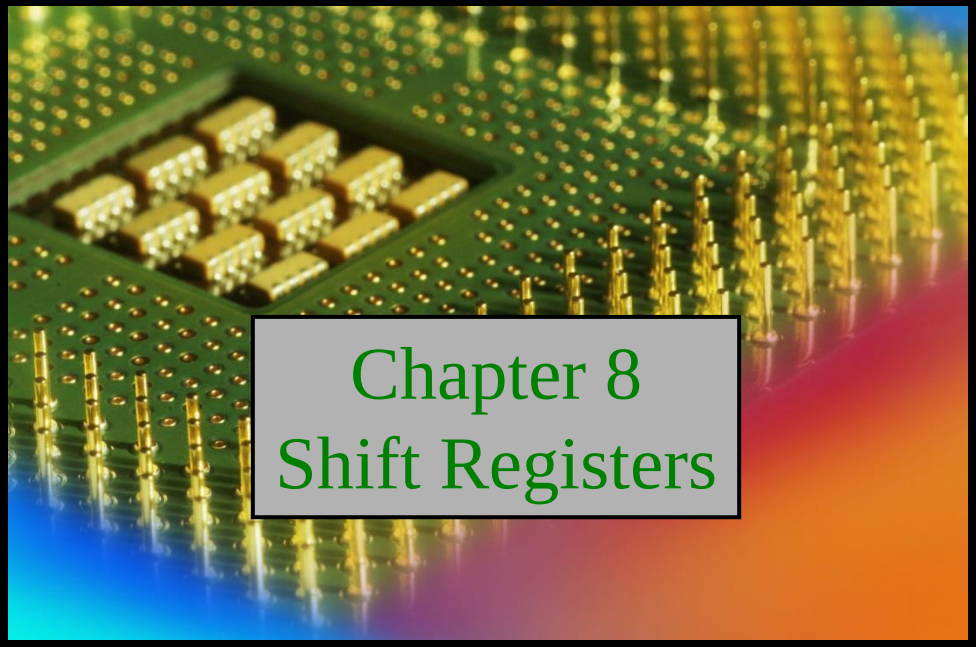
EE-227
Digital Logic Design

Spring-2019



Chapter 8
Shift Registers

Course Instructor: Engr. Khalid Iqbal Soomro

Basic Shift Register Operations

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 8–1illustrates the concept of storing a 1 or a 0 in a D flip-flop. A 1 is applied to the data input as shown, and a clock pulse is applied that stores the 1 by *setting* the flip-flop. When the 1 on the input is removed, the flip-flop remains in the SET state, thereby storing the 1. A similar procedure applies to the storage of a 0 by *resetting* the flip-flop, as also illustrated in Figure 8–1.

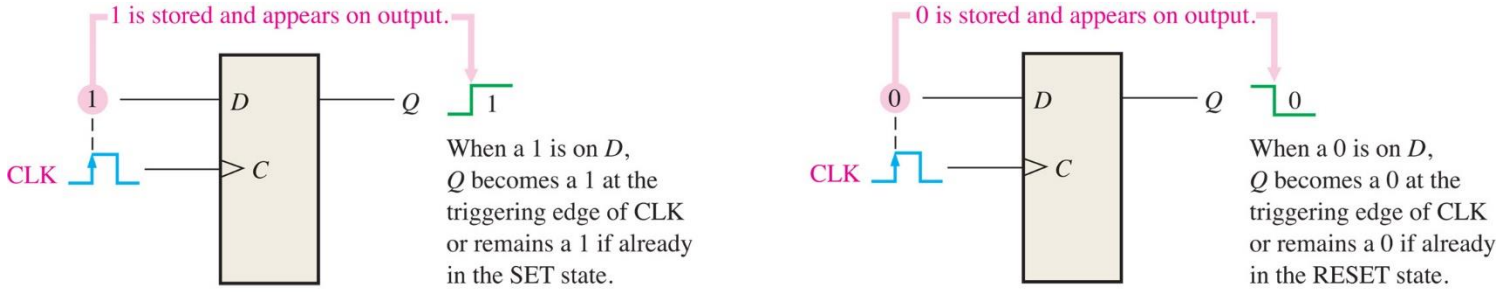


FIGURE 8-1 The flip-flop as a storage element.

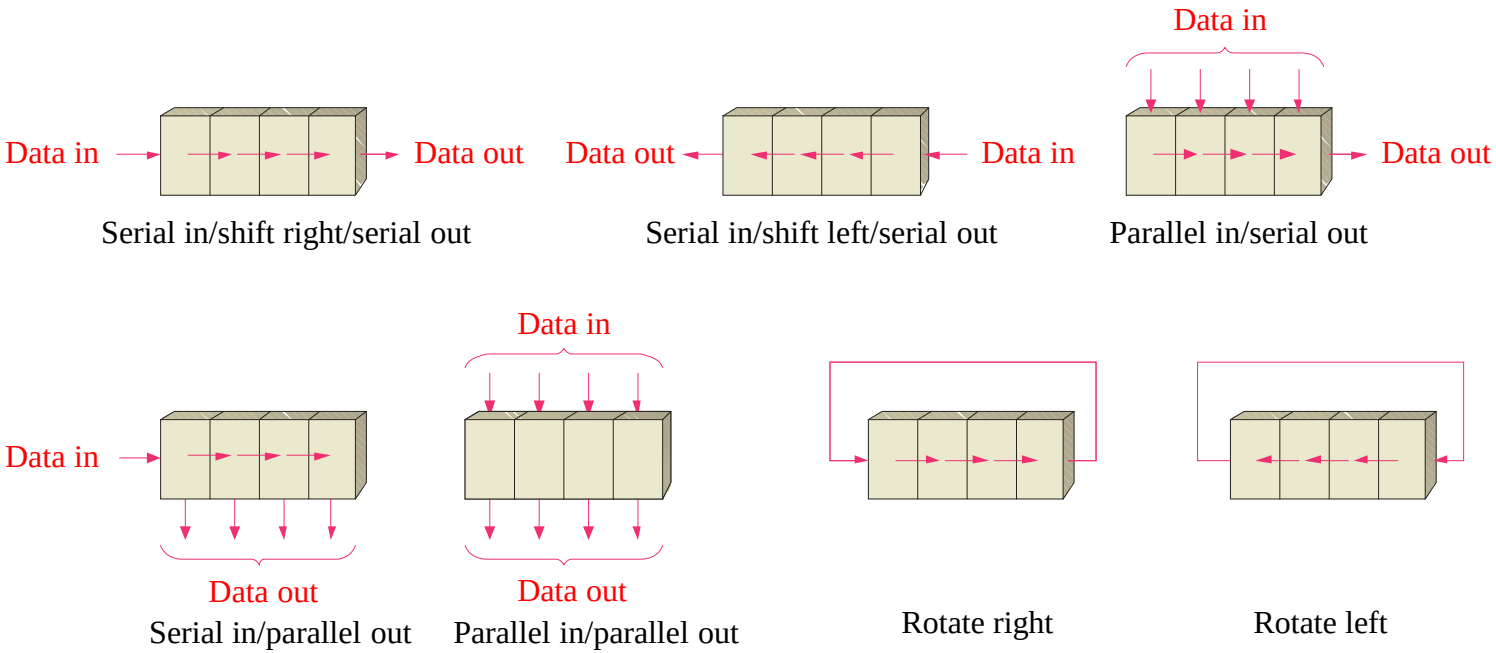
Shift Register Storage Capacity

The *storage capacity* of a register is the total number of bits (1s and 0s) of digital data it can retain. Each **stage** (flip-flop) in a shift register represents one bit of storage capacity; therefore, the number of stages in a register determines its storage capacity.

The *shift capability* of a register permits the movement of data from stage to stage within the register or into or out of the register upon application of clock pulses.

