11.5 Synchronous (or Parallel) Counters

Ripple counters discussed thus far in this chapter are asynchronous in nature as the different flip-flops comprising the counter are not clocked simultaneously and in synchronism with the clock pulses. The total propagation delay in such a counter, as explained earlier, is equal to the sum of propagation delays due to different flip-flops. The propagation delay becomes prohibitively large in a ripple counter with a large count. On the other hand, in a synchronous counter, all flip-flops in the counter are clocked simultaneously in synchronism with the clock, and as a consequence all flip-flops change state at the same time. The propagation delay in this case is independent of the number of flip-flops used.

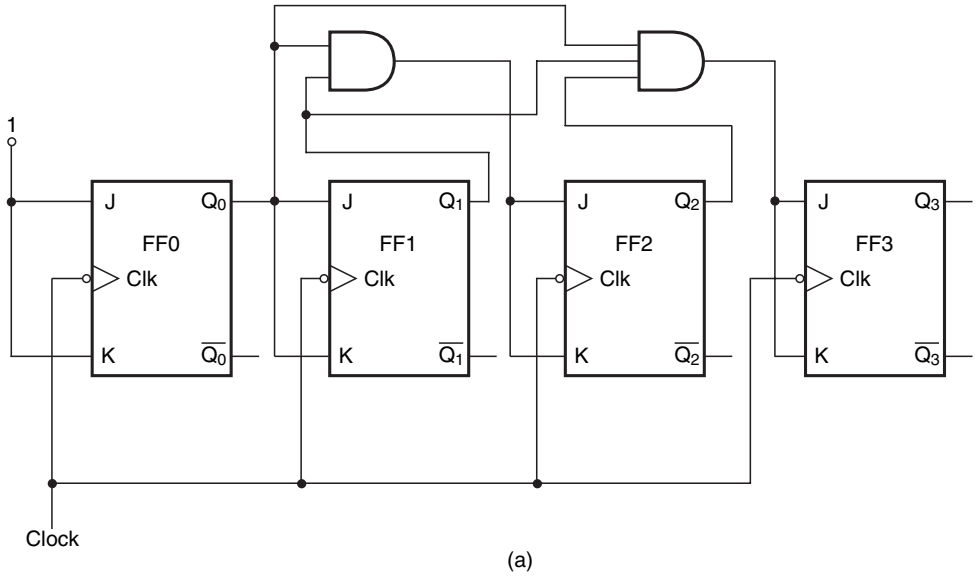
Since the different flip-flops in a synchronous counter are clocked at the same time, there needs to be additional logic circuitry to ensure that the various flip-flops toggle at the right time. For instance, if we look at the count sequence of a four-bit binary counter shown in Table 11.4, we find that flip-flop FF0 toggles with every clock pulse, flip-flop FF1 toggles only when the output of FF0 is in the '1' state, flip-flop FF2 toggles only with those clock pulses when the outputs of FF0 and FF1 are both in the logic '1' state and flip-flop FF3 toggles only with those clock pulses when Q_0, Q_1 and Q_2 are all in the logic '1' state. Such logic can be easily implemented with AND gates. Figure 11.9(a) shows the schematic arrangement of a four-bit synchronous counter. The timing waveforms are shown in Fig. 11.9(b). The diagram is self-explanatory. As an example, ICs 74162 and 74163 are four-bit synchronous counters, with the former being a decade counter and the latter a binary counter.

A synchronous counter that counts in the reverse or downward sequence can be constructed in a similar manner by using complementary outputs of the flip-flops to drive the J and K inputs of the following flip-flops. Refer to the reverse or downward count sequence as given in Table 11.5. As is evident from the table, FF0 toggles with every clock pulse, FF1 toggles only when Q_0 is logic '0', FF2 toggles only when both Q_0 and Q_1 are in the logic '0' state and FF3 toggles only when Q_0, Q_1 and Q_2 are in the logic '0' state.

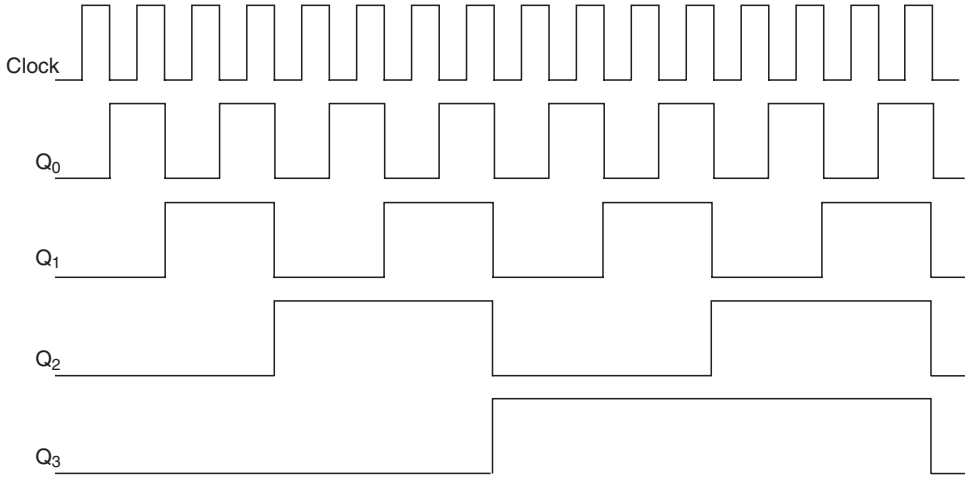
Referring to the four-bit synchronous UP counter of Fig. 11.9(a), if the J and K inputs of flip-flop FF1 are fed from the $\overline{Q_0}$ output instead of the Q_0 output, the inputs to the two-input AND gate are $\overline{Q_0}$ and $\overline{Q_1}$ instead of Q_0 and Q_1, and the inputs to the three-input AND gate are $\overline{Q_0}$, $\overline{Q_1}$ and $\overline{Q_2}$ instead of Q_0, Q_1 and Q_2, we get a counter that counts in reverse order. In that case it becomes a four-bit synchronous DOWN counter.

