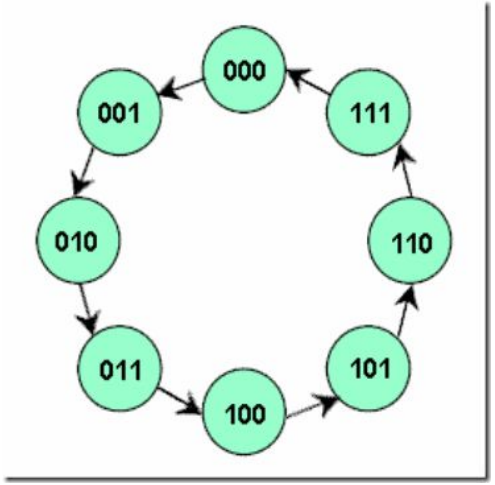
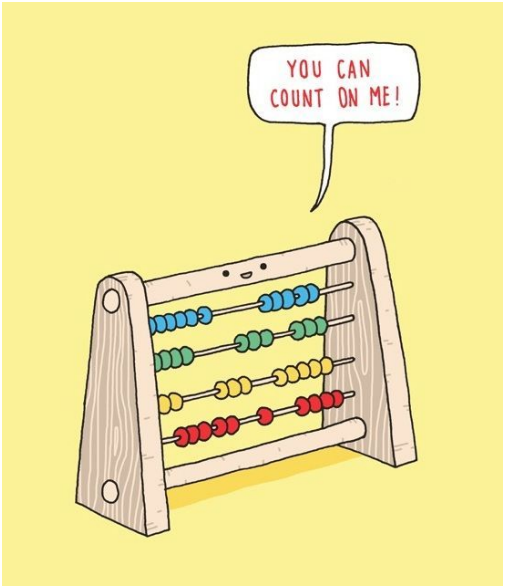


COUNTERS

Chapter 9



Sumaiyah Zahid

Asynchronous Counters

In asynchronous counter flip-flops (FF) do not change states at exactly the same time because they do not have a common clock pulse.

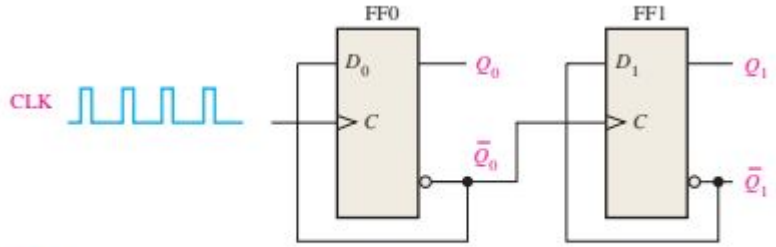
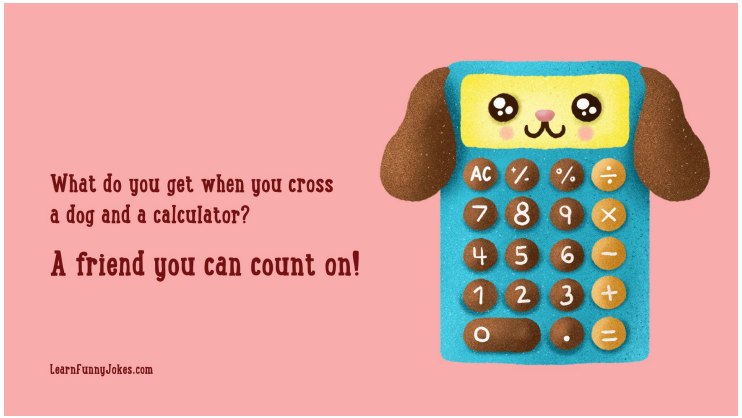


FIGURE 9–4 A 2-bit asynchronous binary counter. Open file F09-04 to verify operation. *Multisim tutorial is available on the website.*



Asynchronous Counters

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

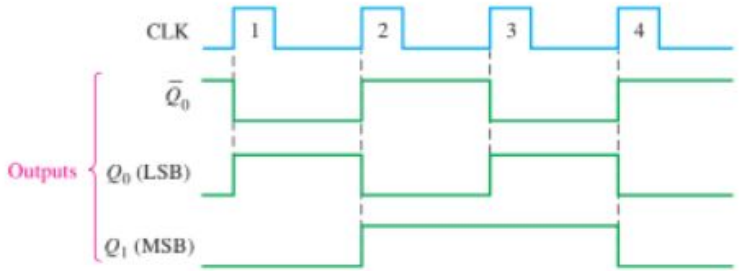
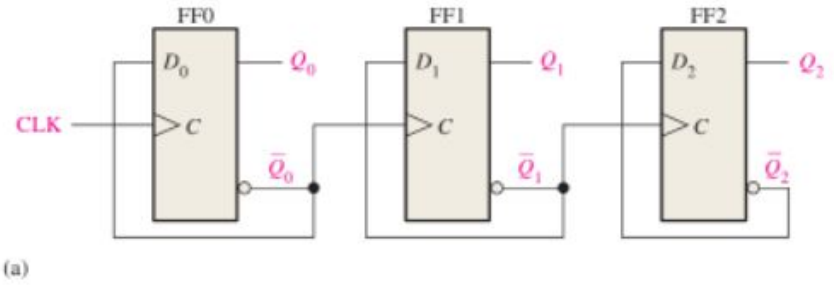
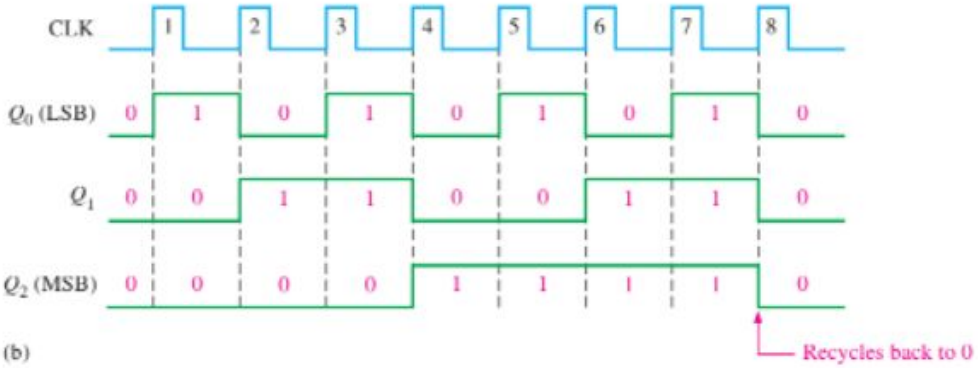


FIGURE 9–5 Timing diagram for the counter of Figure 9–4. As in previous chapters, output waveforms are shown in green.

3-Bit Asynchronous Binary Counter



(a)



(b)

Propagation Delay

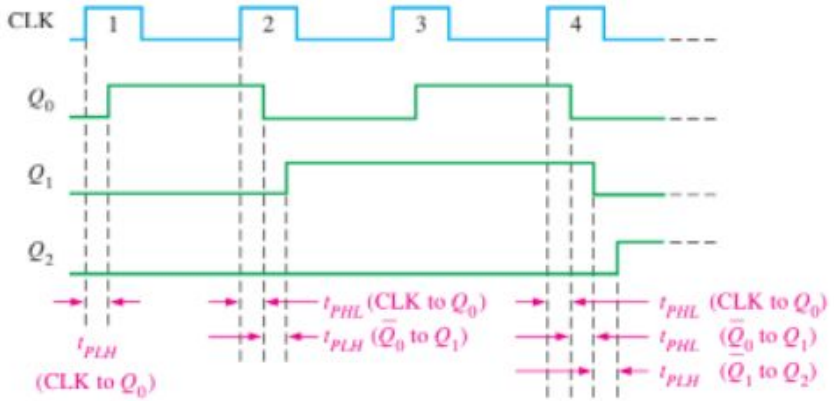
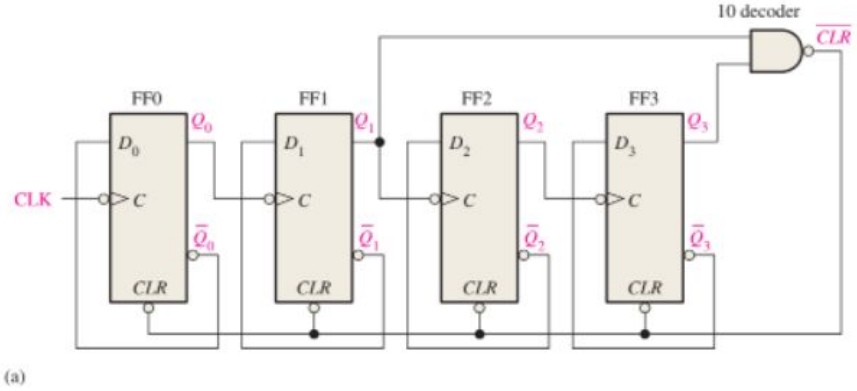
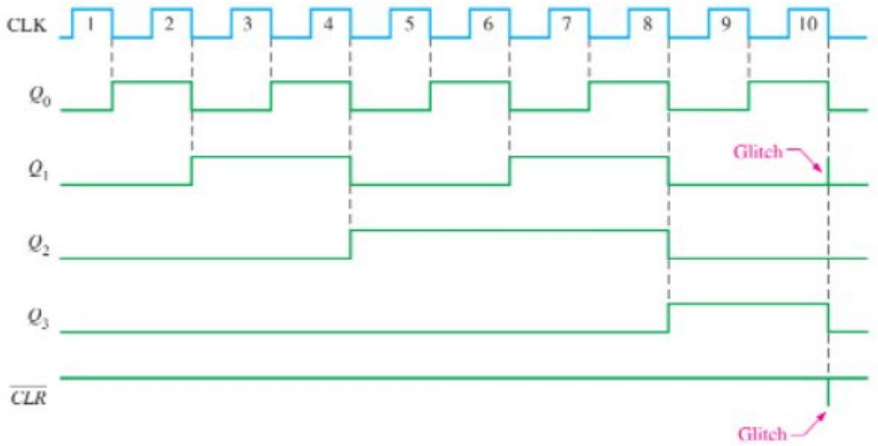


FIGURE 9–7 Propagation delays in a 3-bit asynchronous (ripple-clocked) binary counter.