Shift Register Data Movement types

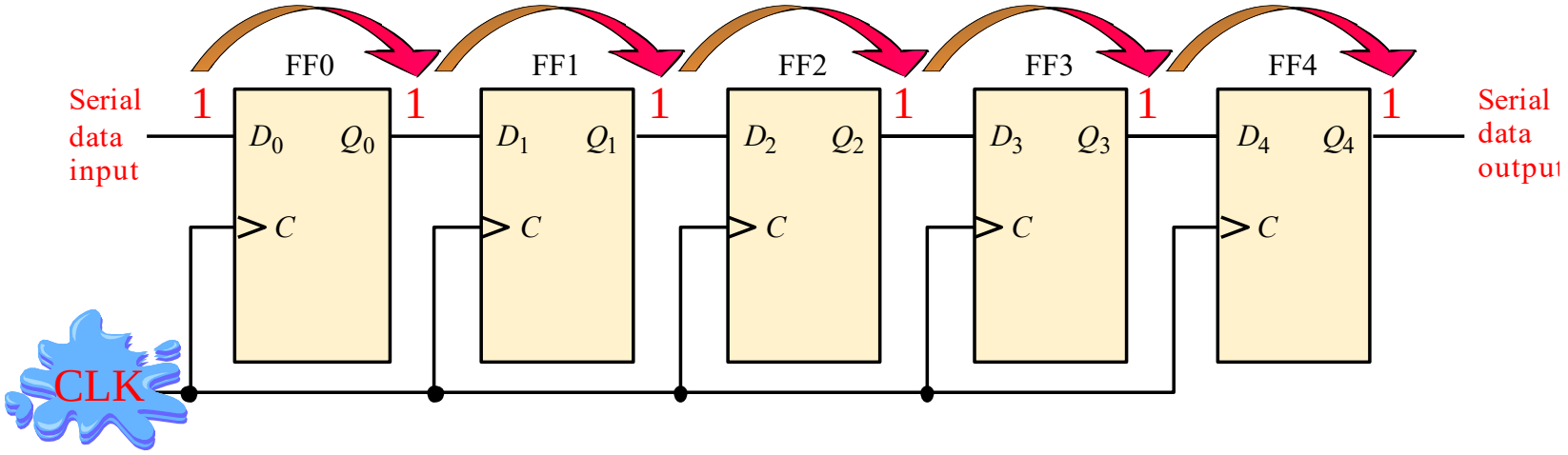
Some basic Shift Register data movements are illustrated here.



Serial-in/Serial out Shift Register

Shift registers are available in IC form or can be constructed from discrete flip-flops as shown here with a five-bit serial-in serial-out register.

Each clock pulse will move an input bit to the next flip-flop. For example, a 1 is shown as it moves across.



InfoNote

Frequently, it is necessary to *clear* an internal register in a processor.

For example, a register may be cleared prior to an arithmetic or other operation.

One way that registers in a processor are cleared is using software to subtract the contents of the register from itself. The result, of course, will always be zero.

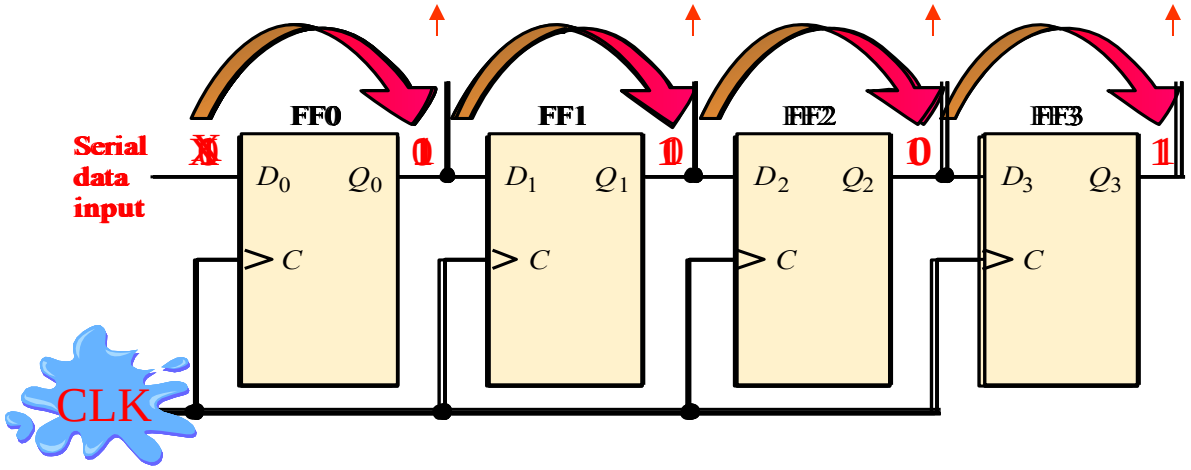
For example, a processor instruction that performs this operation is *SUB AL,AL*. with this instruction, the register named AL is cleared.

A Basic Application

An application of shift registers is conversion of serial data to parallel form.

For example, assume the binary number 1011 is loaded sequentially, one bit at each clock pulse.

After 4 clock pulses, the data is available at the parallel output.



Parallel in/Serial out Shift Register

Figure 8–10 illustrates a 4-bit parallel in/serial out shift register and a typical logic symbol.

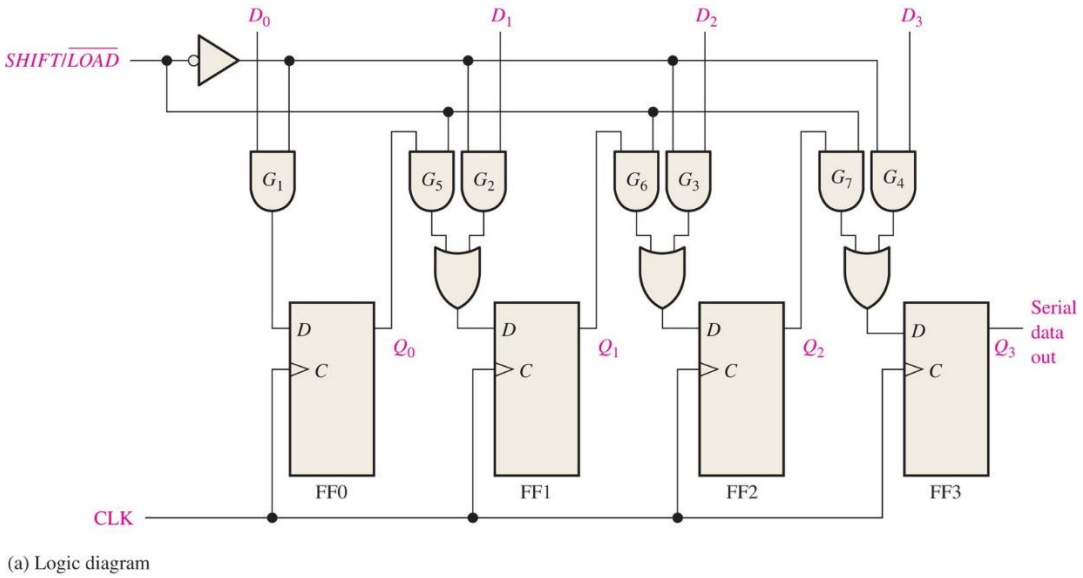


FIGURE 8-10
A 4-bit parallel in/serial out shift register.

Parallel in/Serial out Shift Register

There are four data-input lines, D_0, D_1, D_2, and D_3, and a SHIFT/$\overline{\text{LOAD}}$ input, which allows four bits of data to load in parallel into the register.

When SHIFT/$\overline{\text{LOAD}}$ is LOW, gates G_1 through G_4 are enabled, allowing each data bit to be applied to the D input of its respective flip-flop.

When a clock pulse is applied, the flip-flops with $D = 1$ will set and those with $D = 0$ will reset, thereby storing all four bits simultaneously.

Parallel in/Serial out Shift Register

When SHIFT/$\overline{\text{LOAD}}$ is HIGH, gates G_1 through G_4 are disabled and gates G_5 through G_7 are enabled, allowing the data bits to shift right from one stage to the next.

The OR gates allow either the normal shifting operation or the parallel data-entry operation, depending on which AND gates are enabled by the level on the SHIFT/$\overline{\text{LOAD}}$ input.

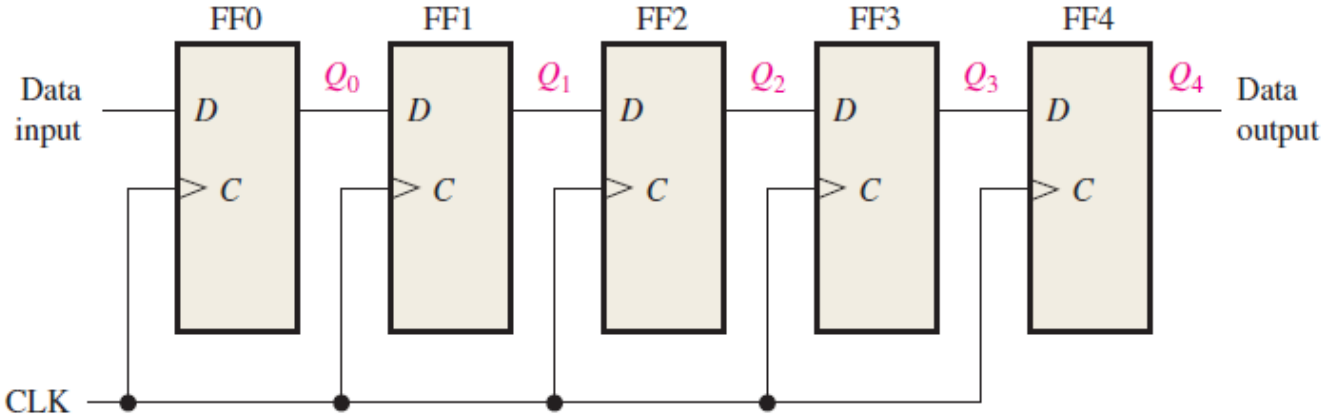
Notice that FF_0 has a single AND to disable the parallel input, D_0. It does not require an AND/OR arrangement because there is no serial data in.

