

Unit - IV

Sequential circuit)— The output of a sequential logic circuit depends on the present input & past outputs. So we need a m/r element to store the past output. The outputs of the m/r elements, is given to the inputs of Combinational logic circuits. Eg:- flip flops, counter etc.

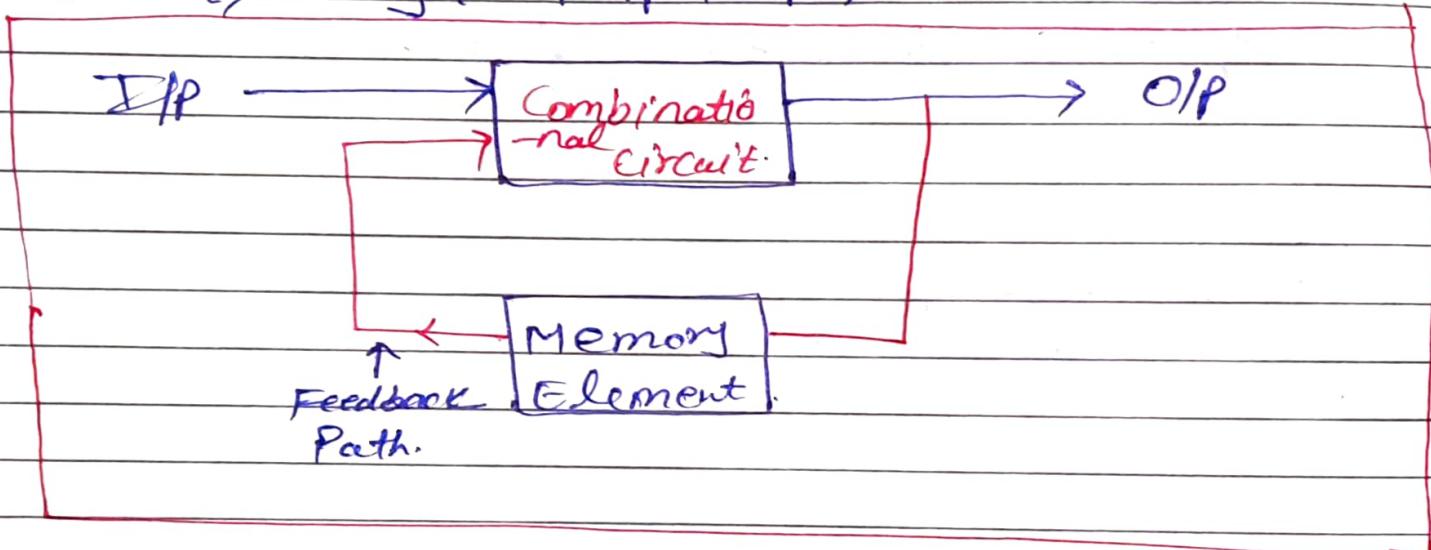


fig:- Sequential circuit

Types of Sequential circuits:-

- 1) Asynchronous circuit do not synchronize with positive edge or negative edge of the clock signal, it means the output of the sequential circuits do not change or affect at the same time & change their states immediately when there is a change in the input signal. so, these circuits ~~do not change at different~~ are faster & independent of the internal clock pulses. These are difficult to design.

External input

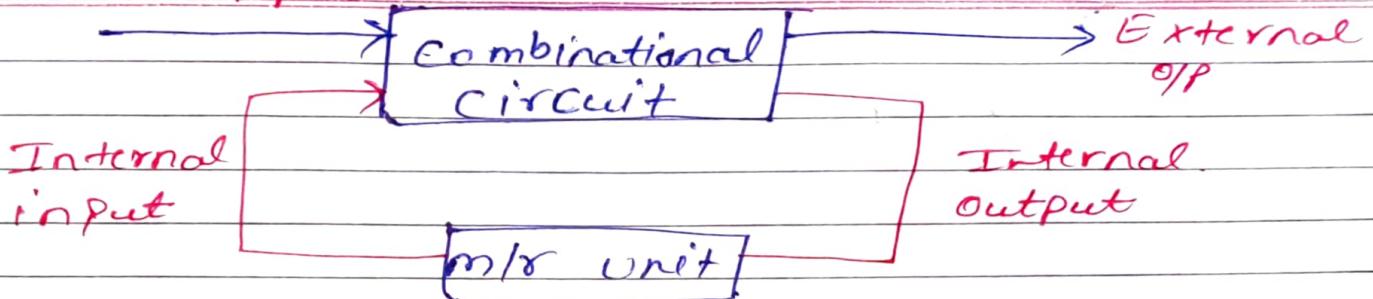


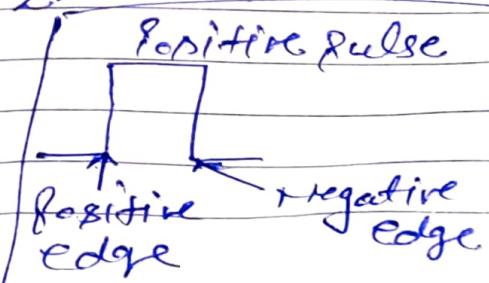
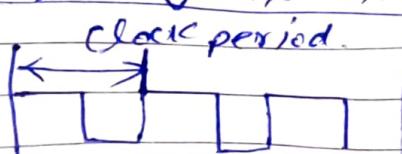
Fig:- Asynchronous Sequential circuit

~~flip flop, register, counter etc.~~

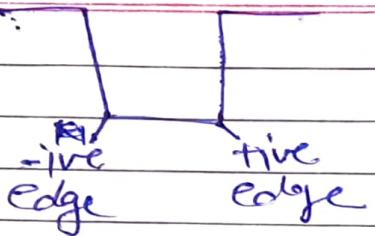
(2) Synchronous Sequential Circuits:- The output of synchronous sequential circuit changes according to the condition, at particular time before signal. Their output can be change only by giving system clock pulse.

Clock :- A clock is a special device that continuously outputs 0's & 1's.

- The time it takes the clock to change from 1 to 0 & back to 1 is called the clock period or clock cycle time.
- The clock frequency is the inverse of the clock period. The unit of measurement for frequency is the hertz.



-ive pulse.



