

1. NUMBER SYSTEM, BOOLEAN ALGEBRA, AND LOGIC GATES

-: Binary Number System

Number \rightarrow Binary Number System

- \leftrightarrow It is \rightarrow base 2_{ten} - \leftrightarrow system

- \leftrightarrow It has \rightarrow symbol \rightarrow 0 & 1 \rightarrow represent number \rightarrow 0 & 1

- \leftrightarrow symbol of 0 & 1 are \rightarrow 0 & 1
 $'1'$ and ' 0 '

Positional System \rightarrow \leftrightarrow

- \leftrightarrow It is \rightarrow position \rightarrow group \rightarrow \leftrightarrow

- \leftrightarrow It is \rightarrow arrange \rightarrow weight

LSB \rightarrow Number system of

- \leftrightarrow MSB \rightarrow Left Most Bit

LSB \rightarrow Right Most Bit

- \leftrightarrow 1 0 1 0 \leftrightarrow

\leftrightarrow Nibble \rightarrow group of Four Bits

byte \rightarrow group of eight bits \rightarrow

- \leftrightarrow 1 0 1 0 \leftrightarrow

D_1	D_0	B_3	B_2	B_1	B_0	D_1	D_0	B_3	B_2	B_1	B_0
0	0	0	0	0	0	0	8	1	0	0	0
0	1	0	0	0	1	0	9	1	0	0	1
0	2	0	0	1	0	1	0	1	0	1	0
0	3	0	0	1	1	1	1	1	0	1	1
0	4	0	1	0	0	1	2	1	1	0	0
0	5	0	1	0	1	1	3	1	1	0	1
0	6	0	1	1	0	1	4	1	1	1	0
0	7	0	1	1	1	1	5	1	1	1	1

:- Octal Numbers System

(8) ~~is~~ base 8 numbers system
 System of Octal Numbers
 (or) System of 8 digits
 7 symbols 0, 1, 2, 3, 4, 5, 6
 represent numbers
 - 8 digits (0 to 7)

Octal	Binary	Decimal
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7
10	001000	8
11	001001	9
12	001010	10
13	001011	11
14	001100	12
15	001101	13
16	001110	14
17	001111	15

:- Hexadecimal Number System

'16' base. Base Number System

Hexadecimal Numbers

Count - 0 to 15 in system

All 16 digits represent 16 numbers.

16 digits symbols also 16

A to F 0 to 9 symbols also 16

alphanumeric numbers (A to F)

- 0 to 15 in system
Numbers or all in computers uses

- 16 digits system

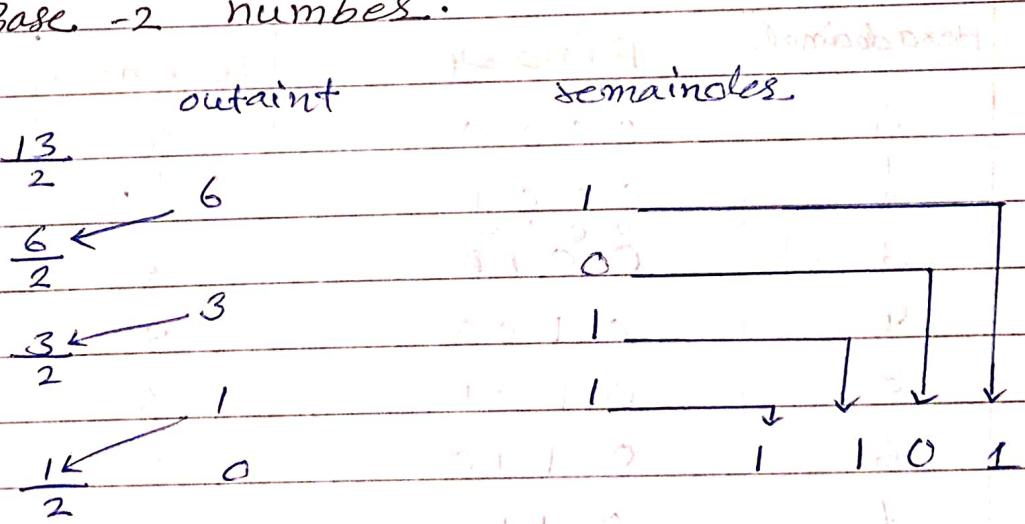
Hexadecimal	Binary	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Decimals to binary Conversion :-

Any decimal number can be converted into its equivalent binary number. For integers, the conversion is obtained by continuous division by 2 and keeping track of the remainders. While for fractional part, the conversion is affected by continuous multiplication by 2 and keeping track of the integers.

- ① Ex:- Convert $(13)_{10}$ to an equivalent Base -2 numbers.

Solⁿ :



thus,

$$(13)_{10} = (1101)_2$$

Ex ② Convert $(0.68625)_{10}$ to an equivalent Base-2 number.

$$\begin{array}{ccccccc}
 \text{Sal}^n: & 0.65625) & \xrightarrow{x_2} & 0.31250 & \xrightarrow{x_2} & 0.62500 & \xrightarrow{x_2} 0.25000 \\
 & \downarrow & & \downarrow & & \downarrow & \downarrow \\
 & 1 & & 0 & & 1 & 0
 \end{array}$$

0.50000

$$\frac{x_2}{1.0000}$$

Thus,

$$(0.68625)_{10} = (010101)_2$$

Ex :- Express the following decimal numbers in the binary form.

- @ 28.5 (b) 10.625 (c) 0.6875

2.2 Integers Past

$$\begin{array}{r} \underline{25} \\ \underline{2} \\ 12 \end{array}$$

quotient

semaines

$$\frac{t}{2} \leftarrow$$

1

1

1

1

These fore,

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

Fractional Part

$$\begin{array}{r} .5 \\ \times 2 \\ \hline 1.0 \\ \downarrow \\ 0 \end{array}$$

ie $(0.5)_{10} = (0.1)_2$

These fore, $(25.5)_{10} = (11001.1)_2$

(b) Integers Part

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

Fractional Part

$$\begin{array}{r} .625 \\ \times 2 \\ \hline 1.250 \\ \downarrow \\ 1 \end{array} \quad \begin{array}{r} 0.250 \\ \times 2 \\ \hline 0.500 \\ \downarrow \\ 0 \end{array} \quad \begin{array}{r} 0.500 \\ \times 2 \\ \hline 1.000 \\ \downarrow \\ 1 \end{array}$$

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

These for $(10.625)_{10} = (1010.101)_2$

८

$$\begin{array}{r}
 .6875 \\
 \times 2 \\
 \hline
 1.3750
 \end{array}
 \quad
 \begin{array}{r}
 0.3760 \\
 \times 2 \\
 \hline
 0.7500
 \end{array}
 \quad
 \begin{array}{r}
 0.7500 \\
 \times 2 \\
 \hline
 1.5000
 \end{array}
 \quad
 \begin{array}{r}
 0.5000 \\
 \times 2 \\
 \hline
 1.0000
 \end{array}$$

$$\text{Therefore, } (0.6875)_{10} = (0.1011)_2$$

Binary - to Decimal Conversion :-

Any binary numbers can be converted into its equivalent decimal numbers using the weights assigned to each bit position.

Q Find the decimal equivalent of the
Binary numbers $(1111)_2$.

Sol": The equivalent decimal number is:

$$\begin{aligned}
 &= 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\
 &= 16 + 8 + 4 + 2 + 1 \\
 &= (31)_{10}
 \end{aligned}$$

Ex.: Determine the decimal numbers represented by the following binary numbers.

(a) $(110101)_2 =$

$$\begin{aligned} &= 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 32 + 16 + 0 + 4 + 0 + 1 \\ &= (53)_{10} \end{aligned}$$

(b) $(101101)_2 =$

$$\begin{aligned} &= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 32 + 0 + 8 + 4 + 0 + 1 \\ &= (45)_{10} \end{aligned}$$

(c) $(1111111)_2 =$

$$\begin{aligned} &= 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + \\ &\quad 1 \times 2^1 + 1 \times 2^0 \\ &= 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 \\ &= (255)_{10} \end{aligned}$$

(d) $(00000000)_2 =$

$$\begin{aligned} &= 0 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + \\ &\quad 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 \\ &= (0)_{10} \end{aligned}$$

Q Determine the decimal numbers

represented by the following binary numbers.

(a) 101101.10101

Soln:-

$$\begin{aligned}(101101.10101)_2 &= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + \\&\quad 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} \\&\quad + 1 \times 2^{-5} \\&= 32 + 0 + 8 + 4 + 0 + 1 + \frac{1}{2} + 0 + \frac{1}{8} + 0 + \frac{1}{32} \\&= (45.65625)_{10}\end{aligned}$$

(b) $(1100.1011)_2 =$

$$\begin{aligned}&= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} \\&\quad + 1 \times 2^{-3} + 1 \times 2^{-4} \\&= 8 + 4 + \frac{1}{2} + \frac{1}{8} + \frac{1}{16} \\&= (12.6875)_{10}\end{aligned}$$

(c) $(0.10101)_2$

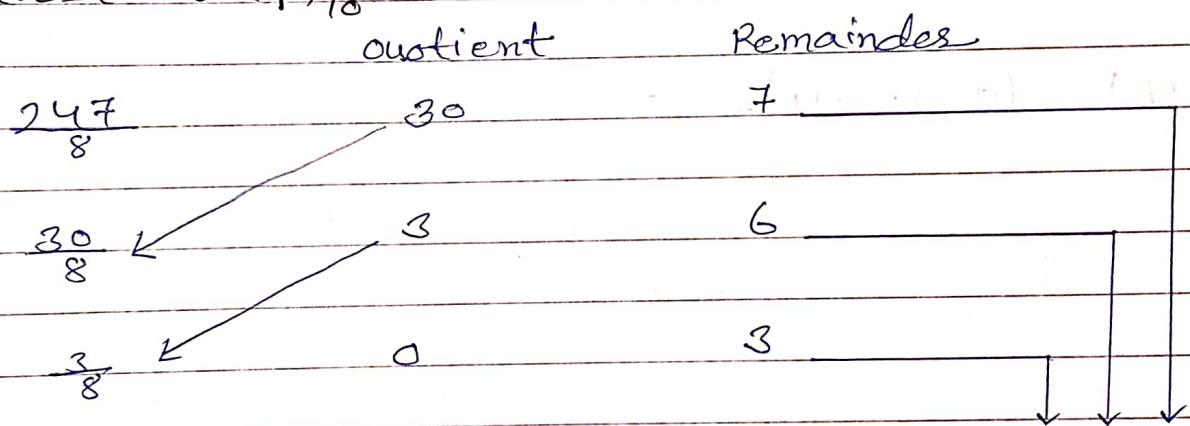
$$\begin{aligned}&= 0 \cdot 5 + 0 + 0 \cdot 125 + 0 + 0 \cdot 3125 \\&= (0.6875)_{10}\end{aligned}$$

Decimal - to - Octal Conversion :-

The conversion from decimal to octal (base-10 to base-8) is similar to the conversion procedure for base-10 to base-2 conversion.

The only difference is that number 8 is in place of 2 for division in the case of integers and for multiplication in the case of fractional numbers.

Ex Convert $(247)_{10}$ into octal.



$$(247)_{10} = (367)_8$$

Ex Convert $(0.6878)_{10}$ into octal.

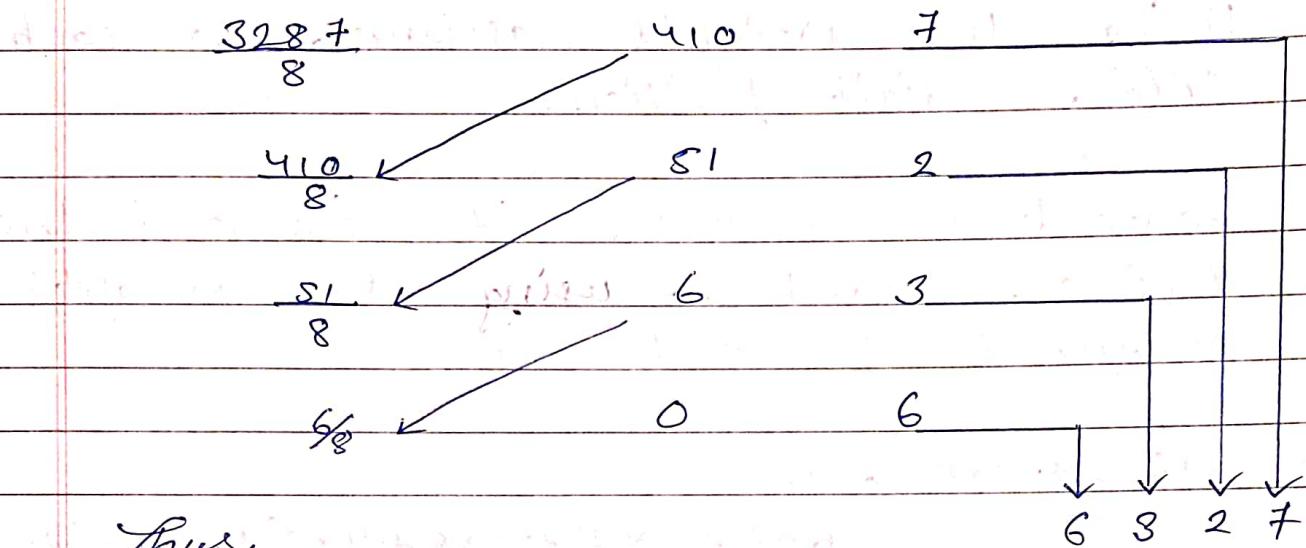
$$\begin{array}{r} 0.6878 \\ \times 8 \\ \hline 5.5000 \\ \downarrow \\ 5 \end{array} \quad \left. \begin{array}{r} 0.5000 \\ \times 8 \\ \hline 4.0000 \\ \downarrow \\ 4 \end{array} \right\}$$

thus,

$$(0.6878)_{10} = (0.54)_8$$

Ex:- Convert $(0.6875)_{10}$ into octal.

Integers Part Subtiant remainders



Thus,

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

Fractional Part:

$$0.6875 \xrightarrow{\times 8} 0.5100098 \xrightarrow{\times 8} 0.40800784 \xrightarrow{\times 8} 0.6406272 \xrightarrow{\times 8} 0.51250176$$

$$\text{Thus, } (0.6875)_{10} = (0.5100098)_8$$

$$\text{Therefore, } (3287.6875)_{10} = (6327.4087)_8$$

From the above examples we observe that conversion for fractional numbers may be equal.

Octal - to - decimal Conversion :-

Any octal number can be converted into its equivalent decimal number. Using the weights assigned to each octal digit position.

Ex:- Convert $(6327.4081)_8$ into its equivalent decimal number. The weights assigned to each di

$$\begin{aligned}
 \text{Soln:- } (6327.4081)_8 &= \\
 &= 6 \times 8^3 + 3 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + 4 \times 8^{-1} + \\
 &\quad 0 \times 8^{-2} + 8 \times 8^{-3} + 1 \times 8^{-4} \\
 &= 3072 + 192 + 16 + 7 + \frac{4}{8} + 0 + \frac{8}{8192} + \frac{1}{4986} \\
 &= (3287.8100098)_{10}
 \end{aligned}$$

Thus,

$$(6327.4081)_8 = (3287.8100098)_{10}$$

Decimal to Hexadecimal Conversion:-

For conversion from decimal to hexadecimal, the procedure used in binary and well as octal systems is applicable. Using 16 as the dividing factor (for integer part) and multiplying (for fractional part) factor.

Ex :- Convert the following decimal numbers
into hexadecimal numbers.

(a) 95.5

Soln :- Integers, part

95	outient	remainders
16	5	15
<u>5</u>	0	5
16		

Thus

$$(95)_{10} = (5F)_{16}$$

Fractional Part

$$\begin{array}{r}
 .5 \\
 \times 16 \\
 \hline
 8.0
 \end{array}$$

↓

8

$$\text{Thus } (0.5)_{10} = (0.8)_{16}$$

Therefore $(95.5)_{10}$

$$= (5F.8)_{16}$$

Ans.

(b) Integers part 675.625

Quotient Remainder

$$\begin{array}{r} 675 \\ \times 16 \\ \hline 42 \\ 42 \\ \hline 2 \\ 2 \\ \hline 0 \\ 10 \\ 10 \\ \hline 2 \\ 2 \\ \hline 3 \\ 3 \\ \hline \end{array}$$

Thus $(675)_{10} = (2A3)_{16}$

Fractional Part

$$\begin{array}{r} 0.625 \\ \times 16 \\ \hline 10.000 \\ \downarrow \\ A \end{array}$$

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

Therefore

$$(675.625)_{10} = (2A3.A)_{16}$$

Hexa decimal - to - Decimal Conversion :-

Hexa decimal numbers can be converted to their equivalent decimal numbers.

Ex:- Obtain decimal equivalent of hexadecimal numbers. $(3A.2F)_{16}$.

Soln:-

$$\begin{aligned}(3A.2F)_{16} &= 3 \times 16^1 + 10 \times 16^0 + 2 \times 16^{-1} + 15 \times 16^{-2} \\&= \frac{48}{16} + \frac{10}{16} + \frac{2}{16} + \frac{15}{16} \\&= (58.1836)_{10}\end{aligned}$$

Note:- These fractional part may be an exact equivalent and therefore, may give a small error.

Binary to octal Conversion:-

Binary numbers can be converted into equivalent octal numbers by maximum group of these bits starting from LSB and moving forwards MSB for integer part of the number and then replacing each group of these bits by its octal representation.

for fractional part, the groupings of these bits are made starting from the binary points.

Ex: Convert $(1001110)_2$ to its octal equivalent.

$$\begin{aligned}(1001110)_2 &= (\dots)_8 \\ &= (116)_8 \\ &= (1.16)_8\end{aligned}$$

Ex: Convert $(0.10100110)_2$ to its equivalent octal number.

$$\begin{aligned}(0.10100110)_2 &= (0.101001100)_2 \\ &= (0.514)_8\end{aligned}$$

Ex: Convert the following binary numbers to octal numbers.

(a) 1100110001.00010111001

Solⁿ $\begin{array}{r} 011 \ 001 \ 110 \ 001 \cdot 000 \ 101 \ 111 \ 001 \\ \underline{\quad}\ \underline{\quad}\ \underline{\quad}\ \underline{\quad}\ \underline{\quad}\ \underline{\quad}\ \underline{\quad}\ \underline{\quad} \\ (3 \ 1 \ 6 \ 1 \ . 0571)_8 \end{array}$

(b) $001 \ 011 \ 011 \ 110 \cdot 110 \ 010 \ 100 \ 110$

$$(4336.8246)_8$$

(c) $111110.001 \cdot 1001100110010.010$

$$(781.4632)_8$$

Octal - to - Binary Conversion :-

Octal numbers can be converted into equivalent binary numbers by replacing each digit by its 3-bit equivalent binary.

Octal Binary Decimal

0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7
10	001000	8
11	001001	9
12	001010	10
13	001011	11
14	001100	12
15	001101	13
16	001110	14
17	001111	15

Ex :- Convert $(736)_8$ into an equivalent binary no.

Soln

$$(736)_8 = (111011110)_2$$

Binary - to Hexadecimal Conversion :-

Binary Numbers can be converted into the equivalent hexadecimal numbers by making groups of four bits starting from LSB and moving towards MSB for integers part and then replacing each group of four bits by its hexadecimal representation.

For the fractional part, the above procedure is repeated starting from the bit next to the binary point and moving towards the right.

Ex:- Convert the following binary numbers to their equivalent hexadecimal numbers.

$$(a) \quad 10100110101111 \quad \text{Ans: } 29AF$$

$$\text{Soln} \quad (0010100110101111)_2 = 11001100$$

$$\therefore (10100110101111)_2 \underset{1100}{\approx} (29AF)_{16}$$

$$(b) \quad (0.0001110101101)_2 = 0.0001110101101$$

$$= (0.1EB4)_{16}$$

Ex Convert the binary numbers into Hexadecimal Numbers.

(a) 0110 0111 0001 . 0001 0111 1001

$$= (671.179)_{16}$$

(b) 0010 1001 1110 . 1100 1010 0110

$$= (2DE.CAG)_{16}$$

(c) 0001 1111 0001 . 1001 1001 1010

$$= (1F1.99A)_{16}$$

Ex:- Hexadecimal - to - Binary Conversion :-

Hexadecimal numbers can be converted into equivalent binary numbers by replacing each hexa digit by its equivalent 4-bit binary numbers.

Ex: Convert $(2F9A)_{16}$ to equivalent binary numbers.

$$(2F9A)_{16} = (0010 1111 1001 1010)_2$$

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

Binary addition :-

Augend	Addend	Carry	Sum	Result
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	0	10

Ex :- Add the binary numbers.

(i) 1011 and 1100

(ii) 0101 and 1110

Binary subtraction :-

Minuend	Subtrahend	Borrow	Difference
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

Ex :- Perform the subtraction.

(i) 1011

$$\begin{array}{r} 0110 \\ - 0101 \\ \hline \end{array}$$

(ii) 1110

$$\begin{array}{r} 0111 \\ - 0111 \\ \hline \end{array}$$

Binary Multiplication :-

✓ decimal multiplication
- ✓ they are similar

Binary Division:-

similar ✓ decimal division ✓
- ✓ they

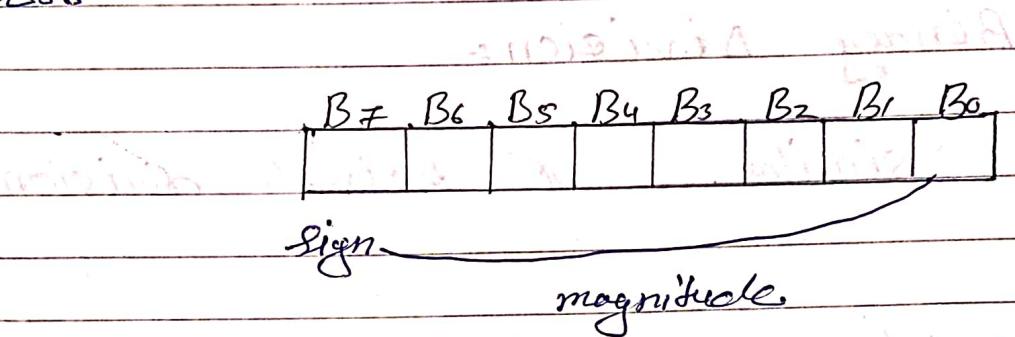
Ex :- Divide 1110101 by 1001.

Write its Complement and its Complement Numbers for a given binary number.

Signed Binary Numbers :-

Until now we have considered only Unsigned numbers, numbers without any +ve or -ve sign. These Unsigned numbers represent only magnitude. In this section we discuss about signed binary numbers.

These numbers are represented by the sign magnitude format. The figure shows the sign magnitude format for 8 bit signed numbers.



-127 to 128

Here, the MSB represents sign of the numbers.

If MSB is 1, number is -ve. and
If MSB is 0, number is +ve. The Remaining bits represents magnitude of the numbers.

Here are some examples of sign-magnitude numbers.

$$21) +6 = 0000\ 0110$$

$$22) -14 = 1000\ 1100$$

$$23) +24 = 0001\ 1000$$

$$24) -64 = 1100\ 0000$$

In case of unsigned 8-bit binary numbers the decimal range is 0 to 255. For signed magnitude 8-bit binary numbers the longest magnitude is reduced from 255 to 127 because we need to represent both +ve and -ve numbers.

$$\text{Maximum +ve number} = 0111\ 1111 = +127$$

$$\text{Maximum -ve number} = 1111\ 1111 = -127$$

IS Complement Representation :-

This is complement of a binary number IS the number that results when we change all is to zeros and zeros to ones:

Ex :- find IS complement of $(1101)_2$

Soln :- $1101 \leftarrow \text{number}$

$0010 \leftarrow \text{IS complement.}$

Ex: Find 1's complement of 1011001

Soln: 1011001 ← number.