EXAMPLE 8–1

Show the states of the 5-bit register in Figure 8–4(a) for the specified data input and clock waveforms. Assume that the register is initially cleared (all 0s).

Solution

The first data bit (1) is entered into the register on the first clock pulse and then shifted from left to right as the remaining bits are entered and shifted. The register contains $Q_4Q_3Q_2Q_1Q_0 = 11010$ after five clock pulses. See Figure 8–4(b).



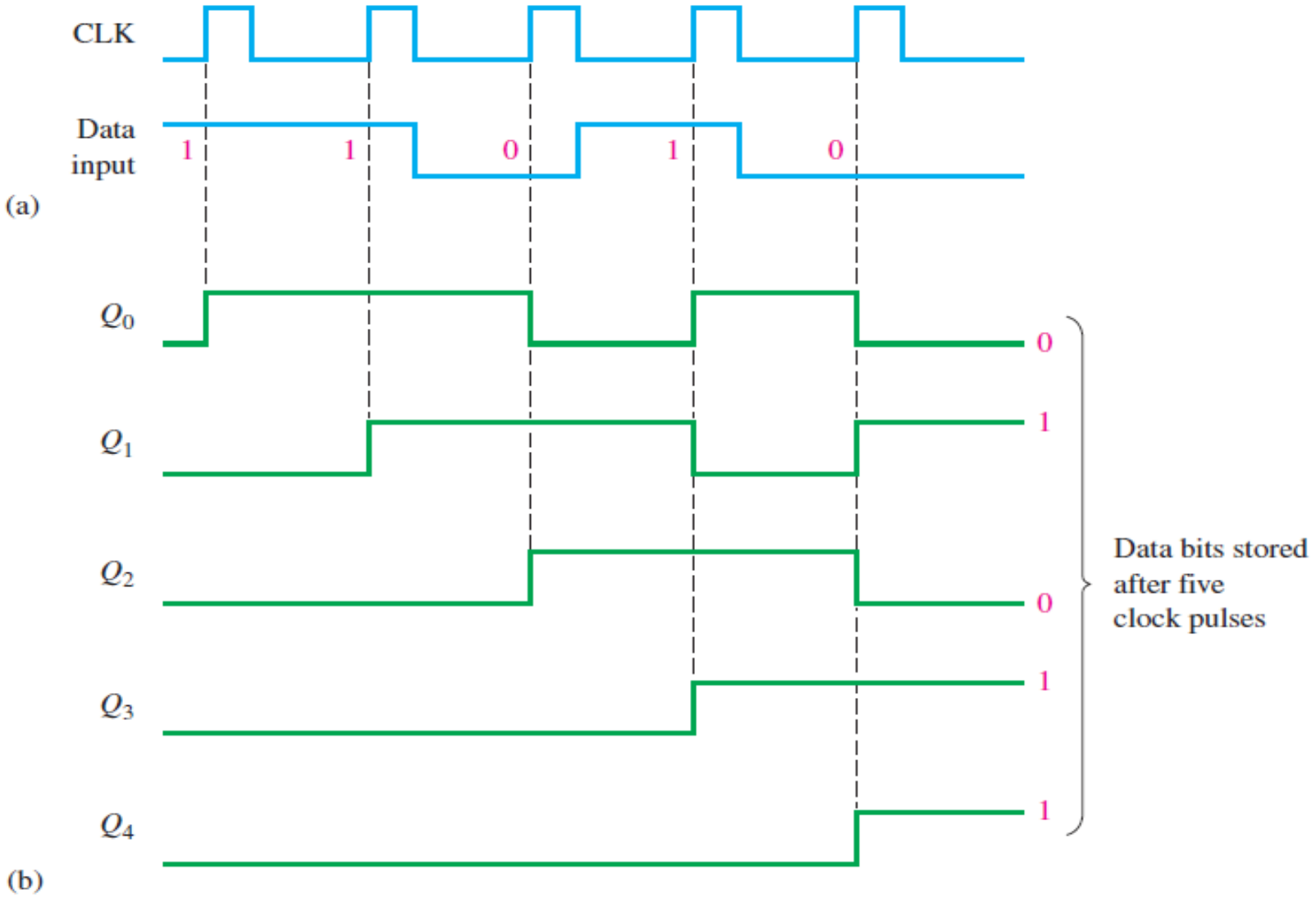

CLK
Data input
1
1
0
1
0
(a)
Q_0
Q_1
Q_2
Q_3
Q_4
0
1
0
1
1
Data bits stored after five clock pulses
(b)

FIGURE 8–4

Parallel in/Parallel out Shift Register

The parallel in/parallel out register employs both methods. Immediately following the simultaneous entry of all data bits, the bits appear on the parallel outputs. *Figure 8–14* shows a parallel in/parallel out shift register.

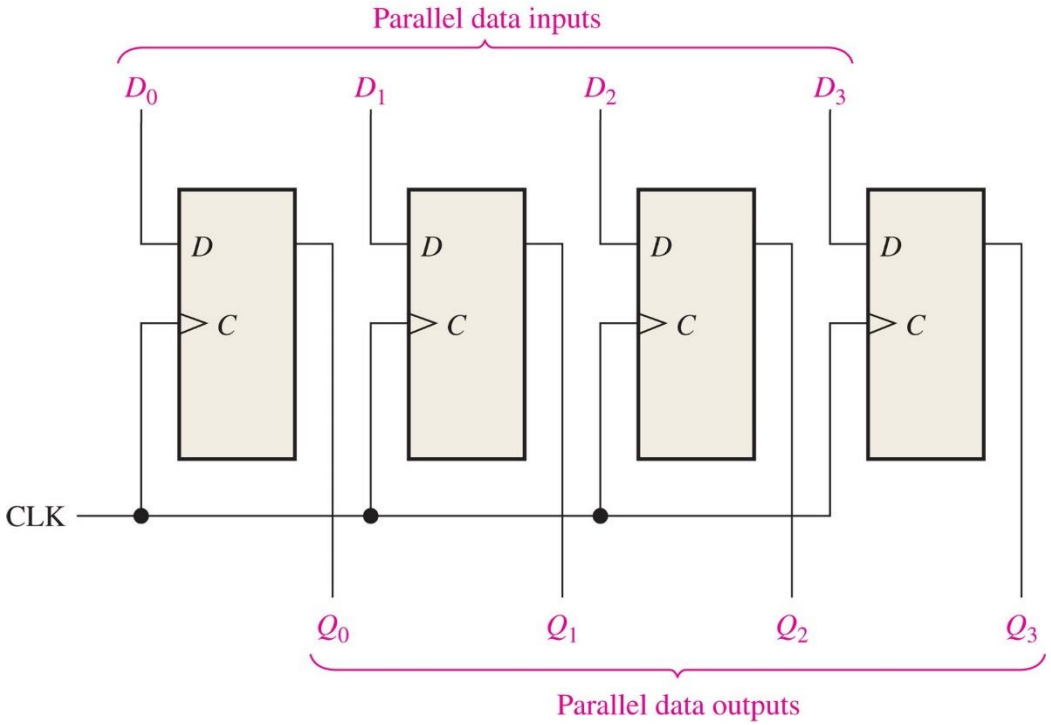


FIGURE 8-14
A parallel in/parallel out register.

Bidirectional Shift Register

A bidirectional shift register is one in which the data can be shifted either left or right.

It can be implemented by using gating logic that enables the transfer of a data bit from one stage to the next stage to the right or to the left, depending on the level of a control line.

A 4-bit bidirectional shift register is shown in Figure 8–17.

Bidirectional Shift Register

A HIGH on the RIGHT/$\overline{\text{LEFT}}$ control input allows data bits inside the register to be shifted to the *right*, and a LOW enables data bits inside the register to be shifted to the *left*.

