UNIT-5

FREQUENCY DIVISION AND COUNTING

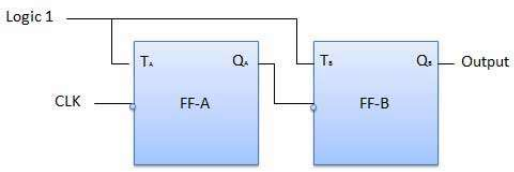
Counter:

A sequential circuit which counts the number of clock pulses is known as 'counter'.

According to the way they are clocked, counters are classified into two types. They are,

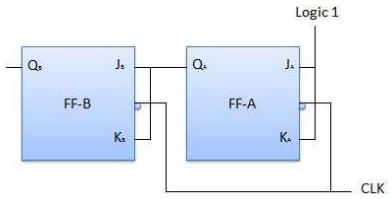
1. Asynchronous counter (or) ripple counter
2. Synchronous counter

Asynchronous Counter:



In asynchronous counter, only the first flip-flop is clocked by an external clock pulse. All subsequent flip-flops are clocked by the output of the preceding flip-flop. Hence, in asynchronous counters, the flip-flops are not clocked simultaneously. Asynchronous counters are also called ripple-counters because of the way the clock pulse ripples it way through the flip-flops.

Synchronous Counter:



In synchronous counter, the clock input is connected to all of the flip-flops and hence they are clocked simultaneously. Synchronous counters are faster than asynchronous counters due to simultaneous clocking of flip-flops.

Distinguish between Asynchronous and Synchronous counters:

	Asynchronous counter	Synchronous counter
1	Only the first flip-flop is clocked by an external clock pulse. All subsequent flip-flops are clocked by the output of the preceding flip-flop.	A single clock pulse is connected to all the flip-flops.
2	Speed of operation is very low.	Speed of operation is very high.
3	Propagation delay is more.	Propagation delay is less.
4	It is called as serial counter.	It is called as parallel counter.
5	Cost is low as less hardware is required.	Cost is more as more hardware is required.

Ripple counter:

Ripple counter is an asynchronous counter. We know that the number of flip-flops used in the counter and the way in which they are connected are to determine the number of states. The number of output states of the counter is called the '*Modulus*' (Mod) of the counter. Note that the maximum possible number of states (maximum modulus) of a couner 2^n, where 'n' is the number of flip-flops in the counter. If we have two flip-flops in a counter, then the maximum possible number of output states of the counter is 2^2 i.e., 4. Now we can call that counter as Mod-4 counter. If we have three number of flip-flops, then the maximum possible number of output states of that counter is 2^3 i.e., 8. Noe we can name that counter as Mod-8 or Modulo-8 counter.

Modulus of a counter:

The number of states in the counter is referred as 'modulus of a counter'. The modulus of a n-bit counter is given by $N = 2^n$

Example : For a 4-bit counter N = 2^4 =16 [$\because$ n=4]

$\therefore$ N=16

Modulo-8 ripple counter:

Modulo-8 ripple counter requires 3 flip-flops.

i.e., Number of flip-flops = $\log_2 N$ [$\because$ N = 8]

= $\log_2 8 = 3$

Modulo-8 ripple counter is a three-bit asynchronous counter, consists of eight states due to its three number of flip-flops. Figure shows the Mod-8 ripple counter, with three numbers of JK flip-flops.

In this counter, the CLK input is connected to the first flip-flop only. The second flip-flop is triggered by the $\overline{Q}_A$ output of the first flip-flop. The third flip-flop is triggered by the $\overline{Q}_B$ output of the second flip-flop.