0100110 ← 1's complement.

ii. Complement Representation :-

The 1's complement is the binary number that results when we add 1 to the 1's complement. It is given as,

$$2^{\text{'s}} \text{ complement} = 1^{\text{'s}} \text{ complement} + 1$$

The 2's complement form is used to represent -ve numbers.

Ex: Find 2's complement of (1001)₂

Soln 1001 ← number

+ 0110 ← 1's complement
—————
0111 ← 2's complement

Ex: Find 2's complement of (1010 0011)₂

1010 0011 ← number

+ 0101 1100 ← 1's complement
—————
0101101 ← 2's complement.

- Perform subtraction of binary numbers
IS & i Complement Method.

When we subtract two signed binary numbers there are four possible cases.

Case 1: Both Positive.

Ex: Subtract 18 from 52

$$\begin{array}{r} 52 \\ - 18 \\ \hline 34 \end{array}$$

$$\begin{array}{r} 52 \\ + (-18) \\ \hline 34 \end{array}$$

i's Complement of 18

00010010 decimal 18

10101101 i's complement

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

11101110 i's complement

00110100

decimal 82

11101110

i's complement of 18

00010010

decimal 34

In 8bit addition carry after 8th bit is ignored. After addition of 52 decimal and i's complement of 18 we get result $(0010010)_2$, which is equivalent to decimal 34 and our answer is verified.

Case II :-

Positive and small negative.

Ex:- Subtract $(-\underline{12})$ from $\underline{48}$

$$\begin{array}{r} 48 \\ - (-\underline{12}) \\ \hline 60 \end{array}$$

0 0 110000

decimal 48

00 00 11 00

decimal 12

00 11 11 00

decimal 60

Case III :- Positive and large negative

Ex:- Subtract -80 from $+12$

$$\begin{array}{r} 12 \\ - (-\underline{80}) \\ \hline 62 \end{array}$$

0000 1100

decimal 12

0011 0010

decimal 80

0011 1101 10

decimal 62

Case IV :- Both negative

Ex:- Subtract -68 from -15

$$\begin{array}{r} -15 \\ - (-\underline{68}) \\ \hline 53 \end{array}$$

is Complement of 15.

0000 1111 numbers.

1111 0000 is complement

$$\begin{array}{r} + \\ \hline 1111 0001 \end{array}$$

is complement

1111 0001 i.e. complement of 15

+ 01000100 decimal 68

0 00110101 decimal 53

In 8 bit addition carry after 8th bit is ignored. After addition of i.e. complement of 15 and 68 we get result which is equivalent to decimal 53.

Q: S \Rightarrow + 0101 intended
 - 7. 1001, complement of subtractend
 - 2 1110

The final carry = 0, therefore, the answer is negative and in 2's complement form,

2's complement of

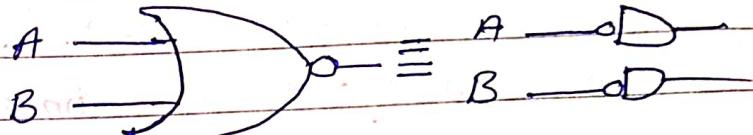
$$1110 = 0010$$

Hence, answer is $(-2)_{10}$

Demorgan's Law:

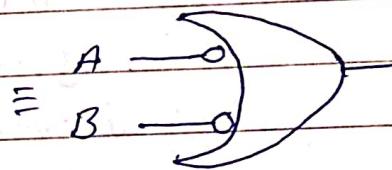
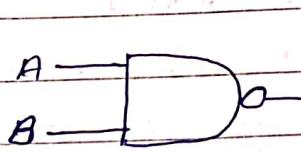
- ~~or~~ Lieg law \Rightarrow Compl.

$$\textcircled{1} \quad \overline{A+B} = \bar{A} \cdot \bar{B}$$



$$\textcircled{2} \quad \overline{A \cdot B} = \bar{A} + \bar{B}$$

NOR \equiv Bubbled AND



NAND \equiv Bubbled OR

$$\textcircled{3} \quad Y = \overline{A \cdot B \cdot C \cdot D}$$

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

$$\textcircled{4} \quad Y = (\overline{A+B}) + \bar{C}$$

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

$$\therefore Y = AC + BC$$

$$\textcircled{5} \quad Y = (A+B) \cdot \overline{C \cdot D} + E + F$$

$$= (A+B) \cdot \overline{CD} + E + F$$

$$= (\overline{A+B}) + \overline{CD} \cdot \bar{E} \cdot \bar{F}$$

$$= (\bar{A} \cdot \bar{B}) + \overline{CD} \cdot \bar{E} \cdot \bar{F}$$

"

Boolean algebraic theorem :-

$$\textcircled{1} \quad A + 0 = A$$

$$\textcircled{2} \quad A \cdot 1 = A$$

$$\textcircled{3} \quad A + 1 = 1, \quad 1 + \bar{A} = 1$$

$$\textcircled{4} \quad A \cdot 0 = 0$$

$$\textcircled{5} \quad A + A = A$$

$$\textcircled{6} \quad A \cdot A = A$$

$$\textcircled{7} \quad A + \bar{A} = 1$$

$$\textcircled{8} \quad A \cdot \bar{A} = 0$$

$$\textcircled{9} \quad A \cdot (B+C) = AB + AC$$

$$\textcircled{10} \quad A + BC = (A+B)(A+C)$$

From R.H.S, $A \cdot A + A \cdot C + B \cdot A + B \cdot C$

$$\Rightarrow A + A \cdot C + B \cdot A + B \cdot C$$

$$\Rightarrow A(1+C) + BA + BC$$

$$\Rightarrow A + BA + BC$$

$$\Rightarrow A(1+B) + BC$$

$$\Rightarrow A + BC = \text{L.H.S.}$$

$$\textcircled{11} \quad A + AB = A$$

from L.H.S $A(1+B)$

$$\Rightarrow A = \text{R.H.S}$$

$$\textcircled{12} \quad A \cdot (A+B) = A$$

From L.H.S $A \cdot A + AB$

$$\Rightarrow A + AB$$

$$\Rightarrow A(1+B) = A$$

$$\textcircled{13} \quad A + \bar{A}B = A+B$$

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

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

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

$$\textcircled{14} \quad A \cdot (\bar{A}+B) = AB$$

From L.H.S $A\bar{A} + AB$

$$AB = \text{R.H.S}$$

$$\textcircled{15} \quad AB + A\bar{B} = A$$

From L.H.S

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

$$\textcircled{16} \quad AB + \bar{A}C = AC + \bar{A}B$$

From R.H.S

$$A\bar{A} + AB + C\bar{A} + CB$$

$$\textcircled{17} \quad \underline{(A+B)(\bar{A}+C)} = AC + \bar{A}B$$

$$\textcircled{18} \quad \underline{A \cdot B \cdot C} = \bar{A} + \bar{B} + \bar{C}$$

$$\textcircled{19} \quad \underline{A + B + C} = \bar{A} \cdot \bar{B} \cdot \bar{C}$$

$$(18) \quad A \cdot B \cdot C = \overline{\overline{A} + \overline{B} + \overline{C}}$$

$$(19) \quad A + B + C = \overline{\overline{A} \cdot \overline{B} \cdot \overline{C}}$$

• Step for Converting to NAND/NOR logic:

Step 1- Draw AND/OR Logic.

Step 2- If NAND H/w has been chosen, Add bubbles on the o/p of each AND Gate, and Add bubbles on the S/P side of each all OR gate.

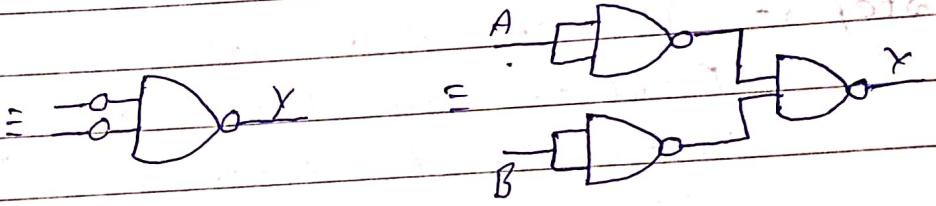
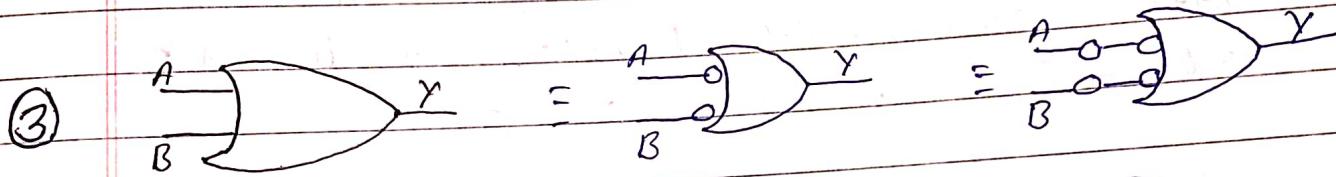
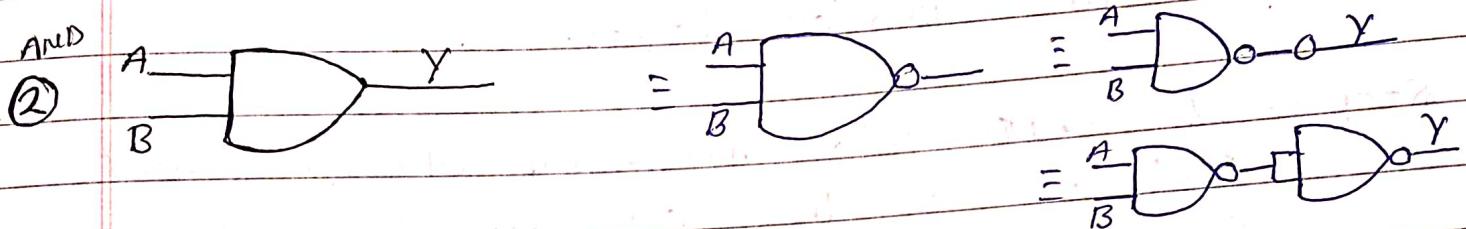
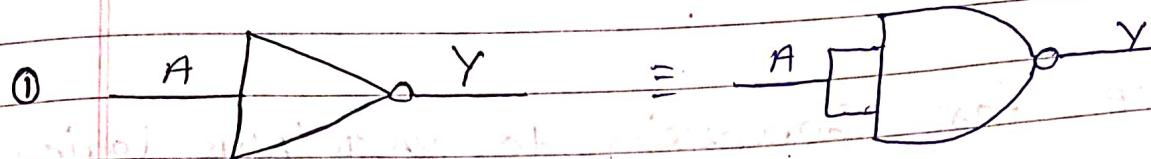
Step 3- If NOR H/w has been chosen, Add bubble on the o/p of each OR Gate, And Add bubble on the S/P side of each AND gate.

Step 4- Add or subtract an inverter on each line. that receive a bubble in step 2 or step 3.

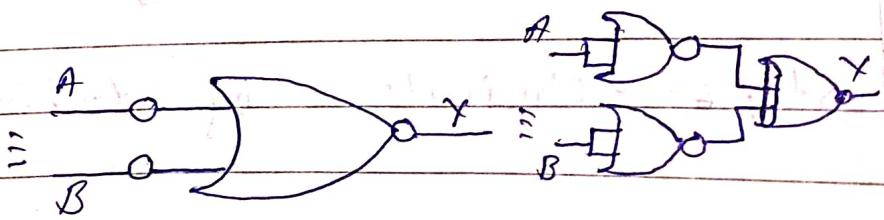
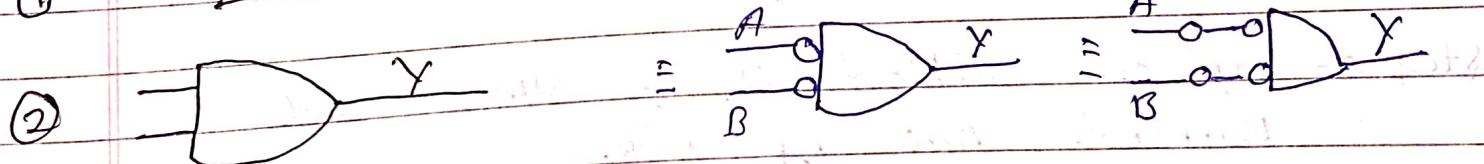
Step 5- Replace bubbled OR by NAND, and Bubbled AND by NOR.

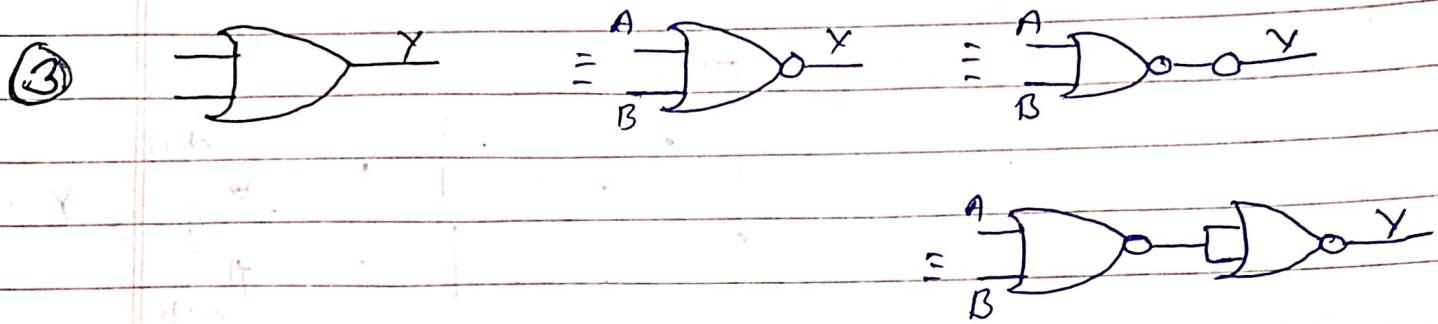
Step 6- Eliminate any double inversion.

Convert Basic Gates by NAND H/W.



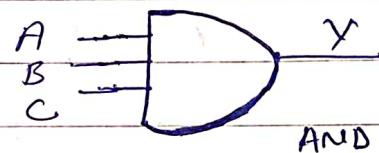
Convert Basic gates by NOR H/W.



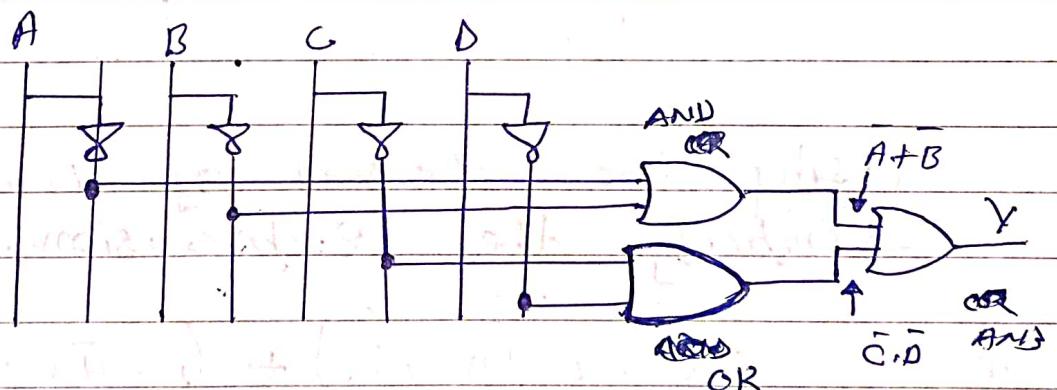


* Simplify the expression and draw the logic diagram.

$$\textcircled{1} \quad Y = (\bar{A} + \bar{B}) BC = (\bar{A} \cdot \bar{B}) BC = ABC = ABC$$

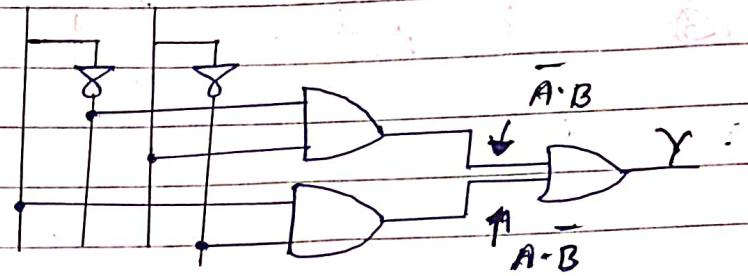


$$\textcircled{2} \quad Y = (A \cdot B) \cdot (C + D) = AB + (C + D) = (\bar{A} + \bar{B}) + (\bar{C} \cdot \bar{D}) \\ = \bar{A} + \bar{B} + \bar{C} \bar{D}$$



$$\textcircled{3} \quad Y = (\bar{A} + \bar{B}) \cdot (\bar{A} + B) = (\bar{A} + \bar{B}) + (\bar{A} + B)$$

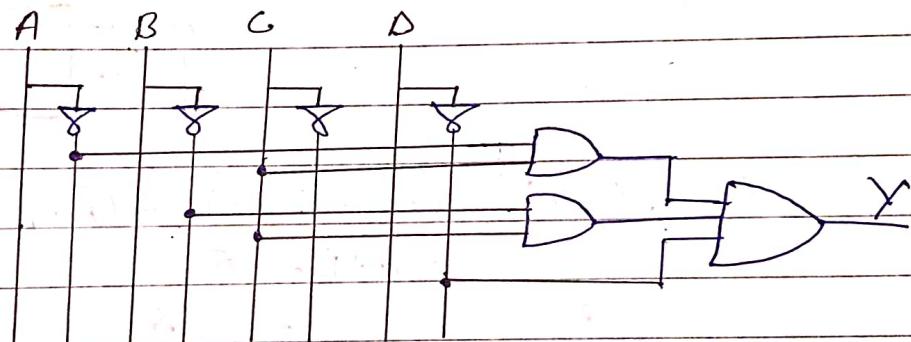
$$= \bar{A} \cdot \bar{B} + \bar{A} \cdot \bar{B} = \bar{A} \cdot \bar{B} + A \cdot \bar{B}$$



$$④ Y = \overbrace{A \cdot B \cdot C + D}^{\text{Simplified}}$$

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

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



Using the theorem of Boolean algebra
Simplify the Expression.

$$\begin{aligned}
 ① Y &= AB(\bar{D} + D\bar{C}) + (A + \bar{A}CD) + B \\
 &= \bar{A}B(\bar{D} + \bar{C}) + (A + CD) + B \\
 &= A + B - (\bar{D} + \bar{C}) + CD + B \\
 &\Leftarrow A + BD + BC\bar{C} + CD + B = A + B(\bar{D} + \bar{C} + 1) + CD \\
 &= A + B + CD
 \end{aligned}$$

$$\begin{aligned}
 ② Y &= A\bar{B}C + (\bar{B}+\bar{C})(\bar{B}+\bar{D}) + (\bar{A}+\bar{C}+\bar{D}) \\
 &= A\bar{B}C + \bar{B}\bar{B} + \bar{B}\bar{D} + \bar{C}\bar{B} + \bar{C}\bar{D} + \bar{A}\cdot\bar{C}\cdot\bar{D} \\
 &\stackrel{WKT \Rightarrow 1+C=1}{=} \bar{B}(A_C + \bar{D} + \bar{C} + 1) + \bar{C}\bar{D}(1+\bar{A}) = \bar{B} + \bar{C}\bar{D} \\
 &= \bar{B}(\bar{C} + \bar{A} + \bar{D} + 1) + \bar{C}\bar{D} = \bar{B} + \bar{C}\bar{D}
 \end{aligned}$$

$$③ \text{ Prove that: } A + \bar{A}B = A + B$$

from L.H.S

$$\begin{aligned}
 &\Rightarrow A(1+B) + \bar{A}B && WKT \Rightarrow 1+B=1 \\
 &\Rightarrow A + AB + \bar{A}B && WKT \Rightarrow A \cdot \bar{A}=0 \\
 &\Rightarrow AA + AB + A\bar{A} + \bar{A}B \\
 &\Rightarrow A(A+B) + \bar{A}(A+B) \\
 &= (A + \bar{A})(A+B) = A+B && R.H.S
 \end{aligned}$$

* Explain standard Representations for logical function : SOP and POS.

SOP \Rightarrow sum of Product

$$\text{Ex: } AB + BC + AC$$

POS \Rightarrow Product of sum

$$\text{Ex: } (A+B) \cdot (A+\bar{C}) \cdot (\bar{A}+\bar{C})$$

Standard SOP and POS Form :-

- Standard SOP :→

Q

$$Y = AC + BC + AB$$

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

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

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

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

Q

$$Y = A + AB + ABC$$

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

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

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

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

Q

Standard POS : \rightarrow sum of products

$$Y = (A+B) \cdot (B+C) \cdot (A+C)$$

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

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

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

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

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

$$Y = A \cdot (A+B) \cdot (A+B+C)$$

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

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

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

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

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

Minterms & Maxterms :-

Variables			Minterm (SOP)	Maxterm (POS)
A	B	C	m_i	m_i
0	0	0	$\bar{A}\bar{B}\bar{C} = m_0$	$A+B+C = m_0$
0	0	1	$\bar{A}\bar{B}C = m_1$	$A+B+\bar{C} = m_1$
0	1	0	$\bar{A}B\bar{C} = m_2$	$A+\bar{B}+C = m_2$
0	1	1	$\bar{A}BC = m_3$	$A+\bar{B}+C = m_3$
1	0	0	$A\bar{B}\bar{C} = m_4$	$\bar{A}+B+C = m_4$
1	0	1	$A\bar{B}C = m_5$	$\bar{A}+B+\bar{C} = m_5$
1	1	0	$AB\bar{C} = m_6$	$\bar{A}+\bar{B}+C = m_6$
1	1	1	$ABC = m_7$	$\bar{A}+\bar{B}+\bar{C} = m_7$

Q Minterm :-

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

$$= m_0 + m_1 + m_3 + m_4$$

$$y = \sum m(0, 1, 3, 4)$$

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

Q

Maxterm :-

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

$$= m_1 \cdot m_3 \cdot m_6$$

$$Y = \pi m(1, 3, 6)$$

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

Q

For a given truth
Table, write the
Standard sum of
Product form.

$$Y = \sum m(2, 3, 6)$$

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

$$= m_2 + m_3 + m_6$$

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

A B C Y

$$0 \quad 0 \quad 0 \quad 0$$

$$0 \quad 0 \quad 1 \quad 0$$

$$0 \quad 1 \quad 0 \quad 1$$

$$0 \quad 1 \quad 1 \quad 1$$

$$1 \quad 0 \quad 0 \quad 0$$

$$1 \quad 0 \quad 1 \quad 0$$

$$1 \quad 1 \quad 0 \quad 1$$

$$1 \quad 1 \quad 1 \quad 0$$

$$Y = \pi m(0, 1, 4, 5, 7)$$

$$= m_0 \cdot m_1 \cdot m_4 \cdot m_5 \cdot m_7$$

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

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

Simplify the following three variable expression using boolean algebra.

