Unit-V

Sequential Circuits : RS Flip Flop, Clocked RS Flip Flop, D Flip Flop, Edge Triggered D Flip Flop, master-Slave Technology and its advantage, Shift Register as Flip Flop system, IC 7496, UP/DOWN counters, 74 series asynchronous counters, 74 series synchronous counter.

5. Sequential Circuits : Flip-Flops and Shift Registers 262-312

Chapter

5

Sequential Circuits : Flip-Flops and Shift Registers

1. Introduction

So far we have discussed design and analysis of combinational circuits in the last chapter. In combination circuits, output at any instant of time depends upon the input at that same instant of time i.e. they cannot store any data. Circuits in which the outputs at any instant of time depends upon inputs at that instant of time as well as past inputs, are known as sequential circuits. These circuits constitute memory elements, and may have combinational logic systems in their design.

The chapter deals with the fundamentals of the sequential logic circuits. The outputs in sequential circuits depend not only on present state of inputs but also it remember past states and because of this these circuit posses the nature of memory elements. These past states are provided by feedback from the output back to the input. Sequential logic circuits can be classified into two broad categories :

1. Asynchronous (or without clock) sequential circuits
2. Synchronous (or clocked) sequential circuits

Asynchronous sequential circuits are in real sense can be considered as combinational circuits with feedback. Their output changes according to input changes and hence has no dependence on definite timing. In synchronous sequential circuits, behaviour can be defined by knowing its signal at different instance of time and this synchronization is achieved by timing device, known as system clock. Hence output is governed by application of clock pulse in these circuits. Since the design of sequential circuits are simpler and more reliable, gained wide importance. Flip-flop, registers and counters are the most common examples of these sequential circuits, which are used in construction of digital systems. Two categories of bistable devices are the latch and flip-flop. Bistable device have two stable states; they can retain either of these states indefinitely. Basic memory elements are flip-flops, which are capable of storing binary information. The flip-flop is a basic building blocks for counters, shift registers, and other sequential control logic.

2. The flip-flops

Flip-flips are the electronic devices used in the digital world for a variety of fields. Flip-flops are used to store data temporarily, to multiply or divide, to count operations, or to receive and transfer information when used properly. Basically, flip-flops are bistable multivibrators that can go into either of the two stable states. Different types of flip-flops are used and are identified by the number of inputs, may have from two up to five inputs depending on the type. But one thing is common between all flip-flops is that all have two and only two, distinct output states, normally labeled normal output Q and complementary output $\bar{Q}$ and should always be complementary (When Q - high (1), then $\bar{Q}$ = low (0) and vice versa). In this chapter we will discuss four types of Flip-flops : R-S, D, T, and J-K Flip-flops.

3. Basic flip-flop circuit (The latch)

The flip-flop can be constructed using two methods. In first method a flip-flop can be constructed using two cross coupled NAND inverters gates or alternatively using two cross coupled NOR inverters gates (Fig. 5.1). Each of these circuits continues to remain in the state 0 or 1, if once attained and hence stored corresponding bit. In this way it is behaving as a memory cell.

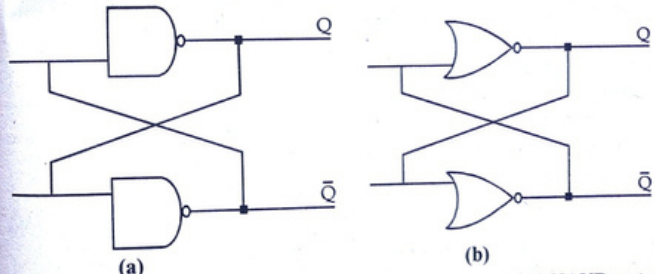
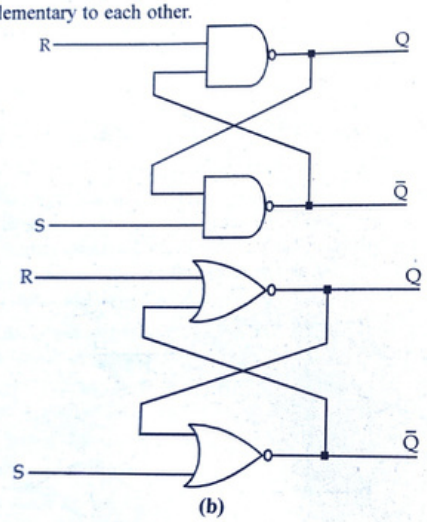


Fig. 5.1 : Construction of flip-flop : (a) Using cross-coupled NAND gates; (b) Using cross-coupleted NOR gates

Let us assume output of first NAND gate is Q = 1 (in this state flip-flop is said to be set (storing a 1), which is also the input of the second NAND gate will be $\bar{Q}$ = 0. Which again makes the input of first NAND gate 0 and because of this Q =1, which again confirms our results. Similarly outputs Q = 0 (in this state flip-flop is said to be **reset** or **clear** (storing a 0) and $\bar{Q}$=1 can also be verified. Hence these crossed coupled gates store a 0 or 1 bit indefinitely i.e. a bit 0 or 1 is locked *(latched) in it and no other data can enter or exit. This property of the circuit is termed as memory and information is latched in these circuits, hence these circuits are also termed as "*latch*".

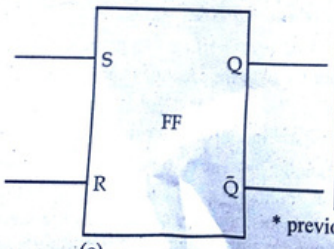
4. R-S Flip-Flop (R-S Latch)

This is the most fundamental type of flip-flop, where S and R stand for set and reset. This flip-flop is used to temporarily hold or store information until it is needed. Basic unit consists of single R-S flip-flop can store one binary digit (1 or 0) if you like to store more digits, lets say 3 then to generate 3 digit number three R-S flip-flops are needed. The standard symbol and circuit diagram for the R-S flip-flop is shown below in Fig. 5.2. This flip-flop sometimes referred as R-S latch. The outputs Q and $\bar{Q}$ are complementary to each other.



(b)

R-S latch as (a) two cross couple NAND inverters; (b) two cross coupled NOR intverter



(c)

R	S	Q	$\bar{Q}$	Condition
0	0	*	*	*Latched*
0	1	1	0	*Set*
1	0	0	1	*Reset*
1	1	1	1	*Jammed*

* previous condition

(d)

Fig. 5.2 : (a) and (b) circuit diagrams; (c) Standard symbol and (d) truth table

The R-S flip-flop has two output condition one is Set (When the Q output is 1 an Q is 0) and another one is rest or clear (When Q is 0 and Q is 1). When the R and S inputs both are at 0, the circuit behave in same way as two cross coupled NOR inverter gates/NAND inverter gates circuit behaves, and in this condition the flip-flop is said latched and the outputs cannot be used in this situation. If S is pulsed 1 while R is held 0, then the Q output is forced 1, and stays 1 even after S returns 0; similarly if R is pulsed 1 while S is held 0, then the Q output is forced 0, and stays 0 even after R returns 0. When both R and S are 1, the flip-flop will hold or "jammed". Hence the condition and input R = 1 and S = 1 should never allowed in this flip-flop. The use of R-S flip-flop should be avoided in this condition. We can also use construction diagrams (Fig. 1) to determine the truth table for the R-S FF.

Asynchronous R-S Flip-Flop

In R-S latch, the desired digital information cannot be stored in any way. Actually as the power start the circuit itself switches to one of the stable state (Q = 1 or 0) and we cannot predict the state. If we change the two cross coupled inverter gates by a two inputs gates, where other two inputs of the inverter gates are used to input the desired digital information, the new modified circuit is now a R-S Flip-flop (Fig. 5.3). The operation of R-S flip-flop is similar to R-S latch, which can be seen from following discussion.

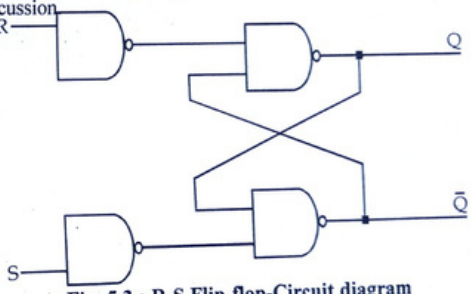


Fig. 5.3 : R-S Flip-flop-Circuit diagram

If R = S = 0, the output Q remains unchanged as of its previous state. If R = 0 and S = 1, the flip-flop output sets to state Q = 1 and $\bar{Q}$ = 0. Similarly if R = 1 and S = 0, the output resets to Q = 0 and $\bar{Q}$ = 1.

Advantage is that if input conditions are changed from S = 1 and R = 0 to S = R = 0 or from S = 0 and R= 1 to S = R = 0, the output remains unchanged.

If S = R = 1, then both of the outputs tried to become 1 or 0, which is not allowed and hence in this condition, use of flip-flop is prohibited.

Clocked R-S Flip-Flop

In this type of R-S flip-flop, a clock pulse is also applied with two inputs, hence set or reset the memory cell always be done in synchronization with a clock (a regulated train of the pulse signals). This type of R-S flip-flop is also termed as

clocked R-S flip-flop. Here two inputs exactly behave in same way as in an unclocked R-S flip-flop.

In this circuit (Fig.-5.4) if clock pulse is high (1), its operation is exactly same as the R-S flip-flop. If clock pulse is low (0), two NOR inverters with input R, S and clock have output always equal to 1, no matter what are the values of the inputs R and S. So it is solely the clock pulse that determines when transition of Q should occur.

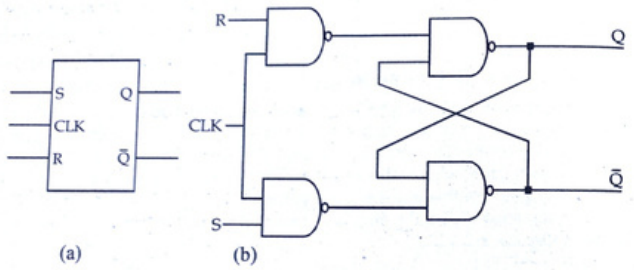


Fig. 5.4: Clocked R-S flip-flop (a) Standard symbol, and (b) circuit diagram

Let us assume that inputs do not change during the presence of the clock pulse, then we can represent the outputs of flip-flop as per the truth table for R-S flip-flop. Assume after nth clock pulses passed, the output is denoted by Q_{n+1}. If we take inputs as R_n and S_n then truth table for clocked R-S flip-flop is given as under :

Inputs		Outputs	
R_n	S_n	Q_{n+1}	$\bar{Q}_{n+1}$
0	0	Q_n	$\bar{Q}_n$
0	1	1	0
1	0	0	1
1	1	?	?

Truth table for clocked R-S flip-flop

Let us verify the truth table, consider a case when a clock pulse is applied. If $R_n = S_n = 0$, the output remain same as if it was there before the application of the clock pulse, i.e. $Q_{n+1} = Q_n$ (as indicated in first row of the truth table).

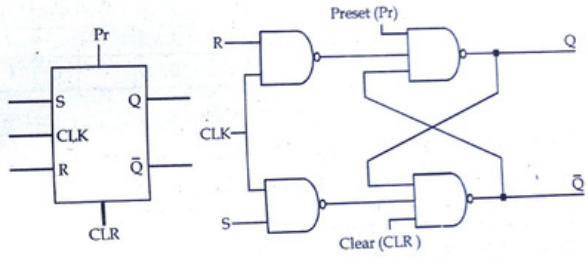
If $R_n = 0$ and Sn = 1, the output $Q_{n+1} = 1$ and $\bar{Q}_{n+1} = 0$, and if $S_n = 0$ and $R_n = 1$, the output $Q_{n+1} = 0$ and $\bar{Q}_{n+1} = 1$ (same as indicated by the second and third row of the truth table).If $R_n = 1$ and $S_n = 1$, the outputs for two NAND inverters with R, S and Clock inputs goes 0 and this condition leads to the invalid results $Q_{n+1} = \bar{Q}_{n+1}$

= 1. Hence the output state of the circuit is undefined, and that's why it is indicated by a question mark in the truth table.

Preset and Clear

In many applications it is required to initially set or reset the flip-flop, i.e. initial state of the flip-flop has to be assigned. This is achieved by using two asynchronous inputs, named as Preset (Pr) and Clear (CLR) inputs. These inputs can be applied any time between the clock pulse and these inputs are not in synchronism with the clock. Fig. 5.5 shows the symbol, circuit diagram and truth table of this flip-flop.

By making Pr input 0 and CLR input 1 we can see that the first output will certainly be Q = 1, which will make $\bar{Q} = 0$. Hence to apply Pr input as 0 sets the flip-flop. By making Pr input 1 and CLR input 0 we can see that the first output Q will become 0 and $\bar{Q}$ will be 1, which will resets the flip-flop.



(a) Symbol **(b) Circuit diagram**

Pr	CLR	CLK	R_n	S_n	Q_{n+1}	$\bar{Q}_{n+1}$
1	1	1	0	0	Q_n	Q_n
0	1	1	0	1	1	0
1	0	1	1	0	0	1
1	1	1	1	1	?	?

