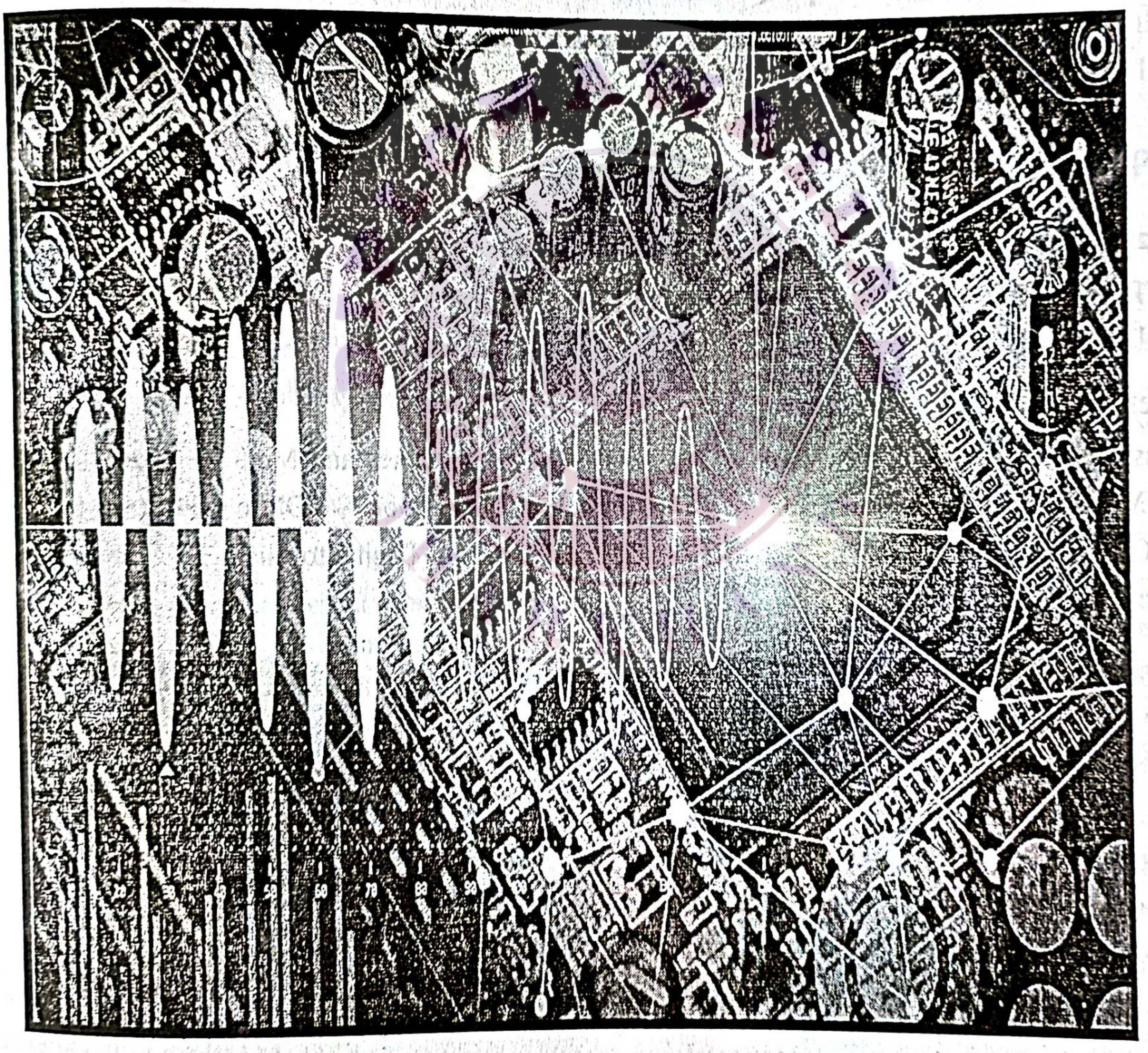
Digital Electronics



SYLLABUS

1. Number Systems & Boolean Algebra

Introduction to different number systems – Binary, Octal, Decimal, Hexadecimal Conversion from one number system to another. Boolean variables – Rules and laws of Boolean Algebra, De-Morgan's Theorem Karnaugh Maps and their use for simplification of Boolean expressions

2. Logic Gates

Logic Gates – AND, OR, NOT, NAND, NOR, XOR, XNOR: Symbolic representation and truth table Implementation of Boolean expressions and Logic Functions using gates Simplification of expressions

3. Combinational Logic Circuits

Arithmetic Circuits – Addition, Subtraction, 1's 2's Complement, Half Adder, Full Adder, Half Subtractor, Full Subtractor, Parallel and Series Adders, Encoder, Decoder Multiplexer – 2 to 1 MUX, 4 to 1 MUX, 8 to 1 MUX. Applications Demultiplexer – 1 to 2 DEMUX, 1- 4 DEMUX, 1- 8 DEMUX

4. Sequential Logic Circuits

Flip Flops – SR, JK, T, D, FF, JK-MS, Triggering Counters – 4 bit Up – Down Counters, Asynchronous/ Ripple Counter, Decade CounterMod 3, Mod 7 Counter, Johnson Counter, Ring Counter Registers – 4bit Shift Register : Serial In Serial Out, Serial in Parallel Out, Parallel In Serial Out, Parallel In Parallel Out

5. Memory Devices

Classification of Memories – RAM Organization, Address Lines and Memory Sixe, Static RAM, Bipolar RAM, cell Dynamic RAM, D RAM, DDR RAM Read Only memory – ROM organization, Expanding memory, PROM, EPROM, EEPROM, Flash memory Data Converters – Digital to Analog converters, Analog to Digital Converters

□□□

Chapter 1

Number System & Boolean Algebra

Q.1. What is Number system ? Explain.

Ans. A number is a mathematical value used for counting or measuring or labelling objects. Numbers are used to perform arithmetic calculations. Examples of numbers are natural numbers, whole numbers, rational and irrational numbers, etc. 0 is also a number that represents a null value.

A number has many other variations such as even and odd numbers, prime and composite numbers. Even and odd terms are used when a number is divisible by 2 or not, whereas prime and composite differentiate between the numbers that have only two factors and more than two factors, respectively.

In a number system, these numbers are used as digits. 0 and 1 are the most common digits in the number system, that are used to represent binary numbers. On the other hand, 0 to 9 digits are also used for other number systems. Let us learn here the types of number systems.

Q.2. Write short notes of

(i) Decimal number system (Bh.2012,2016,2019)

Ans: The decimal number system has a base of 10 because it uses ten digits from 0 to 9. In the decimal number system, the positions successive to the left of the decimal point represent units, tens, hundreds, thousands and so on. This system is expressed in decimal numbers. Every position shows a particular power of the base (10).

Example of Decimal Number System:

The decimal number 1457 consists of the digit 7 in the units position, 5 in the tens place, 4 in the hundreds position, and 1 in the thousands place whose value can be written as:

$= (1\times10^3) + (4\times10^2) + (5\times10^1) + (7\times10^0)$

$= (1\times1000) + (4\times100) + (5\times10) + (7\times1)$

$= 1000 + 400 + 50 + 7$

$= 1457$

(ii) Octal number system

Ans. In the octal number system, the base is 8 and it uses numbers from 0 to 7 to represent numbers. Octal numbers are commonly used in computer applications. Converting an octal number to decimal is the same as decimal conversion and is explained below using an example.

Example: Convert 215_8 into decimal.

Solution: $215_8 = 2\times8^2 + 1\times8^1 + 5\times8^0$

$= 2\times64 + 1\times8 + 5\times1$

$= 128 + 8 + 5$

$= 141_{10}$

(iii) Hexadecimal Number system (Bh.2014)

Ans. In the hexadecimal system, numbers are written or represented with base 16. In the hex system, the numbers are first represented just like in the decimal system, i.e. from 0 to 9. Then, the numbers are represented using the alphabet from A to F.

Q.3 *What is binary number system? Explain* **(Bh.2010-Odd)**

Ans. The binary number system is a base-2 number system. This means it has only two different symbols or digits 0 and 1. All binary numbers consists of a string of 0's and 1's. Examples of binary number are 10, 101 and 1011. A group of eight bits is known as a byte and group of four bits is known as nibble.

Like the decimal number system, binary number system is also positionally weighted. In this system, the position value of each bit corresponds to some power of 2. **For an example** binary number is

$(1101011)_2$

MSB LSB

The binary weights of above binary number is

$(1101011)_2$

$= 1\times2^3+1\times2^2+0\times2^1+1\times2^0+0\times2^{-1}+1\times2^{-2}+1\times2^{-3}$

$= 8+4+0+1+0+\frac{1}{4}+\frac{1}{8}$

$= (13.375)_{10}$.

Q.4 *Convert following into octal number system.*

(a) $(247)_{10}$ (b) $(0.6875)_{10}$ (c) $(3287.51)_{10}$.

Ans. (a) $(247)_{10}$

8	247	Remainder
8	30	7
8	3	6
	0	3

$(247)10 = (367)_8$

(b) $(0.6875)_{10}$

$(0.6875)\times8 = 5.5 = 0.5$ with carry of 5

$0.5\times8 = 4.0 = 0.0$ with carry of 4

$\therefore\ (0.6875)_{10} = (0.54)_8$.

(c) (3287.51)10

Integer

8	3287	Remainder
8	410	7
8	51	2
8	6	3
	0	6

$\therefore\ (3287)_{10} = (6327)_8$

Fraction:

$0.51\times8 = 4.08 = 0.08$ with carry of 4

$0.08\times8 = 0.64 = 0.64$ with carry of 0

$0.64\times8 = 5.12 = 0.12$ with carry of 5

$0.12\times8 = 0.96 = 0.96$ with carry of 0

$0.96\times8 = 7.68 = 0.68$ with carry of 7

$0.68\times8 = 5.44 = 0.44$ with carry of 5

$\therefore\ (0.51)_{10} = (0.405075)_8$

$\therefore\ (3287.51)_{10} = (6327.405075)_8$.

Q.5 *Determine the decimal numbers of the following binary numbers.*

(a) 110101 (b) 1100.1011 (c) 0.10101

Ans. (a) $(110101)_2 = 1\times2^5+1\times2^4+0\times2^3+1\times2^2+0\times2^1+1\times2^0$

$= 32+16+0+4+0+1$

$= (53)_{10}$

(b) $(1100.1011)_2$

$= 1\times2^3+1\times2^2+0\times2^1+0\times2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}+1\times2^{-4}$

$= 8+4+0+0+\frac{1}{2}+0+\frac{1}{8}+\frac{1}{16}$

$= 12+0.6875$

$= (12.6875)_{10}$.

(c) $(0.10101)_2$

$= 1\times2^{-1}+0\times2^{-2}+1\times2^{-3}+0\times2^{-4}+1\times2^{-5}$

$= \frac{1}{2}+0+\frac{1}{8}+0+\frac{1}{32} = (0.65625)_{10}$.

Q.6. *Convert :*

(a) $(3A\,2F)_{16}$ into decimal

(b) $(2F\,9A)_{16}$ into binary

(c) $(0.BF85)_{16}$ into octal

Ans.(a) $(3A.2F)_{16}$

$= 3\times16^1+10\times16^0+2\times16^{-1}+15\times16^{-2}$

$= 48+10+\frac{2}{16}+\frac{15}{16^2}$

$= (58.1836)_{10}$.

(b) $(2F\,9A)_{16} = (0010\ 1111\ 1001\ 1010)_2$

$= (0010\ 1111\ 1001\ 1010)_2$.

(c) $(0.BF85)_{16} = (0.1011\ 1111\ 1000\ 0101)_2$

$= (0.101\ 111\ 111\ 000\ 010\ 100)$

$= (0.577024)_8$

Q.7. *Convert following numbers in Hexadecimal equivalent:*

(a) $(25.625)_{10}$ (b) $(28.50)_{10}$.

Ans.(a) Integer part

16	25	Remainder
16	1	9
	0	1

$(25)_{10} = (19)_{16}$

Fractional part

$0.625\times16 = 10.0 = A$

Thus $(0.625)_{10} = (A)_{16}$

$\therefore\ (25.625)_{10} = (19.A)_{16}$.

(b) $(28.50)_{10}$

$(28)_{10}$

16	28	Remainder
16	1	12 → C
	0	1

$\therefore\ (28)_{10} = (1C)_{16}$

$(0.50)\times16 = 8.00$

$(0.50)_{10} = (0.8)_{16}$

$\therefore\ (28.50)_{10} = (1C.8)_{16}$.

Q.8. *Convert following number in binary equivalent number:*

(a) 37 (b) 256 (c) 10.625.

[Bh.2009]

Ans.(a) $(37)_{10}$

2	37	Remainder
2	18	1
2	9	0
2	4	1
2	2	0
2	1	0
	0	1

$\therefore\ (37)_{10} = (100101)_2$.

(b) $(256)_{10}$

2	256	Remainder
2	128	0
2	64	0
2	32	0
2	16	0
2	8	0
2	4	0
2	2	0
2	1	0
	0	1

$\therefore\ (256)_{10} = (100000000)_2$.

(c) $(10.625)_{10}$

Integer Part

2	10	Remainder
2	5	0
2	2	1
2	1	0
	0	1

Fractional part:

$0.625\times2 = 1.25 \rightarrow 1$

$0.25\times2 = 0.50 \rightarrow 0$

$0.5\times2 = 1.0 \rightarrow 1$

$\therefore\ (0.625)_{10} = (0.101)_2$

$\therefore\ (10)_{10} = (1010)_2$

$\therefore\ (10..625)_{10} = (1010.101)_2$.

Q.9. *Express the following decimal numbers in the binary form.*

(i) 25.5 (ii) 10.625 (iii) 0.6875

Ans. (i) 25.5 :– Integer part

	Quotient	Reminder
$\frac{25}{2}$	12	1
$\frac{12}{2}$	6	0
$\frac{6}{2}$	3	0
$\frac{3}{2}$	1	1
$\frac{1}{2}$	0	1

1 1 0 0 1

Therefore $(25)_{10} = (11001)_2$

Fractional part :

$0.5 \times 2 = 1.0$

i.e., $(0.5)_{10} = (0.1)_2$

Therefore, $(25.5)_{10} = (11001.1)_2$ **Ans.**

(ii) $(10.625)_{10}$:– Integer part

		Q	R
$\frac{10}{2}$	=	5	0
$\frac{5}{2}$	=	2	1
$\frac{2}{2}$	=	1	0

$(10)_{10} = (1010)_2$

$0.625 \times 2 = 1.250$	0.250	0.500
↓	$\times 2$	$\times 2$
1	0.500	1.000
	↓	↓
	0	1

$(0.625)_{10} = (0.101)_2$

$\therefore (10.625)_{10} = (1010.101)_2$ **Ans.**

(iii) $(0.6875)_{10}$

0.6875	0.3750	0.7500	0.5000
$\times 2$	$\times 2$	$\times 2$	$\times 2$
1.3750	0.7500	1.5000	1.0000
↓	↓	↓	↓
1	0	1	1

$\therefore (0.6875)_{10} = (0.1011)_2$ **Ans.**

Q.10. *What do you mean by signed numbers?*

Ans. In the decimal number system, plus (+) sign. is used for positive number and minus (–) sign. is used for negative number. This representation of number is called signed number.

In digital number system, there is only two symbol (0, 1). So, an additional bit is used as the sign. bit and is placed as the most significant bit zero (0) is used to represent +ve number and

$$\begin{array}{r} 1110 \\ -1010 \\ \hline 0100 \end{array}$$

Direct subtraction

Using 1's complement

$$\begin{array}{r} 1110 \\ -0101 \\ \hline 10011 \\ +1 \\ \hline 0100 \end{array}$$ (1's comp. of 1010) → (Adding end around carry)

Final answer

Q.11. *Subtract 1001 from 0111 using one's complement method.*

Ans.

$$\begin{array}{r} 0111 \\ 1001 \\ \hline -0010 \end{array}$$ direct subtraction

using one's complement

$$\begin{array}{r} 0111 \\ +0110 \\ \hline 1101 \end{array}$$ (1's comp. of 1001)

If there is no carry, answer is in 1's complement form and opposite in sign.

$\therefore$ 1's complement of 1101 = 0010 and

Final answer = – 0010.

Therefore it may be noted here

(i) If there is no any carry, it means answer is in –ve sign with 1's complement of result.

(ii) If there is carry, we have to add end annual carry to the result and this is final answer.

Q.12. *How we use 2's complement for binary subtraction? Subtract.* **[Bh.20013,16]**

(a) $(1011)_2$ from $(1100)_2$ and

(b) $(11100)_2$ from $(10011)_2$ using 2's complement representation.

Ans. 2's complement of a binary number is equal to the 1's complement of the number plus one

$\therefore$ 2's complement of a binary number

= 1's complement + 1.

2's complement of 10110

= 01001 + 1 = 01010.

(a) Subtract $(1011)_2$ from $(1100)_2$

2's complement method

$$\begin{array}{r} 1100 \\ -1011 \\ \hline 0001 \end{array}$$

$$\begin{array}{r} 1100 \\ +0101 \\ \hline 10001 \\ \hline 0001 \end{array}$$ (2's comp. of 1011) neglect carry Final answer

(b) Subtract $(11100)_2$ from $(10011)_2$ direct subtraction

$$\begin{array}{r} 10011 \\ -11100 \\ \hline -01001 \end{array}$$

2's comp. Subtraction

$$\begin{array}{r} 10011 \\ +00100 \\ \hline -10111 \end{array}$$ (2's comp. of 11100)

If there is no carry then answer is in 2's complement and opposite in sign.

$\therefore$ 2's comp. of 10111 = 01000 + 1 = 01001 and final answer is – 01001.

Q.13. *What is meant by codes ?* **[Bh.2012,2016,2018]**

Ans. All digital system use some form of binary numbers for their internal operations. But external signal (data) may be numeric alphabets or special characters. For operaion of digital system these data are to be converted into binary format. These are various ways of doing this and this process is known as encoding. For encoding, there are various codes for different purposes. These are:-

1. Weighted codes
2. Non-weighted codes
3. Alphanumeric codes
4. Error detecting and correcting codes.

Q.14. *What is weighted codes? Explain in brief.* **[Bh.20014,17]**

Ans. Weighted binary codes are those which obey the positional weight principle. Each position of number represents a specific weight. Binary codes and BCD (Binary coded Decimal) codes are the example of weighted codes.

Binary codes: In this, the decimal number are converted to their binary equivalents.

BCD codes: In this code, each decimal digit is represented by a 4-bit binary number. The positional weights, to the binary bits in BCD code are 8-4-2-1 with 1 corresponding to LSB and 8 corresponding to MSB. It is also known as 8-4-2-1 code. This code is used in digital calculators, voltmeters etc.

If decimal number = 2 5

BCD equivalent → 0010 0101

$(164)_{10}$ → (0001 0110 0100)BCD

Q.15. *What is Non-weighted codes? Explain in binary.* **[Bh.20014,17]**

Ans. In these codes, the positional weights are not assigned. The example of non-weighted codes are excess-3 and gray code.

Excess-3 code:- The excess - 3 code for a decimal number can be obtained in the same manner as BCD expect that 3 is added to each decimal digit before encoding it in binary.

For example to encode the decimal digit 5 into excess-3 code, we have to add 3 to 5 and otain 8. The digit is encoded in equivalent to 4-bit binary code 1000.

Represent 32 into the excess-3 code

3	2	
+3	+3	
↓ 6	↓ 5	(add 3 to each digit)
0110	0101	(convert to 4-bit binary code)

Gray Code:- This code is often used in digital systems because it has the advantage that only one bit in the numerical representation changes between successive numbers. Its primary applications is in the location of angles on a rotary shaft.

0 – 000

1 – 001

2 – 011

3 – 010 7 – 100

4 – 110

5 – 111
6 – 101
7 – 100

Q.16. *What do you mean by Alphanumeric codes?* **[Bh.20014,17]**

Ans. The alphanumeric codes are designed to represent numbers as well as alphabetic characters. If we use an n-bit binary code, we can represent 2^n element using this code. There is need to represent more than by characters (10 digit, 26 upper case letter, 26 lower case letter) including the lower case letters and special control characters for the transmission of digital information. For this reason the following two codes are normally used.

1. Extended BCD Interchange code (EBCDIC)
2. American standard code for Information Interchange (ASCII).

1. *EBCDIC*:- It is based on the binary coded decimal format. This is an 8-bit code without parity. A nineth bit can be added for parity.

2. *ASCII*:- It is a 7 bit code, with 7 bit we can code upto 128 character, which is enough for the full upper case and lower-case alphabet, numbers, punctuation marks and control characters. It is most widely used Alphanumeric code.

If a parity check is wanted, a parity bit is added to basic 7-bit code in the MSB position.

Q.17. *What is the difference between binary code and BCD?*

Ans. BCD is not a number system like binary. It is a decimal system in which each decimal digit encoded in its binary equivalent. In binary code, complete decimal number is converted into binary form. But in BCD each decimal digit is converted into binary indivisually.

Q.18. *What do you meant by error detecting and correcting codes?*

Ans. When digital data or information is transmitted from one system to the other system or circuit, an error may be occur due to electrical disturbance or noise. This means a signal corresponding to 0 may change to 1 or 1 to 0. In complex digital systems, millions of bits per second are transmitted. Therefore it is wanted that error is as minimum as possible and it must be detectable.

A simple process of improving data transmission in digital system is by adding one additional bit in the data known as parity bit. This extra bit allows the detection of a single bit error in transmission. It is of two type - (i) odd parity (ii) even parity.
This extra bit is transmitted with the code and it is used at the receiving end to detect error. Using parity bit we can only detect single bit error but can not correct it.

Hamming code is one bit error detecting and correcting code.

□□□

OBJECTIVE TYPE QUESTIONS

1. Convert hexadecimal value 16 to decimal.
(A) 2210 (B) 1610
(C) 1010 (D) 2010
Ans. (A)

2. Convert the following decimal number to 8-bit binary. 187
(A) 101110112 (B) 110111012
(C) 101111012 (D) 101111002
Ans. (A)

3. Convert binary 111111110010 to hexadecimal.
(A) EE216 (B) FF216
C. 2FE16 (D) FD216
Ans. (B)

4. Convert the following binary number to decimal. 010112
(A) 11 (B) 35
(C) 15 (D) 10
Ans. (A)

5. Convert the binary number 1001.00102 to decimal.
(A) 90.125 (B) 9.125
(C) 125 (D) 12.5
Ans. (B)

6. Decode the following ASCII message.
1010011101010010101011000100101100 1
0100000100100010000011010010100010 0
(A) STUDYHARD (B) STUDY HARD
(C) stydyhard (D) study hard
Ans. (B)

7. The voltages in digital electronics are continuously variable.
(A) True (B) False
Ans. (B)

8. One hex digit is sometimes referred to as a(n) :
(A) byte (B) nibble
(C) grouping (D) instruction
Ans. (B)

9. Which of the following is the most widely used alphanumeric code for computer input and output ?
(A) Gray (B) ASCII
(C) Parity (D) EBCDIC
Ans. (B)

10. If a typical PC uses a 20-bit address code, how much memory can the CPU address ?
(A) 20 MB (B) 10 MB
(C) 1 MB (D) 580 MB
Ans. (C)

11. Convert 59.7210 to BCD.
(A) 111011 (B) 01011001.01110010
(C) 1110.11 (D) 01011001011110010
Ans. (B)

12. Convert 8B3F16 to binary.
(A) 35647 (B) 011010
(C) 1011001111100011 (D) 1000101100111111
Ans. (D)

13. Which is typically the longest: bit, byte, nibble, word?
(A) Bit (B) Byte
(C) Nibble (D) Word
Ans. (D)

14. Assign the proper odd parity bit to the code 111001.
(A) 1111011 (B) 1111001
(C) 0111111 (D) 0011111
Ans. (B)

15. Convert decimal 64 to binary.
(A) 01010010 (B) 01000000
(C) 00110110 (D) 01001000
Ans. (B)

16. Convert hexadecimal value C1 to binary.
(A) 11000001 (B) 1000111
(C) 111000100 (D) 111000001
Ans. (A)

17. Convert the following octal number to decimal. 178
(A) 51 (B) 82
(C) 57 (D) 15
Ans. (D)

18. Convert the following binary number to octal. 0101111002
(A) 1728 (B) 2728
(C) 1748 (D) 2748
Ans. (D)

19. How many binary digits are required to count to 10010?
(A) 7 (B) 2
(C) 3 (D) 100
Ans. (A)

20. The BCD number for decimal 347 is .....
(A) 1100 1011 1000 (B) 0011 0100 0111
(C) 0011 0100 0001 (D) 1100 1011 0110
Ans. (B)

21. The binary number for octal 458 is .....
(A) 100010 (B) 100101
(C) 110101 (D) 100100
Ans. (B)

22. The sum of 11101 + 10111 equals ..... .
(A) 110011 (B) 100001
(C) 110100 (D) 100100
Ans. (C)

23. Convert the following binary number to decimal.
100110102
(A) 154 (B) 155
(C) 153 (D) 157
Ans. (A)

24. The decimal number 188 is equal to the binary number ..... .
(A) 10111100 (B) 0111000
(C) 1100011 (D) 1111000
Ans. (A)

25. Convert the following binary number to octal.
0011010112
(A) 1538 (B) 3518
(C) 2538 (D) 3528
Ans. (A)

26. How many bits are in an ASCII character?
(A) 16 (B) 8
(C) 7 (D) 4
Ans. (C)

27. A binary number's value changes most drastically when the _____ is changed.
(A) MSB (B) frequency
(C) LSB (D) duty cycle
Ans. (A)

28. Convert decimal 213 to binary.
(A) 11001101 (B) 11010101
(C) 01111001 (D) 11100011
Ans. (B)

29. The decimal number for octal 748 is ..... .
(A) 74 (B) 60
(C) 22 (D) 62
Ans. (B)

30. The sum of the two BCD numbers, 0011 + 0011, is ..... .
(A) 0110 (B) 0111
(C) 0011 (D) 1100
Ans. (A)

31. Convert binary 01001110 to decimal.
(A) 4E (B) 78
(C) 76 (D) 116
Ans. (B)

32. Which is not a word size ?
(A) 64 (B) 28
(C) 16 (D) 8
Ans. (B)

33. The octal numbering system:
(A) simplifies tasks
(B) groups binary numbers in groups of 4
(C) saves time
(D) simplifies tasks and saves time
Ans. (D)

34. The binary number 1110 is equal to the decimal number ..... .
(A) 3 (B) 1
(C) 7 (D) 14
Ans. (D)

35. Convert the following octal number to binary.
768
(A) 1101112 (B) 1111102
(C) 1111002 (D) 1001112
Ans. (B)

36. Convert 1100101000101012 to hexadecimal.
(A) 121035 (B) CA35
(C) 53ACI (D) 530121
Ans. (B)

37. Convert the following decimal number to octal.
281
(A) 1348 (B) 4318
(C) 3318 (D) 1338
Ans. (B)

38. When using even parity, where is the parity bit placed?
(A) Before the MSB (B) After the LSB
(C) In the parity word (D) After the odd parity bit
Ans. (A)

39. Convert the following octal number to decimal.
358
(A) 71 (B) 17
(C) 92 (D) 29
Ans. (D)

40. Convert binary 11001111 to hexadecimal.
(A) 8F16 (B) CE16
(C) DF16 (D) CF16
Ans. (D)

41. Convert 17318 to decimal.
(A) 216.4 (B) 985
(C) 3D9 (D) 1123
Ans. (B)

42. An analog signal has a range from 0 V to 5 V. What is the total number of analog possibilities within this range?
(A) 5 (B) 50
(C) 250 (D) infinite
Ans. (D)

□□□

Chapter 2

Logic Gates

Q.1. Define logic gates ?

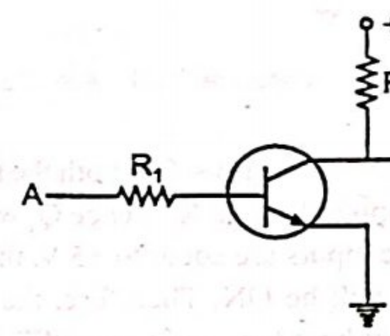
Ans. Logic gates are devices that have two or more inputs and one output.

- Logic gates are the basic elements of a digital system. Each gate is given several inputs to act upon.
- The output will be high or low depending upon the combination of high and low inputs used and the type of gate used.
- Logic gates are the basic building blocks of digital circuits. It is a device which generates an output corresponding to the combinations of input levels. Logic gates can be implemented using diodes and transistors.
- The input and outputs of logic gates are having two states i.e. logic 1 or logic 0 or High or Low value.

Q.1. Draw symbol and truth table of (i) AND gate (ii) NOR gate (iii) NOT gate ?

Ans. Inverter : NOT Gate :

- NOT gate is also known as inverter. It has only one input and one corresponding output. The output is always the compliment of the given input i.e. when input is in logic 0 state, output is logic 1 state and when input is logic 1 state, output is logic 0 state. It is expressed as $Y = \overline{A}$.
- The discrete NOT gate may be realized using transistors. The input A may be either 0V or +5V. When A = 0 V, then Q_1 will be OFF. No Current flows through the resistor and there will be no voltage drop across the resistor. As a result output voltage Y corresponds to +5 V. When the input A = +5 V, the transistor Q_1 is ON and the output voltage $Y = V_{CE(sat)}$ corresponds to 0 V.



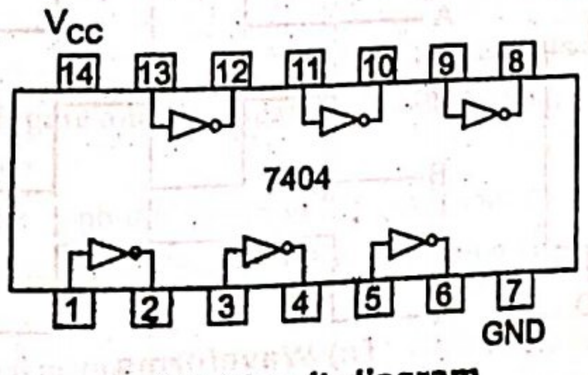
Transistor Inverter NOT Gate

Symbol : The symbol is shown in Fig. (a)

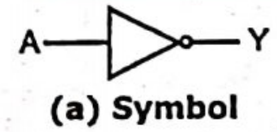
- The circuit performs a basic logic function called "inversion" or complementation. The inverter has one input and one output.
- The output logic level for inverter is always opposite to the logic level of its input. It is expressed as $Y = \overline{A}$.

Truth Table

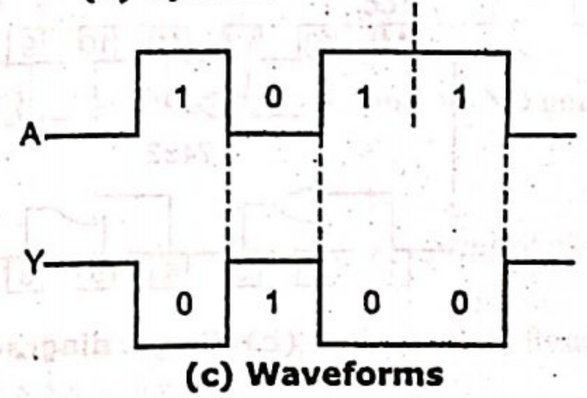
Input	Output
A	$Y = \overline{A}$
0	1
1	0



(b) Circuit diagram



(a) Symbol

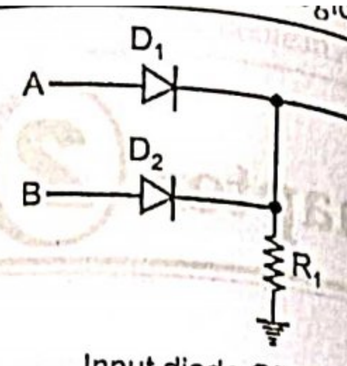


(c) Waveforms

- Fig. (b) shows the pin diagram of a 7404, a TTL Hex Inverter. This IC contains six NOT gates.

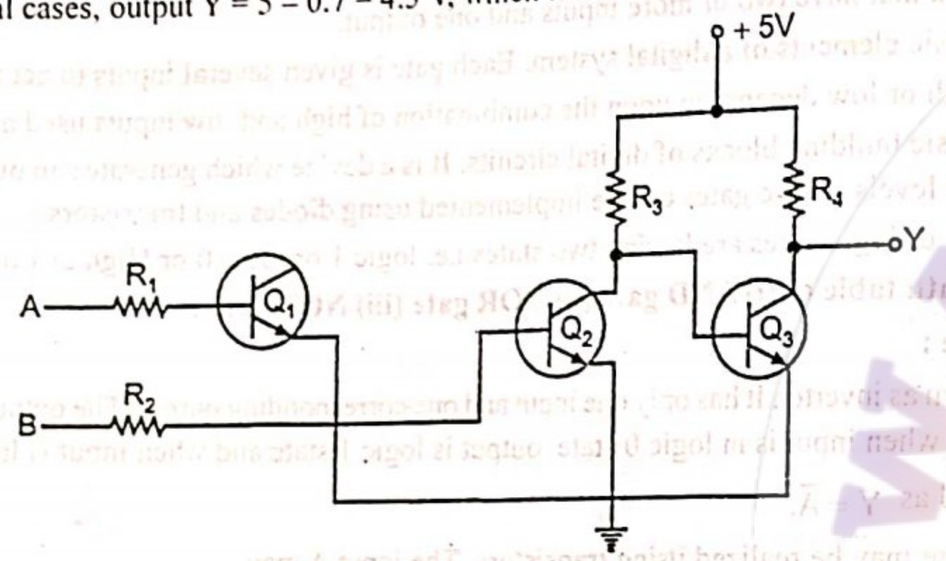
(ii) OR Gate :

- This circuit performs the function of logical addition and is expressed as Y = A + B. The output is high even when any one input is high and output is low when both the inputs are low.
- OR gate produces and output state of logic 1, even if any one of its input is in logic 1 state. It is represented as Y = A + B.



Input diode OR gate

- When A = 0 V, B = 0 V, both the diodes are OFF. No current flows through the resistor and there will be no voltage drop across the resistor, Output Y = 0. When A = +5V, B = +5V, both the diodes are ON, and act as short circuit. Output Y will be +5V. In practical cases, output Y = 5 – 0.7 = 4.3 V, which is also considered as logic 1 state.

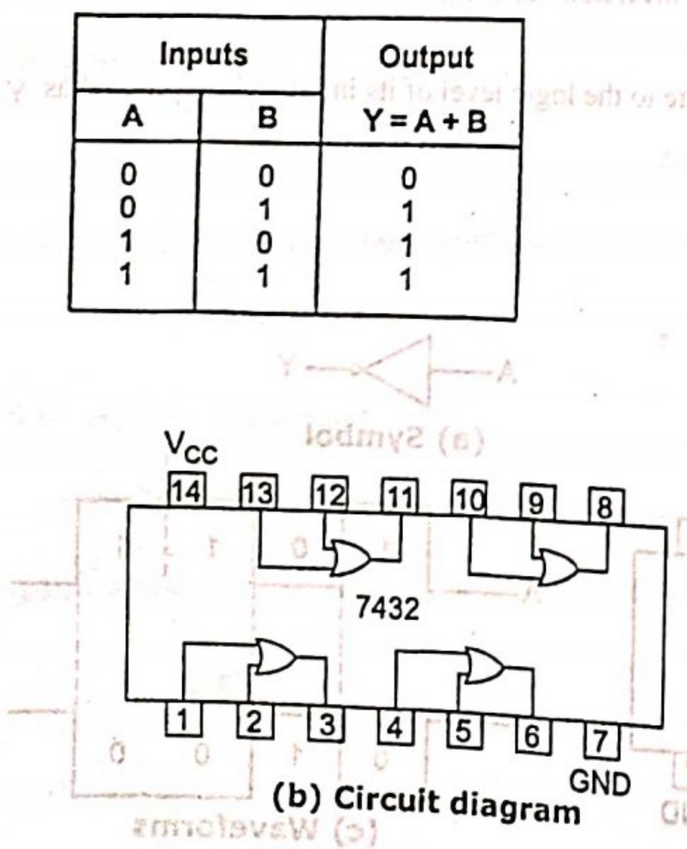


input transistor OR Gate

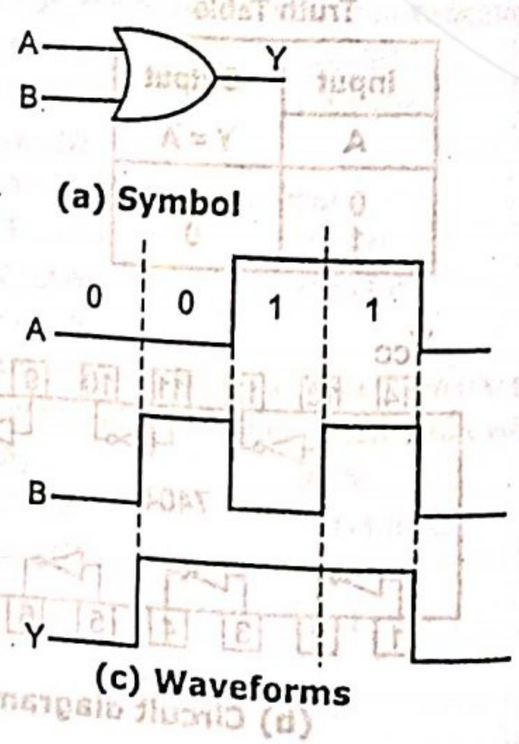
- When A = 0 V and B = 0 V, both the transistors Q_1 and Q_2 are OFF. At the same time, transistor Q_3 gets sufficient base drive from supply through R_3, hence Q_3 will be ON. Thus. output $Y = V_{CE(sat)}$ corresponds to 0 V. When either inputs A and B or both the inputs are equal to +5 V, then the corresponding transistors either Q_1 or Q_2 will be ON or both the transistors Q_1 and Q_2 will be ON. Therefore, the voltage at collector of Q_1 is $V_{CE(Sat)}$ corresponds to 0 V. Thus Q_3 does not receive sufficient base bias and turns OFF. Hence final output voltage corresponds to + 5 V (logic 1 state).

Truth Table

Inputs		Output
A	B	Y = A + B
0	0	0
0	1	1
1	0	1
1	1	1



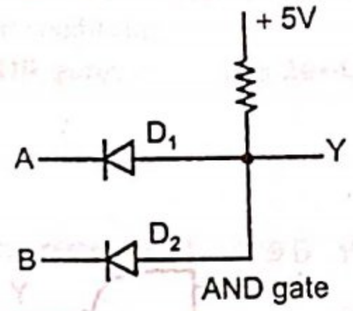
(b) Circuit diagram



(c) Waveforms

(iii) AND Gate :

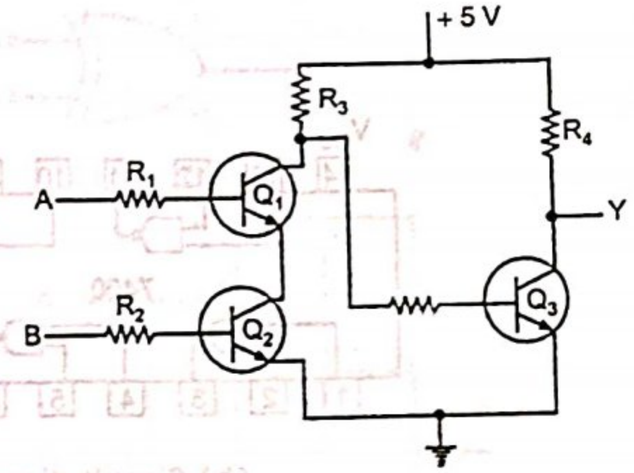
- An AND gate produces logic 1 output state when both of its inputs are at logic 1 state and produces logic 0 at the output if any of its input is at logic 0. If A and B are both at logic 1, then output Y is also at logic 1.
- Discrete AND gates can be realized using diodes or transistors. In diode AND gate, if both the inputs are A = +5 V and B = +5 V, then diodes are OFF. As a result no current flows through the resistor and there is no voltage drop across the resistor. The output Y will be +5 V. Similarly when both the inputs A and B are 0 V, then both the diodes D_1 and D_2 are ON and act as short circuit. The output Y will be 0 V. Practically, output Y is equal to 0.6 V or 0.7 V. which represents logic 0 state.



input diode AND Gate

2 Input Transistor AND Gate

- When A = B = 0 V or when A = 0 V and B = +5 V or when A = +5 V and B = 0 V, both the tran-sistors Q_1 and Q_2 are in OFF state. At the same time, transistor Q_3 gets enough base drive from the supply through resistor R_3 and hence Q_3 will be ON. Thus, output voltage $Y = V_{CE(Sat)}$ corresponds to 0 V. When both the inputs are equal to +5 V, transistors Q_1 and Q_2 will be ON, thus the voltage at the collector of transistor Q_1 will drop. Transistor Q_3 does not get sufficient base drive and hence turns OFF. No current flows through the coHector resistor R_4 of Q_3, therefore no voltage drop across it. Thus, final output voltage corresponds to +5V.

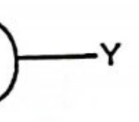


Symbol : The symbol is shown in Fig. 2.8 (a).

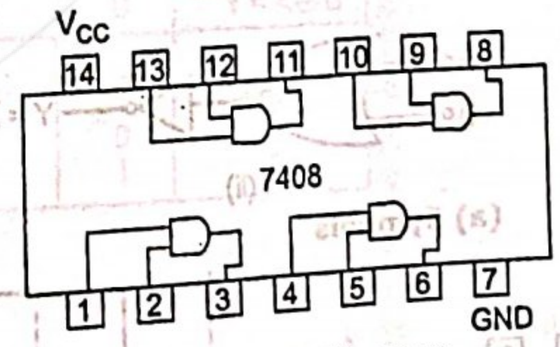
- This circuit performs the function of logical multiplication and is expressed as Y = A · B. The output is high only when both the inputs are high.

Truth Table

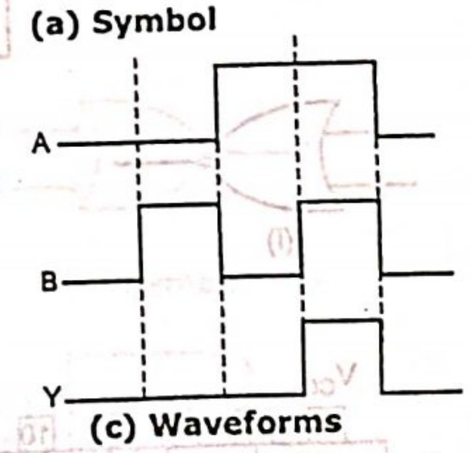
Inputs		Output
A	B	Y = A • B
0	0	0
0	1	0
1	0	0
1	1	1



(a) Symbol



(b) Circuit diagram



(c) Waveforms

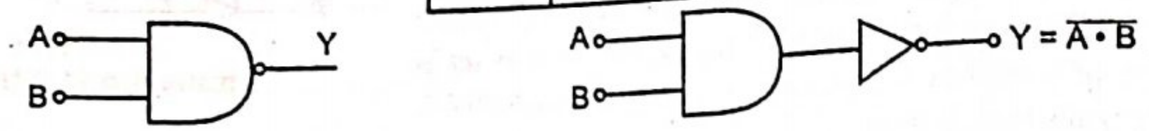
Q.2. Why NAND gate and NOR gates are called as universal gate ? Realize OR gate using NAND gate ?

Ans. NAND Gate :

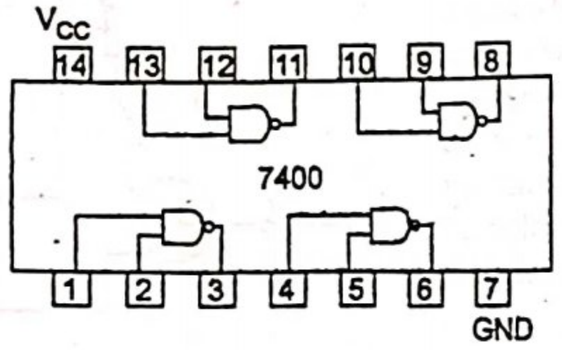
- **Symbol :** The symbol is shown in Fig. (a)(i) and (ii).
- NAND gate means NOT-AND. It has two or more inputs, but only one output which is complement of the AND product of all inputs.
- It is expressed mathematically as $Y = \overline{A \cdot B}$. The truth table and NAND gate can be easily obtained from the truth table of AND gate by simply inverting it to obtain the final output of NAND gate.

