CMP 2203

Lecture 7

Memory Elements

- Logic circuits whose behavior not only depends on present values of the inputs but also past behavior of the circuit are referred to as ***sequential circuits***.
- Consist of storage elements that store values of logic signals that represent the state of the circuits.
- When circuit's inputs change values, new input values either leave the circuit in the same state or cause it to change into a new state.

Synchronous & Asynchronous Sequential Logic

- **Synchronous sequential logic:**

– the time at which transitions between circuit states occurs is controlled by a common clock signal

– changes in all variables occur simultaneously

- **Asynchronous sequential logic**:

– state transitions occur independently of any clock, and normally depend on the time at which input variables change

– outputs do not necessarily change simultaneously

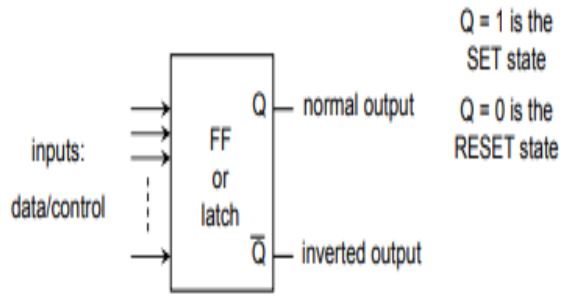
- **Clock**

– a clock signal is a square wave of a fixed frequency

– it is used to trigger state transitions at fixed times in synchronous circuits

Memory Elements: Latches & Flip Flops

- Latches and flip-flops are the basic elements for storing information. One latch or flip-flop can store one bit of information.
- They are bistable devices (2 Stable states)
- Flip-flops and latches (normally) have 2 complementary outputs – usually denoted as Q and Q'.
- There are basically four main types of latches and flip-flops: **SR, D, JK, and T**. The major differences in these flip-flop types are the number of inputs they have and how they change state.



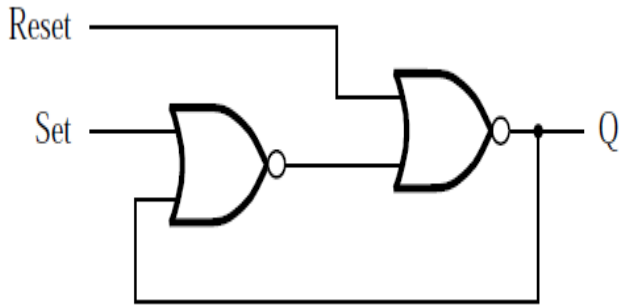
Source: Google images

Memory Elements: Latches & Flip Flops

- The main difference between latches and flip-flops is that for latches, their outputs are constantly affected by their inputs as long as the enable signal is asserted. In other words, when they are enabled, their content changes immediately when their inputs change.
- Flip-flops, on the other hand, have their content change only either at the rising or falling edge of the enable signal. This enable signal is usually the controlling clock signal. After the rising or falling edge of the clock, the flip-flop content remains constant even if the input changes.

Latches

- A latch is a feedback connection of two NOR gates or two NAND gates, which can store one bit of information. It can be set to 1 using the *S* input and reset to 0 using the *R* input.
- Its inputs, Set and Reset, provide the means for changing the state, Q, of the circuit

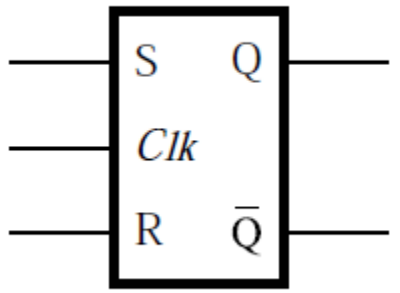


A memory element with NOR gates.

Source : Brown 3rd Edition

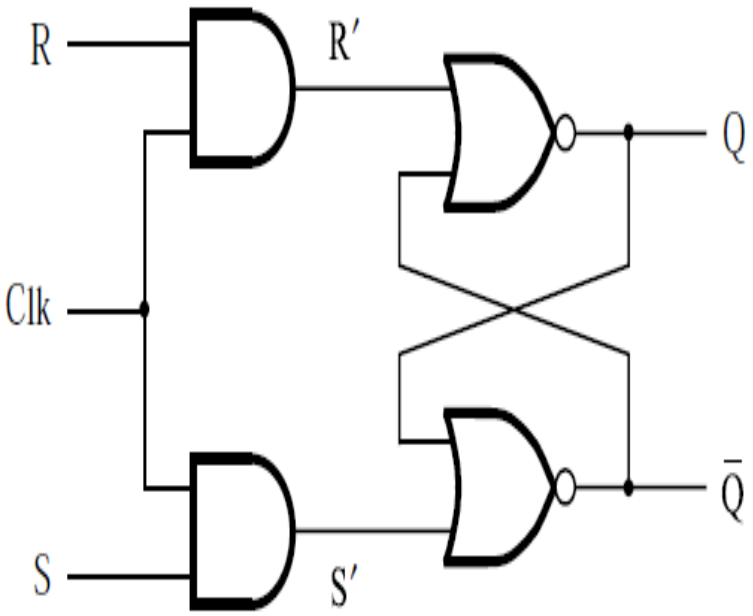
SR Latches

- In order to change the information bit, we need to add inputs to the circuit. The simplest way to add inputs is to replace the two inverters with two NAND gates as shown in Figure. This circuit is called a SR latch.
- In addition to the two outputs Q and Q', there are two inputs S' and R' for set and reset respectively. Following the convention, the prime in S and R denotes that these inputs are active low. The SR latch can be in one of two states: a set state when Q = 1, or a reset state when Q = 0.



