Digital Electronics

Counters

Count: the number of events

Counter: connecting flip-flops by a proper manner

Types of counters

Synchronous Counters

Asynchronous counter

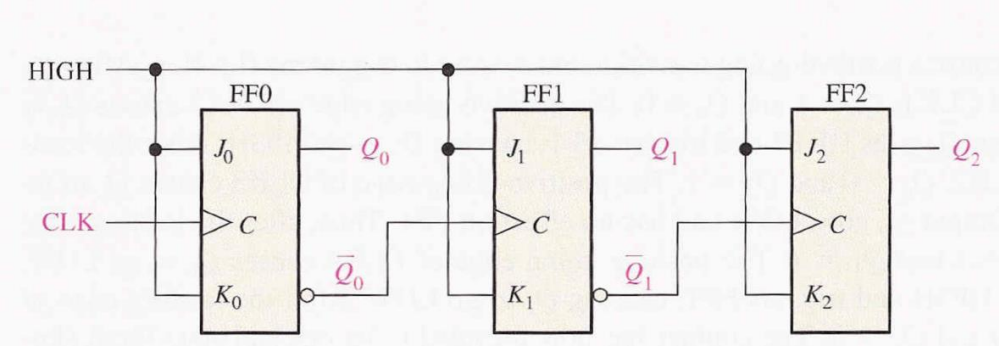
N bit binary counter

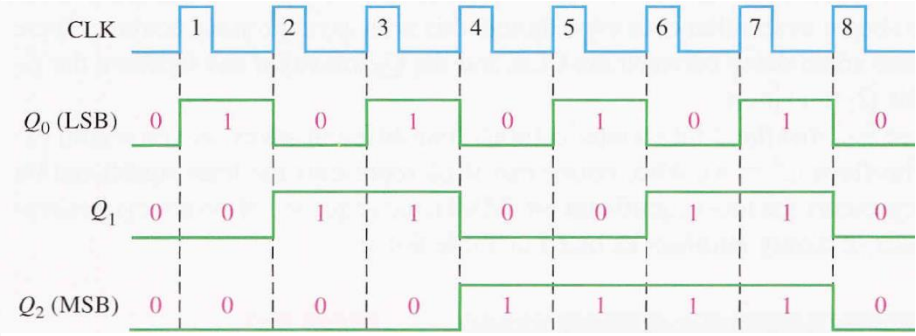
MOD-M counter

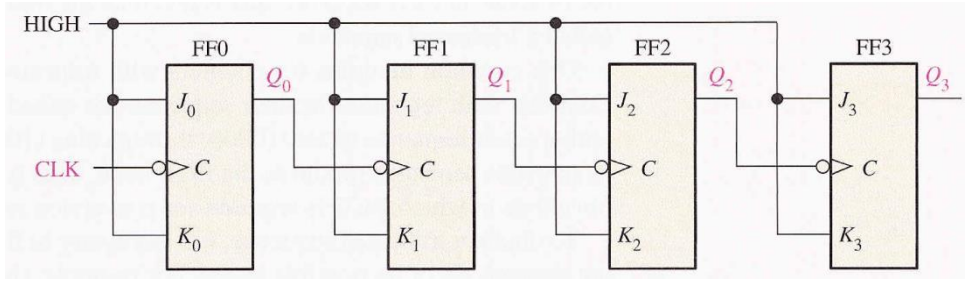
Up/Down *counters*

