6-2 FLIP-FLOPS

A flip-flop circuit can maintain a binary state indefinitely (as long as power is delivered to the circuit) until directed by an input signal to switch states. The major differences among various types of flip-flops are in the number of inputs they possess and in the manner in which the inputs affect the binary state. The most common types of flip-flops are discussed in what follows.

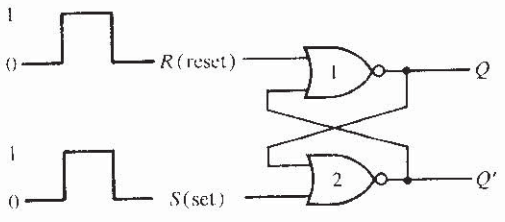
Basic Flip-Flop Circuit

It was mentioned in Sections 4-7 and 4-8 that a flip-flop circuit can be constructed from two NAND gates or two NOR gates. These constructions are shown in the logic diagrams of Figs. 6-2 and 6-3. Each circuit forms a basic flip-flop upon which other more complicated types can be built. The cross-coupled connection from the output of one gate to the input of the other gate constitutes a feedback path. For this reason, the circuits are classified as asynchronous sequential circuits. Each flip-flop has two outputs, Q and Q', and two inputs, *set* and *reset*. This type of flip-flop is sometimes called a *direct-coupled RS* flip-flop, or *SR latch*. The R and S are the first letters of the two input names.

To analyze the operation of the circuit of Fig. 6-2, we must remember that the output of a NOR gate is 0 if any input is 1, and that the output is 1 only when all inputs are 0. As a starting point, assume that the set input is 1 and the reset input is 0. Since gate 2 has an input of 1, its output Q' must be 0, which puts both inputs of gate 1 at 0, so that output Q is 1. When the set input is returned to 0, the outputs remain the same, because output Q remains a 1, leaving one input of gate 2 at 1. That causes output Q' to stay at 0, which leaves both inputs of gate number 1 at 0, so that output Q is a 1. In the same manner, it is possible to show that a 1 in the reset input changes output Q to 0 and Q' to 1. When the reset input returns to 0, the outputs do not change.

When a 1 is applied to both the set and the reset inputs, both Q and Q' outputs go to 0. This condition violates the fact that outputs Q and Q' are the complements of each other. In normal operation, this condition must be avoided by making sure that 1's are not applied to both inputs simultaneously.

A flip-flop has two useful states. When $Q = 1$ and $Q' = 0$, it is in the *set state* (or

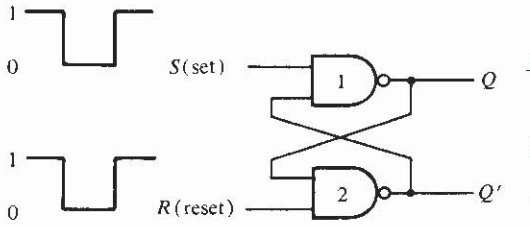


(a) Logic diagram

S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after S = 1, R = 0)
0	1	0	1	
0	0	0	1	(after S = 0, R = 1)
1	1	0	0	

(b) Truth table

FIGURE 6-2
Basic flip-flop circuit with NOR gates



(a) Logic diagram

S	R	Q	Q′	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	

(b) Truth table

FIGURE 6-3
Basic flip-flop circuit with NAND gates

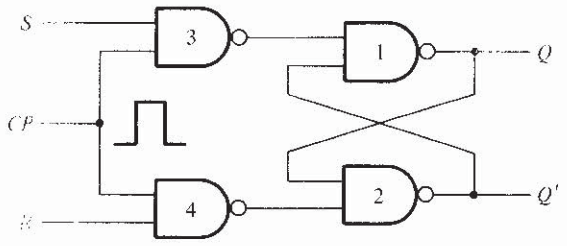
1-state). When $Q = 0$ and $Q' = 1$, it is in the *clear state* (or 0-state). The outputs Q and Q' are complements of each other and are referred to as the normal and complement outputs, respectively. The binary state of the flip-flop is taken to be the value of the normal output.

