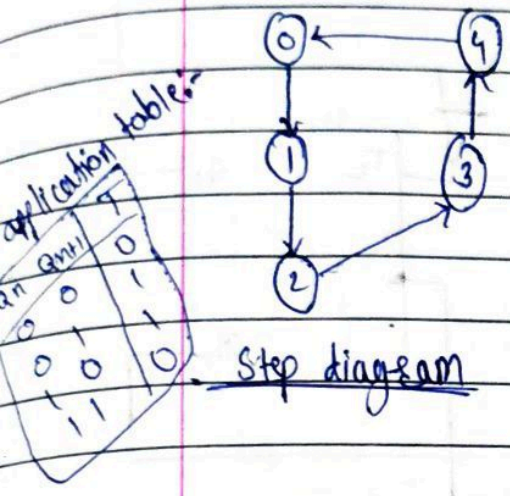
What is shift register ?

A register capable of shifting its binary information in one or both directions is called as a shift register. It consists of a chain of flip-flops in cascade, with the output of one flip-flop connected to the input of the next flip-flop.

Design MOD=5 synchronous counter using T-Flip Flop

⇒

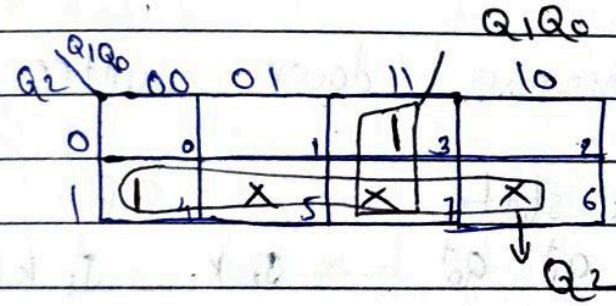


State diagram

Present State			Next state			FF		
Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+	T_2	T_1	T_0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	0	0	0	1	0	0

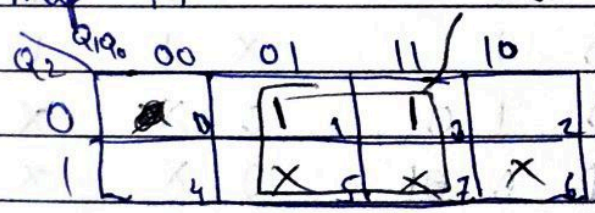
replace 5 by 8 and rewrite it

K-map for T_2



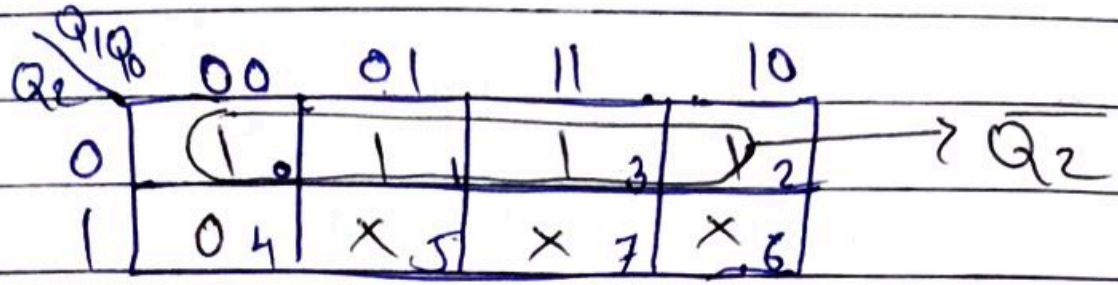
$T_2 = Q_0Q_1 + Q_2$

K-map T_1

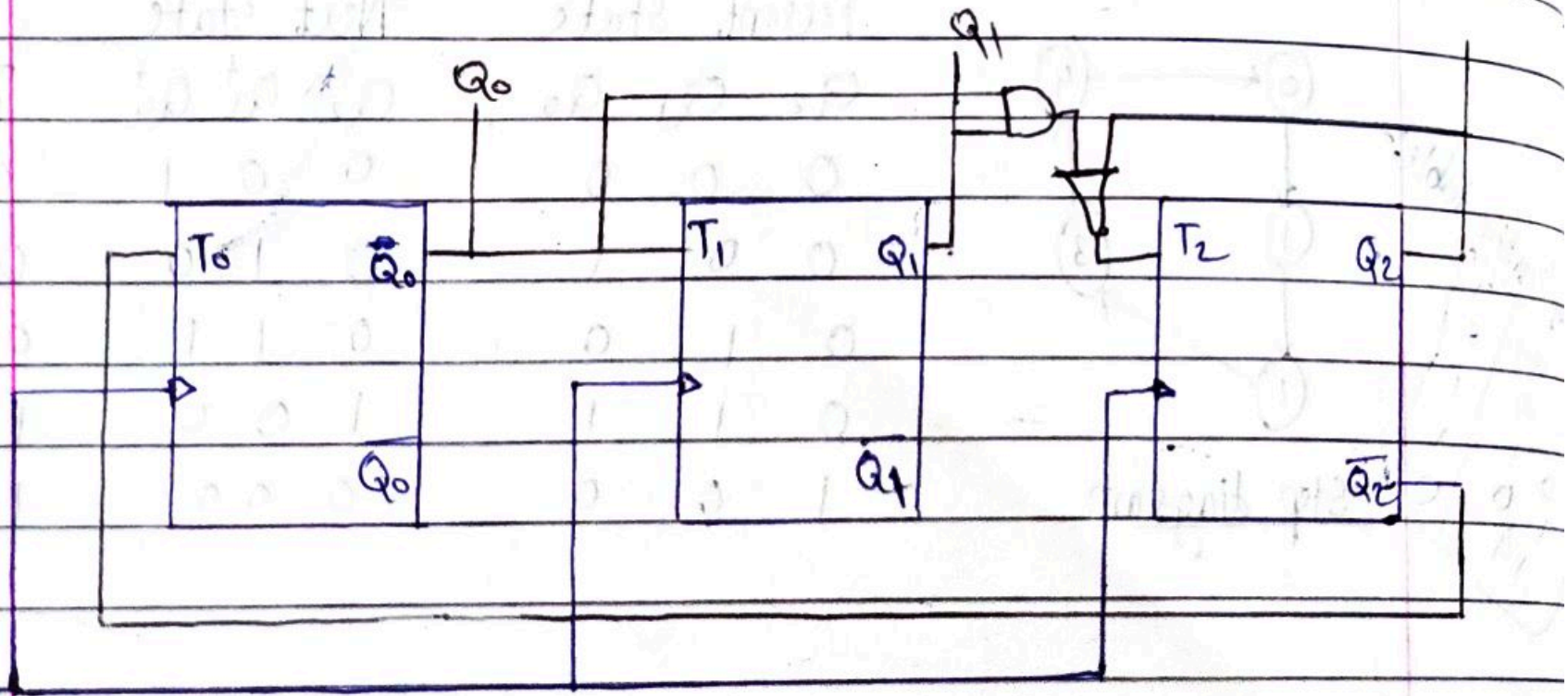


$T_1 = Q_0$

K-map for T_0



$$T_0 = \overline{Q_2}$$



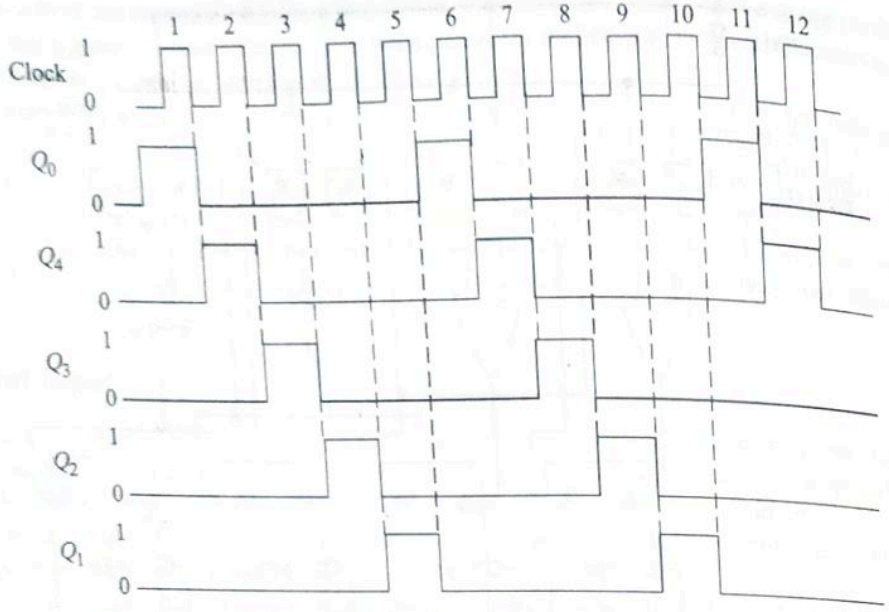


Fig. 8.5 *Output Waveforms of Ring Counter*

8.3.5 Twisted-Ring Counter

In the shift register of Fig. 8.2, if $\overline{Q}_0$ is connected to the serial input, the resulting circuit is referred to as a *twisted-ring, Johnson*, or *moebius* counter. If the clock pulses are applied after clearing the FLIP-FLOPs, square waveform is obtained at the Q outputs as shown in Fig. 8.6.