Asynchronous Decade Counters

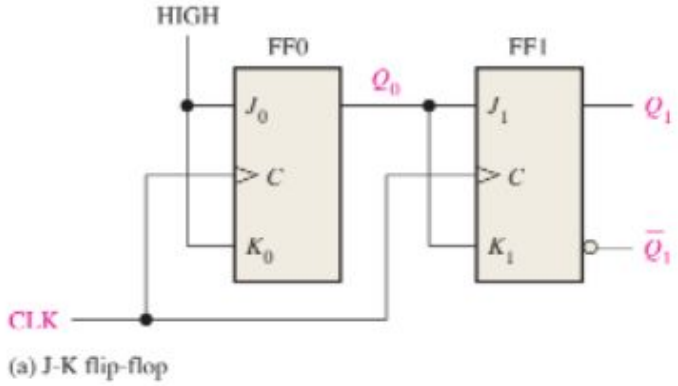


(a)

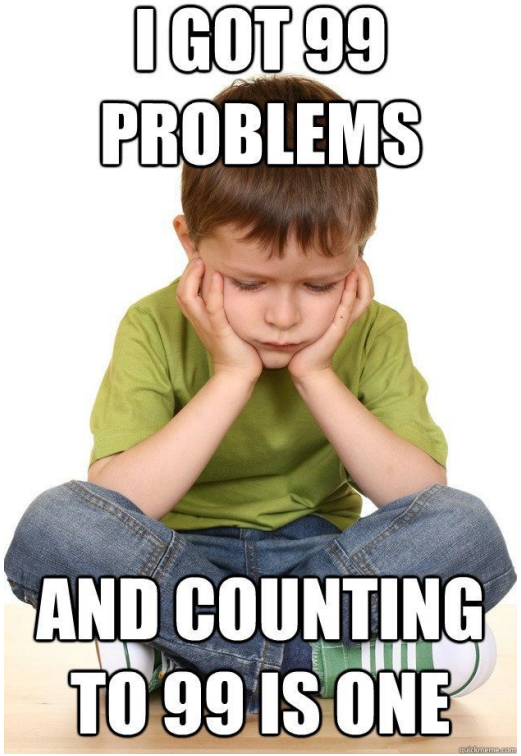


Synchronous Counters

In synchronous counter all the flip-flops in the counter are clocked at the same time by a common clock pulse.



(a) J-K flip-flop



3-Bit Synchronous Binary Counter

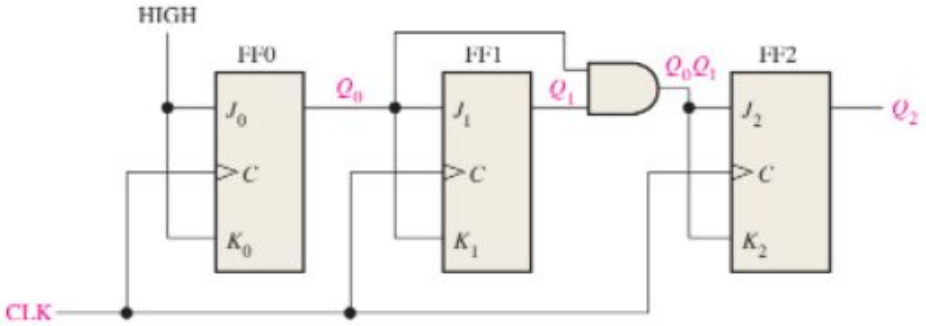


FIGURE 9–15 A 3-bit synchronous binary counter. Open file F09-15 to verify the operation.

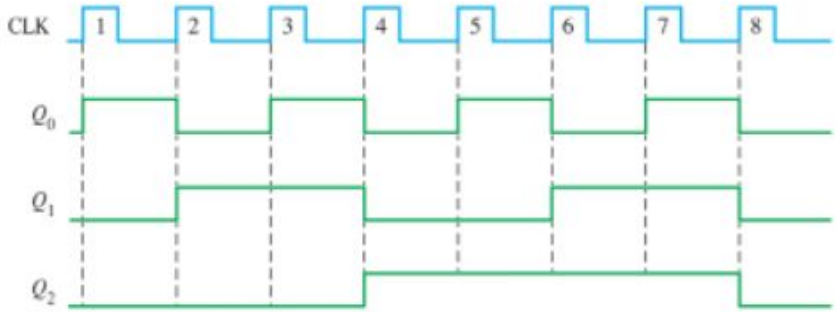
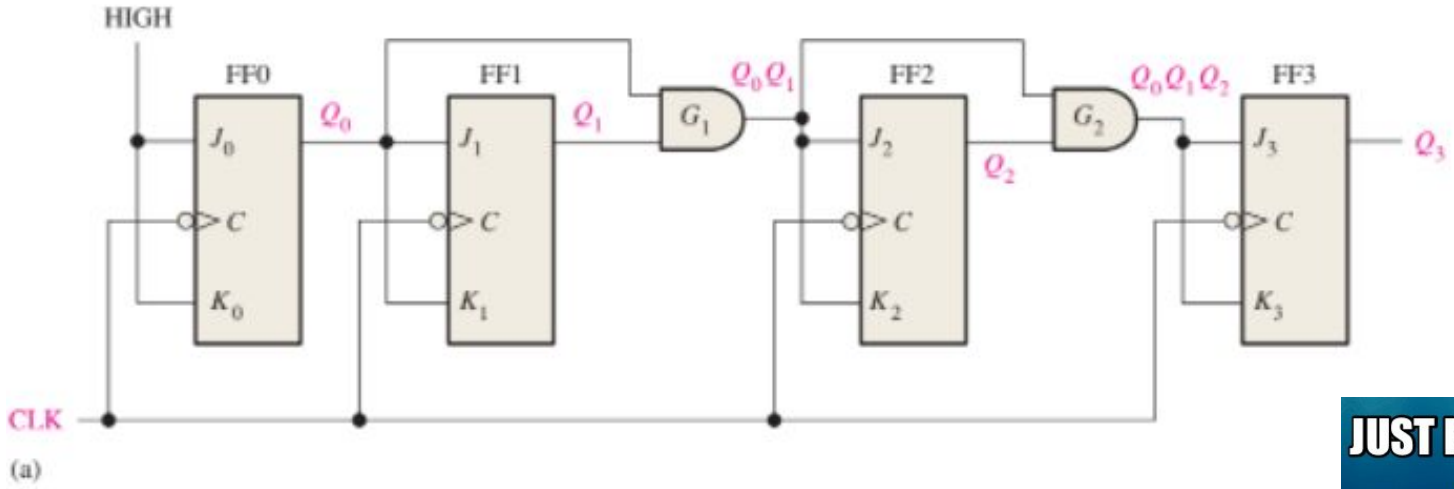


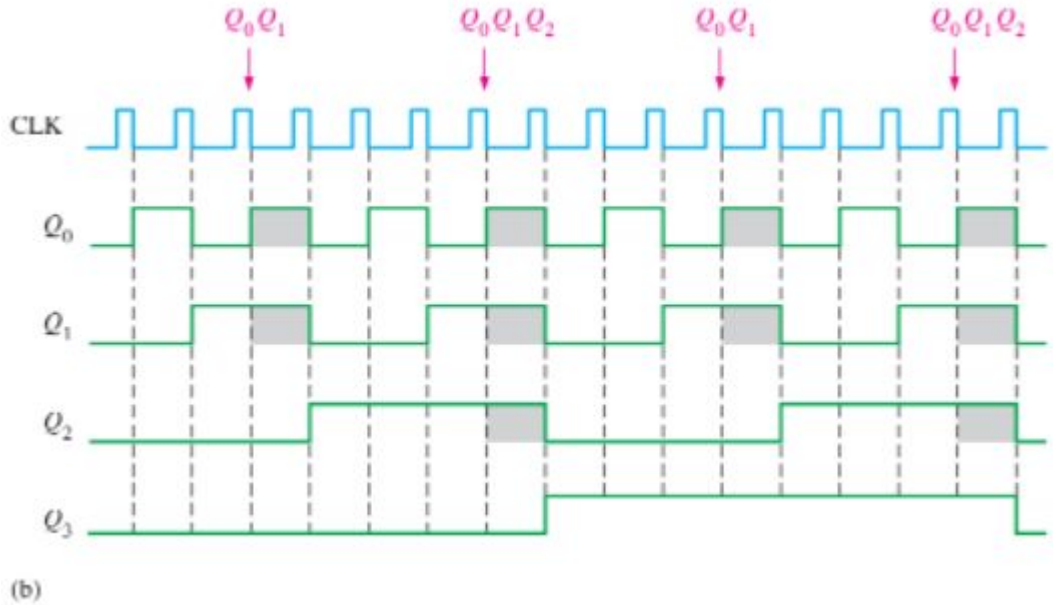
FIGURE 9–16 Timing diagram for the counter of Figure 9–15.

4-Bit Synchronous Binary Counter



(a)

4-Bit Synchronous Binary Counter



4-Bit Synchronous Decade Counter

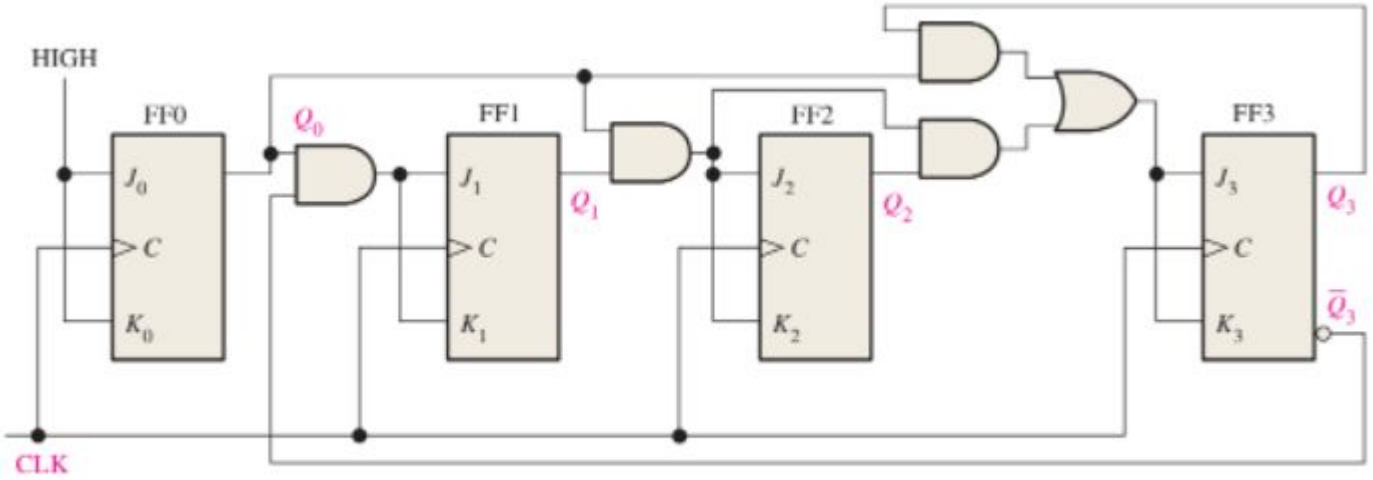


FIGURE 9–18 A synchronous BCD decade counter. Open file F09-18 to verify operation.

