Lesson: Flip-Flops and Counters

Lesson Developer: Vijaya Goel

College / Department: Department of Computer Science, University of Delhi

TABLE OF CONTENTS

1-1 FLIP-FLOPS

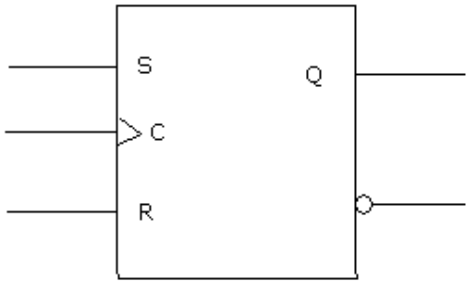
Flip-flops are the memory elements used in clocked sequential circuits.

A flip-flop is a binary storage device that stores one bit of information. It maintains a binary state until directed by a clock pulse to switch the state. It has two outputs, one corresponding to normal value of the bit stored in it and the other is the complement of it. There are basically four main types of flip-flops: SR, D, JK and T. The difference among these is in the number of inputs they possess and in the manner in which the inputs affect the binary state. Flip-flops respond to clock transitions. Latches are also storage elements. However, they operate with signal elements, i.e. , a positive level response in the clock input allows the input changes to change the output while the clock pulse remains at logic 1.

SR flip-flop

It has three inputs, labeled *S* for set, *R* for reset and *C* for clock. It has two outputs labeled Q and Q', which are complement of each other. In the absence of a clock pulse, the output remains unchanged at the previously acquired value, Q_n , which is independent of the present-time data inputs S and R. Only when the clock signal changes from 0 to 1, can the output change to a new value, Q_{n+1} , according to the values in inputs S and R. The graphic symbol is shown in Fig. 1(a).

If R = 0 and S = 1, the output Q is set to 1 when C changes from 0 to 1. If R = 1 and S = 0, the output Q is cleared to 0 when C changes from 0 to 1. If both S and R are 0 during the clock transition, the output does not change. If a 1 is applied to both the *S* and *R* inputs, the output is unpredictable and may go to either 0 or 1, depending on internal timing delays that occur within the circuit. The operation of SR flip-flop is summarized using characteristic table shown in Fig. 1(b). Q(t) denotes *present state* ,i.e. , the binary state of Q output at a given time. Q(t+1) refers to the *next state* one clock pulse later. The indeterminate condition makes the SR flip-flop difficult to manage and therefore it is seldom used in practice.



(a) Graphic symbol

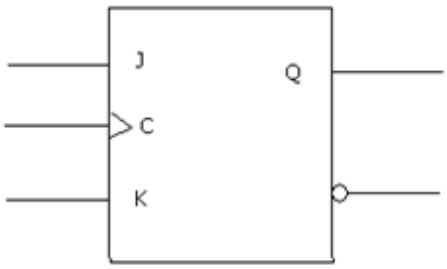
S	R	Q (t+1)	
0	0	Q(t)	No change
0	1	0	Clear to 0
1	0	1	Set to 1
1	1	?	Indeterminate

(b) Characteristic table

Fig. 1 *SR* flip-flop

JK flip-flop

Inputs J and K set and clear the flip-flop as in SR flip-flop. In JK flip-flop, the letter J denotes set and the letter K denotes clear. JK flip-flop defines the invalid state of the SR type. When both J and K inputs are 1, the flip-flop complements its state, i.e., if Q=0, it switches to Q=1 and vice versa.
The graphic symbol and characteristic table are shown in Fig. 2.



(a) Graphic symbol

J	K	Q (t+1)	
0	0	Q(t)	No change
0	1	0	Clear to 0
1	0	1	Set to 1
1	1	Q'(t)	Complement

(b) Characteristic table

Fig. 2 *JK* flip-flop

Consider the block diagram of sequential circuit shown in fig. 3. As can be seen, the outputs of memory elements are connected to the inputs of the memory elements through the combinational circuit, i.e. , there is a feedback path between the combinational circuit and the memory elements. Use of latches as the storage elements causes serious problem. Since the latches operate at signal levels, new output state may occur as a result of change in the inputs applied to the latches while the clock pulse stays at the level-1. This makes the output indeterminate. This is known as **race condition**. It occurs when the clock pulse duration is greater than the signal propagation delay from input to output. This problem occurs in JK flip-flop that are sensitive to *clock pulse duration* (instead of pulse transition). When the value of J and K inputs is 1 while the clock pulse is active, the output keeps toggling again and again in one clock pulse and becomes indeterminate. Two ways to solve the feedback timing problem efficiently are to use: (1) master-slave flip-flop, (2) edge-triggering. Each of the ways is discussed later in this section.

