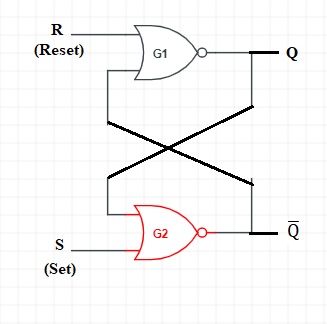
### Latch & Flip-Flop

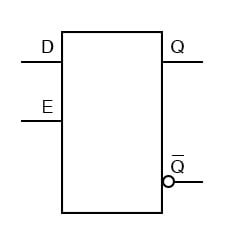
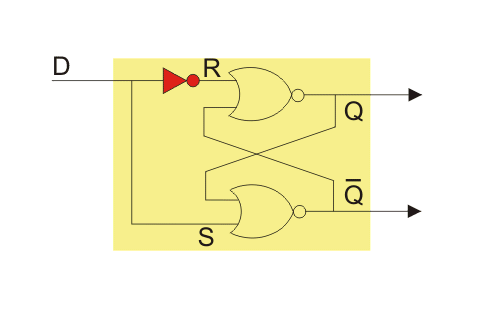
Latch is basic storage element in which we store 0 or 1. Latch as name suggest it holds 0 or 1. In the circuit “R” stands for reset and “S” stand for set. Q and are the output of the latch. When the circuit will be reset Q value will be equal to 0 and when the circuit will be set the Q value will be equal to 1.

**Truth Table of SR Latch:** **Circuit Diagram of SR Latch:**

|  |  |  |
| --- | --- | --- |
| **S** | **R** | **Q** |
| 0 | 0 | **HOLD** |
| 0 | 1 | **0 Set** |
| 1 | 0 | **1 Reset** |
| 1 | 1 | **NOT Allow** |

**Block Diagram of D Latch: Circuit Diagram of D Latch: Truth Table of D Latch:**

|  |  |
| --- | --- |
| **D** | **Q** |
| 0 | **0** |
| 1 | **1** |

****

* A D latch is like an S-R latch with only one input: the “D” input. Activating the D input sets the circuit, and de-activating the D input resets the circuit. Of course, this is only if the enable input (E) is activated as well. Otherwise, the output(s) will be latched, unresponsive to the state of the D input.
* D latches can be used as 1-bit memory circuits, storing either a “high” or a “low” state when disabled, and “reading” new data from the D input when enabled.

# JK Flip Flop:

The SR Flip Flop or Set-Reset flip flop has lots of advantages. But, it has the following switching problems:

* When Set 'S' and Reset 'R' inputs are set to 0, this condition is always avoided.
* When the Set or Reset input changes their state while the enable input is 1, the incorrect latching action occurs.

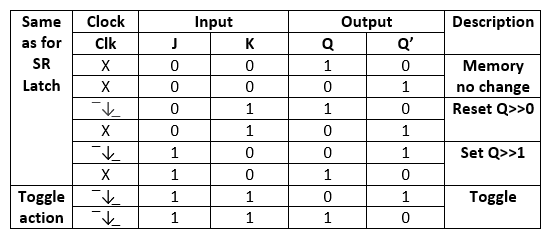
The JK Flip Flop removes these two drawbacks of [SR Flip Flop](https://www.javatpoint.com/sr-flip-flop-in-digital-electronics).

The [JK flip flop](https://www.javatpoint.com/jk-flip-flop-in-digital-electronics) is one of the most used flip flops in digital circuits. The JK flip flop is a universal flip flop having two inputs 'J' and 'K'. In SR flip flop, the 'S' and 'R' are the shortened abbreviated letters for Set and Reset, but J and K are not. The J and K are themselves autonomous letters which are chosen to distinguish the flip flop design from other types.

The JK flip flop work in the same way as the SR flip flop work. The JK flip flop has 'J' and 'K' flip flop instead of 'S' and 'R'. The only difference between JK flip flop and SR flip flop is that when both inputs of SR flip flop is set to 1, the circuit produces the invalid states as outputs, but in case of JK flip flop, there are no invalid states even if both 'J' and 'K' flip flops are set to 1.

