Unit V: Sequential Circuit

A sequential circuit is widely used for storing and processing information such as retaining and manipulating data which makes them an essential component in various electronic devices in the realm of digital electronics. Sequential circuits process data in a sequential manner which makes them essential building blocks of many applications computer processors, memory systems, and communication networks by enabling memory retention and feedback capabilities.

Definition of Sequential Circuit

Unlike a combinational circuit which operates solely based on the present, a sequential stores and processes digital information in a sequential manner i.e. it has an additional element called memory which enables it to consider not only the present but also the past inputs or states. This memory element plays a vital role as it enables the processing of streams of data.

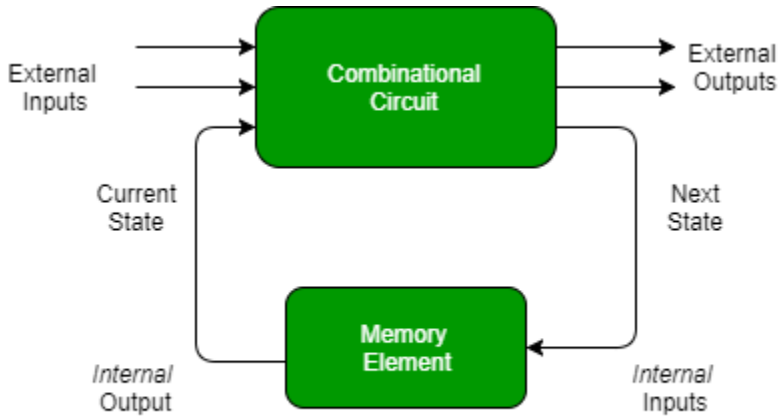


Figure: Sequential Circuit

As shown in the above figure, there are two types of input to the combinational logic:

- External inputs which are not controlled by the circuit.
- Internal inputs, which are a function of a previous output stat**e.**

Examples of Sequential Circuits:

Sequential circuits are used to develop various building units which are discussed below:

- **Flip-Flops:** A flipflop is used as a memory element to store bitwise binary information (0 and 1 only). Flipflops are the main form of sequential circuit where two stable states (set and reset) are working sequentially based on trigger condition (known as clock pulse).

- **Counters:** Counters are specially designed sequential circuits which are simply used for counting pulses or events or various sequences like binary, decimal or Gray code etc. Counters can be found in digital clocks, frequency dividers etc.
- **Shift Registers:** Shift registers are nothing but chain of multiple flipflops which are connected to each other serially to store and shift data serially. It listens to the clock pulses to decide to perform shifting operation. Shift registers are used in data storage and communication system and parallel-to-serial(P2S) or serial-to-parallel(S2P) conversion systems.

5.1 Types of Sequential Circuit

There are two types of sequential circuits available:

1. Synchronous Sequential Circuit

Synchronous circuits synchronize with either positive edge or negative edge of the clock signal, that means, the outputs of synchronous sequential circuits change or affect at the same time. These circuits use clock signal and level input (or pulsed with restrictions on pulse width and circuit propagation). Since they wait for the next clock pulse to arrive to perform the next operation, so these circuits are a bit slower compared to asynchronous. Level output changes state at the start of an input pulse and remains in that until the next input or clock pulse. The synchronous sequential circuit can be locked or unlocked (or pulsed).

For more detail you can refer Synchronous Sequential Circuits in Digital Logic article.

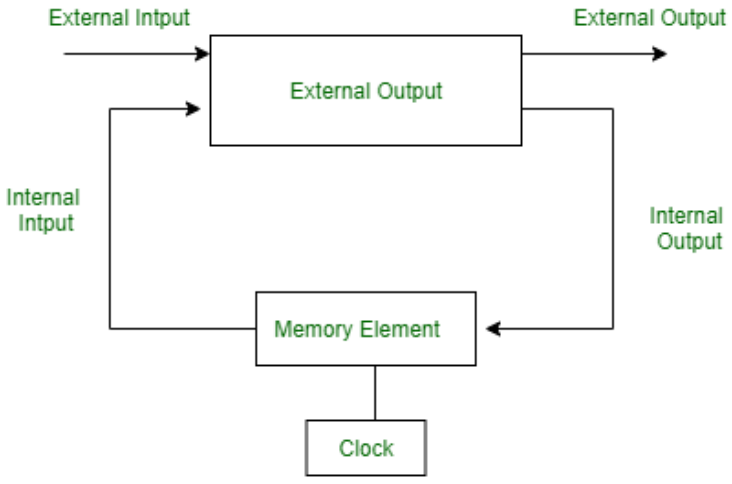


Figure - Synchronous Sequantial Circuit

2. Asynchronous Sequential Circuit

Asynchronous circuits do not synchronize with positive edge or negative edge of the clock signal, that means, the outputs of asynchronous sequential circuits do not change or affect at the same time and change their state immediately when there is a change in the input signal. So, these circuits are faster and independent of the internal clock pulses. But these circuits have uncertainty in the outputs and are difficult to design.

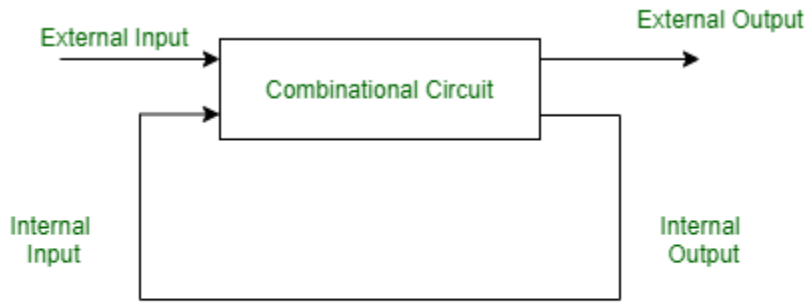


Figure - Asynchronous Sequential Circuit

Applications of Sequential Circuit

Many applications have use of sequential circuits like:

- **Computer Processors:** Most of the modern CPUs contain sequential circuits to fetch instructions, decode data and execute processes. The reason of sequential circuits within CPU is to maintain the flow of work or operations of CPU in sequential manner (one after another to avoid overlapping).
- **Memory Systems:** RAM and ROM are the backbone of memory of computer. And both of these contains sequential circuits for better utilization during storing and retrieving data.
- **Communication Systems:** All communication system consists of sequential circuit for data encoding, decoding and synchronization and ensures secure data transmission and better-quality error detection.

Advantages of Sequential Circuit

Modern technology depends upon sequential circuits for its wide range of advantages which are listed below:

- **Memory Retention:** Sequential circuits are capable of retaining information even if the output changes by storing past states.

