به نام خدا

فصل هفتم:

Latchs, Flip Flops & Timers

7 LATCHES, FLIP-FLOPS, AND TIMERS

CHAPTER OUTLINE

7–1 Latches
7–2 Edge-Triggered Flip-Flops
7–3 Flip-Flop Operating Characteristics
7–4 Flip-Flop Applications
7–5 One-Shots
7–6 The 555 Timer
7–7 Troubleshooting
Digital System Application

CHAPTER OBJECTIVES

- Use logic gates to construct basic latches
- Explain the difference between an S-R latch and a D latch
- Recognize the difference between a latch and a flip-flop
- Explain how S-R, D, and J-K flip-flops differ
- Understand the significance of propagation delays, set-up time, hold time, maximum operating frequency, minimum clock pulse widths, and power dissipation in the application of flip-flops
- Apply flip-flops in basic applications
- Explain how retriggerable and nonretriggerable one-shots differ
- Connect a 555 timer to operate as either an astable multivibrator or a one-shot
- Troubleshoot basic flip-flop circuits

KEY TERMS

- Latch
- Bistable
- SET
- RESET
- Clock
- Edge-triggered flip-flop
- Synchronous
- D flip-flop
- J-K flip-flop
- Toggle
- Preset
- Clear
- Propagation delay time
- Set-up time
- Hold time
- Power dissipation
- One-shot
- Monostable
- Timer
- Astable

INTRODUCTION

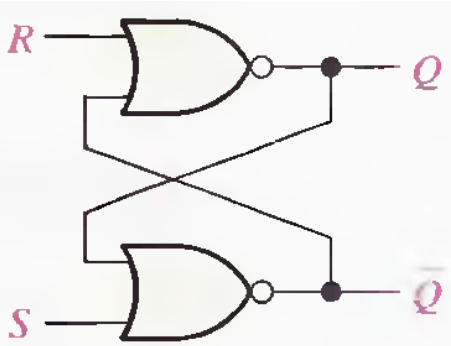
This chapter begins a study of the fundamentals of sequential logic. Bistable, monostable, and astable logic devices called *multivibrators* are covered. Two categories of bistable devices are the latch and the flip-flop. Bistable devices have two stable states, called SET and RESET; they can retain either of these states indefinitely, making them useful as storage devices. The basic difference between latches and flip-flops is the way in which they are changed from one state to the other. The flip-flop is a basic building block for counters, registers, and other sequential control logic and is used in certain types of memories. The monostable multivibrator, commonly known as the one-shot, has only one stable state. A one-shot produces a single controlled-width pulse when activated or triggered. The astable multivibrator has no stable state and is used primarily as an oscillator, which is a self-sustained waveform generator. Pulse oscillators are used as the sources for timing waveforms in digital systems.

7–1 LATCHES

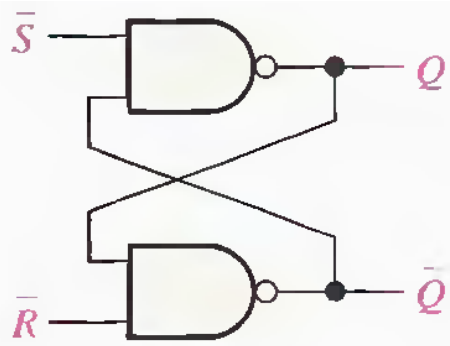
The **latch** is a type of temporary storage device that has two stable states (bistable) and is normally placed in a category separate from that of flip-flops. Latches are similar to flip-flops because they are bistable devices that can reside in either of two states using a feedback arrangement, in which the outputs are connected back to the opposite inputs. The main difference between latches and flip-flops is in the method used for changing their state.

The S-R (SET-RESET) Latch

A latch is a type of **bistable** logic device or **multivibrator.**



(a) Active-HIGH input S-R latch

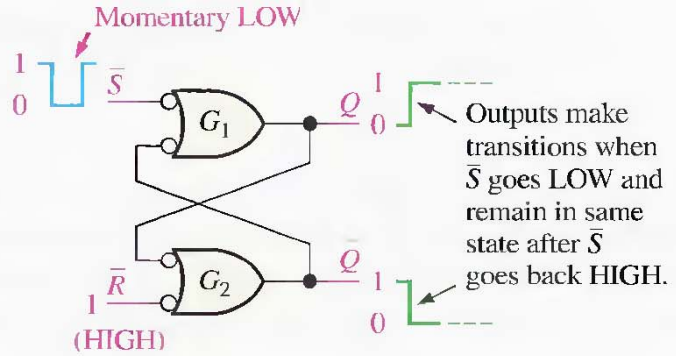


(b) Active-LOW input $\bar{S}$-$\bar{R}$ latch

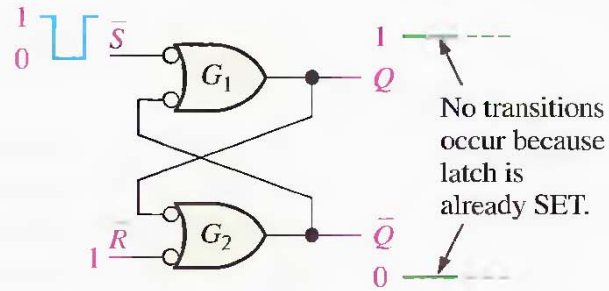
A latch can reside in either of its two states, SET or RESET.

SET means that the *Q* output is HIGH.

RESET means that the *Q* output is LOW.

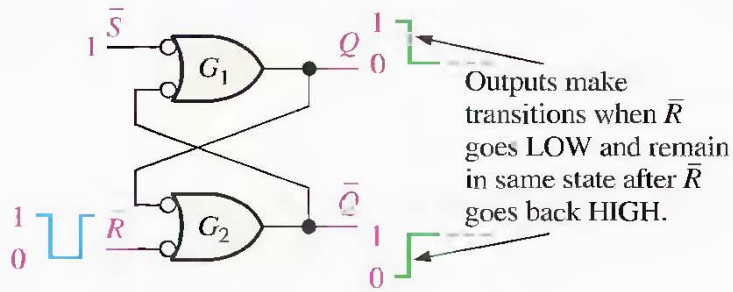


Latch starts out RESET ($Q = 0$).

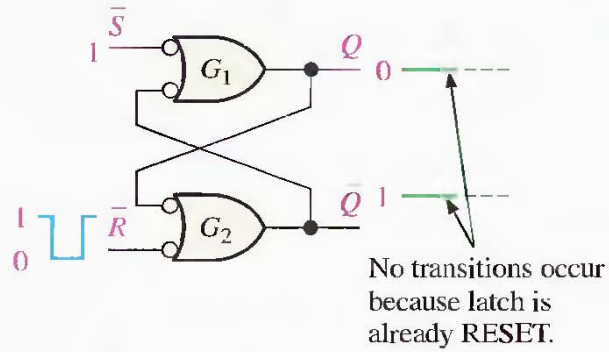


Latch starts out SET ($Q = 1$).

(a) Two possibilities for the SET operation

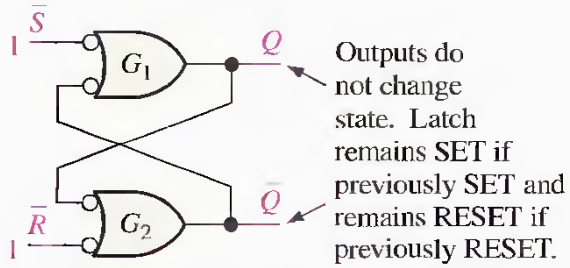


Latch starts out SET ($Q = 1$).



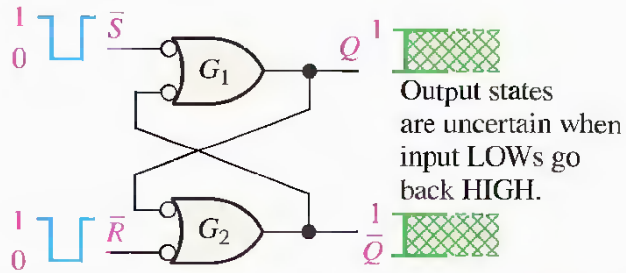
Latch starts out RESET ($Q = 0$).

(b) Two possibilities for the RESET operation



HIGHS on both inputs

(c) No-change condition



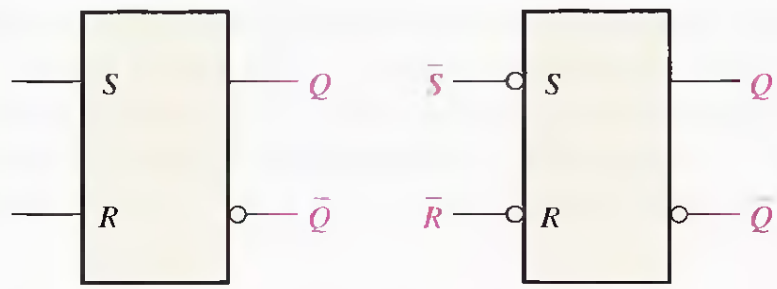
Simultaneous LOWs on both inputs

(d) Invalid condition

TABLE 7–1

Truth table for an active-LOW input $\overline{S}$-$\overline{R}$ latch.

INPUTS		OUTPUTS		COMMENTS
$\overline{S}$	$\overline{R}$	Q	$\overline{Q}$	
1	1	NC	NC	No change. Latch remains in present state.
0	1	1	0	Latch SET.
1	0	0	1	Latch RESET.
0	0	1	1	Invalid condition

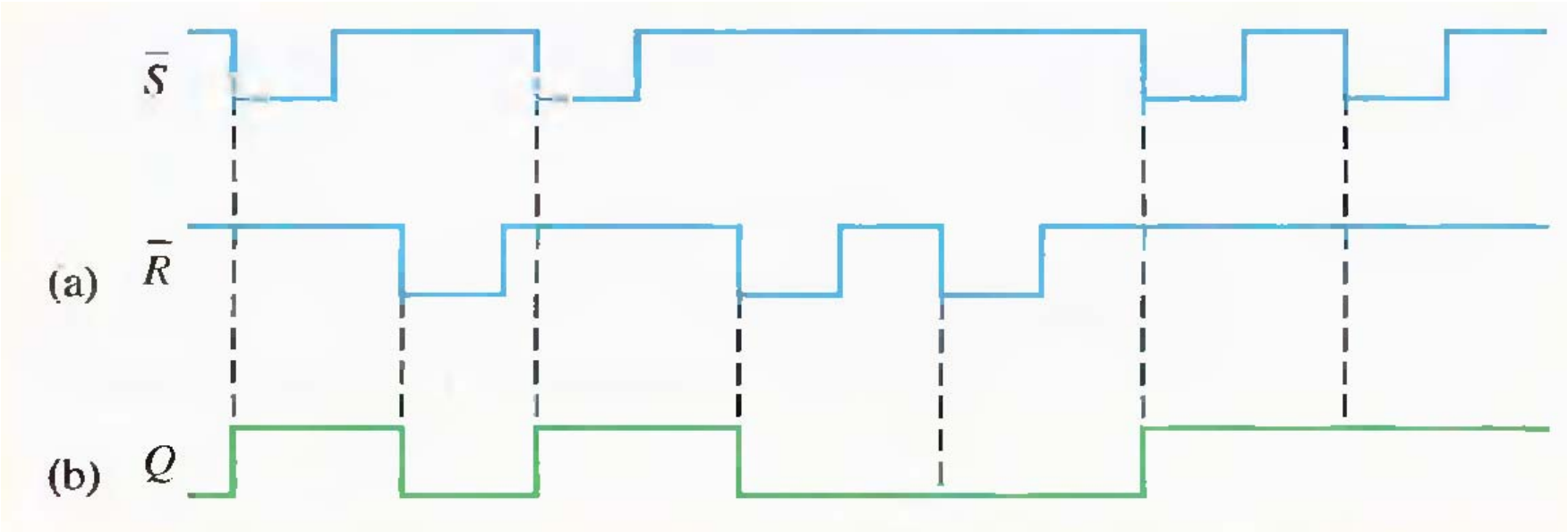


(a) Active-HIGH input S-R latch

(b) Active-LOW input $\overline{S}$-$\overline{R}$ latch

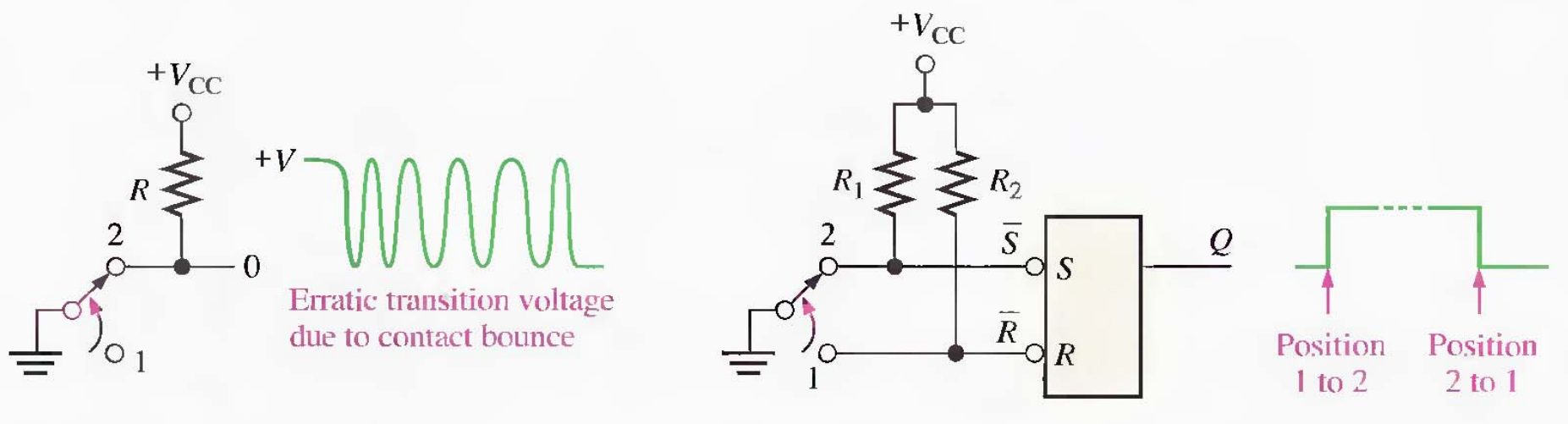
EXAMPLE 7–1

If the $\overline{S}$ and $\overline{R}$ waveforms in Figure 7–5(a) are applied to the inputs of the latch in Figure 7–4(b), determine the waveform that will be observed on the Q output. Assume that Q is initially LOW.



An Application

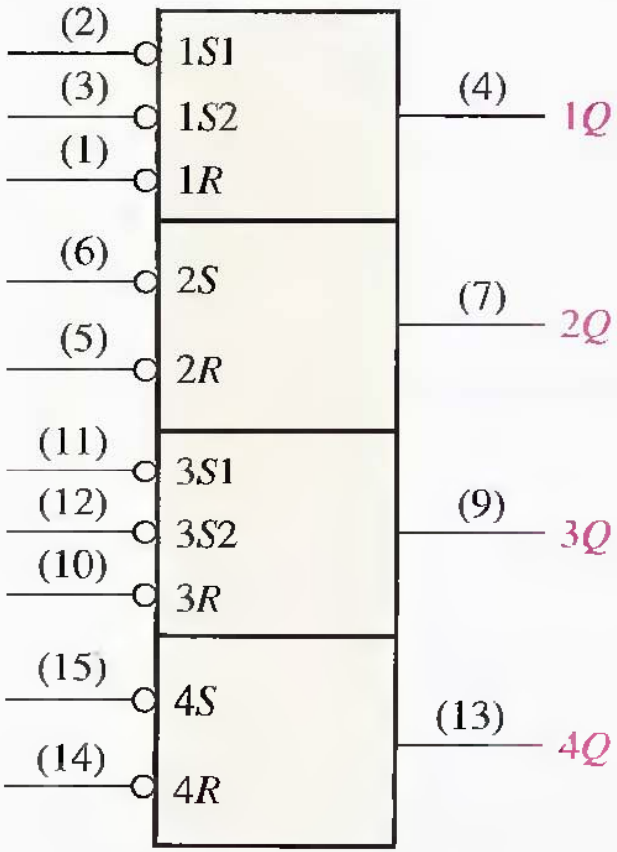
The Latch as a Contact-Bounce Eliminator



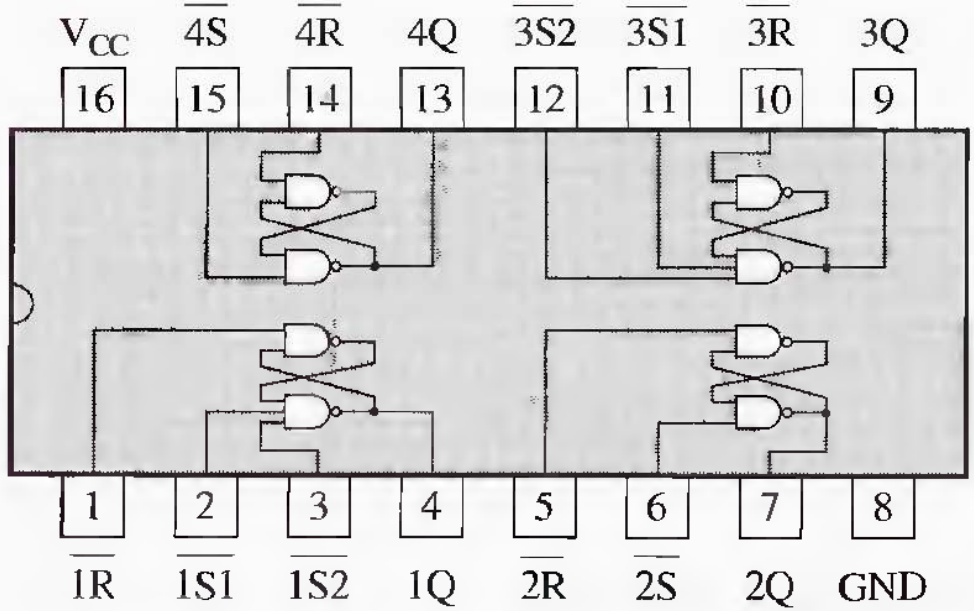
(a) Switch contact bounce

(b) Contact-bounce eliminator circuit

THE 74LS279 SET-RESET LATCH

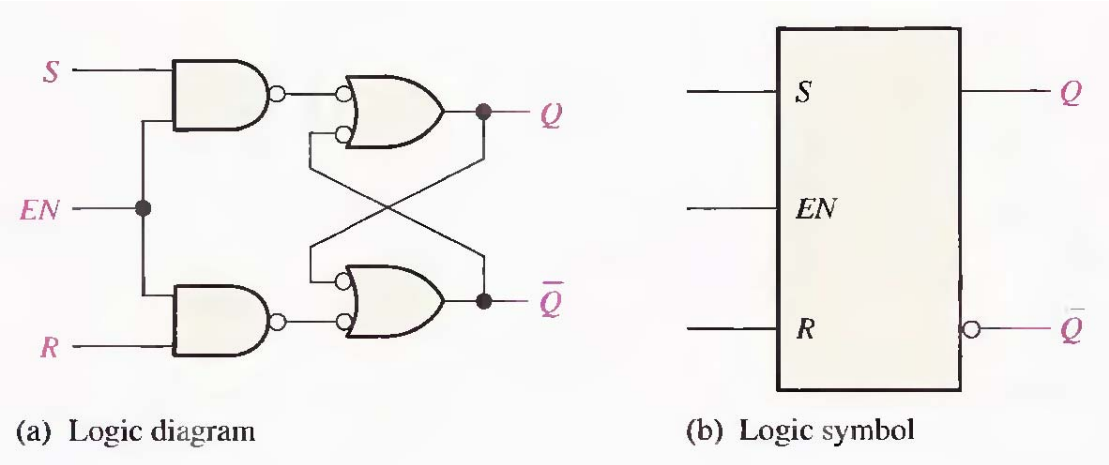


(a) Logic diagram



(b) Pin diagram

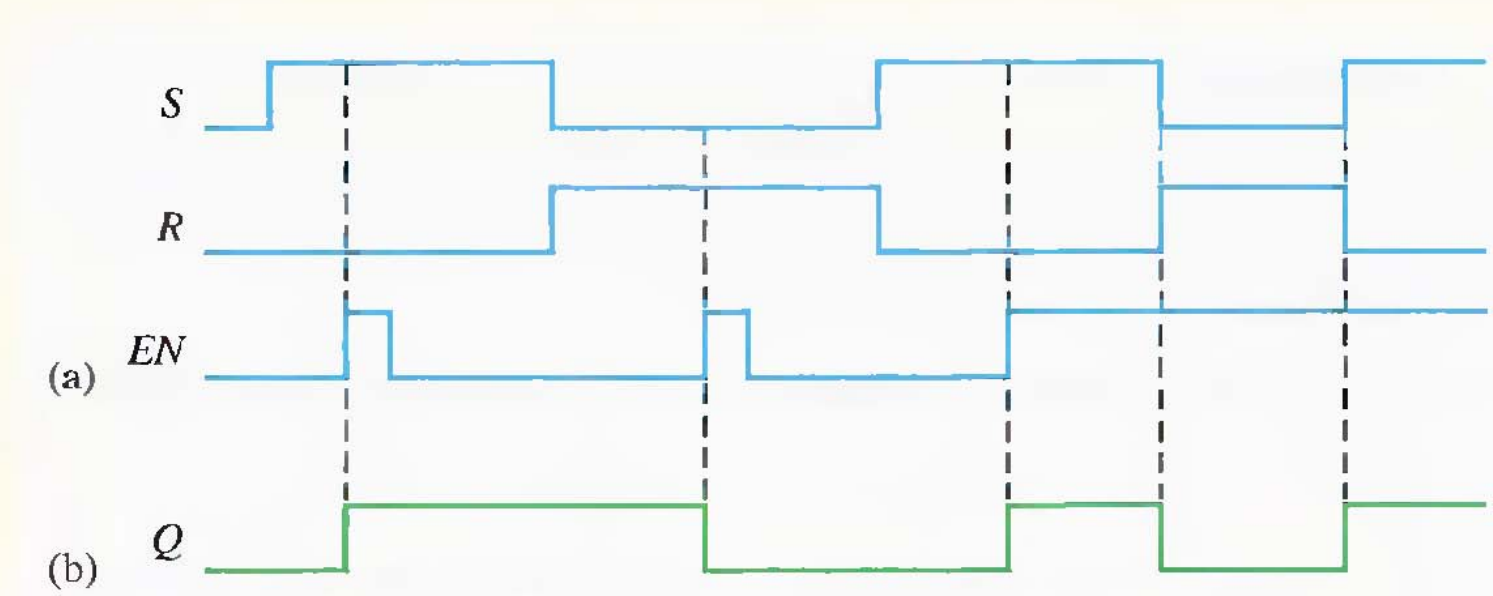
The Gated S-R Latch



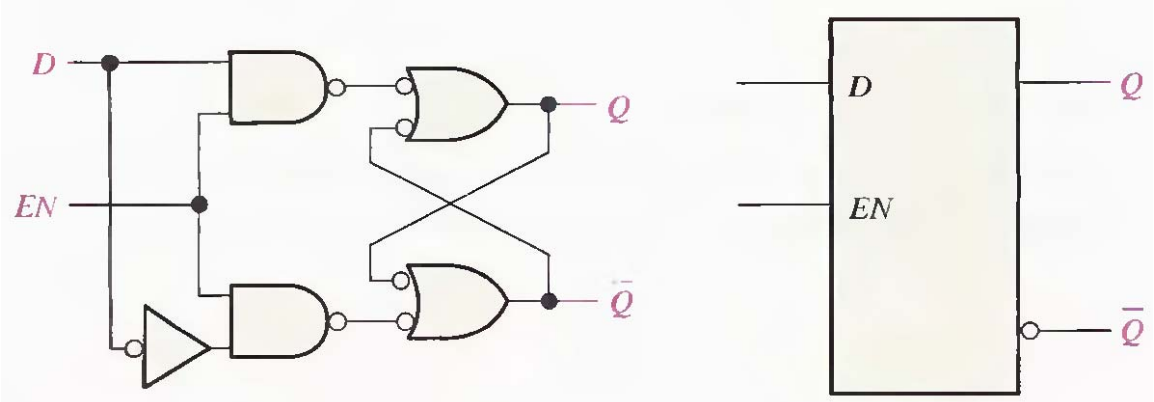
(a) Logic diagram (b) Logic symbol

EXAMPLE 7–2

Determine the Q output waveform if the inputs shown in Figure 7–9(a) are applied to a gated S-R latch that is initially RESET.



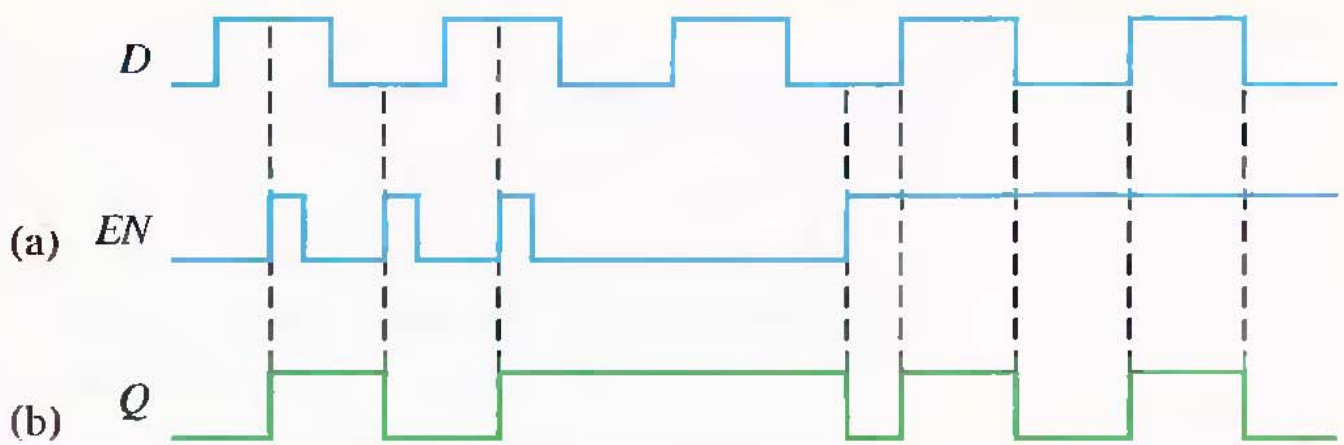
The Gated D Latch



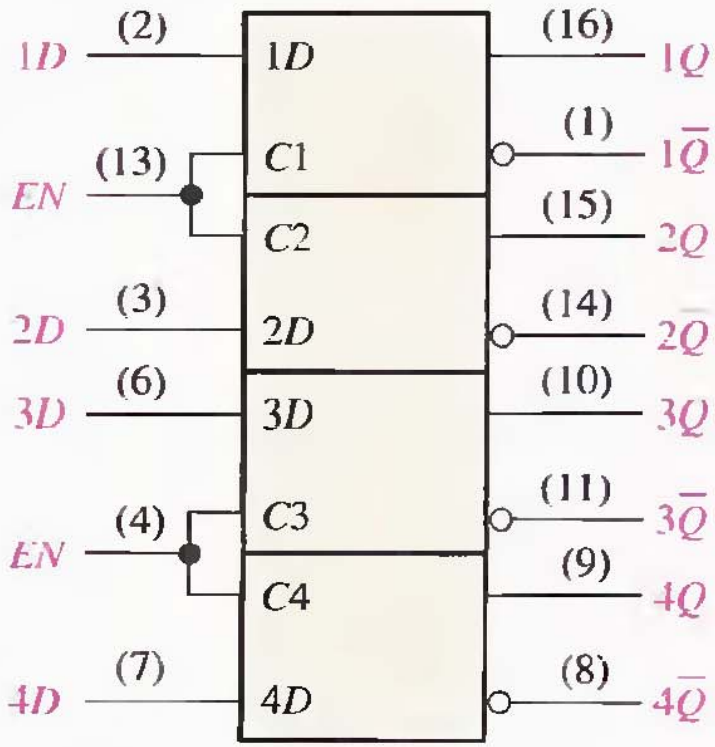
(a) Logic diagram (b) Logic symbol

EXAMPLE 7–3

Determine the Q output waveform if the inputs shown in Figure 7–11(a) are applied to a gated D latch, which is initially RESET.



THE 74LS75 D LATCH



(a) Logic symbol

Inputs		Outputs		
D	EN	Q	$\overline{Q}$	Comments
0	1	0	1	RESET
1	1	1	0	SET
X	0	Q_0	$\overline{Q}_0$	No change

Note: Q_0 is the prior output level before the indicated input conditions were established.

(b) Truth table (each latch)

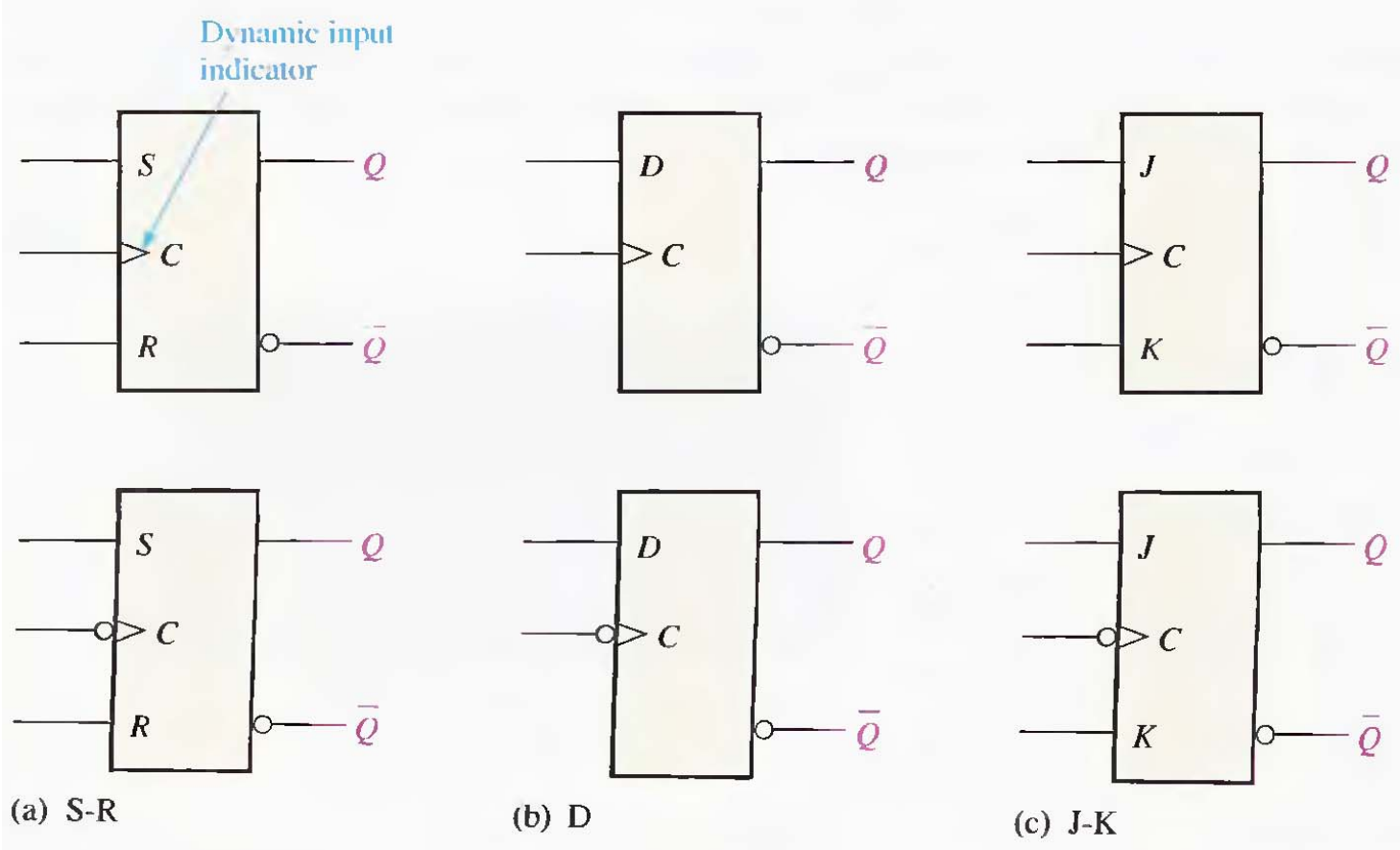
SECTION 7-1 REVIEW

Answers are at the end of the chapter.

1. List three types of latches.
2. Develop the truth table for the active-HIGH input S-R latch in Figure 7–1(a).
3. What is the Q output of a D latch when $EN = 1$ and $D = 1$?

7–2 EDGE-TRIGGERED FLIP-FLOPS

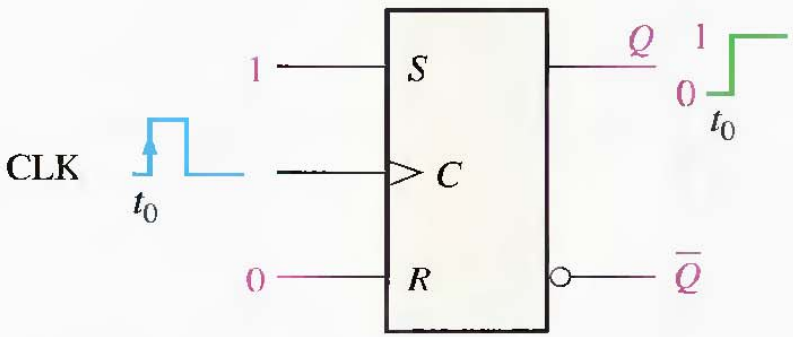
Flip-flops are synchronous bistable devices, also known as *bistable multivibrators*. In this case, the term *synchronous* means that the output changes state only at a specified point on the triggering input called the **clock** (CLK), which is designated as a control input, *C;* that is, changes in the output occur in synchronization with the clock.



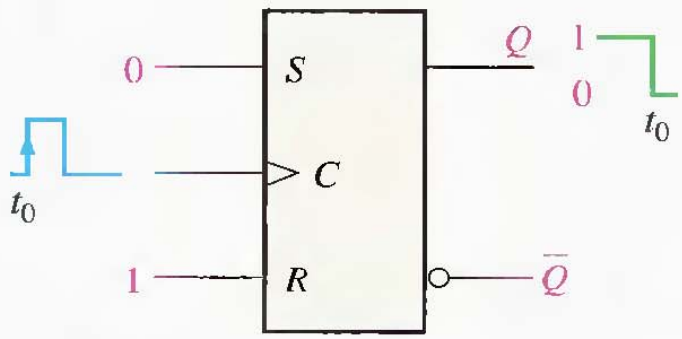
(a) S-R (b) D (c) J-K

The Edge-Triggered S-R Flip-Flop

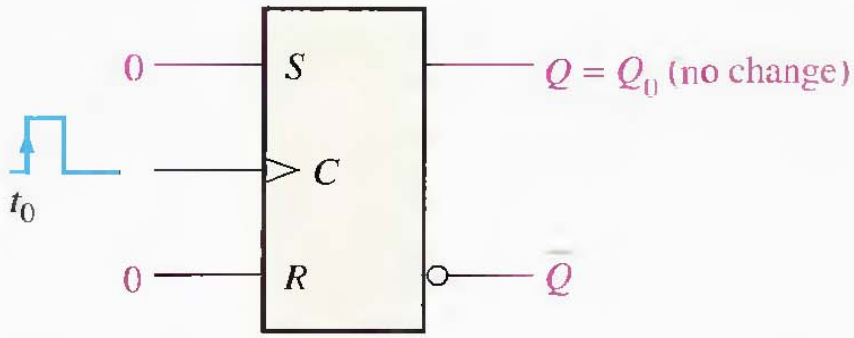
The S and R inputs of the **S-R flip-flop** are called **synchronous** inputs because data on these inputs are transferred to the flip-flop's output only on the triggering edge of the clock pulse.



(a) $S = 1$, $R = 0$ flip-flop SETS on positive clock edge. (If already SET, it remains SET.)



(b) $S = 0$, $R = 1$ flip-flop RESETS on positive clock edge. (If already RESET, it remains RESET.)



(c) $S = 0$, $R = 0$ flip-flop does not change. (If SET, it remains SET; if RESET, it remains RESET.)

Operation of a positive edge-triggered S-R flip-flop.

INPUTS			OUTPUTS		
S	R	CLK	Q	$\overline{Q}$	COMMENTS
0	0	X	Q_0	$\overline{Q}_0$	No change
0	1	↑	0	1	RESET
1	0	↑	1	0	SET
1	1	↑	?	?	Invalid

↑ = clock transition LOW to HIGH

X = irrelevant ("don't care")

Q_0 = output level prior to clock transition

Remember, *the flip-flop cannot change state except on the triggering edge of a clock pulse.*

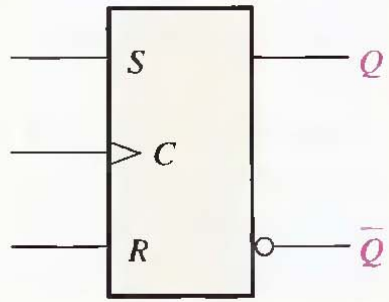
COMPUTER NOTE

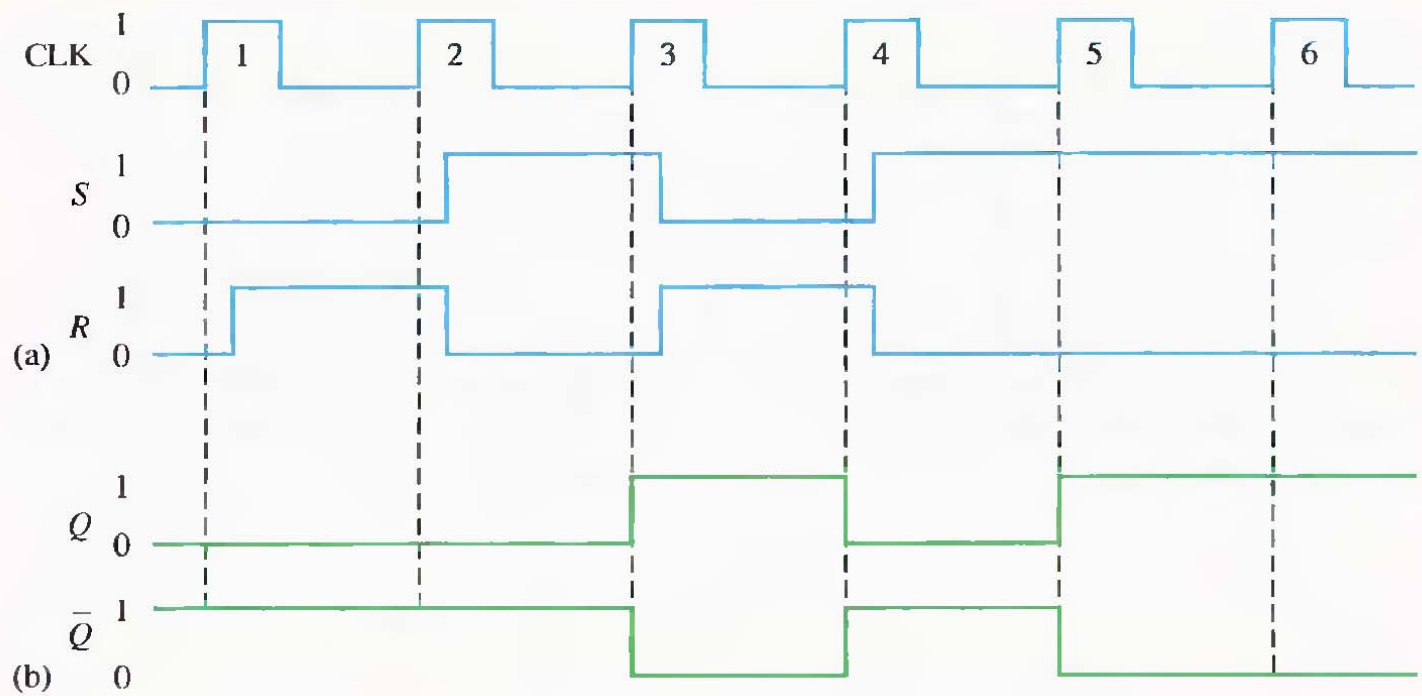
Semiconductor memories in computers consist of large numbers of individual cells. Each storage cell holds a 1 or a 0. One type of memory is the Static Random Access Memory or SRAM, which uses flip-flops for the storage cells because a flip-flop will retain either of its two states indefinitely as long as dc power is applied, thus the term *static*. This type of memory is classified as a *volatile* memory because all the stored data are lost when power is turned off. Another type of memory, the Dynamic Random Access Memory or DRAM, uses capacitance rather than flip-flops as the basic storage element and must be periodically refreshed in order to maintain the stored data.

EXAMPLE 7–4

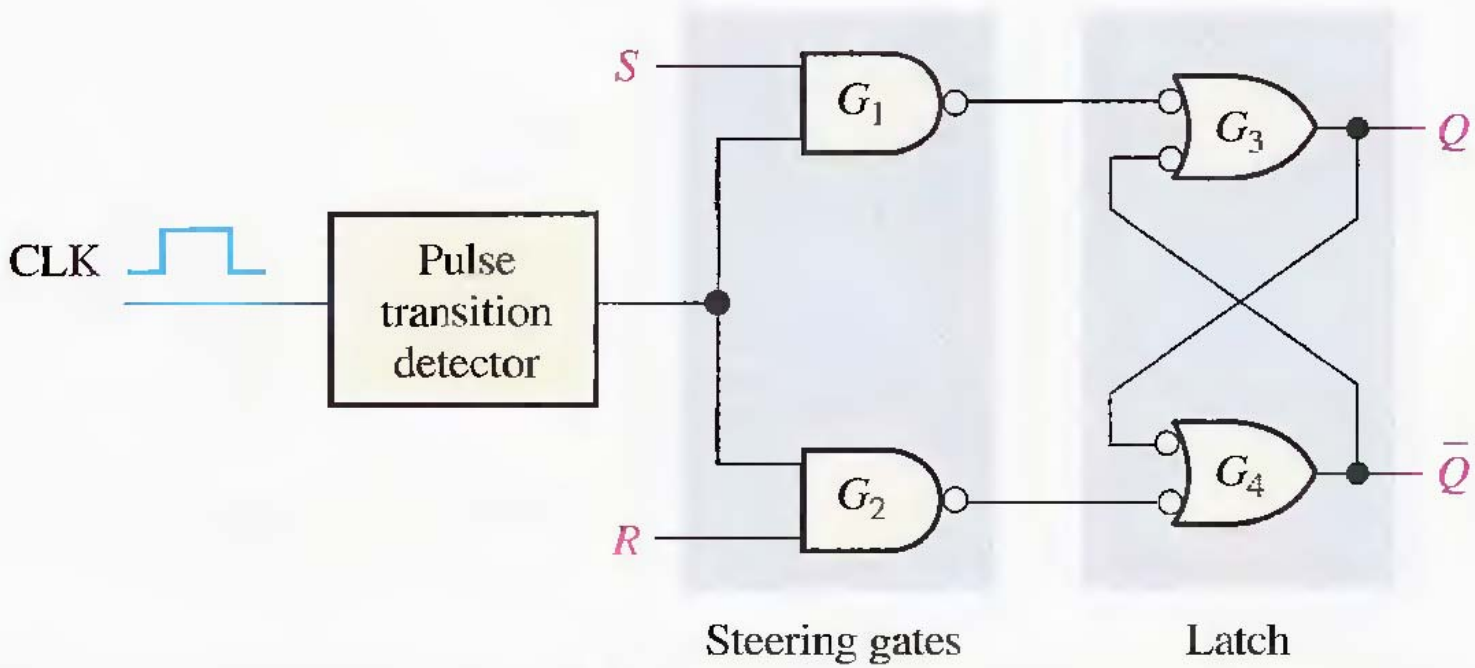
Determine the Q and $\overline{Q}$ output waveforms of the flip-flop in Figure 7–15 for the S, R, and CLK inputs in Figure 7–16(a). Assume that the positive edge-triggered flip-flop is initially RESET.

FIGURE 7–15

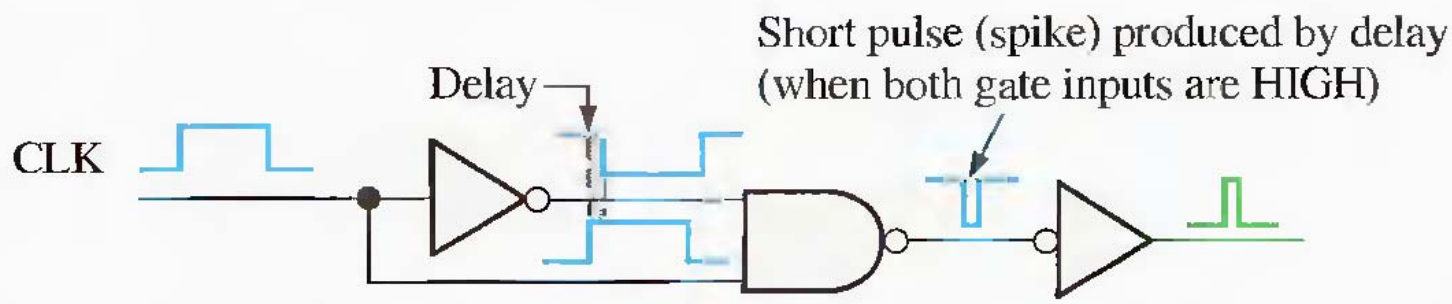




A Method of Edge-Triggering



(a) A simplified logic diagram for a positive edge-triggered S-R flip-flop



(b) A type of pulse transition detector

FIGURE 7–18

Flip-flop making a transition from the RESET state to the SET state on the positive-going edge of the clock pulse.

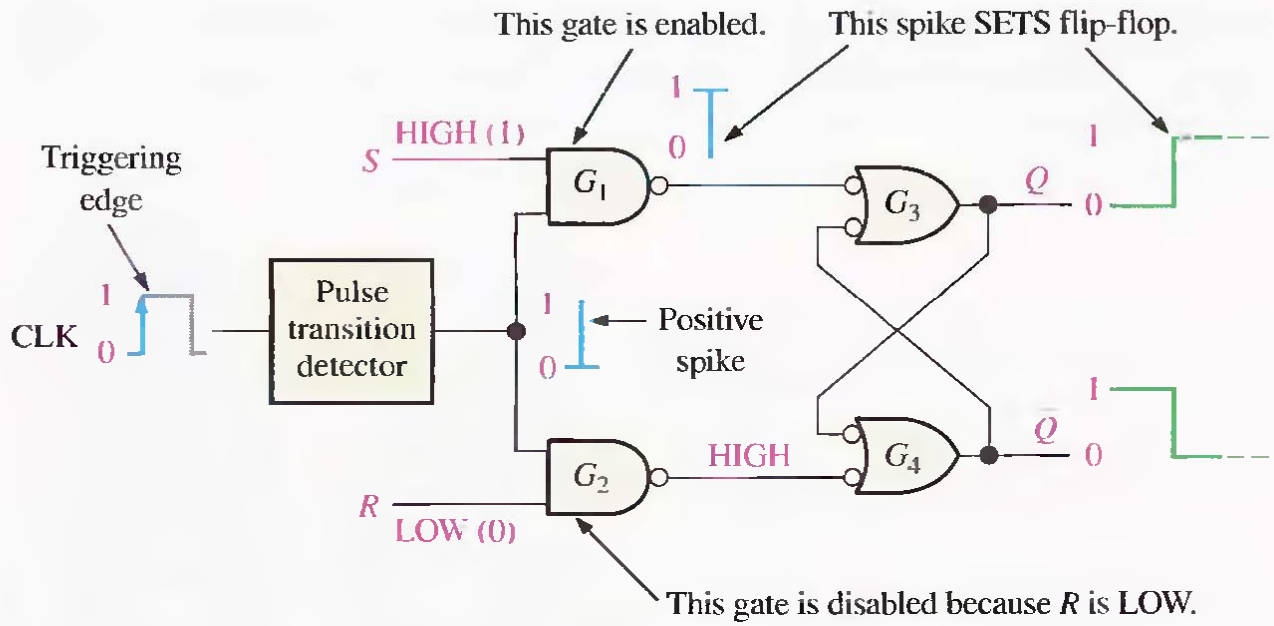
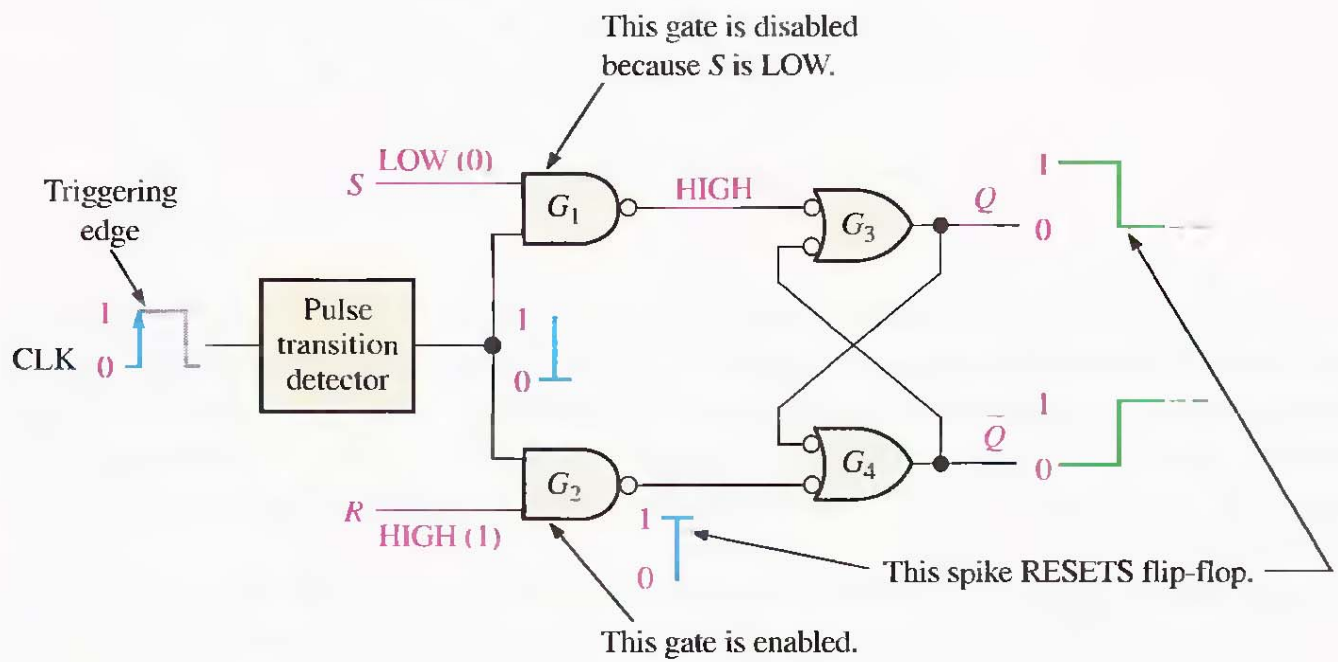


FIGURE 7–19

Flip-flop making a transition from the SET state to the RESET state on the positive-going edge of the clock pulse.



COMPUTER NOTE

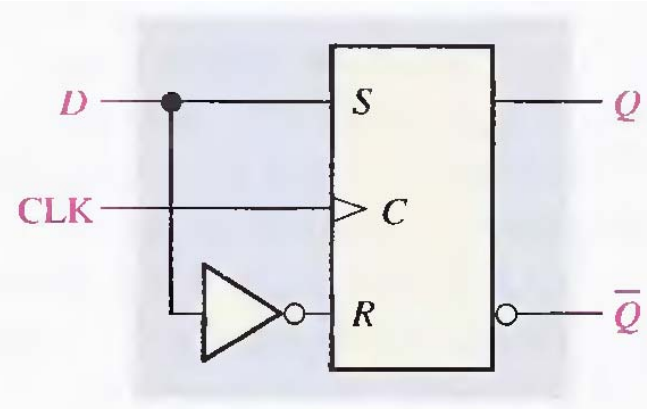
All logic operations that are performed with hardware can also be implemented in software. For example, the operation of a J-K flip-flop can be performed with specific computer instructions. If two bits were used to represent the *J* and *K* inputs, the computer would do nothing for 00, a data bit representing the *Q* output would be set (1) for 10, the *Q* data bit would be cleared (0) for 01, and the *Q* data bit would be complemented for 11. Although it may be unusual to use a computer to simulate a flip-flop, the point is that all hardware operations can be simulated using software.

The Edge-Triggered D Flip-Flop

The **D flip-flop** is useful when a single data bit (1 or 0) is to be stored.

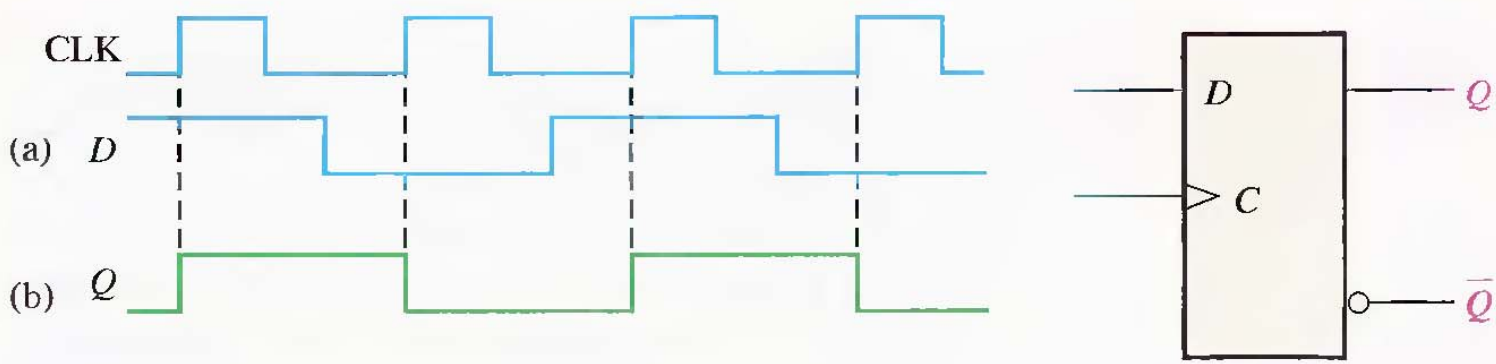
INPUTS		OUTPUTS		
D	CLK	Q	$\overline{Q}$	COMMENTS
1	↑	1	0	SET (stores a 1)
0	↑	0	1	RESET (stores a 0)

↑ = clock transition LOW to HIGH



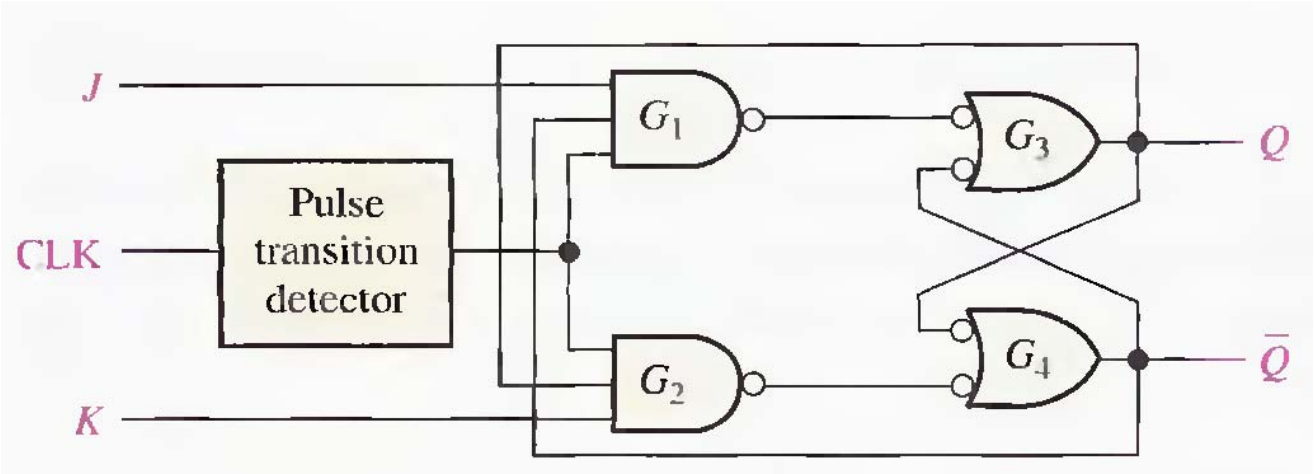
EXAMPLE 7–5

Given the waveforms in Figure 7–21(a) for the D input and the clock, determine the Q output waveform if the flip-flop starts out RESET.



The Edge-Triggered J-K Flip-Flop

The **J-K flip-flop** is versatile and is a widely used type of flip-flop. The functioning of the J-K flip-flop is identical to that of the S-R flip-flop in the SET, RESET, and no-change conditions of operation. The difference is that the J-K flip-flop has no invalid state as does the S-R flip-flop.

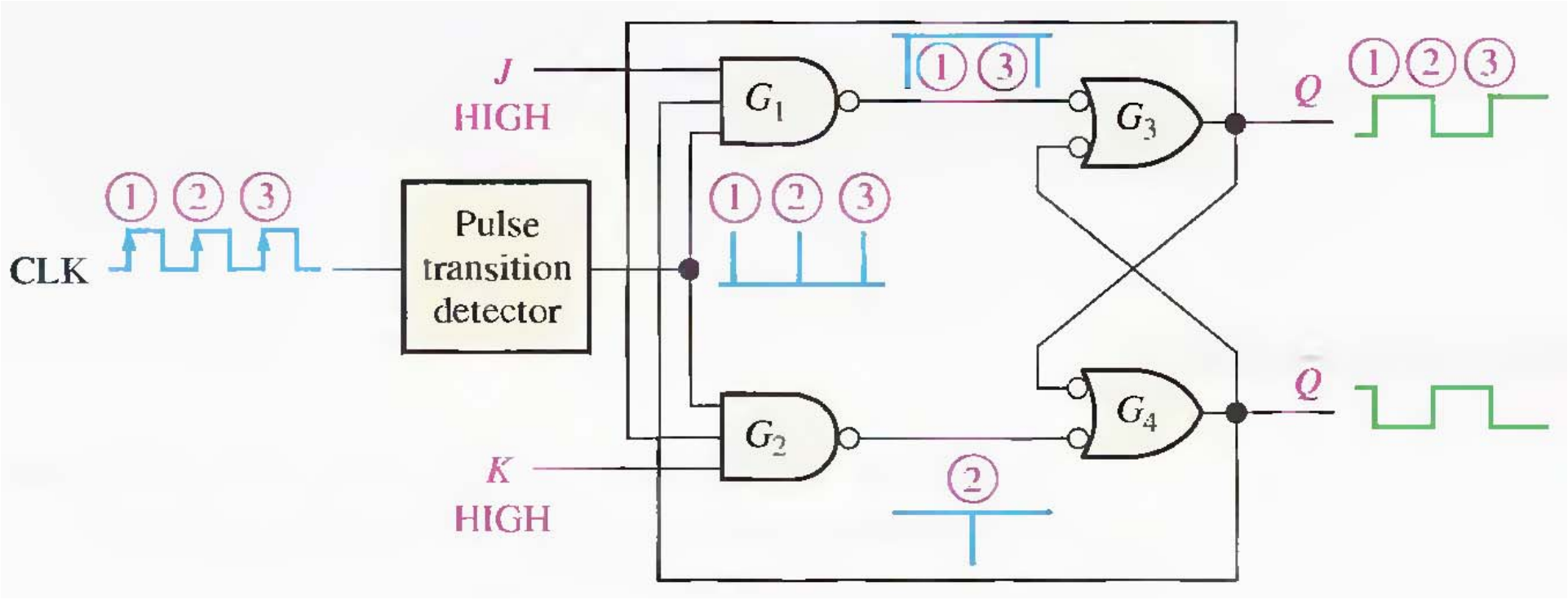


INPUTS			OUTPUTS		
J	K	CLK	Q	$\overline{Q}$	COMMENTS
0	0	↑	Q_0	$\overline{Q}_0$	No change
0	1	↑	0	1	RESET
1	0	↑	1	0	SET
1	1	↑	$\overline{Q}_0$	Q_0	Toggle

↑ = clock transition LOW to HIGH

Q_0 = output level prior to clock transition

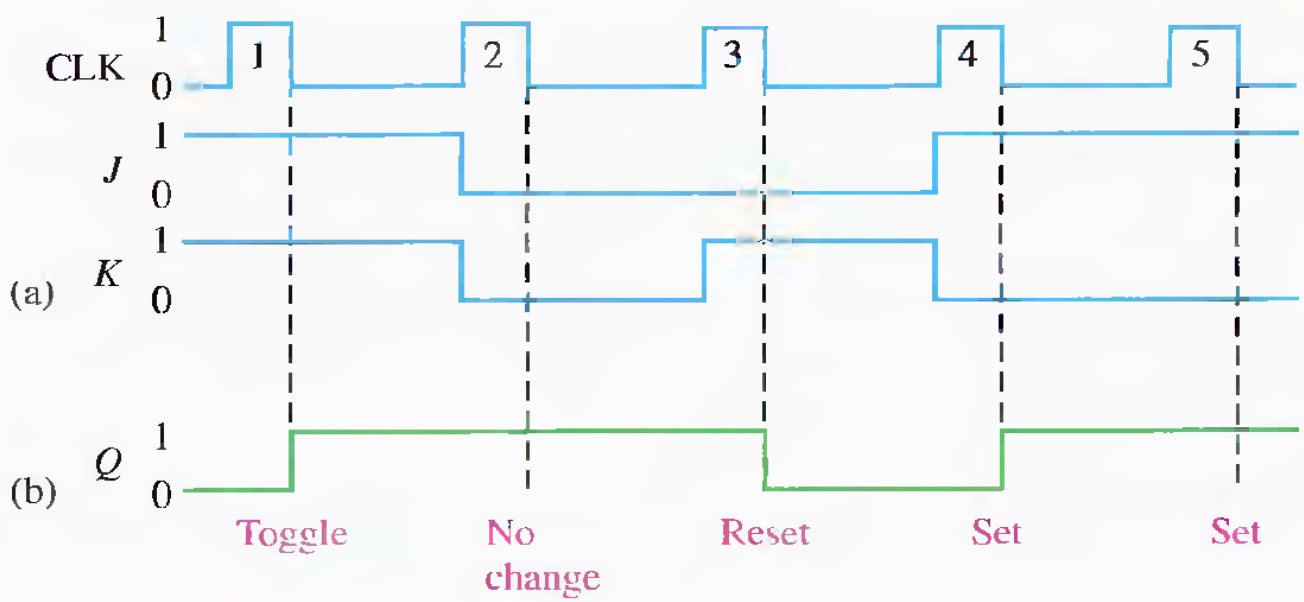
toggle operation.

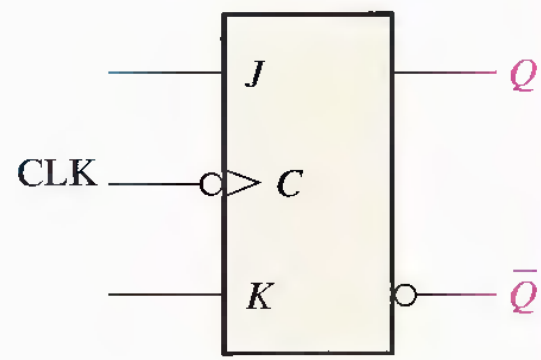


A J-K flip-flop connected for toggle operation is sometimes called a *T flip-flop*.

EXAMPLE 7–6

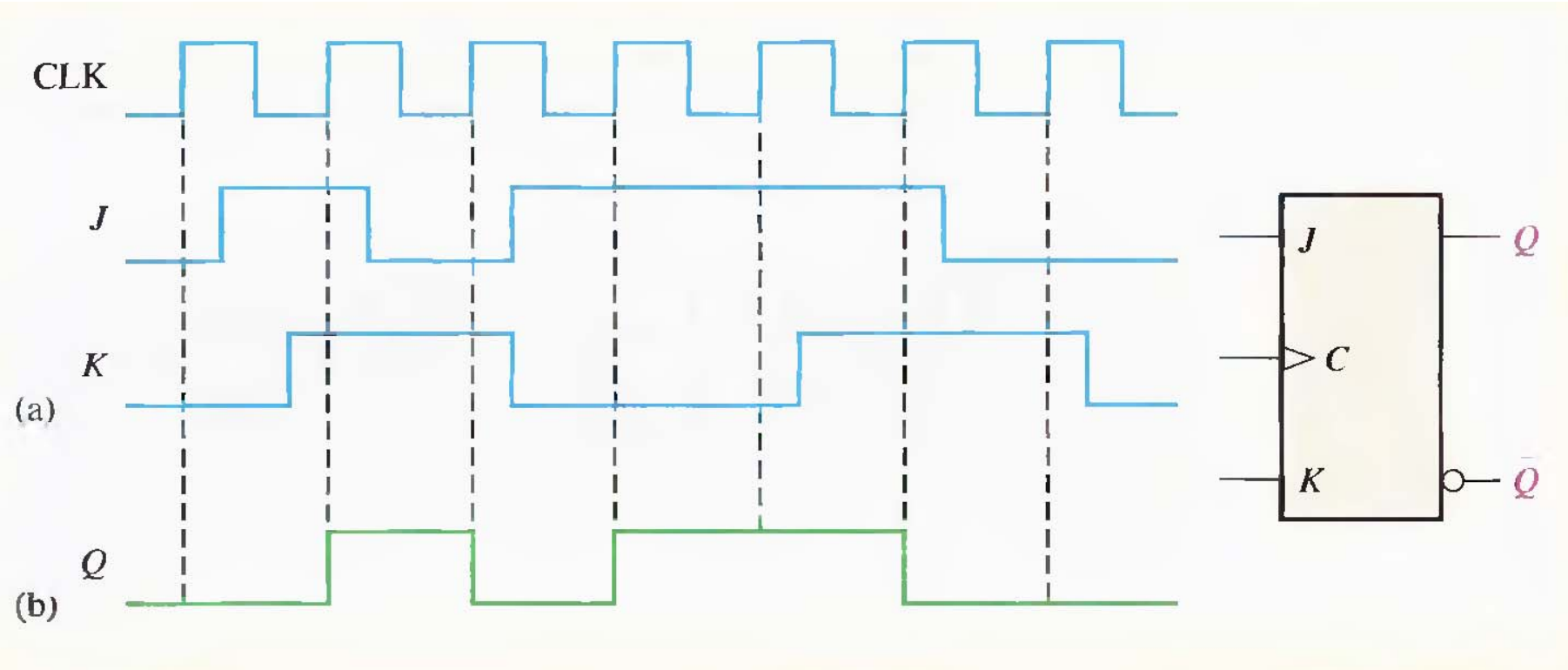
The waveforms in Figure 7–24(a) are applied to the *J*, *K*, and clock inputs as indicated. Determine the *Q* output, assuming that the flip-flop is initially RESET.



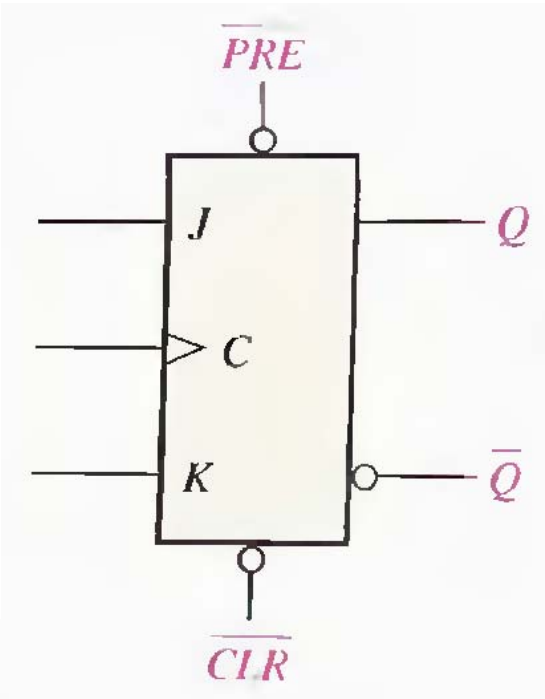


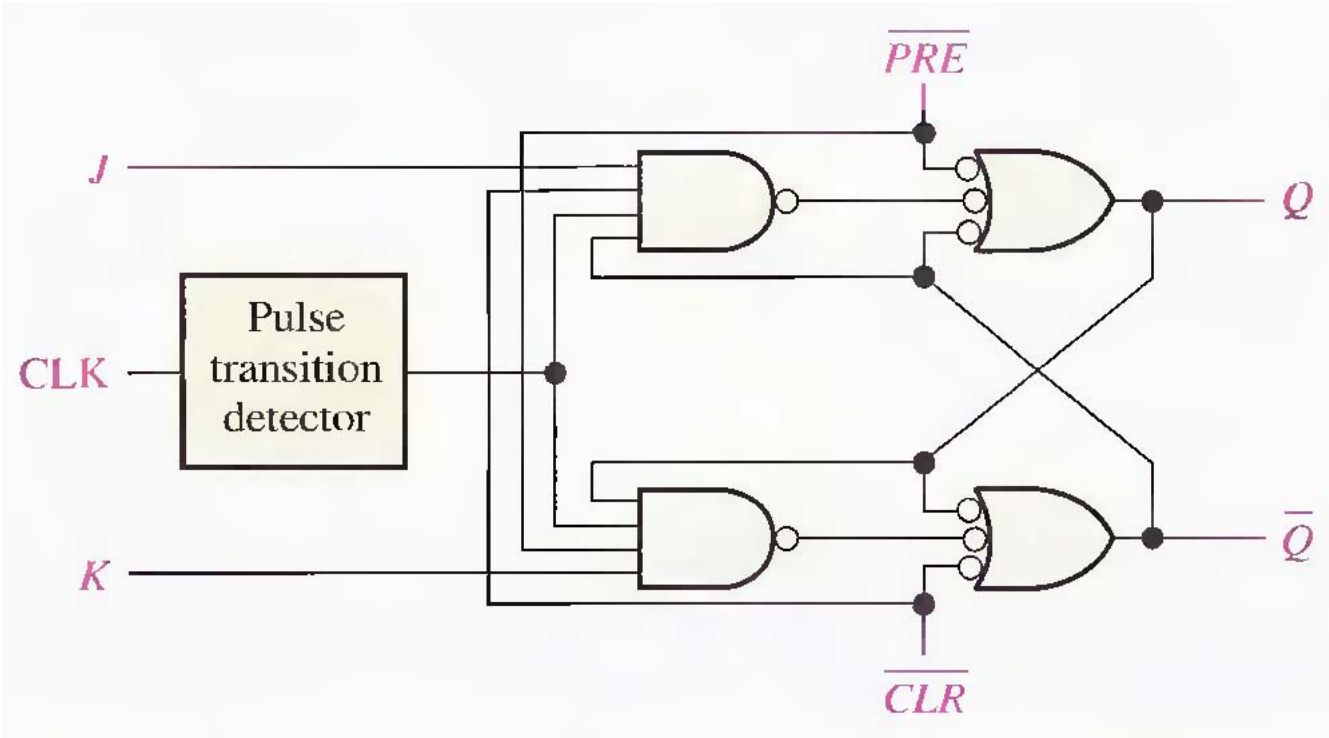
EXAMPLE 7–7

The waveforms in Figure 7–25(a) are applied to the flip-flop as shown. Determine the Q output, starting in the RESET state.



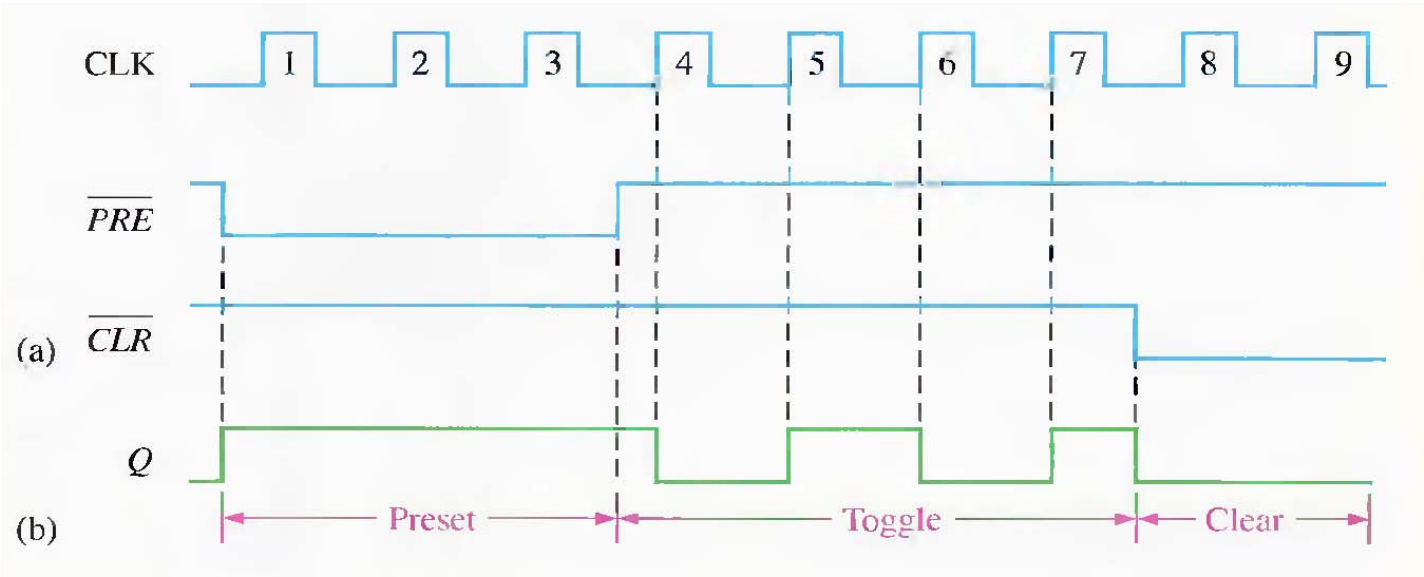
Asynchronous Preset and Clear Inputs

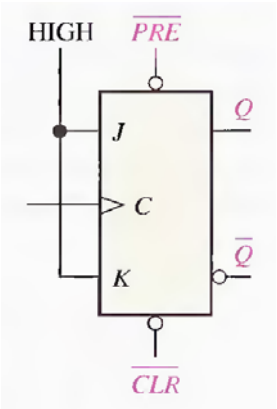




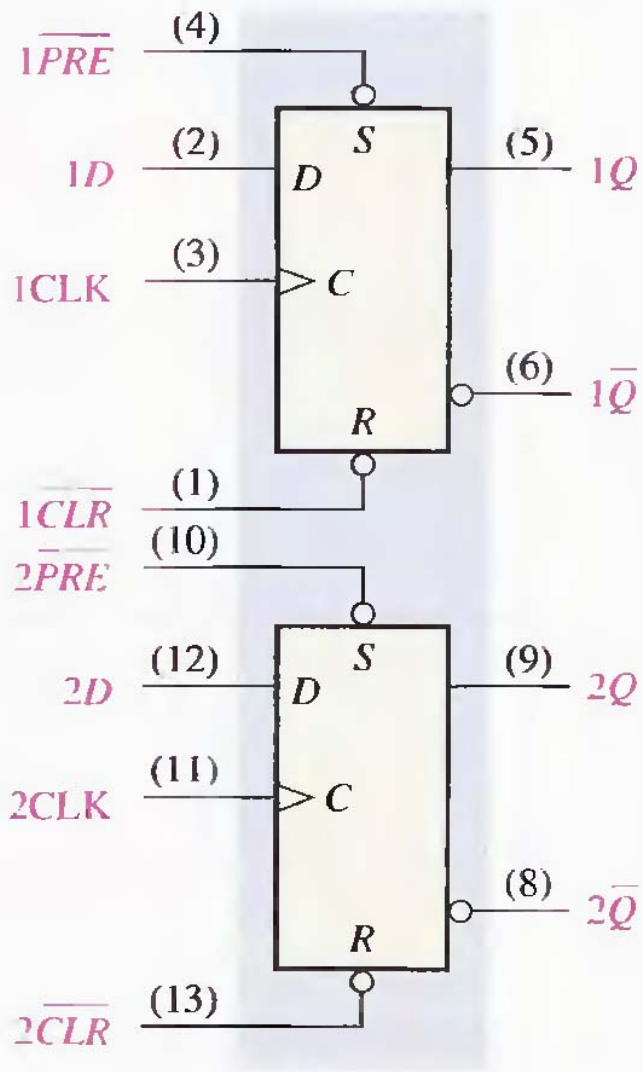
EXAMPLE 7–8

For the positive edge-triggered J-K flip-flop with preset and clear inputs in Figure 7–28, determine the Q output for the inputs shown in the timing diagram in part (a) if Q is initially LOW.

