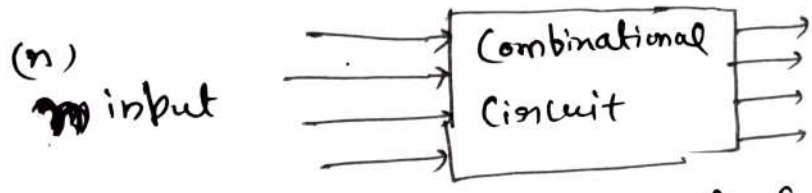
# Unit 3

## Difference between Combinational and Sequential Circuit.

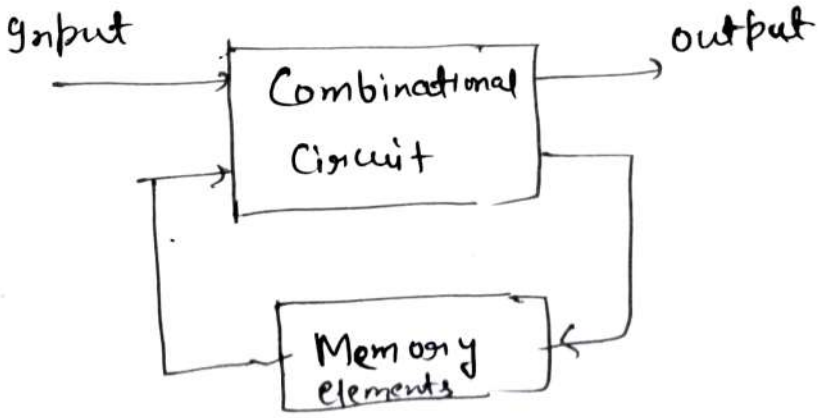
Digital Circuit is classified into Combinational Circuit and Sequential Circuit.

Combination logic output depends on the present inputs levels, whereas sequential logic output not only depends on the input levels, but also stored levels (Previous output history).

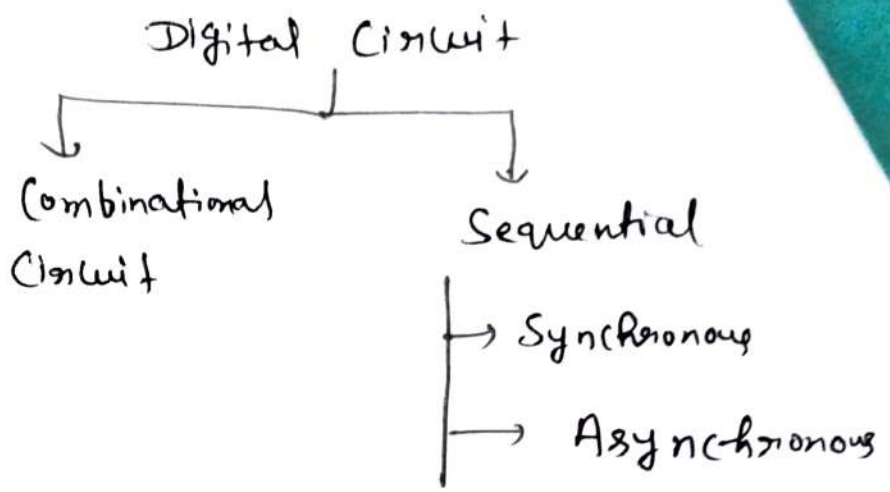
The memory elements are devices capable of storing binary information. The example of Combinational Circuit is Adder, Subtraction, Encoder, decoders etc. The example of sequential Circuit is flip flop, register, Counter.




Block diagram of Combinational Circuit




Block diagram of Sequential Circuit




A Sequential Circuit can further be categorized into Synchronous and Asynchronous :-

(1) Synchronous Sequential Circuit → Output changes at discrete interval of time. Examples of Synchronous Sequential Circuit are FLip flop, Synchronous Counter.

(2) Asynchronous Sequential Circuit → Output Can changed at any instant of time by changing input, Example of asynchronous Sequential Circuit is Asynchronous Counter.

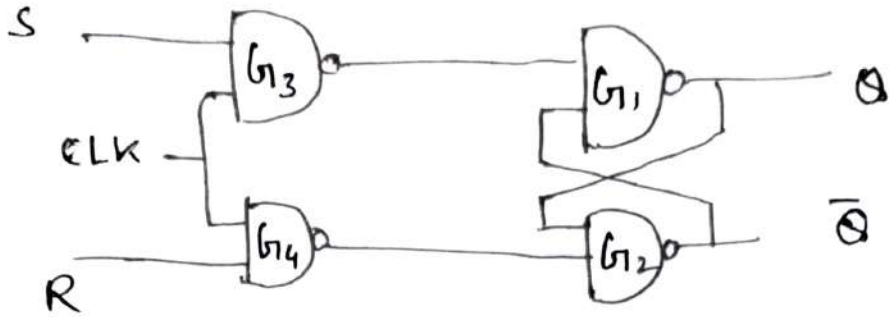
In Synchronous Sequential Circuit, Synchronization is achieved by a timing device called a clock pulse generator. Clock pulses are distributed throughout the system in such a way that the flip flops are affected only with arrival of synchronization pulse. Synchronous Sequential Circuits that use clock pulses in inputs are called clocked sequential Circuits.

