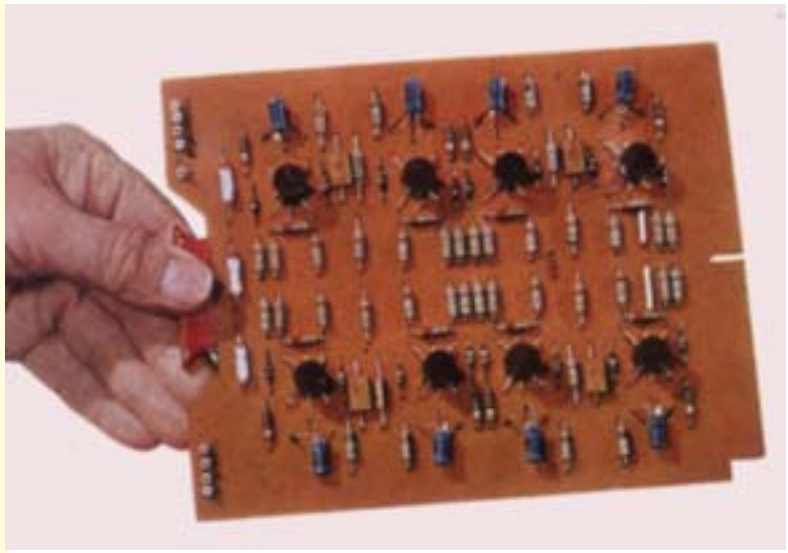
CONTENTS

CHAPTER 72

FLIP-FLOPS AND RELATED DEVICES

Learning Objectives

- Flip-Flop (FF)
- Latch
- NAND Gate Latch
- NOR Gate Latch
- Clocked Signals
- Some Main Ideas Common to Clocked Flip-Flops
- Clocked S-C Flip-Flop
- Clocked J-K Flip-Flop
- Clocked D Flip-Flop
- Parallel Transfer of Data Using D-Flip-Flops
- D Latch (Transparent Latch)
- Clocked J-K Flip-Flop with Asynchronous Inputs
- Flip-Flop Timing Parameters
- IC Flip-Flop Timing Values
- Applications of Flip-Flop
- Flip-Flop Synchronization
- Parallel Data Transfer
- Frequency Division
- Counting
- Schmitt Trigger Devices
- Types of one-shots
- Retriggerable One-Shot
- Actual One-Shot Devices
- Clock Generator Circuits
- Schmitt Trigger Oscillator
- 555 Timer Used as an Astable Multivibrator



A flip-flop is made up of logic gates. The flip-flpos are extensively used as a memory cell in static random access memory of a computer

CONTENTS

72.1. Introduction

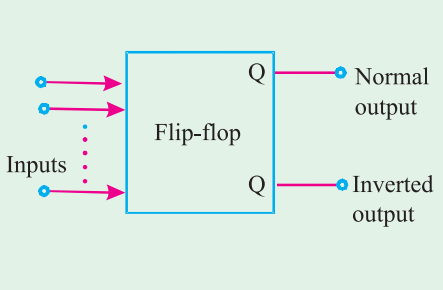
The output logic levels of combinational logic circuits at any instant of time are dependent on the logic levels present at the inputs at that time. Any prior input-logic level conditions have no effect on the present outputs. This is due to the fact that combinational logic circuits have no ***memory.*** However, it will be interesting to know that most digital systems are made up of both combinational logic circuits and memory elements.

The most important memory element is the flip-flop (abbreviated as FF). The FF is made up of an assembly of logic gates. It may be noted that even though a logic gate, by itself, has no storage capability, but several such logic gates can be connected together in ways that permit information to be stored. Several different gate arrangements are used to produce these flip-flops.

The flip-flops are used extensively as a memory cell in static random access memory (SRAM) of a computer.

72.2. Flip-Flop (FF)

We have already discussed in the last article that flip-flop (abbreviated as FF) is made up of logic gates and it permits information to be stored in it. Fig. 72.1 (*a*) shows a general type of symbol used for a flip-flop.



(*a*) Flip-flop symbol

Output states

$Q = 1, \overline{Q} = 0$: It is called HIGH or 1 State; also called SET state

$Q = 1, \overline{Q} = 0$: It is called LOW or 0 state; also called CLEAR or RESET state

(*b*) Operating states of Flip-flop

Fig. 72.1

As seen from this diagram, the flip-flop has two outputs labeled as Q and $\overline{Q}$ that are inverse (or complement) of each other. The Q output is called the ***normal*** flip-flop output and $\overline{Q}$ is the ***inverted*** flip-flop output. It may be carefully noted that whenever we refer to the state of a flip-flop, we are referring to the state of its ***normal*** (Q) output. For instance if we say flip-flop is in the HIGH state, we mean that $Q = 1$. On the other hand if we say flip-flop is in LOW state, we mean that $Q = 0$. It is automatically understood that the $\overline{Q}$ state will always be ***inverse*** of Q, *i. e.*, if $Q = 1$, $\overline{Q} = 0$, and if $Q = 0$, $\overline{Q} = 1$.

Fig. 72.1 (*b*) shows the two possible operating states of a flip-flop. One possible state is $Q = 1$, $\overline{Q} = 0$ and the other $Q = 0$, $\overline{Q} = 1$. The state $Q = 1$, $\overline{Q} = 0$ is called HIGH state or 1 state. It is also called *SET* state. Whenever, the inputs to a flip-flop cause it to go to the $Q = 1$ state, we call this ***setting*** the flip-flop or the flip-flop is set.

In a similar way, the state $Q = 0$, $\overline{Q} = 1$ is called LOW state or 0 state. It is also called *RESET* or CLEAR state. Whenever, the inputs to a flip-flop cause it to go to the $Q = 0$ state, we call this ***resetting*** **or** ***clearing*** the flip-flop. In the later part of the chapter we will see that many flip-flops will have a *SET* input and/or RESET (CLEAR) input that is used to drive the flip-flop into a specific output state.

It may be noted from the flip-flop symbol (Fig. 72.1 (*a*)) that a flip-flop can have one or more inputs. These inputs can be used to switch the flip-flop back and forth between its two possible output states.

It will be shown later that most flip-flop inputs need only to be momentarily activated (or pulsed) in order to cause a chage in the output state. The flip-flop output will remain in the new state even after the input pulse is over. This is flip-flop's memory characteristic.

It will be interesting to know that a flip-flop is also known as a latch and a bistable multivibrator. The term ***"latch"*** is used for certain types of flip-flops that will be described later. The term bistable multivibrator is the more technical name but flip-flop is the one used more frequently among the engineers and technologists.

72.3. Latch

A latch is the most basic type of flip-flop circuit. It can be constructed using NAND or NOR gates. Accordingly the latches are of two types :

1. NAND gate latch and

2. NOR gate latch

Both these types of latch are discussed one by one in the following pages.

72.4. NAND Gate Latch

Construction. Fig. 72.2 shows a latch constructed from NAND gates. It is called NAND gate latch or simply NAND latch. As seen from this diagram, the two NAND gates are cross-coupled. This means output of NAND-1 is connected to one of the inputs of NAND-2. Similarly, the output of NAND-2 is connected to one of the inputs of NAND-1. The NAND gate latch outputs are labeled as Q and $\overline{Q}$ respectively. Under normal conditions, these outputs will always be the inverse (or complement) of each other. The latch has two inputs namely SET and CLEAR. The SET input sets Q output of the latch to 1 state while the CLEAR input sets the Q output to the 0 state.

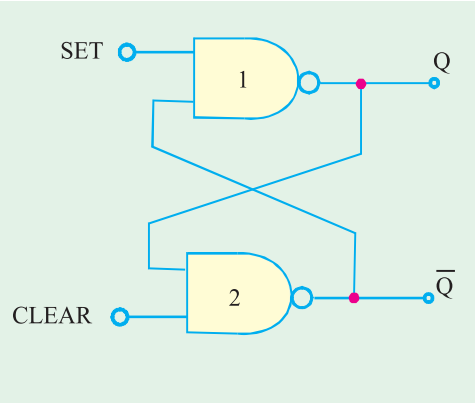


Fig. 72.2. NAND gate latch.

Operation. Let us now understand the operation of NAND gate latch. Normally, the SET and CLEAR inputs of the latch are resting in the HIGH state. Whenever we want to change the latch outputs, one of the two inputs will be pulsed LOW. Let us begin our study by showing that there are two equally likely output states when SET = CLEAR = 1. One possibility is shown in Fig. 72.3 (*a*) where we have $Q = 0$ and $\overline{Q} = 1$. With $Q = 0$, the inputs to the NAND-2 are 0 and 1, which produces $\overline{Q} = 1$. The 1 from $\overline{Q}$ causes NAND-1 to have 1 at both inputs to produce a 0 output at Q. Thus we find that a LOW at the NAND-1 output produces a HIGH at NAND-2 output which in turn keeps the NAND-1 output LOW.

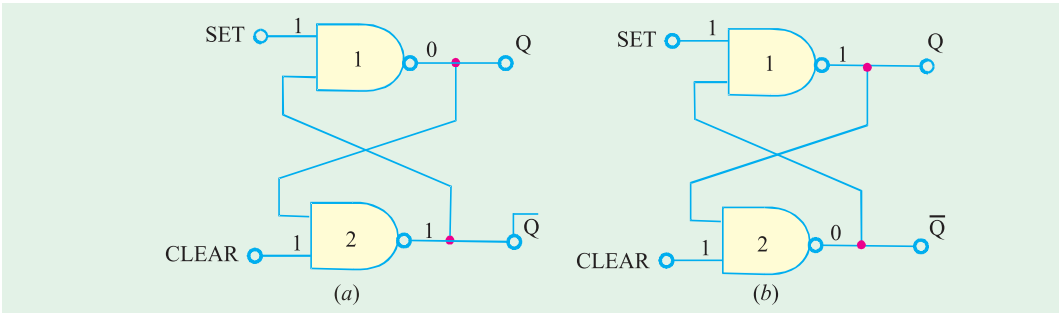


Fig. 72.3. Illustrating the two possible output states of NAND gate latch.

Fig. 72.3 (*b*) shows the second possibility. Here $Q = 1$ and $\overline{Q} = 0$. As seen from this figure, the HIGH from NAND-1 produces a LOW at the NAND-2 output, which in turn keeps the NAND-1 output HIGH. Thus there are two possible output states when SET = CLEAR = 1. However which one of these two states exist will depend on what has occurred previously at the inputs. This is discussed below in more detail where we will take up three situations : **(1)** setting the latch, **(2)** clearing the latch and simultaneous setting and clearing the latch.

1. Setting the Latch. We have already mentioned above that if SET = CLEAR = 1, the output Q can be in two possible states *i.e.* $Q = 0$ or $Q = 1$. Now we will study as what happens when the SET input is momentarily pulsed LOW while CLEAR is kept HIGH.

Fig. 72.4 (*a*) shows what happens when $Q = 0$ prior to the occurrence of the pulse. As SET input is pulsed LOW at time 't_0', Q will go HIGH. This will force $\overline{Q}$ to go LOW so that NAND-1 has now two LOW-inputs. Thus when SET returns to 1 state at t_1, the NAND-1 output remains HIGH which in turn keeps the NAND-2 output LOW.

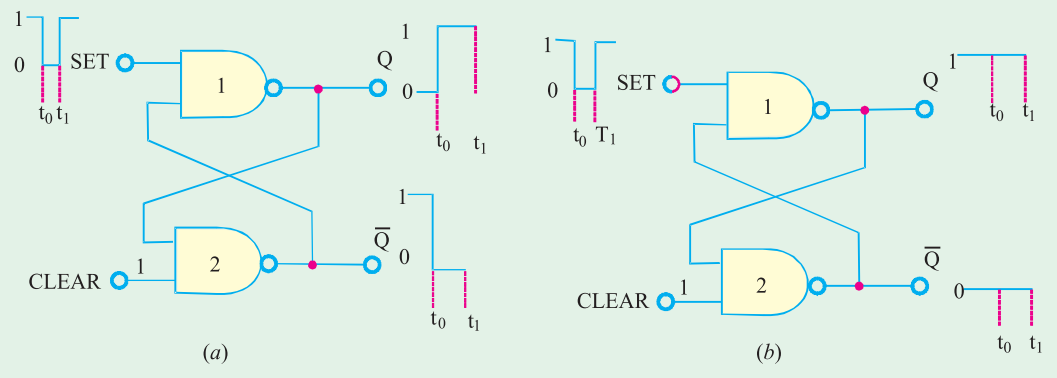


Fig. 72.4. Setting the NAND gate latch

Fig. 72.4 (*b*) shows what happens when $Q = 1$ and $\overline{Q} = 0$. Prior to the application of the SET pulse. Since $\overline{Q} = 0$ is already keeping the NAND-1 output HIGH, the LOW pulse at SET will not change anything. Thus when SET returns HIGH, the latch outputs are still in the $Q = 1$, $\overline{Q} = 0$ state (i.e no change in states).

2. Clearing the latch. Now we will study what happens when CLEAR input is pulsed LOW while SET is kept HIGH. Fig. 72.5 (*a*) shows the situation when $Q = 0$ and $\overline{Q} = 1$ prior to the application of the pulse. Since $Q = 0$ is already keeping the NAND-2 output HIGH, the LOW pulse at CLEAR will not have any effect. When CLEAR returns HIGH, the latch outputs are still $Q = 0$ and $\overline{Q} = 1$.

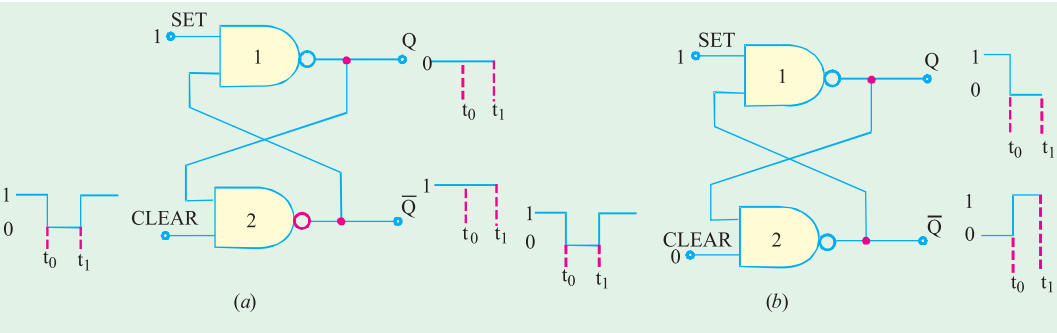


Fig 72.5. Clearing the NAND gate latch.

Fig 72.5 (*b*) shows the situation where $Q = 1$ prior to the occurrence of the CLEAR pulse. As CLEAR is pulsed LOW AT t_0, $\overline{Q}$ will go HIGH and this HIGH forces Q to go LOW because NAND-2 now has two LOW inputs. Thus when CLEAR returns HIGH at t_1, the NAND-2 output remains HIGH. This in turn keeps the NAND-1 output LOW.

Inputs		*Output*
SET	*CLEAR*	
1	1	No change
0	1	$Q = 1$
1	0	$Q = 0$
0	0	$Q = \overline{Q} = 1$ (Invalid)

Fig. 72.6. Truth table for NAND gate latch.

3. Simultaneous setting and clearing. If the SET and CLEAR inputs are simultaneously pulsed LOW, this will produce HIGH levels at both NAND outputs so that $Q = \overline{Q} = 1$. This is an undesirable condition as the two outputs are supposed to be inverse of each other. Moreover, when the SET and CLEAR inputs return HIGH, the resulting output state will depend on which input returns HIGH first. Simultaneous transitions back to 1 state will produce unpredictable results. Because of these reasons the SET = CLEAR = 0 condition is normally not used for the NAND latch and is considered as invalid condition.

The operation of NAND gate latch may be summarized in the form of a truth table as shown in Fig 72.6. Each line of the truth table is described as below :

1. **SET = CLEAR = 1.** This condition is the normal resting state of the latch.The Q and $\overline{Q}$ outputs will remain in the same state in which they were prior to this input condition.
2. **SET = 0, CLEAR = 1.** This condition will always cause the output to go to $Q = 1$ state where it will remain even after SET returns HIGH. This is called setting the latch.
3. **SET = 1, CLEAR = 0.** This condition will always produce $Q = 0$. The output will remain in this state even after CLEAR returns HIGH. This is called clearing or resetting the latch.
4. **SET = CLEAR = 0.** This condition tries to set and clear the latch simultaneously. It produces invalid results and should not be used.

72.5. Alternative Representations of NAND Gate Latch

We have already discussed in the last article about the NAND gate latch operation. It was mentioned that both SET and CLEAR inputs are active-LOW. Further the SET input will set $Q = 1$ when SET goes LOW. On the other hand, the CLEAR input will clear $Q = 0$ when CLEAR goes LOW. Because of this reason NAND gate latch is often drawn using the alternative representation for each NAND gate as shown in Fig 72.7 (*a*). (Recall from Art 3-41 that a NAND gate is equivalent to a bubbled or gate). The bubbles on the inputs of OR gate as well as labelling of the signals as $\overline{\text{SET}}$ and $\overline{\text{CLEAR}}$ indicate the active-LOW status of these inputs.

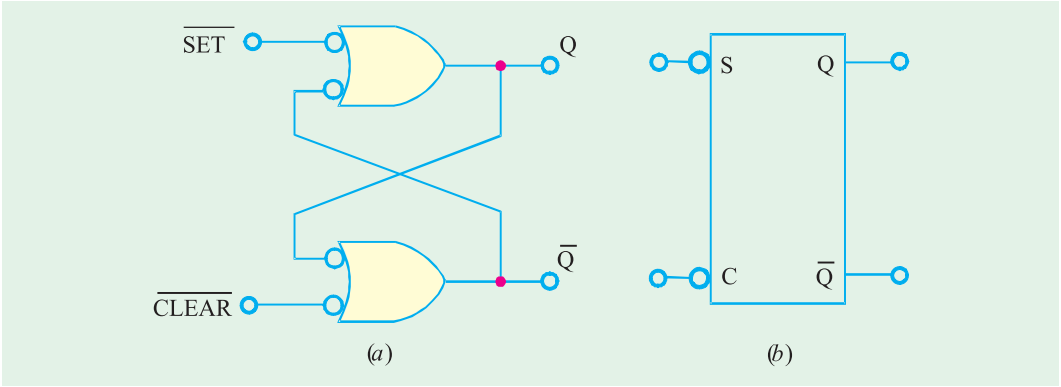


Fig. 72.7. Alternative representations of NAND gate latch.

Fig. 72.7 (*b*) shows a block representation. The *S* and *C* labels represent the SET and CLEAR inputs. While the bubbles at *S* and *C* inputs indicate the active LOW nature of these inputs. So remember whenever we use this block symbol, it represents a NAND gate latch.

Note. The action of ***Clearing*** a flip-flop or a latch is also called ***resetting.*** Both these terms (*i.e.* clearing or resetting) is used interchangeably in the field of digital electronics. Thus a SET-CLEAR latch can also be refered to as SET-RESET latch (or simply *S* - *R* latch).

Example 72.1. *The waveforms shown in Fig* 72.8 (*a*) *are applied to the inputs of the NAND latch shown in Fig.* 72.8 (*b*). *Assume that initially Q* = 0 *and determine the Q – waveform.*

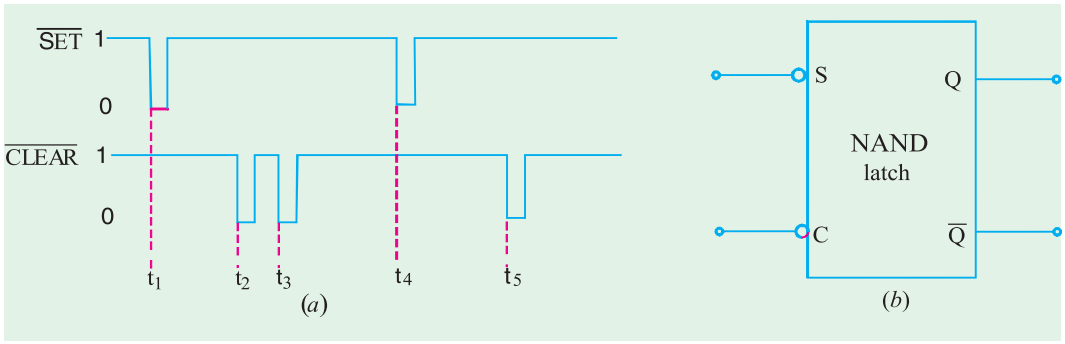


Fig. 72.8

Solution Given :

In order to determine the waveform at *Q*-output, refer to the truth table shown in Fig. 72.6 and the input waveforms. Notice that prior to $t = t_1$, $\overline{SET}$ = $\overline{CLEAR}$ = 1 and *Q* = 0, At $t = t_1$, $\overline{SET}$ goes 0 and $\overline{CLEAR}$ stays at 1. Referring to the truth table in Fig. 72-6, we find that if $\overline{SET}$ = 0, $\overline{CLEAR}$ = 1, *Q* = 1. Therefore the *Q* – output is indicated as 1 at $t = t_1$ in Fig. 72.9.

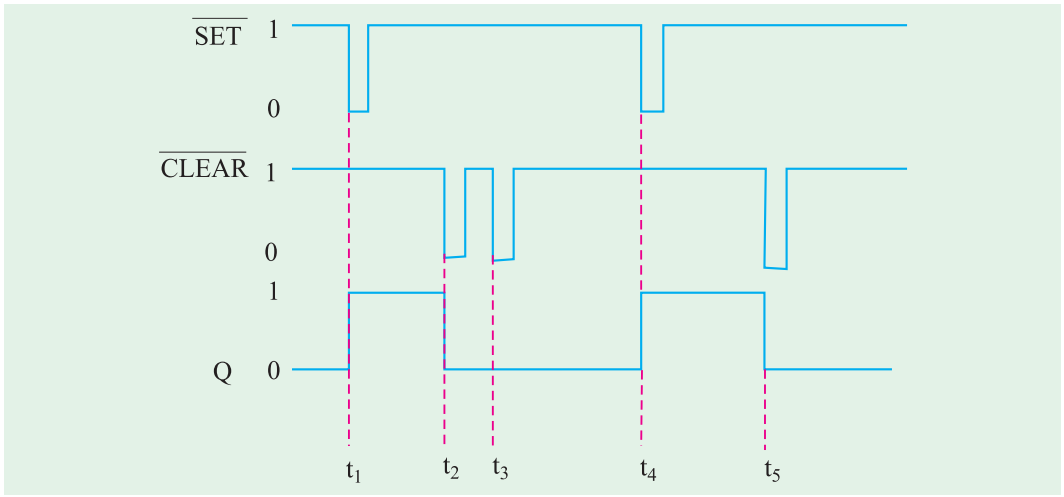


Fig. 72.9

At $t = t_2$, $\overline{SET}$ = 1, and $\overline{CLEAR}$ goes 0. Again referring to the truth table shown in Fig. 72.6, we find that *Q* = 0, therefore we sketch the *Q* output as 0 in Fig. 72.9. Similarly, at $t = t_3$, $\overline{SET}$ = 1, and $\overline{CLEAR}$ = 0. Since there is no change in inputs, the *Q*-output also remains at 0 as indicated in Fig. 72.9. At $t = t_4$, $\overline{SET}$ goes 0, and $\overline{CLEAR}$ is 1, therefore the output goes 1. At $t = t_5$, $\overline{SET}$ = 1 and $\overline{CLEAR}$ goes 0, therefore output goes 0. The complete *Q*-output waveform is shown in Fig. 72.9.

Example 72.2. *If the* $\overline{SET}$ *and* $\overline{CLEAR}$ *waveforms shown in Fig* 72.10 (*a*) *are applied to the inputs of NAND latch shown in Fig.* 72.10 (*b*) *determine the waveform that will be observed on the Q output. Assume that initially Q* = 0.

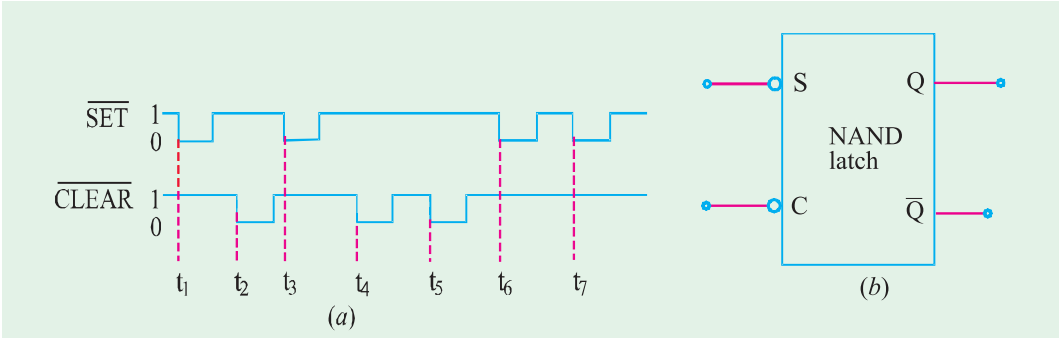


Fig. 72.10

Solution :

In order to determine, the Q-output, let us refer to the truth table shown in Fig. 72 .6 and the input waveforms. At $t = t_1$, $\overline{SET}$ goes from 1 to 0, $\overline{CLEAR}$ = 1, so the output Q goes from 0 to 1 as shown in Fig 72.11. At $t = t_2$, $\overline{SET}$ = 1, $\overline{CLEAR}$ goes from 1 to 0, so does the Q-output (*i.e*. it also goes from 1 to 0). At $t = t_3$, $\overline{SET}$ goes from 1 to 0, $\overline{CLEAR}$ = 1, therefore Q-output goes 1, At $t = t_4$, SET =1, $\overline{CLEAR}$ goes to 0, therefore Q-output goes 0 and so on. The complete waveform at Q-output is shown in Fig. 72.11.

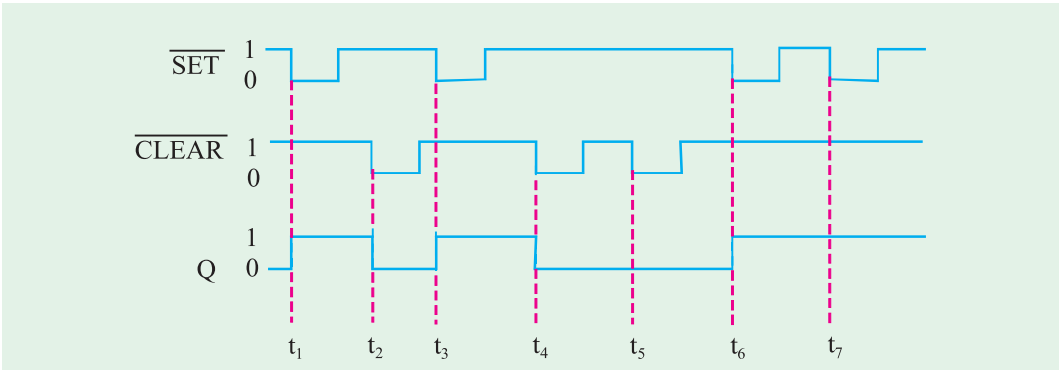


Fig. 72.11

72.6. Application of NAND Latch to Debounce a Mechanical Switch

Consider a mechanical switch with contact points *A* and *B* connected to V_{cc} (+ 5V) supply and ground respectively as shown in the Fig. 72.12 (*a*). It has been found experimentally that when the switch moves from contact position *A* to *B*, it produces several output voltage transitions as shown in Fig. 72.12 (*b*). It is due to a phenomenon called ***switch bounce*** *i.e*. the switch makes and breaks contact with contact *B* several times before coming to rest on contact *B*. In a similar manner, when the switch moves from contact position *B* to *A*, it again produces several output voltage transitions before coming to rest on contact *A*.

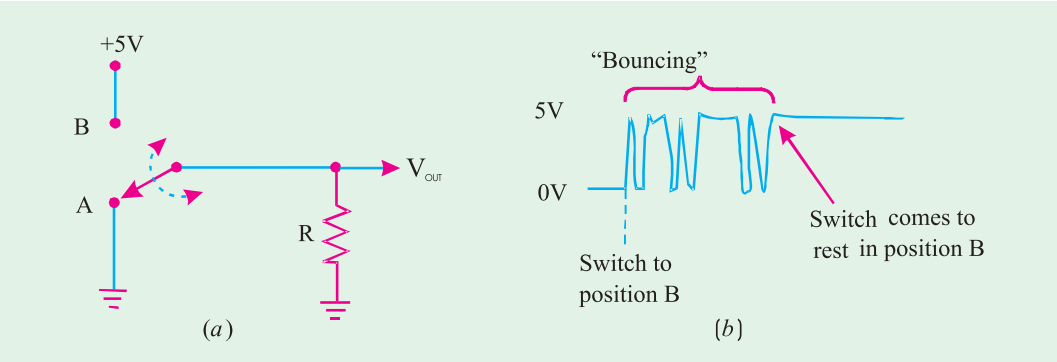


Fig. 72.12. Illustrating Switch Contact bounce

As a matter of fact, the multiple transitions on the output signal generally last only for few milliseconds. But such transitions are unacceptable in many applications. A NAND latch can be used as shown in Fig. 72.13 (*a*) to eliminate the switch bounce.

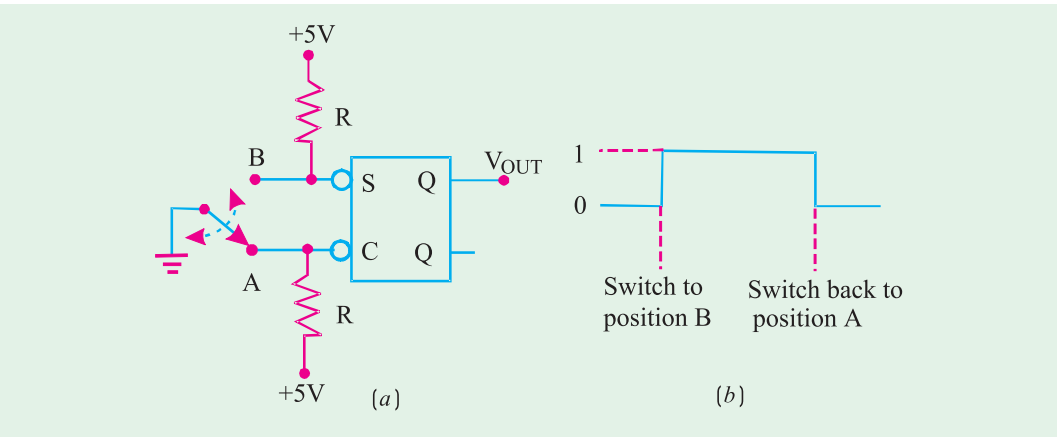


Fig. 72.13. Application of NAND latch to eliminate switch contact bounce.