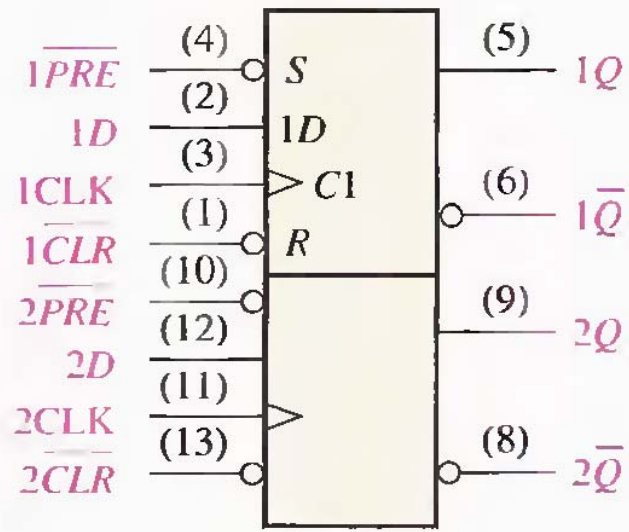




THE 74AHC74 DUAL D FLIP-FLOP

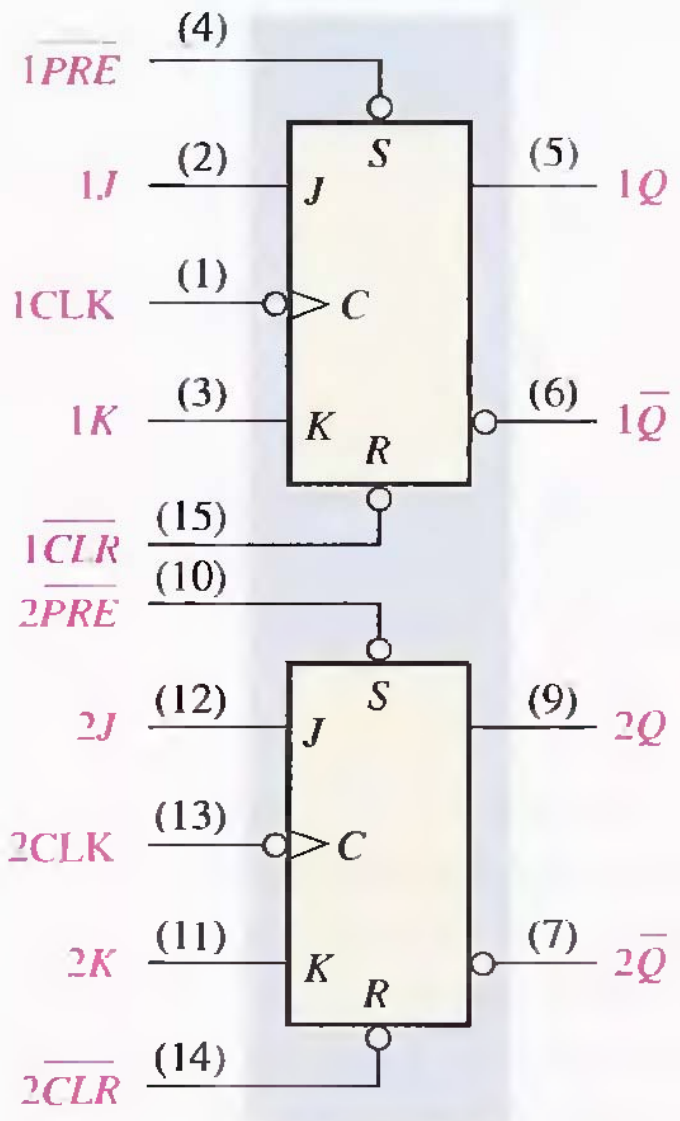


(a) Individual logic symbols

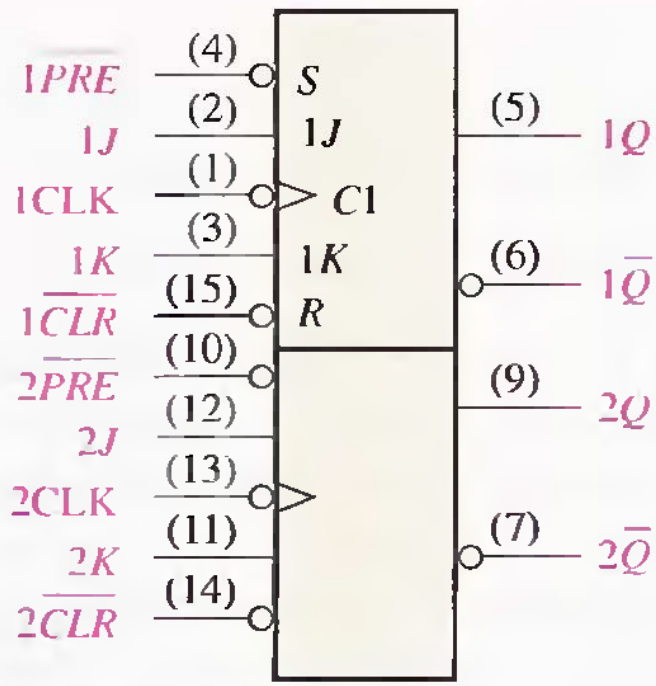


(b) Single block logic symbol
Note: The S and R inside the block indicate that $\overline{PRE}$ SETS and $\overline{CLR}$ RESETS.

THE 74HC112 DUAL J-K FLIP-FLOP



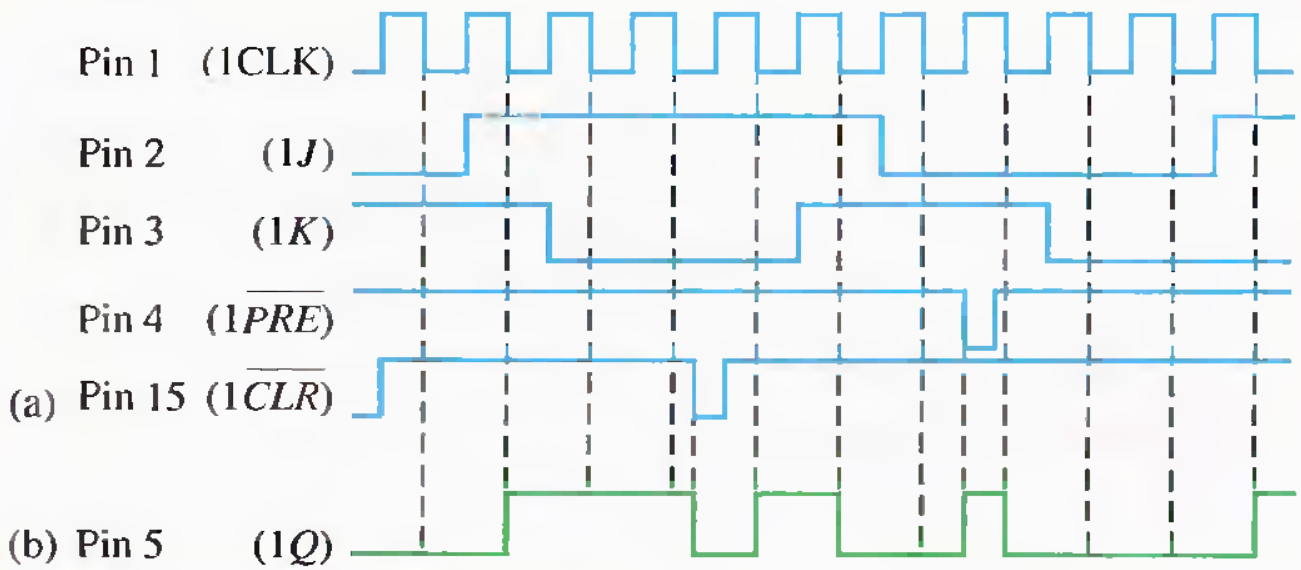
(a) Individual logic symbols



(b) Single block logic symbol

EXAMPLE 7–9

The 1*J*, 1*K*, 1CLK, 1$\overline{PRE}$, and 1$\overline{CLR}$ waveforms in Figure 7–31(a) are applied to one of the negative edge-triggered flip-flops in a 74HC112 package. Determine the 1*Q* output waveform.



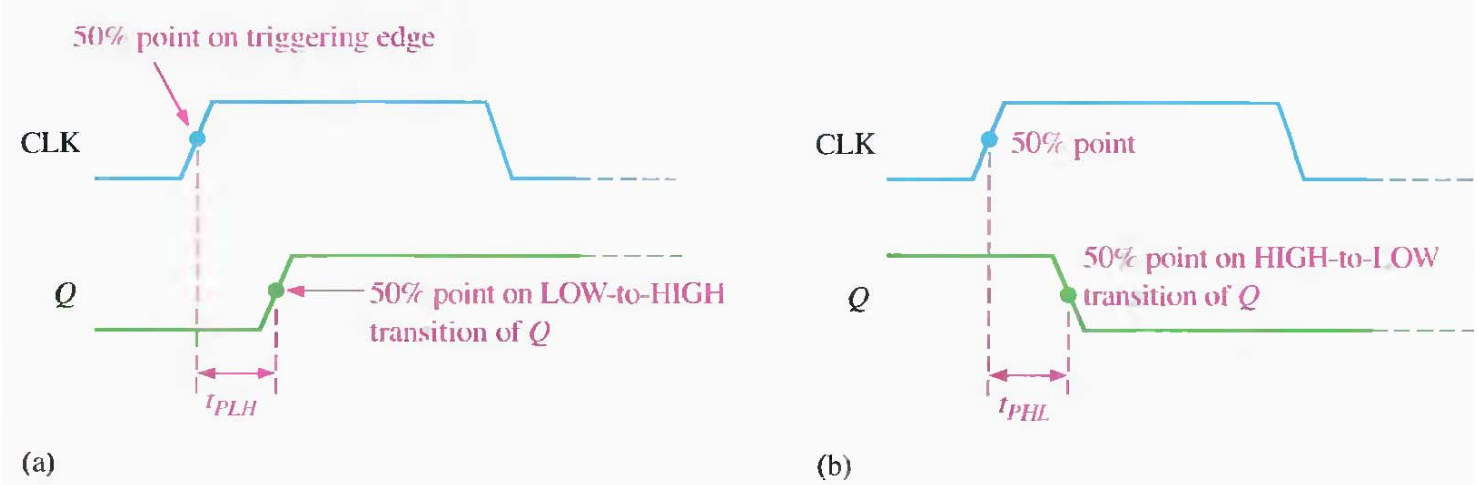
SECTION 7-2 REVIEW

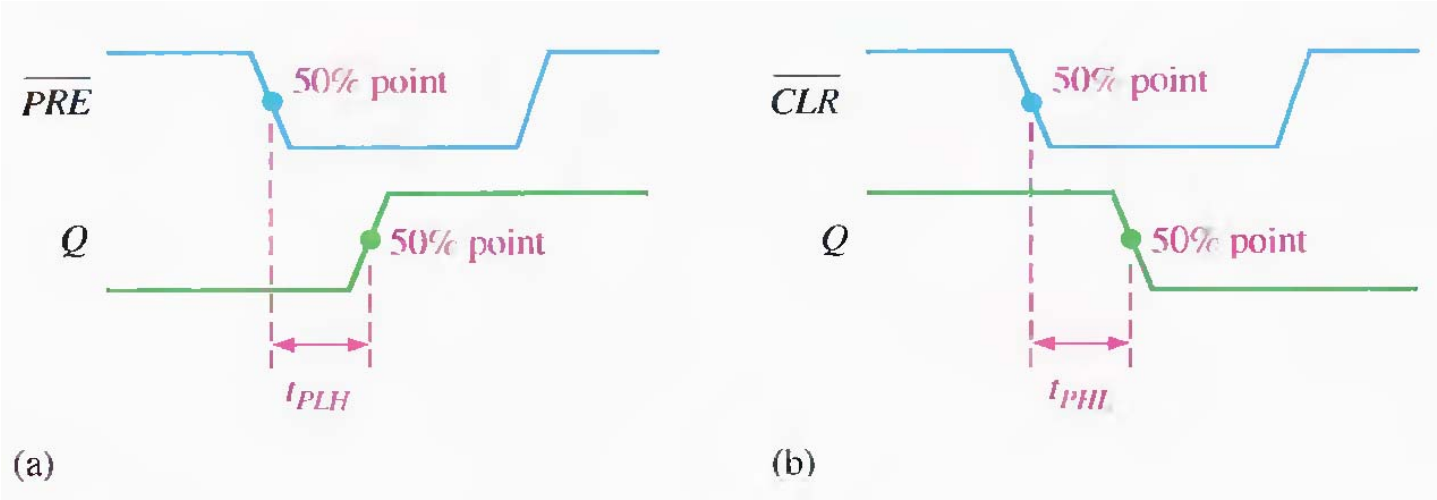
1. Describe the main difference between a gated S-R latch and an edge-triggered S-R flip-flop.
2. How does a J-K flip-flop differ from an S-R flip-flop in its basic operation?
3. Assume that the flip-flop in Figure 7-21 is negative edge-triggered. Describe the output waveform for the same CLK and *D* waveforms.

7–3 FLIP-FLOP OPERATING CHARACTERISTICS

Propagation Delay Times

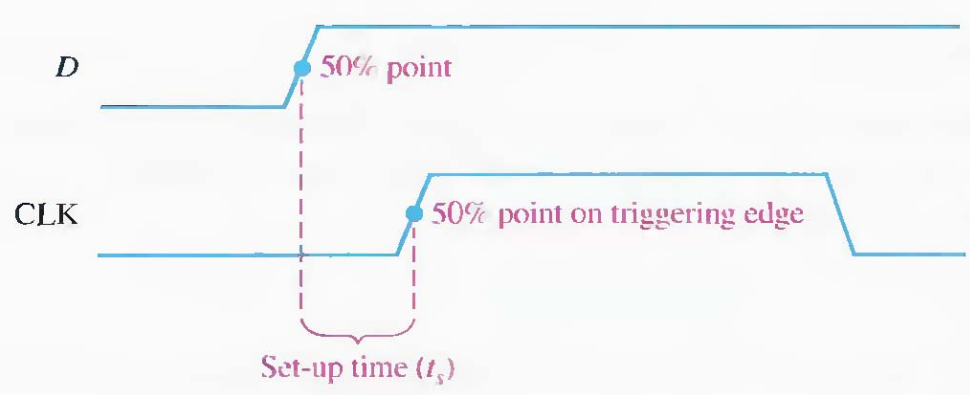
A **propagation delay time** is the interval of time required after an input signal has been applied for the resulting output change to occur. Four categories of propagation delay times are important in the operation of a flip-flop:

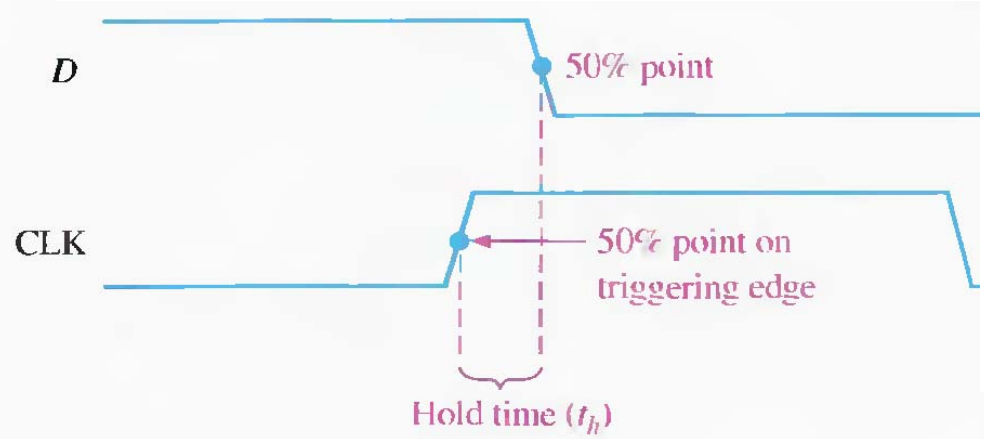




Set-up Time

The **set-up time** (t_s) is the minimum interval required for the logic levels to be maintained constantly on the inputs (J and K, or S and R, or D) prior to the triggering edge of the clock pulse in order for the levels to be reliably clocked into the flip-flop. This interval is illustrated in Figure 7–34 for a D flip-flop.





Hold Time

The **hold time** (t_h) is the minimum interval required for the logic levels to remain on the inputs after the triggering edge of the clock pulse in order for the levels to be reliably clocked into the flip-flop. This is illustrated in Figure 7–35 for a D flip-flop.

HANDS ON TIP

An advantage of CMOS is that it can operate over a wider range of dc supply voltages (typically 2 V to 6 V) than TTL and, therefore, less expensive power supplies that do not have precise regulation can be used. Also, batteries can be used as secondary or primary sources for CMOS circuits. In addition, lower voltages mean that the IC dissipates less power. The drawback is that the performance of CMOS is degraded with lower supply voltages. For example, the guaranteed maximum clock frequency of a CMOS flip-flop is much less at $V_{CC} = 2$ V than at $V_{CC} = 6$ V.

Maximum Clock Frequency

The maximum clock frequency (f_{max}) is the highest rate at which a flip-flop can be reliably triggered. At clock frequencies above the maximum, the flip-flop would be unable to respond quickly enough, and its operation would be impaired.

Pulse Widths

Minimum pulse widths (t_W) for reliable operation are usually specified by the manufacturer for the clock, preset, and clear inputs. Typically, the clock is specified by its minimum HIGH time and its minimum LOW time.

Power Dissipation

The **power dissipation** of any digital circuit is the total power consumption of the device. For example, if the flip-flop operates on a +5 V dc source and draws 5 mA of current, the power dissipation is

$$P = V_{CC} \times I_{CC} = 5\text{ V} \times 5\text{ mA} = 25\text{ mW}$$

The power dissipation is very important in most applications in which the capacity of the dc supply is a concern. As an example, let's assume that you have a digital system that requires a total of ten flip-flops, and each flip-flop dissipates 25 mW of power. The total power requirement is

$$P_T = 10 \times 25\text{ mW} = 250\text{ mW} = 0.25\text{ W}$$

This tells you the output capacity required of the dc supply. If the flip-flops operate on +5 V dc, then the amount of current that the supply must provide is

$$I = \frac{250\text{ mW}}{5\text{ V}} = 50\text{ mA}$$

You must use a +5 V dc supply that is capable of providing at least 50 mA of current.

Comparison of Specific Flip-Flops

PARAMETER	CMOS 74HC74A	CMOS 74AHC74	TTL 74LS74A	TTL 74F74
t_{PHL} (CLK to Q)	17 ns	4.6 ns	40 ns	6.8 ns
t_{PLH} (CLK to Q)	17 ns	4.6 ns	25 ns	8.0 ns
t_{PHL} ($\overline{CLR}$ to Q)	18 ns	4.8 ns	40 ns	9.0 ns
t_{PLH} ($\overline{PRE}$ to Q)	18 ns	4.8 ns	25 ns	6.1 ns
t_s (set-up time)	14 ns	5.0 ns	20 ns	2.0 ns
t_h (hold time)	3.0 ns	0.5 ns	5 ns	1.0 ns
t_W (CLK HIGH)	10 ns	5.0 ns	25 ns	4.0 ns
t_W (CLK LOW)	10 ns	5.0 ns	25 ns	5.0 ns
$t_W(\overline{CLR}/\overline{PRE})$	10 ns	5.0 ns	25 ns	4.0 ns
f_{max}	35 MHz	170 MHz	25 MHz	100 MHz
Power, quiescent	0.012 mW	1.1 mW		
Power, 50% duty cycle			44 mW	88 mW

SECTION 7-3 REVIEW

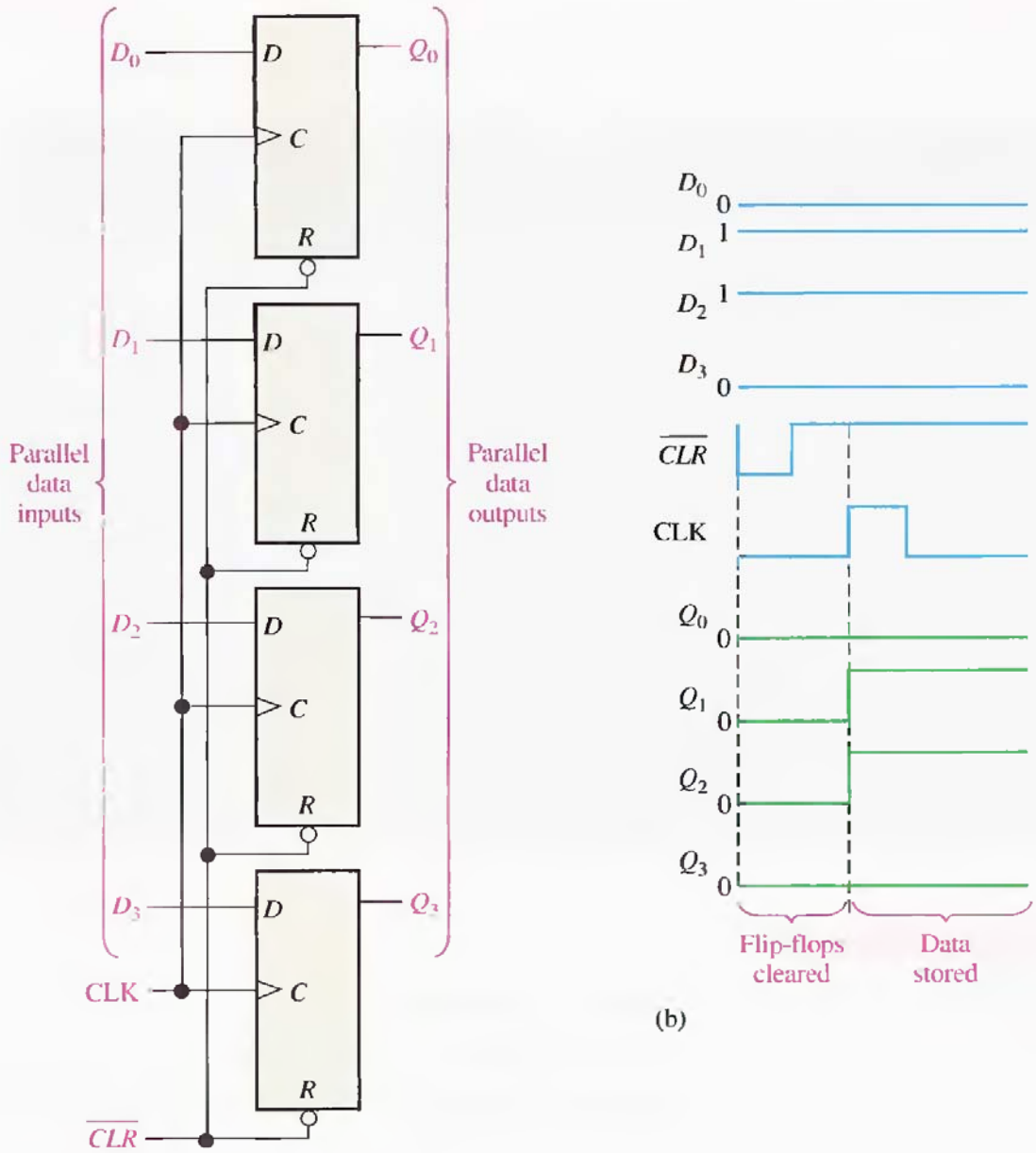
1. Define the following:

   (a) set-up time (b) hold time

2. Which specific flip-flop in Table 7–5 can be operated at the highest frequency?

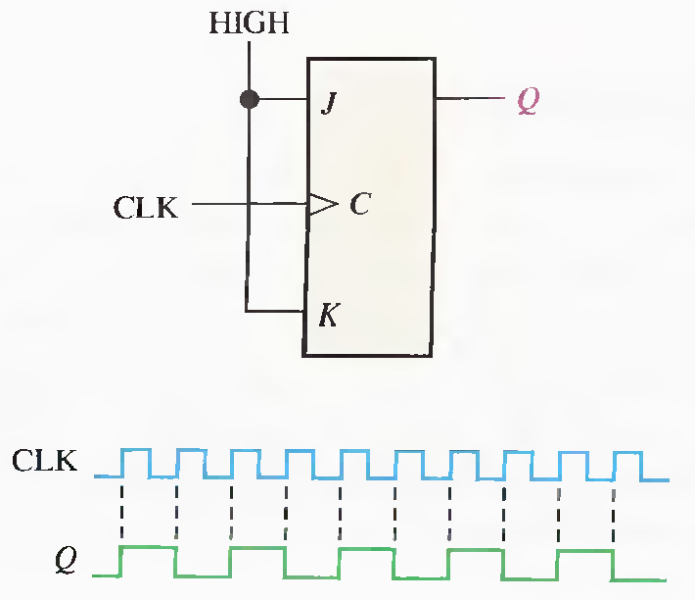
7–4 FLIP-FLOP APPLICATIONS

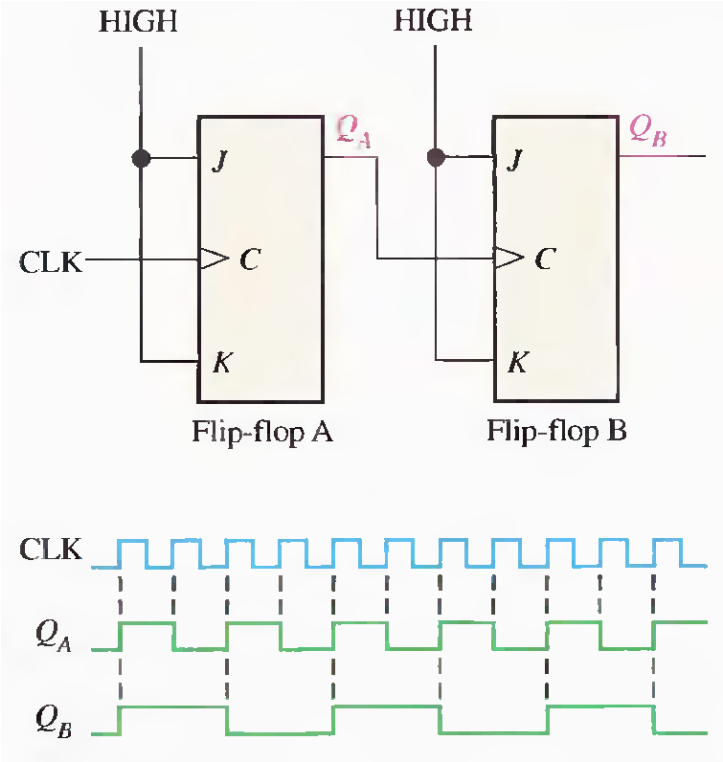
Parallel Data Storage



7–4 FLIP-FLOP APPLICATIONS

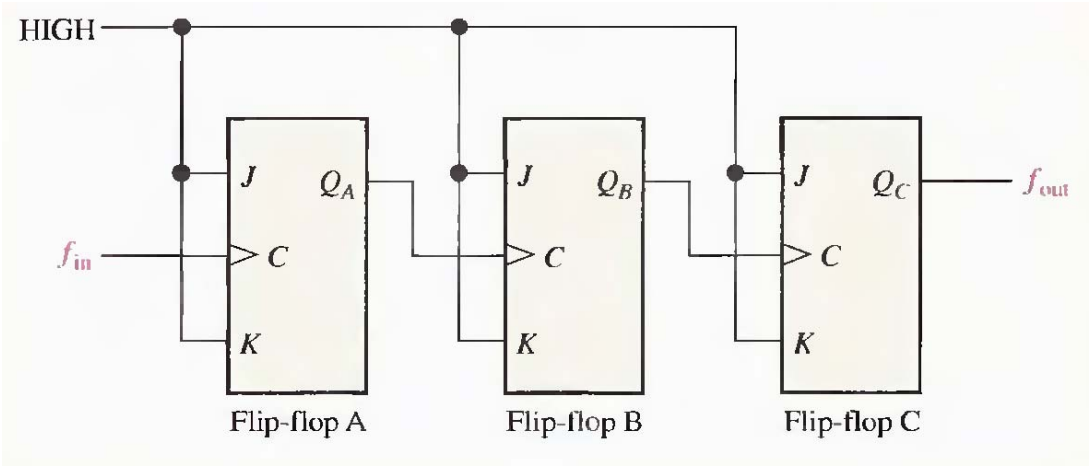
Frequency Division

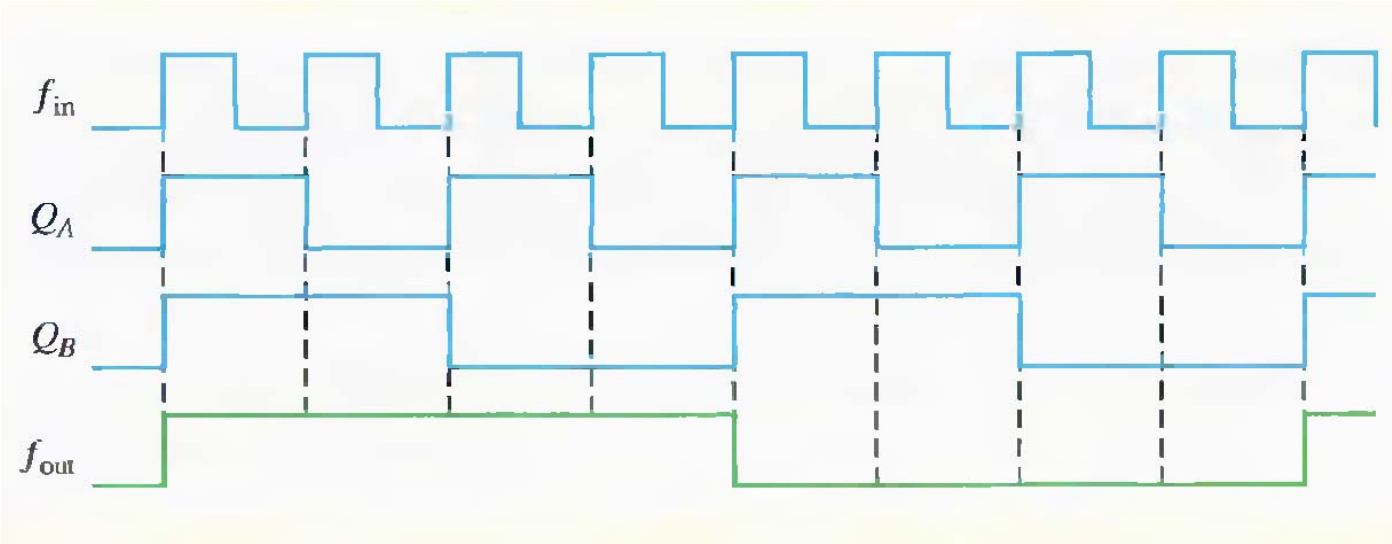




EXAMPLE 7–10

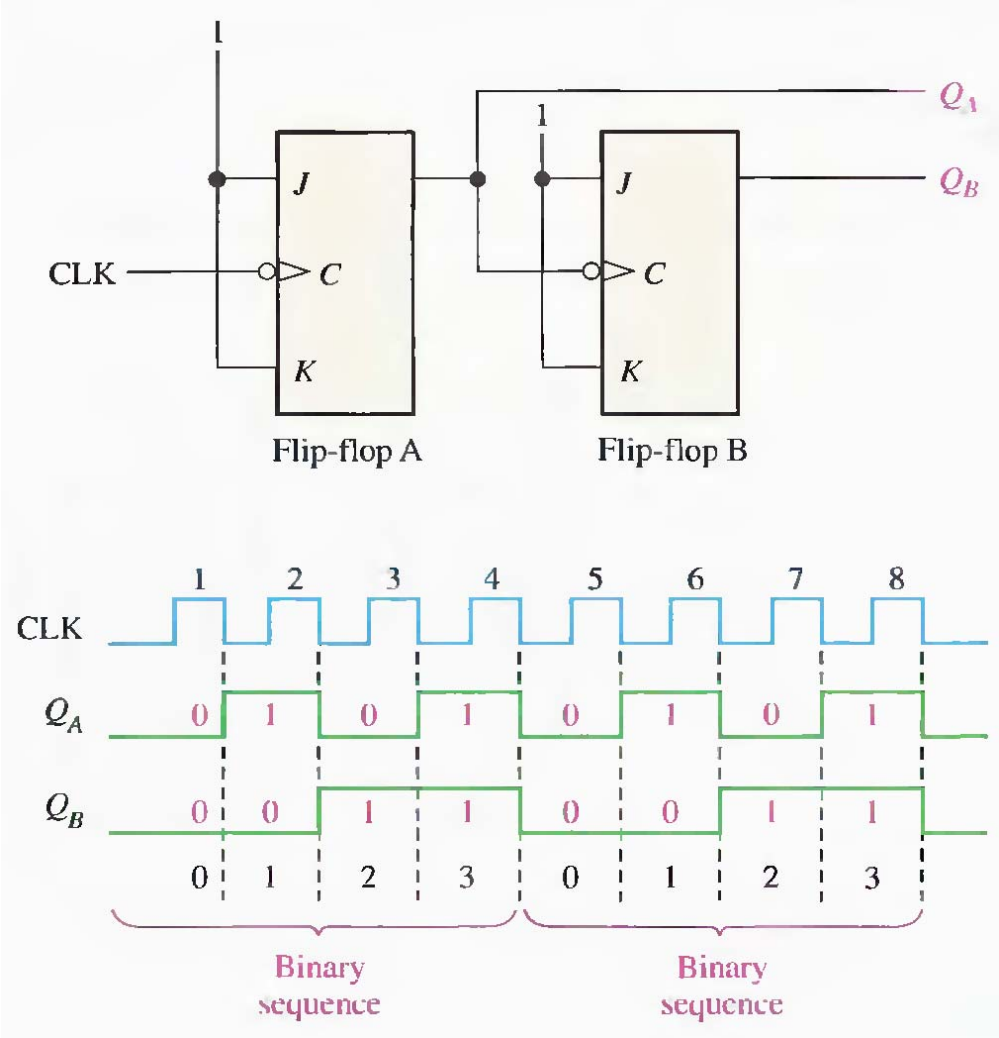
Develop the f_{out} waveform for the circuit in Figure 7–39 when an 8 kHz square wave input is applied to the clock input of flip-flop A.





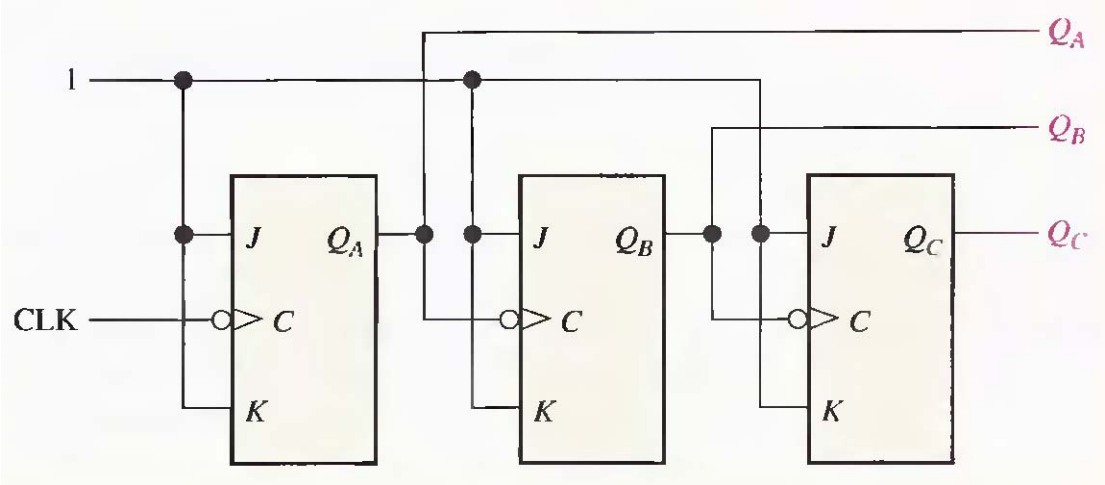
7–4 FLIP-FLOP APPLICATIONS

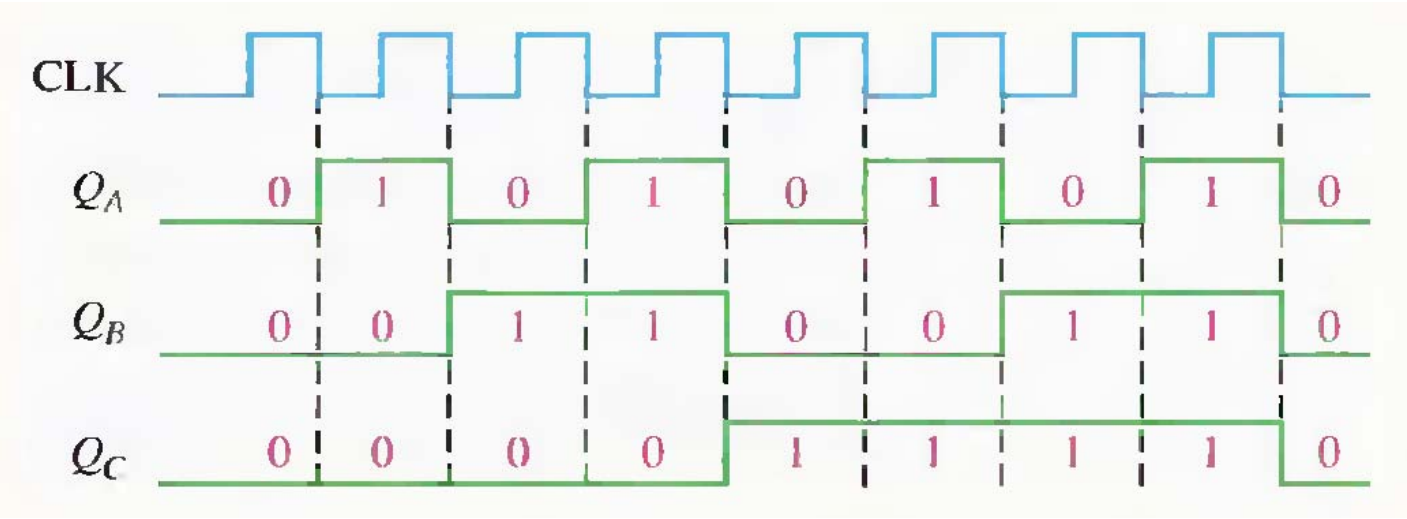
Counting



EXAMPLE 7–11

Determine the output waveforms in relation to the clock for Q_A, Q_B, and Q_C in the circuit of Figure 7–42 and show the binary sequence represented by these waveforms.





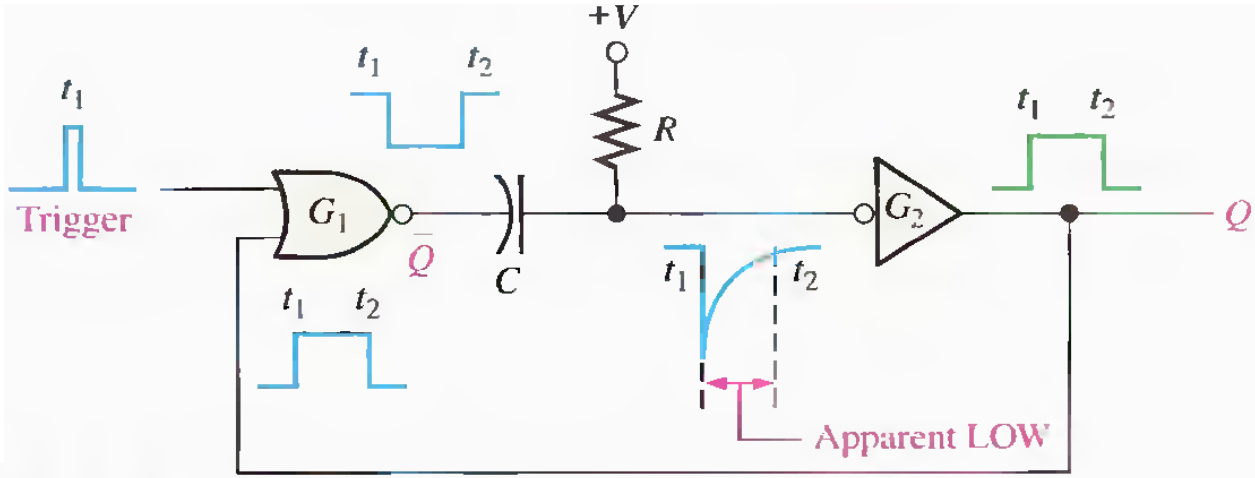
SECTION 7-4 REVIEW

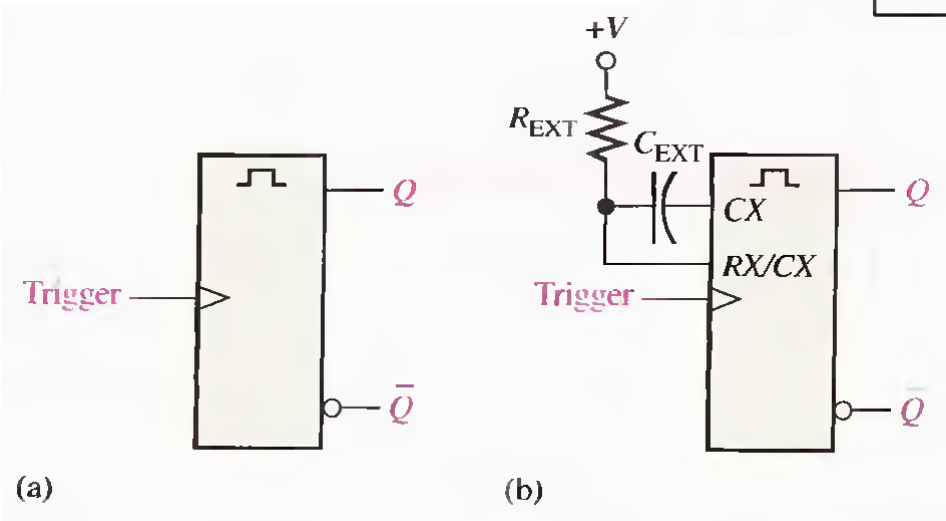
1. What is a group of flip-flops used for data storage called?
2. How must a J-K flip-flop be connected to function as a divide-by-2 device?
3. How many flip-flops are required to produce a divide-by-64 device?

7–5 ONE-SHOTS

The **one-shot** is a **monostable** multivibrator, a device with only one stable state. A one-shot is normally in its stable state and will change to its unstable state only when triggered. Once it is triggered, the one-shot remains in its unstable state for a predetermined length of time and then automatically returns to its stable state. The time that the device stays in its unstable state determines the pulse width of its output.

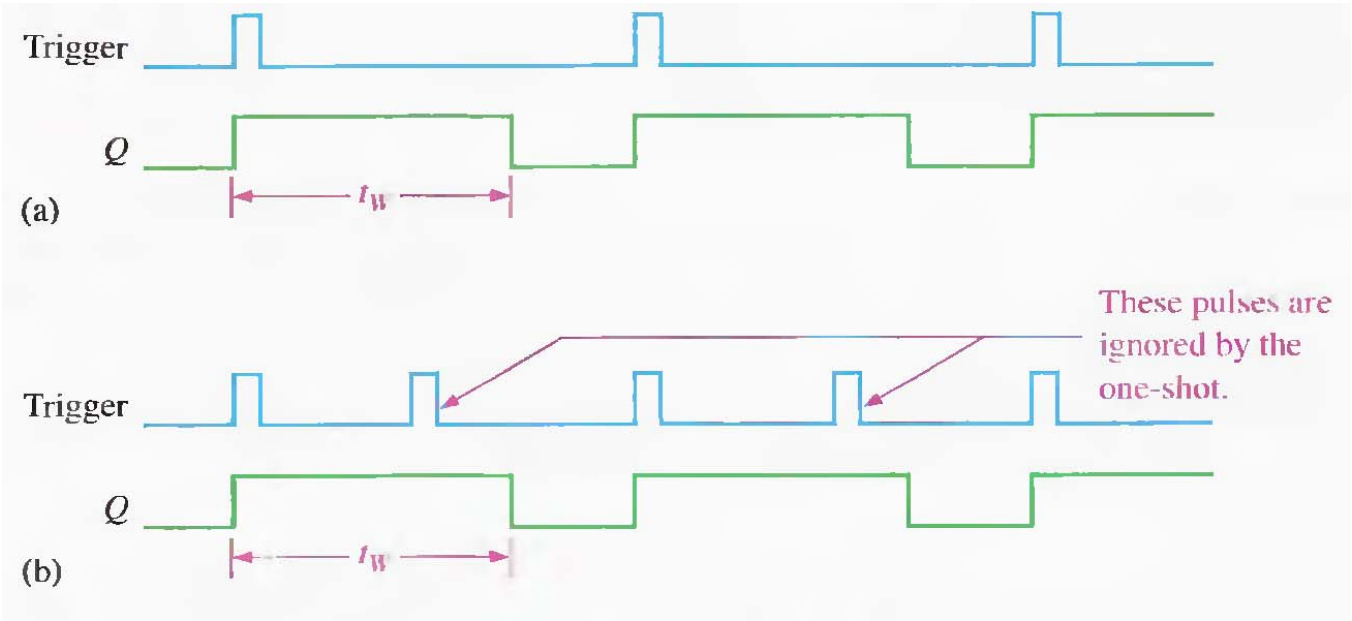
A simple one-shot circuit.



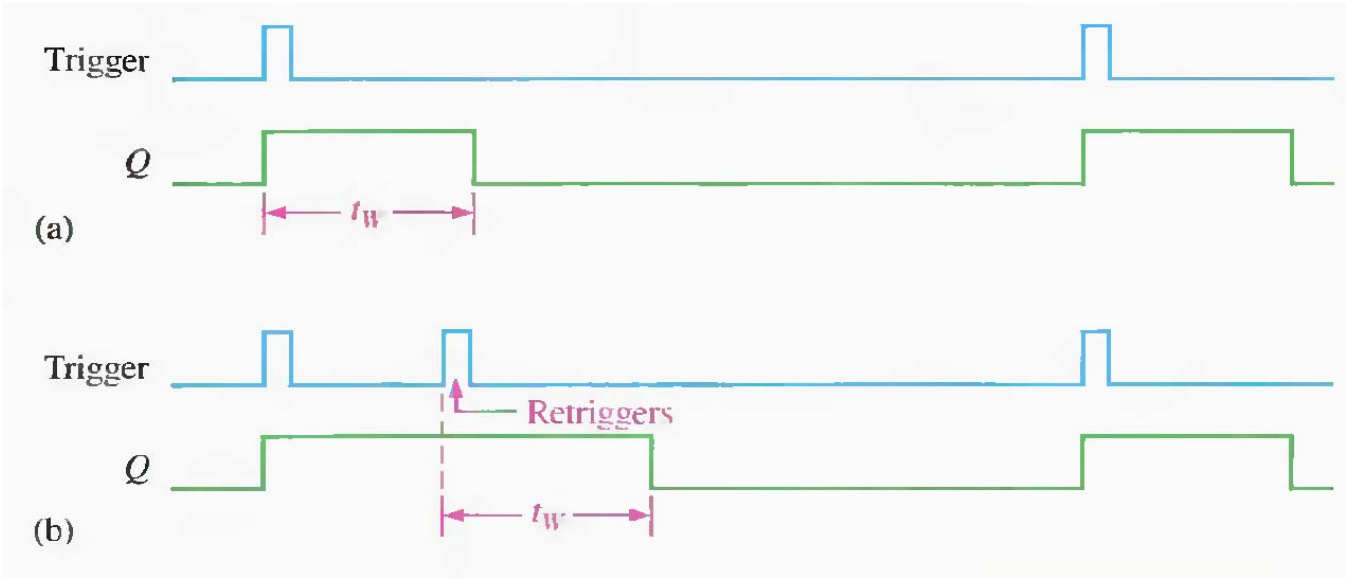


Basic one-shot logic symbols. *CX* and *RX* stand for external components.

Nonretriggerable one-shot action.



Retriggerable one-shot action.



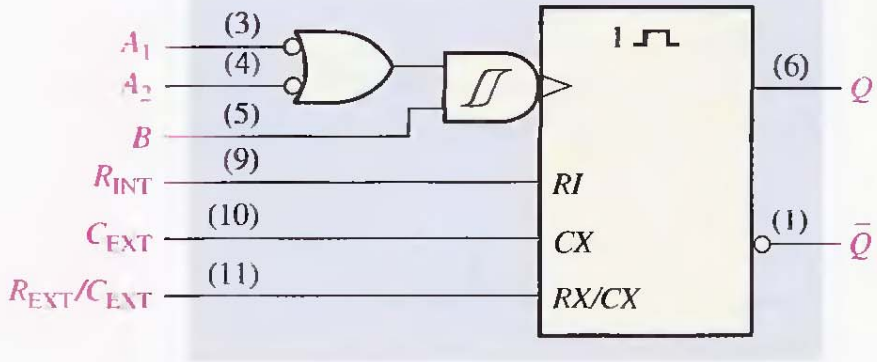
THE 74121 NONRETRIGGERABLE ONE-SHOT

Setting the Pulse Width A typical pulse width of about 30 ns is produced when no external timing components are used and the internal timing resistor (R_{INT}) is connected to V_{CC}, as shown in Figure 7–49(a). The pulse width can be set anywhere between about 30 ns and 28 s by the use of external components. Figure 7–49(b) shows the configuration using the internal resistor (2 kΩ) and an external capacitor. Part (c) shows the configuration using an external resistor and an external capacitor. The output pulse width is set by the values of the resistor (R_{INT} = 2 kΩ, and R_{EXT} is selected) and the capacitor according to the following formula:

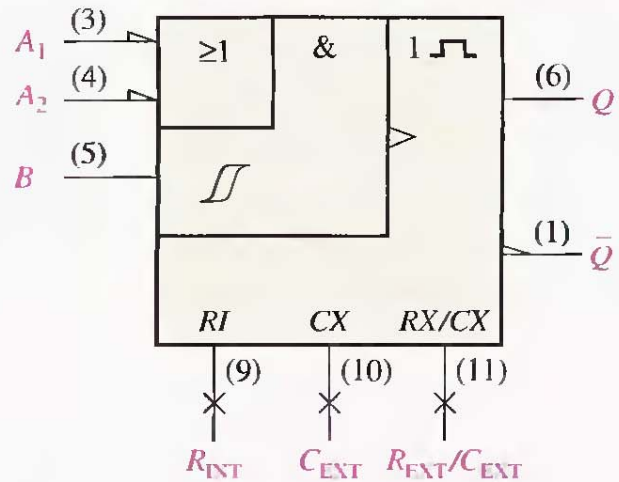
$$t_W = 0.7RC_{EXT}$$

where R is either R_{INT} or R_{EXT}. When R is in kilohms (kΩ) and C_{EXT} is in picofarads (pF), the output pulse width t_W is in nanoseconds (ns).

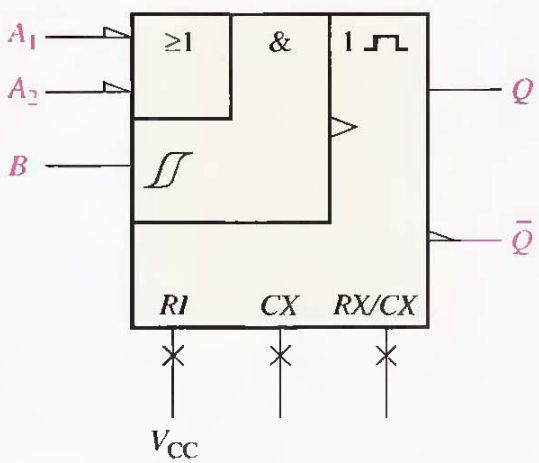
The Schmitt-Trigger Symbol The symbol indicates a Schmitt-trigger input. This type of input uses a special threshold circuit that produces **hysteresis,** a characteristic that prevents erratic switching between states when a slow-changing trigger voltage hovers around the critical input level. This allows reliable triggering to occur even when the input is changing as slowly as 1 volt/second.



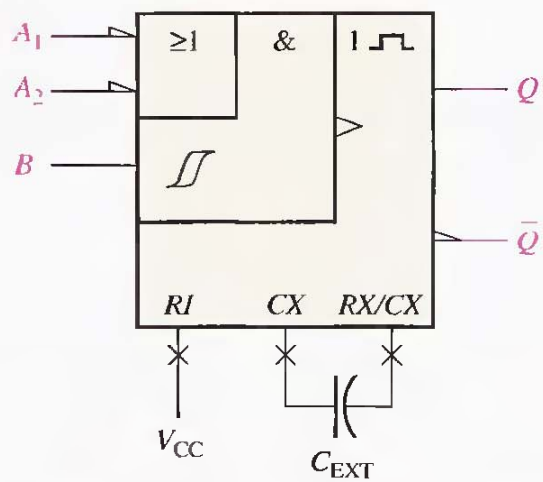
(a) Traditional logic symbol



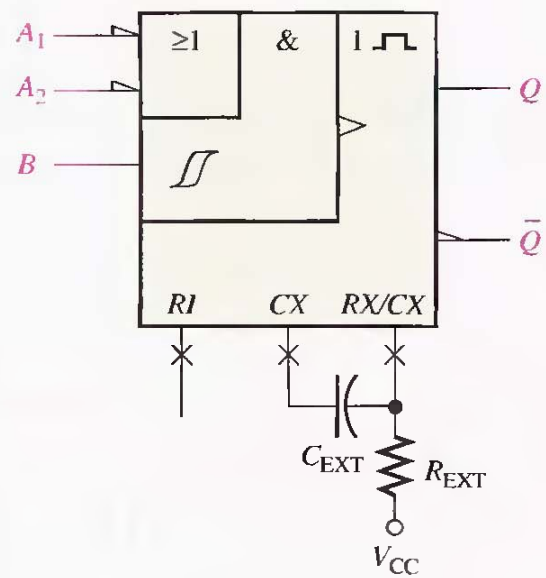
(b) ANSI/IEEE std. 91–1984 logic symbol (× = nonlogic connection). "1 ⎍" is the qualifying symbol for a nonretriggerable one-shot.



(a) No external components
R_{INT} to V_{CC}
$t_W \cong 30$ ns

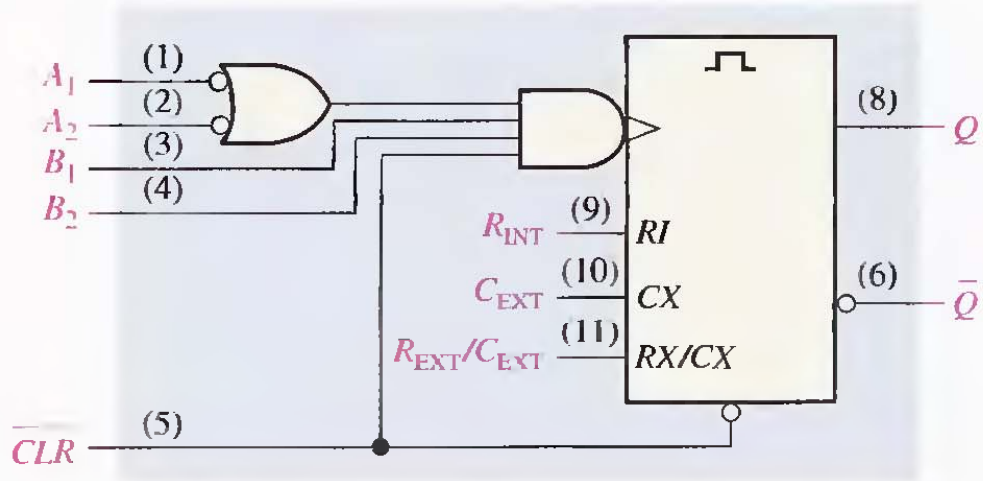


(b) R_{INT} and C_{EXT}
$t_W = 0.7(2\ k\Omega)C_{EXT}$

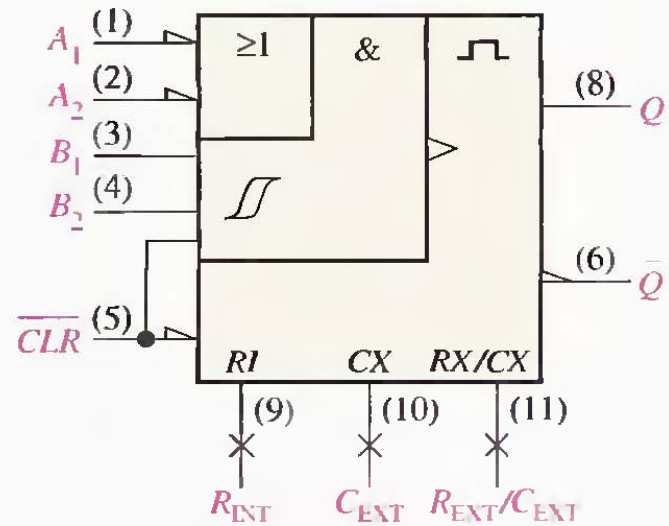


(c) R_{EXT} and C_{EXT}
$t_W = 0.7R_{EXT}\,C_{EXT}$

THE 74LS122 RETRIGGERABLE ONE-SHOT



(a) Traditional logic symbol



(b) ANSI/IEEE std. 91–1984 logic symbol (× = nonlogic connection). ⎍ is the qualifying symbol for a retriggerable one-shot.

A minimum pulse width of approximately 45 ns is obtained with no external components. Wider pulse widths are achieved by using external components. A general formula for calculating the values of these components for a specified pulse width (t_W) is

$$t_W = 0.32RC_{\text{EXT}}\left(1 + \frac{0.7}{R}\right)$$

where 0.32 is a constant determined by the particular type of one-shot, R is in kΩ and is either the internal or the external resistor, C_{EXT} is in pF, and t_W is in ns. The internal resistance is 10 kΩ and can be used instead of an external resistor. (Notice the difference between this formula and that for the 74121, shown in Equation 7–1.)

EXAMPLE 7–12

A certain application requires a one-shot with a pulse width of approximately 100 ms. Using a 74121, show the connections and the component values.

Arbitrarily select R_{EXT} = **39 kΩ** and calculate the necessary capacitance.

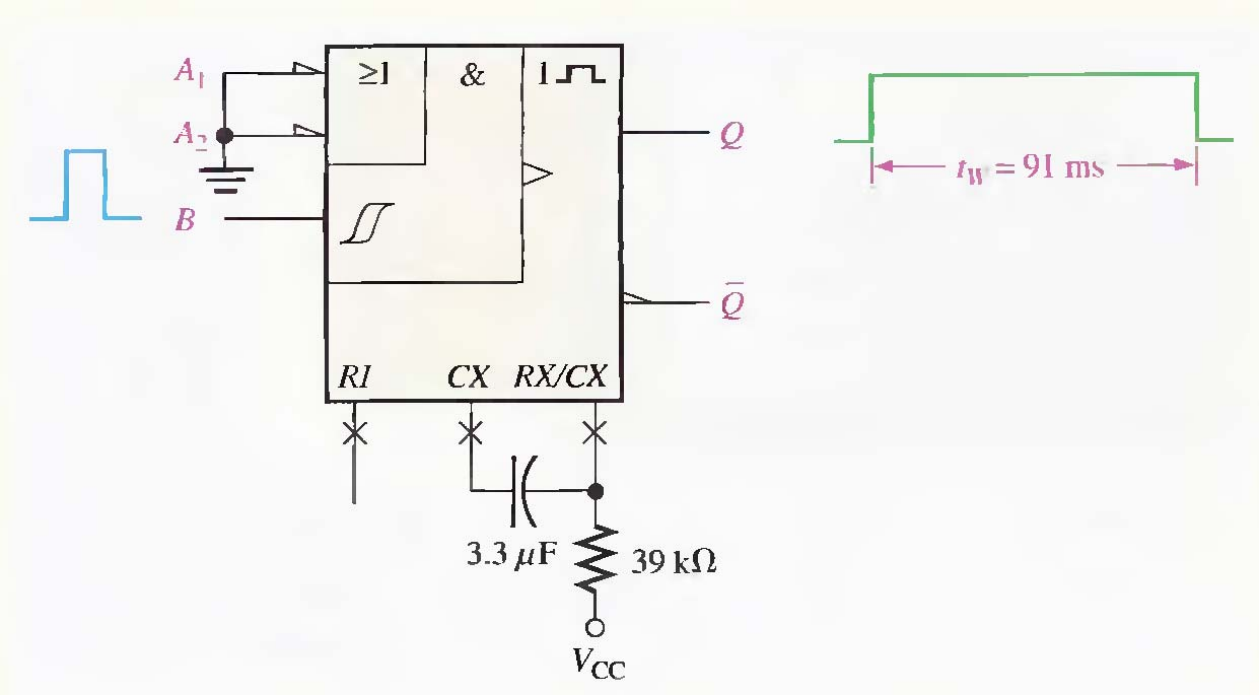
$$t_W = 0.7R_{EXT}C_{EXT}$$

$$C_{EXT} = \frac{t_W}{0.7R_{EXT}}$$

where C_{EXT} is in pF, R_{EXT} is in kΩ, and t_W is in ns. Since 100 ms = 1×10^8 ns,

$$C_{EXT} = \frac{1 \times 10^8 \text{ ns}}{0.7(39 \text{ k}\Omega)} = 3.66 \times 10^{-6} \text{ pF} = \mathbf{3.66\ \mu F}$$

A standard 3.3 μF capacitor will give an output pulse width of 91 ms. The proper connections are shown in Figure 7–51. To achieve a pulse width closer to 100 ms, other combinations of values for R_{EXT} and C_{EXT} can be tried. For example, R_{EXT} = 68 kΩ and C_{EXT} = 2.2 μF gives a pulse width of 105 ms.


A1
A2
B
≥1
&
1
Q
Q̄
RI
CX
RX/CX
3.3 μF
39 kΩ
VCC
tW = 91 ms

EXAMPLE 7–13

Determine the values of R_{EXT} and C_{EXT} that will produce a pulse width of 1 μs when connected to a 74LS122.

Assume a value of C_{EXT} = **560 pF** and then solve for R_{EXT}. The pulse width must be expressed in ns and C_{EXT} in pF. R_{EXT} will be in kΩ.

$$t_W = 0.32R_{EXT}C_{EXT}\left(1 + \frac{0.7}{R_{EXT}}\right) = 0.32R_{EXT}C_{EXT} + 0.7\left(\frac{0.32R_{EXT}C_{EXT}}{R_{EXT}}\right)$$

$$= 0.32R_{EXT}C_{EXT} + (0.7)(0.32)C_{EXT}$$

$$R_{EXT} = \frac{t_W - (0.7)(0.32)C_{EXT}}{0.32C_{EXT}} = \frac{t_W}{0.32C_{EXT}} - 0.7$$

$$= \frac{1000\text{ ns}}{(0.32)560\text{ pF}} - 0.7 = \mathbf{4.88\ k\Omega}$$

Use a standard value of **4.7 kΩ.**

An Application

One practical one-shot application is a sequential timer that can be used to illuminate a series of lights. This type of circuit can be used, for example, in a lane change directional indicator for highway construction projects or in sequential turn signals on automobiles.

Figure 7–52 shows three 74LS122 one-shots connected as a sequential timer. This particular circuit produces a sequence of three 1 s pulses. The first one-shot is triggered by a switch closure or a low-frequency pulse input, producing a 1 s output pulse. When the first one-shot (OS 1) times out and the 1 s pulse goes LOW, the second one-shot (OS 2) is triggered, also producing a 1 s output pulse. When this second pulse goes LOW, the third one-shot (OS 3) is triggered and the third 1 s pulse is produced. The output timing is illustrated in the figure. Variations of this basic arrangement can be used to produce a variety of timed outputs.

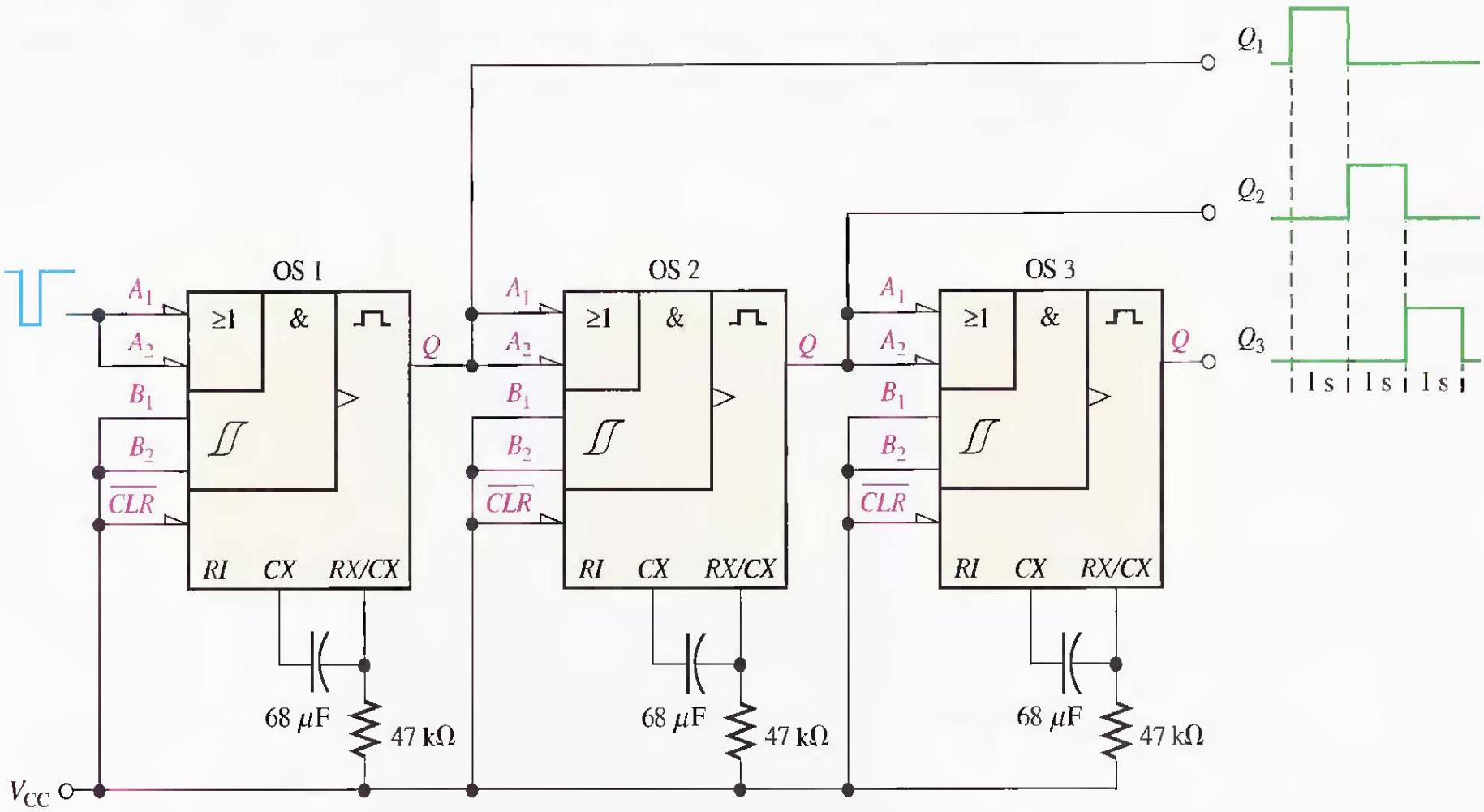


FIGURE 7–52

A sequential timing circuit using three 74LS122 one-shots.

SECTION 7-5 REVIEW

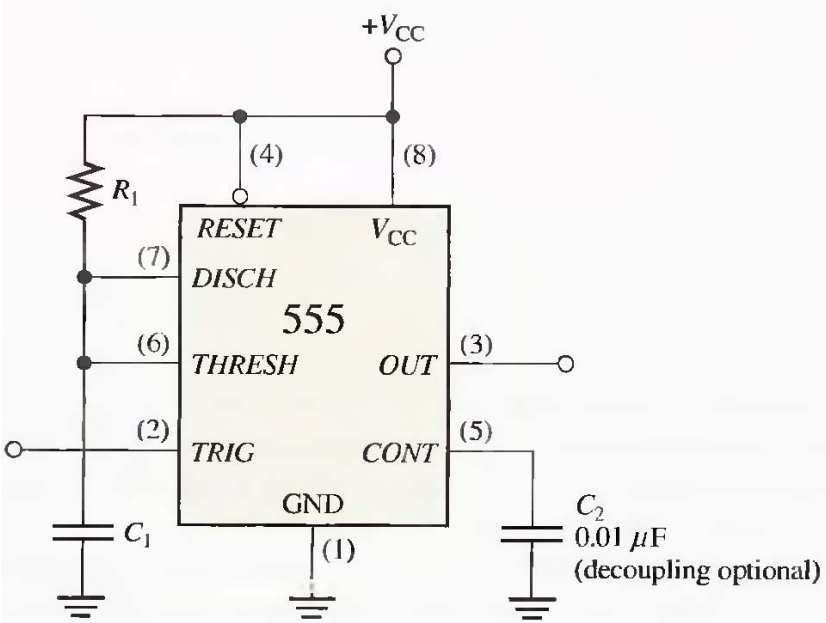
1. Describe the difference between a nonretriggerable and a retriggerable one-shot.
2. How is the output pulse width set in most IC one-shots?

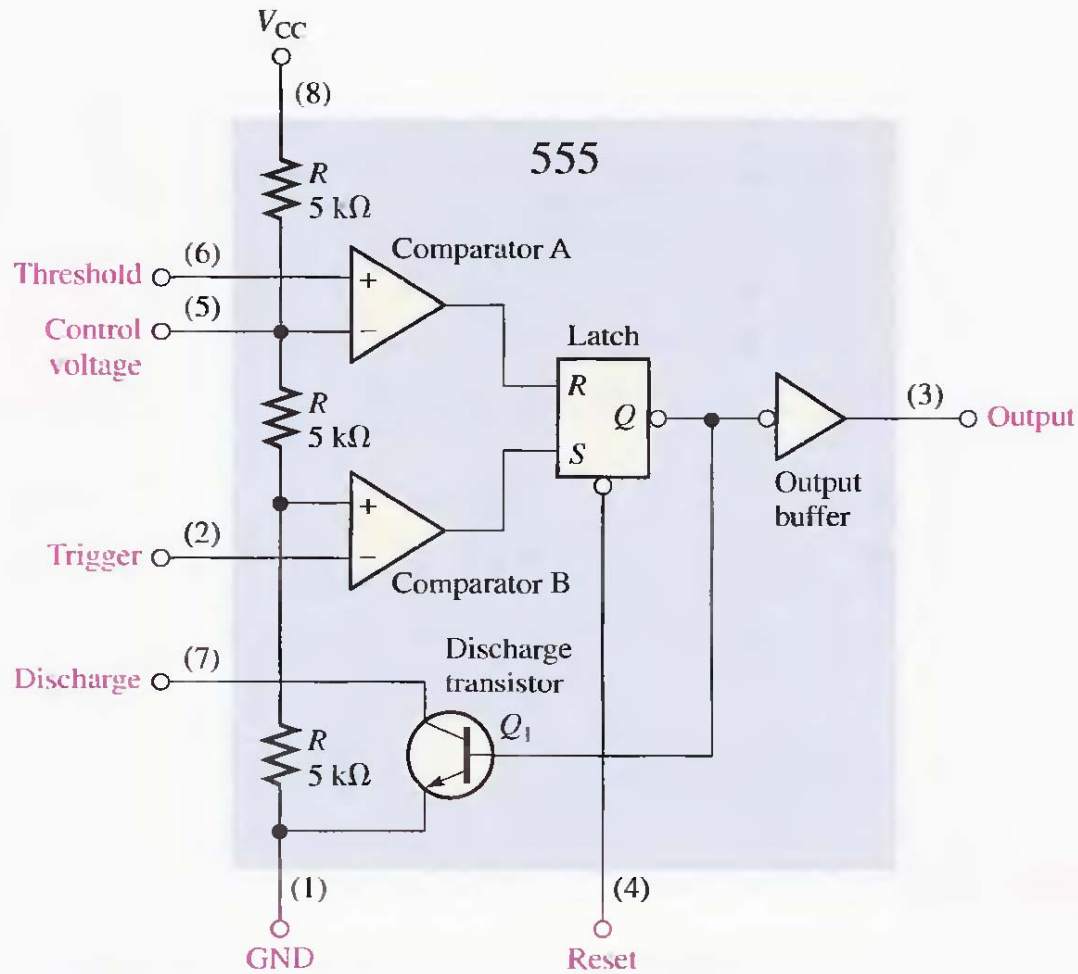
7–6 THE 555 TIMER

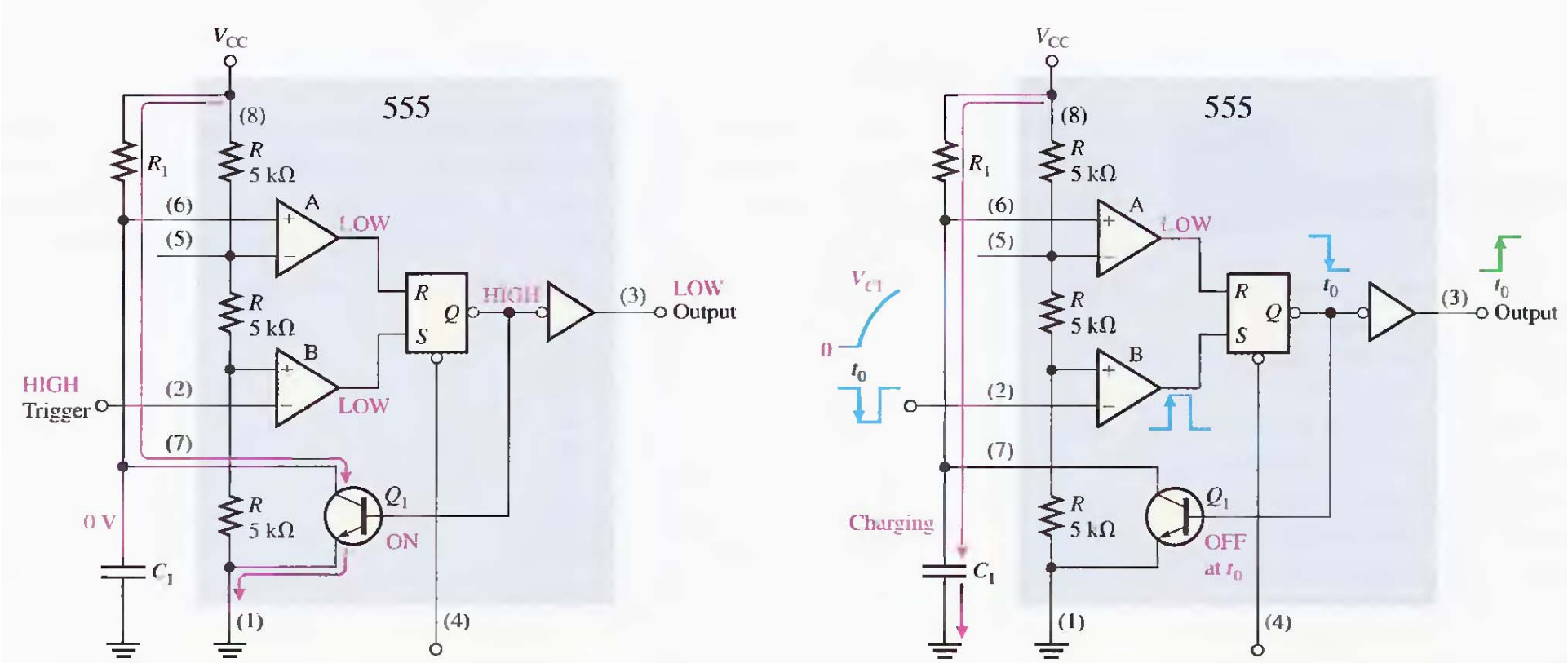
The 555 **timer** is a versatile and widely used IC device because it can be configured in two different modes as either a monostable multivibrator (one-shot) or as an astable multivibrator (oscillator). An astable multivibrator has no stable states and therefore changes back and forth (oscillates) between two unstable states without any external triggering.

