

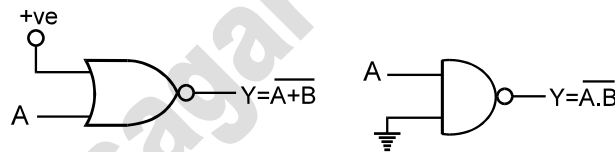
1.31 INTRODUCTION

In this chapter, we have to study three applications of logic gates. The first topic covers the most important logic circuit known as flip-flop. It is the basic building block of semiconductor memory like RAM, ROM, etc. used in computers. The second topic explains the registers. A register is a group of flip-flops used to store a large binary number into its memory. And the third topic is the counters. It is again a combination of flip-flops used to count the number of events in the form of digital pulses. All these applications of flip-flops are very useful to understand the hardware of computer. So student must concentrate on this chapter and prepare it along with first two chapters.

1.32 BASIC TERMINOLOGIES

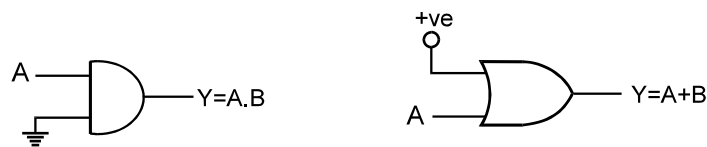
Enable & disable states: Consider the following left figure. One input of NOR gate is permanently connected to positive supply i.e. it is in logic-1 state, keeping other input open. In this case, whatever may be the condition of input A, the output will be always LOW. This condition of NOR gate is called disable condition.

Now for an NAND gate, the condition will be different as shown in right figure. Here one input of NAND gate is permanently grounded i.e. it is in logic-0 state, keeping other input open. Now whatever may be the condition of input A, its output will be always HIGH. This condition of NAND gate is called enable condition. In short, the enable condition means output of the gate remains at logic-1 state irrespective of other input. And disable condition means, remains at logic-0 state irrespective of other input. Note that this definitions are applicable to NAND gate and NOR gates only. For AND & OR it is exactly opposite.



Now refer following two figures. When one input of AND gate is at logic-0, then whatever may be input condition of the other input, its output will be always at logic-0 state. This is called the disabled condition of AND gate.

For an OR gate, the conditions again change. When one input OR gate is at logic-1, then whatever may be the input condition of other input, its output will be always at logic-1 state. This is called as the enabled condition of OR gate. Thus the enable and disable are the relative terms and should be used to a particular gate according to the rules given above.



Floating input: When an input terminal of TTL logic circuit is in floating state i.e. it is connected to nothing, just floating in air, then it is treated as HIGH state or logic-1 state of input. This happens because the emitter current of input transistor inside the circuit is zero. Hence, input is as good as in logic-1 state.

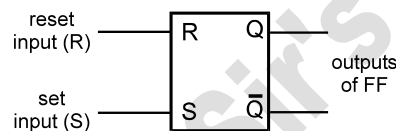
The 'x' state: When the state of input/output of a logic circuit is not known or when the state may be either at logic-1 or logic-0 state, such state is denoted by 'x'.

ANDing & ORing: Suppose there are two input terminals A & B of a logic circuit. Now when they are wired together such that the output $Y = A.B$, then it is said that the input signals A & B are connected in ANDing style. Similarly if $Y = A + B$, it is said that the input signals A & B are connected in ORing style. These terms are also used in Past tense as ANDed and ORed etc.

1.33 FLIP FLOP

A flip-flop has two stable states at its output. *We can consider a flip-flop just like a simple switch.* The switch has two states: ON state and OFF state. Both these states of the switch are stable. If we put the switch in ON state, it remains in that state for “infinite time” unless we change the state. Thus, we say that the ON and OFF states of a switch, remain stable for “infinite time”, unless we change one state to other. We can also consider a flip-flop like a door lock. Hence, flip-flop is popularly called as latch means a lock.

The flip-flop also works just like a switch. Hence, is called as *bistable* multivibrator, means having two stable states. A simple logic circuit of a flip-flop is given below –



It has two inputs known as *RESET* input and *SET* input. It has two outputs known as Q and \bar{Q} . Thus, it can be either *SET* or *RESET* by applying particular input signal at the inputs. The working of above logic circuit can be understood using the truth table.

When we want to *SET* the flip-flop, make $S = 1$ and $R = 0$. Then the outputs will be $Q = 1$ and $\bar{Q} = 0$. Similarly to *RESET* the flip-flop, make $S = 0$ and $R = 1$. Then the outputs will be $Q = 0$ and $\bar{Q} = 1$. Remember that the *SET* or *RESET* states are stable states of a flip-flop. That is if a flip-flop is in *SET* state, it remains in that state for “infinite time”, unless we *RESET* it. If it is *RESET*, it remains in that state for “infinite time”, unless we *SET* it. This property of the flip-flop to remain either in *SET* state or *RESET* state, is called as MEMORY OF THE FLIP-FLOP.

There are different types of flip-flop circuits in this chapter, as given below –

RS flip-flop: Which has two stable states, either *SET* or *RESET* state.

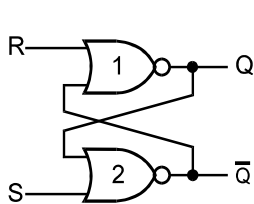
D Flip-flop: This flip-flop has only one input with two stable states, either *SET* or *RESET* state.

