10

Flip-Flops and Related Devices

Having discussed combinational logic circuits at length in previous chapters, the focus in the present chapter and in Chapter 11 will be on sequential logic circuits. While a logic gate is the most basic building block of combinational logic, its counterpart in sequential logic is the flip-flop. The chapter begins with a brief introduction to different types of multivibrator, including the bistable multivibrator, which is the complete technical name for a flip-flop, the monostable multivibrator and the astable multivibrator. The flip-flop is not only used individually for a variety of applications; it also forms the basis of many more complex logic functions. Counters and registers, to be covered in Chapter 11 are typical examples. There is a large variety of flip-flops having varying functional tables, input clocking requirements and other features. In this chapter, we will discuss all these basic types of flip-flop in terms of their functional aspects, truth tables, salient features and application aspects. The text is suitably illustrated with a large number of solved examples. Application-relevant information, including a comprehensive index of flip-flops and related devices belonging to different logic families, is given towards the end of the chapter. Pin connection diagrams and functional tables are given in the companion website.

10.1 Multivibrator

Multivibrators, like the familiar sinusoidal oscillators, are circuits with regenerative feedback, with the difference that they produce pulsed output. There are three basic types of multivibrator, namely the bistable multivibrator, the monostable multivibrator and the astable multivibrator.

10.1.1 Bistable Multivibrator

A *bistable multivibrator* circuit is one in which both LOW and HIGH output states are stable. Irrespective of the logic status of the output, LOW or HIGH, it stays in that state unless a change is

Digital Electronics: Principles, Devices and Applications Anil K. Maini
© 2007 John Wiley & Sons, Ltd. ISBN: 978-0-470-03214-5

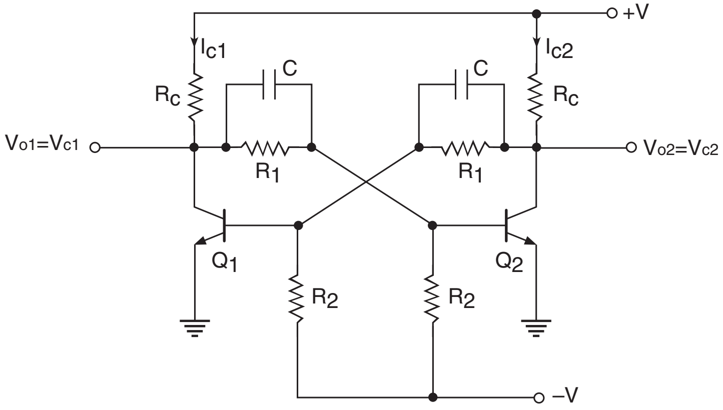


Figure 10.1 Bistable multivibrator.

induced by applying an appropriate trigger pulse. As we will see in the subsequent pages, the operation of a bistable multivibrator is identical to that of a flip-flop. Figure 10.1 shows the basic bistable multivibrator circuit. This is the fixed-bias type of bistable multivibrator. Other configurations are the self-bias type and the emitter-coupled type. However, the operational principle of all types is the same. The multivibrator circuit of Fig. 10.1 functions as follows.

In the circuit arrangement of Fig. 10.1 it can be proved that both transistors Q_1 and Q_2 cannot be simultaneously ON or OFF. If Q_1 is ON, the regenerative feedback ensures that Q_2 is OFF, and when Q_1 is OFF, the feedback drives transistor Q_2 to the ON state. In order to vindicate this statement, let us assume that both Q_1 and Q_2 are conducting simultaneously. Owing to slight circuit imbalance, which is always there, the collector current in one transistor will always be greater than that in the other. Let us assume that $I_{c2} > I_{c1}$. Lesser I_{c1} means a higher V_{c1}. Since V_{c1} is coupled to the Q_2 base, a rise in V_{c1} leads to an increase in the Q_2 base voltage. Increase in the Q_2 base voltage results in an increase in I_{c2} and an associated reduction in V_{c2}. Reduction in V_{c2} leads to a reduction in Q_1 base voltage and an associated fall in I_{c1}, with the result thatV_{c1} increases further. Thus, a slight circuit imbalance has initiated a regenerative action that culminates in transistor Q_1 going to cut-off and transistor Q_2 getting driven to saturation. To sum up, whenever there is a tendency of one of the transistors to conduct more than the other, it will end up with that transistor going to saturation and driving the other transistor to cut-off. Now, if we take the output from the Q_1 collector, it will be LOW ($= V_{CE1}$ sat.) if Q_1 was initially in saturation. If we apply a negative-going trigger to the Q_1 base to cause a decrease in its collector current, a regenerative action would set in that would drive Q_2 to saturation and Q_1 to cut-off. As a result, the output goes to a HIGH ($= +V_{CC}$) state. The output will stay HIGH until we apply another appropriate trigger to initiate a transition. Thus, both of the output states, when the output is LOW and also when the output is HIGH, are stable and undergo a change only when a transition is induced by means of an appropriate trigger pulse. That is why it is called a bistable multivibrator.

10.1.2 Schmitt Trigger

A Schmitt trigger circuit is a slight variation of the bistable multivibrator circuit of Fig. 10.1. Figure 10.2 shows the basic Schmitt trigger circuit. If we compare the bistable multivibrator circuit of Fig. 10.1

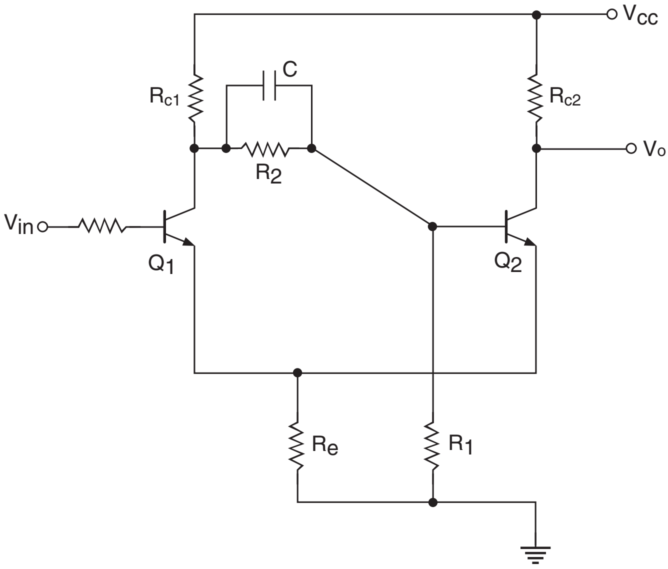


Figure 10.2 Schmitt trigger circuit.

with the Schmitt trigger circuit of Fig. 10.2, we find that coupling from Q_2 collector to Q_1 base in the case of a bistable circuit is absent in the case of a Schmitt trigger circuit. Instead, the resistance R_e provides the coupling. The circuit functions as follows.

When V_{in} is zero, transistor Q_1 is in cut-off. Coupling from Q_1 collector to Q_2 base drives transistor Q_2 to saturation, with the result that V_o is LOW. If we assume that V_{CE2} (sat.) is zero, then the voltage across R_e is given by the equation

$$\text{Voltage across } R_e = [V_{CC}.R_e/(R_e + R_{c2})] \tag{10.1}$$

This is also the emitter voltage of transistor Q_1. In order to make transistor Q_1 conduct, V_{in} must be at least 0.7 V more than the voltage across R_e. That is,

$$V_{in}(\text{min.}) = [V_{CC}.R_e/(R_e + R_{c2})] + 0.7 \tag{10.2}$$

When V_{in} exceeds this voltage, Q_1 starts conducting. The regenerative action again drives Q_2 to cut-off. The output goes to the HIGH state. Voltage across R_e changes to a new value given by the equation

$$\text{Voltage across } R_e = [V_{CC}.R_e/(R_e + R_{c1})] \tag{10.3}$$

$$V_{in} = [V_{CC}.R_e/(R_e + R_{c1})] + 0.7 \tag{10.4}$$

Transistor Q_1 will continue to conduct as long as V_{in} is equal to or greater than the value given by Equation (10.4). If V_{in} falls below this value, Q_1 tends to come out of saturation and conduct less heavily. The regenerative action does the rest, with the process culminating in Q_1 going to cut-off and Q_2 to saturation. Thus, the state of output (HIGH or LOW) depends upon the input voltage level.

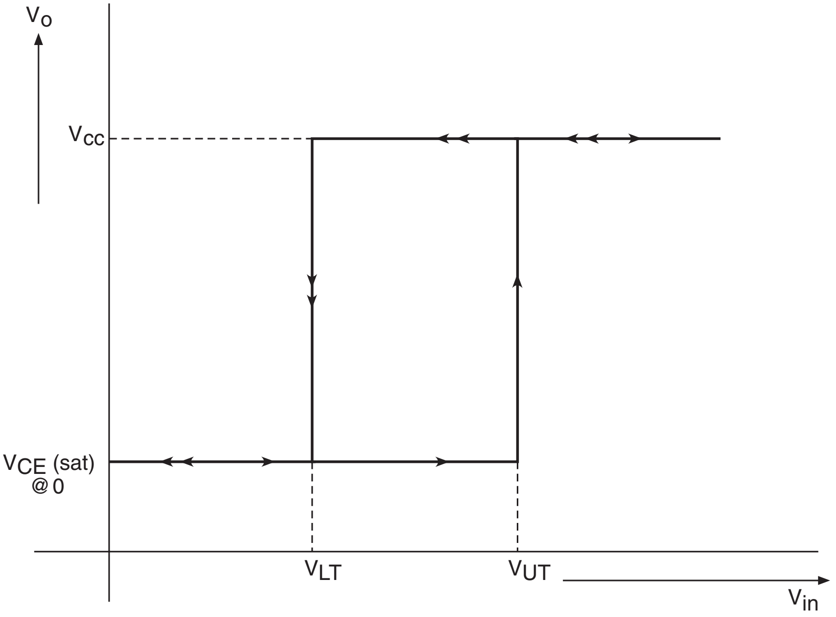


Figure 10.3 Transfer characteristics of a Schmitt trigger.

The HIGH and LOW states of the output correspond to two distinct input levels given by Equations (10.2) and (10.4) and therefore the values of R_{c1},R_{c2},R_e and V_{CC}. The Schmitt trigger circuit of Fig. 10.2 therefore exhibits hysteresis. Figure 10.3 shows the transfer characteristics of the Schmitt trigger circuit. The lower trip point V_{LT} and the upper trip point V_{UT} of these characteristics are respectively given by the equations

$$V_{LT} = [V_{CC}.R_e/(R_e + R_{c1})] + 0.7 \tag{10.5}$$

$$V_{UT} = [V_{CC}.R_e/(R_e + R_{c2})] + 0.7 \tag{10.6}$$

10.1.3 Monostable Multivibrator

A *monostable multivibrator*, also known as a *monoshot*, is one in which one of the states is stable and the other is quasi-stable. The circuit is initially in the stable state. It goes to the quasi-stable state when appropriately triggered. It stays in the quasi-stable state for a certain time period, after which it comes back to the stable state. Figure 10.4 shows the basic monostable multivibrator circuit. The circuit functions as follows. Initially, transistor Q_2 is in saturation as it gets its base bias from $+V_{CC}$ through R. Coupling from Q_2 collector to Q_1 base ensures that Q_1 is in cut-off. Now, if an appropriate trigger pulse induces a transition in Q_2 from saturation to cut-off, the output goes to the HIGH state. This HIGH output when coupled to the Q_1 base turns Q_1 ON. Since there is no direct coupling from Q_1 collector to Q_2 base, which is necessary for a regenerative process to set in, Q_1 is not necessarily

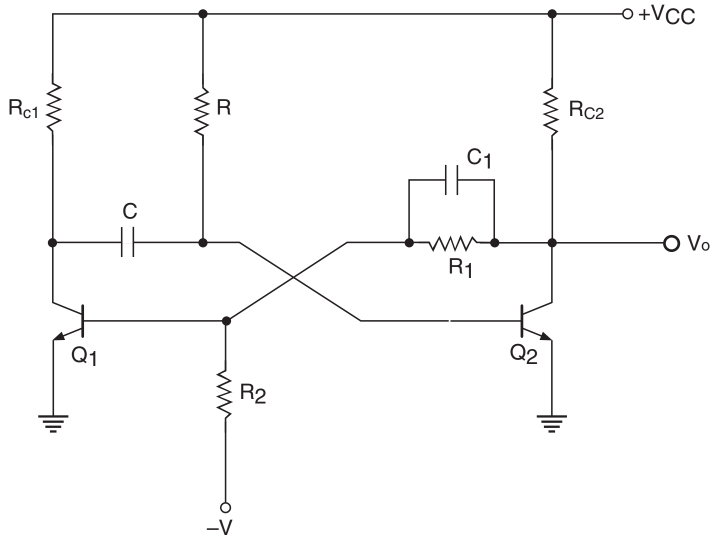


Figure 10.4 Monostable multivibrator.

in saturation. However, it conducts some current. The Q_1 collector voltage falls by $I_{c1}R_{c1}$ and the Q_2 base voltage falls by the same amount, as the voltage across a capacitor (C in this case) cannot change instantaneously. To sum up, the moment we applied the trigger, Q_2 went to cut-off and Q_1 started conducting. But now there is a path for capacitor C to charge from V_{CC} through R and the conducting transistor. The polarity of voltage across C is such that the Q_2 base potential rises. The moment the Q_2 base voltage exceeds the cut-in voltage, it turns Q_2 ON, which, owing to coupling through R_1, turns Q_1 OFF. And we are back to the original state, the stable state. Whenever we trigger the circuit into the other state, it does not stay there permanently and returns back after a time period that depends upon R and C. The greater the time constant RC, the longer is the time for which it stays in the other state, called the quasi-stable state.

10.1.3.1 Retriggerable Monostable Multivibrator

In a conventional monostable multivibrator, once the output is triggered to the quasi-stable state by applying a suitable trigger pulse, the circuit does not respond to subsequent trigger pulses as long as the output is in quasi-stable state. After the output returns to its original state, it is ready to respond to the next trigger pulse. There is another class of monostable multivibrators, called *retriggerable monostable multivibrators*. These respond to trigger pulses even when the output is in the quasi-stable state. In this class of monostable multivibrators, if n trigger pulses with a time period of T_t are applied to the circuit, the output pulse width, that is, the time period of the quasi-stable state, equals $(n-1)T_t + T$, where T is the output pulse width for the single trigger pulse and $T_t < T$. Figure 10.5 shows the output pulse width in the case of a retriggerable monostable multivibrator for repetitive trigger pulses.

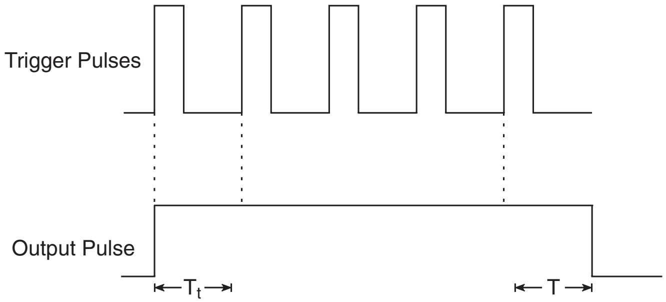


Figure 10.5 Retriggerable monostable multivibrator output for repetitive trigger pulses.

10.1.4 Astable Multivibrator

In the case of an astable multivibrator, neither of the two states is stable. Both output states are quasi-stable. The output switches from one state to the other and the circuit functions like a free-running square-wave oscillator. Figure 10.6 shows the basic astable multivibrator circuit. It can be proved that, in this type of circuit, neither of the output states is stable. Both states, LOW as well as HIGH, are quasi-stable. The time periods for which the output remains LOW and HIGH depends upon R_2C_2 and R_1C_1 time constants respectively. For $R_1C_1 = R_2C_2$, the output is a symmetrical square waveform. The circuit functions as follows. Let us assume that transistor Q_2 is initially conducting, that is, the output is LOW. Capacitor C_2 in this case charges through R_2 and the conducting transistor from V_{CC}, and, the moment the Q_1 base potential exceeds its cut-in voltage, it is turned ON. A fall in Q_1 collector

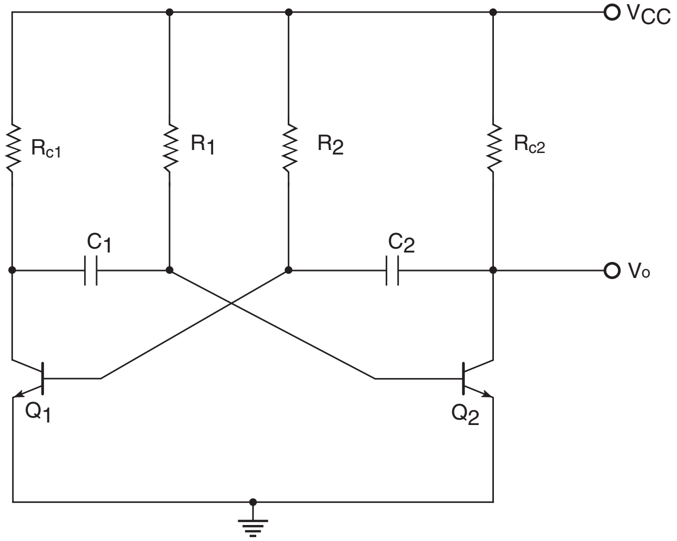


Figure 10.6 Astable multivibrator.

potential manifests itself at the Q_2 base as voltage across a capacitor cannot change instantaneously. The output goes to the HIGH state as Q_2 is driven to cut-off. However, C_1 has now started charging through R_1 and the conducting transistor Q_1 from V_{CC}. The moment the Q_2 base potential exceeds the cut-in voltage, it is again turned ON, with the result that the output goes to the LOW state. This process continues and, owing to both the couplings (Q_1 collector to Q_2 base and Q_2 collector to Q_1 base) being capacitive, neither of the states is stable. The circuit produces a square-wave output.

