5 Bistable Multivibrator Circuits

In this chapter, circuits that have two stable states (i.e. off and on) will be studied. These circuits are also commonly known as flip-flops. As they have two stable states (i.e. logic 0 or 1), they are useful to store one bit of digital data, i.e. as memory elements. Several types of flip-flops will be studied before we look at other multivibrators to generate single and train of pulses.

Figure 5.1 shows a general flip-flop symbol. Usually, there are one or two inputs to the flip-flop and the output also has a complement. The inputs are either logic 0 or 1 and commonly known as set (or preset) input when equal to 1 (HIGH state) and reset (or clear) input when equal to 0 (LOW state).

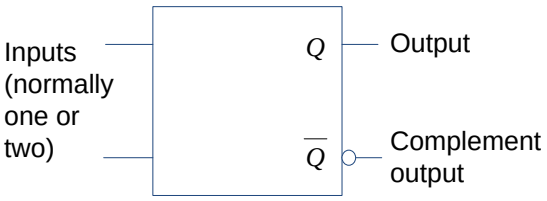


Figure 5.1: General flip-flop symbol.

5.1 S-R flip-flop

S-R flip-flop (also known as set-reset or latch) can be constructed using NOR or NAND gates. Both types of flip-flops are shown in Figure 5.2. The truth table for the S-R flip-flop is shown in Table 5.1. Q^+ here denotes the next state of output Q.

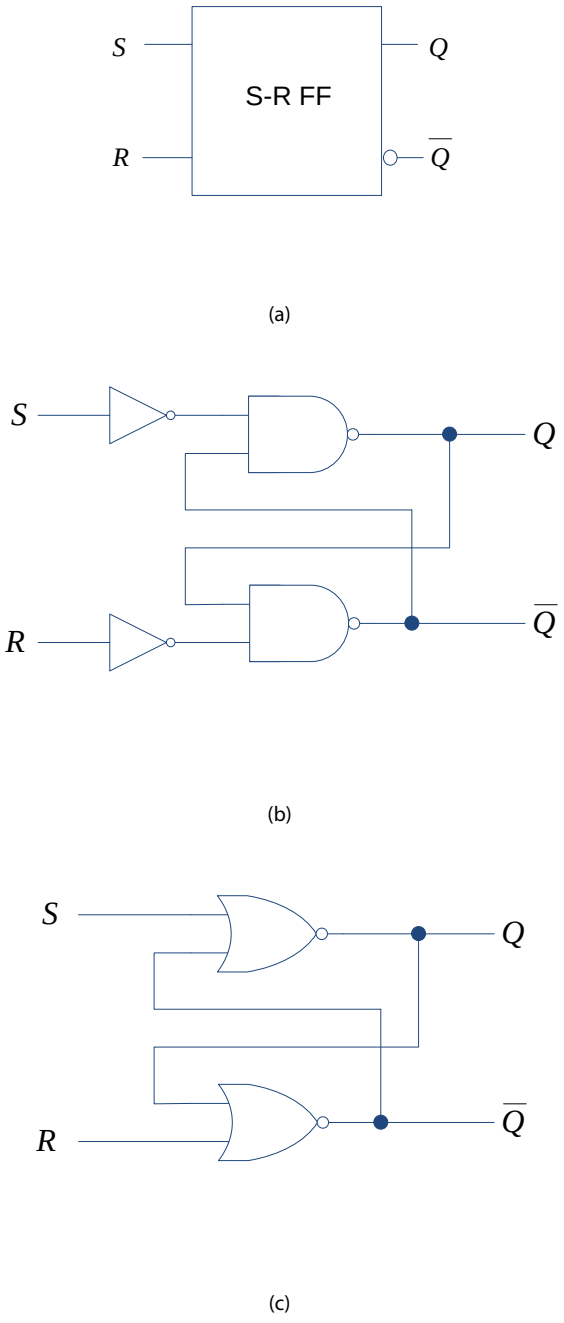


Figure 5.2: S-R flip-flop: (a) general symbol (b) using NAND gates (c) using NOR gates.

Table 5.1: Truth table for S-R flip-flop

S	R	Q^+	
0	0	Q	No change, $Q^+=Q$
1	0	1	Set output $Q^+=1$
0	1	0	Clear output $Q^+=0$
1	1	-	Invalid state

It can be seen that for both NAND and NOR types, there is feedback for the output and complemented output to the inputs. When both S and R inputs are LOW (logic 0), the output of the flip-flop will be the same as its previous state, i.e. no change in the Q state. A HIGH (logic 1) S input to the flip-flop will cause the output Q^+ to change state to HIGH. Similarly, R=1 input will cause the S-R flip-flop's output Q^+=0. It should be obvious that the S input sets the flip-flop to logic 1 while the R input resets the flip-flop to logic 0. S-R flip-flop output is not defined when both inputs are 1, so this situation should be avoided when using the S-R flip-flop. In the above discussion, state of $\overline{Q}$ will be opposite to the state of Q at all times.

5.1.1 S-R flip-flop with Enable input

An enabling input can be used to control the operation of the flip-flops as shown in Figure 5.3. Here the inputs R and S will only have an effect on the output Q^+ if the enable input is 1. When E=1, the NAND gates (in bold) will act as inverters, thereby the circuit behaving exactly like the NAND gate S-R flip-flop in Figure 5.2(b). Table 5.2 gives the truth table values.

