Unit VI: Registers and Counters

6.1 Shift Registers in Digital Logic

Flip flops can be used to store a single bit of binary data (1 or 0). However, in order to store multiple bits of data, we need multiple flip-flops. N flip flops are to be connected in order to store n bits of data. A **Register** is a device that is used to store such information. It is a group of flip-flops connected in series used to store multiple bits of data. The information stored within these registers can be transferred with the help of **shift registers**.

Shift Register is a group of flip flops used to store multiple bits of data. The bits stored in such registers can be made to move within the registers and in/out of the registers by applying clock pulses. An n-bit shift register can be formed by connecting n flip-flops where each flip-flop stores a single bit of data. The registers which will shift the bits to the left are called “Shift left registers”. The registers which will shift the bits to the right are called “Shift right registers”. Shift registers are basically of following types.

6.2 Types of Shift Registers/Modes of shift register

- Serial In Serial Out shift register (SISO)
- Serial In parallel Out shift register (SIPO)
- Parallel In Serial Out shift register (PISO)
- Parallel In parallel Out shift register (PIPO)

Serial-In Serial-Out Shift Register (SISO)

The shift register, which allows serial input (one bit after the other through a single data line) and produces a serial output is known as a Serial-In Serial-Out shift register. Since there is only one output, the data leaves the shift register one bit at a time in a serial pattern, thus the name Serial-In Serial-Out Shift Register. The logic circuit given below shows a serial-in serial-out shift register. The circuit consists of four D flip-flops which are connected in a serial manner. All these flip-flops are synchronous with each other since the same clock signal is applied to each flip-flop.

Compiled by: Er. Gopal Karna Email: info.gopalkarna@gmail.com

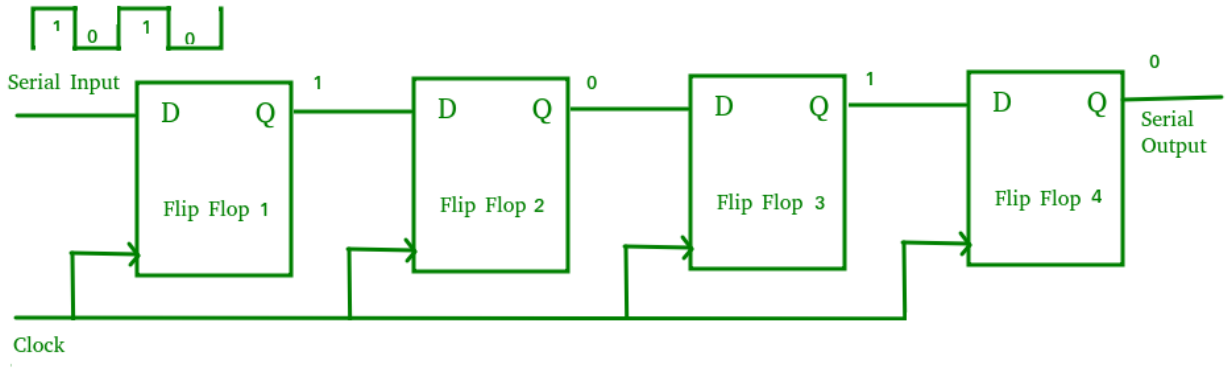


Fig: **Serial-In Serial-Out Shift Register (SISO)**

The above circuit is an example of a shift right register, taking the serial data input from the left side of the flip flop. The main use of a SISO is to act as a delay element.

Serial-In Parallel-Out Shift Register (SIPO)

The shift register, which allows serial input (one bit after the other through a single data line) and produces a parallel output is known as the Serial-In Parallel-Out shift register. The logic circuit given below shows a serial-in-parallel-out shift register. The circuit consists of four D flip-flops which are connected. The clear (CLR) signal is connected in addition to the clock signal to all 4 flip flops in order to RESET them. The output of the first flip-flop is connected to the input of the next flip flop and so on. All these flip-flops are synchronous with each other since the same clock signal is applied to each flip-flop.

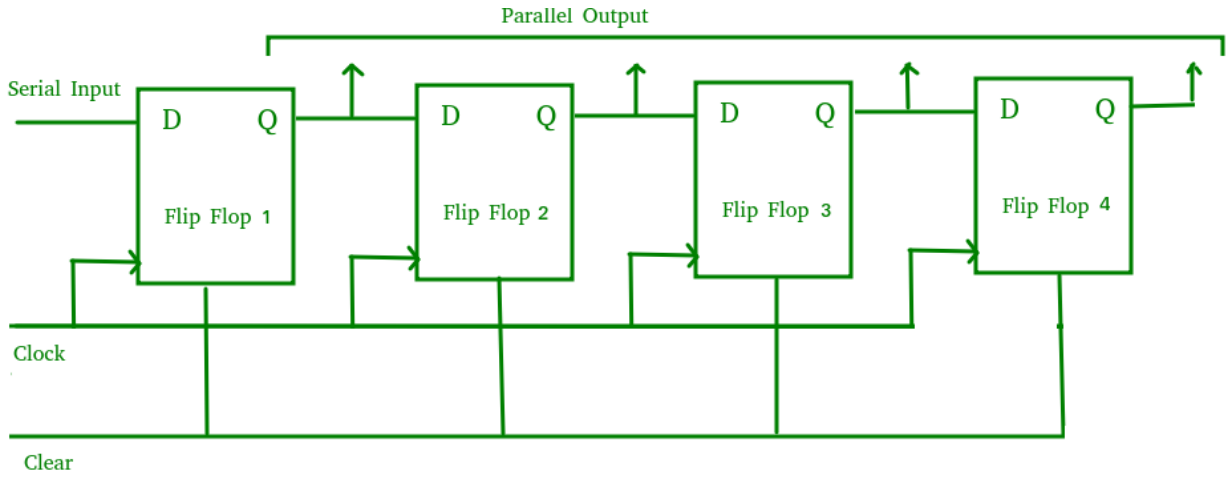


Fig: **Serial-In Parallel-Out Shift Register (SIPO)**

Compiled by: Er. Gopal Karna Email: info.gopalkarna@gmail.com

The above circuit is an example of a shift right register, taking the serial data input from the left side of the flip-flop and producing a parallel output. They are used in communication lines where demultiplexing of a data line into several parallel lines is required because the main use of the SIPO register is to convert serial data into parallel data.

Parallel-In Serial-Out Shift Register (PISO)

The shift register, which allows parallel input (data is given separately to each flip flop and in a simultaneous manner) and produces a serial output is known as a Parallel-In Serial-Out shift register. The logic circuit given below shows a parallel-in-serial-out shift register. The circuit consists of four D flip-flops which are connected. The clock input is directly connected to all the flip-flops but the input data is connected individually to each flip-flop through a multiplexer at the input of every flip-flop. The output of the previous flip-flop and parallel data input are connected to the input of the MUX and the output of MUX is connected to the next flip-flop. All these flip-flops are synchronous with each other since the same clock signal is applied to each flip-flop.

