



Vietnam National University HCMC
International University

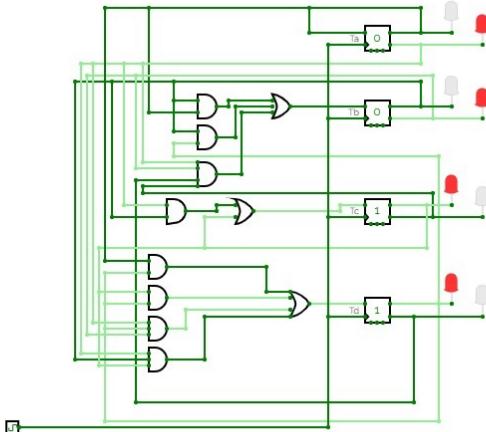


School of
Electrical Engineering

EE053IU

Digital Logic Design

Lecture 9: Counters



INSTRUCTOR: Dr. Vuong Quoc Bao

1. Finite State Machines

- A state machine is a sequential circuit having a limited (finite) number of states occurring in a prescribed order.
- A counter is an example of a state machine; the number of states is called the modulus. Two basic types of state machines are the Moore and the Mealy.
- The Moore state machine is one where the outputs depend only on the internal present state.
- The Mealy state machine is one where the outputs depend on both the internal present state and on the inputs.
- Both types have a timing input (clock) that is not considered a controlling input. A design approach to counters is presented in this section.

General Models of Finite State Machines

- A Moore state machine consists of combinational logic that determines the sequence and memory (flip-flops).
- In the Moore machine, the combinational logic is a gate array with outputs that determine the next state of the flip-flops in the memory.
- For the Mealy machine, the present state affects the outputs, just as in the Moore machine; but in addition, the inputs also affect the outputs. The outputs come directly from the combinational logic and not the memory.

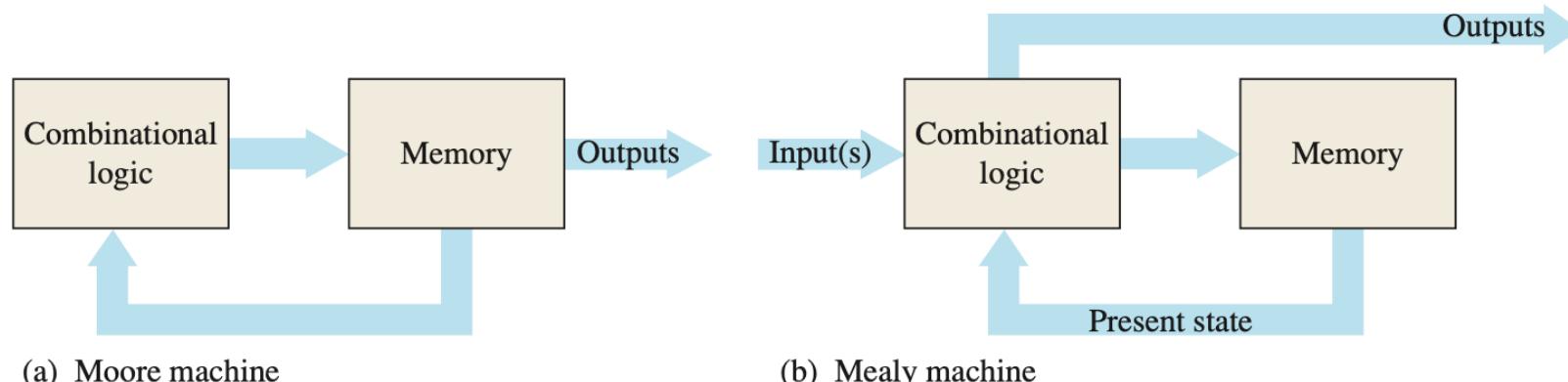
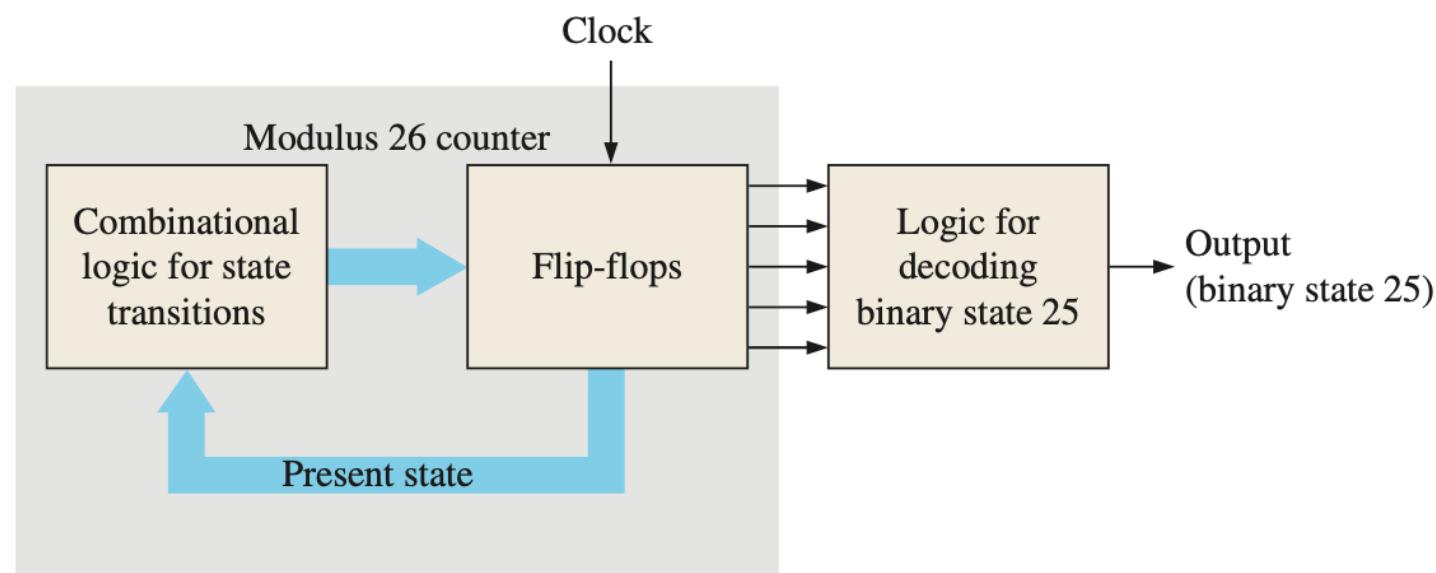
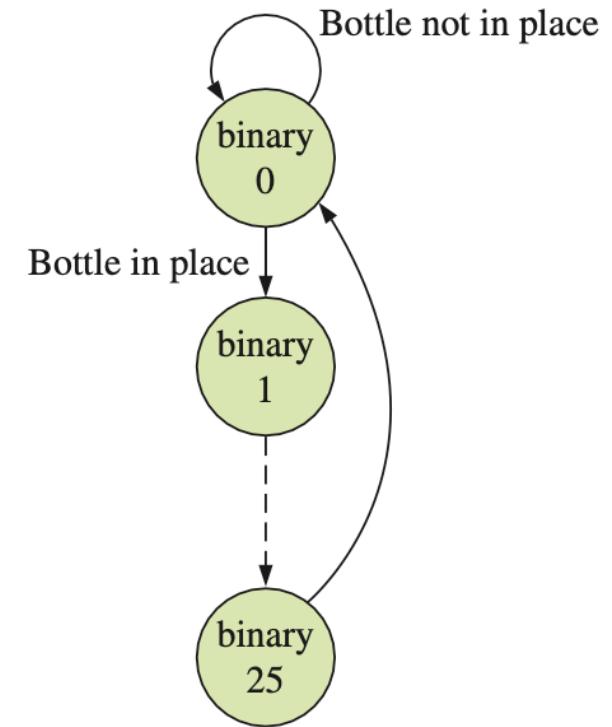


FIGURE 9-1 Two types of sequential logic.

Example of a Moore Machine



(a) Moore machine



(b) State diagram

FIGURE 9–2 A fixed-modulus binary counter as an example of a Moore state machine. The dashed line in the state diagram means the states between binary 1 and 25 are not shown for simplicity.

Example of a Mealy Machine

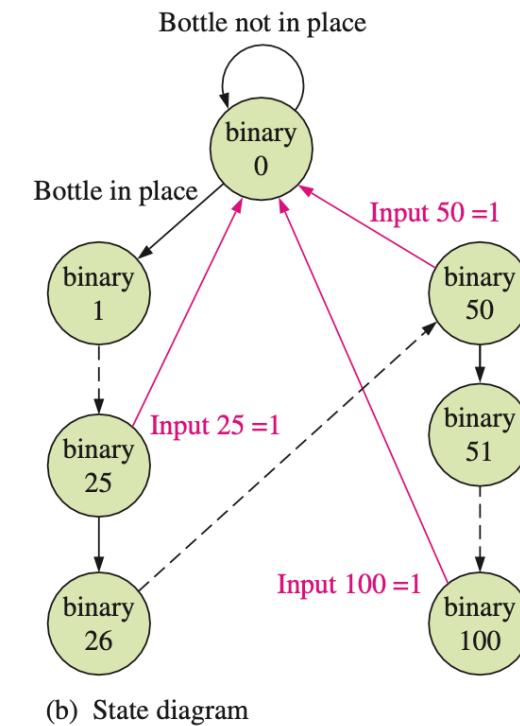
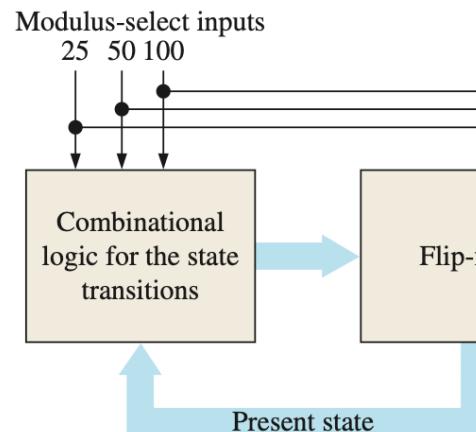


FIGURE 9–3 A variable-modulus binary counter as an example of a Mealy state machine. The red arrows in the state diagram represent the recycle paths that depend on the input number. The black dashed lines mean the interim states are not shown for simplicity.

2. Asynchronous Counters

- The term **asynchronous** refers to events that do not have a fixed time relationship with each other and, generally, do not occur at the same time.
- An **asynchronous counter** is one in which the flip-flops (FF) within the counter do not change states at exactly the same time because they do not have a common clock pulse.

A 2-Bit Asynchronous Binary Counter

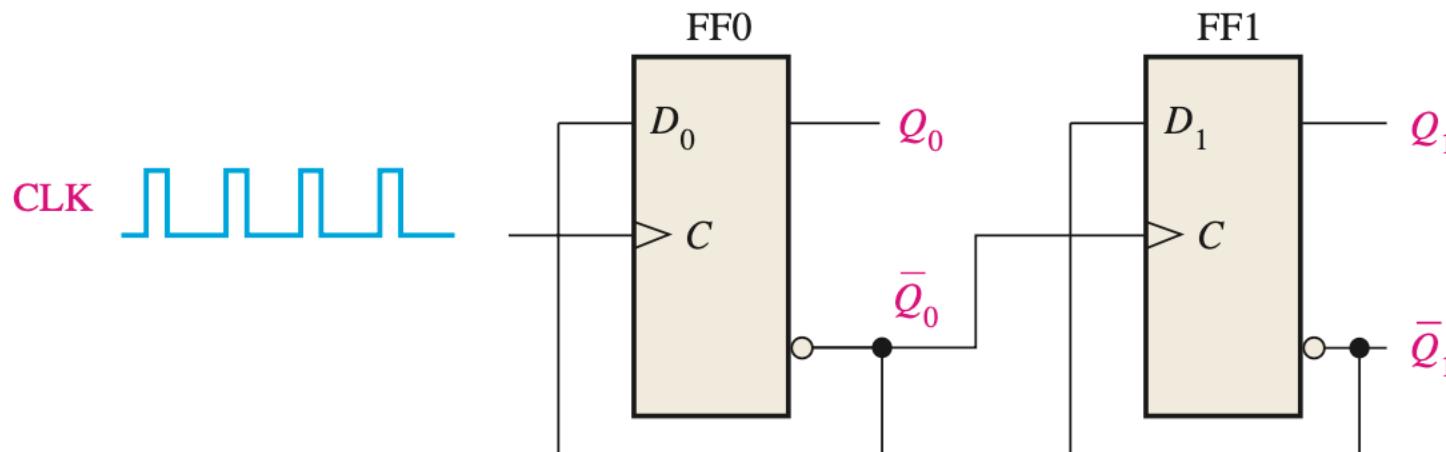


FIGURE 9–4 A 2-bit asynchronous binary counter.

The clock input of an asynchronous counter is always connected only to the LSB flip-flop.

The Timing Diagram

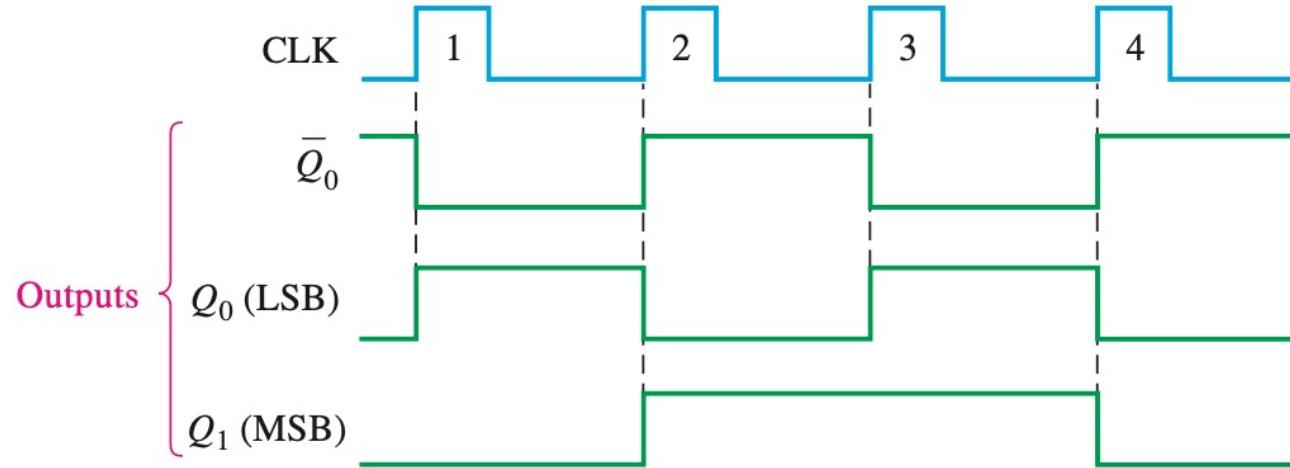


FIGURE 9–5 Timing diagram for the counter of Figure 9–4. As in previous chapters, output waveforms are shown in green.

TABLE 9–1

Binary state sequence for the counter in Figure 9–4.

Clock Pulse	Q_1	Q_0
Initially	0	0
1	0	1
2	1	0
3	1	1
4 (recycles)	0	0

In digital logic, Q_0 is always the LSB unless otherwise specified.