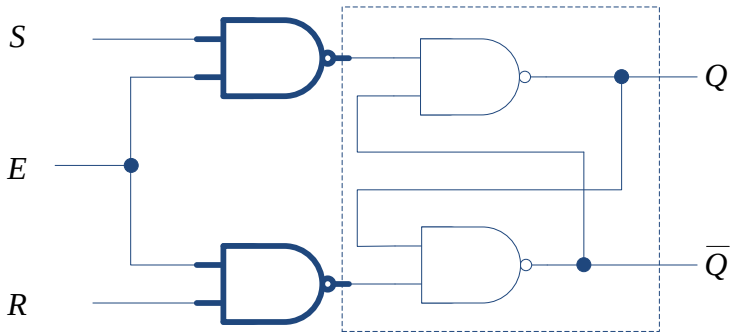


Figure 5.3: S-R flip-flop with Enable input.

Table 5.2: Truth table for S-R flip-flop with Enable input

E	S	R	Q^+	
0	0	0	Q	No change, $Q^+=Q$
0	1	0	Q	No change, $Q^+=Q$
0	0	1	Q	No change, $Q^+=Q$
0	1	1	Q	No change, $Q^+=Q$
1	0	0	Q	No change, $Q^+=Q$
1	1	0	1	Set output $Q^+=1$
1	0	1	0	Clear output $Q^+=0$
1	1	1	-	Invalid state

5.1.2 Clocked S-R flip-flop

Similar to the enable input, there could be a clock (i.e. pulsed) input to the flip-flop. Clocked S-R flip-flop is shown in Figure 5.4 where the edge of the clock (either positive or negative) triggers the change in the flip-flop state. The negative edge of the clock occurs when the clock pulse drops from logic 1 to 0 and is also known as negative going transition (NGT) while the positive going transition (PGT) occurs when the clock pulse goes from logic 0 to 1. An opposite clock edge will not affect the flip-flop output. For example, a negative edge triggered flip-flop will not change state during the positive edge. Table 5.3 shows the truth table for the NGT clocked flip-flop where it can be seen that the flip-flop changes state during the corresponding negative triggering edge of the clock. The PGT clocked flip-flop behaves similarly except that the change (if any) occurs during the positive edge transition of the clock.

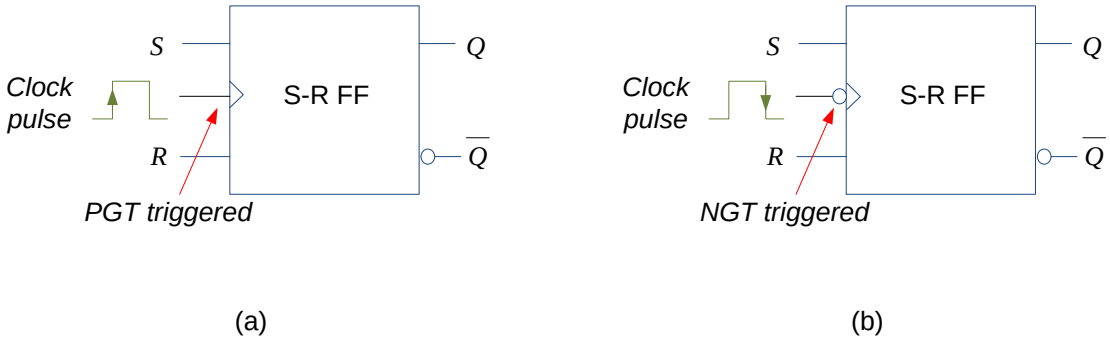


Figure 5.4: Clocked S-R flip-flops: (a) PGT (b) NGT, note the bubble for NGT triggered flip-flop.

Table 5.3: Truth table for NGT clocked S-R flip-flop

Clock	*S*	*R*	Q^+	
↓	0	0	Q	No change, $Q^+=Q$
↓	1	0	1	Set output $Q^+=1$
↓	0	1	0	Clear output $Q^+=0$
↓	1	1	-	Invalid state

A few examples using timing diagrams follow to illustrate the behaviour of clocked S-R flip-flops. Figure 5.5 shows an example on how the timing diagram changes for NGT clocked S-R flip-flop. Any change in the output Q will only occur during NGT (shown by t_1, t_2, …, t_5):

- At time t_1, Q goes to logic 1 as S=1, R=0
- At time t_2, Q remains at logic 1 as S=1, R=0
- At time t_3, Q goes to logic 0 as S=0, R=1
- At time t_4, Q remains at logic 0 as S=0, R=1
- At time t_5, Q goes to logic 1 as S=1, R=0

There won't be any changes during t_{PGT} for negative edge triggered flip-flop.

