

## Unit - 3

### Number System of logic Gates:-

\* A number system can be classified as.

1] Non-positional number system

→ In this number system a digit of a number does not indicate any significance of its position & weight.  
Eg. Roman number system

2] Positional number system

→ In this number system - the position of every digit of a number indicates the significance or weight to be attached to that digit. Eg. decimal number system

- Digital electronics systems are, positional number systems.

\* Representing a Number.

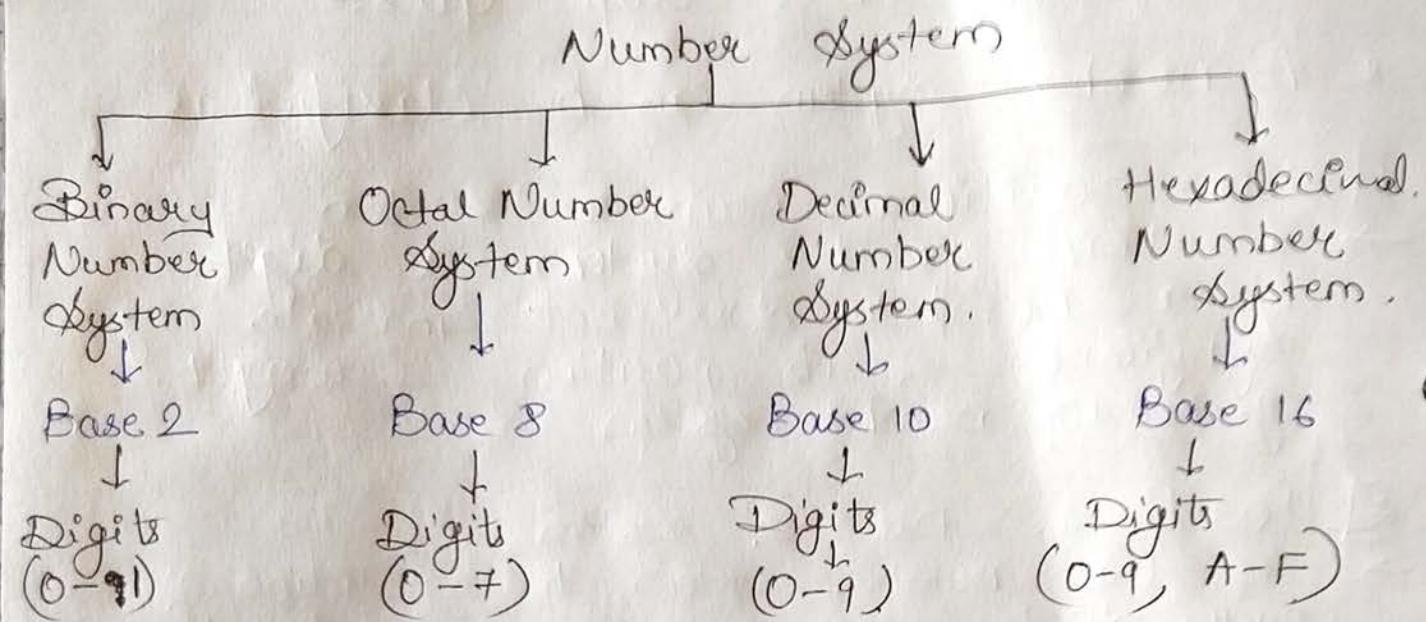
$$(N)_b = \underbrace{[d_{n-1} d_{n-2} \dots d_1 d_0]}_{\text{P.I.D}} \cdot \underbrace{d_{-1} d_{-2} \dots [d_{-m}]}_{\text{Fractional part}} \xrightarrow{\text{decimal or radix point.}}$$

LSD

Where,  $N$  = Number

$b$  = base or radix

## \* Types of Number System:-



## \* Binary Number System.

- A Binary digit is called as bit.  
E.g. 0, 1.
- A group of 4 bits is called nibble.  
Eg. 0101, 1000
- A group of 8 bits is called Byte  
Eg. ~~1000~~ 10011010. [group of 2 nibbles]
- A group of 16 bits [or group of 4 nibbles or 2 bytes] is called as Word.
- A group of 32 bits or 2 words or 4 bytes or 8 nibbles is called Double word.

1) Binary to Decimal Conversion.

Ex1:  $(110010)_2 = (?)_{10}$

$$\begin{array}{ccccccc}
 & 1 & 1 & 0 & 0 & 1 & 0 \\
 \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 & \\
 \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\
 1 \times 2^5 & 1 \times 2^4 & 0 \times 2^3 & 0 \times 2^2 & 1 \times 2^1 & 0 \times 2^0 & \\
 \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\
 32 + 16 + 0 + 0 + 2 + 0 & & & & & & \\
 \underbrace{\hspace{10em}} & & & & & & \\
 = 50 & & & & & & \\
 \end{array}$$

$\therefore (110010)_2 = (50)_{10}$

Ex2:  $(11011.101)_2$

$$\begin{array}{ccccccc}
 & 1 & 1 & 0 & 1 & 1 & . & 1 & 0 & 1 \\
 \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow \\
 2^4 & 2^3 & 2^2 & 2^1 & 2^0 & & 2^{-1} & 2^{-2} & 2^{-3} \\
 \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow \\
 1 \times 2^4 & 1 \times 2^3 & 0 \times 2^2 & 1 \times 2^1 & 1 \times 2^0 & & 1 \times 2^{-1} & 0 \times 2^{-2} & 1 \times 2^{-3} \\
 \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow \\
 16 + 8 + 0 + 2 + 1 & & & & & & 0.5 + 0 + 0.125 & & \\
 \underbrace{\hspace{10em}} & & & & & & \underbrace{\hspace{10em}} & & \\
 & & & & & & & & \\
 & 27 & & + & & & 0.625 & & \\
 & & & & & & & & \\
 = 27.625 & & & & & & & & \\
 \end{array}$$

$\therefore (11011.101)_2 = (27.625)_{10}$

2] Octal to Decimal Conversion.

Ex 1.  $(45)_8 = (?)_{10}$

$\Rightarrow$

$$\begin{array}{r} 4 \ 5 \\ \downarrow \quad \downarrow \\ 8^1 \quad 8^0 \\ \downarrow \quad \downarrow \end{array}$$

$$\begin{aligned} & 4 \times 8^1 + 5 \times 8^0 \\ & = 32 + 5 \\ & = 37 \end{aligned}$$

$$\therefore (45)_8 = (37)_{10}$$

Ex. 2.  $(134.06)_8 = (?)_{10}$

$\Rightarrow$

$$\begin{array}{r} 1 \ 3 \ 4 \\ \downarrow \quad \downarrow \quad \downarrow \\ 8^2 \quad 8^1 \quad 8^0 \\ \downarrow \quad \downarrow \quad \downarrow \end{array}$$

$$1 \times 8^2 + 3 \times 8^1 + 4 \times 8^0$$

$$\underbrace{64 + 24 + 4}$$

$$.92$$

$$= 92.09375$$

$$\begin{array}{r} 0 \ 6 \\ \downarrow \quad \downarrow \\ 8^{-1} \quad 8^{-2} \\ \downarrow \quad \downarrow \end{array}$$

$$0 \times 8^{-1} + 6 \times 8^{-2}$$

$$\underbrace{0 + 0.09375}$$

$$0.09375$$

$$\therefore (134.06)_8 = (92.09375)_{10}$$

3] Hexadecimal

→ to decimal

Ex 1.  $(AC.2B)_{16} = (?)_{10}$

$$\begin{array}{ccccccc}
 4 & & C & . & 2 & & B \\
 \downarrow & & \downarrow & & \downarrow & & \downarrow \\
 16^1 & & 16^0 & & 16^{-1} & & 16^{-2} \\
 \downarrow & & \downarrow & & \downarrow & & \downarrow \\
 4 \times 16^1 + C \times 16^0 + 2 \times 16^{-1} + B \times 16^{-2} \\
 12 \times 16^0 & & & & & & 11 \times 16^{-2}
 \end{array}$$

$$\begin{aligned}
 &= 64 + 12 + 0.125 + 0.0429 \\
 &= (76.1679)_{10}
 \end{aligned}$$

Ex 2.  $(3A.2F)_{16} = (?)_{10}$

$$\begin{array}{ccccccc}
 3 & & A & . & 2 & & F \\
 \downarrow & & \downarrow & & \downarrow & & \downarrow \\
 16^1 & & 16^0 & & 16^{-1} & & 16^{-2} \\
 \downarrow & & \downarrow & & \downarrow & & \downarrow \\
 3 \times 16^1 & & A \times 16^0 & & 2 \times 16^{-1} & & F \times 16^{-2} \\
 & & 10 \times 16^0 & & & & 15 \times 16^{-2}
 \end{array}$$

$$\begin{aligned}
 &= 48 + 10 + 0.125 + 0.0585 \\
 &= (58.71)_{10}
 \end{aligned}$$

$$\therefore (3A.2F)_{16} = (58.71)_{10}$$

4] Octal to Binary

Ex 1.  $(456.732)_8 = (?)_2$

$$\begin{array}{ccccccccc}
 4 & & 5 & & 6 & & \cdot & 7 & 3 & 2 \\
 \downarrow & & \downarrow & & \downarrow & & & \downarrow & \downarrow & \downarrow \\
 100 & & 101 & & 110 & & \cdot & 111 & 011 & 010
 \end{array}$$

$$= (100101110 \cdot 111011010)_2$$

Ex 2.  $(532.125)_8 = (?)_2$

$$\begin{array}{ccccccccc}
 5 & & 3 & & 2 & & \cdot & 1 & 2 & 5 \\
 \downarrow & & \downarrow & & \downarrow & & & \downarrow & \downarrow & \downarrow \\
 101 & & 011 & & 010 & & \cdot & 001 & 010 & 101
 \end{array}$$

$$= (101011010.001010101)_2$$

Octal digit	3 bit equivalent binary
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

5

Binary  $\rightarrow$  Octal

Ex1.  $[100101111.010111011]_2 = (?)_8$

$$\begin{array}{r} 100 \quad 101 \quad 111 \quad \cdot \quad 010 \quad 111 \quad 011 \\ \downarrow \quad \downarrow \quad \downarrow \quad \quad \downarrow \quad \downarrow \quad \downarrow \\ 4 \quad 5 \quad 7 \quad \cdot \quad 2 \quad 4 \quad 3 \end{array}$$

$$= (457.273)_8$$

Ex2:  $(101011101)_2 = (?)_8$

$$\Rightarrow \begin{array}{r} 010 \quad 101 \quad 101 \\ \downarrow \quad \downarrow \quad \downarrow \\ 2 \quad 5 \quad 5 \end{array}$$

Added bit  $\leftarrow$

$$= (255)_8$$

6

Binary  $\rightarrow$  Hexa decimal

Ex1.  $(0010111100.01101101)_2 = (?)_{16}$

$$\Rightarrow \begin{array}{r} 0010 \quad 1111 \quad 1100 \quad \cdot \quad 0110 \quad 1101 \\ \downarrow \quad \downarrow \quad \downarrow \quad \quad \downarrow \quad \downarrow \\ 2 \quad F \quad C \quad \cdot \quad 6 \quad D \end{array}$$

$$\Rightarrow (2FC.6D)_{16}$$

Hexadecimal digit	4bit binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Ex. 2.  $(101011 \cdot 111011)_2 = (?)_{16}$

→  $\underbrace{0010}_{2} \underbrace{1011}_{B} \cdot \underbrace{1110}_{E} \underbrace{1100}_{C}$

$= (2B.EC)_{16}$

7] Hexadecimal → Binary

Ex 1.  $(3A9D.A0C)_{16} = (?)_2$

$$\begin{array}{ccccccc}
 3 & A & 9 & D & . & A & 0 & C \\
 \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow \\
 0011 & 1010 & 1001 & 1101 & . & 1010 & 0000 & 1100
 \end{array}$$

$$= (0011101010011101.101000001100)_2$$

Ex 2.  $(8A9.B4)_{16} = (?)_2$

$$\begin{array}{ccccccc}
 8 & A & 9 & . & B & 4 \\
 \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\
 1000 & 1010 & 1001 & . & 1011 & 0100
 \end{array}$$

$$= (100010101001.10110100)_2$$

8] Octal → Hexadecimal

Ex 1.  $(423.613)_8 = (?)_{16}$

$$\begin{array}{ccccccc}
 \rightarrow & 4 & 2 & 3 & . & 6 & 1 \cancel{8} 3 \\
 & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \cancel{\$} \downarrow \\
 \text{Added} & \underbrace{000100}_1 & \underbrace{010}_1 & \underbrace{011}_3 & . & \underbrace{110001}_5 & \underbrace{011000}_8
 \end{array}$$

$$\begin{array}{ccccccc}
 & 0001 & 0001 & 0011 & . & 1100 & 0101 & 1000 \\
 & \downarrow & \downarrow & \downarrow & . & \downarrow & \downarrow & \downarrow \\
 & 1 & 1 & 3 & . & C & 5 & 8
 \end{array}$$

$$(423.613)_8 = (113.C58)_{16}$$

3 bit Bin

4 bits Grp

Hexadecimal  
Equivalent

$$\text{Ex 2: } (615.25)_8 = (?)_{16}$$

$\begin{array}{r} 6 \quad 1 \quad 5 \\ \downarrow \quad \downarrow \quad \downarrow \\ 110 \quad 001 \quad 101 \\ \downarrow \quad \downarrow \quad \downarrow \\ 000 \quad 1000 \quad 1101 \\ \downarrow \quad \downarrow \quad \downarrow \\ 1 \quad 8 \quad D \end{array}$	$\cdot$ $\begin{array}{r} 2 \quad 5 \\ \downarrow \quad \downarrow \\ 010 \quad 101 \\ \downarrow \quad \downarrow \\ 0101 \quad 0100 \\ \downarrow \quad \downarrow \\ 5 \quad 4 \end{array}$	<span style="margin-right: 20px;">3 bit Binary</span> <span>4 bit Grouping</span>
---	--	---

$$(615.25)_8 = (18D.54)_{16}$$

9] Hexadecimal  $\rightarrow$  Octal Conversion.