Under normal operation, both inputs remain at 0 unless the state of the flip-flop has to be changed. The application of a momentary 1 to the set input causes the flip-flop to go to the set state. The set input must go back to 0 before a 1 is applied to the reset input. A momentary 1 applied to the reset input causes the flip-flop to go the clear state. When both inputs are initially 0, a 1 applied to the set input while the flip-flop is in the set state or a 1 applied to the reset input while the flip-flop is in the clear state leaves the outputs unchanged. When a 1 is applied to both the set and the reset inputs, both outputs go to 0. This state is undefined and is usually avoided. If both inputs now go to 0, the state of the flip-flop is indeterminate and depends on which input remains a 1 longer before the transition to 0.

The NAND basic flip-flop circuit of Fig. 6-3 operates with both inputs normally at 1 unless the state of the flip-flop has to be changed. The application of a momentary 0 to the set input causes output Q to go to 1 and Q' to go to 0, thus putting the flip-flop into the set state. After the set input returns to 1, a momentary 0 to the reset input causes a transition to the clear state. When both inputs go to 0, both outputs go to 1—a condition avoided in normal flip-flop operation.

RS Flip-flop

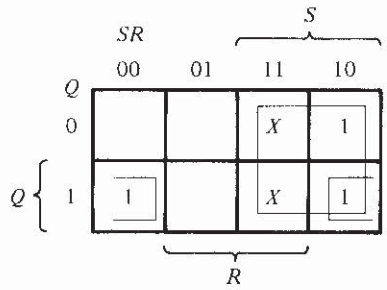
The operation of the basic flip-flop can be modified by providing an additional control input that determines when the state of the circuit is to be changed. An *RS* flip-flop with a clock pulse (*CP*) input is shown in Fig. 6-4(a). It consists of a basic flip-flop circuit and two additional NAND gates. The pulse input acts as an enable signal for the other two inputs. The outputs of NAND gates 3 and 4 stay at the logic 1 level as long as the *CP* input remains at 0. This is the quiescent condition for the basic flip-flop. When the pulse input goes to 1, information from the S or R input is allowed to reach the output. The set state is reached with $S = 1$, $R = 0$, and $CP = 1$. This causes the output of gate 3 to go to 0, the output of gate 4 to remain at 1, and the output of the flip-flop at Q to go to 1. To change to the reset state, the inputs must be $S = 0$, $R = 1$, and $CP = 1$.



(a) Logic diagram

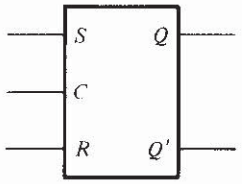
Q	S	R	$Q(t+1)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	Indeterminate
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	Indeterminate

(b) Characteristic table



$Q(t+1) = S + R'Q$
$SR = 0$

(c) Characteristic equation



(d) Graphic symbol

FIGURE 6-4
RS flip-flop

In either case, when CP returns to 0, the circuit remains in its previous state. When $CP = 1$ and both the S and R inputs are equal to 0, the state of the circuit does not change.

An indeterminate condition occurs when $CP = 1$ and both S and R are equal to 1. This condition places 0's in the outputs of gates 3 and 4 and 1's in both outputs Q and Q'. When the CP input goes back to 0 (while S and R are maintained at 1), it is not possible to determine the next state, as it depends on whether the output of gate 3 or gate 4 goes to 1 first. This indeterminate condition makes the circuit of Fig. 6-4(a) difficult to manage and it is seldom used in practice. Nevertheless, it is an important circuit because all other flip-flops are constructed from it.

The characteristic table of the flip-flop is shown in Fig. 6-4(b). This table shows the operation of the flip-flop in tabular form. Q is an abbreviation of $Q(t)$ and stands for the binary state of the flip-flop before the application of a clock pulse, referred to as the *present state*. The S and R columns give the possible values of the inputs, and $Q(t + 1)$ is the state of the flip-flop after the application of a single pulse, referred to as the *next state*. Note that the CP input is not included in the characteristic table. The table must

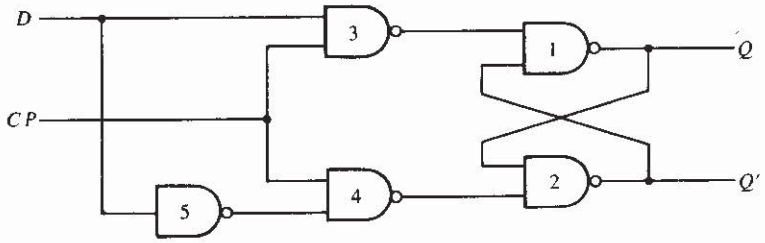
be interpreted as follows: Given the present state Q and the inputs S and R, the application of a single pulse in the CP input causes the flip-flop to go to the next state, $Q(t + 1)$.