Graphical Symbol of an SR Latch

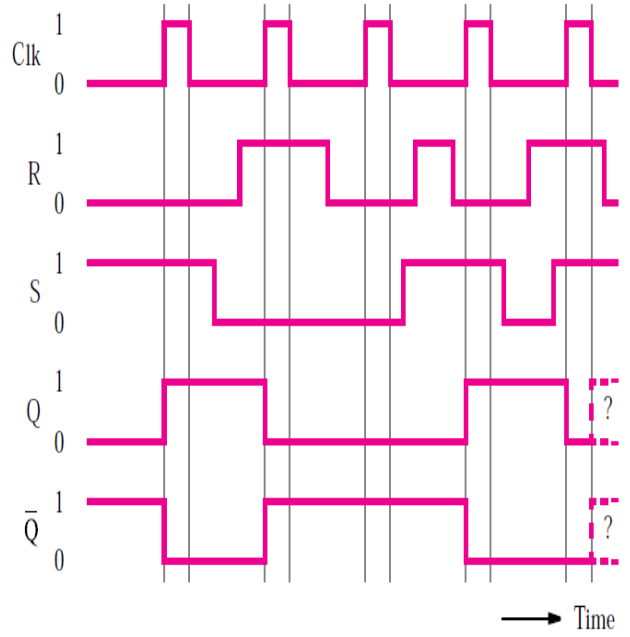
SR Latches



(a) Circuit

Clk	S	R	$Q(t+1)$
0	x	x	$Q(t)$ (no change)
1	0	0	$Q(t)$ (no change)
1	0	1	0
1	1	0	1
1	1	1	x

(b) Characteristic table



(c) Timing diagram

Source : Brown 3rd Ed

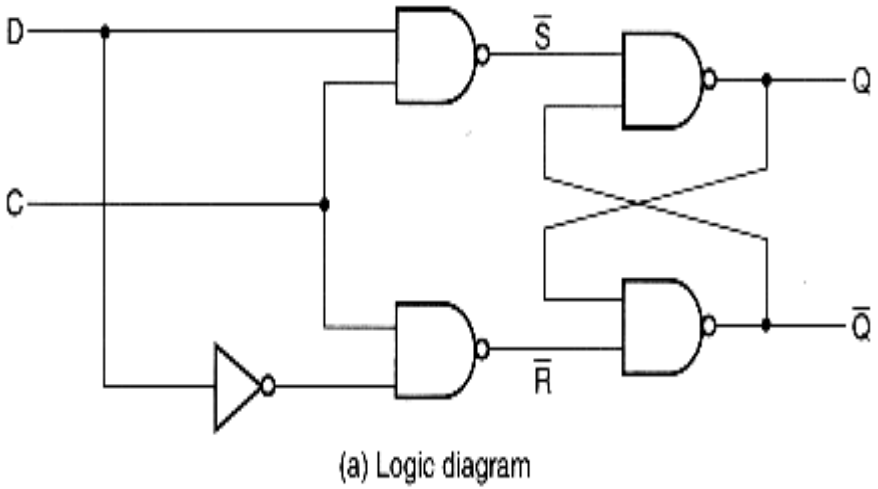
SR Latches

- It includes two AND gates that provide the desired control.
- When the control signal *Clk* is equal to 0, the *S* and *R* inputs to the latch will be 0, regardless of the values of signals *S* and *R*. Hence the latch will maintain its existing state as long as *Clk* = 0.
- When *Clk* changes to 1, the *S* and *R* signals will be the same as the *S* and *R* signals, respectively.
- Circuits of this type, which use a control signal, are called *gated latches*. Because our circuit exhibits set and reset capability, it is called a ***gated SR latch***.

D-Latches

- has a single data input, called *D*, and it stores the value on this input, under the control of a clock signal. It is called a *gated D latch*.
- The D latch is the simple extension of the gated SR latch which removes the possibility of invalid input states. When the enable line of the D latch is high, the output will always reflect the logic level which is present at the D input. When the input of the D latch falls, the last state of the D latch input is trapped and held in the latch. That is why it is also called as a transparent latch. When enable is asserted, the latch is said to be transparent.

