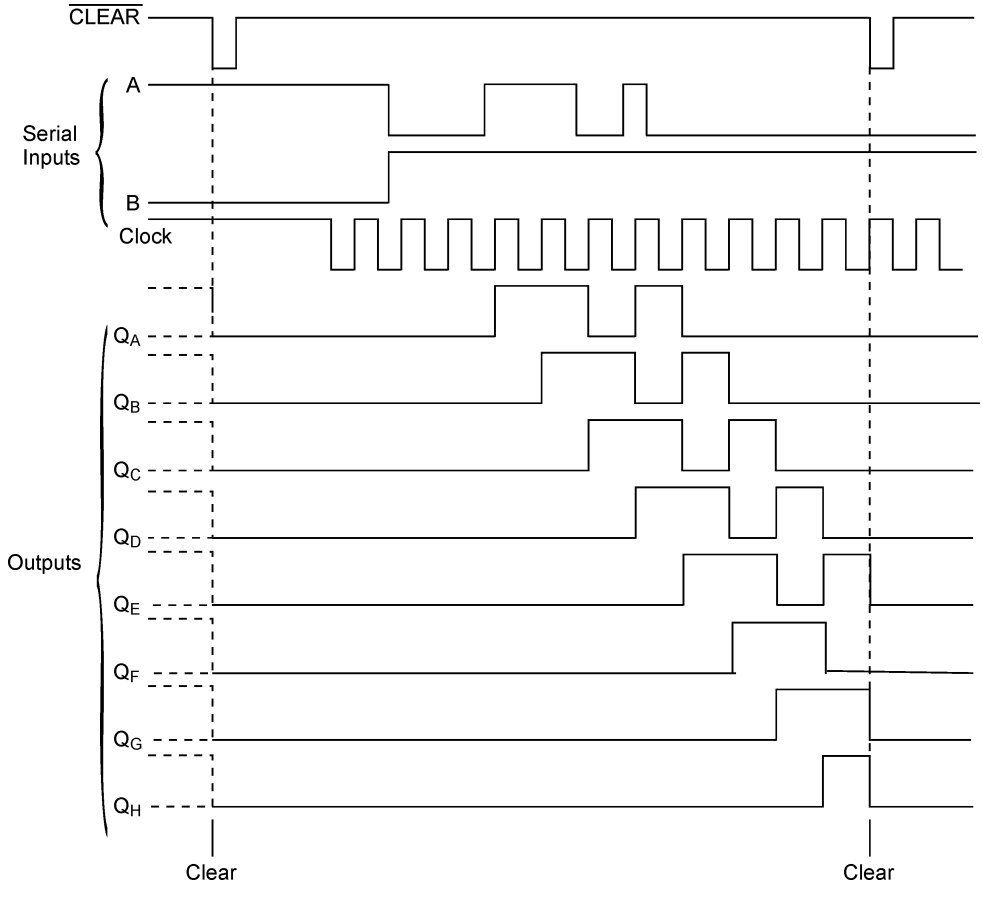


Figure 11.39 Timing waveforms of IC 74164.

11.12.4 Parallel-In Parallel-Out Shift Register

The hardware of a parallel-in parallel-out shift register is similar to that of a parallel-in serial-out shift register. If in a parallel-in serial-out shift register the outputs of different flip-flops are brought out, it becomes a parallel-in parallel-out shift register. In fact, the logic diagram of a parallel-in parallel-out shift register is similar to that of a parallel-in serial-out shift register. As an example, IC 74199 is an eight-bit parallel-in parallel-out shift register. Figure 11.42 shows its logic diagram. We can see that the logic diagram of IC 74199 is similar to that of IC 74166 mentioned in the previous section, except that in the case of the former the flip-flop outputs have been brought out on the IC terminals.