Table 11.4 Count sequence of a four-bit binary counter.

Count	Q_3	Q_2	Q_1	Q_0	Count	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0	8	1	0	0	0
1	0	0	0	1	9	1	0	0	1
2	0	0	1	0	10	1	0	1	0
3	0	0	1	1	11	1	0	1	1
4	0	1	0	0	12	1	1	0	0
5	0	1	0	1	13	1	1	0	1
6	0	1	1	0	14	1	1	1	0
7	0	1	1	1	15	1	1	1	1



(a)



(b)

Figure 11.9 Four-bit synchronous counter.

Table 11.5 Reverse or downward count sequence synchronous counter.

Count	Q_3	Q_2	Q_1	Q_0	Count	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0	8	1	0	0	0
1	1	1	1	1	9	0	1	1	1
2	1	1	1	0	10	0	1	1	0
3	1	1	0	1	11	0	1	0	1
4	1	1	0	0	12	0	1	0	0
5	1	0	1	1	13	0	0	1	1
6	1	0	1	0	14	0	0	1	0
7	1	0	0	1	15	0	0	0	1

11.6 UP/DOWN Counters

Counters are also available in integrated circuit form as UP/DOWN counters, which can be made to operate as either UP or DOWN counters. As outlined in Section 11.5, an UP counter is one that counts upwards or in the forward direction by one LSB every time it is clocked. A four-bit binary UP counter will count as 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111, 0000, 0001, . . . and so on. A DOWN counter counts in the reverse direction or downwards by one LSB every time it is clocked. The four-bit binary DOWN counter will count as 0000, 1111, 1110, 1101, 1100, 1011, 1010, 1001, 1000, 0111, 0110, 0101, 0100, 0011, 0010, 0001, 0000, 1111, . . . and so on.

Some counter ICs have separate clock inputs for UP and DOWN counts, while others have a single clock input and an UP/DOWN control pin. The logic status of this control pin decides the counting mode. As an example, ICs 74190 and 74191 are four-bit UP/DOWN counters in the TTL family with a single clock input and an UP/DOWN control pin. While IC 74190 is a BCD decade counter, IC 74191 is a binary counter. Also, ICs 74192 and 74193 are four-bit UP/DOWN counters in the TTL family, with separate clock input terminals for UP and DOWN counts. While IC 74192 is a BCD decade counter, IC 74193 is a binary counter.

Figure 11.10 shows a three-bit binary UP/DOWN counter. This is only one possible logic arrangement. As we can see, the counter counts upwards when UP control is logic '1' and DOWN

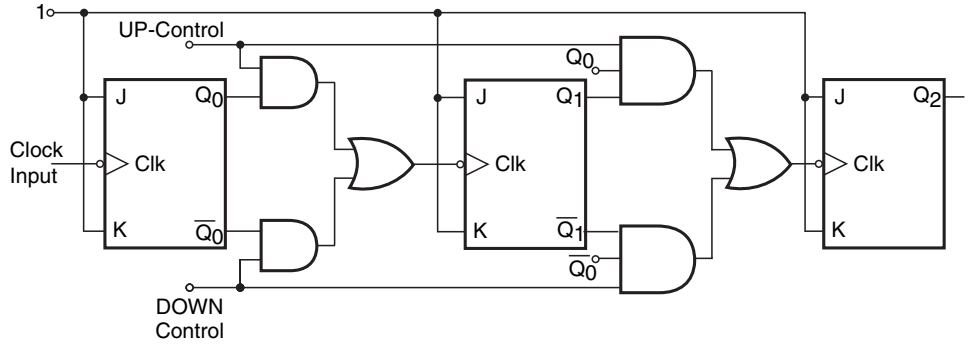


Figure 11.10 Four-bit UP/DOWN counter.

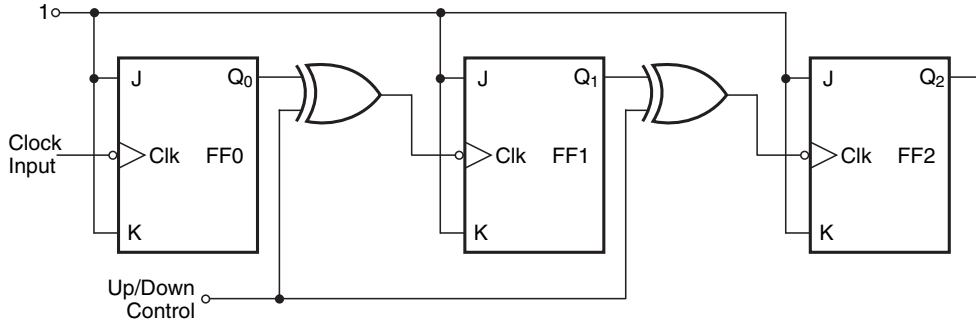


Figure 11.11 Three-bit UP/DOWN counter with a common clock input.

control is logic '0'. In this case the clock input of each flip-flop other than the LSB flip-flop is fed from the normal output of the immediately preceding flip-flop. The counter counts downwards when the UP control input is logic '0' and DOWN control is logic '1'. In this case, the clock input of each flip-flop other than the LSB flip-flop is fed from the complemented output of the immediately preceding flip-flop. Figure 11.11 shows another possible configuration for a three-bit binary ripple UP/DOWN counter. It has a common control input. When this input is in logic '1' state the counter counts downwards, and when it is in logic '0' state it counts upwards.

11.7 Decade and BCD Counters

A *decade counter* is one that goes through 10 unique output combinations and then resets as the clock proceeds further. Since it is an MOD-10 counter, it can be constructed with a minimum of four flip-flops. A four-bit counter would have 16 states. By skipping any of the six states by using some kind of feedback or some kind of additional logic, we can convert a normal four-bit binary counter into a decade counter. A decade counter does not necessarily count from 0000 to 1001. It could even count as 0000, 0001, 0010, 0101, 0110, 1001, 1010, 1100, 1101, 1111, 0000, . . . In this count sequence, we have skipped 0011, 0100, 0111, 1000, 1011 and 1110.

