ECET syllabus

12 modules

1) Number system
   - Binary
   - octal
   - decimal
   - Hexa decimal

   1m

   special Codes
   - gray Code → 1's complement & 2's complement
   - BCD Code
   - Excess-3 Code

2) logic families – 1m

3) Combinational circuits
   - Adder
   - Subtractor
   - encoder, decoder
   - multiplexer, De multiplexer
   - comparator

   2m

4) sequential circuits
   - Latches
   - Flipflops
     - ↓ JK, SR, D, T
   - Counters
     - synchronous
     - Asynchronous
   - Registers
     - ↓ SISO, SIPO, PIPO, PISO

5) A/D Conversion & D/A Conversion

   (3m)

   A → □ → D

   D → □ → A

6) Basic memories (1m)

   RAM ROM

7) logic families (1m)
   - ↓ RTL
   - DTL
   - TTL
   - ECL
   - CMOS

Binary no. system

base = 2, range = 0 – 1

max value → (b – 1)

Octal no system

base = 8, range = 0 – 7, max value = (b – 1) → (8 – 1) = 7

Hexadecimal no system

base = 16, range = 0 – 15, max value = (b – 1) → (16 – 1) = 15

↓

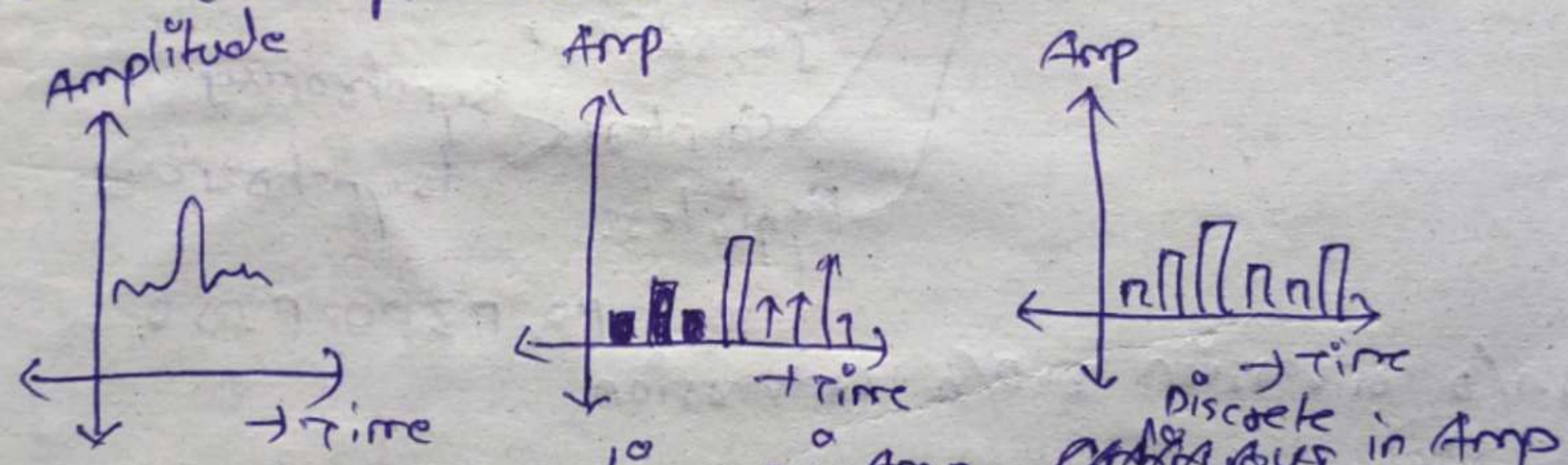
0 – 9, A, B, C, D, E, F

Decimal no system

base = 10, range = 0 – 9, max value = (b – 1) → (10 – 1) = 9

Digital Electronics

→ In this system we deal with only binary number which can be obtained from analog signal by using the following methods.



Continuous in Amp
Continuous in time

Continuous in Amp
Discrete in Time

Discrete in Amp
Discrete in time

Analog signal → [Sampler] → [Quantizer] → [Encoder] → Binary signal

→ A Digital signal discrete in time & also discrete in amplitude

→ the processing of data in digital form is easier than in analog form

Decimal to Binary

base = 10, range = 0–9

Ex: $(100)_2$ convert to Binary

2	100	
2	50	– 0
2	25	– 0
2	12	– 1
2	6	– 0
2	3	– 0
	1	– 1

$(1100100)_2$

Note:
To convert any decimal no system into other no system we take to divide with base (b) & note down the remainder values.
→ For integer conversions we will take remainder values from bottom to top position

Fraction part

Ex: $(0.75)_{10}$ to $(\ )_2$

$0.75 \times 2 = 1.5 \rightarrow 0.5$ with carry "1"
$0.5 \times 2 = 1.0 \rightarrow 0.0$ with carry "1"
$= (0.11)_2$

→ For fraction part conversions we take down the carry from top to bottom position

Decimal to octal

Ex: $(30)_{10}$ to $(\ )_8$

8	30	
	3	– 6

$(36)_8$

Fraction part

Ex: $(0.2)_{10}$ to $(\ )_8$

$0.2 \times 8 = 1.6 \rightarrow 0.6$ with carry "1"
$0.6 \times 8 = 4.8 \rightarrow 0.8$ with carry "4"
$0.8 \times 8 = 6.4 \rightarrow 0.4$ with carry "6"
$= (0.146)_8$

Decimal to Hexadecimal

Ex: $(30)_{10}$ to $(\ )_{16}$

$16 \overline{)30}$

$1 - 14 \rightarrow E$

$\Rightarrow (1E)_{16}$

fraction part

Ex: $(0.20)_{10}$ to $(\ )_{16}$

$0.20 \times 16 = 3.2 \rightarrow 0.2$ with carry 3

$0.2 \times 16 = 3.2 \rightarrow 0.2$ with carry 3

$= (0.33)_{16}$

octal to binary

octal $\rightarrow$ base $= 8 = 2^3$

binary $\rightarrow$ base $= 2$

→ To represent an octal no we have to use "3" digits of binary.

Ex: $(765)_8$ to $(\ )_2$

111 110 101 $= (111110101)_2$

$(10010010)_2$ to $(\ )_8$

MSB LSB

010 010 010

$(222)_8$

octal to decimal

Ex: $(765)_8$ to $(\ )_{10}$

$= 7 \times 8^2 + 6 \times 8^1 + 5 \times 8^0$

$=$

Note:

To convert any no system into decimal no system we have to consider the positional values of given no. system

$(0.25)_8$ to $(\ )_{10}$

$= 2 \times 8^{-1} + 5 \times 8^{-2}$

$= \frac{1}{4} + \frac{5}{64}$

octal to Hexadecimal

base = 8 $= 2^3$ base = 16 $= 2^4$

* O → D → HD
* O → B → HD

$(76)_8$ to $(\ )_{16}$

$(111110)_2$

0011 1110

3 E

$(3E)_{16}$

Note :
Adding zeroes in the MSB positions doesn't effect the value of any no system

Hexadecimal to binary

Ex : $(79)_{16}$ to $(\ )_2$

$16 = 2^4$ → 4 digits of binary to each HD digits.

$(0111\ 1001)_2$

Ex : $(1010.1010)_2$ to $(\ )_{16}$

$(AA)_{16}$

for fraction part we need to group from left ~~right~~ position to right position

Hexadecimal to octal

base = 16 $= 2^4$ base = 8 $= 2^3$

Ex : $(2E)_{16}$ to $(\ )_8$

Convert into binary, Replace HD digit with 4 binary digits.

* H → B → O

$(00101110)_2$

binary to octal

000 101 110

0 5 6

$= (56)_8$

Hexadecimal to Decimal

Ex: $(1E)_{16}$ to $(\ )_{10}$

$= 1\times16^1 + 6\times16^0$

$= 16+14$

$= (30)_{10}$

Binary to Decimal

Ex: $(1011)_2$ to $(\ )_{10}$

$= 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0$

$= 8+0+2+1$

$= (11)_{10}$

fractional part

$(0.11)_2$ to $(\ )_{10}$

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

$= 0.5 + 0.25$

$= (0.75)_{10}$

problem

1) convert $(756.603)_8$ to $(\ )_{16}$

$(111\ 101\ 110 . 110\ 000\ 011)_2$

$(0001\ 1110\ 1110 . 1100\ 0001\ 1000)$

$(1EE . C18)_{16}$

2) Find $(2789.65)_b$

a) 9 b) 8 c) 10 d) 7

→ the max value in given no system is 9 that means the base is 9

3) if $(23)_x = (54)_y$ then the possible values of x & y

a) 8 & 16 b) 10 & 12 c) 9 & 13 d) 8 & 11

decimal kp convert karegal?

$(73)_8 = (54)_{11}$

$7\times8^1 + 3\times8^0 = 5\times11^1 + 4\times11^0$

$= 56+3 \qquad = 55+4$

$= 59 \qquad = 59$

Special Codes

There are 3 types of special codes

1) BCD Code (or) sequential Code 2) Excess-3 Code (or) self complement code 3) Gray Code (or) reflecting Code

BCD Code (Binary Coded Decimal)

→ It is a special code in which 4 bits are used to represent each no.

→ It is also called Binary weighted Code

→ Values range from 0–9.

Decimal no	BCD (8421)	Excess3Code
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

→ For "n" no of bits the possible combination $= 2^n$

→ Range of values $= 0 - 2^{n-1}$

→ For 4 bits the total no. of combinations $= 2^4 = 16$

→ The no. of used states in BCD $= 10$ i.e, 0–9

→ The no. of unused states in BCD $= 6$ i.e, 10–15

Excess 3 Code

→ This is another form of BCD Code in which the code for each decimal digit can be obtained by adding decimal no "3" in binary form to the natural BCD Code

→ It is called self complement Code which is very much useful in performing subtraction operation in Digital systems

Note :

Excess-3 code is a 4-bit code / It is only unweighted code which is self complementary code

Convert decimal no to BCD code

Ex : $(156)_{10}$ to BCD

$(0001\ 0101\ 0110)_2$

Convert binary no to BCD code

Ex : $(1000\ 0101\ 1001)$ to BCD

$(859)_{BCD}$

which is not a BCD code

a) 0101 b) 1110 c) 01100 d) both b & c

Gray code

→ It is also called unit distance code or also minimum error code

EX-OR logic gate truth table

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

$G_3 = B_3$

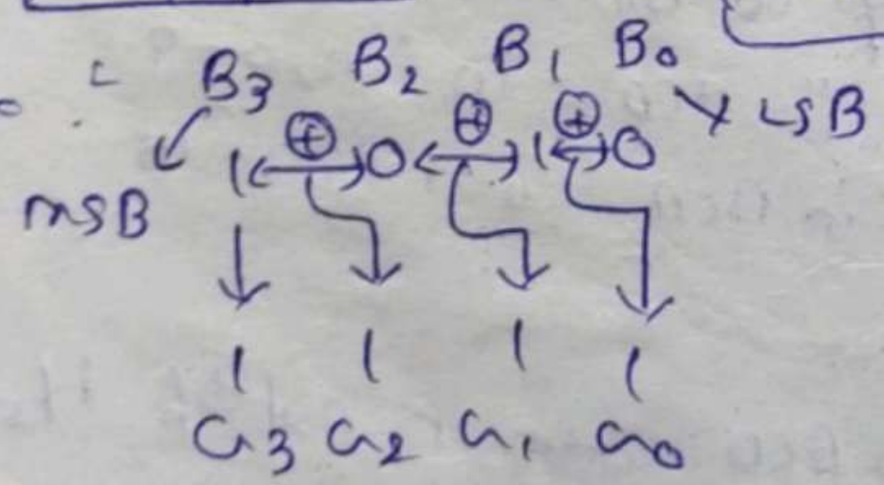
$G_2 = B_3 \oplus B_2$

$G_1 = B_2 \oplus B_1$

$G_0 = B_1 \oplus B_0$

$$\boxed{G_n = B_n \mid G_{n-1} = B_n \oplus B_{n-1}}$$

Ex :



Procedure

1) Copy the MSB bit as it is
2) perform the EX-OR logic operation b/w previous i/p & present i/p
3) copy the result in the present o/p as gray code

Note:

→ Gray code can't be used for arithmetic operations

→ It is also a Non-weighted code

Gray code to Binary code

G_3	G_2	G_1	G_0
1	1	1	1
1	0	1	0
B_3	B_2	B_1	B_0

$B_3 = G_3$

$B_2 = B_3 + G_2$

$B_1 = B_2 + G_1$

$B_0 = B_1 + G_0$

$$\boxed{B_n = G_n \mid B_{n-1} = B_n \oplus G_{n-1}}$$

procedure

1) copy MSB bit as it is
2) perform X-OR operation b/w previous o/p & present i/p
3) copy the result as the next bit

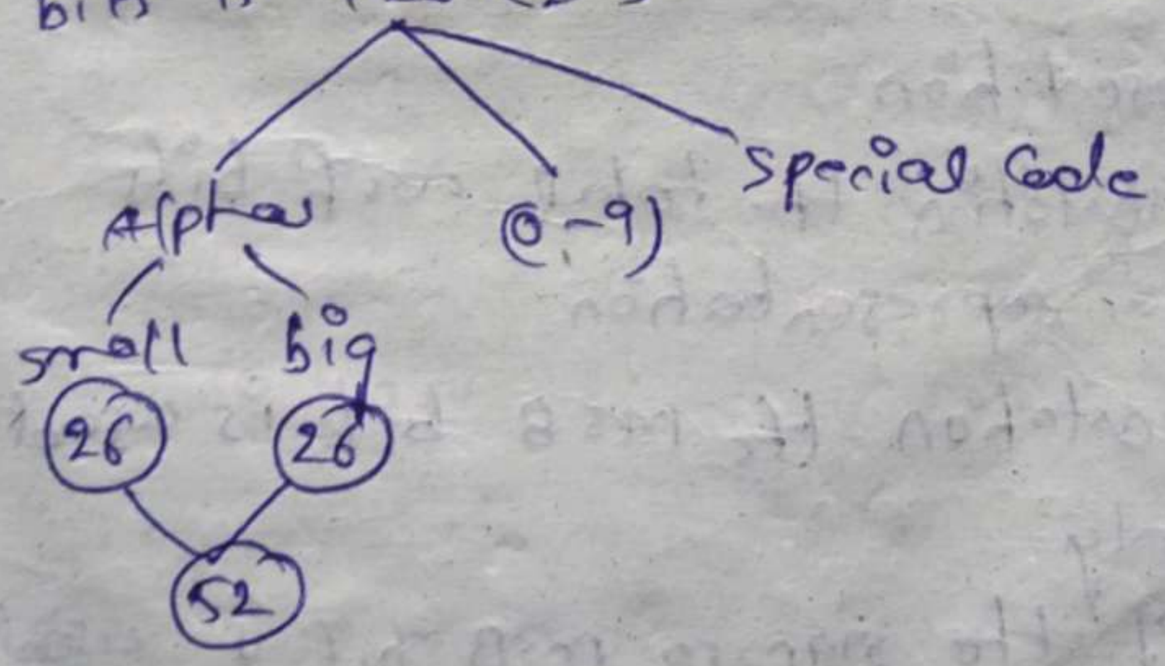
→ Gray code is a reflecting code

Alphanumeric Codes

1) ASCII Code :- (American standard code for info. interchange)

→ It is basically a 7-bit code, $2^7 = 128$, range = 0 to 128

→ the no. of patterns that can be created with 7 bits is 128 (2^7)



EBCDIC code (Extended binary coded decimal interchange code)

→ It is an 8-bit alpha numeric code, $2^8 = 256$

→ It is used by most large computer to communicate alphanumeric data.

Binary addition

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

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

$Carry = AB$

Binary subtraction

A	B	Diff	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

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

$Borrow = \bar{A}B$

Data Representation

→ In DC we can represent diff. no's in the following 3 ways

1) signed representation
2) unsigned representation
3) Complement's representation

→ In unsigned Representation the total no of bits are used for value representation

→ In signed Representation the MSB bit is used to represent sign only

For ex: In a 5 bit the sign no MSB bit is used for sign representation only

the remaining 4 bits are used for value representation

→ In digital computers the subtraction operation takes more time compared to addition bcoz of some complexity. To overcome this problem we are using complement's method

→ For a binary no system we have 2's & 1's complement

→ For a decimal no system we have 10 & 9's complement

<u>1's complement</u>

finding 1's complement we need to complement each & every bit of given binary no.

<u>Ex</u>:- 1001 → Binary no

0110 → 1's complement

<u>2's complement</u>

By adding 1 to the LSB position of 1's complement no we get 2's complement

<u>Ex</u>: 1001 → Binary no

0110 → 1's c.

+1

<u>0111</u> → 2's c.

<u>steps to for 2's complement</u>

→ start scanning from LSB position until we encounter digit 1

→ Copy this bits as it is until digit 1

→ Complement the digits from the next bit onwards

<u>Ex</u>: 1001 → Binary no

↓

<u>0111</u> → 2's c.

Decimal no	Binary no	1's C	2's C
-4	0100	1011	1100
-3	0011	1100	1101
-2	0010	1101	1110
-1	0001	1110	1111
-0	0000	1111	0000
+0	0000	0000	0000
1	0001	0001	0001
2	0010	0010	0010
3	0011	0011	0011
4	0100	0100	0100

→ for n-bit no the possible range of 1's complement is → $-(2^{n-1}-1)$ to $(2^{n-1}-1)$

if $n=8$ → -127 to $+127$

→ for n-bit no the possible range of 2's complement is → $-(2^{n-1})$ to $(2^{n-1}-1)$

if $n=8$ → -128 to $+128$

if $n=5$
then 2's complement representation of range is

-2^{n-1} to $2^{n-1}-1$ — 1's C

$= -2^{5-1}$ to $2^{5-1}-1$ — -15 to $+15$

$= -2^4$ to 2^4-1

$= -16$ to 15

which of the following is ~~Draw~~ the best method for addition of negative number

1) 1's Complement
2) 2's Complement
3) Both 1's & 2's Complents

Addition using 2's complement

1) for the addition of +ve no's there is no need of taking complements of no's
2) for -ve no's we need to convert into 2's complement form in order to perform the addition

Ex: $(20)_{10} - (3)_{10}$

$20 \rightarrow (10100)_2$

$2 \,|\, 20$
$2 \,|\, 10 - 0$
$2 \,|\, 5 - 0$
$2 \,|\, 2 - 1$
$1 - 0$

$2 \,|\, 3$
$1 - 1$

$(00011)_2$ perform 2's complement to get "-3"

$-3 \Rightarrow (11101) \rightarrow -3$

$\Rightarrow (10100) + (11101)$ — $20 + (-3)$

$$\begin{array}{r} 10100 \\ +\ 11101 \\ \hline 10001 \end{array}$$

reject carry

→ From the above problem we have observed that "1" carry is coming out from MSB position. we can reject this carry & the observed is true.

By using 1's complement

Ex: $(20)_{10} - (3)_{10}$

$(20)_{10} \rightarrow (10100)_2$

$(3)_{10} \rightarrow (00011)_2$

$1'sc = 11100 \rightarrow -3$

$(20)_{10}$ 10100 $\quad (20) + (-3)$

$(-3)_{10} \rightarrow$ 11100

$$\begin{array}{r} 10000 \\ +1 \\ \hline 10001 \end{array}$$

carry

Ex: $(3)_{10} - (20)_{10}$

$(3)_{10} \rightarrow (00011)_2 \quad (20)_{10} \rightarrow (10100)_2$

$2'sc = 01100 \rightarrow (-20)_2$

$$\begin{array}{r} 00011 \\ +\ 01100 \\ \hline 01111 \end{array}$$

there is no carry so we perform 2'c

$\boxed{(10001)_2}$

Note: From the above problem we can observed that there is no carry out from the MSB bit that means the result is wrong. In this case we need to perform 2's c to the answer

Ex: $(3)_{10} - (20)_{10}$ ← By using 1's complement

$(3)_{10} \rightarrow (00011)_2$

$(20)_{10} \rightarrow (10100)_2$

1's c: $(01011)_2$

```
  00011
  01011
  -----
  01110
```

1's c = $(10001)_2$ → answer

there is no carry ∴ the ans is wrong so complement the result by using 1's c

Note: In case of 2's c the carry out is present from MSB position then the answer is right & there is no need to complement the answer.

→ the carry out is present of result / absent we need to take 2's c

→ In case of 1's c the carry is present we need to add this carry to the LSB position

→ the carry out is absent the answer is wrong we need to take 1's c of the result

1) which of the following no system representation has two zero's

a) unsigned R b) 2's c.R c) 1's c.R d) None

2) the range of signed decimal no's that can be represented for 6-bit no by using 1's c.

$F = -(2^{n-1}-1)$ to $+(2^{n-1}-1)$

$= -(2^5-1)$ to $+(2^5-1)$

$= -31$ to $+31$

which of the following is invalid BCD code

a) 0011 (3) b) 1101 (13) c) 0101 (5) d) 1001 (9)

which of the following is invalid Excess-3 code
3 nuchi start
a) 0011 b) 0101 c) 0001 d) 1001 aythadhi

the greatest -ve no which can be stored in a computer that has 8-bit word length & uses 2's c arithmetic is
2's c great no
a) -256 b) -255 c) -128 d) -127 → 1's c great no

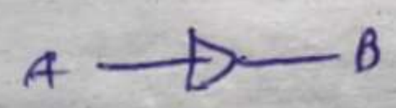
when two n-bit binary no's are added then sum bit contain at the most
a) n bits b) n+1 bits c) n+2 bits d) n+n bits

Boolean Algebra

*) the decimal no 13 can be represented in natural BCD code
a) 1101 b) 0001 0011 c) 0000 1101 d) none

→ To perform the basic operations on binary no's we need to follow some basic laws they are:
1) NOT LAW 2) OR LAW 3) AND LAW
4) Associative law 5) Commutative law 6) Distributive law 7) Demorgan's theorem

1) NOT LAW



A = 0 B = 1
A = 1 B = 0

2) OR LAW

$A + A = A$
$A + 0 = A$
$A + 1 = 1$
$A + \bar{A} = 1$

3) AND LAW

$A \cdot A = A$
$A \cdot 1 = A$
$A \cdot 0 = 0$
$1 \cdot 1 = 1$
$A \cdot \bar{A} = 0$

Commutative law

The commutative law allows the changes in the position of variables either in OR operation (or) in AND operation

$$A+B = B+A$$
$$A \cdot B = B \cdot A$$

Associative law

→ This law allows changes in grouping object variables

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

Distributive law

→ this law allows grouping of the variables individually

$$A(B+C) = AB+AC$$
$$A+(BC) = (A+B) \cdot (A+C)$$

Simply the following expression $F = A + \bar{A}B$

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

using Distributive law

$$= (A+\bar{A})(A+B) \quad [\because A+\bar{A}=1]$$
$$= 1 \cdot (A+B) = A+B$$

Simply the following expression $F = A + A\bar{B} + AB\bar{C} + ABC\bar{D}$

$$F = A + A\bar{B} + AB\bar{C} + ABC\bar{D}$$
$$= A(1+\bar{B}+B\bar{C}+BC\bar{D})$$
$$= A(1)$$
$$= A$$

Boolean expressions

→ there are 2 types of forms

1) Canonical form 2) standard form (Minimized form)

→ The canonical form represents all max possible combinations of any truth tables

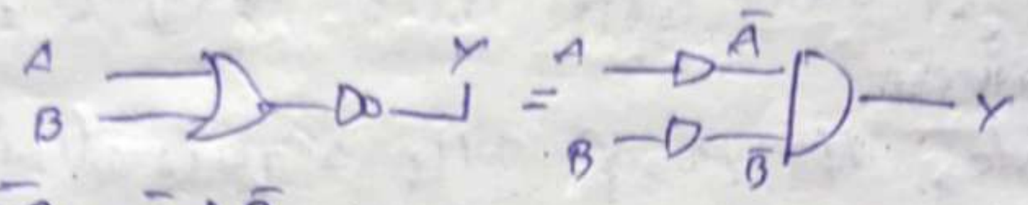
→ the standard form represents the minimised form canonical expression

Demorgan's law

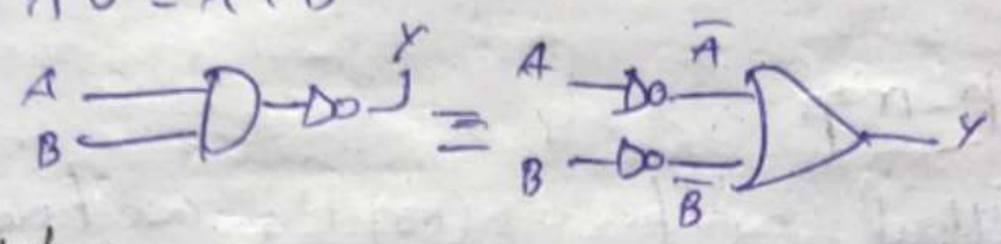
→ In this theorem we are having 2 types of converting

1) The complements of sum of variables = the product of their complements

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



2) $\overline{AB} = \bar{A}+\bar{B}$



Note:

→ In Demorgan's law is zero while converting every variable individually will be complemented OR & AND operations will be exchanged

→ zero & one will be exchanged

K-map

Min terms

→ this term, diff product terms of all possible combinations in which the logic o/p is "1"

Ex: $F(A,B) = \Sigma m(0,1)$

$2^2 = 4$

	0	1
0	1 0	1 1
1	2	3

Max terms

→ this represents the sum terms of all possible combinations where the logic o/p is "0"

Ex: $F(A,B) = \Pi M(0,2)$

	0	1
0	0 0	1
1	0 2	3

Note:

→ Min & Max terms are complement to each other

→ Graphical methods to minimize the logical expressions

→ Any canonical form can be reduced to minimized form (or) standard form by using two methods:-

1) By using boolean algebra laws

2) By using K-map.

→ The minimized or standard form is not unique for any logical expressions

→ For n no. of variables the no. of cells required in K-map = 2^n cells

→ <u>Types of grouping K-map</u>

→ Once the K-map is filled with 0's (or) 1's we have to identify the possible combinations of octate, Quad, pair

<u>pair</u>

→ If there are two 0's (or) 1's adjacent to each other either vertically (or) horizontally we can form a pair

→ A pair eliminates one variable & its complement

<u>Quad</u>

→ If there are four 1's (or) 0's adjacent to each other either vertically (or) horizontally we can form a quad

→ In this case two variables & its complements are eliminated

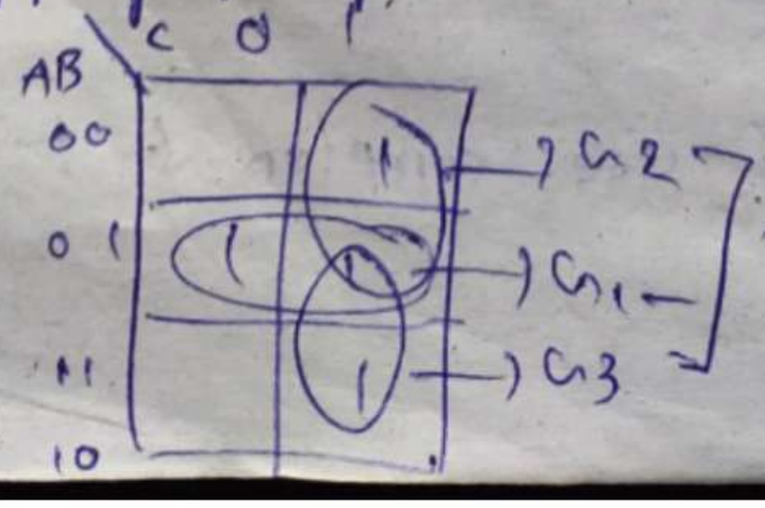
<u>Octate</u>:-

→ If there are eight 1's (or) 0's adjacent to each other we can form an octet. In this case we can eliminate three variables & its complements

<u>Overlapping of groups</u>

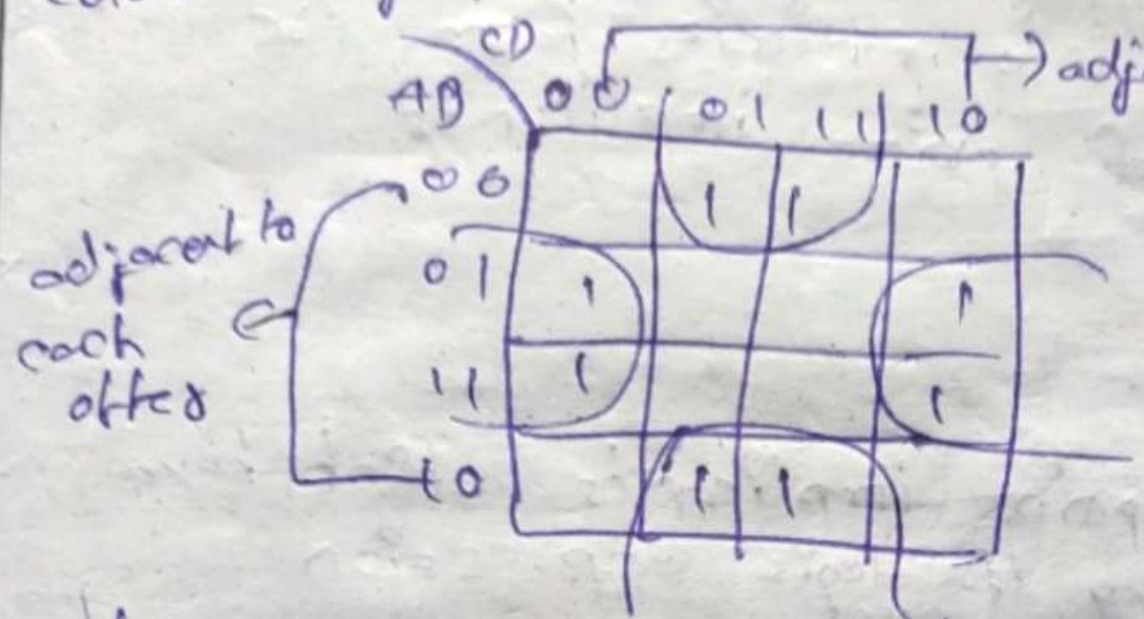
we can use logic 1 (or) 0 in more than one group of pairs, quads & octate. this type of grouping is called overlapping group.

