

4.4 SEQUENTIAL CIRCUITS

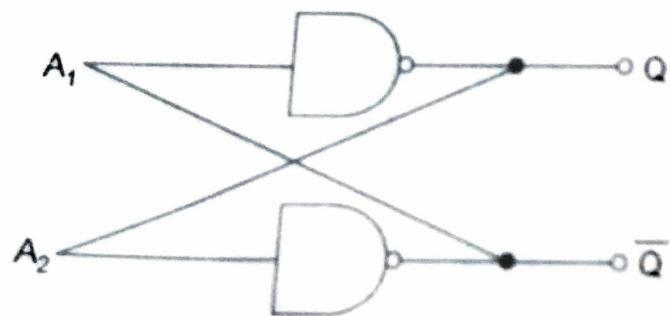
4.4.1. MEMORY CELLS AND FLIP-FLOPS

Basic Memory Cell

A basic memory cell is a circuit that stores one bit information. This one-bit memory element is called a flip-flop or a latch, since it latches (or locks) data in it.

It has two outputs, Q and \bar{Q} , that are always complements of each other. In set state, Q is HIGH (logic 1) and \bar{Q} is LOW (logic 0); whereas in reset state \bar{Q} is HIGH (logic 1) and Q is LOW (logic 0). For a flip-flop to act as a memory device, it should retain the information stored in it. Thus, if it is set state it should remain set and if it is in reset state, it should remain reset.

This function can be realized by using two NAND or NOR inverters connected in a feedback loop as shown in Fig. 4.41.



$$Q = \bar{A}_1 = A_2$$

$$\bar{Q} = \bar{A}_2 = A_1$$

Q	A_2	\bar{Q}	A_1	Q
1	1	0	0	1
0	0	1	1	0

Fig. 4.41 : Basic memory cell

SR CIRCUITS

4.4.2. SR FLIP-FLOP The purpose of storing desired bits in the flip-flops, two-input NAND or NOR gates are used as shown in Fig. 4.42. The following observations are made from this figure and the truth-table.

Truth-table				
S	R	A ₁	A ₂	State
0	0	1	1	No change
0	1	1	0	Reset
1	0	0	1	Set
1	1	0	0	Race

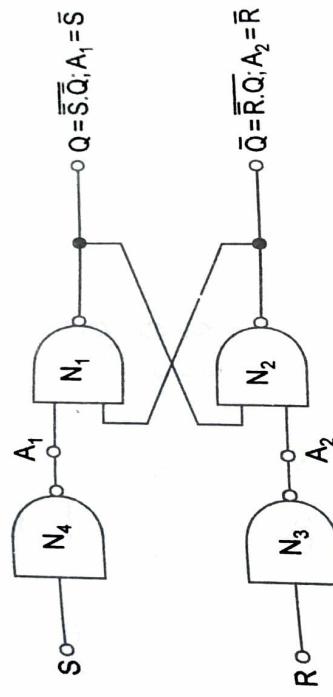


Fig. 4.42 SR flip-flop.

Set State : When $S = 1$ and $R = 0$, A_1 becomes 0, making $Q = 1$. So, $\bar{Q} = \overline{1.1} = 0$. This input condition sets the flip-flop ($Q = 1$, $\bar{Q} = 0$) ; so it is called **set state**.

Reset State : When $S = 0$ and $R = 1$, A_2 becomes 0, making $\bar{Q} = 1$ and $Q = \overline{1.1} = 0$. This input condition resets the flip-flop ($Q = 0$, $\bar{Q} = 1$) ; so it is called **reset state**.

No Change : When $S = R = 0$, $Q = 1$. $\bar{Q} = 1$. $\bar{Q} = Q$ and $\bar{Q} = \overline{Q}$. The flip-flop remains in whatever state it is in, the function is that of a basic memory cell discussed earlier.

Race : When $S = R = 1$, A_1 and A_2 both become 0. So $Q = \bar{Q} = 1$. This is an undesired output state of the flip-flop depends on the relative delays of the two gates. If N_2 is faster, \bar{Q} will become $\overline{1.1} = 0$ first and if S and R now change to 0, A_1 and A_2 will be 1, both Q and \bar{Q} will try to become 0. The actual state of the flip-flop is uncertain, so this condition is not allowed.