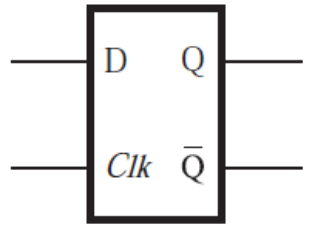
D-Latches



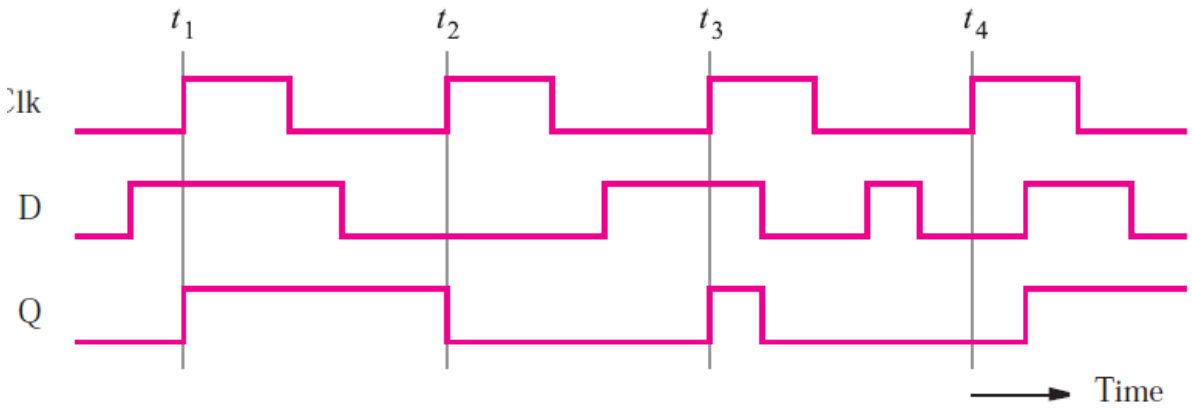
(a) Logic diagram

Clk	D	Q(t + 1)
0	x	Q(t)
1	0	0
1	1	1

(b) Characteristic table



(c) Graphical symbol

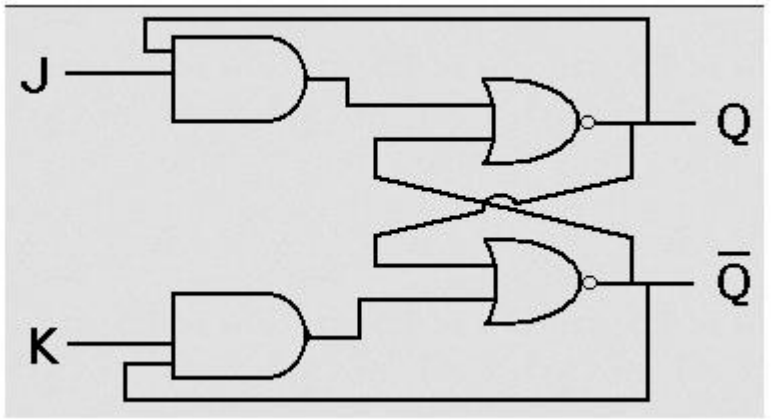


(d) Timing diagram

Source: Brown 3rd Ed

J K Latches

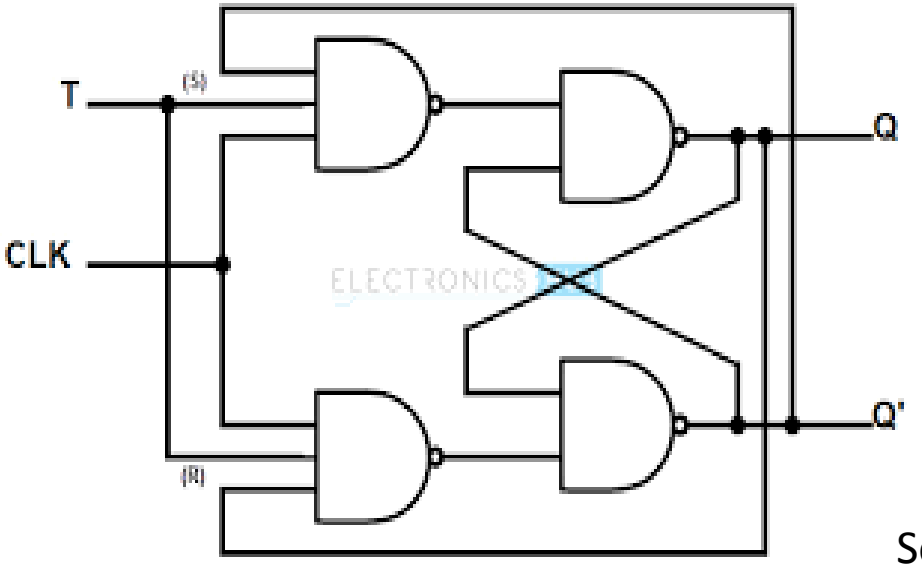
- JK latch is similar to RS latch. This latch consists of 2 inputs J and K as shown in the below figure. The ambiguous state has been eliminated here: when the inputs of Jk latch are high, then output toggles. The output feedback to inputs is the only difference we see here, which is not there in the RS latch.



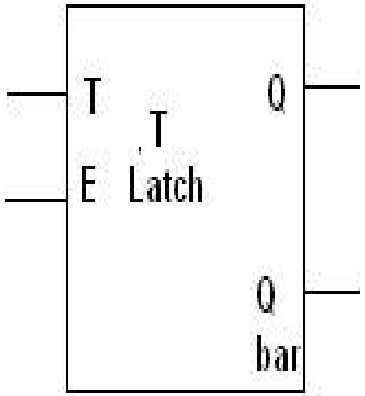
Source: Google images

T Latch

- T latch is formed when the inputs of the JK latch are shorted. When the input is high, then the output toggles.



		Previous		New	
T	E	Q	Q (bar)	Q	Q (bar)
X	0	X	X	PREVIOUS VALUES	
0	1	0	1	0	1
0	1	1	0	1	0
1	1	0	1	1	0
1	1	1	0	0	1