The operation of the switch debouncing circuit may be understood as follows : Let us assume that initially the switch is resting in position *A* so that $\overline{\text{CLEAR}}$ input is LOW and $Q = 0$. When the switch is moved to position *B*, $\overline{\text{CLEAR}}$ 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 few nanoseconds. Now if the switch bounces off contact *B*, $\overline{\text{SET}}$ and $\overline{\text{CLEAR}}$ will both be HIGH and *Q* will not be affected, *i.e.* it will stay HIGH. Thus nothing will happen at *Q* as the switch bounces on and off contact *B* before finally coming to rest in position *B* as shown in Fig. 72.13 (*b*).

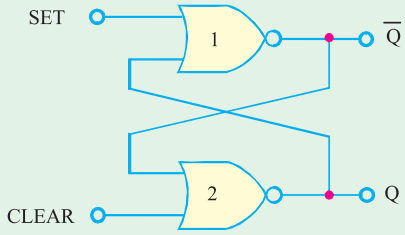
In a similar manner, when the switch is moved from position *B* back to position *A*, it will place a LOW on the $\overline{\text{CLEAR}}$ input as it first makes contact. This clears *Q* to the LOW state. It will remain there ever if the switch bounces on and off contact *A* several times before coming to rest.

It is evident from the above discussion that the output at *Q* will consists of single transition each time the switch is moved from one position to the other.

72.7. NOR Gate Latch

Fig. 72.14 (*a*) shows a latch constructed from two cross-coupled NOR gates. It is called NOR gate latch or simply the NOR latch. As seen in the figure, the arrangement is similar to the

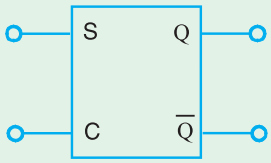
NAND gate latch (shown in Fig 72.2 on page 2637) except that the Q and $\overline{Q}$ outputs have reversed positions.



(*a*) Construction

Inputs		*Output*
SET	*CLEAR*	
0	0	No change
1	0	$Q = 1$
0	1	$Q = 0$
1	1	*Invalid

(*b*) Truth table



(*c*) Logic symbol

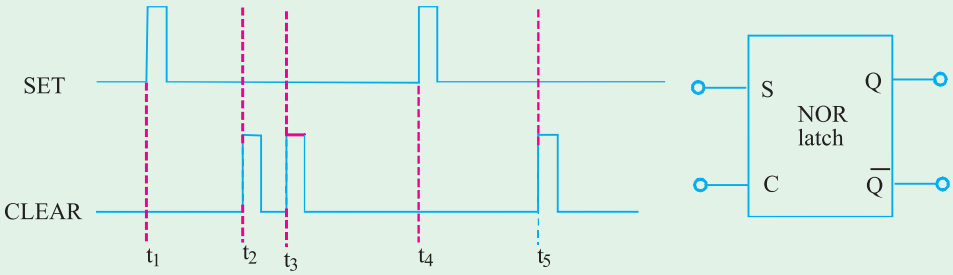
Fig. 72.14. NOR gate latch.

The detailed operation of the NOR latch can be understood exactly in the same manner as for the NAND latch. The results are given in the truth table as shown in Fig 72.14 (*b*) and are summarized as follows :

1. **SET = CLEAR = 0.** This condition is the normal resting state for the NOR latch. It has no effect on the output state. In other words, Q and $\overline{Q}$ will remain in the same state in which they were prior to this input condition.
2. **SET = 1, CLEAR = 0.** This condition will always cause the output to go to $Q = 1$ state where it will remain even after SET returns to 0.
3. **SET = 0, CLEAR = 1.** This condition will always cause the output to go to $Q = 0$ state where it will remain even after CLEAR returns to 0.
4. **SET = 1, CLEAR = 1.** This condition tries to set and clear the latch simultaneously. It produces invalid results and should not be used.

It may be carefully noted that the NOR latch operates exactly in the same manner as the NAND latch. However, the SET and CLEAR inputs are active-HIGH rather than active-LOW. Moreover the normal resting state is SET = CLEAR = 0. Further the output 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 CLEAR input. Fig 72.14 (*c*) shows the logic symbol for the NOR latch. Notice that there are no bubbles on the S and C inputs (unlike NAND latch) which indicates that these inputs are active-HIGH.

Example 72.3. *The waveforms shown in Fig.* 72.15 (*a*) *are applied to the inputs of the NOR latch shown in Fig.* 72.15 (*b*). *Assume that initially,* $Q = 0$ *and determine the Q-waveform.*



(*a*) (*b*)

Fig. 72.15

Solution :

In order to determine the Q-waveform, we will refer to the truth table of the NOR latch (shown in

Fig. 72.14 (*b*) page 2643) and the SET and CLEAR waveforms of Fig. 72.15 (*a*). At $t = t_1$, SET goes 1 and CLEAR is 0, therefore *Q*-output also goes 1. AT $t = t_2$, SET = 0 and CLEAR goes 1, therefore the *Q*-output goes 0. At $t = t_3$, SET = 0 and CLEAR goes 1 again, therefore the Q-output remains 0. At $t = t_4$, SET goes 1 and CLEAR is 0, therefore *Q*-output goes 1. At $t = t_5$, SET = 1 and CLEAR goes 1, therefore *Q*-output goes 0. The complete sketch of the *Q*-waveform along with SET and CLEAR waveforms is as shown in Fig. 72.16.

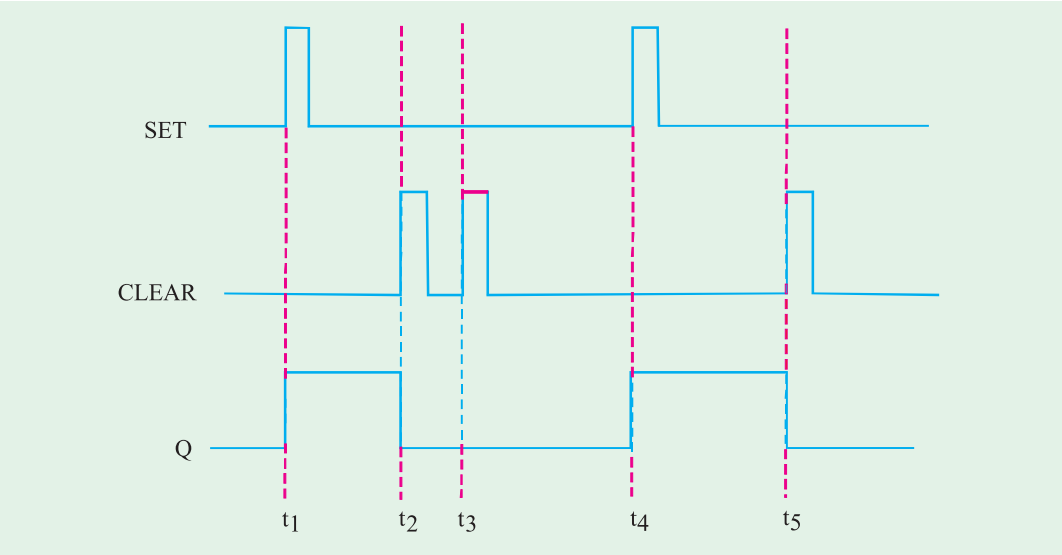


Fig. 72.16

Example 72.4. *If the SET and CLEAR waveforms shown in Fig.* 72.16. (*a*) *are applied to the inputs of NOR latch shown in Fig.* 72.17. (*b*), *determine the waveform that will be observed on the Q-output. Assume that initially Q* = 0.

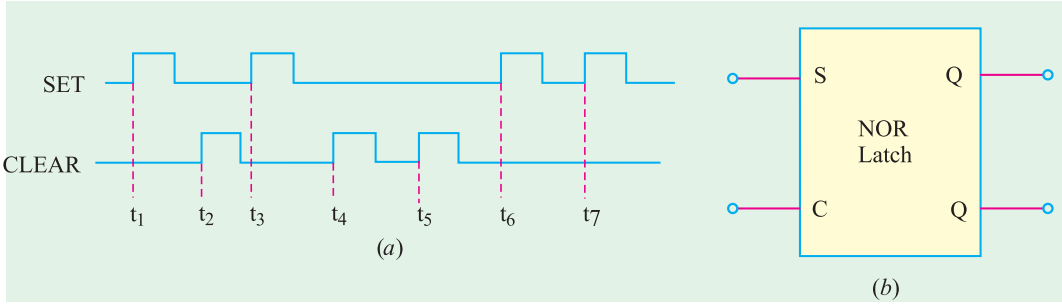


Fig. 72.17

Solution :

In order to determine the Q-waveform, we will refer to the truth table of the NOR latch (shown in Fig. 72.14 (*b*) page 2643) and the SET and CLEAR waveforms of Fig. 72.17. At $t = t_1$, SET goes 1 and CLEAR = 0, therefore *Q* goes 1. At $t = t_2$, SET = 0 and CLEAR goes 1, therefore *Q* goes 0. At $t = t_3$, SET = 1 again, and CLEAR = 0, therefore *Q* goes 1 again. At $t = t_4$, SET = 0. and CLEAR = 1, therefore *Q* goes 0 and so on. A complete sketch of *Q* waveform along with input waveforms is shown in Fig. 72.18.

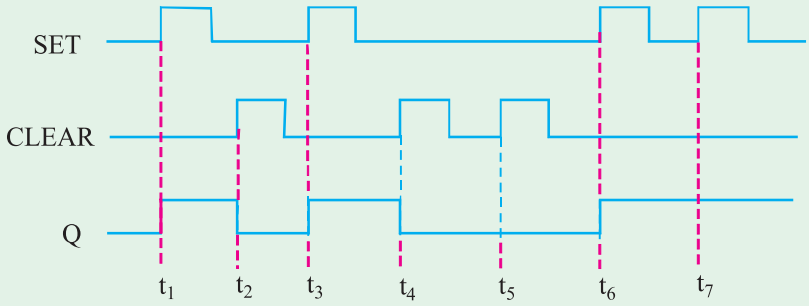


Fig. 72.18

72.8. Clocked Signals

Strictly speaking, the digital systems are of the following two types :

1. Asynchronous systems. In these systems, the outputs of logic circuits can change state any time one or more of the inputs change. Generally, an asynchronous system is more difficult to design and troubleshoot than a synchronous system.

2. Synchronous systems. In these systems, the exact time at which the output can change states are determined by a signal commonly called a ***clock.***

The clock signal is generally a rectangular pulse train as shown in Fig. 72.19 (*a*) or a square wave as shown in Fig. 72.19 (*b*). The clock signal is distributed to all parts of the digital system and most of the system outputs can change state only when the clock makes a transition. The transitions are more commonly referred to as ***edges*** and are pointed out in Fig. 72.19.

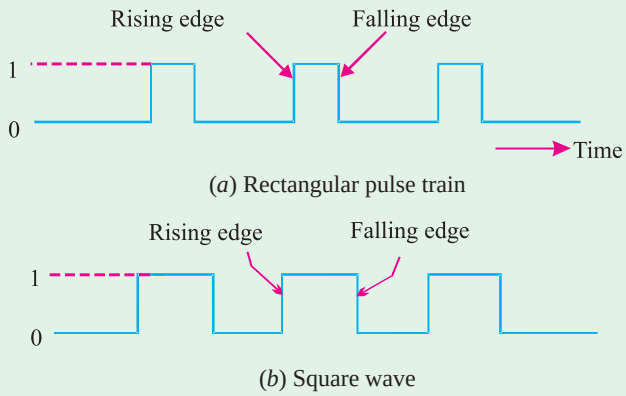


Fig. 72.19

It may be noted that when the clock changes from 0 to 1, this is called rising edge or ***positive-going transition*** (PGT). On the other hand, when the clock changes from 1 to 0, this is called falling edge or ***negative going transition*** (NGT).

As a matter of fact, most digital systems are principally synchronous (although there are always some asynchronous parts). It is due to the fact that synchronous circuits are easier to design and troubleshoot. The synchronization 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.

72.9. Some Main Ideas Common to Clocked Flip-Flops

There are several types of flip-flops that are used in a wide range of applications in the field of digital electronics. Before we begin our study of the different clocked flip-flops, let us describe the main ideas that are common to all of them.

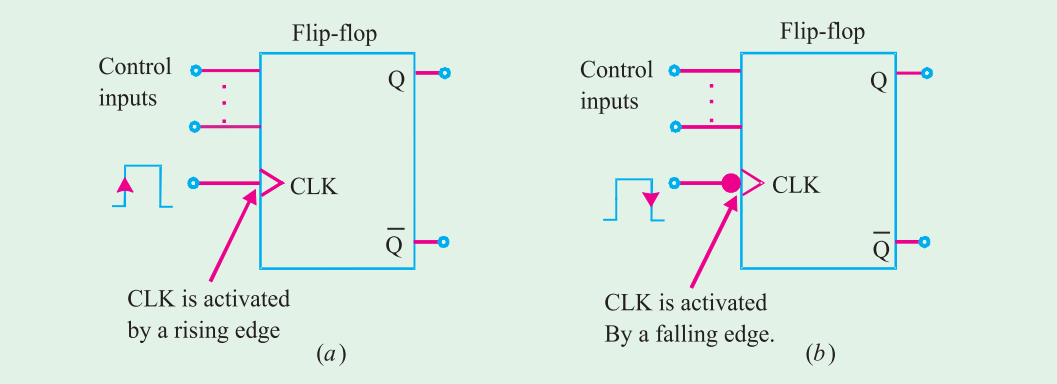


Fig 72.20. Clocked Flip-Flops.

Fig. 72.20. (*a*) and (*b*) shows the logic symbols for a typical clocked flip-flop. As seen, a clocked flip-flop has a clock input and some control inputs. There are described below in more detail.

CLK Input. Clocked flip-flops have a ***clock*** input that is labeled as CLK, CK or CP. However, we will normally use CLK as shown in Fig. 72.20 (*a*) and (*b*). In most clocked flip-flops, the CLK input is ***edge-triggered.*** This means the CLK input is activated by a signal transition. The edge-triggered activation is indicated by the presence of a small triangle on the CLK input. This contrasts with the latches, which are level-triggered.

Fig. 72.20 (*a*) shows a flip-flop with a small triangle on its CLK input to indicate that the input is activated only when a rising edge occurs. It may be noted that no other part of the input pulse will have an effect on the CLK input.

Fig 72.20 (*b*) shows a flip-flop symbol which has a bubble (a small circle) as well as a triangle on its CLK input. This signifies that CLK input is activated ***only*** when a falling edge occurs. Again note that no other part of the input pulse will have an effect on the CLK input.

Control Inputs. Clocked flip-flops 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. Because of this reason, the control inputs are refered to as ***synchronized control inputs.***

72.10. Setup and Hold Times in Clocked Flip-Flops

Strictly speaking, there are two timing requirements that must be met if a clocked flip-flop is to respond relialably to its control inputs when the active CLK edge occurs. These two requirements are : **(1)** set up time, t_S and **(2)** hold time, t_H . Both these requirements are discussed below.

1. Set up time, t_s. It is the time interval immediately preceeding the active edge of the CLK signal during which the control input must be maintained at the proper level. The situation is illustrated in Fig 72.21 (*a*) for a flip-flop that triggers on the rising edge. Usually, the IC manufacturers specify the minimum allowable set up time, t_S (min). If this time requirement is not met, the flip-flop may not respond reliably when the clock edge occurs.

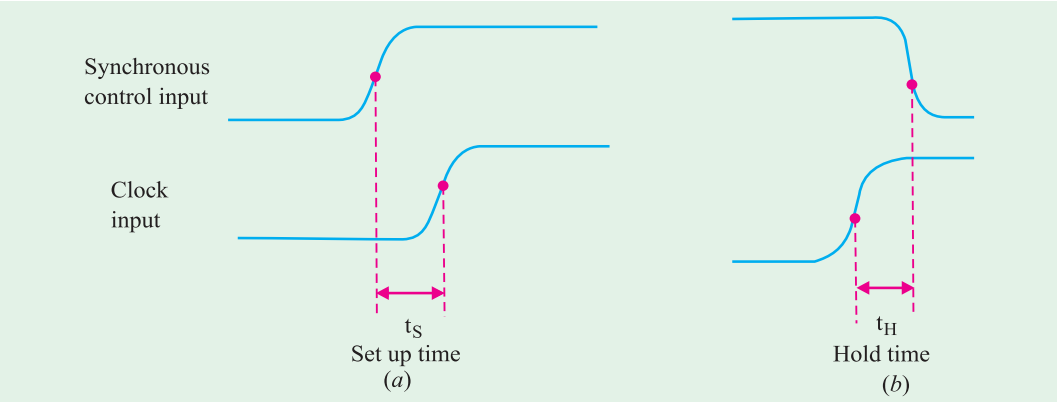


Fig. 72.21. Illustrating the set-up and hold time requirements, for a flip-flop that triggers on the rising edge.

2. Hold time, t_H. It is the time interval immediately following the active edge of the CLK signal during which the synchronous control input must be maintained at the proper level. Refer to Fig. 72.21 (*b*) usually the IC manufacturers specify the minimum acceptable value of hold time, t_H (min). If this requirement is not met, the flip-flop will not trigger reliably.

It is evident from the above discussion that in order to ensure that a clocked flip-flop will respond properly when the active clock edge occurs, the control inputs must be stable (*i.e.* unchanging) for at least a time interval.

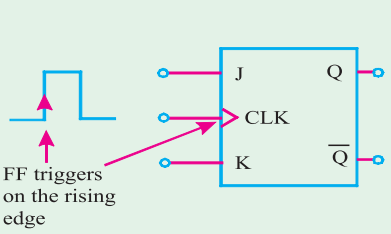
1. t_S (min) prior to the active clock edge.
2. t_H (min) after the active clock edge.

A typical value of t_S (min) is in the range of 5 to 50 ns whereas hold times are generally from 0 to 10 ns. It may be carefully noted that the set up and hold times are measured between the 50 per cent points on the edges.

The set up and hold time requirements are extremely important in synchronous systems. This is due to the reason that there will be many situations where the synchronous control inputs to a flip-flop are changing at approximately the same time as the CLK input.

72.11. Clocked *S* - *C* Flip-Flop

Fig. 72.22 (*a*) shows the logic symbol for a clocked *S* - *C* flip-flop that is triggered by the rising edge of the clock signal. In other words this flip-flop can change output states only when a signal applied to its clock input makes a transition from 0 to 1. The *S* and *C* inputs control the state of the flip-flop in the same manner as discussed earlier for the NOR latch. But it may be noted that the flip-flop does not respond to these inputs until the occurrence of the rising edge of the clock signal.



(*a*) Symbol

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

(*b*) Truth table

Fig. 72.22. Clocked *S*-*C* Flip-Flop.

Fig. 72.22 (*b*) shows the truth table for a clocked *S-C* flip-flop that is triggered by the rising edge of the clock signal. The truth table indicates how the flip-flop output will respond to the rising edge at the CLK input for the various combinations of *S* and *C* inputs. The up arrow (↑) in the truth table indicates that the rising edge is required at CLK input. The label Q_0 indicates the level at *Q* prior to the rising edge. This nomenclature is used quite often by IC manufacturers in their IC data sheets/ manuals.

The operation of *S-C* flip-flop may be understood with the help of input and output waveforms shown in Fig 72.23. Assuming that the setup and hold time requirements are being met in all cases, the waveforms can be analysed as follows :

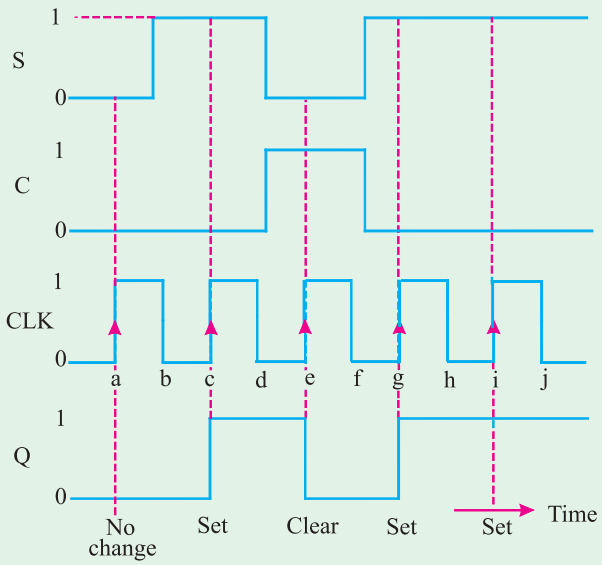
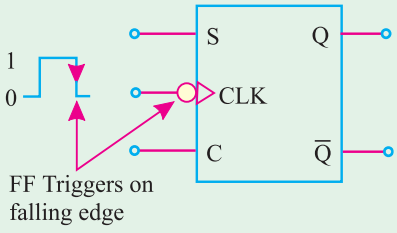


Fig. 72.23. Typical input and output waveforms of a *S-C* flip-flop.

1. Notice that initially all the inputs (S, C and CLK) are 0 and the *Q* output is assumed to be zero, *i.e.* $Q_0 = 0$.
2. When the rising edge of the first clock pulse occurs (refer to point '*a*'), both the *S* and *C* inputs are 0, so the flip-flop output is not affected and remains in the $Q = 0$ state (*i.e.* $Q = Q_0 = 0$).
3. At the occurrence of the rising edge of the second clock pulse (refer to point '*c*'), the *S* input is now HIGH, with *C* input still LOW. This causes the flip-flop output to set to 1 state.
4. At the occurrence of the rising edge of the third clock pulse (refer to point '*e*'), the *S* input is LOW with *C* input HIGH. This causes the flip-flop output to clear to 0 state.
5. The rising edge of the fourth clock pulse sets the flip-flop output to the $Q = 1$ state (refer to point '*g*') because $S = 1$ and $C = 0$.
6. The fifth clock pulse also finds that $S = 1$ and $C = 0$ at the rising edge (refer to point '*i*'). This situation will produce HIGH output. But since *Q* is already HIGH, so it remains in that state.

It may be carefully noted from the waveforms shown in Fig. 72.23, that the flip-flop is not affected by the falling edge of the clock pulses. It may also be noted that *S* and *C* input levels have no effect on the flip-flop except upon the occurrence of a rising edge of the clock signal.



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

(*a*) Symbol (*b*) Truth table

Fig. 72.24. Falling-edge triggered S-C flip-flop.

Fig. 72.24 (*a*) shows the symbol and 72.24 (*b*) the truth table for a clocked *S* - *C* flip-flop that triggers on the falling edge of the CLK input. Notice the presence of a small circle and a triangle on the CLK input of the flip-flop. These indicate that this flip-flop will trigger only when CLK input goes from 1 to 0. This flip-flop operates in the same way as the rising-edge flip-flop except that the output can change states only on the falling edge of the clock pulses (refer to points *b, d, f, h* and *j* in Fig. 72.23). In actual practice both the rising-edge and falling-edge triggered flip-flops are used in digital systems.

Example 72.5. *Fig.* 72.25 (*a*) *shows the SET and CLEAR waveforms applied at the inputs of the clocked S-C flip-flop shown in Fig.* 72.25 (*b*)

Sketch the waveforms at Q and $\overline{Q}$ *outputs of the flip-flop. Assume that initially Q* = 0 *and* $\overline{Q}$ = 1.

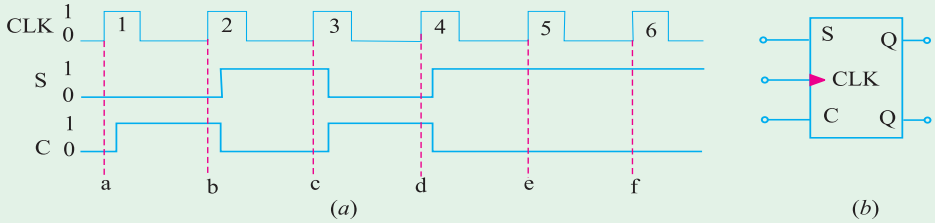


Fig. 72.25

Solution. Given :

In order to determine the *Q*-output, we will refer to the waveforms applied at the flip-flop inputs shown in Fig. 72.25 (*a*) and its truth table shown in Fig. 72.24 (*b*). At point '*a*', $S = C = 0$, therefore $Q_0 = 0$, At point '*b*', $S = 0$ and $C = 1$, therefore *Q* remains 0. At point '*c*', $S = 1$ and $C = 0$, therefore $Q = 1$. At point '*d*', $S = 0$ and $C = 1$, therefore $Q = 0$. At point '*e*', $S = 1$, $C = 0$, therefore $Q = 1$. At point '*f*', *S* and *C* inputs remain the same, therefore *Q* also remains 1. The sketch of *Q*-output along with the CLK, *S* and *C* waveforms is as shown in Fig. 72.26. The $\overline{Q}$ -output waveform is determined by inverting the *Q*-waveform.

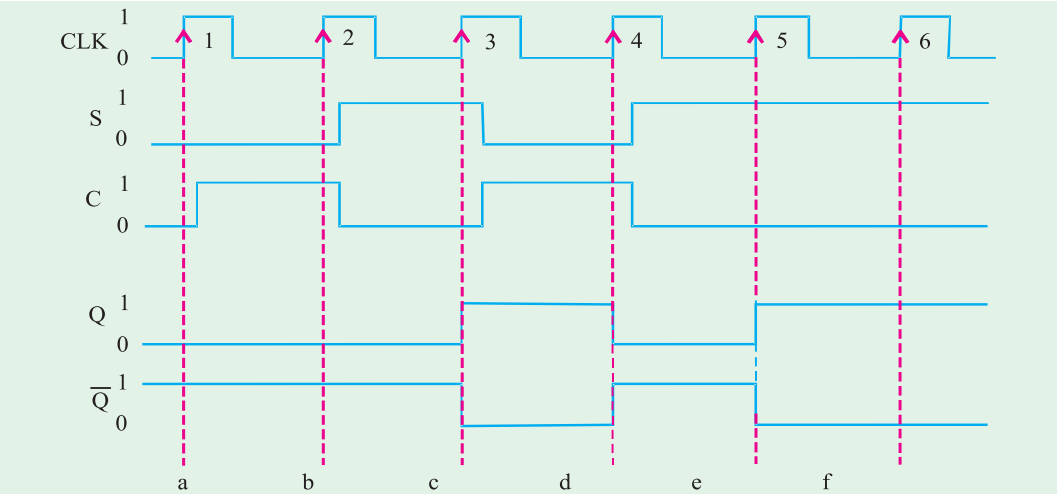


Fig. 72.26

72.12. Internal Circuitry of an Edge-triggered S-C Flip-Flop

Fig . 72.26 shows a simplified version of an internal circuitry of an edge triggered *S-C* flip-flop. Notice that the circuit contains the following three sections :

1. A basic NAND latch
2. A pulse steering circuit and
3. An edge-detector circuit

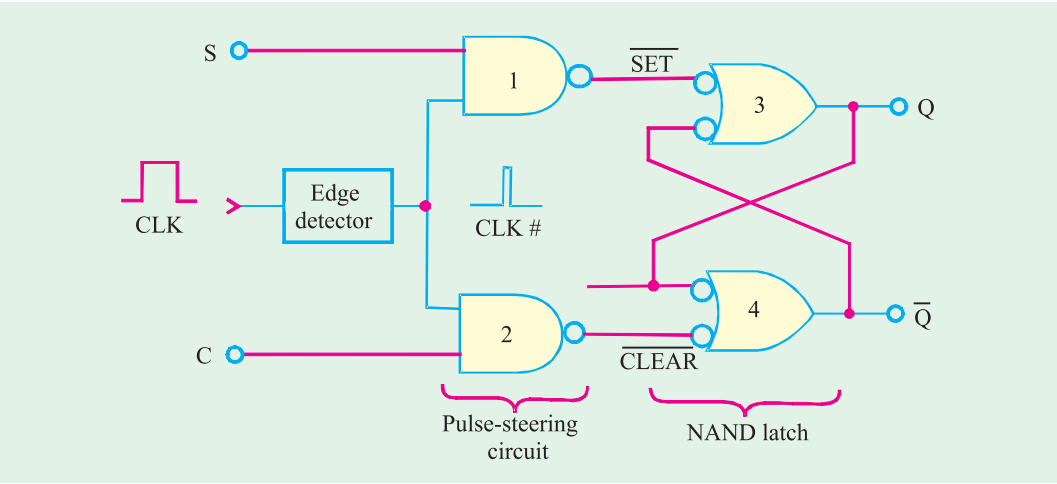


Fig. 72.27. Internal circuitry of an edge-triggered S-C flip-flop.

As seen from the Fig. 72.27, the edge detector produces a narrow positive-going spike (CLK #). The pulse-steering circuit "steers" (or moves) the spike through to the SET and CLEAR input of the latch in accordance with the levels present at *S* and *C* inputs. For example with $S = 1$ and $C = 0$, the spike is inverted and passed through NAND gate-1 to produce a LOW pulse at the SET input of the latch that sets $Q = 1$. With $S = 0$ and $C = 1$, the spike is inverted and passed through NAND gate-2 to produce a LOW pulse at the CLEAR input of the latch that resets $Q = 0$.