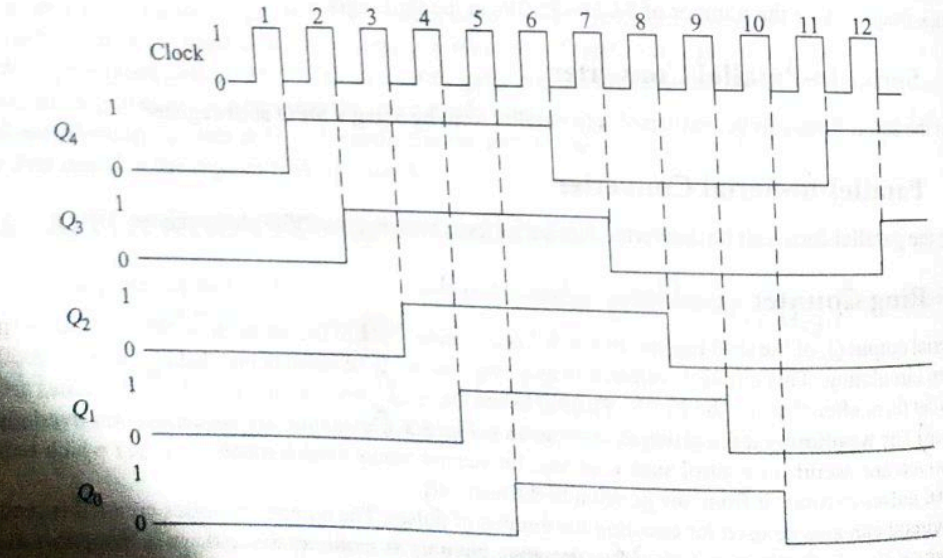


Fig. 8.6 *Output Waveforms of Twisted-Ring Counter*

Similar to ring-counter sequence, the moebius sequence is also useful for control-state counters. It is also useful for the generation of multiphase clock.

The moebius counter is a divide-by-2N counter. For decoding the count, two- input AND gates are required. The decoder circuit for a five-stage counter is shown in Fig. 8.7.

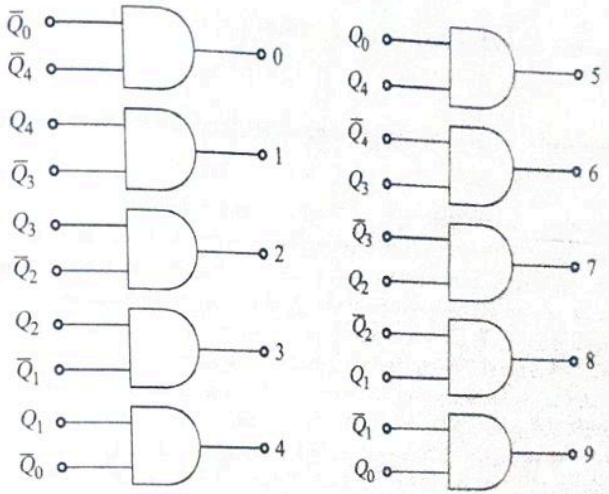


Fig. 8.7 *The Decoding Logic for a 5-Stage Twisted-Ring Counter*

8.3.6 Sequence Generator

A circuit which generates a prescribed sequence of bits, in synchronism with a clock, is referred to as a sequence generator. Such generators can be used as

1. Counters,
2. Random bit generators,
3. Prescribed period and sequence generators, and
4. Code generators.

The basic structure of a sequence generator is shown in Fig. 8.8.

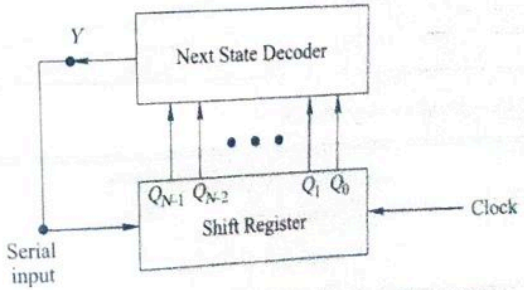


Fig. 8.8 *Basic Structure of a Sequence Generator*

Example 8.10

Design a 3-bit binary UP/DOWN counter with a direction control M. Use J-K FLIP-FLOPs.

Solution

The count sequence is given in Table 8.11. For $M = 0$, it acts as an UP counter and for $M = 1$ as a DOWN counter. The number of FLIP-FLOPs required is 3. The inputs of the FLIP-FLOPs are determined in a manner similar to the one employed in Ex. 8.9.

Table 8.11

Direction control	Counter state			FLIP-FLOP inputs					
M	Q_2	Q_1	Q_0	J_0	K_0	J_1	K_1	J_2	K_2
0	0	0	0	1	×	0	×	0	×
0	0	0	1	×	1	1	×	0	×
0	0	1	0	1	×	×	0	0	×
0	0	1	1	×	1	×	1	1	×
0	1	0	0	1	×	0	×	×	0
0	1	0	1	×	1	1	×	×	0
0	1	1	0	1	×	×	0	×	0
0	1	1	1	×	1	×	1	×	1
1	0	0	0	1	×	1	×	1	×
1	1	1	1	×	1	×	0	×	0
1	1	1	0	1	×	×	1	×	0
1	1	0	1	×	1	0	×	×	0
1	1	0	0	1	×	1	×	×	1
1	0	1	1	×	1	×	0	0	×
1	0	1	0	1	×	×	1	0	×
1	0	0	1	×	1	0	×	0	×
	0	0	0						

From Table 8.11, we obtain

$$J_0 = K_0 = 1$$

The K-maps for J_1, K_1, J_2, and K_2 are shown in Fig. 8.24. From the K-maps, the minimized expressions are obtained as

$$J_1 = K_1 = Q_0\overline{M} + \overline{Q}_0 M$$
$$J_2 = K_2 = \overline{M}\,Q_1 Q_0 + M\overline{Q}_1\,\overline{Q}_0$$