$$\text{Ex 1. } (BC2.4C)_{16} = (?)_8$$

$\Rightarrow$

$\begin{array}{r} B \quad C \quad 2 \quad . \quad 4 \quad C \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ 1011 \quad 1100 \quad 0010 \quad . \quad 0100 \quad 1100 \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ 101 \quad 111 \quad 000 \quad 010 \quad . \quad 010 \quad 011 \quad 000 \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ 5 \quad 7 \quad 0 \quad 2 \quad . \quad 2 \quad 3 \quad 0 \end{array}$	$\begin{array}{r} \text{Added bit} \\ \text{4 bit Binary} \\ \text{8 bits Grouping} \\ \text{Octal Equivalent} \end{array}$
--	---

$$= 5702.230$$

$$\Rightarrow (BC2.4C)_{16} = (5702.230)_8$$

$$\underline{\text{Ex2:}} \quad (A6F \cdot C9)_{16} = (?)_8$$

A      6      F      .      C      9  
 ↓      ↓      ↓           ↓      ↓  
 100    0110    1111      •      1100    10010      4 bit  
 ↓      ↓      ↓           ↓      ↓      Binary  
 101    001    101    111      •      110    010    010      3 bit Group  
 ↓      ↓      ↓           ↓      ↓      ↓  
 5    1    5    7      .      6    2    2

=  $(5157.622)_8$

## Decimal to Binary

$$\text{Ex 1. } (26)_{10} = (?)_2$$

<u>Base.</u>	<u>Quotient</u>	<u>Remainder</u>
2	26	
2	13	0 (LSB)
2	06	1
2	03	0
	(01)	1 (MSB)

$$(26)_{10} = (11010)_2$$

$$\text{Ex 2: } (155.33)_{10} = (?)_2$$

Base	Quotient	Reminder
2	155	LSB
2	77	1
2	38	1
2	19	0
2	9	1
2	4	1
2	2	0
2	1	0
		1

(10011011)  
MSB

Decimal fraction      Base      Product      Carry

0.33	$\times 2$	$= \overbrace{0.66}^0$	MSB
0.66	$\times 2$	$= \overbrace{1.32}^1$	
0.32	$\times 2$	$= \overbrace{0.64}^0$	
0.64	$\times 2$	$= \overbrace{1.28}^1$	
0.28	$\times 2$	$= \overbrace{0.56}^0$	
0.56	$\times 2$	$= \overbrace{1.12}^1$	LSB

$$(155.33)_{10} = (10011011.010101)_2$$

### III Decimal $\rightarrow$ Octal

Ex 1.  $(128)_{10} = (?)_8$

Division	Quotient	Remainder
$\frac{128}{8}$	16	0
$\frac{16}{8}$	2	0

LSB ↑  
↓ 2  
MSB

$$(128)_{10} = (200)_8$$

Ex 2.  $(199.3125)_{10} = (?)_8$

Division	Quotient	Remainder
$\frac{199}{8}$	24	7
$\frac{24}{8}$	3	0

LSB ↑  
↓ (307)<sub>8</sub>

Decimal fraction Base Product Carry  
 $0.3125 \times 8 = 0.25$  3  
 $0.5 \times 8 = 4.0$  1  
 $(0.24)_8$  (LSB)

$$(199.3125)_{10} = (307.24)_8$$

12] Decimal  $\rightarrow$  Hexadecimal

Ex1.  $(2156)_{10} = (?)_{16}$

Division      Quotient      Remainder.

$$\begin{array}{r}
 \underline{2156} \\
 16 \\
 \hline
 134 \\
 \underline{16} \\
 \hline
 8
 \end{array}
 \quad
 \begin{array}{l}
 134 \\
 \textcircled{8} \longrightarrow 6 \\
 8 \text{ (MSB)}
 \end{array}
 \quad
 \begin{array}{l}
 12 \rightarrow C \text{ (LSB)} \\
 \uparrow
 \end{array}$$

$$(2156)_{10} = (86C)_{16}$$

Ex2.  $(199.375)_{10} = (?)_{16}$

Division      Quotient      Remainder.

$$\begin{array}{r}
 \underline{199} \\
 16 \\
 \hline
 12 \\
 \underline{16} \\
 \hline
 0
 \end{array}
 \quad
 \begin{array}{l}
 \textcircled{12} \longrightarrow 12 \text{ (MSB)} \rightarrow C \\
 7 \text{ (LSB)}
 \end{array}$$

Decimal fraction      Base      Product      Carry

$$0.375 \times 16 = \overbrace{6.00}^6$$

$$\therefore (199.375)_{10} = (C7.6)_{16}$$

## \* Binary Arithmetic.

### 1. Binary Addition.

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

Ex 1. Add  $(5)_{10}$  and  $(3)_{10}$

$$(5)_{10} = (0101)_2 ; (3)_{10} = (0011)_2$$

Carry  $\rightarrow$

1	1	1	0	0	0
0	1	0	1		
+	0	0	1	1	
	1	0	0	0	

$$(1000)_2 = (8)_{10}$$

$$\therefore (5)_{10} + (3)_{10} = (8)_{10}$$

Ex 2.  $(11011.11)_2 + (11011.01)_2 = (?)_2$

Carry  $\rightarrow$

1	1	1	0	1	1	1	1	1
1	1	0	1	1	.	1	1	
+	1	1	0	1	1	.	0	1
	1	0	1	1	1	.	0	0

final carry      final sum

Ans:  $(110111.00)_2$

## \* Binary Subtraction.

A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$Ex 1. \quad (7)_{10} - (5)_{10} = 2$$

$$(0111)_2 - (0101)_2$$

$$\begin{array}{r} 0 \quad 1 \quad 1 \quad 1 \\ 0 \quad 1 \quad 0 \quad 1 \\ \hline 0 \quad 1 \quad 1 \quad 1 \end{array} \quad (\text{no})$$

$$\text{Ans: } (00010)_2 = (2)_{10}$$

$$\underline{\text{Ex2.}} \quad (1011.010)_2 - (0110.101)_2$$

$$\begin{array}{r}
 & 1 & 0 & 1 & 1 & \cdot & 0 & 1 & 0 & 1 \\
 - & 0 & 1 & 1 & 0 & \cdot & 1 & 0 & 1 \\
 \hline
 & \boxed{-1} & & \boxed{-1} & & & \boxed{-1} & & \\
 & 0 & 1 & 0 & 0 & \cdot & 1 & 0 & 1
 \end{array}$$

final  
borrow

## # Binary Multiplication

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

Ex1.  $1011 \times 0101 = ?$

$$\begin{array}{r}
 & 1011 \\
 \times & 0101 \\
 \hline
 & 1011 \\
 & 0000 \times \\
 & 1011 \times \times \\
 \hline
 & 0000 \times \times \times \\
 \hline
 & 0110111
 \end{array}$$

Ans.  $(0110111)_2$

Ex2.  $101.01 \times 11.01$

$$\begin{array}{r}
 & 101.01 \\
 \times & 11.01 \\
 \hline
 & 10101 \\
 & 00000 \times \\
 & 10101 \times \times \\
 \hline
 & 10101 \times \times \times \\
 \hline
 & 10001.0001
 \end{array}$$

Ans =  $(10001.0001)_2$

## \* Binary Division

$$\begin{array}{r} 0 \div 1 = 0 \\ 1 \div 1 = 1 \end{array}$$

Ex 1.  $1001 \div 10$

$$\begin{array}{r} 100.1 \\ 10 \overline{)1001} \\ -10 \\ \hline 00 \\ -0 \\ \hline 01 \\ -0 \\ \hline 00 \end{array}$$

Ans :-  $(100.1)_2$

2]  $(11.11)_2 \div (11)_2$

$$\begin{array}{r} 1.01 \\ 11 \overline{)11.11} \\ -11 \\ \hline 00 \\ -0 \\ \hline 11 \\ -11 \\ \hline 00 \end{array}$$

Ans.  $(1.01)_2$

## LOGIC Gates :-

→ In 1850s - the British mathematician George Boole developed a mathematical system for solving logic statements. This is called Boolean algebra & it is helpful in - the design & analysis of digital system.

### \* Classification of Logic gates:-

Logic gates:-

Basic  
→ NOT  
→ AND  
→ OR

Universal  
→ NAND  
→ NOR

Derived  
→ EX-OR  
→ EX-NOR

### \* NOT Gate:-

Symbol :- A

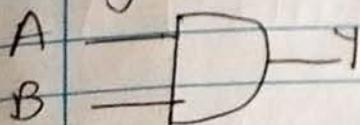


A	Y
0	1
1	0

Expression:-  $Y = \bar{A}$  → Logical inversion

### \* AND Gate:- o/p is high when both i/p are high.

Symbol :-



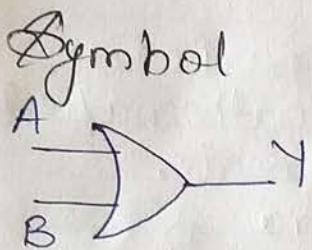
Expression

$$Y = A \cdot B$$

Logical Multiplication

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

\* OR Gate :- O/P is high when at least one i/p is high.



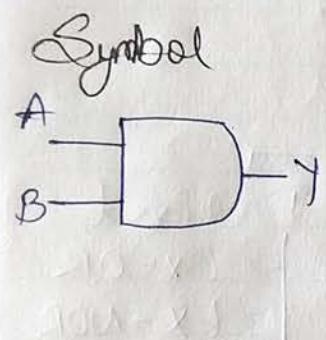
Expression

$$Y = A + B$$

logical Addition

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

\* NAND Gate :- O/P is low when both i/p high

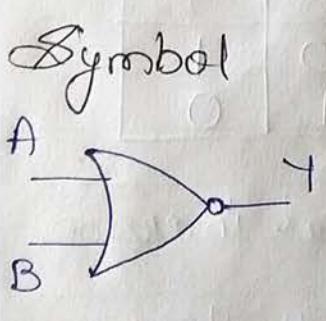


Expression

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

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

\* NOR Gate :- O/P is high when both i/p low

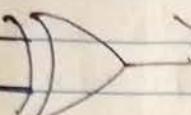


Expression

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

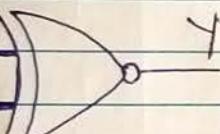
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

\* Ex-OR

Symbol	Expression
	$Y = A \oplus B$
	$Y = \overline{AB} + AB$

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

\* Ex-NOR

Symbol	Expression
	$Y = A \ominus B$ = $\overline{AB} + AB$ or $\overline{A \oplus B}$

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

\* IC Gates :-

IC Number

NOT	→	7404
AND	→	7408
OR	→	7432
NAND	→	7400
NOR	→	7402
EX-OR	→	7486
Ex-NOR	→	74266

## \* Application of Ex-OR Gate:-

- 1) As a magnitude Comparator
- 2) In binary to grey code converter.
- 3) In the adder of subtractor ckf.
- 4) In parity generator & checker ckf.
- 5) As a controlled subtractor.

## \* Application of Ex- NOR Gate:-

- 1) As even parity generator
- 2) As a Comparator
- 3) As even parity checker.