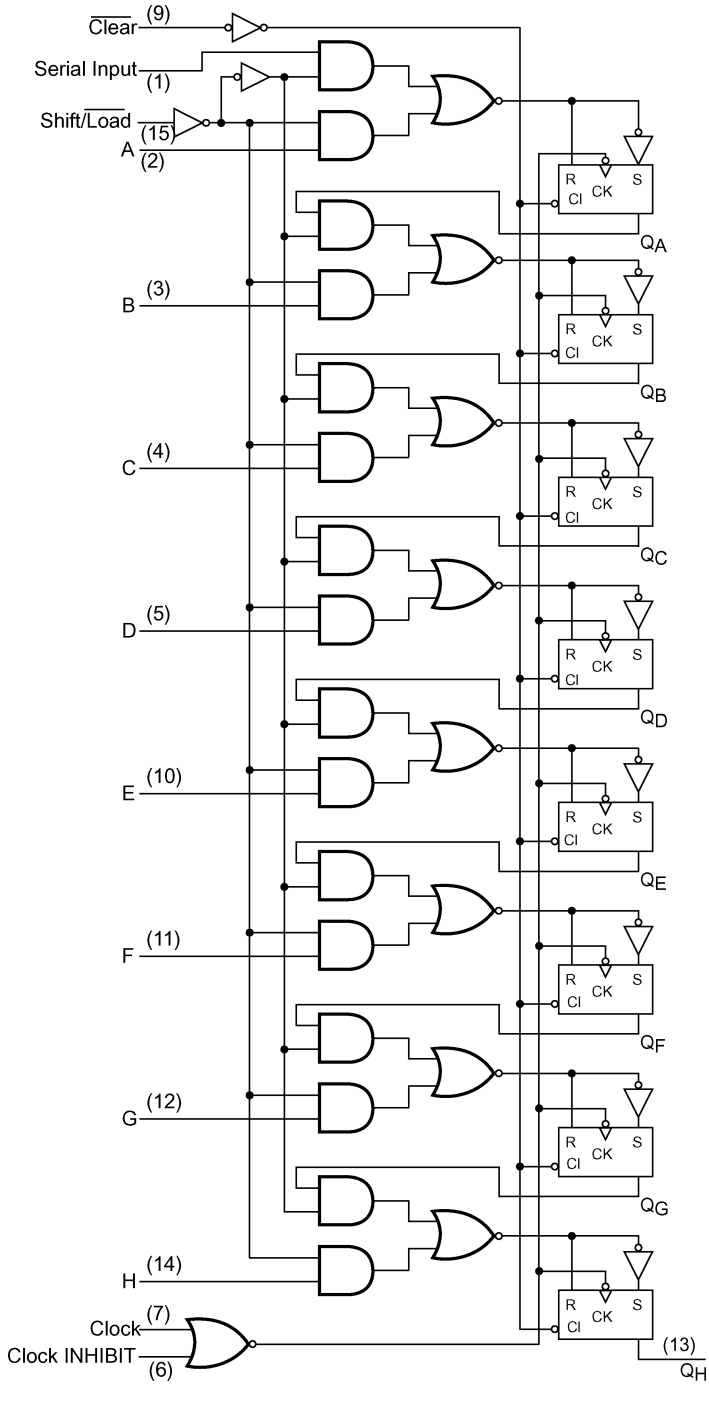


Figure 11.40 Logic diagram of 74166.

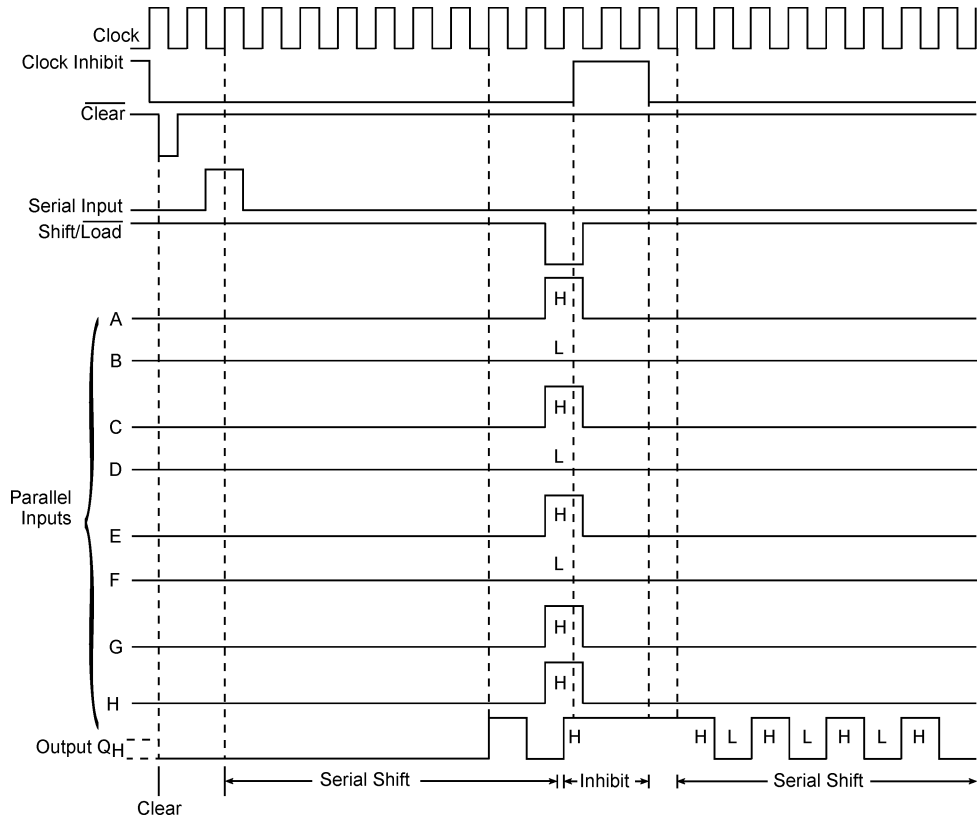


Figure 11.41 Timing waveforms of IC 74166.

11.12.5 Bidirectional Shift Register

A bidirectional shift register allows shifting of data either to the left or to the right. This is made possible with the inclusion of some gating logic having a control input. The control input allows shifting of data either to the left or to the right, depending upon its logic status.

11.12.6 Universal Shift Register

A universal shift register can be made to function as any of the four types of register discussed in previous sections. That is, it has serial/parallel data input and output capability, which means that it can function as serial-in serial-out, serial-in parallel-out, parallel-in serial out and parallel-in parallel-out shift registers.

IC 74194 is a common four-bit bidirectional universal shift register. Figure 11.43 shows the logic diagram of Ic 74194. the device offers four modes of operation, namely (a) inhibit clock, (b) shift right, (c) shift left and (d) parallel load. Clocking of the device is inhibited when both the mode control inputs S_1 and S_0 are in the logic LOW state. shift right and shift left operations are accomplished