10.2 Integrated Circuit (IC) Multivibrators

In this section, we will discuss monostable and astable multivibrator circuits that can be configured around some of the popular digital and linear integrated circuits. The bistable multivibrator, which is functionally the same as a flip-flop, will not be discussed here. Flip-flops are discussed at length from Section 10.3 onwards.

10.2.1 Digital IC-Based Monostable Multivibrator

Some of the commonly used digital ICs that can be used as monostable multivibrators include 74121 (single monostable multivibrator), 74221 (dual monostable multivibrator), 74122 (single retriggerable monostable multivibrator) and 74123 (dual retriggerable monostable multivibrator), all belonging to the TTL family, and 4098B (dual retriggerable monostable multivibrator) belonging to the CMOS family. Figure 10.7 shows the use of IC 74121 as a monostable multivibrator along with a trigger input. The IC provides features for triggering on either LOW-to-HIGH or HIGH-to-LOW edges of the trigger pulses. Figure 10.7(a) shows one of the possible application circuits for HIGH-to-LOW edge triggering, and Fig. 10.7(b) shows one of the possible application circuits for LOW-to-HIGH edge triggering. The output pulse width depends on external R and C. The output pulse width can be computed from $T = 0.7RC$. Recommended ranges of values for R and C are 4–40 K Ω and 10 pf to 1000 μF respectively. The IC provides complementary outputs. That is, we have a stable LOW or HIGH state and the corresponding quasi-stable HIGH or LOW state available on Q and $\overline{Q}$ outputs.

Figure 10.8 shows the use of 74123, a retriggerable monostable multivibrator. Like 74121, this IC, too, provides features for triggering on either LOW-to-HIGH or HIGH-to-LOW edges of the trigger pulses. The output pulse width depends on external R and C. It can be computed from $T = 0.28RC\times[1 + (0.7/R)]$, where R and C are respectively in kiloohms and picofarads and T is in nanoseconds. This formula is valid for $C > 1000$ pF. The recommended range of values for R is 5–50 KΩ . Figures 10.8(a) and (b) give application circuits for HIGH-to-LOW and LOW-to-HIGH triggering respectively. It may be mentioned here that there can be other triggering circuit options for both LOW-to-HIGH and HIGH-to-LOW edge triggering of monoshot.

10.2.2 IC Timer-Based Multivibrators

IC timer 555 is one of the most commonly used general-purpose linear integrated circuits. The simplicity with which monostable and astable multivibrator circuits can be configured around this IC is one of the main reasons for its wide use. Figure 10.9 shows the internal schematic of timer IC 555. It comprises two opamp comparators, a flip-flop, a discharge transistor, three identical resistors and an output stage. The resistors set the reference voltage levels at the noninverting input of the lower comparator and the inverting input of the upper comparator at $(+V_{CC}/3)$ and $(+2V_{CC}/3)$. The outputs of the two comparators feed the SET and RESET inputs of the flip-flop and thus decide the logic status

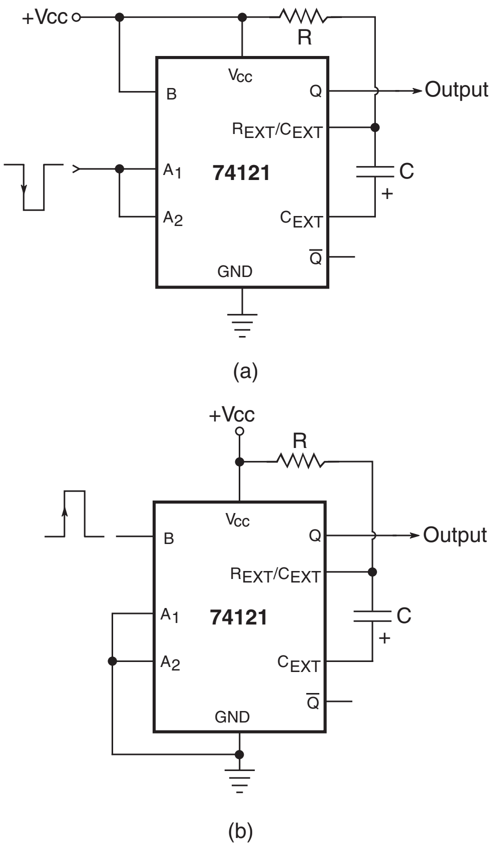


Figure 10.7 74121 as a monoshot.

of its output and subsequently the final output. The flip-flop complementary outputs feed the output stage and the base of the discharge transistor. This ensures that when the output is HIGH the discharge transistor is OFF, and when the output is LOW the discharge transistor is ON. Different terminals of timer 555 are designated as *ground* (terminal 1), *trigger* (terminal 2), *output* (terminal 3), *reset* (terminal 4), *control* (terminal 5), *threshold* (terminal 6), *discharge* (terminal 7) and $+V_{CC}$ (terminal 8). With this background, we will now describe the astable and monostable circuits configured around timer 555.

10.2.2.1 Astable Multivibrator Using Timer IC 555

Figure 10.10(a) shows the basic 555 timer based astable multivibrator circuit. Initially, capacitor C is fully discharged, which forces the output to go to the HIGH state. An open discharge transistor allows the capacitor C to charge from $+V_{CC}$ through R_1 and R_2. When the voltage across C exceeds $+2V_{CC}/3$, the output goes to the LOW state and the discharge transistor is switched ON at the same time.

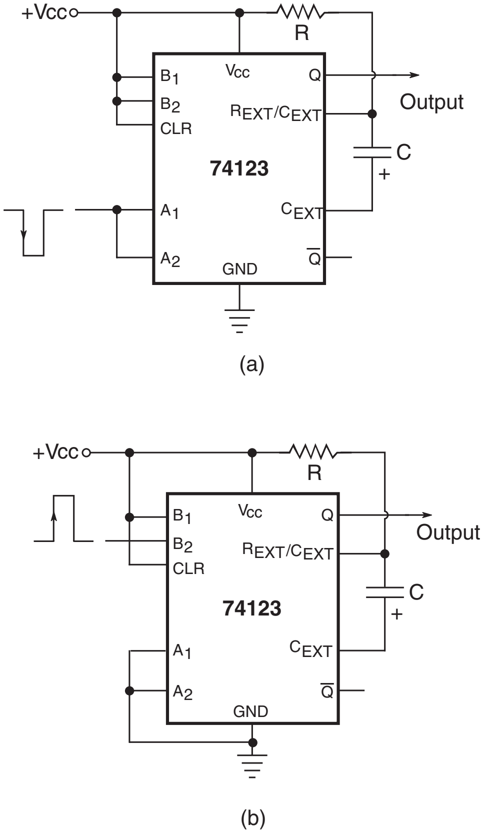


Figure 10.8 74123 as a retriggerable monoshot.

Capacitor C begins to discharge through R_2 and the discharge transistor inside the IC. When the voltage across C falls below $+V_{CC}/3$, the output goes back to the HIGH state. The charge and discharge cycles repeat and the circuit behaves like a free-running multivibrator. Terminal 4 of the IC is the RESET terminal. usually, it is connected to $+V_{CC}$. If the voltage at this terminal is driven below 0.4 V, the output is forced to the LOW state, overriding command pulses at terminal 2 of the IC. The HIGH-state and LOW-state time periods are governed by the charge ($+V_{CC}/3$ to $+2V_{CC}/3$) and discharge ($+2V_{CC}/3$ to $+V_{CC}/3$) timings. these are given by the equations

$$\text{HIGH-state time period } T_{\text{HIGH}} = 0.69(R_1 + R_2).C \tag{10.7}$$

$$\text{LOW-state time period } T_{\text{LOW}} = 0.69R_2.C \tag{10.8}$$

The relevant waveforms are shown in Fig. 10.10(b). The time period T and frequency f of the output waveform are respectively given by the equations

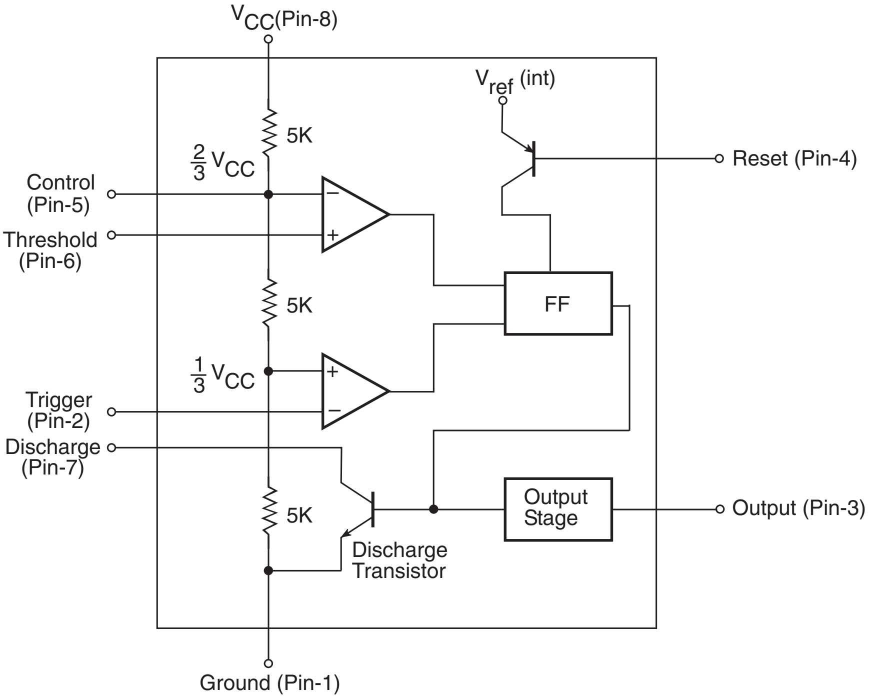


Figure 10.9 Internal schematic of timer IC 555.

$$\text{Time period } T = 0.69(R_1 + 2R_2).C \tag{10.9}$$

$$\text{Frequency } F = 1/[0.69(R_1 + 2R_2).C] \tag{10.10}$$

Remember that, when the astable multivibrator is powered, the first-cycle HIGH-state time period is about 30 % longer, as the capacitor is initially discharged and it charges from 0 (rather than $+V_{CC}/3$) to $+2V_{CC}/3$.

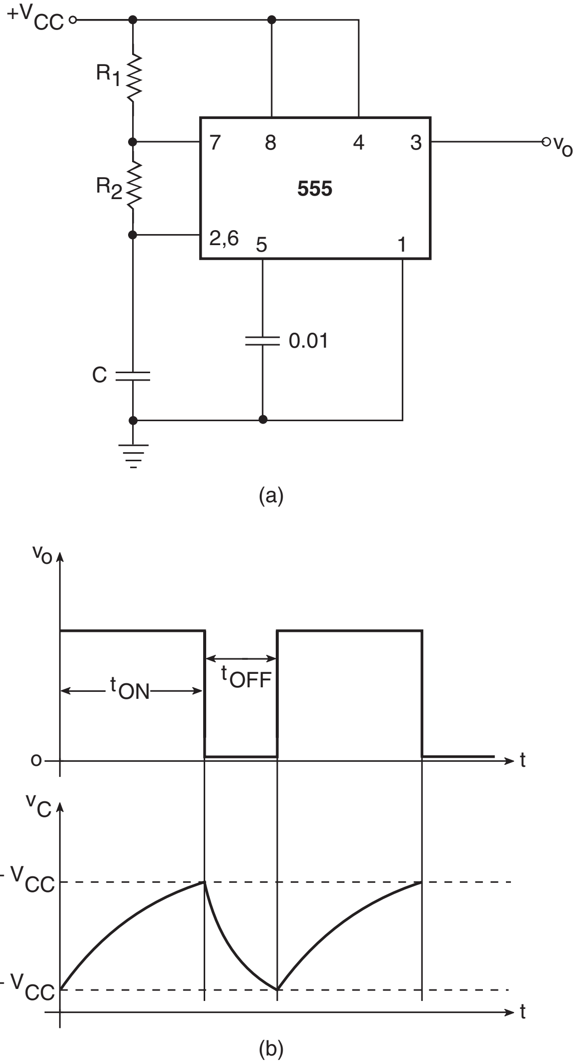
In the case of the astable multivibrator circuit in Fig. 10.10(a), the HIGH-state time period is always greater than the LOW-state time period. Figures 10.10(c) and (d) show two modified circuits where the HIGH-state and LOW-state time periods can be chosen independently. For the astable multivibrator circuits in Fig. 10.10(c) and (d), the two time periods are given by the equations

$$\text{HIGH-state time period} = 0.69R_1.C \tag{10.11}$$

$$\text{LOW-state time period} = 0.69R_2.C \tag{10.12}$$

For $R_1 = R_2 = R$

$$T = 1.38RC \text{ and } f = 1/1.38RC \tag{10.13}$$



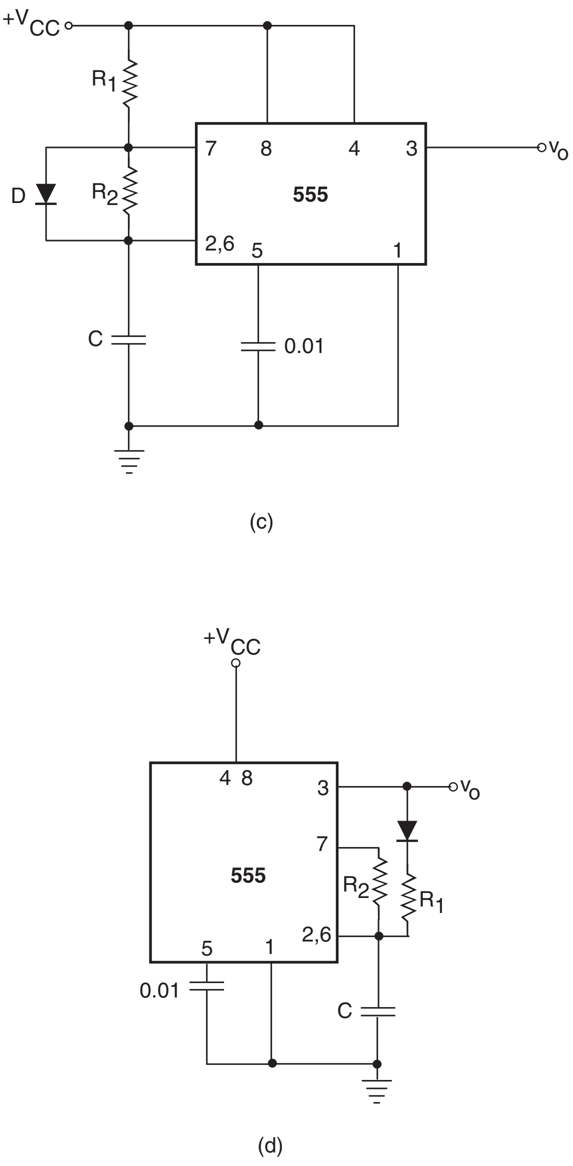
(a)

(b)

Figure 10.10 (a) Astable multivibrator using timer IC 555, (b) astable multivibrator relevant waveforms and (c, d) modified versions of the astable multivibrator using timer IC 555.

10.2.2.2 Monostable Multivibrator Using Timer IC 555

Figure 10.11(a) shows the basic monostable multivibrator circuit configured around timer 555. A trigger pulse is applied to terminal 2 of the IC, which should initially be kept at $+V_{CC}$. A HIGH at terminal 2 forces the output to the LOW state. A HIGH-to-LOW trigger pulse at terminal 2 holds the output in the HIGH state and simultaneously allows the capacitor to charge from $+V_{CC}$ through R. Remember that a LOW level of the trigger pulse needs to go at least below $+V_{CC}/3$. When the capacitor voltage exceeds $+2V_{CC}/3$, the output goes back to the LOW state. We will need to apply another trigger pulse to



(c)

(d)

Figure 10.10 (*continued*).

terminal 2 to make the output go to the HIGH state again. Every time the timer is appropriately triggered, the output goes to the HIGH state and stays there for the time it takes the capacitor to charge from 0 to $+2V_{\mathrm{CC}}/3$. This time period, which equals the monoshot output pulse width, is given by the equation

$$T = 1.1RC \tag{10.14}$$

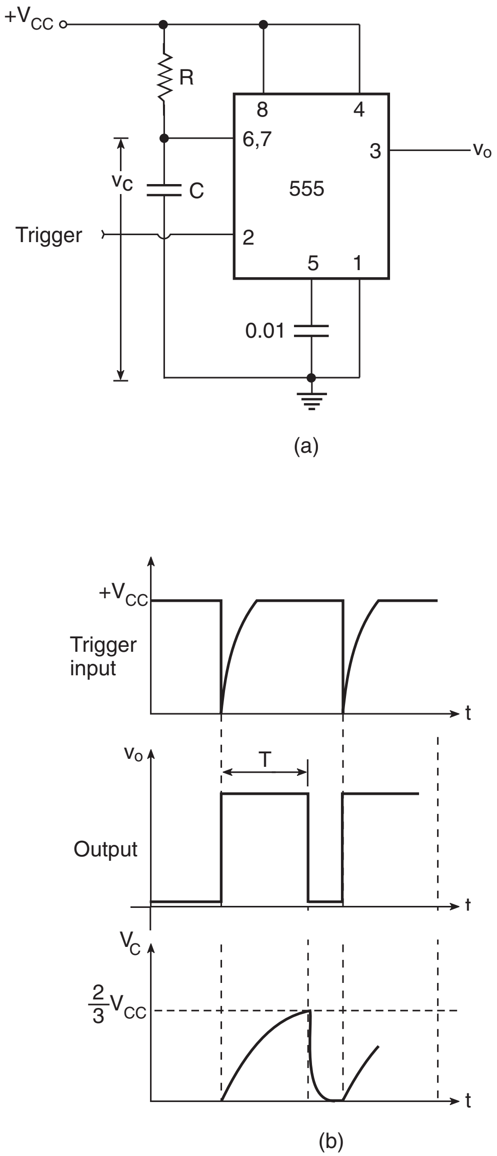
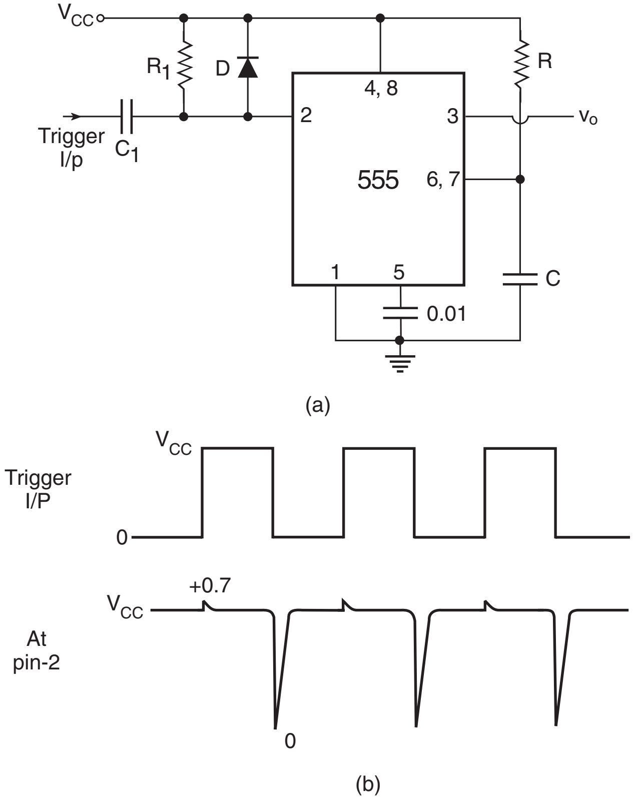


Figure 10.11 (a) Monostable multivibrator using timer 555 and (b) monostable multivibrator relevant waveforms.