4-Bit Synchronous Decade Counter

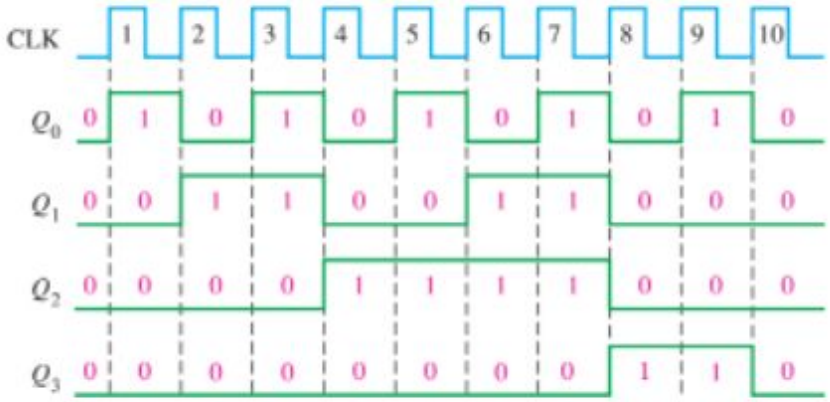
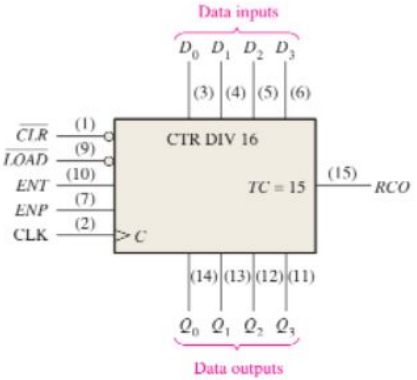
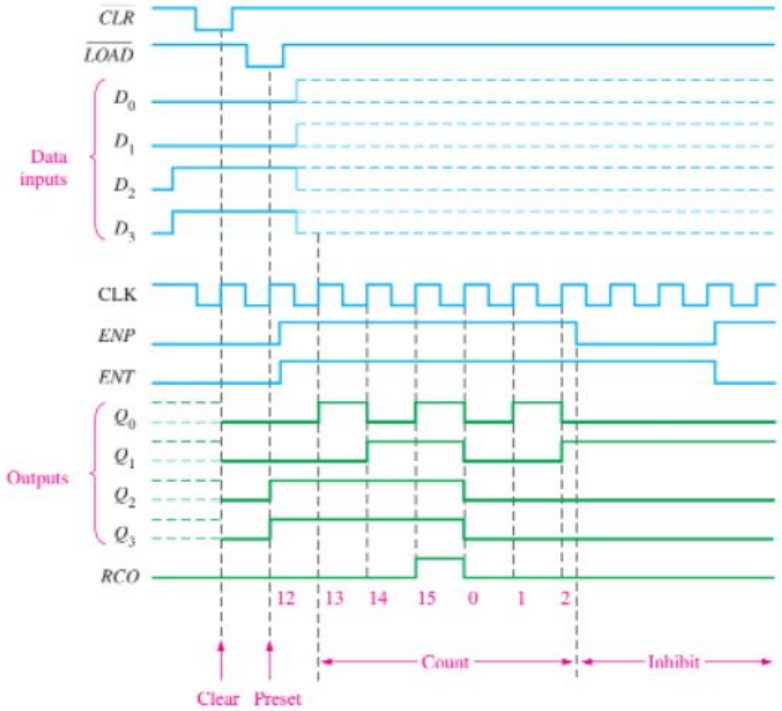


FIGURE 9–19 Timing diagram for the BCD decade counter (Q_0 is the LSB).

4-BIT SYNCHRONOUS BINARY COUNTER





Up/Down Synchronous Counters

Also called a bidirectional counter.

UP UP

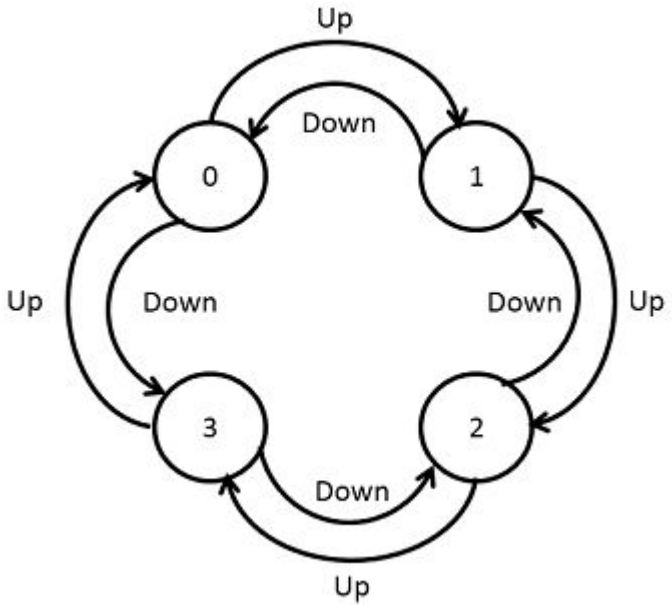
0, 1, 2, 3, 4, 5, 4, 3, 2, 3, 4, 5, 6, 7, 6, 5, etc.

DOWN DOWN

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	↓



Up/Down Synchronous Counters

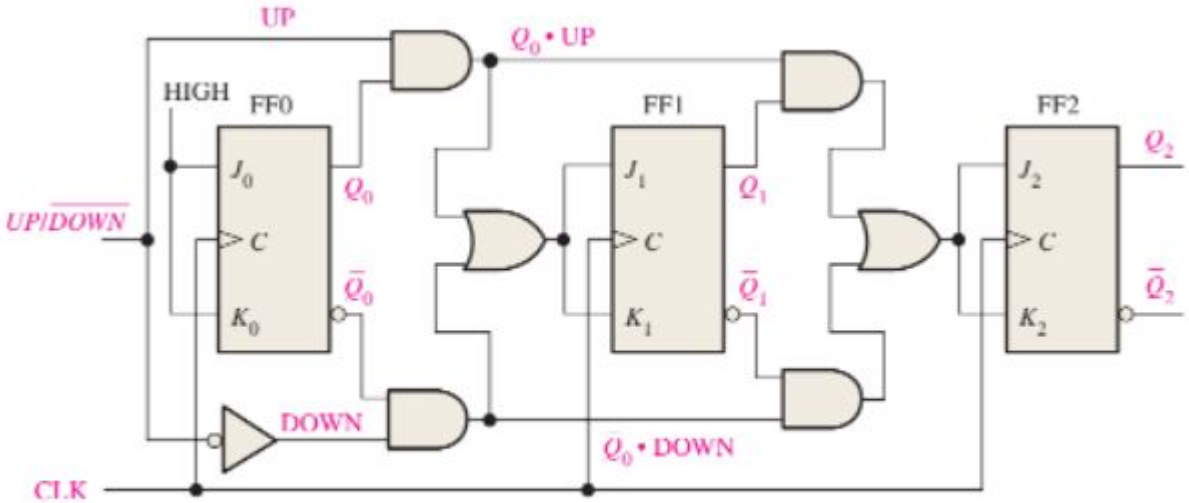


FIGURE 9–22 A basic 3-bit up/down synchronous counter. Open file F09-22 to verify operation.

Up/Down Synchronous Counters

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.

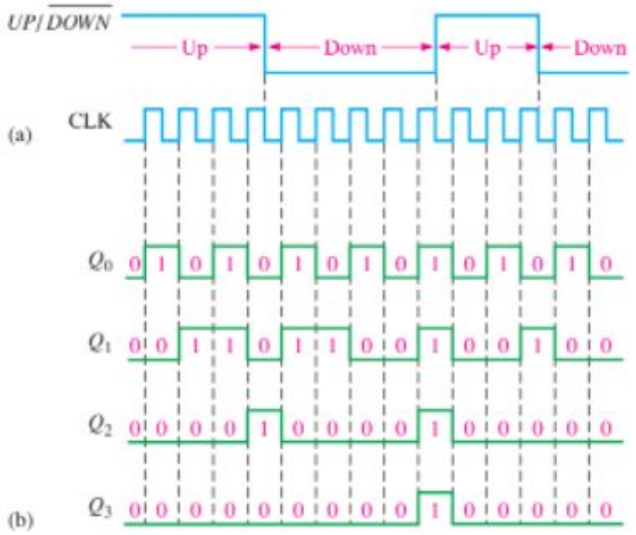
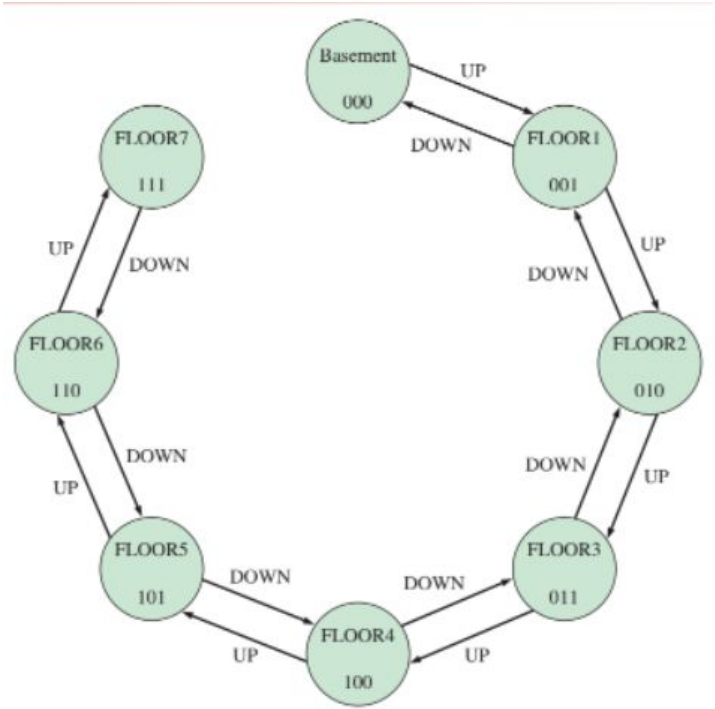


FIGURE 9–23

Up/Down Synchronous Counters

Applications

- Elevators



Cascaded Counters

Asynchronous Cascading

Total states = 4*8=32

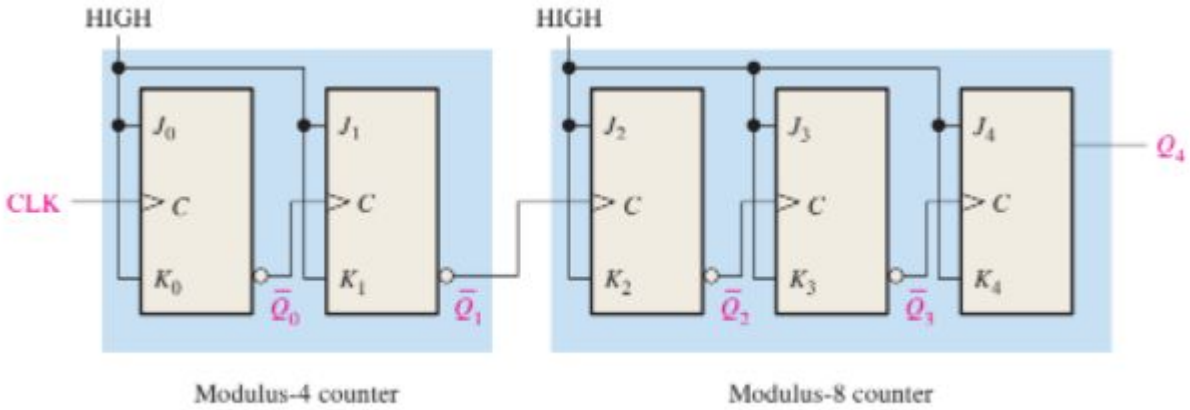


FIGURE 9–35 Two cascaded asynchronous counters (all *J* and *K* inputs are HIGH).