Q $Y = \Sigma m(1, 3, 5, 7)$

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

Q Simplify the following,

$$Y = \Pi M(3, 5, 7)$$

$$\begin{aligned} Y &= (A+\bar{B}+\bar{C}) \cdot (\bar{A}+B+\bar{C}) \cdot (\bar{A}+\bar{B}+\bar{C}) \\ &= (A\bar{A}^o + AB + A\bar{C} + \bar{A}\bar{B} + \bar{B}\bar{B}^o + \bar{B}\bar{C} + \bar{A}\bar{C} + \bar{B}\bar{C} + \bar{C}\bar{C}^o) \\ &\quad (\bar{A}+\bar{B}+\bar{C}) \\ &= (AB + A\bar{C} + \bar{A}\bar{B} + \bar{B}\bar{C} + \bar{A}\bar{C} + \bar{B}\bar{C} + \bar{C}\bar{C}^o)(\bar{A}+\bar{B}+\bar{C}) \\ &= A\bar{A}\bar{B}^o + A\bar{B}\bar{B} + A\bar{B}\bar{C} + \bar{A}\bar{C}\bar{A} + A\bar{C}\bar{B} + A\bar{C}\bar{C}^o + \bar{A}\bar{B}\bar{A} + \bar{A}\bar{B}\bar{B} \\ &\quad + \bar{A}\bar{B}\bar{C} + \bar{B}\bar{C}\bar{A} + \bar{B}\bar{C}\bar{B} + \bar{B}\bar{C}\bar{C}^o + \bar{A}\bar{C}\bar{A} + \bar{A}\bar{C}\bar{B} + \\ &\quad \bar{A}\bar{C}\bar{C}^o + \bar{B}\bar{C}\bar{A} + \bar{B}\bar{C}\bar{B}^o + \bar{B}\bar{C}\bar{C} + \bar{C}\bar{A}\bar{A} + \bar{C}\bar{B}\bar{B} + \bar{C}\bar{C}\bar{C}^o \\ &= AB\bar{C} + A\bar{B}\bar{C} + \underline{A\bar{C}} + \bar{A}\bar{B} + \bar{A}\bar{B}\bar{C} + \bar{B}\bar{C} + \underline{\bar{A}\bar{C}} + \\ &\quad \bar{A}\bar{B}\bar{C} + \bar{B}\bar{C} + \bar{C} \\ &= A\bar{C}(\bar{B}+\bar{B}) + \bar{C}(A+\bar{A}) + \bar{A}\bar{B}(1+\bar{C}) + \bar{B}\bar{C} + \bar{B}\bar{C}(A+1)\bar{A}\bar{C} \\ &= A\bar{C} + \bar{C} + \bar{A}\bar{B} + \bar{C}(B+1) + \bar{B}\bar{C} \\ &= \underline{\bar{A}\bar{C}} + \underline{\bar{C}} + \bar{A}\bar{B} + \bar{C} + \bar{B}\bar{C} = \bar{C}(A+1+B) + \bar{A}\bar{B} \\ &= \bar{C} \cdot (1+B) + \bar{A}\bar{B} = \bar{C} + \bar{A}\bar{B}, \end{aligned}$$

:- K-Map

Graphical \rightarrow LSI or LSI K Method w/ Karnaugh map \rightarrow chart
 - \rightarrow LSI LSI \rightarrow (K-map)

Cells \rightarrow \rightarrow LSI boxes \rightarrow map w/
 - \rightarrow LSI LSI

\rightarrow cells into 2^n cells \rightarrow LSI
 n w. \rightarrow LSI \rightarrow LSI \rightarrow LSI
 - \rightarrow LSI LSI \rightarrow LSI

$2^2 = 4$ cells \rightarrow map w/ 2-variable \rightarrow w/

$2^3 = 8$ cells \rightarrow 3-variable map \rightarrow LSI
 . same. \rightarrow LSI

4 variable

2-variable map

3-variable

Product \rightarrow cells \rightarrow K-map

w/ assigned \rightarrow LSI \rightarrow terms
 - \rightarrow LSI \rightarrow LSI

2-variable K-map 3-variable 4-variable K-map

B		A				C				D			
		0	1	00	01	11	10	00	01	11	10		
0	0	$\bar{A}\bar{B}$	$\bar{A}B$	0	$\bar{A}\bar{B}\bar{C}$	$\bar{A}\bar{B}C$	$\bar{A}BC$	0	$\bar{A}\bar{B}\bar{C}\bar{D}$	$\bar{A}\bar{B}\bar{C}D$	$\bar{A}\bar{B}CD$	0	$\bar{A}\bar{B}C\bar{D}$
	1	$A\bar{B}$	AB		$A\bar{B}\bar{C}$	$A\bar{B}C$	ABC		$A\bar{B}\bar{C}\bar{D}$	$A\bar{B}\bar{C}D$	$\bar{A}BCD$		$A\bar{B}C\bar{D}$
1	0	$\bar{A}\bar{B}$	$\bar{A}B$	1	$\bar{A}\bar{B}\bar{C}$	$\bar{A}\bar{B}C$	$\bar{A}BC$	1	$\bar{A}\bar{B}\bar{C}\bar{D}$	$\bar{A}\bar{B}\bar{C}D$	$\bar{A}BCD$	1	$\bar{A}\bar{B}C\bar{D}$
	1	$A\bar{B}$	AB		$A\bar{B}\bar{C}$	$A\bar{B}C$	ABC		$A\bar{B}\bar{C}\bar{D}$	$A\bar{B}\bar{C}D$	$ABC\bar{D}$		$A\bar{B}C\bar{D}$

Representation of truth table in K-map

A	B	y
0	0	0
0	1	1
1	0	1
1	1	0

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

Representation of 2-variable T.T on K-map.

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

A\BC	00	01	11	10
0	0 ₀	0 ₁	0 ₂	1 ₃
1	1 ₄	1 ₅	1 ₇	1 ₆

OR

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

Representation of 3-variable T.T on k-map.

A	B	C	D	Y	
0	0	0	0	1	AB\CD
0	0	0	1	0	00 10 01 11 10
0	0	1	0	0	01 14 15 17 06
0	0	1	1	1	11 12 013 115 14
0	1	0	0	1	10 08 09 011 110
0	1	0	1	1	
0	1	1	0	0	
0	1	1	1	1	R, $\bar{C}D$, $\bar{C}\bar{D}$, CD , $\bar{C}\bar{P}$
1	0	0	0	0	$\bar{A}\bar{B}$ 10 01 13 02
1	0	0	1	0	$\bar{A}B$ 14 15 17 06
1	0	1	0	1	AB 12 013 115 14
1	0	1	1	0	$A\bar{B}$ 08 09 011 110
1	1	0	0	1	
1	1	0	1	0	
1	1	1	0	1	
1	1	1	1	1	

Representation of 4-variable. TT on K-map

Representing standard SOP on K-map

Ex:- $Y = ABC\bar{C} + ABC + \bar{A}\bar{B}C$, Plot boolean

Expression on K-map.

A\BC	00	01	11	10		$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
0	0 ₀	1 ₁	0 ₃	0 ₂	\bar{A}	0 ₀	1 ₁	0 ₃	0 ₂
1	0 ₄	0 ₅	1 ₇	1 ₆	A	0 ₄	0 ₅	1 ₇	1 ₆

Ex: $Y = \bar{A}\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D} + \bar{A}BC\bar{D} + A\bar{B}CD + AB\bar{C}D.$

Plot Boolean expression on K-map.

		AB				CD						
		00	01	11	10	00	01	11	10			
		00	0 ₀	0 ₁	0 ₃	0 ₂	00	0 ₀	0 ₁	0 ₃	0 ₂	
		01	1 ₄	0 ₅	0 ₇	1 ₆	00	1 ₄	0 ₈	0 ₇	1 ₆	
		11	0 ₁₂	1 ₁₃	0 ₁₅	0 ₁₄	00	AB	0 ₁₂	1 ₁₃	0 ₁₅	0 ₁₄
		10	0 ₉	0 ₃	1 ₁₁	1 ₁₀	00	A <bar>B</bar>	0 ₈	0 ₉	1 ₁₁	1 ₁₀

Representations standard POS on K-map.

Ex: Plot Boolean expression on K-map.

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

		AB				BC						
		00	01	11	10	00	01	11	10			
		00	1 ₀	0 ₁	0 ₃	0 ₂	00	1 ₀	0 ₁	0 ₃	0 ₂	
		1	1 ₄	1 ₈	1 ₇	0 ₆	00	A	1 ₄	1 ₅	1 ₂	0 ₆

(OR)

		B+C				B+ <bar>C</bar>					
		A	1 ₀	0 ₁	0 ₃	0 ₂	A	1 ₀	1 ₁	1 ₃	1 ₂
		A	1 ₄	1 ₈	1 ₂	0 ₆	A	1 ₄	1 ₅	1 ₇	1 ₆
		A	1 ₄	1 ₈	1 ₂	0 ₆	A	1 ₄	1 ₅	1 ₇	1 ₆

Ex:- Plot Boolean Expression on K-map

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

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

$$A+B+C+\bar{D} = m_1$$

$$A+\bar{B}+\bar{C}+D = m_7$$

$$A+B+\bar{C}+\bar{D} = m_3$$

$$\bar{A}+\bar{B}+C+\bar{D} = m_{13}$$

$$A\bar{B}+\bar{C}+D = m_{14}$$

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

Grouping of cells for simplification :-

Grouping of two adjacent ones (Pairs) :- double

$A \backslash B$	00	01	11	10	$A \backslash BC$	00	01	11	10
0	0 ₀	1 ₁	1 ₃	0 ₂	0	0 ₀	0 ₁	1 ₃	0 ₂
1	0 ₄	0 ₅	0 ₈	0 ₆	1	0 ₄	0 ₅	1 ₈	0 ₆

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

$$Y = B \cdot C$$

$A \backslash B$	0	1
0	1 ₀	0 ₁
1	1 ₈	0 ₂

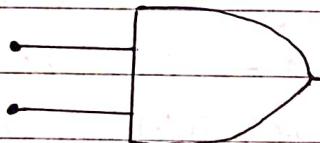
$A \backslash B$	0	1
0	0 ₀	1 ₁
1	0 ₂	1 ₃

Logic Gates

~~Basic Gates (Universal Gates)~~, Ex - OR,
Ex - NOR).

Basic Gates (AND, OR, NOT).

-: AND GATE (1)



logic symbol

$$Y = A \cdot B$$

Logic Expression

logical multiplication \wedge And Gate.

gate \wedge or \wedge لیس لیس refers \wedge or

لیس \wedge لیس high لیس output لیس

لیس \wedge لیس high input لیس

لیس \wedge لیس low input لیس

Gate \wedge لیس لیس low output لیس

لیس \wedge لیس لیس or لیس لیس

لیس \wedge لیس Certain Condition لیس

value لیس output لیس true لیس

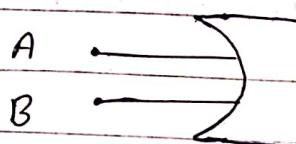
لیس \wedge Gate \wedge لیس high

output لیس \wedge لیس input لیس لیس

لیس \wedge لیس

INPUT		OUTPUT
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Truth table



-: OR - GATE #

$$Y = A + B$$

Logic Symbol.

Logic Expression

in logical addition or OR-Gate

operation of OR-Gate. \Leftrightarrow if we refer

high output will get if two or

if Input goes high or low \Leftrightarrow if

\Leftrightarrow low high value

Input will go to OR Gate or

OR figure \Leftrightarrow true output \Leftrightarrow 1

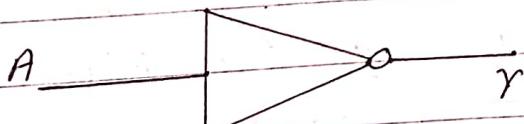
truth table. \Leftrightarrow logical symbol of OR-Gate

\Leftrightarrow 1's 0's or

OR-Gate		
INPUT	OUTPUT	
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Truth table

-: NOT - GATE #



Logic symbol

$$Y = \bar{A}$$

Logic expression

INPUT	OUTPUT
A	Y
0	1
1	0

'Complementation' or 'inversion' or NOT-Gate

ایک سادہ ناٹ یعنی بائی بائی میں اسے میں اسے
نیچے کی طرف output کی میں input
میں اسی truth table میں symbols of logic
کے لیے لکھا گا figure

Explain the Universal Gate

These are two types of Universal Gate.

(1) NAND GATE

(2) NOR GATE

-: Universal Gate

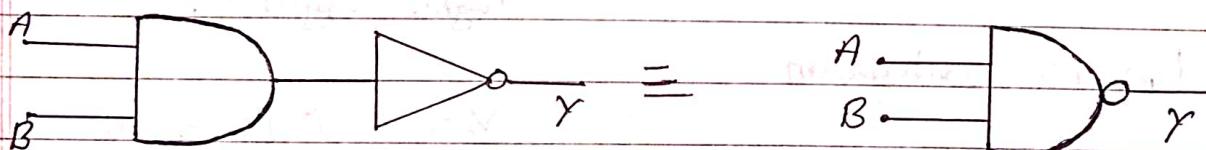
'NAND' w/ logical expression $\bar{A} \cdot \bar{B}$ can

کے طبقہ میں NOR-GATE میں

Universal gate can also be called

as tile like or

-: NAND GATE



logic symbol

INPUT	OUTPUT
A	0
B	0
0	1
1	1
0	1
1	0
1	1

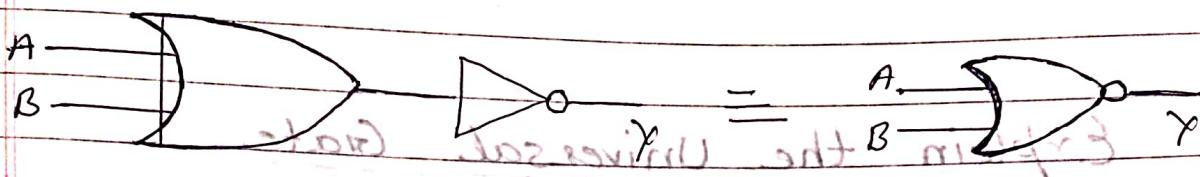
$$Y = \bar{A} \cdot \bar{B}$$

logic expression

Truth table

لیا گے اے operation NOT - AND یعنی
truth یا logic symbols کی مکمل
کے لیے لیا گے figure وہ table

NOR - GATE #²



INPUT		OUTPUT	operation NOT - OR یعنی
A	B	Y	- کے لیے لیا گے اے logical symbol کی figure وہ truth table
0	0	1	- کے لیے لیا گے اے logical symbol کی figure وہ truth table
0	1	0	- کے لیے لیا گے اے
1	0	0	- کے لیے لیا گے اے
1	1	0	- کے لیے لیا گے اے

\therefore Exclusive - OR #

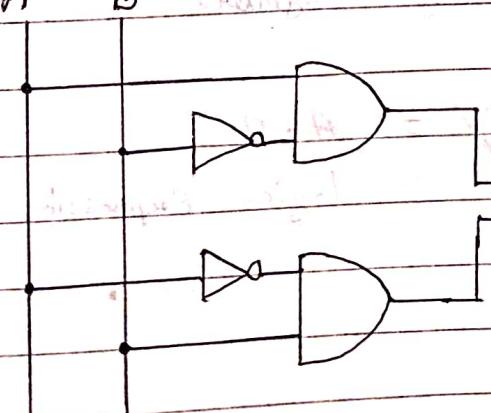


logic symbol

Logic Diagram

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

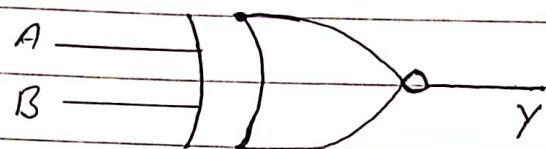
logic Expression



Exclusive NOR Operation \Rightarrow

Fig. shows the standard symbol (of) Exclusive - NOR gate. Its truth Table is given.

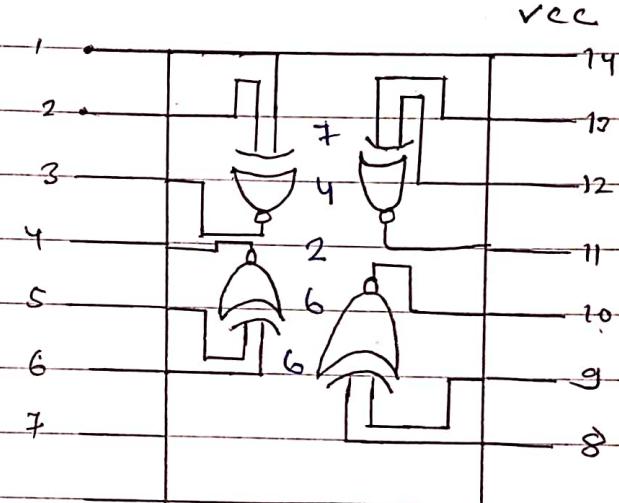
Its logic operation specified as,



$$Y = A \text{ Ex-NOR } B \quad B = A \text{ Ex-OR } B \quad A \oplus B = A \odot B$$

V_{CC} :- voltage Common collector.

* V_{CC} is the power input of a device. It may be positive or negative with respect to GND (ground).



Similar to Ex-OR gate, Ex-NOR Gate is not a basic operation and can be performed using the basic Gates or Universe gates, but because of its importance it has been given a standard symbol.

The Ex-NOR operation is also known as the coincidence operation because it produces output of 1 when its inputs coincides.

INPUTS		OUTPUT
A	B	γ
0	0	1
0	1	0
1	0	0
1	1	1

Grouping of cells for simplification :-

Grouping of two adjacent ones (Pairs) :-

A\BC	00	01	11	10
0	0, 0,	1, 1,	0, 0,	0, 0,
1	0, 1,	0, 0,	0, 1,	0, 0,

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

A\BC	00	01	11	10
0	0, 0,	0, 0,	0, 1,	0, 0,
1	0, 1,	0, 0,	1, 1,	0, 0,

$$Y = B \cdot C$$

A\BC	00	01	11	10
0	0, 0,	0, 1,	0, 0,	0, 0,
1	1, 1,	0, 0,	0, 1,	0, 0,

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

A\BC	00	01	11	10
0	0, 1,	0, 0,	0, 0,	0, 1,
1	0, 0,	0, 1,	0, 0,	0, 0,

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

A\BC	00	01	11	10
0	0, 0,	1, 1,	1, 1,	0, 0,
1	0, 1,	0, 0,	1, 1,	0, 0,

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

A\BC	00	01	11	10
0	0, 0,	1, 1,	1, 1,	0, 0,
1	0, 1,	0, 0,	1, 1,	0, 0,

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

A\B	0	1
0	1, 0,	0, 1,
1	1, 1,	0, 0,

(1)

A\B	0	1
0	0, 0,	1, 1,
1	0, 1,	1, 1,

(2)

A\B	0	1
0	1, 1,	1, 1,
1	0, 0,	0, 0,

(3)

A\B	0	1
0	0, 0,	0, 1,
1	1, 1,	1, 1,

(4)

A\B	0	1
0	1, 1,	0, 0,
1	0, 0,	0, 0,

(5)

A\B	0	1
0	0, 0,	0, 1,
1	0, 1,	1, 1,

(6)

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

$$Y = A \cdot B$$

Grouping of four adjacent ones (quad) :-

$$Y = A \quad Y = \bar{C} \quad Y = C$$

$\bar{A}B$	$C\bar{D}$	00	01	11	10	$\bar{A}B$	$C\bar{D}$	00	01	X11	10
00	0	0	0	1	0	00	0	0	0	0	0
01	0	4	0	1	0	01	0	4	1	1	0
11	0	12	0	1	0	11	0	12	1	1	0
10	0	8	0	1	0	10	0	8	0	0	0

<u>AB</u>	<u>CD</u>	00	01	11	10	<u>AB</u>	<u>CD</u>	00	01	11	10
00	00	0.	0,	0 ₃	0 ₂	00	1.	0,	0,	0 ₃	0 ₂
01	04	0 ₄	0 ₅	0 ₇	0 ₆	01	0 ₄	0 ₅	0 ₇	0 ₆	
11	1 ₂	0 ₁₂	0 ₁₃	0 ₁₅	1 ₁₄	11	0 ₁₂	0 ₁₃	0 ₁₅	0 ₁₄	
10	1 ₃	0 ₃	0 ₉	0 ₁₁	1 ₁₀	10	1 ₃	0 ₉	0 ₁₁	1 ₁₀	

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

\overline{AB}	\overline{CD}	00	01	11	10	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31						
00	00	00	01	11	10	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0G	0H	0I	0J	0K	0L	0M	0N	0O	0P	0Q	0R	0S	0T	0U	0V	0W	0X	0Y	0Z				
01	01	04	05	07	06	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	1G	1H	1I	1J	1K	1L	1M	1N	1O	1P	1Q	1R	1S	1T	1U	1V	1W	1X	1Y	1Z		
11	11	012	013	015	014	018	019	01A	01B	01C	01D	01E	01F	01G	01H	01I	01J	01K	01L	01M	01N	01O	01P	01Q	01R	01S	01T	01U	01V	01W	01X	01Y	01Z	021	022	023	024	025	026	027	028	029	020
10	10	08	19	11	010	011	012	013	014	015	016	017	018	019	01A	01B	01C	01D	01E	01F	01G	01H	01I	01J	01K	01L	01M	01N	01O	01P	01Q	01R	01S	01T	01U	01V	01W	01X	01Y	01Z			

Grouping of eight adjacent ones (octet) :-

	$\bar{B}C$	00	01	11	10
0	1 ₀	1 ₁	1 ₃	1 ₂	
1	1 ₄	1 ₅	1 ₇	1 ₆	

$y = 1$

	$\bar{A}\bar{B}CD$	00	01	11	10
00	0 ₀	0 ₁	0 ₃	0 ₂	
01	1 ₄	1 ₅	1 ₇	1 ₆	
11	1 ₂	1 ₃	1 ₅	1 ₄	
10	0 ₈	0 ₉	0 ₁₁	0 ₁₀	

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

$y = D$

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

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

$$y = \overline{D}$$

Solve the Problem by K-map:-

$$Q8 \quad y = A\bar{B}C + \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C}$$

	$\bar{B}C$	00	01	11	10
0	1 ₀	1 ₁	1 ₃	0 ₂	
1	1 ₄	1 ₅	0 ₇	0 ₆	

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

Q) $y = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + A\bar{B}\bar{C}D + A\bar{B}\bar{C}D + \bar{A}\bar{B}CD.$

\bar{A}	\bar{B}	\bar{C}	\bar{D}	00	01	11	10
00	0	0	0	0	0	0	1
01	1	0	0	1	1	0	0
11	1	1	0	1	1	0	0
10	0	1	0	0	1	0	0

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

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

01	0	10	12
04	05	07	06
12	03	05	14
18	09	11	10
01	08	09	00

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

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

00	1	0	02	1st
04	15	17	16	2nd
12	13	15	014	3rd
08	09	11	010	3rd

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

implement Logic Function w/ truth table \Leftrightarrow

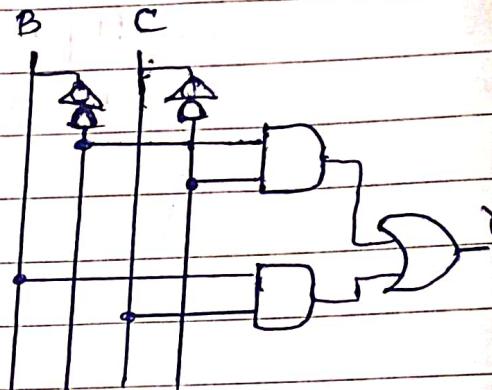
جلى و اىب اىب و جلى \Leftrightarrow اىصال K-map

variable اىب A, B, C O/P variable
- $\frac{A}{B}$ O/P

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

1 ₀	0 ₁	1 ₃	0 ₂
1 ₄	0 ₅	1 ₇	0 ₆

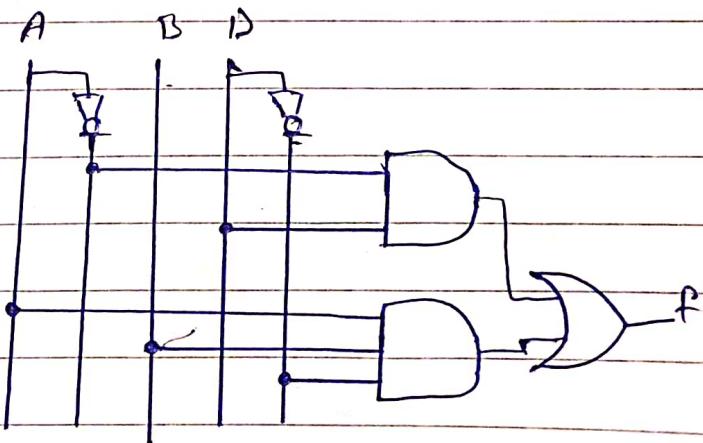
$$Y = \bar{B}\bar{C} + BC$$



reduce \Leftrightarrow k-map \Leftrightarrow Function \Leftrightarrow Q
کریں Implement \Leftrightarrow basic gates \Leftrightarrow C.I.