Figure 10.11(b) shows the relevant waveforms for the circuit of Fig. 10.11(a).

It is often desirable to trigger a monostable multivibrator either on the trailing (HIGH-to-LOW) or leading (LOW-to-HIGH) edges of the trigger waveform. In order to achieve that, we will need an external circuit between the trigger waveform input and terminal 2 of timer 555. The external circuit ensures that terminal 2 of the IC gets the required trigger pulse corresponding to the desired edge of



(a)

(b)

Figure 10.12 555 monoshot triggering on trailing edges.

the trigger waveform. Figure 10.12(a) shows the monoshot configuration that can be triggered on the trailing edges of the trigger waveform. R_1–C_1 constitutes a differentiator circuit. One of the terminals of resistor R_1 is tied to $+V_{CC}$, with the result that the amplitudes of differentiated pulses are $+V_{CC}$ to $+2V_{CC}$ and $+V_{CC}$ to ground, corresponding to the leading and trailing edges of the trigger waveform respectively. Diode D clamps the positive-going differentiated pulses to about +0.7 V. The net result is that the trigger terminal of timer 555 gets the required trigger pulses corresponding to HIGH-to-LOW edges of the trigger waveform. Figure 10.12(b) shows the relevant waveforms.

Figure 10.13(a) shows the monoshot configuration that can be triggered on the leading edges of the trigger waveform. The R_1–C_1 combination constitutes the differentiator producing positive and negative pulses corresponding to LOW-to-HIGH and HIGH-to-LOW transitions of the trigger waveform. Negative pulses are clamped by the diode, and the positive pulses are applied to the base of a transistor switch. The collector terminal of the transistor feeds the required trigger pulses to terminal 2 of the IC. Figure 10.13(b) shows the relevant waveforms.

For the circuits shown in Figs 10.12 and 10.13 to function properly, the values of R_1 and C_1 for the differentiator should be chosen carefully. Firstly, the differentiator time constant should be much smaller than the HIGH time of the trigger waveform for proper differentiation. Secondly, the differentiated pulse width should be less than the expected HIGH time of the monoshot output.

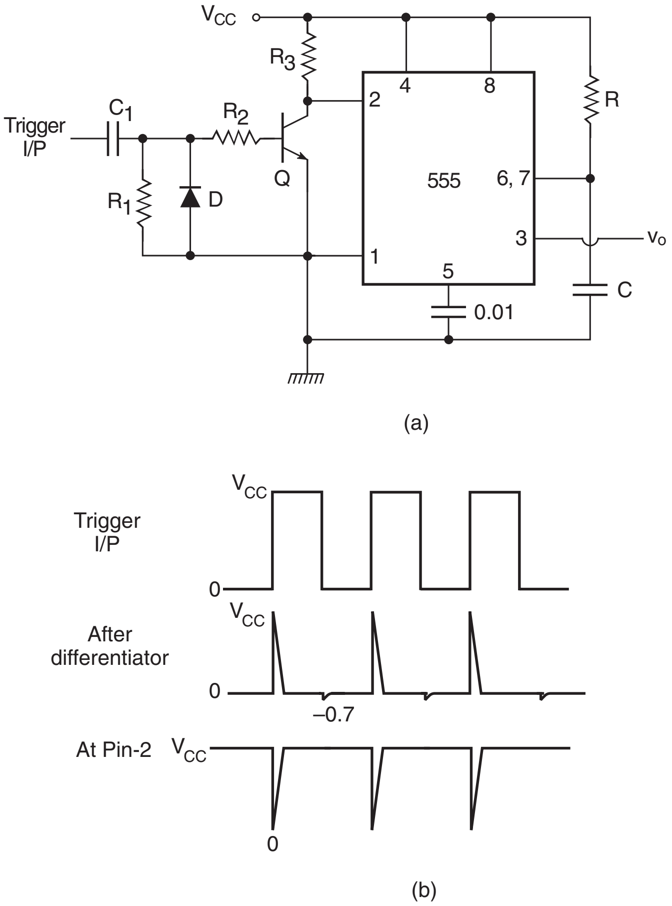


Figure 10.13 555 monoshot triggering on leading edges.

Example 10.1

The pulsed waveform of Fig. 10.14(b) is applied to the RESET terminal of the astable multivibrator circuit of Fig. 10.14(a). Draw the output waveform.

Solution

The circuit shown in Fig. 10.14(a) is an astable multivibrator with a 500 Hz symmetrical waveform applied to its RESET terminal. The RESET terminal is alternately HIGH and LOW for 1.0 ms. When the RESET input is LOW, the output is forced to the LOW state. When the RESET input is HIGH, an astable waveform appears at the output. The HIGH and LOW time periods of the astable multivibrator are determined as follows:

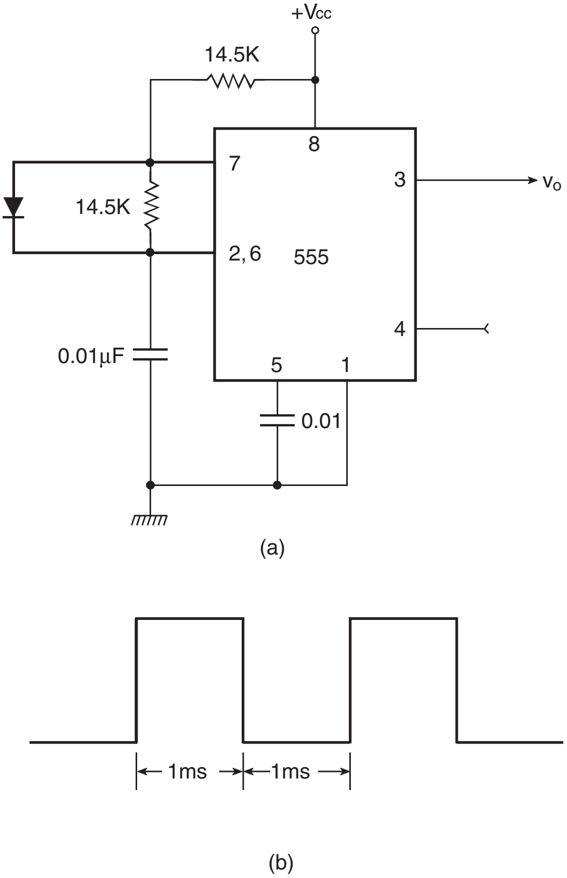


Figure 10.14 Example 10.1.

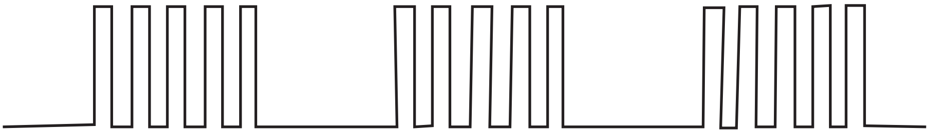


Figure 10.15 Solution to example 10.1.

$$\text{HIGH time} = 0.69 \times 14.5 \times 10^{3} \times 0.01 \times 10^{-6} = 100\,\mu\text{s}$$

$$\text{LOW time} = 0.69 \times 14.5 \times 10^{3} \times 0.01 \times 10^{-6} = 100\,\mu\text{s}$$

The astable output is thus a 5 kHz symmetrical waveform. Every time the RESET terminal goes to HIGH for 1.0 ms, five cycles of 5 kHz waveform appear at the output. Figure 10.15 shows the output waveform appearing at terminal 3 of the timer IC.

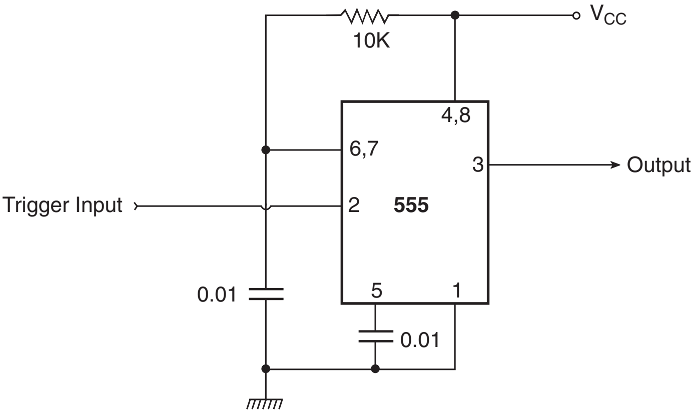


Figure 10.16 Example 10.2.

Example 10.2

Refer to the monostable multivibrator circuit in Fig. 10.16. The trigger terminal (pin 2 of the IC) is driven by a symmetrical pulsed waveform of 10 kHz. Determine the frequency and duty cycle of the output waveform.

Solution

- The frequency of the trigger waveform = 10 kHz.
- The time period between two successive leading or trailing edges = 100 μs.
- The expected pulse width of the monoshot output = $1.1RC = 1.1 \times 10^4 \times 10^{-8} = 110\,\mu\text{s}$.
- The trigger waveform is a symmetrical one; it has HIGH and LOW time periods of 50 μs each. Since the LOW-state time period of the trigger waveform is less than the expected output pulse width, it can successfully trigger the monoshot on its trailing edges.
- Since the time period between two successive trailing edges is 100 μs and the expected output pulse width is 110 μs, only alternate trailing edges of the trigger waveform will trigger the monoshot.
- The frequency of the output waveform = 10/2 = 5 kHz.
- The time period of the output waveform = $1/(5 \times 10^3) = 200$ μs.
- Therefore, the duty cycle of the output waveform = 110/200 = 0.55.

10.3 *R-S* Flip-Flop

A flip-flop, as stated earlier, is a bistable circuit. Both of its output states are stable. The circuit remains in a particular output state indefinitely until something is done to change that output status. Referring to the bistable multivibrator circuit discussed earlier, these two states were those of the output transistor in saturation (representing a LOW output) and in cut-off (representing a HIGH output). If the LOW and HIGH outputs are respectively regarded as ‘0’ and ‘1’, then the output can either be a ‘0’ or a ‘1’. Since either a ‘0’ or a ‘1’ can be held indefinitely until the circuit is appropriately triggered to go to the other state, the circuit is said to have memory. It is capable of storing one binary digit or one bit of digital information. Also, if we recall the functioning of the bistable multivibrator circuit, we find that, when one of the transistors was in saturation, the other was in cut-off. This implies that, if we had taken outputs from the collectors of both transistors, then the two outputs would be complementary. In the flip-flops of various types that are available in IC form, we will see that all these devices offer complementary outputs usually designated as Q and $\overline{Q}$.

The R-S flip-flop is the most basic of all flip-flops. The letters 'R' and 'S' here stand for RESET and SET. When the flip-flop is SET, its Q output goes to a '1' state, and when it is RESET it goes to a '0' state. The $\overline{Q}$ output is the complement of the Q output at all times.

10.3.1 R-S Flip-Flop with Active LOW Inputs

Figure 10.17(a) shows a NAND gate implementation of an R-S flip-flop with active LOW inputs. The two NAND gates are cross-coupled. That is, the output of NAND 1 is fed back to one of the inputs of NAND 2, and the output of NAND 2 is fed back to one of the inputs of NAND 1. The remaining inputs of NAND 1 and NAND 2 are the S and R inputs. The outputs of NAND 1 and NAND 2 are respectively Q and $\overline{Q}$ outputs.

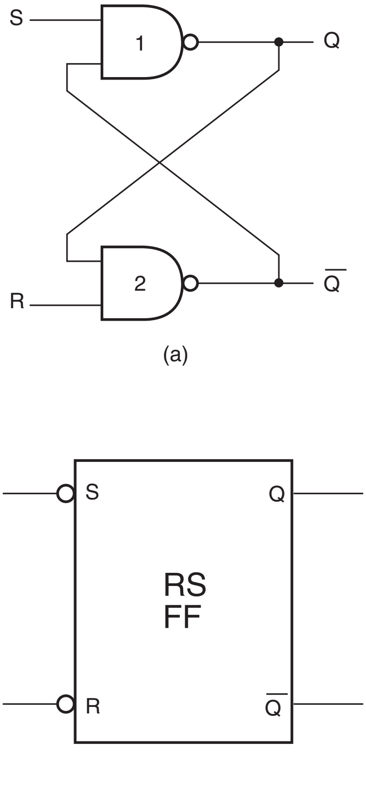
The fact that this configuration follows the function table of Fig. 10.17(c) can be explained. We will look at different entries of the function table, one at a time.

Let us take the case of $R = S = 1$ (the first entry in the function table). We will prove that, for $R = S = 1$, the Q output remains in its existing state. In the truth table, Q_n represents the existing state and Q_{n+1} represents the state of the flip-flop after it has been triggered by an appropriate pulse at the R or S input. Let us assume that $Q = 0$ initially. This '0' state fed back to one of the inputs of gate 2 ensures that $\overline{Q} = 1$. The '1' state of $\overline{Q}$ fed back to one of the inputs of gate 1 along with $S = 1$ ensures that $Q = 0$. Thus, $R = S = 1$ holds the existing stage. Now, if Q was initially in the '1' state and not the '0' state, this '1' fed back to one of the inputs of gate 2 along with $R = 1$ forces $\overline{Q}$ to be in the '0' state. The '0' state, when fed back to one of the inputs of gate 1, ensures that Q remains in its existing state of logic '1'. Thus, whatever the state of Q, $R = S = 1$ holds the existing state.

Let us now look at the second entry of the function table where $S = 0$ and $R = 1$. We can see that such an input combination forces the Q output to the '1' state. On similar lines, the input combination $S = 1$ and $R = 0$ (third entry of the truth table) forces the Q output to the '0' state. It would be interesting to analyse what happens when $S = R = 0$. This implies that both Q and $\overline{Q}$ outputs should go to the '1' state, as one of the inputs of a NAND gate being a logic '0' should force its output to the logic '1' state irrespective of the status of the other input. This is an undesired state as Q and $\overline{Q}$ outputs are to be the complement of each other. The input condition (i.e. $R = S = 0$) that causes such a situation is therefore considered to be an invalid condition and is forbidden. Figure 10.17(b) shows the logic symbol of such a flip-flop. The R and S inputs here have been shown as active LOW inputs, which is obvious as this flip-flop of Fig. 10.17(a) is SET (that is, $Q = 1$) when $S = 0$ and RESET (that is, $Q = 0$) when $R = 0$. Thus, R and S are active when LOW. The term CLEAR input is also used sometimes in place of RESET. The operation of the R-S flip-flop of Fig. 10.17(a) can be summarized as follows:

1. SET = RESET = 1 is the normal resting condition of the flip-flop. It has no effect on the output state of the flip-flop. Both Q and $\overline{Q}$ outputs remain in the logic state they were in prior to this input condition.
2. SET = 0 and RESET = 1 sets the flip-flop. Q and $\overline{Q}$ respectively go to the '1' and '0' state.
3. SET = 1 and RESET = 0 resets or clears the flip-flop. Q and $\overline{Q}$ respectively go to the '0' and '1' state.
4. SET = RESET = 0 is forbidden as such a condition tries to set (that is, $Q = 1$) and reset (that is, $\overline{Q} = 1$) the flip-flop at the same time. To be more precise, SET and RESET inputs in the R-S flip-flop cannot be active at the same time.

The R-S flip-flop of Fig. 10.17(a) is also referred to as an R-S latch. This is because any combination at the inputs immediately manifests itself at the output as per the truth table.