The counter circuit can be drawn using the above expressions

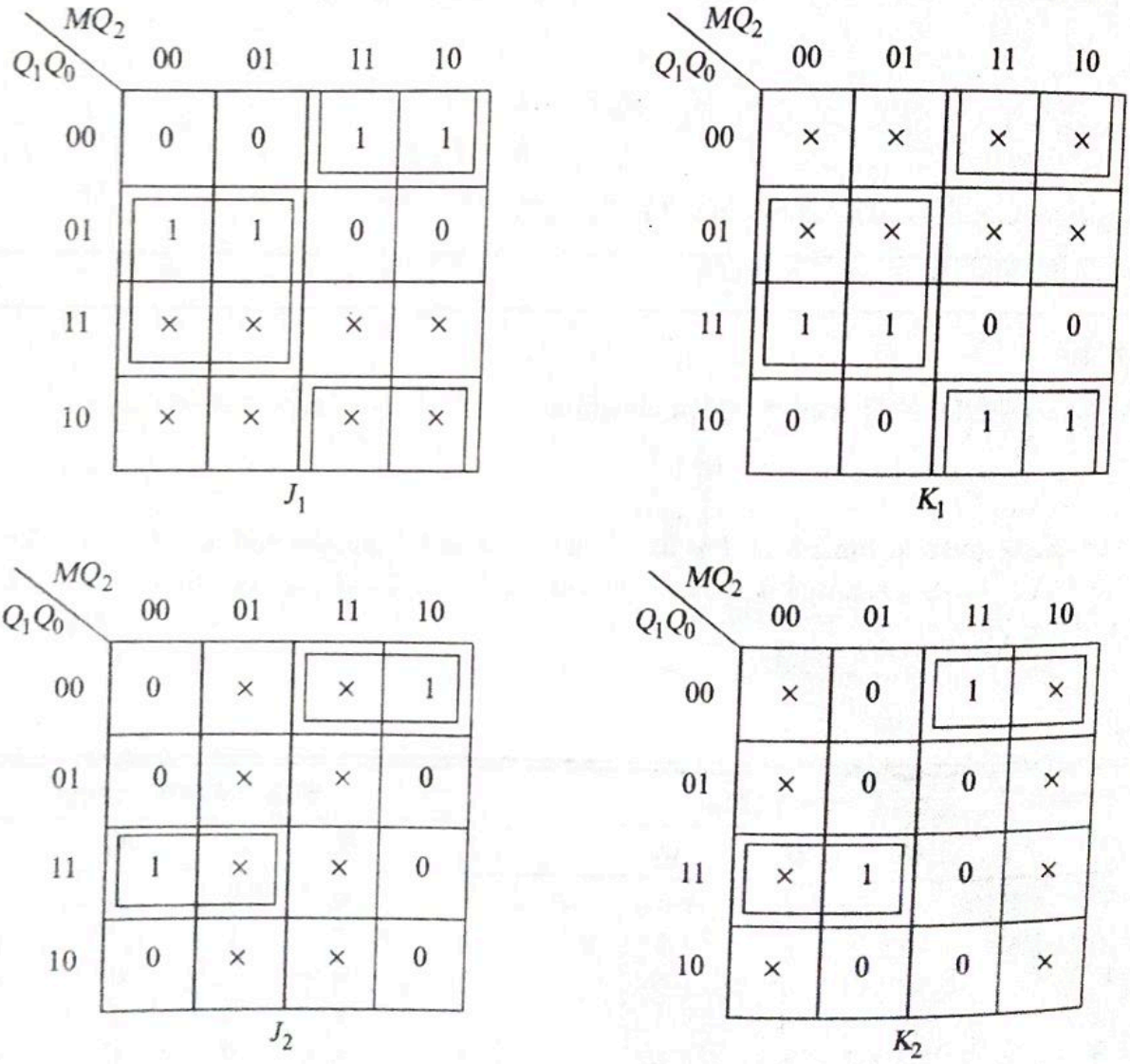


Fig. 8.24 ***K-Maps for Ex. 8.10***

8.15 Race Condition

It is the important to note that, in JK flip flop output is feedback to the input, and therefore change in the output results change in the input. Due to this in the positive half of the clock pulse if J and K are both high then output toggles continuously. This condition is known as race around condition; and must be avoided.

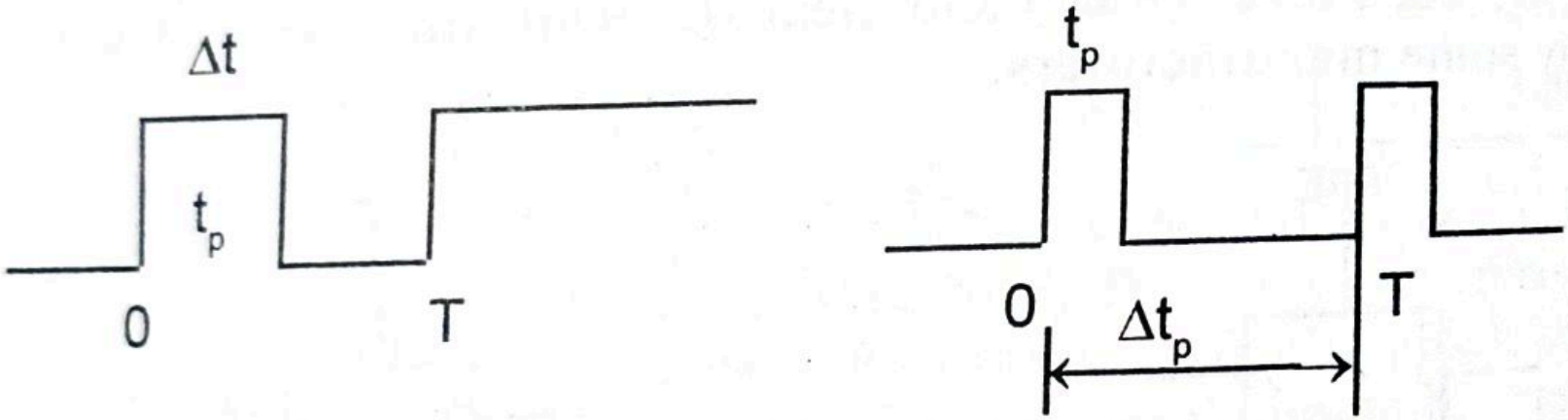


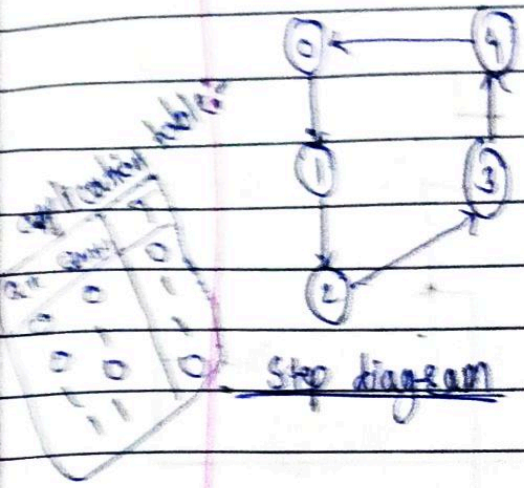
Fig.8.13 Race around condition

The race around conditions can be avoided if $t_p < \Delta t$ as shown in Fig. 8.13. Lumped delay lines can be used in series with the feed back connections hence prevent the race around difficulty. The above race condition can also be avoided in master salve flip flops.

A master-salve flip flop is constructed using two separate flip flops connected serially. The first flip flop serves as the master and second as a slave and over all circuit is referred to as a master-slave flip flop.

Design MOD-5 synchronous counter using T-flipflop

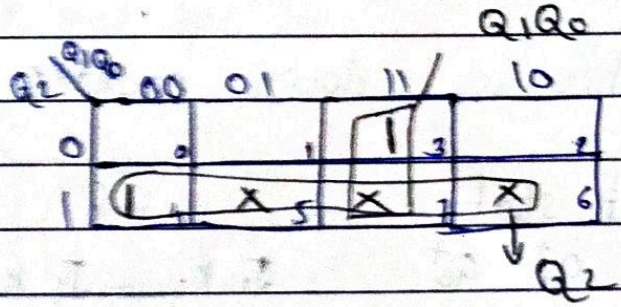
⇒



State diagram

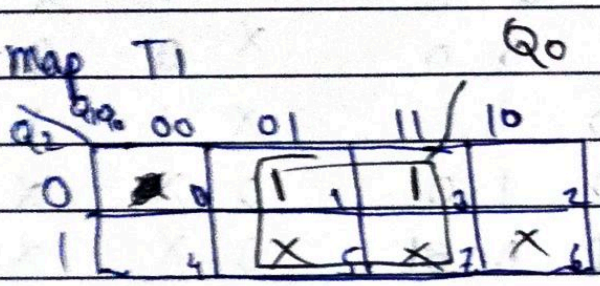
Present State			Next state			FF		
Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+	T_2	T_1	T_0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	0	0	0	1	0	0

