

When  $T = 0$ , both AND gates are disabled and hence there is no change in the previous output. When  $T = 1$  ( $J = K = 1$ ) output toggles. Toggles means that the output is 0 when the previous state is 1 otherwise output is 1 when the previous state is 0. So the output is a complement of the previous output.

Truth Table of T Flip Flop

Present State	Flip-Flop Input	Next State
$Q_n$	T	$Q_{n+1}$
0	0	0
0	1	1
1	0	1
1	1	0

#### 4.14 APPLICATIONS OF FLIP-FLOPS

Flip-flops find wide applications in the following circuits :

- Counters
- Frequency dividers
- Shift and storage registers
- Serial decoding
- Comparators
- Monostable multivibrators
- Timing signal generation
- Data transfer
- Contact bounce elimination, etc.

#### 4.15 INTRODUCTION TO COUNTERS

A counter is a sequential digital circuit whose output states progress in a predictable repeating pattern, advancing by one state for each clock pulse. Counters are one of the simplest types of sequential networks. A counter is usually constructed from two or more flip flops which change their state in a prescribed sequence when input clock pulses are received. This count sequence may be ascending, descending or non-linear. These procedures will be applied to more general types of sequential networks. If a counter has  $n$ -flip-flops, then it may count pulses upto  $2^n$ .

(A.P)

(T.S)

Let us see some terms and definitions related to counters.

**Count Sequence :** It is a specific series of output states through which a counter progresses.

**Present State :** The current state of flip-flop outputs in a sequential circuit.

**Next State :** The desired future state of flip-flop outputs in a sequential circuit after the next clock pulse is applied.

**Memory Section :** A set of flip-flops in a circuit that holds its present state.

**Modulus :** The number of states of a counter.

The counters uses a chain of flip-flops. These flip-flops go through the sequence of states, when clock pulses are applied. This is the main difference between a register and a counter. A specified sequence of states is different for different types of counters.

There are two types of counters. They are :

- ✓ Synchronous counters and
- ✓ Asynchronous (Ripple) counters.

In synchronous counters, the common clock input is connected to all of the flip-flops and all the flip-flops are clocked simultaneously.

In asynchronous counters, the first flip-flop is clocked by the external clock pulse and the successive flip-flops are clocked from the previous flip-flop outputs (may be Q or  $\bar{Q}$ ). Therefore the asynchronous counters are not clocked simultaneously.

## 4.16 MODULUS OF A COUNTER

The number of states of a counter is called its **modulus**.

(The maximum modulus of a 4 bit counter is  $16(2^4 = 16)$ . The count sequence of an MOD 16 up counter is from 0 0 0 0 to 1 1 1 1 (0 to 15 on decimal). In general, an n-bit counter has a maximum modulus of  $2^n$ , and a count sequence from 0 to  $2^n - 1$ , 'n' flip flop are needed to construct n bit counter.

$$\text{Mod number} = 2^n$$

Where n is the number of flip-flops

The counter with n flip flops has maximum mod number  $2^n$ . For example 4 bit binary counter is mod 16 counter ( $2^4 = 16$ ). This basic counter can also be modified to produce MOD numbers less than  $2^n$  by allowing the counter to skip those states that are normally part of counting sequence.

The required flip flop are counter must satisfy the following relation.

$$2^n \geq N$$

Where n is the number of bits, this is equal to the number of flip flop required (one flip flop stores one bit), N is the total number of states.

## 4.17 ASYNCHRONOUS (OR) RIPPLE COUNTERS

- To design the ripple counter, the number of flip-flops required depends on the number of states. The number of the output states of the counter is called the "Modulus" (MOD) of the counter. The maximum number of states of a counter is  $2^n$ , where n is the number of flip-flops in the counter. If we have two flip-flops, then the maximum possible number of output states of that counter is  $2^2$  i.e., 4. Now we can name that counter as MOD-4 or Modulus-4 counter.

### (A) Mod-4 Ripple Counter :

MOD-4 ripple counter is a 2-bit asynchronous counter, which consists of four states due to its two number of flip-flops. Fig. 4.23 shows the MOD-4 ripple counter, with the two numbers of JK flip-flops. Notice that the clock signal is connected to the clock input of only first stage flip-flop. The clock input of the second stage flip-flop is triggered by the  $Q_A$  output of the first stage. Note that the first flip-flop changes its state only when triggered by the negative edge of each clock pulse, but the second flip-flop changes its state only when triggered by the negative-going transition of the  $Q_A$  output. Because of the inherent propagation delay time through a flip-flop, the transition of the input clock pulse and transition of the output  $Q_A$  can never occur at exactly the same time. Therefore, the two flip-flops are never simultaneously triggered which results in asynchronous counter operation. Thus, the effect of an input clock pulse 'ripples' through the counter, taking sometime, due to propagation delays and reaches the last flip-flop. Hence the asynchronous counters are also called as "ripple counters".

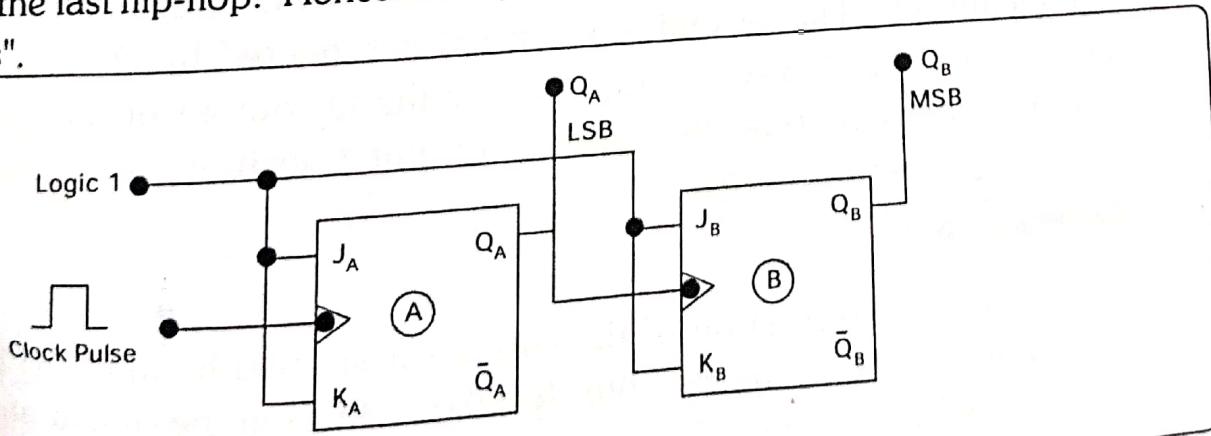


Fig. 4.23 Two-Bit Ripple Counter

Fig. 4.24 shows the timing diagram for two bit asynchronous counter. It illustrates the changes in the state of the flip-flop outputs in response to the clock.

(A.P)

(T.S)

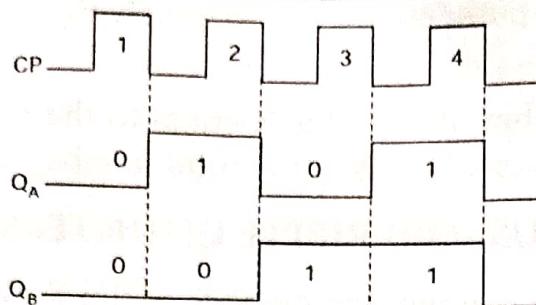


Fig. 4.24 Timing diagram of Two-Bit Ripple Counter

**(B) Module-8 Ripple Counter :**

It is a 3-bit asynchronous counter and contains eight states from 000 to 111 and uses three flip-flops.

Fig. 4.25 shows the MOD-3 ripple counter with 3 JK flip-flops, named as A, B and C, where C output represents the most significant bit of the count and the A flip-flop output indicates the least significant bit of the count.

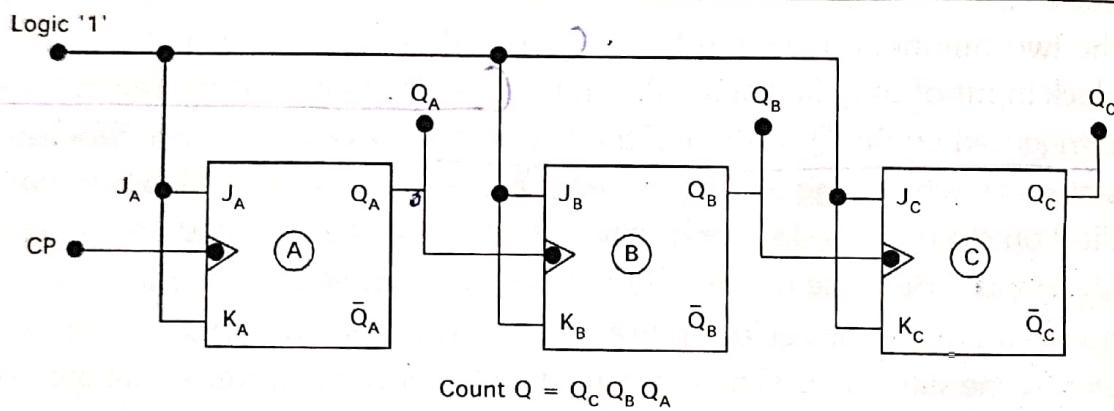


Fig. 4.25 3-Bit Ripple Flip-Flop

Here, first flip-flop (A) is triggered by the CLK which is holding the least significant bit. The second flip-flop (B) is triggered by  $Q_A$  output of first flip-flop. Third flip-flop (C) is triggered by the  $Q_B$  output of the second flip-flop. So, in ripple counters, the flip-flop output transition serves as a source for triggering other flip-flops.

**Operation :**

1. The basic operation of the counter is studied by applying 8-clock pulses one by one to the first flip-flop (A). Observe the output of each flip-flop (ABC) i.e., counter for every clock pulse. Note that the two inputs (J and K) of each flip flop are connected in toggle condition (i.e.,  $J = K = 1$ ) and assume that the counter is initially at RESET. At this state, the counter output is  $Q = Q_C Q_B Q_A = 000$ .