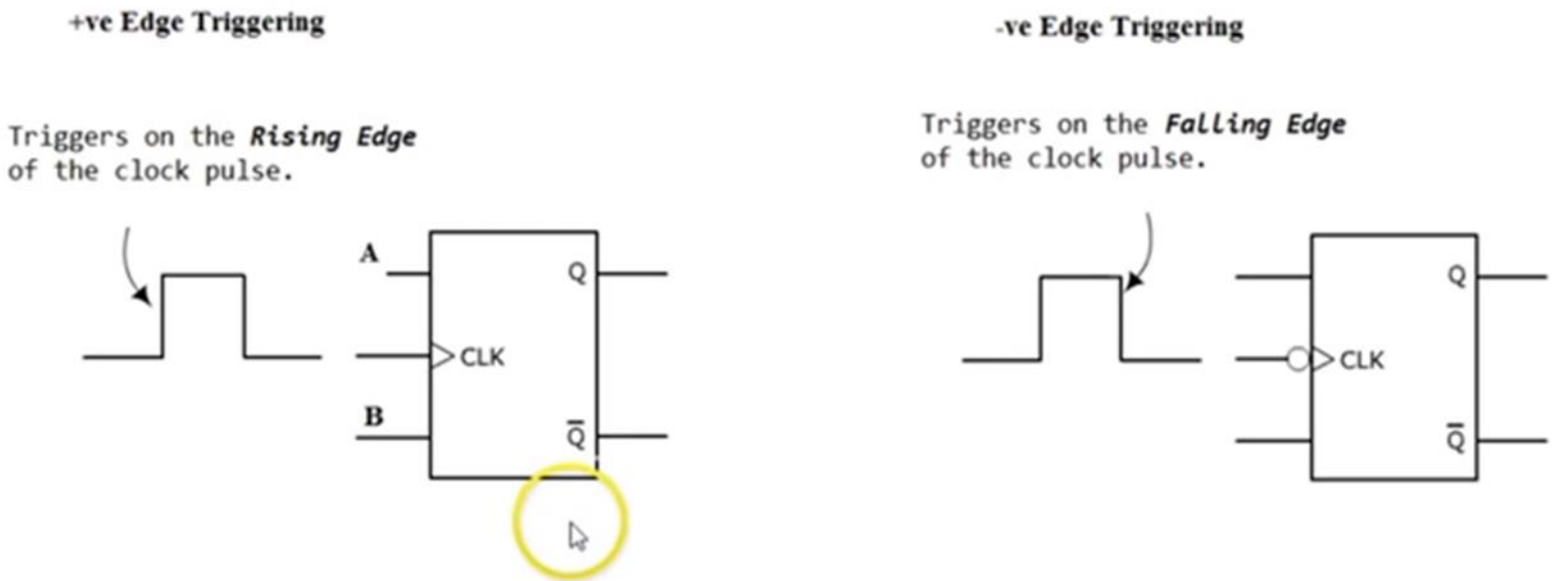


Source: Google images

Level-Sensitive versus Edge-Triggered Storage Elements

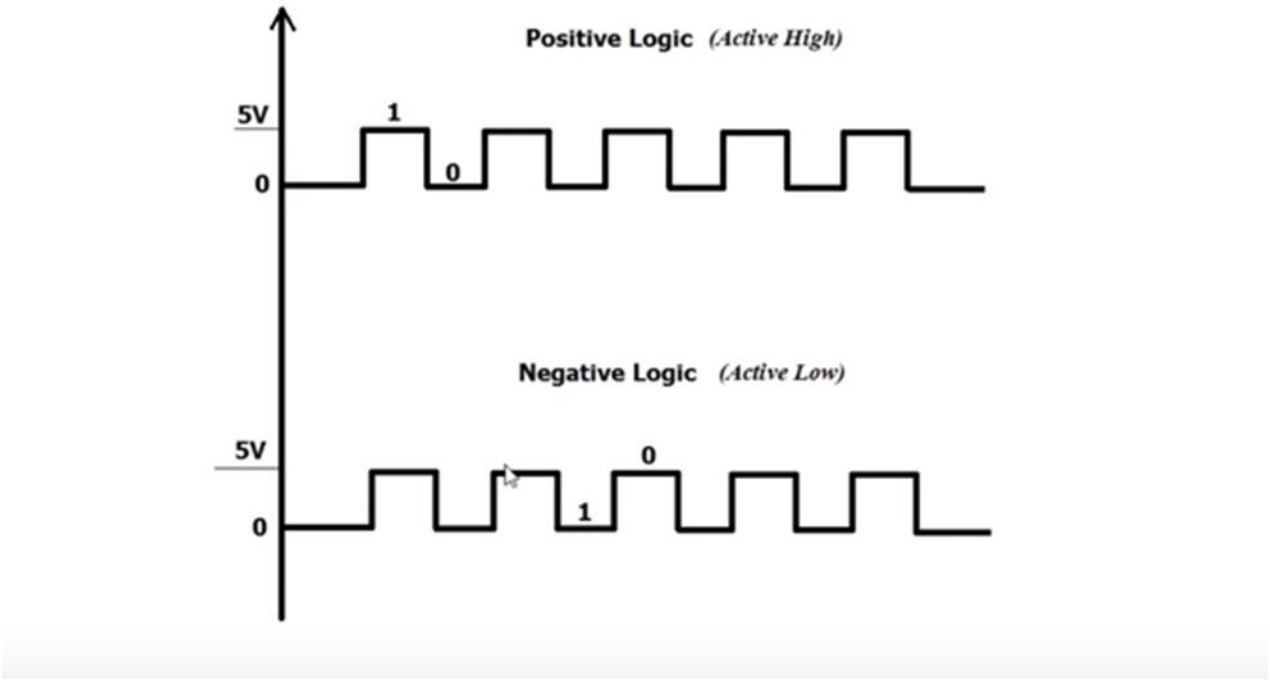
- **Level Triggering:** In level triggering the circuit will become active when the gating or clock pulse is on a particular level. This level is decided by the designer. We can have a negative level triggering in which the circuit is active when the clock signal is low or a positive level triggering in which the circuit is active when the clock signal is high.
- **Edge Triggering:** In edge triggering the circuit becomes active at negative or positive edge of the clock signal. For example if the circuit is positive edge triggered, it will take input at exactly the time in which the clock signal goes from low to high. Similarly input is taken at exactly the time in which the clock signal goes from high to low in negative edge triggering. But keep in mind after the input, it can be processed in all the time till the next input is taken