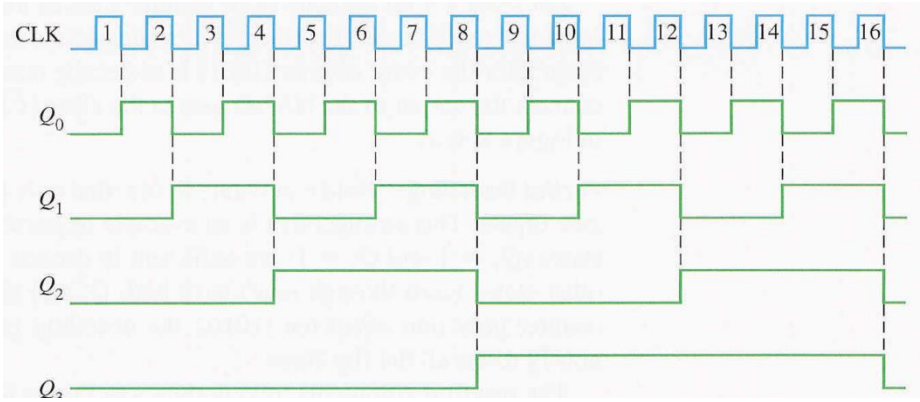
N-bit binary asynchronous counter

Up counter using Positive/Negative edge Triggered FFs









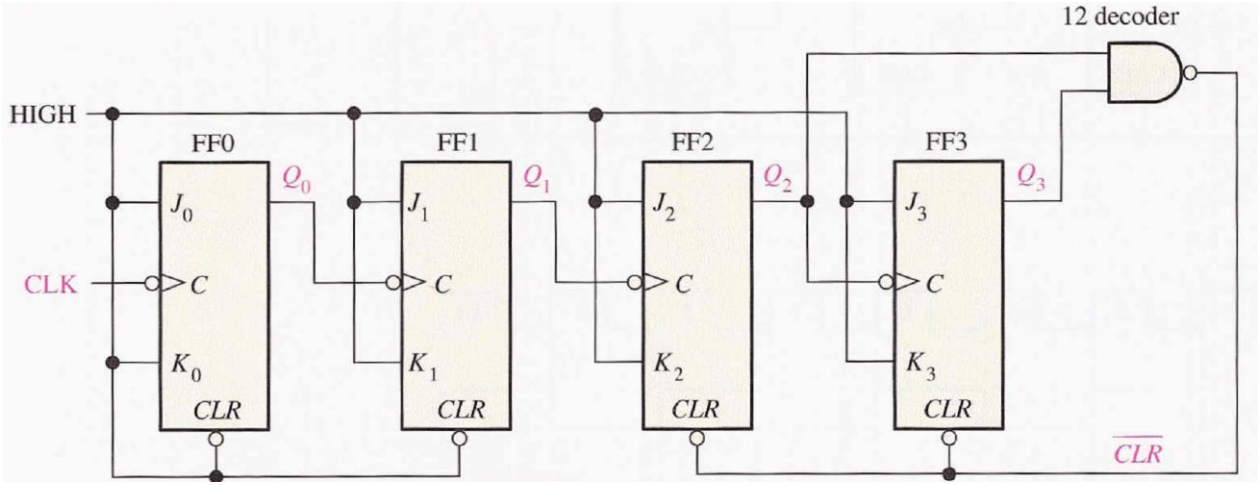
Asynchronous counters are sometimes called **ripple** counters

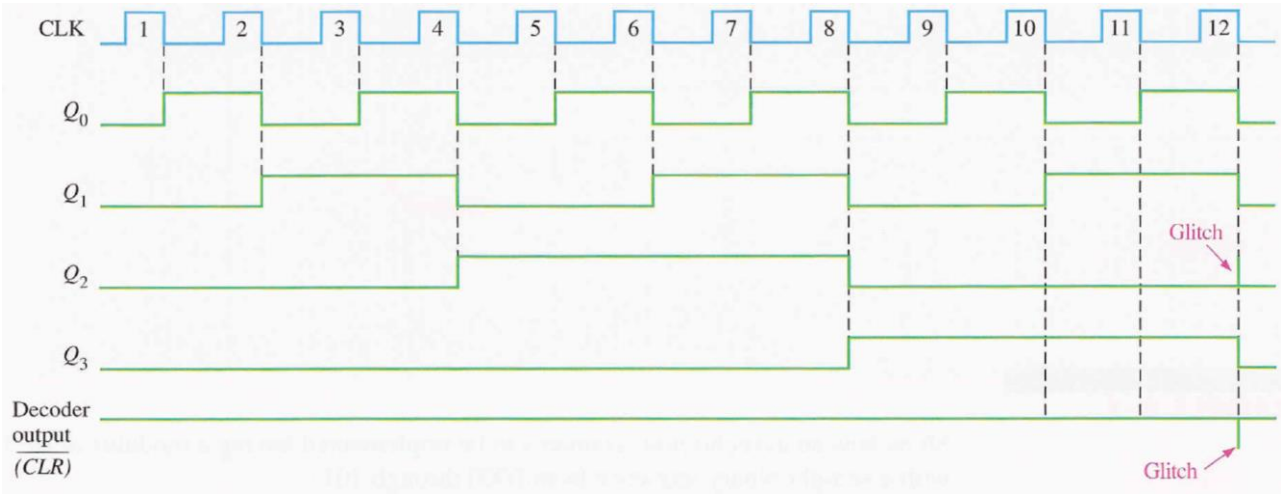
MOD-M Asynchronous Counter

Design method:

Count 0 → 1→2 →…→M-1

Use M[th] stage to reset the counter.