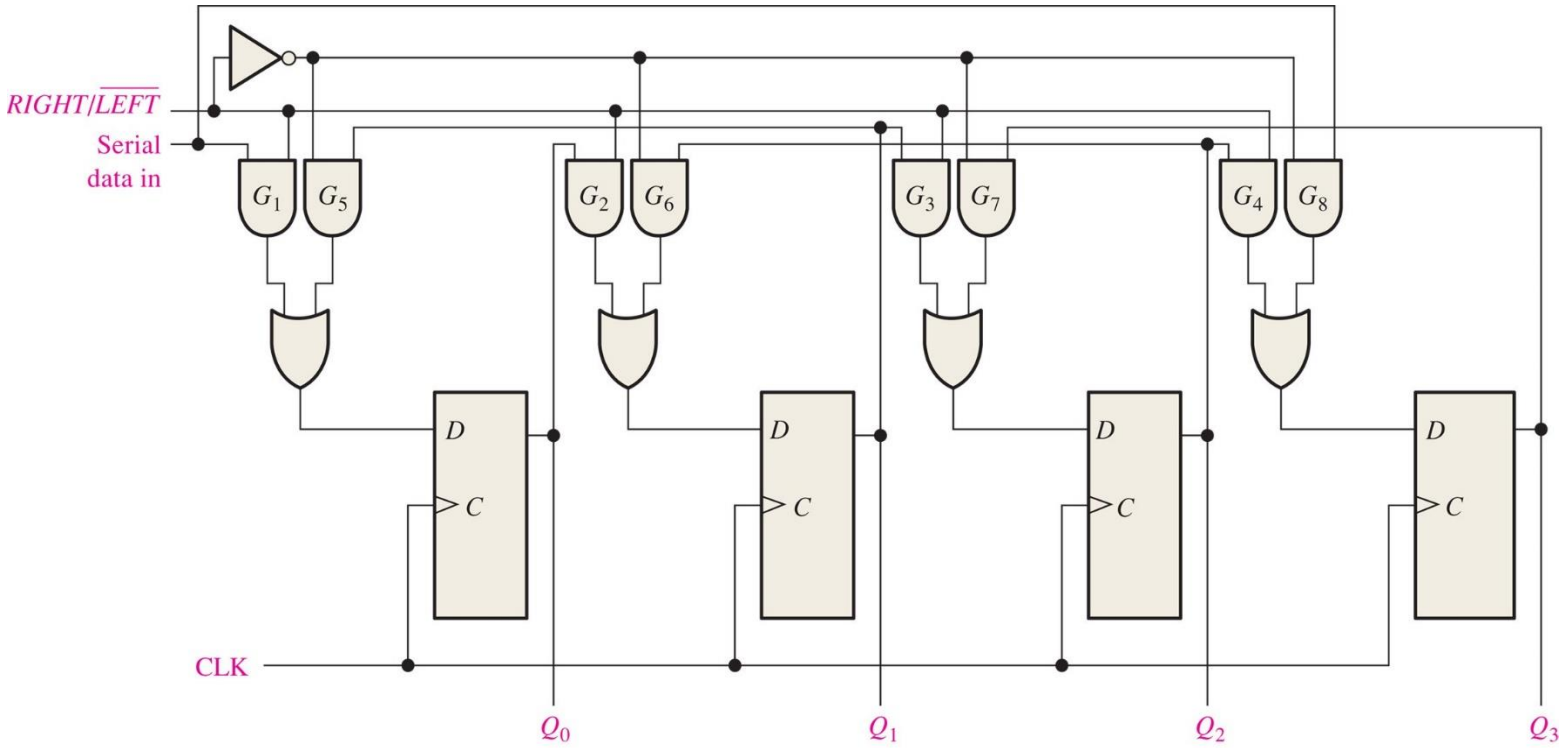


FIGURE 8-17 Four-bit bidirectional shift register.

Bidirectional Shift Register

An examination of the gating logic will make the operation apparent.

When the RIGHT/$\overline{\text{LEFT}}$ control input is HIGH, gates G_1 through G_4 are enabled, and the state of the Q output of each flip-flop is passed through to the D input of the following flip-flop.

When a clock pulse occurs, the data bits are shifted one place to the right.

When the RIGHT/$\overline{\text{LEFT}}$ control input is LOW, gates G_5 through G_8 are enabled, and the Q output of each flip-flop is passed through to the D input of the preceding flip-flop.

When a clock pulse occurs, the data bits are then shifted one place to the left.

Shift Register Counters

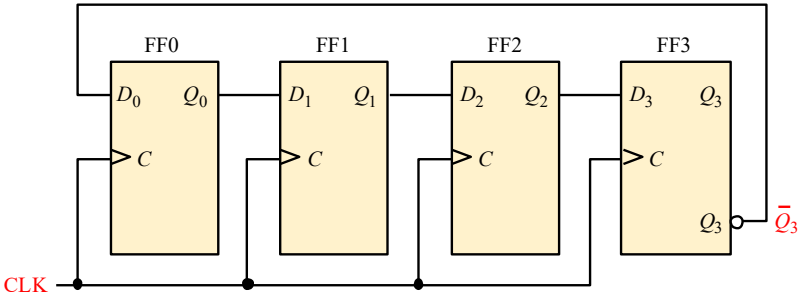
A shift register counter is basically a shift register with the serial output connected back to the serial input to produce special sequences.

These devices are often classified as counters because they exhibit a specified sequence of states.

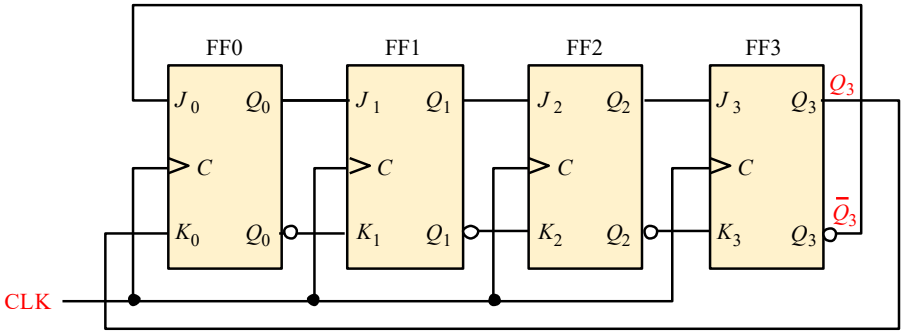
Two important shift register counters are the *Johnson counter* and the *ring counter*

Johnson counter

The Johnson counter can be made with a series of D flip-flops



… or with a series of J-K flip flops. Here Q_3 and $\bar{Q}_3$ are fed back to the *J* and *K* inputs with a "twist".



Johnson Counter

In a **Johnson counter** the complement of the output of the last flip-flop is connected back to the *D* input of the first flip-flop (it can be implemented with other types of flip-flops as well).

If the counter starts at 0, this feedback arrangement produces a characteristic sequence of states, as shown in Table 8–3 for a 4-bit device and in Table 8–4 for a 5-bit device.

Notice that the 4-bit sequence has a total of eight states, or bit patterns, and that the 5-bit sequence has a total of ten states. In general, a Johnson counter will produce a modulus of 2_n - where *n* is the number of stages in the counter.

TABLE 8–3

Four-bit Johnson sequence.

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

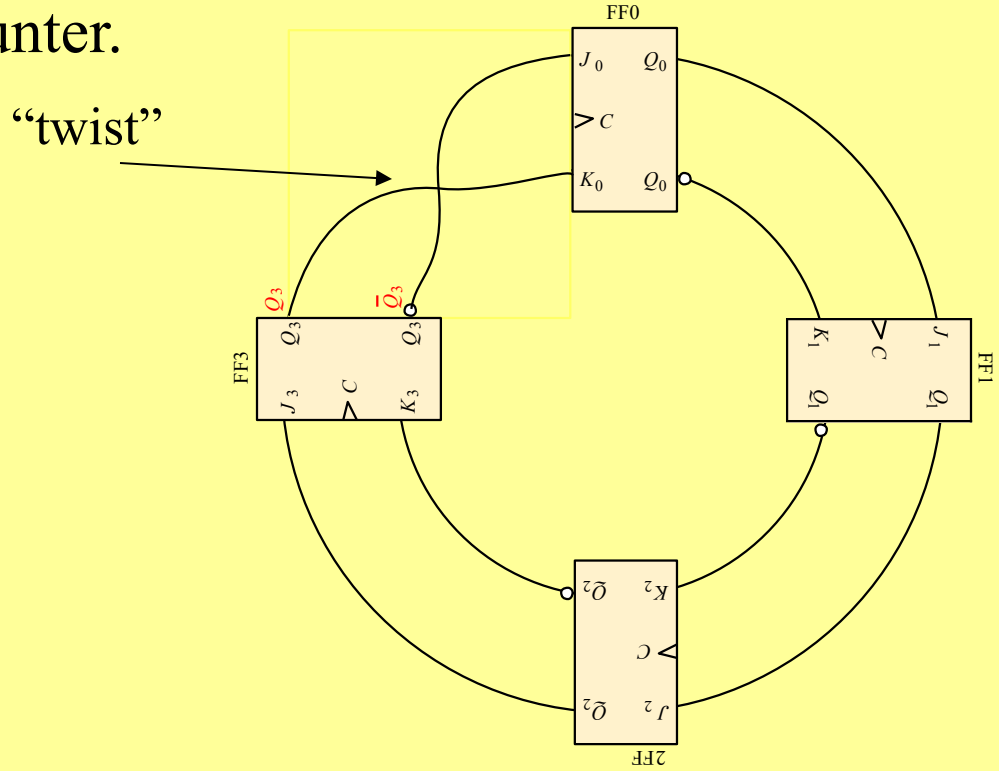
TABLE 8–4

Five-bit Johnson sequence.