The characteristic equation of the flip-flop is derived in the map of Fig. 6-4(c). This equation specifies the value of the next state as a function of the present state and the inputs. The characteristic equation is an algebraic expression for the binary information of the characteristic table. The two indeterminate states are marked with dont't-care X's in the map, since they may result in either 1 or 0. However, the relation $SR = 0$ must be included as part of the characteristic equation to specify that both S and R cannot equal to 1 simultaneously.

The graphic symbol of the RS flip-flop is shown in Fig. 6-4(d). It consists of a rectangular-shape block with inputs S, R, and C. The outputs are Q and Q', where Q' is the complement of Q (except in the indeterminate state).

D Flip-Flop

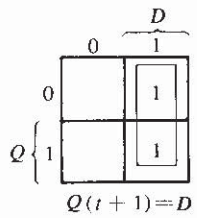
One way to eliminate the undesirable condition of the indeterminate state in the RS flip-flop is to ensure that inputs S and R are never equal to 1 at the same time. This is done in the D flip-flop shown in Fig. 6-5(a). The D flip-flop has only two inputs: D and CP. The D input goes directly to the S input and its complement is applied to the R input. As long as the pulse input is at 0, the outputs of gates 3 and 4 are at the 1 level and the circuit cannot change state regardless of the value of D. The D input is sampled when $CP = 1$. If D is 1, the Q output goes to 1, placing the circuit in the set state. If D is 0, output Q goes to 0 and the circuit switches to the clear state.



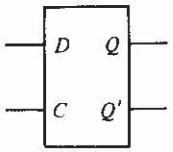
(a) Logic diagram

Q	D	$Q(t+1)$
0	0	0
0	1	1
1	0	0
1	1	1

(b) Characteristic table



(c) Characteristic equation



(d) Graphic symbol

FIGURE 6-5
D flip-flop

The *D* flip-flop receives the designation from its ability to hold *data* into its internal storage. This type of flip-flop is sometimes called a *gated D-latch*. The *CP* input is often given the designation *G* (for *gate*) to indicate that this input enables the gated latch to make possible data entry into the circuit. The binary information present at the data input of the *D* flip-flop is transferred to the *Q* output when the *CP* input is enabled. The output follows the data input as long as the pulse remains in its 1 state. When the pulse goes to 0, the binary information that was present at the data input at the time the pulse transition occurred is retained at the *Q* output until the pulse input is enabled again.

The characteristic table for the D flip-flop is shown in Fig. 6-5(b). It shows that the next state of the flip-flop is independent of the present state since $Q(t + 1)$ is equal to input *D* whether *Q* is equal to 0 or 1. This means that an input pulse will transfer the value of input *D* into the output of the flip-flop independent of the value of the output before the pulse was applied. The characteristic equation shows clearly that $Q(t + 1)$ is equal to *D*.

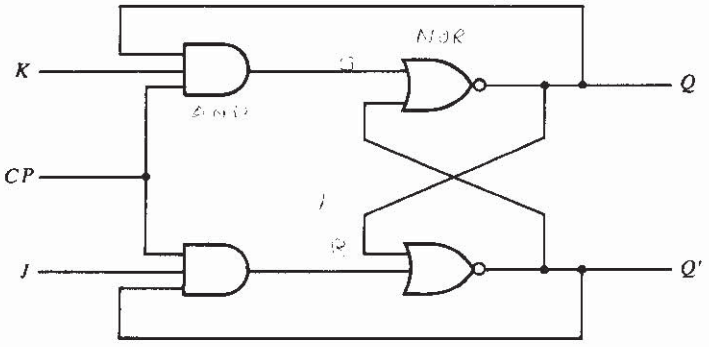
The graphic symbol for the level sensitive *D* flip-flop is shown in Fig. 6-5(d). The graphic symbol for a transition-sensitive *D* flip-flop is shown later in Fig. 6-14.

JK and T Flip-Flops

A *JK* flip-flop is a refinement of the *RS* flip-flop in that the indeterminate state of the *RS* type is defined in the *JK* type. Inputs *J* and *K* behave like inputs *S* and *R* to set and clear the flip-flop, respectively. The input marked *J* is for *set* and the input marked *K* is for *reset*. When both inputs *J* and *K* are equal to 1, the flip-flop switches to its complement state, that is, if $Q = 1$, it switches to $Q = 0$, and vice versa.

