q_1 and Q_2 which share a common load resistor R_L .
- 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_L .
- This makes the output voltage at Y to go low. Whereas when both A and B are low, both transistors Q_1 and Q_2 remain in cut-off state and thus the output at Y approaches that of the supply voltage V_{cc} at high level.
- In the DCTL - two input NAND gate of Fig. 5.4.1(b), both the transistors Q_1 and Q_2 are connected in series.
- Only when both inputs A and B are high, both transistors Q_1 and Q_2 will conduct, as they are in series. This causes output voltage at Y to go low.
- When any one or both inputs A and B are low, both transistors Q_1 and Q_2 cannot conduct. Therefore the output voltage at Y remains high.
- The transistor with lower input impedance draws more current than the other transistors.
- 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. Describe the construction and operation of TTL inverter gate (NOT gate).

Answer

- Fig. 5.5.1 shows a standard TTL circuit for an inverter. Transistor Q_1 is the input coupling transistor, and D_1 is the input clamp diode. Transistor Q_2 is called a phase splitter, and the combination of Q_3 and Q_4 forms the output circuit often referred to as a totem-pole arrangement.

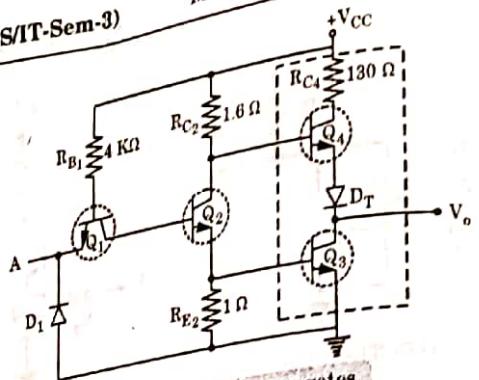


Fig. 5.5.1. TTL inverter gates.

2. When the input is a HIGH, the base-emitter junction of transistor, Q_1 is reverse-biased and the base-collector junction is forward-biased.
3. This condition permits current to flow through R_{B1} and the base-collector junction of transistor, Q_1 into the base of transistor, Q_2 .
4. It derives transistor, Q_2 into saturation. As a result, transistor, Q_3 is turned ON due to ON state of transistor, Q_2 , and its collector voltage, which is the output, is near to ground potential.
5. Therefore a LOW output is produced for a HIGH input. At the same time, the collector of transistor, Q_2 is at a sufficiently LOW voltage level to keep transistor, Q_4 OFF.
6. When the input is LOW, the base-emitter junction of transistor, Q_1 is forward-biased, and the base-collector junction is reverse-biased.
7. The current flows through resistor, R_{B1} and the base-emitter junction of transistor, Q_1 , to the LOW input. A LOW provides a path to ground for the current.
8. No current flows into the base of transistor, Q_2 , so it is OFF. The collector of transistor, Q_2 is HIGH, thus turning transistor, Q_4 ON. A saturated transistor, Q_4 provides a low-resistance path from V_{CC} to the output.
9. Therefore, a HIGH on the output is produced for a LOW on the input. At the same time, the emitter of transistor, Q_2 is at ground potential, keeping transistor, Q_3 OFF.

Table 5.5.1. Operation of TTL inverter.

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

Que 5.8. Describe the construction and operation of TTL NAND gate.

Answer

1. The circuit of the two-input TTL NAND gate is shown in Fig. 5.6.1. The input transistor, Q_1 is a multiple emitter transistor.

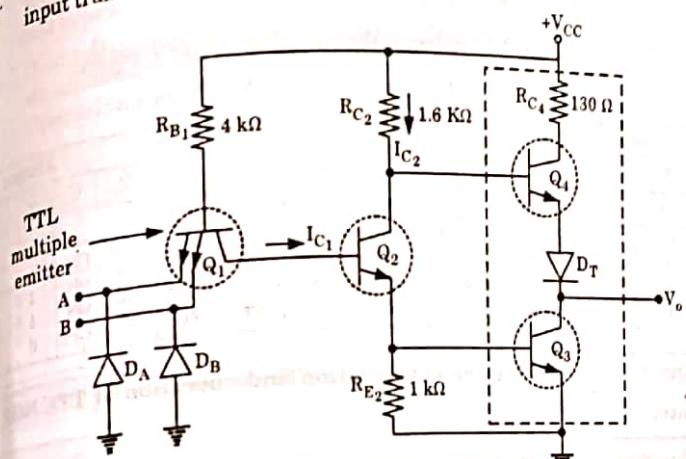


Fig. 5.6.1. TTL NAND gates.

2. Transistor Q_2 is called the phase splitter. Emitter of transistor, Q_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.
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.

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

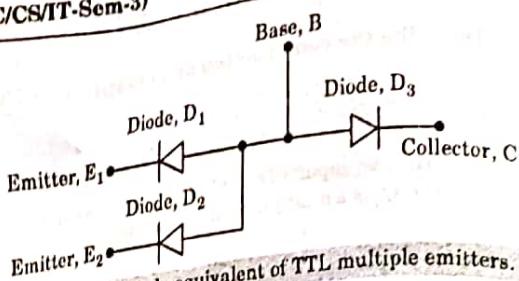


Fig. 5.6.2. Diode equivalent of TTL multiple emitters.

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)	OFF	OFF	ON	1
0	1	Forward bias (ON)	OFF	OFF	ON	1
1	0	Reverse bias (OFF)	OFF	OFF	ON	1
1	1	Reverse bias (OFF)	ON	ON	OFF	0

Que 5.7. Describe the construction and operation of TTL NOR gate.

Answer

TTL NOR gate :

- 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.

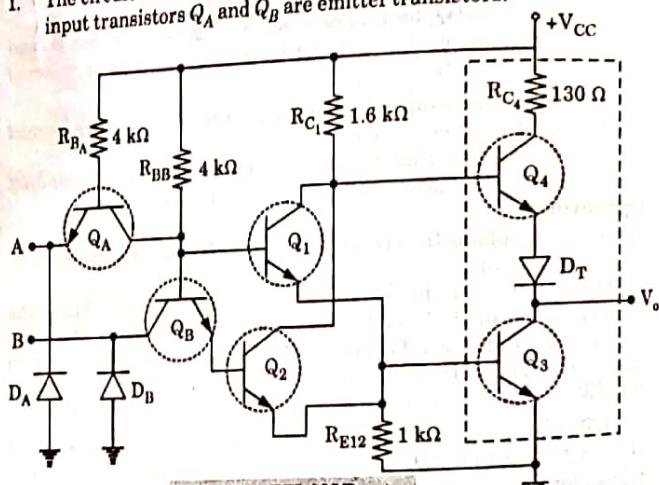


Fig. 5.7.1. TTL NOR gate.

- 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 .
- 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.
- When negative spikes appear at the input terminals, the diodes conduct and bypass the spikes to ground.
- 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_3	Q_4	V_o
		Emitter junction, A	Emitter junction, B			
0	0	Forward bias (ON)	OFF	OFF	OFF	1
0	1	Forward bias (ON)	OFF	ON	ON	0
1	0	Reverse bias (OFF)	OFF	OFF	ON	1
1	1	Reverse bias (OFF)	ON	ON	OFF	0

Que 5.8. State various TTL parameters in brief.

Answer

1. **Current sinking:** A TTL circuit acts as a current sink in LOW state, as it receives current from the input of the gate by which it is driving. Transistor Q_3 is the current-sinking transistor or the pull-down transistor, because it brings the output voltage down to its LOW state.

2. **Current sourcing:** A TTL circuit acts as a current source in the HIGH state, as it supplies current to the gate by which it is driving. Transistor Q_4 is the current-sourcing transistor or the pull-up transistor, because it pulls up the output voltage to its HIGH state.

3. **Floating inputs:** When a TTL input is HIGH (ideally +5 V), the emitter current is approximately zero. When a TTL input is floating no emitter current is possible because of the open circuit. Therefore, a floating TTL input is equivalent to a HIGH output. Because of this, unused TTL inputs are left unconnected : an open input allows the rest of the gate to function properly.

4. TTL loading and fan-out :

The TTL output has a limit, I_{OL} that gives the maximum current it can sink in LOW state and a limit, I_{OH} , gives the maximum current it can source in HIGH state. To determine the fan-out, the drive capabilities of the output, i.e., I_{OL} and I_{OH} and the current requirements of each input, i.e., I_{IL} and I_{IH} are known. So, HIGH and LOW state fan-outs are given by :

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

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

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

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

5. **Unit load :**
Unit load means the current drawn or sourced back by similar gates.
Example : For 7400,
One unit load is $40 \mu\text{A}$ in HIGH state that is known as I_{IH} .
One unit load is 1.6 mA in LOW state that is known as I_{IL} .
- Que 5.9.** The output of 7400 IC is rated as $I_{OH} = 800 \mu\text{A}$ and $I_{OL} = 48 \text{ mA}$. Determine HIGH state and LOW state fan-outs and actual fan-out capability.

Answer

- Given, $I_{OH} = 800 \mu\text{A}$
 $I_{OL} = 48 \text{ mA}$
- HIGH state fan-out = $\frac{I_{OH}}{I_{IH}} = \frac{800 \mu\text{A}}{40 \mu\text{A}} = 20$ unit loads
- LOW state fan-out = $\frac{I_{OL}}{I_{IL}} = \frac{48 \text{ mA}}{1.6 \text{ mA}} = 30$ unit loads
- Actual fan-out capability = $\min \left\{ \frac{I_{OH}}{I_{IH}}, \frac{I_{OL}}{I_{IL}} \right\}$
= $\min \{20, 30\} = 20$ units

- Que 5.10.** Discuss about emitter coupled logic (ECL). What are its characteristics ?

Answer

- 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.
- Its fast speed of operation is due to following reasons :
- 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 voltage swing.
- iv. 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.
- v. 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.

Characteristics :

- Transistors never saturate. So, speed is high with propagation delay, t_{pd} is 360 ps.
- Noise margin is less than near to 250 mV. So, ECL is unreliable for the use in heavy industrial environment.
- Eliminates the use of inverters because ECL circuits produce the output and its complement.
- Fan-out is large because the output impedance is low. Typically, it is almost 25.
- Power dissipation per gate is large, P_D is 40 mW.
- The total current that flows in ECL is more or less constant. So, no noise spikes are generated internally.

- Que 5.11.** Explain the basic circuit and operation of emitter coupled logic (ECL). What are the function of emitter follower ?

Answer**Basic ECL circuit :**

- The basic circuit for emitter-coupled logic is a differential amplifier configuration as shown in Fig. 5.11.1.

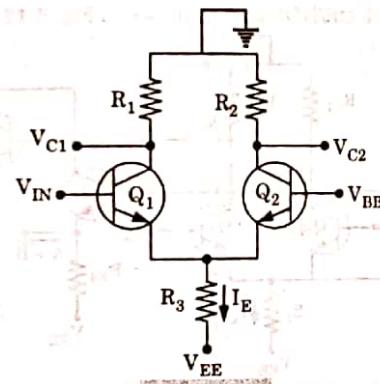


Fig. 5.11.1.

- The V_{EE} supply produces a fixed current I_E , which remains around 3 mA during normal operation. This current is allowed to flow through either transistor Q_1 or transistor Q_2 , depending on the voltage level at V_{IN} .
- In other words, this current switches between collector of Q_1 transistor and collector of Q_2 transistor as V_{IN} switches between its two logic levels of -1.7 V (logical 0 for ECL) and -0.8 V (logical 1 for ECL).
- Table 5.11.1 shows the resulting output voltages for these two conditions at V_{IN} .

Table 5.11.1. Operating states of ECL

V_{IN}	Outputs		Remarks
	Binary logic	V_{C1}	
-1.7 V	Logic 0 Logic 1	0 V -0.9 V	Q_2 conducts Q_1 conducts
-0.8 V		0 V	

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

Que 5.12. Describe the construction and operation of ECL OR/NOR gate.

Answer

ECL OR/NOR gate :

- A two-input ECL OR/NOR gate is shown in Fig. 5.12.1.

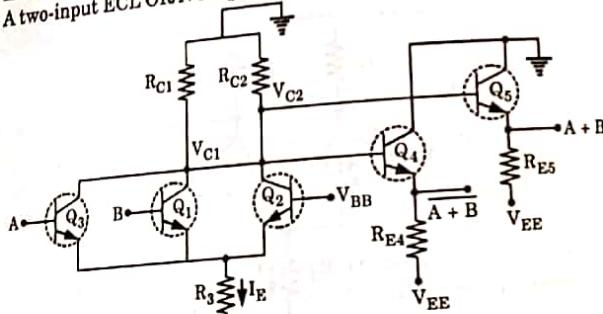


Fig. 5.12.1. ECL NOR and OR gates.

- 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.
- 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).
- Inputs are applied to transistors Q_1 and Q_3 , and transistor Q_2 is supplied with constant -1.3 V.

Table 5.12.1. Operation of ECL OR/NOR gate.

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

Que 5.13. Describe the circuit and performance of CMOS inverter and state the characteristics of CMOS.

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 V), Gate to source voltage V_{GS2} of transistor Q_2 is -5 V, and gate to source voltage V_{GS1} of transistor Q_1 is 0 V. So, transistor Q_2 acts as ON and transistor Q_1 acts as OFF. Therefore, the switching circuit shown in Fig. 5.13.1(b) results in V_o as logic HIGH that is +5 V.

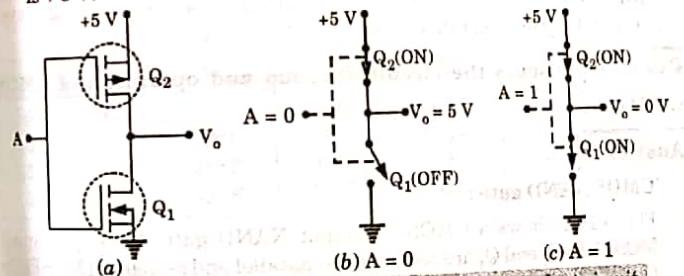


Fig. 5.13.1. (a) CMOS as inverter, (b) and (c) equivalent circuit.

Memory & Programmable Logic Devices

5-14 A (EC/CS/IT-Sem-3)

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

Table 5.13.1. Operation of CMOS inverter.

Input, A	p-channel MOSFET, Q_1	n-Channel MOSFET, Q_2	Output, V_o
LOW (0 V)	ON	OFF	+5 V (HIGH)
HIGH (5 V)	OFF	ON	0 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 V. The 74HC and 74HCT series can operate with V_{DD} values ranging from 2 to 6 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 nW per gate when $V_{DD} = 5$ V, and it increases to 10 nW per gate when $V_{DD} = 10$ 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 V or V_{DD}) or to another input.

Que 5.14. Discuss the circuit diagram and operation of CMOS NAND gate.

Answer

CMOS NAND gate :

1. Fig. 5.14.1 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.

5-15 A (EC/CS/IT-Sem-3)

Digital Logic Design

2. When A is LOW (0 V) and B is also LOW (0 V), p-channel MOSFET Q_1 acts ON, n-channel MOSFET Q_3 acts OFF, p-channel MOSFET Q_2 acts ON and n-channel MOSFET Q_4 acts OFF. Thus, the switching circuit results V_o as logic HIGH i.e., +5 V.
3. When A is LOW (0 V) and B is HIGH (5 V), p-channel MOSFET Q_1 acts ON, n-channel MOSFET Q_3 acts OFF, p-channel MOSFET Q_2 acts OFF and n-channel MOSFET Q_4 acts ON. Thus, the switching circuit results V_o as logic HIGH i.e., +5 V.
4. When A is HIGH (+5 V) and B is LOW (0 V), p-channel MOSFET Q_1 acts OFF, NMOSFET Q_3 acts ON, p-channel MOSFET Q_2 acts ON and n-channel MOSFET Q_4 acts OFF. Thus, the switching circuit results V_o as logic HIGH i.e., +5 V.
5. When A is HIGH (+5 V) and B is also HIGH (+5 V), p-channel MOSFET Q_1 acts OFF, NMOSFET Q_3 acts ON, p-channel MOSFET Q_2 acts OFF and n-channel MOSFET Q_4 acts ON. Thus, the switching circuit results V_o as logic LOW i.e., 0 V.