Clock Pulse	Q_0	Q_1	Q_2	Q_3	Q_4
0	0	0	0	0	0
1	1	0	0	0	0
2	1	1	0	0	0
3	1	1	1	0	0
4	1	1	1	1	0
5	1	1	1	1	1
6	0	1	1	1	1
7	0	0	1	1	1
8	0	0	0	1	1
9	0	0	0	0	1

Johnson Counter

Redrawing the same Johnson counter (without the clock shown) illustrates why it is sometimes called as a “twisted-ring” counter.



Johnson Counter

The Johnson counter is useful when you need a sequence that changes by only one bit at a time, but it has a limited number of states (2_n - where n = number of stages).

The first five counts for a 4-bit Johnson counter that is initially cleared are:

CLK	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

Question

What are the remaining 3 states?

Ring Counter

A ring counter utilizes one flip-flop for each state in its sequence. It has the advantage that decoding gates are not required. In the case of a 10-bit ring counter, there is a unique output for each decimal digit.

A logic diagram for a 10-bit ring counter is shown in Figure 8–24. The sequence for this ring counter is given in Table 8–5. Initially, a 1 is preset into the first flip-flop, and the rest of the flip-flops are cleared.

Notice that the interstage connections are the same as those for a Johnson counter, except that Q rather than $\overline{Q}$ is fed back from the last stage.

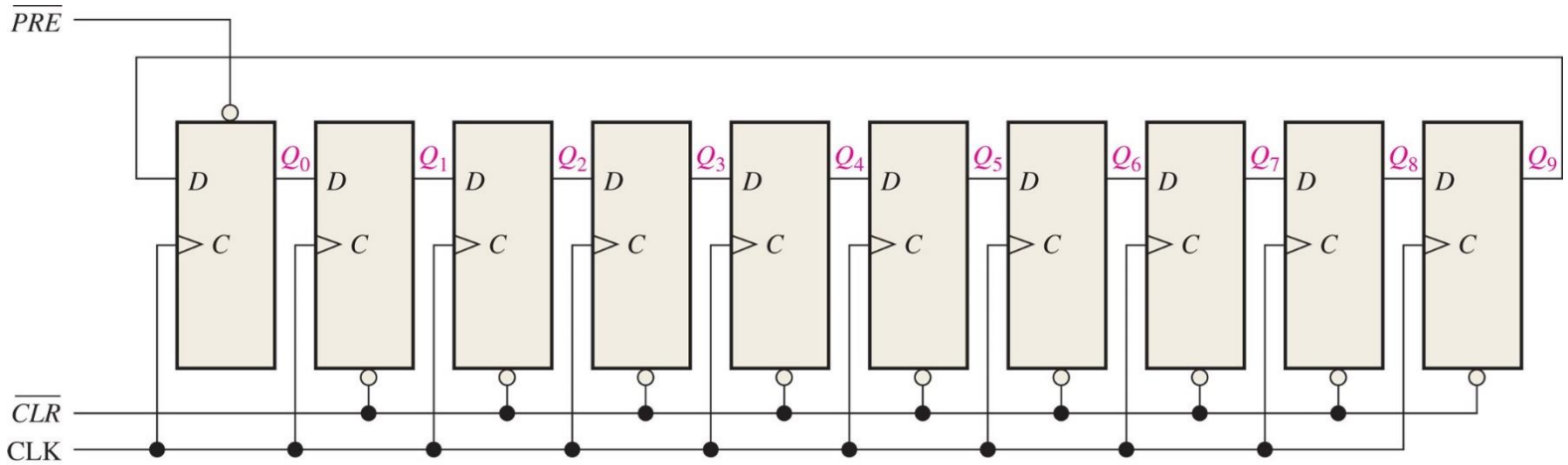


FIGURE 8-24 A 10-bit ring counter.

The ten outputs of the counter indicate directly the decimal count of the clock pulse.

For instance, a 1 on Q_0 represents a zero, a 1 on Q_1 represents a one, a 1 on Q_2 represents a two, a 1 on Q_3 represents a three, and so on. You should verify for yourself that the 1 is always retained in the counter and simply shifted "around the ring," advancing one stage for each clock pulse.

TABLE 8–5

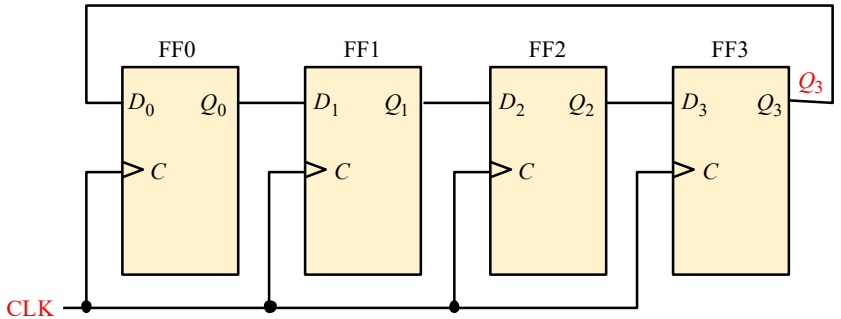
Ten-bit ring counter sequence.

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

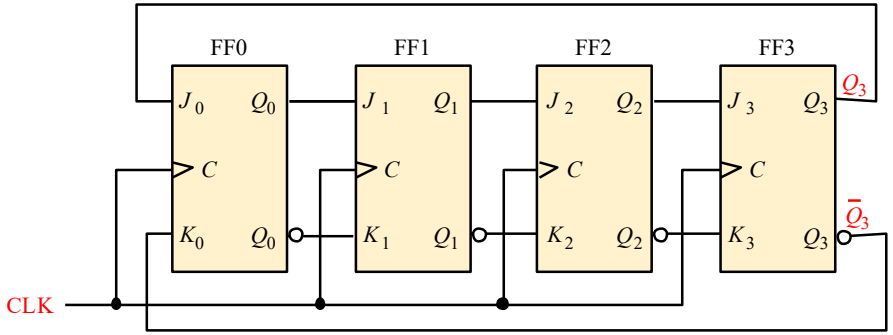
Ring Counter

The ring counter can also be implemented with either D flip-flops or J-K flip-flops.

Here is a 4-bit ring counter constructed from a series of D flip-flops. Notice the feedback.



Like the Johnson counter, it can also be implemented with J-K flip flops.

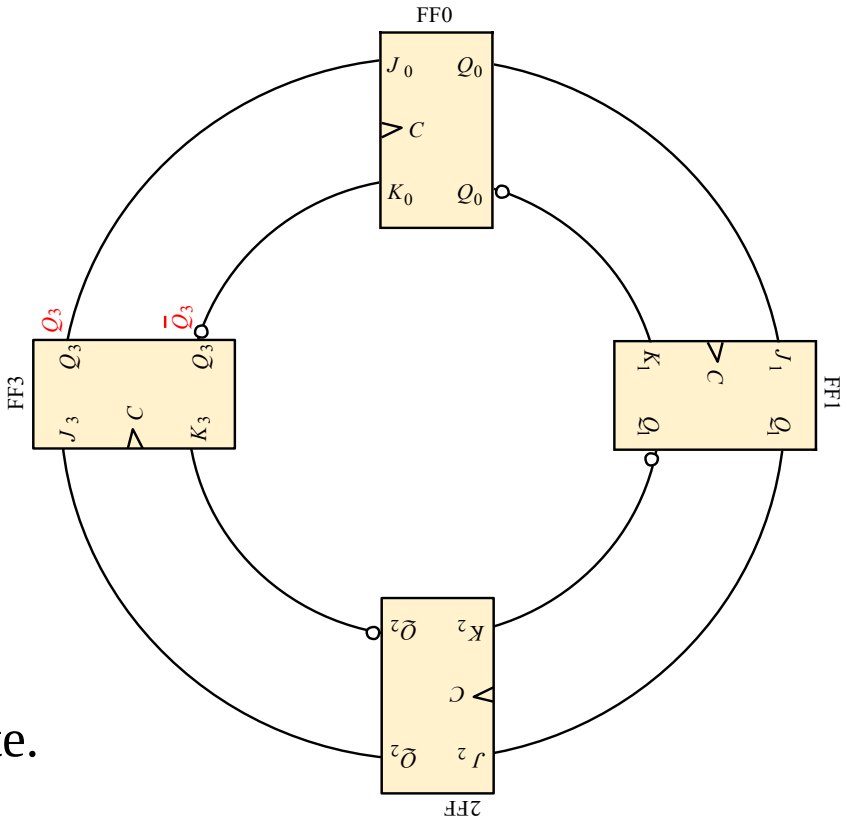


Ring Counter

Redrawing the Ring counter (without the clock shown) shows why it is a “ring”.

The disadvantage to this counter is that it must be preloaded with the desired pattern (usually a single 0 or 1) and it has even fewer states than a Johnson counter (*n*, where *n* = number of flip-flops.