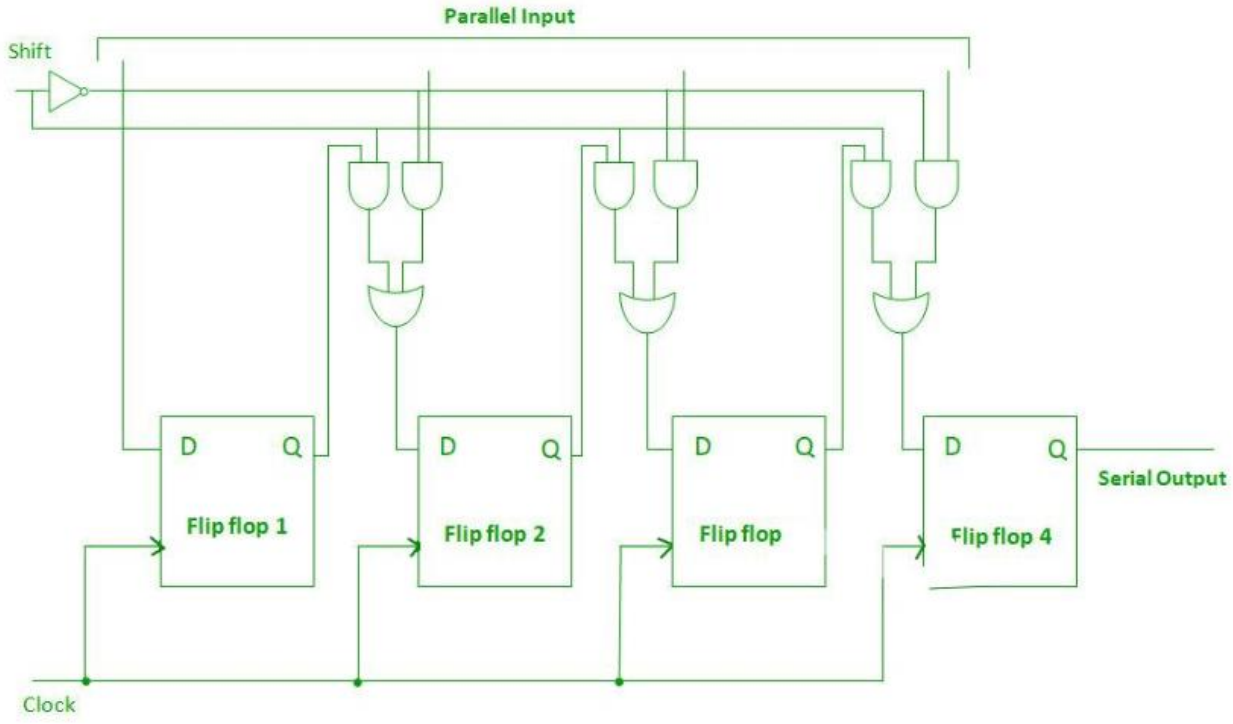


Fig: **Parallel-In Serial-Out Shift Register (PISO)**

A Parallel in Serial Out (PISO) shift register is used to convert parallel data to serial data.

Parallel-In Parallel-Out Shift Register (PIPO)

The shift register, which allows parallel input (data is given separately to each flip flop and in a simultaneous manner) and also produces a parallel output is known as Parallel-In Parallel-Out shift register. The logic circuit given below shows a parallel-in-parallel-out shift register. The circuit consists of four D flip-flops which are connected. The clear (CLR) signal and clock signals are connected to all 4 flip-flops. In this type of register, there are no interconnections between the individual flip-flops since no serial shifting of the data is required. Data is given as input separately for each flip flop and in the same way, output is also collected individually from each flip flop.

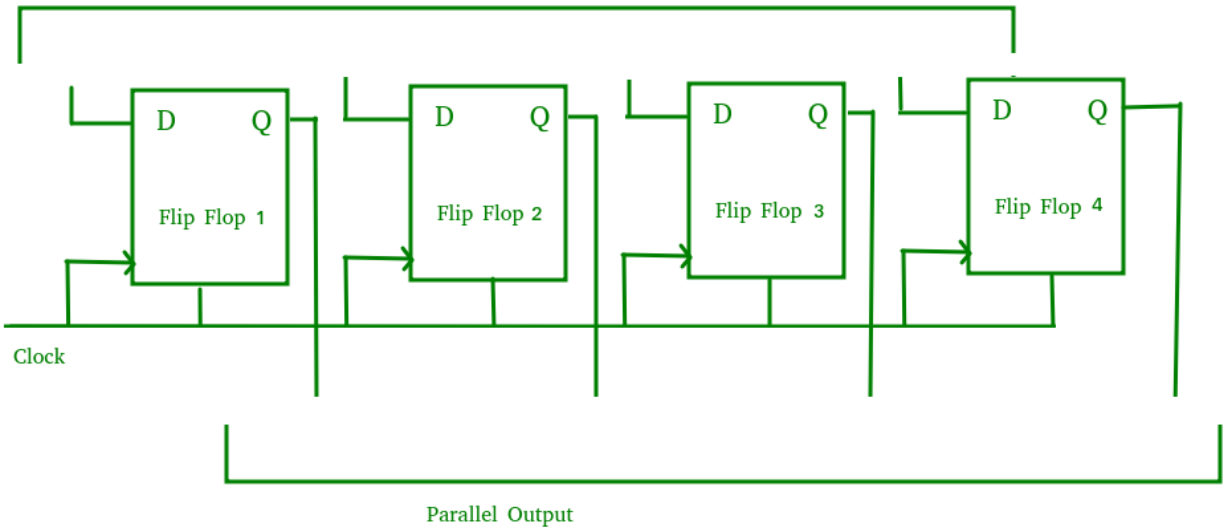


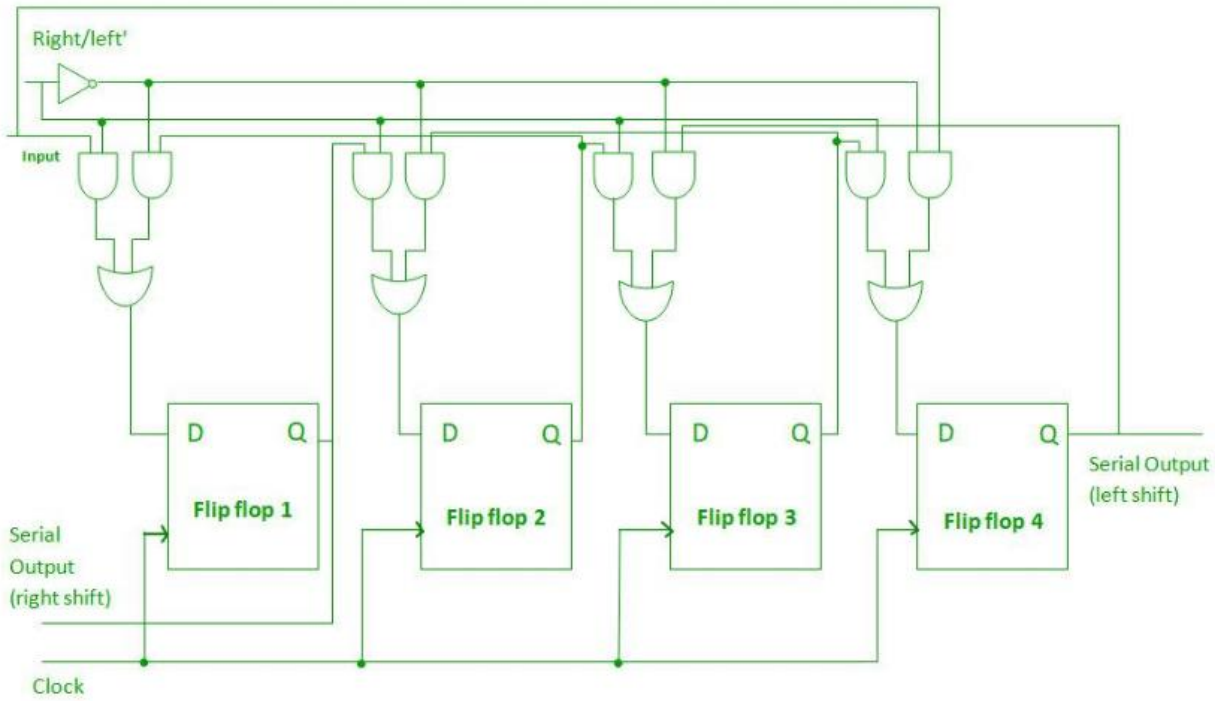
Fig: **Parallel-In Parallel-Out Shift Register (PISO)**

A Parallel in Parallel out (PIPO) shift register is used as a temporary storage device and like SISO Shift register it acts as a delay element.