A *JK* flip-flop constructed with two cross-coupled NOR gates and two AND gates is shown in Fig. 6-6(a). Output *Q* is ANDed with *K* and *CP* inputs so that the flip-flop is cleared during a clock pulse only if *Q* was previously 1. Similarly, output Q' is ANDed with *J* and *CP* inputs so that the flop-flop is set with a clock pulse only when Q' was previously 1. When both *J* and *K* are 1, the input pulse is transmitted through one AND gate only: the one whose input is connected to the flip-flop output that is presently equal to 1. Thus, if $Q = 1$, the output of the upper AND gate becomes 1 upon application of the clock pulse, and the flip-flop is cleared. If $Q' = 1$, the output of the lower AND gate becomes 1 and the flip-flop is set. In either case, the output state of the flip-flop is complemented. The behavior of the *JK* flip-flop is demonstrated in the characteristic table of Fig. 6-6(b).

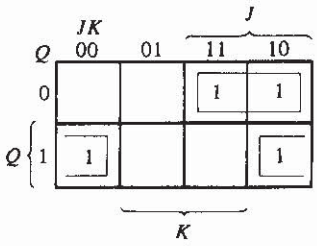
It is very important to realize that because of the feedback connection in the *JK* flip-flop, a *CP* pulse that remains in the 1 state while both *J* and *K* are equal to 1 will cause the output to complement again and repeat complementing until the pulse goes back to 0. To avoid this undesirable operation, the clock pulse must have a time duration that is shorter than the propagation delay time of the flip-flop. This is a restrictive requirement, since the operation of the circuit depends on the width of the pulse. For this reason, *JK* flip-flops are never constructed as shown in Fig. 6-6(a). The restriction on the pulse width can be eliminated with a master–slave or edge-triggered construction, as discussed in the next section. The same reasoning applies to the *T* flip-flop.



(a) Logic diagram

Q	J	K	$Q(t+1)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

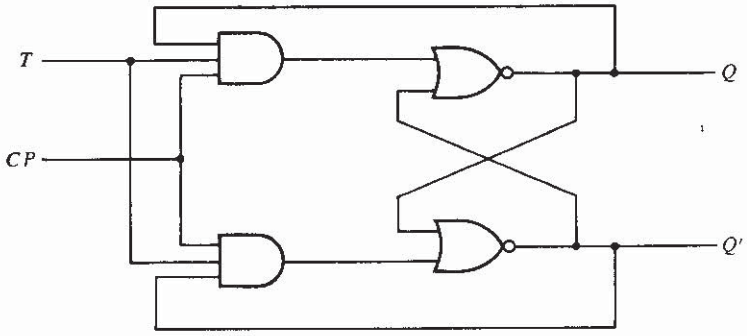
(b) Characteristic table



$Q(t+1) = JQ' + K'Q$

(c) Characteristic equation

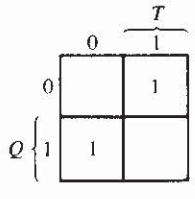
FIGURE 6-6
JK flip-flop



(a) Logic diagram

Q	T	$Q(t+1)$
0	0	0
0	1	1
1	0	1
1	1	0

(b) Characteristic table



$Q(t+1) = TQ' + T'Q$

(c) Characteristic equation

FIGURE 6-7
T flip-flop

The T flip-flop is a single-input version of the JK flip-flop. As shown in Fig. 6-7(a), the T flip-flop is obtained from the JK flip-flop when both inputs are tied together. The designation T comes from the ability of the flip-flop to "toggle," or complement, its state. Regardless of the present state, the flip-flop complements its output when the clock pulse occurs while input T is 1. The characteristic table and characteristic equation show that when $T = 0$, $Q(t + 1) = Q$, that is, the next state is the same as the present state and no change occurs. When $T = 1$, then $Q(t + 1) = Q'$, and the state of the flip-flop is complemented.

