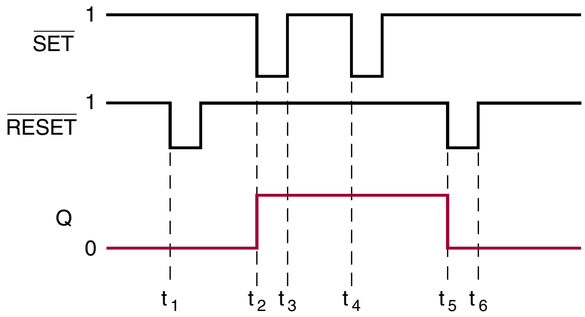
EXAMPLE 5-1

The waveforms of Figure 5-8 are applied to the inputs of the latch of Figure 5-7. Assume that initially $Q = 0$, and determine the Q waveform.

FIGURE 5-8 Example 5-1.



Solution

Initially, $\overline{\text{SET}} = \overline{\text{RESET}} = 1$ so that Q will remain in the 0 state. The LOW pulse that occurs on the $\overline{\text{RESET}}$ input at time t_1 will have no effect because Q is already in the cleared (0) state.

The only way that Q can go to the 1 state is by a LOW pulse on the $\overline{\text{SET}}$ input. This occurs at time t_2 when $\overline{\text{SET}}$ first goes LOW. When $\overline{\text{SET}}$ returns HIGH at t_3, Q will remain in its new HIGH state.

At time t_4 when $\overline{\text{SET}}$ goes LOW again, there will be no effect on Q because Q is already set to the 1 state.

The only way to bring Q back to the 0 state is by a LOW pulse on the $\overline{\text{RESET}}$ input. This occurs at time t_5. When $\overline{\text{RESET}}$ returns to 1 at time t_6, Q remains in the LOW state.

Example 5-1 shows that the latch output "remembers" the last input that was activated and will not change states until the opposite input is activated.

EXAMPLE 5-2

It is almost impossible to obtain a "clean" voltage transition from a mechanical switch because of the phenomenon of **contact bounce.** This is illustrated in Figure 5-9(a), where the action of moving the switch from contact position 1 to 2 produces several output voltage transitions as the switch bounces (makes and breaks contact with contact 2 several times) before coming to rest on contact 2.

The multiple transitions on the output signal generally last no longer than a few milliseconds, but they would be unacceptable in many applications. A NAND latch can be used to prevent the presence of contact bounce from affecting the output. Describe the operation of the "switch debouncing" circuit in Figure 5-9(b).

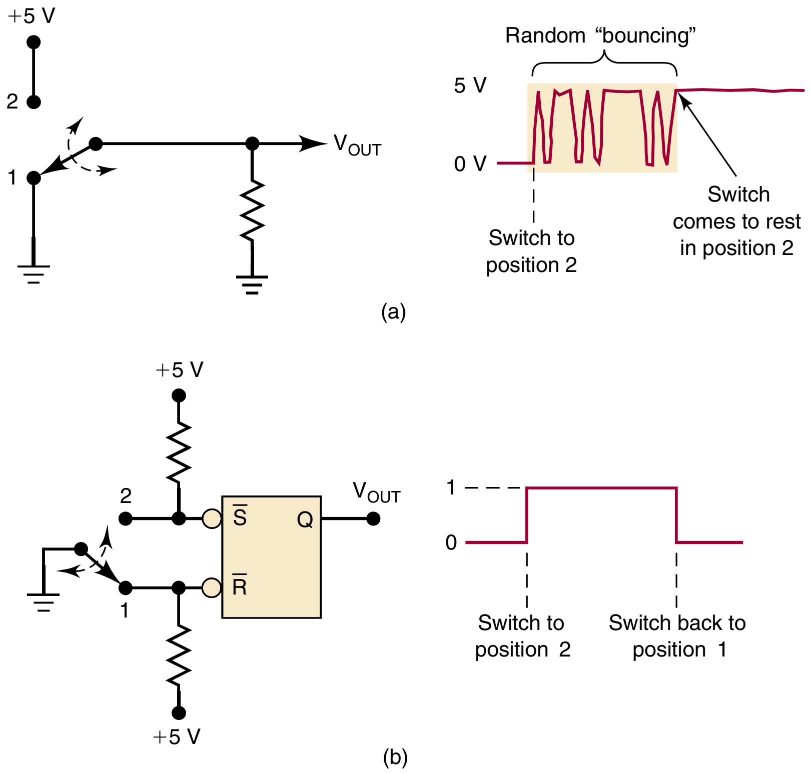


FIGURE 5-9 (a) Mechanical contact bounce will produce multiple transitions; (b) NAND latch used to debounce a mechanical switch.

Solution

Assume that the switch is resting in position 1 so that the $\overline{\text{RESET}}$ input is LOW and $Q = 0$. When the switch is moved to position 2, $\overline{\text{RESET}}$ will go HIGH, and a LOW will appear on the $\overline{\text{SET}}$ input as the switch first makes contact. This will set $Q = 1$ within a matter of a few nanoseconds (the response time of the NAND gate). Now if the switch bounces off contact 2, $\overline{\text{SET}}$ and $\overline{\text{RESET}}$ will both be HIGH, and Q will not be affected; it will stay HIGH. Thus, nothing will happen at Q as the switch bounces on and off contact 2 before finally coming to rest in position 2.

Likewise, when the switch is moved from position 2 back to position 1, it will place a LOW on the $\overline{\text{RESET}}$ input as it first makes contact. This clears Q to the LOW state, where it will remain even if the switch bounces on and off contact 1 several times before coming to rest.

Thus, the output at Q will consist of a single transition each time the switch is moved from one position to the other.

OUTCOME ASSESSMENT QUESTIONS

1. What is the normal resting state of the $\overline{\text{SET}}$ and $\overline{\text{RESET}}$ inputs? What is the active state of each input?
2. What will be the states of Q and $\overline{Q}$ after a FF has been reset (cleared)?
3. *True or false:* The $\overline{\text{SET}}$ input can never be used to make $Q = 0$.
4. When power is first applied to any FF circuit, it is impossible to predict the initial states of Q and $\overline{Q}$. What can be done to ensure that a NAND latch always starts off in the $Q = 1$ state?

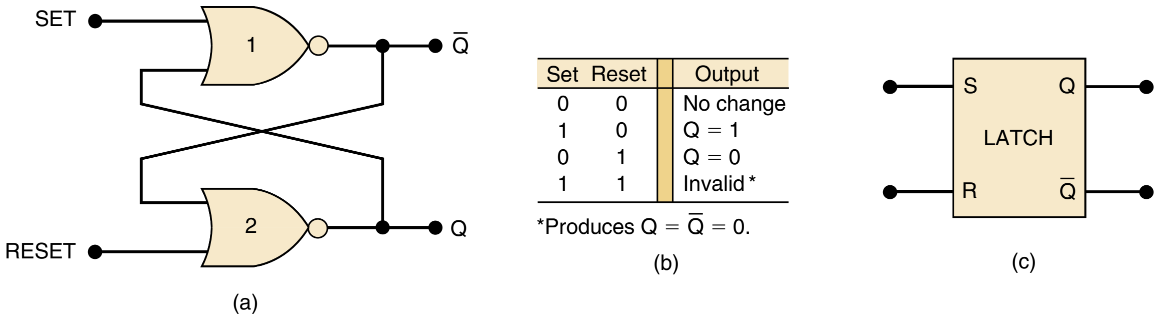
5-2 NOR GATE LATCH

OUTCOMES

Upon completion of this section, you will be able to:

- Predict the output state given any changes to the inputs of a NOR latch.
- Distinguish between active-HIGH and active-LOW control inputs.

Two cross-coupled NOR gates can be used as a **NOR gate latch.** The arrangement, shown in Figure 5-10(a), is similar to the NAND latch except that the Q and $\overline{Q}$ outputs have reversed positions.



Set	Reset	Output
0	0	No change
1	0	Q = 1
0	1	Q = 0
1	1	Invalid*

*Produces Q = $\overline{Q}$ = 0.

(b)

FIGURE 5-10 (a) NOR gate latch; (b) function table; (c) simplified block symbol.

The analysis of the operation of the NOR latch can be performed in exactly the same manner as for the NAND latch. The results are given in the function table in Figure 5-10(b) and are summarized as follows:

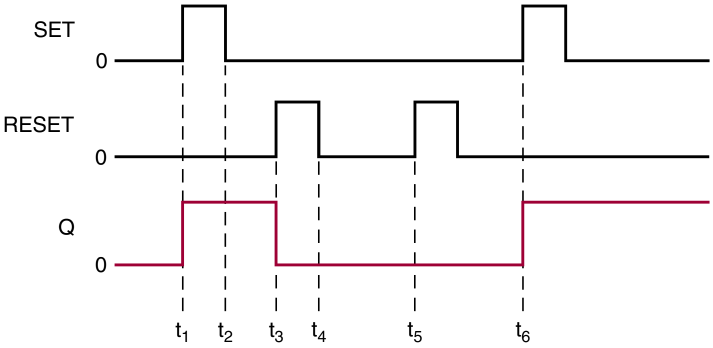
1. SET = RESET = 0. This is the normal resting state for the NOR latch, and it has no effect on the output state. Q and $\overline{Q}$ will remain in whatever state they were in prior to the occurrence of this input condition.
2. SET = 1, RESET = 0. This will always set $Q = 1$, where it will remain even after SET returns to 0.
3. SET = 0, RESET = 1. This will always clear $Q = 0$, where it will remain even after RESET returns to 0.
4. SET = 1, RESET = 1. This condition tries to set and reset the latch at the same time, and it produces $Q = \overline{Q} = 0$. If the inputs are returned to 0 simultaneously, the resulting output state is unpredictable. This input condition should not be used.