A 3-bit Asynchronous Binary Counter

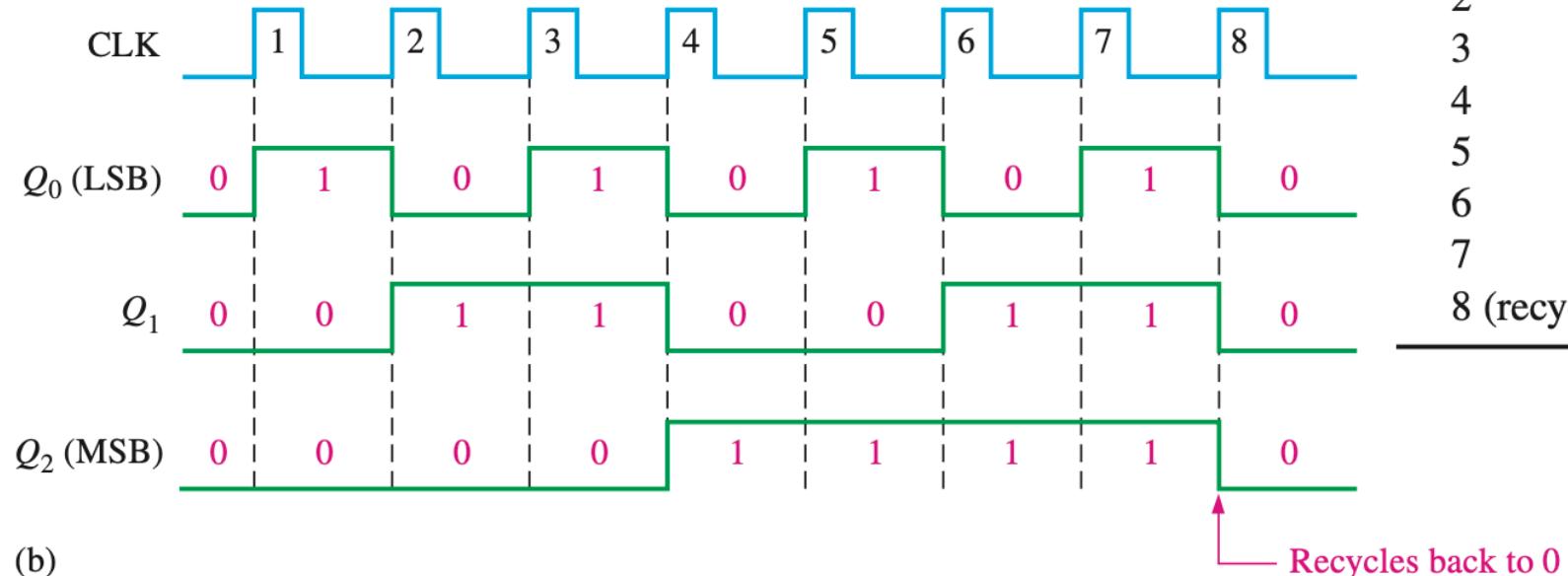
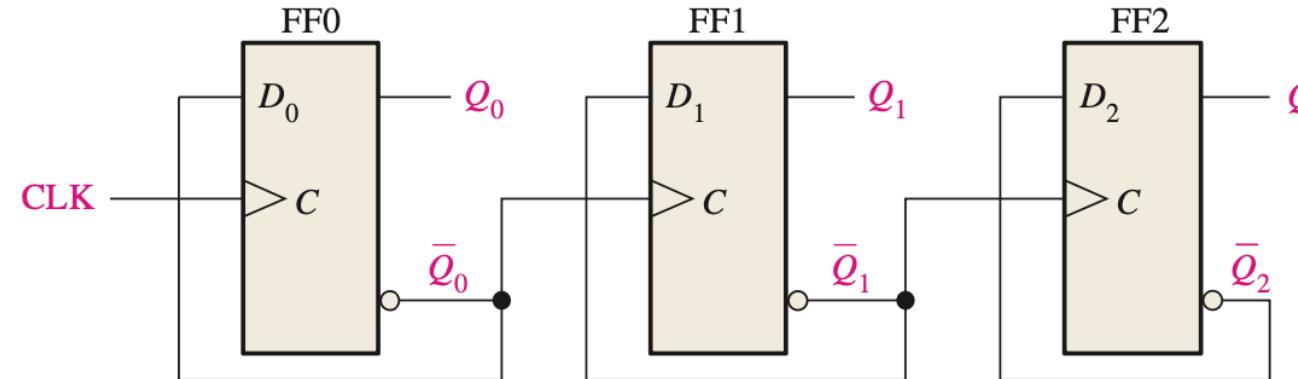


TABLE 9–2

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

FIGURE 9–6 Three-bit asynchronous binary counter and its timing diagram for one cycle.

Propagation Delay

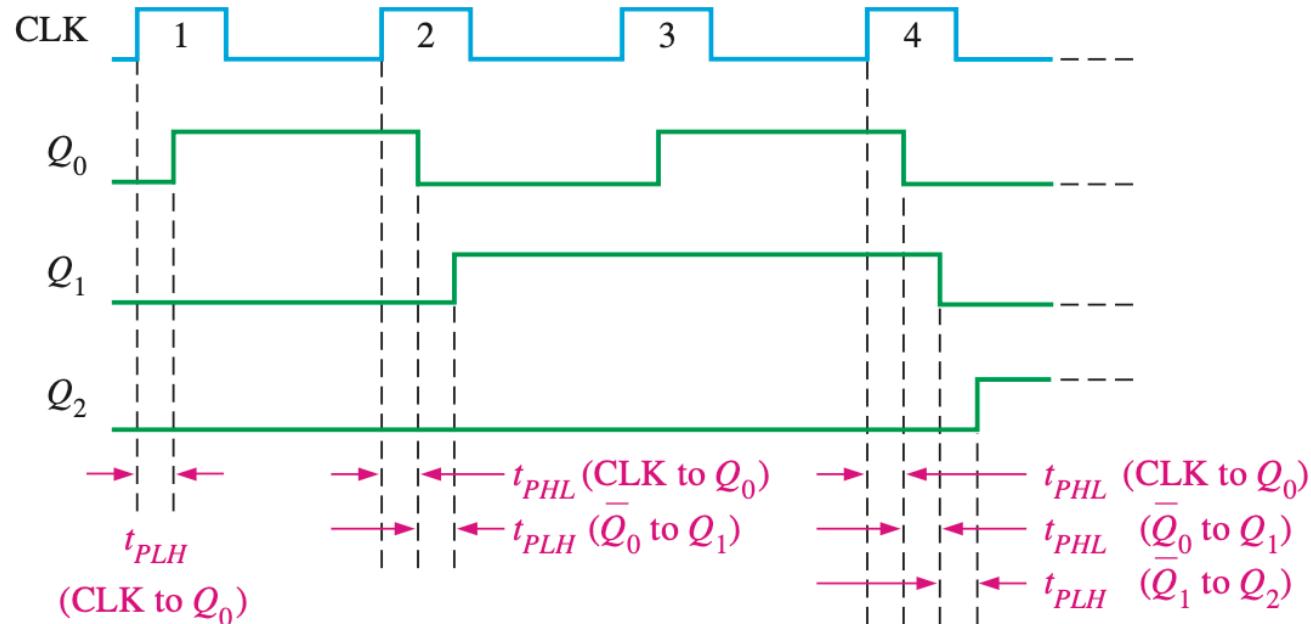
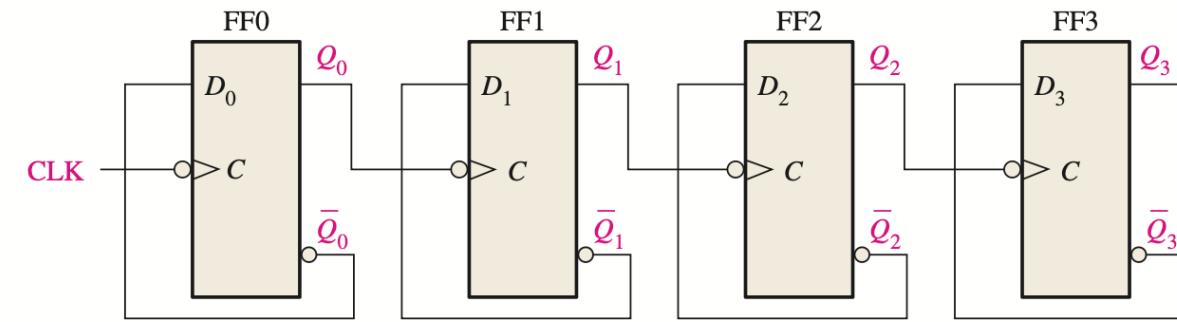


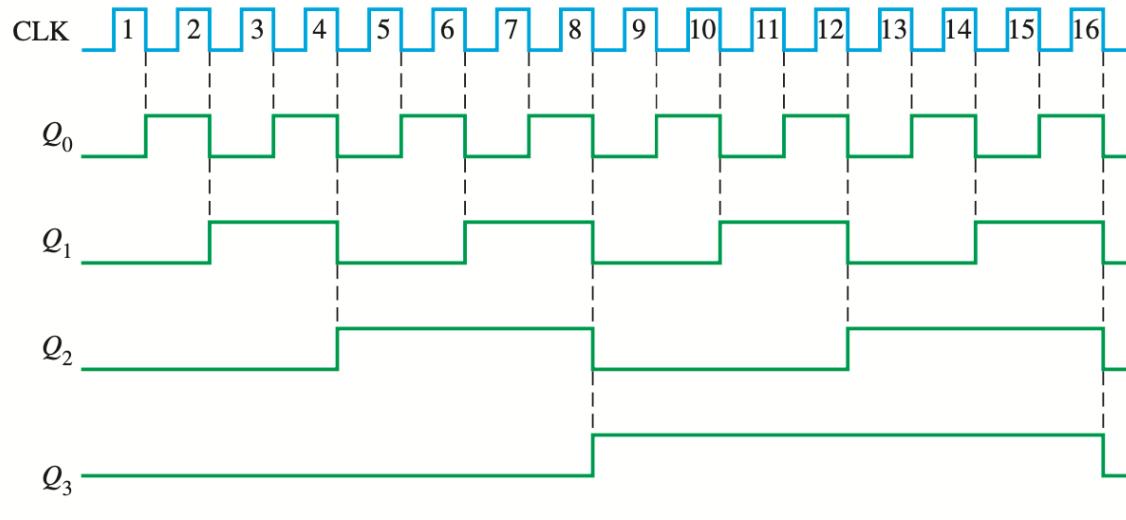
FIGURE 9–7 Propagation delays in a 3-bit asynchronous (ripple-clocked) binary counter.

EXAMPLE 9–1

A 4-bit asynchronous binary counter is shown in Figure 9–8(a). Each D flip-flop is negative edge-triggered and has a propagation delay for 10 nanoseconds (ns). Develop a timing diagram showing the Q output of each flip-flop, and determine the total propagation delay time from the triggering edge of a clock pulse until a corresponding change can occur in the state of Q_3 . Also determine the maximum clock frequency at which the counter can be operated.



(a)



(b)

Solution

The timing diagram with delays omitted is as shown in Figure 9–8(b). For the total delay time, the effect of CLK8 or CLK16 must propagate through four flip-flops before Q_3 changes, so

$$t_{p(tot)} = 4 \times 10 \text{ ns} = 40 \text{ ns}$$

The maximum clock frequency is

$$f_{\max} = \frac{1}{t_{p(tot)}} = \frac{1}{40 \text{ ns}} = 25 \text{ MHz}$$

The counter should be operated below this frequency to avoid problems due to the propagation delay.

FIGURE 9–8 Four-bit asynchronous binary counter and its timing diagram.

Asynchronous Decade Counters

- The **modulus** of a counter is the number of unique states through which the counter will sequence. The maximum possible number of states (maximum modulus) of a counter is 2^n , where n is the number of flip-flops in the counter.
- Counters with ten states in their sequence are called decade counters. A decade counter with a count sequence of zero (0000) through nine (1001) is a BCD decade counter because its ten-state sequence produces the BCD code.
- This type of counter is useful in display applications in which BCD is required for conversion to a decimal readout.

Partial Decoding

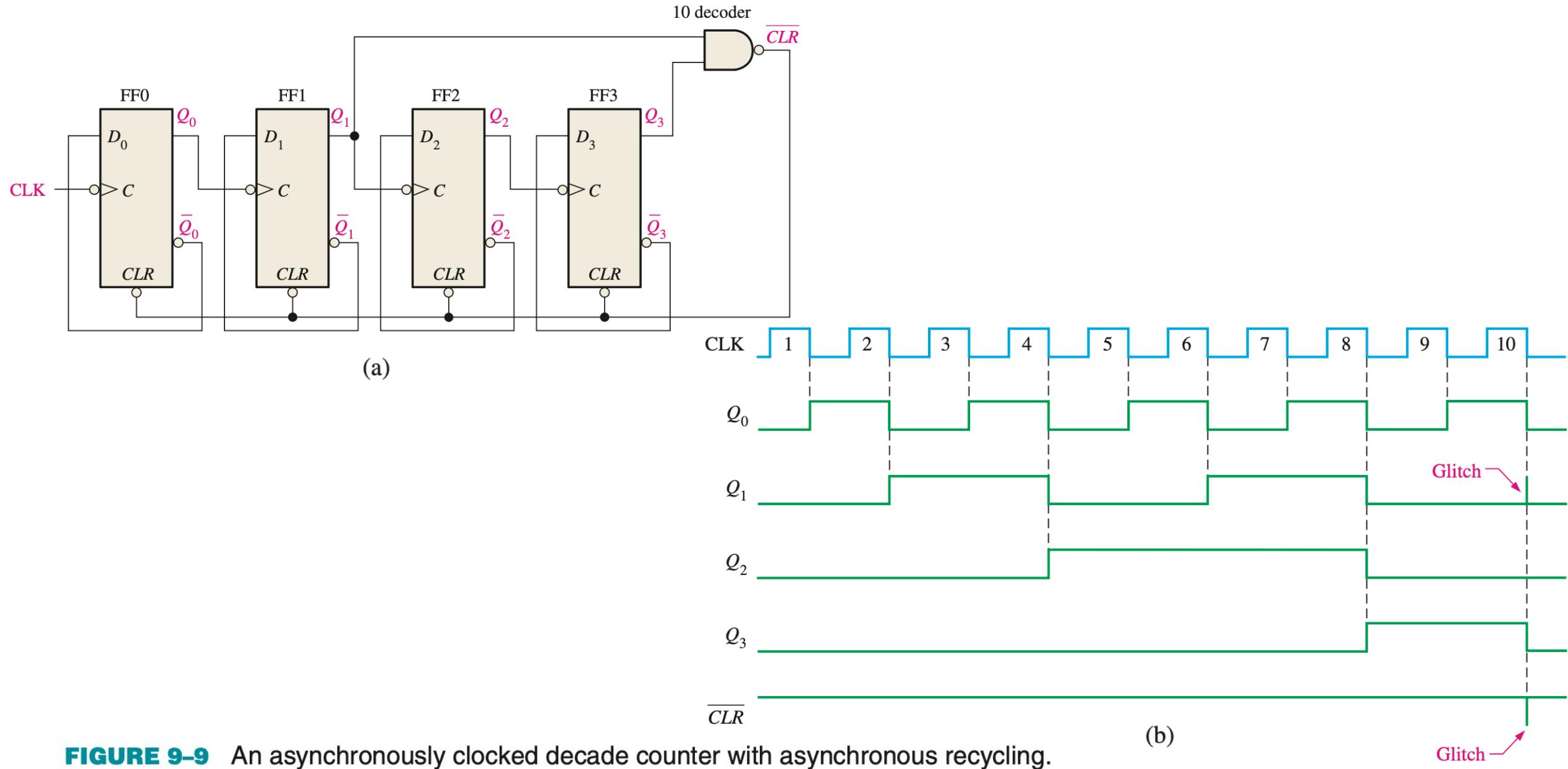


FIGURE 9-9 An asynchronously clocked decade counter with asynchronous recycling.

Show how an asynchronous counter with J-K flip-flops can be implemented having a modulus of twelve with a straight binary sequence from 0000 through 1011.

Solution

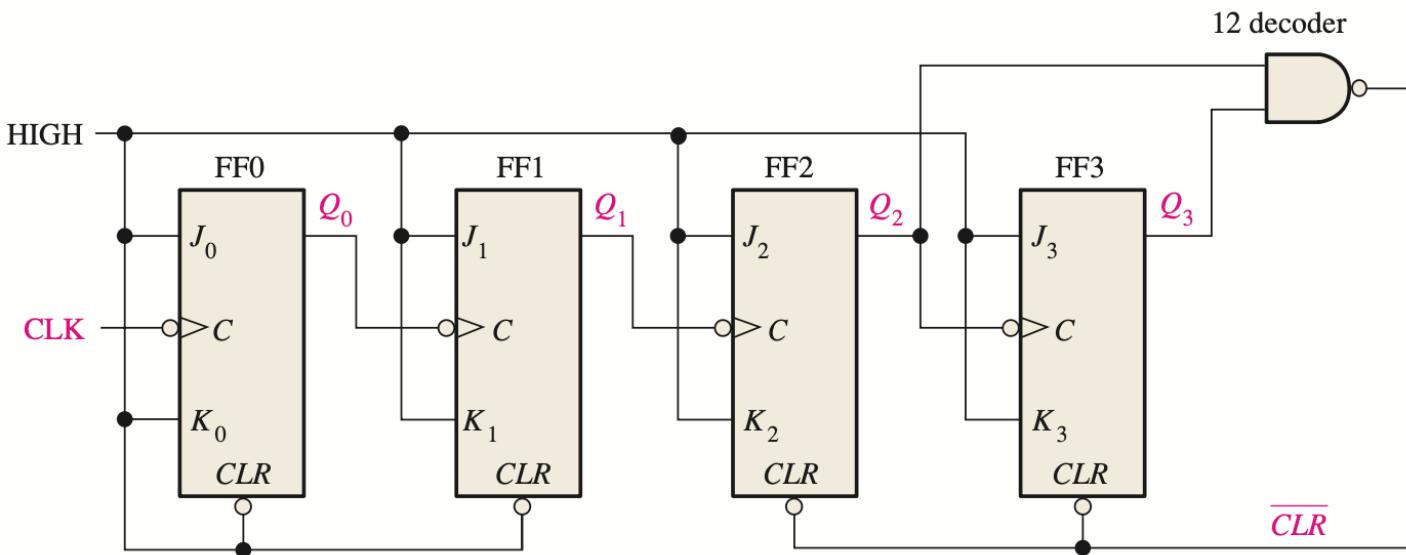
Since three flip-flops can produce a maximum of eight states, four flip-flops are required to produce any modulus greater than eight but less than or equal to sixteen.