Edge Triggering

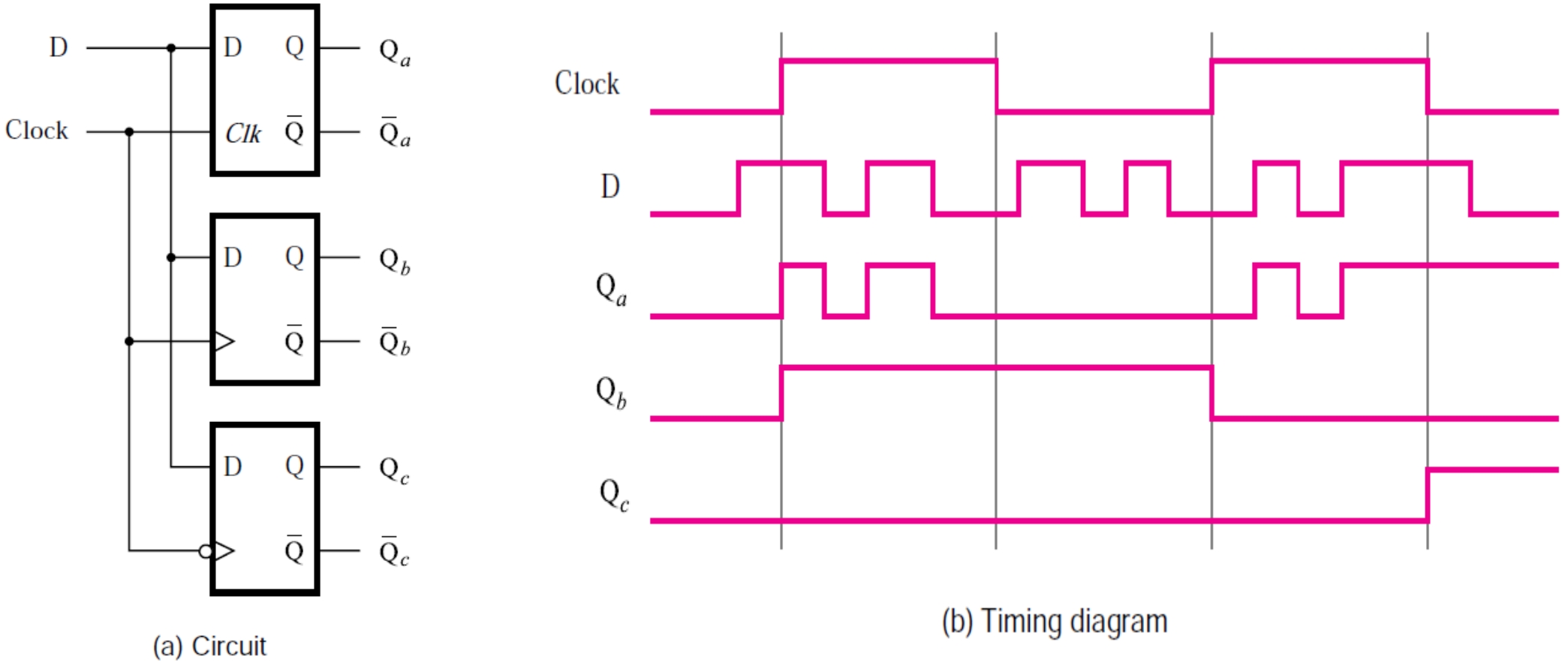


Positive and Negative Edge triggering

Positive and Negative Edge triggering



Level-Sensitive versus Edge-Triggered Storage Elements



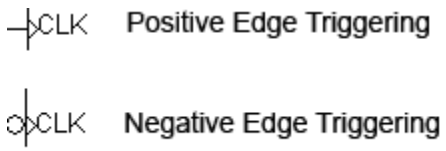
Comparison of level-sensitive and edge-triggered D storage elements.

Level-Sensitive versus Edge-Triggered Storage Elements

- The figure shows three different types of storage elements that are driven by the same data and clock inputs. The first element is a gated D latch, which is level sensitive.
- The second one is a positive-edge-triggered D flip-flop.
- The third one is a negative-edge triggered D flip-flop.
- *D* input changes its values more than once during each half of the clock cycle. Observe that the gated D latch follows the *D* input as long as the clock is high. The positive-edge triggered flip-flop responds only to the value of *D* when the clock changes from 0 to 1. The negative-edge-triggered flip-flop responds only to the value of *D* when the clock changes from 1 to 0.

Flip Flops

- A **flip-flop** is a storage element based on the gated latch principle, which can have its output state changed only on the edge of the controlling clock signal. There are two types;
- **Master-slave flip-flop** is built with two gated latches. The master stage is active during half of the clock cycle, and the slave stage is active during the other half. The output value of the flip-flop changes on the edge of the clock that activates the transfer into the slave stage.
- **Edge-triggered flip-flop** is affected only by the input values present when the active edge of the clock occurs. Symbols used;



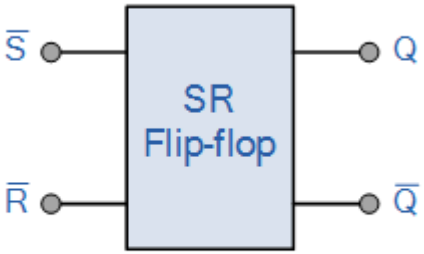
Source: Google Images

Flip Flops

- the state of a flip-flop often depends on the previous state of a circuit (for example, the output of one flip flop may be the input to another), and because each flip-flop and logic gate needs a certain amount of time to switch its output, we usually clock the devices, that is, we synchonize all the flip-flops to change states at the same time with a clocked pulse. Flip-flops are edge triggered; they either change states when the clock goes from 0 to 1 (positive/rising edge) or when the clock goes from 1 to 0 (negative/falling edge).

SR Flip Flop

- The SR flip-flop, also known as a SR Latch, can be considered as one of the most basic sequential logic circuit possible.
- This simple flip-flop is basically a one-bit memory bistable device that has two inputs, one which will “SET” the device (meaning the output = “1”), and is labelled S and one which will “RESET” the device (meaning the output = “0”), labelled R.



Symbol

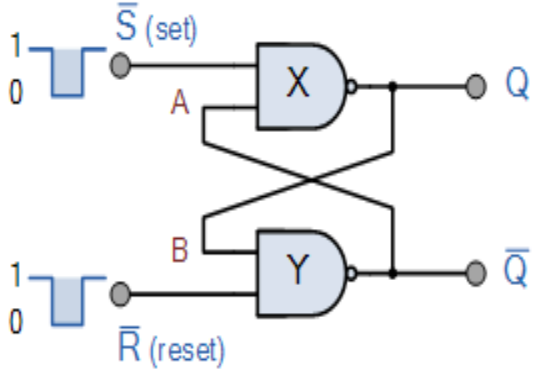
- Then the SR description stands for “Set-Reset”. The reset input resets the flip-flop back to its original state with an output Q that will be either at a logic level “1” or logic “0” depending upon this set/reset condition.