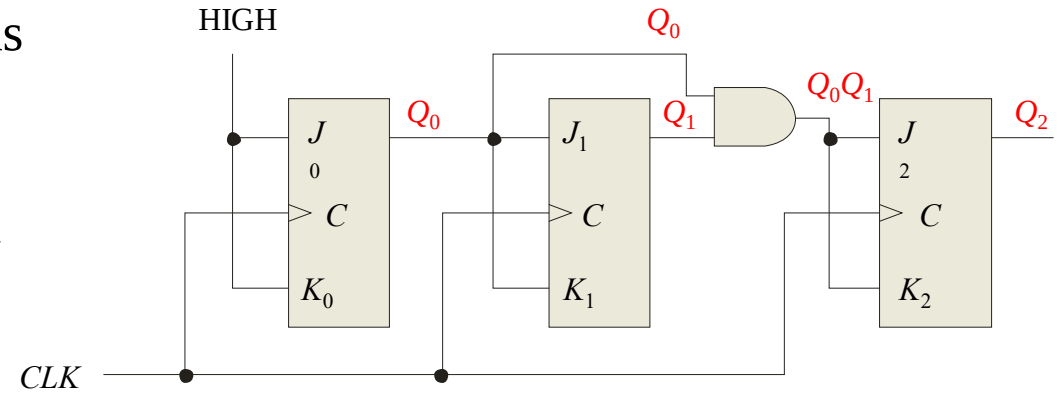




N-bit binary synchronous Counter

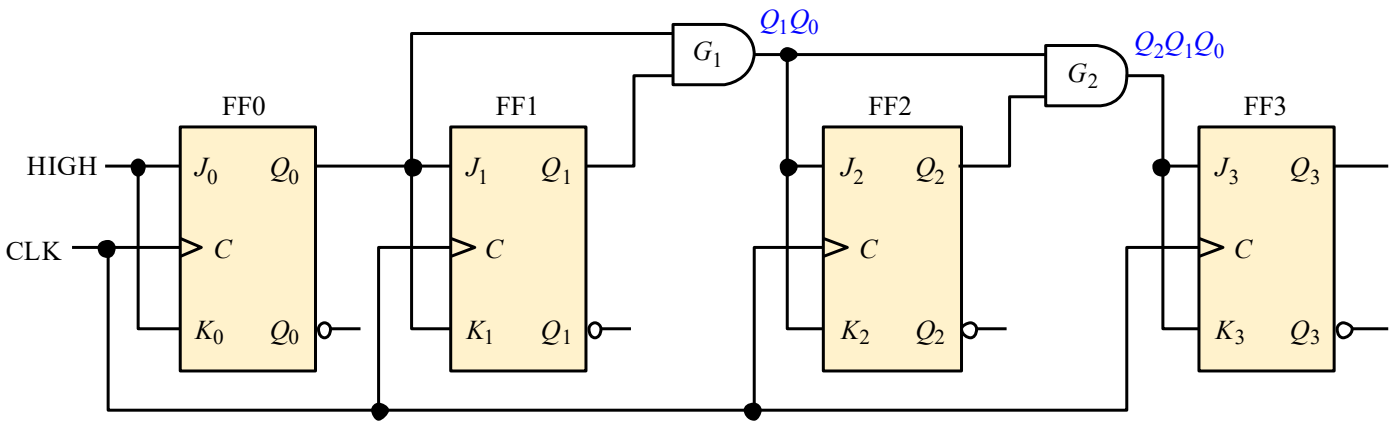
In a **synchronous counter** all flip-flops are clocked together with a common clock pulse. **Synchronous counters overcome the disadvantage of accumulated propagation delays, but generally they require more circuitry to control states changes**.

This 3-bit binary synchronous counter has the same count sequence as the 3-bit asynchronous counter shown previously.

