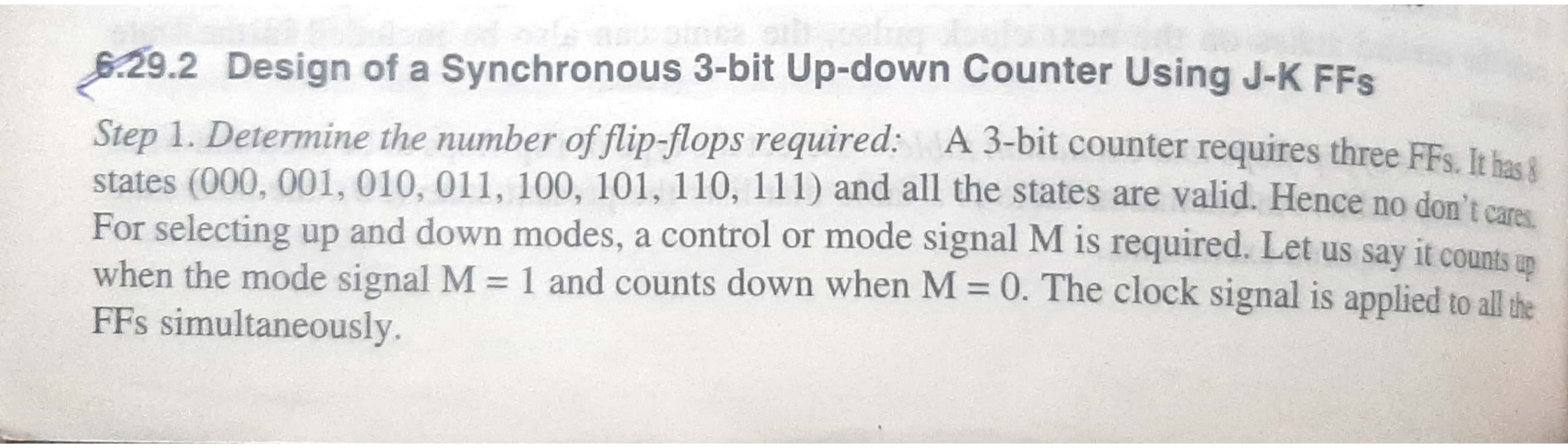
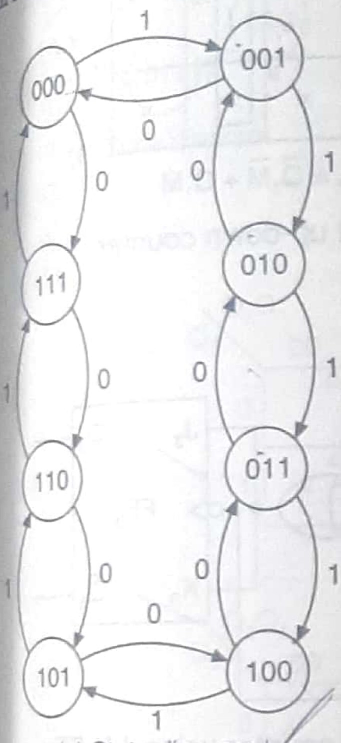


6.29.2 Design of a Synchronous 3-bit Up-down Counter Using J-K FFs

Step 1. Determine the number of flip-flops required: A 3-bit counter requires three FFs. It has 8 states (000, 001, 010, 011, 100, 101, 110, 111) and all the states are valid. Hence no don't cares. For selecting up and down modes, a control or mode signal M is required. Let us say it counts up when the mode signal M = 1 and counts down when M = 0. The clock signal is applied to all the FFs simultaneously.

Step 2. Draw the state diagram: The state diagram of the 3-bit up-down counter is drawn as shown in Figure 6.76a.

Step 3. Select the type of flip-flops and draw the excitation table: JK flip-flops are selected and the excitation table of a 3-bit up-down counter using JK flip-flops is drawn as shown in Figure 6.76b.



(a) State diagram

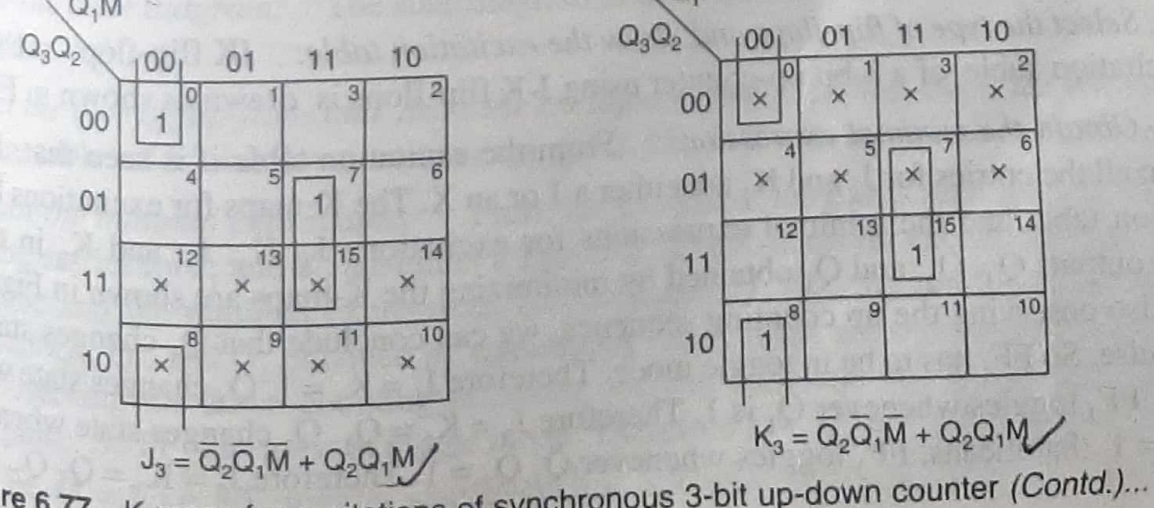
PS			Mode	NS			Required excitations					
Q_3	Q_2	Q_1	M	Q_3	Q_2	Q_1	J_3	K_3	J_2	K_2	J_1	K_1
0	0	0	0	1	1	1	1	×	1	×	1	×
0	0	0	1	0	0	1	0	×	0	×	1	×
0	0	1	0	0	0	0	0	×	0	×	×	1
0	0	1	1	0	1	0	0	×	1	×	×	1
0	1	0	0	0	0	1	0	×	×	1	1	×
0	1	0	1	0	1	1	0	×	×	0	1	×
0	1	1	0	0	1	0	0	×	×	0	×	1
0	1	1	1	1	0	0	1	×	×	1	×	1
1	0	0	0	0	1	1	×	1	1	×	1	×
1	0	0	1	1	0	1	×	0	0	×	1	×
1	0	1	0	1	0	0	×	0	0	×	×	1
1	0	1	1	1	1	0	×	0	1	×	×	1
1	1	0	0	1	0	1	×	0	×	1	1	×
1	1	0	1	1	1	1	×	0	×	0	1	×
1	1	1	0	1	1	0	×	0	×	0	×	1
1	1	1	1	0	0	0	×	1	×	1	×	1