The NOR gate latch operates exactly like the NAND latch except that the SET and RESET inputs are active-HIGH rather than active-LOW, and the normal resting state is SET = RESET = 0. Q will be set HIGH by a HIGH pulse on the SET input, and it will be cleared LOW by a HIGH pulse on the RESET input. The simplified block symbol for the NOR latch in Figure 5-10(c) is shown with no bubbles on the S and R inputs; this indicates that these inputs are active-HIGH.

EXAMPLE 5-3

Assume that $Q = 0$ initially, and determine the Q waveform for the NOR latch inputs of Figure 5-11.

FIGURE 5-11 Example 5-3.



Solution

Initially, SET = RESET = 0, which has no effect on Q, and Q stays LOW. When SET goes HIGH at time t_1, Q will be set to 1 and will remain there even after SET returns to 0 at t_2.

At t_3 the RESET input goes HIGH and clears Q to the 0 state, where it remains even after RESET returns LOW at t_4.

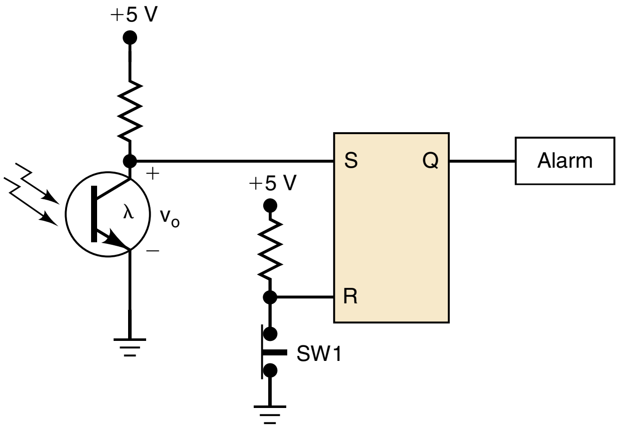
The RESET pulse at t_5 has no effect on Q because Q is already LOW. The SET pulse at t_6 again sets Q back to 1, where it will stay.

Example 5-3 shows that the latch "remembers" the last input that was activated, and it will not change states until the opposite input is activated.

EXAMPLE 5-4

Figure 5-12 shows a simple circuit that can be used to detect the interruption of a light beam. The light is focused on a phototransistor that is connected in the common-emitter configuration to operate as a switch. Assume that the latch has previously been cleared to the 0 state by momentarily opening switch SW1, and describe what happens if the light beam is momentarily interrupted.

FIGURE 5-12 Example 5-4.



Solution

With light on the phototransistor, we can assume that it is fully conducting so that the resistance between the collector and the emitter is very small. Thus, v_0 will be close to 0 V. This places a LOW on the SET input of the latch so that SET = RESET = 0.

When the light beam is interrupted, the phototransistor turns off, and its collector-emitter resistance becomes very high (i.e., essentially an open circuit). This causes v_0 to rise to approximately 5 V; this activates the SET input, which sets Q HIGH and turns on the alarm.

Q will remain HIGH and the alarm will remain on even if v_0 returns to 0 V (i.e., the light beam was interrupted only momentarily) because SET and RESET will both be LOW, which will produce no change in Q.

In this application, the latch's memory characteristic is used to convert a momentary occurrence (beam interruption) into a constant output. The alarm will be deactivated again when the latch is reset by momentarily opening SW1 and allowing the RESET input to be pulled HIGH with the resistor. Note that if we attempt to reset the latch while the light beam is interrupted, it will produce the invalid latch input condition of SET = RESET = 1. It will be necessary to hold SW1 open until the light beam is restored to reset the alarm latch.

Flip-Flop State on Power-Up

When power is applied to a circuit, it is not possible to predict the starting state of a flip-flop's output if its SET and RESET inputs are in their inactive state (e.g., $S = R = 1$ for a NAND latch, $S = R = 0$ for a NOR latch). There is just as much chance that the starting state will be $Q = 0$ as $Q = 1$. It will depend on factors such as internal propagation delays, parasitic capacitance, and external loading. If a latch or FF must start off in a particular state to ensure the proper operation of a circuit, then it must be placed in that state by momentarily activating the SET or RESET input at the start of the circuit's operation. This is often achieved by application of a pulse to the appropriate input.

OUTCOME ASSESSMENT QUESTIONS

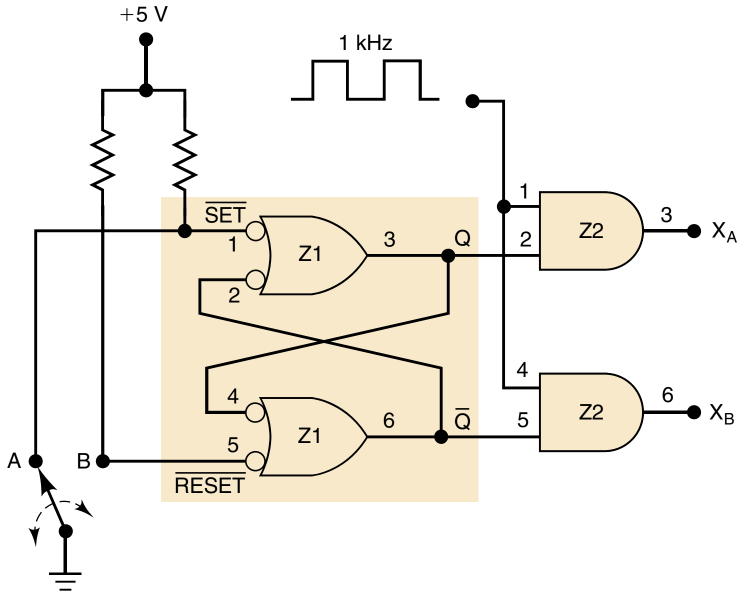
1. What is the normal resting state of the NOR latch inputs? What is the active state?
2. When a latch is set, what are the states of Q and $\overline{Q}$?
3. What is the only way to cause the Q output of a NOR latch to change from 1 to 0?
4. If the NOR latch in Figure 5-12 were replaced by a NAND latch, why wouldn't the circuit work properly?

5-3 TROUBLESHOOTING CASE STUDY

The following two examples present an illustration of the kinds of reasoning used in troubleshooting a circuit containing a latch.

EXAMPLE 5-5

Analyze and describe the operation of the circuit in Figure 5-13.



Switch position	X_A	X_B
A	Pulses	LOW
B	LOW	Pulses

FIGURE 5-13 Examples 5-5 and 5-6.

Solution

The switch is used to set or clear the NAND latch to produce clean, bounce-free signals at Q and $\overline{Q}$. These latch outputs control the passage of the 1-kHz pulse signal through to the AND outputs X_A and X_B.

When the switch moves to position A, the latch is set to $Q = 1$. This enables the 1-kHz pulses to pass through to X_A, while the LOW at $\overline{Q}$ keeps $X_B = 0$. When the switch moves to position B, the latch is cleared to $Q = 0$, which keeps $X_A = 0$, while the HIGH at $\overline{Q}$ enables the pulses to pass through to X_B.

EXAMPLE 5-6

A technician tests the circuit of Figure 5-13 and records the observations shown in Table 5-1. He notices that when the switch is in position B, the circuit functions correctly, but in position A the latch does not set to the $Q = 1$ state. What are the possible faults that could produce this malfunction?

TABLE 5-1

Switch Position	$\overline{SET}$ (Z1-1)	$\overline{RESET}$ (Z1-5)	Q (Z1-3)	$\overline{Q}$ (Z1-6)	X_A (Z2-3)	X_B (Z2-6)
A	LOW	HIGH	LOW	HIGH	LOW	Pulses
B	HIGH	LOW	LOW	HIGH	LOW	Pulses

Solution

There are several possibilities:

1. An internal open connection at Z1-1, which would prevent Q from responding to the $\overline{\text{SET}}$ input.
2. An internal component failure in NAND gate Z1 that prevents it from responding properly.
3. The Q output is stuck LOW, which could be caused by:

   (a) Z1-3 internally shorted to ground

   (b) Z1-4 internally shorted to ground

   (c) Z2-2 internally shorted to ground

   (d) The Q node externally shorted to ground

   An ohmmeter check from Q to ground will determine if any of these conditions are present. A visual check should reveal any external short.