A *BCD counter* is a special case of a decade counter in which the counter counts from 0000 to 1001 and then resets. The output weights of flip-flops in these counters are in accordance with 8421-code. For instance, at the end of the seventh clock pulse, the counter output will be 0111, which is the binary equivalent of decimal 7. In other words, different counter states in this counter are binary equivalents of the decimal numbers 0 to 9. These are different from other decade counters that provide the same count by using some kind of forced feedback to skip six of the natural binary counts.

11.8 Presettable Counters

Presettable counters are those that can be preset to any starting count either asynchronously (independently of the clock signal) or synchronously (with the active transition of the clock signal). The presetting operation is achieved with the help of PRESET and CLEAR (or MASTER RESET) inputs available on the flip-flops. The presetting operation is also known as the 'preloading' or simply the 'loading' operation.

Presettable counters can be UP counters, DOWN counters or UP/DOWN counters. Additional inputs/outputs available on a presettable UP/DOWN counter usually include PRESET inputs, from where any desired count can be loaded, parallel load (*PL*) inputs, which when active allow the PRESET inputs to be loaded onto the counter outputs, and terminal count (*TC*) outputs, which become active when the counter reaches the terminal count.

Figure 11.12 shows the logic diagram of a four-bit presettable synchronous UP counter. The data available on P_3, P_2, P_1 and P_0 inputs are loaded onto the counter when the parallel load ($\overline{PL}$) input goes LOW.

When the $\overline{PL}$ input goes LOW, one of the inputs of all NAND gates, including the four NAND gates connected to the PRESET inputs and the four NAND gates connected to the CLEAR inputs, goes to the logic '1' state. What reaches the PRESET inputs of FF3, FF2, FF1 and FF0 is $\overline{P_3}$, $\overline{P_2}$, $\overline{P_1}$ and $\overline{P_0}$ respectively, and what reaches their CLEAR inputs is P_3, P_2, P_1 and P_0 respectively. Since PRESET and CLEAR are active LOW inputs, the counter flip-flops FF3, FF2, FF1 and FF0 will respectively be loaded with P_3, P_2, P_1 and P_0. For example, if $P_3 = 1$, the PRESET and CLEAR inputs of FF3 will be in the '0' and '1' logic states respectively. This implies that the Q_3 output will go to the logic '1' state. Thus, FF3 has been loaded with P_3. Similarly, if $P_3 = 0$, the PRESET and CLEAR inputs of flip-flop FF3 will be in the '1' and '0' states respectively. The flip-flop output (Q_3 output) will be cleared to the '0' state. Again, the flip-flop is loaded with P_3 logic status when the $\overline{PL}$ input becomes active.

Counter ICs 74190, 74191, 74192 and 74193 are asynchronously presettable synchronous UP/DOWN counters. Many synchronous counters use synchronous presetting whereby the counter is preset or loaded with the data on the active transition of the same clock signal that is used for counting. Presettable counters also have terminal count ($\overline{TC}$) outputs, which allow them to be cascaded together to get counters with higher MOD numbers. In the cascade arrangement, the terminal count output of the lower-order counter feeds the clock input of the next higher-order counter. Cascading of counters is discussed in Section 11.10.

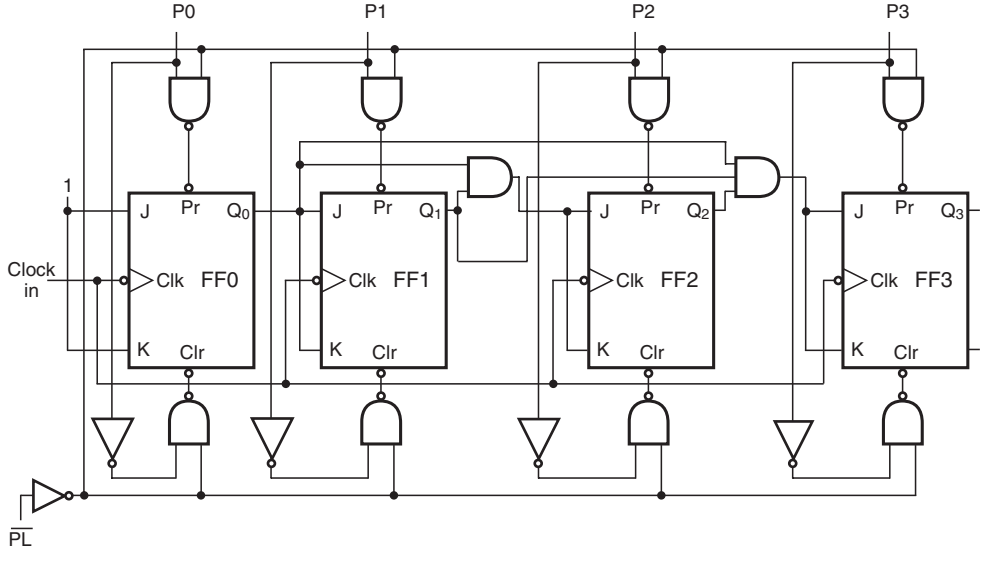


Figure 11.12 Four-bit presettable, clearable counter.

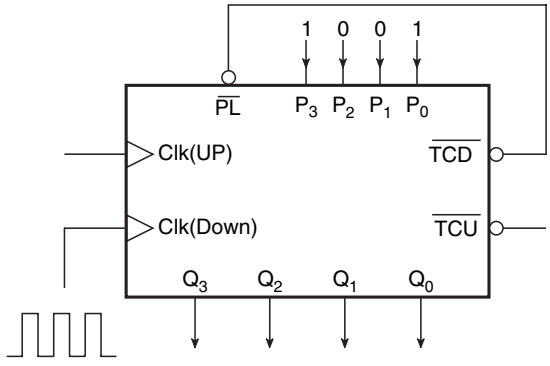


Figure 11.13 Presettable four-bit counter.