(A.P)

(T.S)

2. When the negative-going edge of the first clock pulse occurs, the output of the flip-flop  $Q_A$  goes to HIGH. (i.e.,  $Q_A$  is changing from 0 to 1). It has no effect on the second (B) and third flip-flop (C), their outputs are same as previous state because to trigger them, a negative transition pulse is required. Thus after the falling edge of the first clock pulse,  $Q_A = 1$ ,  $Q_B = 0$  and  $Q_C = 0$ . At this state, the counter output  $Q = 001$ .
3. For the negative going edge of the second clock pulse, the  $Q_A$  goes to LOW (i.e., change in the output from 1 to 0). This indicates that the  $Q_A$  changes to negative edge triggering which is applied as a clock to B-flip-flop. So, the  $Q_B$  changes from 0 to 1. But, this change in  $Q_B$  does not affect any change in the output of the flip-flop C. After the negative edge of the second clock pulse,  $Q_A = 0$ ,  $Q_B = 1$  and  $Q_C = 0$ . At this stage, the counter output  $Q = 010$ .
4. For the negative going edge of the third pulse, the  $Q_A$  changes from 0 to 1 i.e., LOW to HIGH, which is a positive transition pulse which does not affect the flip-flop B. Hence the output of flip-flop  $B = Q_B$  is same as the previous state. As there is no change in the  $Q_B$ , also, there is no change in the output of flip-flop C i.e.,  $Q_A$ . At the end of the 3rd clock pulse, the outputs of flip-flops are  $Q_A = 1$ ,  $Q_B = 1$  and  $Q_C = 0$ . Hence, the count  $Q = Q_C Q_B Q_A = 011$ .
5. When the fourth clock pulse is applied, the output  $Q_A$  goes to LOW, which is a negative transition pulse (HIGH to LOW) is applied as a clock to the flip-flop B. With the application of this pulse, the output of flip-flop B changes ( $Q_B$ ) from HIGH to LOW, which is another negative transition pulse, applied as a clock to the flip-flop C. With this, the output of flip-flop C,  $Q_C$  changes from LOW to HIGH. After the fourth clock pulse, the outputs are  $Q_A = 0$ ,  $Q_B = 0$ , and  $Q_C = 1$ . Hence, the counter output  $Q = Q_C Q_B Q_A = 100$ .
6. When the fifth clock pulse appears, the output  $Q_A$  goes to HIGH from LOW, which does not affect the output of flip-flops B and C. Hence, at the end of the 5th clock pulse the outputs are  $Q_A = 1$ ,  $Q_B = 0$  and  $Q_C = 1$ , so the counter  $Q = Q_C Q_B Q_A = 101$ .
7. When applying the 6th clock pulse, the output  $Q_A$  goes to LOW, which is a negative transition pulse applied as an input to the flip-flop B. With this, the  $Q_B$  changes from LOW to HIGH. Hence at the end of the 6th clock pulse, the outputs are  $Q_A = 0$ ,  $Q_B = 1$  and  $Q_C = 1$ , and the counter output  $Q = Q_C Q_B Q_A = 110$ .

8. The negative edge of the 7th clock pulse causes the  $Q_A$  to go to HIGH, which is a positive transition pulse, which does not affect the  $Q_B$  and hence  $Q_C$  also. After the 7th clock pulse, the outputs are  $Q_A = 1$ ,  $Q_B = 1$  and  $Q_C = 1$ , then the counter output  $Q = Q_C Q_B Q_A = 111$ .
9. The negative-going edge of the 8th clock pulse causes  $Q_A$  to LOW, which is a negative transition pulse, changes the output of flip-flop to LOW and which is also a negative transition pulse applied as a clock to the flip-flop C. Hence  $Q_C$  changes from HIGH to LOW. After the 8th clock pulse, the outputs are  $Q_A = 0$ ,  $Q_B = 0$  and  $Q_C = 0$ . At the stage, the counter output  $Q = Q_C Q_B Q_A = 000$ .

Note that the counter has now recycled back to its initial state, all three flip-flops are in RESET. The operation of the 3-bit ripple counter is shown in the table below.

Count Sequence			Conditions for Complementing
$Q_C$	$Q_B$	$Q_A$	Flip-Flops
0	0	0	Initial condition
0	0	1	At the end of 1st clock pulse negative edge transition.
0	1	0	( $Q_A$ changes from 0 to 1) At the end of 2nd clock pulse, negative edge transition $Q_A$ changes from 1 to 0 and $Q_B$ from 0 to 1.
0	1	1	At the end of 3rd clock pulse $Q_A$ changes from 0 to 1, and no change in $Q_B$ and $Q_C$ .
1	0	0	At the end of 4th clock pulse, $Q_A$ changes from 1 to 0, $Q_B$ changes from 1 to 0 and $Q_C$ from 0 to 1.
1	0	1	At the end of the 5th clock pulse, $Q_A$ changes from 0 to 1 and no change in $Q_B$ and $Q_C$ .
1	1	0	At the end of the 6th clock pulse, $Q_A$ changes from 1 to 0 and $Q_B$ changes from 0 to 1 and no change in $Q_C$ .
1	1	1	At the end of the 7th clock pulse, $Q_A$ changes from 0 to 1 and no change in $Q_B$ and $Q_C$ .

The timing diagram of the 3-bit ripple counter is shown in Fig. 4.26.

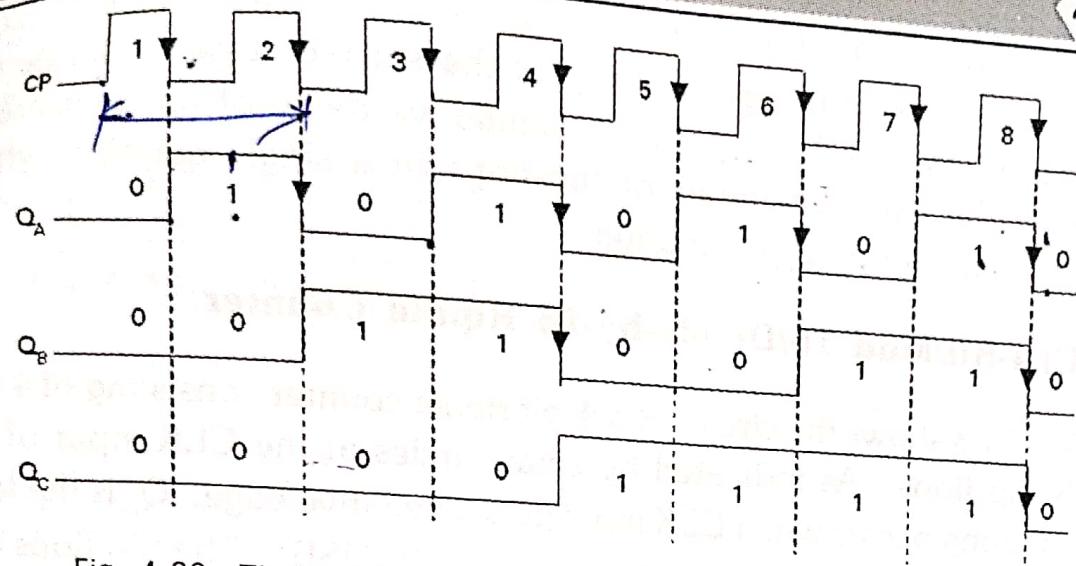


Fig. 4.26 Timing Diagram of the 3-Bit Ripple Counter

**Frequency :** The waveforms in Fig. 4.26 illustrate the fact that the waveform at  $Q_A$  is one-half the frequency of the clock (CP),  $Q_B$  is one-half of the  $Q_A$  and is one-fourth of the clock frequency, or  $Q_C$  is one-eighth of the clock frequency or one-fourth of the  $Q_A$  or one-half of the  $Q_B$ .

If  $CP = 1000$  Hz, then  $Q_A = 500$  Hz,  $Q_B = 250$  Hz and  $Q_C = 125$  Hz. Hence the 3-bit ripple counter is also called as divide-by-8-counter.

**Conclusion :** For every negative edge of the clock there is a change in  $Q_A$ , either 0 to 1 or 1 to 0. So, the  $Q_A$  output frequency is one-half of the clock pulse. For every negative edge of the  $Q_A$ , there is a change in  $Q_B$  either 1 to 0 or 0 to 1. So  $Q_B$  output frequency is one-half of  $Q_A$ . For every negative edge of  $Q_B$  pulse, there is a change in  $Q_C$ .

In Fig. 4.26 timing diagram, we have not considered the propagation delay of flip-flops, for simplicity. If we consider the propagation delay of the flip-flops, we get this timing diagram as shown in Fig. 4.27.

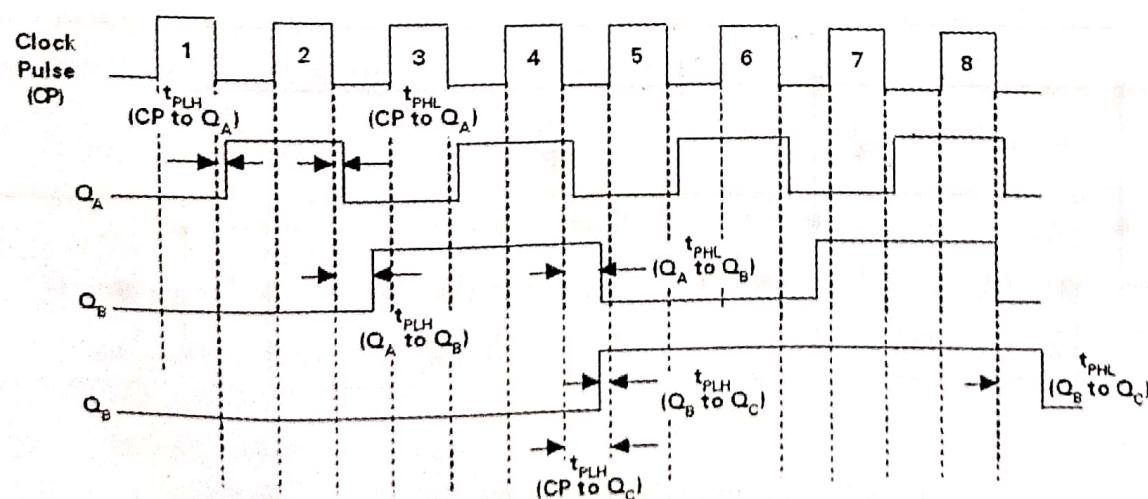


Fig. 4.27 Timing Diagram of a 3-Bit Ripple Counter with Propagation Delay of Each Flip-Flop

(A.P)

(T.S)

From the Fig. 4.27 we can see that the propagation delay of the first stage is added in the propagation delay of the second stage to decide the transition time for the third stage. This cumulative delay of an asynchronous counter limits the use of number of flip-flops in a single counter, which limits the maximum count of the counter.

