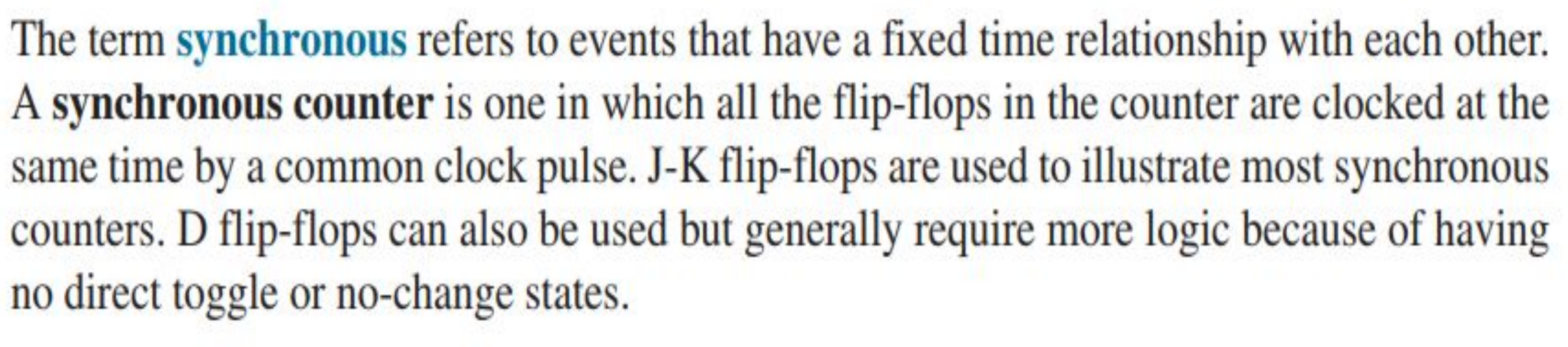
Synchronous Counters

Chapter#9

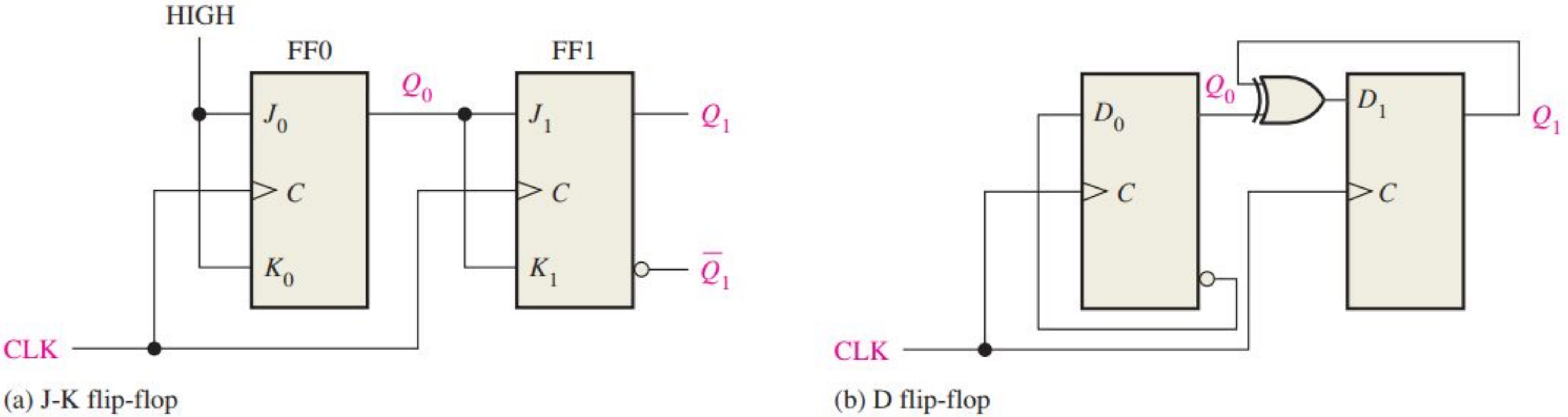
Synchronous Counters



The term synchronous refers to events that have a fixed time relationship with each other. A **synchronous counter** is one in which all the flip-flops in the counter are clocked at the same time by a common clock pulse. J-K flip-flops are used to illustrate most synchronous counters. D flip-flops can also be used but generally require more logic because of having no direct toggle or no-change states.

A 2-Bit Synchronous Binary Counter

Figure 9–12 shows a 2-bit synchronous binary counter. Notice that an arrangement different from that for the asynchronous counter must be used for the J_1 and K_1 inputs of FF1 in order to achieve a binary sequence. A D flip-flop implementation is shown in part (b).



(a) J-K flip-flop

(b) D flip-flop

FIGURE 9–12 2-bit synchronous binary counters.

A 3-Bit Synchronous Binary Counter

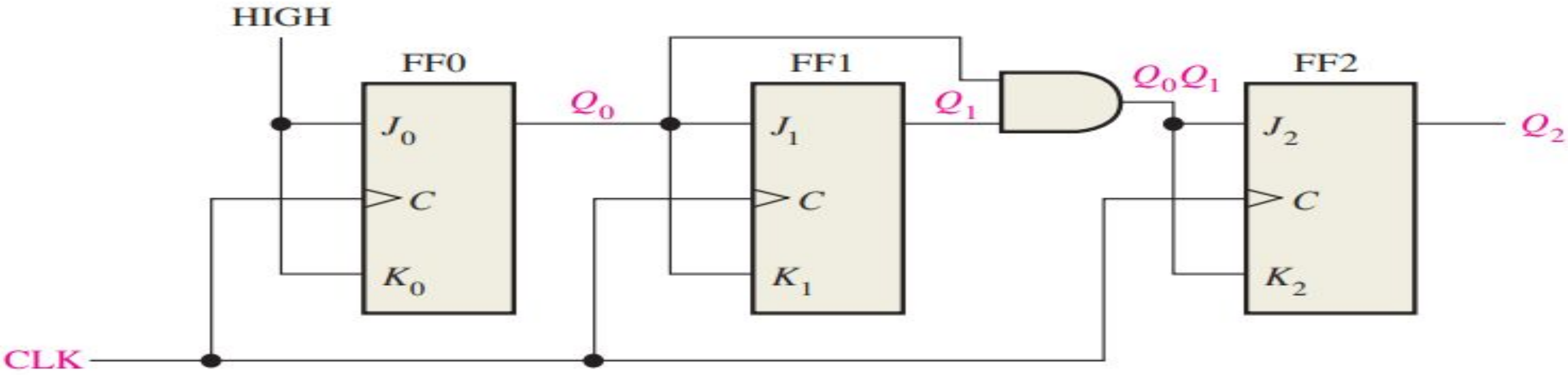


FIGURE 9–15 A 3-bit synchronous binary counter. Open file F09-15 to verify the operation.

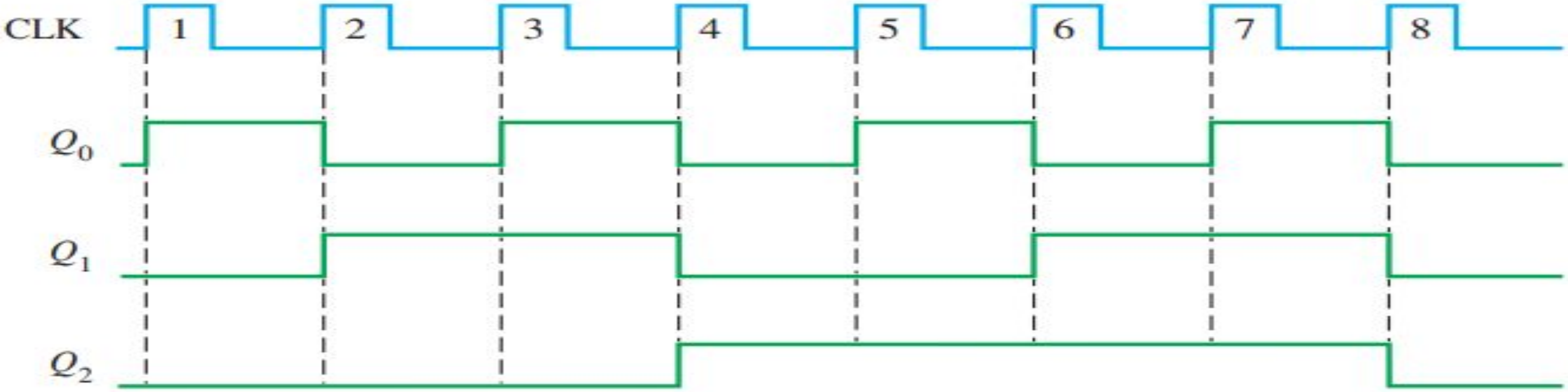


FIGURE 9–16 Timing diagram for the counter of Figure 9–15.

A 3-Bit Synchronous Binary Counter

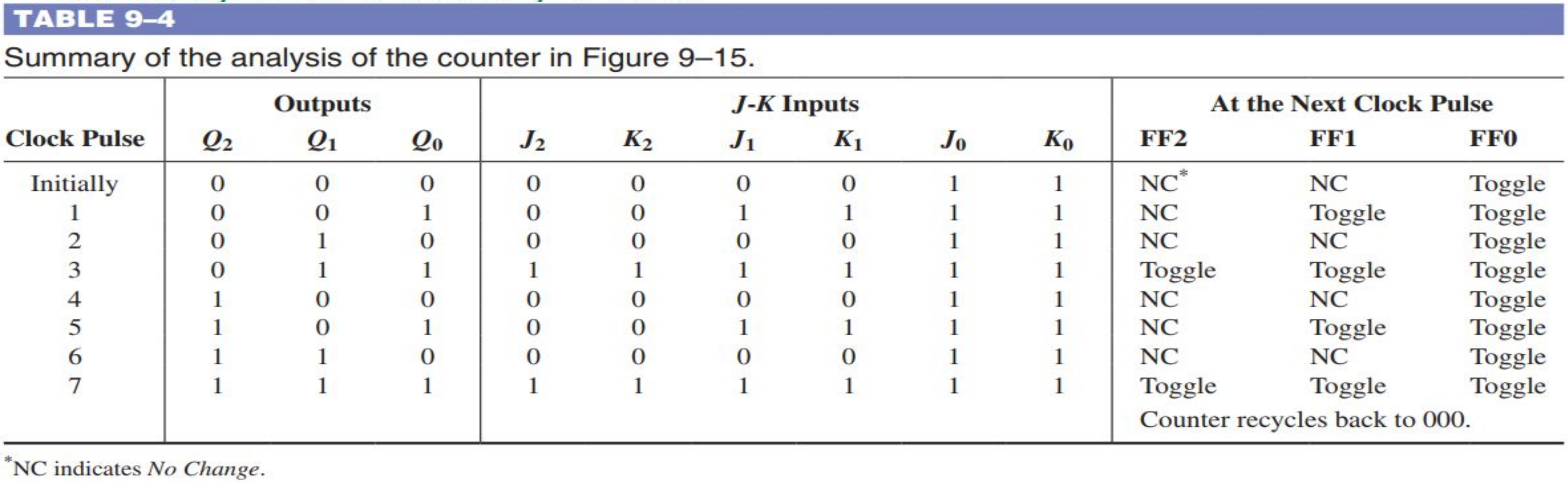


TABLE 9–4

Summary of the analysis of the counter in Figure 9–15.

Clock Pulse	Outputs			*J-K* Inputs						At the Next Clock Pulse		
	Q_2	Q_1	Q_0	J_2	K_2	J_1	K_1	J_0	K_0	FF2	FF1	FF0
Initially	0	0	0	0	0	0	0	1	1	NC*	NC	Toggle
1	0	0	1	0	0	1	1	1	1	NC	Toggle	Toggle
2	0	1	0	0	0	0	0	1	1	NC	NC	Toggle
3	0	1	1	1	1	1	1	1	1	Toggle	Toggle	Toggle
4	1	0	0	0	0	0	0	1	1	NC	NC	Toggle
5	1	0	1	0	0	1	1	1	1	NC	Toggle	Toggle
6	1	1	0	0	0	0	0	1	1	NC	NC	Toggle
7	1	1	1	1	1	1	1	1	1	Toggle	Toggle	Toggle
										Counter recycles back to 000.		