Ex



Rolling:-

→ when we write K-map the 1st row & last Row are adjacent to each other similarly 1st Colum & last column adjacent to each other.



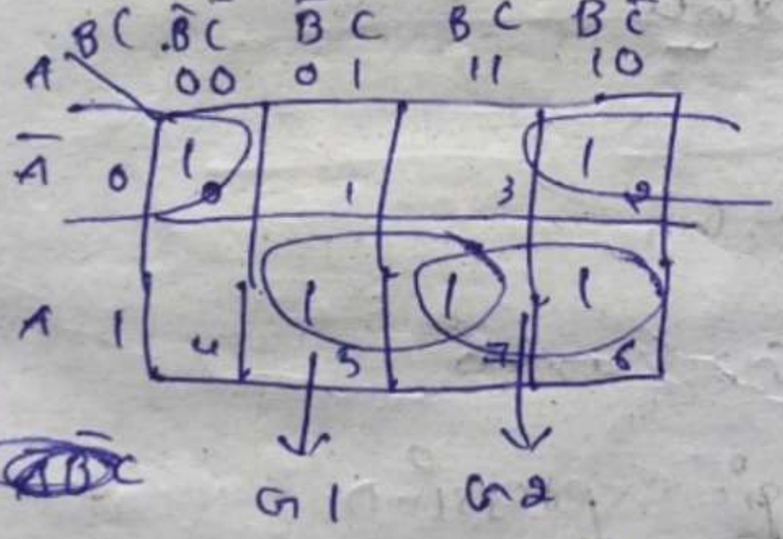
→ there are few terms which are essential when performing K-map

1) Implicant
2) prime Implicant
3) Essential prime Implicant
4) Non essential prime Implicant

Ex: SOP

$f(A,B,C) = \Sigma m(0,2,5,6,7)$

→ 3 = 2^3 = 8 cells



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

$G1 = AC$ $G2 = B\bar{C}$ $G3 = \bar{A}\bar{C}$

→ $G_1 G_3 G_4$ = EPI → G_2 → non e

→ $G_1 G_2 G_4$ = EPI → G_3 → non e

→ n is possible combination of ~~perfect~~ variables present in a logic expression are considered as implicants

Ex: (0, 2, 5, 6, 7)

→ All possible groups in kmap of a logical expression is called prime implicants.

Ex: G_1, G_2, G_3, G_4

→ Among these prime implicants which are absolutely necessary to reduce into minimize form are considered as essential prime implicants.

Ex: G_1, G_3, G_4 are
G_1, G_2, G_4

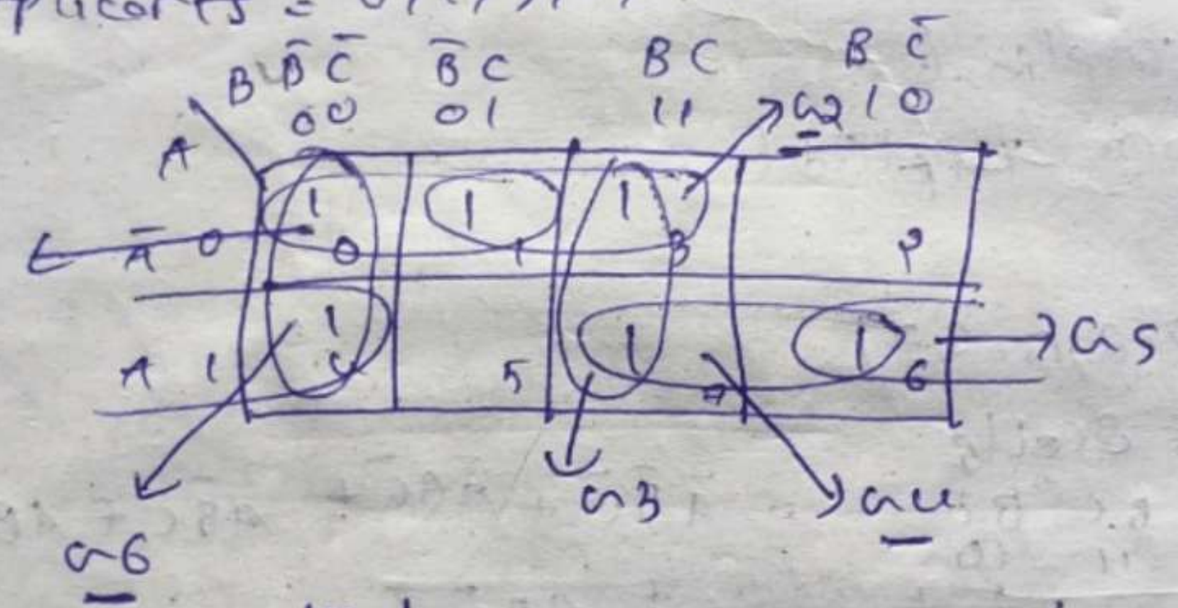
Nor

→ We can reglect few prime implicants which have no role in minimizing the logic expression

Ex: G_2 (or) G_3

$F(A, B, C) = \Sigma m(0, 1, 3, 4, 6, 7)$

Implicants = 0, 1, 3, 4, 6, 7



prime Implicants = G_1 to G_6

Essential Implicants = G_1, G_3, G_5,

Non-Essential Implicants = G_2, G_4, G_6

Minimize the following logic expression

1) $F = A + AB + BC$

$= A + AB + BC$

$A(1+B) + BC$

by using or law

$[1+B = 1]$

$A.1 + BC$

$A + BC$

2) $F = A + AB + AC$

$= \cancel{A(1+B) + A}$

$= A + AB + AC$

$= A(1+B+C)$

by using OR law

$= A.1 = A$

3) $F = A + \bar{A}B$

$= A + \bar{A}B$

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

$= 1.(B+A)$

$= A+B$

) expression

absolutely
ase

t have
n

4) $F = xy + x\bar{y} + \bar{x}y$

by using OR law

$[A + A = A]$

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

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

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

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

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

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

$= x\bar{x} + xy$

$= \cancel{x(x}$

$= 0 + xy = xy$

To convert minimised form to canonical form we have to consider that every possible combination of min term / max term

find canonical form of the following minimised form

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

$= \bar{x}y.1 + z.1.1$

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

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

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

$= 011 + 010 + 111 + 101 + 011 + 001$

$= 3 + 2 + 7 + 5 + 3 + 1$

$= \Sigma m(1, 2, 3, 5, 7)$

Simply the following expression

$F(A, B, C) = \Sigma m(0, 3, 6, 7)$

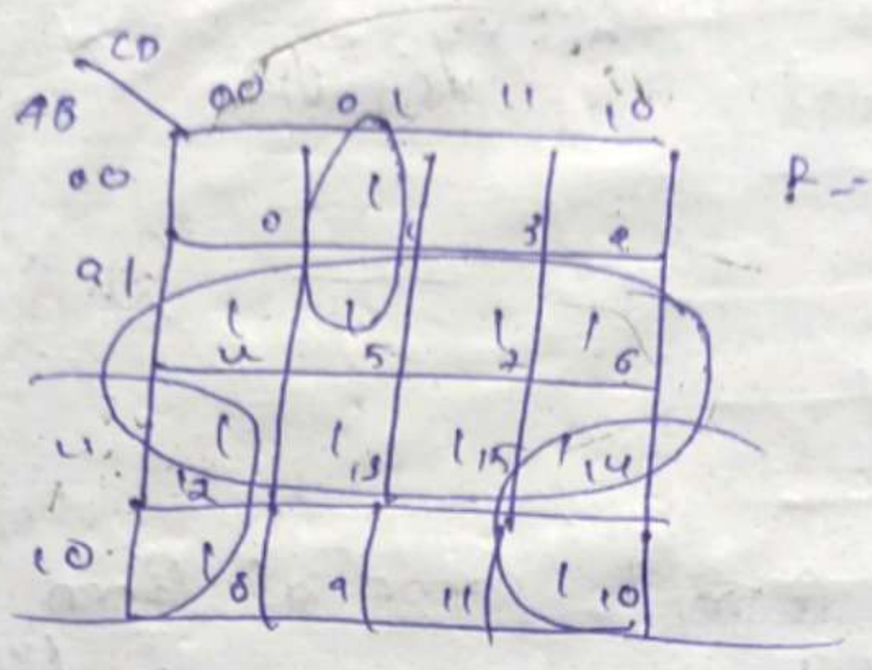
$3 \rightarrow 2^3 = 8$ cells

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

$= \cancel{\bar{A}\bar{B}\bar{C}} + BC + AB$

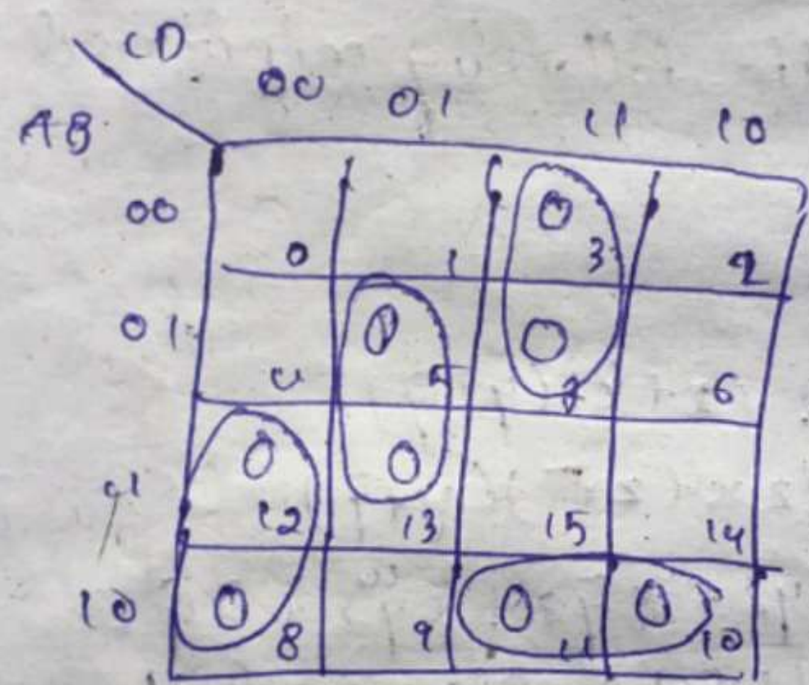
$F(A,B,C,D) = \Sigma m(3,4,5,6,7,8,9,10,11,12,13,14,15)$

$2^4 = 16$



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

or $F(A,B,C,D) = \Pi m(3,5,7,8,10,11,12,13)$



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

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

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

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

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

-) the above sop form can be update by filling the cells with logic 1 in the above problem

-) Hence, we can conclude that sum of product form (SOP) & POS form are complement to each other

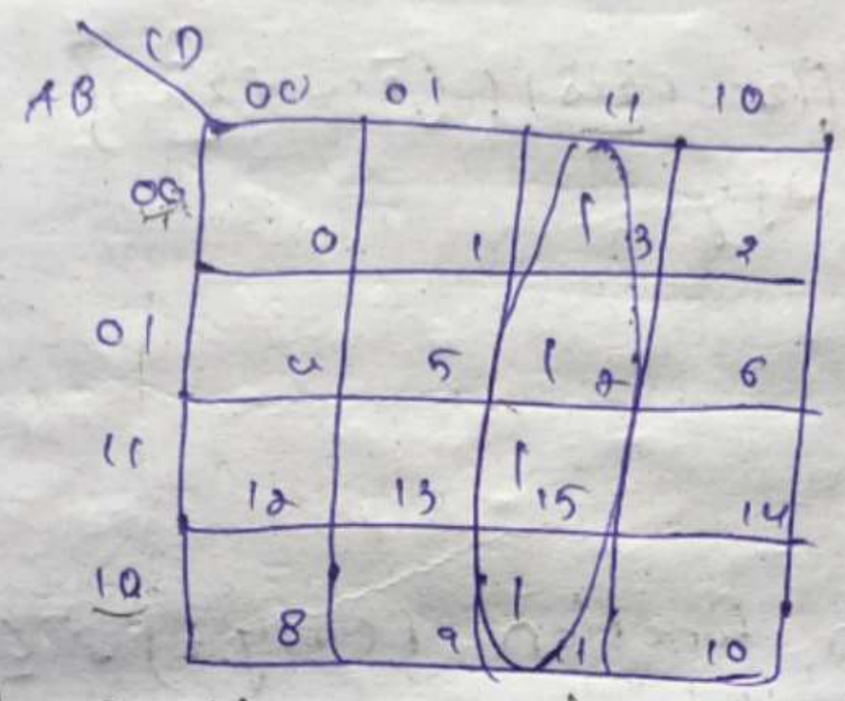
simply following boolean expression

$x(A,B,C,D) = ABCD + \bar{A}\bar{B}CD + \cancel{\bar{A}C\bar{B}D}\ \bar{A}BCD + \bar{A}C\bar{B}D$

a) $\bar{C}D$ b) CD c) $\bar{A}BD + ACD$

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

1111 (15) 1011 (11) 0111 (7) 0011 (3)



$x(A,B,C,D) = CD$

1) which of the following is not considered for forming groups in k-map

1) Rolling G 2) vertical G 3) Diagonal G 4) Horizontal G

2) POS expression can be implemented by using

a) two level OR-AND logic circuits

b) two level AND-OR " "

c) both a & b

d) none

3) SOP can be implemented by using 2 level AND-OR logic circuits

3) which of the following method is used to minimize the boolean expression

a) fourier Transform b) Gray code c) K-map

4) If small n denotes the no. of variables present in a boolean exp then the no. of cells present in K-map

a) $2+n$ b) $2n$ c) n d) 2^n

1) the octal equivalent of the Hexa no (AB.CD) is

a) 258.34 b) 253.632 c) 576.623 d) 526.632

$(AB.CD)_{16}$ to $(\ )_8$

→ $(10101011.1100110)_2$

010 101 011. 110. 011 010

$(253.632)_8$

2) the following.

$[(x+\bar{y}y)(x+\bar{x}y)][(zw+\bar{z}\bar{w})+(\bar{z}w+z\bar{w})]$

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

$(xx+x\bar{x}y)+(0)$

$[(x)]$

3) for the given boolean function $F(x,y,z) = \Sigma(0,2,4,5,6)$ simplified output will be

a) $F = z+x\bar{y}$ b) $F = z+\bar{x}y$ c) $F = \bar{z}+\bar{x}y$ d) $F = \bar{z}+x\bar{y}$

x \ yz	00	01	11	10
0	1 (0)	1 (1)	3	1 (2)
1	1 (4)	1 (5)	7	1 (6)

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

Logic Gates

Types of Gates

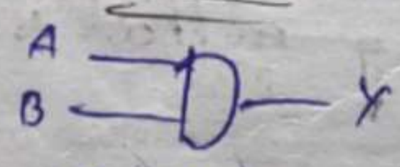
OR Gate :-

Truth table

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

$Y = A+B$

AND GATE

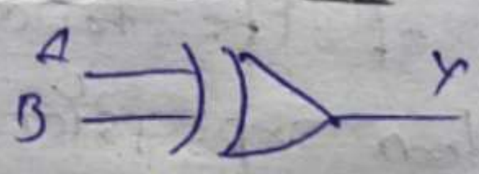


Truth table

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

$Y = AB$

Ex-OR GATE



Truth table

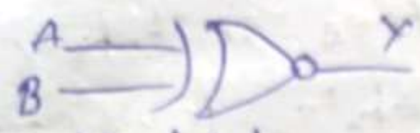
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

$Y = A \oplus B$

3. (D) is

6. 639

Ex-NOR GATE

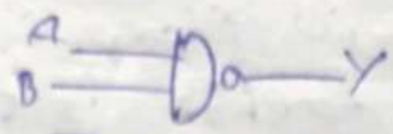


Truth table

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

$Y = A \odot B$

NAND GATE

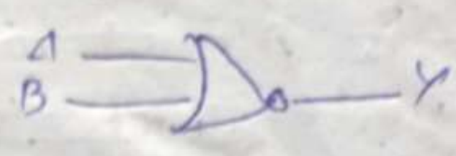


T.T

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

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

NOR GATE



T.T

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

$Y = \overline{A+B}$

Note:

→ NAND & NOR gates are called universal gates since these two gates can be used to represent any logic expression.

→ 1) An all or nothing gate is often more for

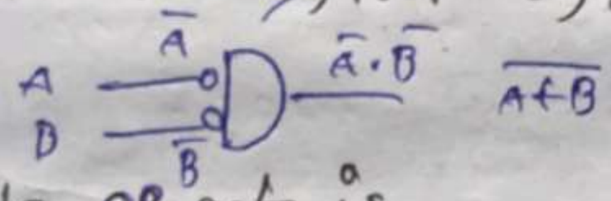
a) OR gate ~~b)~~ AND gate c) X-OR gate d) X-NOR gate

2) An any (or) all gate is often more for

a) AND ~~b)~~ OR c) X-OR d) X-NOR

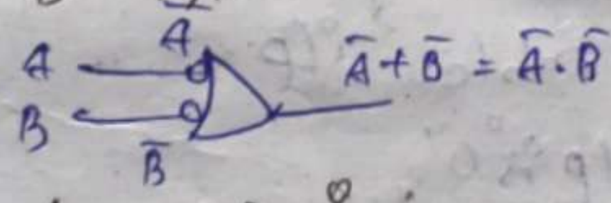
3) A bubble AND gate is

a) OR b) AND ~~c)~~ NOR d) X-OR



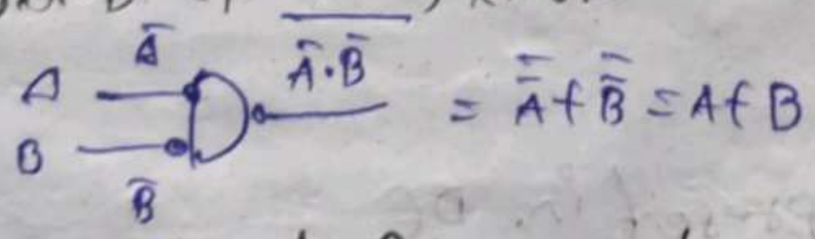
4) A bubble OR gate is

a) OR ~~b)~~ AND c) NOR d) X-OR



5) A bubbled NAND is

~~a)~~ OR b) AND c) NOR d) X-OR



6) The o/p of a gate is low when atleast one of its i/p is high this is true for

a) AND b) NAND ~~c)~~ NOR d) OR

7) The most suitable gate for comparing a bits is

a) AND b) OR c) NAND ~~d)~~ X-OR

X-OR → inequality comparison

X-NOR → Equality "

8) A light is connected to both the switches where one switch is placed in ground floor & another switch is placed which logic function is suitable inorder to perform the operation where the light glows
a) OR b) X-OR c) NAND d) NOR

a) the no. of NAND, NOR gates required to implement diff logic gates

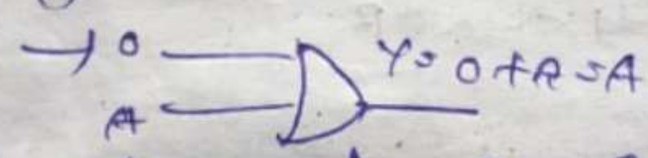
~~AND~~ b)

	NOT	AND	OR	EX-OR	EX-NOR
NAND	1	2	3	4	5
NOR	1	3	2	5	4

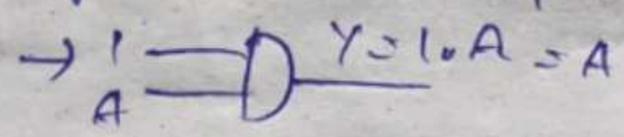
Enable & Disable i/p

→ Enable i/p is the one which is making the o/p of a logic gate is independent of that particular AND & depending upon only on other variable i/p

→ For OR gate enable i/p is "0"



→ for AND gate enable i/p is "1"



→ Disable i/p

→ It is the one which is making o/p of logic gate independent of other variable i/p

→ for AND gate disable i/p is "0"

→ for OR gate disable i/p is "1"

~~Combinational circuits~~

Types of circuits present in DE

1) Combinational circuits 2) Sequential circuits

1) Combinational circuits

→ this circuit depends only on the present i/p's

Note:
→ there is no memory element present in combinational circuits

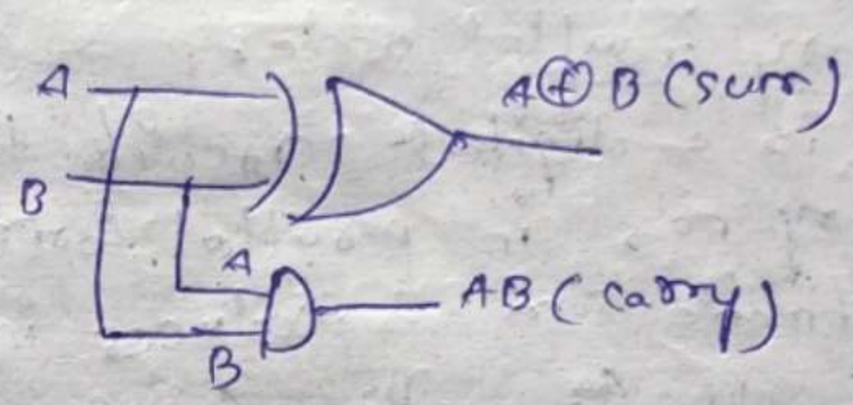
types
1) Adder → Half & full adder, subtractor - HS, FS
2) Encoder & Decoder
3) Multiplexer & De-Multiplexer
4) Comparator

1) Adders

Note: Half adder is used for addition of 2 bits only

1) Half adder

A	B	sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

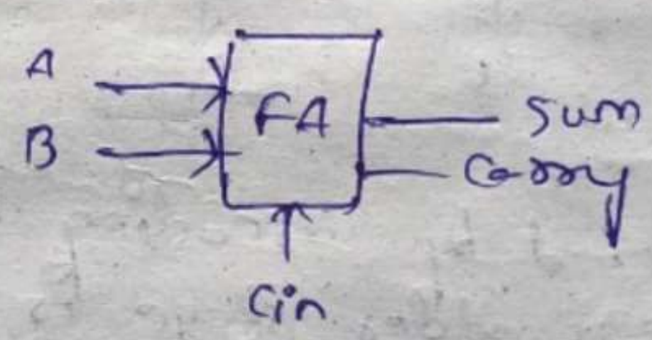


$sum = \bar{A}B + A\bar{B} = A \oplus B$

$carry = AB$

2) Full adder

	A	B	Cin	sum	carry
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1



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

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

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

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

$A, B, C_{in} \rightarrow 3 \rightarrow 2^3 = 8$ cells

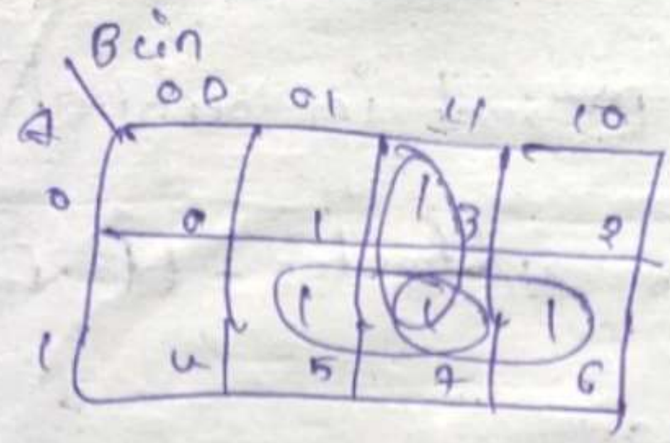
sum =

A \ B Cin	00	01	11	10
0	0	1 (1)	3	1 (2)
1	1 (4)	5	1 (7)	6

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

$Sum = A \oplus B \oplus C_{in}$

Carry =



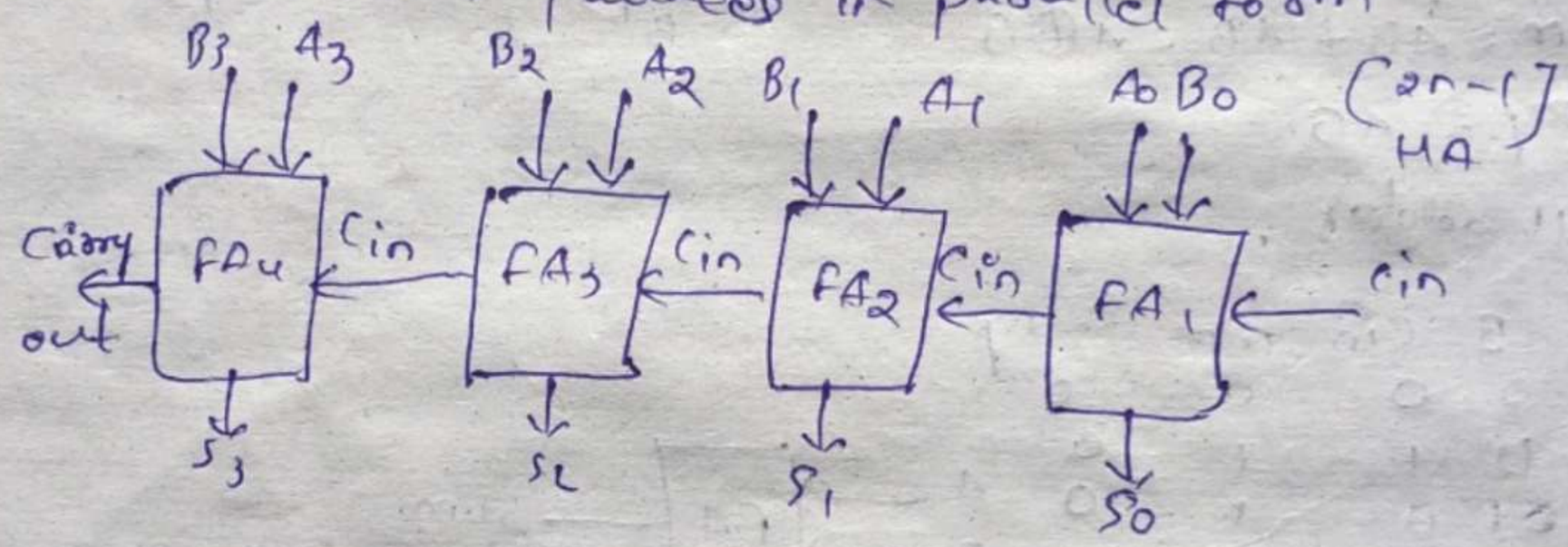
Carry = ACin + AB + BCin

Types of Adders

There are 3 types of adder
1) serial adder 2) parallel adder 3) Carry look ahead adder

Binary parallel adder

→ It is a digital circuit that is used to add binary no's in parallel form & produces the arithmetic sum & produces in parallel form



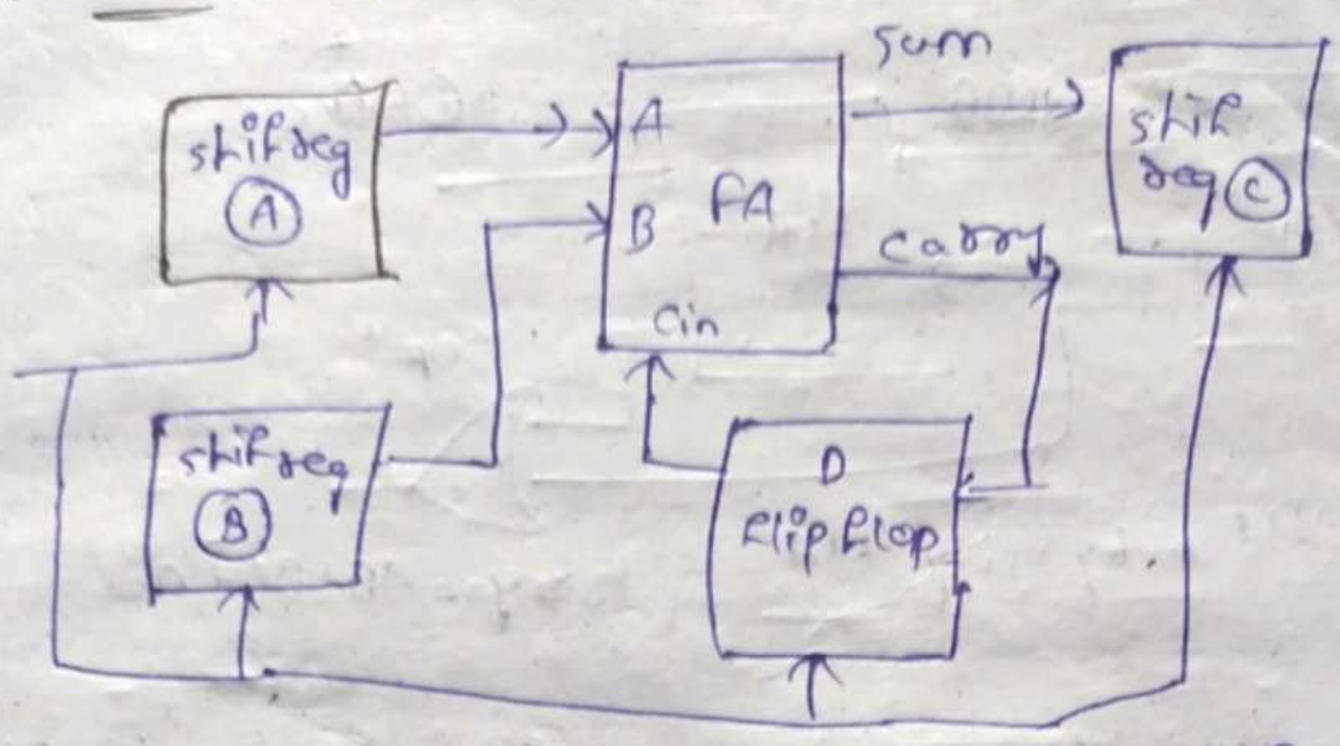
→ It is constructed by cascading full adders one after another each full adder stage is responsible for the addition of two binary digits & also in carry from previous stage