This circuit is called an SR flip-flop. It is an asynchronous circuit because the output changes with changes in the inputs S and R. This is called race condition. Here the state of the flip-flop is uncertain, so it is called race condition. Here the state of the flip-flop depends on the relative delays of the two gates. If N_1 is faster, Q will become 0 first and will make $\bar{Q} = 1$. Similarly, if N_1 is faster, Q will become 0 first and will make $\bar{Q} = 1$. This is called race condition. Here the state of the flip-flop is uncertain, so this condition is not allowed.

4.4.3 CLOCKED SR FLIP-FLOP

The SR flip-flop modified to include clock (CK) pulses is shown in Fig. 4.43.

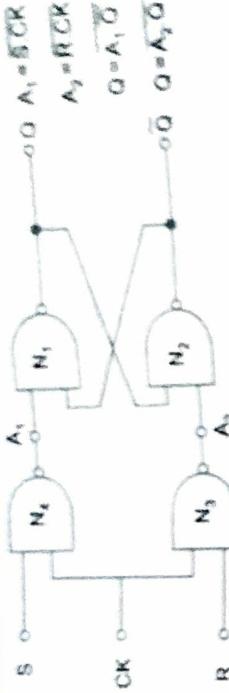


Fig. 4.43 Clocked SR flip-flop.

When $CK = 1$, $A_1 = \bar{S}$ and $A_2 = \bar{R}$, the circuit functions like the SR flip-flop. When $CK = 0$, $A_1 = A_2 = 1$, the circuit reduces to that of a basic memory cell and outputs remain unchanged (latched). Here N_1 and N_2 form the basic latch, whereas N_3 and N_4 control the state of the flip-flop. The truth-table and logic-symbol of a SR flip-flop are as shown in Fig. 4.44.

Truth-table

CK	S R		Q_{n+1}	(SR enabled)
	1	0	0	
1	0	1	0	
1	1	0	1	Not allowed
1	1	1	1	Q_n (SR disabled)
0	x	x	x	

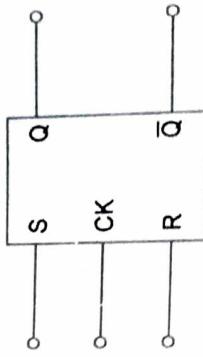


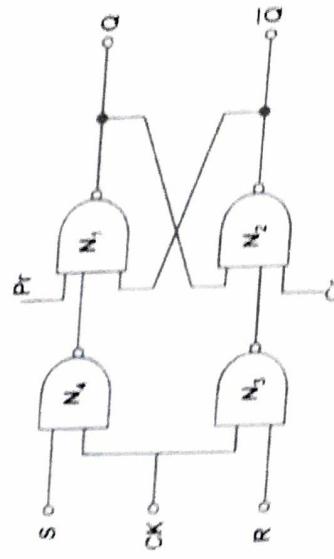
Fig. 4.44 Symbol clocked SR.

4.4.4 TRIGGERING

In level triggering, the output changes only when the clock is in HIGH or LOW level. In the clock positive level triggering. If the clock pulse is inverted before applying it at the input, the output change only when the clock is LOW. This is called negative level triggering. In edge triggering, the changes only when the clock pulse makes a transition (from LOW to HIGH or from HIGH to LOW).

4.4.5. PRESET AND CLEAR

When power is turned on, flip-flop assumes a random state depending upon the time delay of the inputs called preset and clear as shown in Fig. 4.45. These inputs are applied in the interval between



Example

If $CK = 0$, When $Pr = 0$ and $Cr = 1$, Q is 1, so $\bar{Q} = 1, 1, 1 = 0$; the flip-flop is set.

Assume $CK = 0$, $Pr = 1$ and $Cr = 0$, \bar{Q} is 1 and $Q = 1, 1, 1 = 0$; the flip-flop is in reset state or cleared.

When $Pr = 1$ and $Cr = 1, \bar{Q}$ is 1; the flip-flop is in reset state or cleared.

When Pr and Cr are both 1, the circuit functions like normal clocked bit flip-flop.

When $Pr = 0$, both Q and \bar{Q} will become 1, which is not desired. So $Pr = Cr = 0$ is not allowed.