Truth Table

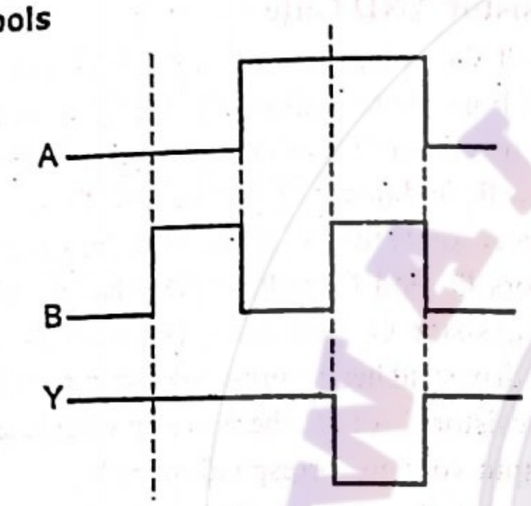
Inputs		Output
A	B	$Y = \overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0



(a) Symbols



(b) Circuit diagram



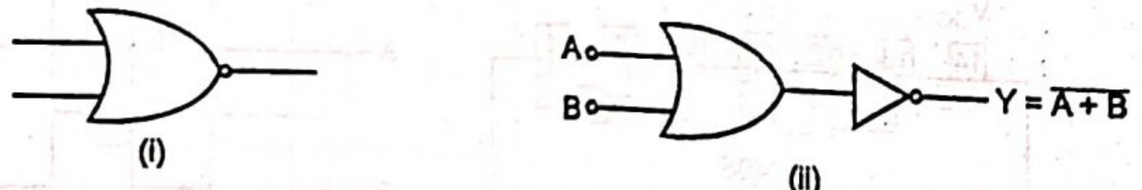
(c) Waveforms

(ii) NOR Gate :

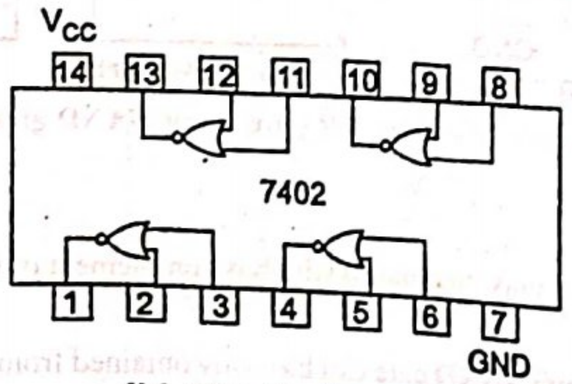
- **Symbol :** The symbol is shown in Fig. (a)(i) and (ii).
- NOR gate actually means NOT-OR. It has two or more inputs, but only one output which is the complement of the logical addition of two or more inputs.

Truth Table

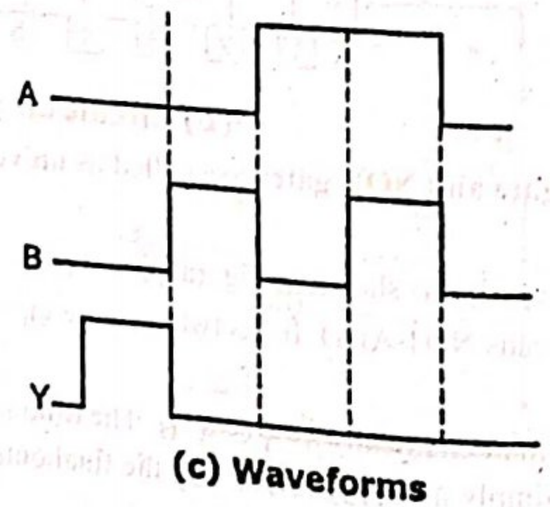
Inputs		Output
A	B	$Y = \overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0



(a) Symbols



(b) Circuit diagram



(c) Waveforms

- The NOR gate output is exact inverse of the OR gate output for all possible input conditions.

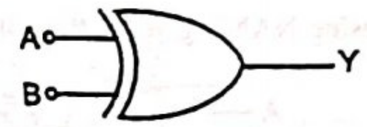
Q.4. Draw the symbol and write logical equation of output fo EX-OR and EX-NOR gates ? [Bh.2014,2017,2019]
Or Draw symbol of EX-OR gate and write its truth table. ?
Or Draw symbol, truth table and logical equation of EX-OR ?

Ans. EX-OR (Exclusive-OR) Gate : Symbol : The symbol is shown in Fig. (a)

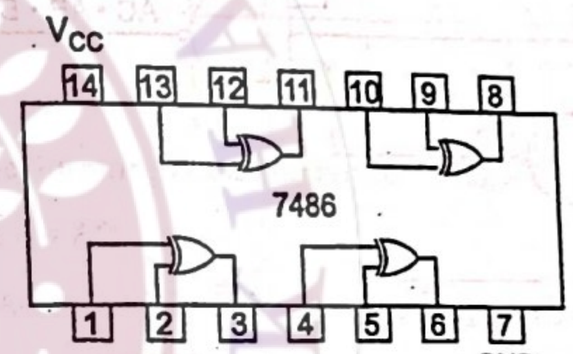
- The EX-OR operation gives high output when any one of its input is high. It is expressed as $Y = A \oplus B$ $Y = \overline{A}B + A\overline{B}$.
- The EX-OR gate is mostly used as a comparator.

Truth Table

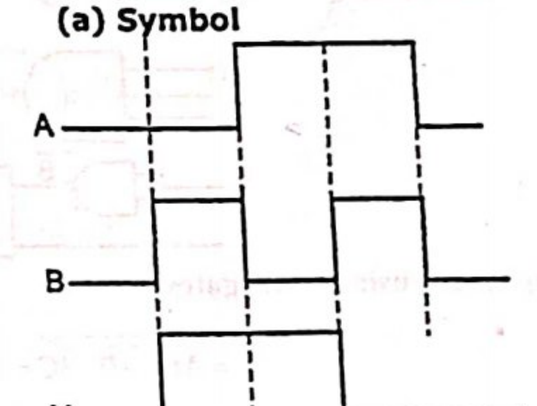
Inputs		Output
A	B	$Y = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0



(a) Symbol



(b) Circuit diagram



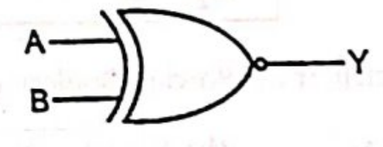
(c) Waveforms

(ii) EX-NOR Gate :

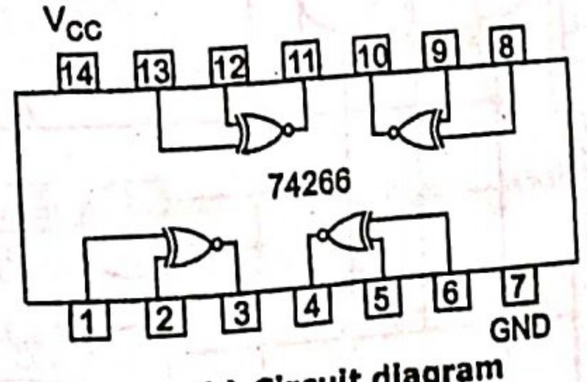
- **Symbol :** The symbol is shown in Fig. (a).
- The EX-NOR gate can be expressed at $Y = \overline{A \oplus B} = AB + \overline{A}\,\overline{B}$.
- The EX-NOR operation gives high output for both inputs low or both inputs high.

Truth Table

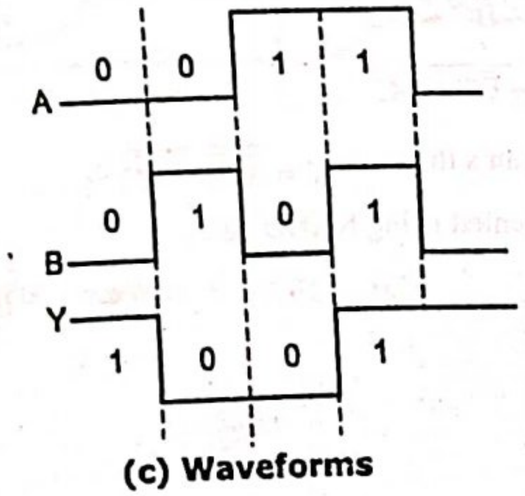
Inputs		Output
A	B	$Y = \overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1



(a) Symbol



(b) Circuit diagram



(c) Waveforms

Solved Problems

Example 1 : Realize the following function using NAND and NOR gates

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

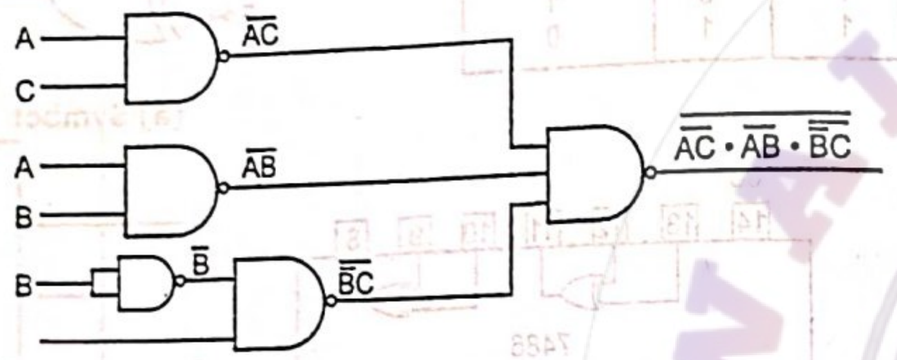
Solution : $Y = (A + \overline{B}C)(C + AB) = AC + AB + \overline{B}C + AB\overline{B}C$

$= AC + AB + \overline{B}C + 0$ $(\because B \cdot \overline{B} = 0)$

$= AC + AB + \overline{B}C = \overline{\overline{AC + AB + \overline{B}C}}$ $(\because \overline{\overline{A}} = A)$

Applying De Morgan's theorem $= \overline{\overline{AC} \cdot \overline{AB} \cdot \overline{\overline{B}C}}$

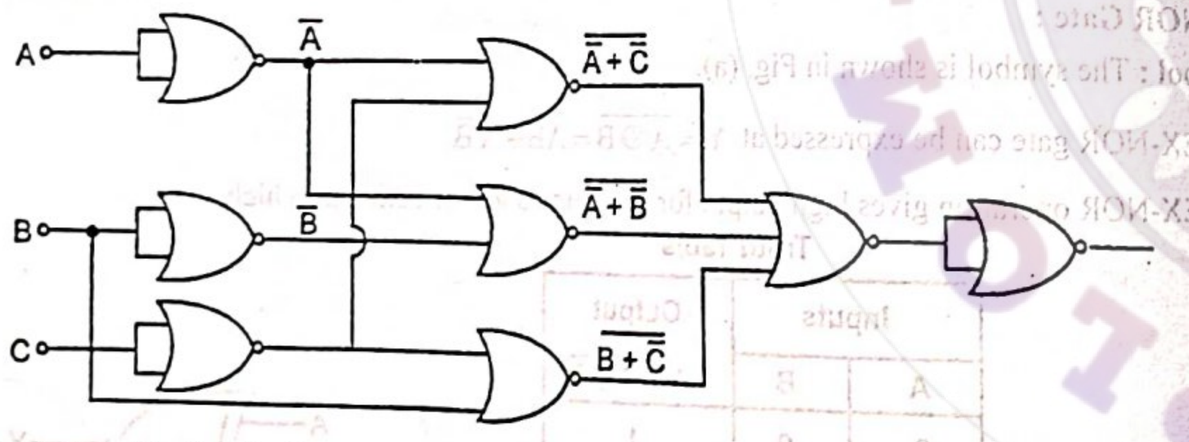
We can implement using NAND gates at this stage.



To implement using NOR gates :

$$Y = \overline{\overline{AC} \cdot \overline{AB} \cdot \overline{\overline{B}C}} = \overline{\overline{A} + \overline{C} \cdot \overline{A} + \overline{B} \cdot \overline{\overline{B}} + \overline{C}}$$

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



Example 2 : Implement the following Boolean expressions using NAND gates only :

(a) $Y = A + \overline{B}C + AC$ (b) $Y = \overline{(A + \overline{B}) \cdot (\overline{A} + C)}$

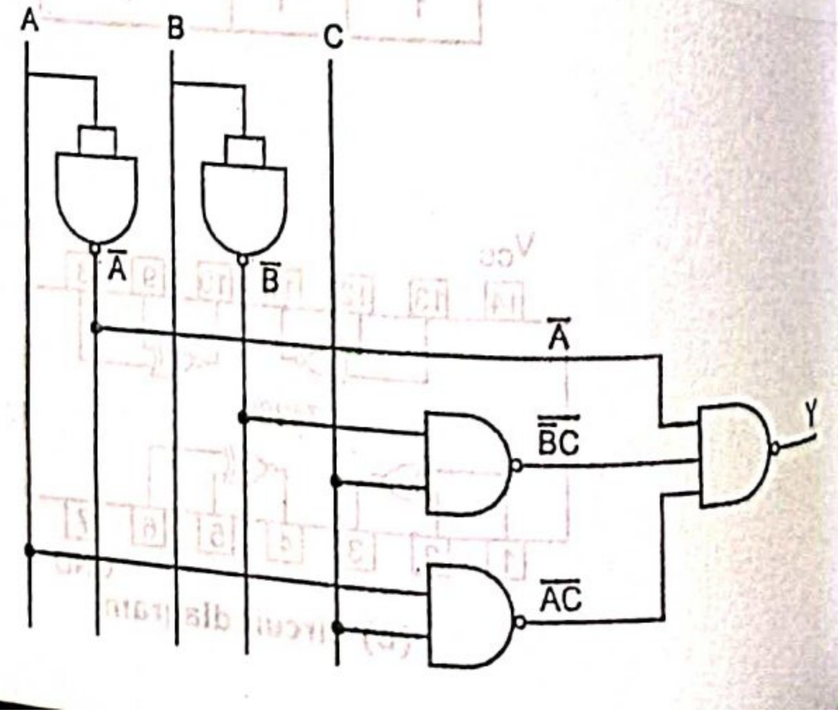
Solution :

(a) $Y = A + \overline{B}C + AC$

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

Applying De Morgan's theorem $Y = \overline{\overline{A} \cdot \overline{\overline{B}C} \cdot \overline{AC}}$

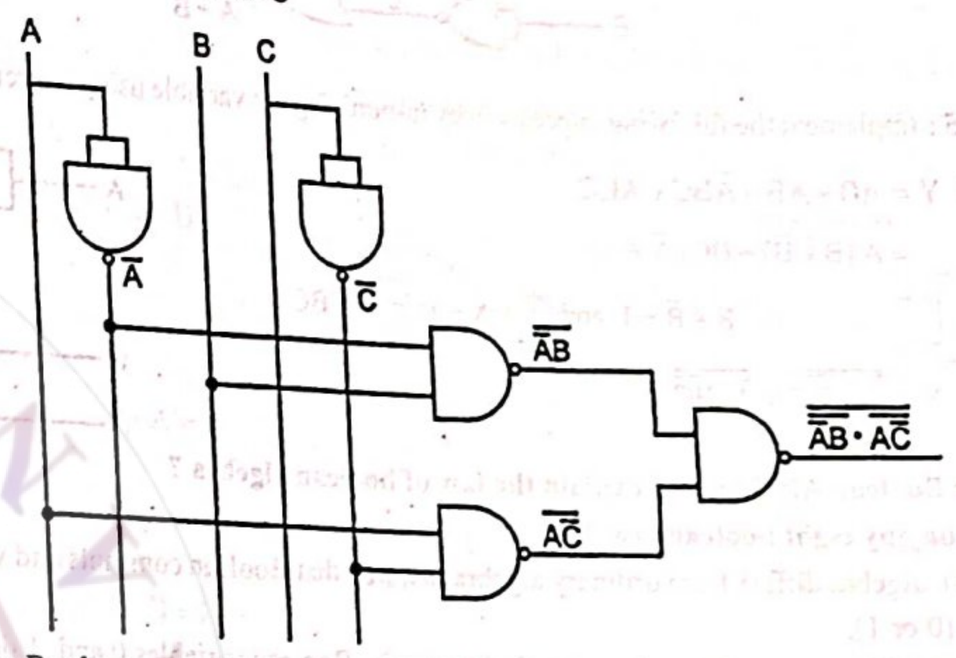
this can be implemented using NAND gates.



(b) $Y = \overline{(A + \overline{B}) \cdot (\overline{A} + C)}$ Using De Morgan's theorem

$= \overline{A + \overline{B}} + \overline{\overline{A} + C}$ Using De Morgan's theorem

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



Example 3 : Implement the following Boolean expression using NOR gates only :

$$Y = A + \overline{B}C + AC.$$

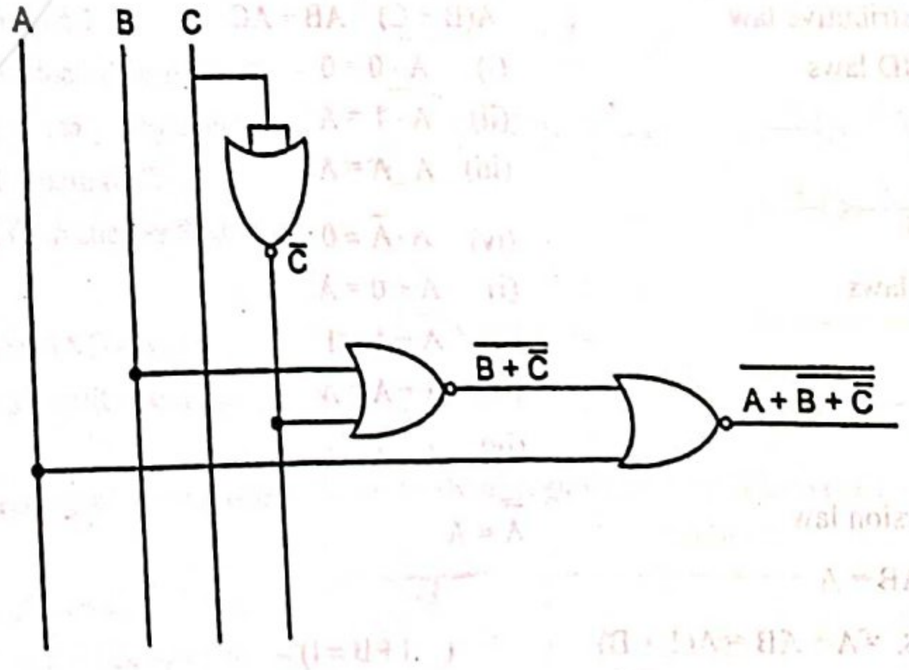
Solution : $Y = A + \overline{B}C + AC = A + AC + \overline{B}C$

$= A(1 + C) + \overline{B}C = \overline{\overline{A + \overline{B}C}} = \overline{A} \cdot \overline{\overline{B}C}$ (Using De Morgan's theorem)

$\overline{\overline{B}C} = \overline{\overline{B}} + \overline{C}$ (Using De Morgan's theorem)

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

Now, we can implement using only NORgates.

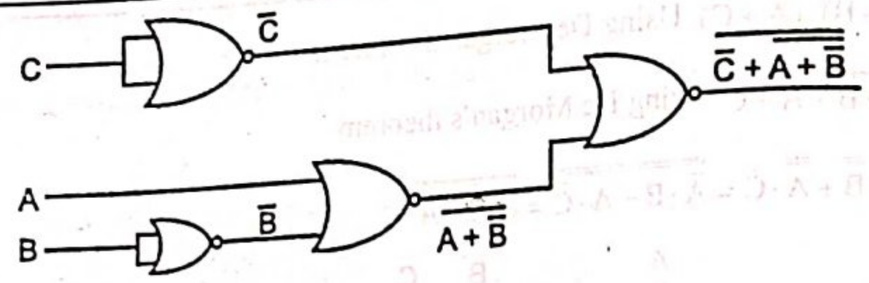


Example 4 : Implement the following Boolean expression using NOR gates only $Y = \overline{A}\overline{B}C + AC$.

Solution : $Y = \overline{A}\overline{B}C + AC = C(A + \overline{A}\overline{B})$

$= C(A + \overline{B})$ $(\because A + \overline{A}\overline{B} = A + \overline{B})$

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



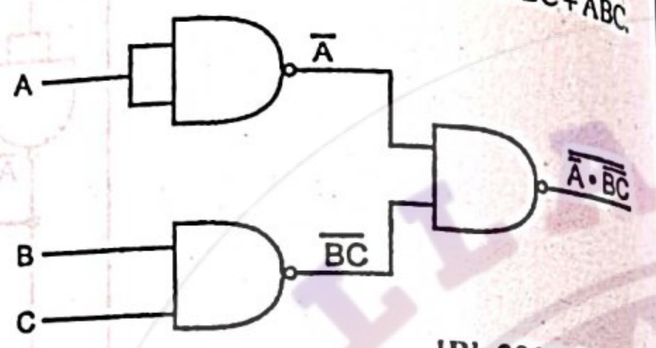
Example 5 : Implement the following expression by minimizing the variable using universal gates $Y = A\overline{B} + AB + \overline{A}BC + ABC$.

Solution : $Y = A\overline{B} + AB + \overline{A}BC + ABC$

$= A(B + \overline{B}) + BC(\overline{A} + A)$

$(\because B + \overline{B} = 1$ and $\overline{A} + A = 1) = A + BC$

$Y = \overline{\overline{A + BC}} = \overline{\overline{A} \cdot \overline{BC}}$



Q.5. Define Boolean Algebra and explain the law of boolean algebra ? |Bh.2014,16,18|

Or Mention any eight boolean law ?

Ans. Boolean algebra differs from ordinary algebra in a way that Boolean constants and variables are allowed to have only two values (0 or 1).

- These variables represent a voltage level of a circuit. Boolean variables 0 and 1 do not represent the actual number but they represent the state of the voltage variable.
- There are three basic operations in Boolean algebra :
  - (i) Logical addition (+) or OR operation.
  - (ii) Logical multiplication (·) or AND operation.
  - (iii) Logical inversion (¯) or NOT operation.

Laws of Boolean Algebra :

(a) Commutative law : $A + B = B + A$

$AB = BA$

(b) Associative law : $A + (B + C) = (A + B) + C$

$A(BC) = (AB)C$

(c) Distributive law : $A(B + C) = AB + AC$

(d) AND laws : (i) $A \cdot 0 = 0$

(ii) $A \cdot 1 = A$

(iii) $A \cdot A = A$

(iv) $A \cdot \overline{A} = 0$

(e) OR laws : (i) $A + 0 = A$

(ii) $A + 1 = 1$

(iii) $A + A = A$

(iv) $A + \overline{A} = 1$

(f) Inversion law : $\overline{\overline{A}} = A$

(g) $A + AB = A$

L.H.S. $= A + AB = A(1 + B)$ $(\because 1 + B = 1)$

$= A \cdot 1 = A$

$\therefore \quad A + AB = A$

(h) $A + \overline{A}B = A + B$

L.H.S. $= A + \overline{A}B$

i.e. $A(1) + \overline{A}B$

i.e. $A(1 + B) + \overline{A}B$ $(\because 1 + B = 1)$

$= A + AB + \overline{A}B$

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

$= A + B$

(i) $\overline{A} + AB = \overline{A} + B$

L.H.S. $= \overline{A} \cdot 1 + AB$

$= \overline{A}(1 + B) + AB$ $(\because 1 + B = 1)$

$= \overline{A} + \overline{A}B + AB$

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

$= \overline{A} + B = RHS$ $(\because \overline{A} + A = 1)$

(j) $\overline{A} + A\overline{B} = \overline{A} + \overline{B}$

L.H.S. $= \overline{A} \cdot 1 + A\overline{B}$

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

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

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

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

$= \overline{A} + \overline{B} = RHS$ $(\because \overline{A} + A = 1)$

(k) $(A + B)(A + C) = A + BC$

L.H.S. $= (A + B)(A + C) = A \cdot A + A \cdot C + B \cdot A + B \cdot C$

$= A + AC + AB + BC$

$= A(1 + C + B) + BC$

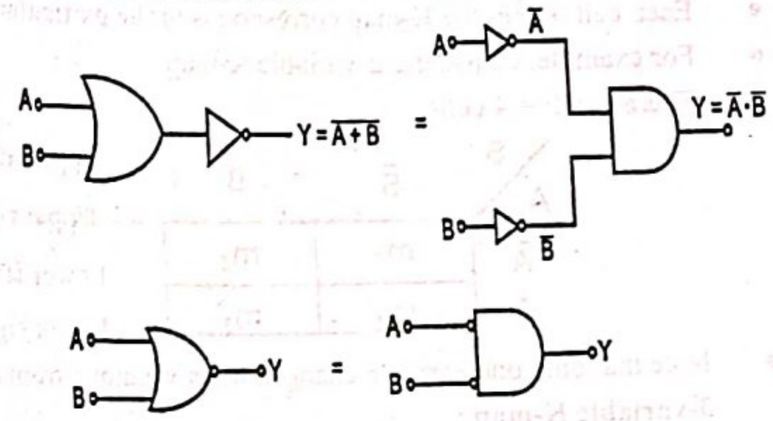
$= A + BC = RHS$ $(\because 1 + C + B = 1)$

- Therefore, $AB + AC = A(B + C)$. Duality theorem can be used for generating new expressions from the given Boolean expressions.

Q.2. State and prove De Morgan's first and second theorem for two variables ?

Ans. De Morgan's First Theorem : $\overline{A + B} = \overline{A} \cdot \overline{B}$.

- It states that the complement of sum is equal to the product of the complements.



Proof of De Morgan's First Theorem :

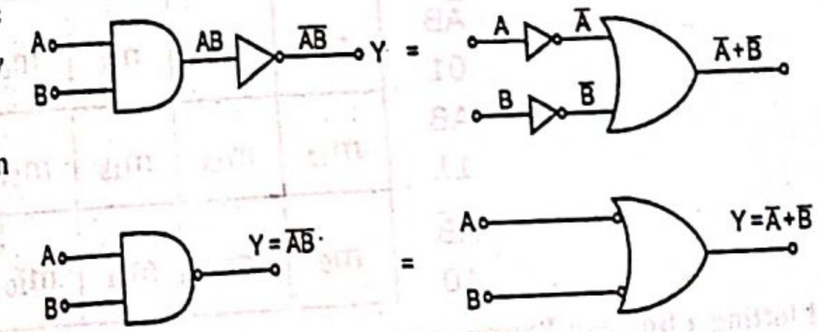
Table : 1

A	B	$\overline{A}$	$\overline{B}$	A + B	$\overline{A+B}$ (LHS)	$\overline{A} \cdot \overline{B}$ (RHS)
0	0	1	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	1	0	0

Hence L.H.S. = R.H.S.

De Morgan's Second Theorem : $\overline{AB} = \overline{A} + \overline{B}$

- It states that the complement of product is equal to the sum of the complements.



Q.1. Explain duality theorem with example ?

Ans. The duality theorem states that, "The dual of an algebraic expression can be obtained by replacing a binary 1 by a binary 0 and by interchanging the AND and OR operators".

- A Boolean relation can be derived from another Boolean relation by
  1. Changing an OR operation to an AND operation.
  2. Changing an AND operation to an OR operation.
  3. Complementing any 0 or 1 appearing in the expression.

For example:

(i) If $A + 0 = A$, then the dual relation is $A \cdot 1 = A$.

This is obtained by changing the OR operation to AND operation and complementing the 0 to 1.

(ii) Consider the expression $(A + B)(A + C) = A + BC$. Then by applying duality theorem, replacing OR operation (+) by AND operation (·) and AND operation (·) by OR operation (+).

Proof of De Morgan's Second Theorem :

Table : 2

A	B	$\overline{A}$	$\overline{B}$	A · B	$\overline{A \cdot B}$ (LHS)	$\overline{A} + \overline{B}$ (RHS)
0	0	1	1	0	1	1
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	0	0

Q.9. Describe Karnaugh Map with 3-variable K-map and 4-variable K-map ?

Ans. Karnaugh Map

- Karnaugh map is a graphical method used to simplify logic equations or to convert a truth tableto its corresponding logic circuit in a simple and systematic manner.
- The Karnaugh map or K-map is made up of squares. Each square represents one term. It is a systematic method for combining terms and obtaining the minimal expression.
- Each n variable map will consist of 2^n cells or squares. Thus a 3-variable map will have $2^3 = 8$ cells and a 4-variable map will have $2^4 = 16$ cells.
- Each cell within the K-map corresponds to the particular combination of the input variables.
- For example, consider a 2-variable K-map.

There are $2^2 = 4$ cells.

A \ B	$\bar{B}$	B
$\bar{A}$	m_0	m_1
A	m_2	m_3

Upper left cell corresponds to $\bar{A}\bar{B}$

Upper right cell corresponds to $\bar{A}\bar{B}$

Lower left cell corresponds to $\bar{A}\bar{B}$

Lower right cell corresponds to AB

- Note that only one variable changes when we move from one cell to adjacent cell in the K-map.

3-variable K-map :

A \ BC	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}_0$	m_0	m_1	m_3	m_2
A_1	m_4	m_5	m_7	m_6

OR

	$\bar{C}$	C
$\bar{A}\bar{B}$	0	1
$\bar{A}B$	2	3
AB	6	7
$A\bar{B}$	4	5

4-variable K-map :

AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	m_0	m_1	m_3	m_2
$\bar{A}B$ 01	m_4	m_5	m_7	m_6
AB 11	m_{12}	m_{13}	m_{15}	m_{14}
$A\bar{B}$ 10	m_8	m_9	m_{11}	m_{10}

OR

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

Plotting a Boolean Expression :

- To plot a Boolean expression on the K-map, perform the following steps :

(i) Transform the Boolean equation to a sum of product SOP expression.

(ii) Fill in the appropriate cells of the K-map by placing al in each cell corresponding to a term in the sum of product (SOP) expression.

Example : Consider the expression : $Y = \bar{A}(\bar{B}C + BC) + \bar{A}\bar{B}\bar{C}$

Solution : $Y \equiv \bar{A}\bar{B}\bar{C} + \bar{A}BC + \bar{A}\bar{B}\bar{C}$

Truth Table

A	B	C	Y	
0	0	0	1	→ $\bar{A}\bar{B}\bar{C} = m_0$
0	0	1	1	→ $\bar{A}\bar{B}C = m_1$
0	1	0	0	
0	1	1	1	→ $\bar{A}BC = m_3$
1	0	0	0	
1	0	1	0	
1	1	0	0	
1	1	1	0	

K-map representation

A \ BC	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	$B\bar{C}$ 11	BC 10
$\bar{A}_0$	1_0	1_1	1_3	0_2
$\bar{A}_1$	0_4	0_5	0_7	0_6

Example 1 : Plot the following truth table on K-map.

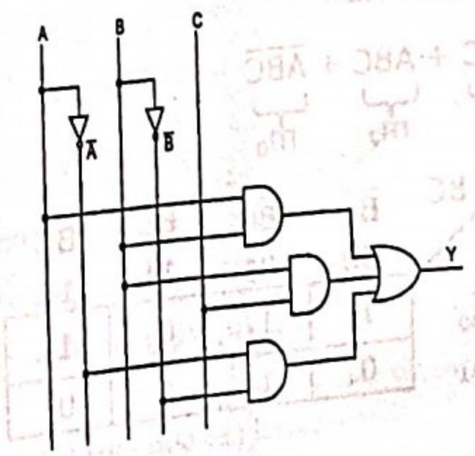
Truth Table

A	B	Y	
0	0	0	
0	1	0	
1	0	0	
1	1	1	→ AB

Solution : **K-map representation :**

A \ B	$\bar{B}_0$	$\bar{B}_1$
$\bar{A}_0$	0_0	0_1
$\bar{A}_1$	0_2	1_3

Example 2 : Represent the following truth table on K-map.



Truth Table

A	B	C	D	Y	
0	0	0	0	0	
0	0	0	1	1	$\bar{A}\bar{B}\bar{C}D = m_1$
0	0	1	0	0	
0	0	1	1	1	$\bar{A}\bar{B}CD = m_3$
0	1	0	0	0	
0	1	0	1	0	
0	1	1	0	1	$\bar{A}BC\bar{D} = m_6$
0	1	1	1	0	
1	0	0	0	0	
1	0	0	1	1	$A\bar{B}\bar{C}D = m_9$
1	0	1	0	0	
1	0	1	1	1	$A\bar{B}CD = m_{11}$
1	1	0	0	0	
1	1	0	1	0	
1	1	1	0	0	
1	1	1	1	1	$ABCD = m_{15}$

Solution : **K-map representation**

AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	0_0	1_1	1_3	0_2
$\bar{A}B$ 01	0_4	0_5	0_7	1_6
AB 11	0_{12}	0_{13}	1_{15}	0_{14}
$A\bar{B}$ 10	0_8	1_9	1_{11}	0_{10}

Example 3 : Represent the given equation on K-map

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

Solution :

$$Y = \underbrace{\bar{A}B\bar{C}}_{m_2} + \underbrace{\bar{A}\bar{B}C}_{m_1} + \underbrace{A\bar{B}C}_{m_5} + \underbrace{ABC}_{m_7} + \underbrace{\bar{A}\bar{B}\bar{C}}_{m_0}$$

A \ BC	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}_0$	1_0	1_1	0_3	1_2
A_1	0_4	1_5	1_7	0_6

Example 4 : Plot the given Boolean expression on K-map.

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

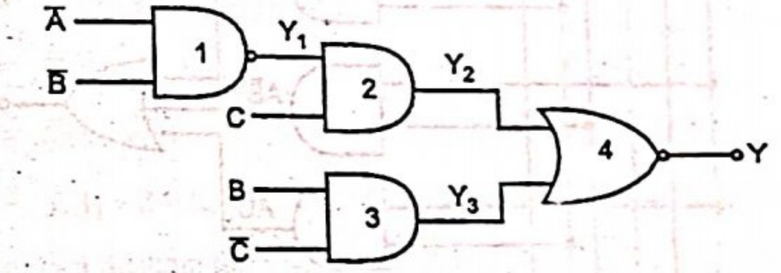
$$Y = \underbrace{\bar{A}\bar{B}C\bar{D}}_{m_2} + \underbrace{\bar{A}B\bar{C}D}_{m_5} + \underbrace{A\bar{B}\bar{C}D}_{m_9} + \underbrace{A\bar{B}C\bar{D}}_{m_{10}}$$

AB \ CD	$\bar{C}\bar{D}$ 00	$\bar{C}D$ 01	CD 11	$C\bar{D}$ 10
$\bar{A}\bar{B}$ 00	0_0	0_1	0_3	1_2
$\bar{A}B$ 01	0_4	1_5	0_7	0_6
AB 11	0_{12}	0_{13}	0_{15}	0_{14}
$A\bar{B}$ 10	0_8	1_9	0_{11}	1_{10}

Hence L.H.S. = R.H.S.

SOLVED PROBLEMS

Example 1 : For the given figure, derive the Boolean expression.



Solution : Output of NAND Gate 1, $Y_1 = \overline{\bar{A}\cdot\bar{B}} = \bar{\bar{A}} + \bar{\bar{B}} = A + B$

Output of AND Gate 2, $Y_2 = (A+B)\cdot C$

Output of AND Gate 3, $Y_3 = B\cdot\bar{C}$

Output of NOR Gate 4, $Y = \overline{Y_2 + Y_3}$

$$Y = \overline{(A+B)\cdot C + B\cdot\bar{C}} = \overline{AC + BC + B\cdot\bar{C}} = \overline{AC + B(C+\bar{C})} = \overline{AC + B} = \overline{AC}\cdot\bar{B}$$

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

Example 2 : For the given function $f = x\bar{y} + \bar{x}y$, find the complement of f.

Solution :

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

$$\therefore \quad \bar{f} = \overline{x\bar{y} + \bar{x}y}$$

Using De Morgan's theorem

$$= \overline{x\bar{y}}\cdot\overline{\bar{x}y}$$

Further applying De Morgan's theorem

$$(\bar{\bar{x}} + \bar{y})\cdot(\bar{\bar{x}} + \bar{\bar{y}})$$

$$= (x + \bar{y})\cdot(x + y) = x\bar{x} + \bar{x}\bar{y} + xy + y\bar{y} \qquad (\because x\bar{x} = 0,\ y\bar{y} = 0)$$

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

Example 3 : Simplify the following expressions with Boolean laws :

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

Solution :

(i)

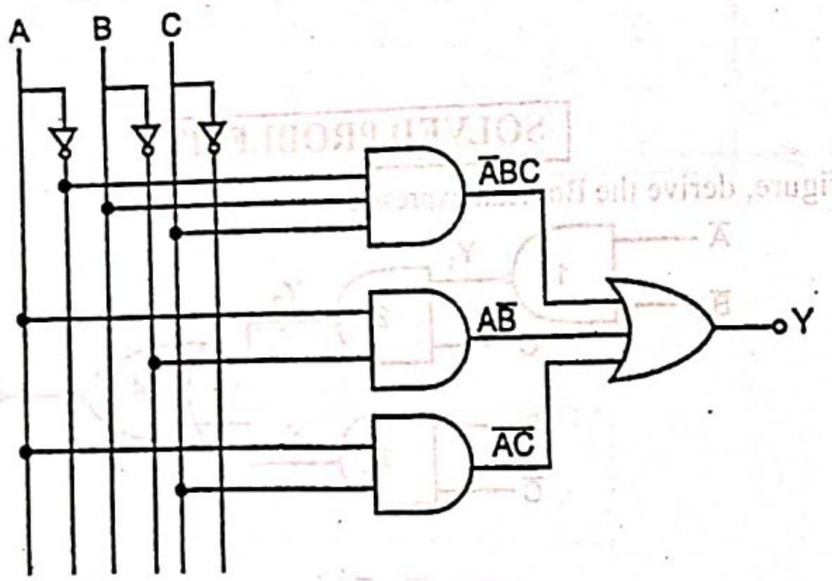
$$Y = (A+C)(A+D)(B+C)(B+D)$$
$$Y = (AA+AD+CA+CD)(BB+BD+CB+CD)$$
$$= (A+AD+AC+CD)(B+BD+BC+CD)$$
$$= A[(1+D+C)+CD]+B[(1+D+C)+CD] \quad (\because 1+C+D=1)$$
$$= (A+CD)(B+CD)$$
$$= AB+ACD+BCD+CD \quad (\because A+B+1=1)$$
$$= AB+CD(A+B+1)$$
$$= AB+CD$$

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

$$B+BC = B(1+C) = B$$
$$= B(B+\bar{B}C)(B+D)$$
$$= (BB+B\bar{B}C)B+D \quad (\because B\bar{B}=0)$$
$$= B(B+D) = BB+BD = B+BD = B(1+D) = B$$

Example 4 : Draw the logical circuit diagram for $Y = \bar{A}BC + A\bar{B} + \overline{AC}$.

Solution :



Example 5 : Prove that $\bar{A}\bar{B}C + \bar{A}BC + A\bar{B} = \bar{A}C + A\bar{B}$.

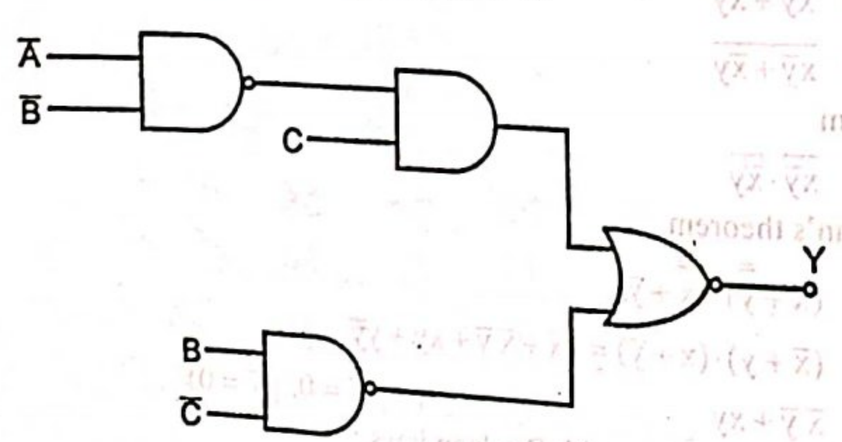
Solution : L.H.S. $= \bar{A}\bar{B}C + \bar{A}BC + A\bar{B} = \bar{A}C(\bar{B}+B) + A\bar{B} = \bar{A}C + A\bar{B} =$ R.H.S.

Example 6 : Simplify the given equation using Boolean laws. $Y = \overline{\overline{A+B} + \overline{\bar{C}(A+B)}}$

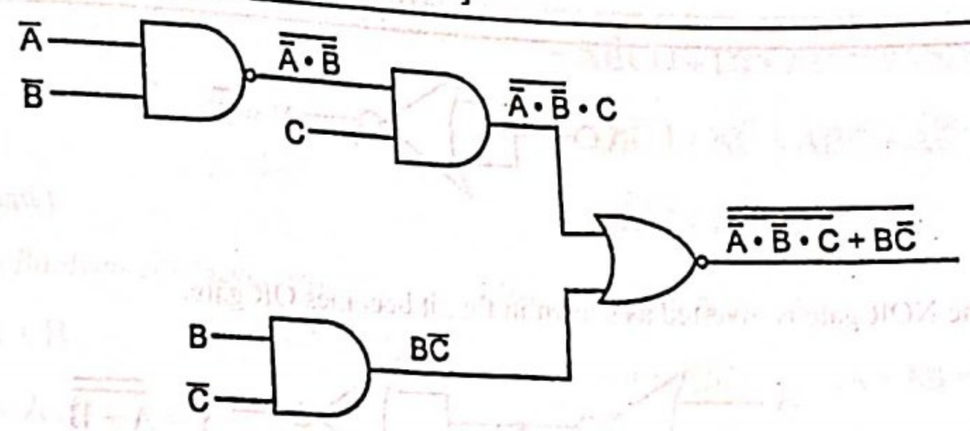
Solution : Applying De Morgan's theorem

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

Example 7 : For the given figure, derive the Boolean expression for Y.



Solution :



$$Y = \overline{\overline{\bar{A}\cdot\bar{B}}\cdot C + B\bar{C}} = \overline{(\bar{\bar{A}}+\bar{\bar{B}})\cdot C + B\bar{C}} = \overline{(A+B)\cdot C + B\bar{C}}$$
$$Y = \overline{AC+BC+B\bar{C}} = \overline{AC+B(C+\bar{C})} = \overline{AC+B}$$
$$Y = \overline{AC}\cdot\bar{B}$$
$$Y = (\bar{A}+\bar{C})\bar{B}$$
$$Y = \bar{A}\bar{B}+\bar{B}\bar{C}$$

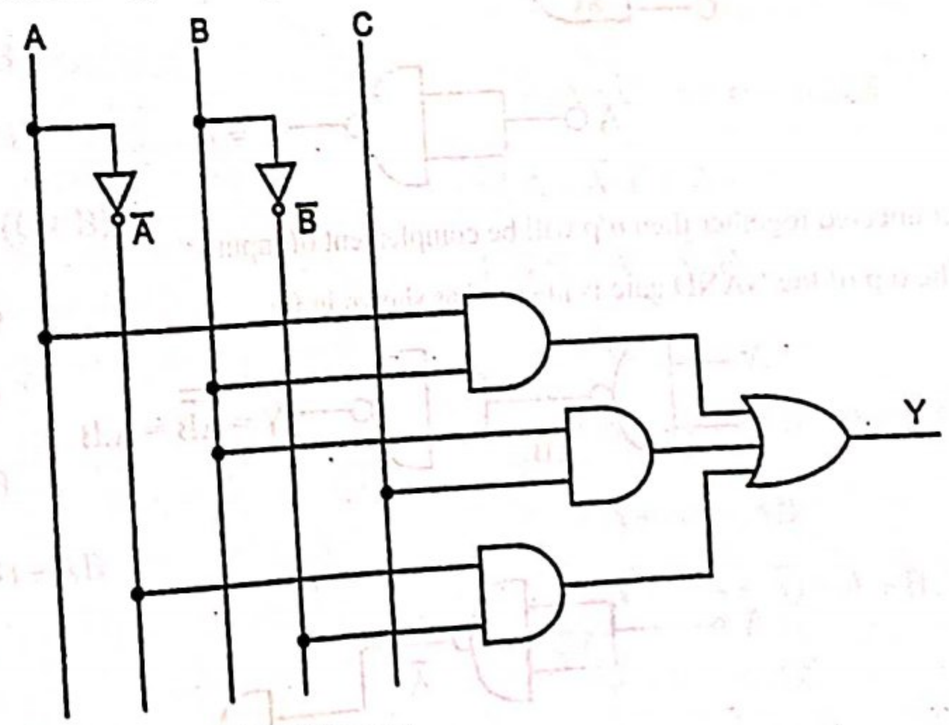
Example 8 : Prove that $\bar{Y}\bar{Z} + \bar{W}\bar{X}\bar{Z} + \bar{W}XY\bar{Z} + WY\bar{Z} = \bar{Z}$.

Solution :

$$\text{L.H.S.} = \bar{Y}\bar{Z} + \bar{W}\bar{X}\bar{Z} + \bar{W}XY\bar{Z} + WY\bar{Z} = \bar{Z}[Y + \bar{W}\bar{X} + \bar{W}XY + WY]$$
$$= \bar{Z}[\bar{Y} + \bar{W}(\bar{X} + XY + WY] \quad (\because \bar{X} + XY = \bar{X} + Y)$$
$$= \bar{Z}[\bar{Y} + \bar{W}(\bar{X}+Y) + WY] = \bar{Z}[\bar{Y} + \bar{W}\bar{X} + \bar{W}Y + WY]$$
$$= \bar{Z}[\bar{Y} + Y(\bar{W}+W) + \bar{W}\bar{X}]$$
$$= \bar{Z}[\bar{Y} + Y + \bar{W}\bar{X}] \quad (\because \bar{Y} + Y = 1)$$
$$= \bar{Z}[1 + \bar{W}\bar{X}] \quad (\because \bar{Y} + Y = 1)$$
$$= \bar{Z}[1] = \bar{Z} = \text{R.H.S.} \quad (\because 1 + \bar{W}\bar{X} = 1)$$

Example 9 : Draw the logical circuit diagram for $Y = \bar{A}BC + A\bar{B} + \overline{AC}$.

Solution :

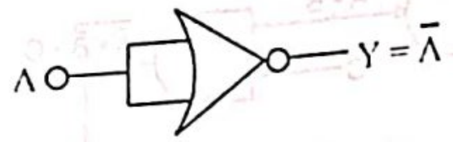


Q *Obtain the following operations using only NOR gates.* **[Bh.2010Even,2016**

(a) NOT (b) OR (c) AND.