On the other hand, it has the advantage of being self-decoding with a unique output for each state.



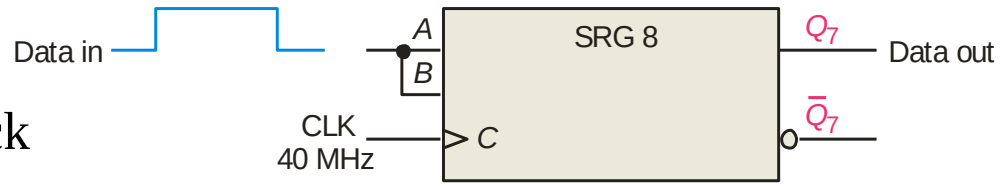
Shift Register Applications

Shift registers can be used to delay a digital signal by a predetermined amount.

Example An 8-bit serial in/serial out shift register has a 40 MHz clock. What is the total delay through the register?

Solution

The delay for each clock is 1/40 MHz = 25 ns



25 ns

The total delay is 8 x 25 ns = 200 ns

= 200 ns

When a data pulse is applied to the serial input as shown in Figure 8–26, it enters the first stage on the triggering edge of the clock pulse. It is then shifted from stage to stage on each successive clock pulse until it appears on the serial output *n* clock periods later.

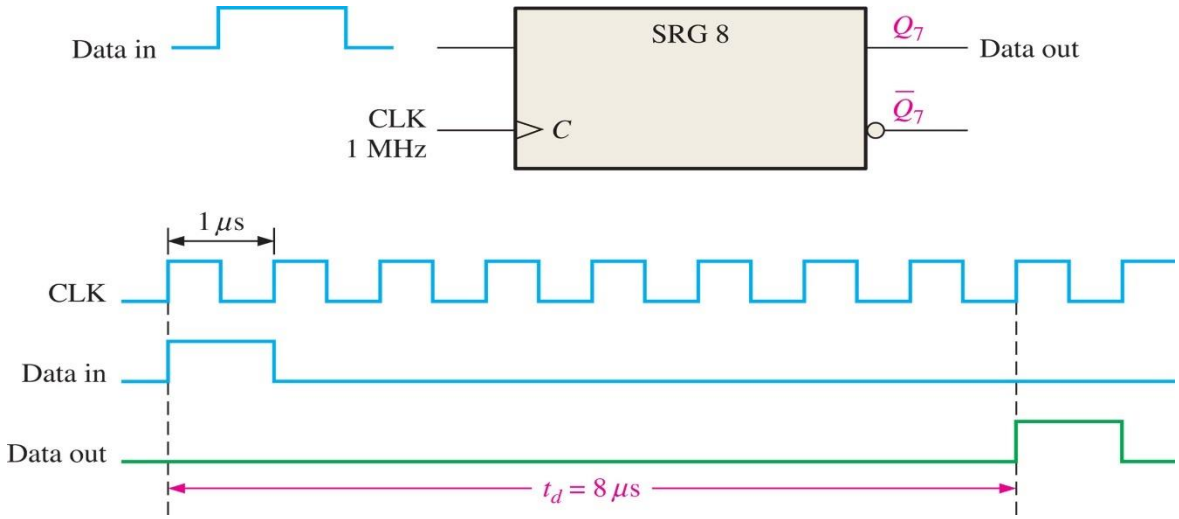


FIGURE 8-26
The shift register as a time-delay device.

This time delay operation is illustrated in Figure 8–26, in which an 8-bit serial in/serial out shift register is used with a clock frequency of 1 MHz to achieve a time delay (*td*) of 8 *us* (8 x 1 *us*).

This time can be adjusted up or down by changing the clock frequency.

The time delay can also be increased by cascading shift registers and decreased by taking the outputs from successively lower stages in the register if the outputs are available, as illustrated in Example 8–6.

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.

Figure 8–34 illustrates the UART in a general microprocessor-based system application.

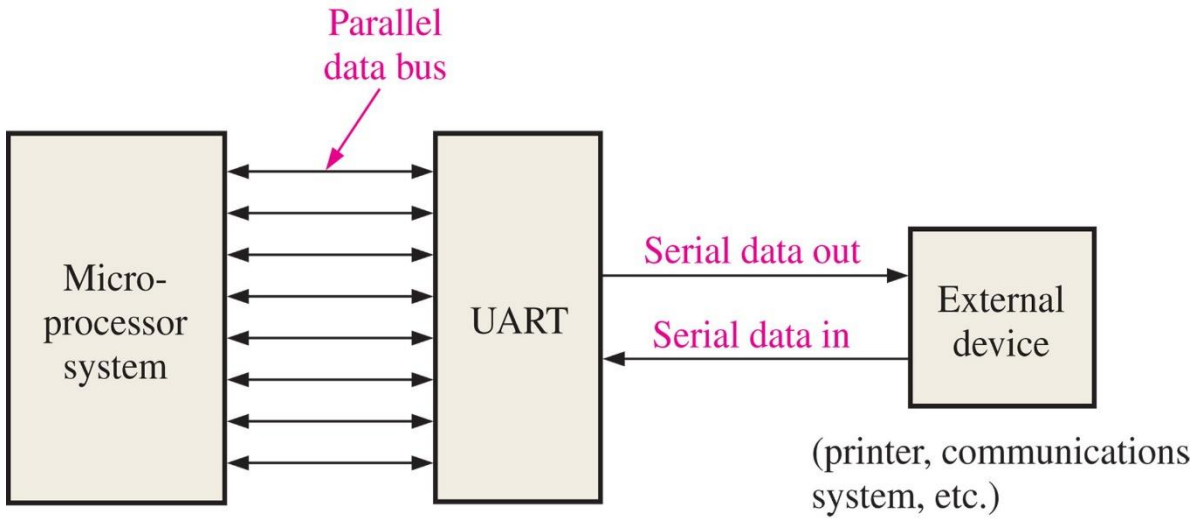
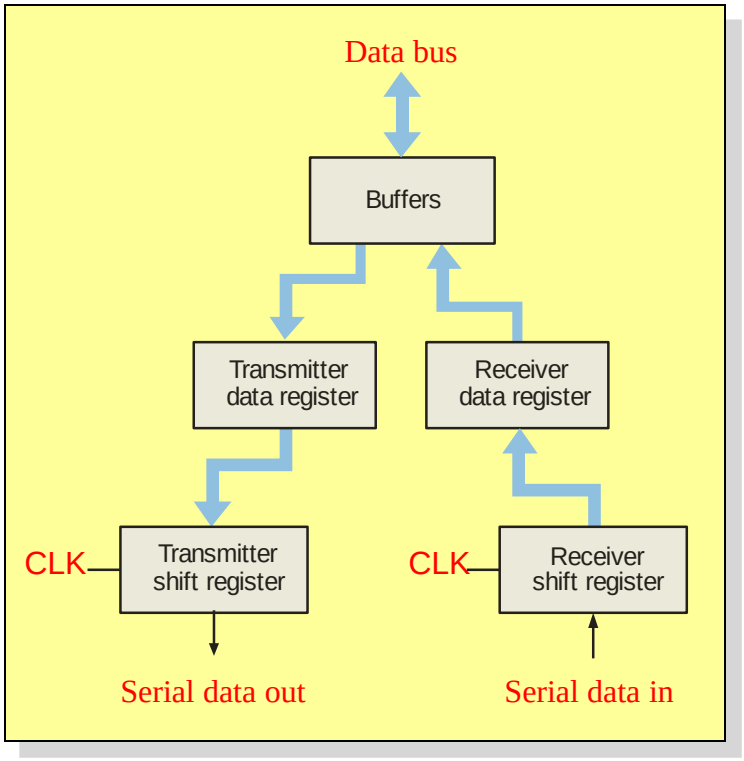


FIGURE 8-34
UART interface.

Shift Register Applications

A UART (Universal Asynchronous Receiver Transmitter) is a serial-to-parallel converter and a parallel to serial converter.