If $Pr = Cr = 0$, applied direct, not in synchronism with the clock pulse. Thus, they are asynchronous Pr and Cr inputs and are also called direct set and direct reset respectively. Since the desired function is performed by corresponding input is Low ($Pr = 0$, $Cr = 1$ sets and $Cr = 0$, $Pr = 1$ clears the flip-flop); they are inputs when the low inputs (indicated by placing a bubble at those inputs). The symbol of a clocked SR flip-flop and its truth-table with active low preset and clear is shown in Fig. 4.46.

Truth-table

CK	Pr	Cr	S	R	Q_{n+1}
1	1	1	0	0	Q_n
			0	1	0
			1	0	1
			1	1	Not allowed
0	0	1	x	x	1 (Set)
0	1	0	x	x	0 (Reset)

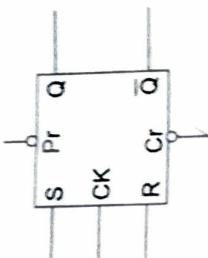


Fig. 4.46 Symbol clocked SR with preset and clear

4.4.6. JK FLIP-FLOP

In an SR flip-flop, the input combination $S = R = 1$ is not allowed because its output is uncertain.

Let us call this as JK flip-flop with inputs J and K. Also, Table 4.9 gives the combined truth-table of SRJK flip-flops.

Table 4.9. Truth-table SR/JK Flip-flops

J	K	Q_n	Q_{n+1}	S	R	
0	0	0	0	0	0/1	Inactive
0	0	1	1	1/0	0	$Q_{n+1} = Q_n$
0	1	0	0	0	0/1	Reset state
0	1	1	0	0	1	$Q_{n+1} = 0$
1	0	0	1	1	0	Set state
1	0	1	1	1/0	0	$Q_{n+1} = 1$
1	1	0	1	1	0	Toggle state
1	1	1	0	0	1	$Q_{n+1} = Q_n$

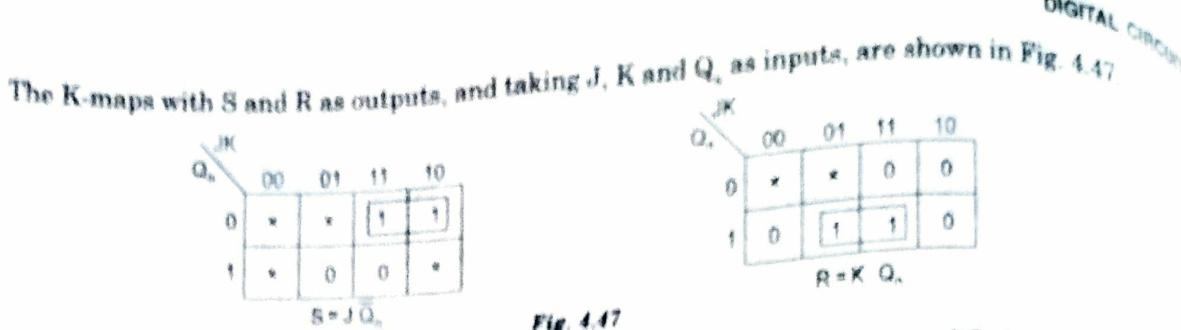


Fig. 4.47

Based on K-map JK-FF has been designed as shown in Fig. 4.48 and its qualified circuit diagram in Fig. 4.49.