6-3 TRIGGERING OF FLIP-FLOPS

The state of a flip-flop is switched by a momentary change in the input signal. This momentary change is called a *trigger* and the transition it causes is said to trigger the flip-flop. Asynchronous flip-flops, such as the basic circuits of Figs. 6-2 and 6-3, require an input trigger defined by a change of signal *level*. This level must be returned to its initial value (0 in the NOR and 1 in the NAND flip-flop) before a second trigger is applied. Clocked flip-flops are triggered by *pulses*. A pulse starts from an initial value of 0, goes momentarily to 1, and after a short time, returns to its initial 0 value. The time interval from the application of the pulse until the output transition occurs is a critical factor that needs further investigation.

As seen from the block diagram of Fig. 6-1, a sequential circuit has a feedback path between the combinational circuit and the memory elements. This path can produce instability if the outputs of memory elements (flip-flops) are changing while the outputs of the combinational circuit that go to flip-flop inputs are being sampled by the clock pulse. This timing problem can be prevented if the outputs of flip-flops do not start changing until the pulse input has returned to 0. To ensure such an operation, a flip-flop must have a signal-propagation delay from input to output in excess of the pulse duration. This delay is usually very difficult to control if the designer depends entirely on the propagation delay of logic gates. One way of ensuring the proper delay is to include within the flip-flop circuit a physical delay unit having a delay equal to or greater than the pulse duration. A better way to solve the feedback timing problem is to make the flip-flop sensitive to the pulse *transition* rather than the pulse duration.

A clock pulse may be either positive or negative. A positive clock source remains at 0 during the interval between pulses and goes to 1 during the occurrence of a pulse. The pulse goes through two signal transitions: from 0 to 1 and the return from 1 to 0. As shown in Fig. 6-8, the positive transition is defined as the *positive edge* and the negative transition as the *negative edge*. This definition applies also to negative pulses.

The clocked flip-flops introduced in Section 6-2 are triggered during the positive edge of the pulse, and the state transition starts as soon as the pulse reaches the logic-1 level. The new state of the flip-flop may appear at the output terminals while the input pulse is still 1. If the other inputs of the flip-flop change while the clock is still 1, the

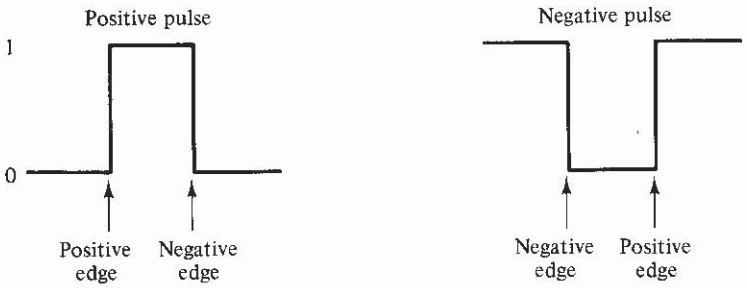


FIGURE 6-8
Definition of clock-pulse transition

flip-flop will start responding to these new values and a new output state may occur. When this happens, the output of one flip-flop cannot be applied to the inputs of another flip-flop when both are triggered by the same clock pulse. However, if we can make the flip-flop respond to the positive- (or negative-) edge transition *only*, instead of the entire pulse duration, then the multiple-transition problem can be eliminated.

One way to make the flip-flop respond only to a pulse transition is to use capacitive coupling. In this configuration, an *RC* (resistor–capacitor) circuit is inserted in the clock input of the flip-flop. This circuit generates a spike in response to a momentary change of input signal. A positive edge emerges from such a circuit with a positive spike, and a negative edge emerges with a negative spike. Edge triggering is achieved by designing the flip-flop to neglect one spike and trigger on the occurrence of the other spike. Another way to achieve edge triggering is to use a master–slave or edge-triggered flip-flop as discussed in what follows.

Master–Slave Flip-Flop

A master–slave flip-flop is constructed from two separate flip-flops. One circuit serves as a master and the other as a slave, and the overall circuit is referred to as a *master–slave flip-flop*. The logic diagram of an *RS* master–slave flip-flop is shown in Fig. 6-9. It consists of a master flip-flop, a slave flip-flop, and an inverter. When clock pulse *CP* is 0, the output of the inverter is 1. Since the clock input of the slave is 1, the flip-flop is enabled and output Q is equal to Y, while Q' is equal to Y'. The master flip-flop is disabled because $CP = 0$. When the pulse becomes 1, the information then at the external R and S inputs is transmitted to the master flip-flop. The slave flip-flop, however, is isolated as long as the pulse is at its 1 level, because the output of the inverter is 0. When the pulse returns to 0, the master flip-flop is isolated, which prevents the external inputs from affecting it. The slave flip-flop then goes to the same state as the master flip-flop.