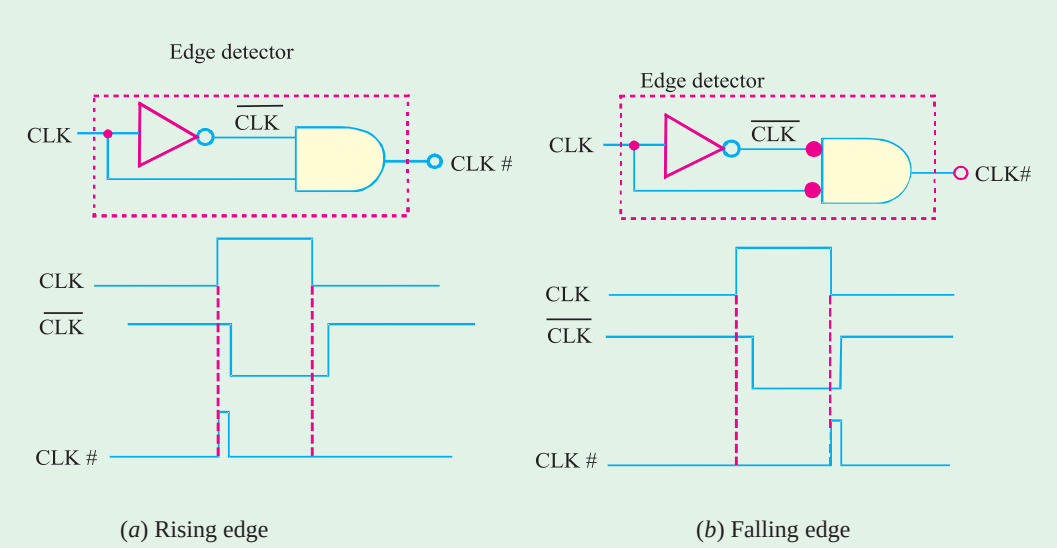


Fig. 72.28. Implementation of edge-detector circuits used in edge triggered flip-flops.

Fig. 72.28 shows the implementation of edge-detector circuits used in edge-triggered flip-flops. Thus Fig. 72.28 (*a*) shows how the spike is generated for edge-triggered flip-flops that trigger on the rising edge of the CLK pulse. As seen from this figure, the INVERTER produces an output with a delay of a few nanoseconds. Because of this the transitions of $\overline{CLK}$ occur a little bit after that of CLK. Next the AND gate produces an output spike that is HIGH only for the few nanoseconds when both CLK and $\overline{CLK}$ are HIGH. As a result of this, we get a narrow pulse at edge detector output (CLK #) which occurs on the rising edge of the CLK.

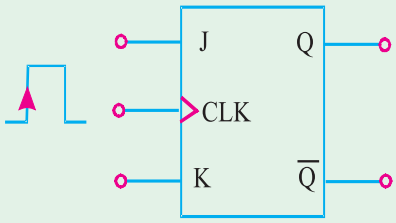
Similarly Fig 72.28(*b*) produces CLK # on the falling edge of CLK for flip-flops that are to trigger on the falling edge.

It may be noted that since the CLK # signal is HIGH for only a few nanoseconds, the *Q* output is affected by the levels at *S* and *C* only for a short time during and after the occurrence of the active edge of CLK. This gives the flip-flop its edge-triggered property.

72.13. Clocked *J-K* Flip-Flop

Fig. 72.28 (*a*) shows the symbol and (*b*) the truth table for a clocked *J-K* flip-flop that is triggered by the rising edge of the clock signal. The *J* and *K* inputs control the state of the flip-flop in the same manner as the *S* and *C* inputs do for the *S-C* flip-flop. However there is one major difference-the $J = K = 1$ condition in *J-K* flip-flop does not result in an ambiguous output unlike $S = C = 1$ in *S-C* flip-flop for, $J = K = 1$ condition, the J-K flip-flop will always go to its opposite state upon the rising edge of the clock signal. This is called ***trigger mode.*** In this mode, if both *J* and *K* are left HIGH, the flip-flop will change states (i.e. toggle) for each rising edge of the clock.

The operation of *J-K* flip-flop for each combination of *J* and *K* is summarized in Fig. 72.29 (*b*) Notice that the truth table is the same except for $J = K = 1$ 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 rising edge of the clock pulse. It is called toggle operation.



(*a*) Symbol

Inputs			*Output*
J	*K*	*CLK*	*Q*
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	Q_0 (toggles)

(*b*) Truth table

Fig. 72.29. Rising edge triggered J-K flip-flop.

In order to understand the operation of J-K flip-flop, let us consider the *J*, *K* and CLK waveforms as shown in Fig. 72.30. Assume that set up and hold time requirements are met. The operation may be explained as below.

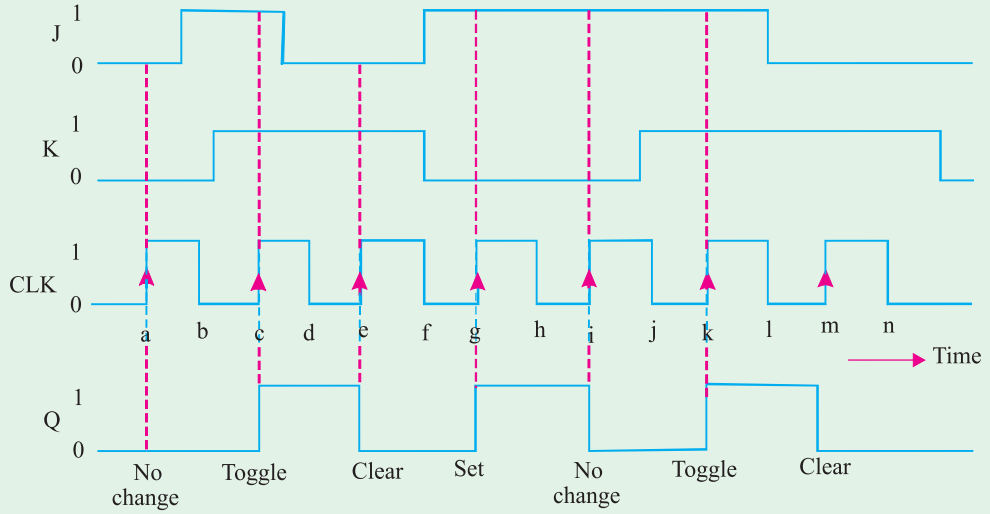
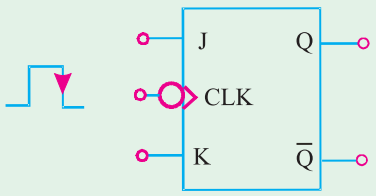


Fig. 72.30

1. Initially all the inputs are 0, and the *Q* output is also assumed to be 0, *i.e.* $Q_0 = 0$.
2. When the rising edge of the first clock pulse occurs (refer to point *a*), the $J = K = 0$ condition exists. Thus the flip-flop does not change its output state, *i.e.* $Q = Q_0 = 0$.
3. When the rising edge of the second clock pulse occurs (refer to point *c*), the $J = K = 1$ condition exists. Thus the flip-flop toggles to its opposite state *i.e.* $Q = \overline{Q}_0 = \overline{0} = 1$.
4. When the rising edge of the third clock pulse occurs (refer to point *e*), $J = 0$ and $K = 1$ condition exists. Thus the flip-flop is cleared to the $Q = 0$ state.
5. When the rising edge of the fourth clock pulse occurs (refer to point *g*), $J = 1$ and $K = 0$ condition exists. This condition sets the output *Q* to 1 state.
6. When the rising edge of fifth clock pulse occurs (refer to point *i*), $J = 1$ and $K = 0$ condition exists. This is the condition that sets the output *Q* to 1 state. However since *Q* is already 1, so it will remain there. Hence no change in the output state.
7. When the rising edge of six clock pulse occurs (refer to point *k*), $J = K = 1$ condition exists. This condition causes the flip-flop to toggle to its opposite state.

8. When the rising edge of seventh clock pulse occurs (refer to point m), $J = 0$ and $K = 1$ condition exists. This condition causes the flop-flop to clear to $Q = 0$ state.

It may be noted from the waveforms that the flip-flop is not affected by the falling edge of the clock pulses. Also the *J* and *K* input levels have no effect except the occurrence of the rising edge of the clock signal. The *J* and *K* inputs by themselves cannot cause the flip-flop to change states.



(*a*) Symbol

Inputs			*Output*
J	*K*	*CLK*	*Q*
0	0	↓	Q_0 (no change)
1	0	↓	1
0	1	↓	0
1	1	↓	Q_0 (toggles)

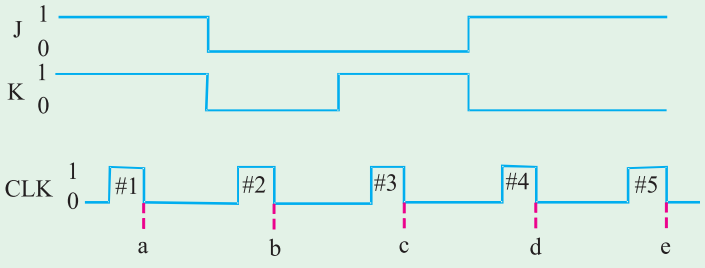
(*b*) Truth table

Fig. 72.31. Falling edge triggered J-K flip-flop.

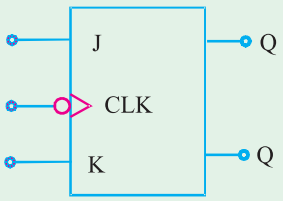
Fig. 72.31 shows the symbol and 72.31 (*b*) the truth table for a clocked *J-K* flip-flop that triggers on the falling edge of the clock pulse. The small circle on the CLK input indicates that the flip-flop will trigger when the CLK input goes from 1 to 0. This flip-flop operates in the same way as the rising edge of the flip-flop of Fig. 72.27 except that the output can change states only on the falling edge of CLK signal (*i.e.* points *b, d, f, h, j,* l and *n*). As a matter of fact, both polarities of edge-triggered *J-K* flip-flops are in common usage in the field of digital electronics.

Strictly speaking, the *J-K* flip-flop is much more versatile than the *S-C* 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.

Example 72.6. Fig 72.32 (*a*) *shows the waveforms applied at J, K and CLK inputs of the clocked J-K flip-flop shown in Fig.* 72.32 (*b*). *Sketch the Q output waveform Assume* $Q = 0$ *initially.*



(*a*)



(*b*)

Fig. 72.32

Solution :

Notice that flip-flop shown in Fig. 72.32 (*b*) is a clocked *J-K* flip-flop. Also notice the presence of a small circle at the CLK input of the flip-flop. This indicates that the flip-flop will trigger coresponding to the falling edge of the clock pulse. So you need to identify the states of *J* and *K* inputs

coresponding to the falling edge points of the CLK pulse waveform *i.e.* points *a, b, c, d* and *e* shown in Fig. 72.32.

Using the truth table of clocked *J-K* flip-flop shown in Fig. 72.31, and the given waveforms the *Q*-waveform sketch is shown in Fig. 72.33. On the arrival of first CLK pulse at point '*a*' $J = K = 1$, the a output does not change, *i.e.* it stays at $Q = 1$ level. On the arrival of third CLK pulse *J-K* flip-flop will toggle, i.e., its Q-output changes from 0 to 1. On the arrival of second CLK pulse, at point '*b*', $J = K = 0$ the *Q*-at point '*c*' $J = 0$, $K = 1$, the Q output goes 0. On the arrival of fourth CLK pulse, at point '*d*' $J = 1$, $K = 0$, the *Q*-output goes 1. On the arrival of fifth CLK pulse at point '*e*' $J = 1$, $K = 0$, the *Q*-output remains at 1 level.

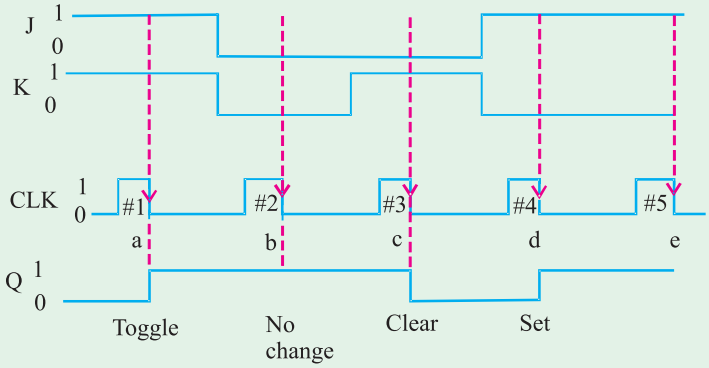


Fig. 72.33

Example 72.7. *What will be the output waveform Q of a J-K flip-flop if the following waveforms are applied at the input? Assume the flip-flop triggers at the falling edge of clock pulse.*

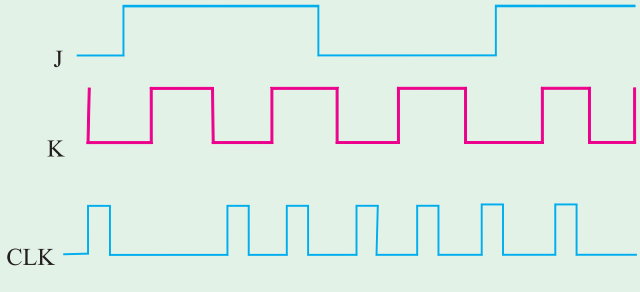


Fig. 72.34

(Grad. IETE Dec. 1996)

Solution :

Recall the truth table of a clocked *J-K* flip-flop, the flip-flop triggers corresponding to the logic levels at the *J* and K inputs. If $J = K = 1$, the flip-flop output remains in its previous state. If $J = 1$, $K = 0$ the flip-flop output goes 1, if $J = 0$, $K = 1$, the flip-flop goes LOW. However, if $J = K = 1$, the flip-flop output toggles. Keeping it in mind, we can sketch the *Q*-output waveform as shownin Fig. 72.35.

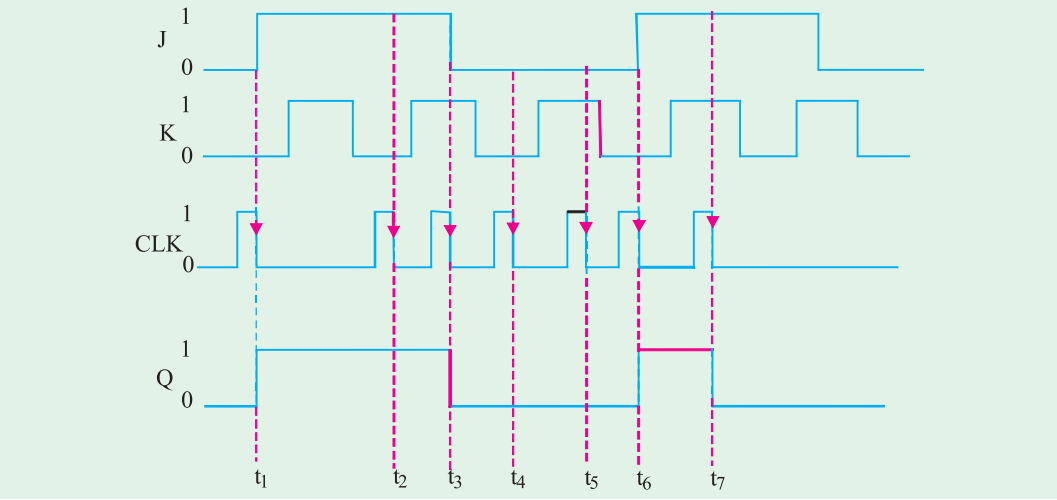


Fig. 72.35

Notice that at t_1, $J = 1, K = 0$ therefore the Q–output goes 1. At t_2, $J = 1, K = 0$ so the Q–output remains 1. At t_3, J goes 0, $K = 1$, therefore the Q-output goes 0. At t_5, J is still 0, K is still 1, therefore the Q-output stays 0. At t_6, J goes 1, K is 0, so the Q-output goes 1. At t_6 both J and K are 1, therefore the Q-output toggles and goes 0, as shown in the Fig. 72.35.

72.14. Internal Circuitry of an Edge-triggered *J-K* Flip-Flop

Fig. 72.36 shows a simplified version of the internal circuitry of an edge-triggered *J-K* flip-flop. As seen, the flip-flop contains the same three sections as the *S-C* triggered flip-flop (refer to Fig 72.27 page 2646), *i.e.*,

1. A basic NAND gate latch
2. A pulse steering circuit and
3. An edge detector circuit

The only difference between the internal circuitry of edge-triggered *J-K*. flip-flop and that of *S-C* flip flop is that in *J-K* flip-flop circuit, Q and $\overline{Q}$ outputs are fed back to the pulse-steering NAND gates.

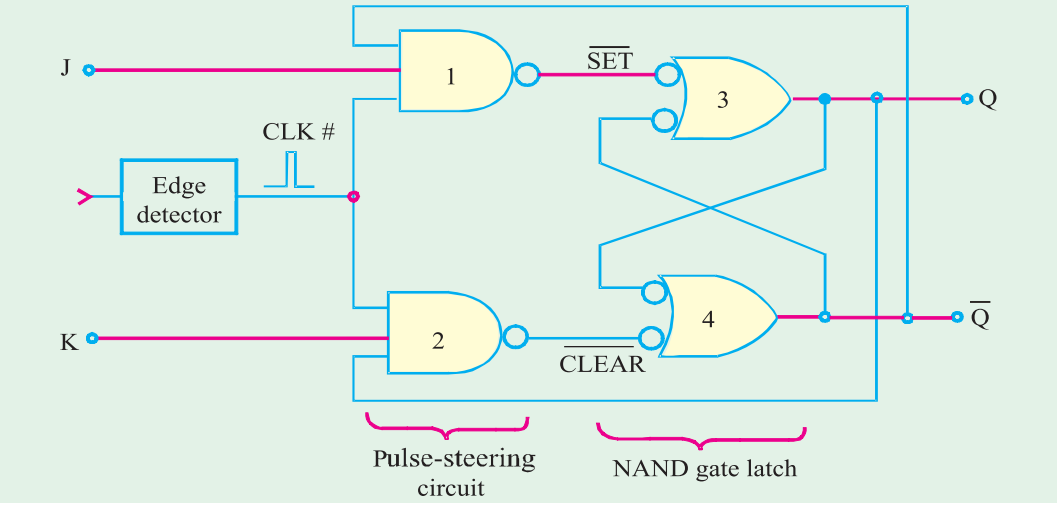


Fig 72.36. A simplified version of an internal circuitry of an edge-triggered *J-K* flip-flop.

It is because of this feedback connection that J-K flip-flop gives the toggle operation for $J = K = 1$ condition.

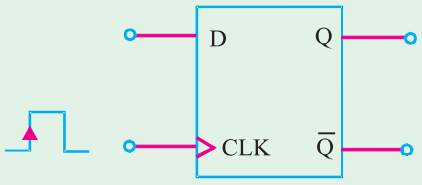
Let us examine the toggle condition in more detail. Assume that $J = K = 1$ and that $Q = 0$ when a CLK pulse occurs. With $Q = 0$ and $\overline{Q} = 1$, NAND gate 1 will steer CLK # (inverted) to $\overline{\text{SET}}$ input of the NAND gate latch to produce $Q = 1$.

If we assume $Q = 1$. When a CLK pulse occurs, NAND gate 2 will steer CLK # (inverted) to the $\overline{\text{CLEAR}}$ input of the latch to produce $Q = 0$. It is evident from the above discussion that Q always ends up in the opposite state.

Note. It may be carefully noted that in order for toggle operation to work, the CLK # pulse must be very narrow. It must return to 0 before the Q and $\overline{Q}$ outputs toggle to their new states. Otherwise the new states of Q and $\overline{Q}$ will cause the CLK # pulse to toggle the latch outputs again.

72.15. Clocked *D* Flip-Flop

Fig. 72.37 (*a*) shows the symbol and 72.37 (*b*) the truth table for a clocked *D* flip-flop that triggers on the rising edge of the clock pulse. Notice that this flip-flop has only one synchronous control input, *D* which stands for data.



(*a*) Symbol

Inputs		*Output*
D	*CLK*	*Q*
0	↑	Q
1	↑	1

(*b*) Truth table

Fig. 72.37. Clocked D flip-flop.

The operation of the clocked *D* flip-flop is very simple. The output *Q* will go to the same that is present on the D input when the rising edge occurs at the CLK. In other words, the level present at D will be stored in the flip-flop at the instant the rising edge occurs.

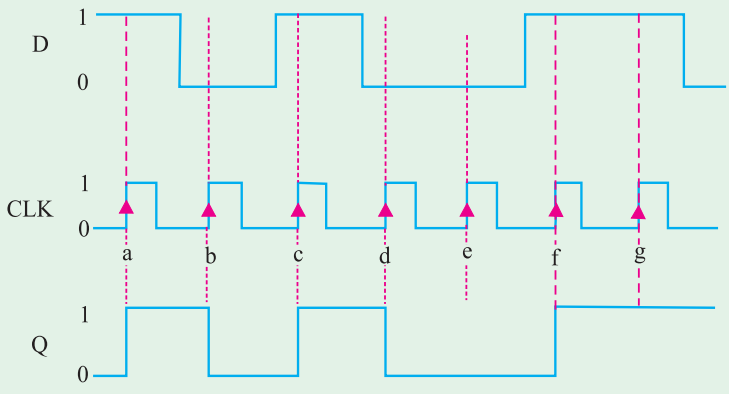


Fig. 72.38. Input and output waveforms to illustrate the operation of clocked *D* flip-flop.

In order to understand the operation of flip-flop in more detail, consider the waveforms at *D* and CLK input as shown in Fig. 72.38. Assume that *Q* is initially 0.

1. When the first rising edge of the CLK pulse occurs (refer to point '*a*'), the *D* input is 1, therefore *Q* will go to 1 state. Even though the *D* input level changes between the points '*a*' and '*b*', it has no effect on *Q*. The output *Q* is storing the 1 that was on *D* at point '*a*'.
2. When the second rising edge of the CLK pulse occurs (refer to point '*b*'), *Q* goes to 0 state since *D* is 0 at that time. The output *Q* stores this 0 value until the rising edge of the third CLK pulse (at point '*c*') causes *Q* to go to 1 since *D* is 1 at that time.
3. In a similar manner, the *Q* output takes on the levels present at *D* when the rising edges occur at points '*d*', '*e*', '*f*' and '*g*'. Notice that *Q* stays 0 at point '*e*' because *D* is still 0.

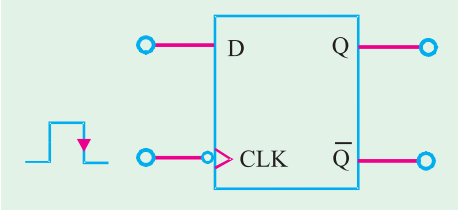


Fig. 72.39. Symbol for a falling-edge triggered *D* flip-flop.

A falling-edge triggered *D* flip-flop operates in the same way as the rising edge triggered *D* flip-flop. However, the difference is that *Q* will take on the value of *D* when a falling edge occurs at the CLK.

A falling-edge triggered *D* flip-flop operates in the same manner as the *D* flip-flop discussed above except that *Q* will take on the value of *D* when a falling edge occurs at the CLK input. The symbol for D flip-flop that triggers on the falling edge has a bubble on the CLK input as shown in Fig. 72.39.

IC 7474 is an example of clocked *D* flip-flop. It contains two rising edge triggered D flip-flops. The pin configuration and some other specification can be found in the data sheet of the device.

72.16. Implementation of *D* Flip-Flop from a *J-K* Flip-Flop

An edge-triggered *D* flip-flop can be obtained easily by adding a single INVERTER to an edge-triggered *J-K* flip-flop as shown in Fig. 72.40. A similar approach can be used to convert a *S-C* flip-flop to a *D* flip-flop.

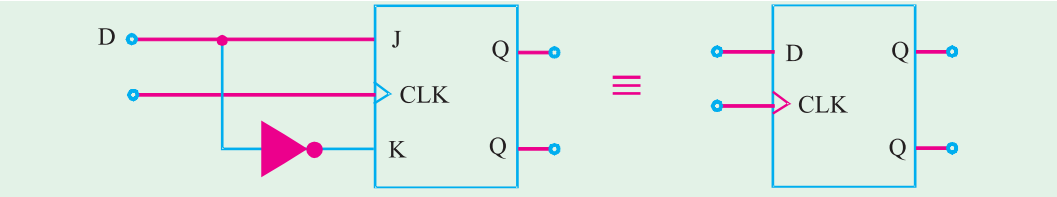


Fig. 72.40. Implementation of *D* flip-flop from a *J-K* flip-flop.

Example 72.8. *Fig* 72.41 *shows a D flip-flop with the input and clock waveforms applied at their respective inputs. Determine the Q* (*or output*) *waveform.*

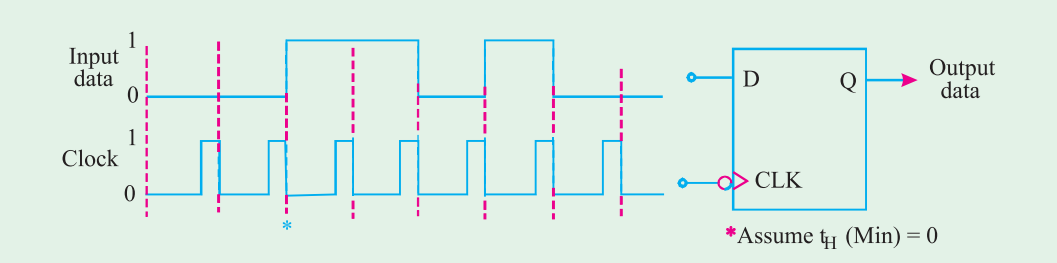


Fig. 72.41

Solution :

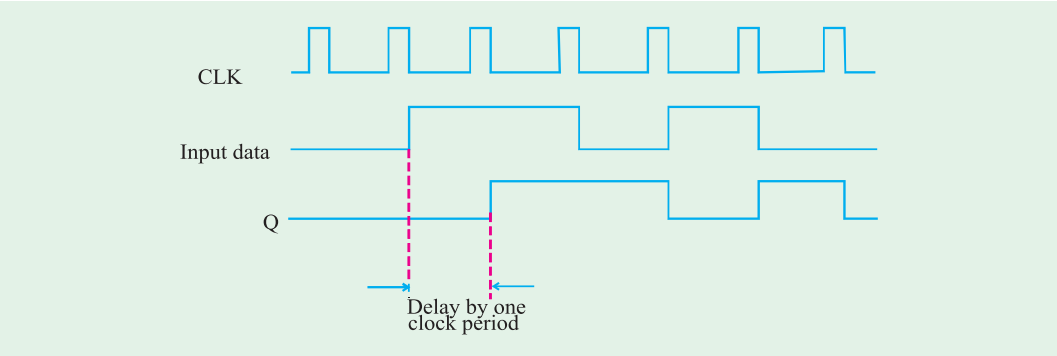


Fig. 72.42

Note. The output (Q) is delayed from the input by one clock period. This is an advantage as a D-Flip-flop is used sometimes to delay a binary waveform so that the binary information appears at the output a certain amount of time after it appears at the D input.

It is also possible to delay the input by two clock periods. This can be achieved by connecting Q to the D input of a second flip-flop and connect the clock signal to the second flip-flop. The output of the second flip-flop will be delayed by 2 clock periods from the input data.

Example 72.9. *An edge-triggered D Flip-flop can be made to operate in the toggle mode by connecting it as shown in Fig.* 72.43. *Assume* $Q = 0$ *initially and determine the* Q (*output*) *waveform.*

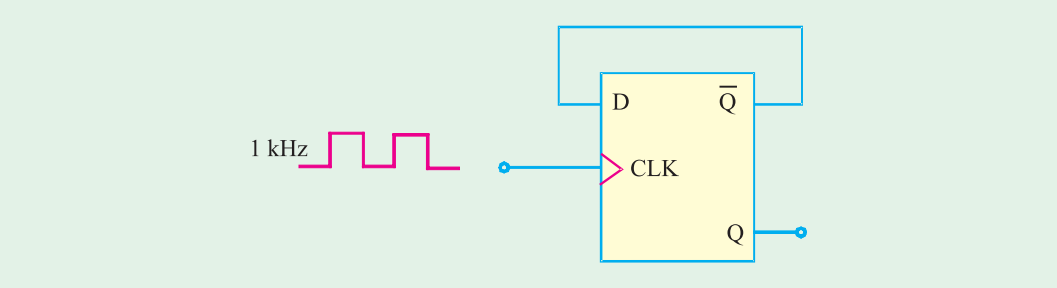


Fig. 72.43

Solution : The clock is 1 kHz waveform. The output waveform can be obtained from the input as shown in Fig. 72.44.

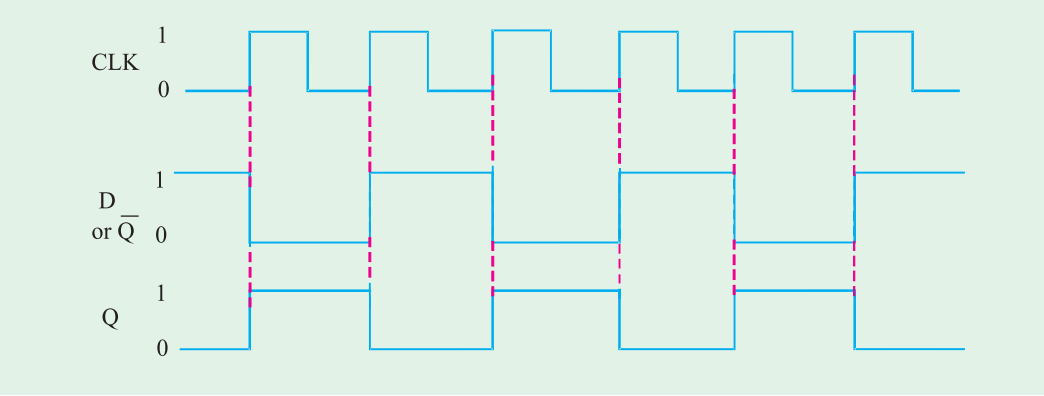


Fig. 72.44

Notice that the output is a square wave of half-the frequency of input *i.e.* 7200 kHz.

Note. A *D*- flip-flop with $\overline{Q}$ tied to its *D*–input as shown in Fig 72.43 is also known as *T* flip-flop where *T* stands for Toggle (or trigger).

72.17. Parallel Transfer of Data Using *D*-Flip-Flops

We have already discussed in Art 72.15 that the *Q*-output of a *D* flip-flop is the same as the *D* input. Let us now study the usefulness of this flip-flop.

Fig. 72.45 shows an application of *D* flip-flop used for parallel transfer of binary data from *X*, *Y*, *Z* – the three outputs of a combinational logic circuit to the outputs Q_1, Q_2, and Q_3 of the *D* flip-flops for storage. The transfer occurs upon application of TRANSFER pulse to the common CLK inputs. The flip-flops can store these values for subsequent processing.