- **Feedback Capabilities:** Not only it saves the previous input states but also it can store past output states which enables feedback response system, necessary for data control and communication systems.
- **Complex Functionality:** Developers can design complex logical units by connecting multiple Flipflops, counters or shift registers i.e. sequential circuits which can perform high end logical tasks.

Disadvantages of Sequential Circuit

Though sequential circuits are widely used, still it faces some drawbacks which are listed below:

- **Increased Complexity:** Sequential circuits are comparatively complex in nature than combinational circuits as it has memory elements and feedback paths. So, developer need to give special care during designing complex circuits with it.
- **Higher Power Consumption:** As sequential circuits contain memory elements it consumes high power during data retaining. Moreover, when it is in idle state still it listens for any data changes and make synchronizations. It constantly consumes high power.
- **Synchronization Challenges:** All synchronization is depending upon the clock pulses. So, proper design is required for correct data storing after data alterations especially where the data changes are more often.

5.2 Triggering of Flip flops

In digital circuits, two methods of triggering are possible, namely edge triggering and level triggering, which trigger the signal to switch from one state to the other. Both form part of digital electronics and help in increasing throughput and controlling the timing of operations in a given system.

What is Clock Pulse?

A clock pulse represents a series of periodic pulses that control the time of operations in digital circuits. Synchronization is used in conjunction with sequential circuits as a clock signal to control the changing of states at certain instances. It is possible to use clock pulses to control other triggered devices such as flip-flop and counters because these elements should change their state only within certain time, for instance at the rising or falling edge of the pulse.

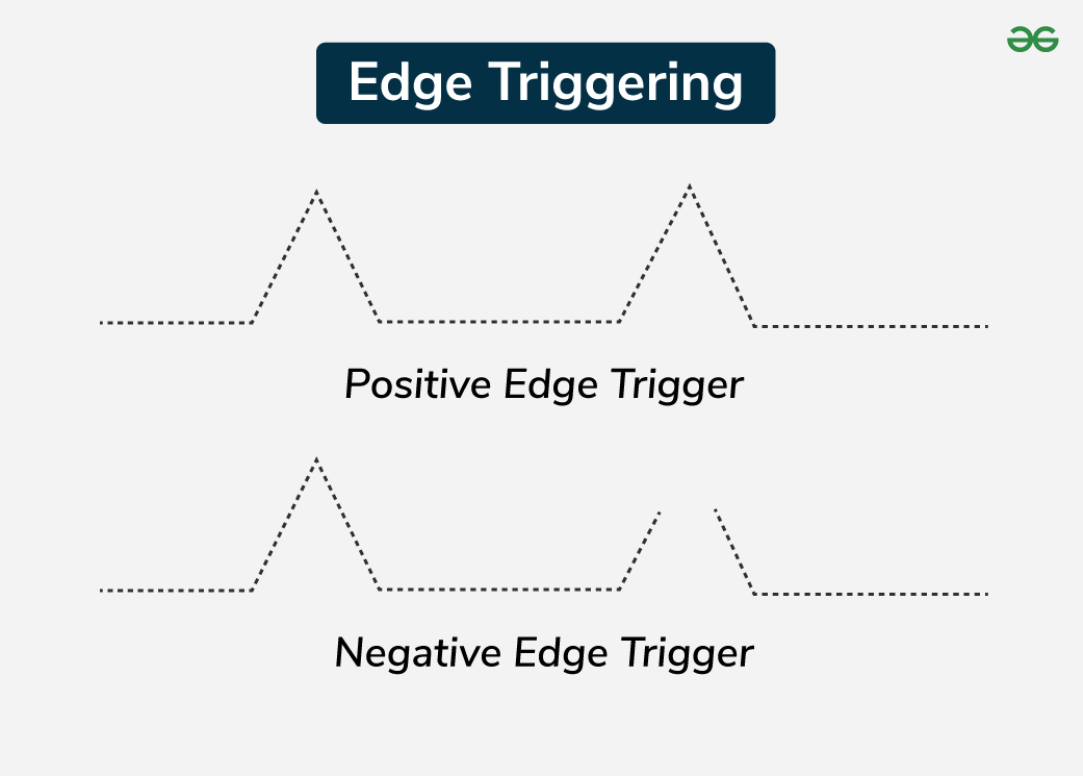
What is Edge Triggering?

It is used wherever it is required to identify the transition in the state of the input signal from low to high or from high to low. It is commonly applied to synchronous circuits, including flip-flops and counters. More often, edge triggering is advantageous when high accuracy of timing is required.

In edge triggering, the rapid change in the input signal that is sampled by the circuit's clock signal leads to a change in the signal. This edge, referred to as the trigger edge of the pulse, may be rising, that is, from a low state to high, or falling, that is, from high to low, depending on the circuit implementation. When the output signal crosses the trigger edge, the circuit changes the state of the output signal.

Advantages of Edge Triggering

- **Precise Timing:** This simply ensures that the signal transition is correct and well timed due to timing needs such as flip-flops and counters.
- **Reliability:** Offers less chances of errors occurring due to noise or variance in the signal that is being received.



Edge Triggering

Disadvantages of Edge Triggering

- **Complex Design:** Demands stricter clock management, which leads to the augmentation of the resulting circuit complexity.
- **High Sensitivity:** May respond to unintended edges by noise, thus resulting in the possibility of an error in computation.

What is Level Triggering?

Level triggering, in contrast, is a kind of triggering that determines a signal at a certain level rather than the change of its state. It gives some output when it is required to know the value of the signal at some point in time and not just a change in state. Level triggering is commonly used in data acquisition and control systems because of the need for a constant level of input signal documentation.

Level triggering is continuously included that checks for the input signal, and the output is produced when the input signal is at the determined level and above. This level can be high or low depending on the design of this circuit type. The output signal depends on the triggered state until the input signal descends below the trigger level.

Advantages of Level Triggering

- **Continuous Monitoring:** Optimum for applications where there is a need for determination of a signal level that is constant, including control systems and data acquisition.
- **Simpler Circuit Design:** This is easier to implement as compared to the edge triggering circuits.

Disadvantages of Level Triggering

- **Less Precise:** It offers less precise timing when compared to the edge triggering technique.
- **Longer Trigger Duration:** As long as the signal remains on the trigger level, the output is in the triggered state, and that may not be required in some applications.