SR Flip Flop

The Set State

- Consider the circuit shown above. If the input R is at logic level “0” (R = 0) and input S is at logic level “1” (S = 1), the NAND gate Y has at least one of its inputs at logic “0” therefore, its output Q must be at a logic level “1” (NAND Gate principles). Output Q is also fed back to input “A” and so both inputs to NAND gate X are at logic level “1”, and therefore its output Q must be at logic level “0”.

- Again NAND gate principals. If the reset input R changes state, and goes HIGH to logic “1” with S remaining HIGH also at logic level “1”, NAND gate Y inputs are now R = “1” and B = “0”. Since one of its inputs is still at logic level “0” the output at Q still remains HIGH at logic level “1” and there is no change of state. Therefore, the flip-flop circuit is said to be “Latched” or “Set” with Q = “1” and Q = “0”.



Circuit

SR Flip Flop

Reset State

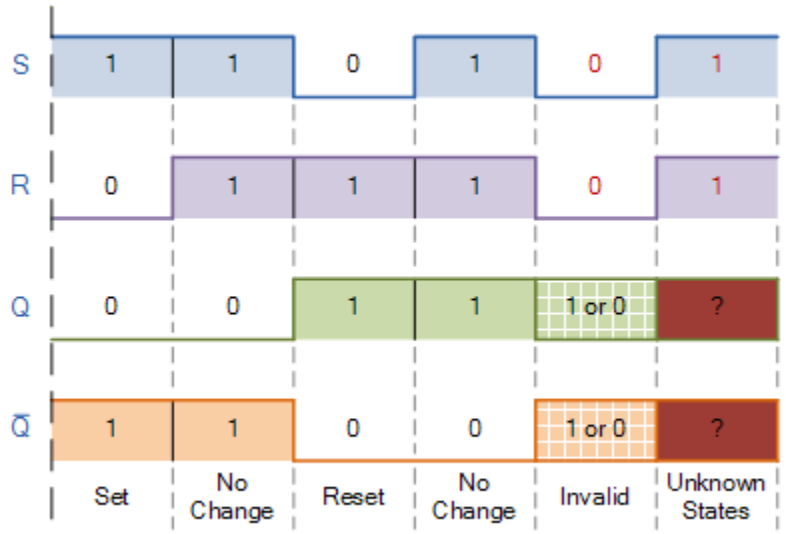
- In this second stable state, Q is at logic level “0”, (not Q = “0”) its inverse output at Q is at logic level “1”, (Q = “1”), and is given by R = “1” and S = “0”. As gate X has one of its inputs at logic “0” its output Q must equal logic level “1” (again NAND gate principles). Output Q is fed back to input “B”, so both inputs to NAND gate Y are at logic “1”, therefore, Q = “0”.

- If the set input, S now changes state to logic “1” with input R remaining at logic “1”, output Q still remains LOW at logic level “0” and there is no change of state. Therefore, the flip-flop circuits “Reset” state has also been latched and we can define this “set/reset” action in the following truth table.

SR Flip Flop

State	S	R	Q		Description
Set	1	0	0	1	Set $\overline{Q}$ » 1
	1	1	0	1	no change
Reset	0	1	1	0	Reset $\overline{Q}$ » 0
	1	1	1	0	no change
Invalid	0	0	1	1	Invalid Condition

Truth Table



Switching diagram

Source : Google images

SR Flip Flop

- It can be seen that when both inputs S = “1” and R = “1” the outputs Q and Q can be at either logic level “1” or “0”, depending upon the state of the inputs S or R BEFORE this input condition existed. Therefore the condition of S = R = “1” does not change the state of the outputs Q and Q.

- However, the input state of S = “0” and R = “0” is an undesirable or invalid condition and must be avoided. The condition of S = R = “0” causes both outputs Q and Q to be HIGH together at logic level “1” when we would normally want Q to be the inverse of Q. The result is that the flip-flop looses control of Q and Q, and if the two inputs are now switched “HIGH” again after this condition to logic “1”, the flip-flop becomes unstable and switches to an unknown data state based upon the unbalance as shown in the switching diagram.