Clock Pulse

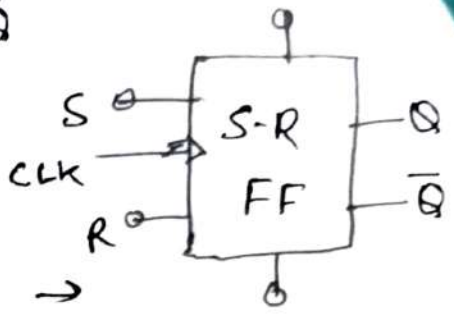
Flip flop → A circuit that changes from 1 to 0 or 0 to 1 when current is applied. It is one bit storage location. Latches and flip-flops are basic elements for storing information. One latch or flip-flop can store one bit of information.

A latch output changes ~~imp~~ immediately when input changes, while flip-flops output only changes when the clock signal changes. A latch does not have clock signal while a flip flop always does. The main difference between a latch and a flip flop is that a latch is level triggered while a flip flop is edge triggered.

① Clocked S-R Flip flop →




S-R Flip flop




Logic Symbol of S-R FF

Truth table of S-R Flip flop →

| CLK | Inputs | | Outputs |
|---|---|---|---|
| | $S_n$ | $R_n$ | $Q_{n+1}$ |
| 1 | 0 | 0 | $Q_n$ |
| 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | Invalid |

If a clock Pulse is Present (CLK=1), If S=1, R=0, output $G_3$ will be 0 and output $G_4$ will be 1. Since one of the input of $G_1$ is 0, its output will 1 and $\bar{Q}=0$. Similarly, S=0, R=1 then output will be $Q=0$ and $\bar{Q}=1$.

The first of these input conditions S=1, R=0 makes $Q=1$, which is referred to as set state, whereas second input

Condition ($S=0, R=1$) makes $Q=0$ which is refered to as the reset state or clear state.

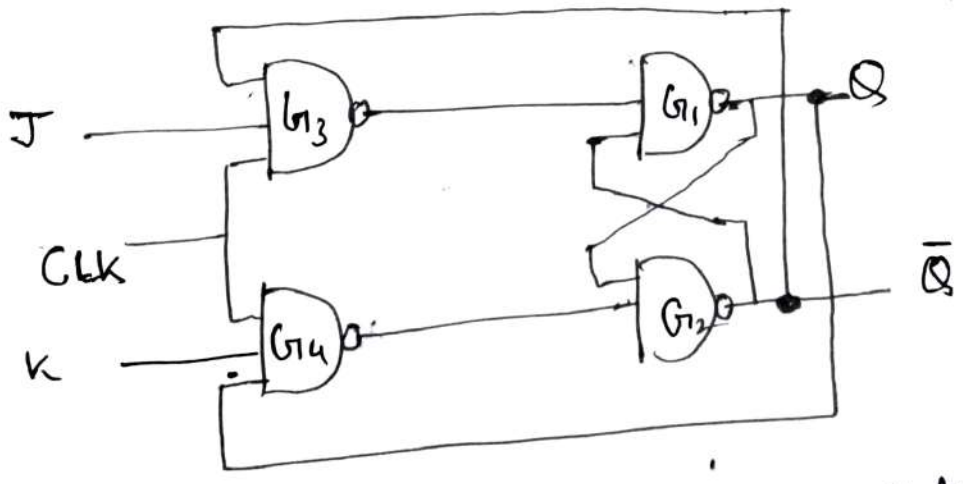
If input conditions are changed from $S=1, R=0$ to $S=R=0$ or from $S=0, R=1$ to $S=R=0$. The output remains unaltered.

If $S=R=1$, both outputs Q and $\bar{Q}$ will try to become 1 which is not allowed therefore, this input condition is prohibited.

The circuit responds to input S and R, only when clock is present.

$S_n$ and $R_n$ denote the inputs and $Q_n$ the output during bit time $n$. $Q_{n+1}$ output Q after pulse passes i.e. in time $n+1$.