11.8.1 Variable Modulus with Presettable Counters

Presettable counters can be wired as counters with a modulus of less than 2^N without the need for any additional logic circuitry. When a presettable counter is preset with a binary number whose decimal equivalent is some number 'X', and if this counter is wired as a DOWN counter, with its terminal count (DOWN mode) output, also called borrow-out (B_o), fed back to the parallel load (PL) input, it works like an MOD-X counter.

We will illustrate this with the help of an example. Refer to Fig. 11.13. It shows a presettable four-bit synchronous UP/DOWN binary counter having separate clock inputs for UP and DOWN counting (both positive edge triggered), an active LOW parallel load input ($\overline{PL}$) and active LOW terminal count UP ($\overline{TCU}$) and terminal count DOWN ($\overline{TCD}$) outputs. This description is representative of IC counter type 74193. Let us assume that the counter is counting down and is presently in the 1001 state at time instant t_0. The $\overline{TCD}$ output is in the logic '1' state, and so is the $\overline{PL}$ input. That is, both are inactive. The counter counts down by one LSB at every positive-going edge of the clock input. Immediately after the ninth positive-going trigger (at time instant t_9), the counter is in the 0000 state, which is the terminal count. Coinciding with the negative-going edge of the same clock pulse, the $\overline{TCD}$ output goes to the logic '0' state, and so does the $\overline{PL}$ input. This loads the counter with 1001 at time instant t_{10}, as shown in the timing waveforms of Fig. 11.14. With the positive-going edges of the tenth clock pulse and thereafter, the counter repeats its DOWN count sequence. Examination of the Q_3 output waveform tells that its frequency is one-ninth of the input clock frequency. Thus, it is an MOD-9 counter. The modulus of the counter can be varied by varying the data loaded onto the parallel PRESET/LOAD inputs.

11.9 Decoding a Counter

The output state of a counter at any time instant, as it is being clocked, is in the form of a sequence of binary digits. For a large number of applications, it is important to detect or decode different states of the counter whose number equals the modulus of the counter. One typical application could be a need to initiate or trigger some action after the counter reaches a specific state. The decoding network therefore is going to be a logic circuit that takes its inputs from the outputs of the different flip-flops constituting the counter and then makes use of those data to generate outputs equal to the modulus or MOD-number of the counter.

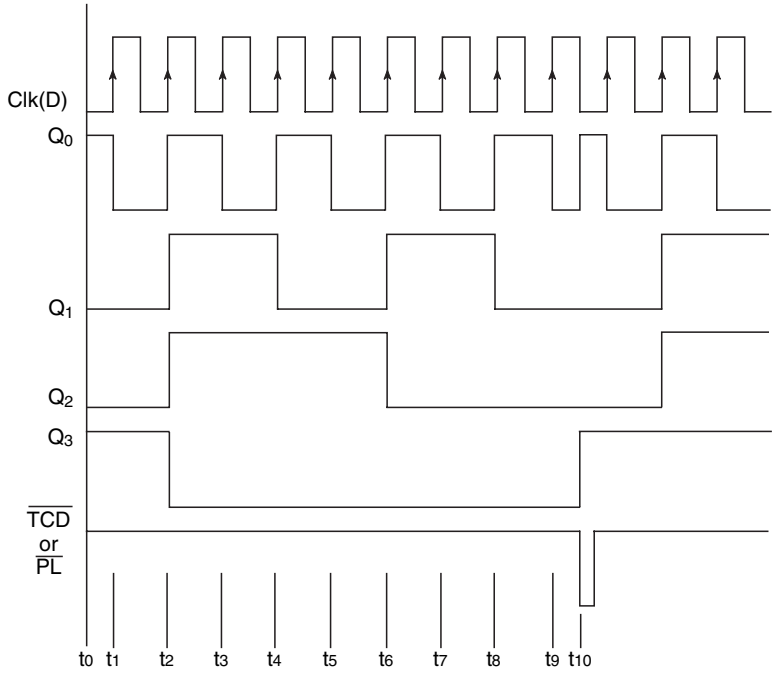
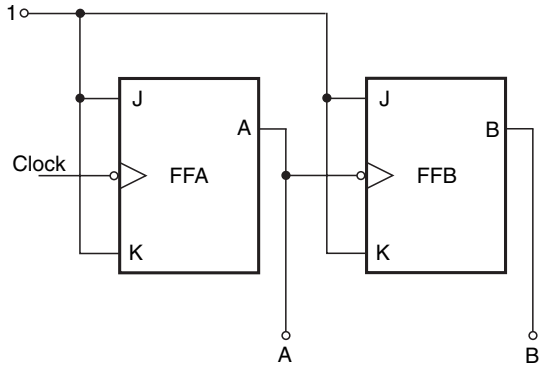


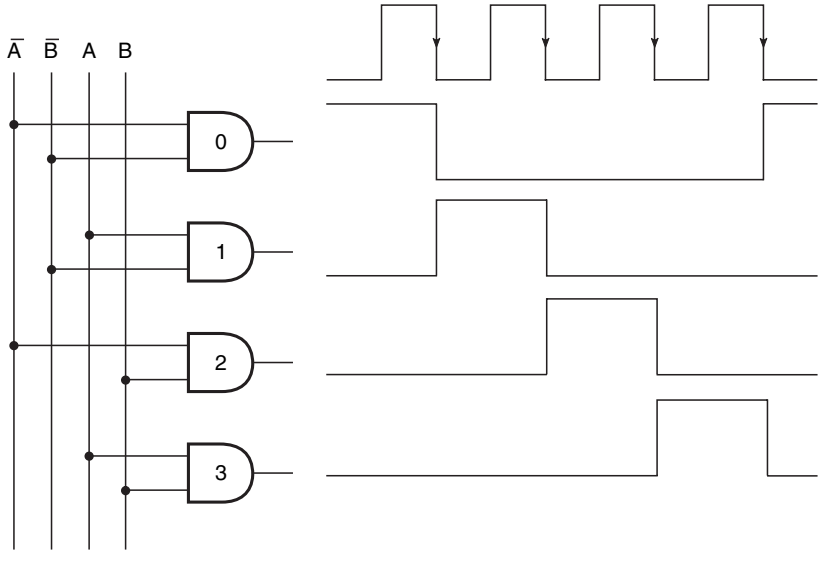
Figure 11.14 Timing waveforms for the counter of Fig. 11.13.

