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

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 of the most common types of shift register counters, the Johnson counter and the ring counter, are introduced in this section.

After completing this section, you should be able to

■ Discuss how a shift register counter differs from a basic shift register ■ Explain the operation of a Johnson counter ■ Specify a Johnson sequence for any number of bits ■ Explain the operation of a ring counter and determine the sequence of any specific ring counter

The 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). This feedback arrangement produces a characteristic sequence of states, as shown in Table 9–1 for a 4-bit device and in Table 9–2 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 $2n$, where n is the number of stages in the counter.

The implementations of the 4-stage and 5-stage Johnson counters are shown in Figure 9–23. The implementation of a Johnson counter is very straightforward and is the same regardless of the number of stages. The Q output of each stage is connected to the D input of the next stage (assuming that D flip-flops are used). The single exception is that the $\overline{Q}$ output of the last stage is connected back to the D input of the first stage. As the sequences in Table 9–1 and 9–2 show, the counter will "fill up" with 1s from left to right, and then it will "fill up" with 0s again.

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

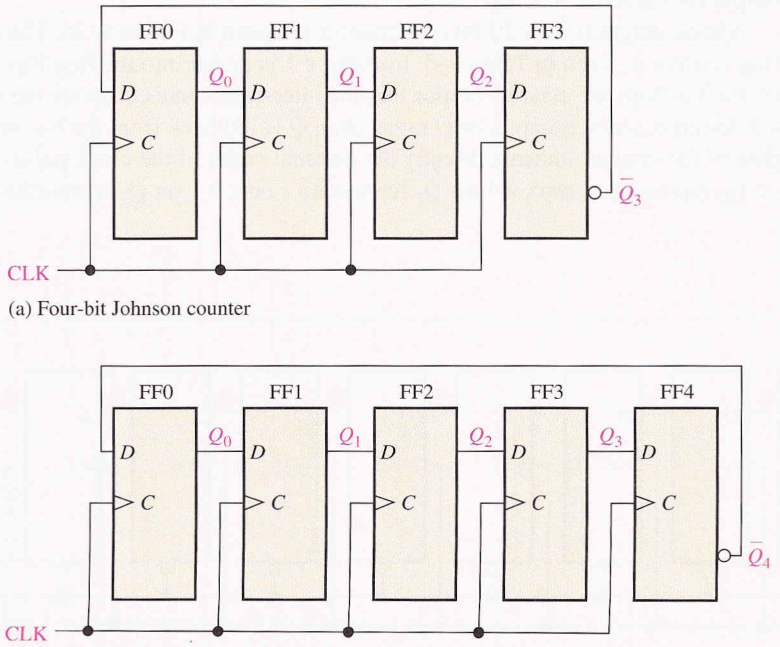
◀ **TABLE 9–1**

Four-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

◀ **TABLE 9–2**

Five-bit Johnson sequence.



(a) Four-bit Johnson counter

(b) Five-bit Johnson counter

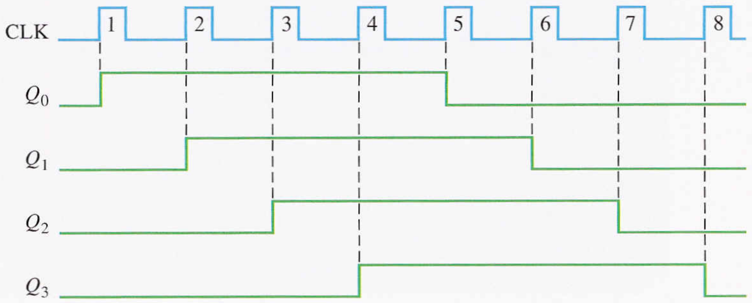
◀ **FIGURE 9–23**

Four-bit and 5-bit Johnson counters.

Diagrams of the timing operations of the 4-bit and 5-bit counters are shown in Figures 9–24 and 9–25, respectively.

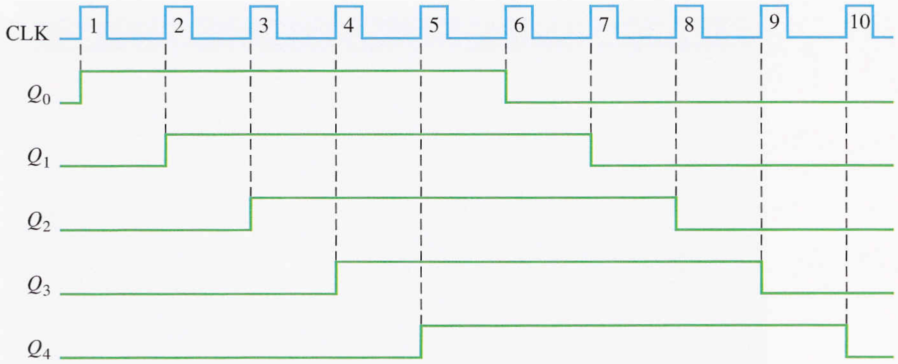
▶ FIGURE 9–24

Timing sequence for a 4-bit Johnson counter.