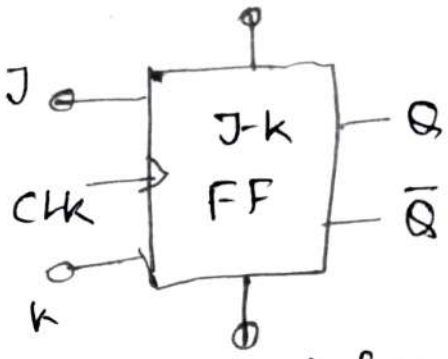
## ② J-K Flip flop →




A J-K Flip flop using NAND gate

Truth table of J-k FF

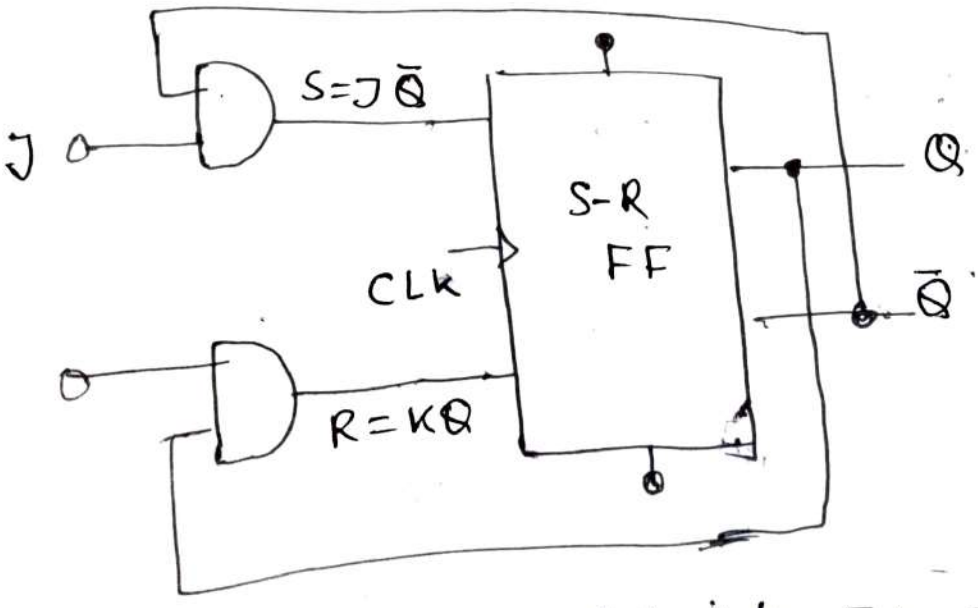
| Inputs | | Output |
|---|---|---|
| $J_n$ | $K_n$ | $Q_{n+1}$ |
| 0 | 0 | $Q_n$ |
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 1 | $\bar{Q}_n$ |




Logic Symbol of J-k FF

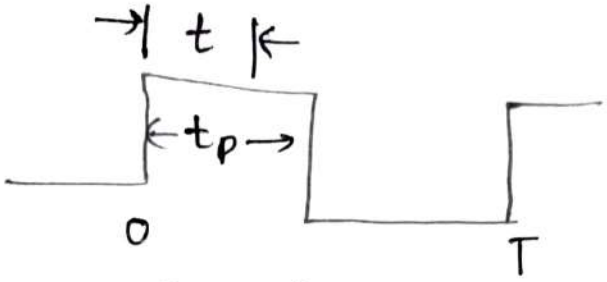
The uncertainty in the state of an S-R FFs when $S_n = S_R = 1$ can be eliminated by Converting it into J-k FFs. The data inputs are J and K which are ANDed with $\bar{Q}$ and Q respectively to obtain S and R inputs

$$S = J\bar{Q}$$
$$R = KQ$$




S-R FFs Converted into J-K FFs

## Race around Condition →




A clock Pulse

The difficulty of both inputs ($S=R=1$) being not allowed in S-R FFs is eliminated in a J-K FFs by using the feedback connection from outputs to inputs of the bates $G_3$ and $G_4$.

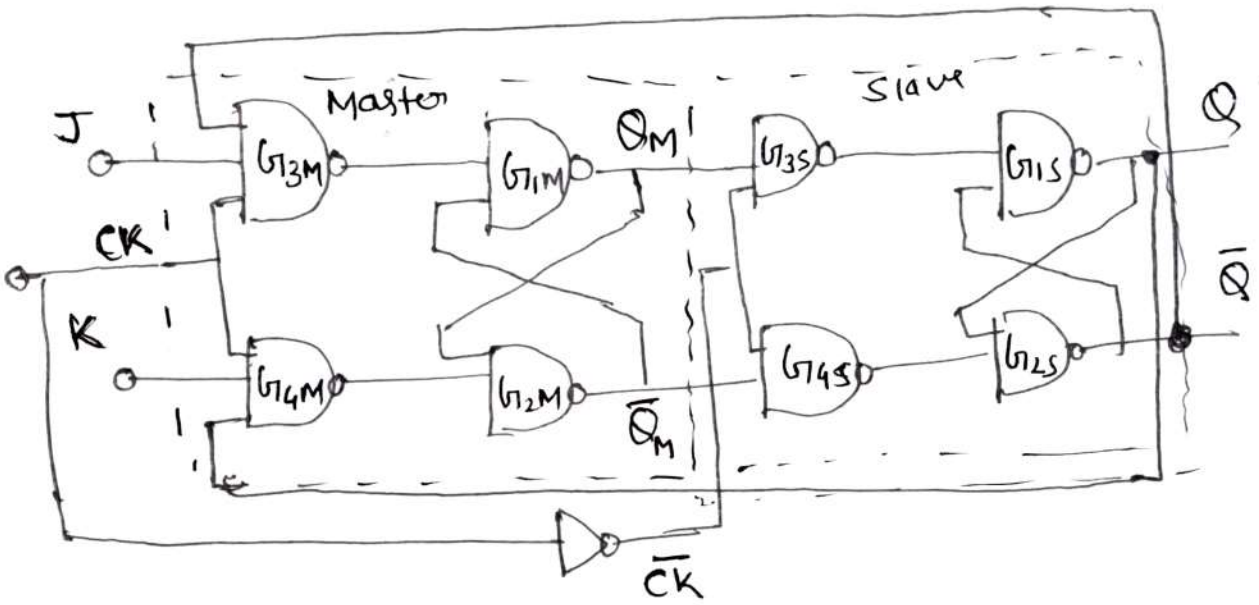
Input do not change during clock pulse ($Ck=1$) which is not true because of feed back Connections. For example, $J=K=1$ and $Q=0$. After a time interval $\Delta t$, the output will change $Q=1$. Now we have $J=K=1$ and $Q=1$ and after another time interval $\Delta t$ output will change back to $Q=0$.

Hence, we conclude that for the duration $t_p$ of clock pulse, the output will oscillate back and forth between 0 and 1. At the end of clock pulse $Q$ is uncertain. This situation is referred to as race around Condition.