What about $\overline{Q}$ internally or externally shorted to V_{CC}? A little thought will lead to the conclusion that this could not be the fault. If $\overline{Q}$ were shorted to V_{CC}, this would not prevent the Q output from going HIGH when $\overline{\text{SET}}$ goes LOW. Because Q *does not* go HIGH, this cannot be the fault. The reason that $\overline{Q}$ looks as if it is stuck HIGH is that Q is stuck LOW, and that keeps $\overline{Q}$ HIGH through the bottom NAND gate.

5-4 DIGITAL PULSES

OUTCOMES

Upon completion of this section, you will be able to:

- Identify the active state of a signal.
- Classify as positive or negative pulse.
- Define terms common to pulse waveforms (positive/negative pulses, leading/trailing edge, rise time, fall time, etc.).
- Measure pulse characteristics on a timing diagram or waveform.

As you can see from our discussion of S-R latches, there are situations in digital systems when a signal switches from a normal inactive state to the opposite (active) state, thus causing something to happen in the circuit. Then the signal returns to its inactive state while the effect of the recently activated signal remains in the system. These signals are called **pulses,** and it is very important to understand the terminology associated with pulses and pulse waveforms. A pulse that performs its intended function when it goes HIGH is called a *positive* pulse, and a pulse that performs its intended function when it goes LOW is called a *negative* pulse. In actual circuits it takes time for a pulse waveform to change from one level to the other. These transition times are called the rise time (t_r) and the fall time (t_f) and are defined as the time it takes the voltage to change between 10 and 90% of the HIGH level voltage as shown on the positive pulse in Figure 5-14(a). The transition at the beginning of the pulse is called the leading edge and the transition at the end of the pulse is the trailing edge. The duration (width) of the pulse (t_w) is defined as the time between the points when the leading and trailing edges are at 50% of the HIGH level voltage. Figure 5-14(b) shows an active-LOW or negative pulse.

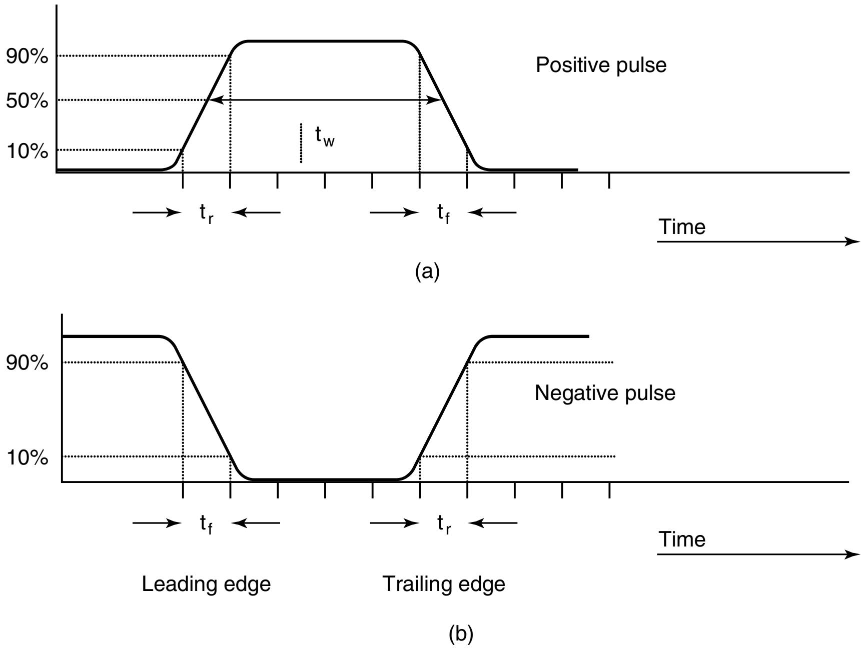


FIGURE 5-14 (a) A positive pulse and (b) a negative pulse.

EXAMPLE 5-7

When a microcontroller wants to access data in its external memory, it activates an active-LOW output pin called $\overline{\text{RD}}$ (read). The data book says that the $\overline{\text{RD}}$ pulse typically has a pulse width t_w of 50 ns, a rise time t_r of 15 ns, and a fall time t_f of 10 ns. Draw a scaled drawing of the $\overline{\text{RD}}$ pulse.

Solution

Figure 5-15 shows the drawing of the pulse. The $\overline{\text{RD}}$ pulse is active-LOW, so the leading edge is a falling edge measured by t_f and the trailing edge is the rising edge measured by t_r.

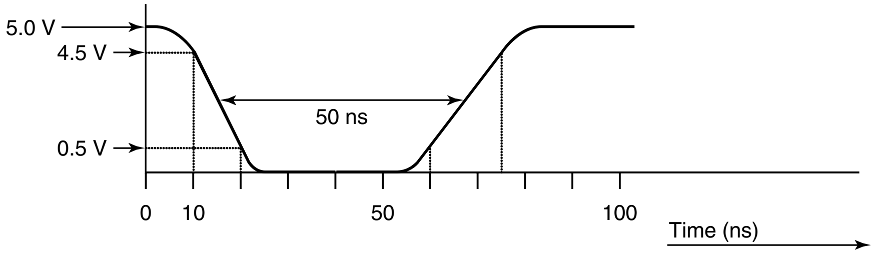


FIGURE 5-15 Example 5-7.

OUTCOME ASSESSMENT QUESTIONS

1. Define the following: rise time, fall time, rising edge, falling edge, leading edge, trailing edge, positive pulse, negative pulse, pulse width.
2. Where is pulse width measured?
3. Where is rise time measured?
4. Where is fall time measured?

5-5 CLOCK SIGNALS AND CLOCKED FLIP-FLOPS

OUTCOMES

Upon completion of this section, you will be able to:

- Differentiate between synchronous and asynchronous inputs for a FF.
- Define edge triggering.
- Define setup time.
- Define hold time.
- Define metastability in a FF.

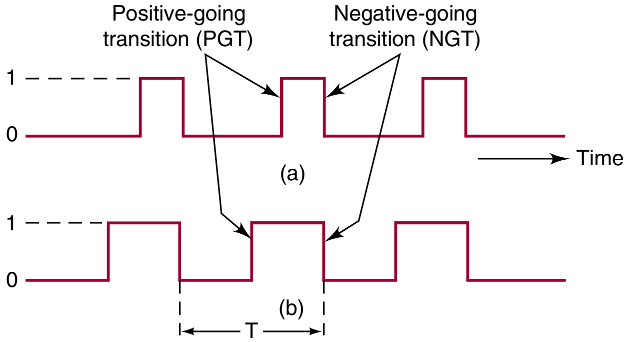
Digital systems can operate either *asynchronously* or *synchronously*. In asynchronous systems, the outputs of logic circuits can change state any time one or more of the inputs change. An asynchronous system is generally more difficult to design and troubleshoot than a synchronous system.

In synchronous systems, the exact times at which any output can change states are determined by a signal commonly called the **clock.** This clock signal is generally a rectangular pulse train or a square wave, as shown in Figure 5-16. The clock signal is distributed to all parts of the system, and most (if not all) of the system outputs can change state only when the clock makes a transition. The transitions (also called *edges*) are pointed out in Figure 5-16. When the clock changes from a 0 to a 1, this is called the **positive-going transition (PGT);** when the clock goes from 1 to 0, this is the **negative-going transition (NGT).** We will use the abbreviations PGT and NGT because these terms appear so often throughout the text.

Most digital systems are principally synchronous (although there are always some asynchronous parts) because synchronous circuits are easier to design and troubleshoot. They are easier to troubleshoot because the circuit outputs can change only at specific instants of time. In other words, almost everything is synchronized to the clock-signal transitions.

The synchronizing action of the clock signals is accomplished through the use of **clocked flip-flops** that are designed to change states on one or the other of the clock's transitions.

FIGURE 5-16 Clock signals.


Positive-going transition (PGT)
Negative-going transition (NGT)
1
0
(a)
Time
1
0
(b)
T

The speed at which a synchronous digital system operates is dependent on how often the clock cycles occur. A clock cycle is measured from one PGT to the next PGT or from one NGT to the next NGT. The time it takes to complete one cycle (seconds/cycle) is called the **period (*T*),** as shown in Figure 5-16(b). The speed of a digital system is normally referred to by the number of clock cycles that happen in 1 s (cycles/second), which is known as the **frequency (*f*)** of the clock. The standard unit for frequency is hertz. One hertz (1 Hz) = 1 cycle/second.

Clocked Flip-Flops

Several types of clocked FFs are used in a wide range of applications. Before we begin our study of the different clocked FFs, we will describe the principal ideas that are common to all of them.

1. Clocked FFs have a clock input that is typically labeled *CLK*, *CK*, or *CP* (clock pulse). We will normally use *CLK*, as shown in Figure 5-17. In most clocked FFs, the *CLK* input is **edge-triggered,** which means that it is activated by a signal transition; this is indicated by the presence of a small triangle on the *CLK* input. This contrasts with the latches, which are level-triggered.

   Figure 5-17(a) is a FF with a small triangle on its *CLK* input to indicate that this input is activated *only* when a positive-going transition occurs; no other part of the input pulse will have an effect on the *CLK* input. In Figure 5-17(b), the FF symbol has a bubble as well as a triangle on its *CLK* input. This signifies that the *CLK* input is activated *only* when a negative-going transition occurs; no other part of the input pulse will have an effect on the *CLK* input.