Bidirectional Shift Register

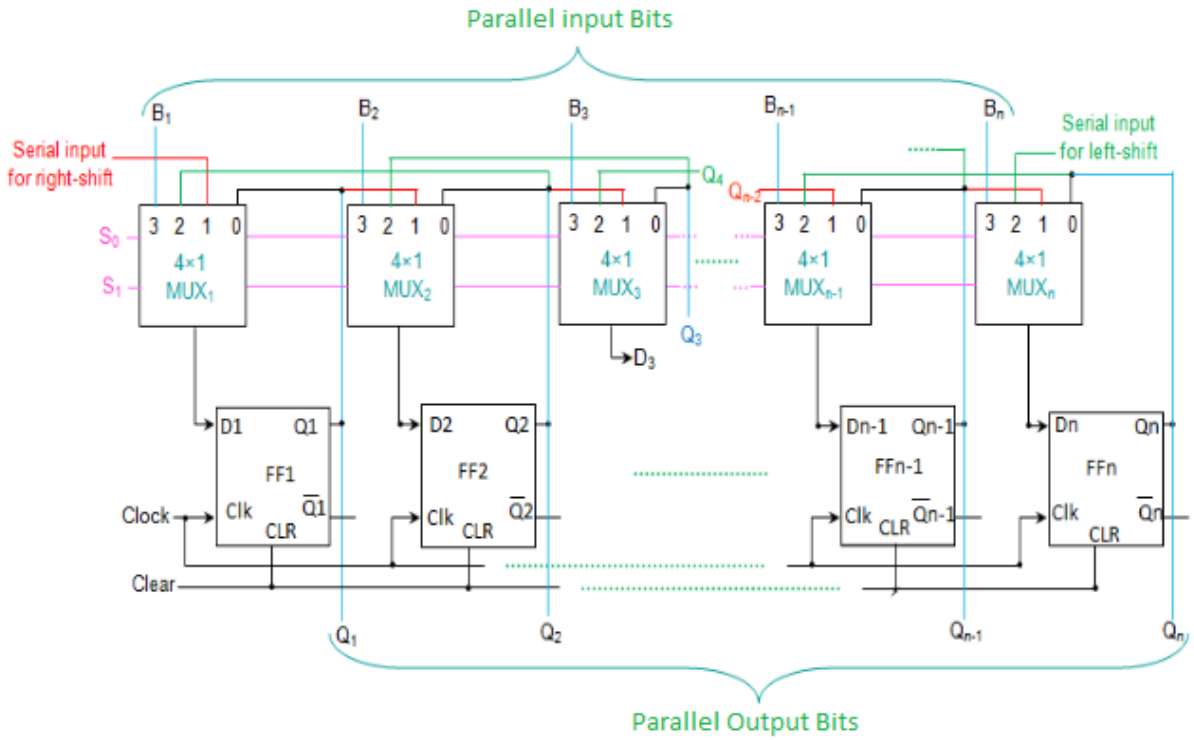
If we shift a binary number to the left by one position, it is equivalent to multiplying the number by 2 and if we shift a binary number to the right by one position, it is equivalent to dividing the number by 2. To perform these operations, we need a register which can shift the data in either direction. Bidirectional shift registers are the registers that are capable of shifting the data either right or left depending on the mode selected. If the mode selected is 1(high), the data will be shifted toward the right direction and if the mode selected is 0(low), the data will be shifted towards the left direction. The logic circuit given below shows a

Bidirectional shift register. The circuit consists of four D flip-flops which are connected. The input data is connected at two ends of the circuit and depending on the mode selected only one gate is in the active state.



Universal Shift Register

Universal Shift Register is a type of register that contains the both right shift and the left shift. It has also parallel load capabilities. Generally, these types of registers are taken as memory elements in computers. But, the problem with this type of register is that it shifts only in one direction. In simple words, you mean that the universal shift register is a combination of the **bidirectional shift register** and the **unidirectional shift register**.



N-bit universal shift register consists of flip-flops and multiplexers. Both are N in size. In this, all the n multiplexers share the same select lines and this select input selects the suitable input for flip-flops.

Shift Register Counter

Shift Register Counters are the shift registers in which the outputs are connected back to the inputs in order to produce particular sequences. There are basically two types:

- Ring Counter
- Johnson Counter

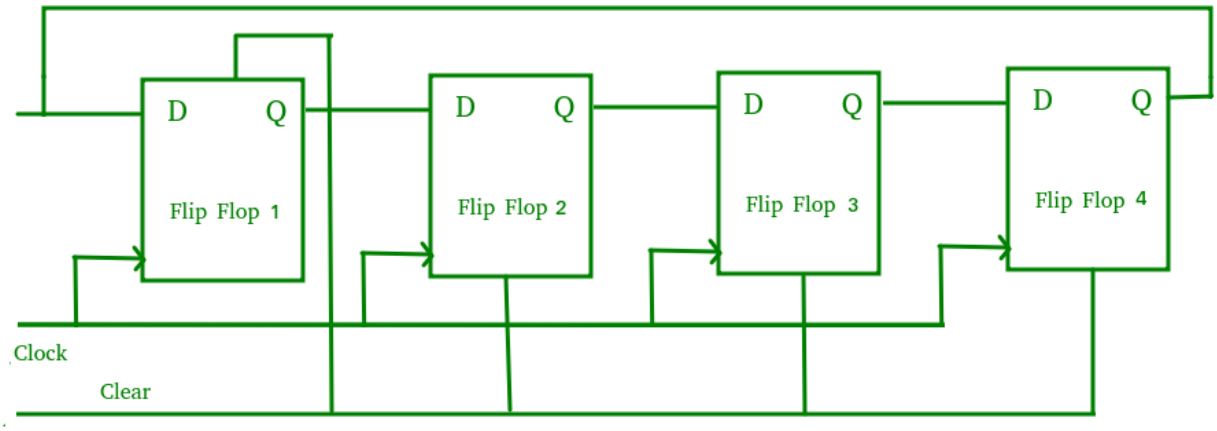
Ring Counter

A ring counter is basically a shift register counter in which the output of the first flip-flop is connected to the next flip-flop and so on and the output of the last flip-flop is again fed back to the input of the first flip-flop, thus the name ring counter. The data pattern within the shift register will circulate as long as clock pulses are applied. The logic circuit given below shows a Ring Counter.

Compiled by: Er. Gopal Karna Email: info.gopalkarna@gmail.com

Clock Pulse	Q1	Q2	Q3	Q4
0	1	0	0	1
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1

The circuit consists of four D flip-flops which are connected. Since the circuit consists of four flip-flops the data pattern will repeat after every four clock pulses as shown in the truth table. A Ring counter is generally used because it is self-decoding. No extra decoding circuit is needed to determine what state the counter is in.