When the counter gets to its last state, 1011, it must recycle back to 0000 rather than going to its normal next state of 1100, as illustrated in the following sequence chart:

Q_3	Q_2	Q_1	Q_0
0	0	0	0
.	.	.	.
.	.	.	.
.	.	.	.
1	0	1	1
1	1	0	0

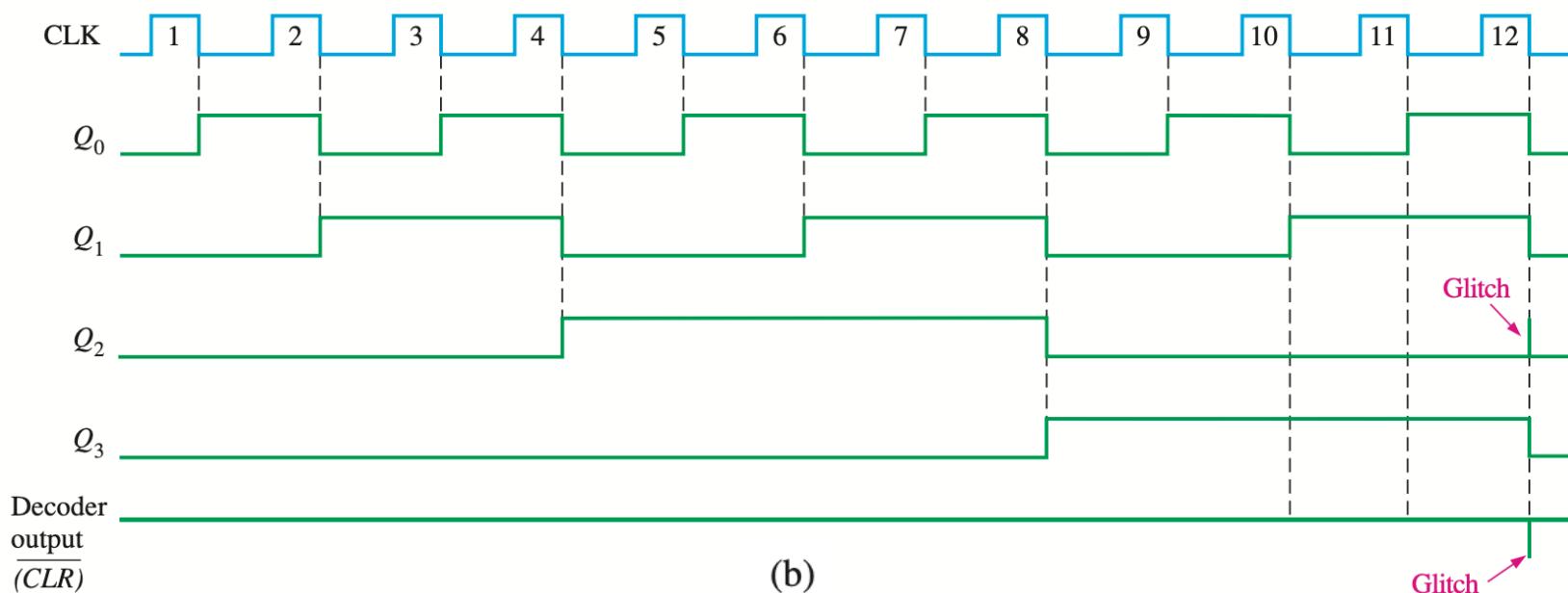
Observe that Q_0 and Q_1 both go to 0 anyway, but Q_2 and Q_3 must be forced to 0 on the twelfth clock pulse. Figure 9–10(a) shows the modulus-12 counter. The NAND gate partially decodes count twelve (1100) and resets flip-flop 2 and flip-flop 3.

Thus, on the twelfth clock pulse, the counter is forced to recycle from count eleven to count zero, as shown in the timing diagram of Figure 9–10(b). (It is in count twelve for only a few nanoseconds before it is reset by the glitch on \overline{CLR} .)



(a)

cho bao nhiêu thì cng thêm 1 s r i ra binary
có bao nhiêu con 1 thì ni by nhiêu chân vô 1 con NAND



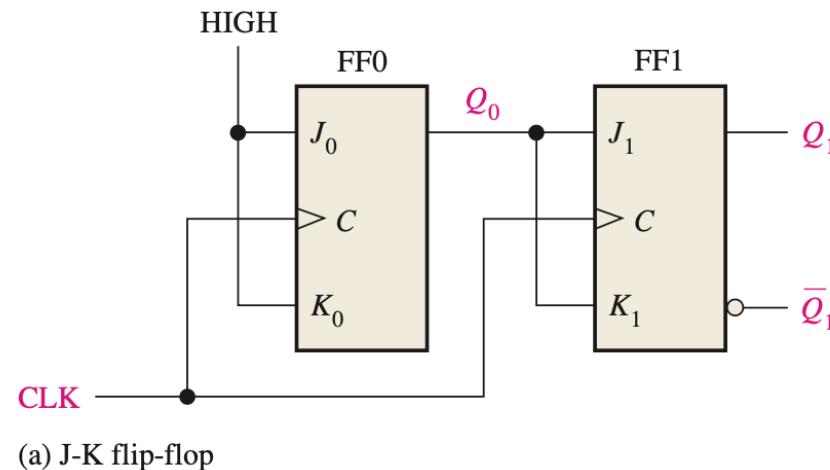
(b)

FIGURE 9–10 Asynchronously clocked modulus-12 counter with asynchronous recycling.

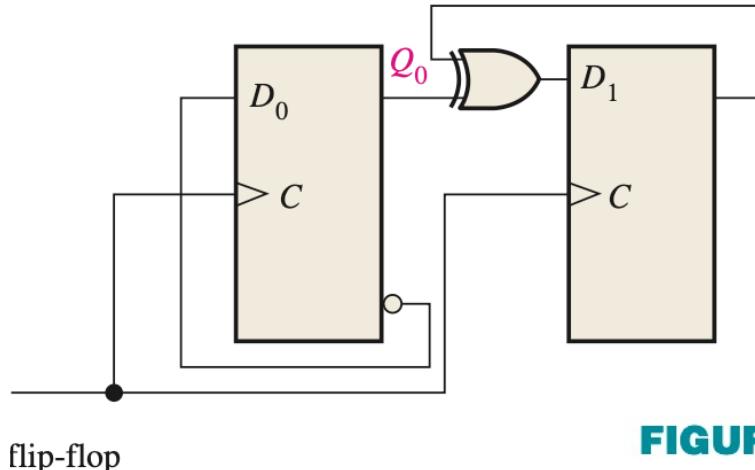
3. Synchronous Counters

- 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.
- The clock input goes to each flip-flop in a synchronous counter.

A 2-Bit Synchronous Binary Counter



(a) J-K flip-flop



flip-flop

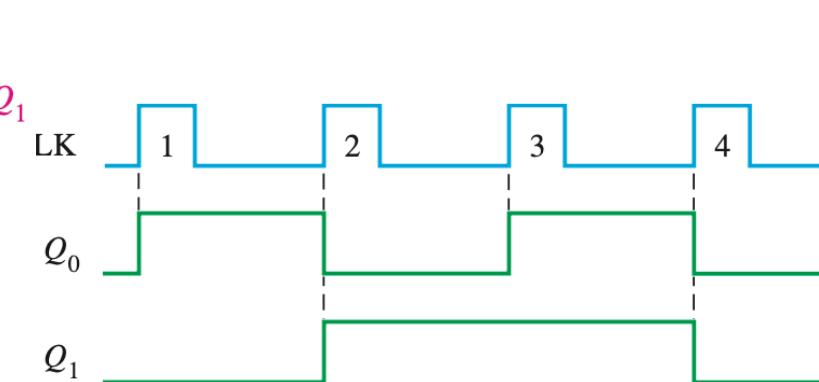
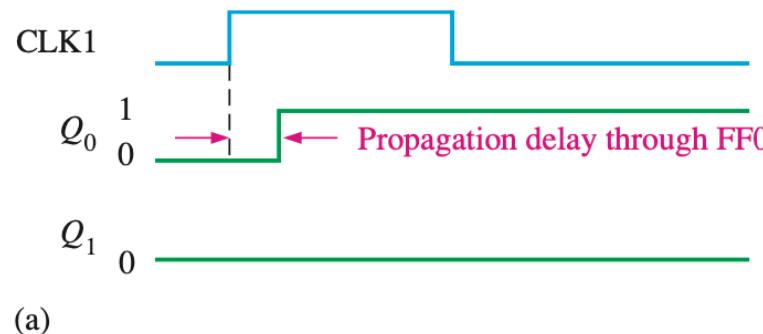
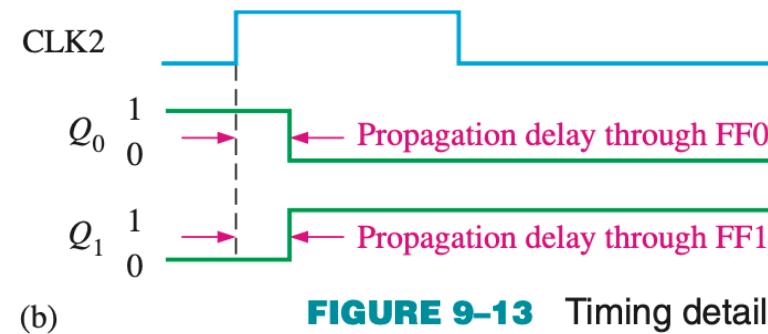


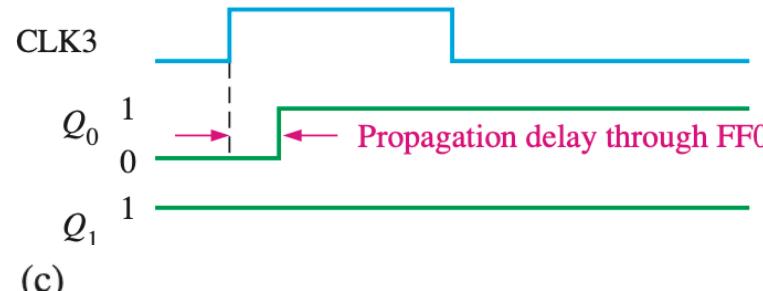
FIGURE 9-12 2-bit synchronous binary counters.



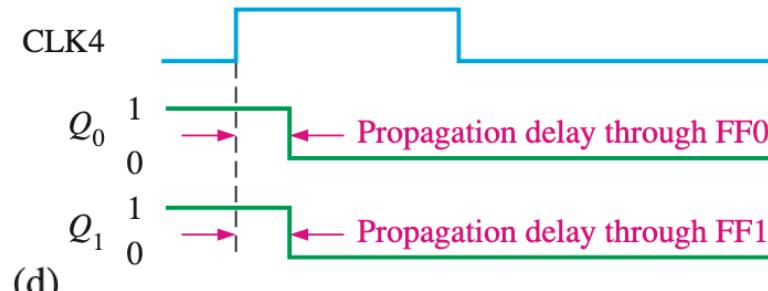
(a)



(b)



(c)



(d)

FIGURE 9-13 Timing details for the 2-bit synchronous counter operation (the propagation delays of both flip-flops are assumed to be equal).

A 3-Bit Synchronous Binary Counter

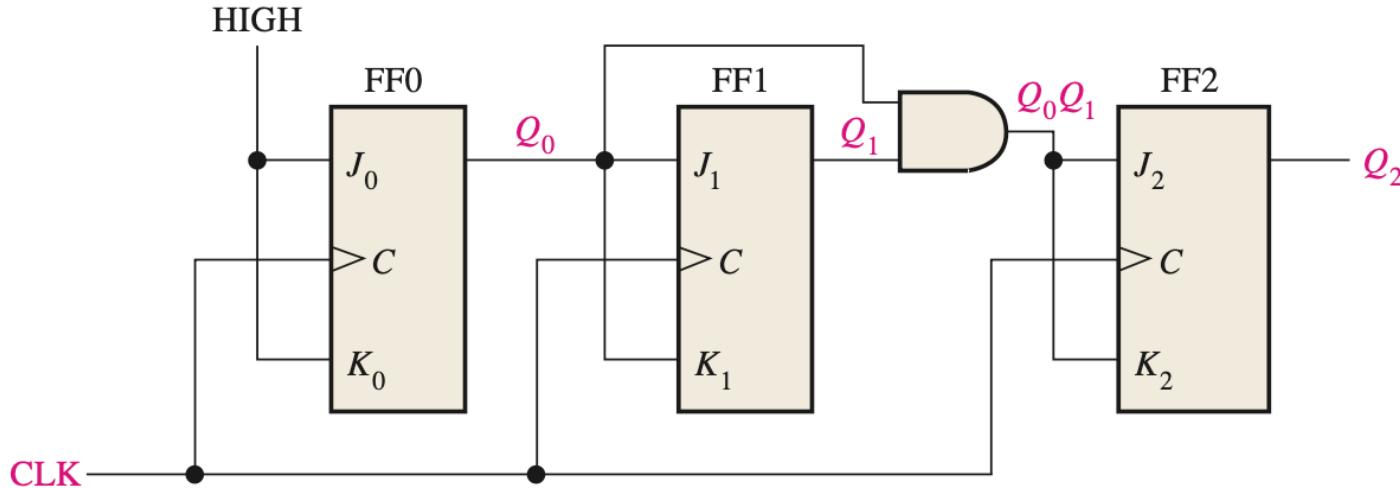


FIGURE 9–15 A 3-bit synchronous binary counter. Open file F09-15 to verify the operation.

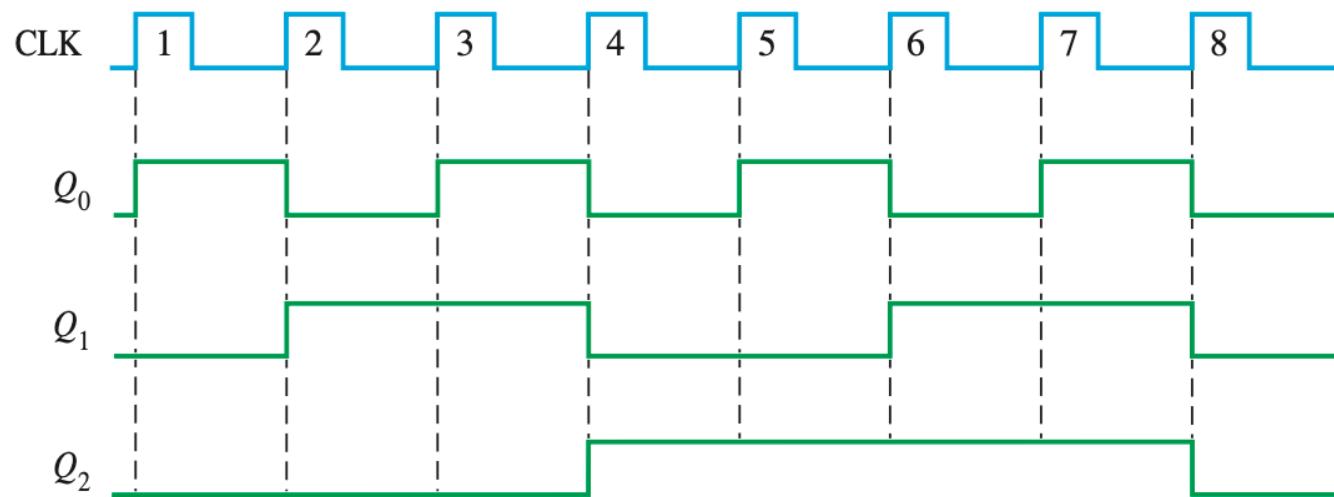


FIGURE 9–16 Timing diagram for the counter of Figure 9–15.

0 0 thi no changes
no delay 1 buoc

A 3-Bit Synchronous Binary Counter

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

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*.