(b) Excitation table

Figure 6.76 Synchronous 3-bit up-down counter.

Step 4. Obtain the minimal expressions: From the excitation table we can conclude that $J_1 = 1$ and $K_1 = 1$, because all the entries for J_1 and K_1 are either X or 1. The K-maps for J_3, K_3, J_2 and K_2 based on the excitation table and the minimal expressions obtained from them are shown in Figure 6.77.

Step 5. Draw the logic diagram: A logic diagram using those minimal expressions can be drawn as shown in Figure 6.78.



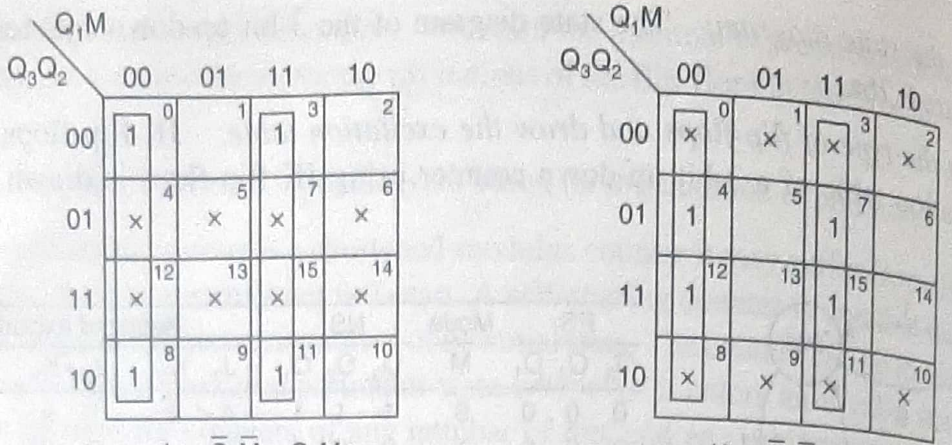
Q_3Q_2 \ Q_1M	00	01	11	10
00	1			
01			1	
11	×	×	×	×
10	×	×	×	×

$J_3 = \bar{Q}_2\bar{Q}_1\bar{M} + Q_2Q_1M$

Q_3Q_2 \ Q_1M	00	01	11	10
00	×	×	×	×
01	×	×	×	×
11			1	
10	1			

$K_3 = \bar{Q}_2\bar{Q}_1\bar{M} + Q_2Q_1M$

Figure 6.77 K-maps for excitations of synchronous 3-bit up-down counter *(Contd.)...*



$J_2 = \bar{Q}_1\bar{M} + Q_1M$ $K_2 = \bar{Q}_1\bar{M} + Q_1M$

Figure 6.77 K-maps for excitations of synchronous 3-bit up-down counter.

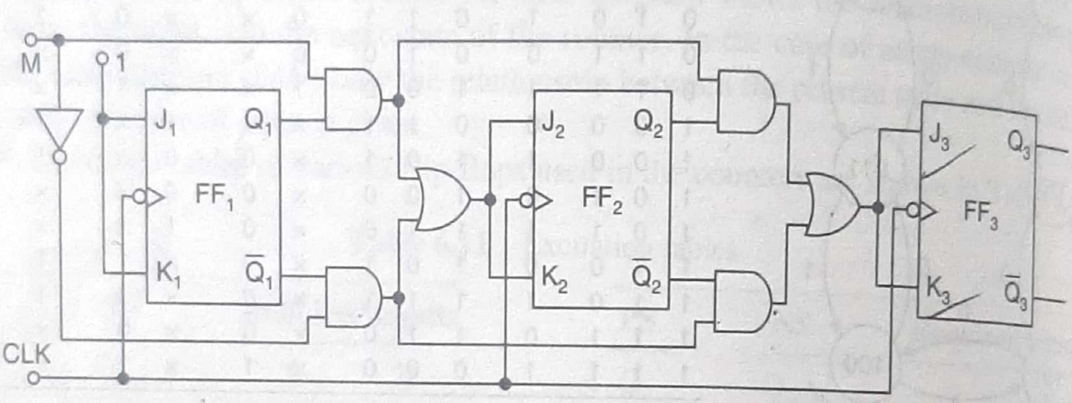


Figure 6.78 Logic diagram of the synchronous 3-bit up-down counter using J-K FFs.