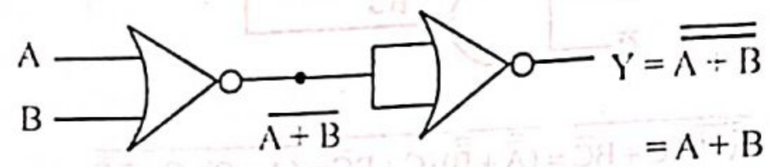
Ans. We assume here 2-input NOR gates:

(a) NOT gate:



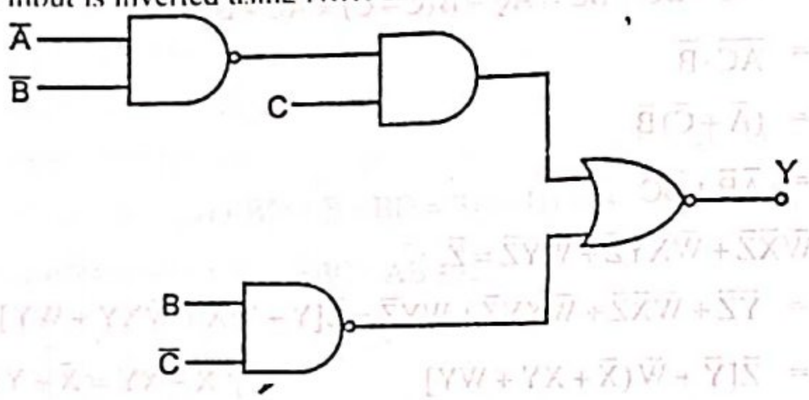
(b) OR gate:

If the output of the NOR gate is inverted as shown in fig.. it becomes OR gate.



$= A + B$

(c) AND gate: If individual input is inverted using NOR gate and then input to another NOR gate its o/p behaves as AND gate operation.



$y = \overline{A} + \overline{B}$

$= \overline{\overline{A}.\overline{B}} = A.B$

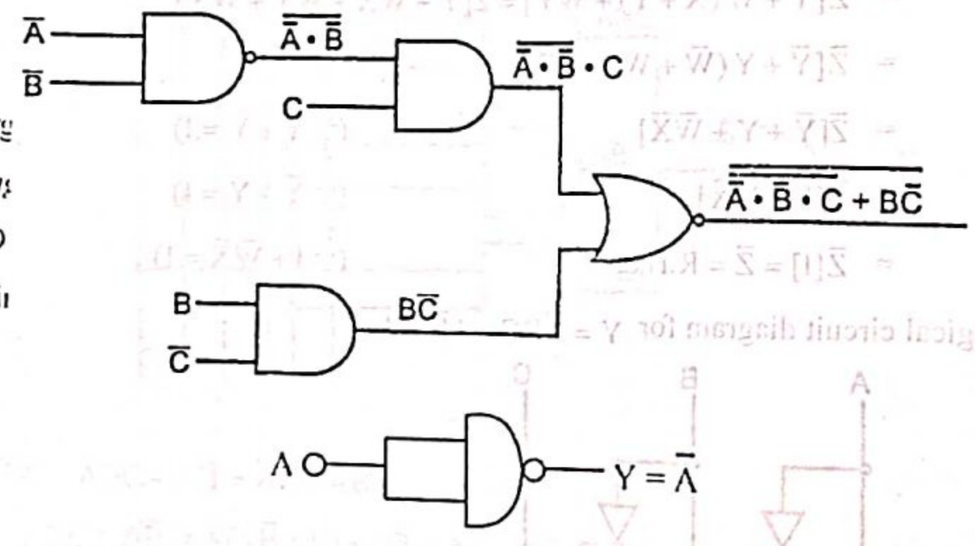
According to De'morg

Q.8 *Obtained the following*

(a) NOT (b) AND

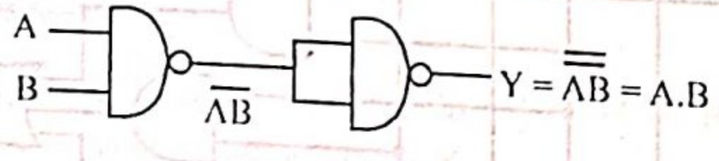
Ans. Let us assume 2-i

(a) NOT gate:

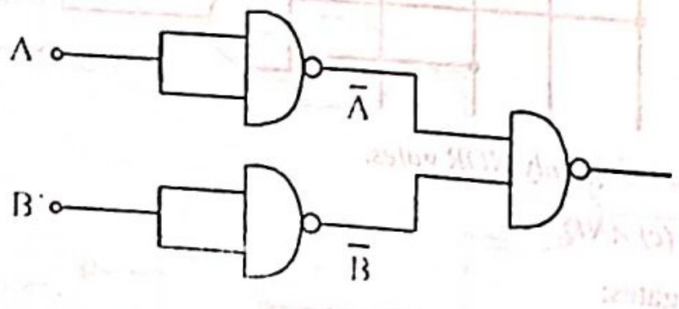


If both inputs are connected together then o/p will be complement of input.

(b) AND gate: If the o/p of the NAND gate is inverted as shown in fig.



(c) OR gate:



$y = \overline{\overline{A}.\overline{B}}$

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

$= A + B.$

(Using De'morgan's law)

Q.9 *Prove the following using Boolean algebraic theorems.*

(a) $A + \overline{A}.B + A.\overline{B} = A + B$ **[Bh.2010 odd]**

Ans. L.H.S. $= A + \overline{A}.B + A.\overline{B}$

$= A(1 + \overline{B}) + \overline{A}B = A + \overline{A}B$ $[\because 1 + \overline{B} = 1]$

$= (A + \overline{B})(A + B) = A + B = R.H.S.$

$[A + \overline{A} = 1]$

(b) $A.B + \overline{A}.B + \overline{A}.\overline{B} = \overline{A} + B$

L.H.S. $= A.B + \overline{A}.B + \overline{A}.\overline{B}$

$= B(A + \overline{A}) + \overline{A}\,\overline{B} = B + \overline{A}\,\overline{B}$

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

$= B + \overline{A} = \overline{A} + B = R.H.S.$

(c)

$\overline{A}BC + A\overline{B}C + AB\overline{C} + ABC = AB + BC + CA$

L.H.S. $= \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$

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

$= \overline{A}BC + A\overline{B}C + AB$

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

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

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

$= \overline{A}BC + AB + AC$

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

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

$= C(A + B) + AB$

$= AC + BC + AB = R.H.S$

Q.10 *Simplify the logical expression:*

Ans. **(a)** $A\overline{B}\overline{C} + A\overline{B}\overline{C}D + A\overline{C}$

$= A\overline{B}\overline{C}(1 + D) + A\overline{C}$

$= A\overline{B}\overline{C}\,1 + A\overline{C} = A\overline{B}\overline{C} + A\overline{C}$

$= A\overline{C}(\overline{B} + 1) = A\overline{C}\,1 = A\overline{C}$

(b) $A + AB + A\overline{B}C$

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

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

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

$= A.A + A\overline{B}C = A(1 + \overline{B}C)$

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

$= A(1.1) = A.1 = A$

Q.11 *Prove that:*

Ans. **(a)** $A + BC = (A + B)(A + C)$

L.H.S. $= A + BC$

$= A\,1 + BC = A(1 + B) + BC$

$= A + AB + BC = A(1 + C) + AB + BC$

$= A + AC + AB + BC$

$= AA + AC + AB + BC$

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

$= (A + C)(A + B) = R.H.S.$

(b) $A + \overline{A}.B = A + B$

L.H.S. $= A + \overline{A}.B$

$= A.1 + \overline{A}\,B$

$= A(1 + B) + \overline{A}B = A.1 + AB + \overline{A}B$

$= A + AB + \overline{A}B$

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

$= A + B = R.H.S.$

Q.12 *Simplify the following Boolean functions to a minimum no. of literals.* **[2009(A)]**

(a) $x + x'y$

(b) $x'y'z + x'yz + xy'$

(c) $xy + x'z + yz$.

Ans. (a) $x + x'y = x(y + 1) + x'y$

$= xy + x + x'y = y(x + x') + x$

$= y \cdot 1 + x = x + y.$

(b) $x'y'z + x'yz + xy'$

$= x'z(y'+y) + xy' = x'z.1 + xy'$

$= x'z + xy'$

(c) $xy + x'z + yz$

Q.13 *What do you mean by POS and SOP forms of Boolean algebra.*

Ans. Logical functions are expresed in terms of logical variables. The values taken by logical variables are in binary form. Any logical function may be expressed in the following two forms:-

(i) Sum of products form (SOP)

(ii) Product of sums form (POS)

1. ***Sum of products forms:-***A sum of products expression consists of product terms logical added. A products term is a logical product of many variables. The variables may or may not be complemented. Examples of SOP are

(i) $AB + \overline{A}B + A\overline{B}$

(ii) $A + BC + A\overline{C}$

(iii) $ABC + \overline{A}B + A\overline{B}C + A$

2. *Product of sums form:-* (POS)

POS expression consists of sum terms logically multiplied. A sum term is the logical addition of several variables. The variables may or may not be complemental exp. of POS are -

(i) $(A + B)(\overline{A} + B)$

(ii) $(A + \overline{B})(A + B + C)(B + C)$

(iii) $(A + \overline{C})(\overline{A} + B + C)(A + \overline{B} + C)$

Q.14 *What is canonical form of a logic expression? Define minterms and maxterms.*

Ans. When each term of a logic expression contains all variables, then it is called a canonical or standard form of a expression. When a SOP form of logic expression is in canonical form. each product term is called minterm and each minterm contains all variables. Canonical form of a sum of product is also known as standard sum of products exp. are

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

(ii) $y = A\overline{B}C + \overline{A}BC + \overline{A}\,\overline{B}C + ABC$

(iii) $y = A\overline{B}CD + AB\overline{C}D + \overline{A}BC\overline{D} + ABCD$

Minterm

Similarly, when a product of sums form of logic expression is in canonical form, each sum term is called a maxterm and each maxterm contains all variables. The canonical POS expression is also called maxterm canonical form or standard products of sums. Exp. are

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

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

Maxterm

Q.15 *Convert the given expression into canonical form:*

(a) $(A + B)(B + C)$ *(b)* $A(A + B)$

Ans. Any POS form can be converted into standard POS by ORing the terms in expression with terms. Formed by ANDing the variables and its complement which are not present in that term

(a) (A + B) (B + C)

$= (A + B + C\overline{C})(A\overline{A} + B + C) \qquad [A\overline{A} = 0]$

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

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

Maxterm

(b) $A(A + B) = (A + B\overline{B})(A + B)$

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

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

$[\because (A + B)(A + B) = (A + B)]$

Q.16 *Write the minterms designation for*

(i) ABCD *(ii)* $A\overline{B}CD$ *(iii)* $A\overline{B}\,\overline{C}\,\overline{D}$

Ans. For minterm designation 0 is written for a letter with bar and a 1 for a letter without a bar. This binary number is expressed as a decimal number and is written as a subscript of m.

(i) A B C D

1 1 1 1 = 15 (decimal equivalent)

$ABCD = m_{15}$.

(ii) $A\overline{B}CD$

1 0 1 1 = 11 (decimal equivalent)

$\therefore \quad A\overline{B}CD = m_{11}$

(iii) $A\,\overline{B}\,\overline{C}\,\overline{D}$

1 0 0 0 = 8 (decimal equivalent)

$\therefore \quad A\,\overline{B}\,\overline{C}\,\overline{D} = m_8$.

Q.17 *Write the maxterm designation for*

(i) $A + \overline{B} + \overline{C}$ (ii) $A + B + C$

Ans. Maxterm designation is similar to minterms designation except the difference that unbarred letters represent as 0 and the barred letters represent as 1.

(i) $A + \overline{B} + \overline{C}$

0 1 1 → 3 (decimal equivalent)

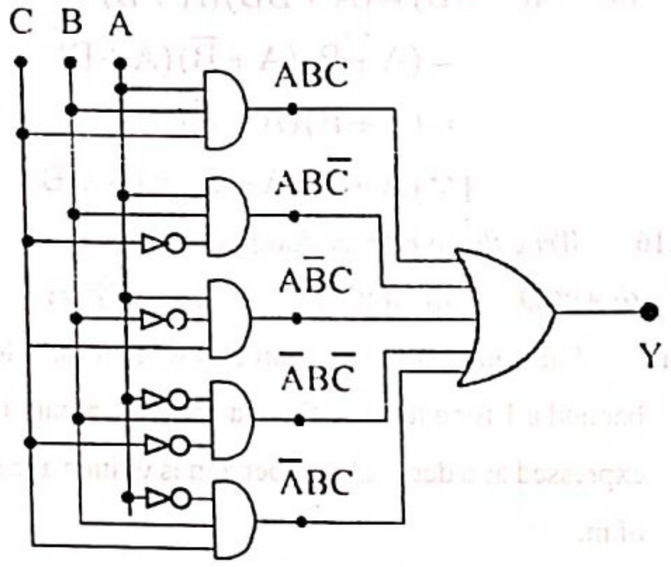
$\therefore \quad A + \overline{B} + \overline{C} = m_3$.

(ii) $A + B + C$

0 0 0 = 0 (decimal equivalent)

$\therefore \quad A + B + C + m_0$.

Q.18 *Find the complete expression for maxterms designation* $y = \pi M(0,2,4,5)$.



(ii) $y = (\overline{A} + B)C + ABC$

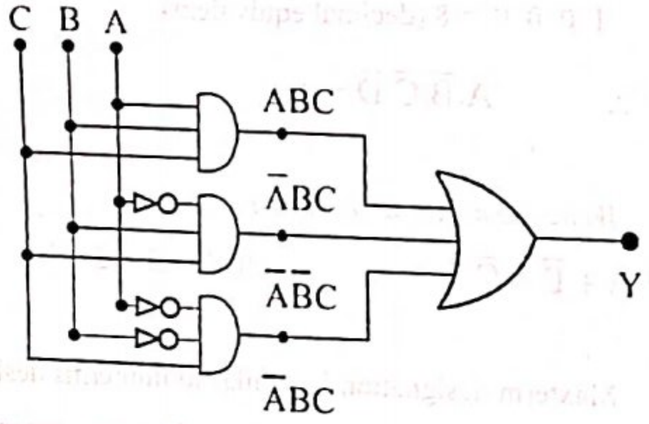
$= \overline{A}C + BC + ABC$

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

$= \overline{A}BC + \overline{A}\,\overline{B}C + ABC + \overline{A}BC + ABC$

$= ABC + \overline{A}BC + \overline{A}\,\overline{B}C$

$[\because ABC + ABC = ABC, \overline{A}BC + \overline{A}BC = \overline{A}BC]$



Q.2 *Design the circuit with only one type of gates (NAND or NOR) for the expression*

$y = (A + B + C)(B + \overline{C}A)$ **[Bh.2010old]**

Ans. (i) Using sum of products form

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

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

$= AB + A\overline{C}A + BCB + BC\overline{C}A$

$= AB + A\overline{C} + BC + O$

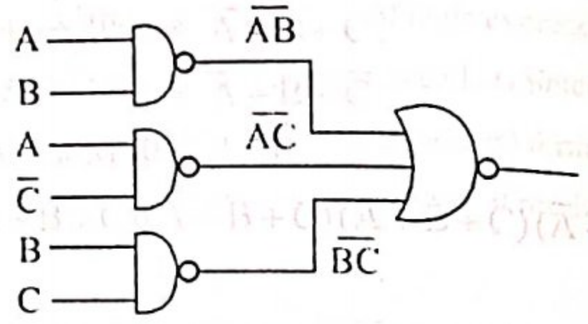
$[C\overline{C} = 0]$

Using De-morgan's theorem

$\overline{y} = \overline{AB + AC + BC}$

$= \overline{AB}.\overline{AC}.\overline{BC}$

We can realize this expression using NAND gate only



$y = \overline{AB}.\overline{AC}.\overline{BC}$

$= \overline{\overline{AB} + \overline{AC} + \overline{BC}}$

$= AB + A\overline{C} + BC$

(ii) Using product of sum form **[2010 (old)]**

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

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

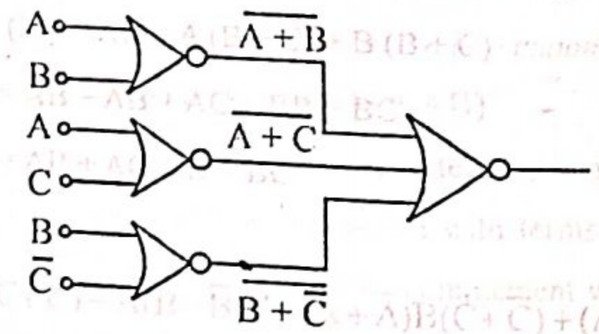
$= (A + B)(A + C)(B + \overline{C})$

using De-morgan's theorem

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

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

We can realize this expression using NOR gate only.



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

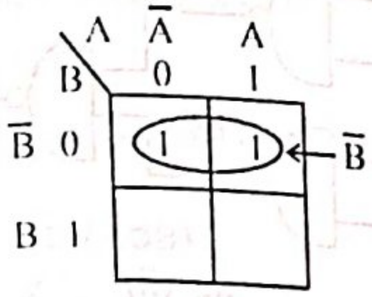
$= \overline{\overline{(A + B)}}.\overline{\overline{A + C}}.\overline{\overline{B + \overline{C}}}$

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

Q.3 *Simplify with the help of karnaugh map*

$y = \overline{A}\,\overline{B} + A\overline{B}$

Ans.



At first we put 1 for each corresponding minterm.

Here first minterm is $\overline{A}\,\overline{B}$

and second minterm is $A\overline{B}$

After that the two adjacent squares showing 1 are grouped together. For simplification it observed that which variable is common to both squares.

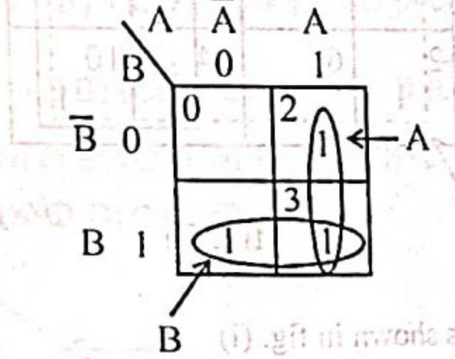
Here $\overline{B}$ is common to both squares. Hence result is $y = \overline{B}$.

Q.4 *Simplify using k-map*

$y = \overline{A}B + AB + A\overline{B}$

Ans. Here variable is 2.

Therefore no. of square will be 4.



$y = \overline{A}B + AB + A\overline{B}$

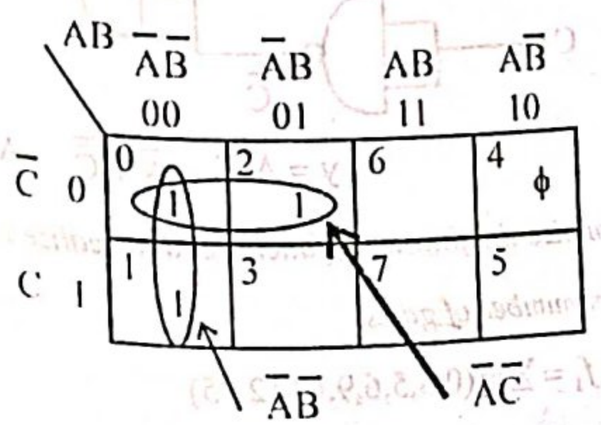
$= \overline{A}B + A(B + \overline{B}) = \overline{A}B + A$

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

Q.5 *Simplify the expression*

$y = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + \overline{A}\,\overline{B}C$ *using k-map method.*

Ans.



$y = \overline{A}\,\overline{B} + \overline{A}\,\overline{C}$.

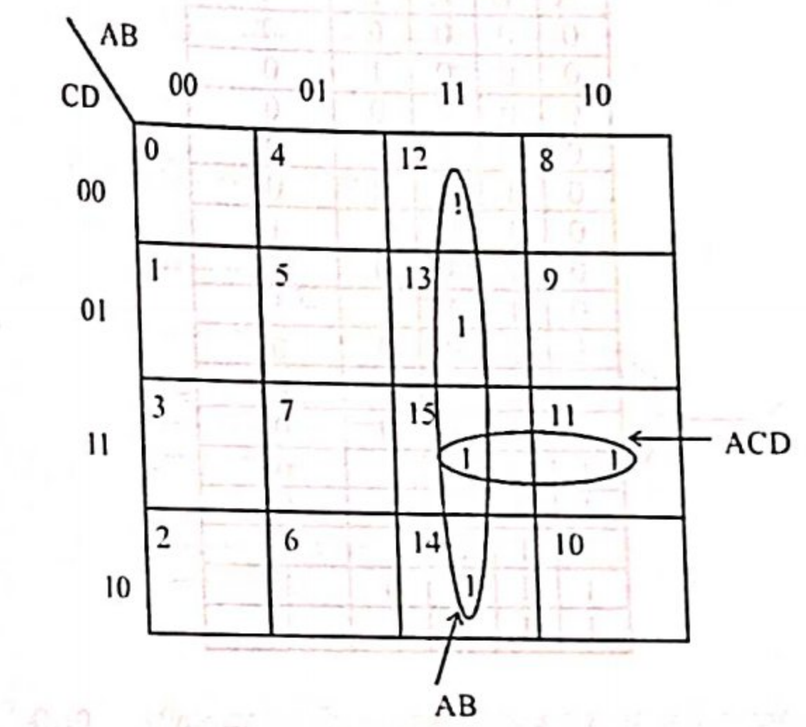
Q.6 *Use karnaugh map to simplify*

$y = ABC\overline{D} + A\overline{B}CD + ABC\overline{D} + AB\overline{C}D + AB\overline{C}\,\overline{D}$

[2004(A)]

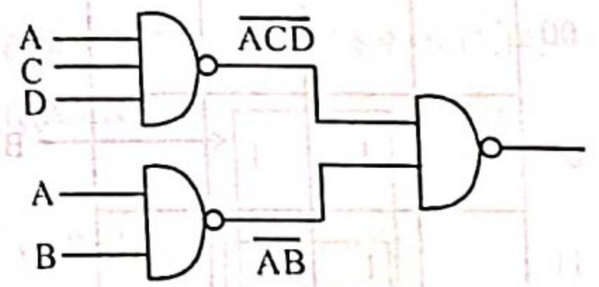
Realize the simplified output with the help of standard logic gates.

Ans.



$y = ACD + AB$

Realization of this expression using NAND gates



$y = \overline{\overline{ACD}.\overline{AB}}$

$= \overline{\overline{ACD}} + \overline{\overline{AB}}$

$= ACD + AB$

Q.7 *Given the logic equation*

$y = ABC + B\overline{C}D + \overline{A}BC$

(a) make a truth table

(b) simplify using k-map

(c) realize y using NAND gate only.

Ans. (a) The truth table is given in table (i)

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

(b) The k-map is shown in fig. (i)

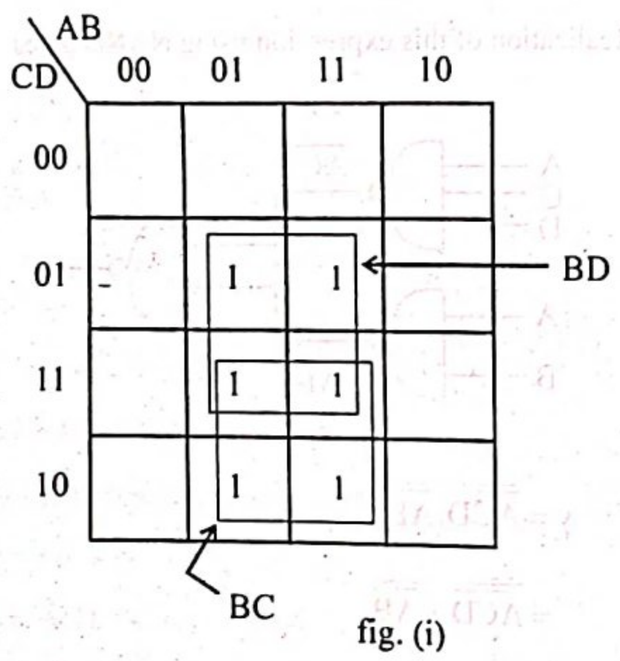


fig. (i)

$Y = BC + BD$

(c) The NAND - NAND realization is given in fig. (ii).

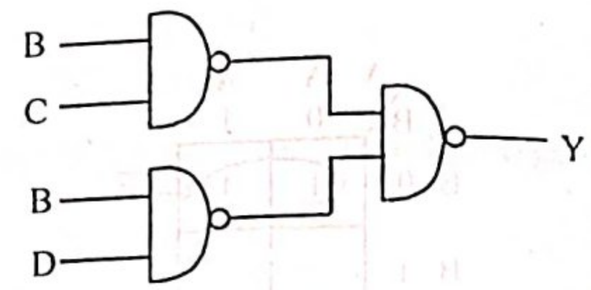


fig. (ii)

Q.8 (a) *Make a k-map for the function*

$$y = AB + A\overline{C} + C + AD + A\overline{B}C + ABC$$

(b) *Express y in standard SOP form*

(c) *Minimize it and realize the minimized expression using NAND gates only.*

Ans.

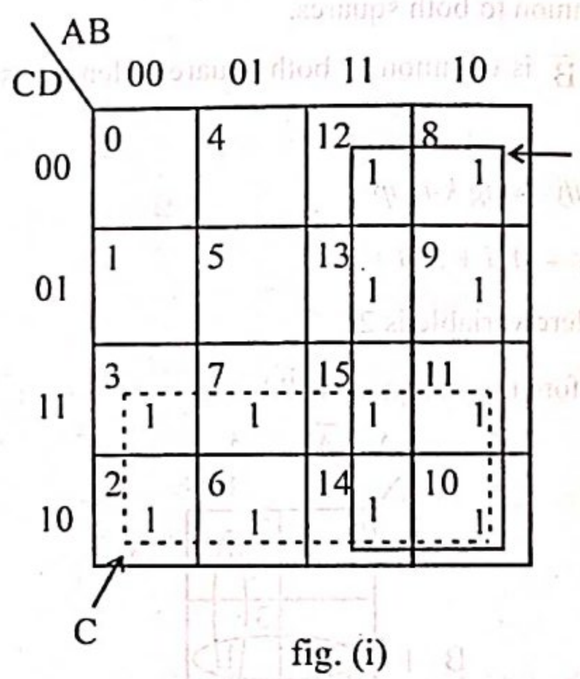


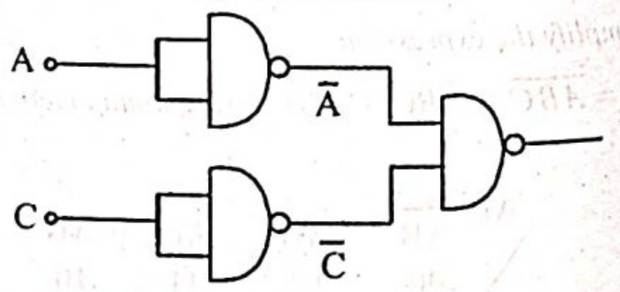
fig. (i)

(a) k-map is shown in fig. (i)

(b) $y = \sum m\,(2, 3, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)$

(c) $y = A + C$

Realization using NAND gate is shown in fig.



$$y = \overline{\overline{A}.\overline{C}} = \overline{\overline{A}} + \overline{\overline{C}} = A + C.$$

Q.9 *Minimize the following functions and realize using minimum number of gates.*

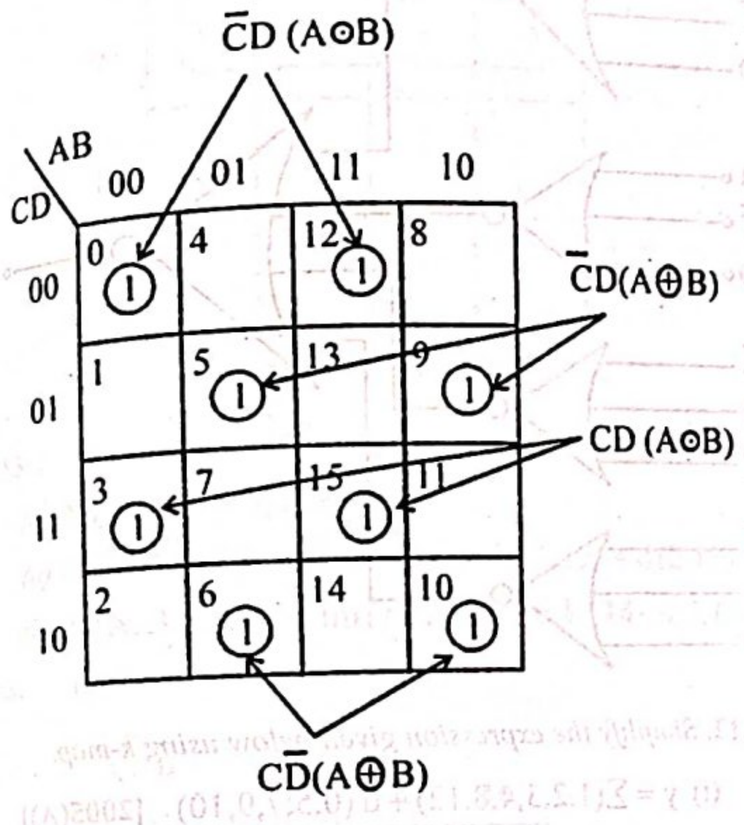
(a) $f_1 = \sum m(0,3,5,6,9,10,12,15)$

(b) $f_2 = \sum m(0,1,2,3,11,12,14,15)$

Ans. (a) $f_1 = \sum m(0,3,5,6,9,10,12,15)$

k-map is shown in fig.

using offset adjacencies shown in fig.

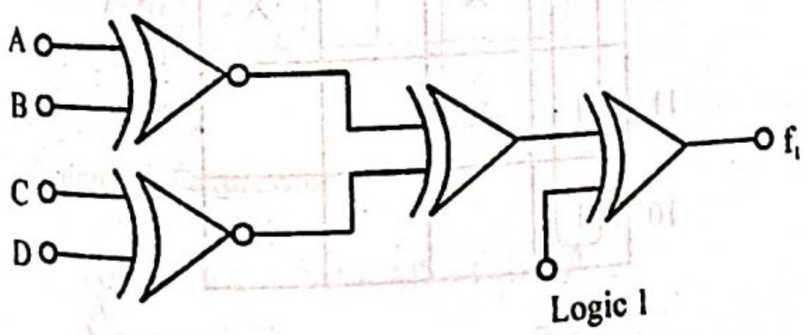


$f_1 = \overline{C}\overline{D}\,(A \odot B) + CD\,(A \odot B) + \overline{C}D\,(A \oplus B)\,C\overline{D}\,A \oplus B$

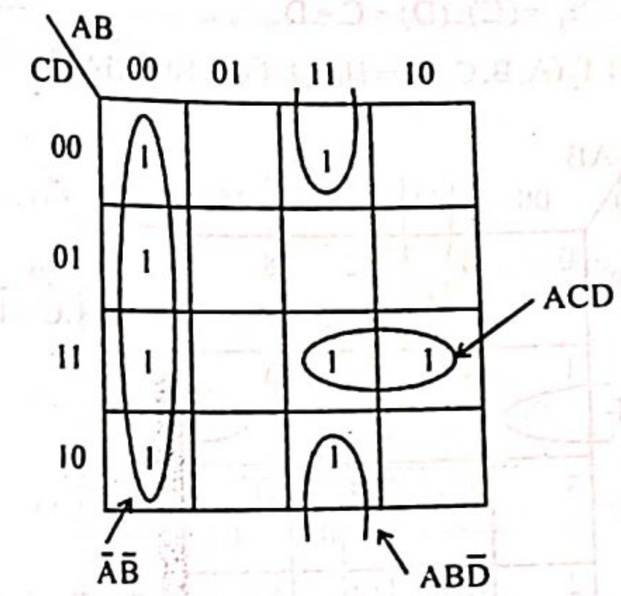
$= (A \odot B)(\overline{C}\overline{D} + CD) + A \oplus B\,(\overline{C}D + C\overline{D})$

$= (A \odot B)(C \odot D) + (A \oplus B)(C \oplus D)$

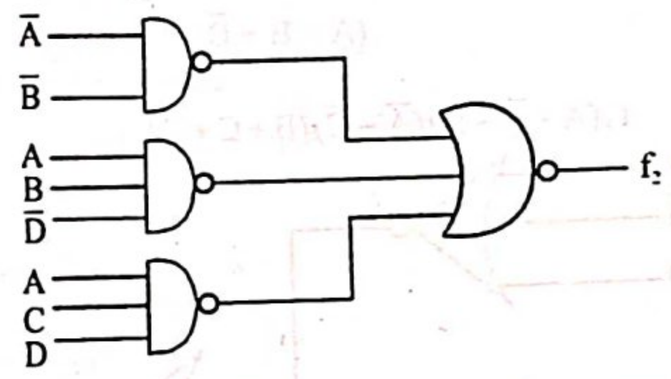
$= (A \oplus B) \odot (C \oplus D)$



(b) $f_2 = \sum m(0,1,2,3,11,12,14,15)$



$f_2 = \overline{A}\,\overline{B} + AB\overline{D} + ACD$



Q.10 *Minimize the following expression using k-maps and realize using NOR gates only*

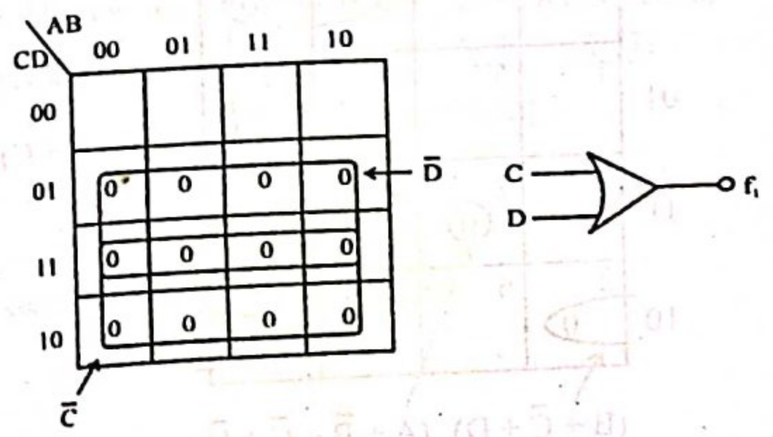
(

$f_1(A,B,C,D) = \Pi m(1,2,3,5,6,7,9,10,11,13,14,15)$

(b) $f_2(A,B,C,D) = \Pi m(1,4,6,9,10,11,14,15)$

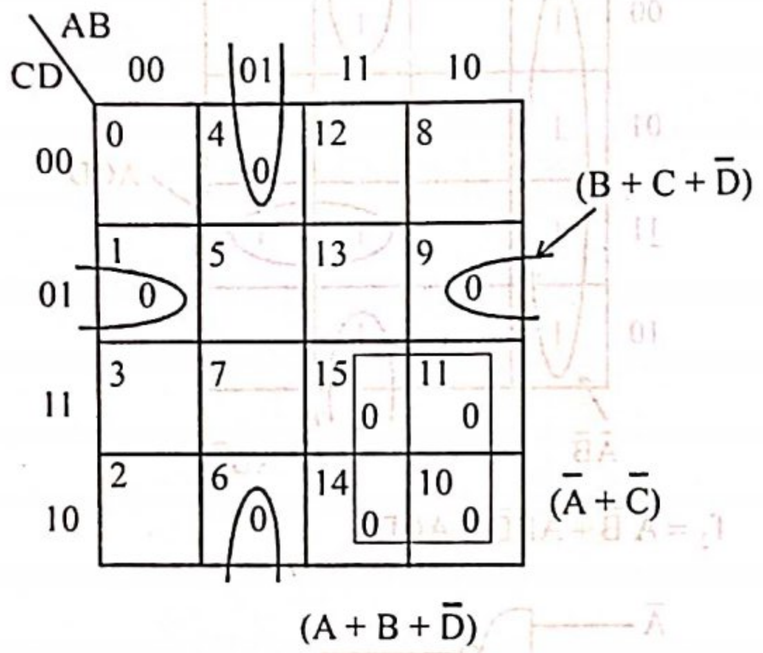
(c) $f_3(A,B,C,D) = \Pi m(2,7,8,9,10,12)$

Ans. (a) The k-map is shown in fig. (i)

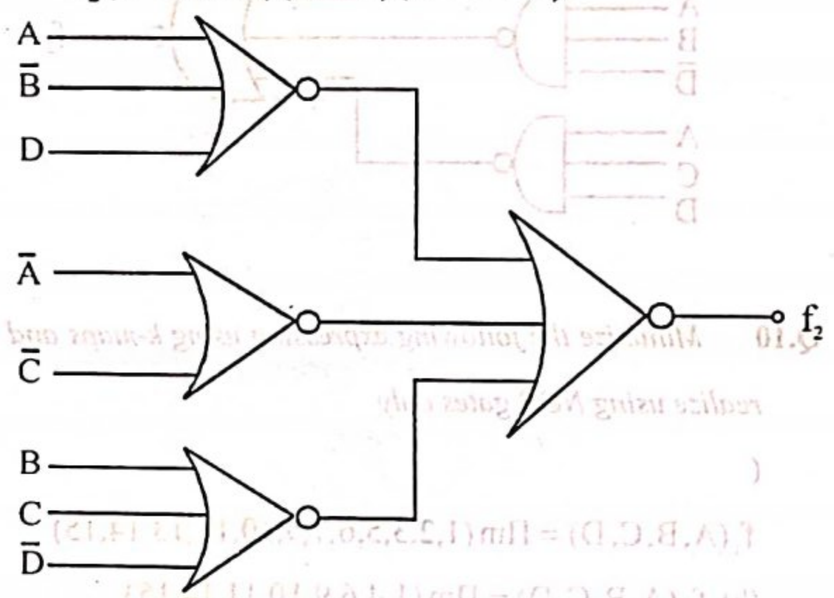


$f_1 = (\overline{C}).(\overline{D}) = \overline{C+D}$

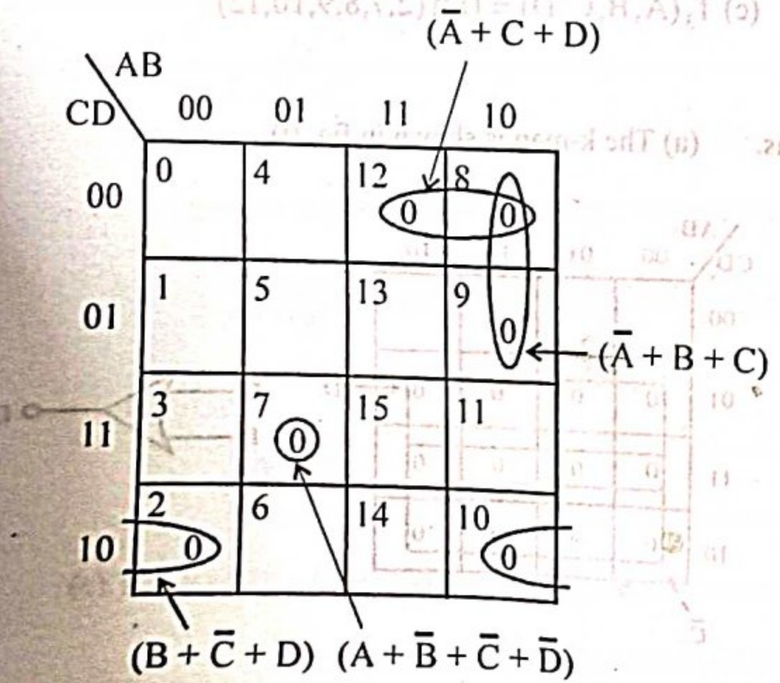
(b) $f_2(A,B,C,D) = \Pi m(1,4,6,9,10,11,14,15)$



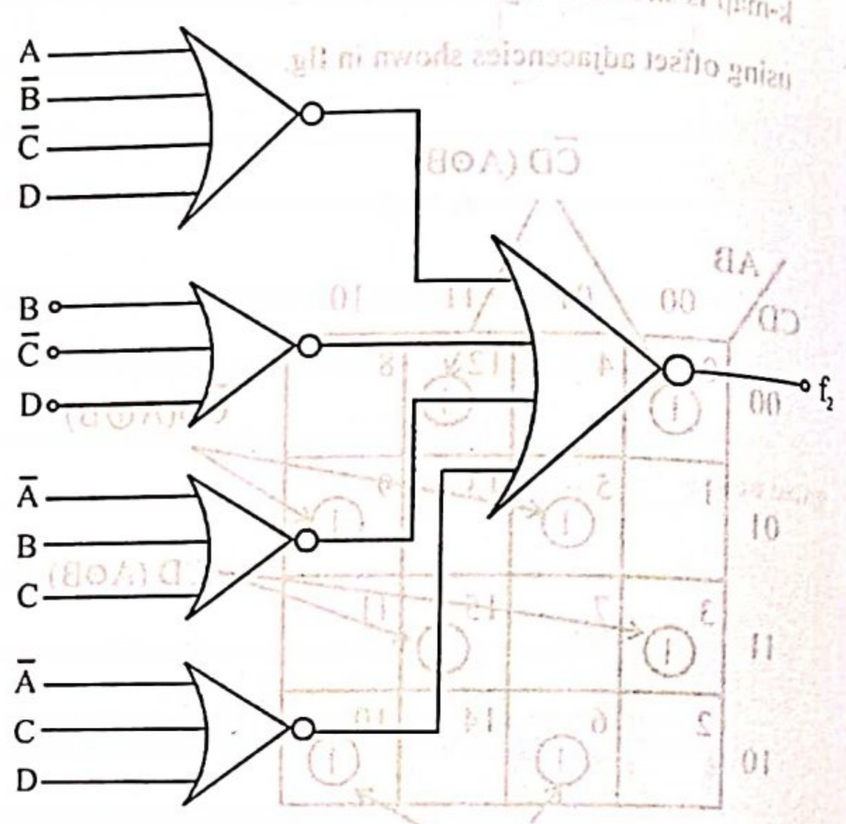
$f_2(A+\overline{B}+D)(\overline{A}+\overline{C})(B+C+\overline{D})$



(c) $f_3(A,B,C,D) = \Pi m(2,7,8,9,10,12)$



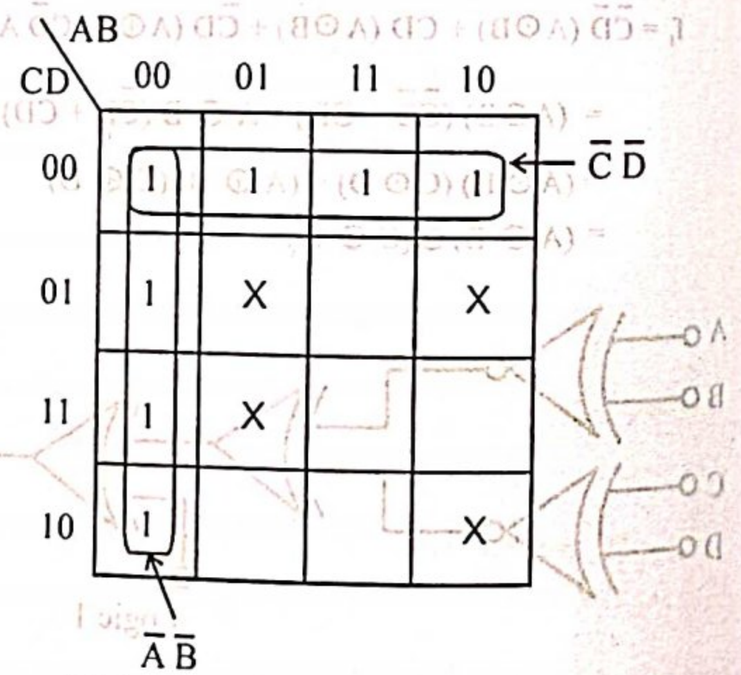
$f_3 = (A+\overline{B}+\overline{C}+D)(B+\overline{C}+D)(\overline{A}+B+C)(\overline{A}+C+D)$



Q.12. *Simplify the expression given below using k-map.*

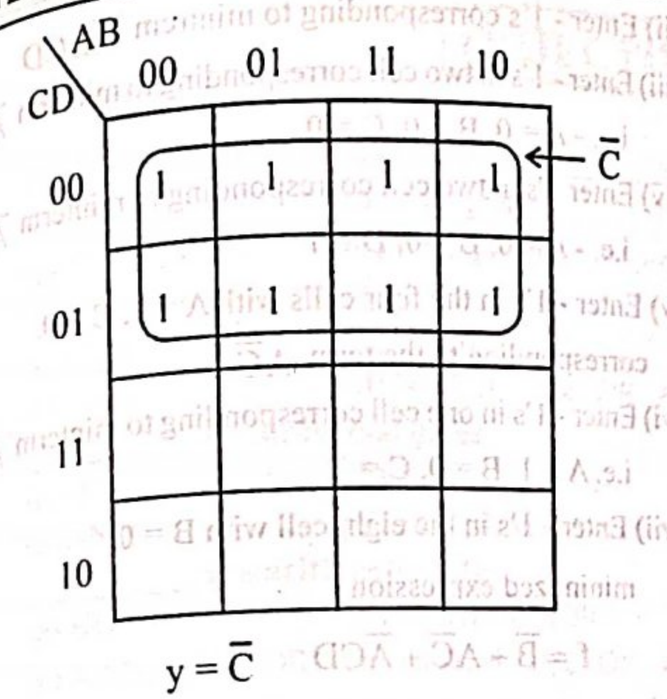
(i) $y = \Sigma(1,2,3,4,8,12) + d(0,5,7,9,10)$ **[2005(A)]**

Ans.



