# Flip-Flops

Flip-flops are synchronous bistable devices, also known as bistable multivibrators. In this case, the term synchronous means that the output changes state only at a specified point (leading or trailing edge) on the triggering input called the clock (CLK), which is designated as a control input, C; that is, changes in the output occur in synchronisation with the clock.

Flip-flops are edge-triggered or edge-sensitive, whereas gated latches are level-sensitive.
A flip-flop is a bistable digital circuit that can store one bit of information—either logic 0 or 1.


*Figure 1 The S and R in SR flip-flop mean 'SET' and 'RESET' respectively.*

## Types Of Flip-flops

There are different types of flip-flops depending on how their inputs and clock pulses cause transition between two states. We will discuss different types of flip-flops such as: S-R, D, J-K, T and Master-Slave. Basically D, J- K, and T are three different modifications of the S-R flip-flop.

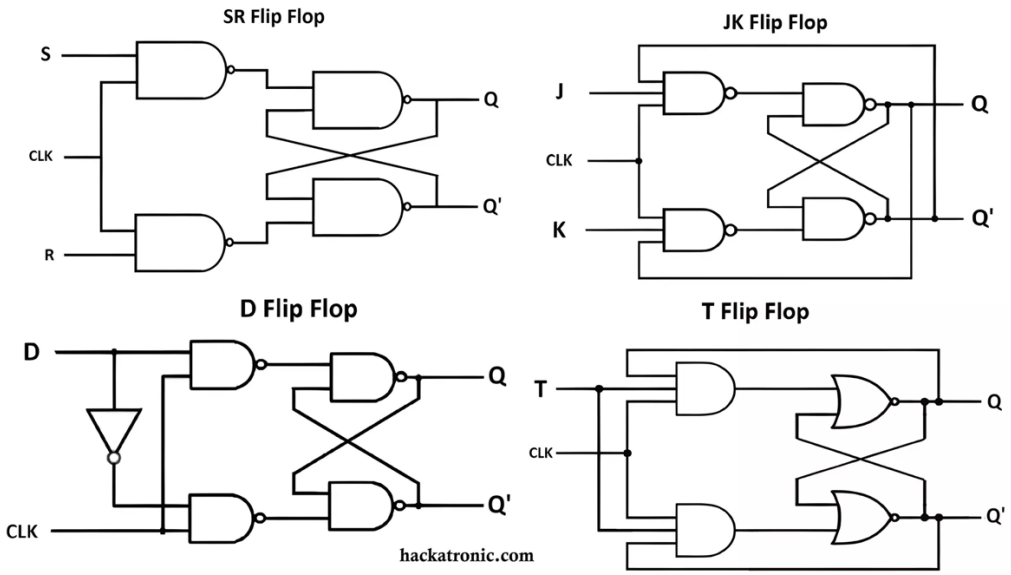



*Figure 2 Types of Flip-Flops*

## 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 $\bar{Q}$. 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.

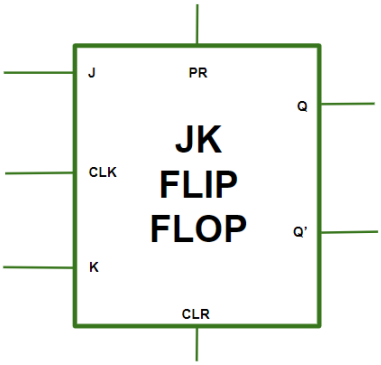



*Figure 3 Block Diagram of JK Flip Flop*

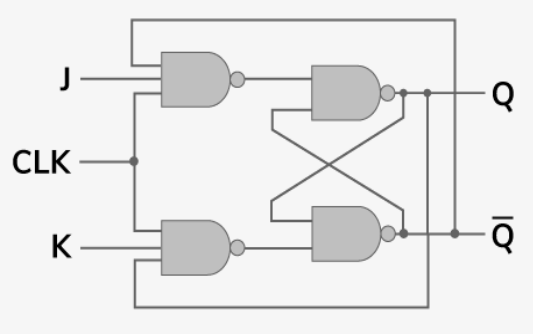
The JK flip flop diagram above represents the basic structure which consists of Clock (CLK), Clear (CLR), and Preset (PR).