Second method: A 3-bit up-down counter can also be realized by designing the up-counter and the down-counter separately and then combining them using a mode signal and additional gates.

6.29.3 Design of Synchronous 3-bit Up-counter

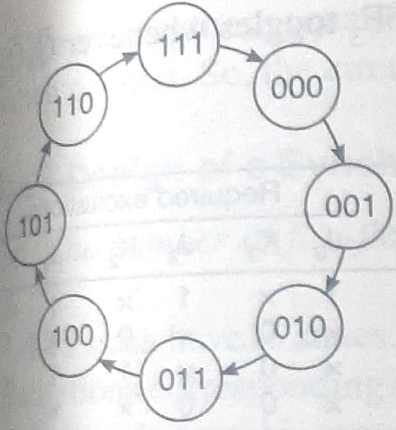
Step 1. Determine the number of flip-flops required: A 3-bit up-counter requires 3 flip-flops. The counting sequence is 000, 001, 010, 011, 100, 101, 110, 111, 000 ...

Step 2. Draw the state diagram: The state diagram of the 3-bit up-counter is drawn as shown in Figure 6.79a.

Step 3. Select the type of flip-flops and draw the excitation table: JK flip-flops are selected and the excitation table of a 3-bit up-counter using J-K flip-flops is drawn as shown in Figure 6.79b.

Step 4. Obtain the minimal expressions: From the excitation table it is seen that, $J_1 = K_1 = 1$, because all the entries for J_1 and K_1 are either a 1 or an X. The K-maps for excitations based on the excitation table and the minimal expressions for excitations J_3, K_3, J_2, and K_2 in terms of the present outputs Q_3, Q_2, and Q_1 obtained by minimizing the K-maps are shown in Figure 6.80.

Also observing the up counting sequence, we can conclude that Q_1 changes state for every clock pulse. So FF_1 has to be in toggle mode. Therefore $J_1 = K_1 = 1$. Q_2 changes state whenever Q_1 is 1, i.e. FF_2 toggles whenever Q_1 is 1. Therefore $J_2 = K_2 = Q_1$. Q_3 changes state whenever $Q_2 = 1$ and $Q_1 = 1$; that means, FF_3 toggles whenever $Q_1\ Q_2 = 1$. Therefore $J_3 = K_3 = Q_1\ Q_2$.

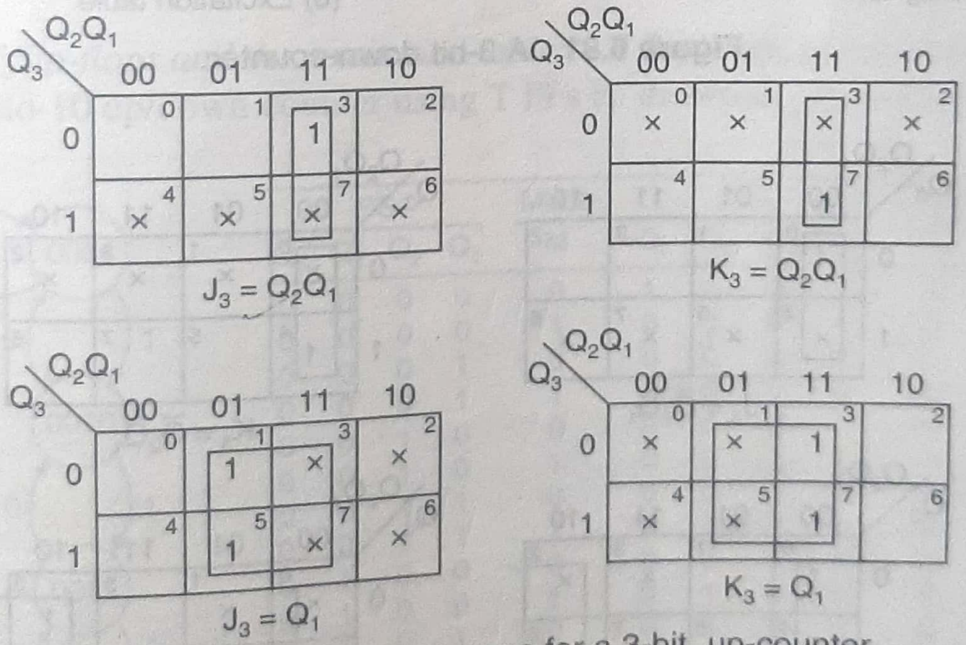


(a) State diagram

PS			NS			Required excitations					
Q_3	Q_2	Q_1	Q_3	Q_2	Q_1	J_3	K_3	J_2	K_2	J_1	K_1
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	1	0	1	×	0	0	×	1	×
1	0	1	1	1	0	×	0	1	×	×	1
1	1	0	1	1	1	×	0	×	0	1	×
1	1	1	0	0	0	×	1	×	1	×	1

(b) Excitation table

Figure 6.79 A 3-bit up-counter.



101101101
11
110110110
101101101

Figure 6.80 Karnaugh maps for a 3-bit up-counter.

(MOD-10)

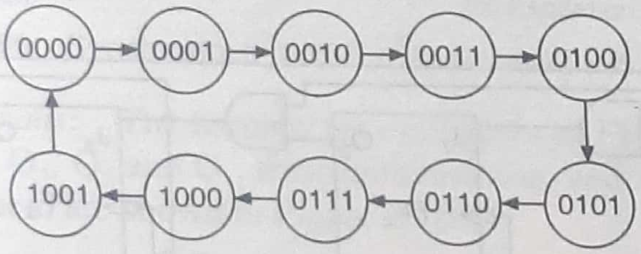
6.29.9 Design of a Synchronous BCD Counter Using J-K FFs

Step 1. The number of flip-flops: A BCD counter is nothing but a mod-10 counter. It is a decade counter. It has 10 states (0000 through 1001). It requires $n = 4$ FFs ($N \leq 2^n$, i.e. $10 \leq 2^4$). Four FFs can have 16 states. After the tenth clock pulse, the counter resets. So, states 1010 through, 1111 are invalid. The entries for excitations corresponding to invalid states are don't cares.