Flip flop

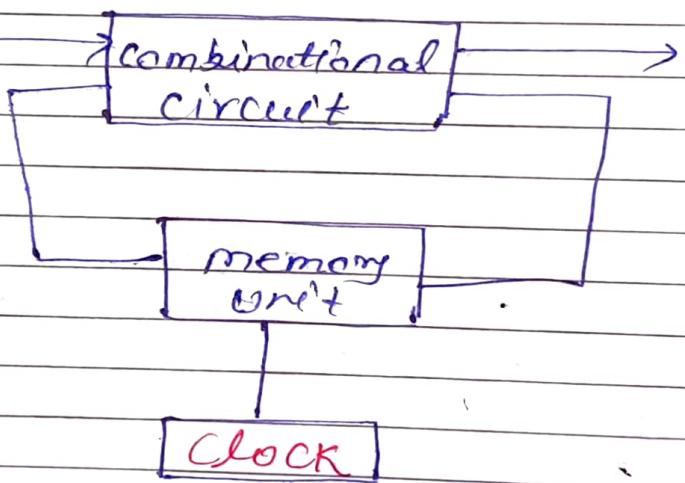


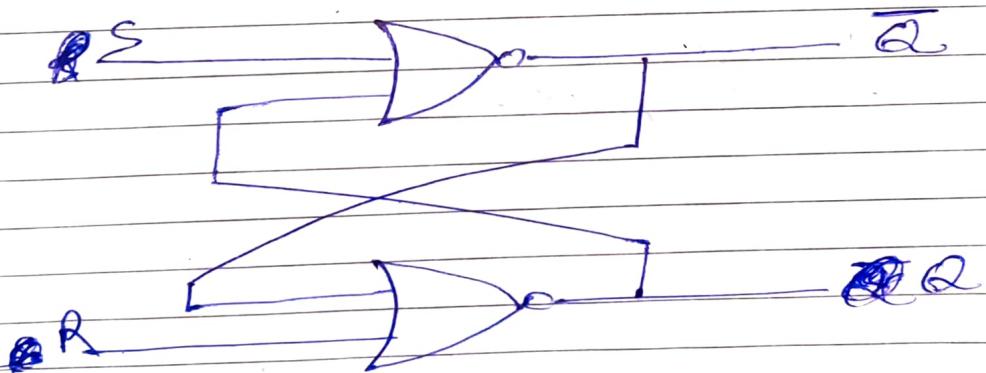
fig:- Synchronous sequential circuit.

flip flops:-

Latches :- A Latch is a basic memory element that operates with signal levels & stores 1 bit of data. Latches are said to be sensitive devices. Latches are useful for storing information & for the design of asynchronous sequential circuits.

SR Latch :- The SR latch is a circuit with two cross-coupled NOR-gate or two cross coupled NAND gates with two input labelled S (Set) & R (Reset) & with two complementary outputs Q & \bar{Q} .

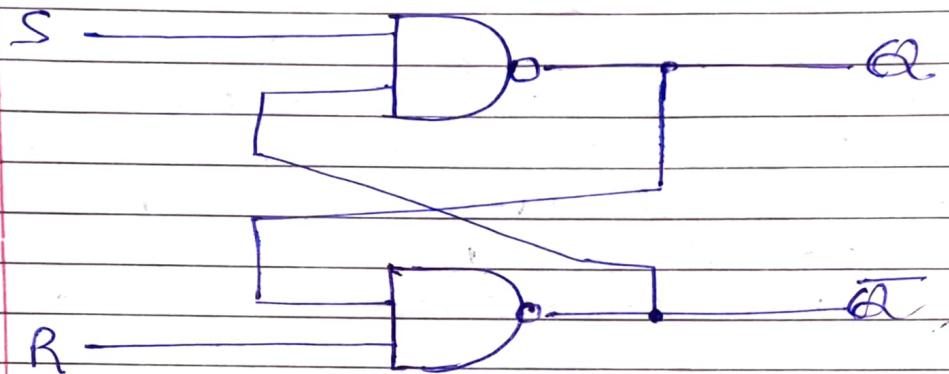
SR Latch Using NOR gates:-



S	R	Q	\bar{Q}
0	0	Held State	
0	1	0	1
1	0	1	0
1	1	invalid state	

I/P	I/P	OR	NOR
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

SR latch using NAND gate :-



S	R	Q	Q'
0	0	invalid	
0	1	1	0
1	0	0	1
1	1	Hold State	

I _P	I _T _P	AND	NAND
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Latches are the basic building blocks of flip flops:-

Flip flop :- A circuit that has two stable states (0, 1) or (High & Low) is treated as a flip flop. These stable states are used to store binary data that can be changed by applying varying inputs. The flip flops are the fundamental building blocks of digital system.

(1) **S-R flip-flop :-** most common flip flop used in the digital system. It is also known as Set-Reset flip flop. The S & R inputs control the state of the flip flop when the clock pulse goes from Low to High. The flip flop will not change until the clock pulse is on a rising edge. When S-R (both) are simultaneously High, it is uncertain whether the output will be high or low.