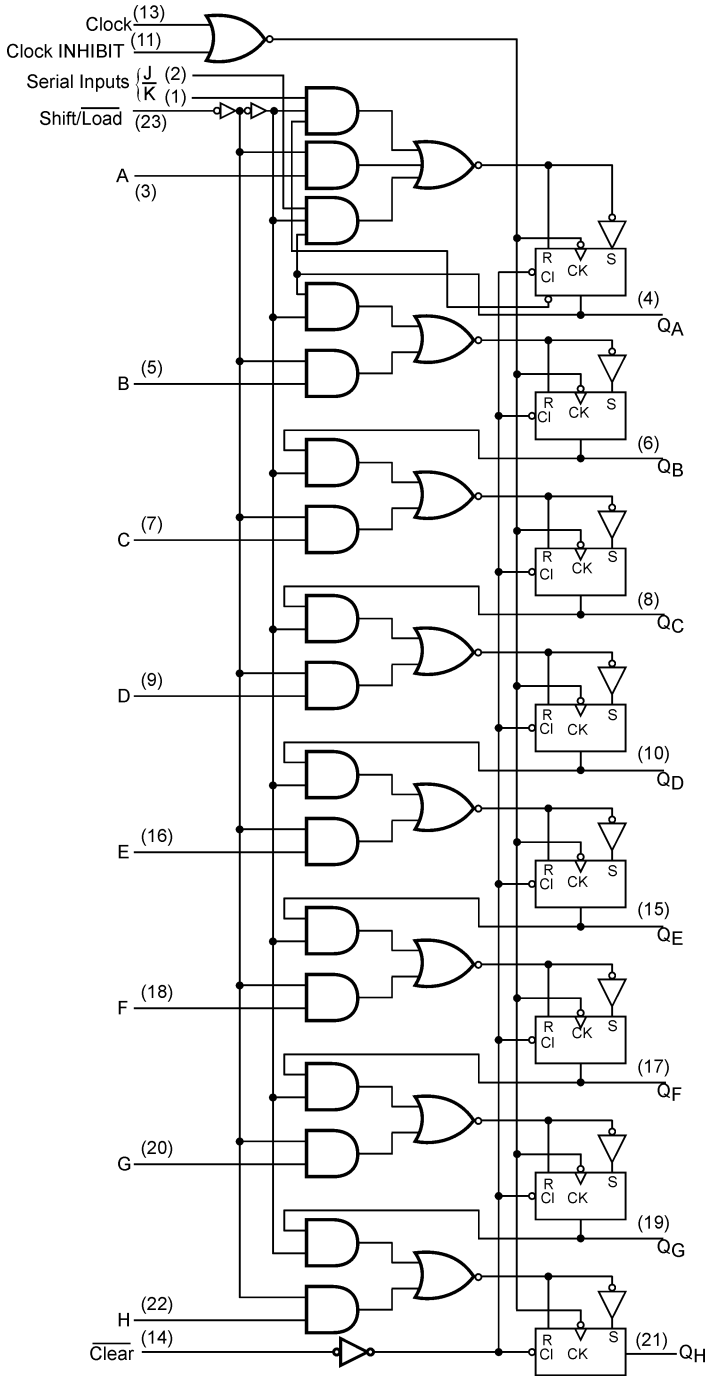


Figure 11.42 Logic diagram of IC 74199.

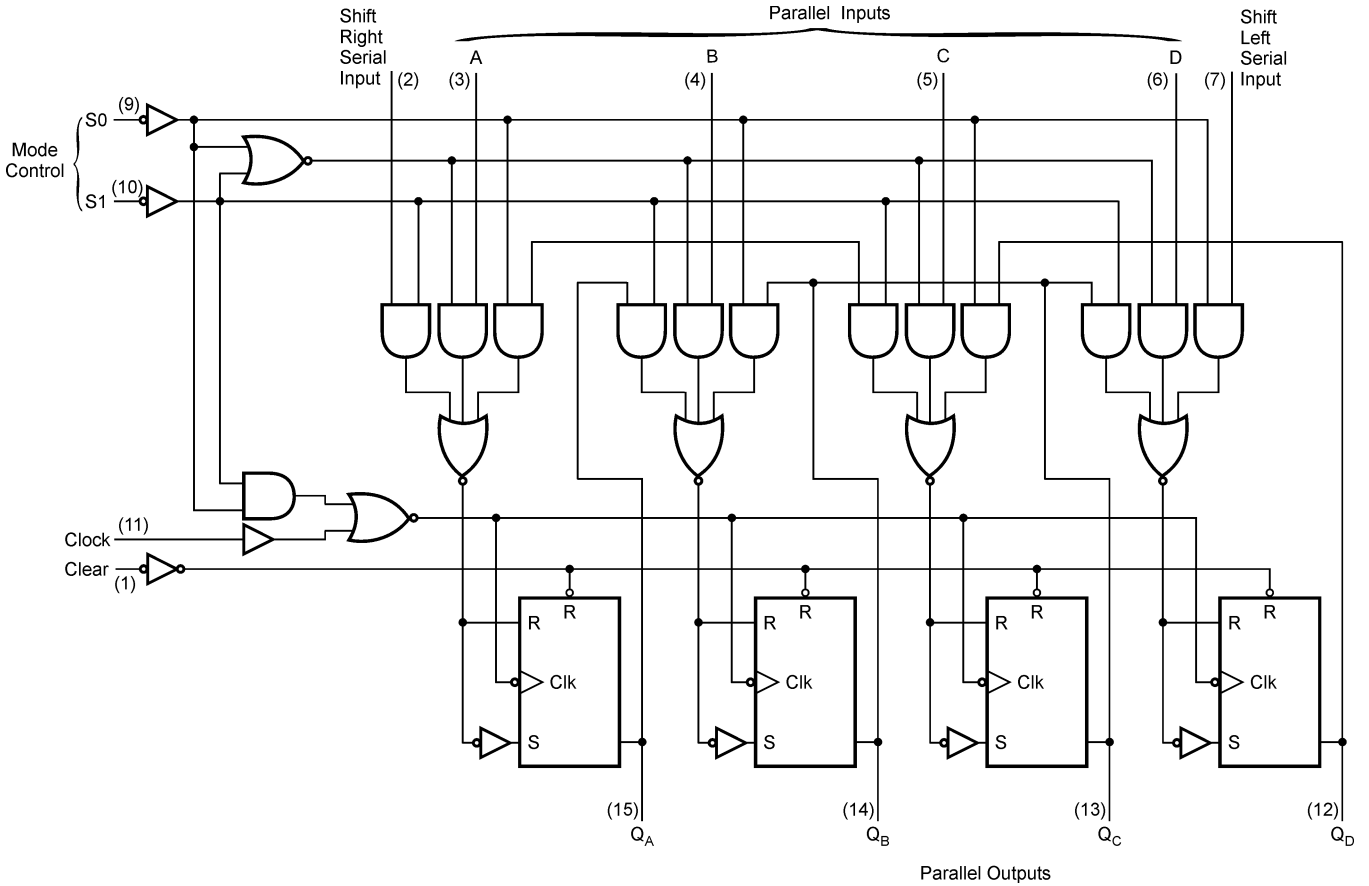


Figure 11.43 Logic diagram of IC 74194.

synchronously with LOW-to-HIGH transition of the clock with S_1 LOW and S_0 HIGH (for shift right) and S_1 HIGH and S_0 LOW (for shift left). Serial data are entered in the case of shift right and shift left operations at the corresponding data input terminals. Parallel loading is also accomplished synchronously with LOW-to-HIGH clock transitions by applying four bits of data and then driving the mode control inputs S_1 and S_0 to the logic HIGH state. Data are loaded into corresponding flip-flops and appear at the outputs with LOW-to-HIGH clock transition. Serial data flow is inhibited during parallel loading. Different modes of operation are apparent in the timing waveforms of Fig. 11.44.