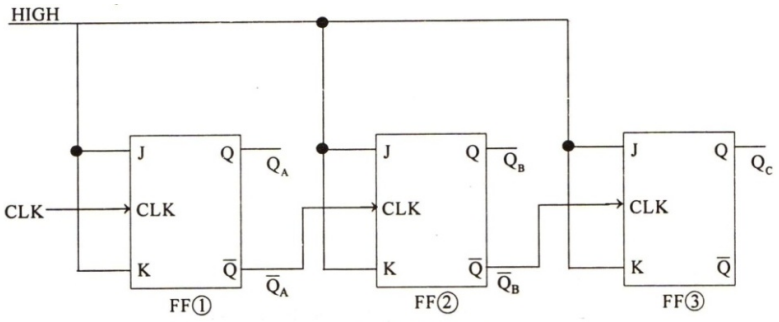


Fig. MOD-8 RIPPLE COUNTER

Note that the first flip-flop changes its state at the positive-going edge of each pulse, but the second and third flip-flops change their states only when triggered by the positive-going transitions of the $\overline{Q}_A$ and $\overline{Q}_B$ outputs respectively. Because of the inherent propagation delay time through a flip-flop; the transition of the input CLK pulse and the transition of the outputs $\overline{Q}_A$ and $\overline{Q}_B$ can never occur at exactly the same time. Therefore, the three 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 some time, due to propagation delays, to reach the last flip-flop. Hence the asynchronous counters are commonly referred to as 'ripple counters'.

Operation:

i) Let us examine basic operation of the counter by applying eight clock pulses one by one to the clock input of the first flip-flop and observe the Q output of the counter at each stage of the application of the CLK pulse. Note that the three flip-flops are connected for toggle operation (J=1, K=1) and are assumed to be initially RESET. Hence at this stage the output of the counter Q = $Q_C\, Q_B\, Q_A$ = 0 0 0.

ii) When the positive-going edge of the *first clock pulse* occurs, the output of the first flip-flop Q_A goes to HIGH. Then at the same time $\overline{Q}_A$ output goes to LOW; but it has no effect on the second and third flip-flops because a positive-going transition must occur to trigger the flip-flop. Thus after the leading edge of clock pulse, Q_A=1, Q_B=0, Q_C=0.

iii) The positive-going edge of the *second clock pulse* causes Q_A to go LOW. Then $\overline{Q}_A$ goes HIGH and triggers the second flip-flop, causing Q_B to go HIGH. Then at the same time $\overline{Q}_B$ goes LOW, but it has no effect on the third flip-flop. After the leading edge of the second clock pulse, Q_A=0, Q_B=1 and Q_C=0. At this stage, the output of the counter is Q = $Q_C\, Q_B\, Q_A$ = 0 1 0.

iv) The positive-going edge of the *third clock pulse* causes Q_A to go HIGH again, Then $\overline{Q}_A$ goes LOW and has no effect on the second and third flip-flops. After the leading edge of the third clock pulse, Q_A=1, Q_B retains it's precious state i.e., Q_B=1 and Q_C=0. Therefore, the output of the counter at this stage is Q = $Q_C\, Q_B\, Q_A$ = 0 1 1.

v) When the *fourth clock pulse* occurs, the third flip-flop takes three propagation delay times to ripple through the counter and causes the change in Q_C from LOW to HIGH. But it has no effect on the first

and second flip-flops. After leading edge of the fourth clock pulse, Q_A=0, Q_B=0 and Q_C=1. The output of the counter at this stage is Q = Q_C Q_B Q_A = 1 0 0.

vi) When the *fifth clock pulse* occurs, the output Q_A goes HIGH $\overline{Q}_A$ goes LOW. Then it has no effect on the second and third flip-flops. But the third flip-flop previous output continues for three propagation times. After leading edge of the fifth clock pulse, Q_A=1, Q_B=0 and Q_C=1. The output of the counter at this stage is Q = Q_C Q_B Q_A = 1 0 1.

vii) When the *sixth clock pulse* occurs, the output Q_A goes LOW $\overline{Q}_A$ goes HIGH. This triggers the second flip-flop; causing Q_B is go HIGH. Q_C continues it's previous state. Therefore, output of the counter at this stage is Q = Q_C Q_B Q_A = 1 1 0.

viii) When the *seventh clock pulse* occurs, the output Q_A goes HIGH $\overline{Q}_A$ goes LOW. Then Q_B and Q_C continue their previous states. Therefore, output of the counter at this stage is Q = Q_C Q_B Q_A = 1 1 1.

ix) The positive-going edge of the *eighth clock pulse* causes Q_A to go LOW. Then $\overline{Q}_A$ goes HIGH and triggers the second flip-flop, causing Q_B to go LOW and even Q_C goes LOW. After the leading edge of eigth clock pulse, Q_A=0, Q_B=0 and Q_C=0. At this stage, the output of the counter is Q=Q_C Q_B Q_A= 0 0 0.