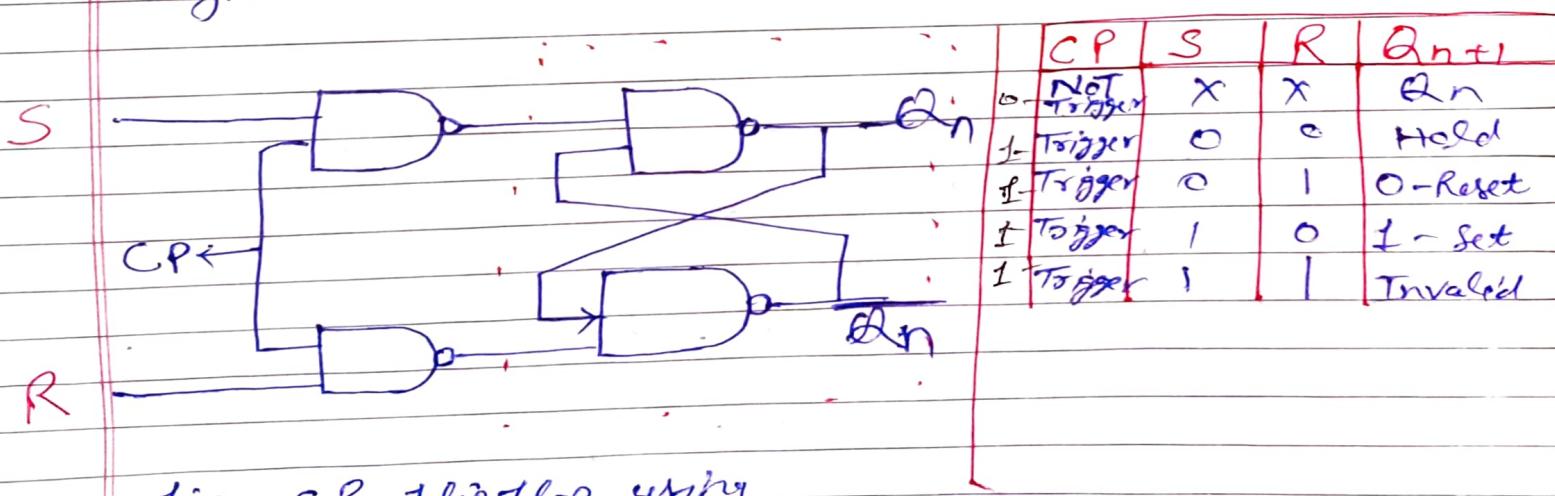
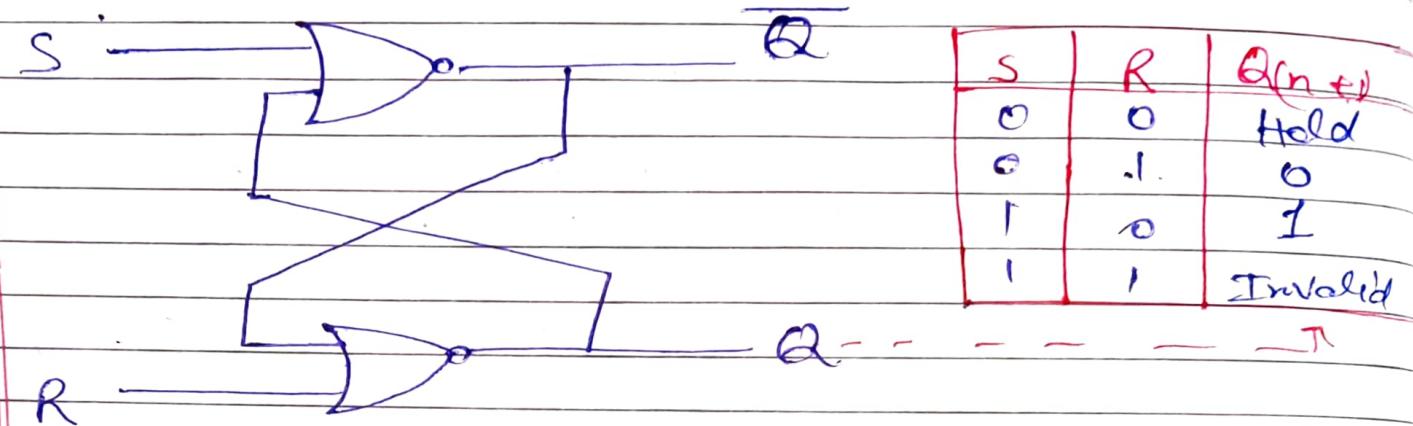


fig:- SR flip flop using
NAND gate *

SR Latch using NOR Gate



S	R	$Q(n+1)$
0	0	Hold
0	1	0
1	0	1
1	1	Invalid

Table for NOR Gate

Input	I OR	H OR
0 0	0	1
0 1	1	0
1 0	1	0
1 1	1	0

Characteristic table for SR Flip-flop:-

S	R	Q_n	Q_{n+1}	
0	0	0	0	→ 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	X	→ don't care
1	1	1	X	

Characteristic equation -

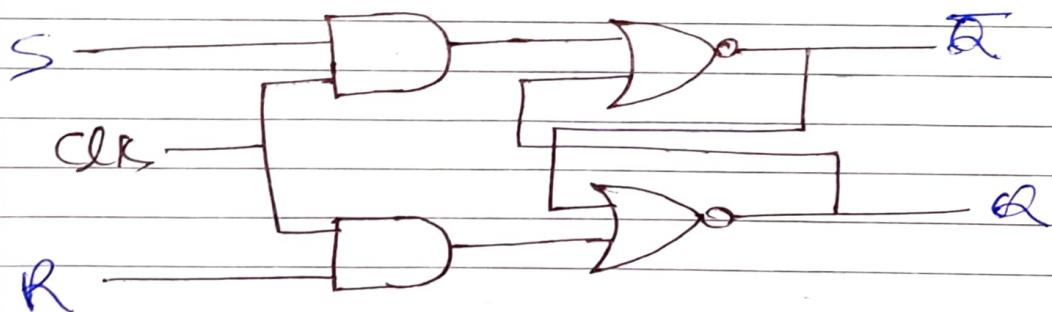
S \ R Qn	00	01	11	10
0	0	1	1	0
1	1	1	*	*

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

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

SR flip flop using Nor Gate OR NOR Latch

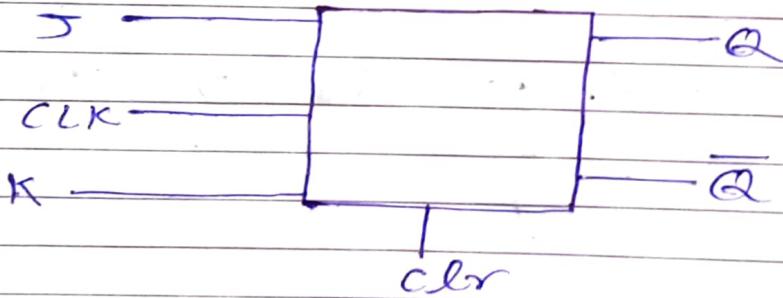


Clock	S	R	Q _{n+1}
not trig.	X	X	Q _n
Triggered	0	0	Hold
"	0	1	0
"	1	0	1
"	1	1	Invalid