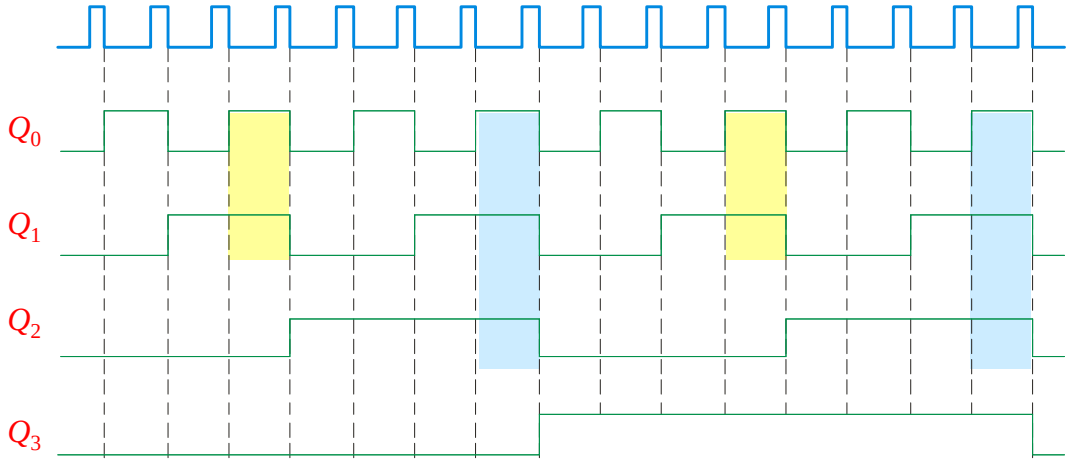


The next slide shows how to analyze this counter by writing the logic equations for each input. Notice the inputs to each flip-flop…

A 4-bit Synchronous Binary Counter



The 4-bit binary counter has one more AND gate than the 3-bit counter just described. The shaded areas show where the AND gate outputs are HIGH causing the next FF to toggle.

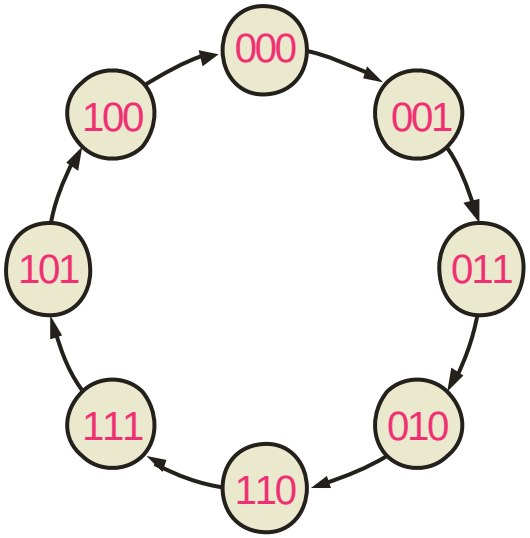


Both binary and Mod-M counters

3-bit Gray Code Counter

Step1: State diagram:

The first step in the design of a counter is to create a state diagram. A state diagram shows the progression of states through which the counter advances when it is clocked.



Step2: Next state table

The second step is to derive a next-state table, which lists each state of the counter (present state) along with the corresponding next state. The next state is the state that the counter goes to from its present state upon application of a clock pulse.

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

Step3: Flip-Flop Transition Table

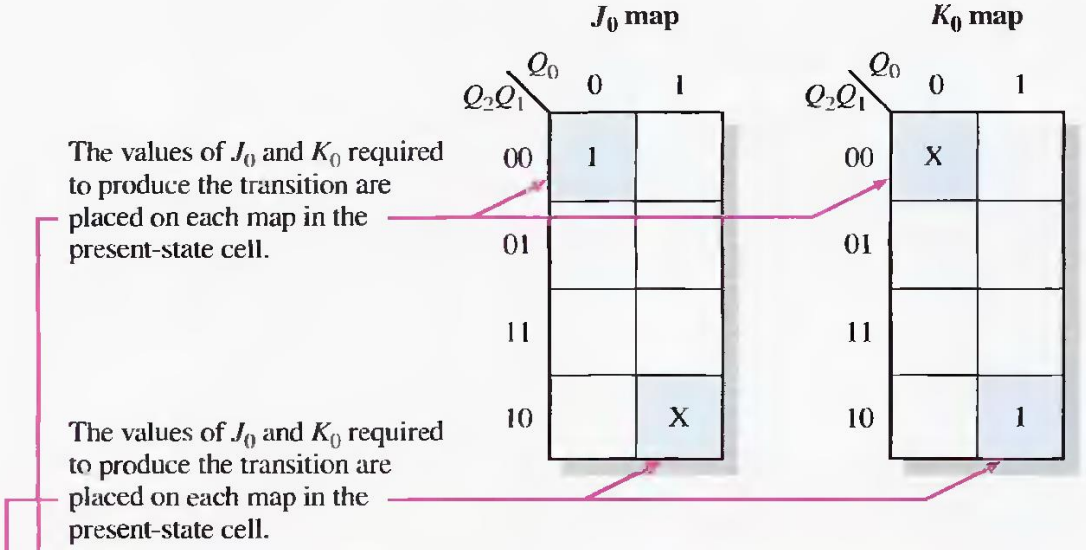
All possible output transitions are listed by showing the Q output of the flip-flop going from present states to next states. Q_N is the present state of the flip-flop (before a clock pulse) and Q_{N+1} is the next state (after a clock pulse).