Note that the counter has now recycled back to its original state; i.e., all the three flip-flops are in RESET.

Table

Clock Pulse	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
8	0	0	0

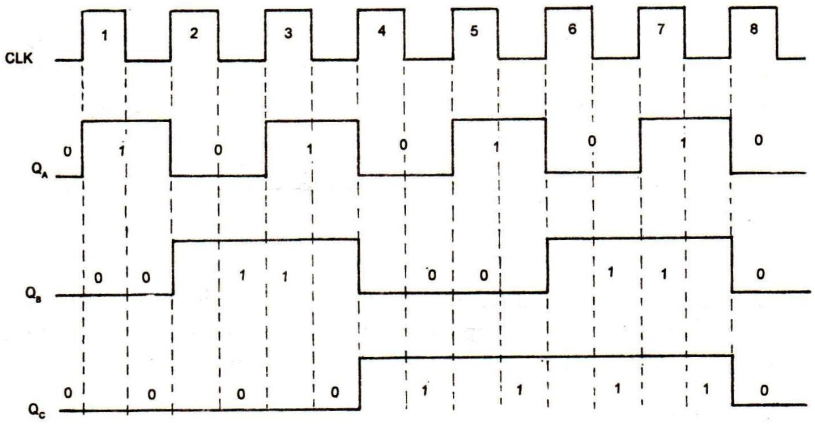


Fig. TIMING DIAGRAM OF MOD-8 RIPPLE COUNTER

Decade Counter (Modulo-10 ripple counter):

Counters with ten states in their sequence are called decade counter. Asynchronous decade counter is also known as BCD counter or a divide-by-10 counter or a Modulo-10 counter. It is the most popularly used counter as it uses decimal number system. The counting sequnece for mod-10 counter is from 0000 to 1001 (0 to 9 decimal numbers). The state diagram of decade counter is shown in figure (1).

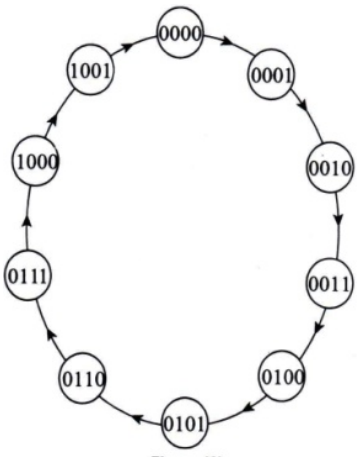


Figure (1)

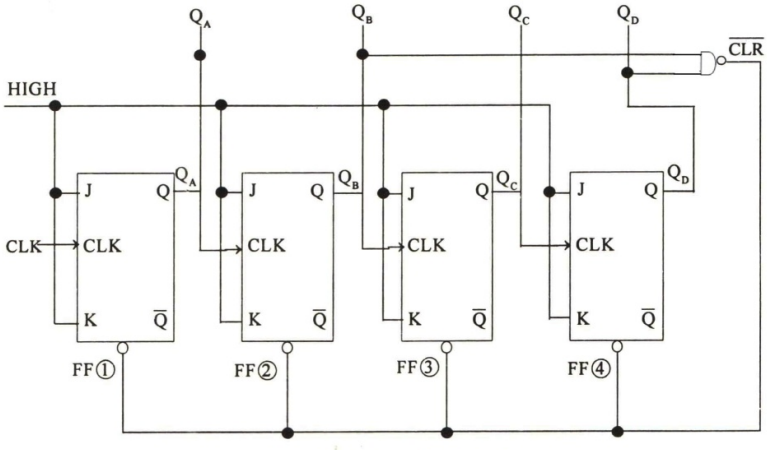


Fig. (2) ASYNCHRONOUS DECADE COUNTER

Decade counter consists of four J-K flipflops and a NAND gate as shown in figure (2). The NAND gate in the figure is used to clear inputs of the flip-flop one the "1010" state is reached. Hence the decade counter must recycle back to the 0000 state after 1001 state.

