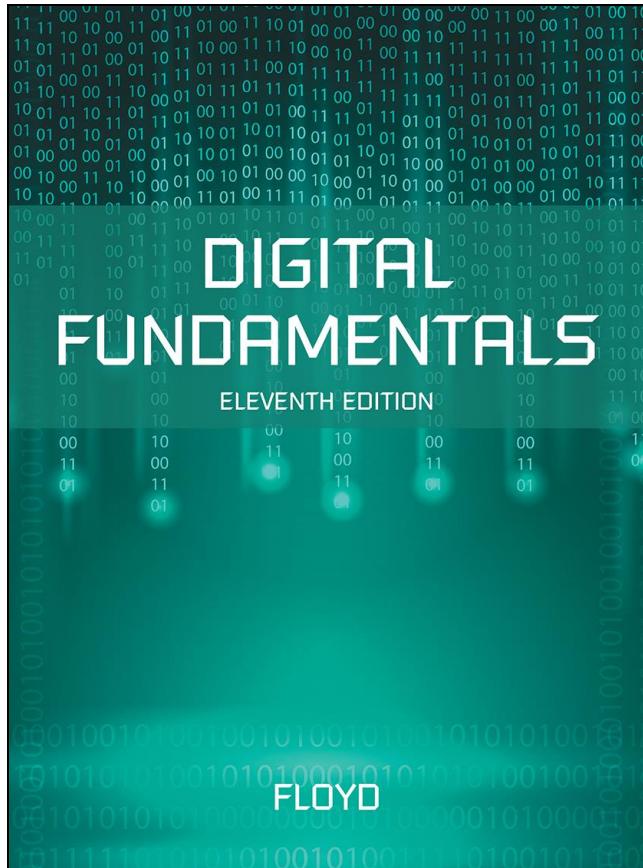


# Digital Fundamentals

ELEVENTH EDITION



## CHAPTER 9

### Counters

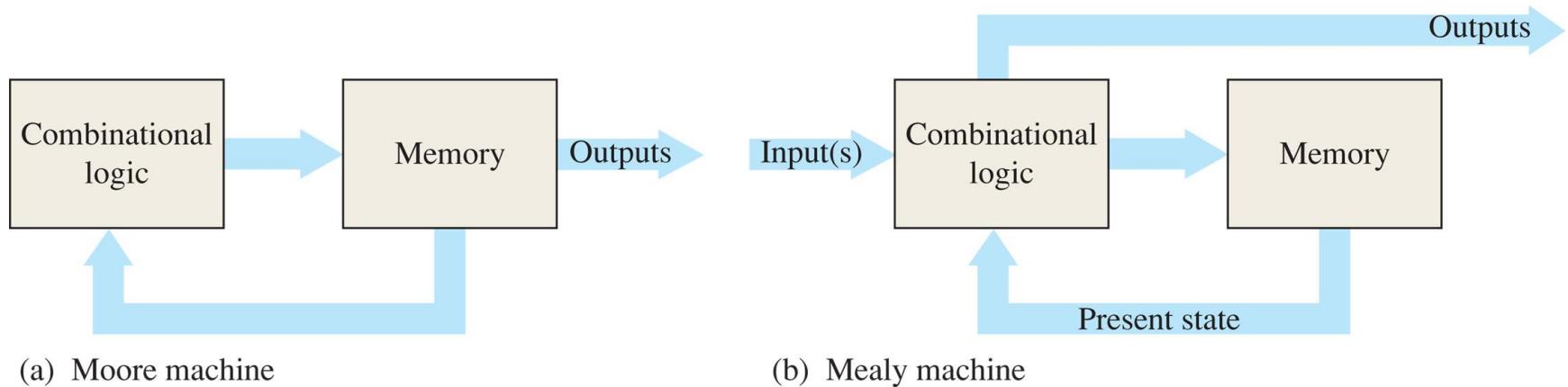
## Finite State Machines

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.

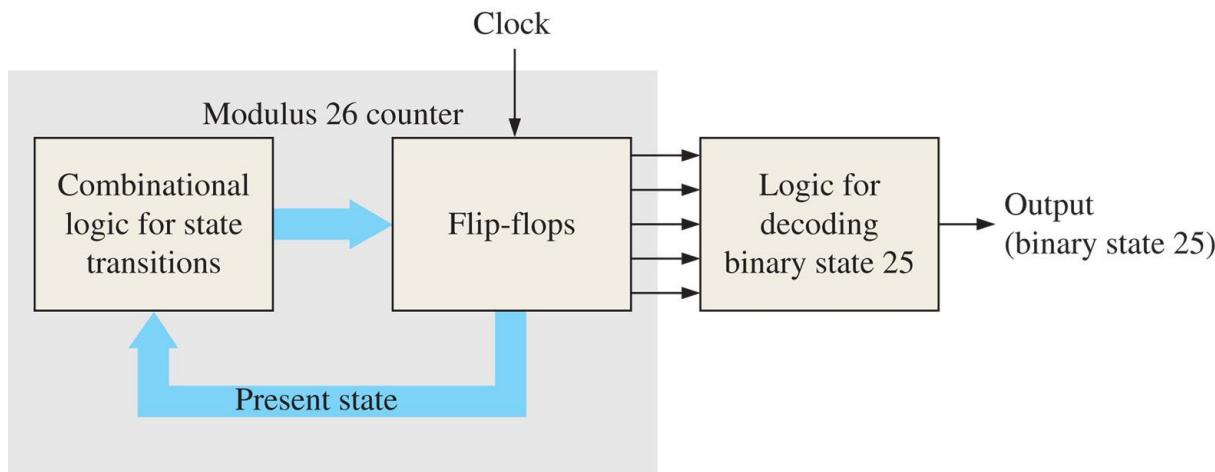
The Moore state machine is one where the outputs depend only on the internal present state.

The Mealy state machine is one where the outputs depend on both the internal present state and on the inputs

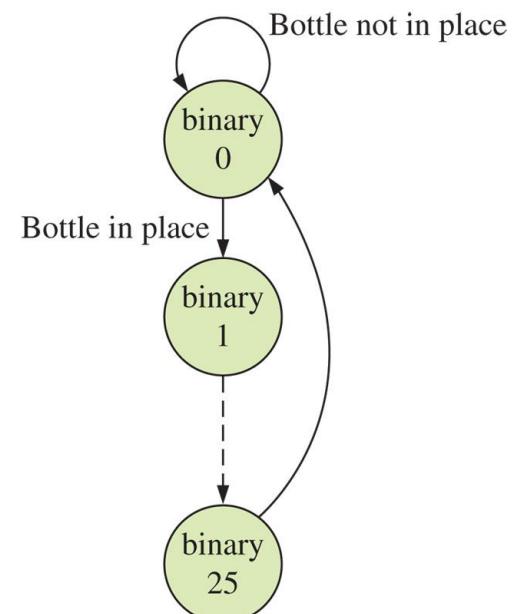
**FIGURE 9-1** Two types of sequential logic.



**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.



(a) Moore machine



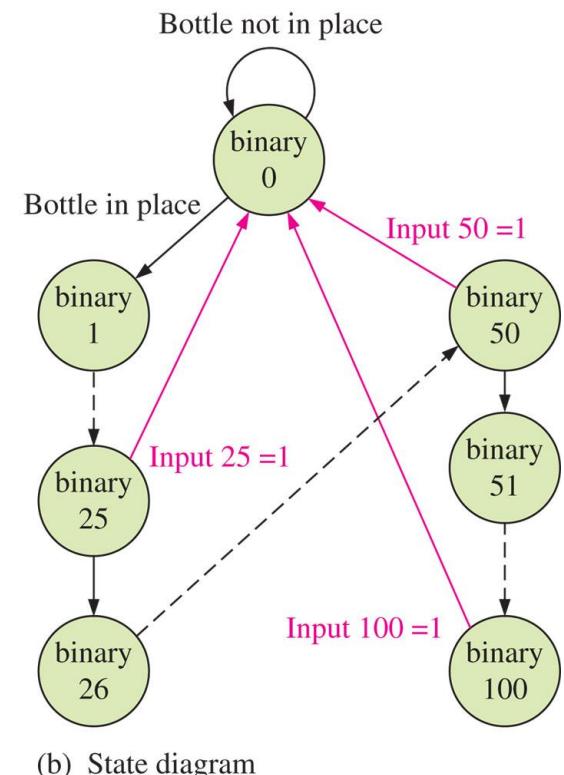
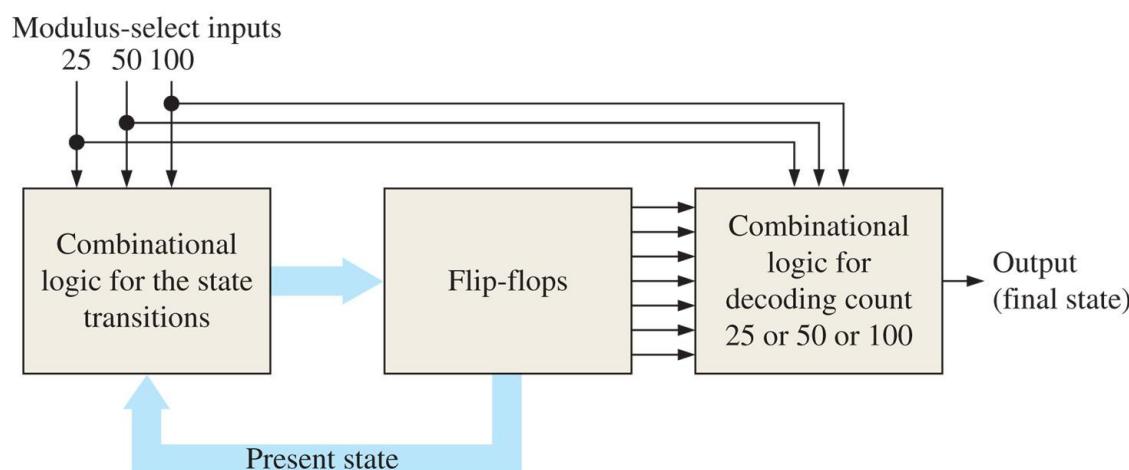
(b) State diagram

There is no input in this case, other than the clock, so the next state is determined only by the present state, which makes this a Moore machine. One tablet is bottled for each clock pulse.

Once a bottle is in place, the first tablet is inserted at binary state 1, the second at binary state 2, and the twenty-fifth tablet when the binary state is 25. Count 25 is decoded and used to stop the flow of tablets and the clock.

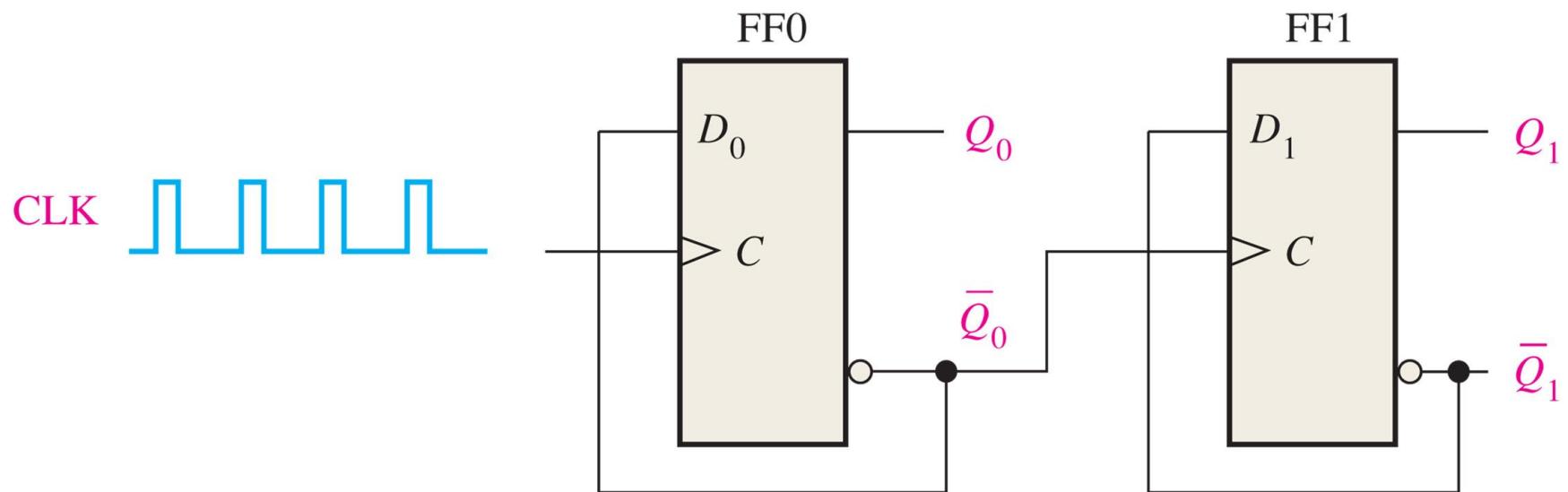
The counter stays in the 0 state until the next bottle is in position (indicated by a 1). Then the clock resumes, the count goes to 1, and the cycle repeats, as illustrated by the state diagram in Figure 9–2(b)

**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.

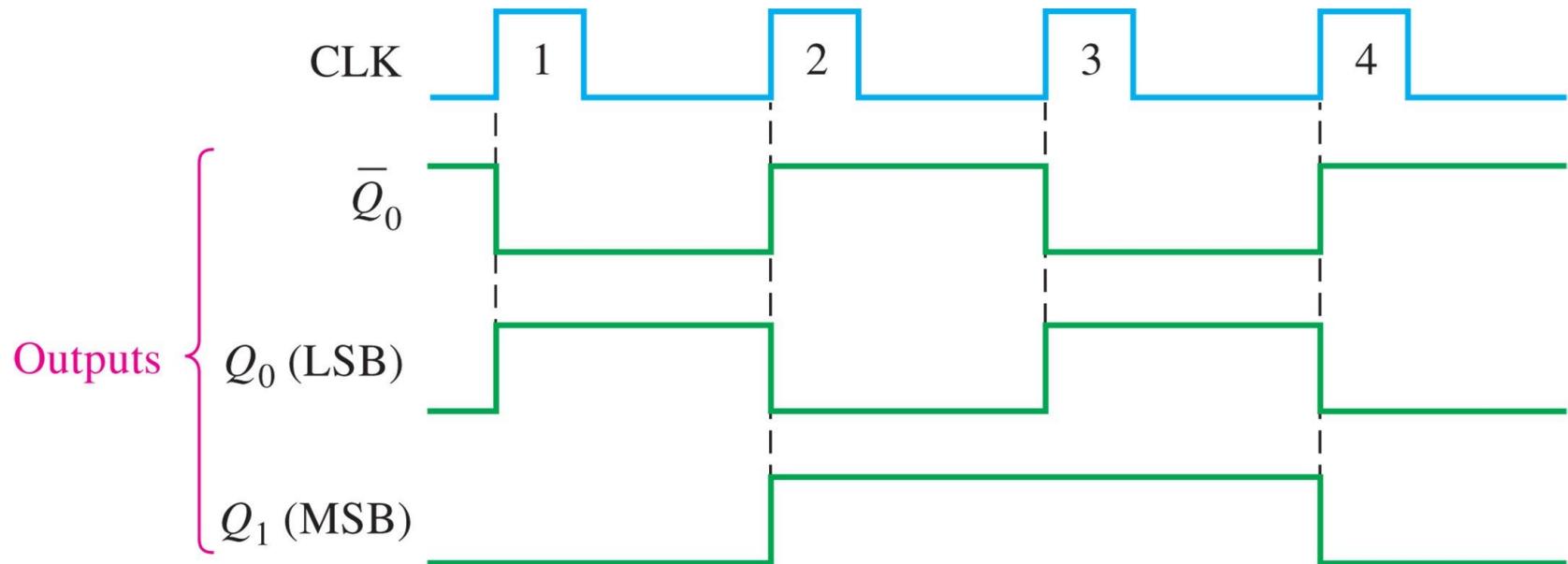


**Example of a Mealy Machine** Let's assume that the tablet-bottling system uses three different sizes of bottles: a 25-tablet bottle, a 50-tablet bottle, and a 100-tablet bottle. This operation requires a state machine with three different terminal counts: 25, 50, and 100. One approach is illustrated in Figure 9–3(a). The combinational logic sets the modulus of the counter depending on the modulus-select inputs. The output of the counter depends on both the present state and the modulus-select inputs, making this a Mealy machine. The state diagram is shown in part (b)

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



**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.

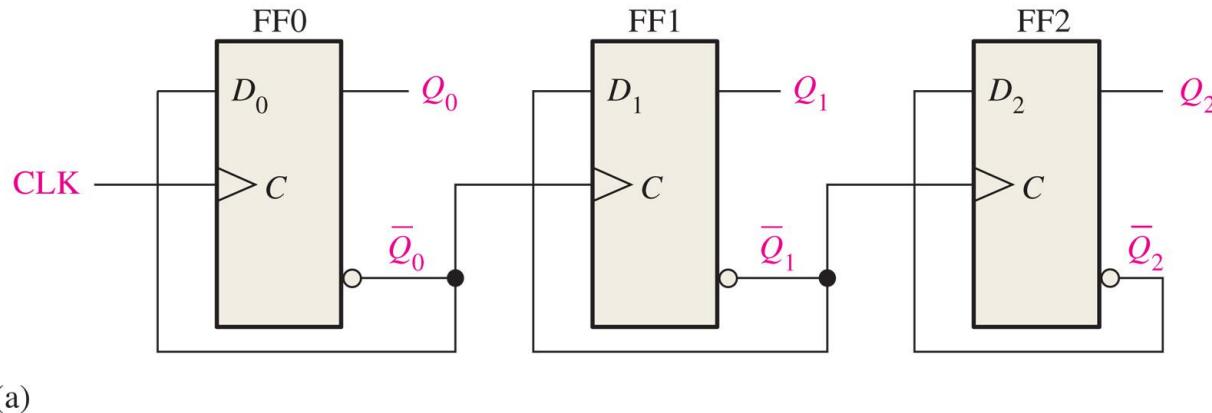
Clock Pulse	$Q_1$	$Q_0$
Initially	0	0
1	0	1
2	1	0
3	1	1
4 (recycles)	0	0

**TABLE 9–2**

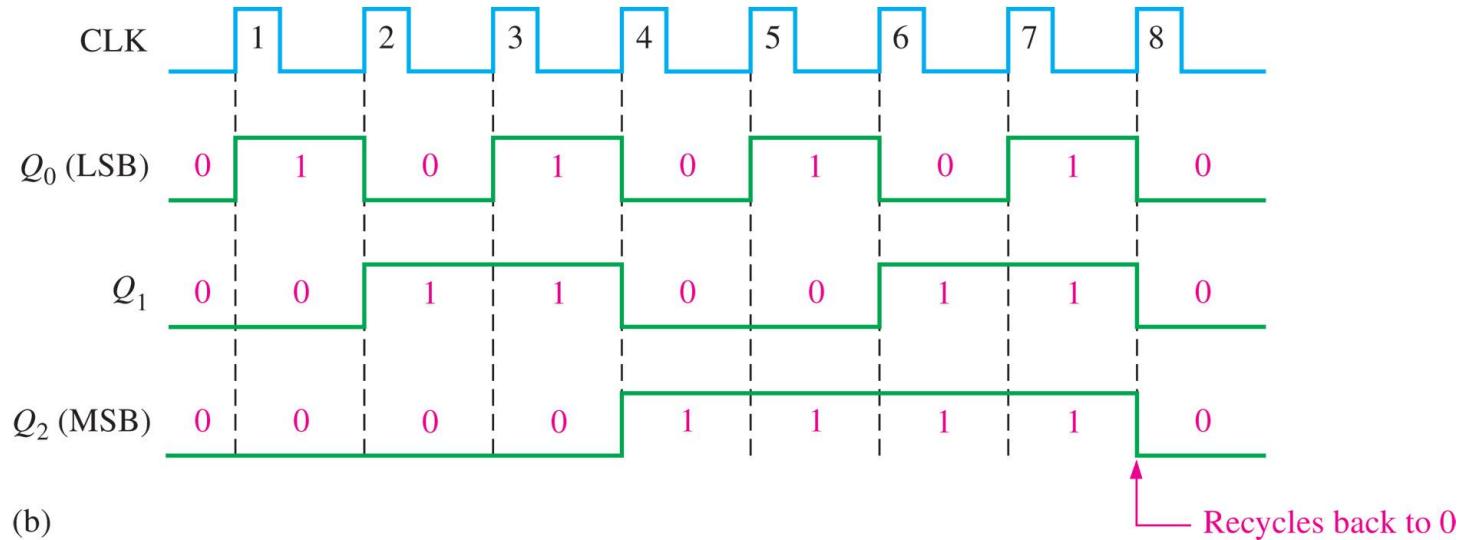
State sequence for a 3-bit binary counter.

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.

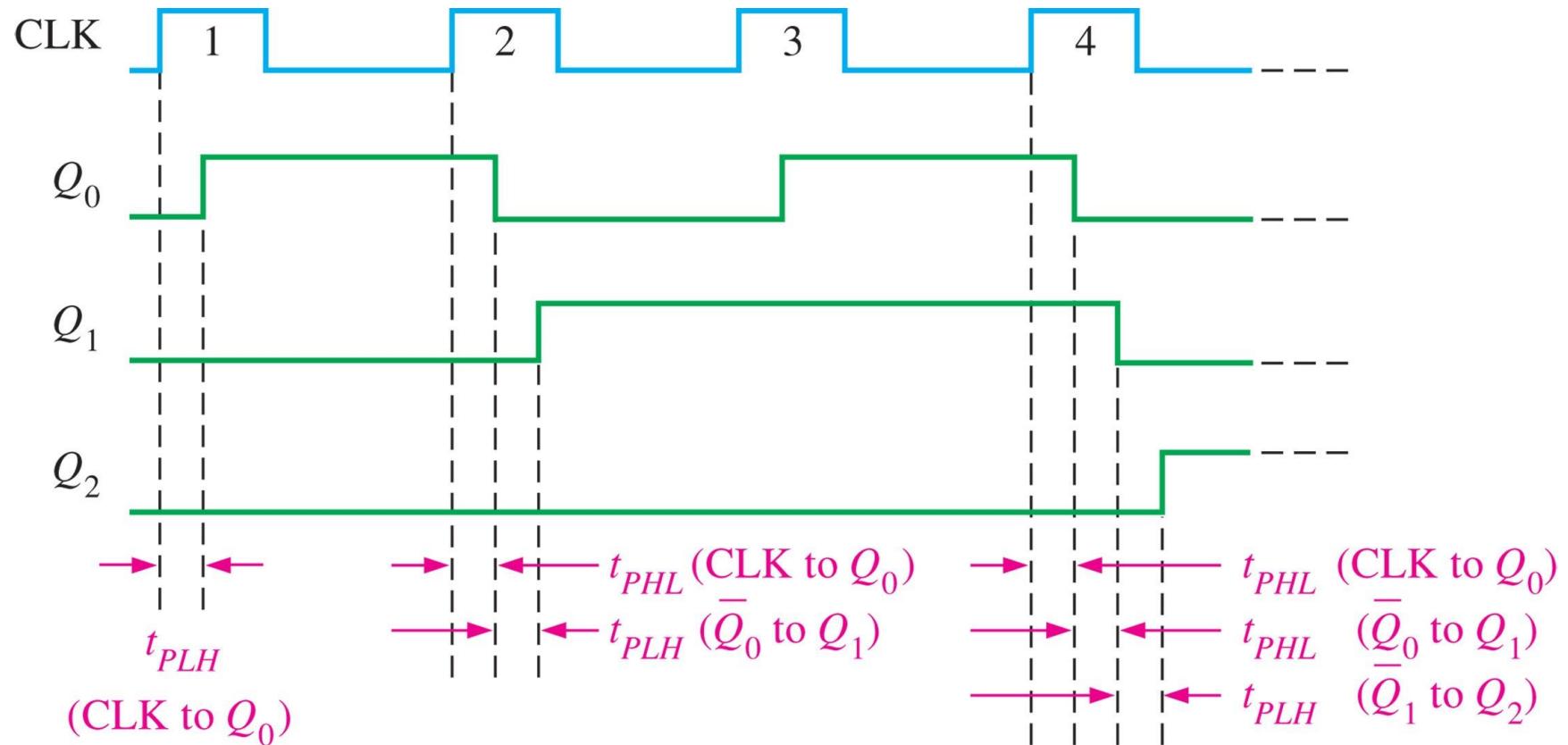


(a)

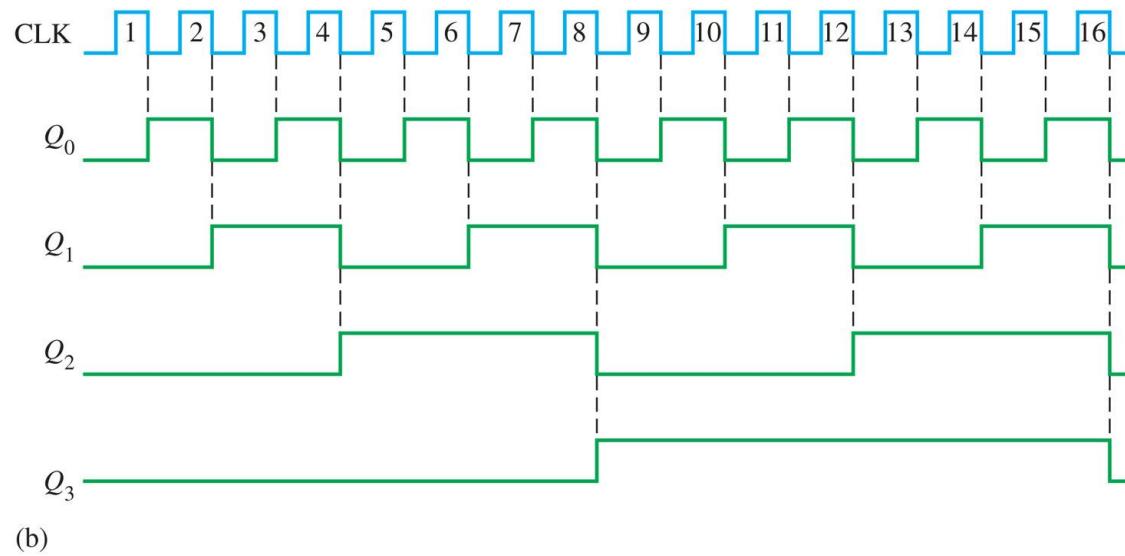
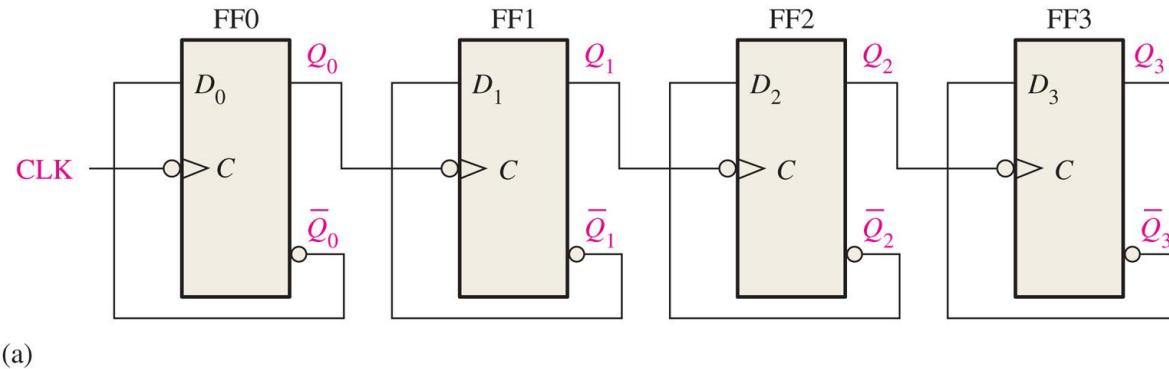


(b)

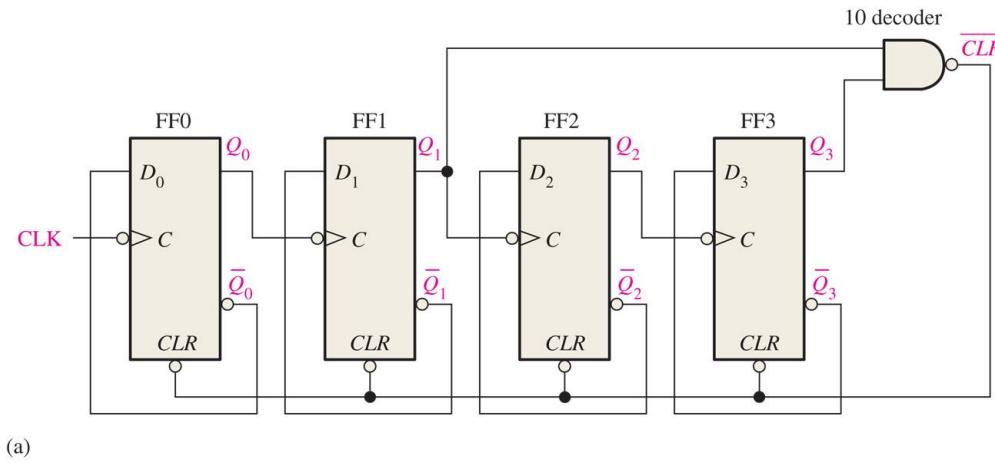
**FIGURE 9-7** Propagation delays in a 3-bit asynchronous (ripple-clocked) binary counter.