▶ FIGURE 9–25

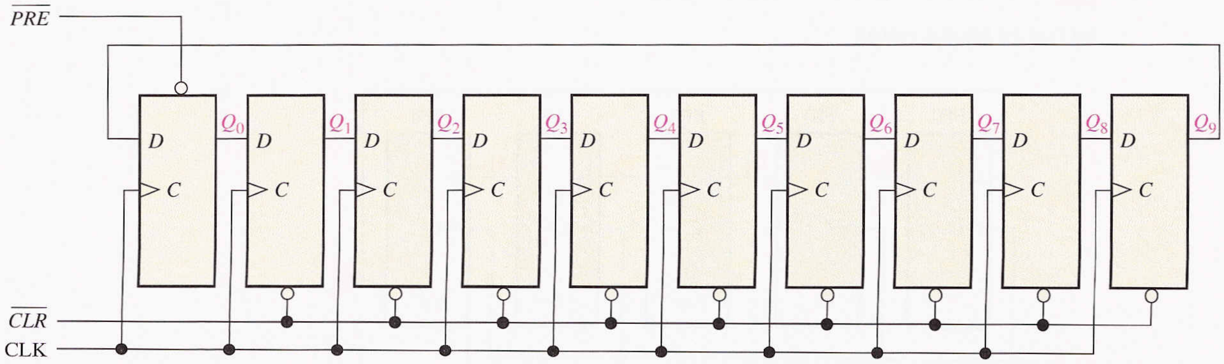
Timing sequence for a 5-bit Johnson counter.

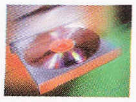


The Ring Counter

The **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 9–26. The sequence for this ring counter is given in Table 9–3. 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. 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





▲ FIGURE 9–26

A 10-bit ring counter. Open file F09-26 to verify operation.

TABLE 9–3

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

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.

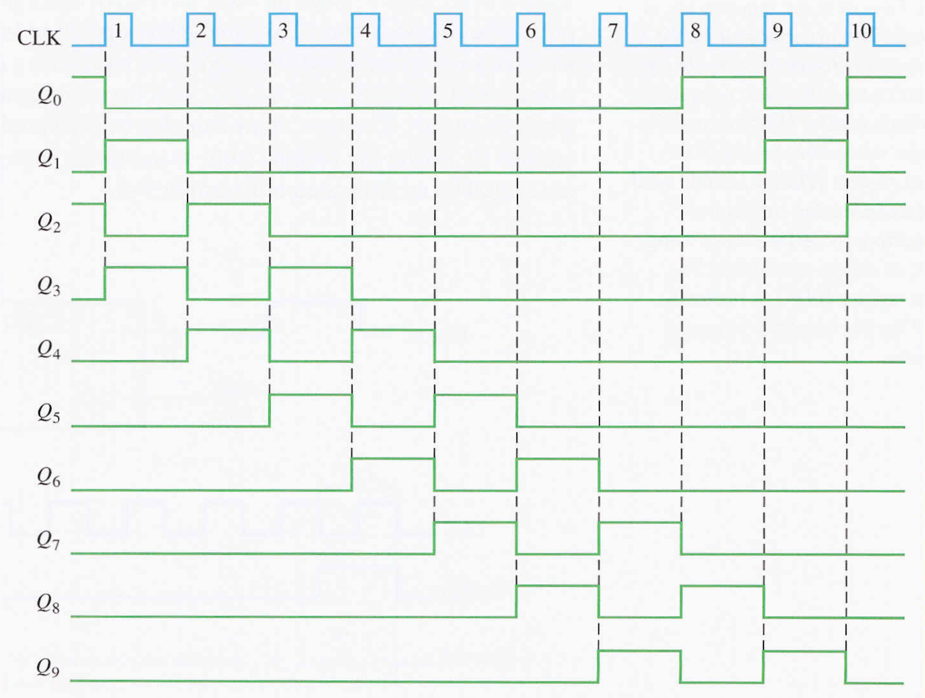
Modified sequences can be achieved by having more than a single 1 in the counter, as illustrated in Example 9–5.

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.

Solution See Figure 9–27.

FIGURE 9–27



Related Problem If a 10-bit ring counter has an initial state 0101001111, determine the waveform for each Q output.

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

Shift registers are found in many types of applications, a few of which are presented in this section.

After completing this section, you should be able to

■ Use a shift register to generate a time delay ■ Implement a specified ring counter sequence using a 74HC195 shift register ■ Discuss how shift registers are used for serial-to-parallel conversion of data ■ Define *UART* ■ Explain the operation of a keyboard encoder and how registers are used in this application

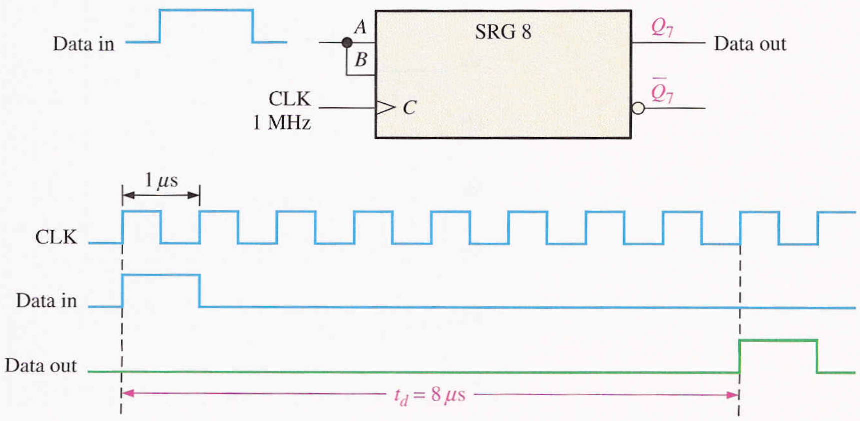
COMPUTER NOTE

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.

Time Delay

The serial in/serial out shift register can be used to provide a time delay from input to output that is a function of both the number of stages (n) in the register and the clock frequency.

When a data pulse is applied to the serial input as shown in Figure 9–28 (A and B connected together), 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. This time-delay operation is illustrated in Figure 9–28, 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 (t_d) of 8 μs (8 × 1 μs). 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 9–6.



▲ **FIGURE 9–28**

The shift register as a time-delay device.