K-map for T_2



$T_2 = Q_0 Q_1 + Q_2$

K-map T_1



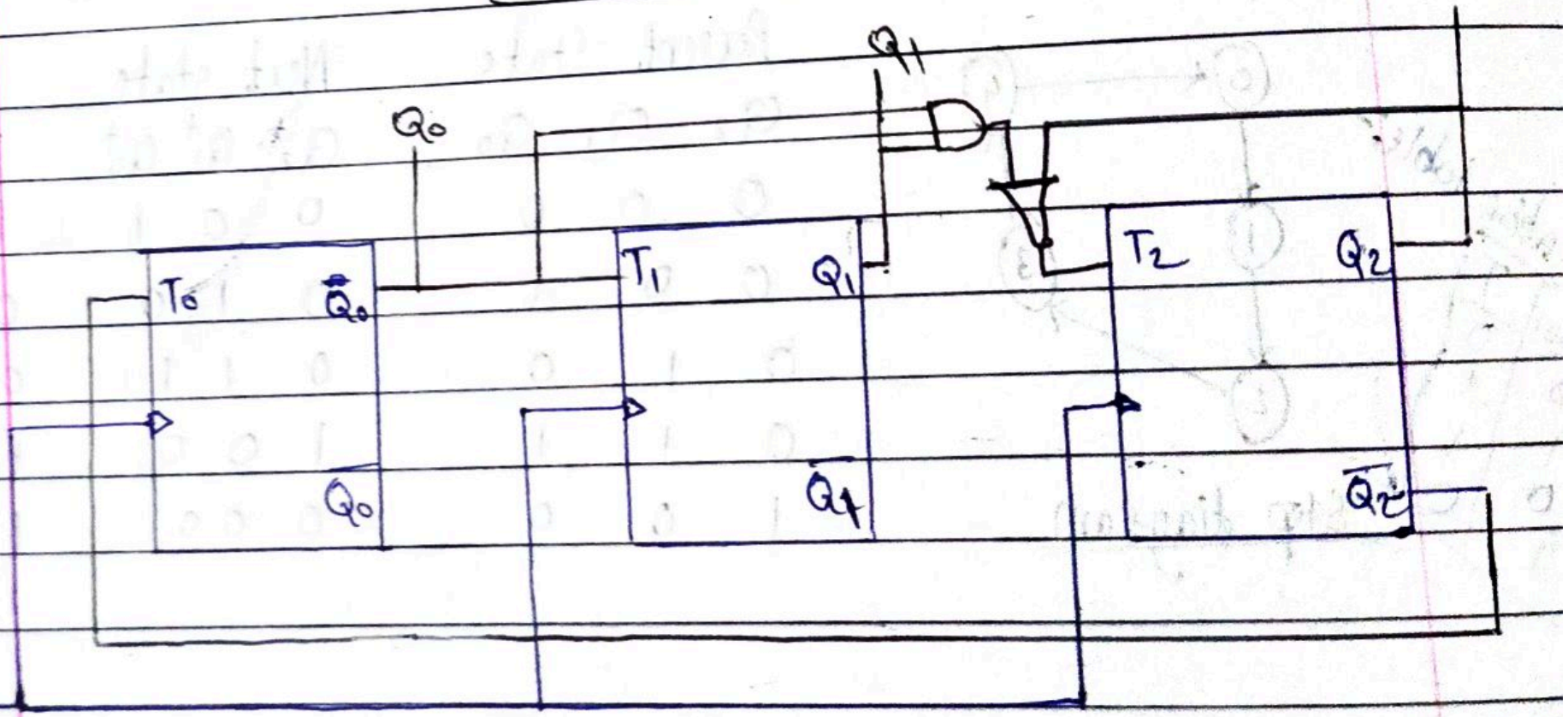
$T_1 = Q_0$

K-map for T_0

Q_2 \ Q_1Q_0	00	01	11	10
0	1 (0)	1 (1)	1 (3)	1 (2)
1	0 (4)	× (5)	× (7)	× (6)

→ $\overline{Q_2}$

$$T_0 = \overline{Q_2}$$



CLK

On this basis of above discussion the block diagram of 3-bit synchronous up counter is given in Figure 9.10.

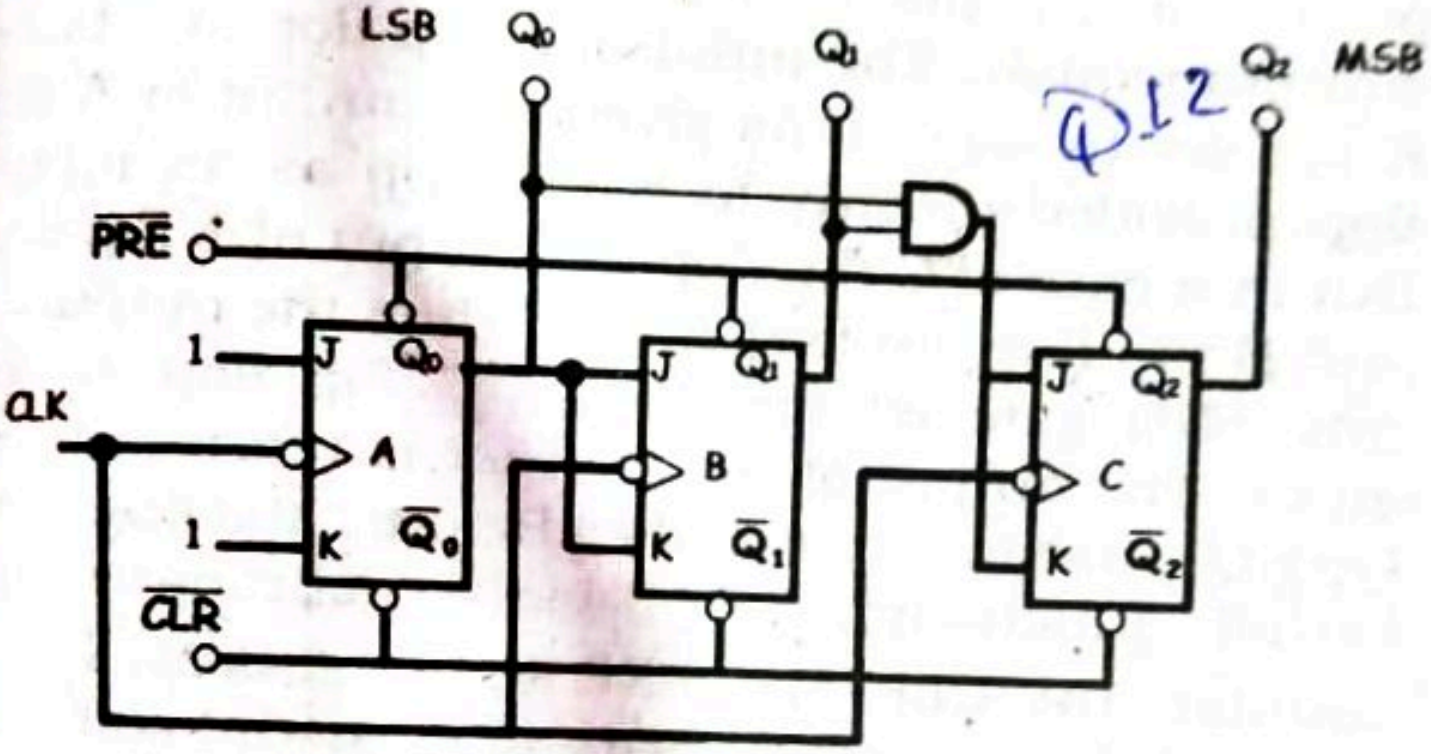


Figure 9.10 Block Diagram of 3-Bit Synchronous Up Counter

Construction:

The block diagram consists of three flip flops, these flip flops are MSJK flip flops. They are named as A, B and C. The $\overline{PRE}$ terminal and $\overline{CLR}$ terminal are connected to the entire 3 flip flops and this terminals are taken out commonly and are labeled as $\overline{PRE}$ and $\overline{CLR}$. By applying logic '0' to $\overline{CLR}$ terminal we can get the initial position.

Input	Output		
CLK	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

Table 9.7 Output of 3-Bit Synchronous Up Counter

Before we discuss the actual working of counter. We will find the constructive factor of block diagram of 3-bit synchronous up-counter systematically. The output of 3-bit synchronous up counter is shown in table 9.7.

Consider the truth table given in table 9.7 the output Q_0 of the least significant flip flop changes for every clock pulse i.e., the output Q_0 of A flip flop toggles (changes from 0 to 1. or 1 to 0) at every clock pulse. Hence A flip flop must be operated in toggle mode therefore, their J and K inputs must be at logic '1' i.e. J = K = 1. It is observed from Q_1 column of above table that output Q_1 toggles when Q_0 changes from '1' to '0'. Hence, Q_0 is directly connect to J and K inputs of a flip flop, as shown in Figure 9.10. Also, from the Q_2 column of table 9.7 it is observed that Q_2 toggle when both Q_0 and Q_1 are at logic '1'. Therefore, both of these terminals can be connected to J and K inputs of C flip flop using AND gate.

