CHAPTER 7

FLIP-FLOPs

7.1 INTRODUCTION

So far we have directed our studies towards the analysis and design of combinational digital circuits. Though very important, it constitutes only a part of digital systems. The other major aspect of digital systems is analysis and design of sequential circuits. However, sequential circuit design depends, to a large extent, on the combinational circuit design discussed earlier.

There are many applications in which digital outputs are required to be generated in accordance with the sequence in which the input signals are received. This requirement can not be satisfied using a combinational logic system. These applications require outputs to be generated that are not only dependent on the present input conditions but they also depend upon the past history of these inputs. The past history is provided by feedback from the output back to the input.

A block diagram of a sequential circuit is shown in Fig. 7.1. It consists of combinational circuits which accept digital signals from external inputs and from outputs of memory elements and generates signals for external outputs and for inputs to memory elements referred to as excitation.

A memory element is some medium in which one bit of information (1 or 0) can be stored or retained until necessary, and thereafter its contents can be replaced by a new value. The contents of memory elements in Fig. 7.1 can be changed by the outputs of the combinational circuit which are connected to its input.

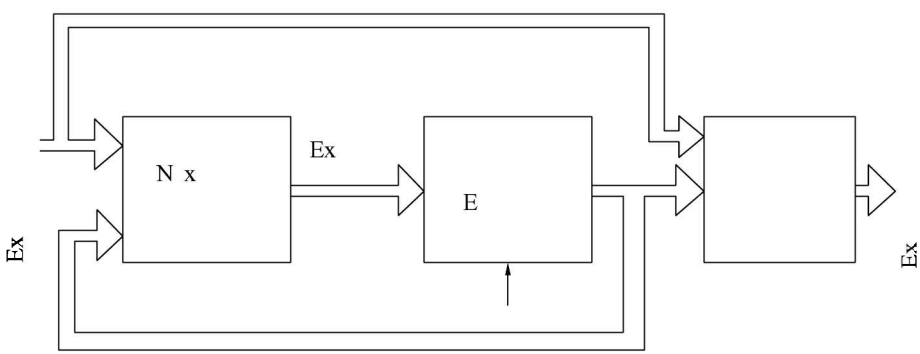


Fig. 7.1 ***Block Diagram of a Sequential Circuit***

The combinational circuit performs certain operations, some of which are used to determine the digital signals to be stored in memory elements. The other operations are performed on external inputs and memory outputs to generate the external outputs.

The above process demonstrates the dependence of the external outputs of a sequential circuit on the external inputs and the present contents of the memory elements (referred to as the *present state* of memory elements). The new contents of the memory elements, referred to as the *next state*; depend on the external inputs and the present state. Hence, the output of a sequential circuit is a function of the time sequence of inputs and the *internal states*.

Sequential circuits are classified in two main categories, known as *asynchronous* and *synchronous* sequential circuits depending on timing of their signals.

A sequential circuit whose behaviour depends upon the sequence in which the input signals change is referred to as an asynchronous sequential circuit. The outputs will be affected whenever the inputs change. The commonly used memory elements in these circuits are time delay devices. These can be regarded as combinational circuits with feedback.

A sequential circuit whose behaviour can be defined from the knowledge of its signal at discrete instants of time is referred to as a synchronous sequential circuit. In these systems, the memory elements are affected only at discrete instants of time. The synchronization is achieved by a timing device known as a *system clock* which generates a periodic train of clock pulses as shown in Fig. 7.2. The outputs are affected only with the application of a clock pulse.

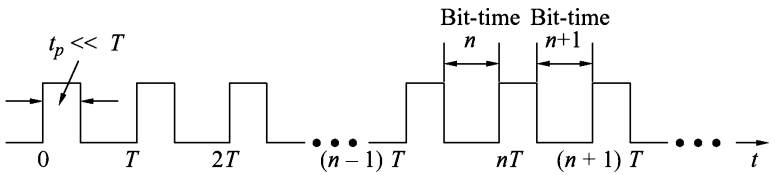


Fig. 7.2 ***A Train of Pulses***

Since the design of asynchronous circuits is more-tedious and difficult, therefore their uses are rather limited.

Synchronous circuits have gained considerable domination and wide popularity and are also known as *clocked-sequential circuits*. The memory elements used are FLIP-FLOPs which are capable of storing binary information.

7.2 A 1-BIT MEMORY CELL

The basic digital memory circuit is known as FLIP-FLOP. It has two stable states which are known as the *1 state* and the *0 state*. It can be obtained by using NAND or NOR gates. We shall be systematically developing a FLIP-FLOP circuit starting from the fundamental circuit shown in Fig. 7.3. It consists of two inverters G_1 and G_2 (NAND gates used as inverters). The output of G_1 is connected to the input of G_2 (A_2) and the output of G_2 is connected to the input of G_1 (A_1).

Let us assume the output of G_1 to be $Q = 1$, which is also the input of $G_2(A_2 = 1)$. Therefore, the output of G_2 will be $\overline{Q} = 0$, which makes $A_1 = 0$ and consequently $Q = 1$ which confirms our assumption.

In a similar manner, it can be demonstrated that if $Q = 0$, then $\overline{Q} = 1$ and this is also consistent with the circuit connections.

From the above discussion we note the following:

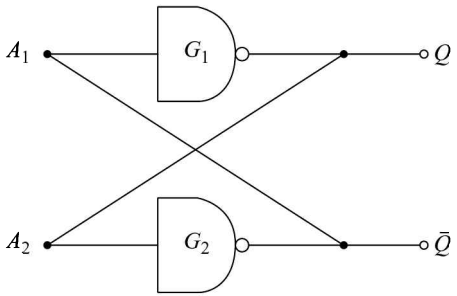


Fig. 7.3 ***Cross-coupled Inverters as a Memory Element***

1. The outputs Q and $\bar{Q}$ are always complementary.
2. The circuit has two stable states; in one of the stable state $Q = 1$ which is referred to as the 1 state (*or set state*) whereas in the other stable state $Q = 0$ which is referred to as the 0 state (*or reset state*).
3. If the circuit is in 1 state, it continues to remain in this state and similarly if it is in 0 state, it continues to remain in this state. This property of the circuit is referred to as *memory*, i.e. it can store 1-bit of digital information.

Since this information is locked or latched in this circuit, therefore, this circuit is also referred to as a *latch*.

In the latch of Fig. 7.3, there is no way of entering the desired digital information to be stored in it. In fact, when the power is switched on, the circuit switches to one of the stable states ($Q = 1$ or 0) and it is not possible to predict the state. If we replace the inverters G_1 and G_2 with 2-input NAND gates, the other input terminals of the NAND gates can be used to enter the desired digital information. The modified circuit is shown in Fig. 7.4. Two additional inverters G_3 and G_4 have been added for reasons which will become clear from the following discussion.

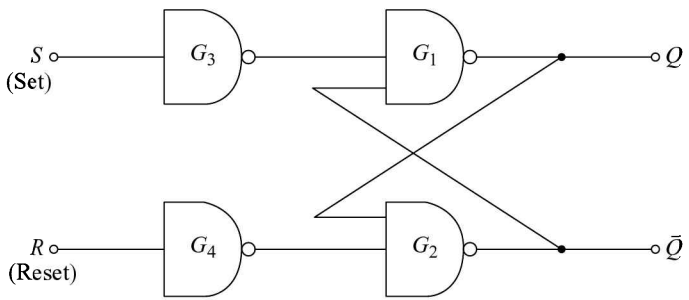


Fig. 7.4 ***The Memory Cell with Provision for Entering Data***

If $S = R = 0$, the circuit is exactly the same as that of Fig. 7.3 (Prob. 7.1). If $S = 1$ and $R = 0$, the output of G_3 will be 0 and the output of G_4 will be 1. Since one of the inputs of G_1 is 0, its output will certainly be 1. Consequently, both the inputs of G_2 will be 1 giving an output $\bar{Q} = 0$. Hence, for this input condition, $Q = 1$ and $\bar{Q} = 0$. Similarly, if $S = 0$ and $R = 1$ then the outputs will be $Q = 0$ and $\bar{Q} = 1$. The first of these two input conditions ($S = 1, R = 0$) makes $Q = 1$ which is referred to as the *set state*, whereas the second input condition ($S = 0, R = 1$) makes $Q = 0$ which is referred to as the *reset state* or *clear state*. This gives us the means for entering the desired bit in the latch.

Now we see what happens if the input conditions are changed from $S = 1, R = 0$ to $S = R = 0$ or from $S = 0, R = 1$ to $S = R = 0$. The output remains unaltered (Prob. 7.2). This shows the basic difference between a combinational circuit and a sequential circuit, even though the sequential circuit is made up of combinational circuits.

The two input terminals are designated as set (S) and reset (R) because $S = 1$ brings the circuit in set state and $R = 1$ brings it to reset or clear state.

If $S = R = 1$, both the outputs Q and $\bar{Q}$ will try to become 1 which is not allowed and therefore, this input condition is prohibited.

7.3 CLOCKED *S–R* FLIP-FLOP

It is often required to set or reset the memory cell (Fig. 7.4) in synchronism with a train of pulses (Fig. 7.2) known as clock (abbreviated as *CK*). Such a circuit is shown in Fig. 7.5, and is referred to as a *clocked set-reset* (*S–R*) FLIP-FLOP.

In this circuit, if a clock pulse is present ($CK = 1$), its operation is exactly the same as that of Fig. 7.4. On the other hand, when the clock pulse is not present ($CK = 0$), the gates G_3 and G_4 are inhibited, i.e. their outputs are 1 irrespective of the values of S or R. In other words, the circuit responds to the inputs S and R only when the clock is present.

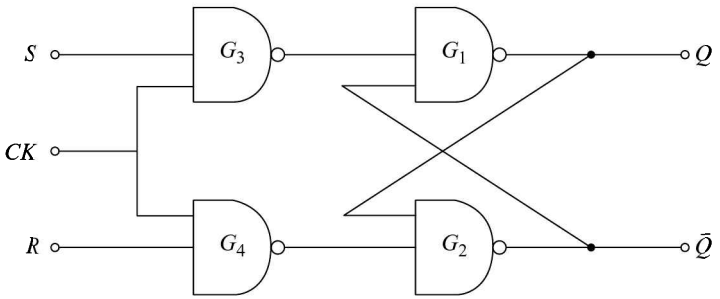


Fig. 7.5 ***A Clocked S–R*** FLIP-FLOP

Assuming that the inputs do not change during the presence of the clock pulse, we can express the operation of a FLIP-FLOP in the form of the truth table in Table 7.1 for the *S–R* FLIP-FLOP. Here S_n and R_n denote the inputs and Q_n the output during the bit time n (Fig. 7.2). Q_{n+1} denotes the output Q after the pulse passes, i.e. in the bit time $n + 1$.

Table 7.1 ***Truth Table of*** *S–R* FLIP-FLOP

Inputs		Output
S_n	R_n	Q_{n+1}
0	0	Q_n
1	0	1
0	1	0
1	1	?