(c) Truth table

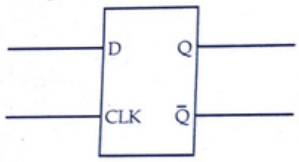
Fig. 5.5 : Clocked R-S flip-flop with asynchronous inputs Pr and CLR: (a) Standard symbol, (b) circuit diagram, and (c) truth table

5. D Flip-flop

An R-S flip-flop is rarely used in actual sequential logic circuits. However, it is the fundamental building block for the very useful D flip-flop. The D flip-flop has only a single input referred as D-input or data input. That data input is connected to the S input of an R-S-flip flop, while the inverse of D is connected to the R input.

This flip-flop is used to obtain two middle rows of the R-S flip-flop.

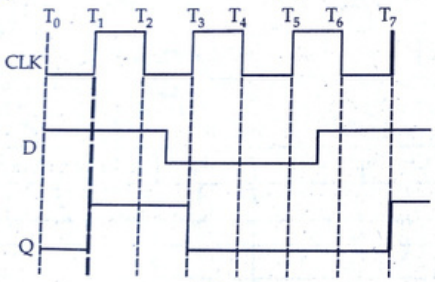
A D flip-flop can be obtained from an R-S flip-flop by connecting an inverter between its two inputs, and then selecting resulting single input as data (D) input terminal. The inputs are the data (D) input and a clock (CLK) input (time pulse to control operations). The D flip-flop is used to store data at a predetermined time and hold it. This flip-flop is also called as Delay flip-flop as the data input is delayed up to one clock pulse before it is seen at the output. The standard symbol, truth table and timing diagram of a D flip-flop is shown in Fig. (5.6). According on the circuit design, the clock (CLK) as a constant input at a given frequency can be selected as a square wave, a constant frequency, or asymmetrical pulses. The data (D) input will be present when there is a need to store information.



$D(=S)$	$\bar{D}(=R)$	CLK	Q	$\bar{Q}$
x	$\bar{X}$	0	Q_0	$\bar{Q}_0$
0	1	1	1	0
1	0	1	0	1

x = any value

(a) (b)



(c)

Fig. 5.6 : D flip-flop : (a) symbol, (b) truth table, and (c) timing diagram

Let's assume that at T_0, clock is 0, D is 1, and Q is 0. At T_1, when the clock goes to 1, Q also goes to 1 and remains at 1 even though D goes to 0 between T_2 and T_3. At T_3, the positive-going pulse of the clock causes Q to go to 0, reflecting the condition of D. The positive-going clock pulse at T_5 causes no change in the output because D is still 0. Between T_5 and T_6, D goes 1, but Q remains 0 until T_7 when the clock goes 1.

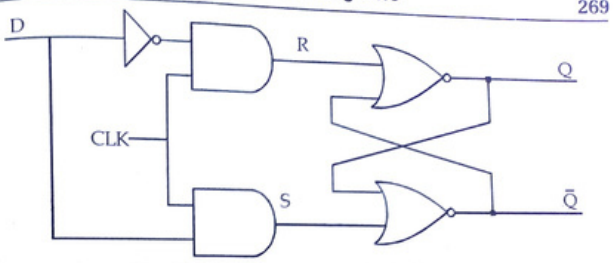


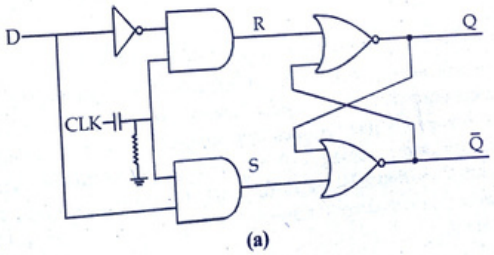
Fig. 5.7 : Circuit diagram of a D flip-flop

When the clock pulse goes to 1, the output changes with D input and hence the flip-flop is said as transparent. This is equivalent to say that input data appears at the output at the end of a clock pulse. So in this process to appear at output, input data get delayed and hence the name is given Delay (D) flip-flop. Circuit diagram for this flip-flop is shown in Fig. 5.7. One can also verify its truth table using its circuit diagram. These flip-flops are used to store one bit information.

Edge Triggered D Flip-Flop

A simple D flip-flop can be converted into an edge-triggered D flip-flop by connecting an RC circuit (or edge detector: these circuits are designed to convert each applied clock pulse into a narrow positive spike, this will be done by utilizing the property of differentiating of a suitable RC circuits) at its clock input. This type of flip-flop acts as a delay device because it has to wait until an active clock edge received. Before which, this cannot transmit its input data to the output.

The circuit diagram, standard symbol and timing diagram for this flip-flop are shown in Fig. (5.8). The operation of this flip-flop is identical to simple D flip-flop except that the transitions of Q occur only at the active clock-edge, as shown in timing diagram.



(a)

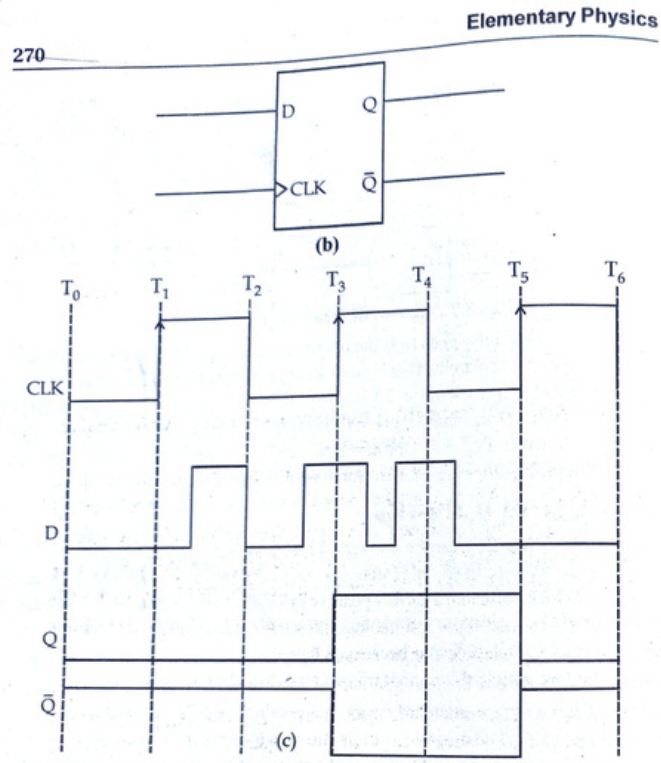
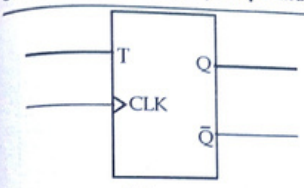


Fig. 5.8 : Edge triggered D flip-flop : (a) Circuit diagram, (b) Standard symbol, and (c) Timing diagram (for positive edge triggering)

6. Toggle Flip-Flop

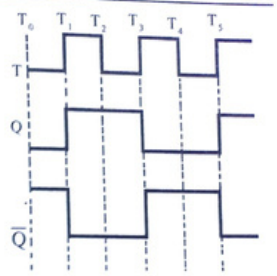
The toggle, or T, flip-flop is a bistable device with one common input apart from the usual clock input and the two outputs. It has the property to change state on command from a common input terminal. The standard symbol for a positive edge triggered T Flip-flop is shown in Fig. 5.9. The T input is be preceded by an inverter, which indicates this flip-flop will toggle on a high(1)-to-low(0) transition of the input pulse and the absence of inverter indicates that the flip-flop will toggle on a low(0)-to-high(1) transition.



(a)

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

(b)



(c)

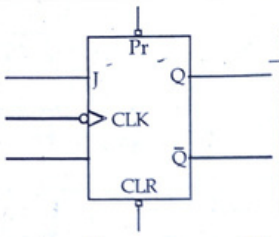
Fig. 5.9 : (a) T flip-flop : Standard symbol, (b) Truth table, and (c) Timing diagram

The timing diagram in Fig. 5.9 (c) shows the toggle input and the resulting outputs Q_{n+1} and $\bar{Q}_{n+1}$. After let us take a special case of an initial condition (T_0) of Q_n being 0 and $\bar{Q}_n$ being 1. At time T_1, the toggle changes from a 0 to 1 condition and the flip-flop changes its state; Q_{n+1} goes to 1 and $\bar{Q}_{n+1}$ goes to 0. At time T_2 the outputs remain the same since the device is switched only by a 0-to-1 transition. At time T_3, when the toggle input goes 1, Q_{n+1} goes 0 and $\bar{Q}_{n+1}$ goes 1 and remain same until the time T_5 (because T input remain 0).

From the above case it is clear that between two complete cycles of toggle input (between T_1 and T_5), one cycle for both outputs Q_{n+1} and $\bar{Q}_{n+1}$ are observed for the same time period. Hence because of this property (as the output cycle is one-half of the input cycle), this device can be used for dividing the input by a factor of 2.

7. J-K Filp-Flop

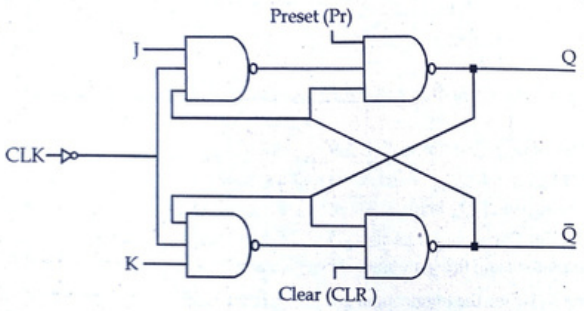
The J-K Flip-Flop is the most widely used flip-flop because of its versatile applications. When properly used it may perform the function of an R-S, T, or D Flip-flops. The J-K flip-flop is a five-input device; two inputs : J and K are available for data input, the CLK input is available for input the clock, and the Pr and CLR inputs are the preset and clear inputs, respectively. The outputs Q and $\bar{Q}$ are the normal complementary outputs like other flip-flops. The standard symbol for the J-K level-clocked Flip-flop is shown in Fig. 5.10.



(a)

Input J	Input K	CLK	Q	Q̄	State
0	0	1	Q_0	Q_0	No change
0	1	1	0	1	Reset
1	0	1	1	0	Set
1	1	1	Q_0	Q_0	Toggle

(b)



(c)

Fig. 5.10 : A negative-edge triggered J-K flip-flop: (a) Standard symbol, (b) Truth table and (c) Circuit diagram

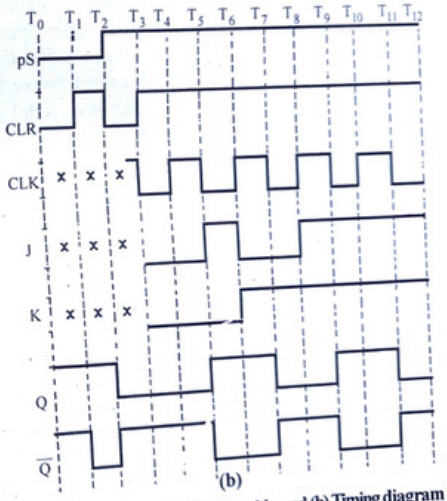
J-K flip-flop is a modified R-S flip-flop except with one major difference, that is the toggle mode of operation. In toggle mode of operation, when J = K = 1, instead of an invalid output is produced, drives the output to switch into opposite state at the active clock-edge.

8.7.1 Preset and Clear

To initially set or reset the J-K flip-flop, two asynchronous inputs Preset (PS) and Clear (CLR) are used. The operation of J-K flip-flop with these inputs is discussed below with timing diagram.