Depending upon the logic status of the decoded output, there are two basic types of decoding, namely *active HIGH* decoding and *active LOW* decoding. In the case of the former the decoder outputs are normally LOW, and for a given counter state the corresponding decoder output goes to the logic HIGH state. In the case of active LOW decoding, the decoder outputs are normally HIGH and the decoded output representing the counter state goes to the logic LOW state.

We will further illustrate the concept of decoding a counter with the help of an example. Consider the two-stage MOD-4 ripple counter of Fig. 11.15(a). This counter has four possible logic states, which need to be decoded. These include 00, 01, 10 and 11. Let us now consider the arrangement of four two-input AND gates as shown in Fig. 11.15(b) and what their outputs look like as the counter clock goes through the first four pulses. Before we proceed further, we have two important observations to make. Firstly, the number of AND gates used in the decoder network equals the number of logic states to be decoded, which further equals the modulus of the counter. Secondly, the number of inputs to each AND gate equals the number of flip-flops used in the counter. We can see from the waveforms of Fig. 11.15(b) that, when the counter is in the 00 state, the AND gate designated '0' is in the logic HIGH state and the outputs of the other gates designated '1', '2' and '3' are in the logic LOW state. Similarly, for 01, 10 and 11 states of the counter, the outputs of gates 1, 2 and 3 are respectively in the logic HIGH state. This is incidentally active HIGH decoding. We can visualize that, if the AND gates were replaced with NAND gates, with the inputs to the gates remaining the same, we would get an active LOW decoder. For a counter that uses N flip-flops and has a modulus of 'X', the decoder will have 'X' number of N-input AND or NAND gates, depending upon whether we want an active HIGH or active LOW decoder.



(a)



(b)

Figure 11.15 MOD-4 ripple counter with decoding logic.

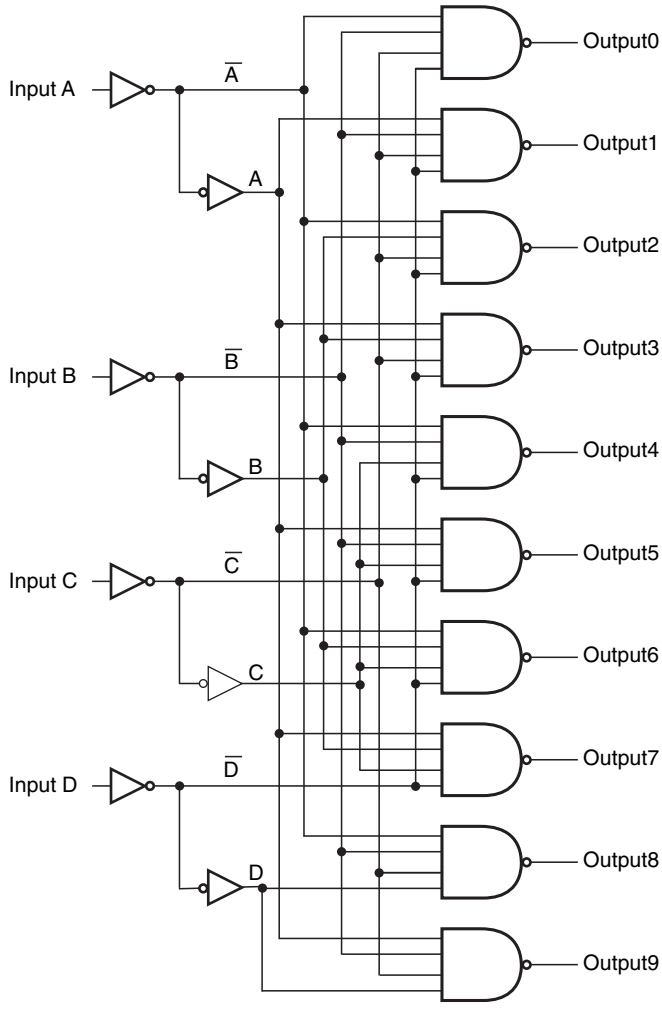


Figure 11.16 Logic diagram of four-line BCD-to-decimal decoder (IC 7442).

Figure 11.16 shows the logic diagram of a four-line BCD to decimal decoder with active low outputs. Full decoding of valid input logic states ensures that all outputs remain off or inactive for all invalid input conditions. Table 11.6 gives the functional table of the decoder of Fig. 11.16. The logic diagram shown in Fig. 11.16 is the actual logic diagram of IC 7442, which is a four-line BCD to decimal decoder in the TTL family.

The decoding gates used to decode the states of a ripple counter produce glitches (or spikes) in the decoded waveforms. These glitches basically result from the cumulative propagation delay as we move from one flip-flop to the next in a ripple counter. It can be best illustrated with the help of the MOD-4 counter shown in Fig. 11.17. The timing waveforms are shown in Fig. 11.18 and are self-explanatory.

Table 11.6 Functional table of the decoder of Fig. 11.16.