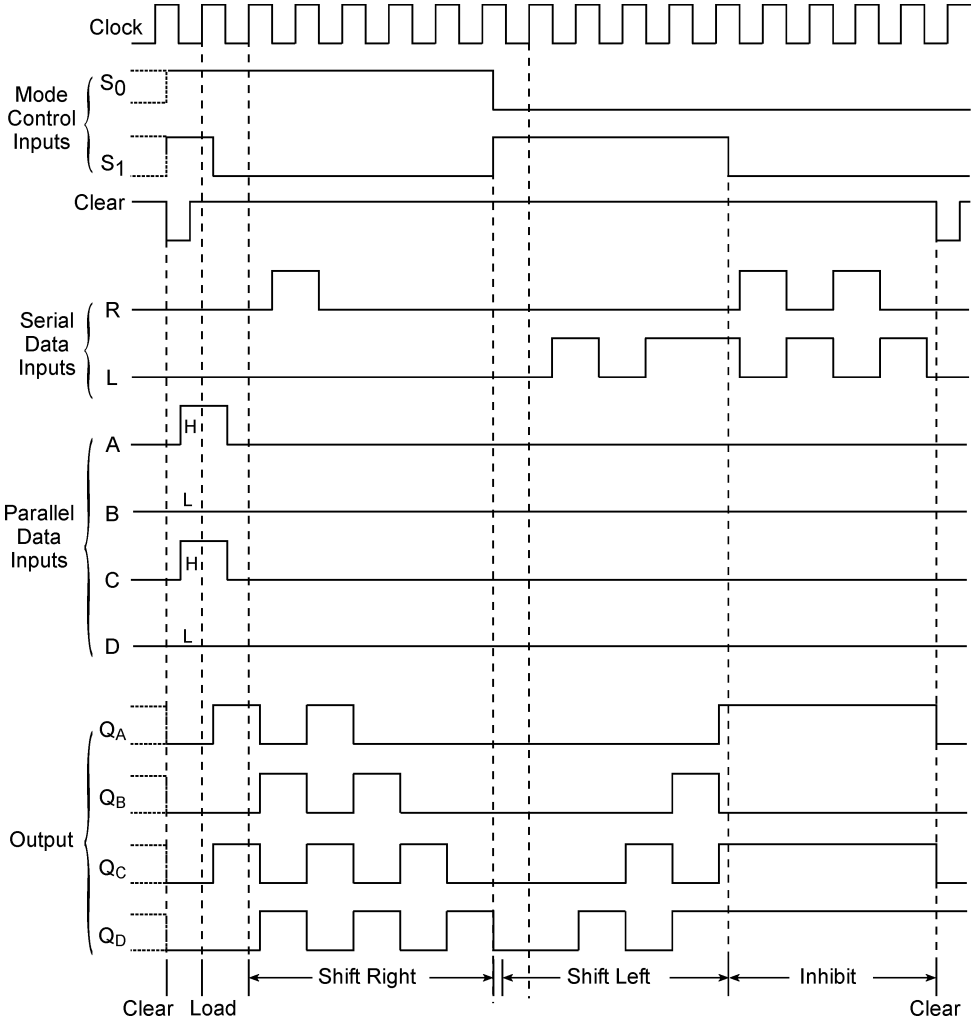


Figure 11.44 Timing waveforms of IC 74194.

11.13 Shift Register Counters

We have seen that both counters and shift registers are some kinds of cascade arrangement of flip-flops. A shift register, unlike a counter, has no specified sequence of states. However, if the serial output of the shift register is fed back to the serial input, we do get a circuit that exhibits a specified sequence of states. The resulting circuits are known as *shift register counters*. Depending upon the nature of the feedback, we have two types of shift register counter, namely the *ring counter* and the *shift counter*, also called the *Johnson counter*. These are briefly described in the following paragraphs.

11.13.1 Ring Counter

A *ring counter* is obtained from a shift register by directly feeding back the true output of the output flip-flop to the data input terminal of the input flip-flop. If *D* flip-flops are being used to construct the shift register, the ring counter, also called a circulating register, can be constructed by feeding back the *Q* output of the output flip-flop back to the *D* input of the input flip-flop. If *J-K* flip-flops are being used, the *Q* and $\overline{Q}$ outputs of the output flip-flop are respectively fed back to the *J* and *K* inputs of the input flip-flop. Figure 11.45 shows the logic diagram of a four-bit ring counter. Let us assume that flip-flop FF0 is initially set to the logic '1' state and all other flip-flops are reset to the logic '0' state. The counter output is therefore 1000. With the first clock pulse, this '1' gets shifted to the second flip-flop output and the counter output becomes 0100. Similarly, with the second and third clock pulses, the counter output will become 0010 and 0001. With the fourth clock pulse, the counter output will again become 1000. The count cycle repeats in the subsequent clock pulses. Circulating registers of this type find wide application in the control section of microprocessor-based systems where one event should follow the other. The timing waveforms for the circulating register of Figure 11.45, as shown in Fig. 11.46, further illustrate their utility as a control element in a digital system to generate control pulses that must occur one after the other sequentially.