$y = \overline{A}\,\overline{B} + \overline{C}\,\overline{D}$

(ii) $y = \Sigma(0,1,4,5,8,9,12,13)$



Q.13 *Minimize the following logic functions and realize using NAND / NOR gates.*

(a) $f_1(A,B,C,D) = \Sigma m(1,3,5,8,9,11,15) + d(2,13)$

(b) $f_2(A,B,C,D) = \Pi m(1,2,3,8,9,10,11,14).d(7,15)$

Ans. (a)

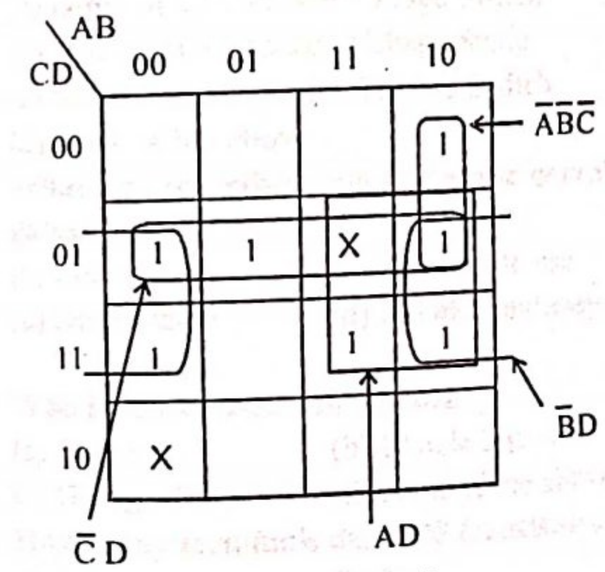
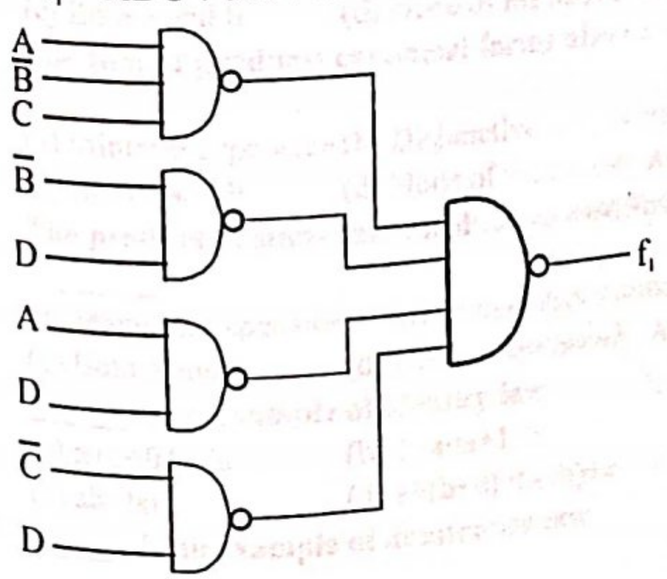


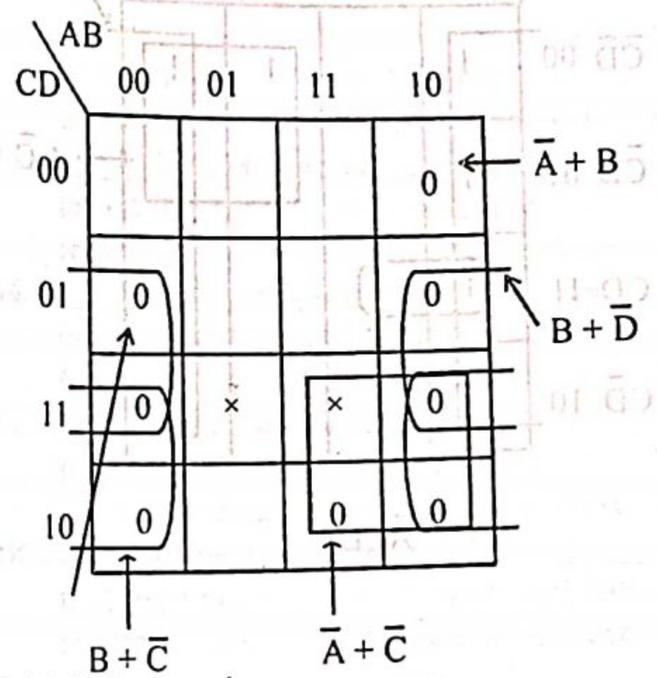
Fig. (i)

minimized expression

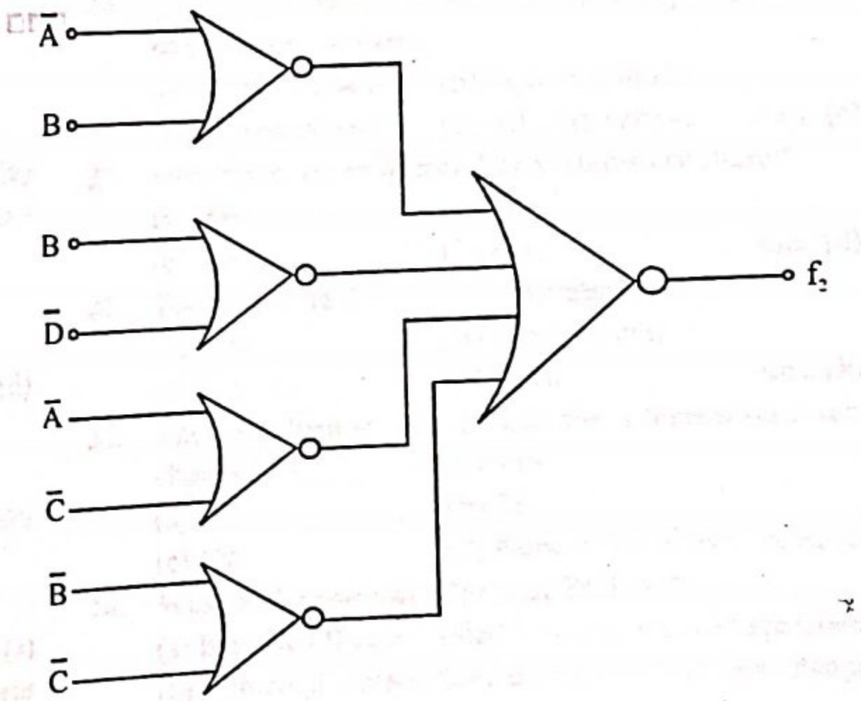
$f_1 = A\overline{B}\overline{C} + \overline{B}D + AD + \overline{C}D$



(b) $f_2(A,B,C,D) = \Pi m(1,2,3,8,9,10,11,14)$



$f_2 = (\overline{A}+B)(B+\overline{D})(\overline{A}+\overline{C})(B+\overline{C})$



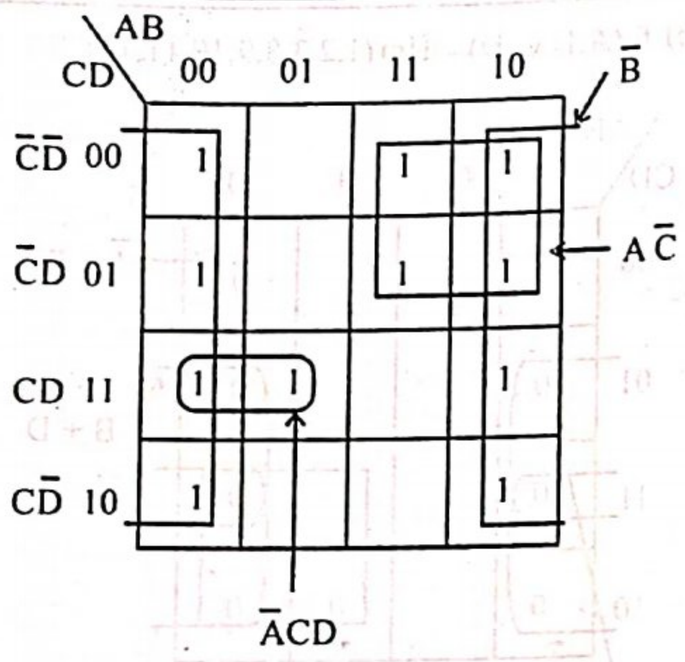
Q.14. *Minimize the four variable logic funciton using k-map*

$f(A,B,C,D) = AB\overline{C}D + \overline{A}BCD + \overline{A}B\overline{C} + \overline{A}B\overline{D} + A\overline{C} + A\overline{B}C + \overline{B}$ **[2009(A)]**

Ans. (i) Enter 1 in the cell with

A = 1, B = 1, C = 0, D = 1

corresponding to minterm $AB\overline{C}D$



(ii) Enter - 1's corresponding to mintrem $\bar{A}BCD$

(iii) Enter - 1's in two cell corresponding to minterm $\bar{A}\bar{B}\bar{C}$
i.e. - A = 0, B = 0, C = 0

(iv) Enter 1's in two cell corresponding to minterm $\bar{A}\bar{B}D$
i.e. - A = 0, B = 0, D = 1

(v) Enter - 1's in the four cells with A = 1, C = 0.
corresponding to the term $A\bar{C}$

(vi) Enter - 1's in one cell corresponding to minterm $A\bar{B}\bar{C}$
i.e. A = 1, B = 0, C = 1

(vii) Enter - 1's in the eight cell with B = 0.
minimized expression

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

□□□

OBJECTIVE TYPE QUESTIONS

1. The logical gates are categorized into _____
(a) One group (b) Two groups
(c) Three groups (d) Four groups Ans. (c)

2. _____ are basic gates
(a) NOT (b) NAND
(c) AND (d) NOT, AND, & OR Ans. (d)

3. _____ are universal gates
(a) NOT (b) NAND & NOR
(c) AND (d) NOT, AND, & OR Ans. (b)

4. _____ are arithmetic gates
(a) NOT (b) NAND & NOR
(c) X-OR & X-NOR (d) NOT, AND, & OR Ans. (c)

5. _____ are the common forms of complex logic gates
(a) OR-AND-Invert (OAI)
(b) AND-OR-Invert (AOI)
(c) Both OAI and AOI
(d) None of the above Ans. (c)

6. What is the standard form of DCDVS logic?
(a) Differential Cascade Voltage Switch
(b) Differential Cascade Voltage Static
(c) Differential Complex Voltage Switch
(d) None of the above Ans. (a)

7. What are the advantages of static complementary gates?
(a) Reliable (b) Not easy to use
(c) Not reliable (d) Reliable and easy to use Ans. (d)

8. Who invented Boolean algebra?
(a) Bardeen (b) Claude Shannon
(c) George Boole (d) None of the above Ans. (c)

9. How many terminals do MOS transistors have?
(a) One (b) Two
(d) Three Four Ans. (c)

10. _____ are the alternative form of canonical form
(a) Sum of products (b) Product of sums
(c) Both a and b (d) None of the above Ans. (a)

11. The sum of products canonical forms also known as _____
(a) Minterm expansion (b) Disjunctive normal form
(c) Both a and b (d) None of the above Ans. (a)

12. The product of sums canonical forms also known as _____
(a) Maxterm expansion (b) Conjunctive normal form
(c) Both a and b (d) None of the above Ans. (a)

13. _____ is an example of identity law
(a) a+0=0+a=a (b) 1+a=a+1=1
(c) ab=ba (d) a+(b+c)=(a+b)+c Ans. (a)

14. _____ is an example of dominance law
(a) a+0=0+a=a (b) 1+a=a+1=1
(c) ab=ba (d) a+(b+c)=(a+b)+c Ans. (b)

15. _____ is an example of commutativity law
(a) a+0=0+a=a (b) 1+a=a+1=1
(c) ab=ba (d) a+(b+c)=(a+b)+c Ans. (c)

16. _____ is an example of associativity law
(a) a+0=0+a=a (b) 1+a=a+1=1
(c) ab=ba (d) a+(b+c)=(a+b)+c Ans. (d)

17. _____ is an example of distributive law
(a) a+0=0+a=a (b) 1+a=a+1=1
(c) a+bc=(a+b)(a+c) (d) a+(b+c)=(a+b)+c Ans. (c)

18. Combinational logic is used to _____
(a) Compute outputs (b) Compute new states
(c) Both a and b (d) None of the above Ans. (b)

19. The sequential logic contains _____
(a) Memory elements
(b) Memory is provided by feedback
(c) Both a and b (d) None of the above Ans. (b)

20. _____ are the methods used to represent negative integer numbers
(a) 1's compliment (b) Sign magnitude
(c) 2's compliment (d) All of the above Ans. (c)

21. How many types of number systems are there?
(a) One (b) Two
(c) Three (d) Four Ans. (d)

22. The base is 16 for _____ number system
(a) Binary (b) Hexadecimal
(c) Decimal (d) Octal Ans. (b)

23. The American standard code for information interchange has _____ characters
(a) 64 (b) 25
(c) 128 (d) None of the above Ans. (c)

24. What is the standard form of ECDIC?
(a) Extended Binary Coded Decimal Interchange Code
(b) Extended Binary Coded hexadecimal Interchange Code
(c) Extended Binary Coded Decimal Information Code
(d) None of the above Ans. (a)

25. How many types of parities are there?
(a) One (b) Two
(c) Three (d) Four Ans. (b)

26. The ones complement of binary number 1010 is _____.
(a) 0101 (b) 1010
(c) 0110 (d) 1110 Ans. (a)

27. The 2's complement of binary number 1010 is _____
(a) 0101 (b) 1010
(c) 0110 (d) 1110 Ans. (c)

28. The base is eight for _____ number system
(a) Binary (b) Hexadecimal
(c) Decimal (d) Octal **Ans. (d)**

29. How many types of IC packages are there ?
(a) One (b) Two
(c) Three (d) Four **Ans. (b)**

30. IC's are categorized into _____.
(a) One (b) Two
(c) Three (d) Four **Ans. (b)**

31. How many gates does ultra large scale integration contain ?
(a) 100 gates (b) 1000 gates
(c) 10000 gates (d) More than 100,000 gates **Ans. (d)**

32. How many gates does very large-scale integration contain?
(a) 100 gates (b) 10,000 to 100,000 gates
(c) 10000 gates (d) None of the above **Ans. (b)**

33. How many gates does large-scale integration contain?
(a) 100 to 10,000 gates (b) 10,000 to 100,000 gates
(c) 10000 gates (d) None of the above **Ans. (a)**

34. How many gates does medium-scale integration contain?
(a) 100 to 10,000 gates (b) 10,000 to 100,000 gates
(c) 10 to 100 gates (d) None of the above **Ans. (d)**

35. The base is ten for _____ number system
(a) Binary (b) Hexadecimal
(c) Decimal (d) Octal **Ans. (c)**

36. The base is two for _____ number system
(a) Binary (b) Hexadecimal
(c) Decimal (d) Octal **Ans. (a)**

□□□

Chapter 3

Combinational Logic Circuits

Q.1. *What is a combinational circuit?*

Ans. In a combinational circuit, the output depends upon the present inputs only. It means its output does not depend on the previous inputs. The combinational circuit has no memory element. It consists of logic gates only.

Q.2. *What are the different characteristics of combinational circuits ?*

Ans. The different characteristics of combinational circuits are :-

(i) In combinational circuits, the output exists as long as the input present.

(ii) A combinational logic circuit can be fabricated from NAND or NOR gates.

(iii) A combinational logic circuit does not have a memory.

Q.3. *What is the difference between combinational logic circuit and sequential logic circuit with suitable examples.* **[BR-2005(A)2010Even]**

Ans. *Combinational logic ckt :-*

(i) Its output depends upon the inputs present at that instant of time.

(ii) It does not have a memory

(iii) It does not have a feedback path from output to input

(iv) It does not have a clock signal

(v) Its circuit is simpler than that of sequential logic circuit.

(vi) It is built using basic gate (AND, OR, NOT) or universal gat (NOR, NAND)

(vii) Its examples are adder, subtractor, multiplexer, demultiplexer, decoder, encode, comparator etc.

Sequential logic circuit :-

(i) Its output depend upon the present input as well as past Input/output.

(ii) It has a memory

(iii) It has a feedback path from output to input

(iv) It may or may not have a clock signal

(v) Its circuit is more complex than that of a combinational logic ckt.

(vi) It is built using basic gates and combinational logic circuit

(vii) Its examples are flip-flop, counters, registers etc.

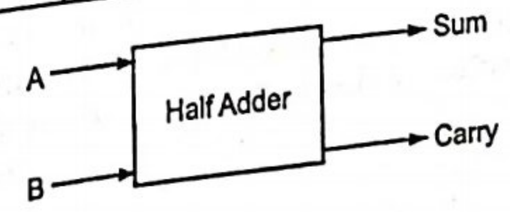
Q.4 *Draw the circuits of half adder and full adder and discuss their working. Draw their truth tables.* **[BR-2003(A), 2004(A)]**

Ans. Half Adder :

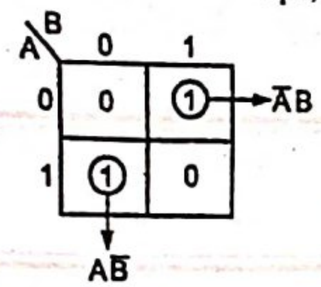
- A binary half adder is a combinational logic circuit which adds two bits of binary data, producing a sum but and a carry bit as the two output signals.
- It has two one bit inputs A and B and two outputs sum and carry.

Truth Table

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

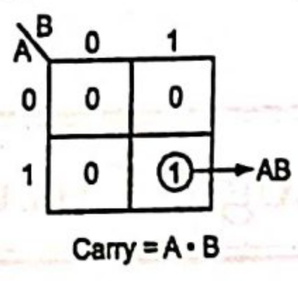


We draw two K-maps, one for sum and other for carry.



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

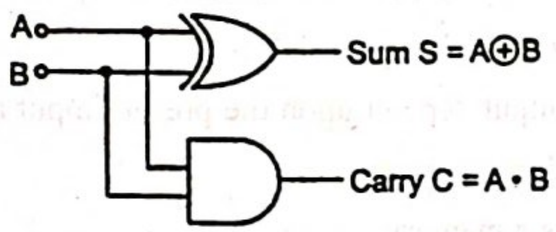
K-map for sum output



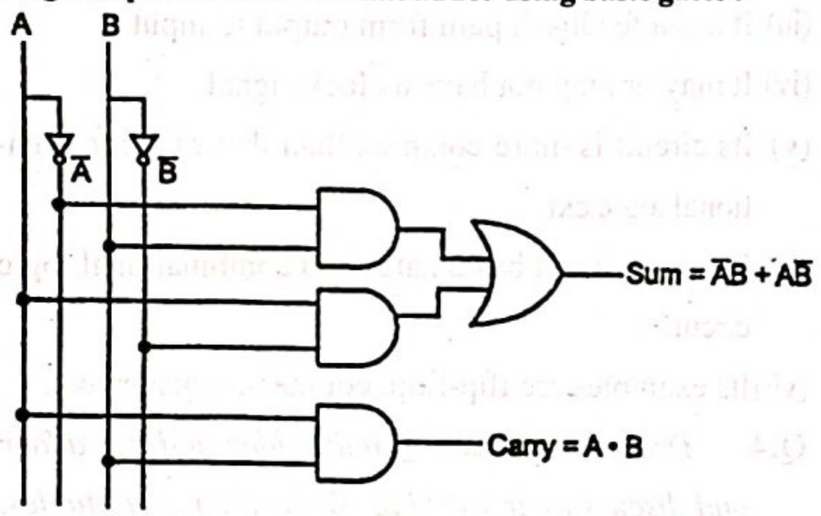
$Carry = A \cdot B$

K-map for carry output

Logic implementation :



Logic implementation of half adder using basic gates :



Half Adder using only NAND gates

Consider the equation for sum,

$$S = \bar{A}B + A\bar{B}$$
$$Carry = AB$$

$$S = \overline{\overline{\bar{A}B + A\bar{B}}} \quad (\because \bar{\bar{A}} = A)$$

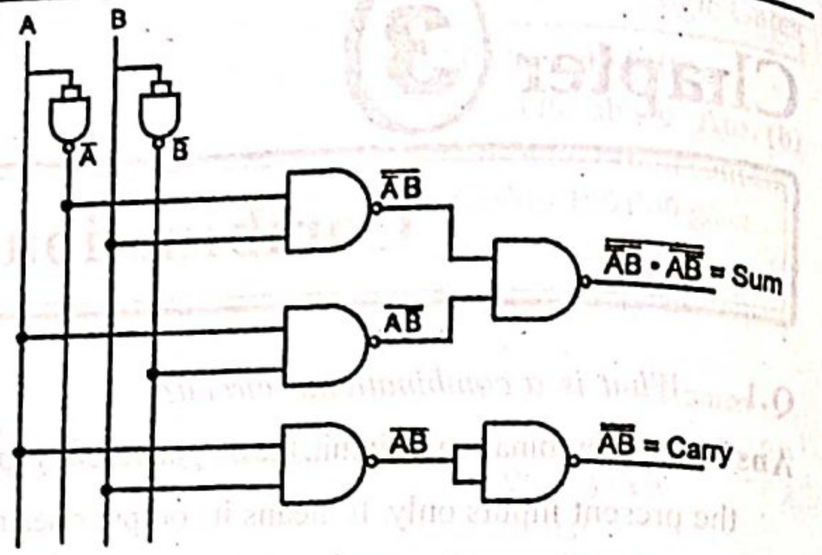
Applying De Morgan's theorem,

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

Now we can implement using NAND gates.

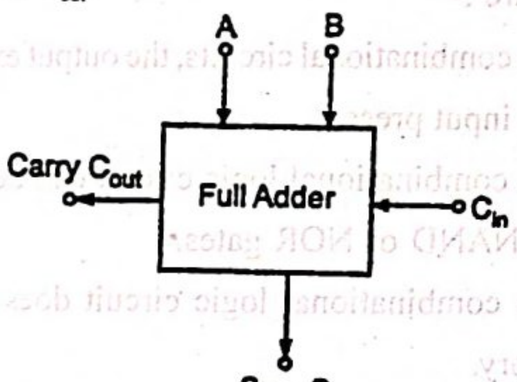
Similarly, $Carry = \overline{\overline{AB}} \quad (\because \bar{\bar{A}} = A)$

A half adder by itself is not of much practical use. A third input is required for carry, thus giving rise to the development of full adder, which .is an adder with three inputs, the third input being required to perform the addition with carry.



Full Adder :

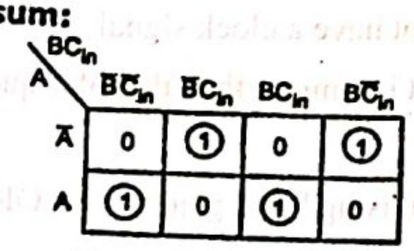
- A full adder is a combinational circuit that performs sum of three input bits. This circuit has three inputs and two outputs.
- The inputs A and B represent the two bits to be added. The C_{in} represents the carry from the previous lower significant position. There are two outputs: sum S and carry C_{out}.



Truth Table

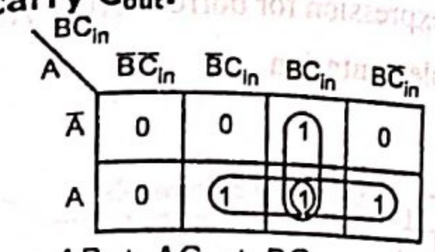
Inputs			Outputs	
A	B	C_{in}	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

K-map for sum:



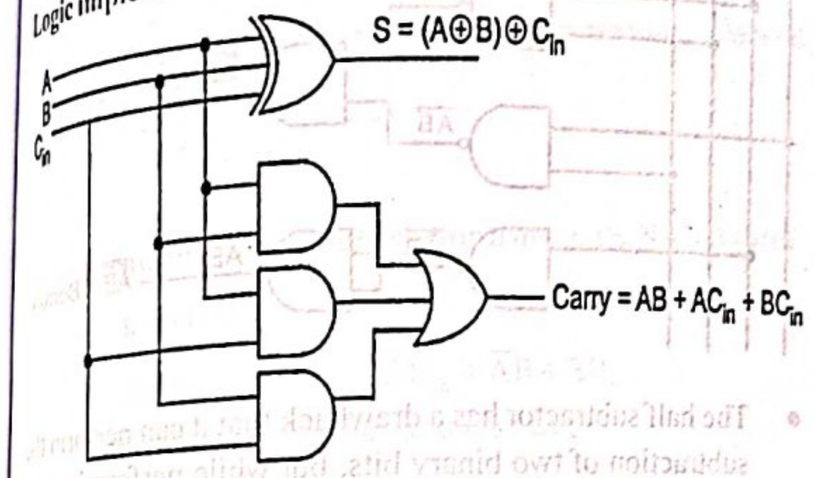
$Sum = \bar{A}\bar{B}\,C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in}$

K-map for carry C_{out}:



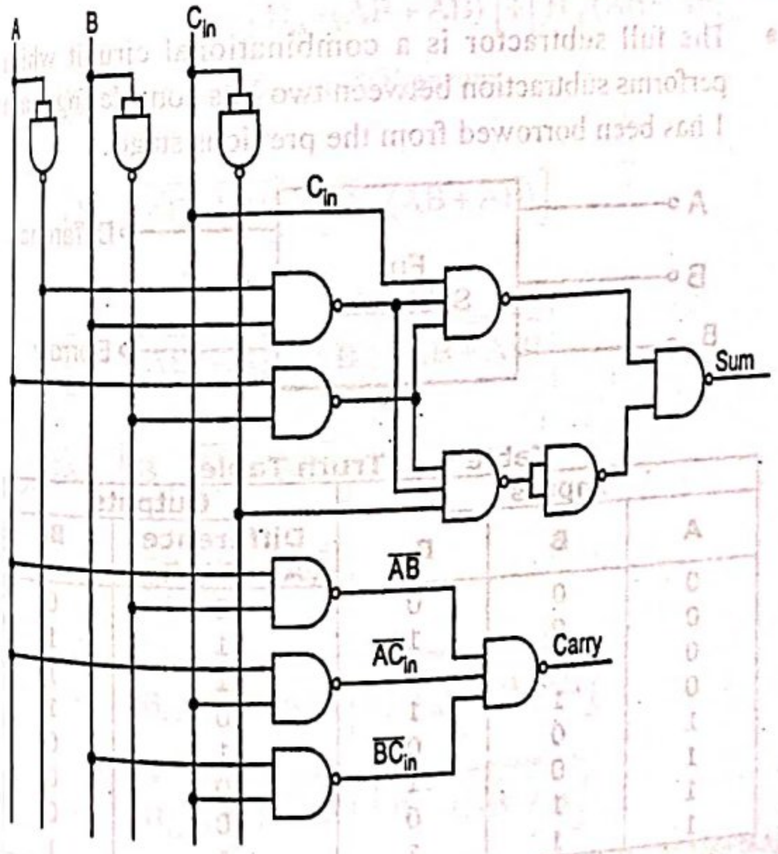
$C_{out} = AB + AC_{in} + BC_{in}$

Logic implementation of full adder :



Full adder circuit using only NAND gates :

- The full adder circuit is expressed using the following expressions :



a

$$Sum\ S = C_{in} \oplus A \oplus B = C_{in}\,\overline{(\bar{A}B + A\bar{B})} + \bar{C}_{in}\,(\bar{A}B + A\bar{B})$$

$$= C_{in}\,\overline{(\bar{A}B \cdot \bar{A}\bar{B})} + \bar{C}_{in}\,\overline{(\bar{A}B + A\bar{B})} = C_{in}\,(\bar{A}B \cdot \bar{A}\bar{B}) + \bar{C}_{in}\,\overline{(\bar{\bar{A}}\bar{B} \cdot \bar{A}\bar{\bar{B}})}$$

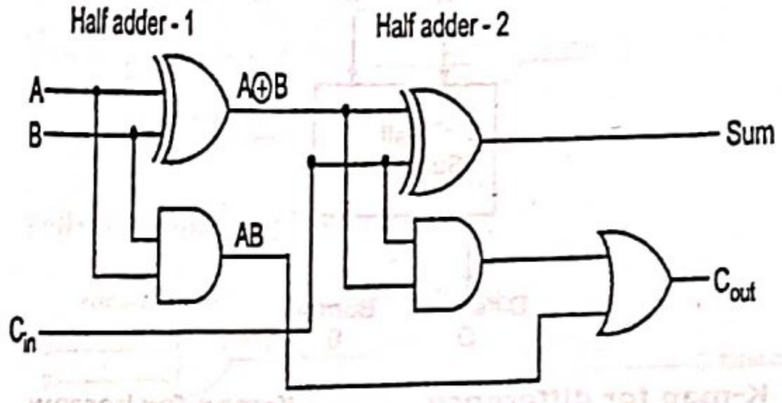
$$= \overline{C_{in}\,(\bar{\bar{A}}\bar{B} \cdot \bar{A}\bar{B}) + \bar{C}_{in}\,(\bar{A}B \cdot \bar{A}\bar{\bar{B}})}$$

$$\therefore S = \overline{C_{in}\,(\bar{A}B \cdot \bar{A}\bar{B}) \cdot \bar{C}_{in}\,(\bar{A}B \cdot \bar{A}\bar{B})}$$

- We can implement using NAND gates at this stage.

Similarly,

$$C_o = \overline{AB + AC_{in} + BC_{in}} = \overline{\overline{AB} \cdot \overline{AC_{in}} \cdot \overline{BC_{in}}}$$



Applications of full adder :

1. Used in digital computers.
2. Full adders act as the basic building blocks of BCD adder IC 7483.

Q.5. *Draw the circuits of half subtractor and full subtractor and discuss their working. Draw their truth tables.*

Or Define sutractor and explain its type ?

Or Define half subtractor with truth table

Ans.

Subtractor :

- The subtraction of two binary numbers can be done by taking the complement of the subtrachend and adding it to minuend. The rules of binary substraction are as as follows :

$0 - 0 = 0$
$0 - 1 = 1$ (Borrow = 1)
$1 - 0 = 1$
$1 - 1 = 0$

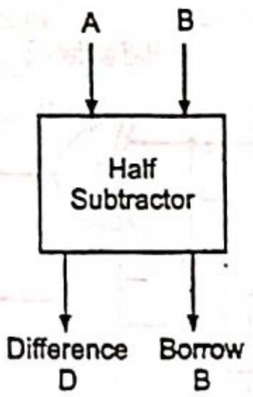
There are two types of binary subtractors :

1. Half subtractor 2. Full subtractor

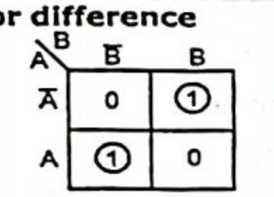
1. Half subtractor : Half subtractor is a combinational circuit with two inputs and two outputs (difference and borrow).

Truth Table

Inputs		Outputs	
A	B	Difference A – B	Borrow B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

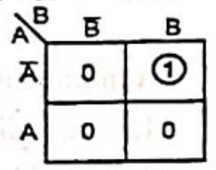


K-map for difference



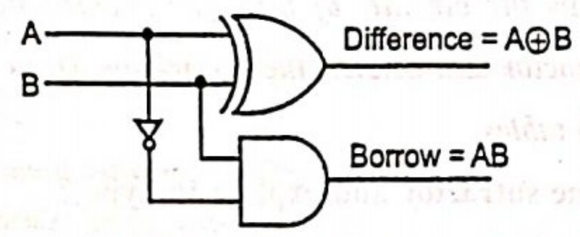
Difference = $\bar{A}B + A\bar{B}$ = $A \oplus B$

K-map for borrow

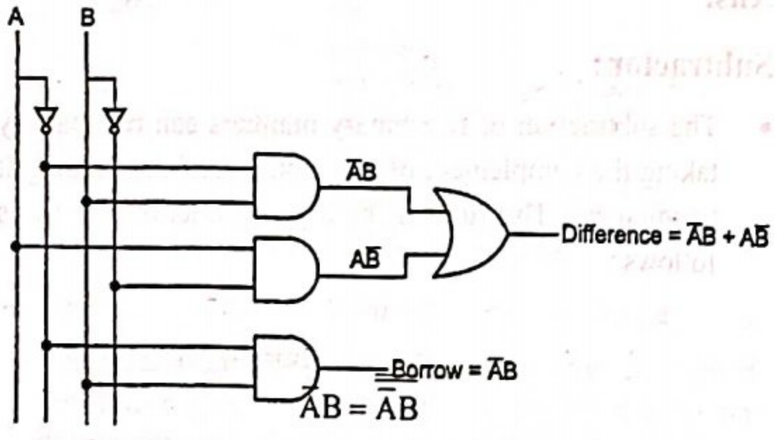


Borrow = $\bar{A}B$

Logic implementation of half subtractor :



Logic implementation using basic gates :



Half subtractor using NAND gates

The expression for difference

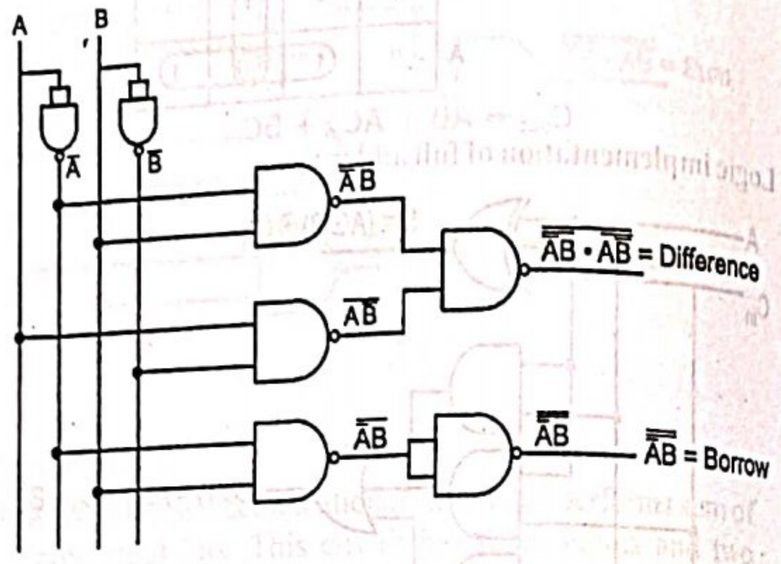
$= \bar{A}B = A\bar{B}$

$= \overline{\overline{\bar{A}B} = A\bar{B}}$ $(\because \bar{\bar{A}} = A)$

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

The expression for borrow $\bar{A}B = \overline{\overline{\bar{A}B}}$

Logic implementation



- The half subtractor has a drawback that it can perform the subtraction of two binary bits, but while performing this subtraction, it does not consider the borrow from the previous stage.

Full Subtractor :

- The full subtractor is a combinational cirucit which performs subtraction between two bits considering that a 1 has been borrowed from the previous stage.

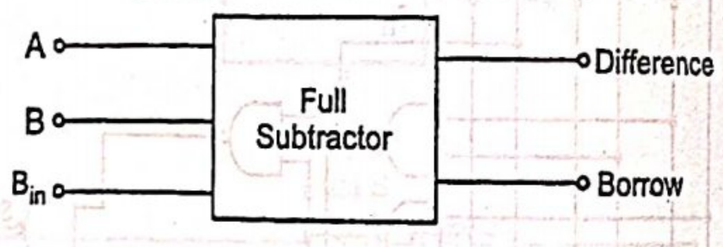
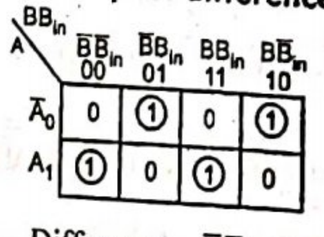


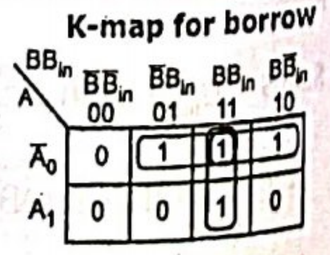
Table Truth Table

Inputs			Outputs	
A	B	B_{in}	Difference (A – B – B_{in})	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

K-map for difference



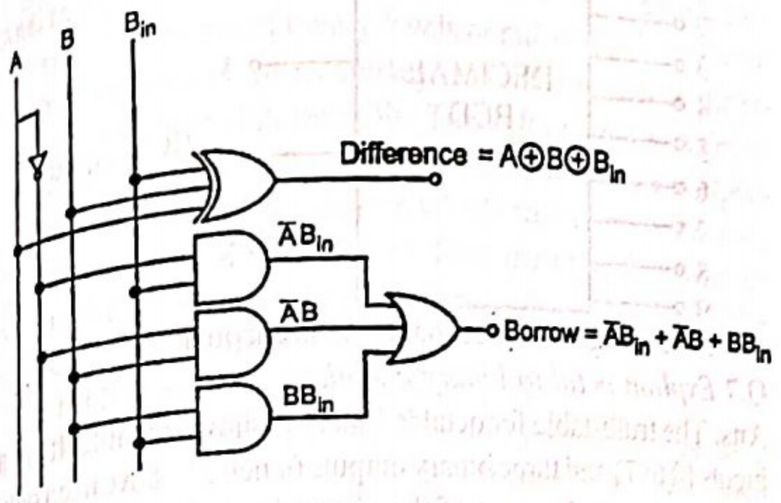
K-map for borrow



Difference = $\bar{A}\bar{B}B_{in} + \bar{A}B\bar{B}_{in} + A\bar{B}\bar{B}_{in} + ABB_{in}$

Borrow = $\bar{A}B_{in} + \bar{A}B + BB_{in}$

Logic implementation of full subtractor :



Implementation of full subtractor using only NAND gates

Difference = $B_{in} \oplus A \oplus B$

Borrow = $\bar{A}B_{in} + \bar{A}B + BB_{in}$

Now difference can be expanded as :

Difference =

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

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

Applying De Morgan's theorem,

Difference =

$\left[B_{in} \cdot \overline{(\bar{A}B \cdot A\bar{B})}\right] + \left[\overline{B_{in}}(\bar{A}B + A\bar{B})\right]$

$= \left[B_{in} \cdot \overline{(\bar{A}B \cdot A\bar{B})}\right] + \left[\overline{B_{in}}\,\overline{(\bar{A}B + A\bar{B})}\right]$

$= \left[B_{in} \cdot \overline{(\bar{A}B \cdot A\bar{B})}\right]$

$= \left[\overline{B_{in} \cdot \overline{(\bar{A}B \cdot A\bar{B})}}\right]$

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

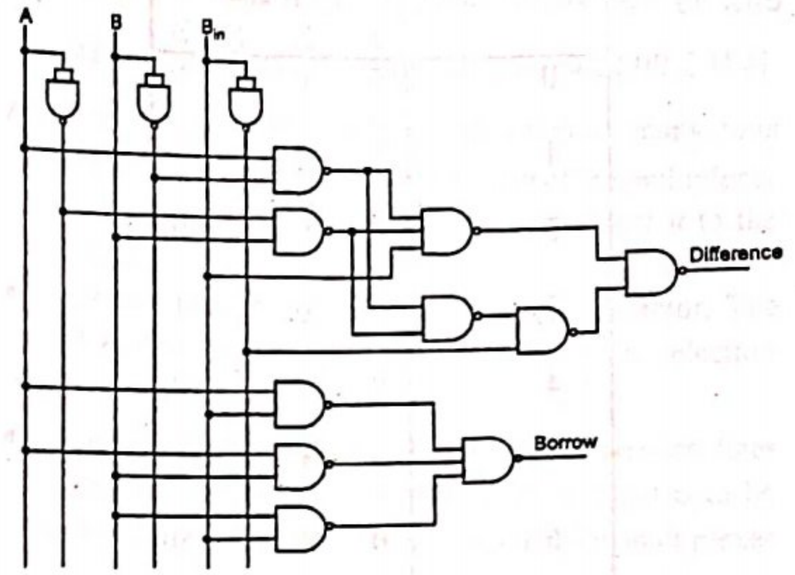
$= \overline{\left[B_{in} \cdot \overline{(\bar{A}B \cdot A\bar{B})}\right] \cdot \left[\overline{B_{in}} \cdot \overline{(\bar{A}B \cdot A\bar{B})}\right]}$

We can implement using only NAND gates at this stage.

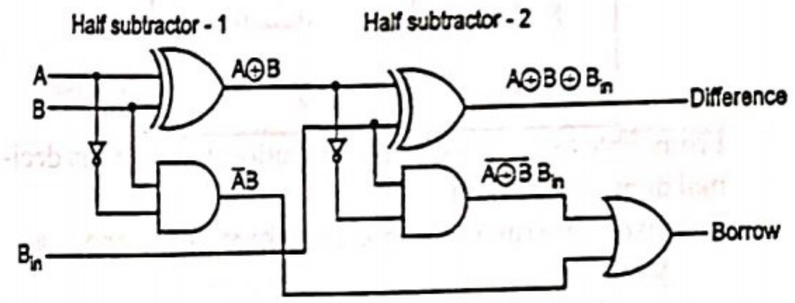
Borrow = $\bar{A}B_{in} + \bar{A}B + BB_{in}$

$= \overline{\overline{\bar{A}B_{in} + \bar{A}B + BB_{in}}} = \overline{\overline{\bar{A}B_{in}} \cdot \overline{\bar{A}B} \cdot \overline{BB_{in}}}$

Logic implementation using only NAND gates :

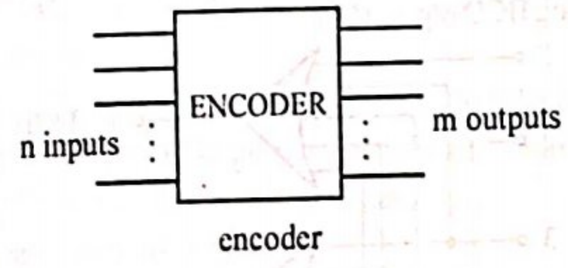


Full subtractor using Half subtractor



Q.6. ***What is encoder? Draw the logic circuit of decimal to BCD encoder and explain its working.***

Ans. Digital computers operates on binary system. But, we work on decimal numbers and alphabets. A encoder is a device which converts alphanumeric characters to binary codes. An encoder may be decimal to binary, hexadecimal to binary, octal to BCD etc, Block diagram of encoder is shown in fig.



It has n input lines and m output lines. At a time only one of the input lines is active and encoder converts is to a coded binary output with m bits.

Logic circuit of a decimal to BCD encoder is shown in fig.

The encoder has 10 inputs (0 to 9) and 4 outputs for the BCD number. Thus it is 10 line to 4 line encoder. Decimal to BCD equivalent number is shown in table (a)

Decimal digit	BCD code y_3	y_2	y_1	y_0
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

From table (a), we can find the relationship between decimal digit and BCD bit.

Thus we can write OR expression for y_3, y_2, y_1 and y_0 as

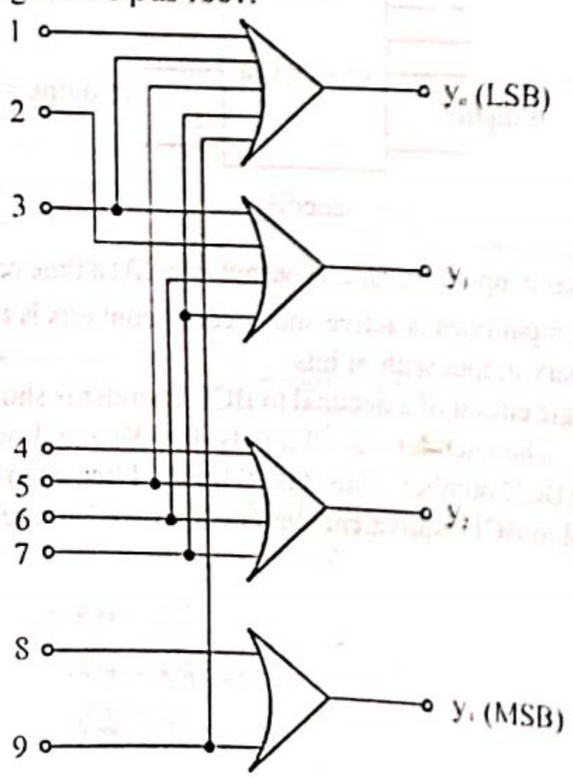
$y_3 = 8 + 9$

$y_2 = 4 + 5 + 6 + 7$

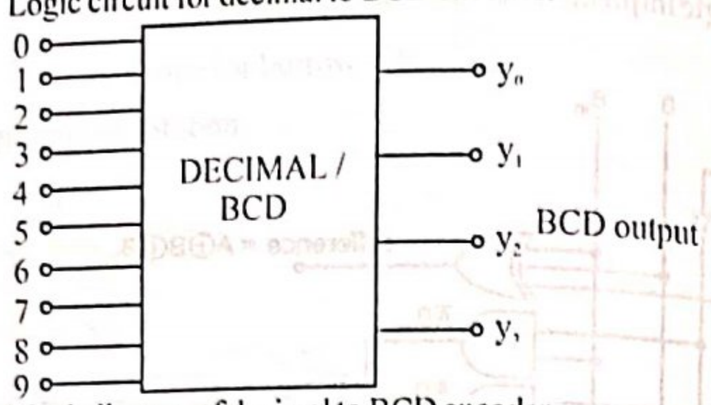
$y_1 = 2 + 3 + 6 + 7$

$y_0 = 1 + 3 + 5 + 7 + 9$

with the help of this expression we can draw the logic circuit of decimal to BCD encoder. When a high signal appears on any of input lines the corresponding OR gates give the BCD output, e.g., if decimal input is 4, high appears on only y_2 and low on y_3, y_1, y_0, thus giving the BCD code for decimal 4 as 0100. Similarly decimal input is 9, then high comes on outputs y_3 and y_0 and low on y_2 and y_1, thus giving BCD o/p as 1001.



Logic circuit for decimal to BCD encodes.



Block diagram of decimal to BCD encoder.