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

	00	01	11	10
00	0 ₀	1 ₁	1 ₃	0 ₂
01	0 ₄	1 ₅	1 ₇	0 ₆
11	1 ₁₀	0 ₁₃	0 ₁₅	1 ₁₄
10	0 ₀	0 ₀	0 ₀	0 ₀



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

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

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

Don't Care condition:-

جیٰ condition کیا ہے میں Logic condi. کیا ہے
 میں Case ۱ - ۲ واقع ہوتے ہیں کیا ہے
 کیا ہے میں Define کیا ہے O/P Level
 Low یا High کیا ہے O/P ~ ۱۰۱ کیا ہے
 O/P Level کیا ہے ۱ - ۰ کیا ہے میں
 ۱۰۱ - ۰ کیا ہے جاتا ہے میں indicate کیا ہے 'x'
 کیا ہے جاتا ہے O/P 'don't care' کیا ہے

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	x
1	1	1	x

$\sum m(0, 2, 4)$ $\sum m(1, 3, 7, 11, 15) \oplus d(0, 2, 4)$

$$q. f(A, B, C, D) = \sum m(1, 3, 7, 11, 15) \oplus d(0, 2, 4)$$

AB	CD	00	01	11	10	
00	X ₀	1 ₁	1 ₃	X ₂		
01	X ₄	0 ₅	1	0		
11	0	0	1	0		
10	0	0	1	0		

$\sum m(0, 2, 4)$ $\sum m(1, 3, 7, 11, 15) \oplus d(0, 2, 4)$

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

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

$$f(A, B, C, D) = \sum m(5, 6, 7, 12, 13) + d(4, 9, 14, 15)$$

Q

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

=

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

$y = B$

Q $f(A, B, C) = \sum m(0, 1, 3, 7) + d(2, 5)$

A

	BC	00	01	11	10
	00	1 ₀	1 ₁	1 ₃	x ₂
0	0	1 ₀	x ₅	1 ₇	0 ₆
1	1	0 ₄	x ₅	1 ₇	0 ₆

=

	BC	00	01	11	10
	00	1 ₀	1 ₁	1 ₃	1 ₂
0	0	1 ₀	1 ₅	1 ₇	0 ₆
1	1	0 ₄	x ₅	1 ₇	0 ₆

$y = -AFC$

Q $F = (W, X, Y, Z) = \sum m(0, 7, 8, 9, 10, 12) + d(2, 5, 13)$

WX

	ZY	00	01	11	10
	00	1 ₀	0 ₁	0 ₃	x ₂
0	0	0 ₀	x ₅	1 ₇	0 ₆
1	1	1 ₄	x ₃	0 ₅	0 ₁₁
2	1	1 ₉	1 ₂	0 ₁₀	1 ₆
3	0	1 ₈	1 ₉	0 ₁₁	1 ₀

=

	ZY	00	01	11	10
	00	1 ₀	0 ₁	0 ₃	1 ₂
0	0	1 ₀	1 ₅	1 ₇	0 ₆
1	1	1 ₂	1 ₃	0 ₁₅	0 ₁₄
2	1	1 ₈	1 ₉	0 ₁₁	1 ₁₀

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

Q

Q -

A	B	C	D	γ
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	x
1	0	1	1	x
1	0	0	0	x
1	0	0	1	x
1	0	1	0	x
1	0	1	1	x

AB		CD		00	01	11	10
00	00	00	00	0	1	0	1
01	01	04	15	0	1	0	1
11	11	02	X ₁₃	1	0	1	0
10	10	15	12	X ₁₂	X ₁₃	X ₁₅	X ₁₄

AB		CD		00	01	11	10
00	00	00	00	0	1	0	1
01	01	04	15	0	1	0	1
11	11	02	X ₁₃	1	0	1	0
10	10	08	19	X ₈	X ₉	X ₇	X ₆

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

Q $f(A, B, C, D) = \pi m(4, 5, 6, 7, 12) + d(11, 13)$

AB		CD		00	01	11	10
00	00	00	00	0	1	1	x ₂
01	01	04	15	0	1	0	0 ₂
11	11	02	X ₁₃	1	0	1	0 ₁₄
10	10	15	12	X ₁₂	X ₁₃	X ₁₅	X ₁₄

AB		CD		00	01	11	10
00	00	00	00	0	1	1	0 ₂
01	01	04	15	0	1	0	0 ₆
11	11	02	X ₁₃	1	0	1	0
10	10	18	19	X ₈	X ₉	X ₁₁	0 ₁₀

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

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

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

	CD ↓	00	01	11	10
AB ↑	00	0 ₀	1 ₁	1 ₃	X ₂
	01	0 ₄	1 ₅	0 ₇	0 ₆
	11	0 ₁₂	X ₃	1 ₅	0 ₁₄
	10	1 ₈	1 ₉	1 ₁₁	0 ₁₀

	CD ↓	00	01	11	10
AB ↑	00	0 ₀	1 ₁	1 ₃	0 ₂
	01	0 ₄	1 ₅	0 ₇	0 ₆
	11	0	1	1	0
	10	1 ₈	1 ₉	1 ₁₁	0 ₁₀

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

Q- $F = \bar{A}\bar{B}\bar{C} + \bar{B}C\bar{D} + ABCD + A\bar{B}\bar{C}$

Q2. Combinational logic circuits

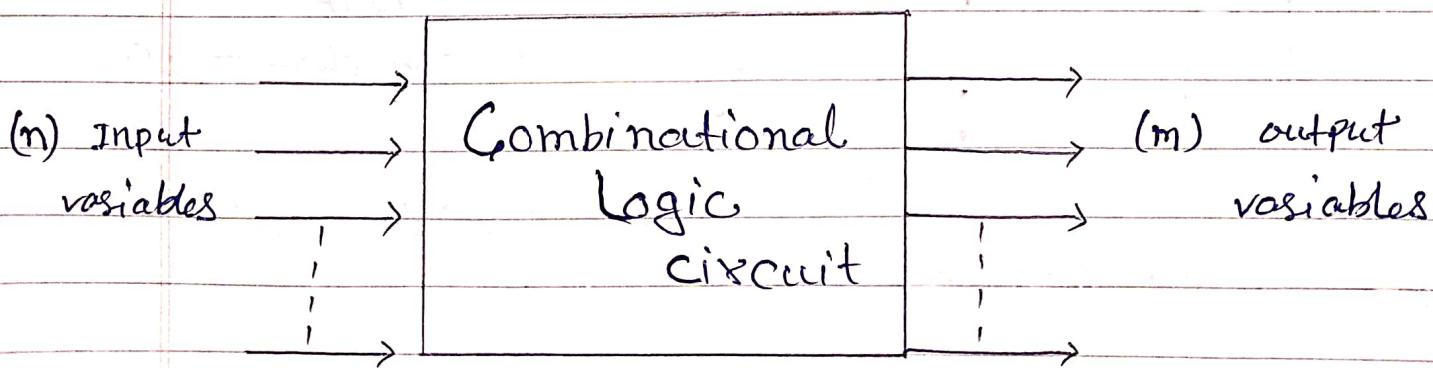
Arithmetic Circuits

-
∴ Combinational Circuits

- Combinational Circuits are made up of logic gates
- All circuit has output logic based on combination of input logic and storage circuit. Output is not stored.
- They involve logic

Input variables go to circuit
Input Combination -
-
-
Input goes to Combinational Circuits
- They output variables as variables

- Input Signal to logic gates
- They accept input variables
- They generate output signals



Block Diagram of a Combinational Circuit

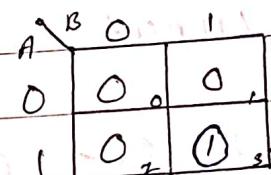
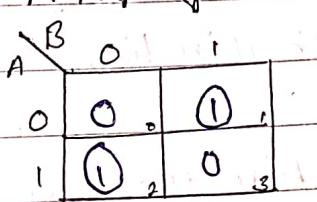
\therefore Half adder

addition of one bit binary numbers \Rightarrow
 It shows 6 logic circuit all of
the 6 logic are Half adders \therefore $S = A \oplus B$
 & "Carry" $C = AB$

Logic diagram Truth table Figure
 It shows K-map & logic symbol

INPUT		OUTPUT	
A	B	S	C
0	0	0	0
0	1	1	0
1	1	1	0
1	1	0	1

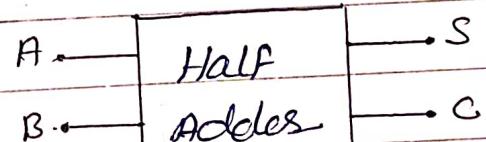
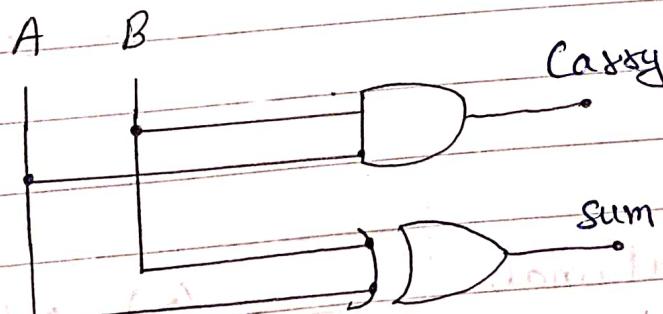
K map of sum (S)



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

$$C = AB$$

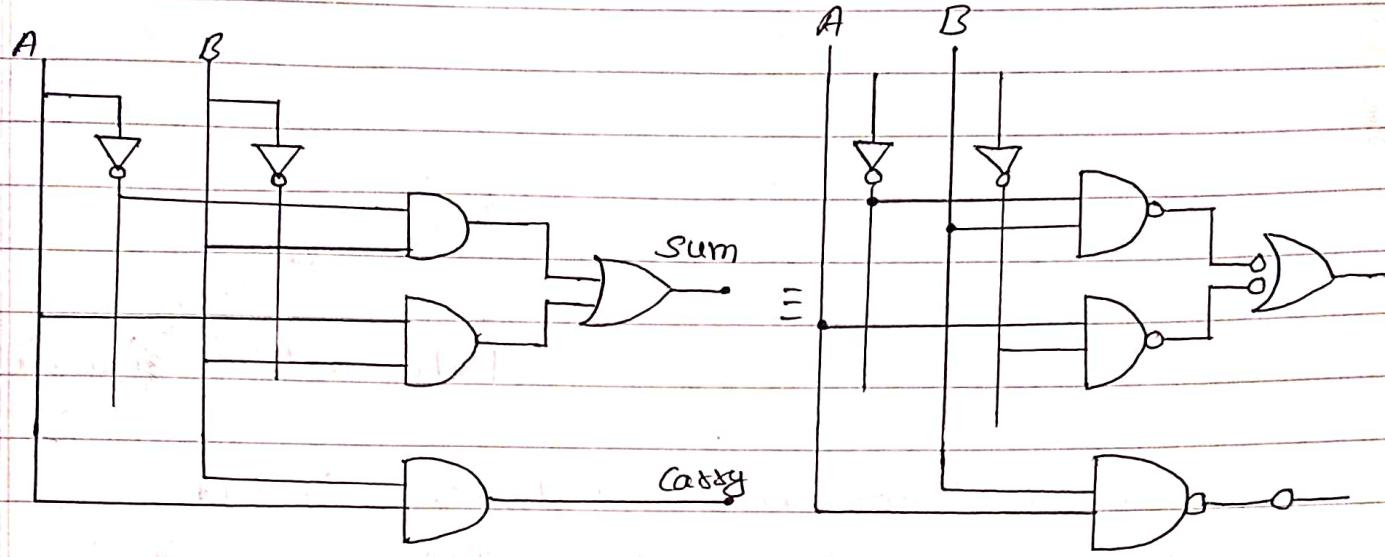
Truth Table



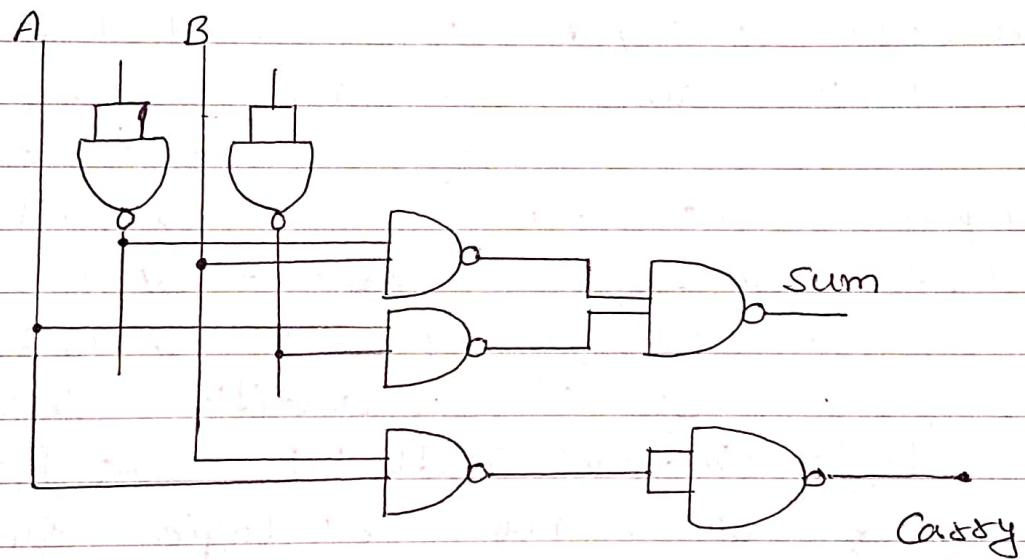
Logic symbol

Logic Diagram

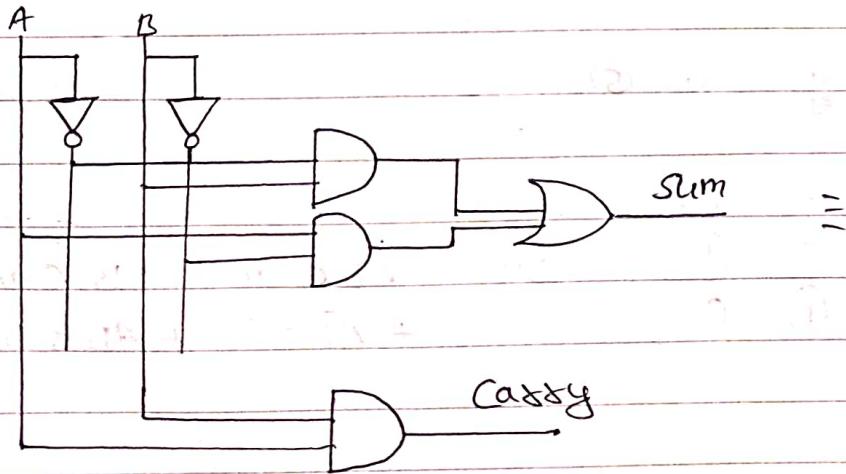
Half adder using NAND gates only :-



III



Half adder using NOR gates only :-



-: Full adder circuit

کی اے three bits of logic circuits hیں
Full adder circuit کی adder hیں جو
- کی bits کے adder
accept of Inputs جو circuit کے بتاتے
output carry وی sum وی کی
- کی generates اے
Inputs کے Input جو Circuit کے
THIRD کی کی represent اے B کے A
کی کے - کی کی represent اے 'C' in Input
- کی کی an addition پہاڑیں
'logic diagram' Truth table کے Figure
لے کے k-map وی logic symbol
- کی کی

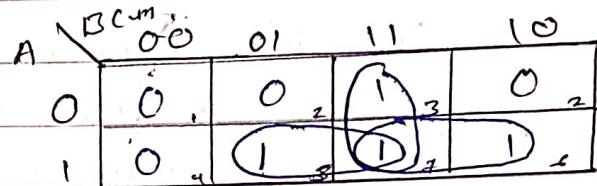
K map of sum (S)

A\BC _{in}	00	01	11	10
0	0 ₀	0 ₁	0 ₃	0 ₂
1	0 ₄	0 ₅	0 ₇	0 ₆

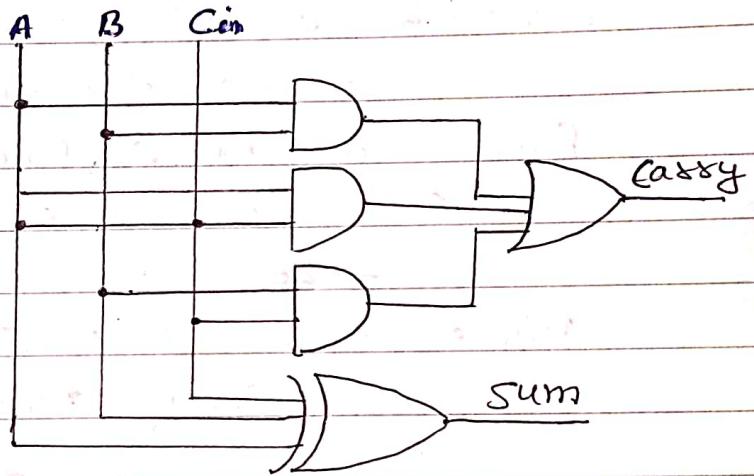
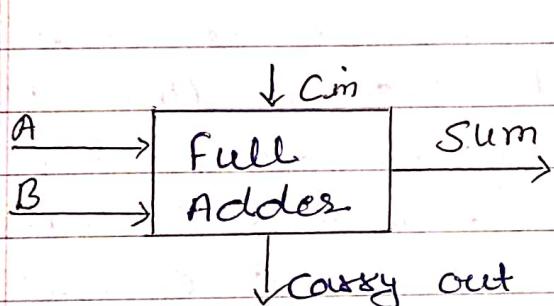
$$\text{Sum} = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + AB{C}_{in}$$

INPUT	OUTPUT			
A	B	Cin	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

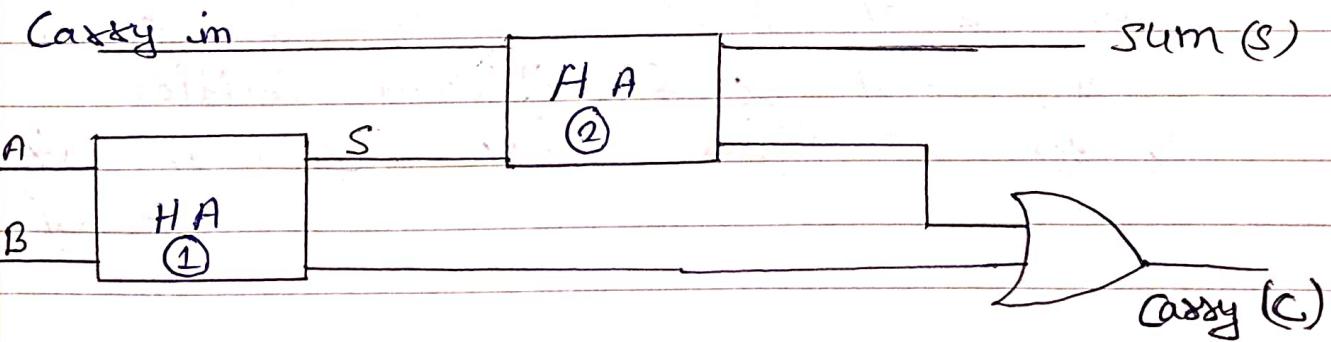
K-map of carry (C)



$$\text{Carry} = AB + AC_{in} + BC_{in}$$



Construction of full adder from two half adders and an OR gate.

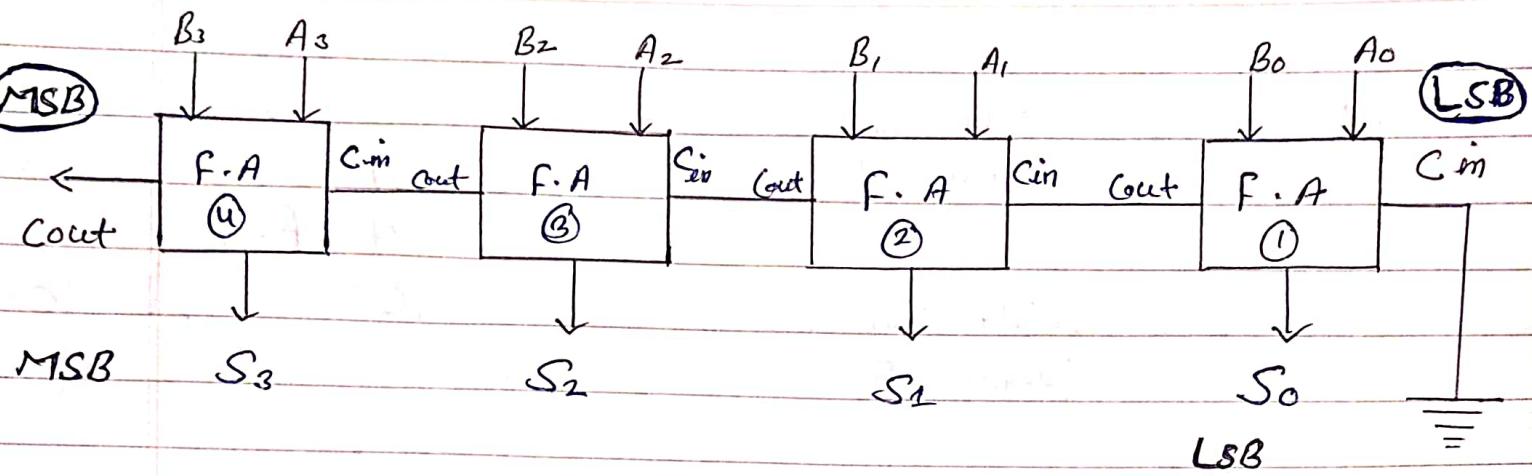


Draw and explain a 4-bit parallel adder using full adder :-

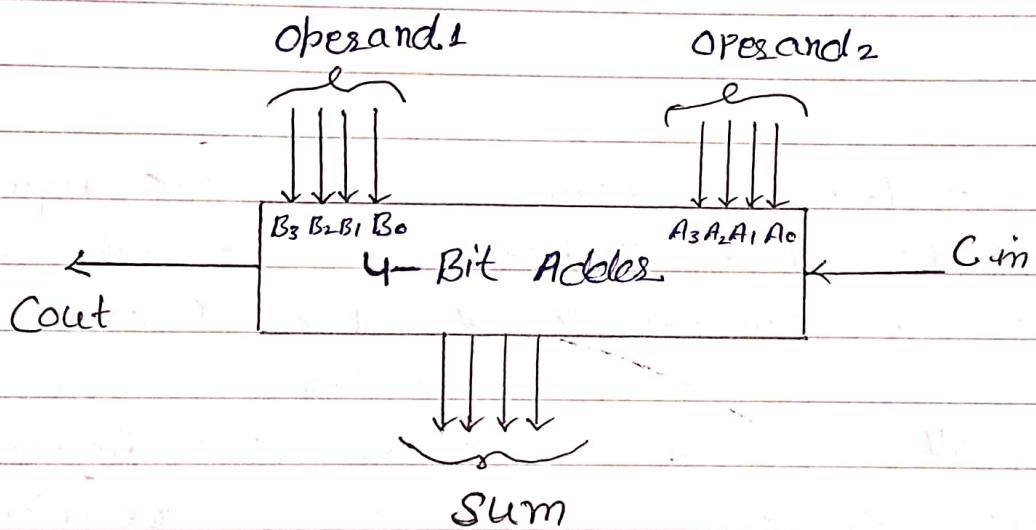
- > Full adder can add two one-bit numbers.
- > One-bit numbers are less capable.
- > At first add at input carry of binary numbers. At 4th bit add full adder extra at first add.
- > 4-bit parallel adder

Construct a 4-bit parallel adder using circuits of full adder. In the first step connect 4 parallel full adder to 4-bit parallel adder. Fig shows how to do it. In the second step connect carry output of adder. In carry input of next adder. -> Connected.