### (C) 4-Bit/Mod 16/Divide-by-16 Ripple Counter :

Fig. 4.28 shows the circuit of a 4-bit ripple counter consisting of 4 edge triggered JK flip flops. As indicated by small circles at the CLK input of flip flops, the triggering occurs when CLK input gets a negative edge.  $Q_0$  is the least significant bit (LSB) and  $Q_3$  is the most significant bit (MSB). The flip flops are connected in series. The  $Q_0$  output is connected to CLK terminal of second flip flop. The  $Q_1$  output is connected to CLK terminal of third flip flop and so on. By adding more flip flop, a counter of any length can be built. It is known as a ripple counter because the carry moves through the flip flops like a ripple on water. Initially, CLR is made low and all flip flops reset giving an output  $Q = 0000$ . When CLR becomes high, the counter is ready to start. As LSB receives its clock pulse, its output changes from 0 to 1 and the total output  $Q = 0001$ . When second clock pulse arrives,  $Q_0$  resets and carries (i.e.,  $Q_0$  goes from 1 to 0 and, second flip flop will receive CLK input). Now the output is  $Q = 0010$ . The third CLK pulse changes  $Q_0$  to 1 giving a total output  $Q = 0011$ . The fourth CLK pulse causes  $Q_0$  to reset and carry and  $Q_1$  also resets and carries giving a total output  $Q = 0100$  and the process goes on. The action is shown in table below. The number of output states of a counter are known as modulus (or mod). A ripple counter with 4 flip flops can count from 0 to 15 and is, therefore, known as mod-16 counter while one with 6 flip flops can count from 0 to 63 and is a mod-64 counter and so on.

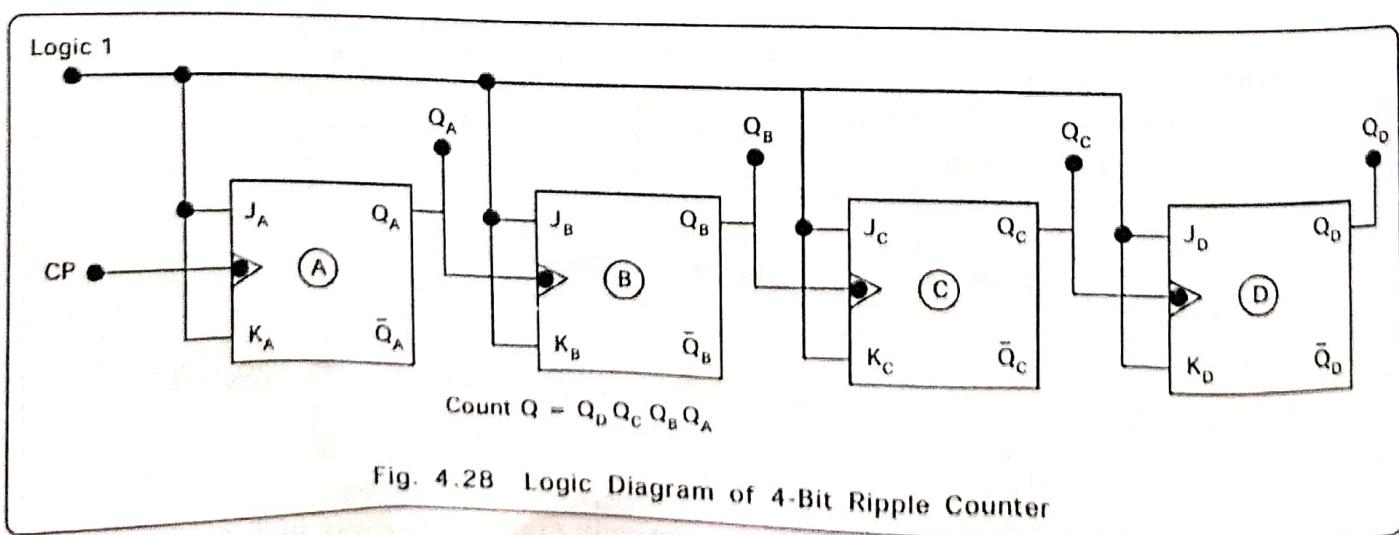


Fig. 4.28 Logic Diagram of 4-Bit Ripple Counter

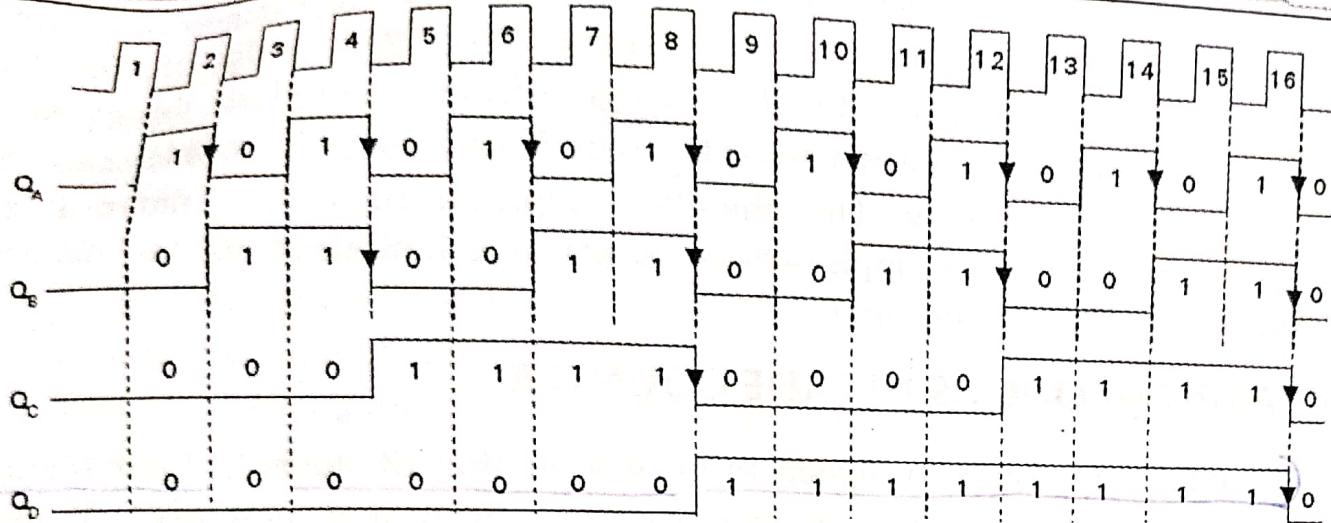


Fig. 4.29 Timing Diagram for 4-Bit Ripple Counter

**Truth Table**

Count	Q <sub>D</sub>	Q <sub>C</sub>	Q <sub>B</sub>	Q <sub>A</sub>	Count	Q <sub>D</sub>	Q <sub>C</sub>	Q <sub>B</sub>	Q <sub>A</sub>
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

Ripple counters are simple to fabricate but have the problem that the carry has to propagate through a number of flip flops. The delay times of all the flip flops are added. Therefore, they are very slow for some applications. Another problem is that unwanted pulses occur at the output of gates.

The timing diagram is shown in Fig. 4.29. FF<sub>0</sub> is LSB flip flop and FF<sub>3</sub> is the MSB flip flop. Since FF<sub>0</sub> receives each clock pulse, Q<sub>0</sub> toggles once per negative clock edge as shown in Fig. 4.29. The remaining flip flops toggle less often because they receive negative clock edge from preceding flip flops. When Q<sub>0</sub> goes from 1 to 0, FF<sub>1</sub> receives a negative edge and toggles. Similarly, when Q<sub>1</sub> changes from 1 to 0, FF<sub>2</sub> receives a negative edge and toggles. Finally when Q<sub>2</sub> changes from 1 to 0, FF<sub>3</sub> receives a negative edge and toggles. Thus whenever a flip flop resets to 0, the next higher flip flop toggles.

(T.S)

From the above discussion it is evident why this counter is known as ripple counter. As the 16th clock pulse is applied, the trailing edge of 16th pulse causes a transition in each flip flop.  $Q_0$  goes from High to Low, this causes  $Q_1$  to go from High to Low which causes  $Q_2$  to go from High to Low which causes  $Q_3$  to go from High to Low. Thus the effect ripples through the counter. It is the delay caused by this ripple which results in a limitation on the maximum frequency of the input signal.

## 4.18 ASYNCHRONOUS DECADE COUNTER

Counters can also be designed to have a number of states in their sequence less than  $2^n$ . The resulting sequence is called a "truncated sequence". To obtain the truncated sequence, it is necessary to force the counter to recycle before going through all of its normal states.

Whenever we want to count to a base N, which is not a power of 2, then we have to consider "divide-by" factor, which is same as the modulus. These counters are called "divide-by-N" counters. The basic counter is modified to produce the required MOD numbers less than  $2^n$  by allowing the counter to "skip states" that are not part of the counting sequence. One of the most common method for doing this is making all flip-flops to reset after the count N with the help of feedback gate in the circuit. Such feedback is provided by a NAND gate, in which the output provides all clear inputs in parallel.

### Design Procedure :

The general procedure for the design of a divide-by-N ripple counter using JK flip-flop with PRESET is described below :

1. Determine the number of flip-flops 'n' required by the equation.  
 $n = \lceil \log_2 N \rceil$   
 where the symbol  $\lceil \log_2 N \rceil$  denotes the smallest integer that is greater than or equal to  $\log_2 N$ .
2. Connect the n flip-flops as a ripple counter.
3. Find the binary representation of  $N - 1$ , where N is the maximum count you need by the counter.
4. Connect all flip-flop outputs that are 1 at the counter ( $N - 1$ ) as inputs to a NAND gate.