Q.7 *Explain octal to binary encoder.*

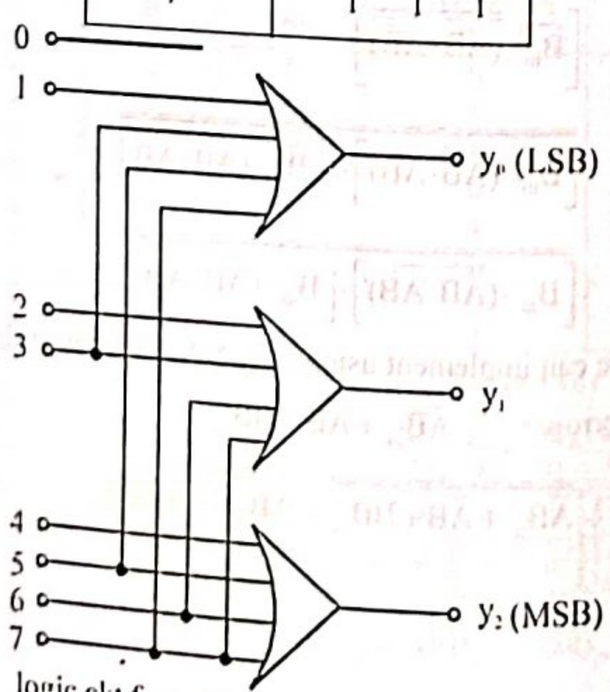
Ans. The truth-table for octal to binary is shown in table. It has 8 inputs (0 to 7) and three binary outputs (since $2^3 = 8$ we need only 3 output lines). The rest of the circuit is similar to decimal to binary encoder. The logic circuit of octal to binary is shown in fig.

This encoder consists of three OR gates. When any of inputs is high, the corresponding OR gate give high output, i.e., if we press 3, y_0 and y_1 output will high and y_2 will low and it gives 011 which is equivalent binary number of octal 3.

If we press 7, all the three OR gates give high output and the output is 111. Similarly we can check the outputs for the remaining inputs.

Truth-table for octal to binary encoder

Input	Output y_2	y_1	y_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1



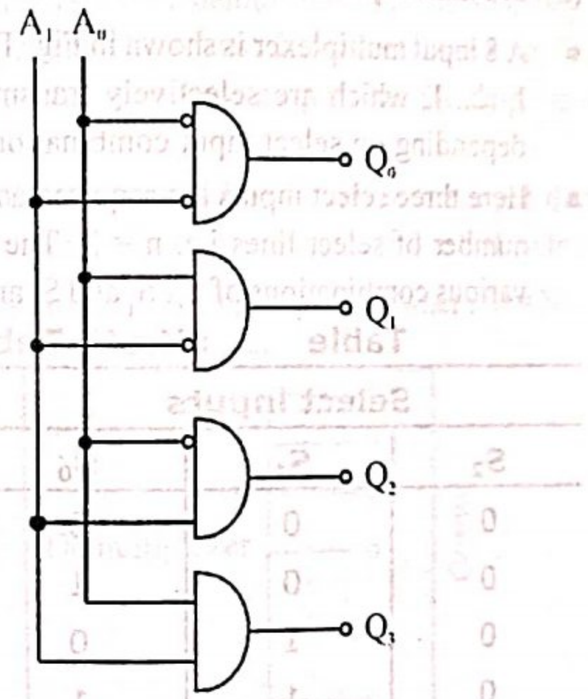
logic ckt for octal to binary encoder.

Q.8 *What is decoder? Explain binary to decimal decoder.* **[Bh.2009,2010]**

Ans. In digital system, instructions as well as numbers are conveyed by means of binary levels or pulse trains. If 4 bits of a character are sent to convey instructions, then 16 different instructions are possible. This information is coded in binary form. But we work on decimal numbers and alphabets. A decoder is a device which converts binary word into alphanumeric charactor. Thus the input to a decoder are the bit 1, 0 and their combinations. The output is the corresponding decimal number.

2 Bit Binary to decimal decoder :-

This decoder is also called 2 line to 4 line decoder because there are 2 inputs and 4 outputs (because $2^2 = 4$). Table shows all the possible combination of input and output words.



(a) gate structure

Truth table

Input A_1	A_0	Output y_0	y_1	y_2	y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

The important characteristics of a decoder is that for each input A_1, A_0 one and only one output is at logic 1. The decoder consists of AND gates. For simplicity we draw a decoder which has two inputs and four outputs. A decoder with n inputs would require 2^n gates and provide 2^n outputs.

Q.9. Explain with block diagram multiplexer and explain its classification?[Bh.2004,2009,2014]

Ans. The multiplexer is a digital circuit which has many input lines and one output line. The function of the multiplexer is to select one of the input lines and connect it to the output.

- The multiplexer is also known as data selector. The selection of desired input is done by means of selection lines.
- Generally there are 2^m input lines and n selection lines whose bit combinations determine which input is to be selected. The functional block diagram for multiplexer (abbreviated as MUX) is shown in Fig.

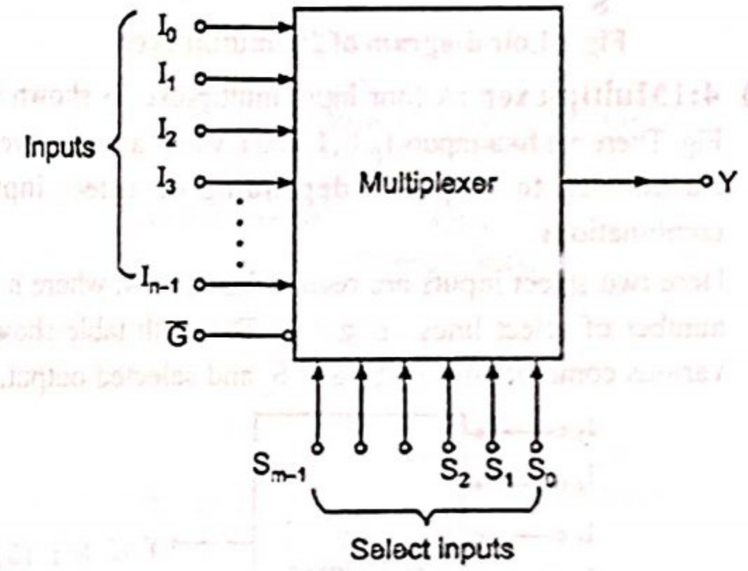
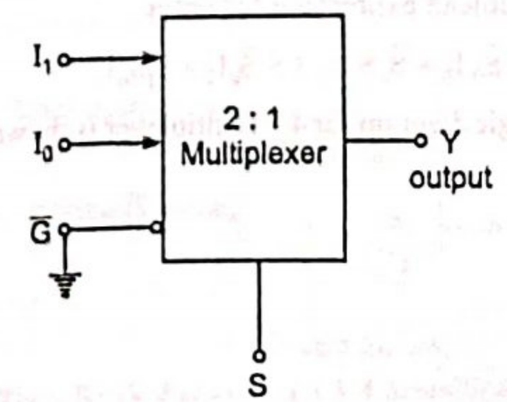


Fig. : Functional block diagram of a multiplexer

- Generally a strobe (or enable) input is incorporated which helps in cascading and it is generally active low which means it performs its intended operation when it is low.
- Standard ICs are available for: 2 : 1 MUX, 4 : 1 MUX, 8:1 MUX and 16:1
- Use of multiplexers offers the following advantages :
  1. Simplification of logic expression is not required.
  2. It minimizes the IC package count.
  3. Logic diagram is simplified.

(1) 2 :1 Multiplexer : A 2 :1 multiplexer is shown in Fig. It has two inputs I_0 and I_1 and one output Y. The number of select lines required are/is one only.



- The Boolean expression for the output is :

$$Y = I_0 \bar{S} + I_1 S$$

when $S = 0; Y = I_0$

when $S = 1; Y = I_1$

- The logical diagram of 2 : 1 multiplexer is shown in Fig.

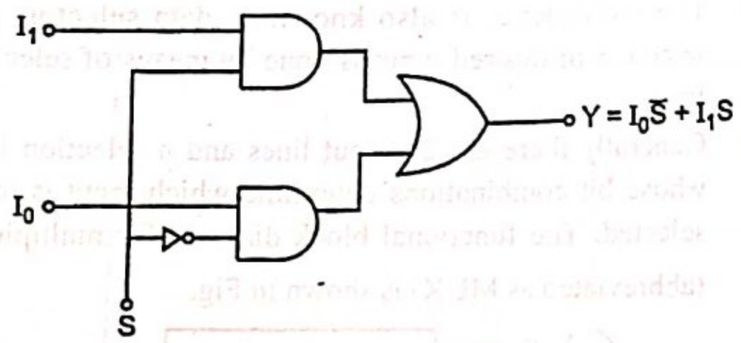


Fig. : Loic diagram of 2:1 multiplexer

(2) 4:1Multiplexer : A four input multiplexer is shown in Fig. There are four inputs I_0, I_1, I_2 and I_3 which are selectively transmitted to output Y depending on select input combinations.

- Here two select inputs are required as $2^n = 4$, where n is number of select lines i.e. n = 2. The truth table shows various combinations of S_0 and S_1 and selected output.

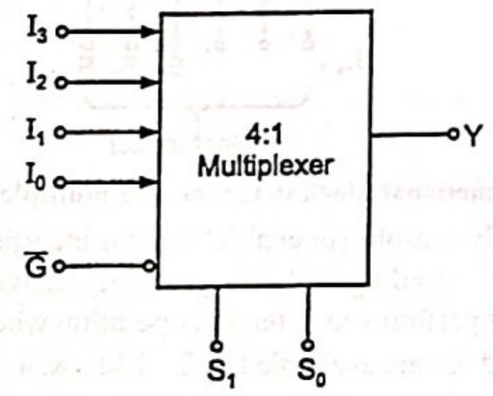


Fig. : Block diagram of 4:1 multiplexer

Select Inputs		Output
S_0	S_1	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Table : Truth Table

The Boolean expression for outut

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

The logic diagram for 4:1 multiplexer is shown in Fig.

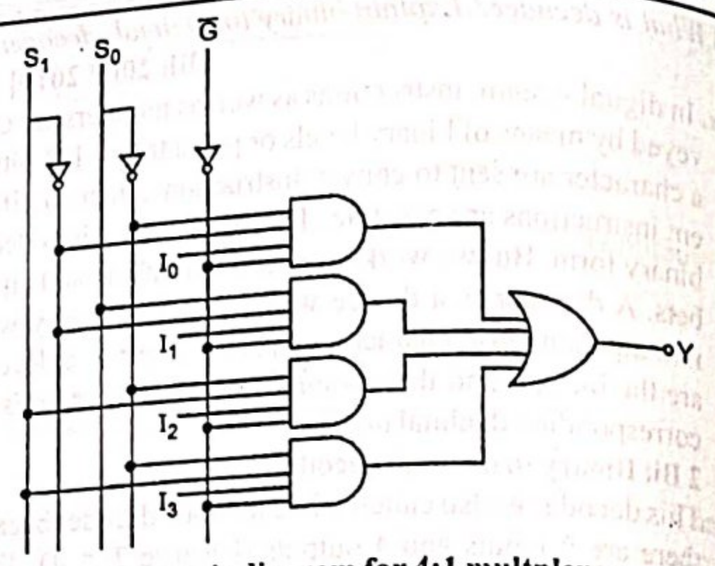


Fig. : Logic diagram for 4:1 multplexer

(3) 8:1 Multiplexer :

- A 8 input multiplexer is shown in Fig. There are 8 inputs I_0 I_1 ..... I_7 which are selectively transmitted to output Y depending on select input combinations.
- Here three select inputs are required as $2^n = 8$, where n is number of select lines i.e. n = 3. The truth table shows various combinations of S_0, S_1 and S_2 and selected output.

Table : Truth Table

Select inputs			Output
S_2	S_1	S_0	Y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

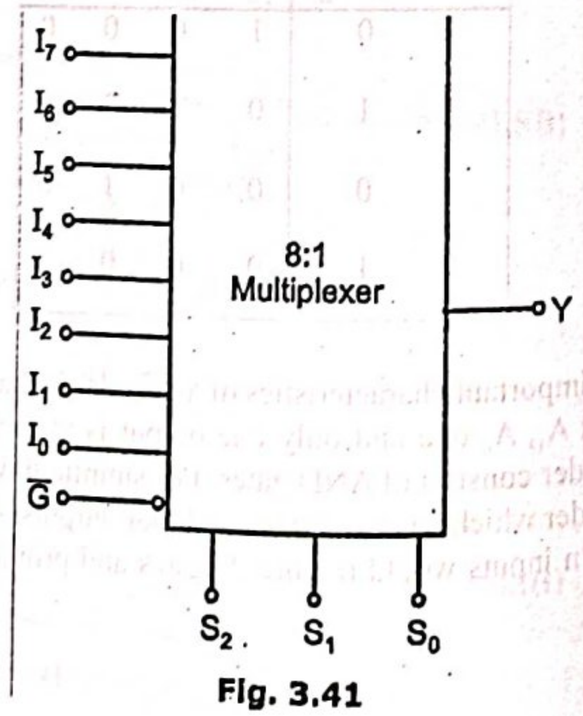


Fig. 3.41

The Boolean expression for output

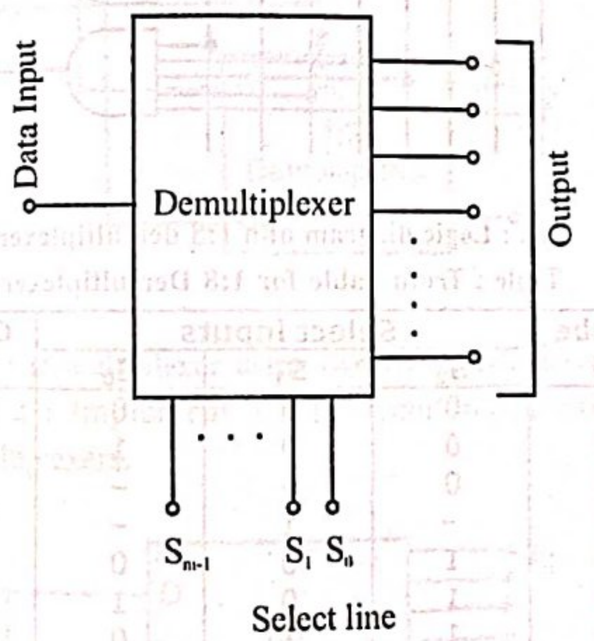
$$Y = \bar{S}_2 \bar{S}_1 \bar{S}_0 I_0 + \bar{S}_2 \bar{S}_1 S_0 I_1 + \bar{S}_2 S_1 \bar{S}_0 I_2 + \bar{S}_2 S_1 S_0 I_3 + S_2 \bar{S}_1 \bar{S}_0 I_4 + S_2 \bar{S}_1 S_0 I_5 + \bar{S}_2 \bar{S}_1 S_0 I_6 + S_2 S_1 S_0 I_7$$

The logic diagram for 8:1 multiplexer is shown in Fig.

Q.10. Explain with block diagram demultiplexer. and classify the demultiplexer **[Bh-2010,2016,2018]**

Ans. Demultiplexer : The demultiplexer is a digital circuit which has one input line and many output lines. It is used to send a single input on one of the output lines, thus it performs reverse operation of the multiplexer. The functional block diagram for a demultiplexer is shown in Fig.

- It has one input and N outputs. The select input code (bit pattern) determines to which output line the data input will be transmitted.
- In other words, the demultiplexer takes one input data source and selectively distributes it to 1 of N output channels. The number of select lines is n where $2^n = N$.



Types of Demultiplexers :

- The demultiplexers are classified as follows :
  1. 1 line to 2 line (1:2) demultiplexer.
  2. 1 line to 4 line (1:4) demultiplexer.
  3. 1 Sine to 8 line (1:8) demultiplexer. 4. 1 line to 16 line (1:16) demultiplexer.

(I) 1:2 Demultiplexer

- The block diagram of a 1:2 demultiplexer is as shown in Fig. It has one data input D and select line input S_0, one strobe (G) or enable (E) input and two outputs Y_0 and Y_1.
- It is active low. D is connected to Y_0 if $S_0 = 0$ and G = 0. Similarly D is connected to Y_1 if $S_1 = 1$ and G = 0.
- If G = 1, then both the outputs will be logic 0 irrespective of the inputs..

Table : Truth Table

Strobe	Select	Outputs	
G	S_0	Y_1	Y_0
1	X	0	0
0	0	0	D
0	1	D	0

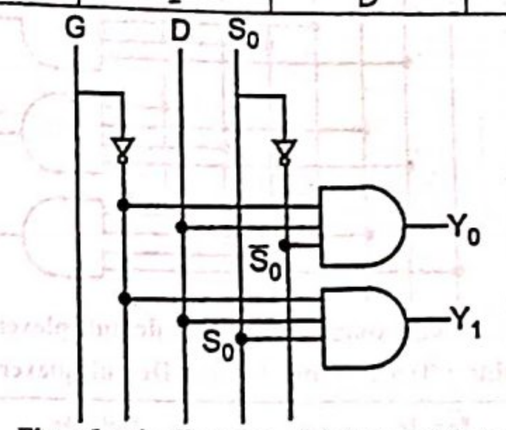


Fig. : Logic diagram of 1:2 demultiplexer

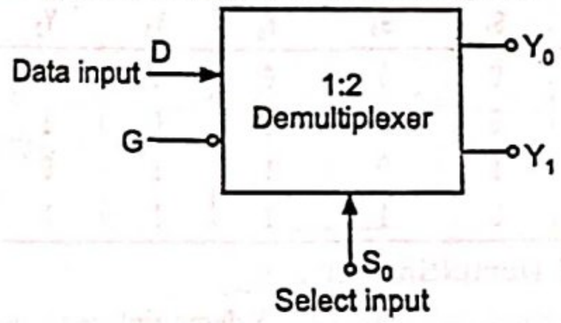


Fig. : Block diagram of 1:2 demultiplexer

(2) 1:4 Demultiplexer

- The block diagram of a 1:4 demultiplexer is as shown in Fig. It has only one data input D, two select inputs, one strobe G (or enable E) input and four outputs Y_0, Y_1, Y_2 and Y_3. The strobe G input may be active low (0) or active high (1) and it is used, for cascading. But the strobe (G) input is normally active low (0).
- The truth table of a 1:4 demultiplexer is as shown in Table. From this table it is clear that D is connected to Y_0 when $S_1S_0 = 00$, it is connected to Y_1 when $S_1S_0 = 01$ and so on.
- The other outputs will remain 0. Here $\bar{G} = 0$. The strobe G input needs to be low i.e. $\bar{G} = 0$ in order to enable the demultiplexer.
- If G = 1, then all the outputs will be 0, irrespective of any data input and select inputs.

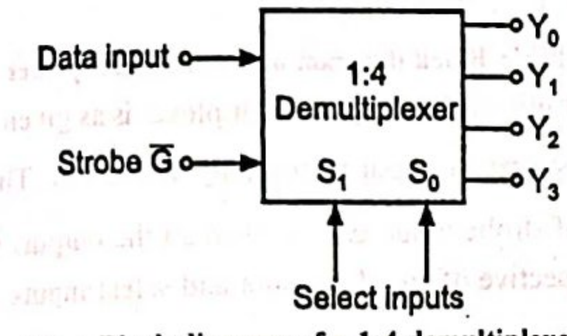


Fig. : Block diagram of a 1:4 demultiplexer

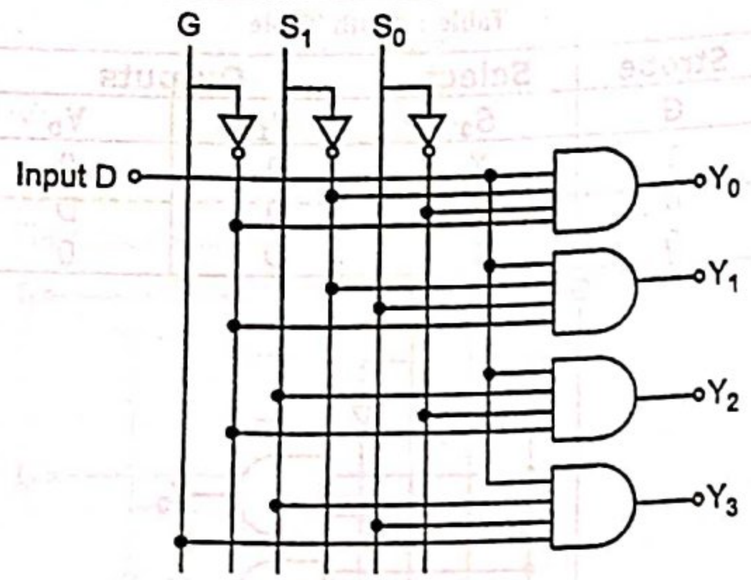


Fig. : Logic diagram of a 1:4 demultiplexer

Table : Truth Table for 1:4 Demultiplexer

Strobe G	Inputs		Outputs			
	S_1	S_0	Y_0	Y_1	Y_2	Y_3
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

(3) 1:8 Demultiplexer :

- The block diagram of a 1:8 demultiplexer is as shown in Fig. It has only one data input D, three select inputs, one strobe G or enable E input and eight outputs Y_0 through Y_7.

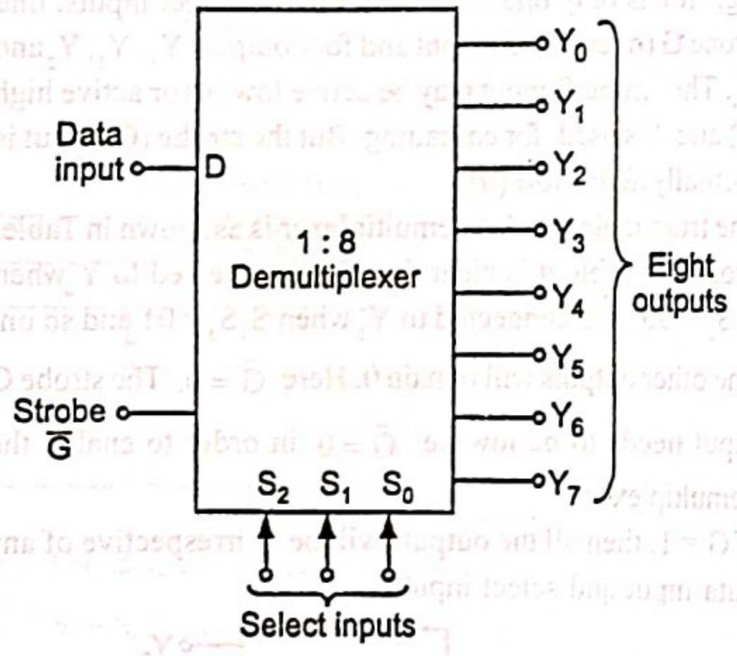


Fig. : Block diagram of a 1:8 demultiplexer

- The truth table of a 1:8 demultiplexer is as given in Table. The strobe $\overline{G}$ input is normally active low. This shows that if strobe input $\overline{G}$ = 1, then all the outputs will be 0 irrespective of any data input and select inputs.

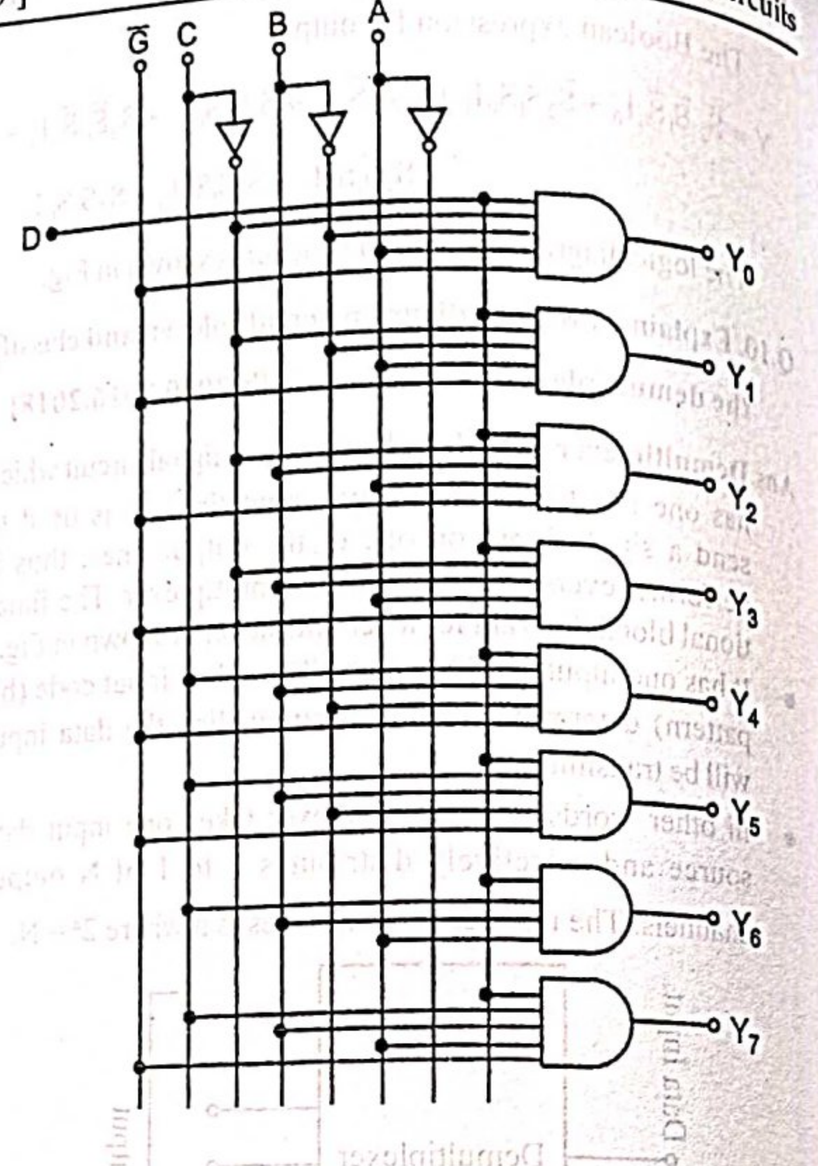


Fig. : Logic diagram of a 1:8 demultiplexer

Table : Truth Table for 1:8 Demultiplexer

Strobe G	Select inputs			Output Y
	S_2	S_1	S_0	
0	0	0	0	Y_0
0	0	0	1	Y_1
0	0	–	–	Y_2
0	–	1	–	Y_3
0	1	0	0	Y_4
0	1	0	1	Y_5
0	1	1	0	Y_6
0	1	1	1	Y_7

Q.11. Explain demultiplexer as decoder ?

Ans. Demultiplexer as Decoder :

- Demultiplexers are commonly called decoders. Demultiplexers consist of 1 input data line. Input data line is not present in decoder. Decoders are the code converters. They consist of number of input lines and number of output lines.
- For example, 1:4 demultiplexer and 2:4 decoder.

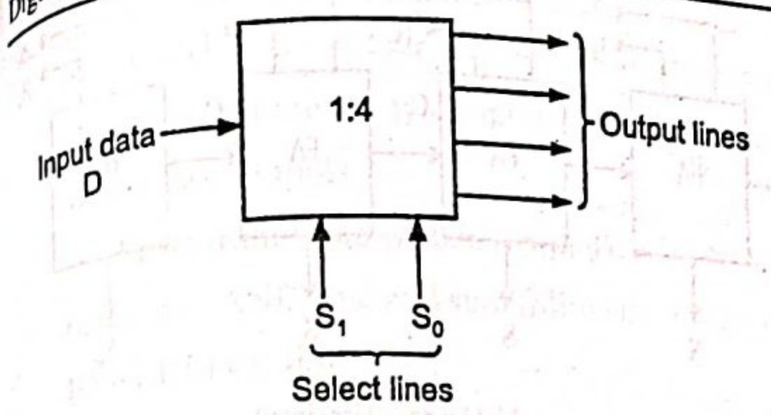


Fig. : demultiplexer

Example 1 : Implement a 1:4 demultiplexer using 1:2 demultiplexers.

Solution:

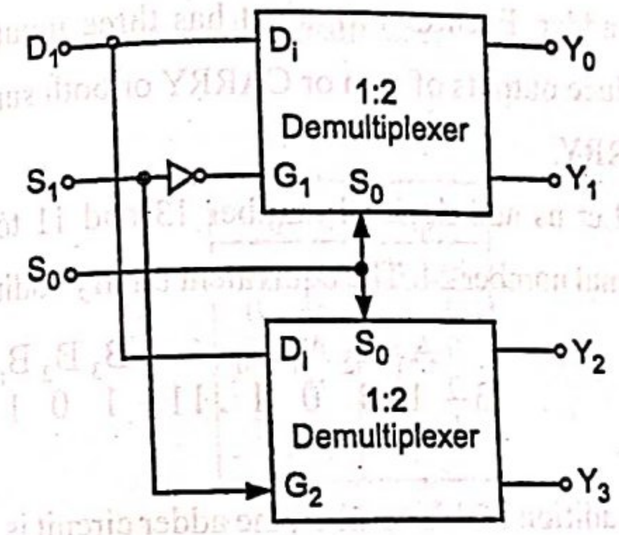


Fig. : Demultiplexer using two 1:2 demultiplexers

Example 2 : Implement a 1:16 demultiplexer using 1:8 demultiplexers.

Solution :

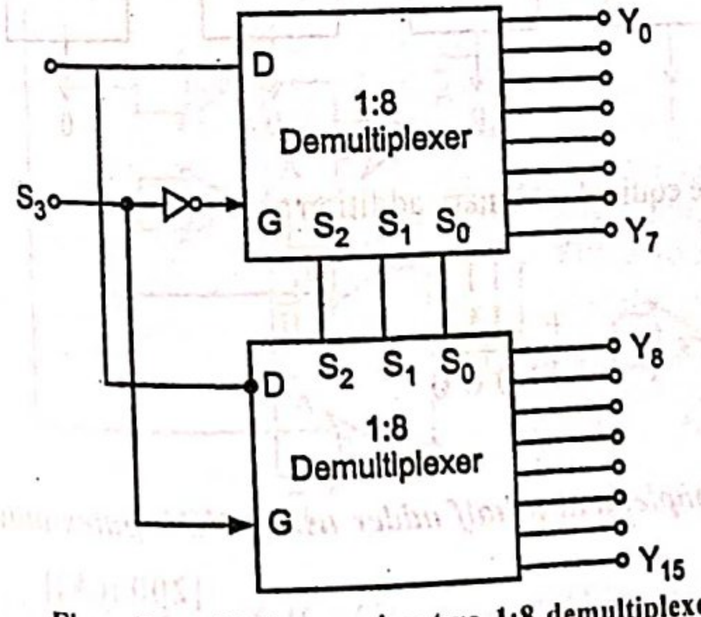


Fig. : Demultiplexer using two 1:8 demultiplexers

Example 3: Implement a 1:16 demultiplexer using 1:4 demultiplexers.

Solution :

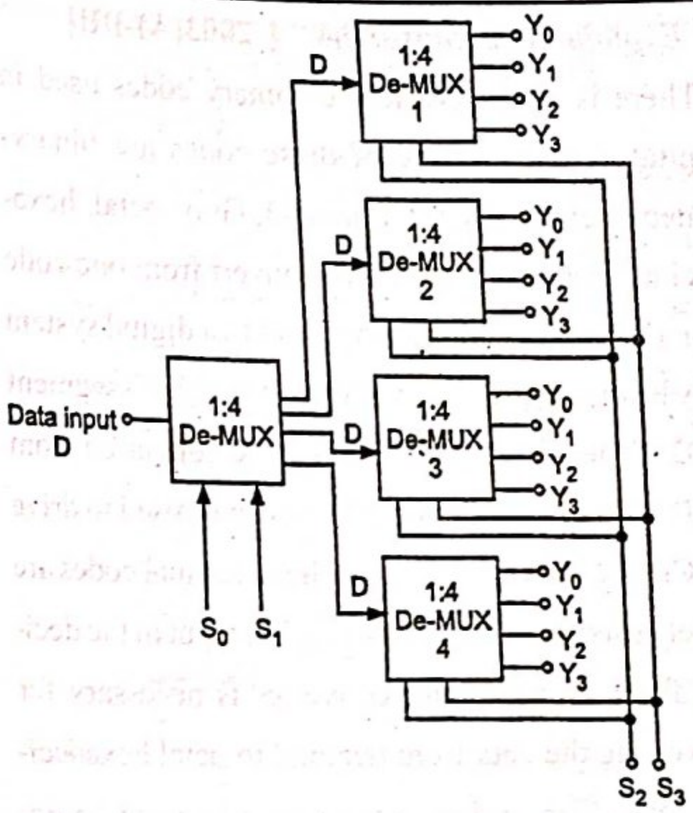


Fig. : demultiplexer using four 1:4 demultiplexers

Example 4 : Implement a 1:32 demultiplexer using 1:8 demultiplexers.

Solution :

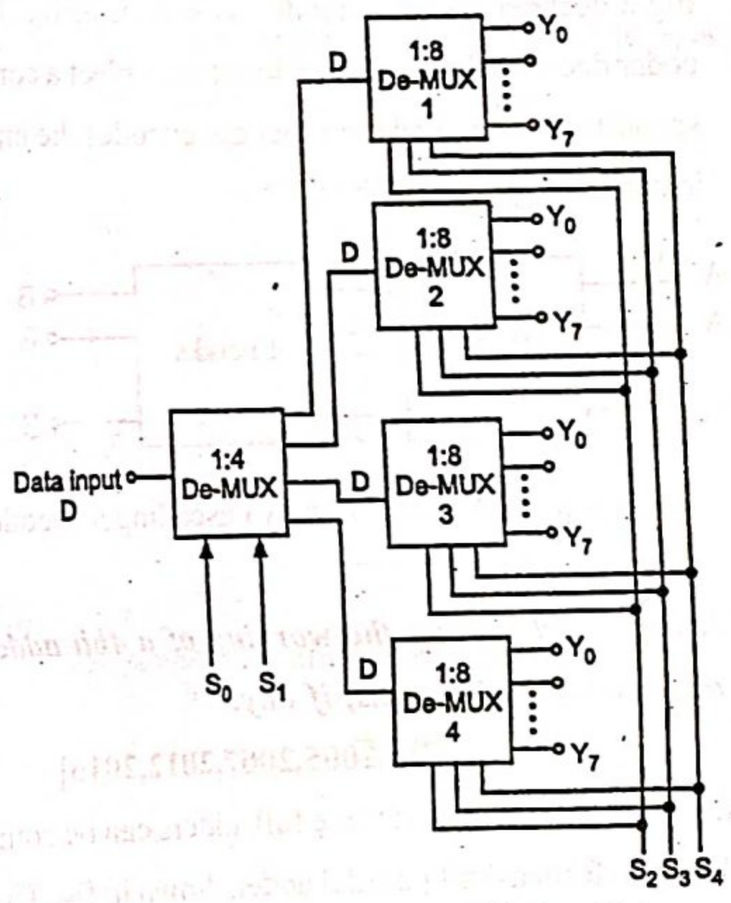
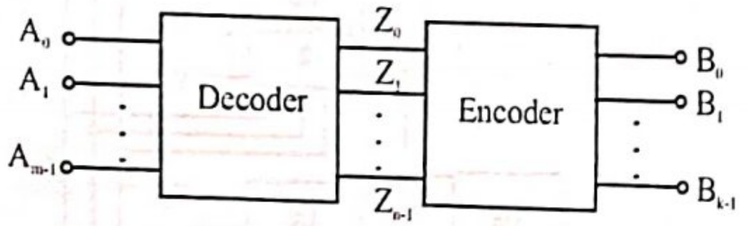


Fig. : demultiplexer using four 31:8 demultiplexers

Q.12. ***Explain code converters.*** **[2003(A)-BR]**

Ans. There is a wide variety of binary codes used in digital systems. Some of these codes are binary-coded-decimal (BCD), Excess-3, Gray, octal, hexa-decimal etc. It is required to convert from one code to another. For example, the input to a digital system may be in natural BCD and output may be 7-segment LEDs. Therefore, the data has to be converted from BCD to 7-segment code before it can be used to drive the CED's. Similary, octal and hexadecimal codes are widely used in microprocessors, but input in the decimal form. Hence coder converter is necessary for converting the data from decimal to octal hexadecimal form. The various code converters can be designed using gates, multiplexer or demultiplexers.

A code converter can be constructed by cascading a decoder and an encoder as shown in fig. Decoder decodes the particular input and select a corresponding output. And then encoder encodes the input in particular pattern of bits.

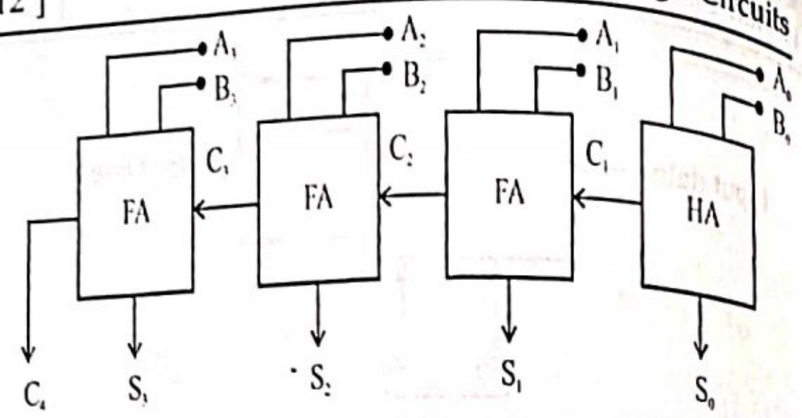


A code converter assembled by cascading a decoder and an encoder.

Q.13. ***Draw and explain the working of a 4bit adder using suitable diagrams, if any.***

[Bh.2005,2007,2012,2016]

Ans. One half adder and three full adders can be combined to form a 4 bit parallel adder shown in fig. The number being added are A_3, A_2, A_1, A_0 and B_3, B_2, B_1, B_0. A_3 and B_3 are the most significant digits while A_0 and B_0 are the least significant digits of these two numbers.



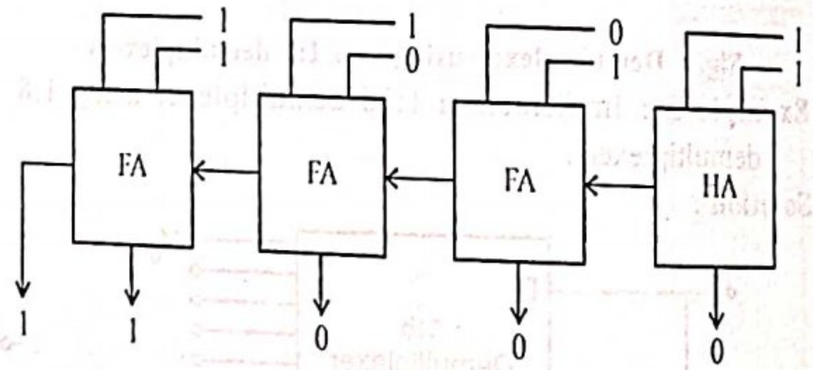
4-bit binary adder circuit

There is no any carry for LSB, hence HA is used for LSB. For other 3 bits we need full adders. The half adder has two inputs A_0, B_0 and produces outputs of either sum S_0 or CARRY C_1. CARRY is fed to next full adder. Each of full addet has three inputs and produce outputs of sum or CARRY or both sum and CARRY.

Let us add decimal number 13 and 11 to give decimal number 24. The equivalent binary addition is

$$\begin{array}{r} A_3\,A_2\,A_1\,A_0 \quad B_3\,B_2\,B_1\,B_0 \\ 13 - 1\;1\;0\;1\;,\;11 - 1\;0\;1\;1 \end{array}$$

For addition of this number, the adder circuit is



The equivalent binary addition as

$$\begin{array}{r} 1\;1\;1\;1 \\ +\;1\;1\;1\;1 \\ \hline 1\;1\;0\;0\;0 \end{array}$$

Q.14. ***Implement a half adder using NOR gates only.***

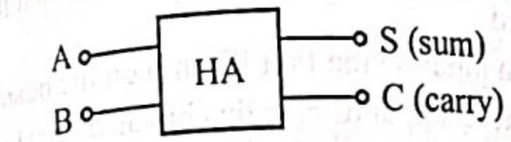
[2004(A)]

Ans. A logic circuit for the addition of two one-bit numbers is referred to as an half-adder. The symbol and truth table of Half-adder is shown in fig. (a) and (b). Here A and B are the two inputs and sum and carry are the two outputs.

From truth-table we can say that carry obeys the funcion of AND gate and sum follow the function of EXCLUSIVE -OR gate.

sum, $S = \overline{A}B + A\overline{B} = A \oplus B$

carry, $C = A \,.\, B.$

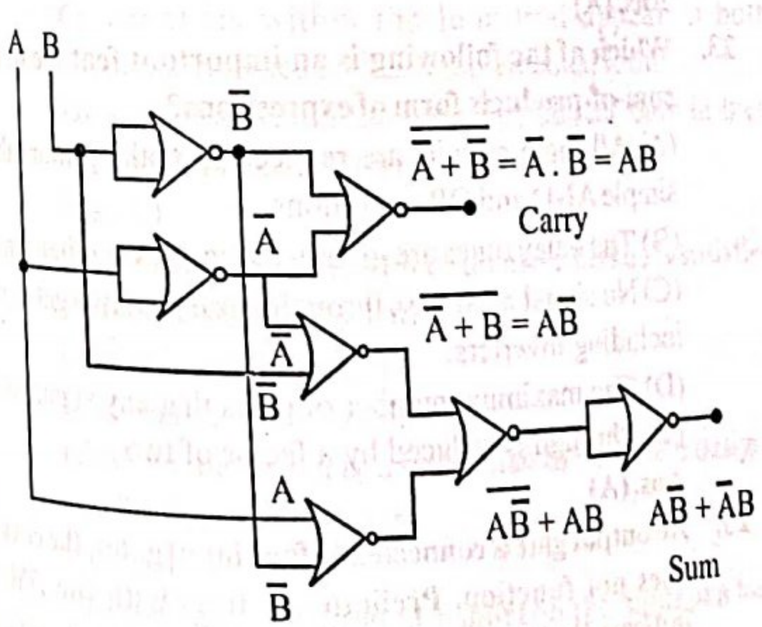


(a) symbol for a half-adder

Inputs		Outputs	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

(b) truth table

Implementation of an half-adder, using NOR gate is shown in fig. (c)

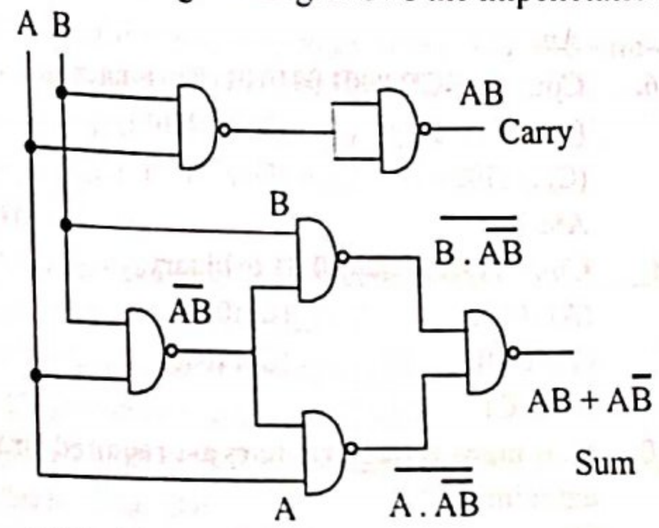


Q.15. ***Realize Half-adder using NAND gate only.***

Ans. For description see Imperientation Half-adder using NOR gate only.

Realization of an half adder using NAND gate is shown in fig.

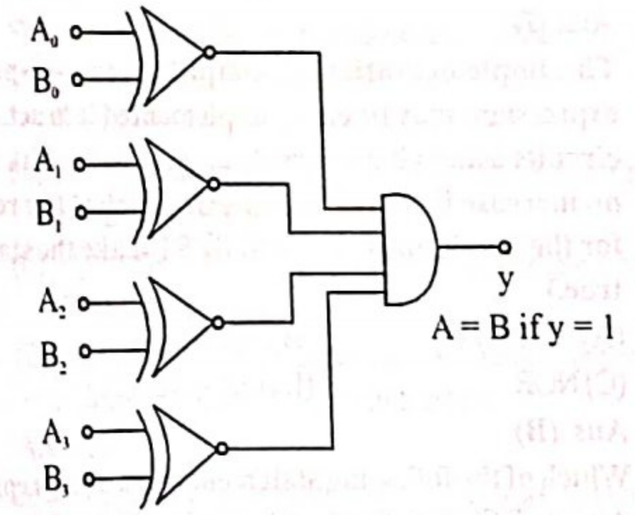
To implement a half-adder circuit using NAND gates only, it is necessary to replace the XOR and AND gates by NAND gates. Fig. shows the impenetation.



Half-adder circuit using only NAND gate.

Q.16. ***Draw the logic circuit for 4 bit comparator.***

Ans. Figure shows the logic circuit for 4 bit comparator. Here 4 XNOR gates is used. Its output indicate one (high) if both numbers are equal.



Magnitude comparator for two 4 bit numbers.