Half adder will be taken note of least significant position. It has one carry input of full adder. In the least significant position '0' is carry given its significant position. In figure we have -> carry given. -> Up



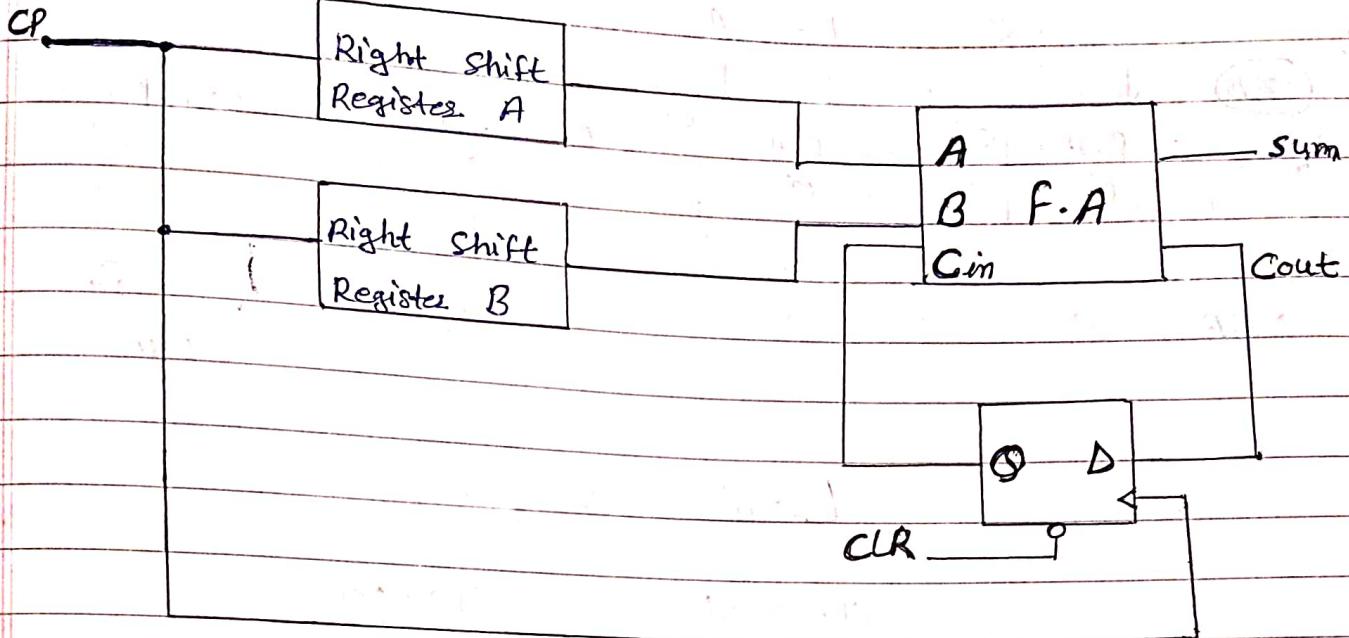
Logic Diagram



Logic symbol

-: Serial adder

logic at serial adder as figure
کے لیے دیکھا باید کے diagram



as logic diagram as in figure

A Right shift Register. as in the it's

add w numbers as in stored in B as

- serially as in just

bit by bit like full adder as figure

as in it will perform w addition

as in carry output as in w addition

w store as in d flip flop as

- as in

as carry input as in like carry gen-

- as in like as in next addition as

as in like cleared as D-flip flop as in

- as in like as in addition as in - as

least significant as in registers like

- as in Bits

right shift of data w/ clock pulse
right on 1-bit w/ one register
of 51 is w/ 5 bits w/ shift job at
previous w/ bits w/ next digit
New inputs w/ carry w/ addition
w/ inputs w/ full adder w/ at
w/ circuit

Ex:- (A)

1011 =

(B)

0101

Compare the series and parallel adder :-

Serial adder

1. Shift registers w/ serial adder 1.
- w/ two stages

2. Full adder w/ serial adder
- w/ first stage w/ adder circuit

3. Sequential circuit w/ serial adder

4. Time of w/ addition
depends w/ number of bits
- w/ time

- \curvearrowleft They slower \curvearrowright 5.

- \curvearrowleft They cheaper \curvearrowright 6.

Parallel adder

5. In \curvearrowleft registers \curvearrowright Parallel adder. 1.

6. In \curvearrowleft \curvearrowright parallel load capacity

- \curvearrowleft

7. Full adder \curvearrowright Parallel adder 2.

Bits \curvearrowleft Binary numbers 'Numbers'

- \curvearrowleft They equal \curvearrowleft numbers \curvearrowleft

8. Purely \curvearrowleft Parallel adder 3.

- \curvearrowleft Combinational circuit

9. - \curvearrowleft time \curvearrowleft in \curvearrowleft Addition 4.

It depends on number of bits

- \curvearrowleft They

- \curvearrowleft They faster \curvearrowright 5.

10. - \curvearrowleft They expensive \curvearrowright 6.

∴ Subtractors

Binary借位逻辑电路

- 从二进制数中减法
从二进制数中“二进制减法”

-: Half subtractor

从一个二进制数中减去另一个二进制数

逻辑电路从二进制数中减法

半减法器从二进制数中减去另一个二进制数

从“二进制位”中减去另一个二进制位

从一个二进制数中减去另一个二进制数

从“借位”中减去另一个二进制数

从“借位”中减去另一个二进制数

从“借位”中减去另一个二进制数

从“借位”中减去另一个二进制数

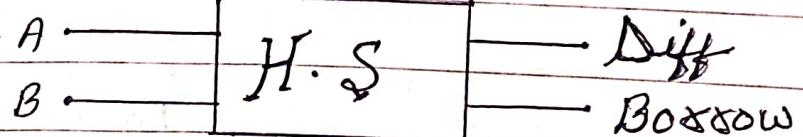
“逻辑图”“真值表”见图。

从K-map到“逻辑符号”

从K-map到逻辑符号

INPUT		OUTPUT	
A	B	Diff	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

真值表



逻辑符号

K-map of Diff

A \ B	0	1
0	0	1
1	1	0

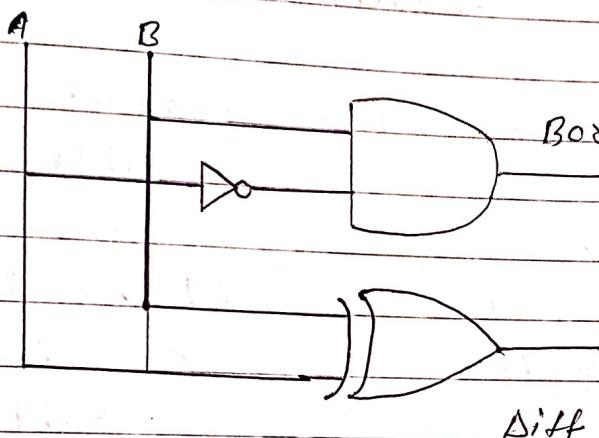
K-map of Borrow

A \ B	0	1
0	0	1
1	0	0

$$\text{Diff} = \bar{A}B + A\bar{B}$$

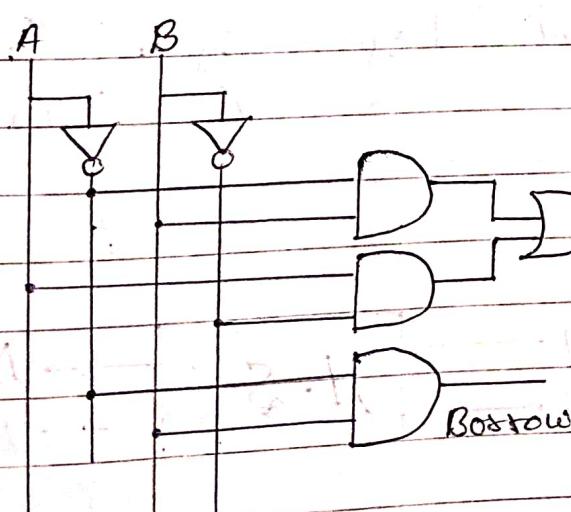
$$= A \oplus B$$

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

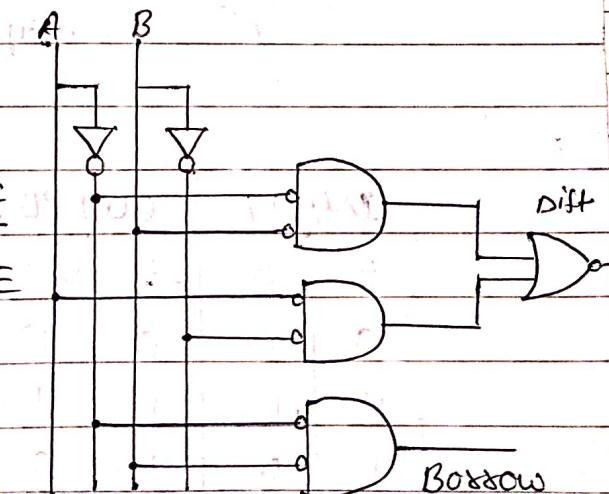


Logic diagram.

Half Subtractor using NOR Gate only.



Borrow signal



Borrow

:- Full-Subtracter

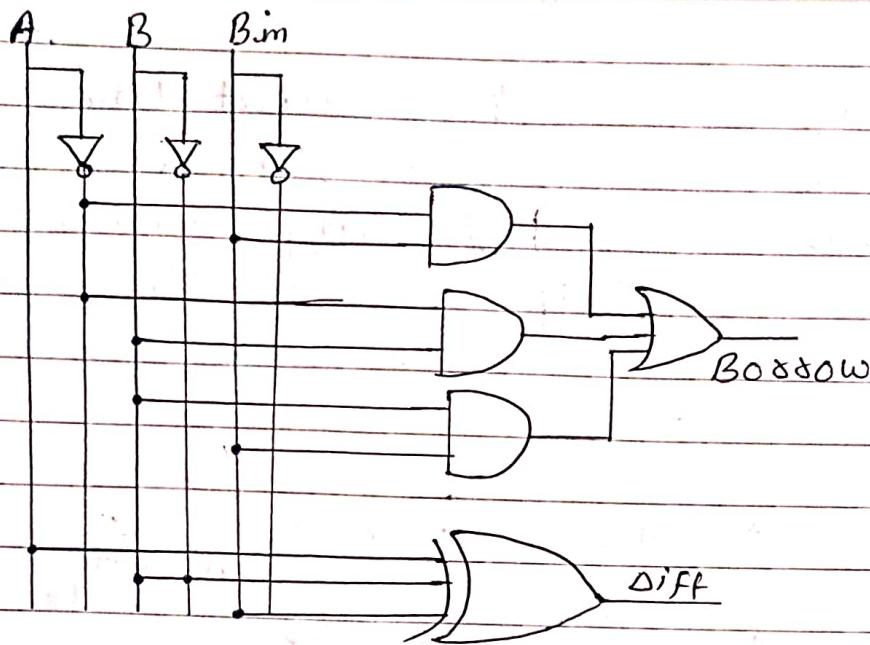
Three bits of logic circuits less
 to subtract we need C_i , C_{i-1} or
 like in full subtractor C_i and C_{i-1}
 Inputs to circuit is A and B & C_{i-1}
 so 'Diff' will accept it
 - it generates output 'Borrow'

so C_i Input to circuit is
 - it represent of B & A Inputs
 it represent of B in third input of
 subtraction - previous -
 - it is

Full Subtractor Half Subtraction
 like full adder Half adder
 example to combinational circuit

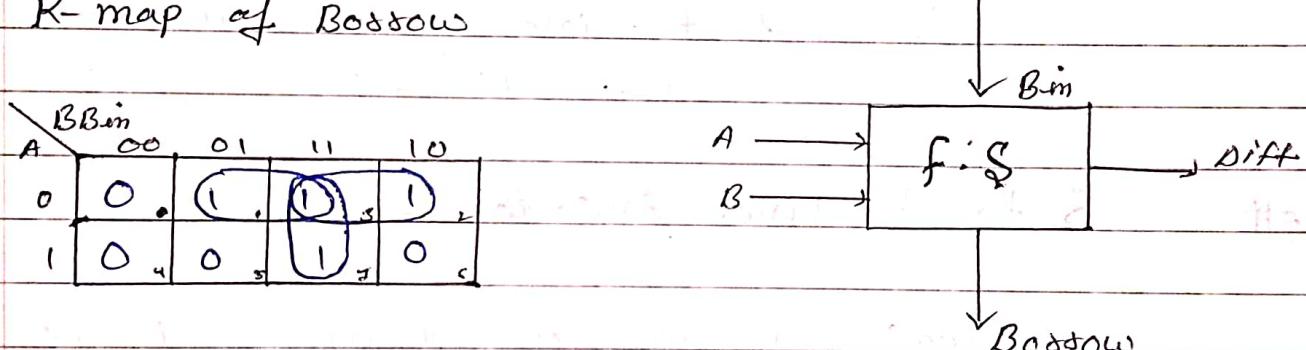
INPUT			OUTPUT		K-map of Diff					
A	B	B_{in}	Diff	Borrow	A	B	00	01	11	10
0	0	0	0	0	0	0	0	1	0	1
0	0	1	1	1	1	1	1	0	1	0
0	1	0	1	1						
0	1	1	0	1						
1	0	0	1	0						
1	0	1	0	0						
1	1	0	0	0						
1	1	1	1	1						

$\text{Diff} = \bar{A}\bar{B}B_{in} + \bar{A}B\bar{B}_{in} + A\bar{B}\bar{B}_{in} + AB{B}_{in}$
 $= B_{in}(\bar{A}\bar{B} + AB) + \bar{B}_{in}(\bar{A}B + A\bar{B})$
 $= B_{in}(A \oplus B) + \bar{B}_{in}(A \oplus B)$
 $= A \oplus B \oplus B_{in}$.



Logic circuit

K-map of Borrow



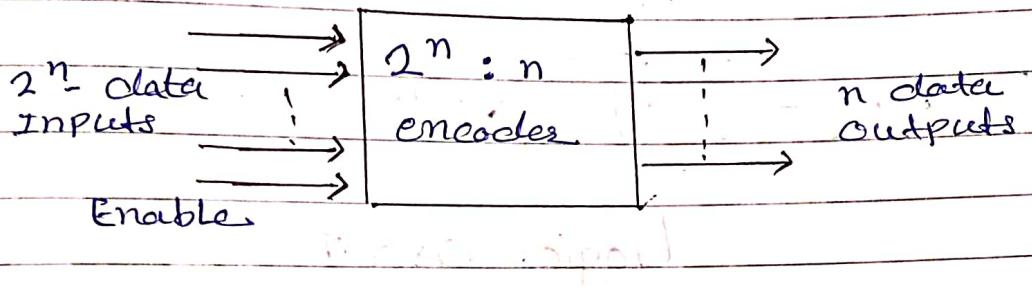
$$Borrow = \bar{A}B_{in} + \bar{A}B + B\bar{B}_{in}$$

logic symbol

\therefore Encoder

Multiple input \leftarrow Encodes
 logic \leftarrow multiple output
 \leftarrow Combinational circuit
 (or fewer) \leftarrow Encodes
 n-output lines \leftarrow Input lines
 \leftarrow Lines

- Binary code uses output lines.
- Input goes to the encoder to generate binary or corresponding value.
- Block diagram of encoder figure.
- 8 bits of



- uses output lines of Encoder.
- Input lines to key

8 to 3 Line Encoder :-

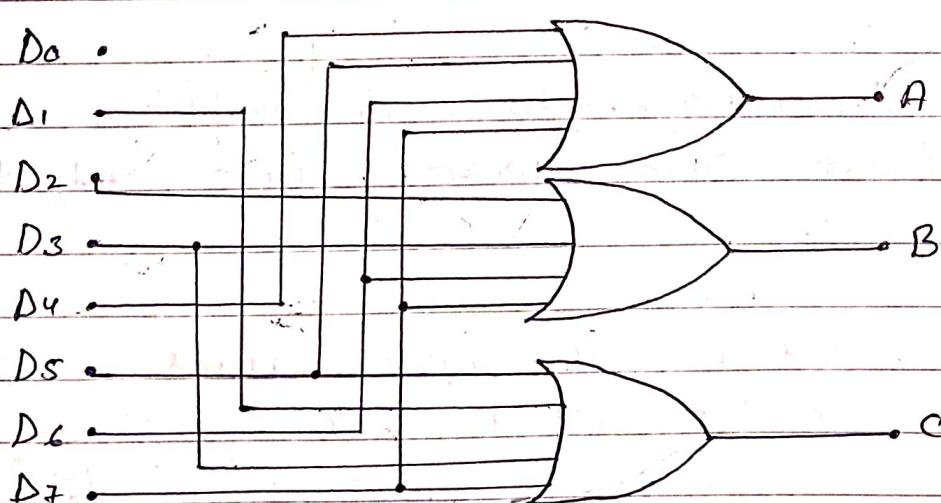
- Binary encoders are simple. It is a 3 outputs for 8 inputs or 1 to 3.
- figures of logic symbol and its logic
- Octal to Binary or 8 bits are Encodes

Octal to 3 encoders. Input
 - 1 digit corresponds to three outputs
 - corresponding to binary numbers.
 generates 3 digits

in which 0 assumed
value '1' in input
- in this case sum = 1

Octal numbers	INPUTS								OUTPUTS		
	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	A	B	C
0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	1
2	0	0	1	0	0	0	0	0	0	1	0
3	0	0	0	1	0	0	0	0	0	1	1
4	0	0	0	0	1	0	0	0	1	0	0
5	0	0	0	0	0	1	0	0	1	0	1
6	0	0	0	0	0	0	1	0	1	1	0
7	0	0	0	0	0	0	0	1	1	1	1

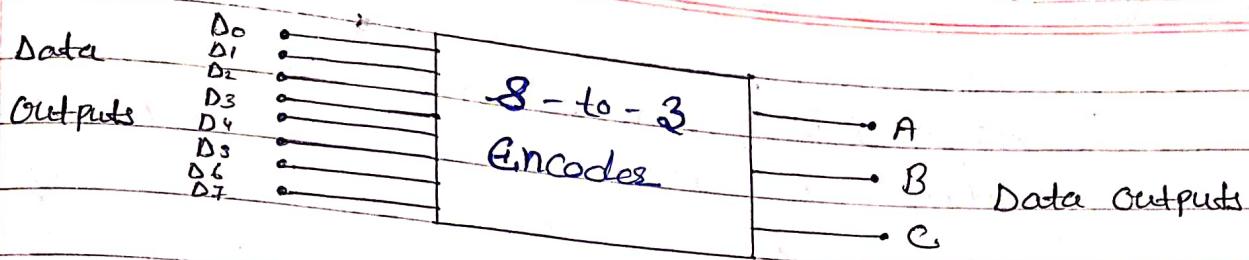
Truth table of octal to Binary Encoder



$$C = D_1 + D_3 + D_5 + D_7$$

Logic diagram B = D₂ + D₃ + D₆ + D₇

$$A = D_4 + D_5 + D_6 + D_7$$



Logic symbol

:- Decodes

Multiple or multiple input \rightarrow Decoder.

Combinational logic circuit to outputs

Output coded \leftrightarrow Input coded \leftrightarrow

\leftrightarrow Input \leftrightarrow \leftrightarrow Convert \leftrightarrow

- \leftrightarrow \leftrightarrow \leftrightarrow Codes output

bits \leftrightarrow \leftrightarrow sub pls of Input code

\leftrightarrow \leftrightarrow also \leftrightarrow Output code \leftrightarrow \leftrightarrow

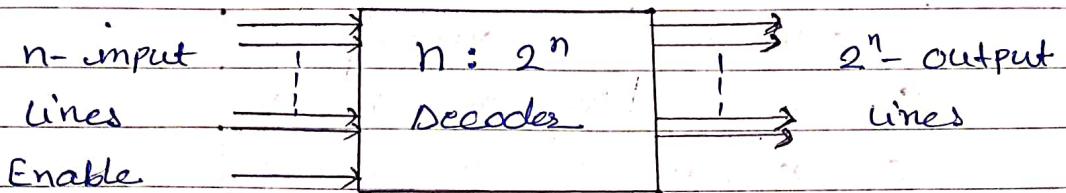
\leftrightarrow \leftrightarrow \leftrightarrow Input code word \leftrightarrow

"Output code word" different

These is a one-to-one mapping
from input code words into output
code words.

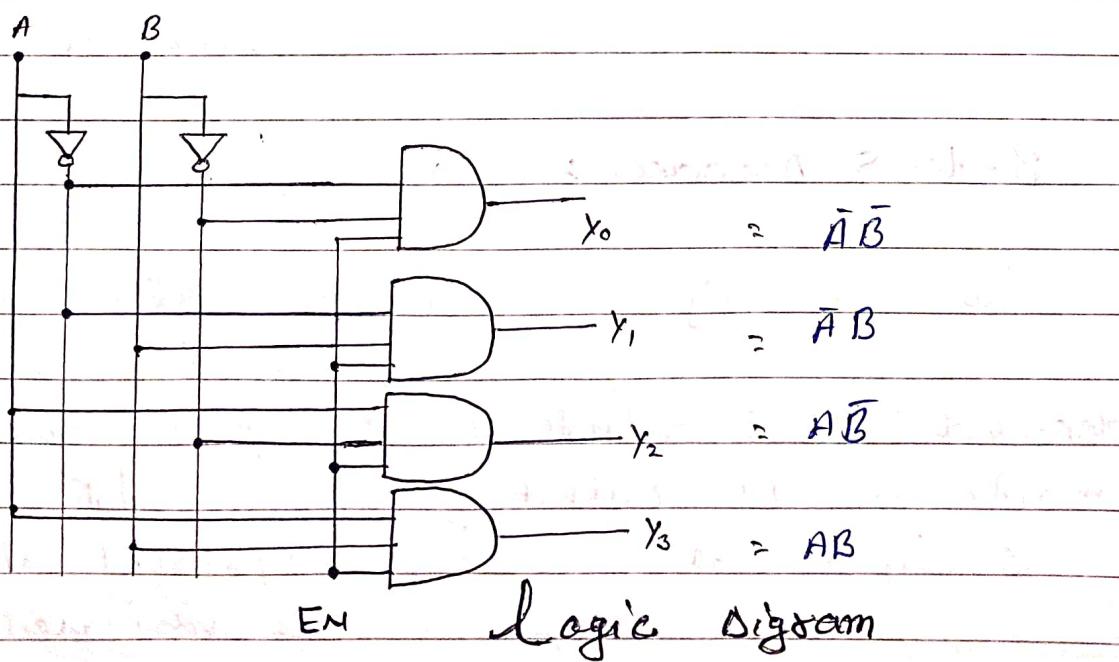
\leftrightarrow Combination \leftrightarrow Input values \leftrightarrow
 \leftrightarrow \leftrightarrow \leftrightarrow 's output \leftrightarrow \leftrightarrow \leftrightarrow
 \leftrightarrow \leftrightarrow \leftrightarrow \leftrightarrow \leftrightarrow \leftrightarrow

\Leftrightarrow n-inputs lines \Leftrightarrow n decades \Leftrightarrow maximum 2^n unique output lines \Leftrightarrow 2^n numbers



2_r to 4 Line Decoders :-

1) decodes 2 to 4 line. (See figure)
 2) Inputs \rightarrow 2 inputs - $\overline{x_1 x_2}$ \rightarrow 4 outputs
 3) Four output - $\overline{x_1 x_2}$ \rightarrow 4 decoded
 represent 4 minterms \rightarrow output
 bit \rightarrow one 2-input variable \rightarrow x_1
 - x_2
 represent as B as A
 Inputs \rightarrow
 y_1 , y_2 , y_3 as outputs \rightarrow $\overline{x_1 x_2}$ \rightarrow 4
 - $x_1 x_2$ as y_3 as



4.

