Digital Electronics

Er. L N Soni

AKS University, Satna

UNIT-4
Counters

S. N.	Combinational Circuit	Sequential Circuit
1	Output depends only upon present input.	O/p depends upon present as well as past i/p.
2	Speed is fast.	Speed is slow.
3	It is designed easy.	It is designed tough as compared to combinational circuits.
4	There is no feedback between input and output.	There exists a feedback path b/w i/p and o/p.
5	This is time independent.	This is time dependent.
6	Elementary building blocks: Logic gates	Elementary building blocks: Flip-flops
7	Used for arithmetic as well as Boolean operations.	Mainly used for storing data.
8	Combinational circuits don’t have capability to store any state.	Sequential circuits have capability to store any state or to retain earlier state.
9	As combinational circuits don’t have clock, they don’t require triggering.	As sequential circuits are clock dependent they need triggering.
10	These circuits do not have any memory element.	These circuits have memory element.
11	It is easy to use and handle.	It is not easy to use and handle.

Counters

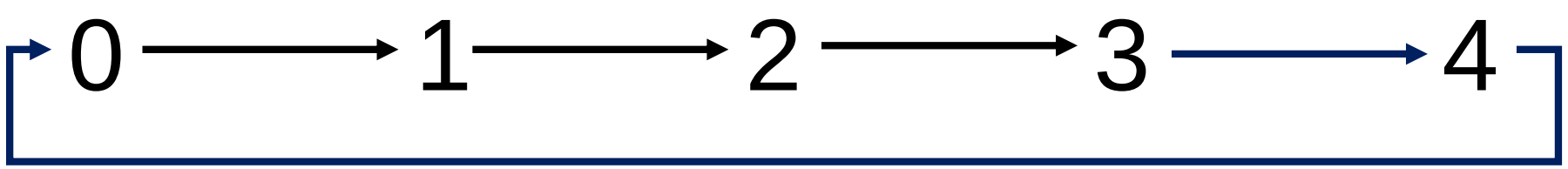
Counter is a sequential circuit. A digital circuit which is used for a counting pulses is known counter.

- Counter is the widest application of flip-flops.
- It is a group of flip-flops with a clock signal applied.
- Counters are used in digital electronics for counting purpose, they can count specific event happening in the circuit.
- Real Life Examples: Washing Machine, AC, Microwave, EVM (Electronic voting machine), Space Ship Launch, Digital Clock, etc.

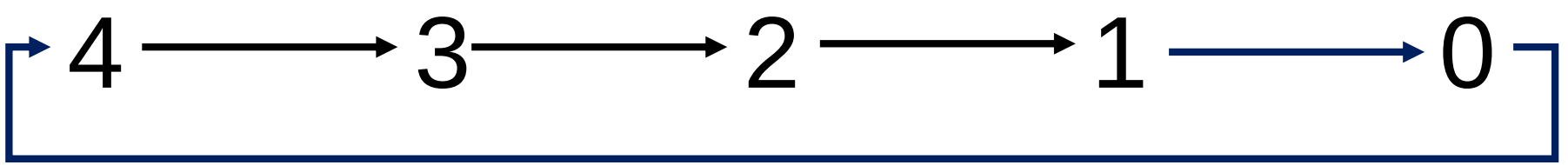
Counter works in two modes

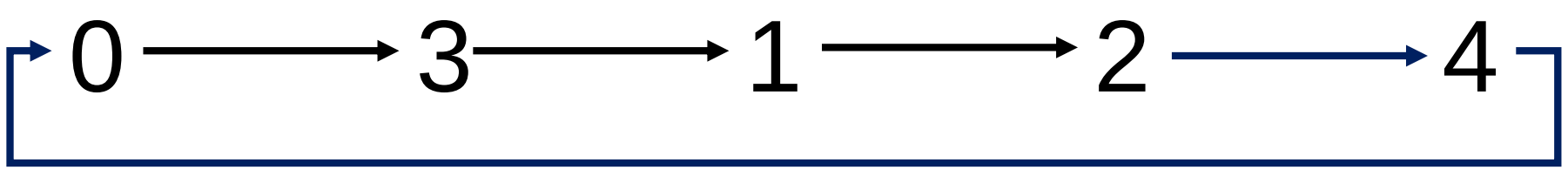
- Up counter
- Down counter

For example, in UP counter a counter increases count for every rising edge of clock (not using level triggering). Ex: Digital Clock, EVM



In Down counter, a counter decreases count for every rising edge of clock (not using level triggering). Ex: Washing Machine, Space Ship Launch

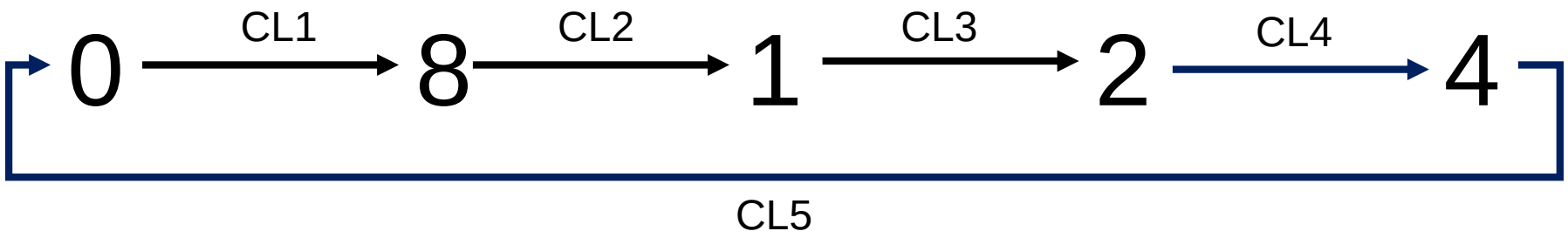




Counters are sequential circuit that count the number of pulses can be either in binary code or BCD form.

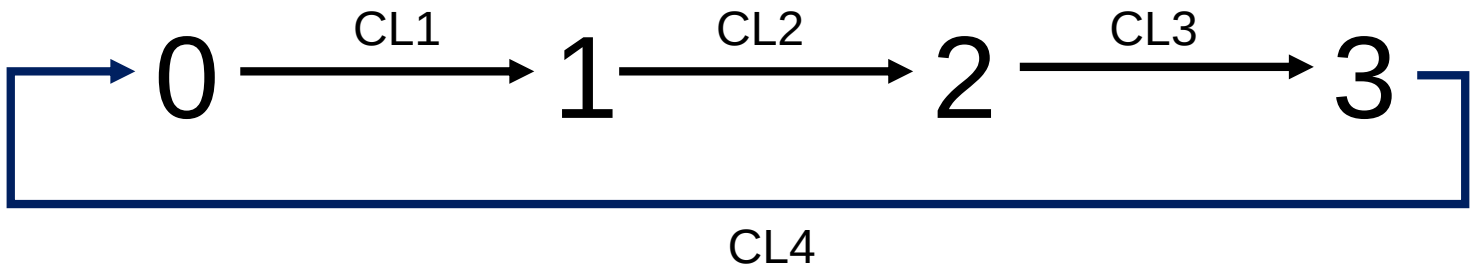
The main properties of a counter are timing, sequencing, and counting.