(a)

(b)

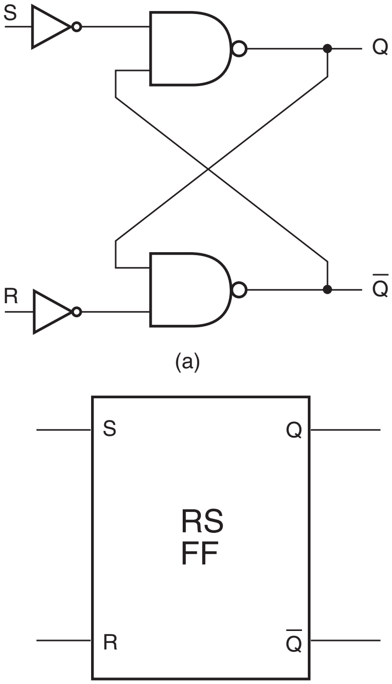
Operation Mode	S	R	Q_{n+1}
No change	1	1	Qn
SET	0	1	1
RESET	1	0	0
Forbidden	0	0	—

(c)

Figure 10.17 *R-S* flip-flop with active LOW inputs.

10.3.2 R-S Flip-Flop with Active HIGH Inputs

Figure 10.18(a) shows another NAND gate implementation of the *R-S* flip-flop. Figures 10.18(b) and (c) respectively show its circuit symbol and function table. Such a circuit would have active HIGH inputs. The input combination $R = S = 1$ would be forbidden as SET and RESET inputs in an *R-S* flip-flop cannot be active at the same time.



(a)

(b)

Operation Mode	S	R	Q_{n+1}
No change	0	0	Q_n
SET	1	0	1
RESET	0	1	0
Forbidden	1	1	—

(c)

Figure 10.18 *R-S* flip-flop with active HIGH inputs.

The *R-S* flip-flops (or latches) of Figs 10.17(a) and 10.18 (a) may also be implemented with NOR gates. The NOR gate counterparts of Fig. 10.17(a) and Fig. 10.18(a) are respectively shown in Figs 10.19(a) and (b).

So far we have discussed the operation of an *R-S* flip-flop with the help of its logic diagram and the function table on lines similar to the case of combinational circuits. We do, however, appreciate that a sequential circuit would be better explained if we expressed its output (immediately after it was clocked) in terms of its present output and its inputs. The function tables of Figs 10.17(c) and 10.18(c) may be redrawn as shown in Figs 10.20(a) and (b) respectively. This new form of representation is known as the characteristic table. Having done this, we could even write simplified Boolean expressions,

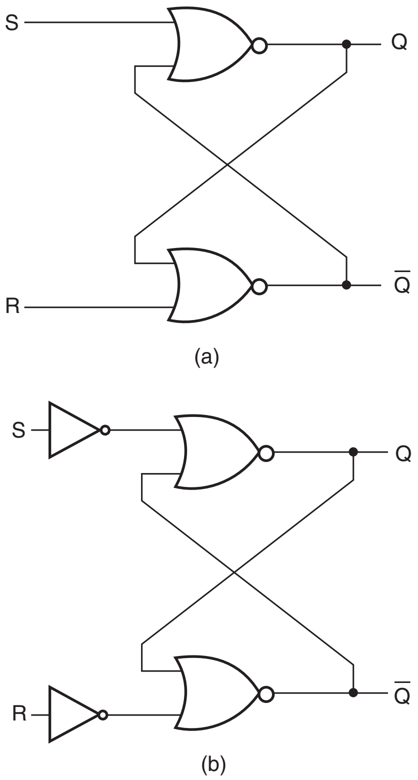


Figure 10.19 NOR implementation of an *R-S* flip-flop.

called characteristic equations, using any of the minimization techniques, such as Karnaugh mapping. The K-maps for the characteristic tables of Figs 10.20(a) and (b) are given in Figs 10.20(c) and (d) respectively. Characteristic equations for *R-S* flip-flops with active LOW and active HIGH inputs are given by the equations

$$Q_{n+1} = \overline{S} + R.Q_n \quad \text{and} \quad S + R = 1 \tag{10.15}$$

$$Q_{n+1} = S + \overline{R}.Q_n \quad \text{and} \quad S.R = 0 \tag{10.16}$$

$S + R = 1$ indicates that $R = S = 0$ is a prohibited entry. Similarly, $S.R = 0$ only indicates that $R = S = 1$ is a prohibited entry.

10.3.3 Clocked R-S Flip-Flop

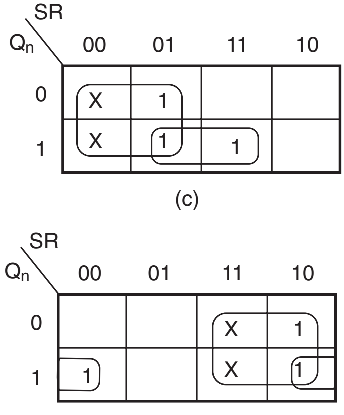
In the case of a clocked *R-S* flip-flop, or for that matter any clocked flip-flop, the outputs change states as per the inputs only on the occurrence of a clock pulse. The clocked flip-flop could be a level-triggered one or an edge-triggered one. The two types are discussed in the next section. For the

Qn	S	R	Q_{n+1}
0	0	0	Indeter
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	Indeter
1	0	1	1
1	1	0	0
1	1	1	1

(a)

Qn	S	R	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	Indeter
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	Indeter

(b)



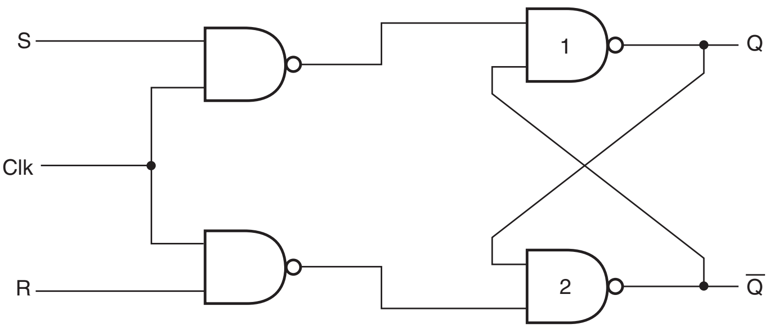
(c)

(d)

Figure 10.20 (a) Characteristic table of an *R*-*S* flip-flop with active LOW inputs, (b) the characteristic table of an *R*-*S* flip-flop with active HIGH inputs, (c) the K-map solution of an *R*-*S* flip-flop with active LOW inputs and (d) the K-map solution of an *R*-*S* flip-flop with active HIGH inputs.

time being, let us first see how the flip-flop of the previous section can be transformed into a clocked flip-flop. Figure 10.21(a) shows the logic implementation of a clocked flip-flop that has active HIGH inputs. The function table for the same is shown in Fig. 10.21(b) and is self-explanatory.

The basic flip-flop is the same as that shown in Fig. 10.17(a). The two NAND gates at the input have been used to couple the *R* and *S* inputs to the flip-flop inputs under the control of the clock signal. When the clock signal is HIGH, the two NAND gates are enabled and the *S* and *R* inputs are passed on to flip-flop inputs with their status complemented. The outputs can now change states as per the status of *R* and *S* at the flip-flop inputs. For instance, when $S = 1$ and $R = 0$ it will be passed on as 0 and 1 respectively when the clock is HIGH. When the clock is LOW, the two NAND gates produce a ‘1’ at their outputs, irrespective of the *S* and *R* status. This produces a logic ‘1’ at both inputs of the flip-flop, with the result that there is no effect on the output states. Figure 10.22(a) shows the clocked *R*-*S* flip-flop with active LOW *R* and *S* inputs. The logic implementation here is a modification of the basic *R*-*S* flip-flop in Fig. 10.18(a). The truth table of this flip-flop, as given in Fig. 10.22(b), is self-explanatory.

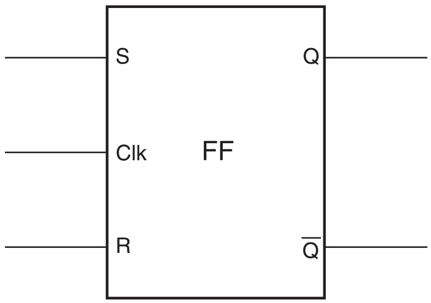


(a)

S	R	Clk	Q_{n+1}
0	0	0	Q_n
0	0	1	Q_n
0	1	0	Q_n
0	1	1	0
1	0	0	Q_n
1	0	1	1
1	1	0	Q_n
1	1	1	Invalid

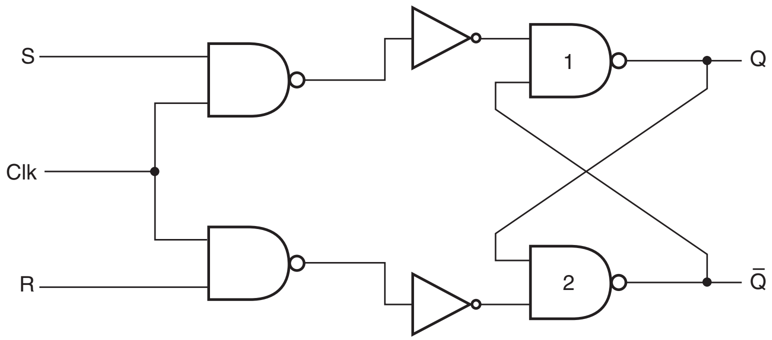
(b)

Figure 10.21 Clocked *R*-*S* flip-flop with active HIGH inputs.



(c)

Figure 10.21 (*continued*).



(a)

S	R	Clk	Q_{n+1}
0	0	0	Q_n
0	0	1	Invalid
0	1	0	Q_n
0	1	1	1
1	0	0	Q_n
1	0	1	0
1	1	0	Q_n
1	1	1	Q_n

(b)

Figure 10.22 Clocked *R-S* flip-flop with active LOW inputs.

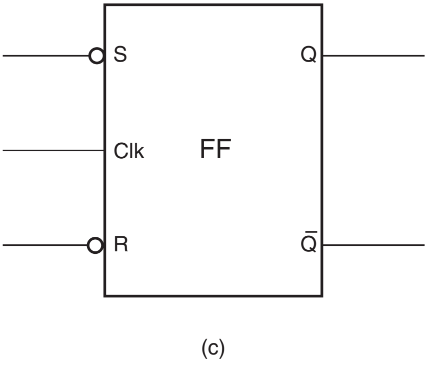


Figure 10.22 (*continued*).

10.4 Level-Triggered and Edge-Triggered Flip-Flops

In a *level-triggered* flip-flop, the output responds to the data present at the inputs during the time the clock pulse level is HIGH (or LOW). That is, any changes at the input during the time the clock is active (HIGH or LOW) are reflected at the output as per its function table. The clocked *R-S* flip-flop described in the preceding paragraphs is a level-triggered flip-flop that is active when the clock is HIGH.

In an *edge-triggered* flip-flop, the output responds to the data at the inputs only on LOW-to-HIGH or HIGH-to-LOW transition of the clock signal. The flip-flop in the two cases is referred to as positive edge triggered and negative edge triggered respectively. Any changes in the input during the time the clock pulse is HIGH (or LOW) do not have any effect on the output. In the case of an edge-triggered flip-flop, an edge detector circuit transforms the clock input into a very narrow pulse that is a few nanoseconds wide. This narrow pulse coincides with either LOW-to-HIGH or HIGH-to-LOW transition of the clock input, depending upon whether it is a positive edge-triggered flip-flop or a negative edge-triggered flip-flop. This pulse is so narrow that the operation of the flip–flop can be considered to have occurred on the edge itself.

Figure 10.23 shows the clocked *R-S* flip-flop of Fig. 10.21 with the edge detector block incorporated in the clock circuit. Figures 10.24 (a) and (b) respectively show typical edge detector circuits for positive

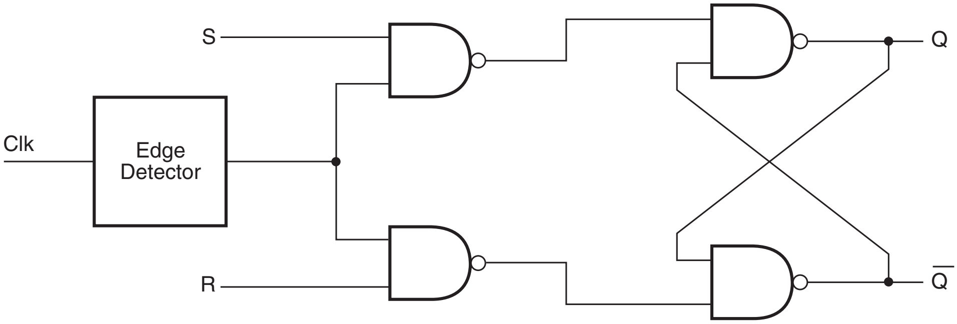


Figure 10.23 Edge-triggered *R-S* flip-flop.

and negative edge triggering. The width of the narrow pulse generated by this edge detector circuit is equal to the propagation delay of the inverter. Figure 10.25 shows the circuit symbol for the flip-flop of Fig. 10.23 for the positive edge-triggered mode [Fig. 10.25(a)] and the negative edge-triggered mode [Fig. 10.25(b)].

10.5 *J-K* Flip-Flop

A *J-K* flip-flop behaves in the same fashion as an *R-S* flip-flop except for one of the entries in the function table. In the case of an *R-S* flip-flop, the input combination $S = R = 1$ (in the case of a flip-flop with active HIGH inputs) and the input combination $S = R = 0$ (in the case of a flip-flop with active LOW inputs) are prohibited. In the case of a *J-K* flip-flop with active HIGH inputs, the output of the flip-flop toggles, that is, it goes to the other state, for $J = K = 1$. The output toggles for $J = K = 0$ in the case of the flip-flop having active LOW inputs. Thus, a *J-K* flip-flop overcomes the problem of a forbidden input combination of the *R-S* flip-flop. Figures 10.26(a) and (b) respectively show the circuit symbol of level-triggered *J-K* flip-flops with active HIGH and active LOW inputs, along with their function tables. Figure 10.27 shows the realization of a *J-K* flip-flop with an *R-S* flip-flop.

The characteristic tables for a *J-K* flip-flop with active HIGH *J* and *K* inputs and a *J-K* flip-flop with active LOW *J* and *K* inputs are respectively shown in Figs 10.28(a) and (b). The corresponding Karnaugh maps are shown in Fig. 10.28(c) for the characteristics table of Fig. 10.28(a) and in Fig. 10.28(d) for the characteristic table of Fig. 10.28(b). The characteristic equations for the Karnaugh maps of Figs 10.28(c) and (d) are respectively

$$Q_{n+1} = J.\overline{Q_n} + \overline{K}.Q_n \tag{10.17}$$

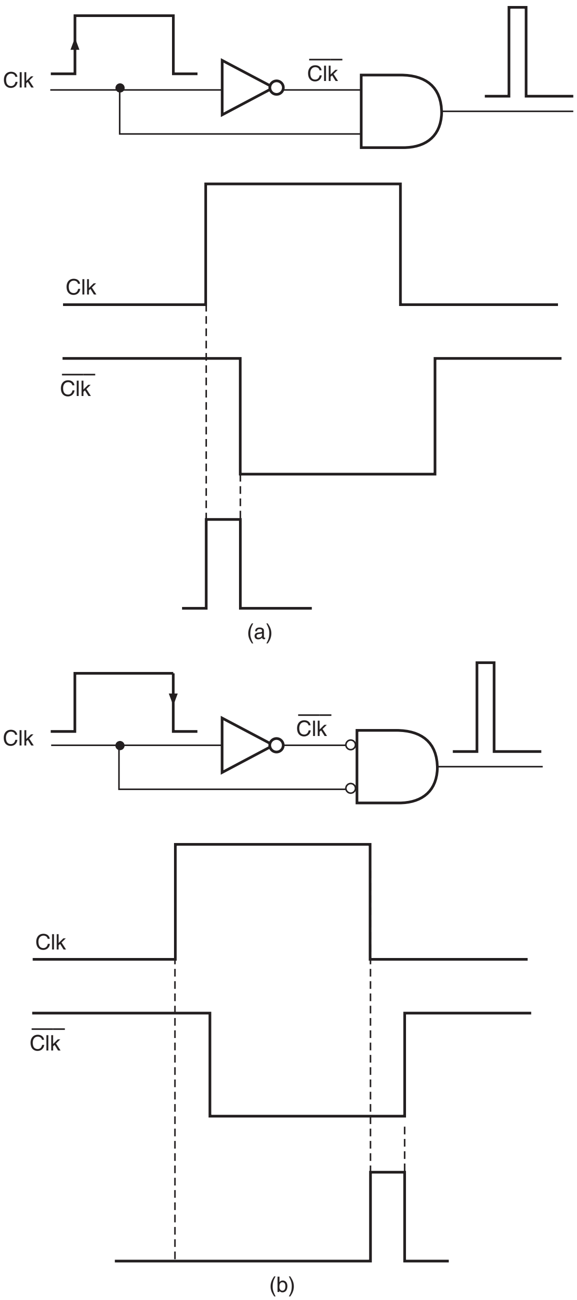
$$Q_{n+1} = \overline{J}.\overline{Q_n} + K.Q_n \tag{10.18}$$

10.5.1 J-K Flip-Flop with PRESET and CLEAR Inputs

It is often necessary to clear a flip-flop to a logic ‘0’ state ($Q_n = 0$) or preset it to a logic ‘1’ state ($Q_n = 1$). An example of how this is realized is shown in Fig. 10.29(a). The flip-flop is cleared (that is, $Q_n = 0$) whenever the CLEAR input is ‘0’ and the PRESET input is ‘1’. The flip-flop is preset to the logic ‘1’ state whenever the PRESET input is ‘0’ and the CLEAR input is ‘1’. Here, the CLEAR and PRESET inputs are active when LOW. Figure 10.29(b) shows the circuit symbol of this presettable, clearable, clocked *J-K* flip-flop. Figure 10.29(c) shows the function table of such a flip-flop. It is evident from the function table that, whenever the PRESET input is active, the output goes to the ‘1’ state irrespective of the status of the clock, *J* and *K* inputs. Similarly, when the flip-flop is cleared, that is, the CLEAR input is active, the output goes to the ‘0’ state irrespective of the status of the clock, *J* and *K* inputs. In a flip-flop of this type, both PRESET and CLEAR inputs should not be made active at the same time.