If $S_n = R_n = 0$, and the clock pulse is applied, the output at the end of the clock pulse is same as the output before the clock pulse, i.e. $Q_{n+1} = Q_n$. This is indicated in the first row of the truth table.

If $S_n = 1$ and $R_n = 0$, the output at the end of the clock pulse will be 1, whereas if $S_n = 0$ and $R_n = 1$, then $Q_{n+1} = 0$. These are indicated in the second and third rows of the truth table respectively.

In the circuit of Fig. 7.4, it was mentioned that $S = R = 1$ is not allowed. Let us see what happens in the *S–R* FLIP-FLOP of Fig. 7.5 if $S_n = R_n = 1$. When the clock is present the outputs of gates G_3 and G_4 are both 0, making one of the inputs of G_1 and G_2 NAND gates 0. Consequently, Q and $\bar{Q}$ both will attain logic 1 which is inconsistent with our assumption of complementary outputs. Now, when the clock pulse has passed away ($CK = 0$), the outputs of G_3 and G_4 will rise from 0 to 1. Depending upon the propagation delays of the gates, either the stable state $Q_{n+1} = 1$ ($\bar{Q}_{n+1} = 0$) or $Q_{n+1} = 0$ ($\bar{Q}_{n+1} = 1$) will result. That means the state of the circuit is undefined, indeterminate or ambiguous and therefore is indicated by a question mark (fourth row of the truth table).

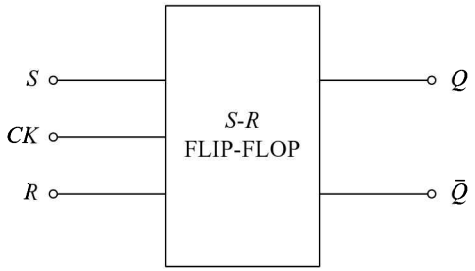


Fig. 7.6 ***Logic Symbol of Clocked*** *S–R* FLIP-FLOP

The condition $S_n = R_n = 1$ is forbidden and it must not be allowed to occur.

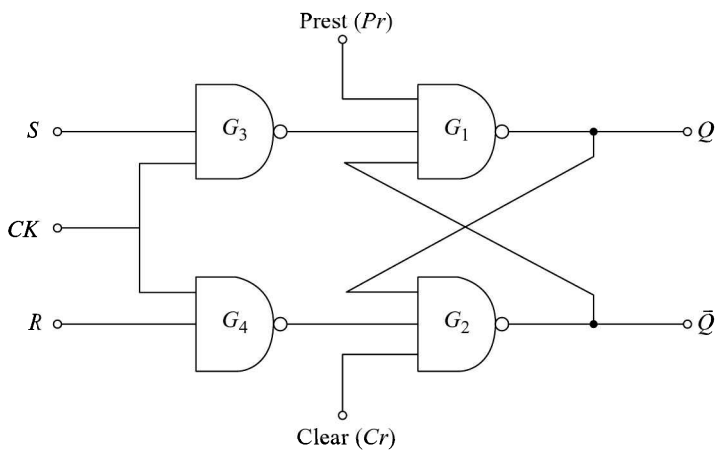
The logic symbol of clocked *S–R* FLIP-FLOP is given in Fig. 7.6.

7.3.1 Preset and Clear

In the FLIP-FLOP of Fig. 7.5, when the power is switched on, the state of the circuit is uncertain. It may come to set ($Q = 1$) or reset ($Q = 0$) state. In many applications it is desired to initially set or reset the FLIP-FLOP, i.e. the initial state of the FLIP-FLOP is to be assigned. This is accomplished by using the direct, or *asynchronous inputs*, referred to as *preset* (*Pr*) and *clear* (*Cr*) inputs. These inputs may be applied at any time between clock pulses and are not in synchronism with the clock. An *S–R* FLIP-FLOP with preset and clear is shown in Fig. 7.7. If $Pr = Cr = 1$ the circuit operates in accordance with the truth table of *S–R* FLIP-FLOP given in Table 7.1.

If $Pr = 0$ and $Cr = 1$, the output of $G_1(Q)$ will certainly be 1. Consequently, all the three inputs to G_2 will be 1 which will make $\bar{Q} = 0$. Hence, making $Pr = 0$ sets the FLIP-FLOP.

Similarly, if $Pr = 1$ and $Cr = 0$, the FLIP-FLOP is reset. Once the state of the FLIP-FLOP is established asynchronously, the asynchronous inputs *Pr* and *Cr* must be connected to logic 1 before the next clock is applied.



(a)

Fig. 7.7 ***(a) An*** *S–R* FLIP-FLOP ***with Preset and Clear, (b) Its Logic Symbol***

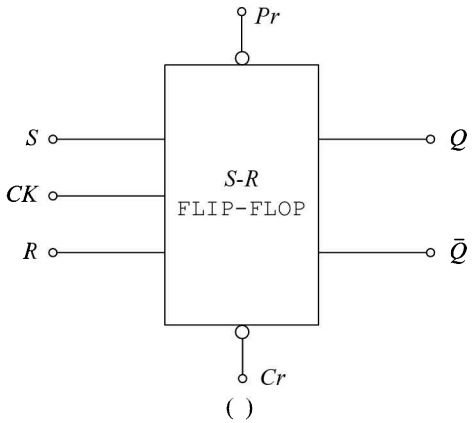


Fig. 7.7 ***(Continued)***

The condition $Pr = Cr = 0$ must not be used, since this leads to an uncertain state.

In the logic symbol of Fig. 7.7*b*, bubbles are used for *Pr* and *Cr* inputs, which means these are active-low, i.e. the intended function is performed when the signal applied to *Pr* or *Cr* is LOW The operation of Fig. 7.7 is summarised in Table 7.2.

The circuit can be designed such that the asynchronous inputs override the clock, i.e. the circuit can be set or reset even in the presence of the clock pulse (Prob. 7.4).

Table 7.2 ***Summary of Operation of S–R*** `FLIP-FLOP`

Inputs			Output	Operation performed
CK	***Cr***	***Pr***	***Q***	
1	1	1	Q_{n+1}(Table 7.1)	Normal `FLIP-FLOP`
0	0	1	0	Clear
0	1	0	1	Preset

7.4 *J–K* `FLIP-FLOP`

The uncertainty in the state of an *S–R* `FLIP-FLOP` when $S_n = R_n = 1$ (fourth row of the truth table) can be eliminated by converting it into a *J–K* `FLIP-FLOP`. The data inputs are *J* and *K* which are `AND`ed with $\bar{Q}$ and *Q*, respectively, to obtain *S* and *R* inputs, i.e.

$$S = J \cdot \bar{Q} \tag{7.1a}$$

$$R = K \cdot Q \tag{7.1b}$$

A *J–K* `FLIP-FLOP` thus obtained is shown in Fig. 7.8. Its truth table is given in Table 7.3*a* which is reduced to Table 7.3*b* for convenience. Table 7.3*a* has been prepared for all the possible combinations of *J*

and K inputs, and for each combination both the states of the output have been considered. The reader can verify this (Prob. 7.5).

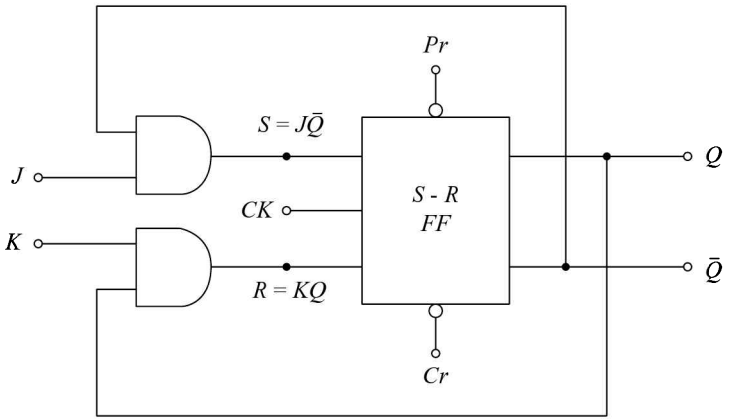


Fig. 7.8 ***An S–R*** `FLIP-FLOP` ***Converted into J–K*** `FLIP-FLOP`

Table 7.3a ***Truth Table for Fig.* 7.8**

Data inputs		Outputs		Inputs to S–R FF		Output
J_n	K_n	Q_n	$\bar{Q}_n$	S_n	R_n	Q_{n+1}
0	0	0	1	0	0	$\left.\begin{matrix}0\\1\end{matrix}\right] = Q_n$
0	0	1	0	0	0	
1	0	0	1	1	0	$\left.\begin{matrix}1\\1\end{matrix}\right] = 1$
1	0	1	0	0	0	
0	1	0	1	0	0	$\left.\begin{matrix}0\\0\end{matrix}\right] = 0$
0	1	1	0	0	1	
1	1	0	1	1	0	$\left.\begin{matrix}1\\0\end{matrix}\right] = \bar{Q}_n$
1	1	1	0	0	1	

Table 7.3b ***Truth Table of J–K*** FLIP-FLOP

Inputs		Output
J_n	K_n	Q_{n+1}
0	0	Q_n
1	0	1
0	1	0
1	1	$\bar{Q}_n$

It is not necessary to use the AND gates of Fig. 7.8, since the same function can be performed by adding an extra input terminal to each NAND gate G_3 and G_4 of Fig. 7.7 (Prob. 7.6). With this modification incorporated in Fig. 7.7, we obtain the *J–K* FLIP-FLOP using NAND gates as shown in Fig. 7.9. The logic symbol of *J–K* FLIP-FLOP is given in Fig. 7.10.

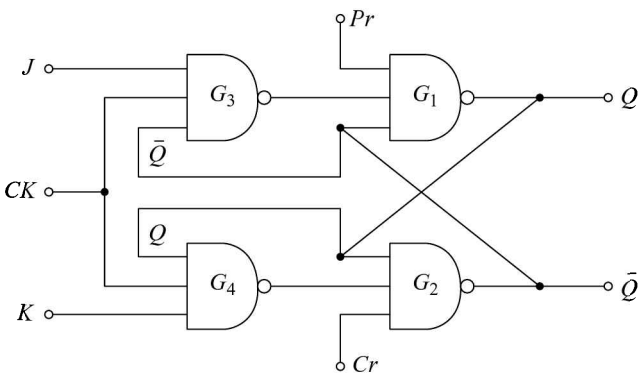


Fig. 7.9 ***A J–K*** FLIP-FLOP ***Using*** NAND ***Gates***

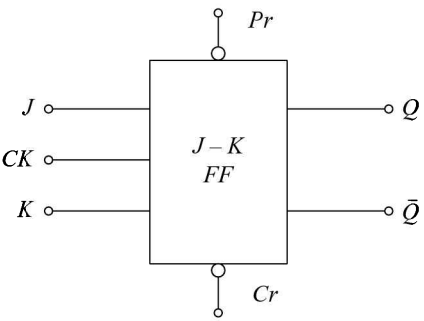


Fig. 7.10 ***Logic Symbols of J–K*** FLIP-FLOP

7.4.1 The Race-Around Condition

The difficulty of both inputs 1($S = R = 1$) being not allowed in an *S–R* FLIP-FLOP is eliminated in a *J–K* FLIP-FLOP by using the feedback connection from outputs to the inputs of the gates G_3 and G_4 (Fig. 7.9). Table 7.3 assumes that the inputs do not change during the clock pulse ($CK = 1$), which is not true because of the feedback connections. Consider, for example, that the inputs are $J = K = 1$ and $Q = 0$, and a pulse as shown in Fig. 7.11 is applied at the clock input. After a time interval Δt equal to the propagation delay through two NAND gates in series, the output will change to $Q = 1$ (see fourth row of Table 7.3b). Now we have $J = K = 1$ and $Q = 1$ and after another time interval of Δt the output will change back to $Q = 0$. Hence, we conclude that for the duration t_p of the clock pulse, the output will oscillate back and forth between 0 and 1. At the end of the clock pulse, the value of Q is uncertain. This situation is referred to as the *race-around condition*.