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

Design using JK-FF

Step4: K-maps

Karnaugh maps can be used to determine the logic required for the 1 and K inputs of each flip-flop in the counter. There is a K-map for the **J** input and a K-map for the **K** input of each flip-flop. In this design procedure, each cell in a K-map represents one of the present states in the counter sequence.



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

Flip-flop transition table

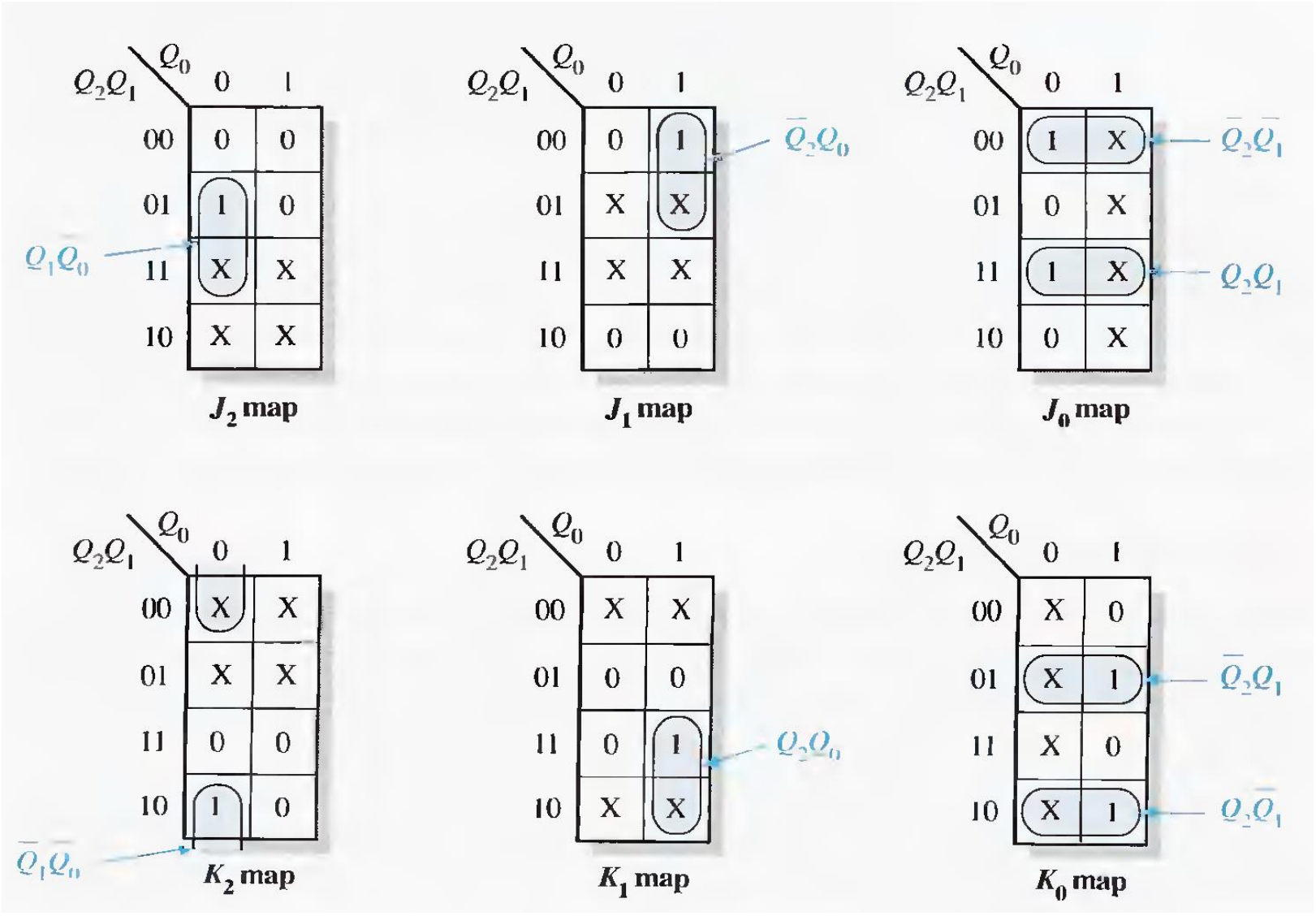
Present State Q_2	Q_1	Q_0	Next State Q_2	Q_1	Q_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0

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.