The timing relationships shown in Fig. 6-10 illustrate the sequence of events that occur in a master–slave flip-flop. Assume that the flip-flop is in the clear state prior to the

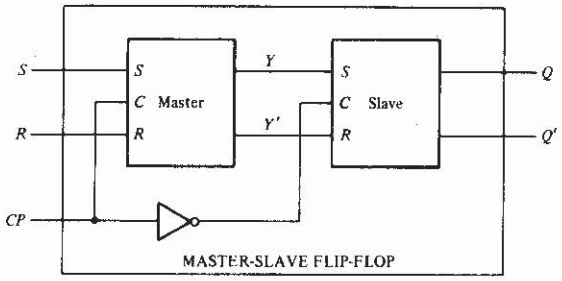


FIGURE 6-9
Logic diagram of a master–slave flip-flop

occurrence of a pulse, so that $Y = 0$ and $Q = 0$. The input conditions are $S = 1$, $R = 0$, and the next clock pulse should change the flip-flop to the set state with $Q = 1$. During the pulse transition from 0 to 1, the master flip-flop is set and changes Y to 1. The slave flip-flop is not affected because its CP input is 0. Since the master flip-flop is an internal circuit, its change of state is not noticeable in the outputs Q and Q'. When the pulse returns to 0, the information from the master is allowed to pass through to the slave, making the external output $Q = 1$. Note that the external S input can be changed at the same time that the pulse goes through its negative-edge transition. This is because, once the CP reaches 0, the master is disabled and its R and S inputs have no influence until the next clock pulse occurs. Thus, in a master–slave flip-flop, it is possible to switch the output of the flip-flop and its input information with the same clock pulse. It must be realized that the S input could come from the output of another master–slave flip-flop that was switched with the same clock pulse.

The behavior of the master–slave flip-flop just described dictates that the state changes in all flip-flops coincide with the negative-edge transition of the pulse. However, some IC master–slave flip-flops change output states in the positive-edge transi-

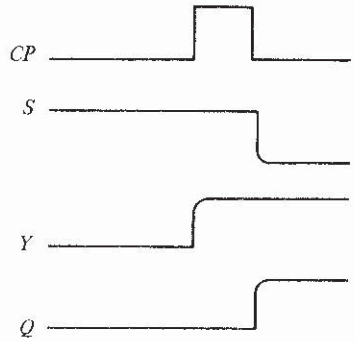


FIGURE 6-10
Timing relationships in a master–slave flip-flop

tion of clock pulses. This happens in flip-flops that have an additional inverter between the *CP* terminal and the input of the master. Such flip-flops are triggered with negative pulses (see Fig. 6-8), so that the negative edge of the pulse affects the master and the positive edge affects the slave and the output terminals.

The master–slave combination can be constructed for any type of flip-flop by adding a clocked *RS* flip-flop with an inverted clock to form the slave. An example of a master–slave *JK* flip-flop constructed with NAND gates is shown in Fig. 6-11. It consists of two flip-flops; gates 1 through 4 form the master flip-flop, and gates 5 through 8 form the slave flip-flop. The information present at the *J* and *K* inputs is transmitted to the master flip-flop on the positive edge of a clock pulse and is held there until the negative edge of the clock pulse occurs, after which it is allowed to pass through to the slave flip-flop. The clock input is normally 0, which keeps the outputs of gates 1 and 2 at the 1 level. This prevents the *J* and *K* inputs from affecting the master flip-flop. The slave flip-flop is a clocked *RS* type, with the master flip-flop supplying the inputs and the clock input being inverted by gate 9. When the clock is 0, the output of gate 9 is 1, so that output Q is equal to Y, and Q' is equal to Y'. When the positive edge of a clock pulse occurs, the master flip-flop is affected and may switch states. The slave flip-flop is isolated as long as the clock is at the 1 level, because the output of gate 9 provides a 1 to both inputs of the NAND basic flip-flop of gates 7 and 8. When the clock input returns to 0, the master flip-flop is isolated from the *J* and *K* inputs and the slave flip-flop goes to the same state as the master flip-flop.