10.5.2 Master–Slave Flip-Flops

Whenever the width of the pulse clocking the flip-flop is greater than the propagation delay of the flip-flop, the change in state at the output is not reliable. In the case of edge-triggered flip-flops, this pulse width would be the trigger pulse width generated by the edge detector portion of the flip-flop



(a)

(b)

Figure 10.24 (a) Positive edge-triggered edge detector circuits and (b) negative edge-triggered edge detector circuits.

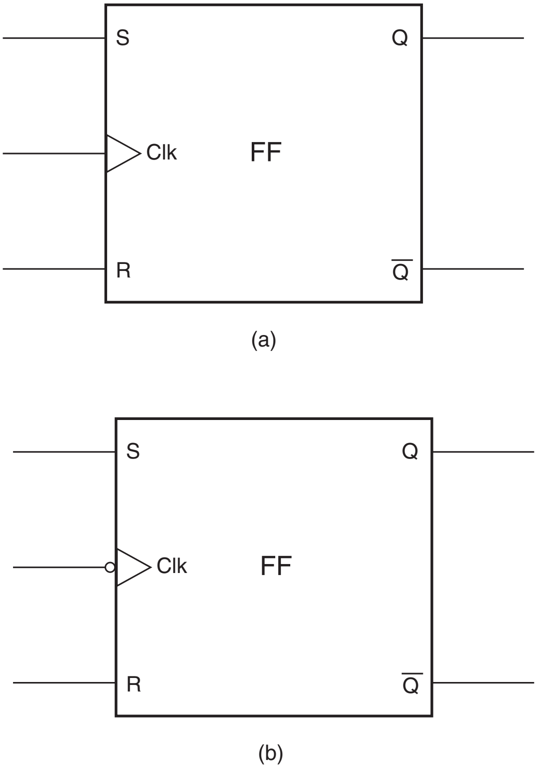
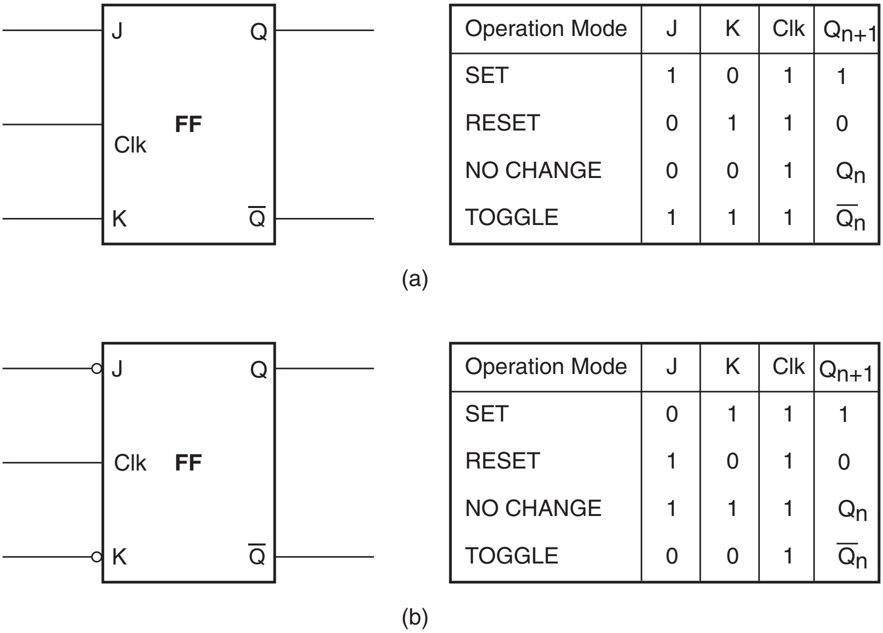


Figure 10.25 (a) Circuit symbol of a positive edge-triggered *R-S* flip-flop and (b) the circuit symbol of a negative edge-triggered *R-S* flip-flop.

and not the pulse width of the input clock signal. This phenomenon is referred to as the *race problem*. As the propagation delays are normally very small, the likelihood of the occurrence of a race condition is reasonably high. One way to get over this problem is to use a *master–slave* configuration. Figure 10.30(a) shows a master–slave flip-flop constructed with two *J-K* flip-flops. The first flip-flop is called the master flip-flop and the second is called the slave. The clock to the slave flip-flop is the complement of the clock to the master flip-flop. When the clock pulse is present, the master flip-flop is enabled while the slave flip-flop is disabled. As a result, the master flip-flop can change state while the slave flip-flop cannot. When the clock goes LOW, the master flip-flop gets disabled while the slave flip-flop is enabled. Therefore, the slave *J-K* flip-flop changes state as per the logic states at its *J* and *K* inputs. The contents of the master flip-flop are therefore transferred to the slave flip-flop, and the master flip-flop, being disabled, can acquire new inputs without affecting the output. As would be clear from the description above, a master–slave flip-flop is a pulse-triggered flip-flop and not an edge-triggered one. Figure 10.30(b) shows the truth table of a master–slave *J-K* flip-flop with active LOW PRESET and CLEAR inputs and active HIGH *J* and *K* inputs. The master–slave configuration has become obsolete. The newer IC technologies such as 74LS, 74AS, 74ALS, 74HC and 74HCT do not have master–slave flip-flops in their series.



Operation Mode	J	K	Clk	Q_{n+1}
SET	1	0	1	1
RESET	0	1	1	0
NO CHANGE	0	0	1	Q_n
TOGGLE	1	1	1	$\overline{Q}_n$

(a)

Operation Mode	J	K	Clk	Q_{n+1}
SET	0	1	1	1
RESET	1	0	1	0
NO CHANGE	1	1	1	Q_n
TOGGLE	0	0	1	$\overline{Q}_n$

(b)

Figure 10.26 (a) *J-K* flip-flop active HIGH inputs and (b) *J-K* flip-flop active LOW inputs.

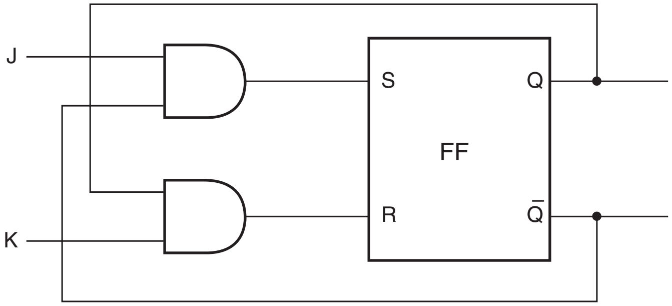


Figure 10.27 Realization of a *J-K* flip-flop using an *R-S* flip-flop.

Example 10.3

Draw the circuit symbol of the flip-flop represented by the function table of Fig. 10.31(a).

Solution

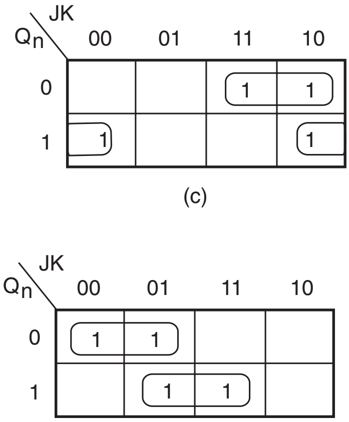
The first three entries of the function table indicate that the *J-K* flip-flop has active HIGH PRESET and CLEAR inputs. Referring to the fourth and fifth entries of the function table, it has active LOW *J* and *K* inputs. The seventh row of the function table confirms this. The output responds to positive (LOW-to-HIGH) edges of the clock input. Thus, the flip-flop represented by the given function table is a presettable, clearable, positive edge-triggered flip-flop with active HIGH PRESET and CLEAR

Q_n	J	K	Q_{n+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

(a)

Q_n	J	K	Q_{n+1}
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

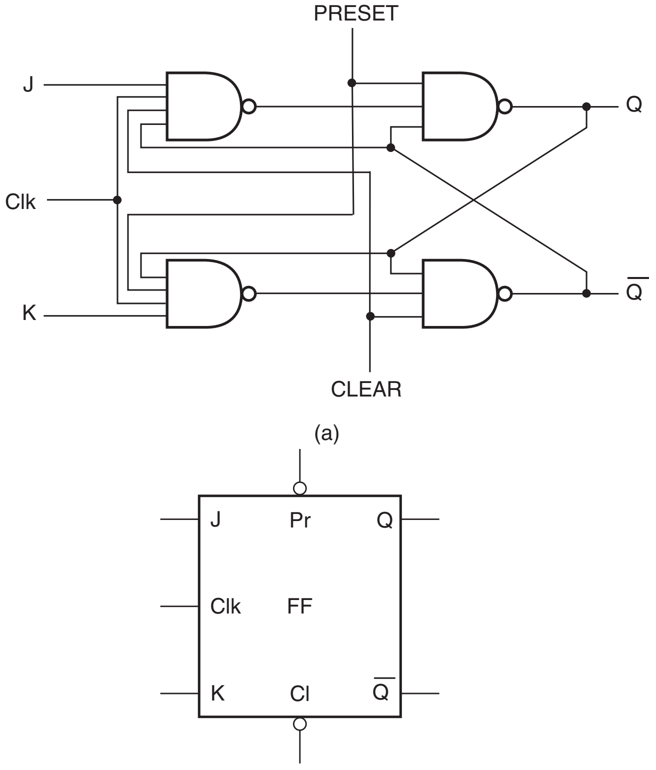
(b)



(c)

(d)

Figure 10.28 (a) Characteristic table of a *J*-*K* flip-flop with active HIGH inputs, (b) the characteristic table of a *J*-*K* flip-flop with active LOW inputs, (c) the K-map solution of a *J*-*K* flip-flop with active HIGH inputs and (d) the K-map solution of a *J*-*K* flip-flop with active LOW inputs.



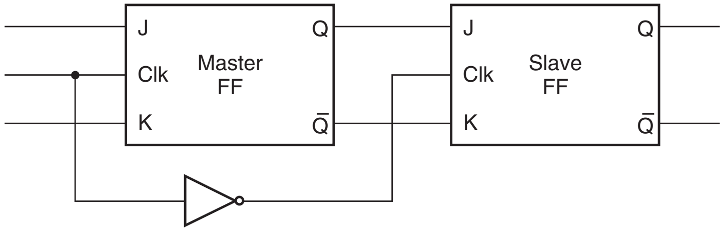
(a)

(b)

PR	CL	CLK	J	K	Q_{n+1}	$\overline{Q_{n+1}}$
0	1	X	X	X	1	0
1	0	X	X	X	0	1
0	0	X	X	X	---	---
1	1	1	0	0	Q_n	$\overline{Q}_n$
1	1	1	1	0	1	0
1	1	1	0	1	0	1
1	1	1	1	1	Toggle	
1	1	0	X	X	Q_n	$\overline{Q}_n$

(c)

Figure 10.29 *J-K* flip-flop with PRESET and CLEAR inputs.



(a)

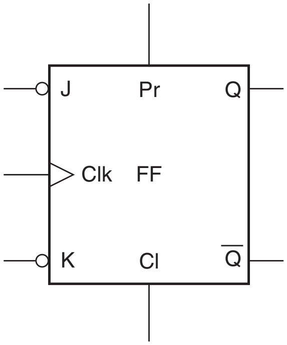
PR	CLR	CLK	J	K	Q_{n+1}	$\overline{Q_{n+1}}$
0	1	X	X	X	1	0
1	0	X	X	X	0	1
0	0	X	X	X	Unstable	
1	1	⎍	0	0	Q_n	$\overline{Q_n}$
1	1	⎍	1	0	1	0
1	1	⎍	0	1	0	1
1	1	⎍	1	1	Toggle	

(b)

Figure 10.30 Master–slave flip-flop.

PR	CLR	CLK	J	K	Q_{n+1}	$\overline{Q_{n+1}}$
1	0	X	X	X	1	0
0	1	X	X	X	0	1
1	1	X	X	X	Unstable	
0	0	↑	0	1	1	0
0	0	↑	1	0	0	1
0	0	↑	1	1	Q_n	$\overline{Q_n}$
0	0	↑	0	0	Toggle	

(a)



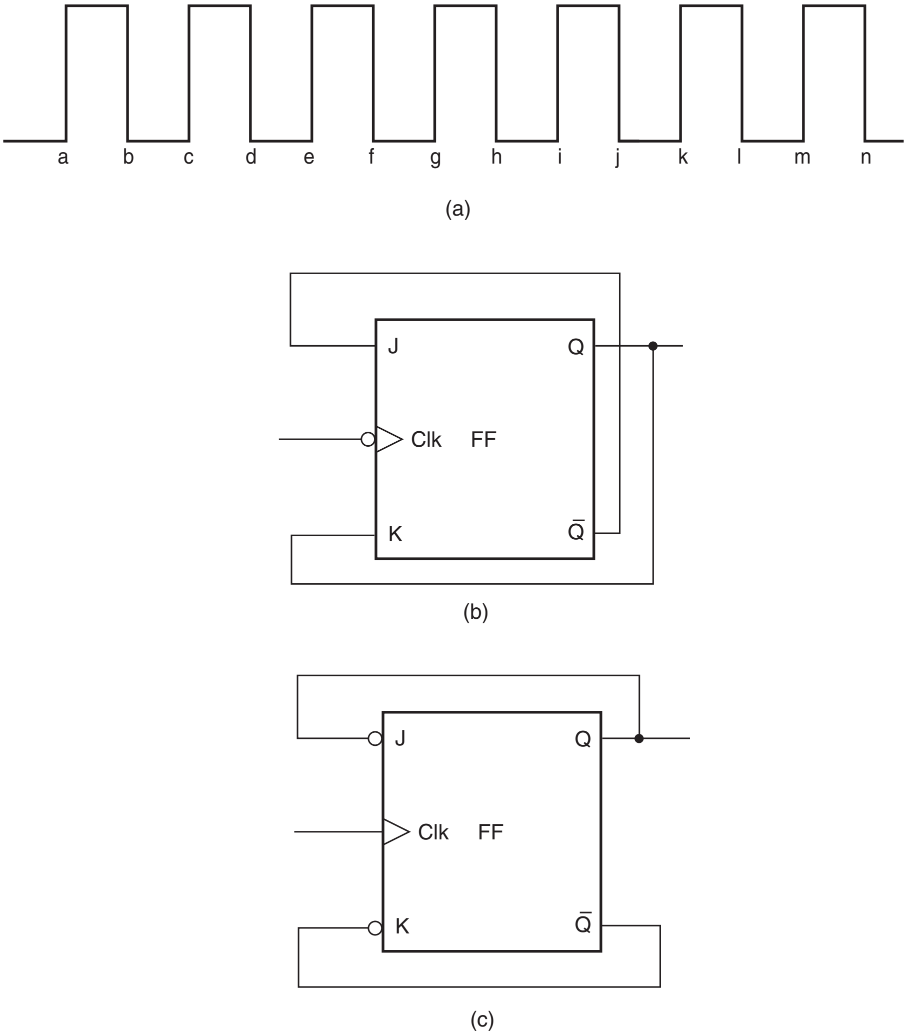
(b)

Figure 10.31 Example 10.3.

and active LOW J and K inputs. Figure 10.31(b) shows the circuit symbol of the flip-flop represented by this truth table.

Example 10.4

The 100 kHz square waveform of Fig. 10.32(a) is applied to the clock input of the flip-flops shown in Figs. 10.32(b) and (c). If the Q output is initially '0', draw the Q output waveform in the two cases. Also, determine the frequency of the Q output in the two cases.

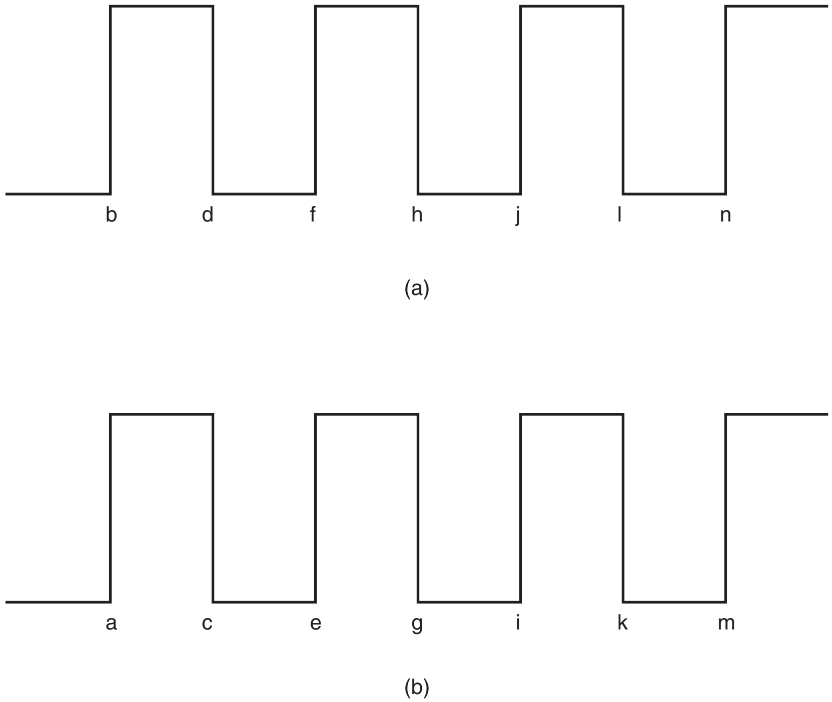


(a)

(b)

(c)

Figure 10.32 Example 10.4.