A 4-Bit Synchronous Binary Counter

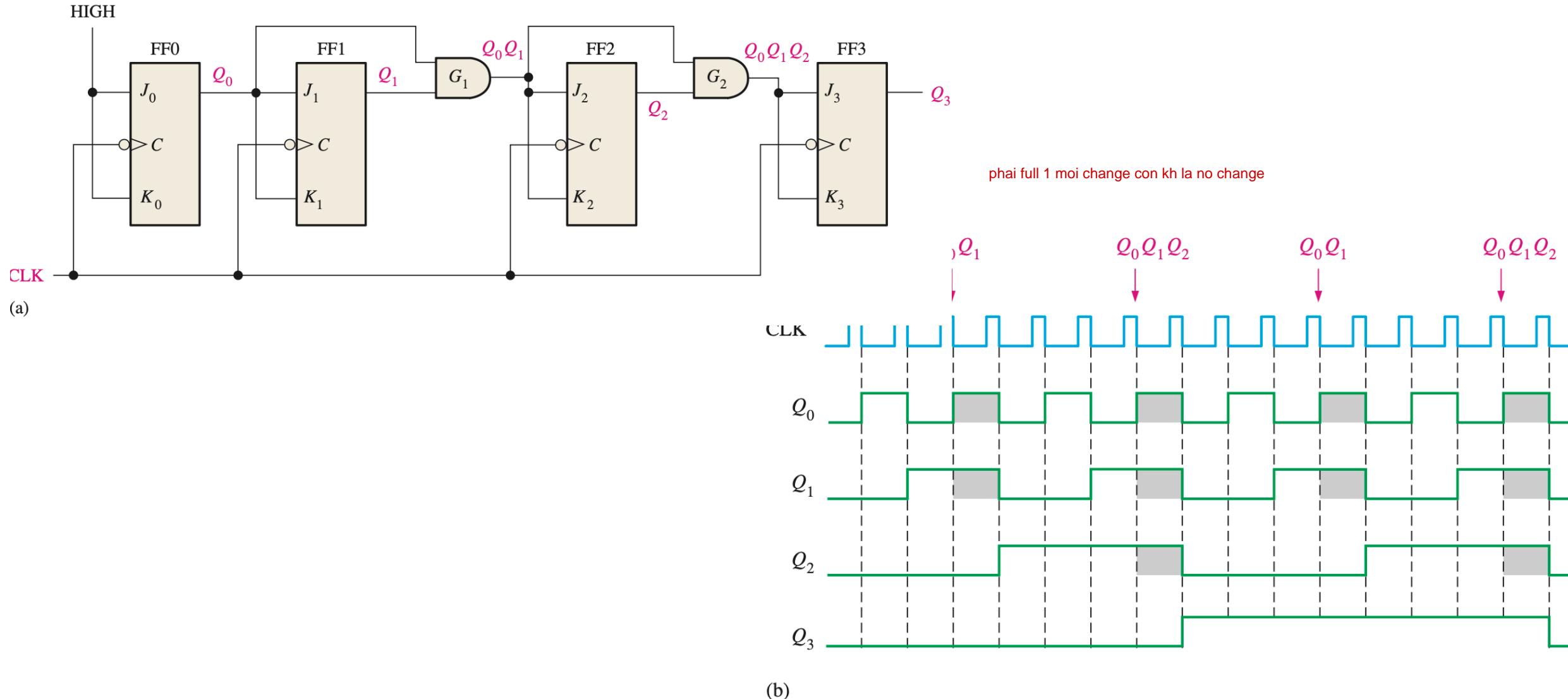


FIGURE 9–17 A 4-bit synchronous binary counter and timing diagram. Times where the AND gate outputs are HIGH are indicated by the shaded areas.

A 4-Bit Synchronous Decade Counter

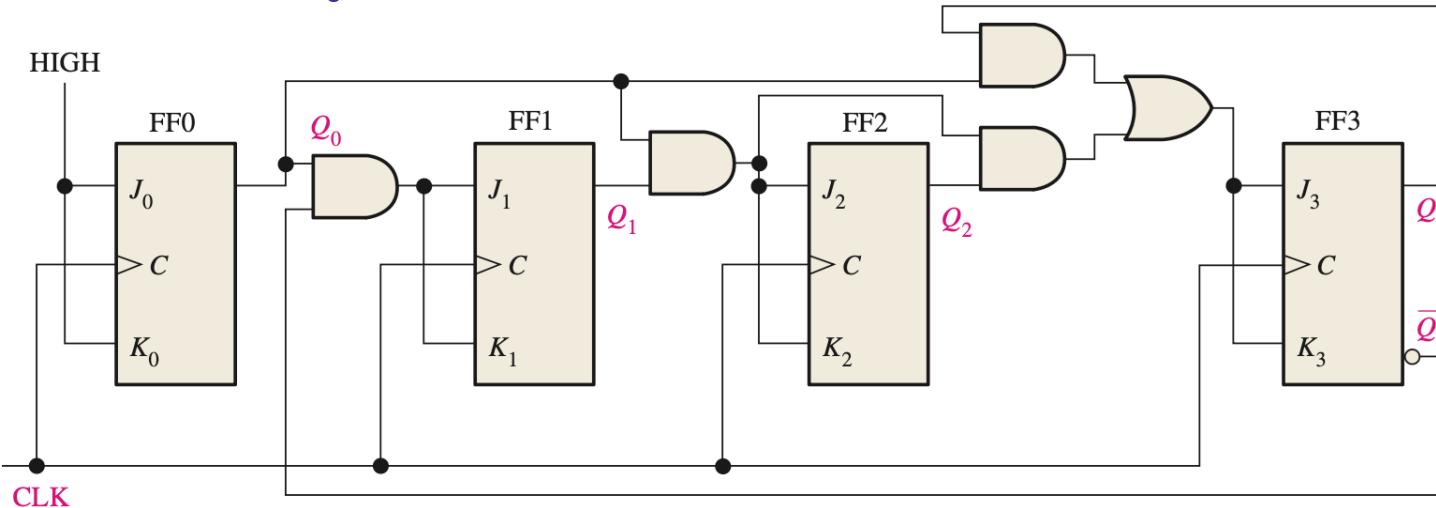


FIGURE 9–18 A synchronous BCD decade counter.

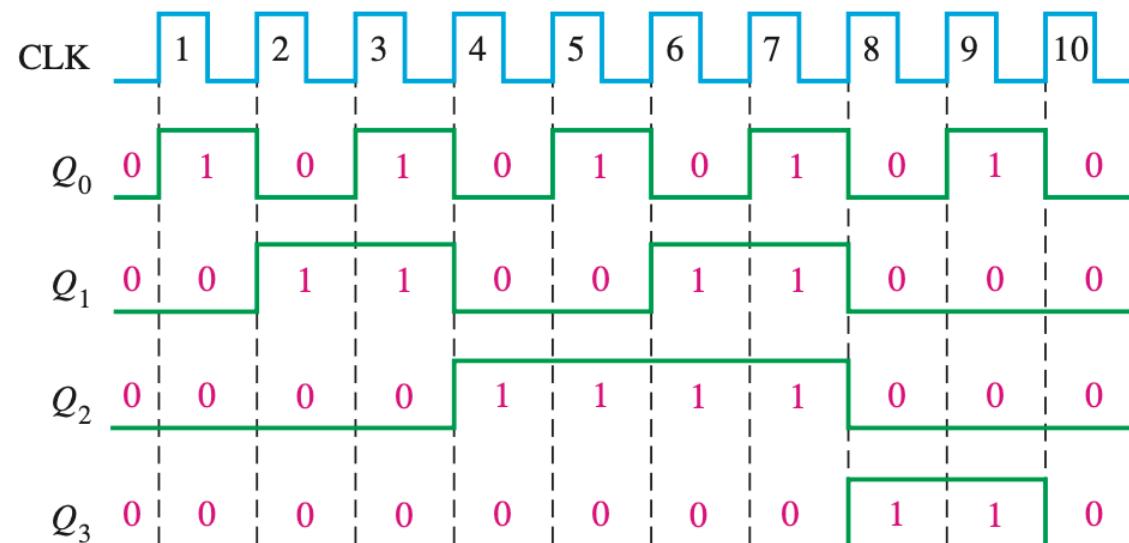


FIGURE 9–19 Timing diagram for the BCD decade counter (Q_0 is the LSB).

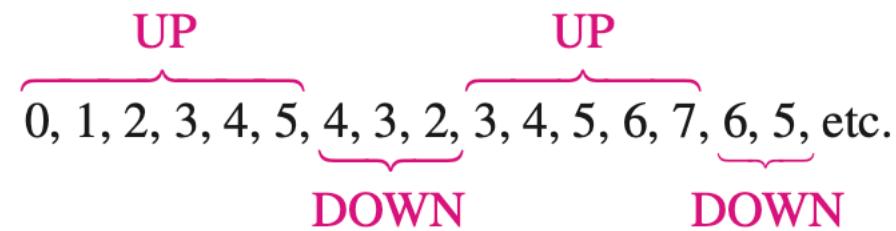
TABLE 9–5

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

4. 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.



4. Up/Down Synchronous Counters

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	↓

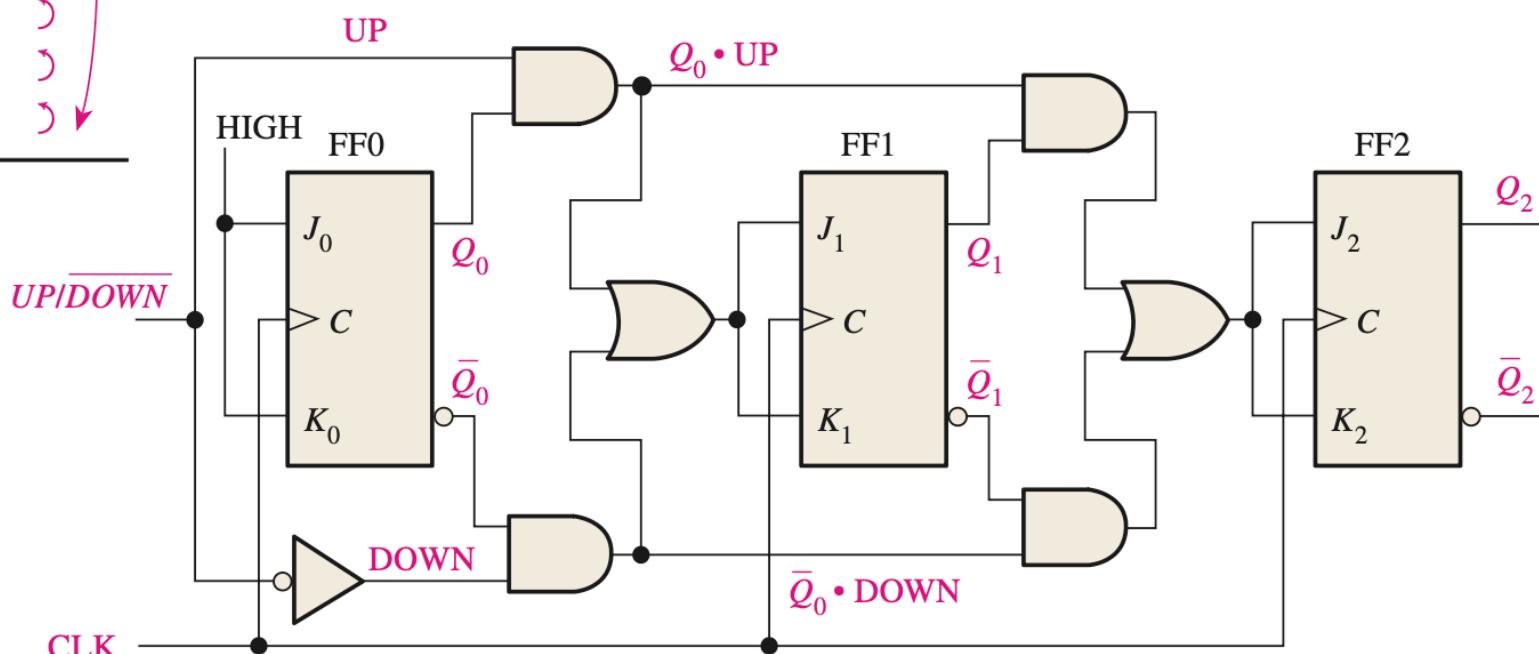


FIGURE 9–22 A basic 3-bit up/down synchronous counter.

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/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.

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.

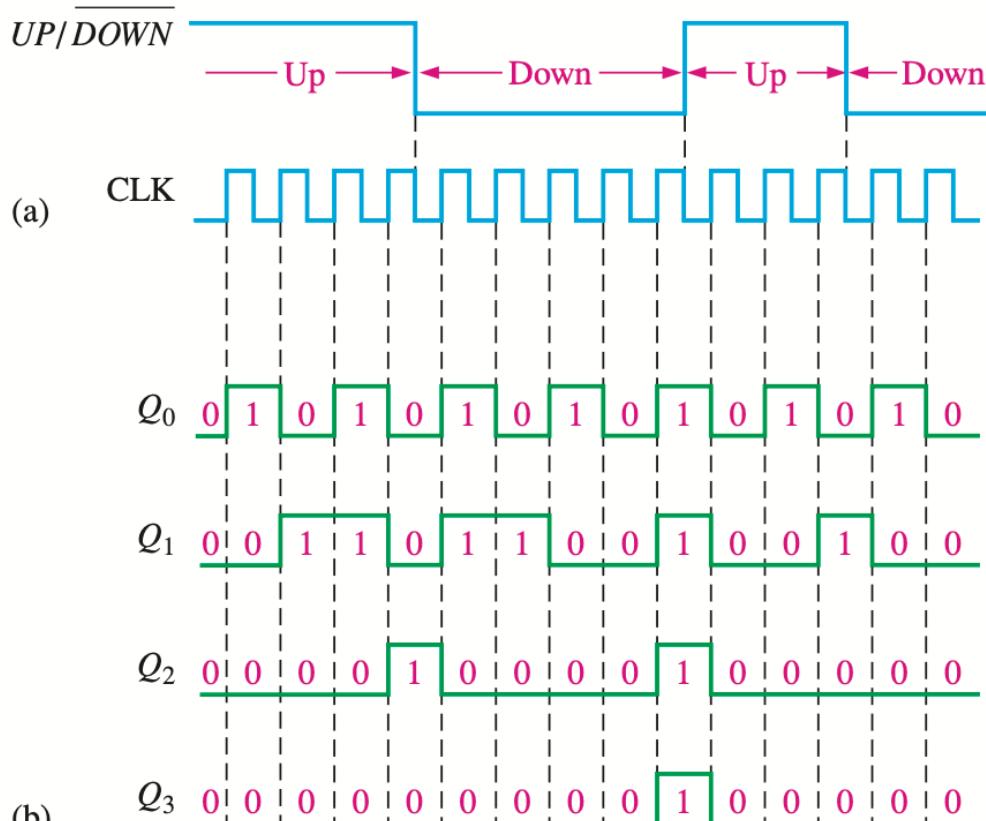


FIGURE 9–23

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	
0	0	1	0	DOWN
0	0	0	1	
0	0	0	0	
1	1	1	1	
0	0	0	0	
0	0	0	1	
0	0	1	0	UP
0	0	0	1	
0	0	0	0	
0	0	0	0	

5. Design of Synchronous Counters

Step 1: State Diagram

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

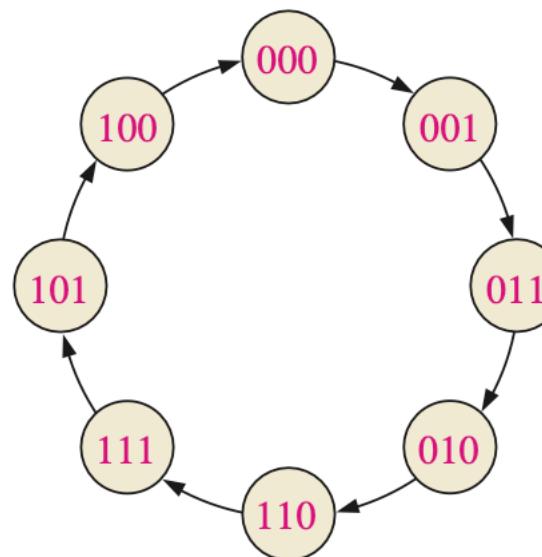


FIGURE 9–26 State diagram for a 3-bit Gray code counter.

Step 2: Next-State Table

The next state is the state that the counter goes to from its present state upon application of a clock pulse.