JK Flip-flop: It has two stable states, either *SET* or *RESET* state, along with one more state known as the *TOGGLE* state. This third state is very useful in counter circuit applications.

Clock signal: Some flip-flops work on clock signal. Clock is a square wave used in memory applications.

1.34 RS FLIP-FLOP USING NOR GATE

Following diagram shows the circuit of RS flip-flop using NOR gates. The two NOR gates are used such that the output of one is connected to the input of other. Also, one input of each gate is taken out as *SET* and *RESET*. The two outputs are Q and \bar{Q} . Due to cross connections between inputs and outputs, flip-flop works as a latch.



R	S	Q	\bar{Q}
0	0	No Change state	
0	1	1	0
1	0	0	1
1	1	Forbidden state	

Remember

The output Q follows input S and output \bar{Q} follows input R .

Working: The circuit has four possible states as given below –

When $R = S = 0$, the flip-flop does not change its output state. It remains in same state known as NO CHANGE STATE.

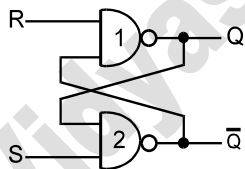
When $R = 0$ and $S = 1$, the flip-flop is now *SET* and remains locked.

When $R = 1$ and $S = 0$, the flip-flop is now *RESET* and remains locked.

When $R = S = 1$, both NOR gates are disabled simultaneously. So we cannot determine output conditions. Hence, it is called FORBIDDEN STATE. It is also called as the RACE state.

1.35 RS FLIP-FLOP USING NAND GATE

Following diagram shows the circuit of RS flip-flop using NAND gates. The two NAND gates are used such that the output of one is connected to the input of other. Also, one input of each gate is taken out as *SET* and *RESET*. The two outputs are Q and \bar{Q} . Due to cross connections between inputs and outputs, the flip-flop works as a latch.



R	S	Q	\bar{Q}
0	0	Forbidden state	
0	1	1	0
1	0	0	1
1	1	No Change state	

Working: The circuit has four possible states as given below –

When $R = S = 0$, both NAND gates are enabled simultaneously. So we cannot determine output conditions. Hence, it is called FORBIDDEN STATE. It is also called as the RACE state.

When $R = 0$ and $S = 1$, the flip-flop is now *SET* and remains locked.

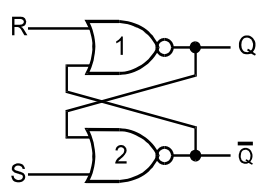
When $R = 1$ and $S = 0$, the flip-flop is now *RESET* and remains locked.

When $R = S = 1$, the flip-flop does not change its output state. It remains in same state known as NO CHANGE STATE.

Note: The RS flip-flop using NAND gates is NEVER used as memory storage circuit, because its first state is forbidden state.

1.36 CONCEPT OF 1-BIT MEMORY CELL

We know that a flip-flop has two stable states. When it is *SET* it remains in that state for infinite time, unless we change the state. When it is *RESET* it remains in that state for infinite time, unless we change the state. So this property of the flip-flop to maintain a particular state for indefinite time is called as memory. Hence, a flip-flop is called as memory unit.

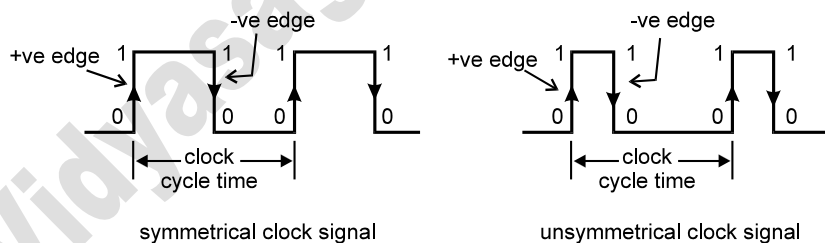


R	S	Q	\bar{Q}
0	0	This state is not used as memory	
0	1	1	0
1	0	0	1
1	1	This state is not used as memory	

When a flip-flop is *SET* its output $Q = 1$ and when it is *RESET* its output $Q = 0$. So it is said that when the flip-flop is *SET* a '1' is stored in the memory of the flip-flop. Similarly, when the flip-flop is *RESET* a '0' is stored in the memory of the flip-flop. Thus, a flip-flop can be used as memory unit. It can store a single binary digit i.e. a bit in the form of memory.

1.37 CONCEPT OF CLOCK

A clock is a periodic waveform, particularly a square wave. It is used to *synchronize digital circuits*. They are of two types: symmetrical and asymmetrical. Both of them have large number of applications in digital electronics. The time required for one complete cycle of clock is called *cycle time*. A clock has two sharp edges. Edge means the instance of the wave at which it changes its voltage from low to high or high to low. Thus there are two edges in a clock signal, they are: the leading edge or the positive edge and falling edge or the negative edge as shown in following figure. The clock signal is connected to flip-flop logic circuit. So the state of flip-flop changes only at positive or negative edge of the clock.



1.38 CONCEPT OF POSITIVE & NEGATIVE LOGIC

Positive logic: in a digital circuit, when logic-1 state is treated as more positive *than* logic-0 state, it is called positive logic.

Negative logic: in a digital circuit, when logic-1 state is treated as more negative *than* logic-0 state, it is called negative logic. Both logics are useful to convert AND gate function into OR gate function or reverse.

This can be explained with examples as follows –

Example: Suppose there are two voltage levels in a logic circuit, 2V and 5V. Then in positive logic, logic-0 means 5V and logic-1 means 2V. Similarly, in negative logic system, logic-1 means 2V and logic-0 means 5V.