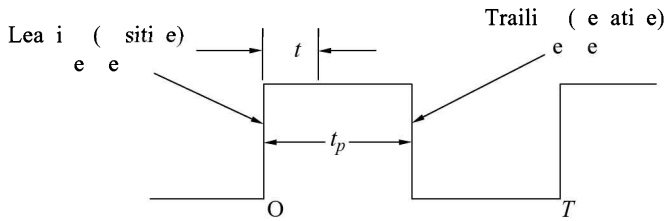


Fig. 7.11 ***A Clock Pulse***

The race-around condition can be avoided if $t_p < \Delta t < T$. However, it may be difficult to satisfy this inequality because of very small propagation delays in ICs. A more practical method for overcoming this difficulty is the use of the master–slave (*M–S*) configuration discussed below.

7.4.2 The Master–Slave *J–K* FLIP-FLOP

A master–slave *J–K* FLIP-FLOP is a cascade of two *S–R* FLIP-FLOPs, with feedback from the outputs of the second to the inputs of the first as illustrated in Fig. 7.12. Positive clock pulses are applied to the first FLIP-FLOP and the clock pulses are inverted before these are applied to the second FLIP-FLOP.

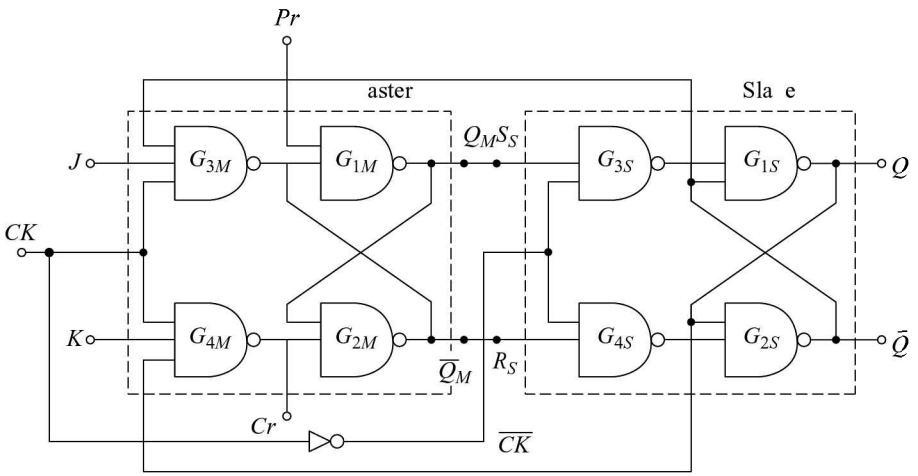


Fig. 7.12 ***A Master–Slave*** *J–K* FLIP-FLOP

When $CK = 1$, the first FLIP-FLOP is enabled and the outputs Q_M and $\overline{Q}_M$ respond to the inputs J and K according to Table 7.3. At this time, the second FLIP-FLOP is inhibited because its clock is LOW ($\overline{CK} = 0$). When CK goes LOW ($\overline{CK} = 1$), the first FLIP-FLOP is inhibited and the second FLIP-FLOP is enabled, because now its clock is HIGH ($\overline{CK} = 1$). Therefore, the outputs Q and $\overline{Q}$ follow the outputs Q_M and $\overline{Q}_M$, respectively (second and third rows of Table 7.3b). Since the second FLIP-FLOP simply follows the first one, it is referred to as the *slave* and the first one as the *master*. Hence, this configuration is referred to as *master—slave* (*M–S*) FLIP-FLOP.

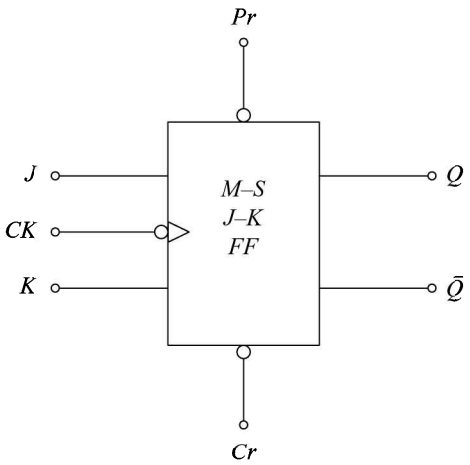


Fig. 7.13 ***A Master–Slave J–K*** `FLIP-FLOP` ***Logic Symbol***

In this circuit, the inputs to the gates G_{3M} and G_{4M} do not change during the clock pulse, therefore the race-around condition does not exist. The state of the master–slave `FLIP-FLOP` changes at the negative transition (trailing edge) of the clock pulse. The logic symbol of a *M–S* `FLIP-FLOP` is given in Fig. 7.13. At the clock input terminal, the symbol > is used to illustrate that the output changes when the clock makes a transition and the accompanying bubble signifies negative transition (change in *CK* from 1 to 0).

7.5 *D*-TYPE `FLIP-FLOP`

If we use only the middle two rows of the truth table of the *S–R* (Table 7.1) or *J–K* (Table 7.3b) `FLIP-FLOP`, we obtain a *D*-type `FLIP-FLOP` as shown in Fig. 7.14. It has only one input referred to as *D*-input or data input. Its truth table is given in Table 7.4 from which it is clear that the output Q_{n+1} at the end of the clock pulse equals the input D_n before the clock pulse.

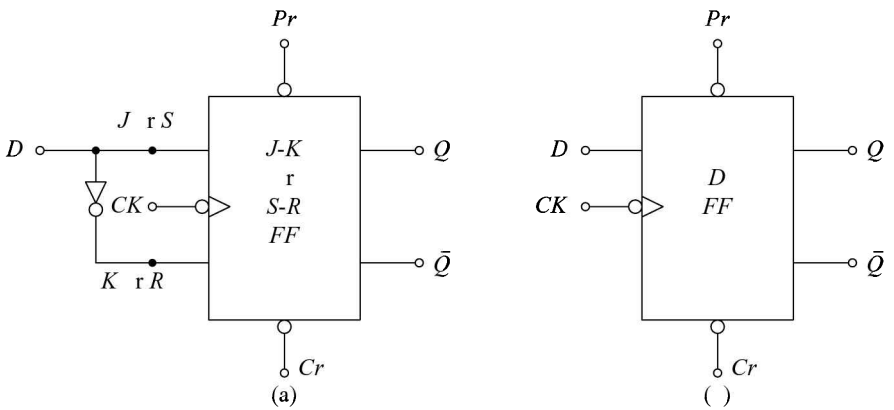


Fig. 7.14 ***(a) A J–K or S–R*** `FLIP-FLOP` ***Converted into a D-type*** `FLIP-FLOP` ***(b) its Logic Symbol***

Table 7.4 ***Truth Table of a D-type*** `FLIP-FLOP`

Input D_n	Output Q_{n+1}
0	0
1	1

This is equivalent to saying that the input data appears at the output at the end of the clock pulse. Thus, the transfer of data from the input to the output is delayed and hence the name *delay* (*D*) FLIP-FLOP. The *D*-type FLIP-FLOP is either used as a delay device or as a latch to store 1-bit of binary information.

7.6 *T*-TYPE FLIP-FLOP

In a *J–K* FLIP-FLOP, if $J = K$, the resulting FLIP-FLOP is referred to as a *T*-type FLIP-FLOP and is shown in Fig. 7.15. It has only one input, referred to as *T*-input. Its truth table is given in Table 7.5 from which it is clear that if $T = 1$ it acts as a toggle switch. For every clock pulse, the output Q changes.

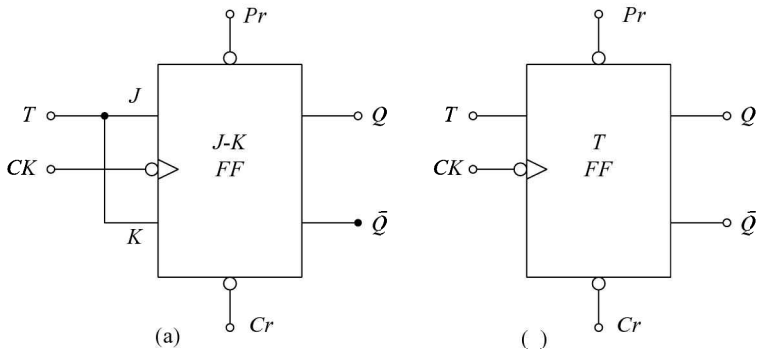


Fig. 7.15 ***(a) A J–K*** FLIP-FLOP ***Converted into a T-type*** FLIP-FLOP ***(b) its Logic Symbol***

Table 7.5 ***Truth Table of T-type*** FLIP-FLOP

Input	Output
T_n	Q_{n+1}
0	Q_n
1	$\bar{Q}_n$

An *S–R* FLIP-FLOP cannot be converted into a *T*-type FLIP-FLOP since $S = R = 1$ is not allowed. However, the circuit of Fig. 7.16 acts as a toggle switch, i.e. the output Q changes with every clock pulse (Prob. 7.11).

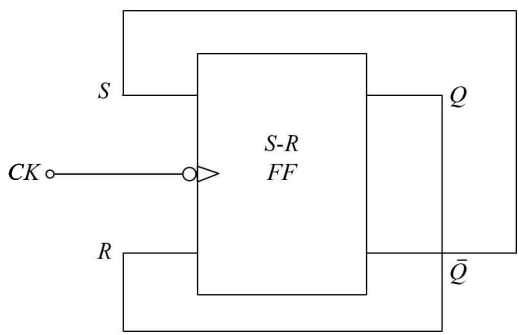


Fig. 7.16 ***An S–R*** FLIP-FLOP ***as a Toggle Switch***

7.7 EXCITATION TABLE OF FLIP-FLOP

The truth table of a FLIP-FLOP is also referred to as the *characteristic table* and specifies the operational characteristic of the FLIP-FLOP.