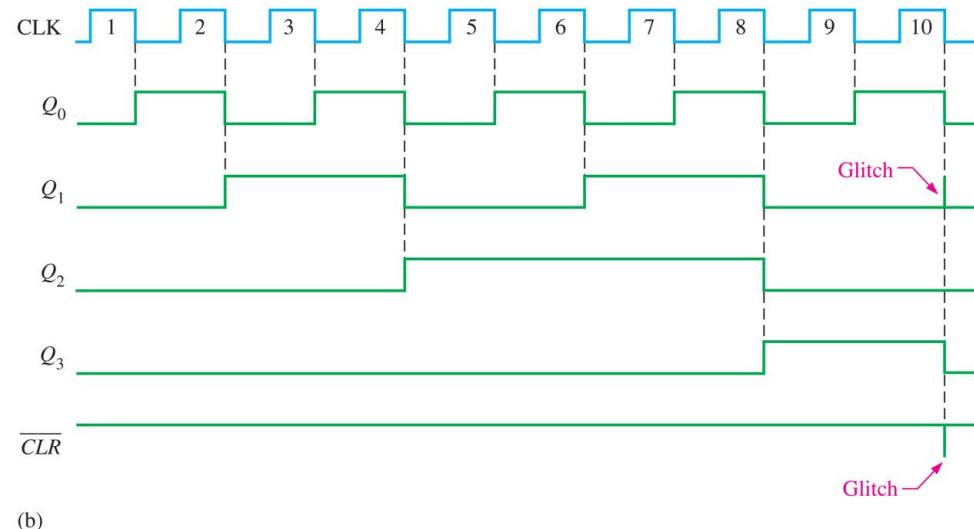
**FIGURE 9-8** Four-bit asynchronous binary counter and its timing diagram.



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

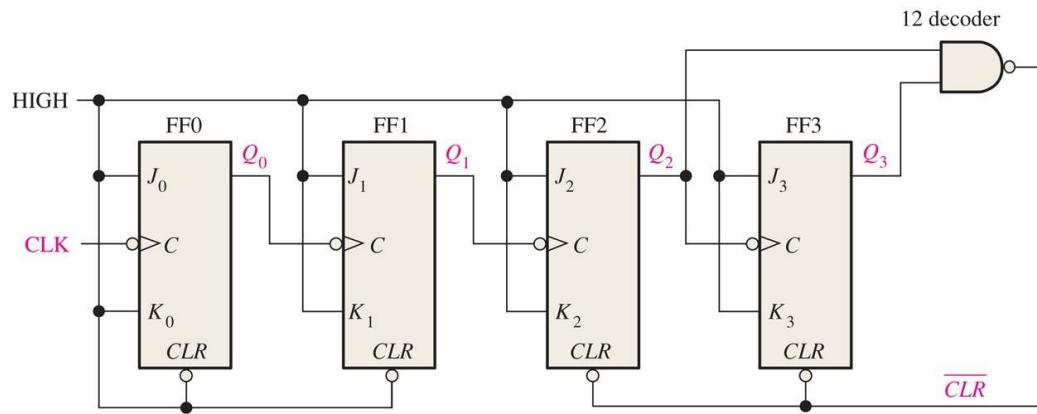


(a)

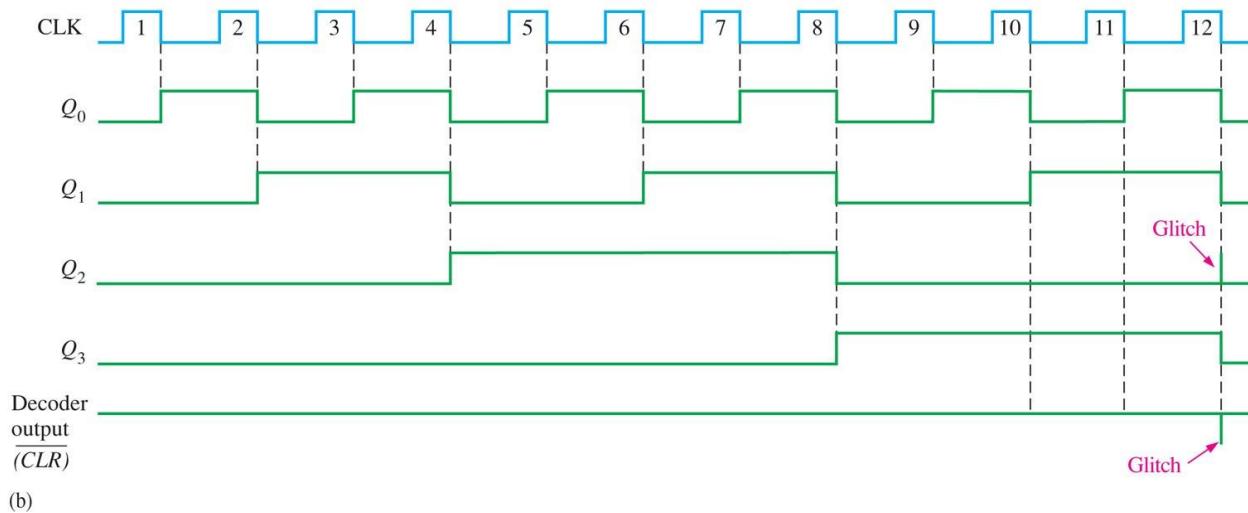


(b)

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

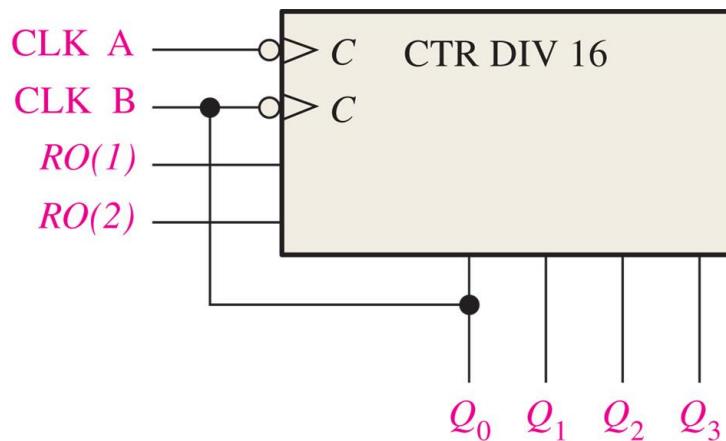


(a)

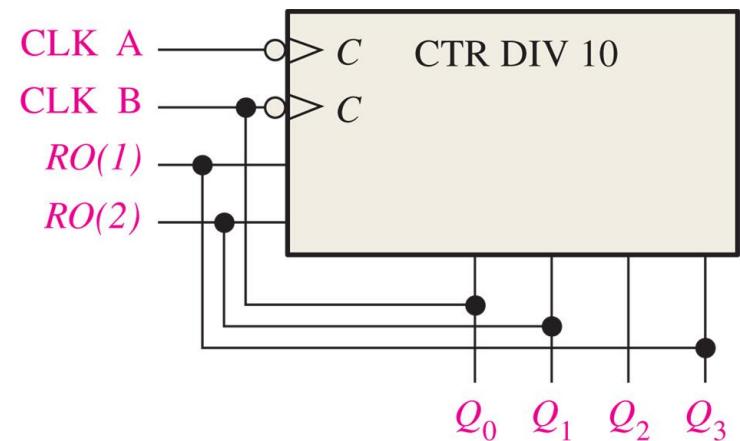


(b)

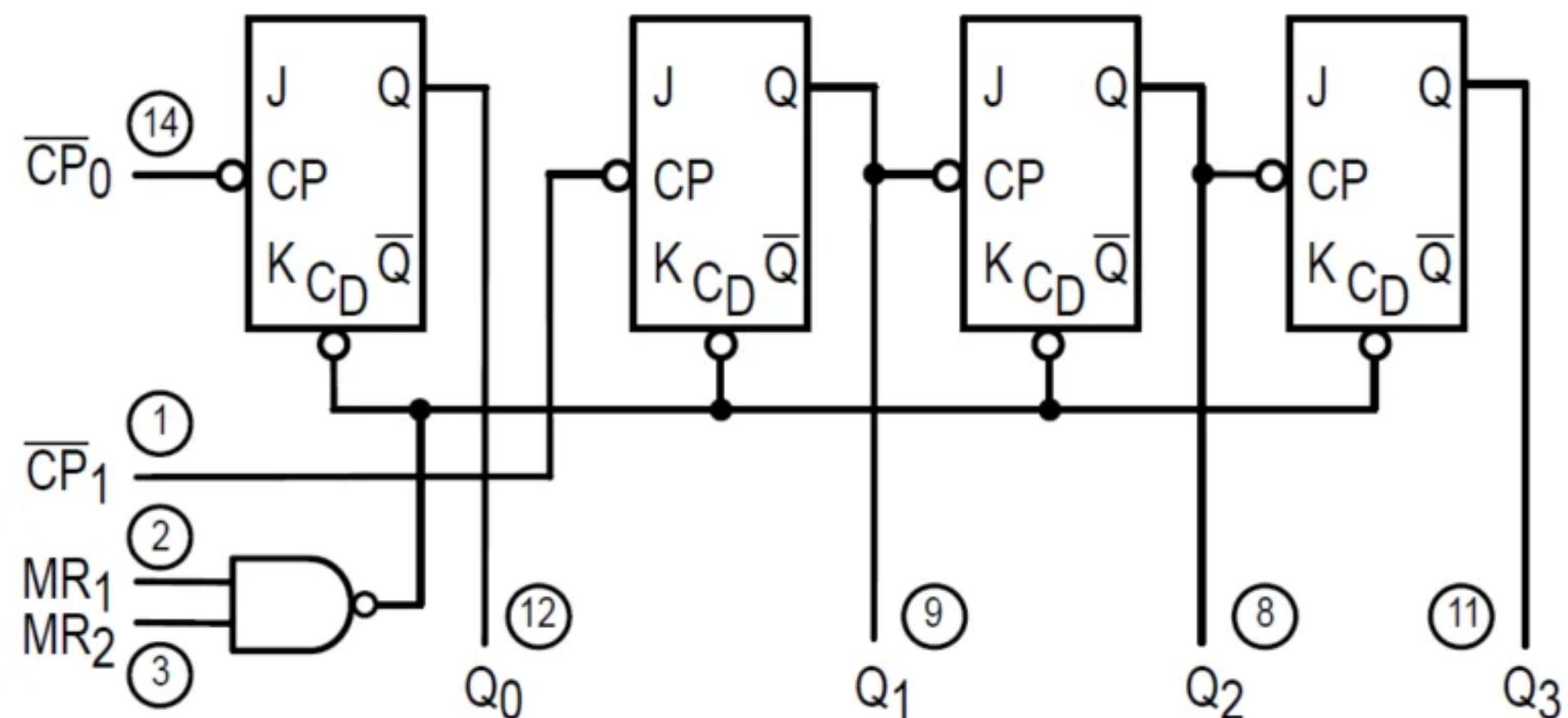
**FIGURE 9-11** Two configurations of the 74HC93 asynchronous counter.  
(The qualifying label, CTR DIV  $n$ , indicates a counter with  $n$  states.)



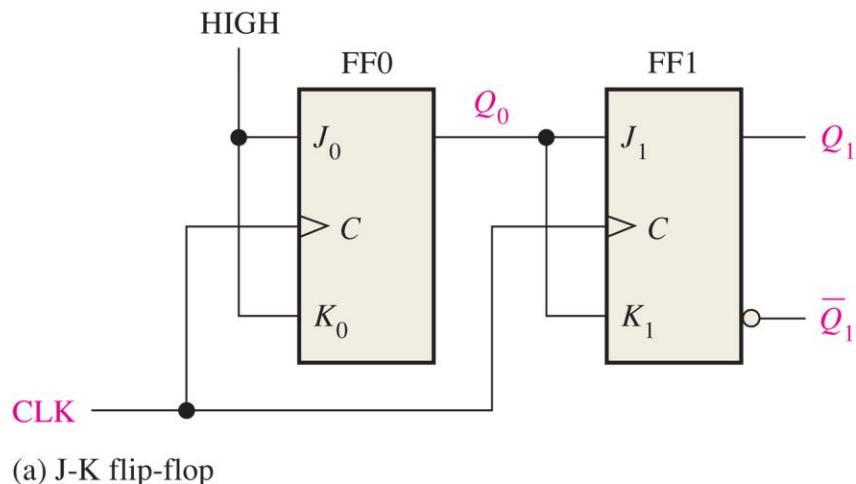
(a) 74HC93 connected as a modulus-16 counter



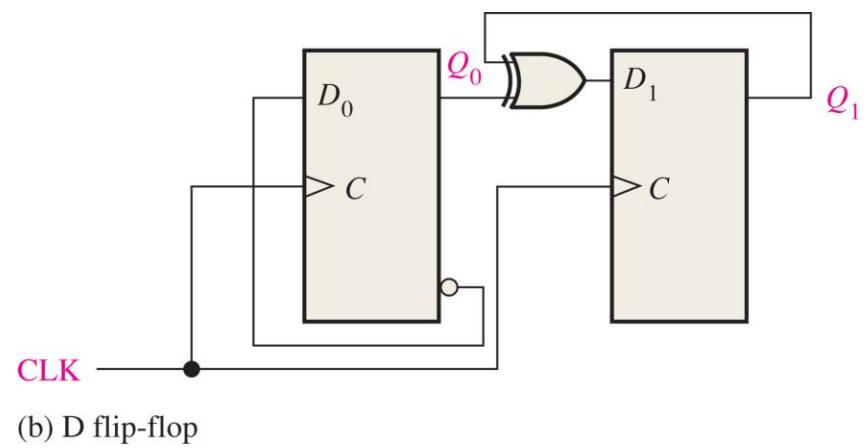
(b) 74HC93 connected as a decade counter



**FIGURE 9-12** 2-bit synchronous binary counters.

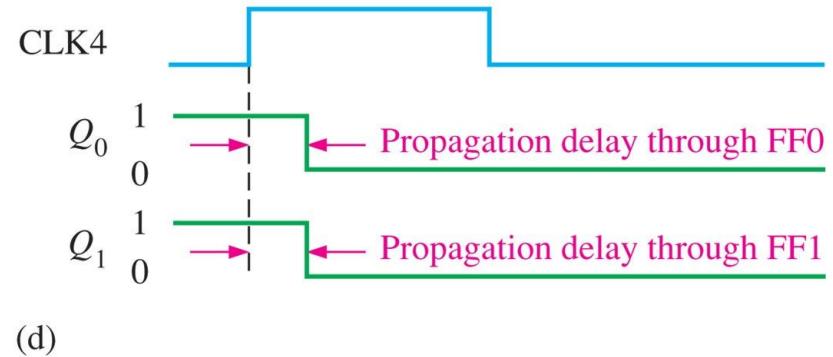
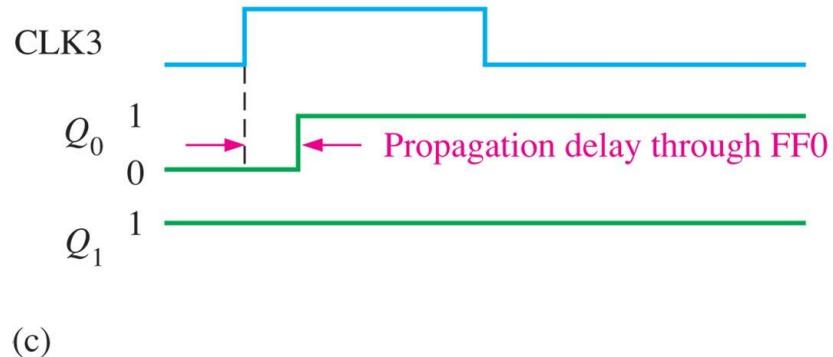
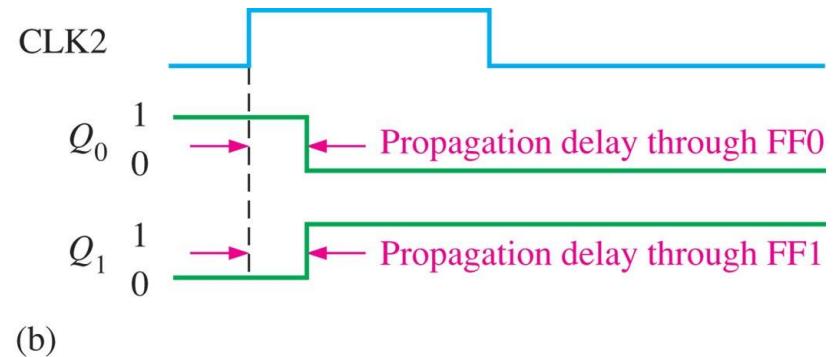
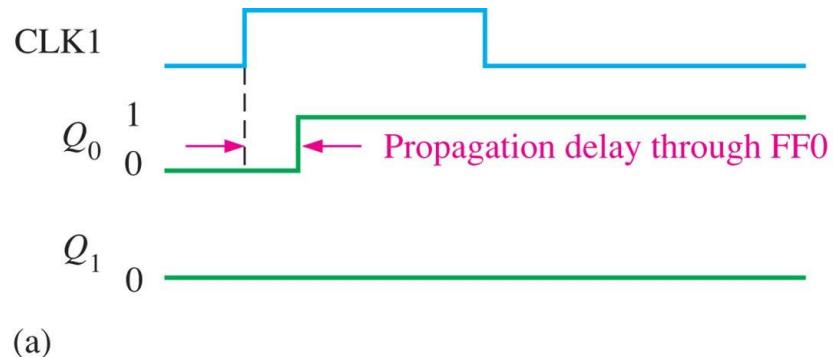


(a) J-K flip-flop

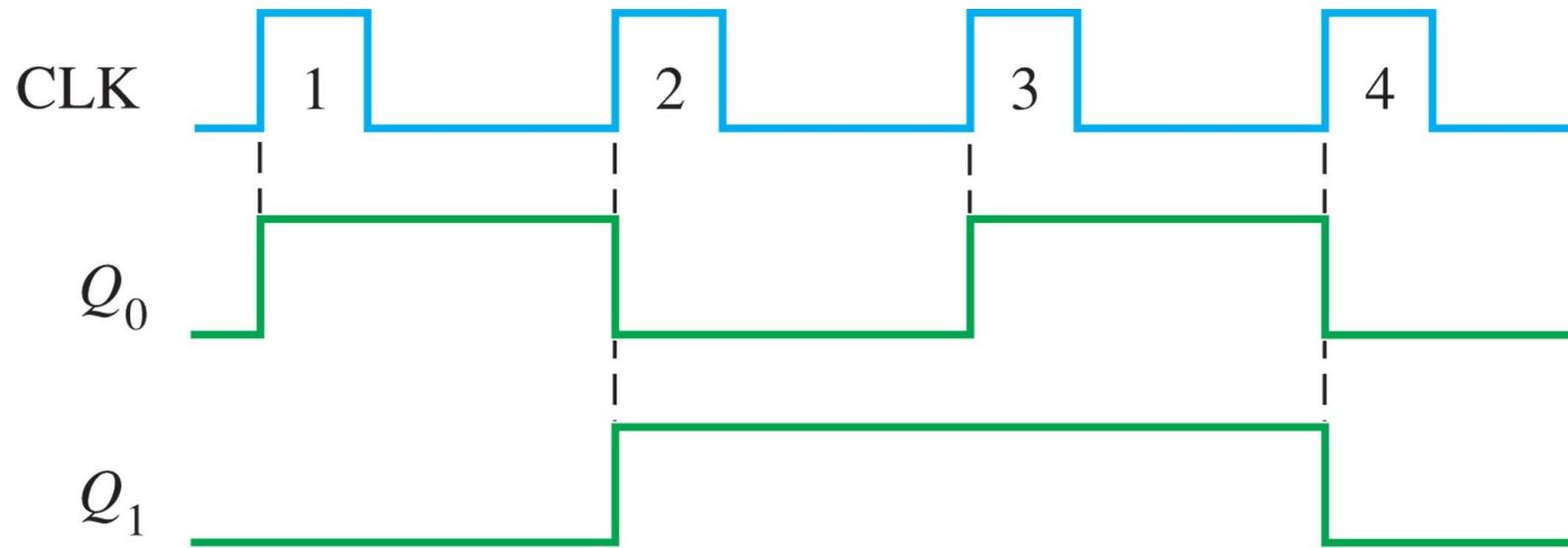


(b) D flip-flop

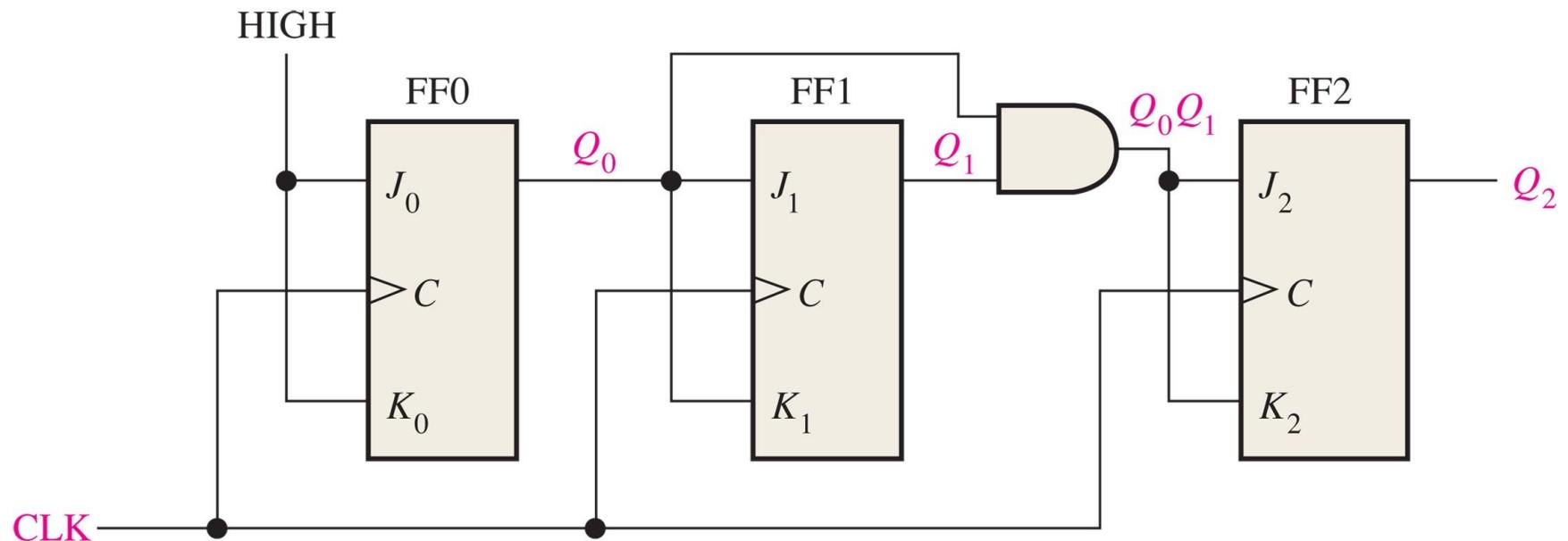
**FIGURE 9-13** Timing details for the 2-bit synchronous counter operation (the propagation delays of both flip-flops are assumed to be equal).



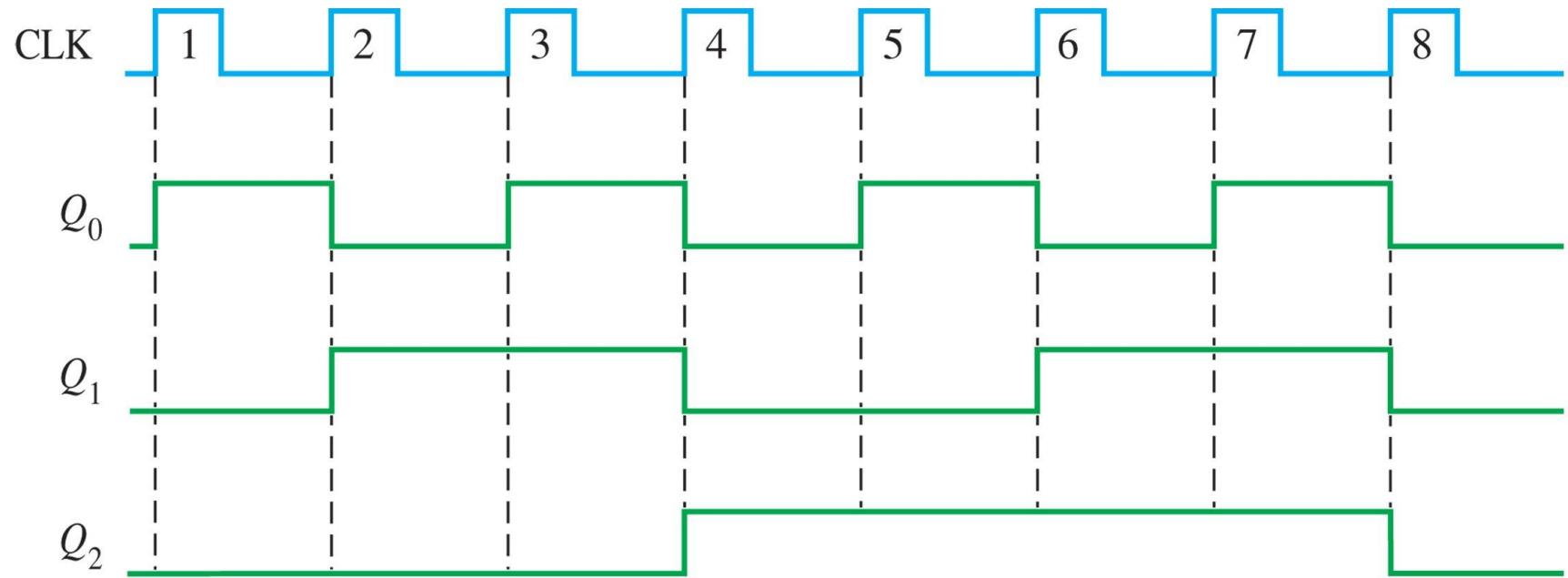
**FIGURE 9-14** Timing diagram for the counters of Figure 9-12.