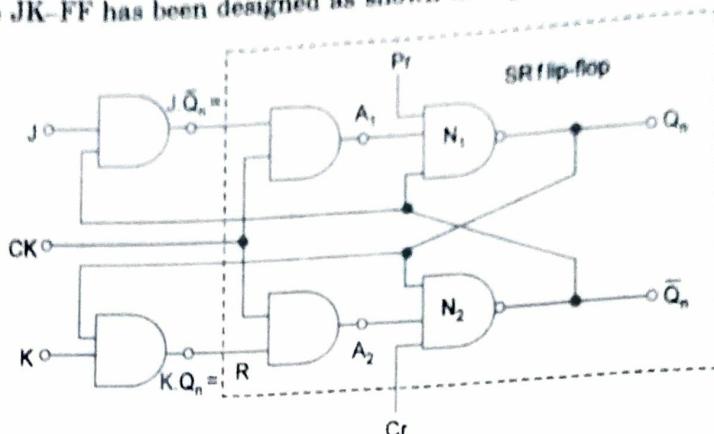


Fig. 4.48

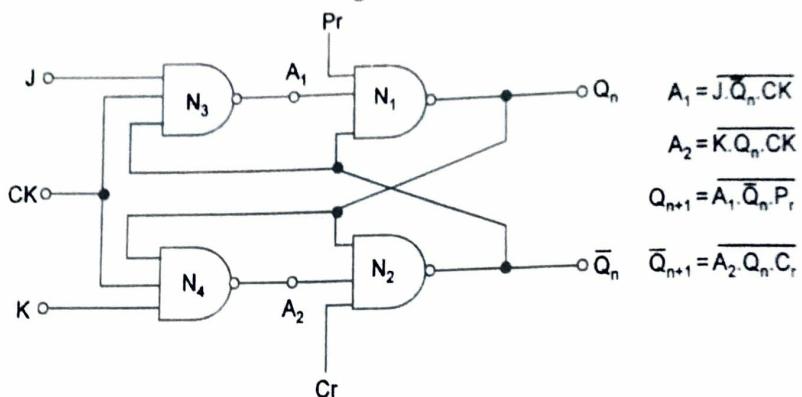


Fig. 4.49

Table 4.10 : Truth-table JK Flip-flop

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	\bar{Q}_n

4.4.7. JK MASTER-SLAVE FLIP-FLOP

This flip-flop is made of two SR flip-flops. The output of the first, called the *master flip-flop*, is given to the inputs of the second, called the *slave flip-flop*. The output of the slave is fed back to the inputs of the master. The master is triggered by a positive clock pulse and the slave is triggered by a negative clock pulse, obtained by inverting the clock pulse applied to the master. The circuit of the flip-flop is shown in Fig. 4.50.

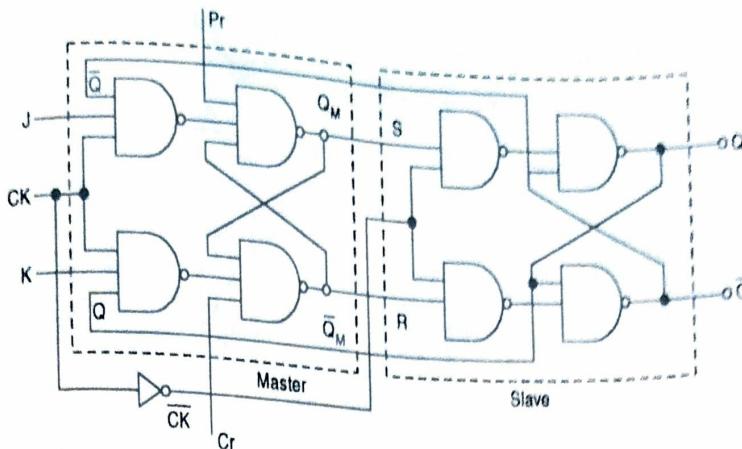


Fig. 4.50. JK master-slave flip-flop

When the clock is high ($CK = 1$, $Pr = 1$, $Cr = 1$), the master is enabled, while the slave is disabled as $\overline{CK} = 0$. So, the master functions like a JK flip-flop and its output appears at the input (S, R) of the slave. Since CK is low, the slave is inactive. Thus Q remains unchanged for the duration of the clock pulse t_p . When the clock goes low, the slave functions like an SR flip-flop and its output changes to Q_M' , which also appears at the input of the master. Since the clock is low, the master is inactive so that Q_M' (and so S and R) remains unchanged as long as the clock remains low.

Thus, when the clock is high, Q_M changes according to JK flip-flop logic and it is transferred to Q when the clock goes low (negative edge transition). This eliminates the race-around condition as the output remains constant for the duration of one clock pulse. This flip-flop is called as master slave flip-flop because the master decides the output, which is simply latched out by the slave. The truth-table of this flip-flop is shown in Table 4.11.

Table 4.11 : Truth-table for MS JK Flip-flop

Pr	Cr	CK	J	K	Q_{n+1}	States
0	0	x	x	x	?	Not desired
0	1	x	x	x	1	Preset
1	0	x	x	x	0	Clear
1	1	x	0	0	Q_n	Inactive
1	1		0	1	0	Zero
1	1		1	0	1	One
1	1		1	1	Q_n	Toggle

4.4.8. D FLIP-FLOP

In a D flip-flop the output follows the input whenever the flip-flop is triggered. This can be realized by means of a JK/SR flip-flop. The JK/SR inputs needs to achieve D logic are given in the truth table of Table 4.12.