What is JK flip-flop:-

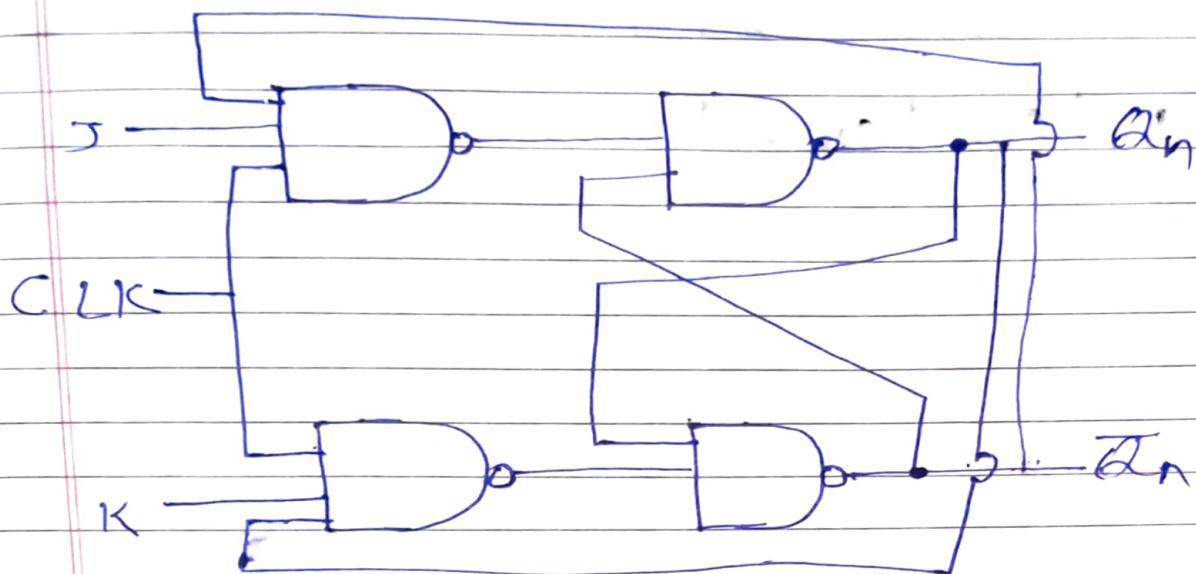
JK Kilby flop, can be used as a basic memory element. It can store binary information & toggle functionality with a diversity of making application with it.

It is one kind of sequential logic circuit which stores binary information in bit wise manner. It consists of two inputs & two outputs. Inputs are J (Set) & K (Reset) & their corresponding outputs are Q & Q'. JK flip flop block diagram as shown below:-



The JK flip flop is a refinement of S-R flip flop in which the S-R type's invalid state is defined.

S-R flip flop is a most basic flip flop. we can design further flip flops using S-R flip flops.



S R Flip flop Truth Table:-

S	R	Q_{n+1}
0	0	Hold
0	1	0
1	0	1
1	1	Invalid

→ This problem of S-R
will resolve by J-K flip flop.

It will toggle i.e. \bar{Q}_n

Case I $Q_n = 1, Q_{n+1} = 0 \quad \left. \right\} \text{Toggle value.}$

Case II $Q_n = 0, Q_{n+1} = 1 \quad \left. \right\} \text{Toggle value.}$

J	K	Q_{n+1}
0	0	Hold
0	1	0
1	0	1
1	1	Toggle.

Characteristic table :-

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

Characteristic equation:-

$$\cancel{J} \cancel{K} \bar{Q}_n = \bar{K} \bar{Q}_n + \bar{Q}_n \bar{K} + \bar{Q}_n \bar{J}$$

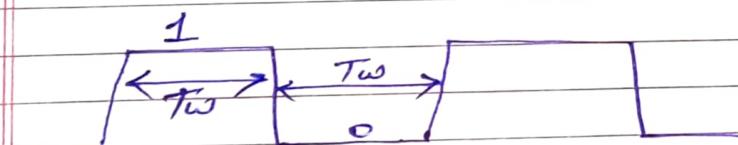
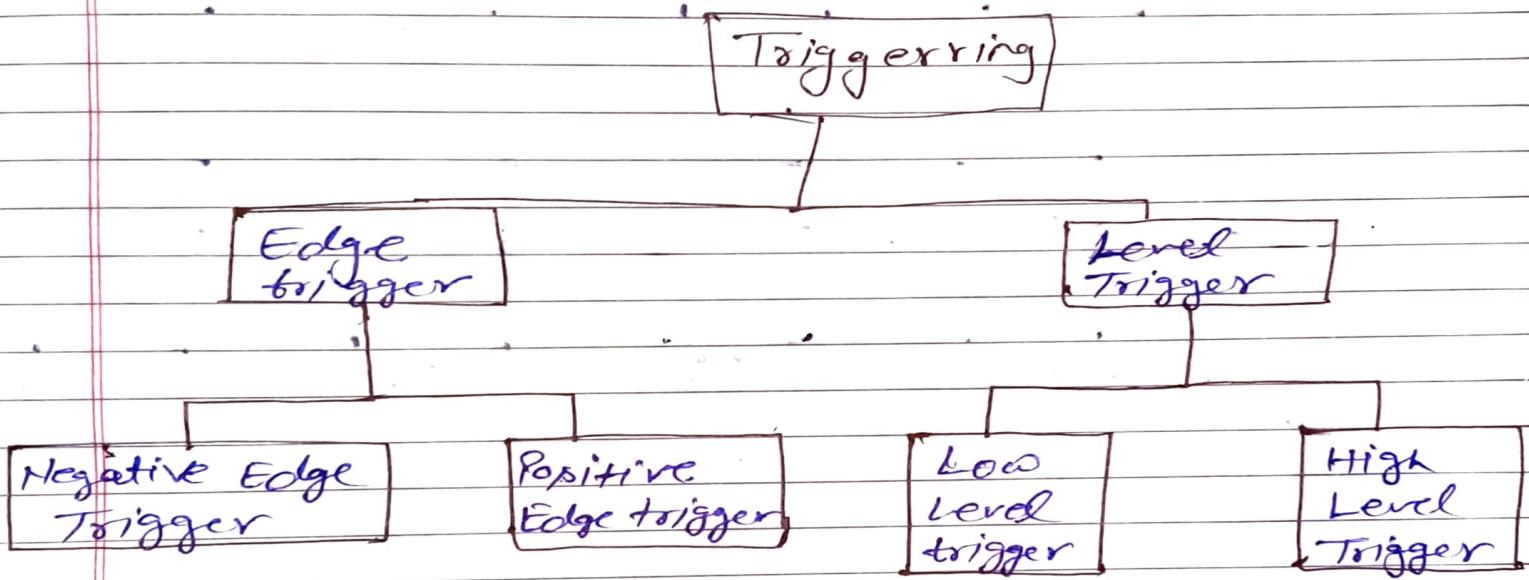
\bar{J}	\bar{I}	$\boxed{1}$	$\boxed{1}$	$\boxed{1}$	\vdots
\bar{J}	\bar{I}	$\boxed{1}$	$\boxed{1}$	$\boxed{1}$	