(a)

(b)

Figure 10.33 Solution to example 10.4.

Solution

Refer to the flip-flop of Fig. 10.32(b). Q is initially ‘0’. This makes the J and K inputs be initially ‘1’ and ‘0’ respectively. With the first trailing edge of the clock input, Q goes to the ‘1’ state. Thus, J and K acquire a logic status of ‘0’ and ‘1’ respectively. With the next trailing edge of the clock input, Q goes to logic ‘0’. This process continues, and Q alternately becomes ‘1’ and ‘0’. The Q output waveform for this case is shown in Fig. 10.33(a). In the case of the flip-flop of Fig. 10.32(c), J and K are initially ‘0’ and ‘1’ respectively. Thus, J is active. With the first leading edge of the clock input, Q and therefore J go to the logic ‘1’ state. The second leading edge forces Q to go to the logic ‘0’ state as now it is the K input that is in the logic ‘0’ state and active. This circuit also behaves in the same way as the flip-flop of Fig. 10.32(b). The output goes alternately to the logic ‘0’ and ‘1’ state. However, the transitions occur on the leading edge of the clock input. Figure 10.33(b) shows the Q output waveform for this case. The frequency of the Q output waveform in the two cases is equal to half the frequency of the clock input, for obvious reasons, and is therefore 50 kHz.

10.6 Toggle Flip-Flop (*T* Flip-Flop)

The output of a *toggle flip-flop*, also called a T flip-flop, changes state every time it is triggered at its T input, called the toggle input. That is, the output becomes ‘1’ if it was ‘0’ and ‘0’ if it was ‘1’.

Figures 10.34(a) and (b) respectively show the circuit symbols of positive edge-triggered and negative edge-triggered *T* flip-flops, along with their function tables.

If we consider the *T* input as active when HIGH, the characteristic table of such a flip-flop is shown in Fig. 10.34(c). If the *T* input were active when LOW, then the characteristic table would be as shown in Fig. 10.34(d). The Karnaugh maps for the characteristic tables of Figs 10.34(c) and (d) are shown in Figs 10.34(e) and (f) respectively. The characteristic equations as written from the Karnaugh maps are as follows:

$$Q_{n+1} = T.\overline{Q_n} + \overline{T}.Q_n \tag{10.19}$$

$$Q_{n+1} = \overline{T}.\overline{Q_n} + T.Q_n \tag{10.20}$$

It is obvious from the operational principle of the *T* flip-flop that the frequency of the signal at the *Q* output is half the frequency of the signal applied at the *T* input. A cascaded arrangement of *nT* flip-flops, where the output of one flip-flop is connected to the *T* input of the following flip-flop, can be used to divide the input signal frequency by a factor of 2^n. Figure 10.35 shows a divide-by-16 circuit built around a cascaded arrangement of four *T* flip-flops.

10.6.1 J-K Flip-Flop as a Toggle Flip-Flop

If we recall the function table of a *J-K* flip-flop, we will see that, when both *J* and *K* inputs of the flip-flop are tied to their active level (‘1’ level if *J* and *K* are active when HIGH, and ‘0’ level when *J* and *K* are active when LOW), the flip-flop behaves like a toggle flip-flop, with its clock input serving as the *T* input. In fact, the *J-K* flip-flop can be used to construct any other flip-flop. That is why it is also sometimes referred to as a *universal flip-flop*. Figure 10.36 shows the use of a *J-K* flip-flop as a *T* flip-flop.

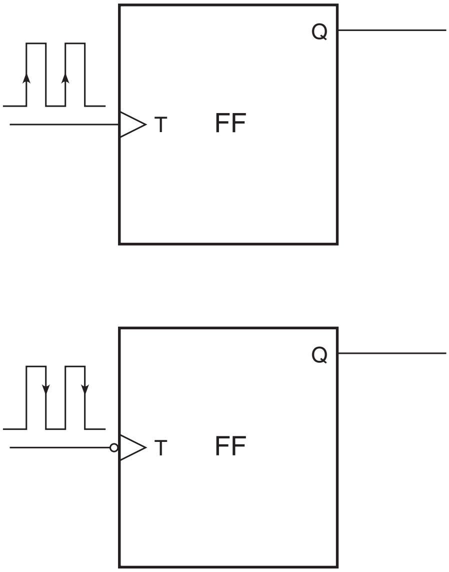
Example 10.5

Refer to the cascaded arrangement of two T flip-flops in Fig. 10.37(a). Draw the Q output waveform for the given input signal. If the time period of the input signal is 10 ms, find the frequency of the output signal? If, in the flip-flop arrangement of Fig. 10.37(a), FF-2 were positive edge triggered, draw the Q output waveform.

Solution

The *Q* output waveform is shown in Fig. 10.37(b) along with the *Q* output of FF-1. The output of the first *T* flip-flop changes state for every negative-going edge of the input clock waveform. Its frequency is therefore half the input signal frequency. The output of the first flip-flop acts as the clock input for the second *T* flip-flop in the cascade arrangement. The second flip-flop, too, toggles for every negative-going edge of the waveform appearing at its input. The final output thus has a frequency that is one-fourth of the input signal frequency:

- Now the time period of the input signal = 10 ms.
- Therefore, the frequency = 100 kHz.
- The frequency of the output signal = 25 kHz.



T	Q_n	Q_{n+1}
↑	0	1
↑	1	0

(a)

T	Q_n	Q_{n+1}
↓	0	1
↓	1	0

(b)

Q_n	T	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

(c)

Q_n	T	Q_{n+1}
0	0	1
0	1	0
1	0	0
1	1	1

(d)

Figure 10.34 (a) Positive edge-triggered toggle flip-flop, (b) a negative edge-triggered toggle flip-flop, (c, d) characteristic tables of level-triggered toggle flip-flops and (e, f) Karnaugh maps for characteristic tables (c, d).

Q_n \ T	0	1
0		1
1	1	

(e)

Q_n \ T	0	1
0	1	
1		1

(f)

Figure 10.34 (*continued*).

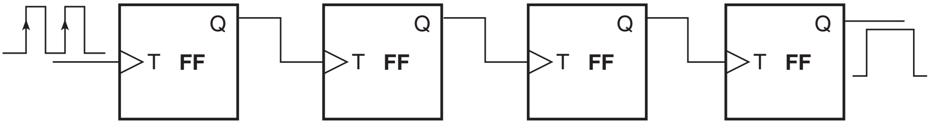


Figure 10.35 Cascade arrangement of *T* flip-flops.

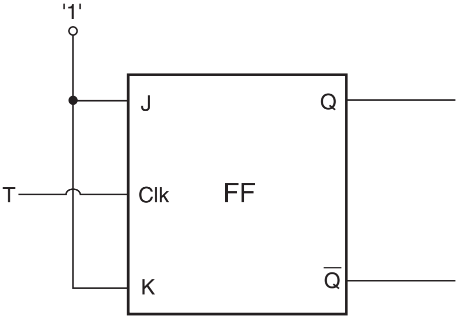


Figure 10.36 *J*-*K* flip-flop as a *T* flip-flop.

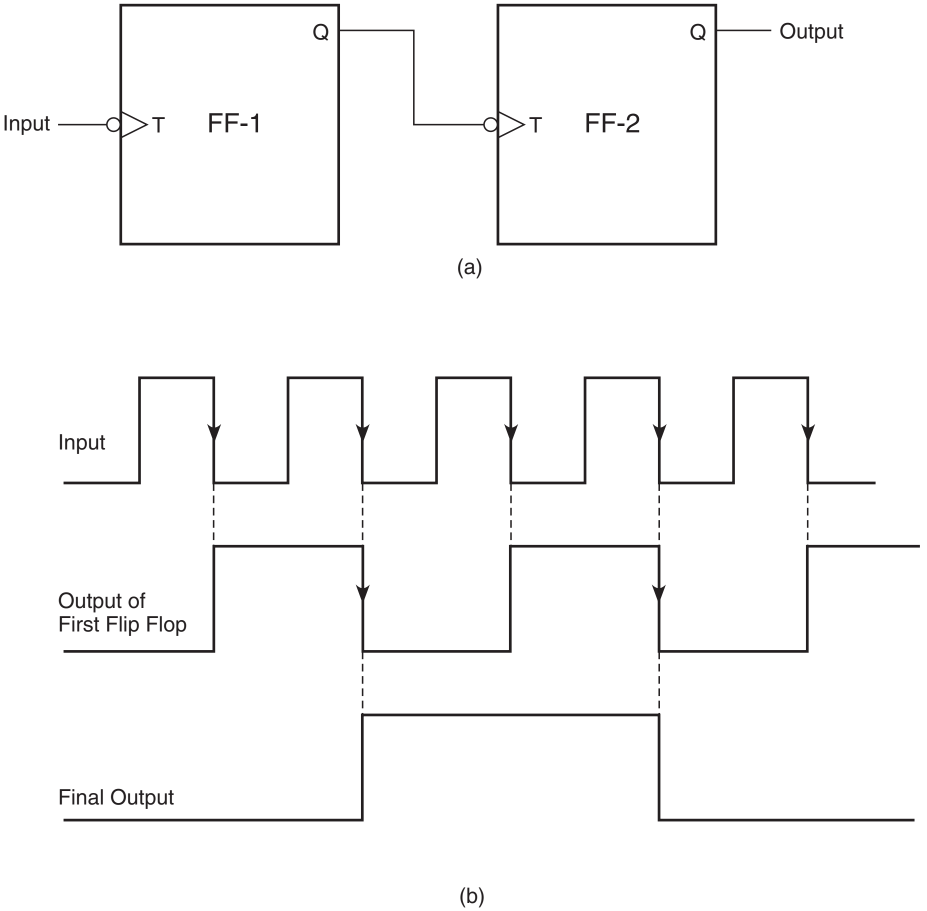


Figure 10.37 Example 10.5.

When the second flip-flop (FF-2) is a positive edge-triggered one, it will respond to the LOW-to-HIGH edges of the waveform appearing at its T input, which is the waveform appearing at the Q output of FF-1. The relevant waveforms in this case are shown in Fig. 10.38.

10.7 *D* Flip-Flop

A D flip-flop, also called a *delay flip-flop*, can be used to provide temporary storage of one bit of information. Figure 10.39(a) shows the circuit symbol and function table of a negative edge-triggered D flip-flop. When the clock is active, the data bit (0 or 1) present at the D input is transferred to the output. In the D flip-flop of Fig. 10.39, the data transfer from D input to Q output occurs on the negative-going (HIGH-to-LOW) transition of the clock input. The D input can acquire new status

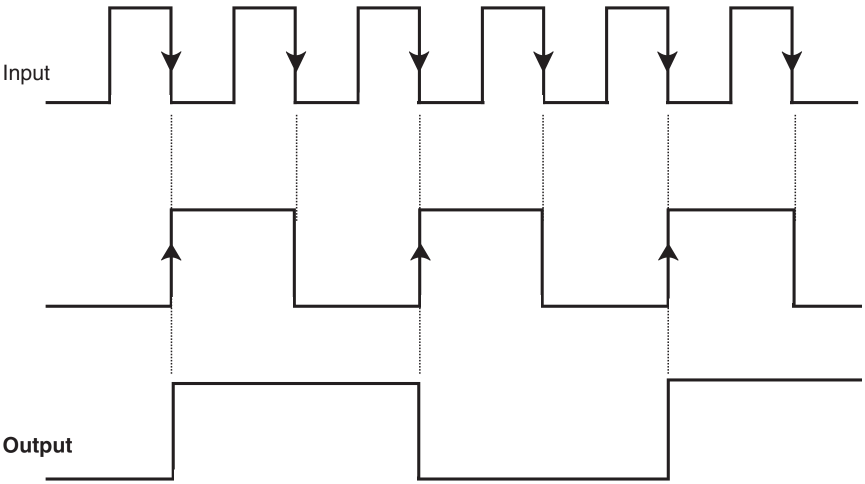


Figure 10.38 Example 10.5.

when the clock is inactive, which is the time period between successive HIGH-to-LOW transitions. The *D* flip-flop can provide a maximum delay of one clock period.

The characteristic table and the corresponding Karnaugh map for the *D* flip-flop of Fig. 10.39(a) are shown in Figs 10.39(c) and (d) respectively. The characteristic equation is as follows:

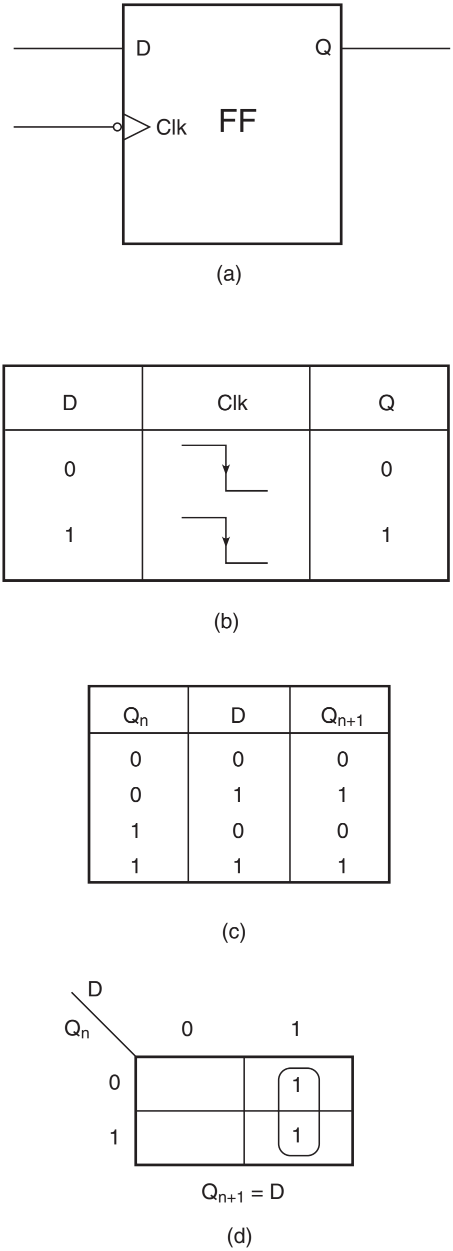
$$Q_{n+1} = D \tag{10.21}$$

10.7.1 J-K Flip-Flop as D Flip-Flop

Figure 10.40 shows how a *J-K* flip-flop can be used as a *D* flip-flop. When the *D* input is a logic '1', the *J* and *K* inputs are a logic '1' and '0' respectively. According to the function table of the *J-K* flip-flop, under these input conditions, the *Q* output will go to the logic '1' state when clocked. Also, when the *D* input is a logic '0', the *J* and *K* inputs are a logic '0' and '1' respectively. Again, according to the function table of the *J-K* flip-flop, under these input conditions, the *Q* output will go to the logic '0' state when clocked. Thus, in both cases, the *D* input is passed on to the output when the flip-flop is clocked.

10.7.2 D Latch

In a *D* latch, the output *Q* follows the *D* input as long as the clock input (also called the ENABLE input) is HIGH or LOW, depending upon the clock level to which it responds. When the ENABLE input goes to the inactive level, the output holds on to the logic state it was in just prior to the ENABLE input becoming inactive during the entire time period the ENABLE input is inactive.



(a)

D	Clk	Q
0	↓	0
1	↓	1

(b)

Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1

(c)

$Q_{n+1} = D$

(d)

Figure 10.39 *D* flip-flop.

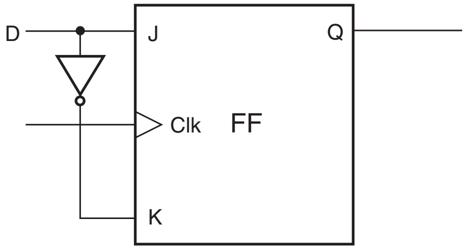


Figure 10.40 *J-K* flip-flop as a *D* flip-flop.

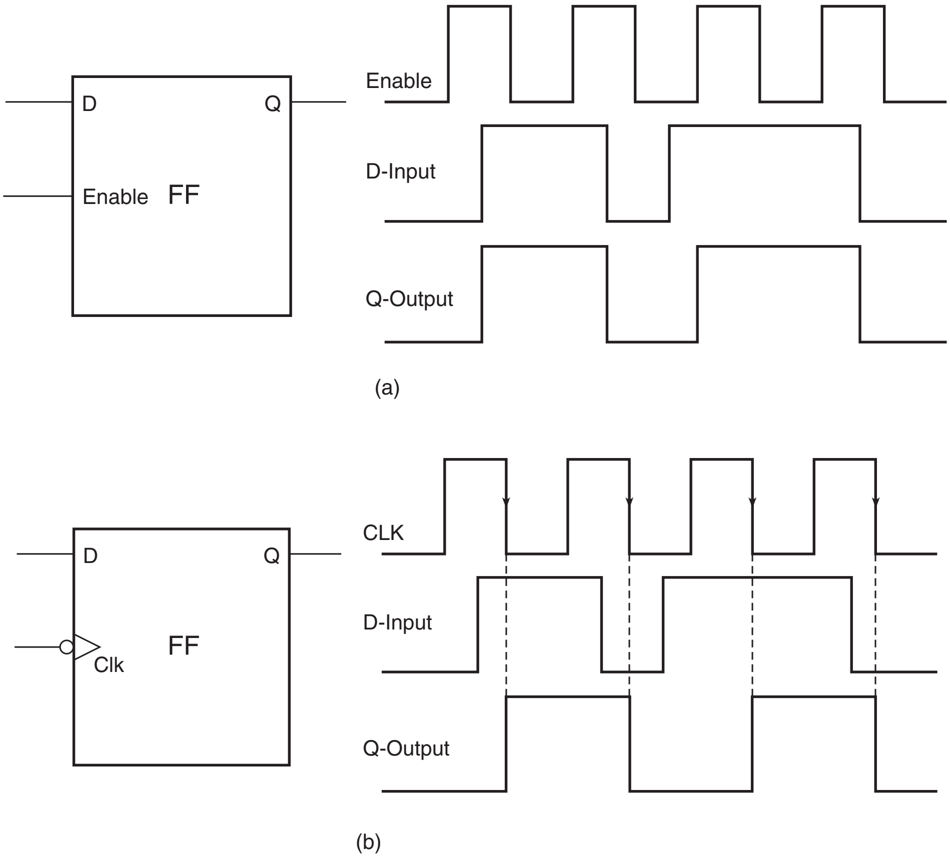


Figure 10.41 Comparison between a *D*-type latch and a *D* flip-flop.