Table 4.12 : Truth-table JKSR Flip-flop

D	Q_n	Q_{n+1}	J	K	S	R
0	0	0	0	0/1	0	0/1
0	1	0	0/1	1	0	1
1	0	1	1	0/1	1	0
1	1	1	0/1	0	0/1	0

The K-maps for J, K, S, R are given in Fig. 4.51 from which it follows that

$$D = J = \bar{K}$$

Also

$$D = S = \bar{R}$$

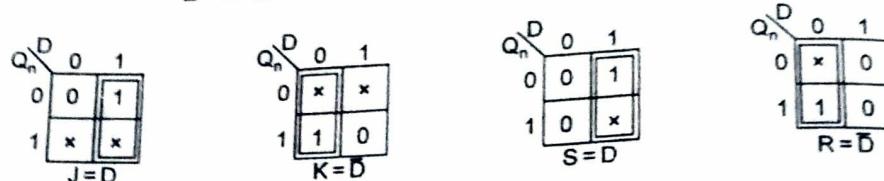


Fig. 4.51

Using these equations the design of a D flip-flop using JK flip-flop/SR flip-flop along with its truth-table as given in Figs. 4.52 (a), (b) and (c) respectively. Here the output follows the input, when the flip-flop triggered. If the flip-flop is edge triggered, the output is the same as the input, but is delayed by one clock pulse. Due to this property, the D flip-flop is used as a delay device. From figure 4.53, short duration spurious pulses in the datas (D) (called **glitches**) are eliminated in the output (Q).

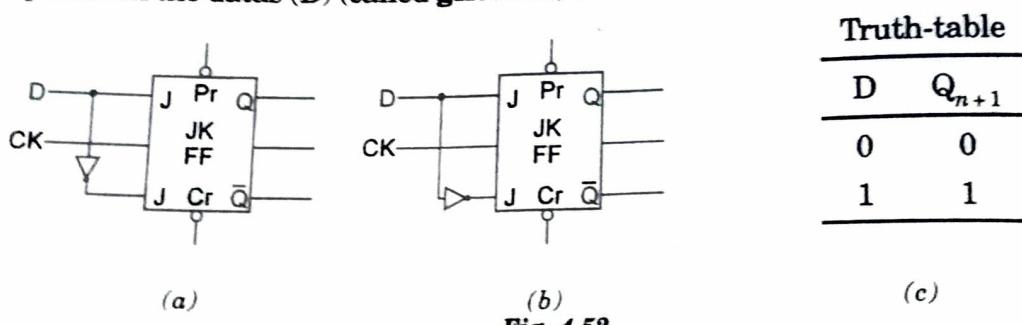


Fig. 4.52

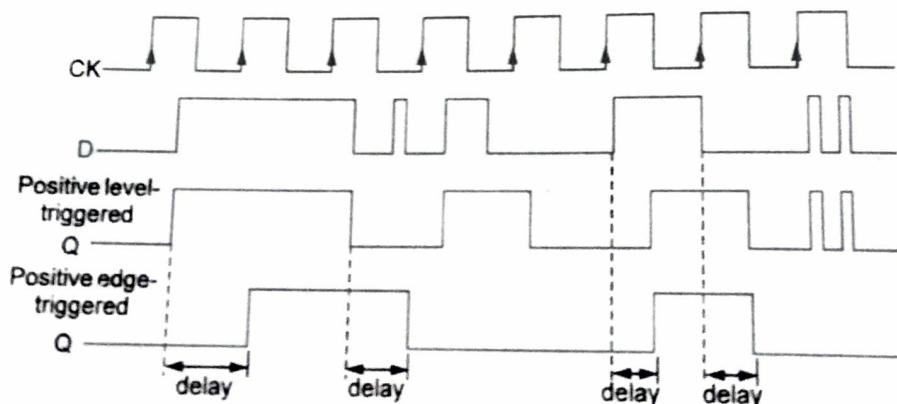
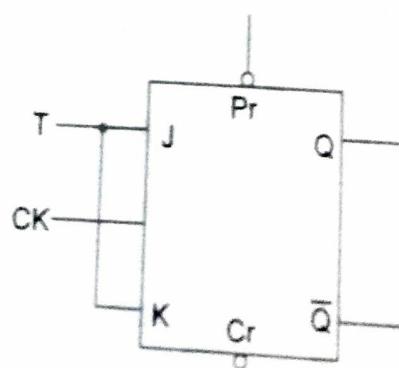