**FIGURE 9-15** A 3-bit synchronous binary counter.



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



**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

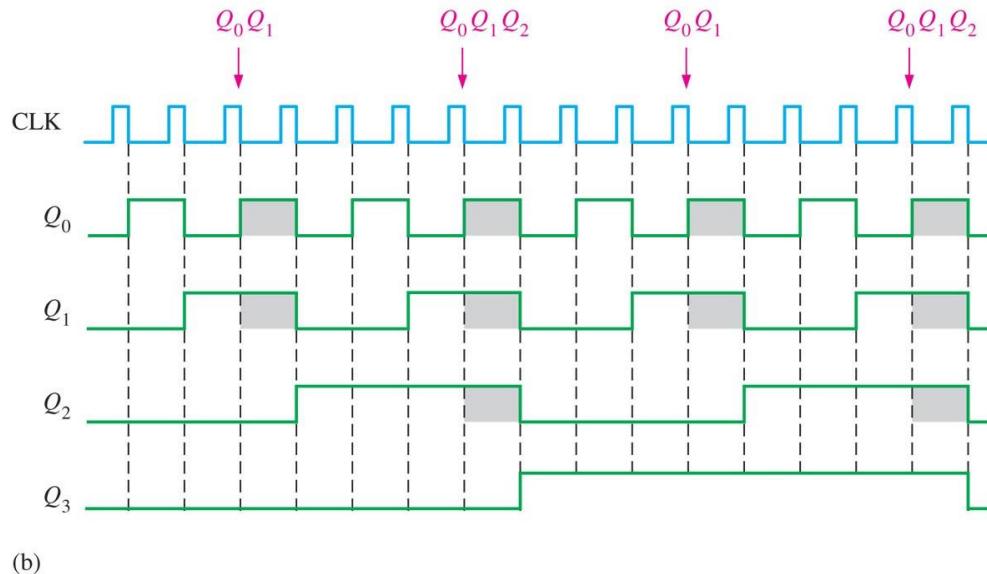
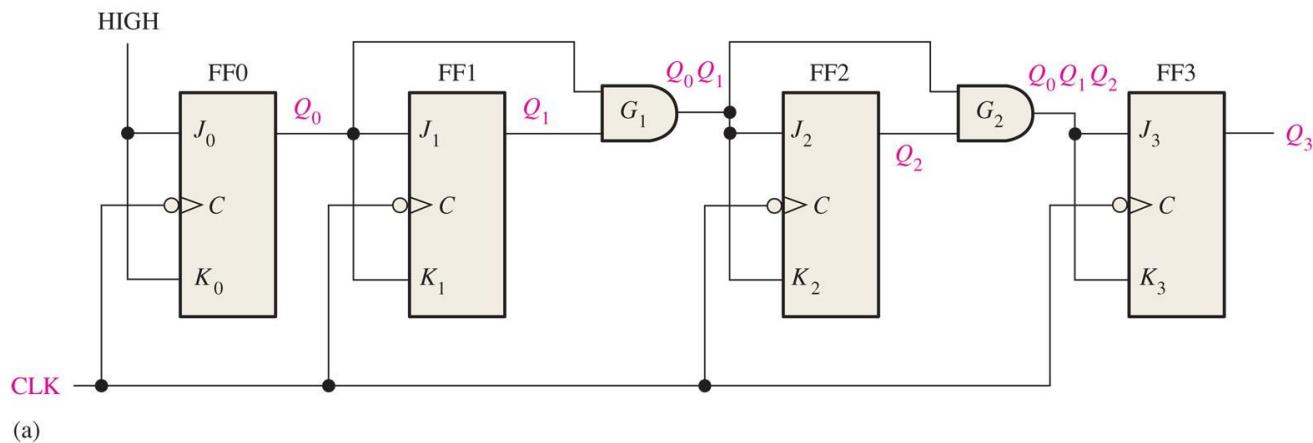
**TABLE 9-4**

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

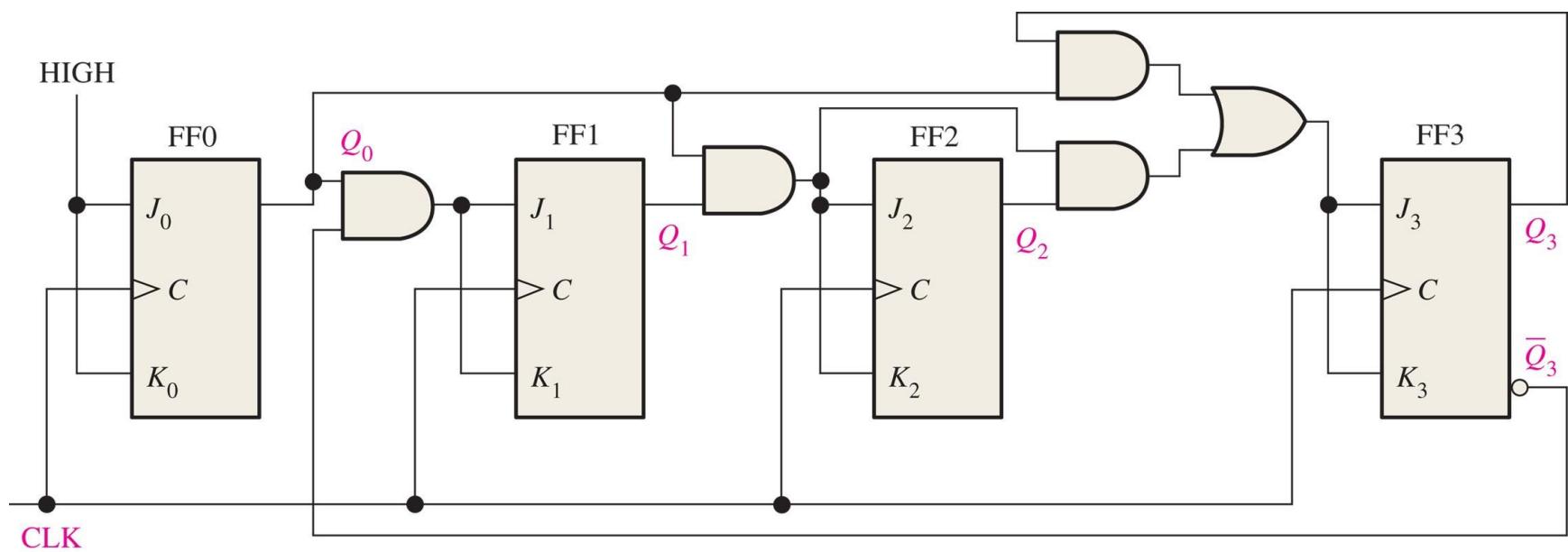
Clock Pulse	Outputs			J-K Inputs					At the Next Clock Pulse			
	$Q_2$	$Q_1$	$Q_0$	$J_2$	$K_2$	$J_1$	$K_1$	$J_0$	$K_0$	FF2	FF1	FF0
Initially	0	0	0	0	0	0	0	1	1	NC*	NC	Toggle
1	0	0	1	0	0	1	1	1	1	NC	Toggle	Toggle
2	0	1	0	0	0	0	0	1	1	NC	NC	Toggle
3	0	1	1	1	1	1	1	1	1	Toggle	Toggle	Toggle
4	1	0	0	0	0	0	0	1	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*.

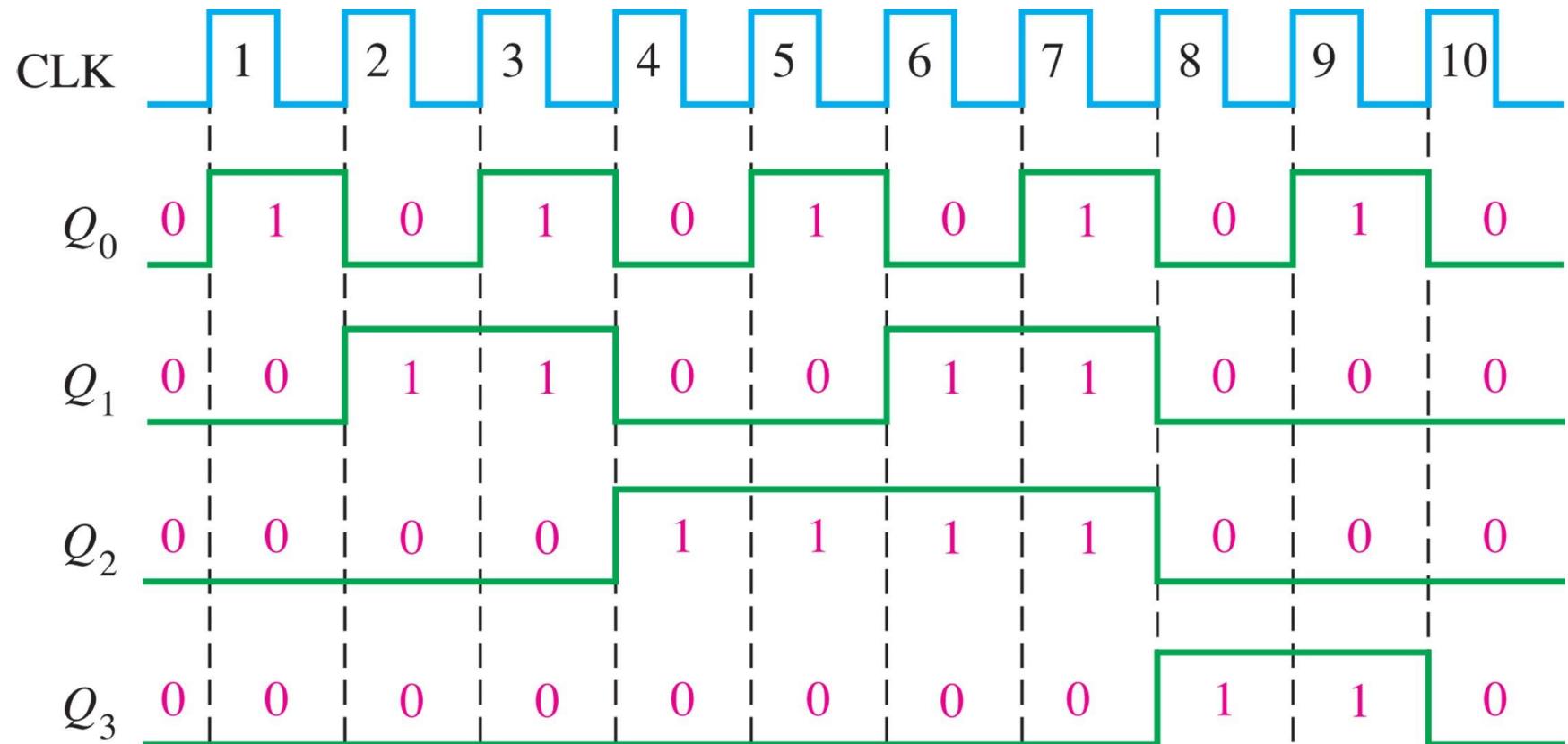
**FIGURE 9-17** A 4-bit synchronous binary counter and timing diagram. Times where the AND gate outputs are HIGH are indicated by the shaded areas.



**FIGURE 9-18** A synchronous BCD decade counter.



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

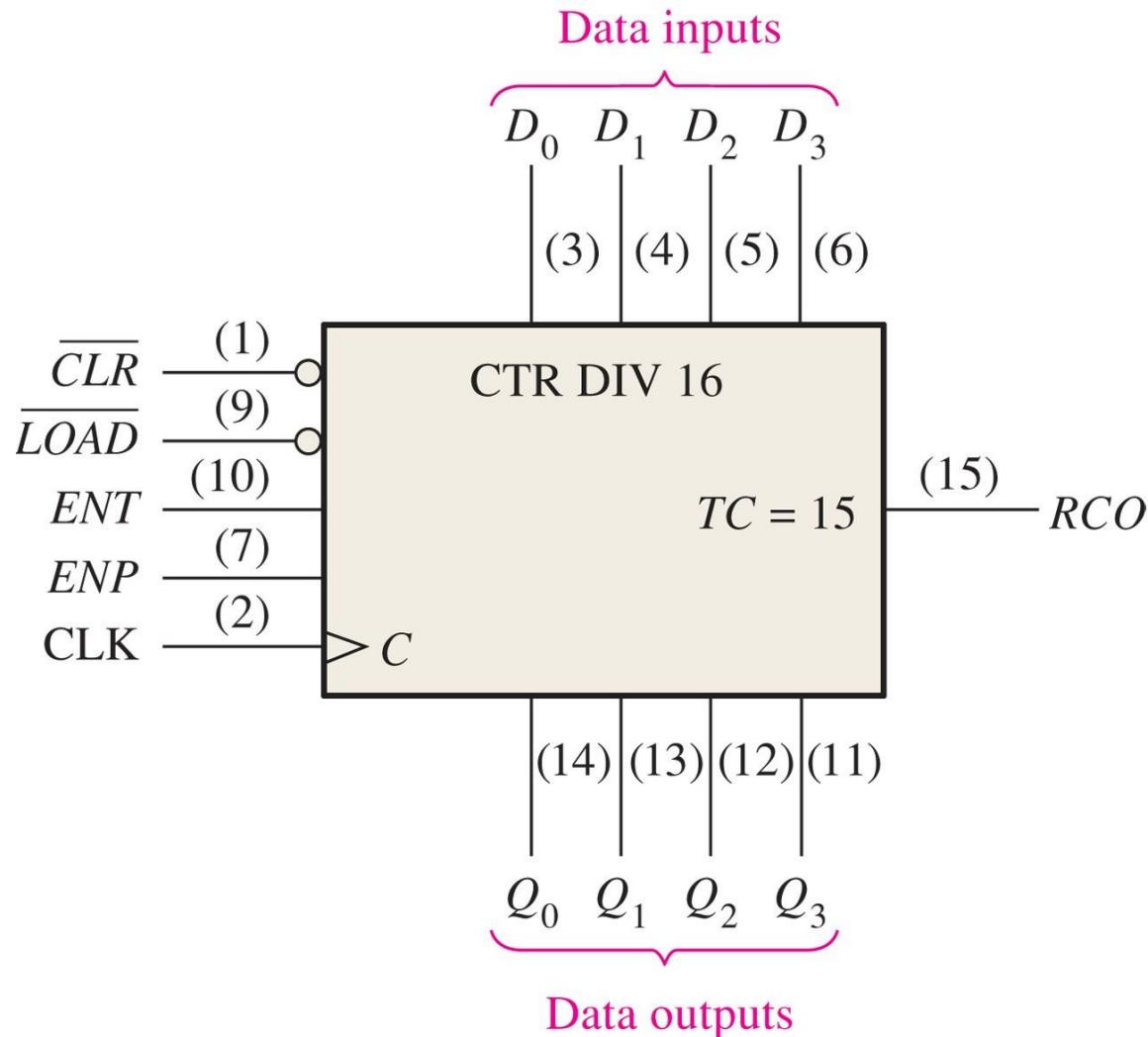


**TABLE 9–5**

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

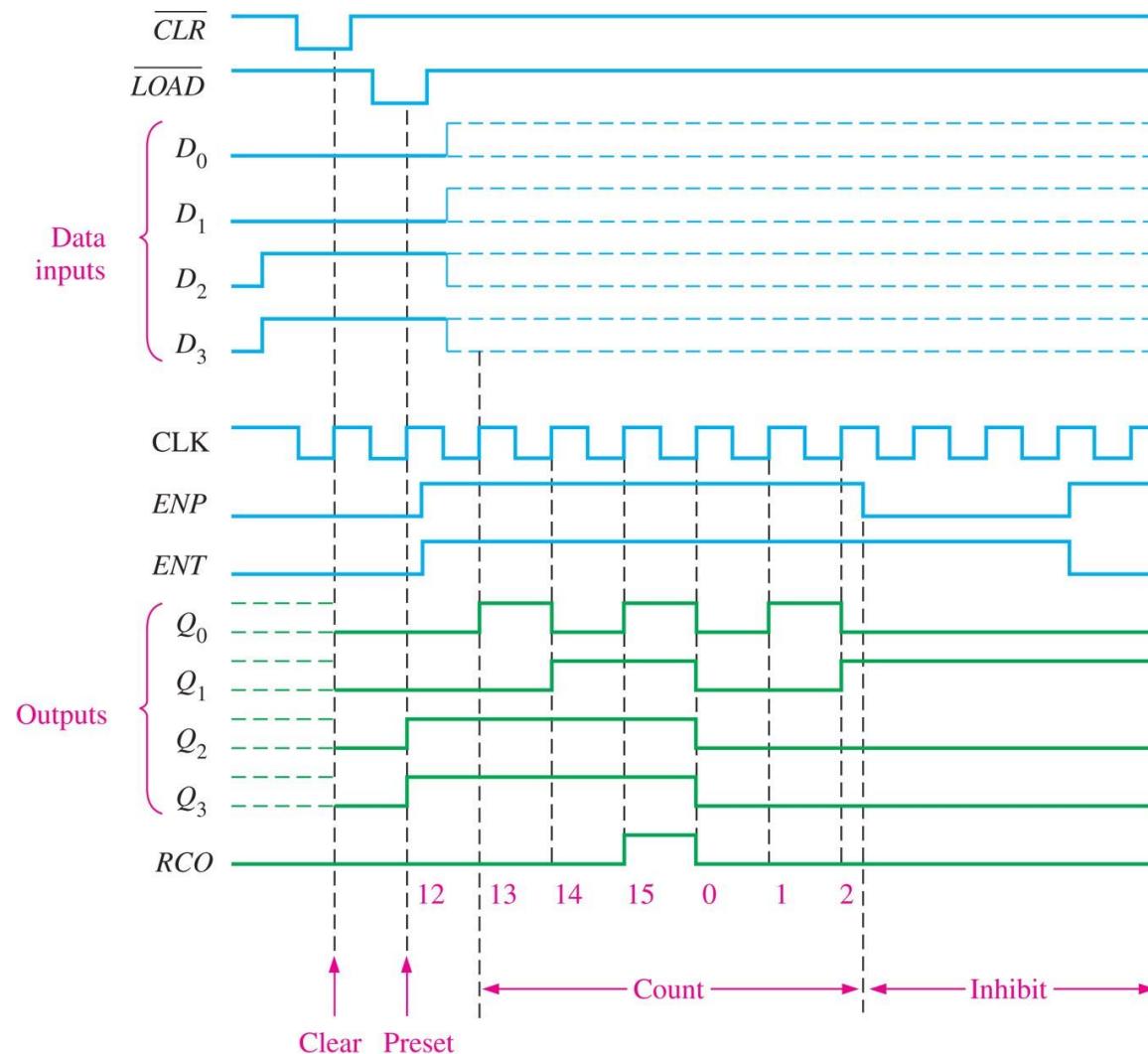
**FIGURE 9-20** The 74HC163 4-bit synchronous binary counter. (The qualifying label CTR DIV 16 indicates a counter with sixteen states.)



First, the counter can be synchronously preset to any 4-bit binary number by applying the proper levels to the parallel data inputs. When a LOW is applied to the LOAD input, the counter will assume the state of the data inputs on the next clock pulse. Thus, the counter sequence can be started with any 4-bit binary number. Also, there is an active-LOW clear input (CLR), which synchronously resets all four flip-flops in the counter. There are two enable inputs, ENP and ENT.

These inputs must both be HIGH for the counter to sequence through its binary states. When at least one input is LOW, the counter is disabled. The ripple clock output (RCO) goes HIGH when the counter reaches the last state in its sequence of fifteen, called the terminal count ( $TC = 15$ ). This output, in conjunction with the enable inputs, allows these counters to be cascaded for higher count sequences.

**FIGURE 9-21** Timing example for a 74HC163.

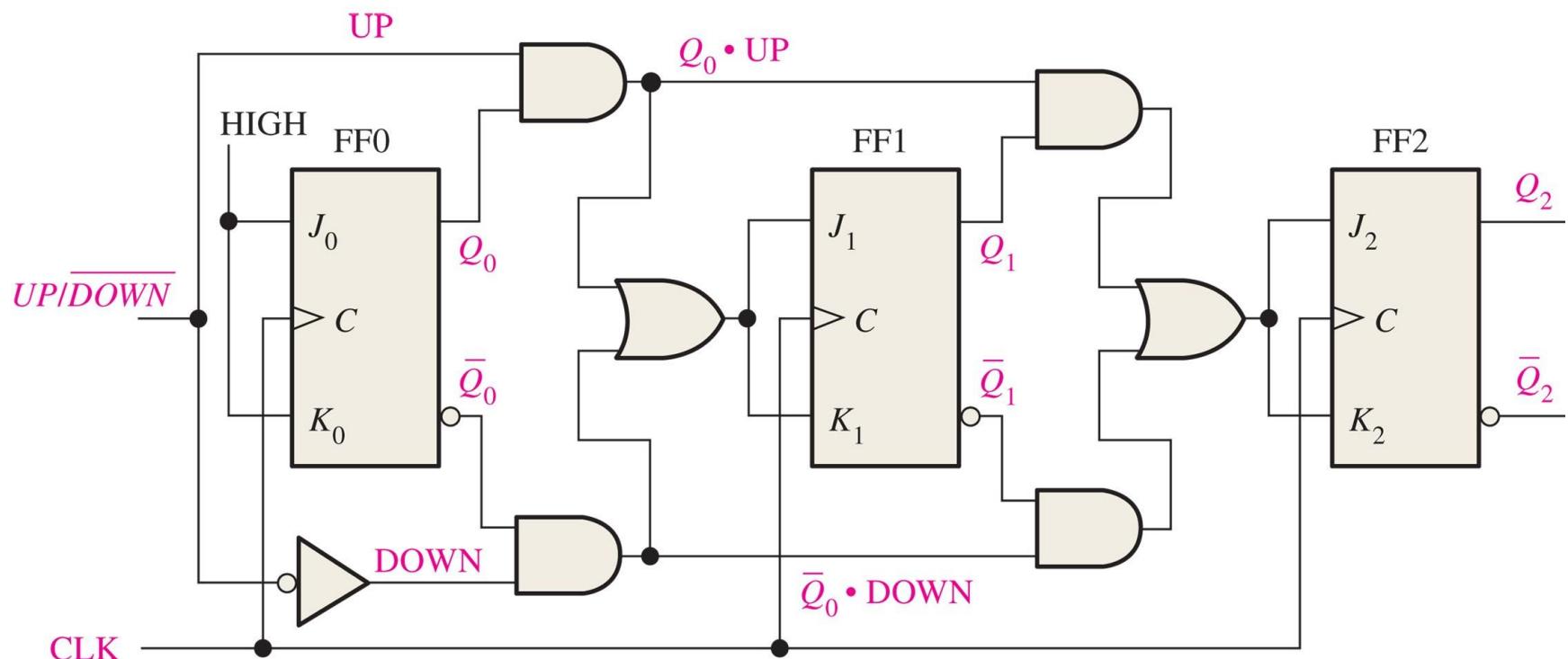


**TABLE 9–6**

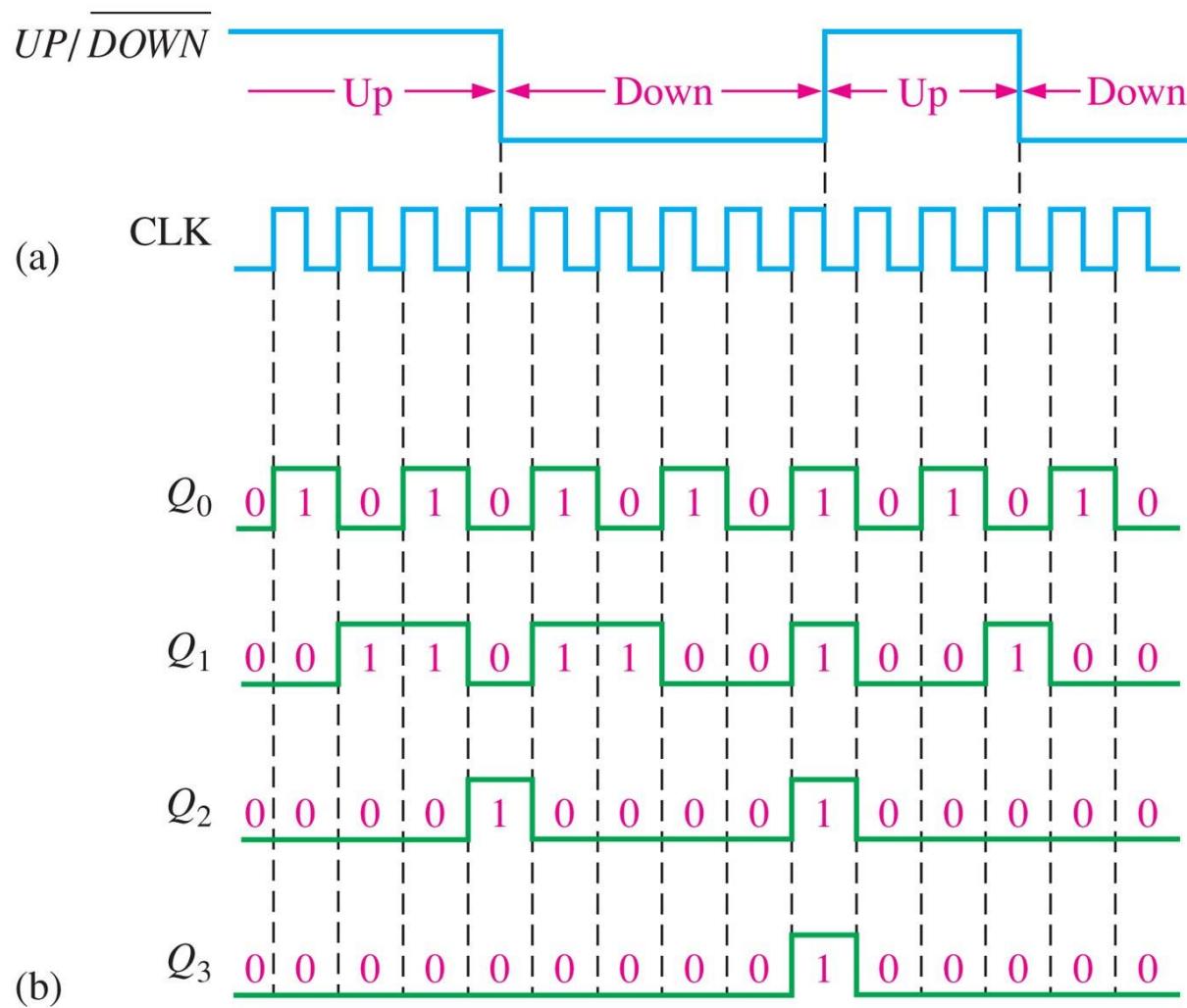
Up/Down sequence for a 3-bit binary counter.