## \* Boolean Laws:-

### 1] Commutative Laws:-

$$A + B = B + A$$

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

### 2] Associative Laws:-

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

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

### 3] AND Laws:-

$$A \cdot 0 = 0$$

$$A \cdot A = A$$

$$A \cdot 1 = A$$

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

### 4) OR Laws:-

$$A + 0 = A$$

$$A + A = A$$

$$A + 1 = 1$$

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

### 5] Inversion Law:-

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

### 6] Distributive Law:-

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

### 7] Other Imp. Laws:- $A + B \cdot C = (A + B) \cdot (A + C)$

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

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

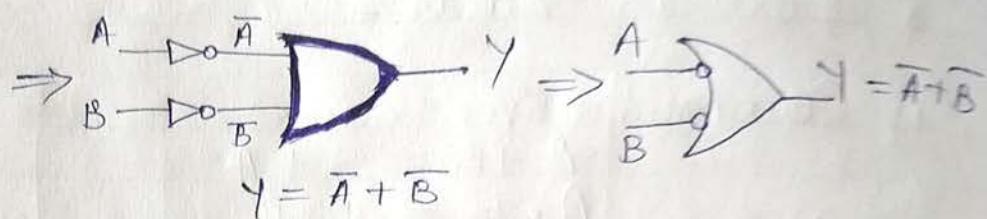
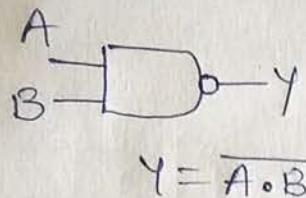
$$A + AB = A$$

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

# \* De-Morgan's Theorems:-

Theorem 1:  $\overline{A \cdot B} = \overline{A} + \overline{B}$

NAND = bubbled OR

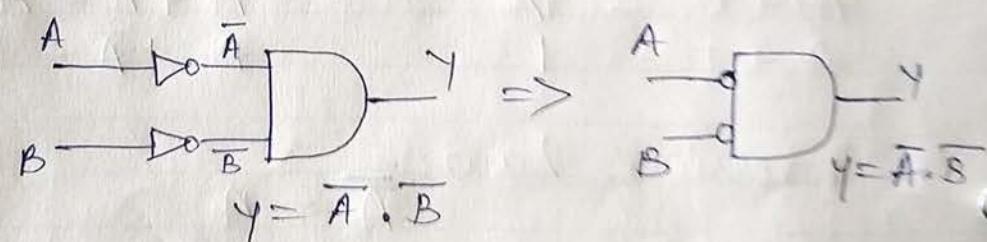
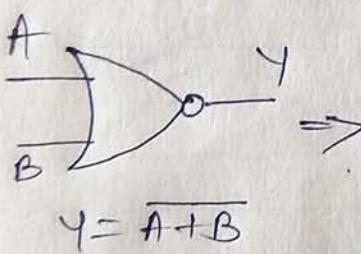


A	B	$\overline{A \cdot B}$	$\overline{A}$	$\overline{B}$	$\overline{A} + \overline{B}$
0	0	1	1	1	1
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	0	0	0

Theorem 2:

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

NOR = Bubbled AND



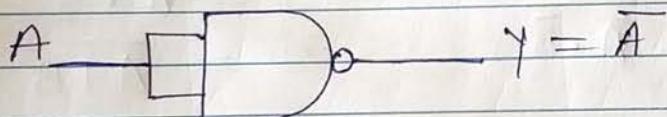
A	B	$\overline{A} + \overline{B}$	$\overline{A}$	$\overline{B}$	$\overline{A} \cdot \overline{B}$
0	0	1	1	1	1
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	0

## \* Universal Gates:-

- with the help of NAND & NOR gates we can implement any boolean expression. so, these gates are called as Universal Gates.
- user can build any logical circuit with the help of only NAND gates or only NOR Gates.

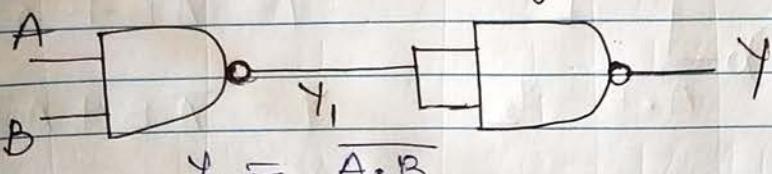
### 1) NOT gate using NAND Gate:-

$$\text{input} = A = B = A$$



$$\begin{aligned} \therefore Y &= \overline{A \cdot B} \\ &= \overline{A \cdot A} \\ &= \overline{A} \end{aligned}$$

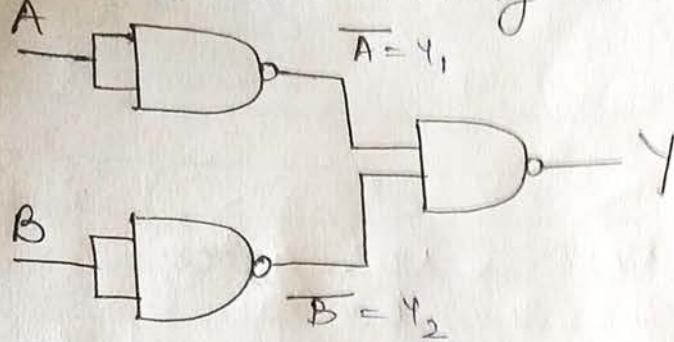
### 2) AND Gate using NAND Gate:-



$$Y_1 = \overline{A \cdot B}$$

$$Y = \overline{Y_1} = \overline{\overline{A \cdot B}} = A \cdot B$$

\* OR Gate using NAND Gate:-



$$Y_1 = \overline{A}, \quad Y_2 = \overline{B}$$

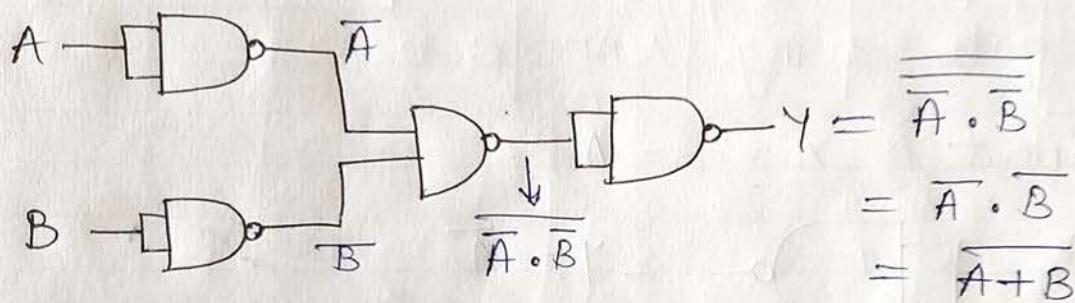
$$Y = \overline{Y_1 \cdot Y_2}$$

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

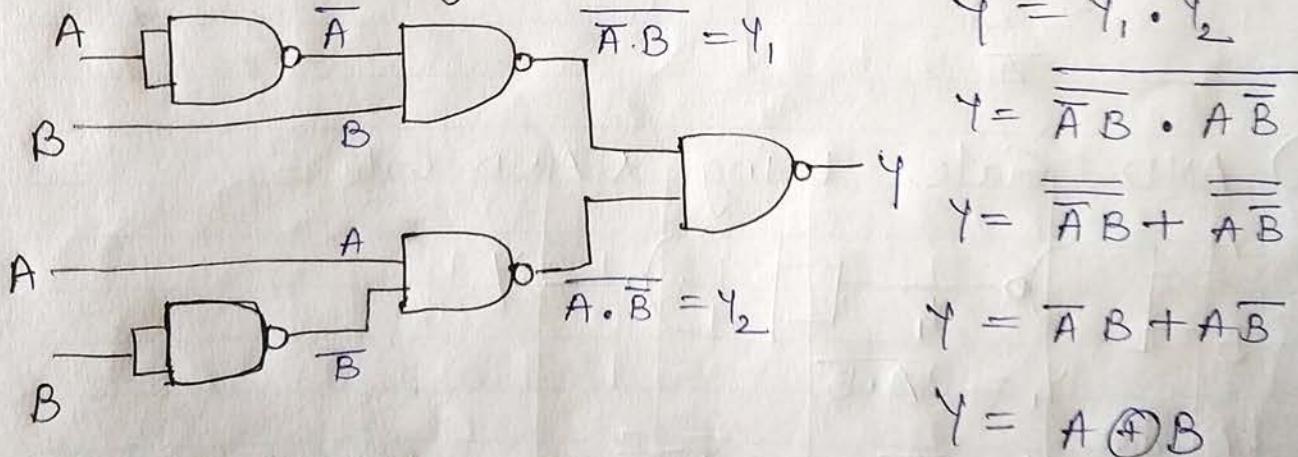
$$Y = \overline{\overline{A}} + \overline{\overline{B}}$$

$$Y = A + B$$

\* NOR Using NAND



\* Ex-OR Using NAND



$$Y = \overline{Y_1 \cdot Y_2}$$

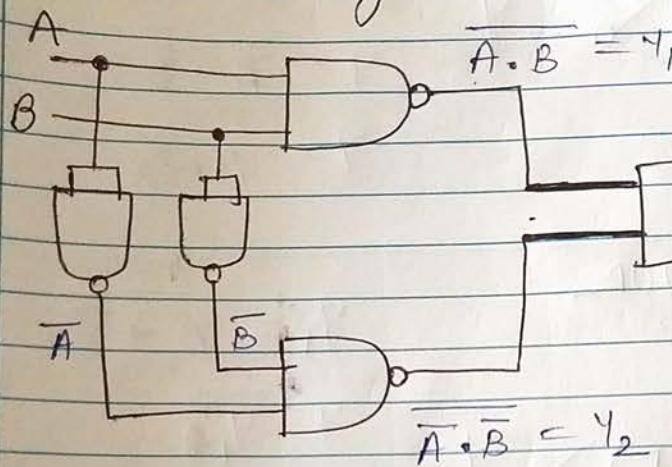
$$Y = \overline{\overline{A}B \cdot A\overline{B}}$$

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

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

$$Y = A \oplus B$$

\* EX-NOR using NAND



$$Y = \overline{Y_1 \cdot Y_2}$$

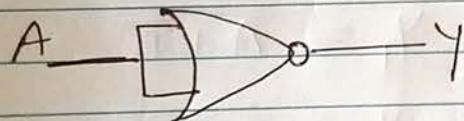
$$Y = \overline{A \cdot B} \cdot \overline{\overline{A} \cdot \overline{B}}$$

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

$$Y = A \odot B$$

\* NOR Gate as Universal gate.

1] NOT using NOR



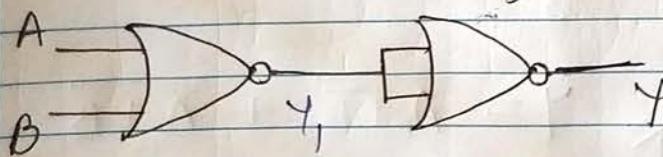
$$A = B = A$$

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

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

$$Y = \overline{A}$$

2] OR Gate using NOR



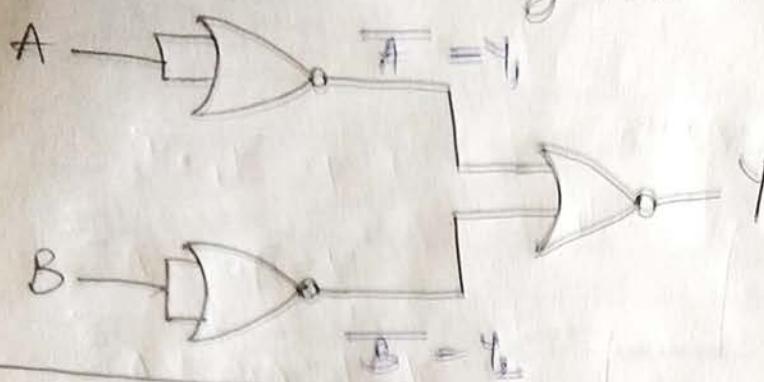
$$Y_1 = \overline{A + B}$$

$$Y = \overline{Y_1}$$

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

$$\therefore Y = A + B$$

3] AND Gate using NOR :