Inputs					Outputs	
Preset	Clear	Clock	J	K	Q	Q̄
0	0	X	X	X	1	1
0	1	X	X	X	1	1
1	0	X	X	X	1	0
1	1	1	0	0	0	0
1	1	1	1	0	1	0
1	1	1	0	1	0	1
1	1	1	1	1	TOGGLE	
1	1	1	X	X	Q	Q̄

X = irrelevant (a)



(b)

Fig. 5.11 : J-K flip-flop: (a) Truth table, and (b) Timing diagram

In Fig. 5.11 (a) and (b) one can see the truth table and the timing diagrams for the J-K Flip-flop, which can be explained watching the timing diagram for the inputs. Line 1 of the Truth Table corresponds to the time T_0 in the timing diagram. The Pr and CLR inputs are both 0. The CLK, J, and K inputs are irrelevant. At this point the flip-flop is jammed, and both Q and $\bar{Q}$ are 1. As with the R-S Flip-flop, this state cannot be used. At time T_1, Pr remains 0 while CLR goes 1. The Q output remains

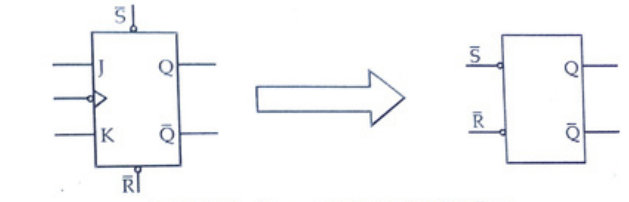
1 and $\overline{Q}$ goes 0. The Flip-flop is in the reset condition (line 2 of the Truth Table). At time T_2, Pr goes 1, CLR goes 0, Q goes 0, and $\overline{Q}$ goes 1. At this point the Flip-Flop is CLEARED (line 3 of the Truth Table). The condition of the CLK, J, and K inputs have no effect on the Pr and CLR actions since these inputs override the other inputs. Starting at T_3, Pr and CLR will be held at 1 so as not to override the other actions of the Flip-Flop. Using the PS and CLR inputs only, the circuit will function as an R-S Flip-Flop. Between T_2 and T_3, the CLK input is applied to the device. Since the CLK input has an inverter, all actions will take place on the negative-going transition of the clock pulse. Line 4 of the Truth Table shows, Pr and CLR 1, a negative-going CLK, and J and K at 0. This corresponds to T_3 on the timing diagram. In this condition the Flip-flop holds the previous condition of the output. In this case the Flip-flop is reset. If the circuit were set when these inputs occurred, it would remain set. At time T_5, we have a negative-going clock pulse and a 1 on the J input. This causes the circuit to set, Q to go 1, and $\overline{Q}$ to go 0. See line 5 of the Truth Table. At time T_6, J goes 0, K goes 1, and the clock is in a positive-going transition. There is no change in the output because all actions take place on the negative clock transition. At time T_7, when J is 0, K is 1; the clock is going negative, the Flip-flop resets, Q goes 0, and $\overline{Q}$ goes 1 (line 6 of the truth table). With both J and K=1 and a negative-going clock (as at T_9 and line 7), the flip-flop will toggle or change state with each clock pulse. It will continue to toggle as long as J and K both remain 1. Line 8 of the Truth Table indicates that as long as the clock is in any condition other than a negative-going transition, there will be no change in the output regardless of the state of J or K.

As mentioned at the beginning of this section, J-K flip-flop may be used as R-S, T, or D Flip-flops. Fig. 5.12 shows how a J-K flip-flop can be used to perform the other functions.

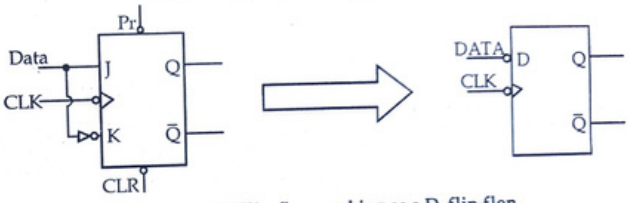
The Race-around condition :

The invalid condition (S=R=1) for R-S flip flop is eliminated in a J-K flip flop using a feedback connection. For inputs J=K=1 and assume Q=0, when a clock pulse is applied, after a time interval of propogation delay, the output changes to Q=1. After another time interval of propagation delay, output again changes back to Q=0. Hence during the clock pulse duration, output Q changes between 0 and 1 and at the end of clock pulse, output Q is uncertain, such situation is known as race-around condition and can be seen for a level triggered J-K flip-flop. To avoid this difficulty Master-Slave J-K flip-flop can be used.

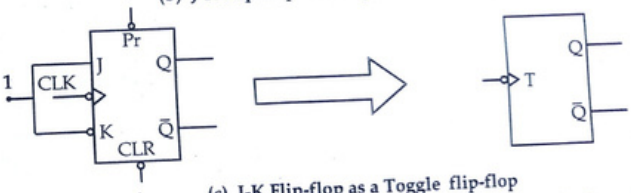
J-K flip-flop implementation as other flip-flops



(a) J-K Flip-flop working as a R-S flip-flop



(b) J-K Flip-flop working as a D flip-flop



(c) J-K Flip-flop as a Toggle flip-flop

Fig. 5.12 : Versatility of J-K flip-flop: (a) As R-S flip-flop, using the Pr and CLR inputs, (b) as D flip-flop, Data applied to the J input and then same data to K input through an inverter, and (c) as a toggle flip-flop, using both J and K inputs 1

In Fig. 5.12 (a) using just the Pr and CLR inputs of the J-K flip-flop will cause it to react like an $\overline{R} - \overline{S}$ flip-flop. In Fig. 5.12 (b), data is applied to the J input. This same data is applied to the K input through an inverter to ensure that the K input is in the opposite state. In this configuration, the J-K performs the same function as a D Flip-flop. In Fig. 5.12 (c) shows both the J and K inputs held at 1. The Flip-flop will change state or toggle with each negative-going transition of the clock just, as a T Flip-flop will do.

8. Master Slave J-K Flip-Flop

The Master-Slave (M-S) J-K flip-flop actually contains two cascaded flip-flops with feedback from the outputs of the second to the inputs of the first. The first flip-flop is known as master is a positively-edge-triggered J-K flip-flop and the second one, known as slave, is a negatively-edge-triggered R-S flip-flop. This implies the following:

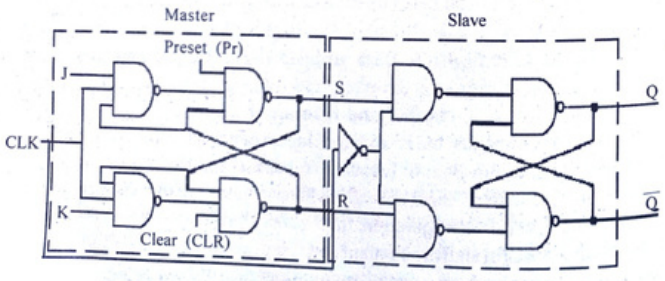
(1) When positive clock-edge (CLK = 1) is applied, first flip-flop triggers and outputs of masters respond to the J and K inputs, and slave remains inoperative.

(2) When the slave triggers at negative clock-edge ($\overline{CLK}$ = 1), master remains inoperative and therefore the outputs Q and $\overline{Q}$ follows the outputs of master.

As second flip-flop always follows the output of masters, it is known as slave here. Circuit diagram and truth table of master-slave J-K flip-flop is shown in Fig. (5.13) which comprises of truth table of J-K and R-S flip-flops.

				Inputs				Q	$\overline{Q}$	State
Preset	Clear	CLK	Clear	J	K	R	S			
0	0	X	0	X	X	X	X	*	*	No output
0	1	X	1	X	X	X	X	1	0	Set
1	0	X	0	X	X	X	X	0	X	Reset
1	1	X	1	0	0	Q_0	$\overline{Q}_0$	Q_0	$\overline{Q}_0$	No change
1	1	1	1	0	1	0	1	0	1	Reset
1	1	1	1	1	0	1	0	1	0	Set
1	1	1	1	1	1	$\overline{Q}_0$	Q_0	$\overline{Q}_0$	Q_0	Toggle

(a) Truth table **X= irrelavant**



(b) Circuit diagram

Fig. 5.13 : Master slave J-K flip-flop: (a) Truth table and (b) Circuit diagram

Advantages : The Master Slave J-K flip-flop toggles once per input cycle. There are two distinct steps in setting the Q output in Master Slave flip-flop. First, the master is set while clock pulse is positive and then slave is set while the clock pulse is negative. One more reason to use this flip-flop is that it provides a symmetrical output. Another reason to use this is to control the starting phase of the clock. Another advantage is race-around condition of J-K flip-flop is eliminated for this flip-flop.

Disadvantages :

Master slave J-K flip-flop is largely replaced by edge-triggered flip-flops and these days these flip-flops are essentially obsolete because it cannot meet with the requirement to maintain the control inputs stable for the entire period of pulse duration.

9. Applications of flip-flops

There are various applications of flip-flops but here in this chapter we will study about the shift registers and their properties of shifting and storing data. Unlike counters these shift registers are used primarily for data storage and typically don't possess any internal sequence of data. Shift registers consists of an arrangements of flip-flops and important for storing and transfer of data in digital systems.

10. Registers

A register is a temporary storage device. Registers are used to store data, memory addresses, and operation codes. Registers are normally referred to by the number of stages they contain or by the number of bits they will store. For instance, an eight-stage register would be called an 8-bit register. The contents of the register are also called a WORD. The contents of an 8-bit register are an 8-bit word. The contents of a 4-bit register are a 4-bit word and so on. Registers are also used in the transfer of data to and from input and output devices such as teletypes, printers, and cathode-ray tubes. Most registers are constructed of flip-flops and associated circuitry. They permit us to load or store data and to transfer the data at the proper time.

11. Shift Registers

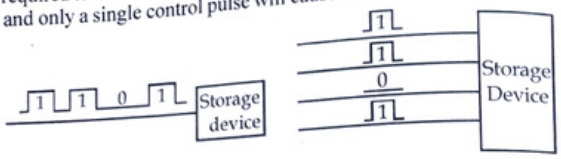
A shift register is one of the applications of flip-flops, which enabled it to store and shift the data bits. A shift register is a register in which the contents may be shifted to one or more places to left or right, and hence can be used to perform a variety of functions. It can be used for serial-to-parallel transfer & vice versa and also for scaling binary numbers. Before we get into the operation of the shift register, let's discuss serial-to-parallel transfer, parallel-to-serial transfer, and scaling.

11.1 Serial and Parallel Transfers

Serial and parallel terms are used to describe a way for data or information transfer from one place to another, serially along a single line (one bit at a time) and parallel (all bits transfer simultaneously).

Fig. 5.14 shows how both of these transfers occur, in Fig. 5.14 (a) the four-bit data 1101 is being transferred to a storage device serially, hence individual control

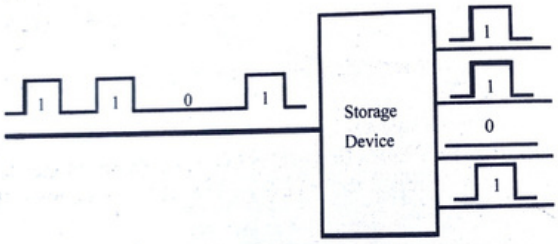
pulse is required to store each bit of data. In Fig. 5.14 (b), the data 1101 is transferred parallel and only a single control pulse will cause the entire word to be stored.



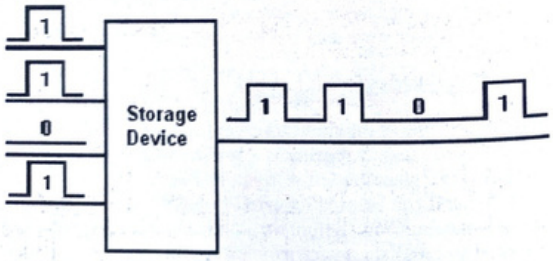
(a) Serial Transfer (b) Parallel Transfer

Fig. 5.14 : Data transfer methods : (a) Serial transfer; (b) Parallel transfer

Serial-to-parallel transfer or parallel-to-serial transfer describes the manner in which data is stored in a storage device and also the manner in which data is transferred from the storage device. Fig. 5.15 (a) shows serial to parallel transfer of data and 5.15 (b) shows how Parallel-to-serial transfer of data taking placed.



(a) Serial-to-parallel transfer



(b) Parallel-to-serial transfer

Fig. 5.15 : Data transfer method : (a) Serial-to-parallel transfer; (b) Parallel-to-serial transfer

Serial transfer depends upon the data length and as long as the data length, the longer the time transfer will take. Although parallel transfer is much faster, it requires less time to transfer the data.

Scaling

Scaling is a way to change the magnitude of a binary number into its decimal counterpart. We all know that shifting binary numbers to the left increases its value, and shifting to the right decreases their value. To explain the meaning of scaling, let us assume that a 5-bit shift register containing the binary number 01010.

0 1 0 1 0 (Decimal equivalent = 10)

If the entire number is shifted one place to the left, it will put the register in the following condition:

1 0 1 0 0 (Decimal equivalent = 20)

The binary number 0 1 0 1 0 has a decimal equivalent value of 10, and 1 0 1 0 0 has decimal equivalent as 20. Hence we can see by shifting the number one place to the left, we have multiplied it by 2. Similarly if shifting a binary number is done to the right, it will decreases the value of the number by a power of 2 for each place. Let's take the example of same 5-bit register containing 0 1 0 1 0 and shift the number to the right.

A shift of one place to the right will result in the register being in the following condition:

0 0 1 0 1 (Decimal equivalent = 5)

Comparing decimal equivalents of binary numbers before and after shifting we can see that it has decreased the value from 10 to 5. Hence we have effectively divided the number by 2.

12. Shift Register Operations

Fig. 5.16 shows a typical 4-bit shift register. There are provisions for serial and parallel input and serial and parallel output.