Operation:

i) The operation of the decade counter is performed by application of ten clock pulses. The CLK 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 of the counter given by Q = Q_D Q_C Q_B Q_A.

ii) The basic operation can be illustrated in the timing diagram, as shown in figure (3).

iii) From the timing diagram; we noticed that the 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 only one delay time after $_{QA}$. The LOW-to HIGH transition of Q_C occurs onle delay time after Q_B. The Q_D occurs only one delay tier after Q_C. In this manner, the counting process will be continued from the count 0000 to 1001.

iv) Notice that only Q_B and Q_D are connected to the NAND gate inputs. The two unique states i.e., Q_B=1 and Q_D=1 are sufficient to decode the count of 1010 because none of the other states (0000 through 1001) have both Q_B and Q_D HIGH at the same time.

Once the counter reaches 1010, the outputs Q_B and Q_D of flip-flops are connected to NAND gate. This gate is used to decode the count 10 (1010) and resets the flip-flops back to '0000'. The truth table of mod-10 decade counter is shown in table.

Clock	Q_3	Q_2	Q_1	Q_0
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	0	0	0	0

Table

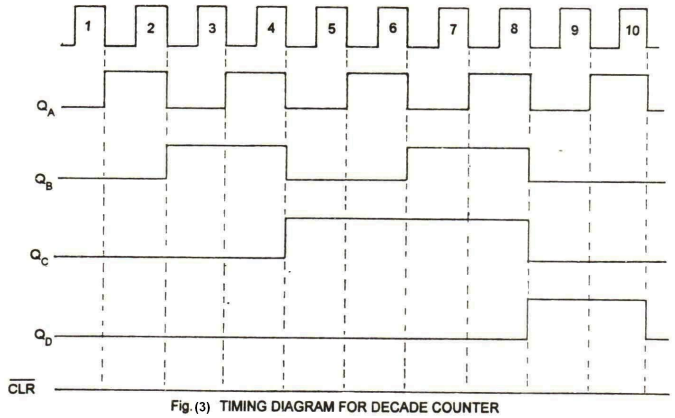


Fig.(3) TIMING DIAGRAM FOR DECADE COUNTER

DRAWBACKS (or) DISADVANTAGES OF RIPPLE COUNTERS:

The main drawbacks of ripple counters are:

1) As the number of bits increases, the number of flip-flops increases. This makes the counting function to be too slow.
2) The *speed of operation is low*, because the flip-flops of these counters are not clocked simultaneously.
3) The *propagation delay time* increases due to transition occuring at the output of previous flip-flop.
4) To obtain a *truncated sequence*, it is necessary to force these counters to recycle before going through all of its normal states.

SYNCHRONOUS COUNTERS:

The term synchronous as applied to counter operation means that the couner is clocked such that each flip-flop in the counter is triggered at the same time. This is done by connecting the clock line to each flip-flop of the counter; hence all the flip-flops are clocked simultaneously. These counters are classified primarily by the sequence of states, by the number of states, or by the number of flip-flops (stages) within the counter. Among various synchronous counters, let us study6 a 4-bit synchronous counter, up/down counter and a 4-bit ring counter; including their operationsand timing diagrams.

Synchronous Binary counters :

Synchronous binary counters can easily be constructed with complementing flip-flops and gates.

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

i) The flip-flop in the lowest-order position is complemented with every pulse.

ii) J and K inputs must be maintained at the logic 1.

iii) The clock pulses are applied to the CLK inputs of all flip-flops.