Example: Suppose there are two voltage levels, 0V and -5V. Then in positive logic, logic-1 means 0V and logic-0 means -5V. Similarly, in negative logic system, logic-1 means -5V and logic-0 means 0V.

1.39 CONCEPT OF CLOCKING A FLIP-FLOP

The flip-flops circuits which we have studied are known as theoretical logic circuits, because they transmit the input signals to the output without any control. But in practical applications like in computer system, control input is required to synchronize the working of flip-flop with the main system. All the flip-flop circuits used as memory in a computer are controlled by a single “clock” source to synchronize with the system.

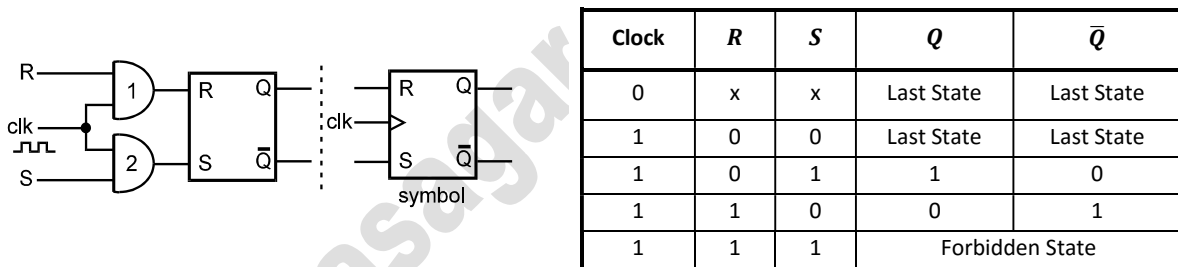
The simple RS flip-flop using NOR gates has some internal drawbacks. This can be understood as follows –

- When $R = 0$ and $S = 1$, the flip-flop is *SET* and a ‘1’ is stored in its memory.
- Similarly when $R = 1$ and $S = 0$, a ‘1’ is stored in its memory.

This circuit of RS flip-flop transmits the input signal to the output side without any control. And in practical circuits controlling input is necessary in a flip-flop, so that the flip-flop can be synchronized with the main system, in which the flip-flops are used. The modern computer systems have multitasking environment, in which such synchronization is very essential to increase the operating speed.

1.39.1 CLOCKED RS FLIP-FLOP

It is also known as controlled RS flip-flop, because it uses controlling signal in its logic circuit. Such circuit is shown below. The circuit has three inputs *RESET*, *SET* and *clk*. The symbol of the RS flip-flop is shown in right-side figure, in the following diagram.



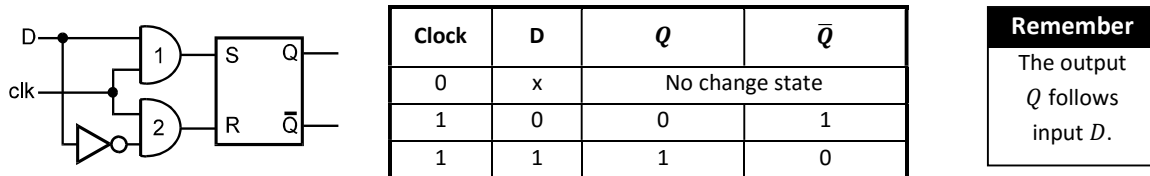
Working: The working of the circuit is given below –

- When $clk = 0$ and $R = S = x$, both AND gates are disabled and the output of the circuit remains in “no change state”.
- When $clk = 1$ and $R = S = 0$, both AND gates are still disabled and the output is in “no change state”.
- When $clk = 1$, $R = 0$ and $S = 1$, output of AND gate-2 become ‘1’ but output of AND gate-1 remains ‘0’.
So flip-flop is *SET* and $Q = 1$, $\bar{Q} = 0$.
- When $clk = 1$, $R = 1$ and $S = 0$, output of AND gate-1 become ‘1’ but output of AND gate-2 remains ‘0’.
So flip-flop is *RESET* and $Q = 0$, $\bar{Q} = 1$.
- Finally, when $Clk = R = S = 1$, the output of the circuit goes in “forbidden state Thus we say that the output Q follows S and \bar{Q} follows R .

Note: The above circuit shows that when $clk = 0$, the circuit does not produce any action and remains in “no change” state. However, when $clk = 1$, then only the circuit responds normally and works as a flip-flop. Thus, we have got a good control over the working of the circuit and we can now store data into the memory of the flip-flop effectively.

1.39.2 CLOCKED D FLIP-FLOP

The clocked RS flip-flop discussed above has two major disadvantages. Firstly it has three input terminals. So to operate it, we require connecting three different signals simultaneously. Secondly the clocked RS flip-flop has forbidden gap which is useless. All these problems are overcome by using clocked D flip-flop logic circuit, as given below.



Working: The working of the circuit is given below –

When $clk = 0$ and $D = x$, both AND gates are disabled and the output of the circuit remains in “no change state”.