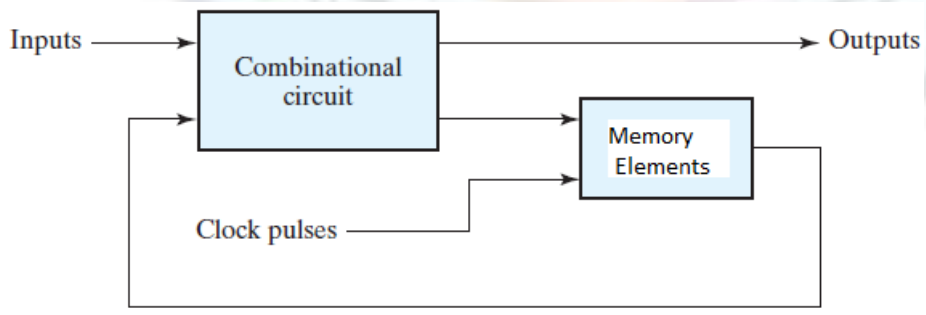
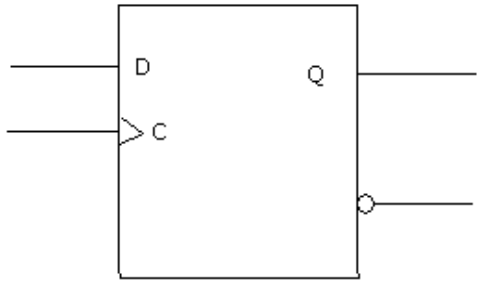


Fig. 3 Block diagram of sequential circuit

D Flip-flop

The D(data) flip-flop has two inputs : D and C(Clock). D flip-flop is obtained from SR flip-flop by adding an inverter between the original S and R inputs and replacing them with just one input D (for data). The D input of the flip-flop is directly given to S and the complement of this value is given as the R input. The value of D is sampled only when CP goes from 0 to 1. If *D* = 1, the Q output goes to 1 on the positive-edge transition of clock pulse. If *D* = 0, output Q goes to 0 on the clock's positive edge transition. It is also called a Delay flip-flop. The graphic symbol and characteristic table are shown in Fig. 4.



(a) Graphic symbol

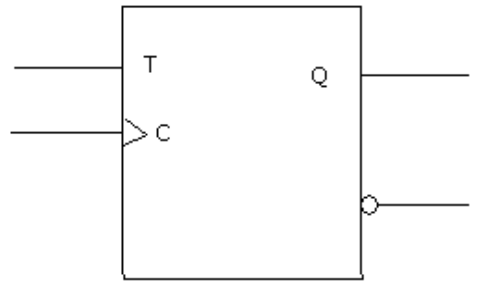
D	Q (t+1)	
0	0	Clear to 0
1	1	Set to 1

(b) Characteristic table

Fig. 4 *D* flip-flop

T flip-flop

The T(Toggle) flip-flop is shown in fig. 5. It is obtained from the JK flip-flop if both J and K inputs are connected together. When T=0(J=0,K=0), the state of flip-flop does not change on clock transition. When T=1(J=1,K=1), the state of flip-flop gets complemented on clock transition.