Fig. 4.53 Wave form of a D flip-flop

4.4.9. T FLIP-FLOP

Two combinations of J and K inputs of a JK flip-flop are used to make a D flip-flop. These were K = 1 and J = 1, K = 0. The remaining two combinations can be used to make a T flip-flop. Combinations are J = K = 0 and J = K = 1. As J = K for both the inputs, this is equivalent to a single T (0/1) connected to both J and K as shown in Fig. 4.54 along with its truth-table. This configuration is called a **T flip-flop**.



4.4.10. EXCITATIONS-TABLES FOR VARIOUS FLIP-FLOPS

Flip-flop Excitation Tables

Table (a) : SR

Q_n	Q_{n+1}	S	R
0	0	0	x
0	1	1	0
0	0	0	1
1	1	x	0

Table (c) : JK

Q_n	Q_{n+1}	J	K
0	0	0	x
0	1	1	x
0	0	x	1
1	1	x	x

Table (b) : D

Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

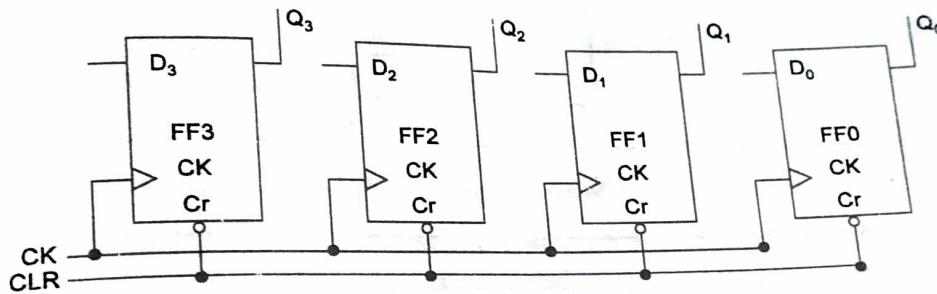
Table (d) : T

Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

4.4.11. REGISTERS

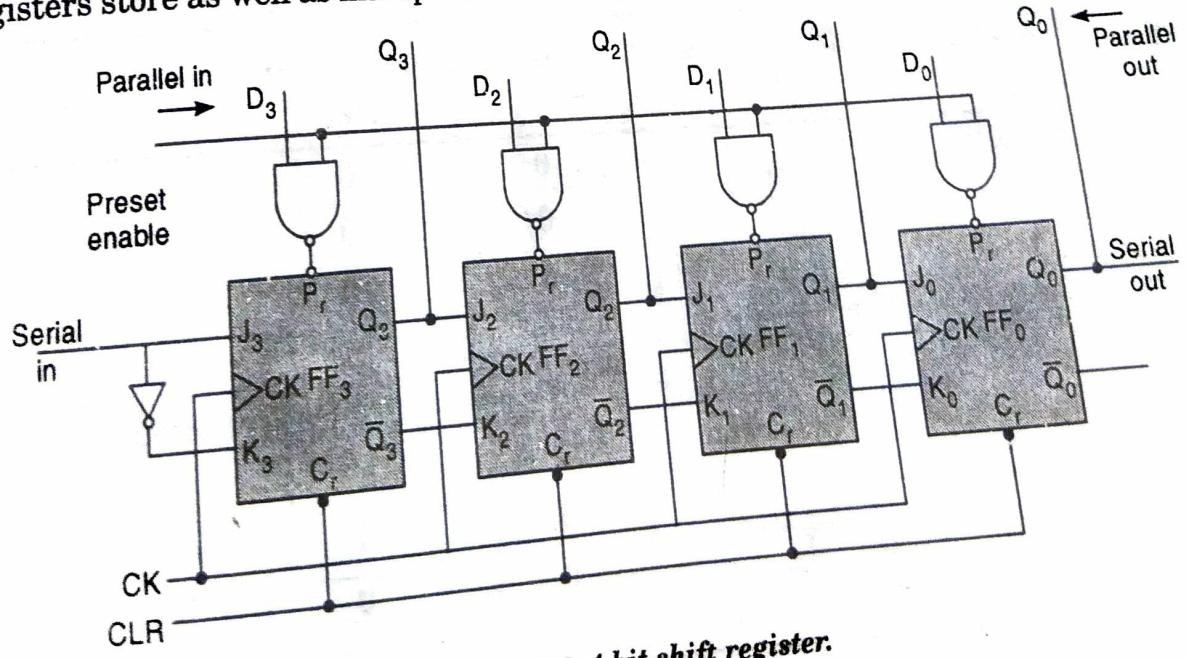
Memory Registers

Memory registers store n bits of data using n flip-flops. Figure 4.55 shows a 4-bit memory register (or buffer) composed of four D flip-flops which are triggered by a common clock. All the four flip-flops can be cleared by giving a 0 at the CLR input. Data input (4-bits).

**Fig. 4.55 4-bit memory register.**

Shift Registers

Shift registers store as well as manipulate data. Fig. 4.56 shows a 4-bit shift register.

**Fig. 4.56 4-bit shift register.**

4.46

the AND gates labeled as R_3, R_2, R_1, R_0 are enabled and the register functions as a shift right register. When M is 0, the AND gates labeled as R_i are disabled and the ones labeled as L_3, L_2, L_1, L_0 are enabled. This makes the register function as a left shift register. The data for shift right operation is entered at the Shift Right Serial-In input (DR_3). The data for shift left register is entered at the Shift Left Serial-In input (DL_3).