A **J-K flip-flop** is a **clocked bistable multivibrator** used in sequential digital circuits to store one bit of information. It is an enhancement of the **SR flip-flop**, designed to eliminate the invalid state that occurs when both S and R are HIGH simultaneously.

| Input / Output | Symbol | Type | Function / Description |
|---|---|---|---|
| **Set Input** | **J** | Synchronous Input | Works like the "Set" input of an SR flip-flop. When J = 1 and K = 0 during the active clock edge, output Q is set to **1**. |
| **Reset Input** | **K** | Synchronous Input | Works like the "Reset" input of an SR flip-flop. When J = 0 and K = 1 during the active clock edge, output Q is reset to **0**. |
| **Clock** | **CLK** | Control Input | Synchronizes the flip-flop operation. The output changes only at the **triggering edge** of the clock signal (positive or negative edge). |
| **Preset (Direct Set)** | **PR** | Asynchronous Input | Forces **Q = 1** regardless of the clock or J, K inputs. Used for initialization. |
| **Clear (Direct Reset)** | **CLR** | Asynchronous Input | Forces **Q = 0** regardless of other inputs. Used for clearing the flip-flop. |
| **Main Output** | **Q** | Output | Represents the **current state** of the flip-flop (logic 1 or 0). |
| **Complementary Output** | **Q' (Q-bar)** | Output | Always the **inverse** of Q (i.e., Q' = NOT Q). |

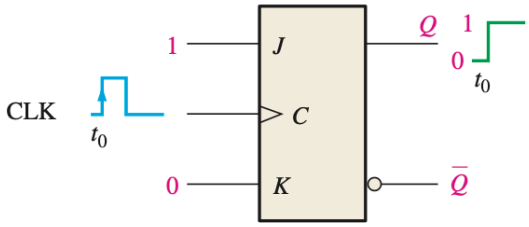
**Email (noorabdalkareem@uomus.edu.iq)**

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.

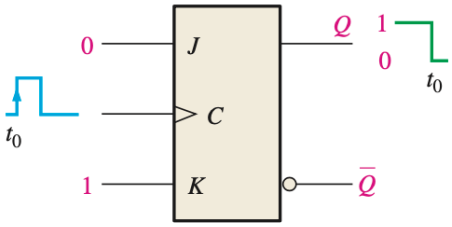



| J | K | Qn | Q(n+1) |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

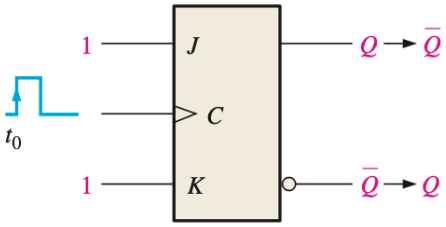
## Operation of a positive edge-triggered JK flip-flop



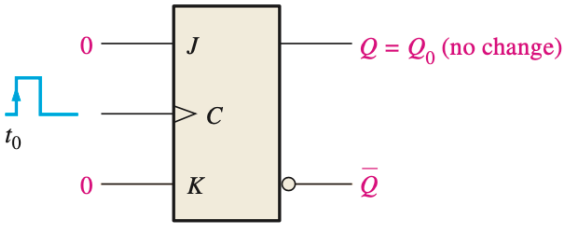

(a) $J = 1$, $K = 0$ flip-flop SETS on positive clock edge. (If already SET, it remains SET.)




(b) $J = 0$, $K = 1$ flip-flop RESETS on positive clock edge. (If already RESET, it remains RESET.)




(c) $J = 1$, $K = 1$ flip-flop changes state (toggle).




(d) $J = 0$, $K = 0$ flip-flop does not change. (If SET, it remains SET; if RESET, it remains RESET.)

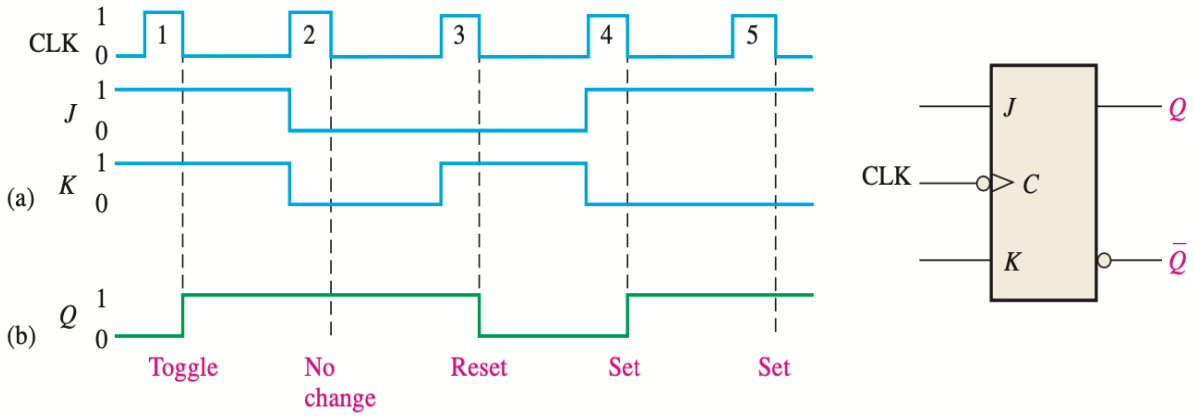
Email (noorabdalkareem@uomus.edu.iq)