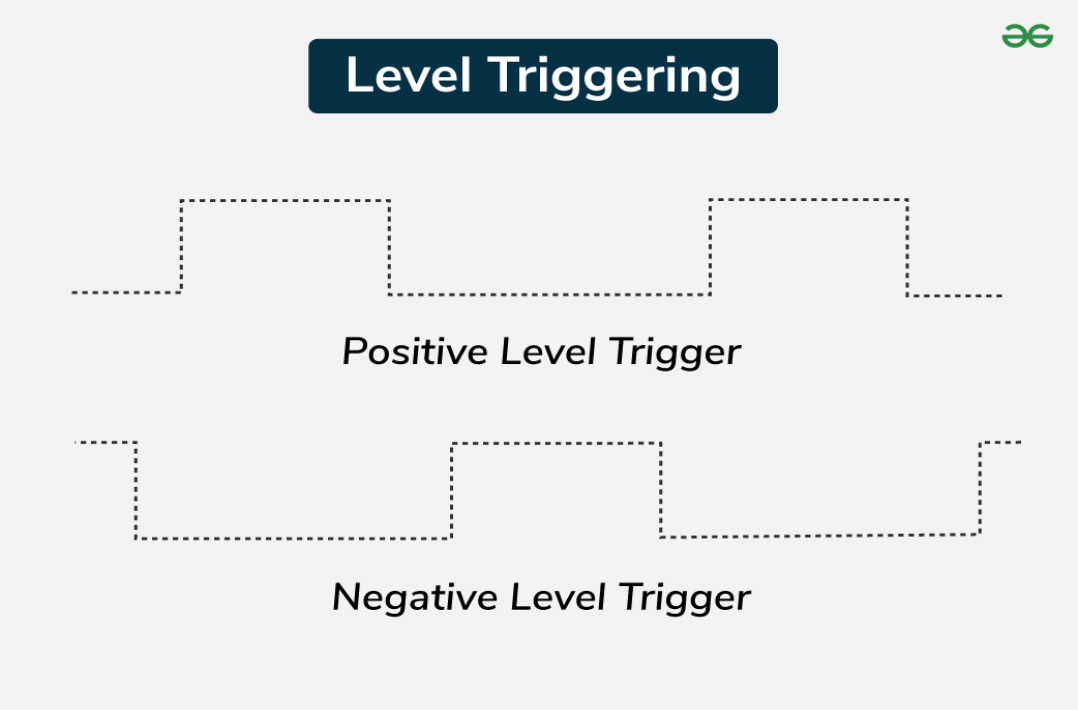


Fig: Level Triggering

Differences between Edge Triggering and Level Triggering

Edge Triggering	Level Triggering
Edge triggering is based on detecting a sharp edge in the input signal.	Level triggering is based on recognizing a specified signal level.
Edge triggering is frequently used in synchronous circuits, such as counters and flip-flops.	Level triggering is frequently employed in applications that call for continuous monitoring of an input signal, such as data acquisition and control system
Edge triggering is particularly helpful in applications that need precise timing.	Level triggering lacks the ability to regulate precise timing.
When edge triggering is used, the output signal is activated when the trigger edge is detected and changes to the opposing state. As long as the input signal is at or above the trigger level.	In level triggering, the output signal will remain in the triggered condition.

5.3: Flip-flops (SR, JK, T, D, Master-slave)

1. SR Flip Flop

It is a Flip Flop with two inputs, one is S and the other is R. **S** here stands for Set and **R** here stands for Reset. Set basically indicates set the flip flop which means output 1 and reset indicates resetting the flip flop which means output 0. Here, a clock pulse is supplied to operate this flip-flop, hence it is a clocked flip-flop.

What is Flip Flop?

Flip-Flop is a term that comes under digital electronics, and it is an electronic component that is used to store one single bit of information.

Diagrammatic Representation of Flip Flop

Since Flip Flop is a sequential circuit so its input is based upon two parameters, one is the **current input** and other is the **output from previous state**. It has two outputs, both are **complementing** of each other. It may be in one of two stable states, either 0 or 1.

Prerequisite: Introduction of Sequential Circuits

Construction of SR Flip Flop

SR Flip Flop Construction using **4 NAND Gates**

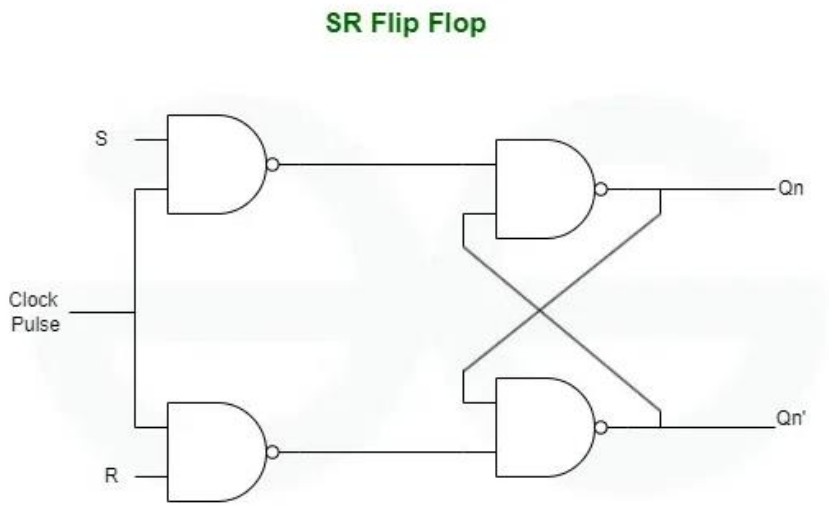


Fig: SR Flip Flop using NAND Gate

Basic Block Diagram of SR Flip Flop

The basic block diagram contains **S** and **R** inputs, and between them is clock pulse, **Q** and **Q'** is the complemented outputs.

SR Flip Flop basic Block diagram

Working of SR Flip Flop

- **Case 1** : Let's say, **S=0** and **R=0** , then output of both AND gates will be 0 and the value of Q and Q' will be same as their previous value, i.e, Hold state.
- **Case 2** : Let's say, **S=0 and R=1** , then output of both AND gates will be 1 and 0, correspondingly the value of Q will be 0 as one of input is 1 and it is a NOR gate so it will ultimately give 0, hence Q gets 0 value, similarly Q' will be 1.
- **Case 3:** Let's say, **S=1 and R=0,** then output of both AND gates will be 0 and 1, correspondingly the value of Q' will be 0 as one of input to NOR gate is 1, so output will be 0 ultimately and these 0 values will go as input to upper NOR gate, and hence Q will become 1.
- **Case 4:** Let's say, **S=1 and R=1,** then output of both AND gates will be 1 and 1 which is invalid, as the outputs should be complement of each other.