Cascaded Counters

Synchronous Cascading

Total counts = 10*10=100

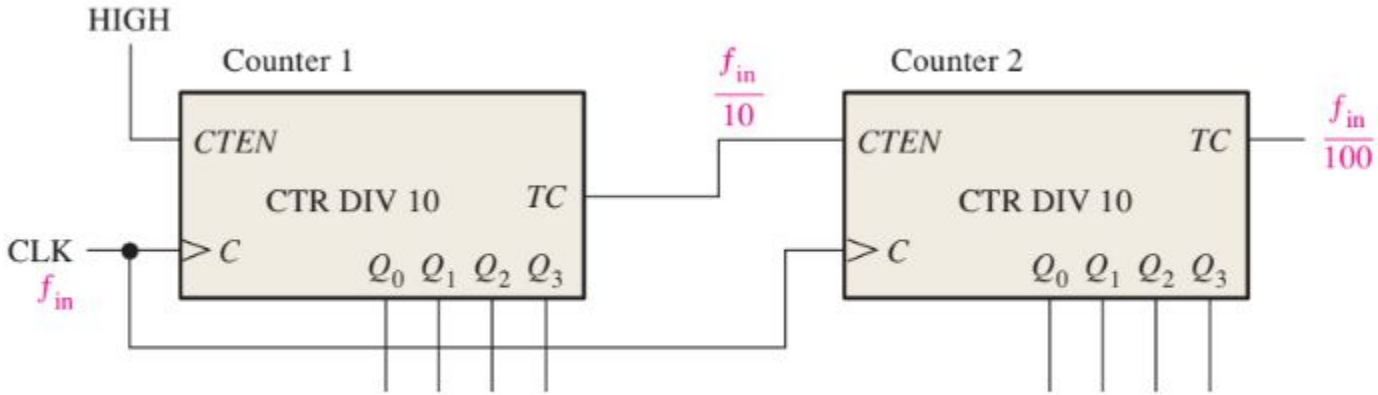


FIGURE 9–37 A modulus-100 counter using two cascaded decade counters.

Cascaded Counters

Synchronous Cascading

Frequency division by 10*10*10= 1000

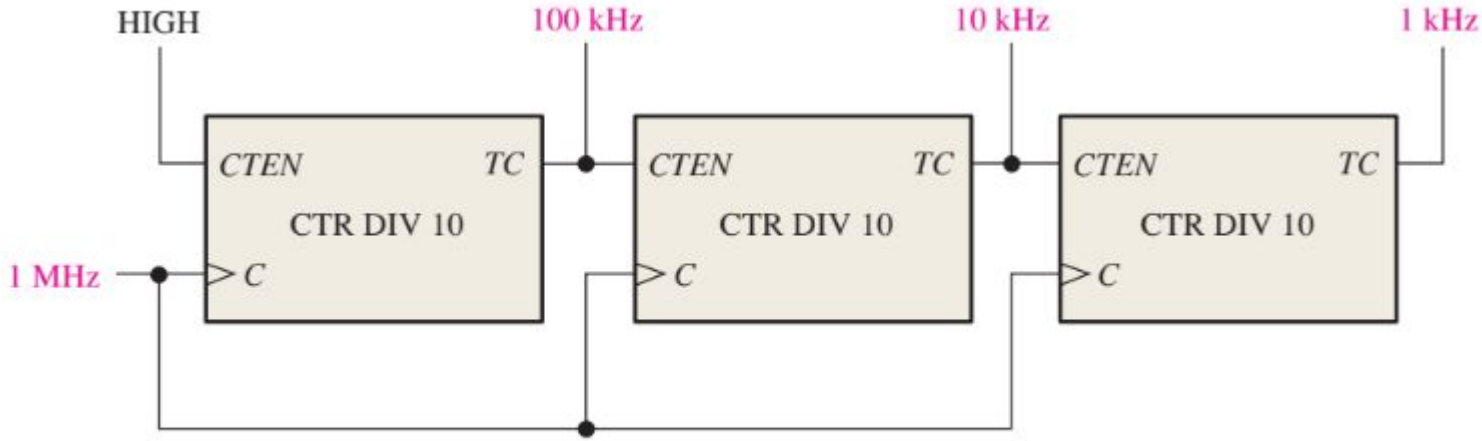
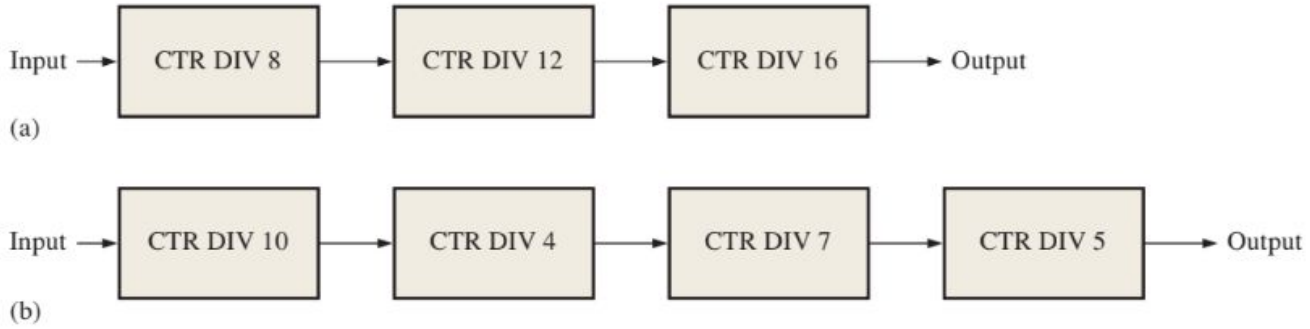


FIGURE 9–38 Three cascaded decade counters forming a divide-by-1000 frequency divider with intermediate divide-by-10 and divide-by-100 outputs.

Cascaded Counters

EXAMPLE 9–6

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



Cascaded Counters

EXAMPLE 9–6

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

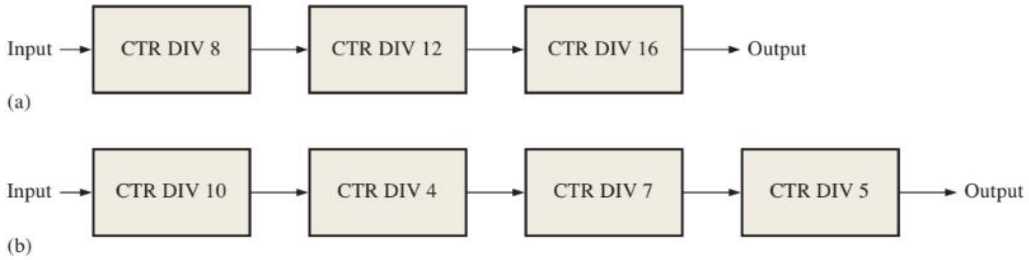


FIGURE 9–39

Solution

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

$$8 \times 12 \times 16 = \mathbf{1536}$$

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

$$10 \times 4 \times 7 \times 5 = \mathbf{1400}$$

Related Problem

How many cascaded decade counters are required to divide a clock frequency by 100,000?

Cascaded Counters

Cascaded Counters with Truncated Sequences

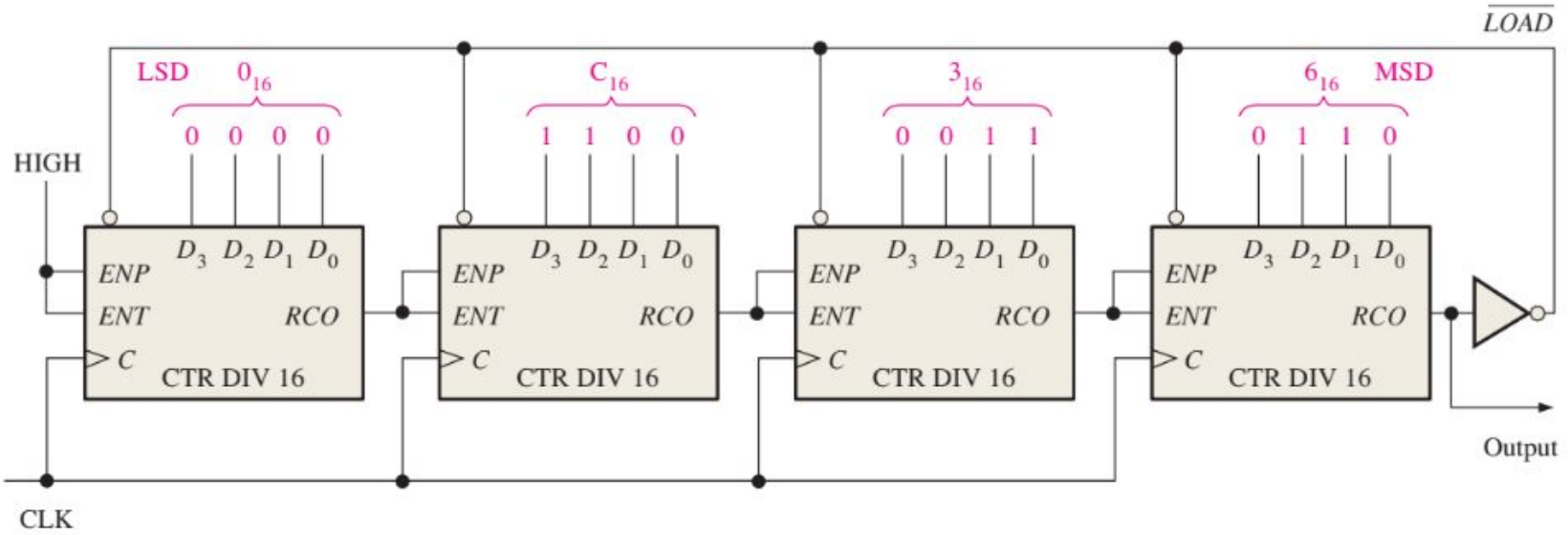


FIGURE 9–41 A divide-by-40,000 counter using 74HC161 4-bit binary counters. Note that each of the parallel data inputs is shown in binary order (the right-most bit D_0 is the LSB in each counter).