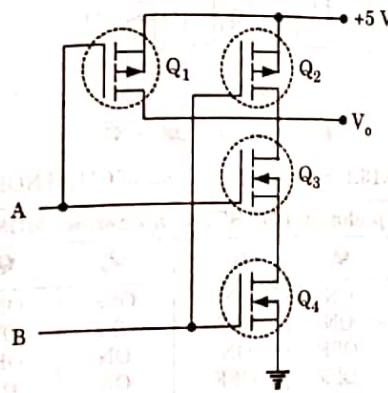


Fig. 5.14.1. CMOS as NAND gate.

Table 5.14.1. Switching operation of CMOS NAND gate

Inputs		p-channel MOSFET	n-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	1
1	0	OFF	ON	ON	OFF	1
1	1	OFF	OFF	ON	ON	0

Que 5.15. Discuss the circuit diagram and operation of CMOS NOR gate.

Answer

CMOS NOR gate :
Fig. 5.15.1 shows a CMOS two-input NOR gate. Here, *p*-channel MOSFETs Q_1 and Q_2 are connected in series and *n*-channel MOSFETs Q_3 and Q_4 are connected in parallel.

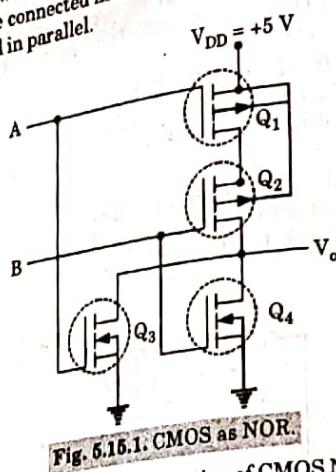


Fig. 5.15.1. CMOS as NOR.

Table 5.15.1. 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 5.16. Discuss the circuit and operation of CMOS transmission gate.

Answer

- Transmission gate :**
1. A transmission gate is simply a digitally controlled CMOS switch. When the switch is open (OFF), the impedance between its terminals is very large.
 2. It is used to implement special logic functions. Since the CMOS gate can transmit signals in both directions, it is called a bilateral transmission gate or bilateral switch.
 3. It is useful for digital and analog applications. The TTL and ECL gates are essentially unidirectional.

5. Fig. 5.16.1 shows the schematic diagram and logic symbols of a CMOS transmission gate. The *n*-channel MOS and *p*-channel MOS transistors are connected in parallel.
6. So, both polarities of input voltages can be switched. The control signal, *C* is connected to the *n*-channel MOSFET and its inverse is connected to the *p*-channel MOSFET.

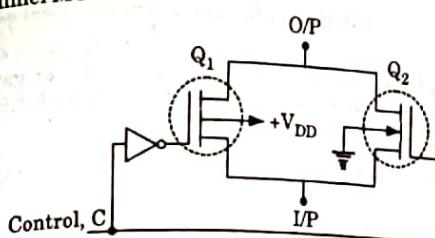


Fig. 5.16.1. CMOS transmission gate.

Table 5.16.1. Operation of CMOS transmission gate

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

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

Que 5.17. Describe the fan-out and fan-in condition of the digital logic gate.

Answer**Fan-out :**

1. The fan-out of a logic gate is defined as the maximum number of standard load that the output of the gate can drive without impairing its normal operation. Fan-out is also called the loading factor.

5-18 A (EC/CS/IT-Sem-3)

Memory & Programmable Logic Devices

2. HIGH state fan-out is the fan-out of the gate when its output is logic 1. LOW state fan-out is the fan-out of the gate when its output is logic 0. The smaller of these two numbers is taken as the actual fan-out.

3. High state fan-out is given by

$$\text{HIGH state fan-out} = \frac{I_{OH}}{I_{IH}} \quad \dots(5.17.1)$$

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.

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

$$\text{LOW state fan-out} = \frac{I_{OL}}{I_{IL}} \quad \dots(5.17.2)$$

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.

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

$$\text{Fan-out} = \text{minimum of } \left\{ \frac{I_{OH}}{I_{IH}}, \frac{I_{OL}}{I_{IL}} \right\} \quad \dots(5.17.3)$$

Fan-in :

- 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.
- A logic designer has to select the fan-in of the gate to accommodate the number of inputs.
- At the hardware level, however, the fan-in provides information about the intrinsic speed of the gate itself.
- 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.

Que 5.18. What do you understand by noise margin of logic circuit? Explain with an example.

Answer

- 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.

Digital Logic Design

5-19 A (EC/CS/IT-Sem-3)

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. 5.18.1.

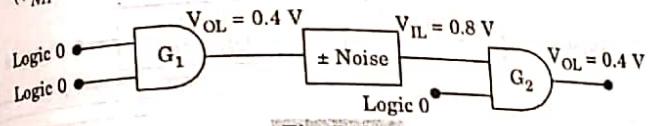


Fig. 5.18.1.

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. 5.18.2. This output acts as input for gate, G_2 . Due to noise, actual input given to gate, G_2 is

$$V_{NH} = V_{OH} - V_{IH}$$

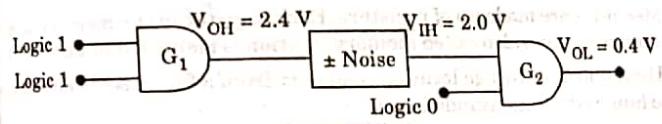


Fig. 5.18.2.

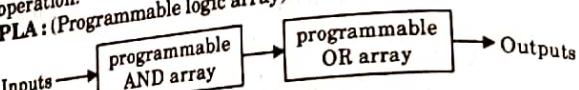
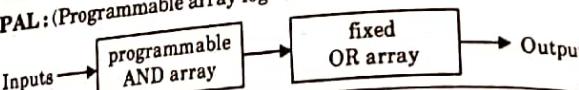
V_{IL} in above equation that acts as input to gate G_2 . Minimum high level noise level is

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

PART-2

RAM, ROM, PLA, PAL.

CONCEPT OUTLINE : PART-2

- RAM :** A RAM stores information or data temporarily that can be read from memory as well as modified or written onto memory.
- ROM :** It is essentially a memory device in which permanent binary information is stored. ROM can perform only the read operation.
- PLA :** (Programmable logic array)
 
- PAL :** (Programmable array logic)
 

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

Que 5.19. What is memory ? Write down the classification of semiconductor memories.

Answer**Memory :**

- Memories are made up of registers. Each register in the memory is one storage location also called memory location is identified by an address.
- The number of storage locations can vary from a few in some memories to hundreds of thousands in others.



Fig. 5.19.1.

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. 5.19.2 shows an overview of semiconductor memory types.

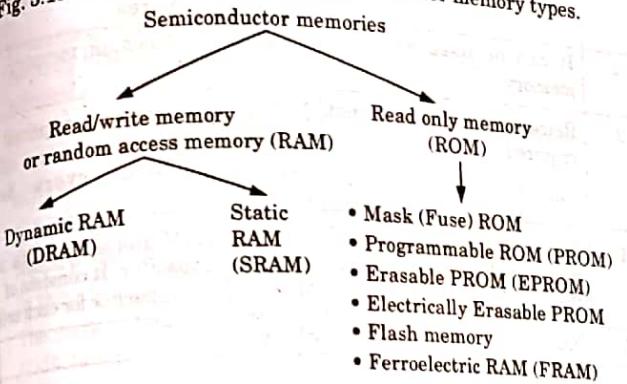


Fig. 5.19.2. Classification of semiconductor memories.

Que 5.20. What is RAM ? Distinguish between SRAM and DRAM.

Also draw static RAM cell.

AKTU 2012-13, Marks 10

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.
- Based on the operation, RAM's are classified into two main categories.
- Dynamic RAM :** The DRAM cell consists of a capacitor to store binary information '1' (high voltage) or '0' (low voltage) and a transistor to access the capacitor. Due to the advantage of low cost and high density, DRAM is widely used for main memory.

- Static RAM :** The SRAM cell consists of a latch therefore, the cell data is kept as long as the power is turned on and refresh operation is not required as in case of DRAM cells. SRAM is mainly used for the cache memory in applications.

5-22 A (EC/CS/IT-Sem-3) Memory & Programmable Logic Devices

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.

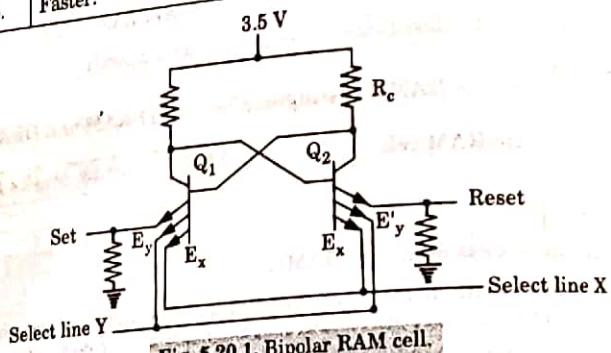


Fig. 5.20.1. Bipolar RAM cell.

Que 5.21. What is ROM ? What are the various types of ROM ?

AKTU 2011-12, Marks 10

Explain.

Answer

ROM:

1. It is the memory device which stores the binary information permanently.
2. It allows only retrieval of permanently stored data and does not permit modifications of the stored information during normal operation.
3. ROMs are non-volatile memories, i.e., the stored data are not lost even when the power supply is off and refresh operation is not required.

Digital Logic Design

5-23 A (EC/CS/IT-Sem-3)



Fig. 5.21.1. ROM block diagram.

4. It consists of k inputs and n outputs. The inputs are the address of memory and the outputs are the data bits of the stored word that is selected by the address.
5. If there are 2^k words then k address input lines are needed to specify the words.
6. ROM does not have data inputs, because it does not have write operation. Following are the various types of ROM :

Mask programmable ROMs :

1. In this, the data is programmed as a part of manufacturing process.
2. The mask ROM is manufactured according to the user's need and the truth table is designed and then embedded in the ROM. It is not reprogrammable.
3. Its manufacturing is costly because the custom masking is done according to the user's needs.
4. It is economical only if a large quantity of the same ROM configuration is to be used.

Programmable ROMs (PROM) :

1. This device is electrically programmable. In this, users can program the chip according to their needs.
2. These are one time programmable (OTP), because the procedure for programming PROMs is irreversible.
3. After the ROMs are programmed, the fixed pattern is permanent and cannot be altered.

Erasable PROMs (EPROM) :

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.

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. What is memory? Discuss the memory organization and operation using diagram of a 32×4 memory and arrangement of memory cells.

AKTU 2013-14, Marks 10

Answer

Memory : Refer Q. 5.19, Page 5-20A, Unit-5.

Logic construction of ROM :

- Fig. 5.22.1 depicts the logic construction of 32×4 ROM. ROM is a combinational circuit with AND gates connected as a decoder and a number of OR gates equal to the number of outputs in the unit.
- The five input variables are decoded into 32 lines by use of 32 AND gates and 5 NOT gates.
- The output of each decoder illustrates one of the minterms of a function of five variables.
- Each one of the 32 address selects one and only one output of the decoder which is connected through the line to each OR gate.
- ROMs are very important unit in the design of digital computer systems.

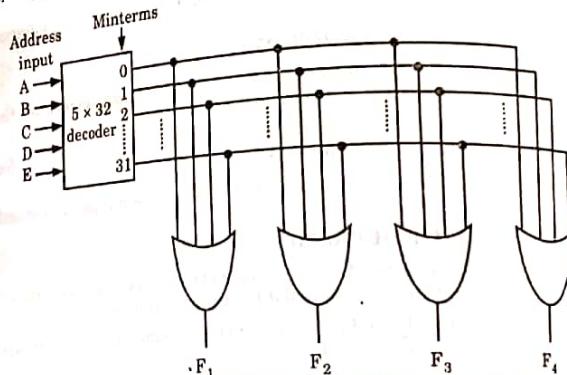


Fig. 5.22.1. Logic construction of 32×4 ROM.

Que 5.23. 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	1
0	1	0	0	0	0	1	0	0
0	1	1	0	0	1	0	0	1
1	0	0	0	1	0	0	0	0
1	0	1	0	1	1	0	0	1
1	1	0	1	0	0	1	0	0
1	1	1	1	1	0	0	0	1

For D₀

B ₁ B ₀	$\bar{B}_1\bar{B}_0$	\bar{B}_1B_0	B ₁ B ₀	B ₁ \bar{B}_0
\bar{B}_2	0	1	1	2
\bar{B}_2	4	5	1	6

$$D_0 = B_0$$

For D₁

B ₁ B ₀	$\bar{B}_1\bar{B}_0$	\bar{B}_1B_0	B ₁ B ₀	B ₁ \bar{B}_0
\bar{B}_2	0	1	3	2
\bar{B}_2	4	5	7	6

$$D_1 = 0$$

For D₂

B ₁ B ₀	$\bar{B}_1\bar{B}_0$	\bar{B}_1B_0	B ₁ B ₀	B ₁ \bar{B}_0
\bar{B}_2	0	1	3	2
\bar{B}_2	4	5	7	6

$$D_2 = B_1\bar{B}_0$$

For D₃

B ₁ B ₀	$\bar{B}_1\bar{B}_0$	\bar{B}_1B_0	B ₁ B ₀	B ₁ \bar{B}_0
\bar{B}_2	0	1	3	2
\bar{B}_2	4	5	7	6

$$D_3 = \bar{B}_2B_1B_0 + B_2\bar{B}_1B_0$$

For D₄

B ₁ B ₀	$\bar{B}_1\bar{B}_0$	\bar{B}_1B_0	B ₁ B ₀	B ₁ \bar{B}_0
\bar{B}_2	0	1	3	2
\bar{B}_2	4	5	7	6

$$D_4 = B_2\bar{B}_1 + B_2B_0$$

For D₅

B ₁ B ₀	$\bar{B}_1\bar{B}_0$	\bar{B}_1B_0	B ₁ B ₀	B ₁ \bar{B}_0
\bar{B}_2	0	1	3	2
\bar{B}_2	4	5	7	6

$$D_5 = B_1B_2$$

Combinational circuit of the function using ROM/PROM can be implemented as :

$$D_0 = B_0$$

$$D_1 = 0$$

$$D_2 = B_1\bar{B}_0$$

$$D_3 = \bar{B}_2B_1B_0 + B_2\bar{B}_1B_0$$

$$D_4 = B_2\bar{B}_1 + B_2B_0$$

$$D_5 = B_1B_2$$

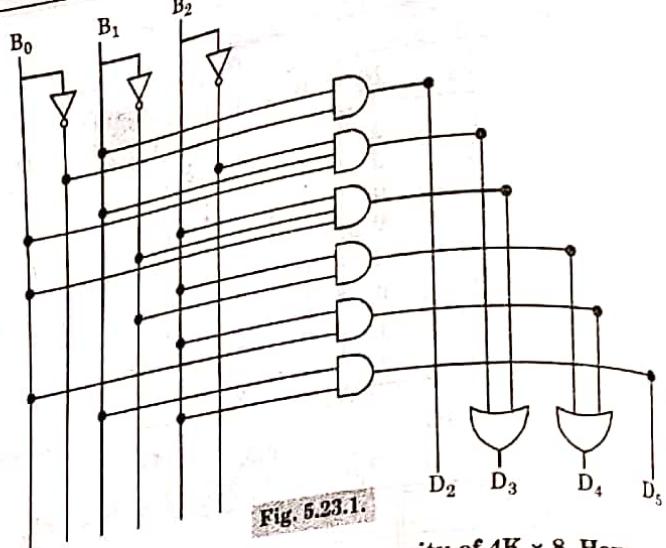


Fig. 5.23.1.