Decimal number	BCD input				Decimal output									
	D	*C*	*B*	*A*	0	1	2	3	4	5	6	7	8	9
0	L	L	L	L	L	H	H	H	H	H	H	H	H	H
1	L	L	L	H	H	L	H	H	H	H	H	H	H	H
2	L	L	H	L	H	H	L	H	H	H	H	H	H	H
3	L	L	H	H	H	H	H	L	H	H	H	H	H	H
4	L	H	L	L	H	H	H	H	L	H	H	H	H	H
5	L	H	L	H	H	H	H	H	H	L	H	H	H	H
6	L	H	H	L	H	H	H	H	H	H	L	H	H	H
7	L	H	H	H	H	H	H	H	H	H	H	L	H	H
8	H	L	L	L	H	H	H	H	H	H	H	H	L	H
9	H	L	L	H	H	H	H	H	H	H	H	H	H	L
Invalid	H	L	H	L	H	H	H	H	H	H	H	H	H	H
Invalid	H	L	H	H	H	H	H	H	H	H	H	H	H	H
Invalid	H	H	L	L	H	H	H	H	H	H	H	H	H	H
Invalid	H	H	L	H	H	H	H	H	H	H	H	H	H	H
Invalid	H	H	H	L	H	H	H	H	H	H	H	H	H	H
Invalid	H	H	H	H	H	H	H	H	H	H	H	H	H	H

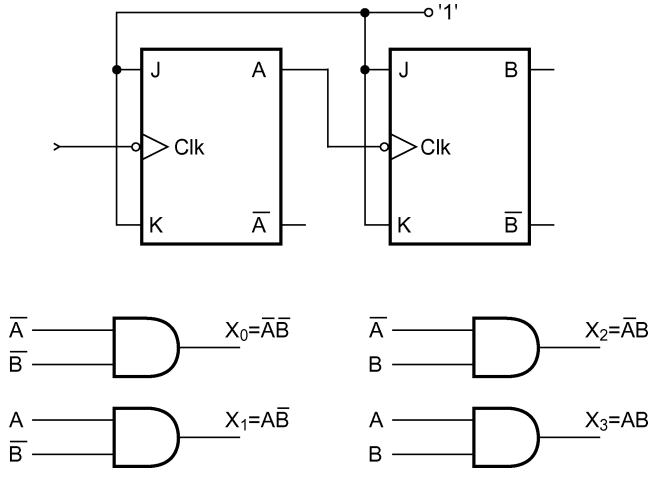


Figure 11.17 MOD-4 counter with decoding gates.

We can see the appearance of glitches at the output of decoding gates that decode X_0 and X_2 states.This problem for all practical purposes is absent in synchronous counters. Theoretically, it can even exist in a synchronous counter if the flip-flops used have different propagation delays.

One way to overcome this problem is to use a strobe signal which keeps the decoding gates disabled until all flip-flops have reached a stable state in response to the relevant clock transition. To implement

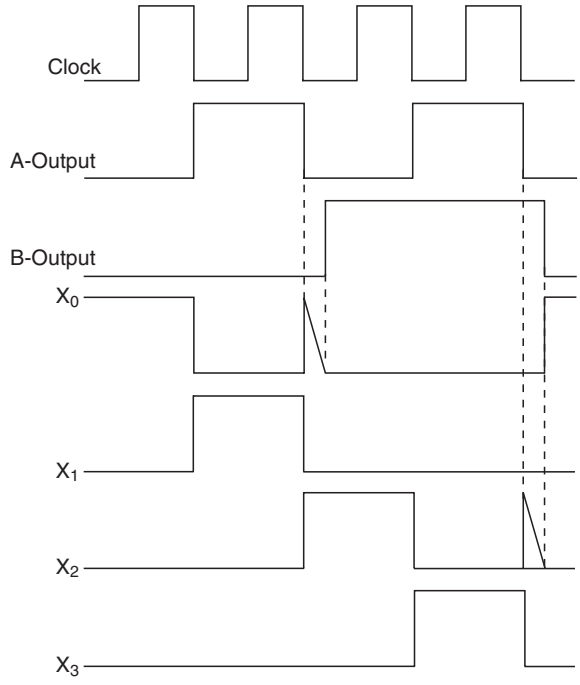


Figure 11.18 Glitch problem in decoders.

this, each of the decoding gates will have an additional input. This additional input of all decoding gates is tied together and the strobe signal applied to the common point.

One such decoder with additional strobe inputs to take care of glitch-related problems is IC 74154, which is a four-line to 16-line decoder in the TTL family. Figure 11.19 shows the internal logic diagram of IC 74154. We can see all NAND gates having an additional input line, which is controlled by strobe inputs $\overline{G}_1$ and $\overline{G}_2$.

11.10 Cascading Counters

A cascade arrangement allows us to build counters with a higher modulus than is possible with a single stage. The terminal count outputs allow more than one counter to be connected in a cascade arrangement. In the following paragraphs, we will examine some such cascade arrangements in the case of binary and BCD counters.

11.10.1 Cascading Binary Counters

In order to construct a multistage UP counter, all counter stages are connected in the count UP mode. The clock is applied to the clock input of a lowest-order counter, the terminal count UP (*TCU*), also called the carry-out (C_o), of this counter is applied to the clock input of the next higher counter stage