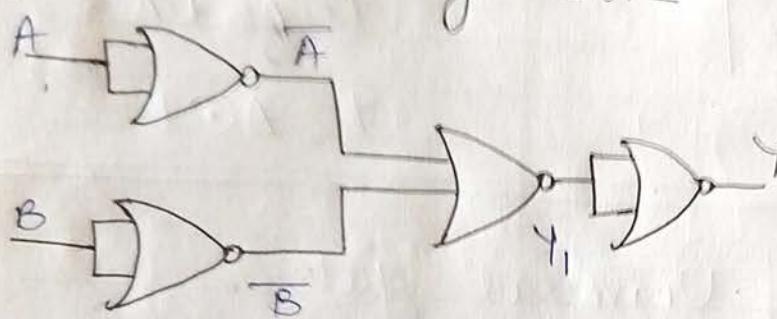
$$Y = \overline{Y_1 + Y_2}$$

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

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

$$Y = A \cdot B$$

4] NAND Using NOR



$$Y_1 = \overline{\bar{A} + \bar{B}}$$

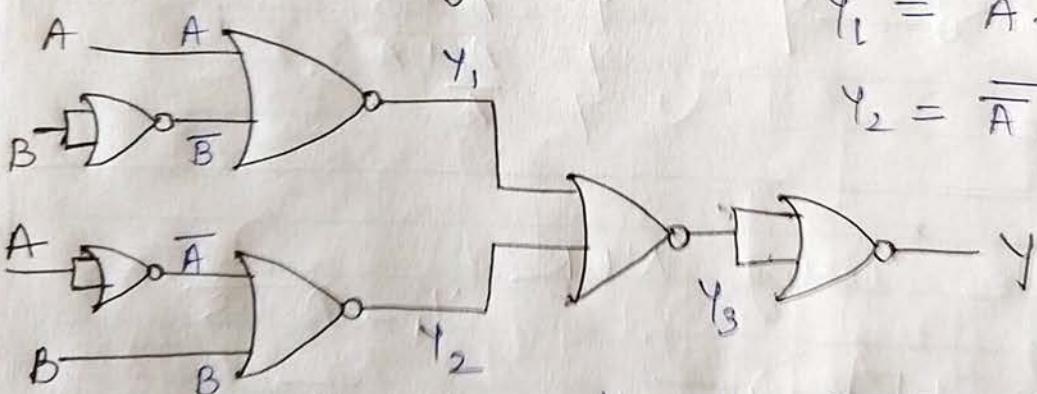
$$Y = \overline{Y_1 \cdot B}$$

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

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

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

5] EXOR Using NOR



$$Y_1 = \overline{A + B}$$

$$Y_2 = \overline{\bar{A} + B}$$

$$Y_3 = \overline{Y_1 + Y_2}$$

$$Y = \overline{\overline{Y_1 + Y_2}}$$

$$Y = Y_1 + Y_2$$

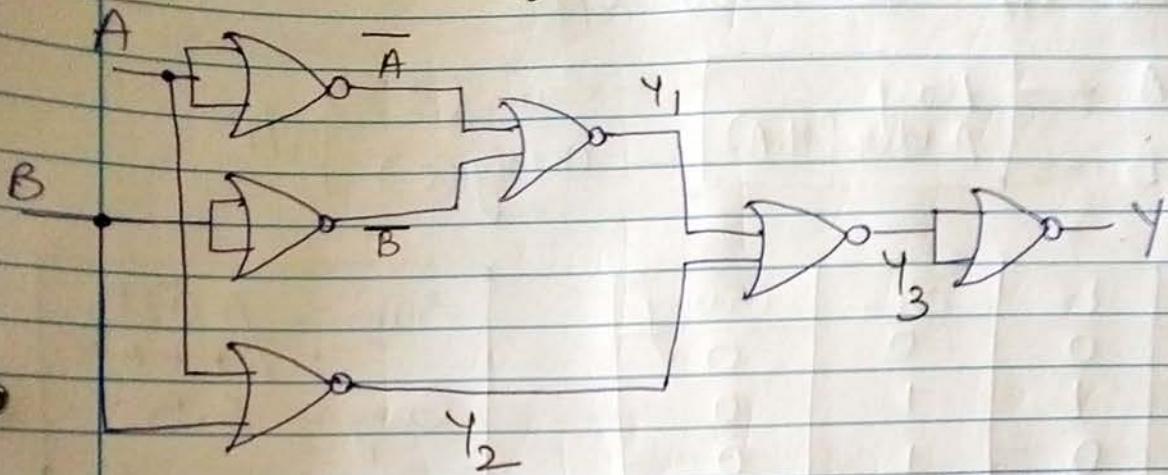
$$Y = \overline{\bar{A} + \bar{B}} + \overline{\bar{A} + B}$$

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

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

$$Y = A \oplus B$$

Ex-NOR Using NOR Gate:-



$$Y_1 = \overline{A + B}$$

$$Y_2 = \overline{\overline{A} + B}$$

$$Y_3 = \overline{Y_1 + Y_2}$$

$$Y = \overline{Y_3} = \overline{\overline{Y_1 + Y_2}} = Y_1 + Y_2$$

$$Y = Y_1 + Y_2$$

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

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

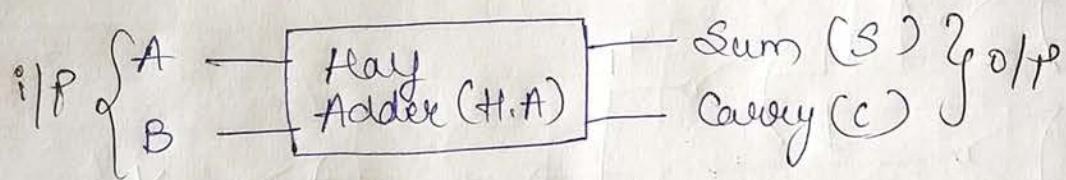
$$Y = AB + \overline{A} \cdot \overline{B}$$

$$Y = A \oplus B$$

\* Address:-

Half Adder

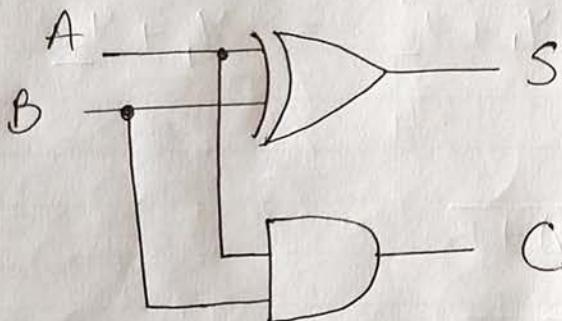
Combinational logic Ckt.  
two i/p & two o/p



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

$$\begin{aligned} \text{Sum} &= A \oplus B \\ &= \overline{A}B + A\overline{B} \\ \text{Carry} &= A \cdot B \end{aligned}$$

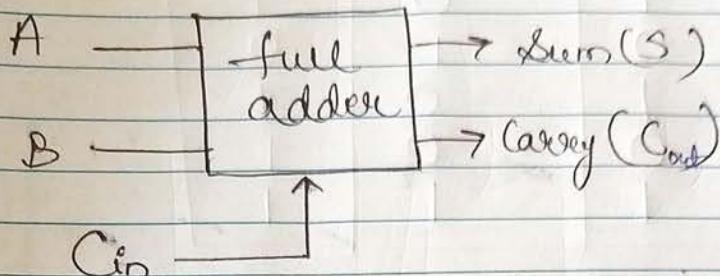
Logic diagram:-



\* Disadvantage:- it can perform 2-bit addition only.

Full adder

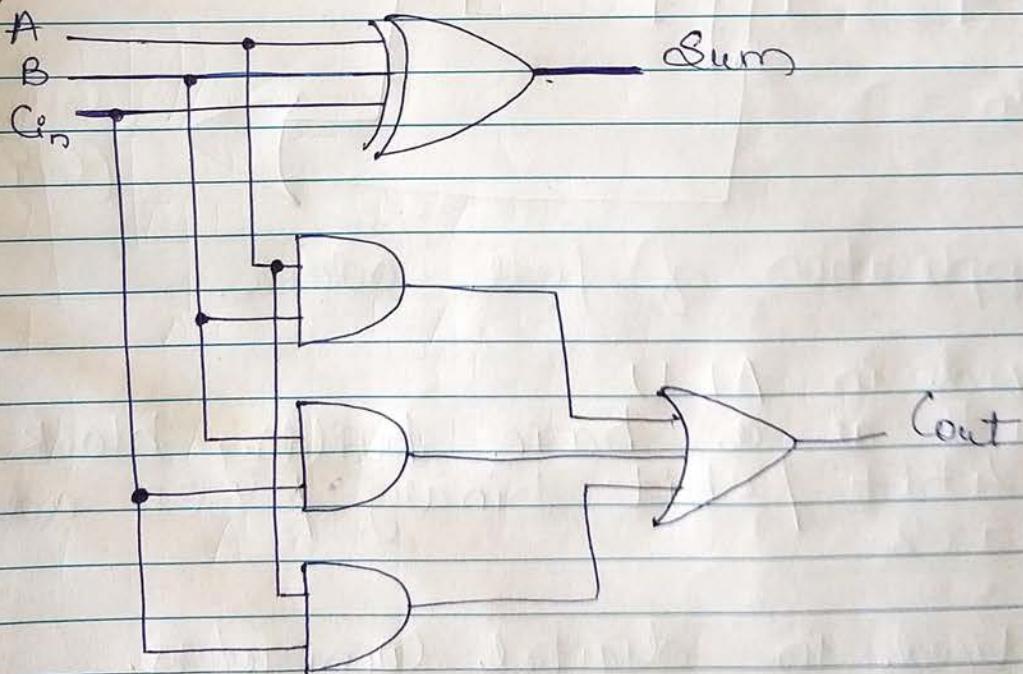
It is 3 i/p & 2 o/p combinational logic circuit.



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

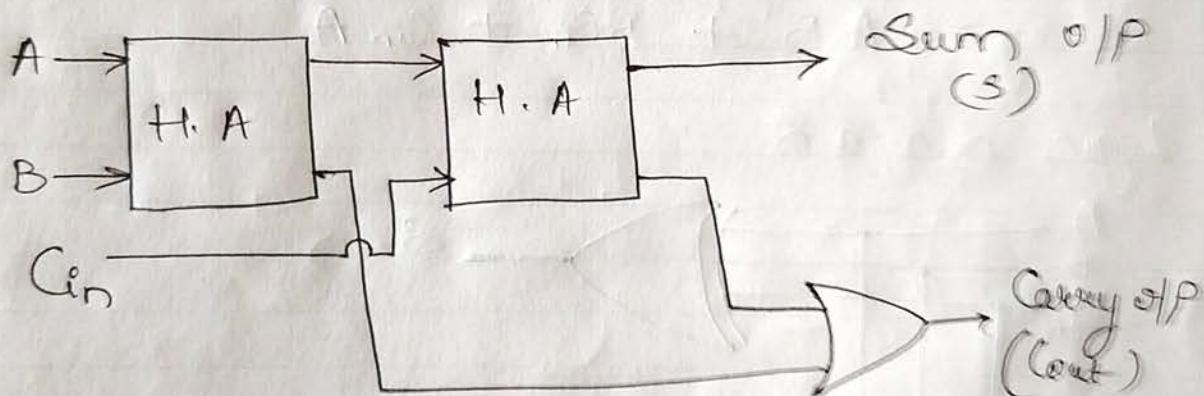
$$\text{Carry} = AB + BC_{in} + C_{in}A$$

Logic diagram



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

\* full adder using half adder.

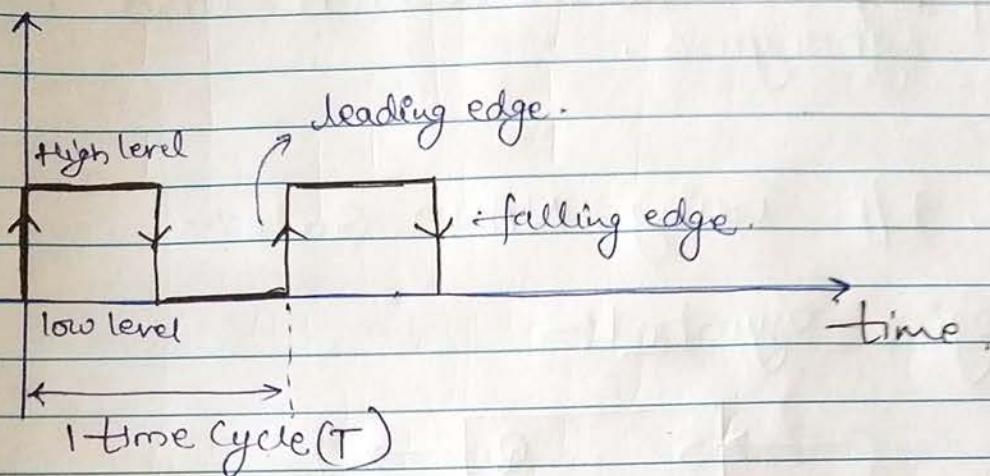


\* Application of full adder.

- 1) Used in ALU
- 2) acts as a basic building block of the 4 bit / 8 bit binary / BCD adder IC's.
- 3) Used in digital computers.

## Clock signal :-

It is basically a timing signal.



## \* Latch & flip flops :-

Latch :- type of temporary storage device which has 2 stable states.

- Different from flip flops.
- Bistable device.
- Level triggered.

flip flops :- bistable device

- Edge triggered device.

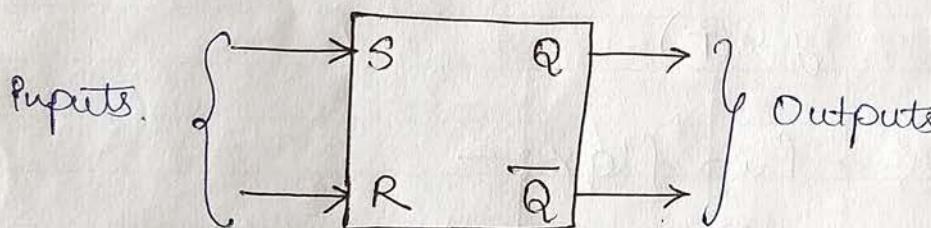
Edge triggered :- Samples its i/p & changes its o/p only at particular instant of time and not continuously.

level triggered:- Samples its o/p continuously according to changes in its i/p.