Complete Karnaugh maps for all the flip-flops in the counter.



Step 5: logic Expressions for Flip-Flop Inputs

$$J_0 = Q_2Q_1 + \overline{Q}_2\overline{Q}_1 = \overline{Q_2 \oplus Q_1}$$

$$K_0 = Q_2\overline{Q}_1 + \overline{Q}_2Q_1 = Q_2 \oplus Q_1$$

$$J_1 = \overline{Q}_2Q_0$$

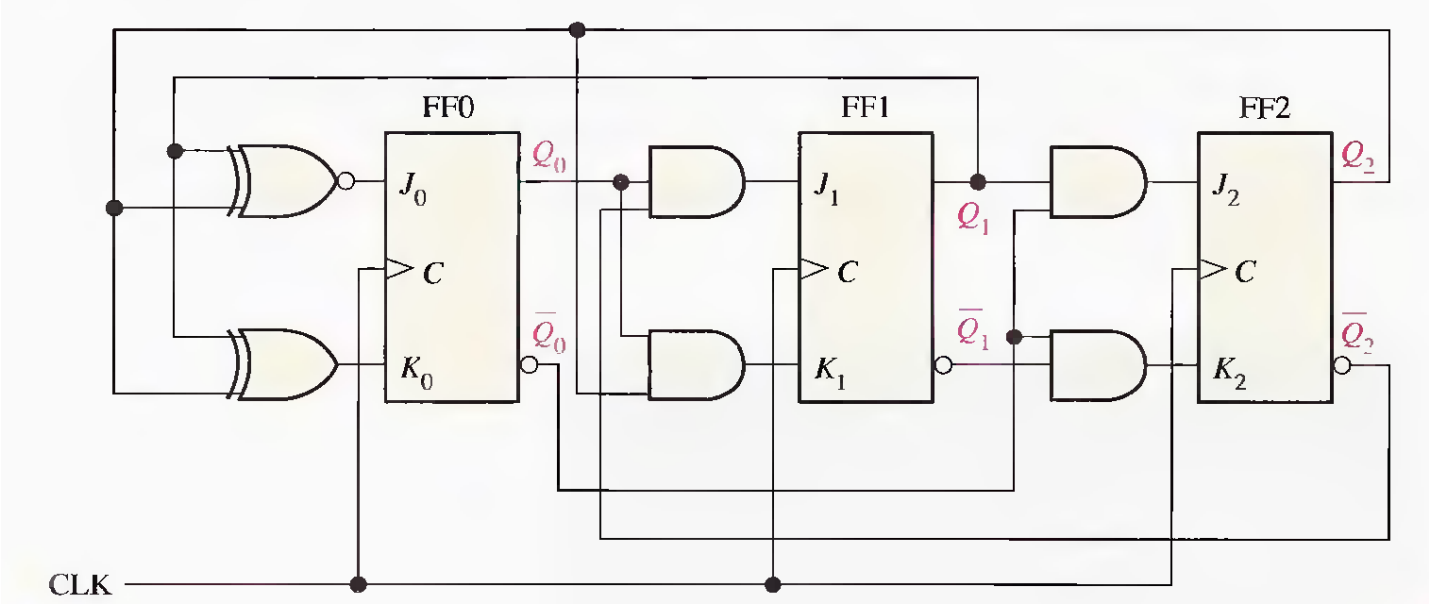
$$K_1 = Q_2Q_0$$

$$J_2 = Q_1\overline{Q}_0$$

$$K_2 = \overline{Q}_1\overline{Q}_0$$

Step 6: Counter Implementation

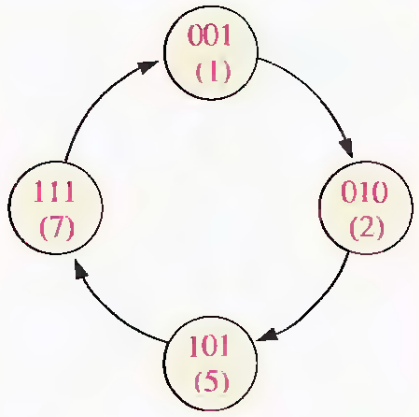
The final step is to implement the combinational logic from the expressions for the J and K inputs and connect the flip-flops to form the complete counter.



EXAMPLE 8–5

Design a counter with the irregular binary count sequence shown in the state diagram of Figure 8–32. Use J-K flip-flops.

▶ **FIGURE 8–32**



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.

