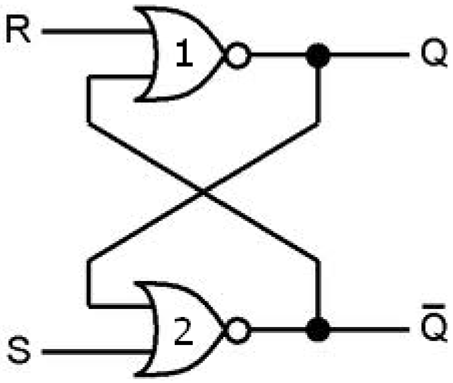
At a point of time in digital electronics, it was realized that you needed to remember past values to calculate new ones. For example, take our very own counters. You need to know what the previous number was, so that you can count to the next one. Nly if you knew the previous number is 2 you can say the next number is 3 right?

By definition, a circuit whose present output depends on present input and past output is called a sequential circuit. Counters are sequential circuits. So a way had to be found to remember the previous state. This led to the development of a latch. A latch is a simple circuit that “latches on” or stores 1 or 0.



As seen from the circuit, a latch has two NOR (complement of OR) gates the output of one given as input of another. One output is always the complement of the other Q and Q’ It will soon be clear why this is so and also why the inputs are named R & S. Assuming that you know the truth of NOR, let’s see how the latch works.

Let’s assume the initial input of R and S is 1 and 0 respectively. If any one of the inputs for NOR is 1, the output is 0. Since R is 1 in 1st NOR, Q has to be zero. This Q is fed to the 2nd NOR. Now since both Q and S are zero, the output Q’ is 1 (notice Q and Q’ are complementary).

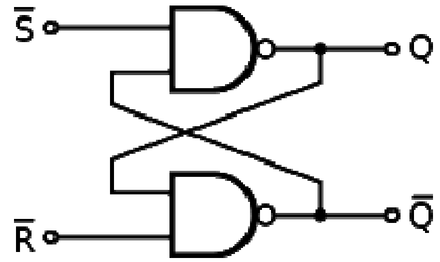
Now let’s make R = 0 so both R and S are 0. Since Q is 0 and S=0, output of 2nd NOR is 1 and this Q’ = 1 makes the output of 1st NOR 0 causing Q = 0. So both Q and Q’ “retain” their previous value or their values haven’t changed. Now let’s make S= 1 and R=0.

Since S=1, Q’ becomes 0. Now Q’ and R are 0 making Q = 1. Now let’s again make both R and S zero. But now Q = 1 and this causes Q’ to be 0. Since both Q’ and R are zero, Q = 1. Again the values are retained.

So when S = 1 & R = 0, Q = 1. This is called set because you are setting the value to 1. In digital vocabulary, set means 1 and reset means 0. Again when S = 0 & R = 1, Q =0. This is reset because the value of Q is resetted to 0. Q and Q’ are noticeably complimentary.

But what happens when both R and S are 1. This condition if forbidden because, such a state gives an undesirable output.

This is called a latch based on NOR. A similar one with NAND looks like this



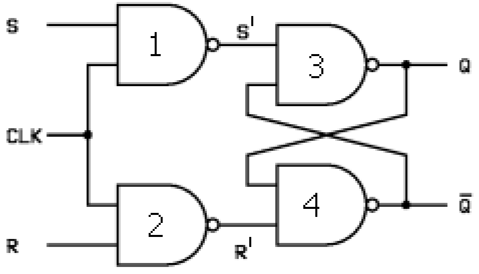
The working is almost similar only the inputs are different, you have complement of S and R which means when S = 0 it is set state and when R = 0 as reset state.

When you want to change Q, you apply suitable values to S and R. But as long as both S & R are 0, Q and Q' don't change (memory). Last assigned value of Q and Q' are retained.

So the characteristics table for this latch is:

NOR latch					NAND latch				
S	R	Q	Q'		S'	R'	Q	Q'	
0	0	(memory)			0	0	(forbidden)		
1	0	1	0	(set)	1	0	0	1	(reset)
0	1	0	1	(reset)	0	1	1	0	(set)
1	1	(forbidden)			1	1	(memory)		

So, we have our basic memory device. But it isn't good enough because S=0, R=0 has to be maintained for memor condition. Any immediate change in S and R affects Q and Q'. We want the latch to change values only when we need them. This is where a clock signal comes. A clocked latch is called a flip-flop.



When the clock is '0', irrespective of S & R, the 1st and 2nd NAND gates output 1. Remember a NAND latch has S', R' inputs. So if S' and R' are 1 (the notations are so used for our easier understanding only) it is memory state. So as long as clock is low any change in S and R will not affect Q and Q'. This is clocking concept.

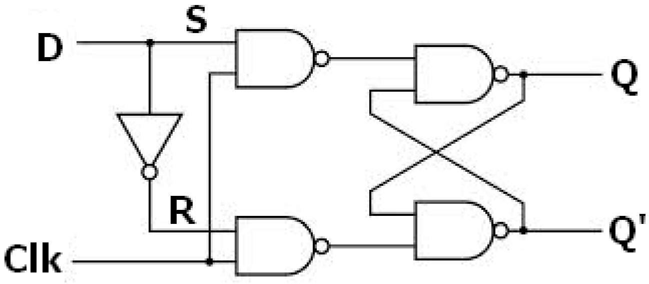
When clock is high, the 1st and 2nd NAND output are complements of S and R and the usual functions of NAND latch is carried out. This is called a S R flip flop.