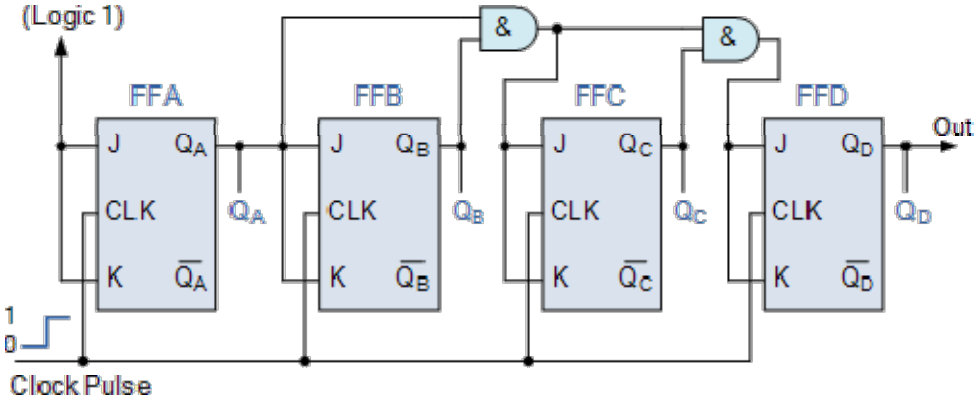
iv) A flip-flop in any other position is complemented with a pulse provided all the bits in the lower-order positions are equal to 1.

v) The binary count dictates that the next higher-order bit should be complemented.

4-BIT SYNCHRONOUS COUNTER:

A counter is said to be synchronous, when a single clock pulse is connected simultaneously to all the flip-flops. A 4-bit synchronous counter requires four flip-flops and two AND gates. The circuit diagram of a 4-bit synchronous counter using JK flip-flop is shown in figure.

In this counter, the JK inputs of FFA is connected to logic '1' which responds for each positive clock edge. The output of FFA is directly connected to the input of FFB. While the outputs Q_A and Q_B are applied as inputs to AND gate G_1 such that the output is indirectly connected to the input of te next flip-flop FFC. Similarly the output of AND gate G_1 and Q_C are applied to AND gate G_2 whose output is connected as input to FFD. This counter counts from 0000 to 1111 for each positive edge pulse.



Operation:

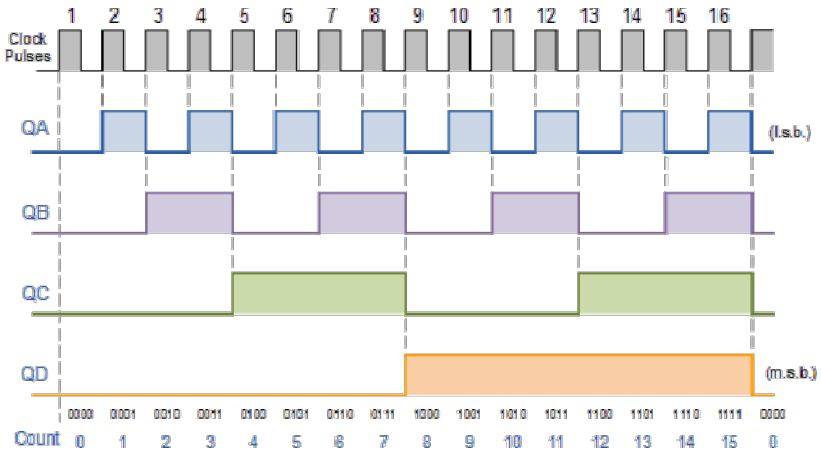
Initially all the flip-flops are reset i.e., 0000 when the first clock pulse is applied to FFA, the output Q_A toggles from 0 to 1 and the remaining outputs are unchanged i.e., 0001. When the second clock pulse is applied, the outputs Q_A and Q_B changes their values and the output becomes 0010. When the third clock pulse is applied, FFA changes its output and the count becomes as 0011. In the same way for each positive edge clock pulse FFA toggles. FFB toggles when Q_A=1, FFC toggles when Q_A=Q_B=1 and FFD toggles only when Q_A=Q_B=Q_C=1. The counting sequence of 4-bit synchronous counter is shown in table.

Clock	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	1
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
16	0	0	0	0

16th clock pulse

Table

The timing diagram of a 4-bit synchronous counter is shown in figure (b).