Que 5.24. A certain memory has a capacity of $4K \times 8$. How many data input and data output lines does it have? How many address lines does it have? What is its capacity in bytes?

Answer

$$\text{Number of data input lines} = 8 \text{ lines}$$

$$\text{Number of data output lines} = 8 \text{ lines}$$

$$\text{Number of address lines} = 4K = 4096 = 2^{12} = 2^n$$

$$\text{Address lines, } n = 12$$

Its capacity of accommodating is 4096 bytes.

Que 5.25. How many address lines and input-output data lines are needed in $2M \times 8$ memory unit?

Answer

$$1M = 2^{20}$$

$$2M \times 8 = 2^1 2^{20} \times 8 = 2^{21} \times 8$$

$$\therefore \text{Number of address lines} = 21$$

$$\text{Number of input-output data lines} = 8$$

Que 5.26. Explain how a multiplexer can be used as ROM?

AKTU 2012-13, Marks 10

Answer

1. The multiplexer, also called the data selector, has n select inputs, 2^n input lines and 1 output line.
2. The 2^n possible combinations of the select inputs connect one of the input lines to the output.
3. When used as a combinational logic device, the n select input represents n variables and the 2^n input lines represent all the minterms of the n -variables. 4×1 multiplexer is shown in Fig. 5.26.1

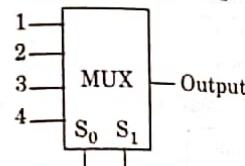
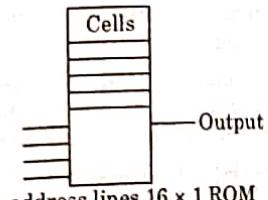


Fig. 5.26.1.

4. ROM is a memory unit that performs the read operation only. This implies that the binary information stored in a ROM is made permanent during the hardware production of the unit.
5. An $2^n \times m$ ROM is an array of binary cells organised into 2^n words of m bits each. As shown in the Fig. 5.26.2, a ROM has 2^n input lines to select one of the 2^n words of the memory and m output lines, one for each bit of the word.



address lines 16 x 1 ROM

Fig. 5.26.2.

6. In ROM, the function is implemented directly. It is extremely flexible and can be reprogrammed to represent an entirely different function. Fig. 5.26.3 shows a ROM cell with the help of multiplexer.

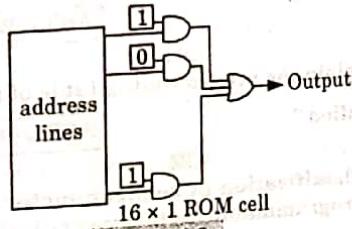


Fig. 5.26.3.

Que 5.27. Draw the basic configuration of three PLDs.

AKTU 2016-17, Marks 10

Answer

1. The PROM is a combinational programmable logic device (PLD), an integrated circuit with programmable gates divided into an AND array and an OR array to provide an AND-OR sum-of-product implementation. Fig. 5.27.1 shows the configuration of the three PLDs.

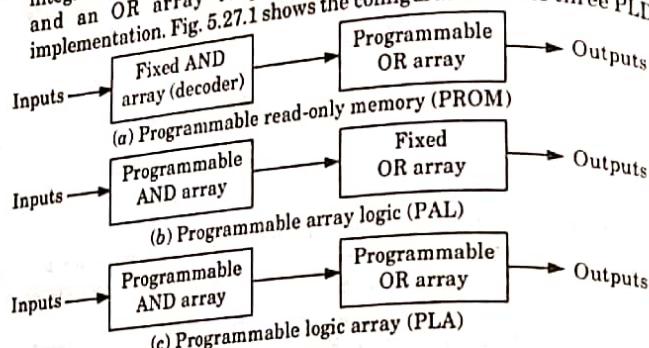


Fig. 5.27.1.

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

Que 5.28. What is the basic architecture of a PLA? How is the capacity of a PLA specified? How is it programmed? Explain.

AKTU 2011-12, Marks 10

OR

What is PLA? Explain the programming table of a PLA. How is the size of a PLA specified?

AKTU 2012-13, Marks 10

OR

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

AKTU 2014-15, Marks 06

Answer

Classification of semiconductor memories : Refer Q. 5.19, Page 5-20A, Unit-5.

Programmable logic array (PLA) :

1. PLAs are used to map irregular combinational function onto regular structures. The PLA provides the designer with a systematic and regular way of implementing output functions of n variable in sum of product form.
2. PLA is one of the regular macro used in the implementation of FSM (finite state machine). PLA functions may be significantly changed more compact in nature. Any of the logical function can be expressed in terms of SOP or POS.
3. PLA can be implemented in several forms, i.e., NOR-NOR, NAND-NAND, NAND-NOR.
4. The structure of PLA is shown in Fig. 5.28.1, and its internal logic with three inputs and two outputs is shown in Fig. 5.28.2.
5. The particular boolean functions implemented in the PLA of Fig. 5.28.2, are

$$F_1 = AB' + AC + A'BC'$$

$$F_2 = (AC + BC)'$$

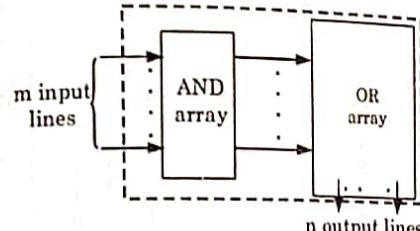


Fig. 5.28.1. PLA structure.

6. The programming table that specifies the PLA of Fig. 5.28.2 is listed in the table 5.28.1. The PLA programming table consists of three sections.
7. The first section lists the product term numerically. The second section specifies the required paths between input and AND gates. The third section specifies the path between the AND and OR gates.
8. For each output variable, we may have a T (true) or C (complement) for programming the XOR gate.
9. The product terms listed on the left are not part of the table, they are included for reference only. For each product term, the inputs are marked with 1, 0 or — (dash).

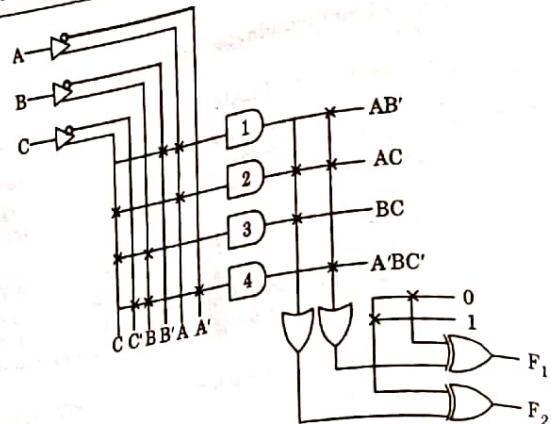


Fig. 5.28.2. PLA with three inputs, four product terms, and two outputs.
10. If the variable in the product term appears in the form in which it is true, the corresponding input variable is marked with a 1. If it appears complemented, the corresponding input variable is marked with a 0. If the variable is absent from the product term, it is marked with a dash.

Table 5.28.1. PLA programming

Product term	Input			Output	
	A	B	C	(T) F_1	(C) F_1
AB'	1	0	—	1	—
AC	2	1	—	1	1
BC	3	—	1	1	—
$A'BC'$	4	0	1	0	1

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

Que 5.29. Draw the logic configuration of four input and four output PAL and explain.

AKTU 2011-12, Marks 10

Answer

1. A programmable array logic has the same structure as a ROM, but has a programmable AND array and a fixed OR (or NOR) array.

2. Because of the fixed OR array, a PAL device is cheaper comparatively and easier to program.
3. However, the lack of shared rows with the column requires that each output function be simplified, with no common product term with others. It is easier to program, but is not flexible.

PAL with the four input and four output :

1. Each input has a buffer-inverter gate and each output is generated by fixed OR gate.

AND gate input

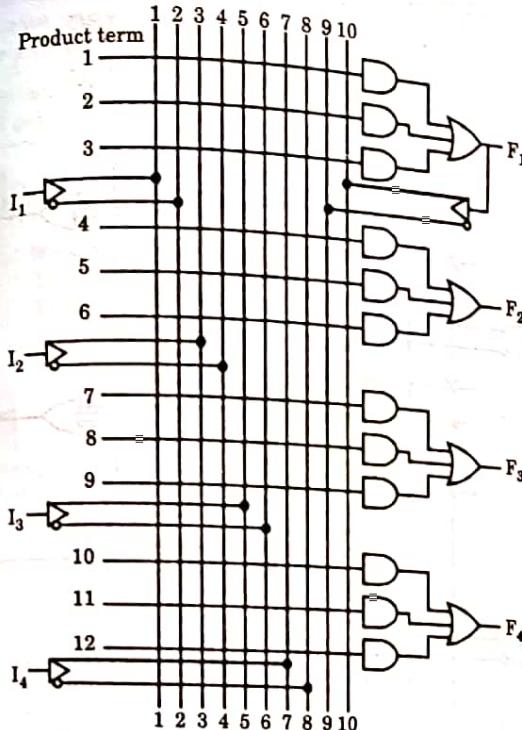


Fig. 5.29.1. PAL with four inputs, four outputs and a three wire AND-OR structure.

2. In designing with a PAL, the boolean function must be simplified to fit into each section unlike the situation with a PLA, a product term cannot be served among two or more OR gates. Therefore, each function can be simplified by itself, without regarding common product terms.

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

Que 5.30. Realize the full adder circuit using the PAL.

Answer

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

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

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

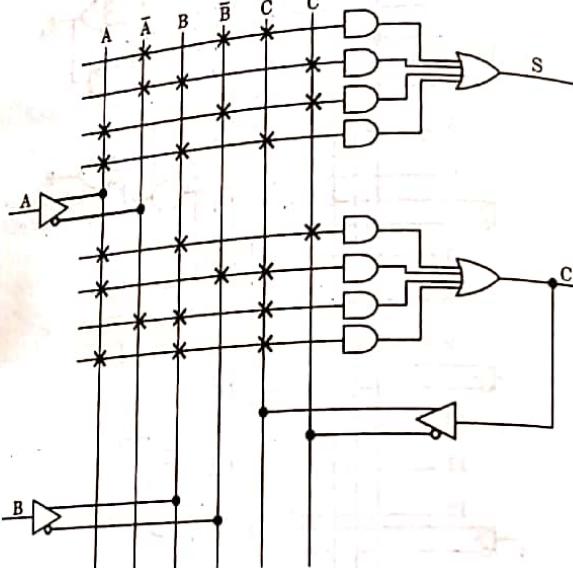


Fig. 5.30.1.

Que 5.31. Differentiate between PAL and PLA.

Answer

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.

Que 5.32. Implement the following four boolean functions with a PAL

$$W(A, B, C, D) = \Sigma(2, 12, 13)$$

$$X(A, B, C, D) = \Sigma(7, 8, 9, 10, 11, 12, 13, 14, 15)$$

$$Y(A, B, C, D) = \Sigma(0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15)$$

$$Z(A, B, C, D) = \Sigma(1, 2, 8, 12, 13)$$

AKTU 2016-17, Marks 15

Answer

1. Simplifying the four functions to a minimum number of terms results in the following boolean functions :

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

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

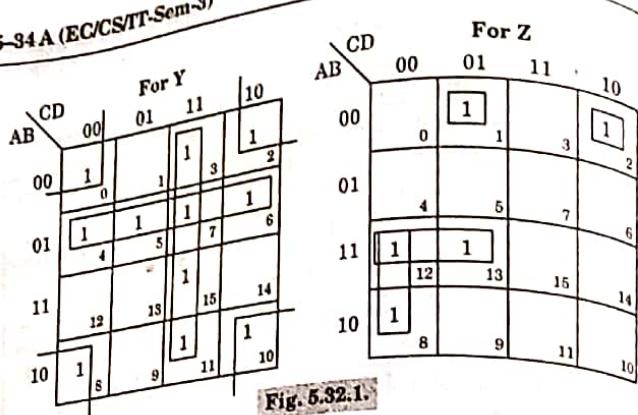


Fig. 5.32.1.

$$W = ABC' + A'B'C'D'$$

$$X = A + BCD$$

$$Y = A'B + CD + B'D'$$

$$Z = ABC' + A'B'CD' + AC'D' + A'B'C'D \\ = W + AC'D' + A'B'C'D$$

2. Table 5.32.1 lists the PAL programming table for the four boolean functions. The table 5.32.1 is divided into four sections with three product terms in each section.

3. The first two sections need only two product terms to implement the boolean function. The last section for output Z needs four product terms. Using the output from W, we can reduce the function to three terms.

4. The fuse map for the PAL as specified in the programming Table 5.32.1 is shown in Fig. 5.32.1 for each 1 or 0 in the Table 5.32.1.

5. We mark the corresponding intersection in the diagram with the symbol for an intact fuse. For each dash, we mark the diagram with blown fuses in both the true and complement inputs.

6. If the AND gate is not used we leave all its input fuse intact. Since the corresponding input receives both the true value and the complement of each input variable, we have $AA' = 0$ and the output of the AND gate is always 0.



Table 5.32.1. PAL programming

Product Term	AND Input					Outputs	
	A	B	C	D	W		
1	1	1	0	—	—	$W = ABC' + A'B'CD'$	
2	0	0	1	0	—	$X = A + BCD$	
3	1	—	—	—	—	$Y = A'B + CD + B'D'$	
4	—	1	1	—	—	$Z = W + AC'D' + A'B'C'D$	
5	—	—	—	1	—		
6	0	1	—	—	—		
7	—	—	1	—	—		
8	—	0	—	1	—		
9	—	—	—	0	—		
10	1	—	0	0	1		
11	0	0	0	1	—		
12	—	—	—	—	—		

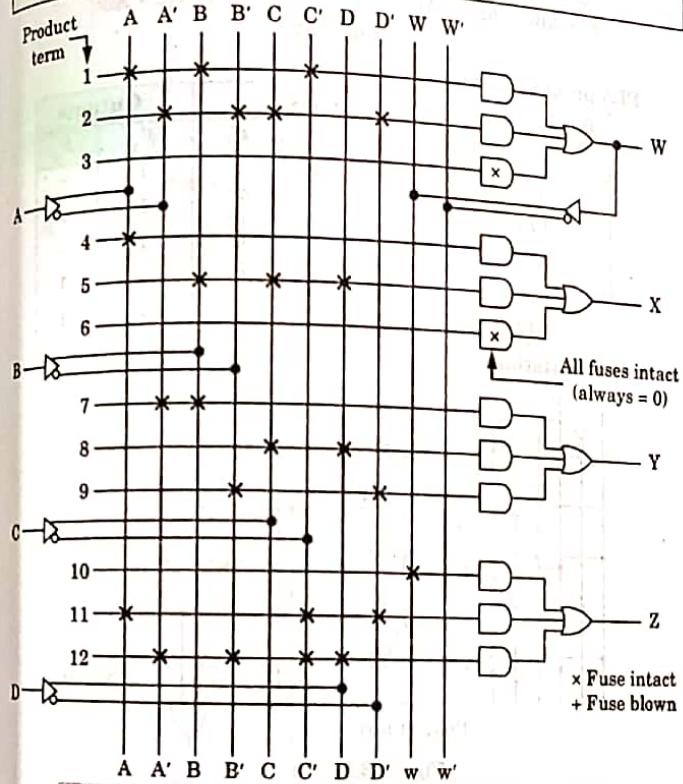


Fig. 5.32.1. Fuse map for PAL as specified in Table 5.32.1.

5-36 A (EC/CS/IT-Sem-3)

Memory & Programmable Logic Devices

Que 5.33. A combinational circuit is defined by the functions :

$$F_1(A, B, C) = \Sigma(3, 5, 6)$$

$$F_2(A, B, C) = \Sigma(0, 2, 7)$$

Implement the circuit with a PLA.