Monostable (One-Shot) Operation

$t_W = 1.1R_1C_1$

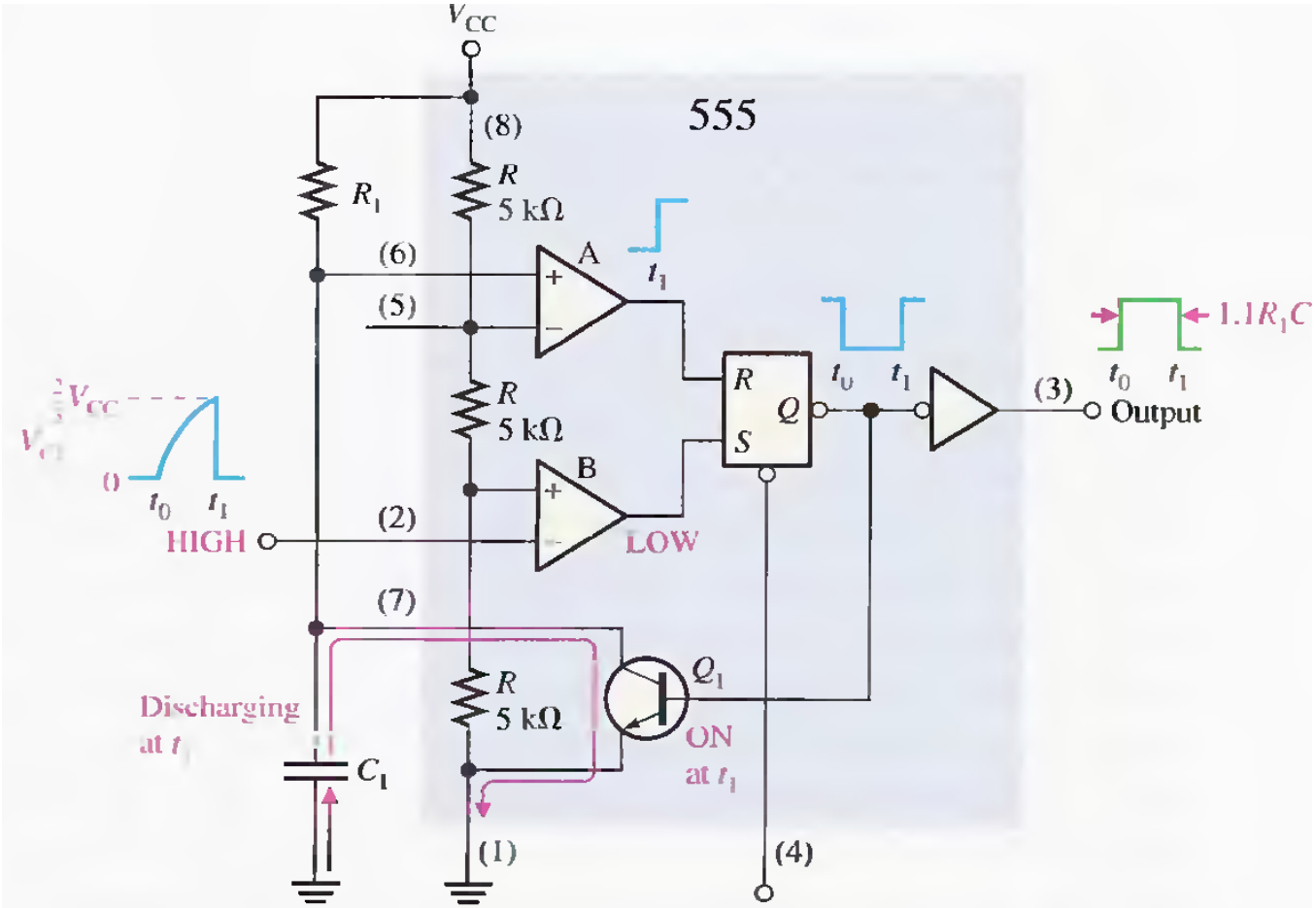






(a) Prior to triggering. (The current path is indicated by the red arrow.)

(b) When triggered



(c) At end of charging interval

EXAMPLE 7–14

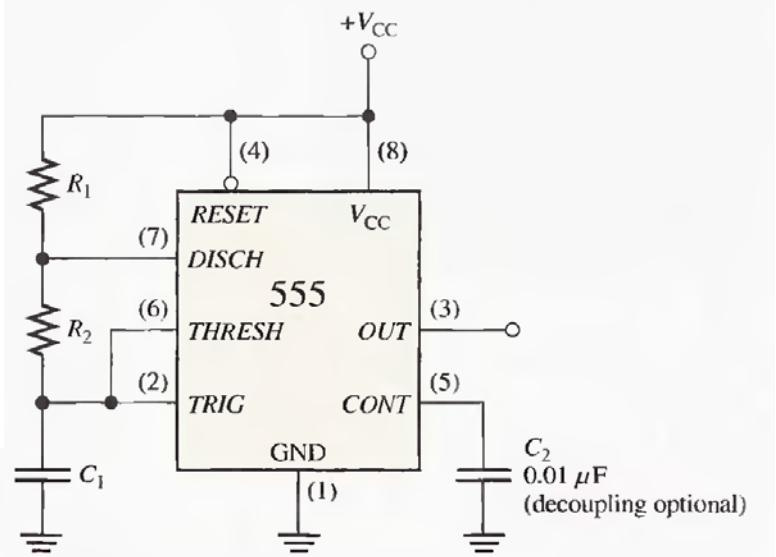
What is the output pulse width for a 555 monostable circuit with $R_1 = 2.2\ \text{k}\Omega$ and $C_1 = 0.01\ \mu\text{F}$?

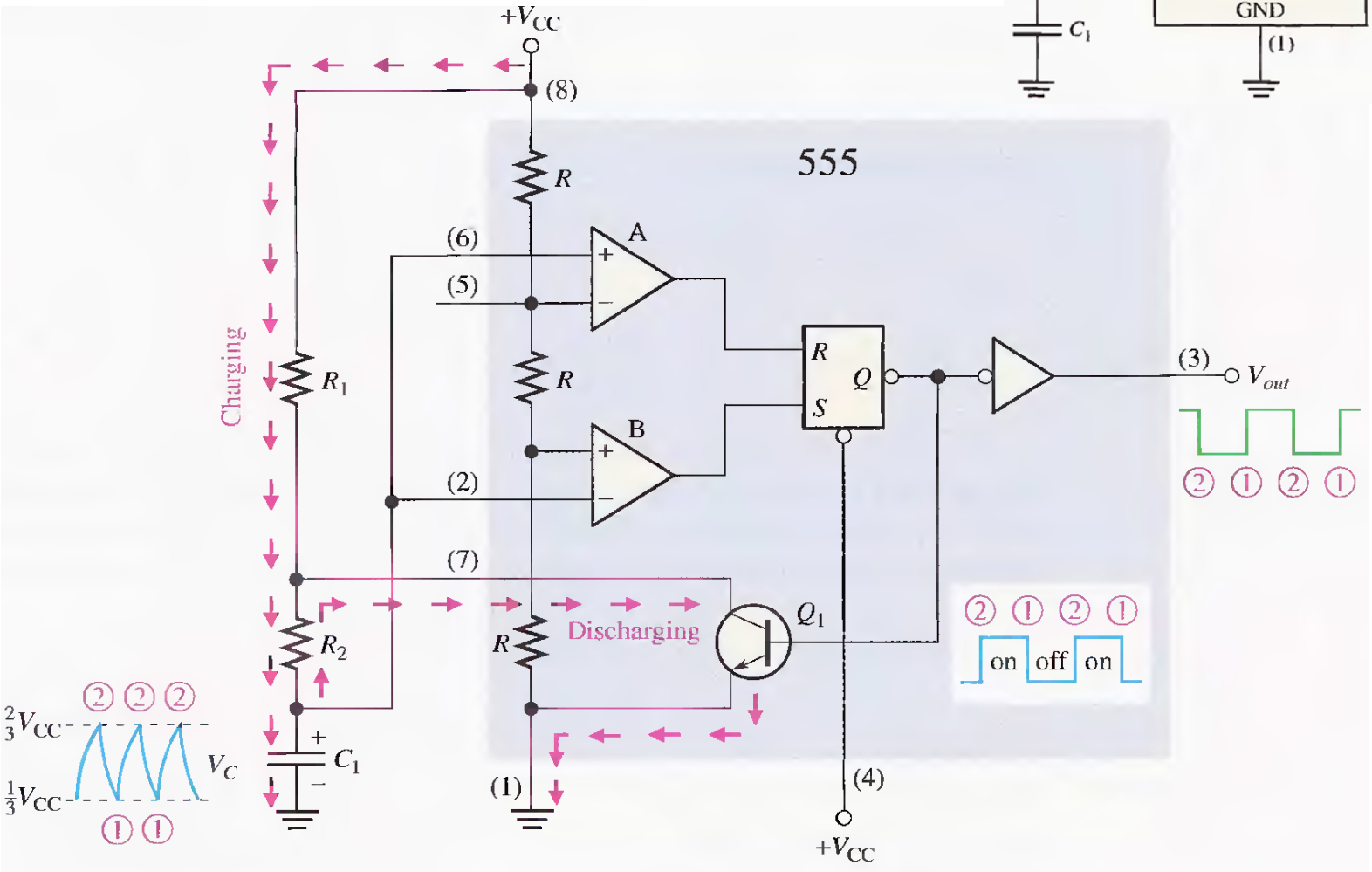
From Equation 7–3 the pulse width is

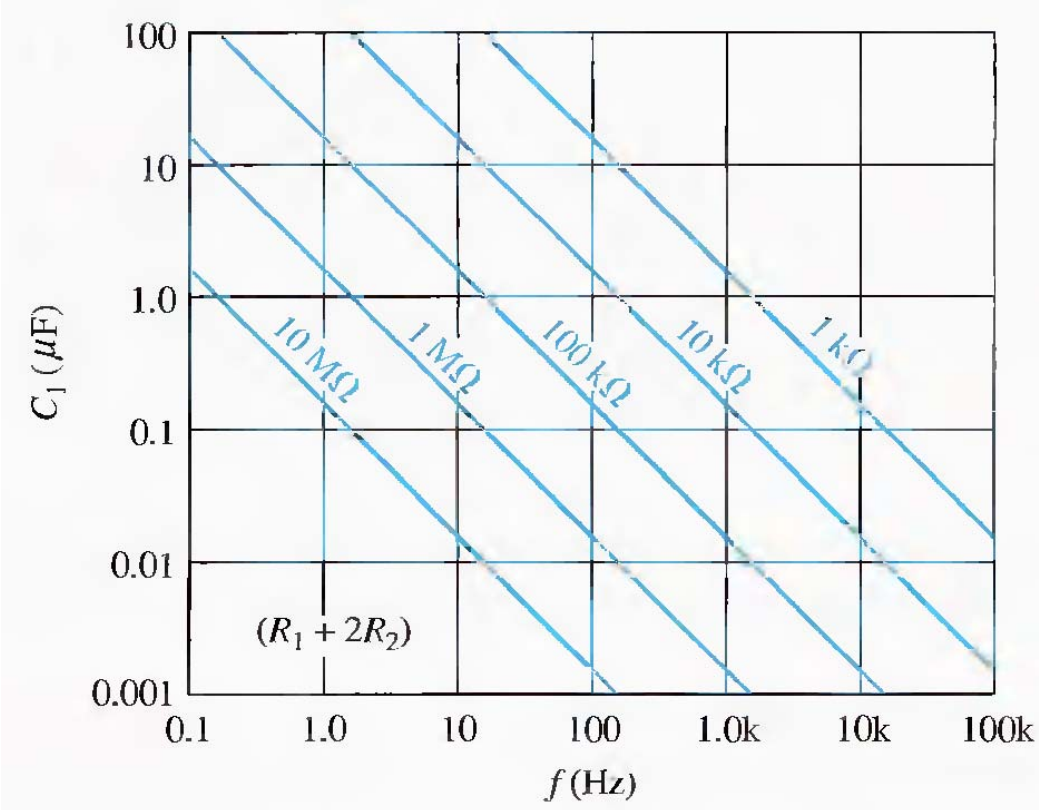
$$t_W = 1.1R_1C_1 = 1.1(2.2\ \text{k}\Omega)(0.01\ \mu\text{F}) = \mathbf{24.2\ \mu s}$$

Astable Operation

$$f = \frac{1.44}{(R_t + 2R_2)C_1}$$







Frequency of oscillation as a function of C_1 and $R_1 + 2R_2$. The sloped lines are values of $R_1 + 2R_2$.

$$t_H = 0.7(R_1 + R_2)C_1$$

$$t_L = 0.7R_2C_1$$

$$T = t_H + t_L = 0.7(R_1 + 2R_2)C_1$$

$$\text{Duty cycle} = \frac{t_H}{T} = \frac{t_H}{t_H + t_L}$$

$$\text{Duty cycle} = \left(\frac{R_1 + R_2}{R_1 + 2R_2}\right)100\%$$

To achieve duty cycles of less than 50 percent, the circuit in Figure 7–56 can be modified so that C_1 charges through only R_1 and discharges through R_2. This is achieved with a diode, D_1, placed as shown in Figure 7–59. The duty cycle can be made less than 50 percent by making R_1 less than R_2. Under this condition, the expression for the duty cycle is

$$\text{Duty cycle} = \left(\frac{R_1}{R_1 + R_2}\right)100\%$$

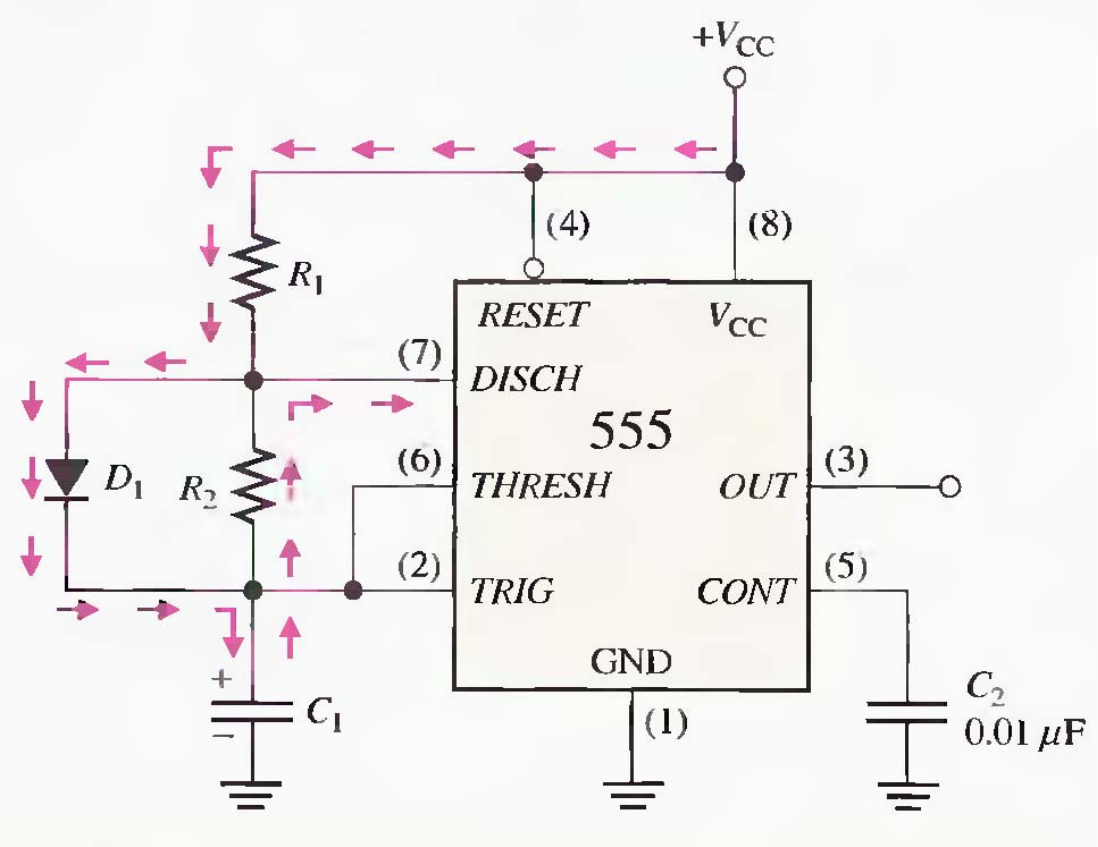


FIGURE 7–59

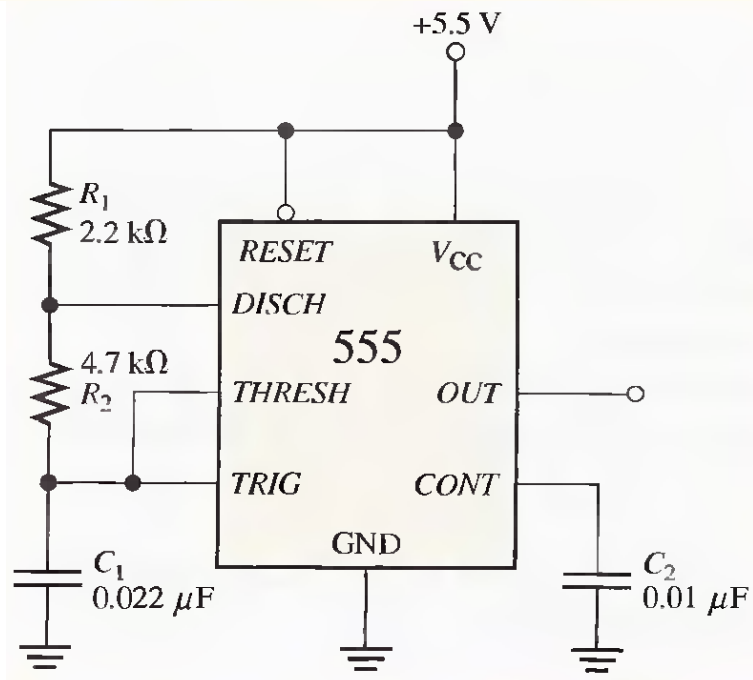
The addition of diode D_1 allows the duty cycle of the output to be adjusted to less than 50 percent by making $R_1 < R_2$.

COMPUTER NOTE

All computers require a timing source to provide accurate clock waveforms. The timing section controls all system timing and is responsible for the proper operation of the system hardware. The timing section usually consists of a crystal-controlled oscillator and counters for frequency division. Using a high-frequency oscillator divided down to a lower frequency provides for greater accuracy and frequency stability.

EXAMPLE 7–15

A 555 timer configured to run in the astable mode (oscillator) is shown in Figure 7–60. Determine the frequency of the output and the duty cycle.



Solution Use Equations 7–4 and 7–7.

$$f = \frac{1.44}{(R_1 + 2R_2)C_1} = \frac{1.44}{(2.2\ \text{k}\Omega + 9.4\ \text{k}\Omega)0.022\ \mu\text{F}} = \mathbf{5.64\ kHz}$$

$$\text{Duty cycle} = \left(\frac{R_1 + R_2}{R_1 + 2R_2}\right)100\% = \left(\frac{2.2\ \text{k}\Omega + 4.7\ \text{k}\Omega}{2.2\ \text{k}\Omega + 9.4\ \text{k}\Omega}\right)100\% = \mathbf{59.5\%}$$

SECTION 7-6 REVIEW

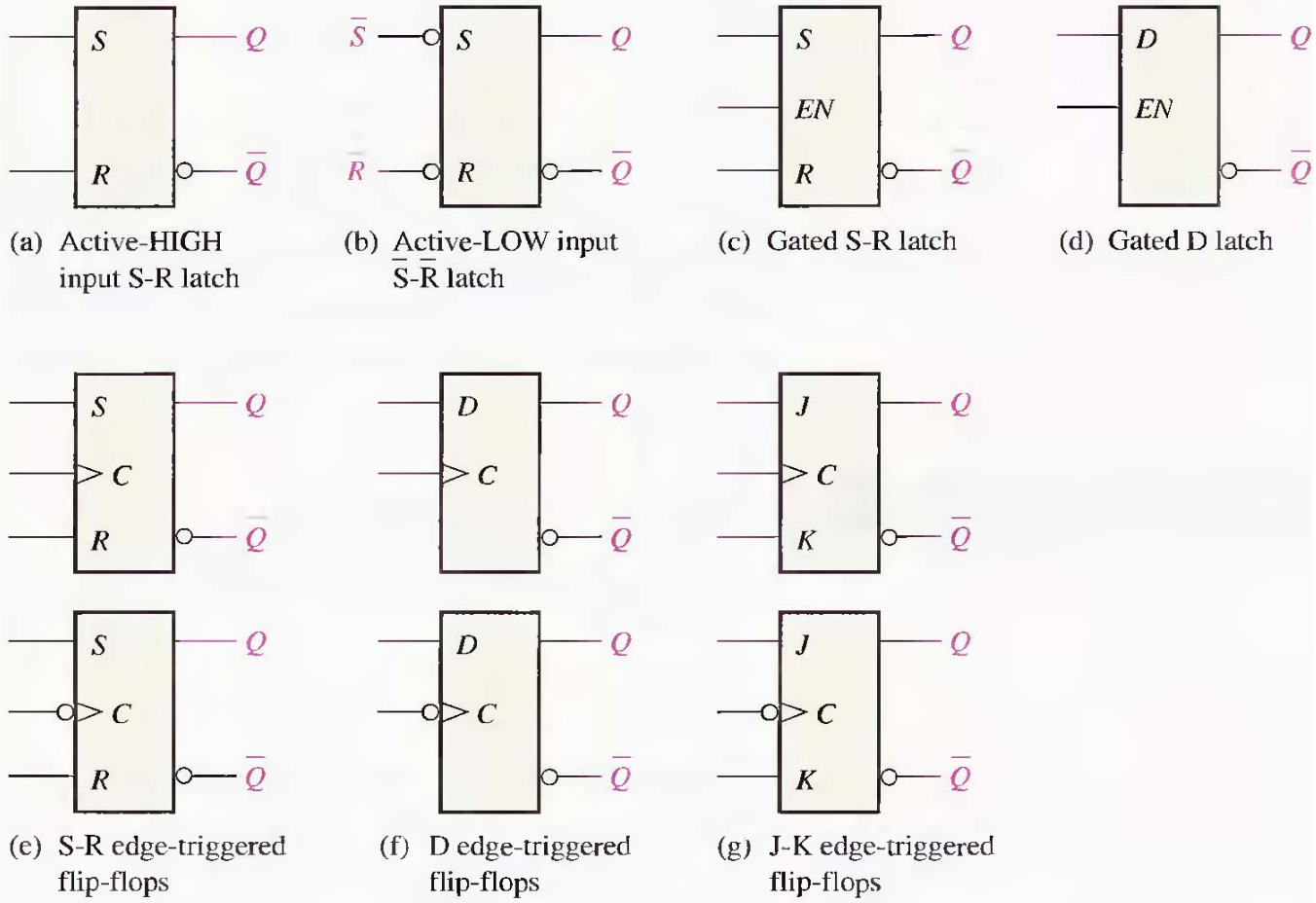
1. Explain the difference in operation between an astable multivibrator and a monostable multivibrator.
2. For a certain astable multivibrator, $t_H = 15$ ms and $T = 20$ ms. What is the duty cycle of the output?

7-7 TROUBLESHOOTING

DIGITAL SYSTEM APPLICATION

SUMMARY

- Symbols for latches and flip-flops are shown in Figure 7–67.



(a) Active-HIGH input S-R latch

(b) Active-LOW input $\overline{S}$-$\overline{R}$ latch

(c) Gated S-R latch

(d) Gated D latch

(e) S-R edge-triggered flip-flops

(f) D edge-triggered flip-flops

(g) J-K edge-triggered flip-flops

- Latches are bistable devices whose state normally depends on asynchronous inputs.
- Edge-triggered flip-flops are bistable devices with synchronous inputs whose state depends on the inputs only at the triggering transition of a clock pulse. Changes in the outputs occur at the triggering transition of the clock.
- Monostable multivibrators (one-shots) have one stable state. When the one-shot is triggered, the output goes to its unstable state for a time determined by an *RC* circuit.
- Astable multivibrators have no stable states and are used as oscillators to generate timing waveforms in digital systems.

KEY TERMS

Astable Having no stable state. An astable multivibrator oscillates between two quasi-stable states.

Bistable Having two stable states. Flip-flops and latches are bistable multivibrators.

Clear An asynchronous input used to reset a flip-flop (make the Q output 0).

Clock The triggering input of a flip-flop.

D flip-flop A type of bistable multivibrator in which the output assumes the state of the D input on the triggering edge of a clock pulse.

Edge-triggered flip-flop A type of flip-flop in which the data are entered and appear on the output on the same clock edge.

Hold time The time interval required for the control levels to remain on the inputs to a flip-flop after the triggering edge of the clock in order to reliably activate the device.

J-K flip-flop A type of flip-flop that can operate in the SET, RESET, no-change, and toggle modes.

Latch A bistable digital circuit used for storing a bit.

Monostable Having only one stable state. A monostable multivibrator, commonly called a *one-shot,* produces a single pulse in response to a triggering input.

One-shot A monostable multivibrator.

Power dissipation The amount of power required by a circuit.

Preset An asynchronous input used to set a flip-flop (make the Q output 1).

Propagation delay time The interval of time required after an input signal has been applied for the resulting output change to occur.

RESET The state of a flip-flop or latch when the output is 0; the action of producing a RESET state.

SET The state of a flip-flop or latch when the output is 1; the action of producing a SET state.

Set-up time The time interval required for the control levels to be on the inputs to a digital circuit, such as a flip-flop, prior to the triggering edge of a clock pulse.

Synchronous Having a fixed time relationship.

Timer A circuit that can be used as a one-shot or as an oscillator.

Toggle The action of a flip-flop when it changes state on each clock pulse.

به نام خدا

فصل هشتم: شمارنده ها

8 COUNTERS

CHAPTER OUTLINE

8–1 Asynchronous Counter Operation
8–2 Synchronous Counter Operation
8–3 Up/Down Synchronous Counters
8–4 Design of Synchronous Counters
8–5 Cascaded Counters
8–6 Counter Decoding
8–7 Counter Applications
8–8 Logic Symbols with Dependency Notation
8–9 Troubleshooting

CHAPTER OBJECTIVES

- Describe the difference between an asynchronous and a synchronous counter
- Analyze counter timing diagrams
- Analyze counter circuits
- Explain how propagation delays affect the operation of a counter
- Determine the modulus of a counter
- Modify the modulus of a counter
- Recognize the difference between a 4-bit binary counter and a decade counter
- Use an up/down counter to generate forward and reverse binary sequences
- Determine the sequence of a counter
- Use IC counters in various applications
- Design a counter that will have any specified sequence of states
- Use cascaded counters to achieve a higher modulus
- Use logic gates to decode any given state of a counter
- Eliminate glitches in counter decoding
- Explain how a digital clock operates
- Interpret counter logic symbols that use dependency notation
- Troubleshoot counters for various types of faults

KEY TERMS

- Asynchronous
- Recycle
- Modulus
- Decade
- Synchronous
- Terminal count
- State machine
- State diagram
- Cascade

INTRODUCTION

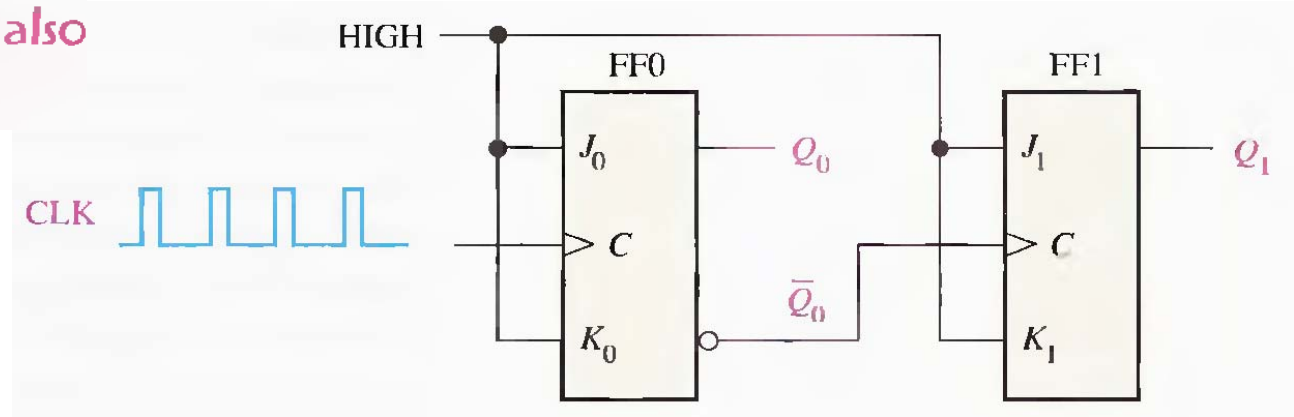
As you learned in Chapter 7, flip-flops can be connected together to perform counting operations. Such a group of flip-flops is a counter. The number of flip-flops used and the way in which they are connected determine the number of states (called the modulus) and also the specific sequence of states that the counter goes through during each complete cycle.

Counters are classified into two broad categories according to the way they are clocked: asynchronous and synchronous. In asynchronous counters, commonly called *ripple counters,* the first flip-flop is clocked by the external clock pulse and then each successive flip-flop is clocked by the output of the preceding flip-flop. In synchronous counters, the clock input is connected to all of the flip-flops so that they are clocked simultaneously. Within each of these two categories, counters are classified primarily by the type of sequence, the number of states, or the number of flip-flops in the counter.

8–1 ASYNCHRONOUS COUNTER OPERATION

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.

Asynchronous counters are also known as ripple counters.



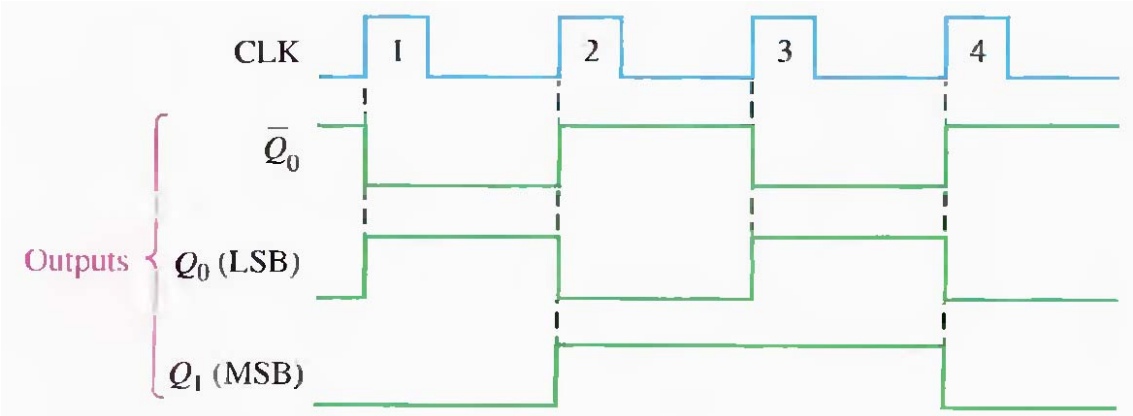
1- all j=k=1

2- کلاک اصلی به ورودی کلاک طبقه اول فقط وارد شده

3- خروجی مکمل هر طبقه کلاک طبقه بعدی را تامین می کند.

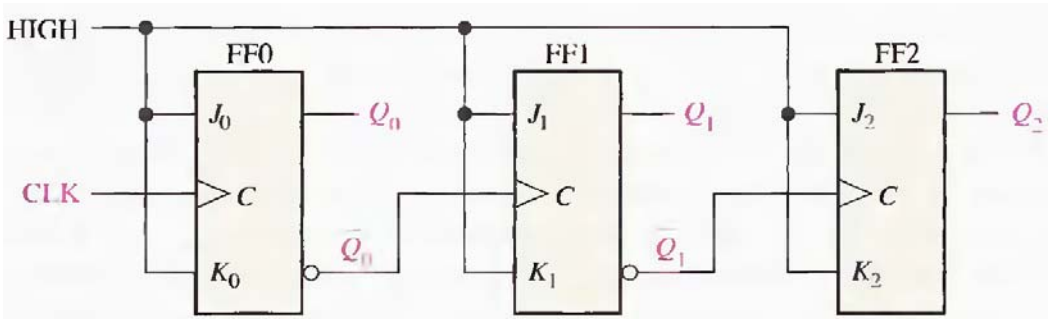
4- فرکانس خروجی هر طبقه نصف فرکانس کلاک ورودی همان طبقه است.

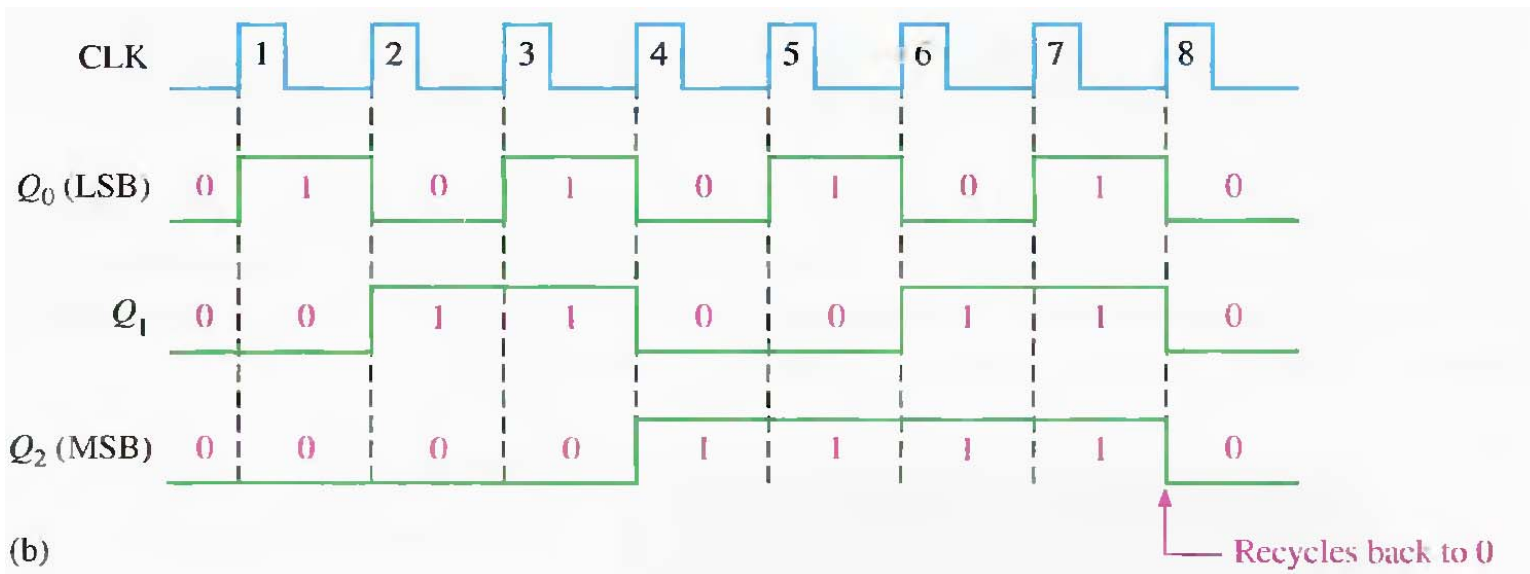
5- اگر کلاک FF ها فعال با لبه منفی باشد باید خود خروجیها به کلاک طبقه بعد متصل شوند.



CLOCK PULSE	Q_1	Q_0
Initially	0	0
1	0	1
2	1	0
3	1	1
4 (recycles)	0	0

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





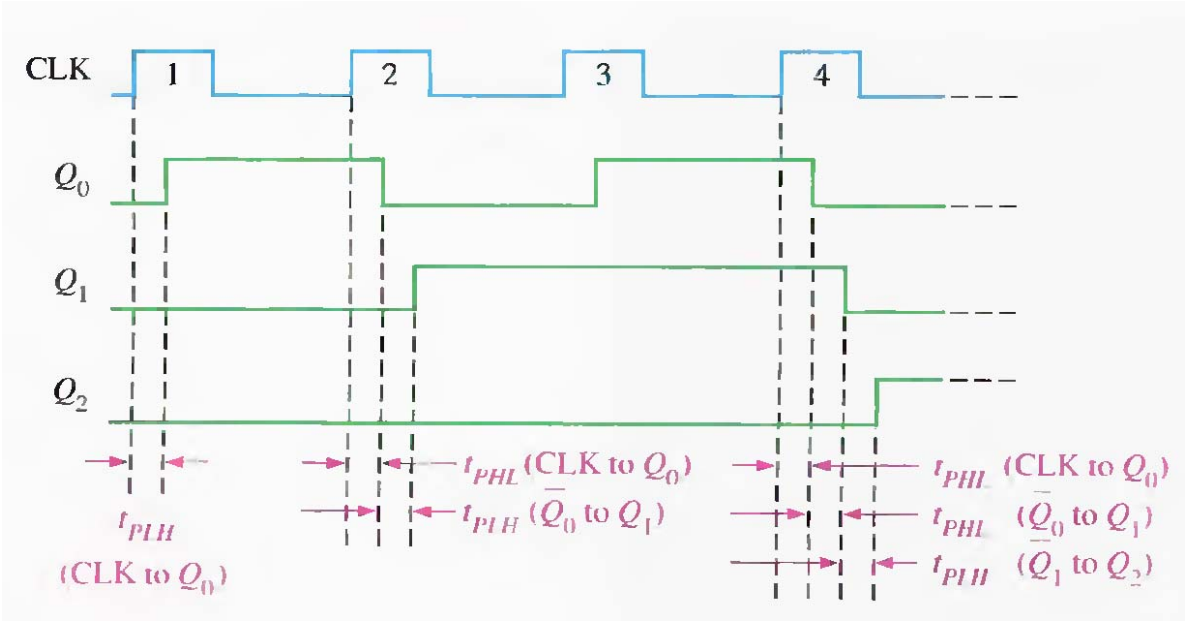
(b)

Propagation Delay

$T_{min} > Nt_p$

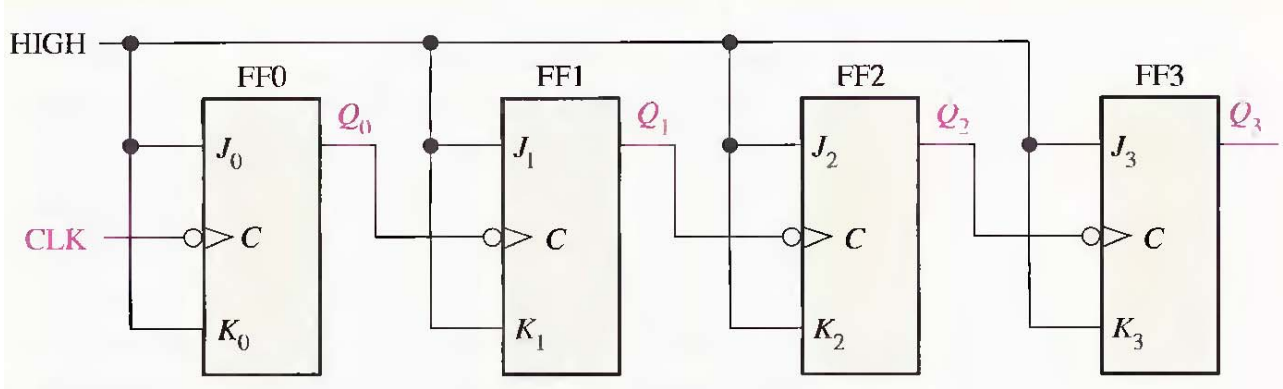
Or

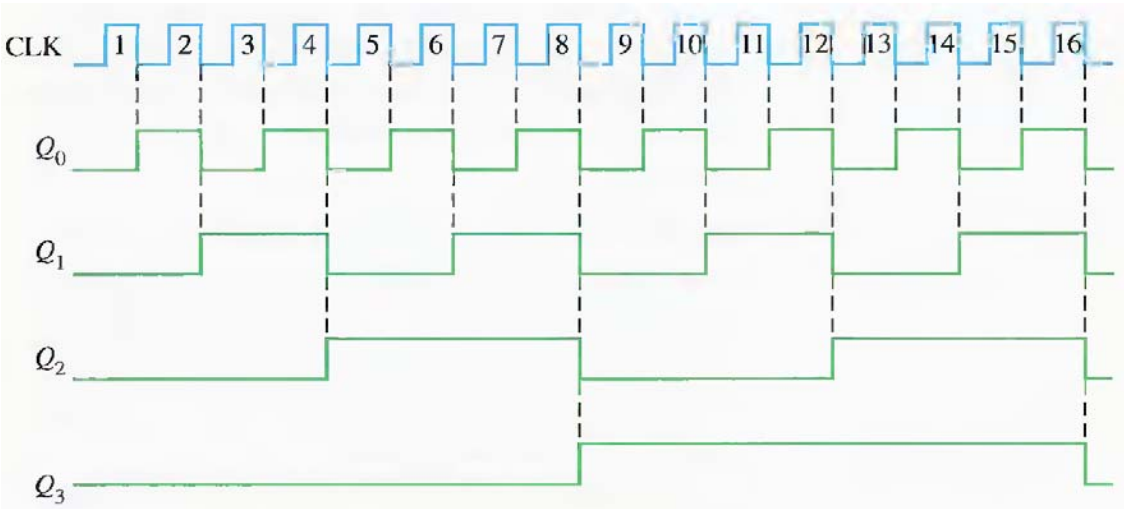
$f_{max} < 1/(Nt_p)$



EXAMPLE 8–1

A 4-bit asynchronous binary counter is shown in Figure 8–5(a). Each 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 counter can have 2^n states, where n is the number of flip-flops.

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

The maximum clock frequency is

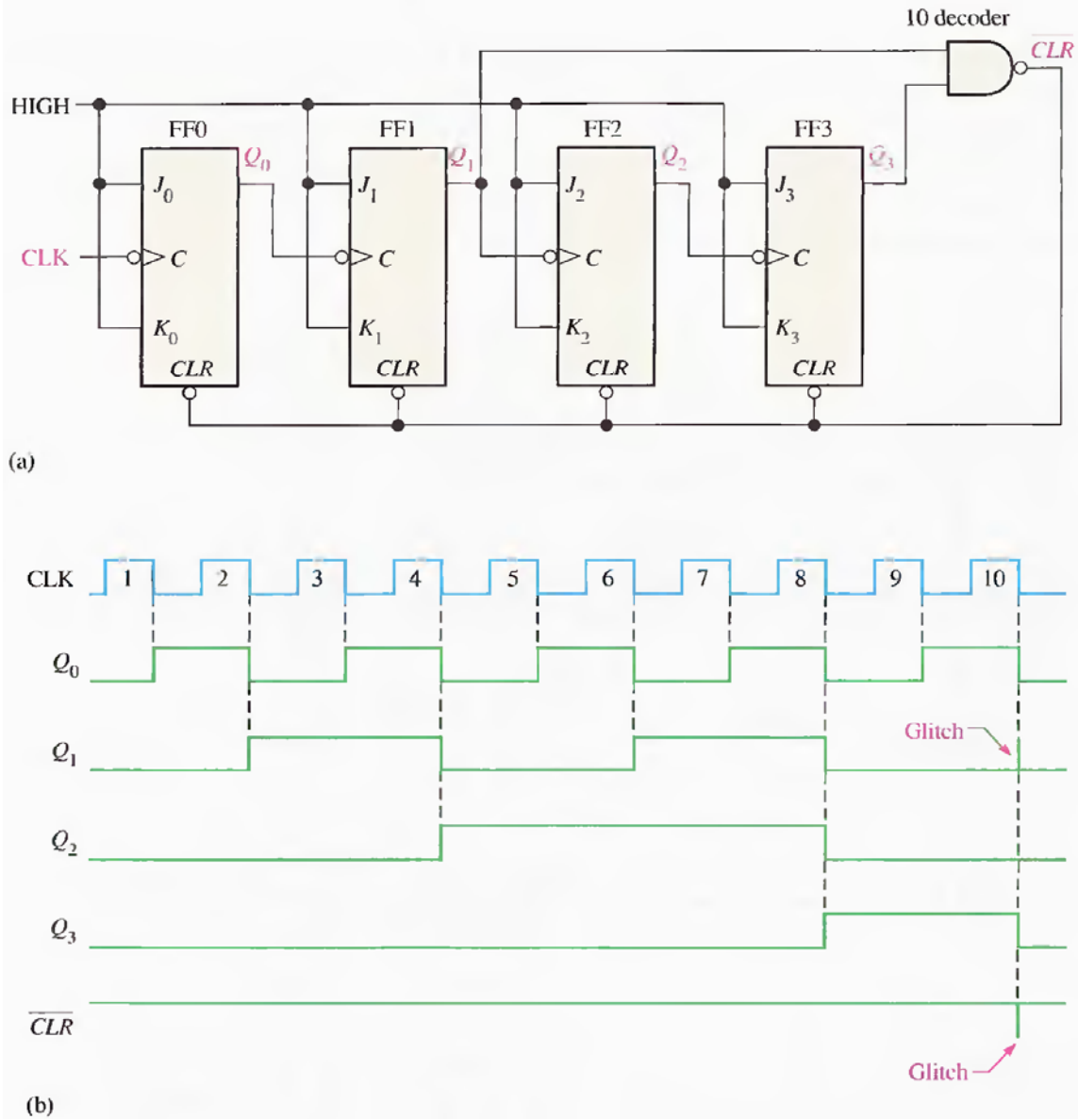
$$f_{max} = \frac{1}{t_{p(tot)}} = \frac{1}{40 \text{ ns}} = \mathbf{25 \text{ MHz}}$$

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,

شمارنده BCD یکی از انواع شمارنده Decade است ولی در کل هر شمارنده ای با 10 حالت پایدار را شمارنده مد 10 یا decade counter می نامند.

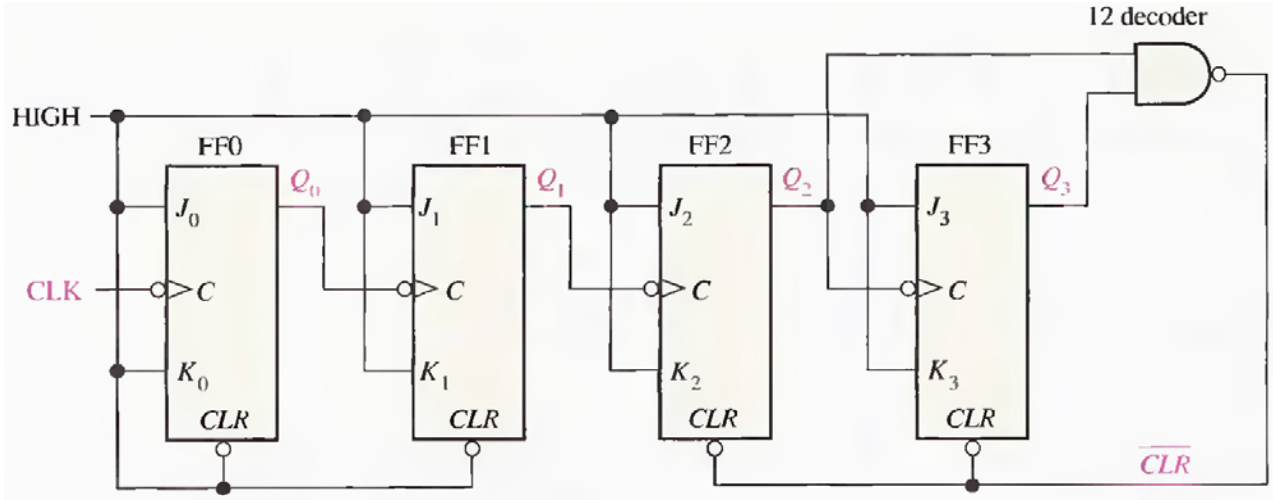
مثل شمارنده 4017

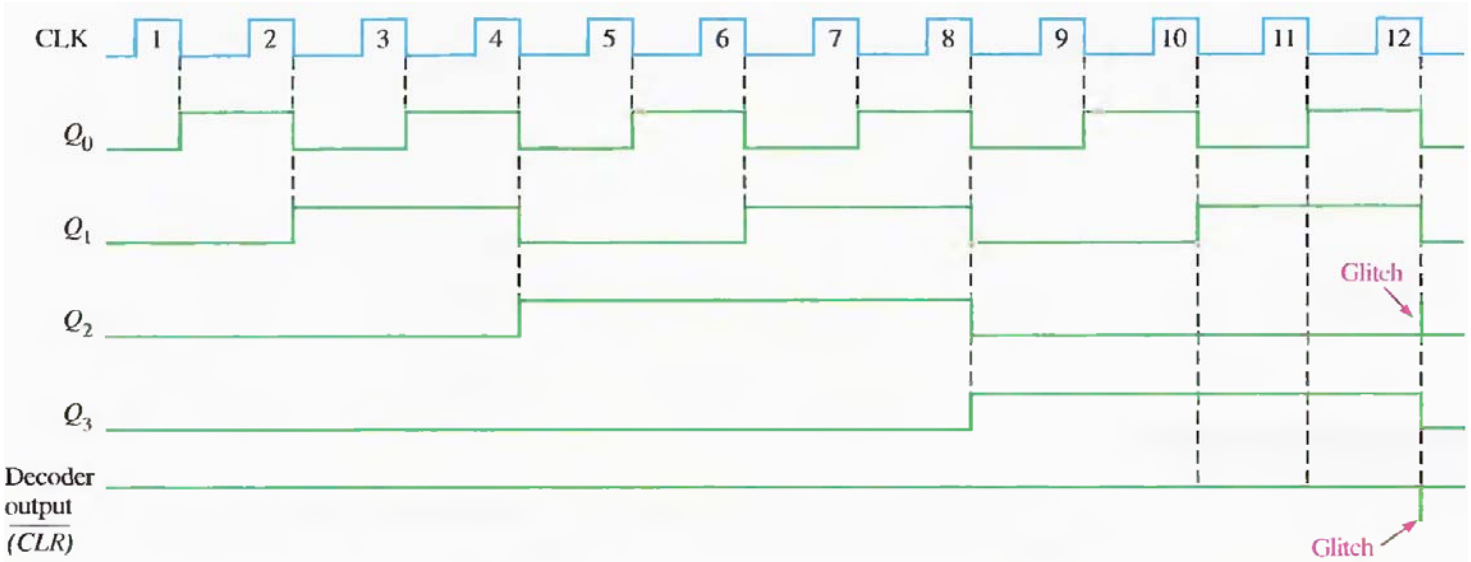


EXAMPLE 8–2

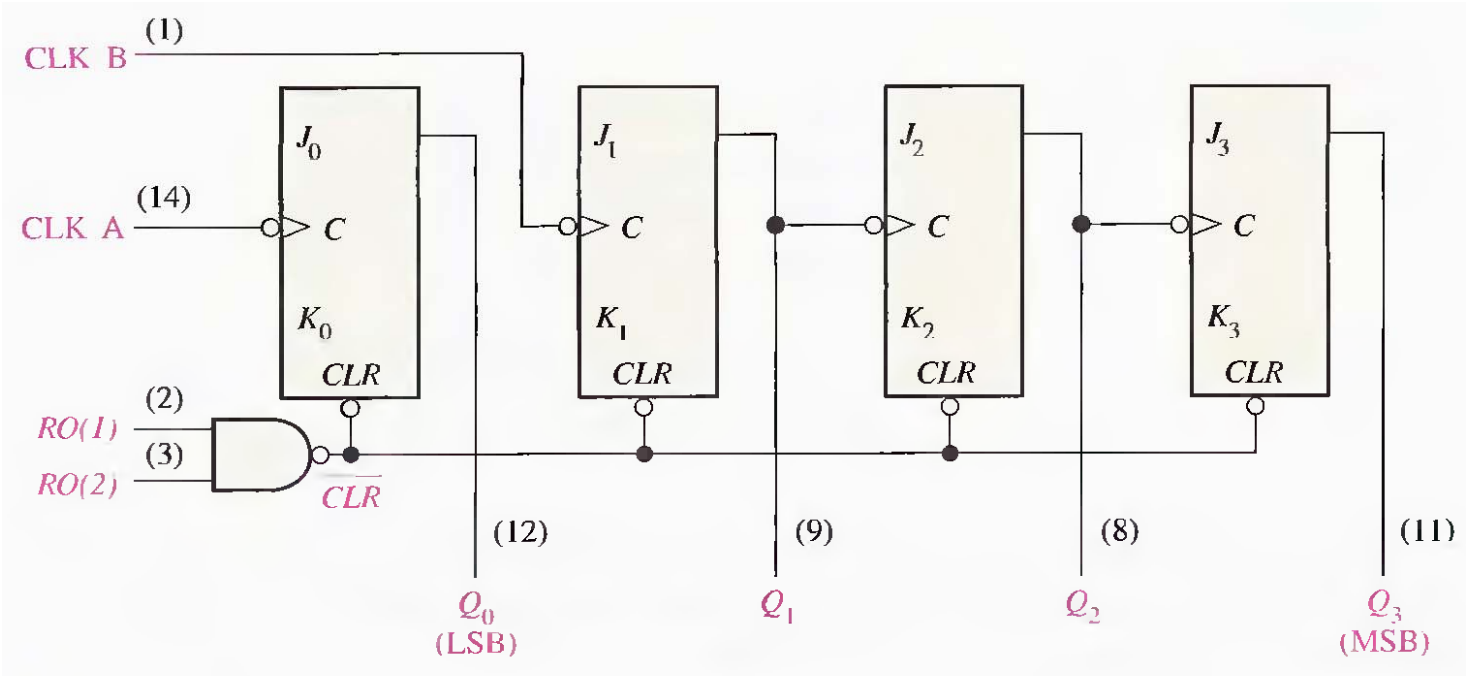
Show how an asynchronous counter can be implemented having a modulus of twelve with a straight binary sequence from 0000 through 1011.

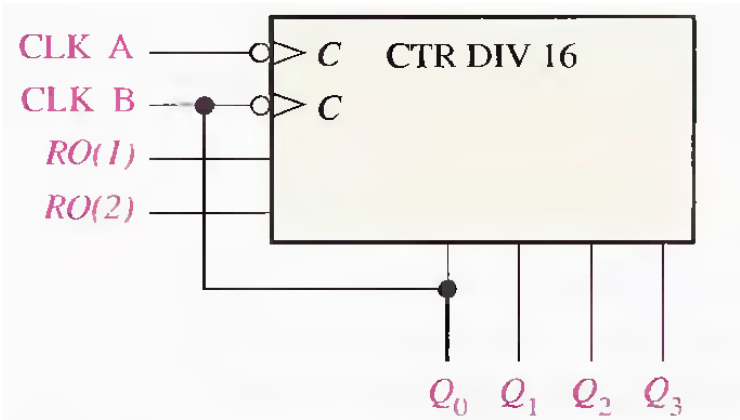
Q_3	Q_2	Q_1	Q_0	
0	0	0	0	Recycles
.	.	.	.	
.	.	.	.	
.	.	.	.	
1	0	1	1	
1	1	0	0	Normal next state



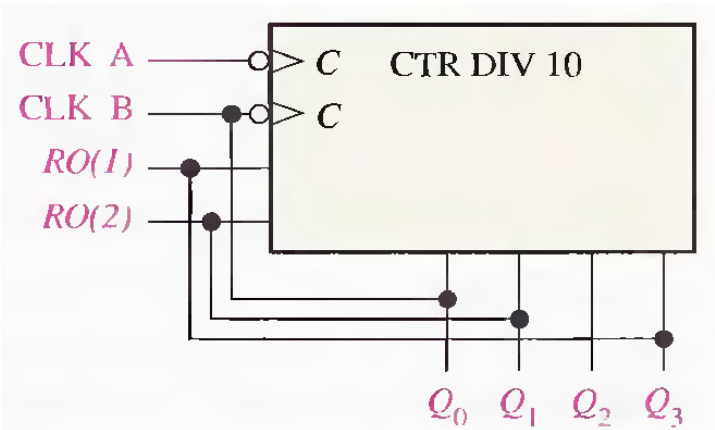


THE 74LS93 4-BIT ASYNCHRONOUS BINARY COUNTER





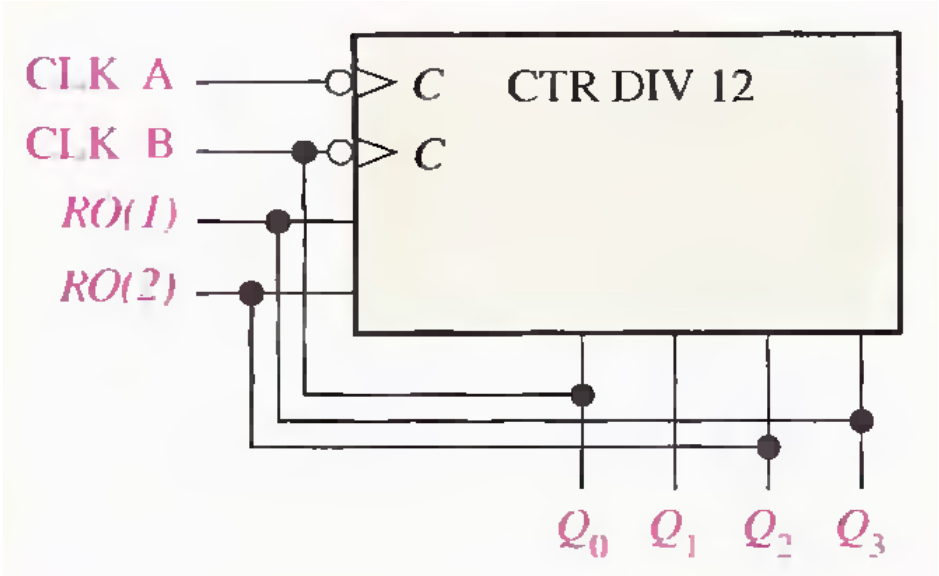
(a) 74LS93 connected as a modulus-16 counter



(b) 74LS93 connected as a decade counter

EXAMPLE 8–3

Show how the 74LS93 can be used as a modulus-12 counter.

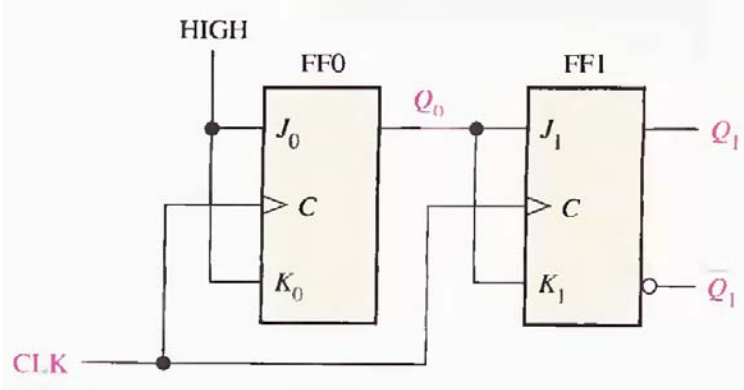


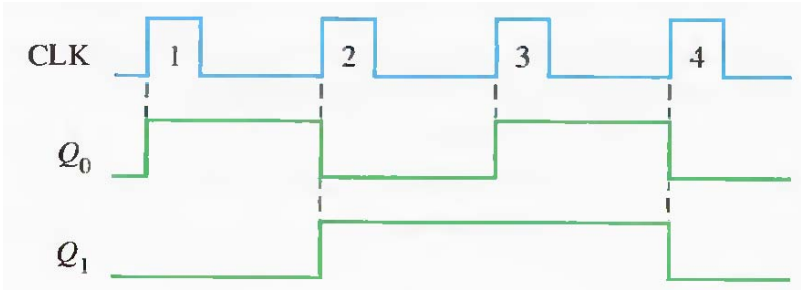
8–2 SYNCHRONOUS COUNTER OPERATION

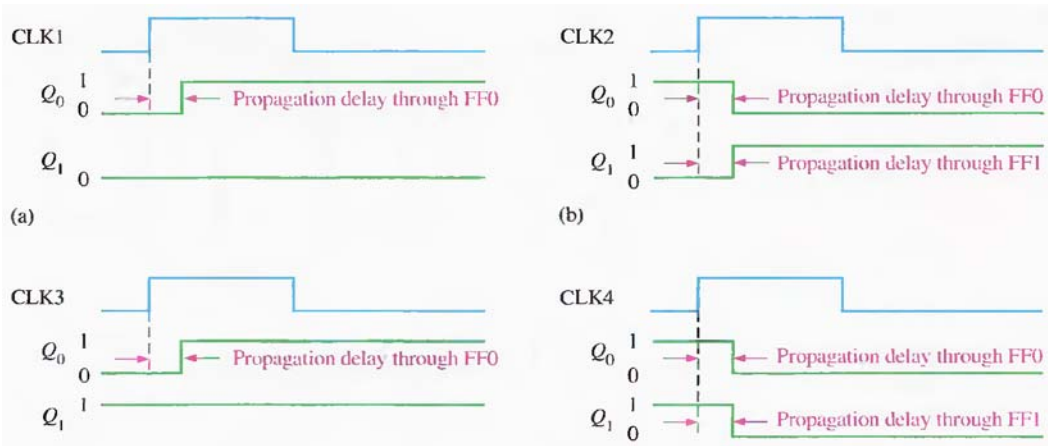
The term **synchronous** refers to events that have a fixed time relationship with each other. A **synchronous counter** is one in which all the flip-flops in the counter are clocked at the same time by a common clock pulse.

Q2	Q1	Q0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1
0	0	0

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

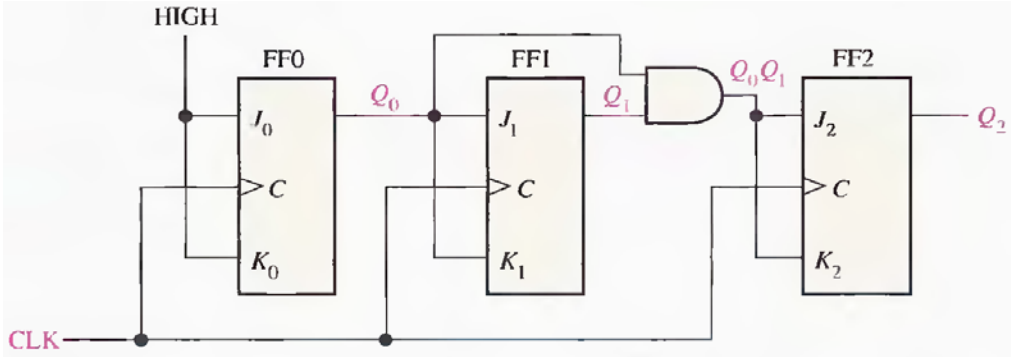


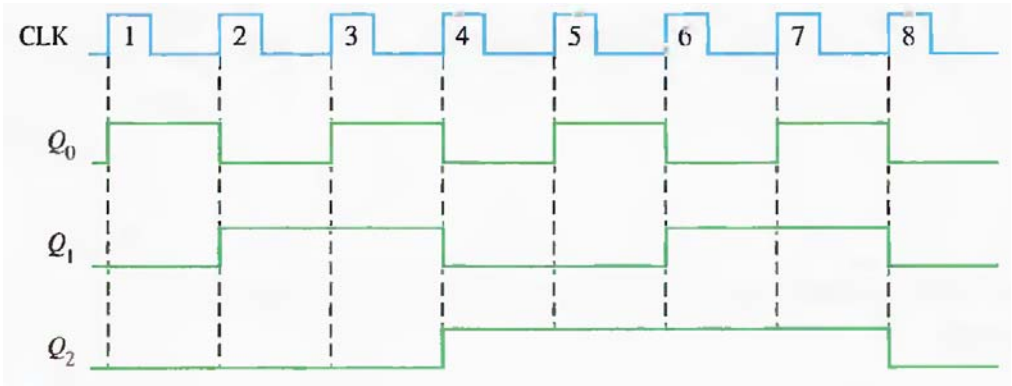




A 3-Bit Synchronous Binary Counter

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

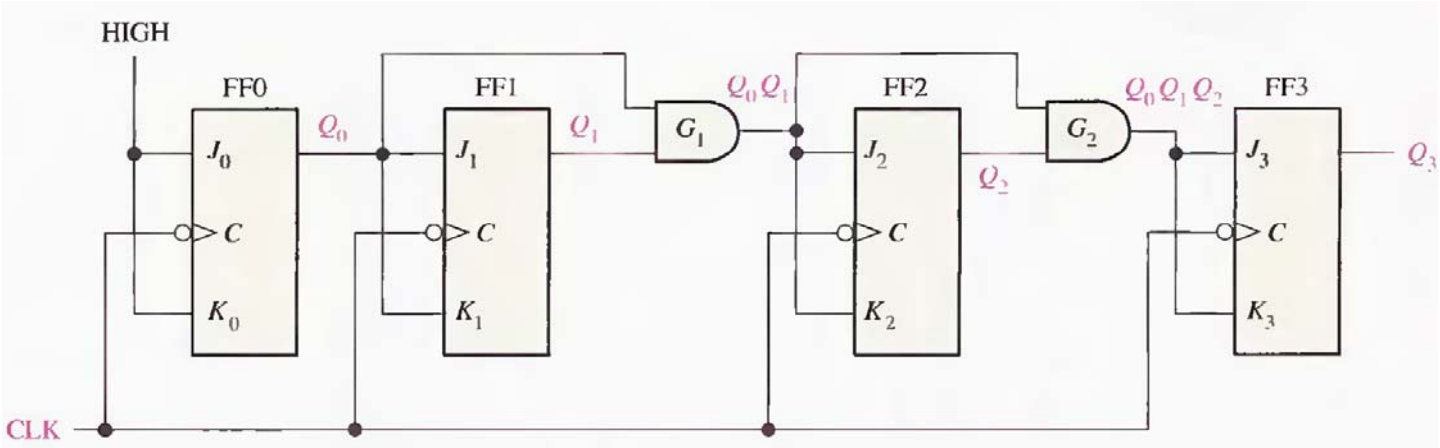


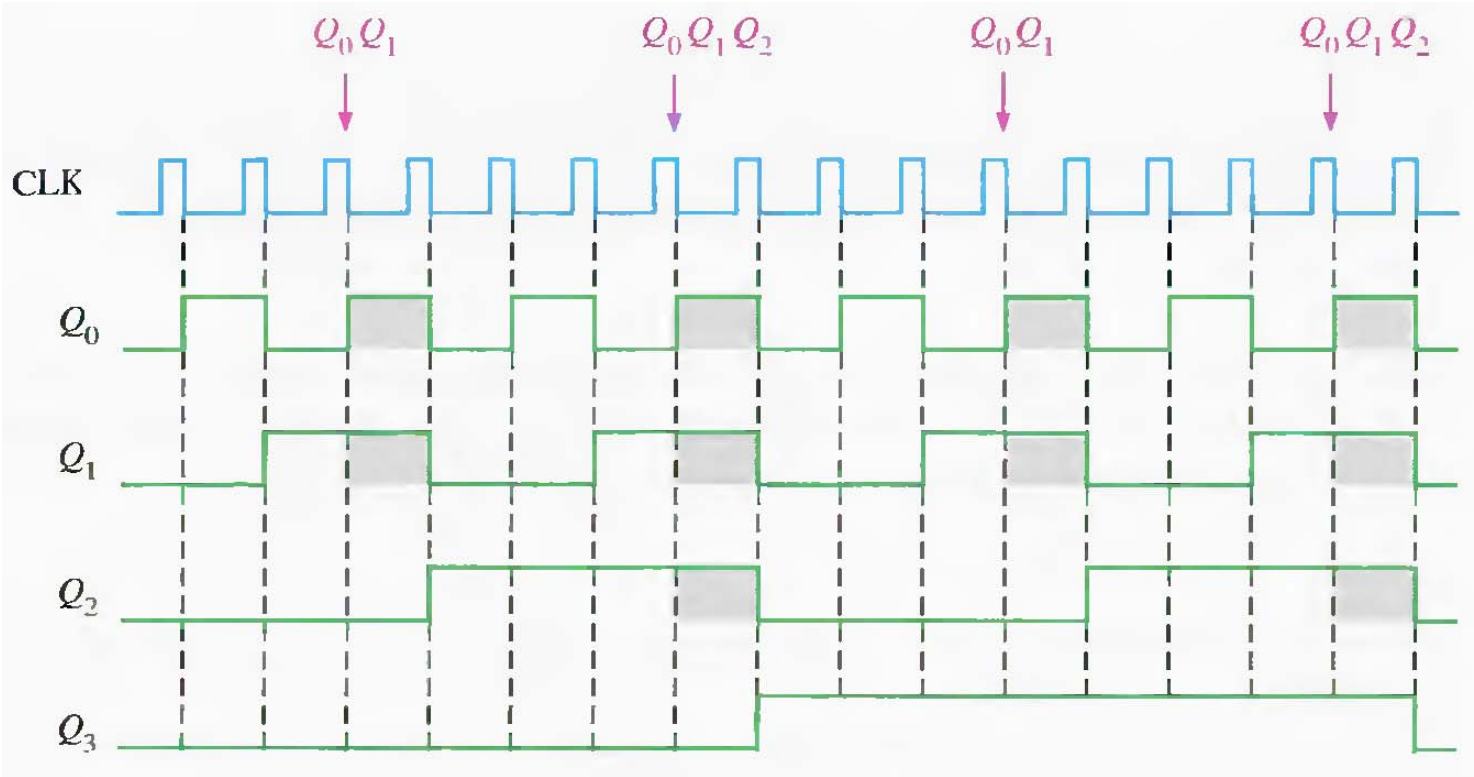


COMPUTER NOTE

The TSC or *time stamp counter* in the Pentium is used for performance monitoring, which enables a number of parameters important to the overall performance of a Pentium system to be determined exactly. By reading the TSC before and after the execution of a procedure, the precise time required for the procedure can be determined based on the processor cycle time. In this way, the TSC forms the basis for all time evaluations in connection with optimizing system operation. For example, it can be accurately determined which of two or more programming sequences is more efficient. This is a very useful tool for compiler developers and system programmers in producing the most effective code for the Pentium.

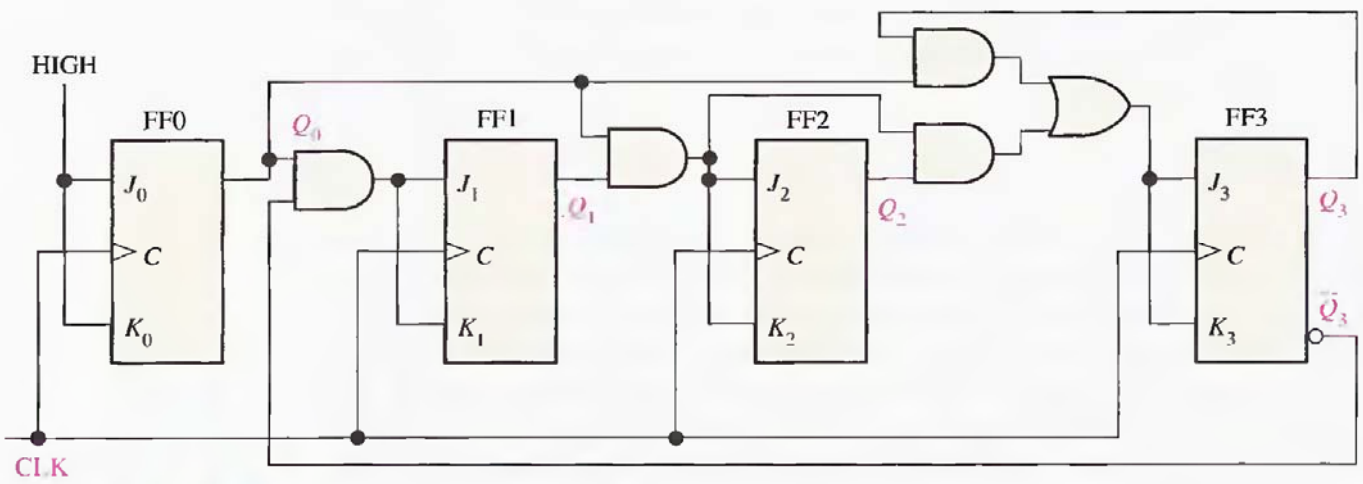
A 4-Bit Synchronous Binary Counter





A 4-Bit Synchronous 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

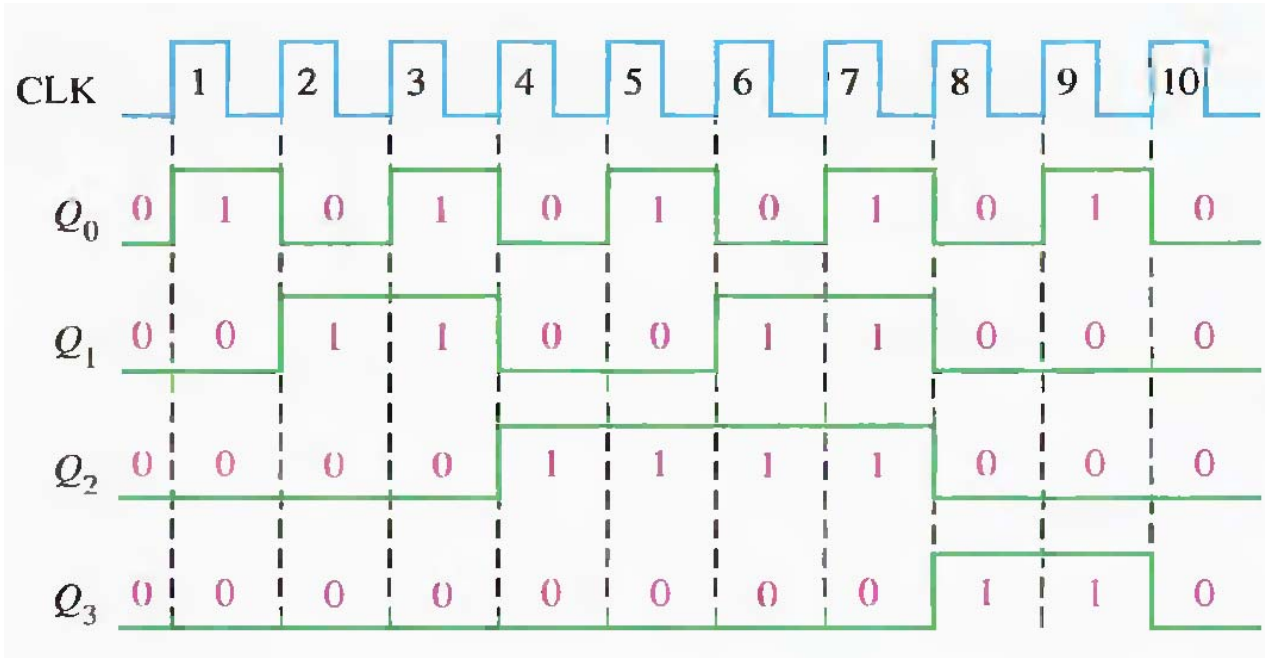


$$J_0 = K_0 = 1$$

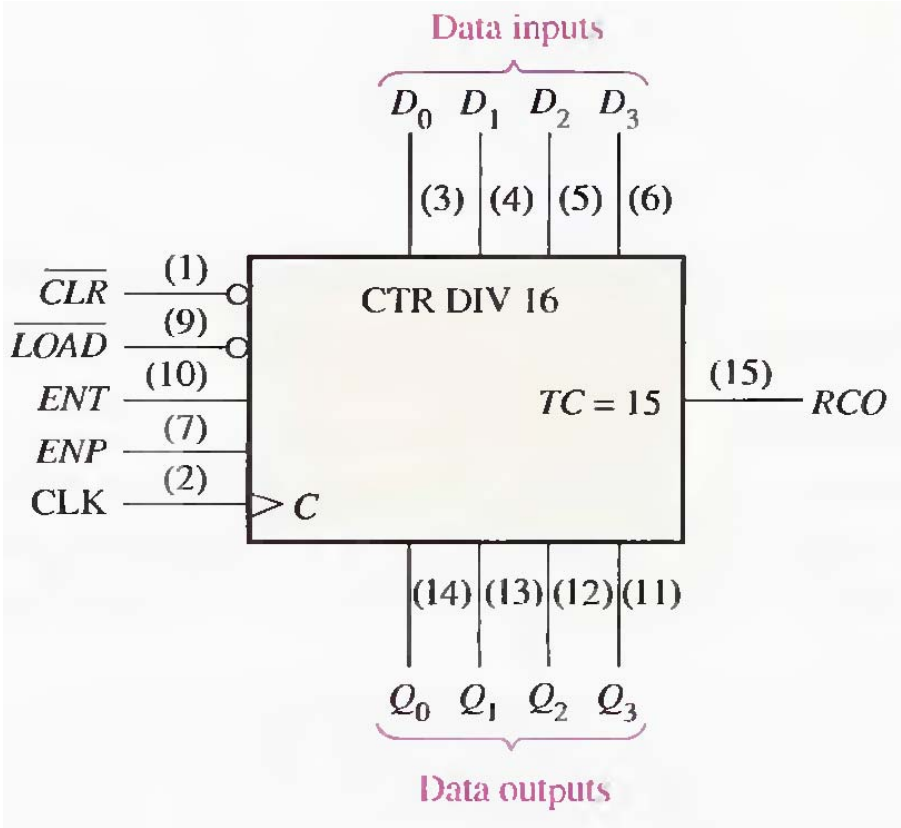
$$J_1 = K_1 = Q_0\overline{Q}_3$$

$$J_2 = K_2 = Q_0Q_1$$

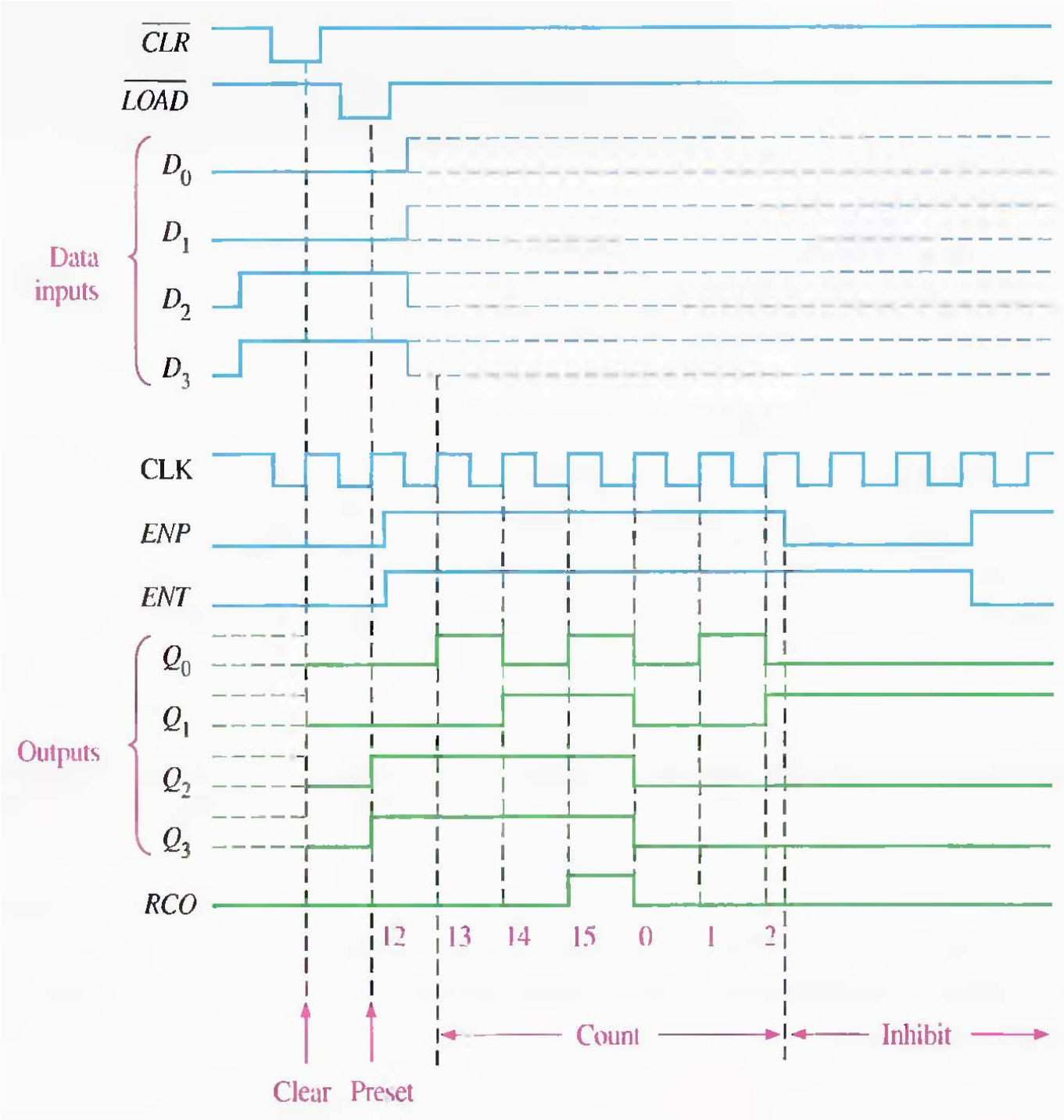
$$J_3 = K_3 = Q_0Q_1Q_2 + Q_0Q_3$$



THE 74HC163 4-BIT SYNCHRONOUS BINARY COUNTER



Synchronous CLEAR and LOAD
Two active high enables ENT & ENP
RCO (ripple clock output) goes high in last state (during 15)
This signal that called terminal count (TC=15) in conjunction with enable inputs allows this counter to be cascaded for higher count sequence.