A more practical method for overcoming this difficulty is the use of master-slave (MS) FF. The race around condition can be avoided if

$t_p < \Delta t < T$.

## The Master Slave JK Flip flop →



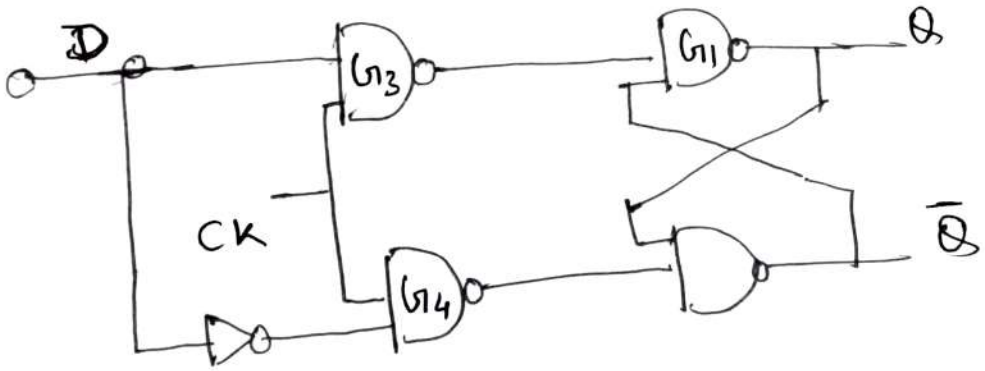

Positive clock pulses are applied to the first FFs and clock pulses are inverted before these are applied to second FFs. When CK = 1, the first FF is enabled and output $Q_M$ and $\bar{Q}_M$ respond to input J and K. At this time, second FF is inhibited because its clock is Low ($\bar{CK} = 0$). When CK goes low ($\bar{CK} = 1$), the first FF is inhibited and second FF is enabled, because now its clock is high ($\bar{CK} = 1$).

Therefore, the output Q and $\bar{Q}$ follows the output $Q_M$ and $\bar{Q}_M$ respectively.

Since second FF simply follows the first one, it is referred to as slave and the first one as master. Hence, this configuration is referred to as master-slave (M-S) FF.

In this circuit, the input $G_{13}$ and $G_{4M}$ do not change during clock pulse, therefore race around condition does not exist.

## D- Flip flop →



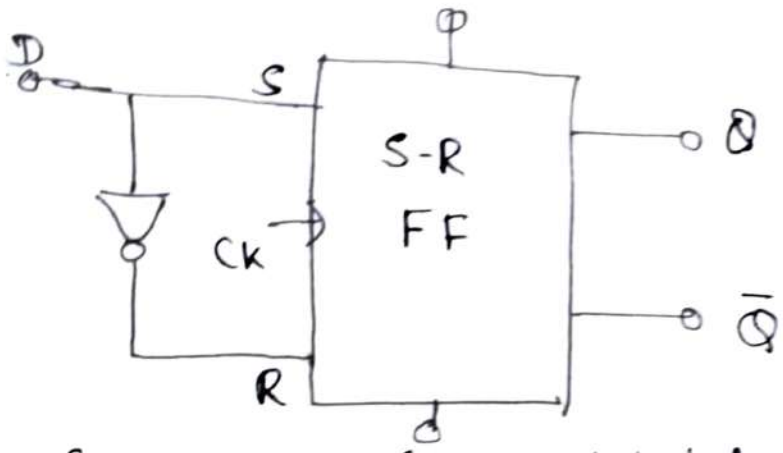

D- FF

If we use only the middle two rows of the truth table of S-R or J-K FF. We obtain D- FF. It has only one input referred to as D input or data input. The output $Q_{n+1}$

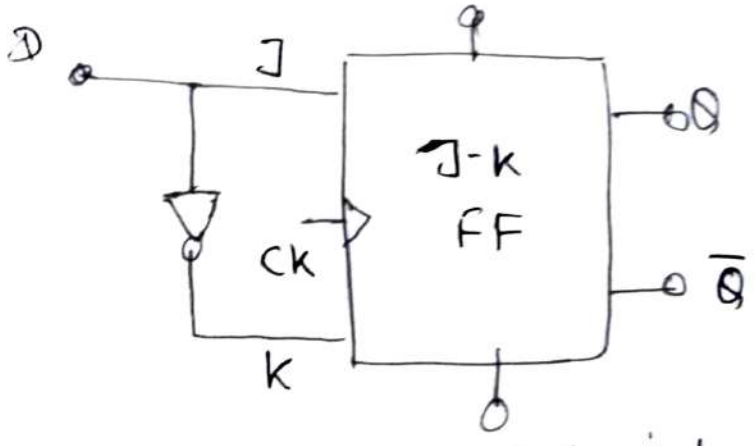
at the end of clock pulse equals the input $D_n$ before the clock pulse.