Cascaded Counters

Cascaded Counters with Truncated Sequences

Also known as full-modulus cascading.

2^16 = 65536

65536 - 40000 = 25536 (63C0 in Hex)

25536 states should be deleted.

Cascaded Counters

SECTION 9–6 CHECKUP

1. How many decade counters are necessary to implement a divide-by-1000 (modulus-1000) counter? A divide-by-10,000?
2. Show with general block diagrams how to achieve each of the following, using a flip-flop, a decade counter, and a 4-bit binary counter, or any combination of these:
   - **(a)** Divide-by-20 counter
   - **(b)** Divide-by-32 counter
   - **(c)** Divide-by-160 counter
   - **(d)** Divide-by-320 counter

Counter Decoding

Active HIGH Decoding

Active Low Decoding

Will use NAND

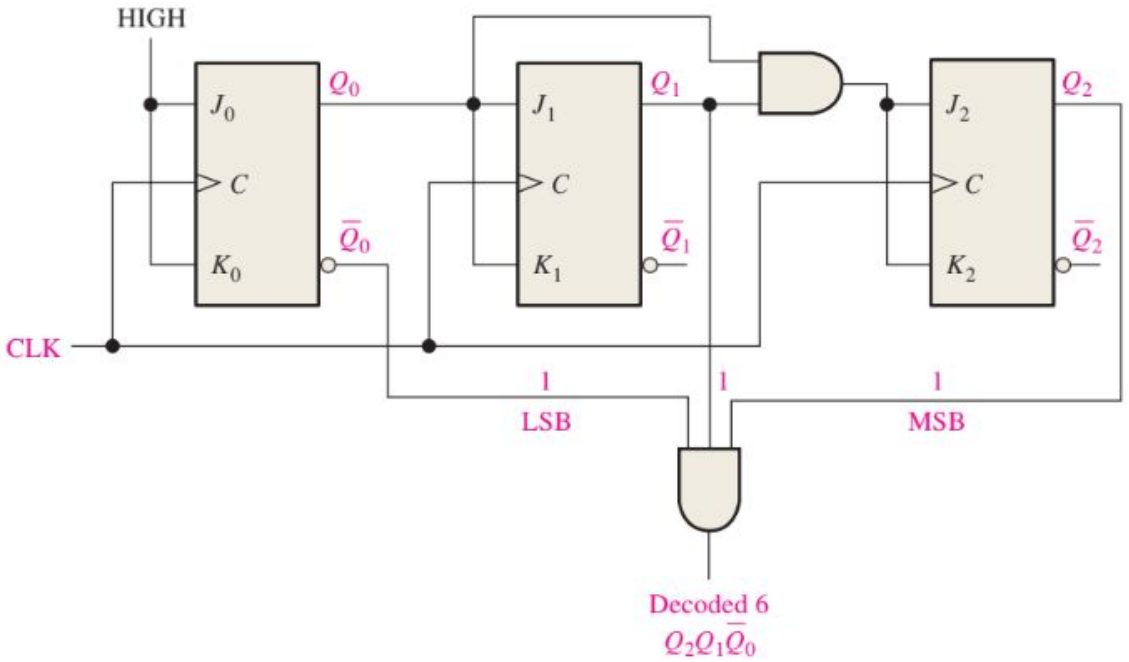


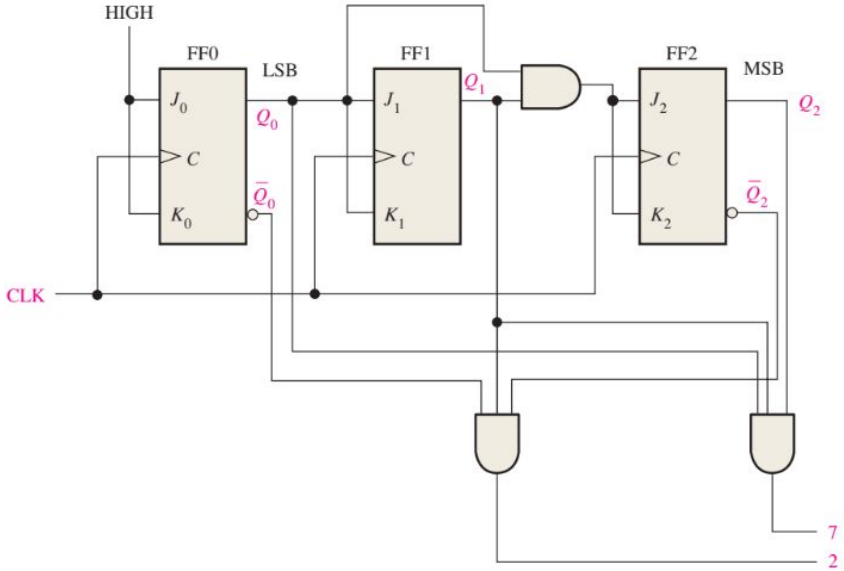
FIGURE 9–42 Decoding of state 6 (110). Open file F09-42 to verify operation.

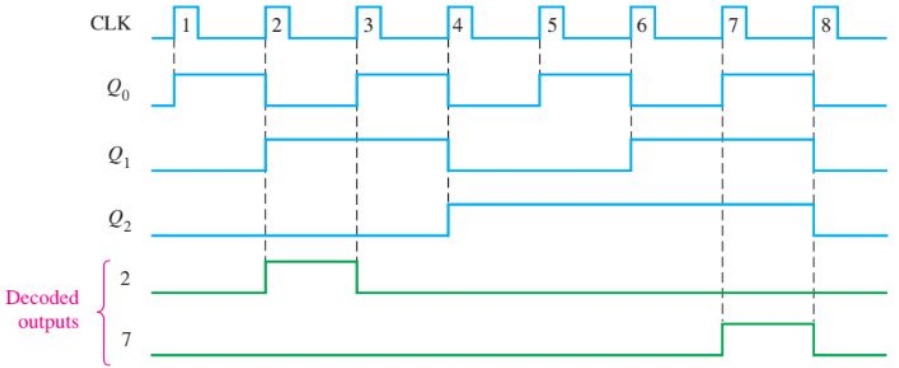
Counter Decoding

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 = $\overline{Q}_2 Q_1 \overline{Q}_0$ and binary 7 = $Q_2 Q_1 Q_0$.

Counter Decoding





Counter Decoding

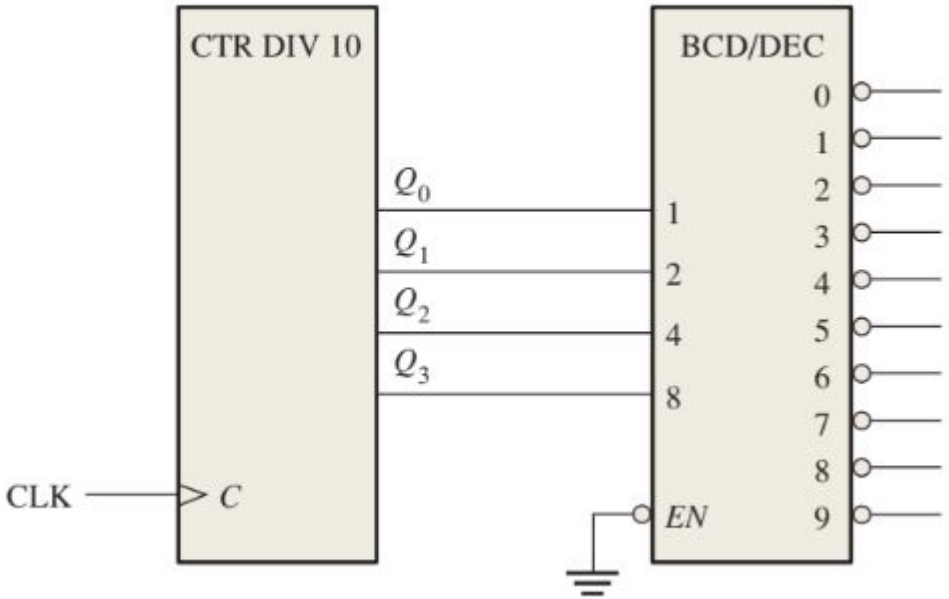
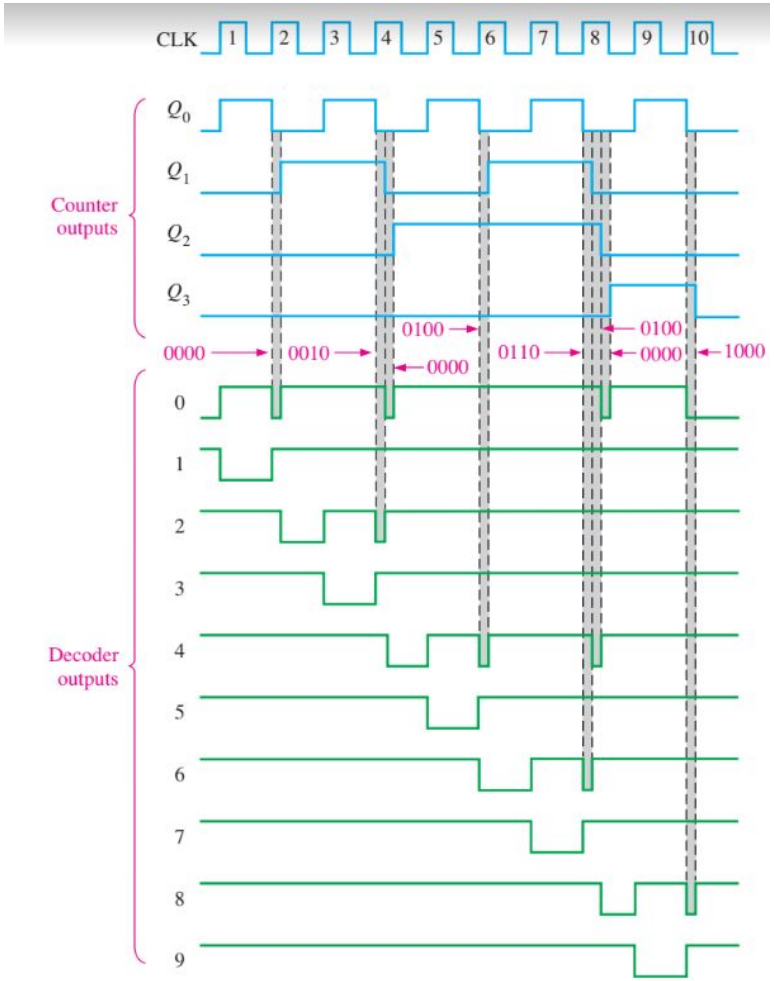


FIGURE 9–44 A basic decade (BCD) counter and decoder.

