Chapter 8

Introduction to Sequential Logic

The logic circuits that we have studied so far are consisted mainly of logic gates (AND, OR, NAND, NOR, INVERT) combinational logic. Those circuits are also consider a memoryless circuit. They act as an one time bit flow circuit that do not the capability to save the previous bit information. In this chapter, we will deal with data storage circuitry that will latch on to (remember) a digital state (1 or 0).

Latches vs Flip-Flops

Latches and flip-flops are both 1 – bit binary data storage devices. The main difference between a latch and a flip-flop is the triggering mechanism. Latches are transparent when enabled, whereas flip–flops are dependent on the transition of the clock signal is either positive edge or negative edge.

Latches are similar to flip-flops because they are bistable devices that can reside in either of two states using a feedback arrangement, in which the outputs are connected back to the opposite inputs. The major difference between latches and flip-flops is that a latch verifies the input constantly and alters the output based on the input change, whereas a flip-flop is a blend of the latch as well as a clock, it checks the input and modifies the time of output which is attuned by the CLK (clock). Some of the other differences between latches and flip – flops are listed in below table.

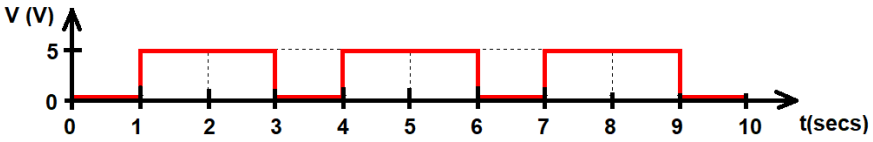
LATCH	FLIP-FLOP
Does not require clock pulse	Has a clock pulse
Asynchronous device	Synchronous device
When it is enabled, the output changes immediately if the input changes	A transition from LOW to HIGH, or vise-versa, will cause the flip-flop to either changes its output or retains it depending on the input signal.
Level triggering is involved	Edge triggering is involved
Faster operation as they do not have to wait for any clock signal.	Slower operation due to clock signal
Power requirement is less	More power requirement
Works based on the enable signal	Works based on the clock signal

A latch is a memory device used to store one bit of data. These are same like flip-flops, however, they are not synchronous devices. They do not work on edges of the clock as flip-flops do. (Difference between latches and flip-flops, 2015)

This chapter focuses mainly in the study of flip-flops instead of latches.

Digital Clock Pulse

Flip-flops are synchronous bi-stable devices that the output changes state only at a specified point on the triggering input called the clock pulse (CLK). Digital clocks change the state between 1, which is electronics equivalent to 5 V, and 0, 0 V.



The clock pulse is designated as a control input that changes in the output depending on the synchronization with the active state of the clock.

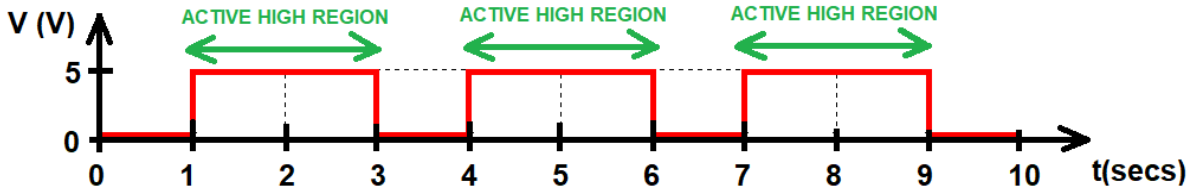
They are four type of digital clock pulse:

- Active High
- Active Low
- Edge Triggering Active Low
- Edge Triggering Active High

Flip-flops used edge triggering clock pulse because it is sensitive to its inputs only at the pulse edge rising or falling transition of the clock.

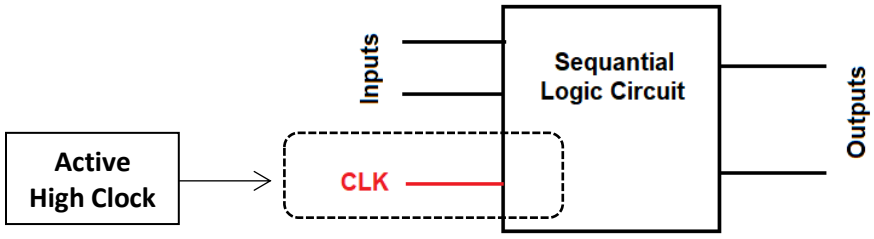
Active High Clock

Sequential logic circuits with active high clock only work during the duration of the HIGH width region, or 5 V region.



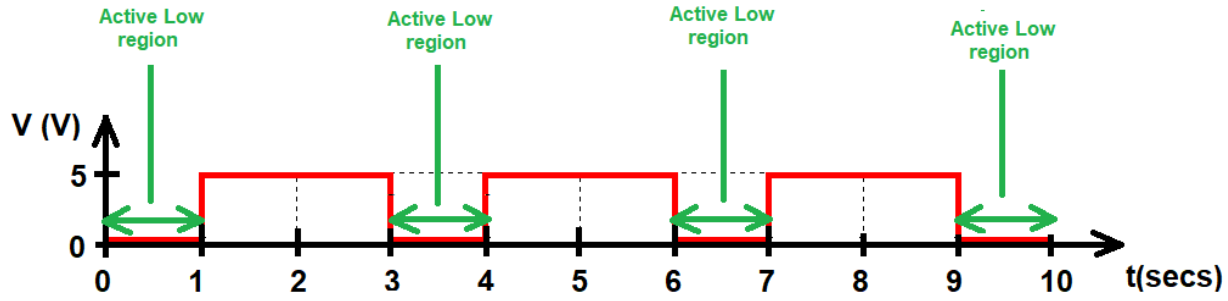
For the diagram above, the sequential logic circuit will work during the **ACTIVE HIGH REGION**, meaning at time between 1 and 3 seconds, 4 and 6 seconds, and 7 and 9 seconds. For the rest of the time, the logic circuit will not check the inputs' signal, instead, it will hold the previous signal until it the next active time.

The Sequential Logic circuit schematic below has an *Active Low Clock*



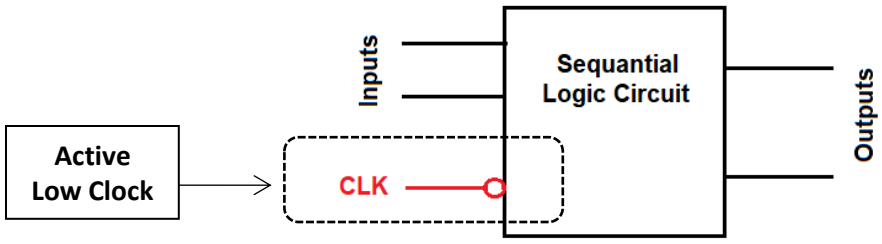
Active Low Clock

Sequential logic circuits with *Active Low Clock* only work during the duration of the LOW width region, or 0 V region.



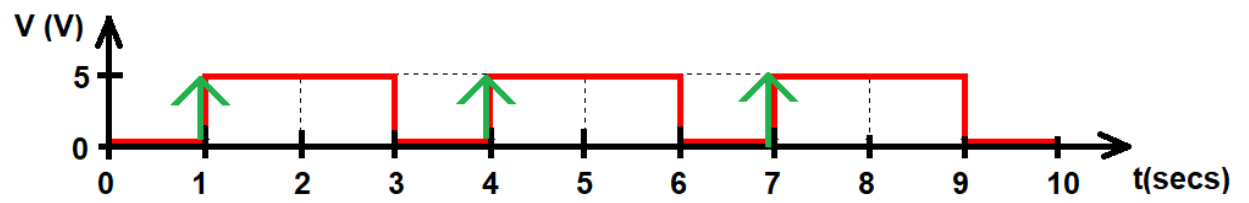
From the diagram above, the sequential logic circuit will work during the **ACTIVE LOW REGION**, meaning at time between 0 and 1 seconds, 3 and 4 seconds, 6 and 7, and 9 and 10 seconds. For the rest of the time, the logic circuit will not check the inputs' signal, instead, it will hold the previous signal until it the next active time.

The logic circuit schematic below has an *Active Low Clock*



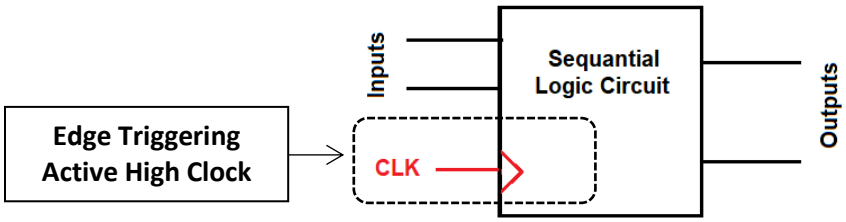
Edge Triggering Active High, rising edge

Sequential logic circuits with *Edge Triggering Active High* only works during the time the pulse changes from LOW to HIGH. We can say that during the time the pulse ***rises***.



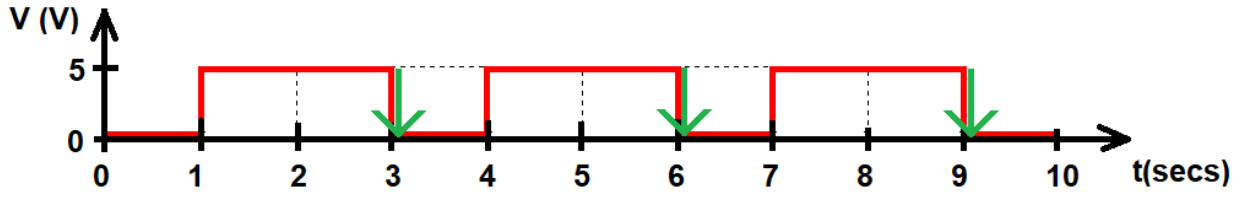
From the diagram above, the sequential logic circuit will work only during the time the pulse rises. In other words, when the pulse changes from logic 0 to logic 1 or, in terms of electronics, from 0 V to 5 V. In this case, at 1 second, 4 second, and 7 second.