Truth table of D- FF →

| Input | output |
|---|---|
| $D_n$ | $Q_{n+1}$ |
| 0 | 0 |
| 1 | 1 |



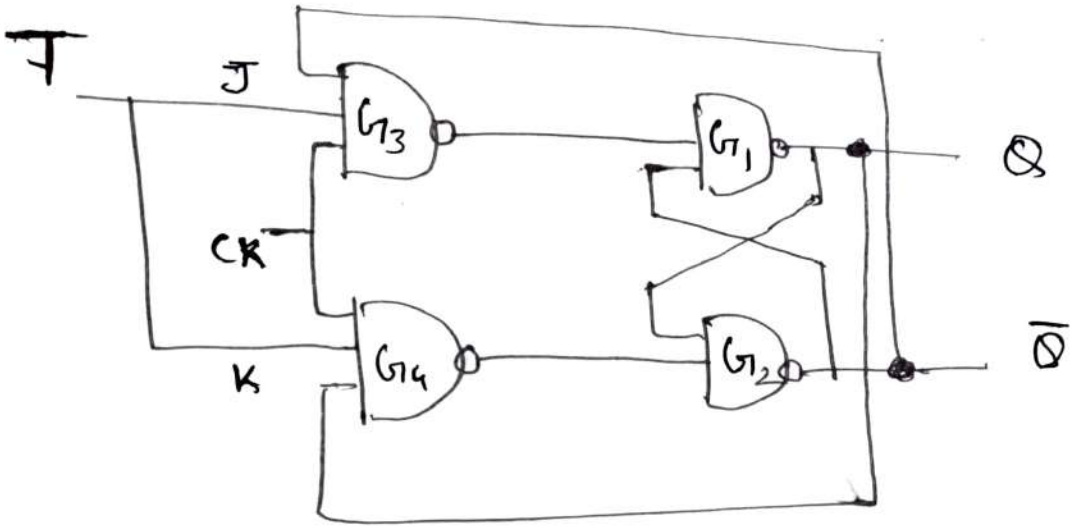

S-R FF Convererted into D-FF




J-K FF Converted into D FF

## T- Flip Flop →

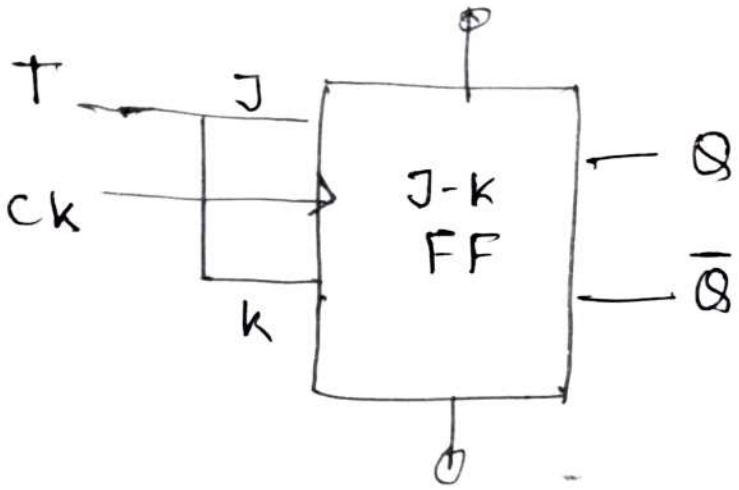
If in a J-K Flip-flop, J = K, the resulting FF is referred to as T- FF. It has only one input, referred to as T- input.




## Truth table of T- FF →

| Input | Output |
|---|---|
| $T_n$ | $Q_{n+1}$ |
| 0 | $Q_n$ |
| 1 | $\overline{Q}_n$ |

From the truth table it is clear that if T = 1, it acts as toggle switch. For every clock pulse output Q change.




A J-K FF Converted into T-FF

An S-R FF Can not be Converted into T FF Since S = R = 1 is not allowed.

Register → A flip-flop can store only one bit of data a '0' or a '1', it is referred to as a single bit register. When more bits of data are stored, a number of FFs are used. A register is a set of FFs used to store binary data. The storage capacity of register is the number of bits (1s and 0s) of digital data it can retain. Loading may be serial or Parallel. In serial loading, data is transfered into register in serial form, ie. one bit at a time whereas parallel loading, whereas the data transferred into register in parallel form meaning that all the FFs are triggered into their new states at same time.

A register may output data either in serial form or in parallel form. Serial output means that the data is transferred out of register, one bit at a time serially. Parallel output means that the entire data stored in the register is available in parallel form, and can be transfered out at same time.

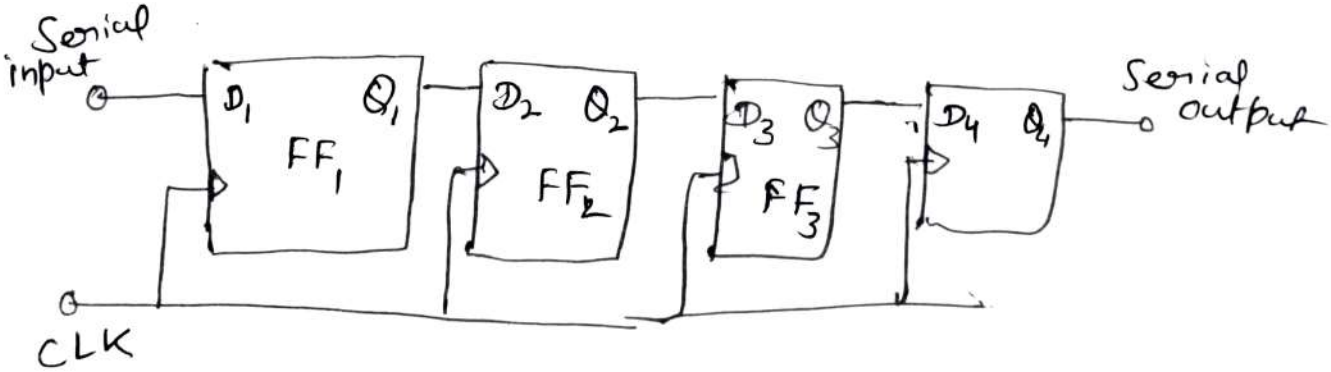
## Data transmission in shift register —