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.



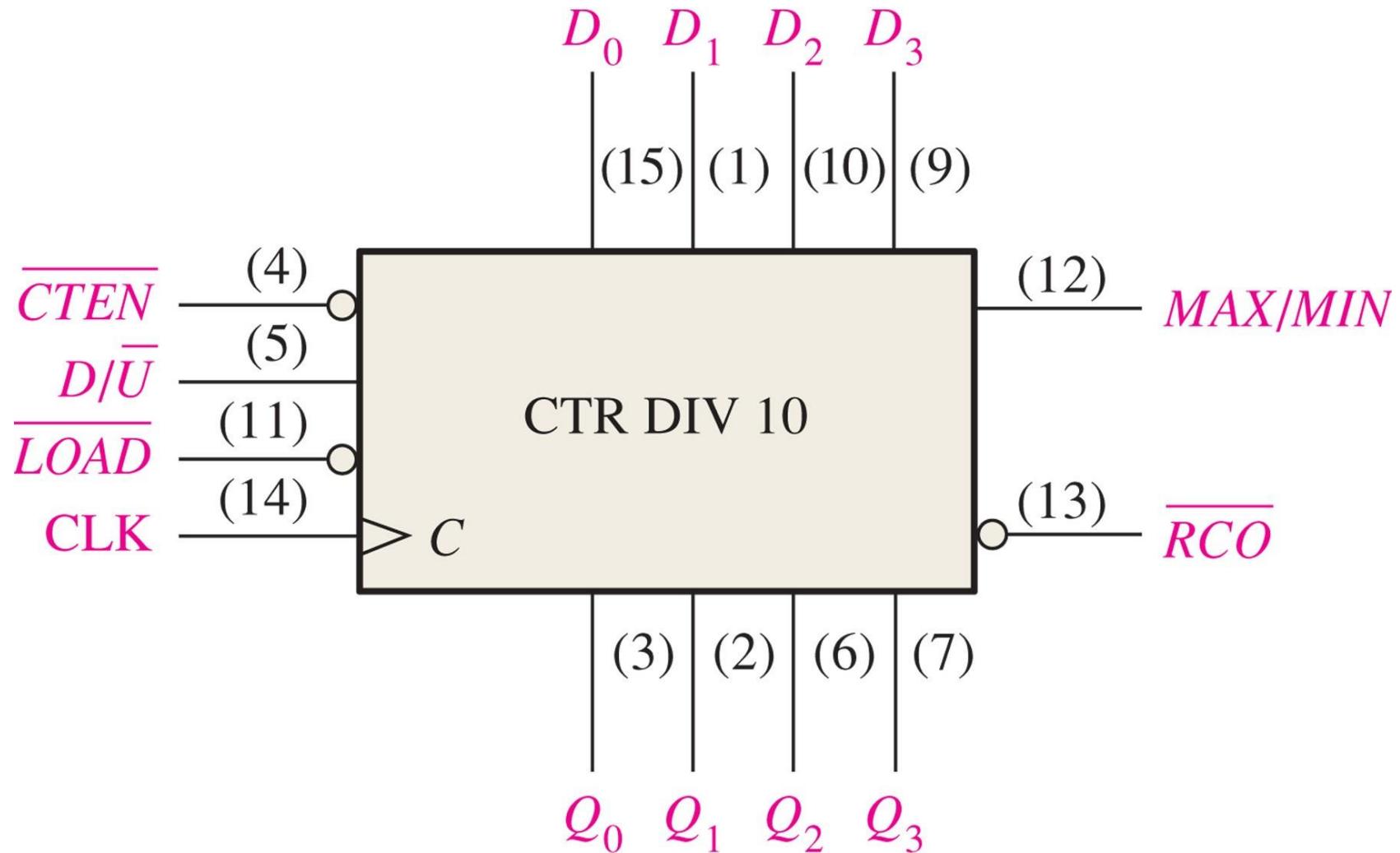
**FIGURE 9-23**



**TABLE 9–7**

$Q_3$	$Q_2$	$Q_1$	$Q_0$	
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	0	1	1	
0	0	1	0	
0	0	0	1	
0	0	0	0	
1	1	1	1	
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	0	1	
0	0	0	0	

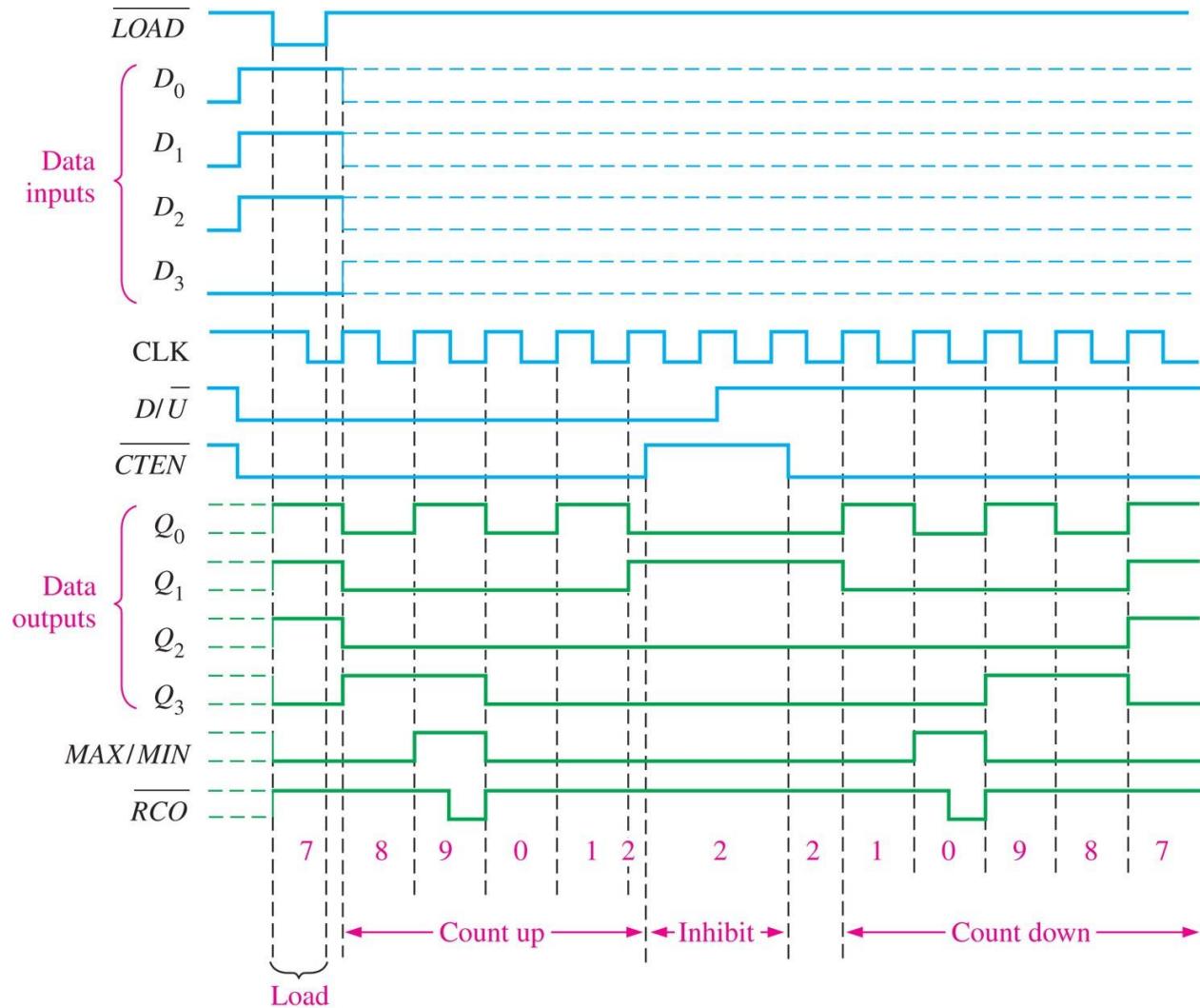
**FIGURE 9-24** The 74HC190 up/down synchronous decade counter.



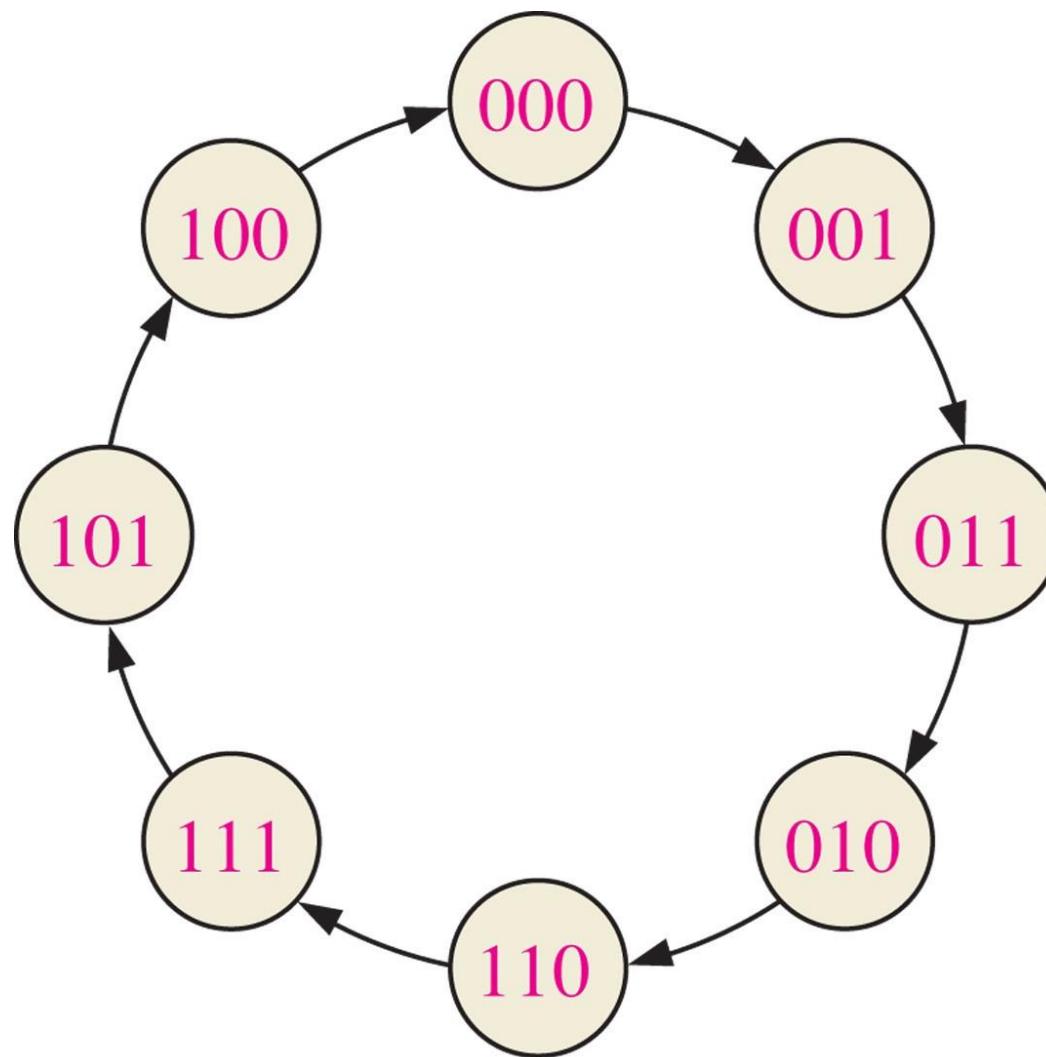
**Fixed-Function Device** Figure 9–24 shows a logic diagram for the 74HC190, an example of an integrated circuit up/down synchronous decade counter. The direction of the count is determined by the level of the up/down input (D/U). When this input is HIGH, the counter counts down; when it is LOW, the counter counts up. Also, this device can be preset to any desired BCD digit as determined by the states of the data inputs when the LOAD input is LOW.

The MAX/MIN output produces a HIGH pulse when the terminal count nine (1001) is reached in the UP mode or when the terminal count zero (0000) is reached in the DOWN mode. The MAX/MIN output, the ripple clock output (RCO), and the count enable input (CTEN) are used when cascading counters.

**FIGURE 9-25** Timing example for a 74HC190.



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



**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

**TABLE 9-9**

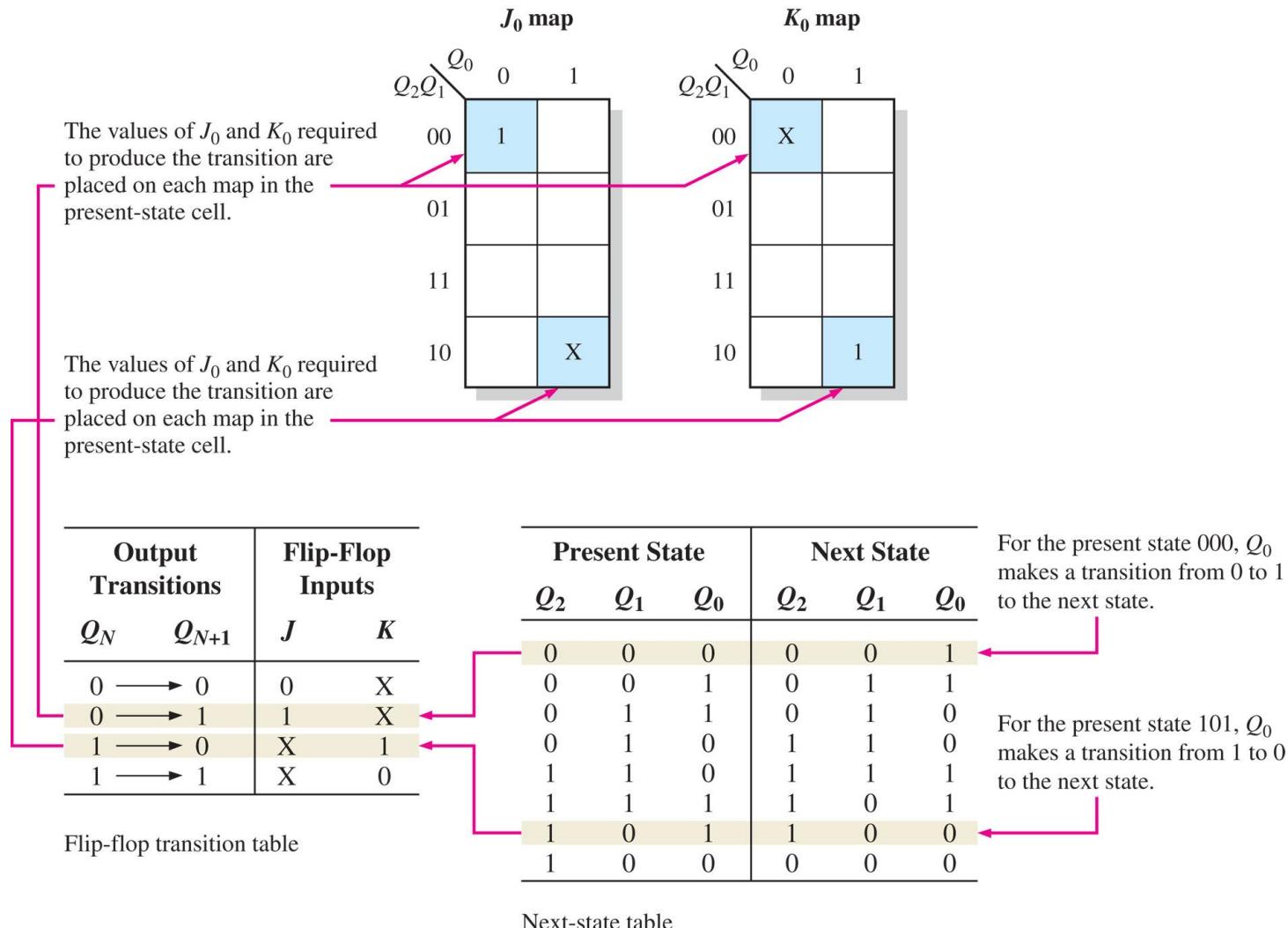
Transition table for a J-K flip-flop.

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

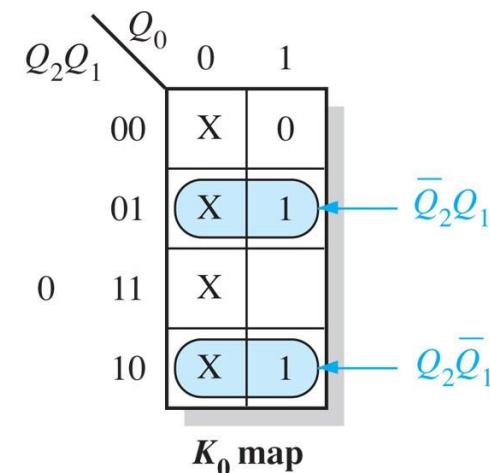
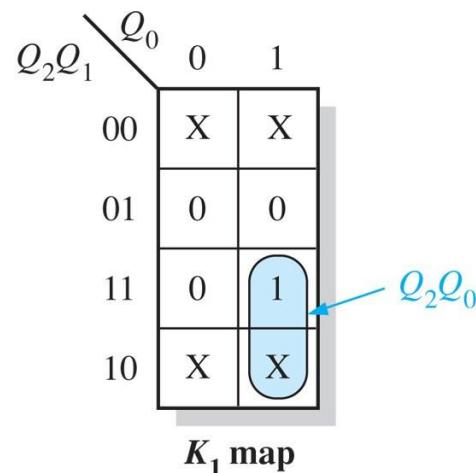
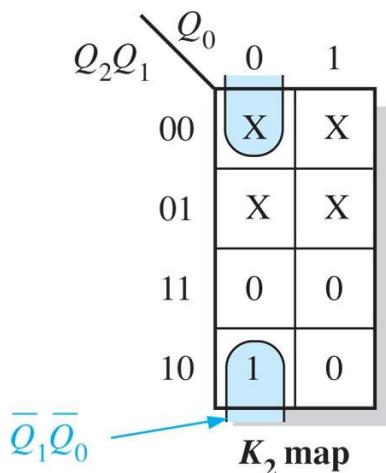
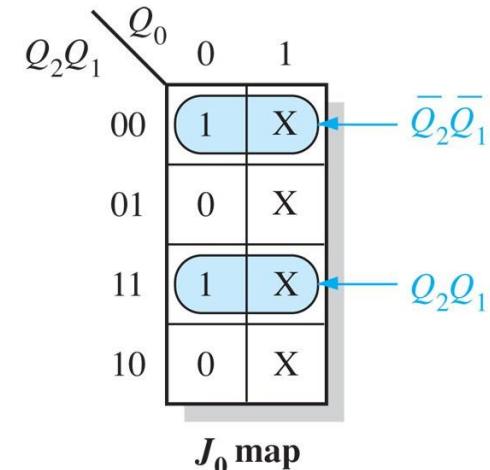
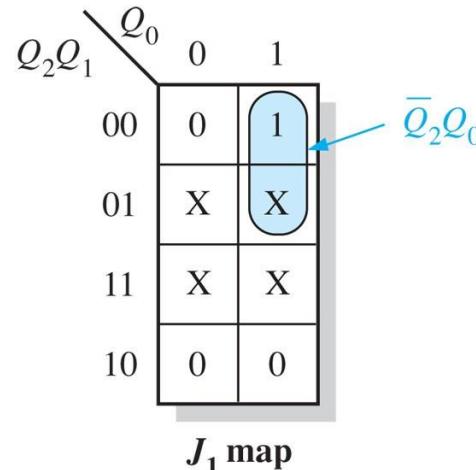
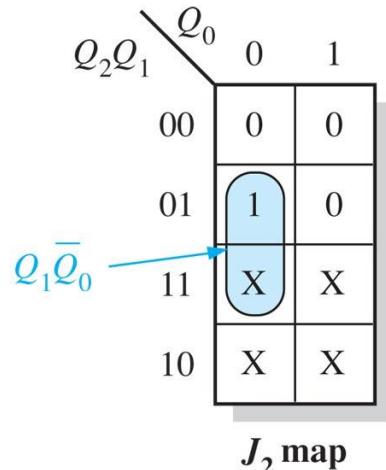
 $Q_N$ : present state $Q_{N+1}$ : next state

X: “don’t care”

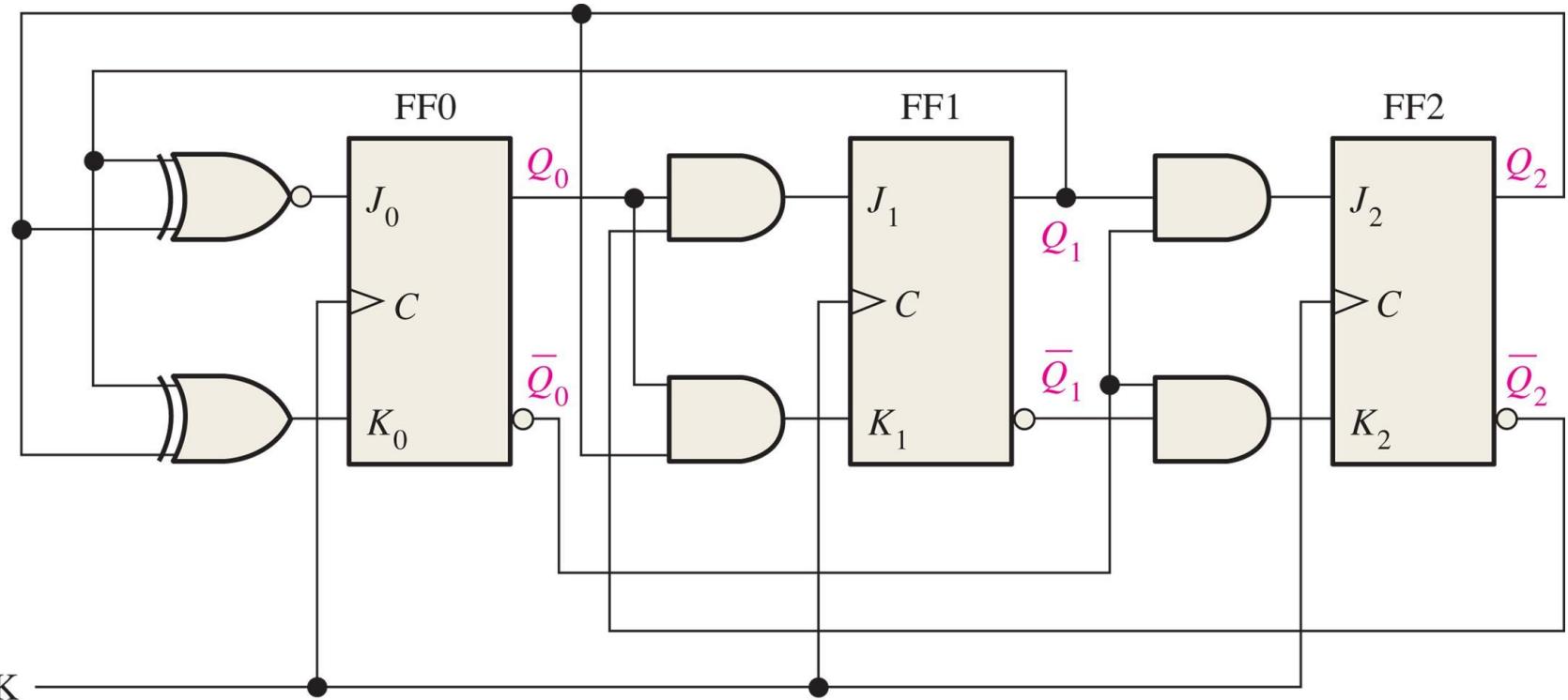
**FIGURE 9-27** Examples of the mapping procedure for the counter sequence represented in Table 9-8 and Table 9-9.



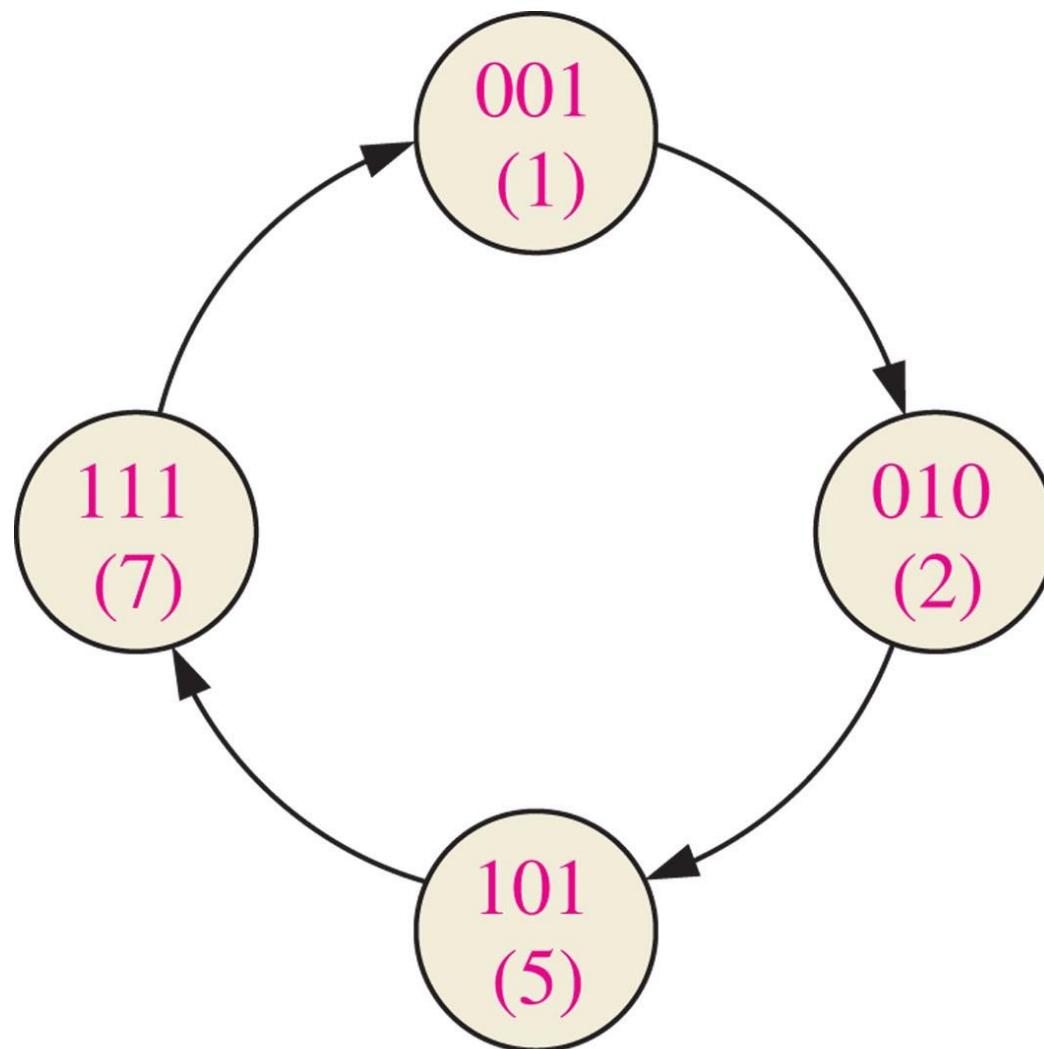
**FIGURE 9-28** Karnaugh maps for present-state  $J$  and  $K$  inputs.