Also feed that clock pulse to the NAND gate

5. Connect the NAND gate output to the present inputs of all for which output i.e.,  $Q = 0$  at the count  $N - 1$ .

The counter resets in the following manner :

At the positive-going edge of the Nth clock pulse, all flip-flops are preset to the 1 state. On the trailing edge of the same clock pulse, all flip-flops count to the 0 state (i.e., the counter recycles). This procedure is illustrated by the following example.

At the positive going edge of the Nth clock pulse, all flip-flops are preset to the 1 state. On the trailing edge of the same clock pulse, all flip flops count to the 0 state (i.e., the counter recycles)

### **One more popular method for designing the ripple counter whose count is $<2^n$ is by using the clear input.**

1. Determine the number n of the flip-flop required by the equation  $n = \lceil \log_2 N \rceil$ , where N = number of states in the counter and N – 1 is the maximum count of the counter and  $\lceil \log_2 N \rceil$  denotes the smallest integer that is greater than or equal to  $\log_2 N$ .
2. Connect the n-flip-flops as a ripple counter.
3. Find the binary representation of N.
4. Connect all flip-flops outputs that are 1 at the count N of the counter, as an input to the NAND gate.
5. Connect the NAND gate output to the clear inputs of all the flip-flops.

The counter resets in the following manner.

As long as the NAND gate output is HIGH (the output of NAND gate is HIGH for all counts except for the count  $= N$ ), it will have no effect on the counter. When it goes LOW (at the count  $= N$ ) however, it will clear all the flip-flops so that the counter immediately goes to the 000 state.

### **MOD-10 Ripple Counter/Decade Counter (or) BCD Counter :**

A decimal counter follows a sequence of ten states and returns to 0 after the count of 9. The counters with ten states in their sequence called are 'decade counters'. These counters are useful in display applications in which BCD is required for conversion to a decimal read out. A counter with a count sequence of 0 (0000) through 9 (1001) is called a BCD counter, because its ten states sequence is the BCD code).

A four-bit asynchronous decade counter with four JK flip-flops is shown in Fig. 4.30.

#### **Operation :**

1. The operation of the decade counter is performed by application of ten clock pulses. The clock input is connected to the first flip-flop only. The second, third and fourth flip-flops are triggered by the outputs  $Q_A$ ,  $Q_B$  and  $Q_C$  respectively. The output Q of the counter is given by

$$Q = Q_D Q_C Q_B Q_A$$

(T.S)

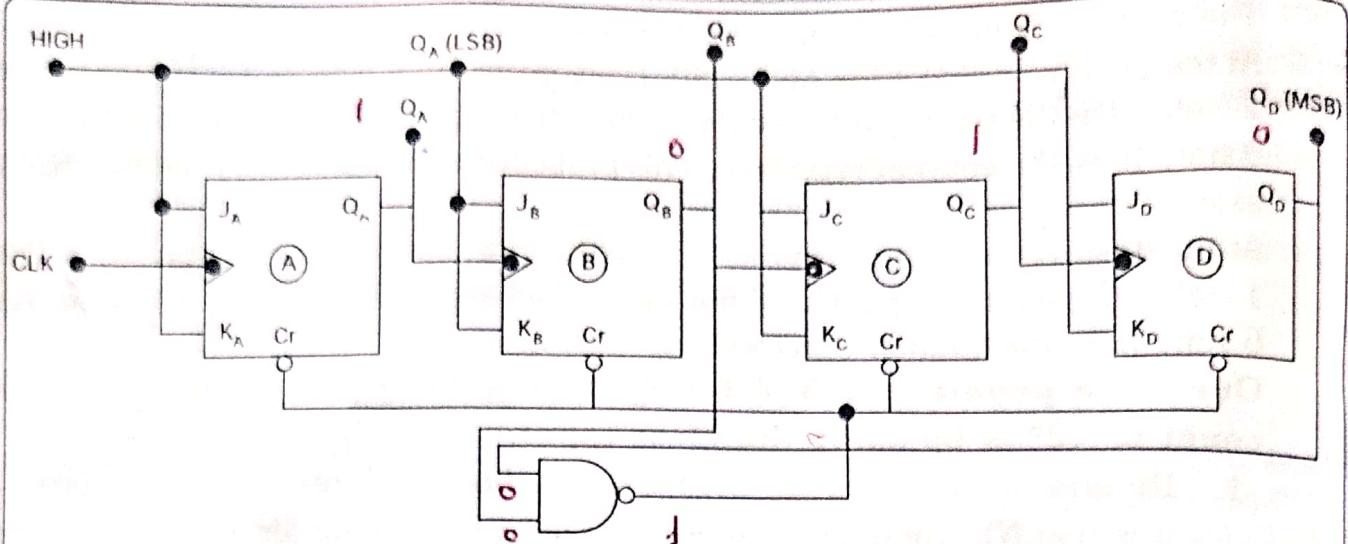


Fig. 4.30 MOD-10 Ripple Counter

2. The timing diagram of the MOD-10 is shown in Fig. 4.31.

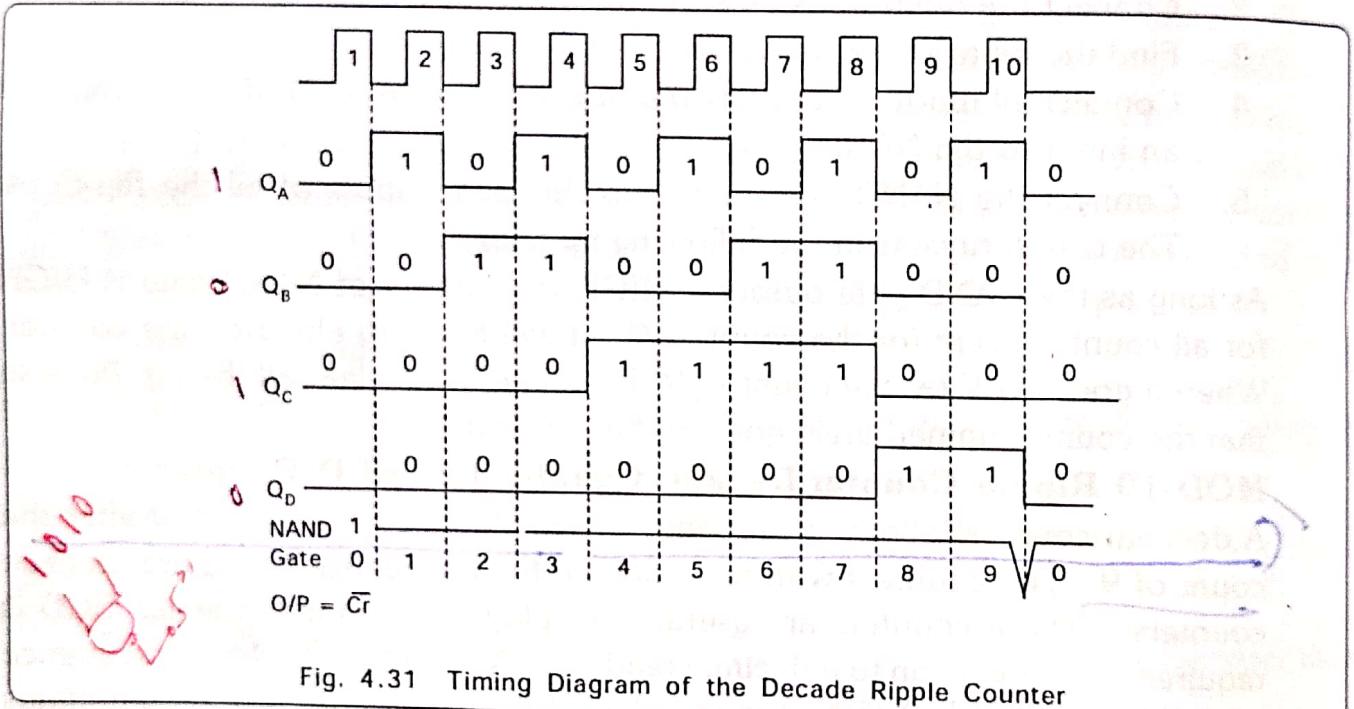


Fig. 4.31 Timing Diagram of the Decade Ripple Counter

From the timing diagram, we notice that output HIGH-to-LOW transition of  $Q_A$  occurs only one delay time after the clock pulse. The HIGH-to-LOW transition of  $Q_B$  occurs one delay time after  $Q_A$ . The LOW-to-HIGH transition of  $Q_C$  occurs one delay time after  $Q_B$ . The  $Q_D$  occurs only one delay time after  $Q_C$ . In this manner, the counting process continues from the count 0000 to 1001.

3. When the counter goes to count 1010, the NAND gate output goes LOW and all flip-flops are in clear condition. Thus, when 10th clock pulse occurs, the counter output Q is 0000 instead of 1010.

The BCD counter of Fig. 3.46 is a decade counter, since it counts from 0 to 9. To count in decimal from 00 to 99, we need a two decade counter. To count from 000 to 999, we need a three-decade counter. Multiple decade counters can be constructed by connecting BCD counters in cascade, for each decade. In the three-decade counter the inputs to the second and third decade counter come from  $Q_D$  of the previous decade. When  $Q_D$  in one decade goes from 1 to 0, it triggers the count for the next higher-order decade while its own decade goes from 9 to 0. For instance, the count after 499 will be 500.

## 4.19 ADVANTAGES AND DISADVANTAGES OF RIPPLE COUNTERS

### **Advantages :**

1. Very simple and straight forward operation and construction and these counters are constructed by suitably interconnecting a number of JK or D flip-flops.
2. They require fewer components.
3. Very easy to design any MOD-counter.