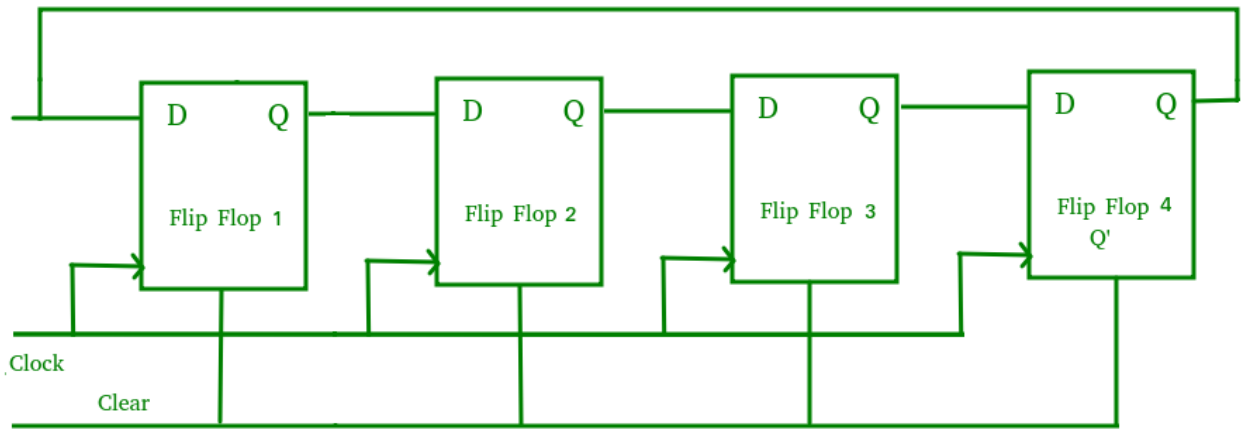


Johnson Counter

A Johnson counter is basically a shift register counter in which the output of the first flip flop is connected to the next flip flop and so on and the inverted output of the last flip flop is again fed back to the input of the first flip flop. They are also known as twisted ring counters. The logic circuit given below shows a Johnson Counter. The circuit consists of four D flip-flops which are connected.

Clock Pulse	Q1	Q2	Q3	Q4
0	0	0	0	1
1	0	0	0	0
2	1	0	0	0
3	1	1	0	0
4	1	1	1	0
5	1	1	1	1
6	0	1	1	1
7	0	0	1	1

An n-stage Johnson counter yields a count sequence of 2n different states, thus also known as a mod-2n counter. Since the circuit consists of four flip-flops the data pattern will repeat every eight clock pulses as shown in the truth table. The main advantage of the Johnson counter is that it only needs n number of flip-flops compared to the ring counter to circulate a given data to generate a sequence of 2n states.



Applications of Shift Registers

- The shift registers are used for temporary data storage.
- The shift registers are also used for data transfer and data manipulation.
- The serial-in serial-out and parallel-in parallel-out shift registers are used to produce time delay to digital circuits.
- The serial-in parallel-out shift register is used to convert serial data into parallel data thus they are used in communication lines where demultiplexing of a data line into several parallel lines is required.
- A Parallel in Serial out shift register is used to convert parallel data to serial data.

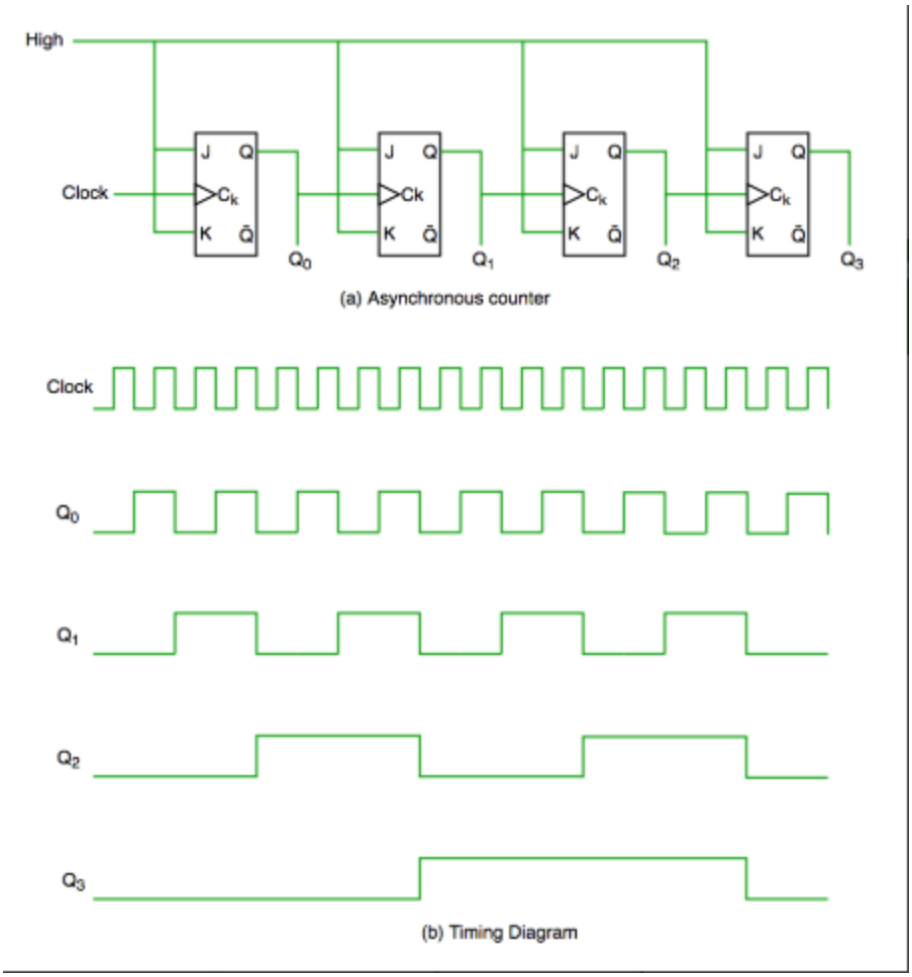
Counter Classification

Counters are broadly divided into two categories

1. Asynchronous counter
2. Synchronous counter

1. **Asynchronous Counter**

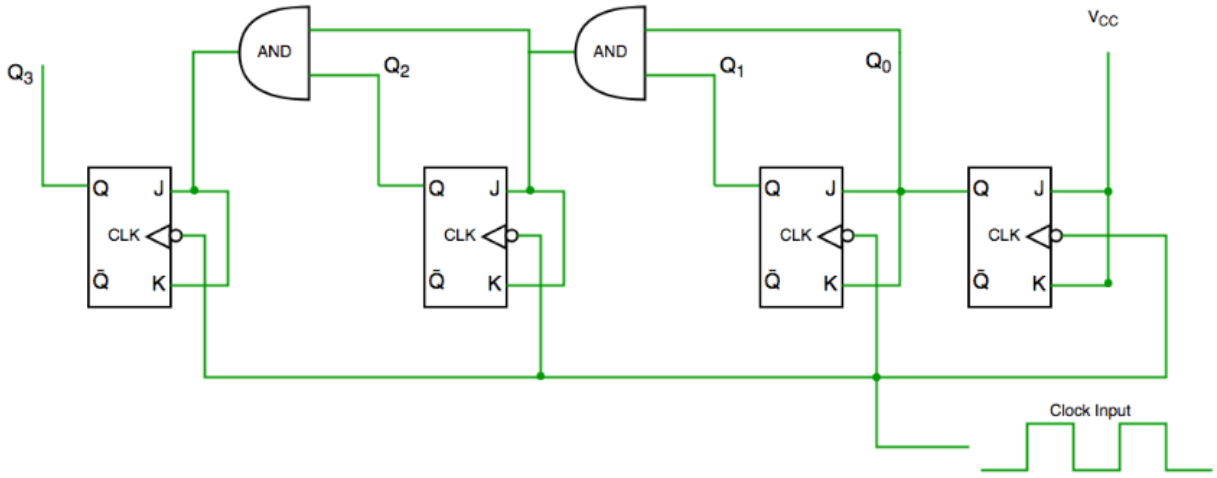
In asynchronous counter we don't use universal clock, only first flip flop is driven by main clock and the clock input of rest of the following flip flop is driven by output of previous flip flops. We can understand it by following diagram-