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

All possible output transitions are listed by showing the Q output of the flip-flop going from present states to next states. To design the counter, the transition table is applied to each of the flip-flops in the counter, based on the next-state 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

- Karnaugh maps can be used to determine the logic required for the J and K inputs of each flip-flop in the counter. There is a Karnaugh map for the J input and a Karnaugh map for the K input of each flip-flop. In this design procedure, each cell in a Karnaugh map represents one of the present states in the counter sequence listed in Table 9–8.
- From the J and K states in the transition table (Table 9–9) a 1, 0, or X is entered into each present-state cell on the maps depending on the transition of the Q output for a particular flip-flop.

Step 4: Karnaugh Maps

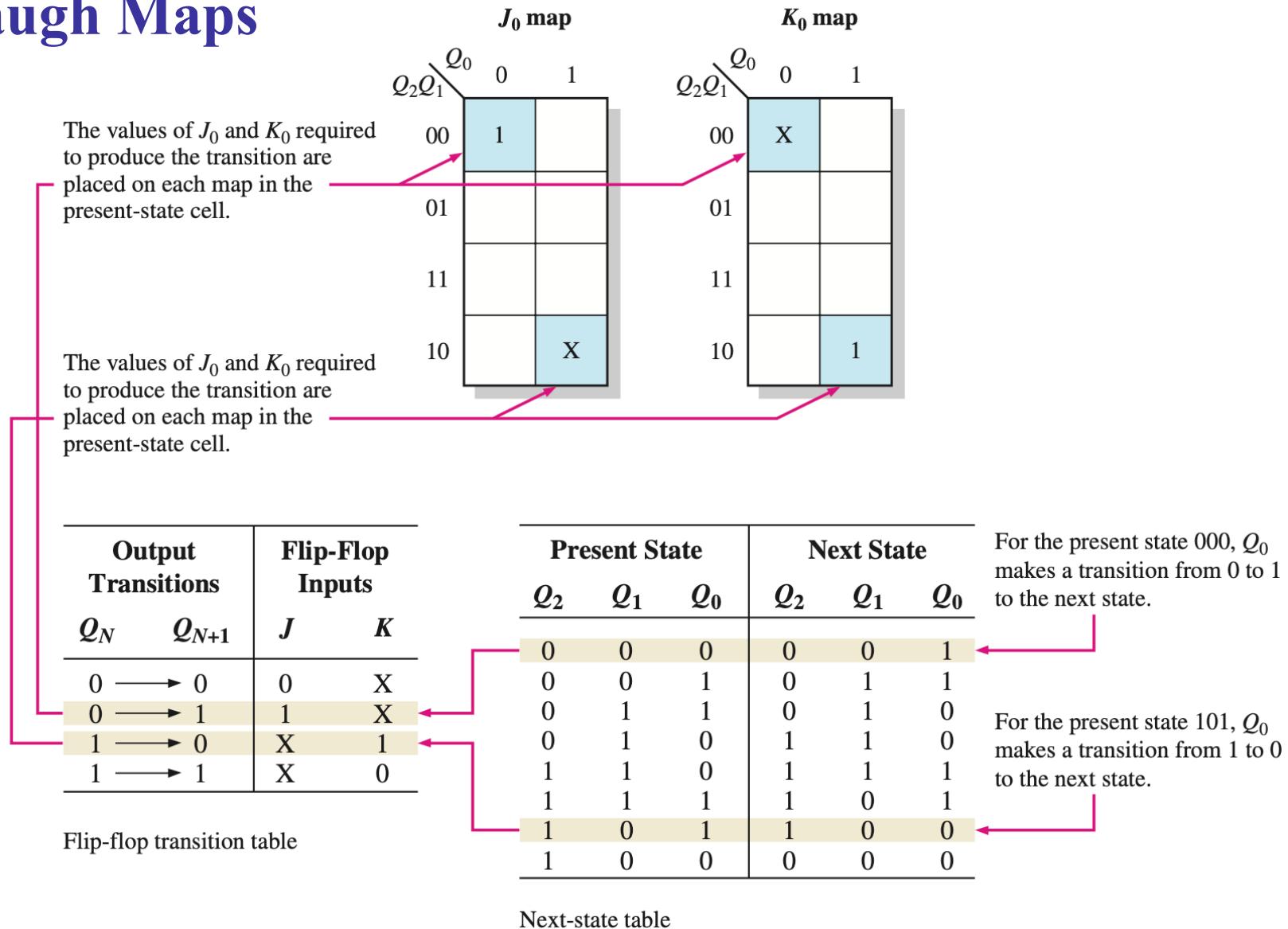


FIGURE 9–27 Examples of the mapping procedure for the counter sequence represented in Table 9–8 and Table 9–9.

Step 4: Karnaugh Maps

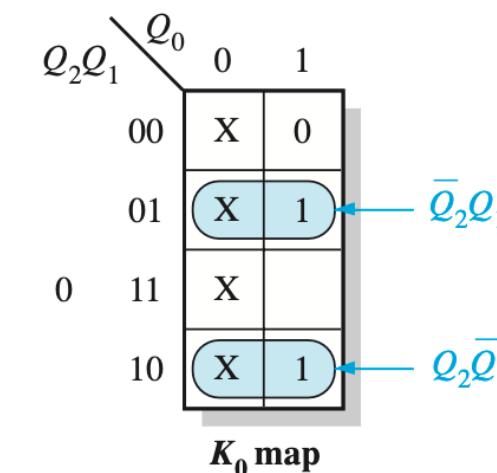
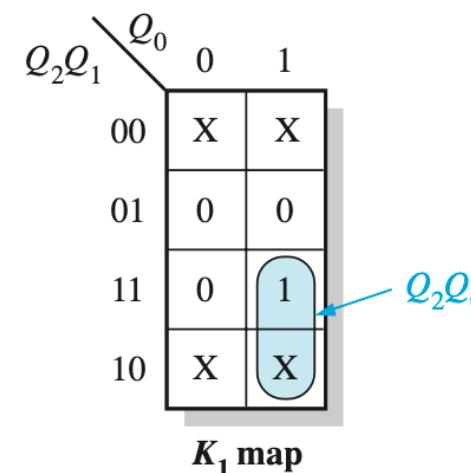
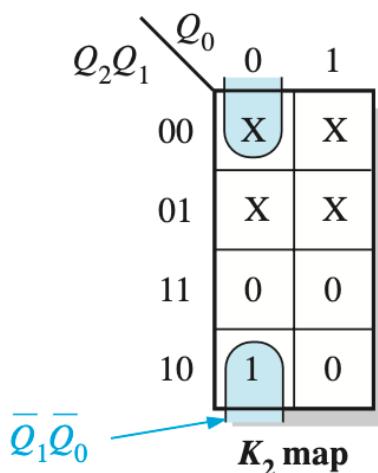
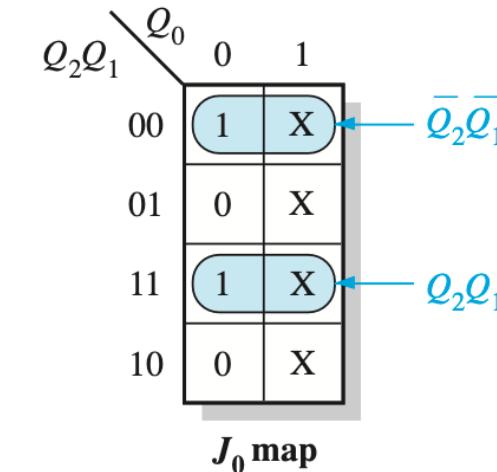
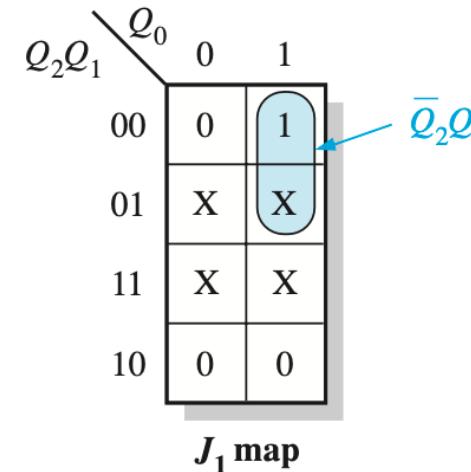
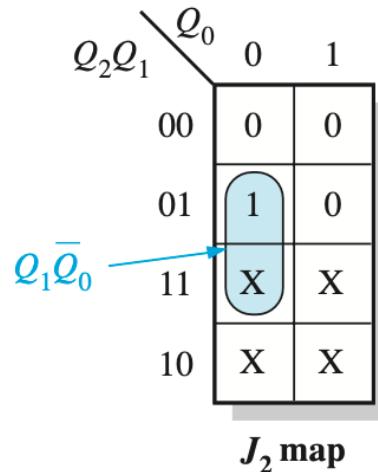


FIGURE 9–28 Karnaugh maps for present-state J and K inputs.

Step 5: Logic Expressions for Flip-Flop Inputs

From the Karnaugh maps of Figure 9–28 you obtain the following expressions for the J and K inputs of each flip-flop:

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

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

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

$$K_1 = Q_2Q_0$$

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

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

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 3-bit Gray code counter as shown in Figure 9–29:

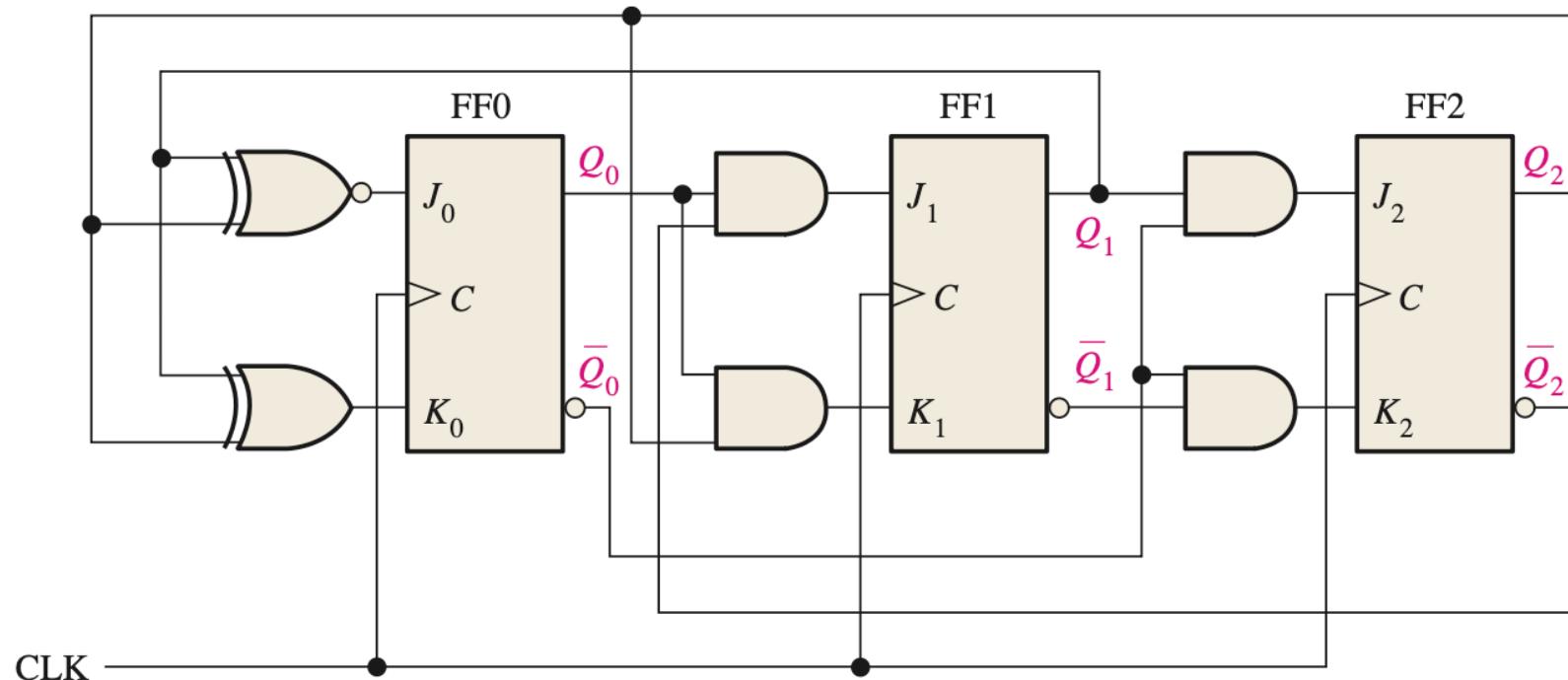


FIGURE 9–29 Three-bit Gray code counter.

A summary of steps used in the design of the 3-bit Gray code counter follows. In general, these steps can be applied to any state machine.

1. Specify the counter sequence and draw a state diagram.
2. Derive a next-state table from the state diagram.
3. Develop a transition table showing the flip-flop inputs required for each transition. The transition table is always the same for a given type of flip-flop.
4. Transfer the J and K states from the transition table to Karnaugh maps. There is a Karnaugh map for each input of each flip-flop.
5. Group the Karnaugh map cells to generate and derive the logic expression for each flip-flop input.
6. Implement the expressions with combinational logic, and combine with the flip-flops to create the counter.

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.

FIGURE 9–30

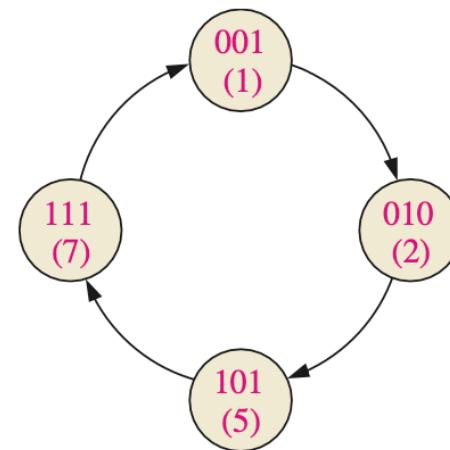


TABLE 9–10

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

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.

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	0	0
0	1	1
1	0	0
1	1	1

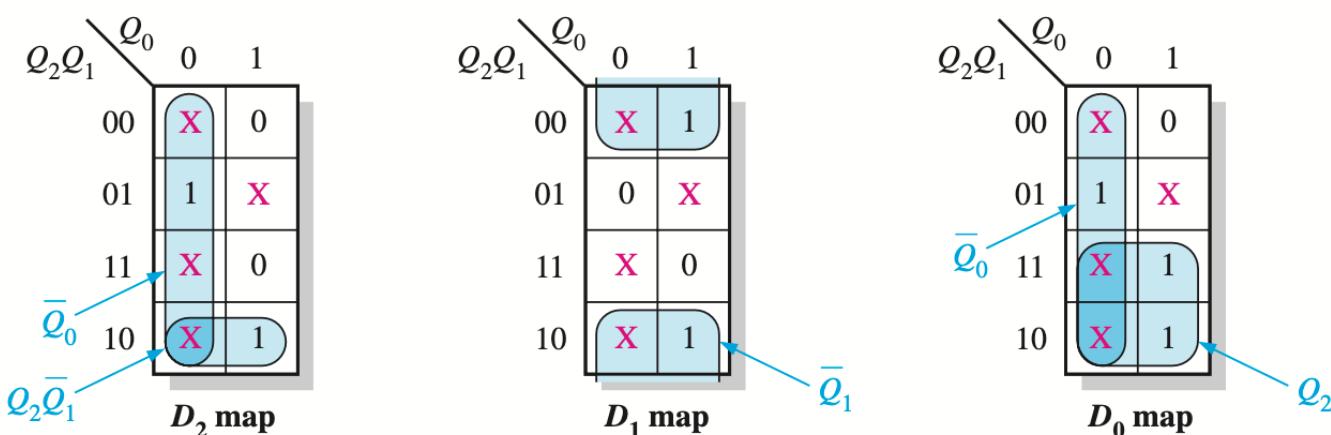


FIGURE 9–31

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.