*NC indicates *No Change*.

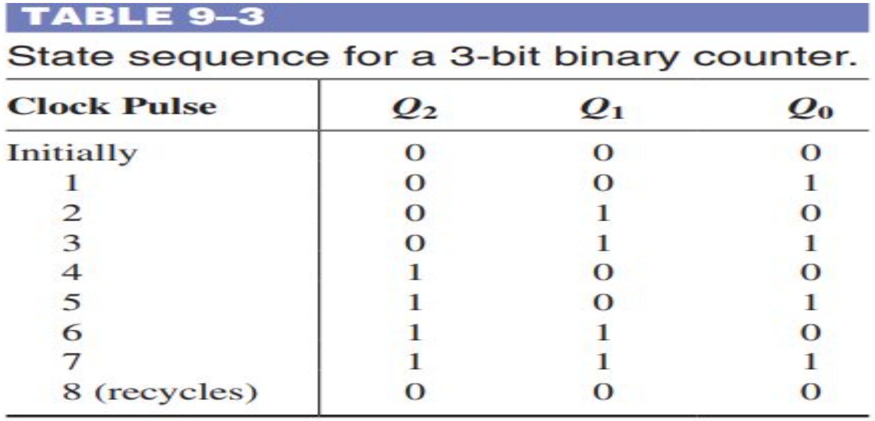


TABLE 9–3

State sequence for a 3-bit binary counter.

Clock Pulse	Q_2	Q_1	Q_0
Initially	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 (recycles)	0	0	0

A 4-Bit Synchronous Binary Counter

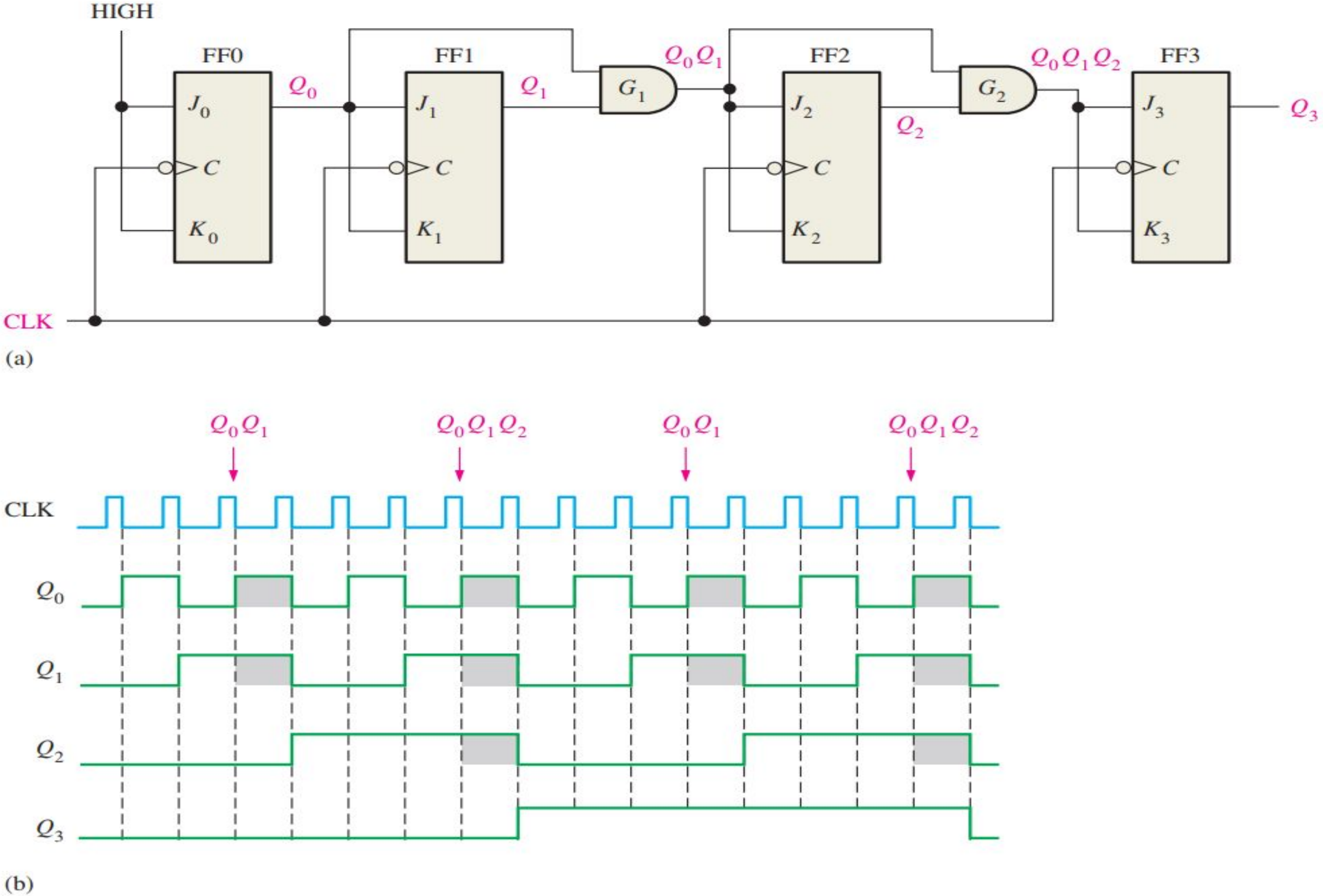


FIGURE 9–17 A 4-bit synchronous binary counter and timing diagram. Times where the AND gate outputs are HIGH are indicated by the shaded areas.

A 4-Bit Synchronous Decade Counter

As you know, a BCD decade counter exhibits a truncated binary sequence and goes from 0000 through the 1001 state. Rather than going from the 1001 state to the 1010 state, it recycles to the 0000 state. A synchronous BCD decade counter is shown in Figure 9–18. The timing diagram for the decade counter is shown in Figure 9–19.

TABLE 9–5

States of a BCD decade counter.

Clock Pulse	Q_3	Q_2	Q_1	Q_0
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 (recycles)	0	0	0	0

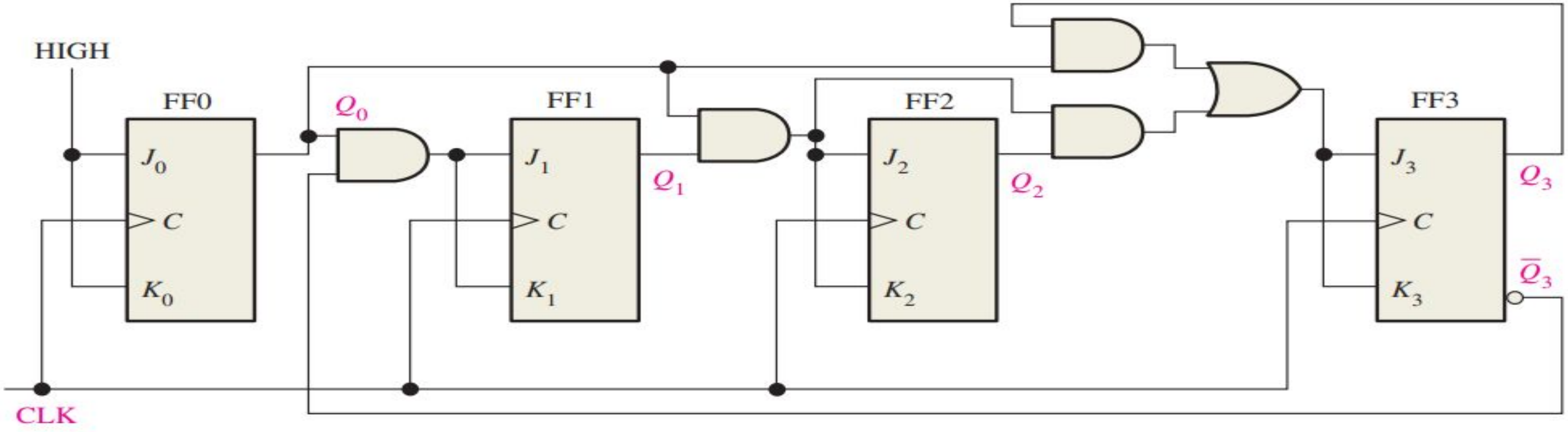


FIGURE 9–18 A synchronous BCD decade counter. Open file F09-18 to verify operation.

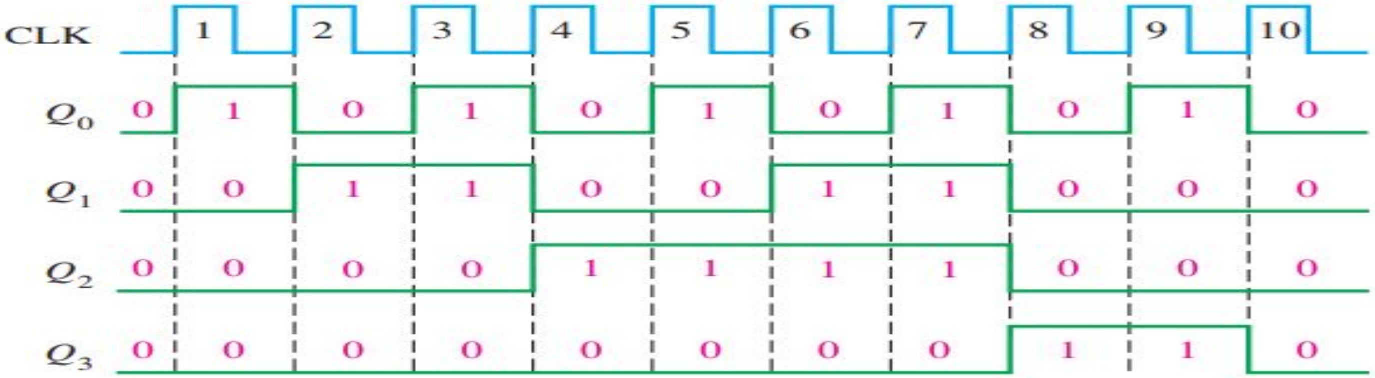
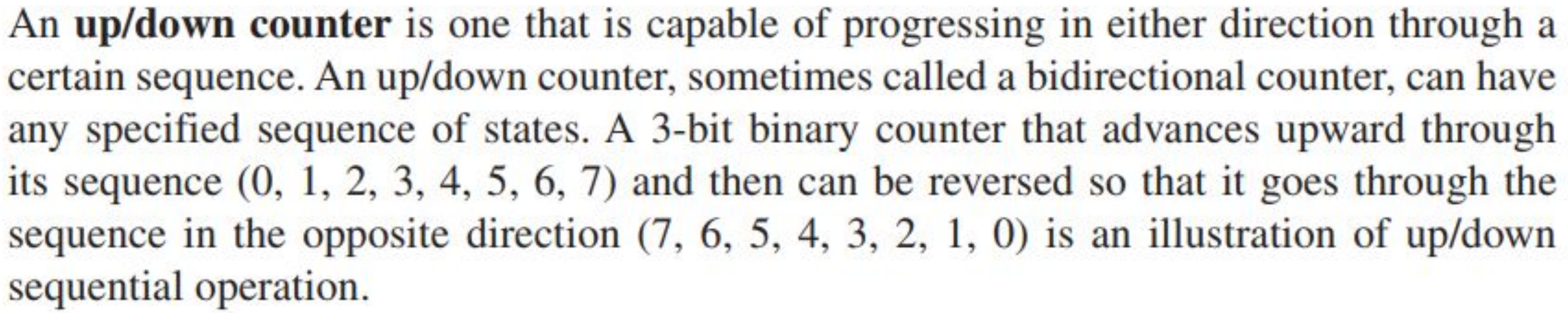
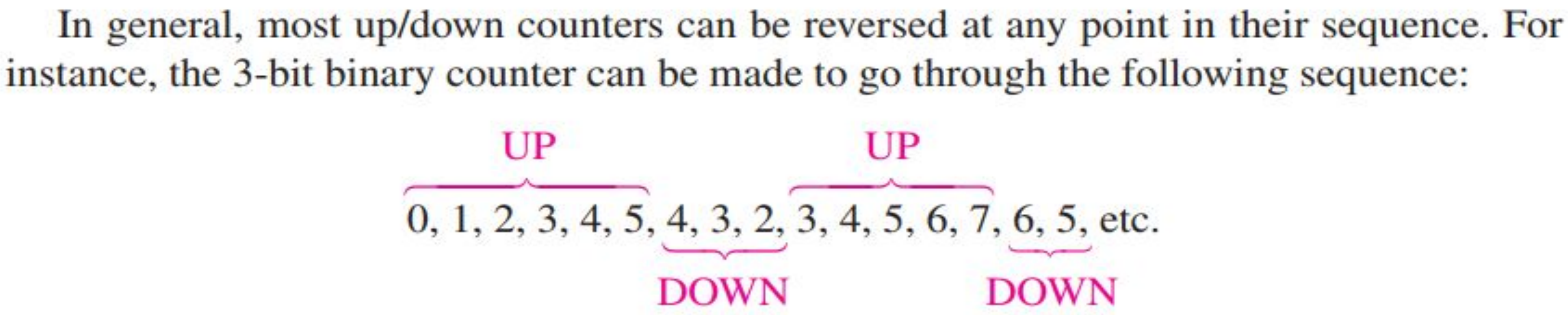