Find Mod Value of a counter



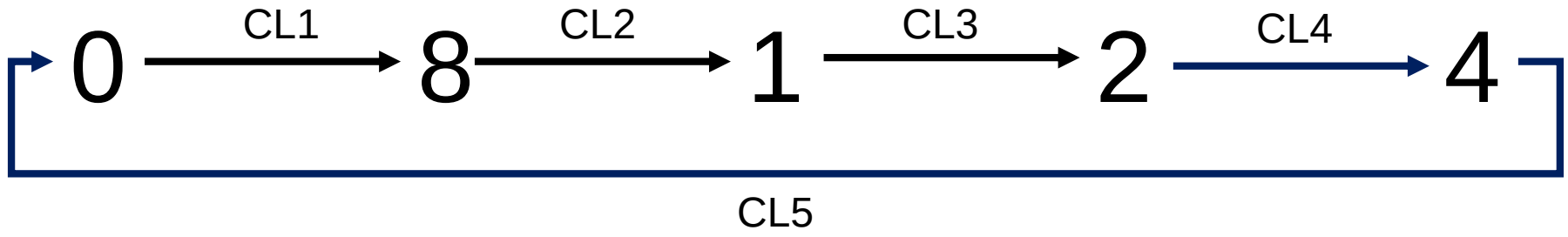
Mod 5 Counter

- Number of different state is called Mod of the counter.



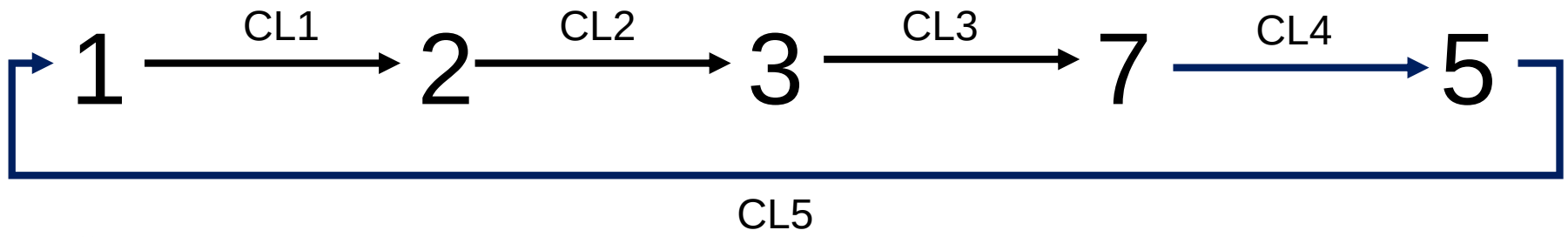
Mod 4 Counter

Que 1: After applying 32 Clock Pulse what is the state value in given counter?



Ans: “1”

Que 2: If the starting state is 3, then after 17 clock pulse what will be the value in the given counter?



Ans: “5”

How to find the no. of flip flop required to design Mod-n Counter

Ex 1: Find the no. of flip flop required to design Mod-5 Counter?

$$2^n \geq Mod \qquad 2^3 \geq 5$$

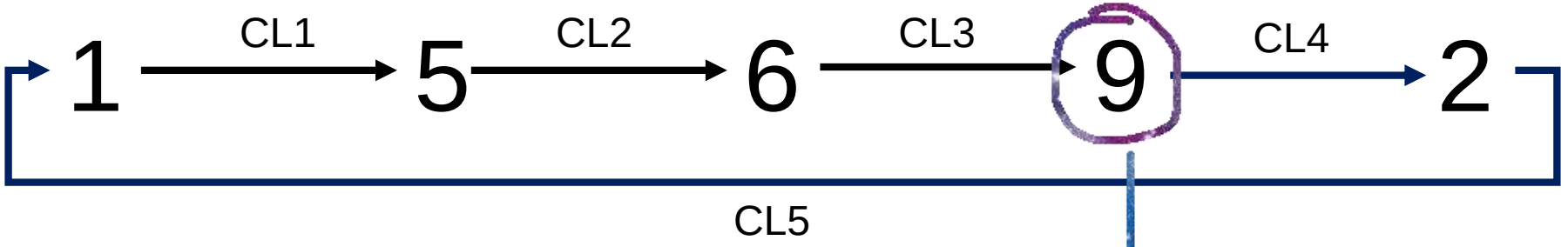
Ans: **3 flip flop** required to design Mod 5 Counter.

Ex 2: Find the no. of flip flop required to design Mod-265 Counter?

$$2^n \geq Mod \qquad 2^9 \geq 265$$

Ans: **9 flip flop** required to design Mod 5 Counter.

Ex 1: Find the no. of flip flop required to design a given Counter?



- Check the largest value in given sequence.

$$2^n \geq Mod$$

$$2^3 \geq 5$$

Ans: “3” That is wrong.

Ans: “4” flip flop.

Types of Counters

The Counters in digital electronics are broadly classified into two main categories.

- Asynchronous Counters
- Synchronous Counters

Synchronous Counters

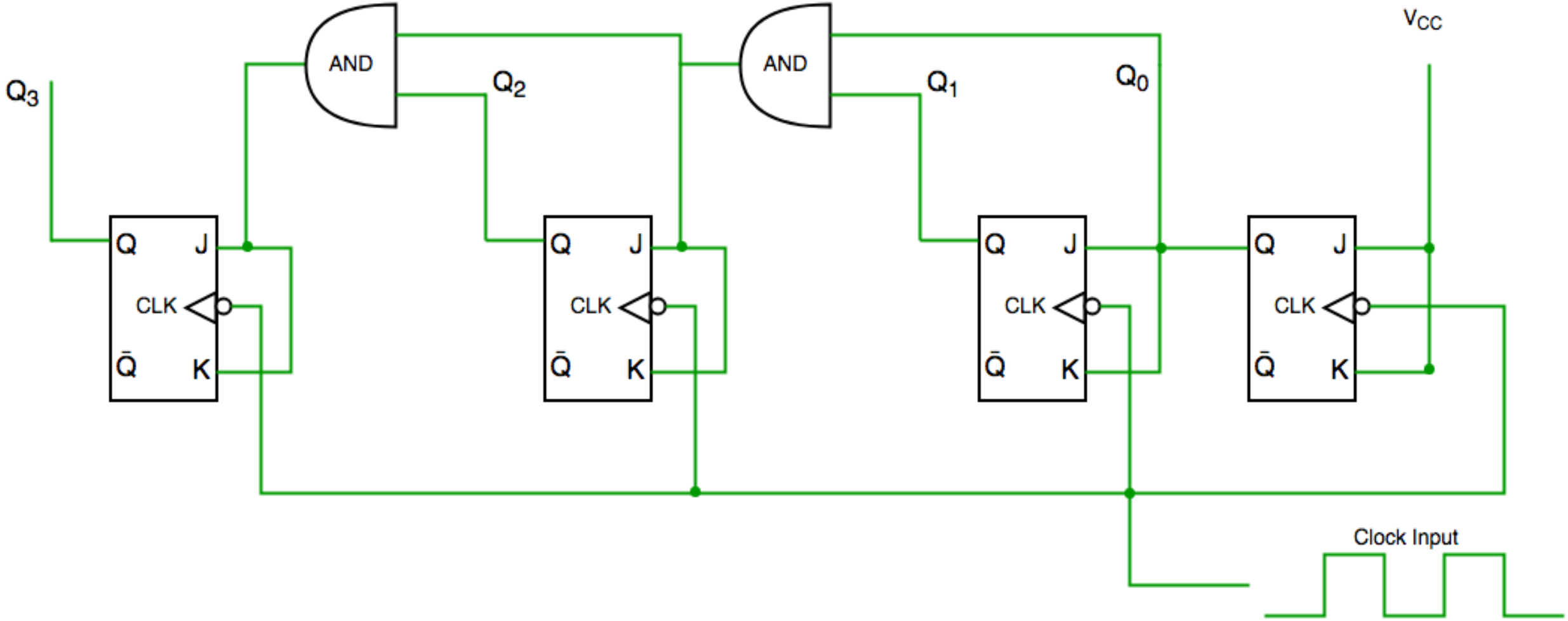
If the "clock" pulses are applied to all the flip-flops in a counter simultaneously, then such a counter is called as synchronous counter.