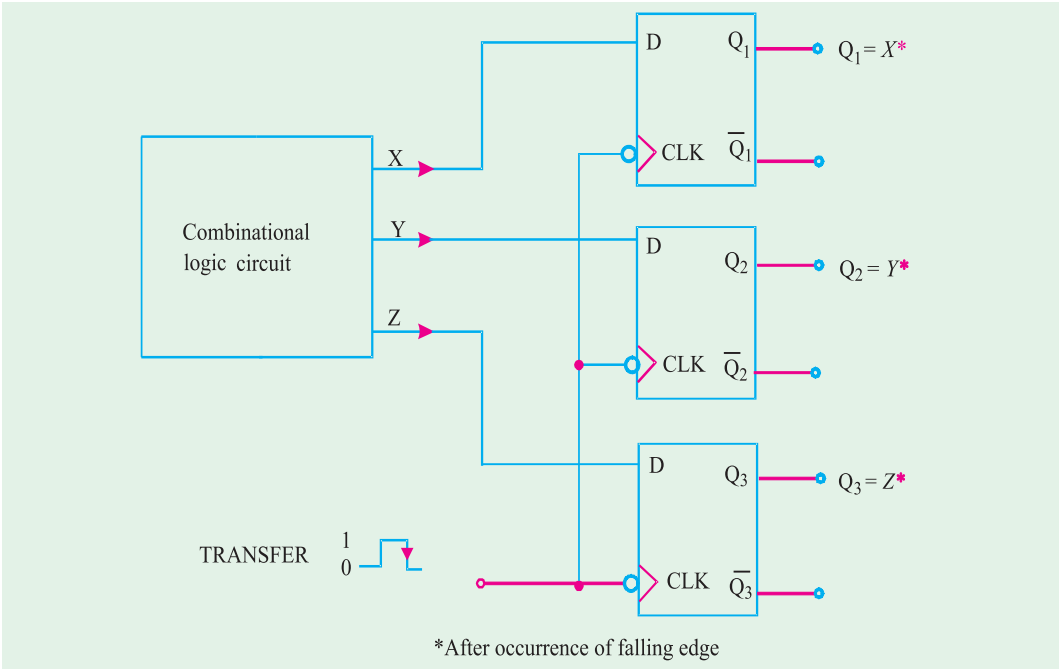
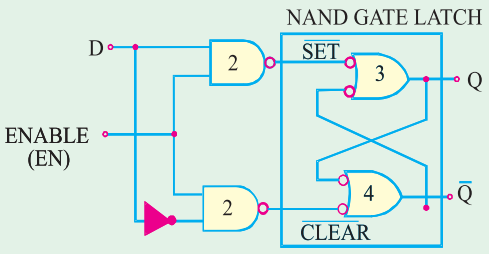


Fig. 72.45. Illustrating parallel transfer of data using *D* flip-flops.

72.18. *D* Latch (Transparent Latch)

We have already discussed in Art. 72.15 about an edge – triggered *D* flip-flop. Such a 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 circuit is not used, the resultant circuit operates somewhat differently. It is called a *D* latch and has the arrangement shown in Fig 72.46 (*a*).



(*a*) Circuit arrangement

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

(*b*) Truth table

Fig. 72.46. D latch.

As seen from Fig. 72.46. (*a*), the circuit contains a NAND gate latch, and the steering NAND gates without the edge-detector circuit. The common input to the steering gates is called an enable input (abbreviated as EN) rather than a clock input because its effect on Q and $\overline{Q}$ outputs is not restricted to occurring only on its transitions. The operation of the *D* latch may be explained as follows:

1. When EN is LOW, whatever be the value of *D* input, it is inhibited from affecting the NAND gate latch. It is due to the reason that the outputs of both the 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 is represented as Q_0 in the truth table. Refer to Fig. 72.46 (*b*).

2. When EN is HIGH, the *D* input will produce a LOW at either $\overline{\text{SET}}$ or the $\overline{\text{CLEAR}}$ inputs of the NAND gate latch to cause Q to become the same level as *D*. If *D* changes while EN is HIGH, Q will follow the change exactly. In other words, while EN = 1, the Q output will look exactly like *D*. Because of this reason, the *D* latch in this mode is called "transparent".

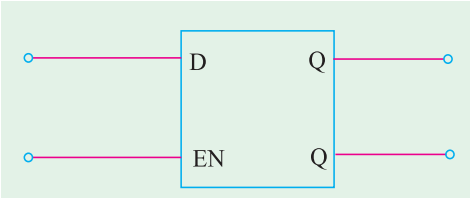


Fig. 72.47. Logic symbol for a D-latchs.

The operation is summarized in the truth table shown in Fig. 72.46 (*b*). The logic symbol for the *D*-latch is shown in Fig. 72.47. It may be noted that even though the EN input operates in the same way as CLK input of an edge-triggered flip-flop, 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 an edge occurs. So remember the *D* latch is not an edge-triggered device.

IC 7475 is an example of *D*-latch. It contains four transparent *D*-latches.

Example 72.10. *Fig.* 72.48 *shows the waveforms applied at the D and EN inputs of a D latch. Sketch the output waveform at Q-output Assume initiually* $Q = 0$

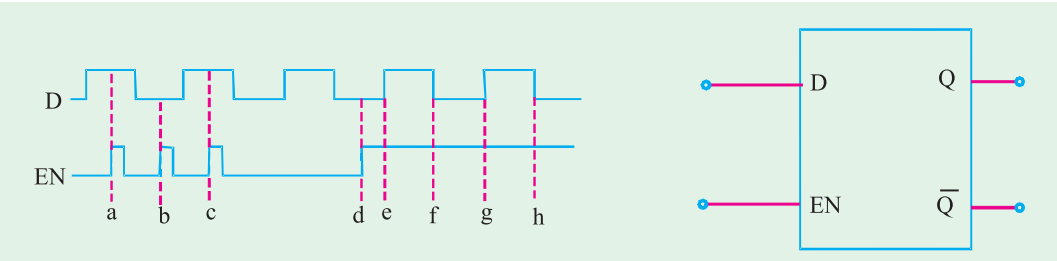


Fig. 72.48

Solution.

Using the truth table of *D*-latch shown in Fig. 72.46, and the input waveforms, the output waveform at Q can be obtained and is sketched as shown in Fig. 72.49. Notice that at point '*a*', EN goes 1, and *D* is 1, therefore a output goes 1. At point '*b*' EN is 1, but $D = 0$, therefore Q goes 0, At point '*c*', EN is 1, therefore Q goes 1 and so on.

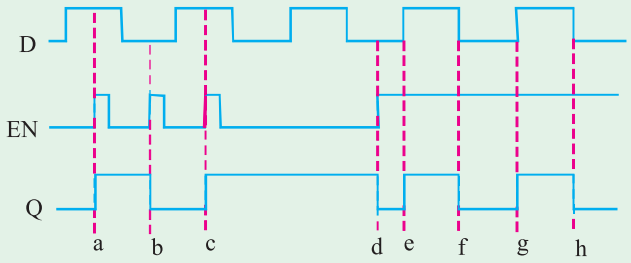
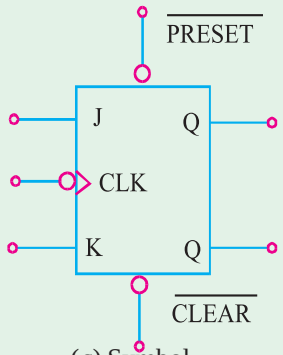


Fig. 72.49

72.19. Clocked *J-K* Flip-Flop with Asynchronous Inputs

Strictly speaking for the clocked flip-flops, the *S, C, J, K* and *D* inputs discussed in the previous articles are refered to as ***control inputs.*** These inputs are also known as ***synchronous inputs.*** These are called synchronous because their effect on the flip-flop output is synchronized with the CLK (*i.e.* clock) input. As discussed earlier, the synchronous control inputs must be used in conjunction with a signal to trigger the flip-flop.

As a matter of fact, most clocked flip-flops also have one-or more ***asynchronous*** inputs. These inputs operate independently of the synchronous inputs and clock input. The asynchronous inputs of a flip-flop can be used to set its output to 1 state or clear to the 0 state at any time regardless of the conditions at the other inputs. In other words, the asynchronous inputs can be used to ***override*** all the other inputs in order to place the flip-flop output in one state or the other. Because of this reason, the asynchronous inputs are also called ***override inputs.***



(*a*) Symbol

Inputs		*Output / Flip-flop response*
$\overline{PRESET}$	$\overline{CLEAR}$	
1	1	Clocked operation*
0	1	Q = 1 (regardless of CLK)
1	0	Q = 0 (regardless of CLK)
0	0	Not used

(*b*) Truth table

Fig . 72.50

Fig. 72.50 shows a clocked J-K flip-flop with two asynchronous inputs namely (1) $\overline{\text{PRESET}}$ and (2) $\overline{\text{CLEAR}}$. Both these inputs are active-LOW. This is indicated by the bubbles on the flip-flop symbol. Fig 72.50 (*b*) shows the truth table for the clocked *J-K* flip-flop with asynchronous inputs. Let us study the various cases given in the truth table :

1. **$\overline{\text{PRESET}}$ = $\overline{\text{CLEAR}}$ = 1.** This condition indicates that both the asynchronous inputs are inactive and the flip-flop is free to respond to the *J, K* and CLK inputs. In other words, the flip-flop operates as a normal clocked flip-flop.
2. **$\overline{\text{PRESET}}$ = 0, $\overline{\text{CLEAR}}$ = 1.** This condition indicates that the asynchronous input, $\overline{\text{PRESET}}$ is activated. Because of this, the output, *Q* is immediately set to 1, no matter what conditions are present at the *J, K* and CLK inputs. It may be carefully noted that the CLK input cannot affect the flip-flop while $\overline{\text{PRESET}}$ = 0.

3. $\overline{\text{PRESET}} = 1$, $\overline{\text{CLEAR}} = 0$. This condition indicates that the asynchronous input, $\overline{\text{CLEAR}}$ is activated. Because of this, the output, Q is immediately set to 0, no matter what conditions are present at the J, K and CLK inputs. It may be carefully noted that the CLK input cannot affect the flip-flop while $\overline{\text{CLEAR}} = 0$.
4. $\overline{\text{PRESET}} = \overline{\text{CLEAR}} = 0$. This condition should not be used because it can produce an ambiguous response.

It will be interesting to know that the $\overline{\text{PRESET}}$ and $\overline{\text{CLEAR}}$ inputs respond to dc levels. This means that if a constant 0 is held on the $\overline{\text{PRESET}}$ input, the flip-flop will remain in the $Q = 1$ state regardless of what is occurring at the other inputs. Similarly, a constant 0 at the $\overline{\text{CLEAR}}$ input holds the flip-flop in $Q = 0$ state. Thus the asynchronous inputs can be used to hold the flip-flop in a particular state for any desired interval of time. However, in actual practice, the asynchronous inputs are used to set or clear the flip-flop to the desired state by application of a momentary pulse.

Many clocked flip-flops that are commercially available as ICs, will have both $\overline{\text{PRESET}}$ and $\overline{\text{CLEAR}}$. However some ICs will have only $\overline{\text{CLEAR}}$ input. Some other ICs will have asynchronous inputs that are active-HIGH. For these ICs, the flip-flop symbol would not have a bubble on the asynchronous inputs.

ICs 7476 and 74LS76 are popular *J-K* flip-flops with asynchronous inputs preset and clear. The preset is designated by $\overline{S}_D$ and clear by $\overline{R}_D$. The IC 7476 is a rising edge triggered while 74LS76 is a falling edge triggered device. Each package contains two *J-K* flip-flops.

Example 72.11. Fig. 72.51 (*a*) *shows the logic symbol for a J-K flip-flop that responds to the falling edge on its clock pulse and has active-LOW asynchronous inputs. The J and K inputs are tied HIGH.*

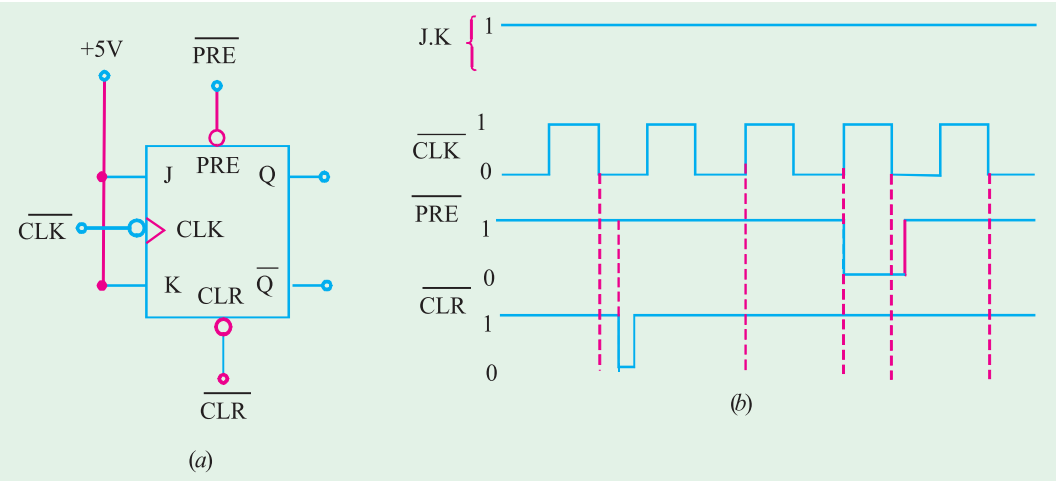


Fig. 72.51

Determine the Q-output in response to the waveforms shown in Fig. 72.51 (*b*). *Assume that initially* $Q = 0$.

Solution. Notice that the *J* and *K* are tied to +5V. The waveforms shown in Fig. 72.51 (*b*) are applied at the inputs of the *J-K* flip flop.

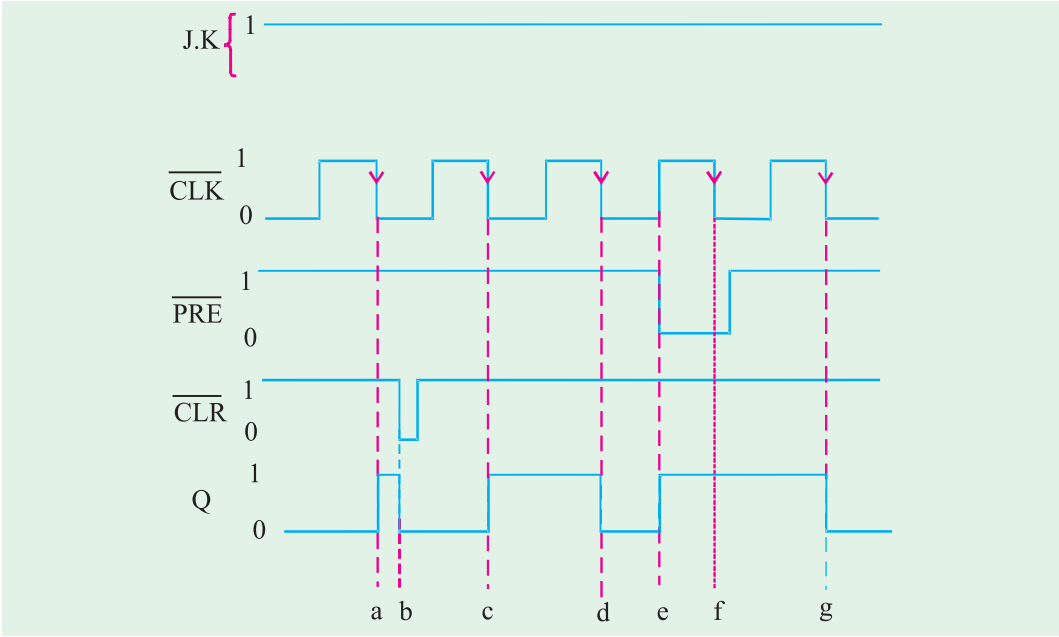


Fig. 72.52

In order to sketch the *Q*-output waveform, recall the truth table of *J-K* flip-flop indicated in Fig. 72.31 (*b*) and Fig. 72.50 (*b*). Notice that when $\overline{PRE}=1$ and $\overline{CLR}=1$, the *J-K* flip-flop will trigger at the falling edge of the clock pulse. Thus at point '*a*', $\overline{PRE}=\overline{CLR}=1$, $J=K=1$, therefore the *Q*-output toggles *i.e.* it goes from 0 to 1. At point '*b*' $\overline{CLR}$ goes 0, therefore *Q* goes 0. At point '*c*', $\overline{PRE}=\overline{CLR}=1$, therefore under normal operation *J-K* flip-flop toggles *i.e.* *Q* goes from 0 to 1 again. At point '*d*' there is no change in $\overline{PRE}$ or $\overline{CLR}$ inputs. So the *Q*-output toggles again, *i.e.* *Q* goes 0. At point '*e*' $\overline{PRE}$ goes 0, therefore *Q*-output goes 1. At point '*g*', $\overline{PRE}=\overline{CLR}=1$, and *Q* toggles to 0. The complete *Q*-output waveform is shown in Fig. 72.52 along with the input waveforms.

Example 72.12. *Determine the Q-output for the JK flip-flop shown in Fig.* 72.53, *Assume that* $Q = 0$ *initially and remember that the asynchronous inputs override all other inputs.*

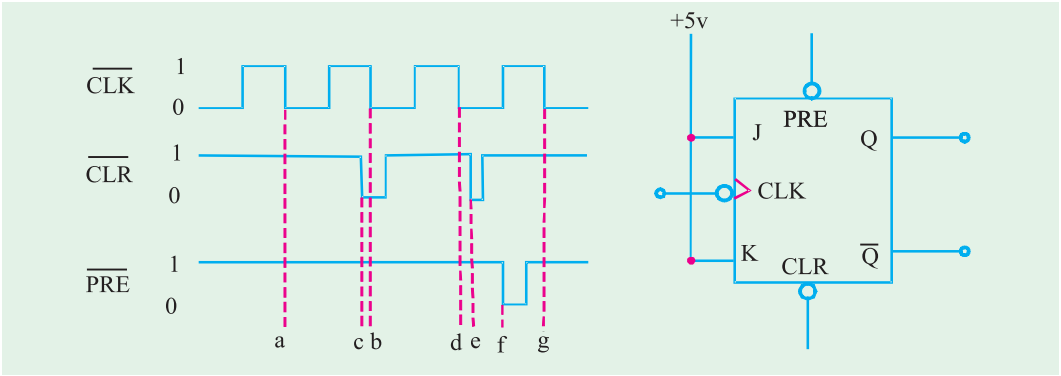


Fig. 72.53

Solution :

The waveform at the *Q*-output can be obtained by recalling the truth-table of a *JK* flip-flop. Shown in Fig. 72.31 (*b*) and 72.50 (*b*).

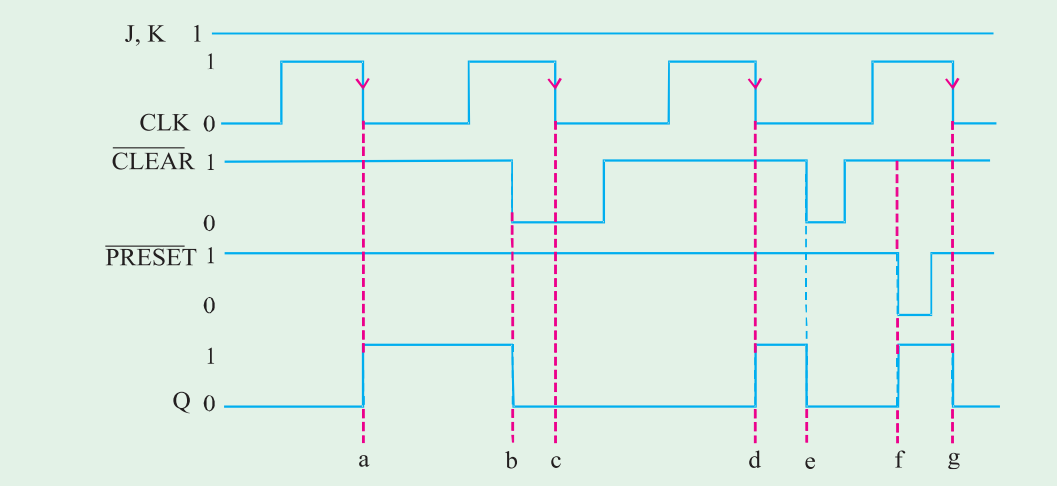


Fig. 72.54

Notice that at point '*a*', $J = K = 1$, $\overline{\text{PRESET}} = \overline{\text{CLEAR}} = 1$, the falling edge of CLK pulse toggles the *J-K* flip-flop *Q* output to 1. At point '*b*' $\overline{\text{CLEAR}}$ goes 0, therefore irrespective of the other inputs, *Q*-output goes 0. At point '*c*' the ouput remains 0 because $\overline{\text{CLEAR}}$ is still 0. At point '*d*', $\overline{\text{PRESET}} = \overline{\text{CLEAR}} = 1$, $J = K = 1$, the *J-K* flip-flop output toggles on the arrival of falling edge of CLK, *i.e. Q* goes 1. At point '*e*', $\overline{\text{CLEAR}}$ goes 0 again, therefore *Q*-output goes 0. At point '*f*', $\overline{\text{PRESET}}$ goes 0, therefore *Q*-output goes 1. Finally at point '*g*', $\overline{\text{CLEAR}} = \overline{\text{PRESET}} = 1$, $J = K = 1$, the output toggles on the arrival of falling edge of CLK pulse, *i.e. Q* goes 0.

The complete sketch of *Q*-output waveform along with other input waveforms is shown in Fig. 72.54.

Example 72.13. *In the circuit of Fig.* 72.55, *inputs A, B and C are all initially LOW. Output Y is supposed to go HIGH only when A, B and C goes HIGH in a certain sequence.*

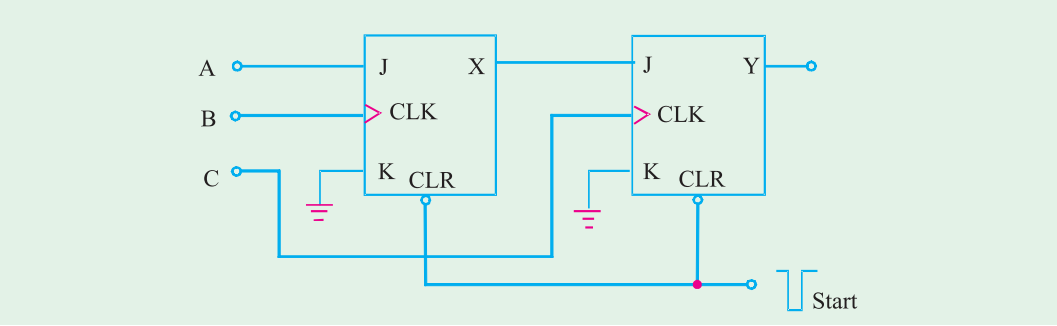


Fig. 72.55

(a) Determine the sequence that will make Y go HIGH.

(b) Explain why the START pulse is needed.

Solution.

(a) Sequence at the inputs (A, B and C)

We know that *Y* can go HIGH only when *C* goes HIGH while *X* is already HIGH.

X can go HIGH only if *B* goes HIGH while *A* is HIGH.

Thus the correct sequence that makes Y go HIGH is A, B, C.

(b) Need for START pulse

We know that the outputs X and Y need to be cleared to 0 before applying the A, B, C signals. To clear the outputs, we need a negative going START pulse at CLR input. (notice that CLR input of JK flip-flop is active LOW).

72.20. Alternative Designations for Asynchronous Inputs

We have already discussed in the last article about the asynchronous clocked *J-K* flip-flop. These flip-flops are available as ICs and have two asynchronous inputs namely $\overline{\text{PRESET}}$ and $\overline{\text{CLEAR}}$ (or PRESET and CLEAR). IC manufacturers have not agreed on what nomenclature is used for these asynchronous inputs. The most common designations are $\overline{\text{PRE}}$ (short for $\overline{\text{PRESET}}$) and $\overline{\text{CLR}}$ (short for $\overline{\text{CLEAR}}$). The designations $\overline{S}_{\text{D}}$ (direct $\overline{\text{SET}}$) and $\overline{R}_{\text{D}}$ (direct $\overline{\text{RESET}}$) are also used. However we will use the labels $\overline{\text{PRE}}$ and $\overline{\text{CLR}}$ to represent asynchronous inputs. Whenever these asynchronous inputs are active–HIGH, we will not use the overbar to indicate their active-HIGH status *i.e.* PRE and CLR.

72.21. Flip-Flop Timing Parameters

The performance, operating requirements and limitations of flip-flops are specified by several timing parameters found on the data sheet for the device refer to Appendix A. Generally the timing parameters are applicable to all CMOS and TTL flip-flops.

1. Setup and Hold Times. These have already been discussed in Art 72.10 (page 2637). They represent requirements that must be met for reliable flip-flop triggering. The manufacturer's IC data sheet will always specify the minimum values of t_S and t_H. For example, IC 7474 has set up time of 20 ns and hold time of 5 ns. On the other hand 74HC112 has set up time of 25ns and zero hold time.

2. Propagation delay. It is the interval of time required after an input signal has been applied for the resulting output change to occur. Fig. 72.56 (*a*) illustrates the propagation delays that occur in response to a rising edge of the CLK input when *Q*-output changes from 0 to 1. Similarly Fig. 72.56 (*b*) illustrates the propagation delay that occur in response to the rising edge of the CLK input when *Q*-output changes from 1 to 0. It may be noted that these delays are measured between the 50% points (between logic 0 and logic 1 voltage levels) on the input and output waveforms. The same type of delays occur in response to the signals on a flip-flops asynchronous inputs (*i.e.* PRESET and CLEAR). The manufacturer's data sheets usually specify propagation delays in response to all inputs and they usually specify the maximum values for t_{PLH} and t_{PHL}. Notice that t_{PLH} is the delay going from logic LOW to HIGH level whereas t_{PHL} is the delay going from logic HIGH to LOW level.

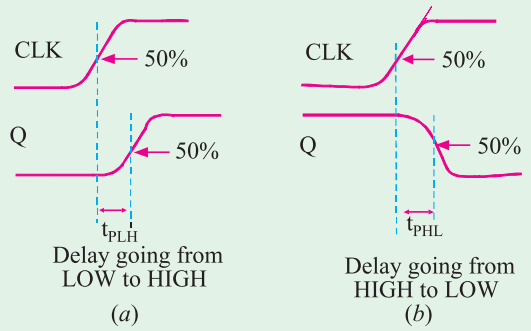


Fig. 72.56

3. Maximum Clock Frequency (f_{max}). This is the highest frequency that may be applied to the CLK input of a flip-flop and still have it triggered reliably. The value of f_{max} limit will vary from flip-flop to flip-flop even with flip-flops having the same device number. For example, f_{max} for IC7474 is 15 MHz, for 74LS112, it is 30 MHz, for 74C74, it is 5 MHz and for 74HC112, it is 20 MHz.

4. Clock Pulse HIGH and LOW Times. The manufacturer will also specify the minimum time duration that the CLK signal must remain LOW before it goes HIGH {represented as $t_w(L)$} and the minimum time that CLK must be kept HIGH before it returns LOW (represented as t_w (H)}. These times are illustrated in Fig. 72.57 (*a*) Failure to meet these minimum time requirements can result in unreliable triggering. It may be noted that these time values are measured between the halfway (50%) points on the signal transitions.

(*a*) Clock Pulse HIGH-LOW times (*b*) Asynchronous active pulse width

Fig. 72.57

5. Asynchronous Active Pulse Width. It is the minimum time duration that a PRESET or CLEAR input must be kept in its active state in order to set or clear the flip-flop reliably. Fig. 72.57 (b) shows $t_w(L)$ for active-LOW asynchronous inputs.

6. Clock Transition Times. Strictly speaking, the rise and fall times of a clock waveform should be kept very short for reliable triggering. If the clock signal takes too long to change from one logic level to the other, the flip-flop may trigger erratically or not at all. IC manufacturers usually do not list a maximum transition time requirement for each flip-flop integrated circuit. Instead, it is usually given as a general requirement for all ICs within a given logic family. For example, the transition times should generally be ≤ 50 ns for TTL devices and ≤ 200 ns for CMOS. These requirements will vary among the different manufacturers and among the various subfamilies within the broad TTL and CMOS logic-families.

72.22. IC Flip-Flop Timing Values

We have already discussed the various flip-flop timing parameters in the last article. Table 72.1. lists all these timing values for the various flip-flops such as 7474, 74 LS112, 74C74 and 74 HC112. Notice that 7474 is a dual edge triggered *D* flip-flop (Standard TTL) device whereas 74LS74 is a dual edge-triggered *J-K* flip-flop (low-power schottky TTL) device. Similarly 74C74 is a dual edge-triggered D flip-flop (metal-gate CMOS) device whereas 74HC112 is a dual edge-triggered *J-K* flip-flop (high-speed CMOS) device.

Table 72.1. IC flip-flop timing values

Parameter	*TTL*		*CMOS*	
(Times in ns)	*7474*	*74LS112*	*74C74*	*74HC112*
t_S	20	20	60	25
t_H	5	0	0	0
t_{PHL} from *CLK* to *Q*	40	24	200	31
t_{PLH} from *CLK* to *Q*	25	16	200	31
t_{PHL} from $\overline{CLR}$ to *Q*	40	24	225	41
t_{PLH} from $\overline{PRE}$ to *Q*	25	16	225	41
t_W(L) *CLK* LOW time	37	15	100	25
t_W(H) *CLK* HIGH time	30	20	100	25
t_W(L) at $\overline{PRE}$ or $\overline{CLR}$	30	15	60	25
f_{Max} in MHz	15	30	5	20

Following are some of the important points which may be noted carefully from Table 72.1.

1. The 74HC series of CMOS devices has timing values that are comparable to those of TTL devices. This can be observed by reading the values below the 7474 column and the values below 74HC112 column.
2. The 74C series devices are much slower than the 74HC series. This can be observed by comparing all the minimum timing requirements below the 74C74 and 74HC112 column.
3. All the flip-flops have nonzero set up time requirement.
4. All of the flip-flops have very low hold-time (t_H) requirements *e.g.* The IC7474 has t_H = 5 ns whereas IC74LS112, 74C74 and 74HC112 has t_H = 0.

Example 72.14. *The data sheet of a certain flip-flop specifies that the minimum HIGH time $t_W(H)$ for the clock pulse is* 30 *ns and the minimum LOW time $t_W(L)$ is 37 ns. What is the maximum operating frequency*?

Solution. Given : $t_W(H)$ = 30 ns = 30 × 10^{-6} s and $t_W(L)$ = 37 ns = 37 × 10^{-6} s.