# How JK Flip-Flop works

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)

The waveforms in Figure below are applied to the *J*, *K*, and clock inputs as indicated. Determine the *Q* output, assuming that the flip-flop is initially RESET.




Email (noorabdalkareem@uomus.edu.iq)

Truth table for a positive edge-triggered J-K flip-flop.

| Inputs | | | Outputs | | |
|---|---|---|---|---|---|
| $J$ | $K$ | CLK | $Q$ | $\overline{Q}$ | Comments |
| 0 | 0 | ↑ | $Q_0$ | $\overline{Q}_0$ | No change |
| 0 | 1 | ↑ | 0 | 1 | RESET |
| 1 | 0 | ↑ | 1 | 0 | SET |
| 1 | 1 | ↑ | $\overline{Q}_0$ | $Q_0$ | Toggle |

↑ = clock transition LOW to HIGH
$Q_0$ = output level prior to clock transition

## Applications of JK Flip-Flop

1. Counters: it's 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.
2. 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.
3. 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

1. Versatility: 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.
2. 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 it's each clock pulse.
3. Error Detection and Correction: We can use a complex circuit built by JK-flipflops which can detect and correct information during data-transmission.

Email (noorabdalkareem@uomus.edu.iq)

## Disadvantages of JK Flip-Flop

1. 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.
2. 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.
3. 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.

Email (noorabdalkareem@uomus.edu.iq)

# T 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 , 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.

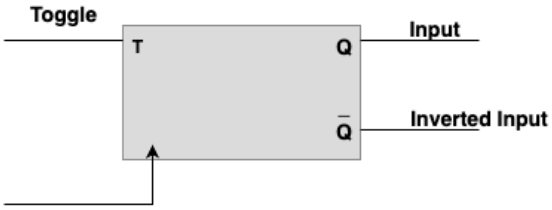



*Figure 4 Block Diagram of T Flip Flop*

# Working of T Flip Flop

**Case 1: T = 0**

- When **T = 0**, the flip-flop **does not change** its output.
- Regardless of how many clock pulses occur, the output **Q remains the same**.
- This is the **"Hold"** condition.

**Case 2: T = 1**

- When **T = 1**, the flip-flop **toggles** its output **at each clock pulse**.
  - If Q = 0 → Q becomes 1.
  - If Q = 1 → Q becomes 0.
- This is why it's called a **"Toggle" flip-flop**.

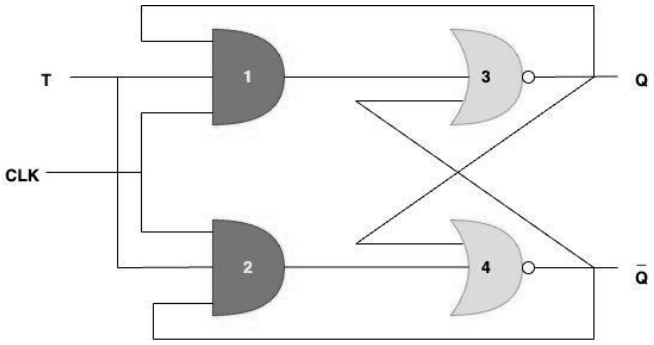



*Figure 5 T Flip Flop circuit*

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

*Figure 6 Truth Table of T Flip Flop*

**Email (noorabdalkareem@uomus.edu.iq)**

## 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 count 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.

## Homework:

1. Define a **JK flip-flop** and describe its operation for all possible input conditions.
2. Explain how a **T flip-flop** can be derived from a JK flip-flop.
3. Compare **JK** and **T flip-flops** in terms of their function, inputs, and typical applications.

**Email (noorabdalkareem@uomus.edu.iq)**