Truth Table of SR Flip Flop

Given Below is the Truth Table of SR Flip Flop

S	R	Qn+1	State
0	0	Qn	Hold
0	1	0	Reset
1	0	1	Set
1	1	X	Invalid

Here, **S** is the Set input, **R** is the reset input, **Qn+1** is the next state and **State** tells in which state it enters

Function Table of SR Flip Flop

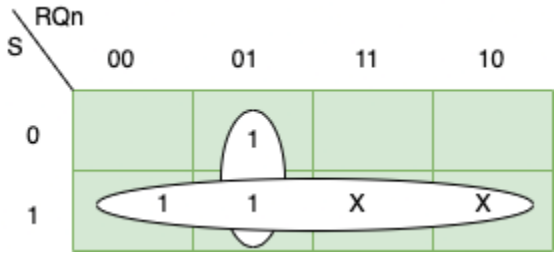
Given Below is the Function Table of SR Flip Flop

S	R	Qn	Qn+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	Invalid
1	1	1	Invalid

Here, **S** is the Set input, **R** is the reset input, **Qn** is the current state input and **Qn+1** is the next state outputs.

Characteristic Equation

- The characteristic equation tells us about what will be the next state of flip flop in terms of present state.
- In order to get the characteristic equation, K-Map is constructed which will be shown as below:



- If we solve the above K-Map then the characteristic equation will be **Qn+1 = S + QnR'**

Excitation Table

- Excitation Table basically tells about the excitation which is required by flip flop to go from current state to next state.

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

- Here, **Qn** is the current state, **Qn+1** is the next state outputs and **S, R** are the set and reset inputs respectively.

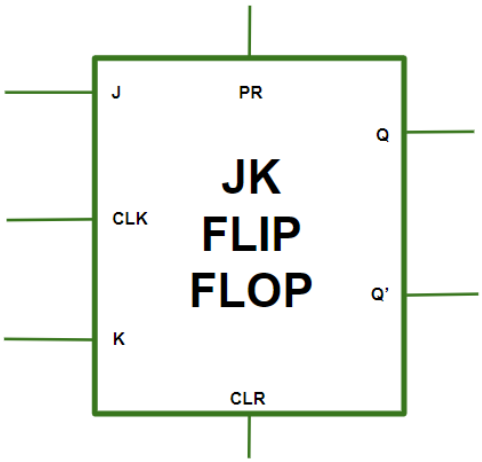
Applications of SR Flip Flop

There are numerous applications of SR Flip Flop in Digital System, which are listed below:

- **Register:** SR Flip Flop used to create register. Designer can create any size of register by combining SR Flip Flops.
- **Counters:** SR Flip Flops used in counters. Counters counts the number of events that occurs in a digital system.
- **Memory:** SR Flip Flops used to create memory which are used to store data, when the power is turned off.
- **Synchronous System:** SR Flip Flop are used in synchronous system which are used to synchronize the operation of different component.

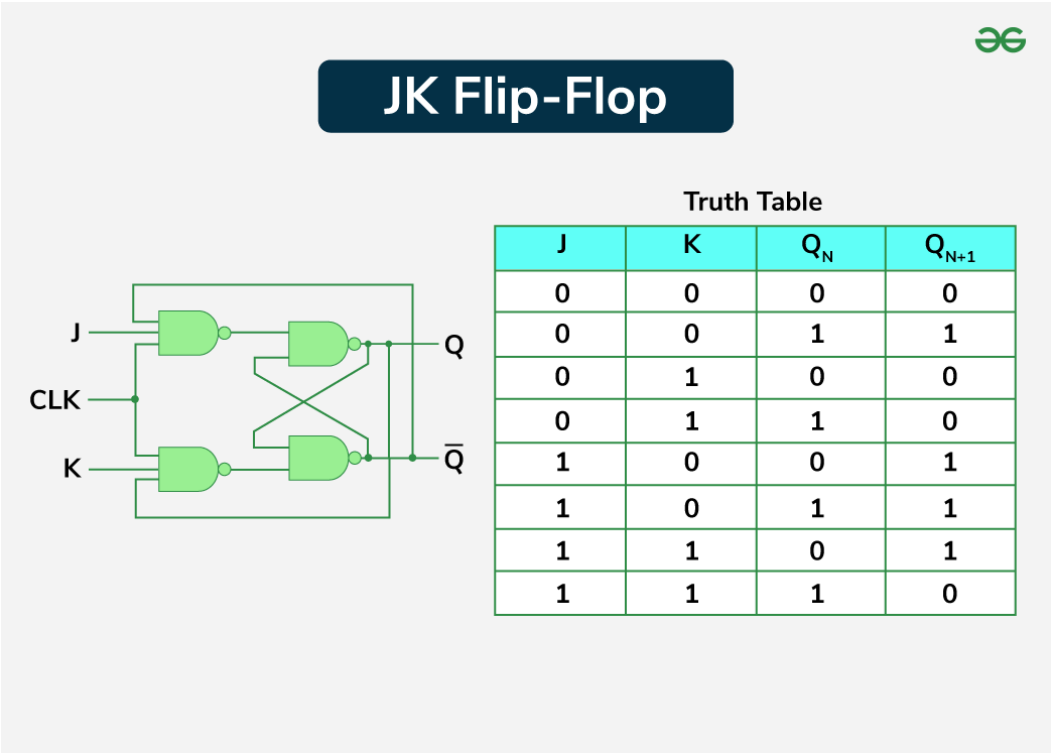
2. JK Flip-Flop

It is one kind of sequential logic circuit which stores binary information in bitwise manner. It consists of two inputs and two outputs. Inputs are Set(J) & Reset(K) and their corresponding outputs are Q and Q'. JK flipflop has two modes of operation which are synchronous mode and asynchronous mode. In synchronous mode, the state will be changed with the clock(clk) signal, and in asynchronous mode, the change of state is independent from its clock signal. Let's see its diagram structure.



The JK flip flop diagram above represents the basic structure which consists of Clock (CLK), Clear (CLR), and Preset (PR).

Below is the circuit diagram of JK Flip Flop. Two 3-input NAND gates are used in place of the original two 2-input AND gates. The outputs at Q and Q' are coupled to each gate's third input. Since the two inputs are now interlocked, the SR flip-flop's cross-coupling enables the previously invalid condition of (S = "1", R = "1") to be employed to perform the "toggle action".



Truth 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

Fig: JK Flip Flop

In a circuit "set", the bottom NAND gate interrupts the J input coming from the "0" position of Q'. In the "RESET" state, the top NAND gate interrupts the K input coming from the 0 positions of Q. We can use Q and Q' to control the input because they are always different. The flip flop is toggled according to the truth table when both inputs "J" and "K" are set to 1.

Truth Table of JK Flip Flop

Characteristic Table

A JK flip-flop is a kind of sequential logic circuit that keeps track of binary data. Its characteristic table shows how the output (Qn+1) changes using inputs (J & K) along with the last state (Qn).