Counter Decoding



Counter Decoding ~ Strobing

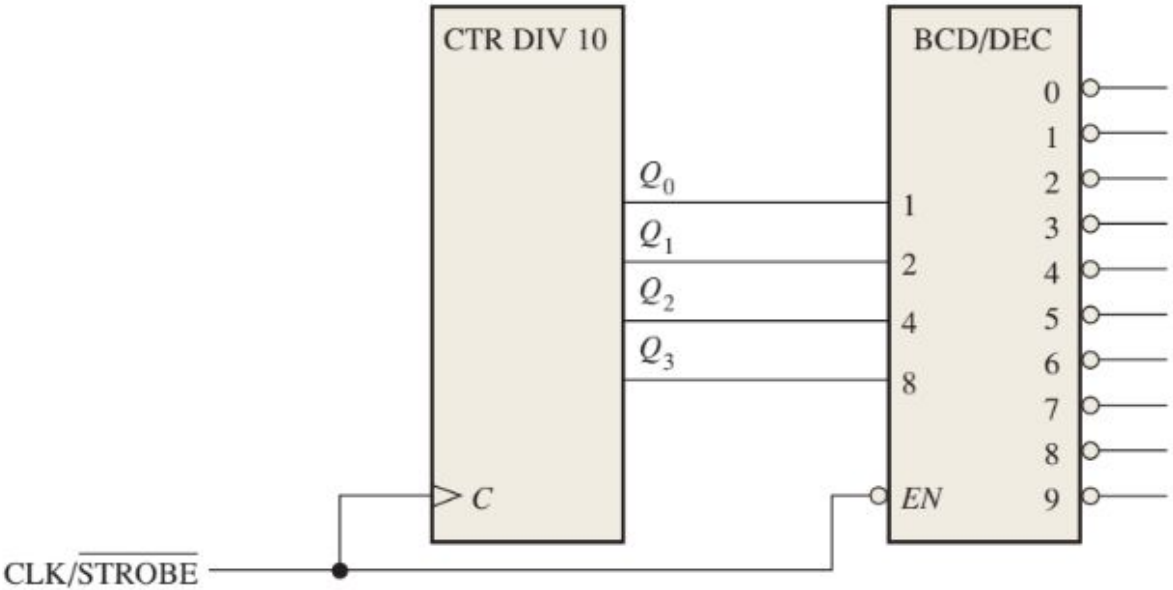


FIGURE 9–46 The basic decade counter and decoder with strobing to eliminate glitches.

Counter Decoding ~ Strobing

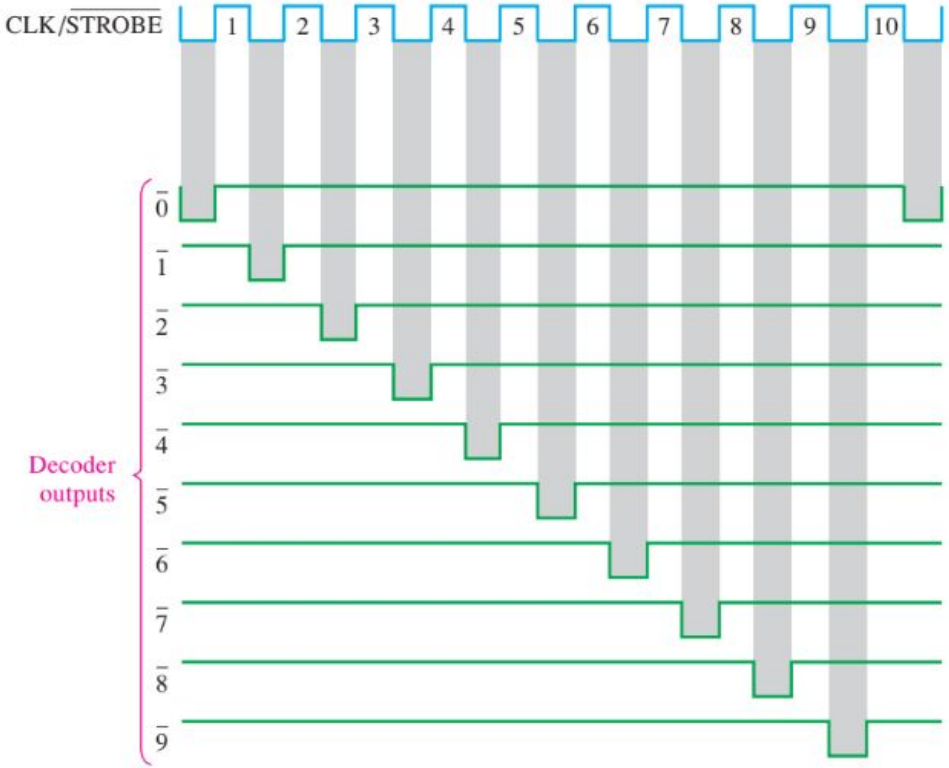
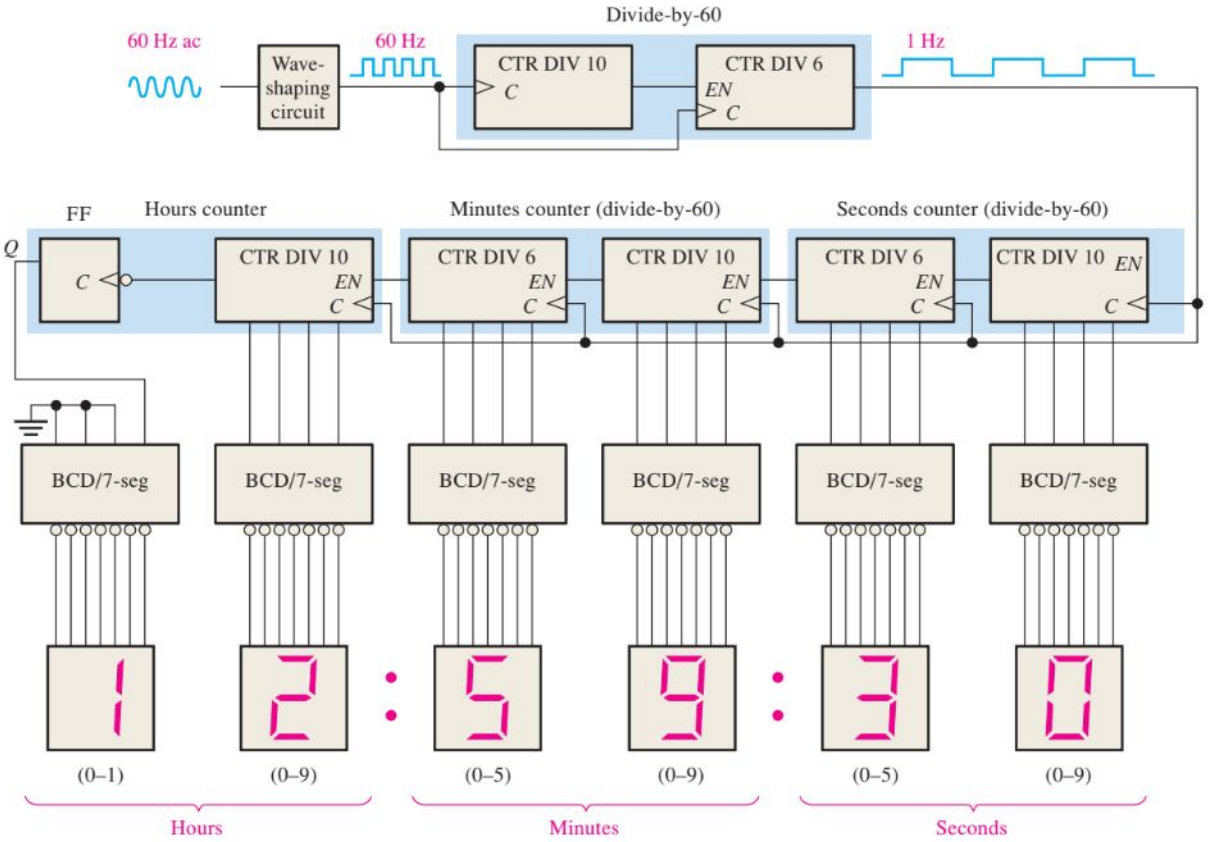


FIGURE 9–47 Strobed decoder outputs for the circuit of Figure 9–46.

Counter Applications

A Digital Clock



Finite State Machines

A state machine is a sequential circuit having a limited (finite) number of states occurring in a prescribed order.

- Moore
  - outputs depend only on the internal present state
- Mealy
  - outputs depend on both the internal present state and on the inputs

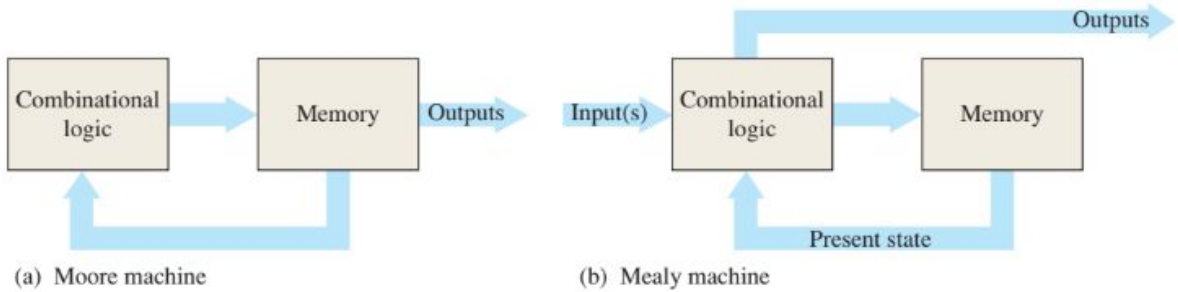


FIGURE 9–1 Two types of sequential logic.

Moore Machine

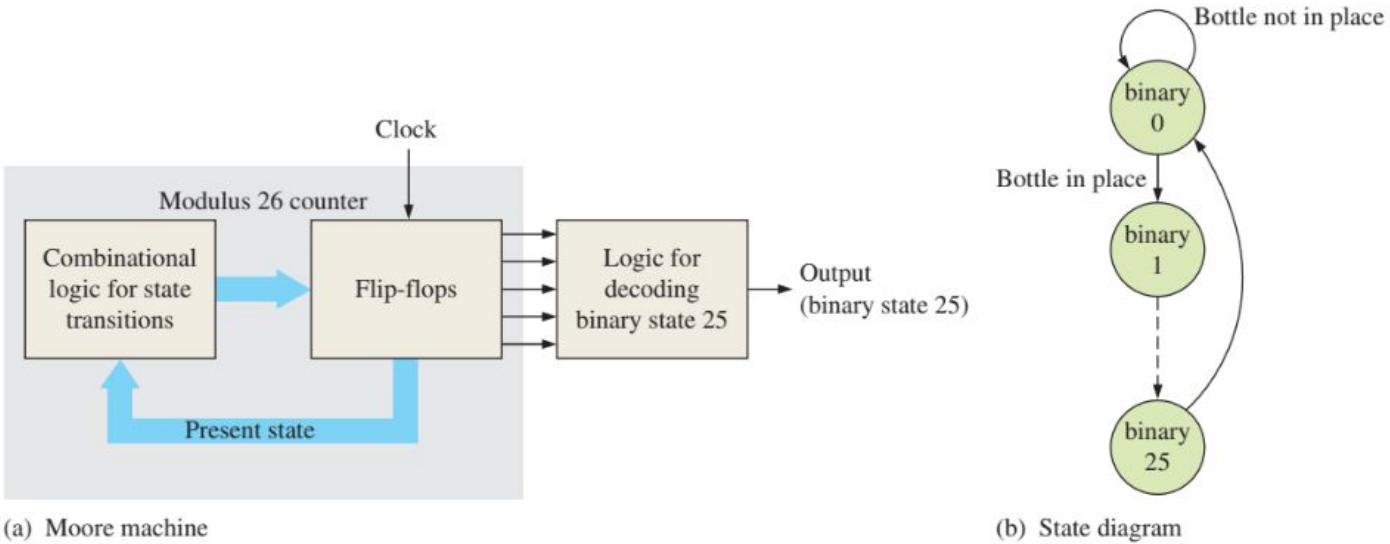


FIGURE 9–2 A fixed-modulus binary counter as an example of a Moore state machine. The dashed line in the state diagram means the states between binary 1 and 25 are not shown for simplicity.