→ The carry out of one stage is fed directly to carry in of next stage

$A_3 A_2 A_1 A_0$
0011 = A
1011 = B
4 bit addition

→ To implement n-bit binary parallel adder
1) we can use n no. of full adders (or)
2) we can use n-1 full adders & 1 Half adder (or)
3) we can use 2n-1 Half adders & n-1 OR gate

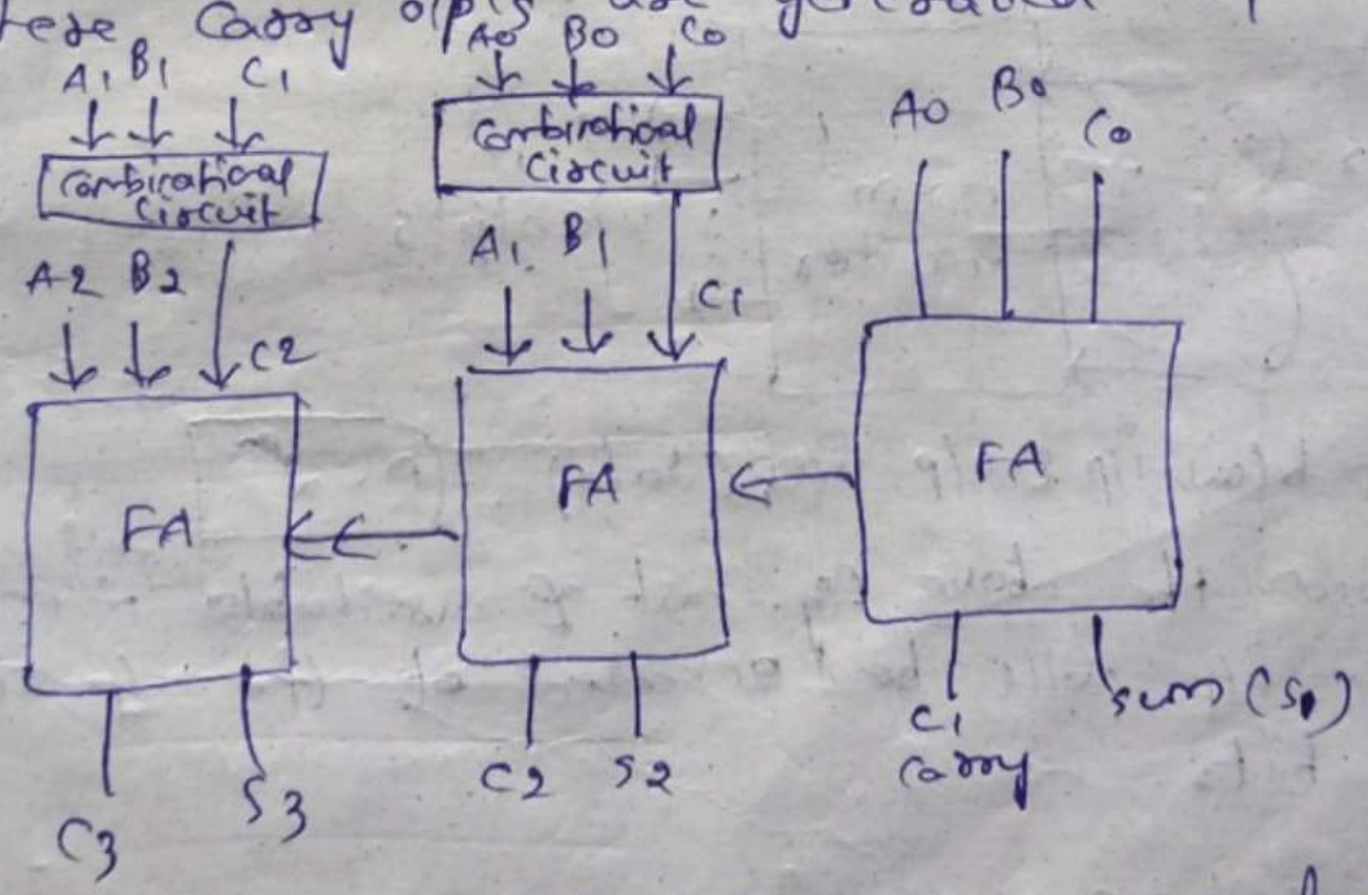
Serial adder (seq. circuit)



→ for n bit serial adder we use only one FA
→ we are using a delay flip-flop inorder to carry the o/p from full adder (carry) to next bit
→ It is not a combinational circuit since we are using a flip-flops.

Carry look ahead adder

→ In Binary parallel adder the carry from each bit addition should be forwarded to the next positions which is creating some delay.
→ To overcome this problem, we are using carry look ahead adder where carry o/p's are generated separately



→ By using carry look ahead adder we can reduce the total propogation delay of the circuit
→ Among this all adders carry look ahead adder is fastest

Note : PQ

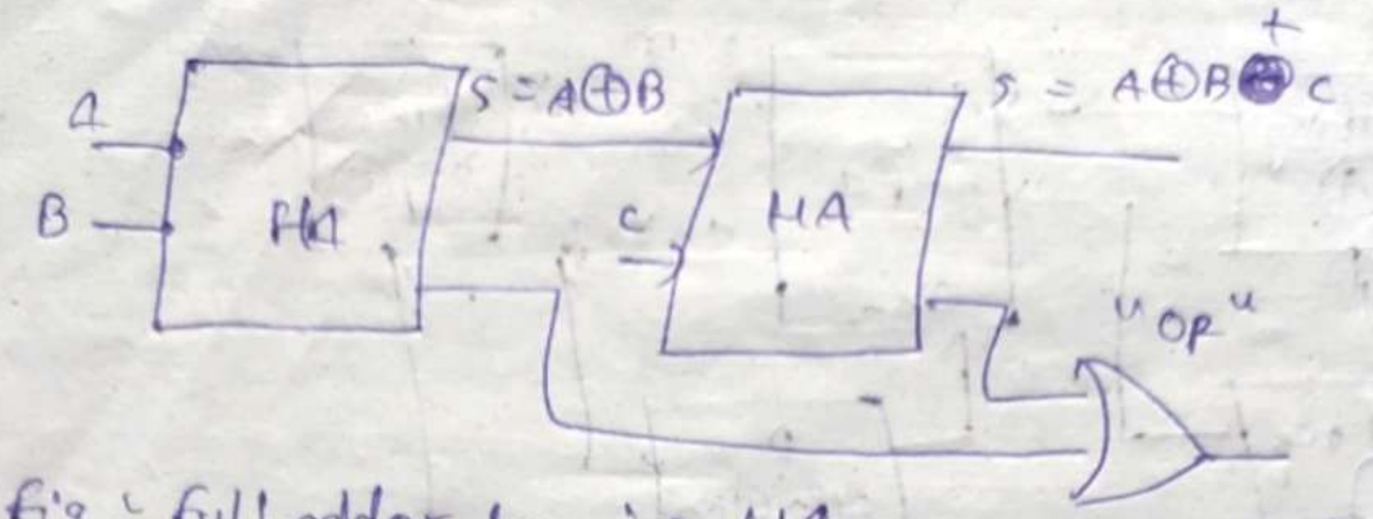


fig : full adder by using HA

Carry = AB+BC+AC

Encoder :

→ It is a Combinational circuit which is having multiple i/p's & multiple o/p's

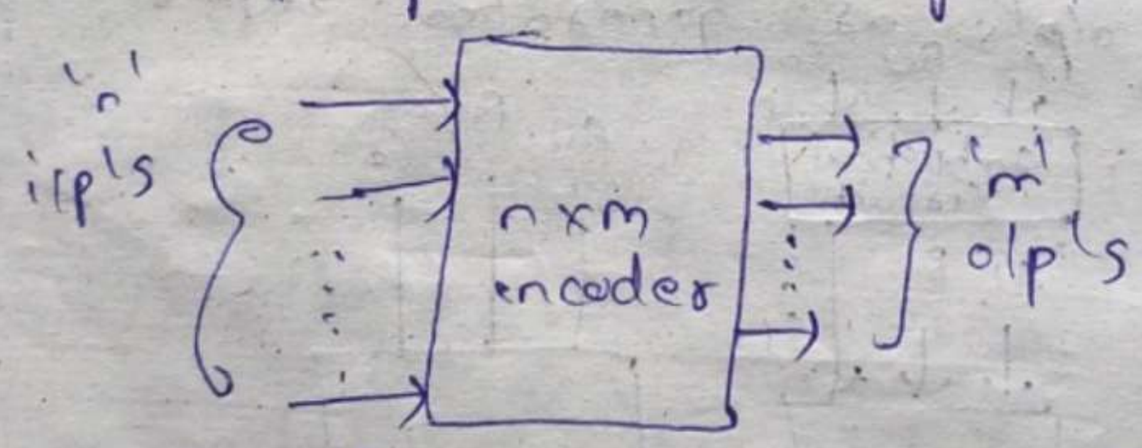
→ Generally, An Encoder is used to assign different Codings to each data present at the i/p.

Ex : A Can be represent in 1010, B → 1011

→ It is mainly used for data reduction while sending the signal from the transmitting end to receiving end which of the following is used to reduce the no. of data lines required for transmission of occurrence of events

a) adder b) decoder c) encoder d) 7 segment decoder

→ It is a active code converter which is used to convert a i/p signal to a coding o/p signal.

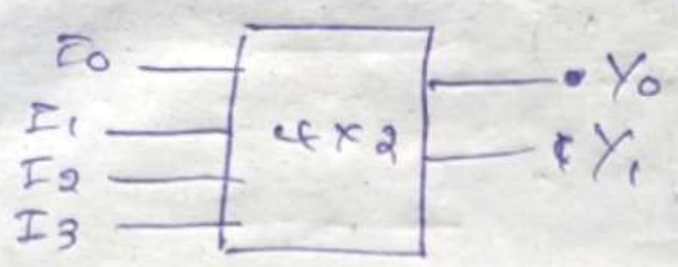


→ Relation b/w i/p & o/p lines (data) :- $\boxed{n = 2^m}$

⇒ As shown the above fig. out of available small 'n' no. of i/p's only one i/p will be encoded at the o/p side in m no. of bits.

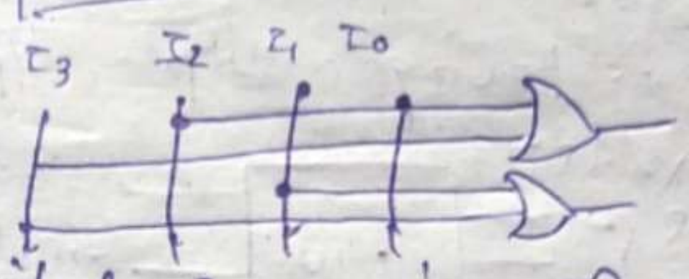
1) Decimal to BCD encoder
2) octal to binary encoder
3) Hexadecimal to binary encoder

4×2 encoder



I_3	I_2	I_1	I_0	Y_1	Y_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

$$\begin{cases} Y_1 = I_2 + I_3 \\ Y_0 = I_1 + I_3 \end{cases}$$

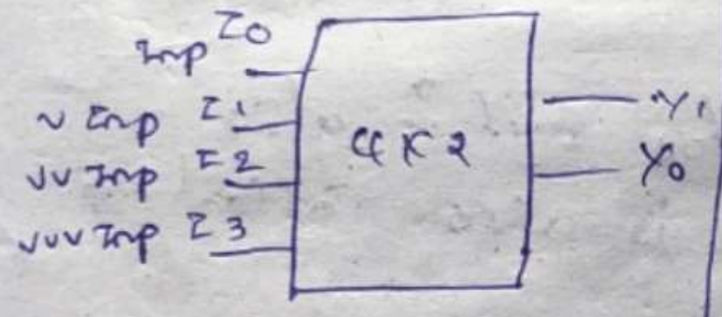


→ Data encoding is digital conversion of data into digital signals i.e., 0's & 1's

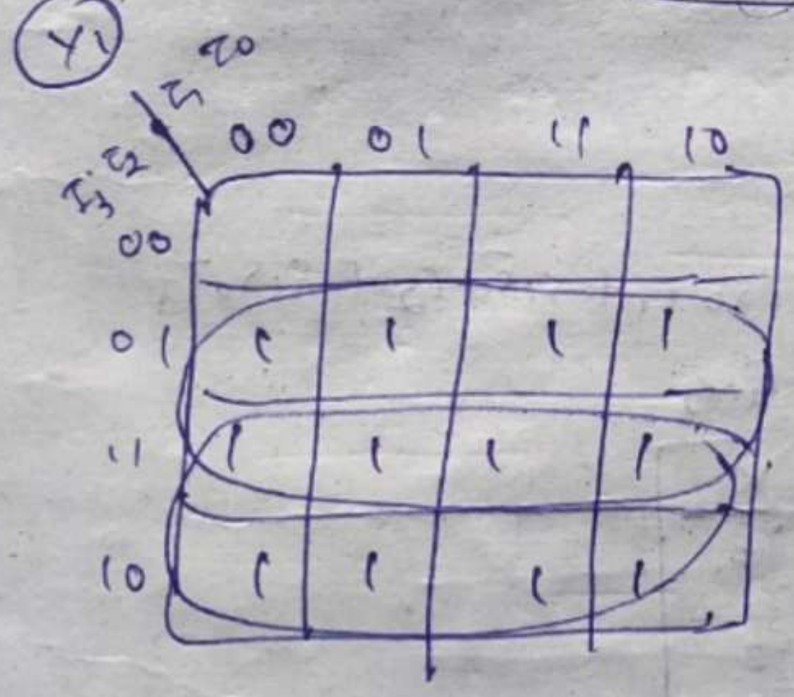
priority encoder

→ As the encoder contains multiple i/p's we can assign diff. priorities to diff. i/p's

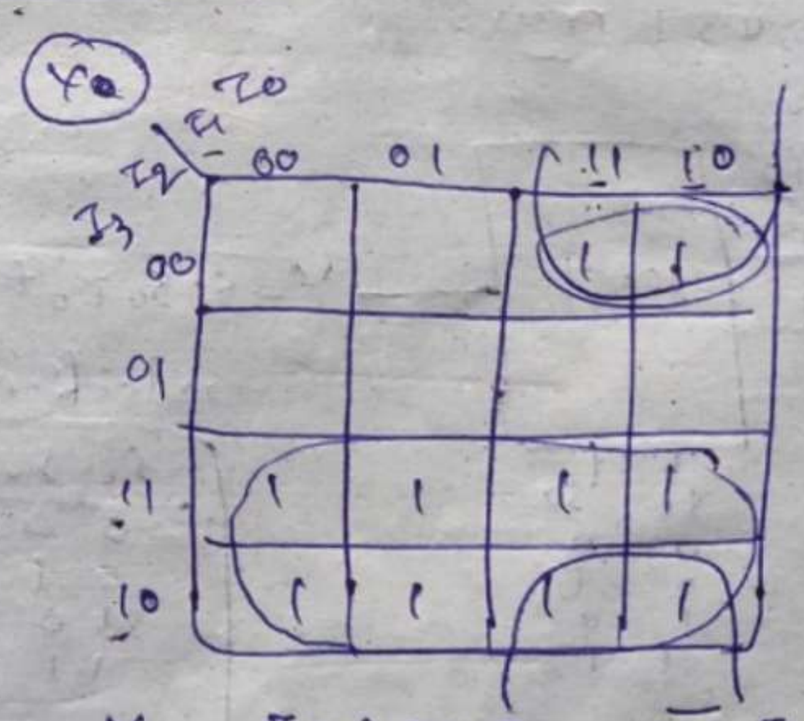
→ Based on these priority levels the o/p is available.



I_3	I_2	I_1	I_0	Y_1	Y_0
0	0	0	0	x	x
0	0	0	1	0	0
0	0	1	x	0	1
0	1	x	x	1	0
1	x	x	x	1	1



$Y_1 = I_2 + I_3$



$Y_0 = I_3 + I_1 \bar{I}_2$

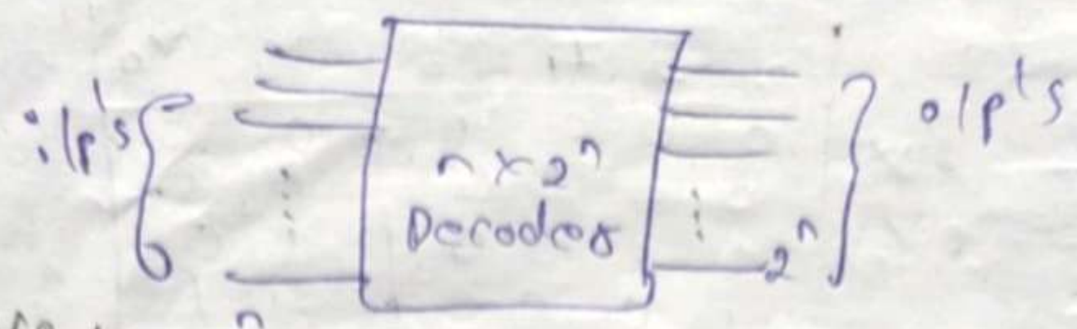
Decoder

→ It is a combinational circuit which has multiple i/p's & multiple o/p's

→ It is used to convert the binary data to other codes.

→ B → Hex → B → Hexadecimal

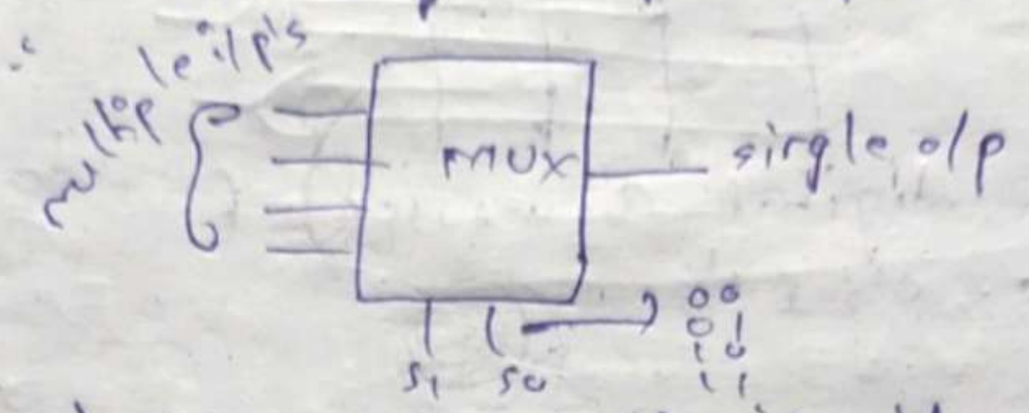
→ BCD → Decimal → BCD → 7 segment display



Multiplexer

→ which is having multiple i/p's & only one o/p.

→ Eg:



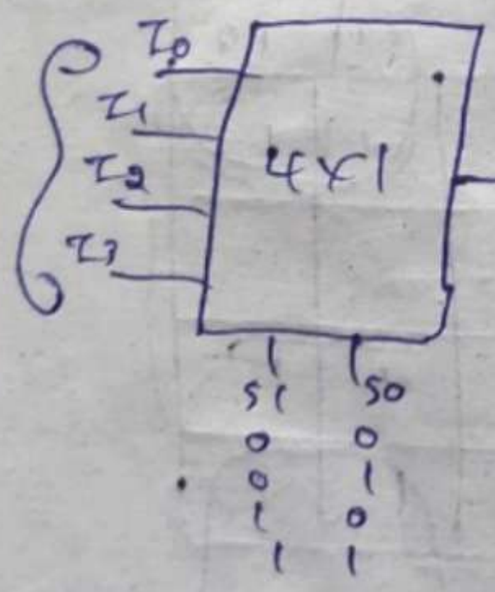
→ The available data at the i/p side is transmitted to the o/p side based on Selection lines

→ The no. of selection lines will be depending upon the no. of i/p lines

→ A multiplexer is also called as parallel to serial data converter.

→ This also called as data selector & it is also called many to one circuit. It is also called as universal circuit

Ex: 4×1 mux



$Y = \bar{S_1}\bar{S_0}I_0 + \bar{S_1}S_0I_1 + S_1\bar{S_0}I_2 + S_1S_0I_3$

selection lines

S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

→ The minimum no. of selection i/p's required to implement 22×1 mux

22×1

no. of i/p's = 22

no. of o/p's = 1

no. of i/ps ≤ = 2^x (selection lines)

$22 \le 2^x$

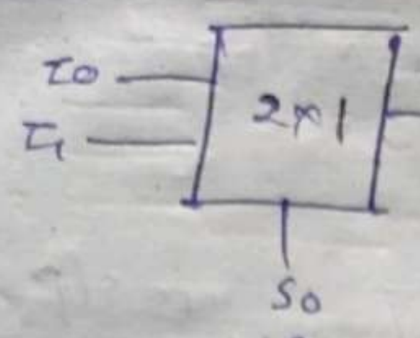
$x = 5$

$22 \le 2^5$

$22 \le 32$

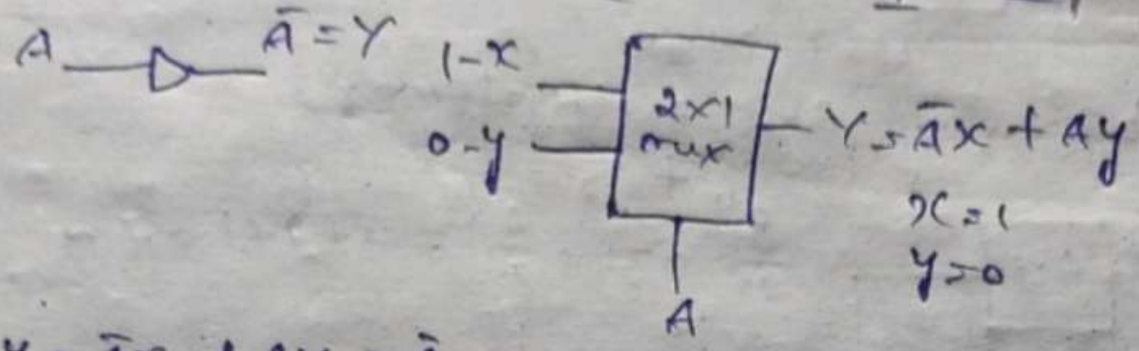
5 selection lines

2x1 mux



$Y = \bar{S_0} I_0 + S_0 I_1$

NOT gate implementation by using 2x1 mux



$Y = \bar{A}x + Ay$

$x = 1$

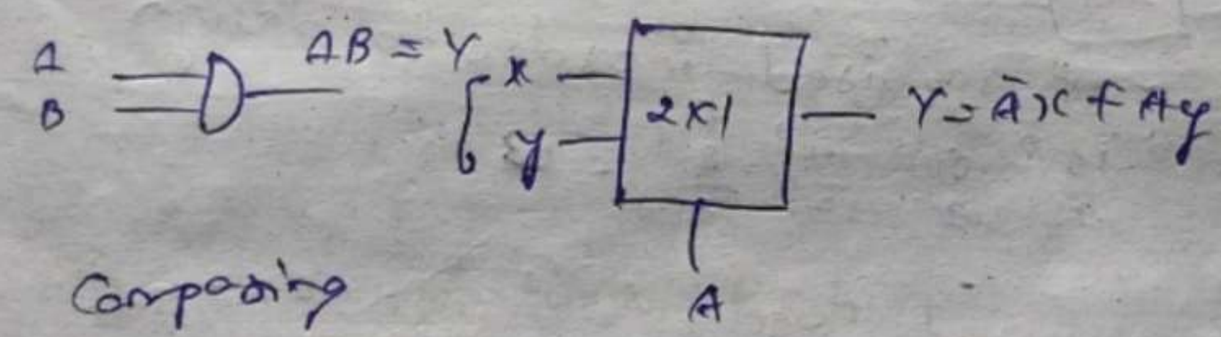
$y = 0$

$\rightarrow y = \bar{A}x + Ay = \bar{A}$

By comparing LHS = RHS

$x = 1 \quad y = 0$

AND gate



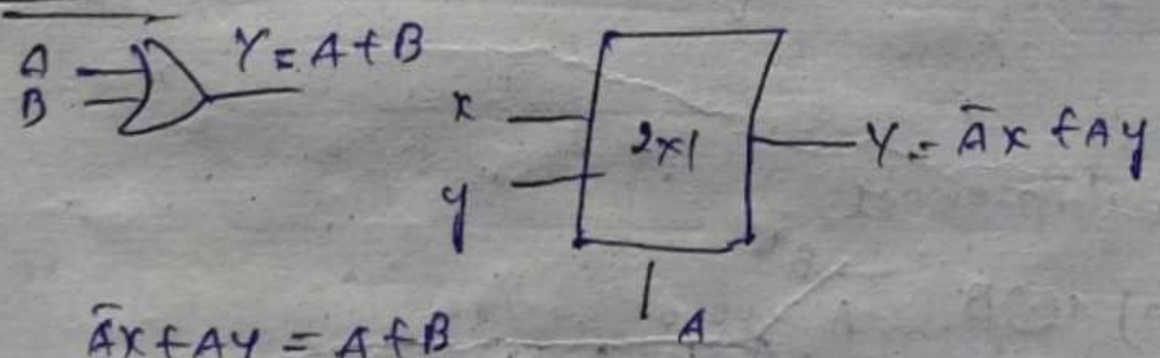
$Y = \bar{A}x + Ay$

Comparing

$\bar{A}x + Ay = AB$

$x = 0 \quad y = B$

OR Gate



$Y = \bar{A}x + Ay$

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

$x = B \quad y = 1$

$\bar{A}B + A$

$A + \bar{A}B$

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

$1 . A + B = A + B$

Ex

NAND

NOR

X-OR

X-NOR

Implementation of Higher order multiplexers by using lower order multiplexers

1) Design 4×1 multiplexer by using 2×1 multiplexer

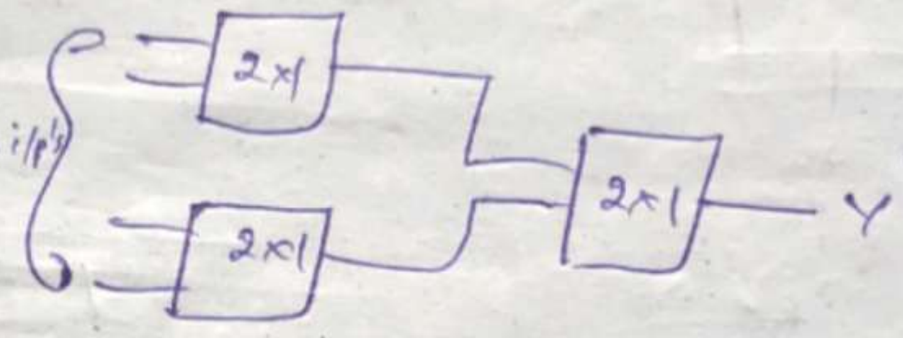
4×1

no. of i/p's = 4

no. of o/p's = 1

shortcut

$\frac{4}{2} + \frac{2}{2} + 1 = 2+1 = 3$



4×1 MUX

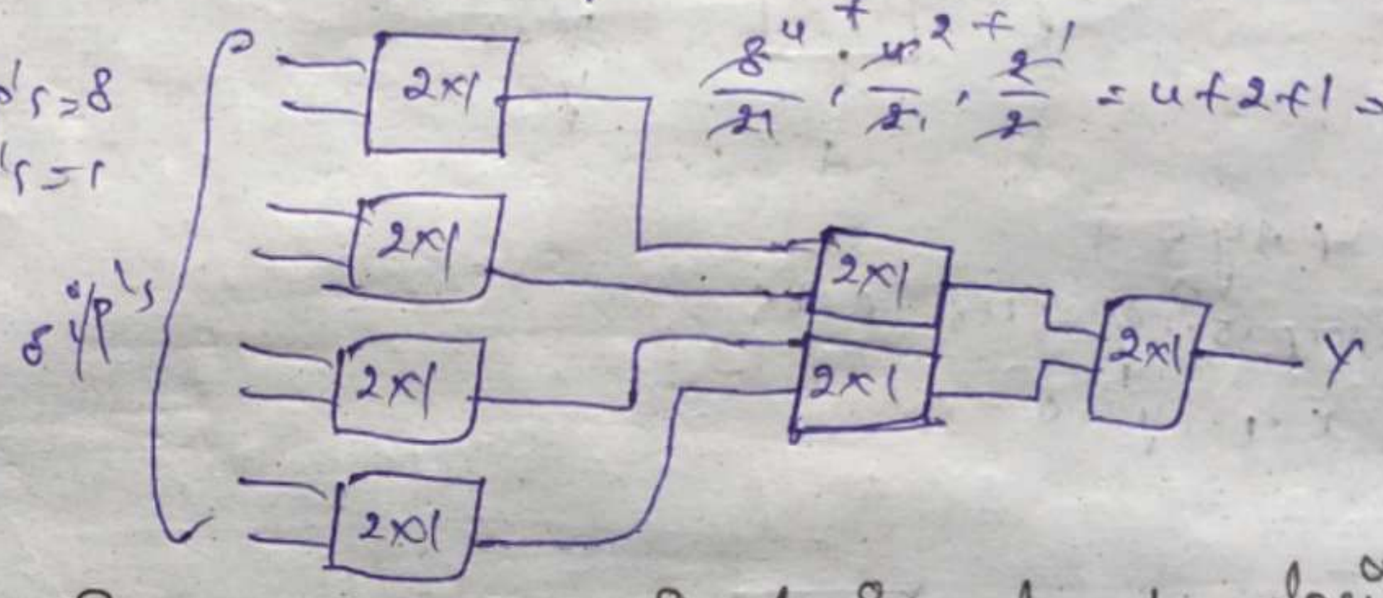
2) Design 8×1 mux by using 2×1 multiplexer