### Block Diagram of JK Flip-Flop: Circuit Diagram of JK Flip-Flop:

### JK Flip FlopJK Flip Flop

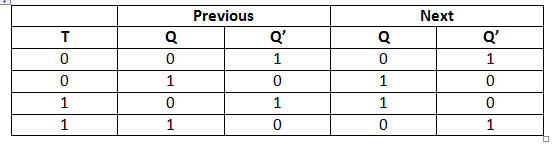
**Truth table of JK Flip-Flop:**

### T Flip-Flop Construction:

The "T Flip Flop" is designed by passing the AND gate's output as input to the [NOR gate](https://www.javatpoint.com/nor-gate-in-digital-electronics) of the "SR Flip Flop". The inputs of the "AND" gates, the present output state Q, and its complement Q' are sent back to each AND gate. The toggle input is passed to the [AND gates](https://www.javatpoint.com/and-gate-in-digital-electronics) as input. These gates are connected to the Clock (CLK) signal. In the "T Flip Flop", a pulse train of narrow triggers are passed as the toggle input, which changes the flip flop's output state. The circuit diagram of the "T Flip Flop" using "SR Flip Flop" is given below:

### Block Diagram of T Flip-Flop: Circuit Diagram of T Flip-Flop:

### T Flip FlopT Flip Flop

**Truth table of T Flip-Flop:**

### Operations of T-Flip Flop

The next sate of the T flip flop is similar to the current state when the T input is set to false or 0.

* If toggle input is set to 0 and the present state is also 0, the next state will be 0.
* If toggle input is set to 0 and the present state is 1, the next state will be 1.

The next state of the flip flop is opposite to the current state when the toggle input is set to 1.

* If toggle input is set to 1 and the present state is 0, the next state will be 1.
* If toggle input is set to 1 and the present state is 1, the next state will be 0.

The "T Flip Flop" is toggled when the set and reset inputs alternatively changed by the incoming trigger. The "T Flip Flop" requires two triggers to complete a full cycle of the output waveform. The frequency of the output produced by the "T Flip Flop" is half of the input frequency. The "T Flip Flop" works as the "Frequency Divider Circuit."

**Counter**

There are two **types of counters** based on the flip-flops that are connected in synchronous or not.

* Asynchronous counters
* Synchronous counters

**Asynchronous Counters**

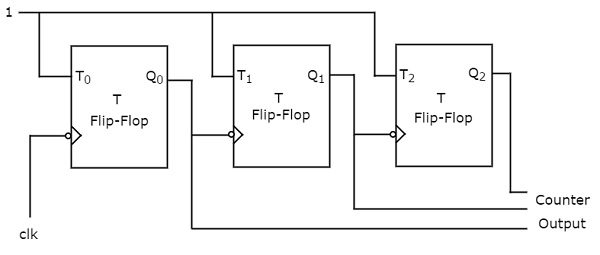
If the flip-flops do not receive the same clock signal, then that counter is called as **Asynchronous counter**. The output of system clock is applied as clock signal only to first flip-flop. The remaining flip-flops receive the clock signal from output of its previous stage flip-flop. Hence, the outputs of all flip-flops do not change *affect* at the same time.

Now, let us discuss the following two counters one by one.

* Asynchronous Binary up counter
* Asynchronous Binary down counter

1. **Asynchronous Binary up Counter:**

An ‘N’ bit Asynchronous binary up counter consists of ‘N’ T flip-flops. It counts from 0 to 2𝑁 − 1. The **block diagram** of 3-bit Asynchronous binary up counter is shown in the following figure.



The 3-bit Asynchronous binary up counter contains three T flip-flops and the T-input of all the flip-flops are connected to ‘1’. All these flip-flops are negative edge triggered but the outputs change asynchronously. The clock signal is directly applied to the first T flip-flop. So, the output of first T flip-flop **toggles** for every negative edge of clock signal.

The output of first T flip-flop is applied as clock signal for second T flip-flop. So, the output of second T flip-flop toggles for every negative edge of output of first T flip-flop. Similarly, the output of third T flip-flop toggles for every negative edge of output of second T flip-flop, since the output of second T flip-flop acts as the clock signal for third T flip-flop.