We know that the minimum time period,

$$t_{min} = t_w(H) + t_w(L)$$
$$= \left(30\times10^{-6} + 37\times10^{-6}\right) = 47\times10^{-6}\,\text{s}$$

and the maximum operating frequency,

$$f_{max} = \frac{1}{t_{min}} = \frac{1}{47\times10^{-6}_{ns}} = 21.276\times10^{6}\ Hz$$

= 21.276 MHz.

Example 72.15. *Using the flip-flop timing values shown in Table* 72.2, *determine the followings* :

Table 72.2.

Parameter	*TTL*		*CMOS*	
(time in ns)	*7474*	*74LS112*	*74C74*	*74HC112*
t_{PHL} from *CLK* to *Q*	40	24	200	31
t_{PLH} from *CLK* to *Q*	25	16	200	31
t_W (L) at $\overline{PRE}$ or $\overline{CLR}$	30	15	60	25

(a) *Assume that initially Q = 0. How long it can take for Q to go HIGH when a rising edge appears at the CLK input of* IC7474?

(b) *Assume that initially Q = 1. How long it can take for Q to go LOW in response to the* $\overline{\text{CLR}}$ *input of an* IC 74HC112?

(c) *What is the narrowest pulse that should be applied to the* $\overline{\text{CLR}}$ *input of the* IC 74LS112 *flip-flop to clear Q reliably?*

Solution.

(a) The rising edge will cause the *Q* output to go from LOW to HIGH. The delay from CLK to *Q* is listed as t_{PLH} = 25 ns for IC7474.

(b) For the IC 74HC112, the time required for *Q* to go from HIGH to LOW in response to $\overline{\text{CLR}}$ input is listed as t_{PHL} = 31 ns.

(c) For the IC 74HC112, the narrowest pulse at the $\overline{\text{CLR}}$ input is listed as $t_w(L)$ = *15* ns.

72.23. Potential Timing Problems in Flip-Flop Circuits

As a matter of fact, in many digital circuits, the output of one flip-flop is connected either directly or through logic gates to the input of another flip-flop and both flip-flops are triggered by the same clock signal. This situation leads to a potential timing problem. In order to understand this problem, consider a situation shown in Fig. 72.58. Here the output of flip-flop (1) Q_1 is connected to the *J* input of flip-flop (2). Notice that both the flip-flops are clocked by the same signal at their CLK inputs.

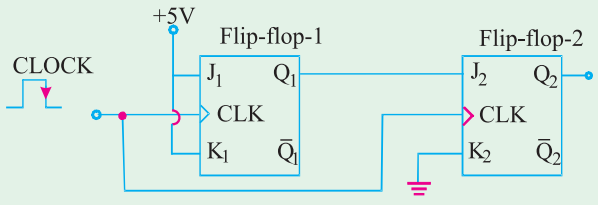


Fig. 72.58

Let us understand the potential timing problem now. Since Q_1 will change on the rising edge of the clock pulse, the J_2 input will be changing as it receives the same rising edge. This could lead to an unpredictable response at Q_2.

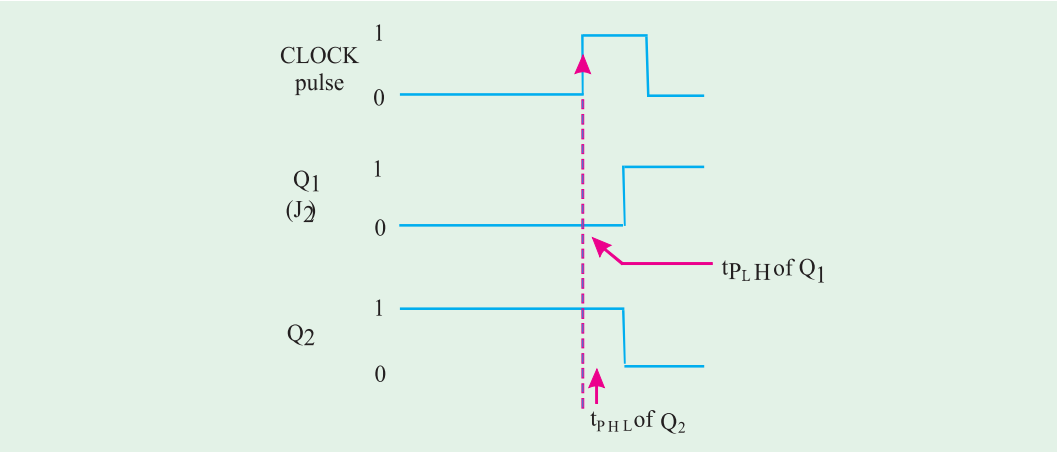


Fig. 72.59

Assume that initially $Q_1 = 0$ and $Q_2 = 1$. Further the flip-flop (1) has $J_1 = K_1 = +5V$ (*i.e* HIGH) and flip-flop (2) has $J_2 = Q_1$ and $K_2 = 0$ prior to the occurrence of rising edge of the clock pulse. When the rising edge occurs, Q_1 will toggle to the HIGH state but it will not actually go HIGH until after its propagation delay, t_{PLH} as shown in Fig. 72.59. The same rising edge will reliably clock Q_2 to the LOW state provided that t_{PLH} is greater than Q_2's hold time requirement, t_H as shown in the figure. If this condition is not met, the response of Q_2 will be unpredictable.

Since all edge-triggered flip-flops have hold time (t_H) requirement that is 5 ns or less. Therefore for those flip-flops, situations like that shown in Fig. 72.59 will not be a problem.

Thus in all of the flip-flop circuits that you will study in this book, we will assume that flip-flop's hold time requirement is short enough to respond reliably according to the following rule:

The flip-flop output will go to a state determined by the logic levels present at its synchronous control inputs just prior to the active clock edge.

Applying the above stated rule to the circuit shown in Fig. 72.59, it says that the output of flip-flop (2), Q_2 will go to a state determined by $J_2 = 1$ and $K_2 = 0$ condition that is present just prior to the occurrence of the rising edge of the clock pulse. The fact that J_2 is changing in response to the same rising edge has no effect.

72.24. Applications of Flip-Flop

Although there is a wide variety of applications of edge-triggered (clocked) flip-flops, yet the following are important from the subject point of view :

1. Flip-flop synchronization
2. Data storage and transfer
3. Serial data transfer-shift registers
4. Frequency division
5. Counting

All these applications are discussed one by one in the following pages.

72.25. Flip-Flop Synchronization

Strictly speaking, most digital systems are principally synchronous in their operation. It means, most of the signals will change states in synchronism with the clock transitions. However, in many cases, there will be an external signal that is not synchronised to the clock. In other words such an external signal is an asynchronous signal. Such signals often occur as a result of a human operator actuating an input switch at some random time relative to the clock signal. This randomness can

produce unpredictable and undesirable results. Now we will study how a flip-flop can be used to synchronize the effect of an asynchronous input.

Fig. 72.60 shows a situation where a signal *X* is generated from a debounced switch that is actuated by an operator. The switch output A goes high when the operator actuates the switch and goes LOW when the operator releases the switch. The switch output *X* is used as an input to control the passage of the control signal through the AND gate so that clock pulses appear at output *Y* only as long as *X* is HIGH.

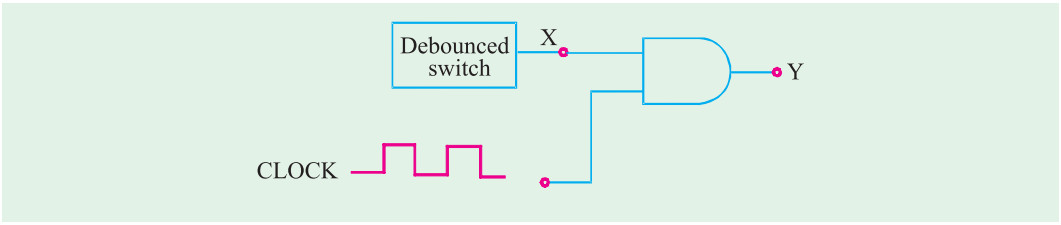


Fig. 72.60

The problem with the circuit shown in Fig. 72.60 is that signal *X* is asynchronous. That is it can change states at any time relative to the clock signal because the exact times when the operator actuates or releases the switch are essentially ***random.*** This can produce partial clock pulses at output *Y* if either transition of *X* occurs while the clock signal is HIGH. It is shown in Fig. 72.61. This type of output is often not acceptable. So a method for preventing the appearance of partial pulses at *Y* must be developed. One possible solution is shown in Fig. 72.61.

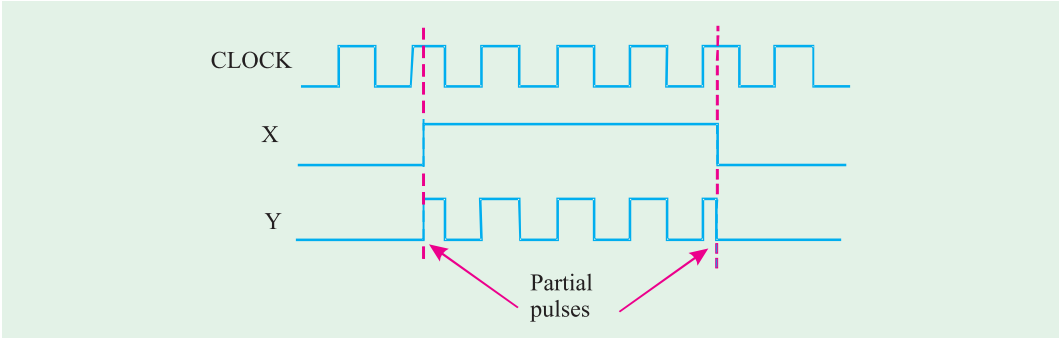


Fig. 72.61

As seen in Fig. 72.62, the *X* signal is connected to the *D* input of a flip-flop. The flip-flop is clocked by the falling edge of the clock signal. Thus when *X* goes HIGH, *Q* will not go HIGH until the next falling edge of the clock at time t_1. This HIGH at *Q* will enable the AND gate to pass subsequent complete clock pulses to *Y* as shown in Fig. 72.61.

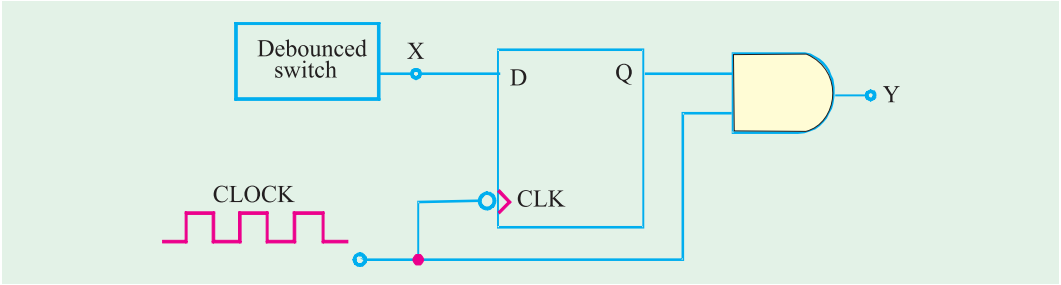


Fig. 72.62

Similarly, when *X* goes LOW, *Q* will not go LOW until the next falling edge of the clock at t_2. Thus the AND gate will not inhibit clock pulses until clock pulse that ends at t_2 has passed through to *Y*. Therefore, output *Y* contains only complete pulses (see Fig. 72.63).

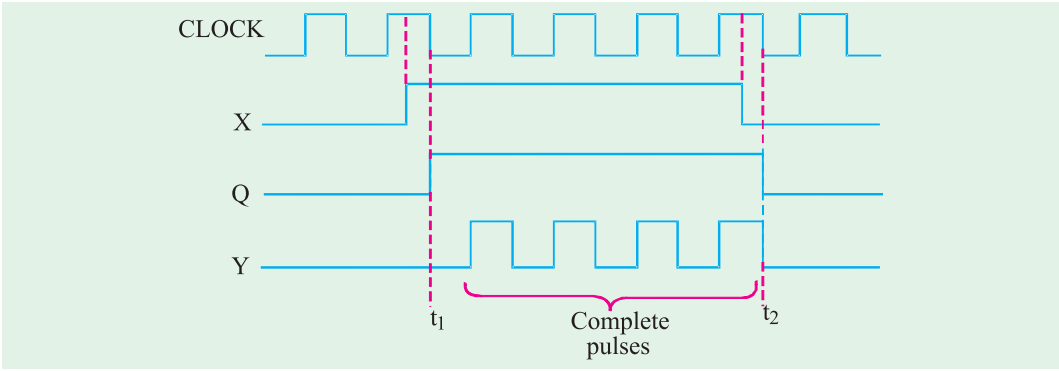


Fig. 72.63

72.26. Data Storage and Transfer

The most common use of flip-flops in the field of digital electronics, is for the storage of data or information. The data may represent numerical values (*e.g.* binary numbers, BCD -binary coded decimal numbers or any data that have been encoded in binary. These data are generally stored in groups of flip-flops called ***registers.*** The operation most often performed on data that are stored in a flip-flop or a register is the ***data transfer*** operation. This operation involves the transfer of data from one flip-flop (or register) to another. The data transfer is of two types: **(*a*)** synchronous and **(*b*)** asynchronous.

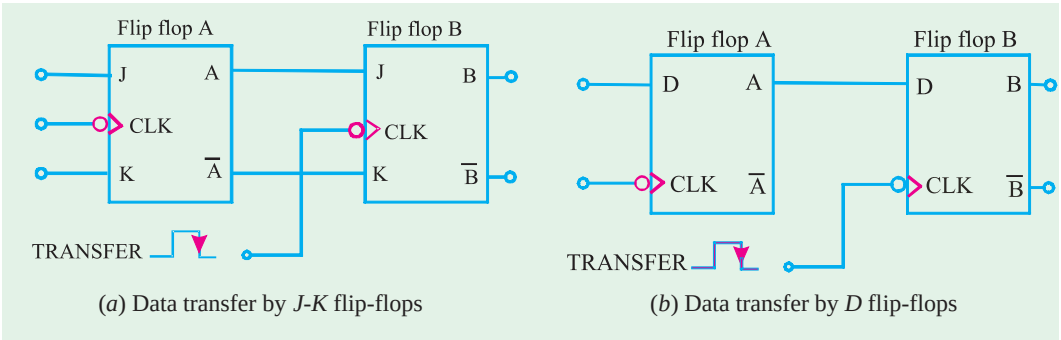


Fig. 72.64

Fig. 72.64 (*a*) and (*b*) shows how synchronous data transfer operation can be accomplished between two flip-flops using *J-K* and *D* flip-flops respectively. In each case, the logic value that is currently stored in flip-flop *A* is transferred to flip-flop *B* upon the falling edge of the TRANSFER pulse. Thus after this falling edge, the *B* output will be the same as the *A* output.

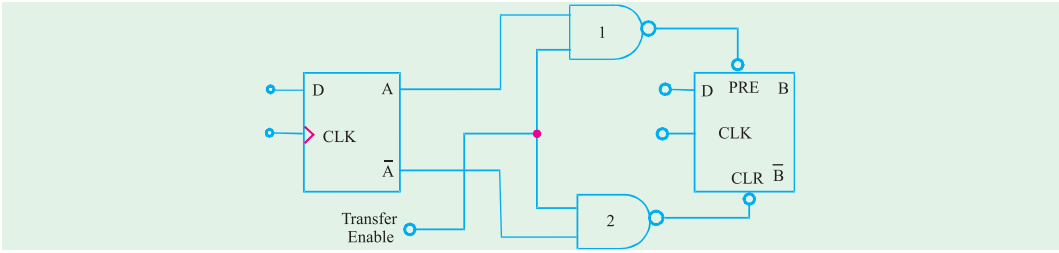


Fig. 72.65. Illustrating asynchronous data transfer operation.

The data transfer illustrated in Fig. 72.64 is called synchronous transfer because the synchronous control and CLK inputs of a flip-flop are used to perform the transfer operation. Fig. 72.64 shows the data transfer that, can be obtained using the asynchronous inputs of a flip-flop. This method called asynchronous transfer can be accomplished using the PRESET and CLEAR inputs of any flip-flop. Notice that flip-flop outputs A and $\overline{A}$ are connected to PRESET and CLEAR through two-input NAND gates 1 and 2 respectively.

In this method, the asynchronous inputs (*i.e.* PRESET and CLEAR) respond to LOW levels. The operation of the circuit may be explained as follows :

When the ***Transfer Enable*** line is held LOW, the two NAND outputs are kept HIGH, with no effect on the flip-flop outputs. But when the ***Transfer Enable*** line is made HIGH, one of the NAND outputs will go LOW depending on the state of A and $\overline{A}$ outputs. This LOW will either set or clear the flip-flop B.

72.27. Parallel Data Transfer

We have already discussed in the last article that flip-flops can be used for data storage as well as for transfer of data from one flip-flop to another. This idea can be further extended for transfer of data from one group of flip-flops (called register A) to another group of flip-flops (called register B). Register A consists of four flip-flops A_1, A_2, A_3 and A_4. Similarly register B also consists of four flip-flops labelled as B_1, B_2, B_3 and B_4 upon application of falling edge of a TRANSFER pulse, the logic level stored in A_1 is transferred to B_1, A_2 is transferred to B_2, A_3 is transferred to B_3 and A_4 is transferred to B_4. It may be noted that the transfer of contents from register A to B occurs at the same time. Because of this reason it is called synchronous transfer or parallel data transfer.

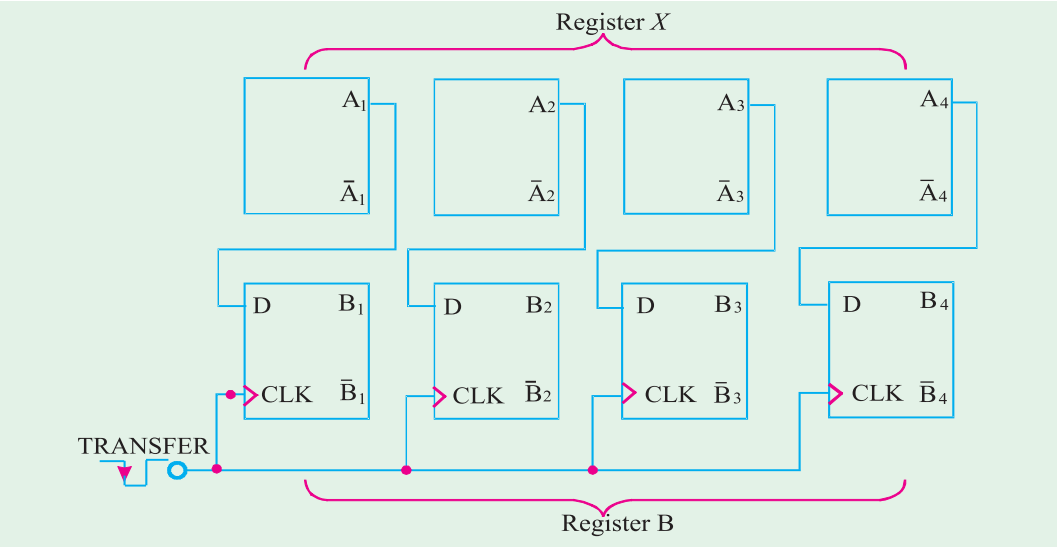


Fig. 72.66. Illustrating parallel data transfer from register X into register Y

It may be carefully noted that parallel data transfer does not change the contents of the register that is the source of data. For example in Fig. 72-66 if $A_1\,A_2\,A_3\,A_4 = 1011$ and $B_1\,B_2\,B_3\,B_4 = 0100$ prior to the occurrence of the TRANSFER pulse, then both registers will be holding 1011 after the TRANSFER pulse. Another point is that the group of four flip-flops indicated above is an example of a basic register used for data storage. In digital systems, data are normally stored in groups of bits (usually 8, 16, 32, 64 …) that represent numbers, codes or other information.

Example 72.16. ***(a)*** *Draw a circuit diagram for the synchronous parallel transfer of data from one three-bit register to another using J-K Flip-flops.* ***(b)*** *Repeat for asynchronous parallel transfer.*

Solution. (*a*) Fig. 72.67 shows the circuit diagram for the synchronous parallel transfer of data from one three-bit register X to another register *Y* using *J-K* flip-flops. Notice that the contents of X_1, X_2, and X_3 are transferred simultaneously into Y_1, Y_2, and Y_3. It may be carefully noted that the circuit allows the parallel transfer of data from the normal outputs as well as the complemented outputs of the flip-flops constituting register *X* .

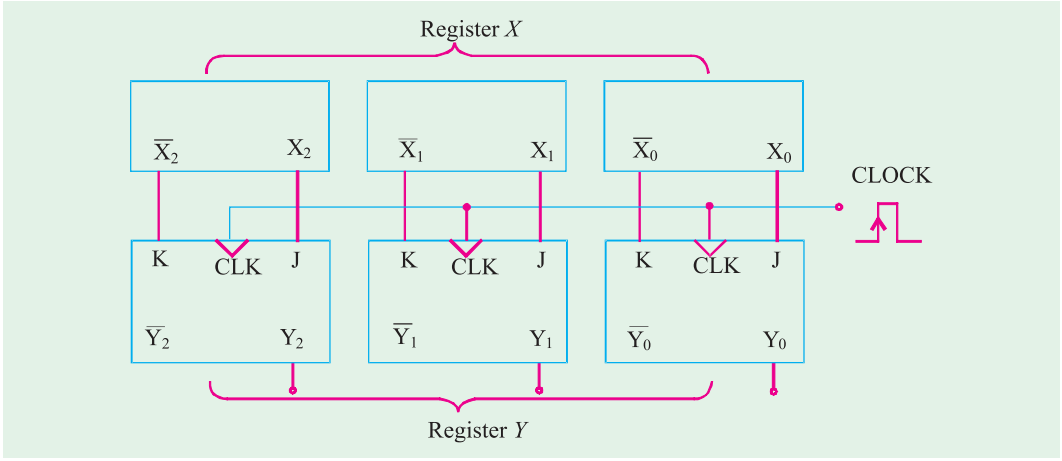


Fig. 72.67. Synchronous parallel transfer of data from one three-bit register to another

(*b*) Fig. 72-68 shows the circuit diagram for the asynchronous parallel data transfer from one three-bit register to another using *J-K* flip-flops. The circuit allows data transfer from either normal outputs X_0, X_1, and X_2 or the complemented outputs $\overline{X}_1$, $\overline{X}_2$ and $\overline{X}_3$.

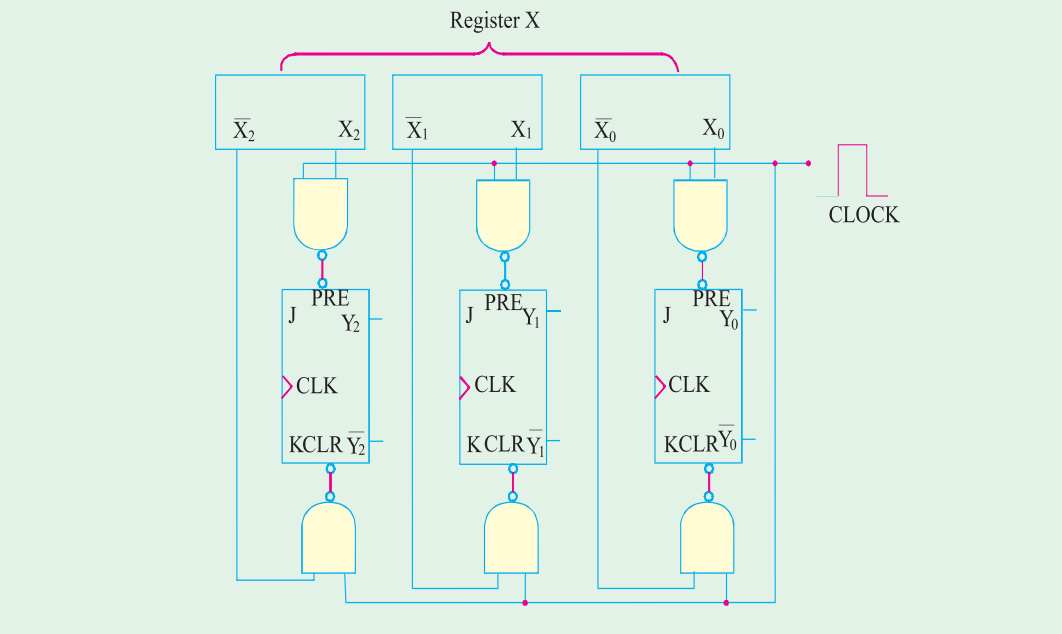


Fig. 72.68. Asynchronous parallel transfer of data from one three-bit register to another.

72.28. Serial Data Transfer : Shift Registers

We have already discussed in the last article that a ***register*** is basically a group of flip-flops to store data. But there is no interconnection between the flip-flops. A ***shift register*** is a group of flip-flops arranged in such a way that the binary numbers stored in the flip-flops are shifted from one flip-flop to the next for every clock pulse. Fig. 72.69 shows a four-bit shift register using *J-K* flip-flops. Notice that the J and K inputs of flip-flops '3' are fed by DATA IN waveform. The *J* and K inputs of flip-flop '2' are fed by the X_3 and $\overline{X}_3$. Similarly *J* and *K* inputs of flip-flop '1' are fed by the X_2 and $\overline{X}_2$ and *J* and *K* inputs of flip-flop '0' are fed by the X_1 and $\overline{X}_1$. The CLK input of all the four flip-flops are connected together to a common input which receives shift pulses. Incidently notice that all the flip-flops trigger on the falling edge of the CLK pulse.

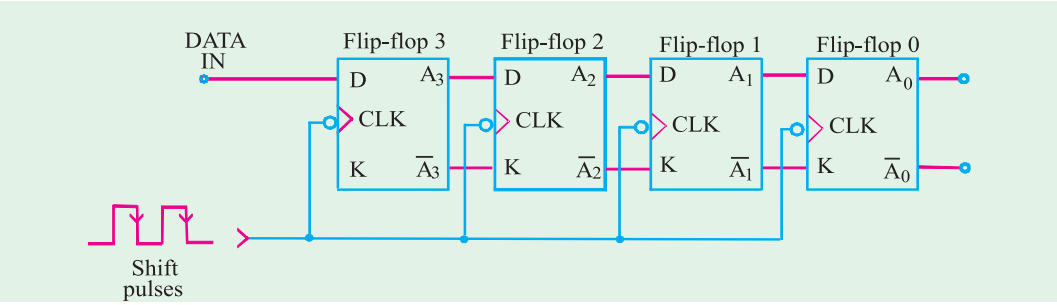


Fig. 72.69. Four-bit shift register.

The operation of the shift register may be understood from the explanation given below. Let us assume that initially all the flip-flops are in the 0 state and input of flip-flop '3' is fed by the DATA IN waveform as shown in Fig. 72.70. The other waveforms shown in Fig. 72.70 indicate how the input data are shifted from left to right from flip-flop to flip-flop as shift pulses are applied. When the first falling edge occurs at t_1, each of the flip-flop outputs A_2, A_1 and A_0 will have $D = 0$ condition present at its inputs because of the state of the flip-flop on its left. The flip-flop '3' will have $D = 1$, because of DATA-IN. Thus at t_1 only A_3 will go HIGH while all the other flip-flop outputs remain LOW. When the second falling edge occurs at t_2, flip-flop '3' will have $D = 0$ because of DATA-IN. The Flip-flop '2' will have $D = 1$ because of current HIGH at A_3. Flip-flops '1' and '0' will still have $D = 0$. Thus at t_2, only flip-flop '2' output A_2 will go HIGH, flip-flop output A_3 will go LOW, and A_2 and A_0 will remain LOW.

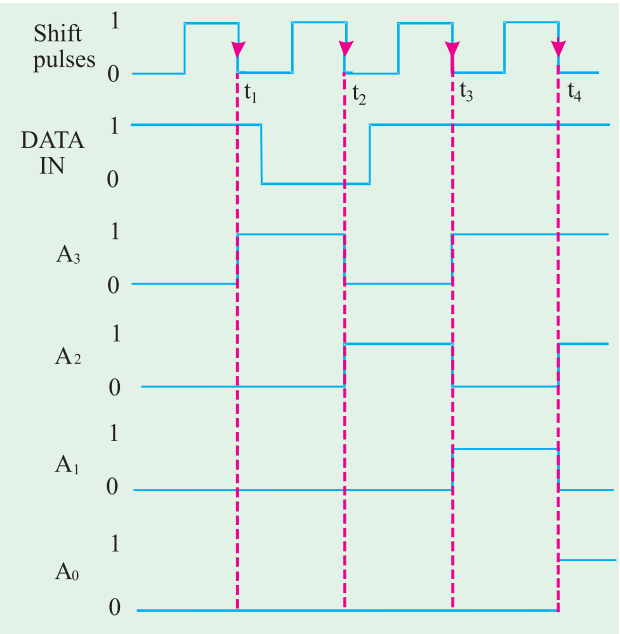


Fig. 72.70. Illustrating serial data transfer.

A similar reasoning can be used to determine how the waveforms change at t_3 and t_4. It may be carefully noted that on each falling edge of the shift pulses, each flip-flop output takes on the level

that was present at the output of the flip-flop on its left just ***prior*** to the falling edge. Of course, flip-flop '3' output X_3 takes on the level that was present at DATA IN just prior to the falling edge.

Note. In the shift register arrangement discussed above, it is necessary that the flip-flops have a very small hold time requirement. This is because of the fact that there are times when the *J* and *K* inputs are changing at about the same time as the CLK edge. For example, the flip-flop '3' output, X_3 changes from 1 to 0 in response to the falling edge at t_2, causing the *J* and *K* inputs of flip-flop '2' output X_2 to change while its CLK input is changing. Actually because of the propagation delay of flip-flop '3', the *J* and *K* inputs of flip-flop '2' won't change for a short time after the falling edge. Because of this reason, a shift register should be implemented using edge-triggered flip-flops that have a t_H value less than one CLK-to-output propagation delay. This later requirement is easily satisfied by most modern edge triggered flip-flops.

72.29. Serial Data Transfer Between Registers

We have already discussed in the last article that a shift register is a group of flip-flops arranged in such a manner that the binary numbers stored in the flip-flops are shifted from one flip-flop to the next after every clock pulse. Now we will study as how data can be serially transferred (or shifted) from one register to the other.