- In a synchronous counter, all the constituting flip-flops are clocked with the same clock input simultaneously. These are also known as parallel counters.

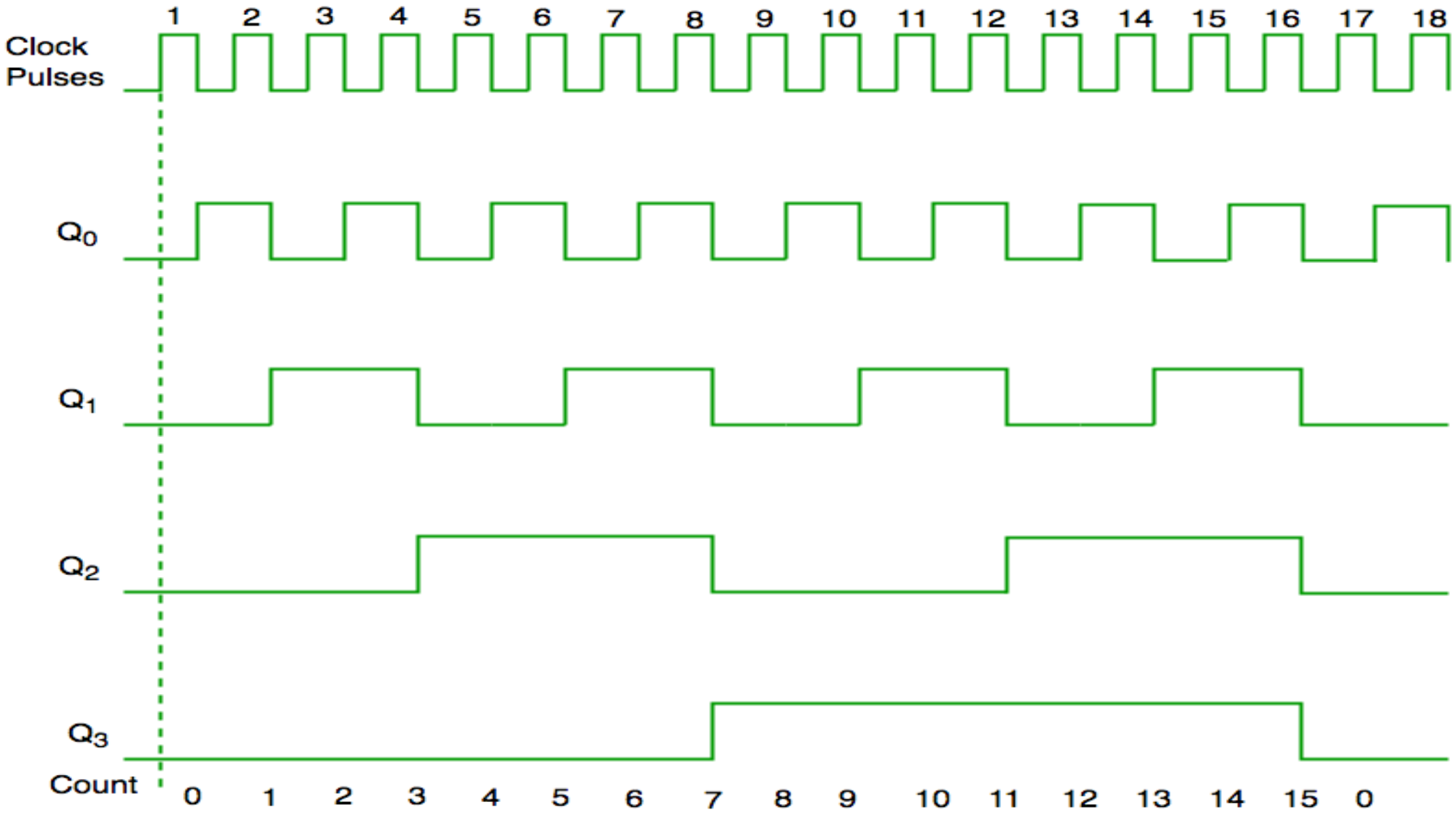
- Basically, all the flip-flops in a synchronous counter are arranged in a cascade connection and each flip-flop is individually connected to an external clock. It allows the clocking of all the flip-flops at the same time instant with the same clock input. It means the output of each flip-flop varies in synchronization with the clock input.

- Due to this, the common clock signal causes the change in the state of each individual flip-flop simultaneously. Resultantly it leads to no ripple effect, thus there is no propagation delay in a synchronous counter.

- Logic gates are used in synchronous counters to control the count sequence.



Synchronous counter



Timing diagram synchronous counter

From circuit diagram we see that Q0 bit gives response to each falling edge of clock while

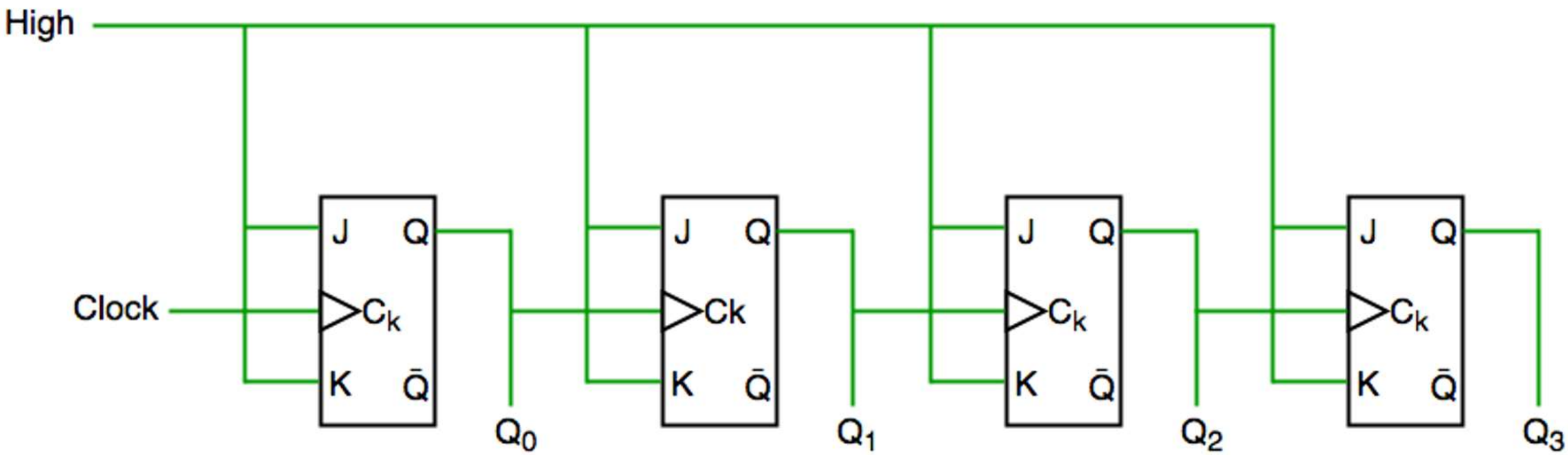
- Q1 is dependent on Q0,
- Q2 is dependent on Q1 and Q0,
- Q3 is dependent on Q2,Q1 and Q0.