AKTU 2013-14, Marks 10

Answer

Simplify the given boolean expression

$$F_1(A, B, C) = \Sigma(3, 5, 6)$$

$$F_2(A, B, C) = \Sigma(0, 2, 7)$$

For F_1 :

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

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

BC		For F_2 :			
A	00	01	11	10	
0	0	1	3	1	2
1	4	5	7	6	

$$F_2 = \bar{A}\bar{C} + ABC$$

Fig. 5.33.1.

PLA program table:

Product term	Inputs			Outputs	
	A	B	C	F_1	F_2
$\bar{A}BC$	0	1	1	1	-
$A\bar{B}C$	1	0	1	1	-
ABC	1	1	0	1	-
$\bar{A}\bar{C}$	0	-	0	-	1
ABC	1	1	1	-	1

Implementation:

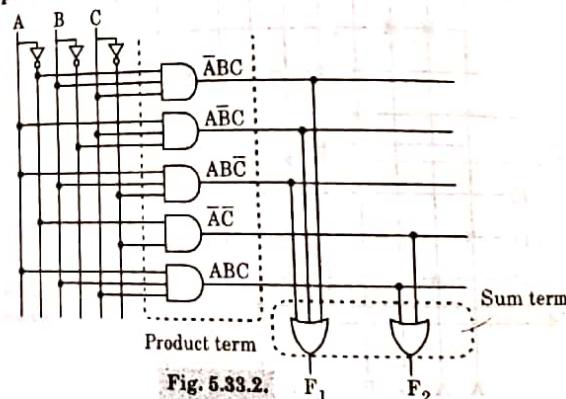


Fig. 5.33.2.

5-37 A (EC/CS/IT-Sem-3)

Digital Logic Design

Que 5.34. Design a 3-bit binary to Gray code converter using PLA.

AKTU 2015-16, Marks 10

Answer
Truth table :

Binary input			Gray output		
B_2	B_1	B_0	G_2	G_1	G_0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

Simplification using K-map :

The K-maps for the Gray outputs are as shown in Fig. 5.34.1.

For G_2

$B_1 B_0$		00	01	11	10
B_2		0	0	0	0
0	0	0	1	0	2
1	1	1	1	1	6

(a)

For G_1

$B_1 B_0$		00	01	11	10
B_2		0	0	0	0
0	0	0	0	1	2
1	1	1	1	0	6

(b)

$B_1 B_0$		00	01	11	10
B_2		0	0	0	0
0	0	1	1	0	2
1	0	1	0	0	6

(c)

Fig. 5.34.1.

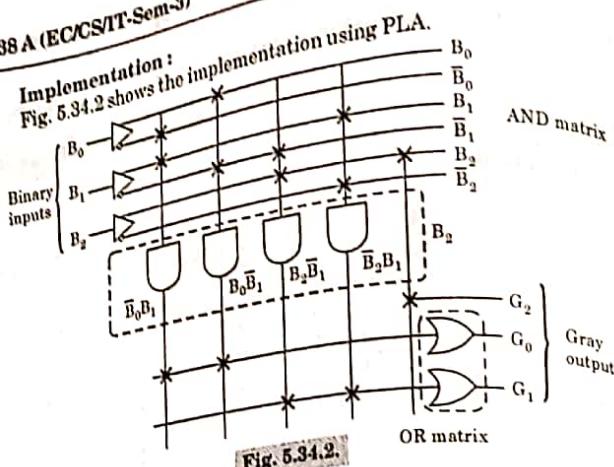
$$G_2 = B_2$$

$$G_1 = \bar{B}_2 B_1 + B_2 \bar{B}_1$$

$$G_0 = \bar{B}_1 B_0 + B_1 \bar{B}_0$$

Memory & Programmable Logic Devices

5-38 A (EC/CS/IT-Sem-3)



PART-3

Circuits of Logic Families, Interfacing of Digital Logic Families, Circuit Implementation Using ROM, PLA, PAL, CPLD, FPGA.

CONCEPT OUTLINE : PART-3

- Interfacing means connecting two different systems or devices, having different electrical characteristics.
- Interface circuit used to connect driver and load circuit.
- **Complex programmable logic device (CPLD)** : CPLD contains a bunch of PLD blocks whose inputs and outputs are connected together by a global interconnection matrix.
- **Field programmable gate array (FPGA)** : FPGA is a high capacity PLD.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Ques 5.35. Explain TTL to CMOS interfacing and CMOS to TTL interfacing.

5-39 A (EC/CS/IT-Sem-3)

Digital Logic Design

Answer

- TTL to CMOS :**
- 1 The MOS and CMOS gates are slower than the TTL gates, but consume less space. Hence, there is an advantage in using TTL and MOS devices in combination.
 - 2 The input current values of CMOS are low as compared to the output current capabilities of any TTL series. Thus, TTL has no problem in meeting the CMOS input current requirements.
 - 3 So, a level translator is used to raise the level of the output voltage of the TTL gate to an acceptable level for CMOS.
 - 4 The presence of the pull-up resistor will cause the TTL output to rise to approximately +5 V in the HIGH state, thereby providing an adequate CMOS input shown in Fig. 5.35.1.

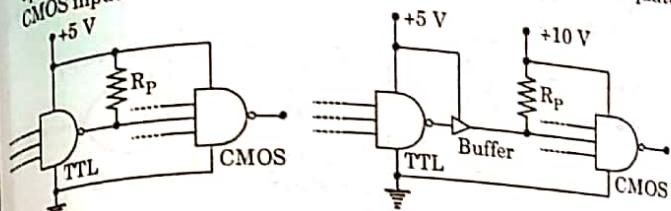


Fig. 5.35.1.

CMOS to TTL :

- 1 The CMOS output can supply enough voltage and current to satisfy the TTL input requirements in the HIGH state. Hence, no special consideration is required for the HIGH state.
- 2 But the TTL input current requirements at LOW state cannot be met directly.
- 3 Therefore, an interface circuit with a LOW input current requirement and a sufficiently high output current rating is required. A CMOS buffer serves this purpose. The arrangement is shown in Fig. 5.35.2.
- 4 When a high voltage CMOS has to drive a TTL gate, a voltage level translator that converts the high voltage input to a +5 V output is used between CMOS and TTL as shown in Fig. 5.35.2.

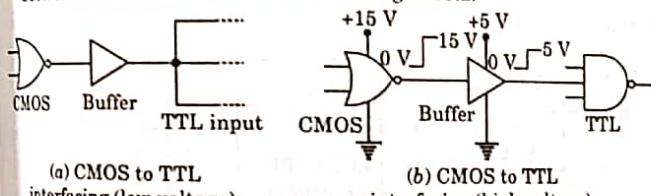


Fig. 5.35.2.

Que 5.36. Explain the interfacing of TTL to ECL and ECL to TTL.

Answer

- TTL to ECL:**
1. The TTL is the most widely used logic family, but its speed of operation is not very high.
 2. The ECL is the fastest family. In some applications, the rate at which input data is to be handled may be much lower than the rate at which the output data is to be handled.
 3. Therefore, it becomes necessary to interconnect the two different logic systems, such as TTL and ECL.
 4. A TTL cannot interface directly with an ECL; it requires a translator as shown in Fig. 5.36.1.

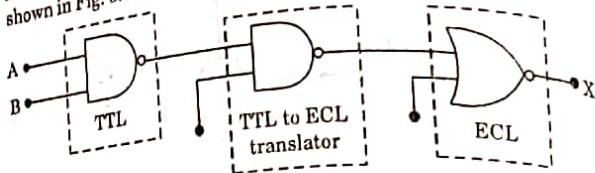


Fig. 5.36.1. TTL driving ECL.

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

ECL to TTL:

1. Sometimes, the input data is at a faster rate, but the output data is at a slower rate like in demultiplexers.
2. An ECL to TTL logic translator will be of use in such cases. It shows that the input logic levels of a translator are compatible with the output logic levels of ECL and the output logic levels of a translator are compatible with the input logic levels of a TTL.
3. Fig. 5.36.2 shows the ECL gate driving a TTL gate.

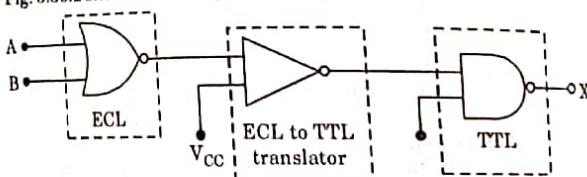


Fig. 5.36.2. ECL driving TTL.

Que 5.37. Implement the following functions using 3-input, 3 product terms and 2 output PLA:

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

$$F_2 = AC + BC$$

Answer

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

$$F_2 = AC + BC$$

	A	B	C	F_1	F_2
$A\bar{B}$	1	0	-	1	-
AC	1	-	1	1	-
BC	-	1	1	-	1

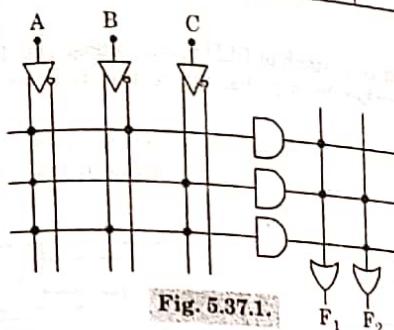


Fig. 5.37.1.

Que 5.38. A combinational circuit is defined by the function

$$F_1 = \Sigma m(1, 5, 7)$$

$$F_2 = \Sigma m(5, 6, 7)$$

Implement the circuit with a PLA.

Answer

1. Determine the simplified boolean expression by using 3 variable K-map. Consider the three variable A, B and C.
2. Next form the program table to implement combinational logic circuit a PLA. We have only three product terms.

Product term	Inputs			Outputs	
	A	B	C	F_1	F_2
(AC)	1	-	1	1	1
($\bar{B}C$)	-	0	1	1	-
(AB)	1	1	-	-	1

3. The programmed PLA is shown in Fig. 5.38.1.

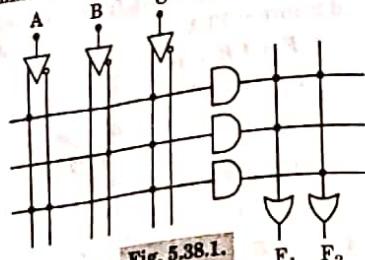


Fig. 5.38.1.

Que 5.38. What do you understand by complex-programmable logic device (CPLD)?

Answer

1. A CPLD contains a bunch of PLD blocks whose inputs and output are connected together by a global interconnection matrix as shown in Fig. 5.39.1.
2. A CPLD has two levels of programmability : each PLD block can be programmed, and then the interconnections between the PLDs can be programmed.

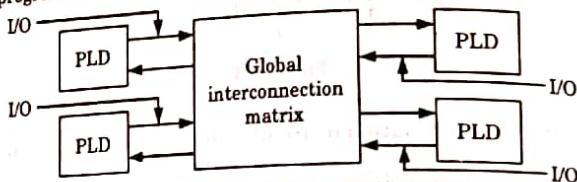


Fig. 5.39.1. Complex-programmable logic device (CPLD).

3. The expansion of PLD using multiple simple-programmable logic device (SPLD) chips has the disadvantages such as personal computer (PC) board area requirement increases with the number of chips.
4. Connecting wires will result in adverse capacitive effects. Power requirement increases with the number of chips.
5. A CPLD is collection of individual PLDs on a single chip and programmable interconnection structure.
6. By using programming methods, the resource available in various PLDs can be shared in different ways to design complex logic functions.

Que 5.40. Discuss the concept of field programmable gate array (FPGA). Describe the various structures of FPGA.

Answer

Field programmable gate array :

FPGA is high capacity PLD. The gate array of FPGA has the ability to be programmed for a function by the user instead of the manufacturer of device.

FPGA consists of three configurable (programmable) logic modules (LMs) : configurable logic blocks (CLBs), input and output blocks and switching matrix for interconnection.

The CLB consists of a combinational logic array, data multiplexer (MUX) and flip-flops. The combinational array function is performed by look-up table (LUT).

The FPGA consists of three main structures :

- Programmable logic structure,
- Programmable routing structure, and
- Programmable input / output (I / O).

Structures of FPGA :

Programmable logic structure :

The programmable logic structure of FPGA consists of a two-dimensional array of CLBs.

Each CLB typically contains one or two flip-flops to allow implementation of sequential logic.

Large designs are partitioned and mapped to a number of CLBs with each CLB configured (programmed) to perform a particular function.

These CLBs are then connected together to fully implement the target design.

Programmable routing structure :

To allow for flexible interconnection of CLB, FPGA has three programmable routing resources.

Vertical and horizontal routing channels which consist of different length wires that can be connected together if needed.

These channels run vertically and horizontally between columns and rows of CLBs as shown in the Fig. 5.40.1.

Connection boxes, which are set of programmable links, can connect input and output pins of the CLBs to wires of the vertical or the horizontal routing channels.

Switch boxes are located at the intersection of the vertical and horizontal channels.

These are a set of programmable links that can connect wire segments in the horizontal and vertical channels.

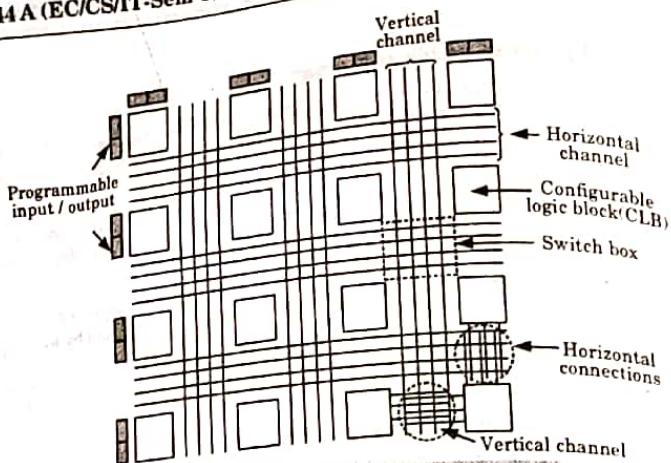


Fig. 5.40.1. Programmable structure of field programmable logic array (FPGA).

iii. Programmable input/output :

- These are mainly buffers that can be configured either as input buffers, output buffers or input/output as shown in Fig. 5.40.1.
- These allow the pins of the FPGA chip to function either as input pins, output pins or input/output pins.

iv. Configurable logic blocks :

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

v. Logic module :

- A logic module (LM) consists of an LUT, a D-type flip-flop and a MUX. Most of the FPGAs are based on 4-input LUT. Fig. 5.40.2 shows a block diagram of a LM with 4-input LUT.
- Output of the LUT becomes the output of the LM either directly or through D-type flip-flop. Thus, the output can be configured for combinational or registered (i.e., through flip-flop).

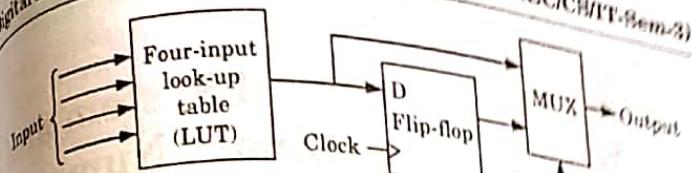


Fig. 5.40.2. Block diagram of logic module.

Que 5.41. What do you mean by look-up table (LUT) ?

Answer

- An LUT consists of a programmable memory and it can be used to generate logic function in SOP form.
- A memory can generate canonical product terms. Fig. 5.41.1 shows a block diagram of an LUT. It consists of a memory and a MUX.
- Let the memory contents be given in Fig. 5.41.1. Since, it is an 8-bit memory therefore an 8:1 MUX is required. If the 3-bit logical input is A_2, A_1 and A_0 , then $y = \Sigma m(1, 2, 5, 7)$.
- The look-up table in most of the commercially available FPGAs is 4-input circuit. Larger LUT allows for more complex logic to be performed per logic block, thus reducing the wiring delay between blocks.

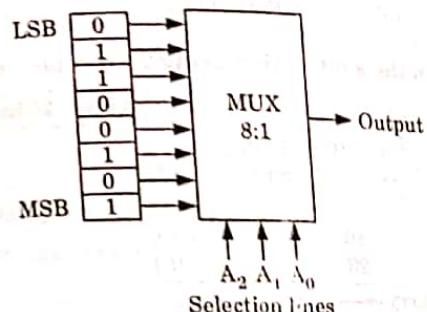


Fig. 5.41.1. Block diagram of LUT.