### **Disadvantages :**

1. The speed of operation is low, because each flip-flop in the counter is not clocked simultaneously. So, these counters are used where the speed of operation is not of particular importance.
2. The propagation delay is the major drawback, because it limits the rate at which the counter can be clocked and creates decoding problems.
3. To obtain a truncated sequence, it is necessary to force these counters to recycle before going through all of its normal states.

## 4.20 SYNCHRONOUS COUNTERS

[March 2009]

In a "synchronous counter", all the flip-flops change their state simultaneously, the operation of each stage being initiated by the clock. This is done by connecting the input clock to each flip-flop of the counter, hence all flip flops are clocked simultaneously. These counters are classified according to (a) sequence of states, (b) number of states or (c) number of flip-flops (stages) used in the counter. Here, all flip-flops are triggered simultaneously. Hence, these counters are called "Parallel Counters".

**Synchronous binary counter :** In a synchronous binary counter, we must note the following :

(A.P)

[T.S]

1. Clock pulse is applied common to all flip flops.
2. The flip flop in the lowest order position is complemented with every clock pulse and the JK inputs are maintained at logic 1 (to get the toggling action).
3. The flip flop in any other position is complemented with a pulse, provided all the bits in the lower order positions are equal to 1.
4. The binary count dictates the next bit to be complemented.

Let us illustrate the above procedure with an example of 2-bit synchronous counter.

#### (A) 2-Bit Synchronous Counter :

Two bit synchronous counter is shown in Fig. 4.32. It has two JK flip flops, namely A and B. Both the negative edge triggered flip flops.

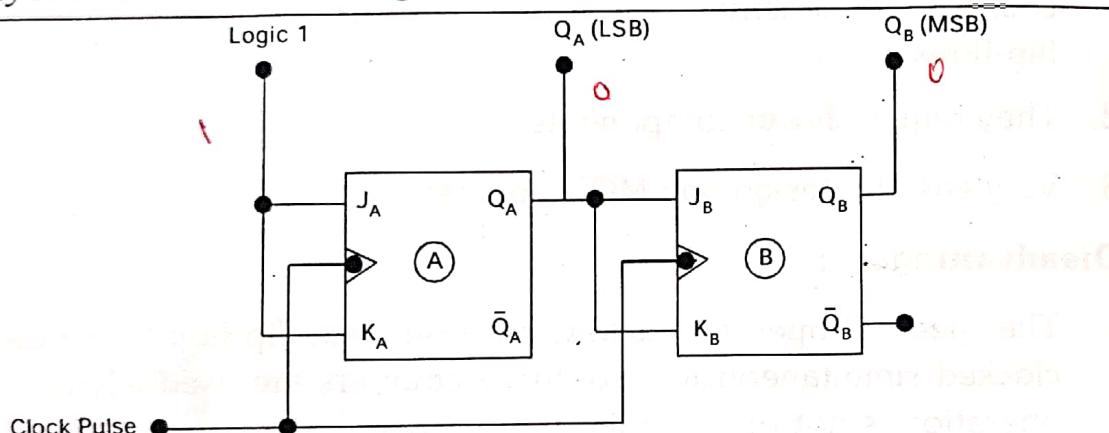


Fig. 4.32 Two Bit Synchronous Counter

Here, the clock signal is connected in parallel to clock inputs of both the flip flops. But the  $Q_A$  output of the first stage is used to drive the J and K inputs of the second stage (flip flop B).

**Operation :** Initially assume that  $Q_A = Q_B = 0$ . For the first clock pulse negative edge, the flip-flop A will toggle, because  $J_A = K_A = 1$ , whereas flip flop B output will remain zero, because  $J_B = K_B = 0$  (since upto this pulse that  $Q_A = 0$ ) and  $Q_A = 1$  only at the end of the pulse i.e., negative edge of the pulse). So, after the first clock pulse,  $Q_A = 1$  and  $Q_B = 0$ .

At the negative going edge of the second, both flip flops will toggle because they both have a toggle condition on their J and K inputs (i.e.,  $J_A = K_A = J_B = K_B = 1$ ). Thus after second clock pulse,  $Q_A = 0$  and  $Q_B = 1$ .

At the negative going edge of the third clock pulse, flip-flop A toggles making  $Q_A = 1$ , but flip flop B remains set i.e.,  $Q_B = 1$ . Hence, at the end of the third clock pulse  $Q_A = Q_B = 1$ .

Finally, at the leading edge of the fourth clock pulse, both flip flops toggle as their JK inputs are at logic 1. This results in  $Q_A = Q_B = 0$  and the counter is recycled back to its original state. The timing details are shown in Fig. 4.33 and 4.34.

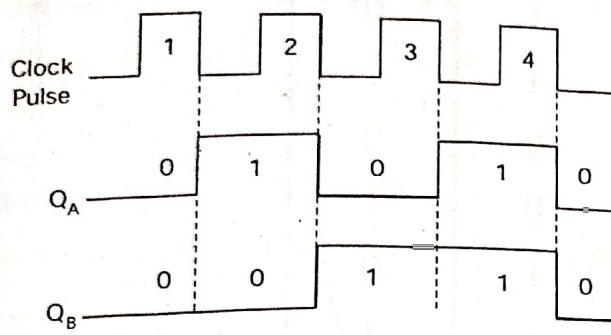


Fig. 4.33 Timing Diagram of 2-Bit Synchronous Counter

CP	$Q_B$	$Q_A$
0, 4	0	0
1, 5	0	1
1, 6	1	0
3, 7	1	1

Fig. 4.34 State Sequence of 2-Bit Synchronous Counter

### (B) 3-Bit Synchronous Binary Counter :

Fig. 4.35 shows 3-bit synchronous binary counter and its timing diagram in Fig. 4.36. The state sequence for this counter is shown in Fig. 4.37.

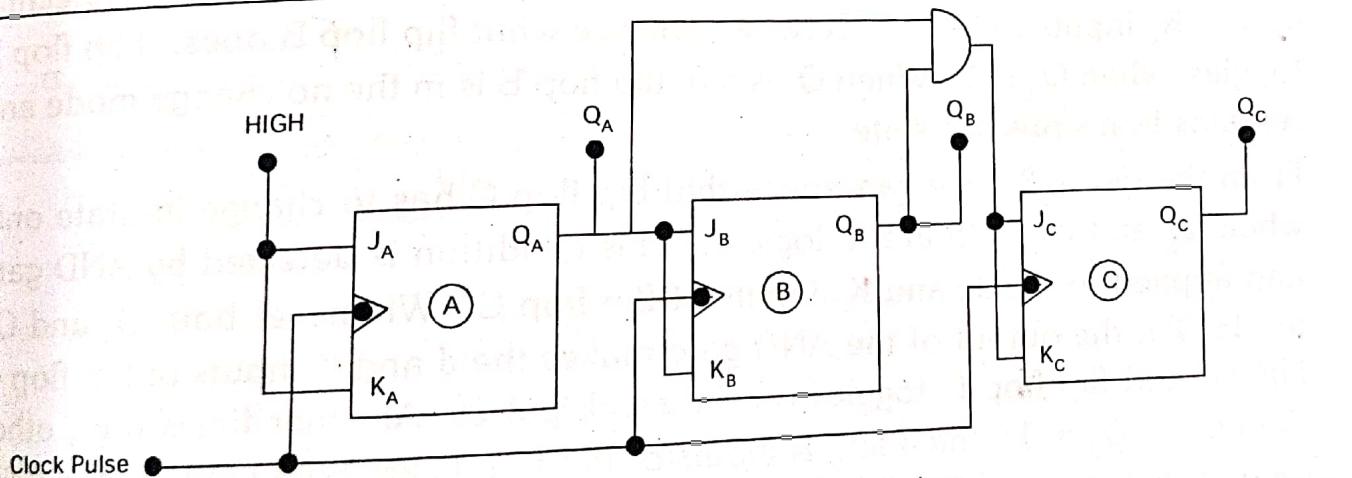


Fig. 4.35 3-Bit Synchronous Counter

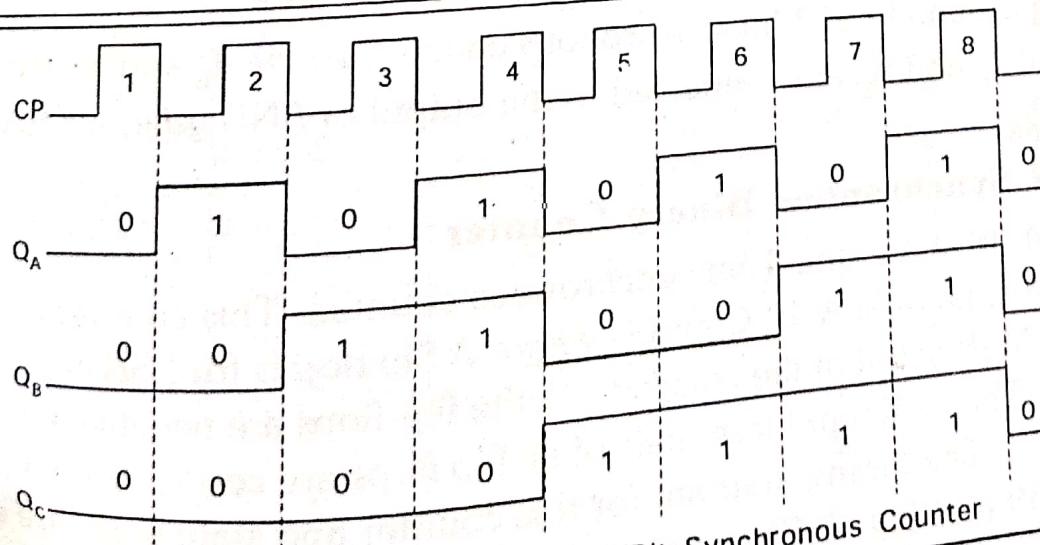


Fig. 4.36 Timing Diagram for 3-Bit Synchronous Counter

(T.S)

CP	$Q_C$	$Q_B$	$Q_A$
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

Fig. 4.37 State Sequence for 3-Bit Binary Counter

Looking at Fig. 4.36 we can see that  $Q_A$  changes on each clock pulse as we progress from its original state to its final state and then back to its original state.