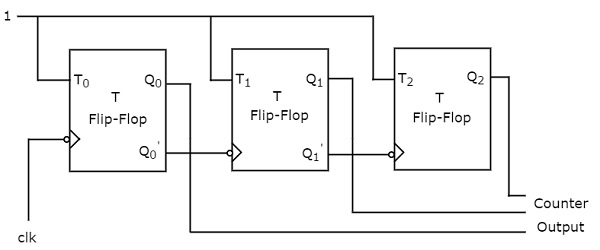
Assume the initial status of T flip-flops from rightmost to leftmost is *Q*2*Q*1*Q*0=000. Here, *Q*2& *Q*0 are MSB & LSB respectively. We can understand the **working** of 3-bit asynchronous binary counter from the following table.

Here *Q*0 toggled for every negative edge of clock signal. *Q*1 toggled for every *Q*0 that goes from 1 to 0, otherwise remained in the previous state. Similarly, *Q*2 toggled for every *Q*1 that goes from 1 to 0, otherwise remained in the previous state.

The initial status of the T flip-flops in the absence of clock signal is *Q*2*Q*1*Q*0=000. This is incremented by one for every negative edge of clock signal and reached to maximum value at 7th negative edge of clock signal. This pattern repeats when further negative edges of clock signal are applied.

1. **Asynchronous Binary Down Counter:**

An ‘N’ bit Asynchronous binary down counter consists of ‘N’ T flip-flops. It counts from 2𝑁 − 1 to 0. The **block diagram** of 3-bit Asynchronous binary down counter is shown in the following figure.



The block diagram of 3-bit Asynchronous binary down counter is similar to the block diagram of 3-bit Asynchronous binary up counter. But, the only difference is that instead of connecting the normal outputs of one stage flip-flop as clock signal for next stage flip-flop, connect the **complemented outputs** of one stage flip-flop as clock signal for next stage flip-flop. Complemented output goes from 1 to 0 is same as the normal output goes from 0 to 1.

Assume the initial status of T flip-flops from rightmost to leftmost is *Q*2*Q*1*Q*0=000. Here, *Q*2& *Q*0 are MSB & LSB respectively. We can understand the **working** of 3-bit asynchronous binary down counter from the following table.

Here *Q*0 toggled for every negative edge of clock signal. *Q*1 toggled for every *Q*0 that goes from 0 to 1, otherwise remained in the previous state. Similarly, *Q*2 toggled for every *Q*1 that goes from 0 to 1, otherwise remained in the previous state.

The initial status of the T flip-flops in the absence of clock signal is *Q*2*Q*1*Q*0=000. This is decremented by one for every negative edge of clock signal and reaches to the same value at 8th negative edge of clock signal. This pattern repeats when further negative edges of clock signal are applied.

Synchronous Counters

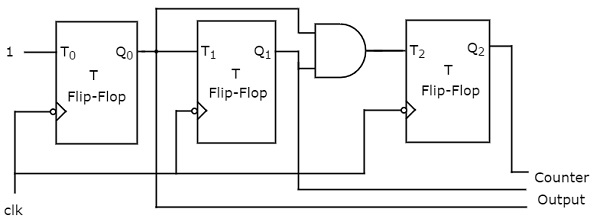
If all the flip-flops receive the same clock signal, then that counter is called as **Synchronous counter**. Hence, the outputs of all flip-flops change *affect* at the same time.

Now, let us discuss the following two counters one by one.

* Synchronous Binary up counter
* Synchronous Binary down counter

1. **Synchronous Binary Up Counter:**

An ‘N’ bit Synchronous binary up counter consists of ‘N’ T flip-flops. It counts from 0 to 2𝑁 − 1. The **block diagram** of 3-bit Synchronous binary up counter is shown in the following figure.