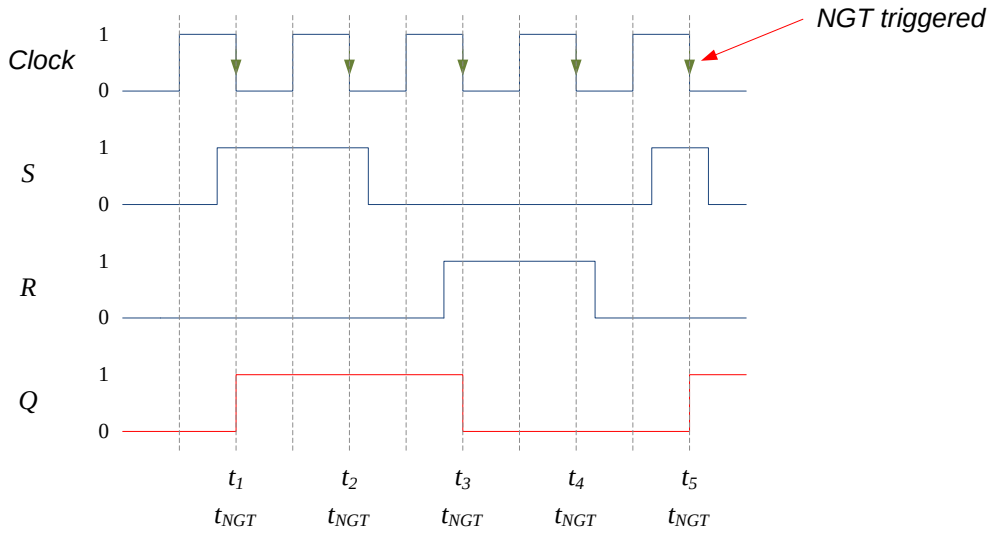


Figure 5.5: Timing diagram for NGT clocked S-R flip-flop example.

A PGT clocked S-R flip-flop timing diagram example is shown in Figure 5.6. Any change in the output Q will only occur during PGT (shown by t_1, t_2, …, t_5):

- At time t_1, Q goes to logic 1 as S=1, R=0
- At time t_2, Q remains at logic 1 as S=0, R=0
- At time t_3, Q goes to logic 0 as S=0, R=1
- At time t_4, Q goes to logic 1 as S=1, R=0
- At time t_5, Q goes to logic 0 as S=0, R=1

There won't be any changes during t_{NGT} for positive edge triggered flip-flop.

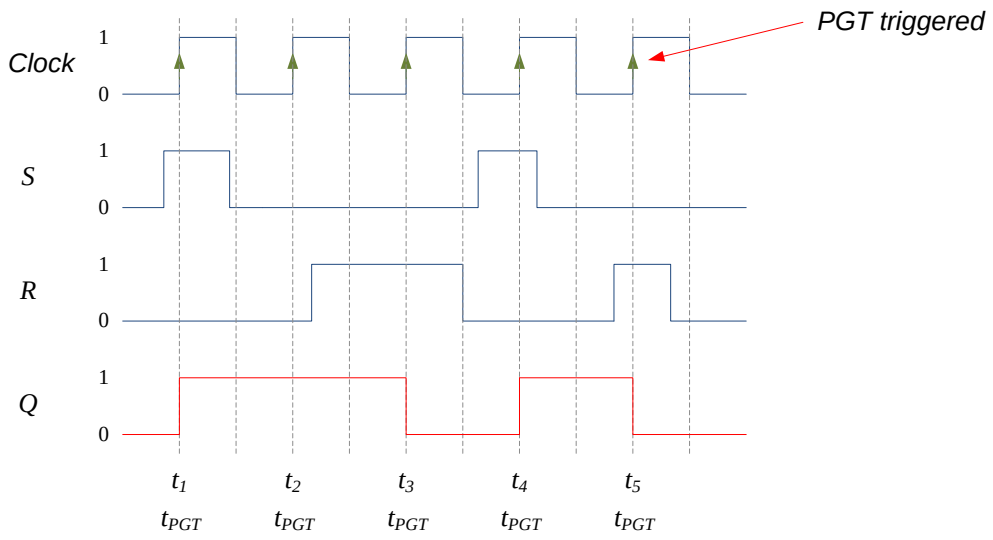


Figure 5.6: Timing diagram for PGT clocked S-R flip-flop example.

5.1.3 Asynchronous flip-flop inputs

The *S* and *R* inputs are known as synchronous inputs as their effects are synchronised to the clock input. Flip-flops can also have asynchronous inputs that can affect the output at any time irrespective of the clock pulse. Figure 5.7 shows the NGT S-R flip-flop symbol with two additional pulse inputs: ($\overline{PRE}$) that sets the output to logic 1 and clear ($\overline{CLR}$) that sets the output to logic 0. Both these inputs are ACTIVE LOW[8] (shown with an overbar, also note the existence of the bubble in the figure), which means that a logic 0 input will affect the flip-flop output rather than logic 1. Asynchronous inputs always take precedence over the *S* and *R* inputs.