D.

E

2×4
Decoder

y_0
 y_1
 y_2
 y_3

INPUTS			OUTPUTS			
En	A	B	y_3	y_2	y_1	y_0
0	x	x	0	0	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	1
1	1	0	0	0	1	0
1	1	1	1	0	0	0

$$A = y_2 + y_3 \quad \bar{A}\bar{B}$$

$$B = y_1 + y_3 \quad \bar{A}B$$

$$A\bar{B}$$

$$AB$$

c) If '1' value is enable input $\sqrt{1}$
outputs $\sqrt{1}$ will be y_0 to y_3
 $\sqrt{0}$ at input $\sqrt{0}$ is - Key achieve.

$y_0 = 1$ $\sqrt{0}$ - Key achieve $\sqrt{0}$ output y_0
- Key $A = B = 0$

$y_1 = 1$ - Key achieve $\sqrt{0}$ output y_1

Soon 101 - Key $B = 1$ 101 $A = 0$ $\sqrt{0}$
value is outputs when $\sqrt{0}$ $En = 0$ $\sqrt{1}$
- Key '0'

3-to-8 Decoder :

Let's see fig of decoder 3×8 line.

decoded $\sqrt{0}$ 8-outputs w/ 3-inputs $\sqrt{1}$
minterms $\sqrt{1}$ output w/ $\sqrt{1}$ $\sqrt{1}$
3-input $\sqrt{1}$ w/ $\sqrt{1}$ represent w/
 $\sqrt{1}$ w/ $\sqrt{1}$ w/ variables

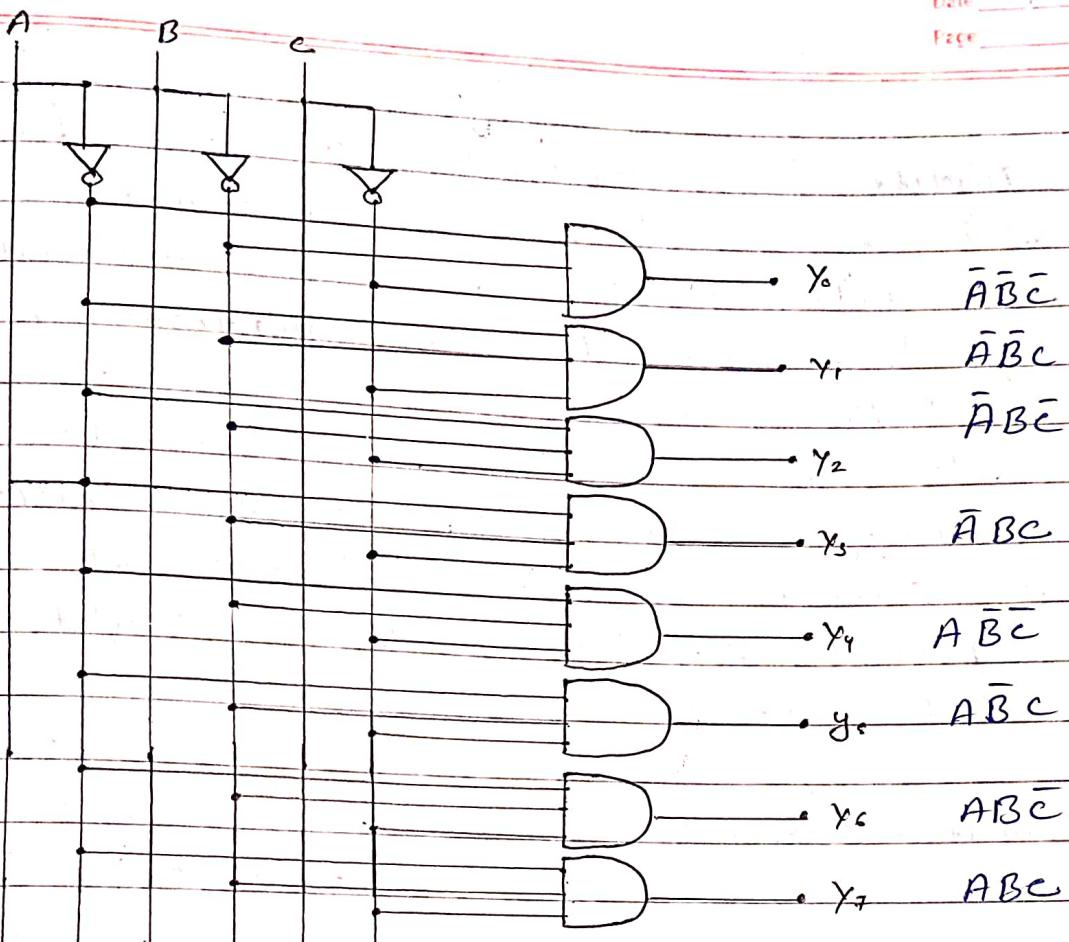
Inputs \Rightarrow Not gates \Rightarrow Compl. fig
 \Rightarrow It is - \Leftrightarrow 3 bits \Rightarrow Complement of
 AND gates \Leftrightarrow It is \Leftrightarrow AND gates
 \Leftrightarrow It generates minterms \Leftrightarrow
 Binary-to-octal \Leftrightarrow decoders \Leftrightarrow
 \Leftrightarrow It is \Leftrightarrow 3 bits \Leftrightarrow converter.
 Binary number \Leftrightarrow Inputs variables
 \Leftrightarrow outputs \Rightarrow It represents \Leftrightarrow
 represents \Rightarrow 3 digits = eight
 Octal number system \Leftrightarrow It is \Leftrightarrow
 \Leftrightarrow

Input-output of decoder 3-to-8
 \Leftrightarrow table. \Leftrightarrow relationship
 \Leftrightarrow 3 bits
 \Leftrightarrow '1' value to line output \Leftrightarrow
 represent \Leftrightarrow minterm \Leftrightarrow 8 - key \Leftrightarrow
 also \Leftrightarrow input lines \Rightarrow 3-bit
 - key equivalent \Leftrightarrow Binary numbers.

\Leftrightarrow variables input \Leftrightarrow C, B, A
 $y_7 \Leftrightarrow x_3 \wedge x_2 \wedge x_1 \wedge x_0$
 \Leftrightarrow output variables

INPUTS			OUTPUTS							
A	B	C	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	0	1	0
1	1	0	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	1

Truth
 Table

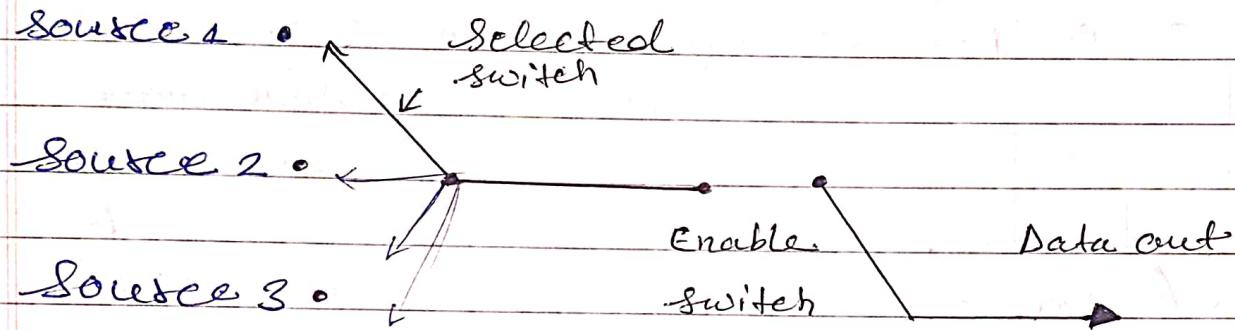


Logic Diagram

: Multiplexers #

Digital switch کی Multiplexes
 "One into many" کو کی Multiplexes.
 Digital information کے لئے
 کوئی کوئی sources کے لئے allow
 کسی signal output line کا
 figure کیا کہ کوئی souted
 Basic کی کیا کیا کیا
 I/P lines کی کیا کیا کیا
 Single output line کی کیا Data
 کو "Data Selector" کی کیا کیا
 کیا کیا کیا کیا

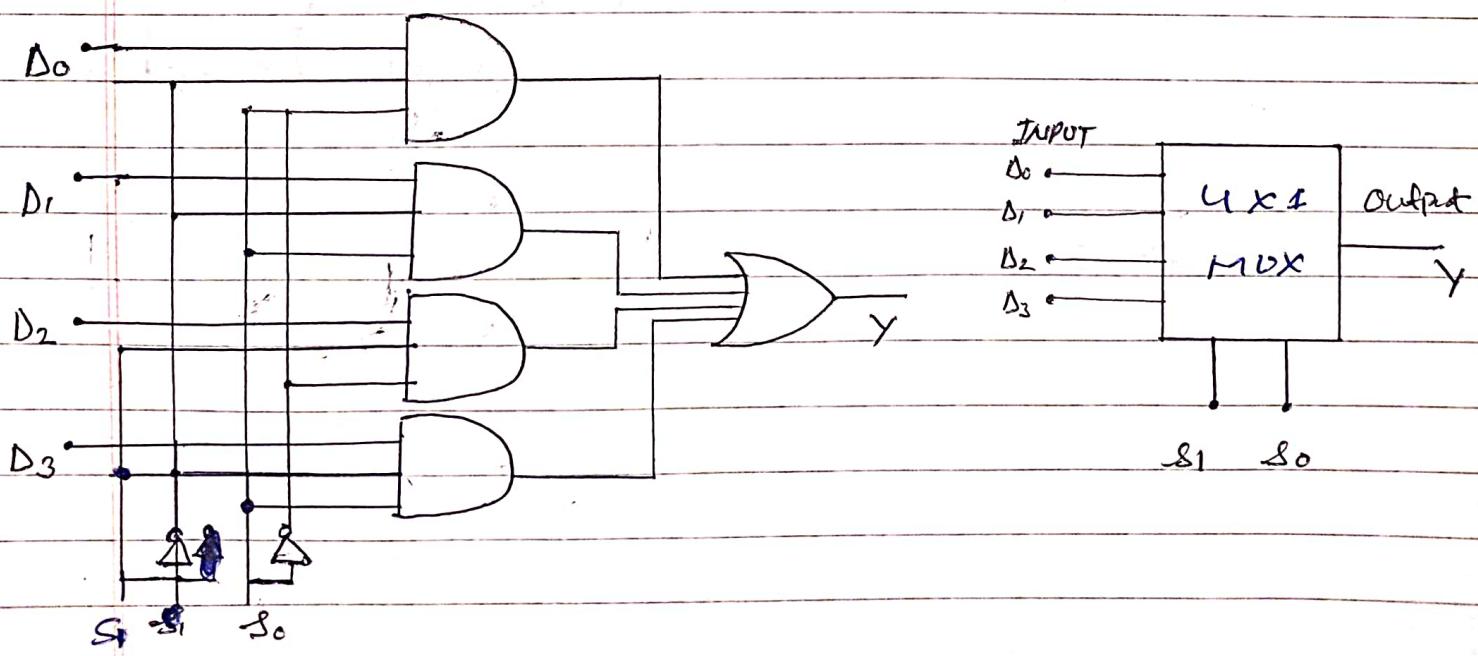
5 input lines, one out
 Set of selection lines & Selection
 pins controlled by
 n input lines '2'
 Combination key - acting Selection lines
 which determine or Bit
 - to select or input L.



Source 4 • Selected switch

-: 4:1 Multiplexes #

-> 4:1 Mux figue



D_3 , D_2 , D_1 , D_0 four inputs
 \Leftrightarrow number k inputs \Leftrightarrow k numbers
 k selection lines \Leftrightarrow k bits
 S_1 , S_0 \Leftrightarrow 2 bits \Leftrightarrow 2 bits
 input \Leftrightarrow 2 bits \Leftrightarrow 2 bits
 \Leftrightarrow 2 bits \Leftrightarrow 2 bits represent \Leftrightarrow 2 bits \Leftrightarrow 2 bits

S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

One logic diagram w. one mux
 \Leftrightarrow 4-And Gate
 \Leftrightarrow AND Gate
 \Leftrightarrow Input

function \Leftrightarrow decode of selection line
 Tables \Leftrightarrow AND gate \Leftrightarrow $S_1 S_0$
 \Leftrightarrow 2 bits \Leftrightarrow select

: Case 1

D_0 \Leftrightarrow AND gate $\Leftrightarrow S_1 S_0 = 00$
 \Leftrightarrow select \Rightarrow associated \Leftrightarrow
 k AND gate \Leftrightarrow 2 bits
 \Leftrightarrow 2 bits \Leftrightarrow '0' value
 \Leftrightarrow 2 bits \Rightarrow $[Y = D_0]$

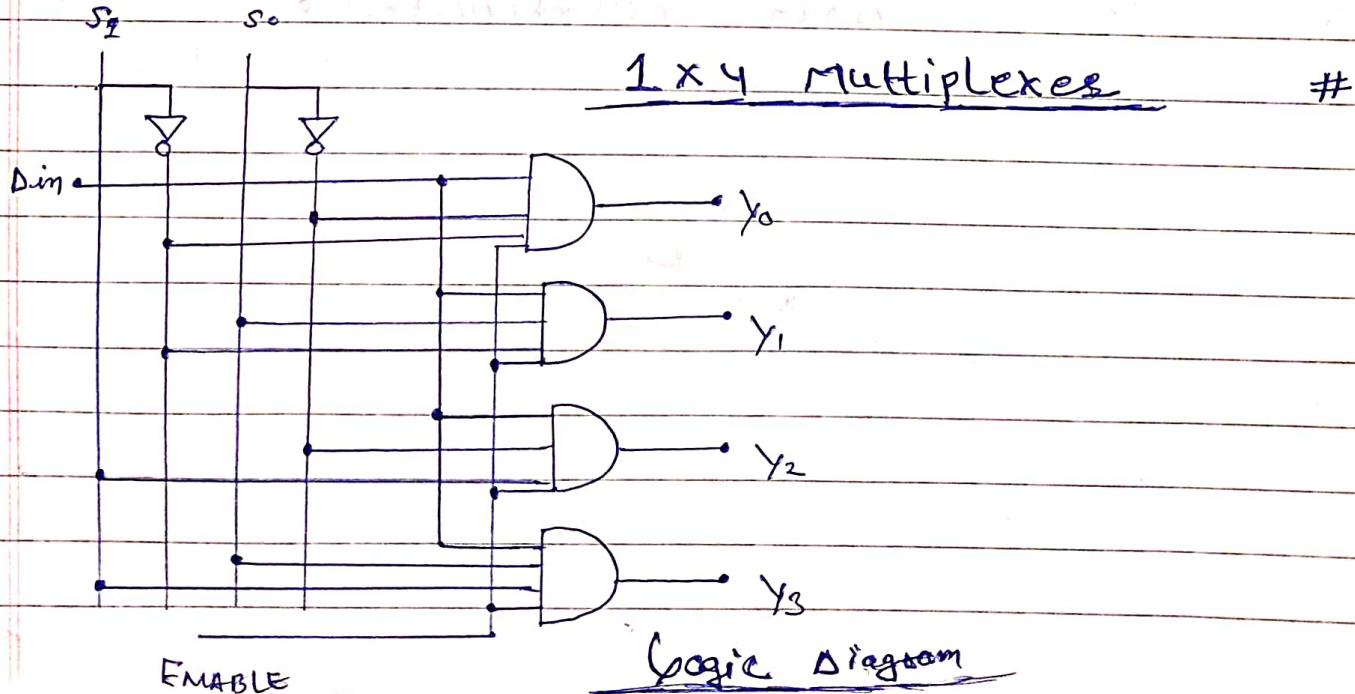
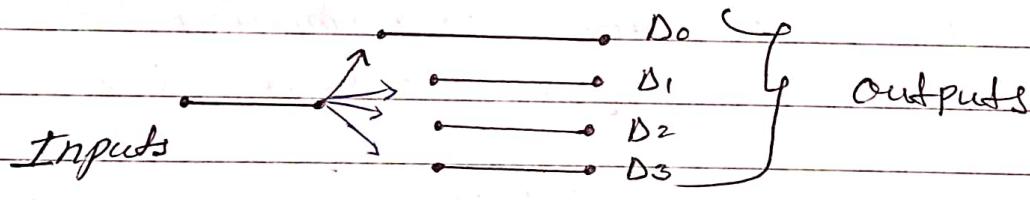
Ans

\therefore Demultiplexers

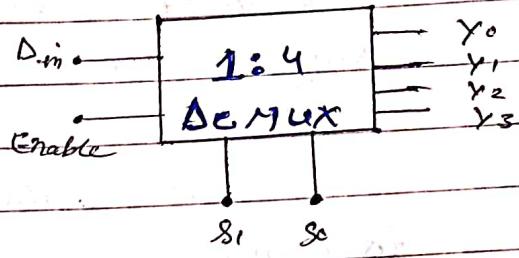
Combinational circuit \leftarrow Demultiplexers.

Reverses multiplexers function \Rightarrow Demultiplexers. \leftarrow logic perform it single \Rightarrow logic circuit \leftarrow receives \Rightarrow information \Rightarrow lines of information \leftarrow \Rightarrow line transmits \Rightarrow 2^n possible output lines

Selection \leftarrow specific output lines bit value \leftarrow 'n' selection lines \leftarrow controlled by enable input = decoders function \leftarrow Demultiplexers. \leftarrow bits



ENABLE	SELECTION LINE		INPUT	OUTPUT
EN	S ₁	S ₀	D _{in}	Y ₀ Y ₁ Y ₂ Y ₃
0	X	X	X	0 0 0 0
1	0	0	0	0 0 0 0
1	0	0	1	1 0 0 0
1	0	1	0	0 0 0 0
1	0	1	1	0 1 0 0
1	1	0	0	0 0 0 0
1	1	0	1	0 0 1 0
1	1	1	0	0 0 0 0
1	1	1	1	0 0 0 1



Block Diagram

Truth Table

- ⑤ 1x4 Demultiplexes or figure
- Single input Or figure - ⑤ 1 1 1 1
- Four outputs or the Path is D_{in}
- ⑤ one input Information Or
- ⑤ directed Or output lines

Or Data distributor

The End

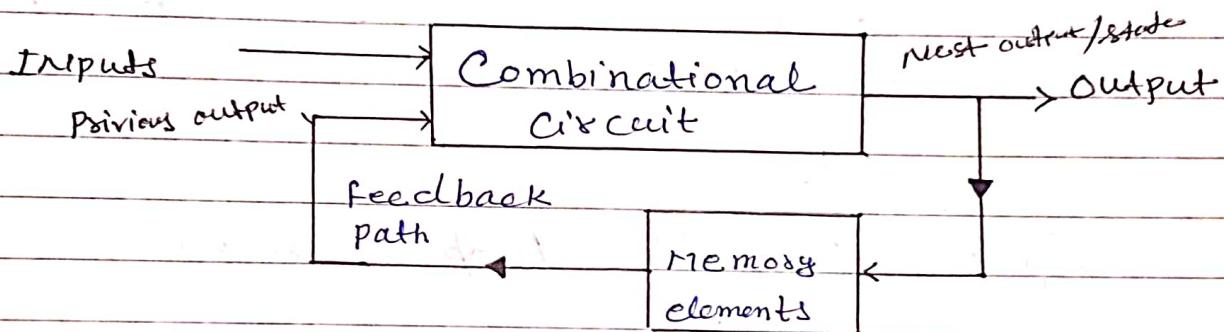
03. Sequential Circuit

Date _____
Page _____

is output from logic circuit but dependent on present input conditions past or inputs or history of output or feedback of history but not only input only or memory elements of circuits to "Sequential Circuit" can include -

Block Diagram of Sequential Circuit

- See figure



Block diagram of a Sequential Circuit

- 1. devices Memory element
 - 2. store Binary information
 - 3. Capable of
 - 4. memory elements are used to store Binary information
 - 5. sequential circuit will be
 - 6. be defined as "state"

-: Flip-flops

اے۔ اے۔ Circuit کی flip flop

Maintains indefinitely at Binary state

switch w/ states کی کسی - کے تو

direct w/ single input کی کے کو

flip-flop کی basic unit کے storage

stable کے جو کوئی device کی کے

or 'SV' لے 'OV' کے two states

کے stores w/ 1-bit information کے

or Units 1-bit memory کے

Types of flip-flops are:-

- ① S-R
- ② J-K
- ③ T
- ④ D flip-flops

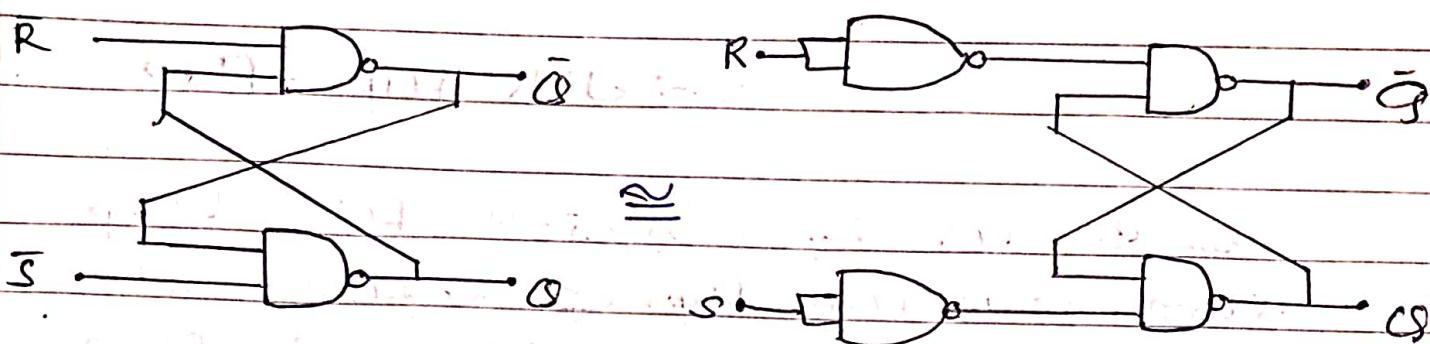
-: S-R flip-flops

Not!

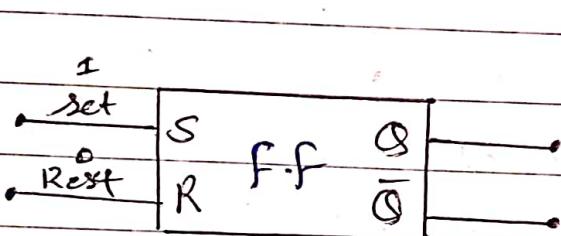
Equivalent a NOR gate, S-R flip-flops

S-R flip-flop کے modify کے

لے کر کے figure کے لے کے



- > list 1 truth table & S-R flip-flop
- > list 2 truth table & R-S flip-flop
- > list 3 logic symbol & figure
- > list 4 state transition & table
- > list 5 logic diagram

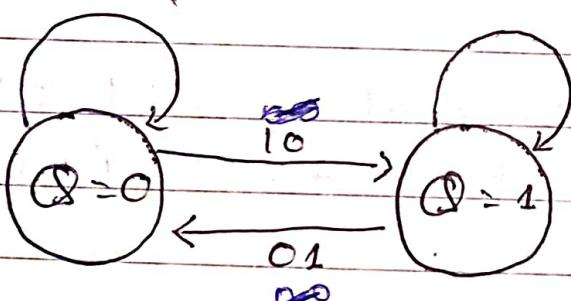


logic symbol.