Mealy Machine

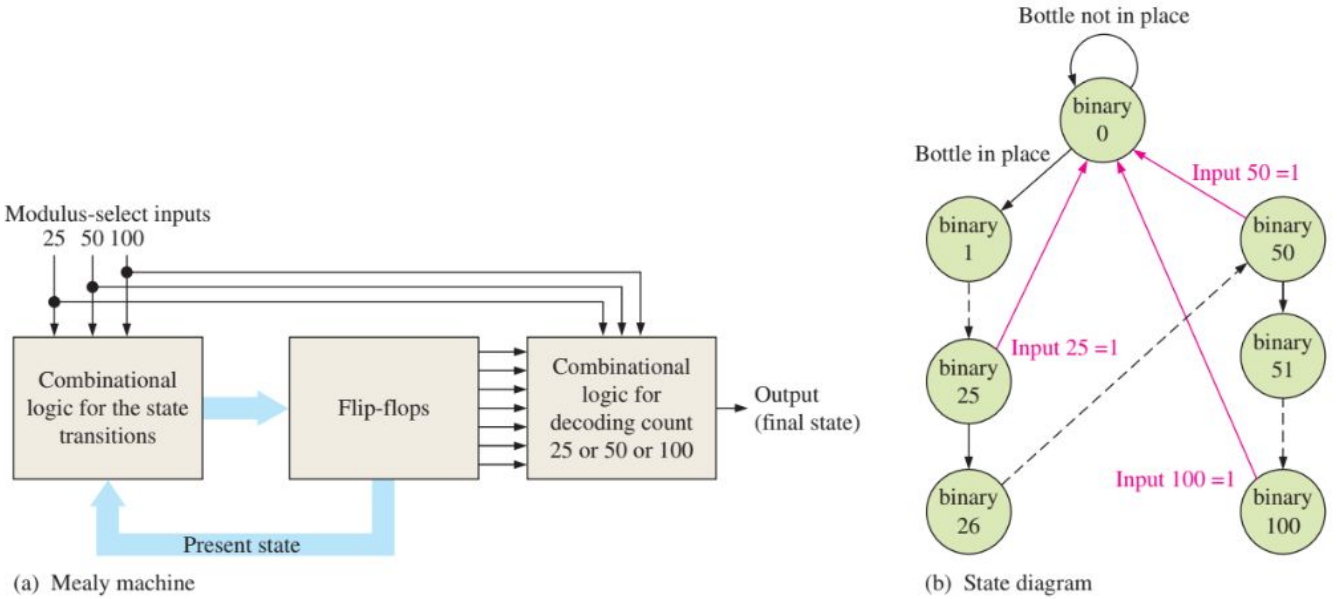


FIGURE 9–3 A variable-modulus binary counter as an example of a Mealy state machine. The red arrows in the state diagram represent the recycle paths that depend on the input number. The black dashed lines mean the interim states are not shown for simplicity.

Designing A State Machine

Step 1: State Diagram

Step 2: Next-State Table

Step 3: Flip-Flop Transition Table

Step 4: Karnaugh Maps

Step 1: State Diagram

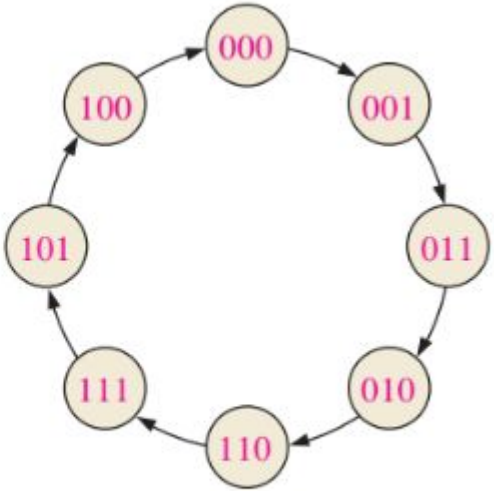


FIGURE 9–26 State diagram for a 3-bit Gray code counter.

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.

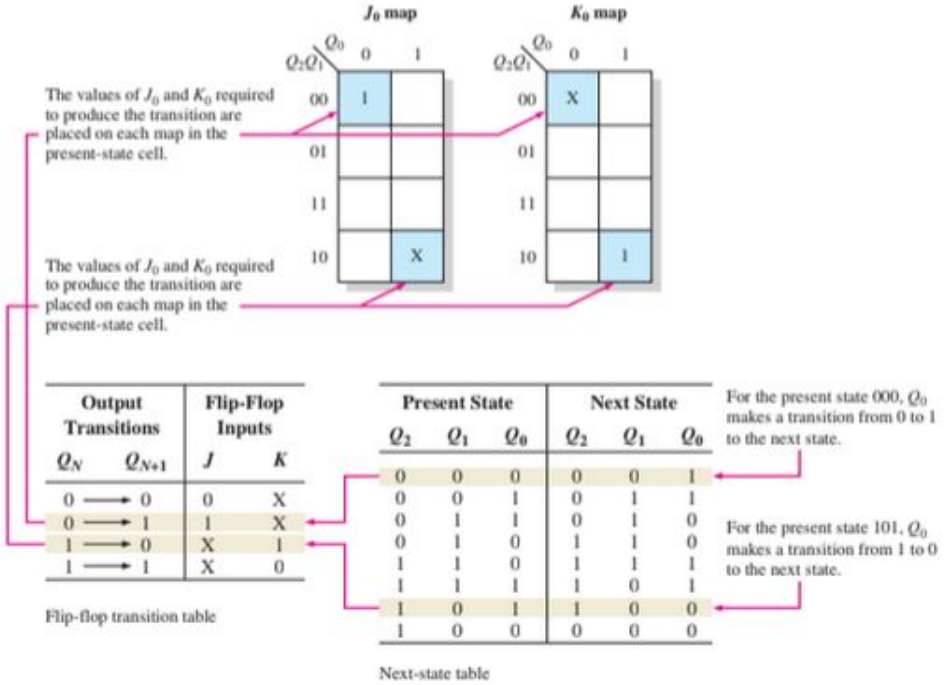
Output Transitions			Flip-Flop Inputs	
Q_N		Q_{N+1}	J	K
0	⟶	0	0	X
0	⟶	1	1	X
1	⟶	0	X	1
1	⟶	1	X	0

Q_N: present state

Q_{N+1}: next state

X: "don't care"

Step 4: Karnaugh Maps



Step 4: Karnaugh Maps

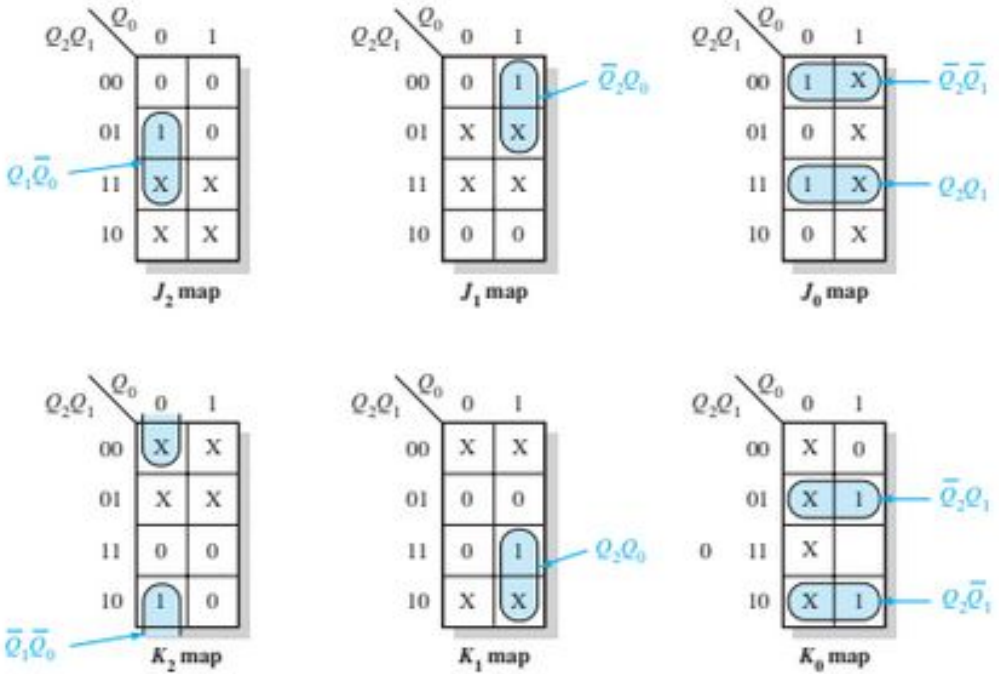


FIGURE 9–28 Karnaugh maps for present-state *J* and *K* inputs.