In the design of sequential circuits, we usually come across situations in which the *present state* and the *next state* of the circuit are specified, and we have to find the input conditions that must prevail to cause the desired transition of the state. By the present state and the next state we mean the state of the circuit prior to and after the clock pulse respectively. For example, the output of an *S–R* FLIP-FLOP before the clock pulse is $Q_n = 0$ and it is desired that the output does not change when the clock pulse is applied. What input conditions (S_n and R_n values) must exist to achieve this?

From the truth table (or the characteristic table) of an *S–R* FLIP-FLOP (Table 7.1) we obtain the following conditions:

1. $S_n = R_n = 0$ (first row)
2. $S_n = 0, R_n = 1$ (third-row)

We conclude from the above conditions that the S_n input must be 0, whereas the R_n input may be either 0 or 1 (don't-care). Similarly, input conditions can be found for all possible situations. A tabulation of these conditions is known as the *excitation table*. It is a very important and useful design aid for sequential circuits. Table 7.6 gives the excitation tables of *S–R*, *J–K*, *T*, and *D* FLIP-FLOPs. This is derived from the characteristic table of the FLIP-FLOP.

Table 7.6 ***Excitation Table of* FLIP-FLOPs**

Present State	Next State	*S–R* FF		*J–K* FF		*T*-FF	*D*-FF
		S_n	R_n	J_n	K_n	T_n	D_n
0	0	0	×	0	×	0	0
0	1	1	0	1	×	1	1
1	0	0	1	×	1	1	0
1	1	×	0	×	0	0	1

7.8 CLOCKED FLIP-FLOP DESIGN

In earlier sections, we defined or specified the operation of different FLIP-FLOPs assuming a circuit without regard to where the circuit came from or how it was designed. In this section, the design of a FLIP-FLOP is given. The design philosophy illustrated is, in fact, a general approach for the design of sequential circuits and systems.

Consider the general model of the FLIP-FLOP shown in Fig. 7.17. Basically, a clocked FLIP-FLOP is a sequential circuit which stores the bits 0 and 1. This operation is accomplished by using a binary cell

coupled with some combinational set/reset decoding logic to allow some input control over the set and reset operations of the cell. The steps for the design of FLIP-FLOP are given below.

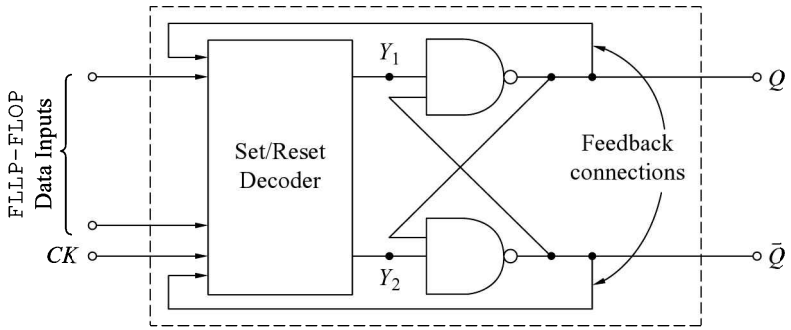


Fig. 7.17 ***The General Model of the*** FLIP-FLOP

Step 1. Examine each row of the given characteristic table, specifying the desired inputs and outputs, and answer the following questions and make a truth table with Y_1 and Y_2 as output variables.

1. Does the cell need to be set ($Q_{n+1} = 1$) for this condition?
2. Does the cell need to be reset ($Q_{n+1} = 0$) for this condition?
3. Does the cell need to be left as it is?

Step 2. Prepare the *K*-map for Y_1 and Y_2 output variables, minimize it and determine the logic for Y_1 and Y_2, respectively. Draw the complete circuit using gates.

The above design steps are illustrated in Example 7.1.

Example 7.1

Using the technique described above, design a clocked *S–R* FLIP-FLOP whose characteristic table is given in Table 7.7.

Solution

Step 1. Determine the values of Y_1 and Y_2 for each row. For example, for the first row $Q_n = 0$ and $Q_{n+1} = 0$. To obtain $Q_{n+1} = 0$, Y_1 must be equal to 1, since $\bar{Q}_n = 1$, Y_2 can be 0 or 1 since $Q_n = 0$. In a similar manner, complete the truth table (Table 7.7).

Table 7.7 ***Truth Table***

Characteristic table					Truth table for decoder	
CK	***S***	***R***	Q_n	Q_{n+1}	Y_1	Y_2
0	0	0	0	0	1	×
0	0	0	1	1	×	1
0	0	1	0	0	1	×
0	0	1	1	1	×	1
0	1	0	0	0	1	×
0	1	0	1	1	×	1
0	1	1	0	0	1	× } *
0	1	1	1	1	×	1 } *
1	0	0	0	0	1	×
1	0	0	1	1	×	1
1	0	1	0	0	1	×
1	0	1	1	0	1	0
1	1	0	0	1	0	1
1	1	0	1	1	×	1
1	1	1	0	×	×	× } **
1	1	1	1	×	×	× } **

*$S = R = 1$ can happen with no clock.
**$S = R = 1$ must not happen.

Step 2. The *K*-maps for Y_1 and Y_2 are given in Fig. 7.18 which give

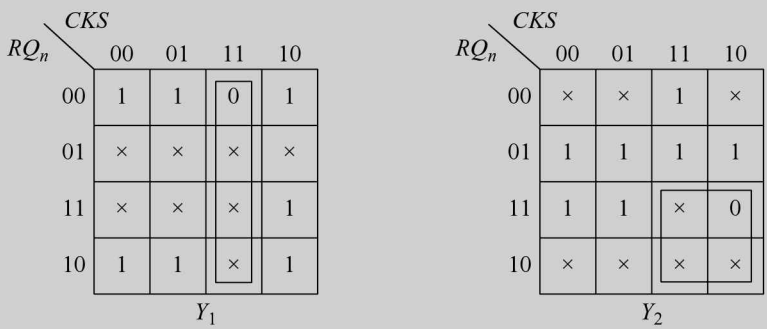


Fig. 7.18 ***K-maps for Ex. 7.1***

$$Y_1 = \overline{CK} + \overline{S} = \overline{CK \cdot S}$$

$$Y_2 = \overline{R} + \overline{CK} = \overline{CK \cdot R}$$

Thus, we see that the circuit resulting from this design is the same as that shown in Fig. 7.5.

7.8.1 Conversion from One Type of FLIP-FLOP to Another Type

In earlier sections, we have discussed conversion from *S–R* to *J–K*, *S–R* (or *J–K*) to *D*-type, and *J–K* to *T*-type FLIP-FLOPs. Now, we shall effect the conversion from one type of FLIP-FLOP to another type by using a formal technique which is similar to the one used above and will be useful in the design of clocked sequential circuits.

Consider the general model for conversion from one type of FLIP-FLOP to another type (Fig. 7.19). In this, we are required to design the combinational logic decoder (conversion logic) for converting new input definitions into input codes which will cause the given FLIP-FLOP to perform as desired.

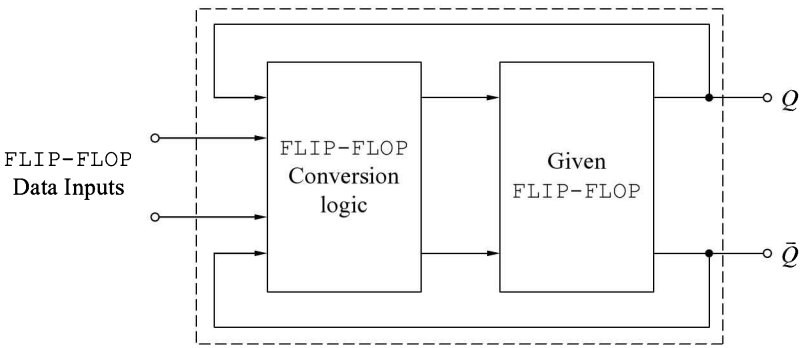


Fig. 7.19 ***The General Model Used to Convert One Type of*** FLIP-FLOP ***to Another Type***

To design the conversion logic we need to combine the excitation tables for both FLIP-FLOPs and make a truth table with data input(s) and Q as the inputs and the input(s) of the given FLIP-FLOP as the output(s). The conventional method of combinational logic design then follows as usual. The conversion is illustrated in Example 7.2.

Example 7.2

Convert an *S–R* FLIP-FLOP to a *J–K* FLIP-FLOP.

Solution

The excitation tables of *S–R* and *J–K* FLIP-FLOPs are given in Table 7.6 from which we make the truth table given in Table 7.8.

Table 7.8 ***Truth Table of Conversion Logic***

Row	*FF* data inputs		Output	*S–R FF* inputs	
	J	***K***	***Q***	***S***	***R***
1	0	0	0	0	×
2	0	1	0	0	×
3	1	0	0	1	0
4	1	1	0	1	0
5	0	1	1	0	1
6	1	1	1	0	1
7	0	0	1	×	0
8	1	0	1	×	0

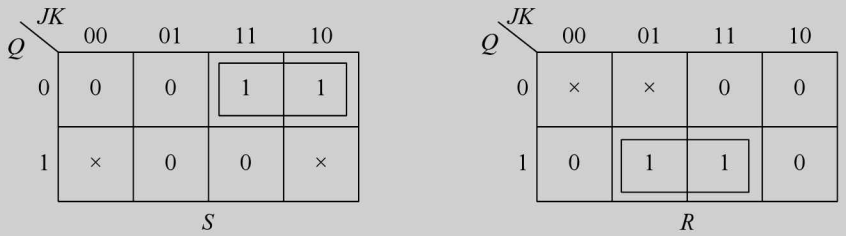


Fig. 7.20 ***K-maps for Ex. 7.2***

The *K*-maps are given in Fig. 7.20, which give

$$S = J \cdot \bar{Q} \qquad \text{and} \qquad R = K \cdot Q$$

Thus, we see that the circuit resulting from this design is the same as that shown in Fig. 7.8.

7.9 EDGE-TRIGGERED FLIP-FLOPs

All FLIP-FLOPs other than the master–slave type discussed in earlier sections are level triggered, i.e. the outputs respond to the inputs (according to the truth table) as long as the clock is present. The only ICs available in this category are latches, for example 7475 and 74100 are transparent latches.

The master–slave FLIP-FLOPs are also referred to as the *pulse-triggered* FLIP-FLOPs, i.e. the outputs respond to the inputs when a pulse is applied at the clock input. The master FLIP-FLOP responds when the

clock is present ($CK = 1$) and the output of the slave will be available at the falling edge of the clock pulse ($CK = 0$). As discussed in Section 7.4, this eliminates the problem of race-around condition. In this the data is *locked-out* at the falling edge of the clock pulse, i.e. the changes occurring at the inputs once CK goes to 0 will not affect the operation of the FLIP-FLOP.

The inputs to the FLIP-FLOP may change during the presence of the clock pulse due to certain operations in the system. This causes uncertainty in the outputs of the FLIP-FLOP which is eliminated by using edge-triggered FLIP-FLOPs.