FIGURE 9–19 Timing diagram for the BCD decade counter (Q_0 is the LSB).

Up/Down Synchronous Counters



An **up/down counter** is one that is capable of progressing in either direction through a certain sequence. An up/down counter, sometimes called a bidirectional counter, can have any specified sequence of states. A 3-bit binary counter that advances upward through its sequence (0, 1, 2, 3, 4, 5, 6, 7) and then can be reversed so that it goes through the sequence in the opposite direction (7, 6, 5, 4, 3, 2, 1, 0) is an illustration of up/down sequential operation.



In general, most up/down counters can be reversed at any point in their sequence. For instance, the 3-bit binary counter can be made to go through the following sequence:

$$\overbrace{0, 1, 2, 3, 4, 5,}^{\text{UP}}\ \underbrace{4, 3, 2,}_{\text{DOWN}}\ \overbrace{3, 4, 5, 6, 7,}^{\text{UP}}\ \underbrace{6, 5,}_{\text{DOWN}}\ \text{etc.}$$

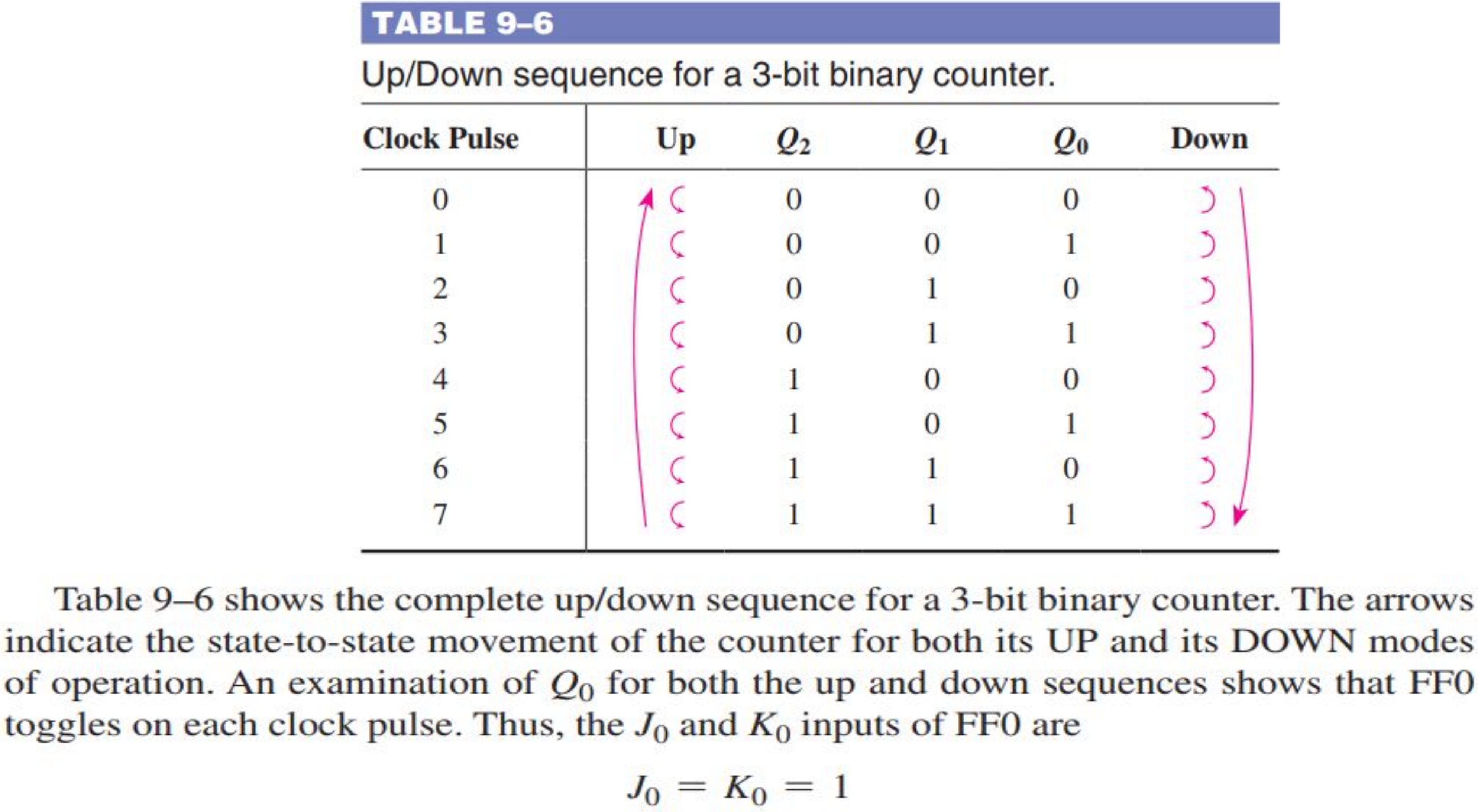


TABLE 9–6

Up/Down sequence for a 3-bit binary counter.

Clock Pulse	Up	Q_2	Q_1	Q_0	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	↓

Table 9–6 shows the complete up/down sequence for a 3-bit binary counter. The arrows indicate the state-to-state movement of the counter for both its UP and its DOWN modes of operation. An examination of Q_0 for both the up and down sequences shows that FF0 toggles on each clock pulse. Thus, the J_0 and K_0 inputs of FF0 are

$$J_0 = K_0 = 1$$

For the up sequence, Q_1 changes state on the next clock pulse when $Q_0 = 1$. For the down sequence, Q_1 changes on the next clock pulse when $Q_0 = 0$. Thus, the J_1 and K_1 inputs of FF1 must equal 1 under the conditions expressed by the following equation:

$$J_1 = K_1 = (Q_0 \cdot \mathrm{UP}) + (\overline{Q}_0 \cdot \mathrm{DOWN})$$

For the up sequence, Q_2 changes state on the next clock pulse when $Q_0 = Q_1 = 1$. For the down sequence, Q_2 changes on the next clock pulse when $Q_0 = Q_1 = 0$. Thus, the J_2 and K_2 inputs of FF2 must equal 1 under the conditions expressed by the following equation:

$$J_2 = K_2 = (Q_0 \cdot Q_1 \cdot \mathrm{UP}) + (\overline{Q}_0 \cdot \overline{Q}_1 \cdot \mathrm{DOWN})$$

Each of the conditions for the J and K inputs of each flip-flop produces a toggle at the appropriate point in the counter sequence.

Figure 9–22 shows a basic implementation of a 3-bit up/down binary counter using the logic equations just developed for the J and K inputs of each flip-flop. Notice that the $UP/\overline{DOWN}$ control input is HIGH for UP and LOW for DOWN.