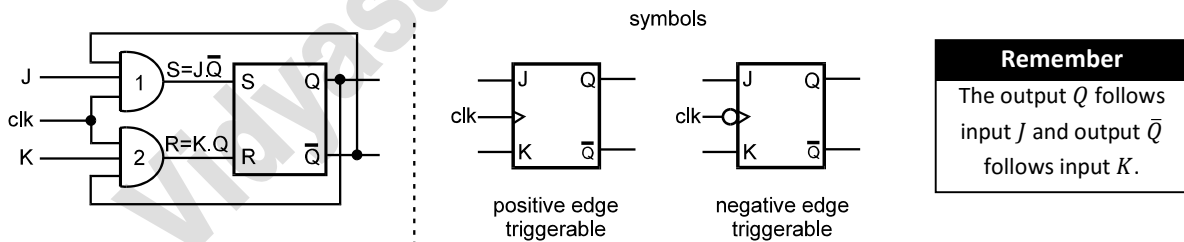
When $clk = 1$ and $D = 0$, the output of AND gate-2 is ‘1’ and output of AND gate-1 is ‘0’. So $R = 1, S = 0$. So $Q = 0$ and flip-flop is *RESET*.

When $clk = 1$ and $D = 1$, the output of AND gate-1 is ‘1’ and output of AND gate-2 is ‘0’. So $R = 0, S = 1$. So $Q = 1$ and flip-flop is *SET*.

Thus we say that the output Q follows the input condition at D .

1.40 JK FLIP-FLOP

It is a special type of flip-flop. It produces one new state at its output, known as the “toggle state”. The toggling of the flip-flop means alternately changing the output condition at every edge of the clock. This state of the flip-flop is very useful in some applications like counters. We shall study the applications of this state in the topic of counters. The circuit of JK flip-flop is given below –



Working: The working of the circuit is given below –

When $clk = 0$ and $J = K = x$, both AND gates are disabled and the output of the circuit remains in “no change state”.

When $clk = 1$ and $J = K = 0$, the AND gates are still disabled and remains in “no change state”

When $clk = 1$ and $J = 0, K = 1$, AND gate-2 is enabled and $Q = 0, \bar{Q} = 1$. So flip-flop is *RESET*.

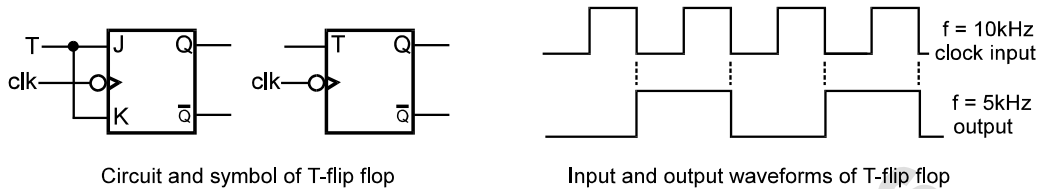
When $clk = 1$ and $J = 1, K = 0$, AND gate-1 is enabled and $Q = 1, \bar{Q} = 0$. So flip-flop is *SET*.

Now when $clk = J = K = 1$, the flip-flop goes in “toggle state” and now at every positive edge of the clock, its output “toggles” i.e. the both the outputs alternately change their states. This condition of the JK flip-flop is called as “Toggle state”. There are two types of JK flip-flop: positive & negative edge triggerable.

1.40.1 APPLICATIONS OF JK FLIP-FLOP: THE T-FLIP-FLOP

The toggle state of JK flip-flop is very useful in some applications. It is used to divide the clock frequency to half. The working and applications of this new state are given below.

The T-flip-flop got its name because it works like a toggle flip-flop only. Its input terminals J and K are connected together to logic-1 level i.e. they are connected to positive supply of the circuit. Its circuit diagram is given below –



Definition: When the input terminals J and K are permanently connected to logic-1 level i.e. connected to positive supply of the circuit, such logic circuit is called as Toggle flip-flop or T-flip-flop.

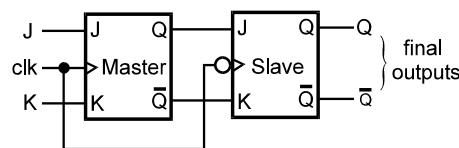
Working: The circuit divides the clock frequency into half. Hence, the T-flip-flop is also called as divide-by-two flip-flop. As shown in above circuit, the flip-flop is negative edge triggerable flip-flop, as a bubble is connected at its clock input terminal. So it alternately changes its output at every negative edge of the clock. So the frequency of the clock signal is divided into half, as shown in the above waveforms. If the frequency of the clock is 10kHz then the frequency at output Q becomes 5kHz.

If number of such T-flip-flops is connected one after another, then we can divide the clock frequency as per our requirements. This application is very useful in the counter circuits.

1.41 JK MASTER-SLAVE FLIP-FLOP

Definition: It is a modified version of JK flip-flop. When output of positive edge triggerable JK flip-flop is connected to the inputs of negative edge triggerable JK flip-flop, master slave flip-flop is produced.

Working: The logic circuit of M/S flip-flop is given below. In this circuit, two JK flip-flops are used, first flip-flop is called *MASTER* flip-flop and the second is called *SLAVE* flip-flop.



The outputs of *MASTER* are connected to the respective inputs of *SLAVE* flip-flop. *MASTER* flip-flop is positive edge triggerable, but *SLAVE* is negative edge triggerable. The clock input terminals of both the flip-flops are connected to a single clock source.

Now *MASTER* changes its output at every positive edge of the clock signal. So its output is copied by the *SLAVE* during negative edge of the clock. So we say that *SLAVE* copies the output states of the *MASTER* and produces same output, but at negative edge of the clock.