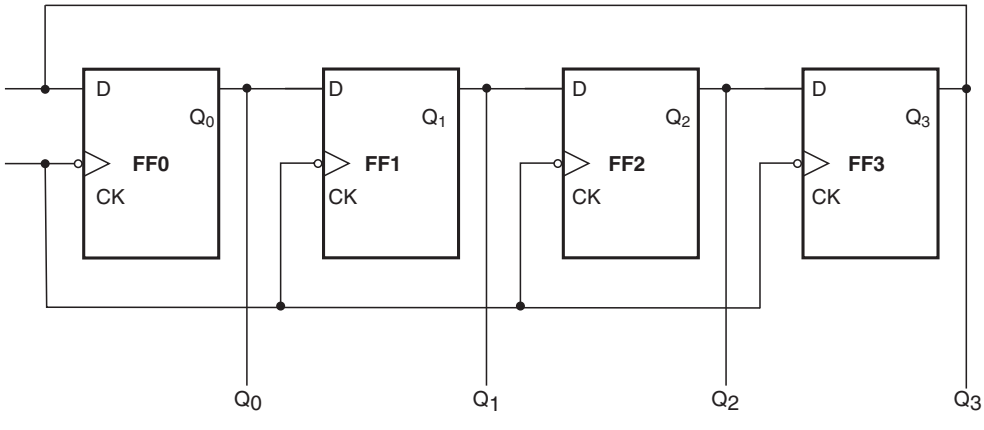


Figure 11.45 Four-bit ring counter.

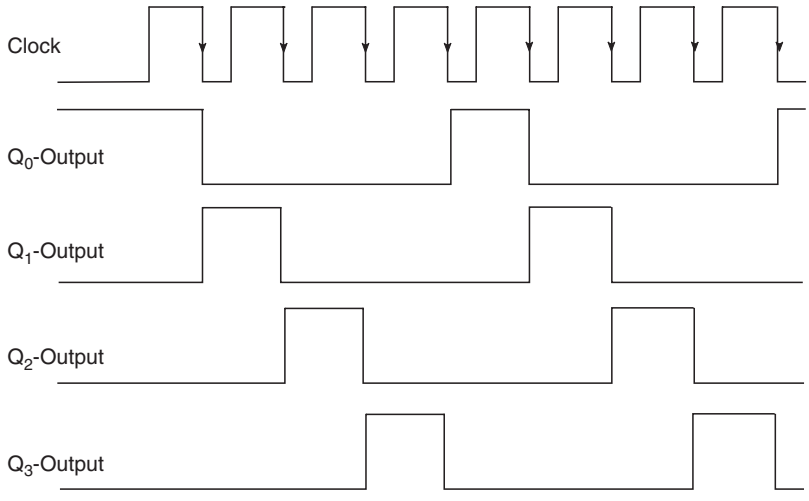


Figure 11.46 Timing waveforms of the four-bit ring counter.

11.13.2 Shift Counter

A *shift counter* on the other hand is constructed by having an inverse feedback in a shift register. For instance, if we connect the Q output of the output flip-flop back to the K input of the input flip-flop and the $\overline{Q}$ output of the output flip-flop to the J input of the input flip-flop in a serial shift register, the result is a shift counter, also called a *Johnson counter*. If the shift register employs D flip-flops, the $\overline{Q}$ output of the output flip-flop is fed back to the D input of the input flip-flop. If R-S flip-flops are used, the Q output goes to the R input and the $\overline{Q}$ output is connected to the S input. Figure 11.47 shows the logic diagram of a basic four-bit shift counter.

Let us assume that the counter is initially reset to all 0s. With the first clock cycle, the outputs will become 1000. With the second, third and fourth clock cycles, the outputs will respectively be 1100, 1110 and 1111. The fifth clock cycle will change the counter output to 0111. The sixth, seventh and eighth clock pulses successively change the outputs to 0011, 0001 and 0000. Thus, one count cycle

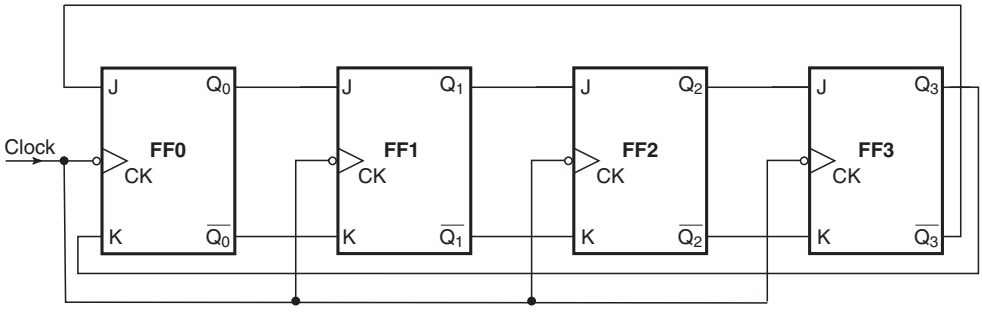


Figure 11.47 Four-bit shift counter.

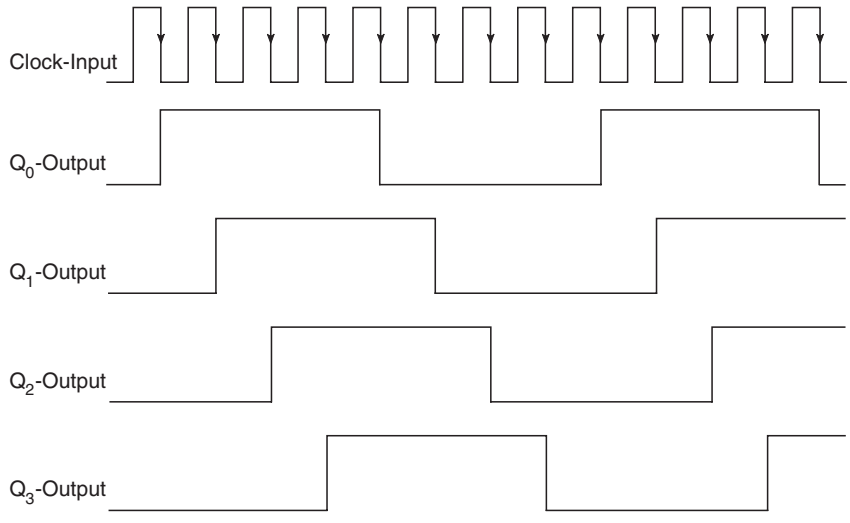


Figure 11.48 Timing waveforms of the shift counter.

is completed in eight cycles. Figure 11.48 shows the timing waveforms. Different output waveforms are identical except for the fact that they are shifted from the immediately preceding one by one clock cycle. Also, the time period of each of these waveforms is 8 times the period of the clock waveform. That is, this shift counter behaves as a divide-by-8 circuit.