UP/DOWN COUNTER:

The counter which is capable of processing 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:

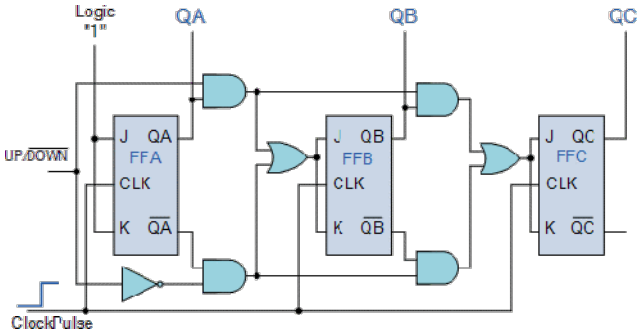
$$\underset{\text{up}}{|\leftarrow up \rightarrow|} \quad \underset{\text{up}}{|\leftarrow \ up \rightarrow|} \quad \underset{\text{up}}{|\ up\ |}$$

$$0,1,2,3,4,3,2,1,2,3,4,5,6,5,4,3,4,5\ldots\text{etc}\ldots.$$

$$\underset{\text{down}}{|\text{down}|} \quad \underset{\text{down}}{|\text{down}|}$$

3-bit up/down synchronous counter:

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

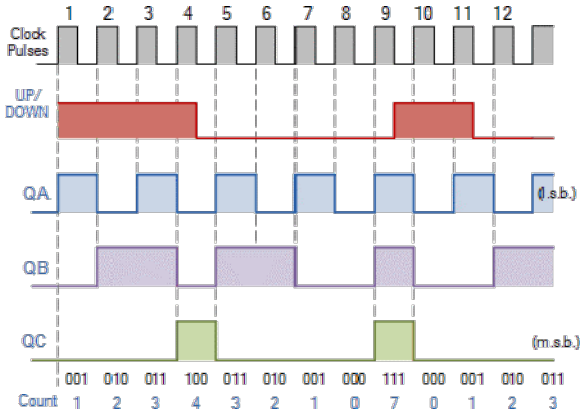


Operation:

In this counter UP/DOWN is a control input, which determines the basic operation of the counter. When this input is at HIGH, counter works as up counter and count values from 000 to 111. When this input is at LOW, counter works as down counter and counts values in reverse direction of up counter i.e., from 111 to 000.

The output of the counter is given by $Q = Q_C Q_B Q_A$.

From the table, 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=K=1.



Clock Pulse	UP	QC	QB	QA	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	

Table : UP/DOWN sequence for a 3-bit binary counter.

a) UP mode of operation:

i) When Q_A=1, then the output of the second flip-flop Q_B changes its state on the next pulse. This can be observed fronm the above table. When the first clock pulse occurs, Q_A=1 and Q_B=0.

ii) When the second clock pulse occurs, Q_A=0 and Q_B=1.

iii) When the third clock pulse occurs, Q_A=1, Q_B=1 and Q_C=0.

iv) When the fourth clock pulse occurs, Q_A=0, Q_B=0 and Q_C=1.

Thus the counter perform the UP mode of operation of the following conditions:

1) The control input $UP/\overline{DOWN}$ 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 $J_B=K_B=Q_A.UP$
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., $J_C=K_C=Q_A.Q_B.UP$

b) DOWN mode of operation:

i) The output Q_B changes its state on the next clock pulse when Q_A=0.

ii) 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 above table.

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

1) $UP/\overline{DOWN}$ 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 J_B=K_B= $\overline{Q_A}$ .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= $\overline{Q_A}$. $\overline{Q_B}$ .DOWN

The UP/DOWN modes of operation are illustrated in the timing diagram as shown in the above figure. Note that the counter starts in all 0s and is positive 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.

NEED FOR A PROGRAMMABLE COUNTER USING FLIP-FLOPS:

A programmable counter is a synchronous circuit in which the initial value of the counter can be predefined.

Need for Programmable Counter:

1) Programmable counter allows to set the modulo nymber according to the application.
2) It is used to generate an accurate time delay.

OPERATION OF A PROGRAMMABLE COUNTER USING FLIP-FLOPS:

Programmable counter or presettable counter is a synchronous counter containing predefined initial value. The circuit of programmable counter using flip-flops is as shown in figure.