1.42 INTRODUCTION TO REGISTERS

Definition: Register is defined as a group of flip-flops arranged in a particular order to store or rotate the data in it. In a register number of flip-flops is connected together such that, when a binary number is entered in the circuit, it can shift in the register (*from one flip-flop to next*) bit-by-bit and shifts out of the register bit-by-bit. The register, which performs the process of data shifting either from MSB to LSB or LSB to MSB is called shift registers. Registers are the basic building blocks of semiconductor memory of the computer like RAM, ROM, etc. The registers can store data in four different styles –

SISO: When data moves in the register, bit-by-bit, it is called Serial In Serial Out register system.

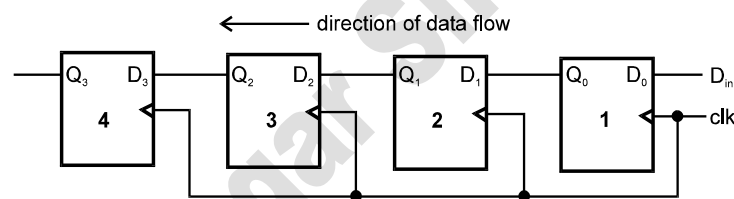
SIPO: When data moves in first register bit-by-bit and then in parallel form, into second register, then such system is called as Serial In Parallel Out register system.

PISO: When data moves in first register in parallel form and then in bit-by-bit style into second register, then such system is called as Parallel In Serial Out register system.

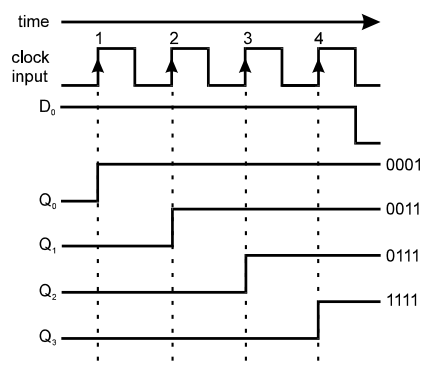
PIPO: When data moves in register, all in parallel form, it is Parallel In Parallel Out register system.

1.42.1 LEFT SHIFT REGISTER

Left shift register means the data in it shifts from *LSB* to *MSB*. In this circuit, four D flip-flops are used as shown in the following circuit diagram. All flip-flops are positive edge triggered type and the output Q_0 of first flip-flop is connected to the input D_1 of second flip-flop and so on. The D_0 input terminal of flip-flop-1 is known as data input D_{in} , where the storage data is connected.



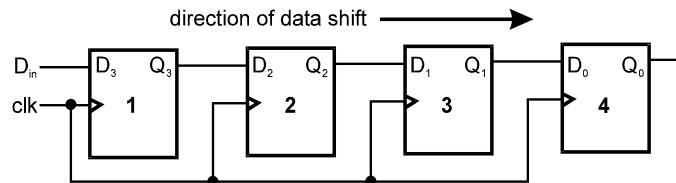
Working: Initially assume that all flip-flops are empty, i.e. $Q_3Q_2Q_1Q_0 = 0000$. Now clock pulses are applied to the circuit, keeping $D_{in} = 1$. Then at first positive edge of clock, flip-flop-1 will change its output state and we get $Q_3Q_2Q_1Q_0 = 0001$. At the second positive edge, the '1' stored in flip-flop-1 will shift to flip-flop-2 and a new '1' will be stored in flip-flop-1. Now the outputs of the circuit will be $Q_3Q_2Q_1Q_0 = 0011$. At third positive edge we get $Q_3Q_2Q_1Q_0 = 0111$. Lastly, the outputs will be $Q_3Q_2Q_1Q_0 = 1111$. In this way, bit-by-bit the data from the source D_{in} will shift into the register memory and will be stored in sequence. Since the data is shifting from *LSB* to *MSB*, this register is called as left shift register. The output waveforms and truth table of the circuit are given below.



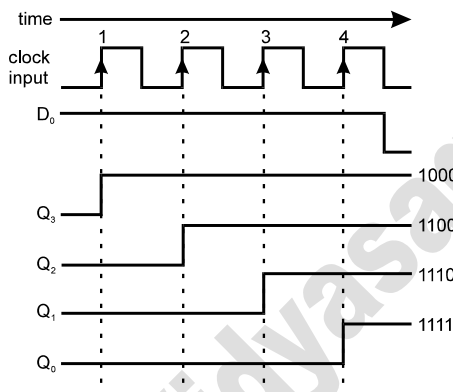
Clock	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1 st	0	0	0	1
2 nd	0	0	1	1
3 rd	0	1	1	1
4 th	1	1	1	1

1.42.2 RIGHT SHIFT REGISTER

Right shift register means the data in it shifts from *MSB* to *LSB*. In this circuit, four D flip-flops are used as shown in the following circuit diagram. All flip-flops are positive edge triggered type and connected to a single clock source. The output Q_3 of first flip-flop is connected to the input D_2 of second flip-flop and so on. The D_3 input terminal of flip-flop-1 is known as data input D_{in} , where the storage data is connected.



Working: Initially assume that all flip-flops are empty, i.e. $Q_3Q_2Q_1Q_0 = 0000$. Now clock pulses are applied to the circuit, keeping $D_{in} = 1$. Then at first positive edge of clock, flip-flop-1 will change its output state and we get $Q_3Q_2Q_1Q_0 = 1000$. At the second positive edge, the '1' stored in flip-flop-1 will shift to flip-flop-2 and a new '1' will be stored in flip-flop-1. Now the outputs of the circuit will be $Q_3Q_2Q_1Q_0 = 1100$. At third positive edge we get $Q_3Q_2Q_1Q_0 = 1110$. Lastly, the outputs will be $Q_3Q_2Q_1Q_0 = 1111$. In this way, bit-by-bit the data from the source D_{in} will shift into the register memory and will be stored in sequence. Since the data is shifting from *MSB* to *LSB*, this register is called as right shift register. The output waveforms and truth table of the circuit are given below.