A *D* flip-flop should not be confused with a *D* latch. In a *D* flip-flop, the data on the *D* input are transferred to the *Q* output on the positive- or negative-going transition of the clock signal, depending upon the flip-flop, and this logic state is held at the output until we get the next effective clock transition. The difference between the two is further illustrated in Figs 10.41(a) and (b) depicting the functioning of a *D* latch and a *D* flip-flop respectively.

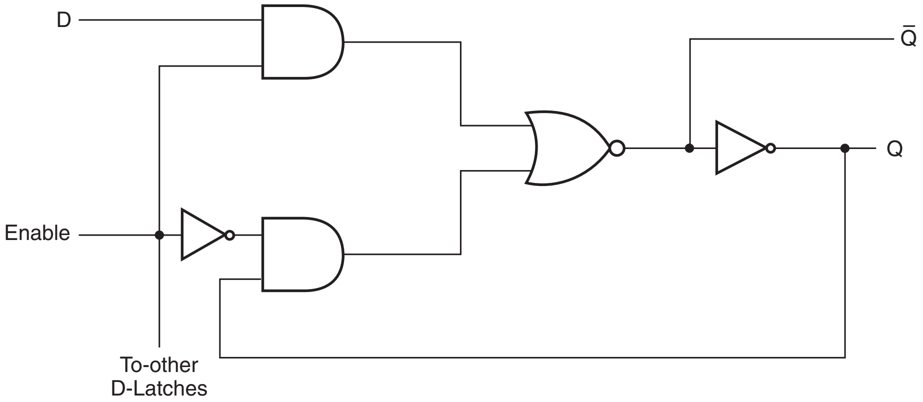


Figure 10.42 Example 10.6.

Example 10.6

Figure 10.42 shows the internal logic circuit diagram of one of the four D latches of a four-bit D latch in IC 7475. (a) Give an argument to prove that the Q output will track the D input only when the ENABLE input is HIGH. (b) Also, prove that the Q output holds the value it had just before the ENABLE input went LOW during the time the ENABLE input is LOW.

Solution

(a) When the ENABLE input is HIGH, the upper AND gate is enabled while the lower AND gate is disabled. The outputs of the upper and lower AND gates are D and logic ‘0’ respectively. They constitute inputs of the NOR gate whose output is $\overline{D}$. The Q output is therefore D.

(b) When the ENABLE input goes LOW, the upper AND gate is disabled (with its output going to logic ‘0’) and the lower AND gate is enabled (with its output becoming the same as the Q output owing to the feedback). The NOR gate output in this case is $\overline{Q}$, which means that the Q output holds its state as long as the ENABLE input is LOW.

10.8 Synchronous and Asynchronous Inputs

Most flip-flops have both synchronous and asynchronous inputs. Synchronous inputs are those whose effect on the flip-flop output is synchronized with the clock input. R, S, J, K and D inputs are all synchronous inputs. Asynchronous inputs are those that operate independently of the synchronous inputs and the input clock signal. These are in fact override inputs as their status overrides the status of all synchronous inputs and also the clock input. They force the flip-flop output to go to a predefined state irrespective of the logic status of the synchronous inputs. PRESET and CLEAR inputs are examples of asynchronous inputs. When active, the PRESET and CLEAR inputs place the flip-flop Q output in the ‘1’ and ‘0’ state respectively. Usually, these are active LOW inputs. When it is desired that the flip-flop functions as per the status of its synchronous inputs, the asynchronous inputs are kept in their inactive state. Also, both asynchronous inputs, if available on a given flip-flop, are not made active simultaneously.

10.9 Flip-Flop Timing Parameters

Certain timing parameters would be listed in the specification sheet of a flip-flop. Some of these parameters, as we will see in the paragraphs to follow, are specific to the logic family to which the flip-flop belongs. There are some parameters that have different values for different flip-flops belonging to the same broad logic family. It is therefore important that one considers these timing parameters before using a certain flip-flop in a given application. Some of the important ones are set-up and hold times, propagation delay, clock pulse HIGH and LOW times, asynchronous input active pulse width, clock transition time and maximum clock frequency.

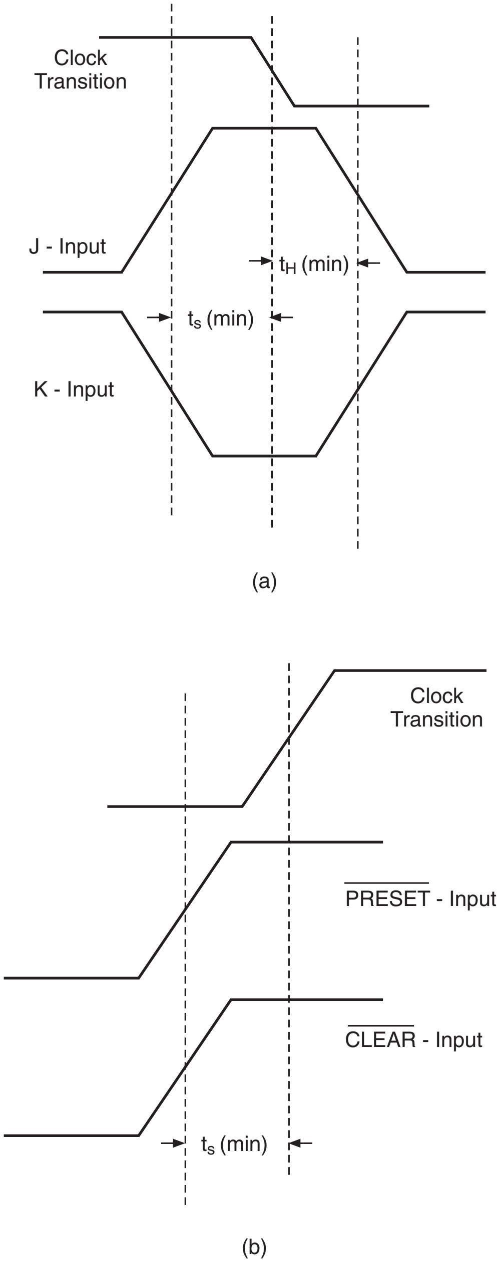
10.9.1 Set-Up and Hold Times

The *set-up time* is the minimum time period for which the synchronous inputs (for example, R, S, J, K and D) and asynchronous inputs (for example, PRESET and CLEAR) must be stable prior to the active clock transition for the flip-flop output to respond reliably at the clock transition. It is usually denoted by t_s (min) and is usually defined separately for synchronous and asynchronous inputs. As an example, if in a J-K flip-flop the J and K inputs were to go to '1' and '0' respectively, and if the flip-flop were negative edge triggered, the set-up time would be as shown in Fig. 10.43(a). The set-up time in the case of 74ALS109A, which is a dual J-K positive edge-triggered flip-flop belonging to the advanced low-power Schottky TTL logic family, is 15 ns. Also, the asynchronous inputs, such as PRESET and CLEAR, if there, should be inactive prior to the clock transition for a certain minimum time period if the outputs have to respond as per synchronous inputs. In the case of 74ALS109A, the asynchronous input set-up time is 10 ns. The asynchronous input set-up time for active low PRESET and CLEAR inputs is shown in Fig. 10.43(b), assuming a positive edge-triggered flip-flop.

The *hold time* t_H (min) is the minimum time period for which the synchronous inputs (R, S, J, K, D) must remain stable in the desired logic state after the active clock transition for the flip-flop to respond reliably. The same is depicted in Fig. 10.43(a) if the desired logic status for J and K inputs is '1' and '0' respectively and the flip-flop is negative edge triggered. The hold time for flip-flop 74ALS109A is specified to be zero. To sum up, for a flip-flop to respond properly and reliably at the active clock transition, the synchronous inputs must be stable in their intended logic states and the asynchronous inputs must be stable in their inactive states for at least a time period equal to the specified minimum set-up times prior to the clock transition, and the synchronous inputs must be stable for a time period equal to at least the specified minimum hold time after the clock transition.

10.9.2 Propagation Delay

There is always a time delay, known as the *propagation delay*, from the time instant the signal is applied to the time the output makes the intended change. The flip-flop data sheet usually specifies propagation delays for both HIGH-to-LOW (t_{pHL}) and for LOW-to-HIGH (t_{pLH}) output transitions. The propagation delay is measured between 50 % points on input and output waveforms and is usually specified for all types of input including synchronous and asynchronous inputs. The propagation delays for LOW-to-HIGH and HIGH-to-LOW output transitions for a positive edge-triggered flip-flop are shown in Fig. 10.44. For flip-flop 74ALS109A, t_{pHL} and t_{pLH} for clock input to output are respectively 18 and 16 ns. The same for the asynchronous input to output for this flip-flop are 15 and 13 ns respectively.



(a)

(b)

Figure 10.43 Set-up and hold times of a flip-flop.

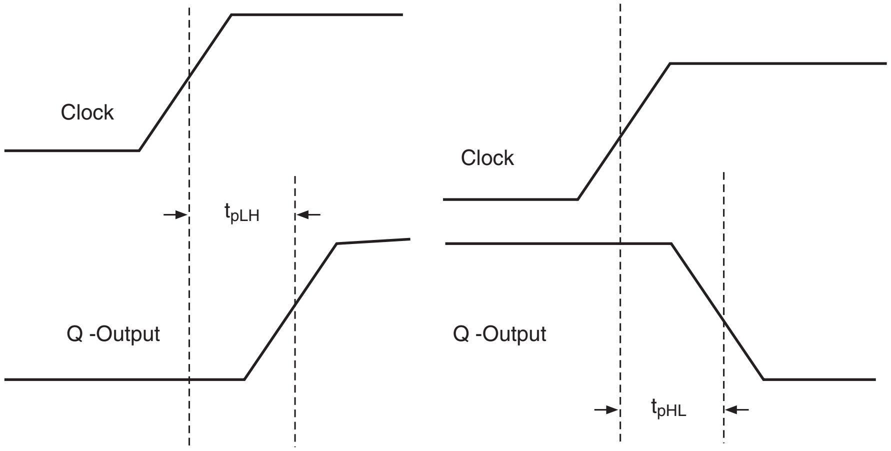


Figure 10.44 Propagation delay.

10.9.3 Clock Pulse HIGH and LOW Times

The clock pulse HIGH time t_W (H) and clock pulse LOW time, t_W (L) are respectively the minimum time durations for which the clock signal should remain HIGH and LOW. Failure to meet these requirements can lead to unreliable triggering. Figure 10.45 depicts these timing parameters. t_W (H) and t_W (L) for 74ALS109A are 4 and 5.5 ns respectively.

10.9.4 Asynchronous Input Active Pulse Width

This is the minimum time duration for which the asynchronous input (PRESET or CLEAR) must be kept in its active state, usually LOW, for the output to respond properly. It is 4 ns in the case of flip-flop 74ALS109A. Figure 10.46 shows this timing parameter.

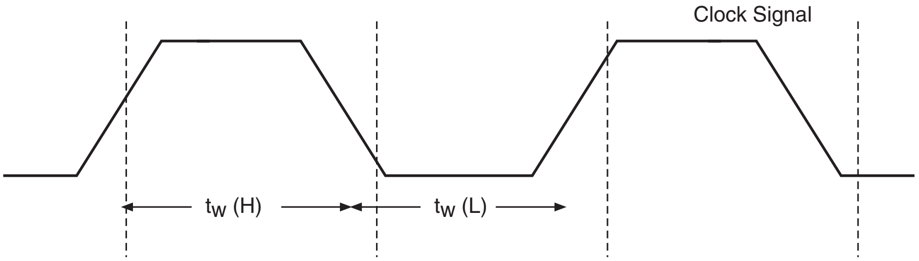


Figure 10.45 Clock pulse HIGH and LOW times.

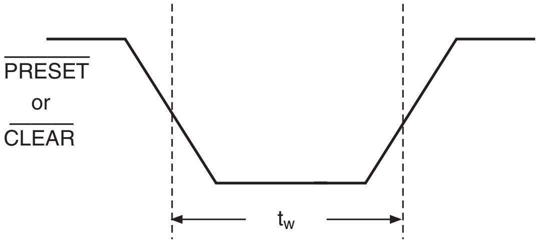


Figure 10.46 Asynchronous input active pulse width.

10.9.5 Clock Transition Times

The manufacturers specify the maximum transition times (rise time and fall time) for the output to respond properly. If these specified figures are exceeded, the flip-flop may respond erratically or even may not respond at all. This parameter is logic family specific and is not specified for individual devices. The allowed maximum transition time for TTL devices is much smaller than that for CMOS devices. Also, within the broad TTL family, it varies from one subfamily to another.

10.9.6 Maximum Clock Frequency

This is the highest frequency that can be applied to the clock input. If this figure is exceeded, there is no guarantee that the device will work reliably and properly. This figure may vary slightly from device to device of even the same type number. The manufacturer usually specifies a safe value. If this specified value is not exceeded, the manufacturer guarantees that the device will trigger reliably. It is 34 MHz for 74ALS109A.

10.10 Flip-Flop Applications

Flip-flops are used in a variety of application circuits, the most common among these being the *frequency division and counting* circuits and *data storage and transfer* circuits. These application areas are discussed at length in Chapter 11 on counters and registers. Both these applications use a cascaded arrangement of flip-flops with or without some additional combinational logic to perform the desired function. Counters and registers are available in IC form for a variety of digital circuit applications.

Other applications of flip-flops include their use for switch debouncing, where even an unclocked flip-flop (such as a NAND or a NOR latch) can be used, for synchronizing asynchronous inputs with the clock input and for identification of edges of synchronous inputs. These are briefly described in the following paragraphs.

10.10.1 Switch Debouncing

Owing to the switch bounce phenomenon, the mechanical switch cannot be used as such to produce a clean voltage transition. Refer to Fig. 10.47(a). When the switch is moved from position 1 to position 2, what is desired at the output is a clean voltage transition from 0 to $+V$ volts, as shown in Fig. 10.47(b). What actually happens is shown in Fig. 10.47(c). The output makes several transitions between 0 and

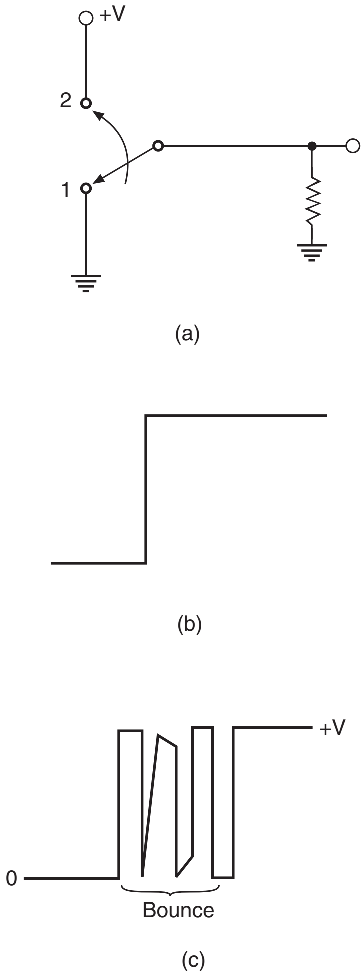


Figure 10.47 Switch bounce phenomenon.

$+V$ volts for a few milliseconds owing to contact bounce before it finally settles at $+V$ volts. Similarly, when it is moved from position 2 back to position 1, it makes several transitions before coming to rest at 0 V. Although this random behaviour lasts only for a few milliseconds, it is unacceptable for many digital circuit applications. A NAND or a NOR latch can solve this problem and provide a clean output transition. Figure 10.48 shows a typical switch debounce circuit built around a NAND latch. The circuit functions as follows.

When the switch is in position 1, the output is at a ‘0’ level. When it is moved to position 2, the output goes to a ‘1’ level within a few nanoseconds (depending upon the propagation delay of the NAND gate) after its first contact with position 2. When the switch contact bounces, it makes and breaks contact with position 2 before it finally settles at the intended position. Making of contact

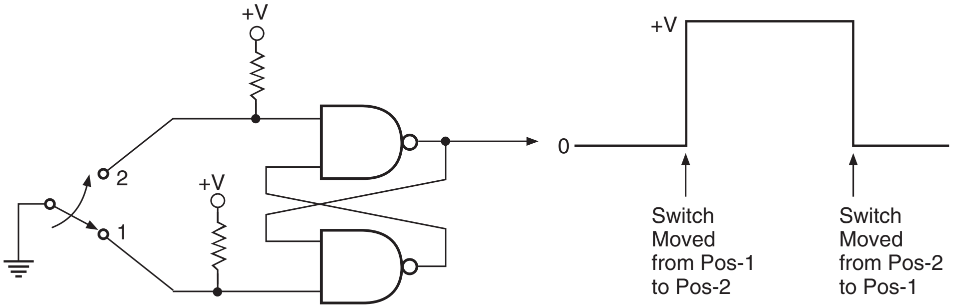

+V
+V
2
1
+V
0
Switch Moved from Pos-1 to Pos-2
Switch Moved from Pos-2 to Pos-1

Figure 10.48 Switch debounce circuit.

always leads to a '1' level at the output, and breaking of contact also leads to a '1' level at the output owing to the fact that the contact break produces a '1' level at both inputs of the latch which forces the output to hold its existing logic state. The fact that when the switch is brought back to position 1 the output makes a neat transition to a '0' level can be explained on similar lines.