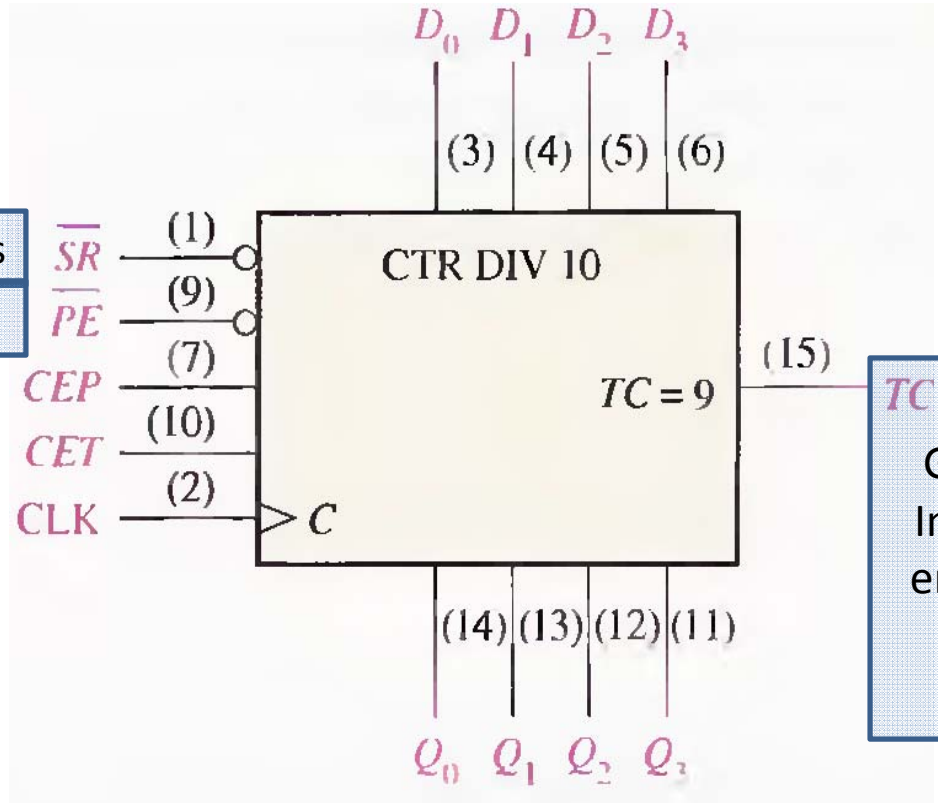

$\overline{CLR}$
$\overline{LOAD}$
Data inputs
D_0
D_1
D_2
D_3
CLK
ENP
ENT
Outputs
Q_0
Q_1
Q_2
Q_3
RCO
12
13
14
15
0
1
2
Count
Inhibit
Clear
Preset

THE 74F162 SYNCHRONOUS BCD DECADE COUNTER

Active low asynchronous

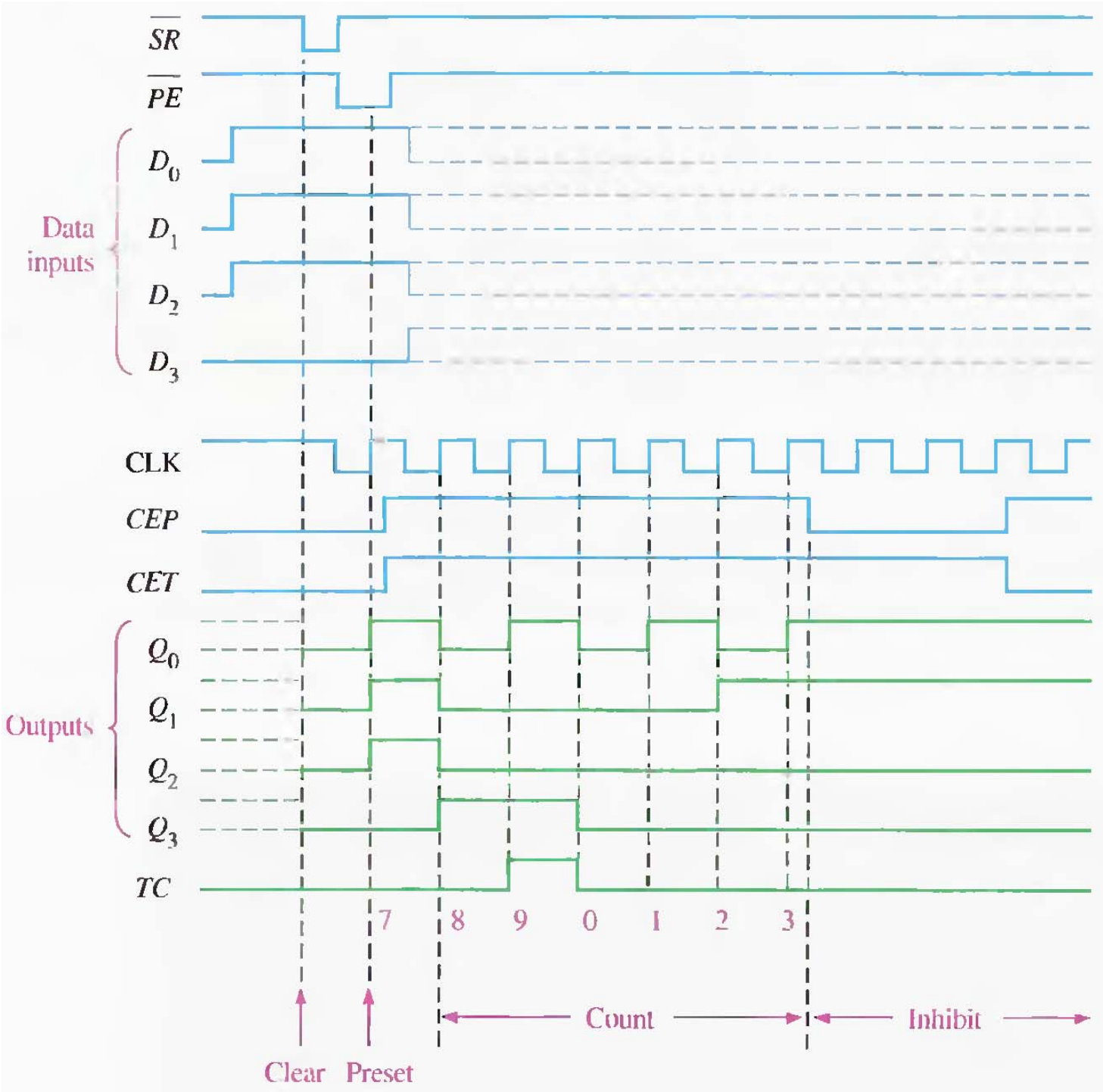
Active low synchronous

Active high



TC

Goes high In state 9
In counjunction with
enable inputs is used
for cascading


$\overline{SR}$
$\overline{PE}$
Data inputs
D_0
D_1
D_2
D_3
CLK
CEP
CET
Outputs
Q_0
Q_1
Q_2
Q_3
TC
7
8
9
0
1
2
3
Count
Inhibit
Clear
Preset

SECTION 8-2 REVIEW

1. How does a synchronous counter differ from an asynchronous counter?
2. Explain the function of the preset feature of counters such as the 74HC163.
3. Describe the purpose of the *ENP* and *ENT* inputs and the *RCO* output for the 74HC163 counter.

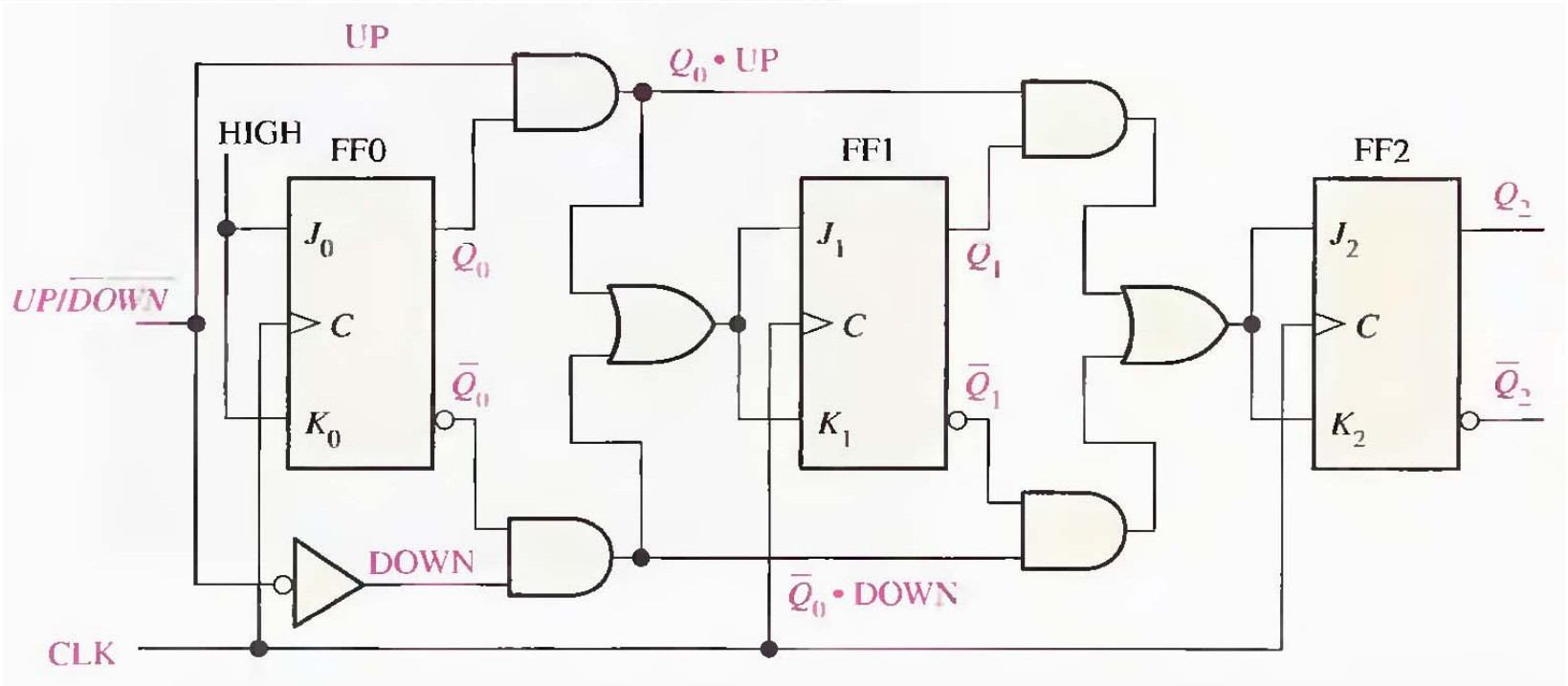
8–3 UP/DOWN SYNCHRONOUS COUNTERS

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	

$$J_0 = K_0 = 1$$

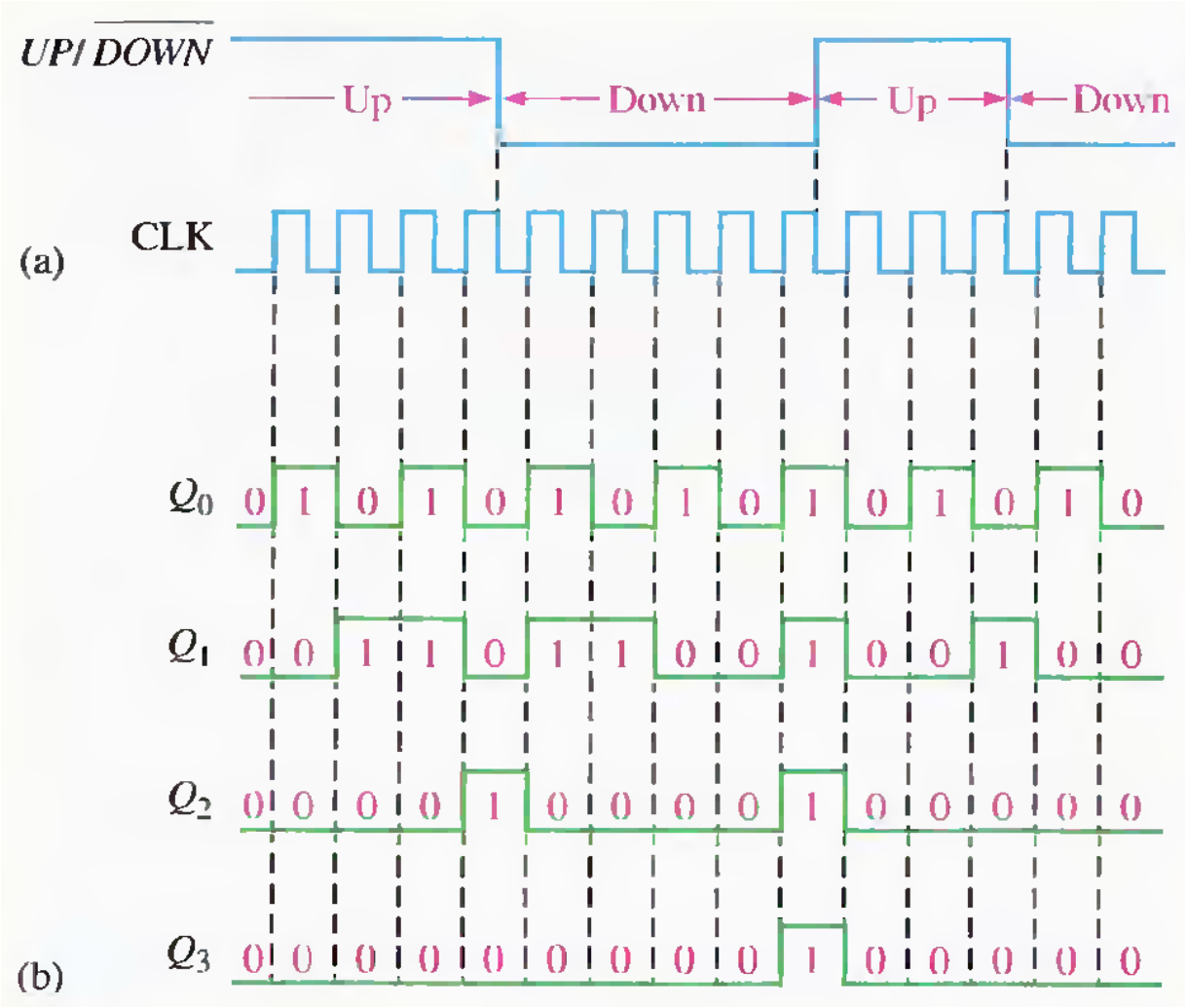
$$J_1 = K_1 = (Q_0 \cdot \text{UP}) + (\overline{Q}_0 \cdot \text{DOWN})$$

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



EXAMPLE 8–4

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 8–24(a). The counter starts in the all 0s state and is positive edge-triggered.



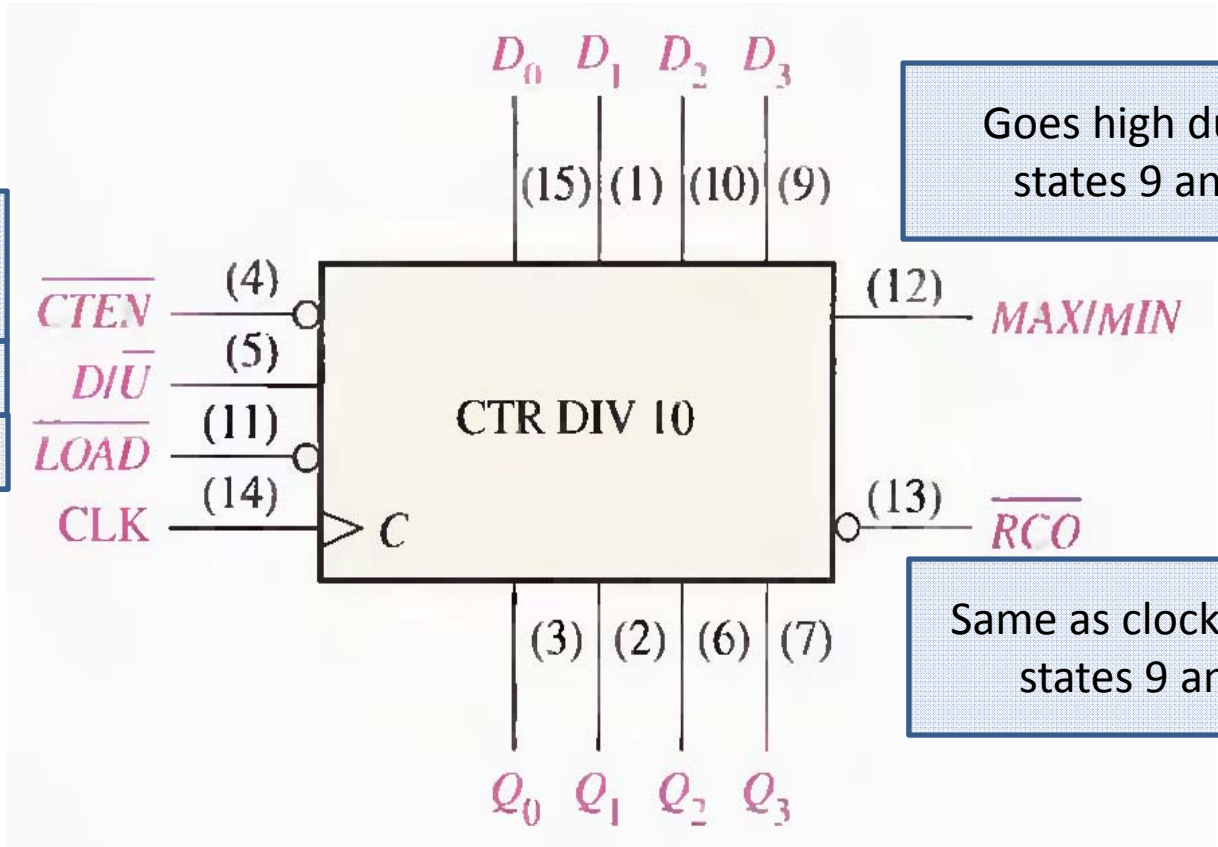
Q_3	Q_2	Q_1	Q_0	
0	0	0	0	UP
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	0	1	1	DOWN
0	0	1	0	
0	0	0	1	
0	0	0	0	
1	1	1	1	
0	0	0	0	UP
0	0	0	1	
0	0	1	0	
0	0	0	1	DOWN
0	0	0	0	

THE 74HC190 UP/DOWN DECADE COUNTER

Counter enable
Active low

Bi-static

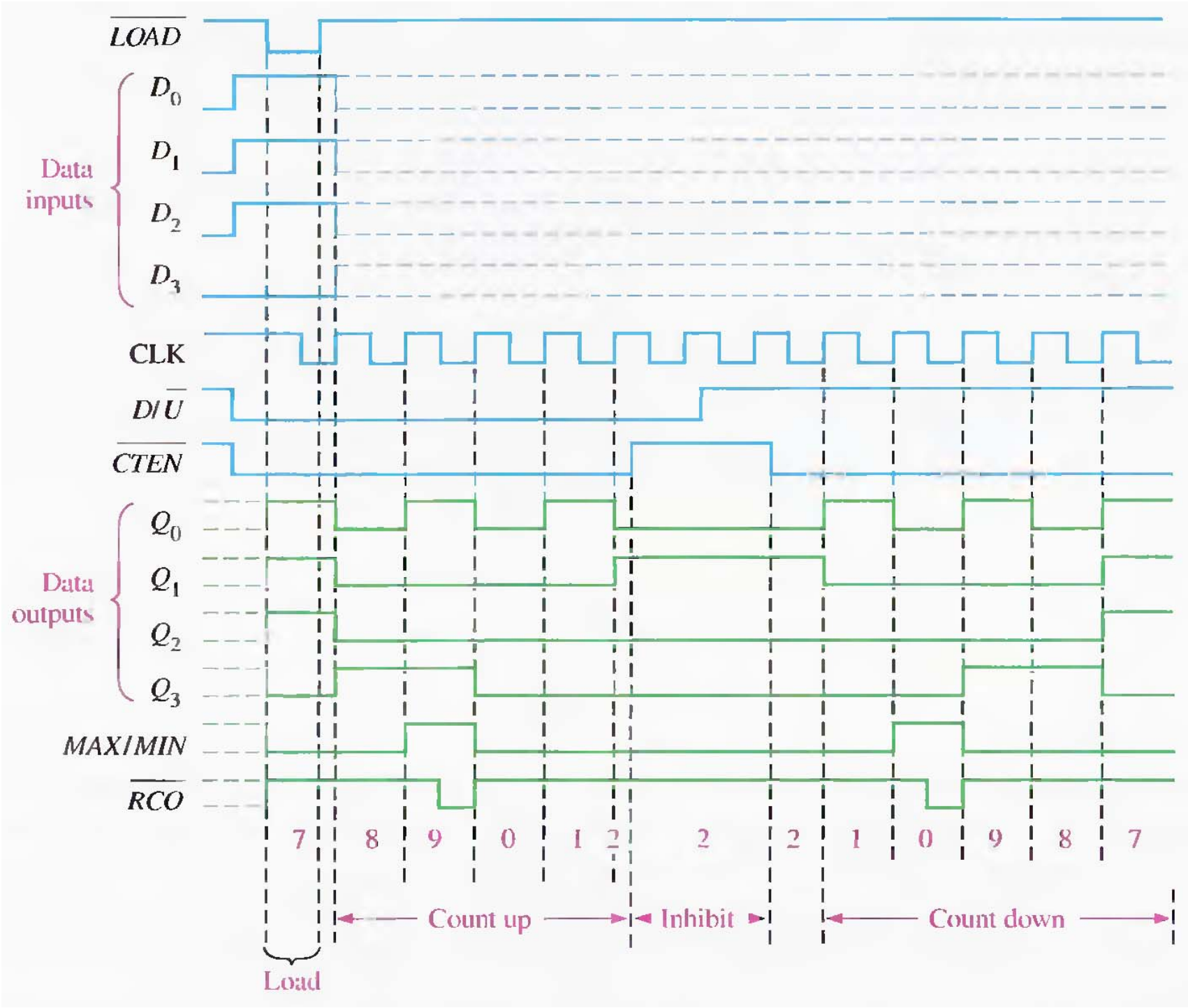
Active low asynchronous



Goes high during states 9 and 0

Same as clock during states 9 and 0

MAX/MIN, RCO and CTEN are used in cascading counter


LOAD
Data inputs
D0
D1
D2
D3
CLK
D/U
CTEN
Data outputs
Q0
Q1
Q2
Q3
MAX/MIN
RCO
7 8 9 0 1 2 2 2 1 0 9 8 7
Count up
Inhibit
Count down
Load

SECTION 8-3 REVIEW

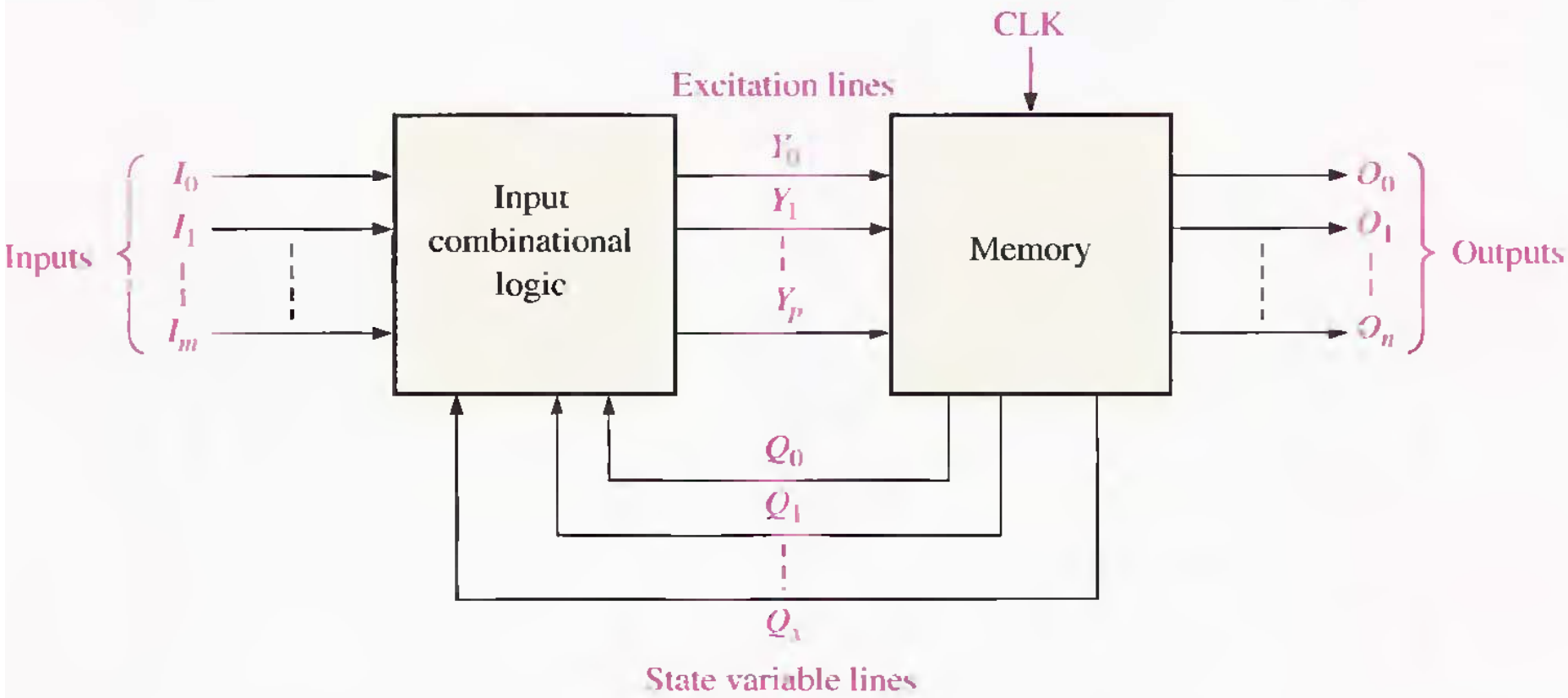
1. A 4-bit up/down binary counter is in the DOWN mode and in the 1010 state. On the next clock pulse, to what state does the counter go?
2. What is the terminal count of a 4-bit binary counter in the UP mode? In the DOWN mode? What is the next state after the terminal count in the DOWN mode?

8–4 DESIGN OF SYNCHRONOUS COUNTERS

In this section, you will see how sequential circuit design techniques can be applied specifically to counter design. In general, sequential circuits can be classified into two types: (1) those in which the output or outputs depend only on the present internal state (called *Moore circuits*) and (2) those in which the output or outputs depend on both the present state and the input or inputs (called *Mealy circuits*). This section is recommended for those who want an introduction to counter design or to state machine design in general. It is not a prerequisite for any other material.

General Model of a Sequential Circuit

Before proceeding with a specific counter design technique, let's begin with a general definition of a **sequential circuit** or **state machine:** A general sequential circuit consists of a combinational logic section and a memory section (flip-flops), as shown in Figure 8–27. In a clocked sequential circuit, there is a clock input to the memory section as indicated.



The information stored in the memory section, as well as the inputs to the combinational logic ($I_0, I_1, \ldots, I_m$), is required for proper operation of the circuit. At any given time, the memory is in a state called the *present state* and will advance to a *next state* on a clock pulse as determined by conditions on the excitation lines ($Y_0, Y_1, \ldots, Y_p$). The present state of the memory is represented by the state variables ($Q_0, Q_1, \ldots, Q_v$). These state variables, along with the inputs ($I_0, I_1, \ldots, I_m$), determine the system outputs ($O_0, O_1, \ldots, O_n$).

Not all sequential circuits have input and output variables as in the general model just discussed. However, all have excitation variables and state variables. Counters are a special case of clocked sequential circuits. In this section, a general design procedure for sequential circuits is applied to synchronous counters in a series of steps.

Step 1: State Diagram

Step 2: Next-State Table

Step 3: Flip-Flop Transition Table

Step 4: Karnaugh Maps

Step 5: Logic Expressions for Flip-Flop Inputs

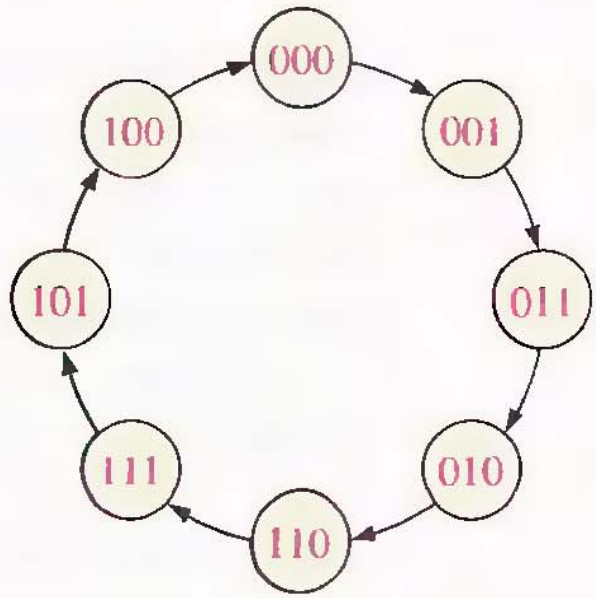
Step 6: Counter Implementation

Step 1: State Diagram

The first step in the design of a counter is to create a state diagram. A **state diagram** shows the progression of states through which the counter advances when it is clocked. As an example, Figure 8–28 is a state diagram for a basic 3-bit Gray code counter. This particular circuit has no inputs other than the clock and no outputs other than the outputs taken off each flip-flop in the counter. You may wish to review the coverage of the Gray code in Chapter 2 at this time.

FIGURE 8–28

State diagram for a 3-bit Gray code counter.



Step 2: Next-State Table

Once the sequential circuit is defined by a state diagram, the second step is to derive a next-state table, which lists each state of the counter (present state) along with the corresponding next state. *The next state is the state that the counter goes to from its present state upon application of a clock pulse.* The next-state table is derived from the state diagram and is shown in Table 8–7 for the 3-bit Gray code counter. Q_0 is the least significant bit.

TABLE 8–7

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

OUTPUT TRANSITIONS			FLIP-FLOP INPUTS	
Q_N		Q_{N+1}	J	K
0	$\longrightarrow$	0	0	X
0	$\longrightarrow$	1	1	X
1	$\longrightarrow$	0	X	1
1	$\longrightarrow$	1	X	0

Q_N: present state

Q_{N+1}: next state

X: "don't care"

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 8–7). For example, for the present state 000, Q_0 goes from a present state of 0 to a next state of 1. To make this happen, J_0 must be a 1 and you don't care what K_0 is ($J_0 = 1$, $K_0 = X$), as you can see in the transition table (Table 8–8). Next, Q_1 is 0 in the present state and remains a 0 in the next state. For this transition, $J_1 = 0$ and $K_1 = X$. Finally, Q_2 is 0 in the present state and remains a 0 in the next state. Therefore, $J_2 = 0$ and $K_2 = X$. This analysis is repeated for each present state in Table 8–7.

Step 4: Karnaugh Maps

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

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

J_0 map

Q_2Q_1 \ Q_0	0	1
00	1	
01		
11		
10		X

K_0 map

Q_2Q_1 \ Q_0	0	1
00	X	
01		
11		
10		1

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

Flip-flop transition table

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

Next-state table

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

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

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

Q_1Q_0

J_2 map

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

$\overline{Q}_2Q_0$

J_1 map

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

$\overline{Q}_2\overline{Q}_1$

Q_2Q_1

J_0 map

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

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

K_2 map

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

Q_2Q_0

K_1 map

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

$\overline{Q}_2Q_1$

$Q_2\overline{Q}_1$

K_0 map

Step 5: Logic Expressions for Flip-Flop Inputs

From the Karnaugh maps of Figure 8–30 you obtain the following expressions for the J and K inputs of each flip-flop:

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

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

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

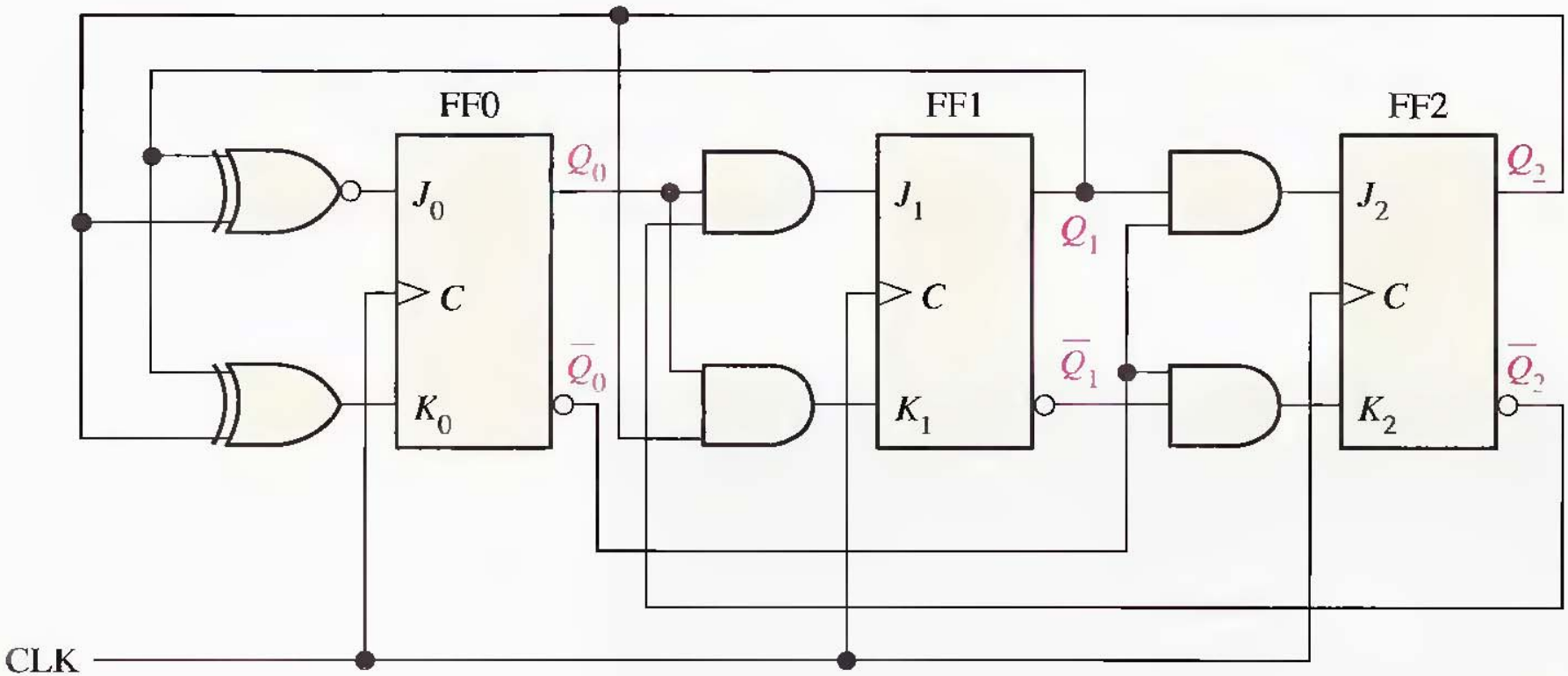
$$K_1 = Q_2Q_0$$

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

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

Step 6: Counter Implementation

The final step is to implement the combinational logic from the expressions for the J and K inputs and connect the flip-flops to form the complete 3-bit Gray code counter as shown in Figure 8–31.



A summary of steps used in the design of this counter follows. In general, these steps can be applied to any sequential circuit.

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 8–5

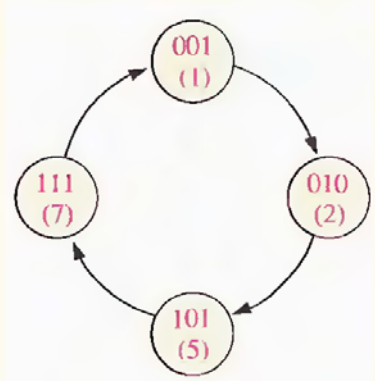
Design a counter with the irregular binary count sequence shown in the state diagram of Figure 8–32. Use J-K flip-flops.

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.

2 FF

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

PRESENT STATE			NEXT STATE		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	1	0	1	0
0	1	0	1	0	1
1	0	1	1	1	1
1	1	1	0	0	1



Step 3: The transition table for the J-K flip-flop is repeated in Table 8–10.

OUTPUT TRANSITIONS			FLIP-FLOP INPUTS	
Q_N		Q_{N+1}	J	K
0	→	0	0	X
0	→	1	1	X
1	→	0	X	1
1	→	1	X	0

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

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

Q_1

J_2 map

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

1

J_1 map

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

1

J_0 map

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

Q_1

K_2 map

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

1

K_1 map

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

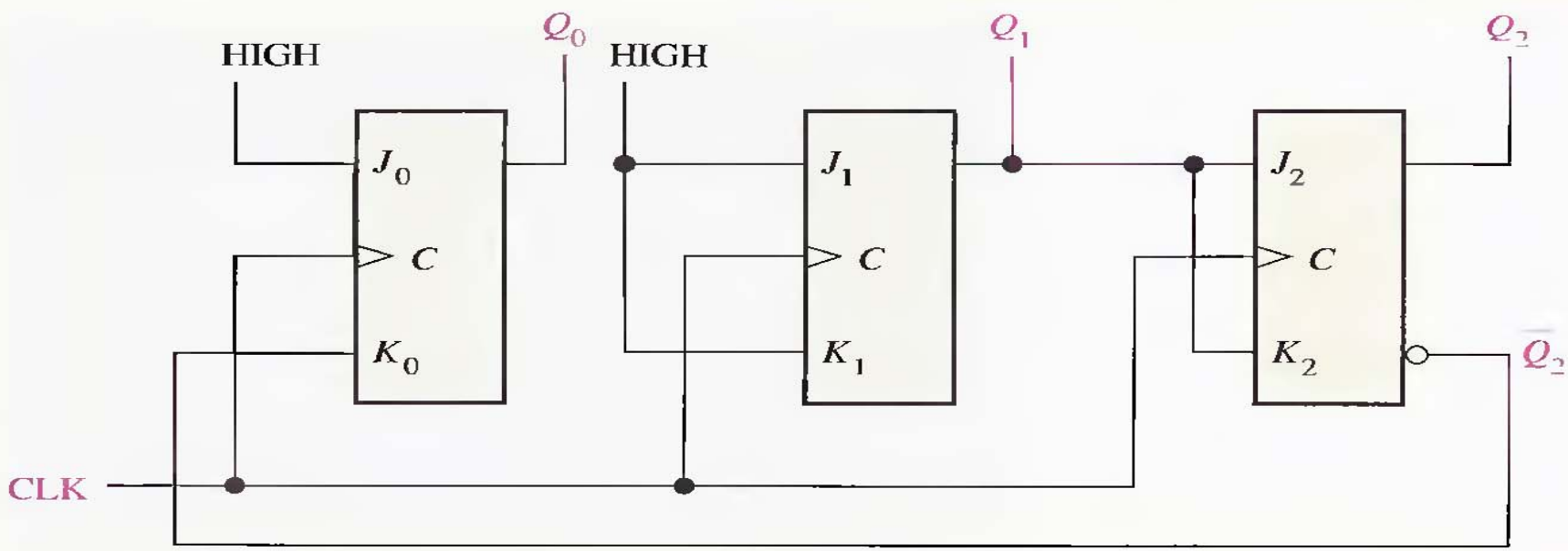
Q_2

K_0 map

Step 5: Group the 1s, taking advantage of as many of the "don't care" states as possible for maximum simplification, as shown in Figure 8–33. Notice that when *all* cells in a map are grouped, the expression is simply equal to 1. The expression for each J and K input taken from the maps is as follows:

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

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



An analysis shows that if the counter, by accident, gets into one of the invalid states (0, 3, 4, 6), it will always return to a valid state according to the following sequences: $0 \rightarrow 3 \rightarrow 4 \rightarrow 7$, and $6 \rightarrow 1$.

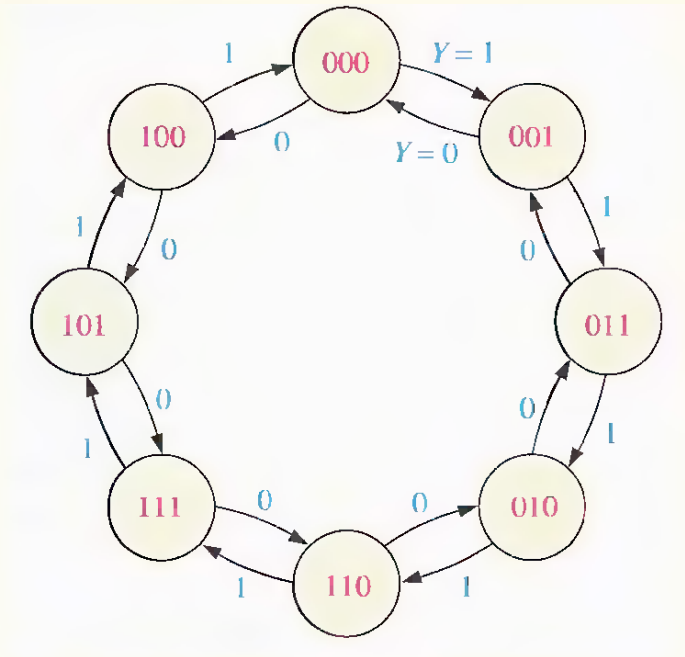
EXAMPLE 8–6

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

Step 1: The state diagram is shown in Figure 8–35. The 1 or 0 beside each arrow indicates the state of the UP/$\overline{\text{DOWN}}$ control input, Y.

PRESENT STATE			NEXT STATE $Y = 0$ (DOWN)			NEXT STATE $Y = 1$ (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.

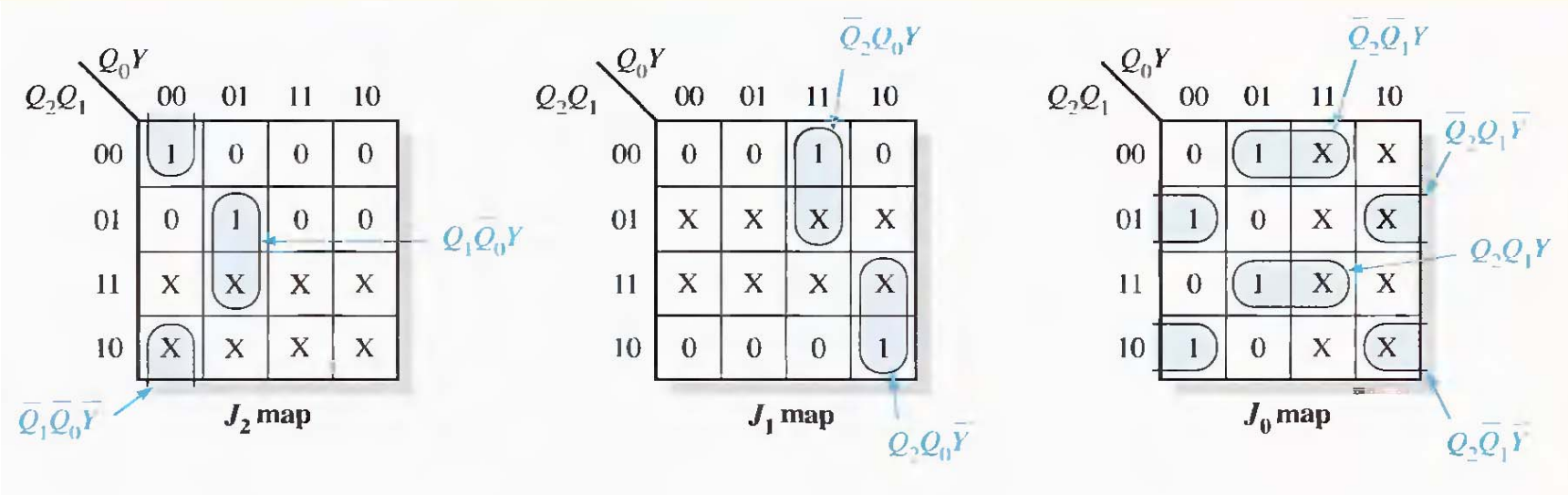


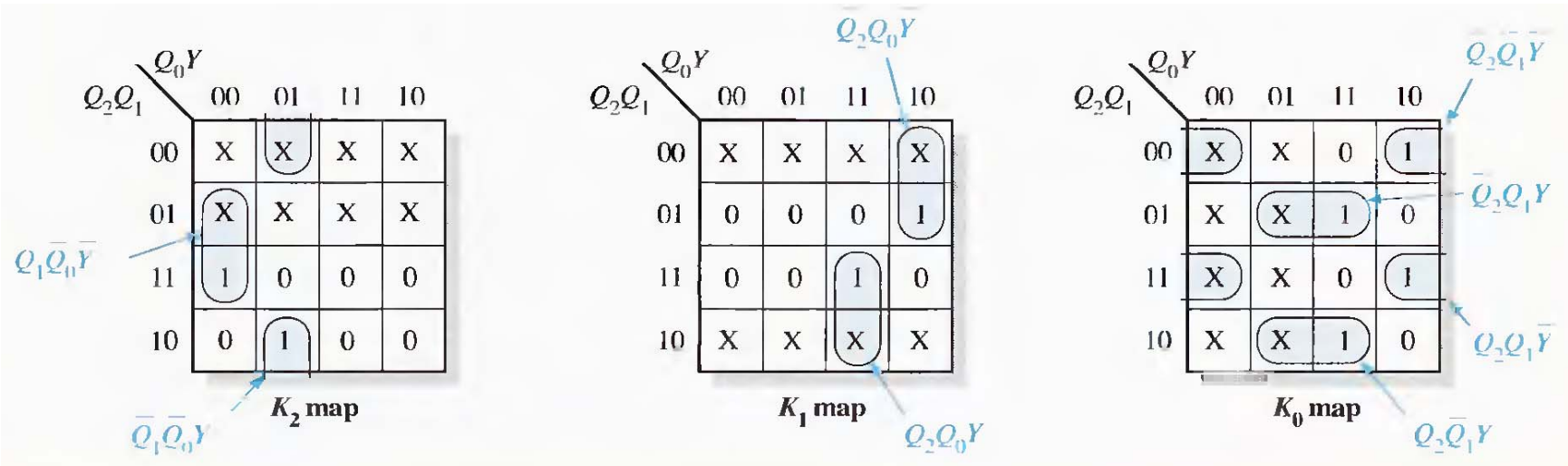
Step 2: The next-state table is derived from the state diagram and is shown in Table 8–11. Notice that for each present state there are two possible next states, depending on the UP/$\overline{\text{DOWN}}$ control variable, Y.

Step 3: The transition table for the J-K flip-flops is repeated in Table 8–12.

OUTPUT TRANSITIONS			FLIP-FLOP INPUTS	
Q_N		Q_{N+1}	J	K
0	$\longrightarrow$	0	0	X
0	$\longrightarrow$	1	1	X
1	$\longrightarrow$	0	X	1
1	$\longrightarrow$	1	X	0

Step 4: The Karnaugh maps for the J and K inputs of the flip-flops are shown in Figure 8–36. 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 8–12 is transferred onto the maps as indicated for each present state of the counter.



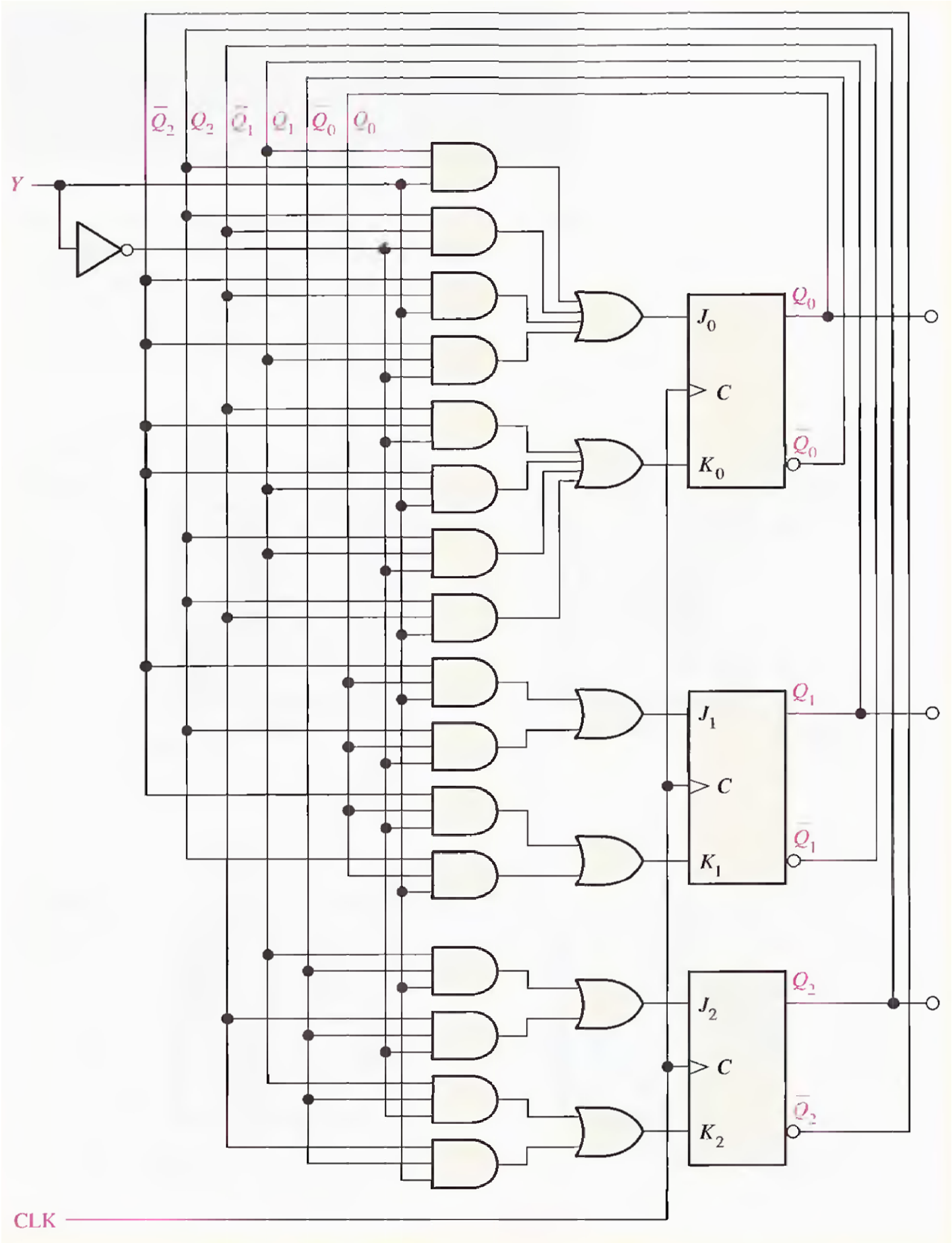


Step 5: The 1s are combined in the largest possible groupings, with "don't cares" (Xs) used where possible. The groups are factored, and the expressions for the J and K inputs are as follows:

$$J_0 = Q_2Q_1Y + Q_2\overline{Q}_1\overline{Y} + \overline{Q}_2\overline{Q}_1Y + \overline{Q}_2Q_1\overline{Y}$$
$$J_1 = \overline{Q}_2Q_0Y + Q_2Q_0\overline{Y}$$
$$J_2 = Q_1\overline{Q}_0Y + \overline{Q}_1\overline{Q}_0\overline{Y}$$

$$K_0 = \overline{Q}_2\overline{Q}_1\overline{Y} + \overline{Q}_2Q_1Y + Q_2\overline{Q}_1Y + Q_2Q_1\overline{Y}$$
$$K_1 = \overline{Q}_2Q_0\overline{Y} + Q_2Q_0Y$$
$$K_2 = Q_1\overline{Q}_0\overline{Y} + \overline{Q}_1\overline{Q}_0Y$$

Step 6: The J and K equations are implemented with combinational logic, and the complete counter is shown in Figure 8–37.


$\overline{Q}_2$
Q_2
$\overline{Q}_1$
Q_1
$\overline{Q}_0$
Q_0
Y
J_0
C
K_0
Q_0
$\overline{Q}_0$
J_1
C
K_1
Q_1
$\overline{Q}_1$
J_2
C
K_2
Q_2
$\overline{Q}_2$
CLK

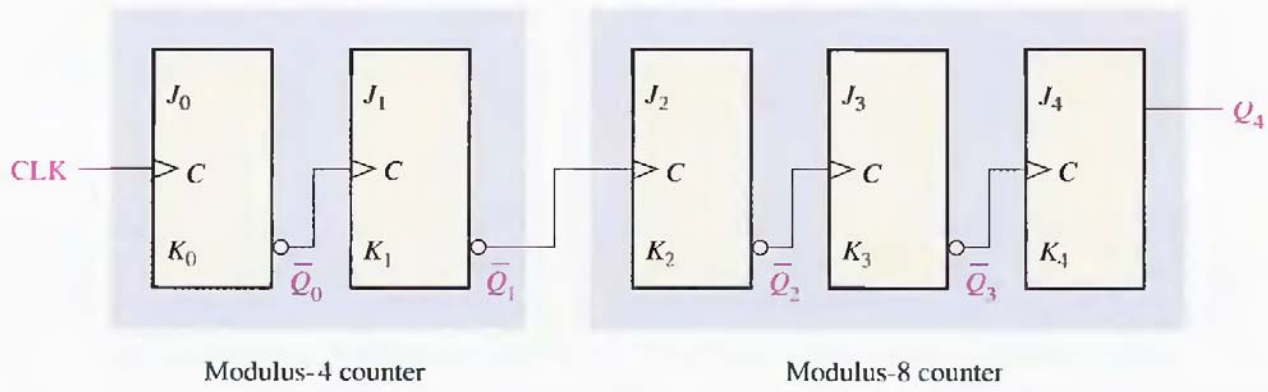
SECTION 8–4 REVIEW

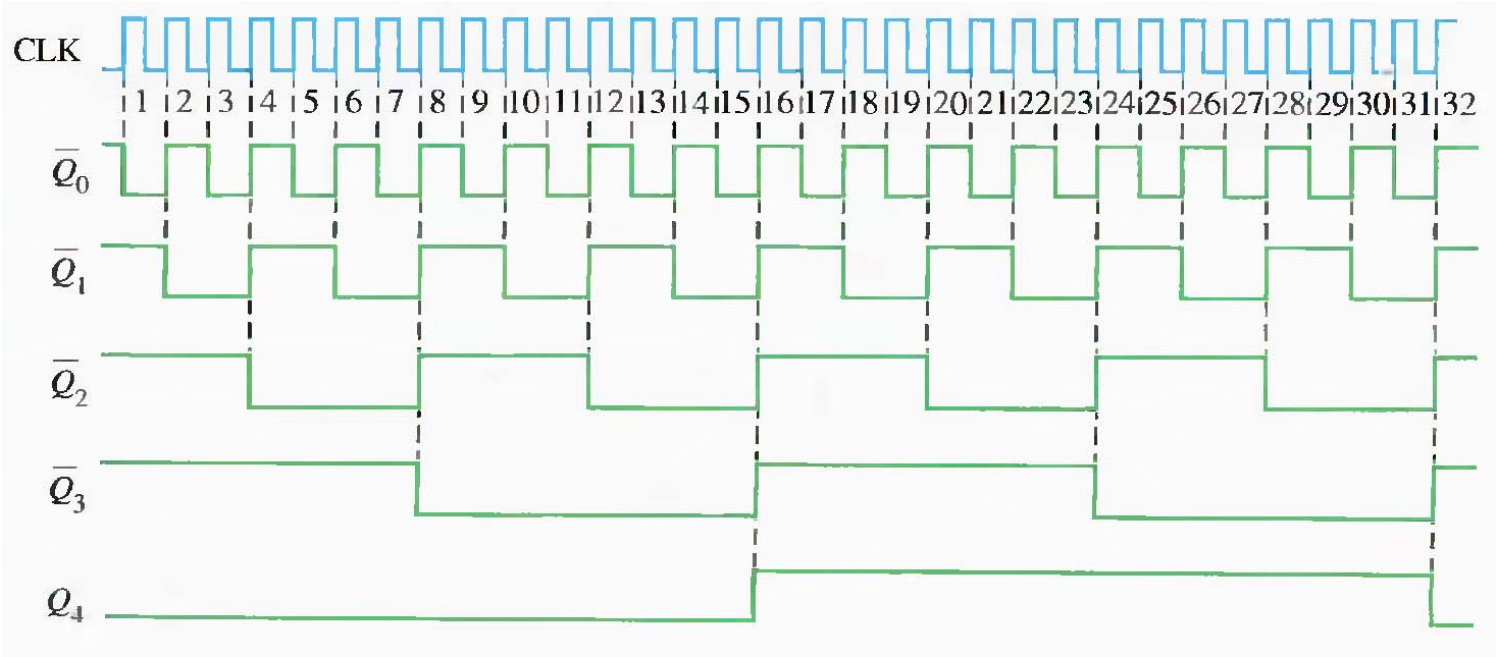
1. A flip-flop is presently in the RESET state and must go to the SET state on the next clock pulse. What must J and K be?
2. A flip-flop is presently in the SET state and must remain SET on the next clock pulse. What must J and K be?
3. A binary counter is in the $Q_3\overline{Q}_2Q_1\overline{Q}_0 = 1010$ state.
   (a) What is its next state?
   (b) What condition must exist on each flip-flop input to ensure that it goes to the proper next state on the clock pulse?

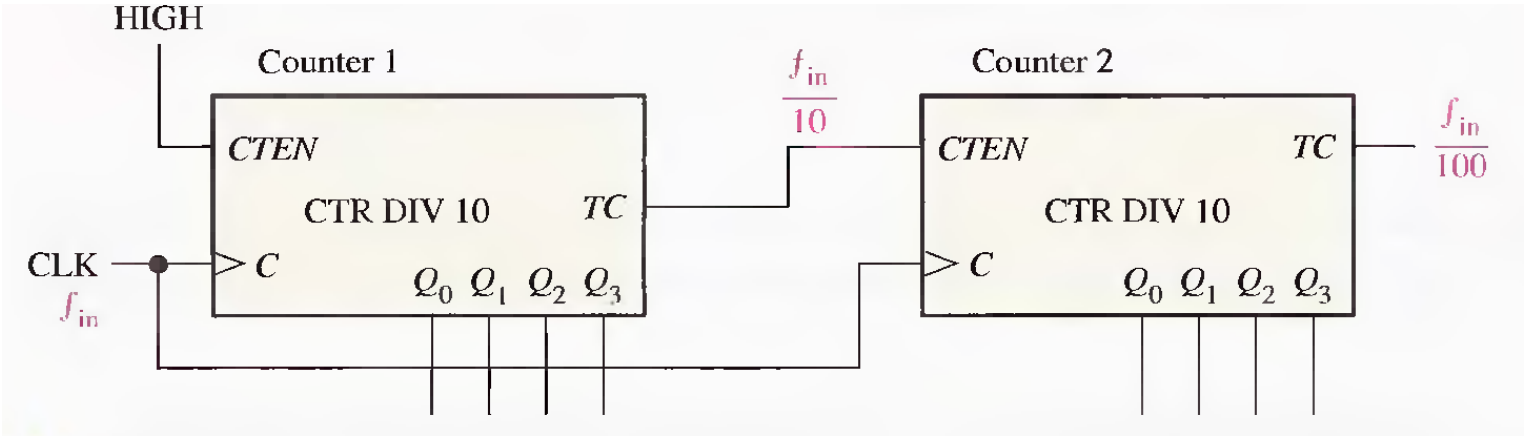
8–5 CASCADED COUNTERS

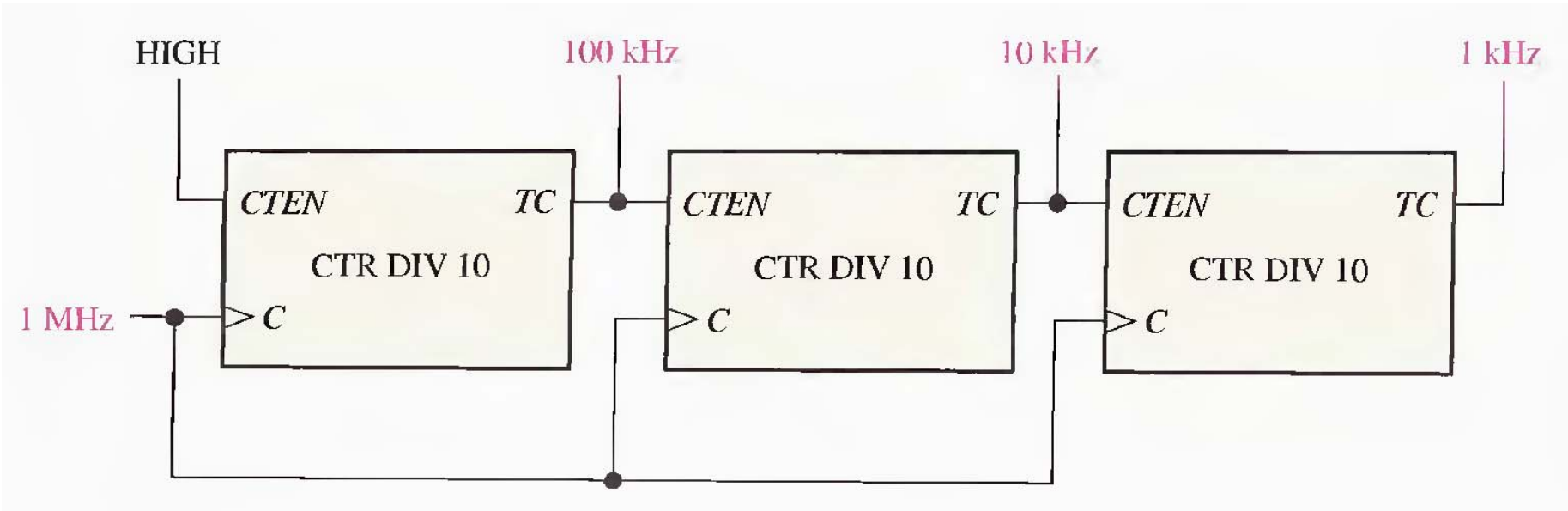
Counters can be connected in cascade to achieve higher-modulus operation. In essence, **cascading** means that the last-stage output of one counter drives the input of the next counter.

The overall modulus of cascaded counters is equal to the product of the individual moduli.





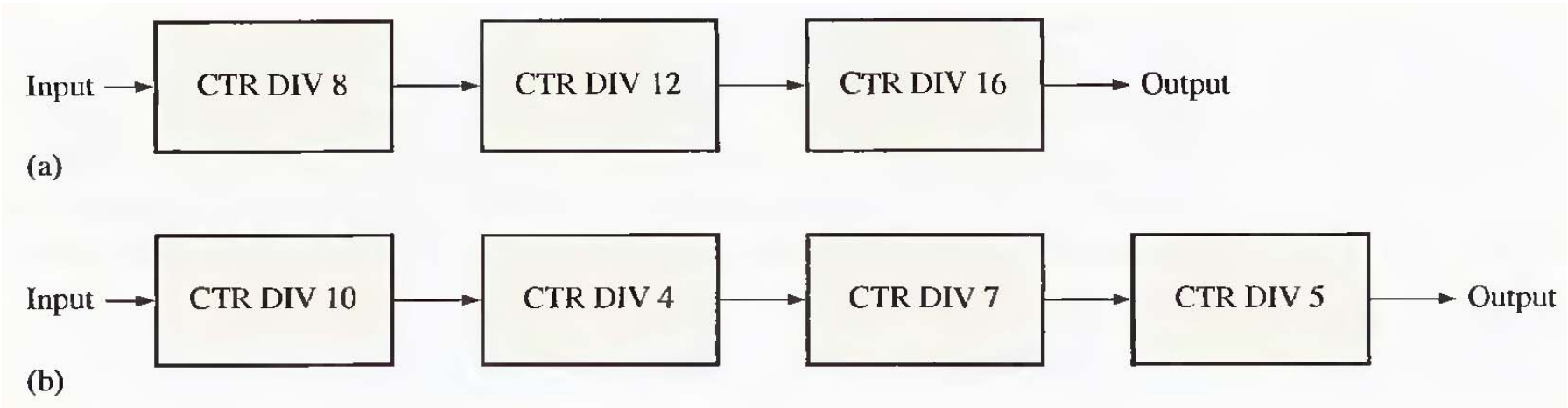

HIGH
Counter 1
CTEN
CTR DIV 10
TC
CLK
f_{in}
C
Q_0 Q_1 Q_2 Q_3
$\frac{f_{in}}{10}$
Counter 2
CTEN
TC
CTR DIV 10
C
Q_0 Q_1 Q_2 Q_3
$\frac{f_{in}}{100}$


HIGH
100 kHz
10 kHz
1 kHz
CTEN
TC
CTR DIV 10
C
1 MHz

COMPUTER NOTE

The time stamp counter (TSC), mentioned in the last computer note, is a 64-bit counter. It is interesting to observe that if this counter (or any full-modulus 64-bit counter) is clocked at a frequency of 100 MHz, it will take 5,849 years for it to go through all of its states and reach its terminal count. In contrast, a 32-bit full-modulus counter will exhaust all of its states in approximately 43 seconds when clocked at 100 MHz. The difference is astounding.

EXAMPLE 8–7 Determine the overall modulus of the two cascaded counter configurations in Figure 8–42.



Solution In Figure 8–42(a), the overall modulus for the 3-counter configuration is

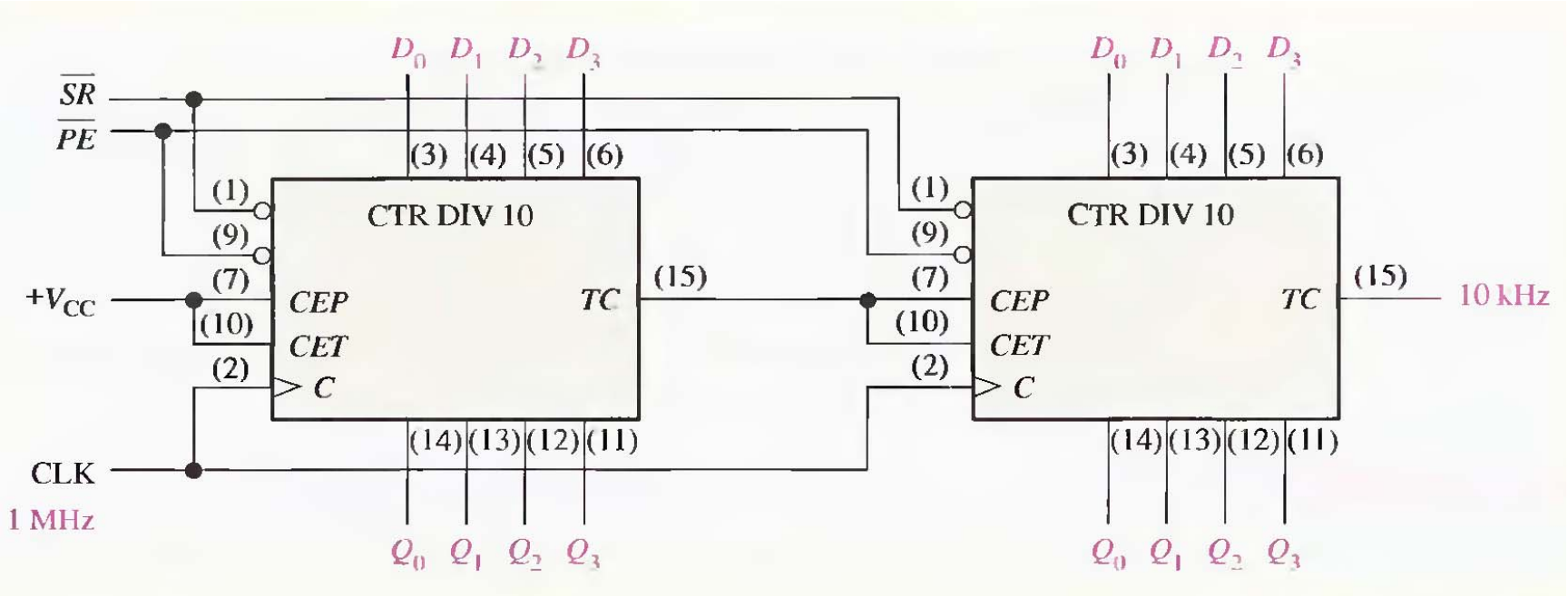
$$8 \times 12 \times 16 = \mathbf{1536}$$

In Figure 8–42(b), the overall modulus for the 4-counter configuration is

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

EXAMPLE 8–8

Use 74F162 decade counters to obtain a 10 kHz waveform from a 1 MHz clock. Show the logic diagram.



Cascaded Counters with Truncated Sequences

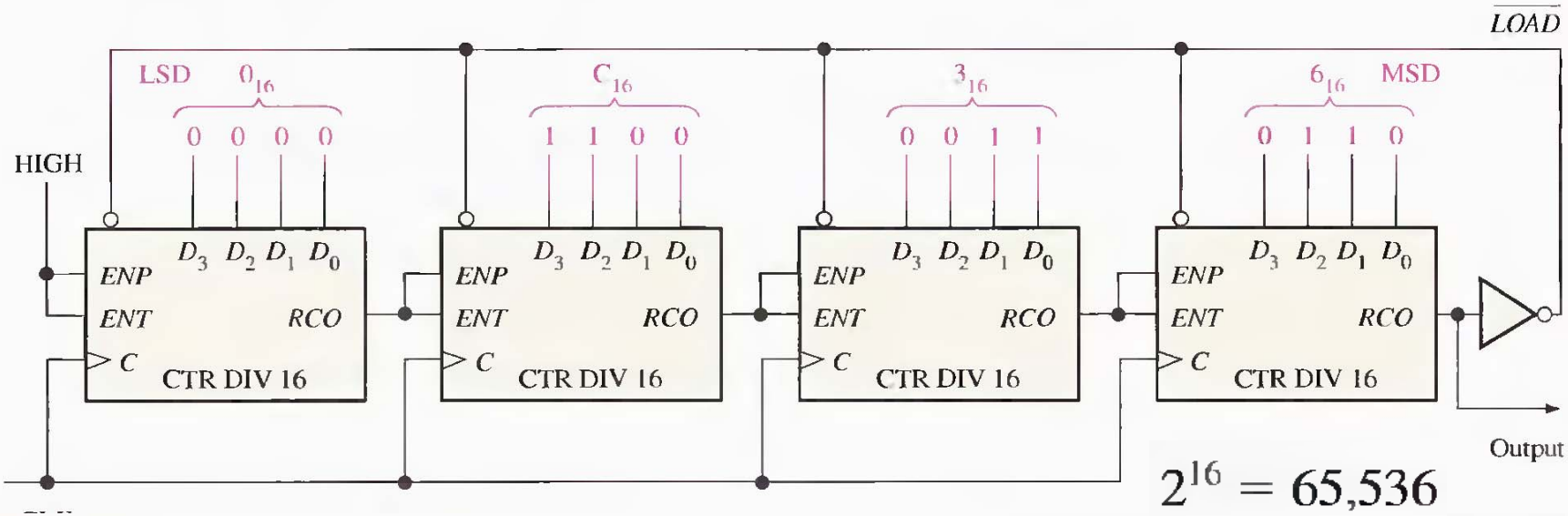


FIGURE 8–44

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).

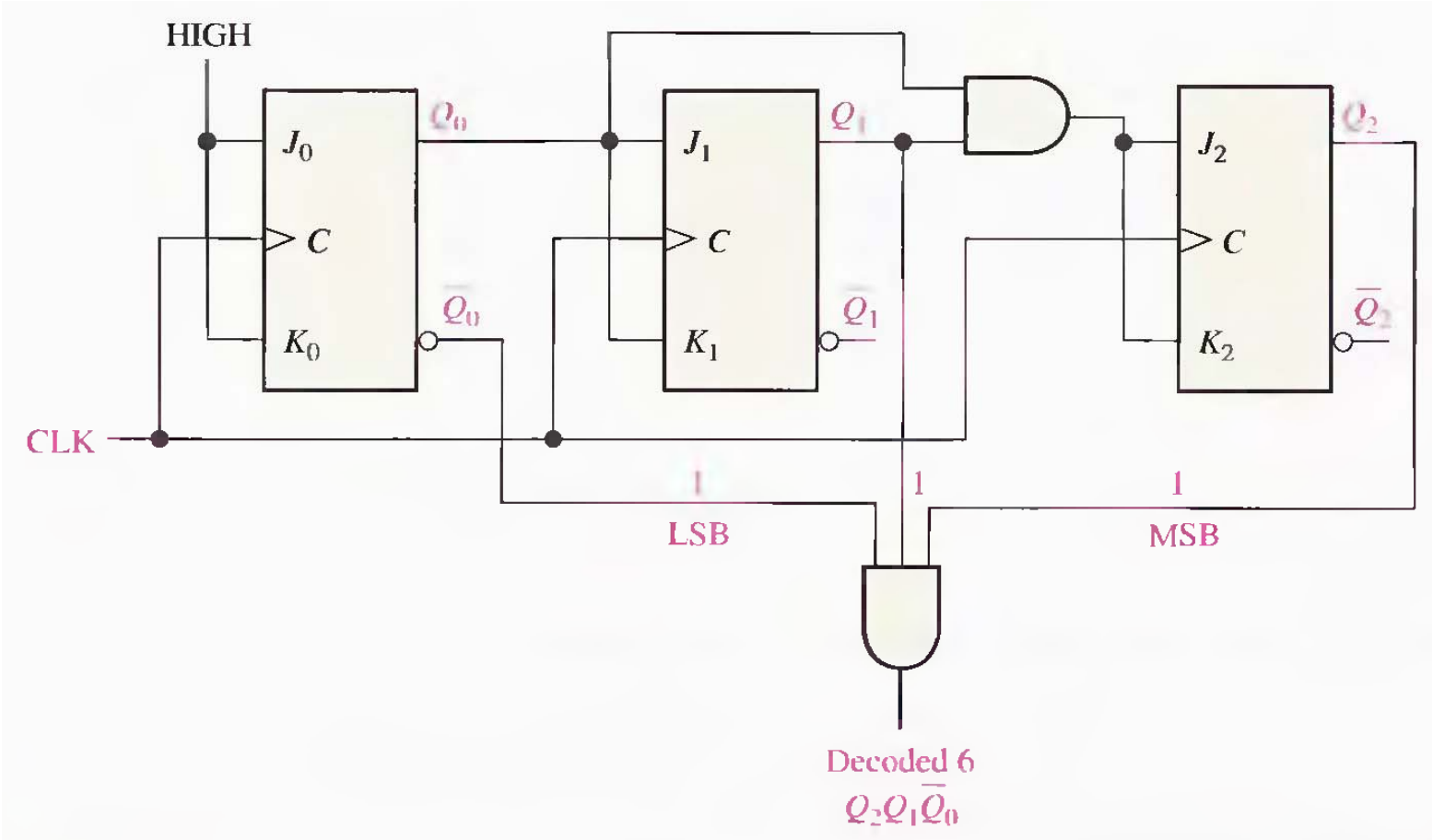
Let's assume that a certain application requires a divide-by-40,000 counter (modulus 40,000). The difference between 65,536 and 40,000 is 25,536, which is the number of states that must be *deleted* from the full-modulus sequence. The technique used in the circuit of Figure 8–44 is to preset the cascaded 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.