Clock	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1 st	1	0	0	0
2 nd	1	1	0	0
3 rd	1	1	1	0
4 th	1	1	1	1

1.43 INTRODUCTION TO COUNTERS

A counter is a combination of flip-flops that counts the number of clock pulses connected to it. It produces the counting result in the form of equivalent binary output in BCD sequence. It has M number of output states. The value of M is called as the modulus or *MOD* of the counter.

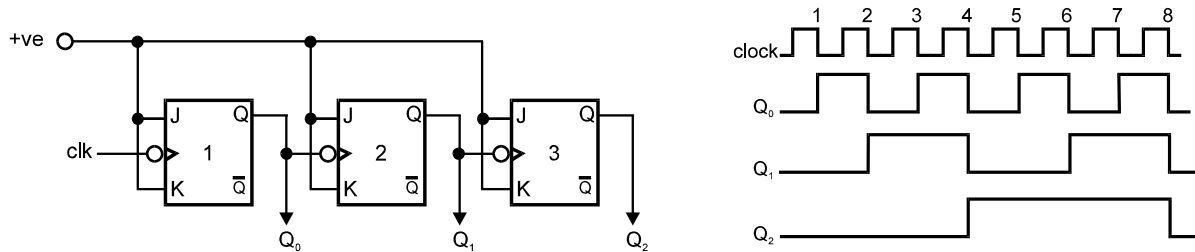
$$M = 2^n \quad \dots \text{ where, } n = \text{number of flip-flop used in counter circuit}$$

Example: For *MOD* 16 counter, $2^n = 2^4 = 16$. $\therefore n = 4$. In another example, if a counter has 7 flip-flops in circuit, then its modulus will be *MOD* 128.

Note: There are two main types of counter circuits. They are synchronous counter and asynchronous counter. In former counter, the source of clock is common to all the flip-flops. In later counter the clock source is connected to first flip-flop only. So the speed of asynchronous counter is less.

1.43.1 ASYNCHRONOUS COUNTER (MOD 8)

The circuit diagram of 3-bit, MOD 8 counter circuit using JK flip-flops is given below. All these flip-flops are negative edge triggerable and in toggle mode. The flip-flops are cascaded in same sequence. So we get, binary count at output of the circuit as $Q_2Q_1Q_0$.



Working: Initially assume that the outputs $Q_2Q_1Q_0 = 000$. When first negative edge arrives, the output of flip-flop-1 becomes logic-1 and $Q_2Q_1Q_0 = 001$. Now at second negative edge, the output of flip-flop-1 changes from '1' to '0'. This is the negative edge for the clock input of flip-flop-2. So the output of flip-flop-2 becomes logic-1. Thus, the outputs will be $Q_2Q_1Q_0 = 010$.

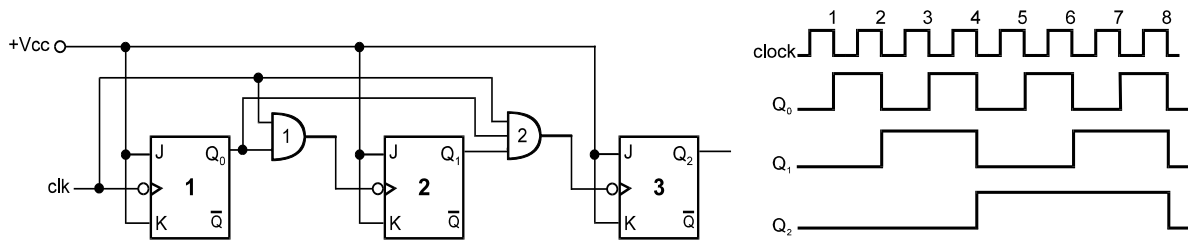
When the third clock pulse arrives, the output of flip-flop-1 again changes from '0' to '1'. This is the positive edge for the clock input of flip-flop-2. Therefore, its output remains unchanged and we get $Q_2Q_1Q_0 = 011$. In the same way, the counter counts the clock pulses from 1 to 8 i.e. the outputs will change from 000 to 111. However, the operating speed of the circuit is low, because the clock inputs of all the flip-flops are connected one after another.

Truth table: The following truth table of the circuit shows the output conditions at every negative edge of the clock. The counter counts from 000 to 111 and repeats again and again.

Clock	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

1.43.2 SYNCHRONOUS COUNTER (MOD 8)

In synchronous counter, all the flip-flops are triggered simultaneously through AND gates i.e. the flip-flops are synchronized. Hence, the circuit is called synchronous counter. The circuit diagram of 3-bit, MOD 8 synchronous counter circuit using JK flip-flops is given below. The flip-flops are negative edge triggerable and in toggle mode. The flip-flops are cascaded in same sequence so we get binary count at output of the circuit as $Q_2Q_1Q_0$. Two AND gates are used which drive the clock input of flip-flop-2 and flip-flop-3 by a single clock source.



The flip-flop-1 changes its output at every negative edge of clock pulse. Hence, the frequency at Q_0 is $\frac{1}{2}$ of clock frequency. Now the output of AND gate-1 goes HIGH when $Q_0 = clk = 1$. Therefore, flip-flop-2 changes its output Q_1 at every alternate clock pulse.