10.3.5 Bidirectional shift register

Definition

Bidirectional shift register A shift register that can serially shift bits left or right according to the state of a direction control input.

The bidirectional shift register can shift data in both direction i.e. left as well as right. The logic gates are arranged in such a way that data bits can be transferred from one stage to the next in either direction depends upon the control signal. Figure 10.6 shows the 4 bit bidirectional shift register.

Case 1

Right / $\overline{\text{Left}}$ = 1 (high)

Now this bidirectional shift registers act as a shift right register. This occurs because gates G_1, G_2, G_3 and G_4 are enabled and the Q output of one stage goes to D input of next stage. At negative edge of each clock, data bits shift one place to the right side.

Case 2

Right/Left = 0 (low)

Now this bidirectional shift register act as a shift left register. In this case gates G_5, G_6, G_7 and G_8 are enabled and Q output of each stage goes to D input of preceding stage. At the negative edge of each clock data bits shift one place to the left.

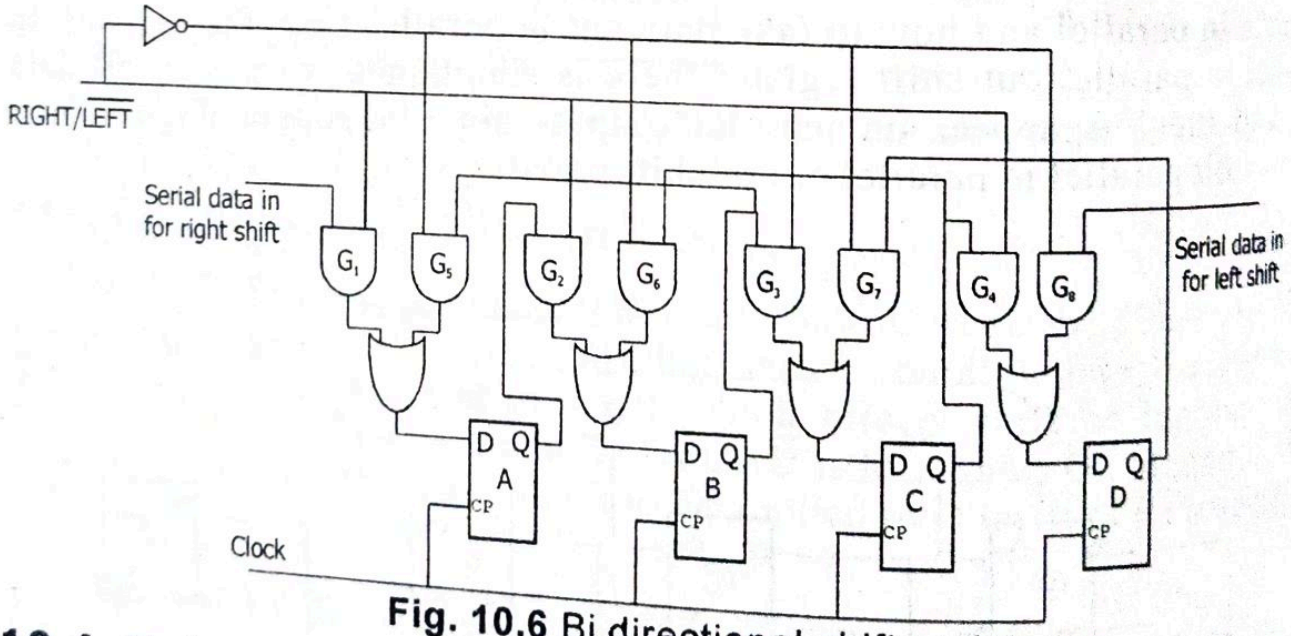


Fig. 10.6 Bi directional shift register

10.4 Universal Shift register

Definition

Universal shift register A shift register that can operate with any combination of serial and parallel inputs and outputs (i.e., serial in/serial out, serial in/parallel out, parallel in/serial out, parallel in/parallel out). A universal shift register is often bi directional as well.

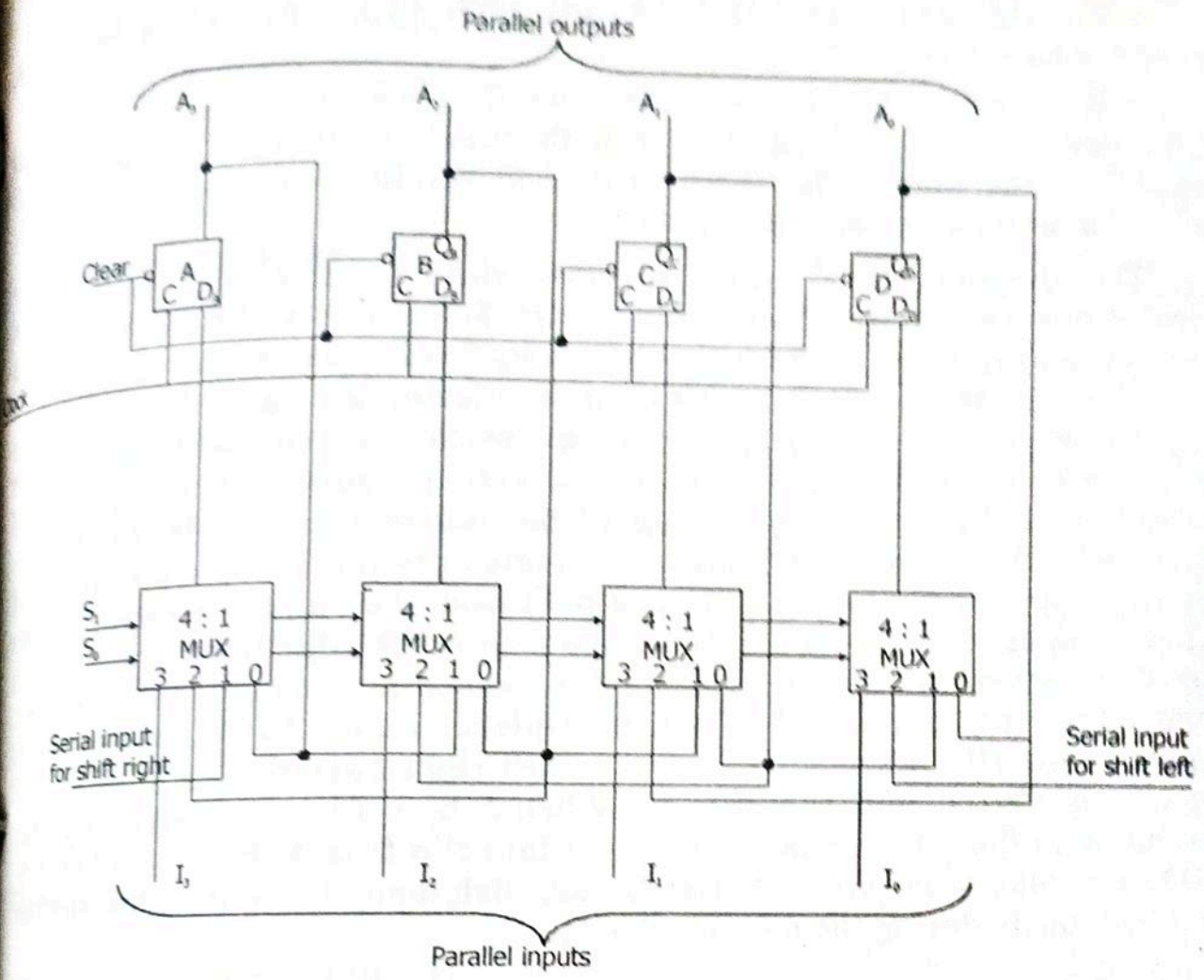


Fig. 10.7 4 bit Universal Shift Register

Figure 10.7 shows the Universal shift register, it performs all the operations of shift register. The different conditions of inputs are selected by using 4 : 1 multiplexer. So, the 4 bit universal shift register requires 4 multiplexer and 4 flip flops. Based on the select line the multiplexer allows one of the many inputs. Table 10.1 shows the function table of universal shift register.