Step 2. The state diagram: The state diagram of the BCD counter is drawn as shown in Figure 6.95a.

Step 3. The type of flip-flops and the excitation table: JK flip-flops are selected and the excitation table of the BCD counter using J-K FFs is drawn as shown in Figure 6.95b.

Step 4. The minimal expressions: The K-maps for the excitations of FFs J_4, K_4, J_3, K_3, J_2, K_2, J_1 and K_1 in terms of the outputs of the FFs Q_4, Q_3, Q_2 and Q_1, based on the excitation table, their minimization and the minimal expressions obtained from them are shown in Figure 6.96.

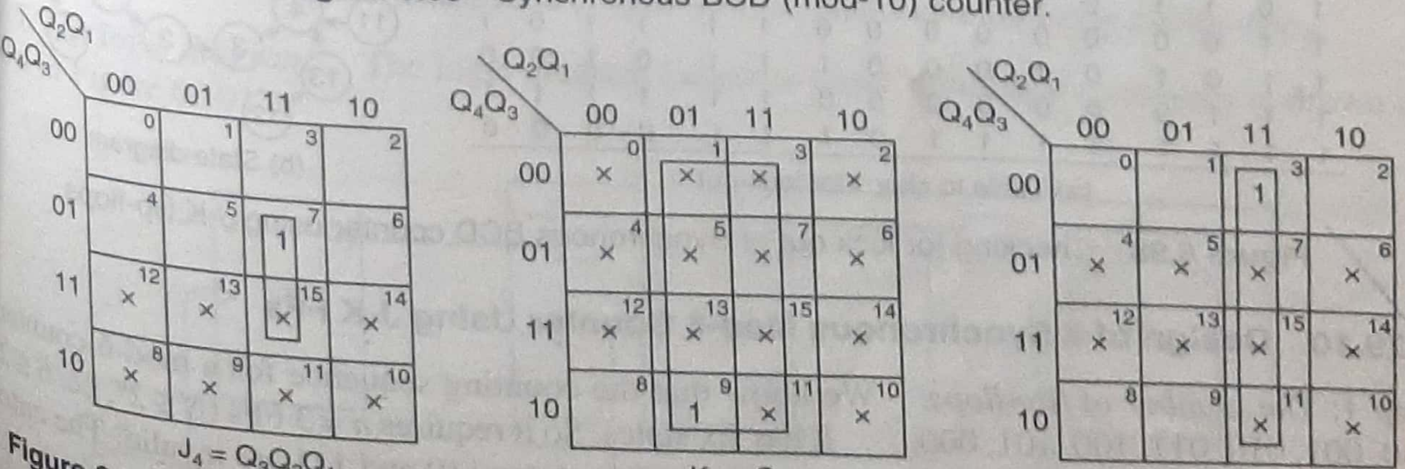


(a) State diagram

PS				NS				Required excitations							
Q_4	Q_3	Q_2	Q_1	Q_4	Q_3	Q_2	Q_1	J_4	K_4	J_3	K_3	J_2	K_2	J_1	K_1
0	0	0	0	0	0	0	1	0	×	0	×	0	×	1	×
0	0	0	1	0	0	1	0	0	×	0	×	1	×	×	1
0	0	1	0	0	0	1	1	0	×	0	×	×	0	1	×
0	0	1	1	0	1	0	0	0	×	1	×	×	1	×	1
0	1	0	0	0	1	0	1	0	×	×	0	0	×	1	×
0	1	0	1	0	1	1	0	0	×	×	0	1	×	×	1
0	1	1	0	0	1	1	1	0	×	×	0	×	0	1	×
0	1	1	1	1	0	0	0	1	×	×	1	×	1	×	1
1	0	0	0	1	0	0	1	×	0	0	×	0	×	1	×
1	0	0	1	0	0	0	0	×	1	0	×	0	×	×	1

(b) Excitation table

Figure 6.95 Synchronous BCD (mod-10) counter.



$Q_4Q_3 \backslash Q_2Q_1$	00	01	11	10
00				
01			1	
11	×	×	×	×
10	×	×	×	×

$J_4 = Q_3Q_2Q_1$

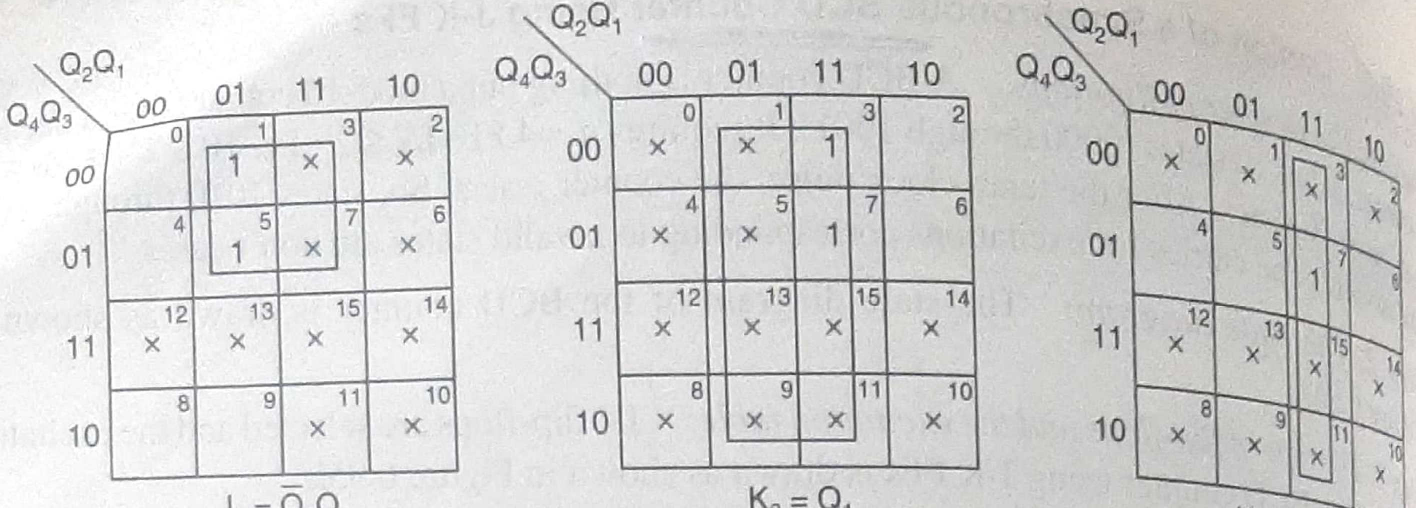
$Q_4Q_3 \backslash Q_2Q_1$	00	01	11	10
00	×	×	×	×
01	×	×	×	×
11	×	×	×	×
10		1	×	×