In the case of an edge-triggered FLIP-FLOP, the transfer of information from data input(s) to the output of the FLIP-FLOP occurs at the positive (or negative) edge of the clock pulse. The only time the outputs can change state is during the brief interval of time when the clock signal is making a transition from the 0 to 1 (↑), or in some circuits, from the 1 to 0 (↓) states. A FLIP-FLOP which responds only to rising (or falling) edge is referred to as *positive-edge-triggered* (or *negative-edge-triggered*). The data lock-out occurs at the end of the edge.

The logic symbol used for an edge-triggered FLIP-FLOP is the same as that of a master–slave FLIP-FLOP. These are shown in Figs. 7.13, 7.14*b*, and 7.15*b*.

The timing specifications of an edge-triggered FLIP-FLOP are illustrated in Fig. 7.21 and are explained below.

Set-up Time (t_s) It is the time required for the input data to settle in before the triggering edge of the clock. Its minimum time is usually specified by the manufacturers.

Hold Time (t_h) It is the time for which the data must remain stable after the triggering edge of the clock. Minimum value of hold time is specified by the manufacturers.

The t_s and t_h timings are shown in Fig. 7.21*a*.

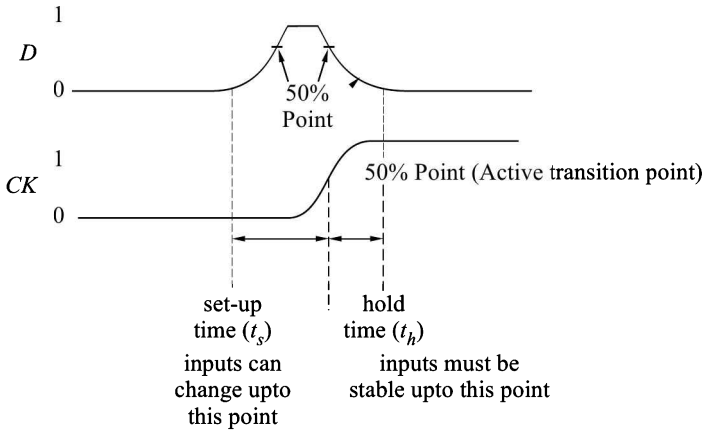
Propagation Delays Similar to propagation delay in combinational circuits, there is delay in the FLIP-FLOP output Q, making a change from HIGH-to-LOW or LOW-to-HIGH when a clock pulse is applied. These delays are specified between the 50% points on the clock and data input waveforms. The propagation delay, when the output Q changes from LOW-to-HIGH and HIGH-to-LOW, are specified as t_{pLH} and t_{pHL} respectively. These are shown in Fig. 7.21*b*.

Clock Pulse Width The minimum time duration for which the clock pulse must remain HIGH (t_{CH}) and LOW (t_{CL}) are specified by the manufacturers. Failure to meet these requirements may result in unreliable triggering. These timings are specified between the 50% points on the clock transitions and are shown in Fig 7.21*c*.

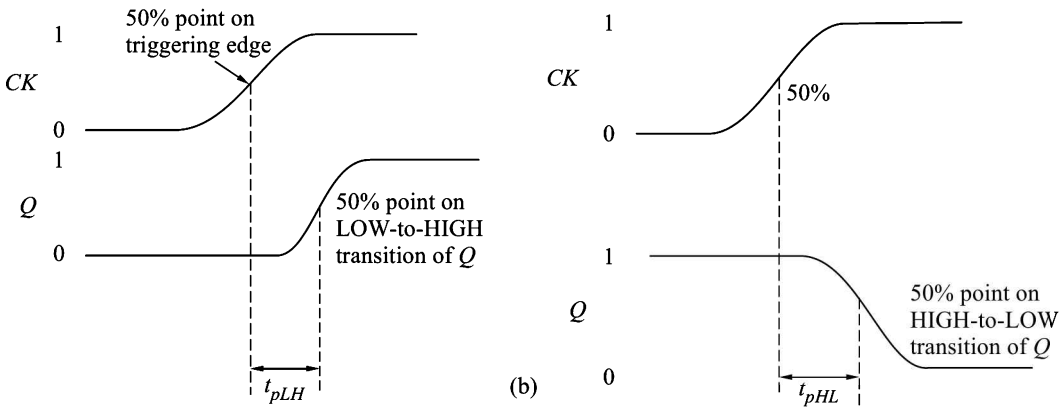
Preset and Clear Pulse Width The manufacturers also specify the minimum time duration for a preset input (t_{pLH}) and clear input (t_{pHL}).

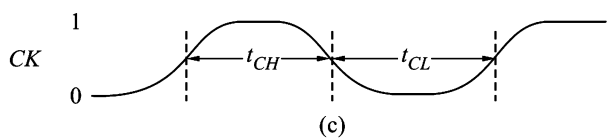
Maximum Clock Frequency The maximum clock frequency (f_{max}) is the highest rate at which a FLIP-FLOP can be reliably triggered. If the clock frequency is higher than this, the FF will be enable to trigger reliably.

Various timing values for some of the commonly used TTL and CMOS FLIP-FLOPs are given in Table 7.9.



(a)





(c)

Fig. 7.21 FLIP-FLOP ***Timings (a) Set-up and Hold Timings (b)Propagation Delays (c) Clock*** LOW ***and*** HIGH ***Timings***

7.9 *Timing Parameters of* TTL *and* CMOS FLIP-FLOP*s*

	TTL			CMOS			
arameter	**74LS74A**	**74LS112**	**74F74**	**74HC74A**	**74HC112**	**74AHC74**	**UN**
et up time)	20	20	2	14	25	5	n
hold time)	5	0	1	3	0	0.5	n
(*CK* to *Q*)	40	24	6.8	17	31	4.6	n
(*CK* to *Q*)	25	16	8	17	31	4.6	n
(*Cr* to *Q*)	40	24	9	18	41	4.8	n
(*Pr* to *Q*)	25	16	6.1	18	41	4.8	n
clock HIGH time)	25	20	4	10	25	5	n
clock LOW time)	25	15	5	10	25	5	n
(*Pr* or *Cr* OW time)	25	15	4	10	25	5	n
$_{AX}$ (Max. equency)	25	30	100	35	20	170	M

Example 7.3

The clock (Fig. 7.22*a*) and input (Fig. 7.22*b*) waveforms are applied to *D* or *J* input of each of the following type of FLIP–FLOPs. Sketch the output waveform in each case.

(a) Positive-edge-triggered *D*-type FLIP–FLOP 74 AHC74.
(b) Positive-level-triggered *D*-type FLIP–FLOP 7475 (transparent latch).
(c) Negative-edge-triggered *J–K* FLIP–FLOP (*K* = 1) 74 HC112.
(d) Master–slave *J–K* FLIP–FLOP (*K* = 1) 7476.

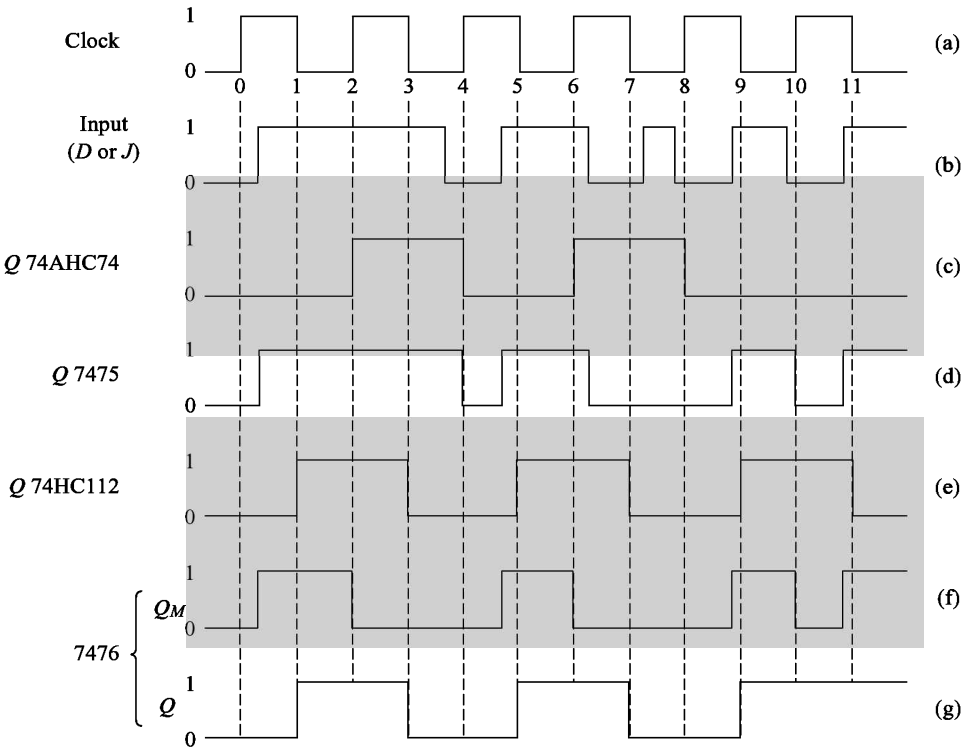


Fig. 7.22 ***Waveforms of Ex. 7.3***

Solution

(a) In the case of positive-edge-triggered *D*-type FLIP–FLOP the output *Q* is same as the input *D* at the positive edge of the clock pulse. The output does not change till the next positive edge arrives. The output (*Q*) waveform is shown in Fig. 7.22c.
(b) 7475 is a transparent latch, i.e. the output (*Q*) follows the input (*D*) as long as *CK* = 1. The output does not change when *CK* = 0. The output (*Q*) waveform is shown in Fig. 7.22*d*.
(c) In the case of negative-edge-triggered *J–K* FLIP–FLOP the output (*Q*) responds to the *J* and *K* inputs present at

the negative edge of the clock pulse (according to *J–K* truth table). The output does not change till the arrival of the next negative edge. The output (Q) waveform is shown in Fig. 7.22*e*.

(d) In the case of the master–slave *J–K* FLIP-FLOP, the output of the master responds to the *J* and *K* inputs present when $CK = 1$ (according to *J–K* truth table). The output of master (Q_M) is shown in Fig. 7.22*f*.

The output of the slave (Q) follows Q_M at the negative edge of the clock-pulse. The output (Q) waveform is shown in Fig. 7.22*g*.

7.10 APPLICATIONS OF FLIP-FLOPs

Some of the common uses of FLIP-FLOPs are as:

1. Bounce elimination switch,
2. Latch,
3. Registers,
4. Counters,
5. Memory, etc.

Some examples of the uses of FLIP-FLOPs are given below.

7.10.1 Bounce-Elimination Switch

Mechanical switches are employed in digital systems as input devices by which digital information (0 or 1) is entered into the system. There is a very serious problem associated with these switches, viz. *switch-bouncing* (or *chattering*). When the arm of the switch is thrown from one position to another, it chatters or bounces several times before finally coming to rest in the position of contact. The bounce is the result of the spring-loaded impact of the switch throw contact and the pole contacts.