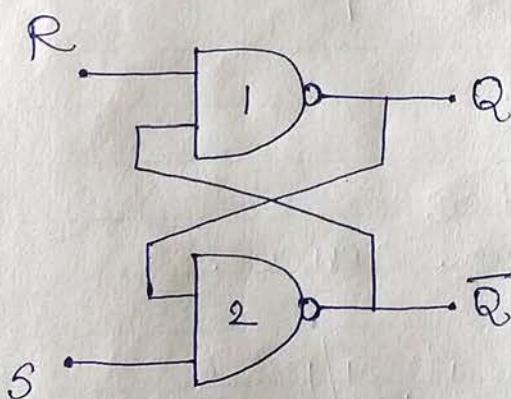
- In practice both Latch and ~~flip-flops~~ are synonymous.

\* SR F/F using NAND Gate:-

logic symbol:-



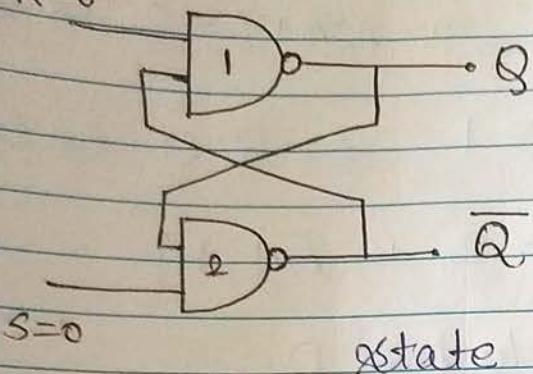
logic diagram.



I/P		O/P		
S	R	Q	$\bar{Q}$	
0	0	0	1	Reset
0	1	1	0	Set
1	0	1	0	
1	1	NC	NC	

- Case 1 :-  $S=0$  ;  $R=0$  (Race)

$R=0$



- any NAND o/p is 0 then o/p is become 1.

- Here,  $S=R=0$

$\therefore Q \neq \bar{Q}$  both

forced to become 1.

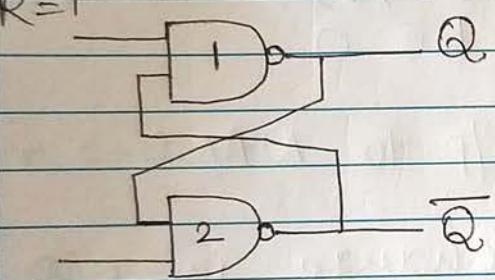
- which is indeterminate

state.

→ This condition is called Race Condition.

- Case 2 :-  $S=0$  ;  $R=1$  (Reset)

$R=1$



-  $S=0$ ;  $\therefore \bar{Q}$  is forced to be 1

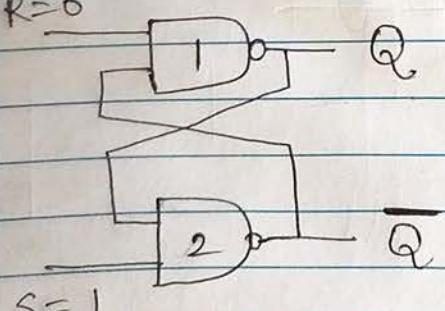
-  $R=1$ ;  $\therefore$  NAND 1 o/p is 0

$\therefore Q=0 \neq \bar{Q}=1$

- Called Reset Condition.

- Case 3 :-  $S=1$  ;  $R=0$  (Set)

$R=0$



-  $R=0$ ;  $\therefore Q$  forced to be 1

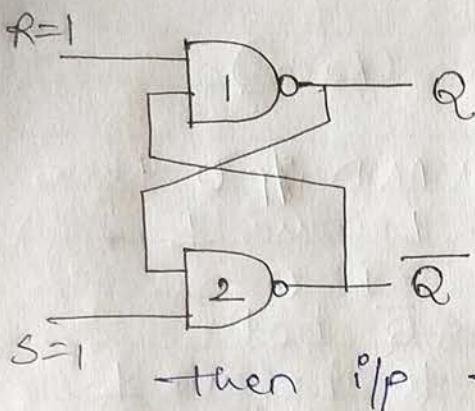
-  $S=1$ ;  $\therefore$  NAND 2 o/p is 0

$\therefore Q=1 \neq \bar{Q}=0$

- Called Set Cond.

$S=1$

- Case 4 :-  $R=1$ ;  $S=1$  No Change.



-  $R=1$ , forces  $Q$  to be 0

$\therefore$  i/p to NAND-2 is 0 & 1

- if  $Q=1$  &  $\bar{Q}=0$  initially.

then i/p to NAND-2 is [1, 1]

$\therefore Q$  continues to be '0'

$\therefore \bar{Q}=0$ , thus there is No change.

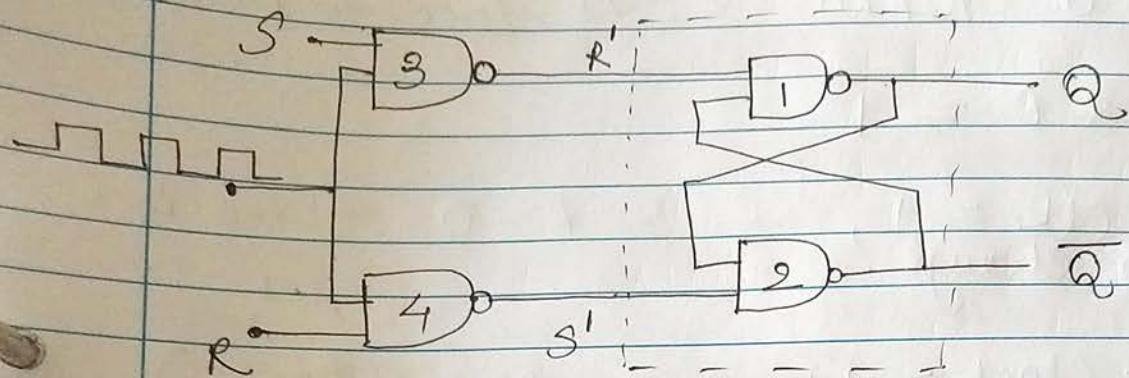
- if  $Q=0$  &  $\bar{Q}=1$  initially then  
i/p to NAND-1 (1, 1)  
 $Q \rightarrow$  continues to be 0

$\therefore Q=0$ ; Hence i/p to NAND-2 are  
(1, 0).

thus  $\bar{Q}$  will continue to be 1.  
i.e.  $\bar{Q}=1$

Hence there is No Change in o/p.  
if  $S=R=1$ .

## Clocked SR F/F:-



Inputs			$Q_{n+1}$	$\bar{Q}_{n+1}$	Output		Remark
clock	S	R	S'	R'	Q	$\bar{Q}$	
0	X	X	1	1	NC	NC	No Change
1	0	0	1	1	NC	NC	No change
1	0	1	0	1	0	1	Reset
1	1	0	1	0	1	0	SET
1	1	1			Race.	Race.	AVOID

Case 1:-  $\text{clock} = 0, S = X ; R = X$  (No change)

clock is 0 O/P of NAND 3 & 4 will be forced to be 1, irrespective of S & R

$$\therefore S' \& R' = 1$$

$\therefore$  clock is 0 then there is no change

- \* Case 2 :- Clock = 1, S = 0, R = 0 (No change)  
 if  $S = R = 0$  NAND 3 & 4 force to become 1  
 $\therefore S' \& R' = 1$   
 for  $\text{clk} = 1$ , &  $S = R = 0$  there is no change.

- \* Case 3 :- Clock = 1; S = 0; R = 1 (Reset)  
 Since  $S = 0$  O/P of NAND 3 i.e  $R' = 1$   
 $R = 1$  O/P of NAND 4 i.e  $S' = 0$   
 $\therefore$  Called RESET condn.

- \* Case 4 :- Clock = 1; S = 1, R = 0 (Set)  
 Since  $S = 1$  O/P of NAND 3 i.e  $R' = 0$   
 $R = 0$  O/P of NAND 4  $\therefore S' = 1$   
 $\therefore$  Called SET condn.

- \* Case 5 :- Clock = 1; S = 1, R = 1 (Race).  
 $S = 1 = R$ , so O/P of NAND 3 & 4.  
 i.e  $S' = R' = 0$

Hence Race condn.

\* Draw back of SR F/F :-

for  $S=R=0$  or  $S=R=1$

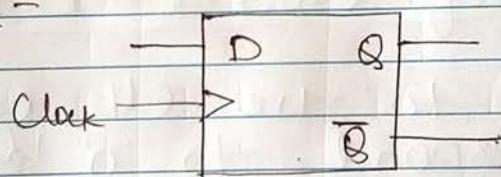
↓  
No change.

↓  
Indeterminate.  
(Race) condn.

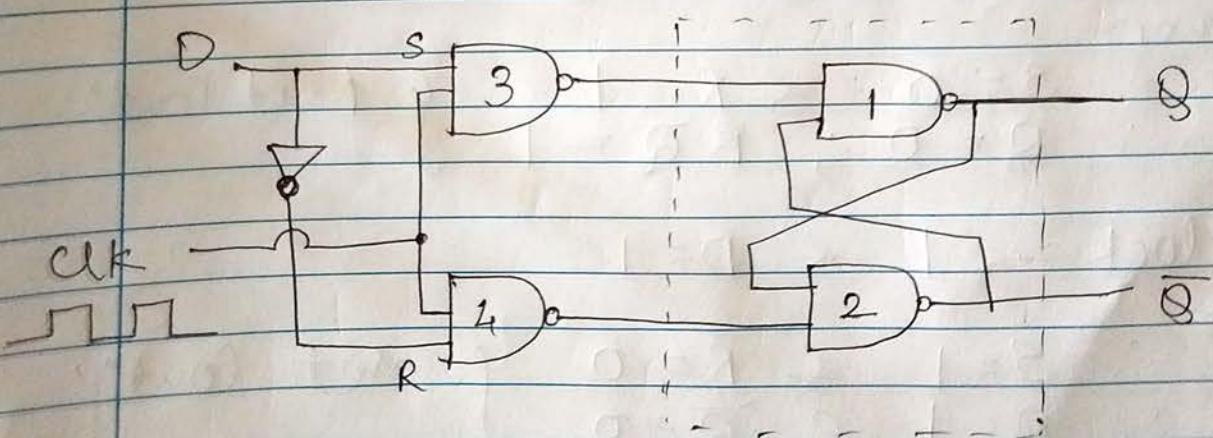
This disadvantage is overcome by D F/F

\* Clocked D/F/F :-

Symbol:-



Circuit diagram:-



- Clocked D FIF  $\rightarrow$  Clocked SR FIF with a logical inverter (NOT Gate) between S & R
  - it has only 1 flop 'D'
  - S & R are complement of each other. due to inverter.
  - Hence  $S = R = 0 \quad \} \text{Never appear.}$   
 $\& S = R = 1 \quad \}$
- to avoid "No change & Race Cond"

### \* Operation of Clocked D FIF: -

- Sensitive to the Clk cycle only. don't respond to '0' level or falling edge.
- Clock = 1 & D = 0  
 then  
 $S = 0, \& R = 1 \quad \} \text{Reset Cond}^n.$   
 $\therefore Q = 0; \& \overline{Q} = 1 \quad \}$
- Clock = 1 & D = 1  
 $\therefore S = 1 \& R = 0 \quad \} \text{Set Cond}^n.$   
 $\therefore Q = 1 \& \overline{Q} = 0 \quad \}$

$\therefore Q$  follows 'D' i/p.

$$\begin{aligned} D = 0 &\Rightarrow Q = 0 \\ D = 1 &\Rightarrow Q = 1 \end{aligned}$$

i/p		o/p	
clk.	D	Q	$\bar{Q}$
0	X	NC	NC
1	0	0	1
1	1	1	0

→ Reset  
→ Set

\* Applications:-

- ① As a delay element
- ② In digital register & counter.

## JK F/F

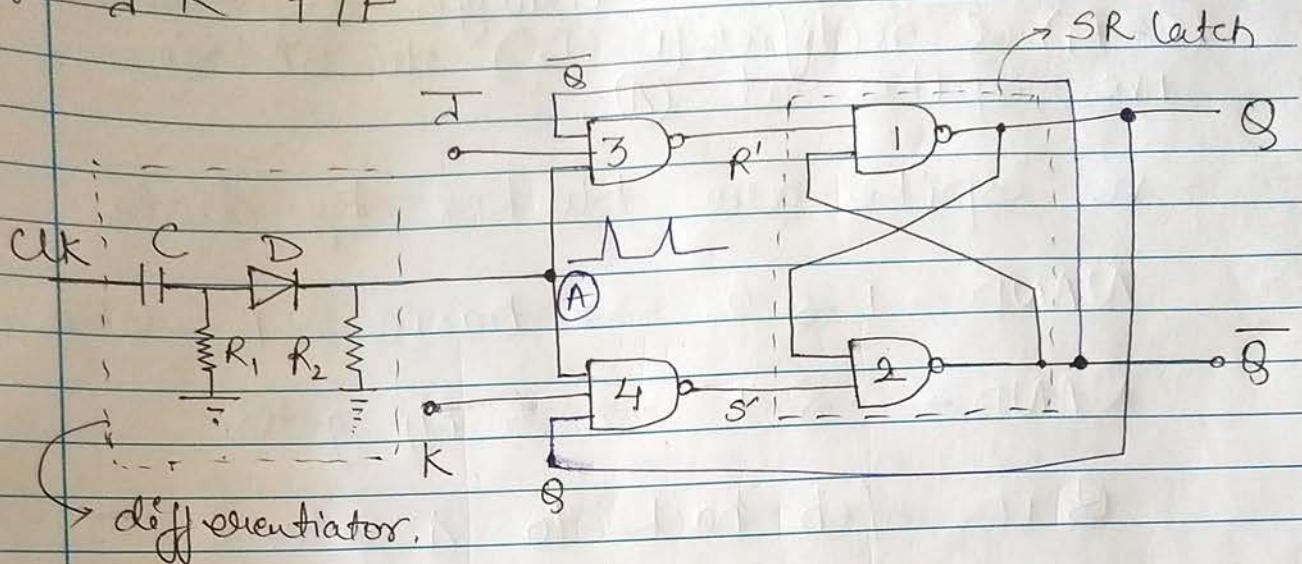


fig. +ve. edge triggered JK F/F

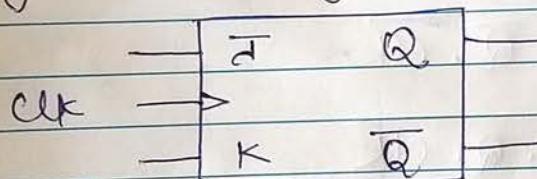


fig. 2. Symbol.