$K_4 = Q_1$

$Q_4Q_3 \backslash Q_2Q_1$	00	01	11	10
00			1	
01	×	×	×	×
11	×	×	×	×
10			×	×

$J_3 = Q_2Q_1$

Figure 6.96 K-maps for excitations of synchronous BCD counter using J-K flip-flops *(Contd.)*...



$J_2 = \bar{Q}_4Q_1$ $K_2 = Q_1$ $K_3 = Q_2Q_1$

Figure 6.96 K-maps for excitations of synchronous BCD counter using J-K flip-flops.

Step 5. The logic diagram: The logic diagram based on those minimal expressions is drawn as shown in Figure 6.97.

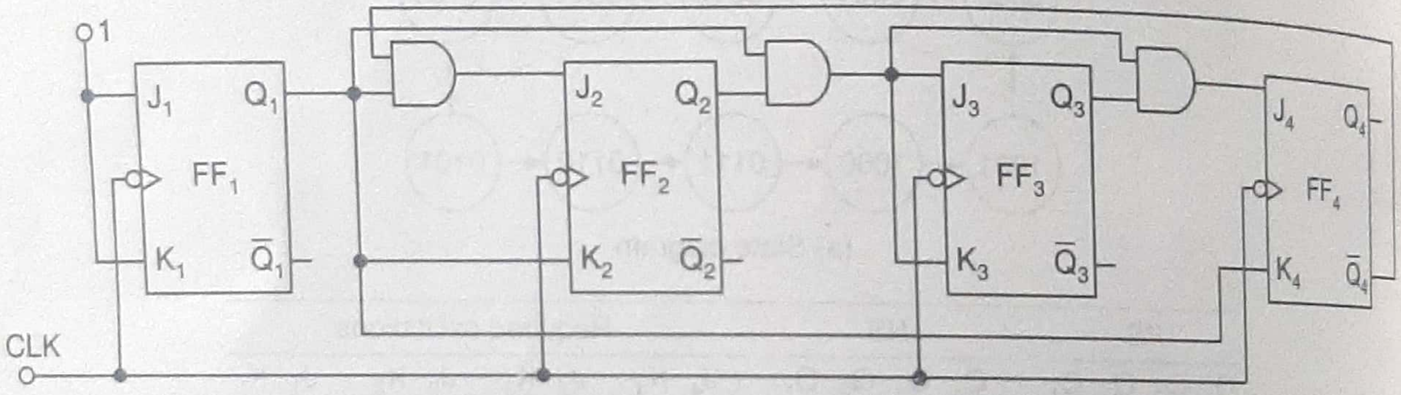
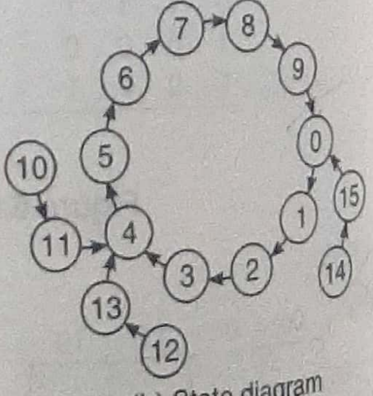


Figure 6.97 Logic diagram of synchronous BCD counter using J-K flip-flops.

The state diagram and the table to check for lock-out are shown in Figure 6.98. The table to check for lock-out shows that, if the counter finds itself in an invalid state initially, it moves to a valid state after one or two clock pulses and then counts in the normal way. Therefore, the counter is self-starting.

PS				Present inputs								NS			
Q_4	Q_3	Q_2	Q_1	J_4	K_4	J_3	K_3	J_2	K_2	J_1	K_1	Q_4	Q_3	Q_2	Q_1
1	0	1	0	0	0	0	0	0	0	1	1	1	0	1	1
1	0	1	1	0	1	1	1	0	1	1	1	0	1	0	0
1	1	0	0	0	0	0	0	0	0	1	1	1	1	0	1
1	1	0	1	0	1	0	0	0	1	1	1	0	1	0	0
1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	1	1	1	0	0	0	0

(a) Table to check for lock-out



(b) State diagram

Figure 6.98 Checking for lock out of Synchronous BCD counter using J-K flip-flops.

6.33.1 Ring Counter

This is the simplest shift register counter. The basic ring counter using D FFs is shown in Figure 6.120. The realization of this counter using J-K FFs is shown in Figure 6.121. Its state diagram and the sequence table are shown in Figure 6.122. Its timing diagram is shown in Figure 6.123. The FFs are arranged as in a normal shift register, i.e. the Q output of each stage is connected to the D input of the next stage, but the Q output of the last FF is connected back to the D input of the first FF such that the array of FFs is arranged in a ring and, therefore, the name *ring counter*.