□□□

OBJECTIVE TYPE QUESTIONS

1. How many 3-line-to-8-line decoders are required for a 1-of-32 decoder?
(A) 1 (B) 2
(C) 4 (D) 8
Ans. (C)

6. Convert BCD 0001 0010 0110 to binary.
(A) 1111110 (B) 1111101
(C) 1111000 (D) 1111111
Ans. (A)

8. Convert BCD 0001 0111 to binary.
(A) 10101 (B) 10010
(C) 10001 (D) 11000
Ans. (C)

10. How many data select lines are required for selecting eight inputs?
(A) 1 (B) 2
(C) 3 (D) 4
Ans. (C)

12. How many 1-of-16 decoders are required for decoding a 7-bit binary number?
(A) 5 (B) 6
(C) 7 (D) 8
Ans. (D)

14. The implementation of simplified sum-of-products expressions may be easily implemented into actual logic circuits using all universal ..... gates with little or no increase in circuit complexity. (Select the response for the blank space that will BEST make the statement true.)
(A) AND/OR (B) NAND
(C) NOR (D) OR/AND
Ans. (B)

15. Which of the following statements accurately represents the two BEST methods of logic circuit simplification?
(A) Boolean algebra and Karnaugh mapping
(B) Karnaugh mapping and circuit waveform analysis
(C) Actual circuit trial and error evaluation and waveform analysis
(D) Boolean algebra and actual circuit trial and error evaluation
Ans. (A)

17. Which of the following combinations cannot be combined into K-map groups?
(A) Corners in the same row
(B) Corners in the same column
(C) Diagonal corners
(D) Overlapping combinations
Ans. (C)

18. As a technician you are confronted with a TTL circuit board containing dozens of IC chips. You have taken several readings at numerous IC chips, but the readings are inconclusive because of their erratic nature. Of the possible faults listed, select the one that most probably is causing the problem.
(A) A defective IC chip that is drawing excessive current from the power supply
(B) A solar bridge between the inputs on the first IC chip on the board
(C) An open input on the first IC chip on the board
(D) A defective output IC chip that has an internal open to Vcc
Ans. (C)

19. Which gate is best used as a basic comparator?
(A) NOR (B) OR
(C) Exclusive-OR (D) AND
Ans. (C)

21. In VHDL, macrofunctions is/are:
(A) digital circuits.
(B) analog circuits.
(C) a set of bit vectors.
(D) preprogrammed TTL devices.
Ans. (D)

22. Which of the following expressions is in the product-of-sums form ?
(A) (A + B)(C + D) (B) (AB)(CD)
(C) AB(CD) (D) AB + CD
Ans. (A)

23. Which of the following is an important feature of the sum-of-products form of expressions?
(A) All logic circuits are reduced to nothing more than simple AND and OR operations.
(B) The delay times are greatly reduced over other forms.
(C) No signal must pass through more than two gates, not including inverters.
(D) The maximum number of gates that any signal must pass through is reduced by a factor of two.
Ans. (A)

25. An output gate is connected to four input gates; the circuit does not function. Preliminary tests with the DMM indicate that the power is applied; scope tests show that the primary input gate has a pulsing signal, while the interconnecting node has no signal. The four load gates are all on different ICs. Which instrument will best help isolate the problem?
(A) Current tracer (B) Logic probe
(C) Oscilloscope (D) Logic analyzer
Ans. (A)

26. The binary numbers A = 1100 and B = 1001 are applied to the inputs of a comparator. What are the output levels?
(A) A > B = 1, A < B = 0, A < B = 1
(B) A > B = 0, A < B = 1, A = B = 0
(C) A > B = 1, A < B = 0, A = B = 0
(D) A > B = 0, A < B = 1, A = B = 1
Ans. (C)

27. A logic probe is placed on the output of a gate and the display indicator is dim. A pulser is used on each of the input terminals, but the output indication does not change. What is wrong?
(A) The output of the gate appears to be open.
(B) The dim indication on the logic probe indicates that the supply voltage is probably low.
(C) The dim indication is a result of a bad ground connection on the logic probe.
(D) The gate may be a tristate device.
Ans. (A)

29. Each "1" entry in a K-map square represents:
(A) a HIGH for each input truth table condition that produces a HIGH output.
(B) a HIGH output on the truth table for all LOW input combinations.
(C) a LOW output for all possible HIGH input conditions.
(D) a DON'T CARE condition for all possible input truth table combinations.
Ans. (A)

30. Looping on a K-map always results in the elimination of:
(A) variables within the loop that appear only in their complemented form.
(B) variables that remain unchanged within the loop.
(C) variables within the loop that appear in both complemented and uncomplemented form.
(D) variables within the loop that appear only in their uncomplemented form.
Ans. (C)

31. What will a design engineer do after he/she is satisfied that the design will work?
(A) Put it in a flow chart
(B) Program a chip and test it
(C) Give the design to a technician to verify the design
(D) Perform a vector test
Ans. (B)

33. What is the indication of a short on the input of a load gate?
(A) Only the output of the defective gate is affected.
(B) There is a signal loss to all gates on the node.
(C) The affected node will be stuck in the LOW state.
(D) There is a signal loss to all gates on the node, and the affected node will be stuck in the LOW state.
Ans. (D)

34. In HDL, LITERALS is/are:
(A) digital systems.
(B) scalars.
(C) binary coded decimals.
(D) a numbering system.
Ans. (B)

35. Which of the following expressions is in the sum-of-products form ?
(A) (A + B)(C + D) (B) (AB)(CD)
(C) AB(CD) (D) AB + CD
Ans. (D)

38. A decoder can be used as a demultiplexer by ..... .
(A) tying all enable pins LOW
(B) tying all data-select lines LOW
(C) tying all data-select lines HIGH
(D) using the input lines for data selection and an enable line for data input
Ans. (D)

9. How many 4-bit parallel adders would be required to add two binary numbers each representing decimal numbers up through 30010 ?
(A) 1 (B) 2
(C) 3 (D) 4
Ans. (C)

40. Which statement below best describes a Karnaugh map?
(A) A Karnaugh map can be used to replace Boolean rules.
(B) The Karnaugh map eliminates the need for using AND and NOR gates.
(C) Variable complements can be eliminated by using Karnaugh maps.
(D) Karnaugh maps provide a visual approach to simplifying Boolean expressions.
Ans. (D)

2. A certain BCD-to-decimal decoder has active-HIGH inputs and active-LOW outputs. Which output goes LOW when the inputs are 1001?
(A) 0 (B) 3
(C) 9 (D) None. All outputs are HIGH.
Ans. (C)

47. The design concept of using building blocks of circuits in a PLD program is called a(n):
(A) hierarchical design.
(B) architectural design.
(C) digital design.
(D) verilog.
Ans. (A)

□□□

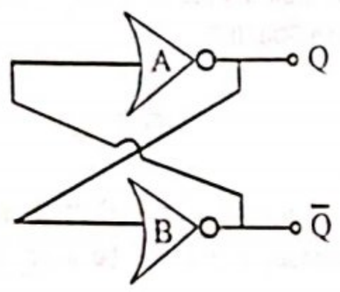
Chapter 4

Sequential Logic Circuit

Q.1 *What is Flip-Flop? Explain some of its applications.*

[Bh.2014,2016,2019]

Ans. A Flip-Flop is a basic memory element used to store one bit of information. It has two stable states which are either low or high. It can be obtained by using NAND or NOR gates. Most elementary form of a flip-flop is knwn as a batch. Since information is locked or latched in this circuit, this circuit is known as a latch. This is shown in fig. below



In the latch circuits shown in fig. above, there is no possibility of entering in it. When the power is switched ON, the circuit switches to one of the stable states (Q = 1 or 0) and it is not position to predict the state. By some modification in latch circuit ddifferent types of Flip-Flop are to be made.

Some application of Flip-Flop are:

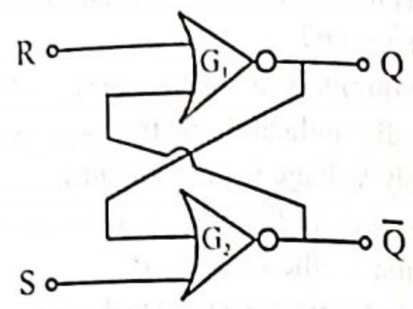
(i) Used in storage circuit

(ii) Counter circuit

(iii) Shift register and

(iv) Computer applications etc.

Flip-Flop as a memory element :- Some modification is made in latch circuit to behave as memory element. There are two inputs S (set) and R (reset). Let as consider four combinations of the input signals R and S.



Case-(i) When S = R = 0

In this case, the control inputs have no control over it. The gate o/p depend only on the logic levels present at the other input. If means Flip-Flop simply remains in its present state.

Case-(ii) When S = 1 and R = 0 with output $\overline{Q} = 0$.

In this case both input of get G_1 is low, hence out will high and both input of gate G_2 is high and hence o/p will be low. This state is called set state.

Case-(iii) When S = 0 and R = 1 with o/p Q = 0.

Since signals at both inputs of gate G_1 are high and signals of both inputs of gate G_2 are low, therefore, output Q of gate G_1 is low and o/p $\overline{Q}$ of gate G_2 is high. This is known as reset state.

Case(iv) When R = 1 and S = 1

In this case, the input signals R and S of the flip-flop forces the o/p both NOR gates tobe zero, which is not allowed and therefore this input condition is prohibited. Truth table is shown in table.

Truth-table for R-S Flip-Flop.

Truth - table for R-S Flip-Flop

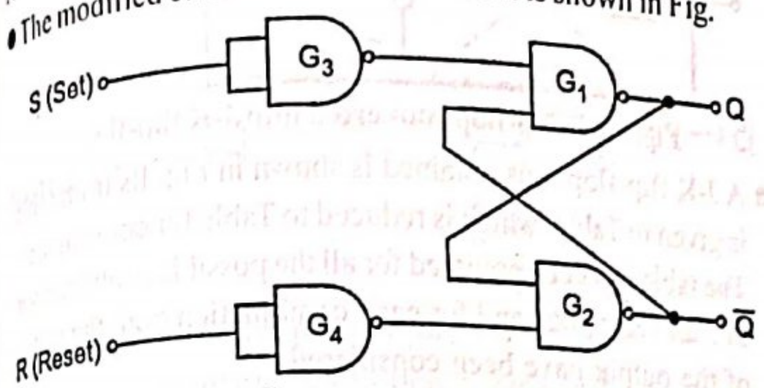
R	S	Q	Action
0	0	last value	no change
0	1	1	set
1	0	0	reset
1	1	2	forbidden

Q.11 *Draw and explain the working principle of a clocked S-R flip-flop and give the truth table.* **[Bh.2015,2018]**

Or *What is clocked flip-flop ? Explain the principle of operation of clocked S-R flip-flop with truth table.*

Ans. SR Flip-Flop :

- The modified circuit of NAND SR latch is shown in Fig.



- The truth table details are same as that of NAND SR latch.

Clocked SR Flip-Flop :

- It is often required to set or reset the memory cell in synchronism with a train of pulses known as clock.
- Such a circuit is shown in Fig. and is referred to as a clocked set-reset (S-R) flip-flop.
- In this circuit, if a clock pulse is present (clk = 1), its operation is exactly the same as that of Fig.
- On the other hand, when the clock pulse is not present (clk = 0), the gates G_3 and G_4 are inhibited i.e. their outputs are 1 irrespective of the values of S or R.
- In other words, the circuit responds to the inputs S and R only when the clock is present.

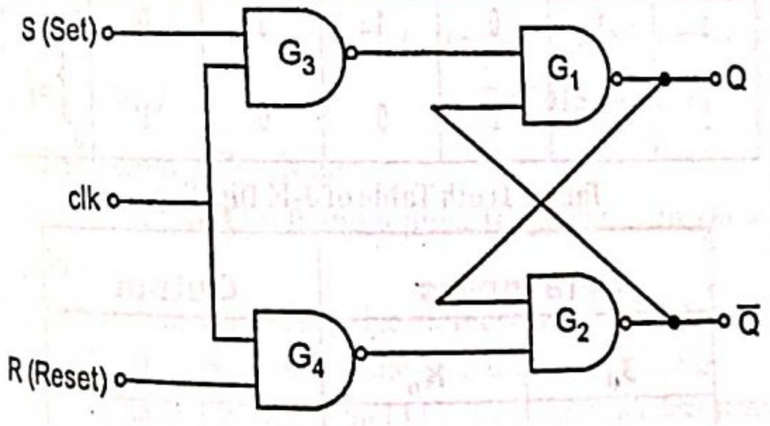


Fig. : Clocked S-R flip-flop

Table : Truth Table of S-R flip-flop

clk	Inputs		Output
	S	R	Q_{n+1}
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	?

- When the clock is present, the truth table of clocked S-R flip-flop is same as that of S-R flip-flop.

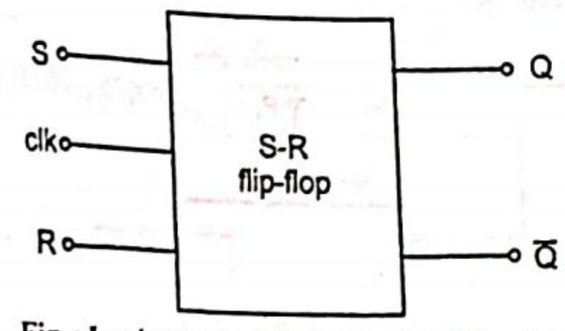


Fig. : Logic symbol of clocked S-R flip-flop

- The logic symbol of clocked S-R flip-flop is shown in Fig.

Q.3. Draw and explain the working principle of a Clocked SR Flip-Flop with Preset and Clear ?

Or Define SR Flip-Flop and write its drawback ?

Ans. In the flip-flop when the power is switched ON, the state of the circuit is uncertain.

- It may come to set (Q = 1) or reset (Q = 0) state.
- In many applications it is desired to initially set or reset the flip-flop i.e. the initial state of the flip-flop is to be assigned. This is accomplished by using preset (P_r) and clear (C_r) inputs.
- These inputs may be applied at any time between clock pulses and are not in synchronism with the clock. An S-R flip-flop with preset and clear is shown in Fig.

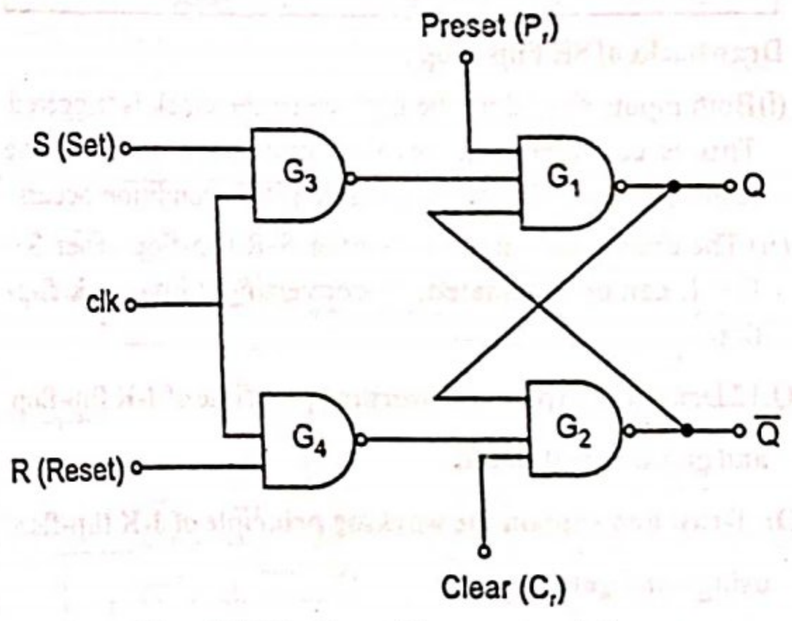


Fig. : S-R flip-flop with preset and clear

- If $P_r = C_r = 1$ the circuit operates in accordance with the truth table of S-R flip-flop given in Table.
- If $P_r = 0$ and $C_r = 1$, the output of G_1 (Q) will certainly be 1.
- Consequently all the three inputs to G_2 will be 1 which will make $\overline{Q} = 0$. Hence making $P_r = 0$ sets the flip-flop.
- Similarly if $P_r = 1$ and $C_r = 0$ then the flip-flop is reset.
- The condition $P_r = C_r = 0$ must not be used, since this leads to an uncertain state.

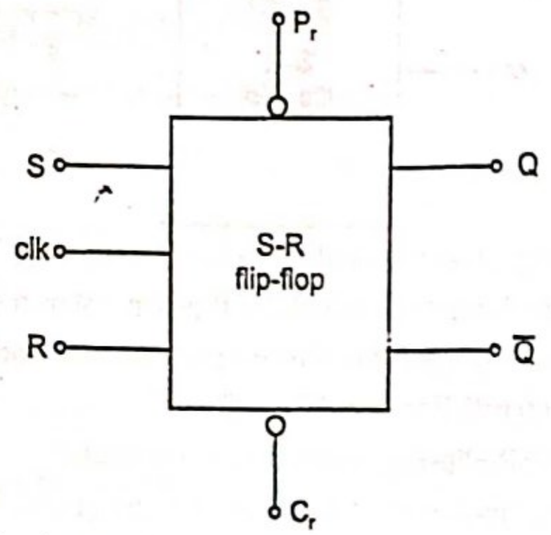


Fig. : Logic symbol of clocked SR flip-flop with preset and clear

- In the logic symbol, bubbles are used for P_r and C_r inputs which means these are active low.

Table : Truth Table

Inputs			Output	Operation
clk	C_r	P_r	Q	performed
0	0	1	0	Clear
0	1	0	1	Preset

Drawbacks of SR Flip-Flop :

(i) Both inputs should not be high when the clock is triggered. This is considered an invalid input condition, and the resulting output is not predictable if this condition occurs.

(ii) The uncertainty in the state of an S-R flip-flop, when S = R = 1, can be eliminated, by converting it into a J-K flip-flop.

Q.12 Draw and explain the working principle of J-K flip-flop and give the truth-table.

Or Draw and explain the working principle of J-K flip-flop using nand gate ?.

Ans.JK Flip-Flop : The uncertainty in the state of an S-R flip-flop when $S_n = R_n = 1$ can be eliminated by converting it into a J-K flip-flop. The data inputs are J and K which are ANDed with $\bar{Q}$ and Q respectively, to obtain S and R inputs i.e.

$$S = J \cdot \bar{Q}$$

$$R = K \cdot Q$$

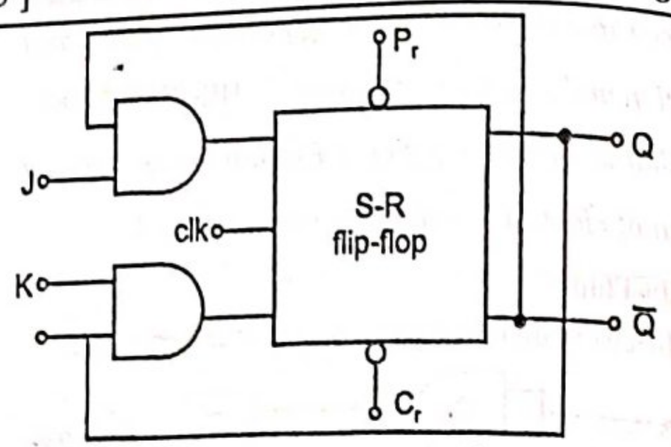


Fig. : S-R flip-flop converted into J-K flip-flop

- A J-K flip-flop thus obtained is shown in Fig. Its truth table is given in Table, which is reduced to Table for convenience. The table has been prepared for all the possible combinations of J and K inputs, and for each combination both the states of the output have been considered.

Table : Truth Table for Fig.

Data inputs		Outputs		Inputs to SR FF		Output	
J_n	K_n	Q_n	$\bar{Q}_n$	S_n	R_n	Q_{n+1}	
0	0	0	1	0	0	0	$= Q_n$
0	0	1	0	0	0	1	
0	1	0	1	0	0	0	$= 0$
0	1	1	0	0	1	0	
1	0	0	1	1	0	1	$= 1$
1	0	1	0	0	0	1	
1	1	0	1	1	0	1	$= \bar{Q}_n$
1	1	1	0	0	1	0	

Table : Truth Table of J-K flip-flop

Data inputs		Output
J_n	K_n	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\bar{Q}_n$

- We obtain J-K flip-flop using NAND gates as shown in Fig.

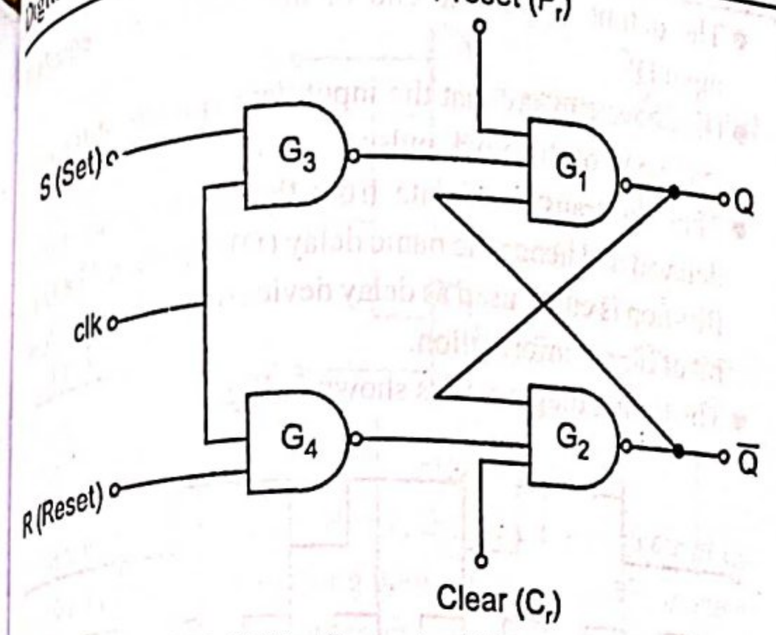


Fig. : J-K flip-flop using NAND gates

- The logic symbol of J-K flip-flop is shown in Fig.

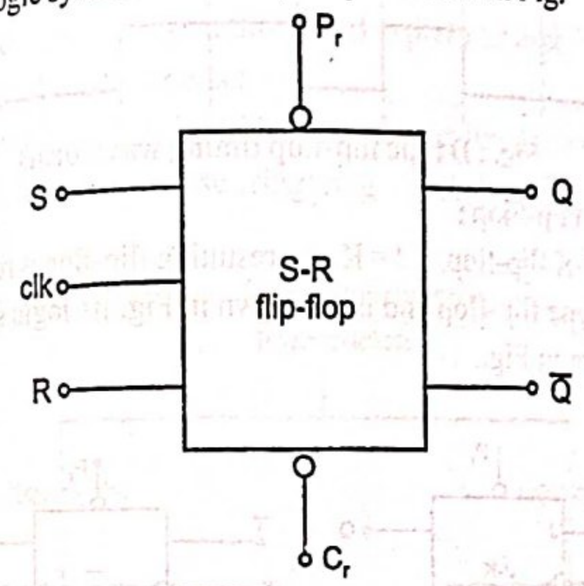
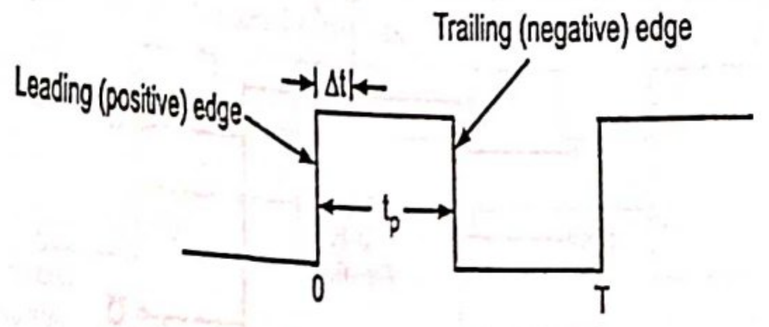


Fig. : Logic symbol of J-K flip-flop

Q.13 *What do you mean by race-around condition? Explain in brief.* **[Bh.2015,2017,2019]**

Ans. Race Around Condition :

- If J = K = 1 and Q = 0 and a pulse is applied at the clock input :
- After a time interval At equal to the propagation delay through two NAND gates in series, the output will change to Q = 1. Now we have J = K = 1 and Q = 1 and after another time interval of Δt the output will change back to Q = 0. Hence for the duration t_p of the clock pulse, the output will oscillate back and forth between 0 and 1.
- At the end of clock pulse, the value of Q is uncertain. This situation is referred to as the race-around condition.
- The race-around condition can be avoided if $t_p < \Delta t < T$. However, it may be difficult to satisfy the inequality because of very small propagation delays in it.
- A more practical method for overcoming this difficulty is the use of the master-slave (M-S) configuration.



Q.14 *Draw the circuit diagram of master-slave flip-flop and explain its working.* **[Bh.2014,2016,2019]**

Ans. Master-Slave JK Flip-Flop

- A master slave JK flip-flop is a cascade of two JK flip-flops, with feedback from the output of the second to the inputs of the first. Positive clock pulses are applied to the first flip-flop and clock pulses are inverted before these are applied to the second flip-flop.
- When clk = 1 the first flip-flop is enabled and the outputs Q_m and $\bar{Q}_m$ respond to the inputs J and K.
- At the same time, the second flip-flop is inhibited. When clk = 0, the second flip-flop is enabled and the first flip-flop is inhibited. Therefore the outputs Q and $\bar{Q}$ follow the outputs Q_M and $\bar{Q}_M$.
- Since the second flip-flop simply follows the first one, it is referred to as the slave and the first one as the master. Hence the configuration is referred as master-slave (M-S) flip-flop.

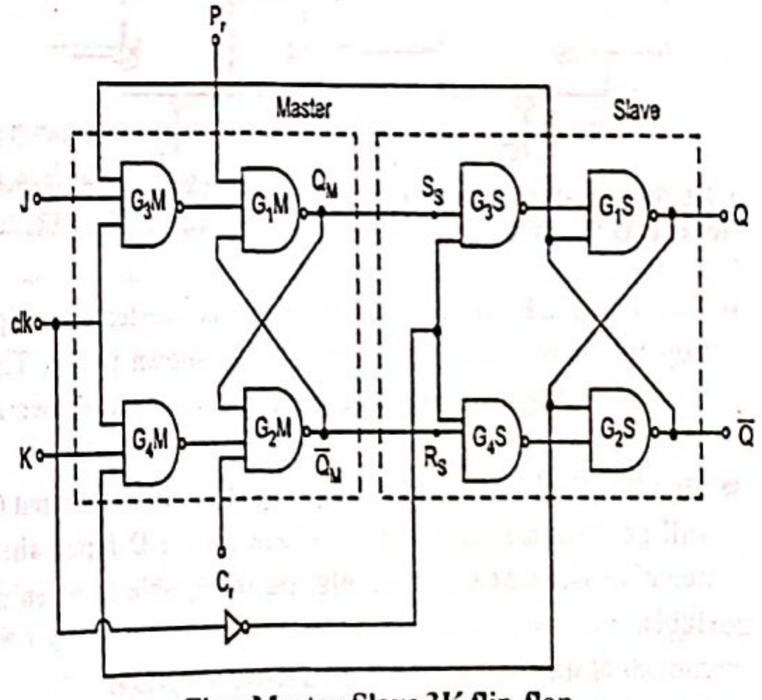


Fig. : Master-Slave 3K flip-flop

- The logic symbol of a M-S J-K flip-flop is shown in Fig.

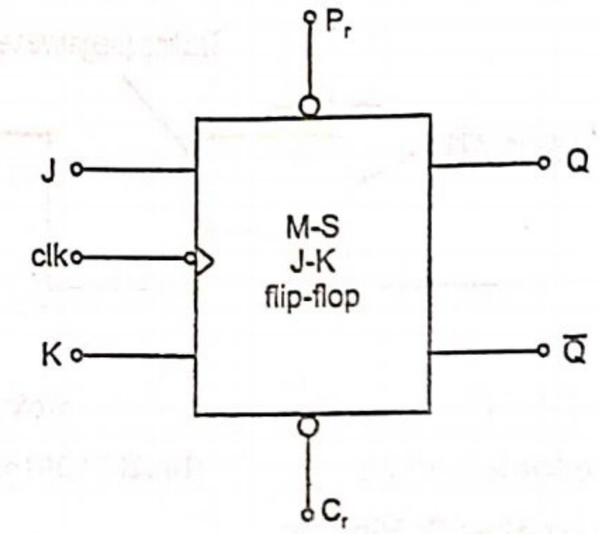
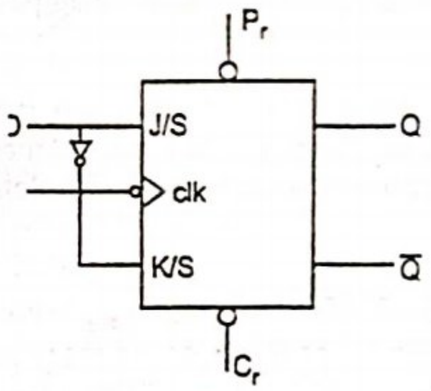


Fig. : Master-Slave JK flip-flop logic symbol

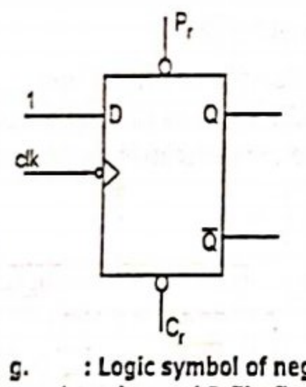
- In this circuit, the inputs to the'gates G_3M and G_4M do not change during the clock pulse. therefore the race-around condition does not exist. The state of the master-slave flip-flop changes at the negative transition of the clock pulse.

Q.16 *Draw the logic symbols and explain the operation of a D - type and a T - type flip-flop.*

Ans. **D-Type Flip-Flop :**



: JK SR flip-flop converted into a D flip-flop



g. : Logic symbol of negative edge triggered D flip-flop

- The SR and JK flip-flop can be easily converted to D flip-flop by simply addition of inverter as shown in Fig. The symbol of negative edge triggered D flip-flop is shown in Fig.
- This flip-flop has only one input i.e. D input. The output Q will go to same state that is present on the D input when negative edge of clock occurs. Its truth table is given in Table.

Table : Truth Table for a D-type flip-flop

clk	Input D_n	Output Q_{n+1}
↓	0	0
↓	1	1

- The output Q_{n+1} at the end of the clock pulse equals the input D_n.
- Hence we can say that the input data appears at the output at the end of the clock pulse.
- Thus the transfer of data from the input to the output is delayed and hence the name delay (D) flip-flop. The D-type flip-flop is either used as delay device or as a latch to store 1-bit of binary information.
- The timing diagram is as shown in Fig.

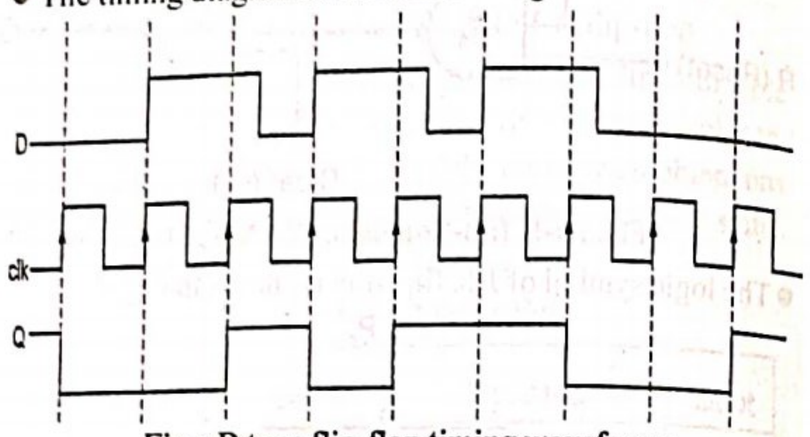
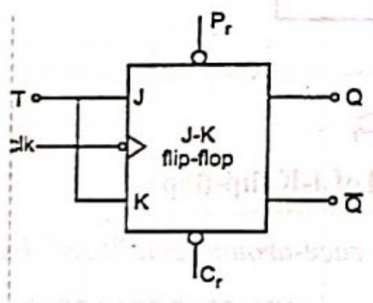


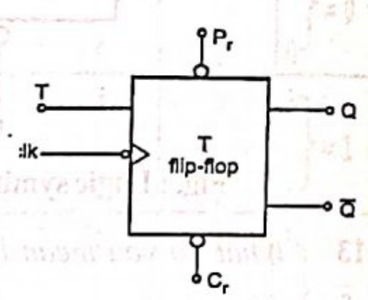
Fig. : D type flip-flop timing waveforms

T-Type Flip-Flop :

- In a J-K flip-flop, if J = K, the resulting flip-flop is referred as a T-type flip-flop and is as shown in Fig. Its logic symbol is shown in Fig.



: JK flip-flop converted into T-type



I. : Logic symbol of negative edge triggered T flip-flop

- It has only one input, referred to as T-input. Its truth table is given in Table.

Table : Truth Table of a T-type flip-flop

clk	Input T_n	Output Q_{n+1}
↓	0	Q_n
↓	1	$\bar{Q}_n$

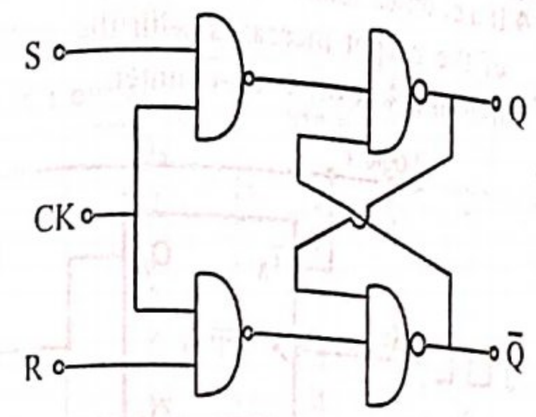
- From above table it is clear that if T = 1, it acts as a toggle switch. For every clock pulse, the output Q changes.
- An S-R flip-flop cannot be converted into T-type flip-flop, since S = R = 1 is not allowed.

Q.7. *Design a S-R flip-flops using NOR gates and draw the truth table. Why S = 1, R = 1 condition is invalid in S-R flip-flop ?*

Ans:

Truth Table of S-R FF

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



(a) S-R flip-flop

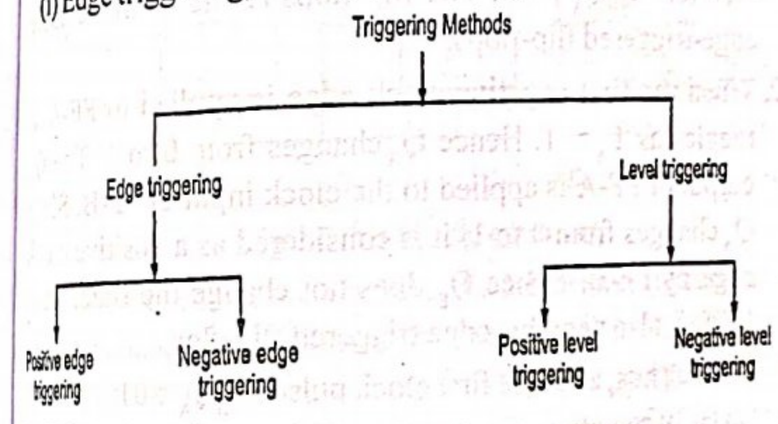
When we give input S = R = 1, the we get a problem at the output i.e., there no toggling between output and its complement i.e., no change or other any output will get. So we cannot use S = R = 1 for the S-R flip-flop.

Q.7. Define triggering methods and explain its type ?

Ans. Triggering Methods :

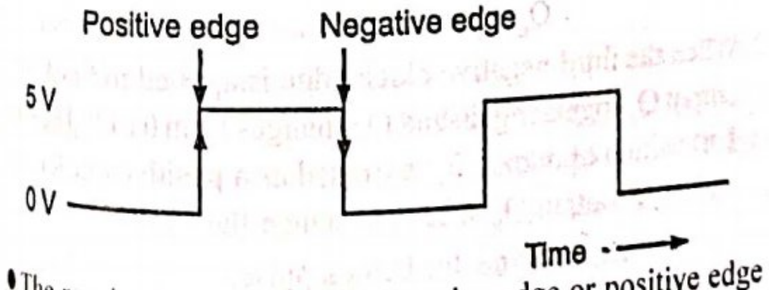
- The result of latches and flip-flops responding to clock input is called as clock pulse triggering which is mainly of two types :

(i) Edge triggering, (ii) Level triggering

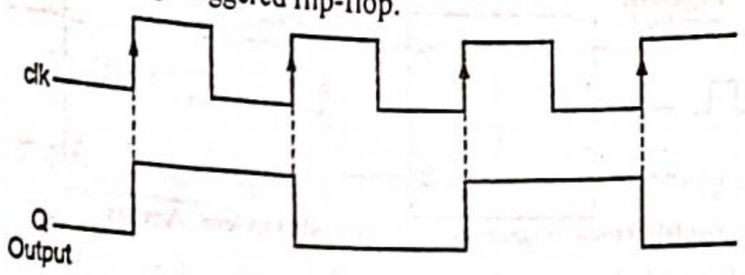


(i) Edge Triggering :

- A typical clock pulse train is shown in Fig.
- A clock pulse can be positive or negative. The pulse goes through two transitions: from 0 to 1 and returns from 1 to 0. Positive transition is defined as positive edge and negative transition is defined as negative edge.



- The result of responding to negative edge or positive edge is called as edge triggering.
- Fig. shows how the output of an edge triggering flip-flop changes to a high on the positive edge of the first clock pulse and back to a low on the positive edge of the second clock pulse. A flip-flop triggered, in this way is known as positive edge triggered flip-flop.



- The other flip-flops respond during negative edge of the clock pulse.
- These are called negative edge triggering flip-flops. Fig. shows how the Q output of such a flip-flop changes low to high on the negative edge of the first clock pulse. It then changes back to low on the next clock pulse negative edge.

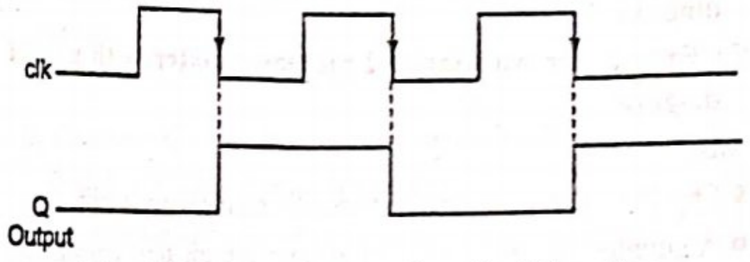
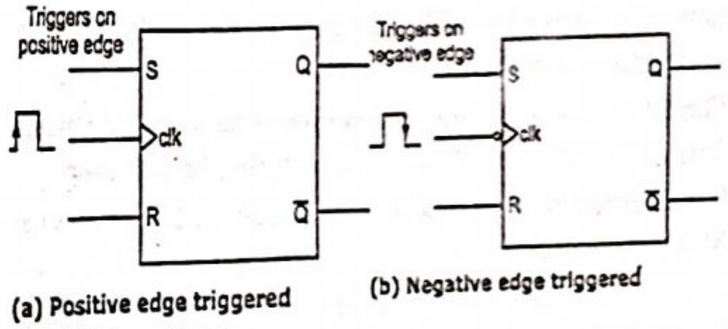


Fig. : Waveforms for negative edge triggering

- Fig. shows two different ways of showing types of edge triggering clock.



(a) Positive edge triggered **(b) Negative edge triggered**

Level Triggering :

- The result of digital circuit responding to the level of clock input is called level triggering. Fig. shows two levels of clock pulse.

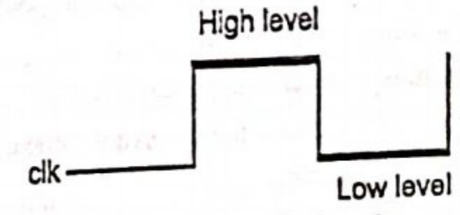


Fig. : Levels of clock pulse

- There are high level triggered latches and low level triggered latches. Data can be transferred from the input to the output of high level triggered latch, while the clock level is high.
- Similarly data is transferred in low level triggered latches while the clock signal is low.
- Fig. shows two different ways of showing types of level triggering clock.

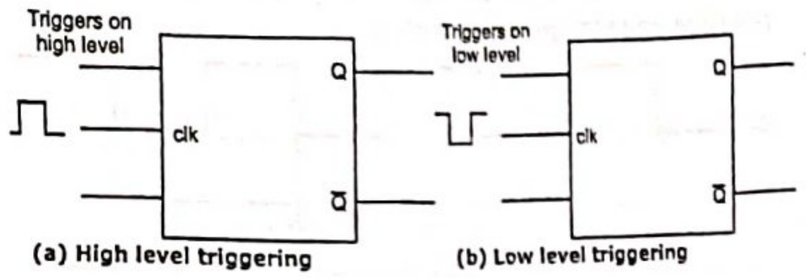


Fig.

Q.6. Define counter. Explain 3-bit up ripple (Asynchronous) counter ?

Or Describe with diagram 3-bit twisted ring counter using D flip-flops.

OrExplain the working of 4 bit ring counter with a neat diagram ?

Or Explain the working of 2 bit ring counter with a neat diagram ?

Ans.

- Flip-flops can be connected serially to form counters.
- A counter can be defined as a logic circuit that counts the number of clock pulses applied at the input. Each count, a binary number is known as the state of the counter. Hence a counter counting in terms of n-bits will have 2^n different states. The number of different states of the counter is known as modulus of a counter.
- The Modulus of a counter represents the number of states through which the counter passes during its operation.
- For example, a 2-bit ripple counter is called a $2^2 = 4$ states i.e. MOD-4 counter.

  a 3-bit ripple counter is called a $2^3 = 8$ states i.e. MOD-8 counter.
- Hence, an n-bit ripple counter is known as modulo-N counter, where MOD number = 2^n.
- A counter circuit consists of flip-flops and combinational elements. Counters can be classified as :

  (i) Asynchronous counter (Ripple counter).

  (ii) Synchronous counter, on the basis of the manner in which the flip-flops are triggered.
- In asynchronous counters the first flip-flop is clocked by the external clock pulse and then each successive flip-flop is clocked by the Q and $\overline{Q}$ output of the previous flip-flop. In asynchronous counter, the flip-flops are not clocked simultaneously.
- In synchronous counters, the clock input is applied to all the flip-flops simultaneously. This increases the speed of operation of synchronous counters.

Asynchronous (Ripple) Counter :

A 2-Bit Asynchronous Up-Counter :

- If a counter counts in such a way that the decimal equivalent of the output increases with the successive clock pulses, then it is known as up-counter.

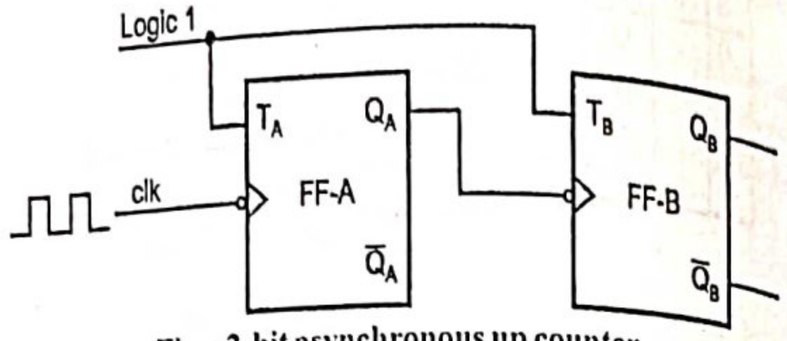


Fig. : 2-bit asynchronous up counter

- Fig. shows the diagram of a 2-bit ripple up counter. A 2-bit counter uses 2 flip-flops. A flip-flop stores 1 bit, therefore, 2-bit counter requires 2 flip-flops and counts $2^2 = 4$ states.
- A toggle T-flip-flop or J-K flip-flop (with J and K inputs connected to logic 1) can be used.
- The clock input is connected to the flip-flop A and the Q_A output of FF-A is connected to the clock input of FF-B.

Counter Operation :

1. Initially, both the outputs Q_A and Q_B are in reset condition, therefore $Q_BQ_A = 00$. The flip-flops A and B are negative edge-triggered flip-flops.
2. When the first negative clock edge is applied to FF-A, it toggles as $T_A = 1$. Hence Q_A changes from 0 to 1. The Q_A output of FF-A is applied to the clock input of FF-B. Since Q_A changes from 0 to 1, it is considered as a positive clock edge by FF-B. Hence Q_B does not change the state, since FF-B is also negative edge triggered flip-flop.

   Thus, after the first clock pulse, $Q_BQ_A = 01$.

   When the second falling clock edge is applied, FF-A toggles again and Q_A changes from 1 to 0. Thus, 1 to 0 transition of the output Q_A is treated as a negative clock edge for FF-B and its output toggles and Q_B changes from 0 to 1.

   Thus, after the second clock pulse,

   $$Q_BQ_A = 10$$
3. When the third negative clock edge is applied to FF-A, the output Q_A toggles again and Q_A changes from 0 to 1. The 0 to 1 transition of output Q_A is treated as a positive clock edge and FF-B output Q_B does not change the state.

   Thus, after the third clock pulse,

   $$Q_BQ_A = 11$$
4. At the fourth negative clock edge, FF-A toggles and Q_A changes from 1 to 0. Hence this negative transition from 1 to 0 is treated as a negative clock edge by FF-B and its output toggles and changes from 1 to 0.

   Thus, after the fourth clock pulse,

$Q_BQ_A = 00$

Table

Clock pulse	Counter outputs Q_B	Counter outputs Q_A	State	Decimal count
Initially	0	0	-	0
1st ↓	0	1	1	1
2nd ↓	1	0	2	2
3rd ↓	1	1	3	3
4th ↓	0	0	4	0

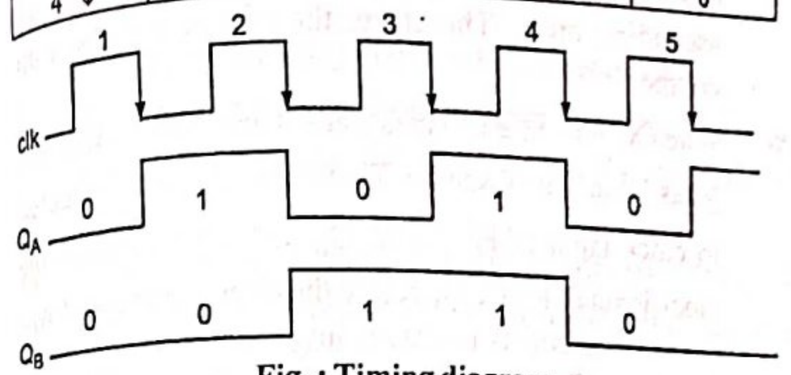


Fig. : Timing diagram

3-Bit Asynchronous Up Counter :

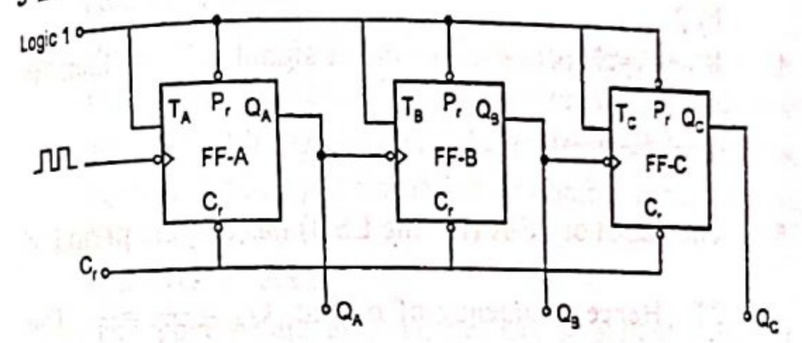


Fig. : 3-bit ripple counter

- Fig. shows the diagram of a 3-bit ripple up counter. A 3-bit counter uses 3 flip-flops. Therefore, it has $2^3 = 8$ states. The clock input is applied to FF-A and Q_A output of FF-A acts as a clock input for FF-B and Q_B output of FF-B acts as the clock input for FF-C and Q_C output of FF-C acts as the clock input for FF-D.