Case No.	I/Ps.			O/Ps		State.
	clk	J	K	$Q_{n+1}$	$\bar{Q}_{n+1}$	
1	0	X	X	No change	No change	FF is disabled.
	1	X	X	NC	NC	
2	↓	X	X	NC	NC	
3	↑	0	0	NC	NC	
4	↑	0	1	0	1	Reset
5	↑	1	0	1	0	Set
6	↑	1	1	$\bar{Q}_n$	$Q_n$	Toggle

→ CLK is passed through differentiator (C-R<sub>1</sub>) & rectifier (D & R<sub>2</sub>) to obtain the spikes at A.

→ -ve spikes are blocked by diode.

→ NAND  $\rightarrow$  1 & 2  $\rightarrow$  basic SR latch.

NAND  $\rightarrow$  3 & 4  $\rightarrow$  3 i/p gates.

Q is fed back to 4 gate

$\bar{Q}$  is fed back to 3 gate.

$$\therefore R' = \frac{1}{\pi \cdot \bar{Q} \cdot CLK}$$

$$S' = \frac{1}{\pi \cdot Q \cdot CLK}$$

Case 1:- CLK = 0 or  $\pm$

→ i.e. i/p is in the form of levels not in pulse.

∴ F/F is disabled

∴ OFF of differentiator = 0 for any level.

∴ Q &  $\bar{Q}$  will not change.

∴ No change condition.

Case 2 :- CLK edge is falling (↓)

Rectifier ( $D - R_2$ ) blocks falling edge

$\therefore V_{tg}$  at  $A = 0$  i.e. logic 0

$\therefore$  I/P of NAND  $3 \& 4 = 0$

$\therefore S' \& R' = 1$

$\therefore Q \& \bar{Q}$  will not change.

$\therefore$  No change condition.

Case 3 :- CLK = ↑,  $\bar{J} = 0, K = 0$

Since  $\bar{J} = 0 \rightarrow R' = 1$

$K = 0 \rightarrow S' = 1$

So  $Q \& \bar{Q}$  will not change even though ↑ CLK pulse is applied.

$\therefore$  No change condition.

Case 4 :- CLK = ↑,  $\bar{J} = 0, K = 1$

$S' = K \cdot Q \cdot \text{CLK}$  &  $R' = \bar{J} \cdot \bar{Q} \cdot \text{CLK}$

if initially  $Q = 1$  &  $\bar{Q} = 0$ .

Then  $S' = \frac{1 \cdot 1 \cdot 1}{1 \cdot 1 \cdot 1} = 0$  &  $R' = \frac{0 \cdot 0 \cdot 1}{0 \cdot 0 \cdot 1} = 1$

$\therefore$  according to SR FF if  $S' = 0$  &  $R' = 1$

then  $Q = 0$  &  $\bar{Q} = 1$

$\therefore$   $\bar{J} K$  FF will Reset.

→ If  $Q=0$  &  $\bar{Q}=1$  before the appin. of CLK pulse then there will be no change in Q/P

Case 5 :- CLK ↑,  $\bar{S}=1$ ,  $K=0$

If previous state'  $Q=0$  &  $\bar{Q}=1$ .  
then  $S'= \frac{1}{K \cdot Q \cdot \text{CLK}}$  &  $R'= \frac{1}{\bar{S} \cdot \bar{Q} \cdot \text{CLK}}$

$$S' = \frac{1}{0 \cdot 0 \cdot 1} = 1 \quad \& \quad R' = \frac{1}{1 \cdot 1 \cdot 1} = 0$$

Hence according to SR FF

$$S'=1 \quad \& \quad R'=0 \quad \text{then}$$

$$Q=1 \quad \& \quad \bar{Q}=0$$

i.e. FF is set.

if  $Q \& \bar{Q}$  i.e.  $Q=1 \& \bar{Q}=0$   
before appin. of CLK then there  
will be No change.

Case 6 :- CLK=↑,  $\bar{S}=1$ ,  $K=1$

① if  $Q=1$ ;  $\bar{Q}=0$

$$S' = \frac{1}{K \cdot Q \cdot \text{CLK}} = \frac{1}{1 \cdot 1 \cdot 1} = 0$$

$$R' = \frac{1}{\bar{S} \cdot \bar{Q} \cdot \text{CLK}} = \frac{1}{0 \cdot 0 \cdot 1} = 1$$

$\therefore S'=0 \quad \& \quad R'=1 \rightarrow \text{Reset Cond.}$   
 $Q_{n+1} = 0 \quad \& \quad \bar{Q}_{n+1} = 1$

(2)

$$Q = 0 \text{ } \& \text{ } \bar{Q} = 1$$

$$\therefore S' = \bar{K} \cdot Q \cdot \text{clk} = 1 \cdot 0 \cdot 1 = 1$$

$$R' = \bar{J} \cdot \bar{Q} \cdot \text{clk} = 1 \cdot 1 \cdot 1 = 0$$

$$\therefore S' = 1 \text{ } \& \text{ } R' = 0 \rightarrow \text{Set Cond.}$$

$$Q_{n+1} = 1 \text{ } \& \text{ } \bar{Q}_{n+1} = 0$$

From above discussion,

when  $J = K = 1$  &  $\text{clk} = \uparrow$

then  $Q$  &  $\bar{Q}$  O/Ps are inverted.

$$\text{i.e. } Q_{n+1} = \bar{Q}_n$$

$$\bar{Q}_{n+1} = Q_n$$

This is called - toggling Mode.

I Application

II Shift Registers.

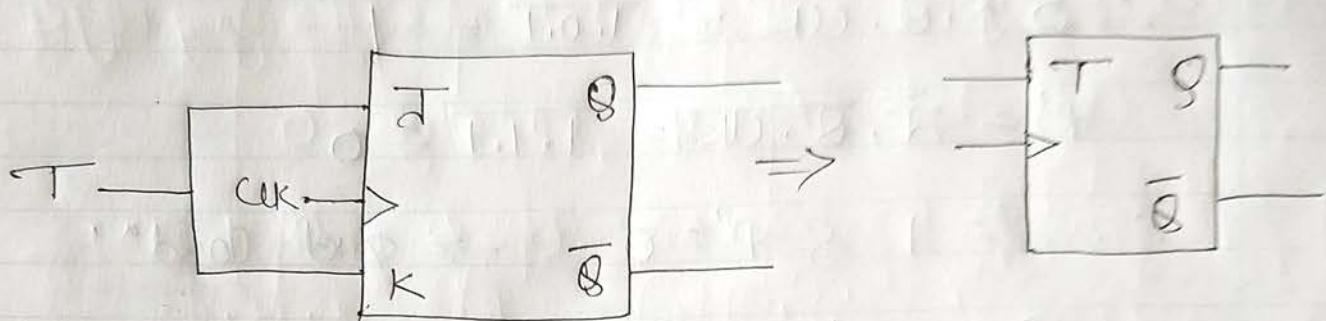
III Counters.

IV Race around Condition.

When  $J = K = 1$  & clk is applied, the O/P will complement itself till the clk is present. Hence at the end of clock pulse, the output is unpredictable.

This condition is called race around Condition.

\*  $T$  flip flop



Inputs	Output	State
C T Q <sub>n</sub>	Q <sub>n+1</sub>	
↑ 0 0	0	No change
↑ 0 1	1	(NC)
↑ 1 0	1	
↑ 1 1	0	Toggle
0 X 0	0	No change
0 X 1	1	(NC)

T	Q <sub>n+1</sub>
0	Q <sub>n</sub>
1	Q <sub>n</sub>

\* Application :-

- ① frequency divider.
- ② ripple counters.

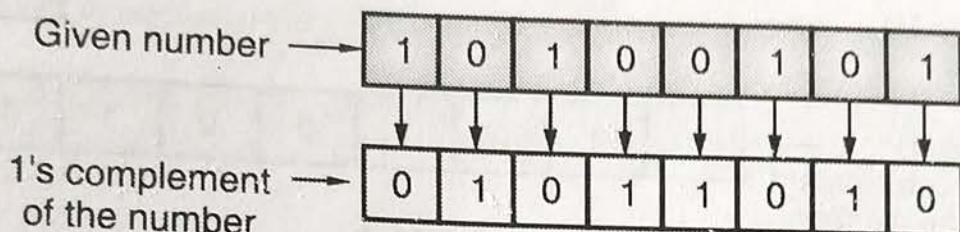
## Comparison - Flip-flop & Latches.

Latches	Flip-flops
1. Latches are constructed from the logic gate.	flip-flops are constructed from Latches & include clock signal.
2. Can modify its O/P any time.	can modify its O/P only when clock pulse is applied.
3. level triggered.	edge triggered.
4. require less power.	require more power.
5. simple to design.	complex to design.
6. more prone to glitches.	highly immune to glitches.

### 3.25.2(A) 1's Complement of a Binary Number

The 1's complement of a binary number is obtained by complementing every bit of the number i.e. a 0 is changed to 1 and vice-versa. (Refer Fig. 3.25.2)

**Example :**



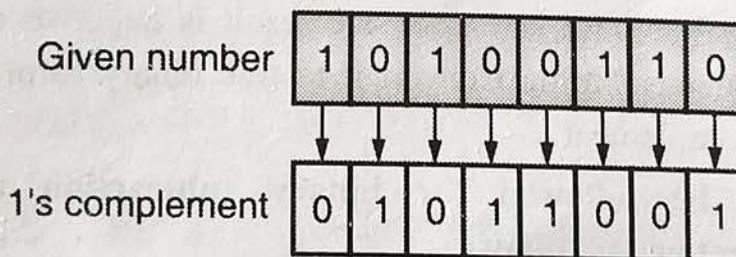
(1A83)Fig. 3.25.2 : 1's complement of a number

It is called as **1's complement** because if we subtract the given number from 11111111 to get the result, it is same as complementing the number.

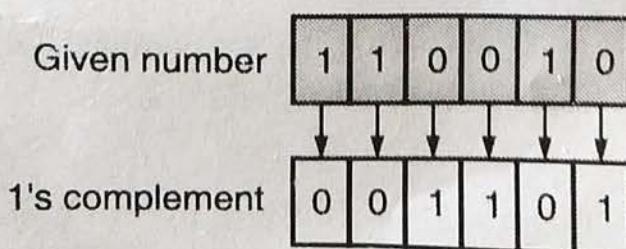
#### Ex. 3.25.2

Obtain the 1's complement of the following numbers.