**FIGURE 9-29** Three-bit Gray code counter.



**FIGURE 9-30**



**TABLE 9–10**

Next-state table.

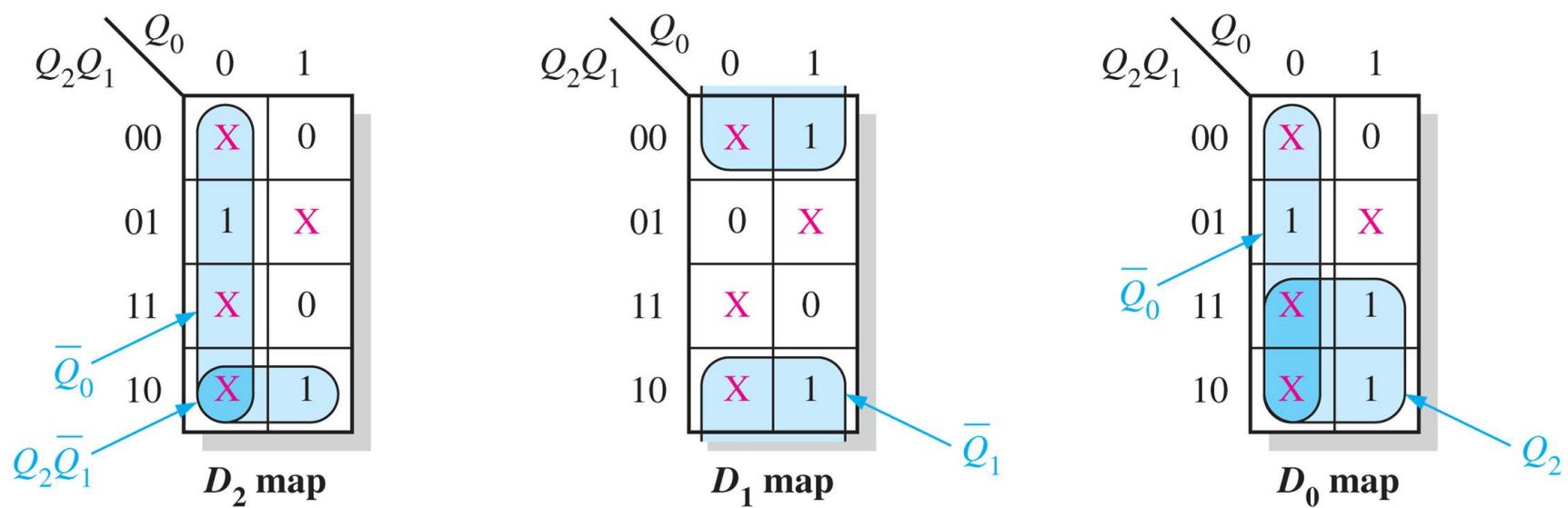
Present State			Next State		
$Q_2$	$Q_1$	$Q_0$	$Q_2$	$Q_1$	$Q_0$
0	0	1	0	1	0
0	1	0	1	0	1
1	0	1	1	1	1
1	1	1	0	0	1

**TABLE 9–11**

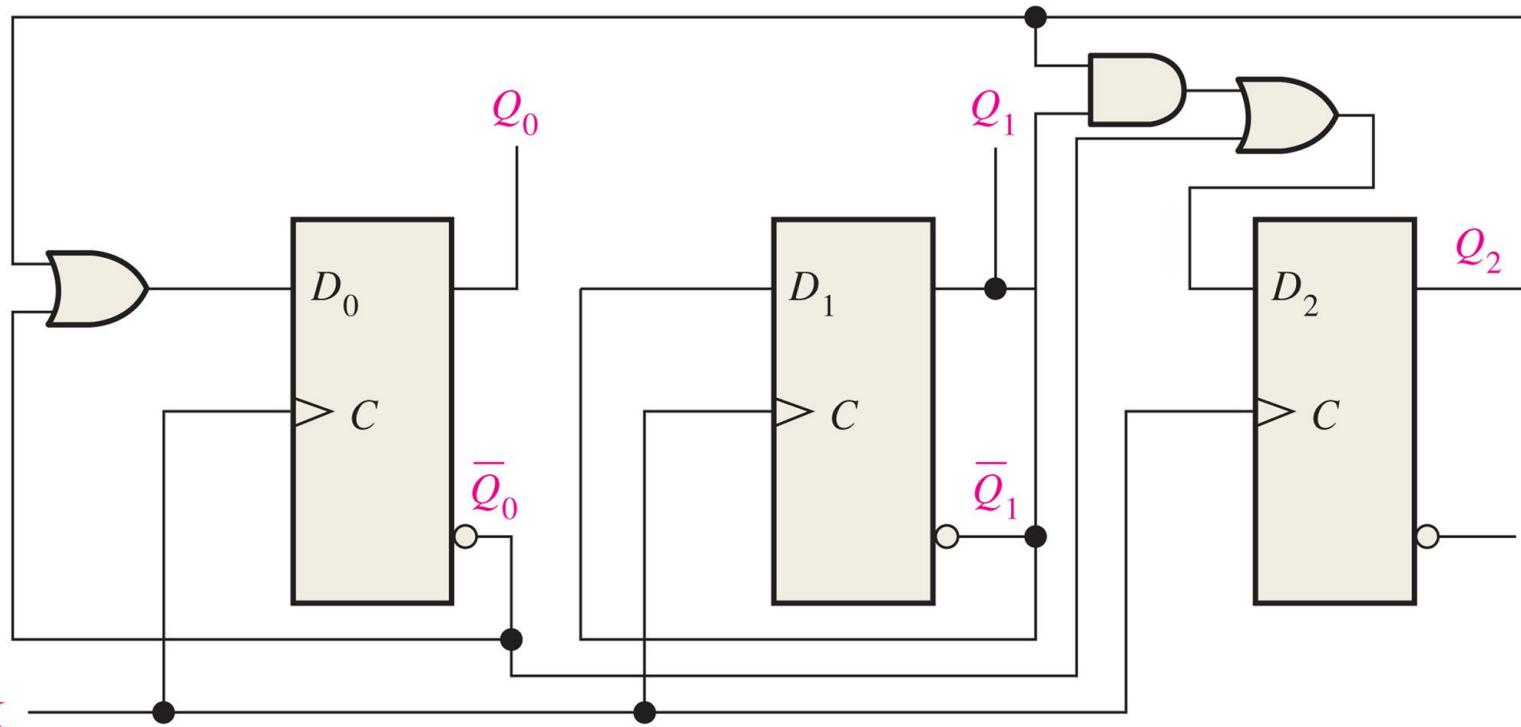
Transition table for a D flip-flop.

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

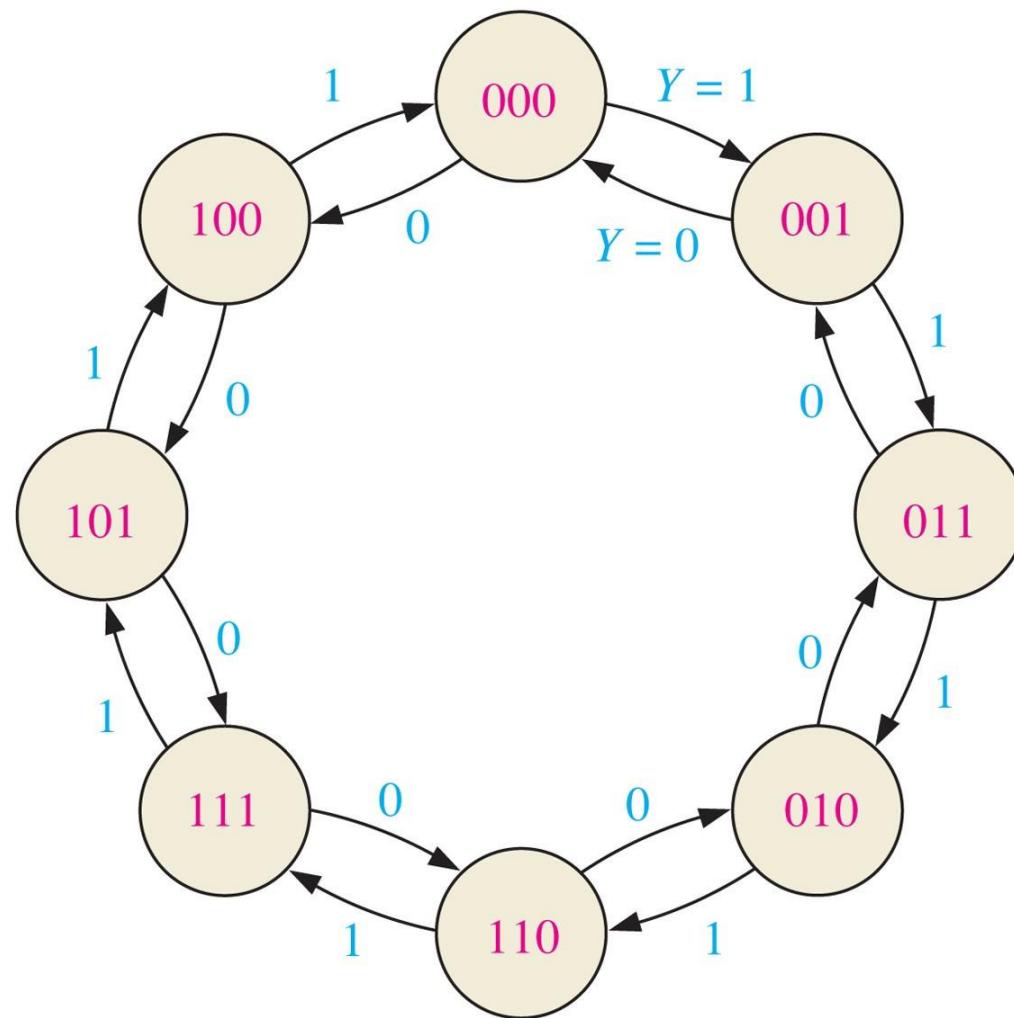
**FIGURE 9-31**



**FIGURE 9-32**



**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			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

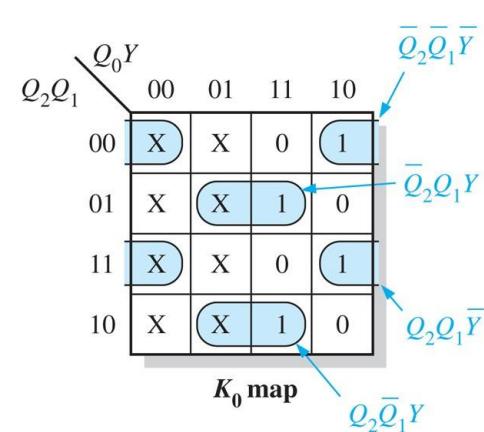
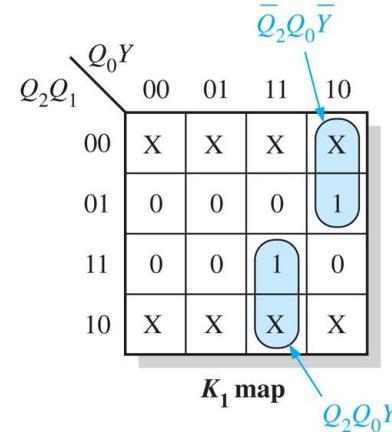
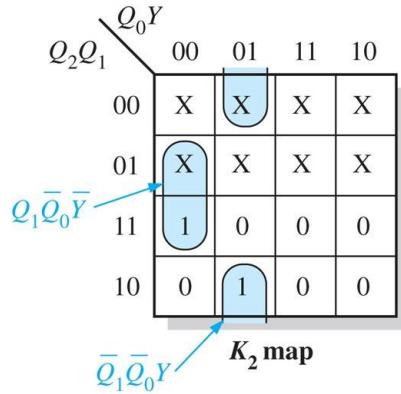
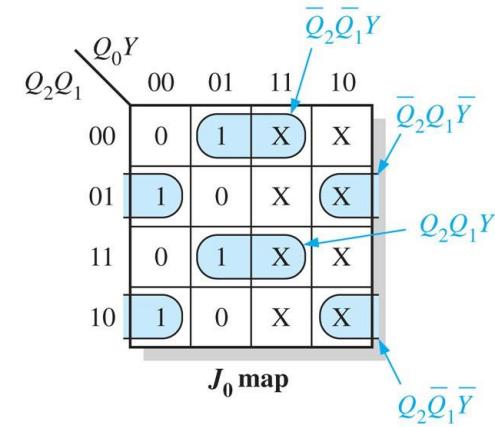
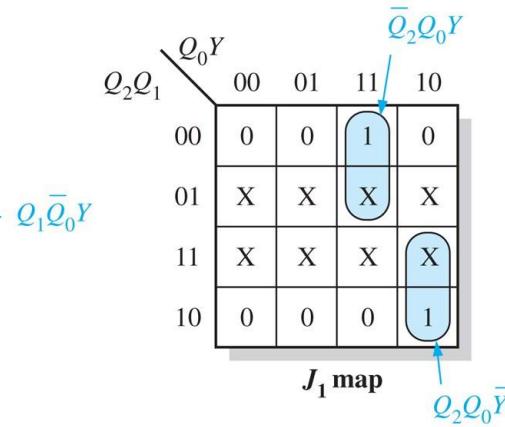
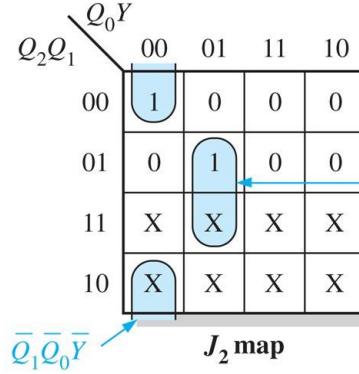
 $Y = \text{UP}/\overline{\text{DOWN}}$  control input.

**TABLE 9–13**

Transition table for a J-K flip-flop.

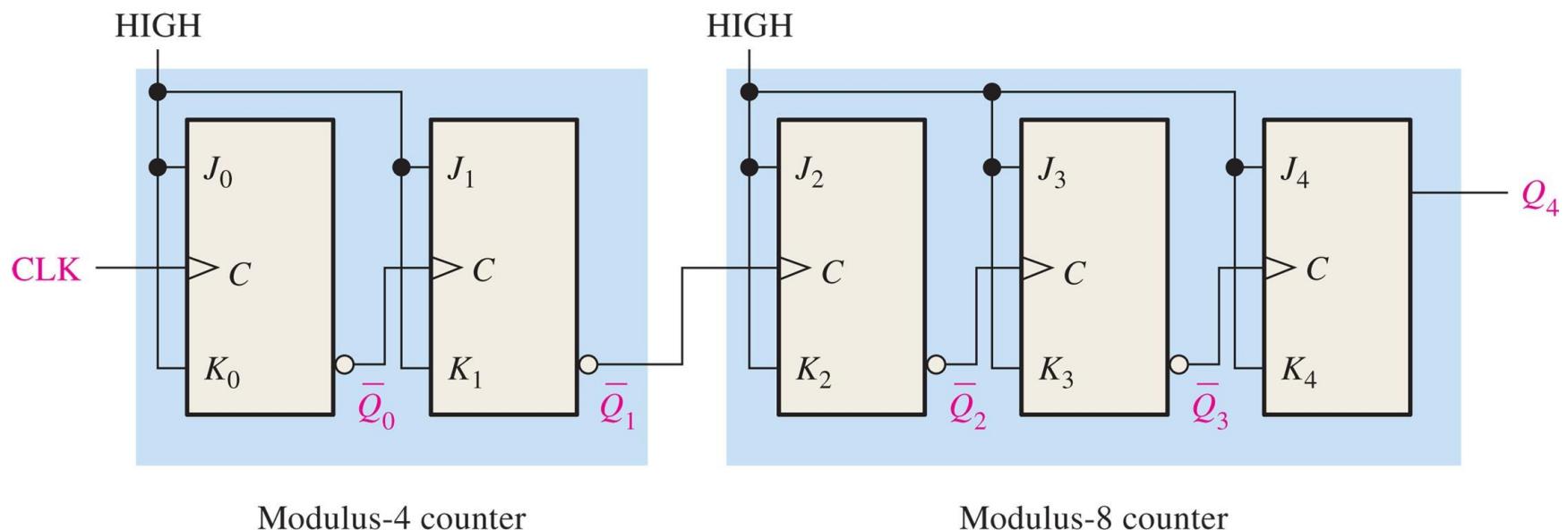
Output Transitions		Flip-Flop Inputs	
$Q_N$	$Q_{N + 1}$	$J$	$K$
0	→ 0	0	X
0	→ 1	1	X
1	→ 0	X	1
1	→ 1	X	0

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

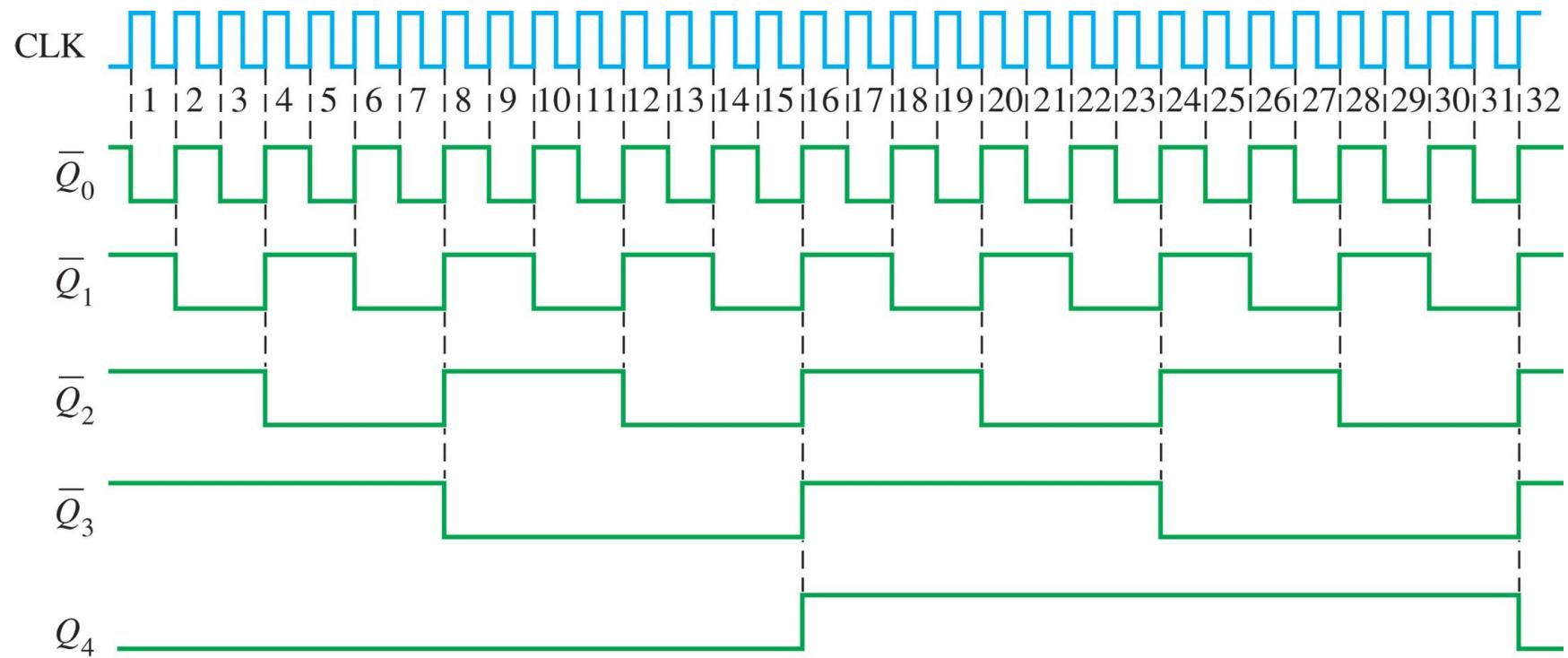


# Asynchronous Cascading

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

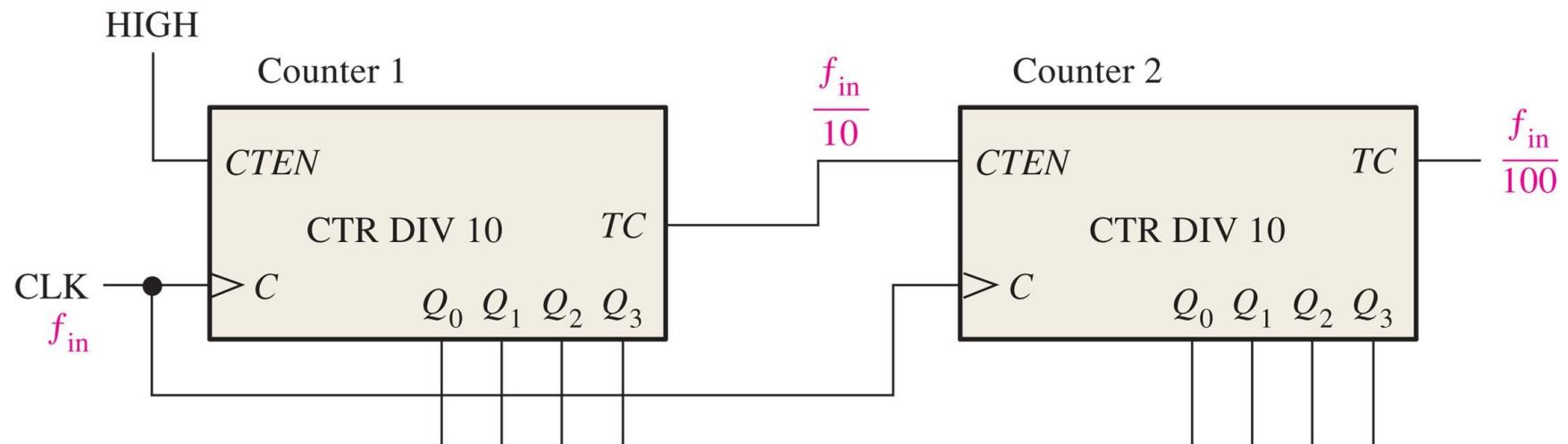


**FIGURE 9-36** Timing diagram for the cascaded counter configuration of Figure 9-35.

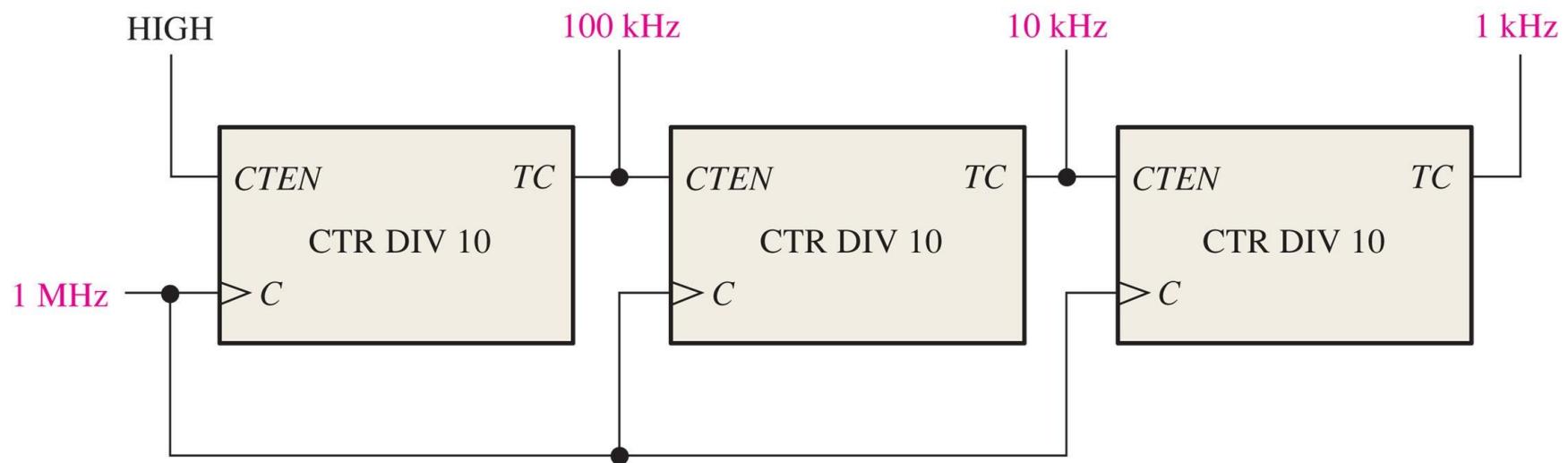


# Synchronous Cascading

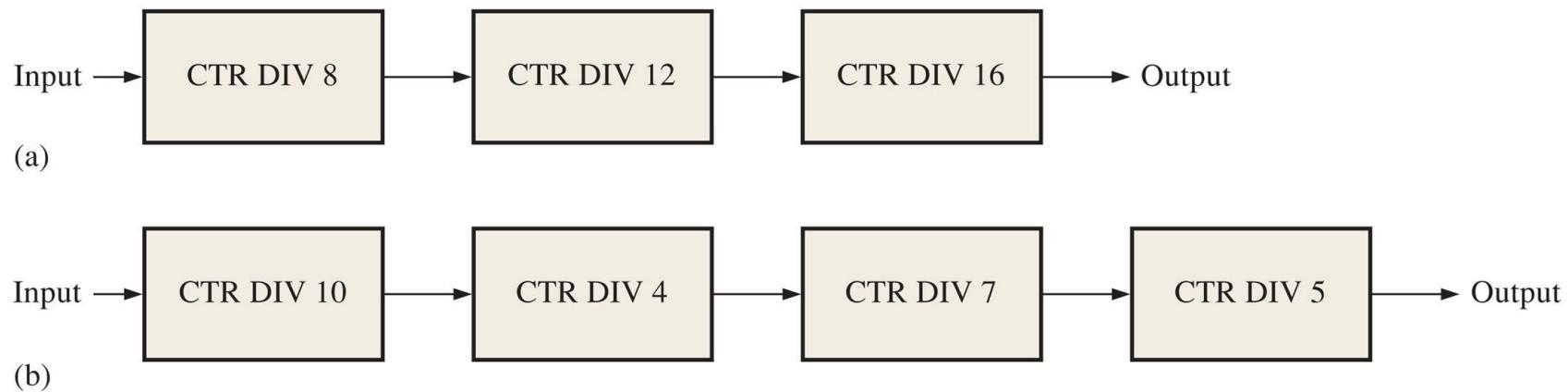
**FIGURE 9-37** A modulus-100 counter using two cascaded decade counters.