8×1

no. of i/p's = 8

no. of o/p's = 1

$\frac{8}{2}^{4} + \frac{4}{2}^{2} + \frac{2}{2}^{1} = 4+2+1 = 7$



→ find the no. of 2×1 mux required in order to design 64×1 mux

$\frac{64}{2}^{32} + \frac{32}{2}^{16} + \frac{16}{2}^{8} + \frac{8}{2}^{4} + \frac{4}{2}^{2} + \frac{2}{2}^{1} = 63$

→ the no. of 4×1 mux required in order to design 64×1

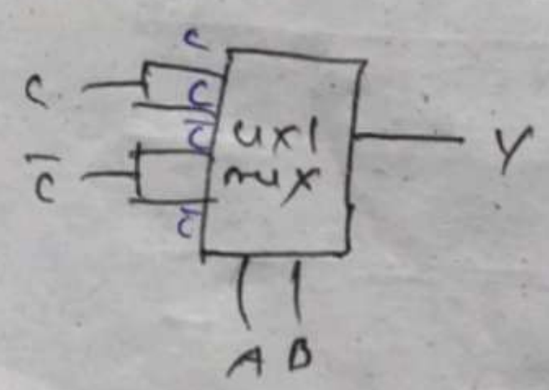
$\frac{64}{4}^{16} + \frac{16}{4}^{4} + \frac{4}{4}^{1} = 21$ ✓

Note

→ For $2^n \times 1$ the no. of 2×1 mux required $2^n - 1$ mux

→ the no. of

Solve the following multiplexers

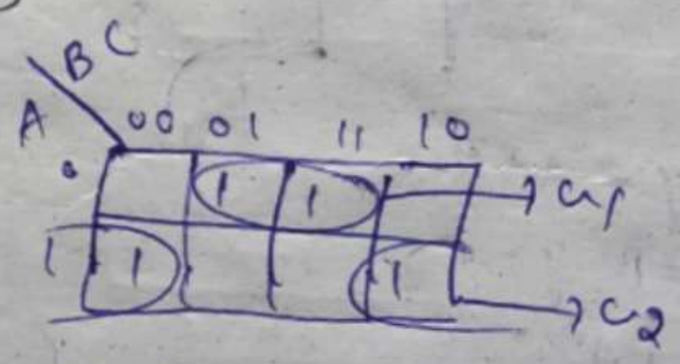


a) $A \oplus B$

b) $A \odot B$

c) $A \oplus C$

d) $A \odot C$



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

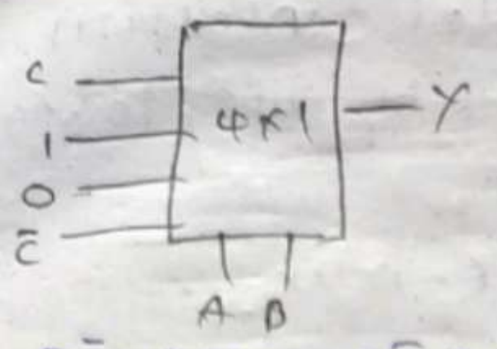
001 011 100 110

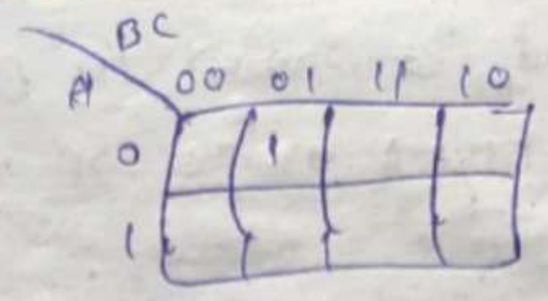
$Y \Rightarrow C_1 + C_2$

$\Rightarrow \bar{A}C + A\bar{C}$

$= \bar{A} \oplus C$

4)



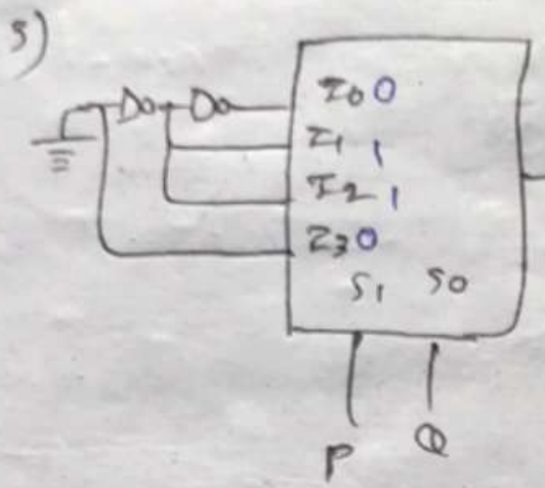


$Y = \bar{A}\bar{B}C + \bar{A}B\cdot 1 + A\bar{B}\cdot 0 + AB\bar{C}$

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

001 01 110

5)



a) Y = AND (P, Q) b) Y = X-OR (P, Q)

$Y = \bar{P}\bar{Q}\cdot 0 + \bar{P}Q\cdot 1 + P\bar{Q}\cdot 1 + PQ\cdot 0$

$= \bar{P}Q + P\bar{Q}$

$= P \oplus Q$

Ans = $P \oplus Q$

6) A decoder with enable i/p can function as

a) X-OR b) MUX c) De-mux d) Encoder

7) A fulladder is characterised

a) 2 i/p's, 2 o/p's b) 2 i/p's, 3 o/p's 3) 2 i/p's, 1 o/p

d) 3 i/p's, 2 o/p

6) Encoder converts [ ]

a) n i/p lines into 2^n o/p lines b)

c) 2^n i/p lines into n o/p line d) n i/p lines, n o/p lines

7) An equivalent

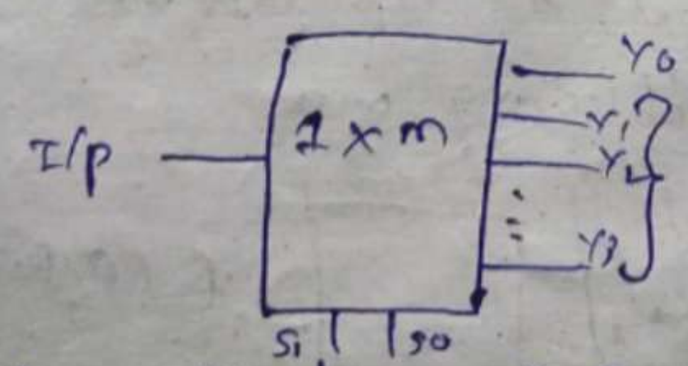
a) b) De-mux c)

8)

DE-MUX

→ this is a combinational circuit which is having only 1 i/p & multiple o/p's

→ the data available at the i/p side is transfered two any one of the o/p's lines.



$Y_0 = \bar{S}_1\bar{S}_0 I$

$Y_1 = \bar{S}_1 S_0 I$

$Y_2 = S_1\bar{S}_0 I$

$Y_3 = S_1 S_0 I$

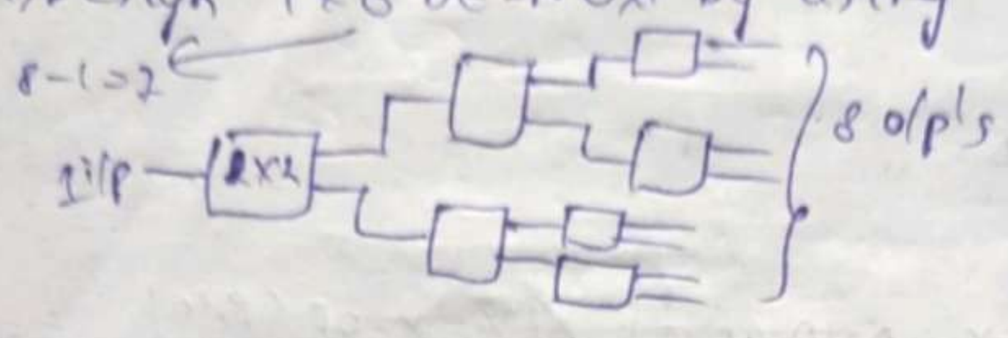
→ It is also called as one to many circuit

→ It is also called as data distributor.

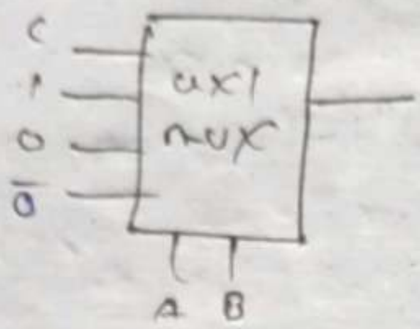
→ It can be used for serial to parallel conversion

Implementation of higher order De-mux by using lower order De-mux

1) Design 1x8 De-mux by using 1x2 De-mux



Q)



$Y = \bar{A}\bar{B}C + \bar{A}B\cdot 1 + A\bar{B}\cdot 0 + AB\bar{0}$

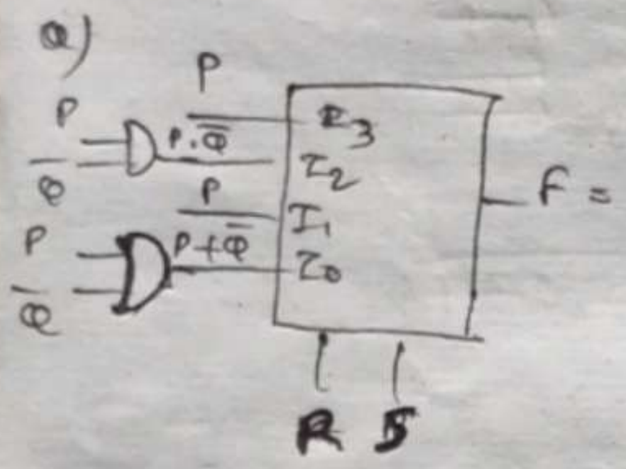
$= \bar{A}\bar{B}C + \bar{A}B + AB$

001 01 11

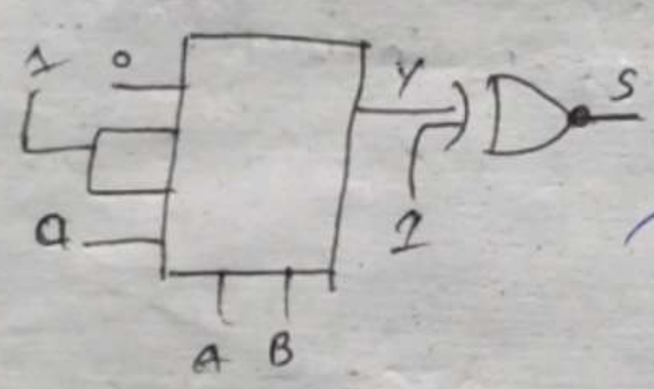
$= \bar{A}\bar{B}C + B(A+\bar{A})$

$= \bar{A}\bar{B}C + B\cdot 1$

Q)



Q)



a) $A \odot B$
b) AB
c) $A \oplus B$
d) $A+B$

$Y = A \oplus B$, $S = 1 \odot Y$ (0→Y, 1→$\bar{Y}$)

XNOR
00 — 1
01 — 0
10 — 0 — Y
11 — 1 — $\bar{Y}$

Note :

1) $A \oplus A = 0$
2) $1 \oplus A = \bar{A}$
3) $0 \oplus A = A$
4) $A \oplus \bar{A} = 1$

X-OR T.T

00 0
01 1
10 1
11 0

X-NOR

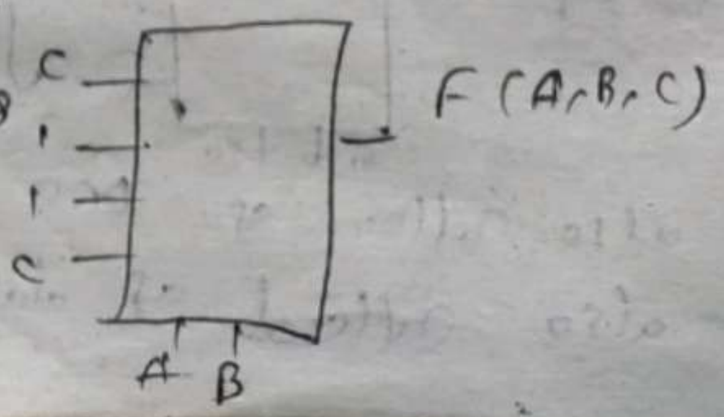
1) $A \odot A = 1$
2) $1 \odot A = A$
3) $0 \odot A = \bar{A}$
4) $A \odot \bar{A} = 0$

problems based on mux to find the i/p combination of mux

1) Find the i/p combinations of following mux

$F = (A, B, C) = \Sigma m(1, 2, 3, 5, 6, 7)$

A, B are selection lines



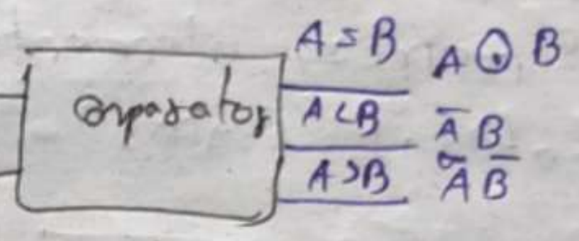
Note: we are not forming group

AB \ C	0	1
00	0	1 → C
01	1	1 → $1 = 0 + \bar{C}$
11	1	1 → $1 = 0 + \bar{C}$
10	0	1 → C

Case (ii) ∴ AC Case (iii) BC

2-Bit Comparator

A	B	A=B	A<B	A>B
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	1	0	0
		=XNOR	$\bar{A}B$	$A\bar{B}$



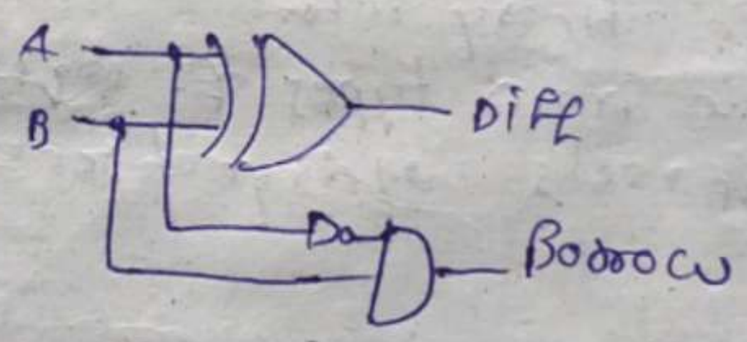
Note:
→ X-NOR logic gate is also called as Coincidence

~~Half subtractor~~

→ X-NOR can be used as equality comparison
→ X-NOR gate is used for inequality comparison

Half subtractor:

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



Diff = $A \oplus B$

Borrow = $\bar{A}B$

Full subtractor

	A	B	C	D	B
0	0	0	0	0	0
1	0	0	1	1	1
2	0	1	0	1	1
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	0
6	1	1	0	0	0
7	1	1	1	1	1

$D = \Sigma m(1, 2, 4, 7) \Rightarrow A \oplus B \oplus C$

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

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

G_1, G_2, G_3

Borrow $= G_1 + G_2 + G_3$

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

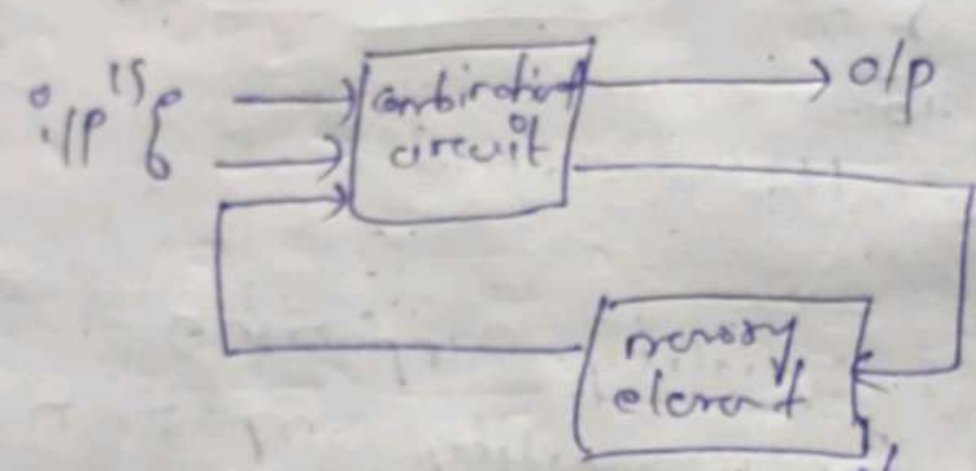
Summary:-

		Sum	Carry
2bit add	← HA	$A \oplus B$	AB
3bit add	← FA	$A \oplus B \oplus C$	$AB + BC + CA$
		Diff.	Borrow
	HS	$A \oplus B$	$\bar{A}B$
	FS	$A \oplus B \oplus C$	$\bar{A}C + \bar{A}B + BC$

Sequential circuits

→ In combinational circuit with memory element can be called as sequential circuit

→ these memory elements are connected in such a way that it forms a feedback path.



sequential circuit

→ the memory elements/ ~~devices~~ devices which are capable of storing the binary info.

→ there are two main types of sequential circuits:-

1) synchronous SC's 2) Asynchronous SC's

latch:-

→ It is an electronic device that changes its o/p immediately on the basis of the applied i/p

→ One can use it to store the binary info. either 0 or 1 at a specified time

→ A latch contains 2 i/p's 'set' & 'Reset' & it also has 2 o/p's

→ It is also a memory device just like the flip flop but it is not synchronous

→ latch is a sequential device that checks all i/p's continuously & changes its o/p accordingly at any time independent of clock signal

Flip flops

→ It is a basic memory element which is used in sequential logic system. It is also called as bistable multivibrator & toggle multivibrator

→ Flip flop is a bistable logic element with 1 or more i/p's & 2 o/p's. The o/p's are complement to each other

→ A flip flop can store binary info. either "0 or 1"

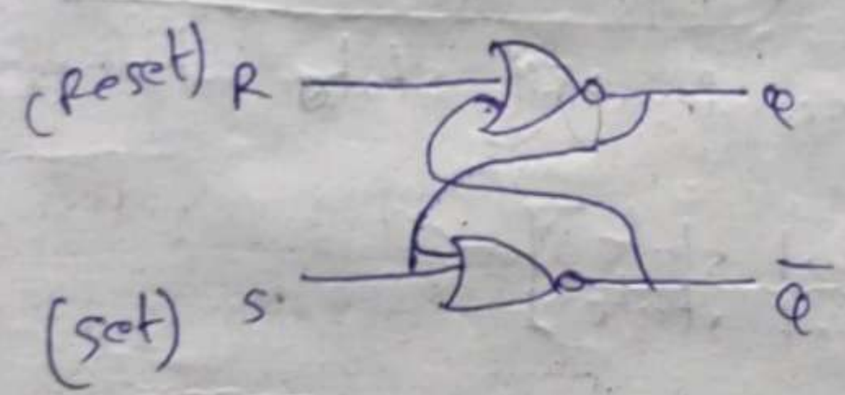
→ Flip flops are used in counters, shift registers & memory devices

Types of flip flops

1) RS flip flop / latch :
2) clocked SR flip flop
3) JK flip flop
4) master slave JK flip flop
5) D- flip flop
6) T - flip - flop

1) R-S latch

R-S latch using NOR gates.



Truth table

S	R	Q	$\bar{Q}$
0	0	0	1
0	1	0	1
1	0	1	0
1	1	not used state	

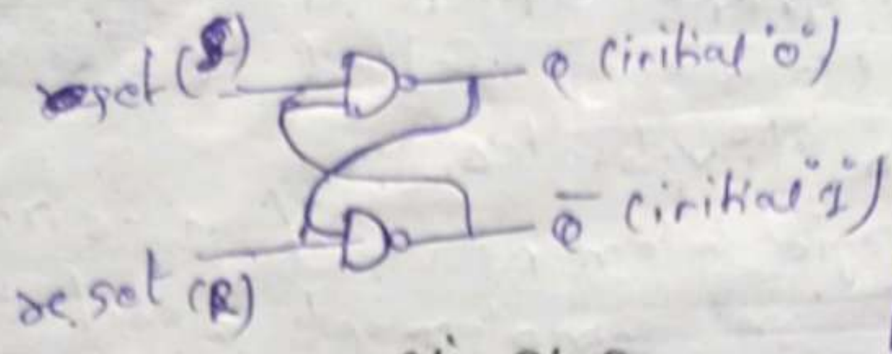
→ This latch can be used as a memory element in digital circuit

→ It can also be used as bounce elimination in keyboard or switches

→ In this case S=1, R=1 is not used since both the o/p's will be "0"'s (bcz logic 1 as a i/p to a NOR gate provides o/p as logic "0"). Both the o/p's should not be zero's ($Q, \bar{Q}$)

SR

→ latch using NAND gate



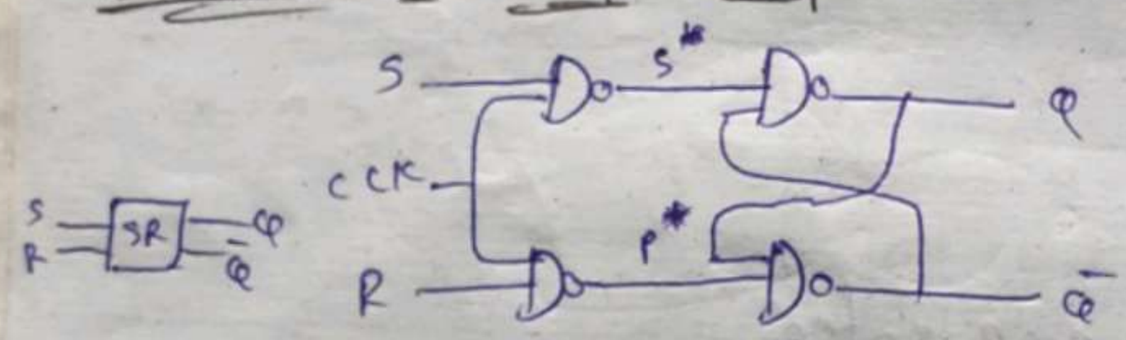
Truth table

S	R	Q	$\bar{Q}$
0	0	not used	
0	1	1	0
1	0	0	1
1	1	memory state	

latch vs flip flop

latch	flip flop
1) It uses a level edge triggering approach	1) It uses an edge triggering approach
2) clock is absent	2) clock is present
3) It is faster	3) It is slower
4) It is Asynchronous circuit	4) It is Synchronous circuit

clocked SR flip-flop



Truth table

CLK	S	R	Q	$\bar{Q}$
0	x	x	memory state	
1	0	0	memory state	
1	0	1	0	1 (Reset)
1	1	0	1	0 (Set)
1	1	1	not used	

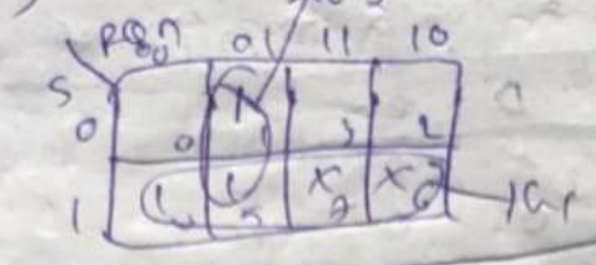
*
→ In SR flip flop we will have an invalid state whenever $S=1, R=1$.

characteristic table

[Qn = present state
Qn+1 = next state]

S	R	Qn	Qn+1	
0	0	0	0	memory state/HOLD
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	
1	0	0	1	set
1	0	1	1	
1	1	0	Is	Invalid state / dont care
1	1	1	Is	

$F = (S, R, Q_n) = Q_{n+1} = \Sigma m(1,4,5)\ \Sigma d(6,7)$



$$\boxed{Q_{n+1} = C_1 + C_2 \Rightarrow S + \bar{R}Q_n}$$

~~Excitation table~~

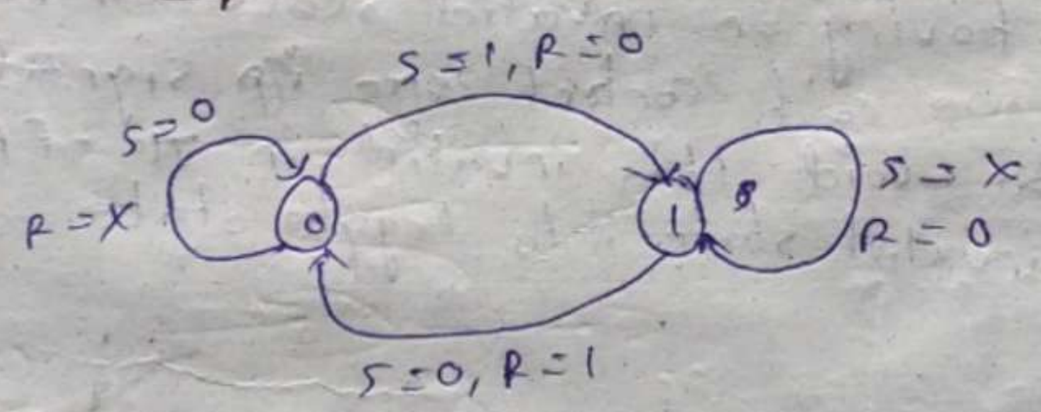
→ Q_n represents the present state, Q_{n+1} represents the next state

Excitation table

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

→ In SR flipflop an invalid state is present when S=1, R=1 this problem can be avoided by using JK flipflop

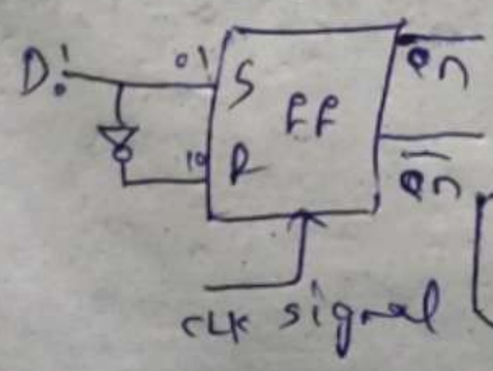
State diagram of SR flip flop



D-flip flop

→ It is a transparent latch which transmits the data from i/p side to o/p side

→ It is generally used to provide propagation delay in sequential circuits.



clk	S	R	Q_n
x	x	x	Q_n
1	0	0	memory state "Q_n"
1	0	1	0 → Reset
1	1	0	1 → set
1	1	1	invalid state/forbidden state

→ D-FF

→ we can convert a SR flip flop into D-flip flop by providing complement's i/p's to both i/ps of SR-flip flop

D-FF truth table :-

clk	D	Q_n
x	x	Q_n
1	0	0
1	0	1

characteristic table :-

	D	Q_n	Q_{n+1}
0	0	0	0
1	0	1	0
2	1	0	1
3	1	1	1

$Q_{n+1} = \Sigma m(2,3)$

$= D\bar{Q}_n + DQ_n$

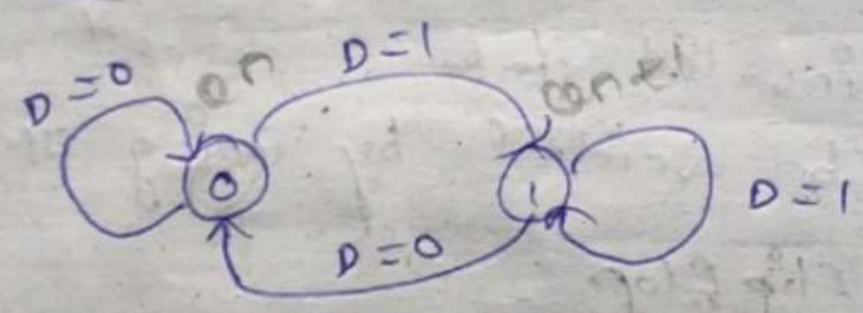
$= D(\bar{Q}_n + Q_n)$

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

excitation table :-

PS Q_n	NS Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

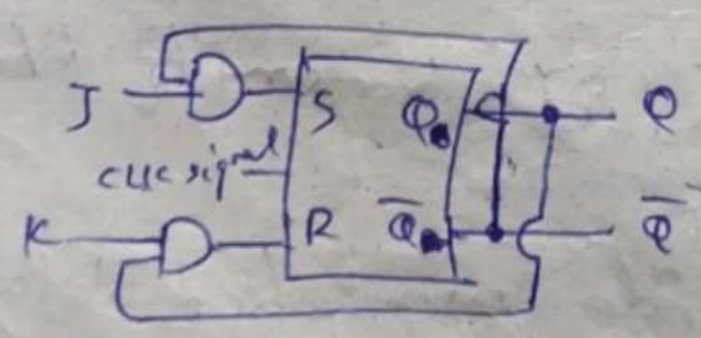
state diagram



→ In SR-FF we are having an invalid state whenever S=1, R=1. Since we can control some i/p signals