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

Excitation Table :-

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

Types of Triggering :-



$$T_W = T/2$$

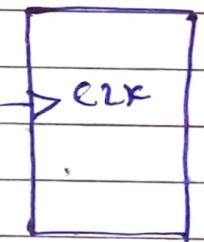
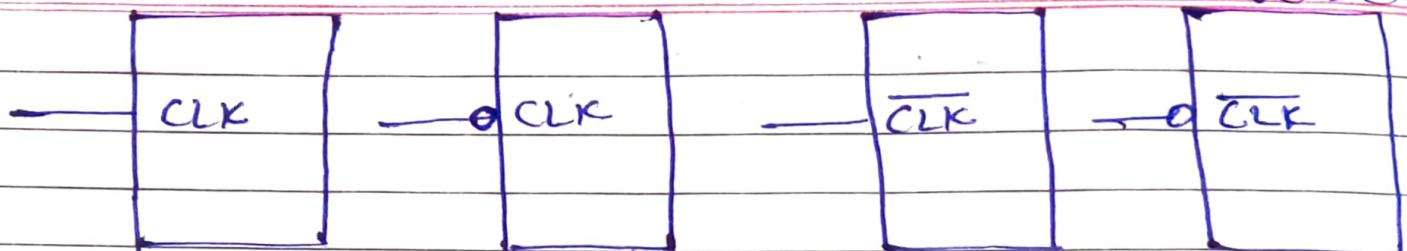
T = total time for clock cycle.

+ve level

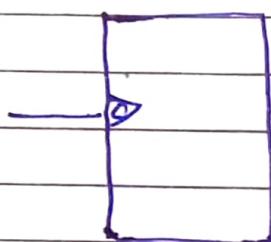
-ve level

-ve level

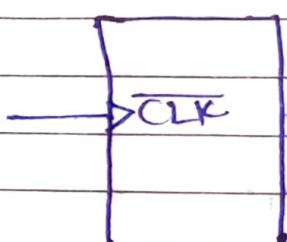
-ve level



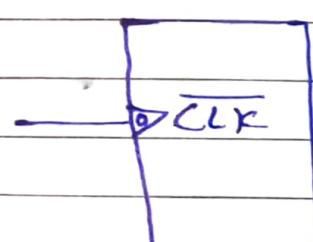
+ve Edge



-ve edge



-ve Edge



-ve Edge.

Race Around Condition :- in JK flip flop

condition 1:- Level triggered J-K flip flop

condition 2:- when $J=K=1$ (Toggle Mode)

condition 3:- $T_w \gg T_d$

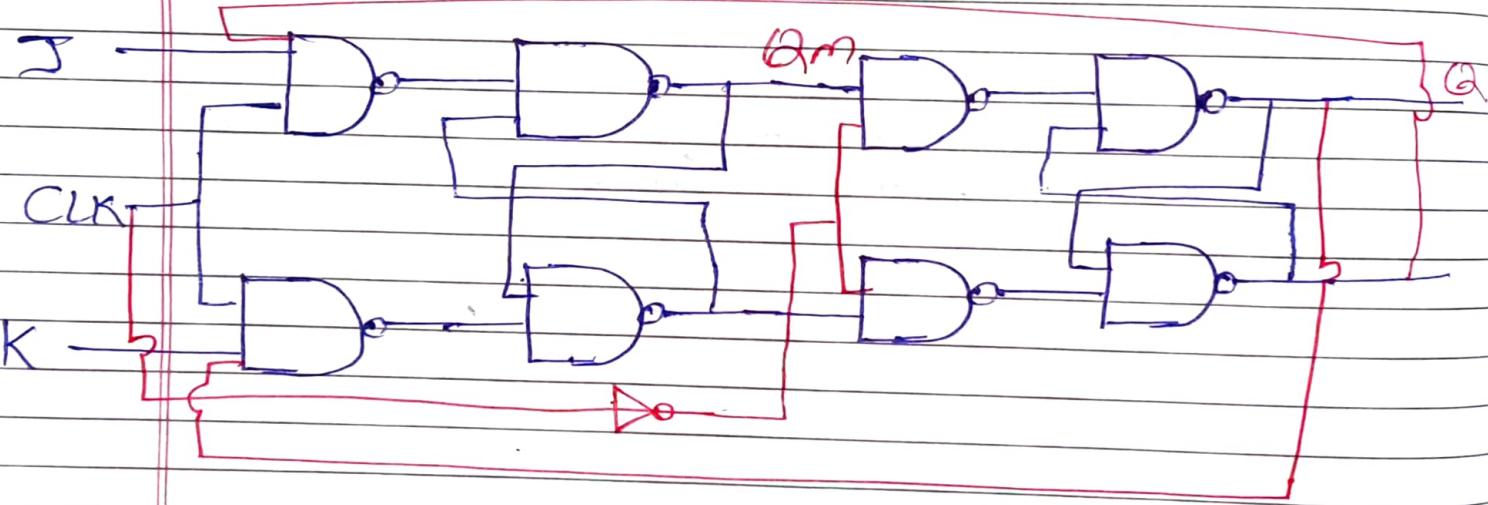
if all three conditions are true then the JK flip flop is in race around condition.



Master slave JK flip flop -

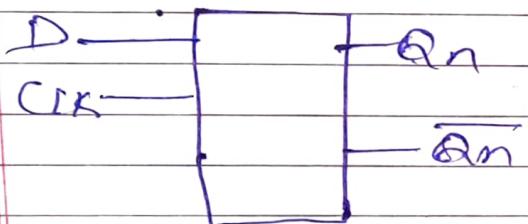
Master slave

J K flip flop is to resolve the race around condition of J K flip flop.



(*) D flip flop or Data flip flop:-

It is simply known as storage.
Block Diagram :-



Truth Table:-

D	Q_{n+1}
0	0
1	1

Characteristic Table:-

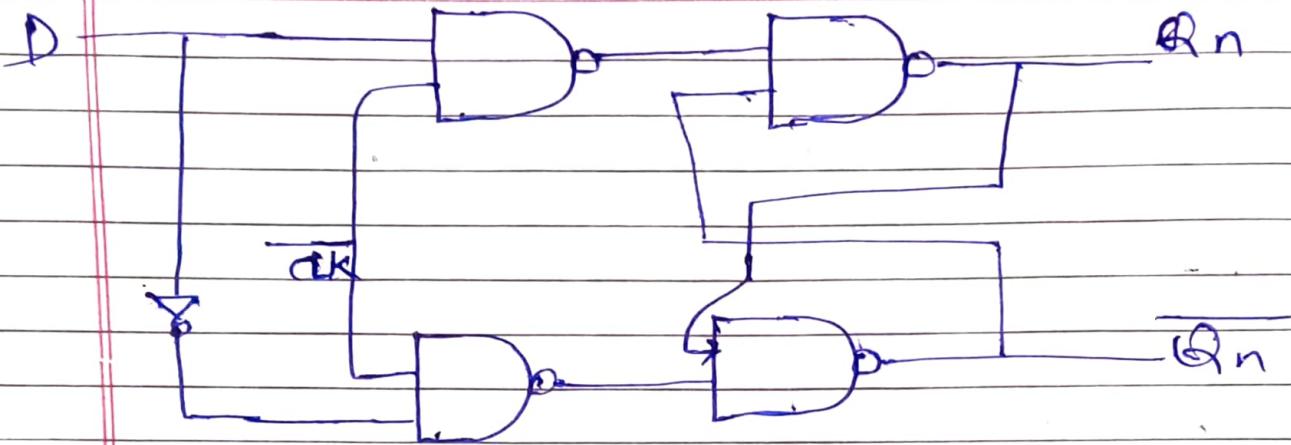
D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