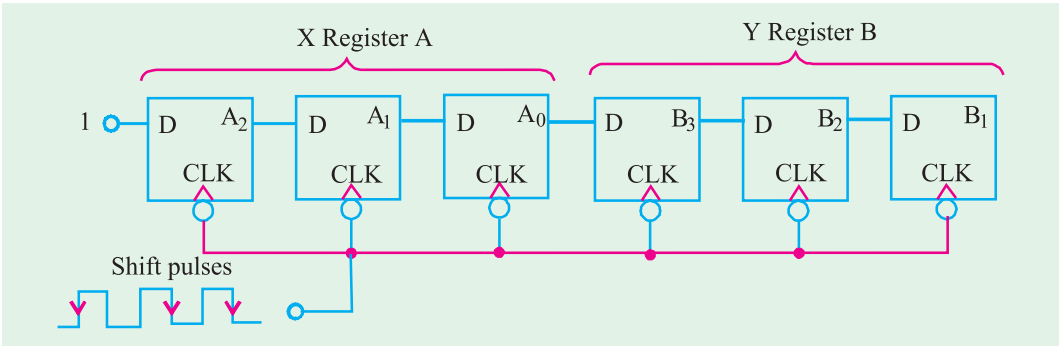


Fig. 72.71

Fig. 72.71 shows three-bit shift registers connected in such a manner that the contents of the A-register will be serially transferred (shifted) into register *B*. Notice how A_0, output of last flip-flop of register *A* is connected to the *D*-input of the first flip-flop of register *B*. The CLK input of all the flip-flops is connected together at one point where the shift pulses are applied.

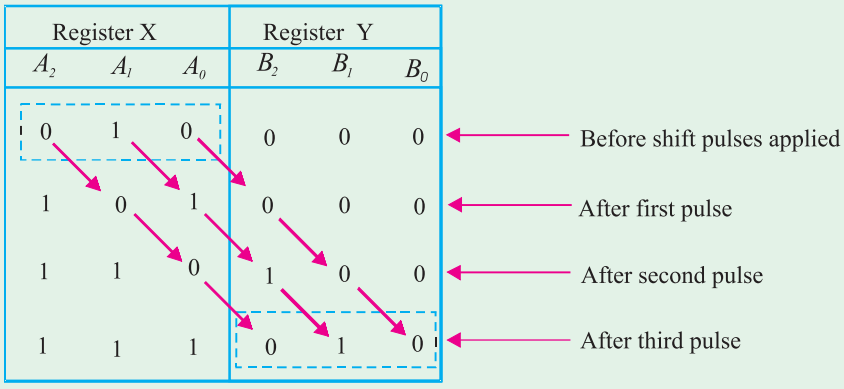


Fig. 72.72

As the shift pulses are applied, the data transfer takes place as follows:

$$A_2 \rightarrow A_1 \rightarrow A_0 \rightarrow B_2 \rightarrow B_1 \rightarrow B_0$$

The flip-flop output A_2 will go to a state determined by its D input. For discussion purpose let us suppose that D is held HIGH, so that A_2 will go HIGH on the first pulse and will remain there. Further let us assume that before any shift pulses are applied, the contents of A register are 010. (*i.e.* $A_2 = 0$, $A_1 = 1$, $A_0 = 0$) and the B register is at 000. As the shift pulse are applied one after the another, the states of each flip-flop output change are indicated in table as shown in Fig. 72.72. Following are some of the points which may be carefully noted from this table.

1. On the falling edge of each clock pulse, each flip-flop takes on the value that was stored in the flip-flop on its left prior to the occurrence of the pulse.
2. After first pulse, A_2 is set to 1, the 0 that was initially in A_2 is in A_1, the 1 that was initially in A_1 is in A_0, the 0 that was initially in A_0 is in B_2, the 0 that was initially in B_2 is in B_1, the 0 that was initially in B_1 is in B_0. The 0 that was initially in Y_0 is lost.
3. After second pulse, A_2 is still at 1 (because D is HIGH), the 1 that was previously in A_2 is in A_1, the 0 that was previously in A_1 is in A_0, the 1 that was previously in A_0 is in B_2, the 0 that was previously in B_2 is in B_1 and 0 that was previously in B_1 is in B_0.
4. After third pulse, A_2 is still at 1 (because D is held HIGH), the 1 that was previously in A_2 is in A_1, the 1 that was previous in A_1 is in A_0, the 0 that was previously in A_0 is in B_2, the 1 that was previously in B_2 is in B_1, the 0 that was previously in B_1 is in B_0.

It is evident from the above discussion that after these shift pulses, the 010 stored in the A register has now been shifted to B register. The register A is now at 111. Notice that it has lost its original data.

Notes.

1. The flip-flops shown in Fig. 72.71 can also be connected easily so that data shifts from right to left. As a matter of fact, there is no general advantage of shifting in one direction over the other. The direction chosen by the digital system designer depends upon the nature of application.
2. In serial data transfer (refer to Fig. 72.71) of N bits of data requires N clock pulses. This means three bits of data required three pulses, four bits requires four pulses and so on. However in parallel transfer, all the data is transferred simultaneously upon the occurrence of a single transfer pulse (refer to Fig. 72.66). In other words it does not matter how many bits are being transferred. It is obvious that parallel transfer is much faster than serial transfer using shift registers. However, parallel transfer requires more interconnections between the sending register (A) and receiving register (B) than does the serial transfer. This is an important consideration when the sending and receiving registers are at a distance from each other. The choice of either parallel or serial transmissions depends on the particular system application and specifications. In actual practice, a combination of two types is used to take advantage of the speed of parallel transfer and economy and simplicity of serial transfer.

72.30. Frequency Division

Many applications require frequency division. For example a quartz (or digital) watch makes use of a quartz crystal to generate a very stable oscillator frequency. This frequency is usually 1MHz or more. In order to advance the "seconds" display once every second, the oscillator frequency is divided by a value that will produce a very stable and accurate 1Hz output frequency. The frequency division can be achieved by using flip-flops.

Fig. 72.73 (*a*) shows a single *J-K* flip-flop connected to toggle (*i.e.* $J = K = 1$). When the clock pulses are applied at its CLK input, the Q output is a square wave with one-half the frequency of the clock input as shown in Fig. 72.73 (*b*). Thus a single flip-flop can be used as a divide-by-2 device. As

seen from the figure, the flip-flop changes state on each falling edge of the clock pulse. This results in an output that changes at half the frequency of the clock waveform.

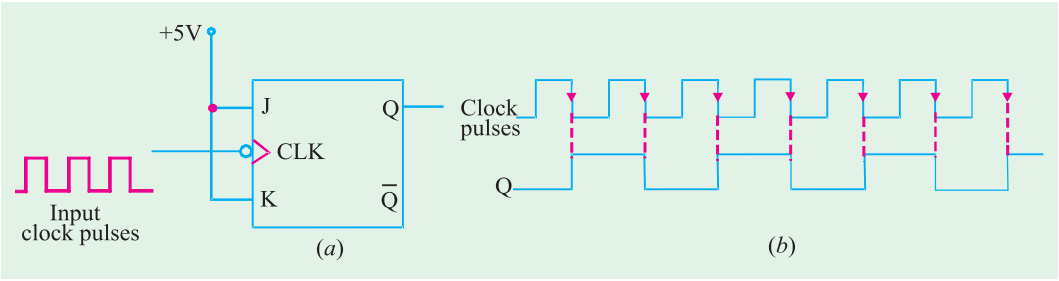


Fig. 72.73

Further division of the clock frequency can be achieved by using the output of one flip-flop as the clock input to the second flip-flop as shown in Fig. 72.74. Notice that the frequency of Q_0 output is divided by 2 by flip-flop 1. Therefore the Q_1 output is one-fourth the frequency of the original clock input. It may be carefully noted that propagation delays are not shown on the timing diagrams.

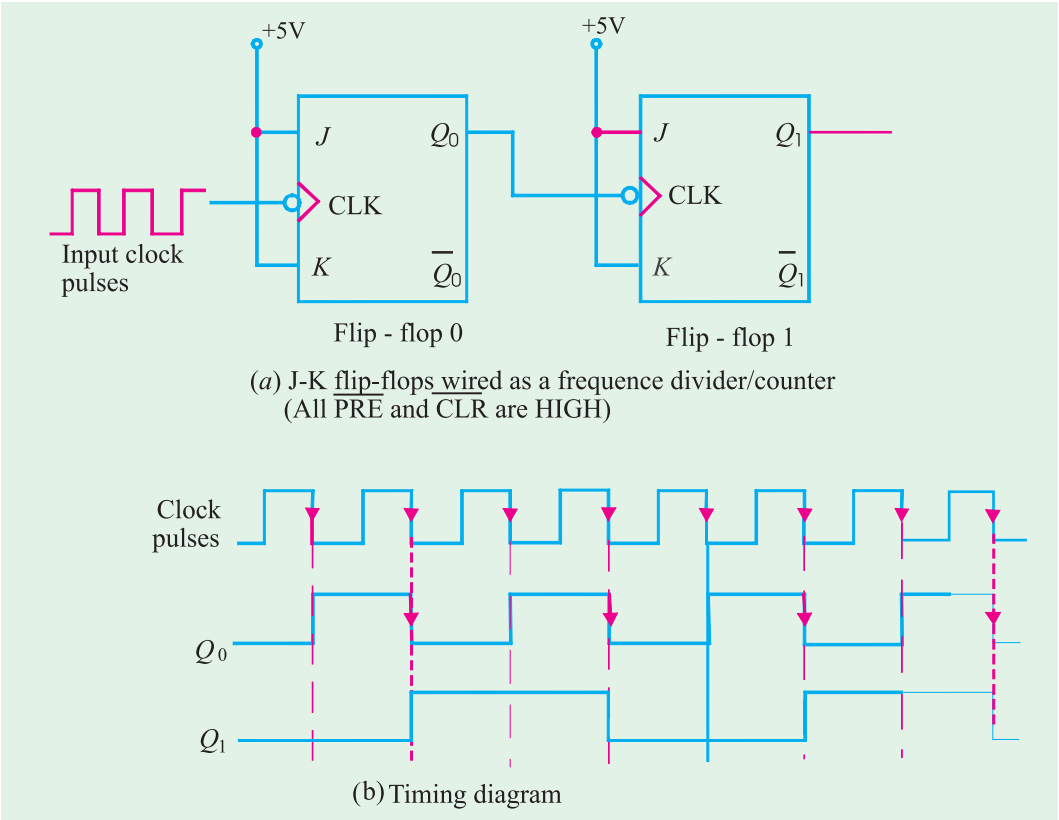


Fig. 72.74

By connecting the flip-flops in the way described above, a frequency division of 2^N is achieved, where N is the number of flip-flops. For example, three flip-flops divide the clock frequency by 2^3= 8, four flip-flops divide the clock frequency by 2^4=16 and so on.

Example 72.17. *How many flip-flops are required to divide a frequency by thirty-two.*

Solution. We know that a single flip-flop can divide the input frequency by 2. Two flip-flops

connected together can divide the frequency by 4. In gereal a frequency division of 2^N can be achieved by using N number of flip-flops. Thus,

$$2^N = 32 \Rightarrow N = 5 \quad \textbf{Ans.}$$

Example 72.18. *Fig. 72.74 shows three J-K flip-flops wired as frequency divider.*

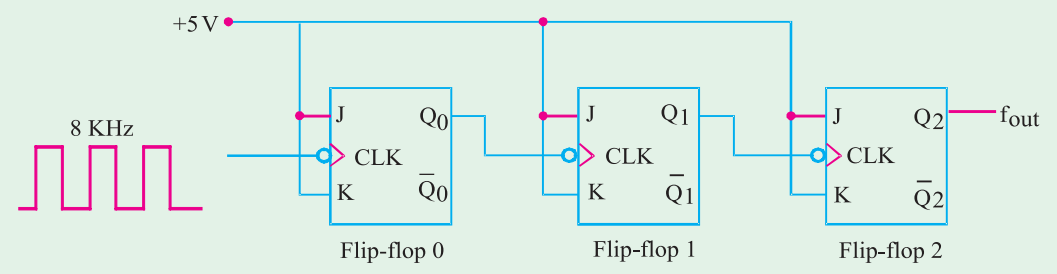


Fig. 72.75

Sketch the frequency waveform at Q_2 output when an 8 kHz square wave input is applied at CLK input of the flip-flop 'D'.

Solution. Note that each flip-flop is connected to toggle (*i.e.* $J = K = 1$). Since these are falling edge triggered flip-flops, the outputs change on the falling edge of clock pulse as shown in Fig. 72.76. There is one output pulse at Q_2 for every eight input pulses, so the output frequency is $f_{out} = 8\,\text{kHz} / 8 = 1\,\text{kHz}$. Fig 72.76 shows the waveforms at Q_0 and Q_1 outputs also.

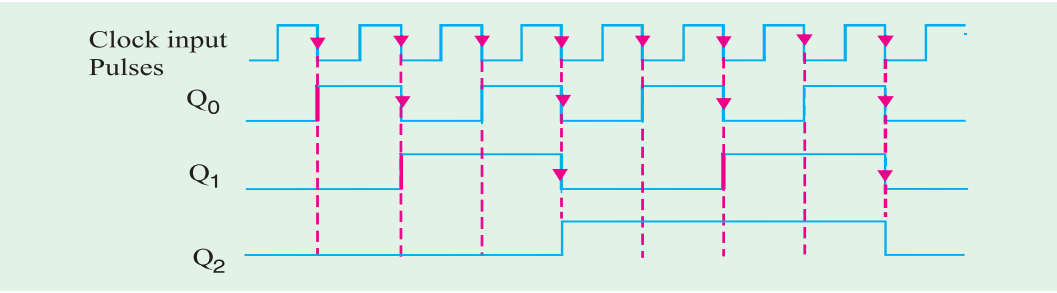


Fig. 72.76

72.31. Counting

Another important application of flip-flops is in digital counters. These are covered in more detail in chapter 7 on counters and Registers Fig 72.77 illustrates the concept of a 2-bit counter. Here two *J-K* flip-flops are wired to toggle (*i.e.* $J = K = 1$). Both the flip-flops are initially RESET. Flip-flop 0 toggles on the falling edge of each clock pulse. The *Q*-output of flip-flop 0 clocks flip-flop 1, so that each time Q_0 makes a HIGH-to-LOW transition, flip-flop 1 toggles. The resulting Q_0 and Q_1 waveforms are shown in the figure.

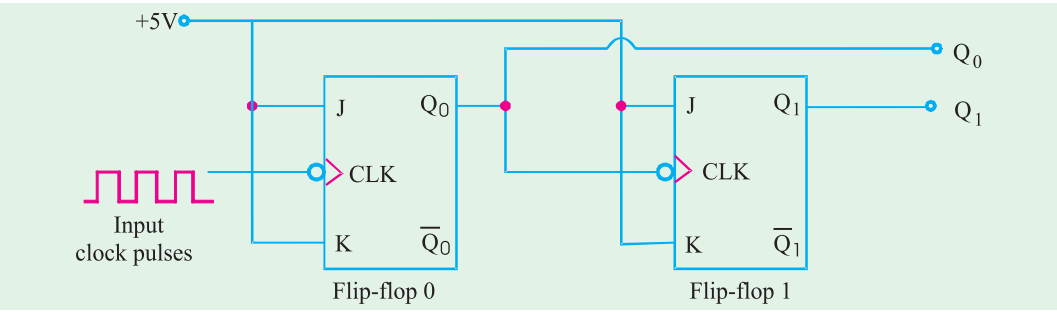


Fig. 72.77. J-K flip-flops wired as a 2-bit counter

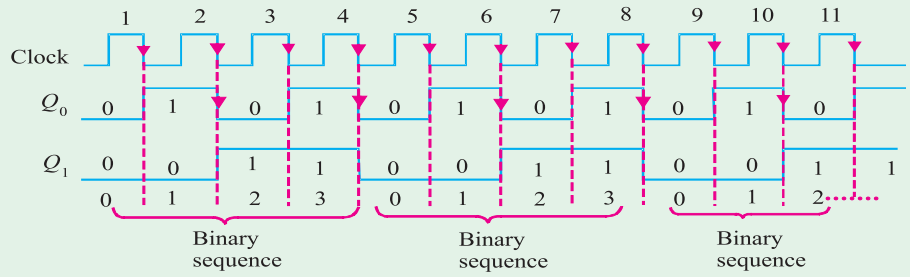


Fig. 72.78

Examine the sequence of Q_0 and Q_1 in Fig. 72.77. Prior to clock pulse 1, $Q_0 = 0$ and $Q_1 = 0$, after clock pulse 1, $Q_0 = 1$ and $Q_1 = 0$, after clock pulse 2, $Q_0 = 0$ and $Q_1 = 0$, and after clock pulse 3, $Q_0 = 1$ and $Q_1 = 1$. If we take Q_0 as the least significant bit, a two bit sequence is produced as the flip-flops are clocked. This binary sequence repeats every four clock pulses as shown in the timing diagram of Fig. 72.78. Thus flip-flops are counting in sequence from 0 to 3 (*i.e.* 00, 01, 10 and 11) and then recycling back to 0 to begin the counting sequence again.

Example 72.19. *Fig. 72.80 shows a J-K flip-flops wired as a 3-bit counter.*

Sketch the output waveforms at Q_0, Q_1 and Q_2. Also show the binary sequence represented by these waveforms.

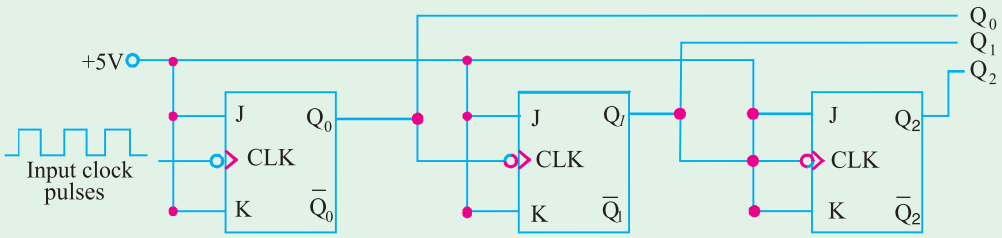


Fig. 72.79

Solution. Fig. 72.80 shows the output waveforms at Q_0, Q_1 and Q_2. Notice that the outputs change on the falling edge of the clock pulses. The outputs go through the binary sequence 000, 001, 010, 011, 100, 101, 110 and 111 as indicated.

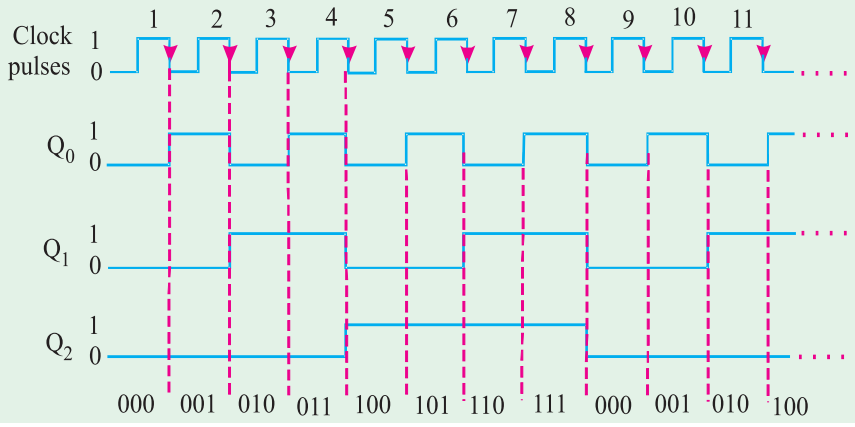
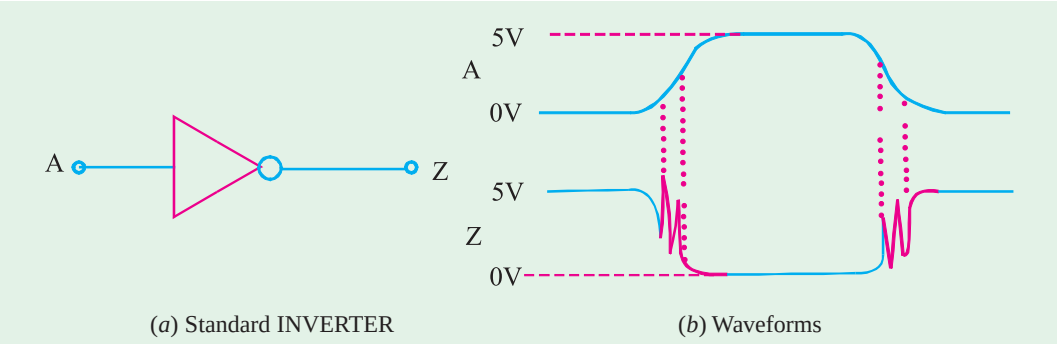


Fig. 72.80

72.32. Schmitt-Trigger Devices

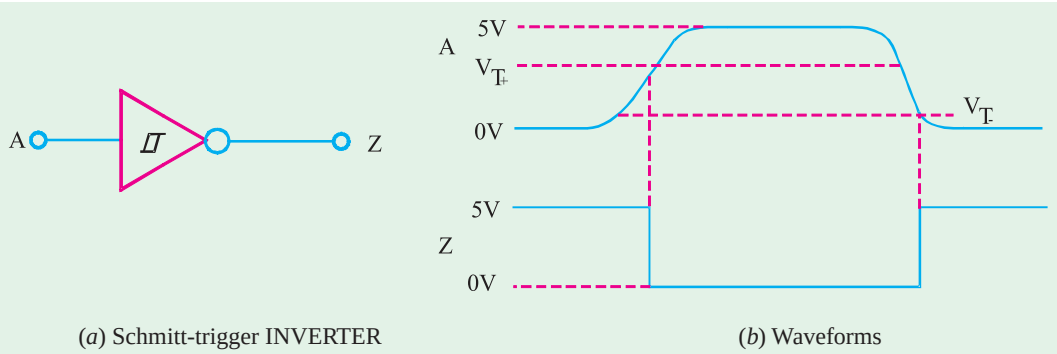
Strictly speaking, a Schmitt –trigger circuit is not classified as a flip-flop but it does exhibit a type of memory characteristic that makes it useful in certain special situations. One of those situations is shown in Fig. 72.81 (*a*). Here a standard INVERTER is being driven by a logic input that has relatively slow transition times. When these transition times exceed the maximum allowed values, the outputs of logic gates and INVERTERS may produce oscillations as the input signal posses through the indeterminate range (refer to Fig. 72.81 (*b*)). The same input conditions can also produce erratic triggering of flip-flops.



(*a*) Standard INVERTER (*b*) Waveforms

Fig. 72.81

A device that has a Schmitt-trigger type of input is designed to accept slow changing signals and produce an output that has oscillation free transitions. The output will generally have a very rapid transition times (typically 10ns) that are independent of the input signal characteristics. Fig. 72.82 (*a*) shows the symbol for a Schmitt-trigger INVERTER and (*b*) shows its response to a slow-changing input.



(*a*) Schmitt-trigger INVERTER (*b*) Waveforms

Fig. 72.82

As seen from Fig. 72.82 (*b*), we find that the output does not change from HIGH to LOW until the input exceeds the rising-edge threshold voltage, V_{T+}. Once the output goes LOW, it will remain there even when the input drops back below V_{T+} (this is its memory characteristic) until it drops all the way down below the falling edge threshold voltage, V_{T-}. The values of the two threshold voltages will vary from one logic family to another. But V_{T-} will always be less than V_{T+}.

It is evident from the above discussion that schmitt trigger is a special device that is used to transform slowly changing waveforms into sharply defined, jitter-free output singals. They are useful for changing clock edges that may have slow rise and fall times into straight vertical edges.

There are several ICs with Schmitt-trigger inputs. The 7414, 74LS14 and 74HC14 are hex INVERTER ICs with Schmitt-trigger inputs. On the other hand 7413, 74LS13 and 74HC13 are dual four-input NANDs with Schmitt-trigger inputs.

72.33. One-Shot (Monostable Multivibrator)

A one-shot (abbreviated as *OS*) is also referred to as monostable multivibrator. It is a digital circuit that is somewhat related to a flip flop. For instance, like a flip-flop, the one-shot has two inputs, Q and $\overline{Q}$ which are inverse (or complement) of each other. However, unlike the flip-flop, the one-shot has only one stable output state (normally, $Q = 0$ and $\overline{Q} = 1$). The one-shot remains in this stable state until it is triggered by an input signal. Once triggered, the one-shot outputs switch to the opposite state ($Q = 1, \overline{Q} = 0$). This state is called quasi-stable (i.e. unstable) state. The one-shot remains in this quasi-stable state for a fixed period of time, t_p, which is determined by the RC time constant. The RC time constant in turn is determined by the values of external components connected to one-shot. After a time, t_p, the one-shot outputs return to their resting state until triggered again.

Fig. 72.83 (*a*) shows the logic symbol for one-shot. As seen from this figure, it has one input labeled as trigger input (T), a normal output (Q), and an inverted output ($\overline{Q}$). It needs two external components: resistor (R_T) and a capacitor (C_T) for its operation. The values of R_T and C_T determines the exact value of time period (t_p) for which the one-shot switches to its quasi stable state. The value of tp can vary from several nanoseconds to several tens of seconds. Fig. 72.83 (*b*) shows the stable state (where $Q = 0$, $\overline{Q} = 1$) and the quasi-stable state (where $Q = 1$ and $\overline{Q} = 0$).

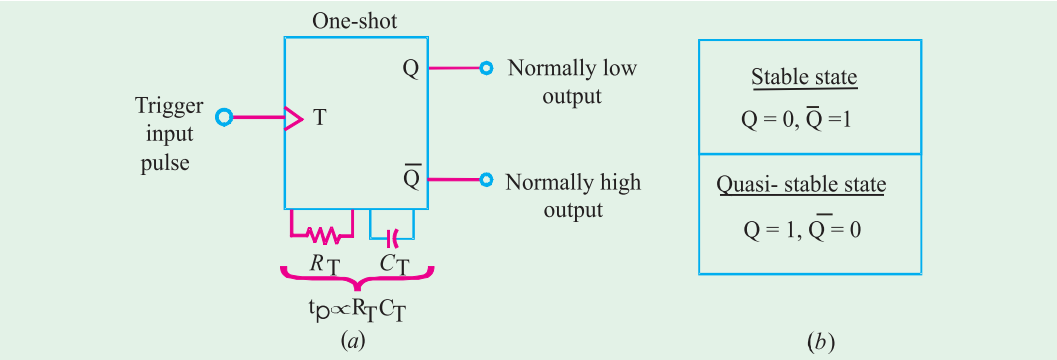


Fig. 72.83. One-shot symbol and Output states.

72.34. Types of One-Shots

Depending upon the operation, the one-shot is of the following two types :

1. Non-retriggerable one-shot and
2. Retriggerable one-shot

Both the type of one-shots are available in the IC form. Now we shall study both these one-shots one by one in the following pages.

72.35. Non-Retriggerable One-Shot

We have already discussed in the last article that one-shot (or a monostable multivibrator) is a device with only one stable state. It is normally in its stable state and will change to its quasi-stable state only when triggered. Once it is triggered, the one-shot remains in its quasi-stable state for a fixed

period of time and then automatically returns to its stable state. The time that the device remains in the quasi-stable state determines the pulse width (t_p) of the output.

Fig. 72.84 (*a*) shows the waveforms of a trigger input signal *T* and output of one-shot. Notice that the rising edge of the trigger signal will trigger the one-shot to its quasi-stable state. The one-shot will remain in the quasi-stable state for a time, t_p after which it automatically returns to its stable state.

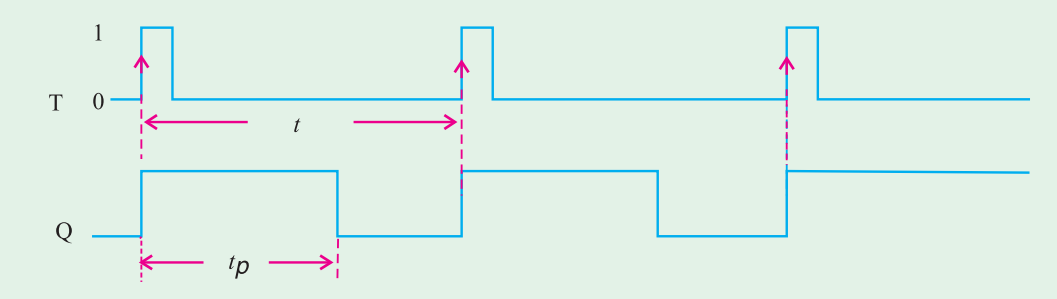


Fig. 72.84

It may be noted from the diagram shown above that one-shot is being triggered at intervals greater than its pulse width (*i.e.* $t > t_p$).

Now we will consider a situation where the one-shot is being triggered at intervals less than its pulse width (*i.e.* $t < t_p$). Refer to Fig. 72.85. Notice that the rising edges of the T signal at points '*b*' has no effect on the one-shot because it has already been trigged to the quasi-stable state by the rising edge at point '*a*'. The one-shot must return to the stable state before it can be triggered. The same argument is valid for the rising edge at point '*d*'. It has no effect on the one –shot because it has already been trigged by the rising edge at point '*c*'.