A number of FFs Connected together such that data may be shifted into and shifted out of them is called a shift register. Data may be shifted into or out of register either in serial form or parallel form. So their are four basic type of shift register

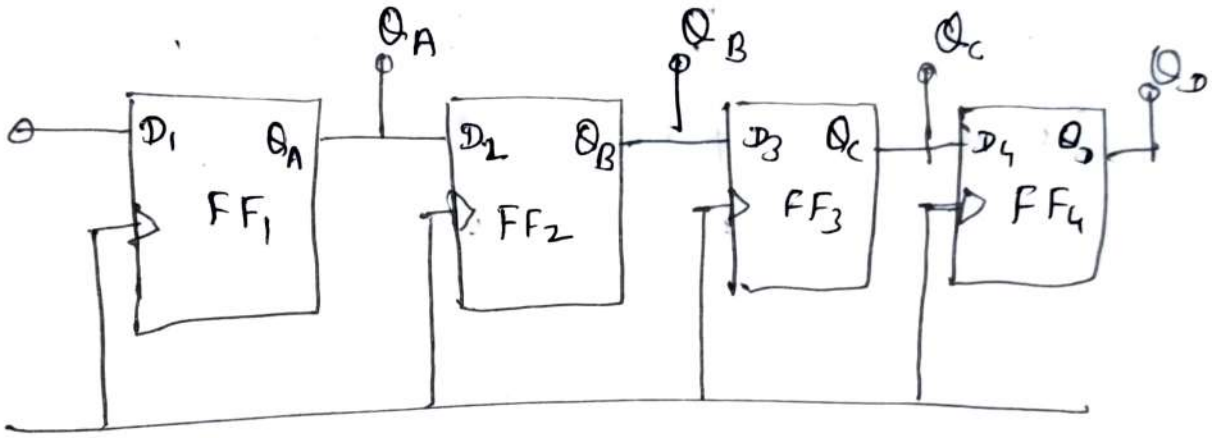
(I) Serial in serial out
(II) Serial in Parallel out
(III) Parallel in serial out
(IV) Parallel in Parallel out

### (I) Serial in Serial out Shift register →

This type of shift register accepts data serially i.e. one bit at a time and also output data serially.



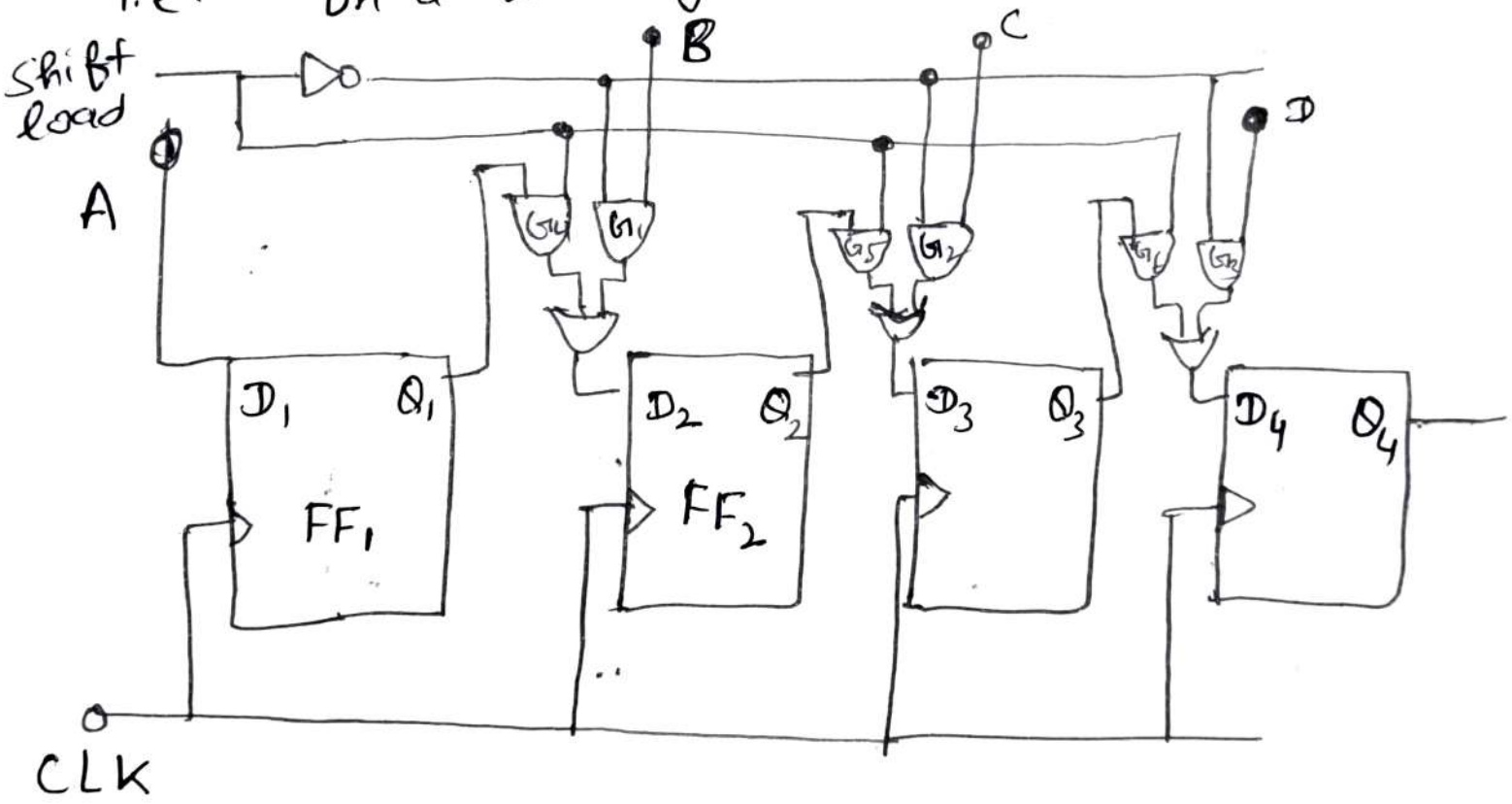
## (2) Serial in Parallel out shift register