In general, a shift counter comprising n flip-flops acts as a divide-by-$2n$ circuit. Shift counters can be used very conveniently to construct counters having a modulus other than the integral power of 2.

Example 11.10

Refer to Fig. 11.49, which shows an application circuit of eight-bit serial-in serial-out shift register type IC 7491 along with the waveform applied at the shorted A and B inputs:

(a) What will be the data bit present at the output at the end of the eleventh LOW-to-HIGH transition of the clock waveform?
(b) If there is a logic '1' at the end of the nth LOW-to-HIGH clock transition at the Q_3 output, what will the Q_5 output at the end of the $(n+2)$th transition be?

Solution

(a) At the end of the eighth LOW-to-HIGH clock transition, the data bits loaded into the register will be 10110010, with the '0' on the extreme right appearing at the Q_7 output (refer to the logic diagram of IC 7491 shown in Fig. 11.37). The ninth clock transition will shift this '0' out of the register, and the next adjacent bit (that is, '1') will take its place on the Q_7 output. Each subsequent clock pulse will shift the bits one step towards the right, with the result that at the end of the eleventh clock transition the Q_7 output will be a logic '0'.
(b) It will be a logic '1' only. The Q_3 output will be shifted two bit positions to the right by two clock transitions.

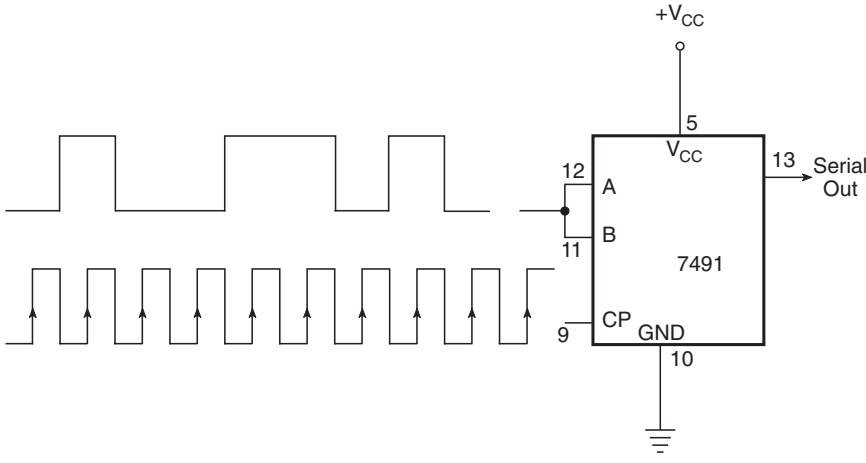


Figure 11.49 Example 11.10.

Example 11.11

Determine the number of flip-flops required to construct (a) a MOD-10 ring counter and (b) a MOD-10 Johnson counter. Also, write the count sequence in the two cases.

Solution

(a) The modulus of a ring counter is the same as the number of bits (or flip-flops). Therefore, the number of flip-flops required = 10. The count sequence is 1000000000, 0100000000, 0010000000, 0001000000, 0000100000, 0000010000, 0000001000, 0000000100, 0000000010, 0000000001 and back to 1000000000.

(b) The modulus of a Johnson counter is twice the number of flip-flops. Therefore, the number of flip-flops = 5. The count sequence is 00000, 10000, 11000, 11100, 11110, 11111, 01111, 00111, 00011, 00001 and back to 00000.

Example 11.12

Refer to the logic circuit of Fig. 11.50. Determine the modulus of this counter and write its counting sequence.

Solution

The LSB of the five-bit ring counter feeds the clock input of the *J-K* flip-flop that has been wired as a toggle flip-flop. The ring counter has a modulus of 5, and the *J-K* flip-flop works like a divide-by-2 circuit. The modulus of the counter circuit obtained by the cascade arrangement of the two is therefore 10. The counting sequence of this arrangement is given in Table 11.15.

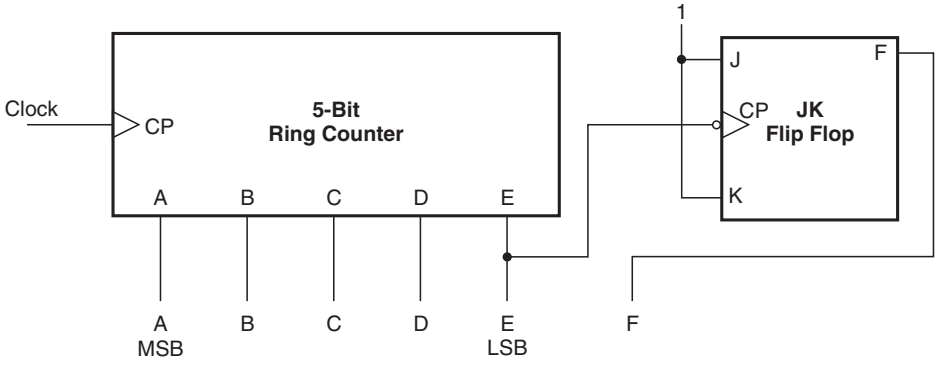


Figure 11.50 Example 11.12.

Table 11.15 Example 11.11.