Table 10.1 Function table of universal shift register

Mode Control		Register Operation
S_1	S_0	
0	0	No Change
0	1	Shift Right
1	0	Shift Left
1	1	Parallel Load

10.6 Ring Counter

The Figure 10.8 shows the ring counter. It uses D flip flops. The flip flop output is set to the flip flop input of next stage, Such that Q_0 sets D_1 input, Q_1 sets D_2 input, Q_2 sets D_3 input and Q_3 is feedback to D_0. Because of this connections, bits are shifted right one position per positive clock and

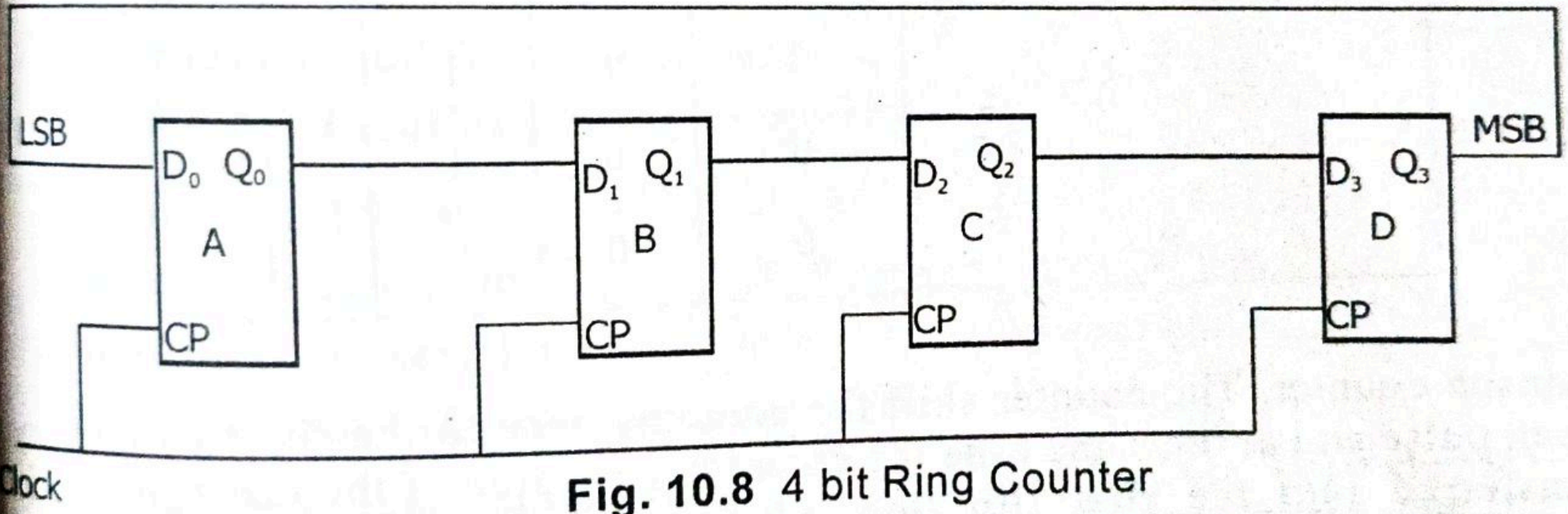


Fig. 10.8 4 bit Ring Counter

feedback to the input. Initially, all the flip flops are cleared by using clear inputs. The first positive clock shifts MSB to LSB position and other bits are to one position right. So that the output becomes Q = 0 0 1 0. This process continues, each output becomes a one after the negative etch transition of the clock pulse and remains one during the next clock cycle.

Logic Diagram

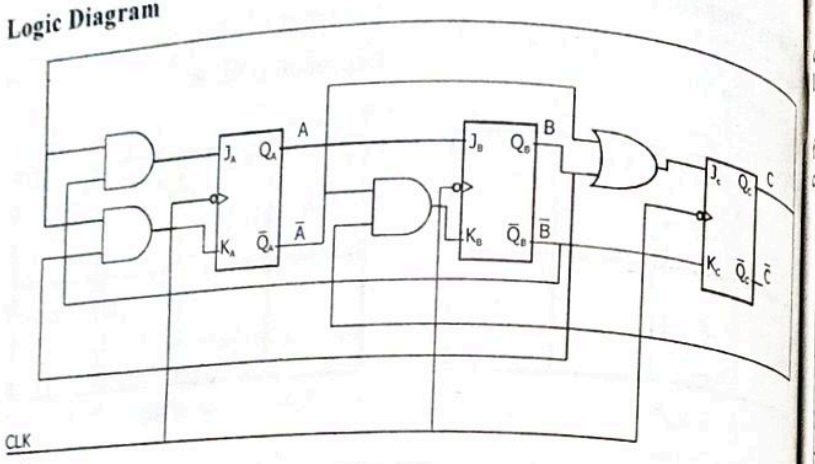


Fig. 9.3

9.8 Up/Down synchronous counter

Definition

UP counter A counter with an ascending sequence

Down counter A counter with an descending sequence.

An up/down counter is one that is capable of progressing in ascending order or descending order through a certain sequence. An up/down counter is also called bidirectional counter. Usually up/down operation of the counter is controlled by UP/DOWN signal. When this signal is HIGH, counter goes through UP sequence, i.e., 0,1,2,3,.....,n. When UP/$\overline{\text{DOWN}}$signal is $\overline{\text{LOW}}$ counter follows reverse sequence i.e. n, n-1, n-2, ...,1,0. For 3-bit counters these sequences are : 0,1,2,3,4,5,6,7 for up sequence and 7,6,5,4,3,2,1,0 for down sequence.

Table 9.2 State table for Three bit UP/DOWN counter

CP	UP	Q_C	Q_B	Q_A	DOWN
0		0	0	0	
1		0	0	1	
2		0	1	0	
3		0	1	1	
4		1	0	0	
5		1	0	1	
6		1	1	0	
7		1	1	1	

The arrows in the Table 9.2 indicate the state-to-state movement of the counter for both its UP and its DOWN modes of operation. Figure 9.3 shows logic diagram the three bit up/down counter

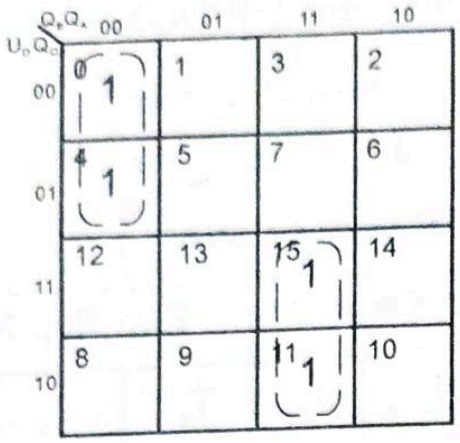
Let us design the 3 bit UP/DOWN synchronous counter, using T flip-flop. Table 9.3 shows the excitation table for 3-bit UP/DOWN synchronous counter.

Table 9.3 State Table of Updown counter

Input	Present State			Next State			Flip-flop Inputs		
UP/$\overline{\text{DOWN}}$ (U_D)	Q_C	Q_B	Q_A	Q_{C+1}	Q_{B+1}	Q_{A+1}	T_C	T_B	T_A
0	0	0	0	1	1	1	1	1	1
0	0	0	1	0	0	0	0	0	1
0	0	1	0	0	0	1	0	1	1
0	0	1	1	0	1	0	0	0	1
0	1	0	0	0	1	1	1	1	1
0	1	0	1	1	0	0	0	0	1
0	1	1	0	1	0	1	0	1	1
0	1	1	1	1	1	0	0	0	1
1	0	0	0	0	0	1	0	0	1
1	0	0	1	0	1	0	0	1	1
1	0	1	0	0	1	1	0	0	1
1	0	1	1	1	0	0	1	1	1
1	1	0	0	1	0	1	0	0	1
1	1	0	1	1	1	0	0	1	1
1	1	1	0	1	1	1	0	0	1
1	1	1	1	0	0	0	1	1	1