Counter operation :

1. Initially, all the outputs Q_A, Q_B, Q_C and Q_D are in reset condition. Therefore, $Q_DQ_CQ_BQ_A = 000$. All the flip-flops are negative edge-triggered flip-flops.
2. When the first negative clock edge is applied to FF-A, it toggles as $T_A = 1$. Hence Q_A changes from 0 to 1. The Q_A output of FF-A is applied to the clock input of FF-B. Since Q_A changes from 0 to 1, it is considered as a positive edge and FF-B does not change the state. Since Q_B does not change the state, FF-C is not triggered and Q_C also does not change the state. Hence, after the first clock pulse,

   $$Q_DQ_CQ_BQ_A = 0001$$
3. When the second negative clock edge is given to FF-A, Q_A changes from 1 to 0. Hence, a 1 to 0 transition is a negative edge for FF-B, hence its Q_B output changes from 0 to 1. A 0 to 1 transition in output Q_B is treated as positive transition, hence FF-C is not triggered, hence output Qcdoes not change the state. Hence, after the second clock pulse,

   $$Q_DQ_CQ_BQ_A = 0010$$
4. When the third negative edge is given to FF-A, Q_A changes from 0 to 1. Therefore, positive edge for FF-B, hence Q_B does not change the state and Q_C also does not change the state. Hence, after the third clock pulse,

   $$Q_DQ_CQ_BQ_A = 0011$$
5. When the fourth negative edge is applied to FF-A, Q_A changes from 1 to 0. Therefore, negative edge for FF-B, hence Q_B changes from 1 to 0. Therefore, negative edge for FF-C, hence Q_C changes from 0 to 1. Hence, after the fourth clock pulse,

   $$Q_DQ_CQ_BQ_A = 0100$$
6. When the fifth negative edge is applied to FF-A, Q_A changes from 0 to 1. Therefore, positive edge for FF-B, hence Q_B does not change the state and Q_C also does not change the state. Therefore, after the fifth clock pulse,

   $$Q_DQ_CQ_BQ_A = 0101$$
7. When the sixth negative edge is applied to FF-A, Q_A changes from 1 to 0. Therefore, negative edge for FF-B, hence Q_B changes the state from 0 to 1. Therefore, positive edge for FF-C, hence Q_C does not change the state. Therefore, after the sixth clock pulse,

   $$Q_DQ_CQ_BQ_A = 0110$$
8. When the seventh negative clock pulse is applied, Q_A changes from 0 to 1, hence a positive edge for FF-B, hence Q_B does not change the state and Q_C also does not change the state. Hence after the seventh clock pulse,

   $$Q_DQ_CQ_BQ_A = 0111$$
9. In this way, the output state of flip-flop will change after the occurrence of the negative edge of clock pulse.
10. Hence, this counter is called a binary counter. Also it is called a MOD-16 counter.

4-Bit Asynchronous Up Counter :

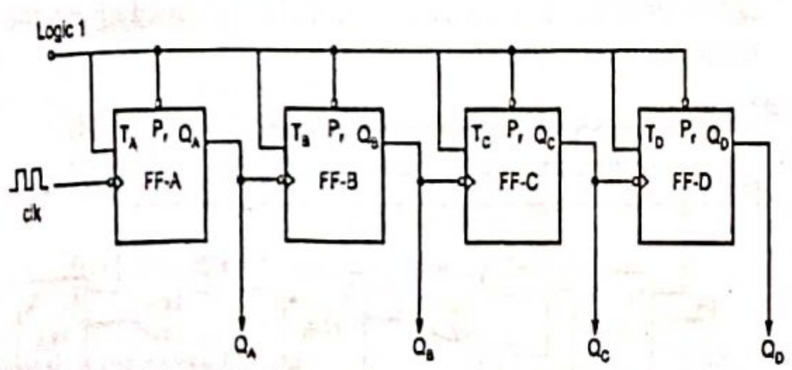


Fig. : 4-bit asynchronous up counter

Table : Truth Table

Clock	Flip-flop outputs				State	Decimal count
	Q_D	Q_C	Q_B	Q_A		
Initially	0	0	0	0	1	0
1st ↓	0	0	0	1	2	1
2nd ↓	0	0	1	0	3	2
3rd ↓	0	0	1	1	4	3
4th ↓	0	1	0	0	5	4
5th ↓	0	1	0	1	6	5
6th ↓	0	1	1	0	7	6
7th ↓	0	1	1	1	8	7
8th ↓	1	0	0	0	9	8
9th ↓	1	0	0	1	10	9
10th ↓	1	0	1	0	11	10
11th ↓	1	0	1	1	12	11
12th ↓	1	1	0	0	13	12
13th ↓	1	1	0	1	14	13
14th ↓	1	1	1	0	15	14
15th ↓	1	1	1	1	16	15

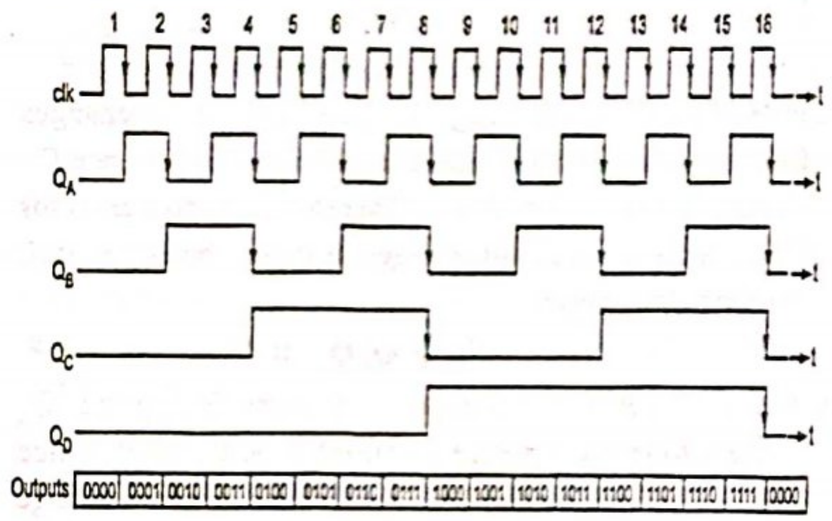


Fig. : Waveforms for 4-bit asynchronous up counter

Q.9. Define Up/Down counter .Describe the working of Up / Down counter ?

Or Define Up/Down counter and explain modulus of a Counter with state diagram of a counter ?

Ans. We can design a counter which can count in ascending or descending order, using the same circuit. It will count in either upward or downward direction depending on the status of mode selection input.

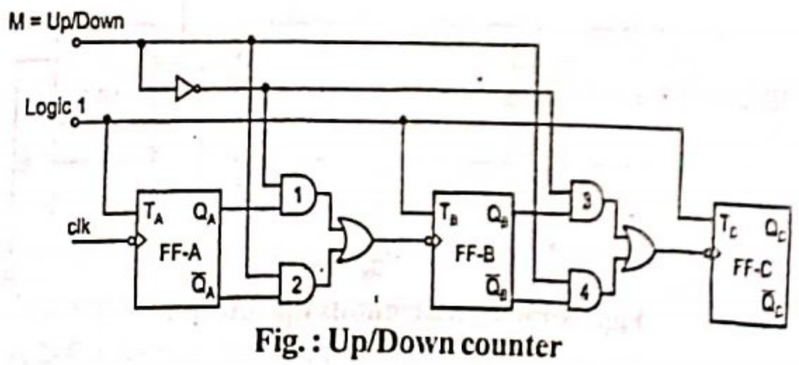


Fig. : Up/Down counter

Working :

A mode control input is used, so that when

M = 0, the counter counts in ascending order i.e. UP counting.

M = 1, the counter counts in descending order i.e. DOWN counting

- When M = 0, $\bar{M}$ = 1, gates 1 and 3 are enabled whereas gates 2 and 4 are disabled. Therefore, Q_A gets connected to the clock input of FF-B and Q_B gets connected to the clock input of FF-C. Hence, the counter counts in ascending order. Therefore, the circuit acts as a up counter.
- When M = 1, $\bar{M}$ = 0, gates 2 and 4 are enabled whereas gates 1 and 3 are disabled. Therefore, Q_A gets connected to clock input of FF-B and $\bar{Q}_B$ gets connected to the clock input of FF-C. Therefore the circuit acts as a down counter and counts in descending order.

Frequency Division in Asynchronous Counters :

- The toggle mode of flip-flop divides the clock frequency by 2.
- If one cycle period of the clock signal is T sec then the clock frequency $f = \left(\frac{1}{T}\right)$ Hz.
- The output of FF-A (i.e. the LSB) has a cycle period of 2T. Hence, frequency of output $Q_A = \frac{1}{2T} = \frac{f}{2}$. The period of Q_B is 4T. Therefore, the frequency of $Q_B = \frac{1}{4T} = \frac{f}{4}$. Similarly, the period of Q_C is 8T.

Therefore, frequency $f = \frac{1}{8T}$.

Modulus of a Counter :

- The total number of discrete states through which a counter can pass is called modulus of a counter.
- A counter consisting of a flip-flop can count $N = 2^n$ discrete states.
- In this case, modulus of n-bit counter will be N and this counter is called modulus N or Mod-N counter.
- The maximum count that can be recorded by a binary number counter consisting of n flip-flops will be $(2^n - 1)$. The states of mod 4 counter are 0000, 0001, 0010, 0011.

State diagram of a counter :

- The state diagram of a counter represents the states of a counter graphically.
- For example for a 3-bit up counter the state diagram is shown in Fig.

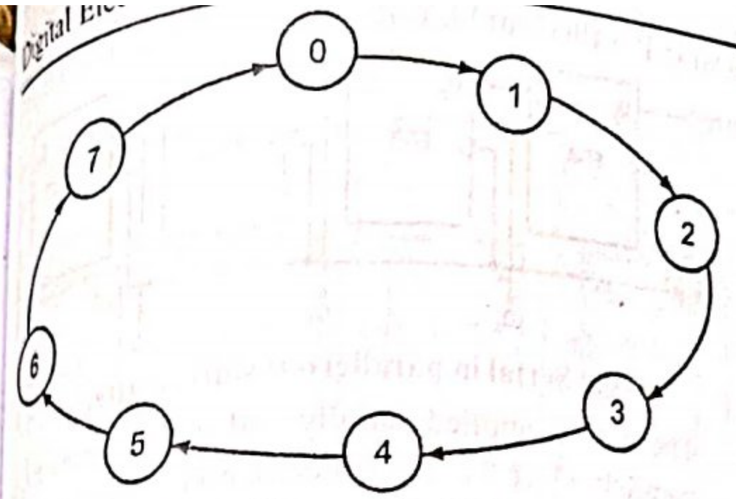


Fig. : State diagram of a 3-bit up counter

- The number written inside a circle represents the state numbers. Whereas, the arrow shows the direction of counter.

Q.3. Define register .and explain shift register with its type ?

Or Describe working of SISO shift register with proper circuit diagram ?

Or Describe working of PISO shift register with proper circuit diagram ?

Ans. Registers : A group of flip-flops used to store a word, is known as a register. A flip-flop can store 1-bit information.

- Thus an n-bit register has a group of n flip-flops and can store any binary information or number containing n-bits.

Shift Registers :

- The binary data in a register can be shifted from one stage to another stage within the register or into or out of the register after application of the clock pulses.
- This type of shifting is essential for certain arithmetic and logic operations used in microprocessors.
- The group of registers called 'shift registers' are very important in applications involving storage and transfer of data in a digital system.

Types of Shift Registers. They are as follows :

Serial In Serial Out Shift Register (SISO) :

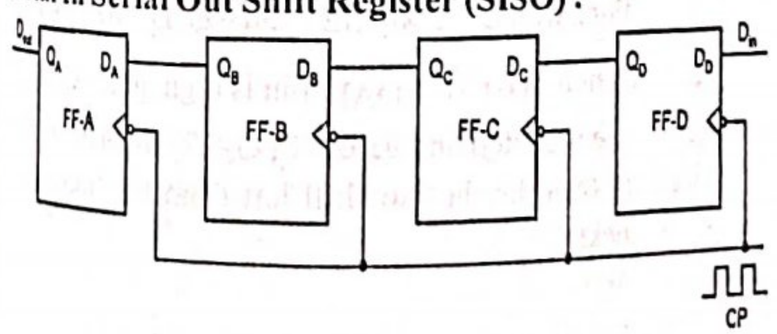
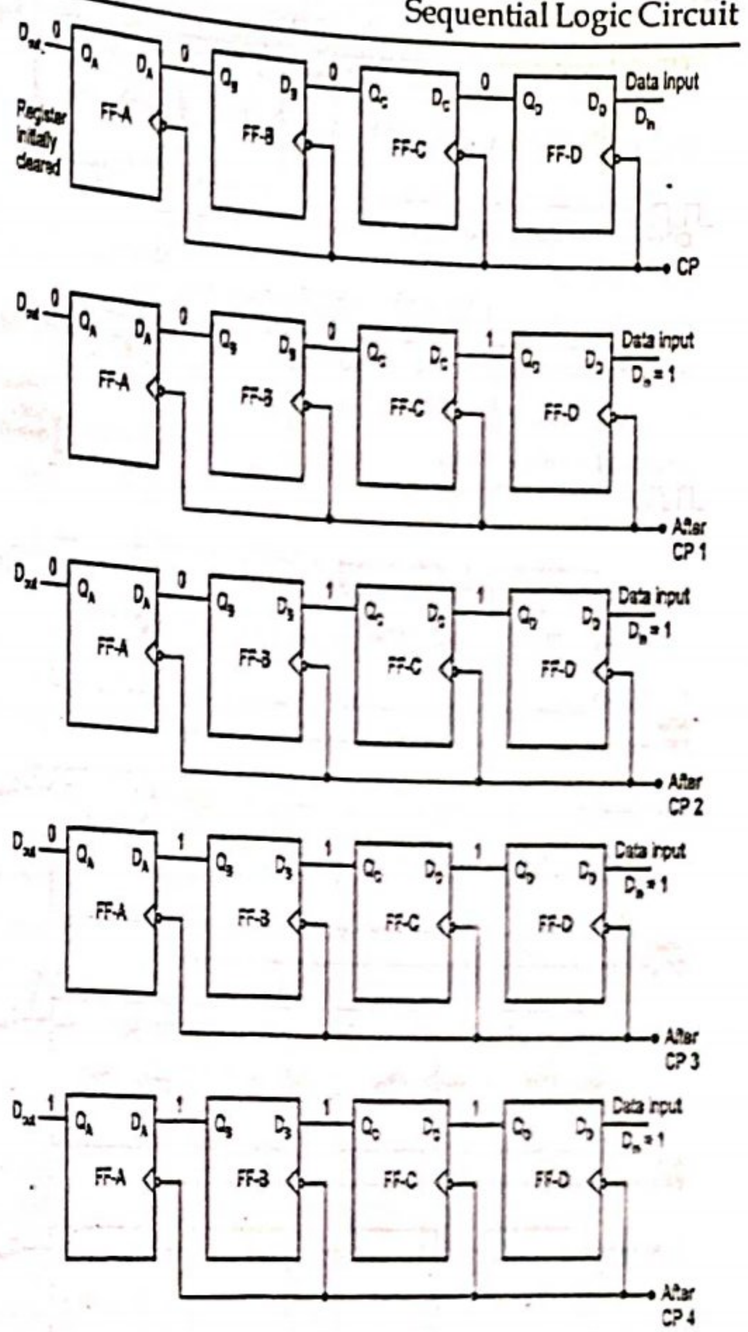


Fig. : Shift-left register



- Consider the entry of the 4-bit binary number 1111 into the register, starting with the leftmost bit. Initially, the register is cleared.

$\therefore \quad Q_A Q_B Q_C Q_D = 0000$

(i) When data 1111 is applied serially, i.e.

$D_n = 1, Q_A Q_B Q_C Q_D = 0000$

The application of first negative clock edge, sets the rightmost flip-flop, then the data stored is

$Q_A Q_B Q_C Q_D = 0001$

(ii) When the next falling edge is applied, flip-flop Q_c sets. Therefore, the data in the register is

$Q_A Q_B Q_C Q_D = 0011$

(iii) When the third falling edge is applied, flip-flop Q_B sets. Therefore, the register contents become

$Q_A Q_B Q_C Q_D = 0111$

(iv) The fourth negative clock edge gives

$Q_A Q_B Q_C Q_D = 1111$

Shift-right register :

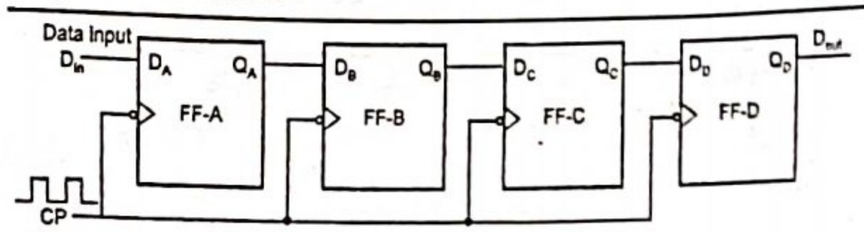


Fig. : Shift-right register

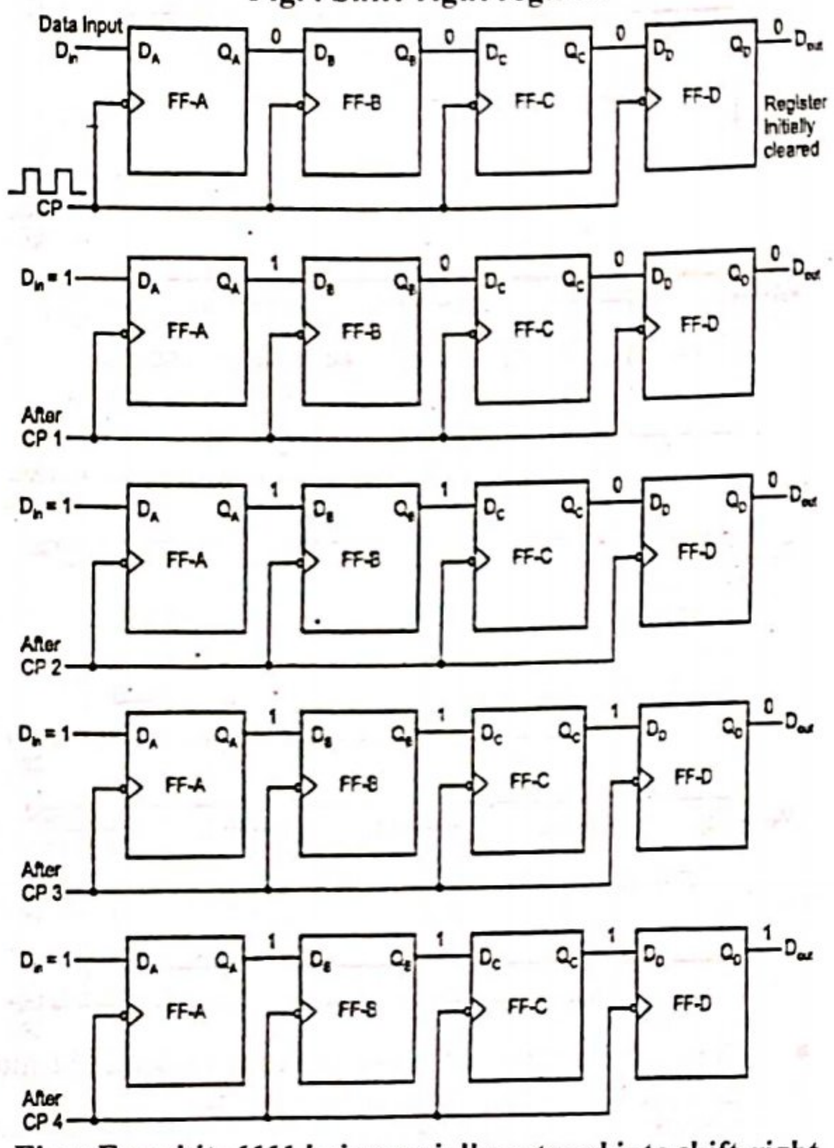


Fig. : Four bits 1111 being serially entered into shift-right register

- Initially, the register is cleared, so $Q_A\ Q_B\ Q_C\ Q_D = 0000$. When data 1111 is applied serially, leftmost bit is applied as D_{in}.

$$Din = 1$$

(i) On arrival of first negative clock edge, FF-A sets and the stored word becomes $Q_A\ Q_B\ Q_c$

$$Q_D = 1000.$$

(ii) When the next negative clock edge is applied, FF-B sets.

$$\therefore \quad Q_A Q_B Q_C Q_D = 1100$$

(iii) When the third falling clock edge is applied, FF-C sets.

$$\therefore \quad Q_A Q_B Q_C Q_D = 1110$$

(iv) On the fourth falling clock edge,

$$Q_A Q_B Q_C Q_D = 1111$$

Serial In Parallel Out Shift Register (SIPO) :

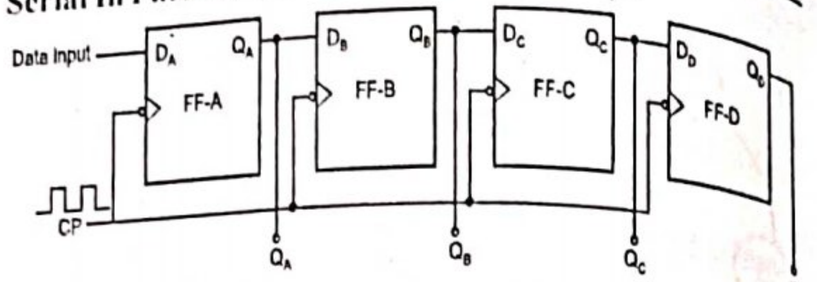


Fig. : Serial in parallel out shift register

- The data is applied serially and the output is taken parallely. Once the data is stored, each bit appears at the respective outputs and all the bits are available simultaneously.

Parallel In Serial Out Shift Register (PISO) :

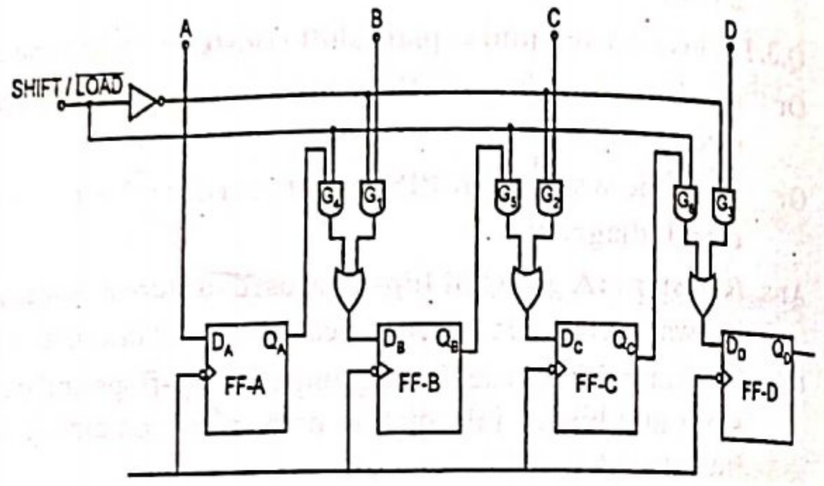


Fig. : Parallel in serial out shift register

- In this type of shift register, the bits are loaded simultaneously into their respective stages using parallel lines. A, B, C, D are the four input lines for loading the data. The SHIFT/ $\overline{LOAD}$ is the control input which permits shifting or loading operation of the register.

1. When SHIFT/ $\overline{LOAD}$ pin is low, gates G_1, G_2, G_3 are enabled and the data is loaded in the respective flip-flops. When the clock pulse is applied, the flip-flops with D = 1 will SET and with D = 0 will RESET.
2. When SHIFT/ $\overline{LOAD}$ pin is high, gates G_1, G_2, G_3 are disabled and gates G_4, G_5, G_6 are enabled. This allows the data to shift left from one stage to the next.

Parallel In Parallel Out Register (PIPO) :

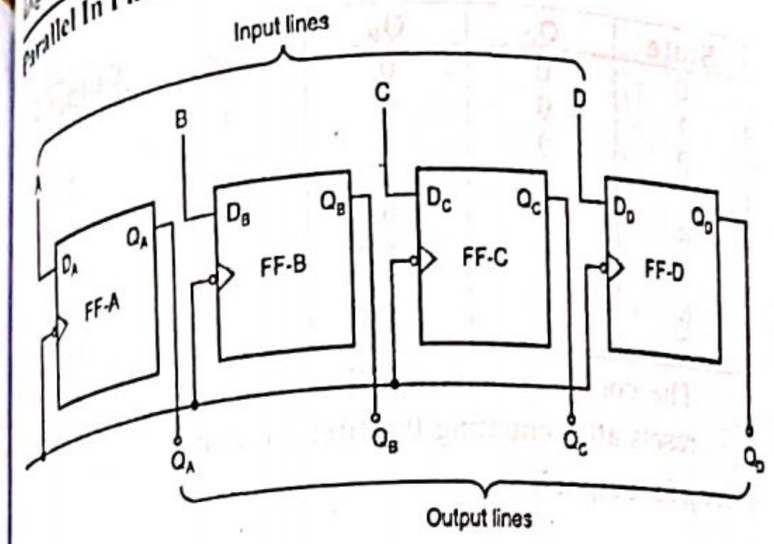


Fig. : Parallel in parallel out register output

- In parallel in parallel out register, the data bits are loaded simultaneously on 4-parallel input lines and the outputs are taken simultaneously on 4 parallel output lines.

Q.6. Define **Universal Shift Register and describe its operation ?**

Or **Describe the operation of Universal Shift Register ?**

Or **Draw with neat sketch explain Universal Shift Register?**

Ans. **Universal Shift Register :** It is a shift register which can shift data in both directions as well as load it parallely. Thus this register is capable of performing three operations :

(i) Parallel loading

(ii) Shifting the data serially to the left

(iii) Shifting the data serially to the right.

- The type of operation performed will be decided by the status of the mode control input.
- When mode control is at logic i, data is loaded parallely and when it is at logic 0, it will shift the data serially.
- When mode control is connected to ground, it will act as a bidirectional register.
- It has separate clock inputs one for shift operation and other for load operation.

Operation:

1. When mode, control (M) is connected to logic 1, AND gates 2, 4, 6, 8 will be enabled and AND gates 1, 3, 5, 7 will be disabled. The 4-bit binary data will be loaded parallely. The clock-2 input will be applied to the flip-flops, since M = 1, AND gate-10 is enabled and gate-9 is disabled.
2. When mode control (M) is connected to logic 0, AND gates 1, 3, 5, 7 will be enabled and gates 2, 4, 6, 8 will be disabled. The data will be shifted serially. The clock-1 input will be applied to the flip-flops, since M = 0, AND gate-9 is enabled and gate-10 is disabled. The data is shifted serially to right from Q_A to Q_D.
3. When mode control (M) is connected to logic 1, AND gates 2, 4, 6, 8 will be enabled. This mode permits parallel loading of the register and shift-left operation. The shift-left operation can be accomplished by connecting the output of each flip-flop to the parallel input of the previous flip-flop and serial input is applied at the input D.

Q.9. Define **Universal Shift Register 1C 7495 with its feature and pin configuration ?**

Ans. Universal Shift Register 1C 7495 :

General description:

- IC 7495 is a TTL MSI shift register.
- It is a 4-bit shift register with serial and parallel synchronous operating modes.
- Because of its capability to operate in all the possible modes, it is called as a universal shift register.

Features :

The important features of this chip are as follows :

- Synchronous shift-left capacity.
- Synchronous parallel loading is possible.
- It has separate clock inputs one for shift operation and the other for load operation.
- Expansion with shift right is possible. That means cascading of two or more 7495 ICs for more than 4-bits is possible.

Pin configuration :

- The pin configuration of IC 7495 is as shown in Fig. 1 and Fig. 2 shows the internal logic diagram.

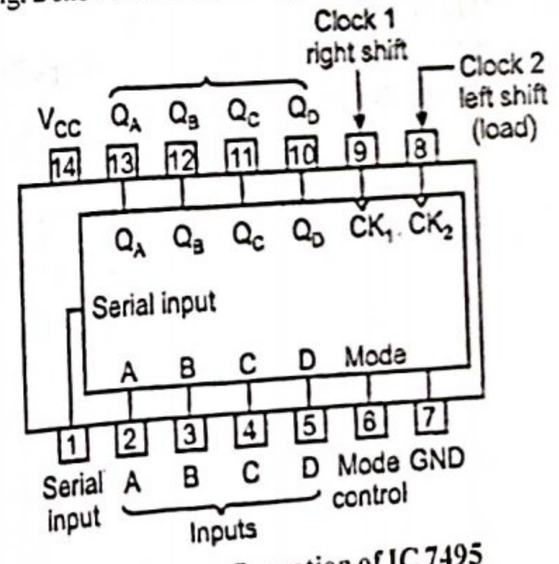


Fig. Pin configuration of IC 7495

Table

Pin name	Function
(1) Serial in	Serial input
(2, 3, 4, 5) A, B, C, D	Parallel inputs
Mode (6)	Mode control
Clock - 2 (8)	Clock for left shift and load
Clock - 1 (9)	Clock for right shift
Q_A to Q_D (10 to 13)	Outputs

Fig. Internal logic diagram of IC 7495

- A, B, C, D are the inputs to the four internal flip-flops with A acting as LSB and D as MSB. Q_A through Q_D are the corresponding outputs.
- The internal logic diagram of IC 7495 is as shown in Fig.

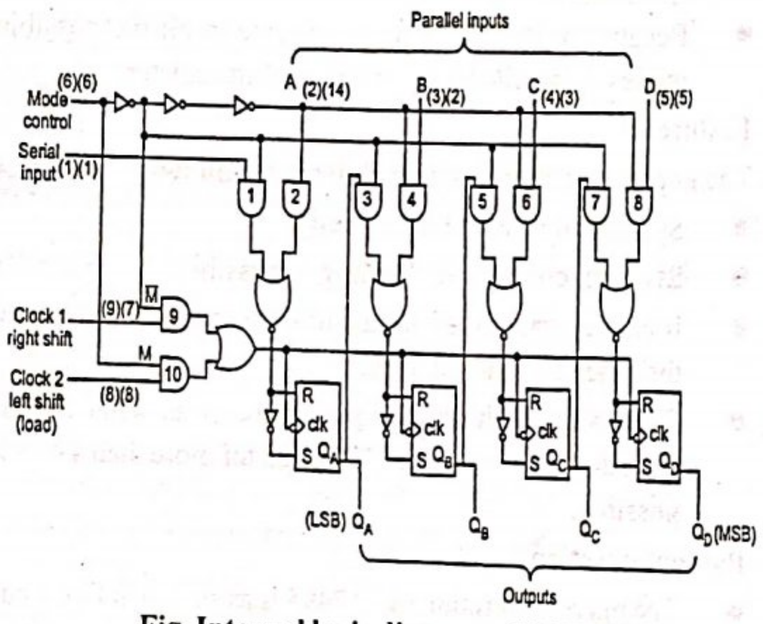


Fig. Internal logic diagram of IC 7495

Example 1: Design a Mod-5 ripple counter.

Solution : A Mod-5 counter will require 3 flip-flops. Therefore, $2^3 = 8$ states. Out of 8, we will count 5 states.

Table : Truth Table

State	Q_C	Q_B	Q_A	Output Y
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	0

The counter counts the first 5 states from 0 to 4 and resets after counting the first 4 counts.

K-map for output Y :

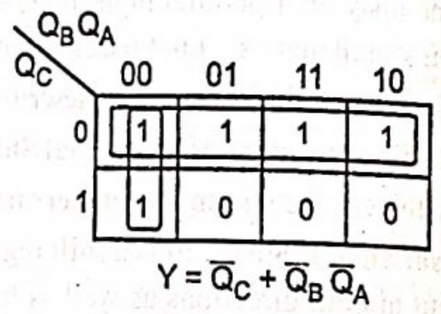


Fig. : K-map for output Y

Logic Diagram :

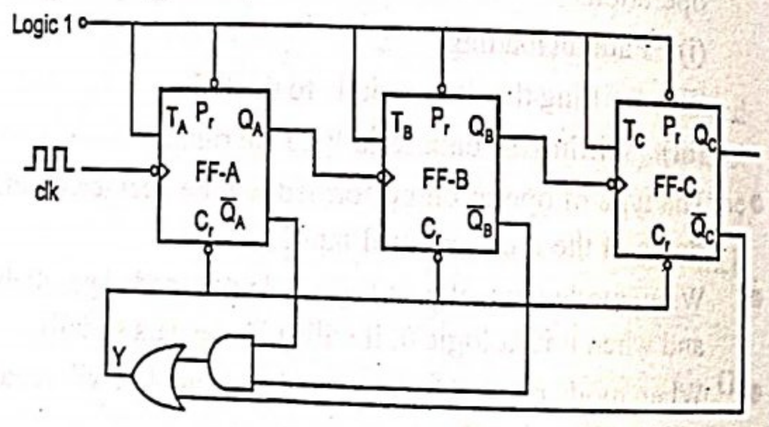


Fig. : Logic diagram

□□□

OBJECTIVE TYPE QUESTIONS

1. **Latches constructed with NOR and NAND gates tend to remain in the latched condition due to which configuration feature?**
   (a) Low input voltages (b) Synchronous operation
   (c) Gate impedance (d) Cross coupling
   Ans. (d)
2. **One example of the use of an S-R flip-flop is as .....**
   (a) Transition pulse generator
   (b) Racer
   (c) Switch debouncer
   (d) Astable oscillator
   Ans. (c)
3. **The truth table for an S-R flip-flop has how many VALID entries?**
   (a) 1 (b) 2
   (c) 3 (d) 4
   Ans. (c)
4. **When both inputs of a J-K flip-flop cycle, the output will .....**
   (a) Be invalid (b) Change
   (c) Not change (d) Toggle
   Ans. (c)
5. **Which of the following is correct for a gated D-type flip-flop?**
   (a) The Q output is either SET or RESET as soon as the D input goes HIGH or LOW
   (b) The output complement follows the input when enabled
   (c) Only one of the inputs can be HIGH at a time
   (d) The output toggles if one of the inputs is held HIGH
   Ans. (a)
6. **A basic S-R flip-flop can be constructed by cross-coupling of which basic logic gates?**
   (a) AND or OR gates (b) XOR or XNOR gates
   (c) NOR or NAND gates
   (d) AND or NOR gates
   Ans. (c)
7. **The logic circuits whose outputs at any instant of time depends only on the present input but also on the past outputs are called .....**
   (a) Combinational circuits (b) Sequential circuits
   (c) Latches (d) Flip-flops
   Ans. (b)
8. **Whose operations are more faster among the following?**
   (a) Combinational circuits (b) Sequential circuits
   (c) Latches (d) Flip-flops
   Ans. (a)
9. **How many types of sequential circuits are?**
   (a) 2 (b) 3
   (c) 4 (d) 5
   Ans. (a)
10. **The sequential circuit is also called .....**
    (a) Flip-flop (b) Latch
    (c) Strobe (d) Adder
    Ans. (b)
11. **The basic latch consists of .....**
    (a) Two inverters (b) Two comparators
    (c) Two amplifiers (d) Two adders
    Ans. (a)
12. **In S-R flip-flop, if Q = 0 the output is said to be .....**
    (a) Set (b) Reset
    (c) Previous state (d) Current state
    Ans. (b)
13. **The output of latches will remain in set/reset untill .....**
    (a) The trigger pulse is given to change the state
    (b) Any pulse given to go into previous state
    (c) They don't get any pulse more
    (d) The pulse is edge-triggered
    Ans. (a)
14. **What is a trigger pulse ?**
    (a) A pulse that starts a cycle of operation
    (b) A pulse that reverses the cycle of operation
    (c) A pulse that prevents a cycle of operation
    (d) A pulse that enhances a cycle of operation
    Ans. (a)
15. **The circuits of NOR based S-R latch classified as asynchronous sequential circuits, why ?**
    (a) Because of inverted outputs
    (b) Because of triggering functionality
    (c) Because of cross-coupled connection
    (d) Because of inverted outputs & triggering functionality
    Ans. (c)

□□□

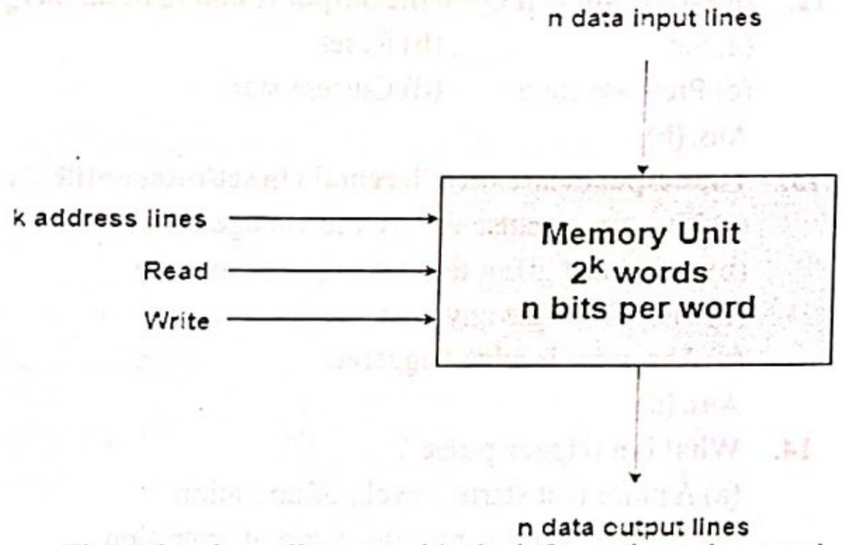
Chapter 5

Memory Devices

Q. 1. Draw the RAM organization explain it briefly

Ans: In random-access memory(RAM) the memory cells can be accessed for information transfer from any desired random location. Communication between a memory and its environment is achieved through data input and output lines, address selection lines, and control lines that specify the direction of transfer.

A block diagram of a RAM unit is shown below:



The n data input lines provide the information to be stored in memory, and the n data output lines supply the information coming out of particular word chosen among the 2^k available inside the memory. The two control inputs specify the direction of transfer desired.

The two operations that a random access memory can perform are the write and read operations. The write signal specifies a transfer-in operation and the read signal specifies a transfer-out operation. On accepting one of these control signals. The internal circuits inside the memory provide the desired function. 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 into the address lines.
2. Apply the data bits that must be stored in memory into the data input lines.
3. Activate the write input.

The memory unit will then take the bits presently available in the input data lines and store them in the specified by the address lines. The steps that must be taken for the purpose of transferring a stored word out of memory are as follows:

1. Apply the binary address of the desired word into the address lines.
2. Activate the read input.

The memory unit will then take the bits from the word that has been selected by the address and apply them into the output data lines.

Q 2. If the no of adress lines in a memory is 10, and data lines are 2. What will be the memory size.

Solution: Total memory size= $2^{\text{address lines}}$ * data lines

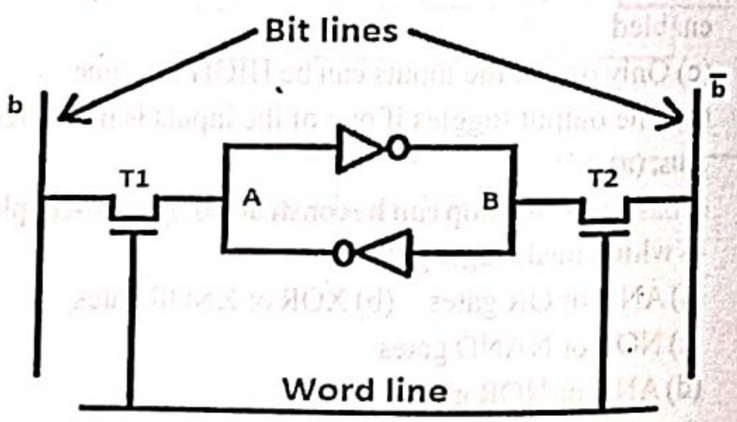
$= 2^{10}* 2= 1024* 2$ bits

Q 3. Explain Static RAM. What are characteristics of Static RAM?

Ans: Static Random Access Memory is a type of RAM that retains the information in its memory as long as power is being supplied. Static RAM provides faster access to the data and is more expensive compared with DRAM. SRAM does not need to be refreshed periodically. This type of memory requires constant power. Thus this type of memory is called volatile memory.

SRAM memories are used to build Cache Memory.

The below figure shows a cell diagram of SRAM. A latch is formed by two inverters connected as shown in the figure. Two transistors T1 and T2 are used for connecting the latch with two-bit lines. The purpose of these transistors is to act as switches that can be opened or closed under the control of the word line.



For **Read operation**, the word line is activated by the address input to the address decoder. The activated word line closes both the transistors (switches) T1 and T2. Then the bit values at points A and B can transmit to their respective bit lines. The sense/write circuit at the end of the bit lines sends the output to the processor.

For **Write operation**, the address provided to the decoder activates the word line to close both the switches. Then the bit value that is to be written into the cell is provided through the sense/write circuit and the signals in bit lines are then stored in the cell.

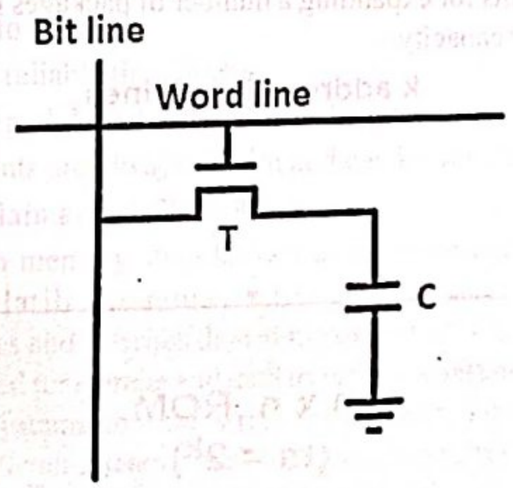
Characteristic of Static RAM

- Long life
- No need to refresh
- Faster
- Used as cache memory
- Large size
- Expensive
- High power consumption

Q 4. Explain Dynamic RAM. What are characteristics of Dynamic RAM(DRAM)?

Ans: DRAM is a type of RAM module that stores each bit of data within a separate capacitor. This is a proficient way to store the data in memory because it requires less physical space to store data.

DRAM can hold more amounts of data by a particular chip size. Capacitors in DRAM are need to be constantly recharged to keep their charge, thus, DRAM requires more power. DRAM, unlike SRAM, must be continually refreshed in order to maintain the data. This is done by placing the memory on a refresh circuit that rewrites the data several hundred times per second. The main memory is generally made up of DRAM chips.



Above figure shows the DRAM cell. Information is stored in a DRAM cell in the form of a charge on a capacitor and this charge needs to be periodically recharged.

For storing information in this cell, transistor T is turned on and an appropriate voltage is applied to the bit line. This causes a known amount of charge to be stored in the capacitor. After the transistor is turned off, due to the property of the capacitor, it starts to discharge. Hence, the information stored in the cell can be read correctly only if it is read before the charge on the capacitors drops below some threshold value.

Characteristics of Dynamic RAM

- Short data lifetime
- Needs to be refreshed continuously
- Slower as compared to SRAM
- Used as RAM
- Smaller in size
- Less expensive
- Less power consumption

Q 5. What are differences between static RAM and Dynamic RAM.

Ans: Difference between SRAM and DRAM :

Below table lists some of the differences between SRAM and DRAM:

SRAM	DRAM
1. SRAM has lower access time, so it is faster compared to DRAM.	1. DRAM has higher access time, so it is slower than SRAM.
2. SRAM is costlier than DRAM.	2. DRAM costs less compared to SRAM.
3. SRAM requires constant power supply, which means this type of memory consumes more power.	3. DRAM offers reduced power consumption, due to the fact that the information is stored in the capacitor.
4. Due to complex internal circuitry, less storage capacity is available compared to the same physical size of DRAM memory chip.	4. Due to the small internal circuitry in the one-bit memory cell of DRAM, the large storage capacity is available.
5. SRAM has low packaging density.	5. DRAM has high packaging density.

Q.6. Explain bipolar RAM with suitable diagram.

Ans: A bipolar memory cell (bipolar RAM) is a high-speed random- access memory consists of a cross-coupled pair of transistors.