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



**FIGURE 9-39**



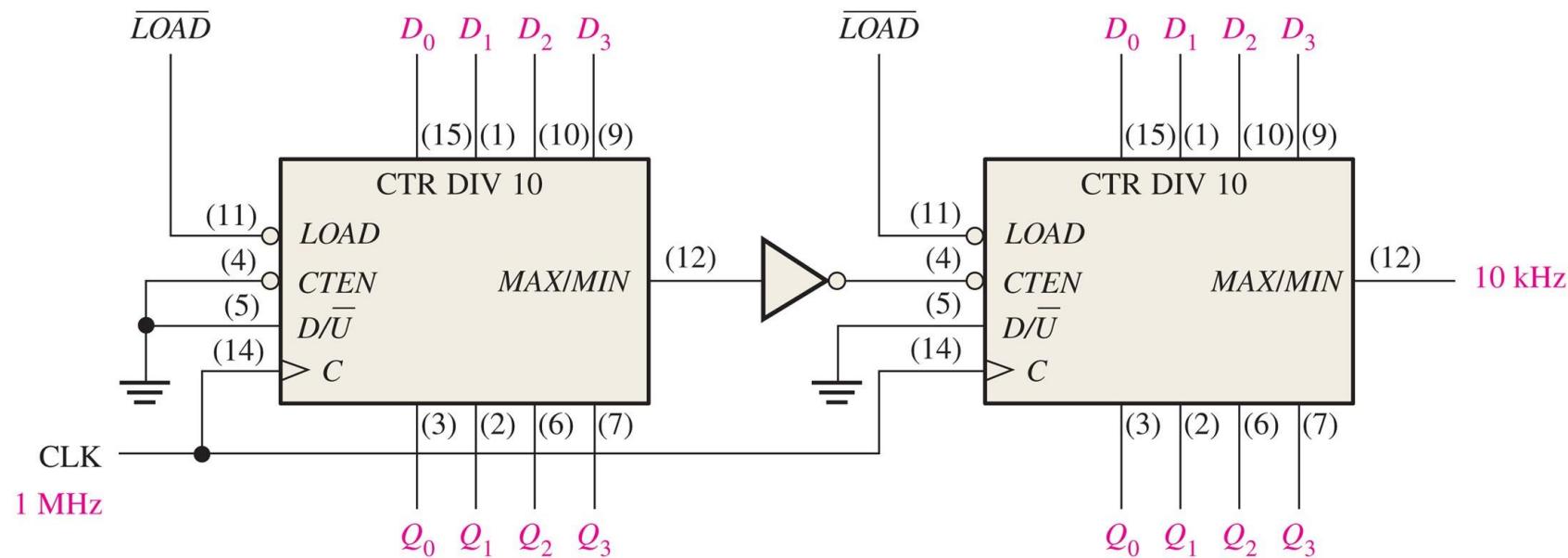
**Solution** In Figure 9–39(a), the overall modulus for the 3-counter configuration is  $8 * 12 * 16 = 1536$

In Figure 9–39(b), the overall modulus for the 4-counter configuration is  $10 * 4 * 7 * 5 = 1400$

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

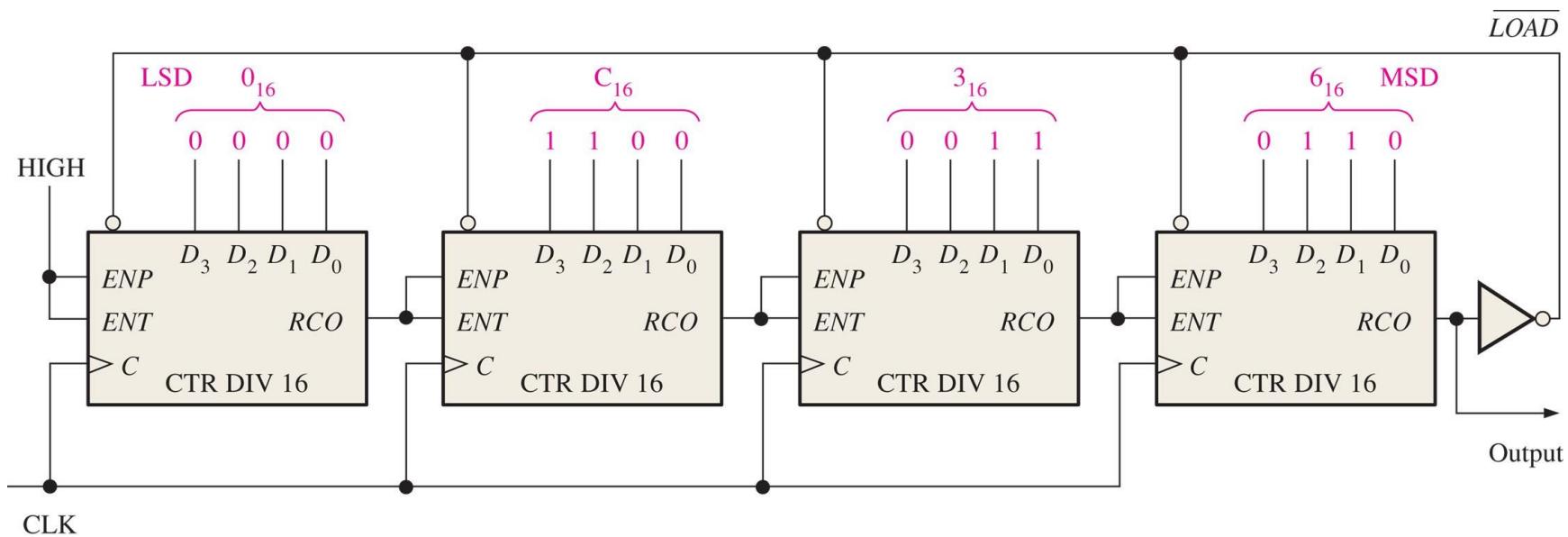
**Solution** To obtain 10 kHz from a 1 MHz clock requires a division factor of 100. Two 74HC190 counters must be cascaded as shown in Figure 9–40.

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. D3 D2 D1



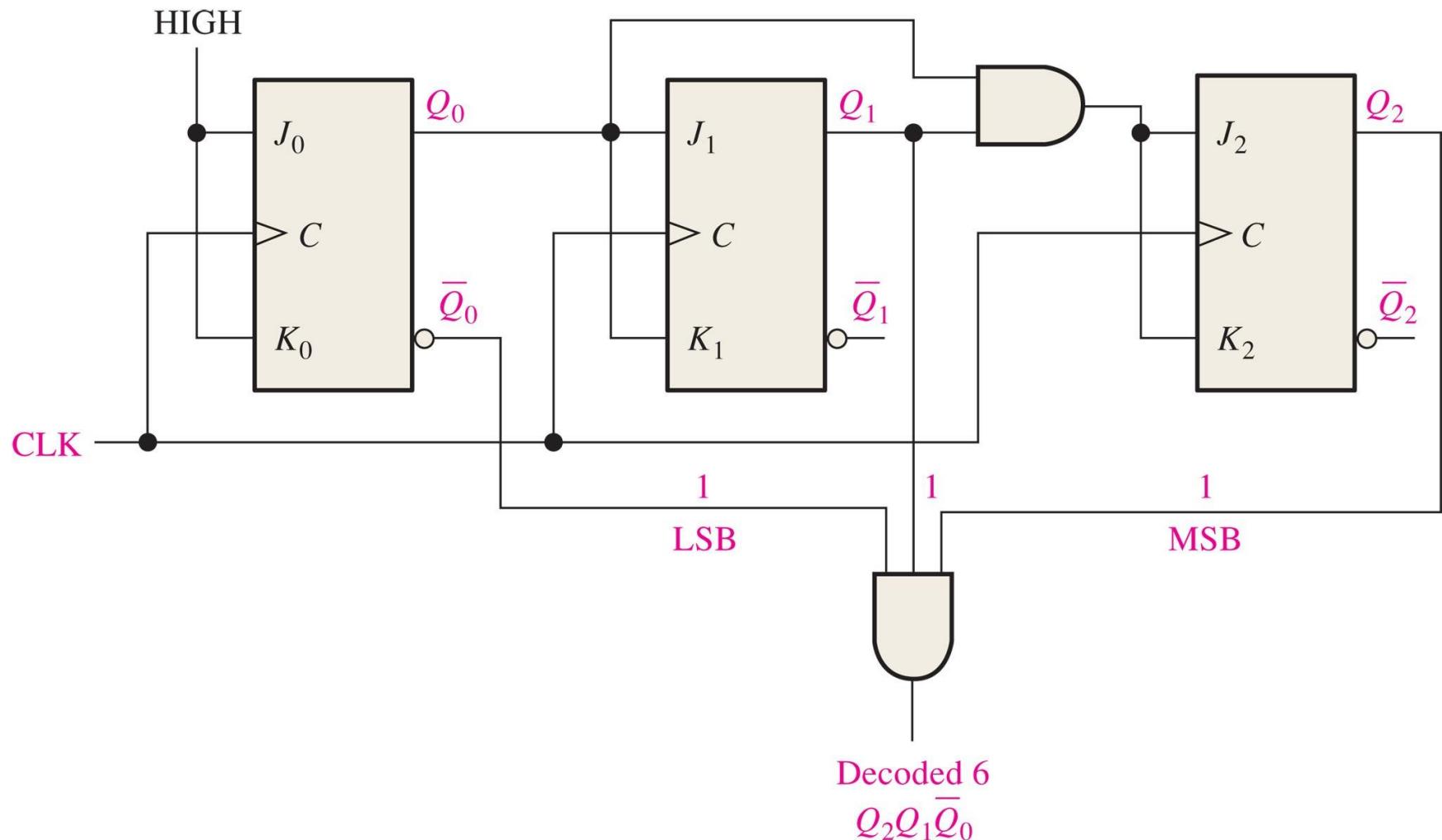
**FIGURE 9-41** A divide-by-40,000 counter using 74HC161 4-bit binary counters. Note that each of the parallel data inputs is shown in binary order (the right-most bit  $D_0$  is the LSB in each counter).

$$2^{16} = 65,536$$

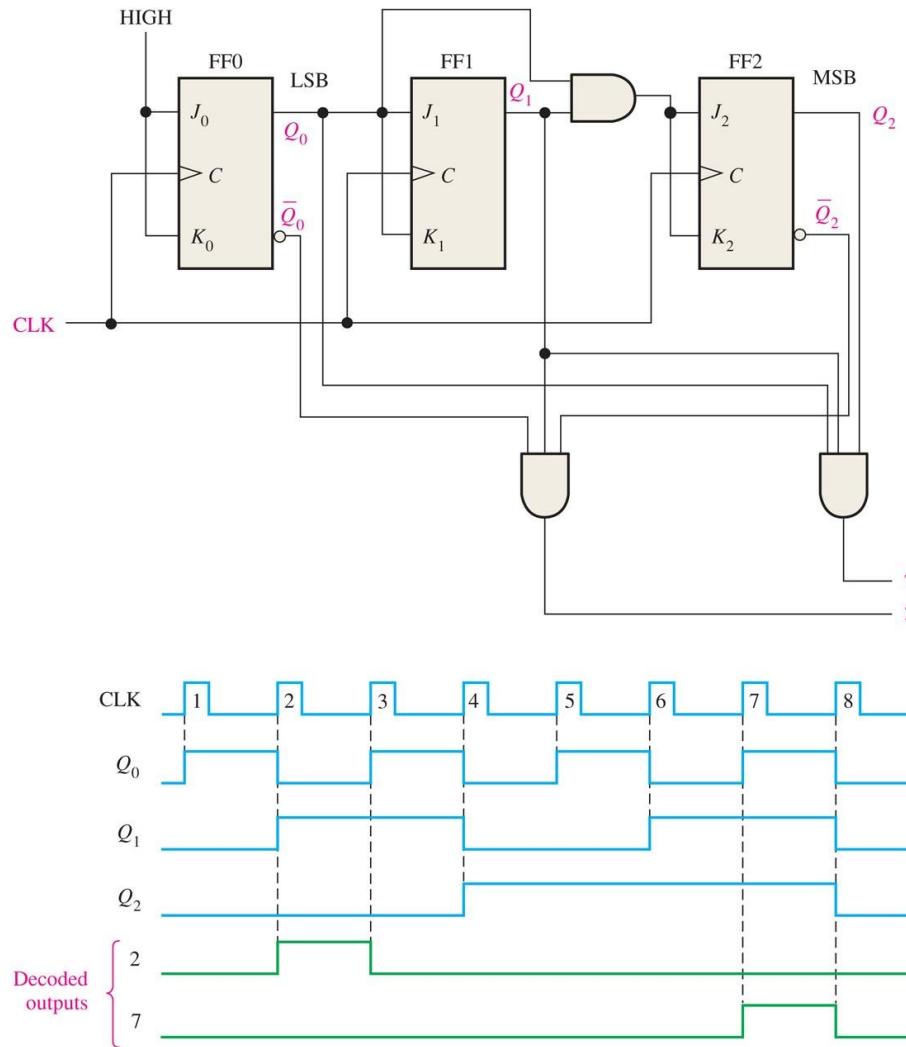


Let's assume that a certain application requires a divide-by-40,000 counter (modulus 40,000). The difference between 65,536 and 40,000 is 25,536, which is the number of states that must be deleted from the full-modulus sequence. The technique used in the circuit of Figure 9–41 is to preset the cascaded counter to 25,536 (63C0 in hexadecimal) each time it recycles, so that it will count from 25,536 up to 65,535 on each full cycle. Therefore, each full cycle of the counter consists of 40,000 states.

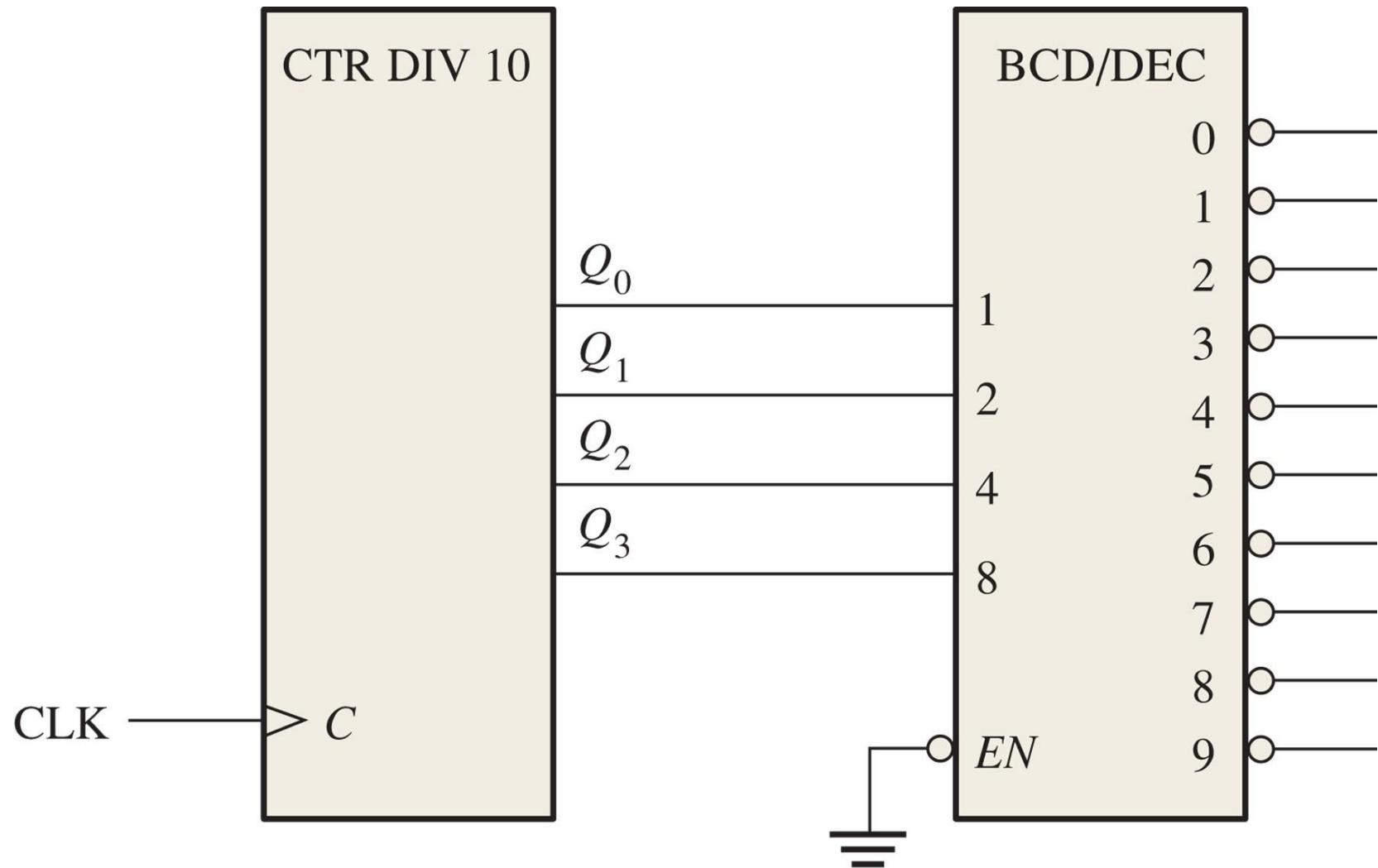
**FIGURE 9-42** Decoding of state 6 (110).



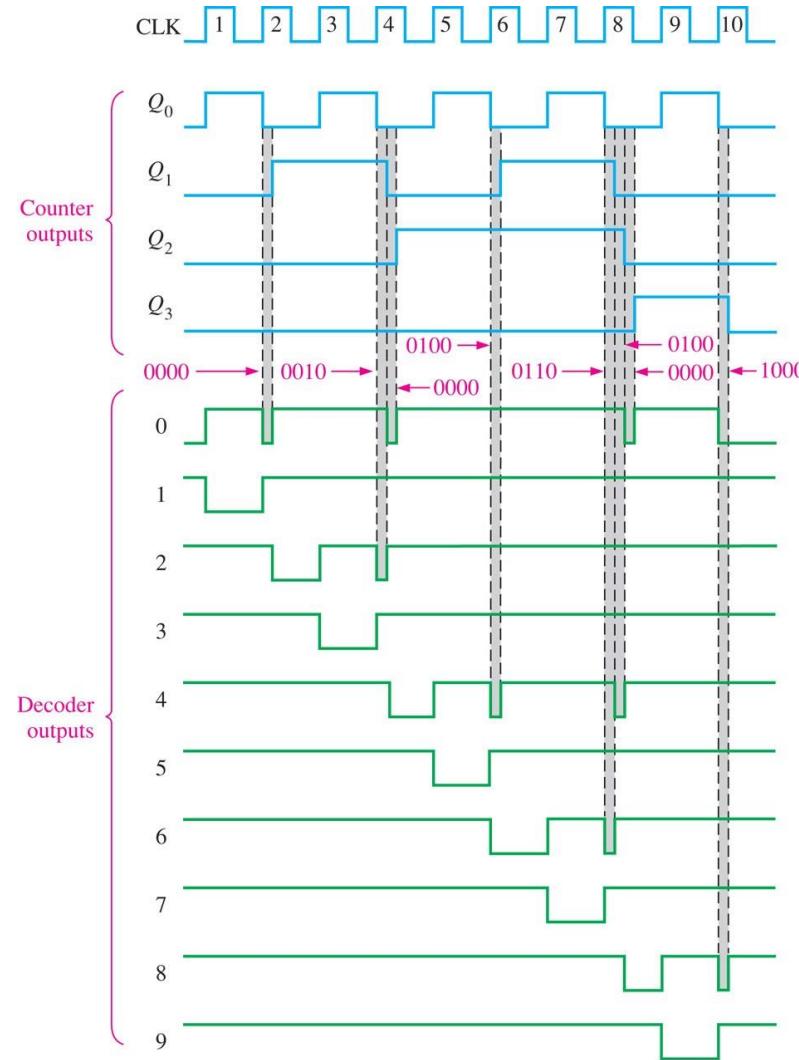
**FIGURE 9-43** A 3-bit counter with active-HIGH decoding of count 2 and count 7.



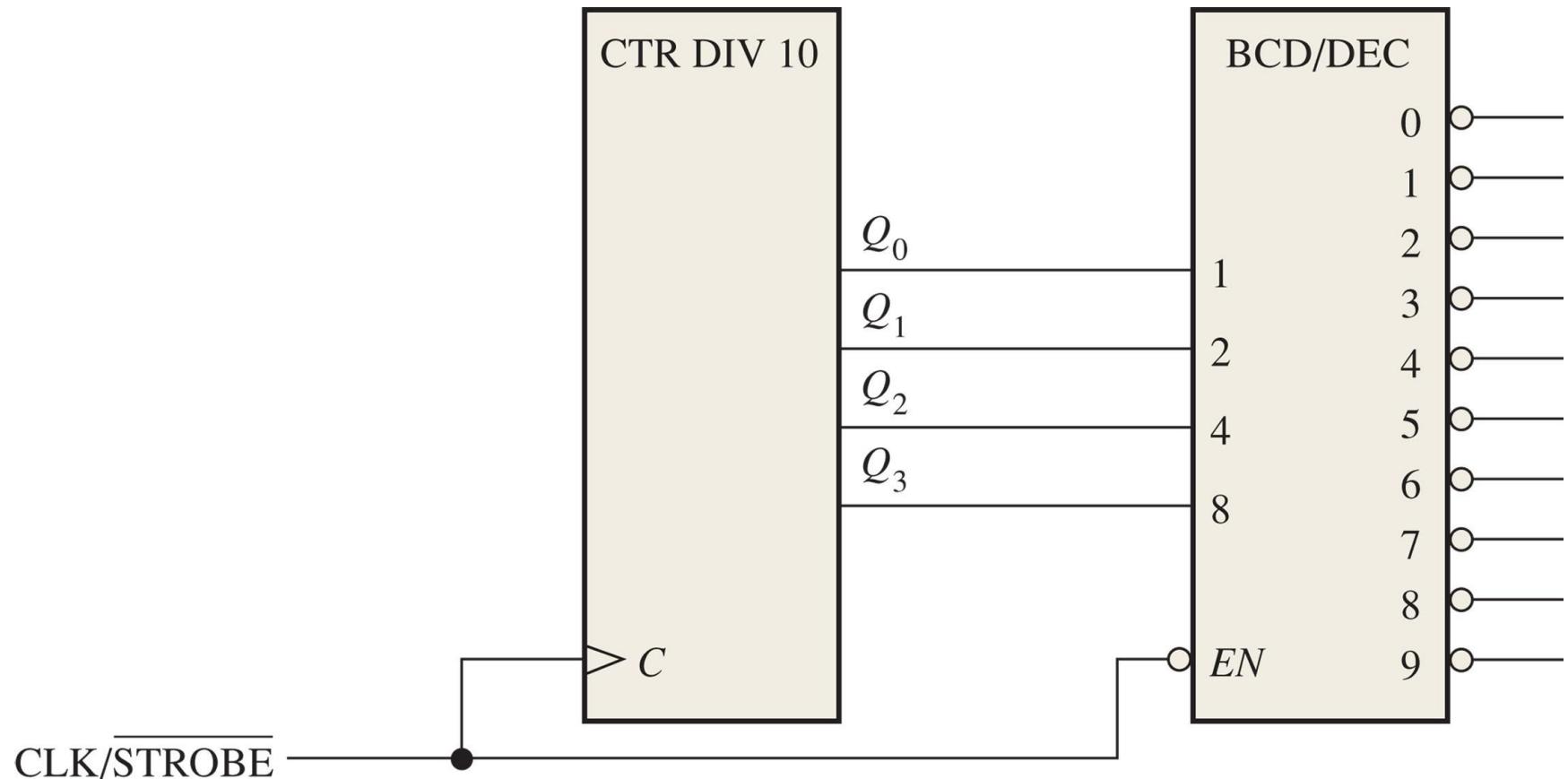
**FIGURE 9-44** A basic decade (BCD) counter and decoder.



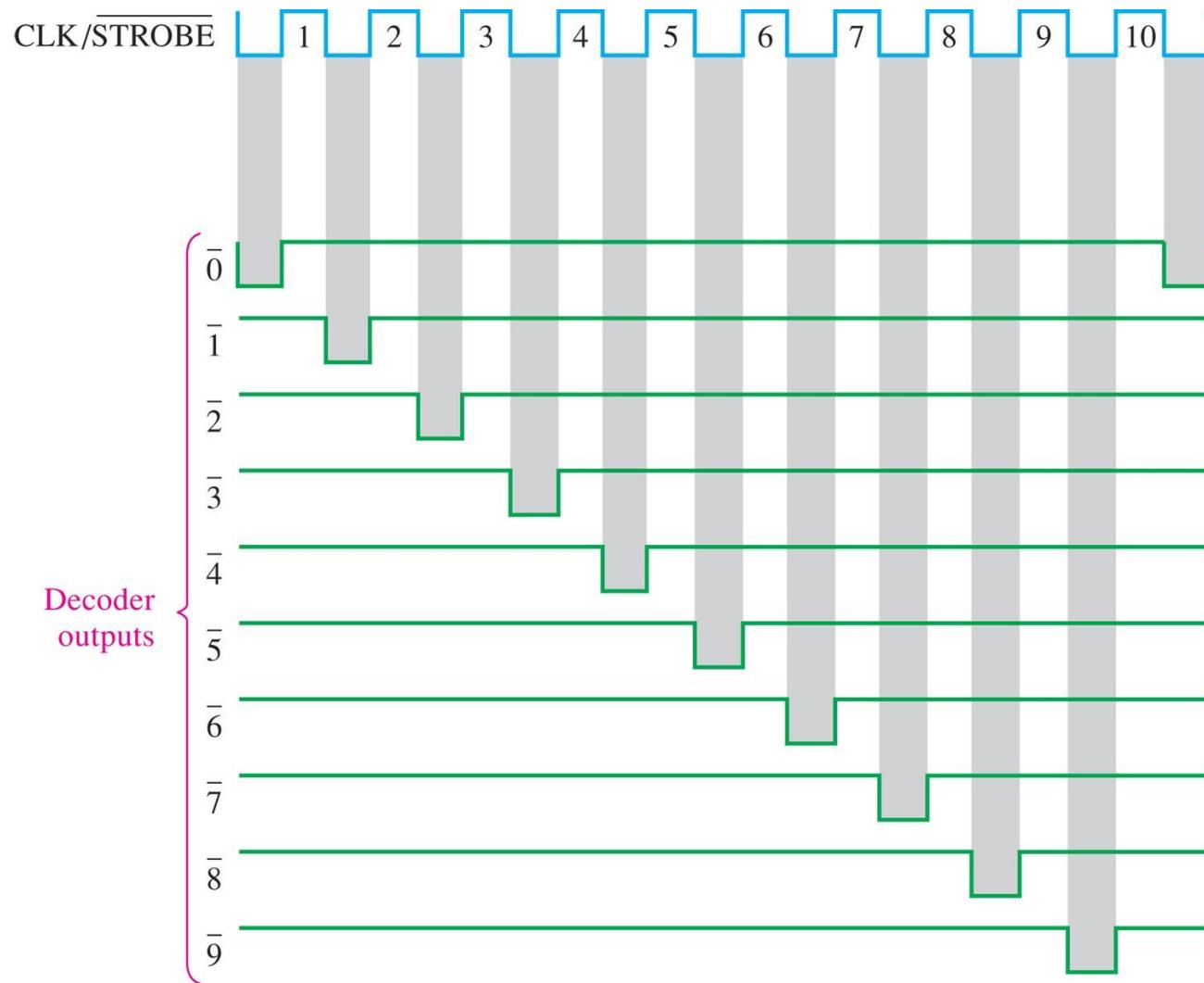
**FIGURE 9-45** Outputs with glitches from the decoder in Figure 9-44. Glitch widths are exaggerated for illustration and are usually only a few nanoseconds wide.