The logic circuit schematic below has an *Edge Triggering Active High*



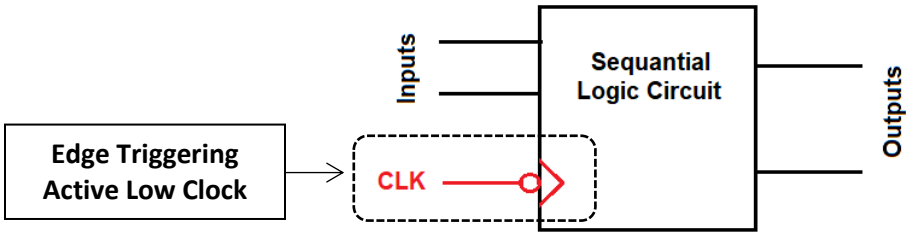
Edge Triggering Active Low, falling edge

Sequential logic circuits with *Edge Triggering Active Low* only works during the time the pulse changes from HIGH to LOW. We can say that during the time the pulse ***falls***.



From the diagram above, the sequential logic circuit will work only during the time the pulse falls. In other words, when the pulse changes from logic 1 to logic 0 or, in terms of electronics, from 5 V to 0 V. In this case, at 3 second, 6 second, and 9 second.

The logic circuit schematic below has an *Edge Triggering Active Low Clock*



This new type of digital circuitry is called **sequential logic,** or **flip flop circuits,** because it is controlled by and used for controlling other circuitry in a specific sequence dictated by a control clock or enable/disable control signals.

Flip Flops

Flip flop is formed using logic gates connecting in a way that they can save the previous bit information. They are basic building blocks in the memory of electronic devices and each flip flop can store one bit of data. The term flip–flop is used as they can switch between the states under the influence of a control signal (clock or enable). They can 'flip' to one state and 'flop' back to other state. They have two stable states and hence they are bistable multivibrators. The two stable states are High (logic 1) and Low (logic 0).

Computers and calculators use Flip-flop for their memory. A combination of number of flip flops will produce some amount of memory.

In conclusion:

- Flip-flops are a binary storage device because they can store binary data (0 or 1).
- Flip-flops are edge sensitive or edge triggered devices. They are sensitive to the transition rather than the duration or width of the clock signal.
- They are also known as signal change sensitive devices which mean that the change in the level of clock signal will bring change in output of the flip flop.
- A Flip–flop works depending on clock pulses.
- Flip flops are also used to control the digital circuit's functionality. They can change the operation of a digital circuit depending on the state.

Types of flip flops

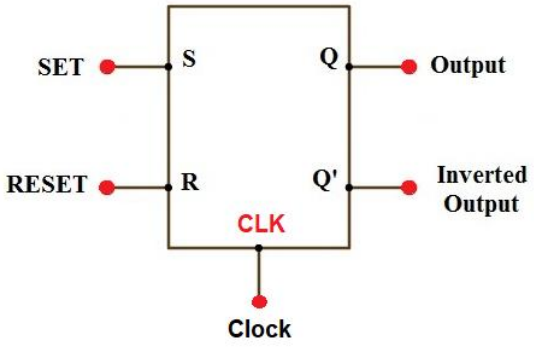
Based on their operations, flip flops are basically 4 types. They are

1. R-S flip flop
2. D flip flop
3. J-K flip flop
4. T flip flop

S-R Flip Flop

The S-R flip-flop is basic flip-flop among all the flip-flops. All the other flip flops are developed after S-R-flip-flop. S-R stands for SET and RESET. This can also be called R-S flip-flop. Difference is R-S is inverted S-R flip-flop.

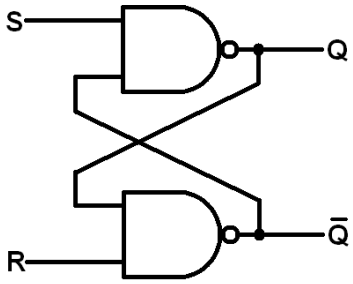
The S-R flip flop schematic is represented as shown below:



Any flip flop can be build using logic gates. NAND and NOR gates were used as they are universal gates.

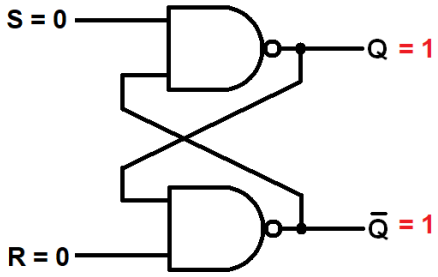
S-R Flip-flop made of NAND gates

Here is the SR flip-flop built using NAND gates.

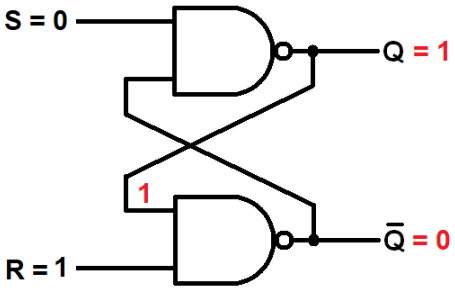


Let us analyze the bit flow within the S-R flip flop. For this we are going to analyze four cases:

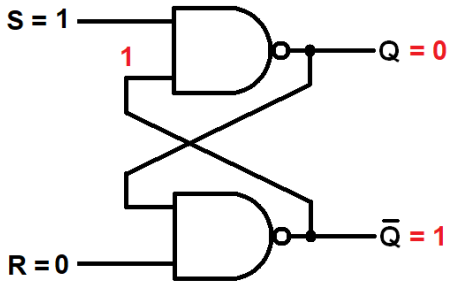
Case 1) S = 0, R = 0 ➔ **Invalid state** because if Q is equal to 1, $\bar{Q}$ must be equal to 0



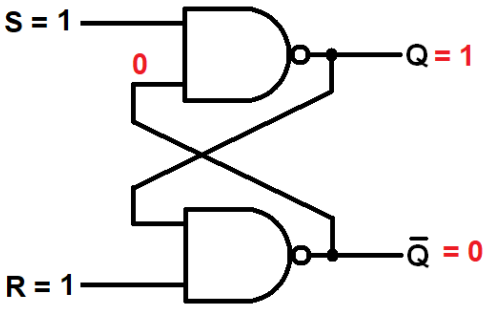
Case 2) S = 0, R = 1 ➔ **Q is set to 1. Set state**

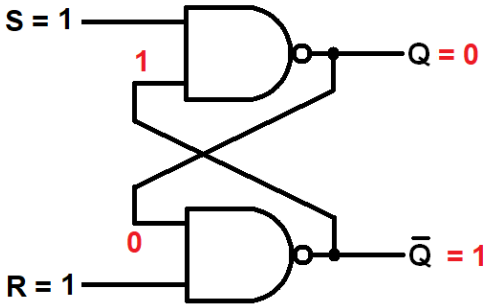


Case 3) S = 1, R = 0 ➔ **Q is set to 0. Reset state**



Case 4) S = 1, R = 1 ➔ **No change. Hold state**

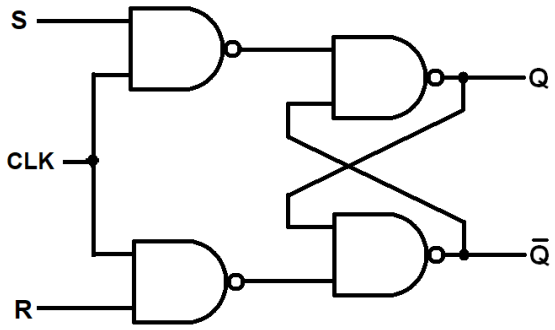




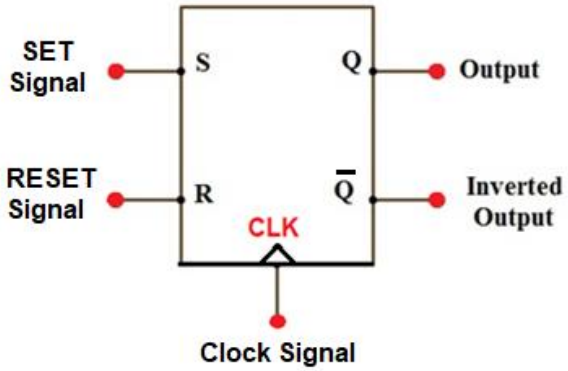
According to the four cases, the resulting truth table for a S-R flip-flop made of NAND gates is:

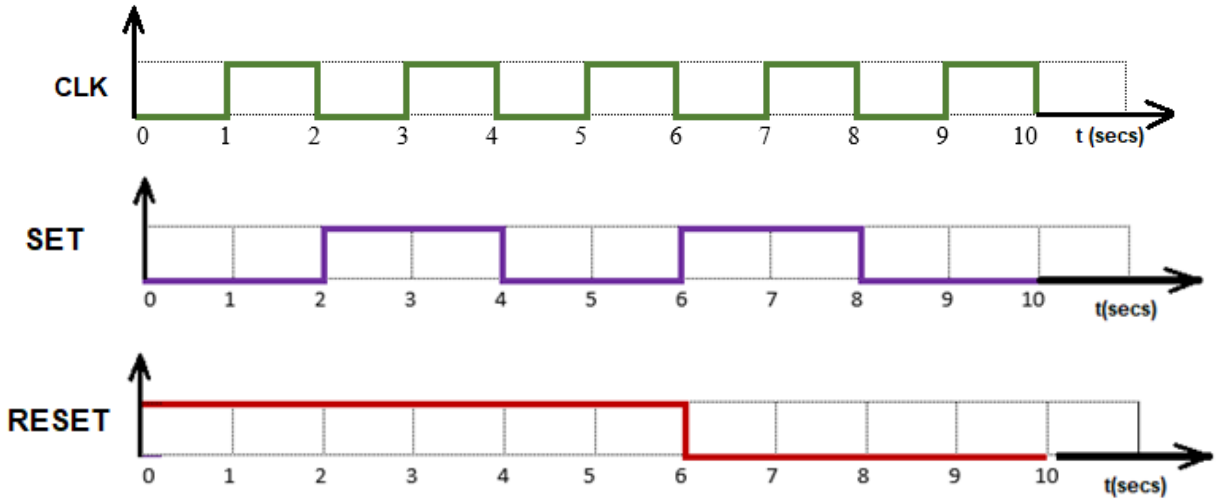
Table 8.1 S-R flip-flop made of NAND gates Truth Table				
Inputs		**Outputs**		**Condition**
S	**R**	**Q**	$\bar{Q}$	
0	0	1	1	Invalid State, not used
0	1	1	0	Sets Q to 1. Set state
1	0	0	1	Reset Q to 0. Reset state
1	1	Q	$\bar{Q}$	No change condition; resting state; holding condition

The circuit of a clocked SR flip–flop using NAND gates is shown below:

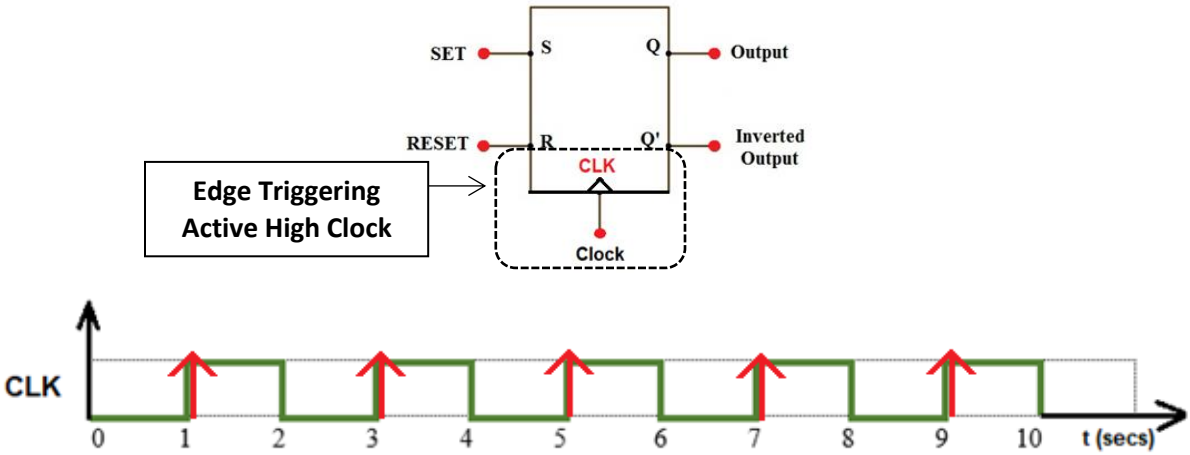


Example 8.1) For the following S-R flip-flop (NAND), sketch the input Q and $\bar{Q}$ assuming the initial Q = 0



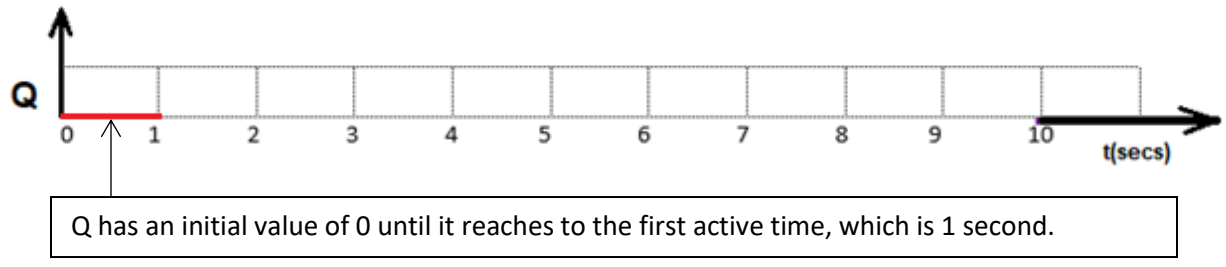


The first step is to identify the type of clock that the S-R flip-flop works with, in this case we have an *Edge Triggering Active High* clock. This means that the S-R FF will check input SET and RESET at time equal to 1, 3, 5, 7, and 9 second:

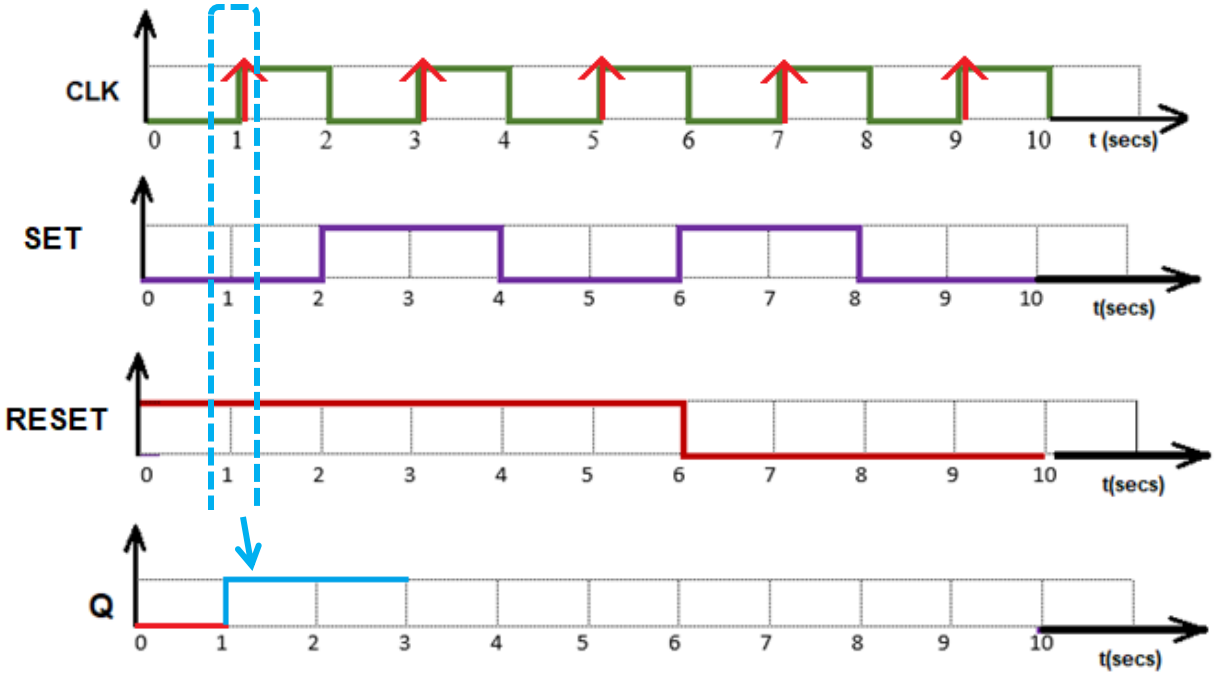


Now, using Table 8.1 S-R flip-flop made of NAND gates Truth Table, sketch the output Q depending on inputs S and R at the active time according to its clock. This means that we have to check the FF's input of SET and RESET at time equal 1, 3, 5, 7, and 9 second.

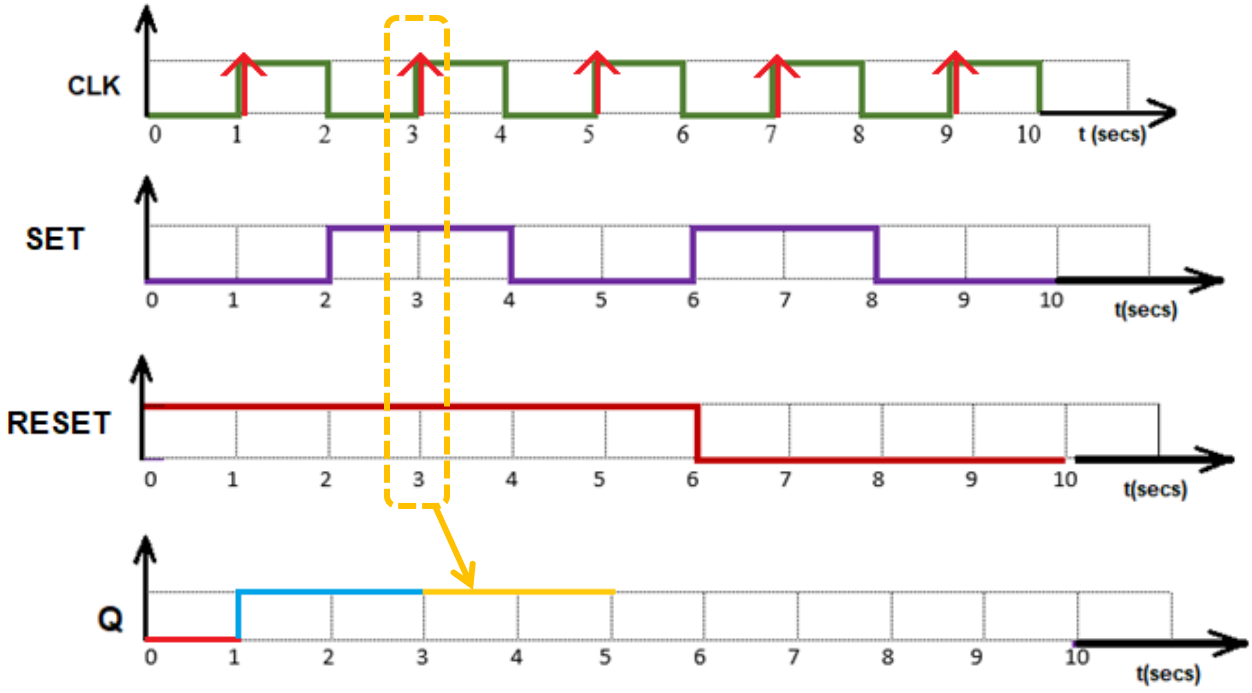
To sketch the output Q, first, remember that Q has an initial value of 0 until it reaches to the first active time, which is 1 second. Then from 0 second to 1 second, Q = 0.