(a) Graphic symbol

T	Q (t+1)	
0	Q(t)	No change
1	Q′(t)	Complement

(b) Characteristic table

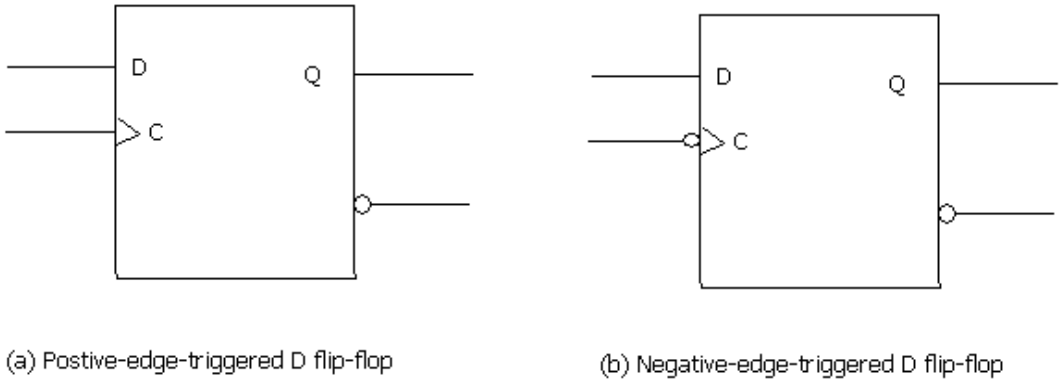
Fig. 5 *T* flip-flop

Edge-triggered flip-flops

A clock pulse has two signal transitions: from 0 to 1 and the return from 1 to 0. In an edge-triggered flip-flop, the output changes only when the clocking signal changes state ,i.e. , either at the positive edge (rising edge) or at the negative edge (falling edge) of the clock pulse. All the flip-flops discussed above are edge-triggered.

Fig. 6 shows the positive and the negative edge-triggered *D* flip-flop. The symbol > refers to *dynamic indicator* indicating that the flip-flop responds to the edge transition of the clock. A

bubble outside the block adjacent to the dynamic indicator designates a negative edge for triggering the circuit. The absence of a bubble designates a positive-edge response.

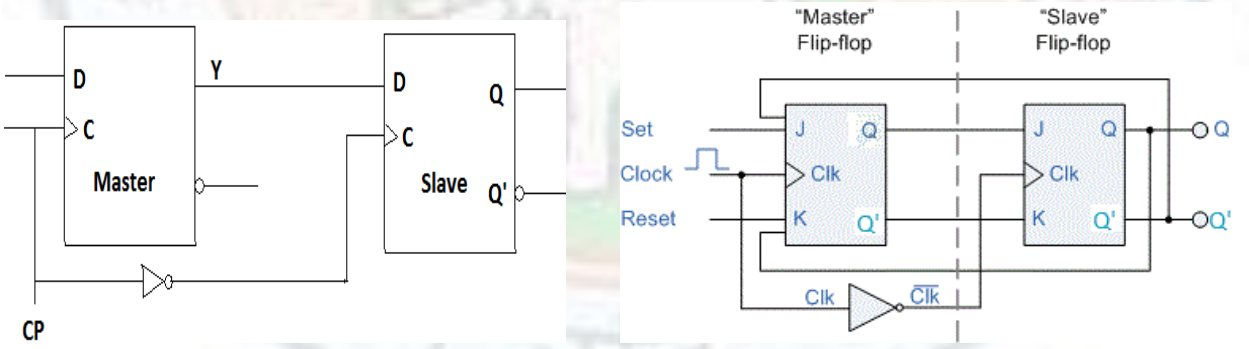


(a) Postive-edge-triggered D flip-flop (b) Negative-edge-triggered D flip-flop

Fig. 6 Edge-triggered D flip-flops

Master-Slave Flip-flop

Consider the circuit given in Fig. 7(a). It gives the logic diagram for master-slave *D* flip-flop. It has two flip-flops(called master and slave respectively) and an inverter. The slave section is basically the same as the master section except that it is clocked on the inverted clock pulse and is controlled by the outputs of the master section rather than by the external inputs. The D input is sampled to change the output Q only at the negative edge of the clock pulse CP. The output of the NOT gate is 1 when the CP is 0. This enables slave flip-flop and its output Q is equal to the output Y of master flip-flop. The master flip-flop is disabled when CP is at logic 0 level. When the CP goes 1, the data from the external D input is given to the master. As long as the CP stays at the 1 level, the slave is disabled because the output of the inverter is equal to 0. Any input change affects the master output Y. However, it cannot affect the slave output.



(a) (b)

Fig. 7 Master-Slave Flip-flops

Thus, the output of the flip-flop can be changed only during the negative edge transition of the clock. The circuit can be changed to respond to the positive clock edge by connecting the slave flip-flop directly to the clock pulse and the master flip-flop to the complement of the clock pulse.

Fig. 7(b) shows the block diagram of master-slave JK flip-flop. Notice that in master-slave JK flip-flop, the output of the slave is given as a feedback to the input of the master flip-flop.