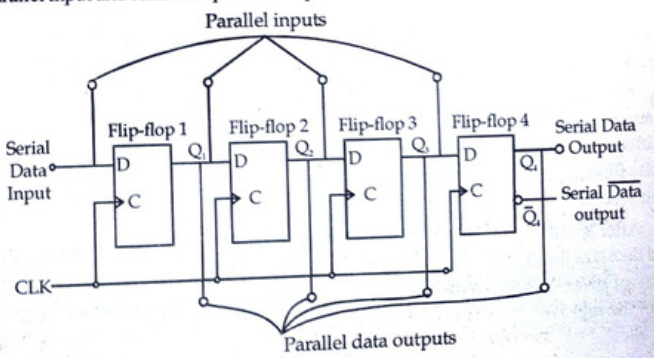


Fig. 5.16 : A four bit shift register

Before any operation takes place, a CLEAR pulse is applied to each flip-flop to ensure that all the Q output are 0. The simplest modes of operation for this register are the parallel inputs and outputs. Parallel data is applied to the SET inputs of the flip-flops and results in either a 1 or 0 output, depending on the input. The outputs of the flip-flops may be sampled for parallel output.

Basic Data movement in Shift Registers

The digital data is a shift register can be entered and retrieved in two ways ; in serial from (one bit at a time) and in paralled form (all bits at the same time). This all we have seen till above sections in parallel to serial transfer and in serial to paralled transfer. Accordingly, the shift registers can be classified to operate in the following four modes ;

1. Serial in / Serial out (SISO) shift registers
2. Serial in / Paralled out (SIPO) shift registers
3. Paralled in / Serial out (PISO) shift registers
4. Paralled in / Parallel Out (PIPO) shift registers

As we have discussed according to direction of data movement we can classify our shift registers in above mentioned four categories. Here in this section we will discuss one by one each of these categories.

Serial in/serial out shift registers

A serial in/serial out shift register accepts and produces stored information to its output in a serial manner. These registers can shifts the data by one bit per clock pulse. Hence it requires as many numbers of flip-flops as the number of bits to be shifted, and an equal number of clock pulses also. as we have shown in fig. (5.16) the serial input and serial output points are used for its working.

To illustrate let us assume a four bit data 1 0 1 0 is to be transferred into the register. For serial entry of this data we have to use four flip-flop stages so register can store up to four bit of data.

Now look at the entry of the right most digit 0 on flip-flop 1 in Fig. 5.17. This will make D = 0 for flip-flop 1 and when first CLK pulse is applied, flip-flop 1 is reset and store 0.

Now second bit 1 is applied making D = 1 for flip-flop 1 and D = 0 for flip-flop 2 and on application of second CLK pulse, flip-flop 1 is set and store 1 and 0 is shifted to flip-flop 2. Similarly third bit 0 is applied and on application of third CLK pulse flip-flop 1 is reset and store 0, and 1 is shifted to flip-flop 2 and output of flip-flop 3 become 0.

After application of fourth CLK pulse fourth bit 1 is also shifted to flip-flop 1 and then 0 to flip-flop 2, 1 to flip-flop 3, and 0 to flip-flop 4. This complete the serial entry of given data into shift register. Similarly data can be extracted out serially from the register.

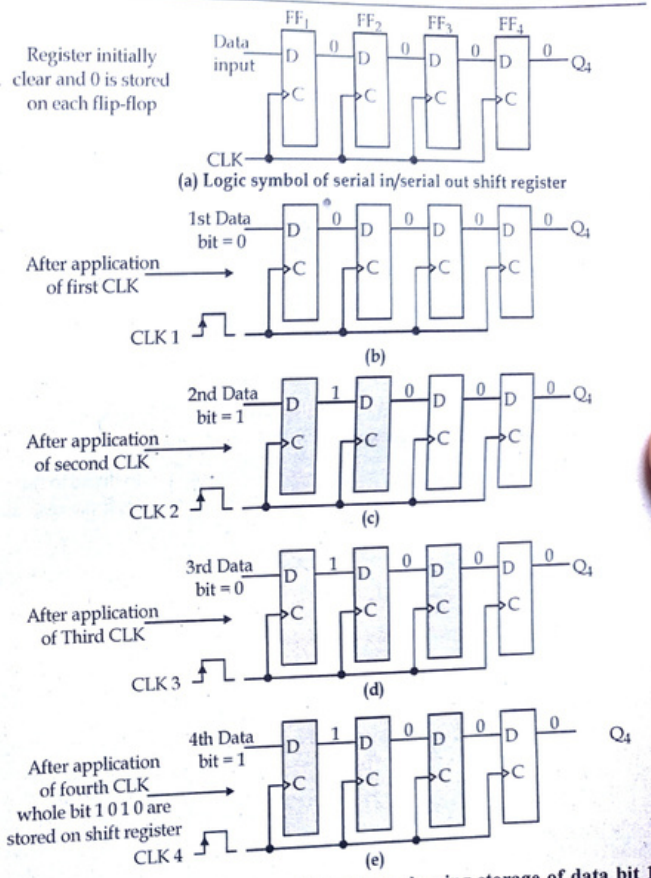


Fig. 5.17 : A serial in/serial out shift register showing storage of data bit 1 0 1 0.

Serial in/parallel out shift registers

In these type of shift registers data bits are entered serially as we have discussed for serial in/serial out shift register in Fig. 5.17. The difference is in the way of obtaining data from these registers in parallel manner so that output of each stage is

available simultaneously rather than on a bit by bit as we have obtained in case of serial in/serial out shift register. A typical 4 bit serial in/parallel out register with four stages is shown in Fig. 5.18.

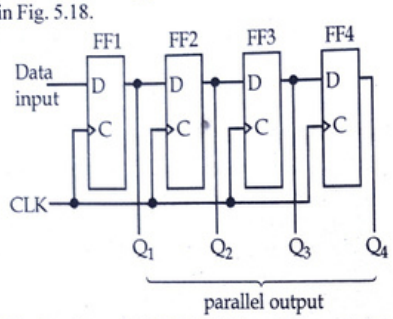


Fig. 5.18 : Logic symbol of Serial in/parallel out shift register.

Parallel in/serial out shift Registers

For parallel entry of data inputs, all bits are simultaneously introduced to their respective stages on parallel lines rather than bit-by-bit entry as with the case in serial data inputs. In these type of registers data retrieval is in the form of serial transfer. A logic symbol of this register is shown in Fig. 5.19.

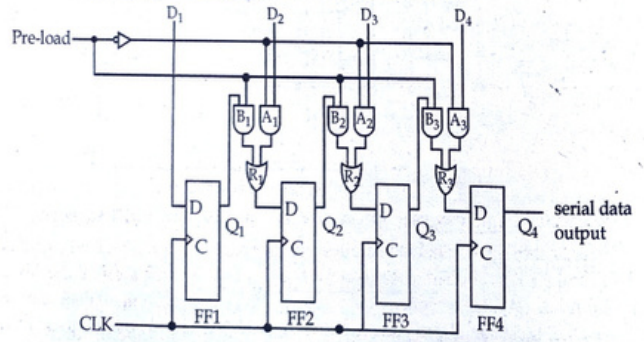


Fig. 5.19 : Logic symbol of parallel in/serial out shift registers

Fig. 5.19 shows a 4 bit parallel in/serial out shift register. It is clear from the Fig. that this register have four data inputs D_1, D_1, D_3, D_4 and a preload input (It allows parallel entry of data into the register).

When preload is low (0), gates A_1, A_2, A_3 are enabled and data bits are applied to respective data inputs D_1, D_2, D_3, D_4, when a CLK pulse is applied accordingly

with D = 1 flip-flops will set and those with D = 0, will reset and hence these registers can simultaneously store all four bits.

When preload is high (1), gates B_1, B_2, B_3 are enabled and A_1, A_2, A_3 are disabled. The gates B_1, B_2, B_3 are enabled causes gates R_1, R_2, R_3 to shift data inputs to right from one stage to the next. Here OR gates R_1, R_2, R_3 allow not only the normal shifting operation but also parallel data-entry operation.

Parallel in/Parallel out shift Registers

The parallel in/parallel out shift registers allows both the entry of data inputs simultaneously on all statges, and all bits appear the parallel outputs simultaneously at the occurrence of a CLK pulse.

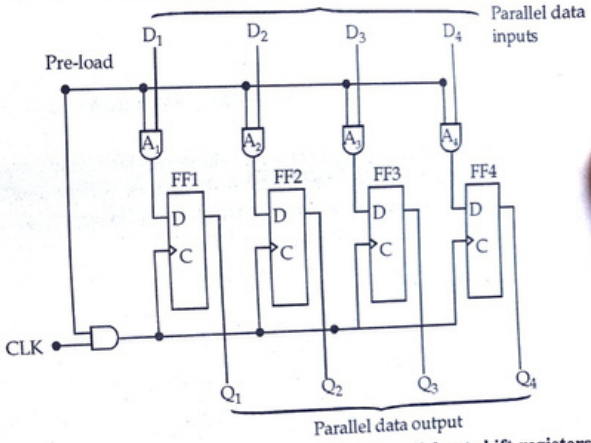


Fig. : 5.20 : Logic diagram of parallel in / parallel out shift registers.

Fig. 5.20 shows a parallel in / parallel out shift register. When preload signal is high (1) and CLK is applied, each data bit D_1, D_2, D_3, D_4 shift through the AND gates A_1, A_2, A_3, A_4 into their respective stages. This whole shifting of data require only one CLK pulse into the register and in next CLK pulse the whole word can also be retrieved across Q outputs. When preload is low (0), no data can enter into flip-flops because all AND gates will be disabled.

13. IC 7496

IC 7496 is a 5-bit parallel-in/parallel-out shift register. It have five input pin D_1, D_2, D_3, D_4 and D_5 to parallel input the data. It also have five output pin Q_1, Q_2, Q_3,

Q_4, and Q_5 to parallel access the output. Basically IC 7496, for simultaneous entry of all data bits, produces the bits at parallel output simultaneously on input of one clock pulse.

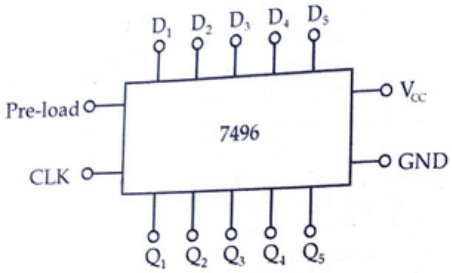


Fig. 5.21 : IC 7496

The pin preload is there to control the parallel entry of the data. When preload is low (0), data entry is restricted, when preload is high (1), data enter through D_1, D_2, D_3, D_4 and D_5 parallely.

14. Applications of shift Registers

The primary function of shift registers is to store and shift the data bits. However, these registers can be used for other applications as :

(1) Shift registers can be used to introduce a time delay in digital signals by properly selecting ; the total number of flip-flop stages and frequency of clock pulse. The time delay is given By

$$\Delta t = \frac{n}{f}$$ where n = Number of flip-flop stages

f = Clock frequency.

This happens because each flip-flop delays the input by a factor $\left(\frac{1}{f}\right)$, before it appears at the output to retrieve.

(2) Used to convert serial data into parallel form.

(3) Used to convert parallel data into serial form.

(4) Used to make ring counters (we will study counters in next chapter).

(5) Used as sequential memories.

(6) Used to produce a specified sequence of bits in synchronism with a clock.

15. Digital Counters

In the previous topics you have read about the flip-flops and shift registers. Flip-flops can be used to perform counting when we connect them and also used to perform shift registers operations to store and shift the data. In this topic we will read about the digital counters, which are sequential circuits, and are used to count the occurrence of clock pulse. A digital counter is simple device that counts and generate a specific sequence of bits. Counters may be used to count operations, quantities, periods of time, for dividing frequencies, for addressing information in storage, or for temporary storage. Counters are made using a series of flip-flops connected together to perform counting operations. The total number of flip-flops used and the way in which they are connected determine the number of stable states, a counter can indicate is called *modulus*. For example, the modulus of a four-stage counter would be 16 since, it is capable of indicating 0000 to 1111 counts. To describe the count capability of counters, one-more term is used called modulo, i.e. for a four-stage binary counter the count capability will be modulo-16, and so on.

Counters are of many types and design but almost all of them are made by connecting flip-flops as the basic units. These counters can be designed to count the upward sequence, or in the downward sequence. According to synchronization with a clock the digital counters are classified into two broad categories :

1. Asynchronous (or Ripple) counters
2. Synchronous (or parallel) counters

In asynchronous type of counters all flip-flop are not clocked simultaneously, the first flip-flop is clocked by the external clock and the other flip-flop stages are clocked by the output of the preceding flip-flop. In synchronous type of counters all flip-flop stages are simultaneously clocked i.e. all of the flip-flops have a clock input connected to them. Within each of abovementioned categories counters are primarily classified by the number of states, type of the sequence, or the the number of flip-flops stage in the counter.

16. Modules of a Counter

Counters are identified by their modules like mod-6, mod-5 etc. The modules of a counter is a number of unique states or total number of counts that the counter will sequence through in one complete counting cycle (i.e. resets back to its initial conditions). The modules of a counter (maximum modules) is defined as :

$$\text{Modules} = 2^n$$

Where n is the number of flip-flops in the counter and counter is named as mod-2^n counter. These counters are also known as divide by 2^n counters because they divide the clock frequency by a number 2^n. These counters usually follows the

natural binary number sequence and hence also termed as binary counters.