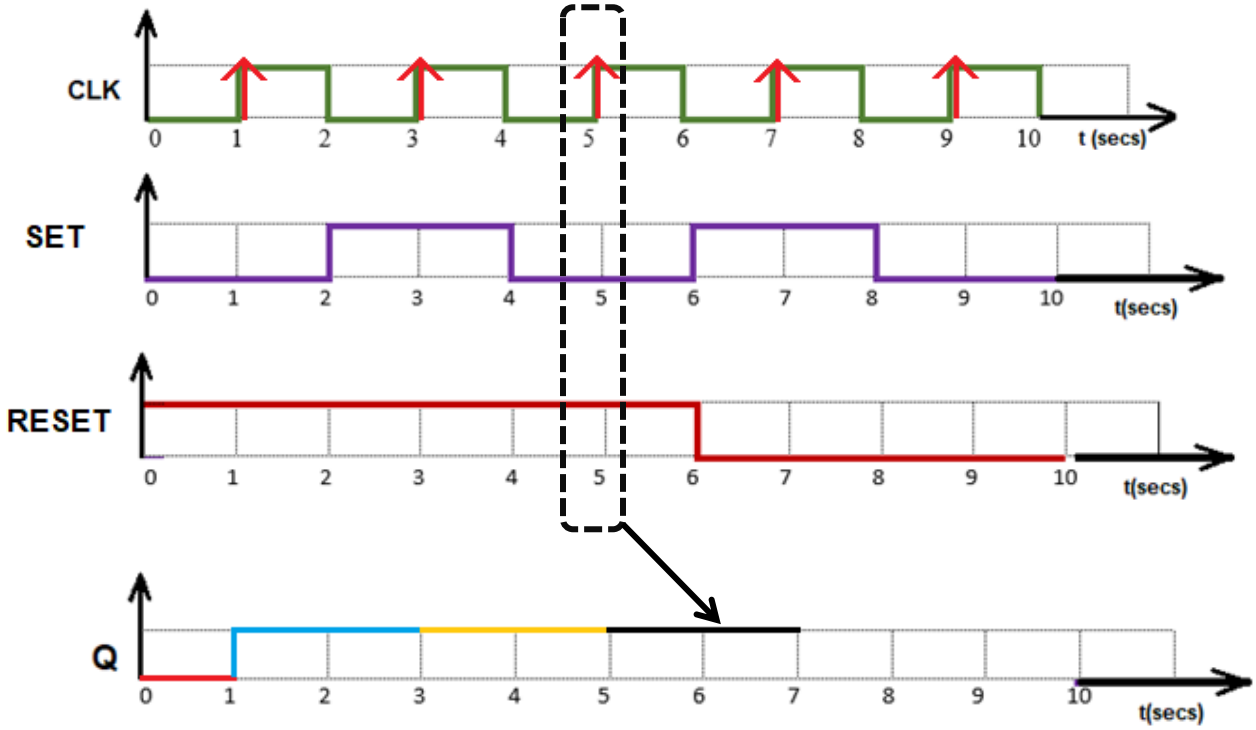
At time equal to 1 second, the flip-flop will be active and check the input SET and RESET signal at 1 second. At 1 second, SET is 0 and RESET is 1, therefore, if you check Table 8.1, Q should be 1. In this case, Q will be 1 and hold 1 until the next active time, which is 3 seconds.



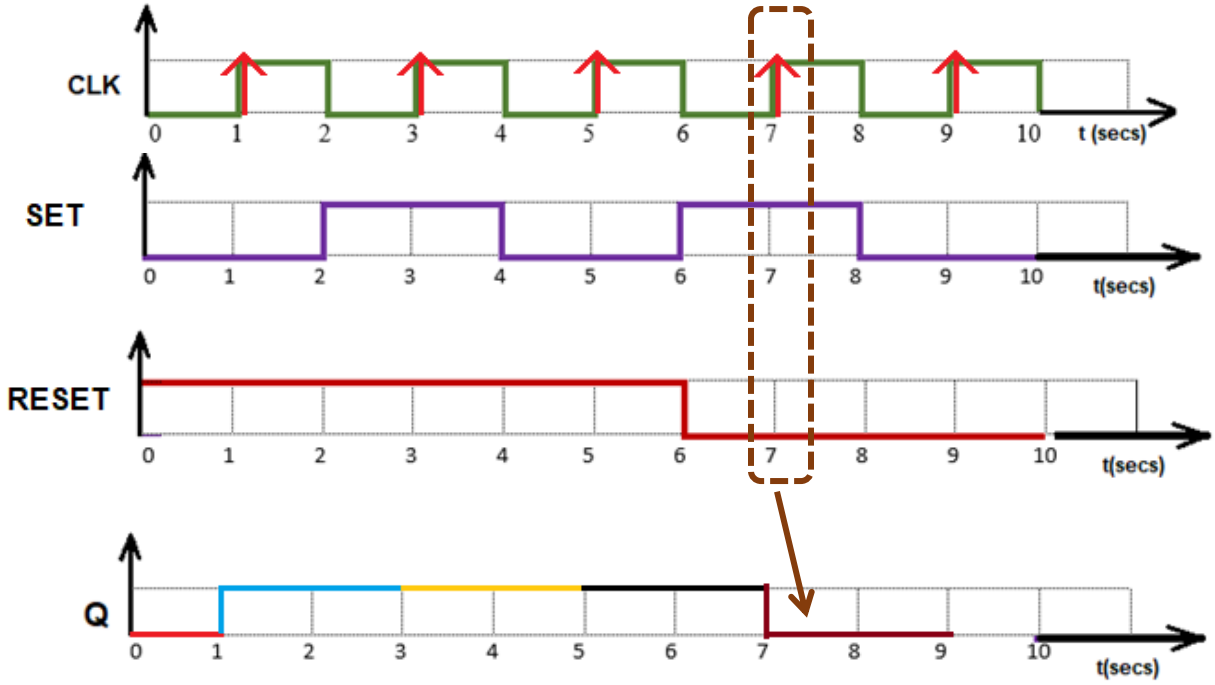
At time equal to 3 second, the flip-flop will be active again and check the input SET and RESET signal at 3 second. At 3 seconds, SET is 1 and RESET is 1, therefore, if you check Table 8.1, Q has a NO CHANGE state, meaning that Q keeps the previous value of 1. Q will be 1 and hold 1 until the next active time, which is 5 seconds.



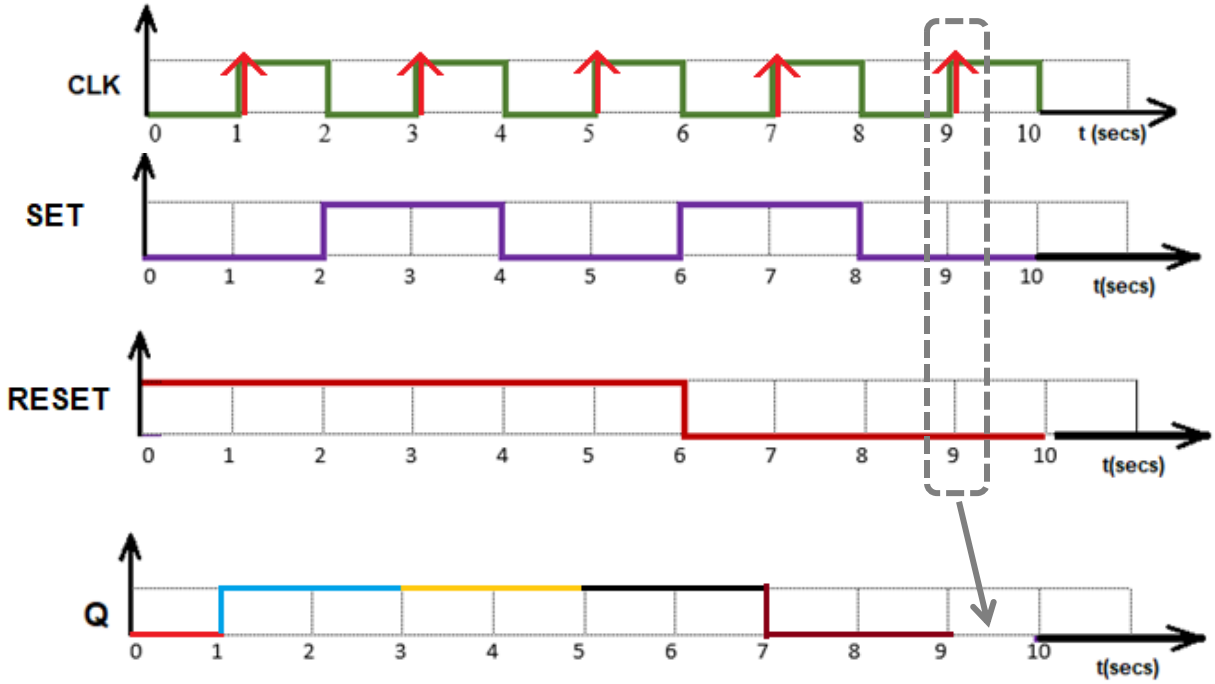
At time equal to 5 second, the flip-flop will be active again and check the input SET and RESET. At 5 seconds, SET is 0 and RESET is 1, therefore, if you check Table 8.1, Q should be 1. Q will be 1 and hold 1 until 7 seconds.