In a sequential circuit, if a 1 is to be entered through a switch, then the switch is thrown to the corresponding position. As soon as it is thrown to this position, the output is 1 but the output oscillates between 0 and 1 for some time due to make and break (bouncing) of the switch at the point of contact before coming to rest. This changes the output of the sequential circuit and creates difficulties in the operation of the system. This problem is eliminated by using bounce-elimination switches.

Example 7.4

Show that the circuit of Fig. 7.23*a* acts as a bounce-elimination (chatterless) switch.

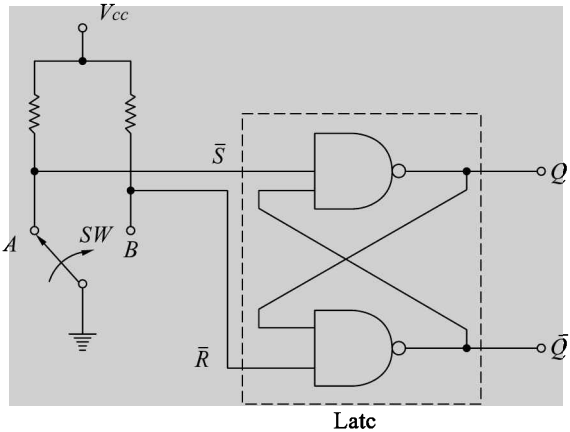
Solution

The waveforms at $\overline{S}$, $\overline{R}$, Q, and $\overline{Q}$ are illustrated in Fig. 7.23*b*. The switch (SW) is thrown from position *A* to *B* at $t = 0$. Therefore, at $t = 0^+$ the voltage at $\overline{S}$ will be V_{CC} (logic 1) and will continue to remain so as long as the switch is not thrown to position *A* again.

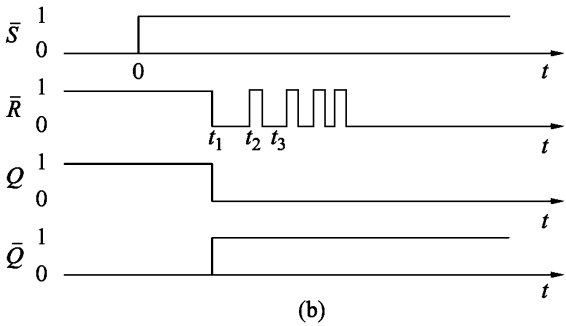
At $\overline{R}$ (B), the voltage at $t = 0^-$ is V_{CC} (logic 1) and goes to 0 V (logic 0) at t_1 (t_1 being the time delay of the switch). The switch arm makes contact at B at $t = t_1$, and then bounces off. Therefore, the level at $\overline{R}$ changes from 0 to 1 and vice-versa. This is illustrated in Fig. 7.23*b*.

Between $t = 0$ and $t = t_1$, both the inputs $\overline{S}$ and $\overline{R}$ are at logic 1 and therefore, Q does not change. The output Q changes at t_1 and becomes 0. Now, even when $\overline{R}$ is changing at t_2, t_3, etc. Q does not change. This shows that it is a chatterless switch.

The latch used in Fig. 7.23*a* can be replaced by the IC 74279 which is a quad $\overline{S} - \overline{R}$ latch.



(a)



(b)

Fig. 7.23 ***(a) A Bounce-Elimination Switch (b) Waveforms of $\overline{S}$, $\overline{R}$, Q, and $\overline{Q}$***

7.10.2 Registers

A register is composed of a group of FLIP-FLOPs to store a group of bits (word). For storing an N-bit word, the number of FLIP-FLOPs required is N (one FLIP-FLOP for each bit). A 3-bit register using 7474 positive-edge-triggered FLIP-FLOPs is shown in Fig. 7.24. The bits to be stored are applied at the D-inputs which are clocked in at the leading-edge of the clock pulse. In this register, the data to be entered must be available in parallel form. Other types of registers will be discussed in Chapter 8.

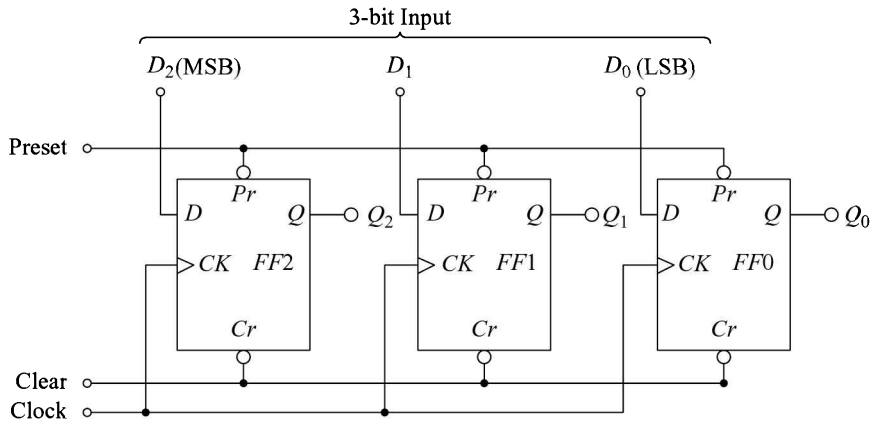


Fig. 7.24 ***A 3-bit Register Using*** `FLIP-FLOPs`

7.10.3 Counters

Digital counters are often needed to count events. For example, counting the number of tablets filled in a vial. Electrical pulses corresponding to the event are produced using a transducer and these pulses are counted using a counter.

The counters are composed of `FLIP-FLOPs`. A 3-bit counter consisting of three `FLIP-FLOPs` is shown in Fig. 7.25. A circuit with *n*- `FLIP-FLOPs` has 2^n possible states. Therefore, the 3-bit counter can count from decimal 0 to 7.

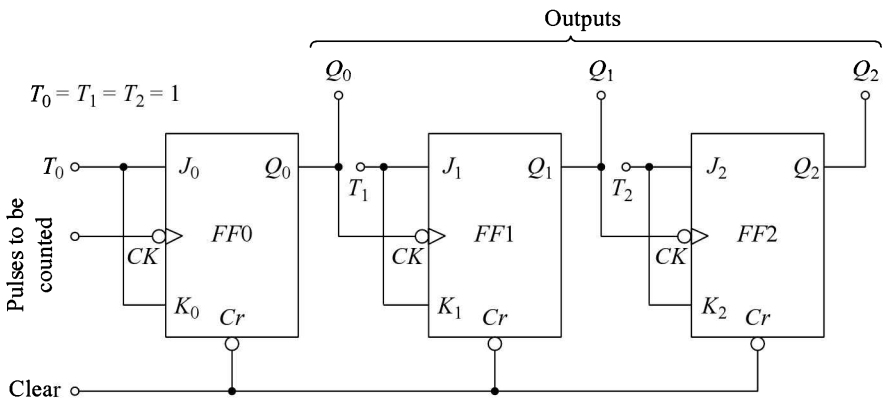


Fig. 7.25 ***A 3-bit Counter Using*** `FLIP-FLOPs`

The `FLIP-FLOPs` used are 74107 *J–K* master–slave `FLIP-FLOPs`, used as *T*-type. The pulses to be counted are connected at the clock input of FF0. The Q_0 output of FF0 is connected to the clock input of FF1 and similarly Q_1 is connected to the clock input of FF2.

The FLIP-FLOPs are cleared by applying logic 0 at the clear input terminal momentarily. For normal counting operation, it is to be maintained at logic 1. The pulses and the output waveforms are illustrated in Fig. 7.26.

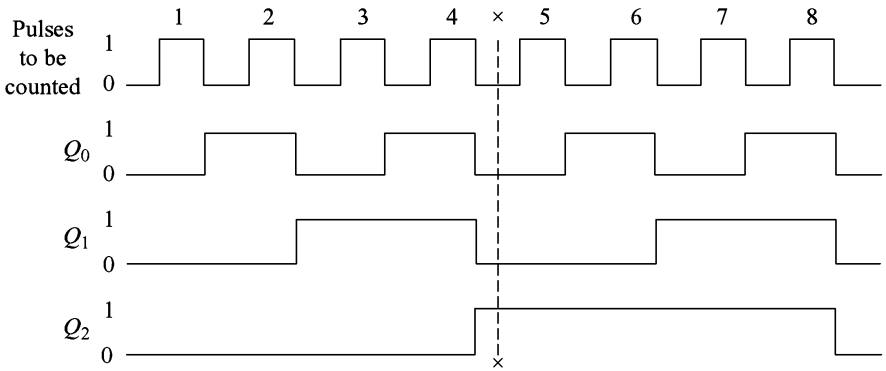


Fig. 7.26 ***Waveforms of Counter of Fig. 7.25***

The output Q_0 of the least-significant stage changes at the negative edge of each pulse (since $T_0 = 1$). The output Q_1 changes at the negative edge of each Q_0 pulse (since Q_0 acts as *CK* for FF1 and $T_1 = 1$) and the output Q_2 changes at the negative edge of each Q_1 pulse (since Q_1 acts as *CK* for FF2 and $T_2 = 1$).

At any time, the decimal equivalent of the binary number $Q_2 Q_1 Q_0$ is the number of pulses counted till that time. For example, at × the count is 100 (decimal 4). The circuit resets after counting eight pulses.

There are various types of counters, some of which will be discussed in the next chapter.

7.10.4 Random-Access Memory

In computers, digital control systems, information processing systems, etc. it is necessary to *store* digital data and *retrieve* the data as desired. For this purpose, earlier only magnetic memory devices were possible, whereas these days it has become possible to make memory devices using semiconductor devices. Semiconductor memories have become very popular because of their small size (available in ICs) and convenience to use. Chapter 11 deals with various semiconductor memories.

FLIP-FLOPs can be used for making memories in which data can be stored for any desired length of time and then read out whenever required. In such a memory, data can be put into (*writing* into the memory) or retrieved from (*reading* from the memory) the memory in a random fashion and is known as *random-access memory*.

A 1-bit read/write memory is shown in Fig. 7.27 which is the basic memory element and memory ICs are built around a system of basic 1-bit cell.

In this memory cell, a level *D* FLIP-FLOP is used which has *Q* output that follows the *D* input as long as *CK* terminal is at logic 1. The moment the *CK* input changes to logic 0, the *Q* output does not change and it retains the *D* input level that existed just before the transition from 1 to 0 at input *CK*. This input is used to select the memory cell. In the 1-bit cell shown there are three inputs — D_i (data input), A_n (address select) and $R/\overline{W}$ (read/write control) and one output D_0 (data output). $A_n = 1$ enables the cell for reading or writing

operation, $R/\overline{W}$ at logic 1 is for reading from the cell and logic 0 for writing into the cell. As long as $A_n = 0$, all input and output activities are blocked, and the cell is in the hold mode where its stored output is protected.