Some of the counters whose moduls $N \neq 2^n$ defined as mod-N counters e.g. mod-5, mod-6, mod-10 counters etc. The number of flip-flops required to construct such an mod-N counter (divide by N counter) is the smallest value of n ; for which

$$N < 2^n$$

e.g. for mod-5, mod-6, mod-7 counters total number of flip-flop stages must be 3 because $2^3 > 5, 6, 7$ so n must have smallest value 3. The counting sequence in these counters may or may not be in natural binary sequence.

The table-1 below summarizes different types, total counts (states) and highest decimal counterpart of the counts for the different counters.

Table 1 : Digital counters types

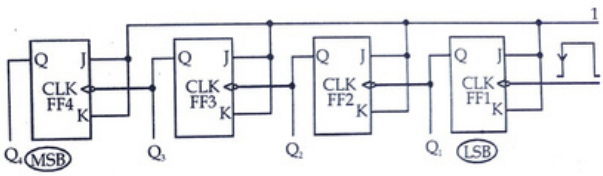
Type	Total counts or states	Number of flip-flop stages	Highest decimal counterpart
mod-2^n	2^n	n	$2^n - 1$
mod-2	2	1	1
mod-4	4	2	3
mod-8	8	3	7
mod-16	16	4	15
mod-N	N	n ($2^n > N$)	N–1
mod-3	3	2	2
mod-5	5	3	4
mod-6	6	3	5
mod-7	7	3	6
mod-9	9	4	8
mod-10	10	4	9

17. Asynchronous mod-16 Ripple Counter

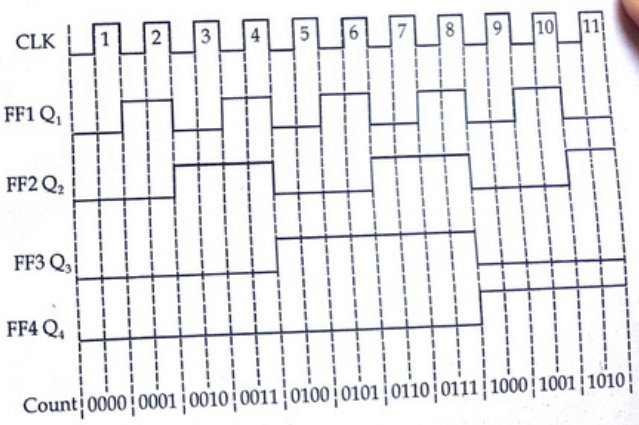
Ripple counters (not synchronized with the clock) are so called because these counters count like a chain reaction that ripples through the counter because of the involvement of the time. The flip-flops in these counters are cascaded in series and output of each is connected to input of the other flip-flop. The clock is connected to first-flop, and hence each flip-flop triggers when required pulse signal reaches to it. These counters are known as ripple counters because effect of first clock pulse is available first to FF1, and this effect cannot be felt by FF2 because of propagation delay through FF1. Thus the effect of clock pulse ripples through the counter, takes some time , due to propagation delay. This cumulative delay of ripple counter is a

major disadvantage and creates decoding problems; hence the cumulative delay must be less than the period of the input clock pulse.

This will become more evident with the explanation using the following circuit, Fig. 5.22 (a), shows a basic four-stage, or modulo-16, ripple counter. The inputs and output waveforms are shown in Fig. 5.22 (b). The four flip-flops are connected to perform a toggle operation ; and used to divides the input by a factor 2 by each flip-flop. The 1s on the J and K inputs enable the flip-flops to toggle. The inverters on the clock inputs indicate that the flip-flops change state on the negative-going pulse.



(a) Counter circuit diagram



(b) Timing diagram

Clock Pulse	Q_4	Q_3	Q_2	Q_1
Initially	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	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
16	0	0	0	0

(c) Truth Table

Fig. 5.22 : Four-stage ripple counter : (a) Counter circuit diagram ; (b) Timing diagram and (c) Truth Table

Working :

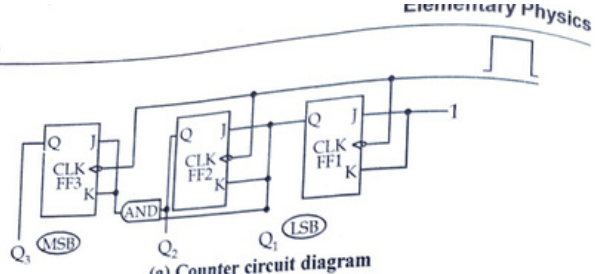
1. Assume that initially all the flip-flops are reset and have outputs Q_1, Q_2 Q_3 and Q_4 equal to 0 respectively for FF1, FF2, FF3 and FF4 and the count indicated will be 0000 by the counter. A bubble indicate at the clock input shows that all flip-flops will trigger at the falling edge of the clock pulse.
2. The negative-going pulse of clock pulse 1 cases FF1 to set and hence the output Q_1 of the flip-flop changes to state 1 and counter counts 0001.
3. The negative-going pulse of clock pulse 2 toggles FF1, causing it to reset and hence the output Q_1 of the flip-flop changes from stage 1 to 0. This negative going input to FF2 causes it to set and hence the output Q_2 of the flip-flop changes to state 1. Now after two clock pulse counter is showing 0010.
4. Clock pulse 3 causes FF1 to set again hence the output Q_1 of the flip-flop changes from state 0 to 1. This setting of FF1 does not going to affect the output of FF2, hence the output Q_2 of the flip-flop remains to state 1. So after three clock pulses, the counter indicates count 0011.
5. Clock pulse 4 causes FF1 to reset, which causes FF2 to reset, which causes FF3 to set, and counter now shows counts of 0100.
6. This setting and resetting of the flip-flops will continue until all the flip-flops are set and all the output are changes to state. 1. At that time the counter shows the counts 1111.
7. Now Clock pulse 16 will cause FF1 to reset and output Q_1 of the flip-flop changes from state 1 to 0. This will cause FF2 to reset, in order and the output Q_2, Q_3, and Q_4 changes from state 1 to 0 and now counter shows the count 0000 i.e. returns to its initial states.

Disadvantages :

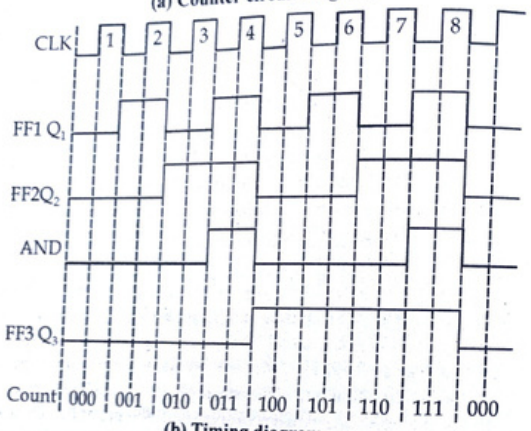
1. As the ripple count is asynchronous, it can produce errors in outputs when the clock speed is very high. This happens because a high-speed clock can cause the lower stage flip-flops to change state before the upper stages could react to the previous clock pulse.
2. They generate narrow spikes at the outputs.
3. They operate at lower speeds because of the cumulative propagation delay.

18. Synchronous mod-8 Counter

The word synchronous to the events that have a fixed time relationship with each other and as the name appears these counters are clocked at the same time by a common input clock pulse. For High-frequency operations it is required that all the flip-flops used to construct the counter must be triggered at the same time to prevent errors in indication the counts. For this type of operation we use this type of counter. The propagation delay is considerably reduced to the delay of only one flip-flop plus the delay time of the control gates and hence the operating speed of the counter is increased and this allows the use of clock pulses with higher frequencies. The synchronous counter is similar to a ripple counter with two differences: first is all flip-flops are simultaneously clocked and second is to ensure the toggling of flip-flops in proper sequence additional gates are used. A logic diagram of a three-stage (modulo-8) synchronous counter is shown in Fig. 5.23 (a). The clock input is connected to each of the flip-flops to prevent possible errors in the counts. A high (1) is connected to the J and K inputs of all the flip-flops to ensure the toggle mode of operation. The output of FF1 is wired to the J and K inputs of FF2 and to the one input of the AND gate. The output of FF2 is wired to the other input of the AND gate. The AND output is connected to the J and K inputs of FF3.



(a) Counter circuit diagram



(b) Timing diagram

Count	Q_3	Q_2	Q_1
000	0	0	0
001	0	0	1
010	0	1	0
011	0	1	1
100	1	0	0
101	1	0	1
110	1	1	0
111	1	1	1
000	0	0	0

(c) Truth Table

Fig. 5.23 : Three-stage synchronous counter: (a) Counter circuit diagram; (b) Timing diagram and (c) Truth Table

To understand the working of three stage synchronous counter consider the timing diagram and truth table.

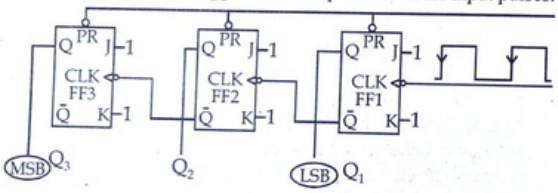
Working :

1. Assume the following initial conditions: The outputs of all the flip-flops, the clock, and the AND gate are 0 and the J and K inputs to FF1 are high (1). The negative-going portion of the clock pulse will be use throughout the explanation.
2. Clock pulse 1 causes FF1 to set and hence the outputs Q_1 of the flip-flip changes to state 1, indicating a binary count of 001. The 1 is also now applied to the J and K inputs of FF2 and one input of the AND gate. In this case the FF2 and FF3 are unaffected by the first clock pulse because the J and K inputs were 0 at the time of application of clock pulse.
3. As clock pulse 2 goes 0, FF1 resets, and hence the output Q_1 of FF1 goes to 0. In this case, FF2 will triggered, and hence the output Q_2 of FF2 goes to 1 and showing a count of 010. The 1 form FF2 is also applied to the input of AND gate, which is not activated at this time because the signal from FF1 is now a 0. Similarly a 0 is also present on the J and K inputs of FF3, so this clock pulse 2 is not going to trigger FF3.
4. Clock pulse 3 toggles FF1 again and makes the output Q_1 of the flip-flop changes to state 1. Since the J and K inputs to FF2 were 0 at the onset of clock pulse 3 which means FF2 does not toggle but remains at 1, hence indicating a count of 011. With both FF1 and FF2 showing output 1, 1s are input to both inputs of the AND gate, resulting 1 is applied to inputs J and K of FF3. But no change will be observed in the output of FF3 on clock pulse 3 because the J and K inputs were 0 at that time.
5. Just before the onset of clock pulse 4, we have the following conditions: FF1 and FF2 are at output state 1, and the AND gate is enabled and a 1 is applied to the J and K inputs of FF3. Hence all of the flip-flops will toggle with the next clock pulse. Hence at clock pulse 4, FF1 and FF2 are going to reset and have output 0, and FF3 will sets and hence the output Q_3 of FF3 will be 1. The output of the AND gate goes to 0, and counter indicates a count of 100. It appears that the clock pulse and the AND output both go to 0 at the same time, but the clock pulse arrives at FF3 before the AND gate goes to 0 because of the transit time of the signal through FF1, FF2 and the AND gate.
6. Between clock pulse 4 and clock pulse 8, FF3 remains set with output 1 because the J and K inputs are 0 and flip-flop will not toggle. On the application of clock pulse 5 and 6, FF1 and FF2 toggle in the same sequence as they did on clock pulses 1 and 2.

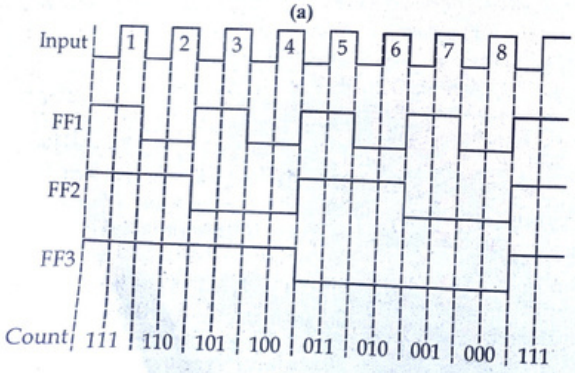
7. Clock pulse 7 results in all of the flip-flops being set and hence all the outputs Q_1, Q_2 and Q_3 will now 1 and the AND output being 1. Here counter indicate the count 111.
8. Clock pulse 8 causes all the flip-flops to reset and all the outputs will be 0, indicating a count of 000. The next clock pulse (9) will restart the counting sequence again.