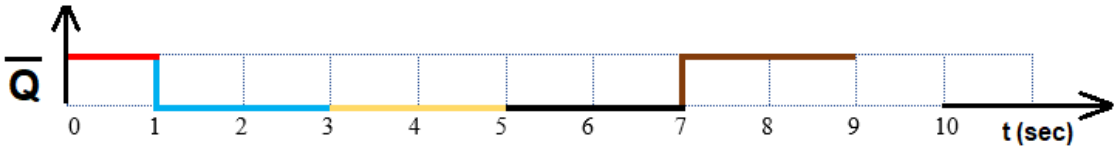
At time equal to 7 second, the flip-flop will be active again and check the input SET and RESET. At this time, SET is 1 and RESET is 0, therefore, Q should be 0. Q will be 0 and hold 0 until 9 seconds:



At time equal to 9 second, the flip-flop will be active again and check the input SET and RESET. At 9 seconds, SET is 0 and RESET is 0, therefore, if you check Table 8.1, Q has an INVALID state.

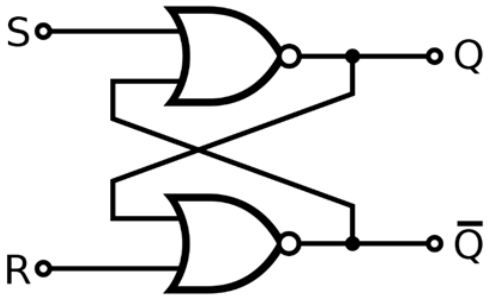


The output signal $\bar{Q}$ is the inverse of signal Q. It means that if Q is 1 $\bar{Q}$ becomes 0, and vice-versa.



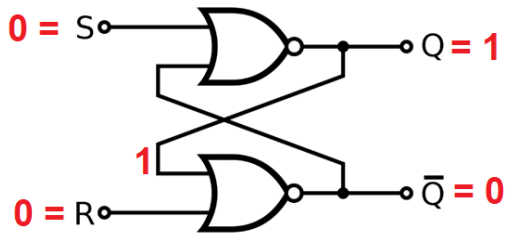
S-R Flip-flop made of NOR gates

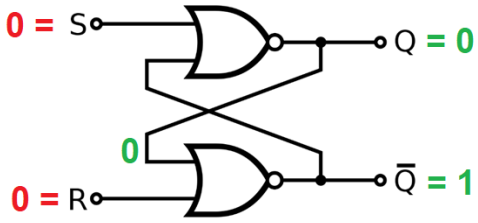
Here is the SR flip-flop built using NOR gates.



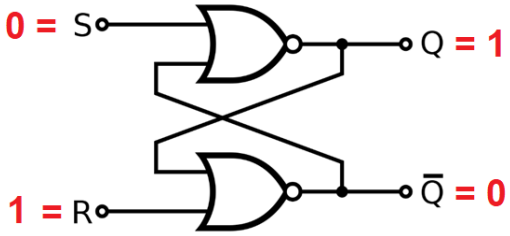
Let us analyze the bit flow within the S-R flip flop. For this we are going to analyze four cases:

Case 1) S = 0, R = 0 ➔ **No change condition; resting state; holding condition**

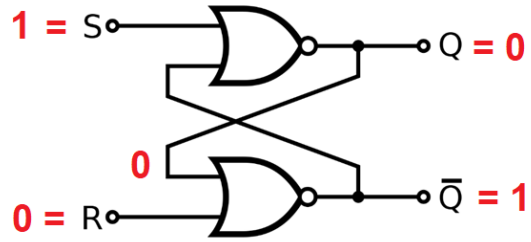




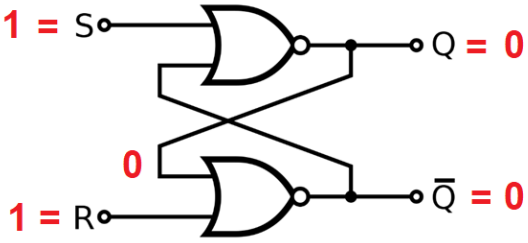
Case 2) S = 0, R = 1 ➔ **Q is set to 1. Set state**



Case 3) S = 1, R = 0 ➔ **Q is set to 0. Reset state**



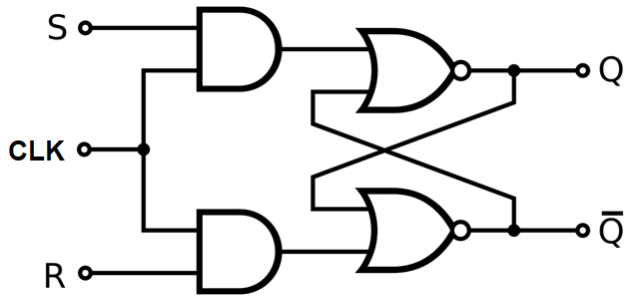
Case 4) S = 1, R = 1 ➔ **Invalid State, not used**



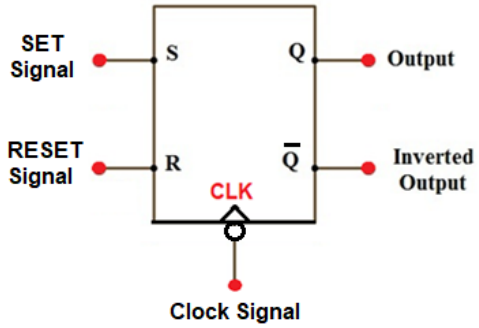
According to the four cases, the resulting truth table for a S-R flip-flop made of NOR gates is:

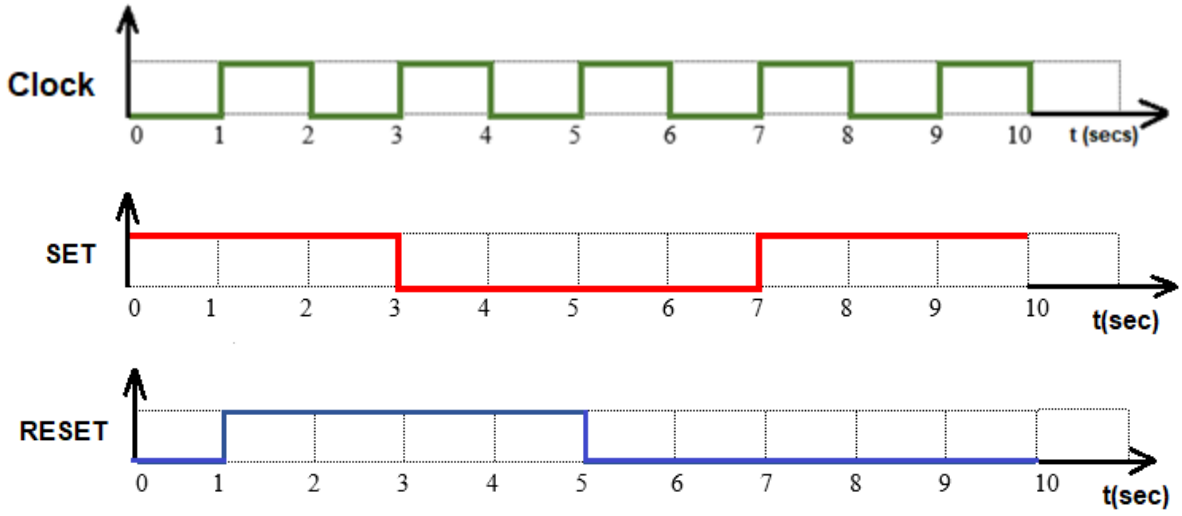
Table 8.2 S-R flip-flop made of NOR gates Truth Table				
Inputs		**Outputs**		**Condition**
S	**R**	**Q**	$\bar{Q}$	
0	0	Q	$\bar{Q}$	No change condition; resting state; holding condition
0	1	1	0	Sets Q to 1. Set state
1	0	0	1	Reset Q to 0. Reset state
1	1	0	0	Invalid State, not used

The circuit of a clocked SR flip–flop using NOR gates is shown below:

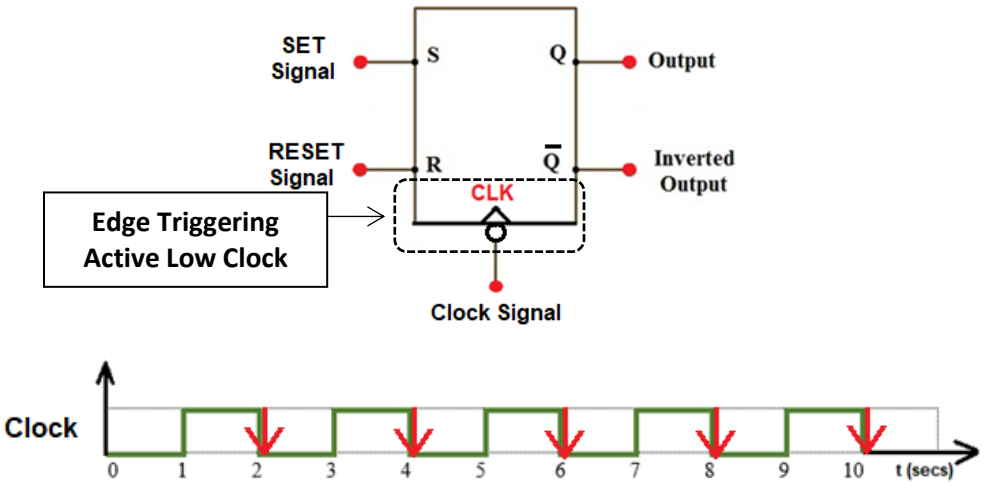


Example 8.2) For the following S-R flip-flop (NOR), sketch the input Q and $\bar{Q}$ assuming the initial Q = 1