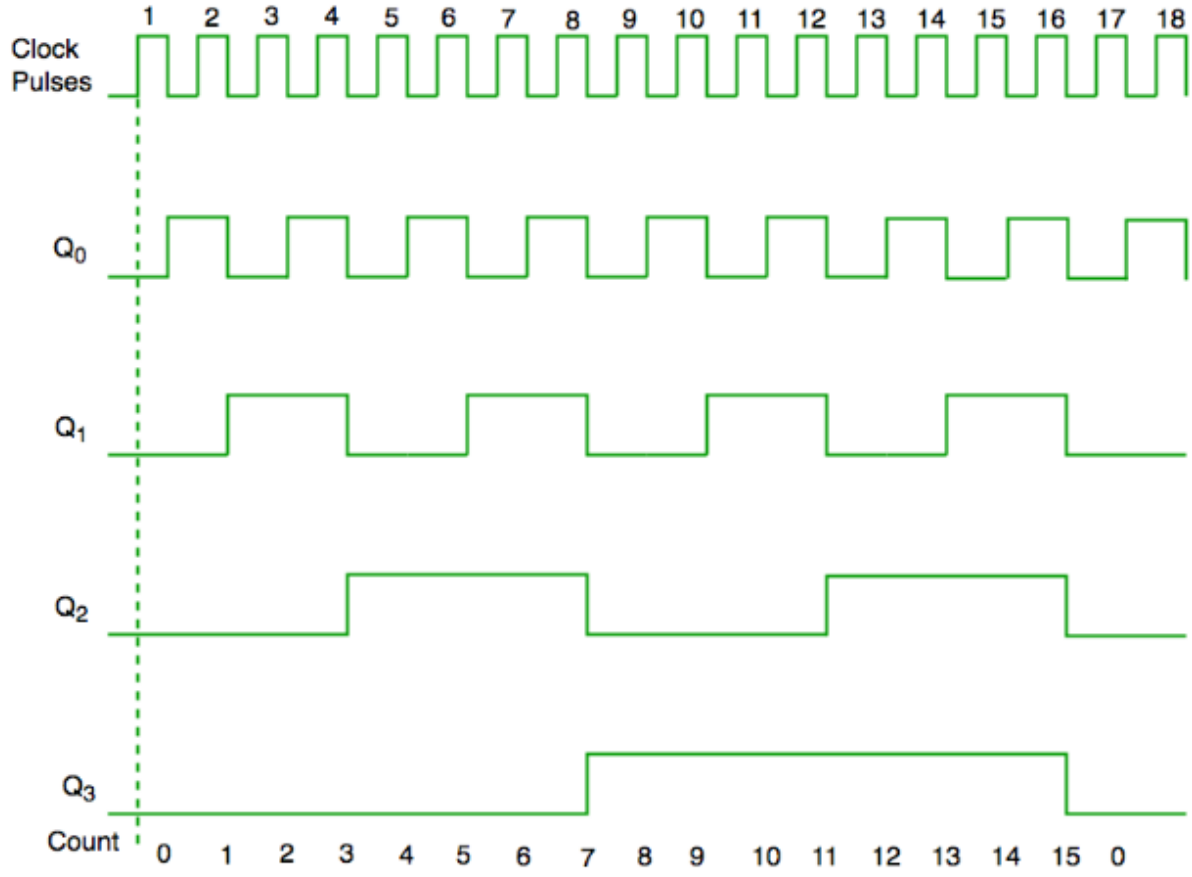


It is evident from timing diagram that Q0 is changing as soon as the rising edge of clock pulse is encountered, Q1 is changing when rising edge of Q0 is encountered (because Q0 is like clock pulse for second flip flop) and so on. In this way ripples are generated through Q0, Q1, Q2, Q3 hence it is also called **RIPPLE counter and serial counter.** A ripple counter is a cascaded arrangement of flip flops where the output of one flip flop drives the clock input of the following flip flop

2. **Synchronous Counter**

Unlike the asynchronous counter, synchronous counter has one global clock which drives each flip flop so output changes in parallel. The one advantage of synchronous counter over asynchronous counter is, it can operate on higher frequency than asynchronous counter as it does not have cumulative delay because of same clock is given to each flip flop. It is also called as parallel counter.





Timing diagram synchronous counter

From circuit diagram we see that Q0 bit gives response to each falling edge of clock while Q1 is dependent on Q0, Q2 is dependent on Q1 and Q0, Q3 is dependent on Q2, Q1 and Q0.

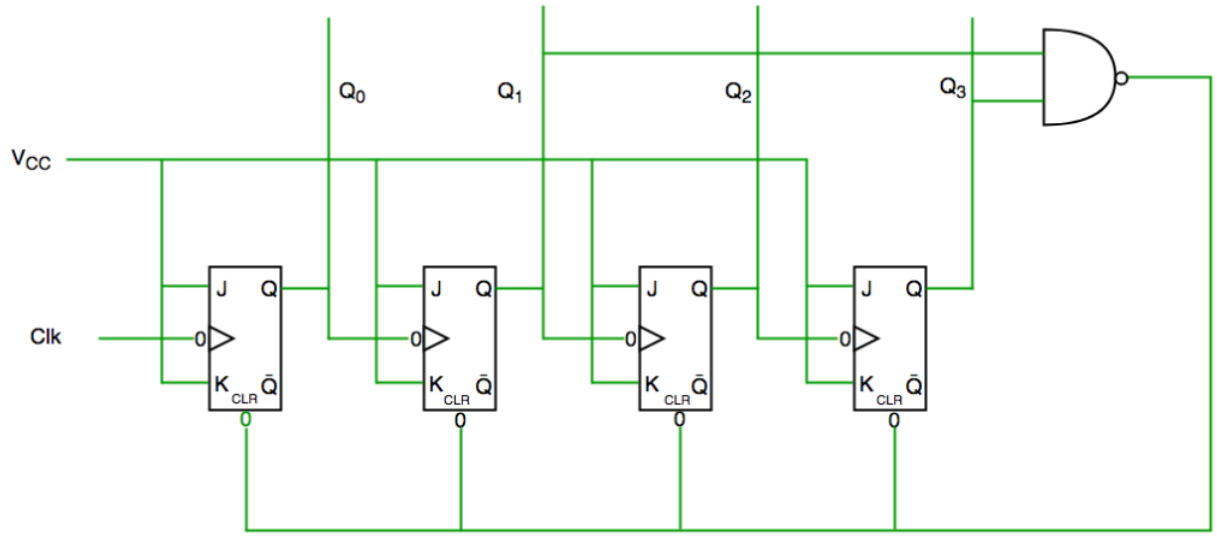
Decade Counter

A decade counter counts ten different states and then reset to its initial states. A simple decade counter will count from 0 to 9 but we can also make the decade counters which can go through any ten states between 0 to 15(for 4-bit counter).