4.4.12. COUNTERS

Binary counters are most important and widely used digital circuits. A counter is a circuit that counts the number of occurrences of an input (in terms of positive or negative edge transitions in the case of a binary input). Each count, a binary number, is called a state of the counter. Hence, a counter counting in terms of n -bits has 2^n different states. The number of different states of a counter is also known as *modulus of the counter*. Thus an n -bit counter is a modulo 2^n counter.

Counter circuits are primarily constituted of flip-flops which alongwith combinational elements are used for the generation of control signals.

Depending on the manner in which the flip-flops are triggered, counters can be divided into two major categories :

1. Asynchronous Counter (ripple counter), and

2. Synchronous Counter

In case of an asynchronous counter, all the flip-flops are not clocked simultaneously, whereas in a synchronous counter all the flip-flops are clocked simultaneously.

If the counter counts in such a way that the decimal equivalent of the output increases with successive clock pulses. It is called an UP counter. If it decreases it is called a DOWN counter. An UP/DOWN counter can also be designed which can count in any direction depending upon the control input.

A decoding circuit can be connected to the output of a counter in such a way that the output of the circuit goes high (or low) only when the counter contents are equal to a given state.

4.4.13. ASYNCHRONOUS COUNTERS

Figure 4.59 shows a 4-bit (modulo-16) asynchronous counter. Four negative edge triggered JK flip flops are converted into T flip-flops by making $J = K = 1$ for all four of them. The Q output of one flip-flop is fed to the CK input of the next one.

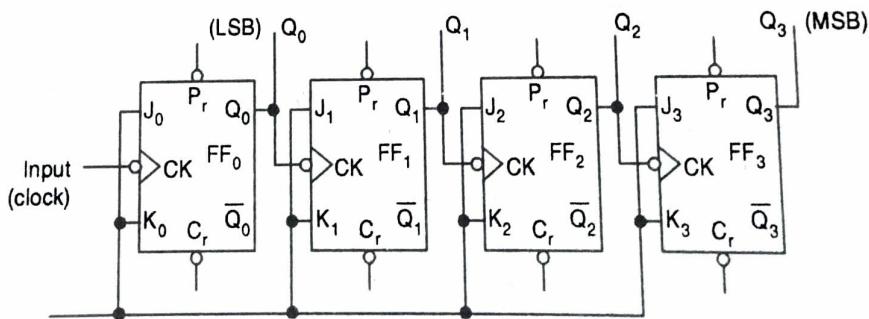


Fig. 4.59. 4-bit asynchronous counter