	1	0	S	R	Q	\bar{Q}	state
Set	0	0	NC	NC	NC	NC	No change
Rest	0	1	0	1	0	1	Reset
	1	0	1	0	1	0	set
	1	1	0	0	0	0	indeterminate

$SR = 00 \text{ or } 01$

$SR = 00 \text{ or } 10$



State-transition

-: J-K flip-flop

↳ R-S flip-flop \Rightarrow J-K flip flop

* \Rightarrow J-K flip-flop - \Rightarrow refinement

state - indeterminate R-S flip flop

\Leftrightarrow R is \Rightarrow Inputs 'K' & 'J'

flip-flop - \Leftrightarrow behave as R, S

- \Leftrightarrow \Leftrightarrow \Leftrightarrow reset \Rightarrow set or

\Leftrightarrow '1' K & J Inputs \Rightarrow state

state flip-flop \Leftrightarrow \Leftrightarrow L & U

\Leftrightarrow switch true & complement

- \Leftrightarrow L & U

e.g.:-

$Q = 1 \Leftrightarrow$ L & switch $\Rightarrow Q = 0 \Rightarrow$ U

- vice-versa

- \Leftrightarrow \Leftrightarrow \Leftrightarrow set condition \Rightarrow

$$Q_{n+1} = \bar{Q}_n \text{ (or)} Q_{(n+1)} = \bar{Q}(t)$$

where,

Q_{n+1} = next state = $Q(t+1)$

Q_n = present state = $Q(t)$

Indeterminate R-S flip-flop \Rightarrow \Leftrightarrow

\Leftrightarrow feedback technique \Rightarrow state

\Leftrightarrow eliminate \Leftrightarrow \Leftrightarrow \Leftrightarrow

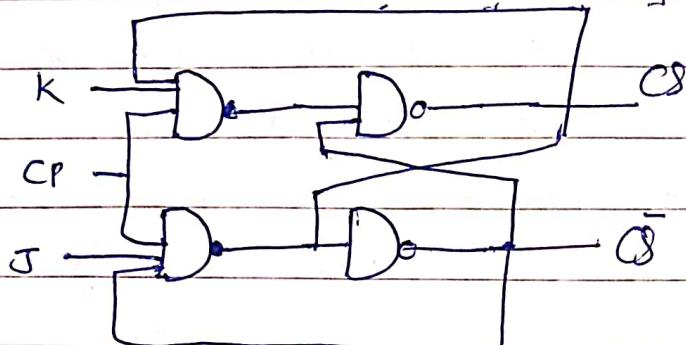
\Leftrightarrow \Leftrightarrow \Leftrightarrow

two cross coupled J-K flip-flop

two AND gates or coupled NOR gates

or it is constructed as shown in

as follows fig.



CP	J	Q	Q(t+1)	J	K	Qn+1
K	Q	Q	Q(t+1)	0	0	Qn
logic symbol	0	0	1	0	0	Qn
	0	1	0	1	1	0
	0	1	1	1	0	1
	1	0	0	1	0	1
	1	0	1	0	1	0
	1	1	0	1	1	1
	1	1	1	0	1	0

Truth table

Q	JK	00	01	10	11
0	0	0	1	1	
1	1	0	0	1	

$$Q(t+1) = \bar{J}Q + \bar{Q}K$$

Characteristic equations.

E D- flip - FloP

gives the truth table of RS flip-flop

\leftarrow It can realize a certain output based on input values but unpredictable. It does not change its behavior.

avoid w' input conditions \leftarrow - \leftarrow lie

Complement to agrees with a verb
- subject

Modified clocked SR flip flop

is also a D-flip flop or

— 191 —

S- input directly or D- Input

Complement of El. S. & U. S.

is it w appley Go R-Input w

$\therefore T$ -flip-flop

"toggle flip-flop" or T flip-flop
 JK flip-flop \Leftarrow تبديلی \Leftarrow \Rightarrow
 $\Rightarrow T$ -flip-flop \Leftarrow Modification ک
 من \Leftarrow JK flip flop \Leftarrow تبديلی
 میں کسی \Rightarrow terminals کی \Rightarrow J
 \Rightarrow $\neg J = K$ \Rightarrow $\neg J$ کو $\neg K$ کا برابر \Rightarrow
 \Rightarrow figure of ~~unlocked~~ - T-flip flop
 \Leftarrow \Rightarrow کیا کیا

- Construct w/ Master-Slave flip-flop #
- w/ J-K flip flops
 - connect \bar{Q} of one to J of another
 - connect \bar{Q} of one to K of another
 - connect J and K of one to C_{in} and \bar{C}_{in} of another
 - connect C_{in} and \bar{C}_{in} of one to J and K of another
 - connect C_{in} and \bar{C}_{in} of one to C_{in} and \bar{C}_{in} of another
- Master-Slave flip flop
 - C_{in} & "slave" Cb of "master"
 - C_{in} & "slave" C_{in}
- Logic Diagram of flip-flop
 - C_{in} & C_{in} as figure
 - slave flip flop flip-flop
 - C_{in} inverted
 - clock pulse of master flip flop
 - drive Cb. at positive edge
 - clock w/ slave flip-flop
 - C_{in} at negative edge of pulse
 - C_{in} at drive

Clock Input 6 masters $Z_{out} \cdot CP = 1$ (1)
 JK Inputs 1 master $Z_{out} = '1'$
 slave Z_{out} - by 1st JK according to
 $Z_{out} = Z_{J1} \cdot Z_{K1}$ (responal) Z_{out}
 ✓ negative edge w.r.t. clock input
 $- Z_{out} = Z_{J1} \cdot Z_{K1}$

'1' clock input 6 slave die epic 227
slave or outputs 6 masters flip flop
masters. يكىن فى 6 flip flops
ست كل دللي (responds) يعنى
positive edge of clock input or it is
لما يجيء

\leftarrow flop master slave flip flop
 \rightarrow tie race around problem

\rightarrow state output K masters flip-flop
determined by inputs $k = 2^J$
Positive clock pulse goes to \rightarrow tie up
 \rightarrow output state of masters

\rightarrow inputs \rightarrow slave flip-flop
 \rightarrow slave flip-flop \rightarrow tie up transformed
 \rightarrow output state of slave J inputs
negative clock \rightarrow tie determined
by \rightarrow pulse

-: Triggering of flip-flops

Synchronous Bistable flip-flops

"Input", output \rightarrow device

\rightarrow specified point \rightarrow "triggering"
 \rightarrow tie changes of state \rightarrow

\rightarrow tie \rightarrow clock pulse (CP) \rightarrow

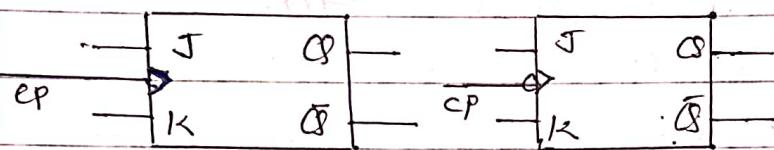
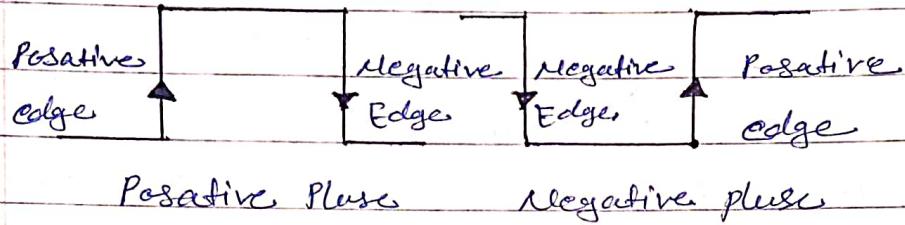
\rightarrow clock changes \rightarrow output
 \rightarrow tie \rightarrow synchronisation

triggered flip-flop or \rightarrow tie \rightarrow flip-flop

Edge triggered flip-flops ①

Level triggered flip-flops ②

⇒ Edge triggered flip-flop ①



positive edge negative edge
triggered flip triggered flip
flop. flop.

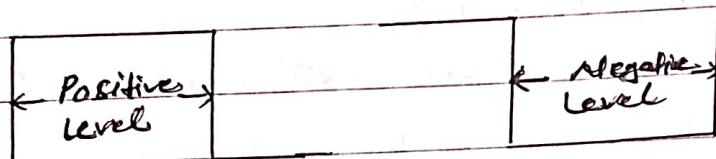
K flip-flop or also K edge triggering
(rising edge) or clock pulse state
negative (falling edge) or positive edge
or flip flop or edge

✓ or output or L indicates or
clock pulse or sensitive or Input
clock pulse or figure or transition or
or edges negative or positive or
or UP or DOWN

or UPward arrow or rising edge
arrow downward or falling edge
negative edge triggering or positive
or L indicates or

-:- Level triggered flip flop

clock or negative clocking or positive
clock or "level triggering" or
clock flip-flop or \leftarrow L.L.
respond to High level L. low at
 \leftarrow L.L.



input or \leftarrow o/p & flip-flop
- \leftarrow L.L. or state also at values
negative or Positive level or figure
 \leftarrow L.L. or level
 \leftarrow L.L. state & flip-flop or
positive level at \leftarrow clock Positive
 \leftarrow L.L. or triggered flip-flop
 \leftarrow L.L. state & flip-flop or
 \leftarrow clock negative
triggered flip flop negative level
 \leftarrow L.L. or

Note:-

- The main drawback of level triggering is that as long as the clock is positive OR negative, the flip-flop changes its state more than once or many times. So the change in inputs if the inputs are made stable for the entire clock duration, then the output changes only once.

Unit 9.11: ~~Basics of Digital Electronics~~ :- Counters

- In electronic circuit it's Counter.
- It has one clock input.
- Count in numbers of clock pulses.
- It is capable to count number of clock pulses and its count.
- Clock input - It is given by clock source.
- It has number of count pulse.
- Pulse originates from external source.
- Prescribed at time of its origin.
- It is done at random or intervals.

↓ List of straight forward binary codes.

- Output is the count.
- Specified 6 states i.e. 0 to BCD will be.
- In counters output sequence appears.
- Used in instruments like counters.
- To measure time.
- In Application.

numbers of occurrences of event like K_1 (2)

- It is also called as count of

objects in industrial process like K_1 (3)

- It is also called as direct counting of

HTML / W.R.U

voltage / or digital circuits \rightarrow digital circuit \rightarrow direct count or pulse current

Digital voltmeters / ammeters

Frequency divider \rightarrow Counters \rightarrow (5)

Counters are classified.

Into two types.

- ① Ripple or asynchronous or non-synchronous or serial counters.
- ② Synchronous or parallel counter.

→ flip-flops \rightarrow synchronous counter
→ blocks connect \rightarrow common clock input
→ clocked \rightarrow previous flip flop \rightarrow asynchronous counter
counter \rightarrow triggered \rightarrow cumulative setting times
or \rightarrow flip flop \rightarrow asynchronous counter
 \rightarrow clocked

→ follow binary sequence \rightarrow counter
n-bit binary counter \rightarrow n-bit binary counter
count on binary \rightarrow 0 to $2^n - 1$ - n bits

\rightarrow LJK g. Mod - N counter \downarrow modules counter

N = number of state as the counter $O/P = 2^n$

Asynchronous or Ripple Counter:-

K flip flop required or ripple counter
 depend on number of states number
 b output states counter - LJK
 Modulus Mod K counter or number
 maximum no. of states Counter - LJK
 flip flop or counter n \rightarrow 2^n
 $- \rightarrow$ number K

or 2 flip flop \rightarrow 2 states \rightarrow 1
 maximum possible no. of output states counter
 Modulus - 4 counter Mod - 4 or 2^2 $4 = 2^2$
 $- \rightarrow$ 2 bits or 2^2 bits

Mod - 4 ripple counter :-

2-bit asynchronous counter \rightarrow Mod - 4 ripple counter
 2 flip flop \rightarrow 2 bits \rightarrow four states \rightarrow 2

\rightarrow let's draw figure of state diagram
 of ripple counter Mod - 4 or figure
 \rightarrow let's draw LJK JK flip flops

flip flop first stage اسے کے لئے اس کے clock T/P clock signal اور کے لئے connect

first stage clock input کے لئے 2nd stage flip flop
- کے لئے triggered اسی Q_A O/P

state K first flip flop کے لئے Note

clock pulse کے لئے کے لئے triggered کے لئے negative edge
of state K flip flop کے لئے negative going transition Q_A O/P کے لئے pulse
کے لئے triggered کے لئے

inherent propagation delay times is 1 flip flop

b) Q_A O/P or transition K Input clock pulse

or کے لئے کے لئے of transition

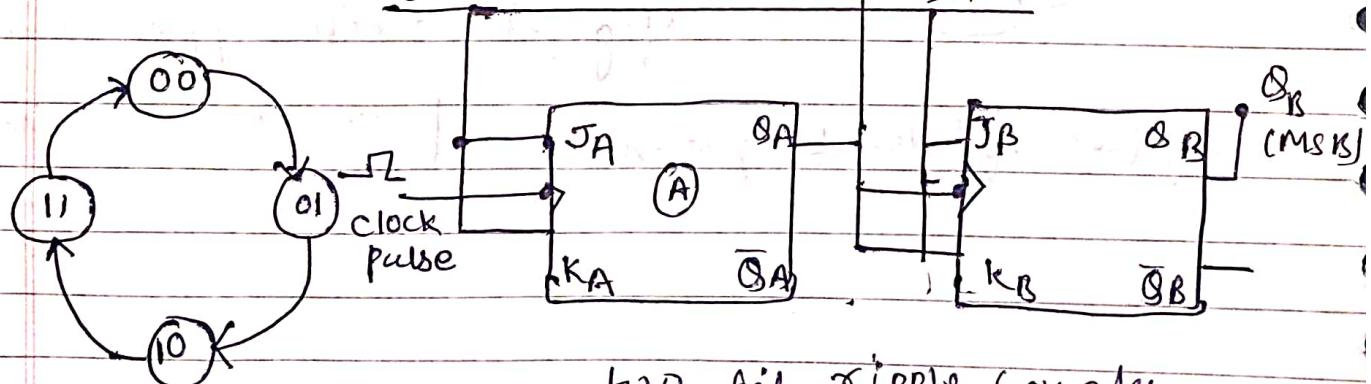
or کے لئے کے لئے flip flop کے لئے - کے لئے

result کے لئے کے لئے triggered

or کے لئے کے لئے asynchronous counter operating

"ripple counters" کے لئے asynchronous counter

Logic کے لئے کے لئے کے لئے



two bit ripple counter

CP	1	2	3	4	
QA	0	1	0	1	0
QB	0	1	1	1	0

Timing Diagram of two bit ripple counter

MOD-8 ripple counter is—
 eight stages \rightarrow 3-bit asynchronous counter
 \rightarrow 2 bits \rightarrow 111 \rightarrow 000
 fig (b) flip flops
 State diagram
 3-bit ripple counter

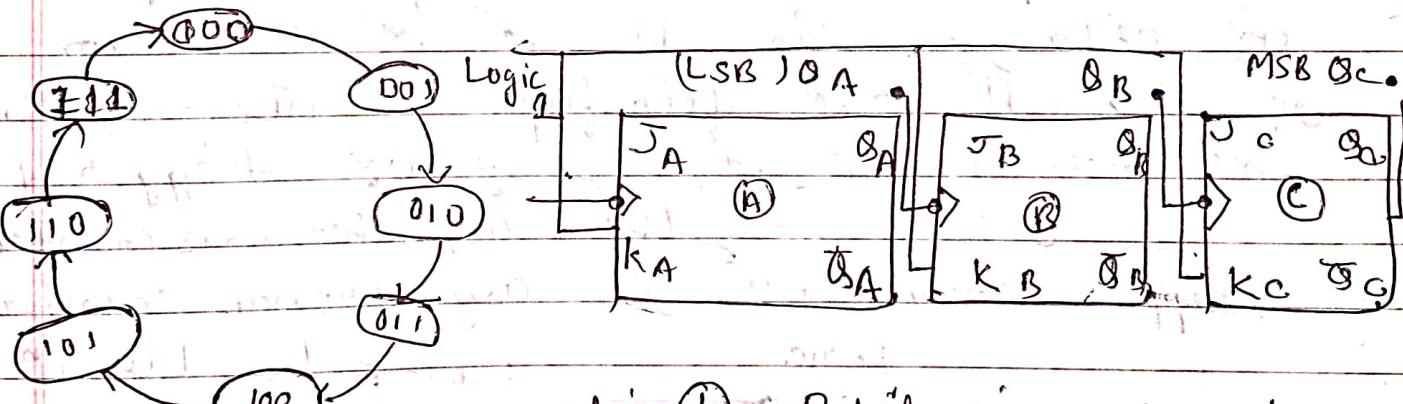


fig (b) 3-bit ripple counter

fig (a) State Diagram

$$\text{Count} = Q = Q_C Q_B Q_A$$

Mo) - 8 ripple counters in 3 JK flip-flops as figure (b)
 count output b̄ C flip flop $\xrightarrow{Q_1 Q_2 Q_3}$
 A flip flop $\xrightarrow{Q_1 Q_2}$ indicate MSB
 indicate LSB count output b̄
 1 clock of first flip flop Q_1 - $\xrightarrow{Q_1 Q_2 Q_3}$
 hold of LSB $\xrightarrow{Q_1 Q_2 Q_3}$ triggered of 2nd flip flop (B) - $\xrightarrow{Q_1 Q_2 Q_3}$
 triggered $\xrightarrow{Q_1 Q_2 Q_3}$ OA O/P of first flip flop - $\xrightarrow{Q_1 Q_2 Q_3}$

o/p of 2nd flip-flop of 3rd flip-flop (c)
 $\xrightarrow{Q_1 Q_2 Q_3}$ triggered of OB 2nd flip-flop (B) $\xrightarrow{Q_1 Q_2 Q_3}$ hold of LSB
 $\xrightarrow{Q_1 Q_2 Q_3}$ OA output of first flip-flop of flip flops ripple counters $\xrightarrow{Q_1 Q_2 Q_3}$
 $\xrightarrow{Q_1 Q_2 Q_3}$ transition outputs of flip flops triggering counters $\xrightarrow{Q_1 Q_2 Q_3}$ flip flop
 $\xrightarrow{Q_1 Q_2 Q_3}$ output of flip flop $\xrightarrow{Q_1 Q_2 Q_3}$

$\xrightarrow{Q_1 Q_2 Q_3}$ 2nd flip-flop of 3rd flip-flop (c)
 $\xrightarrow{Q_1 Q_2 Q_3}$ triggered of OB output flip flop of ripple counters $\xrightarrow{Q_1 Q_2 Q_3}$
 $\xrightarrow{Q_1 Q_2 Q_3}$ flip flop of transition output of flip flops triggering sources $\xrightarrow{Q_1 Q_2 Q_3}$ flip flop
 $\xrightarrow{Q_1 Q_2 Q_3}$ serve

Count Sequence	Conditions for Complementing flip flops
Q _C Q _B Q _A 0 0 0	Initial Condition At the end of 1st clock pulse (Negative edge transition) Q _A Changes from 0 to 1 at the end of 2nd clock pulse, negative.
0 0 1 0 1 0 0 1 1 1 0 0 1 0 1	Edge transition Q _A changes from 1 to 0 and Q _B from 0 to 1. At the end of 3rd clock pulse Q _A change from 0 to 1, and no change in Q _B and Q _C .
1 1 0	At the edge of the 6 th clock pulse, Q _A changes from 1 to 0 And Q _B changes from 0 to 1 and
1 1 1	No change in Q _C . At the end of the 7 th clock pulse, Q _A changes from 0 to 1 and no change in Q _B And Q _C

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

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

Timing diagram of the 3-bit ripple

Decode (or Divide - by - 10 or modula - 10)

Ripple Counters \Rightarrow

$$\text{For } N=10, n = \lceil \log_2 N \rceil \Rightarrow n = \lceil \log_2^{10} \rceil = [3.322] = 4$$

at ten states of Decimal Counters

at 9 $10_1 - \leftarrow$ 5 10_1 follow at sequence

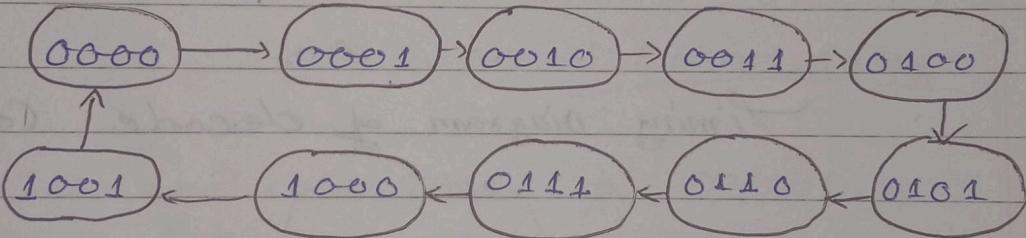
10 $10_1 - \leftarrow$ 6 10_1 is '0' is at Count

at Counters is at ten states \leftarrow
- at 5 10_1 decode Counters

who is display application Counters \leftarrow

- \leftarrow 5 10_1 is at BCD Count \leftarrow

State diagram of sequence at State
 \leftarrow 5 10_1 \leftarrow



State diagram

ten clock pulse of operation is Decode Counters

clock input \leftarrow 5 10_1 is performed \leftarrow at

Connect at first flip flop is at
flip flop 4th 3rd 2nd \leftarrow 5 10_1 is

is triggered \leftarrow at Q_A, Q_B, Q_C is

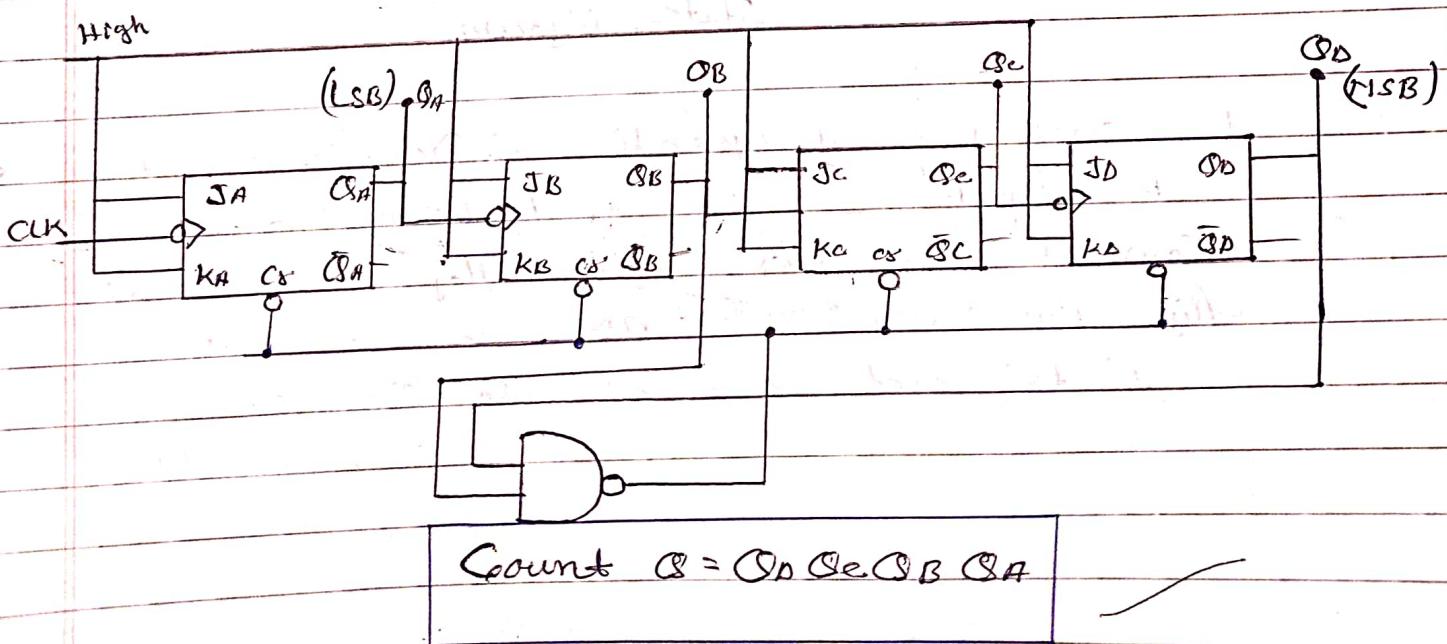
\leftarrow \uparrow

timing diagram of Mod-10 Counter
or Up Counter see figure 5

Clock pulse (CP)	1	2	3	4	5	6	7	8	9	10
Q_A	0	1	0	1	0	1	0	1	0	1
Q_B	0	0	1	1	0	0	1	1	0	0
Q_C	0	0	0	0	1	1	1	1	0	0
Q_D	0	0	0	0	0	0	0	0	1	1
MAXD gate o/p	0	1	2	3	4	5	6	7	8	9

$= \frac{Q}{CS}$

Timing Diagram of Decade Counter



Advantages and disadvantages of Asynchronous counters -

-: Advantages

JK of Counters are very simple. ①

Intersconnect w flip-flops are L.