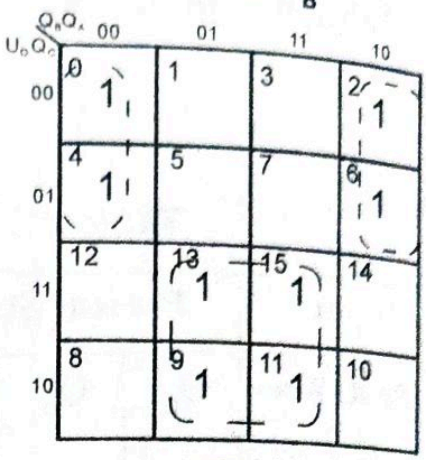
K-Map Simplification

For T_C



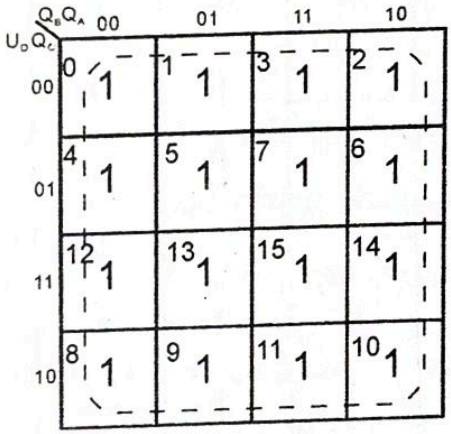
$$T_C = \overline{U}_D Q_B \overline{Q}_A + U_D Q_B Q_A$$

For T_B



$$T_B = \overline{U}_D Q_A + U_D Q_A$$

For T_A



$$T_A = 1$$

Logic Diagram

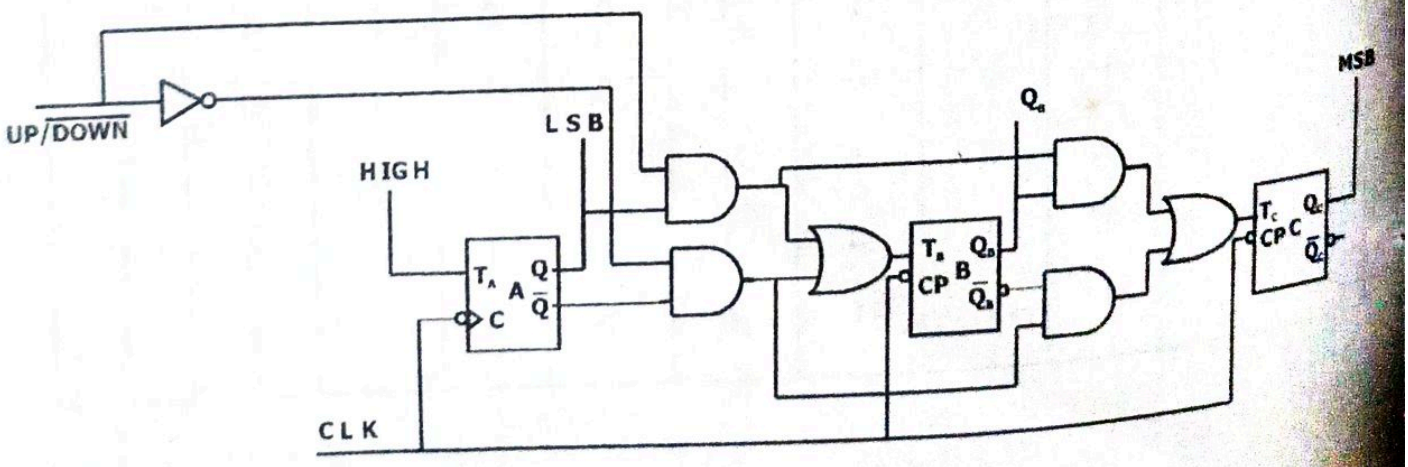


Fig. 9.3 Three bit UP/DOWN Counter

Design a 3-bit synchronous counter using *J-K* FLIP-FLOPs.

Solution

The number of FLIP-FLOPs required is 3. Let the FLIP-FLOPs be FF0, FF1, and FF2 and their inputs and outputs are given below:

FLIP-FLOP	Inputs	Output
FF0	J_0, K_0	Q_0
FF1	J_1, K_1	Q_1
FF2	J_2, K_2	Q_2

Table 8.10

Counter state			FLIP-FLOP inputs					
			FF0		FF1		FF2	
Q_2	Q_1	Q_0	J_0	K_0	J_1	K_1	J_2	K_2
0	0	0	1	×	0	×	0	×
0	0	1	×	1	1	×	0	×
0	1	0	1	×	×	0	0	×
0	1	1	×	1	×	1	0	×
1	0	0	1	×	0	×	1	×
1	0	1	×	1	1	×	×	×
1	1	0	1	×	×	1	×	0
1	1	1	×	1	×	0	×	0
0	0	0			×	1	×	0
							×	1

The count sequence and the required inputs of FLIP-FLOPs are given in Table 8.10. The inputs to the FLIP-FLOPs are determined in the following manner:

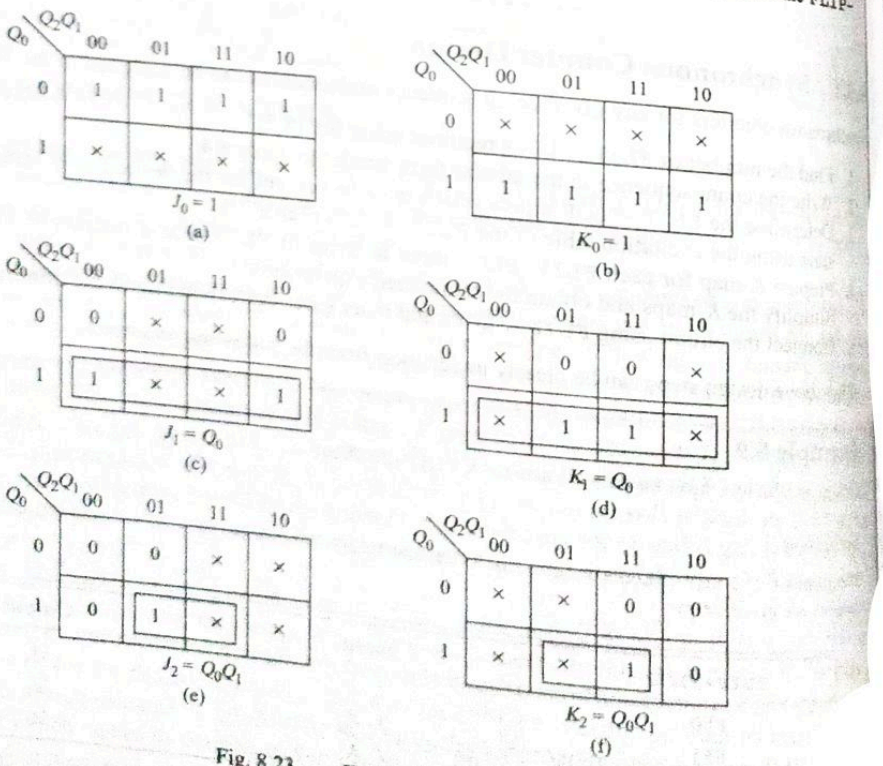


Fig. 8.23 *K-Maps of Ex. 8.9*

Consider one column of the counter state at a time and start from the first row, for example, consider Q_0. Before the first pulse is applied, $Q_0 = 0$ and it is required to be 1 at the end of the first clock pulse. Therefore, to achieve this condition, the values of J_0 and K_0 are 1 and × respectively (from the excitation Table 7.6). These are entered in the table in the row corresponding to 0 pulse. When the second clock pulse is applied Q_0 is to change from 1 to 0, therefore, the required inputs are

$$J_0 = \times, K_0 = 1$$

In a similar manner inputs of each FLIP-FLOP are determined.

Now, we prepare the K-maps (Fig. 8.23) with Q_2, Q_1, and Q_0 as input variables and FLIP-FLOP inputs as output variables. We then minimize the K-maps and the resulting minimized expressions are:

$$J_0 = 1, \quad K_0 = 1$$
$$J_1 = Q_0, \quad K_1 = Q_0$$
$$J_2 = Q_0Q_1, \quad K_2 = Q_0Q_1$$

The resulting counter circuit is same as the circuit of Fig. 8.22.