Digital System and Binary Numbers (2 Marks Questions)

1.1. Convert $(153.513)_{10}$ to an octal number?

AKTU 2016-17, Marks 02

Ans: 1. $(153)_{10}$ to octal :

$$\begin{array}{r} 8 \mid 153 \quad | 1 \\ 8 \quad 19 \quad 3 \\ \hline 8 \quad 2 \quad 2 \\ \hline 0 \end{array}$$

(0.513)₁₀ to octal :
 $0.513 \times 8 = 4.104$
 $0.104 \times 8 = 0.832$
 $0.832 \times 8 = 6.656$
 $0.656 \times 8 = 5.248$
 $0.248 \times 8 = 1.984$
 $0.984 \times 8 = 7.872$
 $(0.513)_{10} = (0.406517\dots)_8$

2. So, $(153)_{10} = (231)_8$ $(153.513)_{10} = (231.406517\dots)_8$

1.2. Perform the subtraction using 2's complement 46-23.

AKTU 2015-16, Marks 02

Ans: $(46)_{10} \rightarrow (00101110)_2$
 $(-23)_{10} \rightarrow$ 2's complement of 23

$$\begin{array}{r} = 11101001 \\ 46 = 00101110 \\ - 23 = +11101001 \end{array}$$

$$\begin{array}{r} \text{Neglect carry } \leftarrow 100010111 \\ 46 - 23 = (00010111)_2 = 23 \end{array}$$

1.3. What is signed binary number?

Ans: 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.

1.4. In how many ways the binary codes are classified?

- Ans:** 1. Weighted codes 2. Non-weighted codes
- 3. Reflective codes 4. Sequential codes
- 5. Alphanumeric codes
- 6. Error detecting and correcting codes.

1.5. Define cyclic codes.

Ans: When a bit pattern of two consecutive numbers differ by only one bit position, these codes are called cyclic codes.

1.6. Write the advantages of Gray code over the straight binary number sequence.

AKTU 2016-17, Marks 02

Ans: 1. The Gray code is used in applications in which the normal sequence of binary numbers generated by the hardware may produce an error or ambiguity during the transition from one number to the next. The Gray code eliminates this problem, since only one bit change its value during any transition between two numbers.

2. Gray code represents analog data by a continuous change in the angular position of a shaft. Gray code eliminates ambiguity between the angle of the shaft and the value encoded by the sensor.

1.7. Convert the following expression into canonical POS form

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

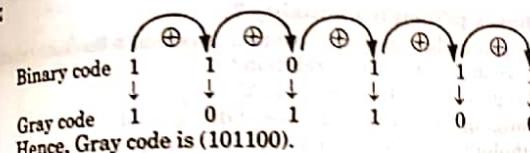
$$Y = (A + B)(B + C)(A + C)$$

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

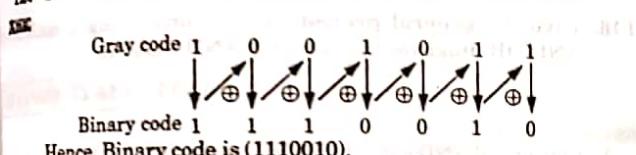
Canonical form,

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

1.8. Convert the binary number $(110111)_2$ into Gray code.



1.9. Convert gray code 1001011 to binary.



1.10. What are the universal gates and why we call them as universal?

Ans: NAND and NOR gates are universal gates, these are called universal because it is possible to implement any logic gate or expression using only NAND or only NOR gates.

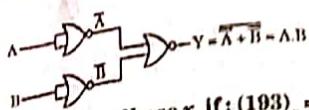
1.11. Implement two input AND gate using NAND gate.

Ans:



- 1.12. Implement two input AND gate using NOR gate.

Ans:



- 1.13. Determine the value of base x , if: $(193)_x = (623)_8$

Ans:

$$(193)_x = (623)_8$$

Converting octal into decimal: $6 \times 8^2 + 2 \times 8 + 3 = (403)_{10}$

$$(193)_x = 1 \times x^2 + 9 \times x^1 + 3 \times x^0 = (623)_{10}$$

$$x^2 + 9x + 3 = 403$$

$$x^2 + 9x - 400 = 0$$

$$(x+20)(x-20) = 0$$

$$x = 16 \text{ or } x = -25$$

Since, negative is not applicable, hence

$$x = 16$$

$$(193)_{10} = (623)_8$$

- 1.14. Find 9's and 10's complement of the following decimal number:

i.

$$24,681,234$$

$$\text{ii. } 63,325,600$$

9's complement

ii.

$$99999999 - 24681234 = 75318765$$

$$75318765 + 1 = 75318766$$

10's complement

$$99999999 - 63325600 = 36674399$$

$$36674399 + 1 = 36674400$$

9's complement

10's complement

- 1.15. What is principle of duality?

Ans:

The principle of duality theorem says that a boolean relation can derive another boolean relation by,

1. Changing each OR sign to an AND sign
2. Changing each AND sign to an OR sign and
3. Complementing any 0 or 1 appearing in the expression.

For example : Dual of relation $A + \bar{A} = 1$ is $A \cdot \bar{A} = 0$

- 1.16. Give the general procedure for converting a multilevel AND-OR diagram into an all NAND diagram.

AKTU 2016-17, Marks 02

Ans:

1. Convert all AND gates to NAND gates with AND-invert graphic symbols.
2. Convert all OR gates to NAND gate with invert-OR graphic symbol.
3. Check all bubbles in the diagram. For every bubble that is not compensated by another small circle along the same line, insert an inverter (a one-input NAND gate) or complement the input literal.

- 1.17. Realize an EX-OR gate using NAND gates only.

AKTU 2015-16, Marks 02

- Ans: EX-OR gate using NAND gates:

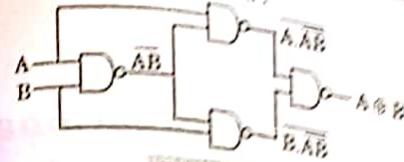


Fig. 1.18.1.

- 1.18. Convert the decimal number 32.75 in octal, binary, hexadecimal and Gray.

AKTU 2015-16, Marks 02

Ans: For octal :

8 32 0		
8 4 4		
8 0		
32 = (40) ₈		

$$\text{and } .75 = .75 \times 8 = 6.00$$

$$\text{Hence, } (32.75)_{10} = (40.6)_{8}$$

For binary :

2 32 0		
2 16 0		
2 8 0		
2 4 0		
2 2 0		
2 1 1		
0		

$$32 = (100000)_2$$

$$\text{and } .75 = .75 \times 2 = 1.50$$

$$0.50 \times 2 = 1$$

$$\text{Hence, } (32.75)_{10} = (100000.11)_2$$

For hexadecimal :

16 22 0		
16 2 2		
16 0		
22 = (16) ₁₆		

$$\text{and } .75 = .75 \times 16 = 12.00$$

$$\text{Hence, } (32.75)_{10} = (16.C)_{16}$$

For Gray :



$$\text{Hence, } (32.75)_{10} = (11000010)_{\text{Gray}}$$

- 1.19. Simplify the following boolean expression to a minimum number of literals : $(x'y' + z')' + z + xy + wz$

AKTU 2015-16, Marks 03

Ans:

Given, $(x'y' + z')' + z + xy + wz$

Using De-Morgan's theorem

$$(x'y' + z')' = (x + y)z$$

$$(x'y' + z')' + z + xy + wz = (x + y)z + z + xy + wz$$

$$= z[1 + (x + y)] + xy + wz = z + xy + wz$$

$$= z[1 + w] + xy = z + xy$$

☺☺☺



Combinational Logic (2 Marks Questions)

2.1. Define combinational circuits.

ANS: It consists of input variables, logic gates and output variables. Logic gates accept signal from input variable and generate output signals. This process transforms binary information from given input data to the required output data.

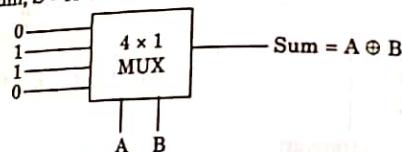
2.2. Define carry look ahead adder.

ANS: It is an additional technique which speeds up the addition process by eliminating the problem due to inter-state carry delay.

2.3. Design a half adder using multiplexer.

AKTU 2015-16, Marks 02

ANS: For sum, $S = A \oplus B = \bar{A}B + A\bar{B}$



For carry, $C = AB$

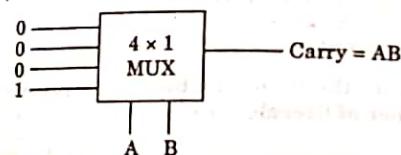


Fig. 2.3.1. Half adder using multiplexer.

2.4. Give the expression of full adder output.

ANS: Sum output, $S = ABC + A\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} = A \oplus B \oplus C$
Carry output, $C_0 = AB + AC + BC$

2.5. Draw the logic diagram of half subtractor.

AKTU 2016-17, Marks 02

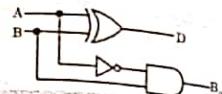


Fig. 2.5.1. Half subtractor.

2.6. Explain multiplexer circuit.

ANS: It is a combinational circuit that selects binary information from one of many lines and directs it to a single output line. The selection of particular input line is controlled by a set of selection lines. There are 2^n input lines and n select lines whose bit combination determines which input is selected.

**2.7. Implement the following function using 8×1 MUX,
 $Y(A, B, C) = \sum m(0, 1, 3, 5, 7)$.**

ANS

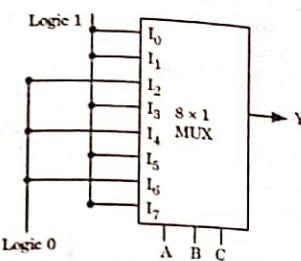


Fig. 2.7.1.

2.8. Give the circuit diagram and output equation for 4×1 MUX.

ANS

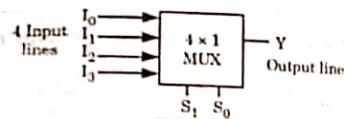


Fig. 2.8.1.

Output equation, $Y = I_0\bar{S}_1\bar{S}_0 + I_1\bar{S}_1S_0 + I_2S_1\bar{S}_0 + I_3S_1S_0$.

**2.9. Implement the following expression using $4 : 1$ MUX.
 $Y(A, B) = \bar{A}B + A\bar{B}$**

ANS

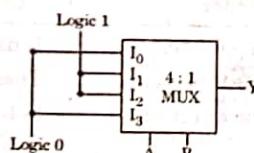


Fig. 2.9.1.

Ques. 2.10. What do you mean by priority encoder?

Ans. It is an encoder circuit that includes the priority function. The operation of priority encoder is such that, if two or more inputs are equal to 1 at the same time, the input having the highest priority will take precedence.

Ques. 2.11. Which circuit is used at input of communication system?

Ans. Encoder is used at input of communication system.

Ques. 2.12. Specify the purpose of valid bit indicator in priority encoder.

AKTU 2016-17, Marks 02

Ans. In priority encoder, valid bit indicator is set to 1, when one or more inputs are equal to 1. If all inputs are 0, there is no valid input and valid bit indicator is equal to 0. The other two outputs are not inspected when valid bit indicator equals to 0 and are specified as don't care conditions.

Table 2.12.1. Truth table of a priority encoder.

Inputs				Outputs		
D_0	D_1	D_2	D_3	x	y	v
0	0	0	0	x	x	0
1	0	0	0	0	0	1
x	1	0	0	0	1	1
x	x	1	0	1	0	1
x	x	x	1	1	1	1

Ques. 2.13. What are the circuits by which all the expressions are implemented?

Ans. Multiplexer and decoder, both are used for implementing all boolean expression.

Ques. 2.14. Write the difference between decoder and demultiplexer.

AKTU 2015-16, Marks 02

Ans.

S.No.	Decoder	Demultiplexer
1.	It has n input line.	It has 1 input line.
2.	Enable line is present.	There is no enable line.
3.	There is no select line.	Select line is present.

@@@



Sequential Logic and its Applications (2 Marks Questions)

Ques. 3.1. Define sequential circuits.

Ans. Sequential circuits consist of a combinational circuit to which storage elements are connected to form a feedback path.

Ques. 3.2. What do you mean by flip-flops?

Ans. Storage elements that are controlled by a clock transition are flip-flops. It is a binary storage device capable of storing one bit of information. Flip-flops are edge triggered devices.

Ques. 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.

Ques. 3.4. Give the function table of SR latch.

AKTU 2016-17, Marks 02

Ans.

S	R	Q_{z+1}
0	0	Q_z (Preset state)
0	1	0
1	0	1
1	1	x

Ques. 3.5. Draw Master-Slave flip-flop.

AKTU 2015-16, Marks 02

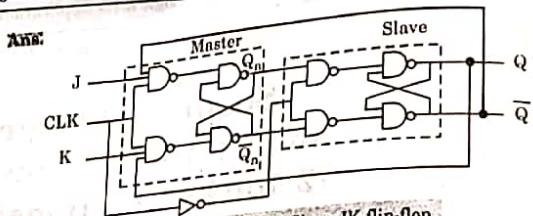


Fig. 3.5.1. Master-Slave JK flip-flop.

3.6. Express the characteristic equation for the JK flip-flop.
AKTU 2016-17, Marks 02

Ans: JK flip-flop : $Q_{n+1} = J\bar{Q}_n + \bar{K}Q_n$

3.7. What is race around condition ?

Ans: A race around condition is said to exist in an asynchronous sequential circuit when two or more binary state variable change value in response to a change in input variable. It is eliminated by using master-slave JK flip-flop.

3.8. 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.

3.9. What is the need of a register ?

Ans: A register is needed because of two main functions :
1. Temporary storage 2. Shifting capability

3.10. 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)

3.11. Give the application of shift register.

Ans: Shift register are used in many application as :
1. Time delay 2. Serial to parallel data converter
3. Parallel to serial data converter

3.12. What is counter ?

Ans: A counter is a sequential logic circuit capable of counting the number of clock pulses arriving at its clock input. This count sequence may be ascending, descending or non sequence. A specified sequence of states appears at the counter output.

3.13. Define modulus counter.

Ans: A number of states through which counter passes before returning to starting state or initial state is called modulus counter.

3.14. Give the classifications of counters.

- Ans:** There are two types of counters :
1. Ripple/asynchronous or non-synchronous/serial counter
2. Synchronous or parallel counter

3.15. Define state table.

Ans: The time sequence of inputs, outputs and flip-flop states can be enumerated in a state table, it is also called transition table. It is consist of four sections :

1. Present state
2. Input state
3. Next state
4. Output state

3.16. Define asynchronous counter or ripple counter.

Ans: A binary asynchronous/ripple counter consists of a series connection of complementing flip-flops, with the output of each flip-flop connected to the clock input of the next higher-order flip-flop. The flip-flop holding the least significant bit receives the incoming clock pulses.

3.17. Define synchronous counter.

Ans: The synchronous counter is one in which all the flip-flops are triggered simultaneously by the clock pulse, so they are also called as parallel counters. Synchronous counter may reduce the delay occurred in asynchronous counter.

3.18. The contents of a four bit register are initially 1011. The register is shifted six times to the right with serial input being 101111. What are the contents of the register after each shift ?

AKTU 2016-17, Marks 02

Ans: Given : Serial input = 101111, initial content = 1011.

	CLK	$D_{in} = D_3$	$Q_3 = D_2$	$Q_2 = D_1$	$Q_1 = D_0$	Q_0
initially	↓		1	0	1	1
1 st	↓	1	1	1	0	1
2 nd	↓	1	1	1	1	0
3 rd	↓	1	1	1	1	1
4 th	↓	1	1	1	1	1
5 th	↓	0	0	1	1	1
6 th	↓	1	1	0	1	1

3.19. What are the required number of flip-flops in a MOD-18 asynchronous counter, MOD-16 synchronous counter,

MOD-16 Johnson counter ?