10.10.2 Flip-Flop Synchronization

Consider a situation where a certain clock input, which works in conjunction with various synchronous inputs, is to be gated with an asynchronously generated gating pulse, as shown in Fig. 10.49. The output in this case has the clock pulses at one or both ends shortened in width, as shown in Fig. 10.49. This problem can be overcome and the gating operation synchronized with the help of a flip-flop, as shown in Fig. 10.50.

10.10.3 Detecting the Sequence of Edges

Flip-flops can also be used to detect the sequence of occurrence of rising and falling edges. Figure 10.51 shows how a flip-flop can be used to detect whether a positive-going edge A follows or precedes another positive-going edge B. The two edges are respectively applied to *D* and clock inputs of a

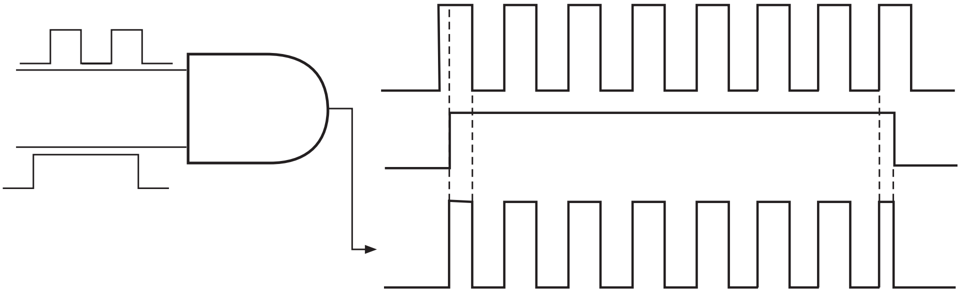


Figure 10.49 Gating of a clock signal.

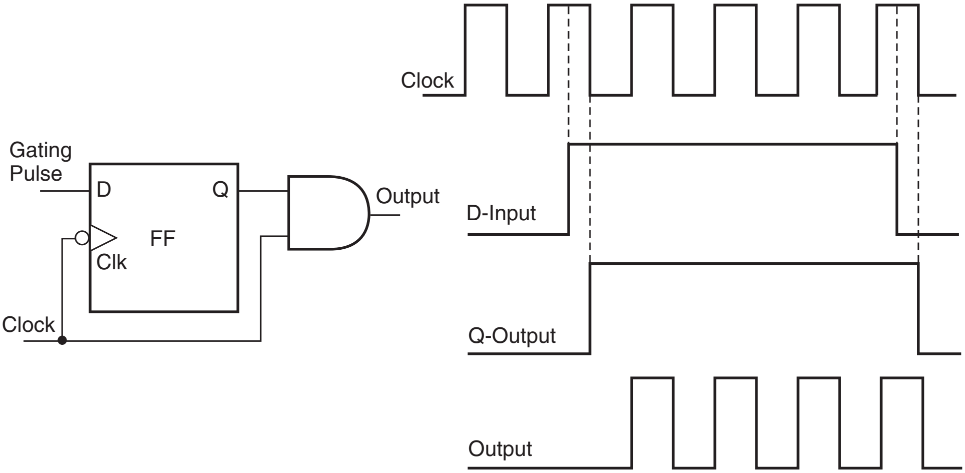


Figure 10.50 Flip-flop synchronization.

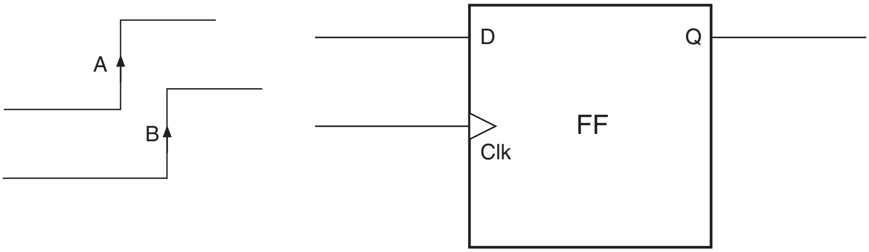


Figure 10.51 Detection of the sequence of edges.

positively edge-triggered *D* flip-flop. If edge A arrives first, then, on arrival of edge B, the output goes from 0 to 1. If it is otherwise, it stays at a ‘0’ level.

Example 10.7

Figure 10.52 shows two pulsed waveforms A and B, with waveform A leading waveform B in phase, as shown in the figure. Suggest a flip-flop circuit to detect this condition by producing (a) a logic ‘1’ Q output and (b) a logic ‘0’ Q output.

Solution

(a) A positive edge-triggered *D* flip-flop, as shown in Fig. 10.53(a), can be used for the purpose. Waveform A is applied to the *D* input, and waveform B is applied to the clock input. If we examine the two waveforms, we will find that, on every occurrence of the leading edge of waveform B,

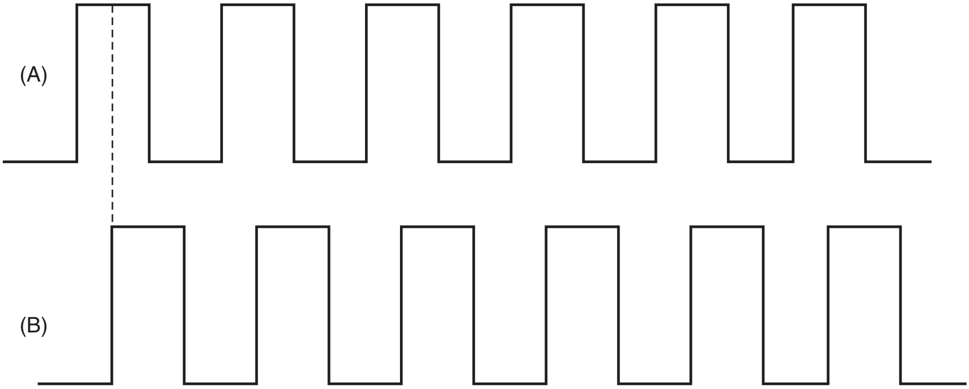
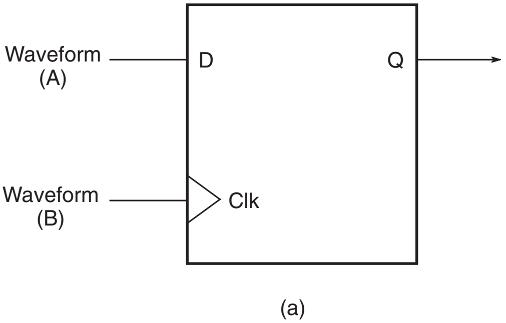


Figure 10.52 Example 10.7.



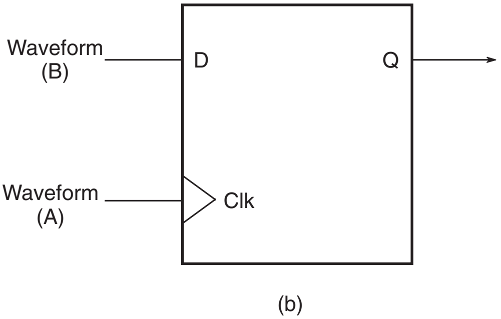


Figure 10.53 Solution to example 10.7.

waveform A is in a logic '1' state. Thus, the Q output in this case will always be in a logic '1' state.

(b) By interchanging the connections of waveforms A and B as shown in Fig. 10.53(b), the Q output will be in a logic '0' state as long as waveform A leads waveform B in phase. In this case, on every occurrence of the leading edge of waveform A (clock input), waveform B (D input) is in a logic '0' state.

10.11 Application-Relevant Data

Table 10.1 lists popular type numbers of flip-flops belonging to TTL, CMOS and ECL logic families. Application-relevant information of some of the popular type numbers is given in the companion website. The information given includes the pin connection diagram, package style and function table.

Table 10.1 Popular type numbers of flip-flops belonging to the TTL, CMOS and ECL logic families.

IC type number	Function	Logic family
54/7473	Dual J-K negative edge-triggered flip-flop with CLEAR	TTL
54/7474	Dual D-type positive edge-triggered flip-flop with PRESET and CLEAR	TTL
54/7475	Four-bit D-type latch	TTL
54/7476	Dual J-K flip-flop with PRESET and CLEAR	TTL
54/7478	Dual J-K flip-flop with PRESET and CLEAR	TTL
54/74107	Dual J-K flip-flop with CLEAR	TTL
54/74109	Dual J-K positive edge-triggered flip-flop with PRESET and CLEAR	TTL
54/74112	Dual J-K negative edge-triggered flip-flop with PRESET and CLEAR	TTL
54/74113	Dual J-K negative edge-triggered flip-flop with PRESET	TTL
54/74114	Dual J-K negative edge-triggered flip-flop with PRESET and CLEAR	TTL
54/74121	Monostable multivibrator	TTL
54/74122	Retriggerable monostable multivibrator	TTL
54/74123	Dual retriggerable monostable multivibrator	TTL
54/74174	Hex D-type flip-flop with CLEAR	TTL
54/74175	Quad edge triggered D-type flip-flop with CLEAR	TTL
54/74221	Dual monostable multivibrator	TTL
54/74256	Dual four-bit addressable latch	TTL
54/74259	Eight-bit addressable latch	TTL
54/74273	Octal D-type flip-flop with MASTER RESET	TTL
54/74279	Quad SET/RESET latch	TTL
54/74373	Octal transparent latch (three-state)	TTL
54/74374	Octal D-type flip-flop (three-state)	TTL
54/74377	Octal D-type flip-flop with common ENABLE	TTL
54/74378	Hex D-type flip-flop with ENABLE	TTL
54/74379	Four-bit D-type flip-flop with ENABLE	TTL
54/74533	Octal transparent latch (three-state)	TTL
54/74534	Octal D-type flip-flop (three-state)	TTL
54/74573	Octal D-type latch (three-state)	TTL
54/74574	Octal D-type flip-flop (three-state)	TTL

(continued overleaf)

Table 10.1 (*continued*).

IC type number	Function	Logic family
4013	Dual *D*-type flip-flop	CMOS
4027	Dual *J*-*K* flip-flop	CMOS
4042	Quad *D*-type latch	CMOS
4044	Quad *R*-*S* latch with three-state output	CMOS
4047	Low-power monostable/astable multivibrator	CMOS
4076	Quad *D*-type flip-flop with three-state output	CMOS
40174	Hex *D*-type flip-flop	CMOS
40175	Quad *D*-type flip-flop	CMOS
4511	BCD to seven-segment latch/decoder/driver	CMOS
4528	Dual retriggerable resettable monostable multivibrator	CMOS
4543	BCD to seven-segment latch/decoder/driver for LCD	CMOS
4723	Dual four-bit addressable latch	CMOS
4724	Eight-bit addressable latch	CMOS
MC10130	Quad *D*-type latch	ECL
MC10131	Dual *D*-type master/slave flip-flop	ECL
MC10133	Quad *D*-type latch (negative transition)	ECL
MC10135	Dual *J*-*K* master/slave flip-flop	ECL
MC10153	Quad latch (positive transition)	ECL
MC10168	Quad *D*-type latch	ECL
MC10175	Quint latch	ECL
MC10176	Hex *D*-type master/slave flip-flop	ECL
MC10198	Monostable multivibrator	ECL
MC10231	High-Speed dual *D*-type M/S flip-flop	ECL
MC1666	Dual clocked *R*-*S* flip-flop	ECL
MC1668	Dual clocked latch	ECL
MC1670	*D*-type master/slave flip-flop	ECL
MC1658	Voltage-controlled multivibrator	ECL

Review Questions

1. Briefly describe the operational aspects of bistable, monostable and astable multivibrators. Which multivibrator closely resembles a flip-flop?
2. What is a flip-flop? Show the logic implementation of an *R*-*S* flip-flop having active HIGH *R* and *S* inputs. Draw its truth table and mark the invalid entry.
3. With the help of the logic diagram, describe the operation of a clocked *R*-*S* flip-flop with active LOW *R* and *S* inputs. Draw the truth table of this flip-flop if it were negatively edge triggered.
4. What is a clocked *J*-*K* flip-flop? What improvement does it have over a clocked *R*-*S* flip-flop?
5. Differentiate between:

   (a) synchronous and asynchronous inputs;
   (b) level-triggered and edge-triggered flip-flops;
   (c) active LOW and active HIGH inputs.

6. Briefly describe the following flip-flop timing parameters:

(a) set-up time and hold time;
(b) propagation delay;
(c) maximum clock frequency.

7. Draw the truth table for the following types of flip-flop:

   (a) a positive edge-triggered J-K flip-flop with active HIGH J and K inputs and active LOW PRESET and CLEAR inputs;
   (b) a negative edge-triggered J-K flip-flop with active LOW J and K inputs and active LOW PRESET and CLEAR inputs.

8. What is meant by the race problem in flip-flops? How does a master–slave configuration help in solving this problem?
9. Differentiate between a D flip-flop and a D latch.
10. Draw the function table for (a) a negative edge-triggered D flip-flop and (b) a D latch with an active LOW ENABLE input.
11. With the help of a schematic arrangement, explain how a J-K flip-flop can be used as a (a) a D flip-flop and (b) a T flip-flop.
12. With the help of a suitable circuit, briefly explain how a D flip-flop can be used to detect the sequence of occurrence of edges of synchronous inputs.

Problems

1. A 100 kHz clock signal is applied to a J-K flip-flop with $J = K = 1$.

   (a) If the flip-flop has active HIGH J and K inputs and is negative edge triggered, determine the frequency of the Q and $\overline{Q}$ outputs.
   (b) If the flip-flop has active LOW J and K inputs and is positive edge triggered, what should the frequency of the Q and $\overline{Q}$ outputs be? Assume that Q is initially '0'.

   (a) Q output = 50 kHz, $\overline{Q}$ output = 50 kHz;
   (b) both outputs remain in a logic '0' state

2. In a Schmitt trigger inverter circuit, the two trip points are observed to occur at 1.8 and 2.8 V. At what input voltage levels will this device make (a) HIGH-to-LOW transition and (b) LOW-to-HIGH transition?

   (a) 2.8 V; (b) 1.8 V

3. In the case of a presettable, clearable J-K flip-flop with active HIGH J and K inputs and active LOW PRESET and CLEAR inputs, what would the Q output logic status be for the following input conditions, assuming that Q is initially '0', immediately after it is clocked?

   (a) $J = 1$, $K = 0$, PRESET = 1, CLEAR = 1;
   (b) $J = 1$, $K = 1$, PRESET = 0, CLEAR = 1;
   (c) $J = 0$, $K = 1$, PRESET = 1, CLEAR = 0;
   (d) $J = K = 0$, PRESET = 0, CLEAR = 1.

   (a) 1; (b) 1; (c) 0; (d) 1

4. Figure 10.54 shows the function table of a certain flip-flop. Identify the flip-flop.

Negative edge-triggered J-K flip-flop with active HIGH J and K inputs and active LOW PRESET and CLEAR inputs

Pr	Cl	Clk	J	K	Qn+1	$\overline{\text{Qn+1}}$
1	0	X	X	X	0	1
0	1	X	X	X	1	0
0	0	X	X	X	Unstable	
0	0	↓	1	1	1	0
0	0	↓	0	1	0	1
0	0	↓	1	1	Toggle	
0	0	↓	0	0	Qn	Qn

Figure 10.54 Problem 4.

5. Derive the expression for Q_{n+1} in terms of Q_n and J and K inputs for a clocked J-K flip-flop with active LOW J and K inputs. Q_n and Q_{n+1} have the usual meaning.

$$Q_{n+1} = \overline{J}.\overline{Q_n} + K.Q_n$$

6. Consider a J-K flip-flop (J-$\overline{K}$ flip-flop to be more precise) where an inverter has been wired between the external $\overline{K}$ input and the internal K input as shown in Fig. 10.55. With the help of a characteristic table, write the characteristic equation for this flip-flop.

$$Q_{n+1} = J.\overline{Q_n} + K.Q_n$$

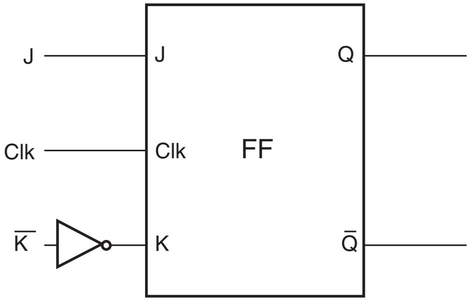


Figure 10.55 Problem 6.

Further Reading

1. Cook, N. P. (2003), *Practical Digital Electronics*, Prentice-Hall, NJ, USA.
2. Whitaker, C. (1996) *The Electronics Handbook,* CRC Press (in cooperation with IEEE Press), FL, USA.
3. Tokheim, R. L. (1994) *Schaum's Outline Series of Digital Principles*, McGraw-Hill Companies Inc., USA.
4. Tocci, R. J. (2006), *Digital Systems – Principles and Applications*, Prentice-Hall Inc., NJ, USA.
5. Malvino, A. P. and Leach, D. P. (1994) *Digital Principles and Applications*, McGraw-Hill, USA.