SECTION 8–5 REVIEW

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

8–6 COUNTER DECODING

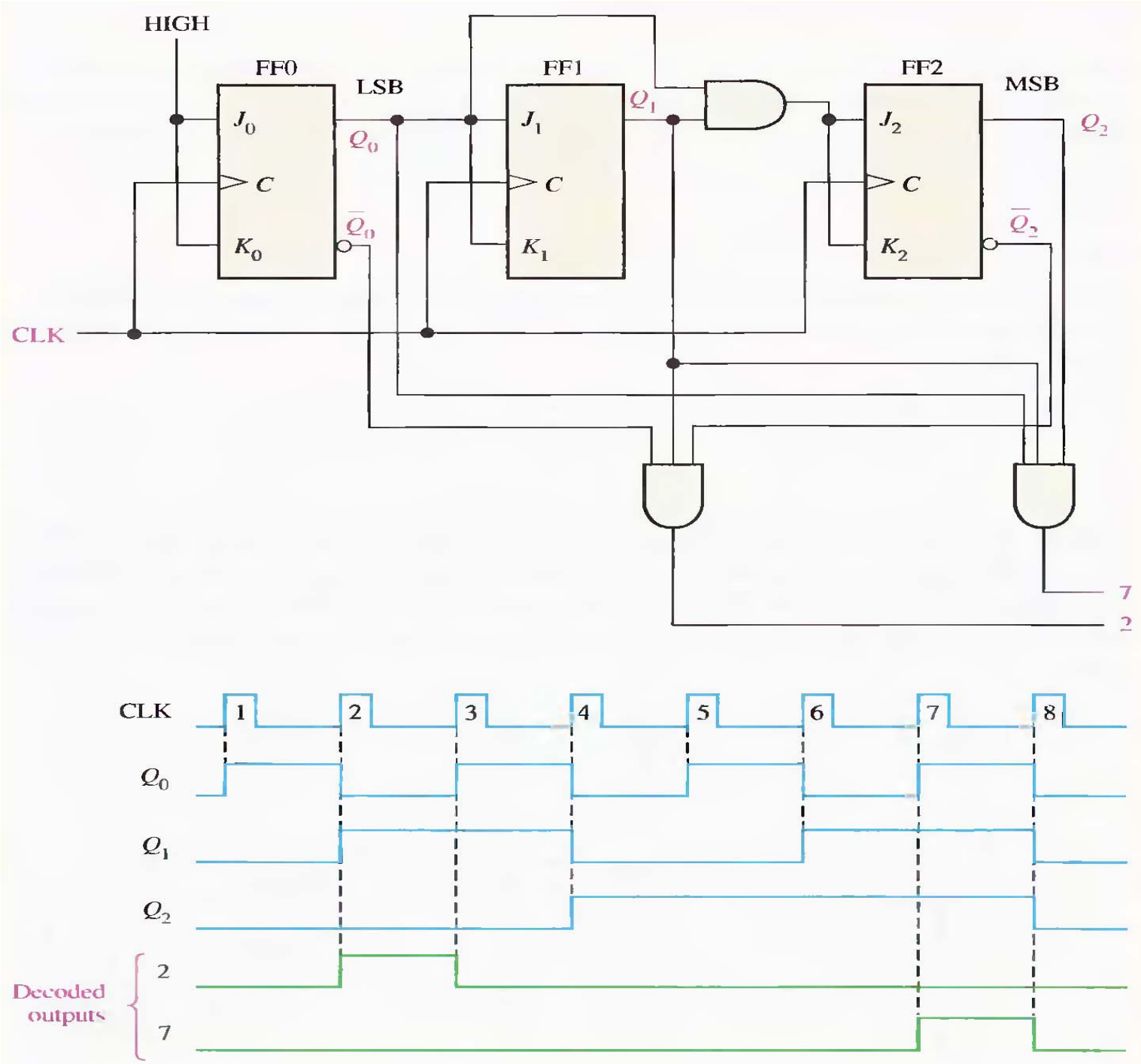
In many applications, it is necessary that some or all of the counter states be decoded. The decoding of a counter involves using decoders or logic gates to determine when the counter is in a certain binary state in its sequence. For instance, the terminal count function previously discussed is a single decoded state (the last state) in the counter sequence.



Suppose that you wish to decode binary state 6 (110) of a 3-bit binary counter. When $Q_2 = 1$, $Q_1 = 1$, and $Q_0 = 0$, a HIGH appears on the output of the decoding gate, indicating that the counter is at state 6. This can be done as shown in Figure 8–45. This is called *active-HIGH decoding.* Replacing the AND gate with a NAND gate provides active-LOW decoding.

EXAMPLE 8–9

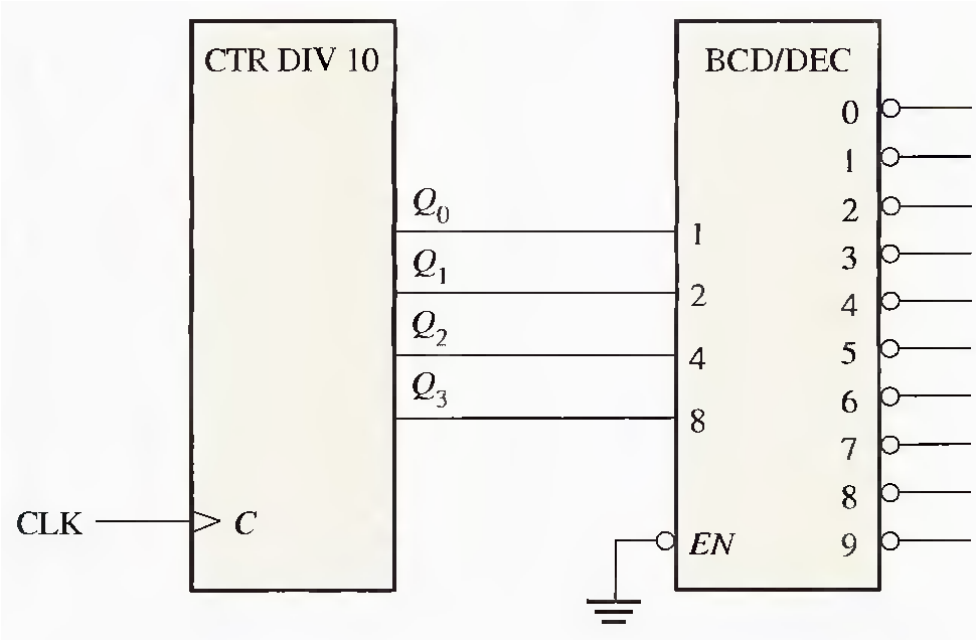
Implement the decoding of binary state 2 and binary state 7 of a 3-bit synchronous counter. Show the entire counter timing diagram and the output waveforms of the decoding gates. Binary 2 $= \overline{Q}_2 Q_1 \overline{Q}_0$ and binary 7 $= Q_2 Q_1 Q_0$.

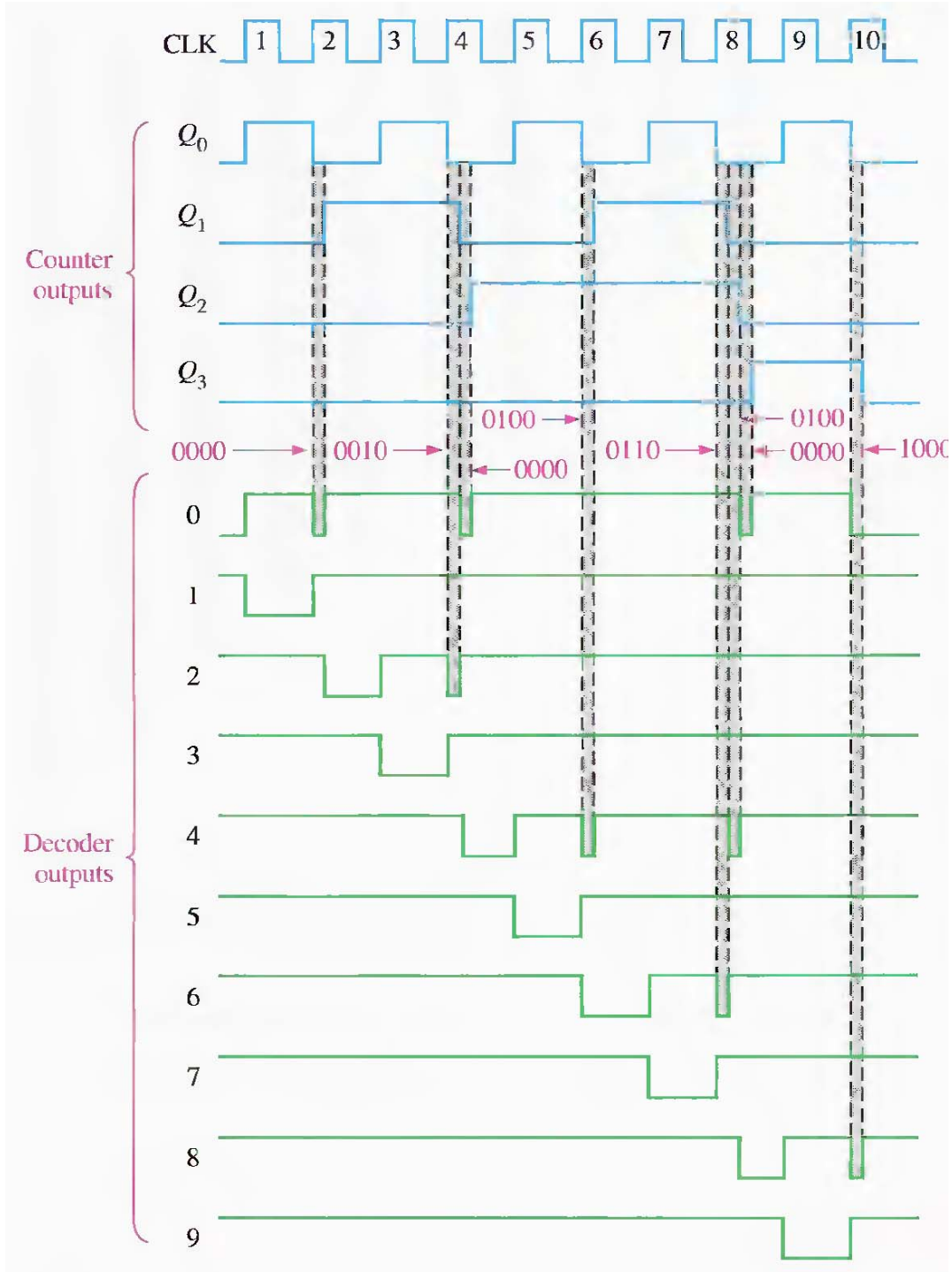

HIGH
FF0
LSB
FF1
FF2
MSB
J_0
C
K_0
Q_0
$\overline{Q}_0$
J_1
C
K_1
Q_1
J_2
C
K_2
Q_2
$\overline{Q}_2$
CLK
7
2
CLK
1
2
3
4
5
6
7
8
Q_0
Q_1
Q_2
Decoded outputs
2
7

Decoding Glitches

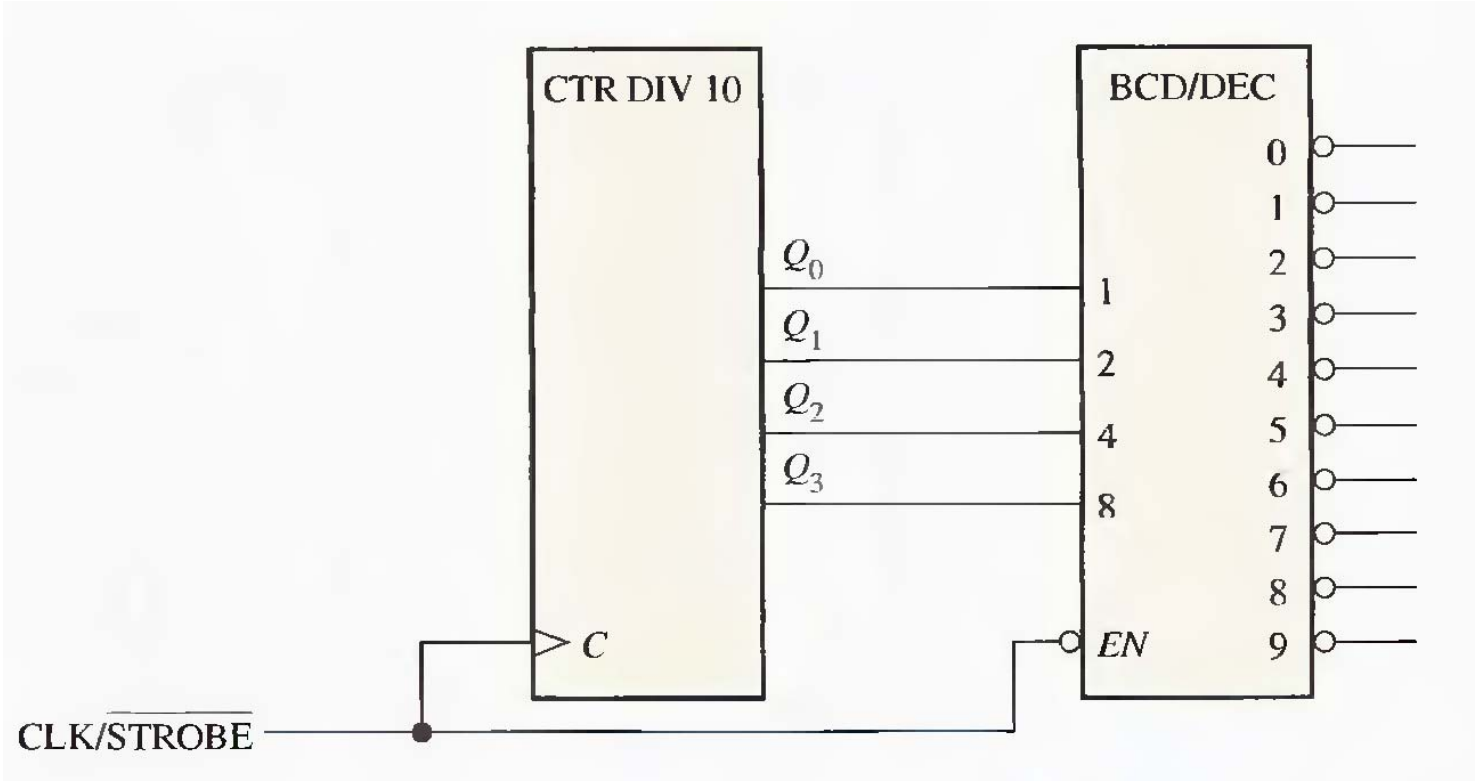
A glitch is an unwanted spike of voltage.

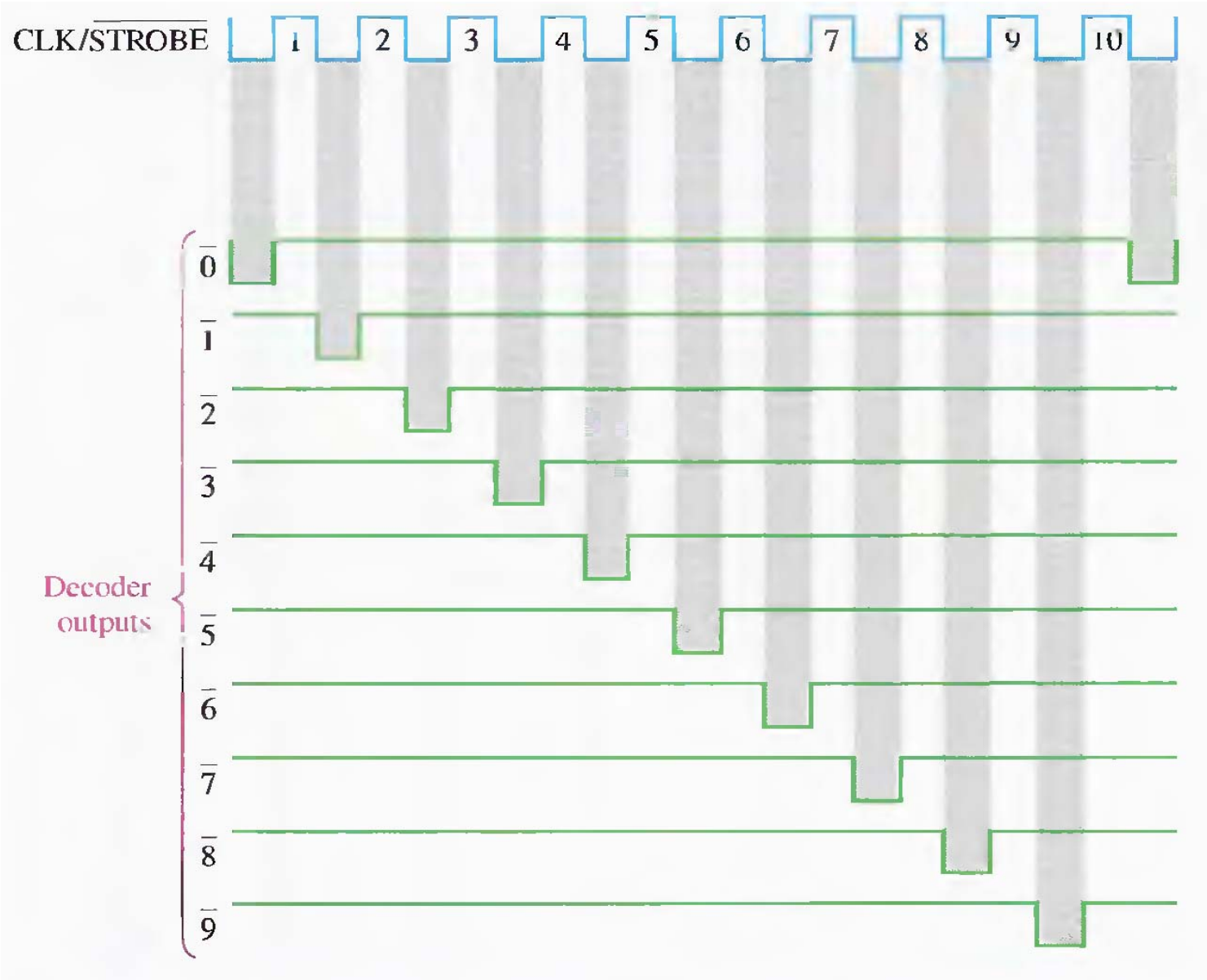
The problem of glitches produced by the decoding process was discussed in Chapter 6. As you have learned, the propagation delays due to the ripple effect in asynchronous counters create transitional states in which the counter outputs are changing at slightly different times. These transitional states produce undesired voltage spikes of short duration (glitches) on the outputs of a decoder connected to the counter. The glitch problem can also occur to some degree with synchronous counters because the propagation delays from the clock to the Q outputs of each flip-flop in a counter can vary slightly.




CLK
1
2
3
4
5
6
7
8
9
10
Counter outputs
Q_0
Q_1
Q_2
Q_3
0000
0010
0100
0000
0110
0100
0000
1000
Decoder outputs
0
1
2
3
4
5
6
7
8
9

One way to eliminate the glitches is to enable the decoded outputs at a time after the glitches have had time to disappear. This method is known as *strobing* and can be accomplished in the case of an active-HIGH clock by using the LOW level of the clock to enable the decoder, as shown in Figure 8–49. The resulting improved timing diagram is shown in Figure 8–50.




CLK/STROBE
1
2
3
4
5
6
7
8
9
10
$\overline{0}$
$\overline{1}$
$\overline{2}$
$\overline{3}$
$\overline{4}$
$\overline{5}$
$\overline{6}$
$\overline{7}$
$\overline{8}$
$\overline{9}$
Decoder outputs

SECTION 8–6 REVIEW

1. What transitional states are possible when a 4-bit asynchronous binary counter changes from

   (a) count 2 to count 3

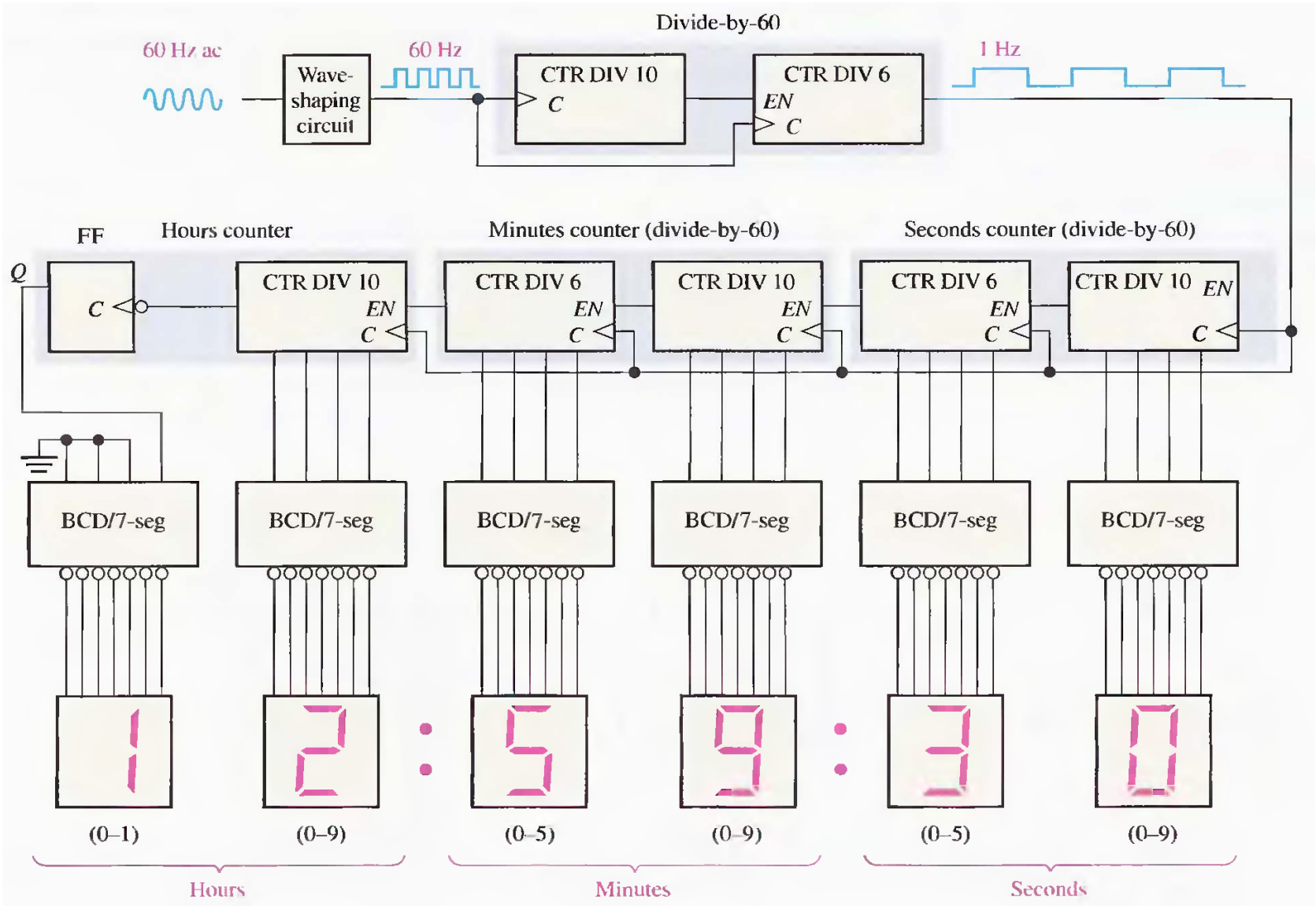
   (b) count 3 to count 4

   (c) count 10_{10} to count 11_{10}

   (d) count 15 to count 0

8–7 COUNTER APPLICATIONS

A Digital Clock



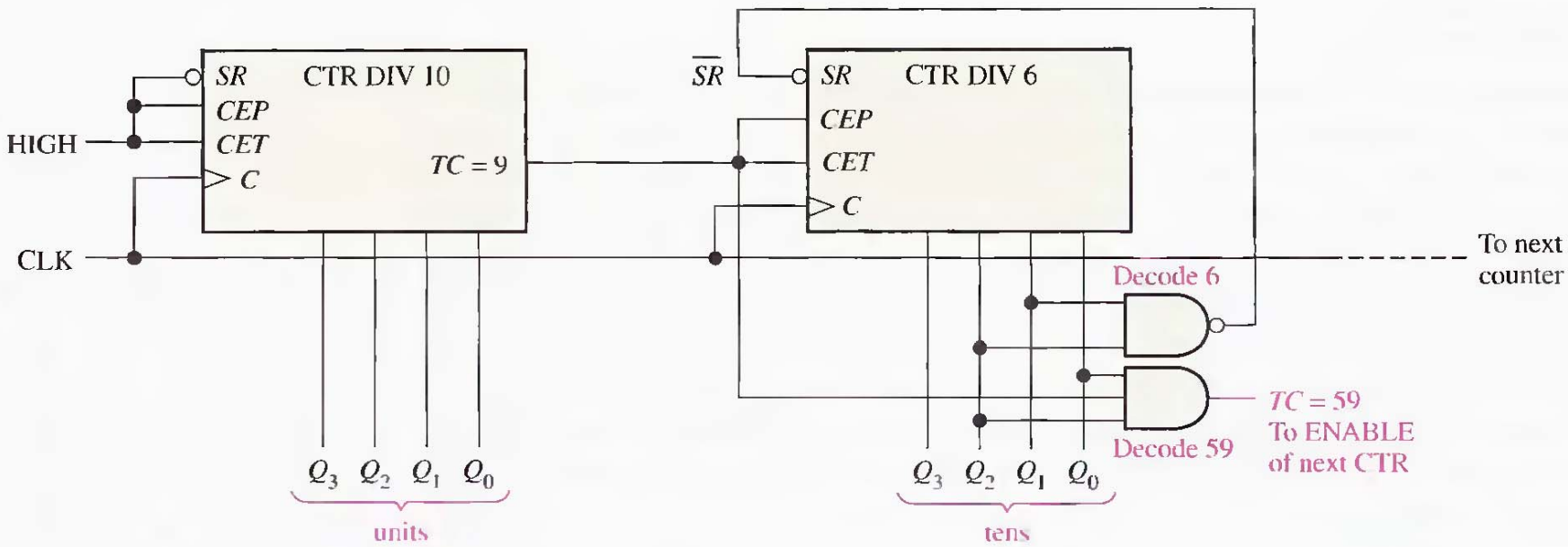
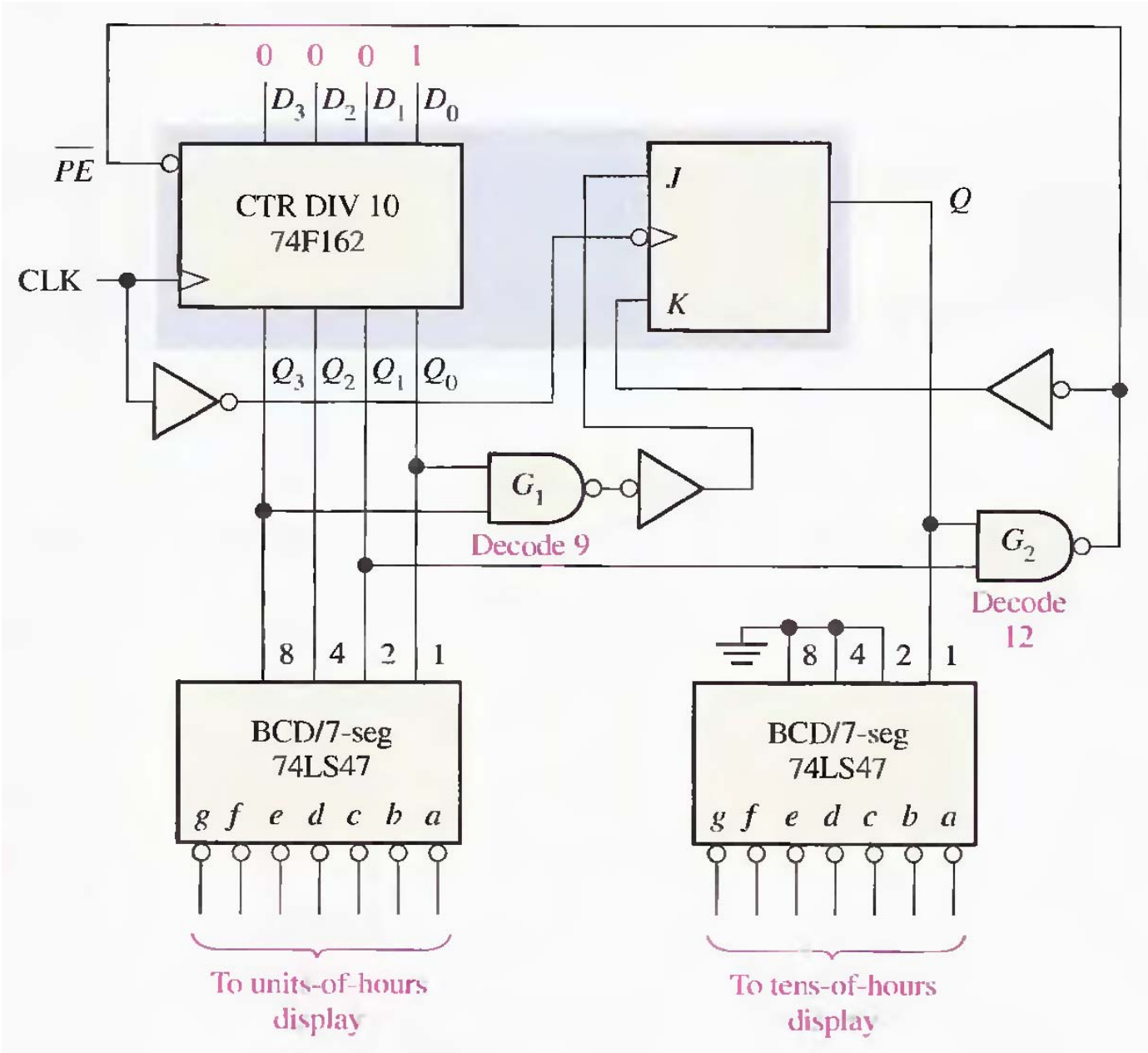
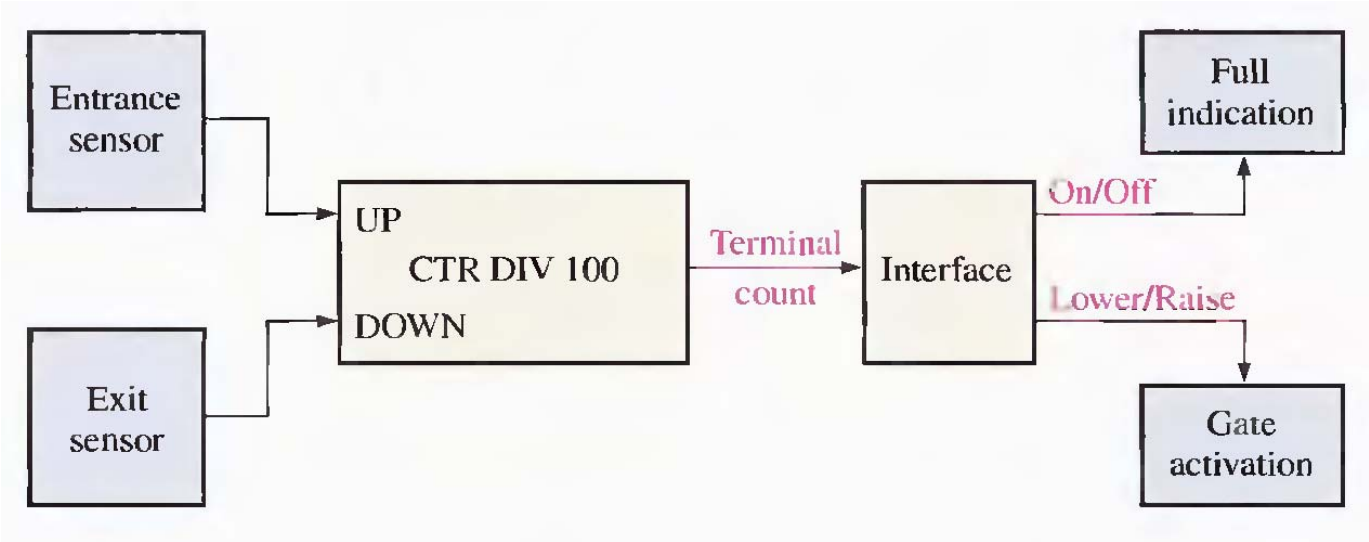


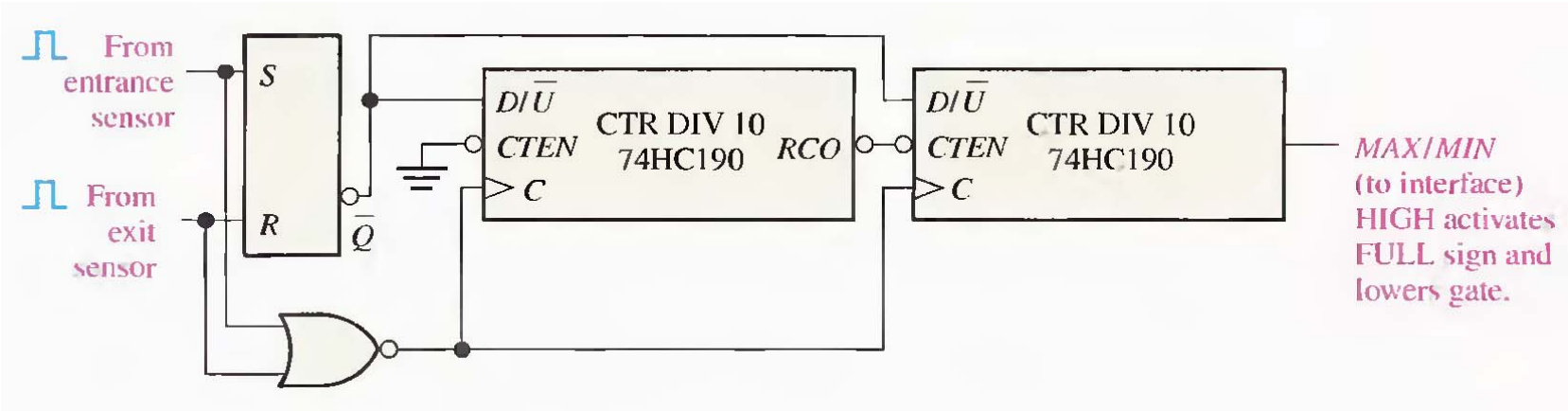
FIGURE 8–52

Logic diagram of typical divide-by-60 counter using 74F162 synchronous decade counters. Note that the outputs are in binary order (the right-most bit is the LSB).

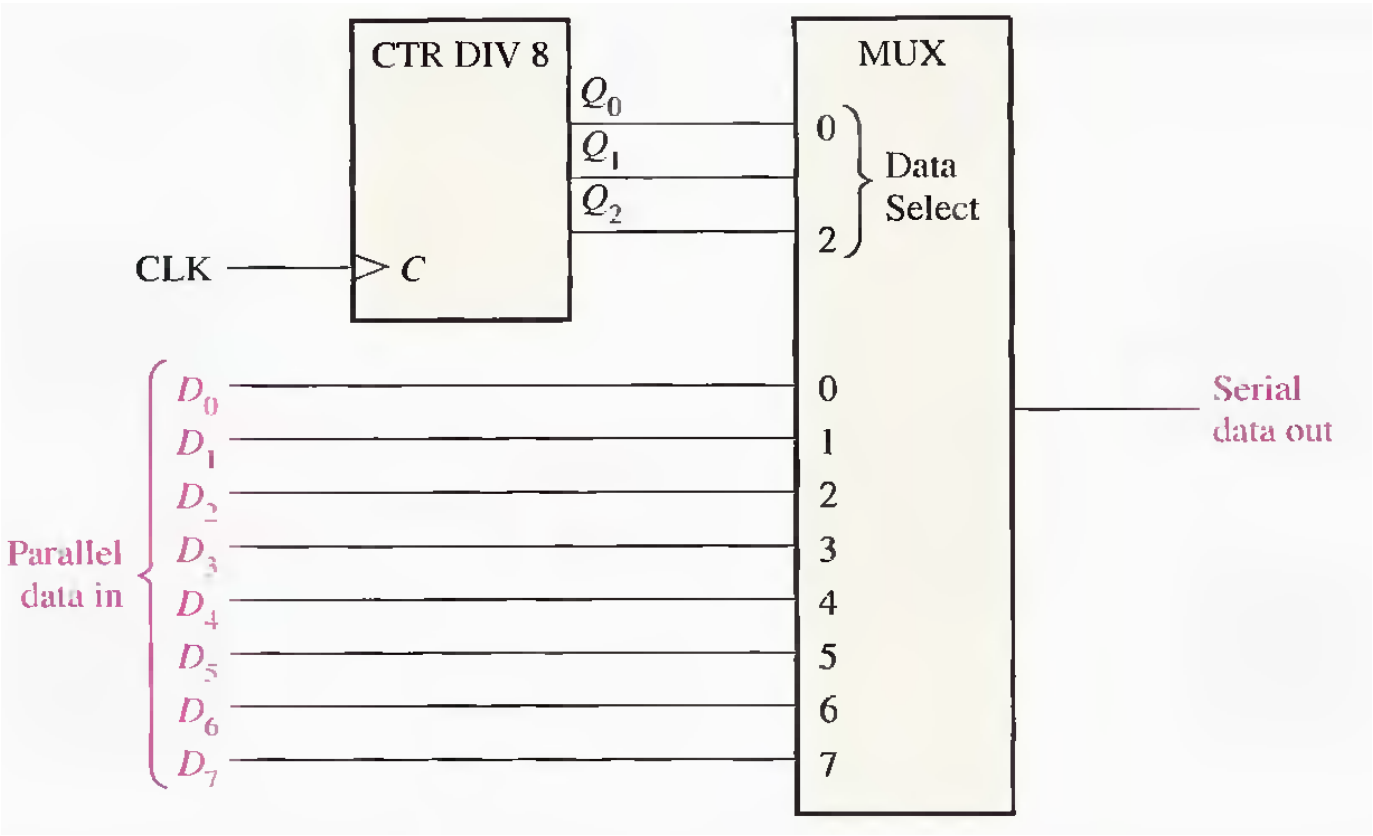

0 0 0 1
D_3 D_2 D_1 D_0
$\overline{PE}$
CTR DIV 10
74F162
CLK
Q_3 Q_2 Q_1 Q_0
J
K
Q
G_1
Decode 9
G_2
Decode 12
8 4 2 1
BCD/7-seg
74LS47
g f e d c b a
8 4 2 1
BCD/7-seg
74LS47
g f e d c b a
To units-of-hours display
To tens-of-hours display

Automobile Parking Control



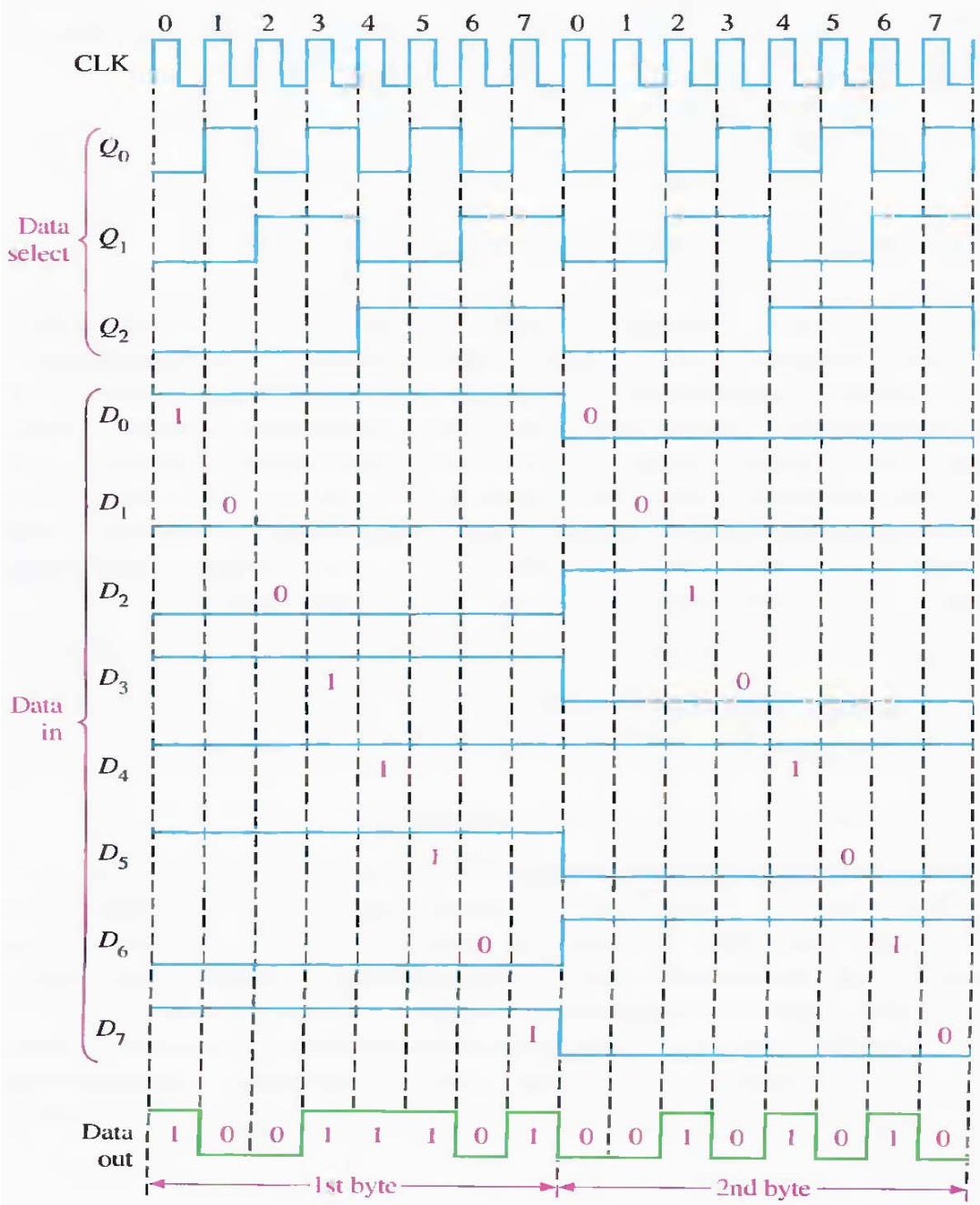


Parallel-to-Serial Data Conversion (Multiplexing)



COMPUTER NOTE

Computers contain an internal counter that can be programmed for various frequencies and tone durations, thus producing "music." To select a particular tone, the programmed instruction selects a divisor that is sent to the counter. The divisor sets the counter up to divide the basic peripheral clock frequency to produce an audio tone. The duration of a tone can also be set by a programmed instruction; thus, a basic counter is used to produce melodies by controlling the frequency and duration of tones.



SECTION 8-7 REVIEW

1. Explain the purpose of each NAND gate in Figure 8–53.
2. Identify the two recycle conditions for the hours counter in Figure 8–51, and explain the reason for each.

8–8 LOGIC SYMBOLS WITH DEPENDENCY NOTATION

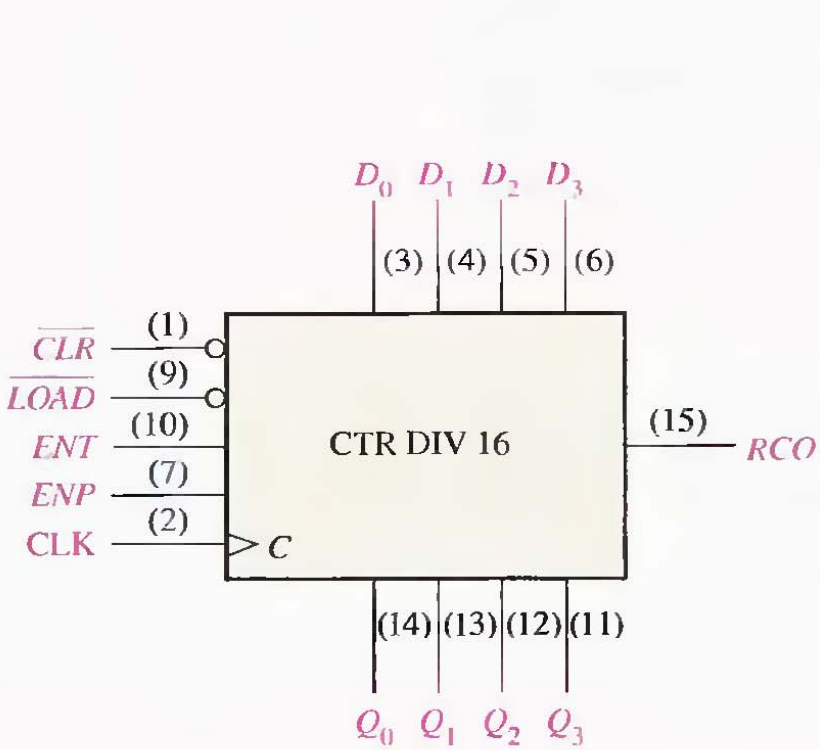
The 74HC163 4-bit synchronous binary counter is used for illustration. For comparison, Figure 8–58 shows a traditional block symbol and the ANSI/IEEE symbol with dependency notation. Basic descriptions of the symbol and the dependency notation follow.

Common Control Block The upper block with notched corners in Figure 8–58(b) has inputs and an output that are considered common to all elements in the device and not unique to any one of the elements.

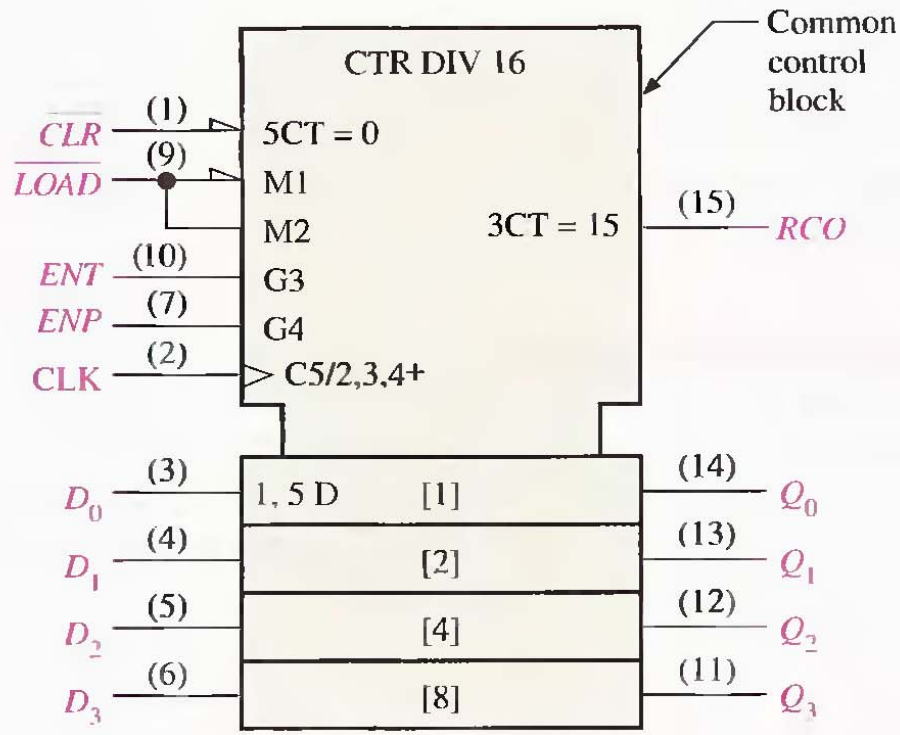
Individual Elements The lower block in Figure 8–58(b), which is partitioned into four abutted sections, represents the four storage elements (D flip-flops) in the counter, with inputs D_0, D_1, D_2, and D_3 and outputs Q_0, Q_1, Q_2, and Q_3.

Qualifying Symbol The label "CTR DIV 16" in Figure 8–58(b) identifies the device as a counter (CTR) with sixteen states (DIV 16).

Control Dependency (C) As shown in Figure 8–58(b), the letter *C* denotes control dependency. Control inputs usually enable or disable the data inputs (*D, J, K, S,* and *R*) of a storage element. The *C* input is usually the clock input. In this case the digit 5 following *C* (C5/2,3,4+) indicates that the inputs labeled with a 5 prefix are dependent on the clock (synchronous with the clock). For example, 5CT = 0 on the $\overline{CLR}$ input indicates that the clear function is dependent on the clock; that is, it is a synchronous clear. When the $\overline{CLR}$ input is LOW (0), the counter is reset to zero ($CT = 0$) on the triggering edge of the clock



(a) Traditional block symbol



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

pulse. Also, the 5 D label at the input of storage element [1] indicates that the data storage is dependent on (synchronous with) the clock. All labels in the [1] storage element apply to the [2], [4], and [8] elements below it since they are not labeled differently.

Mode Dependency (M) As shown in Figure 8–58(b), the letter *M* denotes mode dependency. This label is used to indicate how the functions of various inputs or outputs depend on the mode in which the device is operating. In this case the device has two modes of operation. When the $\overline{LOAD}$ input is LOW (0), as indicated by the triangle input, the counter is in a preset mode (M1) in which the input data (D_0, D_1, D_2, and D_3) are synchronously loaded into the four flip-flops. The digit 1 following *M* in M1 and the 1 in the label 1, 5 D show a dependency relationship and indicate that input data are stored only when the device is in the preset mode (M1), in which $\overline{LOAD} = 0$. When the $\overline{LOAD}$ input is HIGH (1), the counter advances through its normal binary sequence, as indicated by M2 and the 2 in C5/2,3,4+.

AND Dependency (G) As shown in Figure 8–58(b), the letter *G* denotes AND dependency, indicating that an input designated with *G* followed by a digit is ANDed with any other input or output having the same digit as a prefix in its label. In this particular example, the G3 at the *ENT* input and the 3CT = 15 at the *RCO* output are related, as indicated by the 3, and that relationship is an AND dependency, indicated by the *G*. This tells us that *ENT* must be HIGH (no triangle on the input) *and* the count must be fifteen ($CT = 15$) for the *RCO* output to be HIGH.

Also, the digits 2, 3, and 4 in the label C5/2,3,4+ indicate that the counter advances through its states when $\overline{LOAD} = 1$, as indicated by the mode dependency label M2, and when $ENT = 1$ and $ENP = 1$, as indicated by the AND dependency labels G3 and G4. The + indicates that the counter advances by one count when these conditions exist.

SECTION 8-8 REVIEW

1. In dependency notation, what do the letters *C*, *M*, and *G* stand for?
2. By what letter is data storage denoted?

8–9 TROUBLESHOOTING

The troubleshooting of counters can be simple or quite involved, depending on the type of counter and the type of fault. This section will give you some insight into how to approach the troubleshooting of sequential circuits.

After completing this section, you should be able to

- Detect a faulty counter
- Isolate faults in maximum-modulus cascaded counters
- Isolate faults in cascaded counters with truncated sequences
- Determine faults in counters implemented with individual flip-flops

DIGITAL SYSTEM APPLICATION

The traffic light control system that was started in Chapter 6 and continued in Chapter 7 is completed in this chapter. In Chapter 6, the combinational logic was developed.

In Chapter 7, the timing circuits were developed.

In this chapter, the sequential logic is developed and all the blocks are connected to produce the complete traffic control system. The overall system block diagram is shown again in Figure 8–63.

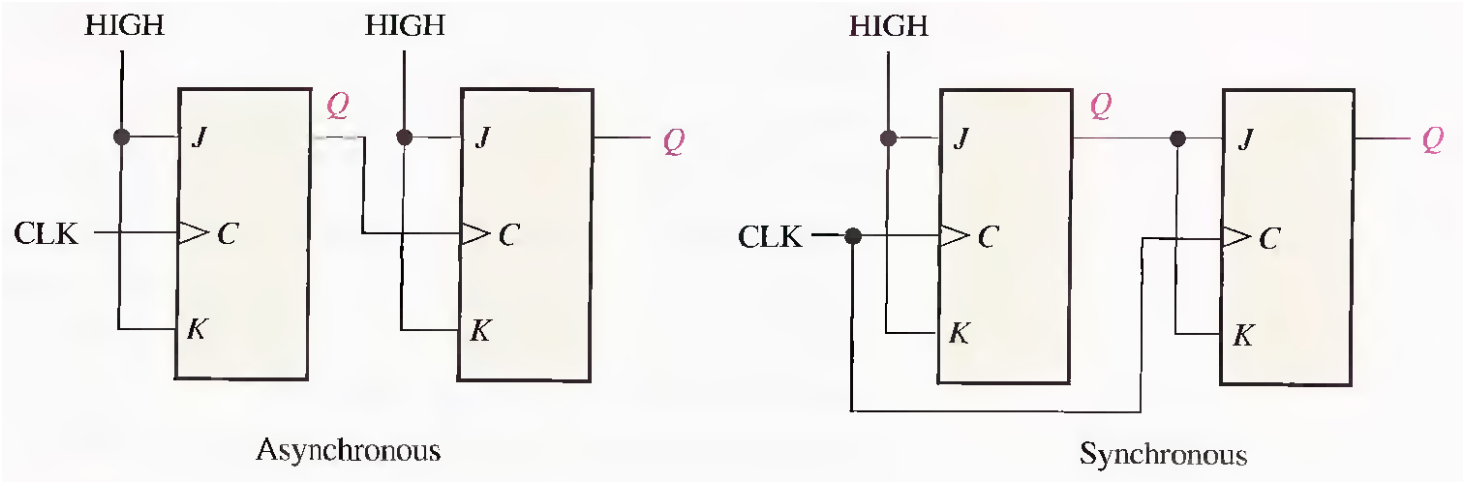
Sequential Logic Requirements

The sequential logic controls the sequencing of the traffic lights based on inputs from the timing circuits and the vehicle sensor. The sequential logic will produce a 2-bit Gray code sequence for the four states of the system that are indicated in Figure 8–64.

Block Diagram The sequential logic consists of a 2-bit Gray code counter and associated input logic, as shown in Figure 8–65.

SUMMARY

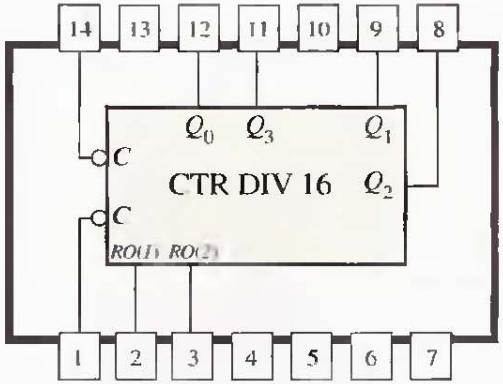
- Asynchronous and synchronous counters differ only in the way in which they are clocked, as shown in Figure 8–71. Synchronous counters can run at faster clock rates than asynchronous counters.



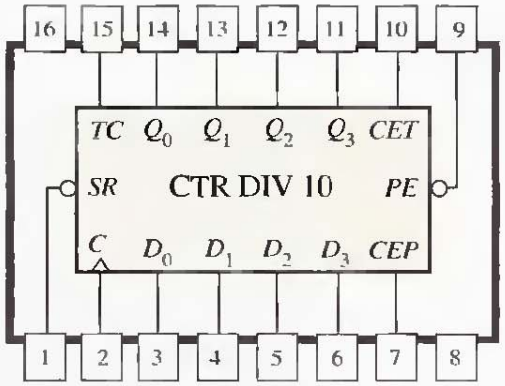
- Connection diagrams for the IC counters introduced in this chapter are shown in Figure 8–72.
- The maximum modulus of a counter is the maximum number of possible states and is a function of the number of stages (flip-flops). Thus,

$$\text{Maximum modulus} = 2^n$$

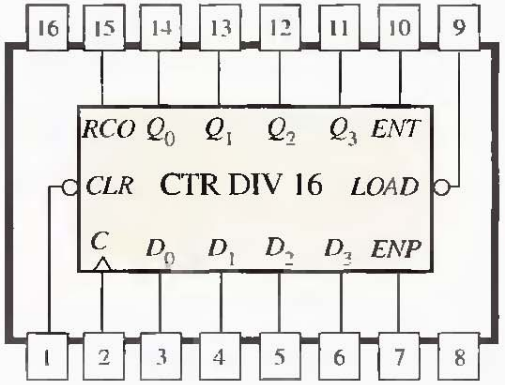
  where n is the number of stages in the counter. The modulus of a counter is the *actual* number of states in its sequence and can be equal to or less than the maximum modulus.
- The overall modulus of cascaded counters is equal to the product of the moduli of the individual counters.



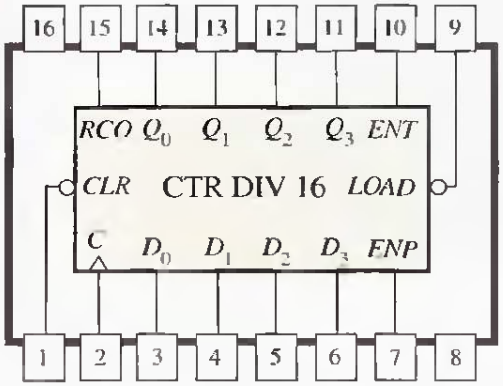
74LS93 4-bit asynchronous binary counter



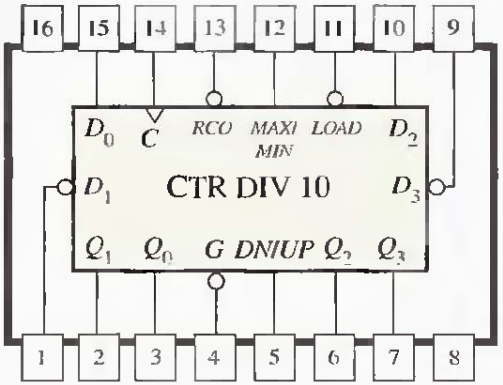
74F162 synchronous BCD decade counter with asynchronous clear



74HC161 4-bit synchronous binary counter with asynchronous clear



74HC163 4-bit synchronous binary counter with synchronous clear



74HC190 synchronous up/down decade counter
(C is Count Enable.)

KEY TERMS

Asynchronous Not occurring at the same time.

Cascade To connect "end-to-end" as when several counters are connected from the terminal count output of one counter to the enable input of the next counter.

Decade Characterized by ten states or values.

Modulus The number of unique states through which a counter will sequence.

Recycle To undergo transition (as in a counter) from the final or terminal state back to the initial state.

State diagram A graphic depiction of a sequence of states or values.

State machine A logic system exhibiting a sequence of states conditioned by internal logic and external inputs; any sequential circuit exhibiting a specified sequence of states.

Synchronous Occurring at the same time.

Terminal count The final state in a counter's sequence.

به نام خدا

فصل نهم: رجیسترها (ثباتها)

9 SHIFT REGISTERS

CHAPTER OUTLINE

9–1 Basic Shift Register Functions
9–2 Serial In/Serial Out Shift Registers
9–3 Serial In/Parallel Out Shift Registers
9–4 Parallel In/Serial Out Shift Registers
9–5 Parallel In/Parallel Out Shift Registers
9–6 Bidirectional Shift Registers
9–7 Shift Register Counters
9–8 Shift Register Applications
9–9 Logic Symbols with Dependency Notation
9–10 Troubleshooting
Digital System Application

CHAPTER OBJECTIVES

- Identify the basic forms of data movement in shift registers
- Explain how serial in/serial out, serial in/parallel out, parallel in/serial out, and parallel in/parallel out shift registers operate
- Describe how a bidirectional shift register operates
- Determine the sequence of a Johnson counter
- Set up a ring counter to produce a specified sequence
- Construct a ring counter from a shift register
- Use a shift register as a time-delay device
- Use a shift register to implement a serial-to-parallel data converter
- Implement a basic shift-register-controlled keyboard encoder
- Interpret ANSI/IEEE Standard 91-1984 shift register symbols with dependency notation
- Use shift registers in a system application

KEY TERMS

- Register
- Stage
- Shift
- Load
- Bidirectional

INTRODUCTION

Shift registers are a type of sequential logic circuit closely related to digital counters. Registers are used primarily for the storage of digital data and typically do not possess a characteristic internal sequence of states as do counters. There are exceptions, however, and these are covered in Section 9–7.

In this chapter, the basic types of shift registers are studied and several applications are presented. Also, a troubleshooting method is introduced.

9–1 BASIC SHIFT REGISTER FUNCTIONS

A **register** is a digital circuit with two basic functions: data storage and data movement. The storage capability of a register makes it an important type of memory device.

FIGURE 9–1

The flip-flop as a storage element.

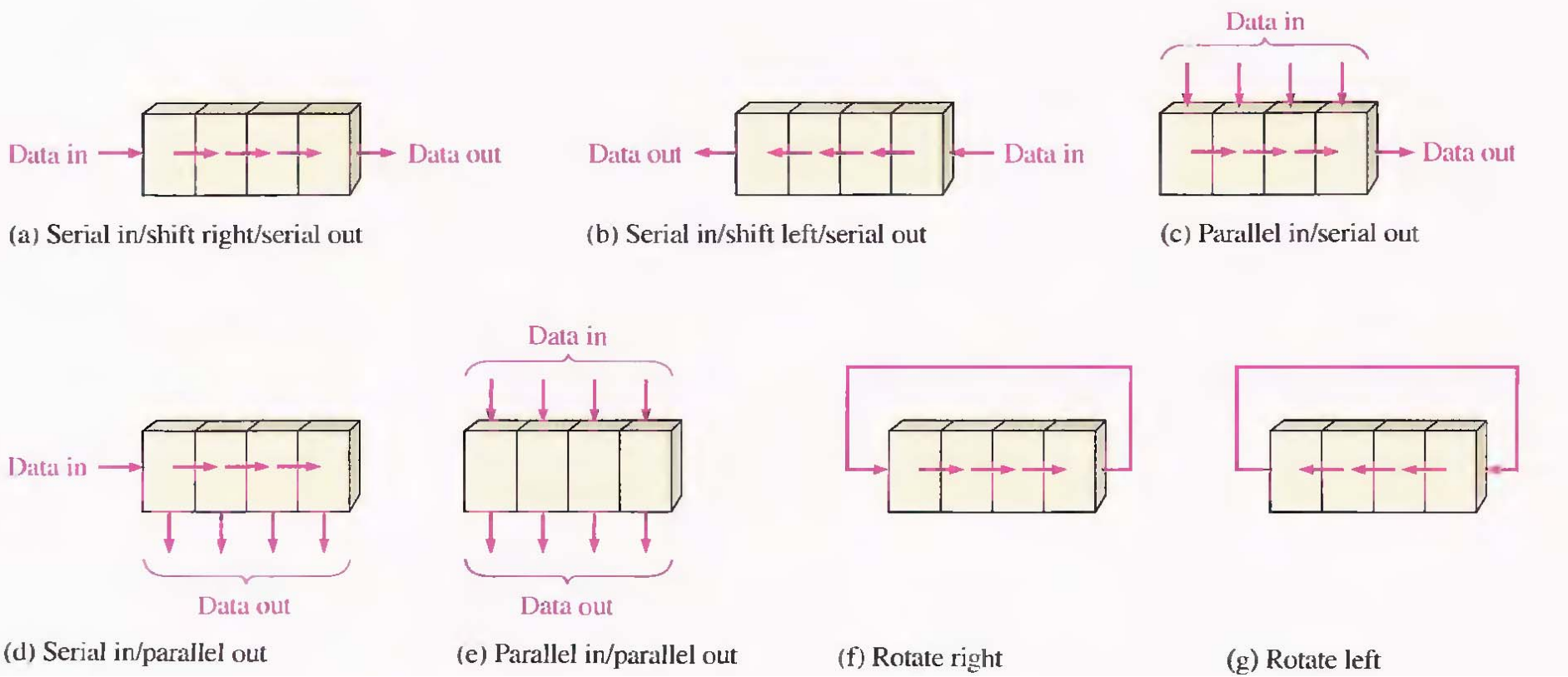
SECTION 9–1 REVIEW

Answers are at the end of the chapter.

1. Generally, what is the difference between a counter and a shift register?
2. What two principal functions are performed by a shift register?

has no specified sequence of states, except in certain very specialized applications.

A register can consist of one or more flip-flops used to store and shift data.



(a) Serial in/shift right/serial out

(b) Serial in/shift left/serial out

(c) Parallel in/serial out

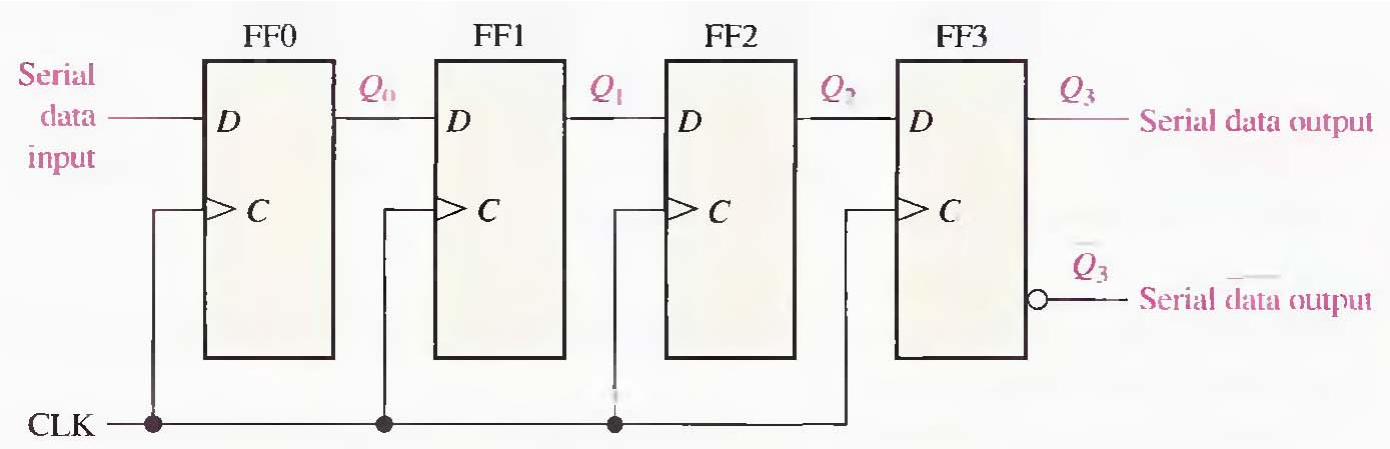
(d) Serial in/parallel out

(e) Parallel in/parallel out

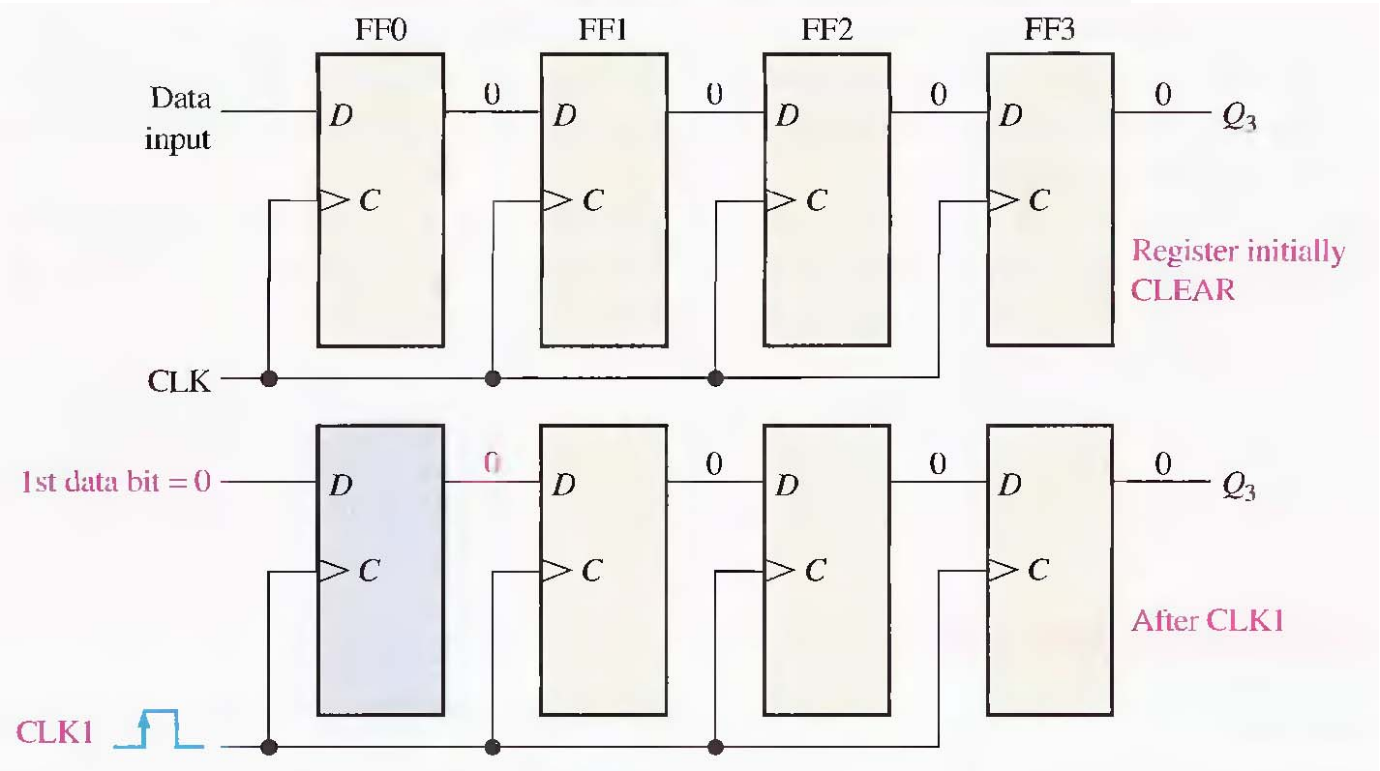
(f) Rotate right

(g) Rotate left

9–2 SERIAL IN/SERIAL OUT SHIFT REGISTERS



Four bits (1010) being entered serially into the register.



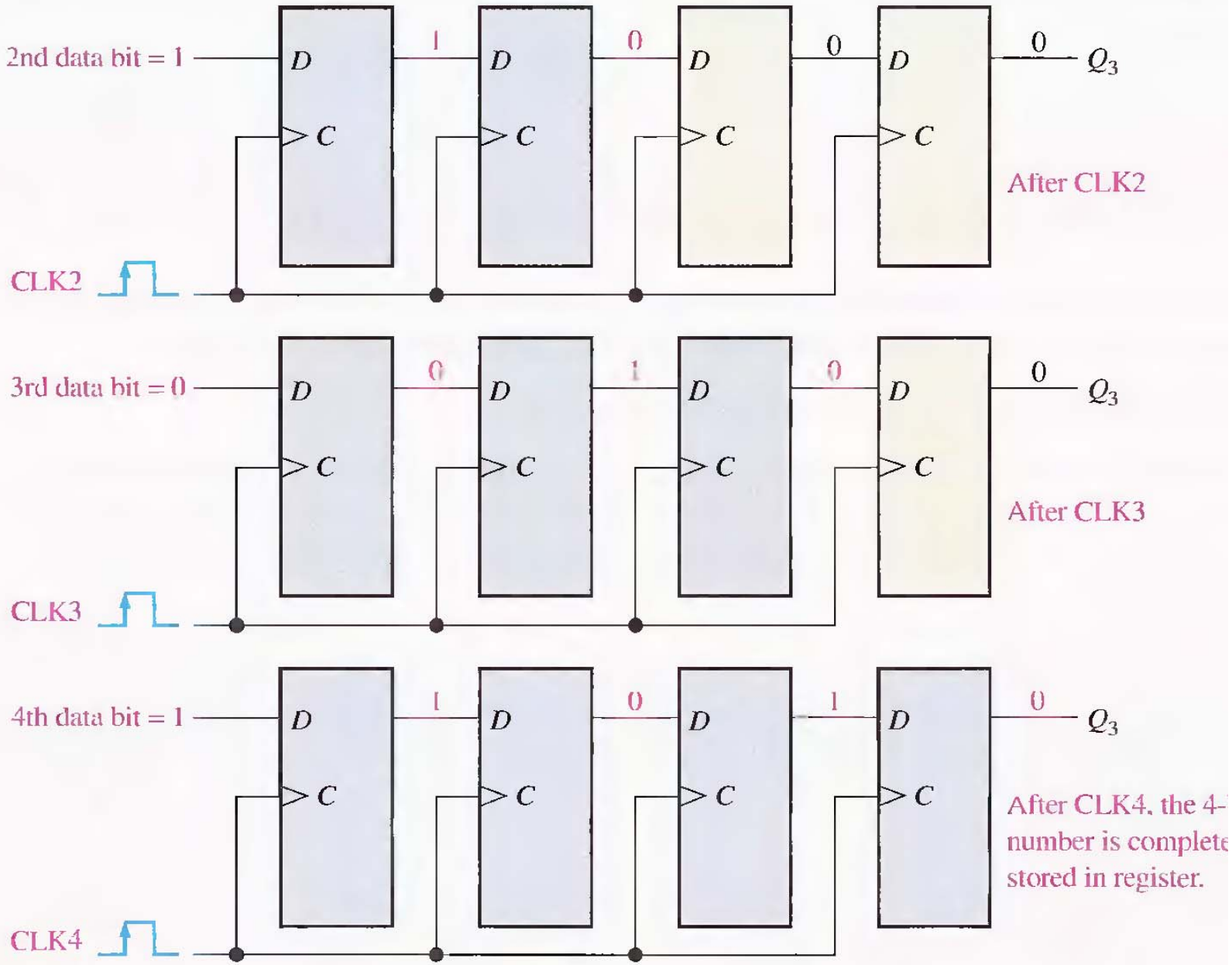
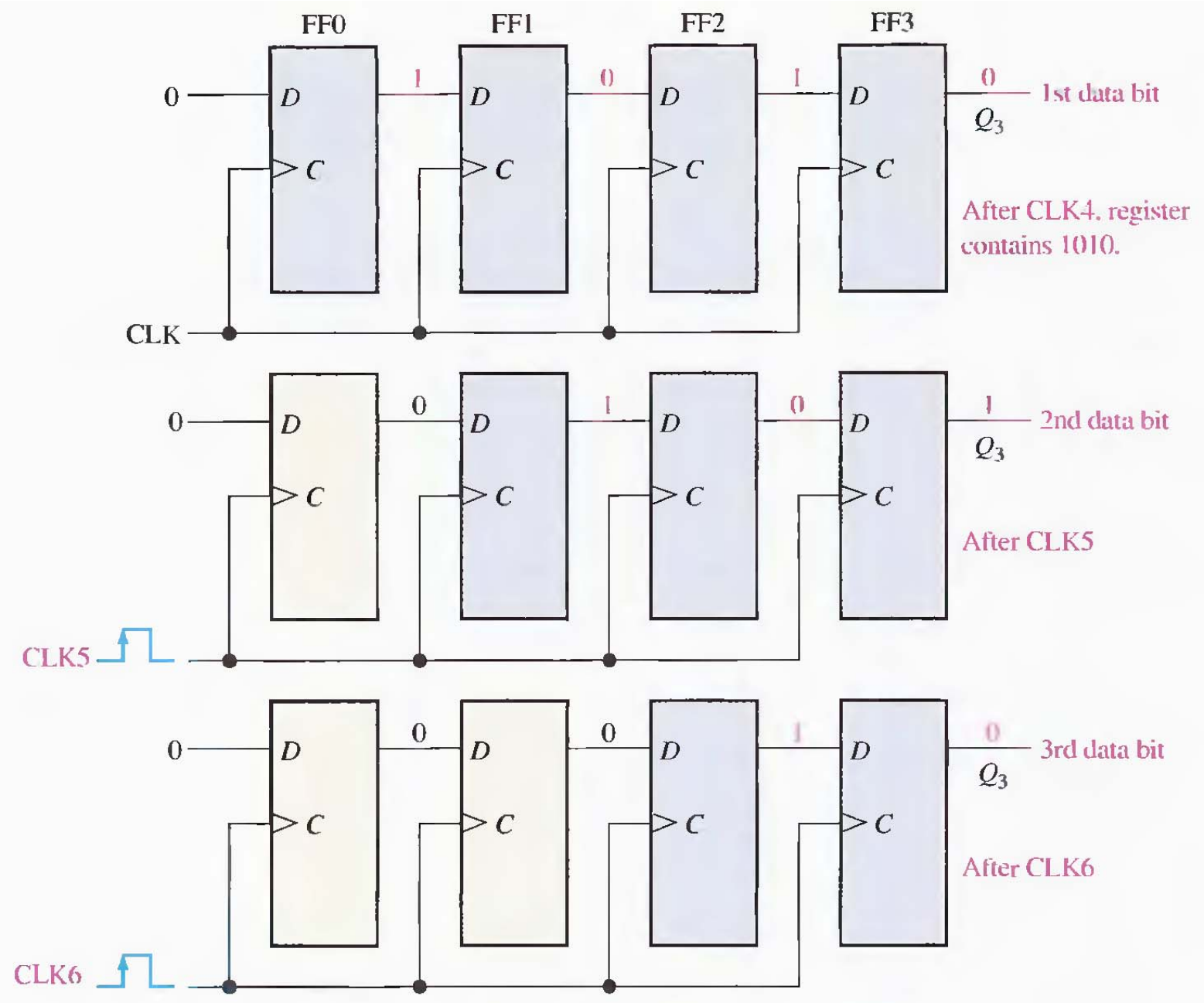


FIGURE 9–4

Four bits (1010) being entered serially into the register.