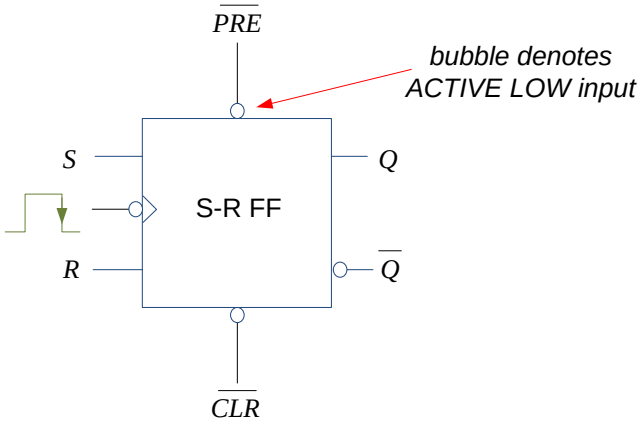


Figure 5.7: NGT S-R flip-flop symbol with asynchronous inputs.

Figure 5.8 illustrates the effect of these asynchronous inputs using a timing diagram. When $\overline{PRE}$ and $\overline{CLR}$ equals logic 1, the flip-flop behaves exactly as an NGT clocked S-R flip-flop. However, when either pulse becomes active (i.e. goes to logic 0), the effect on output *Q* is immediate (i.e. independent of the clock pulse):

- At time t_1, *Q* goes to logic 1 as *S*=1, *R*=0
- At time t_{12}, *Q* goes to logic 0 as $\overline{CLR} = 0$
- At time t_2, *Q* goes to logic 1 as *S*=1, *R*=0
- At time t_3, *Q* goes to logic 0 as *S*=0, *R*=1
- At time t_{34}, *Q* goes to logic 1 as $\overline{PRE} = 0$
- At time t_4, *Q* goes to logic 0 as *S*=0, *R*=1
- At time t_5, *Q* goes to logic 1 as *S*=1, *R*=0

8 *S* and *R* inputs that either sets or resets the flip-flop on logic 1 are examples of ACTIVE HIGH inputs.

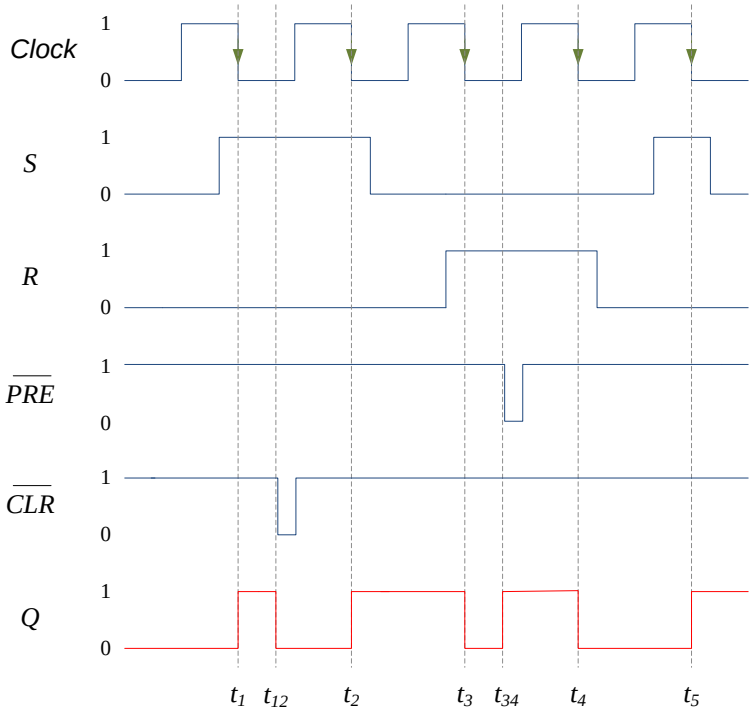


Figure 5.8: NGT S-R flip-flop timing diagram example with asynchronous inputs.

5.2 J-K flip-flop

R-S flip-flop is not very commonly used in digital systems due to the invalid state that can occur when both inputs are logic 1. J-K (named after Jack Kilby) flip-flop overcomes this problem by toggling (i.e. going to opposite state) when inputs J=K=1. Table 5.4 shows the truth table for this flip-flop.

Table 5.4: Truth table for J-K flip-flop

J	K	Q^+	
0	0	Q	No change, $Q^+=Q$
1	0	1	Set output $Q^+=1$
0	1	0	Clear output $Q^+=0$
1	1	$\overline{Q}$	Toggle, $Q^+ = \overline{Q}$

Similar to R-S flip-flop, J-K flip-flop can have enable input, clocked (NGT or PGT) and asynchronous inputs. Figure 5.9 shows the PGT J-K flip-flop symbol.

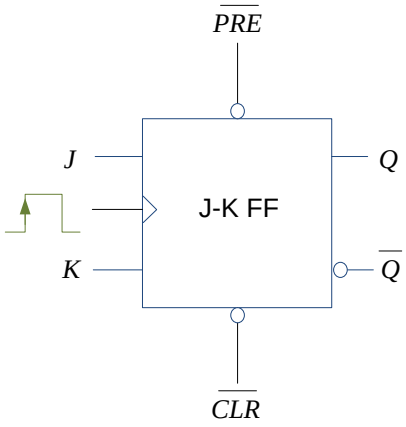


Figure 5.9: PGT J-K flip-flop symbol with asynchronous inputs.