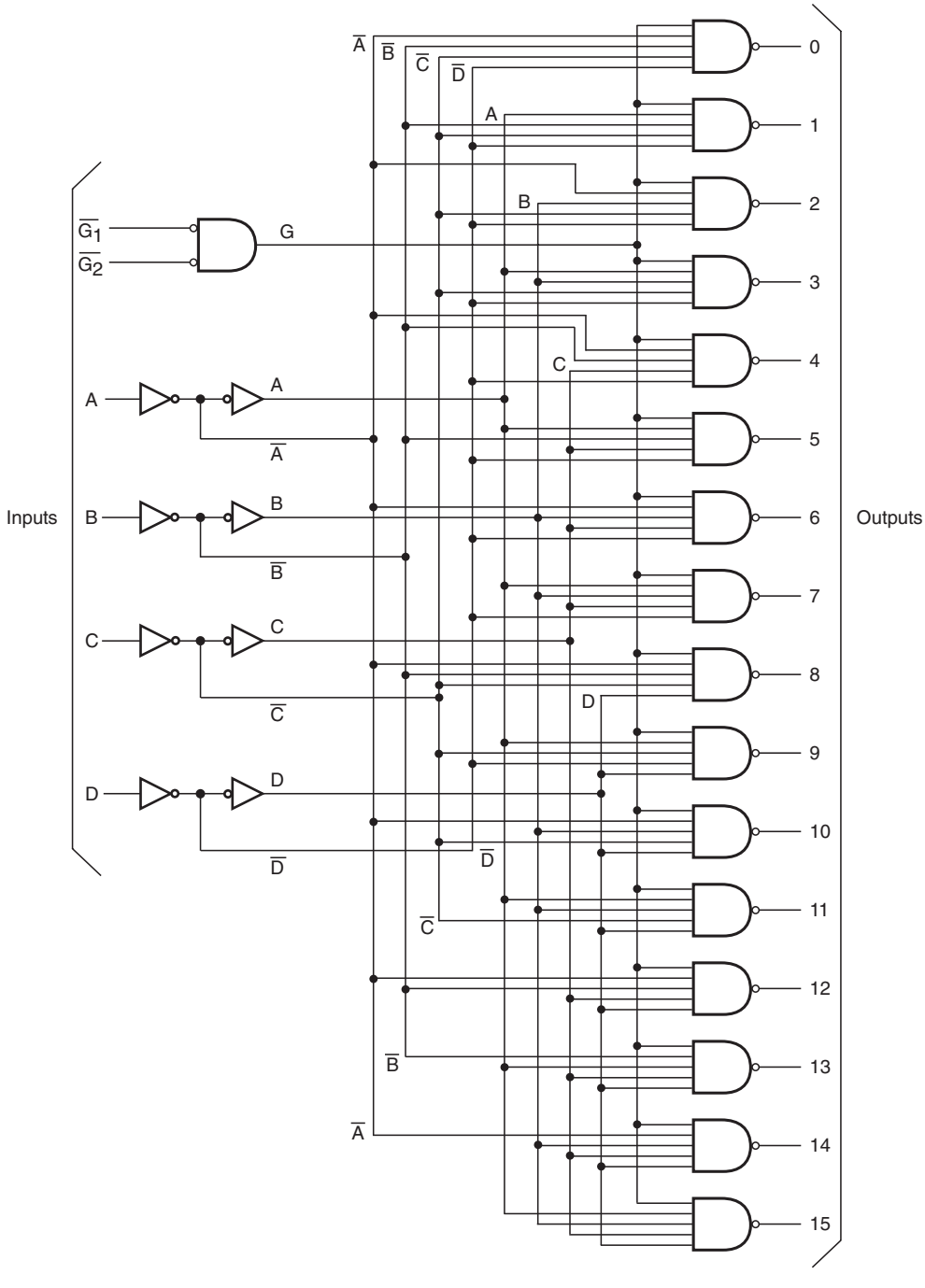


Figure 11.19 Logic diagram of IC 74154.

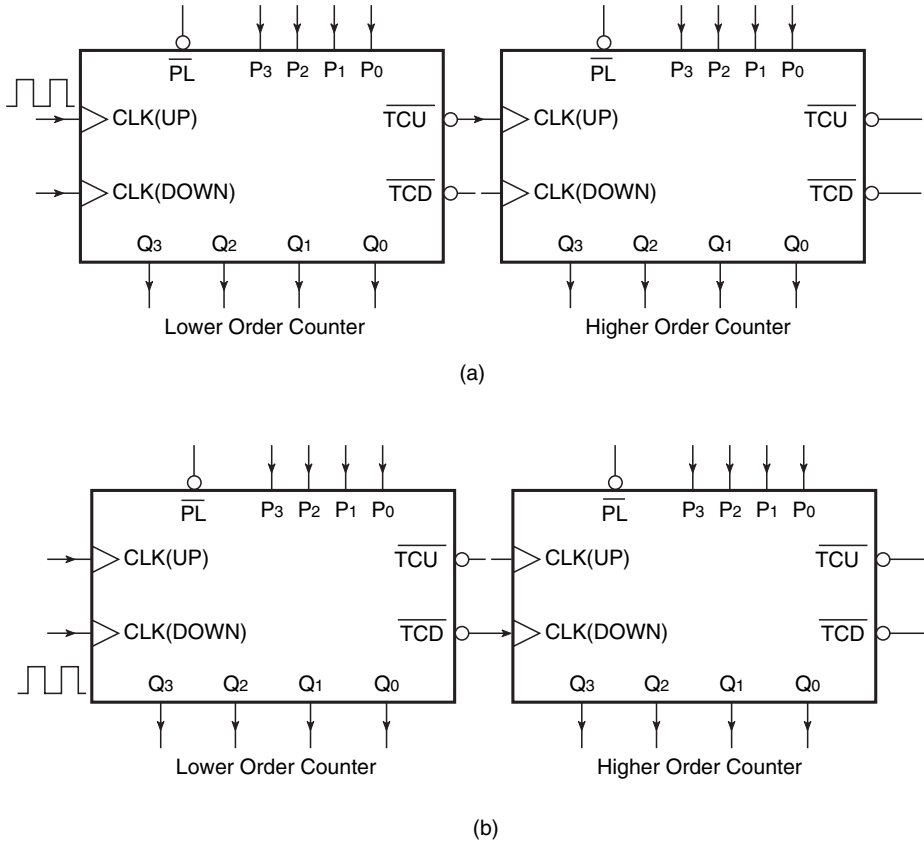


Figure 11.20 Cascading binary counters.

and the process continues. If it is desired to build a multistage DOWN counter, all counters are wired as DOWN counters, the clock is applied to the clock input of the lowest-order counter and the terminal count DOWN (*TCD*), also called the borrow-out (B_o), of the lowest-order counter is applied to the clock input of the next higher counter stage. The process continues in the same fashion, with the TCD output of the second stage feeding the clock input of the third stage and so on. The modulus of the multistage counter arrangement equals the product of the moduli of individual stages. Figures 11.20(a) and (b) respectively show two-stage arrangements of four-bit synchronous UP and DOWN counters respectively.