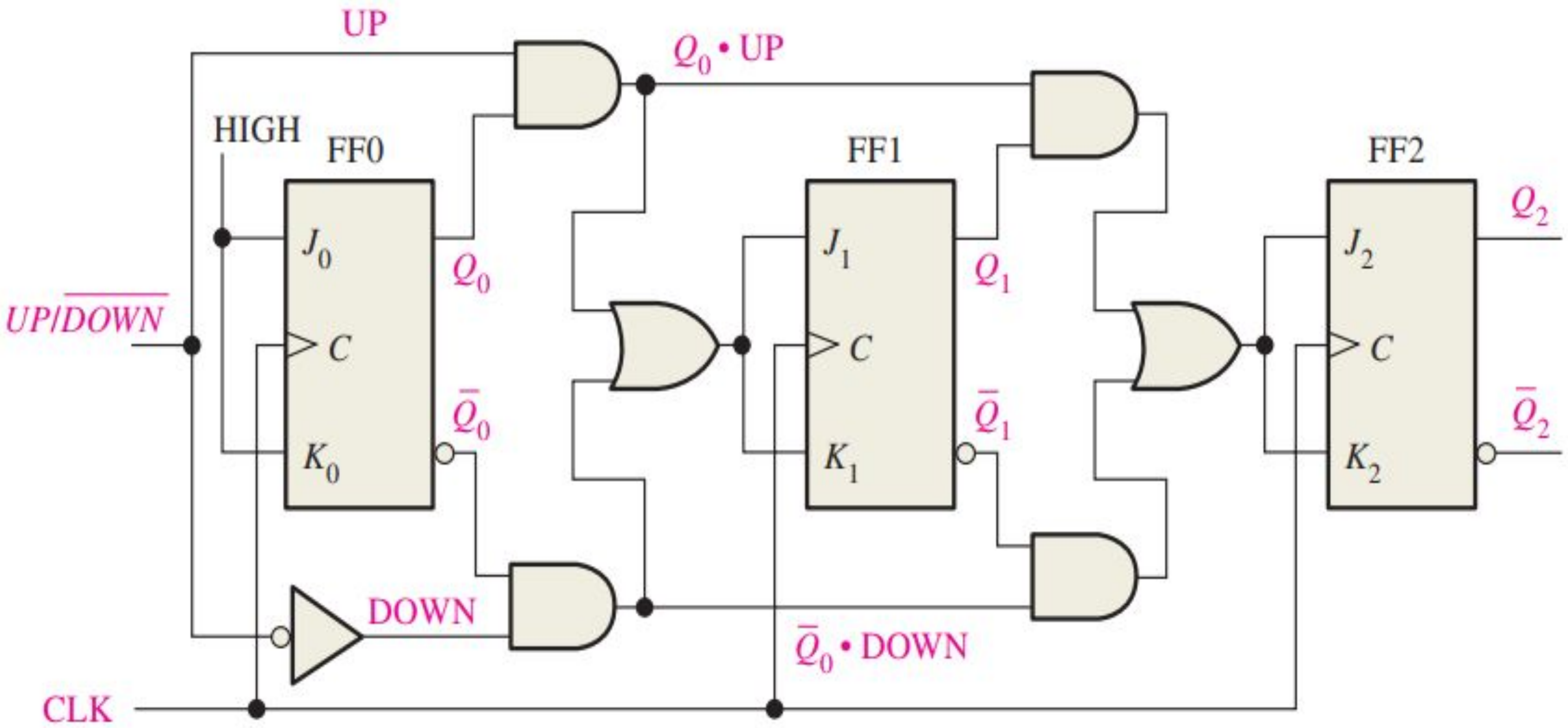
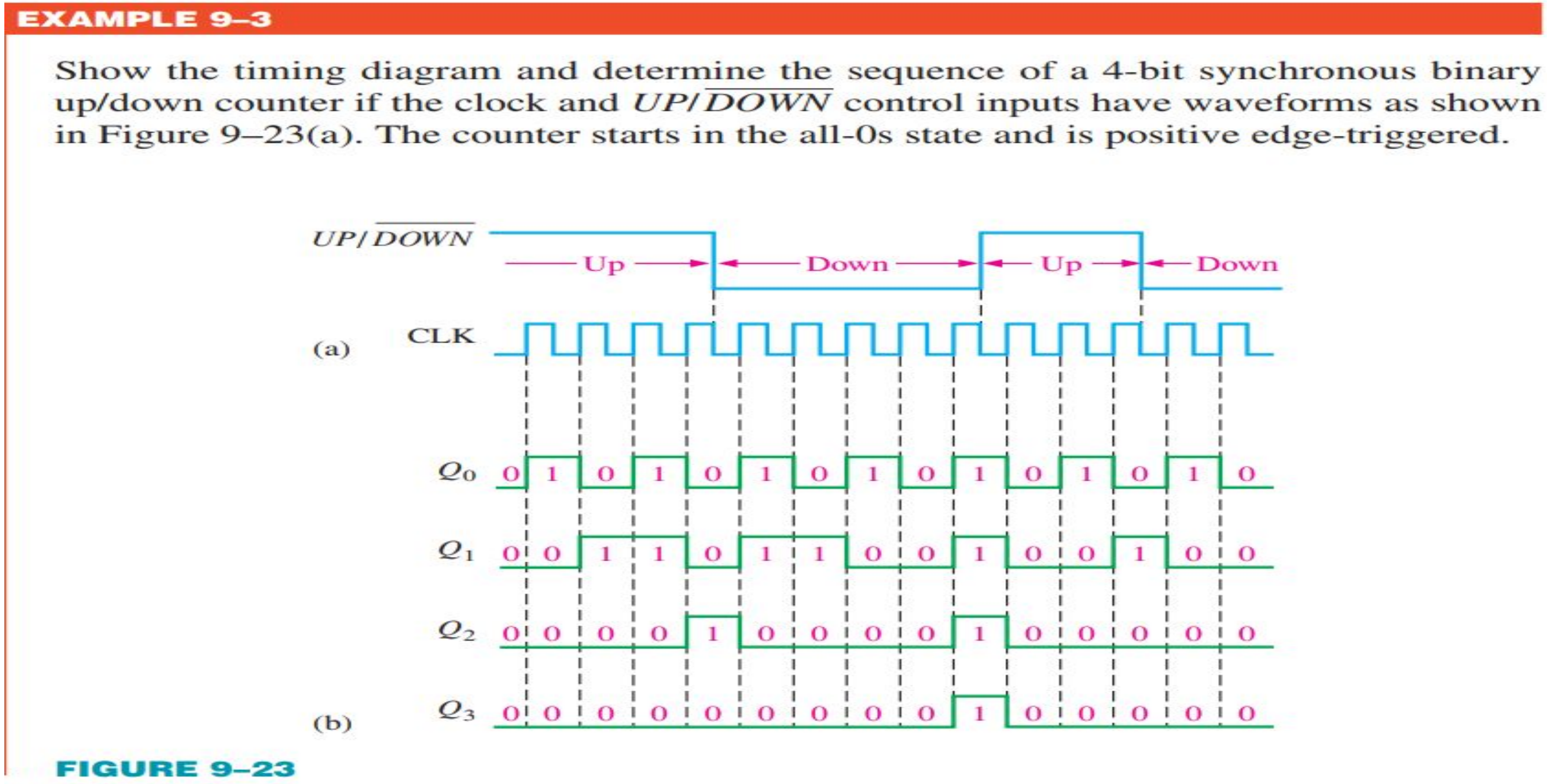


FIGURE 9–22 A basic 3-bit up/down synchronous counter. Open file F09-22 to verify operation.

Up/Down Synchronous Counter Timing Diagram



EXAMPLE 9–3

Show the timing diagram and determine the sequence of a 4-bit synchronous binary up/down counter if the clock and $UP/\overline{DOWN}$ control inputs have waveforms as shown in Figure 9–23(a). The counter starts in the all-0s state and is positive edge-triggered.

$UP/\overline{DOWN}$
Up
Down
Up
Down
(a)
CLK
Q_0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
Q_1 0 0 1 1 0 1 1 0 0 1 0 0 1 0 0
Q_2 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0
Q_3 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
(b)

FIGURE 9–23

Solution

The timing diagram showing the Q outputs is shown in Figure 9–23(b). From these waveforms, the counter sequence is as shown in Table 9–7.

TABLE 9–7

Q_3	Q_2	Q_1	Q_0	
0	0	0	0	UP
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	0	1	1	DOWN
0	0	1	0	
0	0	0	1	
0	0	0	0	
1	1	1	1	
0	0	0	0	UP
0	0	0	1	
0	0	1	0	
0	0	0	1	DOWN
0	0	0	0	

Related Problem

Show the timing diagram if the $UP/\overline{DOWN}$ control waveform in Figure 9–23(a) is inverted.

Design of Synchronous Counters

Step 1: State Diagram

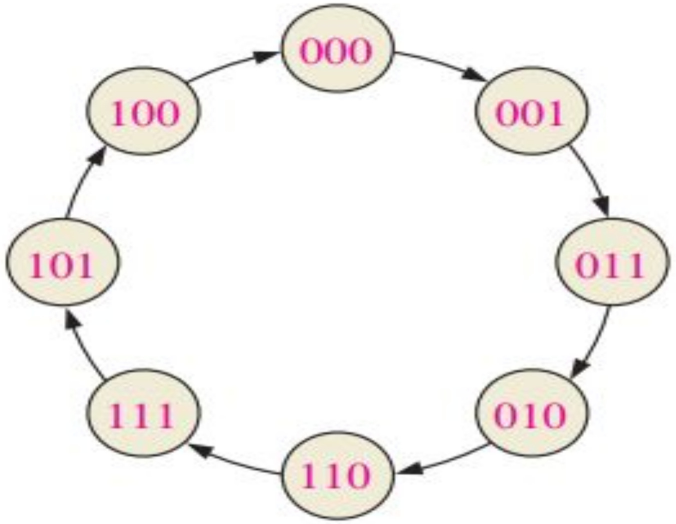


FIGURE 9–26 State diagram for a 3-bit Gray code counter.

Step 2: Next-State Table

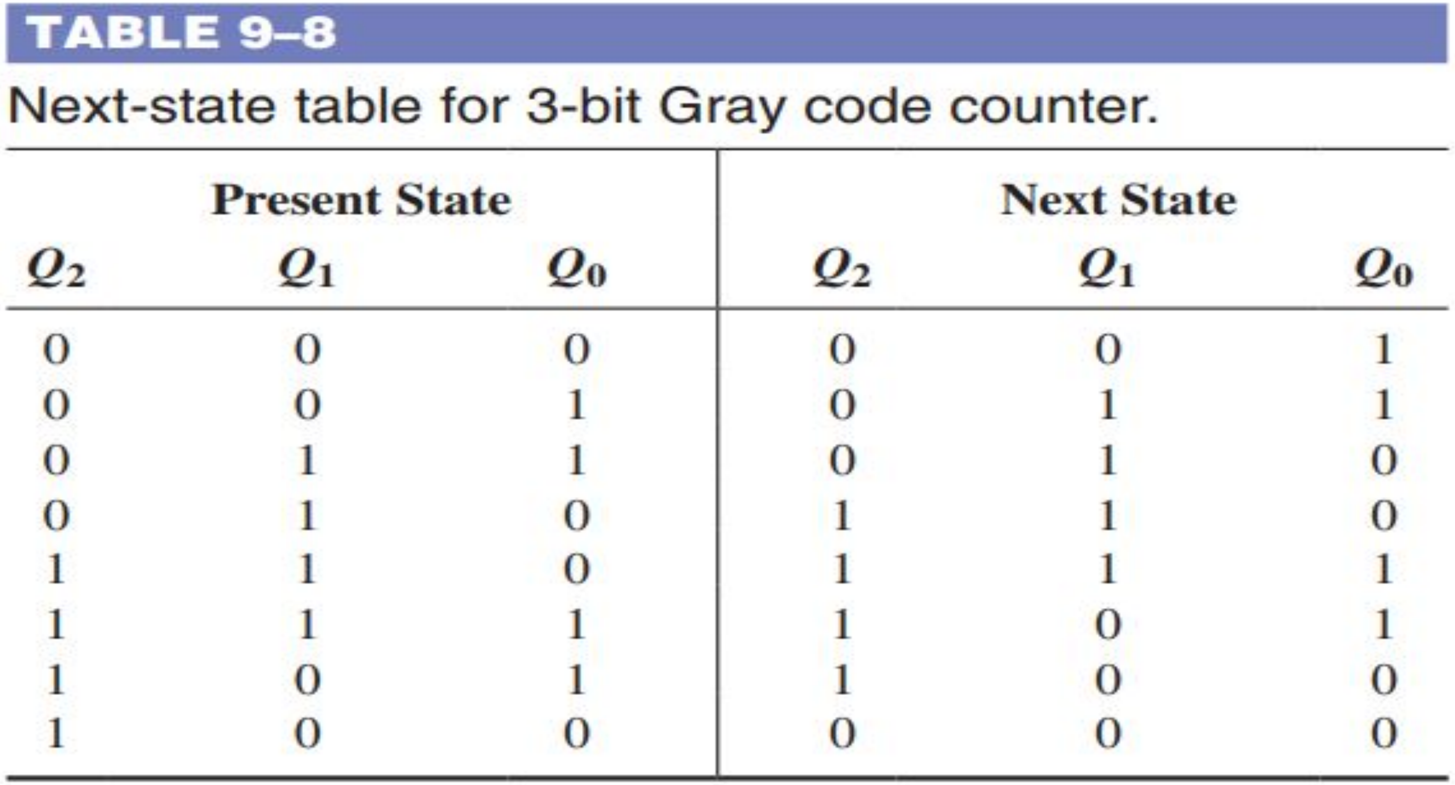


TABLE 9–8

Next-state table for 3-bit Gray code counter.