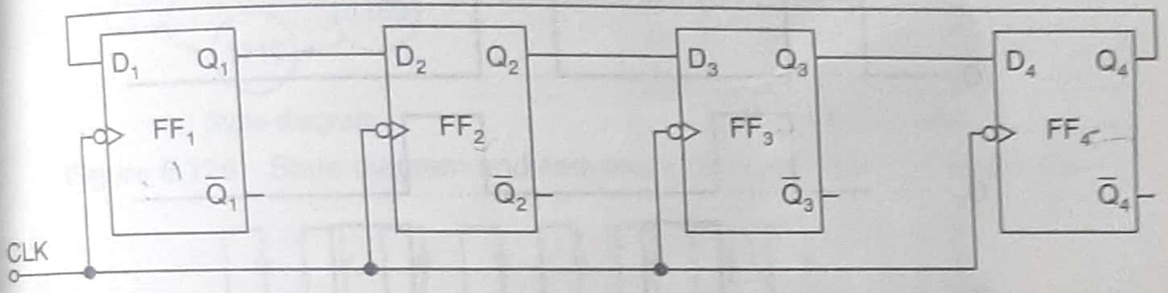


Figure 6.120 Logic diagram of a 4-bit ring counter using D flip-flops.

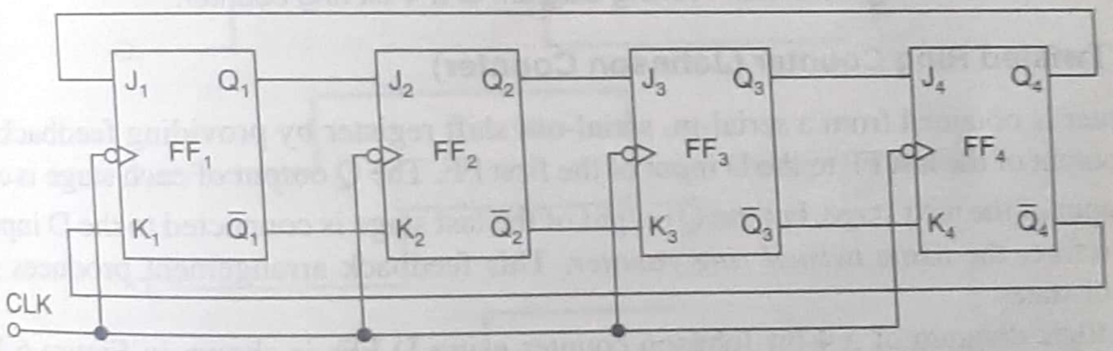
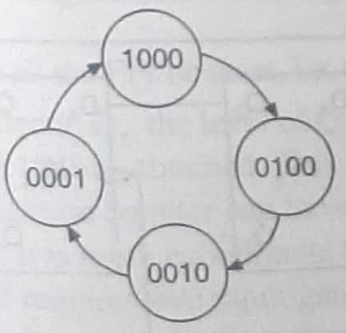


Figure 6.121 Logic diagram of a 4-bit ring counter using J-K flip-flops.



(a) State diagram

Q_1	Q_2	Q_3	Q_4	After clock pulse
1	0	0	0	0
0	1	0	0	1
0	0	1	0	2
0	0	0	1	3
1	0	0	0	4
0	1	0	0	5
0	0	1	0	6
0	0	0	1	7

(b) Sequence table

Figure 6.122 State diagram and sequence table of a 4-bit ring counter.

In most instances, only a single 1 is in the register and is made to circulate around the register as long as clock pulses are applied. Initially, the first FF is preset to a 1. So, the initial state is 1000, i.e. $Q_1 = 1$, $Q_2 = 0$, $Q_3 = 0$ and $Q_4 = 0$. After each clock pulse, the contents of the register are shifted to the right by one bit and Q_4 is shifted back to Q_1. The sequence repeats after four clock pulses. The number of distinct states in the ring counter, i.e. the mod of the ring counter is equal to the number of FFs used in the counter. An *n*-bit ring counter can count only *n* bits, whereas an *n*-bit ripple counter can count 2^n bits. So, the ring counter is uneconomical compared to a ripple counter, but has the advantage of requiring no decoder, since we can read the count by simply noting which

FF is set. Since it is entirely a synchronous operation and requires no gates external to FFs, it has the further advantage of being very fast.

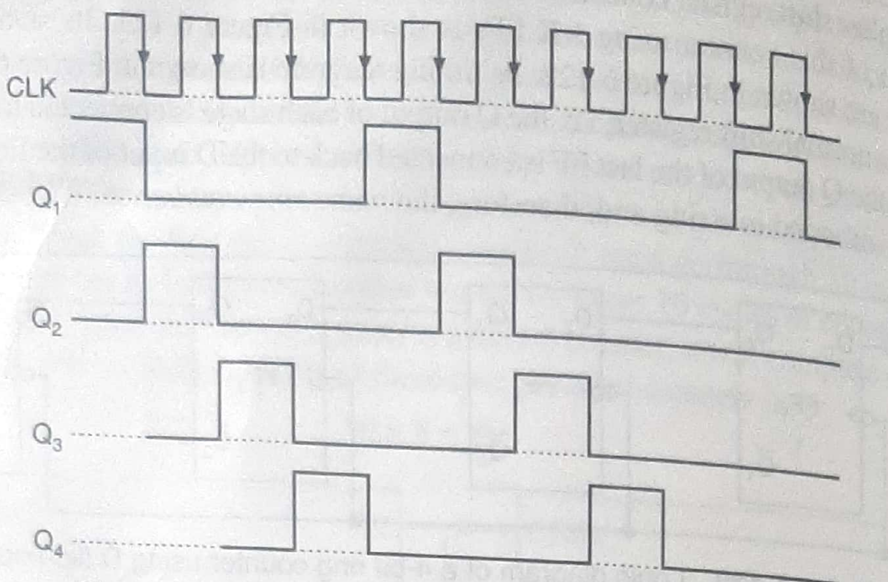


Figure 6.123 Timing diagram of a 4-bit ring counter.