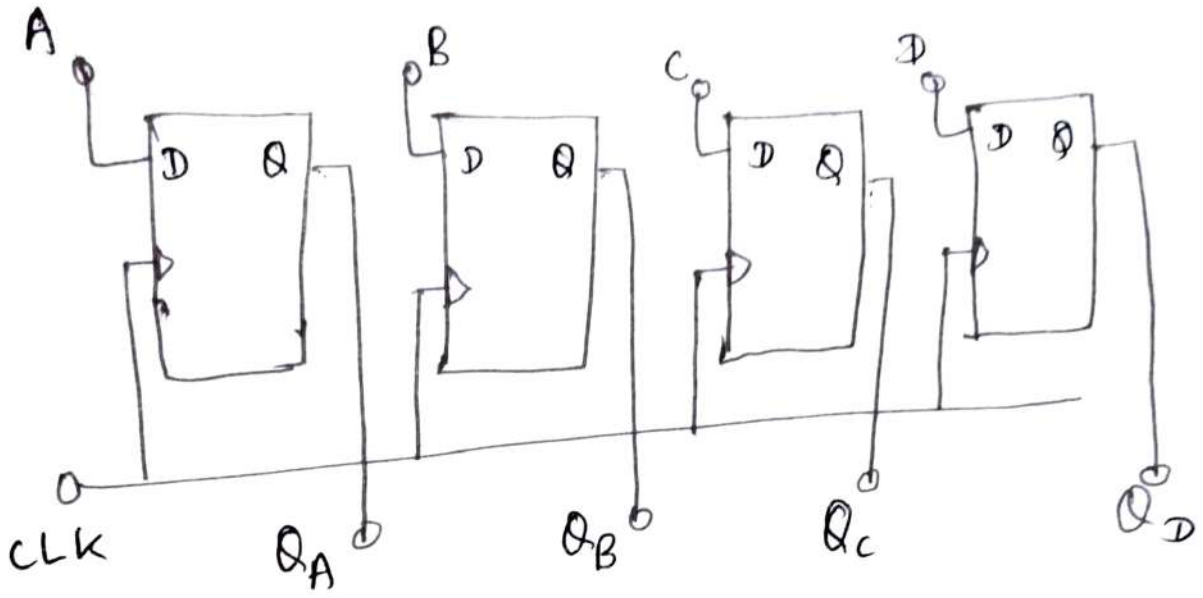

In this type of register, the data bits are entered into register serially but data stored in register is shifted out in parallel form. Once data bit are stored, each bits appears on its respective output line and all bits are available simultaneously, rather than on a bit by bit basis as with serial output.

## (3) Parallel in Serial Out shift register

The data bits are entered simultaneously into their respective stages on parallel lines, rather than on a bit by bit basis on one line as with serial data inputs but data are transfered out of register serially i.e. on a bit by bit basis on single line