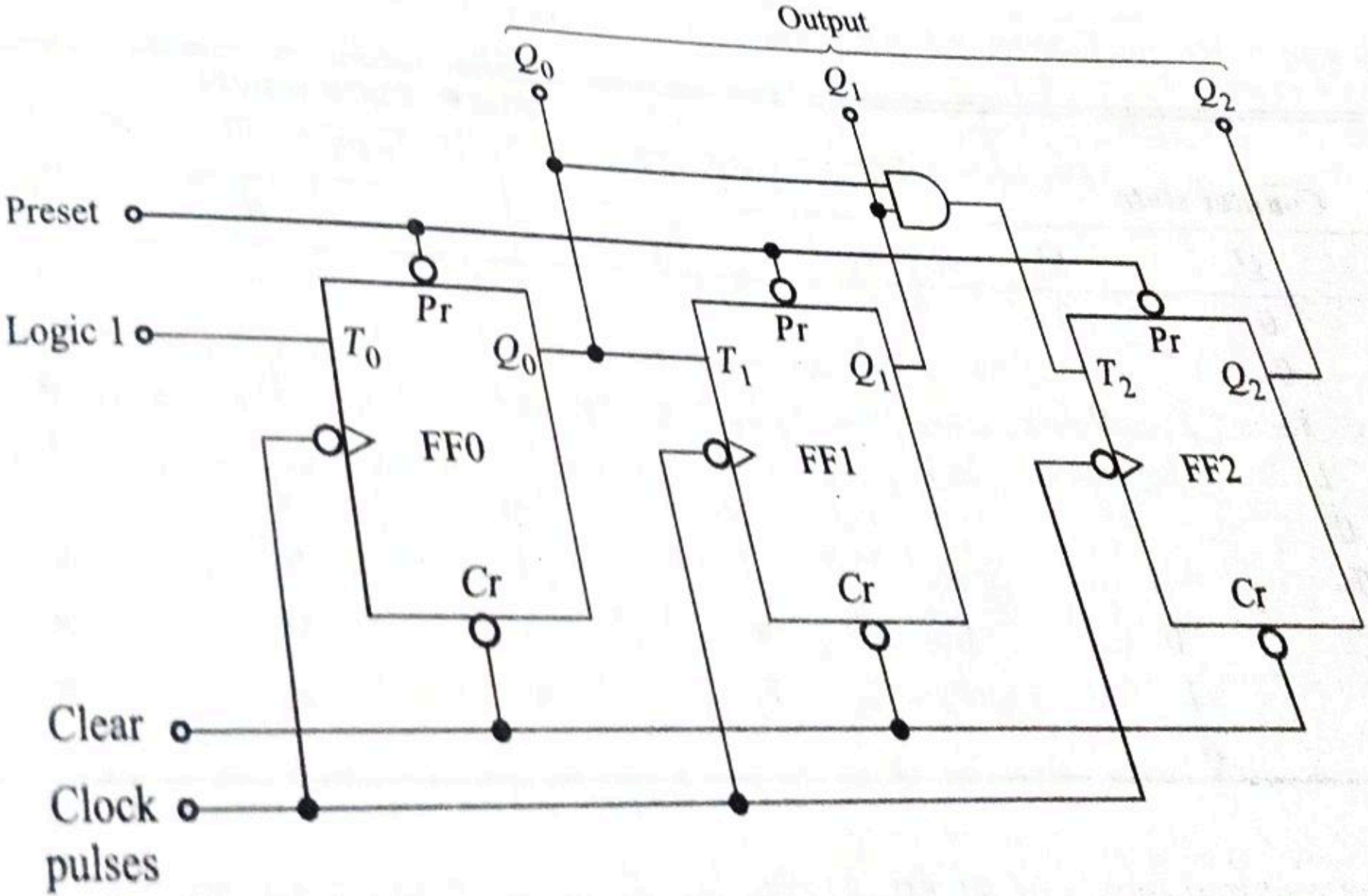


Fig. 8.22 **A 3-bit Synchronous Counter**

nchronous Counter Design

7.4.1 The Race-Around Condition

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

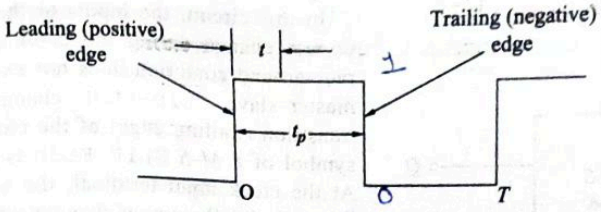


Fig. 7.11 ***A Clock Pulse***

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

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

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

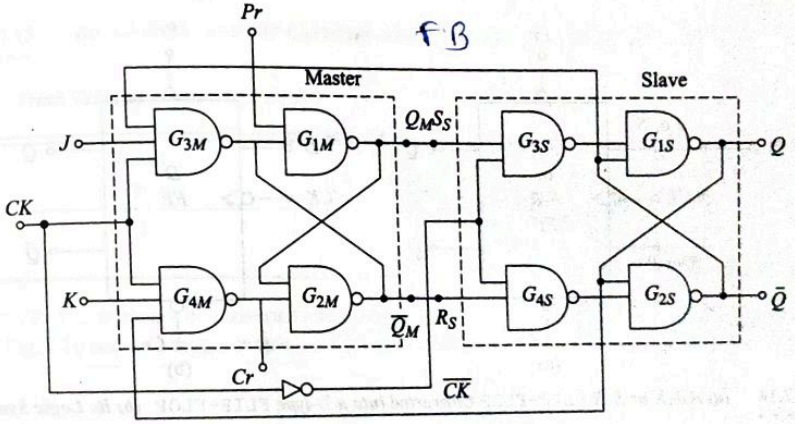


Fig. 7.12 ***A Master–Slave J–K*** FLIP-FLOP

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

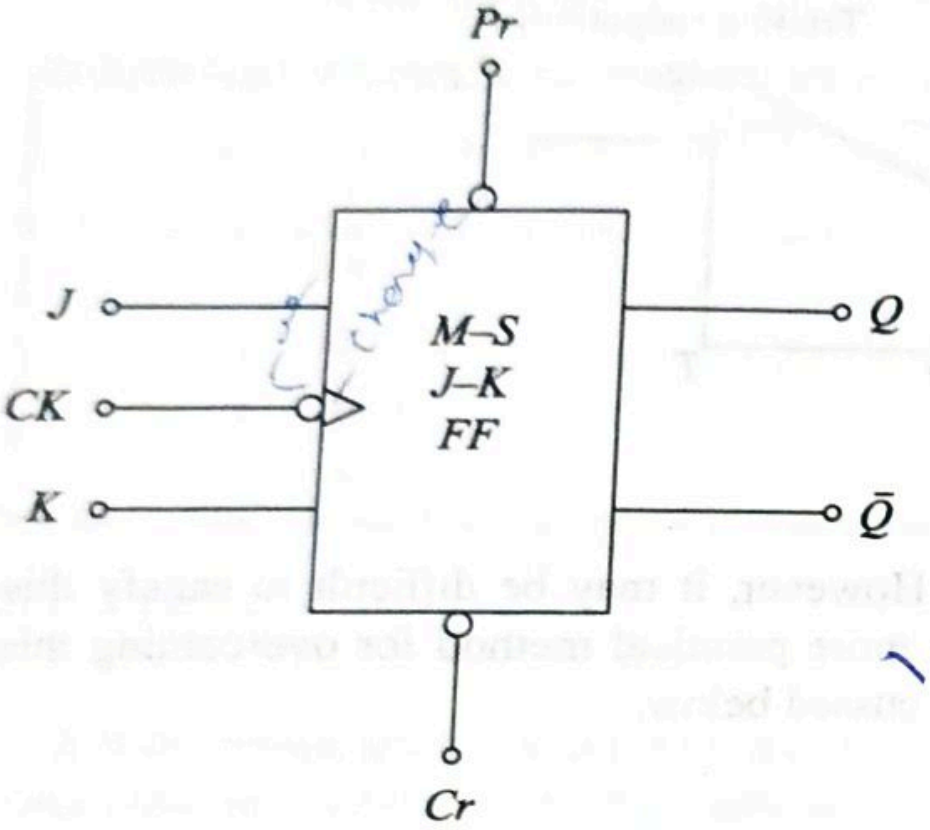


Fig. 7.13 ***A Master–Slave J–K*** FLIP-FLOP ***Logic Symbol***

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