Clock pulse	Q3	Q2	Q1	Q0
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

Truth table for simple decade counter



Decade counter circuit diagram

We see from circuit diagram that we have used NAND gate for Q3 and Q1 and feeding this to clear input line because binary representation of 10 is

1010

And we see Q3 and Q1 are 1 here, if we give NAND of these two bits to clear input then counter will be clear at 10 and again start from beginning.

Important point: Number of flip flops used in counter are always greater than equal to ($\log_2 n$) where n=number of states in counter.

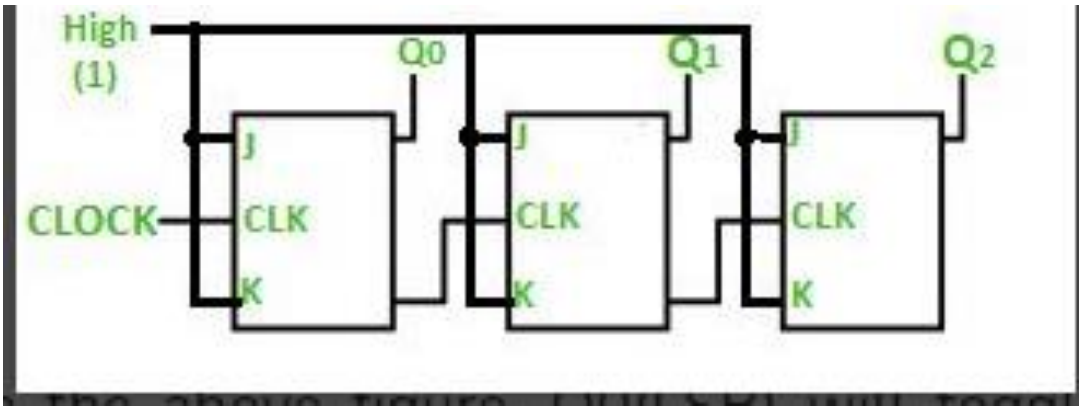
6.3 Ripple Counter

Ripple counter is a cascaded arrangement of flip-flops where the output of one flip-flop drives the clock input of the following flip-flop. The number of flip flops in the cascaded arrangement depends upon the number of different logic states that it goes through before it repeats the sequence a parameter known as the modulus of the counter. A n-bit ripple counter can count up to 2n states. It is also known as MOD n counter. It is known as ripple counter because of the way the clock pulse ripples its way through the flip-flops. Some of the features of ripple counter are:

- It is an asynchronous counter.
- Different flip-flops are used with a different clock pulse.
- All the flip-flops are used in toggle mode.
- Only one flip-flop is applied with an external clock pulse and another flip-flop clock is obtained from the output of the previous flip-flop.
- The flip-flop applied with an external clock pulse act as LSB (Least Significant Bit) in the counting sequence.

A counter may be an up counter that counts upwards or can be a down counter that counts downwards or can do both i.e. Count up as well as count downwards depending on the input control. The sequence of counting usually gets repeated after a limit. When counting up, for the n-bit counter the count sequence goes from 000, 001, 010, ... 110, 111, 000, 001, ... etc. When counting down the count sequence goes in the opposite manner: 111, 110, ... 010, 001, 000, 111, 110, ... etc.

A 3-bit Ripple counter using a JK flip-flop is as follows:



In the circuit shown in the above figure, Q0(LSB) will toggle for every clock pulse because JK flip-flop works in toggle mode when both J and K are applied 1, 1, or high input. The following counter will toggle when the previous one changes from 1 to 0.

Truth Table is as follows:

Counter State	Q_2	Q_1	Q_0
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

The 3-bit ripple counter used in the circuit above has eight different states, each one of which represents a count value. Similarly, a counter having n flip-flops can have a maximum of 2 to the power n states. The number of states that a counter owns is known as its mod (modulo) number. Hence a 3-bit counter is a mod-8 counter. A mod-n counter may also be described as a divide-by-n counter. This is because the most significant flip-flop (the furthest flip-flop from the original clock pulse) produces one pulse for every n pulse at the clock input of the least significant flip-flop (the one triggers by the clock pulse). Thus, the above counter is an example of a divide-by-4 counter.

Timing diagram

Let us assume that the clock is negative edge triggered so the above the counter will act as an up counter because the clock is negative edge triggered and output is taken from Q.

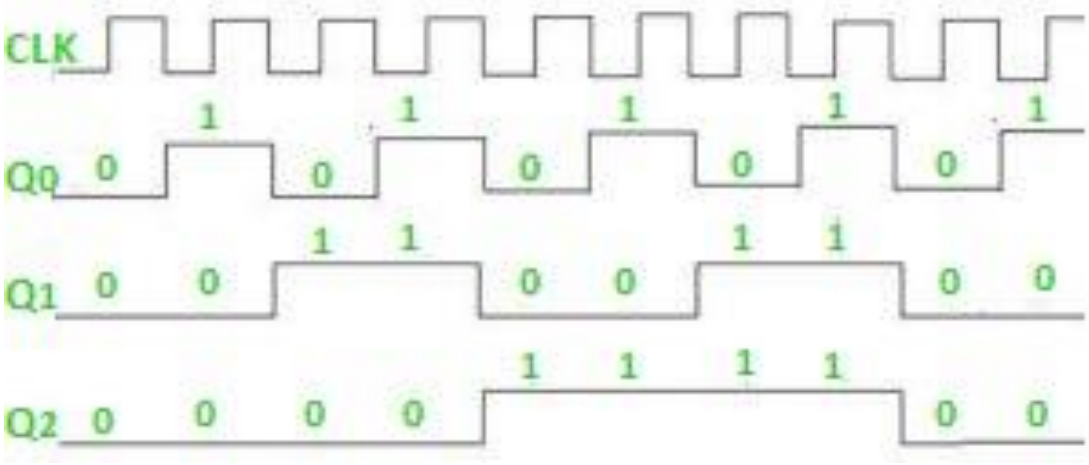


Fig: **Timing diagram**

Counters are used very frequently to divide clock frequencies and their uses mainly involve digital clocks and in multiplexing. The widely known example of the counter is parallel to serial data conversion logic.

Advantages of Ripple Counter in Digital Logic

- Can be easily designed by T flip-flop or D flip-flop.
- Can be used in low speed circuits & divide by n-counters.
- Used as Truncated counters to design any mode number counters (i.e. Mod 4, Mod 3)

Disadvantages of Ripple Counter in Digital Logic

- Extra flip-flop is needed to do resynchronization.
- To count the sequence of truncated counters, additional feedback logic is needed.
- Propagation delay of asynchronous counters is very large, while counting the large number of bits.
- Counting errors may occur due to propagation delay for high clock frequencies.

6.4 Synchronous 3 bit Up/Down counter

Overview:

- These types of counters fall under the category of synchronous controller counter.
- Here the mode control input is used to decide whether which sequence will be generated by the counter.
- In this case, mode control input is used to decide whether the counter will perform up counting or down counting.
- Designing of such a counter is the same as designing a synchronous counter but the extra combinational logic for mode control input is required.

Steps to design Synchronous 3 bit Up/Down Counter:

1. Decide the number and type of FF -

- Here we are performing 3 bit or mod-8 Up **or Down counting,** so 3 Flip Flops are required, which can count up to 2^3-1 = 7.
- Here T Flip Flop is used.