J-K flip flop we can avoid this invalid in JK-FF by turning this invalid state into toggle state

JK-flip flop



→ As shown in the above fig. the SR-FF i/p's provided with AND gate o/p's of J,K along with $\bar{Q}$, Q bcoz of this connection we can avoid S=1, R=1 state

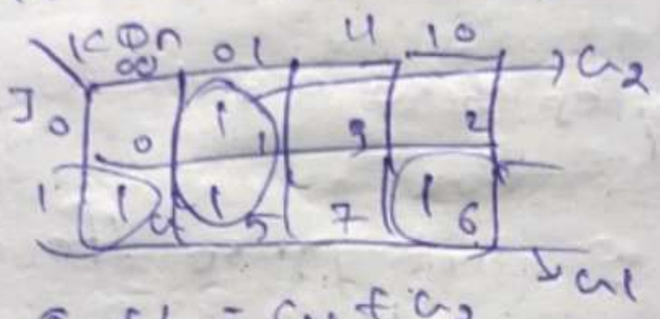
Truth table :

CLK	J	K	Q_n
x	x	x	x
1	0	0	memory state
	0	1	0 (reset)
	1	0	1 (set)
	1	1	$\overline{Q_n}$ Toggle op

characteristic table :

	J	K	Q_n	Q_{n+1}
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

$Q_{n+1} = \Sigma m(1,4,5,6)$



$Q_{n+1} = a_1 + a_2$

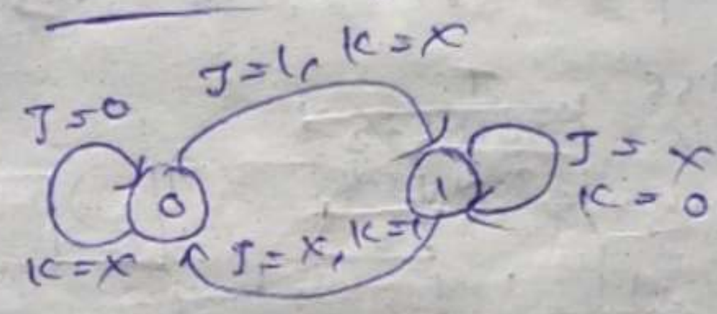
$= J\overline{Q_n} + \overline{K}Q_n$

Excitation table :

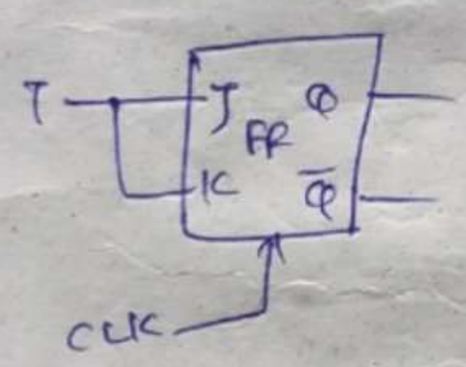
PS Q_n	NS Q_{n+1}	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

i/p possibilities

state diagram



T-Flip Flop



Truth table

CLK	T	Q_n	
x	x	Q_n	
1	0	Q_n	memory state
1	1	$\overline{Q_n}$	Toggle o/p

characteristic table :

$\Rightarrow$ we can convert JK flip flop into T-FF by shorting both JK i/p's.

characteristic table :

	T	Q_n	Q_{n+1}
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0

$Q_{n+1} = \Sigma m(1,2)$

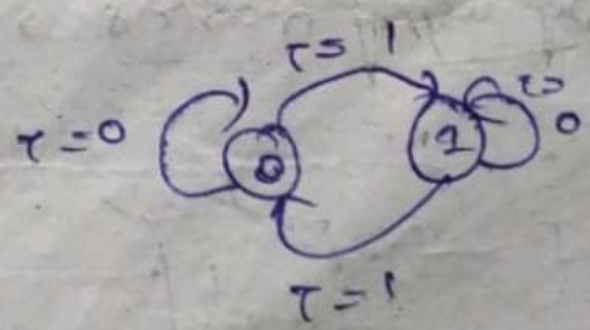
$= \overline{T}Q_n + T\overline{Q_n}$

$= T \oplus Q_n$

Excitation T :

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

state diagram



summary

char. eqns :

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

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

$Q_{n+1} = 0$

$Q_{n+1} = T \oplus Q_n$

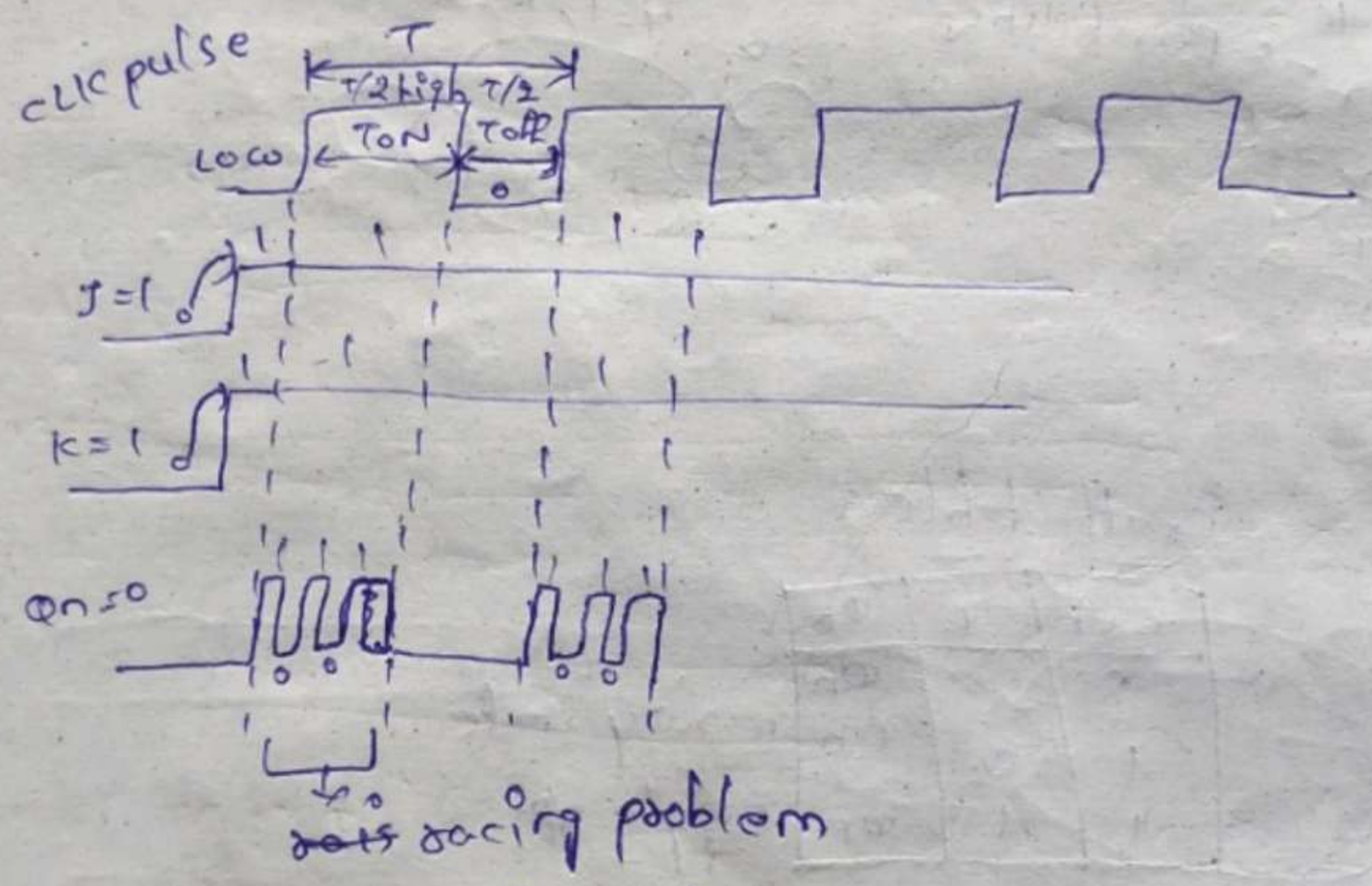
	J	K	Q_n
SR-FF	0	0	memory state
	0	1	0 (reset)
	1	0	1 (set)
	1	1	$\bar{Q}_n$ toggle o/p

→ D-FF → T-FF

→ wherever J=1, K=1 the o/p toggles b/w "0 & 1" as shown in below fig.

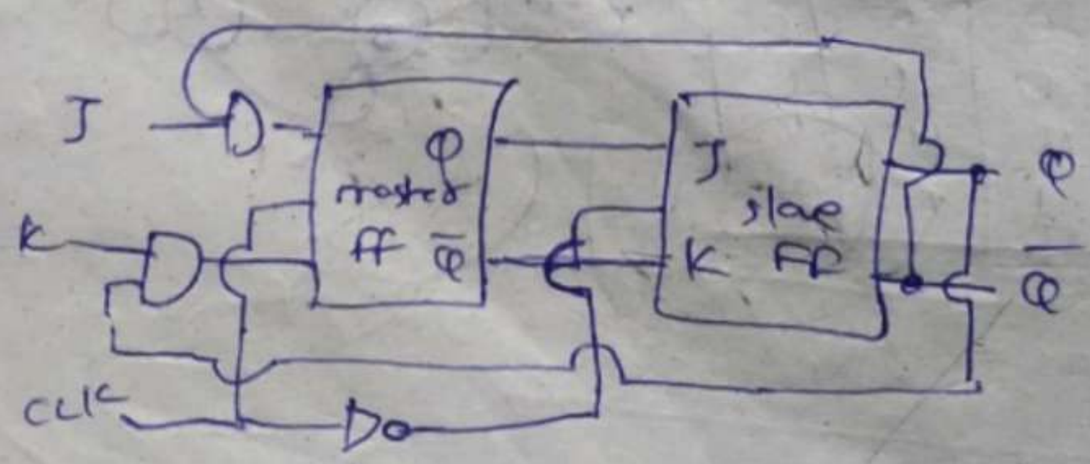
→ which means during on period of clock pulse the o/p is changing multiple times this is called noise around condition.

→ this problem can be avoided by using three methods:



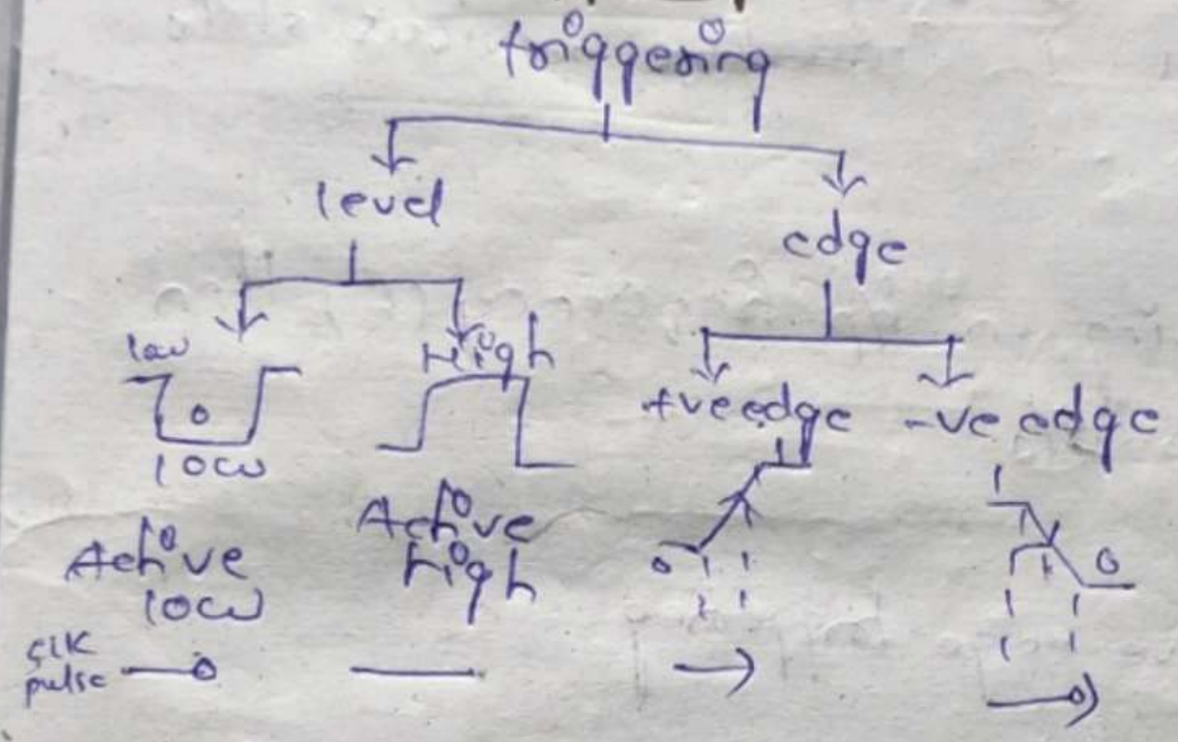
1) By using master slave JK-FF
2) By using edge triggering
3) By providing $T_{pulsewidth} < T_{propagation\ delay} < T_{clk}$ (T/2)

1) By using master slave JK-FF



→ The master FF is applied with i/p CLK pulse & the slave FF is applied with inverted CLK
→ Boz of this connection the o/p's of master FF & slave FF changes at diff. times
→ The o/p of master slave FF changes wherever the slave ~~off stage~~ FF o/p changes
→ master is level triggering & slave FF is edge triggering

<u>Types of triggering</u>



1) When of an edge triggered JK-FF are 1 then the FF is
a) raise ground condition b) Toggle cond. c) reset cond.
d) e

2) The no. of 2 to 4 lines decoders ~~will~~ to require to construct a 3 to 8 line decoder
1) 1 2) 2 3) 3 4) 4

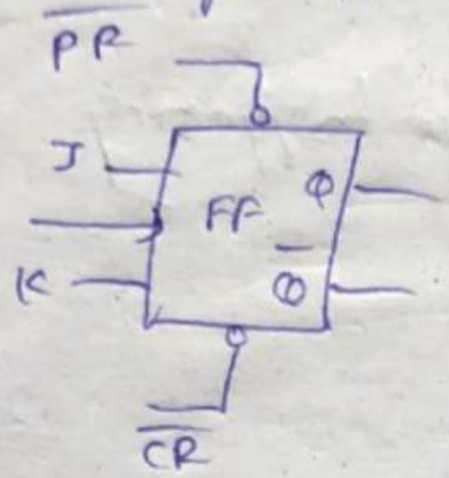
3) A JK-FF can be made from ~~SR~~ SR-FF by using two additional gates
1) OR G 2) AND G 3) NOR G 4) NAND G.

4) which one of the following FF in remains in one state indefinitely until it is directed by an i/p signal to do otherwise
a) JK FF b) SR FF c) D-FF d) T-FF

5) The data router is a ____
a) multiplexer b) Demux c) encoder d) adder
6)

Asynchronous i/p's

→ Preset & clear are used to override the FF o/p's
→ These i/p's are independent of clock pulses so these are called asynchronous i/p's & also called overriding i/p's

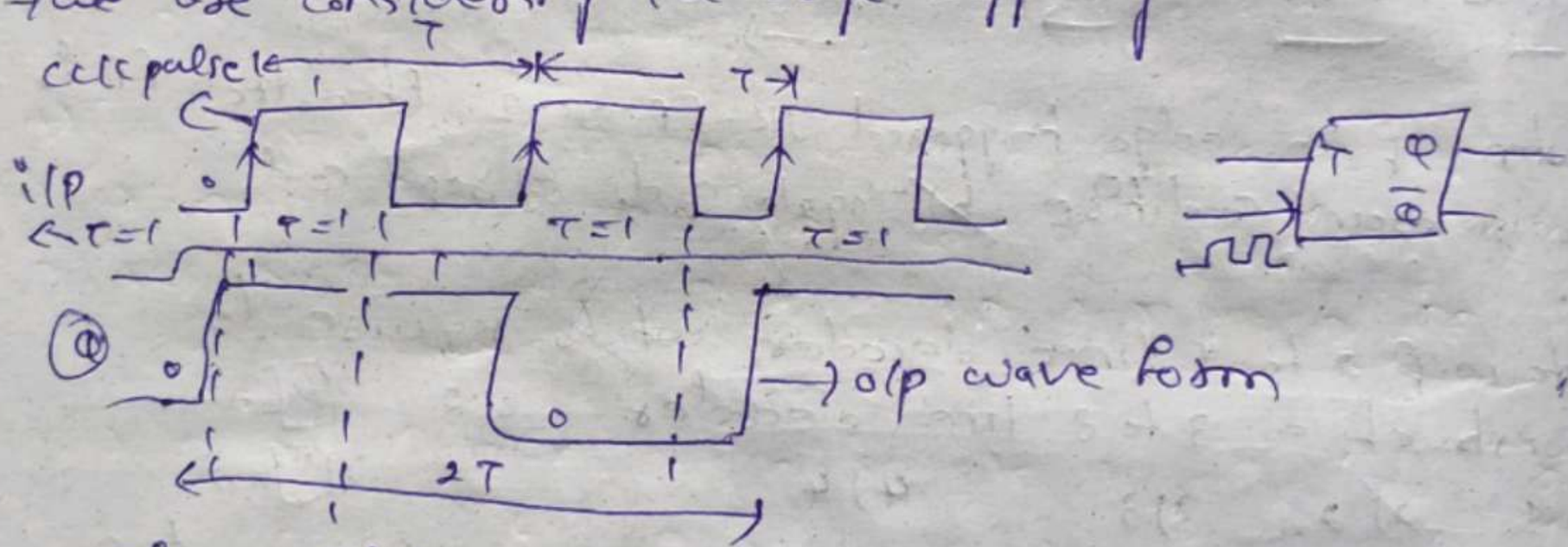


PR	CR	o/p operation
0	0	~~Invalid state~~ Invalid state
0	1	set Q=1
1	0	reset Q=0
1	1	~~Inactive~~ previous state

→ If a toggle FF is having an i/p frequency (f), then the o/p frequency is "f/2"

Toggle FF

→ we are considering +ve edge triggering



Time period (τ) = $\frac{1}{\text{frequency}}$, for clk pulse = $\frac{1}{\tau}$

I/p clock frequency. o/p clk f₀ = $\frac{1}{2\tau}$ → $\frac{1}{2}\left(\frac{1}{\tau}\right)$

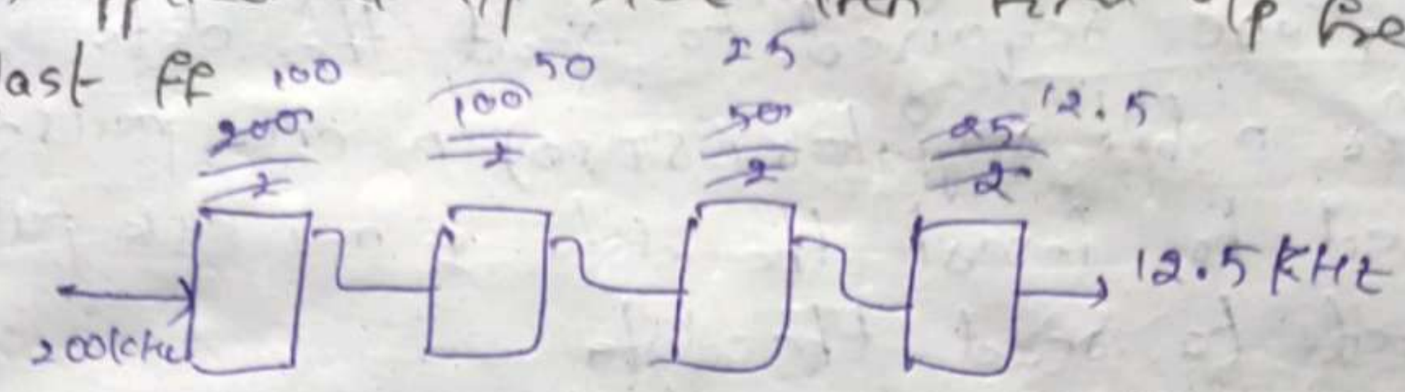
$= \frac{1}{2}(f) = \frac{f}{2}$

1) if a T-FF is provided ~~T-FF~~ with i/p frequency of 100 kilohz (clk) then the o/p frequency

→ $f_{i/p} = 100\,kHz$

o/p frequency $= \frac{f}{2} = \frac{\overset{50}{\cancel{100}}}{\cancel{2}} = 50\,kHz$

2) Consider 4 FF's are cascaded which are operated in toggle mode FF. A clock pulse of frequency 200kHz is applied at i/p side then find o/p frequency of last FF



Registers

→ In order to store more no. of bits we use more no. of FF's in diff. connections

→ Registers are sequential circuits which are combination's of FF's & logic gates.

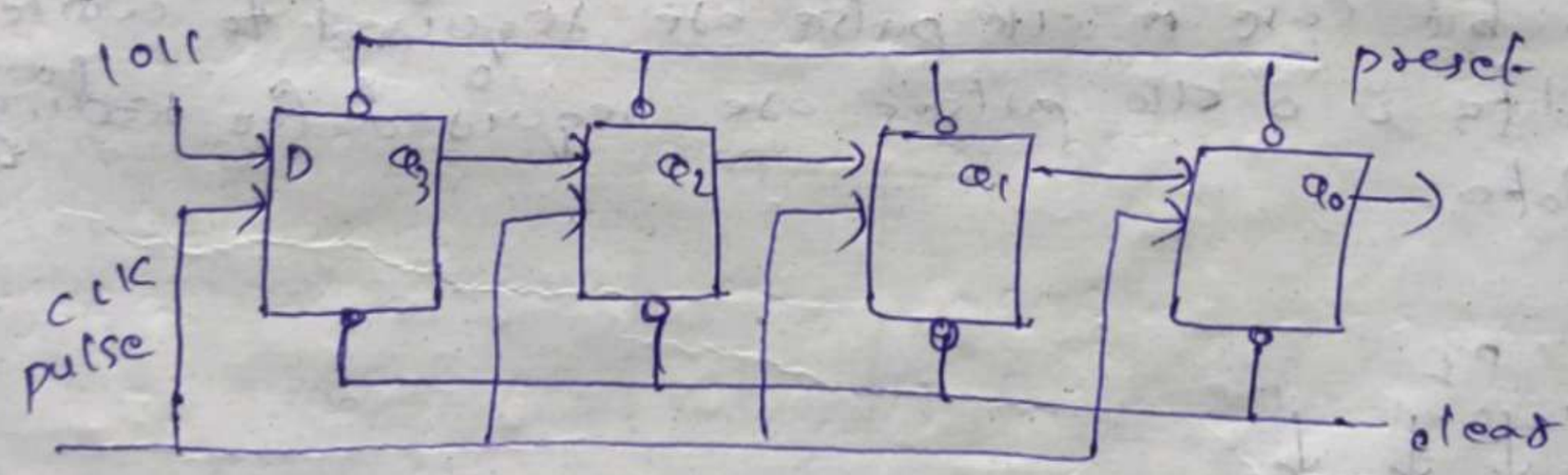
→ FF's are used for storage & logic gates are used for diff. arithmetic / boolean operations

Based on the type of connection they are classified into :

1) SISO 2) SIPO 3) PISO 4) PIPO

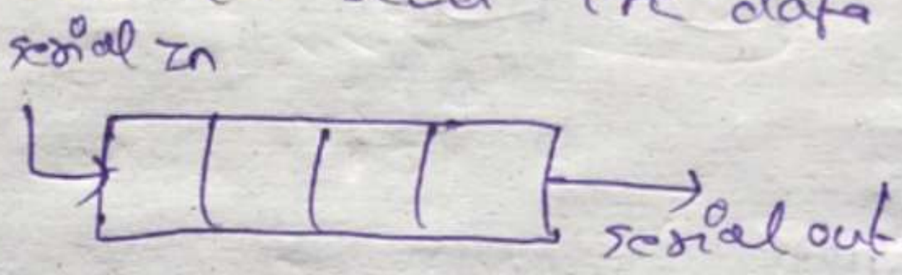
→ In register to store n bits we need n no. of FF's

1) SISO :-



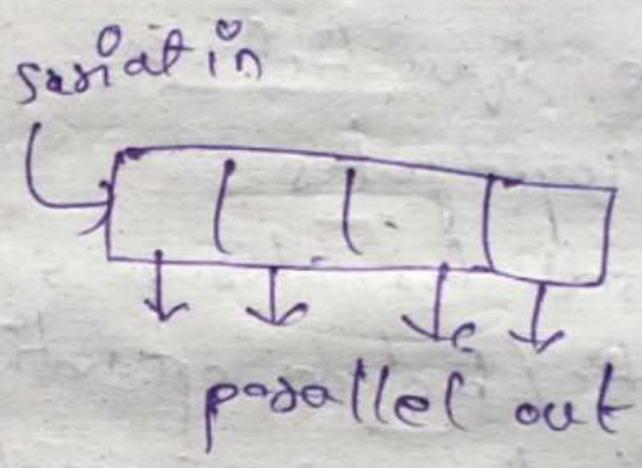
1011	clk pulse	Q_3	Q_2	Q_1	Q_0	
	0	0	0	0	0	
	1	1	Q_2	Q_1	Q_0	4 clk
	2	0	1	Q_1	Q_0	
	3	0	1	1	Q_0	
	4	1	0	1	1	
	5th	Q_3	1	0	1	3 clk
	6th	Q_3	Q_2	1	0	
	7th	Q_3	Q_2	Q_1	1	

→ As show in above table for 4-bit (SISO) 4 clk pulses are required to insert the data & three pulses are required to read the data
Similarly for n-bit register (SISO) n clk pulses are required to insert the data & n-1 clk pulses are required to read the data



Serial in parallel out

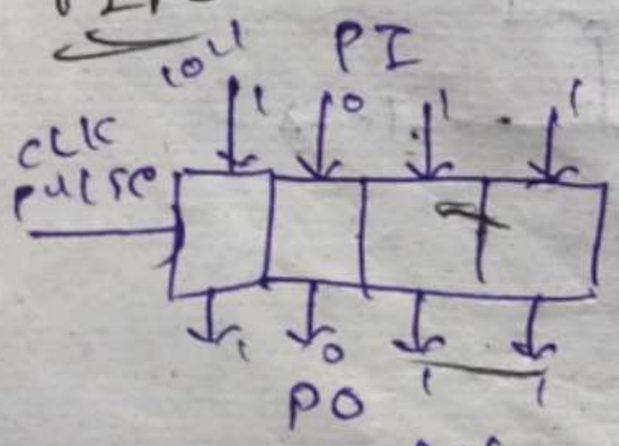
1011	Q_3	Q_2	Q_1	Q_0
1st	1	Q_2	Q_1	Q_0
2nd	1	1	Q_1	Q_0
3rd	0	1	1	0
4th	1	0	1	1



→ For serial in operation we need 4 clk pulses.
→ For parallel out operation we don't need any clk pulses since the o/p is taken from each FF directly

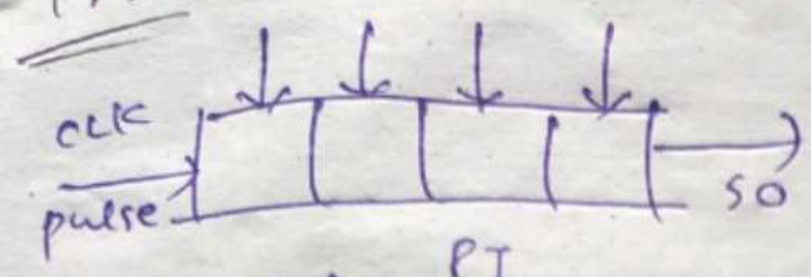
→ For n bit case n clk pulse are required to write the data & '0' clk pulses are required for reading the data.