Four bits (1010) being serially shifted out of the register and replaced by all zeros.



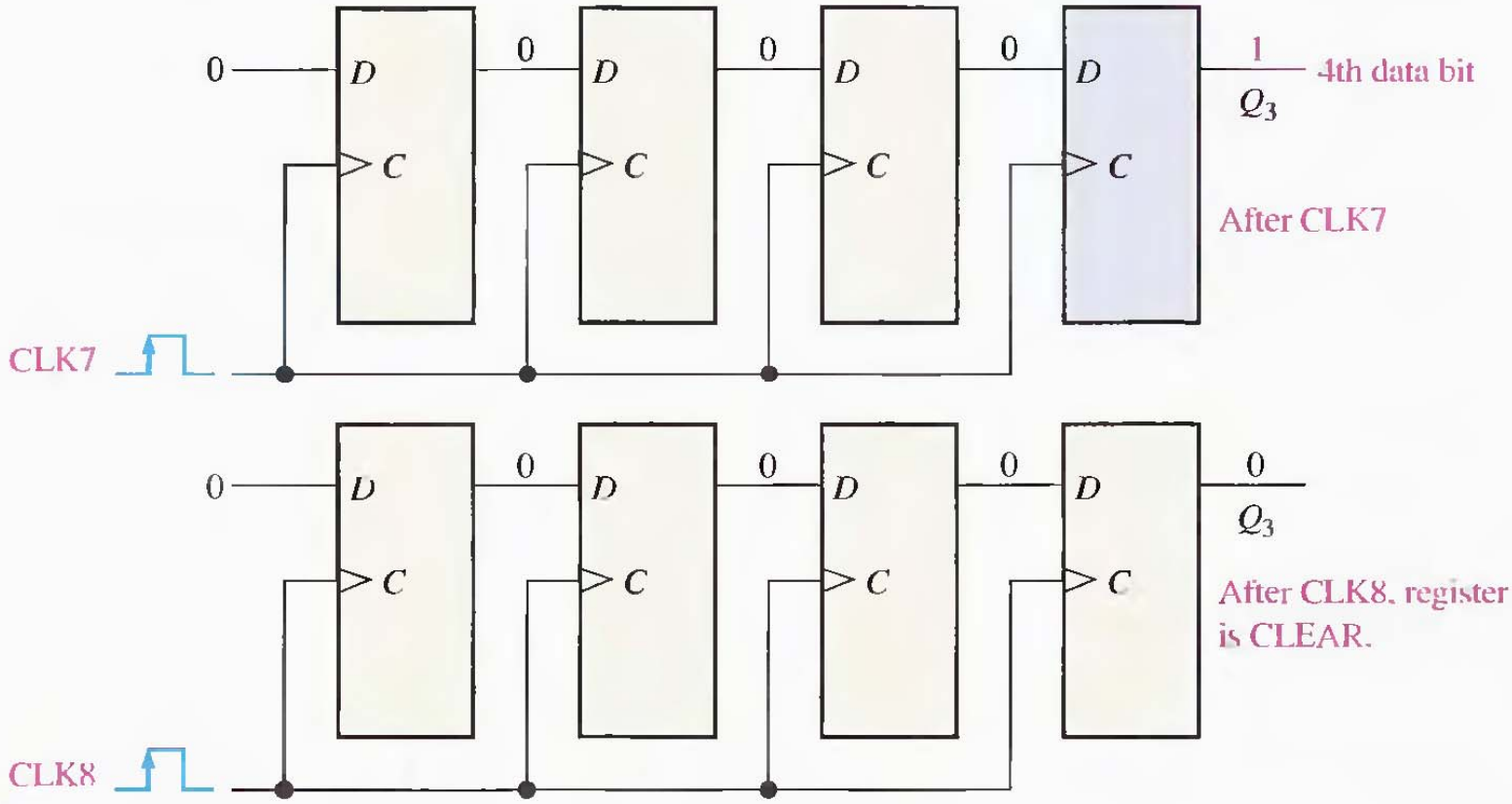
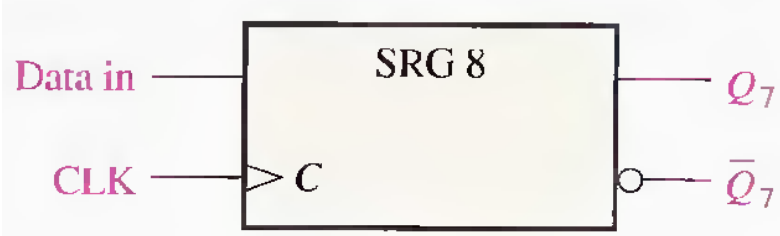


FIGURE 9–5

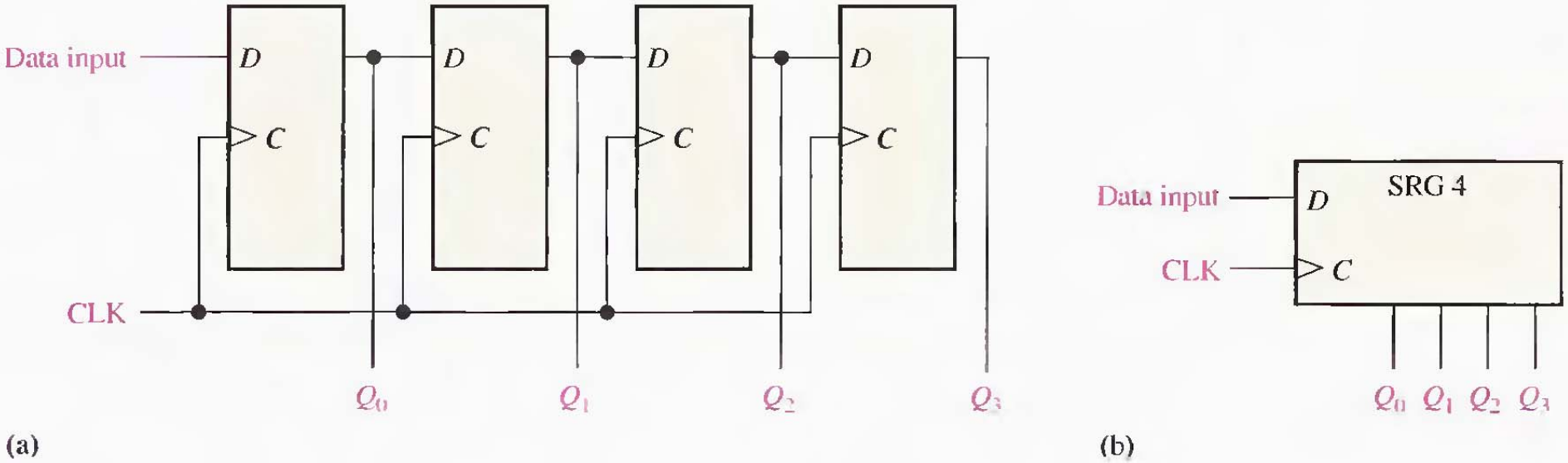
Four bits (1010) being serially shifted out of the register and replaced by all zeros.



SECTION 9-2 REVIEW

1. Develop the logic diagram for the shift register in Figure 9–3, using J-K flip-flops to replace the D flip-flops.
2. How many clock pulses are required to enter a byte of data serially into an 8-bit shift register?

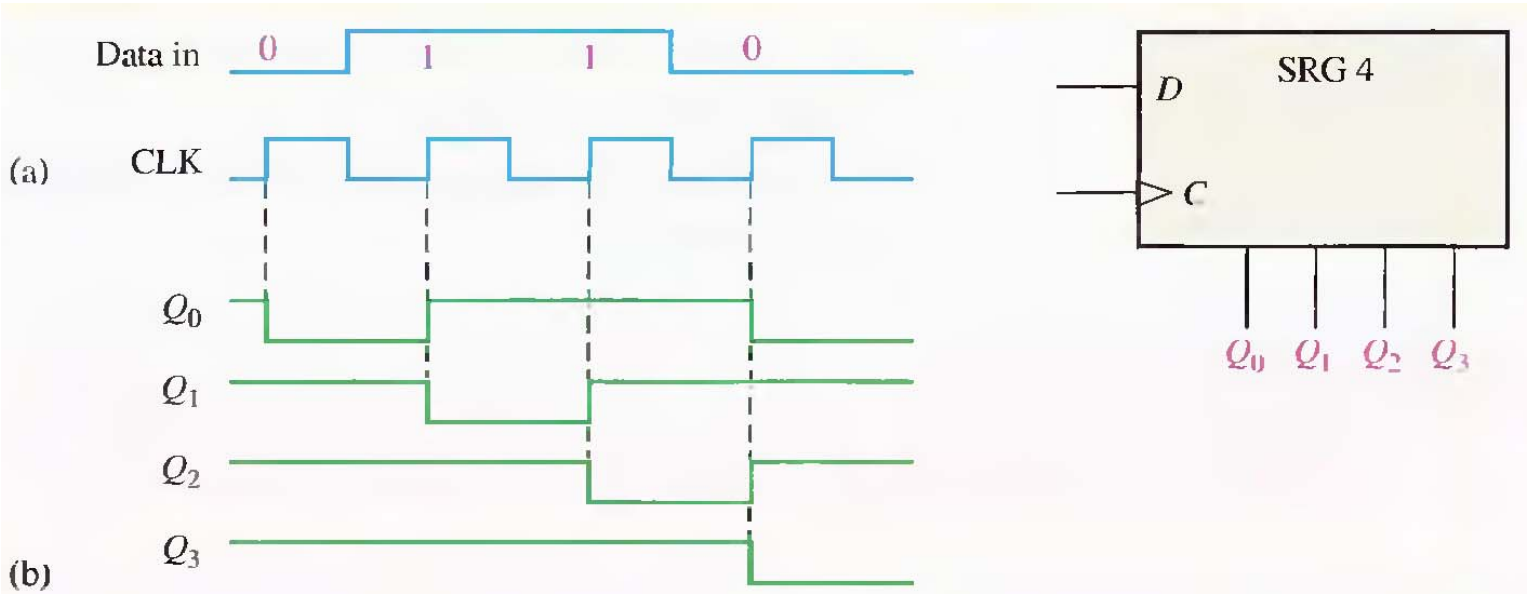
9–3 SERIAL IN/PARALLEL OUT SHIFT REGISTERS



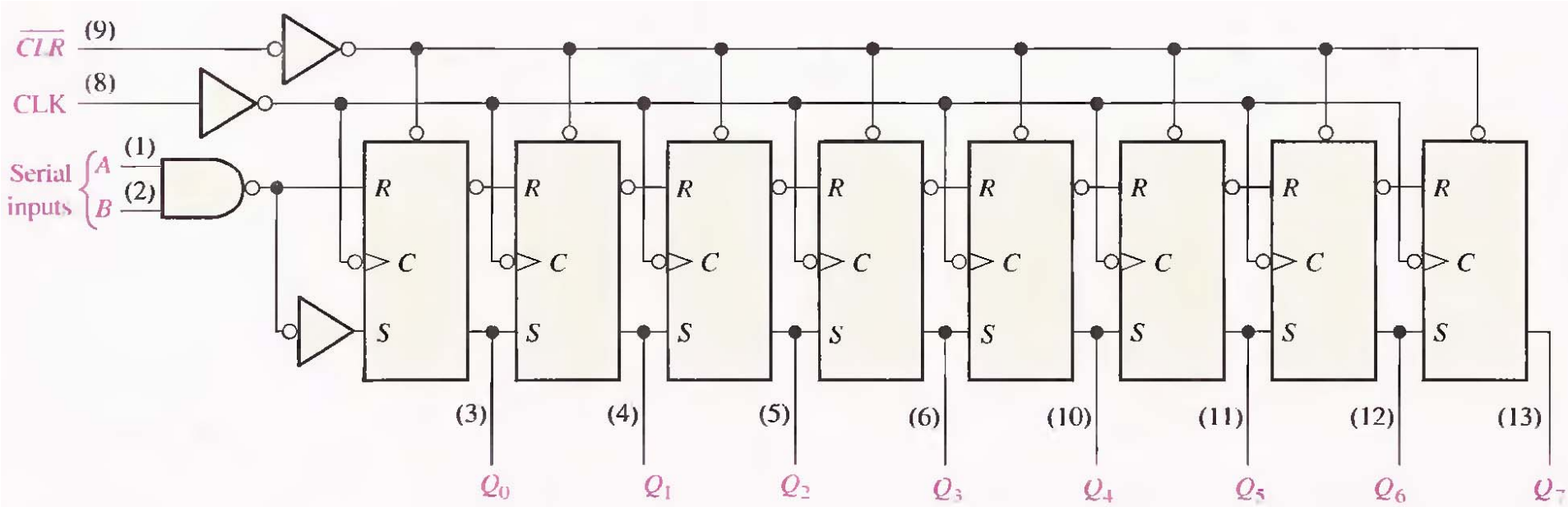
EXAMPLE 9–2

Show the states of the 4-bit register (SRG 4) for the data input and clock waveforms in Figure 9–9(a). The register initially contains all 1s.

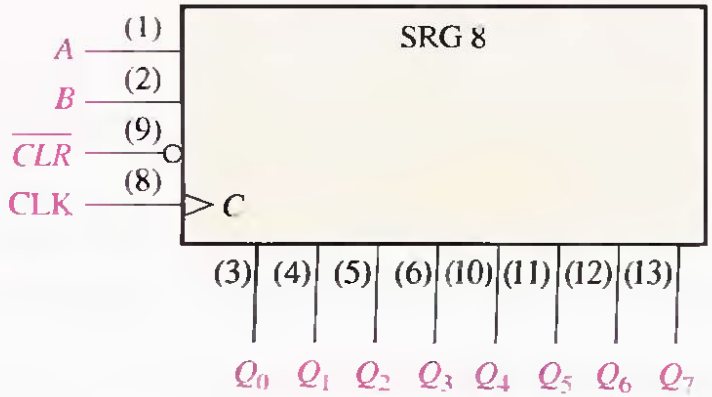
The register contains 0110 after four clock pulses. See Figure 9–9(b).



THE 74HC164 8-BIT SERIAL IN/PARALLEL OUT SHIFT REGISTER

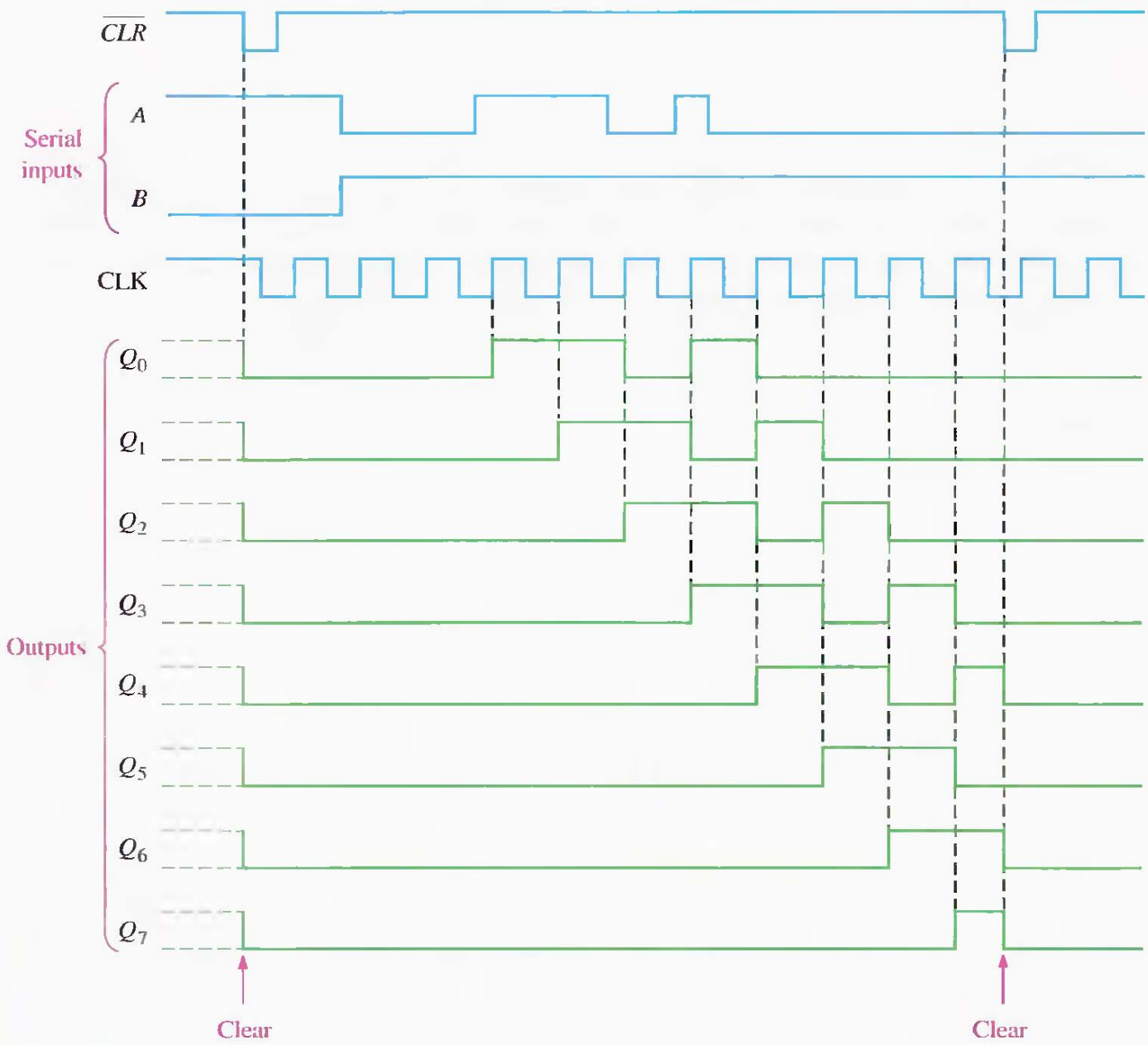


(a) Logic diagram



(b) Logic symbol

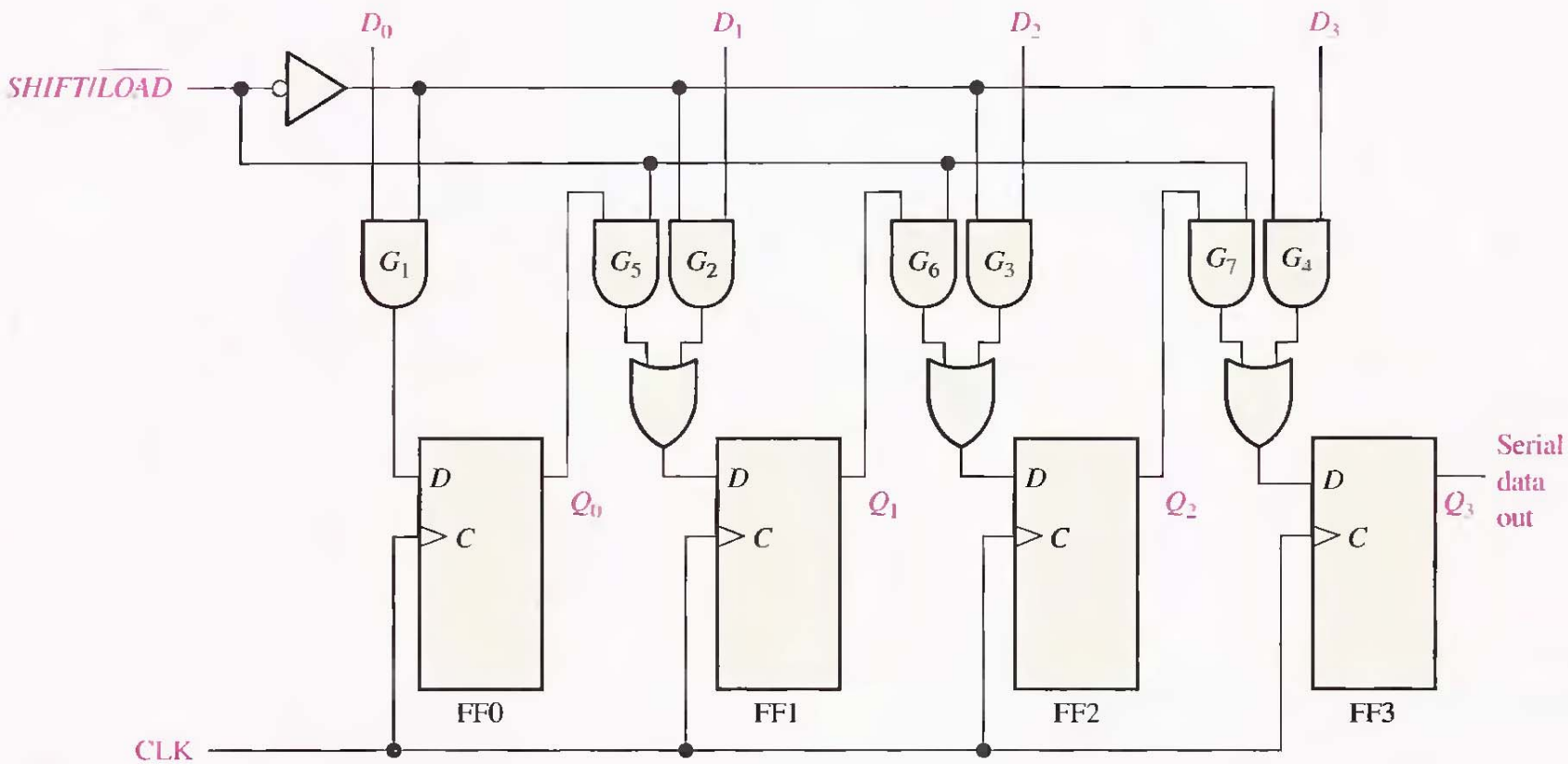
A sample timing diagram for the 74HC164 is shown in Figure 9–11. Notice that the serial input data on input *A* are shifted into and through the register after input *B* goes HIGH.



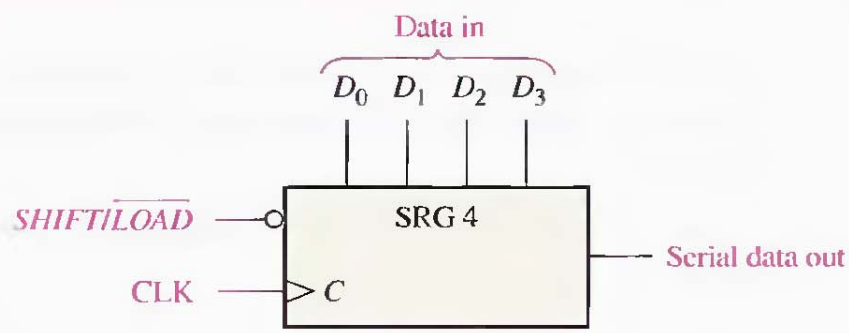
SECTION 9-3 REVIEW

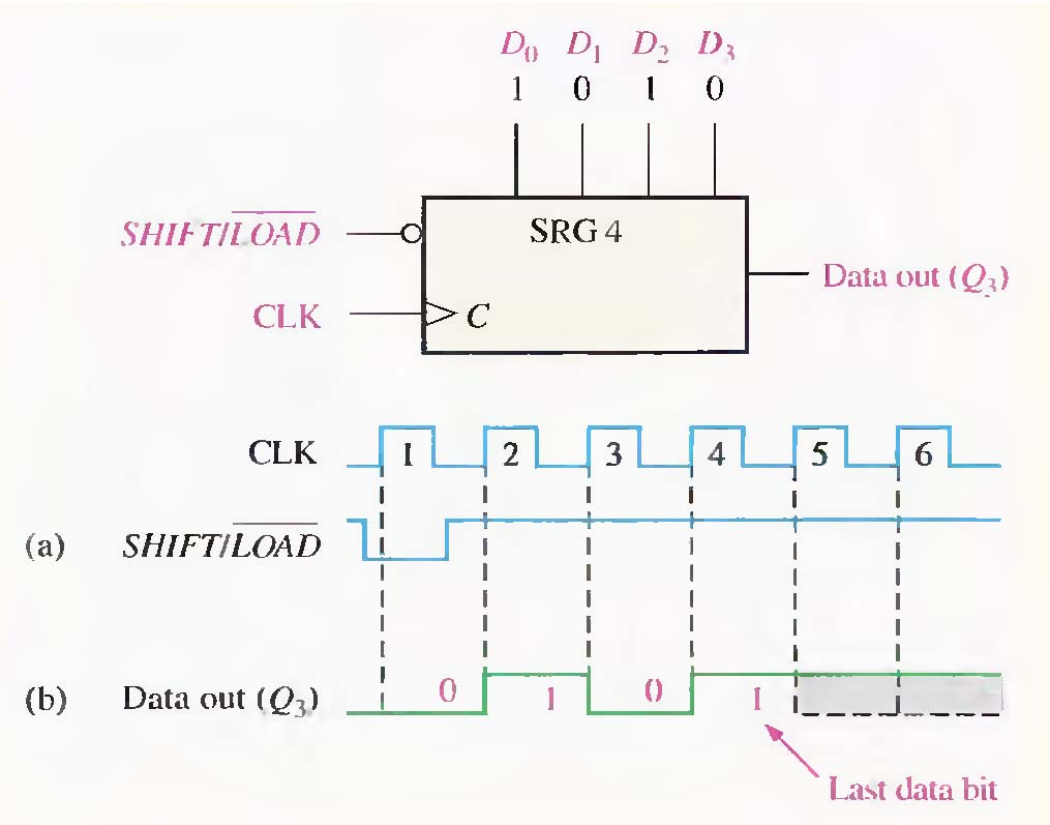
1. The bit sequence 1101 is serially entered (right-most bit first) into a 4-bit parallel out shift register that is initially clear. What are the *Q* outputs after two clock pulses?
2. How can a serial in/parallel out register be used as a serial in/serial out register?

9-4 PARALLEL IN/SERIAL OUT SHIFT REGISTERS

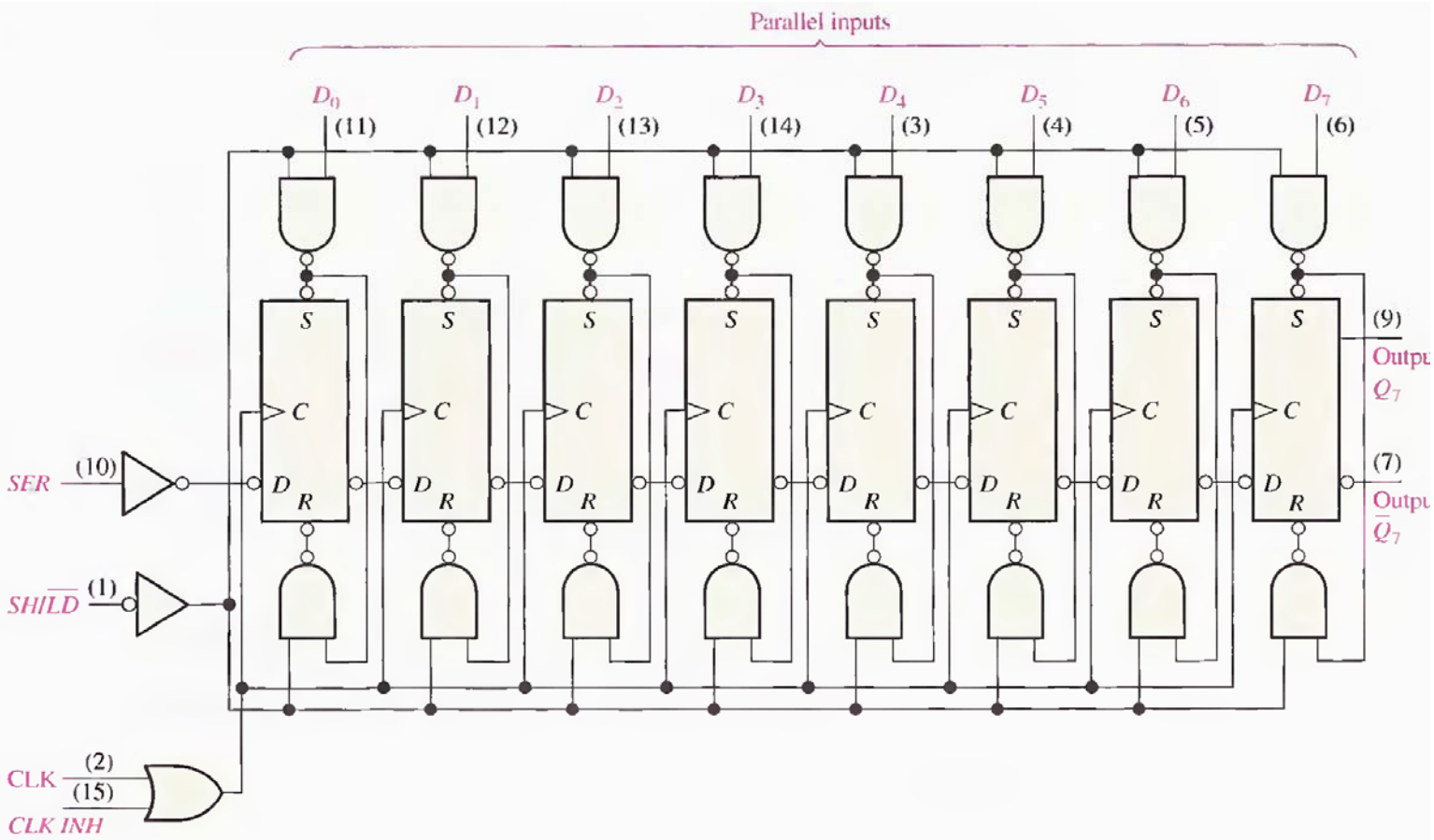


(a) Logic diagram

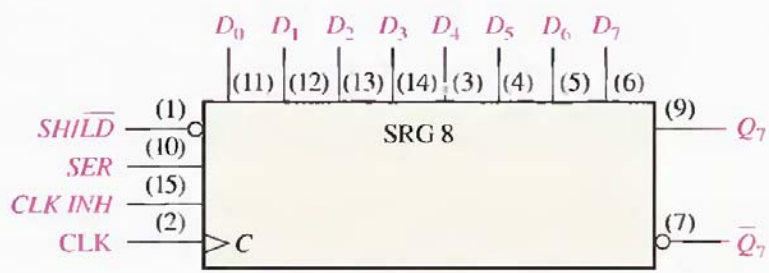



D_0 D_1 D_2 D_3
1 0 1 0
SHIFT/$\overline{LOAD}$
SRG 4
CLK
C
Data out (Q_3)
CLK
1 2 3 4 5 6
(a) SHIFT/$\overline{LOAD}$
(b) Data out (Q_3)
0 1 0 1
Last data bit

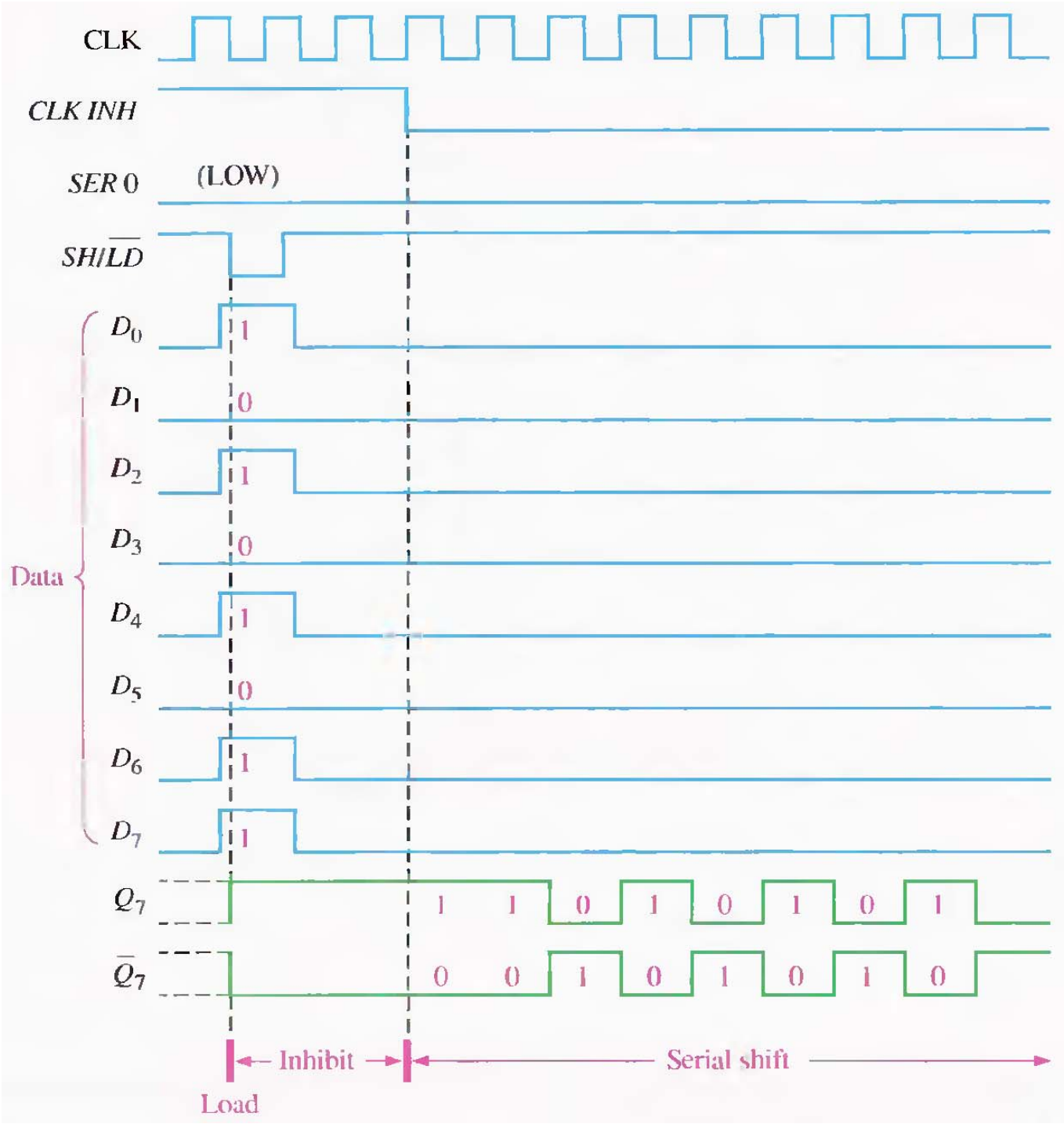
THE 74HC165 8-BIT PARALLEL LOAD SHIFT REGISTER



(a) Logic diagram



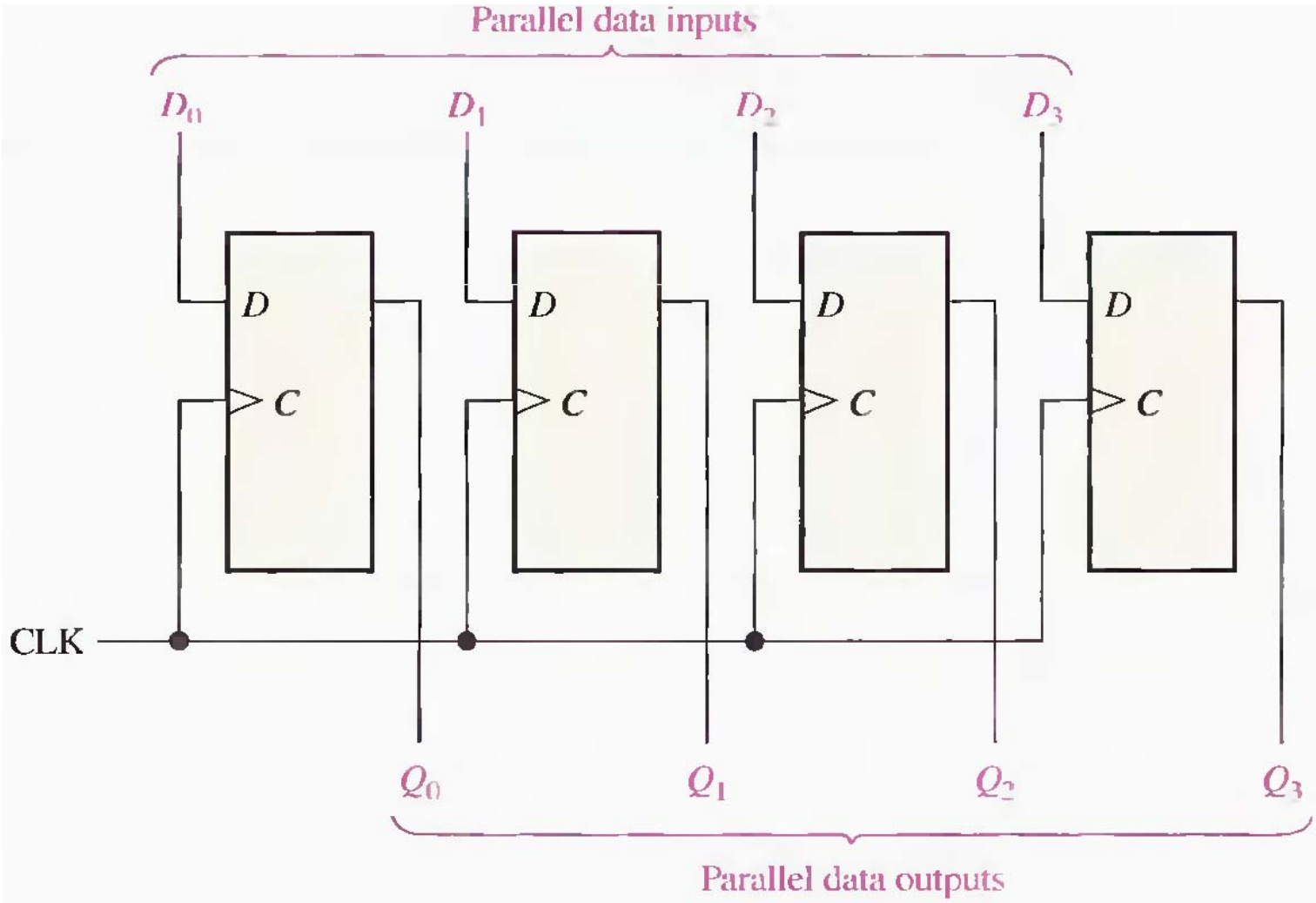
(b) Logic symbol


CLK
CLK INH
SER 0
(LOW)
SH/$\overline{LD}$
D_0
D_1
D_2
D_3
D_4
D_5
D_6
D_7
Data
1
0
1
0
1
0
1
1
Q_7
$\overline{Q}_7$
1 1 0 1 0 1 0 1
0 0 1 0 1 0 1 0
Inhibit
Serial shift
Load

SECTION 9–4 REVIEW

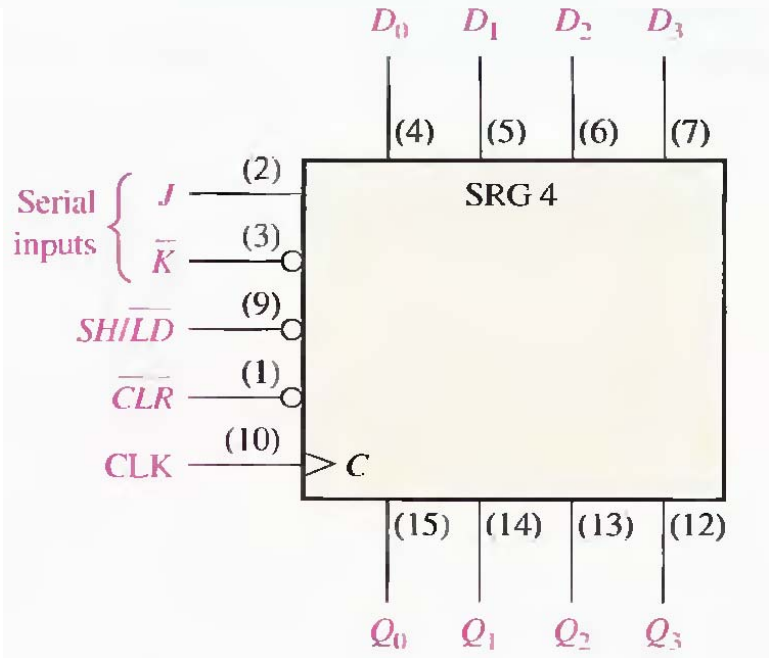
1. Explain the function of the *SHIFT*/$\overline{LOAD}$ input.
2. Is the parallel load operation in a 74HC165 shift register synchronous or asynchronous? What does this mean?

9–5 PARALLEL IN/PARALLEL OUT SHIFT REGISTERS



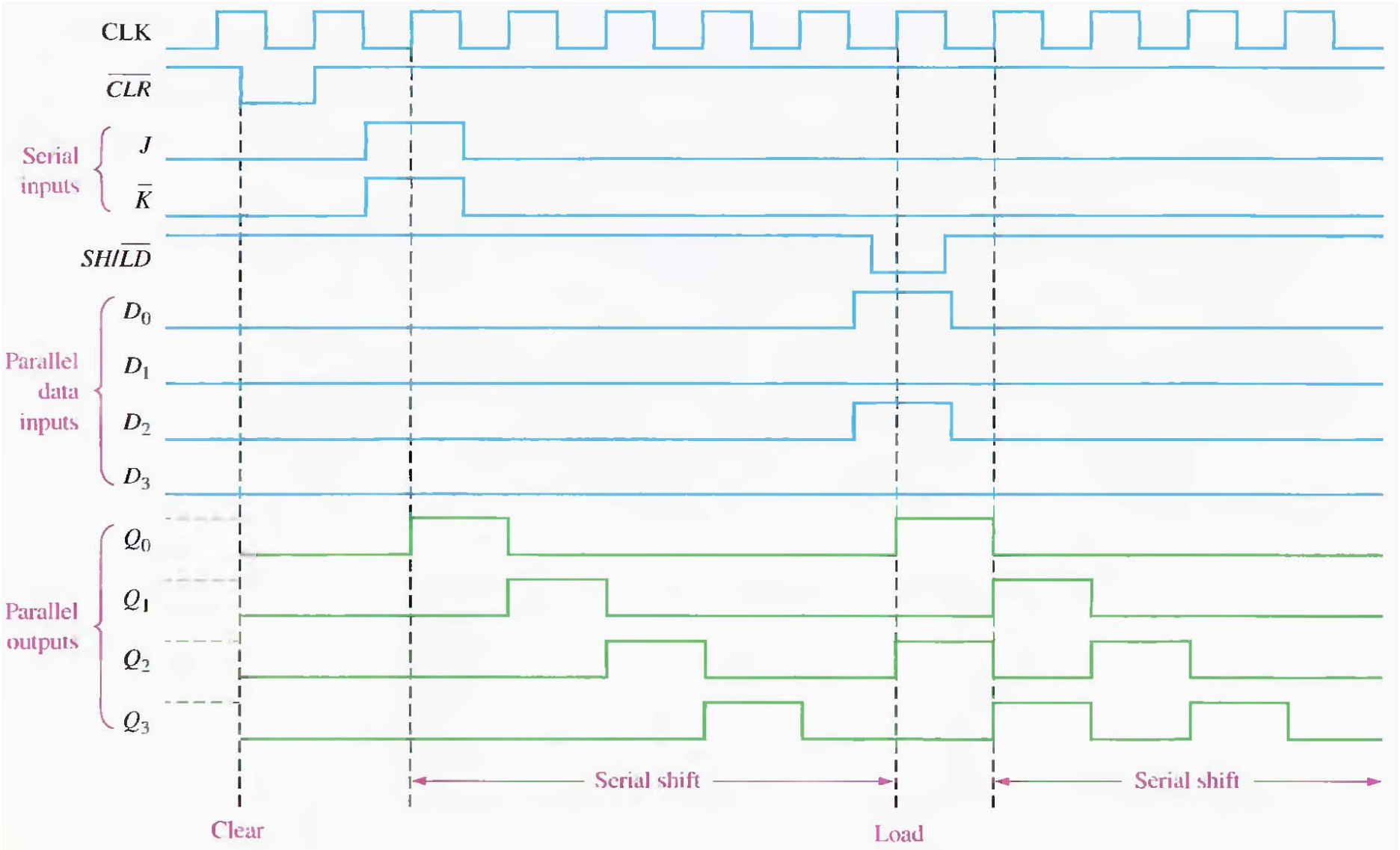
THE 74HC195 4-BIT PARALLEL-ACCESS SHIFT REGISTER

The 74HC195 can be used for parallel in/parallel out operation. Because it also has a serial input, it can be used for serial in/serial out and serial in/parallel out operations. It can be used for parallel in/serial out operation by using Q_3 as the output. A typical logic block symbol is shown in Figure 9–17.



When the $SHIFT/\overline{LOAD}$ input ($SH/\overline{LD}$) is LOW, the data on the parallel inputs are entered synchronously on the positive transition of the clock. When $SH/\overline{LD}$ is HIGH, stored data will shift right (Q_0 to Q_3) synchronously with the clock. Inputs J and $\overline{K}$ are the serial data inputs to the first stage of the register (Q_0); Q_3 can be used for serial output data. The active-LOW clear input is asynchronous.

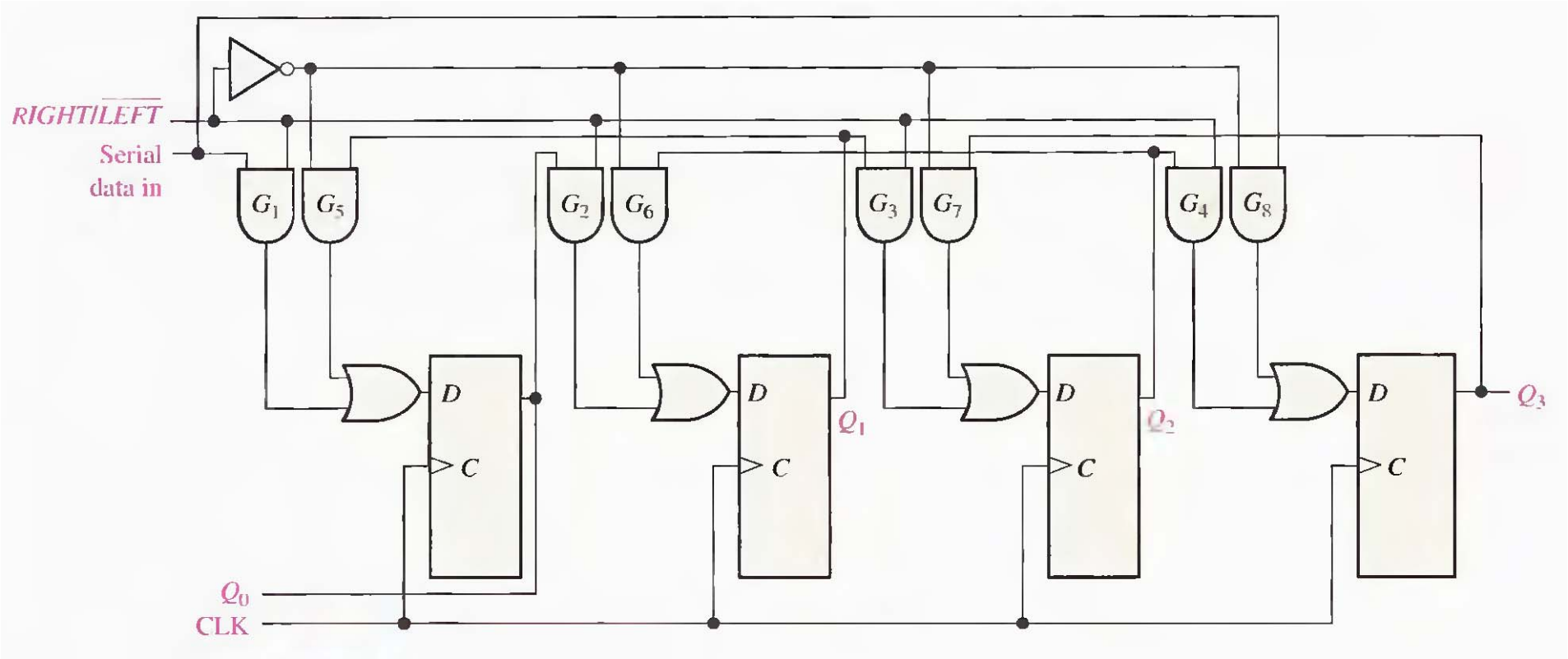
The timing diagram in Figure 9–18 illustrates the operation of this register.



SECTION 9-5 REVIEW

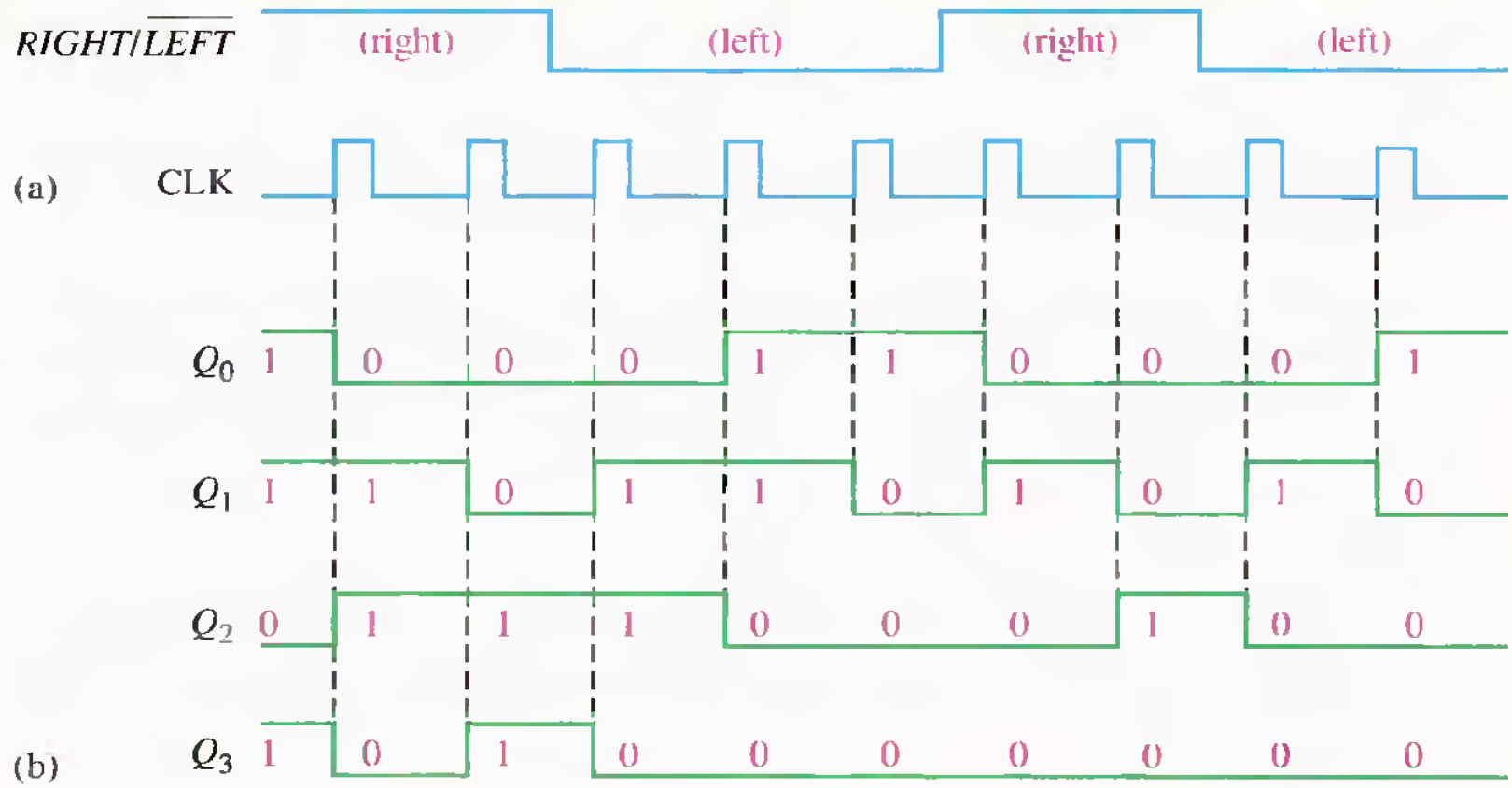
1. In Figure 9–16, $D_0 = 1$, $D_1 = 0$, $D_2 = 0$, and $D_3 = 1$. After three clock pulses, what are the data outputs?
2. For a 74HC195, $SH/\overline{LD} = 1$, $J = 1$, and $\overline{K} = 1$. What is Q_0 after one clock pulse?

9–6 BIDIRECTIONAL SHIFT REGISTERS



EXAMPLE 9–4

Determine the state of the shift register of Figure 9–19 after each clock pulse for the given $RIGHT/\overline{LEFT}$ control input waveform in Figure 9–20(a). Assume that $Q_0 = 1$, $Q_1 = 1$, $Q_2 = 0$, and $Q_3 = 1$ and that the serial data-input line is LOW.



See Figure 9–20(b).

THE 74HC194 4-BIT BIDIRECTIONAL UNIVERSAL SHIFT REGISTER

The 74HC194 is an example of a universal bidirectional shift register in integrated circuit form. A **universal shift register** has both serial and parallel input and output capability. A logic block symbol is shown in Figure 9–21, and a sample timing diagram is shown in Figure 9–22.

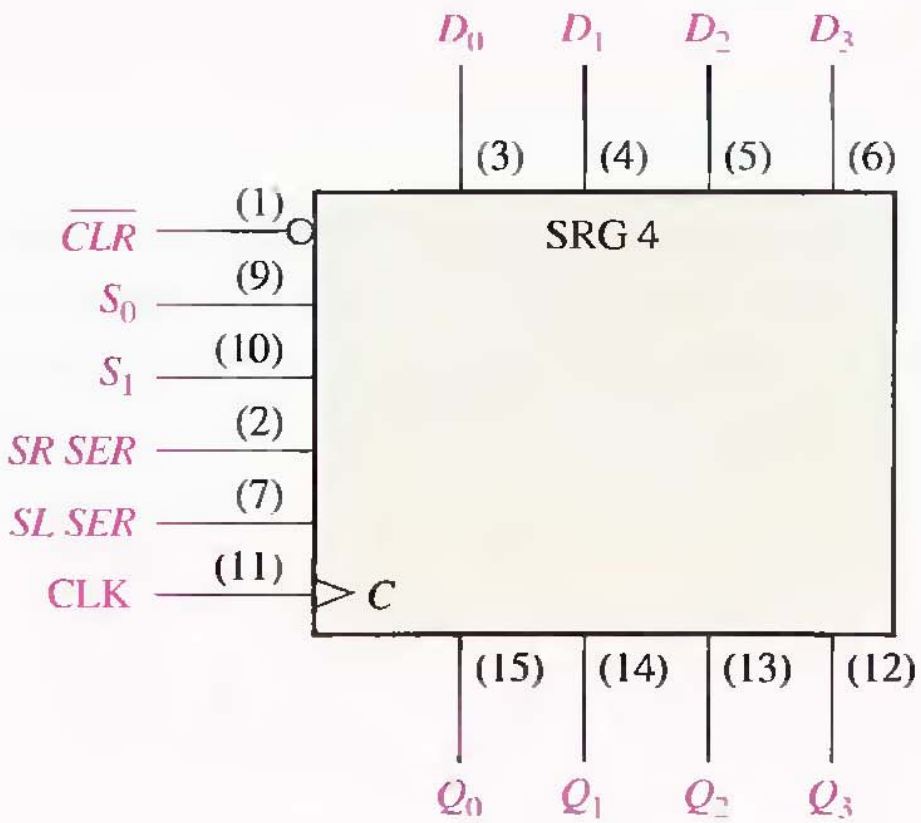
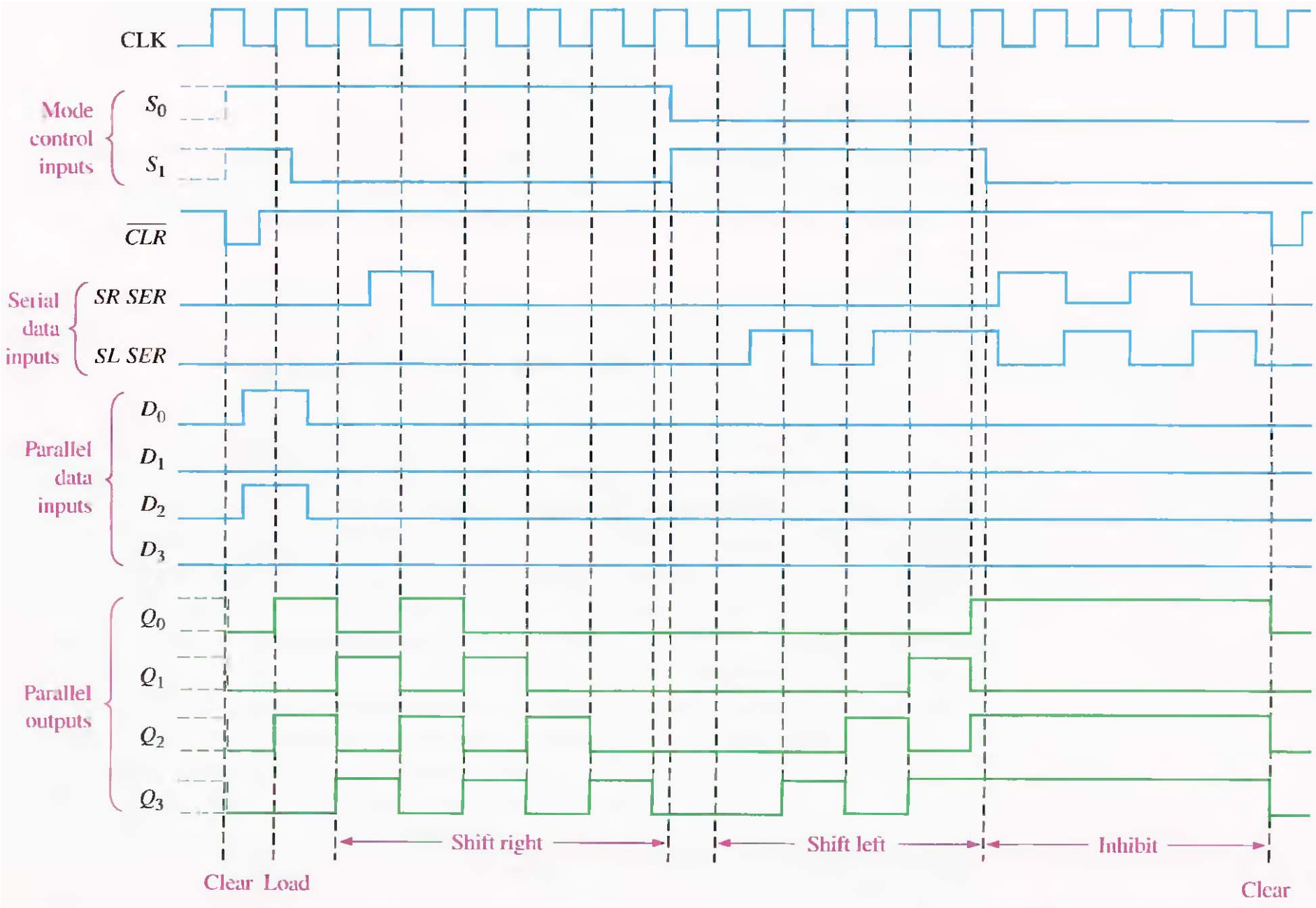


FIGURE 9–21

The 74HC194 4-bit bidirectional universal shift register.


CLK
Mode control inputs
S_0
S_1
$\overline{CLR}$
Serial data inputs
SR SER
SL SER
Parallel data inputs
D_0
D_1
D_2
D_3
Parallel outputs
Q_0
Q_1
Q_2
Q_3
Shift right
Shift left
Inhibit
Clear
Load
Clear

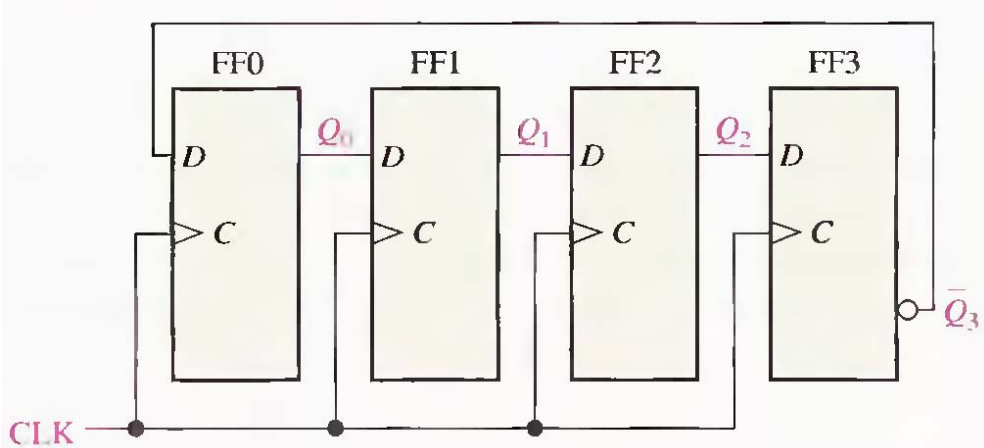
Parallel loading, which is synchronous with a positive transition of the clock, is accomplished by applying the four bits of data to the parallel inputs and a HIGH to the S_0 and S_1 inputs. Shift right is accomplished synchronously with the positive edge of the clock when S_0 is HIGH and S_1 is LOW. Serial data in this mode are entered at the shift-right serial input (*SR SER*). When S_0 is LOW and S_1 is HIGH, data bits shift left synchronously with the clock, and new data are entered at the shift-left serial input (*SL SER*). Input *SR SER* goes into the Q_0 stage, and *SL SER* goes into the Q_3 stage.

SECTION 9-6 REVIEW

1. Assume that the 4-bit bidirectional shift register in Figure 9–19 has the following contents: $Q_0 = 1$, $Q_1 = 1$, $Q_2 = 0$, and $Q_3 = 0$. There is a 1 on the serial data-input line. If $RIGHT/\overline{LEFT}$ is HIGH for three clock pulses and LOW for two more clock pulses, what are the contents after the fifth clock pulse?

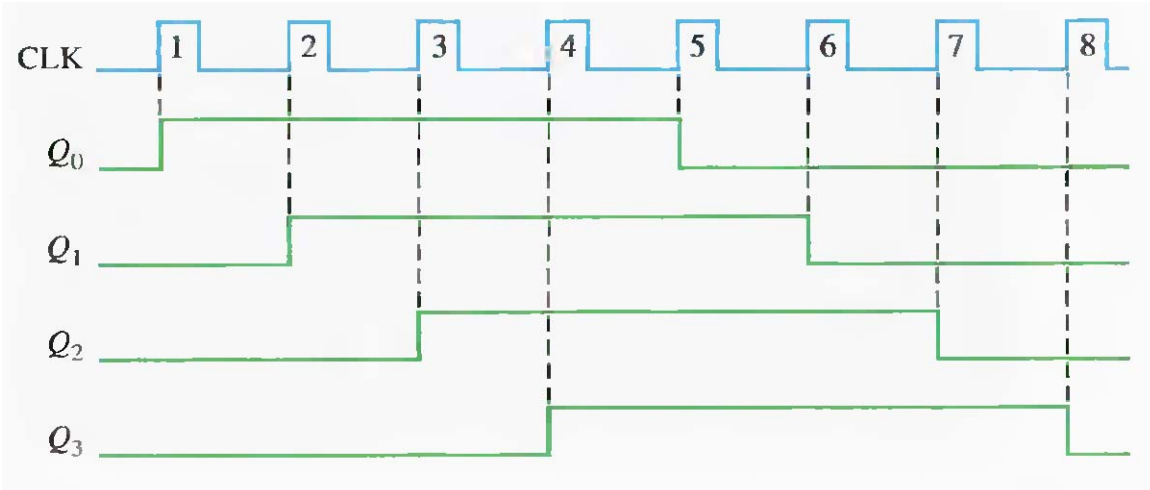
9–7 SHIFT REGISTER COUNTERS

The Johnson Counter

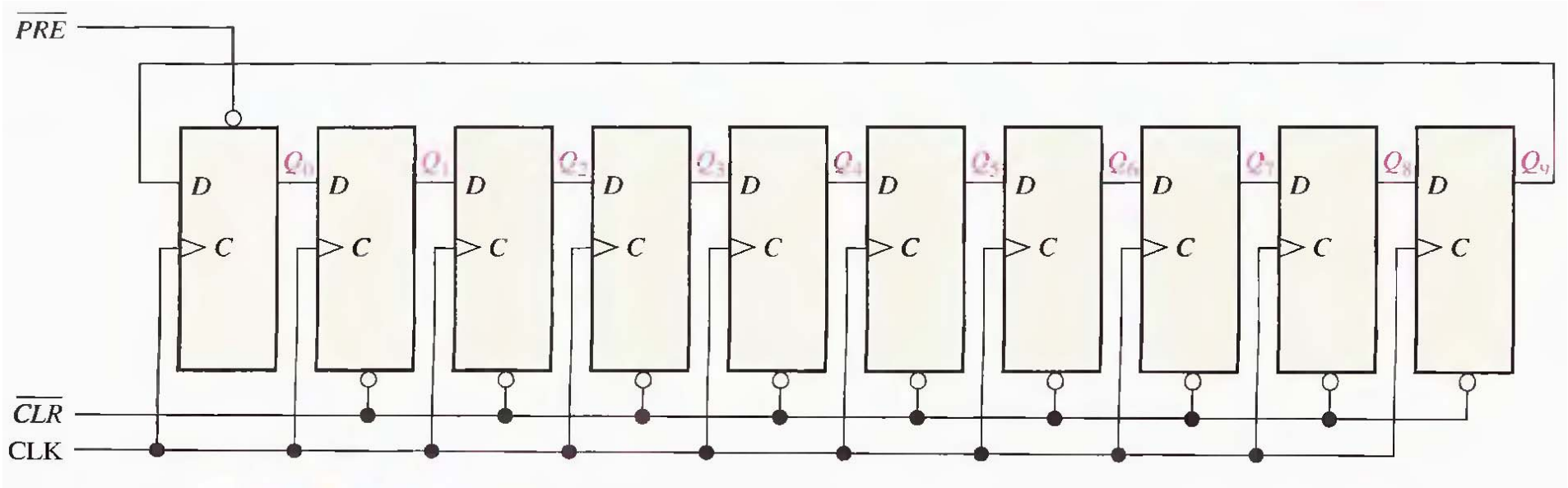


(a) Four-bit Johnson counter

CLOCK PULSE	Q_0	Q_1	Q_2	Q_3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1



The Ring Counter

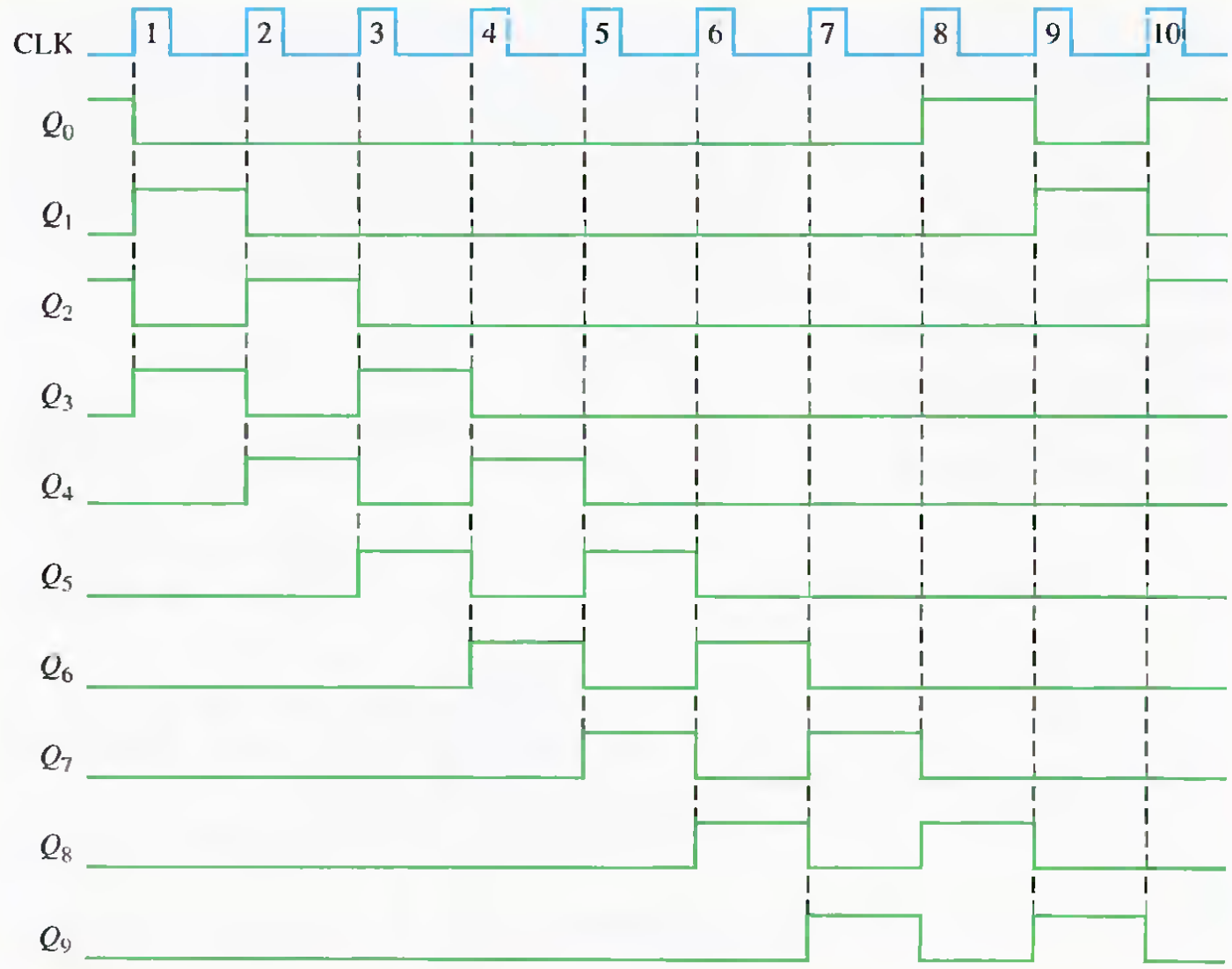


CLOCK PULSE	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9
0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	1

EXAMPLE 9–5

If a 10-bit ring counter similar to Figure 9–26 has the initial state 1010000000, determine the waveform for each of the Q outputs.

See Figure 9–27.

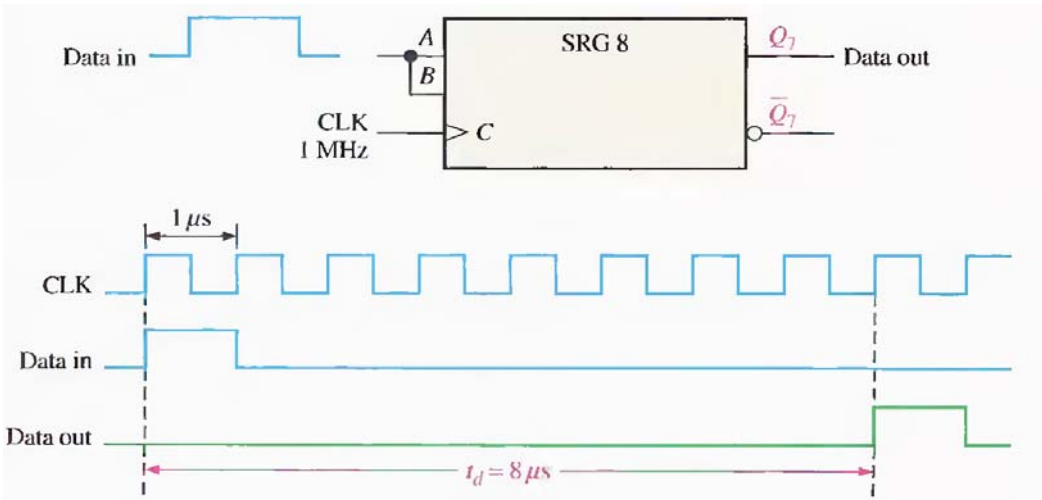


SECTION 9-7 REVIEW

1. How many states are there in an 8-bit Johnson counter sequence?
2. Write the sequence of states for a 3-bit Johnson counter starting with 000.

9–8 SHIFT REGISTER APPLICATIONS

Time Delay



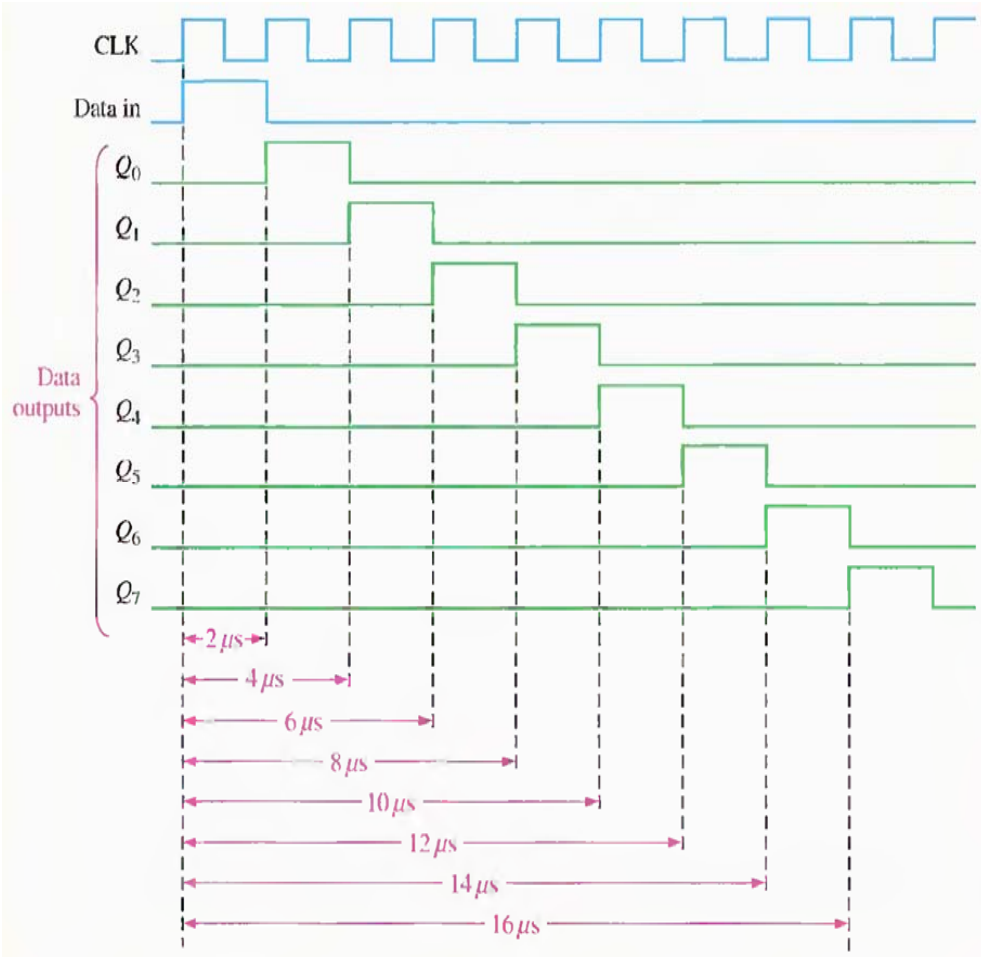
COMPUTER NOTE

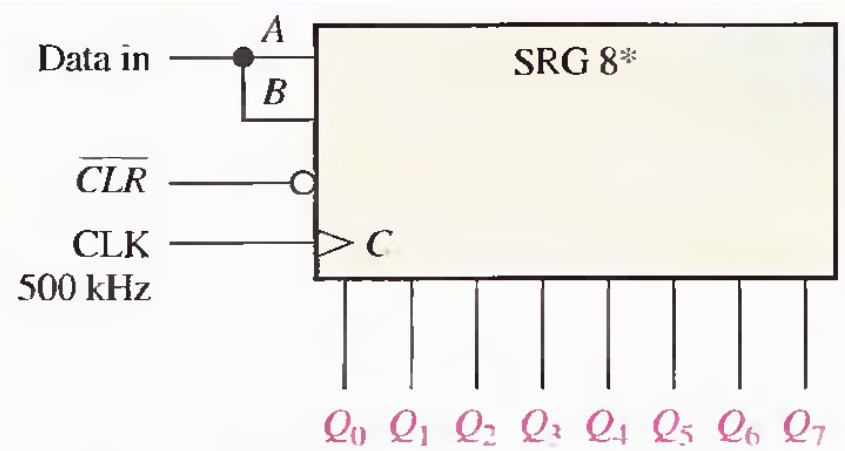
The general-purpose registers in the Pentium are all 32-bit registers that can be used for temporary data storage as well as specific uses. Four of these registers are as follows. The *accumulator* (EAX) is used mainly for temporary storage of data and instruction operands. The *base register* (EBX) is used to store a value temporarily. The *count register* (ECX) is mainly used to determine the number of repetitions in certain loop, string, shift, or rotate operations. The *data register* (EDX), is normally used for the temporary storage of data.