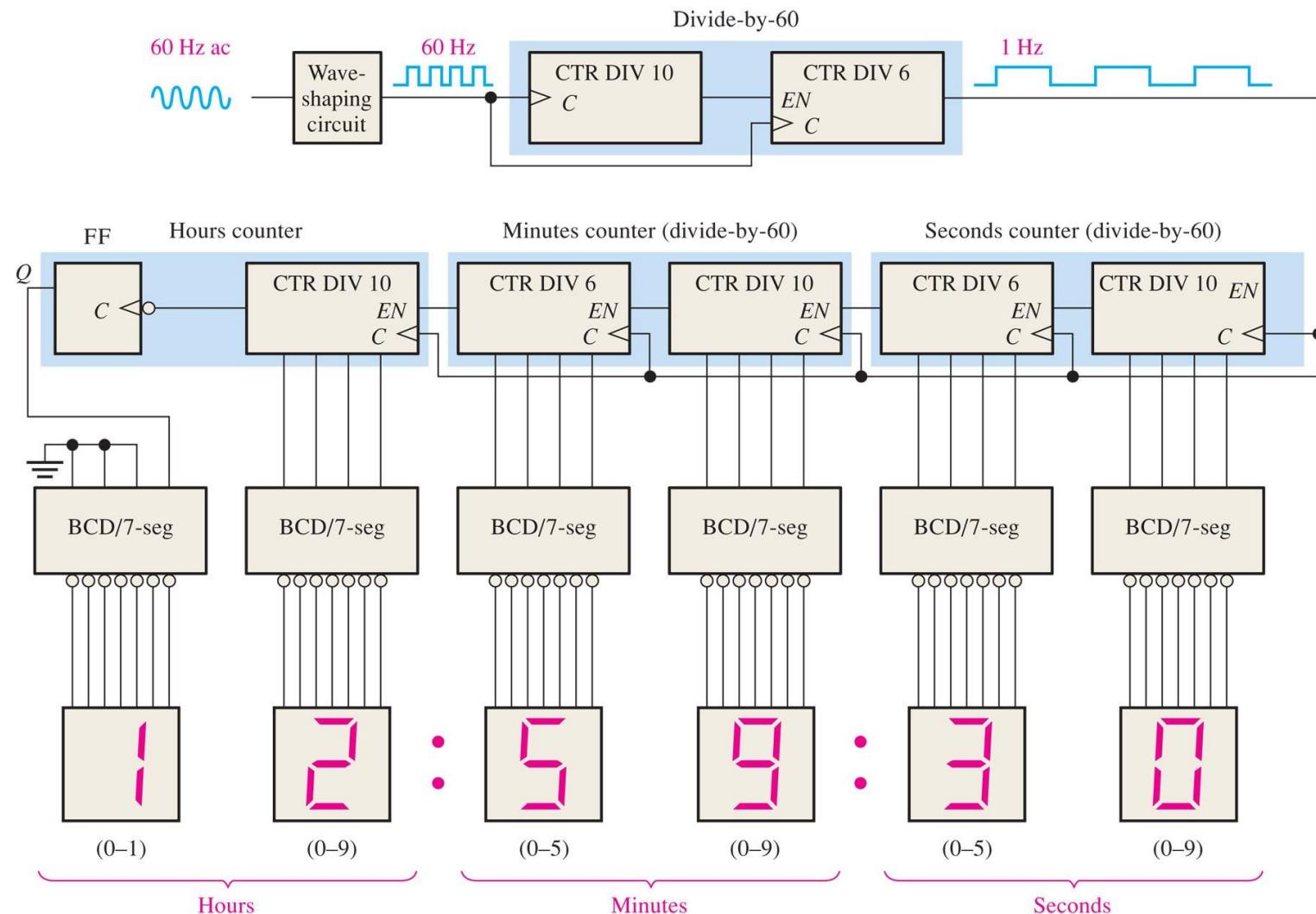
**FIGURE 9-46** The basic decade counter and decoder with strobing to eliminate glitches.



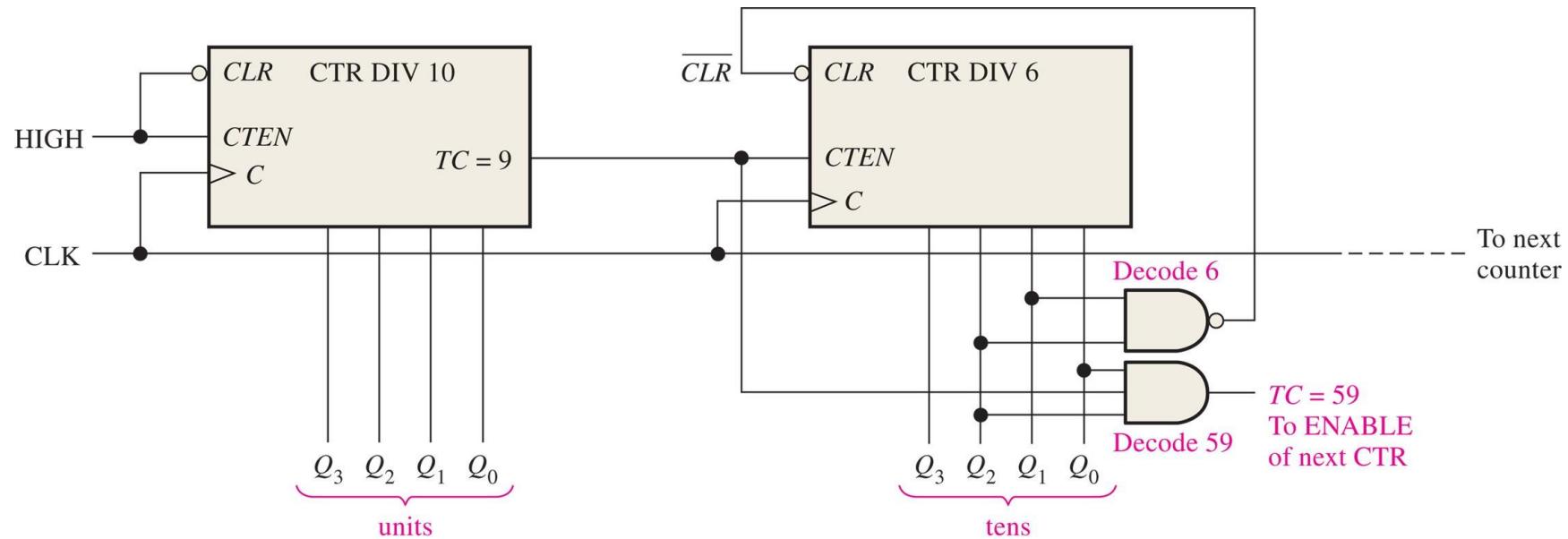
**FIGURE 9-47** Strobed decoder outputs for the circuit of Figure 9-46.



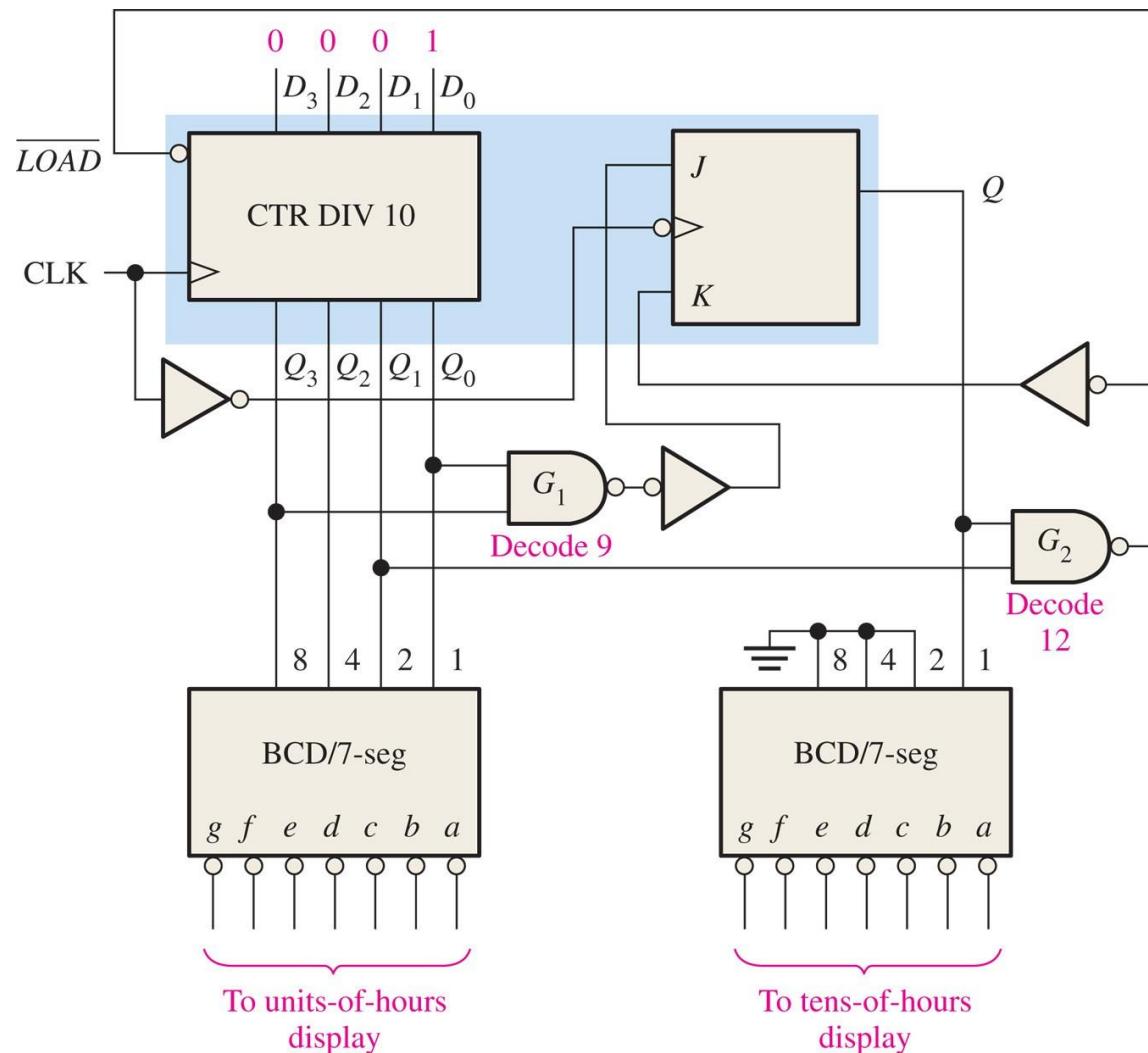
**FIGURE 9-48** Simplified logic diagram for a 12-hour digital clock. Logic details using specific devices are shown in Figures 9-49 and 9-50.



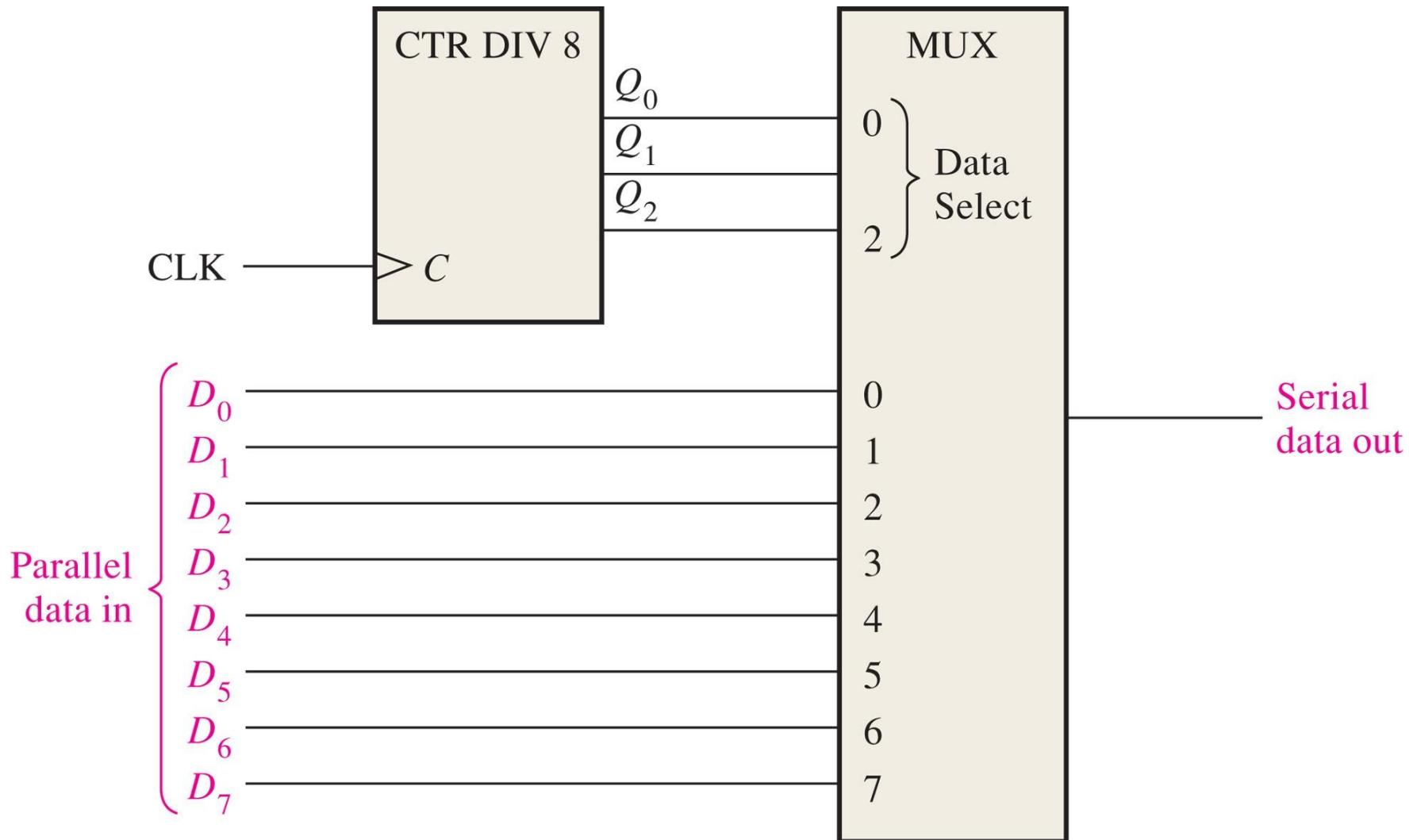
**FIGURE 9-49** Logic diagram of typical divide-by-60 counter using synchronous decade counters. Note that the outputs are in binary order (the right-most bit is the LSB).



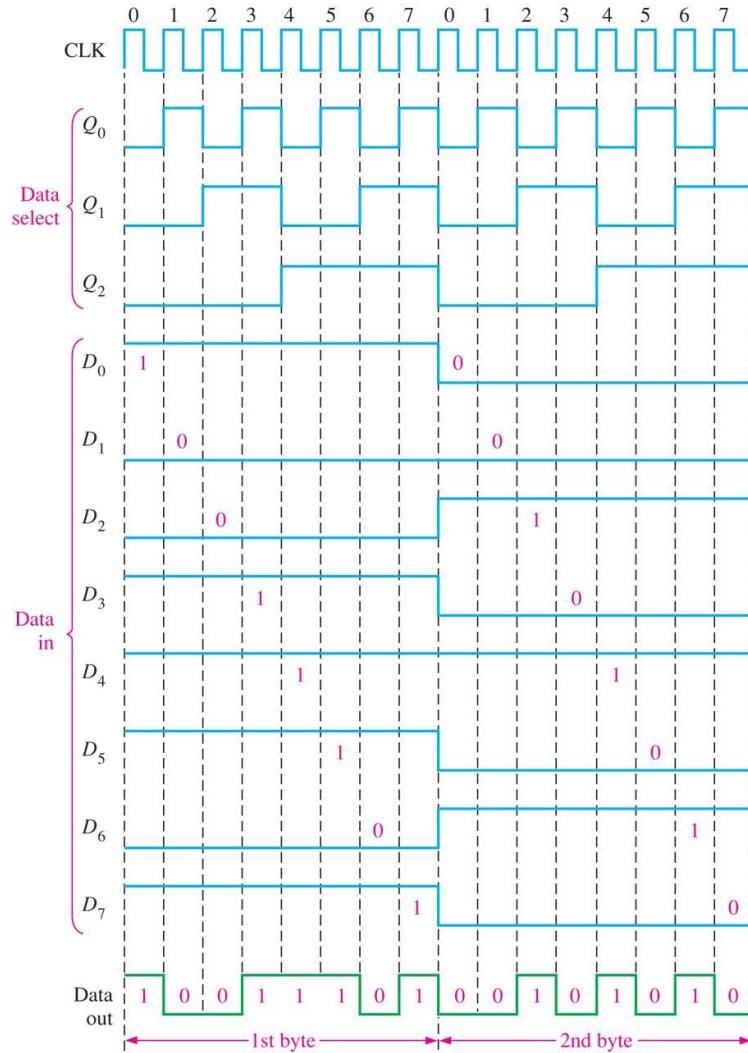
**FIGURE 9-50** GS Logic diagram for hours counter and decoders. Note that on the counter inputs and outputs, the right-most bit is the LSB.



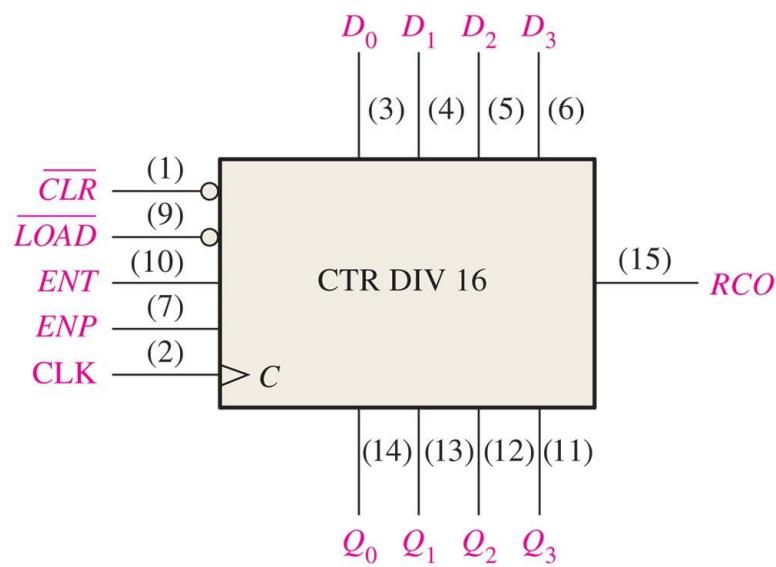
**FIGURE 9-53** Parallel-to-serial data conversion logic.



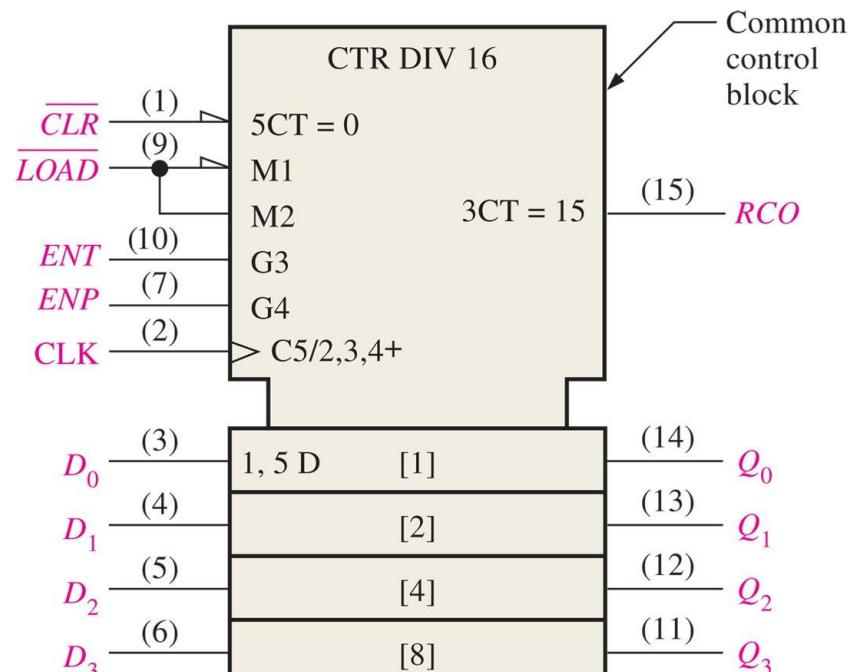
**FIGURE 9-54** Example of parallel-to-serial conversion timing for the circuit in Figure 9-53.



**FIGURE 9-55** The 74HC163 4-bit synchronous counter.

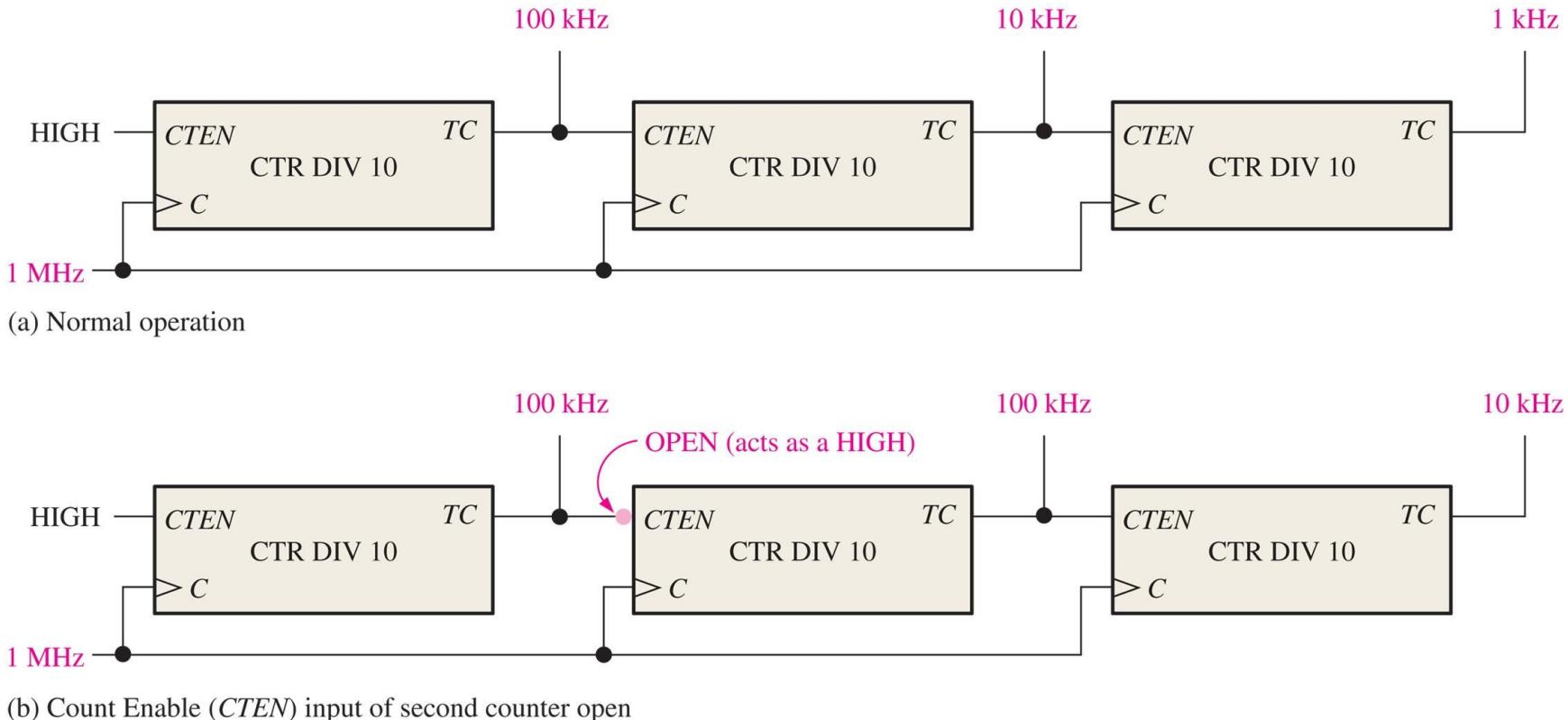


(a) Traditional block symbol

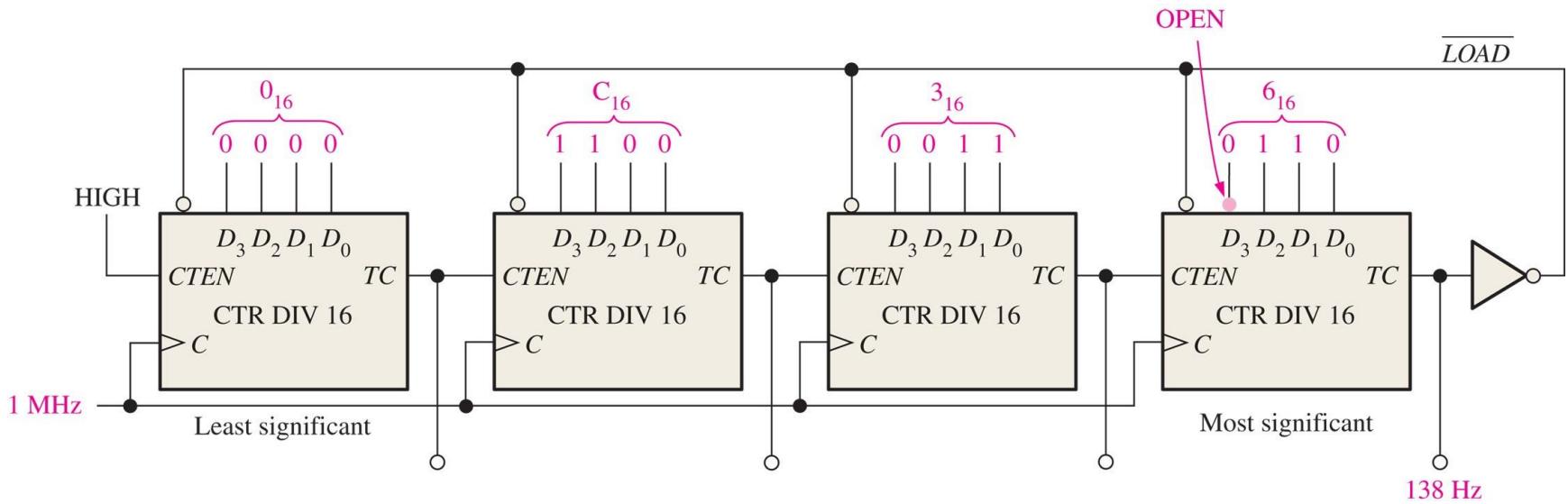


(b) ANSI/IEEE Std. 91-1984 logic symbol

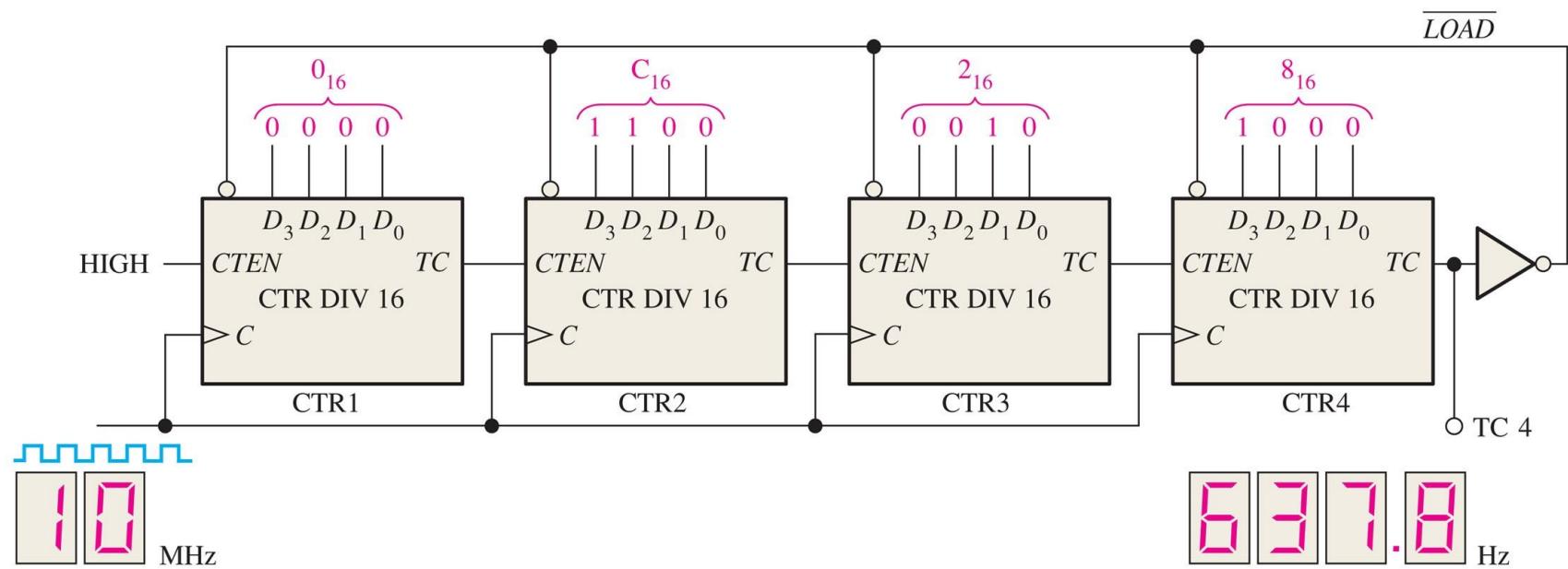
**FIGURE 9-56** Example of a failure that affects following counters in a cascaded arrangement.