**Soln. :** (Refer Fig. Ex. 3.25.2(a) and (b))



(1A84)Fig. Ex. 3.25.2(a)



(1A85)Fig. Ex. 3.25.2(b)

## 3.25.2(B) 2's Complement of a Binary Number

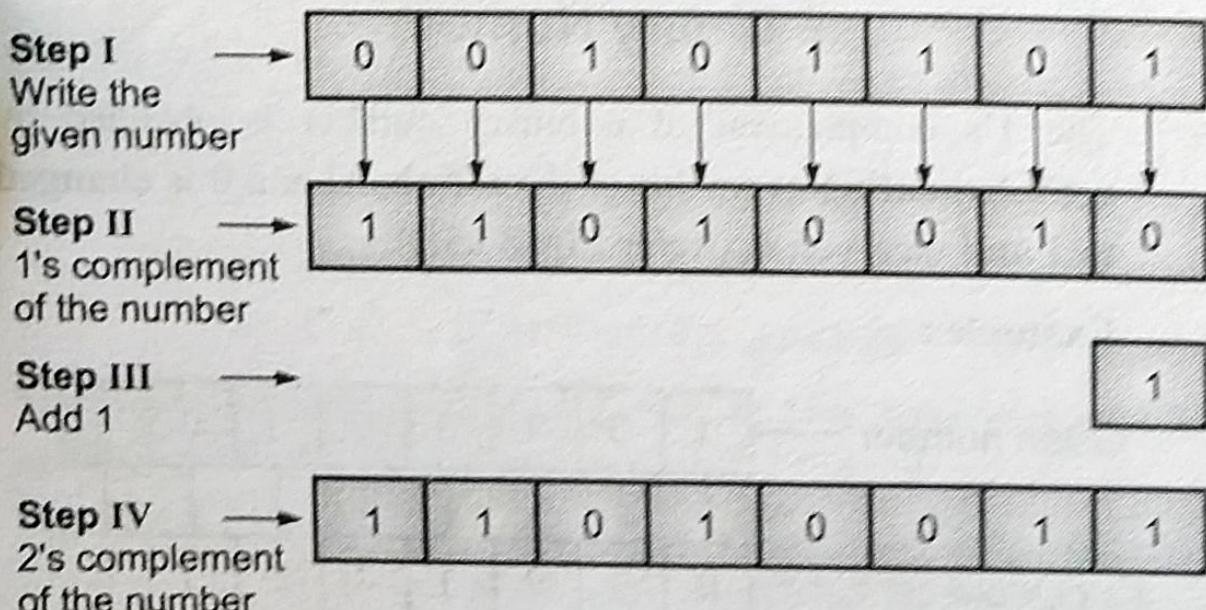
- 2's complement arithmetic is used in computers for handling negative numbers. Instead of using separate circuits for binary addition and subtraction, only adding circuits, i.e., adders are used. The complement of the subtrahend is added to the minuend rather than subtracting the number.
- The 2's complement of a binary number can be obtained by :
  - (1) Adding 1 to the LSB of 1's complement of that number.

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

- (2) Beginning from the LSB, write down each bit up to and including the first one and then complement the remaining bits.
- (3) Subtracting the given n-bit number from  $2^n$ .

**Example :** To find 2's complement of 00101101

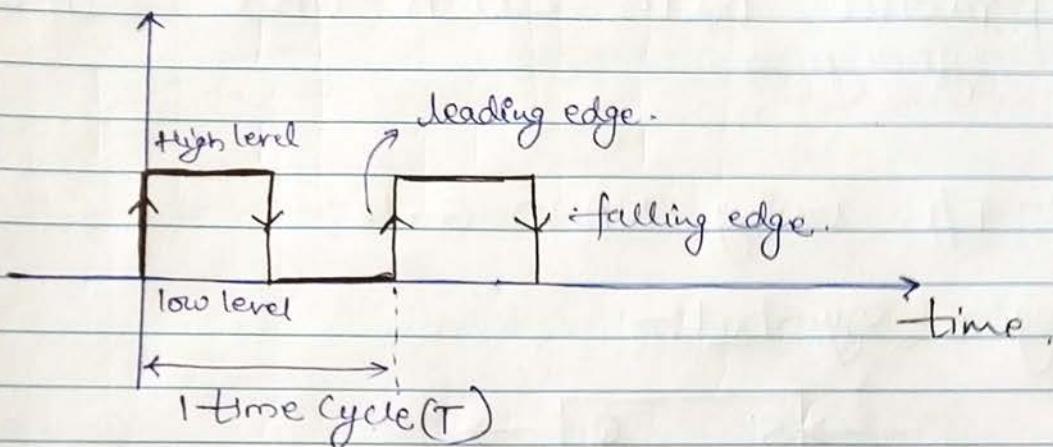
(Refer Fig. 3.25.3)



(1A86) Fig. 3.25.3

## \* Clock signal :-

- It is basically a timing signal.



## \* Latch & flip flops :-

Latch :- type of temporary storage device which has 2 stable states.

- Different from flip flops.
- Bistable device.
- level triggered.

~~flip flops~~ :- bistable device

- Edge triggered device.

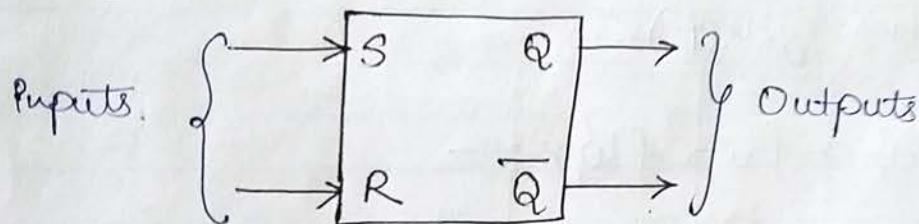
Edge triggered :- Samples its inputs & changes its output only at particular instant of time and not continuously.

level triggered:- Samples its  $i/p$  & changes its  $o/p$  continuously according to changes in its  $i/p$ .

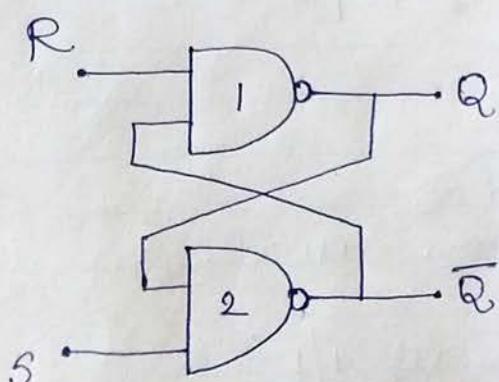
- In practice both Latch and flip-flops are synonymous.

\* SR F/F using NAND Gate:-

logic symbol:-



logic diagram.

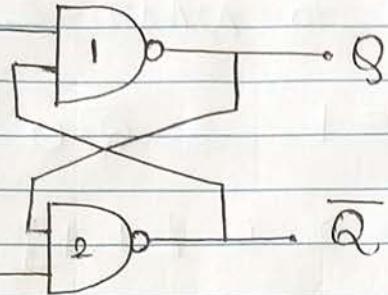


$i/p$		$o/p$	
S	R	Q	$\bar{Q}$
0	0	1	0
0	1	0	1
1	0	0	1
1	1	NC	NC

→ Reset  
→ Set

- Case 1 :-  $S=0$  ;  $R=0$  (Race)

$R=0$

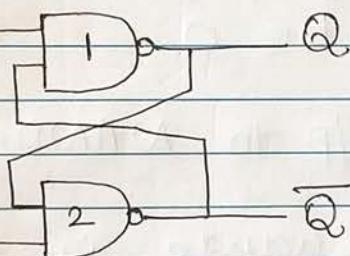


$S=0$

state. This condition is called Race Condition.

- Case 2 :-  $S=0$  ;  $R=1$  (Reset)

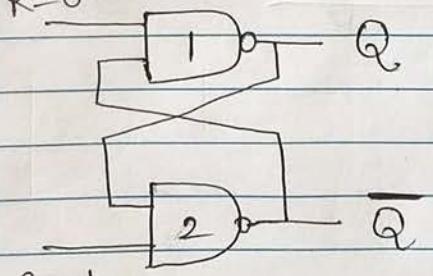
$R=1$



$S=0$

- Case 3 :-  $S=1$  ;  $R=0$  (Set)

$R=0$



$S=1$

- any NAND o/p is 0 then o/p is become 1.
- Here  $S=R=0$   
 $\therefore Q \neq \bar{Q}$  both forced to become 1.
- which is indeterminate

-  $S=0$  ;  $\therefore \bar{Q}$  is forced to be 1

-  $R=1$  ;  $\therefore$  NAND1 o/p is 0

$\therefore Q=0 \text{ } \& \text{ } \bar{Q}=1$

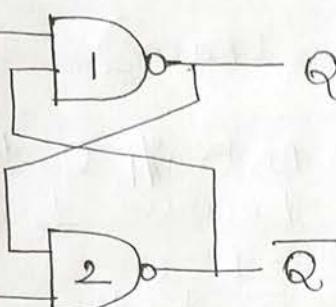
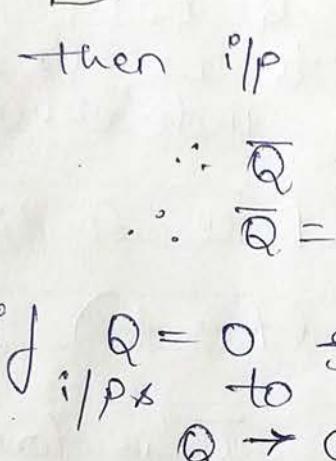
- Called Reset Condition.

-  $R=0$  ;  $\therefore Q$  forced to be 1

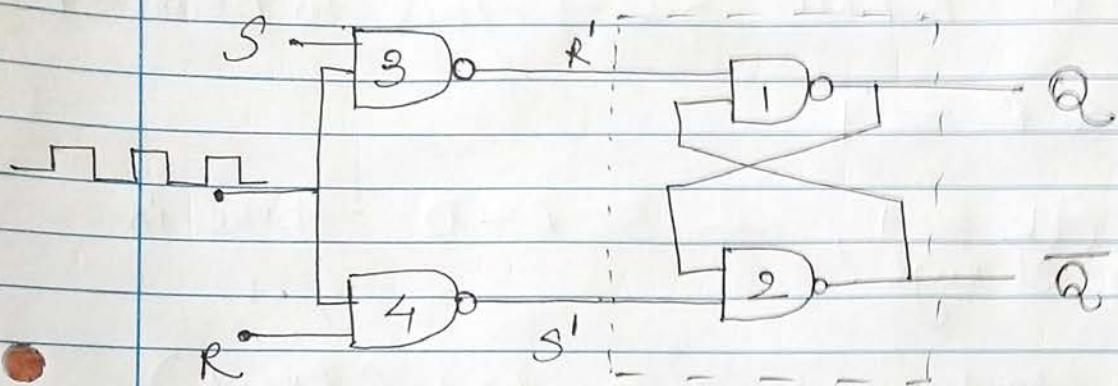
-  $S=1$  ;  $\therefore$  NAND2 o/p is 0

$\therefore Q=1 \text{ } \& \text{ } \bar{Q}=0$

- Called Set Cond.

- Case 4 :-  $R=1$  ;  $S=1$       No Change.
    - $R=1$  
    - $R=1$  , forces  $Q$  to be 0  
 $\therefore$  i/p to NAND-2 is 0&1
    - $S=1$  
    - if  $Q=1$  &  $\bar{Q}=0$  initially.  
 - then i/p to NAND-2 is  $[1, 1]$ 
      - $\therefore \bar{Q}$  continue to be '0'
      - $\therefore \bar{Q}=0$  , thus there is no change.
    - if  $Q=0$  &  $\bar{Q}=1$  initially then
      - i/p to NAND-1  $(1, 1)$
      - $Q \rightarrow$  continues to be 0
      - $\therefore Q=0$  ; Hence i/p to NAND-2 are  $(1, 0)$ .
      - thus  $\bar{Q}$  will continue to be 1.
- Hence there is No Change in o/p.  
 if  $S=R=1$ .

## \* Clocked SR FF:-



Inputs			$Q_{n+1}$	$\bar{Q}_{n+1}$	Output		Remark
Clock	S	R	$S'$	$R'$	Q	$\bar{Q}$	
0	X	X	1	1	NC	NC	No change
1	0	0	1	1	NC	NC	No change
1	0	1	0	1	0	1	Reset
1	1	0	1	0	1	0	SET
1	1	1			Race	Race	AVOID

Case 1:-  $\text{clock} = 0, S = X, R = X$  (No change)

Clock is 0, O/P of NAND 3 & 4 will be forced to be 1, irrespective of S & R.

$$\therefore S' \& R' = 1$$

$\therefore$  Clock is 0 then there is no change.

- \* Case 2 :- Clock = 1, S = 0, R = 0 (No change)  
 if  $S = R = 0$  NAND 3 & 4 force to become 1  
 $\therefore S' \text{ & } R' = 1$   
 ∵ for  $\text{clk} = 1$ , &  $S = R = 0$  → there is no change.
- \* Case 3 :- Clock = 1; S = 0; R = 1 (Reset)  
 Since  $S = 0$  O/P of NAND3 i.e.  $R' = 1$   
 $R = 1$  O/P of NAND4 i.e.  $S' = 0$   
 ∴ Called RESET condn.
- \* Case 4 :- Clock = 1; S = 1, R = 0 (Set)  
 Since  $S = 1$  O/P of NAND3 i.e.  $R' = 0$   
 $R = 0$  O/P of NAND4  $\therefore S' = 1$   
 ∴ Called set condn.
- \* Case 5 :- Clock = 1; S = 1, R = 1 (Race).  
 $S = 1 = R$ , so O/P of NAND 3 & 4.  
 i.e.  $S' = R' = 0$   
 Hence race condn.

\* Draw back of SR F/F :-

for  $S=R=0$  or  $S=R=1$

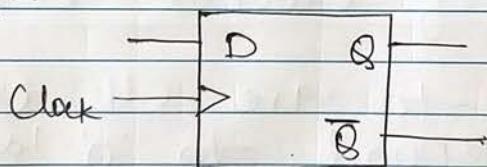
$\downarrow$   
No change.

$\downarrow$   
Indeterminate.  
(Race) cond<sup>n</sup>.