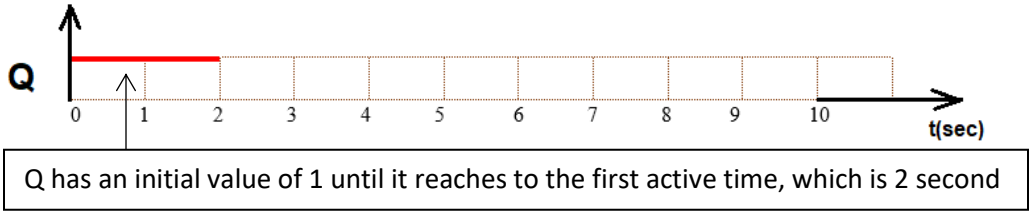




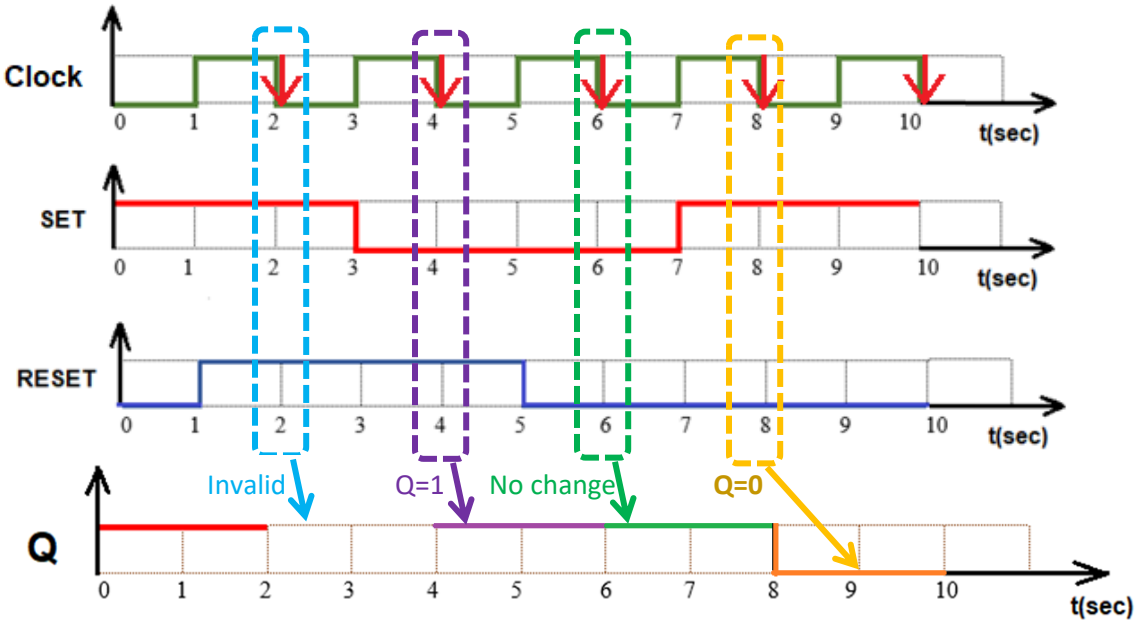
The first step is to identify the type of clock that the S-R flip-flop works with, in this case we have an *Edge Triggering Active Low* clock. This means that the S-R FF will check input SET and RESET at time equal to 2, 4, 6, 8, and 10 second:



To sketch the output Q, first, remember that Q has an initial value of 1 until it reaches to the first active time, which is 2 second. Then from 0 second to 2 second, Q = 1.

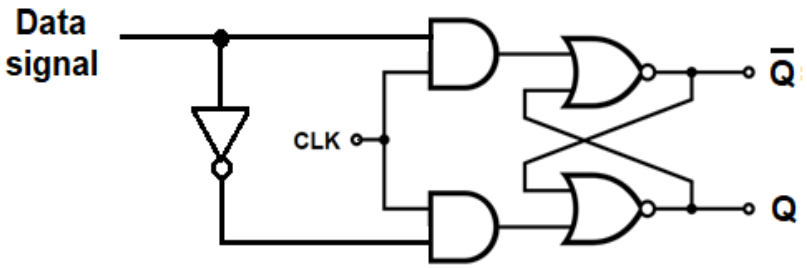


Now, using Table 8.2, S-R flip-flop made of NAND gates Truth Table, sketch the output Q depending on inputs S and R at the active time according to its clock. This means that we have to check the FF's input of SET and RESET at time equal 2, 4, 6, 8, and 10 second.

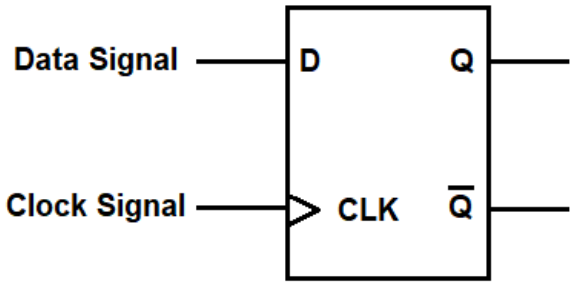


D flip flop

The D stands for Data and it is an improved of the S-R sequential logic circuit. The disadvantage of the basic S-R flip flop in the *Invalid* condition. In order to prevent this from happening an inverter can be connected between the SET and RESET inputs to produce another type of flip flop circuit known as a Data flip flop.

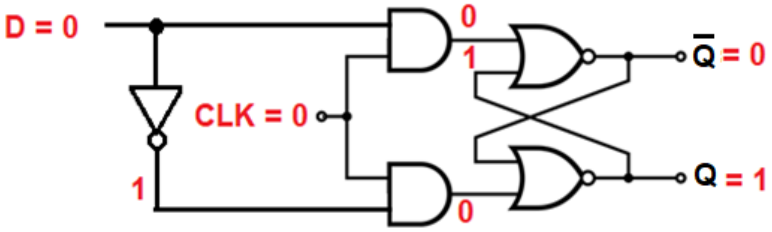


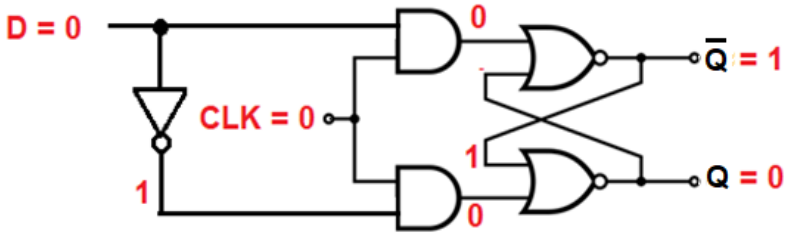
The D flip-flop is one of the most widely used flip–flops. It has a clock signal (CLK) as one input and Data (D) as other input. There are two outputs and these outputs are complement to each other. The symbol of D flip–flop circuit schematic shown below.



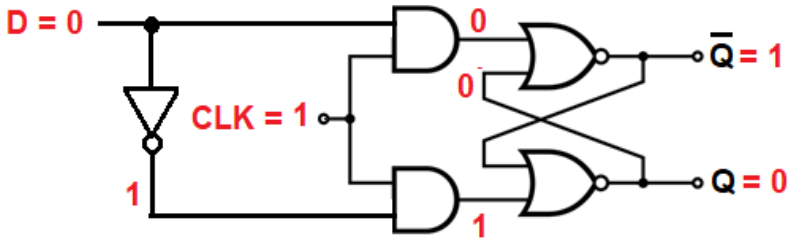
Let us analyze the bit flow within the D flip flop. For this we are going to analyze four cases:

Case 1) D = 0, CLK = 0 ➔ **<u>No change condition; resting state; holding condition</u>**

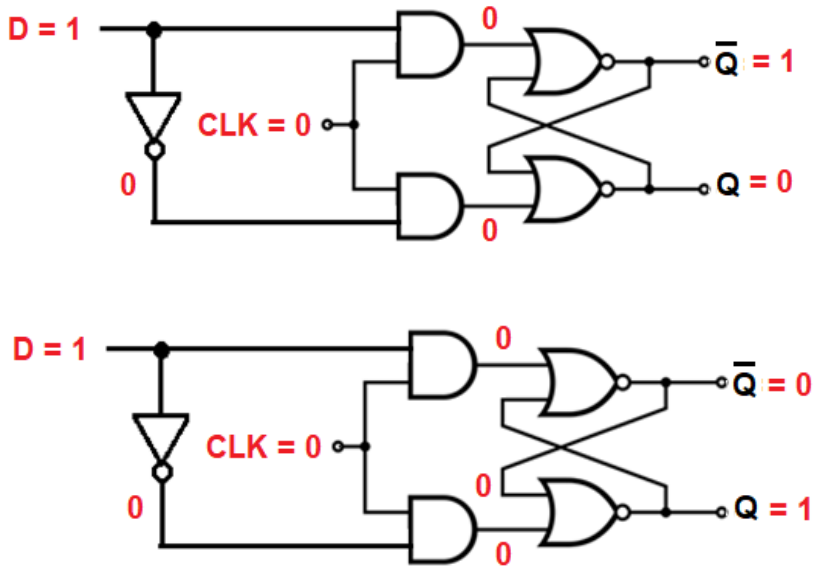




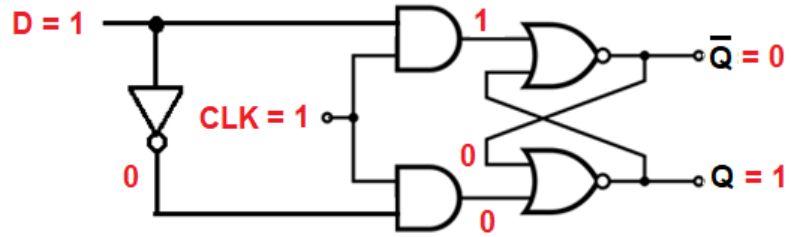
Case 2) D = 0, CLK = 1 ➔ **<u>Reset Q to 0. Reset State</u>**