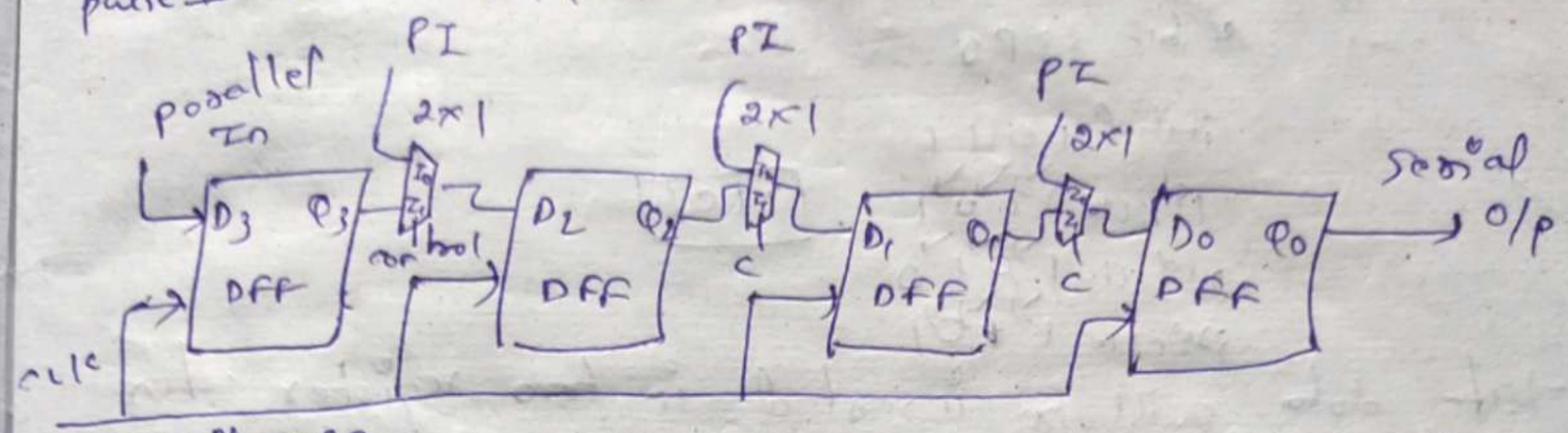
PIPO



→ In a 4-bit case only one clk pulse is required to insert the data parallely.
→ 0 clk pulses are required to read the data
→ This is same for n-bit case also

PISO :





n-bit case

→ In parallel In operation only one clk pulse is required

→ for serial out operation n-1 clk pulses are required

→ As showed in the above fig if the control signal is "0" then this register act as parallel in register

→ If the control signal is 1 then this register act as serial out register

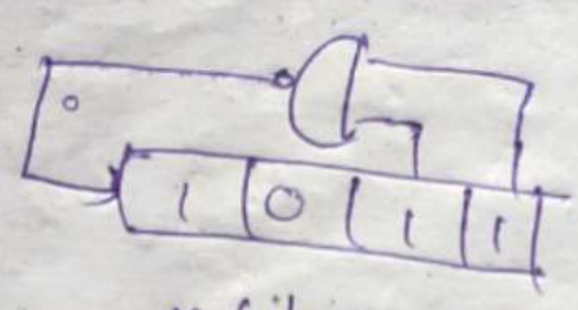
Summary

Types of register	no. of clk pulse write	read	total no. of clk pulses
SISO	n	n-1	n+n-1 = 2n-1
SIPO	n	0	n
PISO	1	n-1	1+n-1 = n
PIPO	1	0	1

→ SIPO is also called as Temporal code to spatial code conversion

→ PISO is also called as spatial code to temporal code conversion

→ In the circuit shown below of a four-bit SIPO register is loaded with 1011 after 3 clk pulses the data present in the register is



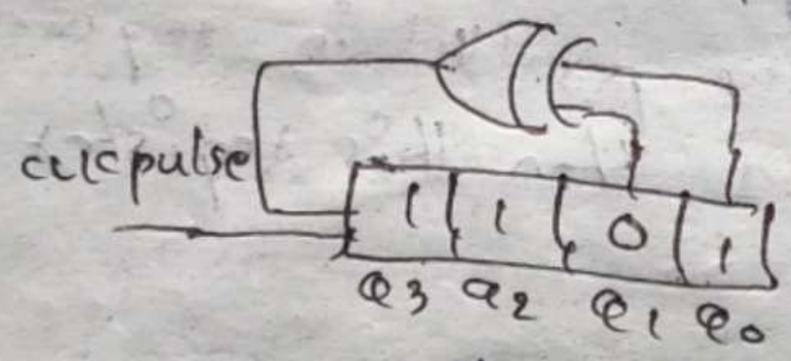
4 bit SIPO

initial 1 0 1 1
1st clk 0 1 0 1 → 1101
2nd clk 1 0 1 0
3rd clk 1 0 1 1

→ The data will be inserted in two register when the clock pulse is applied only

→ preset & clear i/ps are called as asynchronous i/ps, overriding i/ps which are independent of clk pulse

2) As shown in below fig SIPO a 4-bit register is loaded with 1101 the sequence of o/p Q_3

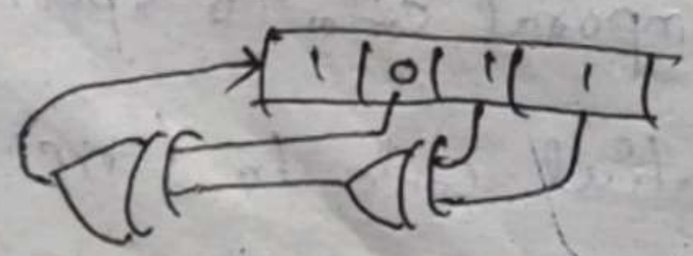


X-OR
0 0 0
0 1 1
1 0 1
1 1 0

seq. of Q_3
1 1 1 0 0 0

initial 1 1 0 1
1st clk 1 1 1 0
2nd clk 1 1 1 1
3rd clk 0 1 1 1
4th clk 0 0 1 1
5th clk 0 0 0 1

3) As shown in below fig. the register is loaded with 1011 find the no. of clk pulses are required such that the data remains same



1011
1st 0101
2nd 0010
3rd 1001
4th 1100
5th 1110
6th 0111
7th 1011

→ After 7 clk pulses the data remains same in register

→ In the above problem the data present in register after 25 clk pulses 1100

~~4) In the above~~ 1

Universal register

→ If a register can perform shift left, shift right, serial in, serial out, parallel in, parallel out then this is called Universal register

shift left register

→ In shift left operation the data will be shifted towards left side so that LSB position is inserted with new data

shift right register

→ In this register the data is shifted towards right that means MSB position is inserted with new data

Ex: shift left

0101 → (5) ×2
1010 → 10

shift right

1010 → (10) ÷2
0101 → 5

concept Related to boolean algebra

Duality:

→ For any logic expression we can find the dual form by following two steps:

1) we can interchange logic 0 & 1
2) we can interchange OR & AND operations

Ex:

$A+1 = 1 \Leftrightarrow A.0 = 0$

1) find the duality of the following expression

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

OR → AND
0 → 1

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

→ Digital design - morris Mano
→ DE - Anil

2) For a JK-FF with i/ps J = K = 1 & provided with asynchronous i/p's preset & clear inactive low condition then the o/p will be

a) memory condition 2) set 3) reset 4) Invalid Cond^n

3) The maximum no. of prime implicants in n variable boolean function is

a) $2(n-1)$ b) $n/2$ c) 2^n d) 2^{n-1}

4) For the identity $AB+\bar{A}C+BC = AB+\bar{A}C$ then its dual form is

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

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

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

d) None

Counters:

→ It is a combination of different no. of flip-flop's which are used to count the clk pulses.

→ These are classified into 2 types:

1) Synchronous counters

2) Asynchronous counters

→ In synchronous counters the clk pulse is applied to all the FF's at a time

→ In Asynchronous counter the clk pulse is applied to only one of the FF's

→ A counter can also be used as frequency divider, it is also used in wave form generator & for measurement of distance in radars.

Asynchronous Counters:

→ In this type of counters we are having three counters:

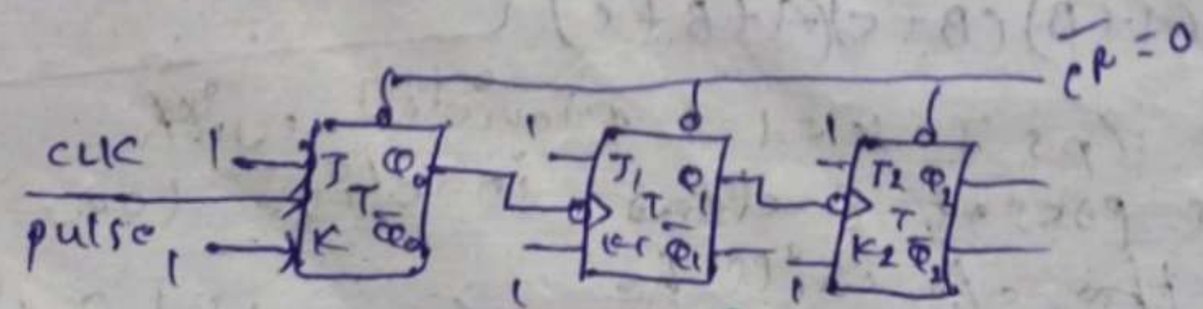
1) Ripple up counter 2) Ripple down counter

3) Ripple up-down counter

Ripple up counter:

→ In this counter we can count the clk pulses from low value to high value

3-bit counter



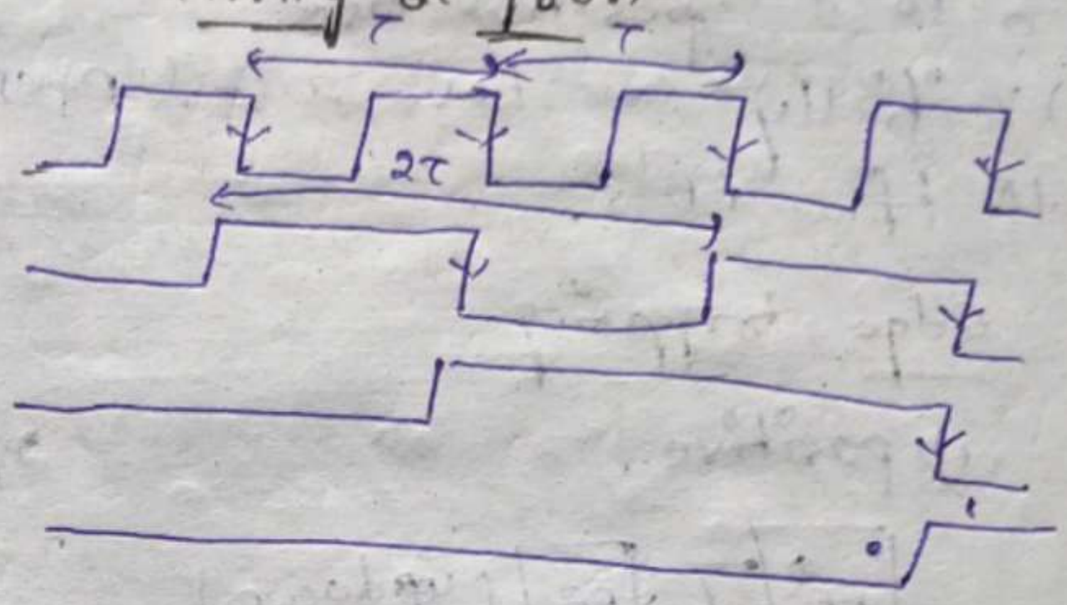
→ As shown in above circuit the clk pulse is applied to the 1st ~~put~~ FF which is considered as LSB bit

→ for remaining FF's the clk pulse is applied from the previous FF o/p (from Q / $\bar{Q}$)

→ since the circuit represents negative edge triggering the FF is activated whenever the o/p changes from 1 to 0 ($\bar{Q}$)

	Q_2	Q_1	Q_0 (LSB)
	0	0	0
1st clk	0	0	1
2nd clk	0	1	0
3rd clk	0	1	1
4th clk	1	0	0
5th clk	1	0	1
6th clk	1	1	0
7th clk	1	1	1

timing diagram



Frequency Calculator

$t_{clk} = n t_{pd}$

$t_{clk} \geq 3 t_{pd}$

$f = \frac{1}{T} \rightarrow f = \frac{1}{n t_{pd}} \Rightarrow \boxed{f_{max} = \frac{1}{n t_{pd}}}$

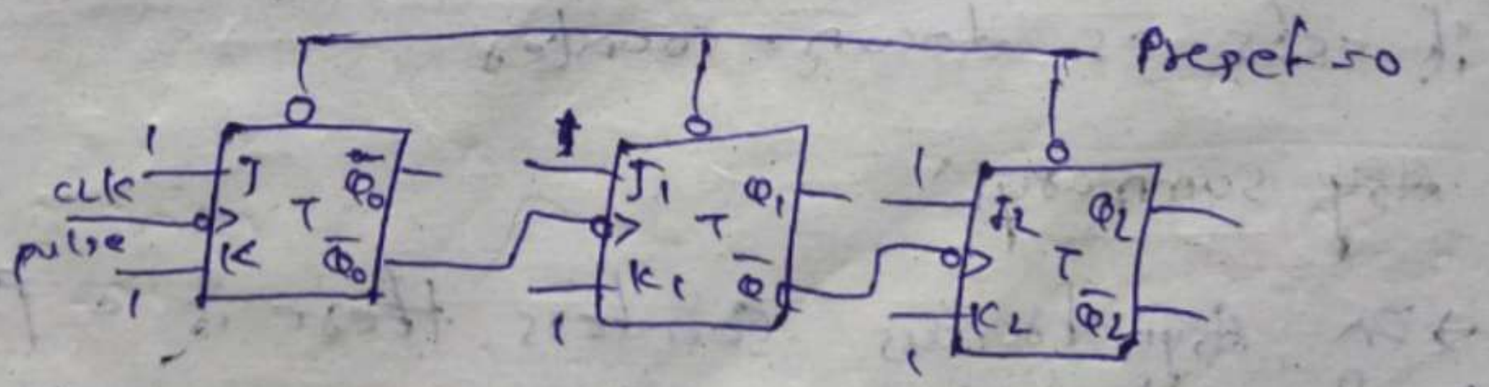
→ As shown in the timing diagram Q_0 toggles per every clk pulse. Next, Q_1 toggles whenever Q_0 changes from "1 to 0".

→ similarly Q_2 toggles whenever Q_1 changes from "1 to 0".

→ So, in asynchronous ripple counters for n-bits we need n no. of FF's

3-bit down counter

	Q_2	Q_1	Q_0
	1	1	1
1st	1	1	0
2nd	1	0	1
3rd	1	0	0
4th	0	1	1
5th	0	1	0
6th	0	0	1
7th	0	0	0



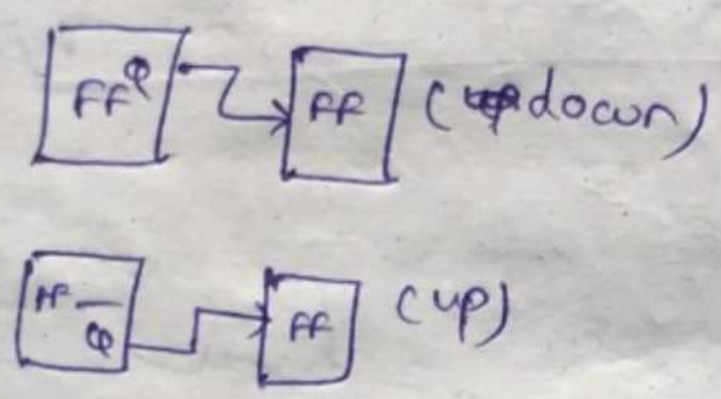
→ As shown in above fig. the clk pulse is applied to the 1st FF which is considered as LSB bit

→ since, $\bar{Q}$ is considered as clk pulse for remaining FF's even though -ve edge triggering is used the o/p of FF changes wherever Q changes from "0 to 1" positive rise

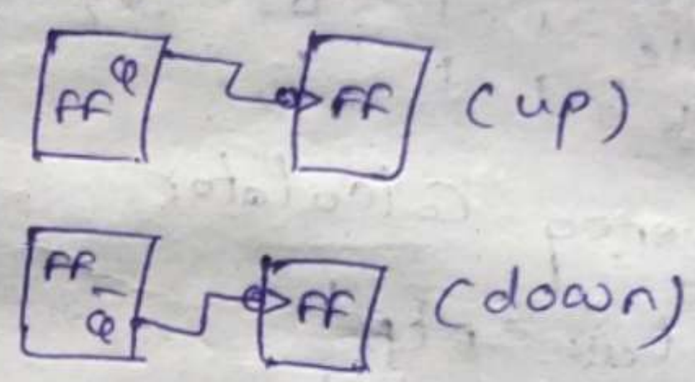
→ Initially we have used preset i/p in order to set all the FF's.

Edge triggering

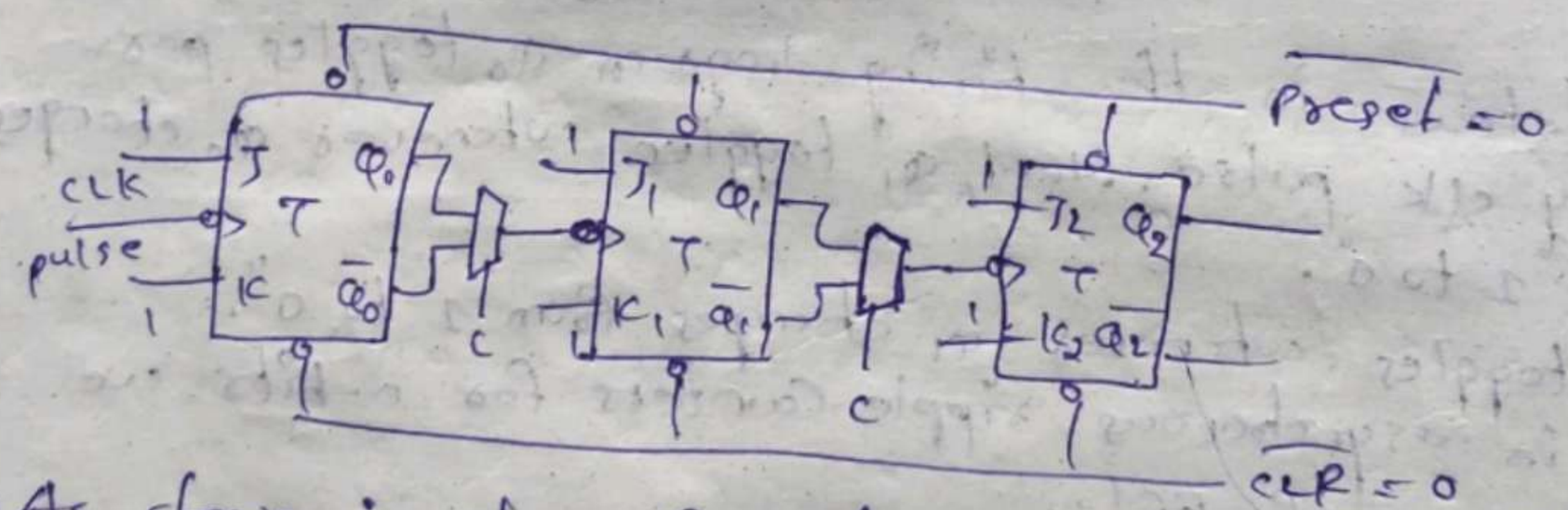
1) positive :-



2) -ve :-



3-bit up-down Counter



→ As shown in above fig. wherever the control signal "c" = 0 it acts as up-counter. & wherever "c" = 1 it acts as down-counter

Summary :

→ In Asynchronous counters there is a possibility of having decoding error

→ It is slower compared synchronous circuits

1) In a 4-bit synchronous circuit which is applied with as the i/p frequency of 12 MHz then the o/p frequency is = $\frac{\text{i/p frequency}}{2^n}$

o/p frequency = $\frac{i/p \text{ frequency}}{2^n} = \frac{12\times10^6}{16}$

$f = 12MHz$ $n = 4$ bits

Modulus N counter

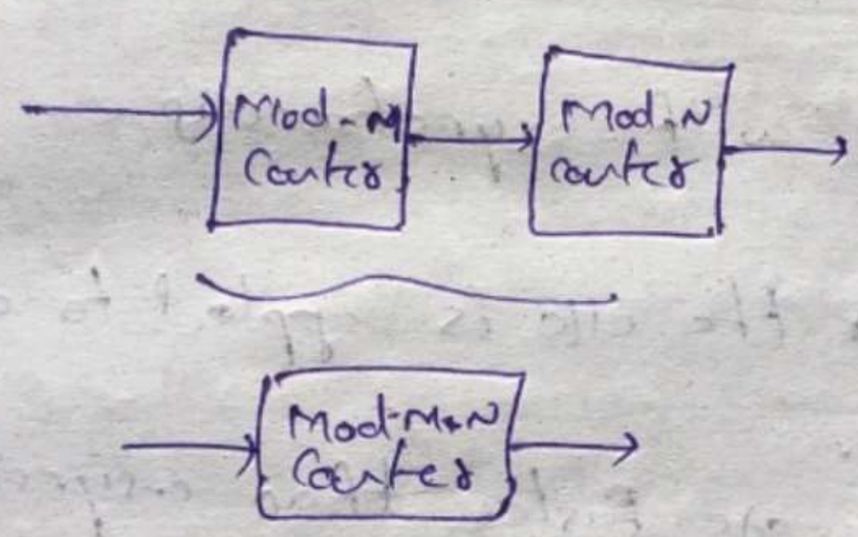
→ for a n-bit counter the no. of states will be 2^n

→ Based on the application we can use only few states which is represented by Mod Number / modulus.

→ 2^n states are divided into no. of used states & no. of unused states.

→ This Modulus number decides the repetation of different states

→ Modulus N counter with cascade connection



1) A modulus 10 counter is applied with 10MHz clk frequency then the o/p. frequency will be

$f = 10MHz$, Mod-10 ⇒ $f_{o/p} = \frac{10MHz}{10} = 1MHz$

2) In a circuit mod-3, mod-4, mod-5 are cascaded. If we apply a clk frequency of 30MHz at the i/p side the o/p frequency will be.

→ mod-3 → mod-4 → mod-5

→ mod 3×4×5

30MHz → mod 60 → $f_o = \frac{f_i}{60} \rightarrow \frac{30}{60} = \frac{1}{2} = 0.5$ MHz

Mod-10 counter / BCD counter

1) The no. of flip-flops required in order to design BCD counter

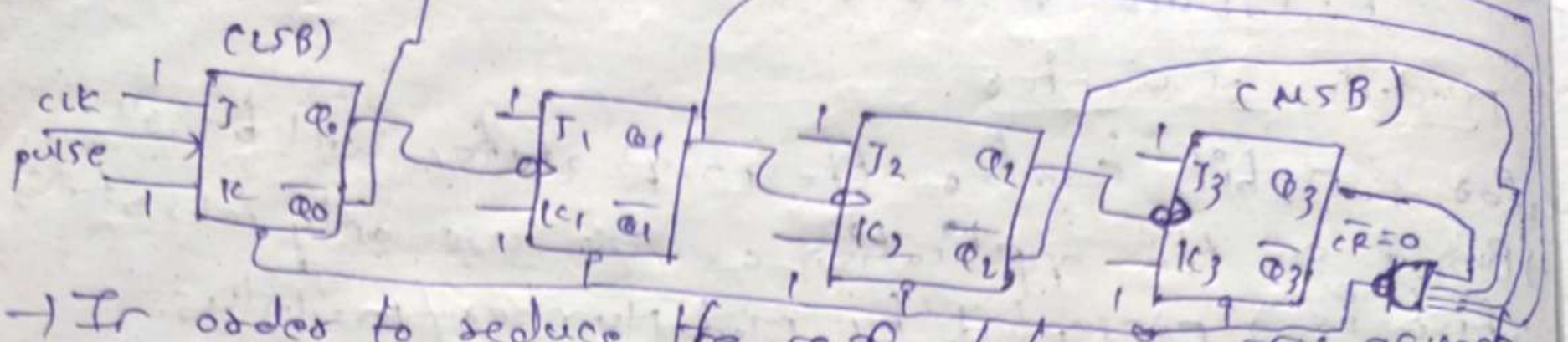
a) 3 FF b) 4 FF c) 5 FF d) 6 FF

mod-10 counter
no. of states = 10 = N

$N \le 2^n$

$10 \le 2^4 \Rightarrow 10 \le 1$

$\boxed{n = 4}$



→ In order to reduce the no. of states in any asynchronous circuit we should use the feedback connection

→ since, A BCD counter counts from "0–9" in the 11th state that means "1010" the total circuit should be reset

→ for this operation we are considering ($Q_3 \overline{Q_2} Q_1 \overline{Q_0}$) as feedback connection as shown in the fig

» Design mod-5 counter with asynchronous

<u>synchronous Counters</u>

→ In synchronous counter the clk is applied to all the FF's simultaneously.

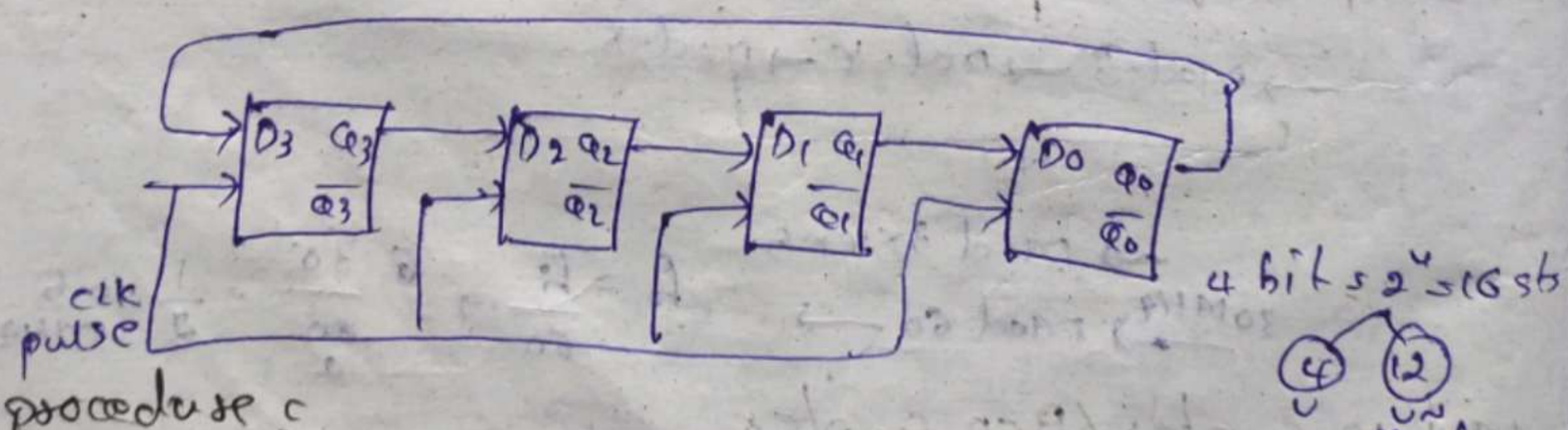
→ synchronous counters are faster than asynchronous counter.

→ There is no possibility of decoding error

→ Based on the feedback connection they are classified into 2 types:

1) Ring counter 2) Thomson/twisted ring counter

<u>4-bit ring counter</u>



<u>procedure</u>:

→ The ring counter is a <u>shift register</u> with feedback correction

→ In the circuit the last FF o/p is taken as i/p for the 1st FF

→ In ring counter we consider 1-FF o/p as logic 1 which is rotated among all the FF's with the application of each clk pulse

→ In a ring counter the no. of FF's of small "n" will have n-used states only

n-bit → 'n' no. of FF's

* 'n' FF → 2^n states

$$2^n = n + (2^n - n)$$

used sts unused sts

→ The ring counter is faster than ripple counter

→ The total time period in a ring counter is one propogation delay of FF

time period (τ) = 1 tpd

tpd → propogation delay

→ In a ring counter if 'f' is the i/p frequency then the o/p frequency is $f_0 = \frac{f}{n}$ → no. of bits / no. of FF's

→ A ring counter can be used as wave form generator & it also used successive approximation ADC (analog to digital convertor)

1) How many FF's are required to design mod-32 binary counter

a) 3 b) 4 c) 5 d) 6

$N \le 2^n$

$32 \le 2^5$

2) A mod-16 counter is holding the count 1001 after 31 clk pulses the counter value is

a) 1001 b) 1100 c) 1000 d) 1111

mod-16 counter → 0 – 15 (1111)

$\boxed{n=4}$

→ 1001 31 → 1001 – 1 → 1000

32 → 32 → 1001

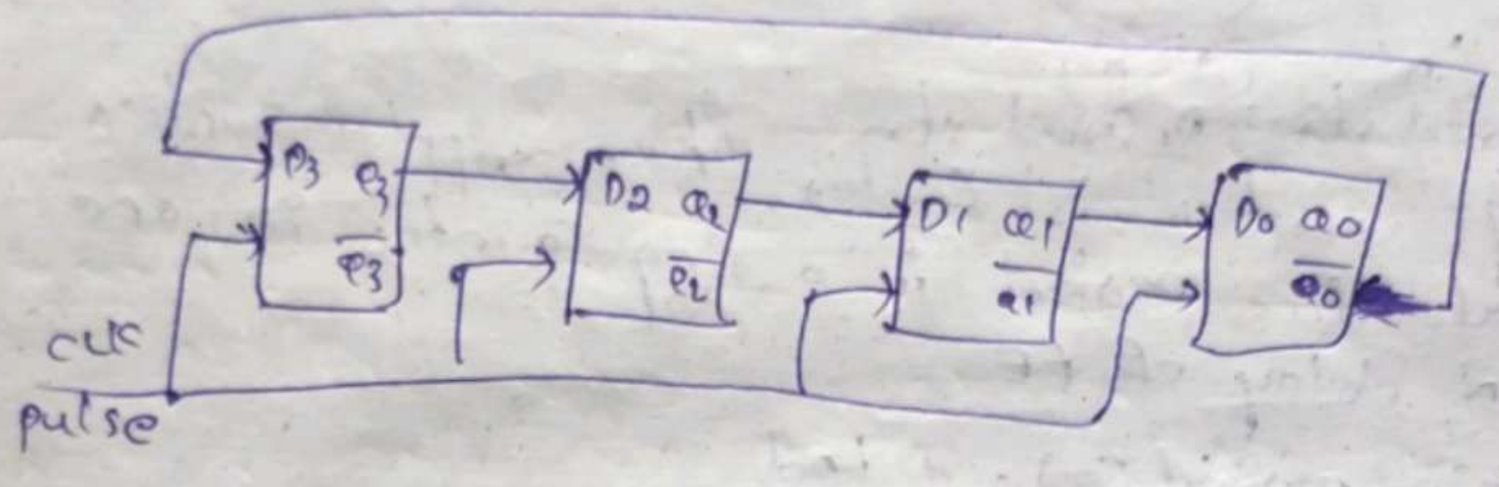
→ As mention the counter is holding 1001 initially so, after 16 clk pulses the data will again 1001.