Asynchronous Counters

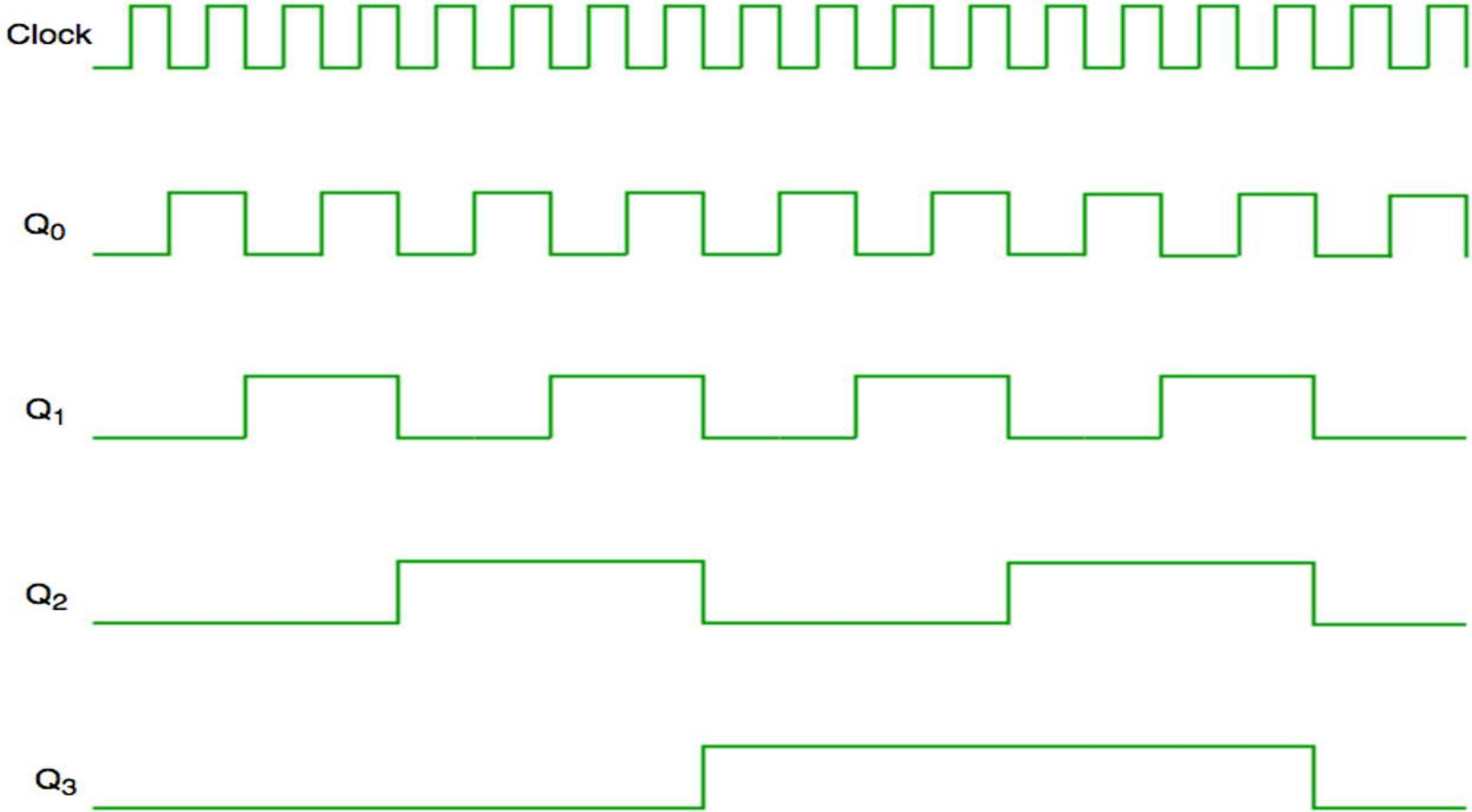
These are the counters in which we do not use universal clock, main clock is only applied to the first flip flop and then for rest of flip flops the output of previous flip flop is taken as a clock.

- The output of the first flip-flop acts as the input of the next adjacent flip-flop in the forward direction. In this manner, the clock input ripples through the counter. Hence, these counters are also known as ripple counters.

- Due to the ripple effect, the timing signal in an asynchronous counter gets delayed by some amount on passing through each flip flop. Hence, it results in a propagation delay.



(a) Asynchronous counter



(b) Timing 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.

Application of Counters

Here are the applications of the counters in digital electronics:

- **Frequency Measurement and Division**
  The counter is used to measure the frequency of a signal, simply by counting the no of cycles in a particular given time period and the counter is also used to divide the input clock frequency by a fixed integer value.
- **Timing**
  The counter is also used to generate timing signals like pulse-width modulated (PWM) signals. These signals are commonly used in power electronics to control the speed of motors and regulate the brightness of LEDs.
- **Binary Arithmetic**
  Binary arithmetic operations like addition, subtraction, multiplication, and division are used in digital systems by counters.
- **Data Storage**
  Counters can also be used as memory elements in a digital circuit if take an example of a binary operator. The binary counter can be used to store a binary value that represents a state in a digital system.
- **Digital Signal Processing**
  Counters are also used in digital signal processing applications like filtering and signal analysis.

S.NO	Synchronous Counter	Asynchronous Counter
1.	In synchronous counter we use a universal clock that is common to all flip flops through out the circuit.	In asynchronous counter main clock is only applied to the first flip flop and then for rest of flip flops the output of previous flip flop is taken as a clock.
2.	Synchronous Counter is faster in operation as compared to Asynchronous Counter.	Asynchronous Counter is slower as compared to synchronous counter in operation.
3.	Synchronous Counter does not produce any decoding errors.	Asynchronous Counter produces decoding error.
4.	Synchronous Counter is also called Parallel Counter.	Asynchronous Counter is also called Serial Counter.
5.	Synchronous Counter designing as well implementation are complex due to increasing the number of states.	Asynchronous Counter designing as well as implementation is very easy.
6.	Synchronous Counter will operate in any desired count sequence (UP/DOWN/Up and Down).	Asynchronous Counter will operate only in fixed count sequence (UP/DOWN).
7.	Synchronous Counter examples are: , .	Asynchronous Counter examples are: UP counter, Ripple DOWN counter.
8.	In synchronous counter, propagation delay is less.	In asynchronous counter, there is high propagation delay.

Key	Synchronous Counter	Asynchronous Counter
Trigger	In case of Synchronous Counters, all the constituent flip-flops are triggered with same clock simultaneously.	In case of Asynchronous Counters, there is triggering of different flip-flops with different clock.
Operation Speed	Operation speed of a synchronous counter is faster as compared to that of an asynchronous counter.	The operation speed of an asynchronous counter is comparatively slower than a synchronous counter.
Error Prone	Synchronous Counters are less error-prone; they hardly produce any decoding errors because each flip-flop is individually clocked.	Asynchronous Counters are more error-prone and produce decoding errors in the system.
Complexity	All the flip-flops in a synchronous counter coordinate with the clock, hence its design and implementation is complex as compared to that of an asynchronous counter.	In an asynchronous counter, the output of one flip-flop acts as the input of the next flip-flop, hence its design and implementation is quite simple.
Sequence	A Synchronous counter can be operated in any desired count sequence, as it could get manipulated by changing the clock sequence.	An Asynchronous counter can operate only in a fixed count sequence, i.e., UP and DOWN.
Delay	There is no propagation delay observed in case of Synchronous Counters.	In case of asynchronous counters, there is a subsequent propagation delay from one flip-flop to another.