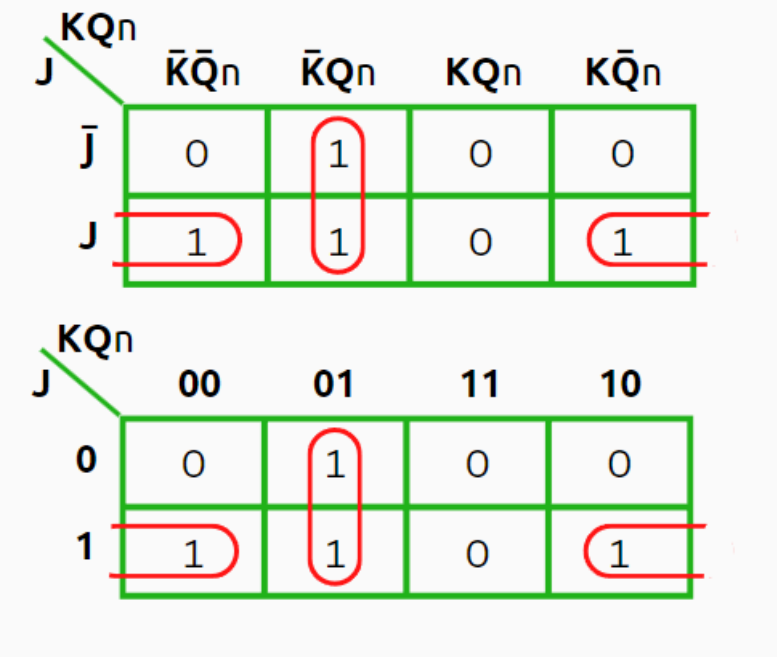
INPUT			OUTPUT
J	K	Qn	Q(n+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Fig: JK Flip Flop Characteristic Table

Let's break down the characteristic table. There are four states to understand:

1. J = 0, K = 0 (No Change): Here, the output Q(n+1) stays the same. This means the next state (Q(n+1)) is just like the current one (Qn).
2. J = 0, K = 1 (Reset State): In this case, the next state is reset to 0 (Q(n+1) = 0), no matter what the current state is (Qn).
3. J = 1, K = 0 (Set State): Now, the next state gets set to 1 (Q(n+1) = 1), again, no matter what's happening in the current state (Qn).
4. J = 1, K = 1 (Toggle State): In this state, the output Q(n+1) toggles. So, if the current state is set (Qn = 1), it will change to 0 (Q(n+1) = 0). If it's reset (Qn = 0), then it flips to 1 (Q(n+1) = 1).

Characteristic Equation: $\bar{K}Qn + J\bar{Q}n$



Group 1: $\bar{J}\bar{K}Qn + J\bar{K}Qn$

Group 2: $J\bar{K}\bar{Q}n + JK\bar{Q}n$

Equation: $\bar{J}\bar{K}Qn + J\bar{K}Qn + J\bar{K}\bar{Q}n + JK\bar{Q}n$

$= (\bar{J} + J)\bar{K}Qn + J\bar{Q}n(\bar{K}+K)$ *.....(**Distribution Law**)*

$= 1.\bar{K}Qn + 1.J\bar{Q}n$ *.....(**Compliment Law**, $\bar{J} + J = 1$ and $\bar{K}+K = 1$)*

$= \bar{K}Qn + J\bar{Q}n$ *.....(**Identity Law,** 1&A = A)*

Excitation Table

The excitation table shows us which input combinations we should use for a JK flip-flop to get the output we want.

- X - Don't Care
- Qn - Current State
- Q(n+1) - Next State
- J and K - Two input values

Here, X for Don't Care. This means it can be either 0 or 1, & it won't change how the flip-flop works.

INPUT		OUTPUT	
Qn	Q(n+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

This table guides us on what input values to use so we can move from the current (Qn) to the next state (Q(n+1)).

Let's look at the transitions

1. From **0 to 0**: When the current state is 0 (Qn = 0) and you want the next one to stay 0 (Q(n+1) = 0), then set J = 0 & K can be either 0 or 1, which is shown as X (Don't Care).
2. From **0 to 1**: If the current state is 0 (Qn = 0) but you want the next state to flip to 1 (Q(n+1) = 1), then J should be set to 1 & K = X (doesn't matter for K).
3. From **1 to 0**: When you start with a current state of 1 (Qn = 1) & wish for it to switch to 0 (Q(n+1) = 0), then J should be X and K must be set at 1.
4. From **1 to 1:** If you're at a current state of 1 (Qn = 1) & want it to stay at 1 for the next state (Q(n+1) = 1), then just set J = X and K should be set at 0.

This table is used a lot in circuit design. It helps show the inputs needed to get those desired output transitions.

Applications of JK Flip-Flop

- **Counters: Counters** are very essential components for the application of frequency dividers and event sequencers where there is a need of storing and propagating the count value. We can design binary synchronous and asynchronous counters using JK-flipflop.
- **Shift Registers:** For data storage and manipulation, serial-to-parallel or parallel-to-serial data conversion the shift registers are widely used. Registers can store and shift the binary data in a sequential manner. We can design it by JK-flipflops.

- **Memory Units:** JK-flipflop itself act as a memory unit to store binary information. By making a sequential chain of JK-flipflops we can use it even as RAM.

Advantages of JK Flip-Flop

- **Versatility:** As discussed above, JK-flipflops can be used as a basic memory element or a primary building block of further complex memory design. It is very much adaptive as it can be operated in both synchronous and asynchronous modes.
- **Toggle Functionality:** The application which are required to get output as its complement of input that also can be developed by JK-flipflops as when J=K=1 it triggers toggle state which gives output which is complement with its each clock pulse.
- **Error Detection and Correction:** We can use a complex circuit built by JK-flipflops which can detect and correct information during data-transmission.

Disadvantages of JK Flip-Flop

- **Complexity:** Compared to other types of flipflops (D, T, SR), JK flipflop requires additional logic gates to implement which consumes extra memory resources and increases complexity to operate.
- **Propagation Delay:** This is the major problem present in JK-FF. Propagation delay results a timing delay in certain application which are time-flow sensitive.
- **Race Problem:** This issue arises when the clock input's timing pulse isn't given enough time to turn "Off" before the output Q's state is altered.