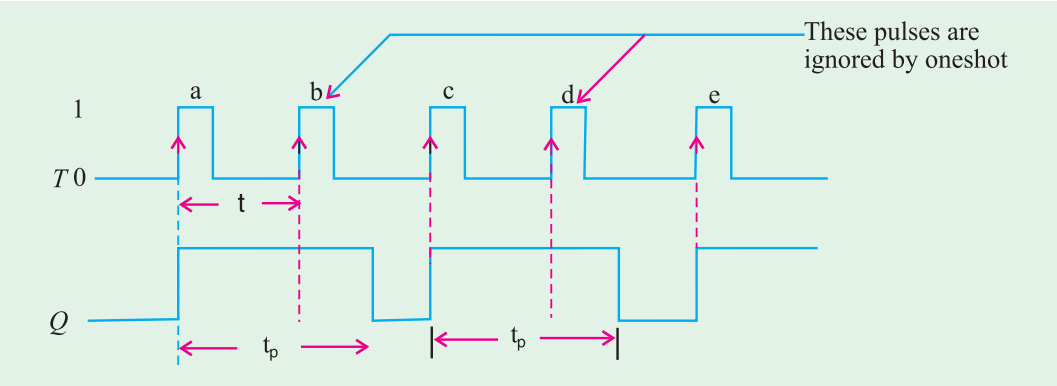


Fig. 72.85

It may be carefully noted that one-shot output pulse duration (t_p) is always the same regardless of the input pulses. The pulse duration t_p, depends only on the values of external components R_T and C_T and the internal one-shot circuitry. In a typical monoshot, the value of t_p is given by the relation.

$$t_p = 0.7\, R_T C_T$$

Example 72.20. *Fig.* 72.84 *shows three non-retriggerable one-shots connected in a timing chain that produces three sequential output pulses. Note the* "1" *in front of the pulse on each one-shot symbol to indicate non-retriggerable operation.*

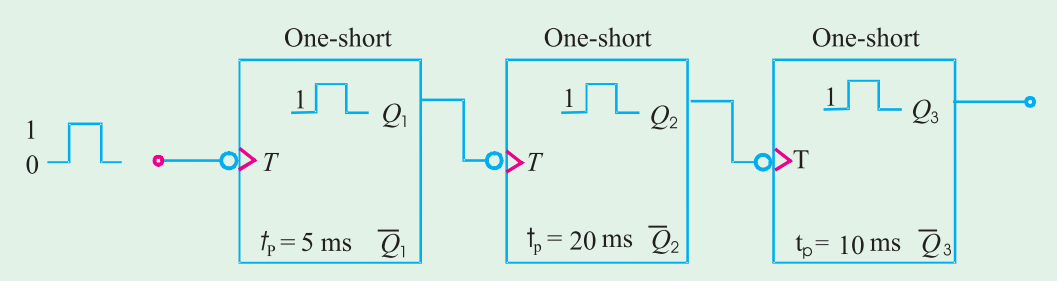


Fig. 72.86

Draw a timing diagram showing the relationship between the input pulse and the three one-shot outputs. Assume an input pulse duration of 10 *ms.*

Solution. Notice that all the one-shots trigger at the falling edge of the input signal. To begin with, the first one-shot triggers at the falling edge of the input signal and the second one-shot triggers at the falling edge of Q_1. The third one-shot triggers at the falling edge of Q_2.

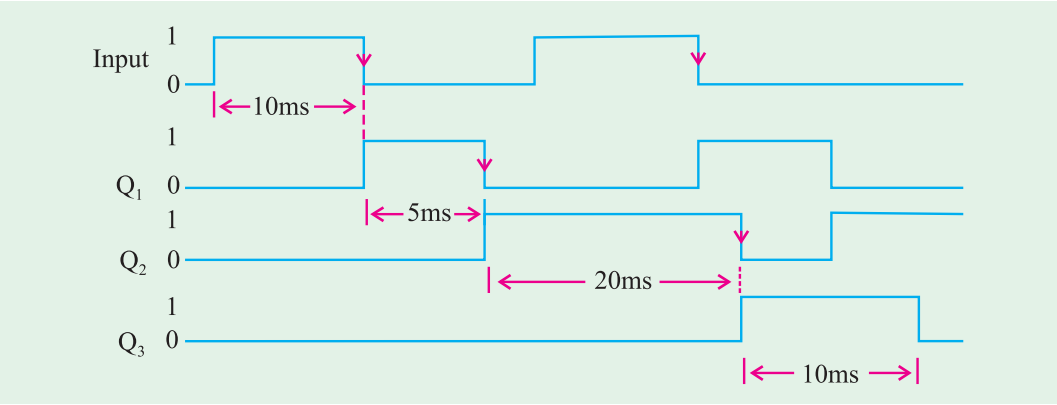


Fig. 72.87

Fig. 72.87 shows a sketch of the timing waveforms at Q_1, Q_2 and Q_3 along with the input trigger waveform.

Example 72.21. *Sketch the waveforms at Q and* $\overline{Q}$ *outputs of a one-shot which is being triggered by a signal shown in Fig.* 72.88. *Assume that one-shot triggers at the rising edge of the trigger signal and has a* $t_p = 1.5$ *ms.*

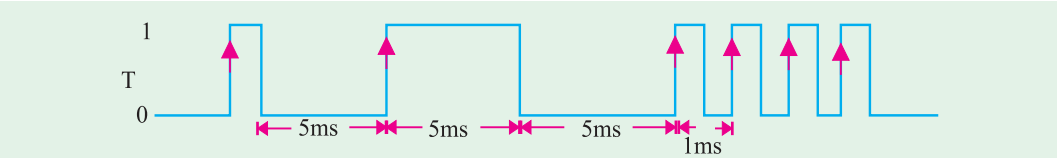


Fig. 72.88

Solution. Fig. 72.89 shows the waveforms at the one-shot outputs Q and $\overline{Q}$ in response to the trigger signal (T). Notice that the rising edges at points *a*, *b*, *c* and *e* of the *T* signal, will trigger one-shot to its quasi-stable state. The one-shot will remain in the quasi-stable state for a time, t_p after which it automatically returns to its stable state.

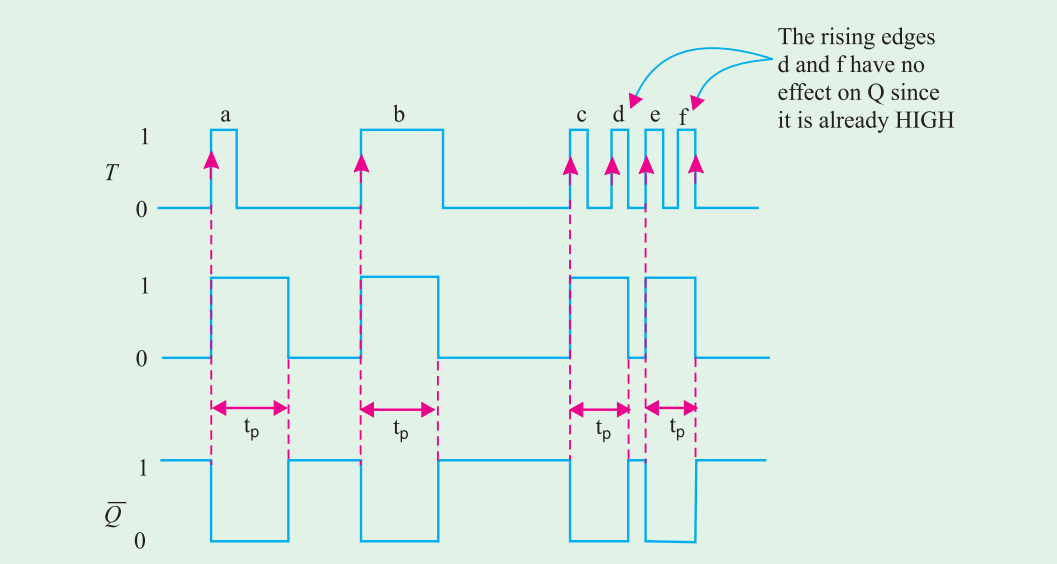


Fig. 72.89

The rising edges of the *T* signal at points '*d*' and '*f*' have no effect on the one-shot because it has already been triggered to the quasi-stable state. The one-shot must return to the stable state before it can be triggered.

72.36. Retriggerable One-Shot

This one-shot usually operates in the same manner as the non-retriggerable one-shot except for one-major difference. The major difference is that retriggerable one-shot can be retriggered while it is in the quasi-stable state, and it will begin a new t_p interval. Fig. 72.90 compares the response of both types of one-shot using a t_p of 2 ms interval.

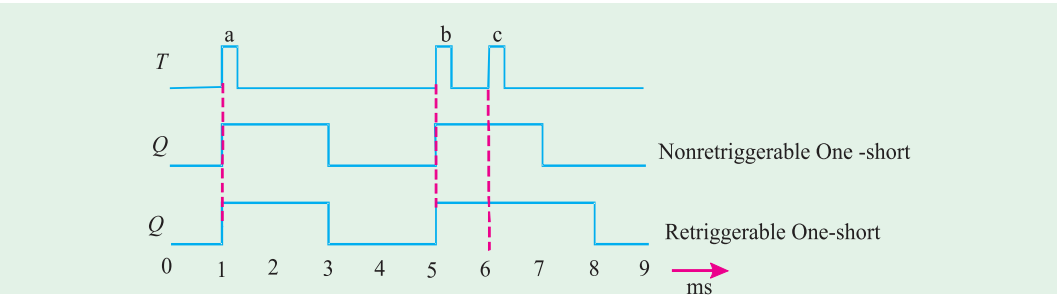


Fig. 72.90

As seen from the diagram, both types of one-shot respond to the first trigger pulse indicated as '*a*' at t = 1 ms by going HIGH for 2 ms and then returning LOW. The second trigger pulse indicated as '*b*' at t = 5 ms triggers both one – shots to the HIGH state. The third trigger pulse indicated as '*c*', at t = 6 ms has no effect on the non-retriggerable one-shot because it is already in the quasi-stable state. However, this trigger pulse will retrigger the retriggerable one-shot to begin a new t_p = 2 ms internal. Thus it will stay HIGH for 2 ms after this third trigger pulse.

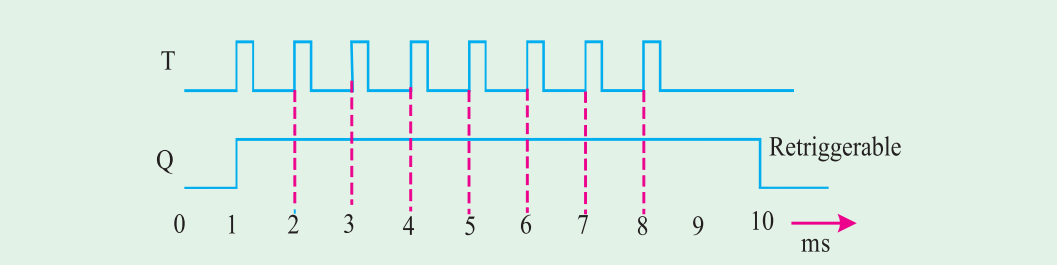


Fig. 72.91

It is evident from the discussion above that a retriggerable one-shot begins a new t_p interval each time a trigger pulse is applied, regardless of the current state of its *Q*-output. As a matter of fact, trigger pulses can be applied at a rate fast enough that one-shot will always be retriggered before the end of the t_p interval and *Q*-output will remain HIGH. Such a situation is shown in Fig. 72.91. As seen from this diagram, eight pulses are applied every 1 ms. The *Q*-output does not return LOW until 2 ms after the last trigger pulse.

Example 72.22. *A retriggerable one-shot can be used as a pulse-frequency detector that detects when the frequency of a pulse input is below a predetermined value. A simple example of this application is shown in Fig.* 72.92. *The operation begins by momentarily closing switch SW*1.

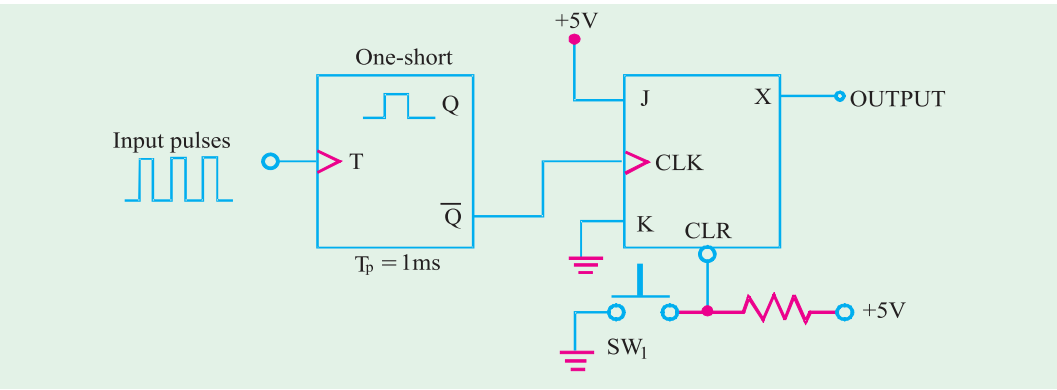


Fig. 72.92

(*a*) *Describe how the circuit responds to input frequencies above* 1*kHz.* (*b*) *Describe how the circuit responds to input frequencies below* 1*kHz.* (*c*) *How would you modify the circuit to detect when the input frequency drops below* 50 kHz?

Solution.

(*a*) Circuit response to input frequencies above 1kHz. Closing the switch SW1, clears *X* to Zero. Since the one-shot has t_p = 1 ms, the one-shot will be retriggered before the end of the t_p interval for frequencies greater than 1 kHz. Thus $\overline{Q}$ will stay LOW. As a result of this, there is no clock input to *JK* flip-flop and hence *X* will remain LOW.

(*b*) Circuit response to input frequencies below 1kHz. If the input frequency falls below 1 kHz, the $\overline{Q}$ will return HIGH before the one-shot is triggered again. This positive-going-trigger will clock *JK* flip-flop and *X* will change to HIGH.

(*c*) Modification of the circuit to detect frequencies below 50 kHz. In order to detect frequencies below 50 kHz, we will have to change t_p of the one-shot to 1/50 kHz = 20 ms.

72.37. Actual One-Shot Devices

There are several one-shot ICs that are available commercially in both the non-retriggerable and retriggerable versions. Table 72.3 shows some of the most popular one-short ICs available commercially. As seen from the table 72.3 we find that 74121 and 74221 ICs are non-retriggerable one-shots. Where as 74122 and 74123 ICs retriggerable one-shots.

Table 72.3. Actual one-shot devices

Device	*Description*
74121	A *single non-retriggerable* one-shot IC in standard TTL series.
74221	A *dual non-retriggerable* one-shot IC in standard TTL series.
74LS221	A *dual non-retriggerable* one-shot IC in Low-power SchottkyTTL series
74HC221	A *dual non-retriggerable* one-shot IC in high-speed CMOS series
74122	A *single retriggerable* one-shot IC in standard TTL series
74LS122	A single retriggerable one-shot in IC low-power Schottky TTL series.
74123	A dual retriggerable one-shot IC in standard TTL series
74LS123	A dual retriggerable one-shot IC in lower-power Schottky TTL series
74HC123	A dual retriggerable one-shot IC in high-speed CMOS series.

Fig. 72.93 shows the logic symbol for the 74121 non-retriggerable one-shot IC. As seen from the diagram, the 74121 IC contains internal logic gates to allow inputs $\overline{A}_1$, $\overline{A}_2$, and B to trigger the device in a variety of ways. The B input is a Schmitt-Trigger type of input that is allowed to have slow transition times and still reliably trigger the one-shot. The pins labelled R_{EXT}, R_{EXT} /C_{EXT} are used to connect an external resistor and capacitor to achieve the desired output pulse duration.

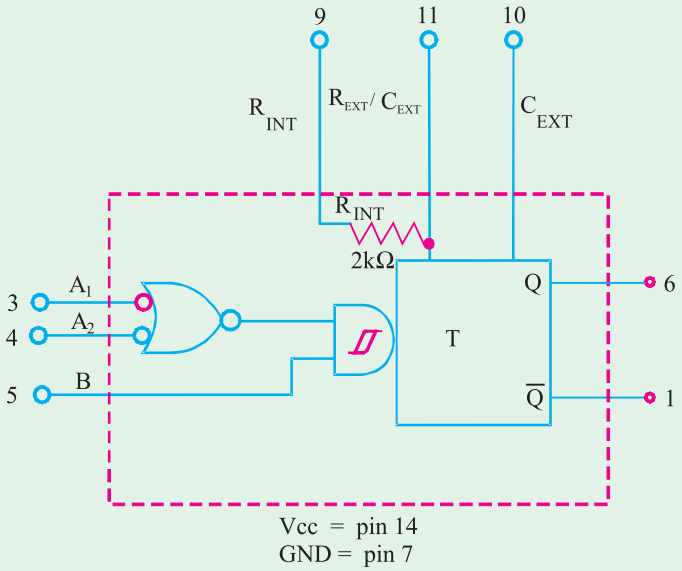


Fig. 72.93. Logic symbol of 74121.

A typical pulse width of 30 ns is produced when no external components are used and the internal timing registor (2 k Ω) is connected to V_{CC}. The pulse width can be set anywhere between 30 ns and 28 ns by the use of external components, (*i.e.* R_{EXT} and C_{EXT}). Mathematically the pulse width.

$$t_p = 0.7\, R\, C_{EXT} \quad \ldots(i)$$

where R is either R_{INT} or R_{EXT}. Notice that if R is in kilohms (kΩ) and C_{EXT} is in picofarads (PF), the output pulse width (t_p) is in nanoseconds. Further if R is in kilohms (kΩ) and C_{EXT} is in microfarads (μF), the output pulse width (t_p) is in microseconds (μS).

Table 72.4 shows the truth table of 74121. Notice that for the first four rows of the table, the Q-output is LOW.

Table 72.4

Inputs			Output	
A_1	$\overline{A}_2$	B	Q	$\overline{Q}$
L	X	H	L	L
X	L	H	Ll	Ll
X	X	L	L	L
H	H	X	L	L
H	↓	H	⎍	⎎
↓	H	H	⎍	⎎
↓	↓	H	⎍	⎎
L	X	↑	⎍	⎎
X	L	↑	⎍	⎎

H = High voltage level
L = Low voltage level
X = Don't case
↑ = Rising edge
↓ = Falling edge

It is because of the fact that no trigger pulse is applie at any one of the three inputs. In the next five lines of the truth table, notice that we have a trigger pulse at one of the inputs $(A_1\ A_2$ or $B)$ and the other two inputs are connected LOW or HIGH.

Fig. 72.94 (a) shows the logic symbol and 72.94 (b) the truth table for 74123 a dual retriggerable one = shot. As seen from the diagram, the 74123 IC contains two retrigger able one-shots. Each retriggerable one-shot contains $\overline{A}_1$, B_1 and R_D to trigger the device in a variety of ways.

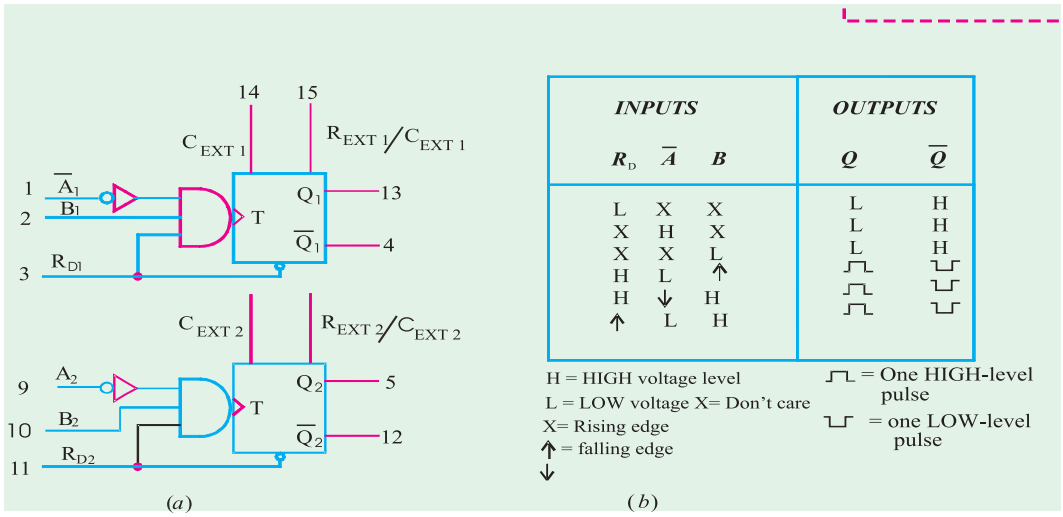


Fig. 72.94

A minimum pulse width of approximately 45 ns is obtained with no external components. Wider pulse widths are achieved by using external components. A general formula for calculating the values of these components for a specified pulse width (t_p), provided $C_{EXT} > 1000$ pF,

$$t_p = 0.28\, \mathrm{R}\, C_{\mathrm{EXT}} \left(1 + \frac{0.7}{\mathrm{R}}\right) \quad \ldots. (ii)$$

where 0.28 is a constant determined by a particular type of one-shot, R is either the internal or external resistor, C_{EXT} is in pF and t_p is in ns.

Notes.

1. While selecting the value of R_{EXT}, care should be taken that its value must be in kilohms ($k\Omega$) so that the circuit current is in milli amperes (mA). If the resistor value is chosen in ohms (Ω), the circuit current will be in amperes which the IC will not be able to handle it. On the other hand, if the value of R_{EXT} is in megohms ($M\Omega$), the current will be in micro amperes (μA) which may be too small for internal operation of 74121. Moreover, resistances with values in the range of megohms are susceptible to electrostatic noise.
2. For a desired pulse width t_w, we can determine the values of external components in two ways.
   - ***(i)*** Arbitrarily select the value of capacitor, C_{EXT} and then using equation $t_W = 0.7\ R_{EXT}\ C_{EXT}$, determine the value of $R_{EXT.}$ be in microfarads (μF).
   - ***(ii)*** Arbitrarily select the value of R_{EXT} and then using the equation, $t_W = 0.7\ R_{EXT}\ C_{EXT}$, determine the value of C_{EXT}.
   - ***(iii)*** If t_w is required to be in nanoseconds, choose R_{EXT} in kilohms ($k\Omega$) so that C_{EXT} is in picofarads. However if the desired t_w is required to be in milliseconds, still select R_{EXT} is in kilohms ($k\Omega$), but C_{EXT} will now be in microfareds (μF). Care should be taken while selecting the component values in the sense that the components are availably only in certain standard values.
3. While selecting the value of C_{EXT}, care should be taken that it is much larger than any stray capacitance that might be encountered in a typical electronic circuit. Values of capacitance less than 100 pF (0.0001 μF) may be unsuitable because it is not uncommon for there to be 50 pF of stray capacitance between traces in a printed-circuit board.

Example 72.23. *A certain digital circuit requires a one-shot with a pulse width of 10* ms. *Determine the values of external components to be connected with an* IC *74121 to produce the desired pulse width. Also show the connections of the external components to 74121.*

Solution.

Let R_{EXT} = The value of external resistor and

C_{EXT} = The value of external capacitor.

Then we know that the pulse width for a 74121 IC (t_p),

$$10 \times 10^{-3} = 0.7\ R_{EXT}\ C_{EXT} \qquad \text{... (i)}$$

Since the desired pulse width is in milli seconds, let us arbitrarily select $C_{EXT} = 1$ μF. Then from equations (*i*).

$$10 \times 10^{-3} = 0.7 \times R_{EXT} \times (1 \times 10^{-6})$$

$$\therefore \quad R_{EXT} = \frac{10 \times 10^{-6}}{0.7\,(1 \times 10^{-6})} = 14.28 \times 10^{-3}\ \Omega = 14.28\ k\Omega$$

In order to select a resistor value of 14.28 $k\Omega$, we can use a 10 $k\Omega$ fixed resistor with a 5 $k\Omega$ potentiometer.

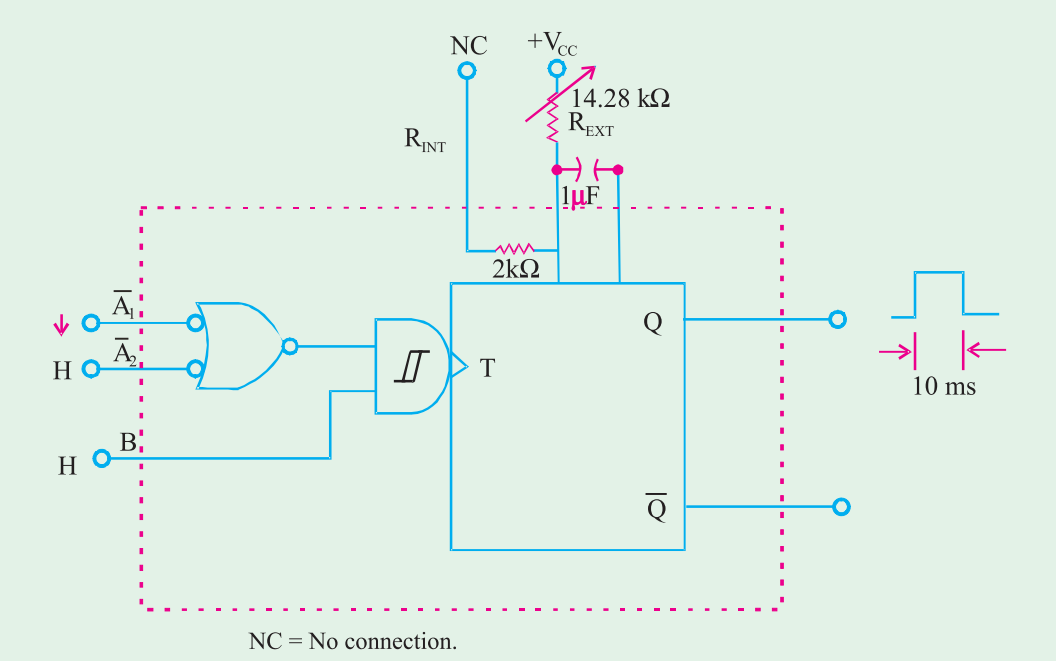


Fig. 72.95

Fig. 72.95 shows a 14.28 k Ω resistor with 1 μF capacitor connected to the 74121 IC for producing a desired pulse width of 10 ms. Notice the settings on $\overline{A}_1$, $\overline{A}_2$ and *B* inputs of 74121 as well.

Example 72.24. *Design a circuit using a* 74121 *to convert a* 50 *kHz,* 80% *duty cycle square wave to a* 50 *kHz,* 50% *duty cycle square wave.*

Solution.

Fig. 72.96 shows the given square wave. Notice that it has a time period of 1/50 kHz (= 20 μs) and has a pulse width of 80% of 20 μS (= 16 μs).

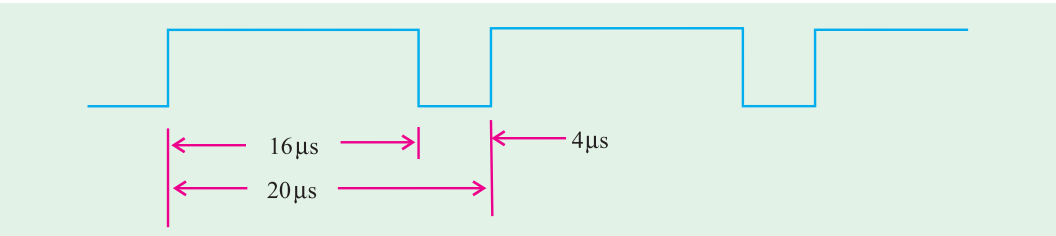


Fig. 72.96

Now we have to stretch the 4 μs negative pulse to 10 μs to make the duty cycle 50 %. If we use the falling edge on the negative pulse to trigger $\overline{A}_1$ input to a 74121 and set the output pulse width (t_p) to 10 μs, we should have the solution as shown in 72.97 (*b*). The output will be taken from $\overline{Q}$ because it provides a negative pulse when triggered.

Let R_{EXT} = The value of external resister and

C_{EXT} = The value of external capacitor,

Then the pulse width (t_p)

$$10 \times 10^{-6} = 0.7\, R_{EXT} C_{EXT}$$

Let us arbitrarily select C_{EXT} = 0.001 μ*F* (equal to 1000pf); then

$$10\ \mu s = 0.7\, R_{EXT}$$

$$R_{EXT} = \frac{10 \times 10^{-6}}{0.7 \times 0.001 \times 10^{-6}}$$

or

$$= 14.4\,k\Omega$$

In order to select a resistor of 14.4 k Ω, we can choose a fixed resistor of 10 k Ω with a 5 k Ω potentiometer.

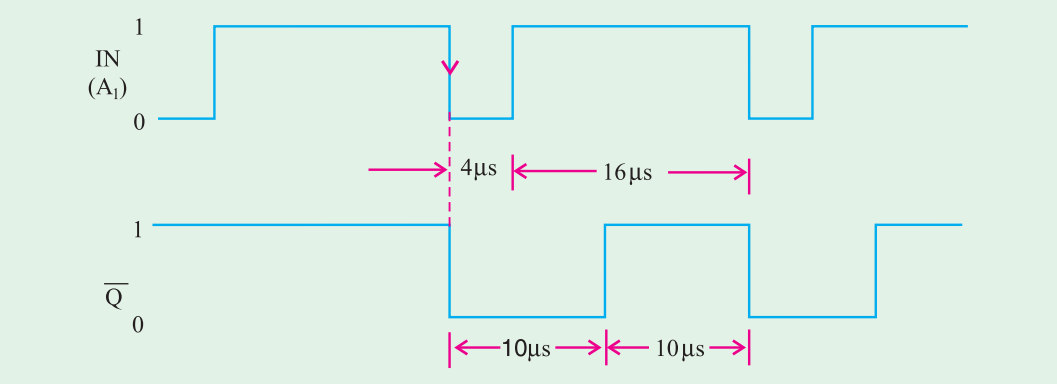


Fig. 72.97

Fig. 72.98 shows the circuit connections for producing the desired waveform. Notice that the input waveform is applied at $\overline{A}_1$ input while $\overline{A}_2$ and B are tied HIGH. The output is obtained from $\overline{Q}$.

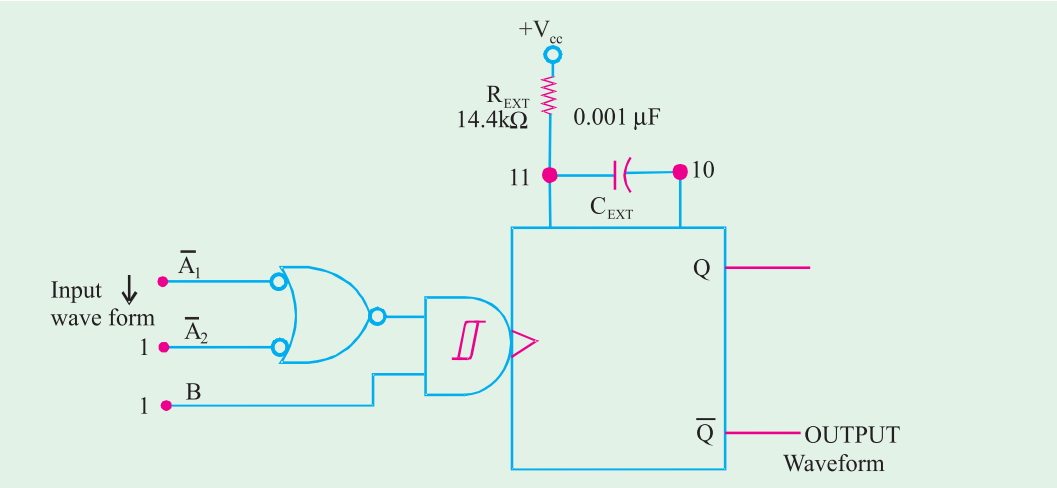


Fig. 72.98

Example 72.25. *Determine the value of R_{EXT} and C_{EXT} that will produce a pulse width of* 50 µS *when connected to a* 74123.