→ so after 32 clk pulse the data is 1001 since the question mentions 31 clk the data is 1001 – 1 which is 1000

1) A mod-16 counter is holding the count 1010. so, the counter value will be after 37 clk pulses
a) 1001 b) 1000 c) 1001 d) 1111

16 → 100
32 → 11010
37th → 11010
1111

4-bit Johnson ring counter



	Q_3	Q_2	Q_1	Q_0	
	0	0	0	0	
1st	1	0	0	0	4-bit
2nd	1	1	0	0	
3rd	1	1	1	0	2×4 = 8 sts
4th	1	1	1	1	2n used sts
5th	0	1	1	1	$2^n - 2n$ unused sts
6th	0	0	1	1	
7th	0	0	0	1	
8th	0	0	0	0	

→ For a 4-bit Johnson counter the total no. of used states are "8" & the total no. of unused states are $2^4 - 8 \Rightarrow 8$

Note:
→ In a n-bit Johnson counter the total no. of used states = 2n
→ The total no. of unused states = $2^n - 2n$ where n = no. of FF's
→ In Johnson counter the last FF o/p $\overline{Q}$ is taken as feedback
→ Based on the no. of used states Johnson counter is far better than ring counter
→ In ring counter there is a possibility of lock out condition which can be avoided in Johnson counter

→ In a Johnson count if i/p frequency is f then o/p frequency is $\frac{f}{2n}$

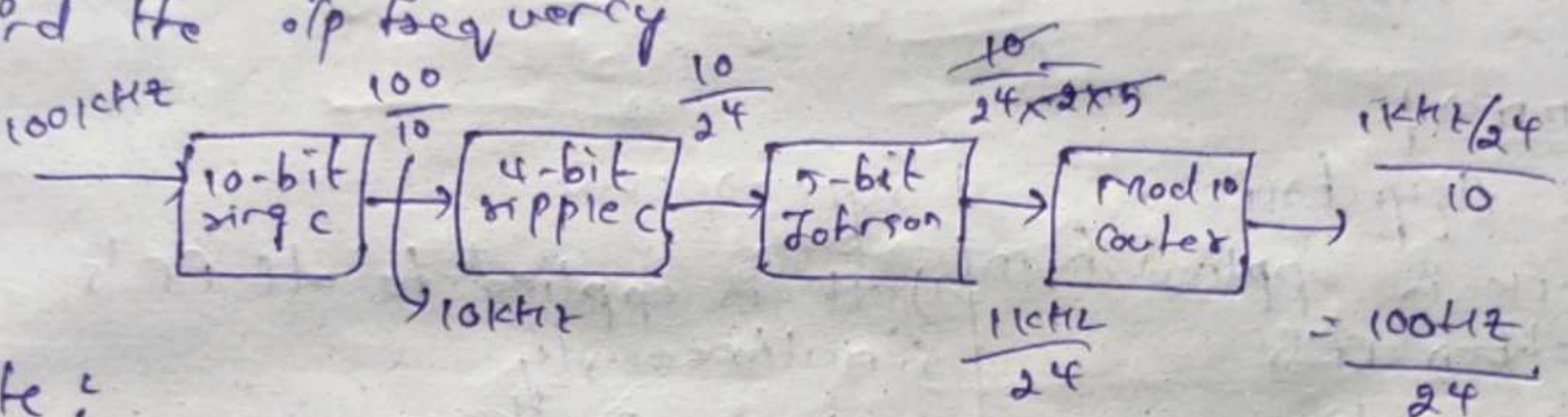
→ It is slower than ring counter

1) A 5-bit Johnson counter is applied with 100kHz then the o/p frequency is

sol, n=5 f i/p = 100 kHz

$$f_{o/p} = \frac{100}{2\times5} = \frac{10\cancel{0}}{1\cancel{0}} = 10\,kHz$$

2) The following circuit i/p frequency is 100kHz then find the o/p frequency



Note:

Type of counter	no. of used states	frequency counter	counter scaling
1) Ripple counter	2^n	$\frac{f}{2^n}$	$2^n:1$
2) mod-N counter	N	$\frac{f}{N}$	nil
3) Ring counter	n	$\frac{f}{n}$	nil
4) Johnson counter	2n	$\frac{f}{2n}$	2n:1
5) T-FF	1FF	f/2	

3) The no. of FF's are required to design a decade counter

a) 2 b) 3 c) 4 d) none

4) A 5-bit counter

a) has a modulus of 5 c) can't have modulus i.e., >32

b) has a modulus of 2 d) has modulus i.e., ≤32

$N \le 2^n \rightarrow N \le 2^5 \rightarrow N \le 32$

5) The minimum no. of FF's are required to construct a mod-10 Johnson counter & a mod-5 ring counter respectively

a) 10, 5 b) 5, 10 c) 5, 5 d) 10, 10

6) The decoding errors are fast more lightly in the case of
a) ring counter b) Johnson counter c) ripple counter
4) more

7) A 5-bit asynchronous counter is made of 5 FF's with each propogation delay 12 nanosecond's then the total propogation delay of the counter is
a) 12ns b) 24 ns c) 36 ns d) 60ns.

$$\tau = n \cdot t_{pd} \rightarrow 5 \times 12 = 60ns$$

8) A decade counter is also called as bcd counter.

Summary:

Asynchronous	Synchronous
1) clk is applied from other FF's	1) clk is applied to all FF's simultaneously
2) It is slower	2) It is faster
3) Decoding error is present	3) No decoding error is present
4) Ex: Ripple counter	4) Ex: Ring counter, Johnson counter
5) only sequence counting is possible up/down	5) Any count sequence is possible

→ In a synchronous series counter for a n-bit case the total clk pulse $\tau_{clk} \geq t_{pd} + (n-2)\, t_{pd}$ And gate

$$f_{max} = \frac{1}{t_{pd} + (n-2)\, t_{pd}\ \text{And gate}}$$

Synchronous parallel counter

→ In this case the total time period is sum of propagation delay of FF & propogation delay of AND gate

$$\tau_{clk} \geq t_{pd} + t_{pd}\ \text{AND gate}$$

$$f_{max} = \frac{1}{t_{pd} + t_{dp}\ \text{AND gate}}$$

AD & DA Conversion

→ In general ~~ppl~~ communication the data should be converted from analog to digital & digital to analog

→ In this process of conversion there will be different errors which can be reduced by using proper circuit based on application

→ these applications are decided based on accuracy & response time

Types of DA converters

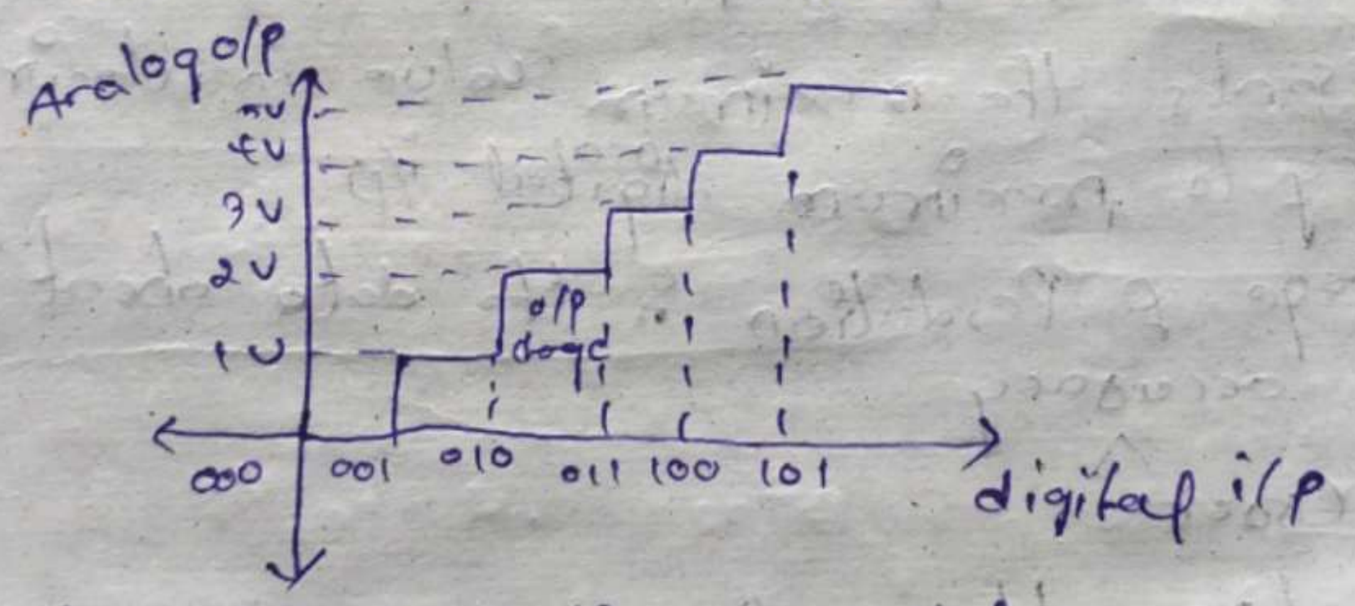
1) n-bit weighted type resistor DAC
2) R-2R DAC type

Resolution :- It is defined as the change in analog o/p corresponding to 1 LSB increment in binary i/p

$$\text{Resolution} = \frac{V_{ref}}{(2^n - 1)}$$

n = no. of bits in the i/p

V_{ref} = the reference voltage corresponding to logic 1/max value



→ Analog o/p = Resolution × digital i/p

$$= \frac{V_{ref}}{(2^n - 1)} \times \left(\sum_{n=0}^{\infty} 2^i \text{ bit} \right)$$

Full scale voltage

→ this is maximum voltage available at the o/p side corresponding to the maximum digital i/p where all bits are logic 1.

→ It is represented by V_{FS}

percentage of Resolution

→ It is defined as the ratio of resolution to the full scale voltage

$$\% \text{ Resolution} = \frac{\text{Resolution}}{V_{FS}} \times 100$$

$$= \frac{V_r/(2^n-1)}{V_{FS}} \times 100$$

at max voltage case $\boxed{V_r = V_{FS}}$

$$= \frac{V_{FS}/(2^n-1)}{V_{FS}} \times 100 = \boxed{\frac{1}{2^n-1} \times 100}$$

full scale Voltage

A o/p = Resolution × (Dig i/p)

$$V_{FS} = \frac{V_r}{2^n-1} \times (__)$$

at max i/p case (2^n-1)

$$V_{FS} = \frac{V_r}{\cancel{2^n-1}} \times \cancel{2^n-1} \Rightarrow \boxed{V_{FS} = V_r}$$

→ "n" represents the no. of bits

→ "2^n-1" represents the maximum value in decimal corresponding to maximum digital i/p

→ the percentage of Resolution is the data about the system accuracy

Maximum Error

→ In any ADC/DAC the maximum error is less than or equal to resolution

$$\boxed{\text{Error} \le \text{Resolution}}$$

MCQ

1) In a 4-bit DAC the reference voltage is 10V then the analog o/p voltage corresponding to binary data 1001

a) 3V b) 6V c) 9V d) 12V

no. of bits = 4, $V_{ref} = 10V$ i/p = 1001

$$\text{Analog o/p} = R \times (\text{dig i/p})$$

$$= \frac{V_{ref}}{2^n - 1} \times (\text{digi/p})$$

$$= \frac{10}{2^4 - 1} \times (9) \rightarrow \frac{\cancel{10}^{2}}{\cancel{15}_{3}} \times \cancel{9}^{3} = 6V$$

2) In a 8-bit DAC with the reference voltage of 20V which is having the digital i/p has 1010 then find Resolution

$$R = \frac{V_{ref}}{2^n - 1} \rightarrow \frac{20}{2^8 - 1} \rightarrow \frac{20}{255}$$

3) In a DAC the percentage of Resolution is 0.4%. then find the no. of bits.

a) 5 b) 7 ~~c)~~ 8 d) 9.

$$\% \text{ of Resolution} = \frac{1}{2^n - 1} \times 100 \rightarrow 0.4 = \frac{1}{2^n - 1} \times 100$$

$$= (2^n - 1) \times 0.4 = 100 \rightarrow 2^n - 1 = \frac{100}{0.4} = \cancel{\frac{1000}{4}}\ 250$$

$$2^n - 1 = 250 \rightarrow \boxed{2^n = 251}$$

$$\boxed{2^8 = 256}$$

4) In a DAC the no. of bits is "10" & $V_{ref} = 5V$ then find maximum value of error.

$$\text{Error} \leq \text{Resolution}$$

$$= \frac{V_r}{2^n - 1} \Rightarrow \frac{5}{2^{10} - 1} \rightarrow \frac{5}{1024 - 1} \rightarrow \frac{5}{1023} \rightarrow \frac{5}{(1000)} \text{ approx} = 5mV$$

$\boxed{\text{Ans} = 4.96 \text{ approx value}}$

5) For a 10-bit representation (variables) the max. no. of self dual functions

a) 2^n b) 2^{n-1} ~~c)~~ $2^{2(n-1)}$ d) 2^{2^n}

Ex:- For n = 1

$\rightarrow 2^{2(1-1)}$

$\rightarrow 2^{2^0}$

$\rightarrow 2^1$

$= 2$ A, $\bar{A}$

For n = 2

$2^{2(2-1)}$

$= 2^{2^1}$ A B

$= 4$ A $\bar{A}$ B $\bar{B}$

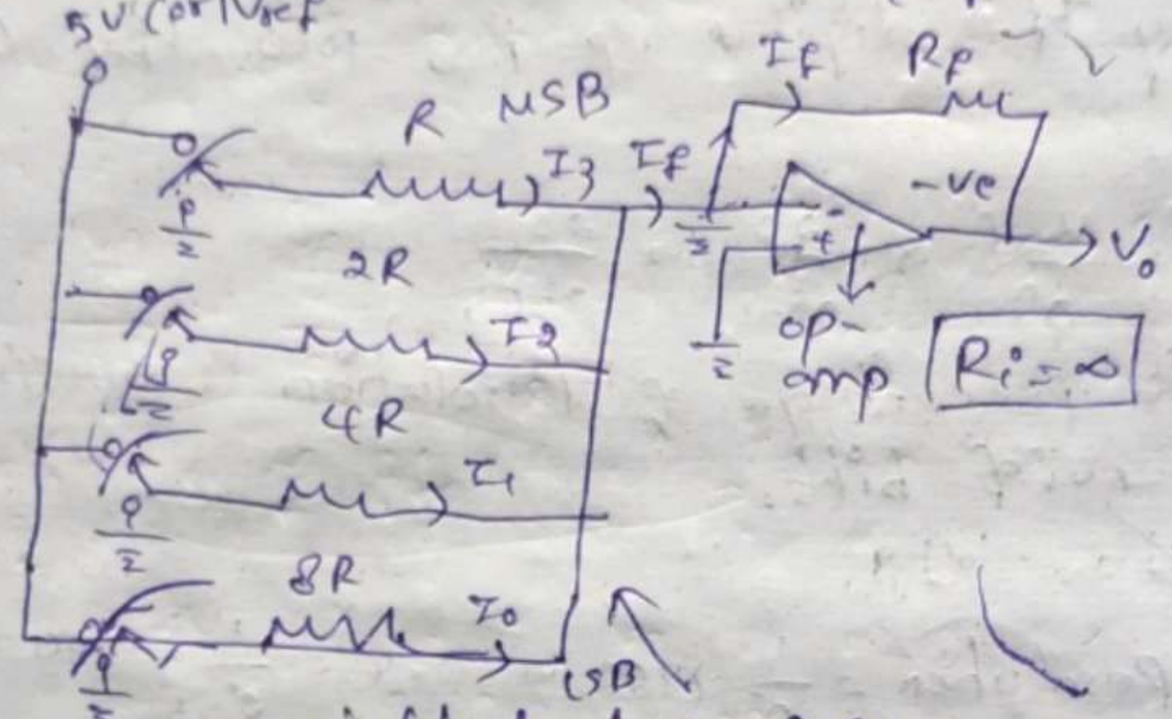
weighted type resistor

→ R-2R resistor (ladder type ADC)

→ while converting a digital data into analog form we need to consider two reference voltages one is logic "1" & another is logic "0"

→ logic "1" represents the maximum reference voltage

→ logic "0" represents the least / ground voltage



weighted type DAC

the total current ⇒ $I_f = I_0 + I_1 + I_2 + I_3$

$$I_f = \frac{V_{ref}}{8R} + \frac{V_{ref}}{4R} + \frac{V_{ref}}{2R} + \frac{V_{ref}}{R}$$

the o/p voltage

⇒ for inverting case [op. Amp]

$$\boxed{V_o = -I_f R_f} \qquad \boxed{V_o = \frac{-V_{ref}}{8R}(b_0 + 2b_1 + 4b_2 + 8b_3) R_f}$$

for n-bit case :

$$V_o = \frac{-V_{ref}}{2^{(n-1)}R}(b_0 + 2^1 b_1 + 2^2 b_2 + \dots 2^{n-1} b_n) R_f$$

→ In Binary weighted DAC as no. of bits increased we need to use more no. of resistors bcoz of this, this circuit is less efficient

→ In Binary weighted DAC. the LSB resistance is maximum & the MSB resistance is minimum

→ $\boxed{\text{LSB resistance} = (2^{n-1}) \cdot \text{MSB resistance}}$

→ To overcome this problem we are using R-2R ladder n/w in which we are using fixed value resistance

→ In weighted type n/w the given MSB resistance value is 10kΩ & digital i/p is 101. the feedback resistance is 5kΩ & the V_{ref} is 5V

1) what is the LSB resistance value
2) what is the total current
3) what is the o/p voltage

1) LSB resistance value $= (2^{n-1}) \cdot$ MSB resistance

$$= 2^{(3-1)} \cdot 10 \Rightarrow 2^2 \times 10 \Rightarrow 40k\Omega$$

2) Total current $\Rightarrow I_f = I_0 + I_1 + I_2$

$b_2 b_1 b_0$
101

$$= \frac{V}{40} b_0 + \cancel{\frac{V}{20} b_1} + \frac{V}{10} b_2$$

$$= \frac{5}{40} + \frac{5}{10} \Rightarrow \boxed{\frac{25}{40k}}$$

3) o/p voltage $(V_0) = -R_f I_f$

$R_f = 5\cancel{0}kΩ$

$$\boxed{V_0 = +\cancel{5}\left(-\frac{25}{\cancel{40}k_{8}}\right) = -\frac{25}{8k}}$$

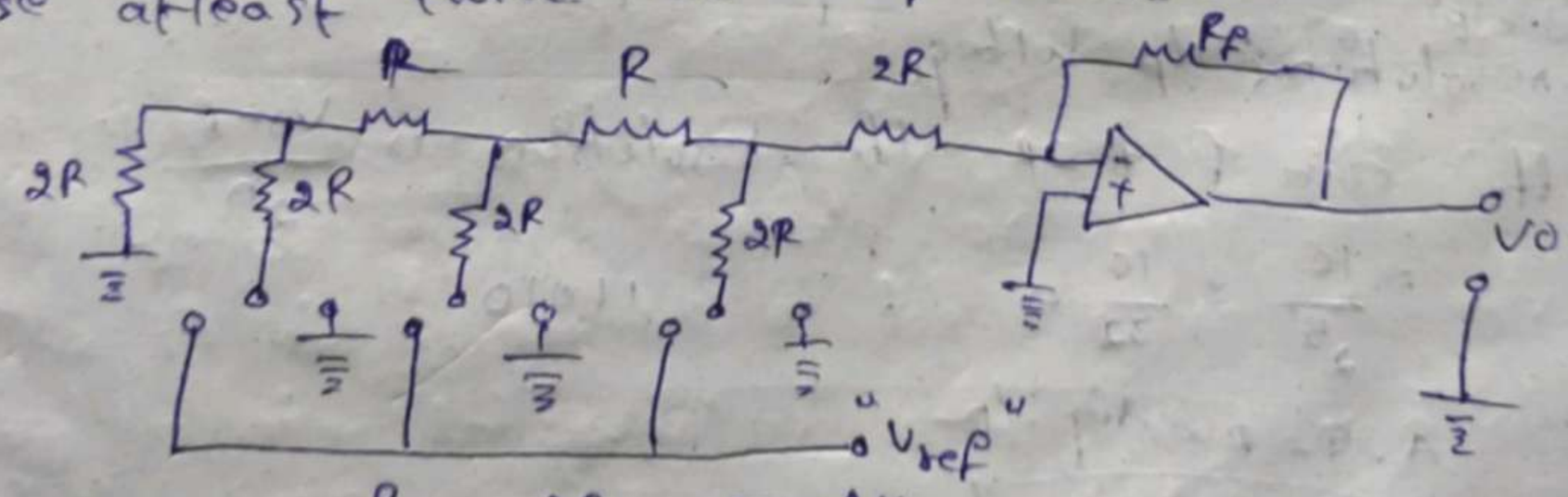
R-2R ladder type DAC

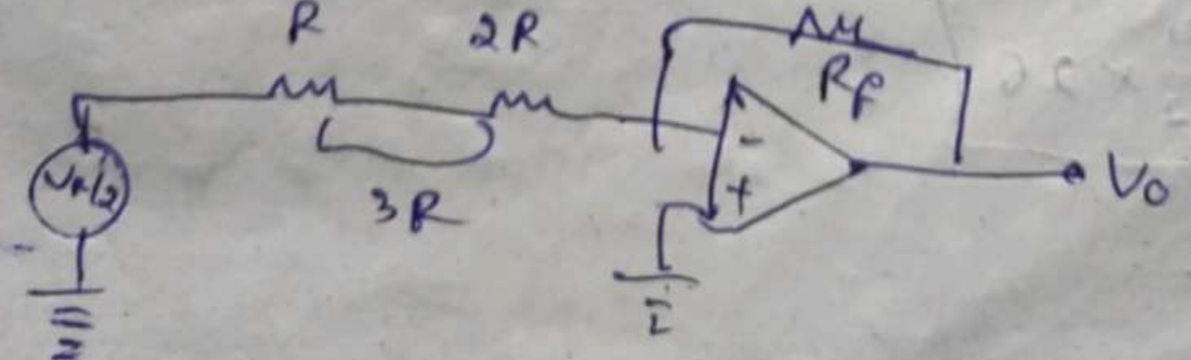
→ In this type of DAC we are using fixed resistances $(r_1 - r_2)$

→ this provides more accurate data compared to binary weighted DAC

→ In this DAC also we are using an operation amplifier in inverting mode

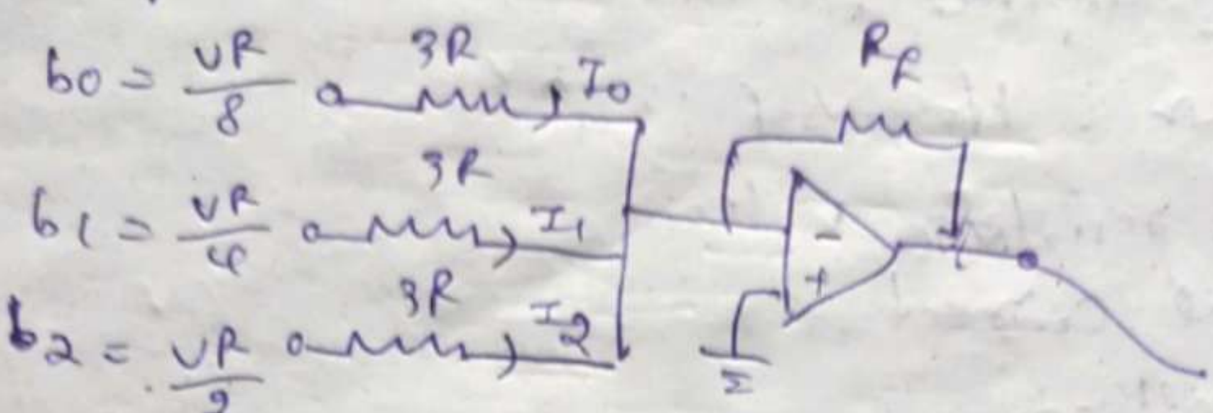
→ Based on the no. of bits of digital data we need to use atleast twice the no. of resistors.





$V_0 = -I_f R_f$

→ The equivalent circuit for different i/p's of digital data



→ As stour in above circuit whenever the i/p bit is logical 1 that particular connection is activated

→ $I_f = I_0 + I_1 + I_2$

$$= \frac{\frac{V_R}{8}}{3R} + \frac{\frac{V_R}{4}}{3R} + \frac{\frac{V_R}{2}}{3R}$$ By using ohms law $\boxed{I = \frac{V}{R}}$

→ Summation of the total circuit is individual branch

$$= \frac{\frac{V_R}{8}}{3R} b_0 + \frac{\frac{V_R}{4}}{3R} b_1 + \frac{\frac{V_R}{2}}{3R} b_2$$

$$= \frac{V_R}{8 \times 3R}\left[b_0 + 2b_1 + 4b_2\right]$$

For n-bit case : $I_f = \frac{V_R}{2^n 3R}\left[b_0 + 2^1 b_1 + 2^2 b_2 + \dots\right]$

→ The dynamic range of any ADC is analog to digital conversion is $(6n + 18)$ dB

11 In a R-2R ladder DAC the no. of digital i/ps are 5 bits & V_{ref} is 10V, the digital i/p is 11010 & find the value of resolution & o/p voltage

A) For the case of R-2R the resolution is $\frac{V_{ref}}{2^n}$

$$= \frac{10}{2^5} = \frac{10}{32} \Rightarrow$$ 11010

A.O = R × dig i/p

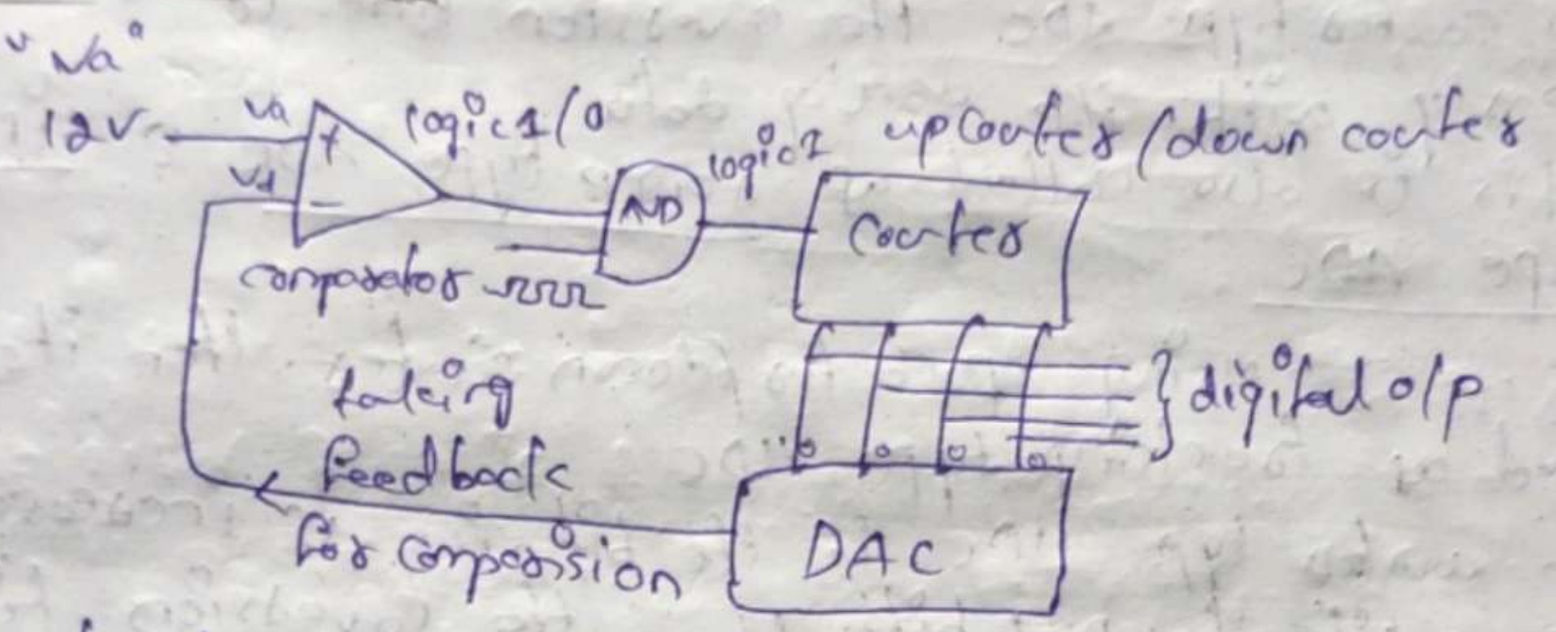
$$\boxed{AO = \frac{10}{32} \times 26}$$

Types of ADC

→ In the process of converting an analog data into digital form we are using 4 types of ADC's

1) Counter type ADC
2) successive approximation ADC
3) Flash type ADC
4) Dual slope Integrating type ADC

Counter type ADC



→ In counter type ADC we are using a comparator & a counter (up counter)

→ the comparator is used to compare the analog i/p data with o/p digital data

→ initially the comparision b/w i/p & o/p data is taken as either logic "1" (or) logic "0"

- if $V_a > V_d$ then comparator o/p is logic "1"
- if $V_a < V_d$ the comparator o/p is logic "0"

→ this comparator o/p is provided as i/p for AND logic system

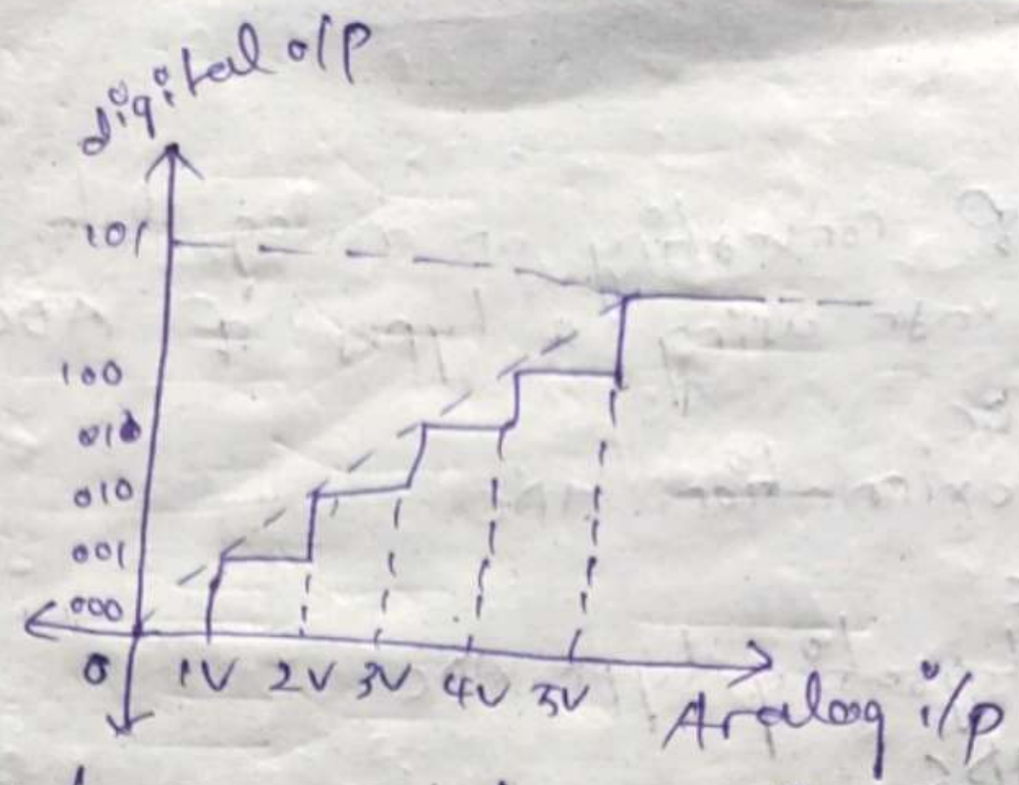
→ the counter will start increasing whenever the AND logic system o/p is logic "1"

→ the counter o/p is the digital o/p which is again converted into analog form

→ In order to compare with i/p analog data for this purpose we are using DAC

- Min. conversion time $= 1 \times T_c$
- Max conversion time $= (2^n - 1) T_c$
- Avg. conversion time $= \frac{T_c + (2^n - 1) T_c}{2} \Rightarrow \frac{\cancel{T_c} + 2^n T_c - \cancel{T_c}}{2}$

$\Rightarrow 2^{n-1} T_c$ //



→ In counter type ADC the conversion time T_c is varying directly with i/p analog data

→ This is also called as <u>ramp type ADC</u> & <u>single slope ADC</u>

→ If we are using up counter/down counter then it is called as tracking type ADC

→ In counter type ADC if the bit size is increased from n-bits to n+1 bits then the conversion time is almost double.