(4) Parallel in Parallel out shift register →



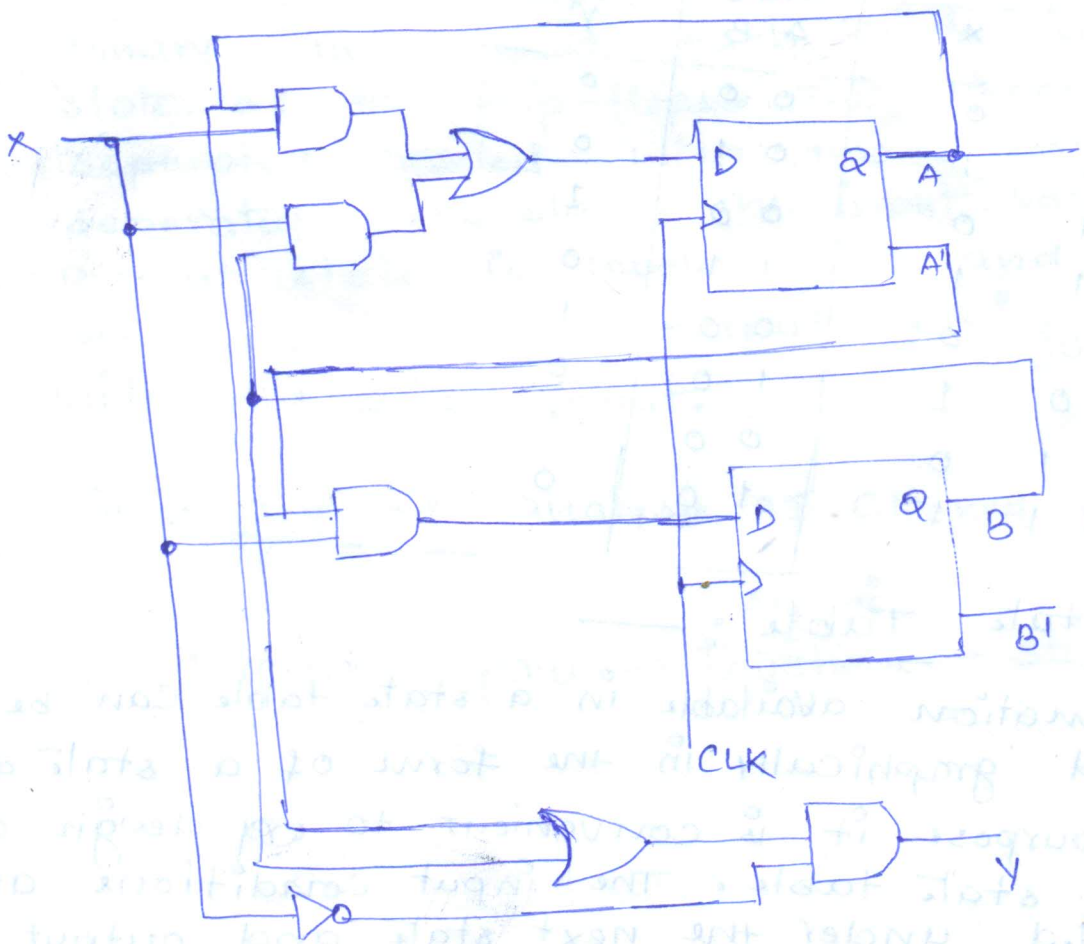

The data entered into register in parallel form and also data is taken out of register in parallel form.

# Analysis of Clocked Sequential circuits:-

Analysis of a clocked sequential circuit describes what a given circuit will do under certain operating conditions. The behaviour of a clocked sequential is determined from the inputs, the outputs and the state of the Flip-Flops. The output and next state both are a function of the inputs and present state. The analysis of a sequential circuit consists of obtaining a table or diagram for the time sequence of inputs, outputs and internal states.

A logic diagram is recognized as a clocked sequential circuits if it includes flip-flops with clock inputs. The logic diagram may or may not include combinational logic gates.

**State Equation:-** A state equation (transition equation) specifies the next state as a function of the present state and inputs.



D input determines the next state value. So

$$A(t+1) = A(t)x(t) + B(t)x(t) = Ax + Bx$$

$$B(t+1) = A'(t)x(t) = A'x$$

The left side of equation, denotes the next state of the flip-flop one clock edge later. The right side of the equation specifies the present state and input condit that make the next state equal to 1.

The present state value of the output can be expressed as $Y(t) = [A(t) + B(t)]x'(t)$

$$= (A+B)x'$$

State Table:-

The time sequence of inputs, outputs and flip-flop states can be enumerated in a state table (transition table).

| Present state | | Input | Next state | | output |
|---|---|---|---|---|---|
| A | B | x | A | B | Y |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 |

Reduced state Table:-

The information available in a state table can be represented graphically in the form of a state diagram For this purpose it is convenient to design a reduced state table. The input conditions are enumerated under the next state and output sections.

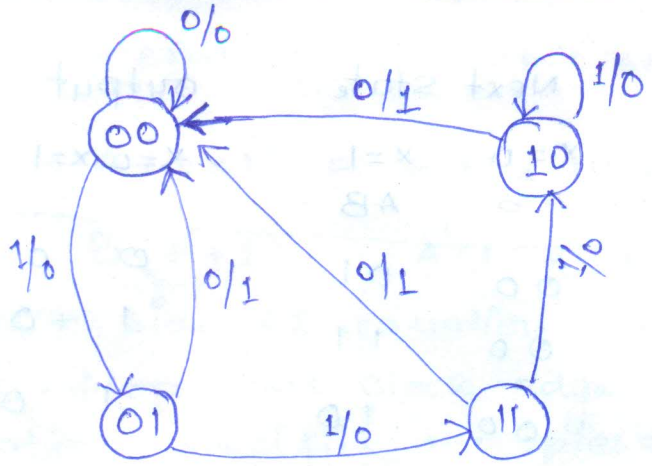
| Present state | | Next state | | | output | |
|---|---|---|---|---|---|---|
| A | B | x=0 | | x=1 | x=0 | x=1 |
| | | A | B | AB | | |
| 0 | 0 | 0 | 0 | 01 | 0 | 0 |
| 0 | 1 | 0 | 0 | 11 | 1 | 0 |
| 1 | 0 | 0 | 0 | 10 | 1 | 0 |
| 1 | 1 | 0 | 0 | 10 | 1 | 0 |

State diagram :-

In state diagram, a state is represented by a circle, and the transitions between states are indicated by directed lines connecting the circles. Each line originates at a present state and terminates at a next state, depending on the input applied when the circuit is in the present state. The state diagram provi the same information as the state table. The binary number inside each circle identifies the state of the flip-flops. The directed lines are ~~seperated~~ lebeled with two binary numbers seperated by a slash. The input value during the present state is lebeled first and the number aft the slash gives the output during the present st with the given input.

Summary of analysis of Clocked Sequential Circuit

Circuit diagram → Algebric Equations → state table → state diagra




(State diagram)