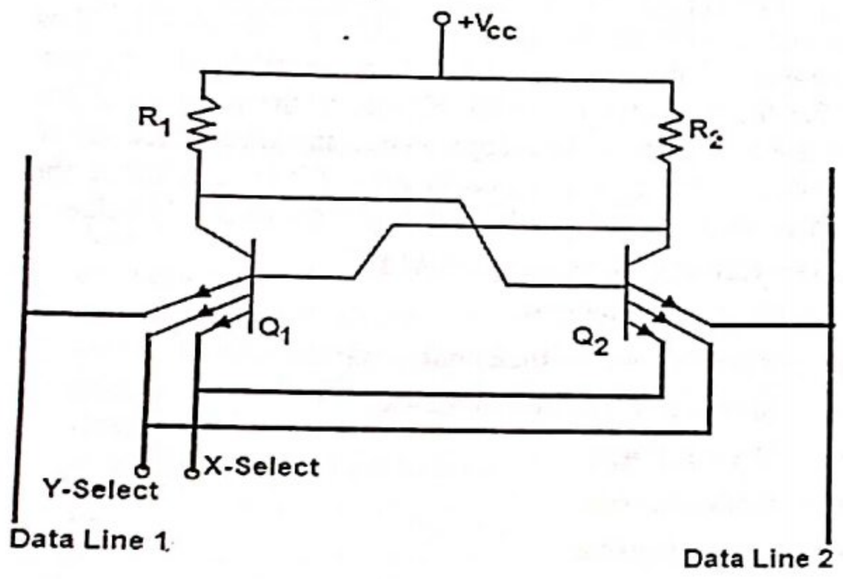


Figure shows bipolar RAM cell. The two multiple emitter transistor are cross coupled to form a simple bistable latch. the connections to the emitter allow the cell to be selected and data to be entered or read out. When X, Y select lines are at low voltage, current flows through transistor Q_1 or Q_2 to the line which is at low voltage. When particular cell is selected corresponding selct lines are raised at high voltage.

A typical static bipolar RAM chip has a capacity of 256 bits and an access time of 50ns

Q.7. Briefly explain DDR RAM. What are advantages and disadvantages?

Ans: DDR-RAM stands for Double Data Rate Synchronous Dynamic Random-Access Memory. These are the computer memory that transfers the data twice as fast as regular chips because DDR memory can send and receive signals twice per clock cycle as a comparison.

They are widely used in applications that are demanding high speed, memory, for example, graphic cards that need to access a large amount of information in a very short time to achieve the best graphics processing efficiency to improve the gaming.

DDR1, DDR2, and DDR3 are the types of DDR RAM memories that use the 2.5, 1.8, and 1.5V supply voltages respectively, thus it produces less heat and provides more efficiency in power management

Advantages of DDR RAM

- It offers much faster speeds than SDRAM.
- Each generation of DDR are updated and following with its successor DDR2, DDR3, DDR4.

Disadvantages of DDR RAM

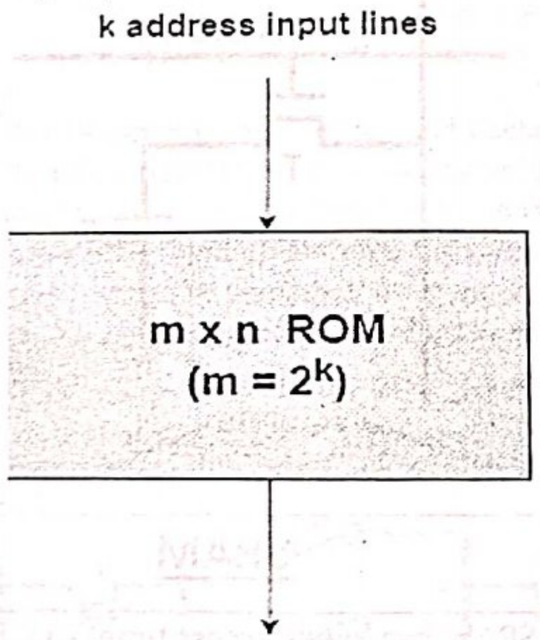
- It can't be used with old motherboards.
- Some machines support only slower RAM.
- They are not physically fit in memory slots due to its notches
- There pinouts are totally different from others.

Q 8. Explain ROM organization briefly.

Ans: Read-only memory(ROM) is a memory unit that performs the read operation only; it does not have a write capability. This implies that the binary information stored in a ROM is made permanent during the hardware production of the unit and cannot be altered by writing different words into it.

a ROM is restricted to reading words that are permanently stored within the unit. The binary information to be stored, specified by the designer, is then embedded in the unit to form the required interconnection pattern. ROMs come with special internal electronic fuses that can be programmed for a specific configuration. Once the pattern is established, it stays within the unit even when power is turned off and on again.

An m x n ROM is an array of binary cells organized into m words of n bits each. As shown in the block diagram below, a ROM has k address input lines to select one of 2^k = m words of memory, and n input lines, one for each bit of the word. An integrated circuit ROM may also have one or more enable inputs for expanding a number of packages into a ROM with larger capacity.



ROMs find a wide range of applications in the design of digital systems. As such, it can implement any combinational circuit with k inputs and n outputs. When employed in a computer system as a memory unit, the ROM is used for storing fixed programs that are not to be altered and for tables of constants that are not subject to change. ROM is also employed in the design of control units for digital computers. As such, they are used to store coded information that represents the sequence of internal control variables needed for enabling the various operations in the computer

Q.9. Explain the types of ROM briefly. What are advantages of ROM?

Ans: Following are the types of Rom:

Programmable read-only memory (PROM) : PROM is read-only memory that can be modified only once by a user. PROMs are used in digital electronic devices to store permanent data, The key difference from a standard ROM is that the data

is written into a ROM during manufacture, while with a PROM the data is programmed into them after manufacture.

EPROM (Erasable and Programmable Read Only Memory): It is a type of PROM. It is a non volatile memory, because it is not erased due to power on or off. EPROM can be erased by exposing it to ultra-violet light for a duration of up to 40 minutes.

EEPROM (Electrically Erasable and Programmable Read Only Memory): EEPROM is programmed and erased electrically. It can be erased and reprogrammed about ten thousand times. Both erasing and programming take about 4 to 10 ms (millisecond). In EEPROM, any location can be selectively erased and programmed. EEPROMs can be erased one byte at a time, rather than erasing the entire chip. Hence, the process of reprogramming is flexible but slow.

The advantages of ROM are as follows :

- Non-volatile in nature
- Cannot be accidentally changed
- Cheaper than RAMs
- Easy to test
- More reliable than RAMs
- Static and do not require refreshing
- Contents are always known and can be verified

Q. 10. Explain about Flash Memory.

Ans: Flash memory, also known as flash storage, is a type of non volatile memory, that erases data in units called blocks and rewrites data at the byte level. Flash memory is widely used for storage and data transfer in consumer devices, enterprise systems and industrial applications. Flash memory retains data for an extended period of time, regardless of whether a flash-equipped device is powered on or off.

Flash memory is used in enterprise data center server, storage and networking technology, as well as in a wide range of consumer devices, including USB flash drives — also known as memory sticks — SD cards, mobile phones, digital cameras, tablet computers and PC cards in notebook computers.

The two main types of flash memory, NOR flash and NAND flash. The NAND type is found mainly in memory cards, USB flash drives, feature phones, smart phones.

Q.11. Define data converter and write its classification

Or Define data converter and explain digital to analog converter with block diagram ?

Or With block diagram explain digital to analog converter ?

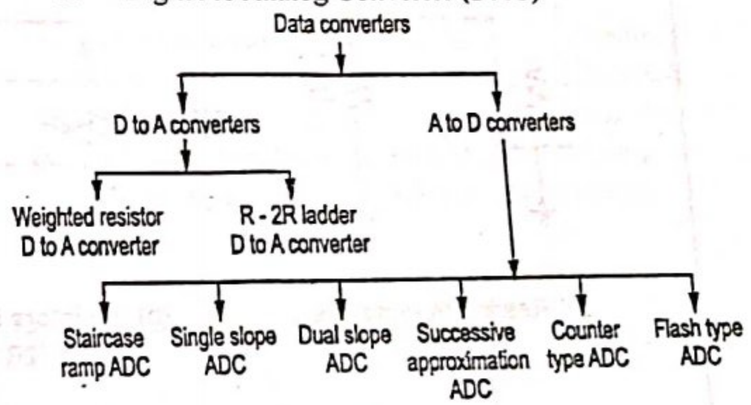
Or Define digital to analog converter with type and explain one of them briefly ?

Ans. It is often necessary that before processing the analog data, by a digital system, it should be changed to an equivalent digital form. Similarly, after processing the data, it may be desirable that the final result obtained in the digital form be converted back to the analog form. Therefore, data converters are necessary in digital systems.

- A combinational digital circuit which converts the one form of data into the other or vice-versa is called as a data converter.

Classification of Data Converters :

- Data converters are classified as follows:
  1. Analog to Digital Converter (ADC)
  2. Digital to Analog Converter (DAC)



Digital to Analog Converters :

- The combinational logic system which converts the analog signal to digital signal is called the digital to analog converter (D/A). In short, it is called as DAC.

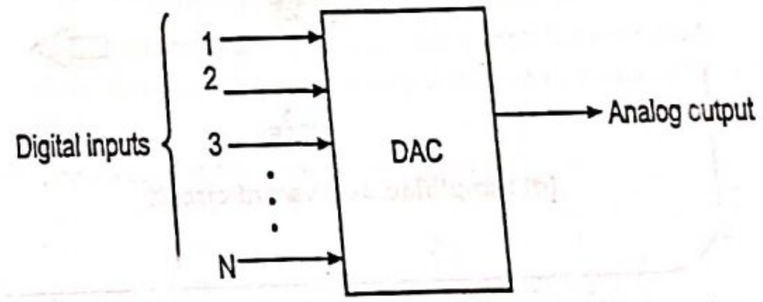


Fig. : Block diagram of Digital to Analog CD/A) converter

- The input to a DAC is an N-bit binary signal in parallel form. The analog output voltage V_o is expressed by the equation

$$V_o = K(2^{N-1} b_{N-1} + 2^{N-2} b_{N-2} + \ldots + 2^2 b_2 + 2^1 b_1 + 2^0 b_0)$$

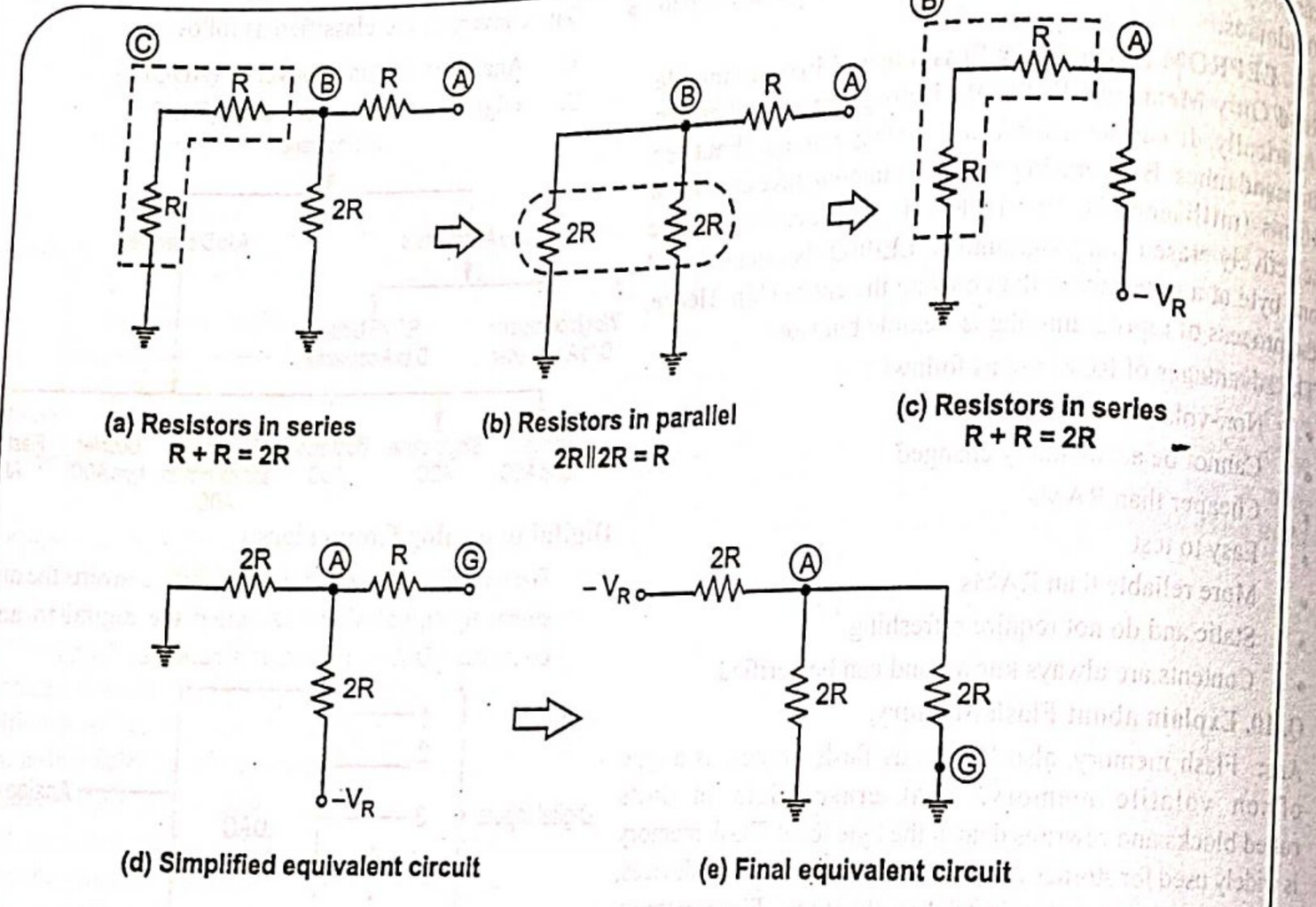
where K is the proportionality constant.

- **The two types of commonly used DACs are :**
  1. Weighted-Resistor D/A converter and
  2. R-2R Ladder D/A converter.

(1) Weighted-Resistor D/A Converter :

- Fig. shows the logic circuit of a binary weighted resistor type D/A converter (DAC).
- It uses a network of binary weighted resistors and op-amp summing amplifier.
- The resistors $2^1 R = 2R, 2^2 R = 4R, \ldots, 2^n R$ are from the network of binary weighted resistors.
- There are n number of digitally controlled electronic switches used one per digit bit.
- They are SPOT type switches.

- Thus the binary weighted resistors can be either connected to the ground or negative reference voltage ($-V_R$) through digitally controlled switches $S_1, S_2, ..., S_n$.
- Depending upon the positions of various switches, the currents $I_1, I_2, ..., I_n$ will start flowing through the weighted resistors 2R, 4R, ..., 2^nR respectively as shown in Fig.
- The n-bit digital word has di as MSB and d_n as LSB.
- R_F is the feedback resistor and I_0 is the output current.



(a) Resistors in series R + R = 2R (b) Resistors in parallel 2R||2R = R (c) Resistors in series R + R = 2R

(d) Simplified equivalent circuit (e) Final equivalent circuit

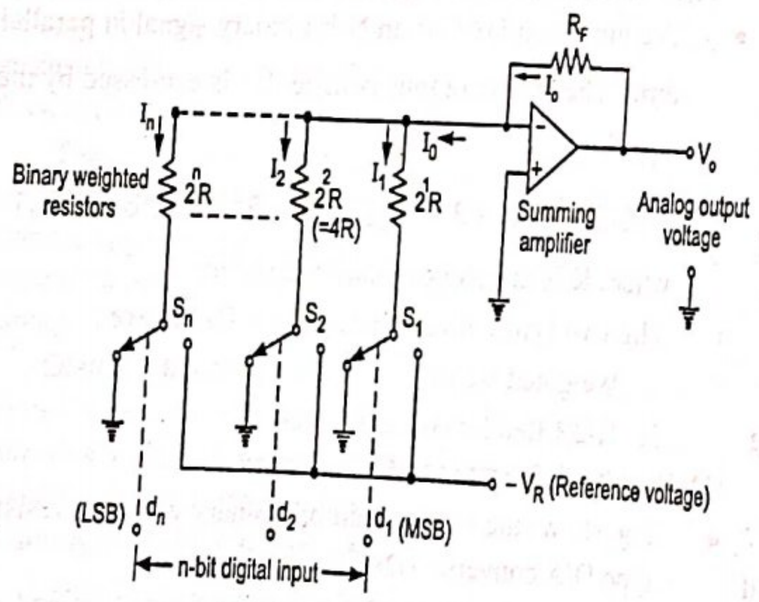


Fig. : Binary weighted resistor type DAC

- Assuming the op-amp to be an ideal, the output current I_0 can be expressed as the sum of the individual currents flowing through the weighted resistors.

$$I_0 = I_1 + I_2 + ... + I_n \quad .....(1)$$

$$I_0 = \frac{V_R}{2R}d_1 + \frac{V_R}{4R}d_2 + ... + \frac{V_R}{2^nR}d_n$$

- Note that $d_1, d_2, ..., d_n$ are binary digits and can have a value either '0' or '1'.

$$\therefore \quad I_0 = \frac{V_R}{R}\left[d_1 2^{-1} + d_2 2^{-2} + ... + d_n 2^{-n}\right]$$

The output voltage V_0 is given by

$$V_0 = I_0 R_F$$

$$= V_R \cdot \frac{R_F}{R}\left[d_1 2^{-1} + d_2 2^{-2} + ... + d_n 2^{-n}\right]$$

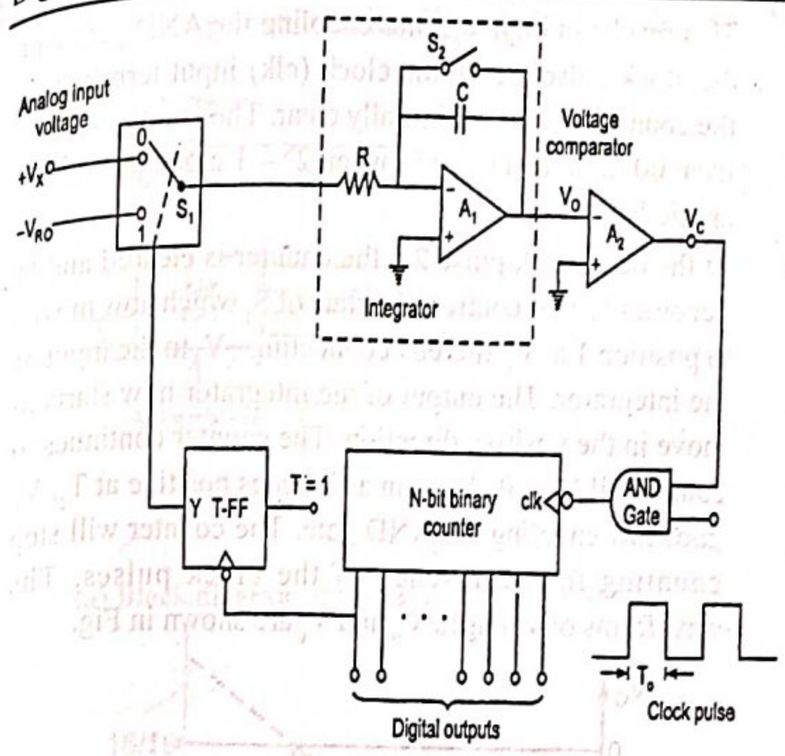
This is the required expression for output voltage.

If $$R_F = R$$

$$\therefore \quad V_0 = V_R\left[d_1 2^{-1} + d_2 2^{-2} + ... + d_n 2^{-n}\right]$$

By substituting the values of $d_1, d_2, ... d_n$ in the above equation, we can obtain the value of the corresponding analog output voltage V_0.

The use of a negative reference voltage ($-V_R$) gives a



Advantages :

1. Simple implementation.
2. Easy computations.

Disadvantages :

1. The accuracy and stability depend upon the accuracy of the resistors used.
2. It requires a wide range of resistor values.
3. The use of wide range of resistor values restricts its use. It has a poor resolution for limited range of resistors.
4. A very high wattage resistor is required for MSB position.
5. A precision resistor is required for each bit position of digital input.

(2) R-2R Ladder Type D/A Converter :

- The binary ladder network largely overcomes the problem of the weighted resistor network.
- This type of circuit also has a resistive network to produce binary weighted currents but uses only two values of resistors, namely R and 2R.
- It uses a ladder network containing series-parallel combinations of two resistors of values Rand 2R.

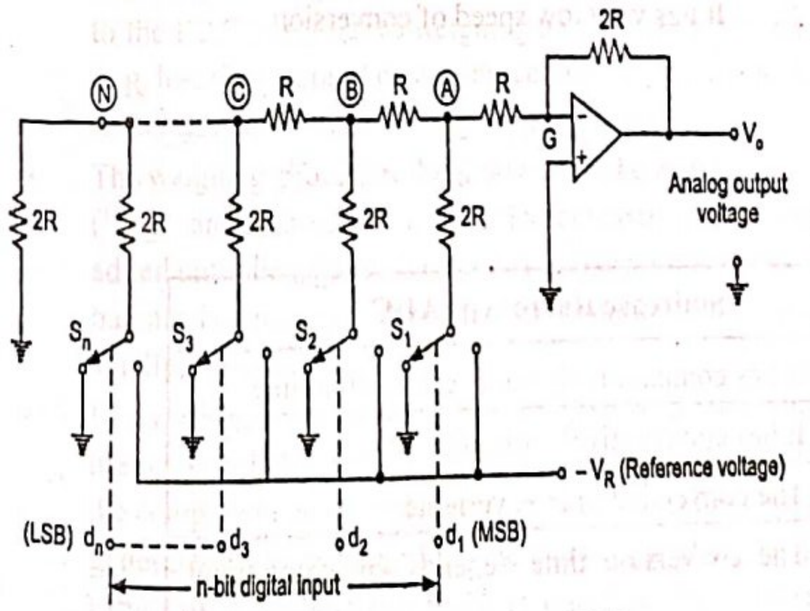


Fig. : A n-bit R-2R ladder type DAC

- Fig. shows the circuit diagram of a binary ladder type D/A converter with sets of identical resistors R and 2R.
- It consists of a R-2R ladder network and op-amp inverting amplifier.
- The value of resistor R can be between 2.5 kΩ to 10 kΩ.
- The resistor 2R can either be connected to the reference voltage ($-V_R$) line or grounded through controlled switches $S_1, S_2, S_3, ..., S_n$.
- The simplified circuit of a 3-bit (d_1, d_2, d_3 = 100) binary ladder type DAC is shown in Fig.. This simplified circuit is further reduced to the equivalent circuit shown in Fig. The equivalent resistance to the left of node (A) in Fig. is only 2R and the node G is at virtual ground potential.

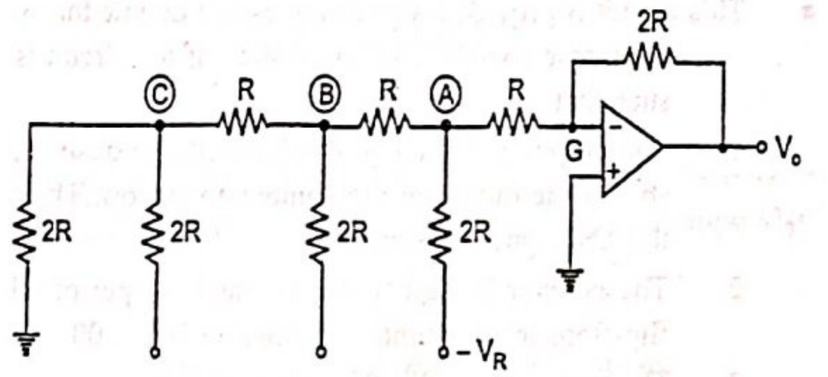


Fig. : A simplified 3-bit R-2R ladder type PAC

- As the two resistors R and 2R are in parallel with each other, their parallel combination results in a resistance of 2R/3.
- The voltage at the node (A) is given by

$$V_A = \frac{\frac{2R}{3}}{\left[2R + \left(\frac{2R}{3}\right)\right]} \times (-V_R) = \frac{\frac{2R}{3}}{\frac{8R}{3}} \times (-V_R) = -\frac{V_R}{4}$$

- The ouput voltage of an op-amp inverting amplifier is given by :

$$V_0 = -\left(\frac{R_F}{R}\right)V_i = -\left(\frac{2R}{R}\right) \times \left(\frac{-V_R}{4}\right) = +\frac{V_R}{2}$$

- For a digital input of $d_1 d_2 d_3 = 100$, the analog output produced is $\frac{V_R}{2}$.

Q.12. Define analog to digital converter and explain its type ?

Or Define analog to digital converter write its type and explain briefly one of them ?

Ans. In electronics, an analog-to-digital converter is a system that converts an analog signal, such as a sound picked up by a microphone or light entering.

Types of ADC :

- Analog to Digital converters are of following types :
  (A) Ramp (Staircase Ramp) type ADC.
  (B) Dual Slope ADC.
  (C) Successive Approximation type ADC.

Dual-Slope Analog to Digital Converter :

- Both the drawbacks of a single-slope ADC are overcome in the Dual-Slope ADC.

Principle of Operation :

- Fig. shows the functional block diagram of a Dual-Slope ADC. It consists of four major blocks:
  1. an integrator, 2. a comparator, 3. a binary counter and 4. a switch driver, T flip-flop.
- This circuit is provided with a single-pole double throw electronic switch. The initial state of the circuit is such that :
  1. The output of the integrator is small and positive, so that the output of the comparator is low. Thus, the AND gate is disabled.
  2. The counter is kept reset, so that Y output of all flip-flops in the counter are reading 000 ... 00.
  3. The toggle mode flip-flop is kept reset.

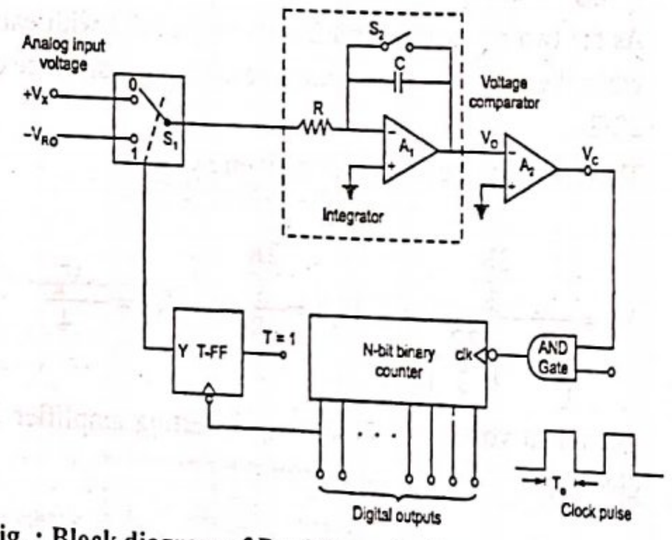


Fig. : Block diagram of Dual Slope Analog to Digital Converter

- The conversion process begins at t = 0 with the switch Si in position 0, thereby connecting the analog voltage V_x to the input of the integrator. The integrator output is :

$$V_o = \frac{-1}{\tau}\int_0^t V_x dt = -\left(\frac{V_a}{\tau}\right)t$$

- This results in high V_C, thus enabling the AND gate and the clock pulses reach the clock (clk) input terminal of the counter which was initially clear. The counter counts from 00 ... 00 to 11 ... 111 when $2^N - 1$ clock pulses are applied.
- At the next clock pulse 2^N, the counter is cleared and Q becomes 1. This controls the state of S_1 which now moves to position 1 at T_1, thereby connecting $-V_R$ to the input of the integrator. The output of the integrator now starts to move in the positive direction. The counter continues to count until $V_0 < 0$. As soon as V_0 goes positive at T_2, V_c goes low enabling the AND gate. The counter will stop counting in the absence of the clock pulses. The waveforms of voltages V_0 and V_c are shown in Fig.

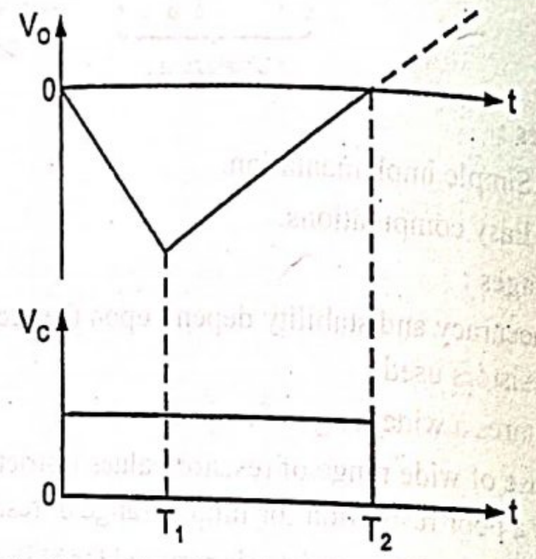


Fig. : Waveforms of Dual-Slope Analog to Digital Converter

Advantages :

1. It is simple and relatively inexpensive.
2. It has high conversion accuracy.
3. It is more stable and of low cost.
4. It is not affected by time, temperature and input voltage.
5. It does not require crystal oscillator for stability.
6. It is less sensitive to noise.

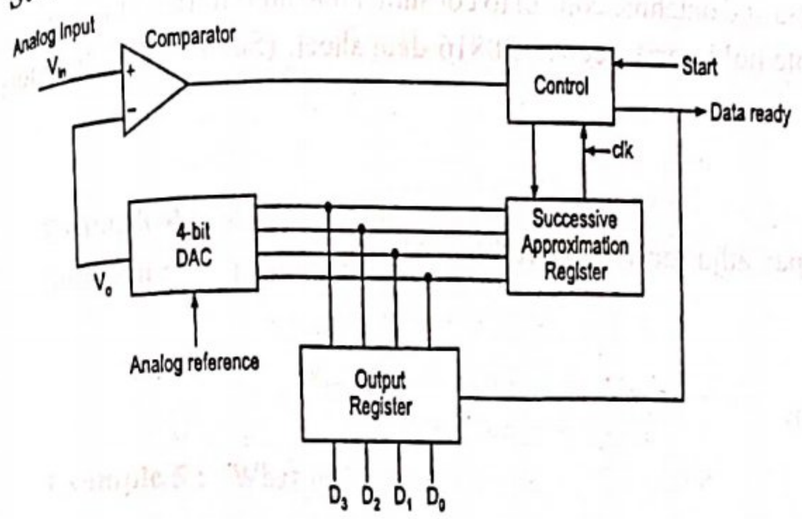
Disadvantages :

1. It has large conversion time as compared to any other ADC.
2. It has very low speed of conversion.

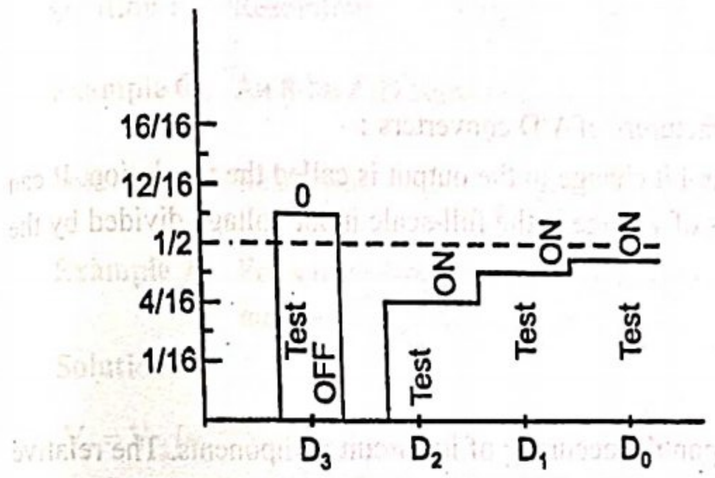
Comparison of Dual Slope and Ramp type ADC

Dual Slope ADC	Staircase Ramp type ADC
1. It has large conversion time.	1. It has comparatively lower conversion time.
2. It has very good accuracy.	2. It has comparatively low accuracy.
3. The conversion time is constant.	3. The conversion time is variable.
4. The conversion time is independent of analog input.	4. The conversion time depends on analog input.

Successive Approximation Type ADC :



(a) Block diagram of Successive Approximation ADC



(b) Conversion process for a 4-bit converter

Fig. : Successive Approximation Analog to Digital Converter

- Fig. (a) shows the block diagram of a successive approximation ADC. It includes three major elements : the DAC, the successive approximation register (SAR) and the comparators.
- The conversion technique involves comparing the output of the DAC V_0 with the analog input signal V_{in}. The digital input to the DAC is generated by using the successive approximation method. When the DAC output matches the analog signal, the input to DAC is the equivalent digital signal.
- The successive approximation method of generating input to the DAC is similar to weighing an unknown material (e.g. less than 1 gram) on a chemical balance with a set of such fractional weight as $^1/_2$ g, $^1/_4$ g, $^1/_8$ g etc.
- The weighing procedure begins with the heaviest weight ($^1/_2$ g) and subsequent weight (in decreasing order) are added until the balance is tipped. The weight that tips the balance is removed and the process is continued until the smallest weight is used.
- In the case of a 4-bit ADC, bit D_3 is turned on first and the output of DAC is compared with an analog signal. If the comparator changes the state, indicating that the output generated by D_3 is larger than the analog signal, bit D_3 is turned off in the SAR and bit D_2 is turned on. The process continues until the input reaches bit D_0.
- Fig. (b) illustrates a 4-bit conversion process. When bit D_3 is turned on, the output exceeds the analog signal and therefore bit D_3 is turned off. When the next three successive bits are turned on, the output becomes approximately equal to the analog signal.

Advantages :

1. It is more simple.
2. It has low resolution.
3. It has high speed of conversion.
4. It has constant conversion time.
5. It can produce n-bit of A/D conversion with only n-clock pulses.

Disadvantages :

1. The logic circuit is complex.
2. It is more expensive.
3. It requires DAC.

Comparison of Dual Slope and Successive Approximation Methods of ADC

	Dual Slope ADC	Successive Approximation ADC
1.	It has large conversion time.	It has moderately large conversion time.
2.	It is faster.	It is slower.
3.	It is very cheap.	It has moderate cost.
4.	It is used in all those applications which need very high accuracy.	It is used in data acquisition system.

Q.8. Explain 8-Bit up Compatible A/D Converters with 8-Channel Multiplexer ?

Ans. General Description :

- The ADC0808, ADC0809 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel muitiplexer and microprocessor compatible control logic, The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8-single-ended analog signals.
- The device eliminates the need for external zero and full-scale adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and latched TTL TRISTATE outputs.
- The design of the ADC0808, ADC0809 has been optimized by incorporating the most desirable aspects of several A/D conversion techniques. The ADC0808, ADC0809 offers high speed, high accuracy, minimal

temperature dependence, excellent long-term accuracy and repeatability, and consumes minimal power. These features make this device ideally suited to applications from process and machine control to consumer and automotive applications. For 16-channel multiplexer with common output (sample/hold port), see ADC0816 data sheet. (See AN-247 for more information.)

Features :

1. Easy interface to all microprocessors.
2. Operates ratiometrically or with 5 V DC or analog span adjusted voltage reference.
3. No zero or full-scale adjust required.
4. 8-channel multiplexer with address logic.
5. 0 V to 5 V input range with single 5 V power supply.
6. Outputs meet TTL voltage level specifications.
7. Standard hermetic or molded 28-pin DIP package.
8. 28-pin molded chip carrier package.

Specifications of ADC :

- The following specifications are usually specified by the manufacturers of A/D converters :

1. **Resolution :** The voltage input change necessary for a one-bit change in the output is called the resolution. It can also be expressed as a percentage. The resolution in terms of voltage is the full-scale input voltage divided by the total number of bits.

$$\% \text{ Resolution} = \frac{V_{FS}}{2^n - 1} \times 100$$

2. **Accuracy :** The accuracy of the A/D converter depends upon the accuracy of its circuit components. The relative accuracy of an A/D converter is the maximum deviation of the digital output from the ideal linear line.
3. **Conversion time :** The conversion time is the time required for conversion from an analog input voltage to the stable digital output. This conversion time is also called as speed.
4. **Linearity :** Linearity is conventionally equal to the deviation of the performance of the converter from a best straight line.
5. **Differential linearity :** The differential linearity is defined as the maximum amount of voltage change necessary to cause the digital output to change one bit minus the ideal voltage change necessary to change one bit.
6. **Monotoxicity :** In response to a continuously increasing input .signal, the output of an A/D converter should not at any point decrease or skip one or more codes. This is called the monotoxicity of the A/D converter.
7. **Analog input voltage :** This is the maximum allowable input voltage range.
8. **Format of Digital output :** An A/D converter can be made for any standard digital code.
9. **Quantization error :** The approximation process is known as quantization. The error due to the quantization process is known as quantization error.

SOLVED PROBLEMS

Example 1: The LSB of a 3-bit DAC represents 0.2 V. What value of voltage will be represented by the following binary words : (i) 011, (ii) 110 ?

Solution : The weight of the LSB is 0.2 V. Therefore, the weights of the other bits are 0.4 and 0.8.
(i) The output voltage for input 011 will be $V_0 = 0 + 0.4\text{ V} + 0.2\text{ V} = 0.6\text{ V}$
(ii) The output voltage for input 110 will be $V_0 = 0.8 + 0.4 + 0 = 1.2\text{ V}$

Example 2 : The LSB of 3-bit DAC is 0.5 V. What value of voltage will be represented by the following codes : (i) 011, (ii) 110 ?

Solution : The weight of the LSB is 0.5 V. Therefore, the weights of the other bits are 0.25 V and 0.125 V respectively.
(i) The output voltage for input Oil will be $V_0 = 0 + 0.25\text{ V} + 0.5\text{ V} = 0.75\text{ V}$
(ii) The output voltage for input 110 will be $V_0 = 0.125 + 0.25 + 0 = 0.375\text{V}$

Example 3 : What is the full-scale output of a 5-bit binary weighted resistor for logic '0' = 0 V and logic ' 1' = +10 V?

Solution : The full-scale output for DAC is given as :

$$V_{FS} = \left[\frac{2^N - 1}{2^N}\right] \times V \quad \text{where } N = 5 = \left[\frac{2^5 - 1}{2^5}\right] \times 10 = 9.6875\text{ V}$$

Example 4 : A 10-bit DAC has a step size of 5 mV. Calculate the full-scale output voltage.

Solution : Total number of steps $= 2^N - 1$
$= 2^{10} - 1 = 1023$
V_{FS} = Total number of steps × Step size
$= 1023 \times 5\text{ mV} = 5.115\text{ V}$

Example 5 : What is the resolution in volts, for a 6-bit DAC, with +10 V full-scale output voltage ?

Solution : Resolution $= \frac{1}{2^N - 1} \times V_{FS} = \frac{1}{2^6 - 1} \times 10 = 0.1587\text{ V}$

Example 6 : An 8-bit A/D converter has a maximum voltage of 15 V. What voltage change would each bit represent ?

Solution : Resolution $= \frac{V_{FS}}{2^n - 1} = \frac{15}{2^8 - 1} = 58.82\text{ mV}$

Example 7 : For a 6-bit binary R-2R ladder, assume '0' = 0 V and '1' = +10 V. Find the output voltage for following digital inputs : (i) 1010001, (ii) 0011001, (iii) 111011, (iv) 100111

Solution :

$$V_0 = V_{FS}[d_1 2^{-1} + d_2 2^{-2} + d_3 2^{-3} + d_4 2^{-4} + d_5 2^{-5} + ... + d_n 2^{-n}]$$

(i) **101001 :**

$$V_0 = 10[1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} + 0 \times 2^{-5} + 1 \times 2^{-6}] = 6.40625\text{ V}$$

(ii) **0011001 :**

$$V_0 = 10[0 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4} + 0 \times 2^{-5} + 0 \times 2^{-6} + \times 2^{-7}] = 1.953\text{ V}$$

(iii) **111011 :**

$$V_0 = 10[1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} + 1 \times 2^{-5} + 1 \times 2^{-6}] = 9.21875\text{ V}$$

(iv) **100111 :**

$$V_0 = 10[1 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} + 1 \times 2^{-5} + 1 \times 2^{-6}] = 6.09375\text{ V}$$

Example 8 : Calculate the analog output for 5-bit weighted resistor type DAC for inputs : (i) 10110, (ii) 10001. Assume logic '0' = 0 V and logic '1' = 10 V.

Solution : $V_0 = V_R[d_1 2^{-1} + d_2 2^{-2} + d_3 2^{-3} + d_4 2^{-4} + d_5 2^{-5}]$

(i) **10110 :**

$$V_0 = 10[1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4} + 0 \times 2^{-5}]$$
$$= 6.875\text{ V}$$

(ii) **10001 :**

$$V_0 = 10[1 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-3} + 0 \times 2^{-4} + 1 \times 2^{-5}]$$
$$= 5.31$$

Example 9 : A 4-bit D/A converter produces an output voltage of 4.5 V for an input code of 100%. What will be the value of output voltage for an input code of 0011 ?

Solution : $V_0 = V_{FS}[d_1 2^{-1} + d_2 2^{-2} + d_3 2^{-3} + d_4 2^{-4}]$

$$= 4.5[0\times2^{-1}+0\times2^{-2}+1\times2^{-3}+1\times2^{-4}]$$

$$= 0.84375 \text{ V}$$

Example 10 : If an 8-bit A/D converter is driven by a 200 kHz clock, find (1) Maximum conversion time, (2) Average conversion time.

Solution : $T_{max} = (2^n - 1)$ clock periods $= \frac{2^n - 1}{F} = \frac{2^8 - 1}{200} = 1.275$ m sec.

Example 11 : The LSB of 3-bit DAC represents 0.2 V. What value will be represented by following binary words ? (i) 100, (ii) 110 ?

Solution : Let the 3-bit digital word to the DAC be $d_1d_2d_3$. d_1 is the MSB and d_3 is the LSB.

$$V_0 = V_R [d_1 2^{-1} + d_2 2^{-2} + d_3 2^{-3}]$$

The LSB intends the smallest change i.e. the resolution is 0.2 V.

Therefore the output for (001) is 0.2 V.

$$V_R = \frac{V_0}{[d_1 2^{-1} + d_2 2^{-2} + d_3 2^{-3}]} = \frac{0.2}{[0\times2^{-1}+0\times2^{-2}+1\times2^{-3}]} = 1.6\text{V}$$

1. 100 $\quad V_0 = V_R [d_1 2^{-1} + d_2 2^{-2} + d_3 2^{-3}] = 1.6[1\times2^{-1}+0\times2^{-2}+0\times2^{-3}] = 0.8\text{V}$

2. 110 $\quad V_0 = V_R [1\times2^{-1}+1\times2^{-2}+0\times2^{-3}] = 1.2\text{V}$

□□□

OBJECTIVE TYPE QUESTIONS

1. **Computer memory is measured in terms of :**
   (a) bytes
   (b) nibble
   (c) bits
   Ans. (a)
2. **A hard disk can store upto**
   (a) Terabyte
   (b) Gigabyte
   (c) megabyte
   Ans. (a)
3. **1 Kilobyte (KB) =**
   (a) 1024 bytes
   (b) 4 bits
   (c) 8 bits
   Ans. (a)
4. **Raw facts that have been collected but not organized are known as :**
   (a) Data
   (b) None of these
   (c) Information
   Ans. (a)
5. **Primary memory is of ..... types - ..... and ..... .**
   (a) two, RAM, ROM
   (b) TWO, CD, DVD
   (c) three, RAM and ROM
   Ans. (a)
6. **1 Terabyte (TB) =**
   (a) 1024 GB
   (b) 1024 KB
   (c) 1024 MB
   Ans. (a)
7. **How many types of ROM are there ?**
   (a) three
   (b) four
   (c) two
   Ans. (a)
8. **Binary 0 and 1 represents two states ..... and ..... .**
   (a) OFF, ON
   (b) None of these
   (c) ON, OFF
   Ans. (A)
9. **..... is used for reading / writing information from a CD.**
   (a) CD drive
   (b) pen drive
   (c) None
   Ans. (a)
10. **DVD storage capacity varies from ..... to ..... .**
    (a) 4 GB to 17 GB
    (b) 2 GB to 17 GB
    (c) 4 GB to 18 GB
    Ans. (a)
11. **The full form of ROM is :**
    (a) Read only memory
    (b) Random only memory
    (c) Read one memory
    Ans. (a)
12. **DVD stands for :**
    (a) Digital versatile Disc
    (b) Digital Versatile Device
    (c) Digital Versatile Drive
    Ans. (a)
12. **A ..... is a portable storage device.**
    (a) Compact Disc
    (b) USB Flash Drive
    (c) Digital Video disc
    Ans. (a)
13. **Which is used to record movies with high quality of video and sound :**
    (a) DVD
    (b) CD
    (c) USB flash drive
    Ans. (a)
14. **How much data a CD can store ?**
    (a) 650 MB to 900 MB
    (b) 450 MB to 900 MB
    (c) 650 MB to 900 GB
    Ans. (a)
15. **Which can be attached to any USB port ?**
    (a) USB flash drive
    (b) CD
    (c) DVD
    Ans. (a)
16. **1 Nibble = ... bits.**
    (a) 4
    (b) 8
    (c) 2
    Ans. (a)
17. **Full form of IPO is :**
    (a) Input, Process, Output
    (b) Information, Process, output
    (c) Process, Input, output
    Ans. (a)
18. **Which memory stores the data permanently ?**
    (a) ROM
    (b) Both
    (c) RAM
    Ans. (a)

19. Processed data are called ..... .
(a) None
(b) Information
(c) Input
Ans. (a)

20. A combination of 8 bits makes a :
(a) byte (b) gigabyte
(c) nibble
Ans. (a)

21. Which device store a large amount of data permanent
(a) secondary (b) storage
(c) primary
Ans. (a)

22. One byte is capable of holding ..... character.
(a) one (b) three
(c) two
Ans. (a)

□□□