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:

$$D_0 = \bar{Q}_0 + Q_2$$

$$D_1 = \bar{Q}_1$$

$$D_2 = \bar{Q}_0 + Q_2\bar{Q}_1$$

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

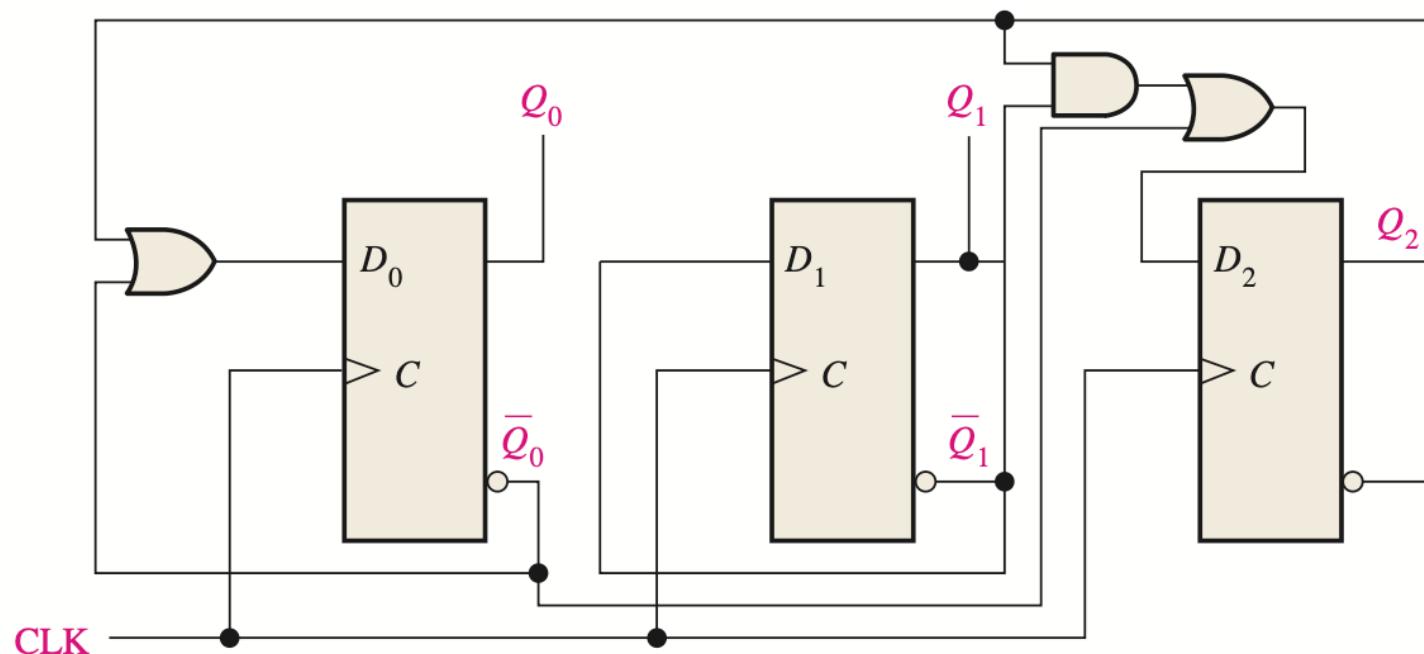


FIGURE 9–32

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/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/DOWN control input, Y .

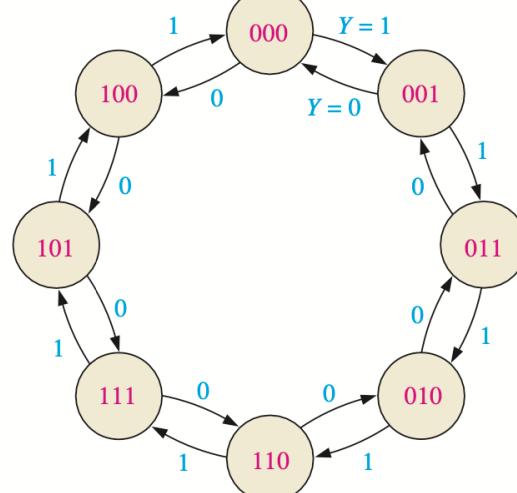


FIGURE 9–33 State diagram for a 3-bit up/down Gray code counter.

TABLE 9–12

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

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

Y = UP/DOWN control input.

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/DOWN control variable, Y .

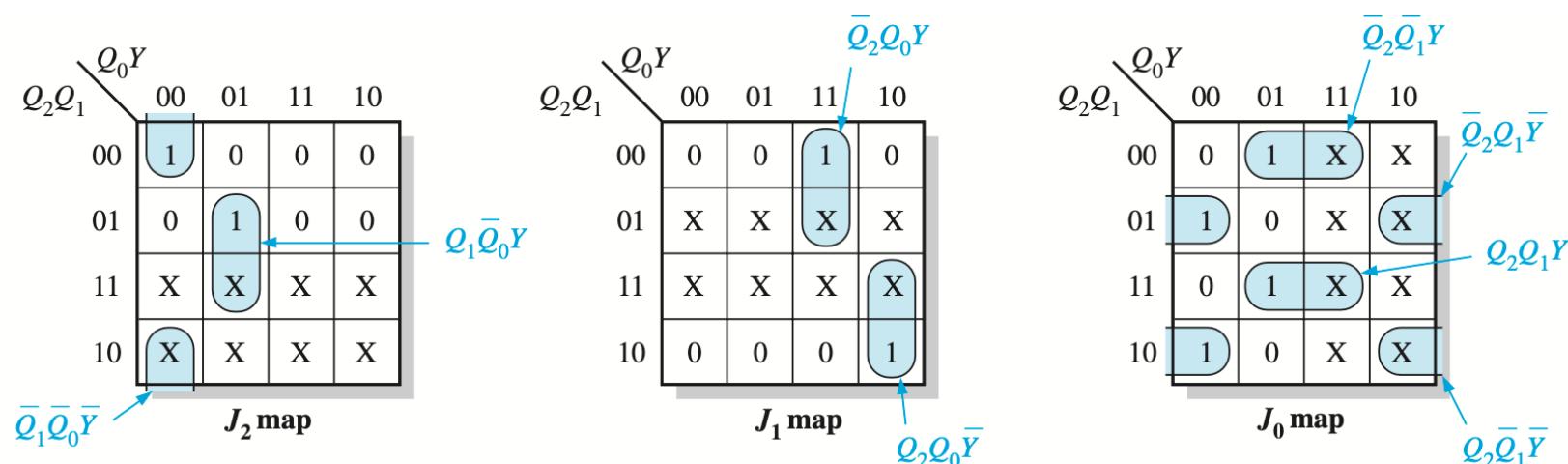
Step 3: The transition table for the J-K flip-flops is repeated in Table 9–13.

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/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.



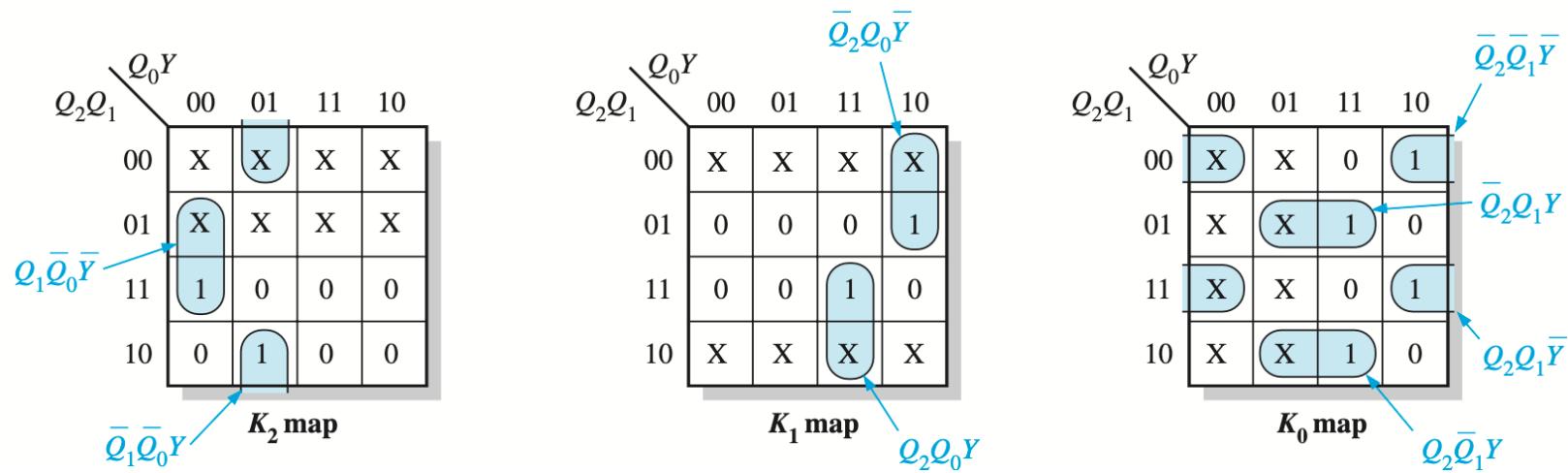


FIGURE 9–34 J and K maps for Table 9–12. The UP/DOWN control input, Y , is treated as a fourth variable.

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:

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

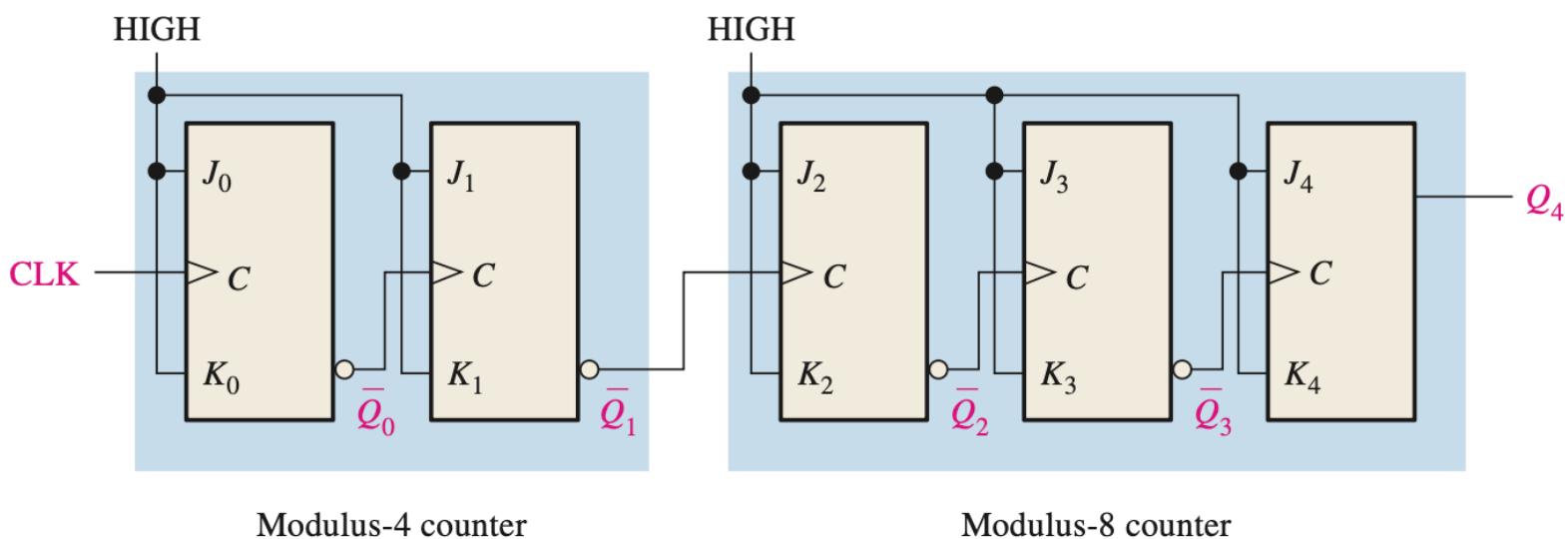
$$J_1 = \bar{Q}_2Q_0Y + Q_2Q_0\bar{Y} \quad K_1 = \bar{Q}_2Q_0\bar{Y} + Q_2Q_0Y$$

$$J_2 = Q_1\bar{Q}_0Y + \bar{Q}_1Q_0\bar{Y} \quad K_2 = Q_1\bar{Q}_0\bar{Y} + \bar{Q}_1Q_0Y$$

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

6. Cascaded Counters

Asynchronous Cascading



Modulus-4 counter

Modulus-8 counter

FIGURE 9–35 Two cascaded asynchronous counters (all J and K inputs are HIGH).

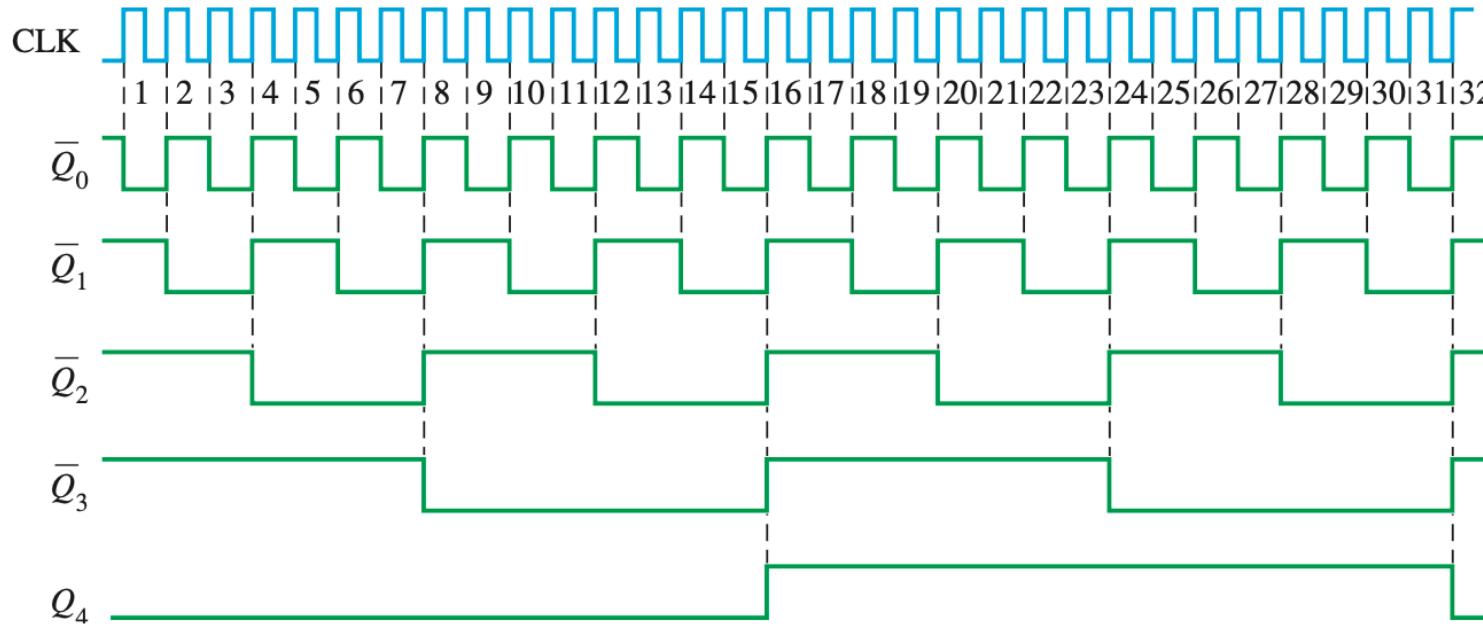


FIGURE 9–36 Timing diagram for the cascaded counter configuration of Figure 9–35.

Synchronous Cascading

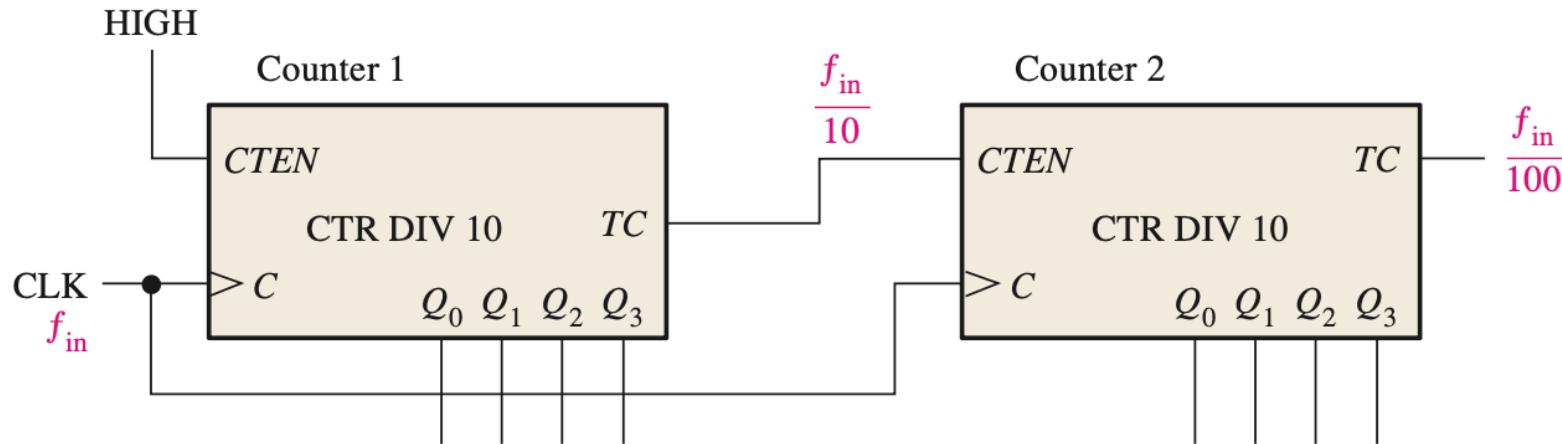


FIGURE 9–37 A modulus-100 counter using two cascaded decade counters.