A timing diagram example for J-K flip-flop is given in Figure 5.10. The previous discussions for S-R flip-flop hold for J-K flip-flop except that when J=K=1, the output toggles from its previous state:

- At time t_1, Q goes to logic 1 as J=1, K=0
- At time t_2, Q toggles to logic 0 as J=1, K=1
- At time t_3, Q remains at logic 0 as J=0, K=0
- At time t_4, Q toggles to logic 1 as J=1, K=1
- At time t_5, Q goes to logic 0 as J=0, K=1

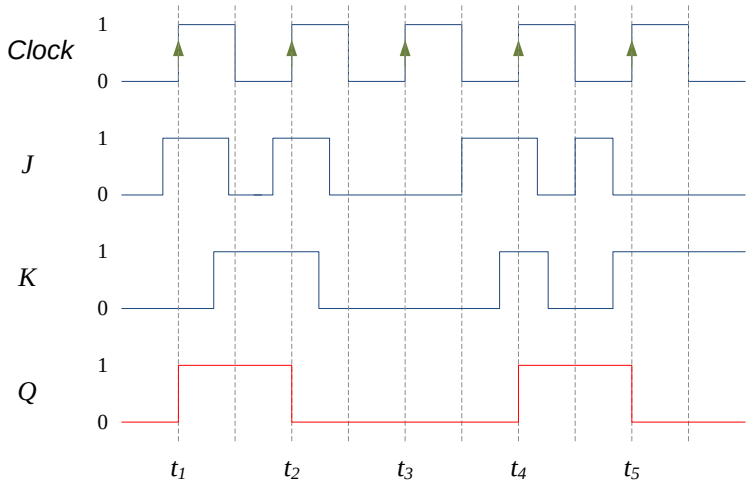


Figure 5.10: PGT J-K flip-flop timing diagram example.

Figure 5.11 gives a timing diagram example of NGT J-K flip-flop with asynchronous inputs:

- At time t_1, Q toggles to logic 1 as J=1, K=1
- At time t_{12}, Q goes to logic 0 as $\overline{CLR} = 0$
- At time t_2, Q goes to logic 1 as J=1, K=0
- At time t_3, Q goes to logic 0 as J=0, K=1
- At time t_{34}, Q goes to logic 1 as $\overline{PRE} = 0$
- At time t_4, Q remains at logic 1 as J=0, K=0
- At time t_5, Q remains at logic 1 as J=1, K=0

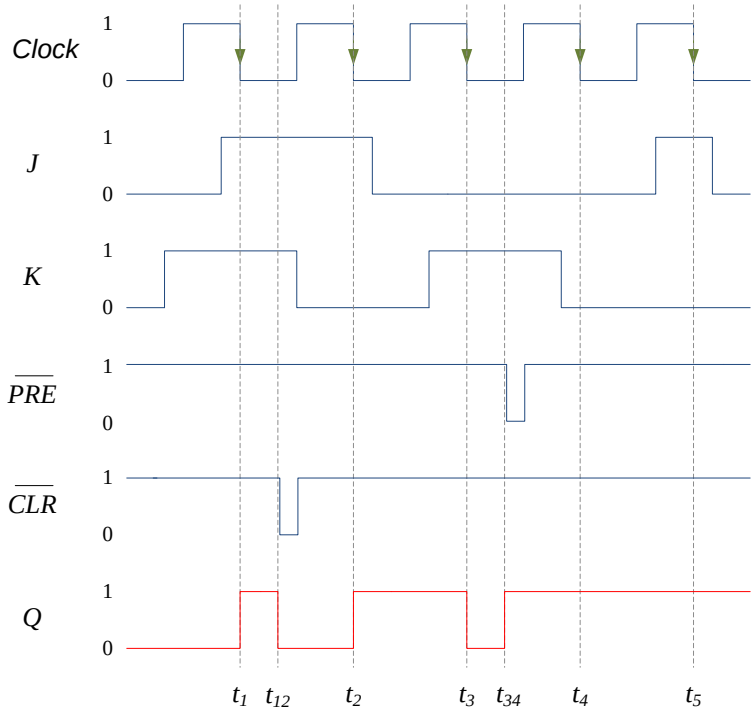


Figure 5.11: NGT J-K flip-flop timing diagram example with asynchronous inputs.

5.2.1 Master-slave flip-flop

As we will see in a later chapter, a sequence of flip-flops are usually connected to each other with a single clock and an example is shown in Figure 5.12. Since there could be a delay in the clock pulse to arrive at FF_2 as compared to FF_1 due to the longer wiring, the output can become unpredictable. To avoid this problem, a master-slave flip-flop can be used where FF_1 is the master and FF_2 is the slave. The inputs to FF_1 are used to determine the output of the master during *CLK*=HIGH and this output is then transferred to the slave when *CLK*=LOW. However, master-slave flip-flops have become obsolete with the design of modern edge-triggered flip-flops that responds with sufficient speed and reliability.

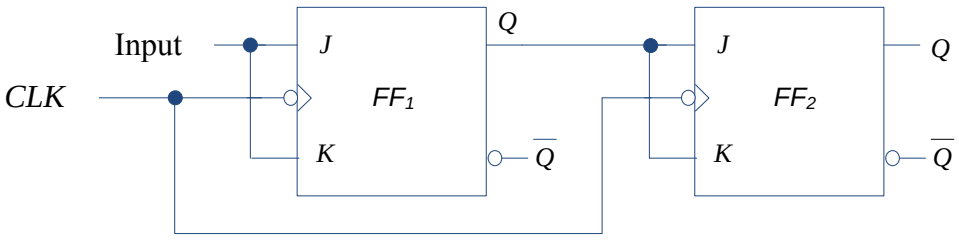


Figure 5.12: Two flip-flops connected with a single clock.