To produce this operation, flip flop A is held in the toggle mode by connecting  $J_A$  and  $K_A$  inputs to HIGH. Now, we will see what flip flop B does. Flip flop B toggles, when  $Q_A$  is 1. When  $Q_A$  is a 0, flip flop B is in the no change mode and remains in its present state.

From the Fig. 4.37, we can notice that flip flop C has to change its state only when  $Q_B$  and  $Q_A$  both are at logic 1. This condition is detected by AND gate and applied to the  $J_C$  and  $K_C$  inputs of flip flop C. Whenever both  $Q_A$  and  $Q_B$  are HIGH, the output of the AND gate makes the J and K inputs of flip flop C HIGH, and flip flop C toggles on the clock pulse. At other times (i.e., other than  $Q_A = Q_B = 1$ ), the J and K inputs of flip flop C are held LOW by the AND gate output, and the flip flop does not change its state.

In the logic diagram of 2-bit synchronous counter, so the  $J_B$  and  $K_B$  are connected by  $Q_A$  and  $J_C$  and  $K_C$  are connected to the output of AND gate, whose inputs are  $Q_A$  and  $Q_B$ .

### (C) 4-Bit Synchronous Binary Counter :

The Fig. 4.38 shows the 4-bit synchronous counter. This counter contains four JK flip flops, namely A, B, C and D where A flip flop is for LSB count and D flip flop is for MSB count of the counter. All the flip-flops are negative edge triggered flip flops. The CLK input terminals of all flip flops are connected to the common clock pulse. The timing diagram for this counter and state sequence are shown in Fig. 4.39 and Fig. 4.40 respectively.

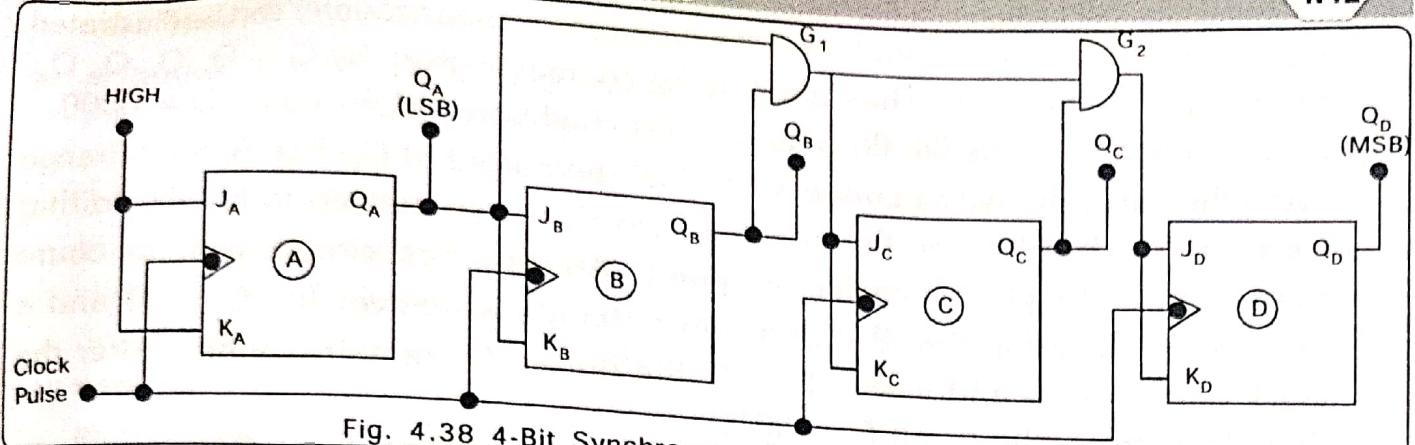


Fig. 4.38 4-Bit Synchronous Counter (Ripple Carry)

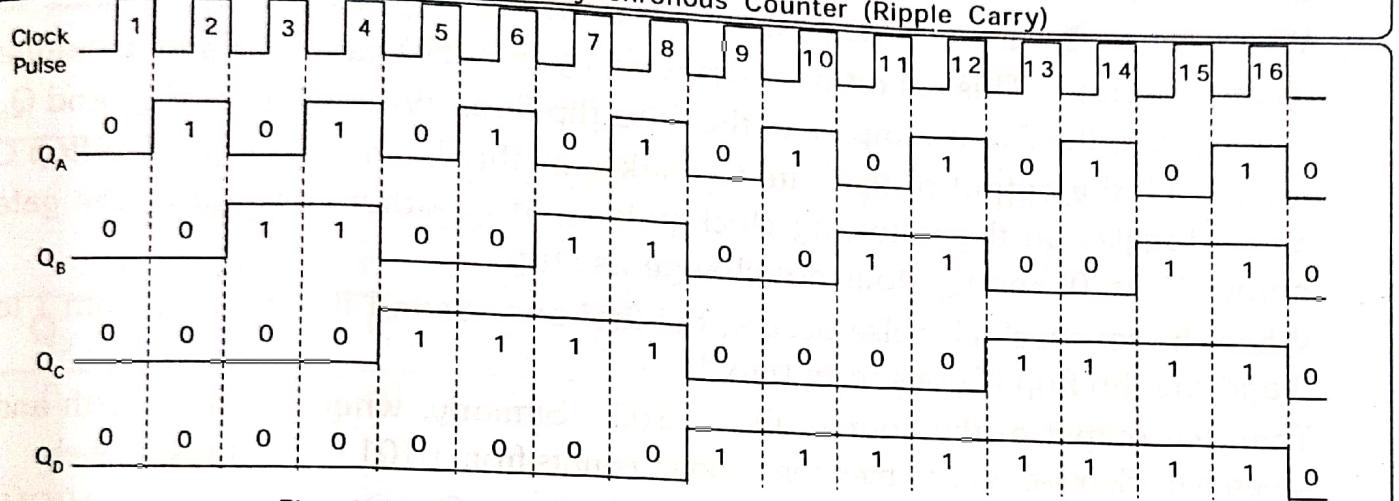


Fig. 4.39 Timing Diagram of 4-bit Synchronous Counter

CP	$Q_D$	$Q_C$	$Q_B$	$Q_A$
0	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	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Fig. 4.40 State Sequence

(A.P)

(T.S)

**Operation :** The basic operation of the 4-bit synchronous counter can be illustrated from its timing diagram. The output of the counter is given by  $Q = Q_D Q_C Q_B Q_A$ . Assume initially, all the flip flops are in reset condition and the count  $Q = 0000$ . From the Fig. 4.40, we can understand that the output of flip flop A must change for every clock pulse. So the inputs  $J_A$  and  $K_A$  are connected to HIGH, so that the output  $Q_A$  toggles on each clock pulse. After the first clock pulse, the count  $Q = 0001$ . The flip-flop B output must change whenever the  $Q_A = 1$  and a clock pulse occurs and when  $Q_A = 0$ , the output  $Q_B$  remains same. After the 2nd clock pulse, the count  $Q = 0010$ .

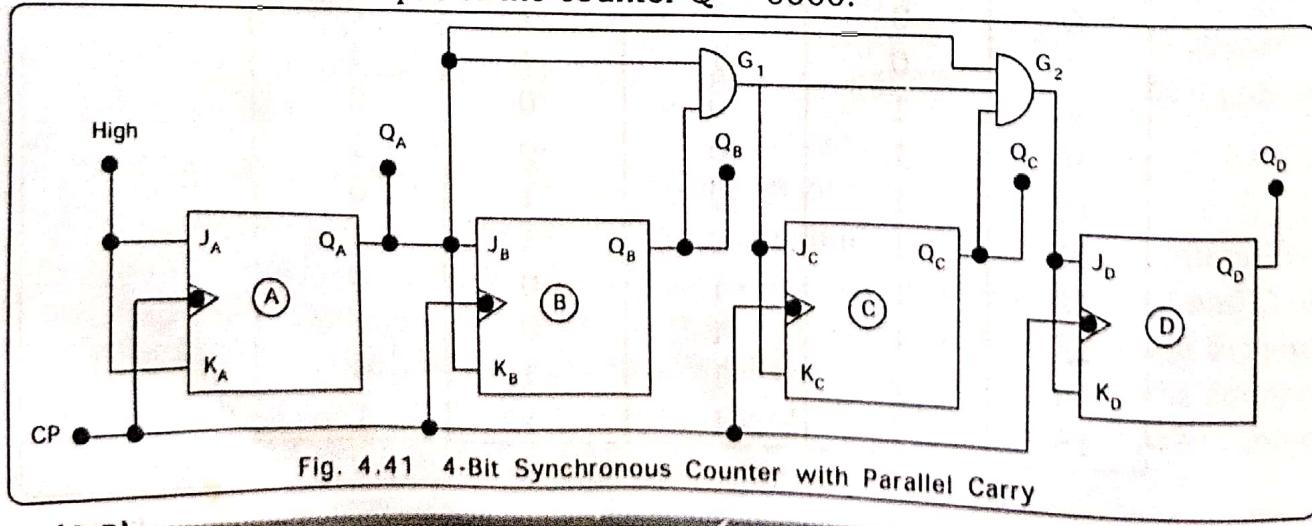
When  $Q_A = Q_B$  are at HIGH, then the third flip flop output  $Q_C$  is made to change its state. This condition is detected by the AND gate ( $G_1$ ) and its output is applied to the J and K inputs of the third flip flop. Whenever both  $Q_A$  and  $Q_B$  are HIGH, the output of the gate  $G_1$  makes as the  $J_C$  and  $K_C$  of the flip flop C and it toggles on the following clock pulse. At all other conditions, the gate output  $G_1$  is '0' and  $Q_C$  does not change its state.

When the fourth clock pulse occurs, the first and second flip flops go from 1 to 0 and the flip flop C goes from 0 to 1.

Then the output of the counter  $Q = 0100$ . Similarly, when the fifth, sixth and seventh clock pulse occurs, then the counter counts from 0101 to 0111 respectively.