Clock pulse	Outputs					
	A	*B*	*C*	*D*	*E*	*F*
1	1	0	0	0	0	0
2	0	1	0	0	0	0
3	0	0	1	0	0	0
4	0	0	0	1	0	0
5	0	0	0	0	1	0
6	1	0	0	0	0	1
7	0	1	0	0	0	1
8	0	0	1	0	0	1
9	0	0	0	1	0	1
10	0	0	0	0	1	1
11	1	0	0	0	0	0

It is very simple to write the count sequence. Firstly, we write the first 10 states of the ring counter output (designated by *A*, *B*, *C*, *D* and *E*). The logic status of *F* can be written by examining the logic status of *E*. *F* toggles whenever *E* undergoes '1' to '0' transition.

Example 11.13

Refer to the logic circuit arrangement of Fig. 11.51 built around an eight-bit serial-in/parallel-out shift register, type number 74164. A and B are the data inputs. The serial data feeding the register are obtained by an ANDing operation of A and B inputs inside the IC. $\overline{MR}$ *is an active LOW master reset. Write the logic status of register outputs for the first eight clock pulses.* Q_0 *represents the first flip-flop in this serial shift register.*

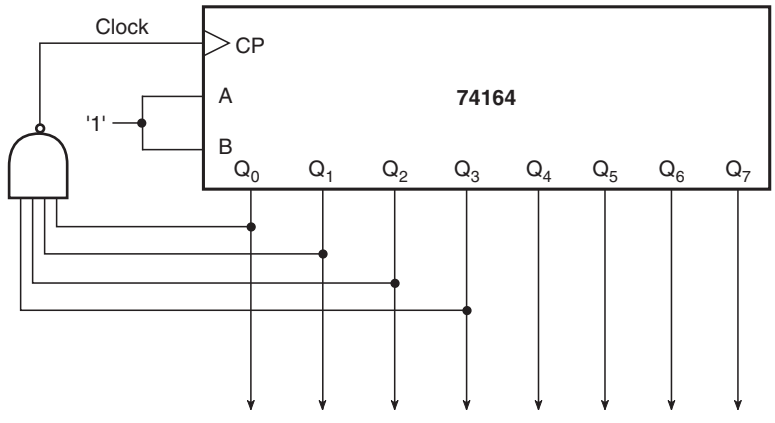


Figure 11.51 Example 11.13.

Solution

Initially, all outputs are in the logic '0' state. Since $A = B = 1$, the serial input to the shift register is a logic '1'. The $\overline{\text{MR}}$ input is initially inactive. For the first three clock pulses, the output status is 10000000, 11000000 and 11100000. With the fourth clock pulse, the output tends to go to 11110000, but it cannot be stable state as the NAND output goes from '1' to '0'. This resets the register to 00000000. Thus, the register transits from 11100000 to 00000000. With the fifth, sixth and seventh clock pulses, the circuit goes through 10000000, 11000000 and 11100000. The eight clock pulse again resets it to 00000000.

11.14 IEEE/ANSI Symbology for Registers and Counters

We introduced IEEE/ANSI symbology for digital integrated circuits as contained in IEEE/ANSI Standard 91-1984 in Section 4.22 of Chapter 4 on logic gates and related devices. A brief description of salient features of this symbology and its particular significance to sequential logic devices such as flip-flops, counters, registers, etc., was given, highlighting the use of dependency notation to provide almost complete functional information of the device. In this section, we will illustrate IEEE/ANSI symbology for counters and registers with the help of IEEE/ANSI symbols of some popular devices.

11.14.1 Counters

As an illustration, we will consider IEEE/ANSI symbols of a decade counter, type number 7490, and a presettable four-bit binary UP/DOWN counter, type number 74193. The IEEE/ANSI notation for IC 7490 and IC 74193 is shown in Figs 11.52(a) and (b) respectively.

The upper portion of the notation represents the common control block that affects all flip-flops constituting the counter. The lower portion represents individual flip-flops. Before we interpret different labels and inputs/outputs for the two counter ICs, we should know the following:

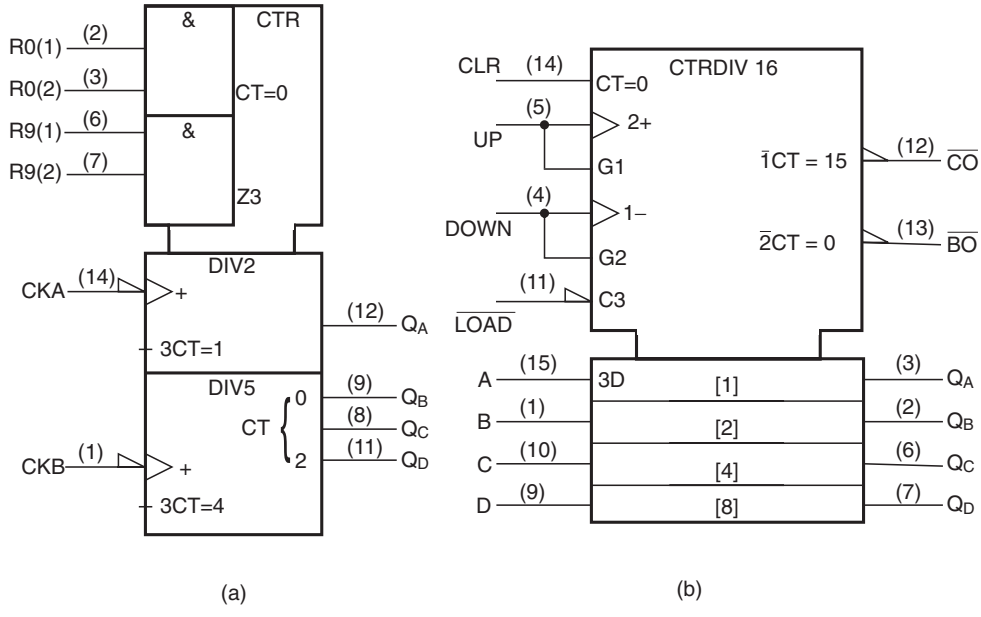


Figure 11.52 IEEE/ANSI notation for (a) IC 7490 and (b) IC 74193.