AKTU 2015-16, Marks 02

- Ans:** MOD-16 asynchronous counter : 4 flip-flops required.
MOD-16 synchronous counter : 4 flip-flops required.
MOD-16 Johnson counter : 4 flip-flops required.





Synchronous and Asynchronous Sequential Circuits (2 Marks Questions)

4.1. Differentiate between synchronous and asynchronous sequential circuits.

AKTU 2015-16, Marks 02

Ans:

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.

4.2. Compare Mealy and Moore model of finite state machine.

AKTU 2016-17, Marks 02

Ans:

S.No.	Mealy model	Moore model
1.	Its output is a function of present input as well as present state.	Its output is the function of present state only.
2.	It requires less number of states for implementing same function.	It requires more number of states for implementing same function.

4.3. What are the types of asynchronous sequential circuits ?

Ans: There are two types of asynchronous circuits :

1. Fundamental mode circuits and
2. Pulse mode circuits.

4.4. What do you mean by fundamental mode operation ?

AKTU 2014-15, Marks 3.5

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. What do you mean by pulses mode operation ?

Ans: The input variables are pulses instead of levels. The pulse width must not be so long that it is still present after the new state is reached. Pulses should not occur simultaneously on two or more input lines. Input variables are used only in the uncomplemented or the complemented forms, but not both.

4.6. 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.

4.7. Define the term hazards.

Ans: Consider a logic circuit, which is expected to give a logic 1 output, momentarily logic 0 because of finite propagation delays of various gates, this unwanted switching transient is called hazard.

4.8. Give the classification of hazards.

Ans:

1. Static hazard : (i) Static-1 hazard (ii) Static-0 hazard
2. Dynamic hazard
3. Essential hazard.

4.9. What are the conditions of occurrence for static hazard ?

Ans: Static -1 hazard occurs in SOP circuits, whereas static-0 hazard occur in POS circuits. In static-1 hazard, the regular output of circuit is at logic 1. But during transient, the output change from 1 to 0 and then 0 to 1.

4.10. 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.

4.11. 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.

4.12. What is the method to eliminate race around conditions ?

Ans: Use of Master-Slave JK flip-flop can eliminate race around condition in digital circuits.





Memory and Programmable Logic Devices (2 Marks Questions)

- 5.1. What do you mean by bipolar and unipolar logic families?**
- Ans:** Bipolar logic family : In this logic family, the current flows because of both electrons and holes being charge carriers.
 Unipolar logic family : It include *p*-channel metal oxide semiconductor field-effect transistor (PMOS), *n*-channel metal oxide field effect transistor (NMOS) and CMOS. The current flow due to any one types of carrier.

- 5.2. Draw the circuit diagram of DTL NAND gate.**

Ans

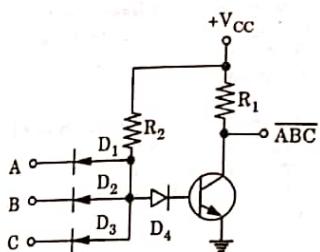


Fig. 5.2.1. DTL NAND gate.

- 5.3. What are the drawback of DCTL logic family?**
- Ans:** DCTL suffers from two serious drawbacks :

1. Current hogging
2. Low noise margin.

- 5.4. What are the uses of ECL family?**

Ans: ECL family is used in very high frequency application where its speed is superior. It is used in superfast computers and high-speed special purpose applications.

- 5.5. What are the difference between CMOS gate and TTL gate?**

- Ans:**
1. CMOS gate can transmit signal in both direction but TTL and ECL gates are essentially unidirectional.
 2. CMOS consists of *n*-MOS and *p*-MOS but TTL and ECL uses transistors.

- 5.6. What do you mean by fan-out and fan-in?**

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.

$$\text{Fan-out} = \text{Minimum of } \left\{ \frac{I_{OH}}{I_m}, \frac{I_{OL}}{I_u} \right\}$$

Fan-in : The fan-in of a logic gate refers to the number of inputs that the gate is designed to handle.

- 5.7. 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.

- 5.8. 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.

- 5.9. Define memory read and write operation.**

Ans: The process of storing new information into memory is referred to as a memory write operation and the process of fetching the stored information from the memory is referred to as a memory read operation.

- 5.10. 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.

- 5.11. 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.

- 5.12.** How many address lines and data I/O lines are required for a $10K \times 12$ memory?

Ans: $10K \times 12 = 2^4 \times 2^{10} \times 12$
Address lines = 14
Data lines = 12

AKTU 2015-16, Marks 02

- 5.13.** 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.
3.	RAM can be classified into two main categories: a. Dynamic RAM b. Static RAM	ROM can be classified as: a. Mask ROM. b. Programmable ROM.

- 5.14.** What do you mean by CPLD?

Ans: Complex-programmable logic device (CPLD) contains a bunch of PLD blocks whose input and output are connected together by a global interconnection matrix.

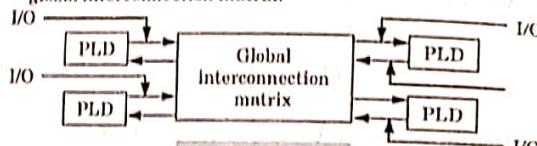


Fig. 5.14.1. CPLD.

- 5.15.** What do you mean by FPGA?

Ans: FPGA (Field Programmable Gate Array) is high capacity PLD. The gate array of FPGA has the ability to be programmed for a function by the user instead of the manufacturer of device.

- 5.16.** Write the steps that must be taken for the purpose of transferring a new word to be stored into memory.

AKTU 2016-17, Marks 02

Ans: The steps that must be taken for the purpose of transferring a new word to be stored into memory are as follows :

1. Apply the binary address of the desired word to the address lines.
2. Apply the data bits that must be stored in memory to the data input lines.
3. Activate the write input.



B.Tech.

(SEM. III) ODD SEMESTER THEORY EXAMINATION, 2014-15
SWITCHING THEORY AND LOGIC DESIGN

Time : 2 Hours

Total Marks : 50

Notes : 1. Attempt all questions.
2. All questions carry equal marks.

1. Attempt any four parts of the following :

a. Convert the following numbers into desired base : $(A6BF5)_{16} = (?)_2 = (?)_{gray}$

b. $(17 - 135)$ using 2's complement

Ans: Refer Q. 1.5, Page 1-5A, Unit-1.

- c. Simplify the following boolean expression to a minimum number of literals.

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

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

Ans: Refer Q. 1.22, Page 1-21A, Unit-1.

- d. Simplify the following expression into product of sum (POS) form

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

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

Ans: Refer Q. 1.21, Page 1-19A, Unit-1.

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

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

Ans: Refer Q. 1.32, Page 1-29A, Unit-1.

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

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

Ans: Refer Q. 1.27, Page 1-23A, Unit-1.

- g. For the Hamming code 1001101001 received at the receiver end, correct this code for error if any ?

Ans: Refer Q. 1.15, Page 1-13A, Unit-1.

2. Attempt any two parts of the following : (6 x 2 = 12)
- Design a BCD to 7 segment decoder. Assume positive logic, minimize the function.
- Ans:** Refer Q. 2.29, Page 2-29A, Unit-2.
- Design the following boolean function using 4×1 multiplexer.

$$F(A, B, C, D) = \Sigma m(0, 1, 3, 4, 8, 9, 15)$$
- Ans:** Refer Q. 2.23, Page 2-25A, Unit-2.
- Design and explain the logic and circuit of 4-bit magnitude comparator.
- Ans:** Refer Q. 2.17, Page 2-20A, Unit-2.
- Attempt any two parts of the following : (6 x 2 = 12)
- Distinguish between synchronous and asynchronous digital sequential circuit. Design modulo-5 counter.
- Ans:** Refer Q. 4.37, Page 4-49A, Unit-4.
- Explain race around condition and its remedy in brief. Realise T flip-flop to SR flip-flop.
- Ans:** Refer Q. 3.9, Page 3-10A, Unit-3.
- Write down the classification of semiconductor memories. Draw and explain the programmable logic array (PLA).
- Ans:** Refer Q. 5.28, Page 5-28A, Unit-5.
- Attempt any two parts of the following : (6 x 2 = 12)
- Explain hazard and its types. Define critical race and non-critical race. Also explain the elimination of hazards in asynchronous circuits.
- Ans:** Refer Q. 4.33, Page 4-46A, Unit-4.
- With the help of diagram, explain the operation of universal shift register.
- Ans:** Refer Q. 3.18, Page 3-23A, Unit-3.
- An asynchronous sequential circuit described by the following excitation and output functions.

$$Y = X_1 X_2 + (X_1 + X_2)Y \text{ and } Z = Y.$$

where X_1 and X_2 = input variables
 Y = excitation function
 Z = output function
- Draw the logic diagram of the circuit.
 - Derive transition table.
 - Output map and obtain a flow table.
- Ans:** Refer Q. 4.19, Page 4-26A, Unit-4.

(SEM. III) ODD SEMESTER THEORY EXAMINATION, 2015-16
SWITCHING THEORY AND LOGIC DESIGN

Time: 3 Hours

Total Marks : 100

- SECTION - A
1. Attempt all parts. All parts carry equal marks. Write answer of each part in short : (2 x 10 = 20)
- Write the difference between decoder and demultiplexer.
2. Refer Q. 2.14, Page SQ-7A, 2 Marks Questions, Unit-2.
- Perform the subtraction using 2's complement 46 - 23.
3. Refer Q. 1.2, Page SQ-1A, 2 Marks Questions, Unit-1.
- Realize an EX-OR gate using NAND gates only.
4. Refer Q. 1.17, Page SQ-3A, 2 Marks Questions, Unit-1.
- How many address lines and data I/O lines are required for a $16K \times 12$ memory ?
5. Refer Q. 5.12, Page SQ-15A, 2 Marks Questions, Unit-5.
- Simplify the following boolean expression to a minimum number of literals : $(x'y' + z')' + z + xy + wz$
6. Refer Q. 1.19, Page SQ-4A, 2 Marks Questions, Unit-1.
- Differentiate between asynchronous and synchronous sequential circuits.
7. Refer Q. 4.1, Page SQ-11A, 2 Marks Questions, Unit-4.
- What are the required number of flip-flops in a MOD-16 asynchronous counter, MOD-16 synchronous counter, MOD-16 Johnson counter ?
8. Refer Q. 3.19, Page SQ-10A, 2 Marks Questions, Unit-3.
- Design a half adder using multiplexer.
9. Refer Q. 2.3, Page SQ-5A, 2 Marks Questions, Unit-2.
- Convert the decimal number 32.57 in octal, binary hexadecimal and Gray.
10. Refer Q. 1.18, Page SQ-4A, 2 Marks Questions, Unit-1.
- Draw Master-Slave flip-flop.
11. Refer Q. 3.5, Page SQ-8A, 2 Marks Questions, Unit-3.
- SECTION - B
- Attempt any five questions from this section : (10 x 5 = 50)
- Design a 4-bit magnitude comparator using one bit comparator modules.
12. Refer Q. 2.17, Page 2-20A, Unit-2.
- Prepare Hamming code for the message "01001001010" assuming even parity. Also explain error detection and correction capabilities at the receiver by assuming an error in any one of the received bits.

- Ques:** Refer Q. 1.14, Page 1-11A, Unit-1.
4. Using a decoder and external gates, design the combinational circuit defined by the following three boolean functions :
 $F_1 = xyz' + zx$, $F_2 = xy'z' + z'y$, $F_3 = x'y'z' + xy$
- Ans:** Refer Q. 2.26, Page 2-27A, Unit-2.
5. Design a 3-bit binary to Gray code converter using PLA.
- Ques:** Refer Q. 5.34, Page 5-37A, Unit-5.
6. Draw and explain 4-bit universal shift register.
- Ans:** Refer Q. 3.18, Page 3-23A, Unit-3.
7. Describe the hazards in digital circuits. How are these removed ?
- Ques:** Refer Q. 4.33, Page 4-46A, Unit-4.
8. What do you understand by state reduction ? Reduce the following state diagram.
- Ques:** Refer Q. 4.10, Page 4-15A, Unit-4.
9. Derive the state table and state diagram for the sequential circuit shown in Fig. 1.

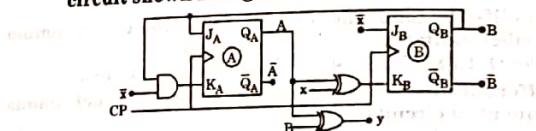


Fig. 1.

Ques: Refer Q. 4.5, Page 4-8A, Unit-4.**SECTION - C**Attempt any two questions from this section. $(15 \times 2 = 30)$

10. Minimize the following using Quine-McCluskey method :
 $F(W, X, Y, Z) = \Sigma(0, 3, 5, 6, 7, 10, 12, 13) + \Sigma d(2, 9, 15)$

Ans: Refer Q. 1.31, Page 1-28A, Unit-1.

11. a. Design a 3-bit asynchronous up-down counter using T flip-flop.

Ans: Refer Q. 3.22, Page 3-26A, Unit-3.

- b. Design a full adder using two half adders.

Ans: Refer Q. 2.6, Page 2-7A, Unit-2.

12. Design the clocked sequential circuit for the following state diagram using JK flip-flops.

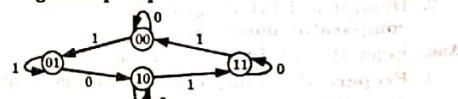


Fig. 2.

Ans: Refer Q. 4.4, Page 4-6A, Unit-4.**B.Tech.**
(SEM. III) ODD SEMESTER THEORY EXAMINATION, 2016-17
SWITCHING THEORY AND LOGIC DESIGN

Time : 3 Hours

Total Marks : 100

SECTION - A

1. Attempt all parts. All parts carry equal marks. Write answer of each part in short :
- a. Convert $(153.513)_{10}$ to an octal number. $(2 \times 10 = 20)$
- Ans:** Refer Q. 1.1, Page SQ-1A, 2 Marks Questions, Unit-1.
- b. Write the advantages of Gray code over the straight binary number sequence.
- Ans:** Refer Q. 1.6, Page SQ-2A, 2 Marks Questions, Unit-1.
- c. Give the general procedure for converting a multilevel AND-OR diagram into an all NAND diagram.
- Ans:** Refer Q. 1.16, Page SQ-3A, 2 Marks Questions, Unit-1.
- d. Draw the logic diagram of half subtractor.
- Ans:** Refer Q. 2.5, Page SQ-5A, 2 Marks Questions, Unit-2.
- e. Specify the purpose of valid bit indicator in priority encoder.
- Ans:** Refer Q. 2.12, Page SQ-7A, 2 Marks Questions, Unit-2.
- f. Give the function table of SR latch.
- Ans:** Refer Q. 3.4, Page SQ-8A, 2 Marks Questions, Unit-3.
- g. Express the characteristic equation for the JK flip-flop.
- Ans:** Refer Q. 3.6, Page SQ-9A, 2 Marks Questions, Unit-3.
- h. Compare Mealy and Moore model of finite state machine.
- Ans:** Refer Q. 4.2, Page SQ-11A, 2 Marks Questions, Unit-4.
- i. The contents of a four bit register are initially 1011. The register is shifted six times to the right with serial input being 101111. What are the contents of the register after each shift ?
- Ans:** Refer Q. 3.18, Page SQ-10A, 2 Marks Questions, Unit-3.
- j. Write the steps that must be taken for the purpose of transferring a new word to be stored into memory.
- Ans:** Refer Q. 5.16, Page SQ-15A, 2 Marks Questions, Unit-5.

SECTION - B2. Attempt any five questions from this section : $(10 \times 5 = 50)$

- a Simplify the boolean function.

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

which has the don't care conditions

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

ANS: Refer Q. 1.20, Page 1-18A, Unit-1.