From the Fig. 4.41 we can observe that whenever  $Q_A$ ,  $Q_B$  and  $Q_C$  are at HIGH, the fourth flip flop D changes its output. This condition is detected by AND gate ( $G_2$ ), so that when a clock pulse occurs, the D flip-flop will change the state. Note that for all other times, the  $J_D$  and  $K_D$  inputs are in LOW and it is no change condition.

Then, at the end of eight clock pulse the counter output  $Q = 1000$ . Similarly from the ninth clock pulse to fifteenth clock pulse, the counter counts from 1001 to 1111. When the sixteenth clock pulse arrives, the four flip flops change from 1 to 0, then the output of the counter  $Q = 0000$ .



(A.P)

(T.S)

**(D) Mod-5 Synchronous Counter :**

A mod-5 counter will have 5 combinations of the output and will require three flip-flops. This counter will progress through a natural binary count as shown in the truth table corresponding to binary equivalent of decimal numbers 0-1-2-3-4. The truth table is shown in table and provides the basis for implementing the proper connections. In a counter with three flip-flops we will have eight states. In mode-5 counter we use only 0-4 states and the other states 5-6-7 are don't care states. The Karnaugh map for the present values of the inputs for various combinations of present output are shown in Fig. 4.42. The reduced expression obtained from these Karnaugh maps are as follows.

$$J_1 = Q_3 \quad J_2 = Q_1 \quad J_3 = Q_1 \quad Q_2$$

$$K_1 = 1 \quad K_2 = Q_1 \quad K_3 = 1$$

Present flip-flop outputs			Desired flip-flop outputs			Present inputs to flip-flops					
$Q_1$	$Q_2$	$Q_3$	$Q_3$	$Q_2$	$Q_1$	$J_3$	$K_3$	$J_2$	$K_2$	$J_1$	$K_1$
0	0	0	0	0	1	0	x	0	x	1	x
0	0	1	0	1	0	0	x	1	x	x	1
0	1	0	0	1	1	0	x	x	0	1	x
0	1	1	1	0	0	1	x	x	1	x	1
1	0	0	0	0	0	x	1	0	x	0	x

These expressions are realized by using an AND gate and is shown in Fig. 4.43. The working of this circuit can be explained as follows. Initially  $Q_1 = Q_2 = Q_3 = 0$  and  $J_1 = 1$  ( $= \bar{Q}_3 = 1$ ) and  $K = 1$  so this flip-flop A will change its state from 0 to a 1 on the arrival of the first pulse while  $Q_2$  and  $Q_3$  will remain unchanged to a 0. This will force B flip flop to change state on the occurrence of next pulse as B-flip-flop is fed by the output of flip-flop A which has changed to 1 i.e., for flip-flop B,  $J = K = 1$  ( $= Q_1 = 1$ ). On the arrival of second pulse thus  $Q_2$  will become a 1 and  $Q_1$  will become a 0. On the arrival of third pulse only  $Q_1$  will change from a 0 to a 1 and similarly on the arrival of 4th pulse flip flop A and B both will change from a 1 to a 0 and also flip-flop C will change its state from a 0 to a 1. Now when the next pulse arrives only flip-flop C will change which will set the counter to read zero.

$Q_2 Q_1$	00	01	11	10	
$Q_3$	0	1	x	x	1
	0	0	x	x	x

$$J_1 = Q_3$$

(a)

$Q_2 Q_1$	00	01	11	10	
$Q_3$	0	x	1	1	x
	1	x	x	x	x

$$K_1 = 1$$

(b)

$Q_2 Q_1$	00	01	11	10	
$Q_3$	0	0	1	x	x
	0	0	x	x	x

$$J_2 = Q_1$$

(c)

$Q_2 Q_1$	00	01	11	10	
$Q_3$	0	x	x	1	0
	1	0	x	x	x

$$K_2 = Q_1$$

(d)

$Q_2 Q_1$	00	01	11	10
$Q_3$	0	0	1	0
	1	x	x	x

$$J_3 = Q_1 Q_2$$

(e)

$Q_2 Q_1$	00	01	11	10
$Q_3$	0	x	x	x
	1	1	x	x

$$K_3 = 1$$

(f)

Fig. 4.42 Karnaugh Map for Mode-5 Counter

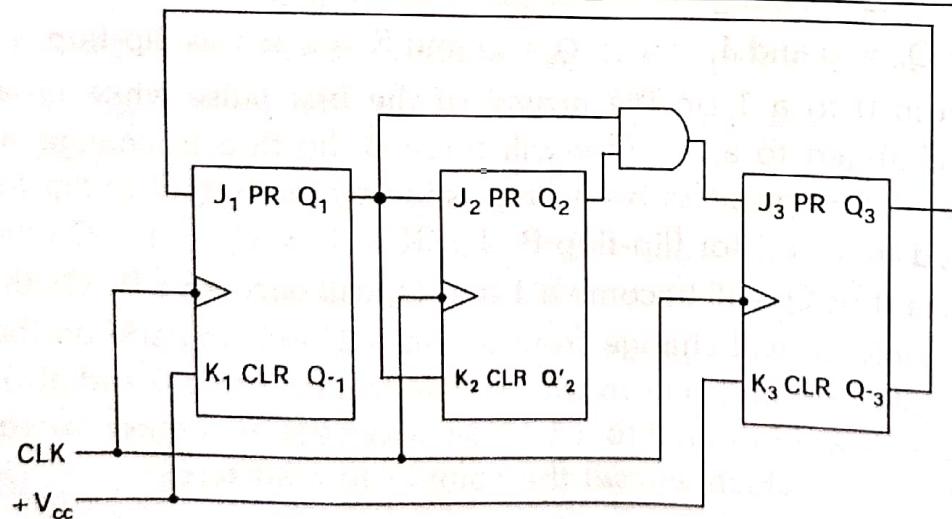


Fig. 4.43 Mod-5 Counter

(A.P)

(T.S)

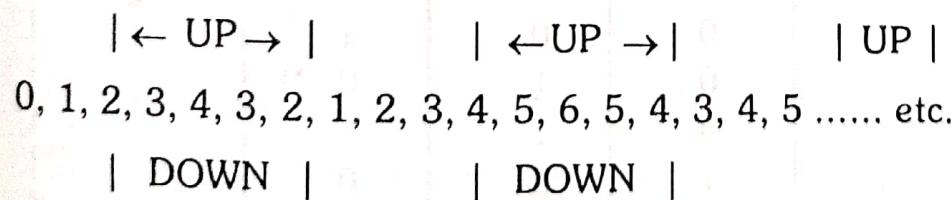
## 4.21 DIFFERENCES BETWEEN SYNCHRONOUS AND ASYNCHRONOUS COUNTERS

Asynchronous Ripple Counters	Synchronous Counters
1. The Delay time of all flip flops are added. Therefore, there is considerable propagation delay.	The clock pulses are applied to all flip flops simultaneously. Hence there is minimum propagation delay.
2. Frequency of operation is lesser than in a synchronous counter.	Frequency of operation can be much higher than that in ripple counter.
3. The maximum frequency depends on modulus.	The maximum frequency does not depend on modulus.
4. Circuit is simple.	Circuit is complex.
5. Minimum number of logic devices are needed.	The number of logic devices required is more than in ripple counter.
6. Standard designs not available.	Standard designs available.
7. Less costly than synchronous counter.	Most costly than ripple counter.

## 4.22 ASYNCHRONOUS 3-BIT UP-DOWN COUNTER

The counter which is capable of progressing in either direction through a certain counting sequence is known as 'up/down counter'. This counter is also known as 'bidirectional counter'. For example, when a 2-bit binary counter is with up/down sequential operations ; then its upward and downward counting sequences are 0, 1, 2, 3 and 3, 2, 1, 0.

In general, most up/down counters can be reversed at any point in their sequence. For example, the 3-bit binary counter can also be illustrated as follows :



### 3-Bit Up / Down Synchronous Counter :

The complete up/down sequence for a 3-bit binary counter is shown in the Table. The arrows indicate the state - to - state movement of the counter for both its UP and DOWN modes of operations.

A basic 3-bit up/down counter is shown in Fig. 4.44. In this counter, the flip-flop in the lower - order position is complemented with every clock pulse.

(A.P)

(T.S)

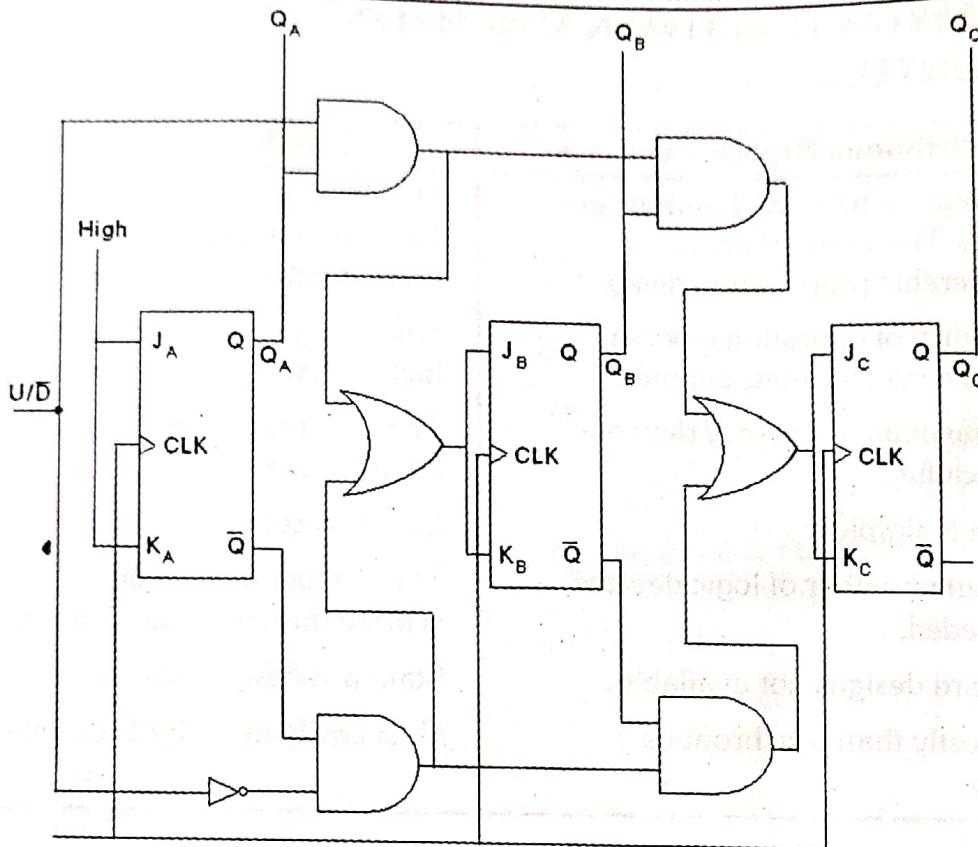


Fig. 4.44 3-Bit Up-Down Synchronous Counter

**Operation :**

From the Fig. 4.44, notice that UP/DOWN is a control input, which determines the basic operation of the counter. When this input is at HIGH, then we have UP mode of operation and when it is at LOW, then we have DOWN mode of operation. The output of the counter is given by  $Q = Q_C Q_B Q_A$ .