Present State			Next State		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0

Step 3: Flip-Flop Transition Table

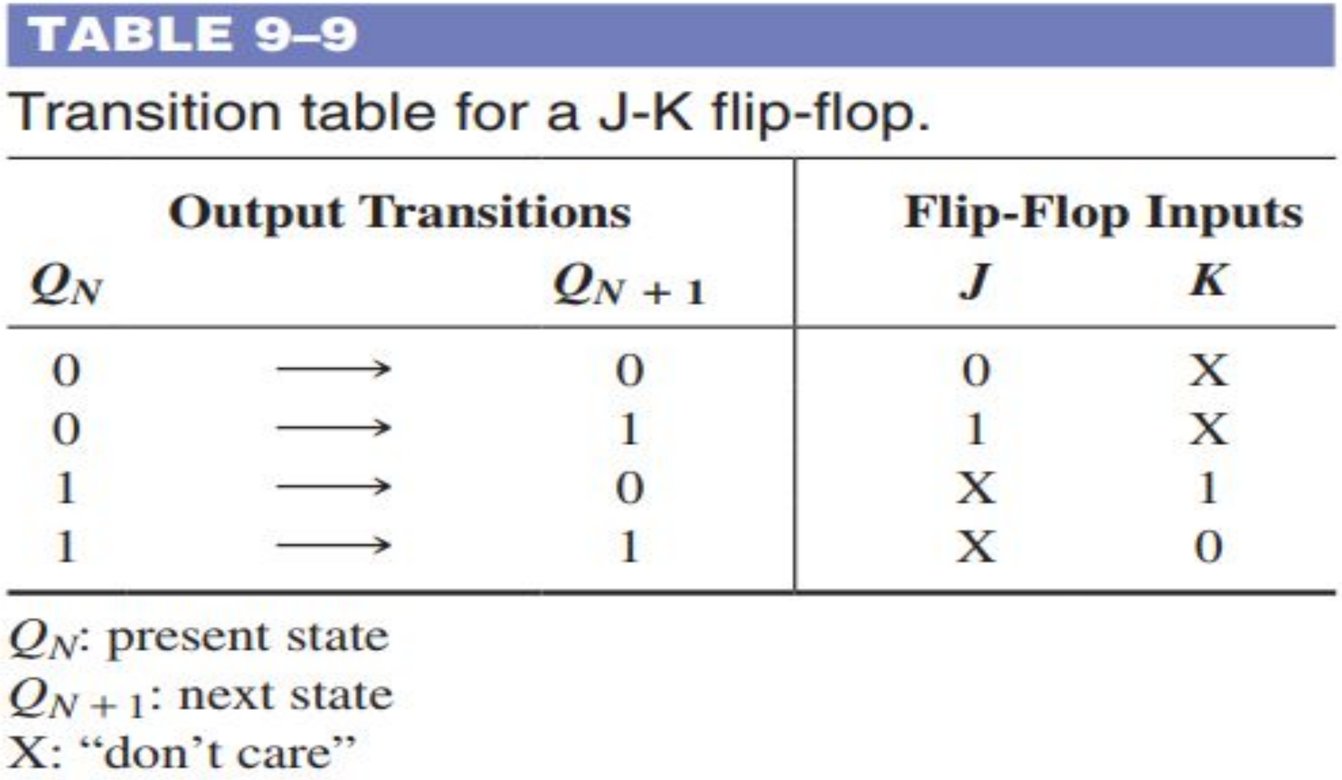


TABLE 9–9

Transition table for a J-K flip-flop.

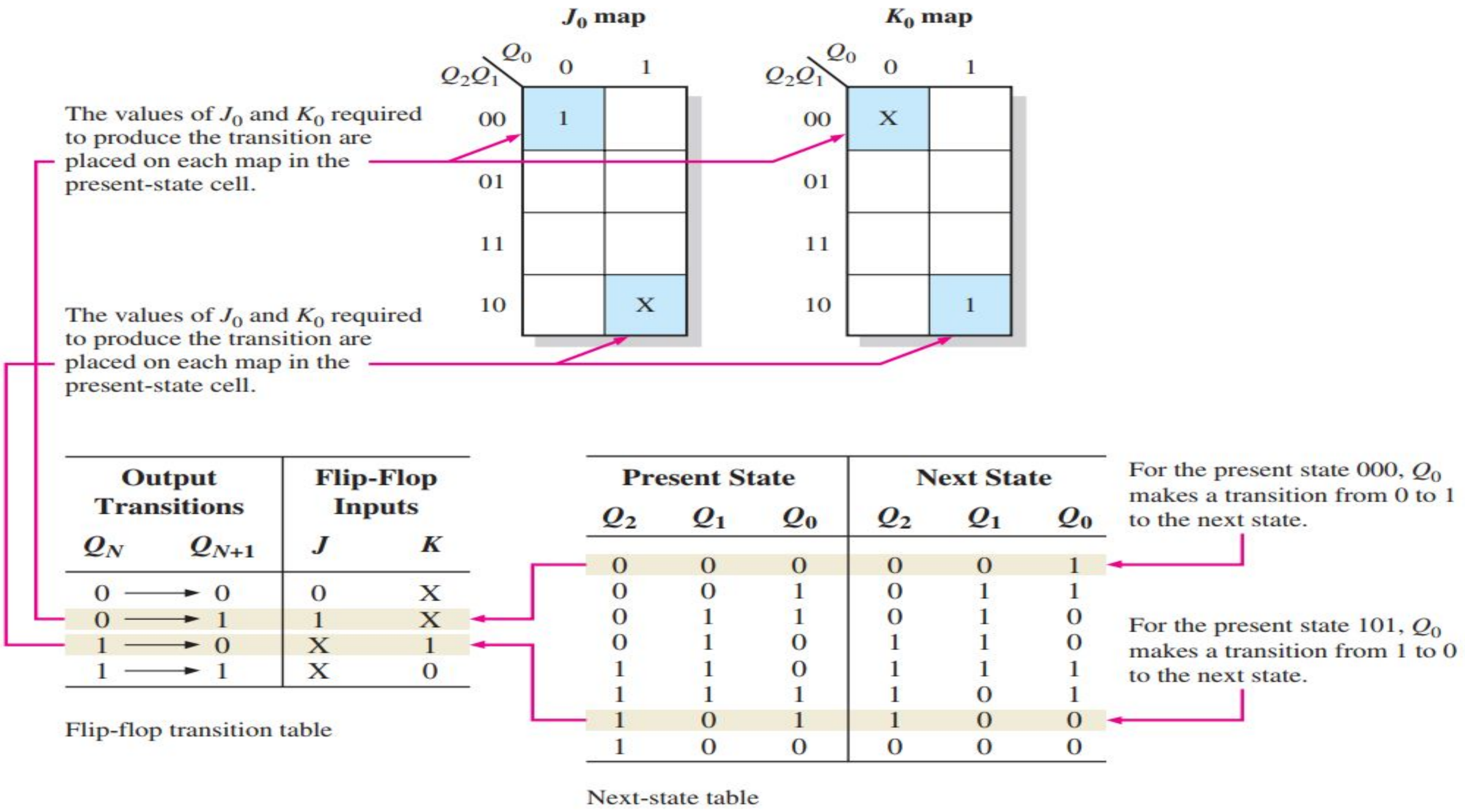
Output Transitions			Flip-Flop Inputs	
Q_N		Q_{N+1}	J	K
0	⟶	0	0	X
0	⟶	1	1	X
1	⟶	0	X	1
1	⟶	1	X	0

Q_N: present state

Q_{N+1}: next state

X: "don't care"

Step 4: Karnaugh Maps



J_0 map

Q_2Q_1 \ Q_0	0	1
00	1	
01		
11		
10		X

K_0 map

Q_2Q_1 \ Q_0	0	1
00	X	
01		
11		
10		1

The values of J_0 and K_0 required to produce the transition are placed on each map in the present-state cell.

The values of J_0 and K_0 required to produce the transition are placed on each map in the present-state cell.

Output Transitions		Flip-Flop Inputs	
Q_N	Q_{N+1}	J	K
0 →	0	0	X
0 →	1	1	X
1 →	0	X	1
1 →	1	X	0

Flip-flop transition table

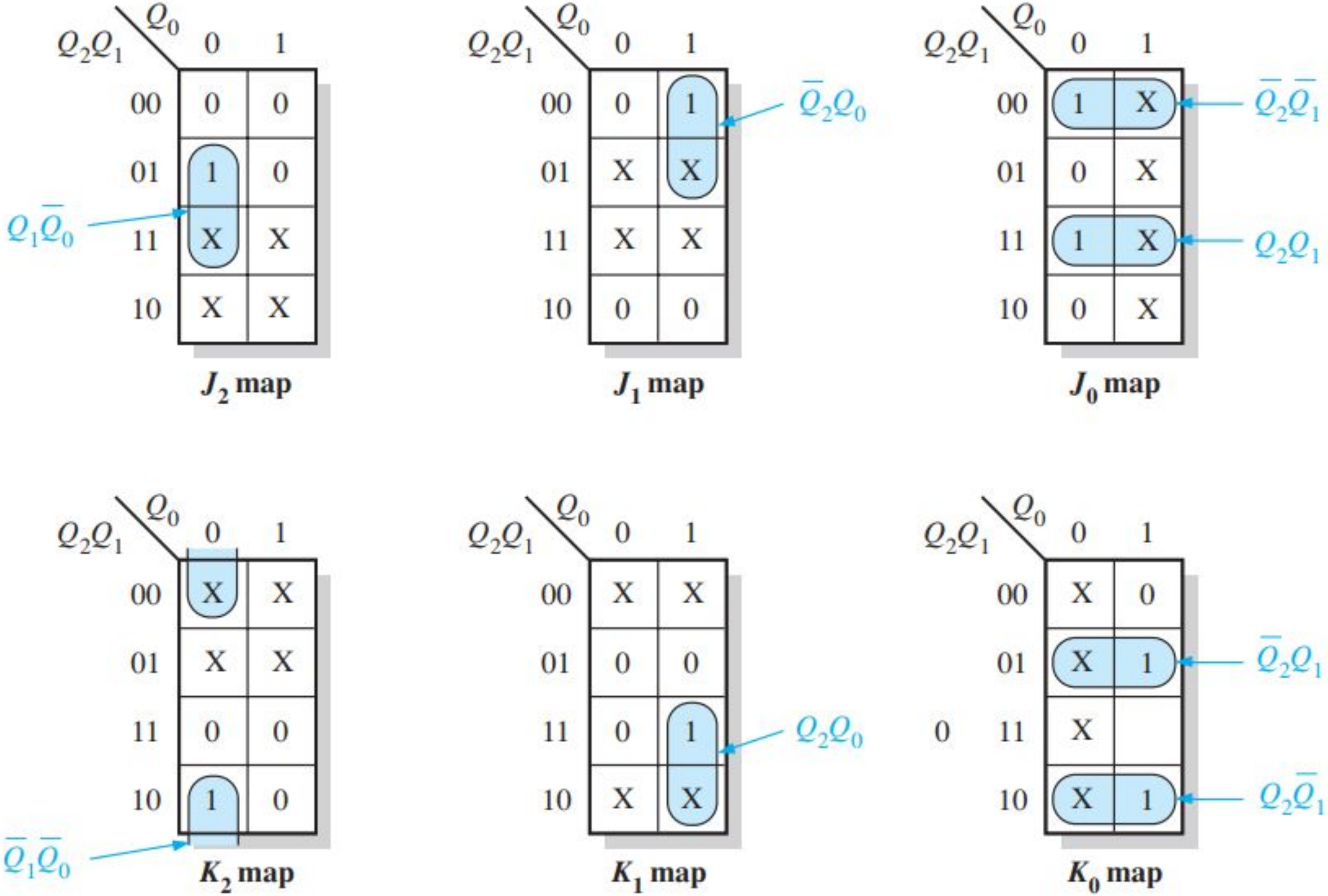
Present State			Next State		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0

Next-state table

For the present state 000, Q_0 makes a transition from 0 to 1 to the next state.

For the present state 101, Q_0 makes a transition from 1 to 0 to the next state.

FIGURE 9–27 Examples of the mapping procedure for the counter sequence represented in Table 9–8 and Table 9–9.



Q_2Q_1 \ Q_0	0	1
00	0	0
01	1	0
11	X	X
10	X	X

***J_2* map**

Q_2Q_1 \ Q_0	0	1
00	0	1
01	X	X
11	X	X
10	0	0

***J_1* map**

Q_2Q_1 \ Q_0	0	1
00	1	X
01	0	X
11	1	X
10	0	X

***J_0* map**

Q_2Q_1 \ Q_0	0	1
00	X	X
01	X	X
11	0	0
10	1	0

***K_2* map**

Q_2Q_1 \ Q_0	0	1
00	X	X
01	0	0
11	0	1
10	X	X

***K_1* map**

Q_2Q_1 \ Q_0	0	1
00	X	0
01	X	1
11	X	
10	X	1

***K_0* map**

FIGURE 9–28 Karnaugh maps for present-state *J* and *K* inputs.