Ripple Counter

- Ripple counter is a special type of Asynchronous counter in which the clock pulse ripples through the circuit.

Features of the Ripple Counter:

- Different types of flip flops with different clock pulse are used.
- It is an example of an asynchronous counter.
- The flip flops are used in toggle mode.
- The external clock pulse is applied to only one flip flop.
- The output of this flip flop is treated as a clock pulse for the next flip flop.
- In counting sequence, the flip flop in which external clock pulse is passed, act as LSB.

Based on their circuitry design, the counters are classified into the following types:

- Up Counter: The up-counter counts the states in ascending order.
- Down Counter: The down counter counts the states in descending order.
- Up-Down Counter: The up and down counter is a special type of bi-directional counter which counts the states either in the forward direction or reverse direction. It also refers to a reversible counter.

Binary Ripple Counter

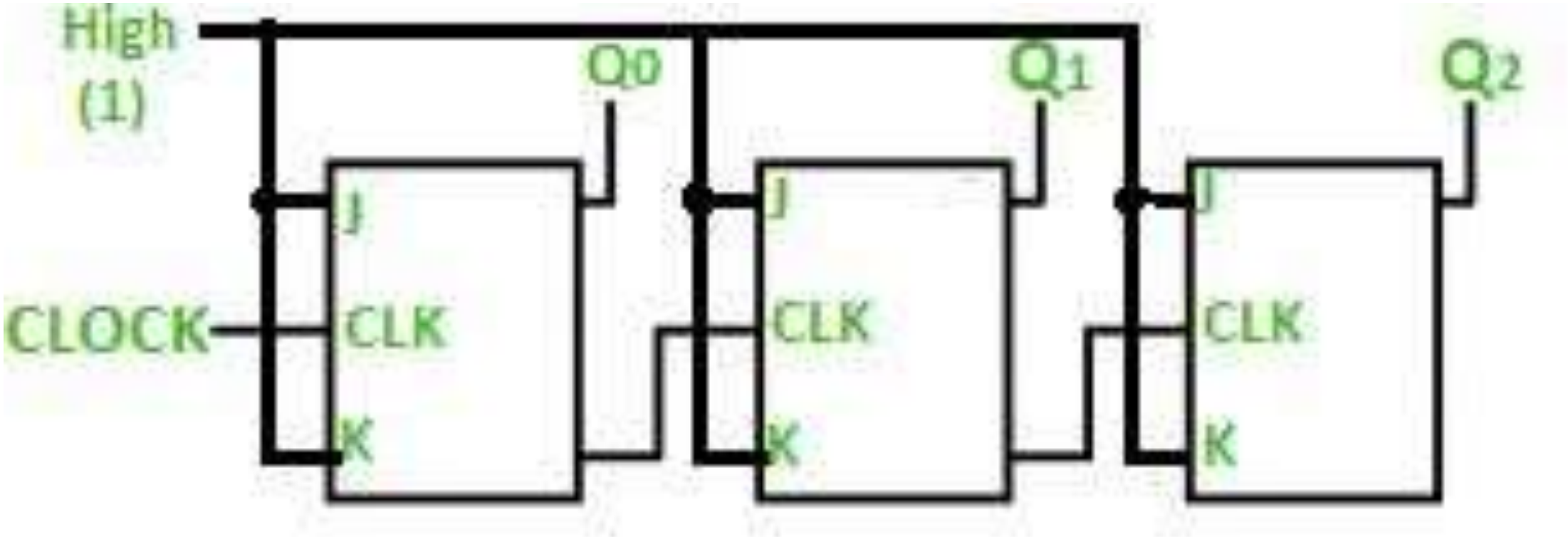
- The flip flops having similar conditions for toggling like T and JK are used to construct the Ripple counter.
- In the circuit design of the binary ripple counter, two JK flip flops are used.
- The high voltage signal is passed to the inputs of both flip flops.
- This high voltage input maintains the flip flops at a state 1. In JK flip flops, the negative triggered clock pulse use.

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 J K 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

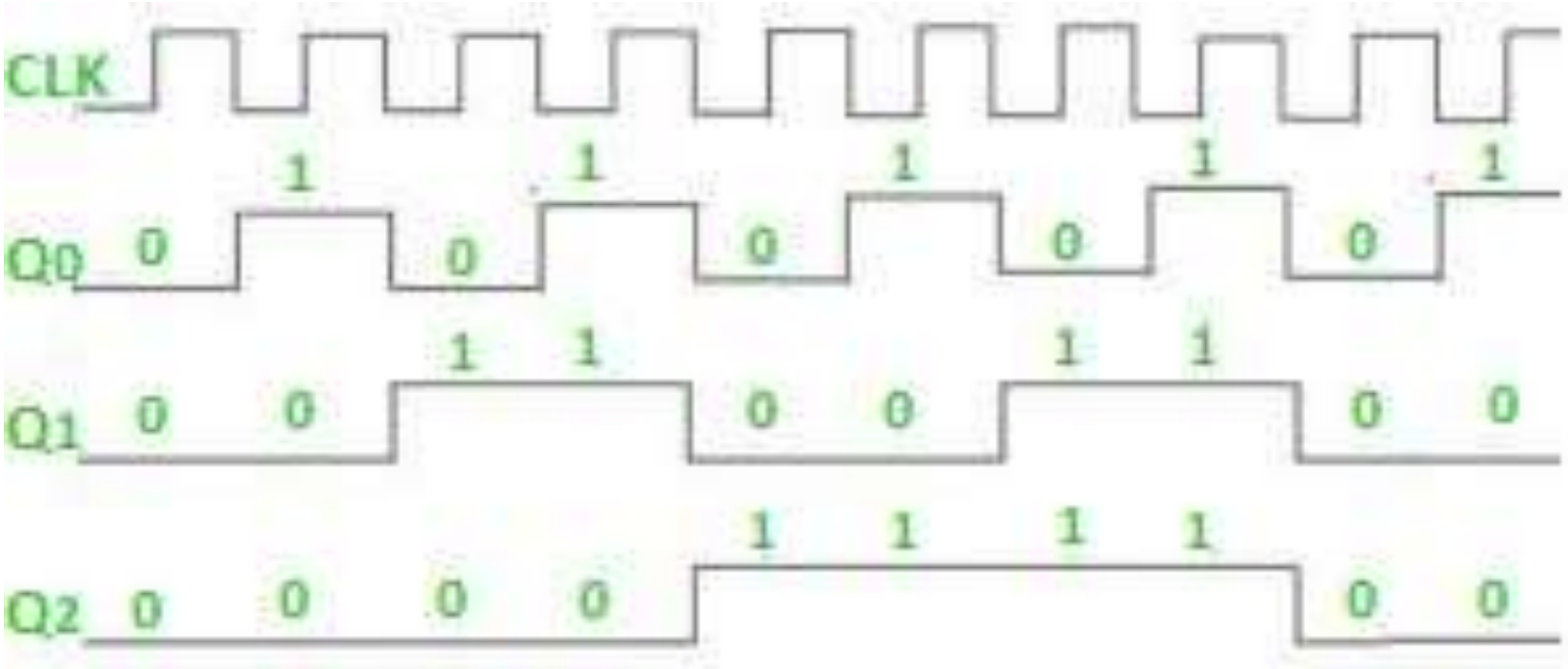
- The 3-bit ripple counter used in the circuit above has eight different states, each one of which represents a count value.