UARTs are commonly used in small systems where one device must communicate with another. Parallel data is converted to asynchronous serial form and transmitted. The serial data format is:

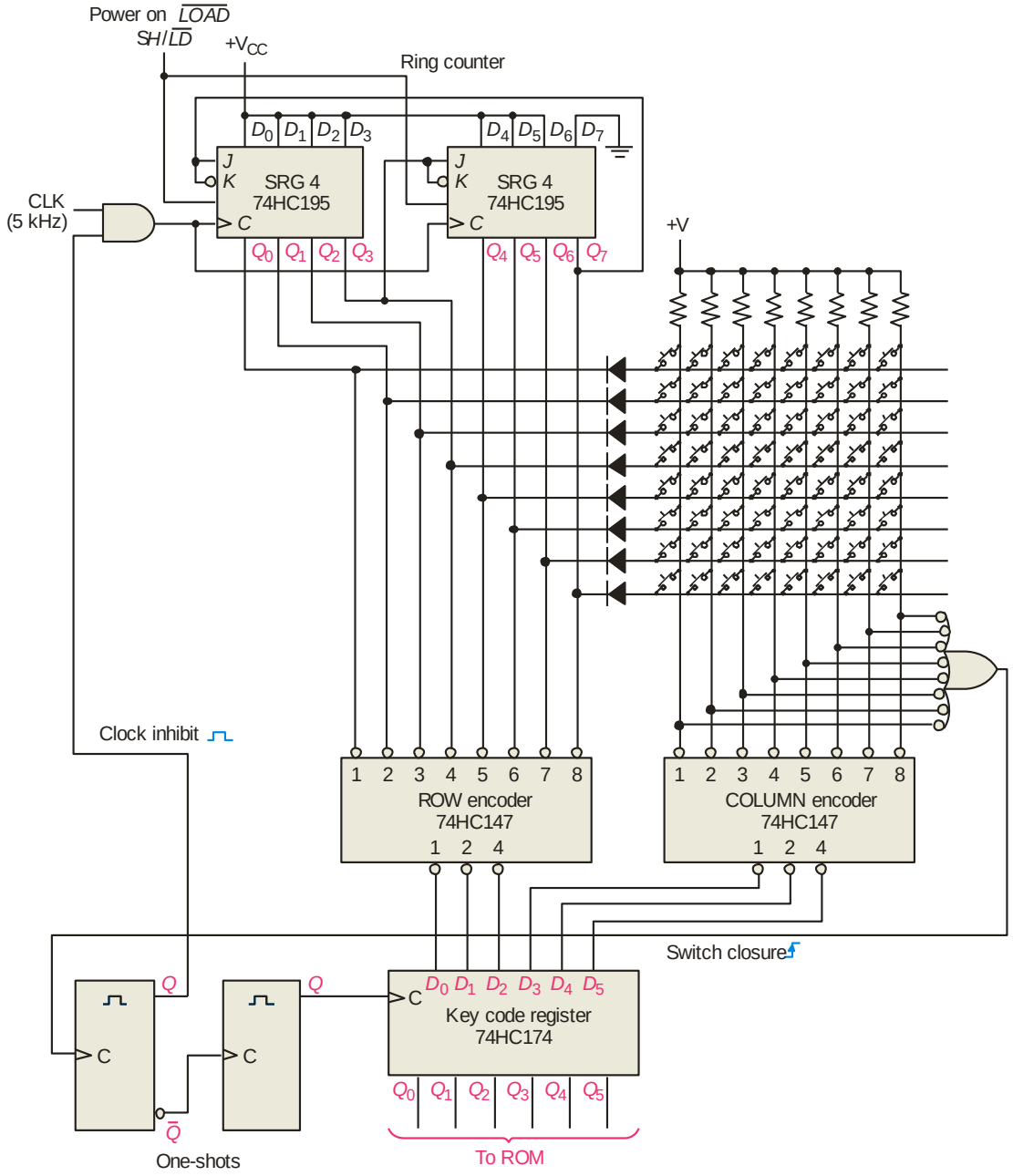


Keyboard Encoder

The keyboard encoder is an example where a ring counter is used in a small system to encode a key press.

Two 74HC195 shift registers are connected as an 8-bit ring counter preloaded with a single 0. As the 0 circulate in the ring counter, it “scans” the keyboard looking for any row that has a key closure.

When one is found, a corresponding column line is connected to that row line. The combination of the unique column and row lines identifies the key. The schematic is shown on the following slide…


Power on
SH/LD
+VCC
Ring counter
D0 D1 D2 D3
D4 D5 D6 D7
J
K
SRG 4
74HC195
C
Q0 Q1 Q2 Q3
Q4 Q5 Q6 Q7
CLK
(5 kHz)
+V
Clock inhibit
1 2 3 4 5 6 7 8
ROW encoder
74HC147
1 2 4
COLUMN encoder
74HC147
Switch closure
Q
C
D0 D1 D2 D3 D4 D5
Key code register
74HC174
Q0 Q1 Q2 Q3 Q4 Q5
To ROM
One-shots

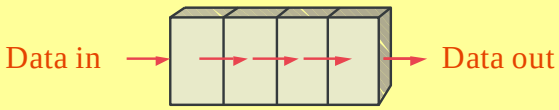
Key Terms

Register	One or more flip-flops used to store and shift data.
Stage	One storage element in a register.
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.
Load	To enter data in a shift register.
Bidirectional	Having two directions. In a bidirectional shift register, the stored data can be shifted right or left.

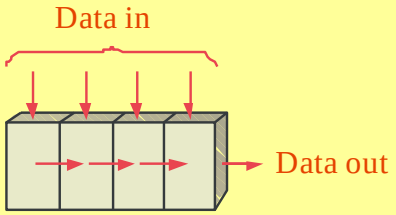
Quiz

1. The shift register that would be used to delay serial data by 4 clock periods is

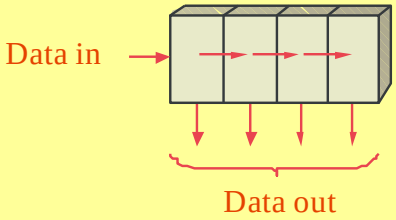
a.



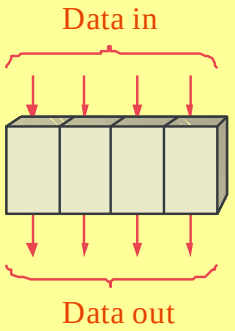
c.



b.

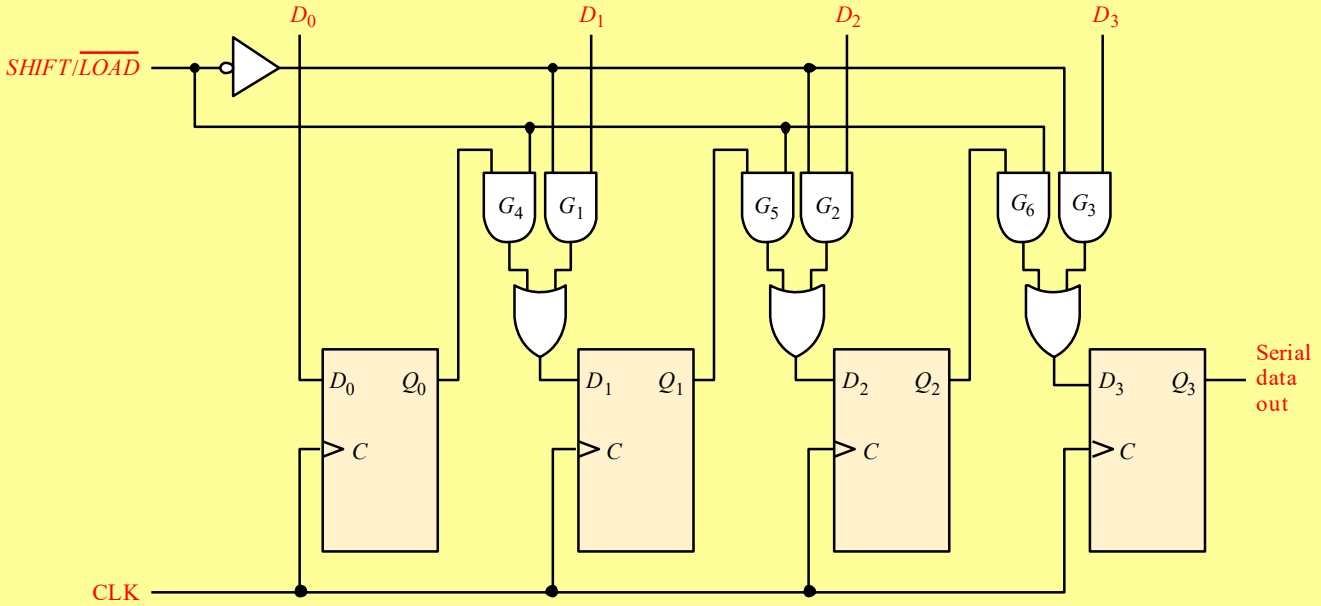


d.