Solution.

Let R_{EXT} = The value of external resistor and

C_{EXT} = The value of external capacitor

Then we know that the pulse width for a 74123 IC (t_p),

$$50 \times 10^{-6} = 0.28\, R_{EXT}\, C_{EXT} \left(1 + \frac{0.7}{R_{EXT}}\right) \quad \text{... (i)}$$

Let us arbitrarily select C_{EXT} = 0.01 μF (recall that C_{EXT} has to be greater than 1000 pF), then, from equation (*i*),

$$50 \times 10^{-6} = 0.28\ R_{EXT}\ (0.01 \times 10^{-6}) \left(1 + \frac{0.7}{R_{EXT}}\right)$$

$$= 0.28\ (R_{EXT} + 0.7)\ (0.01 \times 10^{-6})$$

$$\therefore \qquad R_{EXT} + 0.7 = \frac{50 \times 10^{-6}}{0.28 \times (0.01 \times 10^{-6})} = 17857\ \Omega$$

or $$R_{EXT} = 17857 - 0.7 = 17856.3\ \Omega = 17.9\ \text{k}\ \Omega$$

Thus we can use a fixed resistor of 15 k and 5 k potentiometer to select $R_{EXT} = 17.9\ k\ \Omega$.

72.38. Clock Generator Circuits

We have already discussed in Art. 72.2 that flip-flop has two stable states due to which they are known as bistable multivibrators. Further in Art. 72.33 we have discussed that one-shot has one stable. state due to which they are referred to as monostable multivibrators. Now we shall discuss a third type of multivibrator which has no stable states. Such a multivibrator is called an astable or free-running multivibrator. An astable multivibrator switches back and forth (*i.e.* oscillates) between two unstable states without any external triggering.

There are several types of astable multivibrators that are in common use but the following three are important from the subject point of view;

1. Schmitt-Trigger oscillator
2. 555 Timer used as astable oscillator.
3. Crystal-controlled clock generator.

Now we shall discuss all the three oscillators one by one in the following pages.

72.39. Schmitt-Trigger Oscillator

Fig. 72.99 shows a Schmitt-trigger INVERTER connected as an oscillator. The signal at V_{OOT} is an approximate square wave with a frequency that depends upon the values of *R* and *C* values. The relationship between the frequency and RC values is shown in Fig. 72.100 for three different Schmitt-trigger INVERTERS.

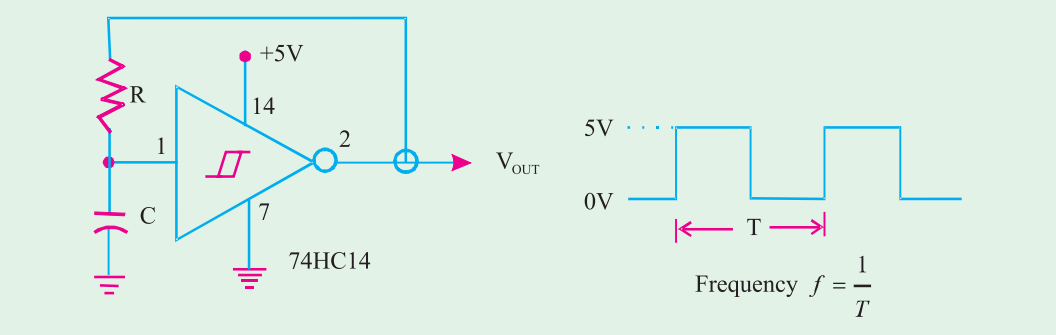


Fig. 72.99

As seen from this diagram, for 7414 Schmitt-trigger INVERTER, the frequency is given by, f = 0.8/RC. Notice that for this IC, the value of $R \leq 500\ \Omega$. For 74LS14 IC, the frequency of the output square wave is again 0.8/RC but the value of $R \leq 2\ k\Omega$. On the other hand for 74HC14, the frequency of the output square wave is 1.2/RC, where the value of *R* must be $10\ \text{M}\Omega$.

IC	Frequency	Remarks
7414	≈ 0.8/RC	(R ≤ 500 Ω)
74LS14	≈ 0.8/RC	(R ≤ 2 kΩ)
74HC14	≈ 1.2/RC	(R ≤ 10 MΩ)

Table 72.5

72.40. 555 Timer Used as an Astable Multivibrator

The 555 IC is a very popular, general purpose timer IC. It can be connected as a one-shot or as an astable multivibrator or as a free running oscillator. Fig. 72.100 shows how external components can be connected to a 555 so that it operates as a free-running oscillator. The output of the oscillator is a repetitive rectangular waveform as shown in Fig. 72.100 (*b*).

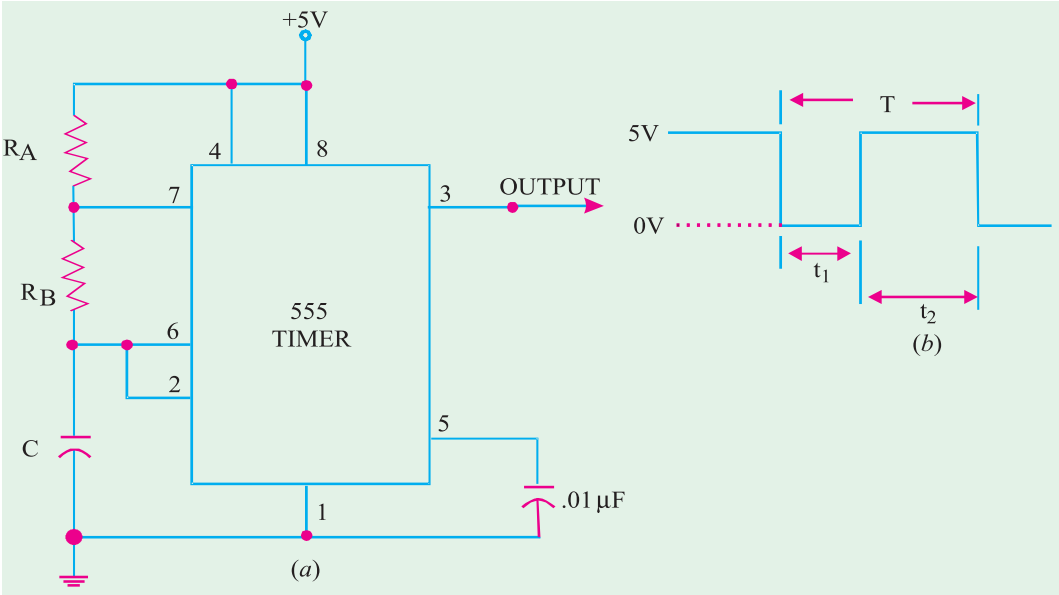


Fig. 72.100. 555 timer used as an astable multivibrator.

As seen from the diagram, the output switches between two logic levels with the time intervals ($t_1$and t_2) at each level. These two time intervals are determined by the *R* and *C* values.

The time interval t_1 is given by the equation,

$$t_1 = 0.7\ R_B C \qquad \text{... (i)}$$

and the time interval,

$$t_2 = 0.7\ (R_A + R_B)\ C \qquad \text{... (ii)}$$

∴ The time period, of the output waveform

$$T = t_1 + t_2$$

Substituting the values of t_1 and t_2 from equations (*i*) and (*ii*), the time period,

$$\text{or} \qquad T = 0.7\ R_B C + 0.7\ (R_A + R_B)\ C$$
$$= 0.7\ (R_A + 2\ R_B)\ C$$

The frequency of the output waveform,

$$f = \frac{1}{T} = \frac{1}{0.7\ (R_A + 2\ R_B)\ C}$$

$$= \frac{1.44}{(R_A + 2R_B)C}$$

Further, the duty cycle of the output waveform,

$$\text{duty cycle} = \frac{t_2}{T} \times 100\ \%$$

It may be noted from equations (*i*) & (*ii*) that t_1 and t_2 cannot be equal unless R_A is made zero. This cannot be done without producing excess current through the device. This means, it is impossible to produce an output waveform with a perfect 50% duty cycle. However, it is possible to get very close to 50% duty cycle (*i.e.* $t_1 \approx t_2$) by selecting,

$$R_B \gg R_A$$

It may be carefully noted that the value of R_A must be greater than 1k Ω and the value of C must be greater than 500 pF.

Example 72.26. *Fig.* 72.101 *shows the circuit of a* 555 *timer used as an astable multivibrator.*

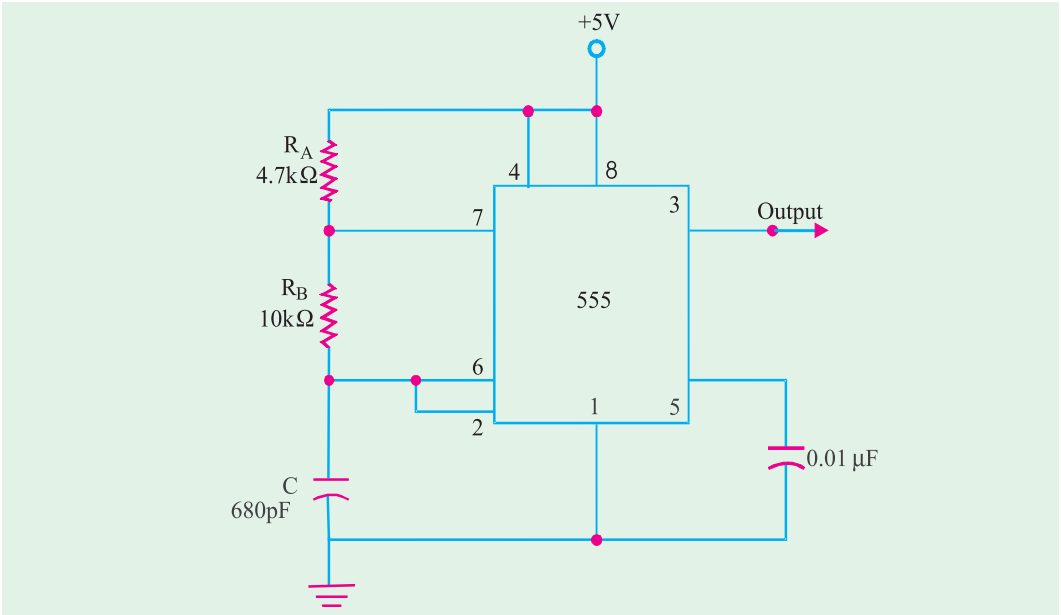


Fig. 72.101

Determine the values t_1, t_2, frequency and duty cycle of the output waveform.

Solution.

$$t_1 = 0.7\ R_B\ C \times 0.7 \times (10 \times 10^3) \times (680 \times 10^{-12})$$
$$= 4.76 \times 10^{-6}\ s = 4.76\ \mu s$$
$$t_2 = 0.7\ (R_A + R_B)\ C$$
$$= 0.7 \times ((4.7 \times 10^3) + (10 \times 10^3)) \times (680 \times 10^{-12})$$
$$= 6.997 \times 10^{-6}\ s \approx 6.997\ \mu s$$
$$T = t_1 + t_2 = 4.76 + 6.997 = 11.757\ \mu s$$
$$f = \frac{1}{T} = \frac{1}{11.757 \times 10^{-6}} = 8.5 \times 10^4 = 85\ \text{kHz}$$

$$\text{Duty cycle} = \frac{t_2}{T} \times 100\% = \frac{6.997}{11.757} = 100\% = 59.5\ \%$$

Tutorial Problems No. 72.1

1. The waveforms of Fig 72.100 are applied to the inputs of a NAND gate latch shown in Fig.72.102. Assume that initially $Q = 0$, and determine the waveform at Q-output.

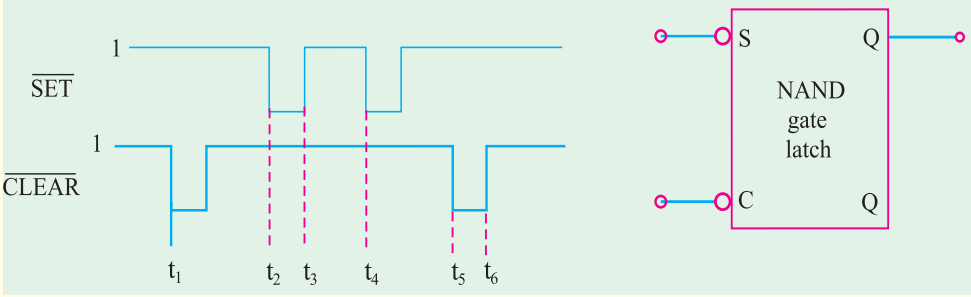


Fig. 72-102

2. The waveforms of Fig. 72.103 (*a*) are applied to the inputs of a NOR gate latch shown in Fig. 72.103 (*b*). Assume that initially $Q = 0$ and determine the waveform at Q output.

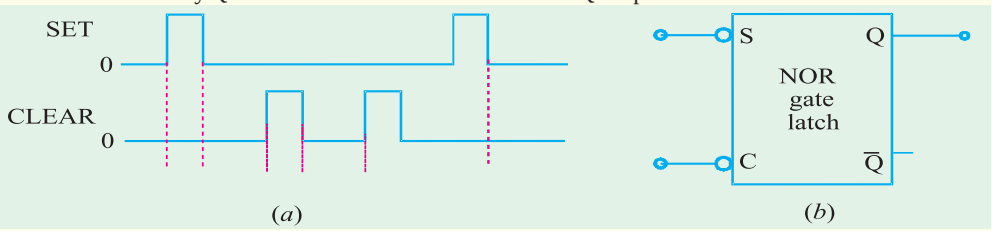


Fig. 72.103

3. Apply the *J-K* and CLK waveforms to a flip-flop shown in Fig. 72.104. Assume that $Q = 0$ initially. Sketch the output waveform.

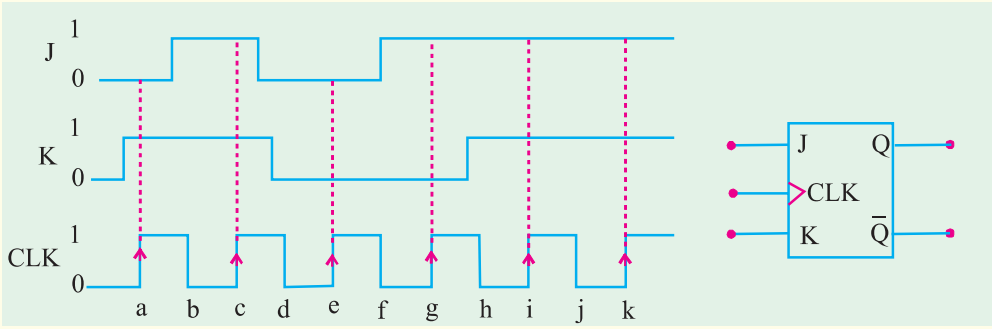


Fig. 72.104

4. Determine the output waveform for a positive-edge triggered *D* flip-flop for the *J*, *K* and CLK waveforms as shown in Fig. 72.105.

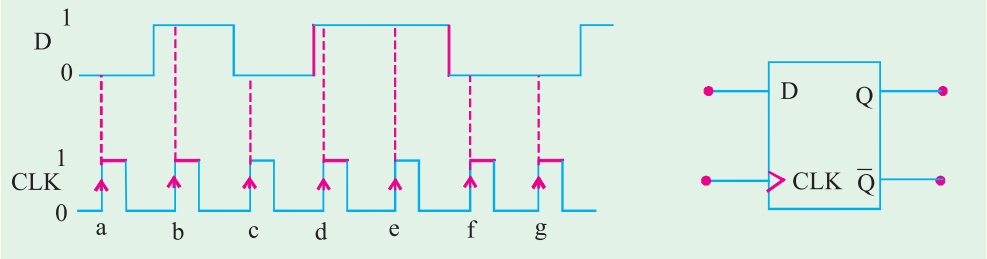


Fig. 72.105

5. Determine the *Q* waveform for a *D* latch with ENABLE (EN) and DATA (*D*) input waveforms shown in Fig. 72.106. Assume $Q = 0$ initially.

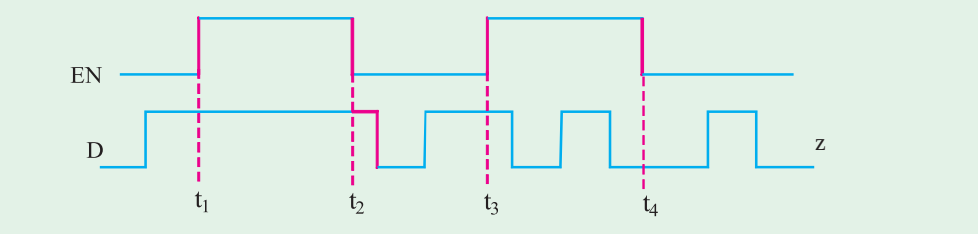


Fig. 72.106

6. Fig. 72.107 (*a*) shows the symbol for a *J-K* flip-flop that responds to a NGT (*i.e.* negative going trigger) on its clock input and has asynchronous inputs. Sketch the output waveform in response to the $\overline{CLK}$, $\overline{PRE}$ and $\overline{CLR}$ waveforms as shown in Fig. 72.107 (*b*).

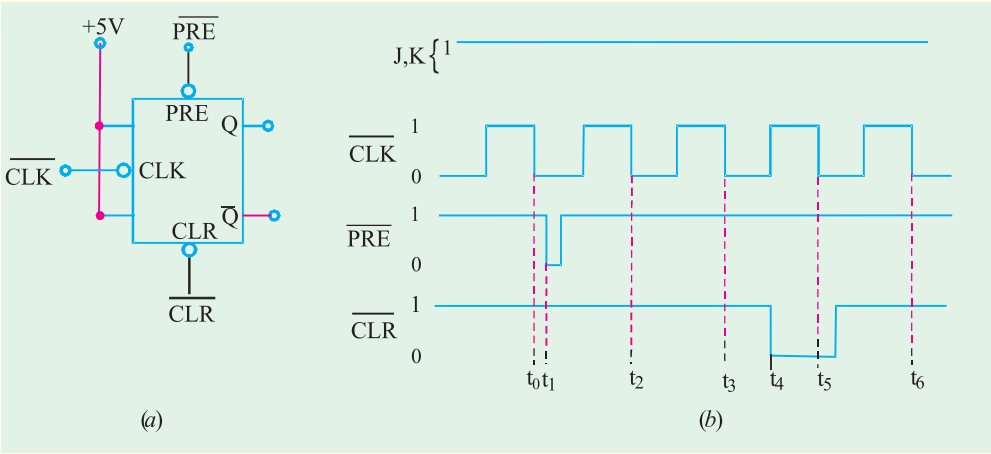


Fig. 72.107

7. Determine the *Q* output for a negative edge triggered *J-K* flip-flop for the input waveforms shown in Fig. 72.106. Assume $t_H = 0$ and that $Q = 0$ initially.

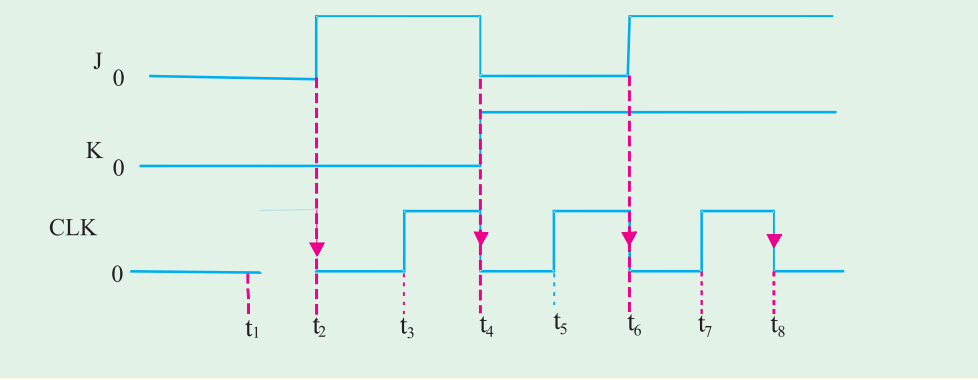


Fig. 72.108

8. A 20-kHz clock signal is applied to a *J-K* flip-flop with $J = K = 1$. Sketch the output waveform and determine its frequency. (**Ans** 10 kHz)

9. In a J-K flip-flop, $J = K = 1$. A 1 MHz square wave is applied at its clock input. It has a propagation delay of 50 ns. Draw the input square wave and the output waveform expected at Q. Show the propagation delay time.

10. *A D* flip-flop has following datasheet information :

Set up-time = 5 ns; Hold time = 10 ns; Propagation time = 15 ns

How far ahead of the triggering clock edge must the data be applied.

11. Draw the block diagram of a 4-bit shift register using *D* flip-flops. If initially all the flip-flop outputs are in zero state, prepare the truth table when the input sequence is 1, 1, 0, 1, 0. Draw the above shift register using *J-K* flip-flops only.

12. Consider a 4-bit shift register using J-K flip-flops. Assume that initially, $Q_0 = 1$, $Q_1 = 1$, $Q_2 = 1$ and $Q_3 = 1$. Sketch the output of each flip-flop if an input sequence 101101 is applied to D_0 synchronously with the clock (Q_0 is LSB).

13. Calculate the frequency and the duty cycle of the 555 astable multivibrator output for $C = 0.001\ \mu F$, $R_A = 2.2\ k\Omega$ and $R_B = 100\ k\Omega$. (**Ans** 50.5%)

OBJECTIVE TESTS – 72

1. If a NAND latch has a 1 on the SET input and a 0 on the CLEAR input, then the SET input goes to 0, the latch will be:

(*a*) HIGH (*b*) LOW

(*c*) Invalid (*d*) None of these

2. The invalid state of a NAND latch occurs when

(*a*) $S = 1$, $C = 0$

(*b*) $S = 0$, $C = 1$

(*c*) $S = 1$, $C = 1$

(*d*) $S = 0$, $C = 0$

3. The invalid state of a NOR latch occurs when

(*a*) $S = 1$, $C = 0$

(*b*) $S = 0$, $C = 1$

(*c*) $S = 1$, $C = 1$

(*d*) $S = 0$, $C = 0$

4. Like a latch, the flip-flop belongs to a category of logic circuits known as

(*a*) monostable multivibrators

(*b*) astable multivibrators

(*c*) bistable multivibrators

(*d*) none of these

5. Which of the following flip-flops is used as a latch?

(*a*) *J-K* flip-flop

(*b*) *S-C* flip-flop

(*c*) *D* flip-flop

(*d*) *T* flip-flop

6. A flip-flop which can have an uncertain output state is :

(*a*) *J-K* flip-flop

(*b*) *S-C* flip-flop

(*c*) *D* flip-flop

(*d*) *T* flip-flop

7. The purpose of a clock input to a flip-flop is to

(*a*) clear the device

(*b*) set the device

(*c*) always cause the output to change the states

(*d*) cause the output to assume a state dependent on the controlling (*S - C, J-K* or *D*) inputs.

8. A feature that distinguishes the *J-K* flip-flop from an *S-C* flip-flop is the

(*a*) toggle condition

(*b*) preset input

(*c*) type of clock

(*d*) clear input.

9. A J-K flip-flop is in the toggle condition when

(*a*) $J = 1$, $K = 0$

(*b*) $J = 1$, $K = 1$

(*c*) $J = 0$, $K = 0$

(*d*) $J = 0$, $K = 1$

10. A *J-K* flip-flop with $J = 1$ and $K = 1$ has a 10 kHz clock input. The Q = output is,

(*a*) constantly LOW

(*b*) constantly HIGH

(*c*) a 5 kHz square wave

(*d*) a 10 kHz square wave

11. For an edge-triggered *D* flip-flop,

(*a*) a change in the state of the flip-flop can occur only at a clock pulse edge

(*b*) the state that flip-flop goes to depends on the *D* input

(*c*) the output follows the input at each clock pulse

(*d*) all of these answers

12. The flip-flop shown in Fig. 72.109 logically behaves as

(*a*) a *D* flip-flop (*b*) *a J-K* flip-flop

(*c*) a *T* flip-flop (*d*) an *R-S* flip-flop

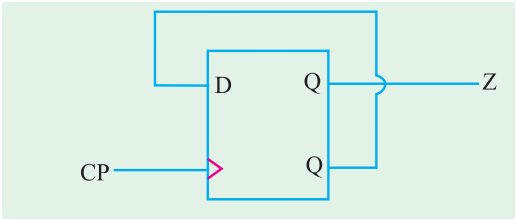


Fig. 72.109

(Grad. I.E.T.E. June, 1997)

13. A *J-K* flip-flop is a device to

(*a*) divide the frequency by 2

(*b*) divide the frequency by 4

(*c*) generate waveform of same frequency as that of the input

(*d*) cannot be used for frequency division.

14. A 1 µs pulse can be converted into a 1 ms pulse by using

(*a*) a monostable multivibrator

(*b*) an astable multivibrator

(*c*) a bistable multivibrator

(*d*) a *J-K* flip-flop

(U.P.S.C. Engg. Services, 1999)

15. The following is not a sequential circuit

(*a*) *J-K* flip-flop (*b*) counter

(*c*) full-adder (*d*) shift register

(Dip. I.E.T.E., Dec. 1996)

16. A one = shot is a type of

(*a*) monostable multivibrator

(*b*) astable multivibrator

(*c*) bistable multivibrator

(*d*) timer

17. An astable multivibrator

(*a*) requires a periodic trigger input

(*b*) has one stable states

(*c*) is an oscillator

(*d*) produces a non-periodic pulse output.

18. A 1 msec pulse can be converted to a 10 msec pulse by using:

(*a*) an astable multivibrator

(*b*) a monostable multivibrator

(*c*) a bistable multivibrator

(*d*) a J-K flip-flop

(U.P.S.C. Engg. Services 1990)

19. A retriggerable one shot is one which

(*a*) can be triggered only once

(*b*) has two quassi-stable states

(*c*) cannot be triggered until full pulse has been outputted.

(*d*) is capable of being triggered while the output is being generated.

20. The output of the circuit shown in Fig. 72.110 will be

(*a*) delayed pulses

(*b*) squarewaves

(*c*) trianglular waves

(*d*) trapezoidal waves.

(U.P.S.C Engg. Services 1997)

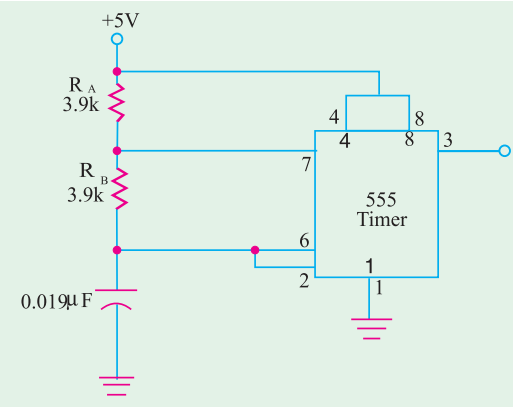


Fig. 72.110

21. The input waveform V_i and the output waveform V_o of a Schmitt NAND are shown in Fig 72.111. The duty cycle of the output waveform is

(*a*) 100 % (*b*) 85.5 %
(*c*) 72.2 % (*d*) 25 %

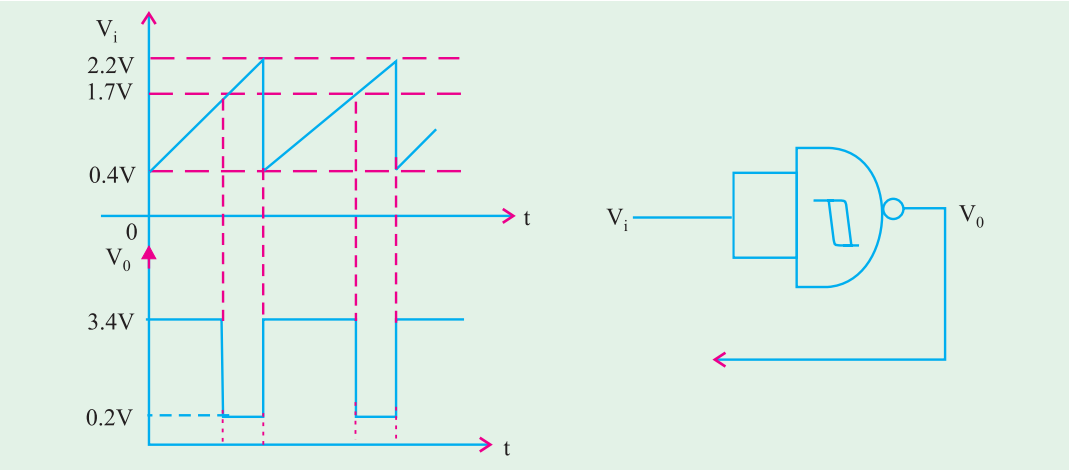


Fig. 72.111

(U.P.S.C Engg. Services. 1999)

22. Which of the following circuit is used for production of delays?

(*a*) astable multivibrator
(*b*) bistable multivibrator
(*c*) monostable multivibrator
(*d*) schmitt-trigger

23. A Schmitt-trigger can be used as a

(*a*) comparator only
(*b*) square-wave generator only
(*c*) flip-flop only
(*d*) comparator, square wave generator or flip-flop.

24. A 555 IC timer can be used to operate as

(*a*) as monostable multivibrator
(*b*) an astable multivibrator
(*c*) a voltage controlled oscillator
(*d*) all of the above

ANSWERS

1. (*a*)	**2.** (*d*)	**3.** (*c*)	**4.** (*c*)	**5.** (*b*)	**6.** (*b*)
7. (*d*)	**8.** (*a*)	**9.** (*b*)	**10.** (*c*)	**11.** (*d*)	**12.** (*c*)
13. (*a*)	**14.** (*a*)	**15.** (*c*)	**16.** (*a*)	**17.** (*c*)	**18.** (*b*)
19. (*d*)	**20.** (*b*)	**21.** (*c*)	**22.** (*c*)	**23.** (*a*)	**24.** (*d*)

GO To FIRST