5.3 D flip-flop

D flip-flop is also known as data flip-flop since it can store a single bit of data. The output of the flip-flop Q follows the *single* input D at the respective clock pulses. Figure 5.13 shows the D flip-flop symbol.

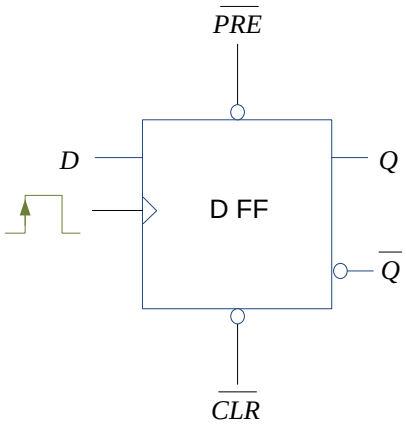


Figure 5.13: PGT D flip-flop general symbol.

Table 5.5 gives the truth table for D flip-flop. The output Q will follow the input D at either NGT or PGT clock depending on whether it is negative or positive edge triggered flip-flop. The D flip-flop can also have asynchronous inputs such as $\overline{PRE}$ and $\overline{CLR}$ that affect the output Q independently of the clock.

Table 5.5: Truth table for D flip-flop

D	Q^+	
0	0	$Q^+=D$
1	1	$Q^+=D$

Figure 5.14 gives an example of the D flip-flop timing diagram:

- At time t_1, Q goes to logic 1 as D=1
- At time t_2, Q goes to logic 0 as D=0
- At time t_{23}, Q goes to logic 1 as $\overline{PRE}=0$
- At time t_3, Q remains at logic 1 as D=1
- At time t_4, Q remains at logic 1 as D=1
- At time t_{45}, Q goes to logic 0 as $\overline{CLR}=0$
- At time t_5, Q remains at logic 0 as D=0

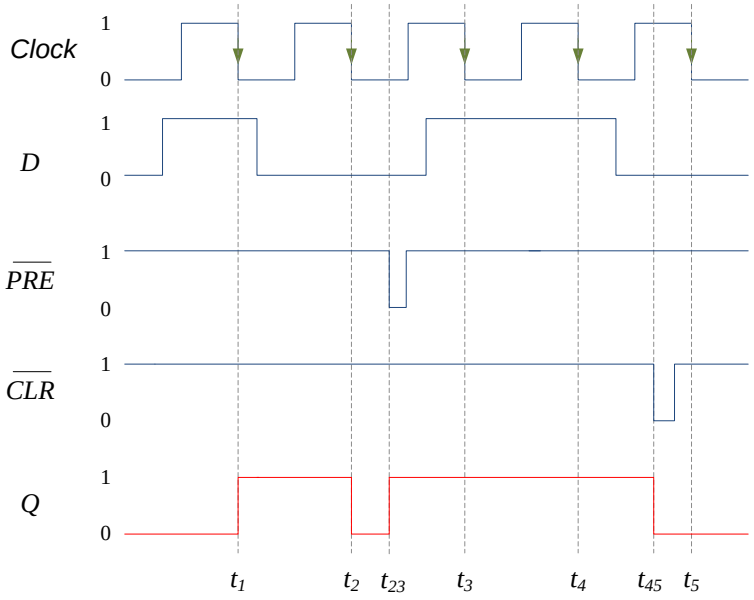


Figure 5.14: NGT D flip-flop timing diagram example.

Figure 5.15 shows how a J-K flip-flop can be used to construct a D flip-flop. When D=1, inputs to J-K flip-flop: $D = J = 1$ and $K = \overline{D} = 0$ and hence, Q = 1. Similarly, when D=0, inputs to J-K flip-flop: $D = J = 0$ and $K = \overline{D} = 1$ and hence, Q = 0. So the output Q follows input D as in the D flip-flop.

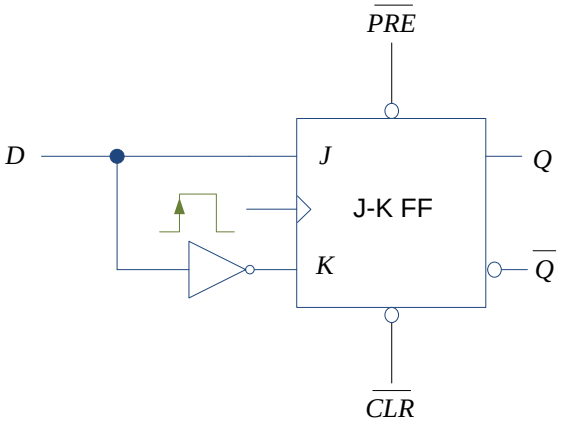


Figure 5.15: PGT D flip-flop constructed using J-K flip-flop.