3. D Flip Flop

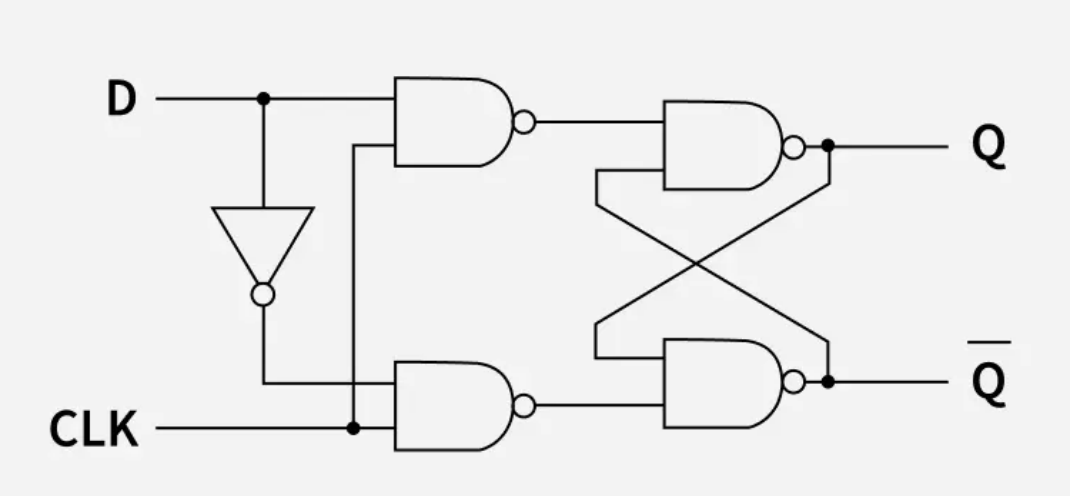
D flip flop is an electronic device that is known as "delay flip flop" or "data flip flop" which is used to store single bit of data.

- D flip flops can be either synchronous or asynchronous.
- The clock signal is required for the synchronous version of D flip flops but not for the asynchronous one.
- The D flip flop has two inputs, data and clock input which controls the flip flop.

- When clock input is high, the data is transferred to the output of the flip flop.
- When the clock input is low, the output of the flip flop is held in its previous state.

Block Diagram of D Flip Flop

A D flip-flop is created by modifying an SR flip-flop. The S input is connected to the D input, and the R input is connected to the inverted D input. As a result, a D flip-flop functions similarly to an SR flip-flop, but with complementary inputs, preventing any possibility of an invalid intermediate state. One major issue with the SR flip-flop is the race around condition, which is eliminated in the D flip-flop due to the inverted inputs. The circuit diagram of the D flip-flop is shown in the figure below:



Logic Circuit of D Flip Flop

Working of D Flip Flop

D flip flop consist of a single input D and two outputs (Q and Q'). The basic working of D Flip Flop is as follows:

- When the clock signal is low, the flip flop holds its current state and ignores the D input.
- When the clock signal is high, the flip flop samples and stores D input.
- The value that was previously fed into the D input is reflected at the flip flop's Q output.
    - If D = 0 then Q will be 0.
    - If D = 1 then Q will be 1.
- The Q' output of the flip flop is complemented by the Q output.

- If Q = 0 then Q' will be 1.
- If Q = 1 then Q' will be 0.

D	CLK	$\bar{Q}$
0	1 (Raising Edge)	0
1	1 (Raising Edge)	1

Truth Table of D Flip Flop

Characteristic Table of D Flip Flop

The characteristic table of the D flip flop displays the behavior of the flip flop for each combination of input and current state. The characteristic table for a D flip flop is as follows.

D	Q(Current)	Q(n+1) (Next)
0	0	0
0	1	0
1	0	1
1	1	1

Fig: Characteristics table of D Flip Flop

- D is the input, and Q is current state, Q(n+1) is the next state outputs.
- Q(n+1) will always be zero when D is 0, irrespective of current state of flip flop.
- When the input of the flip flop is 1, next state of flip flop will always be 1, regardless of the current state of flip flop.

Characteristic Equation of D Flip Flop

$$Q(n+1)= D$$

The characteristics equation of D flip flop consists of a Boolean expression that explains the relationship between the input and output of the flip flop. The characteristic equation for a D flip flop is:

Q(n+1) =D(n)

- Q(n+1) represents the output of flip flop at the next clock cycle.
- D(n) is the input to the flip flop at the current clock cycle, and n represents the current clock cycle.
- This characteristic equation of D flip flop states "**that the output of the flip flop at the next clock cycle will be equal to the input at the current clock cycle**".

D Flip Flop Excitation Table

Her, Q(n) represents the current state of the flip flop, and D(n) represents the current input of the flip flop. Whereas Q(n+1) represents the next state of the flipflop.

- When the Q(n) is 0 and the D(n) is also 0, then the Q(n+1) becomes 0. This situation explains the condition of "hold" state.
- When the Q(n) is 0 but D(n) is 1, then the Q(n+1) becomes 1. This situation explains the condition of "set" state.
- When the Q(n) is 1 but D(n) is 0, then the Q(n+1) becomes 0. This situation explains the condition of "reset" state.
- When the Q(n) is 1 and the D(n) is also 1, then the Q(n+1) becomes 1. This situation explains the condition of "hold" state.

Q(n)	D(n)	Q(n+1) (Next)
0	0	0
0	1	1
1	0	0
1	1	1

Advantages of D Flip Flop

- D flip flop is very simple to design.
- The computation speed of D flip flop is very fast compared to other flip flops.
- D flip flop requires very few components to design which makes it simple to understand.

Note: D flip flops are glitch prone. When input varies fast, flip flop output may glitch. Digital circuit glitches are hard to identify and fix.

Application of D Flip Flop

D flip flop has numerous applications in digital system is described as follows:

- **Memory:** D flip flop is used to create memory circuit for holding the data.
- **Registers:** D flip flop is used to create register, which can hold data in digital system. By using the D flip flop the designer can built any size of register as per the requirement.
- **Counters:** D flip flops are used to create the counters which counts the number of events occurred in the digital system.
- **Synchronous System:** D flip flop is having in developing the synchronous system.

4. T Flip Flop

Flip-flop is a term that comes under digital electronics, and it is an electronic component that is used to store one single bit of information. Since Flip Flop is a sequential circuit its input is based upon two parameters, one is the current input and the other is the output from the previous state.

- It has two outputs, both are complements of each other.
- It may be in one of two stable states, either 0 or 1.

Fig:Basic diagram of Flip Flop

T flip flop is known as **Toggle Flip Flop** because it is able to toggle its output depending upon on the input. Toggle basically indicates that the bit will be flipped i.e., either from 1 to 0 or from 0 to 1. Here, a clock pulse is supplied to operate this flip flop, hence it is a clocked flip-flop.