6.33.2 Twisted Ring Counter (Johnson Counter)

This counter is obtained from a serial-in, serial-out shift register by providing feedback from the inverted output of the last FF to the D input of the first FF. The Q output of each stage is connected to the D input of the next stage, but the $\overline{Q}$ output of the last stage is connected to the D input of first stage, therefore, the name *twisted ring counter*. This feedback arrangement produces a unique sequence of states.

The logic diagram of a 4-bit Johnson counter using D FFs is shown in Figure 6.124. The realization of the same using J-K FFs is shown in Figure 6.125. The state diagram and the sequence table are shown in Figure 6.126. The timing diagram of a Johnson counter is shown in Figure 6.127.

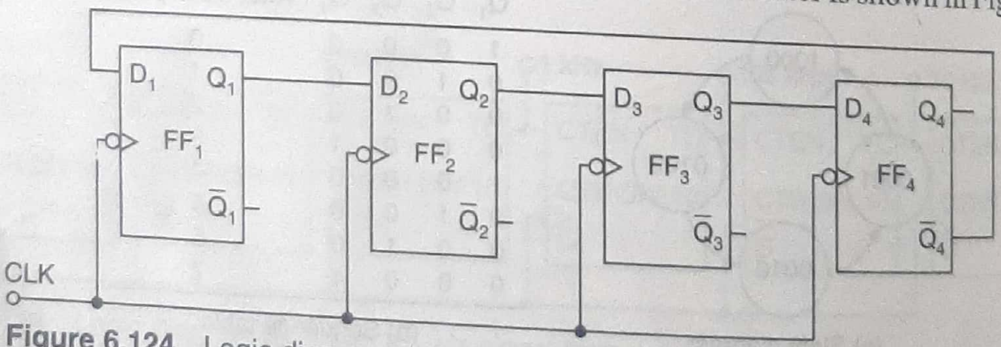


Figure 6.124 Logic diagram of a 4-bit twisted ring counter using D flip-flops.

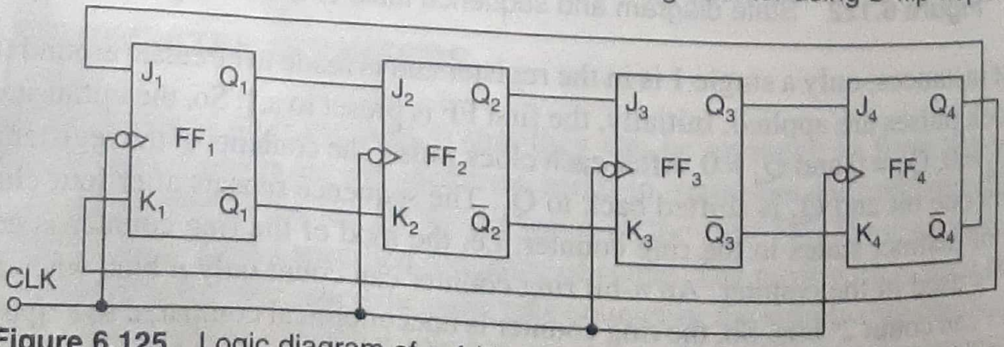
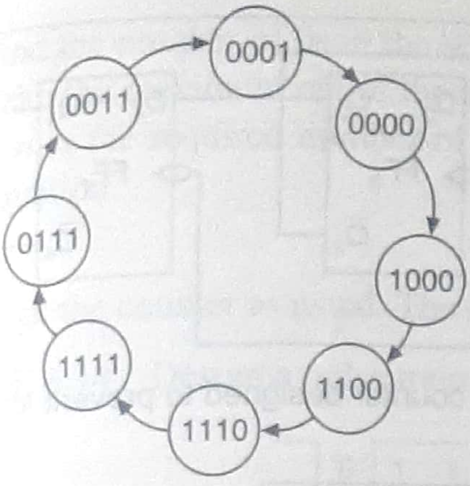


Figure 6.125 Logic diagram of a 4-bit twisted ring counter using J-K flip-flops.



(a) State diagram

Q_1	Q_2	Q_3	Q_4	After clock pulse
0	0	0	0	0
1	0	0	0	1
1	1	0	0	2
1	1	1	0	3
1	1	1	1	4
0	1	1	1	5
0	0	1	1	6
0	0	0	1	7
0	0	0	0	8
1	0	0	0	9

(b) Sequence table

Figure 6.126 State diagram and sequence table of a twisted ring counter.