5.4 T flip-flop

The final flip-flop to be considered in this chapter is T flip-flop. The truth table of the T flip-flop is given in Table 5.6 assuming it is triggered by an NGT clock. The output for T flip-flop toggles at T=1 thereby giving a clock like waveform but with half the frequency as shown by the timing diagram in Figure 5.16. When T=0, the output Q does not change.

Table 5.6: Truth table for NGT T flip-flop

	T	Q^+	
	0	Q	No change, $Q^+ = Q$
	1	$\overline{Q}$	Q^+ toggles, $Q^+ = \overline{Q}$

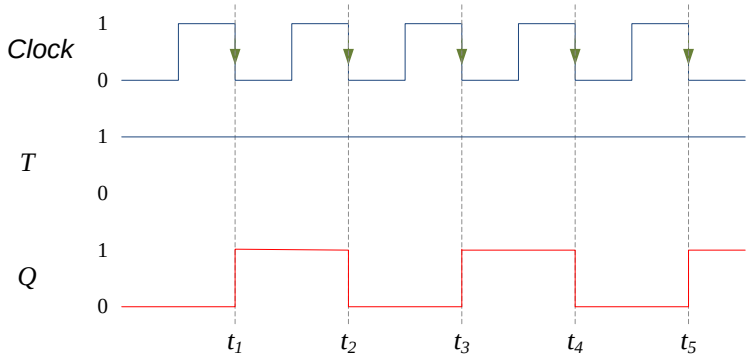


Figure 5.16: NGT T flip-flop where T=1, hence flip-flop operates in toggle mode at each clock trigger.

Figure 5.17 shows the general T flip-flop symbol and also how a J-K flip-flop can be used to construct a T flip-flop by tying J-K inputs together. When J=K=1, the flip-flop output toggles and when J=K=0, the flip-flop output does not change.

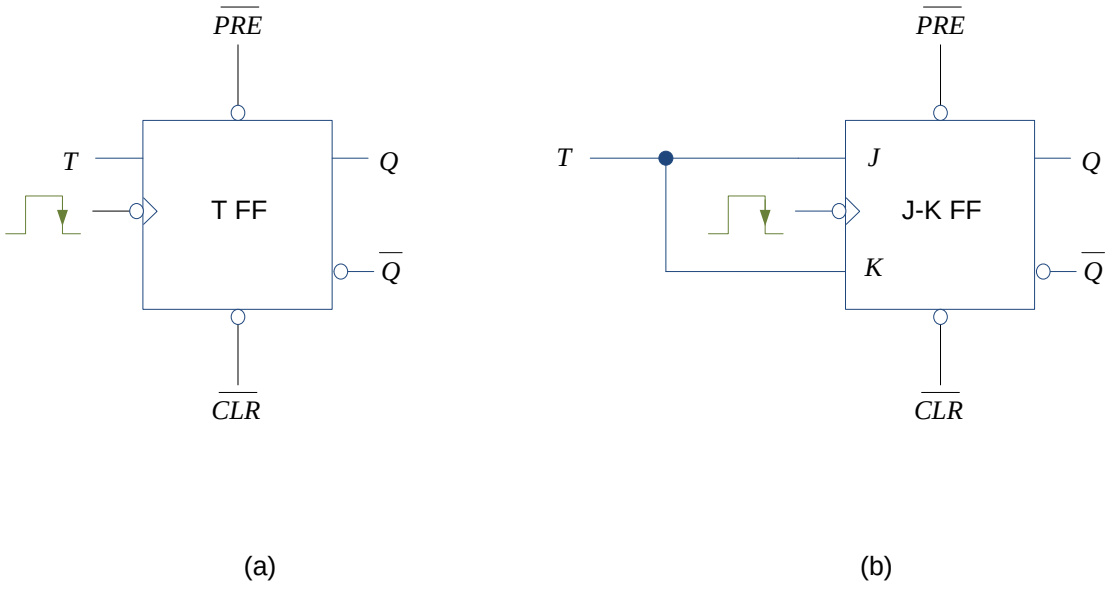


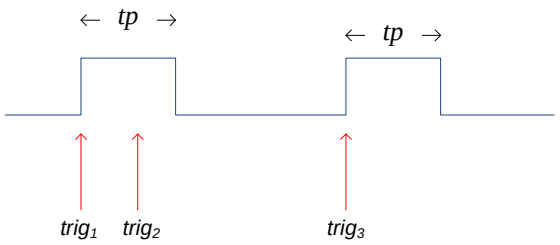
Figure 5.17: NGT T flip-flop: (a) general symbol (b) constructed using J-K flip-flop.

5.5 Monostable and astable multivibrators

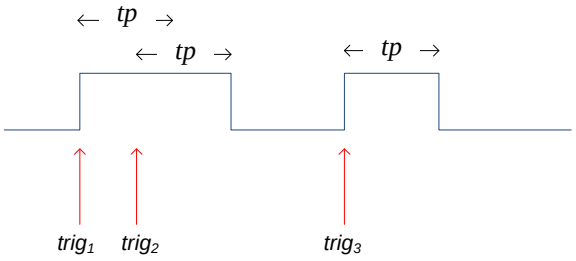
So far, we have considered flip-flops that have two stable states. In this section, we will look at two devices, one that give short trigger pulses and another that gives two states that are free running. Monostable multivibrator is also known as one shot as it has one stable state (normally Q=0) and the other state (normally Q=1) occurs for a specific *tp* duration when triggered. Astable multivibrator does not have a stable state but switches continuously between two states (i.e. Q=0 and Q=1) which results in a train of square (or rectangular) wave pulses at a frequency determined by values of connected resistors and capacitors. Square wave pulses (i.e. with a 50% duty cycle) could be used as clock input.