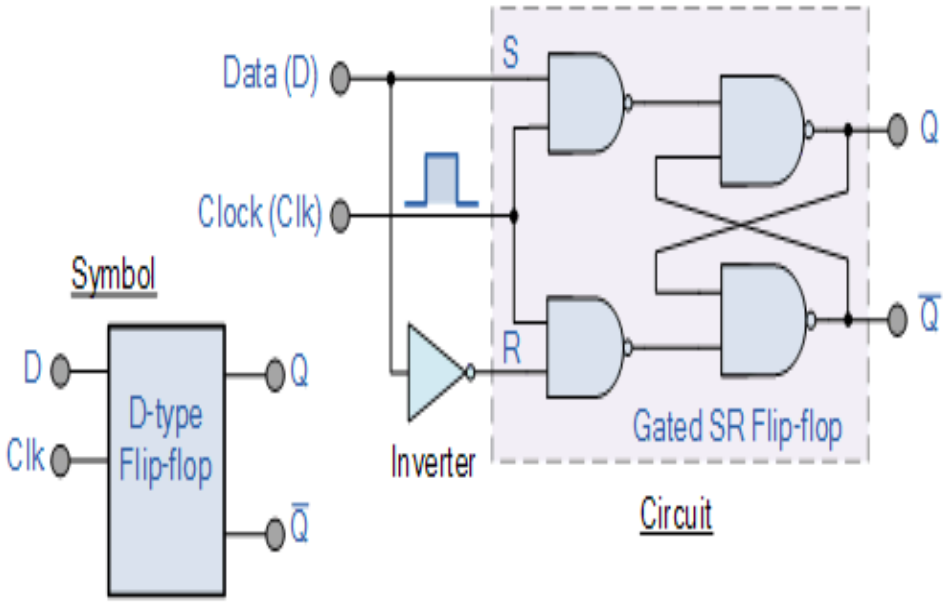
SR Flip Flop

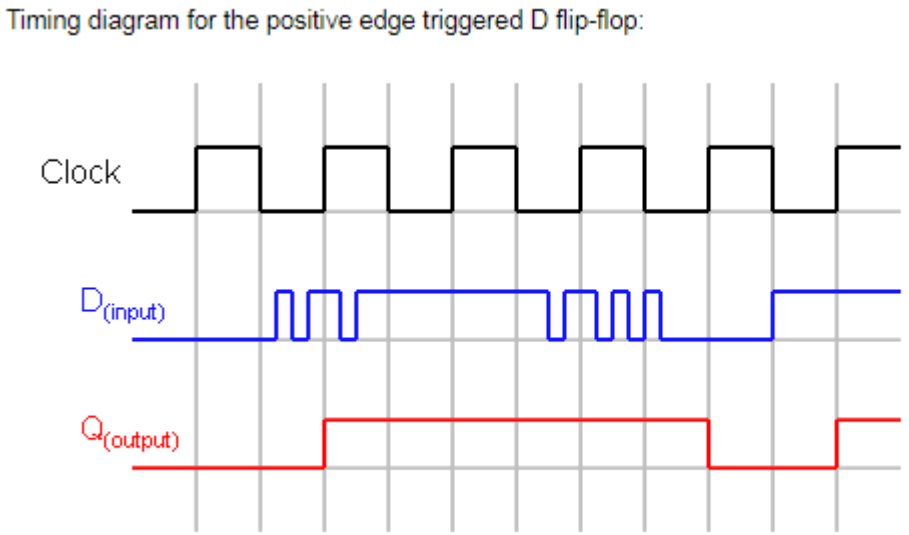
- This unbalance can cause one of the outputs to switch faster than the other resulting in the flip-flop switching to one state or the other which may not be the required state and data corruption will exist. This unstable condition is generally known as its Meta-stable state.
- By adding two extra NAND gates, the timing of the output changeover after a change of logic states at S and R can be controlled by applying a logic 1 pulse to the clock input.
- The main advantage of the clock input is that the output of this flip-flop can now be synchronized with many other circuits or devices that share the same clock. This arrangement could be used for a basic memory location by, for example, applying different logic states to a range of 8 flip-flops, and then applying a clock pulse to CK to cause the circuit to store a byte of data.

D Flip Flop

- The D-type flip-flop is a modified Set-Reset flip-flop with the addition of an inverter to prevent the S and R inputs from being at the same logic level.
- constructed from a gated SR flip-flop with an inverter added between the S and the R inputs to allow for a single D (data) input.
- Then this single data input, labelled “D” and is used in place of the “Set” signal, and the inverter is used to generate the complementary “Reset” input thereby making a level-sensitive D-type flip-flop from a level-sensitive SR-latch as now S = D and R = not D as shown.

D Flip Flop





Source : Google Images

D Flip Flop

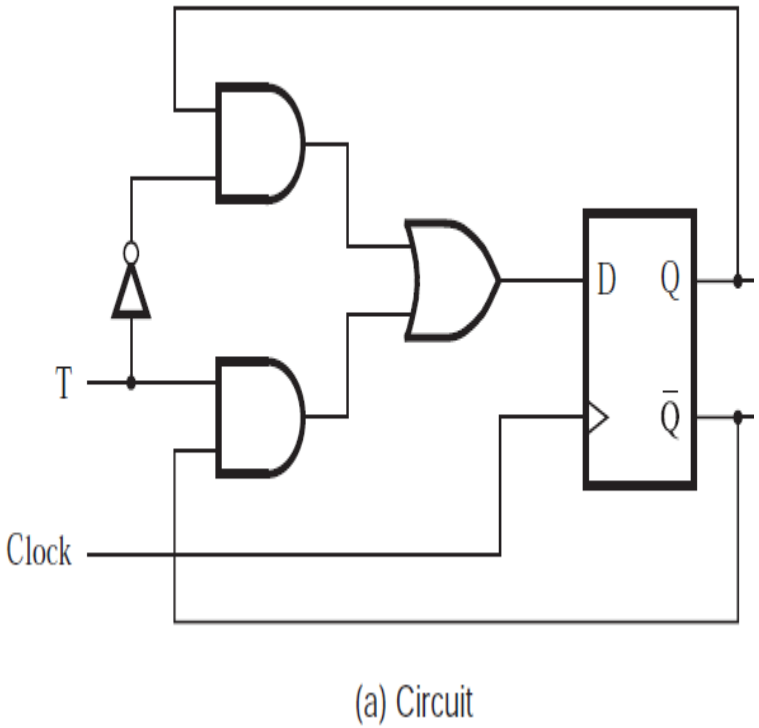
- When a clock input is applied to the D flip flop or during the falling edge of the clock signal, there will be no change in the output. It will retain its previous value at the output Q. If the clock signal is high (rising edge to be more precise) and if D input is high, then the output is also high and if D input is low, then the output will become low. Hence the output Q follows the input D in the presence of clock signal.