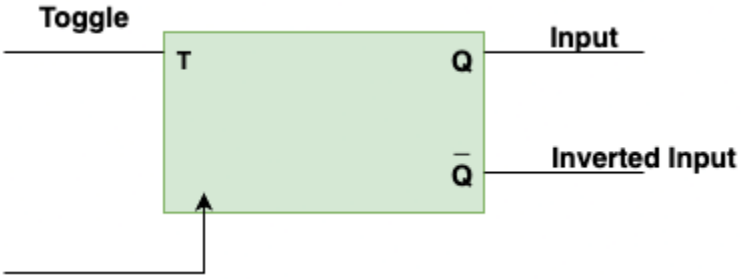


Fig: Block Diagram of T Flip Flop

Here block diagram contains Toggle and clock inputs, Q and Q' is the complemented inputs.

Truth Table of T Flip Flop

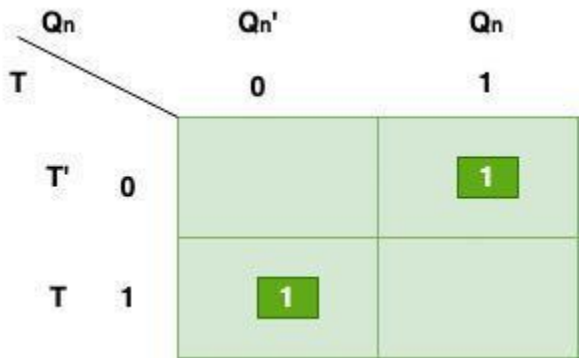
T	Q	Q(t+1)
0	0	0
1	0	1
0	1	1
1	1	0

Truth Table of T Flip Flop

- Here, T is the Toggle input, Q is present state input, Qt+1 is the next state output.
- From here we can see that, whenever Toggle (T) is 0, next state output (Qt+1) is same as current state input (Q).
- Whenever Toggle (T) is 1, next state output (Qt+1) will be complement of current state input (Q) which means it gets toggled.

Characteristic Equation

- The characteristic equation tells us about what will be the next state of flip flop in terms of present state.
- In order to get the characteristic equation, K-Map is constructed which will be shown as below:



K-Map for T Flip Flop

- If we solve the above K-Map then the characteristic equation will be

Q(n+1) = TQn' + T'Qn = T ⊕ Qn

Excitation Table

Excitation Table basically tells about the excitation which is required by flip flop to go from current state to next state.

Qt	Q(t+1)	T
0	0	0
1	0	1
0	1	1
1	1	0

Excitation Table for T Flip Flop

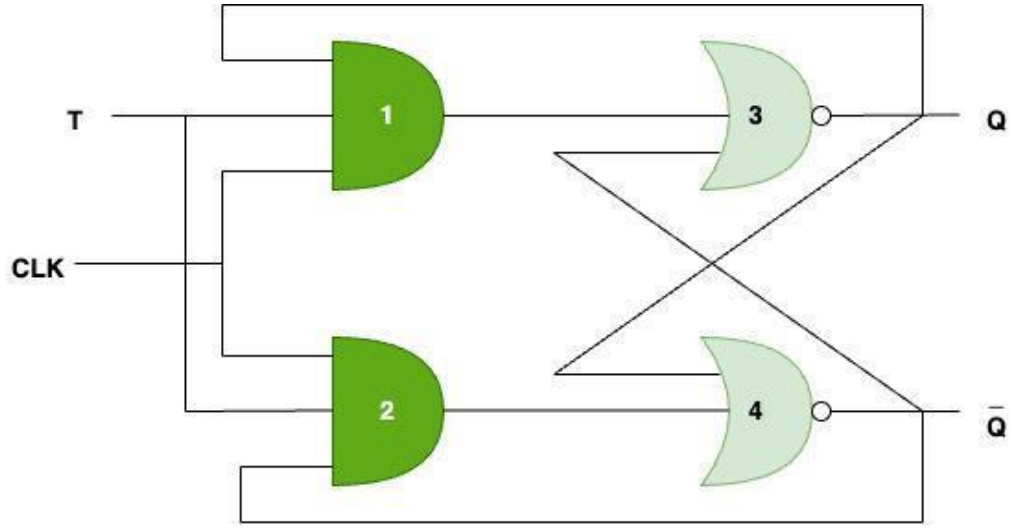
- Here, whenever T is 0, Qt+1 is same as input Q.
- And, whenever T is 1, Qt+1 is compliment of input Q.

Working of T Flip Flop

Case 1: Let's say, T = 0 and clock pulse is high i.e, 1, then output of both, AND gate 1, AND gate 2 will be 0, gate 3 output will be Q and similarly gate 4 output will be Q' so both the values of Q and Q' are same as their previous value, which means Hold state.

Case 2: Let's say, T=1, then output of both AND gate 1 will be (T * clock * Q), and since T and clock both are 1, then the output of AND gate 1 will be Q, and similarly output of AND gate 2 will be (T * clock * Q') i.e, Q'. Now, gate 3 output will be (Q'+Q)' and let's say Q' is zero, then gate 3 output will be (0+Q)' which

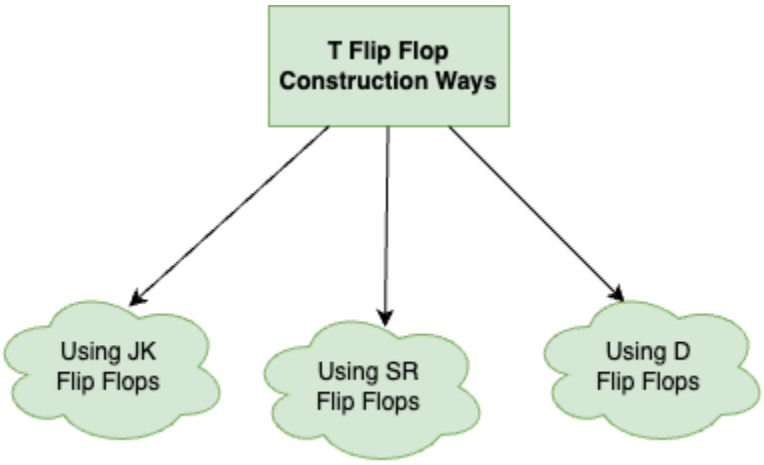
means Q' and similarly gate 4 output will be (Q+Q')' and since Q' is zero, so gate 4 output will be Q' which means 0 as Q' is zero. Hence in this case we can say that the output toggles, because T=1.