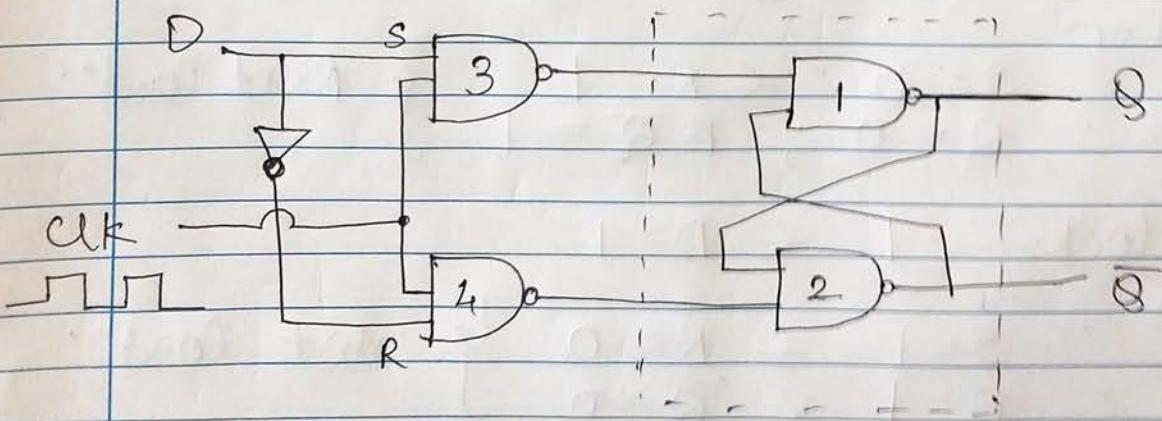
This disadvantage is overcome by  
'D' F/F

\* Clocked DFF :-

Symbol:-



Circuit diagram:-



- Clocked D F/F  $\rightarrow$  Clocked SR F/F with a logical inverter (NOT Gate) between S & R
- it has only 1 I/P 'D'
- S & R are complement of each other. due to inverter.
- Hence  $S = R = 0 \quad \} \text{Never appear.}$   
 $\& S = R = 1 \quad \}$

$\&$  avoid "No change & Race Cond".

### \* Operation of Clocked D F/F: -

$\rightarrow$  Sensitive to the Clk cycle only. don't respond to '0' level or falling edge.

- Clock = 1 & D = 0  
 then

$$S = 0 \quad \& R = 1 \quad \} \text{Reset cond.} \\ \therefore Q = 0 ; \bar{Q} = 1$$

- Clock = 1 & D = 1

$$\therefore S = 1 \quad \& R = 0 \quad \} \text{Set cond.} \\ \therefore Q = 1 \quad \& \bar{Q} = 0$$

$\therefore Q$  follows 'D' i/p.

$$D = 0 \Rightarrow Q = 0$$

$$D = 1 \Rightarrow Q = 1$$

i/p		o/p	
CK	D	$Q$	$\bar{Q}$
0	X	NC	NC
1	0	0	1
1	1	1	0

$\rightarrow$  Reset

$\rightarrow$  Set

\* Applications:-

- ① As a delay element
- ② In digital register & counter.

21 F/F

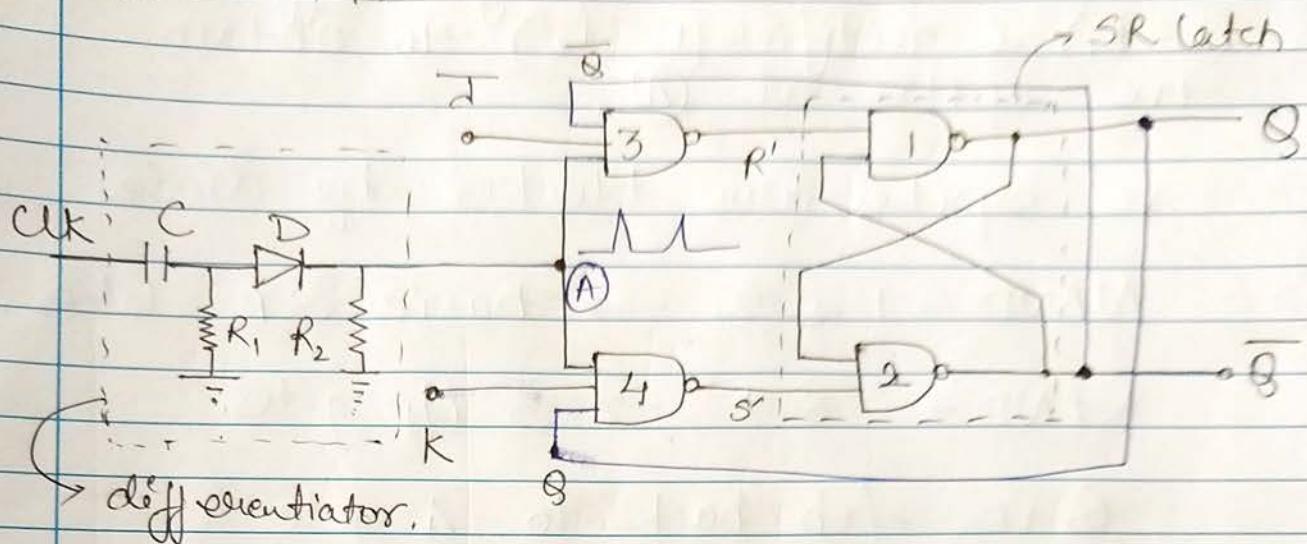
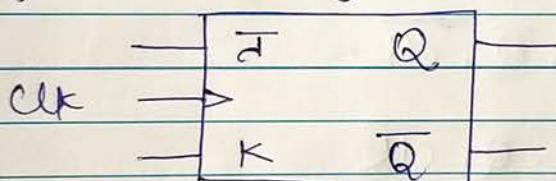


fig 1. +ve edge triggered JK F/F



## fig 2. Symbol.

Case No	I/Ps.			O/Ps		State
	Clk	J	K	$Q_{n+1}$	$\bar{Q}_{n+1}$	
1	0 or 1	X	X	No change	No change	
	1	X	X	NC	NC	FF is
2	↓	X	X	NC	NC	disabled
3	↑	0	0	NC	NC	
4	↑	0	1	0	1	Reset
5	↑	1	0	1	0	Set
6	↑	1	1	$Q_n$	$\bar{Q}_n$	-toggle

- CLK is passed through differentiator  $(C - R_1)$  & rectifier  $(D \& R_2)$  to obtain the spikes at A.
- -ve spikes are blocked by diode.
- $NAND \rightarrow 1 \& 2 \rightarrow$  basic SR latch.
- $NAND \rightarrow 3 \& 4 \rightarrow$  3 i/p gates.

$Q$  is fed back to 4 gate

$\bar{Q}$  is fed back to 3 gate.

$$\therefore R' = \overline{J \cdot \bar{Q} \cdot CLK}$$

$$S' = \overline{K \cdot Q \cdot CLK}$$

Case 1:- CLK = 0 or 1

→ i.e. i/p is in the form of level not in pulse.

∴ F/F is disabled

∴ OFF of differentiator = 0 for any level.

∴  $Q$  &  $\bar{Q}$  will not change.

∴ No change condition.

Case 2 :- CLK edge is falling (↓)

Rectifier (D-R<sub>2</sub>) blocks falling edge

∴ V<sub>TG</sub> at A = 0 i.e. logic 0

∴ i/P of NAND 3 & 4 = 0

∴ S' & R' = 1

∴ Q &  $\bar{Q}$  will not change.

∴ No change condition.

Case 3 :- CLK = ↑, J = 0, K = 0

Since J = 0  $\rightarrow R' = 1$

K = 0  $\rightarrow S' = 1$

So Q &  $\bar{Q}$  will not change even though ↑ CLK pulse is applied.

∴ No change condition.

Case 4 :- CLK = ↑, J = 0, K = 1

$S' = \overline{K \cdot Q \cdot CLK}$  &  $R' = \overline{J \cdot \bar{Q} \cdot CLK}$

if initially Q = 1 &  $\bar{Q} = 0$ .

Then  $S' = \overline{1 \cdot 1 \cdot 1} = 0$  &  $R' = \overline{0 \cdot 0 \cdot 1} = 1$

∴ according to SR FF if S' = 0 & R' = 1

then Q = 0 &  $\bar{Q} = 1$

∴ JK FF will Reset.

→ If  $Q=0$  &  $\bar{Q}=1$  before the app'ng of CLK pulse then there will be no change in Q/P

Case 5 :- CLK  $\uparrow$ ,  $\bar{T}=1$ ,  $K=0$

if previous state'  $Q=0$  &  $\bar{Q}=1$ .  
then  $S'= \overline{K \cdot Q \cdot CLK}$  &  $R'= \overline{\bar{T} \cdot \bar{Q} \cdot CLK}$

$$S' = \overline{0 \cdot 0 \cdot 1} = 1 \quad \& \quad R' = \overline{1 \cdot 1 \cdot 1} = 0$$

Hence according to SR F/F

$$S'=1 \quad \& \quad R'=0 \quad \text{then}$$

$$Q=1 \quad \& \quad \bar{Q}=0$$

i.e. F/F is set.

if  $Q \& \bar{Q}$  i.e  $Q=1 \& \bar{Q}=0$   
before app'ng of CLK then there  
will be No change.

Case 6 :- CLK  $= \uparrow$ ,  $\bar{T}=1$ ,  $K=1$

① if  $Q=1$  ;  $\bar{Q}=0$

$$S' = \overline{K \cdot Q \cdot CLK} = \overline{1 \cdot 1 \cdot 1} = 0$$

$$R' = \overline{\bar{T} \cdot \bar{Q} \cdot CLK} = \overline{1 \cdot 0 \cdot 1} = 1$$

$\therefore S'=0 \quad \& \quad R'=1 \rightarrow$  Reset Cond'

$$Q_{n+1} = 0 \quad \& \quad \bar{Q}_{n+1} = 1$$

② If  $Q=0$  &  $\bar{Q}=1$

$$\therefore S' = K \cdot Q \cdot \bar{Ck} = 1 \cdot 0 \cdot 1 = 1$$

$$R' = \bar{T} \cdot \bar{Q} \cdot Ck = \bar{1} \cdot 1 \cdot 1 = 0$$

$\therefore S'=1$  &  $R'=0 \rightarrow$  2nd cond

$$Q_{n+1} = 1 \quad \& \quad \bar{Q}_{n+1} = 0$$

From above discussion,

when  $T=K=1$  &  $Ck=1$

then  $Q$  &  $\bar{Q}$  O/Ps are inverted.

$$\text{i.e. } Q_{n+1} = \bar{Q}_n$$

$$\bar{Q}_{n+1} = Q_n$$

This is called - toggling Mode.

# Application

1] Shift Registers.  
2] Counters.

# Race around Condition.

When  $T=K=1$  &  $Ck$  is applied,

the O/P will complement itself

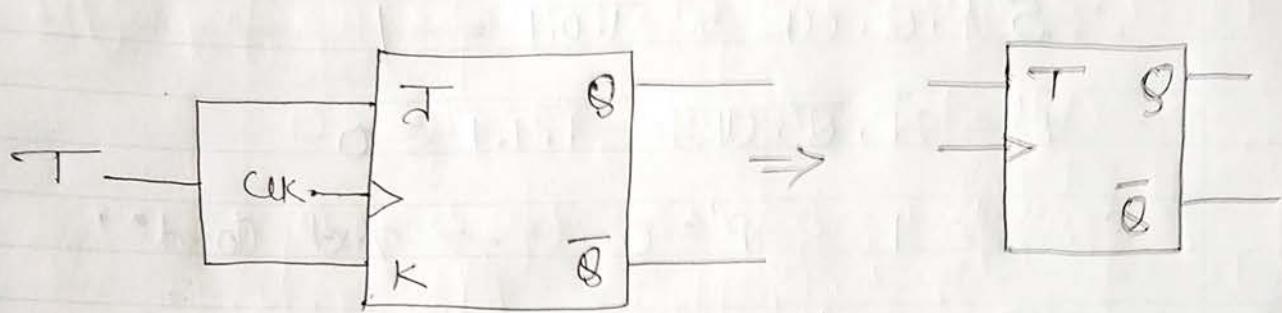
till the  $Ck$  is present. Hence at

the end of clock pulse, the output

is unpredictable.

→ This condition is called race around condition.

## \* T - flip flop



Inputs	Output		state.	
C	T	$Q_n$	$Q_{n+1}$	
↑	0	0	0	No change (NC)
↑	0	1	1	
↑	1	0	1	
↑	1	1	0	Toggle
0	x	0	0	No change (NC)
0	x	1	1	

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

## \* Application :-

- ① frequency divider.
- ② ripple counters.

## 11 Comparison - flip-flop & Latches.

### Latches

### Flip-flops

1. Latches are constructed from the logic gate	flip-flops are constructed from Latches & include clock signal.
2. Can modify its output any time	can modify its output only when clock pulse is applied.
3. level triggered	edge triggered.
4. require less power	require more power.
5. simple to design	complex to design
6. more prone to glitches.	highly immune to glitches.