- 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.

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.



Advantages of Ripple Counter in Digital Logic

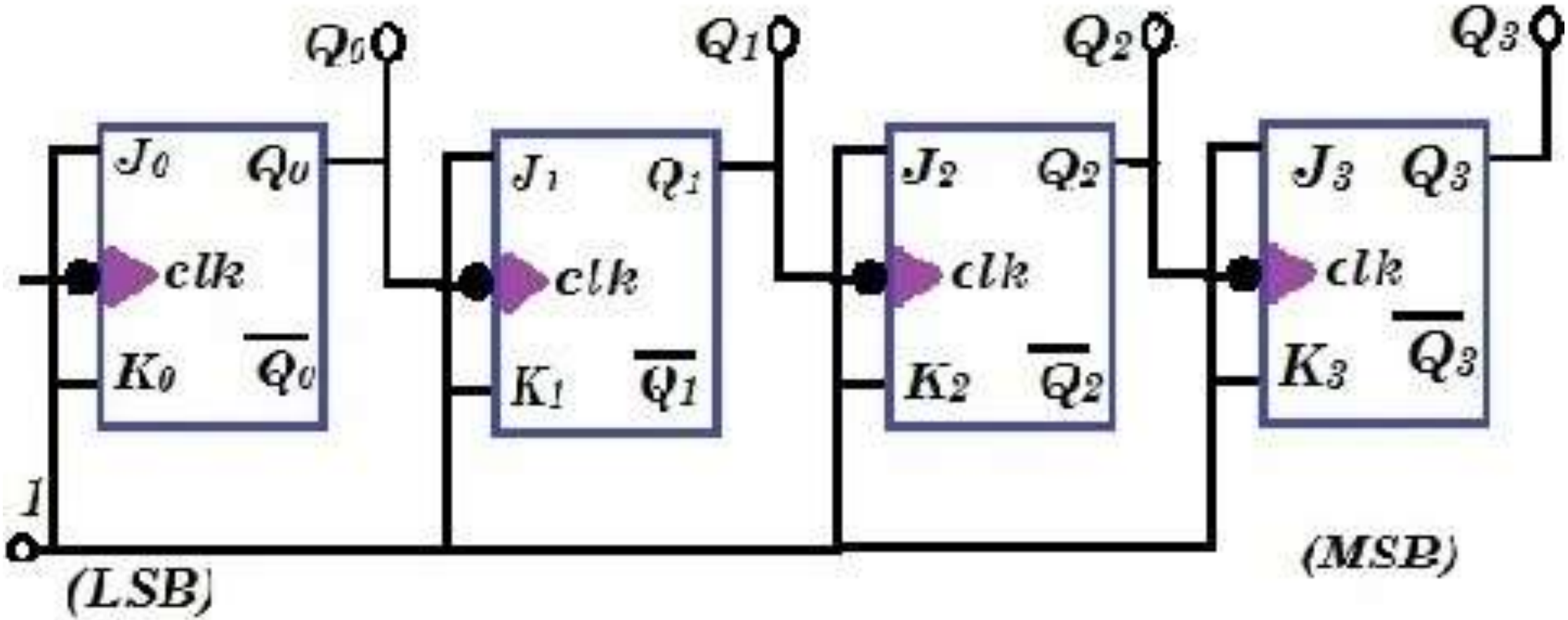
- Can be easily designed by T flip-flop or D Flip Flop.
- Can be used in low speed circuits & divide by n-counters.
- Counters to design any mode number counters (i.e. Mod 4, Mod 3)

Disadvantages of Ripple Counter in Digital Logic

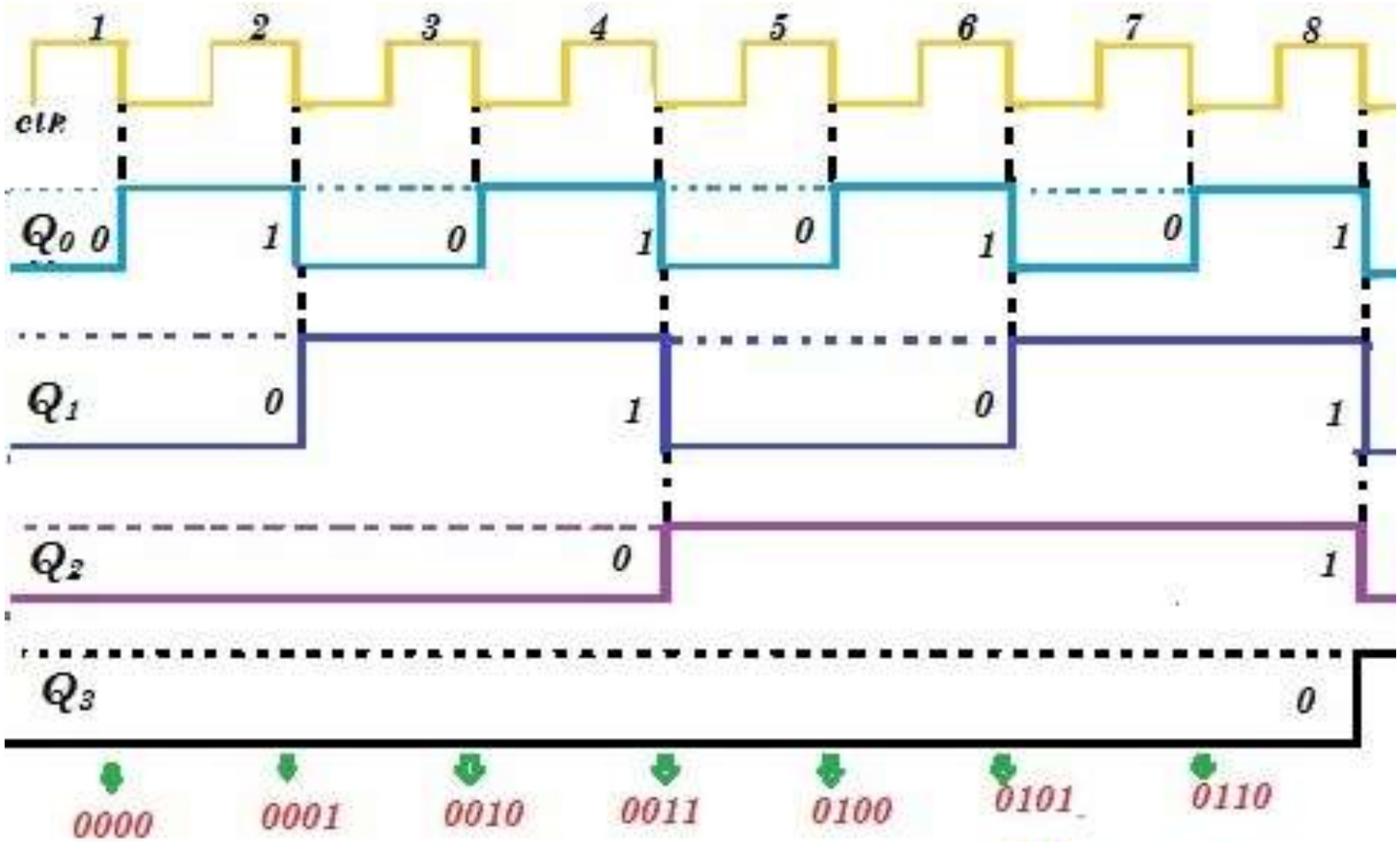
- Extra flip-flop are needed to do resynchronization.
- To count the sequence of truncated counters, additional feedback logic is needed.
- Propagation delay of asynchronous counters is very large, while counting the large number of bits.
- Counting errors may occur due to propagation delay for high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

(Mod 12 Counter) 4-bit Ripple Counter Using JK Flip flop –

- Circuit Diagram and Timing Diagram
- In 4-bit ripple counter, n value is 4 so, 4 JK flip flops are used and the counter can count up to 16 pulses.
- Below the circuit diagram and timing diagram are given along with the truth table.