Hence, the frequency at Q_1 is $\frac{1}{4}$ of clock frequency. The output of AND gate-2 goes HIGH, when $clk = Q_0 = Q_1 = 1$. Therefore, flip-flop-3 changes its output Q_2 at every alternate state of Q_0 . Hence, the frequency at Q_0 is $\frac{1}{8}$ of clock frequency or $\frac{1}{4}$ of the frequency at Q_0 or $\frac{1}{2}$ of the frequency at Q_1 .

When first negative edge arrives, flip-flop-1 changes its output and we get $Q_2Q_1Q_0 = 001$. When second clock pulse arrives at its positive edge the AND gate-1 is enabled and its output becomes HIGH. When the negative edge of second clock pulse appears, flip-flop-1 and flip-flop-2 both change their outputs and we get $Q_2Q_1Q_0 = 010$. In this way, the counter counts from 000 to 111 in BCD sequence.

Truth table: The following truth table of the circuit shows the output conditions at every negative edge of the clock. The counter counts from 000 to 111 and repeats again and again.

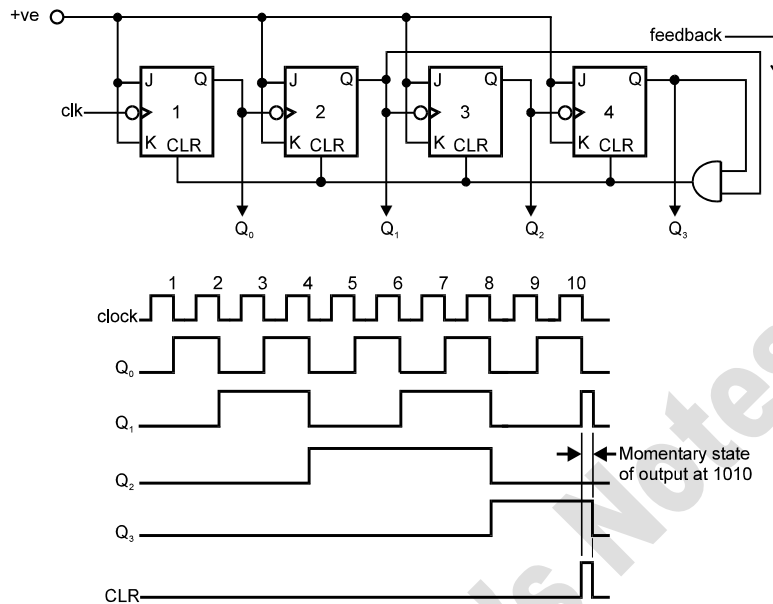
Clock	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

1.43.3 DECADE COUNTER (MOD 10)

Definition: Decade counter is defined as the counter circuit which can count the binary sequence in decimal style i.e. from 0000 to 1111. Decade counter is the modified version of *MOD 16* counter. In this, feedback from NAND gate is used, so that last six steps of *MOD 16* counter are cut off, hence, it is called *MOD 10* i.e. decade counter. Here all the flip-flops are negative edge triggered and in toggle mode.

Working: Suppose initially the outputs $Q_3Q_2Q_1Q_0 = 0000$, i.e. the counter is in *RESET* state. Now the clock pulses are applied to flip-flop-1. Then output Q_0 changes at every negative edge. So we get $Q_3Q_2Q_1Q_0 = 0001$. Now at second negative edge, Q_0 changes from '1' to '0'. This is negative change for flip-flop-2. So we get $Q_3Q_2Q_1Q_0 = 0010$.

In this way, at third negative edge we get $Q_3Q_2Q_1Q_0 = 0011$ and so on. Lastly at the tenth negative edge we get $Q_3Q_2Q_1Q_0 = 1010$. But this is a momentary state. Now $Q_3 = Q_1 = 1$. So output of AND gate, goes HIGH and all the flip-flops are *RESET* to 0000. In this way, the cycle repeats continuously.



Truth table: The following truth table of the circuit shows the output conditions at every negative edge of the clock. The 10th negative edge is momentary edge at which counter *RESETs* to 0000. Thus the counter counts from 0000 to 1001 and repeats again and again.

Clock	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10 momentary pulse	1	0	1	0

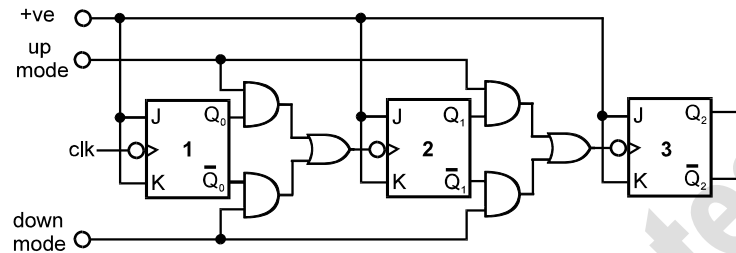
1.43.4 UP DOWN COUNTER (MOD 8)

Definition: The up down counter is defined as a counter, which can count in up or down style, depending on its control inputs. Thus, it can count from 000 to 111 as well as from 111 to 000.

Working: The output of up-down counter is incremented and decremented in sequence. Hence, it is called as the up-down counter. Here the flip-flops are in toggle mode and are negative edge triggered. Both outputs Q and \bar{Q} of previous flip-flop are connected to clock input of the next flip-flop in OR combination.

The circuit has up-control input and down-control input *separately* connected to two AND gates. When we make $UP = 1$ and $DOWN = 0$, the outputs $Q_2Q_1Q_0$ change from 000 to 111 i.e. in up sequence.