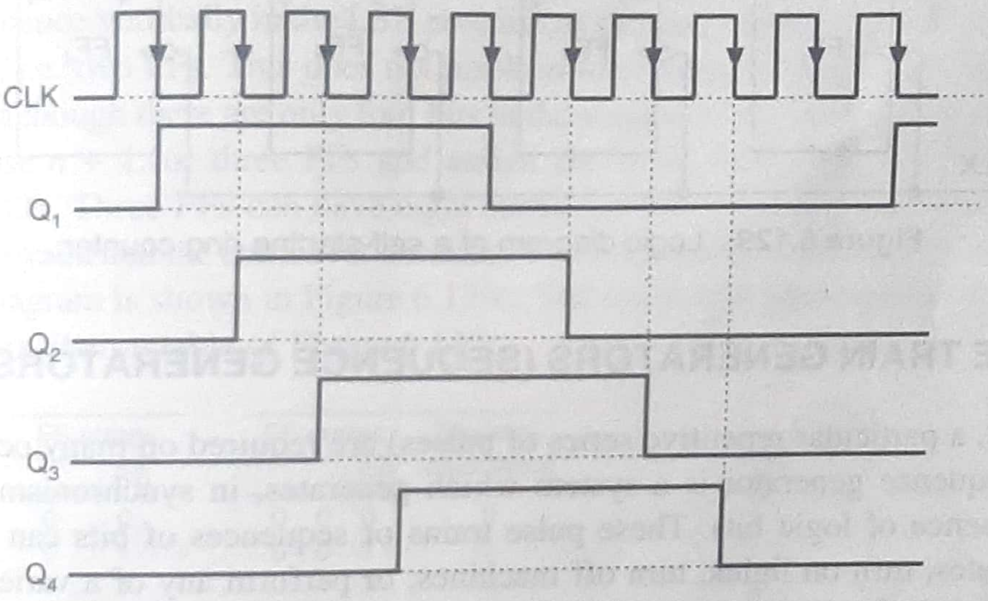


Figure 6.127 Timing diagram of a 4-bit twisted ring counter.

Let initially all the FFs be reset, i.e. the state of the counter be 0000. After each clock pulse, the level of Q_1 is shifted to Q_2, the level of Q_2 to Q_3, Q_3 to Q_4 and the level of $\overline{Q}_4$ to Q_1 and the sequence given in Figure 6.126b is obtained. This sequence is repeated after every eight clock pulses.

An *n* FF Johnson counter can have 2*n* unique states and can count up to 2*n* pulses. So, it is a mod-2*n* counter. It is more economical than the normal ring counter, but less economical than the ripple counter. It requires two input gates for decoding regardless of the size of the counter. Thus, it requires more decoding circuitry than that by the normal ring counter, but less than that by the ripple counter. It represents a middle ground between the ring counter and the ripple counter.

Both types of ring counters suffer from the problem of lock-out, i.e. if the counter finds itself in an unused state, it will persist in moving from one unused state to another and will never find its way to a used state. This difficulty can be corrected by adding a gate. With this addition, if the counter finds itself initially in an unused state, then after a number of clock pulses, depending on the state, the counter will find its way to a used state and thereafter, follow the desired sequence. A Johnson counter designed to prevent lock-out is shown in Figure 6.128. A self-starting ring counter (whatever may be the initial state, single 1 will eventually circulate) is shown in Figure 6.129.

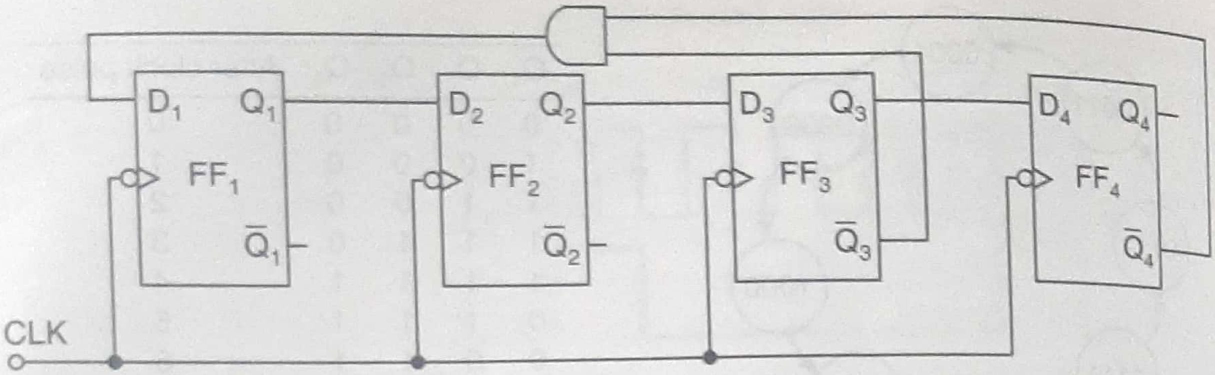


Figure 6.128 Logic diagram of a 4-bit Johnson counter designed to prevent lock-out.

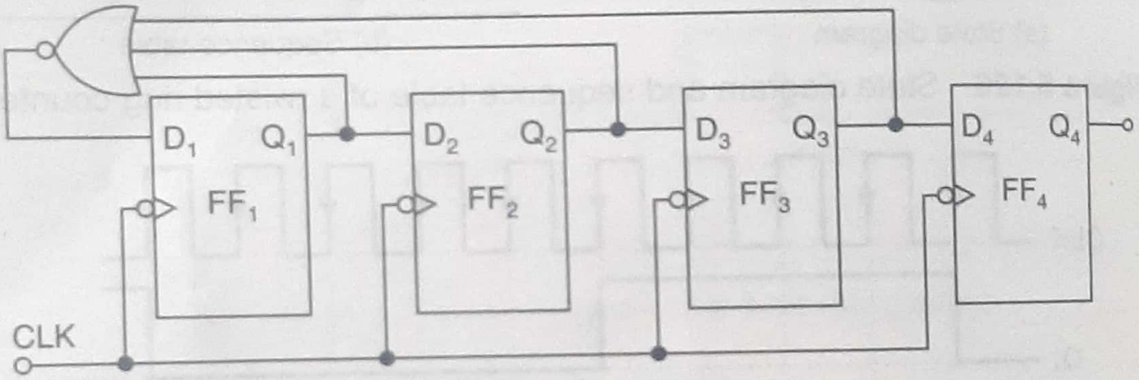


Figure 6.129 Logic diagram of a self-starting ring counter.