Now consider a digital system containing many master–slave flip-flops, with the outputs of some flip-flops going to the inputs of other flip-flops. Assume that clock-pulse inputs to all flip-flops are synchronized (occur at the same time). At the beginning of each clock pulse, some of the master elements change state, but all flip-flop outputs remain at their previous values. After the clock pulse returns to 0, some of the outputs

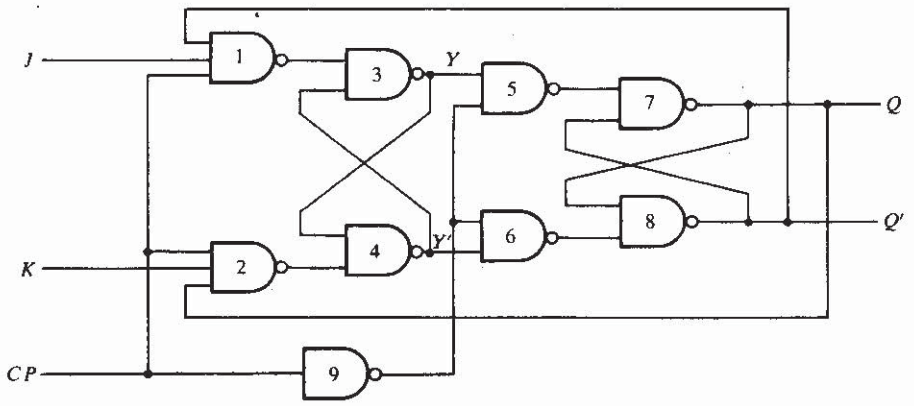


FIGURE 6-11
Clocked master–slave *JK* flip-flop

change state, but none of these new states have an effect on any of the master elements until the next clock pulse. Thus, the states of flip-flops in the system can be changed simultaneously during the same clock pulse, even though outputs of flip-flops are connected to inputs of flip-flops. This is possible because the new state appears at the output terminals only after the clock pulse has returned to 0. Therefore, the binary content of one flip-flop can be transferred to a second flip-flop and the content of the second transferred to the first, and both transfers can occur during the same clock pulse.

Edge-Triggered Flip-Flop

Another type of flip-flop that synchronizes the state changes during a clock-pulse transition is the *edge-triggered* flip-flop. In this type of flip-flop, output transitions occur at a specific level of the clock pulse. When the pulse input level exceeds this threshold level, the inputs are locked out and the flip-flop is therefore unresponsive to further changes in inputs until the clock pulse returns to 0 and another pulse occurs. Some edge-triggered flip-flops cause a transition on the positive edge of the pulse, and others cause a transition on the negative edge of the pulse.

The logic diagram of a D-type positive-edge-triggered flip-flop is shown in Fig. 6-12. It consists of three basic flip-flops of the type shown in Fig. 6-3. NAND gates 1 and 2 make up one basic flip-flop and gates 3 and 4 another. The third basic flip-flop comprising gates 5 and 6 provides the outputs to the circuit. Inputs S and R of the third basic flip-flop must be maintained at logic-1 for the outputs to remain in their steady-state values. When $S = 0$ and $R = 1$, the output goes to the set state with $Q = 1$. When $S = 1$ and $R = 0$, the output goes to the clear state with $Q = 0$. Inputs S and R

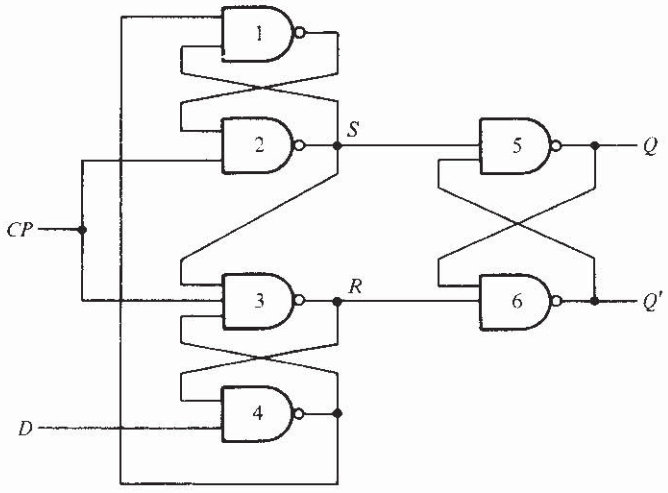


FIGURE 6-12
D-type positive-edge-triggered flip-flop