Case 3) D = 1, CLK = 0 ➔ **<u>No change condition; resting state; holding condition</u>**



Case 4) D = 1, CLK = 1 ➔ **<u>Set Q to 1. Set state</u>**



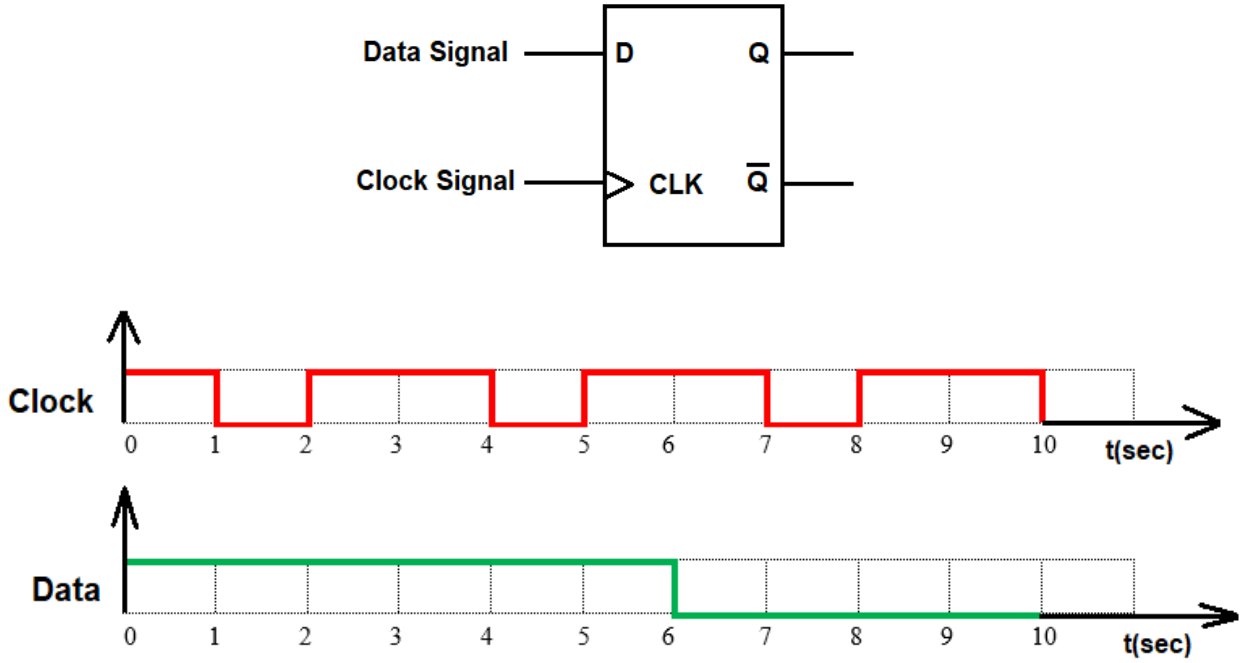
Working with the bit flow through the D flip-flop circuit will generate the following truth table:

Table 8.3 D flip-flop Truth Table

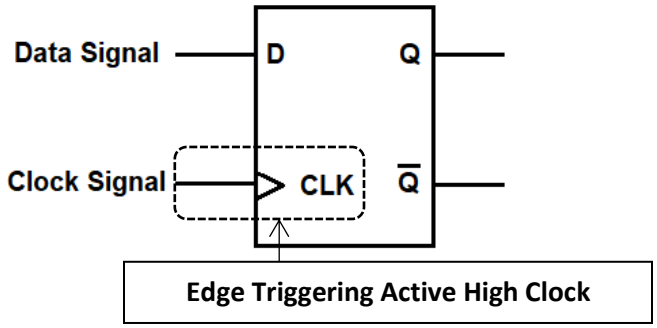
Inputs		**Outputs**		**Condition**
D	**CLK**	**Q**	$\overline{Q}$	
0	0	Q	$\overline{Q}$	No change condition; resting state; holding condition
0	1	0	1	Reset Q to 0. Reset state
1	0	Q	$\overline{Q}$	No change condition; resting state; holding condition
1	1	1	0	Set Q to 1. Set state

According to Table 8.3, when the clock is LOW there will be no change in the output of the flip-flop and it will hold the previous signal. On the other hand, when the clock signal is HIGH, the output Q follows Data input signal.

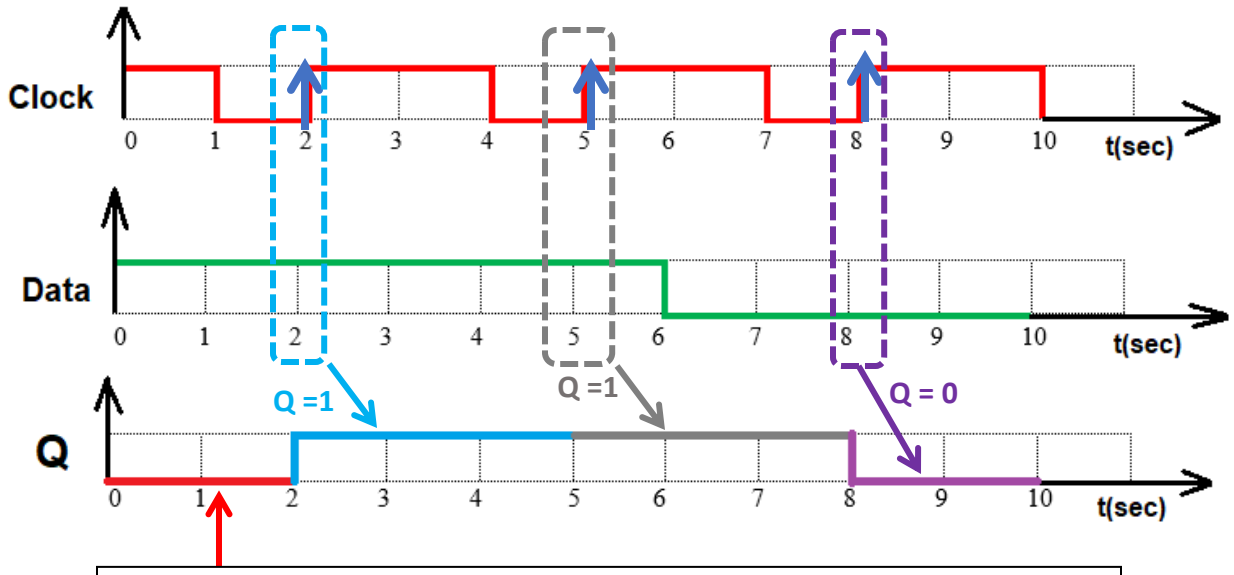
Example 8.3) For the following D flip-flop, sketch the input Q and $\bar{Q}$ assuming the initial Q = 0



The first step is to identify the type of clock that the D flip-flop works with, in this case we have an *Edge Triggering Active High* clock. This means that the D FF will check input Data signal at time equal to 2, 5, and 8 second:

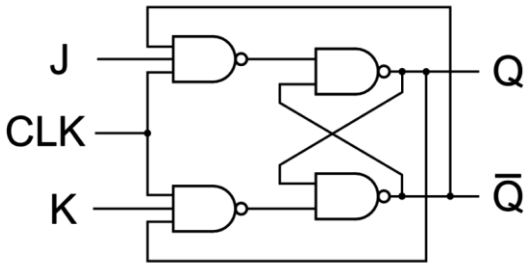


To sketch the output Q, first, remember that Q has an initial value of 0 until it reaches to the first active time, which is 2 second. Then from 0 second to 2 second, Q = 0. After it, using Table 8.3, D flip-flop Truth Table, sketch the output Q depending on Data signal and the Clock signal. This means that we have to check the FF's input Data and Clock at time equal 2, 5, and 8 second



JK Flip-Flop

JK flip–flop is named after Jack Kilby, an electrical engineer who invented Integrated Circuit, IC. A JK flip–flop is a modification of SR flip–flop. This modification allows JK flop-flop to have four different output.