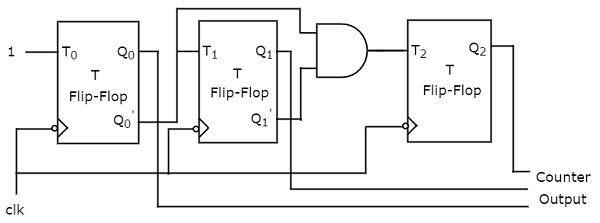


The 3-bit Synchronous binary up counter contains three T flip-flops & one 2-input AND gate. All these flip-flops are negative edge triggered and the outputs of flip-flops change *affect* synchronously. The T inputs of first, second and third flip-flops are 1, *Q*0 & *Q*1*Q*0 respectively.

The output of first T flip-flop **toggles** for every negative edge of clock signal. The output of second T flip-flop toggles for every negative edge of clock signal if *Q*0 is 1. The output of third T flip-flop toggles for every negative edge of clock signal if both *Q*0 & *Q*1 are 1.

1. **Synchronous Binary down Counter:**

An ‘N’ bit Synchronous binary down counter consists of ‘N’ T flip-flops. It counts from 2𝑁 − 1 to 0. The **block diagram** of 3-bit Synchronous binary down counter is shown in the following figure.



The 3-bit Synchronous binary down counter contains three T flip-flops & one 2-input AND gate. All these flip-flops are negative edge triggered and the outputs of flip-flops change *affect* synchronously. The T inputs of first, second and third flip-flops are 1, *Q*0′ &' *Q*1′*Q*0′respectively.

The output of first T flip-flop **toggles** for every negative edge of clock signal. The output of second T flip-flop toggles for every negative edge of clock signal if *Q*0′ is 1. The output of third T flip-flop toggles for every negative edge of clock signal if both *Q*1′ & *Q*0′ are 1.

**Shift Register**

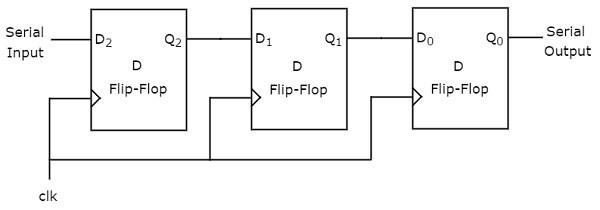
We know that one flip-flop can store one-bit of information. In order to store multiple bits of information, we require multiple flip-flops. The group of flip-flops, which are used to hold *store* the binary data, is known as **register**.

If the register is capable of shifting bits either towards right hand side or towards left hand side is known as **shift register**. An ‘N’ bit shift register contains ‘N’ flip-flops. Following are the four types of shift registers based on applying inputs and accessing of outputs.

* Serial In − Serial Out shift register
* Serial In − Parallel Out shift register
* Parallel In − Serial Out shift register
* Parallel In − Parallel Out shift register

**Serial In − Serial Out *SISO* Shift Register**

The shift register, which allows serial input and produces serial output is known as Serial In – Serial Out ***SISO*** shift register. The **block diagram** of 3-bit SISO shift register is shown in the following figure.

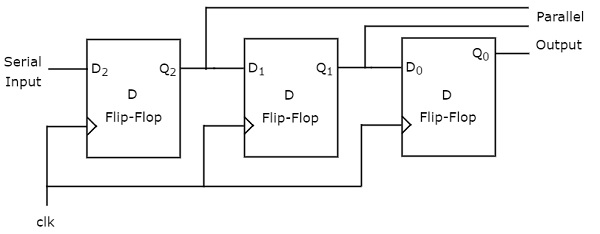


This block diagram consists of three D flip-flops, which are **cascaded**. That means, output of one D flip-flop is connected as the input of next D flip-flop. All these flip-flops are synchronous with each other since, the same clock signal is applied to each one.

In this shift register, we can send the bits serially from the input of left most D flip-flop. Hence, this input is also called as **serial input**. For every positive edge triggering of clock signal, the data shifts from one stage to the next. So, we can receive the bits serially from the output of right most D flip-flop. Hence, this output is also called as **serial output**.

**Serial In - Parallel Out *SIPO* Shift Register**

The shift register, which allows serial input and produces parallel output is known as Serial In – Parallel Out ***SIPO*** shift register. The **block diagram** of 3-bit SIPO shift register is shown in the following figure.