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

ANS: Refer Q. 1.26, Page 1-22A, Unit-1.

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

ANS: Refer Q. 2.8, Page 2-10A, Unit-2.

- d. Draw the logic diagram of a two to four line decoder using NOR gates only.

ANS: Refer Q. 2.27, Page 2-28A, Unit-2.

- e. Construct a JK flip-flop, using a D flip-flop, a two to four one line multiplexer and an inverter.

ANS: Refer Q. 3.11, Page 3-18A, Unit-3.

- f. Design a hazard free circuit of the following boolean function $F(x_1, x_2, x_3) = \Sigma(1, 5, 6, 7)$

ANS: Refer Q. 4.32, Page 4-48A, Unit-4.

- g. Describe the operation of four bit synchronous binary counter with neat sketch.

ANS: Refer Q. 3.28, Page 3-35A, Unit-3.

- h. Draw the basic configuration of three PLDs.

ANS: Refer Q. 5.27, Page 5-28A, Unit-5.

SECTION - C

Note: Attempt any two questions from this section. $(15 \times 2 = 30)$

3. Minimize the following switching function using Quine-McCluskey method.

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

ANS: Refer Q. 1.30, Page 1-27A, Unit-1.

4. Design a combinational circuit that converts a BCD code to excess-3 code.

ANS: Refer Q. 2.4, Page 2-4A, Unit-2.

5. Implement the following four boolean functions with a PAL.

$$W(A, B, C, D) = \Sigma(2, 12, 13)$$

$$X(A, B, C, D) = \Sigma(7, 8, 9, 10, 11, 12, 13, 14, 15)$$

$$Y(A, B, C, D) = \Sigma(0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15)$$

$$Z(A, B, C, D) = \Sigma(1, 2, 8, 12, 13)$$

ANS: Refer Q. 5.32, Page 5-33A, Unit-5.



B. Tech.

(SEM. III) ODD SEMESTER THEORY EXAMINATION, 2017-18 DIGITAL LOGIC DESIGN

Time : 3 Hours

Max. Marks : 70

Note: Attempt all sections. If require any missing data; then choose suitably.

SECTION-A

1. Attempt all questions in brief:

2. Write four advantages of digital systems over analog system. $(2 \times 7 = 14)$

- ANS:** 1. Digital communication system is more robust than analog system because it can resist the corruption of signal much better in presence of channel noise.
2. Implementation of digital hardware in digital communication system is flexible and permits the use of microprocessor, digital switching etc.
3. In digital system, it is possible to multiplex several digital signals that offer more efficient use of available bandwidth.
4. In digital system, it is easy to store large quantities of information.

3. Write the excitation table and characteristic equation of JK flip-flop.

ANS: Excitation table : Refer Q. 3.7, Page 3-9A, Unit-3.

Characteristic equation : Refer Q. 3.6, Page 3Q-9A, 2 Marks Question, Unit-3.

4. Write the difference between combinational and sequential circuits.

ANS:

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.

d. What is $(33)_6 + (45)_6$
Ans: $(33)_6 = 3 \times 6^1 + 3 \times 6^0 = (21)_{10}$
 $(45)_6 = 4 \times 6^1 + 5 \times 6^0 = (29)_{10}$
 $(33)_6 + (45)_6 = (50)_{10}$

Now,

6	50
6	8
6	1
0	1

So, $(33)_6 + (45)_6 = (50)_{10} = (122)_6$

e. Implement the expression $Y = ABC' + BD + E$ using NAND gate only.

Ans: There are five variables, A, B, C, D and E
 $Y = \overline{Y} = \overline{ABC'} + \overline{BD} + \overline{E} = \overline{ABC'} \cdot \overline{BD} \cdot \overline{E}$

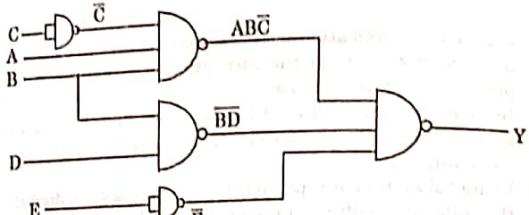


Fig. 1.

f. Convert the following :

i. $(562.13)_7 = (?)_{10}$
ii. $(467.342)_8 = (?)_{10}$

Ans: i. $(562.13)_7 = 5 \times 7^2 + 6 \times 7^1 + 2 \times 7^0 + 1 \times 7^{-1} + 3 \times 7^{-2}$
 $= (289.20)_{10}$

ii. $(467.342)_8 = 4 \times 8^2 + 6 \times 8^1 + 7 \times 8^0 + 3 \times 8^{-1} + 4 \times 8^{-2} + 2 \times 8^{-3}$
 $= (311.441)_{10}$

g. What is race around condition ?

Ans: Refer Q. 3.7, Page SQ-9A, 2 Marks Questions, Unit-3.

SECTION-B

2. Attempt any three of the following : (7 x 3 = 21)

a. Simplify the following Boolean function using K-map

$$Y = \sum m(0, 1, 3, 5, 6, 7, 9, 11, 16, 18, 19, 20, 21, 22, 24, 25)$$

$$\text{Ans: } Y = \sum m(0, 1, 3, 5, 6, 7, 9, 11, 16, 18, 19, 20, 21, 22, 24, 26)$$

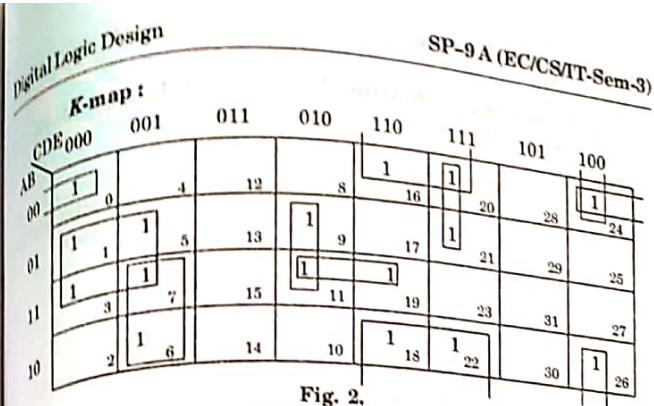


Fig. 2.

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

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

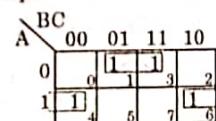
b. Write the steps for combinational circuit designing and design a circuit of three input which gives an high output whenever the sum of LSB and MSB bit is 1.

Ans: Step for designing combinational logic circuit : Refer Q. 2.3, Page 2-4A, Unit-2.

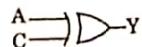
Let the three input be A, B and C, so according to the question, the truth table will be as follows :

LSB		MSB		Output
A	B	C		
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



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



- e. Implement the function $F = \sum m(0, 1, 3, 4, 7, 8, 9, 11, 14, 15)$ using 8:1 MUX.

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

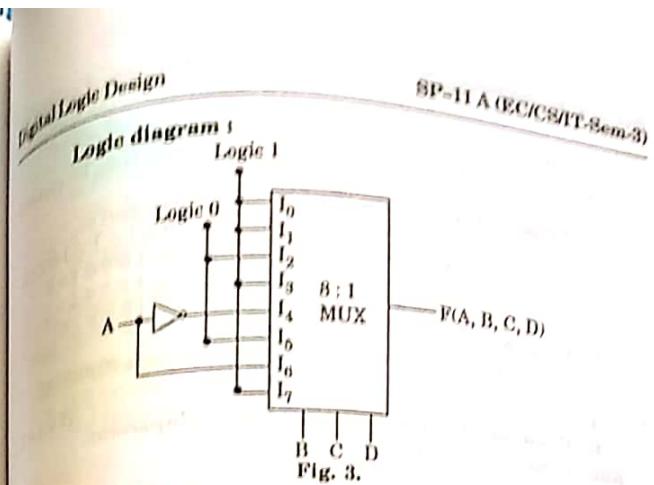
Decimal	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

2. A is assumed to be MUX input and B, C, D are used as selection lines.

- 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 :

\bar{A}	I ₀	I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇
0	①	②	2	③	④	5	6	⑦
1	⑧	⑨	10	⑩	12	13	⑪	⑫
	1	1	0	1	\bar{A}	0	A	1



- d. Draw and explain the PISO, PIPO register.
Ans: Refer Q. 3.17, Page 3-20A, Unit-3.

- e. Draw and explain 4-bit by 3-bit multiplier.
Ans: Refer Q. 2.15, Page 2-18A, Unit-2.

SECTION-C

3. Attempt any one part of the following : (7 x 1 = 7)
a. Design a universal shift register that performs HOLD, SHIFT RIGHT, SHIFT LEFT, and LOAD.
Ans: Refer Q. 3.18, Page 3-23A, Unit-3.
- b. Generate the Hamming code for the word 11011. Assume that a single error occurs while storing the generated Hamming code. Explain how this single error is detected.

1. Given data : 11011, it is a 5-bit data word
Required parity bit, k is given by

$$n \leq 2^k - k - 1$$

$$5 \leq 2^4 - 4 - 1$$

$$k = 4$$

Hence, 4 parity bits are required for 5-bit message.

2. Bit position

1	2	3	4	5	6	7	8	9
P ₁	P ₂	1	P ₄	1	0	1	P ₈	1

3. Parity bit is calculated as follow :

$$P_1 = 1 \oplus 1 \oplus 1 \oplus 1 = 0 \text{ (XOR of bits 3, 5, 7, 9)}$$

$$P_2 = 1 \oplus 0 \oplus 1 = 0 \text{ (XOR of bits 3, 6, 7)}$$

$$P_3 = 1 \oplus 0 \oplus 1 = 0 \text{ (XOR of bits 5, 6, 7)}$$

$$P_4 = 1 \text{ (XOR of bits 9)}$$

3. So, Hamming code for the data word 11011 will be,

Code : 00101011

4. Assume a single error generated while storing code, let received code be 001110111

5. Error correction : The check bits are determined as

$$\begin{aligned}C_1 &= \text{XOR of bit } 1, 3, 5, 7, 9 = 0 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 0 \\C_2 &= \text{XOR of bits } 2, 3, 6, 7 = 0 \oplus 1 \oplus 0 \oplus 1 = 0 \\C_3 &= \text{XOR of bits } 4, 5, 6, 7 = 1 \oplus 1 \oplus 0 \oplus 1 = 1 \\C_4 &= \text{XOR of bits } 2, 3, 6, 7 = 1 \oplus 1 \oplus 0 = 0\end{aligned}$$

6. Error bit location : $C_3 C_4 C_2 C_1 = 0100$

7. Thus, there is error in 4th bit. So, correct code is 001010111.

4. Attempt any one part of the following : (7 x 1 = 7)

a. Draw and explain 4-bit magnitude comparator.

Refer Q. 2.17, Page 2-20A, Unit-2.

b. Draw a decimal adder to add BCD numbers.

Refer Q. 2.12, Page 2-14A, Unit-2.

5. Attempt any one part of the following : (7 x 1 = 7)

a. Draw and explain the operation of a RTL NOR gate.

ANS

1. The basic circuit of the RTL digital logic family is the NOR gate shown in Fig. 4. Each input is associated with one resistor and one transistor.

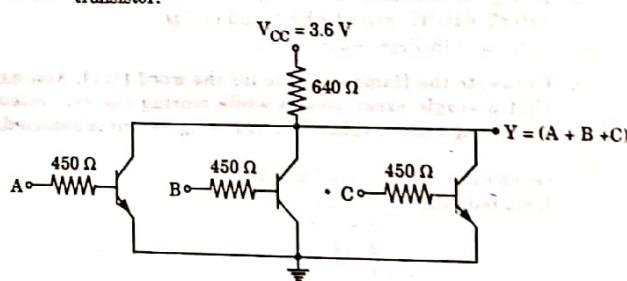


Fig. 4. Basic RTL NOR gate.

2. The collectors of the transistors are tied together at the output. The voltage levels for the circuit are 0.2 V for the low level and from 1 to 3.6 V for the high level.
3. If any input of the RTL gate is high, the corresponding transistor is driven into saturation and the output goes low, regardless of the states of the other transistors.
4. If all inputs are low at 0.2 V, all transistors are cut-off because $V_{BE} < 0.6$ V and the output of the circuit goes high, approaching the value of the supply voltage V_{CC} .

5. 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 Ω resistor.
6. A simple calculation shows that if I_{DS} drops to 20, the output voltage drops to about 1 V when the fan-out is 5.

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

Refer Q. 5.6, Page 5-7A, Unit-5.

b. Attempt any one part of the following :

c. An asynchronous sequential logic circuit is described by the following excitation and output function

$$\begin{aligned}y &= X_1 X_2 + (X_1 + X_2)Y \\Z &= y\end{aligned}$$

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

Refer Q. 4.19, Page 4-26A, Unit-4.

d. Design a 3-bit up/down ripple counter.

1. The 3-bit up/down ripple counter, which can count in upward direction of sequence from 000, 001, 010, 011, 100, 101, 110, 111 and downward direction of sequence from 111, 110, 101, 100, 011, 010, 001, 000.

2. 3-bit counter consists of 3 flip-flops. In ripple counter, a flip-flop output transition serves as a source for triggering other flip-flops.

3. The control signal M is used to select the direction of count sequence. Fig. 5 shows the 3-bit ripple.

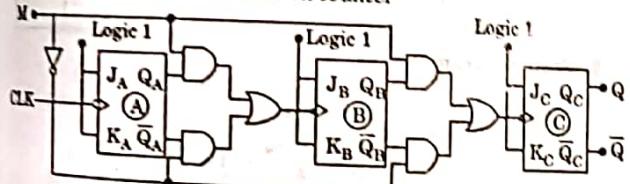
 $M = 1$; counter acts as up-counter $M = 0$; counter acts as down counter

Fig. 5. 3-bit ripple up-down.

7. Attempt any one part of the following : (7 x 1 = 7)

a. Write short notes on RAM and PLA

RAM: Refer Q. 5.20, Page 5-21A, Unit-5.

PLA: Refer Q. 5.28, Page 5-28A, Unit-5.

- b. Derive the state table and state diagram of the synchronous sequential circuit shown below (X is an input to the circuit). Explain the circuit function.

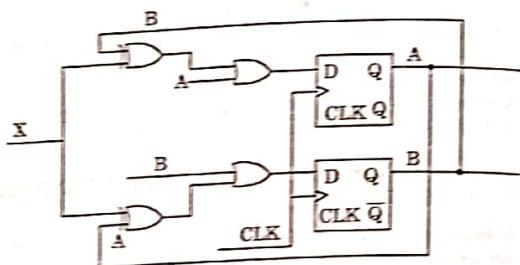


Fig. 6.

- From the circuit shown in Fig. 6, the output equation can be obtained as,

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

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

- The state table for the circuit shown in Fig. 6.

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

- State diagram:

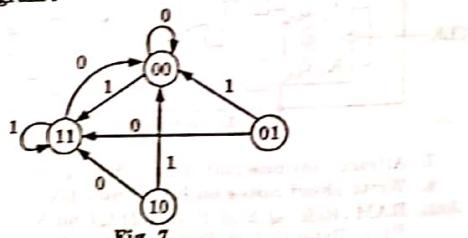


Fig. 7.

Minimize the given Boolean function using K-map.

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

B. Tech.

(SEM. III) ODD SEMESTER THEORY
EXAMINATION, 2018-19
DIGITAL LOGIC DESIGN

Time : 3 Hours

Max. Marks : 70

Note: Be precise in your answer. In case of numerical problem assume data wherever not provided.

SECTION-A

1. Attempt all of the following questions : $(2 \times 7 = 14)$
- a. What is modulus of a counter ?

Refer Q. 3.13, Page SQ-9A, 2 Marks Question, Unit-3.

- b. How many flip-flops are required to design Mod-5 ring counter and Mod-5 Johnson counter ?

Mod-5 Ring counter - 5 Flip-flop.

Mod-5 Johnson counter - 3 Flip-flop.