Step 5: Logic Expressions for Flip-Flop Inputs

$$
\begin{aligned}
J_0 &= Q_2Q_1 + \overline{Q}_2\overline{Q}_1 = \overline{Q_2 \oplus Q_1} \\
K_0 &= Q_2\overline{Q}_1 + \overline{Q}_2Q_1 = Q_2 \oplus Q_1 \\
J_1 &= \overline{Q}_2Q_0 \\
K_1 &= Q_2Q_0 \\
J_2 &= Q_1\overline{Q}_0 \\
K_2 &= \overline{Q}_1\overline{Q}_0
\end{aligned}
$$

Step 6: Counter Implementation

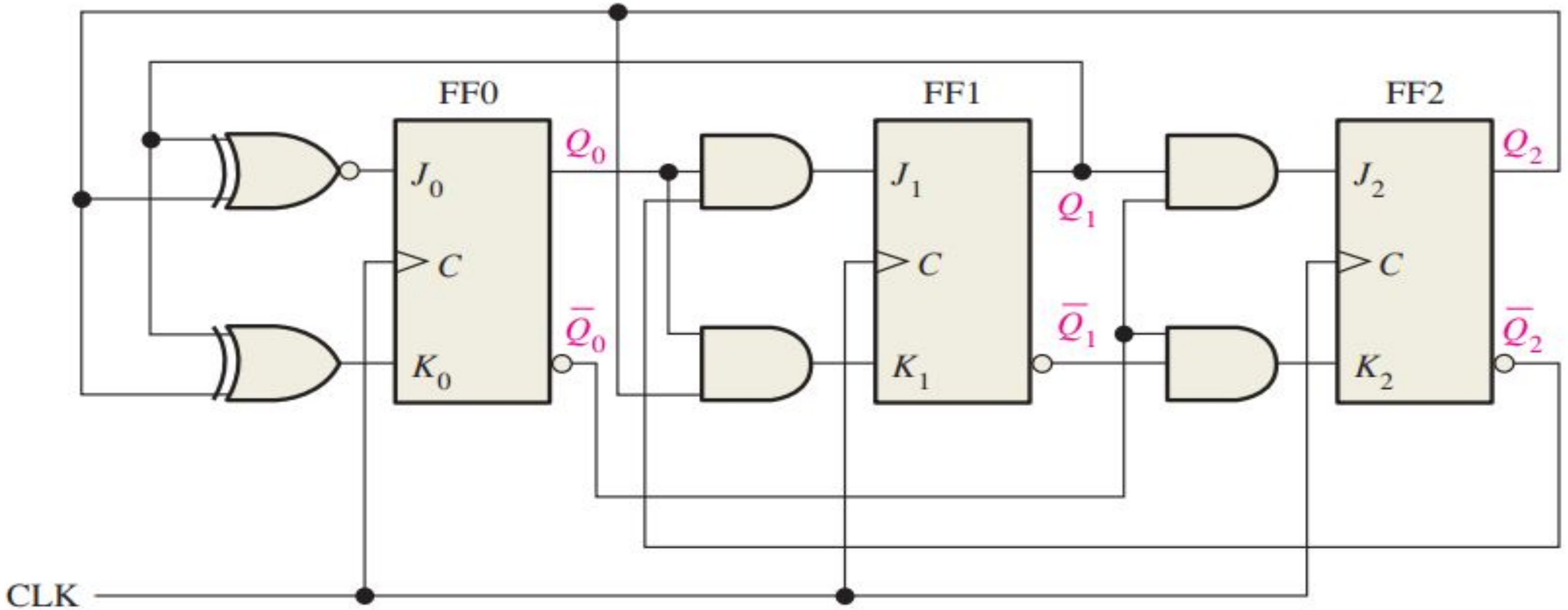
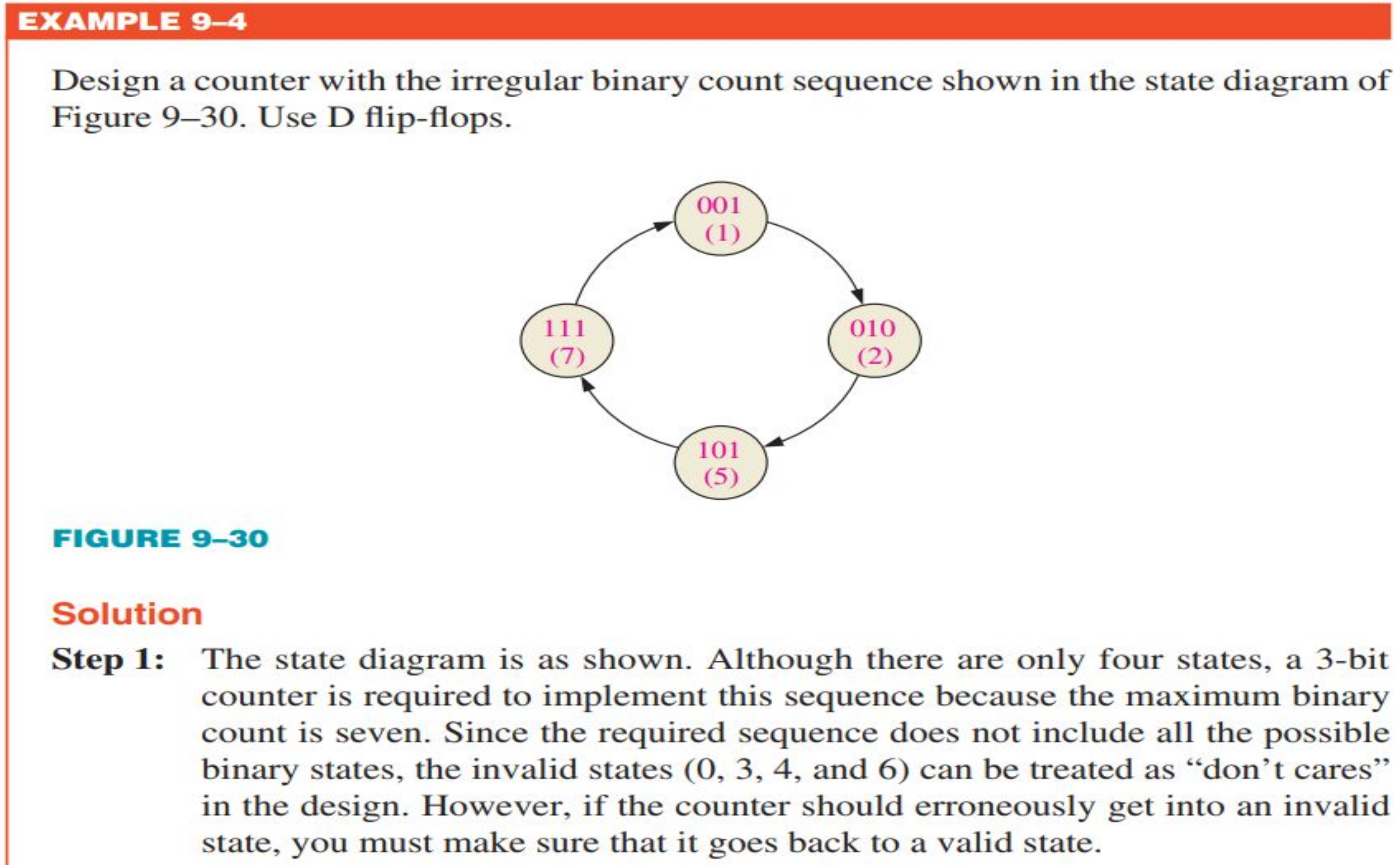


FIGURE 9–29 Three-bit Gray code counter. Open file F09-29 to verify operation.

Example



EXAMPLE 9–4

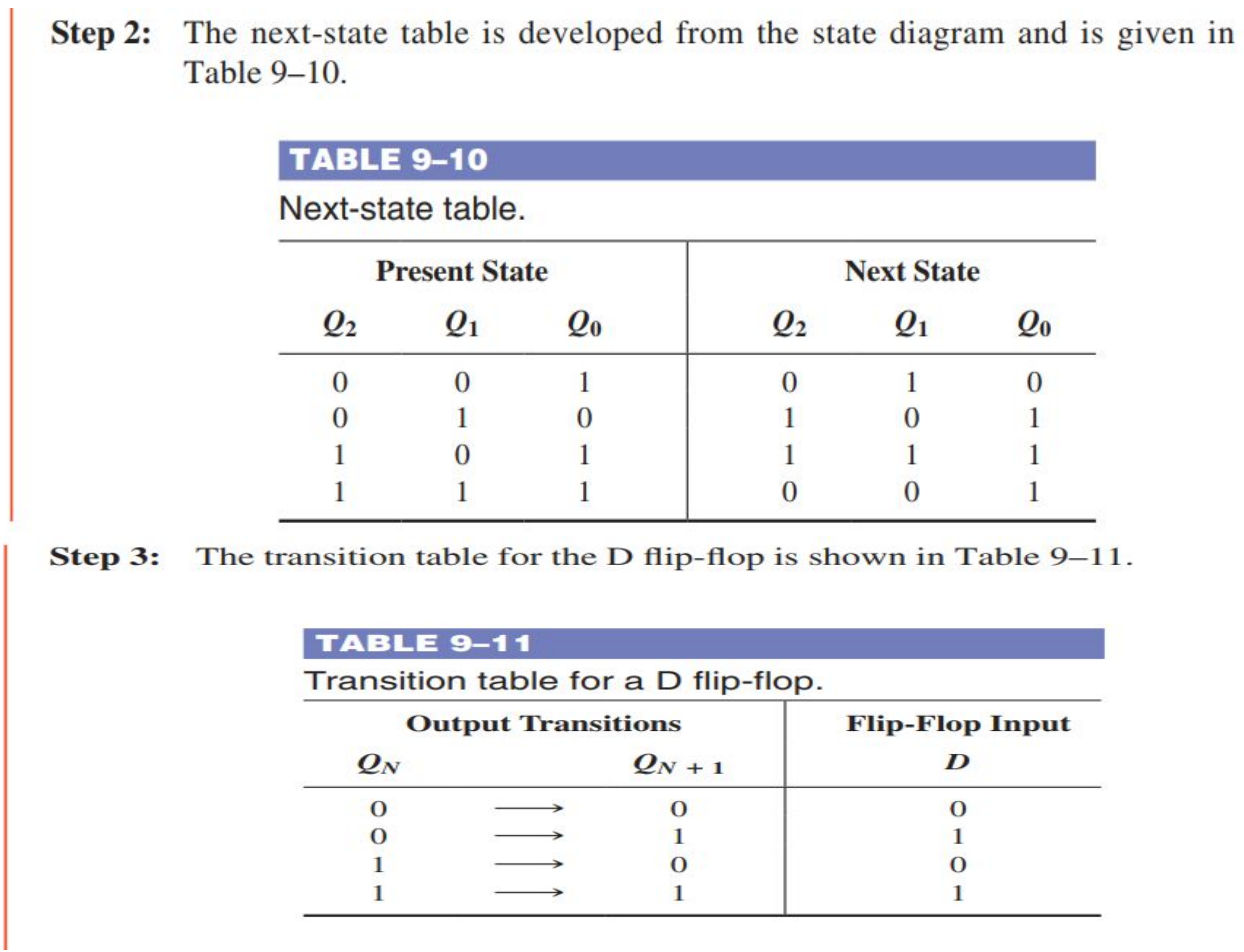
Design a counter with the irregular binary count sequence shown in the state diagram of Figure 9–30. Use D flip-flops.

001
(1)
010
(2)
101
(5)
111
(7)

FIGURE 9–30

Solution

Step 1: The state diagram is as shown. Although there are only four states, a 3-bit counter is required to implement this sequence because the maximum binary count is seven. Since the required sequence does not include all the possible binary states, the invalid states (0, 3, 4, and 6) can be treated as "don't cares" in the design. However, if the counter should erroneously get into an invalid state, you must make sure that it goes back to a valid state.



Step 2: The next-state table is developed from the state diagram and is given in Table 9–10.