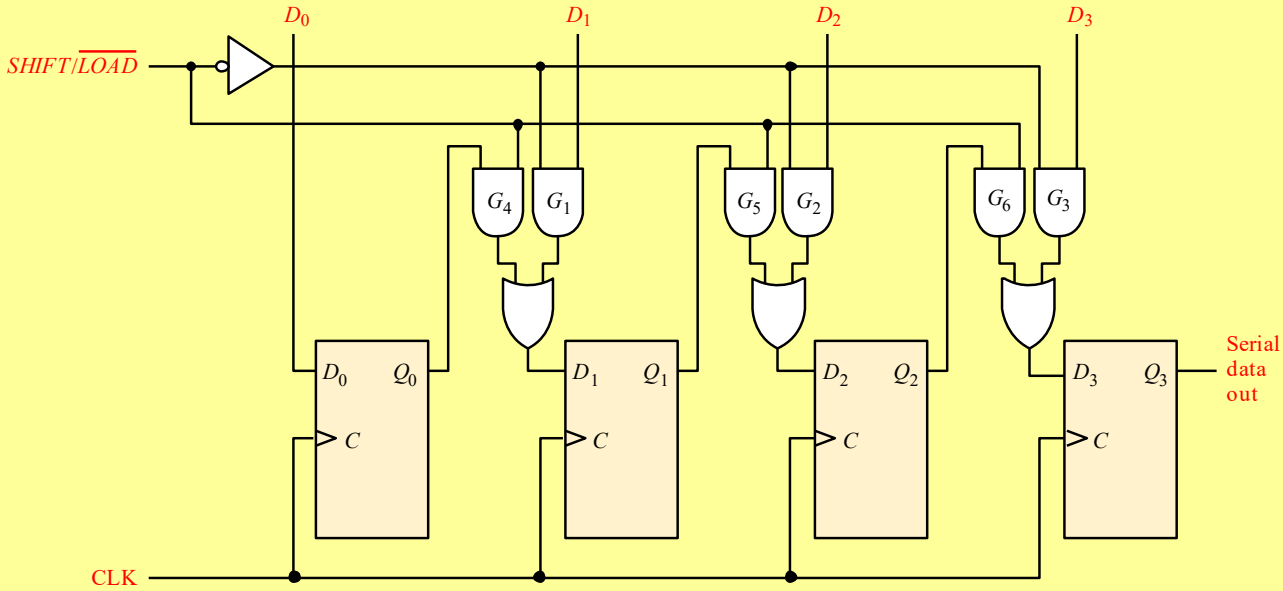
Quiz

2. The circuit shown is a
   a. serial-in/serial-out shift register
   b. serial-in/parallel-out shift register
   c. parallel-in/serial-out shift register
   d. parallel-in/parallel-out shift register



Quiz

3. If the *SHIFT/$\overline{LOAD}$* line is HIGH, data
   a. is loaded from D_0, D_1, D_2 and D_3 immediately
   b. is loaded from D_0, D_1, D_2 and D_3 on the next CLK
   c. shifted from left to right on the next CLK
   d. shifted from right to left on the next CLK



Quiz

4. A 4-bit parallel-in/parallel-out shift register will store data for
   a. 1 clock period
   b. 2 clock periods
   c. 3 clock periods
   d. 4 clock periods

Quiz

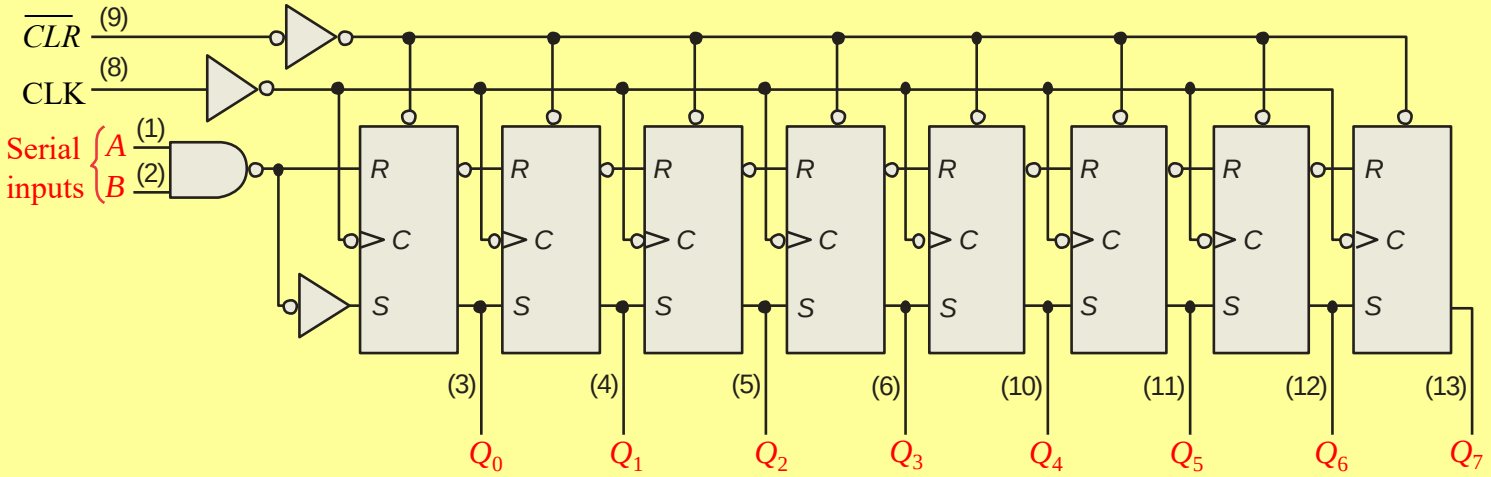
5. The 74HC164 (shown) has two serial inputs. If data is placed on the *A* input, the *B* input

a. could serve as an active LOW enable

b. could serve as an active HIGH enable

c. should be connected to ground

d. should be left open



Quiz

6. An advantage of a ring counter over a Johnson counter is that the ring counter
   a. has more possible states for a given number of flip-flops
   b. is cleared after each cycle
   c. allows only one bit to change at a time
   d. is self-decoding

Quiz

7. A possible sequence for a 4-bit ring counter is

   a. … 1111, 1110, 1101 …

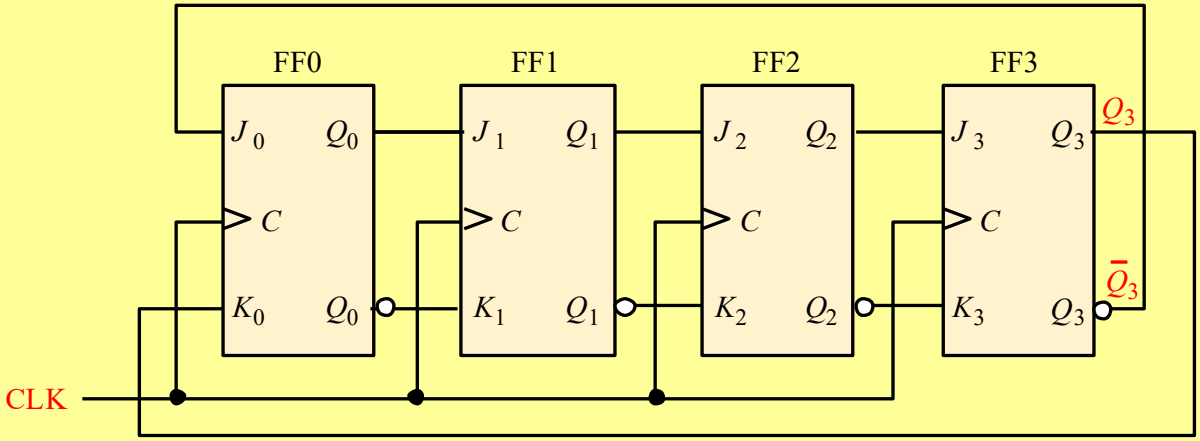
   b. … 0000, 0001, 0010 …

   c. … 0001, 0011, 0111 …

   d. … 1000, 0100, 0010 …

Quiz

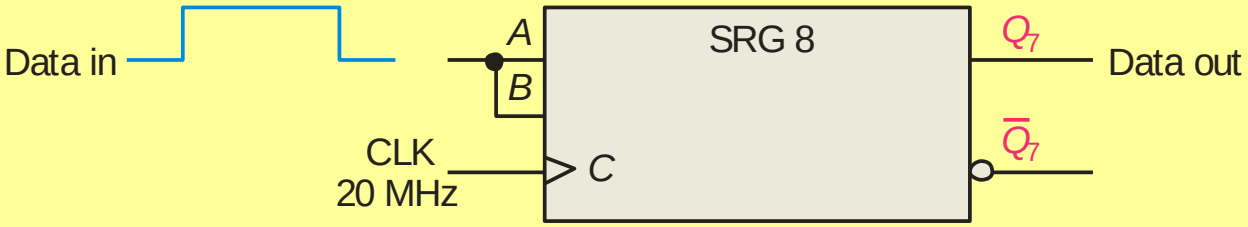
8. The circuit shown is a
   a. serial-in/parallel-out shift register
   b. serial-in/serial-out shift register
   c. ring counter
   d. Johnson counter



Quiz

9. Assume serial data is applied to the 8-bit shift register shown. The clock frequency is 20 MHz. The first data bit will show up at the output in

   a. 50 ns
   b. 200 ns
   c. 400 ns
   d. 800 ns



Quiz

10. For transmission, data from a UART is sent in

a. asynchronous serial form

b. synchronous parallel form

c. can be either of the above

d. none of the above

Quiz

Answers:

1. a
2. c
3. c
4. a
5. b
6. d
7. d
8. d
9. c
10. a