11.10.2 Cascading BCD Counters

BCD counters are used when the application involves the counting of pulses and the result of counting is to be displayed in decimal. A single-stage BCD counter counts from 0000 (decimal equivalent '0') to 1001 (decimal equivalent '9') and thus is capable of counting up to a maximum of nine pulses. The output in a BCD counter is in binary coded decimal (BCD) form. The BCD output needs

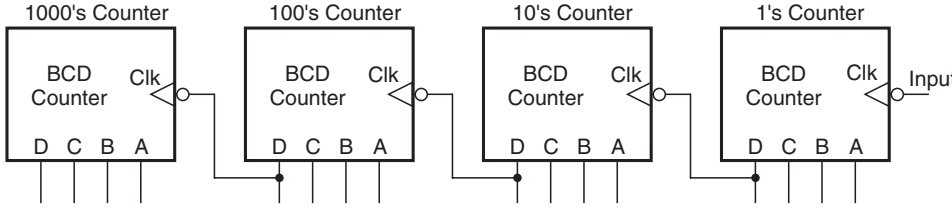


Figure 11.21 Cascading BCD counters.

to be decoded appropriately before it can be displayed. Decoding a counter has been discussed in the previous section. Coming back to the question of counting pulses, more than one BCD counter stage needs to be used in a cascade arrangement in order to be able to count up to a larger number of pulses. The number of BCD counter stages to be used equals the number of decimal digits in the maximum number of pulses we want to count up to. With a maximum count of 9999 or 3843, both would require a four-stage BCD counter arrangement with each stage representing one decimal digit.

Figure 11.21 shows a cascade arrangement of four BCD counter stages. The arrangement works as follows. Initially, all four counters are in the all 0s state. The counter representing the decimal digit of 1's place is clocked by the pulsed signal that needs to be counted. The successive flip-flops are clocked by the MSB of the immediately previous counter stage. The first nine pulses take 1's place counter to 1001. The tenth pulse resets it to 0000, and '1' to '0' transition at the MSB of 1's place counter clocks 10's place counter. 10's place counter gets clocked on every tenth input clock pulse. On the hundredth clock pulse, the MSB of 10's counter makes a '1' to '0' transition which clocks 100's place counter. This counter gets clocked on every successive hundredth input clock pulse. On the thousandth input clock pulse, the MSB of 100's counter makes 'l' to '0' transition for the first time and clocks 1000's place counter. This counter is clocked thereafter on every successive thousandth input clock pulse. With this background, we can always tell the output state of the cascade arrangement. For example, immediately after the 7364th input clock pulse, the state of 1000's, 100's, 10's and 1's BCD counters would respectively be 0111, 0011, 0110 and 0100.

Example 11.6

Figure 11.22 shows a cascade arrangement of two 74190s. Both the UP/DOWN counters are wired as UP counters. What will be the logic status of outputs designated as A, B, C, D, E, F, G and H after the 34th clock pulse?

Solution

The cascade arrangement basically constitutes a two-stage BCD counter that can count from 0 to 99. The counter shown on the left forms 1's place counter, while the one on the right is 10's place counter. The ripple clock ($\overline{RC}$) output internally enabled by the terminal count ($\overline{\text{TC}}$) clocks 10's place counter on the tenth clock pulse and thereafter on every successive tenth clock pulse. At the end of the 34th clock pulse, 1's counter stores the binary equivalent of '4' and 10's counter stores the binary equivalent of '3'. Therefore, the logic status of A, B, C, D, E, F, G and H outputs will be 0, 0, 1, 0, 1,1, 0 and 0 respectively.

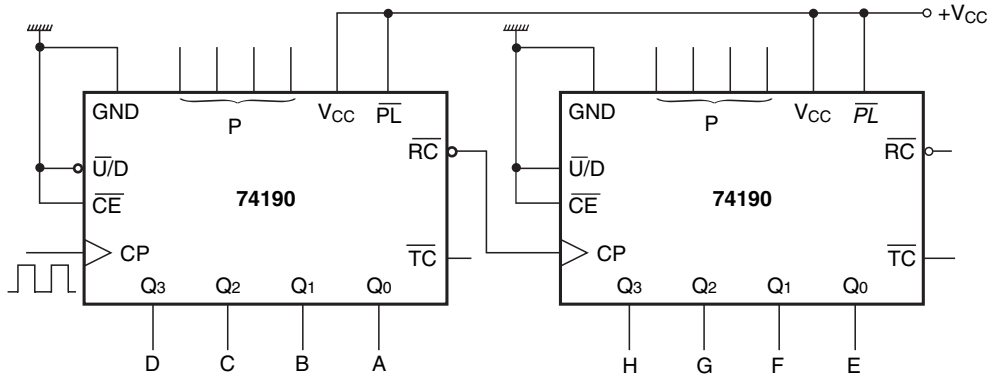


Figure 11.22 Cascade arrangement of two 74190s (example 11.6).

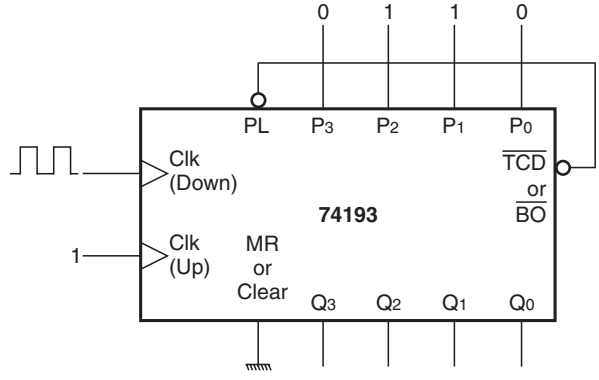


Figure 11.23 Presettable counter (example 11.7).

Example 11.7

Determine the modulus of the presettable counter shown in Fig. 11.23. If the counter were initially in the 0110 state, what would be the state of the counter immediately after the eighth clock pulse be?

Solution

- This presettable counter has been wired as a DOWN counter.
- The preset data input is 0110.
- Therefore, the modulus of the counter is 6 (the decimal equivalent of 0110).
- Now, the counter is initially in the 0110 state.
- Therefore, at the end of the sixth clock pulse, immediately after the leading edge of the sixth clock pulse, the counter will be in the 0000 state.