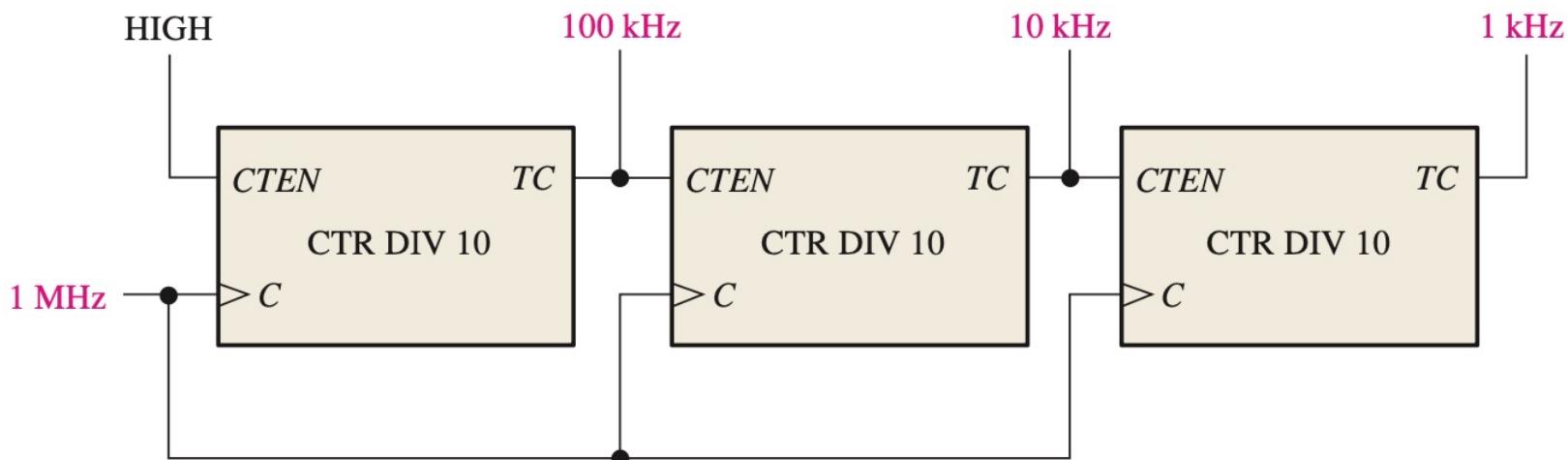
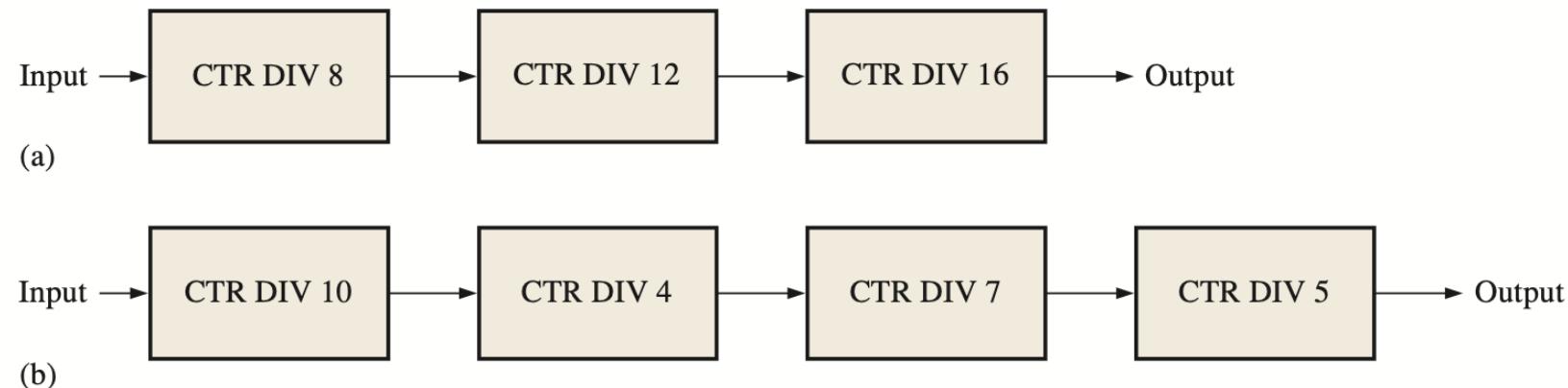


FIGURE 9–38 Three cascaded decade counters forming a divide-by-1000 frequency divider with intermediate divide-by-10 and divide-by-100 outputs.

EXAMPLE 9–6

Determine the overall modulus of the two cascaded counter configurations in Figure 9–39.

**FIGURE 9–39****Solution**

In Figure 9–39(a), the overall modulus for the 3-counter configuration is

$$8 \times 12 \times 16 = 1536$$

In Figure 9–39(b), the overall modulus for the 4-counter configuration is

$$10 \times 4 \times 7 \times 5 = 1400$$

EXAMPLE 9–7

Use 74HC190 up/down decade counters connected in the UP mode to obtain a 10 kHz waveform from a 1 MHz clock. Show the logic diagram.

Solution

To obtain 10 kHz from a 1 MHz clock requires a division factor of 100. Two 74HC190 counters must be cascaded as shown in Figure 9–40. The left counter produces a terminal count (*MAX/MIN*) pulse for every 10 clock pulses. The right counter produces a terminal count (*MAX/MIN*) pulse for every 100 clock pulses.

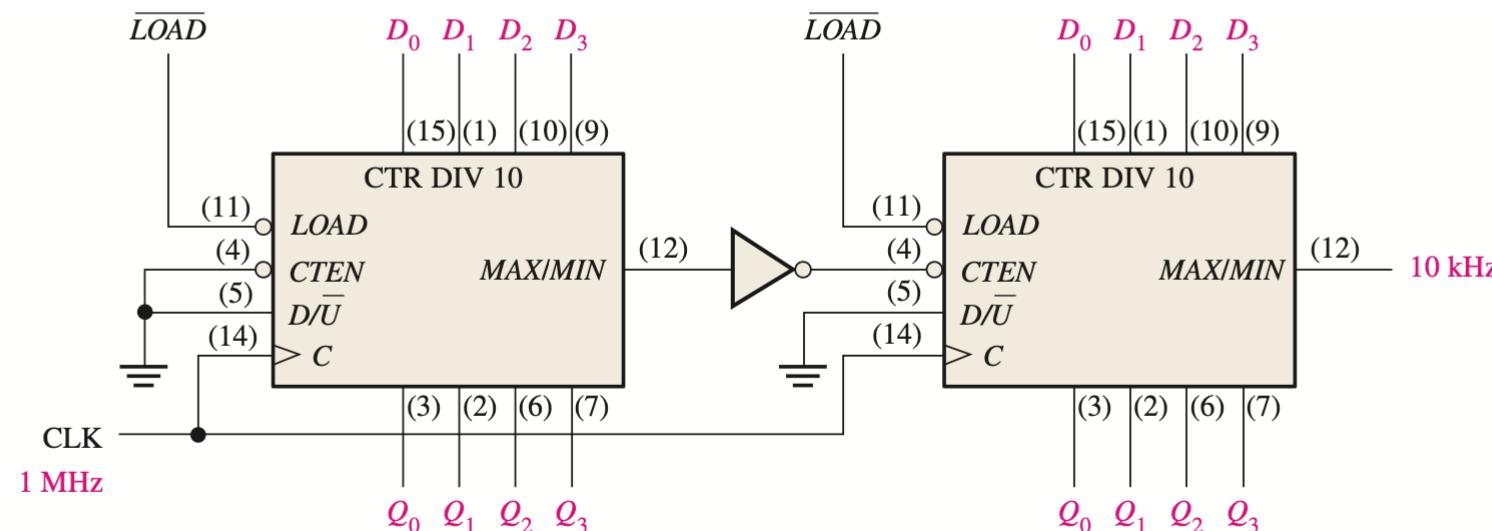


FIGURE 9–40 A divide-by-100 counter using two 74HC190 up/down decade counters connected for the up sequence.

Cascaded Counters with Truncated Sequences

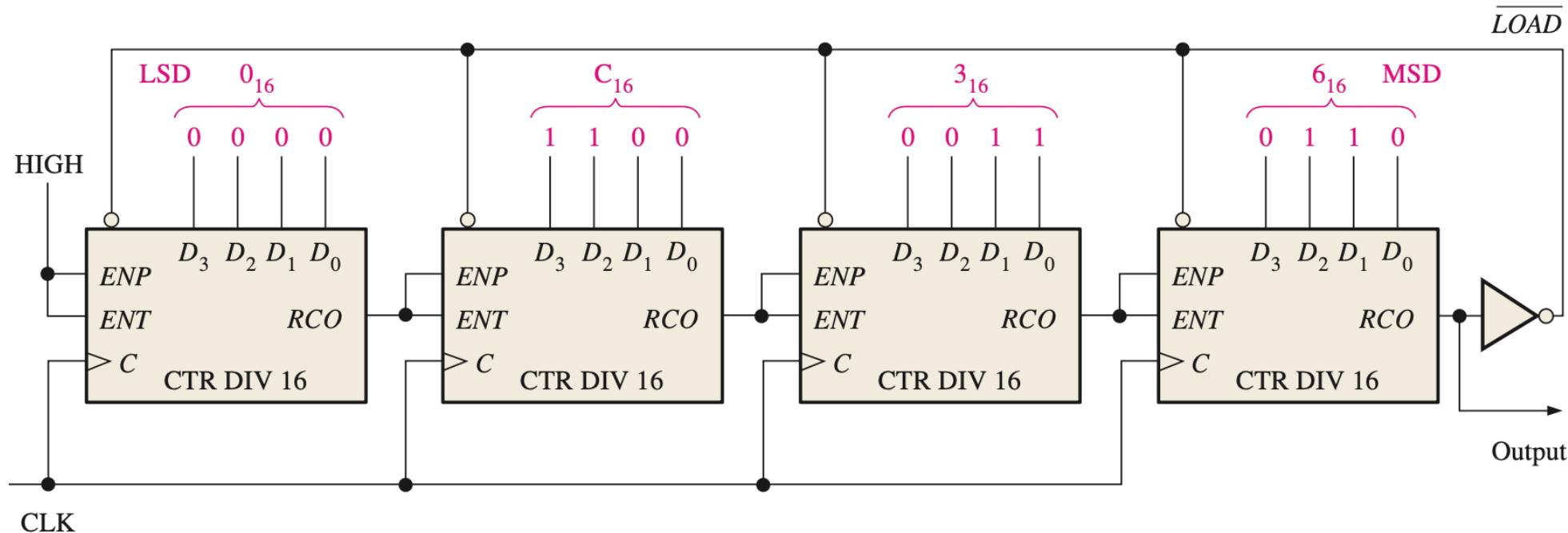


FIGURE 9-41 A divide-by-40,000 counter using 74HC161 4-bit binary counters. Note that each of the parallel data inputs is shown in binary order (the right-most bit D_0 is the LSB in each counter).

7. Counter Decoding

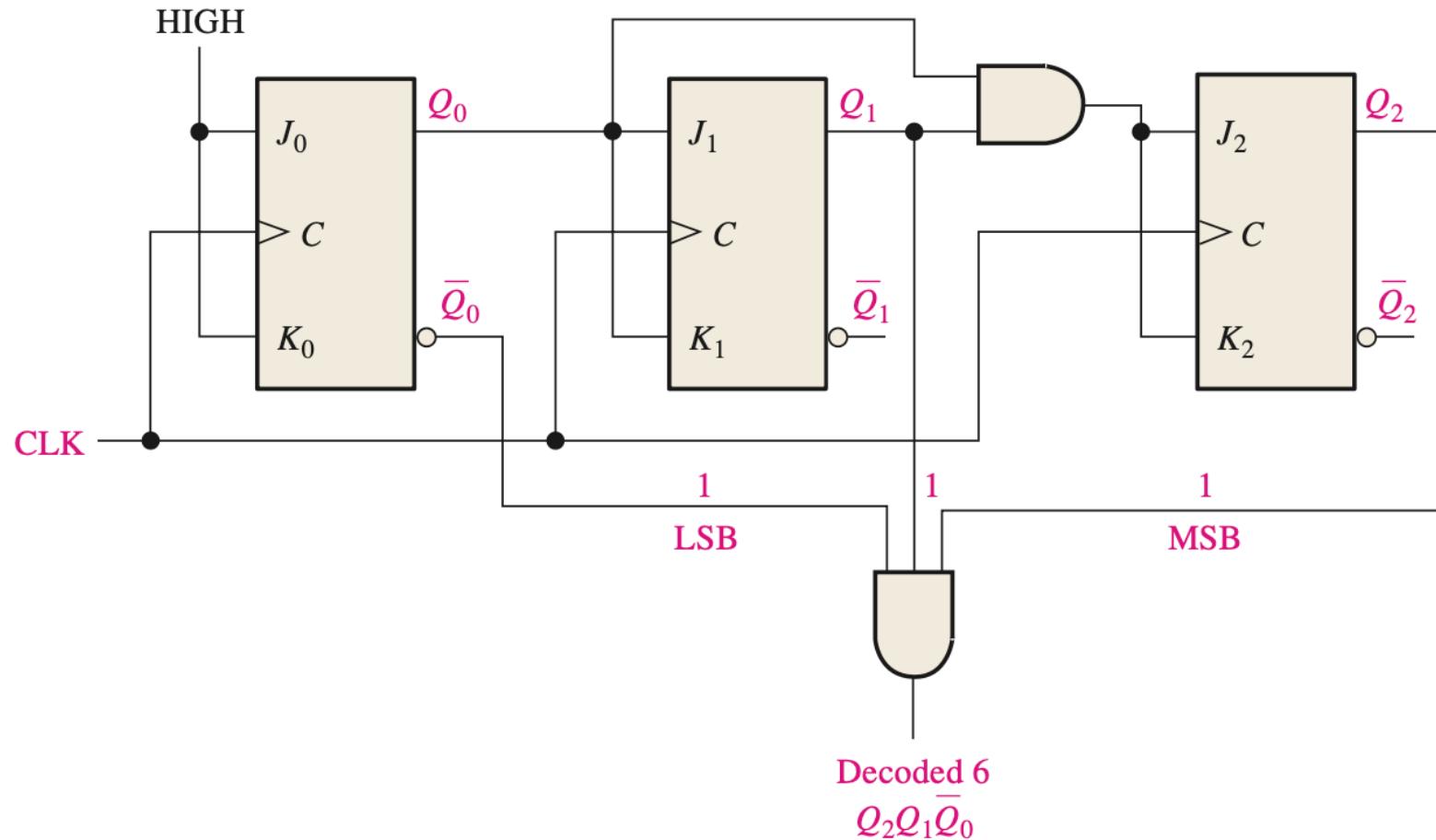


FIGURE 9–42 Decoding of state 6 (110).

EXAMPLE 9-8

Implement the decoding of binary state 2 and binary state 7 of a 3-bit synchronous counter. Show the entire counter timing diagram and the output waveforms of the decoding gates. Binary 2 = $\bar{Q}_2Q_1\bar{Q}_0$ and binary 7 = $Q_2Q_1Q_0$.