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

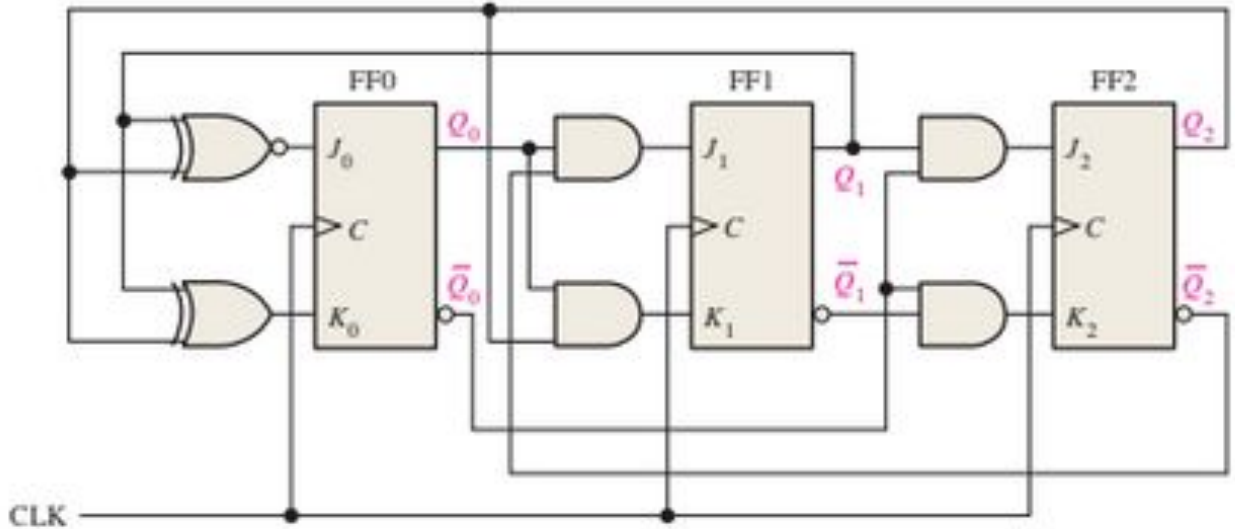


FIGURE 9–29 Three-bit Gray code counter. Open file F09-29 to verify operation.

Synchronous Counters

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.

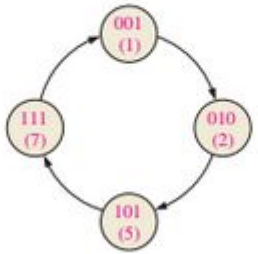


FIGURE 9–30

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	⟶	0	0
0	⟶	1	1
1	⟶	0	0
1	⟶	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.

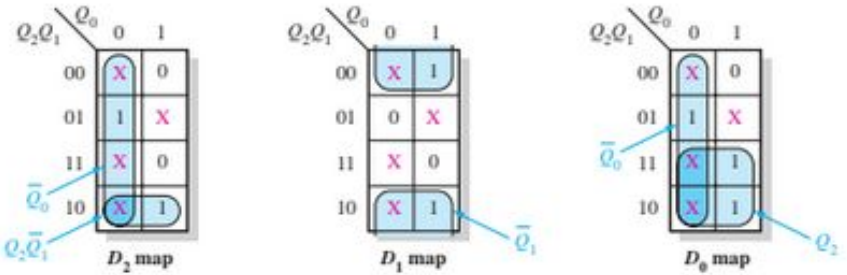


FIGURE 9–31

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.

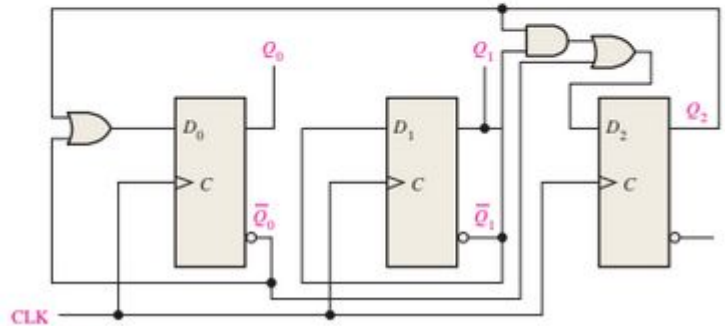


FIGURE 9–32

Synchronous Counters

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*.

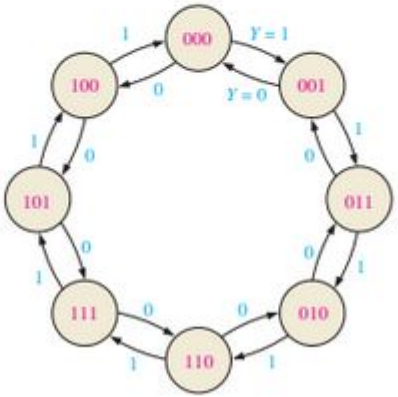


FIGURE 9–33 State diagram for a 3-bit up/down Gray code counter.

Synchronous Counters

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.

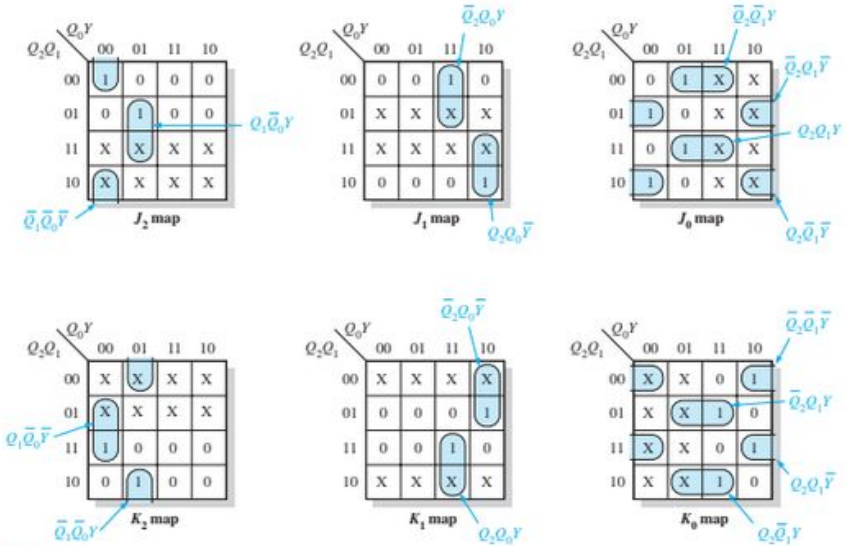


FIGURE 9–34 J and K maps for Table 9–12. The UP/$\overline{\text{DOWN}}$ control input, Y, is treated as a fourth variable.

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.

Related Problem

Specify the number of flip-flops, gates, and inverters that are required to implement the logic described in Step 5.

Practice Questions

Section 9–5 Design of Synchronous Counters

18. Determine the sequence of the counter in Figure 9–73.

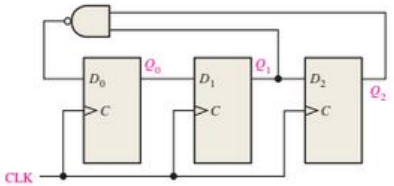


FIGURE 9–73

19. Determine the sequence of the counter in Figure 9–74. Begin with the counter cleared.

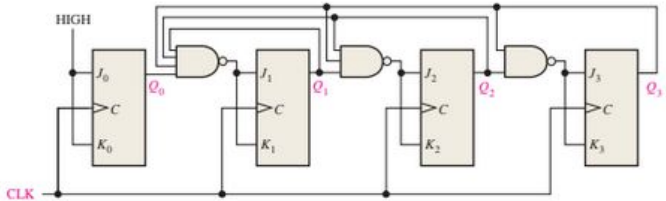


FIGURE 9–74

20. Design a counter to produce the following sequence. Use J-K flip-flops.

00, 10, 01, 11, 00, . . .

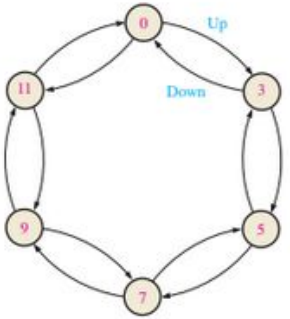
21. Design a counter to produce the following binary sequence. Use J-K flip-flops.

1, 4, 3, 5, 7, 6, 2, 1, . . .

22. Design a counter to produce the following binary sequence. Use J-K flip-flops.

0, 9, 1, 8, 2, 7, 3, 6, 4, 5, 0, . . .

23. Design a binary counter with the sequence shown in the state diagram of Figure 9–75.



Counter Applications

Automobile

Parking Control

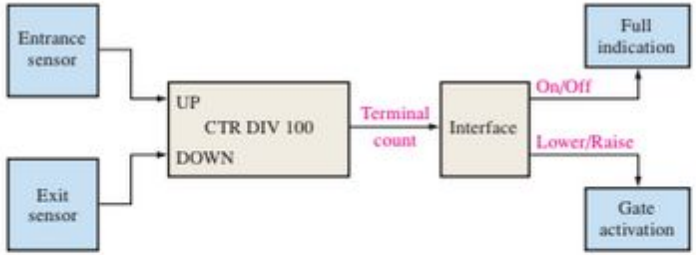


FIGURE 9–51 Functional block diagram for parking garage control.

A logic diagram of the up/down counter is shown in Figure 9–52. It consists of two cascaded up/down decade counters. The operation is described in the following paragraphs.

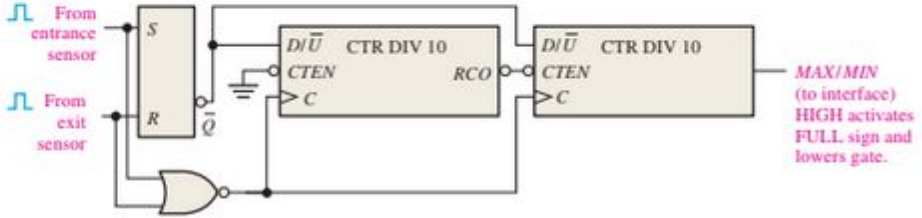


FIGURE 9–52 Logic diagram for modulus-100 up/down counter for automobile parking control.

Counter Applications

Parallel-to-Serial

Data Conversion

(Multiplexing)

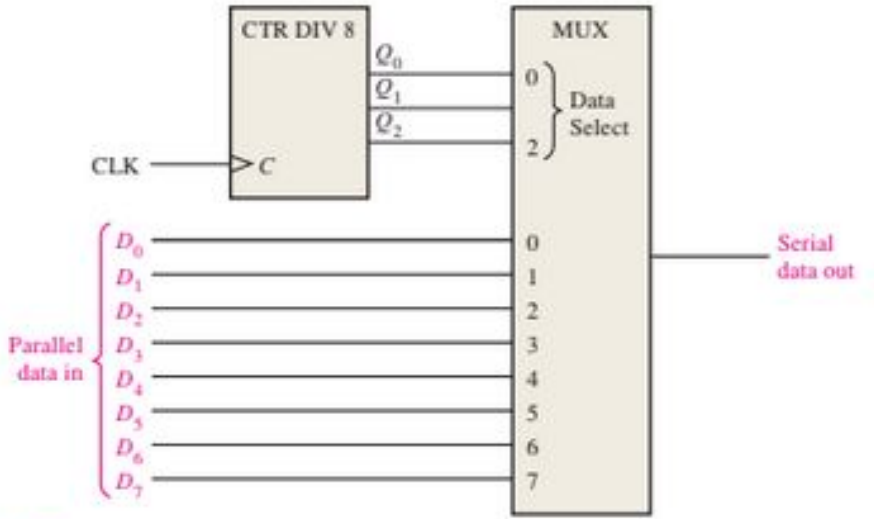


FIGURE 9–53 Parallel-to-serial data conversion logic.

End is the new Beginning

Best of luck for your future

It's been a nice journey with you guys