Max conversion time $= (2^n - 1)T_c$

<u>Ex</u>: $n = 7$ bits $\longrightarrow n+1 = 8$ bits

$(MCT) = (2^7 - 1)T_c$ $\quad (MCT)_2 = (2^8 - 1)T_c$

$= 127\,T_c$ $\quad = 255\,T_c$

$(MCT)_2 - (MCT) = \cancel{127 - 255}\ 255 - 127$

$= 128\,T_c = 2^7 T_c$

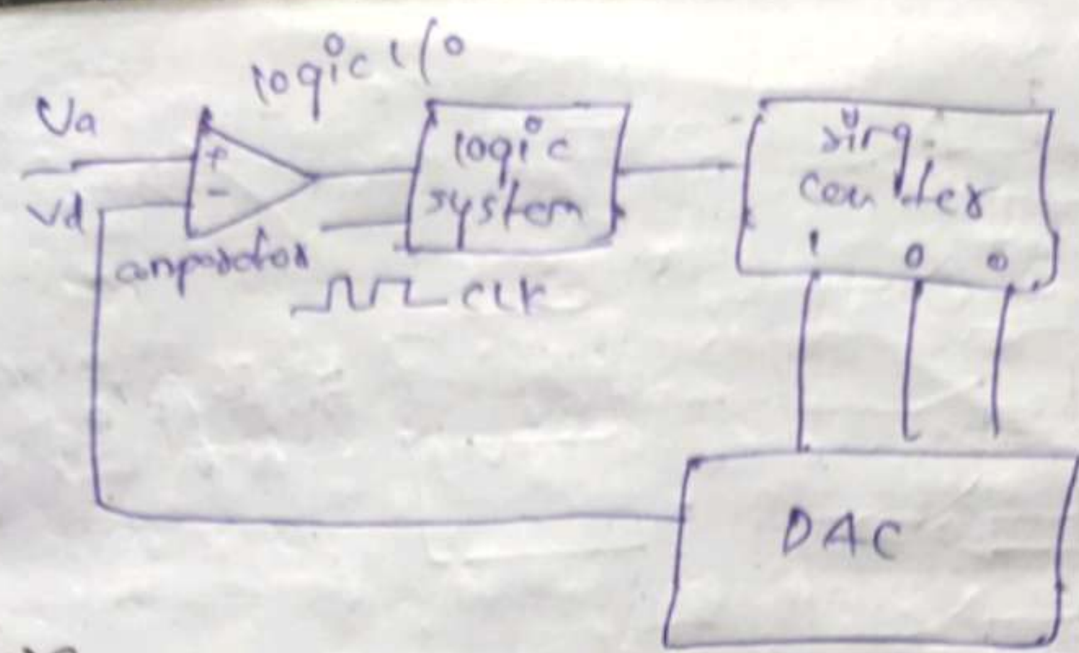
$\boxed{\text{change in conversion time} = 2^n T_c}$

<u>Successive approximation type ADC</u>

→ In this ADC we are using same components as in the case of counter type ADC expect a ring counter

→ this ring counter is used to set one of the FF's

→ the o/p from ring counter is converted into analog form by using DAC which is used to compare with i/p analog data.

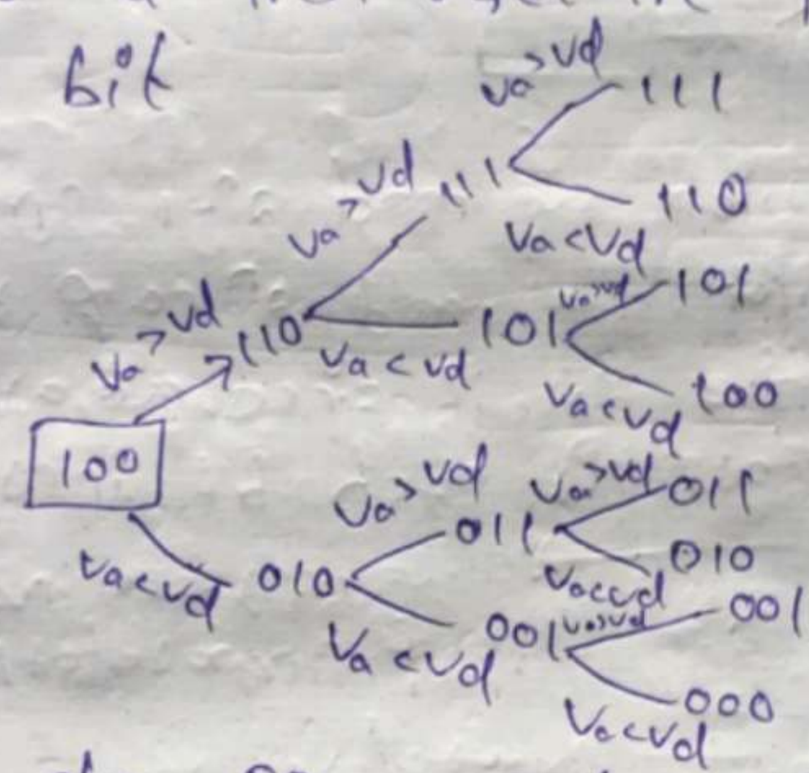


procedure

step-1: if Va > Vd then set the the next bit

step-2: if Va < Vd then reset the present bit & set the next bit

ex:



→ As shown in above fig a control circuit is used to reset the current bit & to set next bit whenever Va < Vd.

→ In this type of ADC the conversion time is independents of analog i/p voltage

→ For n bit conversion it requires n no. of clk pulses

→ Total conversion time = nT_c (n = no. of bits)

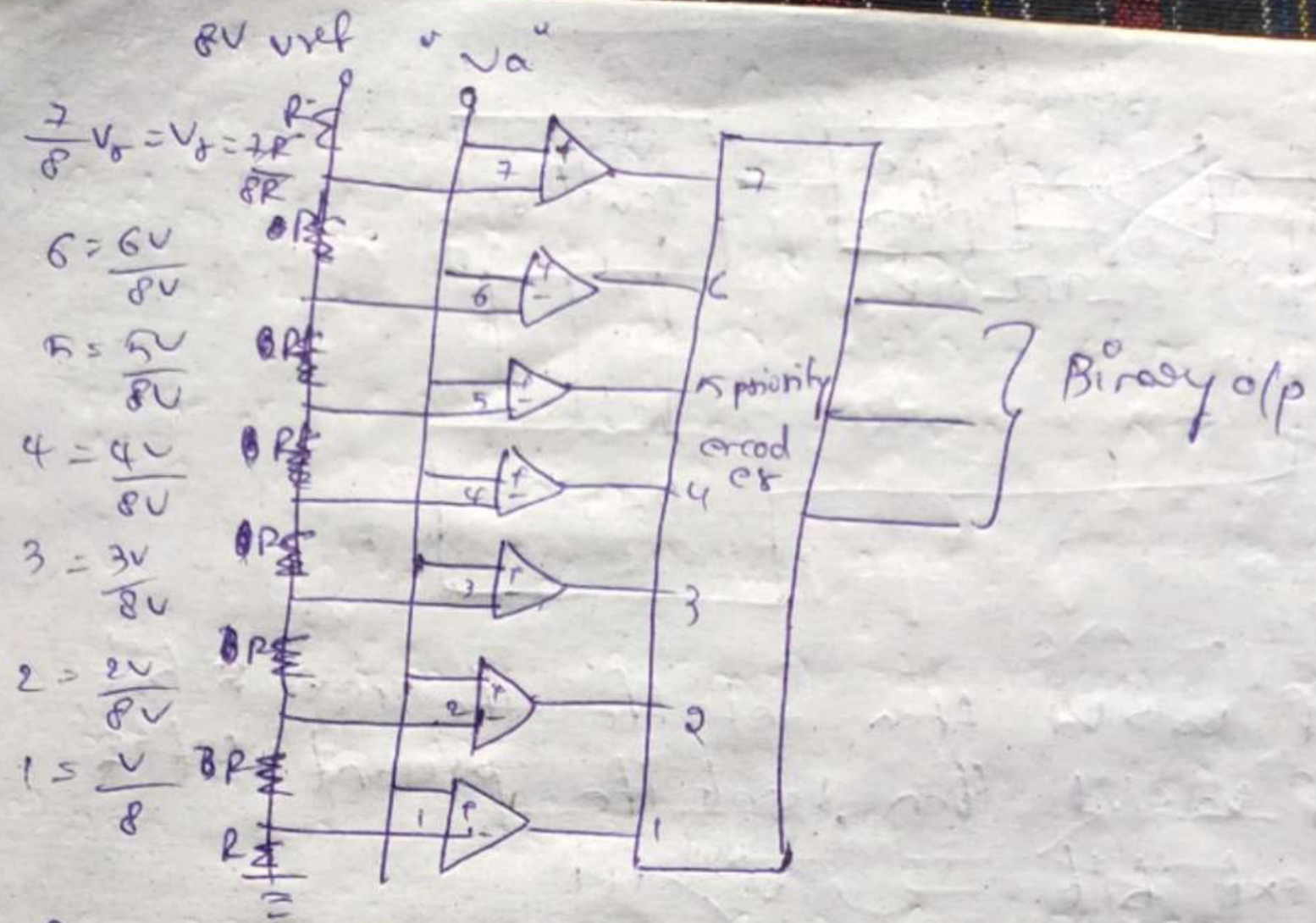
In SAR type ADC to convert an analog i/p voltage 10V the conversion time is 20 microseconds. what will be the conversion time for 2V in i/p analog data

a) 4 sec b) 10 sec c) 15 sec d) 20 sec

Flash type ADC

→ In flash type ADC the max. no. of clk pulses for n-bit conversion is only 1 clk pulse

→ It is the fastest type ADC among all ADC's

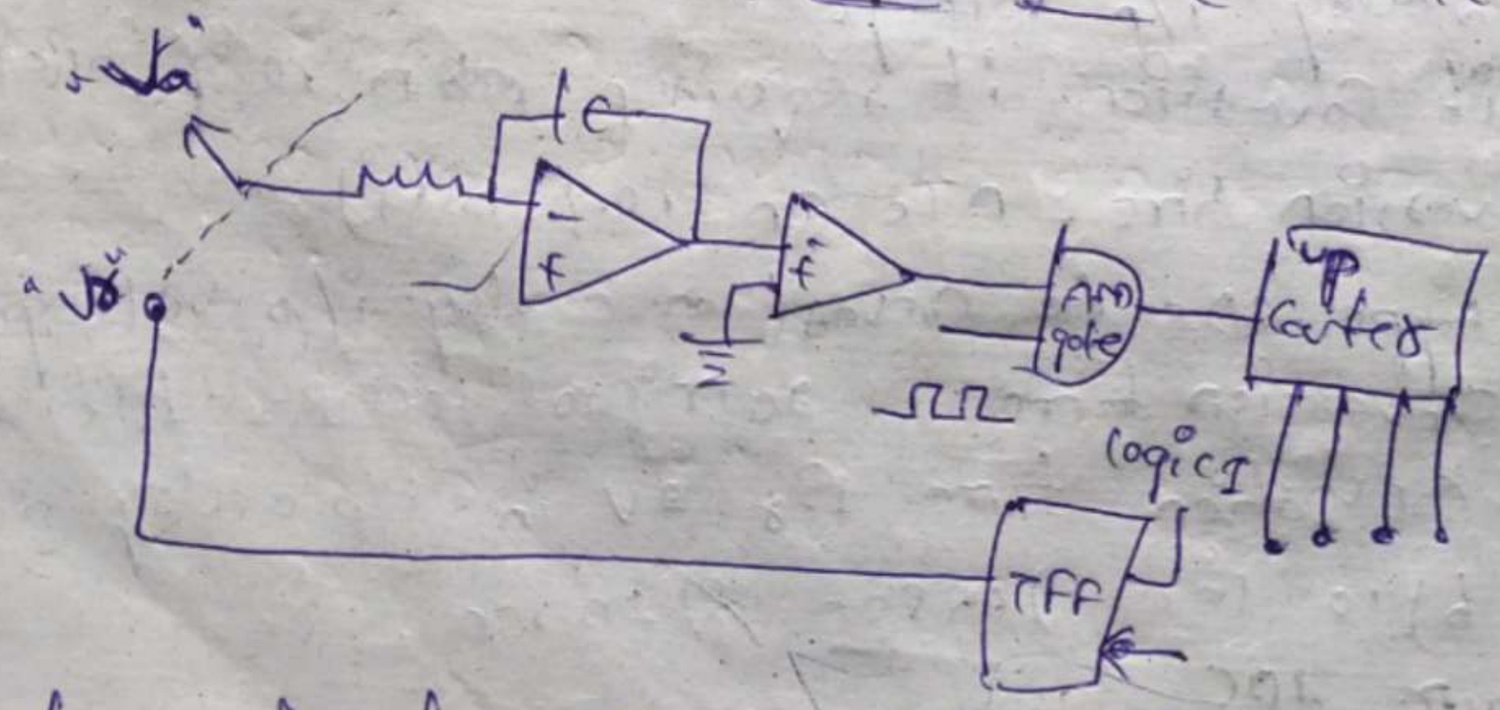


→ for n-bit conversion we need $2^n - 1$ comparator
→ for n-bit conversion we need 2^n no. of resistors
→ It is more sensitive circuit since, we are increasing the no. of comparators for each bit increment

Ex: n=3 → if we increase n=4
no. of comparator $= 2^3 - 1 = 7$
no. of comparators $= 2^4 - 1 = 15$

→ The additional no. of comparators required for 1 bit increment is 2^n no. of comparators. $15 - 7 = 8 \rightarrow 2^n [n=3]$
→ The total conversion time (T_c) in the ADC is $1 T_c$

Dual slope integrating type ADC (accurate ADC)



→ As shown in above fig an integrator is used at the i/p side. which is used to integrate both the analog voltage & reference voltage

→ In this we are using a T-FF & a counter. This T-FF is used to toggle the connected b/w analog voltage & the reference voltage.

→ whenever the analog i/p is connected the o/p of integrator is negative voltage bcoz of this the o/p of next comparator is logic 1.

→ which enables the AND logic gate so, the counter values will start increasing (whenever clk pulse is applied)

→ After 2^n clk pulses the counter will again reaches the zero (t_1) at this time the T-FF toggles to logic 1 & switch is connected to the reference voltage

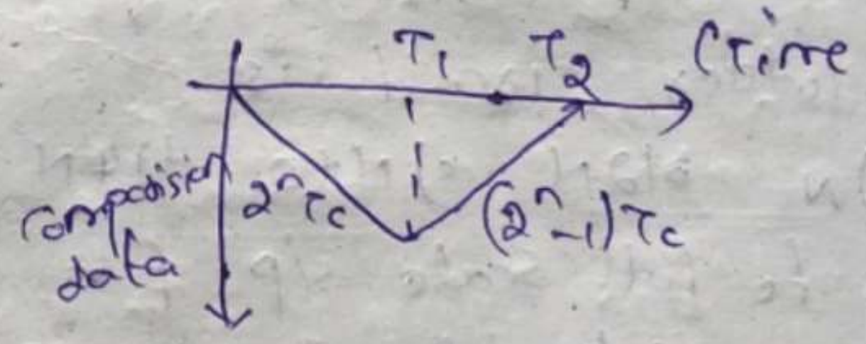
→ During this reference voltage integration the o/p of integrator is negative up to time T_2 bcoz of this the counter will again ~~st~~ count the clk pulses

→ At this point T_2 the o/p of integrator positive voltage therefore, the comparator o/p is logic "0"

→ The total time conversion in dual slope integrating type ADC is $\Rightarrow 2^n T_c + (2^n - 1) T_c$

$\Rightarrow 2 \cdot 2^n T_c$

$= 2^{(n+1)} T_c$.



Note:

→ It is having highest conversion time among all ADC's

→ It is most accurate ADC which is mostly used in digital volt meter

Summary:

Type of ADC	max Conversion Time (T_c)
1) Counter	$(2^n - 1) T_c$
2) SA DAC	$n T_c$
3) Flash DAC	$1 \cdot T_c \rightarrow$ fast
4) Dual slope integrating ADC	$2^{n+1} T_c \rightarrow$ slow

2019

1) The ADC used in digital voltmeters & multimeters is
a) counter ADC b) SA ADC c) Flash ADC d) Dual slope ADC

2) Find out the integrated type ADC
a) Flash type b) counter ADC c) Dual slope d) SA ADC

3) An n-bit D-A converter gives a max voltage of then the resolution is
a) V_{max} b) $V_{max}/2^n-1$ c) V_{max}^2

4) The minimum no. of comparators required to build 8-bit flash ADC
A) 255 [$\because 2^8-1 \Rightarrow 255$]

5) The main disadvantage of Dual slope ADC is
a) slow sensitivity b) high cost c) slow conversion time
d) temp immunity

6) The resolution of 12-bit DAC with full scale voltage of 10V is
a) 2.44 millivolts b) 8.33 mV c) 1.2 mV d) 10V

$\Rightarrow \frac{V_r}{2^n-1} \Rightarrow \frac{10}{2^{12}-1} \Rightarrow \frac{10}{4095} =$

7) The no. of resistors required for n-bit DAC in R-2R ladder circuit is
a) N b) 2N c) N+2 d) 4N

8) The full scale o/p of a DAC is 20 milliampere. if the resolution is 80 microampere then the percentage of reduction is.
a) 8 b) 7 c) 3.9 d) 0.39

$\frac{1}{2^n-1} \times 100 \Rightarrow$

Memories

→ The memories are classified into 2 types :
1) primary memory
2) Secondary memory

→ primary memories are basically very less storage elements but easy to access the data (faster)
Ex :- RAM, PROM

→ These both devices are used for random access

→ Secondary memories are used for storing latch data
Ex :- Hard disk, CD, DVD, Pen Drive

→ The speed of access is less often compared to primary memories

RAM :

→ It is a type of memory in which the all address locations are accessible in an equal amount of time & can be accessed in any order (Random)

→ We can do both read & write operations

→ RAM looses the data wherever the power is turned off that's why it is called a volatile memory

ROM :

→ It is a type of memory in which the data is stored permanently / semi permanently

→ We can read the data but we can't write

→ It is also having random access capability

→ It can store the information even the power is turned off that's why it is called non-volatile memory

Types of RAM :

Static RAM

→ All static RAM's are characterised by flipflops (or) storage cells

• Cache memory :-

→ It is relatively small & high speed memory which stores most recently used instruction

Dynamic RAM

→ It can store a data bit in a small capacitor

Ex :- MOSFET

Disadvantages

→ The storage capacitor can't hold its charge over in extended period of time. so, we need to refresh periodically in order to store the data

Types of ROM

1) Maskable ROM
2) programmable ROM
3) E-PROM
4) EE-PROM
5) UV PROM

1) Maskable ROM :-

→ It is the one in which the manufacturer programs the data which can't be changed

• PROM (programmable ROM)

→ It is similar to Maskable ROM but the program can be done by the user itself.

→ we can only write one time but can be read multiple times.

EPROM (Erasable programmable ROM)

→ It can be reprogrammed multiple times only when the initial data is erased

EEPROM (electrically erasable programmable ROM)

→ In this ROM we can erase & program the data by using electrical pulses.

UVPROM

→ we can erase the data by exposing the memory chip to high intensity UV radiations

Flash memory

→ these are high density read/write memories that are non-volatile

capacitor

age over in
refresh

M

programs

program

multiple

only

le ROM)

e data

rosy

that

Q) the storage element for a static RAM is
a) Diode b) Capacitor c) resistor d) FF.

Q) Cache memory uses
a) SRAM b) DRAM c) magnetic memory d) ROM

Q) One of the advantages DRAM is
a) it needs to be refresh periodically
b) it is small & acquires less space
c) it has a large access time
d) it makes use of MOSFET

Q) A PROM can be program only once after pgming its contents are
a) permanently fix b) temporarily fix
c) can be stored only once d) can be change a fixed no. of times.

Q) the no. of memory IC's of capacity 1K×4 require to construct 8K×8 memory chip is

$$\frac{8K\times 8}{1K\times 4} \rightarrow \frac{8\times 2^{10}\times 8}{1\times 2^{10}\times 4} = \frac{64}{4} = 16$$

→ A memory chip can be represented by $2^m \times n$
m = no. of address bits
n = no. of data bits in each address location

→ the total memory capacity is defined based on the total no. of bits stored in a memory chip

Ex: 64K memory chip
1) 16×4 → $2^4 \times 4$ → 16 nibbles
2) 32×2 → $2^5 \times 2$ →
3) 8×8 → $2^3 \times 8$ → 8 bytes.

Q) the static RAM has 15 address lines & 4 data lines then which of the following is
a) 32K byte memory c) 16K byte memory
b) 32K nibble memory d) 16K nibble memory

Sol: no. of address lines = 15
data lines = 4

$2^m \times n$

$= 2^{15} \times 4$

$= 2^{10} \times 2^5 \times 4$

$= 32K$ nibble memory

→ To increase the bit capacity or length of each memory location the memory IC's should be connected in parallel.

→ To design a DRAM generally 1 or 2 transistors are used serially. to design SRAM 6 transistors are used

logic families

→ there are 2 types:
1) Unipolar logic family 2) bipolar logic family.

1) Unipolar logic families:

→ Unipolar means only one type of carriers are present.

Ex: NMOS, CMOS, DMOS

2) Bipolar logic families

→ Bipolar means two type of carriers are present
1) Majority carriers 2) Minority carriers

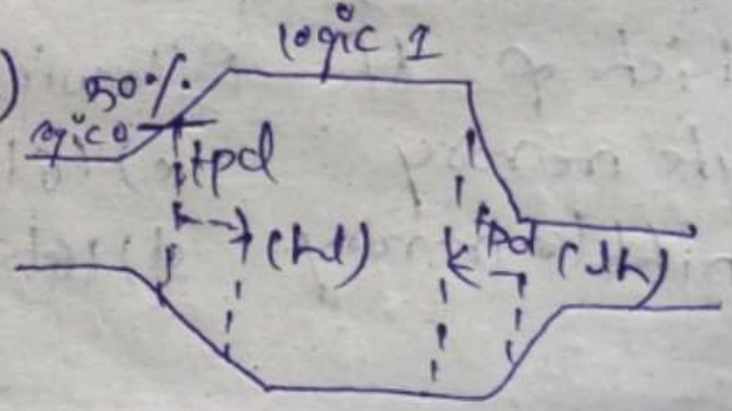
classification of Bipolar logic families

1) Saturated logic family
Ex: TTL, DTL and I^2L

2) unsaturated logic family
Ex: ECL, schottky

propagation delay:

$$t_{pd} = \frac{t_{pd}(hl) + t_{pd}(lh)}{2}$$



→ It is time taken by any logic gate to change its o/p with respect to changes in i/p.

Power dissipation (P_D)

$$P_D = V_{cc} \times T_c (avg)$$

→ It is the amount of power consumed by a gate when all the i/p's are fully present (logic 1)

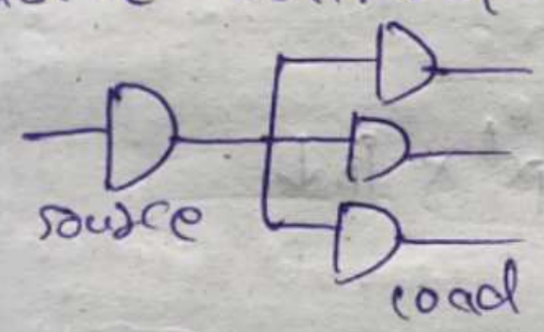
Figure of merit :

→ generally any logic family with flow figure of merit is referred

$$FOM = t_{pd} \times P_D$$

Fan out

→ The no. of standard gates/loads that o/p of a gate can drive without disturbing its normal operation



Eg.: fan out for high load

$$(fan\,out)_H = \frac{I_{OH}}{I_{IH}} \quad (fan\,out)_L = \frac{I_{OL}}{I_{IL}}$$

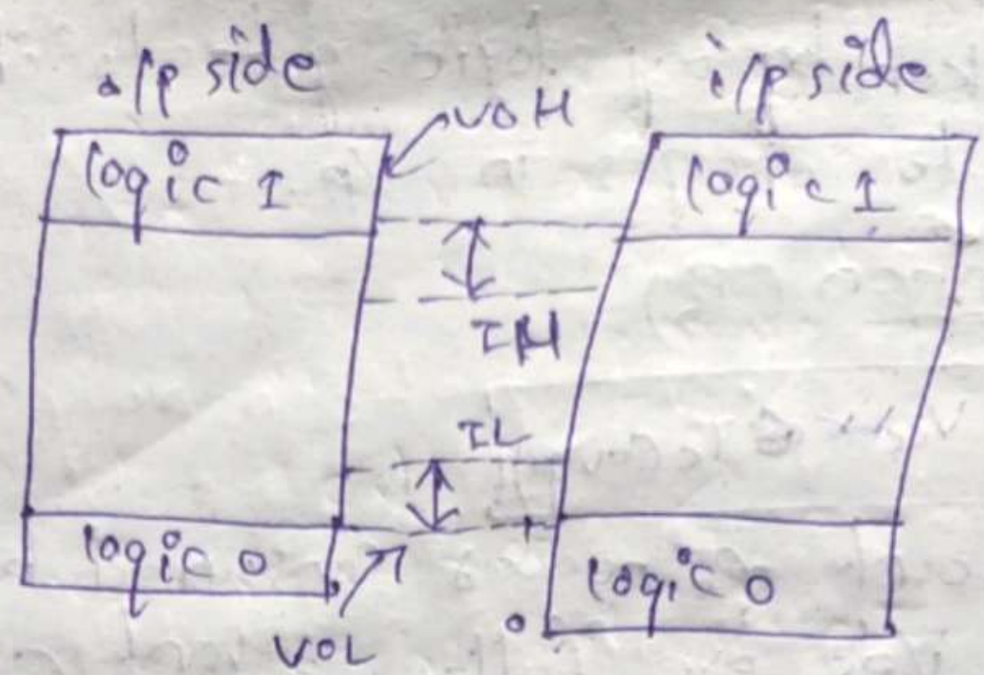
$$fan\,out = \min[FO_H, FO_L]$$

Fan In :

→ The maximum no. of i/p's that are connected to a logic gate without degradation of i/p voltage levels

Noise Margin :

→ It is the minimum noise voltage that can be applied to a logic gate i/p which will not affect the o/p.



→ Among all logic families ECL (emitter coupled logic) is the fastest one. Since, it is operated in ~~non~~ non-saturation region

→ ~~fo~~ fan out is very high in CMOS circuits

→ power dissipation is very less in CMOS circuits

→ Noise margin is also high in CMOS circuits

ECL	CMOS
FOM = ↓ tpd × PD ↑	FOM = ↑ tpd × PD ↓