Characteristic equation:-

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

Excitation Table:-

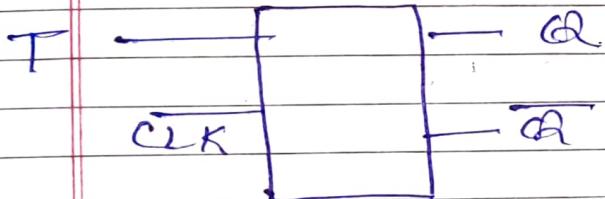
Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1



D flip flop by SR flip flop.

(*) T flip flop :- Also known as Toggle flip flop

• Block diagram:-



- Truth Table :-

T	Q_{n+1}
0	Q_n
1	\bar{Q}_n

(*) Characteristic Table :-

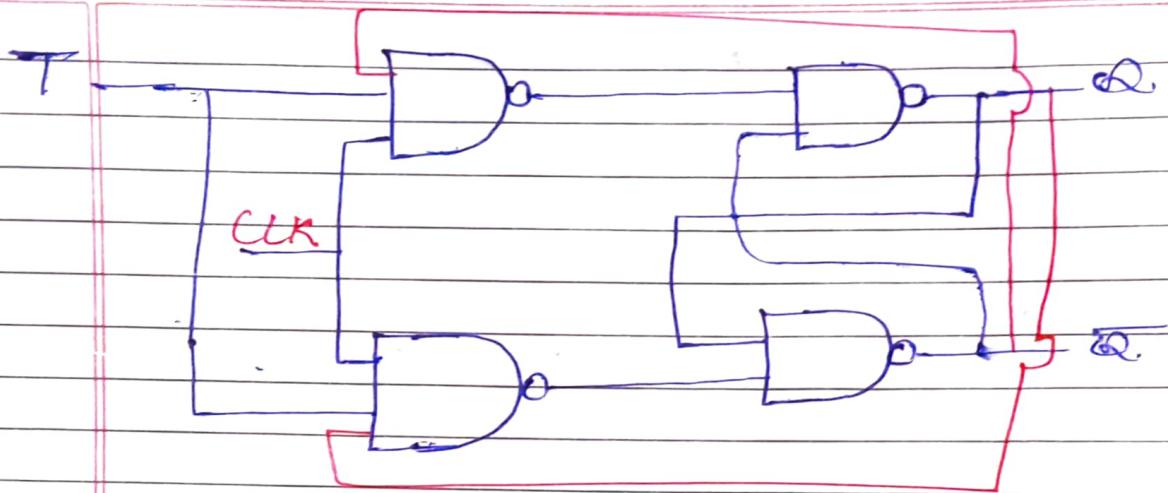
T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Characteristic Table :-
equation :-

$$\text{Q}_{n+1} = T \oplus Q_n$$

(*) Excitation Table :-

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



Q) Conversion of flip flops:-

SR to D flip flop:-

SR \rightarrow given

D \rightarrow Required.

Use of SR - Excitation Table
~~or~~ D - characteristic Table

Q) D - flip flop Characteristic Table + S.R. Excitation

D	Qn	Qn+1	S	R	Qn	Qn+1	S	R
0	0	0	0	X	0	0	0	X
0	1	0	0	1	0	1	1	0
1	0	1	1	0	1	0	0	1
1	1	1	X	0	1	1	X	0

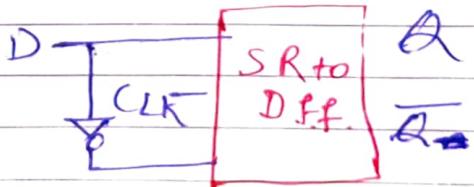
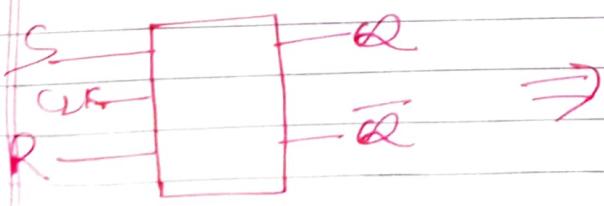
Characteristic equation according to 'S' \neq 'R'

S	Qn	Qn
D		
D	X	X

$$\boxed{S = D}$$

R	Qn	Qn
D	X	D
D		

$$\boxed{R = \bar{D}}$$



Convert T to JK flip flop:-

T-given \rightarrow Excitation Table

JK-required \rightarrow characteristic Table.

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

Q_n	Q_{n+1}	$Q_{n+1} \bar{Q}_n$
0	0	0
0	1	1
1	0	1
1	1	0

$\Rightarrow K_{an}$

\bar{J}	\bar{K}_{an}	K_{an}	K_{an}	K_{an}
J	D	D	I	I

$$T = K Q_n + J \bar{Q}_n$$



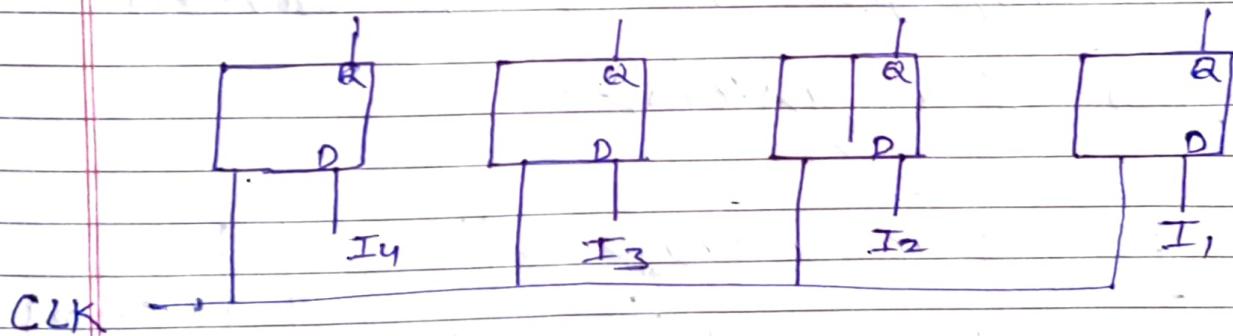
Registers:- (*) It is a group of flip flops.