EXAMPLE 9–6

Determine the amount of time delay between the serial input and each output in Figure 9–29. Show a timing diagram to illustrate.

The clock period is 2 μs. Thus, the time delay can be increased or decreased in 2 μs increments from a minimum of 2 μs to a maximum of 16 μs, as illustrated in Figure 9–30.



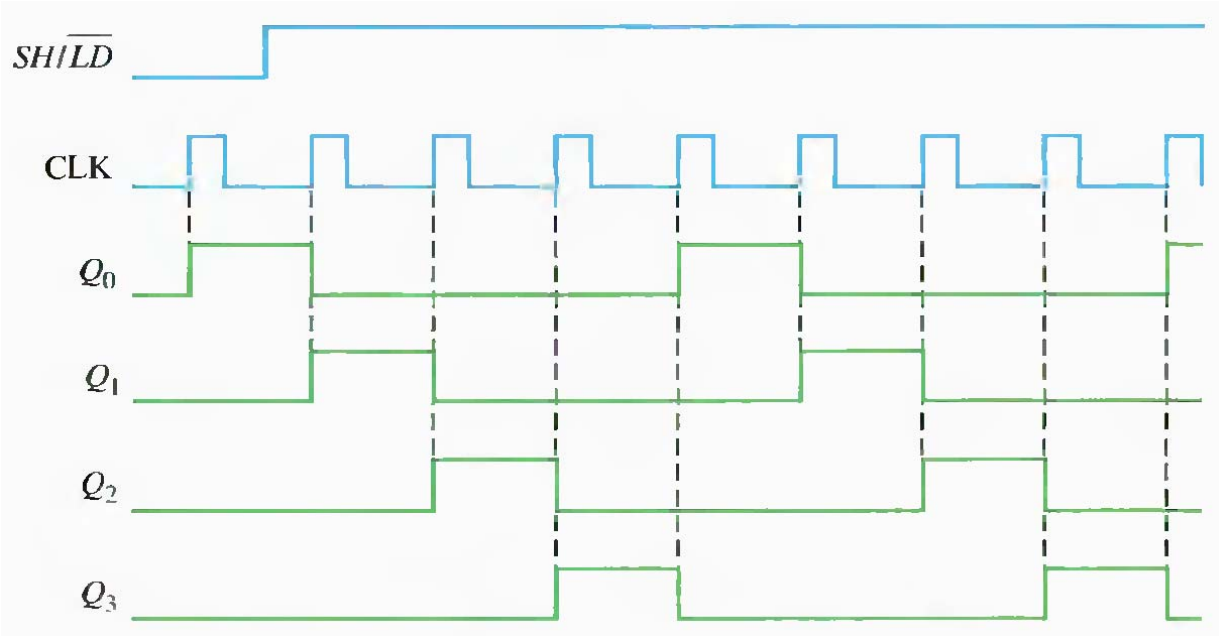


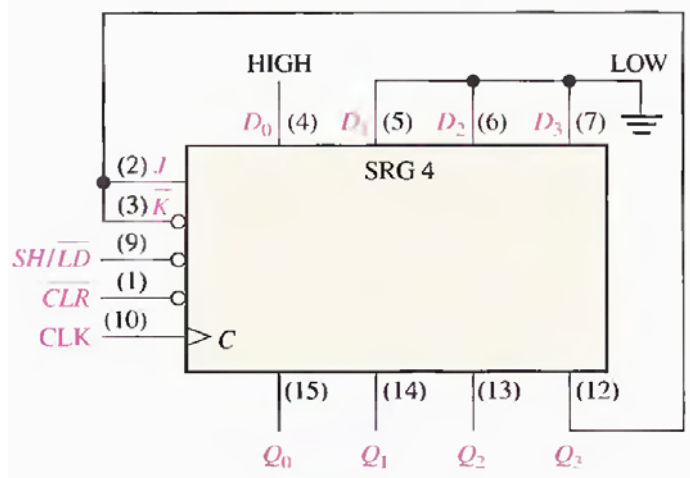
* Data shifts from Q_0 toward Q_7.

A RING COUNTER USING THE 74HC195 SHIFT REGISTER

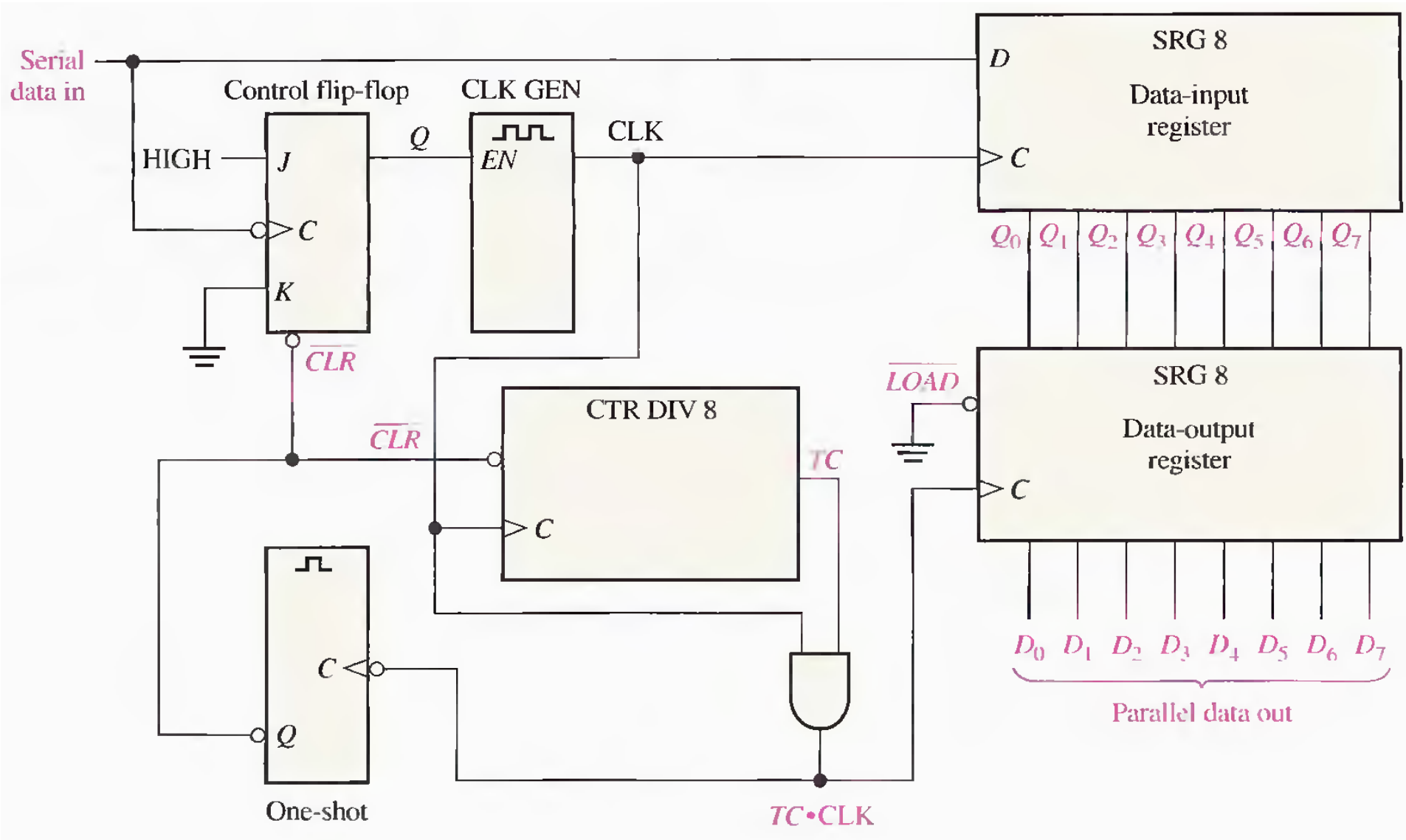
If the output is connected back to the serial input, a shift register can be used as a ring counter. Figure 9–31 illustrates this application with a 74HC195 4-bit shift register.

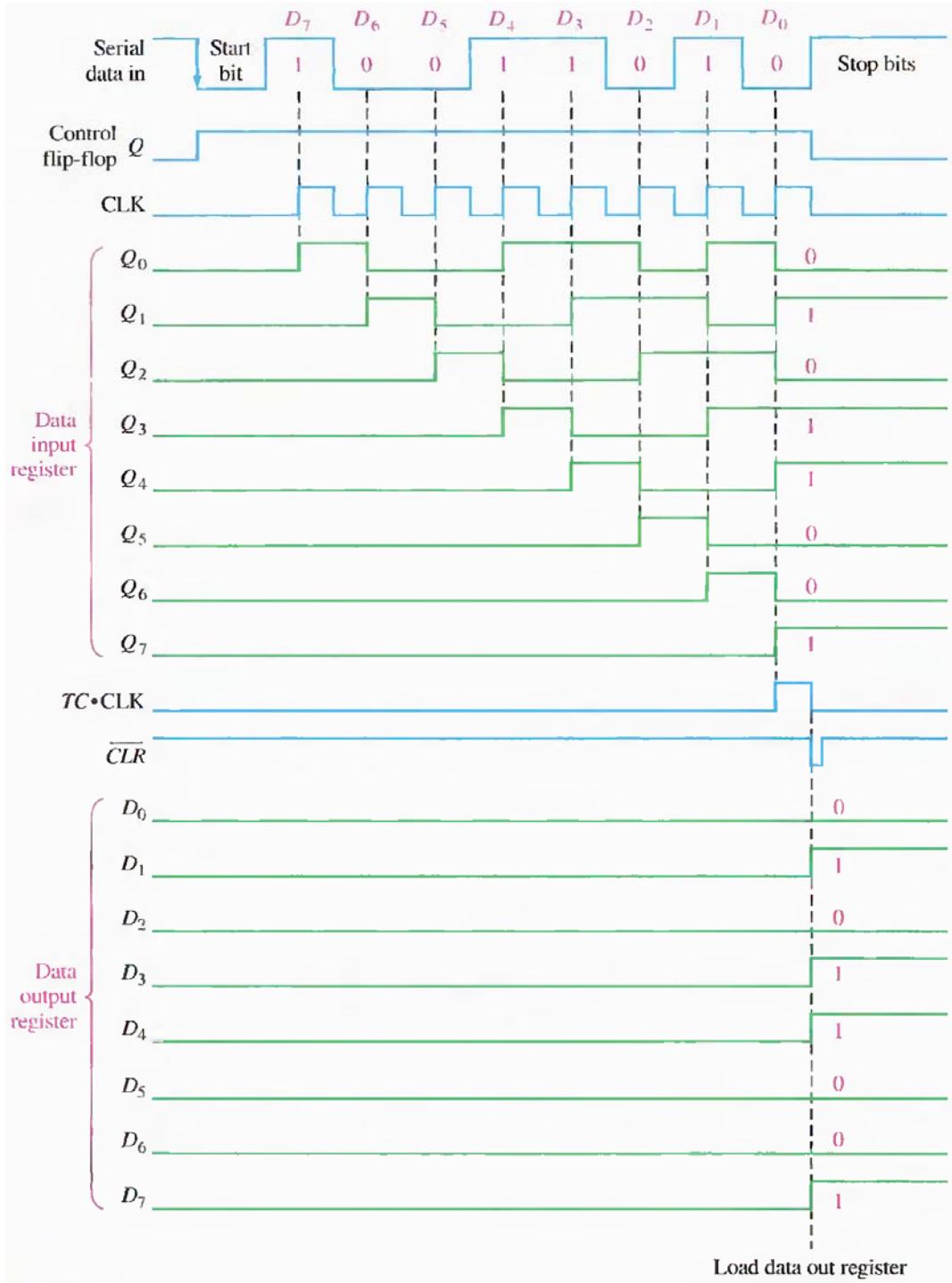
Initially, a bit pattern of 1000 (or any other pattern) can be synchronously preset into the counter by applying the bit pattern to the parallel data inputs, taking the $SH/\overline{LD}$ input LOW, and applying a clock pulse. After this initialization, the 1 continues to circulate through the ring counter, as the timing diagram in Figure 9–32 shows.





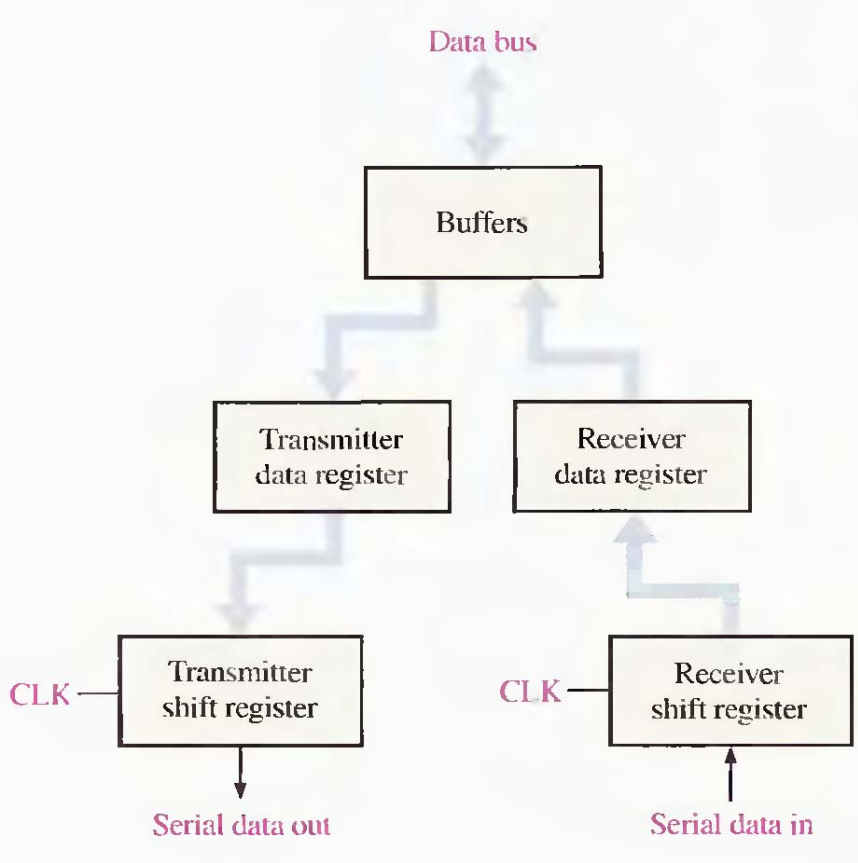
Serial-to-Parallel Data Converter

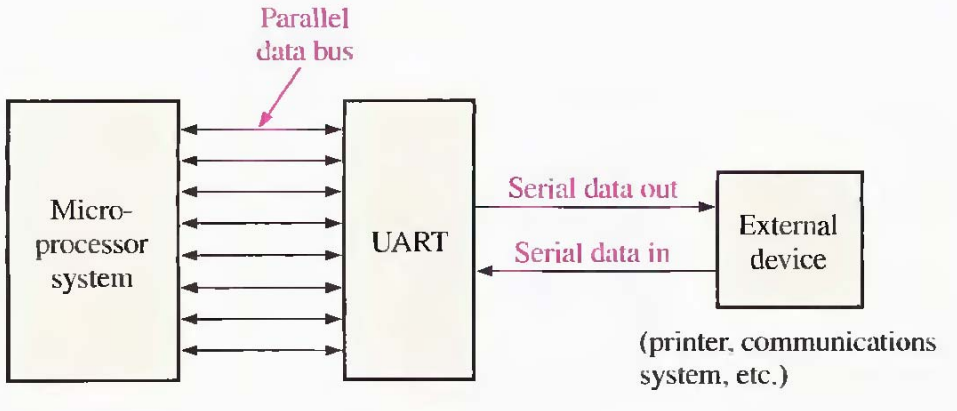



D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
Serial data in
Start bit
1 0 0 1 1 0 1 0
Stop bits
Control flip-flop Q
CLK
Data input register
Q_0 0
Q_1 1
Q_2 0
Q_3 1
Q_4 1
Q_5 0
Q_6 0
Q_7 1
$TC \bullet CLK$
$\overline{CLR}$
Data output register
D_0 0
D_1 1
D_2 0
D_3 1
D_4 1
D_5 0
D_6 0
D_7 1
Load data out register

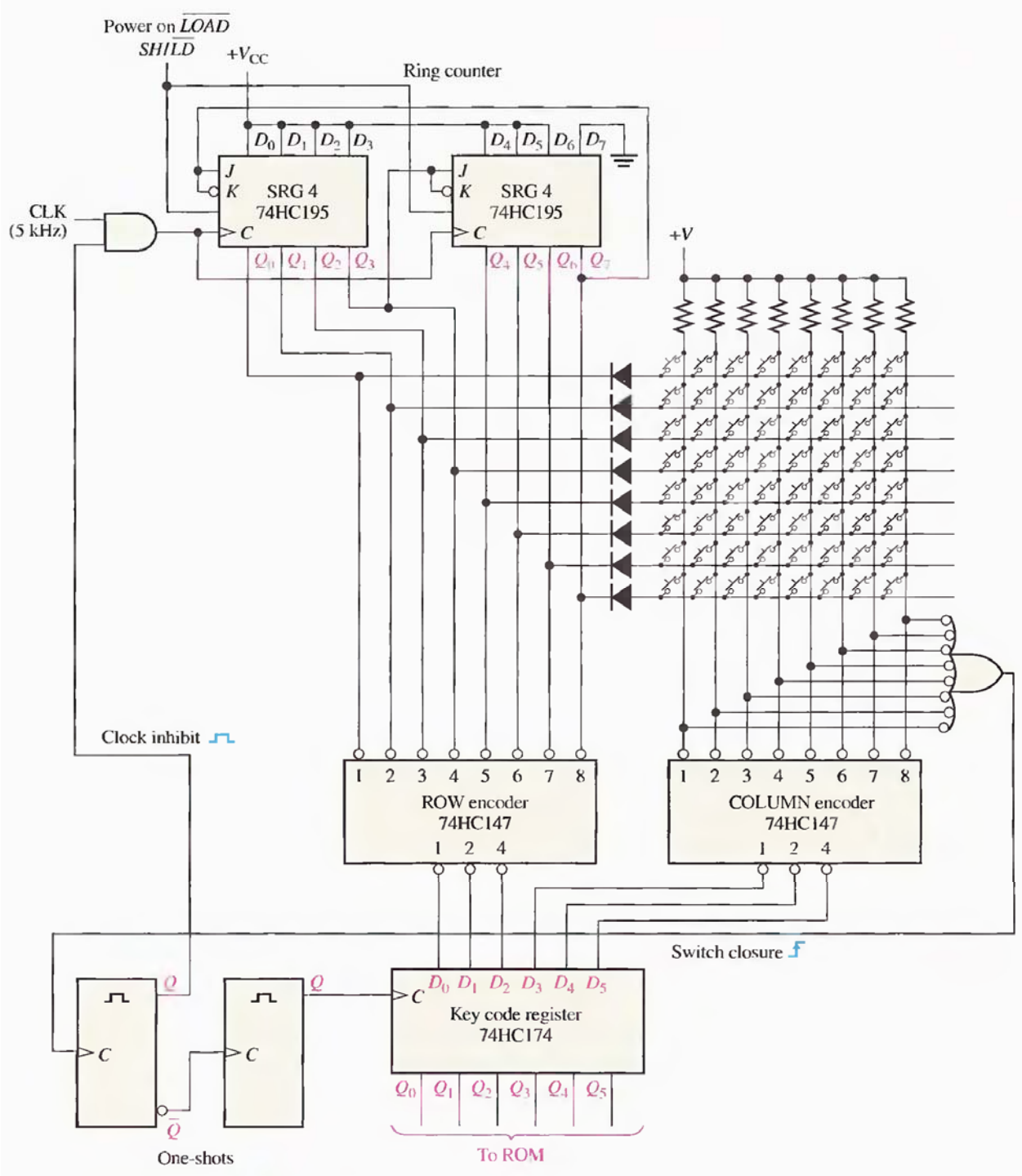
Universal Asynchronous Receiver Transmitter (UART)

As mentioned, computers and microprocessor-based systems often send and receive data in a parallel format. Frequently, these systems must communicate with external devices that send and/or receive serial data. An interfacing device used to accomplish these conversions is the UART (Universal Asynchronous Receiver Transmitter). Figure 9–36 illustrates the UART in a general microprocessor-based system application.





Keyboard Encoder



SECTION 9–8 REVIEW

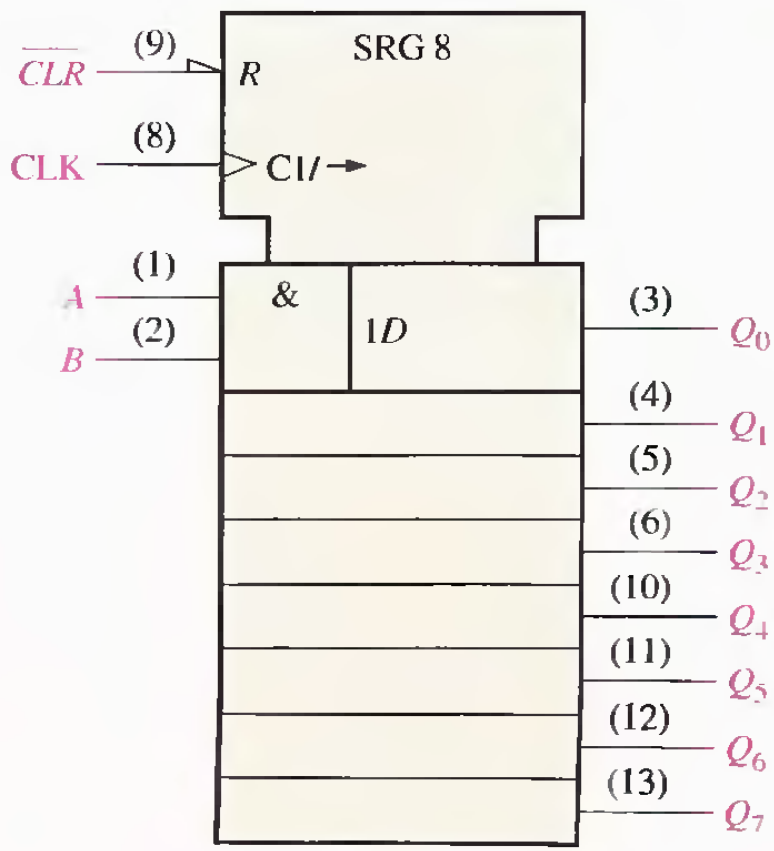
1. In the keyboard encoder, how many times per second does the ring counter scan the keyboard?
2. What is the 6-bit ROW/COLUMN code (key code) for the top row and the left-most column in the keyboard encoder?
3. What is the purpose of the diodes in the keyboard encoder? What is the purpose of the resistors?

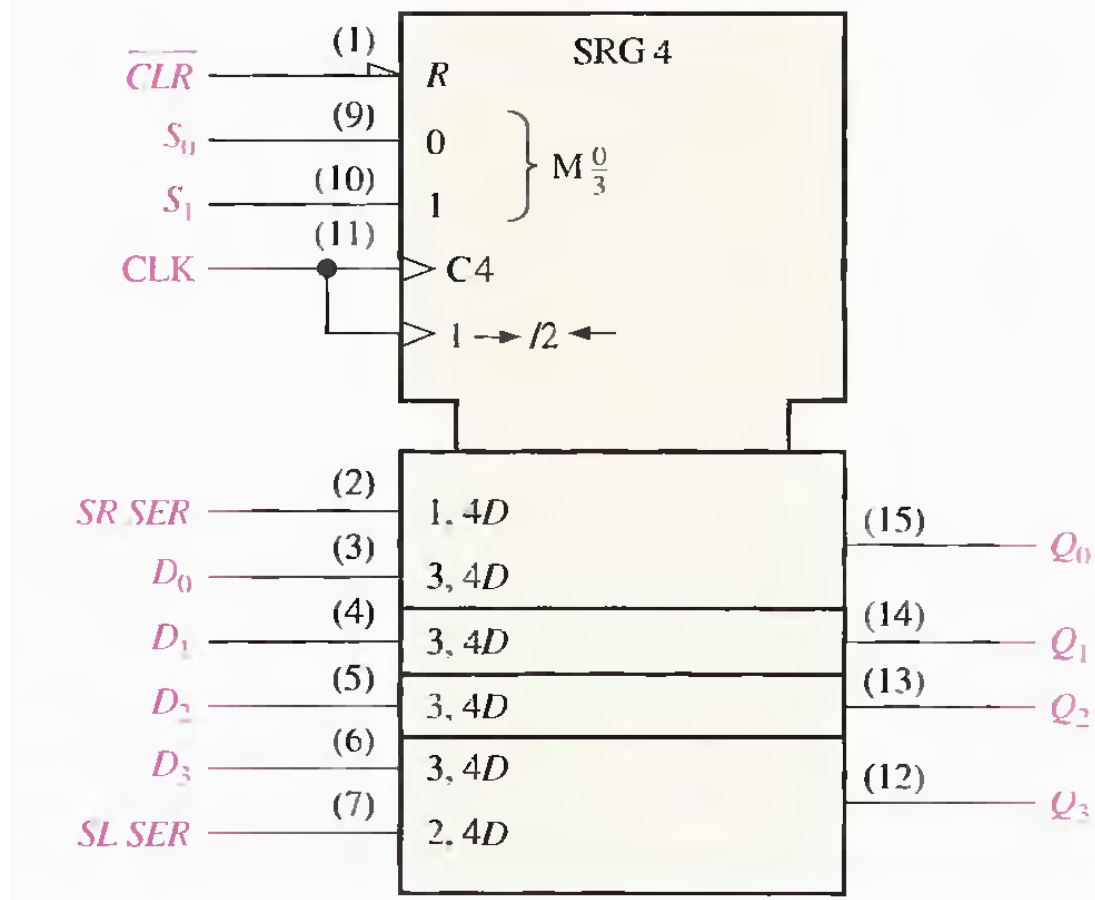
9–9 LOGIC SYMBOLS WITH DEPENDENCY NOTATION

Do nothing:	$S_0 = 0, S_1 = 0$	(mode 0)
Shift right:	$S_0 = 1, S_1 = 0$	(mode 1, as in 1, 4*D*)
Shift left:	$S_0 = 0, S_1 = 1$	(mode 2, as in 2, 4*D*)
Parallel load:	$S_0 = 1, S_1 = 1$	(mode 3, as in 3, 4*D*)

74HC194.

74HC164.





SECTION 9-9 REVIEW

1. In Figure 9–40, are there any inputs that are dependent on the mode inputs being in the 0 state?
2. Is the parallel load synchronous with the clock?

9–10 TROUBLESHOOTING

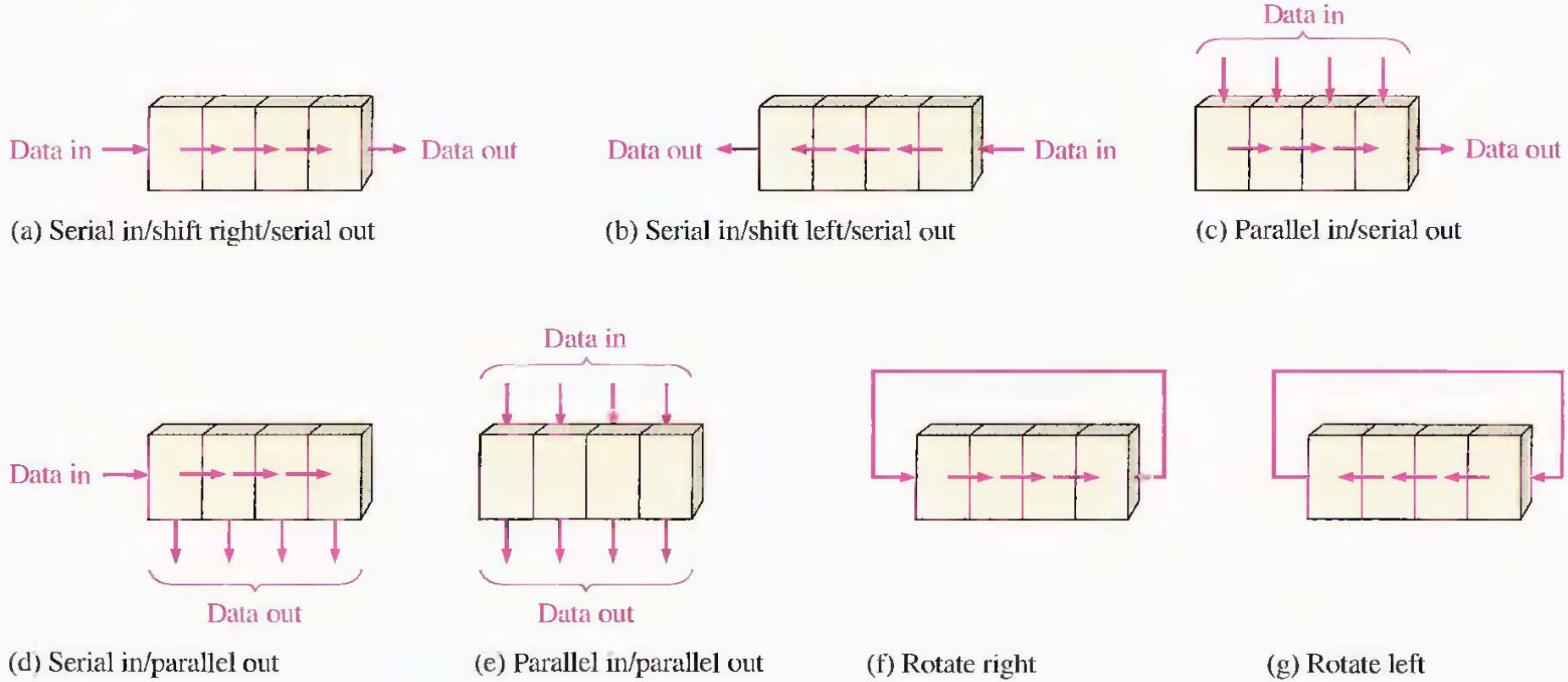
HANDS ON TIP

When measuring digital signals with an oscilloscope, you should always use dc coupling, rather than ac coupling. The reason that ac coupling is not best for viewing digital signals is that the 0 V level of the signal will appear at the *average* level of the signal, not at true ground or 0 V level. It is much easier to find a "floating" ground or incorrect logic level with dc coupling. If you suspect an open ground in a digital circuit, increase the sensitivity of the scope to the maximum possible. A good ground will never appear to have noise under this condition, but an open will likely show some noise, which appears as a random fluctuation in the 0 V level.

DIGITAL SYSTEM APPLICATION

SUMMARY

- The basic types of data movement in shift registers are illustrated in Figure 9–46.



(a) Serial in/shift right/serial out (b) Serial in/shift left/serial out (c) Parallel in/serial out

(d) Serial in/parallel out (e) Parallel in/parallel out (f) Rotate right (g) Rotate left

- Shift register counters are shift registers with feedback that exhibit special sequences. Examples are the Johnson counter and the ring counter.
- The Johnson counter has $2n$ states in its sequence, where n is the number of stages.
- The ring counter has n states in its sequence.

KEY TERMS

Bidirectional Having two directions. In a bidirectional shift register, the stored data can be shifted right or left.

Load To enter data into a shift register.

Register One or more flip-flops used to store and shift data.

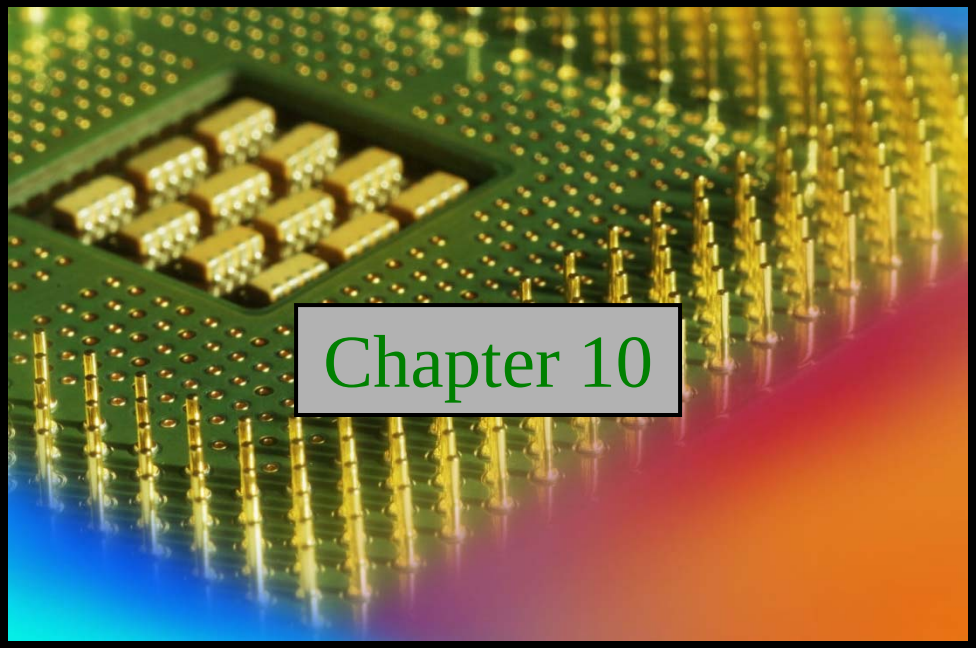
Shift To move binary data from stage to stage within a shift register or other storage device or to move binary data into or out of the device.

Stage One storage element in a register.

Digital Fundamentals

Tenth Edition

Floyd



© 2008 Pearson Education

BASICS OF SEMICONDUCTOR MEMORY

Memory is the portion of a system for storing binary data in large quantities.

Semiconductor memories consist of arrays of elements that are generally either latches or capacitors.

Bits, Bytes, Nibbles and Words

A complete unit of information is called a word and generally consists of one or more bytes. Some memories store data in 9-bit groups; a 9-bit group consists of a byte plus a parity bit.

In programing word is defined as two byte.

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

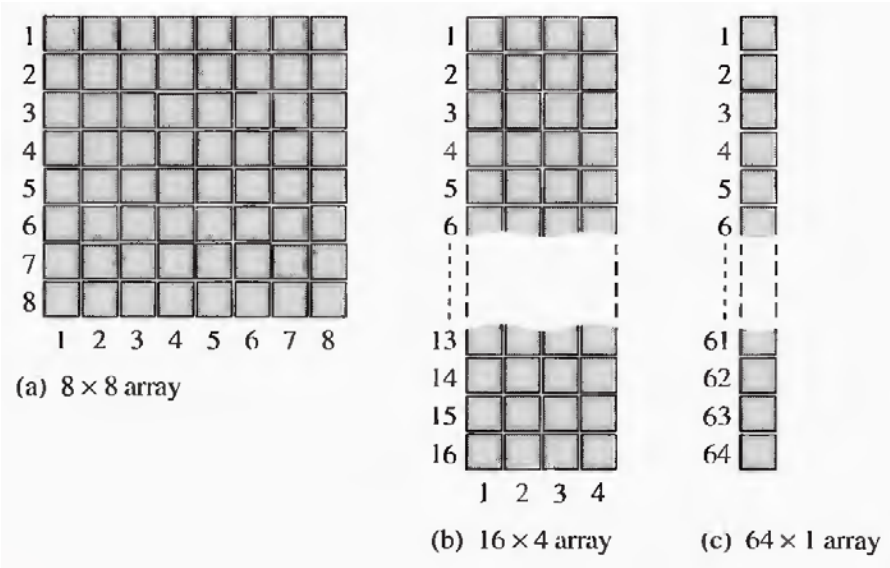
Summary

The Basic Semiconductor Memory Array

Each storage element in a memory can retain either a 1 or a 0 and is called a cell. Memories are made up of arrays of cells, (i.e. 64 cell array)

Each block in the memory array represents one storage cell. and its location can be identified by specifying a row and a column.

The 64-cell array can be organized in several ways based on units of data.



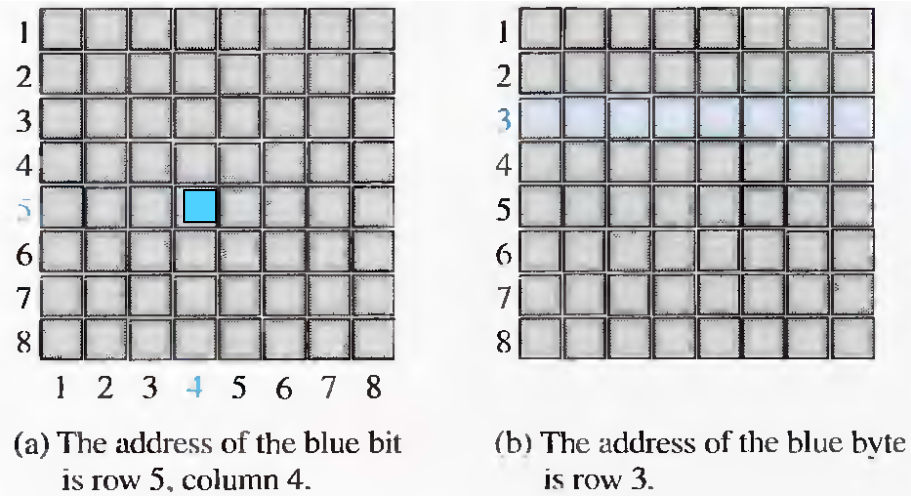
© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

Summary

Memory Address and Capacity

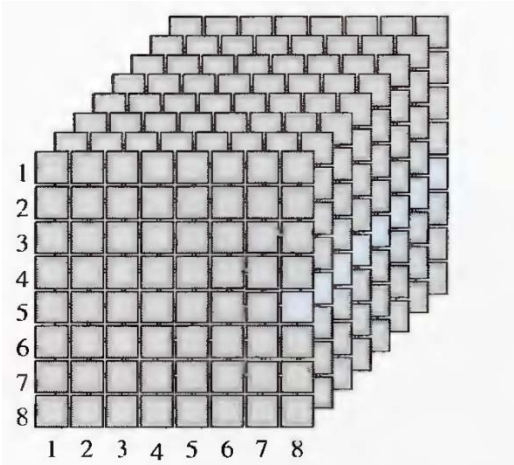
The location of a unit of data in a memory array is called its address.

The capacity of a memory is the total number of data units that can be stored. (i.e. 64 bits, 8 bytes, which is also 64 bits. 64 bytes. 256 MB)



(a) The address of the blue bit is row 5, column 4.

(b) The address of the blue byte is row 3.



The address of the blue byte is row 5, column 8.

3D array.

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

Summary

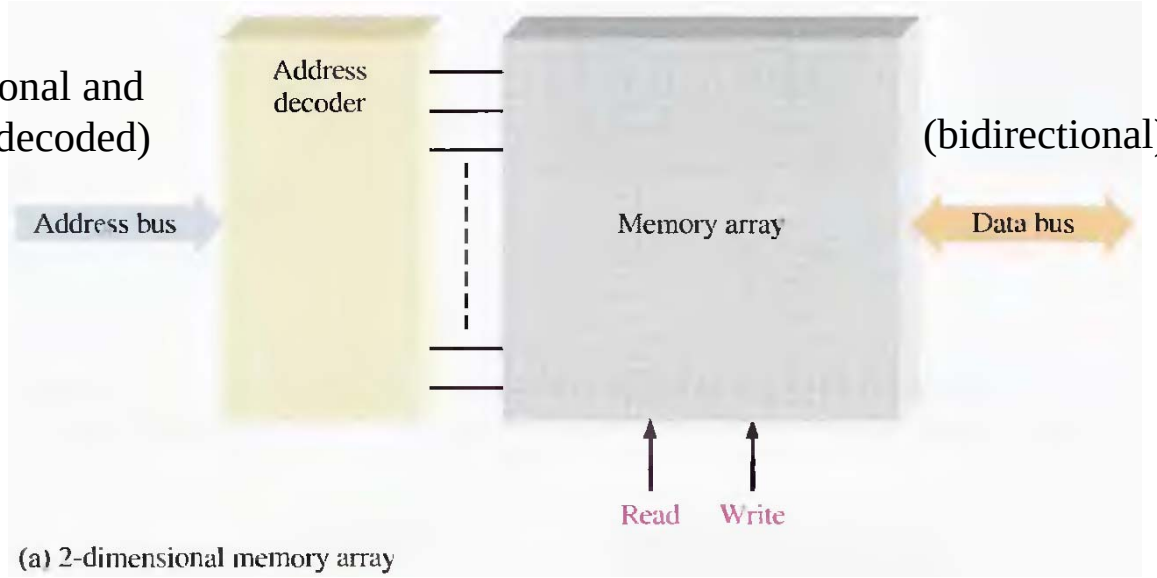
Basic Memory Operations

The write operation puts data into a specified address in the memory, and the read operation copies data out of a specified address in the memory.

The addressing operation, which is part of both the write and the read operations, selects the specified memory address.

(unidirectional and internally decoded)

(bidirectional)



(a) 2-dimensional memory array

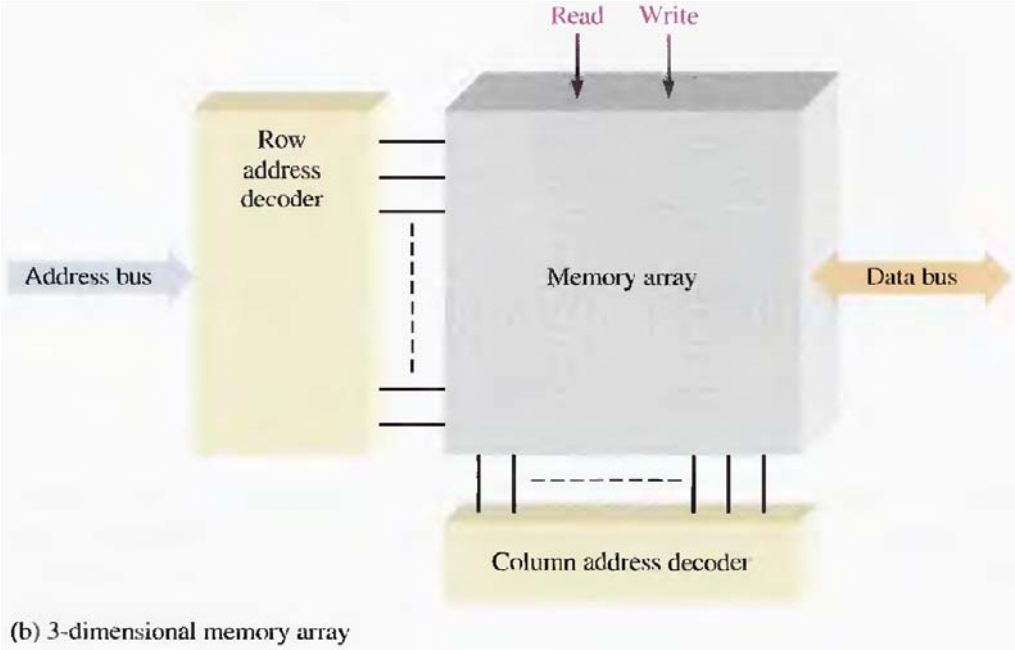
© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

Summary

Basic Memory Operations

In the case of the 3-dimensional memory array there are two decoders, one for the rows and one for the columns.

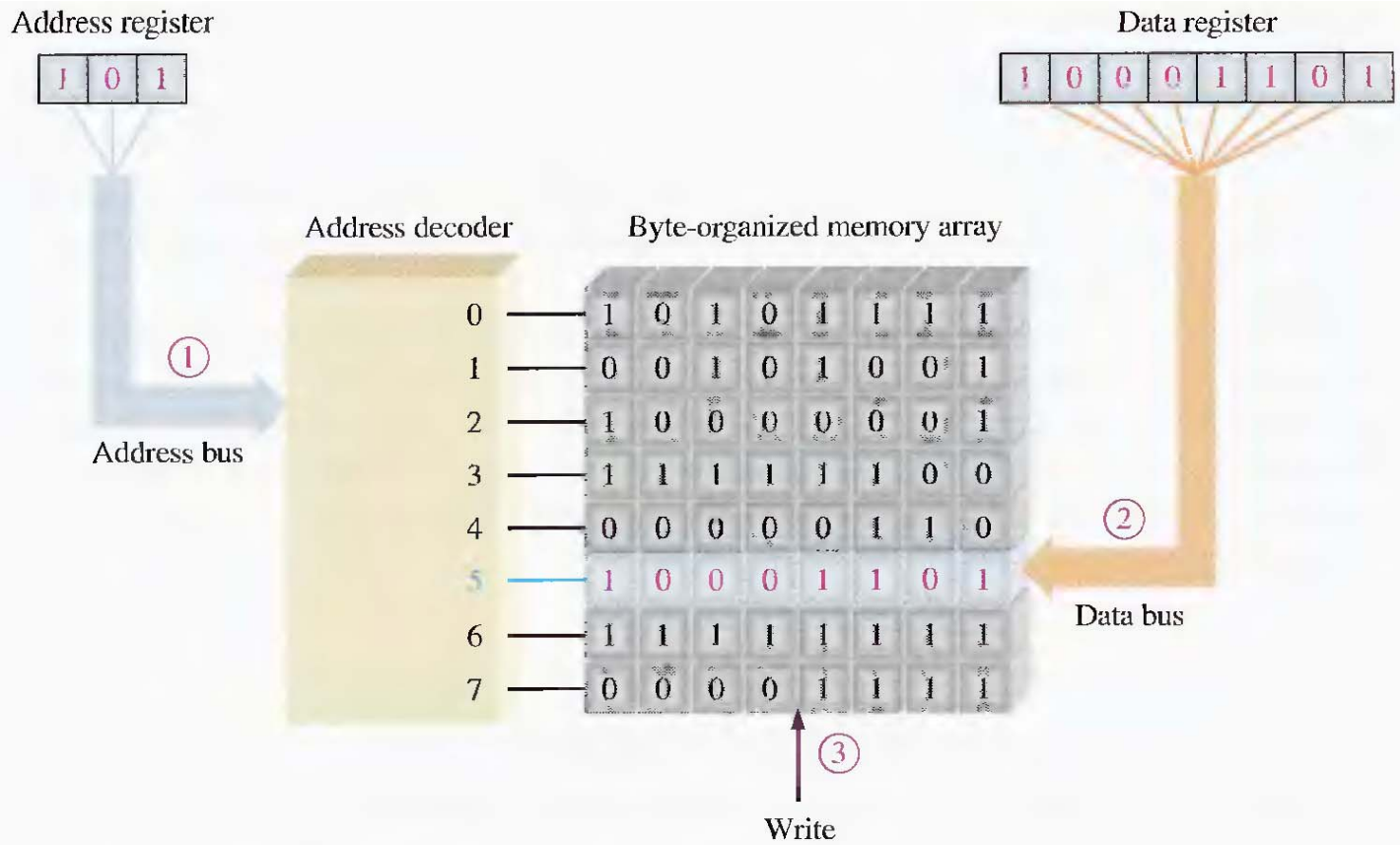
The number of lines in the address bus depends on the capacity of the memory.



(b) 3-dimensional memory array

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

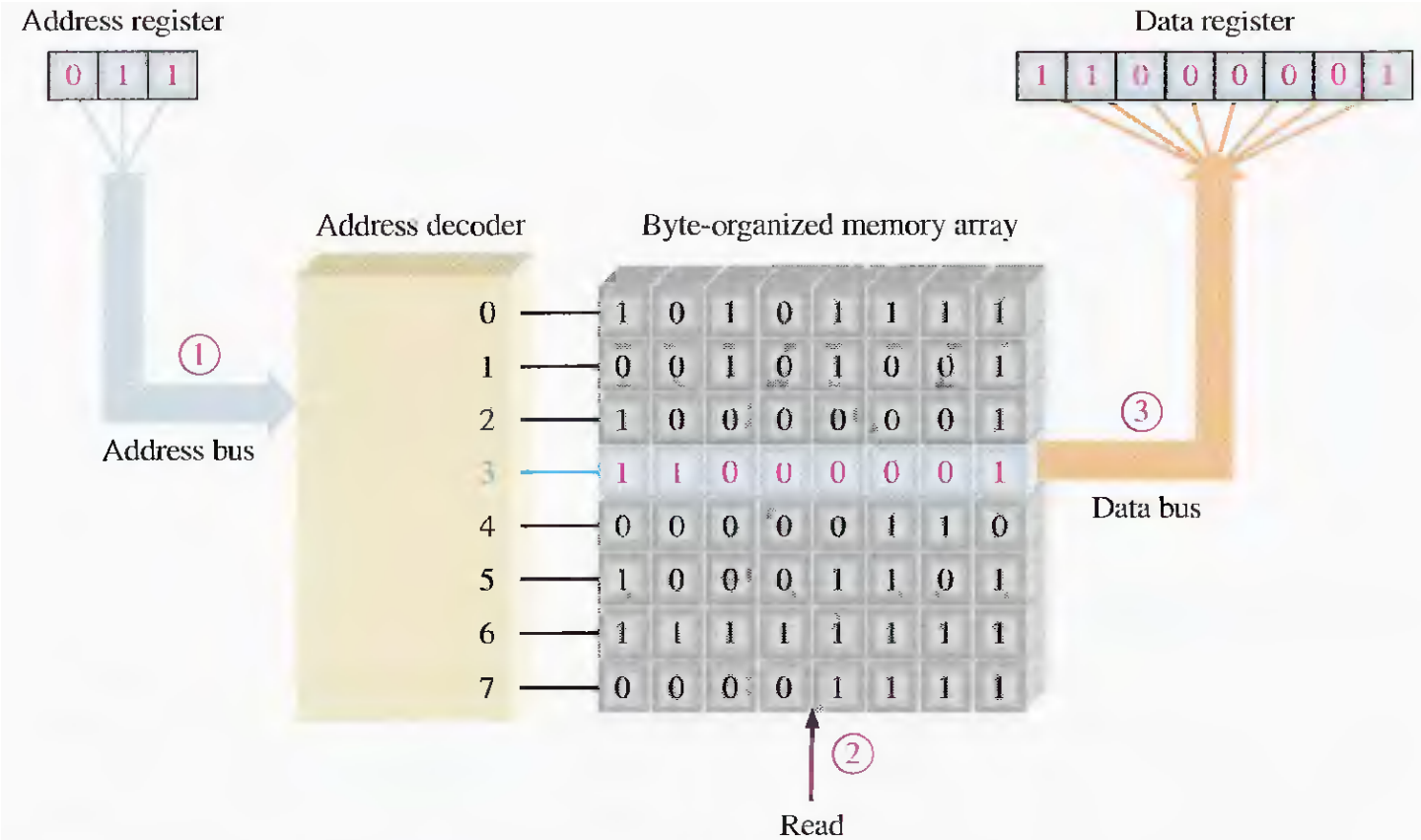
The Write Operation



1. Address code 101 is placed on the address bus and address 5 is selected.
2. Data byte is placed on the data bus.
3. Write command causes the data byte to be stored in address 5, replacing previous data.

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

The Read Operation



1. Address code 011 is placed on the address bus and address 3 is selected.
2. Read command is applied.
3. The contents of address 3 is placed on the data bus and shifted into data register. The contents of address 3 is not erased by the read operation.

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

RAMs and ROMs

The two major categories of semiconductor memories are the RAM and the ROM.

RAM (random-access memory) is a type of memory in which all addresses are accessible in an equal amount of time and can be selected in any order for a read or write operation.

read and write capability

volatile memories.

ROM (read-only memory) is a type of memory in which data are stored permanently or semi permanently. Data can be read from a ROM, but there is no write operation as in the RAM. The ROM, like the RAM, is a random-access memory but the term RAM traditionally means a random-access read/write memory.

Because ROMs retain stored data even if power is turned off, they are nonvolatile memories.

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

The RAM Family

Static RAMs generally use latches as storage elements and can therefore store data indefinitely as long as DC power is applied.

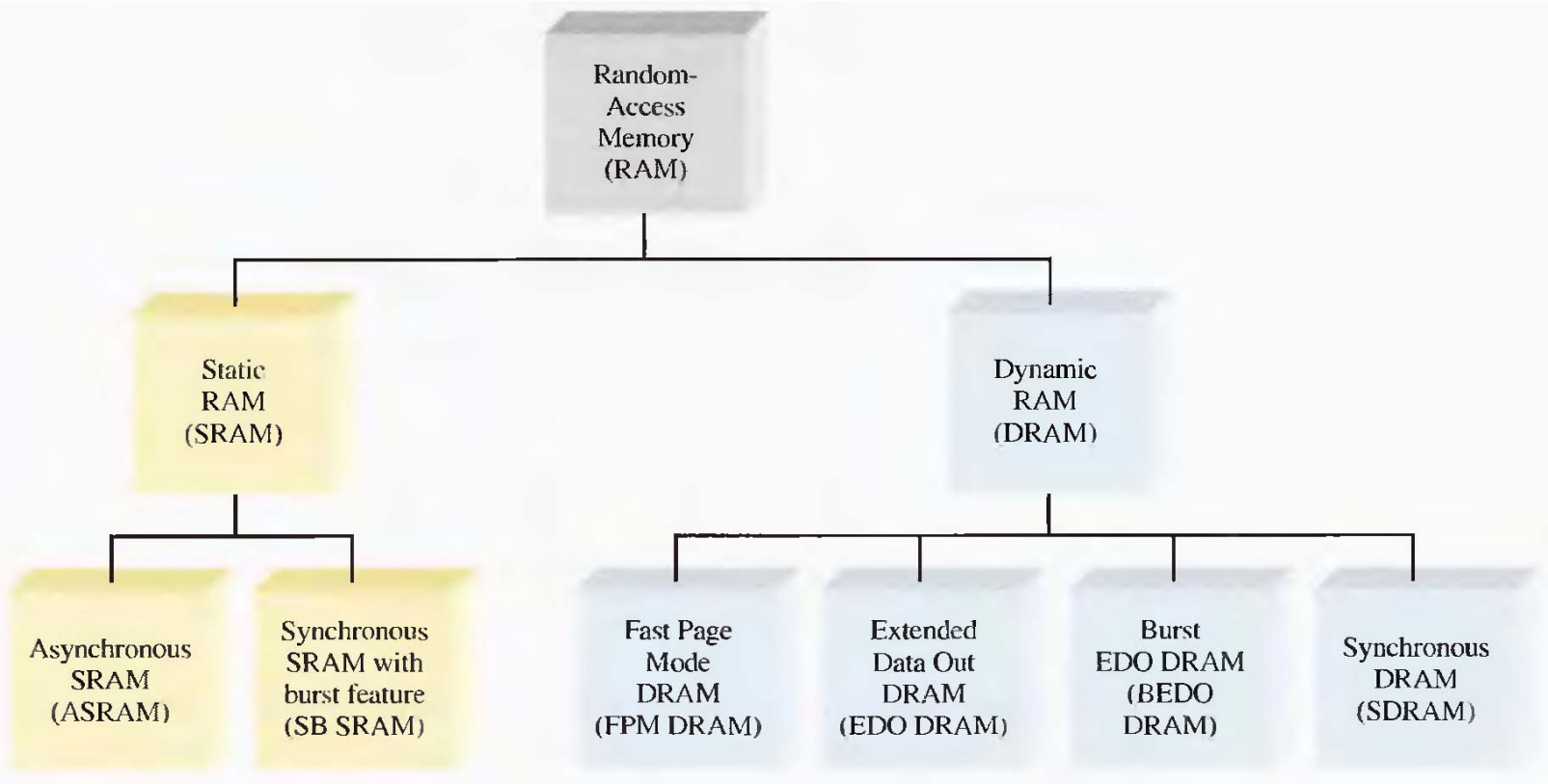
Dynamic RAMs use capacitors as storage elements and cannot retain data very long without the capacitors being recharged by a process called refreshing.

Both SRAMs and DRAMs will lose stored data when dc power is removed and, therefore, are classified as volatile memories.

Data can be read much faster from SRAMs than from DRAMs. However, DRAMs can store much more data than SRAMs for a given physical size and cost

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

The RAM Family



© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

Static RAMs (SRAMs)

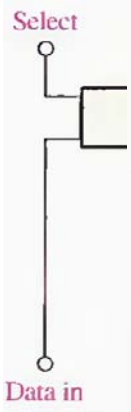
SRAM

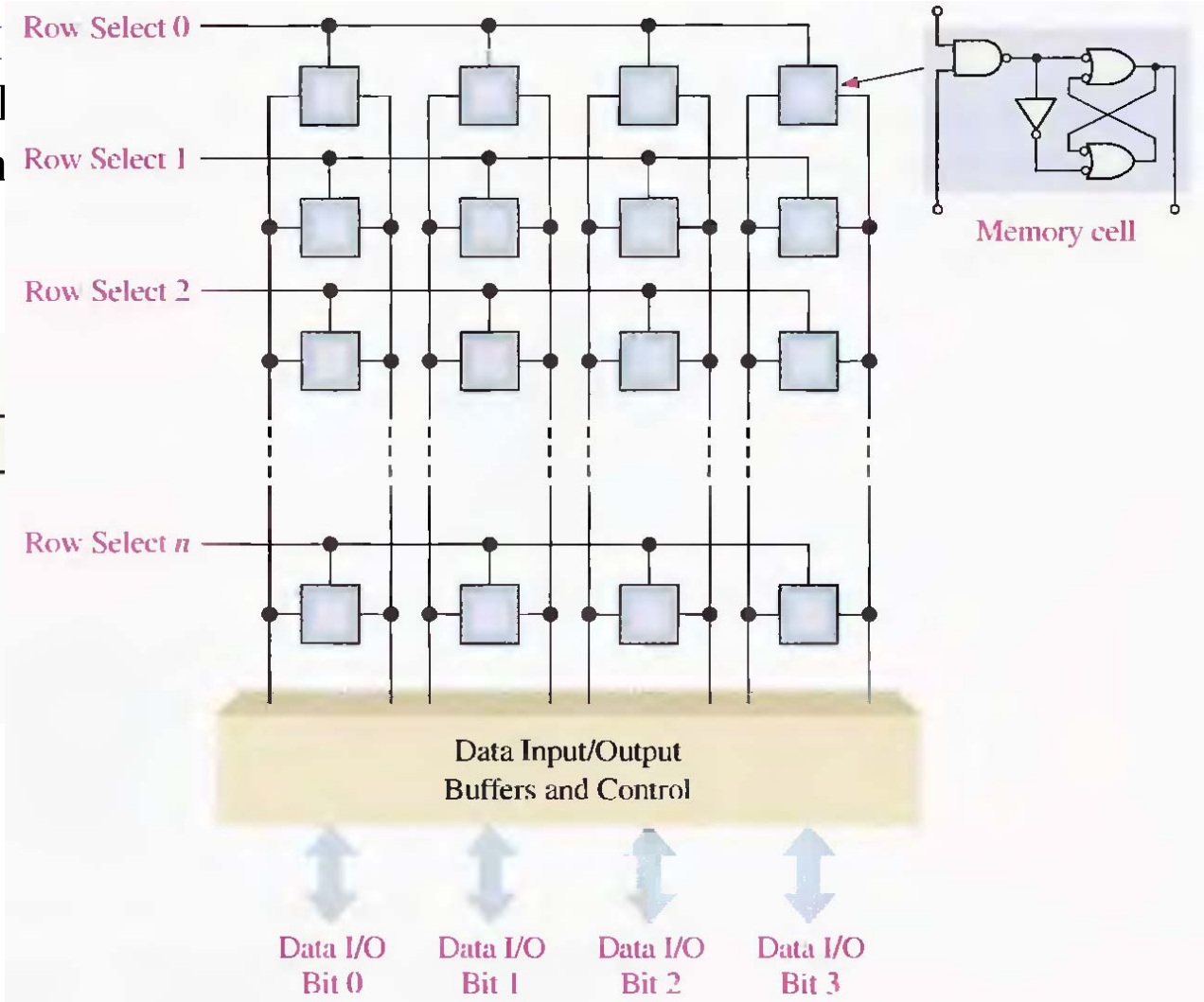
Select l

Data in

n the

: on the



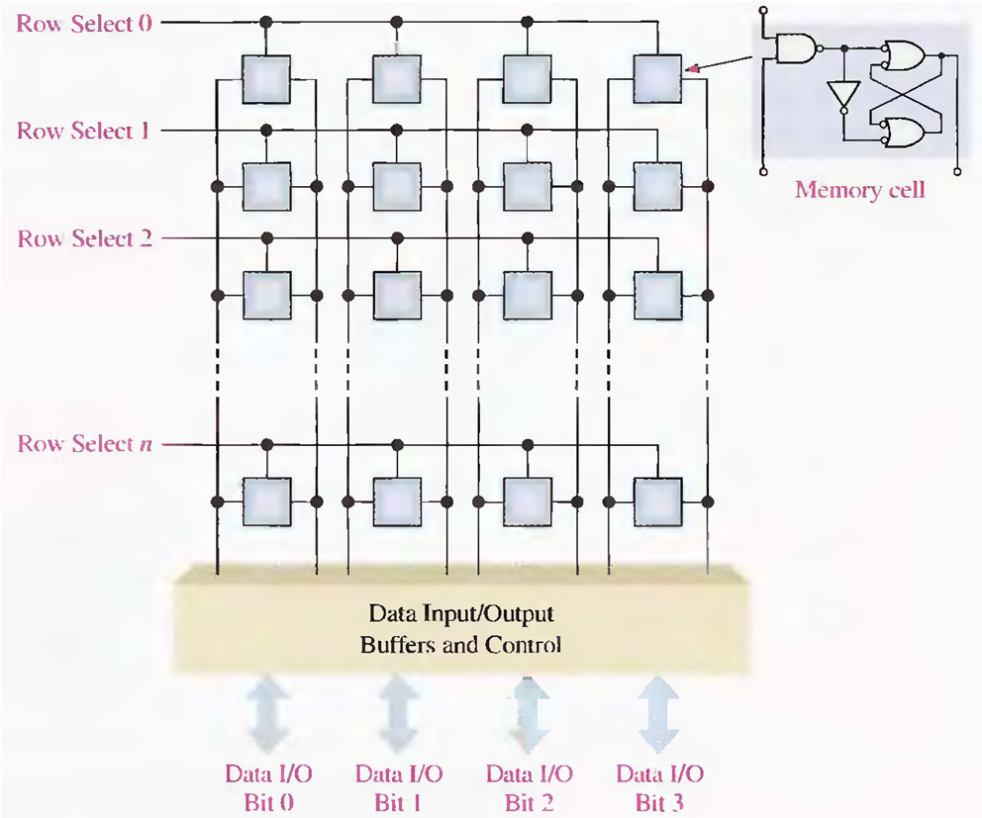


© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

Static RAMs (SRAMs)

To write a data unit: the Row Select line is taken to its active state and four data bits are placed on the Data I/O lines. The Write line is then taken to its active state, which causes each data bit to be stored in a selected cell in the associated column.

To read a data unit, the Read line is taken to its active state, which causes the four data bits stored in the selected row to appear on the Data I/O lines.

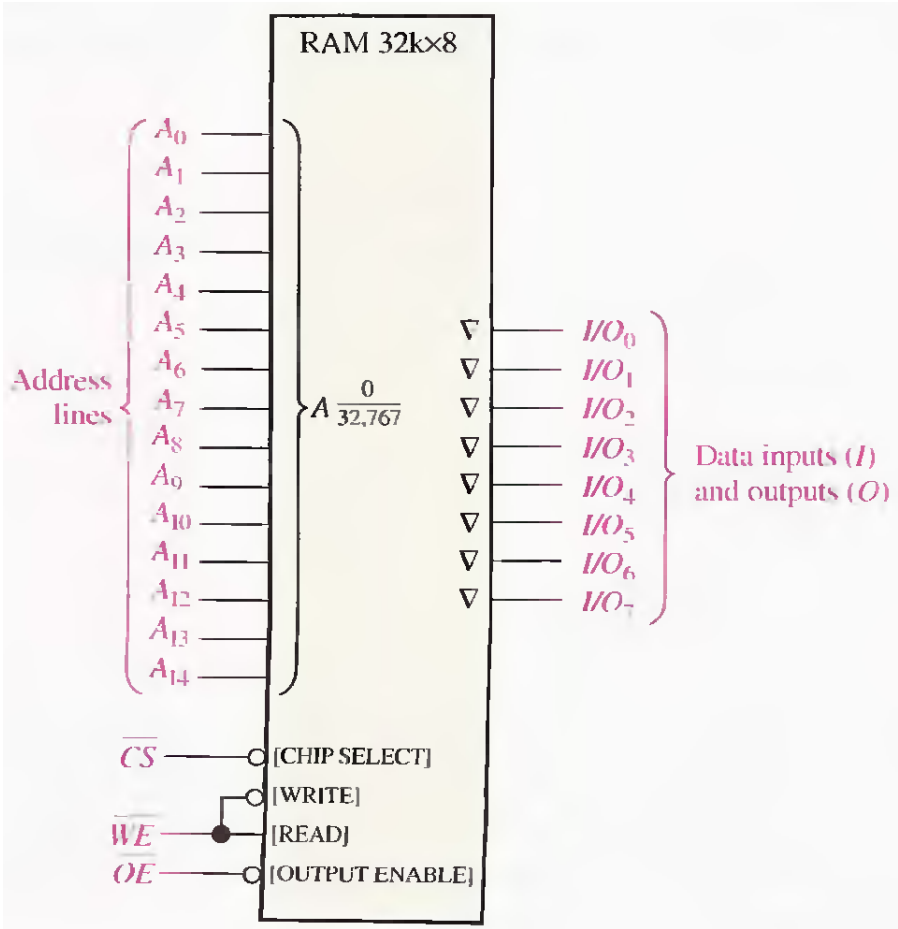


© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

Basic Asynchronous SRAM Organization

An asynchronous SRAM is one in which the operation is not synchronized with a system clock.

32k x 8 bit with tri state output



© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

Tristate Outputs and Buses

Tristate buffers in a memory allow the data lines to act as either input or output lines and connect the memory to the data bus in a computer. These buffers have three output states:

HIGH (1)

LOW (0)

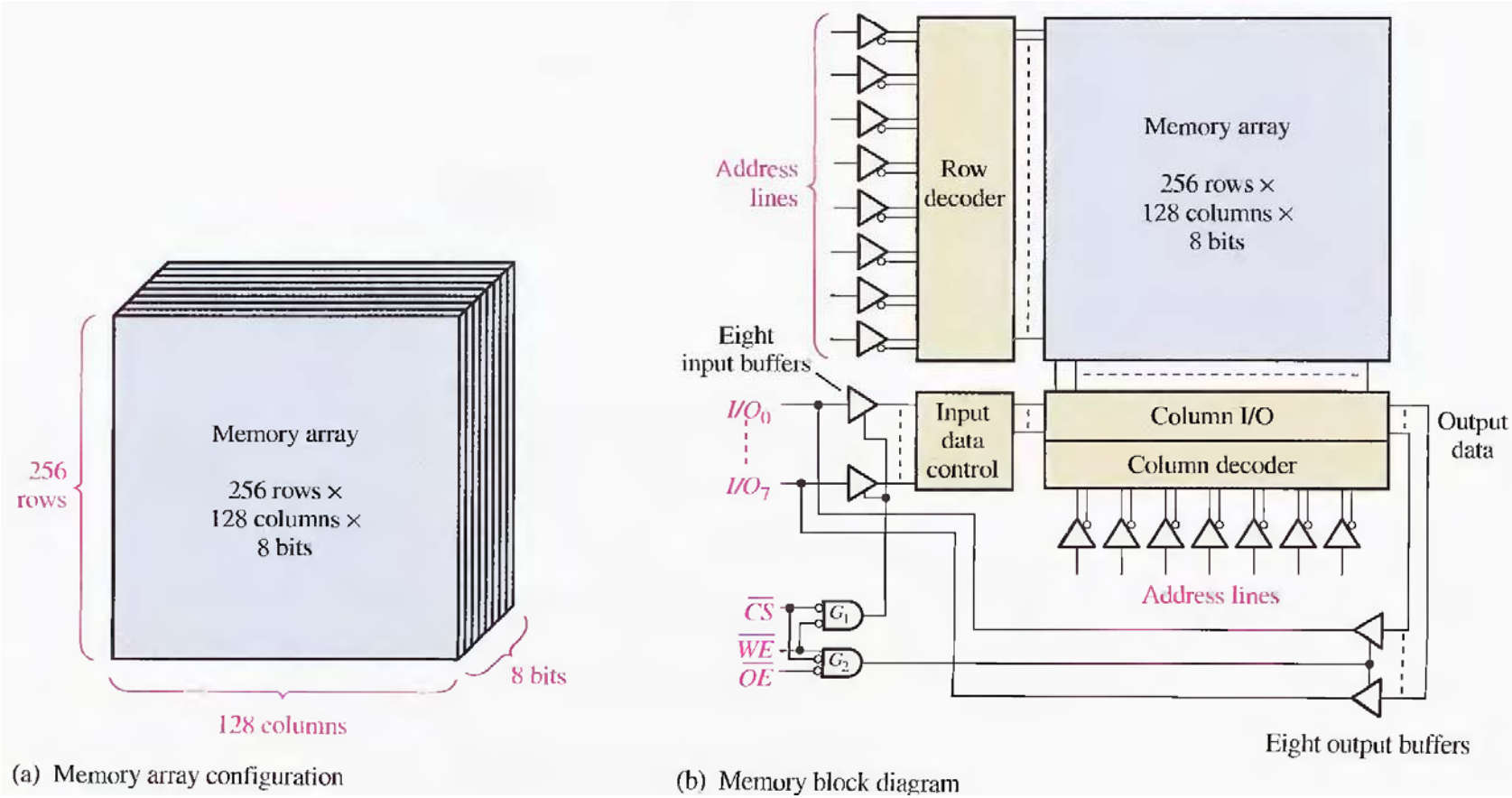
and HIGH-Z (open).

Physically bus is a set of conductive paths that serve to interconnect two or more functional components of a system or several diverse systems.

Electrically. a bus is a collection of specified voltage levels and/or current levels and signals that allow the various devices connected to the bus to communicate and work properly together.

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

Memory Array

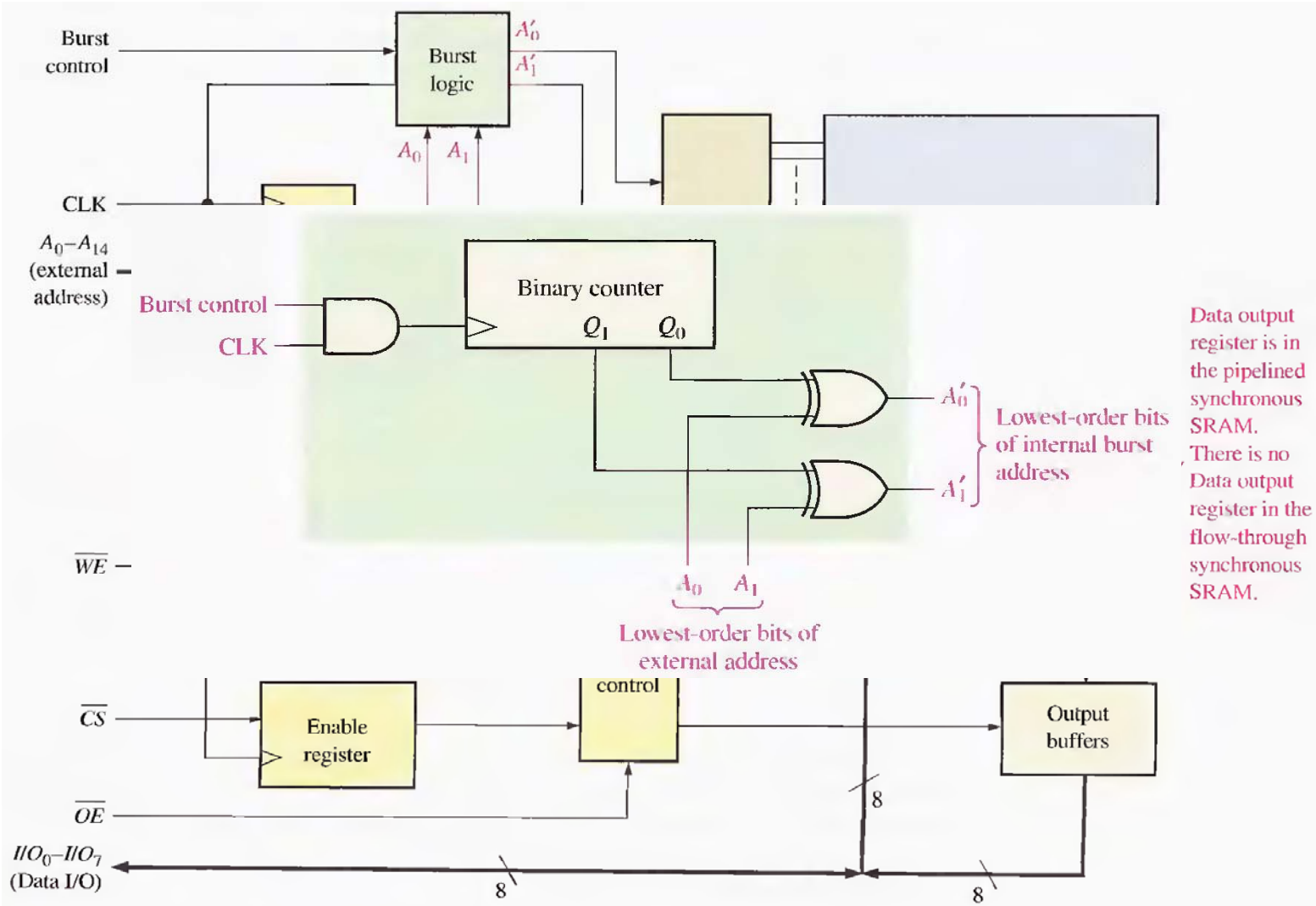


(a) Memory array configuration

(b) Memory block diagram

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

Basic Synchronous SRAM with Burst Feature



© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

Cache Memory

Refrigerator Vs. shop.

Cache memory is a relatively small, high-speed memory that stores the most recently used instructions or data from the larger but slower main memory.

Computer programs tend to get instructions or data from one area of main memory before moving to another area. Basically, the cache controller "guesses" which area of the slow dynamic memory the CPU will need next and moves it to the cache memory.

What will happen If the cache controller guesses right or wrong?