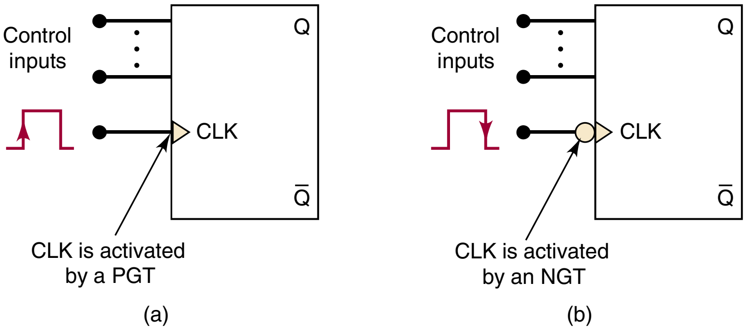


FIGURE 5-17 Clocked FFs have a clock input (*CLK*) that is active on either (a) the PGT or (b) the NGT. The control inputs determine the effect of the active clock transition.

2. Clocked FFs also have one or more **control inputs** that can have various names, depending on their operation. The control inputs will have no effect on *Q* until the active clock transition occurs. In other words, their effect is synchronized with the signal applied to *CLK*. For this reason they are called **synchronous control inputs.**

   For example, the control inputs of the FF in Figure 5-17(a) will have no effect on *Q* until the PGT of the clock signal occurs. Likewise, the control inputs in Figure 5-17(b) will have no effect until the NGT of the clock signal occurs.

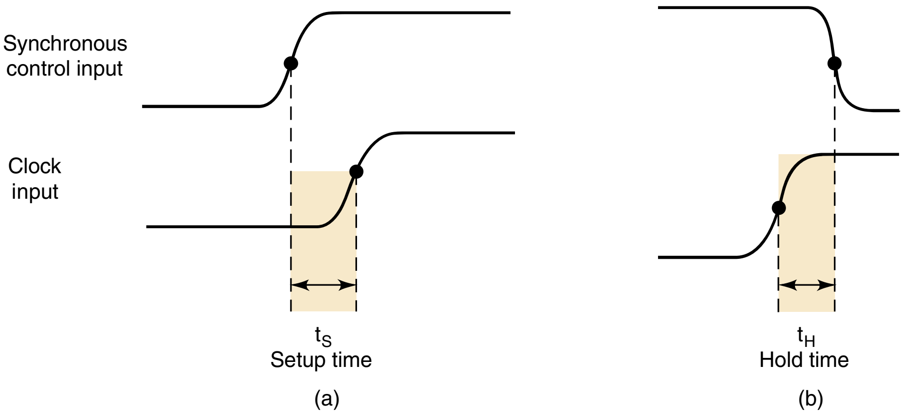
3. In summary, we can say that the control inputs get the FF outputs ready to change, while the active transition at the *CLK* input actually *triggers* the change. The control inputs control the WHAT (i.e., what state the output will go to); the *CLK* input determines the WHEN.

Setup and Hold Times

A flip-flop that triggers reliably will respond to the active clock edge by reading the input and, after a predictable propagation delay, updating its output appropriately. This section investigates some timing requirements that must be met if the flip-flop is expected to trigger reliably. Unreliable triggering could mean that the output settles in the wrong state. For example, assume a flip-flop is initially reset. If we change the data on the control inputs too closely to the clock edge, we may think we commanded the flip-flop to set but it may remain reset. Another possible response is that the flip-flop will perform in an unacceptable way before settling to either HIGH or LOW. In other words, there can be some short-term abnormal voltages present on the output that are referred to as ***metastable states.*** The output may begin to change from its original to the opposite state but then returns to its original state. A third possibility is that it may start to change, hesitate in the invalid region, and then proceed to settle in the opposite state. In any case, metastability can confuse the other logic circuits and cause the system to respond improperly.

Two timing requirements must be met if a clocked FF is to respond reliably to its control inputs when the active *CLK* transition occurs. These requirements are illustrated in Figure 5-18 for a FF that triggers on a PGT.

FIGURE 5-18 Control inputs must be held stable for (a) a time t_S prior to active clock transition and for (b) a time t_H after the active clock transition.

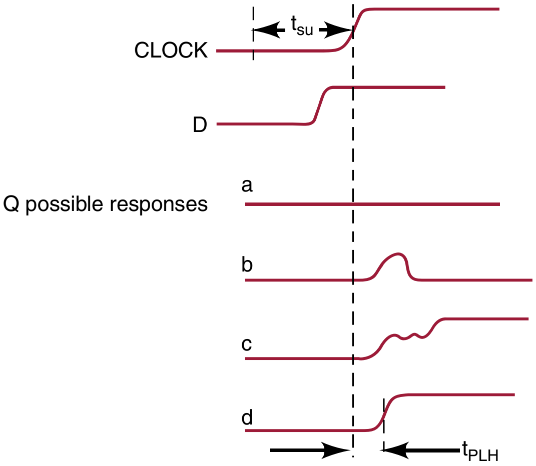


The **setup time, t_S,** is the time interval immediately preceding the active transition of the *CLK* signal during which the control input must be maintained at the proper level. IC manufacturers usually specify the minimum allowable setup time t_S(min). If this time requirement is not met, the FF may not respond reliably when the clock edge occurs.

The **hold time, t_H,** is the time interval immediately following the active transition of the *CLK* signal during which the synchronous control input must be maintained at the proper level. IC manufacturers usually specify the minimum acceptable value of hold time t_H(min). If this requirement is not met, the FF will not trigger reliably.

Figure 5-19 shows the four possible results that may occur for any given flip-flop when its published setup or hold times are violated. Possibility (a)

FIGURE 5-19 (a) Q does not change; (b) meta-stable response, Q settles LOW; (c) metastable response, Q settles HIGH; (d) Q changes as intended.



is that the output does not respond at all to the HIGH input. Possibility (b) is that the output tries to go HIGH but falls back LOW. Possibility (c) is that the output begins to go HIGH, waivers indecisively in the invalid region, and then responds appropriately by settling HIGH. Possibility (d) is that the flip-flop may respond as intended by going HIGH.

Thus, to ensure that a clocked FF will respond properly when the active clock transition occurs, the control inputs must be stable (unchanging) for at least a time interval equal to t_S(min) *prior* to the clock transition, and for at least a time interval equal to t_H(min) *after* the clock transition. These time intervals are required to allow for the propagation delays of the internal gates that control the operation of the flip-flop devices.

IC flip-flops will have minimum allowable t_S and t_H values in the nanosecond range. Setup times are usually in the range of 5 to 50 ns, whereas hold times are generally from 0 to 10 ns. Notice that these times are measured between the 50 percent points on the transitions.

These timing requirements are very important in synchronous systems because, as we shall see, there will be many situations where the synchronous control inputs to a FF are changing at approximately the same time as the *CLK* input.

OUTCOME ASSESSMENT QUESTIONS

1. What two types of inputs does a clocked FF have?
2. What is meant by the term *edge-triggered*?
3. *True or false:* The *CLK* input will affect the FF output only when the active transition of the control input occurs.
4. Define the setup time and hold time requirements of a clocked FF.
5. *True or false:* Metastable states are the greatest benefit of using clock flip-flops.
6. What causes a flip-flop to exhibit a metastable state?