This circuit consists of three D flip-flops, which are cascaded. That means, output of one D flip-flop is connected as the input of next D flip-flop. All these flip-flops are synchronous with each other since, the same clock signal is applied to each one.

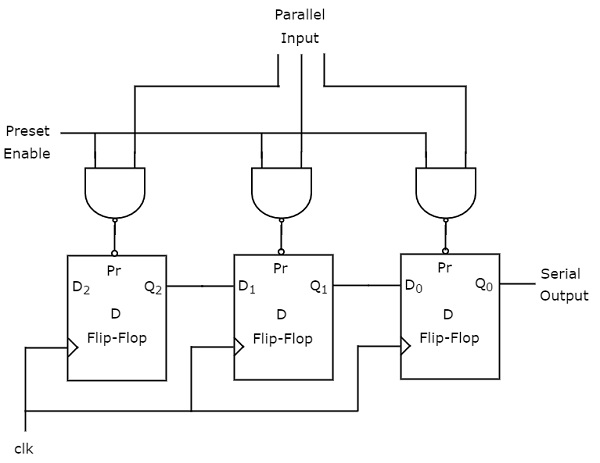
In this shift register, we can send the bits serially from the input of left most D flip-flop. Hence, this input is also called as **serial input**. For every positive edge triggering of clock signal, the data shifts from one stage to the next. In this case, we can access the outputs of each D flip-flop in parallel. So, we will get **parallel outputs** from this shift register.

**Parallel In − Serial Out *PISO* Shift Register**

The shift register, which allows parallel input and produces serial output is known as Parallel In − Serial Out ***PISO*** shift register. The **block diagram** of 3-bit PISO shift register is shown in the following figure.

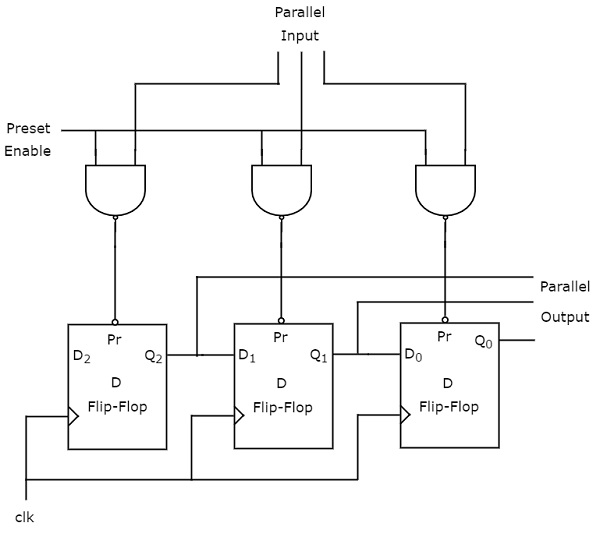
This circuit consists of three D flip-flops, which are cascaded. That means, output of one D flip-flop is connected as the input of next D flip-flop. All these flip-flops are synchronous with each other since, the same clock signal is applied to each one.

In this shift register, we can apply the **parallel inputs** to each D flip-flop by making Preset Enable to 1. For every positive edge triggering of clock signal, the data shifts from one stage to the next. So, we will get the **serial output** from the right most D flip-flop.

****

**Parallel In - Parallel Out *PIPO* Shift Register**

The shift register, which allows parallel input and produces parallel output is known as Parallel In − Parallel Out ***PIPO*** shift register. The **block diagram** of 3-bit PIPO shift register is shown in the following figure.



This circuit consists of three D flip-flops, which are cascaded. That means, output of one D flip-flop is connected as the input of next D flip-flop. All these flip-flops are synchronous with each other since, the same clock signal is applied to each one.

In this shift register, we can apply the **parallel inputs** to each D flip-flop by making Preset Enable to 1. We can apply the parallel inputs through preset or clear. These two are asynchronous inputs. That means, the flip-flops produce the corresponding outputs, based on the values of asynchronous inputs. In this case, the effect of outputs is independent of clock transition. So, we will get the **parallel outputs** from each D flip-flop.