(*) It is used to store information.

(*) n-bit register will contain n flip flops & will store n-bit word.

In addition to flip flops, a register may have combinational gates that perform data processing tasks.

The gates control when & how new information is transferred into the register.



4-bit Register

A register can be 2-bit register, 4-bit register, 8-bit register.

There are the following operations which are performed by the registers:-

(1) Fetch:- (*) It is used to take the instructions given by the users.

(2) To fetch the instruction stored in to the main memory.

(2) Decode:- The decode operation is used to interpret the instructions. The operation performed on the instructions is identified by the CPU.

In simple words, the decode operation is used to decode the instructions.

(3) Execute:- This operation is used to store the result produced by the CPU into the memory. After storing the result, it is displayed on the user screen.

Types of Registers:-

PC (Program Counter)		
AR (Address Register)	CPU 16 bit	Memory 4096 words 16 bits per word
IR (Instruction Register)		
TR (Temporary Register)		DR (Data Register)
I N P R	B U T R	A.C. (Accumulator)

MAR or Memory Address Register:-

It is a special type of register that contains the memory address of the data & instruction.

The main task of the MAR is to access instruction & data from memory in the execution phase. The MAR stores the address of memory location where the data is to be read or to be stored by the CPU.

Program Counter :- The program counter is also called an instruction address register or instruction pointer. The next memory address of the instruction, which is going to be executed after completing the execution of current instruction is contained in the program counter.

means the program counter contains the memory address of the location of the next instruction.

Accumulator Register :- CPU mostly used Accumulator Register. It is used to store the system result. All the results will be stored in the accumulator reg. whenever CPU produces some results.

MDR or Memory Data Register :-

It is a part of computer's control unit. It contains the data that we want to store in the computer storage or data fetched from the computer storage.

The MDR works as a buffer that contains anything for which the processor is ready to use it. The MDR contains the copied

data of the memory for the processor.

The data which is to be read out or written into the address location is contained in the Memory Data Register.

Index Register:- It is a hardware element that holds the number. The number adds to the computer instruction's address to create an effective address.

In CPU, the index register is a processor register used to modify the operand address during the running program.

Memory Buffer Register:- MBR

It contains the metadata of the data & instruction written in or read from M/R.

In other words, it is used to store the data upcoming data/instruction from the M/R & going to M/R.

Data Register:- The data register is used to temporarily store the data. This data transmits to or from a peripheral Device.

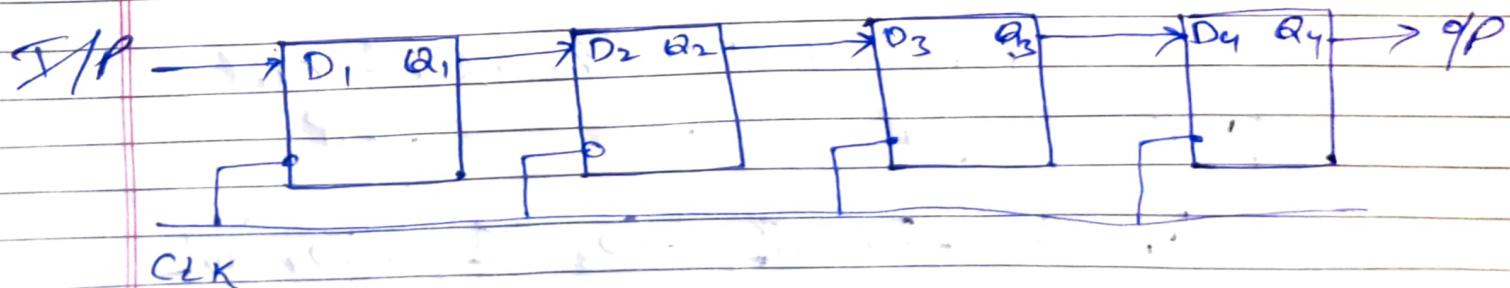
Shift Registers:-

- (*) Shift Registers are used to implement arithmetic operations.
eg - Left shift & Right shift.
- (*) Basic flip flop used in the register is D flip flop.