18. Up/Down Counter

Till this section the counters that you have learned about are up counters, which counts in the upward sequence from lower counts (0) to higher counts to a given number. In this section we will discuss about the down counters, which counts in a downward sequence i.e. starts at a given higher counts and count down to lower counts (0). Up counter are sometimes termed as increment counters. Down counters are called decrement counters. A three-state asynchronous down counter is shown in Fig. 5.24 (a). It is clear from the Fig. 5.24 that PR (preset) input of the J-K flip-flops is used in making circuit of this counter. All the J and K inputs of flip-flops are connected to 1's to enable the toggle mode of operation on the input pulses.



(a)



(b)

Count	Q_3	Q_2	Q_1
111	1	1	1
110	1	1	0
101	1	0	1
100	1	0	0
011	0	1	1
010	0	1	0
001	0	0	1
000	0	0	0
111	1	1	1

(c)

Fig. 5.24 : A three stage asynchronous down counter : (a) circuit diagram; (b) Timing diagram and (c) Truth Table

Working :

1. Initially a negative-going pulse is applied to all PR terminals of J-K flip-flops to start the countdown. Because of this all the flip-flops are set and the outputs Q_1, Q_2 and Q_3 will be 1 and the count shown by the counter is 111. Similar instance of time, 0's are also applied to the CLK inputs of FF2 and FF3 by the $\overline{Q}$ outputs of FF1 and FF2, but they are not allowed to toggle as the preset input take lead over other inputs.
2. First clock pulse CLK1 causes FF1 to toggle and output Q_1 goes to 0. Notice that $\overline{Q}_1$ goes to 1 but no change occurs in FF2 or FF3 because this is a positive transition. Hence the output Q_2 and Q_3 remains at 1 and Q_1 goes to 0 and the indicated count is 110.
3. Second clock pulse CLK2 toggle FF1 again and its output Q_1 goes to 1. When Q_1 goes to 1, $\overline{Q}_1$ goes to 0. This negative-transition signal causes FF2 to toggle and reset and Q_2 goes to 0. A 1 is input at the CLK input of FF3 by $\overline{Q}_2$ going to 1 and hence FF3 remains in previous condition at 1 and hence the indicated count is 101.
4. At third clock pulse CLK3, FF1 toggles again and reset its output Q_1 to 0, which in turn makes $\overline{Q}_1$ to 1 and a positive-transition signal is applied to the CLK input of FF2, which makes output Q_2 to remain at 0 because of this Q_3 will also remain at 1. Hence the count shown by the counter is 100.
5. Fourth clock pulse CLK4 toggles FF1 and causes it to set, output Q_1 goes to 1. Because of this $\overline{Q}_1$ goes to 0 and hence a negative-transition signal is applied to the CLK input of FF2, which makes FF2 to toggle. This causes

FF2 to set and output Q_2 will goes to 1. Complementary output of FF2 $\overline{Q}_2$, then goes to 0, which causes FF3 to reset and its output Q_3 will go to 0. Hence the indicated count is now 011.

6. The next pulse, fifth clock pulse CLK5, toggle FF1 again and this make its output Q_1 to go to 0 and other FF2 and FF3 remain in their previous conditions as positive-transmission signal is applied at their clock input. The count is now 010.
7. The sixth clock pulse CLK6, FF1 toggles and its output Q_1 goes 1, which makes its complementary output $\overline{Q}_1$ to 0 and this negative-transition signal is applied to CLK input of FF2 and its output Q_2 will go to 0. Similarly for positive-transition signal at CLK input of FF3, makes its output Q_3 at 0. The count indicated by the counter is now 001.
8. Seventh clock pulse CLK7 toggles FF1 and its output will be 0. Now FF2 and FF3 will not toggle because of the application of positive-transition signal and hence the counter indicates 000.
9. On the negative-going signal of eighth clock plus CLK8, all flip-flops are set, and all the outputs Q_1, Q_2 and Q_3 go to 1. The CLK pulse toggles FF1, making output Q_1 go to 1. As output $\overline{Q}_1$ goes 0, the negative -transition signal causes FF2 to toggle. As FF2 toggles, output Q_2, goes to 1 and output $\overline{Q}_2$ goes to 0, causing FF3 to toggles and set. This makes output Q_3 to go to 1 and hence all the outputs are now at 1. The counter is now ready to again start counting down from 111 with the next CLK pulse.

19. 74 Series Counter IC's

We have discussed various counters up to last sections, which can be obtained using flip-flops. Here in this section we will discuss about some of the commercially available counter IC's, both in asynchronous and synchronous categories. Some of important IC's are enlisted in table 2 given below, categorised according to their features.

Table-2 : Some of the important commercially available counter IC's

Asynchronous counter IC's

S.No.	Counter IC No.	Counter Type	Features
1. 2.	7490 74290	BCD counter (decade type)	Set, Reset
3.	7492	Divide-by-12 (Mod-12)	Reset
4. 5.	7493 74293	Divide-by-16 Counter (Mod-16)	Reset

6. 7.	74176 74196	Presettable Decade (BCD) counter	Preset, load, Clear
8. 9.	74177 74197	Presettable divide-by-16 (mod-16) counter	Reset, load, Clear
10.	74390	Dual Decade counter	Reset
11.	74393	Dual divide-by-16 (mod-16) counter	Reset
12.	74490	Dual Decade (BCD) Counter	Set, Reset
Synchronous Counter IC's			
13. 14.	74160 74162	Presettable Decade (BCD) up-counter	Preset, clear, load
15. 16. 17.	74161 74163 74168	Presettable divide-by-16 (Mod-16) up-counter Decade (BCD) Up-down counter	Preset, clear, load Preset
18.	74169	Divide-by-16 Up-down counter	Preset
19.	74190	Presettable decade (BCD) up-down counter	Preset
20.	74191	Presettable divide-by-16 up-down counter	Preset
21.	74192	Decade (BCD) up-down counter	Preset, clear
22.	74193	Divide-by-16 (mod-16) up-down counter	Preset, clear

All of the above mentioned IC's consists of four flip-flops and synchronous counter IC's are positive-edge triggered i.e. change of states, synchronous loading and clearing take place on the positive going edge of the input clock pulse, whereas asynchronous counter IC's are negative-edge triggered.

20. 74 series asynchronous counter IC's

IC-7490

The asynchronous counter IC 7490 is a decade counter whose internal structure is shown below in Fig. 5.25. It actually consists of a single flip-flop and a 3-bit asynchronous counter i.e. total four flip-flops are used. Basically it consists of two separate sections, a divide-by-2 (mod-2) counter and a divide-by-5 (mod-5) counter. These two counters can be used to perform the operation of separate mod-2, mod-5 and mod-10 (in combination) counters.

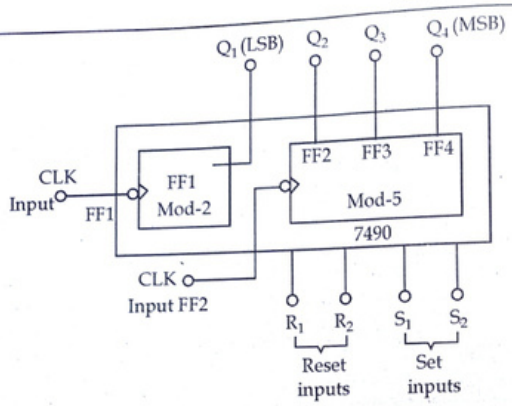


Fig. 5.25 : Internal structure of asynchronous decade (BCD) counter IC 7490.

There are two clock inputs for this counter IC, one for flip-flop (FF1) and another for flip-flop (FF2). It also consists of two reset inputs R_1 & R_2, which are kept high (at 1) for clearing all the flip-flops. There are two set inputs S_1 & S_2, which are when kept at high (1), set the counter for count 1001.

7490 as a divide-by-10 (decade) counter

When the outpout Q_1, is connected to input of the second flip-flop (FF2), and the clock pulse is applied to input of the first flip-flop (FF1). Here Mod-2 counter is followed by mod-5 counter and the counter IC 7490 progresses in natural binary counts and this forms a normal decade counter.

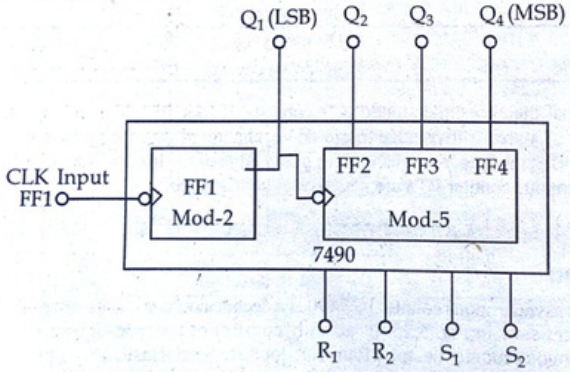


Fig. 5.26 : IC 7490 as a divide-by-10 counter.

Fig. 5.26 shows the IC-7490 implementation as a divide-by-10 (decade) counter. The counting sequence shown by this counter is given in table 3.

7490 as divide-by-6 (mod-6) counter

When Q_1 input of FF1 is connected to input of FF2, this IC counter behaves as divide-by-10 counter. For counting up to 6(0110), the outputs Q_2 & Q_3 should be reset to 0 to obtain a count 0000, and the counter reset to its initial position. Hence outputs Q_2 and Q_3 are connected to reset inputs R_1 and R_2, then counter can go up to 0101, after which it reset back to 0000 and hence it will act as a divide-by-6 counter.

Table : 3 Counting sequence for IC-7490 as a decade counter

Clock	Mod-5			Mod-2
pulse	Q_4	Q_3	Q_2	Q_1
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	0	0	0	0

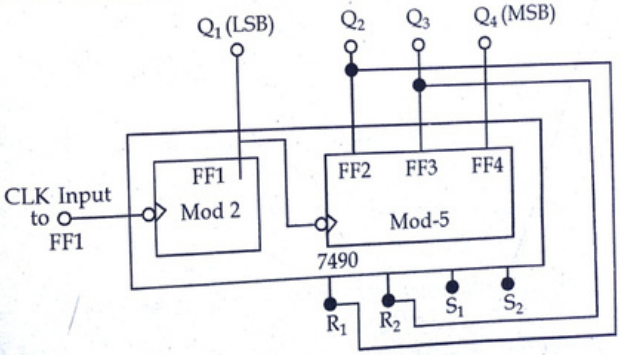


Fig. 5.27 : IC-7490 implementation as a divide-by- counter.

Fig. 5.27 shows the implementation of IC-7490 as a divide-by-6 counter, here outputs Q_2 and Q_3 are reset to 0 to stop the counting sequence at 0101. Table 4 below shows the counting sequence of the counter.

Table 4 : Counting sequence of IC counter 7490 as a divide-by-6 counter

Clock pulse	Mod-5			Mod-2
	Q_4	Q_3	Q_2	Q_1
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	0	0	0

IC-7492

This IC counter operates similarly as IC-7490, except that it does not have the set inputs (S_1 & S_2), it is a divide-by-12 counter and have two separate sections of mod-2 and mod-6 counters. Fig. 5.28 shows the internal structure of this IC-7492.

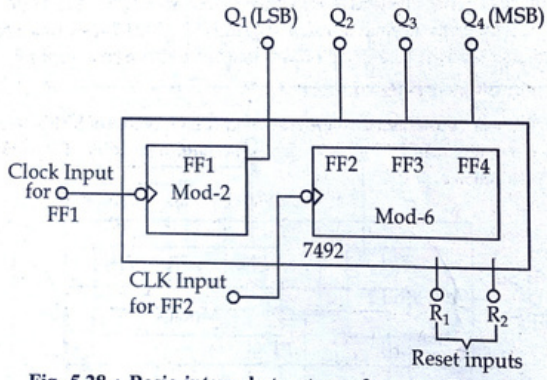


Fig. 5.28 : Basic internal structure of counter IC-7492

This counter IC can be implemented as a divide-by-12 counter, if output Q_1 of FF1 is connected to input of flip-flop FF2. Also simultaneous division of 2,6,12 are performed at its outputs Q_1, Q_2 and Q_4 respectively by this counter.

IC-7493/74293

This counter IC's actually consists of a single flop-flow (Mod-2) counter and a 3-bit (Mod-8) asynchronous counter. This IC counters can be used to perform the operation as divide-by-2 counter [if first section of mod-2 counter is used], as a divide -by-8 counter [if second section of mod-8 counter is used], and as divide-by-16 counter [in combination].

These counter IC's also have two gated reset inputs R_1 & R_2, which when high (1), the counter reset to its initial position 0000 state.

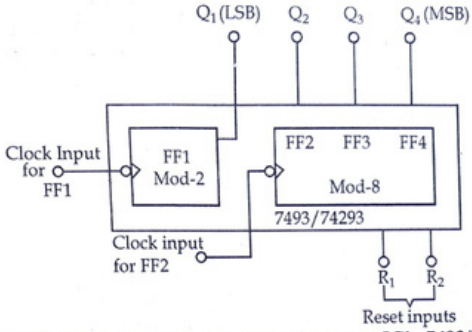


Fig. 5.29 : Basic internal structure of counter IC's 7493/74293

Fig. 5.29 shows the internal structure of counter IC's 7493/74293, which consists of two counter sections mod-2 and Mod-8 counters and two reset inputs R_1 & R_2.