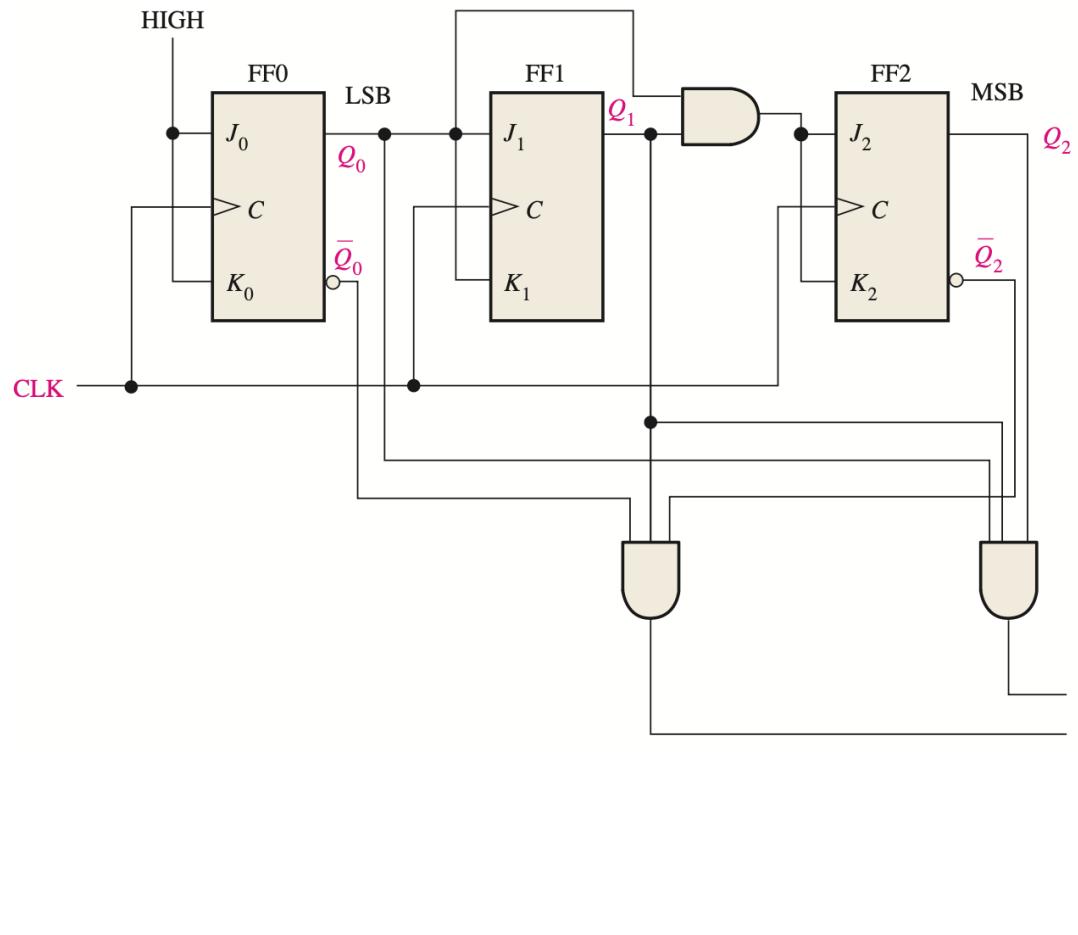


FIGURE 9-43 A 3-bit counter with active-HIGH decoding of count 2 and count 7.

8. Counter Applications

A Digital Clock

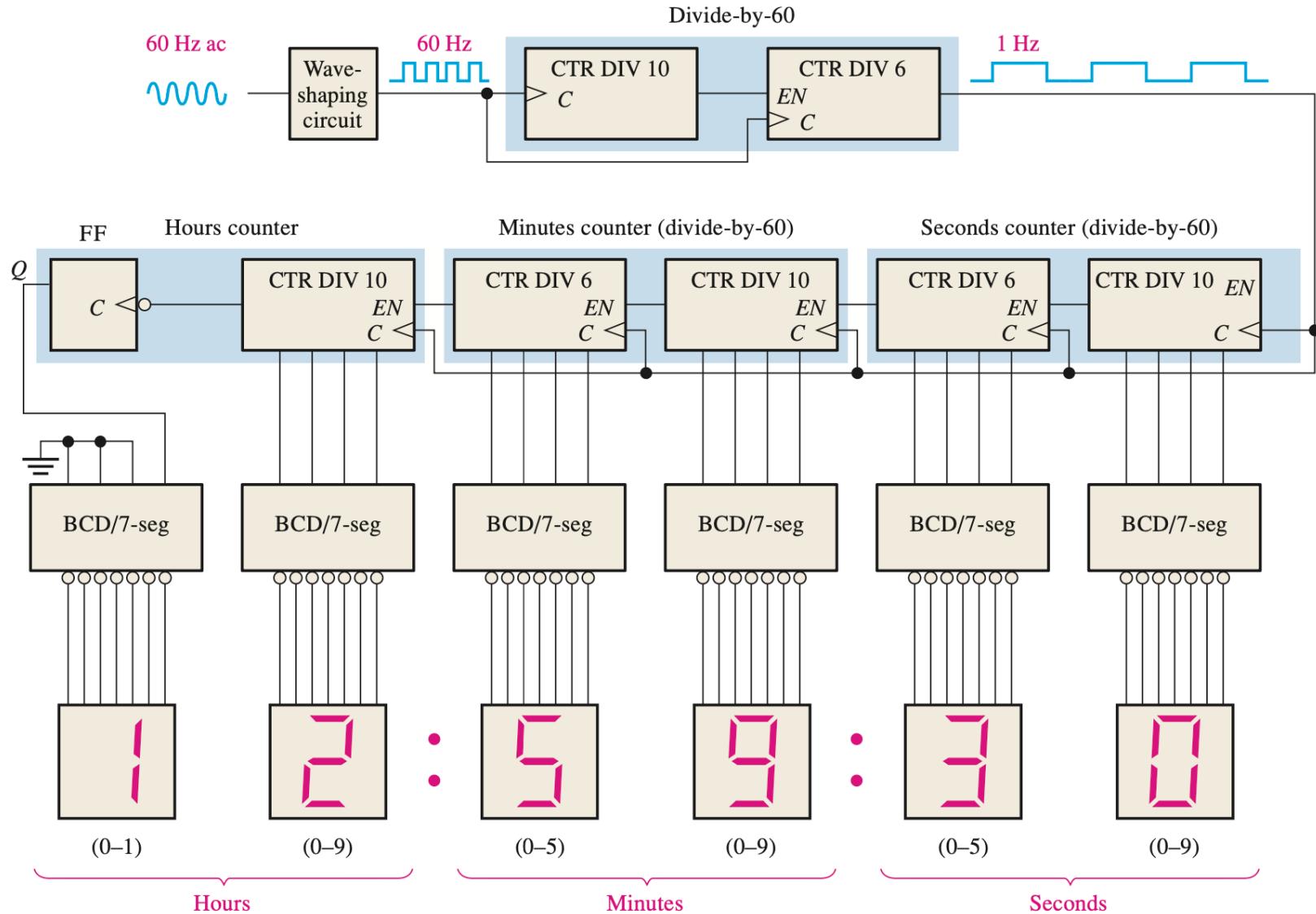


FIGURE 9–48 Simplified logic diagram for a 12-hour digital clock.

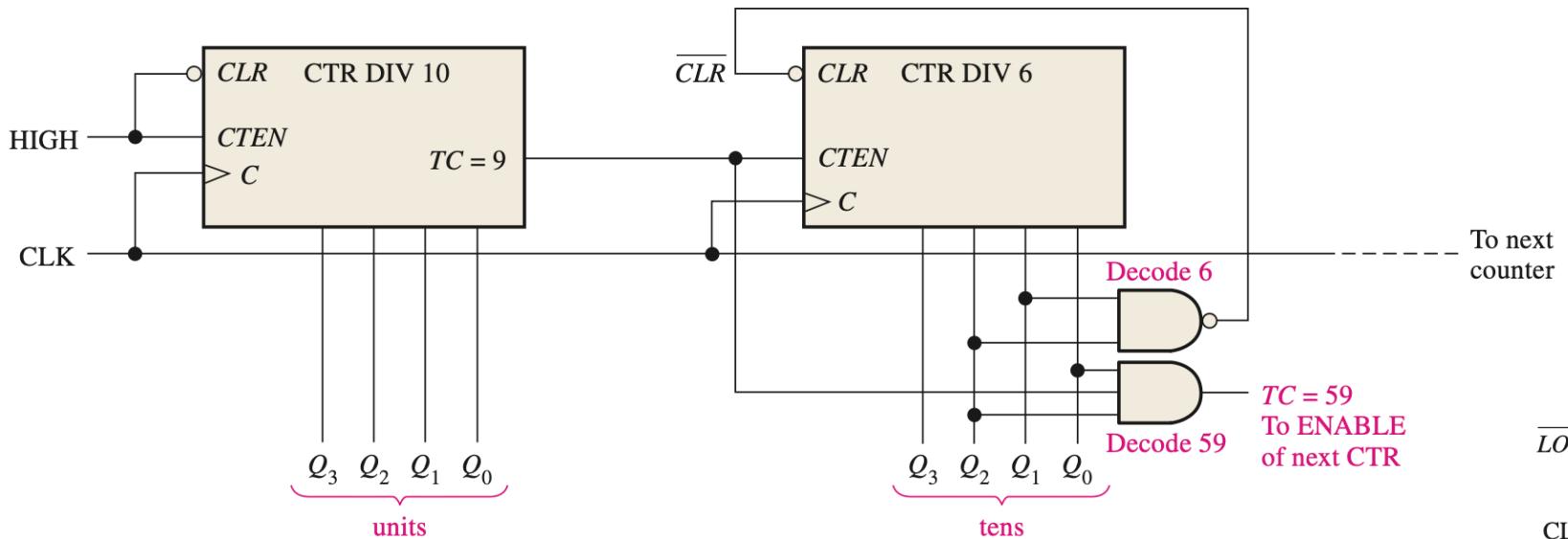


FIGURE 9–49 Logic diagram of typical divide-by-60 counter using synchronous decade counters. Note that the outputs are in binary order (the right-most bit is the LSB).

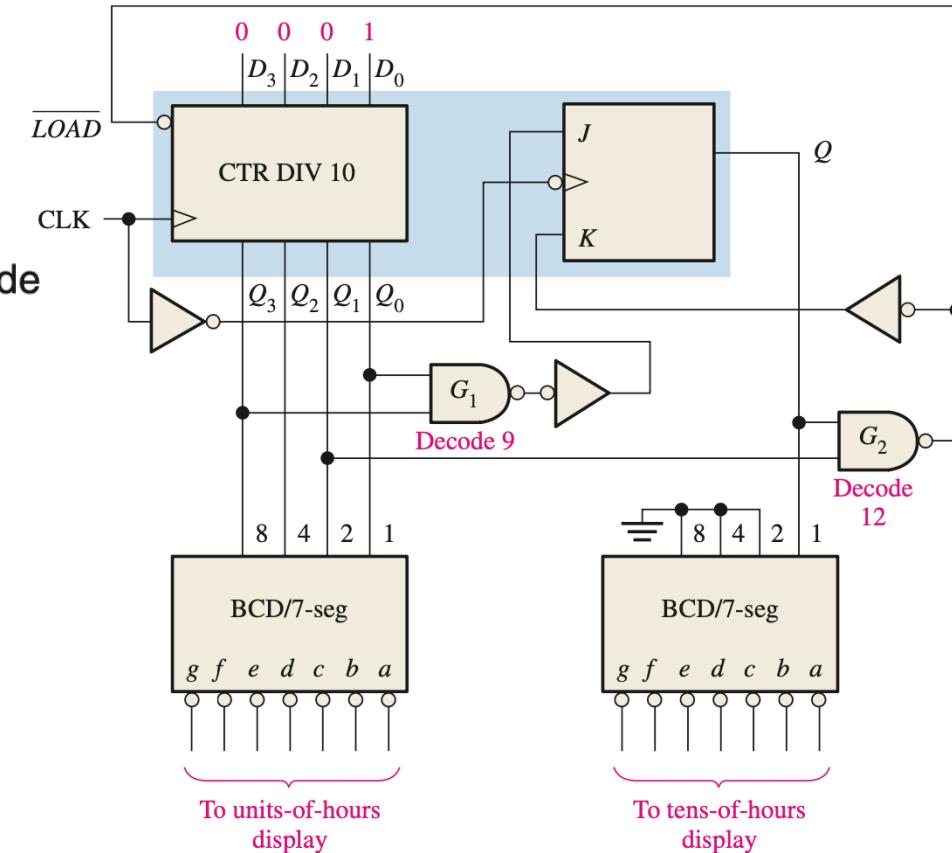


FIGURE 9–50 Logic diagram for hours counter and decoders. Note that on the counter inputs and outputs, the right-most bit is the LSB.

A Digital Clock

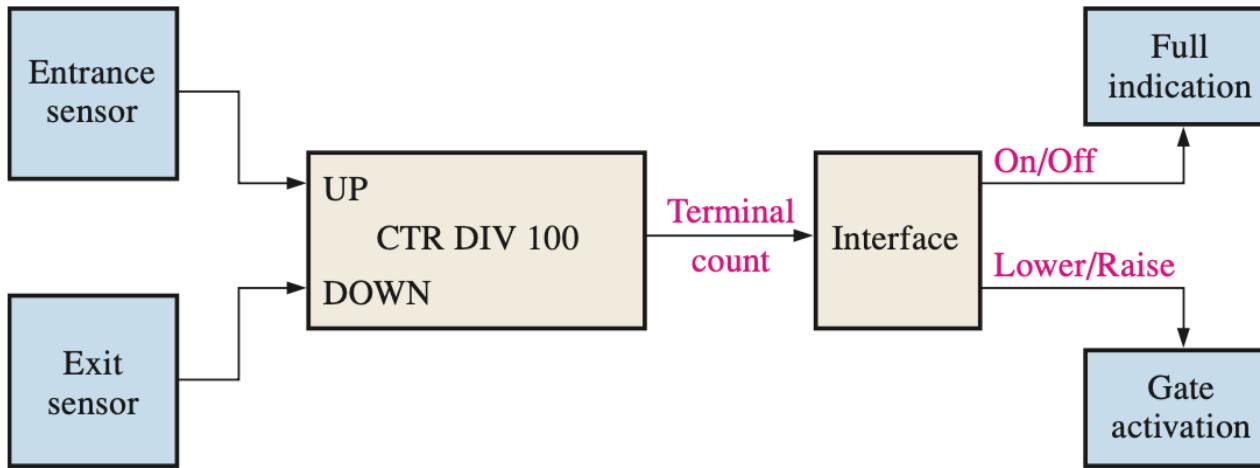


FIGURE 9-51 Functional block diagram for parking garage control.

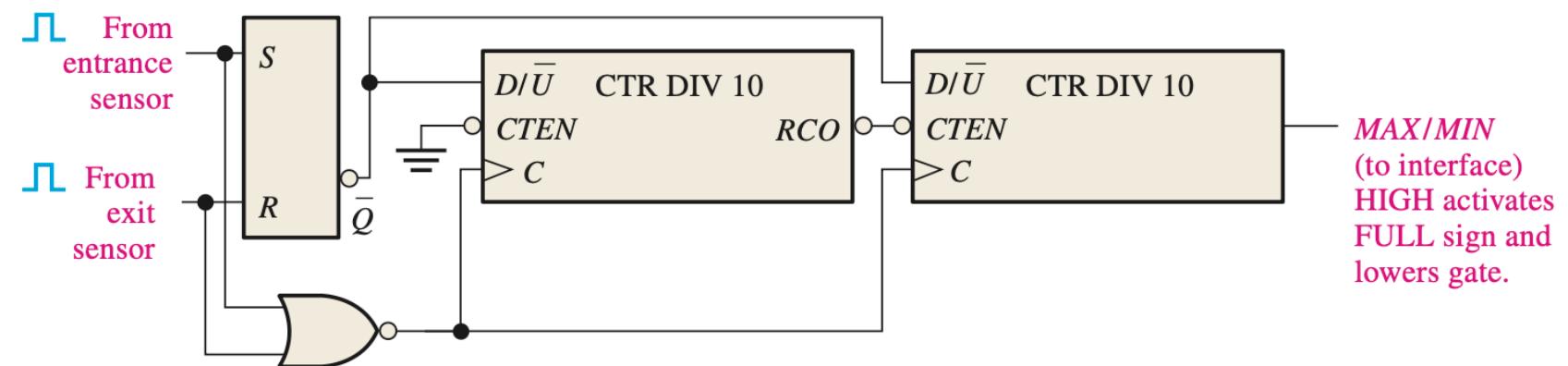
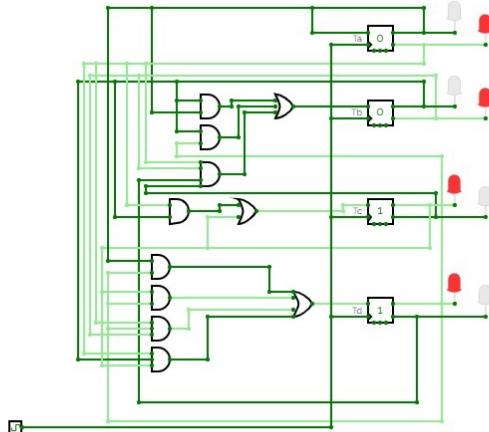


FIGURE 9-52 Logic diagram for modulus-100 up/down counter for automobile parking control.



THE END

Lecture 9: Counters



INSTRUCTOR: Dr. Vuong Quoc Bao