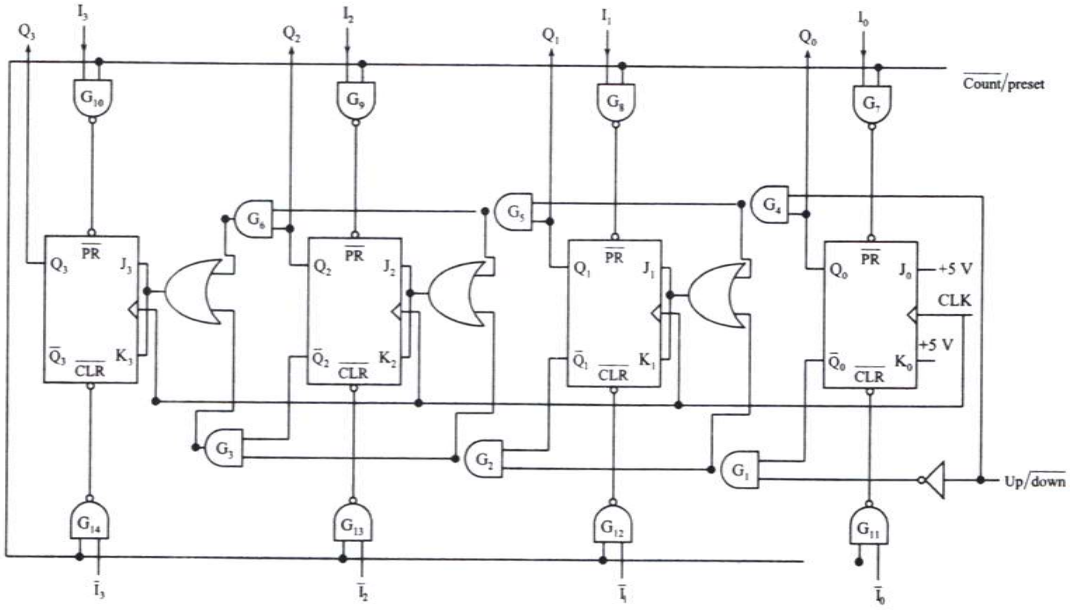


Figure: Programmable Counter

In the figure I_0, I_1, I_2 and I_3 are the initial values given to the flip-flops. The programmable counter is identical to the working of up-down counter mode but additionally 6 NAND gates are provided. These NAND gates preload the desired initial values into the flip-flop.

When $\overline{Count}/\text{Pr}eset=0$ then $\overline{CLR}=\overline{PR}=0$. At this mode, the counter starts counting pulses as no set and reset occurs.

When $\overline{Count}/\text{Pr}eset=1$ the NAND gates G_7 to G_{14} are enabled and produces outputs, based on initial values I_0 to I_3.

If I_0=1 then G_7=0 and G_{11}=1

$\therefore$ Q_0=1 (Set)

If I_1=0 then G_7=1 and G_{11}=0

$\therefore$ Q_0=0 (Reset)

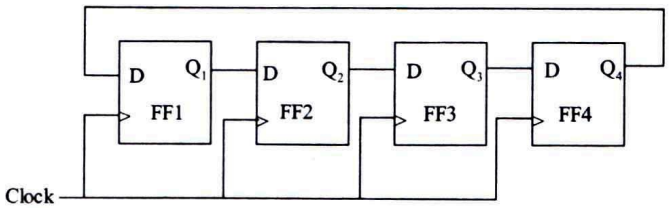
Similarly, the corresponding flip-flops get set or reset based on I_1, I_2 and I_3.

RING COUNTER :

Timing signals that control the sequence of operations are essential in digital computer systems. The counter which is used for their basic timing is known as a 'Ring Counter'. A ring counter is a circular shift register with only one flip-flop being set at any particular time and all other flip-flops are cleared. The single bit is shifted from one flip-flop to the other to produce the sequence of timing signals. Basically the "ring" part comes from the fact that the output can be fed back into the input to form a ring.