2. **Write excitation table of Flip Flop -**

Previous state(Q_n)	Next state(Q_{n+1})	T
0	0	0
0	1	1
1	0	1
1	1	0

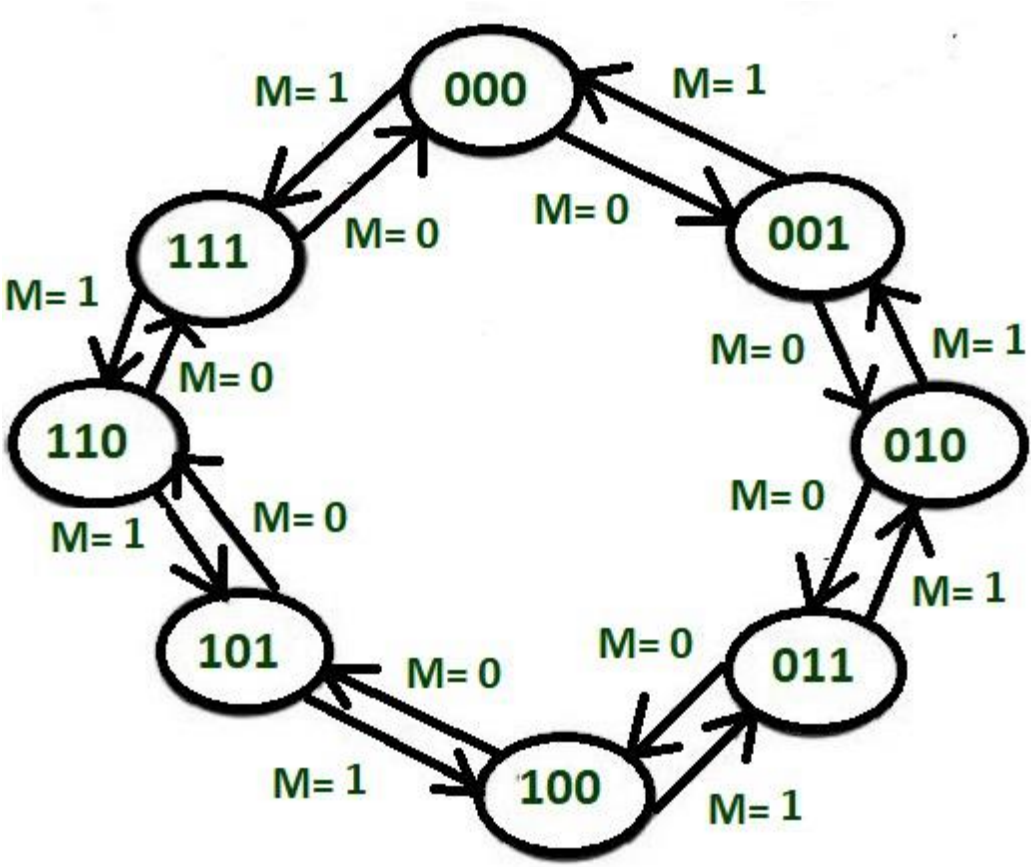
Excitation table of T FF

3. Decision for Mode control input M -

- When M=0, then the counter will perform up counting.
- When M=1, then the counter will perform down counting.

Compiled by: Er. Gopal Karna Email: info.gopalkarna@gmail.com

4. Draw the state transition diagram and circuit excitation table -



State transition diagram for 3 bit up/down counting.

5. Circuit excitation table -

The circuit excitation table represents the present states of the counting sequence and the next states after the clock pulse is applied and input T of the flip-flops. By seeing the transition between the present state and the next state, we can find the input values of 3 Flip Flops using the Flip Flops excitation table. The table is designed according to the required counting sequence.

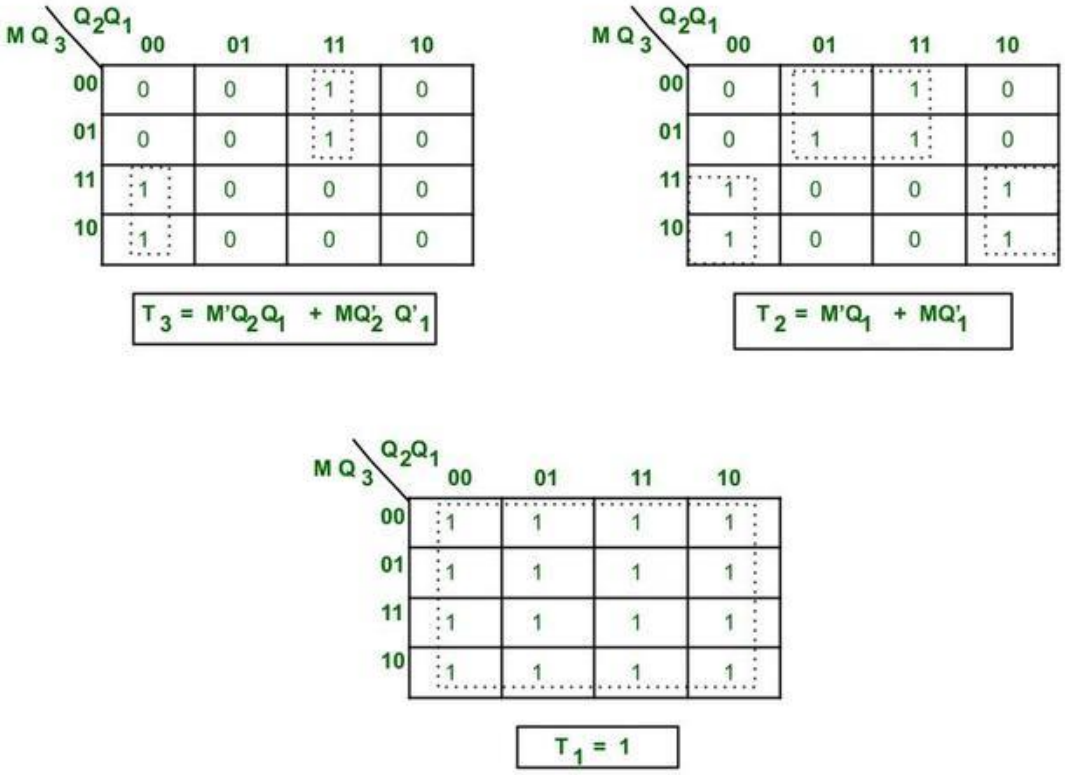
M	Q_3	Q_2	Q_1	Q_3^*	Q_2^*	Q_1^*	T_3	T_2	T_1
0	0	0	0	0	0	1	0	0	1
0	0	0	1	0	1	0	0	1	1
0	0	1	0	0	1	1	0	0	1
0	0	1	1	1	0	0	1	1	1
0	1	0	0	1	0	1	0	0	1
0	1	0	1	1	1	0	0	1	1
0	1	1	0	1	1	1	0	0	1
0	1	1	1	0	0	0	1	1	1
1	0	0	0	1	1	1	1	1	1
1	0	0	1	0	0	0	0	0	1
1	0	1	0	0	0	1	0	1	1
1	0	1	1	0	1	0	0	0	1
1	1	0	0	0	1	1	1	1	1
1	1	0	1	1	0	0	0	0	1
1	1	1	0	1	0	1	0	1	1
1	1	1	1	0	1	0	0	0	1

Circuit excitation table

If there is a change in the output state of a flip flop (i.e. 0 to 1 or 1 to 0), then the corresponding T value becomes 1 otherwise 0.