The circuit schematic of the JK FF looks as following:

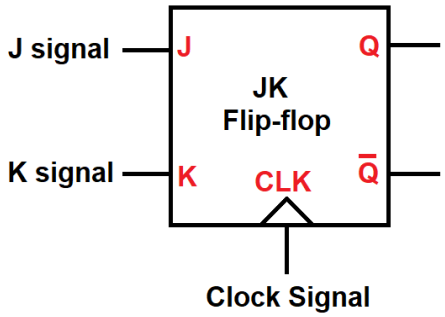
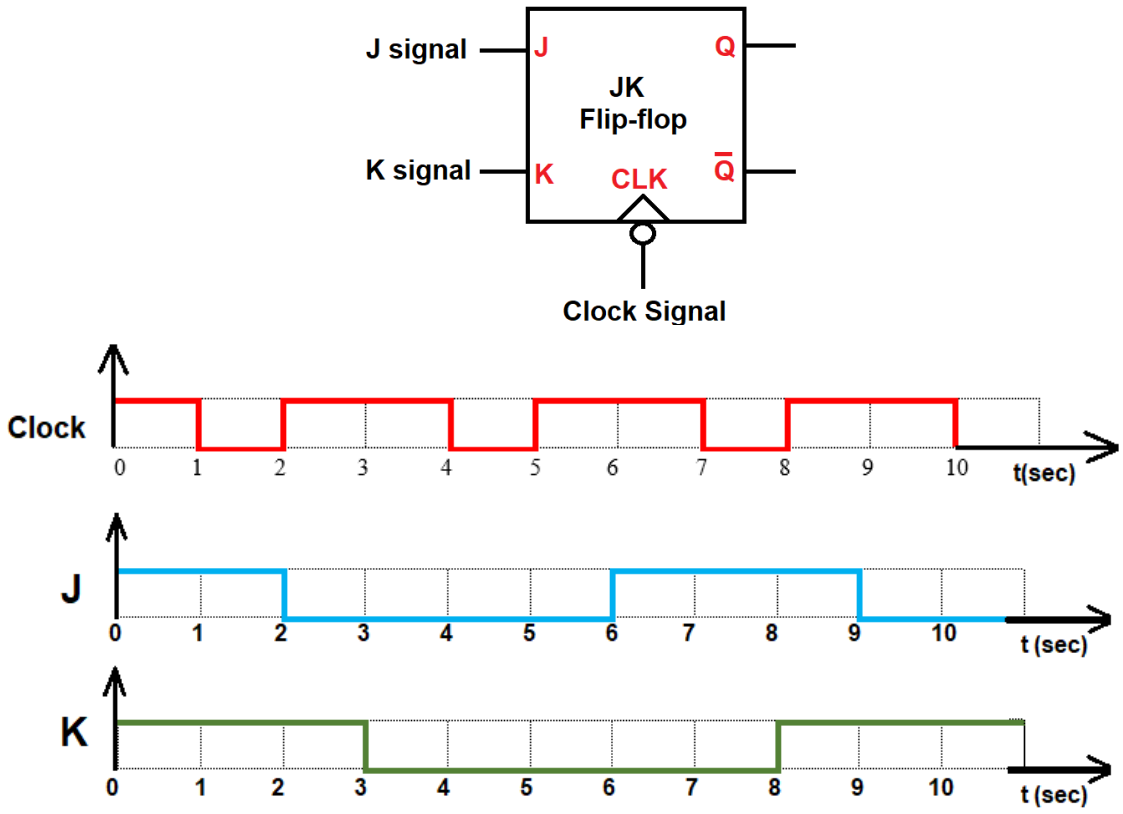
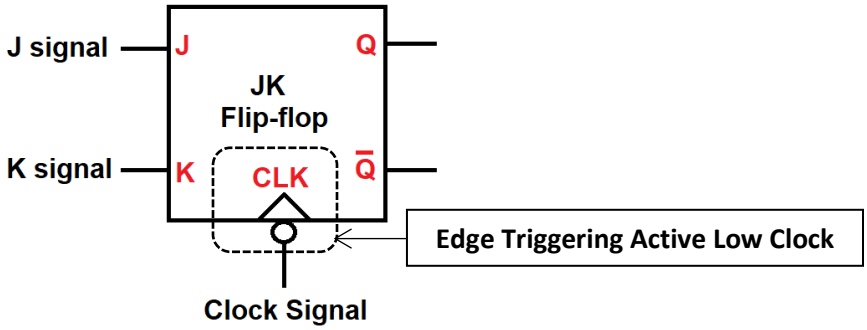


Table 8.4 JK flip-flop Truth Table				
Inputs		**Outputs**		**Condition**
J	**K**	**Q**	$\bar{Q}$	
0	0	Q	$\bar{Q}$	No change condition; resting state; holding condition
0	1	0	1	Reset Q to 0. Reset state
1	0	1	0	Set Q to 1. Set state
1	1	$\bar{Q}$	Q	Invert the Q signal. Toggle.

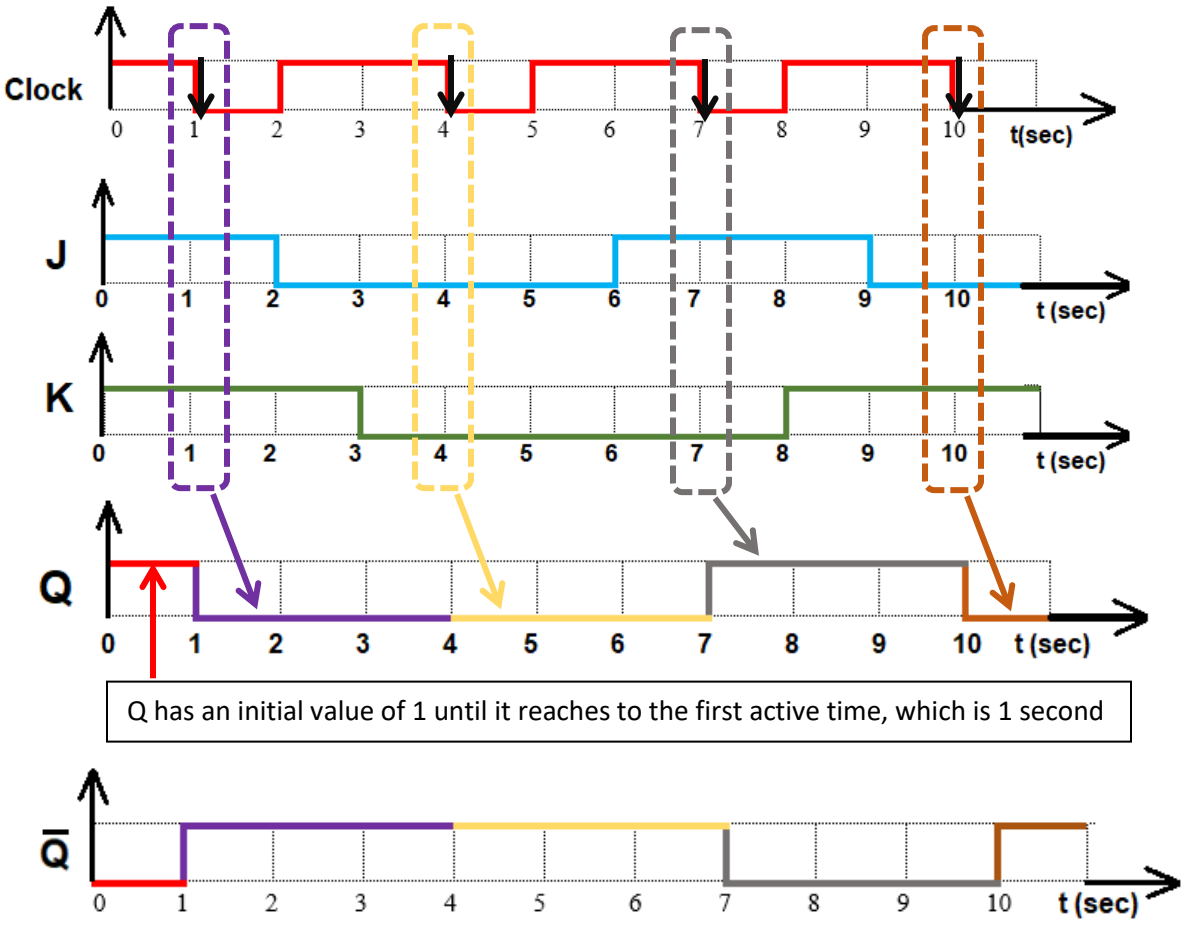
Example 8.4) For the following JK flip-flop, sketch the input Q and $\bar{Q}$ assuming the initial Q = 1



The first step is to identify the type of clock that the JK flip-flop works with, in this case we have an *Edge Triggering Active Low* clock. This means that the JK FF will check input J and K signal at time equal to 1, 4, 7, and 10 second.



To sketch the output Q, first, remember that Q has an initial value of 1 until it reaches to the first active time, which is 1 second. Then from 0 second to 1 second, Q = 1. After it, using Table 8.4, JK flip-flop Truth Table, sketch the output Q depending on the J and K signal. This means that we have to check the FF's input J and K signal at time equal to 1, 4, 7, and 10 second.



The output signal $\bar{Q}$ is the inverse of signal Q. It means that if Q is 1 $\bar{Q}$ becomes 0, and vice-versa.

Where We Use Flip Flops?

Flip flops are widely used in

- Registers: As the flip flops have two stable states, we use them in memory elements like registers, for data storage. Generally we use registers in electronic devices like computers.
- Counters: The groups of interconnected flip flops are uses as counters, to count the increment or decrement of an event occurrence.
- Frequency division: Flip flops are used as frequency division circuits, which divide the input frequency to exactly to its half. Frequency division circuits are used to regularize the frequency of electronic circuits.
- Data transfer: We use shift registers (A special-type of registers) to transfer the data from one flip flop to another, which are connected in a specific order.

(Introduction to flip flops , 2015)

References

All About Circuits. (2019). Retrieved from Edge-triggered Latches: Flip-Flops: https://www.allaboutcircuits.com/textbook/digital/chpt-10/edge-triggered-latches-flip-flops/

Clock Signal. (2019). Retrieved from Wikipedia - The Free Encyclopedia: https://en.wikipedia.org/wiki/Clock_signal

Difference between latches and flip-flops. (2015). Retrieved from Electronics Projects Focus: https://www.elprocus.com/difference-between-latches-and-flip-flops/

Introduction to flip flops . (2015, June 1). Retrieved from Electronics Hubs: https://www.electronicshub.org/flip-flops/