Construction of T Flip Flop

We can construct T flip flops in three ways namely:

- By using SR Flip Flops.
- By using D Flip Flops.
- By using JK Flip Flops



Ways to Construct T Flip Flop

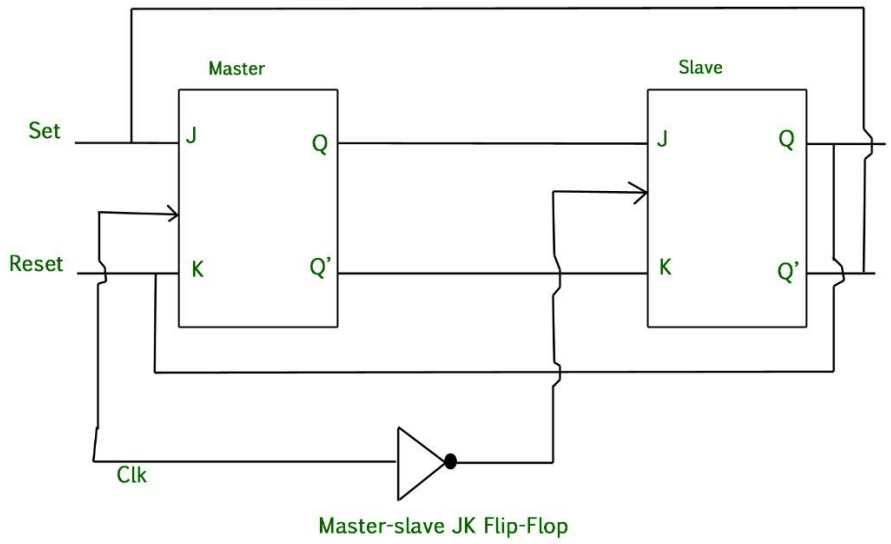
Applications of T Flip Flop

There are numerous applications of T Flip Flop in digital system, which are listed below:

- **Counters**: T Flip Flops are used in counters. Counters counts the number of events that occurs in a digital system.
- **Data Storage**: T Flip Flops used to create memory which are used to store data, when the power is turned off.
- **Synchronous logic circuits**: T flip-flops can be used to implement synchronous logic circuits, which are circuits that perform operations on binary data based on a clock signal. By synchronizing the logic circuit's operations to the clock signal using T flip-flops, the circuit's behavior can be made predictable and reliable.
- **Frequency division**: It is used to divide the frequency of a clock signal by 2. Flip-flop will toggle its output every time the clock signal transitions from high to low or low to high, hence dividing the clock frequency by 2.
- **Shift registers**: T flip-flops can be used in shift registers which are used to shift binary data in one direction.

Master-Slave JK Flip Flop

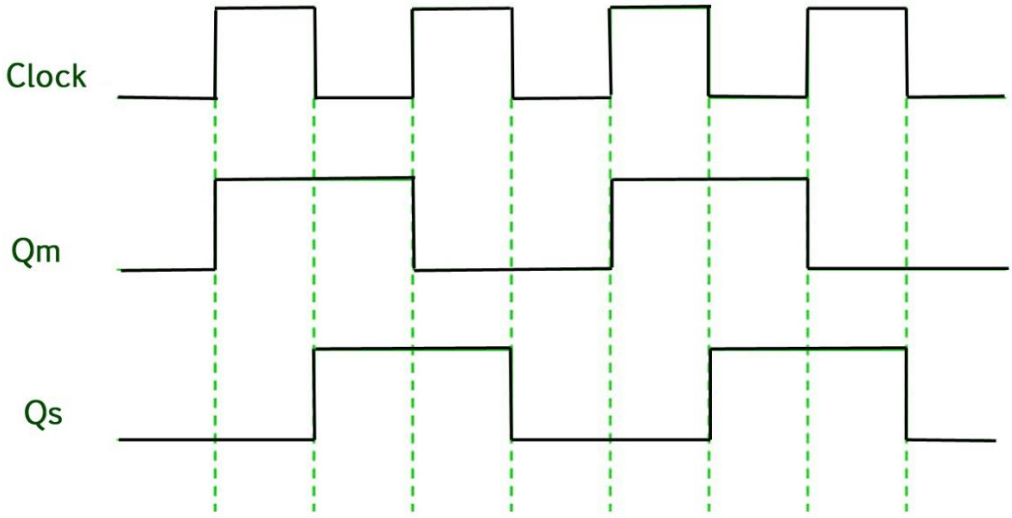
Prerequisite -Flip-flop types and their Conversion **Race Around Condition in JK Flip-flop -** For J-K flip-flop, if J=K=1, and if clk=1 for a long period of time, then Q output will toggle as long as CLK is high, which makes the output of the flip-flop unstable or uncertain. This problem is called race around condition in J-K flip-flop. This problem (Race Around Condition) can be avoided by ensuring that the clock input is at logic "1" only for a very short time. This introduced the concept of **Master Slave JK** flip flop. **Master Slave JK flip flop -** The Master-Slave Flip-Flop is basically a combination of two JK flip-flops connected together in a series configuration. Out of these, one acts as the **"master"** and the other as a **"slave"**. The output from the master flip flop is connected to the two inputs of the slave flip flop whose output is fed back to inputs of the master flip flop. In addition to these two flip-flops, the circuit also includes an **inverter**. The inverter is connected to clock pulse in such a way that the inverted clock pulse is given to the slave flip-flop. In other words, if CP=0 for a master flip-flop, then CP=1 for a slave flip-flop and if CP=1 for master flip flop then it becomes 0 for slave flip flop.



Master-slave JK Flip-Flop

Working of a master slave flip flop

1. When the clock pulse goes to 1, the slave is isolated; J and K inputs may affect the state of the system. The slave flip-flop is isolated until the CP goes to 0. When the CP goes back to 0, information is passed from the master flip-flop to the slave and output is obtained.
2. Firstly, the master flip flop is positive level triggered and the slave flip flop is negative level triggered, so the master responds before the slave.
3. If J=0 and K=1, the high Q' output of the master goes to the K input of the slave and the clock forces the slave to reset, thus the slave copies the master.
4. If J=1 and K=0, the high Q output of the master goes to the J input of the slave and the Negative transition of the clock sets the slave, copying the master.
5. If J=1 and K=1, it toggles on the positive transition of the clock and thus the slave toggles on the negative transition of the clock.
6. If J=0 and K=0, the flip flop is disabled and Q remains unchanged.



-

Fig: Timing diagram

1. When the Clock pulse is high the output of master is high and remains high till the clock is low because the state is stored.
2. Now the output of master becomes low when the clock pulse becomes high again and remains low until the clock becomes high again.
3. Thus, toggling takes place for a clock cycle.
4. When the clock pulse is high, the master is operational but not the slave thus the output of the slave remains low till the clock remains high.
5. When the clock is low, the slave becomes operational and remains high until the clock again becomes low.
6. Toggling takes place during the whole process since the output is changing once in a cycle.

This makes the Master-Slave J-K flip flop a Synchronous device as it only passes data with the timing of the clock signal.

THE END