TABLE 9–10

Next-state table.

Present State			Next State		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	1	0	1	0
0	1	0	1	0	1
1	0	1	1	1	1
1	1	1	0	0	1

Step 3: The transition table for the D flip-flop is shown in Table 9–11.

TABLE 9–11

Transition table for a D flip-flop.

Output Transitions			Flip-Flop Input
Q_N		Q_{N+1}	D
0	$\longrightarrow$	0	0
0	$\longrightarrow$	1	1
1	$\longrightarrow$	0	0
1	$\longrightarrow$	1	1

Step 4: The *D* inputs are plotted on the present-state Karnaugh maps in Figure 9–31. Also “don’t cares” can be placed in the cells corresponding to the invalid states of 000, 011, 100, and 110, as indicated by the red Xs.

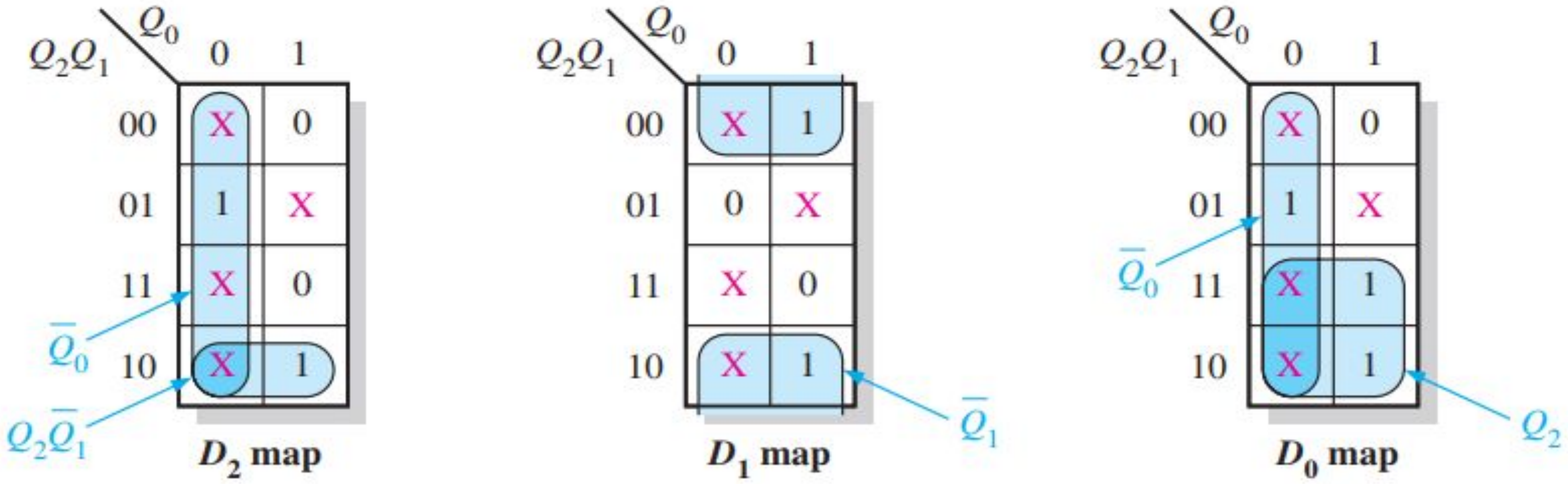


FIGURE 9–31

Step 5: Group the 1s, taking advantage of as many of the "don't care" states as possible for maximum simplification, as shown in Figure 9–31. The expression for each D input taken from the maps is as follows:

$$D_0 = \overline{Q}_0 + Q_2$$
$$D_1 = \overline{Q}_1$$
$$D_2 = \overline{Q}_0 + Q_2\overline{Q}_1$$

Step 6: The implementation of the counter is shown in Figure 9–32.

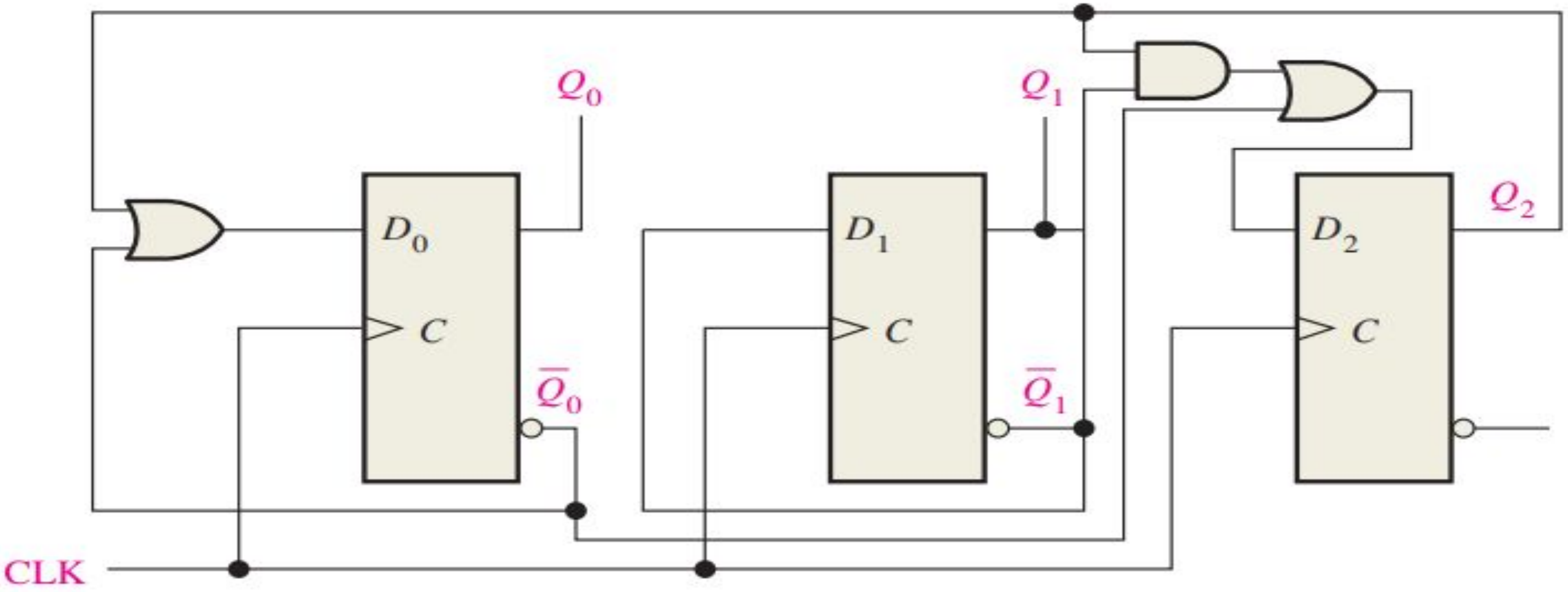
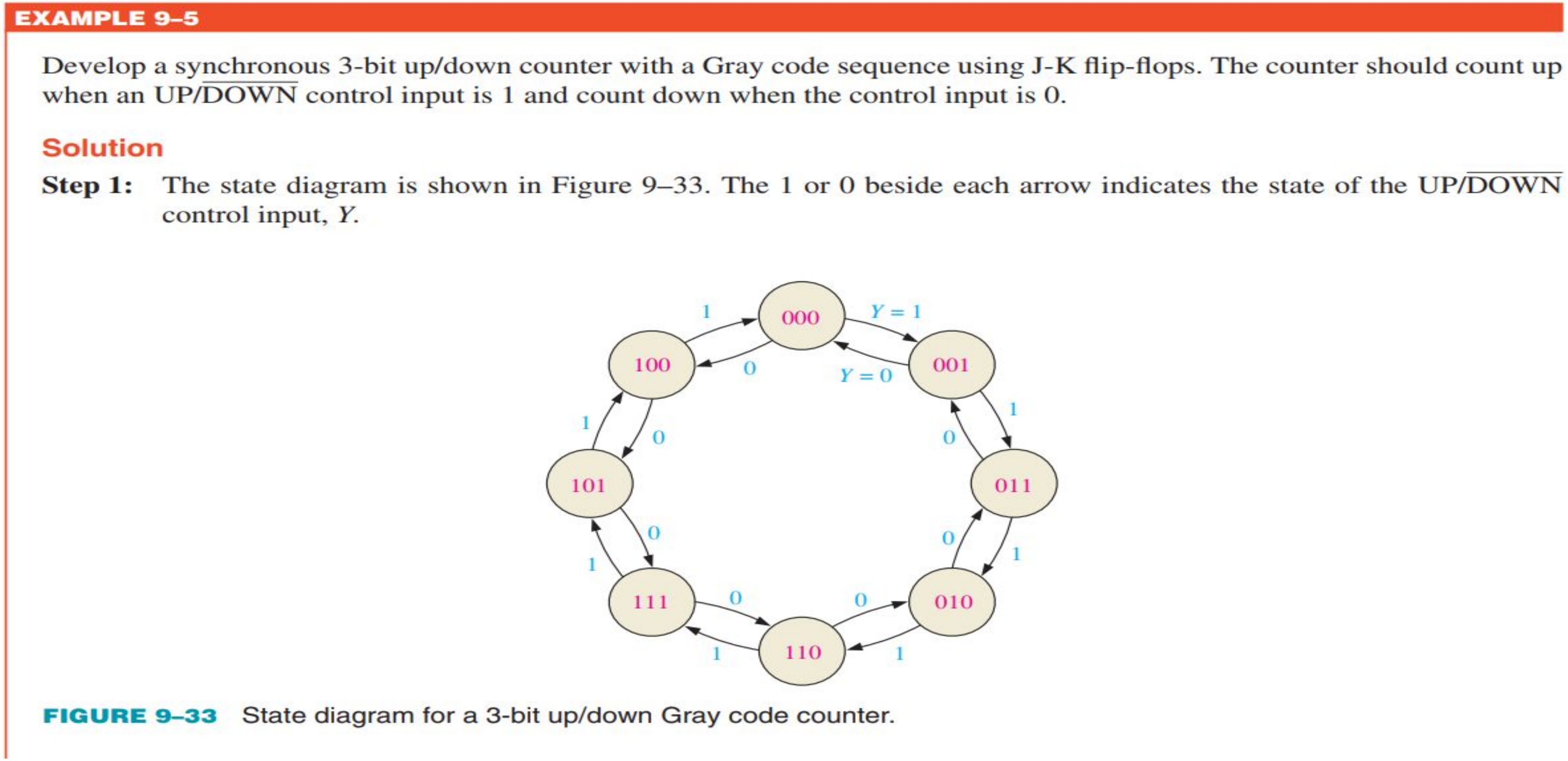