Direct Inputs

Asynchronous (or direct) inputs in flip-flops change the state of the flip-flop independent of the clock. The *clear* (or direct reset) input clears the flip-flop to 0. Placing a 0 on the Clear input will force the flip-flop into the state Q = 0. If Clear = 1, then this input will have no effect on the output state. The *preset* (or direct set) input sets the flip-flop to 1. Preset = 0 forces the flip-flop into the state Q = 1, while Preset = 1 has no effect. Asynchronous inputs, just like synchronous inputs, can be engineered to be active-high or active-low. If they're active-low, there will be an inverting bubble at that input lead on the block symbol. Fig. 8 shows a *D* flip-flop with active-low asynchronous inputs. When a digital system is powered on, the state of the flip-flops is indeterminate. These inputs are used to bring all flip-flops in the system to an initial state prior to their clocked operation.

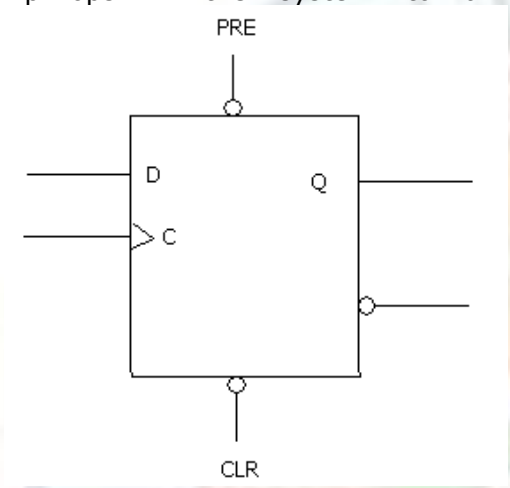


Fig. 8 *D* flip-flop with asynchronous inputs

1-2 COUNTERS

A counter can be defined as a register that goes through a predetermined sequence of values upon the application of clock input. A counter that follows the binary number sequence is called a *binary counter*. An *n*-bit binary counter that cycles through all 2^n states in ascending (or descending) order. It consists of *n* flip-flops and can count in binary from 0 through $2^n - 1$. A counter may also follow any other sequence of states.

A modulo- *N* counter (also known as a divide-by- *N* counter) follows the sequence of *N* states repeatedly.

There are two categories of counters: asynchronous counters and synchronous counters. In an asynchronous counter, flip-flop corresponding to LSB is clocked by external clock pulse and each successive flip-flop is clocked by output of previous flip-flop. In other words, output transition of a flip-flop serves as a source for triggering other flip-flops. Asynchronous counters are also called **ripple counters** because of the way the clock pulse ripples it way through the flip-flops.

In a synchronous counter, all flip-flops receive the common clock. Synchronous counters are faster than asynchronous counter because in synchronous counter all flip flops are clocked simultaneously.

Binary Ripple Counter

A binary ripple counter is an asynchronous counter where only the first flip-flop is clocked by an external clock. Each subsequent flip-flop is triggered by the transition occurring in the preceding flip-flop.

A counter may count up or count down or count up and down depending on the input control. The count sequence usually repeats itself. When counting up, the count sequence goes from 0000, 0001, 0010, ... 1110 , 1111 , 0000, 0001, ... etc. When counting down the count sequence goes in the opposite manner: 1111, 1110, ... 0010, 0001, 0000, 1111, 1110, ... etc.

Going through a count sequence from 0000,0001,0010, and so on, we note that the least significant bit, A_0, toggles once with each count pulse input. A_1 is toggled every time that A_0 changes from 1 to 0. A_2 is toggled every time that A_1 changes from 1 to 0 and so on for any other higher order bits of a ripple counter. A 4-bit binary ripple counter constructed using *T* flip-flops is shown in Fig. 9.

In a count-down counter, LSB is toggled with every count pulse. Any higher-order bit is toggled if the preceding bit has a transition from 0 to 1. In this counter, the complement terminals Q' of all flip-flops give the outputs. Therefore, the diagram of a binary countdown counter looks the same as the binary ripple counter in Fig. 9 , provided that all flip-flops trigger on the positive edge of the clock. (The bubble in the *C* inputs must be removed.)