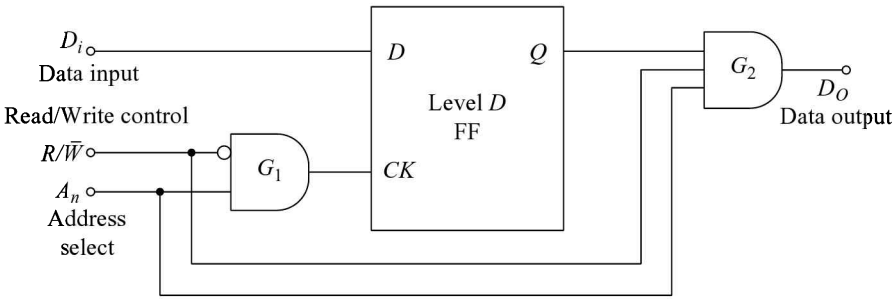


Fig. 7.27 ***A 1-bit Read/Write Memory Cell***

The complete function of this cell can be understood from the function table of Table 7.10. The read operation is nondestructive, that is, the stored bit can be read out any number of times without disturbing it. The stored bit will be protected as long as power is on. Therefore, this type of memory is known as *volatile memory*.

Table 7.10 ***Function Table of 1-bit Memory Cell***

Inputs			Mode
A_n	$R/\overline{W}$	D_i	
0	×	×	Hold, $D_0 = 0$
1	0	0	Write 0 into memory, $D_0 = 0$
1	0	1	Write 1 into memory, $D_0 = 0$
1	1	×	Read, D_0 = stored D_i bit.

As far as writing into the cell is concerned, it is not required to be cleared before entering the new bit. Whenever a new bit is entered the earlier one gets destroyed automatically.

SUMMARY

The basic element of sequential circuits, FLIP-FLOP has been introduced here. This is a basic memory element which is used to store 1-bit of digital information. The four types of commonly used FLIP-FLOPs *S–R*, *J–K*, *T*-type, and *D*-type have been covered in detail, including their design using gates.

The various triggering systems have been discussed which will be very helpful in understanding the detailed operation of the FLIP-FLOP and other circuits containing these FLIP-FLOPs.

Simple examples of the uses of FLIP-FLOPs in registers, counters, memory elements, etc. have been included. These topics will be covered in greater detail in later chapters.

GLOSSARY

Asynchronous sequential circuit A sequential circuit whose behaviour depends upon the sequence in which the input signals change. It is event driven and does not require clock pulses.

Bit time Amount of time to transmit a single bit.

Bouncing Moving back and forth between two states before reaching a final state. Same as chattering.

Bounce-elimination circuit A circuit that eliminates the effect of switch bouncing.

Characteristic table A table describing operation of a FLIP-FLOP.

Chatterless switch A switch in which the bouncing effect has been eliminated.

Clear Setting the contents of a FLIP-FLOP or a circuit containing FLIP-FLOPs or a memory to zero.

Clear input The input used for clearing a digital circuit.

Clock A train of pulses of usually constant frequency that synchronize the operation of any synchronous sequential circuit including a microprocessor based system.

Clock cycle The interval between successive positive or negative transitions in a clock.

Clocked FLIP-FLOP A FLIP-FLOP that responds to the data inputs only on the occurrence of the appropriate clock signal.

Clock frequency The number of clock cycles per second.

Clocked sequential circuit The sequential circuits whose operation is synchronized with the application of clock pulses, between which no changes of state occur.

Counter A digital circuit that can count the number of pulses.

Data Information in digital (binary) form.

Debouncing switch Same as chatterless switch.

D-type FLIP-FLOP A FLIP-FLOP whose output follows the input when triggered.

Edge-triggered FLIP-FLOP A FLIP-FLOP whose state changes on the rising (positive) or falling (negative) edge of a clock pulse.

Excitation table A tabular representation of the operation of a FLIP-FLOP.

Falling edge A transition from high to low voltage.

Frequeucy The repetition rate of a cyclic signal. It is expressed in Hertz (Hz).

Hold time In FLIP-FLOPs and memories, a minimum amount of time after the application of a clock (or Read/write) signal, when data(s) or address inputs must remain stable.

J–K FLIP-FLOP A FLIP-FLOP with inputs J and K. The state of the FLIP-FLOP does not change when $J = K = 0$, whereas it changes with every clock pulse when $J = K = 1$. The FLIP-FLOP is set when $J = 1$ and $K = 0$ and reset when $J = 0$ and $K = 1$. All the above operations are performed in synchronism with the clock.

Latch A temporary storage device consisting of D-type FLIP-FLOPs. Its contents are fixed at their current values by a transition of the clock and remain fixed until the next clock transition occurs.

Leading edge A transition from low to high voltage.

Level-triggereed FLIP-FLOP A FLIP-FLOP that is triggered (i.e. the outputs respond to the inputs) when the level of the clock signal is appropriate.

Master-slave FLIP-FLOP A FLIP-FLOP consisting of the cascade of two FLIP-FLOPs; the first FF is the master and the second FF is the slave. The master FF is triggered when the clock is HIGH and the slave FF follows the master FF when the clock goes LOW.

Preset The state of a FLIP-FLOP when $Q = 1$ and $\bar{Q} = 0$

Preset input The input used for setting a FLIP-FLOP.

Pulse-triggered FLIP-FLOP A FLIP-FLOP that requires a clock pulse for triggering. It is same as the master-slave FF.

Random-access memory (RAM) The ability to address a semiconductor memory for read-and-write operation in which any memory location can be accessed at random. It is same as Read and Write memory.

Read (memory) Process of getting (retrieving) a word from a memory.

Register An array of FLIP-FLOPs used to store binary information.

Reset Same as clear.

Reset input Same as the clear input

Rising edge Same as the leading edge.

Set $Q = 1$ state of a FLIP-FLOP.

Setup time The time required for the input data to settle in before the triggering edge of the clock pulse.

S–R FLIP-FLOP A FLIP-FLOP with inputs S and R. The state of the FF does not change when $S = R = 0$. It is set when $S = 1$ and $R = 0$; reset when $S = 0$ and $R = 1$. $S = R = 1$ is not allowed.

Stable state A state in which a digital circuit remains forever unless changed by a triggering signal.

State The value of FLIP-FLOP(s) output in a digital circuit.

Switch bouncing Fluctuations in the switch positions between ON and OFF when the switch position is changed.

Synchronous sequential circuit A sequential circuit whose operation is synchronized with the application of clock pulses, between which no change of state can occur.

Toggling Causing as FF to change its state.

Trailing edge Same as the falling edge.

Trigger To cause change of state.

T-type *FLIP-FLOP* A FLIP-FLOP with one data input (T) which changes state for every clock pulse if $T = 1$ and does not change the state if $T = 0$.

Unstable state The state which is not a stable state. The circuit comes out of this state without applying any triggering signal.

Write (memory) Process of placing (storing) a word into a specific memory location.

REVIEW QUESTIONS

7.1 FLIP-FLOP is a _____ element.

7.2 Number of FLIP-FLOPs required for storing n-bit of information is _____.

7.3 In an S–R FLIP-FLOP $S = R = 1$ _____ permitted.

7.4 'Preset' and 'Clear' inputs are used in a FLIP-FLOP for making Q = _____ and _____ respectively.
7.5 In a J–K FLIP-FLOP if $J = K = 1$, its Q output will be _____ when a clock pulse is applied.
7.6 Master-slave configuration is used in a J–K FLIP-FLOP to eliminate _____.
7.7 In a T FLIP-FLOP, the Q output _____ when $T = O$ and clock pulse is applied.
7.8 A FLIP-FLOP has _____ states.
7.9 A latch is used to store 1 _____ of data.
7.10 A negative edge-triggered FLIP-FLOP changes state at the _____ of the clock pulse.
7.11 An active low 'Clear' input clears the FLIP-FLOP when it is _____.
7.12 A FLIP-FLOP with active-low 'preset' input will have $\overline{Q}$ = _____ when preset is connected to LOW.
7.13 A chatterless switch can be implemented using a _____.
7.14 A tabulation specifying inputs required for a FLIP-FLOP to change from a present state to a specified next state is known as _____.
7.15 Registers and Counters can be designed using _____.

PROBLEMS

7.1 Show that the circuit of Fig. 7.4 with $S = R = 0$ is the same as that of Fig. 7.3.
7.2 In a circuit of Fig. 7.4 if the inputs change from
(a) $S = 1$, $R = 0$ to $S = R = 0$, and
(b) $S = 0$, $R = 1$ to $S = R = 0$,
show that the outputs do not change.
7.3 Design an S–R latch using two 2-input NOR gates
7.4 In the FLIP-FLOP circuit of Fig. 7.28 show that if
(a) $Pr = 0$ and $Cr = 1$, then $Q = 1$ (independent of S, R, and CK).
(b) $Pr = 1$ and $Cr = 0$, then $Q = 0$ (independent of S, R, and CK).

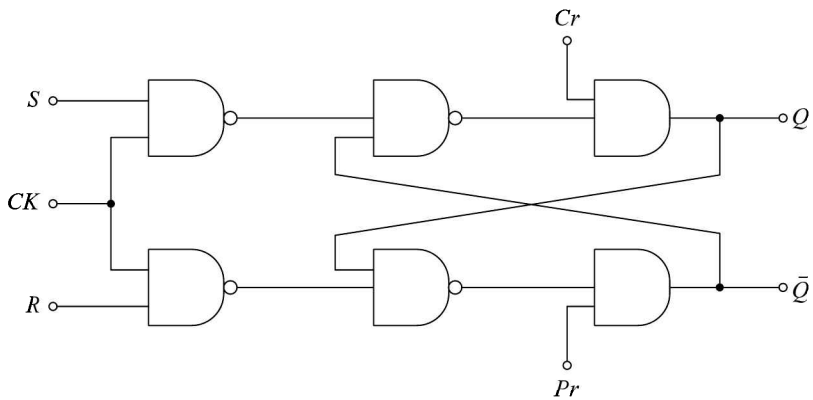


Fig. 7.28 FLIP-FLOP ***Circuit for Problem 7.4***

(c) $Pr = Cr = 1$, then it functions as a clocked *S–R* FLIP-FLOP.