5.5..1 Monostable multivibrator

Monostable multivibrator could be divided into two types: non-retriggerable and retriggerable. Non-retriggerable monostable multivibrator will ignore any trigger request during a *tp* pulse while the retriggerable one will re-trigger the pulse for another *tp* duration. The effects of both multivibrators are illustrated in the examples given in Figure 5.18. For non-retriggerable monostable multivibrator, $trig_2$ has no effect since it is within the duration of the *tp* pulse triggered by $trig_1$. However, for retriggerable monostable multivibrator, *trig2* has the effect of extending the one shot pulse by *tp* duration.



(a)



(b)

Figure 5.18: Monostable multivibrator: (a) non-retriggerable (b) retriggerable.

5.5.2 Astable multivibrator

Astable multivibrator designed using 555 timer IC is shown in Figure 5.19. It generates rectangular pulses with duration t_A and t_B. Duty cycle is defined as $t_B/(t_A + t_B)$. To generate clock pulses, the duty cycle has to be 50%, i.e. $t_A = t_B$.

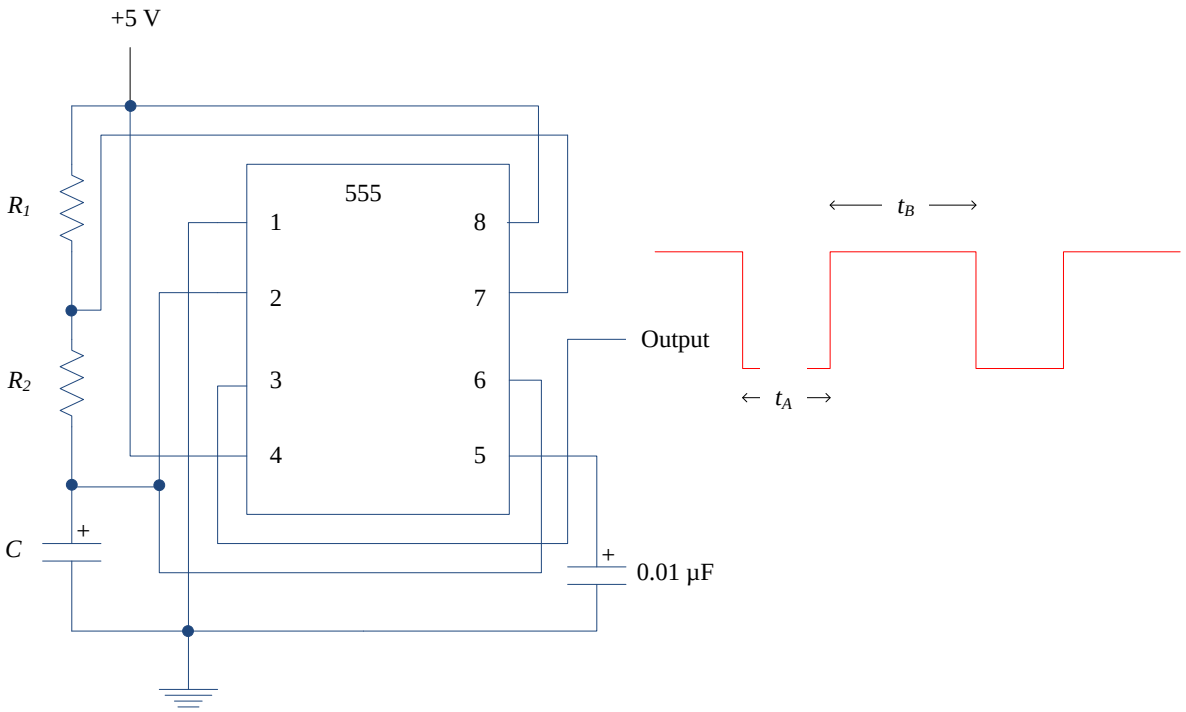


Figure 5.19: Astable multivibrator using 555 timer IC.

The values of the resistors R_1 and R_2 and the capacitor C will affect the durations of t_A and t_B:

$t_A = 0.693\ R_2C$

$t_B = 0.693\ (R_1 + R_2)C$

The frequency of the pulse is given by, $freq = 1/(t_A + t_B)$.

Consider an example where R_1=4.7 kΩ, R_2 = 10.0 kΩ, and C = 100ϱF, we get

$t_A = 0.693\ R_2C$

= 0.693 x (10 kΩ) x 100 μF

= 0.693 x (10000 Ω) x 0.0001 F

= 0.693 s

$t_B = 0.693\ (R_1 + R_2)C$

= 0.693 x (4700 Ω+10000 Ω) x 100 μF

= 1.01871 s

Frequency = $1/(t_B + t_A) = 0.58421$ Hz.