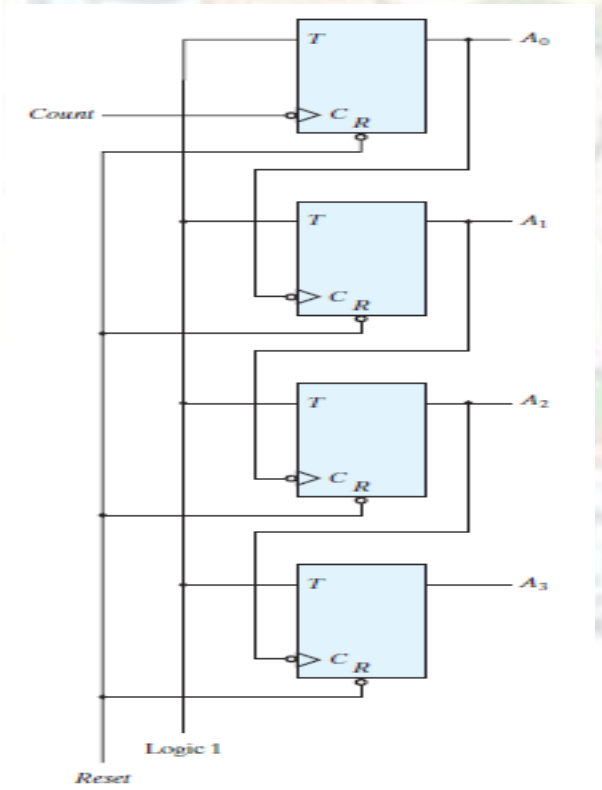


Fig. 9 Four-bit binary Ripple counter

Synchronous Counters

In synchronous counters, all flip-flops are triggered by common clock pulse input. This is different from ripple counter where flip-flops are triggered one at a time in succession.

In this counter, the flip-flop corresponding to LSB is toggled with every clock pulse. A flip-flop in any other position is toggled on the next clock pulse provided all the preceding (lower-order) bits are equal to 1.

Four-bit synchronous binary counter is depicted in Fig. 10. The counter is enabled by *Count_enable*. If the enable input is 0, the clock does not change the state of the counter since all *J* and *K* inputs are equal to 0. If the *Count_enable* is set, the circuit will starting counting from the last count value and repeating itself. It is possible to extend the counter to more stages, such that each stage has a flip-flop and an AND gate. As shown in the figure, positive edge transition of the clock triggers all the flip-flops. Note that the polarity of the clock is important in the ripple counter and not here.