**Up/Down Sequence for a 3-Bit Binary Counter**

Clock Pulse	UP	$Q_c$	$Q_b$	$Q_a$	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	

From the table, notice that the output of the first flip-flop  $Q_a$  changes for every clock pulse for both UP and DOWN sequences of the counter. It means that the first flip-flop toggles on each clock pulse. Hence J and K inputs of the first flip-flop are at HIGH ; i.e.,  $J_A = K_A = 1$ .

(i) **UP mode of operation :**

(a) When  $Q_A = 1$ , then the output of the second flip-flop  $Q_B$  changes its state on the next clock pulse. This can be observed from the table. When the first clock pulse occurs,  $Q_A = 1$  and  $Q_B = 0$ . Then  $Q_B$  changes its state from 0 to 1 on the next clock pulse i.e., at the second clock pulse.

(b) The output of the third flip-flop  $Q_C$  changes its state on the next clock pulse when  $Q_A = Q_B = 1$ . This can be observed from the table. The output of the counter is given at the third clock pulse as  $Q_A = 1$ ,  $Q_B = 1$  and  $Q_C = 0$ . Then, when the fourth clock pulse occurs,  $Q_C$  changes its state from 0 to 1.

Thus, the counter shown in the Fig. 4.45, perform the UP mode of operation on the following conditions :

- (1) The control input UP/DOWN must be at HIGH.
- (2) The J and K inputs of the second flip-flop must be equal to 1 under the conditions expressed by the following expression, i.e.,

$$J_B = K_B = Q_A \cdot UP$$

- (3) Similarly the inputs of the third flip-flop must be equal to 1 under the conditions expressed by the following expression, i.e.,

$$J_C = K_C = Q_A \cdot Q_B \cdot UP$$

(ii) **DOWN mode of operation :**

- (a) The output  $Q_B$  changes its state on the next clock pulse when  $Q_A = 0$ .
- (b)  $Q_C$  changes its state on the next clock pulse when  $Q_A = Q_B = 0$ .

The above two changes can be observed from the table.

Thus the counter perform DOWN mode of operation on the following conditions :

- (1) UP/DOWN input must be at LOW.
- (2) The J and K inputs of the second flip-flop must be equal to 1 under the conditions expressed by the following expression, i.e.,

$$J_B = K_B = \bar{Q}_A \cdot DOWN$$

- (3) The J and K inputs of the third flip-flop must be equal to 1 under the conditions expressed by the following expression, i.e.,

$$J_C = K_C = \bar{Q}_A \cdot \bar{Q}_B \cdot DOWN$$

(A.P)

(T.S)

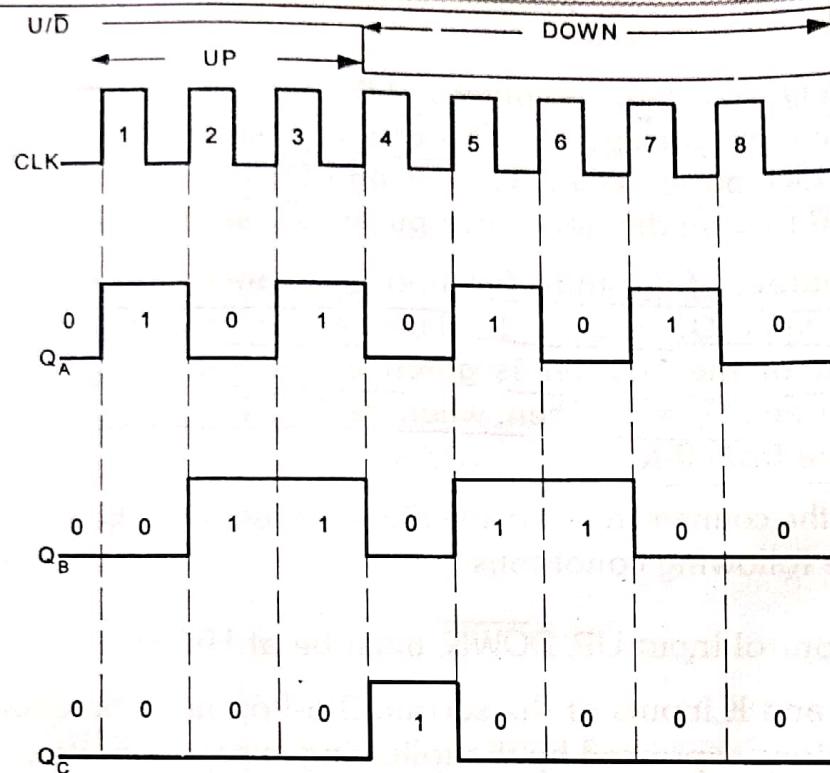


Fig. 4.45 Timing Diagram for Up Down Counter

The UP and DOWN modes of operations are illustrated in the timing diagram, shown in Fig. 4.45. Note that the counter starts in the all 0s state and is positive edge-triggered. Let the UP operation is performed upto the trailing edge of the 4th clock pulse and during the other clock pulses the DOWN operation is performed.

## 4.23 APPLICATIONS OF COUNTERS

Counters are used in a variety of applications. They are used for :

- Direct counting and scaling.
- Measurement of frequency.
- Measurement of time period.
- Measurement of speed.
- Waveform generation.
- Generating timing sequences.
- Frequency dividers.
- Equipment operation sequencing and mathematical manipulation.
- Conversion of analog and digital information from one form to another.

**4.24 FLIP FLOP, REGISTER AND COUNTER ICs****A. Flip-Flop ICs**

[Oct./Nov., 2012, 2013, April 2015]

IC Number	Logic Family	Functional Description
<b>(A) Flip-Flop ICs</b>		
54/7470	TTL	Dual JK with PRESET and CLEAR
54/74H71	TTL	JK Master-Slave with PRESET and CLEAR
54/7472	TTL	JK Master-Slave with PRESET and CLEAR
54/74110	TTL	JK Master, with data lock out
54/74111	TTL	Dual JK Master-Slave with data lock out
54/7473	TTL	Dual JK, negative edge triggered
54/7474	TTL	Dual D, positive edge triggered
54/7475	TTL	Quad D - Latch
54/7476	TTL	Dual JK with SET and CLEAR
54/7478	TTL	Dual JK with RESET
54/74107	TTL	Dual JK with CLEAR
54/74174	TTL	Hex D flip flop with CLEAR
54/74175	TTL	Quad D flip flop with CLEAR
54/74279	TTL	Quad SR latch
4013	CMOS	Dual D
4027	CMOS	Dual JK
4042	CMOS	Quad D Latch
4044	CMOS	Quad RS latch with tristate output
4076	CMOS	Quad D with tristate output
40174	CMOS	Hex D
40175	CMOS	Quad D
MC 10131	ECL	Quad D master-slave
MC 10135	ECL	Dual JK master-slave
MC 1666	ECL	Dual Clocked RS

IC Number	Logic Family	Functional Description
<b>B. Register ICs</b>		
7491	TTL	8-bit shift register, SISO
7494	TTL	4-bit shift register, PISO
7495	TTL	4-bit shift register
7496	TTL	5-bit shift register
74164	TTL	8-bit shift register, SIPO
74165	TTL	8-bit shift register, S/PISO
74195	TTL	4-bit shift register, S/PIS/PO
74194	TTL	4-bit bidirectional universal shift register
74198	TTL	8-bit bidirectional universal shift register
74LS299	TTL	8-bit universal shift register
4014 B	CMOS	8-bit synchronous shift register, S/PISO
4015 B	CMOS	Dual 4-bit shift register, SIPO
4021 B	CMOS	8-bit shift registers S/PISO
4031	CMOS	64-bit shift register
4034	CMOS	8-bit bidirectional shift register
4035	CMOS	4-bit universal shift register
4062	CMOS	200-bit dynamic shift register
4076	CMOS	4-bit D-type register
4094	CMOS	8-bit shift and store bus register
4731	CMOS	Quad 64-bit shift register
40108	CMOS	4 × 4 multiport register
MC 10141	ECL	4-bit universal shift register
<b>C. Counter ICs</b>		
7490	TTL	Decade counter
7492	TTL	Divide by 12 counter
7493	TTL	4-bit binary counter
74142 to 74144	TTL	4-bit counters/latches/drivers

64176, 74177	TTL	Presetable counter/latch
74160 to 74163	TTL	4-bit synchronous counter
74190, 74191	TTL	Synchronous up/down counters
74290	TTL	Decade counter
74390, 74490	TTL	Dual decade counters
4017	CMOS	5-bit Johnson counter
4020	CMOS	14-bit ripple counter
4022	CMOS	Divide by 8 counter
4029	CMOS	Synchronous. Presetable up/down counter
4059	CMOS	Programmable divide by N counter
4510	CMOS	Presetable up/down decade counter
4518	CMOS	Dual 4-bit decade counter
4520	CMOS	Dual 4-bit binary counter
MC 10136	ECL	Universal hexadecimal counter
MC 10137	ECL	Universal decade counter
MC 10154	ECL	4-bit binary counter
MC 10178	ECL	4-bit binary counter