- c. Determine the value of base x , if $(193)_x = (623)_2$.

Refer Q. 1.13, Page SQ-3A, 2 Marks Question, Unit-1.

- d. Write the advantages of Gray code over the straight binary number sequence.

Refer Q. 1.6, Page SQ-2A, 2 Marks Question, Unit-1.

- e. What do you mean by fan-out and fan-in ?

Refer Q. 5.6, Page SQ-14A, 2 Marks Question, Unit-5.

- f. Define cyclic codes.

Refer Q. 1.5, Page SQ-2A, 2 Marks Question, Unit-1.

- g. What is race around condition ?

Refer Q. 3.7, Page SQ-9A, 2 Marks Question, Unit-3.

SECTION-B

2. Attempt any three of the following questions : $(7 \times 3 = 21)$

- a. Minimize the given boolean function using K-map.

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

Refer Q. 1.19, Page 1-18A, Unit-1.

- b. Minimize the following using Quine-McCluskey method :

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

Ans:

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

Table 1.

No. of 1's	Minterms	Binary		Minterms (2 cell)		Binary		Min-terms (4 cell)		Binary						
		A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
0	m_0	0	0	0	0	0	0	1	✓	0	0	0	0	-	0, 1, 8*, 9	0 - 0 0 -
1	m_1	0	0	0	0	1	0	0	✓	0	-	0	0	0	0	0, 2*
	dm_2	0	1	0	0	0	1	✓	0	-	0	0	1	-	1, 9	
2	m_9	0	1	0	0	1	8*, 9	✓	0	1	0	0	-	-	-	-
	m_{24}	1	1	0	0	0	8*, 24	-	1	0	0	0	-	-	-	-
3	dm_{11}	0	1	0	1	1	9, 11*	0	1	0	-	1	-	-	-	-
4	m_{15}	0	1	1	1	1	11*, 15	0	1	-	1	1	-	-	-	-
	m_{25}	1	1	1	0	1	15, 31*	-	1	1	1	1	-	-	-	-
	m_{30}	1	1	1	1	0	29, 31*	1	1	1	-	1	-	-	-	-
5	dm_{31}	1	1	1	1	1	30, 31*	1	1	1	1	-	-	-	-	-

2. List of prime implicants:

Table 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 3.

Prime implicant	Minterm				
	✓	✓	✓	✓	✓
0	1	9	15	24	29
3	0	1	9	15	24
8*, 24	✓				
9, 11*		✓			
11*, 15			✓		
15, 31*				✓	
29, 31*					✓
30, 31*					✓
0, 1, 8*, 9	✓	✓	✓	✓	✓

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

Write a short note on priority encoder.

See Page 2-32A, Unit-2.

Implement the following Boolean function.

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

i 4:1 MUX ii 2:1 MUX

iii

i 4:1 MUX :

Implementation table :

	AB	$\bar{A}\bar{B}$	$A\bar{B}$	$\bar{A}B$
(0)	$\bar{C}\bar{D}$	①	④	③
(1)	$\bar{C}D$	①	5	⑨
(2)	$C\bar{D}$	2	6	10
(3)	CD	②	7	11

$$\text{First column } (I_1) = \bar{C}\bar{D} + \bar{C}D + CD$$

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

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

$$\text{Second column } (I_2) =$$

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

[D → Ex-or]

$$\begin{aligned}
 &= \bar{C}\bar{D} + \bar{C}D + CD \\
 &= \bar{C}(\bar{D} + D) + CD \\
 &= \bar{C} + CD = \bar{C} + D
 \end{aligned}$$

Fourth column (I_3)

$$\begin{aligned}
 &= C\bar{D} + CD \\
 &= C(\bar{D} + D) = C
 \end{aligned}$$

Logic diagram :

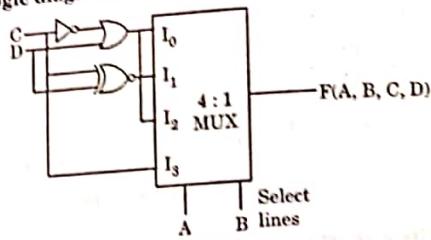


Fig. 1.

ii. Implementation using 2:1 MUX :

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

Implementation table :

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

First column (I_0)

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

Second column (I_1)

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

Logic diagram :

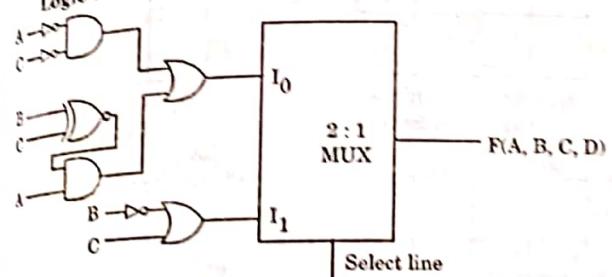


Fig. 2.

e Design Binary code to Gray code converter.

m Truth table :

Binary code				Gray code			
D	C	B	A	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	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

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

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

Expression for G_2

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

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

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

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

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

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

$$G_3 = D$$

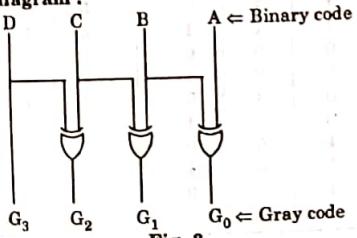
Logic diagram :

Fig. 3.

Expression for G_1

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

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

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

$$G_3 = D$$

SECTION-C

3. Attempt any one part of the following questions : (7 x 1 = 7)
 a.i. Draw a BCD adder circuit and explain its working.
 Ans: Refer Q. 2.12, Page 2-14A, Unit-2.

ii. Convert the SR flip-flop to JK flip-flop.
 Ans: Refer Q. 3.12, Page 3-14A, Unit-3.

b. What do you mean by shift register ? What is the need of shift register ? Draw and explain bidirectional shift register.
 Shift register : Refer Q. 3.16, Page 3-20A, Unit-3.
 Need : Refer Q. 3.16, Page 3-20A, Unit-3.

Bidirectional shift register :
 1. It consists of four D flip-flops, four OR gates, eight AND gates and one NOT gate as shown in Fig. 4.

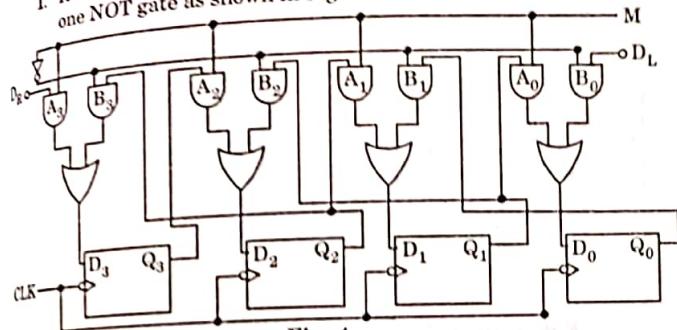


Fig. 4.

Operation :

- When mode control $M = 1$, all the A AND gates (A_1, A_2, A_3, A_4) 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.

4. Attempt any one part of the following questions : (7 x 1 = 7)
 a.i. Design a modulo-4 UP/DOWN counter using JK flip-flop.

- Ans:
- 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.
 - Two flip flops are required to design modulo-4 UP/DOWN counter.

State diagram :

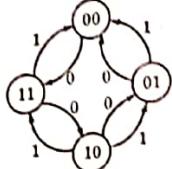


Fig. 5.

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			
	Q_A	Q_B	Q_{A+1}	Q_{B+1}	J_A	K_A	J_B	K_B
0	0	0	1	1	1	x	1	x
0	0	1	0	0	0	x	x	1
0	1	0	0	1	x	1	1	x
0	1	1	1	0	x	0	x	1
1	0	0	0	1	0	x	1	x
1	0	1	1	0	1	x	x	1
1	1	0	1	1	x	0	1	x
1	1	1	0	0	x	1	x	1

K-map simplification :

		Expression for J_A			
		$Q_A Q_B$	$\bar{Q}_A Q_B$	$Q_A \bar{Q}_B$	$\bar{Q}_A \bar{Q}_B$
		00	01	11	10
0		0	1	3	x
1		4	5	7	6
		1	x	x	x

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

Expression for J_B

		Expression for J_B			
		$Q_A Q_B$	$\bar{Q}_A Q_B$	$Q_A \bar{Q}_B$	$\bar{Q}_A \bar{Q}_B$
		00	01	11	10
0		0	1	3	2
1		4	5	7	6
		1	x	x	1

$$J_B = 1$$

		Expression for K_A			
		$Q_A Q_B$	$\bar{Q}_A Q_B$	$Q_A \bar{Q}_B$	$\bar{Q}_A \bar{Q}_B$
		00	01	11	10
0		0	1	3	2
1		4	5	7	6
		x	x	1	x

		Expression for K_B			
		$Q_A Q_B$	$\bar{Q}_A Q_B$	$Q_A \bar{Q}_B$	$\bar{Q}_A \bar{Q}_B$
		00	01	11	10
0		0	1	3	2
1		4	5	7	6
		1	x	x	1

$$K_B = 1$$

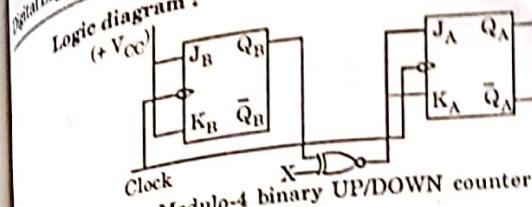
Logic diagram :
(+ V_{CC})

Fig. 6. Modulo-4 binary UP/DOWN counter.

- b. Design a ripple decade counter using JK flip-flop.
b. 4-bit binary ripple down counter : Refer Q. 3.24, Page 3-29A, Unit-3.
- b. What are critical race and non-critical race ? How can they be avoided ?
Refer Q. 4.20, Page 4-28A, Unit-4.
- b. Describe the hazards in digital circuits. How are these removed ? Design a hazards free circuit of the following Boolean function :
 $F(A, B, C) = \Sigma m(1, 2, 3, 5)$
Refer Q. 4.32, Page 4-43A, Unit-4.

Attempt any one part of the following questions : (7 x 1 = 7)
i. Describe the circuit and performance of CMOS inverter and state the characteristics of CMOS.

- ii. Differentiate between PLA and PAL. Realize the full adder circuit using PAL.
iii. Difference : Refer Q. 5.31, Page 5-32A, Unit-5.
Full adder using PAL : Refer Q. 5.30, Page 5-32A, Unit-5.

- iv. Discuss the concept of field programmable gate array (FPGA). Discuss the various structures of FPGA.
Refer Q. 5.40, Page 5-42A, Unit-5.

- v. Tabulate the truth table for 8 x 4 ROM that implements the Boolean function :
 $A(x, y, z) = \Sigma m(1, 2, 4, 6)$
 $B(x, y, z) = \Sigma m(0, 1, 6, 7)$
 $C(x, y, z) = \Sigma m(2, 6)$
 $D(x, y, z) = \Sigma m(1, 2, 3, 5, 7)$

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

Ans. Truth table for the given function :

Inputs			Outputs			
x	y	z	A	B	C	D
0	0	0	0	1	0	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	0	0	1
1	0	0	1	0	0	0
1	0	1	0	0	0	1
1	1	0	1	1	1	0
1	1	1	0	1	0	1

Logic diagram :
The ROM is programmed for truth table.

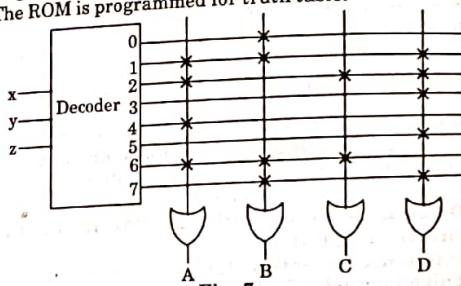


Fig. 7.

6. Attempt any one part of the following questions : (7 x 1 = 7)
 a. An asynchronous sequential logic circuit is described by the following excitation and output function

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

$$Z = Y$$

- i. Draw the logic diagram of the circuit.
 ii. Derive the transition table and output map.
 iii. Describe the behavior of the circuit.

- Ans.** Refer Q. 4.19, Page 4-26A, Unit-4.
 b. i. The code 101101010 is received, correct any errors. There are four parity bits and odd parity is used.

Ans. Received codeword

D ₉	P ₈	D ₇	D ₆	D ₅	P ₄	D ₃	P ₂	P ₁
1	0	1	1	0	1	0	1	0

Step 1: Analyze bits 1, 3, 5, 7
 $P_1 D_3 D_5 D_7 = 0 0 0 1 \rightarrow$ odd parity
 So there is no error

Step 2: Analyze bits 2, 3, 6, 7
 $P_2 D_3 D_6 D_7 = 1 0 1 1 \rightarrow$ odd parity
 So there is no error.

Step 3: Analyze bits 4, 5, 6, 7
 $P_4 D_5 D_6 D_7 = 1 0 1 1 \rightarrow$ odd parity
 So there is no error.

Step 4: Analyze bits 8, 9, 10, 11, 12, 13, 14, 15
 $P_8 P_9 = 0 1 \rightarrow$ odd parity
 So there is no error.

ii. Draw a full subtractor circuit using NAND gate.

Full subtractor using only NAND gates

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

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

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

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

By using the above expressions for D and B_{out} , the full subtractor is implemented using only NAND gates as shown in Fig. 8.

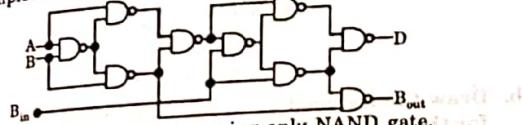


Fig. 8. Full subtractor using only NAND gate.

7. Attempt any one part of the following questions : (7 x 1 = 7)
 a. Derive the state table and state diagram for the sequential circuit is shown in Fig. 9.

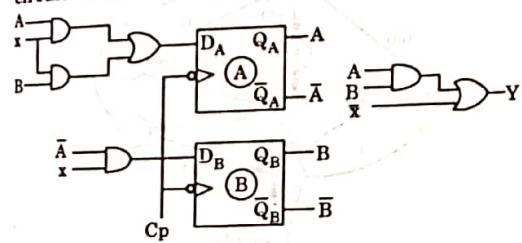


Fig. 9.

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

ANS

Scanned with CamScanner

$$Y = AB + \bar{B}$$

$$D_A = Ax + Bx$$

$$D_B = \bar{A}x$$

2. From eq. (1), (2) and (3) the state table will be

(1)

(2)

(3)

Table 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

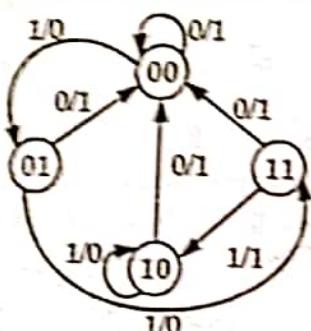


Fig. 10.

b. Draw the reduced state table and reduced state diagram for the state table given in Fig. 11.

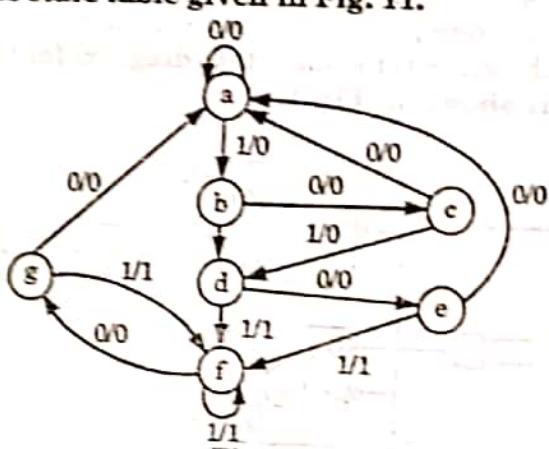


Fig. 11.

Refer Q. 4.10, Page 4-15A, Unit-4.