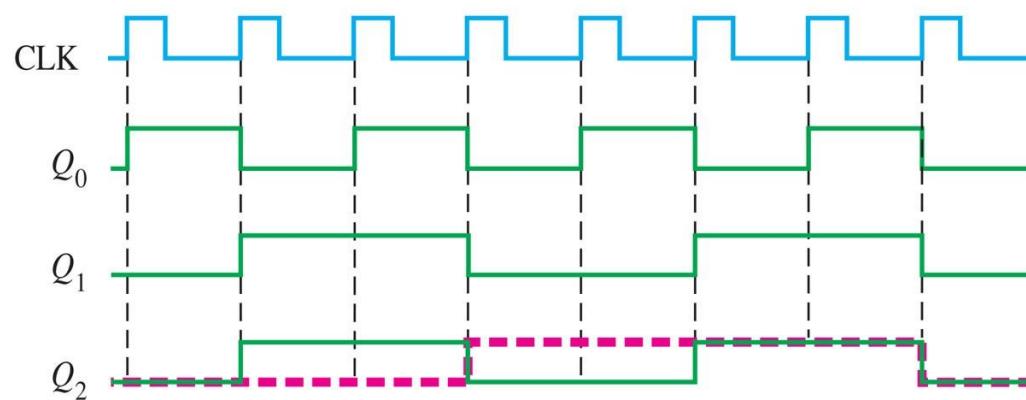
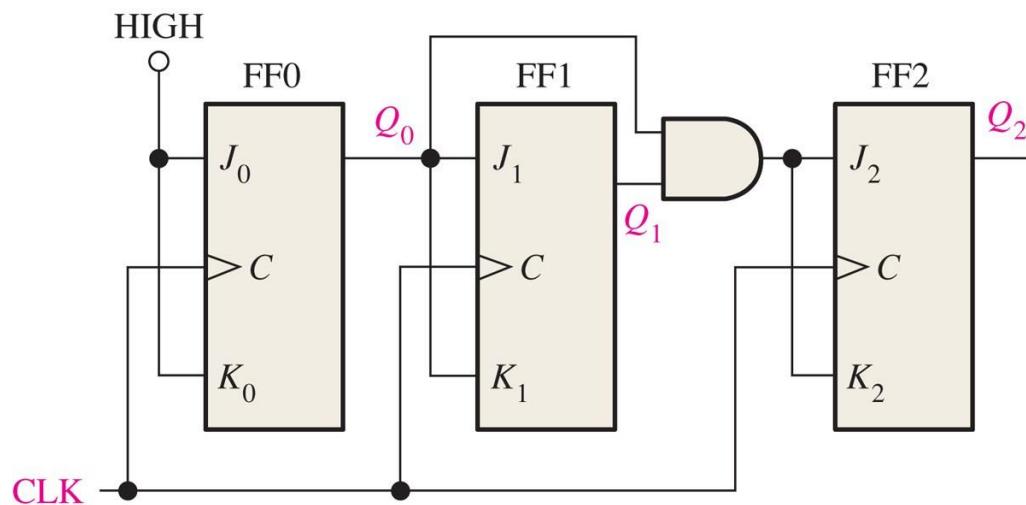
**FIGURE 9-57** Example of a failure in a cascaded counter with a truncated sequence.



**FIGURE 9-58**

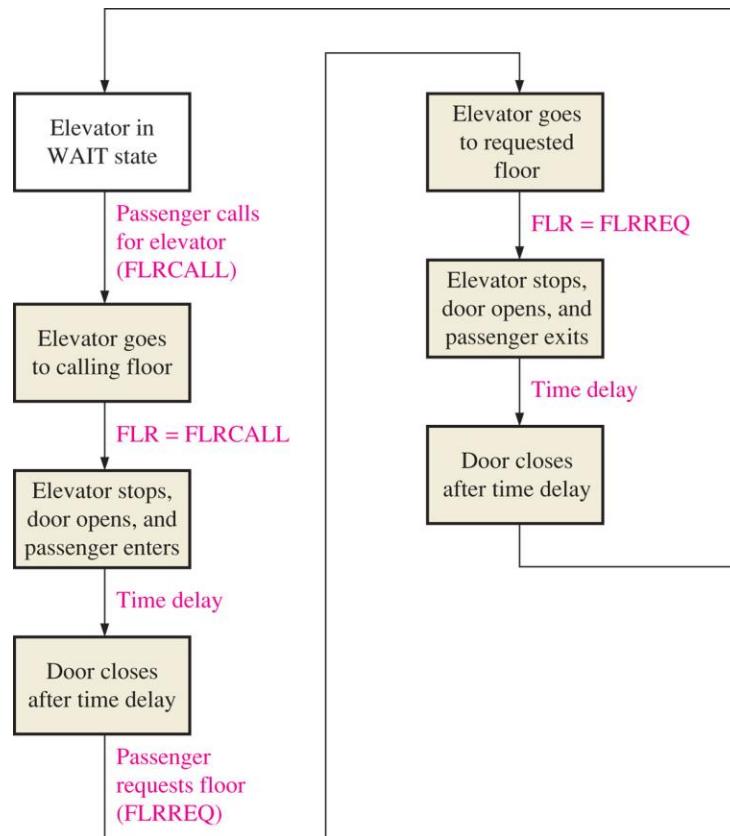


**FIGURE 9-59**



**FIGURE 9-60** One cycle of the elevator operation.

For simplicity, there is only one floor call and one floor request for each elevator cycle. A cycle occurs when the elevator is called to a given floor to pick up a passenger and the passenger is delivered to a requested floor.

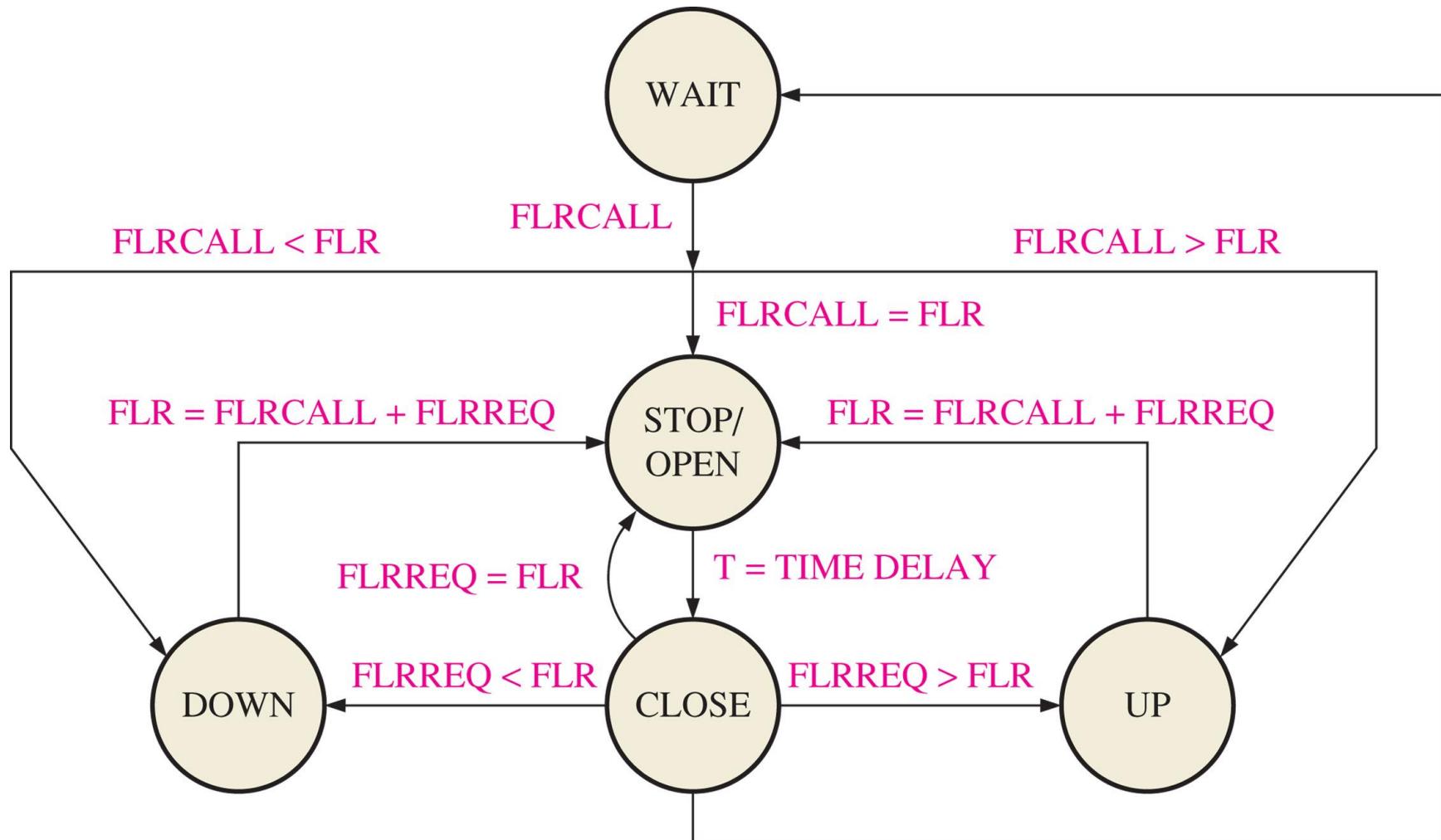


The five states in the elevator control sequence are WAIT, DOWN, UP, STOP/OPEN, and CLOSE. In the WAIT state, the elevator is waiting on the last floor serviced for an external call button (FLRCALL) on any floor to be pressed. When there is a call for the elevator from any floor, the appropriate command (UP or DOWN) is issued.

When the elevator arrives and stops at the calling floor, the door opens; the person enters and presses a number to request a destination floor. If the number of the requested floor is less than the number of the current floor, the elevator goes into the DOWN mode.

If the number of the requested floor is greater than the number of the current floor, the elevator goes into the UP mode. The elevator goes to the STOP/OPEN mode at the requested floor to allow exit. After the door is open for a specified time, it closes and then goes back to the WAIT state until another floor call is received.

**FIGURE 9-61** Elevator controller state diagram.



**WAIT** The system always begins in the WAIT state on the floor last serviced. When a floor call (FLRCALL) signal is received, the control logic determines if the number of the calling floor is greater than the current floor (FLRCALL > FLR), less than the current floor (FLRCALL < FLR), or equal to the current floor (FLRCALL = FLR) and puts the system in the UP mode, DOWN mode, or OPEN mode, respectively.

**DOWN** In this state, the elevator moves down toward the calling floor.

**UP** In this mode, the elevator moves up toward the calling floor.

**STOP/OPEN** This state occurs when the calling floor has been reached. When the number of the floor where the elevator is equals the number of the calling or requested floor, a signal is issued to stop the elevator and open the door.

**CLOSE** After a preset time (T) to allow entry or exit, the door closes. The signals used by the elevator controller are defined as follows:

**FLR** Number of floor represented by a 3-bit binary code.

**Floor sensor pulse** A pulse issued at each floor to clock the floor counter to the next state.

**FLRCALL** Number of floor where a call for elevator service originates, represented by a 3-bit binary code.

Call pulse A pulse issued in conjunction with FLRCALL to clock the 3-bit code into a register.

FLRREQ Number of floor to which the passenger desires to go, represented by a 3-bit binary code. Request pulse A pulse issued in conjunction with FLRREQ to clock the 3-bit code into a register.

UP A signal issued to the elevator motor control to cause the elevator to move from a lower floor to a higher floor.

DOWN A signal issued to the elevator motor control to cause the elevator to move from a higher floor to a lower floor.

STOP A signal issued to the elevator motor control to cause the elevator to stop.

OPEN A signal issued to door motor control to cause the door to open.

CLOSE A signal issued to the door motor control to cause the door to close.

**FIGURE 9-62** Elevator controller block diagram.

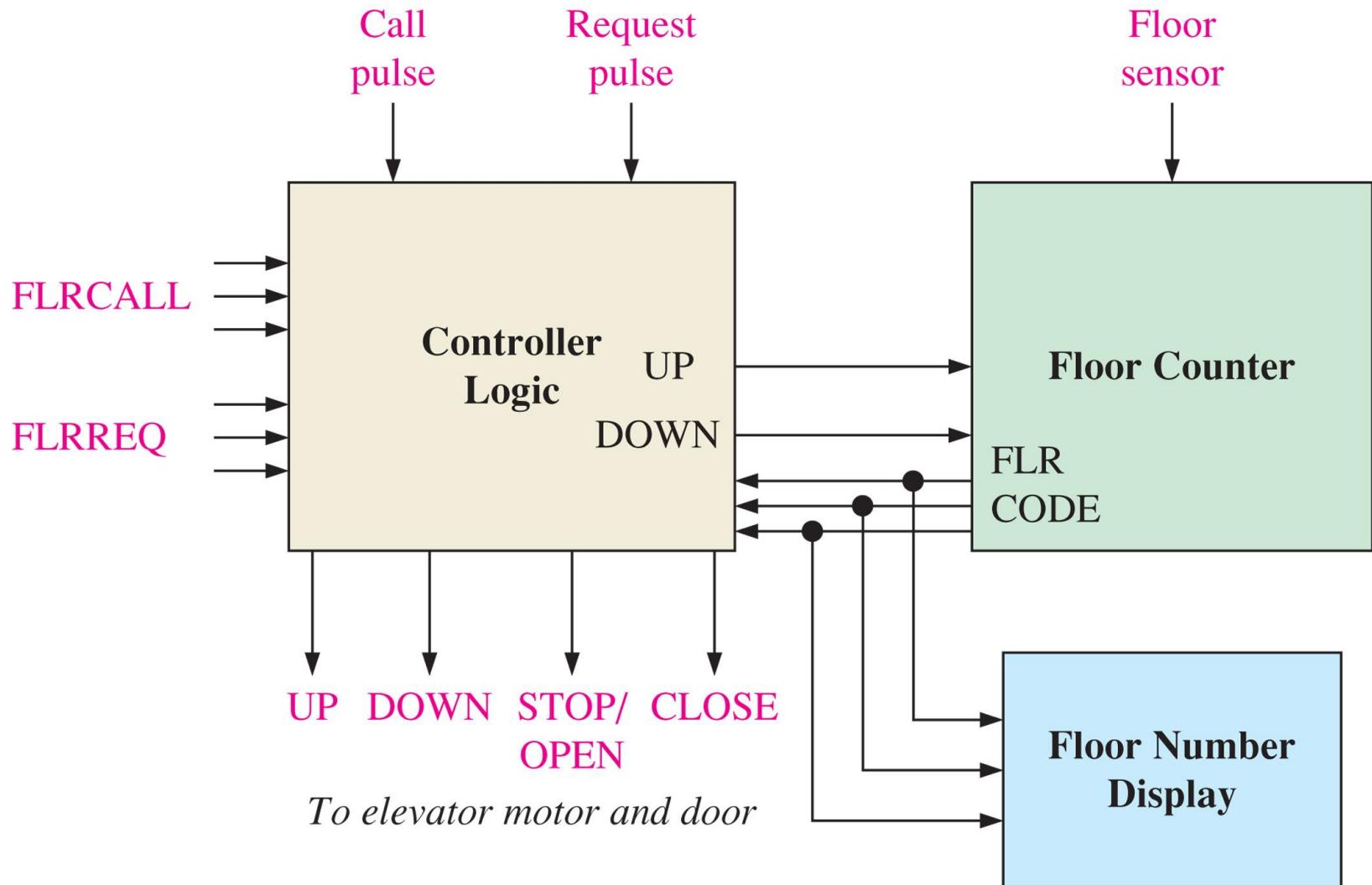
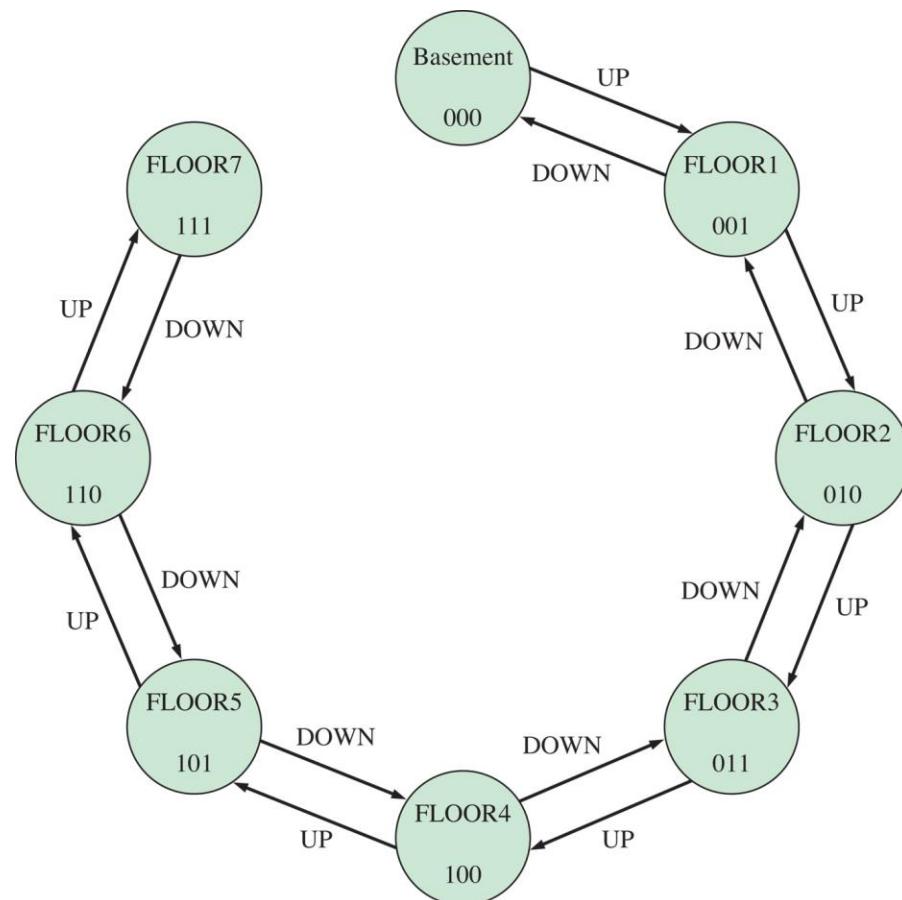


Figure 9–62 shows the elevator controller block diagram, which consists of controller logic, a floor counter, and a floor number display. Assume that the elevator is on the first floor in the WAIT state. The floor counter contains 001, which is the first floor code. Suppose the FLRCALL (101) comes in from the call button on the fifth floor. Since FLRCALL 7 FLR (101 7 001), the controller issues an UP command to the elevator motor.

As the elevator moves up, the floor counter receives a floor sensor pulse as it reaches each floor which advances its state (001, 010, 011, 100, 101).

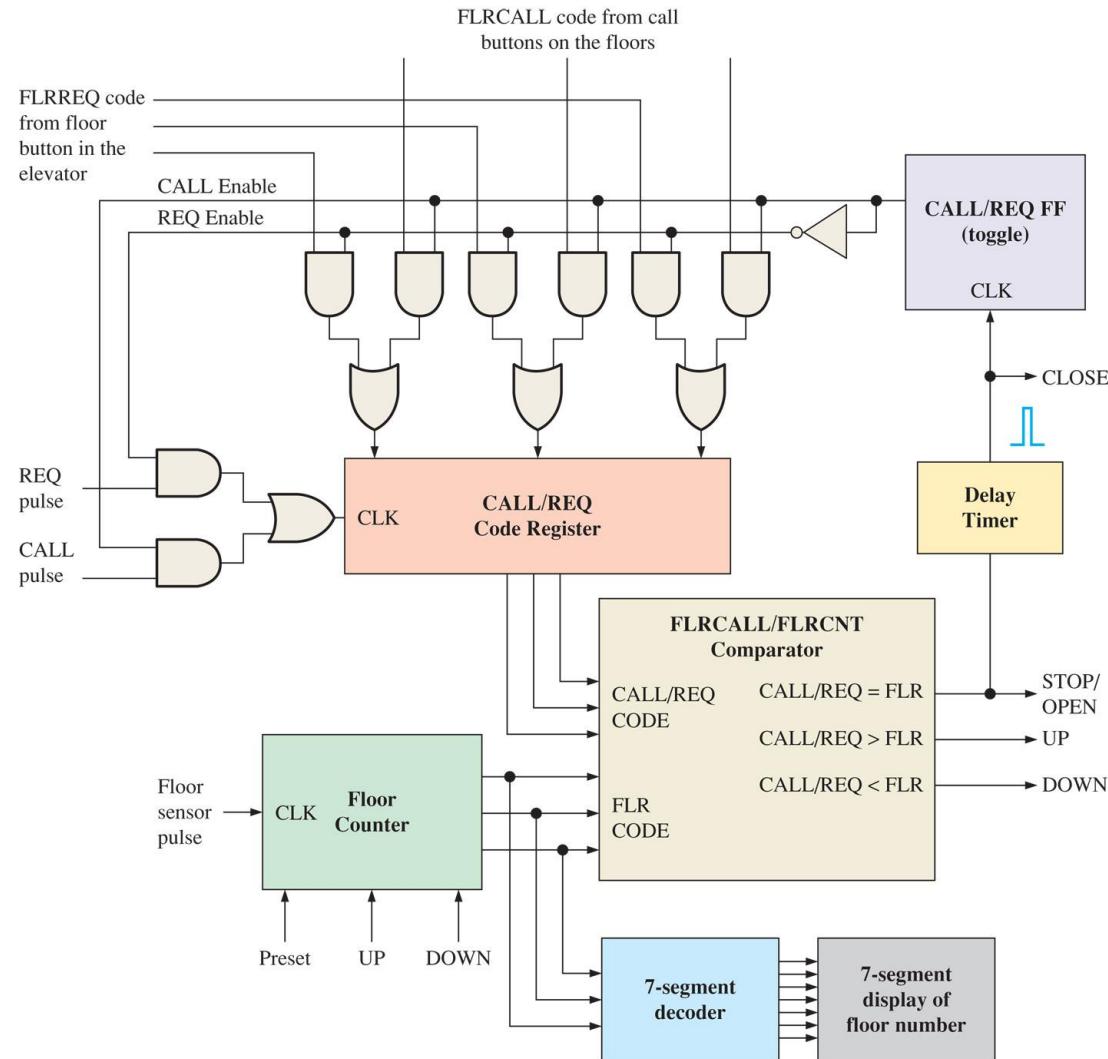
When the fifth floor is reached and  $\text{FLR} = \text{FLRCALL}$ , the controller logic stops the elevator and opens its door. The process is repeated for a FLRREQ input

The floor counter sequentially tracks the number of the floor and always contains the number of the current floor. It can count up or down and can reverse its state at any point under the direction of the state controller and the floor sensor input. A 3-bit counter is required since there are eight floors ( $2^3 = 8$ ) including the basement, as shown in the floor counter state diagram in Figure 9–63.



**FIGURE 9–63** Floor counter state diagram.

**FIGURE 9-64** Elevator controller logic diagram.



Operation of Elevator Controller The elevator controller logic diagram is shown in Figure 9–64.

Elevator action is initiated by either a floor call (FLRCALL) or a floor request (FLRREQ).

Keep in mind that FLRCALL is when a person calls the elevator to come to a particular floor.

FLRREQ is when a passenger in the elevator requests to go to a specified floor. This simplified operation is based on a CALL/REQ sequence; that is, a call followed by a request followed by a call

As you know, FLRCALL and FLRREQ are 3-bit codes representing specific floors. When a person presses a call button on a given floor, the specific 3-bit code for that floor is placed on the inputs to the CALL/REQ code register and a CALL pulse is generated to enter the code into the register.

The same process occurs when a request button is pressed inside the elevator. The code is input to the CALL/REQ code register, and a REQ pulse is generated to store the code in the register

The elevator does not know the difference between a call and a request. The comparator determines if the destination floor number is greater than, less than, or equal to the current

floor where the elevator is located. As a result of this comparison, either an UP command, a DOWN command, or an OPEN command is issued to the elevator motor control.

As the elevator moves toward the desired floor, the floor counter is either incremented at each floor as it goes up or decremented at each floor as it goes down.

Once the elevator reaches the desired floor, a STOP/OPEN command is issued to the elevator motor control and to the door control. After a preset time, the delay timer issues a CLOSE signal to the elevator door control.

As mentioned, this elevator design is limited to one floor call and one floor request per cycle.

## Initialization

The initial one-time setup requires that the elevator be placed at the basement level and the floor counter be preset to 000.

After this, the counter will automatically move through the sequence of states determined by the elevator position.