5-6 CLOCKED S-R FLIP-FLOP

OUTCOMES

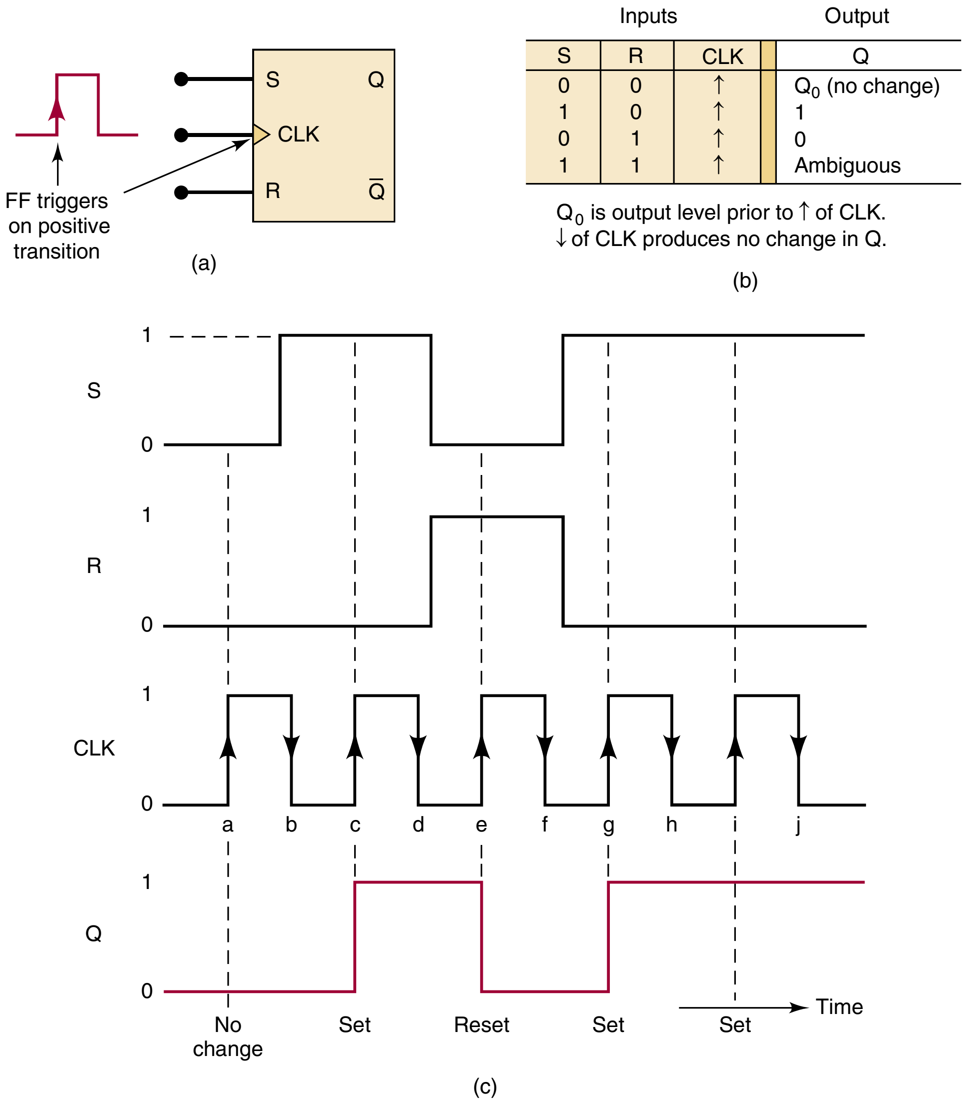
Upon completion of this section, you will be able to:

- Predict the *Q* output given any change on any input, *S*, *R*, or clock.
- Draw a timing diagram to demonstrate input changes and resulting output states.

Figure 5-20(a) shows the logic symbol for a **clocked S-R flip-flop** that is triggered by the positive-going edge of the clock signal. This means that the FF can change states *only* when a signal applied to its clock input makes a transition from 0 to 1. The *S* and *R* inputs control the state of the FF in the same manner as described earlier for the NOR gate latch, but the FF does not respond to these inputs until the occurrence of the PGT of the clock signal.

The function table in Figure 5-20(b) shows how the FF output will respond to the PGT at the *CLK* input for the various combinations of *S* and *R* inputs. This function table uses some new nomenclature. The up arrow (↑) indicates that a PGT is required at *CLK*; the label Q_0 indicates the level

FIGURE 5-20 (a) Clocked S-R flip-flop that responds only to the positive-going edge of a clock pulse; (b) function table; (c) typical waveforms.



Inputs			Output
S	R	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	Ambiguous

Q_0 is output level prior to ↑ of CLK.
↓ of CLK produces no change in Q.

(b)

at Q prior to the PGT. This nomenclature is often used by IC manufacturers in their IC data manuals.

The waveforms in Figure 5-20(c) illustrate the operation of the clocked S-R flip-flop. If we assume that the setup and hold time requirements are being met in all cases, we can analyze these waveforms as follows:

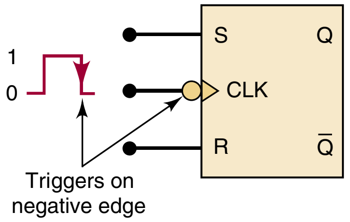
1. Initially all inputs are 0 and the Q output is assumed to be 0; that is, $Q_0 = 0$.
2. When the PGT of the first clock pulse occurs (point *a*), the S and R inputs are both 0, so the FF is not affected and remains in the $Q = 0$ state (i.e., $Q = Q_0$).
3. At the occurrence of the PGT of the second clock pulse (point *c*), the S input is now HIGH, with R still LOW. Thus, the FF sets to the 1 state at the rising edge of this clock pulse.
4. When the third clock pulse makes its positive transition (point *e*), it finds that $S = 0$ and $R = 1$, which causes the FF to clear to the 0 state.
5. The fourth pulse sets the FF once again to the $Q = 1$ state (point *g*) because $S = 1$ and $R = 0$ when the positive edge occurs.

6. The fifth pulse also finds that $S = 1$ and $R = 0$ when it makes its positive-going transition. However, Q is already HIGH, so it remains in that state.
7. The $S = R = 1$ condition should not be used because it results in an ambiguous condition.

It should be noted from these waveforms that the FF is not affected by the negative-going transitions of the clock pulses. Also, note that the S and R levels have no effect on the FF, except upon the occurrence of a positive-going transition of the clock signal. The S and R inputs are synchronous *control* inputs; they control which state the FF will go to when the clock pulse occurs. The *CLK* input is the **trigger** input that causes the FF to change states according to what the S and R inputs are when the active clock transition occurs.

Figure 5-21 shows the symbol and the function table for a clocked S-R flip-flop that triggers on the *negative*-going transition at its *CLK* input. The small circle and triangle on the *CLK* input indicates that this FF will trigger only when the *CLK* input goes from 1 to 0. This FF operates in the same manner as the positive-edge FF except that the output can change states only on the falling edge of the clock pulses (points *b*, *d*, *f*, *h*, and *j* in Figure 5-20). Both positive-edge and negative-edge triggering FFs are used in digital systems.

FIGURE 5-21 Clocked S-R flip-flop that triggers only on negative-going transitions.

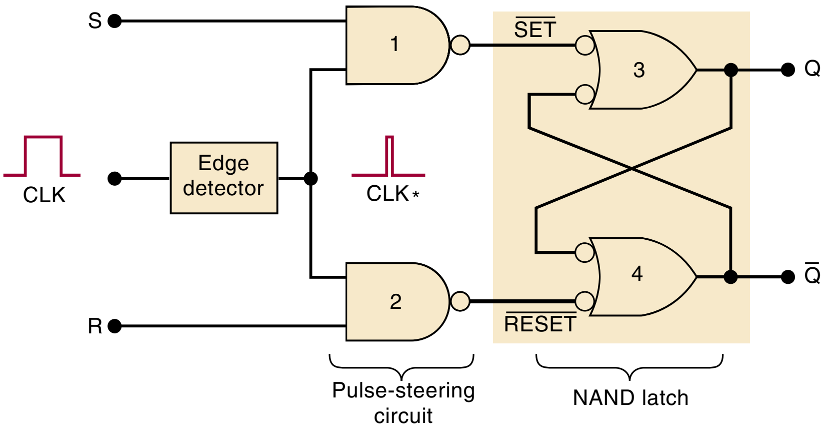


Inputs			Output
S	R	CLK	Q
0	0	↓	Q_0 (no change)
1	0	↓	1
0	1	↓	0
1	1	↓	Ambiguous

Internal Circuitry of the Edge-Triggered S-R Flip-Flop

A detailed analysis of the internal circuitry of a clocked FF is not necessary because all types are readily available as ICs. Although our main interest is in the FF's external operation, our understanding of this external operation can be aided by taking a look at a simplified version of the FF's internal circuitry. Figure 5-22 shows this for an edge-triggered S-R flip-flop.

FIGURE 5-22 Simplified version of the internal circuitry for an edge-triggered S-R flip-flop.



The circuit contains three sections:

1. A basic NAND gate latch formed by NAND-3 and NAND-4
2. A **pulse-steering circuit** formed by NAND-1 and NAND-2
3. An **edge-detector circuit**

As shown in Figure 5-22, the edge detector produces a narrow positive-going spike (*CLK**) that occurs coincident with the active transition of the *CLK* input pulse. The pulse-steering circuit "steers" the spike through to the $\overline{\text{SET}}$ or the $\overline{\text{RESET}}$ input of the latch in accordance with the levels present at *S* and *R*. For example, with $S = 1$ and $R = 0$, the *CLK** signal is inverted and passed through NAND-1 to produce a LOW pulse at the $\overline{\text{SET}}$ input of the latch that sets $Q = 1$. With $S = 0, R = 1$, the *CLK** signal is inverted and passed through NAND-2 to produce a low pulse at the $\overline{\text{RESET}}$ input of the latch that resets $Q = 0$.

Figure 5-23(a) shows how the *CLK** signal is generated for edge-triggered FFs that trigger on a PGT. The INVERTER produces a delay of a few nanoseconds so that the transitions of $\overline{CLK}$ occur a little bit after those of *CLK*. The AND gate produces an output spike that is HIGH only for the few nanoseconds when *CLK* and $\overline{CLK}$ are both HIGH. The result is a narrow pulse at *CLK**, which occurs on the PGT of *CLK*. The arrangement of Figure 5-23(b) likewise produces *CLK** on the NGT of *CLK* for FFs that are to trigger on a NGT.

Because the *CLK** signal is HIGH for only a few nanoseconds, *Q* is affected by the levels at *S* and *R* only for a short time during and after the occurrence of the active edge of *CLK*. This is what gives the FF its edge-triggered property.