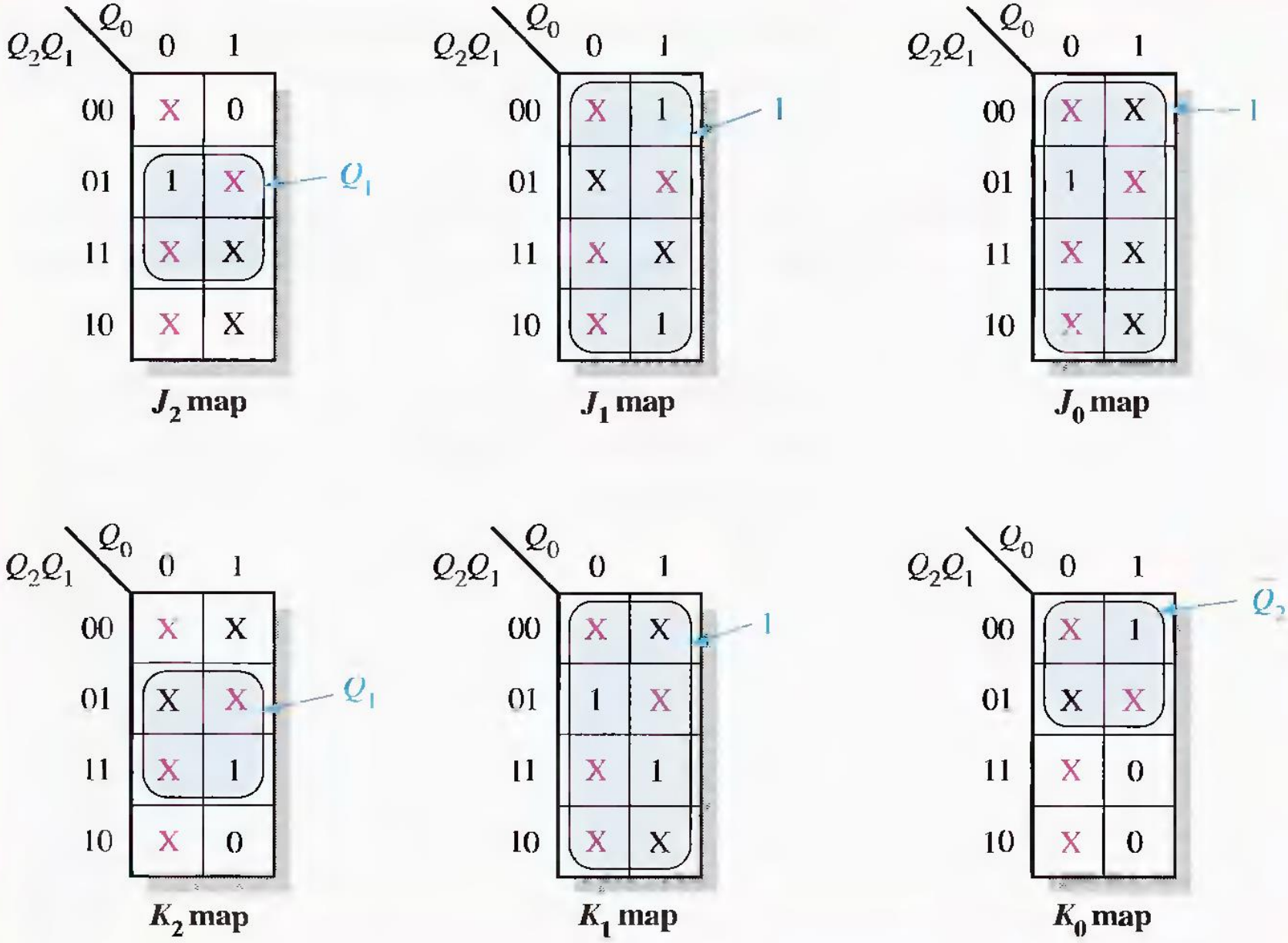
Step 2: The next-state table is developed from the state diagram and is given in Table 8–9.

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 J-K flip-flop is repeated in Table 8–10.