S R Flip Flop

Clk	S	R	Q	Q’
0	X	X	(memory)	
1	0	0	(memory)	
1	0	1	0	1
1	1	0	1	0
1	1	1	(forbidden)	

X – don’t care or whatever value

Even after clocking the forbidden condition exists. Now let us analyze the characteristics table of S R flip-flop. We don’t want both S = 1 and R = 1 because it is forbidden. We don’t need S = 0 and R = 0, because we can use the clock for memory purposes. The essential required conditions are when S = 1 and R = 0 or S = 0 and R = 1. In both the essential conditions S and R are complements. So, if S and R are made to be complements then we can effectively remove both the forbidden condition and the redundant condition. So we take one input ‘D’ and we apply the same to ‘S’ and we complement D and apply that to R.



And this is called the D flip flop. D stands for delay or data. It is more suited to be called Data flip flop because, as long as the clock is high the value of D is obtained at Q. If clock is low the value of D doesn’t affect Q and Q has the last retained value.

D flip flop

Clock	D	Q	Q’
0	X	(memory)	
1	1	1	0
1	0	0	1

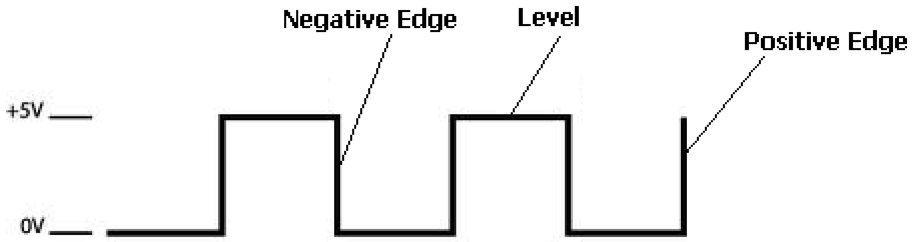
One should actually talk about JK flip flops before T flip flops but to talk about JK flip flop concepts one has to also talk about master-slave configuration etc all of which is out of the scope of this article. For sake of blunt knowledge, a T flip lop is a JK flip flop in which J = K. For now, the characteristics table of T flip flop will be more than sufficient.

T flip flop

Clock	T	Q	Q’	
0	X	(memory)		
1	0	Q	Q’	(no change)
1	1	Q’	Q	(both outputs are complemented)

So when T = 1, Q is complemented. This complementing action is called toggling and so toggle – T flip flop.

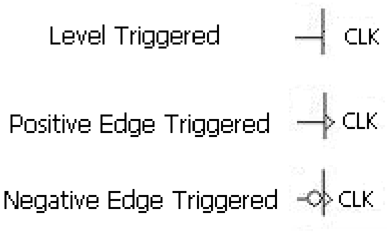
One more basic concept is triggering or how does the clock enable or disable the flip flop.



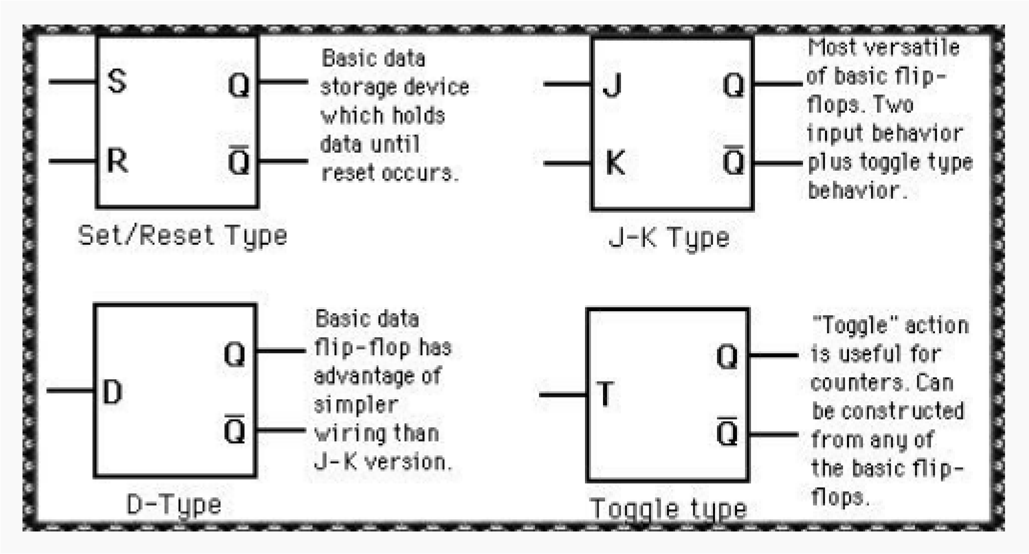
We saw that as long as clock remains high changes can occur. Such a change is called level triggering. The change is triggered as long as the clock is in logic high level.

More complicated flip flop circuits have been developed (especially for high frequency applications) that respond only to changes in the clock signal i.e., change from low to high or high to low. If a flip flop is triggered by a change in clock from low to high it is called positive edge trigger. If a flip flop is triggered by a change in clock from high to low it is called negative edge trigger.

Commercially available D flip-flop IC 7474 is positive edge triggered. Commercially available JK flip-flop IC 7476 is negative edge triggered. Following are representations of the triggering often seen in IC datasheets etc.



This might be useful to understand flip flop’s uses.


S
R
Q
Q̄
Basic data storage device which holds data until reset occurs.
Set/Reset Type
J
K
Q
Q̄
Most versatile of basic flip-flops. Two input behavior plus toggle type behavior.
J-K Type
D
Q
Q̄
Basic data flip-flop has advantage of simpler wiring than J-K version.
D-Type
T
Q
Q̄
"Toggle" action is useful for counters. Can be constructed from any of the basic flip-flops.
Toggle type