7493 as a divide-by-16 counter

IC-7493 can be used as a divide-by-16 counter if Q_1 output of FF1 is connected to clock input of the FF2. Fig. 5.30 shows the implementation of IC-7493 as a divide-by-16 counter.

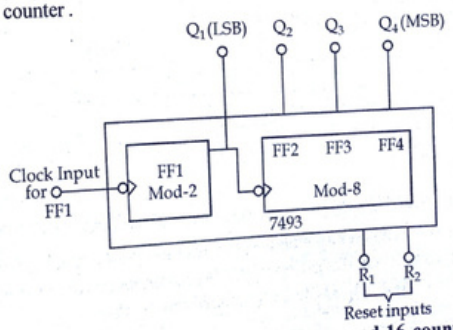


Fig. 5.30 : IC-7493 implementation as a mod-16 counter.

Table-5 shows the counting sequence for this counter IC 7493 implementation as a mod-16 counter.

Table 5 : Counting sequence for IC-7493 as a mod-16 counter

Clock pulse	Mod-5			Mod-2
	Q_4	Q_3	Q_2	Q_1
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	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
16	0	0	0	0

It is clear from the table that this counter goes form counting 0(0000) to 15 (1111) and then at sixteen clock pulse returns back to its initial state (0000).

This IC-7493 can also be implemented as mod-10, mod-12 counters etc. e.g. for implementation as a mod-10 counter, Q_2 and Q_4 outputs are connected to reset inputs R_2 and R_1 respectively. This resets the counter to 0000 state after having state 1010. Fig. (5.31) shows IC-7493 implementation as a mod-10 decade counter.

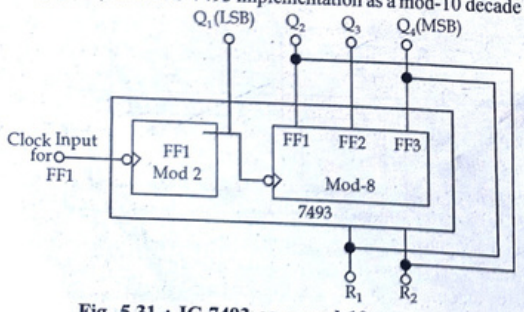


Fig. 5.31 : IC-7493 as a mod-10 counter

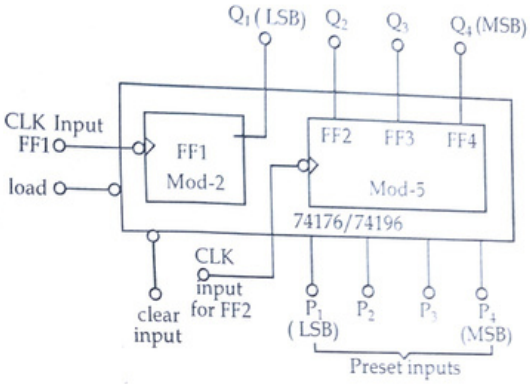


Fig. 5.32 : Internal structure of IC counter 74176/74196

These IC counters are the presentable versions of IC counter 7490. Both of these counters are BCD (decade) counter with two sections of mod-2 and mod-5 counters. These counters have clear, load and four preset inputs as shown in fig. 5.32. The counter is cleared by connecting a low (0) to clear input. At high (1) clear input, if load input is low (0), counting stops and any binary number present at four preset inputs can be loaded into the counter. For normal counting operation it is desired to have clear and load inputs at high (1).

IC-74177/74197

These counter IC's are the presettable version of the IC counter 7493. Both of these IC counters are mod-16 counter; consisting of two sections mod-2 and mod-8 counters. These counters also have clear, load and four preset inputs as shown in Fig. 5.33. These counters clear when clear is at low (0). When clear is at high (1) and if load input is low (0), counting stops and counter shows binary numbers at the preset inputs.

These counters can also be used as a variable mod-N counter where N have value equal to N = 15 - P, here P is the binary number at four preset inputs. If we apply preset input equal to binary equivalent of P, then this counter can work as a mod-(15-P) or mod-N counter. This is possible if a four input NAND gate is connected in between the outputs Q_1, Q_2, Q_3 and Q_4 and load input.

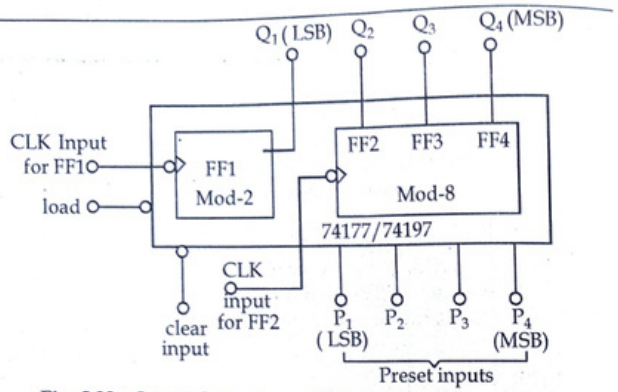


Fig. 5.33 : Internal structure of IC-74177/74197 counters

74177 as a divide-by-12 counter

We know to design a mod-N counter using IC-74177 counter we have to preset the IC with binary number P = 15 – N. Here we have to design a mod-12 counter so N = 12. Hence :

$P = 15 - 12 = 1111 - 1100 = 0011$

so preset inputs are $P_1 = 1$, $P_2 = 1$, $P_3 = 0$ and $P_4 = 0$ and as counter output become 1111, it will loaded with preset output 0011.

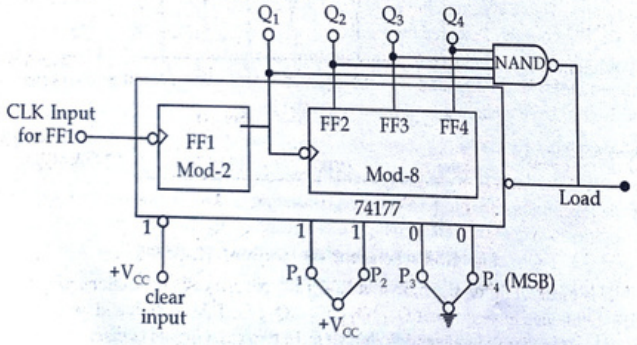


Fig. 5.34 : IC-74177 implementation as a divide-by-12 counter.

To load preset count 0011 into the counter the load signal should go low (0) at the highest count 1111, for this we use a NAND gate as shown in the Fig. 5.34.

The twelve counting sequence for this mod-12 counter will be:

Preset states

0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

Where arrow indicate that as counter go to the highest state 1111, it will return back to the preset state 0011 and hence counting sequence for this counter will be from 0011 to 1110; twelve in total number.

IC-74390

The IC-74390 is a dual BCD counter BCD counter, which have two mod-10 units, where each unit in itself or in combination with the there one, can be used to obtain higher moduli. Its internal structure is shown in Fig. 5.35.

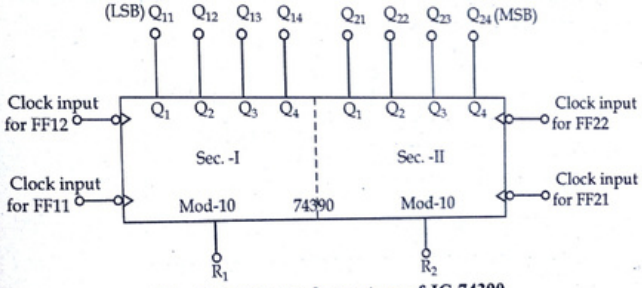


Fig. 5.35 : Internal structure of IC-74390

It is clear from the fig. 5.35 that IC-74390 consists of two sections of mod-10 each and hence have outputs Q_{11}, Q_{12}, Q_{13}, Q_{14} for first section and outputs Q_{21}, Q_{22}, Q_{23}, Q_{24} for second section. It also have two reset inputs to reset the counter.

IC-74393

This counter IC-74393 is the dual unit, which consists of two sections of mod-16 (divide-by-16) counters. Here each unit itself or in connection with other unit,

can be used to obtain counters of higher moduli. As IC-74390 it also have two reset inputs to reset the counter except its internal structure consists of two sections of mod-16 counter whereas IC-74390 have two sections of mod-10 counter.

IC-74490

It is also a dual units consists of two sections of mod-10 (BCD) counters. In this regard this IC is similar to IC-74390 except one difference that this IC-74490 have two set inputs also in its internal structure.

21. 74 Series synchronous counter IC's

It is already discussed that all synchronous IC counters are positive-edge triggered and each of them responds to the positive going edge of the input pulse.

IC-74160/74162

These counter IC have several features in addition to the basic functions previously discussed for synchronous binary counters.

These counter IC's are of decade (BCD) up-counter, which can be synchronously preset to any binary member less than 1001, by applying the proper levels to the preset inputs P_1, P_2, P_3 and P_4. These counter IC's have two enabled inputs, ENP and ENT. These inputs must be high (1) for the counter to sequence through its binary states. When any one of them is low (0), the counter becomes disabled asynchronously. Ripple clock (RC) output goes to high (1) when the counter attain its highest count (here it is 1001), else this output (RC) is normally low. It also have one more input load, when a low (0) is applied to it, the counter will take the state of preset inputs on the next clock pulse. Hence the counting sequence can be started with presettable binary number.

Fig. 5.36 shows the block diagram of IC-74160/74162.

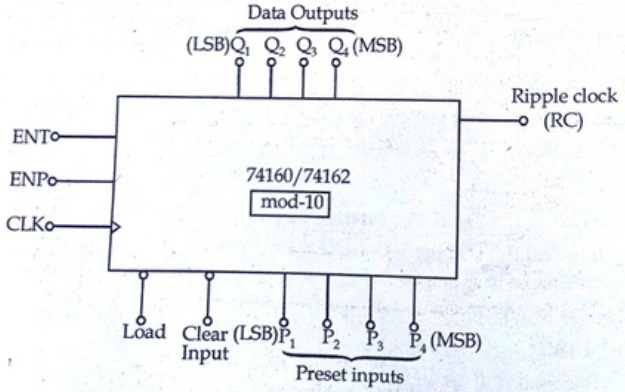


Fig.5.36 : **Block diagram of IC-74160/74162 synchronous counter**

Both of these IC's can be used to obtain a counter of any modulus less than 10. The RC output, with ENT and ENP enable inputs, allows these counters to be cascaded for higher count sequences.

IC-74161/74163

In these IC's there are two enable inputs, ENT and ENP, one clear input, one load inputs, four presettable inputs and one ripple clock (RC) output similar to IC-74160/74162, where function of each input is similar to IC-74160/74162. The difference from IC-74160/74162 is that these counter IC's are a mod-16 (UP-) counter, and can be used to make a counter with modulus less than 16. Fig. 5.37 shows the block diagram of these IC-74161/74163.

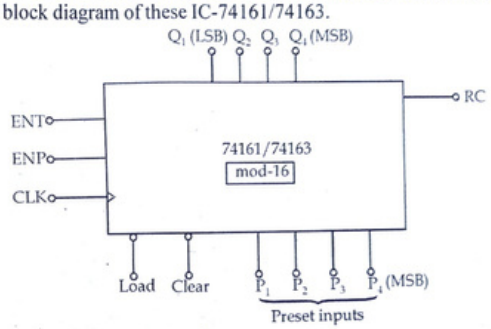


Fig. 5.37 : Block diagram of IC-74161/74163

74163 as a binary mod-12 counter

We already know for a mod-12 counter the terminal count is 1011 (11_{10}) so the states of the counter must be from 0000 to 1011 and as soon as the output will be 1100 it will return to 0000 state. For this outputs Q_3 and Q_4 will be connected to a NAND gate and then connected to the clear input, which clear the counter as soon as the output is 1100.

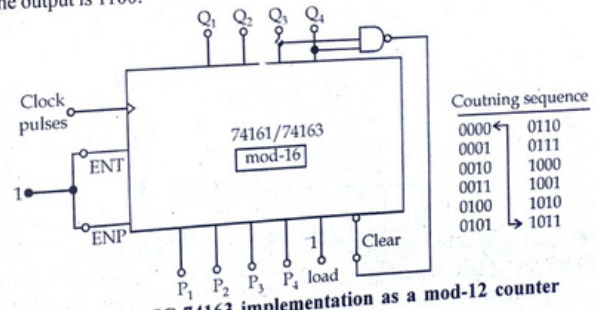


Fig. 5.38 : IC-74163 implementation as a mod-12 counter

It is clear from the Fig. 5.38 that using the IC-74163, the count can be terminated at any desired value and hence a counter with any modulus (less than 16) can be obtained. In general, for obtaining a mod-N (less than counter, the preset input P is given by P = 16-N in binary from should be given at preset inputs.