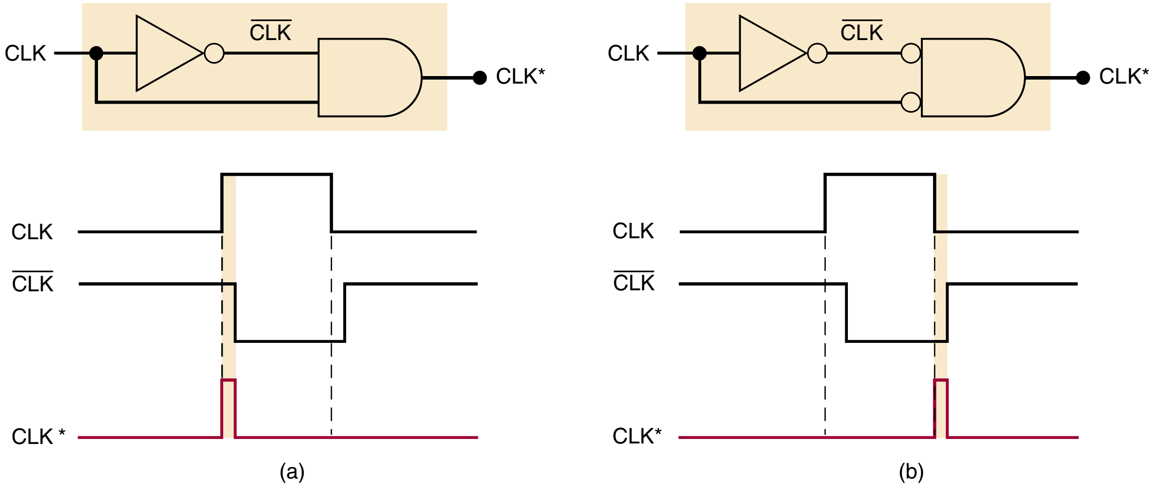


FIGURE 5-23 Implementation of edge-detector circuits used in edge-triggered flip-flops: (a) PGT; (b) NGT. The duration of the *CLK** pulses is typically 2–5 ns.

OUTCOME ASSESSMENT QUESTIONS

1. Suppose that the waveforms of Figure 5-20(c) are applied to the inputs of the FF of Figure 5-21. What will happen to *Q* at point *b*? At point *f*? At point *h*?
2. Explain why the *S* and *R* inputs affect *Q* only during the active transition of *CLK*.

5-7 CLOCKED J-K FLIP-FLOP

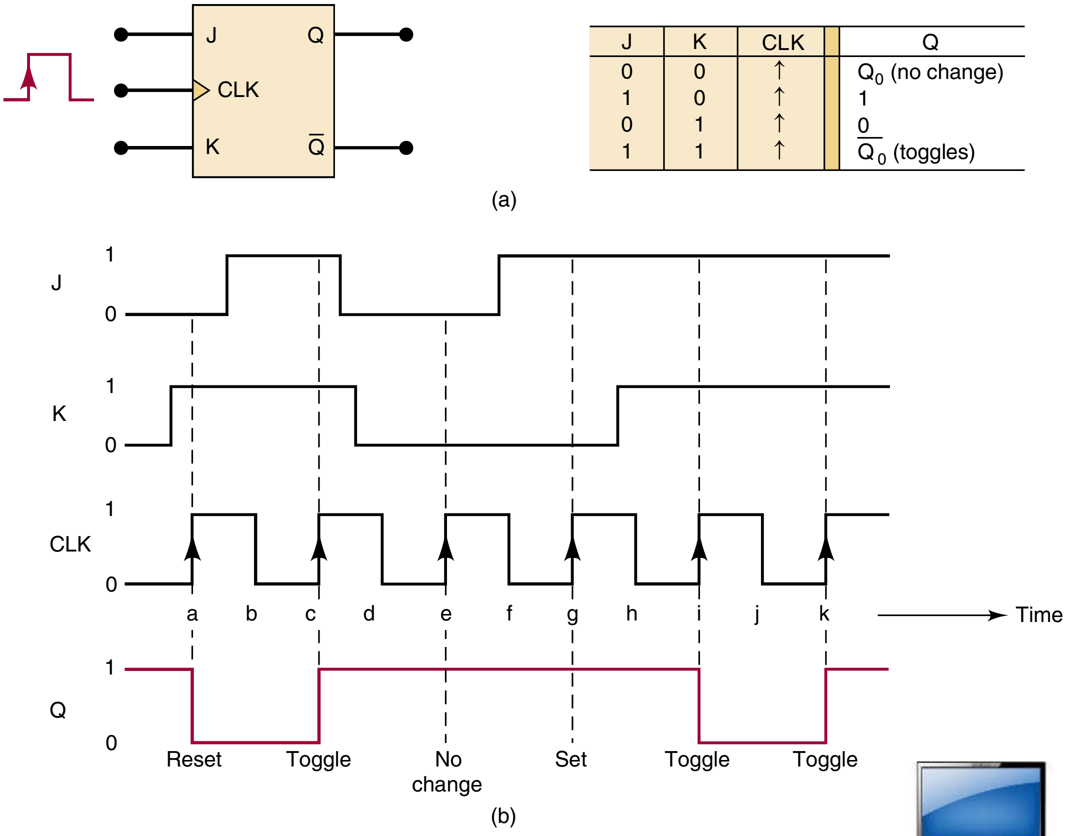
OUTCOMES

Upon completion of this section, you will be able to:

- Identify the purpose of all J-K input combinations.
- Differentiate between a J-K and S-R FF.
- For any change in inputs, *J, K,* clk, predict the output state on *Q*.

Figure 5-24(a) shows a **clocked J-K flip-flop** that is triggered by the positive-going edge of the clock signal. The *J* and *K* inputs control the state of the FF in the same ways as the *S* and *R* inputs do for the clocked S-R flip-flop except for one major difference: *the* $J = K = 1$ *condition does not result in an ambiguous output.* When *J* and *K* are both 1, the FF will always go to its *opposite* state upon the positive transition of the clock signal. This is called the **toggle mode** of operation. In this mode, if both *J* and *K* are left HIGH, the FF will change states (toggle) for each PGT of the clock.

The function table in Figure 5-24(a) summarizes how the J-K flip-flop responds to the PGT for each combination of *J* and *K.* Notice that the function table is the same as for the clocked S-R flip-flop (Figure 5-20) except for the $J = K = 1$ condition. This condition results in $Q = \overline{Q}_0$, which means that the new value of *Q* will be the inverse of the value it had prior to the PGT; this is the toggle operation.



J	K	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	$\overline{Q}_0$ (toggles)

FIGURE 5-24 (a) Clocked J-K flip-flop that responds only to the positive edge of the clock; (b) waveforms.

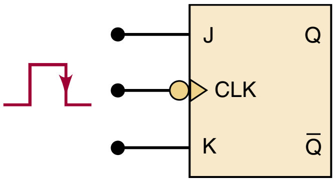
The operation of this FF is illustrated by the waveforms in Figure 5-24(b). Once again, we assume that the setup and hold time requirements are being met.

1. Initially all inputs are 0, and the Q output is assumed to be 1; that is, $Q_0 = 1$.
2. When the positive-going edge of the first clock pulse occurs (point *a*), the $J = 0, K = 1$ condition exists. Thus, the FF will be reset to the $Q = 0$ state.
3. The second clock pulse finds $J = K = 1$ when it makes its positive transition (point *c*). This causes the FF to *toggle* to its opposite state, $Q = 1$.
4. At point *e* on the clock waveform, J and K are both 0, so that the FF does not change states on this transition.
5. At point *g*, $J = 1$ and $K = 0$. This is the condition that sets Q to the 1 state. However, it is already 1, and so it will remain there.
6. At point *i*, $J = K = 1$, and so the FF toggles to its opposite state. The same thing occurs at point *k*.

Note from these waveforms that the FF is not affected by the negative-going edge of the clock pulses. Also, the J and K input levels have no effect except upon the occurrence of the PGT of the clock signal. The J and K inputs by themselves cannot cause the FF to change states.

Figure 5-25 shows the symbol for a clocked J-K flip-flop that triggers on the negative-going clock-signal transitions. The small circle on the *CLK* input indicates that this FF will trigger when the *CLK* input goes from 1 to 0. This FF operates in the same manner as the positive-edge FF of Figure 5-24 except that the output can change states only on negative-going clock-signal transitions (points *b*, *d*, *f*, *h*, and *j*). Both polarities of edge-triggered J-K flip-flops are in common usage.

The J-K flip-flop is much more versatile than the S-R flip-flop because it has no ambiguous states. The $J = K = 1$ condition, which produces the toggling operation, finds extensive use in all types of binary counters. In essence, the J-K flip-flop can do anything the S-R flip-flop can do *plus* operate in the toggle mode.