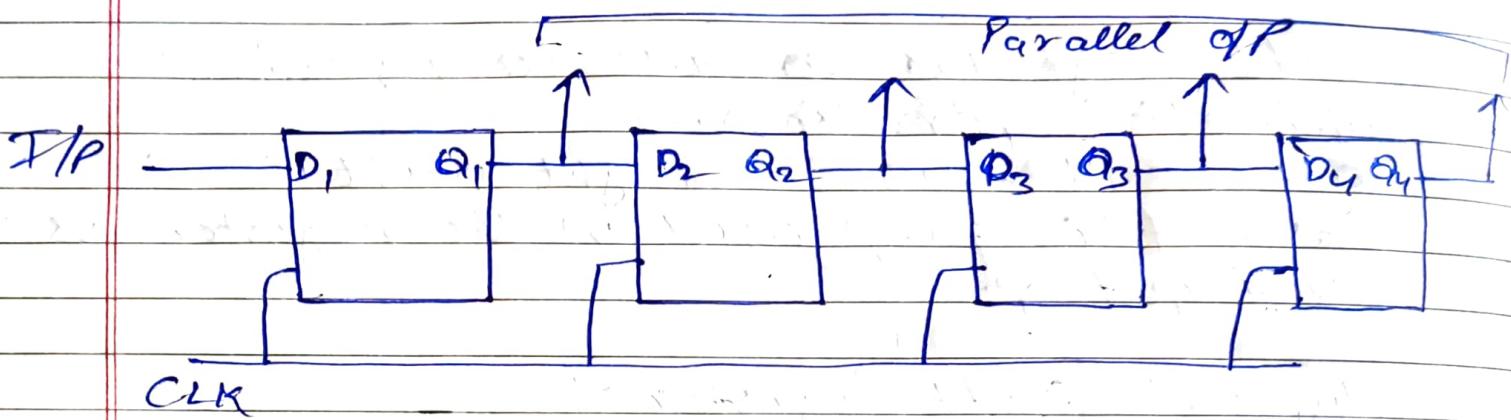
Types of Shift Registers:-

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

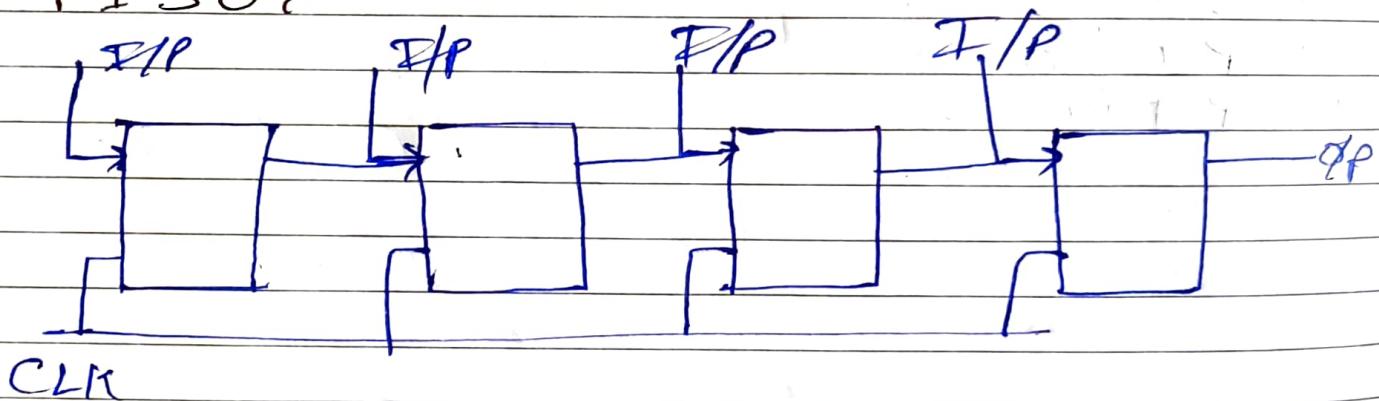
(1) SISO :- Serial Input serial Output



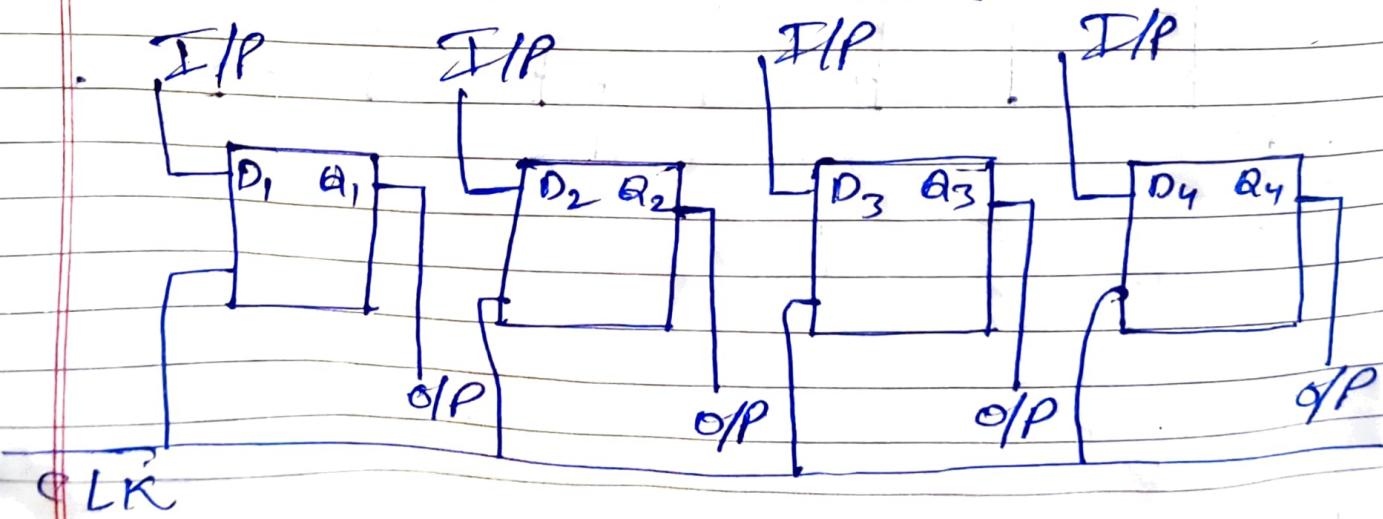
(2) SIPO :- Serial In + Parallel out



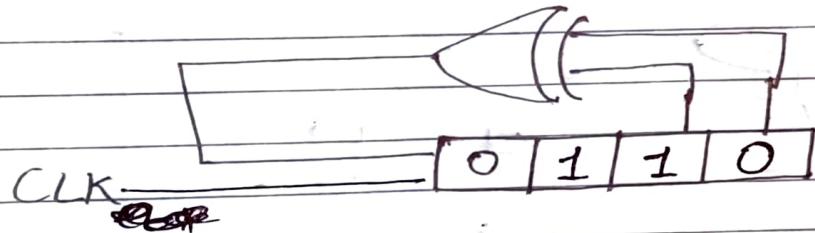
(3) PISO:-



(4) PIPO:- Parallel IN + Parallel out



MODE	Clocks needed for n-bit Shift Register		
	Loading	Reading	Total
SISO	n	n-1	2n-1
SIPO	n	0	n
PISO	1	n-1	n
PIPO	1	0	1



In a 4 bit right shift register. How many clock pulses are required to change the content of register all 1's?

Introduction to counters:- It is a sequential circuit.

- (*) Flip flops are essential part of counter.
- (*) used for counting of clock pulses.
- (*) If pulses are generated by an event, then it is called Event counter.
- (*) n No. of flip flops can count up to 2^n No. of pulses.

Types of counters:-

Asynchronous counter
or Ripple counter

Synchronous Counter
or Parallel counters

(1) flip flops are connected in a way that QP of 1st flip flop drives the clock of the next FF.

There is no connection between QP of the 1st flip flop & the clock of the next flip flop.

(2) Flip flops are not clocked simultaneously.

Flip flops are clocked simultaneously.

(3) Ckt is simple for more no. of states

Ckt becomes complicated as no. of states increases.

(4) Speed is slow as the clock is propagated through no. of stages

Speed is high as the clock is given at the same time.



→ we Toggle mode
JK, T flip flop

any flip flop. Can be used

Synchronous/Asynchronous counters

↓
UP counter

↓
down counter

↓
UP/down counter