4-BIT RING COUNTER:

A ring counter is also known as circular shift register. It is used to control the sequence of operation of digital system by using timing signals. In this counter, output of each stage is connected to the input of the successive stage. The output of the last flip-flop is connected as data input to the first flip-flop. The circuit diagram of a ring counter is shown in figure.



Operation:

In this circuit, clock pulse is applied to all the flip-flops simultaneously. Initially, at any given point of time, only one flip-flop is set and the remaining flip-flops are cleared i.e., 0 0 0 1. After the first clock pulse, 1 in LSB is shifted to second flip-flop i.e., FF2 and the counter output is 0 0 1 0. This counting continues till the counter counts 1 0 0 0 at 4th clock pulse. The counting sequence of the ring counter is shown in table.

Clock	Counting Sequence			
	Q_4	Q_3	Q_2	Q_1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	1	0	0
4	1	0	0	0

Table

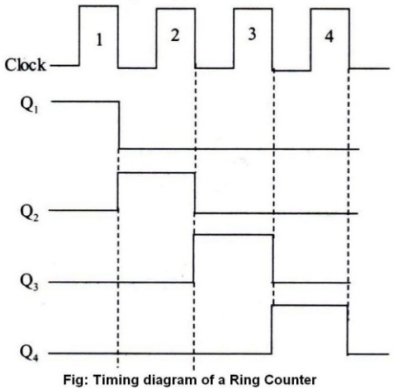


Fig: Timing diagram of a Ring Counter

Applications of Counter:

Some of the applications of the counters are:

1) Event counters
2) Frequency measurement
3) Digital Clock
4) Period measurement
5) Parallel to Serial data conversion (Multiplexing)

State Diagrams for D-Flip Flop, T-Flip Flop, S-R Flip Flop, J-K Flip Flop

Mealy and Moore Machines:

Mealy Machine is defined as a machine in the theory of computation whose output values are determined by both its current state and current inputs. In this machine at most one transition is possible.

It has 6 tuples: (Q, q0, ∑, ▲, δ, λ') (*** Tuple → a data structure consisting of multiple parts)

1. Q is a finite set of states
2. q0 is the initial state
3. ∑ is the input alphabet
4. ▲ is the output alphabet
5. δ is the transition function that maps Q×∑ → Q
6. 'λ' is the output function that maps Q×∑→ ▲

The diagram is as follows:

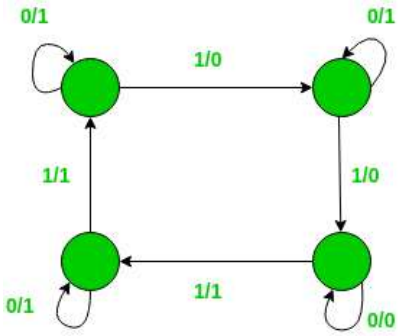


Figure - Mealy machine

Moore Machine:

Moore's machine is defined as a machine in the theory of computation whose output values are determined only by its current state. It has also 6 tuples

(Q, q0, ∑, ▲, δ, λ)

1. Q is a finite set of states
2. q0 is the initial state
3. ∑ is the input alphabet
4. ▲ is the output alphabet
5. δ is the transition function that maps Q×∑ → Q
6. λ is the output function that maps Q → ▲

Diagram:

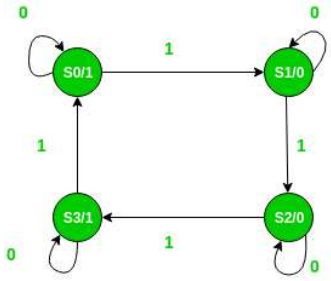


Figure - Moore machine

Differences between the Mealy machine and Moore machines:

Moore Machine	Mealy Machine
Output depends only upon the present state.	Output depends on the present state as well as present input.
Moore machine also places its output on the transition.	Mealy Machine places its output on the transition.
More states are required.	Less number of states are required.
There is less hardware requirement for circuit implementation.	There is more hardware requirement for circuit implementation.
They react slower to inputs(One clock cycle later).	They react faster to inputs.
Synchronous output and state generation.	Asynchronous output generation.
Output is placed on states.	Output is placed on transitions.
Easy to design.	It is difficult to design.

* * *