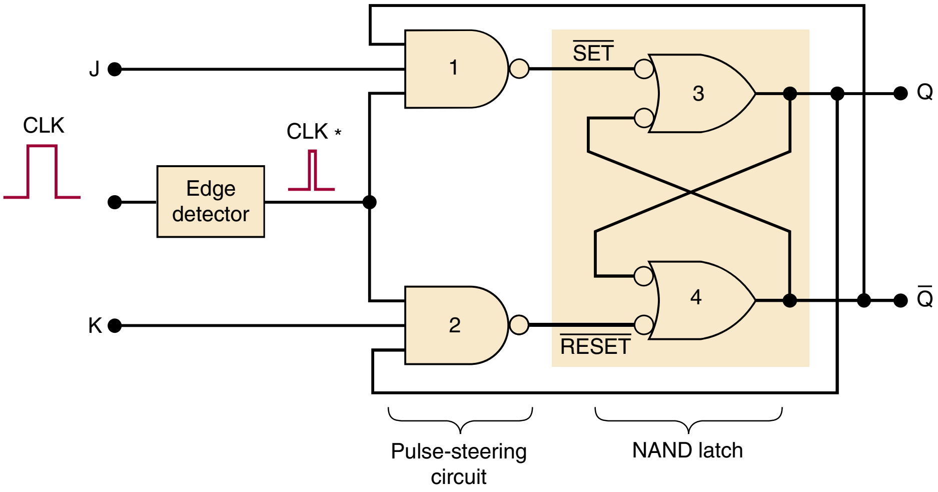
J	K	CLK	Q
0	0	↓	Q_0 (no change)
1	0	↓	1
0	1	↓	0
1	1	↓	$\overline{Q_0}$ (toggles)

FIGURE 5-25 J-K flip-flop that triggers only on negative-going transitions.

Internal Circuitry of the Edge-Triggered J-K Flip-Flop

A simplified version of the internal circuitry of an edge-triggered J-K flip-flop is shown in Figure 5-26. It contains the same three sections as the edge-triggered S-R flip-flop (Figure 5-22). In fact, the only difference between the two circuits is that the Q and $\overline{Q}$ outputs are fed back to the pulse-steering NAND gates. This feedback connection is what gives the J-K flip-flop its toggle operation for the $J = K = 1$ condition.

FIGURE 5-26 Internal circuit of the edge-triggered J-K flip-flop.



Let's examine this toggle condition more closely by assuming that $J = K = 1$ and that Q is sitting in the LOW state when a CLK pulse occurs. With $Q = 0$ and $\overline{Q} = 1$, NAND gate 1 will steer CLK^* (inverted) to the $\overline{\text{SET}}$ input of the NAND latch to produce $Q = 1$. If we assume that Q is HIGH when a CLK pulse occurs, NAND gate 2 will steer CLK^* (inverted) to the $\overline{\text{RESET}}$ input of the latch to produce $Q = 0$. Thus, Q always ends up in the opposite state.

In order for the toggle operation to work as described above, the CLK^* pulse must be very narrow. It must return to 0 before the Q and $\overline{Q}$ outputs toggle to their new values; otherwise, the new values of Q and $\overline{Q}$ will cause the CLK^* pulse to toggle the latch outputs again.

OUTCOME ASSESSMENT QUESTIONS

1. *True or false:* A J-K flip-flop can be used as an S-R flip-flop, but an S-R flip-flop cannot be used as a J-K flip-flop.
2. Does a J-K flip-flop have any ambiguous input conditions?
3. What *J-K* input condition will always set Q upon the occurrence of the active CLK transition?

5-8 CLOCKED D FLIP-FLOP

OUTCOMES

Upon completion of this section, you will be able to:

- Predict the response of a DFF for any sequence of events on its inputs.
- Create or interpret timing diagrams that demonstrate clocked flip-flops.

Figure 5-27(a) shows the symbol and the function table for a **clocked D flip-flop** that triggers on a PGT. Unlike the S-R and J-K flip-flops, this flip-flop has only one synchronous control input, D, which stands for *data*. The operation of the D flip-flop is very simple: Q will go to the same state that is present on the D input when a PGT occurs at CLK. In other words, the level present at D will be *stored* in the flip-flop at the instant the PGT occurs. The waveforms in Figure 5-27(b) illustrate this operation.

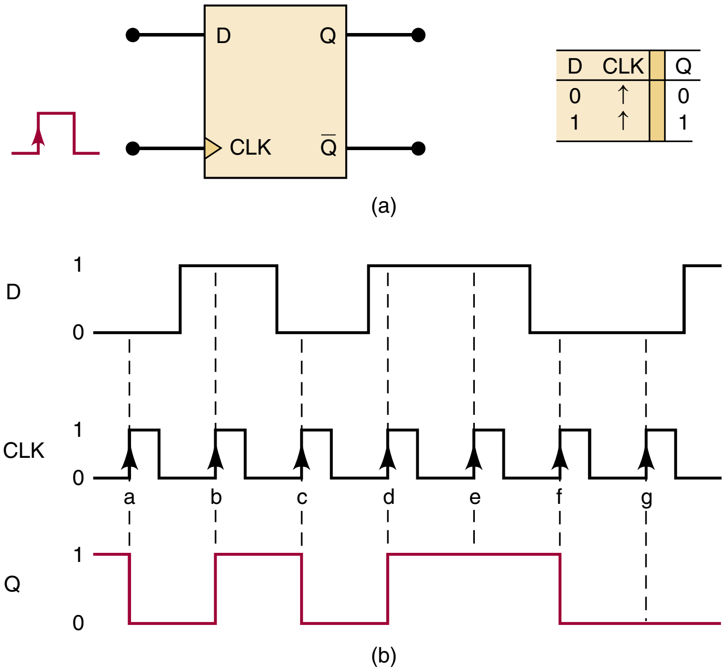


FIGURE 5-27 (a) D flip-flop that triggers only on positive-going transitions; (b) waveforms.

Assume that *Q* is initially HIGH. When the first PGT occurs at point *a*, the *D* input is LOW; thus, *Q* will go to the 0 state. Even though the *D* input level changes between points *a* and *b*, it has no effect on *Q*; *Q* is storing the LOW that was on *D* at point *a*. When the PGT at *b* occurs, *Q* goes HIGH because *D* is HIGH at that time. *Q* stores this HIGH until the PGT at point *c* causes *Q* to go LOW because *D* is LOW at that time. In a similar manner, the *Q* output takes on the levels present at *D* when the PGTs occur at points *d*, *e*, *f*, and *g*. Note that *Q* stays HIGH at point *e* because *D* is still HIGH.

Again, it is important to remember that *Q* can change only when a PGT occurs. The *D* input has no effect between PGTs.

A negative-edge-triggered D flip-flop operates in the same way just described except that *Q* will take on the value of *D* when a NGT occurs at *CLK*. The symbol for the D flip-flop that triggers on NGTs will have a bubble on the *CLK* input.

Implementation of the D Flip-Flop

An edge-triggered D flip-flop is easily implemented by adding a single INVERTER to the edge-triggered J-K flip-flop, as shown in Figure 5-28. If you try both values of *D*, you should see that *Q* takes on the level present at *D* when a PGT occurs. The same can be done to convert a S-R flip-flop to a D flip-flop.

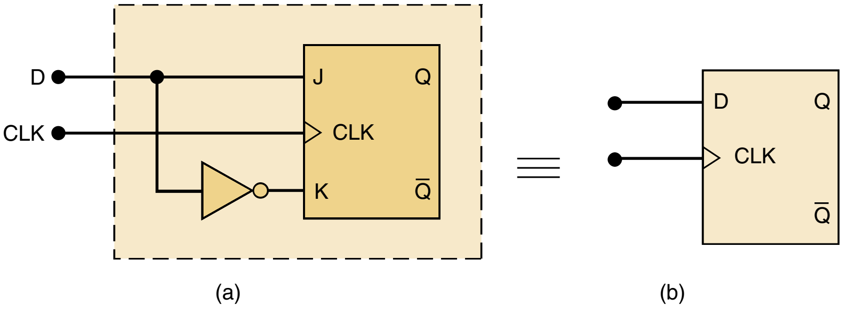


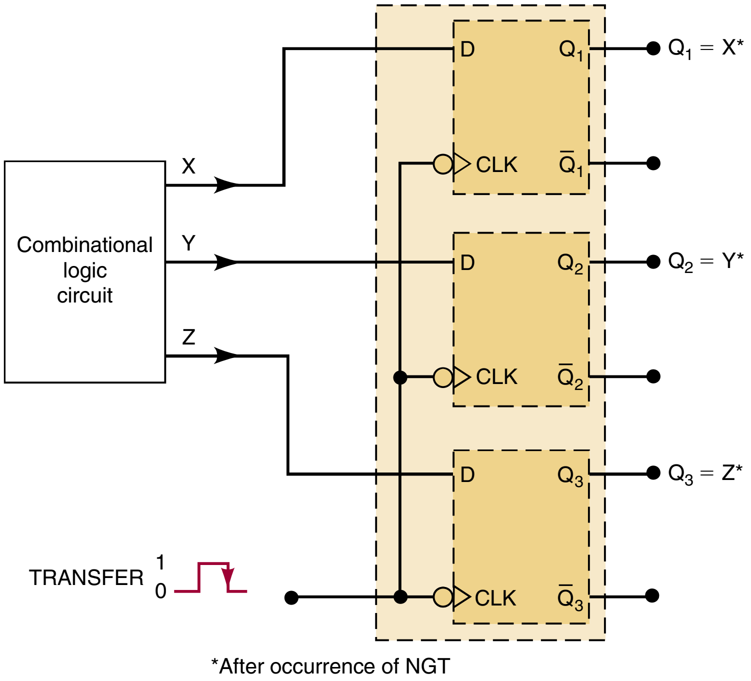
FIGURE 5-28 Edge-triggered D flip-flop implementation from a J-K flip-flop.