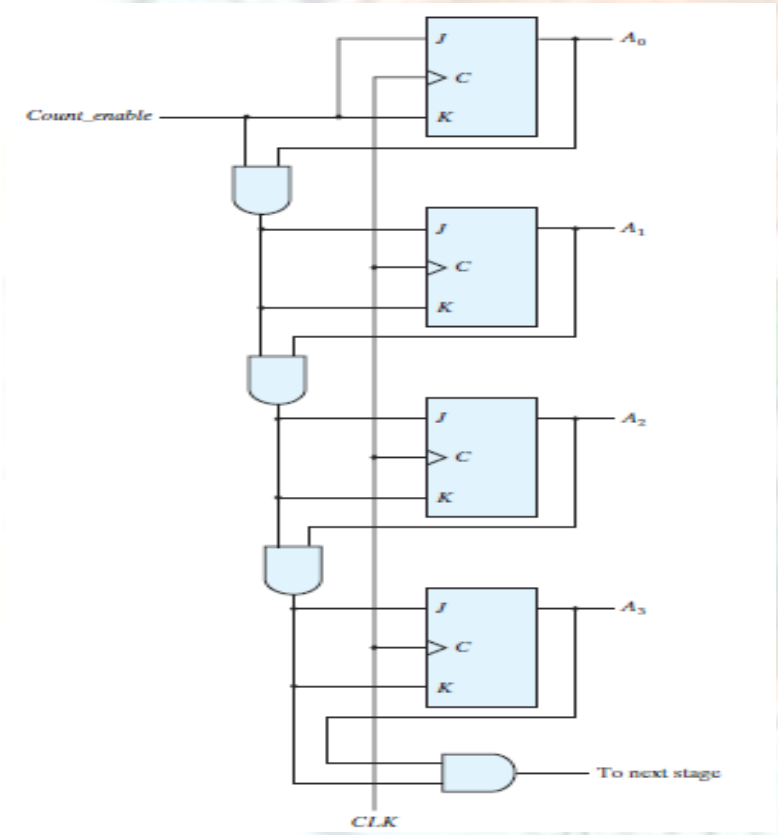


Fig. 10 Four-bit synchronous binary counter

SUMMARY

- Flip-flops are the memory elements used in clocked sequential circuits. A flip-flop is a binary storage device that stores one bit of information.
- There are basically four main types of flip-flops: SR, D, JK and T.
- An edge-triggered flip-flop changes output either at the positive edge or at the negative edge of the clock input.
- Flip-flops respond to clock transitions. Latches are also storage elements. However, they operate with signal elements i.e. respond to pulse duration instead of pulse transition.
- Race condition is when the output keeps toggling again and again in one clock pulse and becomes indeterminate. It occurs when the clock pulse duration is greater than the signal propagation delay from input to output.
- In a master-slave flip-flop, two flip-flops (called master and slave respectively) and an inverter are used. The slave section is basically the same as the master section except that it is clocked on the inverted clock pulse and is controlled by the outputs of the master section rather than by the external inputs.
- Asynchronous (or direct) inputs in flip-flops change the state of the flip-flop independent of the clock. *Clear* and *Preset* are two asynchronous inputs.
- A counter can be defined as a register that goes through a predetermined sequence of values upon the application of clock input.
- Counters are of two types: asynchronous and synchronous.

EXERCISE

1. For what values of *J* and *K* inputs, a J-K flip-flop toggle?

   A. J = 1, K = 1

   B. J = 0, K = 1

   C. J = 1, K = 0

   D. J = 0, K = 0

2. Which of the following describes the operation of a negative edge-triggered D flip-flop?

   A. If both inputs are HIGH, the output will toggle.

   B. The output will follow the input on the trailing edge of the clock.

   C. When both inputs are LOW, an invalid state exists.

   D. None of these

3. On a negative edge-triggered S-R flip-flop, the outputs reflect the input condition when _____.

   A. the clock pulse is LOW

   B. the clock pulse is HIGH

   C. the clock pulse transitions from LOW to HIGH

   D. the clock pulse transitions from HIGH to LOW

4. What is the disadvantage of an S-R flip-flop?

   A. It has no enable input.

   B. It has an invalid state.

   C. It has no clock input.

   D. It has only a single output.

5. A positive edge-triggered D flip-flop will store a 0 when _____.

A. the D input=1 and the clock makes a HIGH to LOW transition

B. the D input=0 and the clock makes a LOW to HIGH transition

C. the D input is HIGH and the clock is LOW

D. the D input is HIGH and the clock is HIGH

6. Asynchronous inputs will cause the flip-flop to respond immediately independent of the clock input.

   A. True B. False

7. A positive edge-triggered flip-flop will accept inputs only when the clock is HIGH.

   A. True B. False

8. A negative edge-triggered flip-flop changes states when the clock input has a LOW to HIGH transition.

   A. True B. False

9. A ripple counter is an asynchronous counter.

   A. True B. False

10. A 4-bit counter has a maximum modulus of _____.

    A. 6 B. 10 C. 3 D. 16

11. How many different states does a 3-bit asynchronous counter have?

    A. 2 B. 4

    C. 8 D. 16

12. The final output of a modulus-8 counter occurs one time for every _____.

    A. 8 clock pulses

    B. 16 clock pulses

    C. 24 clock pulses

    D. 32 clock pulses

13. Explain the working of master-slave flip-flop.
14. Why synchronous counters are faster than asynchronous counters?
15. State the use of asynchronous inputs.
16. How is *D* flip-flop obtained from *SR* flip-flop?
17. What does the symbol > in block diagram of flip-flops denotes?
18. What do you mean by Mod-N counter?
19. What is the difference between flip-flops and latches?
20. What do you understand by race condition?

REFERENCES

Suggested Readings

1. M.M. Mano, Computer system Architecture, 3rd edition, Prentice Hall of India, 2008.
2. M.M. Mano, Digital Design,2nd edition.

Web Links

1. http://highered.mcgraw-hill.com/sites/dl/free/0072823151/56549/vra23151_ch07.pdf
2. http://www.brighthubengineering.com/diy-electronics-devices/46493-types-of-flip-flop-circuits-explained-rs-jk-d-t/