CLK	D	Q	Q'
↓	X	Q_{PREV}	Q'_{PREV}
↑	1	1	0
↑	0	0	1

T Flip Flop

- Uses a positive edge triggered D flip flop
- The feedback connections make the input signal D equal to either the value of Q or Q under the control of the signal that is labeled T. On each positive edge of the clock, the flip-flop may change its state Q(t). If T = 0, then D = Q and the state will remain the same, that is, Q(t + 1) = Q(t). But if T = 1, then D = Q and the new state will be Q(t + 1) = Q(t). Therefore, the overall operation of the circuit is that it retains its present state if T = 0, and it reverses its present state if T = 1.
- The name T flip-flop derives from the behavior of the circuit, which “toggles” its state when T = 1.
- The toggle feature makes the T flip-flop a useful element for building counter circuits

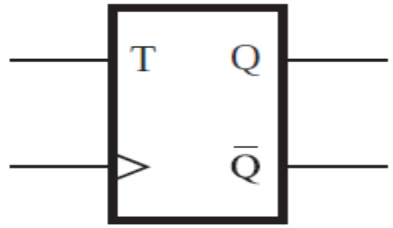
T Flip Flop



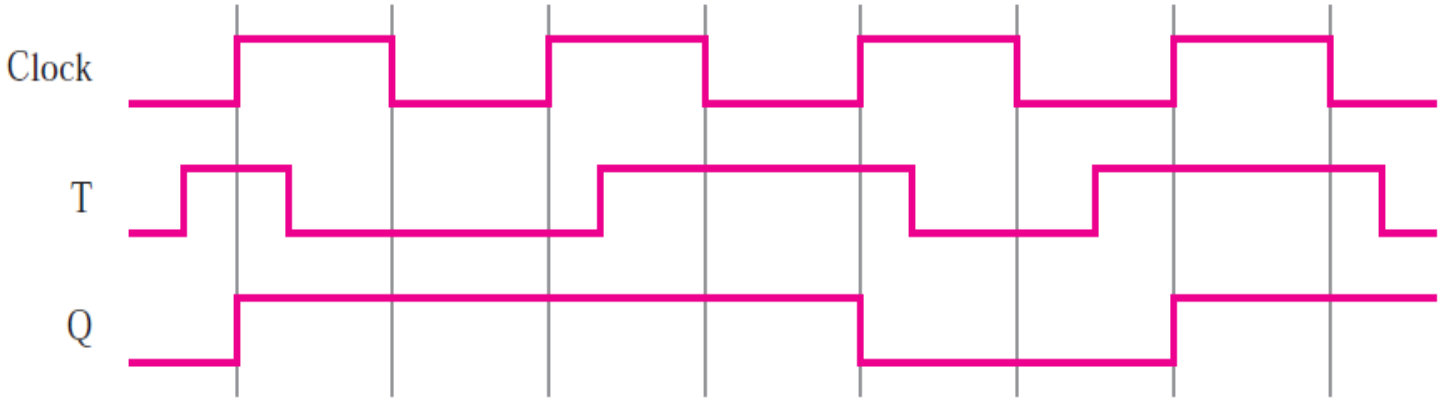
(a) Circuit

T	Q(t + 1)
0	Q(t)
1	Q̄(t)

(b) Characteristic table



(c) Graphical symbol



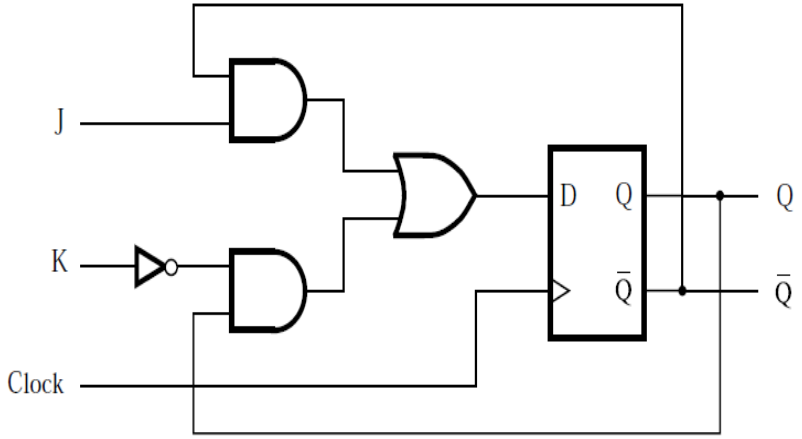
(d) Timing diagram

Source: Brown 3rd Ed

JK Flip Flop

- Instead of using a single control input, *T*, we can use two inputs, *J* and *K*. The circuit is called a ***JK flip-flop***.
- For this circuit the input *D* is defined as

$D = JQ + KQ$

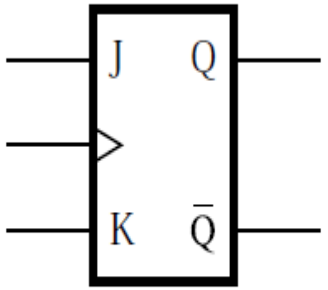


(a) Circuit

Source :Brown 3rd Edition

J	K	Q(t + 1)
0	0	Q(t)
0	1	0
1	0	1
1	1	$\bar{Q}(t)$

(b) Characteristic table



(c) Graphical symbol

JK Flip Flop

- It combines the behaviors of SR and T flip-flops in a useful way. It behaves as the SR flip-flop, where $J = S$ and $K = R$, for all input values except $J = K = 1$.
- For the latter case, which has to be avoided in the SR flip-flop, the JK flip-flop toggles its state like the T flip-flop.

1. The JK flip-flop is a multipurpose circuit which can be used for; straight storage purposes, just like the D and SR flip-flops.
2. also serve as a T flip-flop by connecting the *J* and *K* inputs together

NOTE

- Applications of Latches and Flip flops
- Master – slave D flip flops
- Edge triggered D flip flops

References

- Stephen Brown, Zvonko Vranesic, Fundamentals of Digital Logic with VHDL Design, 3rd Ed, Chapter 7