Parallel Data Transfer

At this point you may well be wondering about the usefulness of the D flip-flop because it appears that the Q output is the same as the D input. Not quite; remember, Q takes on the value of D only at certain time instances, and so it is not identical to D (e.g., see the waveforms in Figure 5-27).

In most applications of the D flip-flop, the Q output must take on the value at its D input only at precisely defined times. One example of this is illustrated in Figure 5-29. Outputs X, Y, Z from a logic circuit are to be transferred to FFs Q_1, Q_2, and Q_3 for storage. Using the D flip-flops, the levels present at X, Y, and Z will be transferred to Q_1, Q_2, and Q_3, respectively, upon application of a TRANSFER pulse to the common CLK inputs. The FFs can store these values for subsequent processing. This is an example of **parallel data transfer** of binary data; the three bits X, Y, and Z are all transferred *simultaneously.*

FIGURE 5-29 Parallel transfer of binary data using D flip-flops.



OUTCOME ASSESSMENT QUESTIONS

1. What will happen to the Q waveform in Figure 5-27(b) if the D input is held permanently LOW?
2. *True or false:* The Q output will equal the level at the D input at all times.
3. Can J-K FFs be used for parallel data transfer?

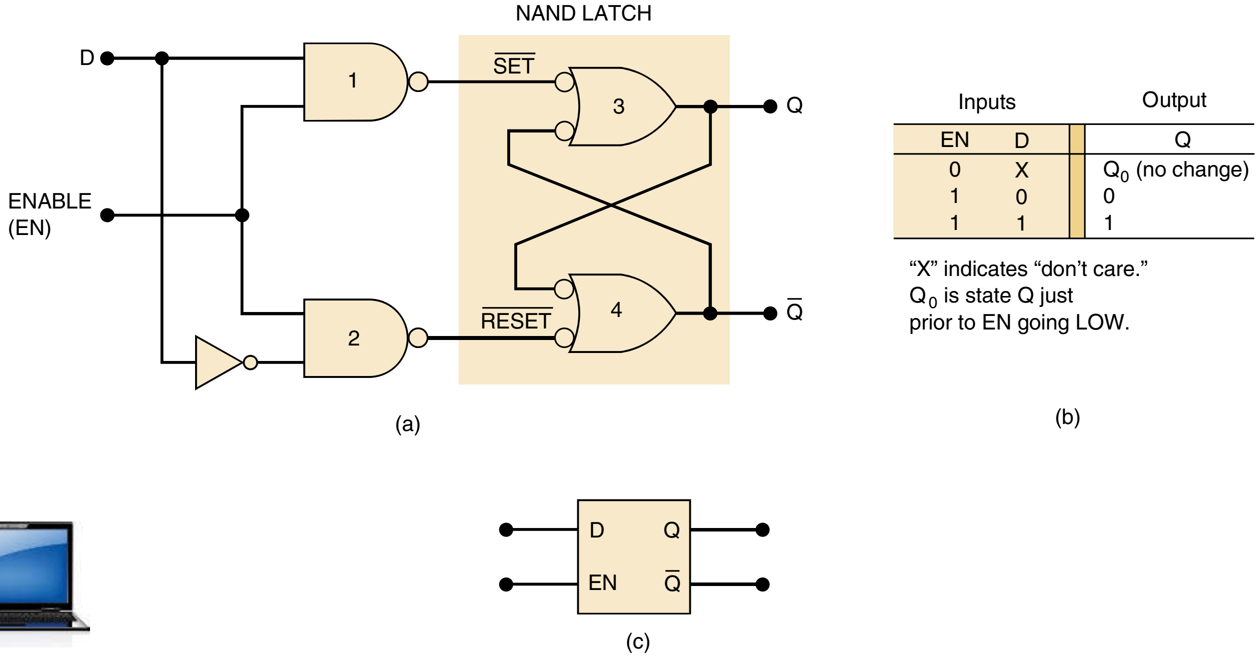
5-9 D LATCH (TRANSPARENT LATCH)

OUTCOMES

Upon completion of this section, you will be able to:

- Differentiate between a D latch and a D FF operation.
- Identify the latched and transparent conditions of a D latch.
- Predict outputs for any input change in latched and transparent mode of a D latch.

The edge-triggered D flip-flop uses an edge-detector circuit to ensure that the output will respond to the *D* input *only* when the active transition of the clock occurs. If this edge detector is not used, the resultant circuit operates somewhat differently. It is called a **D latch** and has the arrangement shown in Figure 5-30(a).



Inputs		Output
EN	D	Q
0	X	Q_0 (no change)
1	0	0
1	1	1

"X" indicates "don't care."
Q_0 is state Q just
prior to EN going LOW.

FIGURE 5-30 D latch: (a) structure; (b) function table; (c) logic symbol.

The circuit contains the NAND latch and the steering NAND gates 1 and 2 *without* the edge-detector circuit. The common input to the steering gates is called an *enable* input (abbreviated *EN*) rather than a clock input because its effect on the *Q* and $\overline{Q}$ outputs is not restricted to occurring only on its transitions. The operation of the D latch is described as follows:

1. When *EN* is HIGH, the *D* input will produce a LOW at either the $\overline{\text{SET}}$ or the $\overline{\text{RESET}}$ inputs of the NAND latch to cause *Q* to become the same level as *D*. If *D* changes while *EN* is HIGH, *Q* will follow the changes exactly. In other words, while $EN = 1$, the *Q* output will look exactly like *D*; in this mode, the D latch is said to be "transparent."
2. When *EN* goes LOW, the *D* input is inhibited from affecting the NAND latch because the outputs of both steering gates will be held HIGH. Thus, the *Q* and $\overline{Q}$ outputs will stay at whatever level they had just before *EN* went LOW. In other words, the outputs are "latched" to their current level and cannot change while *EN* is LOW even if *D* changes.

This operation is summarized in the function table in Figure 5-30(b). The logic symbol for the D latch is given in Figure 5-30(c). Note that even though the *EN* input operates much like the *CLK* input of an edge-triggered FF, there is no small triangle on the *EN* input. This is because the small triangle symbol is used strictly for inputs that can cause an output change only when a transition occurs. *The D latch is not edge-triggered.*

EXAMPLE 5-8

Determine the Q waveform for a D latch with the EN and D inputs of Figure 5-31. Assume that $Q = 0$ initially.

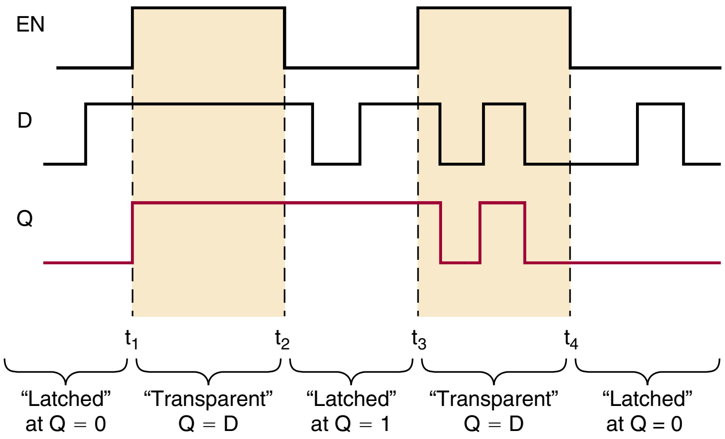


FIGURE 5-31 Waveforms for Example 5-8 showing the two modes of operation of the transparent D latch.

Solution

Prior to time t_1, EN is LOW, so that Q is "latched" at its current 0 level and cannot change even though D is changing. During the interval t_1 to t_2, EN is HIGH so that Q will follow the signal present at D. Thus, Q goes HIGH at t_1 and stays there because D is not changing. When EN returns LOW at t_2, Q will latch at the HIGH level that it has at t_2 and will remain there while EN is LOW.

At t_3 when EN goes HIGH again, Q will follow the changes in the D input until t_4 when EN returns LOW. During the interval t_3 to t_4, the D latch is "transparent" because the variations in D go through to the output Q. At t_4 when EN goes LOW, Q will latch at the 0 level because that is its level at t_4. After t_4 the variations in D will have no effect on Q because it is latched (i.e., $EN = 0$).

OUTCOME ASSESSMENT QUESTIONS

1. Describe how a D latch operates differently from an edge-triggered D flip-flop.
2. *True or false:* A D latch is in its transparent mode when $EN = 0$.
3. *True or false:* In a D latch, the D input can affect Q only when $EN = 1$.

5-10 ASYNCHRONOUS INPUTS

OUTCOMES

Upon completion of this section, you will be able to:

- Distinguish between synchronous and asynchronous inputs.
- For any change on any synchronous or asynchronous input, predict the output state of Q.