1. Letter 'C' represents control dependency. Use of the letter 'C' in the label of a certain input means that that particular input controls the entry of data into a storage element such as a flip-flop. The storage element or elements that are controlled by this input are indicated by a digit used as a suffix to the letter 'C'. The same digit appears as a prefix in the labels of all those storage elements that are controlled by this input.
2. Letter 'G' represents an AND dependency. The use of the letter 'G' followed by a digit in the label of an input means that this input is internally ANDed with another input or output and that the input or output will have the same digit as a prefix in its label.
3. Plus (+) and minus (−) signs in the labels indicate the count direction, with the former implying an UP count sequence and the latter implying a DOWN count sequence. These signs are used with clock inputs.

We will now interpret different inputs and outputs for the two counters. We will begin with IC 7490. Reset inputs R_0 (1) and R_0 (2) have an AND dependency, and when both of them are driven to the logic HIGH state the counter is reset to all 0s. Reset inputs R_9 (1) and R_9 (2) also have an AND dependency when both of them are driven to the logic HIGH state, the divide-by-2 portion of the counter is reset to count '1' (which is also the logic '1' state for the flip-flop true output) and the divide-by-5 portion of the counter is reset to count '4' (which is the 100 state for the counter outputs). If the two portions were used in cascade, the counter output would become 1001, which would mean that the counter is reset to count '9'. Clock A (CKA) and clock B (CKB) inputs allow the two portions of the counter to count in the upward sequence as indicated by the (+) sign.

We will now look at the IEEE/ANSI symbol of the other counter, that is, the counter IC type number 74193. Label CTR DIV16 means that IC 74193 is a divide-by-16 counter. Label CT = 0 with master

reset (*MR*) input implies that the counter is reset to all 0s when the *MR* input is in the logic HIGH state. Label C3 with parallel load (*PL*) input means that the data on parallel load inputs P_0, P_1, P_2 and P_3 are loaded onto the corresponding flip-flops when the *PL* input is in the logic LOW state. We can see the prefix 3 in the labels of the flip-flops. The *CPU* input has an AND dependency with the *TCU* output and *CPD* input. In the case of the former, the *TCU* output goes to the logic LOW state when the *CPU* is LOW and the count reaches '15'. In the case of the latter, the *CPU* input should be in the logic HIGH state in order to allow the *CPD* to perform the count DOWN function. Similarly, the *CPD* input has an AND dependency with the *TCD* output and *CPU* input. In the case of the former, the *TCD* output goes to the logic LOW state when the *CPD* is LOW and the count reaches '0'. In the case of the latter, the *CPD* input should be in the logic HIGH state in order to allow the *CPU* to perform the count UP function.

11.14.2 Registers

As an illustration, we will consider IEEE/ANSI symbols of a serial-in serial-out shift register, type number 7491, and a serial-in parallel-out shift register, type number 74164. Figures 11.53(a) and (b) show the IEEE/ANSI notations for IC 7491 and IC 74164 respectively.

We will begin with shift register type number 7491. Label SRG8 stands for eight-bit shift register. Label C1/→ with the clock input means that the relevant clock transition performs two functions. Firstly, it loads data onto the data input as indicated by prefix '1' with the *D* input. Secondly, it performs a right shift operation. The *A* and *B* inputs have an AND dependency. When data are entered through either of the two inputs, the other input must be held in the logic HIGH state to allow the data bit to be loaded onto the data input terminal.

We will now consider shift register type number 74164. Label 'R' stands for reset operation. Whenever the *MR* input is driven to the logic LOW state, the shift register is reset to all 0s. The rest of the notations have already been explained in the case of register type number 7491.

11.15 Application-Relevant Information

Table 11.16 lists the commonly used IC counters and registers belonging to the TTL, CMOS and ECL logic families. Application-relevant information on more popular type numbers is given in the companion website. The information includes the pin configuration diagram, functional table and timing waveforms in some cases.

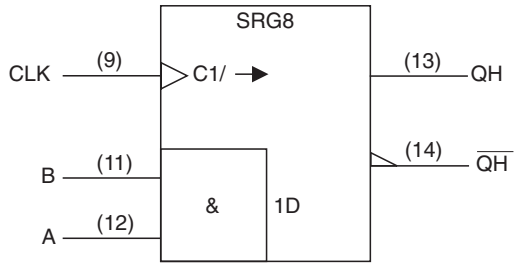
Review Questions

1. Differentiate between:

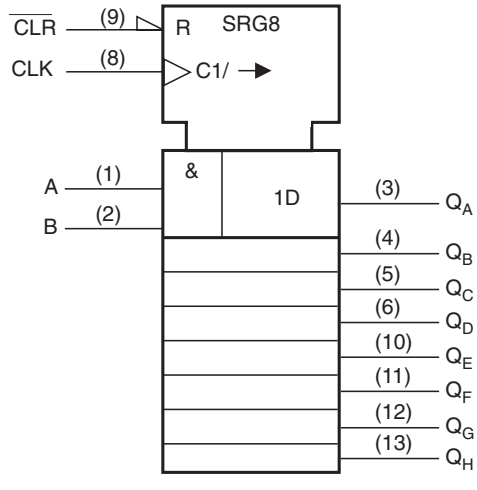
   (a) asynchronous and synchronous counters;
   (b) UP, DOWN and UP/DOWN counters;
   (c) presettable and clearable counters;
   (d) BCD and decade counters.

2. Indicate the difference between the counting sequences of:

   (a) a four-bit binary UP counter and a four-bit binary DOWN counter;
   (b) a four-bit ring counter and a four-bit Johnson counter.



(a)



(b)

Figure 11.53 IEEE/ANSI notation for (a) IC 7491 and (b) IC 74164.

3. Briefly describe:

   (a) how the architecture of an asynchronous UP counter differs from that of a DOWN counter;
   (b) how the architecture of a ring counter differs from that of a shift counter.

4. Briefly explain why the maximum usable clock frequency of a ripple counter decreases as more flip-flops are added to the counter to increase its MOD-number.
5. Why is the maximum usable clock frequency in the case of a synchronous counter independent of the size of counter?
6. How can presettable counters be used to construct counters with variable modulus?