Similarly when $UP = 0$ and $DOWN = 1$, the outputs $\bar{Q}_2\bar{Q}_1\bar{Q}_0$ are connected to clock inputs through gates. So the outputs $Q_2Q_1Q_0$ change from 111 to 000 i.e. in down sequence. In up counting the BCD counting starts from 000 and ends up at 111, and then repeats again. However, in down counting the BCD counting starts from 111 and ends up at 000, and then repeats again. This change in the counting style takes place when $UP = 1$ or $DOWN = 1$.



Truth table: There are two truth tables of the circuit, since it can count either in UP or $DOWN$ sequence. Both the truth tables are given below –

Clock	Q_2	Q_1	Q_0
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

Up counting truth table

Clock	Q_2	Q_1	Q_0
1	1	1	1
2	1	1	0
3	1	0	1
4	1	0	0
5	0	1	1
6	0	1	0
7	0	0	1
8	0	0	0

Down counting truth table

1.43.5 RING COUNTER

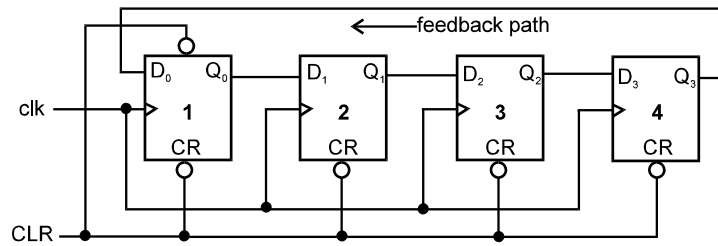
Definition: The ring counter is a counter circuit in which a single binary '1' is rotated through all the flip-flops used in the circuit.

Working: Ring counter is different from the other circuits we have studied. In it, a single '1' rotates through a number of flip-flops, like a ring.

As shown in the following circuit, four 'D' flip-flops are cascaded one after another. The output of flip-flop-4 is connected back to the input of flip-flop-1. Also, the output of flip-flop-1 is connected to the input of flip-flop-2, and so on.

Thus, it is similar to left shift register, because the data i.e. a single binary '1' is shifted bit-by-bit from one flip-flop to another at each positive edge of the clock. However, finally when the '1' comes out of Q_3 it is again fed back to D_0 to produce the ring, so this action is like rotate left ring. Suppose there is a '1' stored in the first flip-flop.

This happens because when power supply is connected, all the flip-flops are automatically cleared and a '1' is stored in first flip-flop. Hence the output of the circuit will be $Q_3Q_2Q_1Q_0 = 0001$.



Now when the first clock pulse arrives, flip-flop-1 changes its output state at positive edge of the clock and the data at shifts from LSB to MSB position. Hence, other bits also shift left by one position. Therefore, we get $Q_3Q_2Q_1Q_0 = 0010$.

At the positive edge of second clock pulse, the data moves one bit further. So we get the output as $Q_3Q_2Q_1Q_0 = 0100$. At the third clock pulse we get $Q_3Q_2Q_1Q_0 = 1000$. Lastly at the fourth clock pulse the circuit comes back to the original data position of $Q_3Q_2Q_1Q_0 = 0001$. This happens because output Q_3 gives the feedback of '1' to the input D_0 .

Truth table: The truth table of the ring counter is given below. In this we can easily understand that how a '1' rotates through the flip-flops one-by-one and produces the effect of a ring.

Clock	Q_3	Q_2	Q_1	Q_0
1	0	0	0	0
2	0	0	0	1
3	0	0	1	0
4	0	0	1	1
5	0	1	0	0
6	0	1	0	1
7	0	1	1	0
8	0	1	1	1
9	1	0	0	0
10	1	0	0	1
11	1	0	1	0
12	1	0	1	1
13	1	1	0	0
14	1	1	0	1
15	1	1	1	0
16	1	1	1	1

Self Examination

Objective questions

1. The forbidden state is avoided in _____ flip-flop.
2. At the output of single T-flip-flop the clock input frequency connected to the flip-flop is reduced to _____.
3. When a binary word is to be stored then the _____ memory circuit is used.
4. The 1-bit memory cell has _____ stable states.
5. In left shift register circuit, the data shifts from _____ to _____ in sequence.
6. In _____ system, the data shift from one system to another bit-by-bit and then it shifts all at time.
7. The propagation delay in asynchronous counter is _____.
8. In decade counter, the circuit can count from _____ to _____.
9. The largest binary number counted by an 'n' cascaded flip-flop is _____.
10. In 5-bit ripple counter, the number of states through which the counter can progress is given as _____.

Long answer questions (4 Marks)

1. What is RS flip-flop? How does it work? Draw its circuit and explain its working.
2. Draw the circuit of D flip-flop and explain its working with truth table.
3. How JK flip-flop works? Explain its working with circuit diagram and truth table.
4. What is register? Explain the working of left shift register with diagram and waves.
5. Draw a neat circuit of 3-bit ripple counter and explain its working with waveforms and truth table.
6. How we can reduce the propagation delay in asynchronous counter by using 3-bit synchronous counter circuit? Explain its working with diagram and truth table.
7. Compare between asynchronous and synchronous counter circuits with any four points.
8. How MS flip-flop works? What is the basic idea behind it? Explain its working with circuit diagram, truth tables and waveforms.
9. Draw the circuit of ring counter and explain its working with truth table.
10. What is clock? Explain its basic concept.