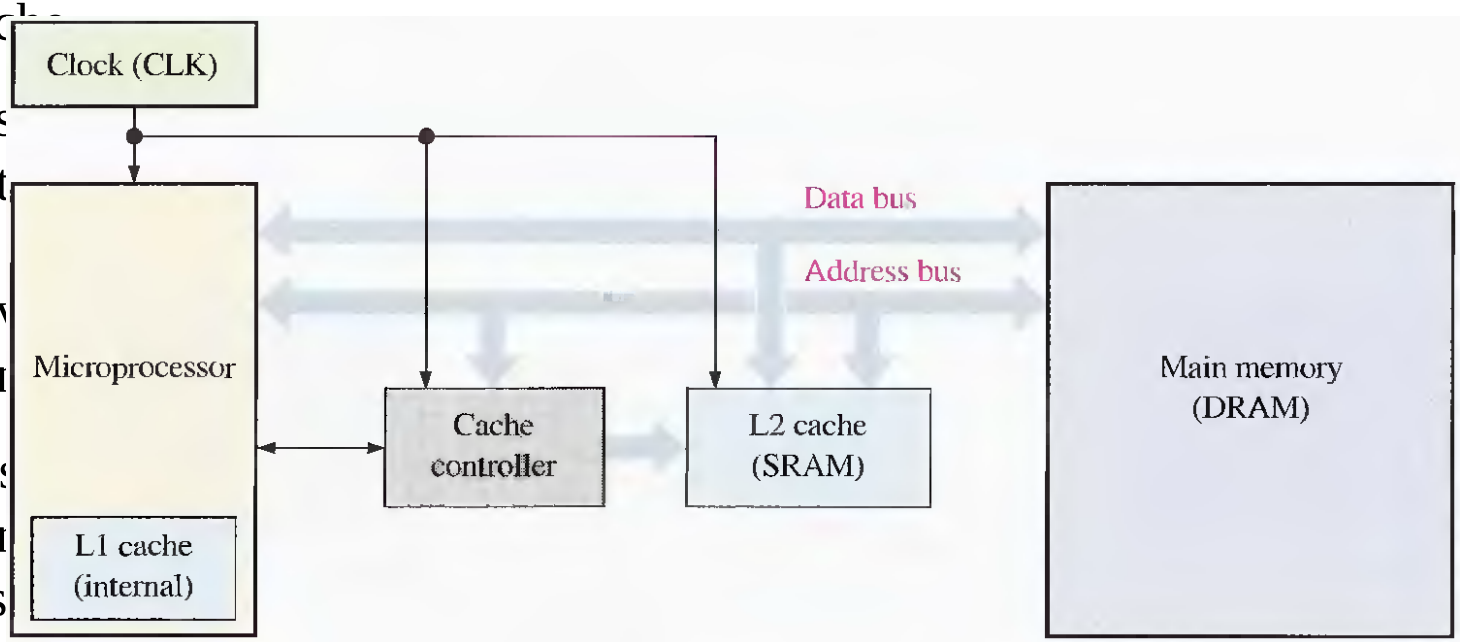
© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

L 1 and L2 Caches

A first-level cache (L1 cache) is usually integrated into the processor chip and has a very limited storage capacity. LI cache is also known as primary cac

A s ips
ext an
Ll ay
hav
co

Als
har
fas



© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

Dynamic RAM (DRAM) Memory Cells

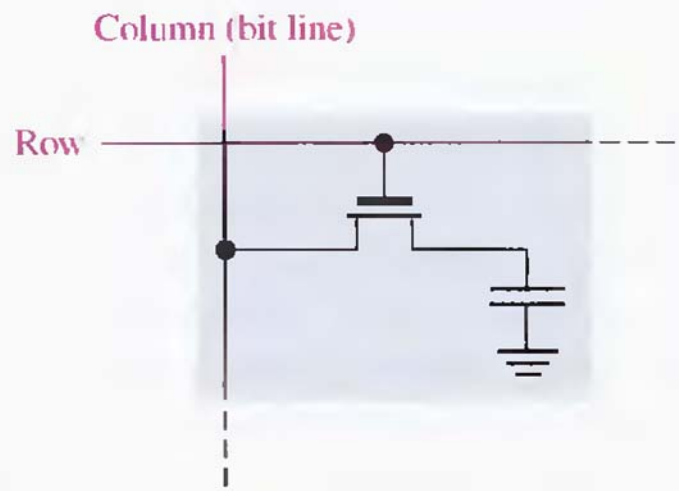
Dynamic memory cells store a data bit in a small capacitor rather than in a latch.

Cell is very simple, so very large memory arrays can be constructed on a chip at a lower cost per bit.

Capacitor cannot hold its charge for a long time unless its charge is refreshed periodically and this causes complexity of memory and related control circuit.

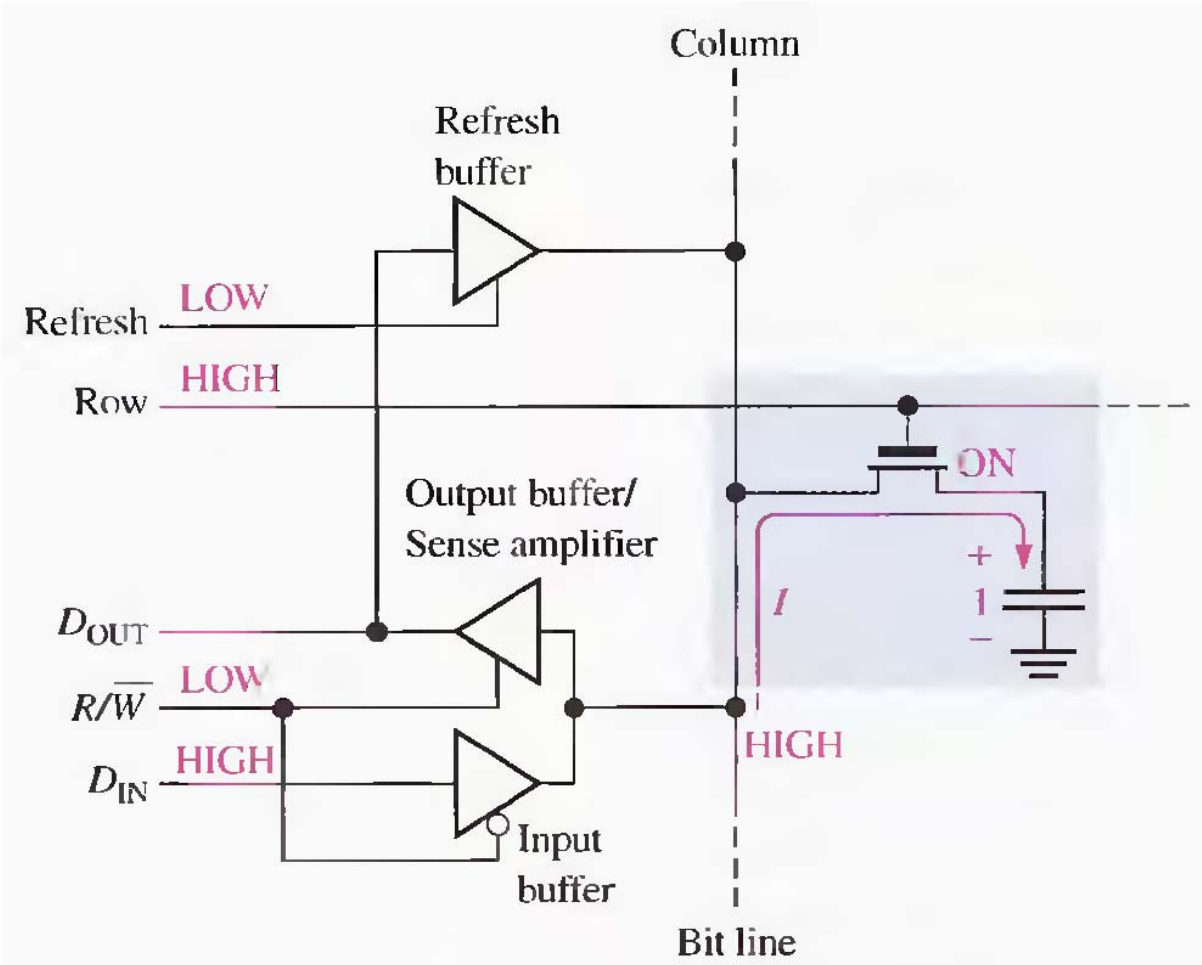
FIGURE 10–16

A MOS DRAM cell.



© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

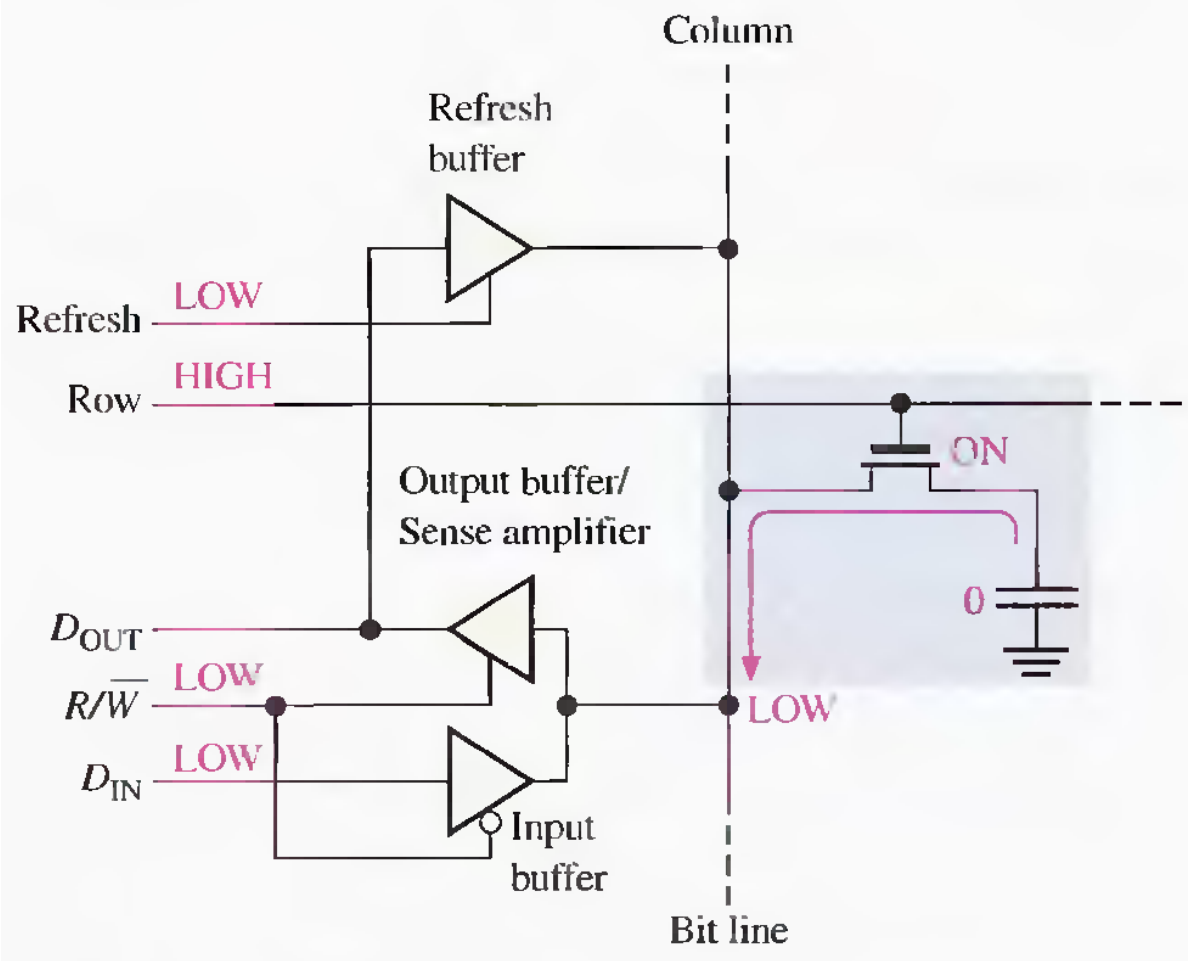
Dynamic RAM (DRAM) Memory Cells



(a) Writing a 1 into the memory cell

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

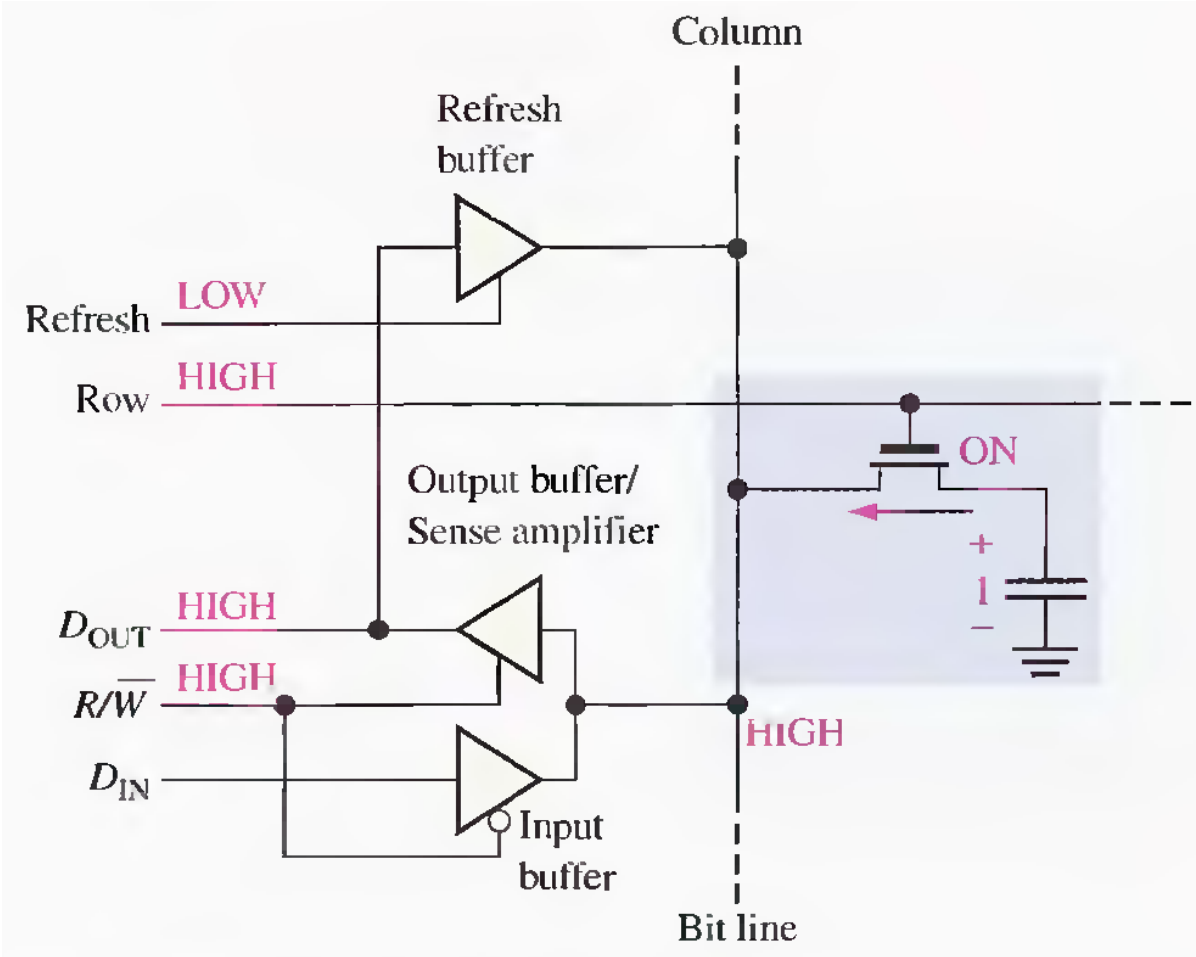
Dynamic RAM (DRAM) Memory Cells



(b) Writing a 0 into the memory cell

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

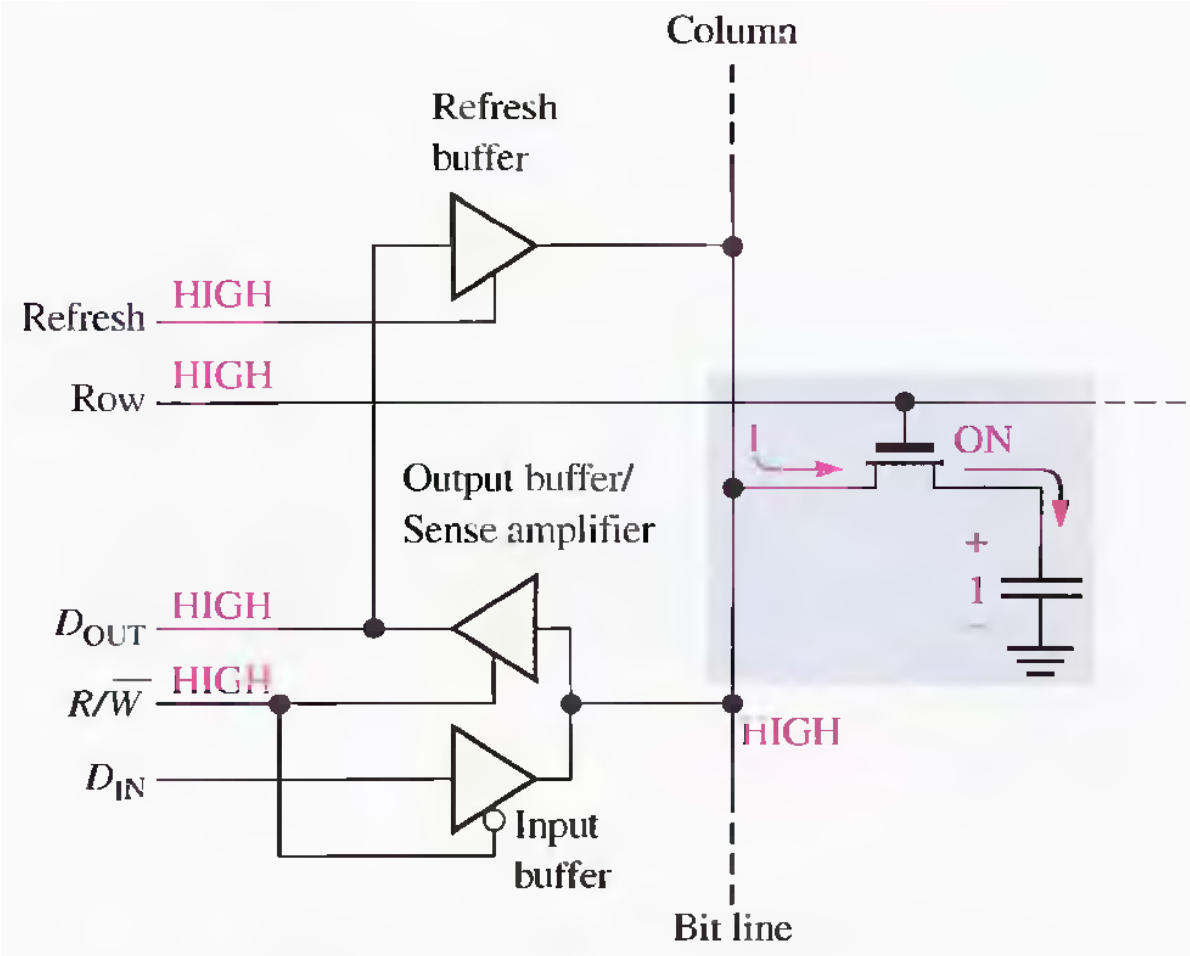
Dynamic RAM (DRAM) Memory Cells



(c) Reading a 1 from the memory cell

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

Dynamic RAM (DRAM) Memory Cells

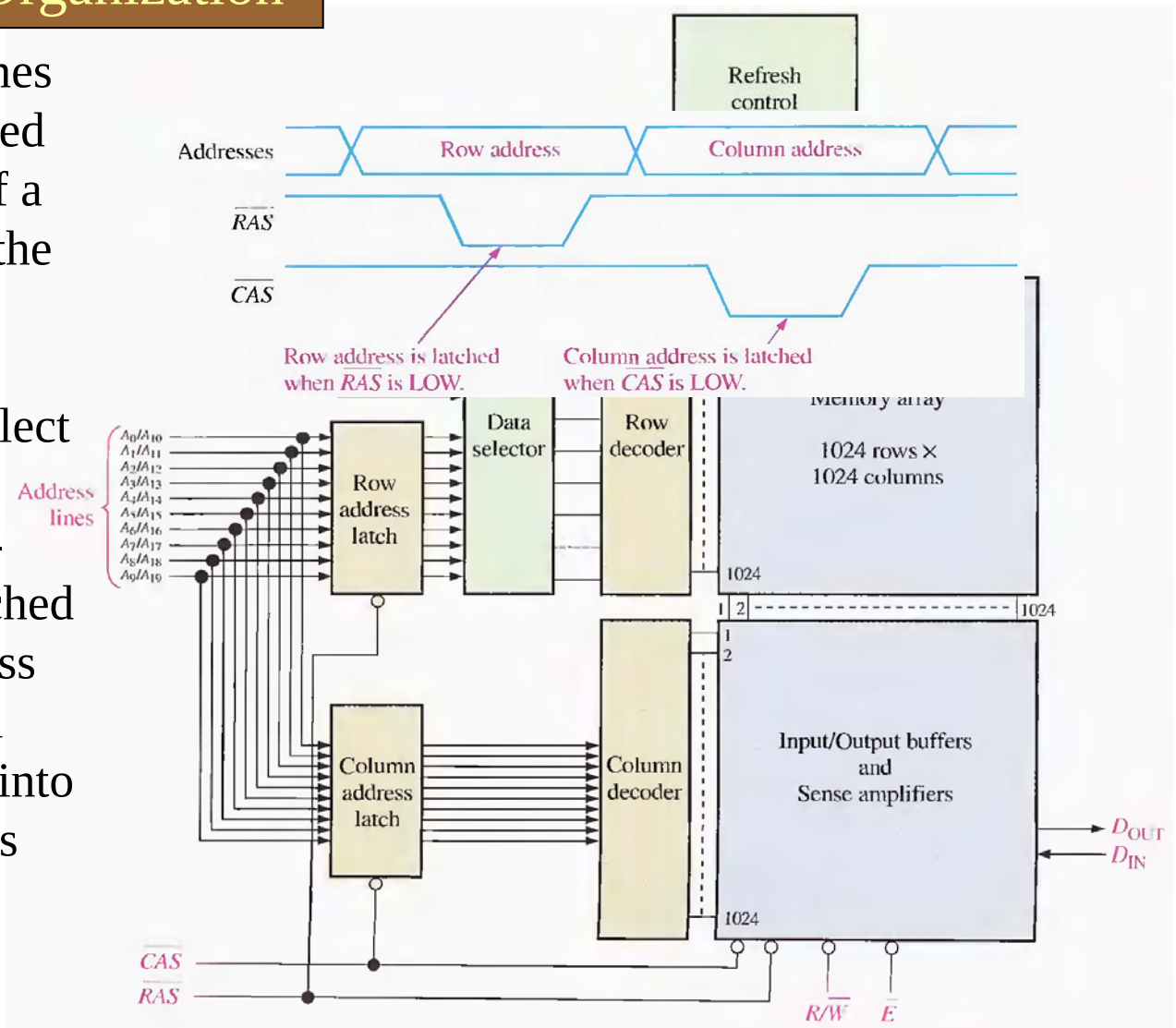


(d) Refreshing a stored 1

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

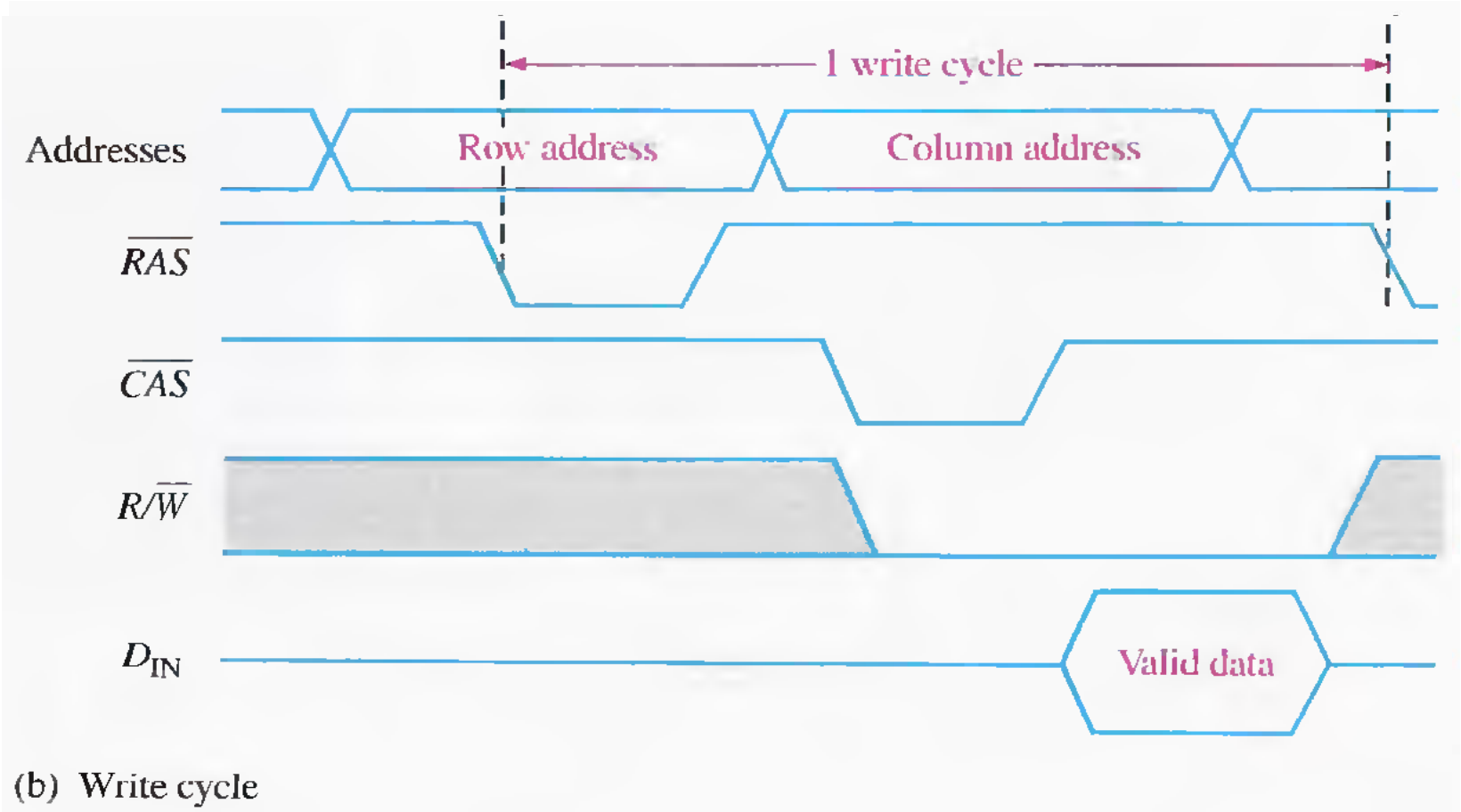
Basic DRAM Organization

The ten address lines are time multiplexed at the beginning of a memory cycle by the row address select (RAS) and the column address select (CAS) into two separate 10-bit ad- row address is latched into the row address latch. then column address is latched into the column address latch. .



© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

Read and Write Cycles



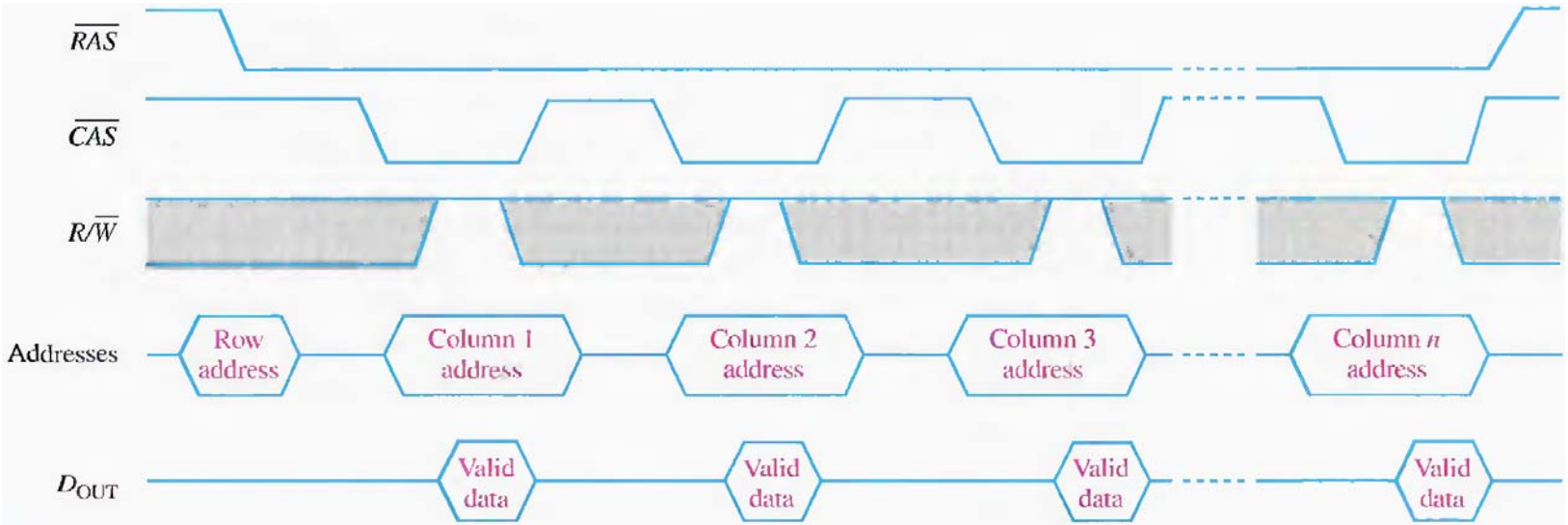
(b) Write cycle

آیا با توجه به ساختار پایه R/W حافظه همواره در حال خواندن و نوشتن است؟

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

Fast Page Mode

For example, a fast page mode cycle for the DRAM in Figure 10-18 requires CAS to go active 1024 times for each row selected by RAS.



When CAS goes to it s non-asserted state (HIGH), it disables the data outputs. Therefore, the transition of CAS to HIGH must occur only after valid data are latched by the external system.

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

Refresh Cycles

Typically, a DRAM must be refreshed every 8 ms to 16 ms, although for some devices the refresh period can exceed 100 ms.

A read operation automatically refreshes all the addresses in the selected row. However, in typical applications, you cannot always predict how often there will be a read cycle and so you cannot depend on a read cycle to occur frequently enough to prevent data loss.

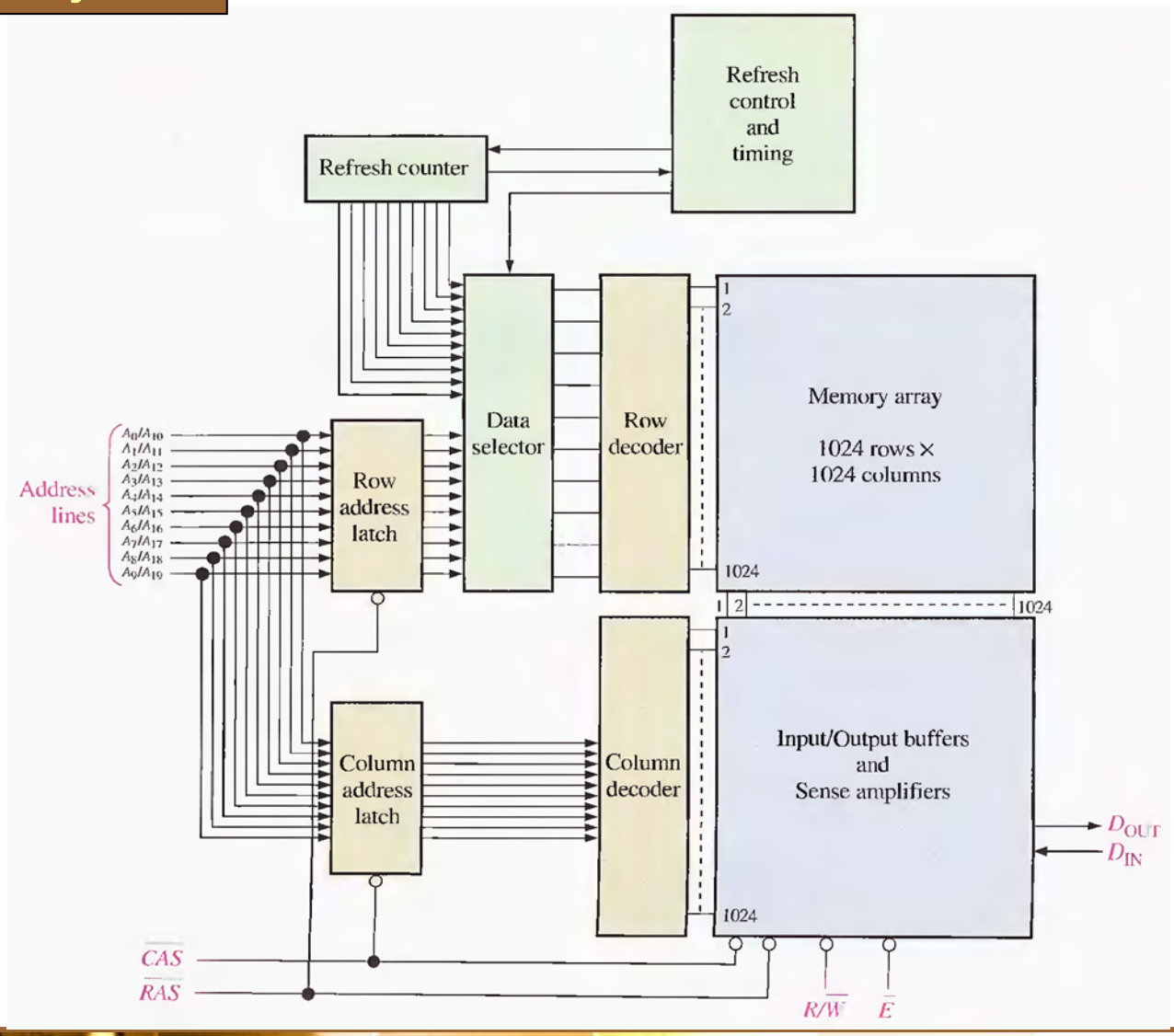
Burst refresh: All rows in the memory array are refreshed consecutively each refresh period. For a memory with a refresh period of 8 ms, a burst refresh of all rows occurs once every 8 ms. The normal read and write operations are suspended during a burst refresh cycle.

Distributed refresh: Each row is refreshed at intervals interspersed between normal read or write cycles. If memory had 1024 rows. As an example, for an 8 ms refresh period. each row must be refreshed every 8ms/I024 = 7.8 µs when distributed refresh is used.

he two types of refresh ope ratio ns are RAS-only refresh and CAS before RAS refresh.

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

Refresh Cycles



© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

Types of DRAMs

Fast Page Mode (FPM) DRAM: traditionally has been the most common type used in computers until the development of the EDO DRAM.

EDO DRAM : The Extended Data Output DRAM, sometimes called hyper page mode DRAM, is similar to the FPM DRAM. The key difference is that the CAS signal in the EDO DRAM does not disable the output data when it goes to its nonasserted state because the valid data from the current address can be held until CAS is asserted again. This means that the next column addres can be accessed before the external system accepts the current valid data. The idea is to speed up the access time.

BEDO DRAM : The Burst Extended Data Output DRAM is an EDO DRAM with address burst capability. Recall from the discussion of the synchronous burst SRAM that the address burst feature allows up to four addresses to be internally generated from a single external address, which saves some access time. This same concept applies to the BEDO DRAM.

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

Types of DRAMs

Synchronous DRAM (SDRAM):

Faster DRAMs are needed to keep up with the ever-increasing speed of microprocessors. The Synchronous DRAM is one way to accomplish this.

Like the synchronous static RAM discussed earlier, the operation of the SDRAM is synchronized with the system (CPU) clock. The same basic ideas described in relation to the synchronous burst SRAM, also apply to the SDRAM.

With asynchronous memories, the microprocessor must wait for the DRAM to complete its internal operations. However, with synchronous operation, the DRAM latches addresses, data, and control information from the processor under control of the system clock. This allows the processor to handle other tasks while the memory read or write operations are in progress

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

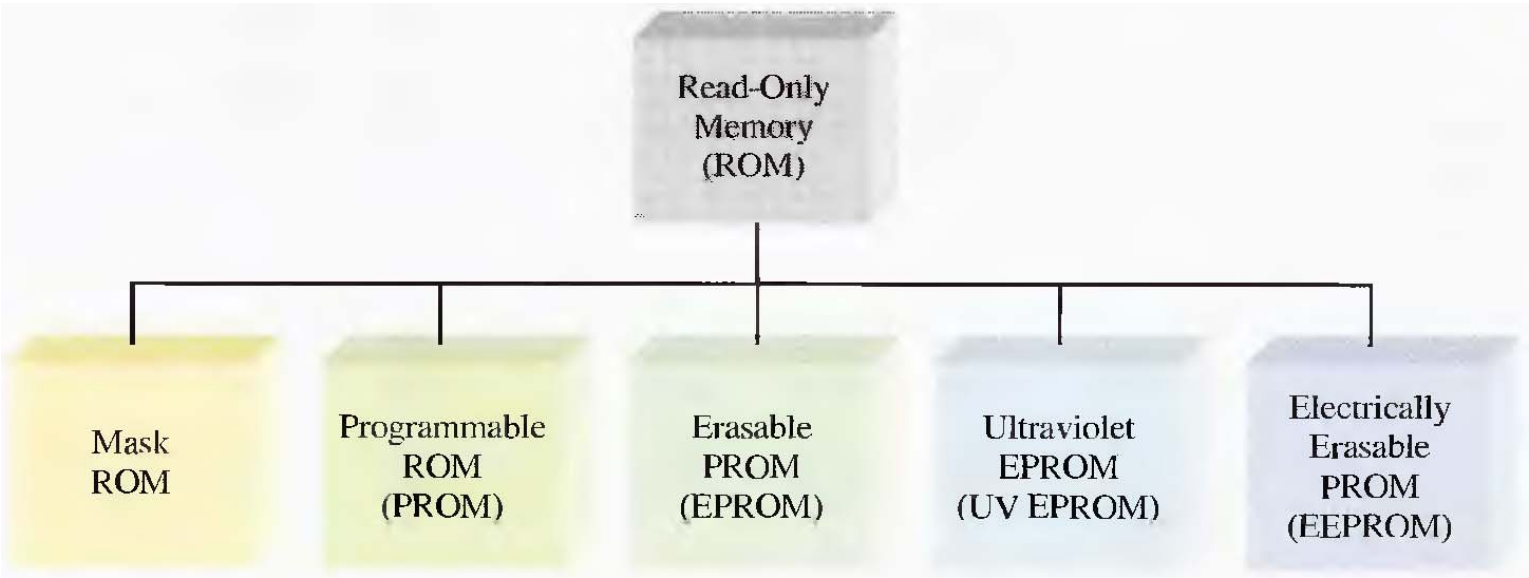
SECTION 10-2 REVIEW

1. List two types of SRAM.
2. What is a cache?
3. Explain how SRAMs and DRAMs differ.
4. Describe the refresh operation in a DRAM.
5. List four types of DRAM.

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

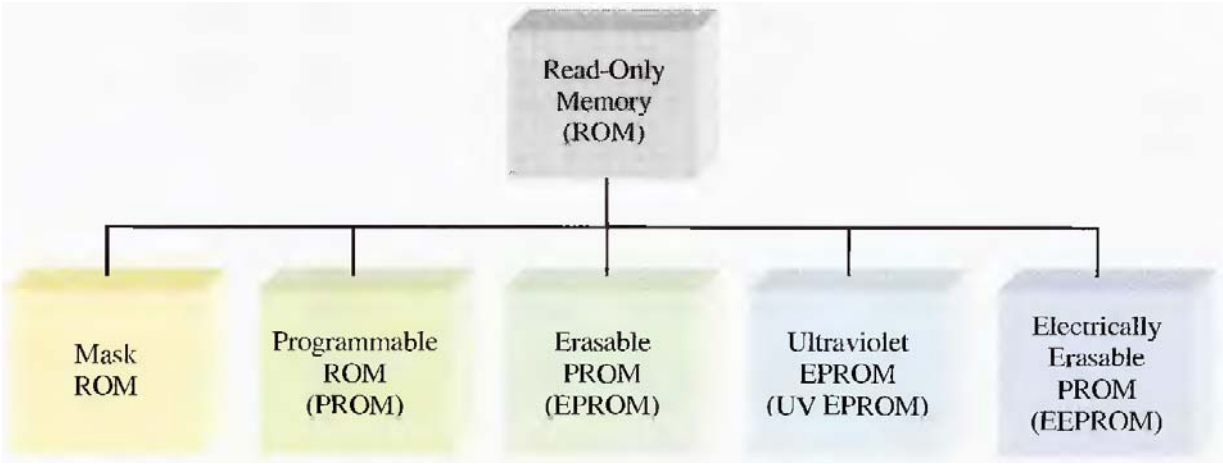
READ-ONLY MEMORIES (ROMs)

- A ROM contains permanently or semi permanently stored data
- Cannot be changed or cannot be changed without specialized equipment.
- Are used in system application&, such as tables. conversions, or programmed instructions for system initialization (BIOS) and operation.
- Nonvolatile memories.

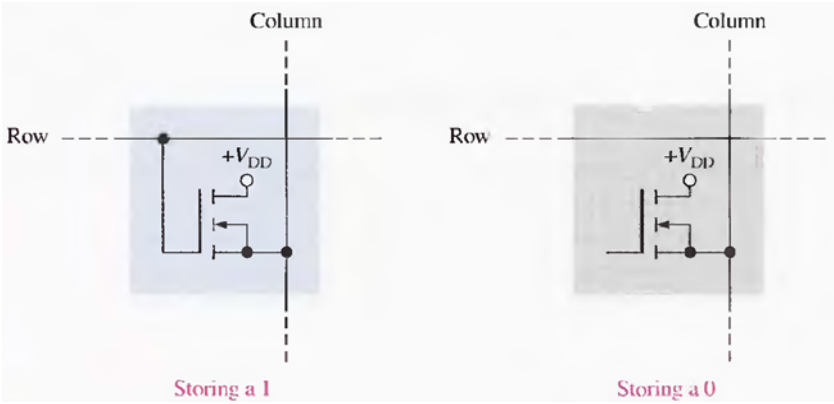


© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

READ-ONLY MEMORIES (ROMs)

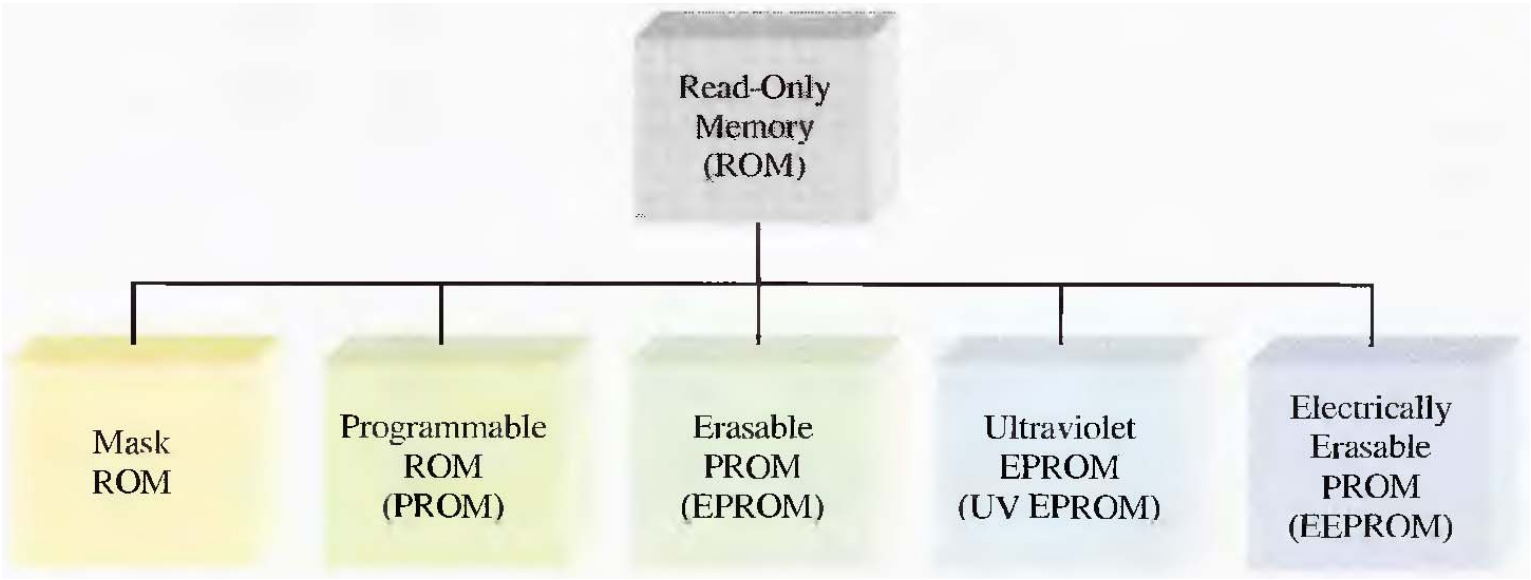


MASK ROM (usually named ROM) Is the type in which the data are permanently stored in the memory during the manufacturing process to provide widely used standard functions.



© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

READ-ONLY MEMORIES (ROMs)



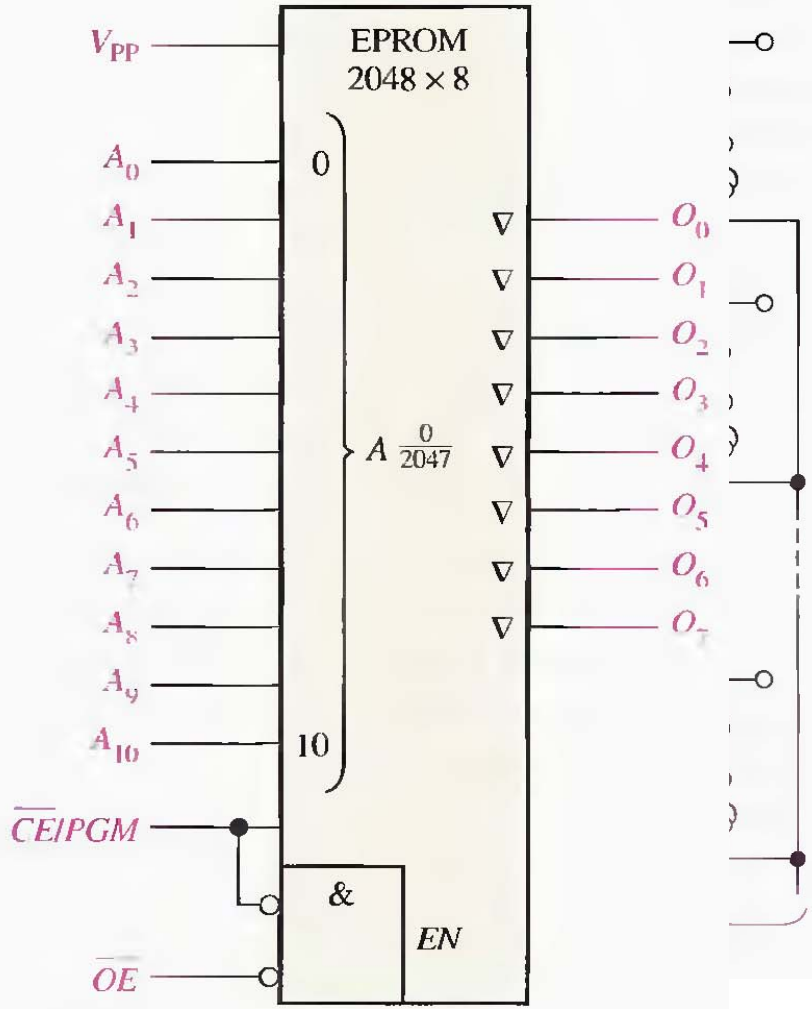
The PROM, or programmable ROM, is the type in which the data are electrically stored by the user with the aid of specialized equipment. Both the mask ROM and the PROM can be of either MOS or bipolar technology.

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

PROM STRUCTURE

FIGURE 10–31

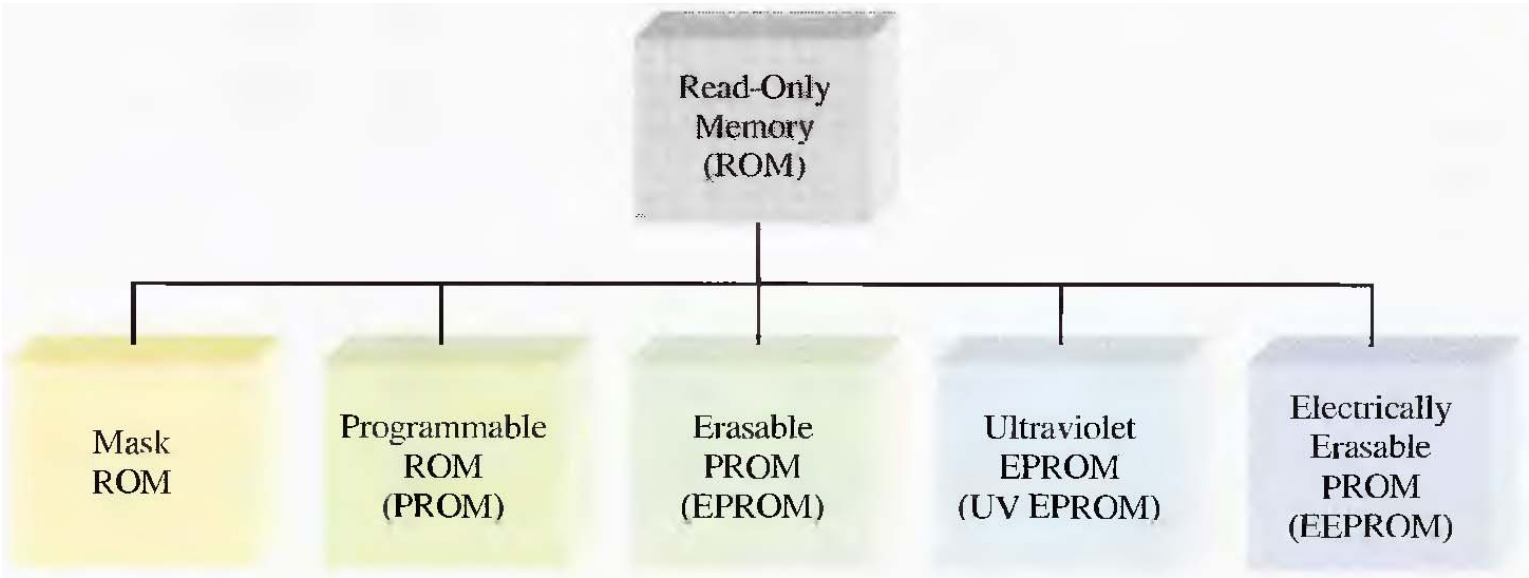
The logic symbol for a 2048 × 8 UV EPROM.



MOS
fusibl
comm

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

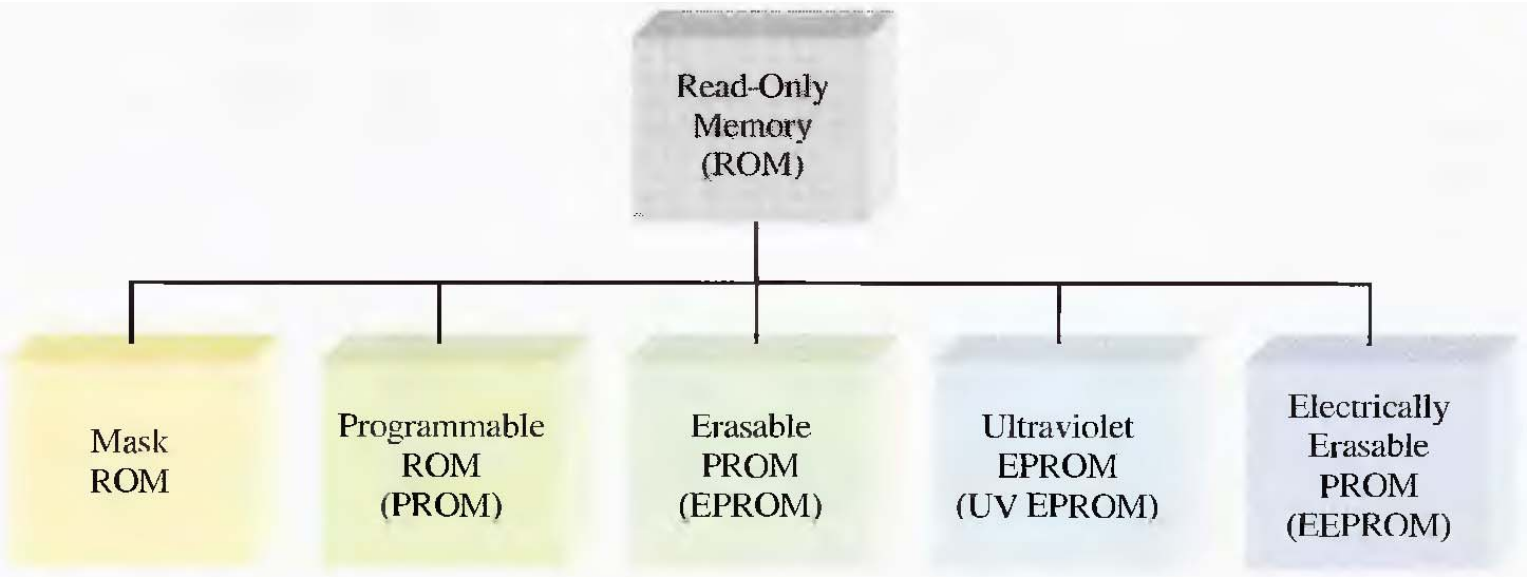
READ-ONLY MEMORIES (ROMs)



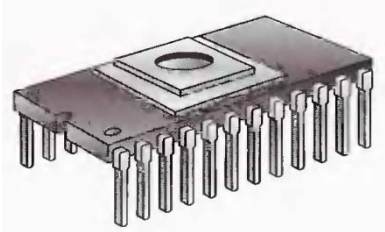
The EPROM, or erasable PROM. is strictly a MOS device.

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

READ-ONLY MEMORIES (ROMs)

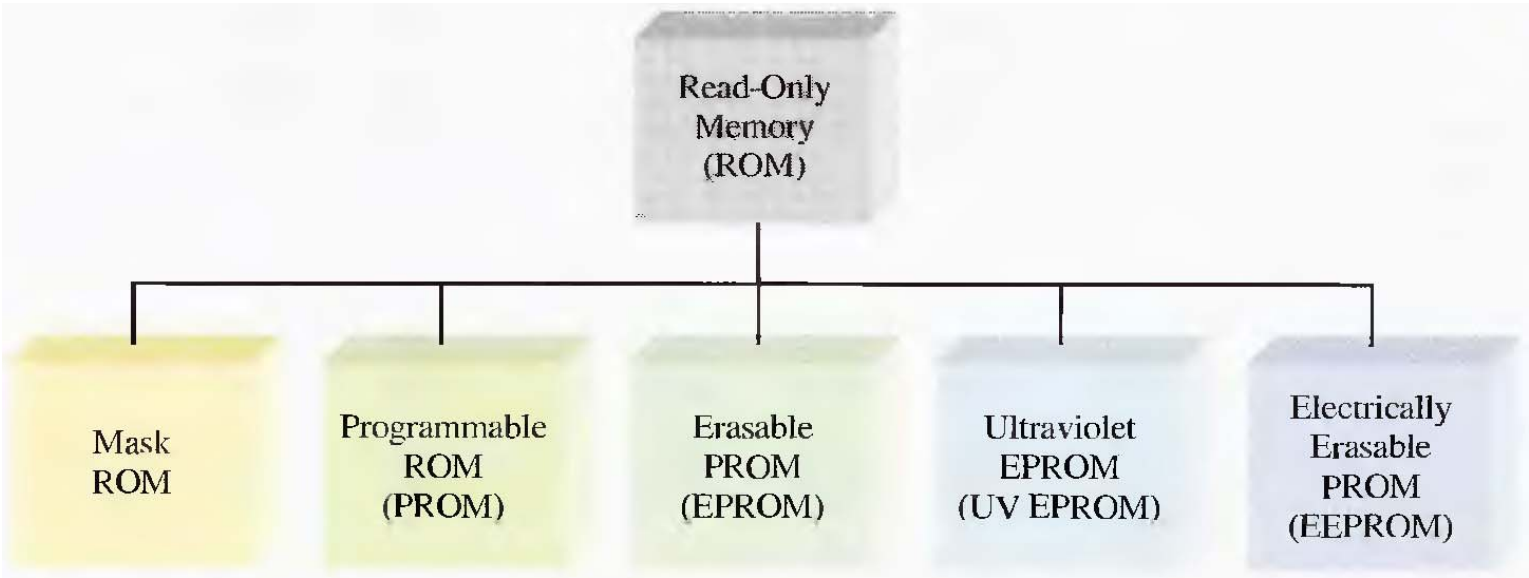


The UV EPROM is electrically programmable by the user, but the stored data must be erased by exposure to ultraviolet light over a period of several minutes.



© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

READ-ONLY MEMORIES (ROMs)

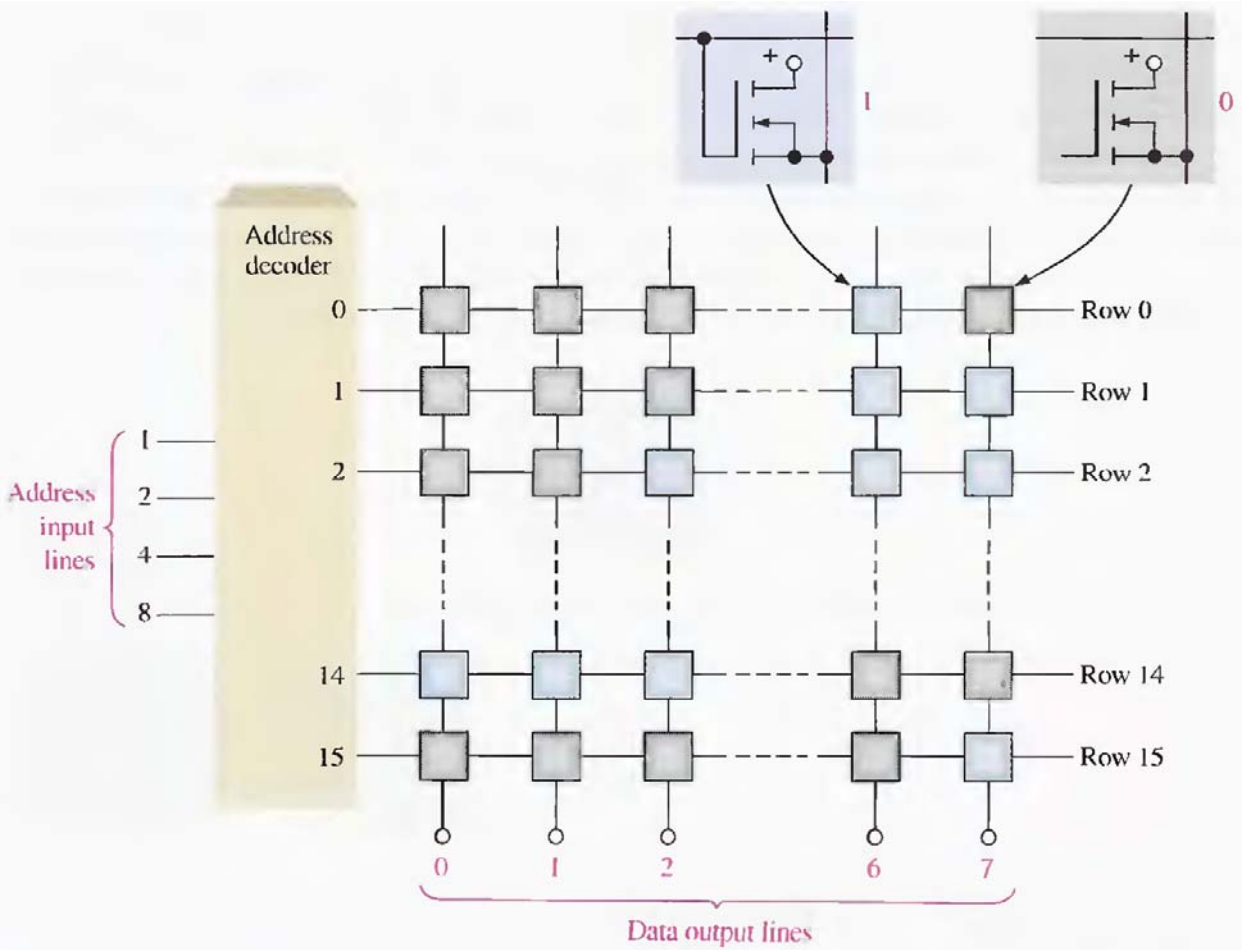


The electrically erasable PROM (EEPROM or E2PROM) can be erased in a few milliseconds.

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

MASK ROM's (ROMs)

The blue squares represent stored I s, and the gray squares represent stored 0's.



© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

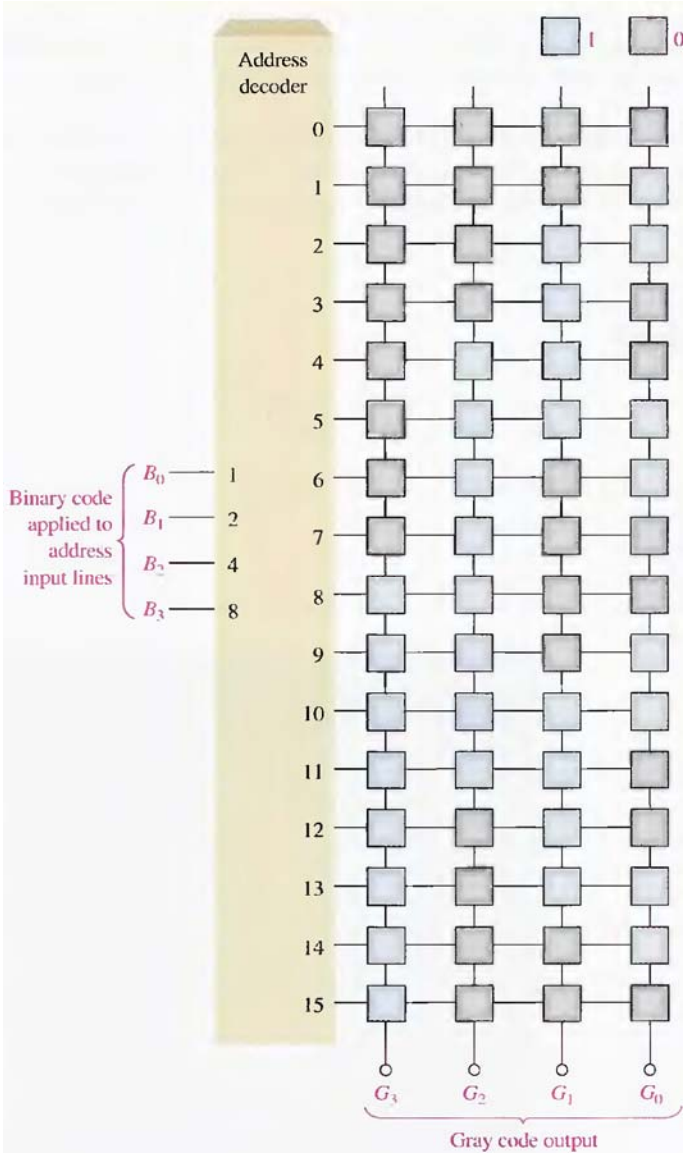
EXAMPLE 10-1

Show a basic ROM, programmed for a 4-bit binary-to-Gray conversion.

BINARY				GRAY			
B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

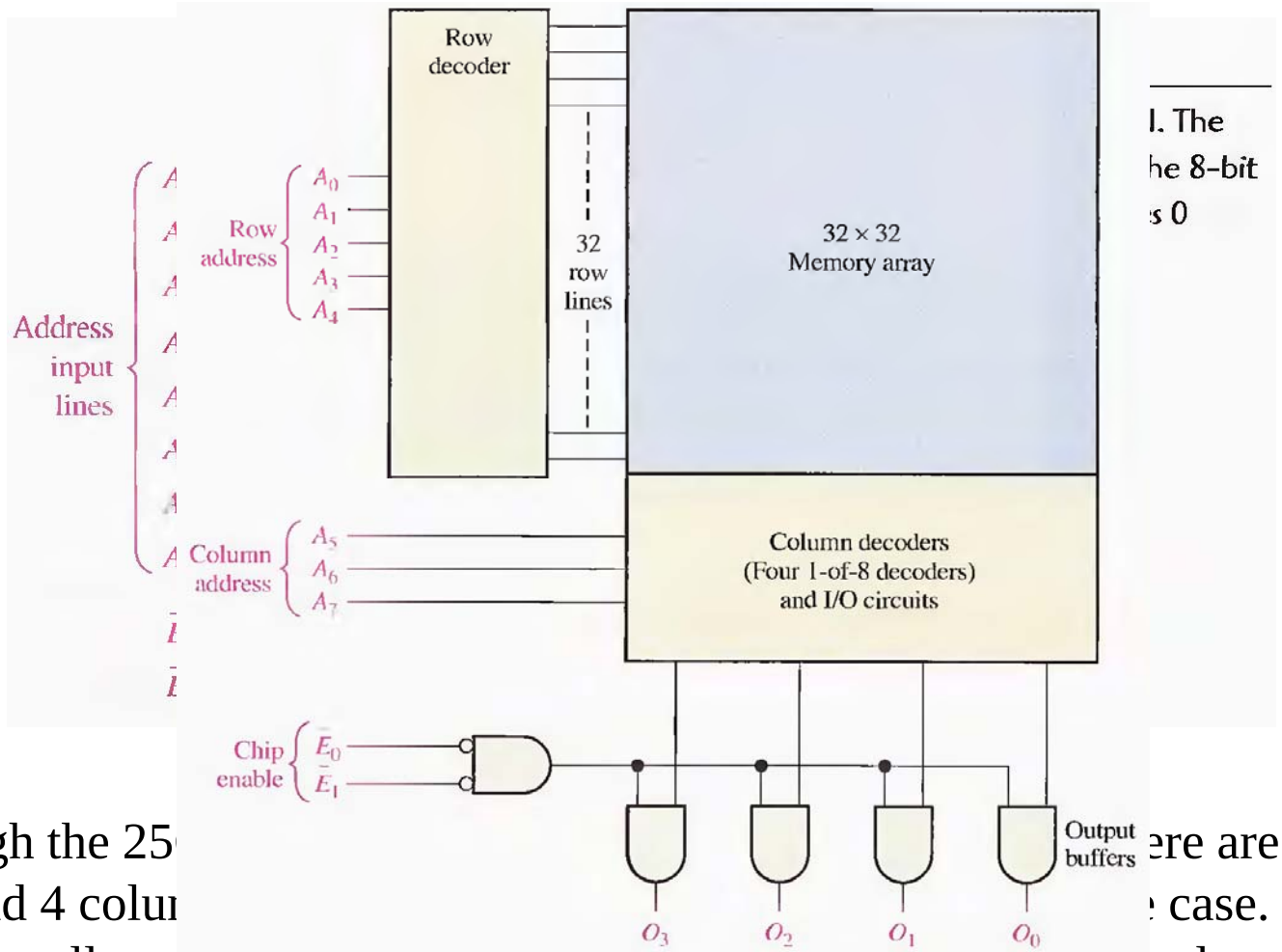
© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

EXAMPLE 10-1



© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

Internal ROM Organization

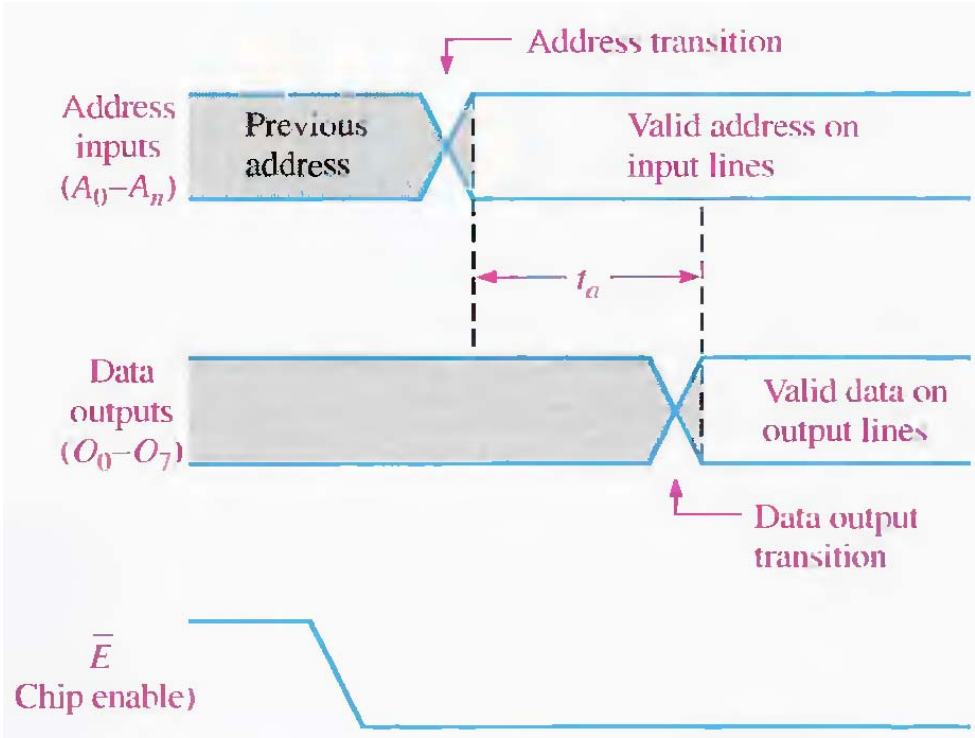


Although the 25 ... ere are 256 rows and 4 colu ... case. The memory cell array is actually a 32 x 32 matrix (32 rows and 32 columns)

Actually, the column decoder consists of four 1-of-8 multiplexer

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

ROM Access Time



Access time can also be measured from the activation of the chip enable (E) input to the occurrence of valid output data when a valid address is already on the input lines.

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

SECTION 10-3 REVIEW

1. What is the bit storage capacity of a ROM with a 512 X 8 organization?
2. List the types of read-only memories.
3. How many address bits are required for a 2048-bit memory organized as a 256 X 8 memory?

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

FLASH MEMORIES

The ideal memory has high storage capacity, non-volatility, in-system read and write capability, comparatively fast operation, and cost effectiveness. None of previous mentioned technologies has all of them.

This high density is achieved in flash memories with a storage cell that consists of a single floating-gate MOS transistor. A data bit is stored as charge or the absence of charge on the floating gate depending if a 0 or a 1 is stored.

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

MEMORIES COMPARISON

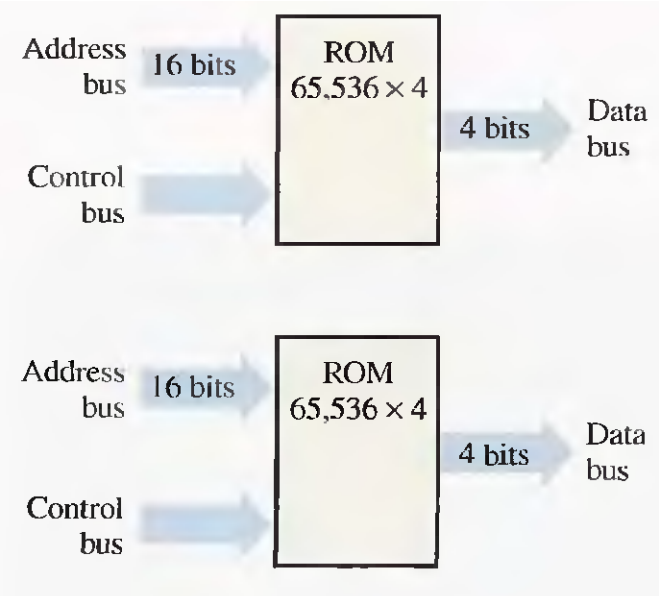
TABLE 10–2

Comparison of types of memories.

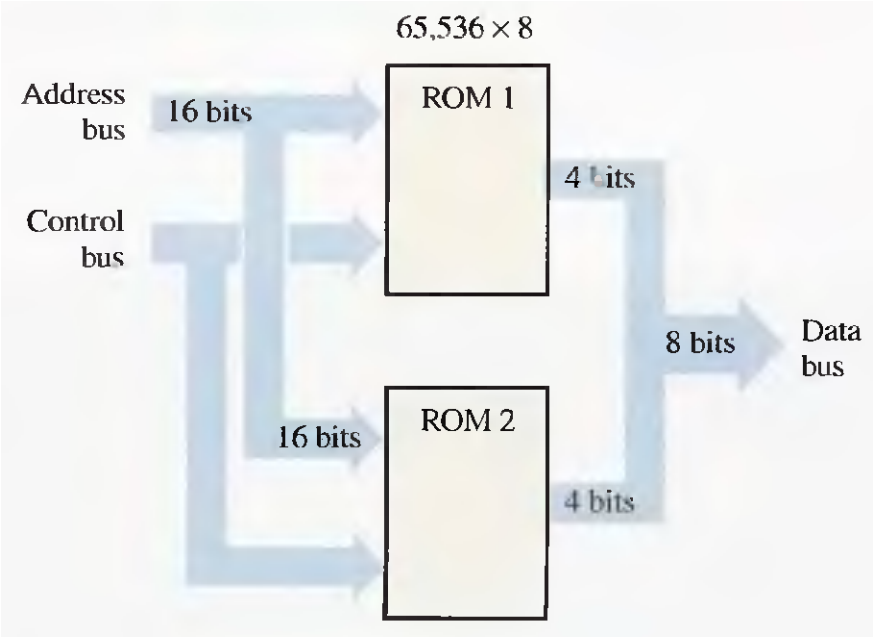
MEMORY TYPE	NONVOLATILE	HIGH-DENSITY	ONE-TRANSISTOR CELL	IN-SYSTEM WRITABILITY
Flash	Yes	Yes	Yes	Yes
SRAM	No	No	No	Yes
DRAM	No	Yes	Yes	Yes
ROM	Yes	Yes	Yes	No
EPROM	Yes	Yes	Yes	No
EEPROM	Yes	No	No	Yes

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

Word-length Expansion



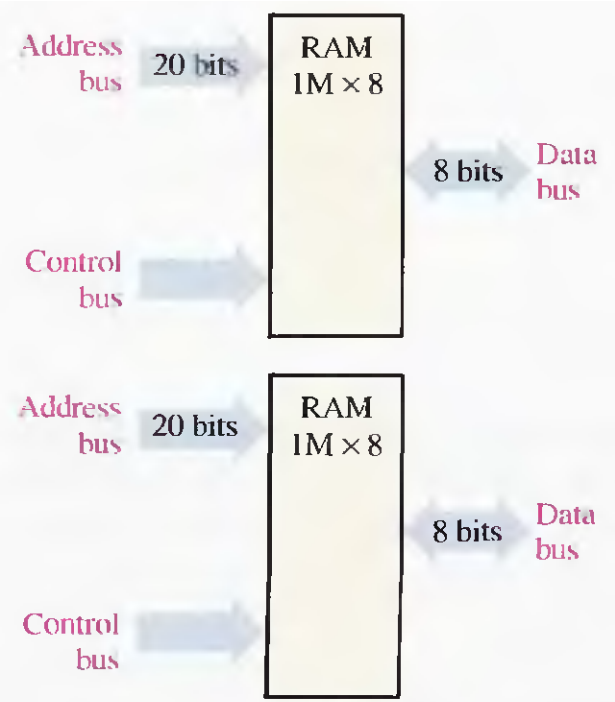
(a) Two separate 65,536 × 4 ROMs



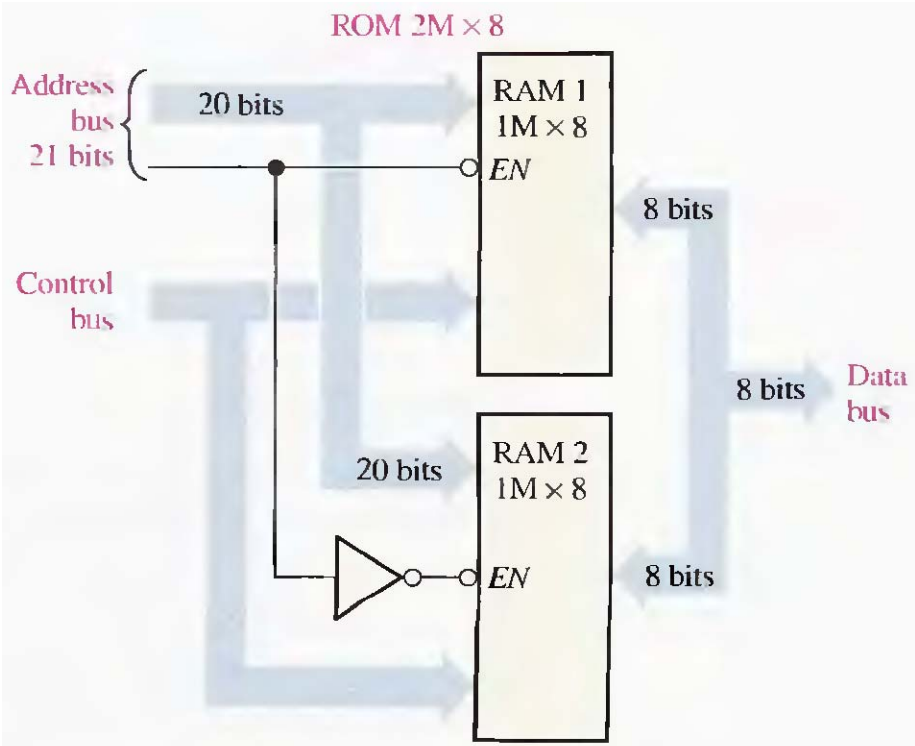
(b) One 65,536 × 8 ROM from two 65,536 × 4 ROMs

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved

Word-Capacity Expansion



(a) Individual memories each store 1,048,576 8-bit words



(b) Memories expanded to form a 2M × 8 RAM requiring a 21-bit address bus

© 2009 Pearson Education, Upper Saddle River, NJ 07458. All Rights Reserved