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

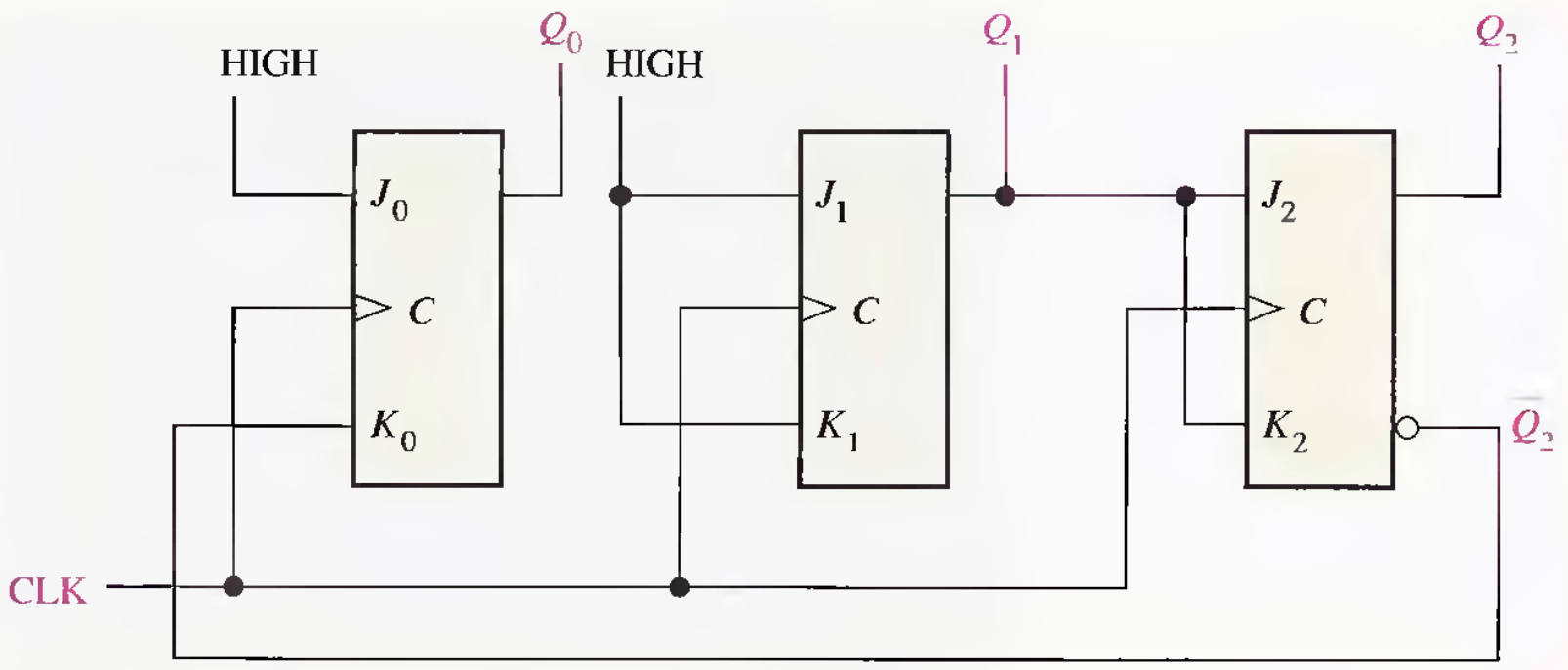
Step 4: The J and K inputs are plotted on the present-state Karnaugh maps in Figure 8–33. Also "don't cares" can be placed in the cells corresponding to the invalid states of 000, 011, 100, and 110, as indicated by the red Xs.


Q_2Q_1 \ Q_0
0 1
00 X 0
01 1 X
11 X X
10 X X
Q_1
J_2 map
Q_2Q_1 \ Q_0
0 1
00 X 1
01 X X
11 X X
10 X 1
1
J_1 map
Q_2Q_1 \ Q_0
0 1
00 X X
01 1 X
11 X X
10 X X
1
J_0 map
Q_2Q_1 \ Q_0
0 1
00 X X
01 X X
11 X 1
10 X 0
Q_1
K_2 map
Q_2Q_1 \ Q_0
0 1
00 X X
01 1 X
11 X 1
10 X X
1
K_1 map
Q_2Q_1 \ Q_0
0 1
00 X 1
01 X X
11 X 0
10 X 0
Q_2
K_0 map

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 8–33. Notice that when *all* cells in a map are grouped, the expression is simply equal to 1. The expression for each J and K input taken from the maps is as follows:

$$J_0 = 1, K_0 = \overline{Q}_2$$
$$J_1 = K_1 = 1$$
$$J_2 = K_2 = Q_1$$

Step 6: The implementation of the counter is shown in Figure 8–34.



Important Tables

Q Q^+	D
0 0	0
0 1	1
1 0	0
1 1	1

Q Q^+	T
0 0	0
0 1	1
1 0	1
1 1	0

Q Q^+	S R
0 0	0 X
0 1	1 0
1 0	0 1
1 1	X 0

Q Q^+	J K
0 0	0 X
0 1	1 X
1 0	X 1
1 1	X 0