- or like to construct

uses of fewer components only ②

- or less

or circuit of MOS - Counter ③

- or less to design

-: Disadvantage

- or low speed of operation ①

JK of Counters to get one clock operation speed is less. or less to obtain

- or low importance of it is

1. or less propagation delay ②

- or Draw back

Synchronous Counters

flip-flops also go synchronous counters

Change of state will go to Q₁ & Q₂

or operation of stage j - or k

- or line initialized as in or clock

flip-flops j or counters as well as

connect this or input-clock or

Q₁ to Q₂ flip-flop will rise

- or clocked

Parallel Counter or Counter as it

- or like us see

2-bit Synchronous Counter

so fig of 2-bit synchronous counter

J-K flip-flops or Q_A & Q_B by J-K

negative edge triggered sides so it's

sides of clock signal or J & K

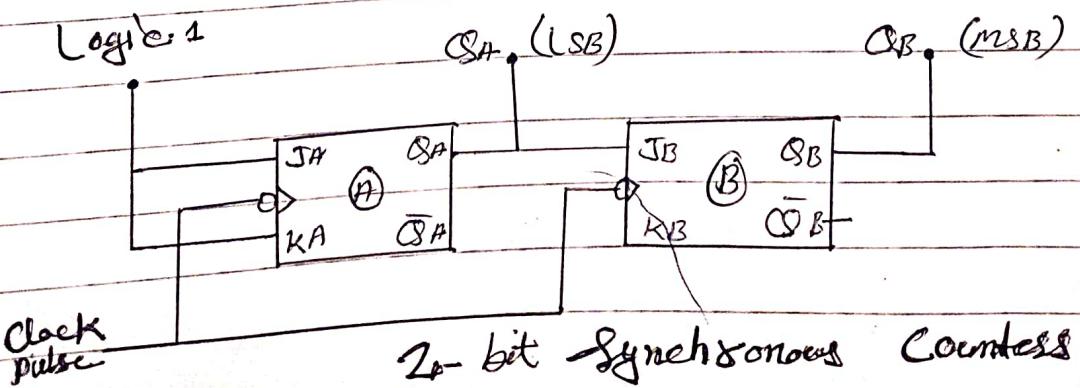
or clock inputs or flip-flops

or we can connect to parallel

J line to Q_A output or first stage

K inputs or J or second stage (B)

- or like us see in or drive or



Operation :-

Initially $Q_A = Q_B = 0$, for the 1st clock pulse, flip flop A will toggle, because $J_A = K_A = 1$ while flip flop B output will remain zero because $J_B = K_B = 0$ for the 2nd clock pulse, both flip-flop will toggle because they both have a toggle condition on their J and K inputs.

Thus after second clock pulse.

$$Q_A = 0 \text{ and } Q_B = 1.$$

for the 3rd clock pulse, flip flop A toggles making $Q_A = 1$. But flip flop B remains set i.e. $Q_B = 1$. Hence at the end $Q_A = Q_B = 1$.

CP	Q_B	Q_A
0, 4	0	0
1, 5	0	1
2, 6	1	0
3, 7	1	1

Clock pulse	1	2	3	4
	0	1	0	1

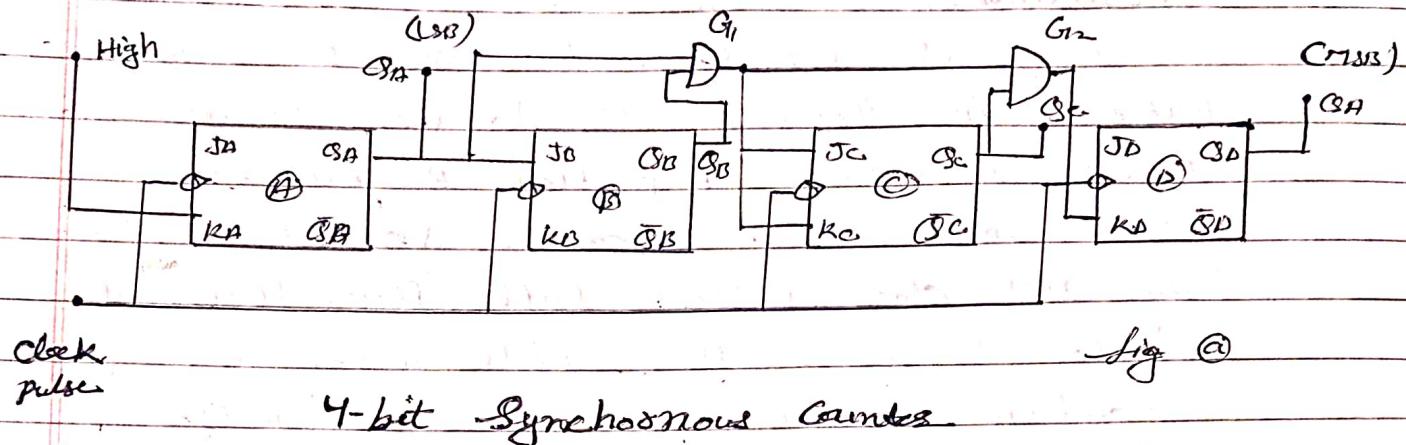
State sequence of 2 bit sync. Counter.

	0	1	0	1	0
Q_A	0	1	0	1	0

	0	0	1	1	0
Q_B	0	0	1	1	0

Fig: Timing diagram of 2-bit sync. Counter:

For, the 4th clock pulse, both flip-flops toggle as their JK inputs are at logic 1. This results in $Q_A = Q_B = 0$ and the counter is recycled back to its original state.



clock pulse

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Q_{A0}	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Q_{B0}	0	0	1	1	0	0	1	1	0	0	1	1	0	0
Q_{C0}	0	0	0	1	1	1	0	0	0	0	1	1	1	0
Q_{D0}	0	0	0	0	0	0	0	1	1	1	1	1	1	1

Fig (2)

Timing diagram of 4-bit sync counters

CP	Q_0	Q_1	Q_2	Q_3	Q_4
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	1

4-bit sync. Counters Go fig @

Counters \rightarrow $\text{LSB} \rightarrow \text{MSB}$ or

$\text{MSB} \rightarrow \text{LSB}$ or flip-flops \rightarrow

$C_3 \rightarrow C_2 \rightarrow C_1 \rightarrow C_0$

LSB Count \rightarrow A flip-flop \rightarrow

A flip-flop \rightarrow $\text{Q}_0 \rightarrow$

$\text{Q}_1 \rightarrow \text{Q}_2 \rightarrow \text{Q}_3 \rightarrow \text{MSB}$ Count

\rightarrow Center

fig (3)

Negative edge triggered "flip-flops" will
 or flip-flops will be flip-flops
 Common clock or clock input terminals
 - or by connecting to pulse
 timing diagram of Counter of
 6 bits or fig w state sequence will
 - or by

: Operation

basic or 4-bit sync. counters
 or timing diagram of operation
 or output of Counter - or by logic
 class or Q
 $Q = Q_0 Q_1 Q_2 Q_3$
 or 6 flip-flops will initially be
 or - bits each or reset condition
 Connect or High to K_A or J_A or
 clock or output Q_0 or or by
 first clock or - bits toggle or pulse
 - bits $Q = 0010$ Count or or pulse
 seventh or sixth, fifth etc or
 Count or Counter or clock pulse
 - bits 0111 or 0101
 Q_A or or or etc or fig
 - bits High value to Q_0 or Q_1
 - bits output to 4th flip-flop

This is an AND gate (G₁₂) at Condition C
 A flip-flop $\leftarrow 101 - \text{clock}\rightleftharpoons$ detect
 edges of 101 - clock state
 output K AND Gate (G₁₂) C = Condition
 state K A flip-flop $\leftarrow 101 - \text{clock} = 0$
 end of Eight clock Pulse $- K = 0$ (initial)

fifteen \leftrightarrow ninth clock pulse. $C_0 = 1$
 Count 6₁₀ Counters \leftarrow E clock pulse.
 sixteen \leftrightarrow -B₀, \leftarrow E 1111 \leftrightarrow 1001
 \leftrightarrow B₁₀ Counters. \leftarrow E clock pulse
 '0' \leftrightarrow '1' state 6 flip-flops
 Count 6₁₀ Counters. \leftarrow -B₁₀ \leftarrow
 $-B_{10} \quad C_8 = 0000$

Ring Counters

"Use of Construction of Counter
last flip flop at Counter Set
first flip- at Counter at output or
Ctrl, or Control input at flop
"direct or as in line by or Connect
"Circulating shift register" ↳ Feed back"
last flip flop is set at line or
Or first flip flop is value. line
↳ ctrl set - or line or Shift
ring Counter or assignments
- or line

at at shows 5 D-flip flop to figure

- at 4 bits or 4-bit ring counter

at connect right out of flip-flops

right to left information or.

right at QD on gate will right at

LS flip flop at MSB flip flop ie

- right at

at registers right to left ie

right at QD on gate will

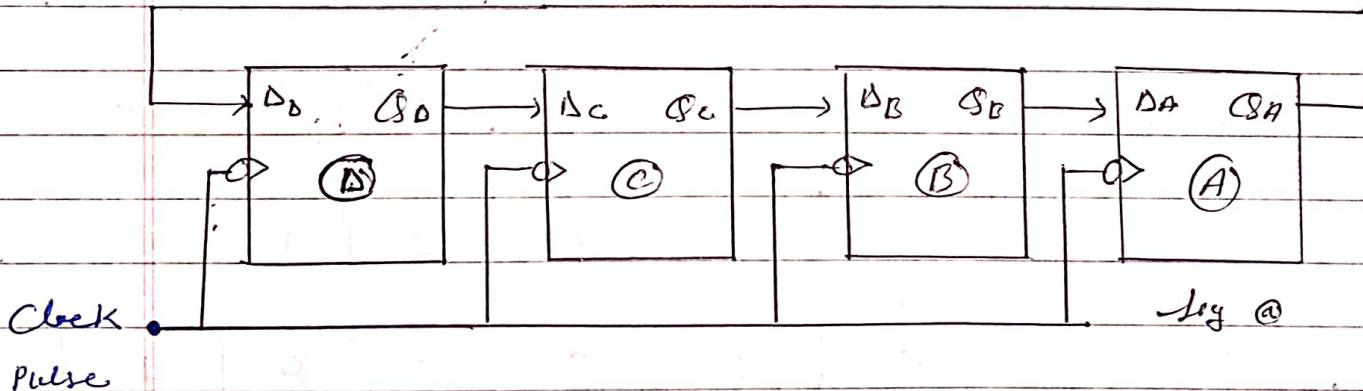
at registers right to left ie

clock will at bit single in

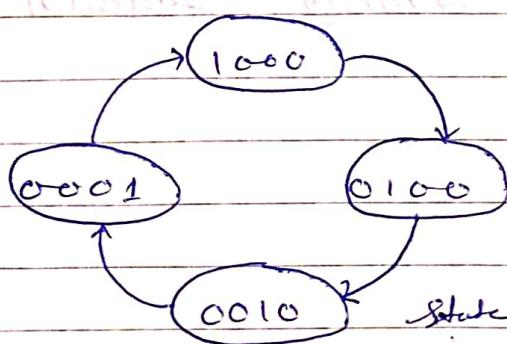
circulate at right apply at pulse

- at bit by

- at 4-bit ring counter at at at



4-Bit ring Counter



CP	C_0	C_1	C_2	C_3	<u>C_4</u>
0	1	0	0	0	0
1	0	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	0
4	1	0	0	0	0
5	0	1	0	0	0
6	0	0	1	0	0
7	0	0	0	1	0
8	1	0	0	0	0

Sequence Table Fig ①

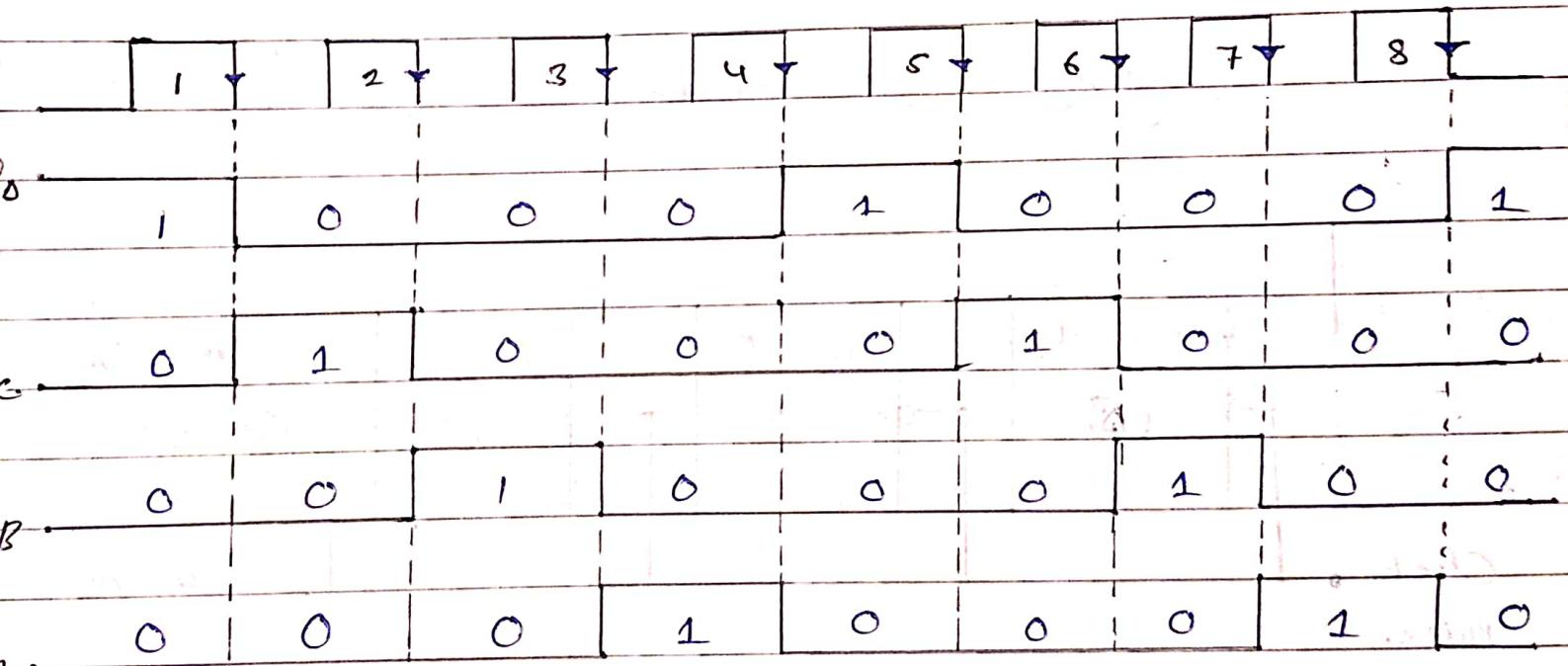


Fig 2 Timing diagram of sing Counters

States various of flip flop \Rightarrow fig(1) ③

can apply of clock pulse of or
- \approx Up to low and - \approx down

1st $CSD = 1$ from starting stable state

- \approx $O_C = O_B = O_A = 0$

O_C or O_B '1' of first clock pulse.

K Counter or \approx - \approx shift up or
Second clock - K or 0100 or 1000 state
estate K Counter at 3rd pulse

- \approx 0001

O_D at 1 K O_A at 4th clock pulse

- \approx 110 as transfer or
1000 state K Counter or 0

- \approx Initial state or \approx 110

Differences between Synchronous And Asynchronous Counters

Asynchronous

only 1st

1st or 2nd Counter flip flop or at ①
first flip flop or \approx 110 or connect
or next flip flop is output is
- \approx 110 desire or clock

2nd or 3rd or flip flop or ②

- \approx 110 or 0 is clocked or

Lecture 1
Complex logic circuit design (3)

- Number of states \rightarrow Complex

Propagation delay - Low speed (4)
Line Problem \rightarrow delay

Synchronous

At a time in
Line connection \rightarrow next output \rightarrow first flip flop
- clock input \rightarrow flip flop

Flip flop will use \rightarrow Clock (2)
- line will be clocked

- Simple logic circuit (3)
- number of states even
- Clock

Problem \rightarrow Propagation delay (4)

The End

UNIT IV

Date _____

Page _____

Digital Memories

-
∴ Digital memories

↪ memories of digital system

data is binary information

- can be stored at right place

↪ use device as memory unit

↪ Semiconductor devices as LSI

↪ magnetic device

↪ Large numbers of memory unit

can store of binary words

- can be used as

↪ Binary bits, Binary words

- can be combination

Semiconductor Memories

Basic types of ROM and RAM

- can memories Semiconductor

Read only Memories ↪

Random Access Memory ↪

Read only memories

Memories \rightarrow Read only memories

Information \rightarrow $\text{G} \rightarrow \text{C}$
stored \rightarrow P ermanently \rightarrow Data

Memory non-volatile \rightarrow ROM

C an C ode is over-written \rightarrow Data

Pointers \rightarrow C an L ib \rightarrow code

Can startup programs \rightarrow software

\rightarrow L ib \rightarrow C an S oft

C an G o \rightarrow Data \rightarrow G o \rightarrow ROM

C an G o \rightarrow G o directly Access

G o RAM \rightarrow data \rightarrow ROM \rightarrow C an G o \rightarrow

C an G o \rightarrow L ib \rightarrow Transfers \rightarrow C an G o

L ib \rightarrow access \rightarrow processes

Advantages of Rom

Advantage \rightarrow C an G o \rightarrow Rom \rightarrow
 L ib \rightarrow non-volatile nature

C an G o \rightarrow Hasol-wiring \rightarrow chip PROM \rightarrow
software \rightarrow C an G o \rightarrow L ib \rightarrow C an G o \rightarrow L ib

Applications of ROM

Data applications of ROM
or computers to store
Updates of software firmware of
BIOS

BIOS of computer to ROM
باینی systems Basic input/output
Computers to viral code to
boot up of operating system in
units, circuit or circuit board

Types of ROM

PROM (i)

EPROM (ii)

EEPROM (iii)

\therefore PROM (i)

Memory Programmable Read only
پروگراممبل ریڈ اولی
memories of memory of objects
and computers program to control
to be stored data
of each version called PROM ROM
to be stored in memory called
Program and manufacturing
called PROM

جہاز جیسا کام
فونکشن

-: Uses

- Used in cell phones (i)
- Consoles video game (ii)
- Medical devices (iii)

EPROM (2)

Erasable Programmable ROM EPROM

- ROM لایا ایں ایں - ROM
 (Erase) لیکوں ور data ور جو گا
 جو کو تجسس کو ایں ایں کو
 - لیکوں لیکوں ور data

-Properties of EPROM

کو ایں ایں ور data کو جو کو
 جو لیکوں ایں UV light کے

memory non-volatile کی ایں

-: Uses

-: EEPROM (3)

Electrically Erasable \leftrightarrow EEPROM

\leftrightarrow One - \leftrightarrow Programmable ROM

- \leftrightarrow Like like \leftrightarrow E² PROM

\leftrightarrow ~~reprogram~~ \leftrightarrow ~~like~~ \leftrightarrow EEPROM

Electronic devices in computing

right store of data \leftrightarrow like this go go

- \leftrightarrow Like like \leftrightarrow like

\leftrightarrow \leftrightarrow \leftrightarrow \leftrightarrow Electric single

\leftrightarrow remove \leftrightarrow data \leftrightarrow content

- \leftrightarrow Like

\leftrightarrow data Byte By Byte \leftrightarrow EEPROM

- \leftrightarrow Like like \leftrightarrow delete

(Random Access memory) RAM (II)

Main memory \leftrightarrow Computer Ram

directly \leftrightarrow \leftrightarrow Like Past \leftrightarrow

- \leftrightarrow Like accessible \leftrightarrow CPU

\leftrightarrow \leftrightarrow \leftrightarrow write \leftrightarrow Read \leftrightarrow

- \leftrightarrow Like like

\leftrightarrow Like memory volatile \leftrightarrow \leftrightarrow

\leftrightarrow \leftrightarrow \leftrightarrow off power \leftrightarrow

\leftrightarrow \leftrightarrow lost information \leftrightarrow

- \leftrightarrow

Dynamic RAM

1113

-Dynamic MOS RAM or DRAM

-Metal Oxide Semiconductor (MOS)
Consists of transistors with capacitors

-Semiconductor memory DRAM

Computer uses web jobs to store code and program data
Code and program data is stored in DRAM

Advantages

-Cheaper than SRAM

-Power consumption less than SRAM

-Only one transistor in design

-Less cost per bit than SRAM cost but

-Also slower than SRAM

-(less) speed

Date _____
Page _____

Address line
Row

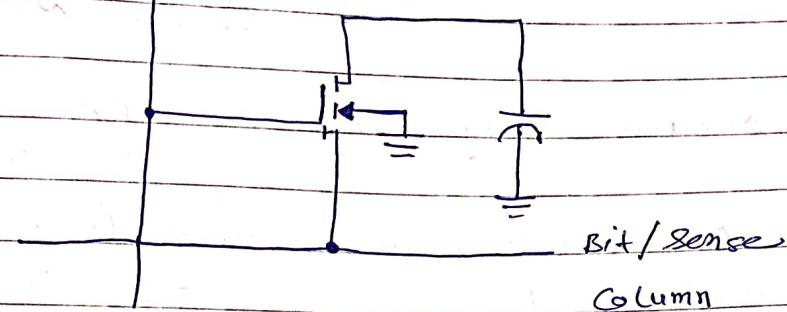


Fig :- Dynamic RAM

- Flash Memory

electronic Flash memory

memory storage non-volatile

electrically medium

reprogrammed (erased)

-

type main Flash memory

-

NOR Flash

NAND flash

Flash NAND NOR Flash

- fast

Kiss flash NAND

-

NOR NAND flash

store w/ data capacity

-

device memory NAND flash

- access serially