7.5 Verify Table 7.3a.

7.6 Determine the output Y_1 of Fig. 7.29*a* and Y_2 of Fig. 7.29*b* and show that $Y_1 = Y_2$.

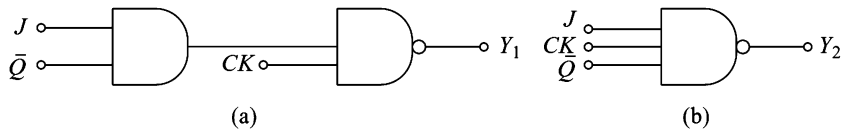


Fig. 7.29 ***Figures for Problem 7.6***

7.7 Identify Q and $\bar{Q}$ outputs of the clocked *J–K* FLIP-FLOP shown in Fig. 7.30.

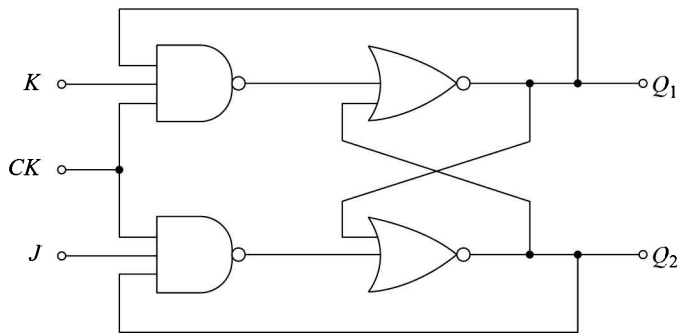


Fig. 7.30 ***Figure for Problem 7.7***

7.8 For the circuit shown in Fig. 7.31 the clock and input waveforms shown in Fig. 7.32 are applied. Sketch the waveforms of Q and $\bar{Q}$ if the FLIP-FLOP is edge-triggered.

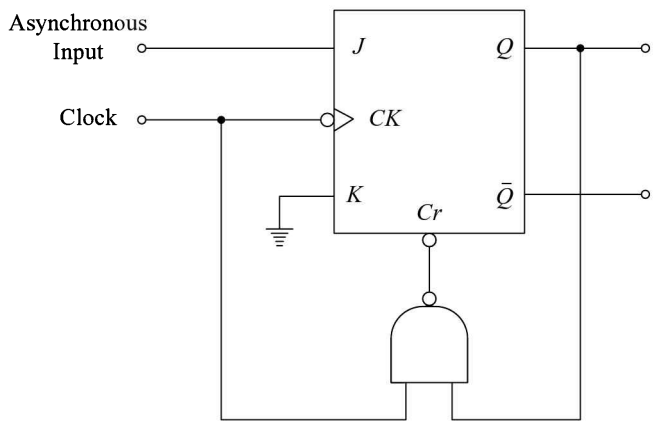


Fig. 7.31 ***Figure for Problem 7.8***

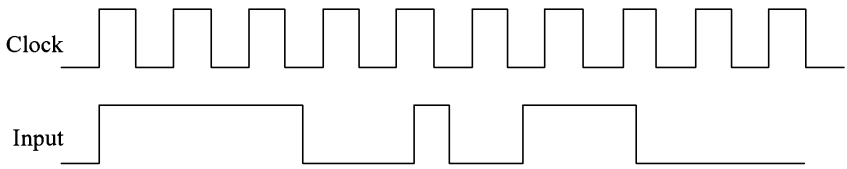


Fig. 7.32 ***Waveforms for Problem 7.8***

7.9 Repeat Problem 7.8 if the FLIP-FLOP is master–slave.

7.10 If the waveforms shown in Fig. 7.33 are applied to the circuit of Fig. 7.31, sketch the output (Q) waveform. Assume *M–S* FLIP-FLOP.

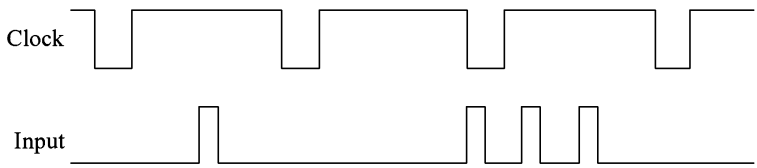


Fig. 7.33 ***Waveforms for Problem 7.10***

7.11 Verify that the circuit of Fig. 7.16 acts as a toggle switch.

7.12 Prepare the truth table for the circuit of Fig. 7.34 and show that it acts as a *T*-type FLIP-FLOP.

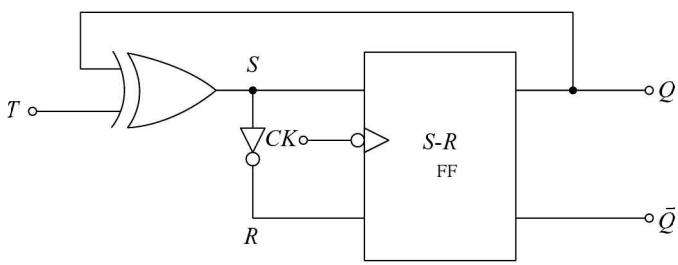


Fig. 7.34 ***Circuit for Problem 7.12***

7.13 If $\bar{Q}$ output of a *D*-type FLIP-FLOP is connected to D input, it acts as a toggle switch. Verify.

7.14 Using the method outlined in Section 7.8, design the following FLIP-FLOPs:
(a) *J–K*
(b) *D*-type
(c) *T*-type

7.15 Using the conversion method outlined in Section 7.8, carry out the following conversions:

(a) *S–R* to *D*
(b) *J–K* to *D*
(c) *D* to *J–K*
(d) *S–R* to *T*
(e) *J–K* to *T*
(f) *T* to *J–K*
(g) *T* to *D*
(h) *D* to *S–R*
(i) *D* to *T*
(j) *T* to *S–R*
(k) *J–K* to *S–R*

7.16 Figure 7.35 shows a positive-edge-triggered *D* `FLIP-FLOP`. Verify its operation.

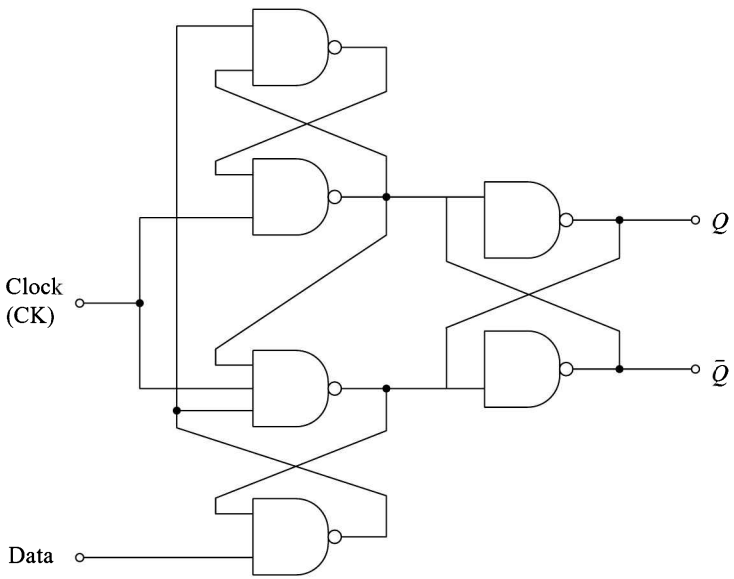


Fig. 7.35 ***Positive-edge-triggered*** *D* `FLIP-FLOP`.

7.17 IC 7411I is a master–slave *J–K* `FLIP-FLOP` with data lock out at the positive edge of the clock (the state changes at the negative edge of the clock).

If the waveform shown in Fig. 7.22*a* is applied at the clock input and the waveform of 7.22*b* is applied at the *J* input, sketch the output (*Q*) waveform when
(a) *K* is connected to logic 0.
(b) *K* is connected to logic 1.

7.18 Verify the operation of the debounce switches shown in Fig. 7.36.

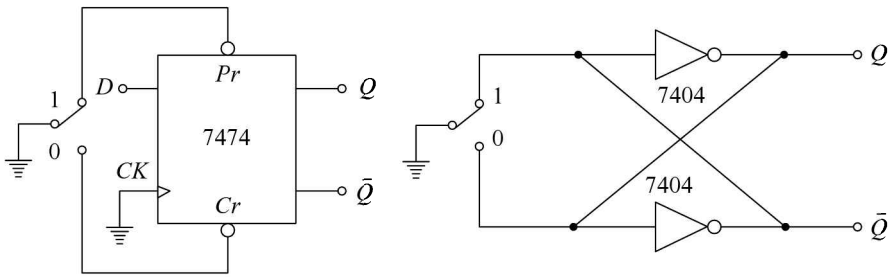


Fig. 7.36 ***Debounce Switches for Problem 7.18***

7.19 Consider the circuit of Fig. 7.37 consisting of positive-edge-triggered FLIP-FLOPs. Δt_1 and Δt_2 are the time delays introduced in the clock (CLOCK SKEW) by buffer devices and the propagation delay of wires. At the rising edge of the clock, new data enters the source FF and the content of source FF enters the destination FF. Show the effect of clock skew ($\Delta t_2 > \Delta t_1$) on the operation of the circuit. Discuss the problems created by clock skew for data transmission.

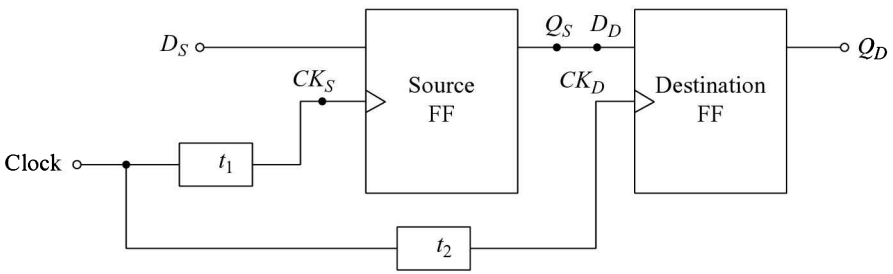


Fig. 7.37 ***Circuit for Problem 7.19***

7.20 A mod-3 counter (resets after every three pulses) is shown in Fig. 7.38. The FLIP-FLOPs used are master–slave *J–K*. Sketch the waveforms of Q_0 and Q_1 when clock pulses are applied and verify its operation. Assume $Q_0 = Q_1 = 0$ initially.

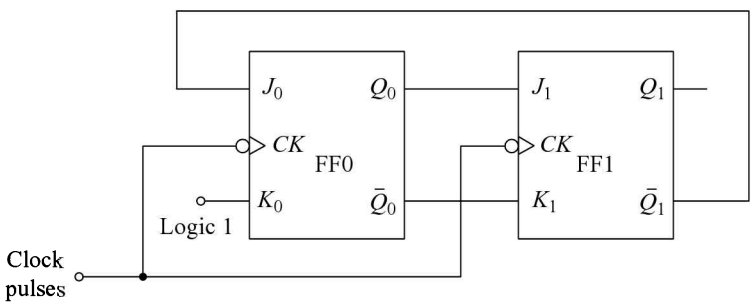


Fig. 7.38 ***A Mod-3 Counter***

7.21 In the circuit shown in Fig. 7.39, the time constant (RC) is very small. Explain the operation of this circuit.

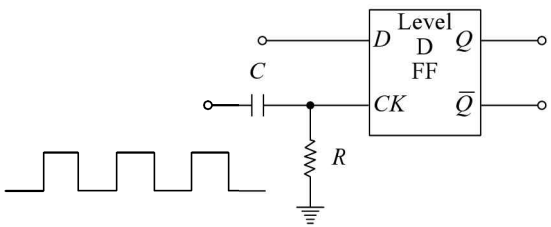


Fig. 7.39 ***Circuit for Problem 7.21***