IC-74168/74169

Each of IC-74168/74169, as given by the block diagram in Fig. 5.39, is a synchronous UP/Down counter with two enable inputs ENT and ENP same as discussed for IC's 74160/74161 etc. but here these inputs are active-low. Ripple clock (RC) output is normally held high (1) and goes to low (0) when (i) highest count is reached for up counting and (ii) lowest count (0000) is reached for down-counting. These counter have an U/$\overline{D}$ U/$\overline{D}$ = 1 for up counting and U/$\overline{D}$ = 0 for down counting.

No clear terminal is available in these types of IC's and hence to terminate the count, a NAND gate is connected in between the counter outputs and load. The preset inputs can be given to start the counting with the required starting state.

The counter IC's-74168/74169 are similar in basic structure as discussed above except that IC-74168 is a mod-10 (Decade) counter and IC-74169 is a mod-16 (4-bit) counter.

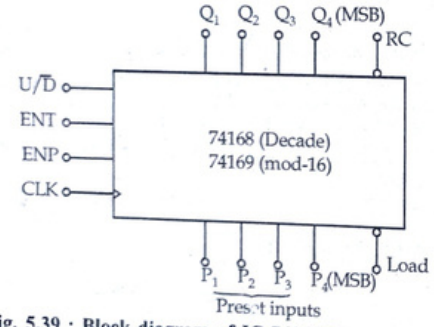


Fig. 5.39 : Block diagram of IC-74168/74169 counters

IC-74190/74191

These IC counters have only one enable input EN, which is active low. These are also UP/Down synchronous counters as shown in the block diagram of Fig. 5.40. The IC-74190 counter is a decade counter and IC-74191 is a mod-16 counter.

These counters have most of all same input and output terminals, which IC-74168/74169 posses. The RC output is normal held high (1) goes low (0), when counter attain the highest (1001 for 74190 and 1111 for 74191) count for up counting and the lowest count (0000) for down counting with the clock input as low.

These counters have one more output terminal MAX/MIN output, normally at 0 and goes to high (1) when the terminal count is reached ; maximum for up counting and minimum (0000) for down counting.

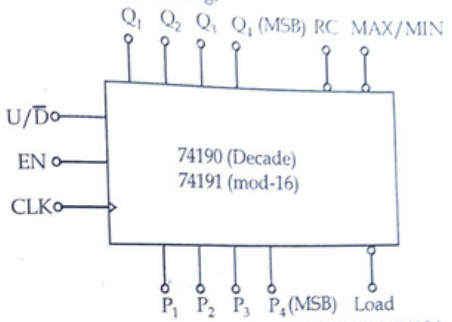


Fig. 5.40 Block diagram of synchronous IC-74190/74191 counters

Frequency division using IC-74190/74191

To use these counter as a frequency divider, the RC output is connected to the load and the required frequency division is obtained at the MAX/MIN output terminal in the form of frequency divided output waveform. When the preset inputs are set at a binary (equivalent decimal N) number, the frequency waveform (f_{out}) and the frequency of the input-clock (f_{in}) are related with each other as shown by the relations. For IC-74190 counter (mod-10).

$$f_{out} = \frac{f_{in}}{N} \; ; 1 \le N \le 9 \text{ (Down counting)}$$

$$= \frac{f_{in}}{9-N} \; ; 1 \le N \le 8 \text{ (Up counting)}$$

For IC-74191 counter (mod-16)

$$f_{out} = \frac{f_{in}}{N} \; ; | \le N \le 15 \text{ (Down counting)}$$

$$= \frac{f_{in}}{15-N} \; ; | \le N \le 14 \text{ (Up counting)}$$

IC-74192/74193

Unlike other IC-counters, these counters have two clock inputs ; one for up counting CLK-Up terminal and another for down counting CLK-Down terminal. For up counting clock pulse is applied at CLK-Up terminal and for down counting clock pulse is applied to CLK-Down terminal while in both cases other terminal is kept high (1).

These IC's have two more outputs : carry and borrow outputs, normally kept at high (1). For up (down) counting, when the count reaches maximum (minimum), the carry (Borrow) output goes to low (0) and borrow (carry) output remains at high (1).

The block diagram for these counter is shown in Fig. 5.41, where IC-74192 is a decade (BCD) counter and IC-74193 is a mod-16 counter.

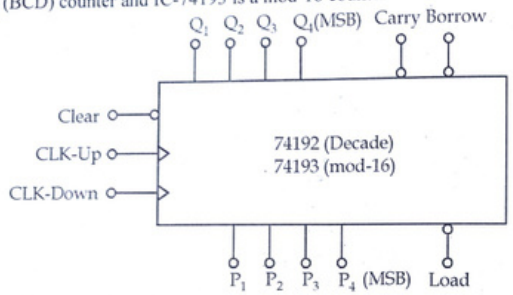


Fig. 5.41 : Block diagram of IC-74192/74193

Frequency division using IC-74192/74193

These counter IC's can also be used for frequency division in similar manner to the one used for IC-74190/74191 except one difference that here carry (or borrow) output is to be connected to the load input for up (or down) counting.

22. Applications of digital counters

The digital counters are very useful and versatile devices. These counters are used in various applications in digital computers, data processing systems and industrial control systems etc. Some of these applications are :

1. **Digital clock :** A counter application is in time keeping systems. This digital clock displays seconds, minutes and hours.
2. **Direct counting :** These counters are very useful in direct counting applications.
3. **Frequency dividers :** As we have seen that these counters can also be used for dividing the frequency.
4. Measurement of frequency and the time of any event.
5. Automobile (car) parking control.
6. **Multiplexing :** Digital counters are also useful for parallel-to-serial data conversion.
7. Counting the sequence of operations in a digital computer.

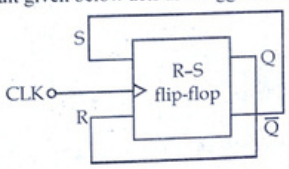
Questions

Very Short Answer Type Questions

1. What is sequential circuit ?
2. Write the difference between combinational and sequential circuit.
3. What is an intermdiate state in R-S Flip Flop.
4. What is a D Flip-Flop ?
5. Draw truth table of D Flip-Flop.
6. Draw diagram of positive Edge-Triggering.
7. What are the advantage of master slave technology ?
8. What do you mean by edge-triggering ?
9. Draw block diagram of R-S Flip-Flop ?
10. What is the clocked J-K Flip-Flop?
11. What are present and clear inputs for a Flip-Flop ?
12. Give the difference between positive and negative edge triggering.
13. What is a shift register ?
14. Name the different types of shifts registers.
15. What is a digital counter ?
16. Define modulus counter.
17. What is a synchronous counter?
18. What is the difference between Asynchrous counter and synchronous counter?
19. What is a Asynchronous counter ?
20. Draw the pin out diagram for IC 7493. **[Raj. 2008]**

Long Answer Type Questions

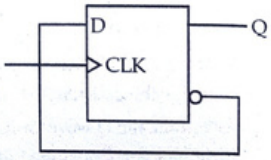
1. **In an RS flip-flop, if the inputs changes from :**
   (i) S = 1, R = 0 to S = R = 0 and
   (ii) S = 0, R = 1 to S = R = 0,
   Explain whether the outputs Q & $\overline{Q}$ change or not and why?
2. Verify that the circuit given below acts as a toggle switch.



3. Explain to convert a D-type flip-flop to act as a toggle switch and also verify truth table of the converted flip-flop.

[**Hint :-** $\overline{Q}$ output is connected to D input]

4. A D flip-flop is connected as shown in the Fig. given below. Explain its operation with clock pulse and also show what function this flip-flop performs.



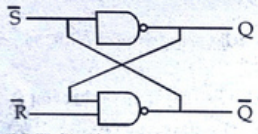
5. Make circuit for R-S flip-flop using NAND and NOR gates. Convert one of these circuit into clocked R-S flip and make table for this.

[Raj. 2007]

6. Make circuit for clocked J-K flip-flop and write truth table for this and explain it. What is the advantage offered by a clocked J-K flip-flop over clocked R-S flip-flop ? **[Raj. 2004, 2007]**

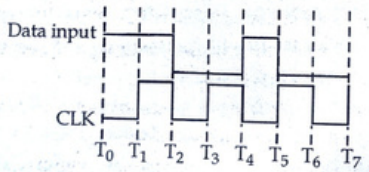
7. Show how to convert $\overline{RS}$ flip-flop in the below Fig. into an RS flip-flop.

[Raj. 2005]

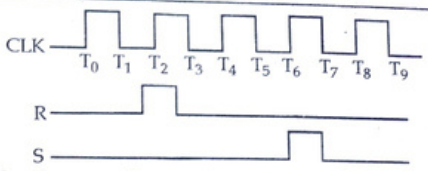


8. (a) What is the clocked J-K flip-flop ? Write truth table with circuit diagram.
(b) What are preset and clear inputs for a flip-flop ? **[R. U. BCA 2004]**

9. The data input and clock pulse are shown below. Compare the Q output for positive edge triggered, negative-edge triggered and level-triggered D-flip-flops. Assume that flip-flops are initially reset

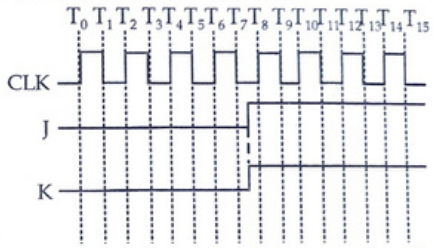


10. To drive a clocked R-S flip-flop, waveforms are shown below. If initially flip-flop is reset, explain the following :
(i) When does Q again Reset to 0 ?
(ii) At what time Q goto 1.



11. Describe the advantage of master slave technology. **[Raj. 2006]**

12. Determine the Q waveform if the signal shown below are applied to the inputs of the J-K flip-flop. assume that initially Q is 0.



13. What do you mean by edge-triggering ? How we can achieve positive and negative edge triggering?

14. Explain the truth table of a J-K flip-flop with its circuit diagram.

15. Implement J-K, D and T flip-flops using a R-S flip-flop by introducing some input-control logics.

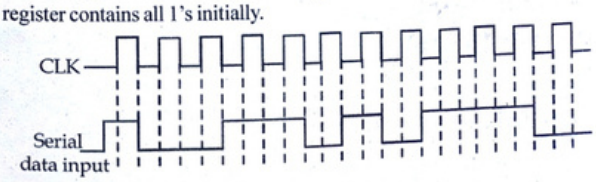
16. Construct a 5-bit shift register than can be used to any of the four modes of a shift register (SISO, SIPO, PISO, PIPO). **[Raj. 2007]**

17. Explain the process of entering of the digital word and reading of digital word in the shift-register. **[Raj. 2007]**

18. Give a schematic of a 4-bit serial-in-parallel out shift register and show the status of 4 stages for the data nibble 0 1 1 0.

19. Discuss various applications, of shift registers.

20. Determine the states of each flip-flop in serial in-serial out shift register and show the Q waveforms for the data input and clock shown below. Assume that register contains all 1's initially.



21. Show a timing diagram showing the parallel outputs for the serial-in parallel-out shift register. Use the waveforms in problem 20. For clock and data input with the register initially clear.
22. What modulus counter can be constructed with the use of four flip-flops ? **[Raj. 2006]**
23. Differentiate between asynchronous and synchronous counters. **[Raj. 2006]**
24. Explain the working of a 3-bit binary down asynchronous counter with its count sequence. **[Raj. 2006]**
25. Construct the following ripple counters using J-K flip-flops :
    (1) Divide-by-6 (Mod-6)
    (2) Divide-by-9 (Mod-9)
26. Describe a 3-bit synchronous counter with its waveforms and counting sequence. How it is different to an asynchronous counter with same moduli ?
27. For a presettable 8-bit counter, what number would you preset to get a divide-by-118 counter ?
28. A presetable counter with 8 flip-flops is preset with a number 120, Find the modulus of the counter.
29. If Q_4 output of IC-7490 is connected to input of flip-flop FF1 and clock pulses are applied at input of FF2, show the count sequence and waveforms of output Q_1.
30. Show how a IC-74293 can be used as a mod-15 counter.
31. Show IC-7493 implementation as a mod-12 counter with its counting sequence.
32. Show IC-74177 implementation as a mod-11 and a mod-7 counter. Also show its counting sequence.
33. Show using IC-74163 a mod-9 counter can be obtained. Make use of the Ripple Clock (RC) output and preset inputs.
34. Obtain a mod-11 counter using IC-74161 using preset input and ripple clock (RC) output.
35. Show how a IC-7493; 4-bit asynchronous counter can be used for each of the following moduli :
    (a) 10 (b) 13 (c) 14 (d) 15
36. Obtain a mod-12 up counter using IC-74193.
37. Construct mod-12 ripple counter ring IC-7492 and show that the frequency divisions of 3, 6 and 12 are obtained at the outputs Q_2, Q_3 and Q_4 respectively.
38. Construct an mod-50 ripple counter and also tell what flip-flop outputs should be connected to the clearing NAND gate to form it.

■■■