6. Find a simplified equation using k map -

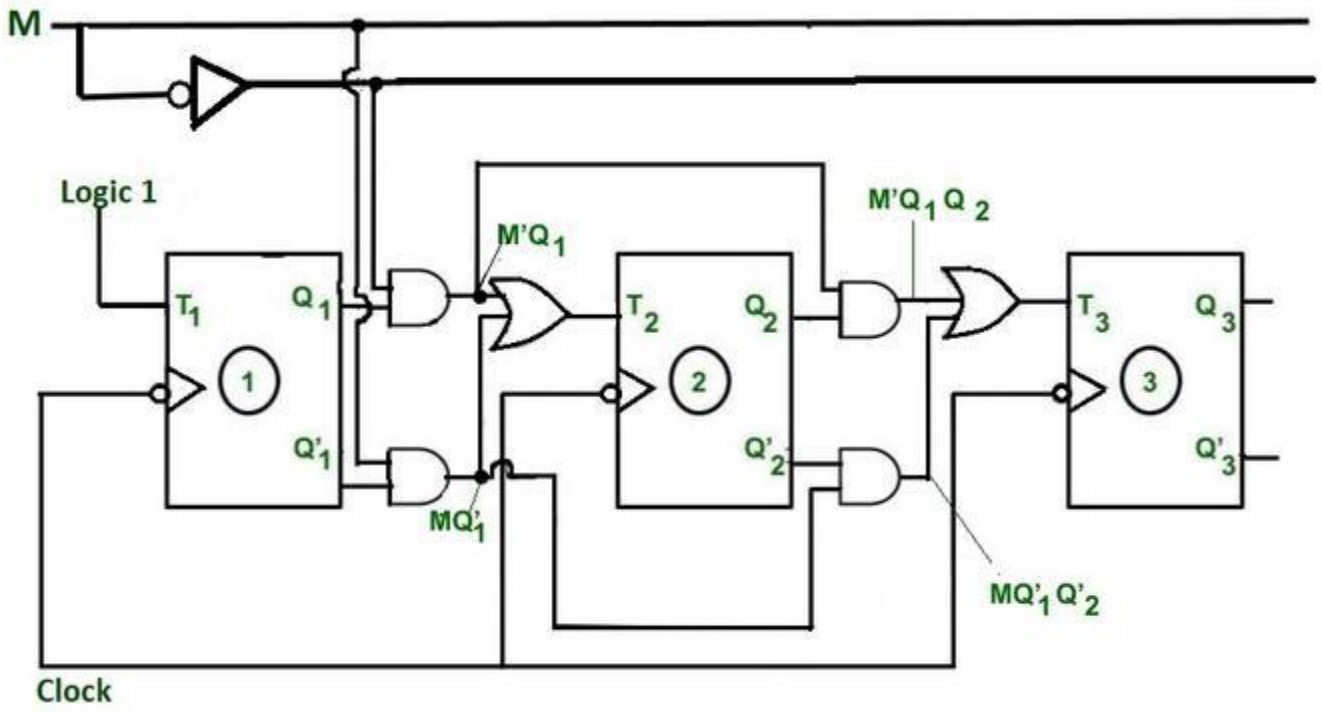
Here we are finding the minimal Boolean expression for each Flip Flop input T using k map.



Simplified equation for K map

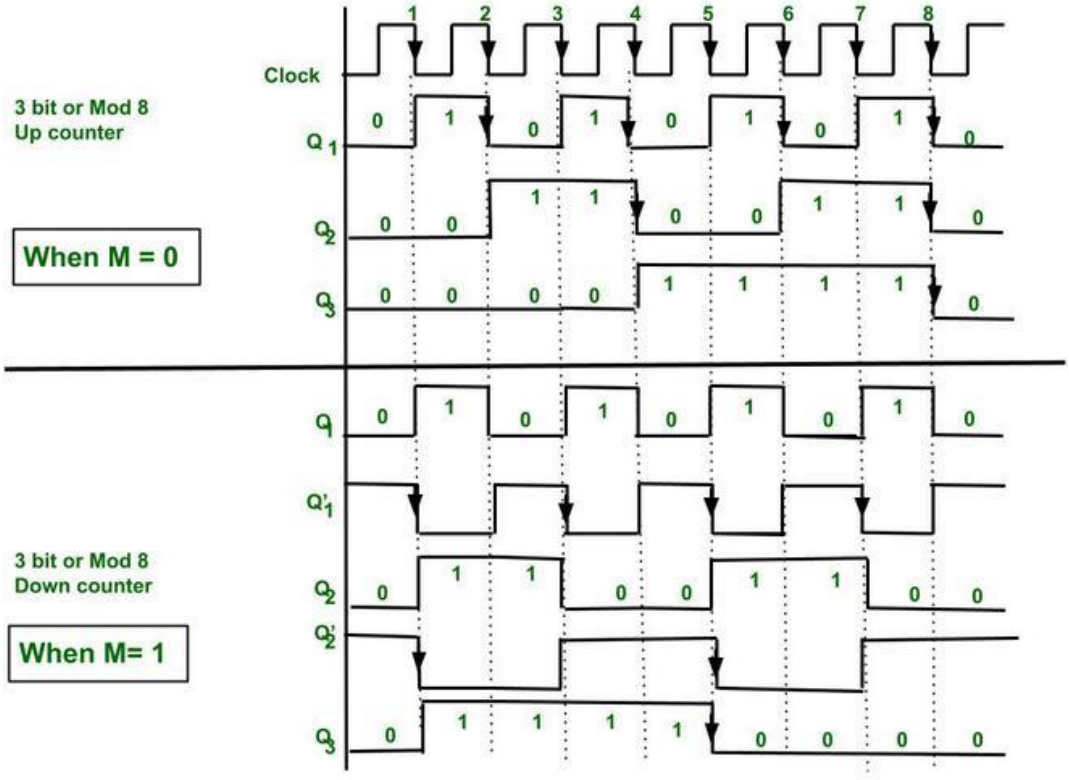
7. Create a circuit diagram -

The simplified expression for Flip Flops is used to design circuit diagrams. Here all the connections are made according to simplified expressions for Flip Flops.



3 bit synchronous up/down counter.

8. Timing Diagram -



Timing diagram for 3 bit synchronous Up/Down counter

Explanation:

Here -ve edge triggered clock pulse is used for toggling purpose.

Previous state(Q_n)	T	Next state(Q_{n+1})
0	0	0
0	1	1
1	0	1
1	1	0

Characteristics table of T FF

After every falling edge, when T = 1, the output state of Flip Flop will toggle.

- Initially $Q_3 = 0$, $Q_2 = 0$, $Q_1 = 0$.

Case 1: When M=0, then M'= 1

- $T_3 = M'Q_2Q_1 + MQ'_2Q'_1 = Q_2Q_1$.
- $T_2 = M'Q_1 + MQ'_1 = 1.\ Q_1 = Q_1$.
- $T_1 = 1$.
- Because $T_1 = 1$, therefore FF1 output state toggles for every falling edge. The output state of FF 2 will toggle when $Q_1 = 1$ and the falling edge of the clock pulse occurs. The output state of FF 3 will toggle only when $Q_2.\ Q_1 = 1$ and the falling edge of the clock pulse occurs.
- In this way, after every falling edge, state transition takes place and we can get our desired counting sequence.

Case 2: When M=1, then M' =0

- $T_3 = M'Q_2Q_1 + MQ'_2Q'_1 = Q'_2Q'_1$
- $T_2 = M'Q_1 + MQ'_1 = 1.\ Q_1 = Q'_1$.
- $T_1 = 1$.

- Because T_1= 1, therefore FF1 output state toggles for every falling edge. The output state of FF 2 will toggle when Q'_1 = 1 and the falling edge of the clock pulse occurs. The output state of FF 3 will toggle only when Q'_2. Q'_1= 1 and the falling edge of the clock pulse occurs.
- In this way, after every falling edge, state transition takes place and we can get our desired counting sequence.

THE END

Compiled by: Er. Gopal Karna Email: info.gopalkarna@gmail.com