FIGURE 9–32

Up/Down Counter Design



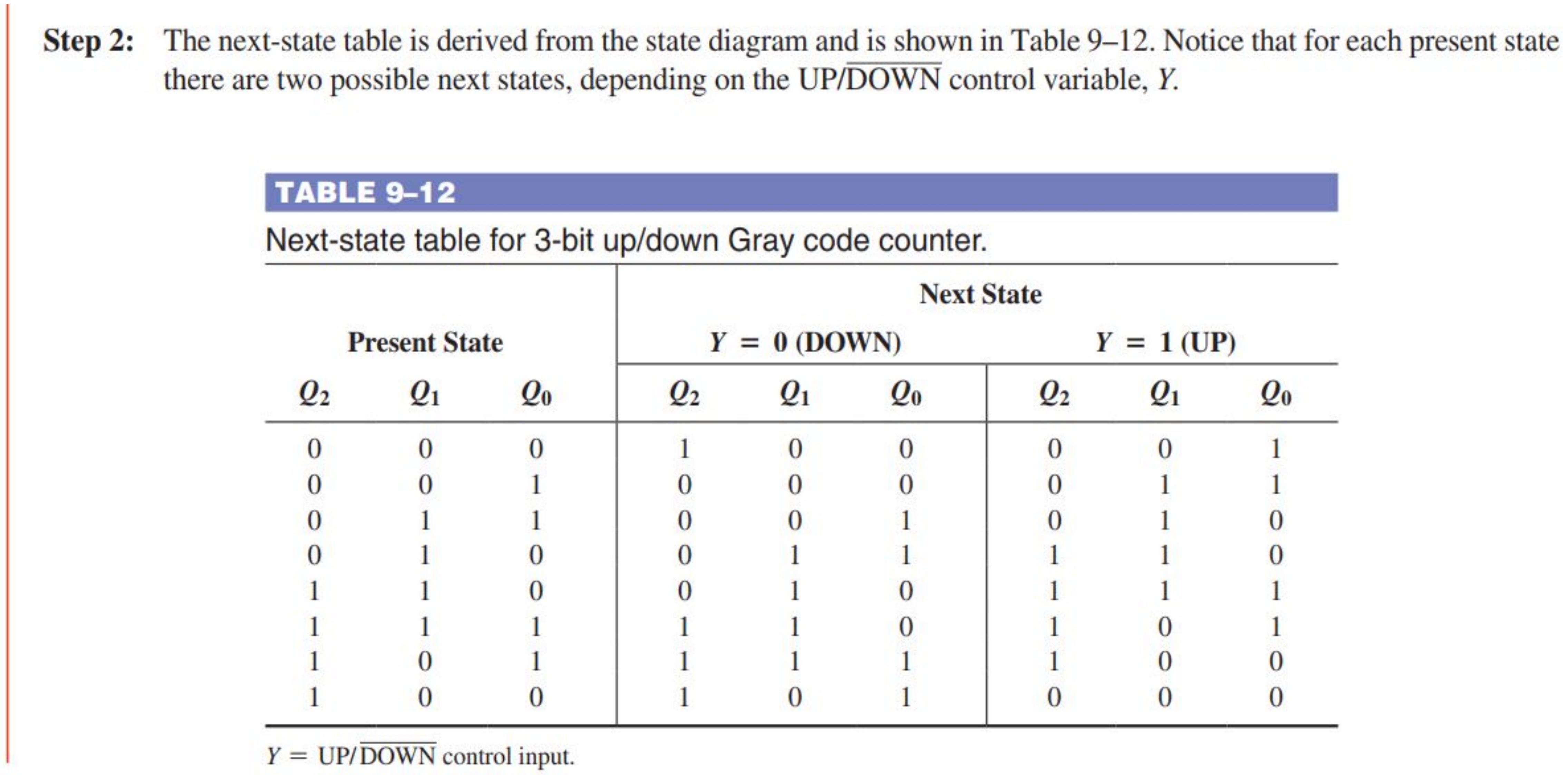
EXAMPLE 9–5

Develop a synchronous 3-bit up/down counter with a Gray code sequence using J-K flip-flops. The counter should count up when an UP/$\overline{\text{DOWN}}$ control input is 1 and count down when the control input is 0.

Solution

Step 1: The state diagram is shown in Figure 9–33. The 1 or 0 beside each arrow indicates the state of the UP/$\overline{\text{DOWN}}$ control input, *Y*.

FIGURE 9–33 State diagram for a 3-bit up/down Gray code counter.



Step 2: The next-state table is derived from the state diagram and is shown in Table 9–12. Notice that for each present state there are two possible next states, depending on the UP/$\overline{\text{DOWN}}$ control variable, Y.

TABLE 9–12

Next-state table for 3-bit up/down Gray code counter.

Present State			Next State					
			$Y = 0$ (DOWN)			$Y = 1$ (UP)		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1	1
0	1	1	0	0	1	0	1	0
0	1	0	0	1	1	1	1	0
1	1	0	0	1	0	1	1	1
1	1	1	1	1	0	1	0	1
1	0	1	1	1	1	1	0	0
1	0	0	1	0	1	0	0	0

Y = UP/$\overline{\text{DOWN}}$ control input.

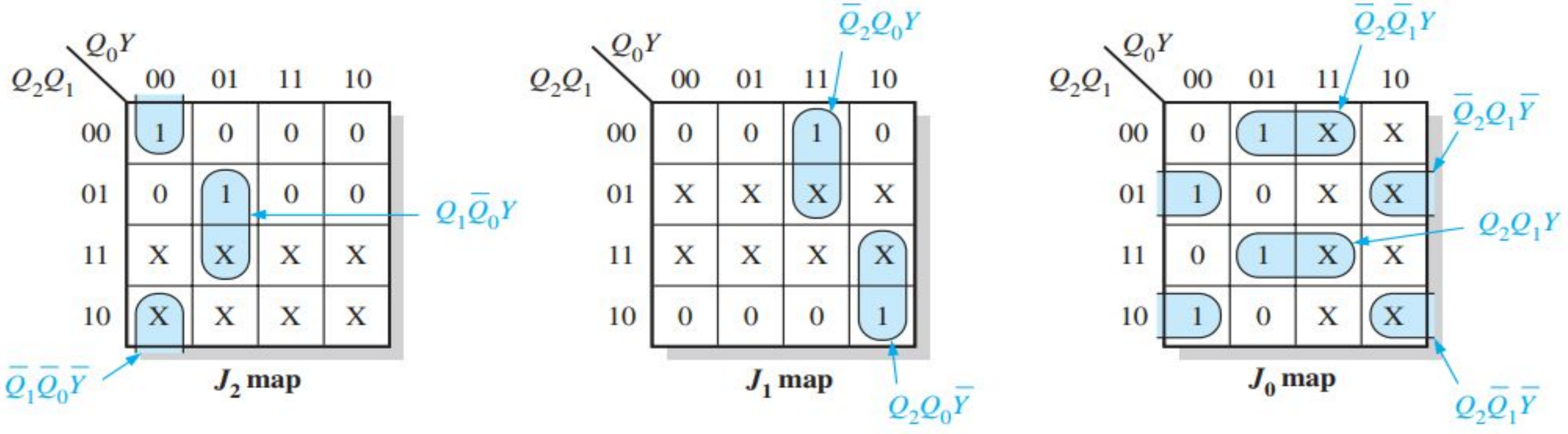
Step 3: The transition table for the J-K flip-flops is repeated in Table 9–13.

TABLE 9–13

Transition table for a J-K flip-flop.

Output Transitions			Flip-Flop Inputs	
Q_N		Q_{N+1}	J	K
0	⟶	0	0	X
0	⟶	1	1	X
1	⟶	0	X	1
1	⟶	1	X	0

Step 4: The Karnaugh maps for the J and K inputs of the flip-flops are shown in Figure 9–34. The UP/$\overline{\text{DOWN}}$ control input, Y, is considered one of the state variables along with Q_0, Q_1, and Q_2. Using the next-state table, the information in the "Flip-Flop Inputs" column of Table 9–13 is transferred onto the maps as indicated for each present state of the counter.



Q_2Q_1 \ Q_0Y	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	X	X	X	X
10	X	X	X	X

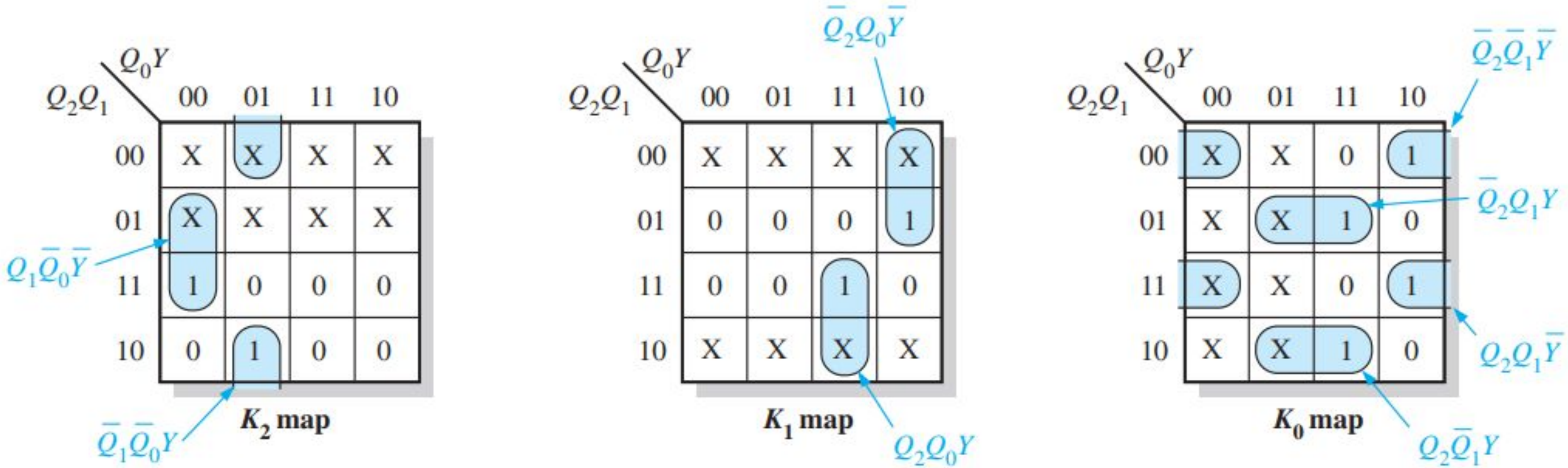
J_2 map

Q_2Q_1 \ Q_0Y	00	01	11	10
00	0	0	1	0
01	X	X	X	X
11	X	X	X	X
10	0	0	0	1

J_1 map

Q_2Q_1 \ Q_0Y	00	01	11	10
00	0	1	X	X
01	1	0	X	X
11	0	1	X	X
10	1	0	X	X

J_0 map



Q_2Q_1 \ Q_0Y	00	01	11	10
00	X	X	X	X
01	X	X	X	X
11	1	0	0	0
10	0	1	0	0

K_2 map

$Q_1\overline{Q}_0\overline{Y}$
$\overline{Q}_1\overline{Q}_0Y$

Q_2Q_1 \ Q_0Y	00	01	11	10
00	X	X	X	X
01	0	0	0	1
11	0	0	1	0
10	X	X	X	X

K_1 map

$\overline{Q}_2Q_0\overline{Y}$
Q_2Q_0Y

Q_2Q_1 \ Q_0Y	00	01	11	10
00	X	X	0	1
01	X	X	1	0
11	X	X	0	1
10	X	X	1	0

K_0 map

$\overline{Q}_2\overline{Q}_1\overline{Y}$
$\overline{Q}_2Q_1Y$
$Q_2Q_1\overline{Y}$
$Q_2\overline{Q}_1Y$

FIGURE 9–34 *J* and *K* maps for Table 9–12. The UP/$\overline{\text{DOWN}}$ control input, *Y*, is treated as a fourth variable.

Step 5: The 1s are combined in the largest possible groupings, with "don't cares" (Xs) used where possible. The groups are factored, and the expressions for the *J* and *K* inputs are as follows:

$$J_0 = Q_2Q_1Y + Q_2\overline{Q}_1\overline{Y} + \overline{Q}_2\overline{Q}_1Y + \overline{Q}_2Q_1\overline{Y} \qquad K_0 = \overline{Q}_2\overline{Q}_1\overline{Y} + \overline{Q}_2Q_1Y + Q_2\overline{Q}_1Y + Q_2Q_1\overline{Y}$$

$$J_1 = \overline{Q}_2Q_0Y + Q_2Q_0\overline{Y} \qquad K_1 = \overline{Q}_2Q_0\overline{Y} + Q_2Q_0Y$$

$$J_2 = Q_1\overline{Q}_0Y + \overline{Q}_1\overline{Q}_0\overline{Y} \qquad K_2 = Q_1\overline{Q}_0\overline{Y} + \overline{Q}_1\overline{Q}_0Y$$

Step 6: The *J* and *K* equations are implemented with combinational logic. This step is the Related Problem.