4 bit Ripple Counter using JK Flip Flop


1
2
3
4
5
6
7
8
clk
Q0 0
1
0
1
0
1
0
1
Q1
0
1
0
1
Q2
0
1
Q3
0
0000
0001
0010
0011
0100
0101
0110

Synchronous Series Counter:

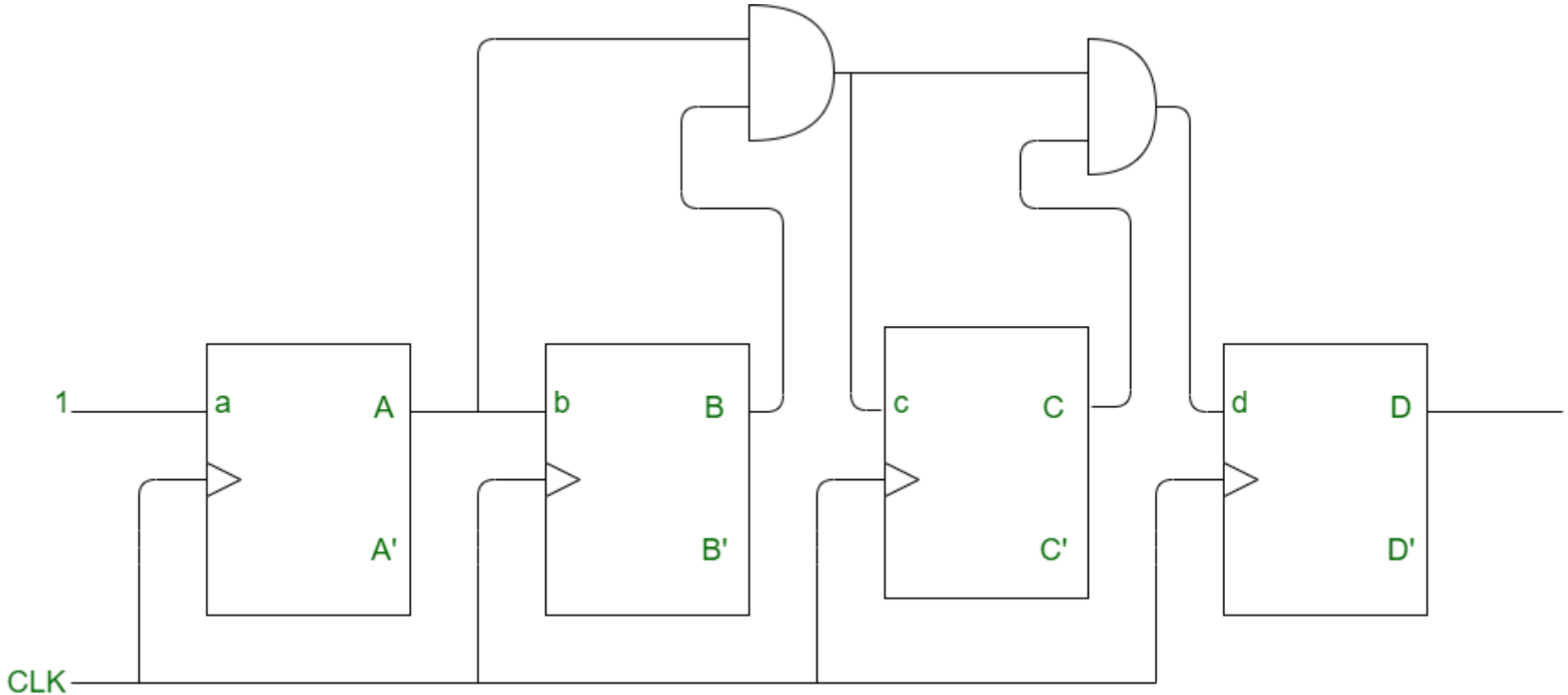
Synchronous Series Counter is such a synchronous counter where inputs of flip-flops are connected in such a manner that only those flip-flops that are toggle on a given clock will have input value as the logic 1.

सिंक्रोनस सीरीज काउंटर एक ऐसा सिंक्रोनस काउंटर है जहां फ्लिप-फ्लॉप के इनपुट इस तरह से जुड़े होते हैं कि केवल वे फ्लिप-फ्लॉप जो किसी दिए गए clk पर टॉगल करते हैं, उनमें तर्क 1 के रूप में इनपुट मान होगा।

The advantage of this counter is that it reduces the decoding error.

Block Diagram of Synchronous Series Counter:

Here is the block diagram of a 4-bit (MOD-16) series counter.


1
a
A
A'
b
B
B'
c
C
C'
d
D
D'
CLK

In the above counter:

- Toggles for every CLK pulse is applied.
- Toggles when A = 1 and CLK pulse is applied.
- Toggles when B = 1 and A = 1 and CLK pulse is applied.
- Toggles when C = B = A = 1 and CLK pulse is applied.

Truth Table for Series Carry Counter:

Truth Table for series carry counter

CLK	D	C	B	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
16	0	0	0	0

What do you mean by decoding a counter?

Counter decoding can be used to shorten sequence, to enable other logic circuits when a specific count state is reached.

Decoding error in counters:

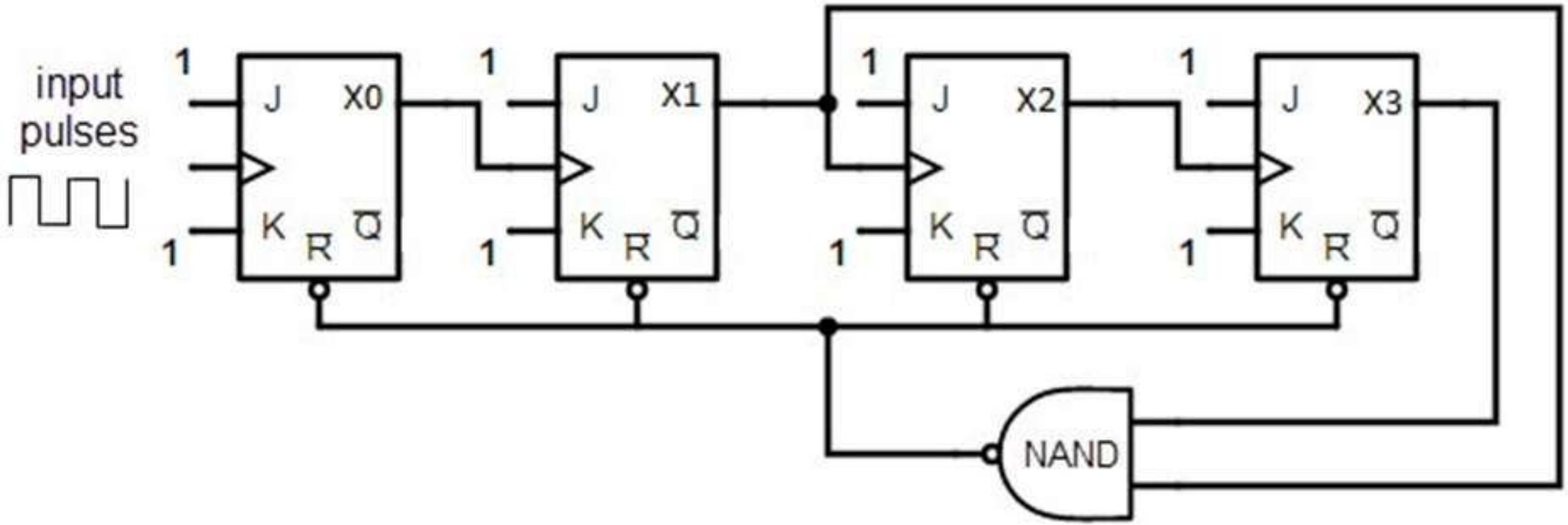
Decoding errors are not produced in synchronous counters while asynchronous counters are likely to produce decoding errors due to the reason that in asynchronous counter the output of the previous flip-flop acts as the clock signal to the adjacent flip flop.

What is a BCD Counter or Decade Counter?

- A BCD counter is one of the 4-bit binary counters, which counts from 0 to a pre-determined count with an applied clock signal.
- When the count reaches the predetermined count value, it resets all the flip-flops and starts to count again from 0.
- This type of counter is designed by using 4 JK flip flops and counts from 0 to 9, and the result is represented in digital form.
- After reaching the count of 9 (1001), it resets and starts again.

BCD or Decade Counter Circuit

- BCD or decade counter circuit is designed by using JK flip flops and NAND gate.
- The BCD counter design is very simple, and it requires 4 JK flip flops because it is a 4-bit binary counter.
- The design of the decade counter is shown below.
- This is an asynchronous implementation of a cascadable, 4-bit, binary-coded decimal counter. In total, the circuits needs just the four flipflops and one additional AND gate.



BCD or Decade Counter Circuit

- From the figure, we observe that the outputs of J and K are connected to logic 1.
- The input pulses or clock input of each flip flop is fed as output to the next flip flop, but not the last flip flop.
- The CLR input is fed to the NAND gate output parallelly to reset all the flip flops when the count reaches.

Decade Counter Operation

At an initial stage (before operation), the count of the decade counter is 0000.

When the clock signal is given as input, then the operation starts and counts the binary output.

For the first clock pulse, the decade counter counts up to 9 (1001).

The decade counter counts 0 to 9 for a given clock signal. When it reaches the count, it resets all the flip-flops and the cycle is repeated.

When the inputs X1 and X3 of the NAND gate is high, the output will be low.

If the output of the NAND gate is connected to the clear input, then it resets all the stages of flip flops of the decade counter.

That means when the input pulse reaches the count from 0 to 9, then it stops counting and starts the count from 0 again.

The Truth Table of Decade Counter

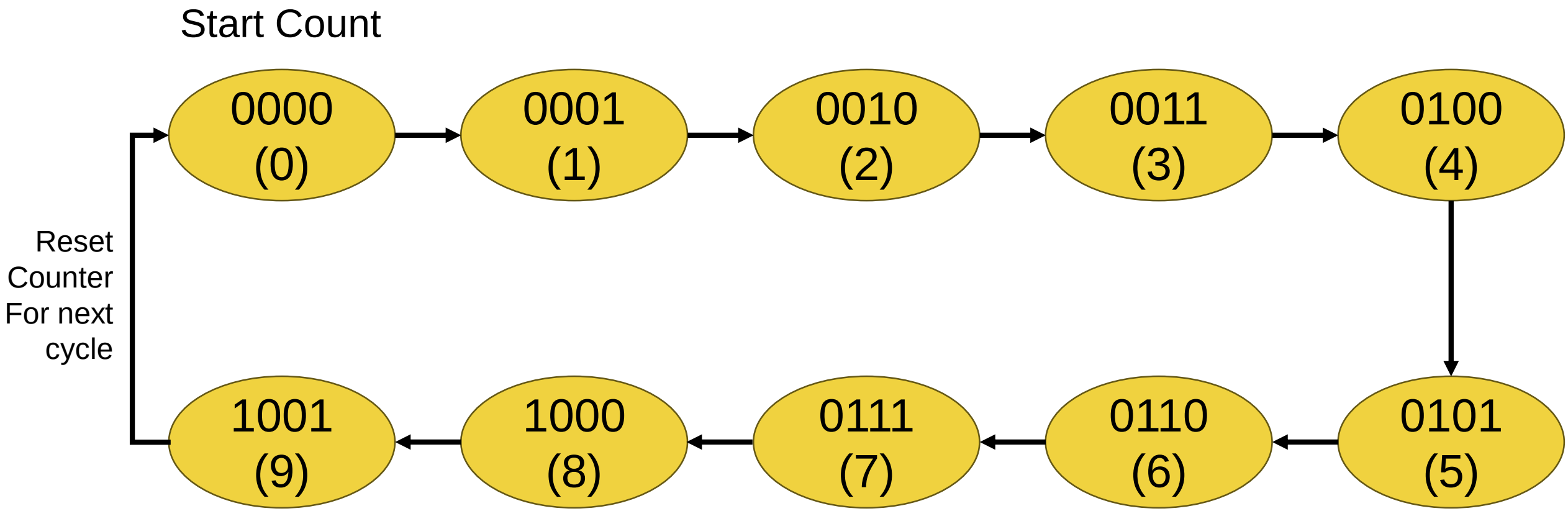
Input Pulses/Clock Pulses	QD	QC	QB	QA
0	0	0	0	0 (Resets)
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

It counts the decimal input pulses and displays the output in binary form. The output of the NAND gate is zero when the input pulse count reaches 9 (1001).

The inputs X1 and X3 of the NAND gate decode the count to display the output in binary form. When the count reaches 9, it resets all the flip-flops by triggering the output of the NAND gate from 1 to 0.

State Diagram of Decade Counter

The state diagram of the decade counter is shown below.



State Diagram

Similarly, the BCD counter is a Mod-10 counter, which resets to zero after counting from 0 (0000) to 9 (1001), represents the result in decimal form.

Hence, it is called a binary coded decimal counter (BCD Counter).

Note:

Q. In a JK Binary Counter from 0 to 9, why is the NAND gate connected to the second and fourth J-K flip flop and not the first and fourth?

Ans. The circuit you cited is a ripple counter, not a synchronous counter. It actually has eleven states, 0000 through 1010, but as soon as the last state is reached, the NAND gate immediately (asychronously) resets the flip-flops to the 0000 state.

What did we learn?

Digital Electronics

Er. L N Soni

AKS University, Satna

UNIT - 4
Shift Register

Register

One flip-flop can store one bit of information. In order to store multiple bits of information, we require multiple flip-flops.

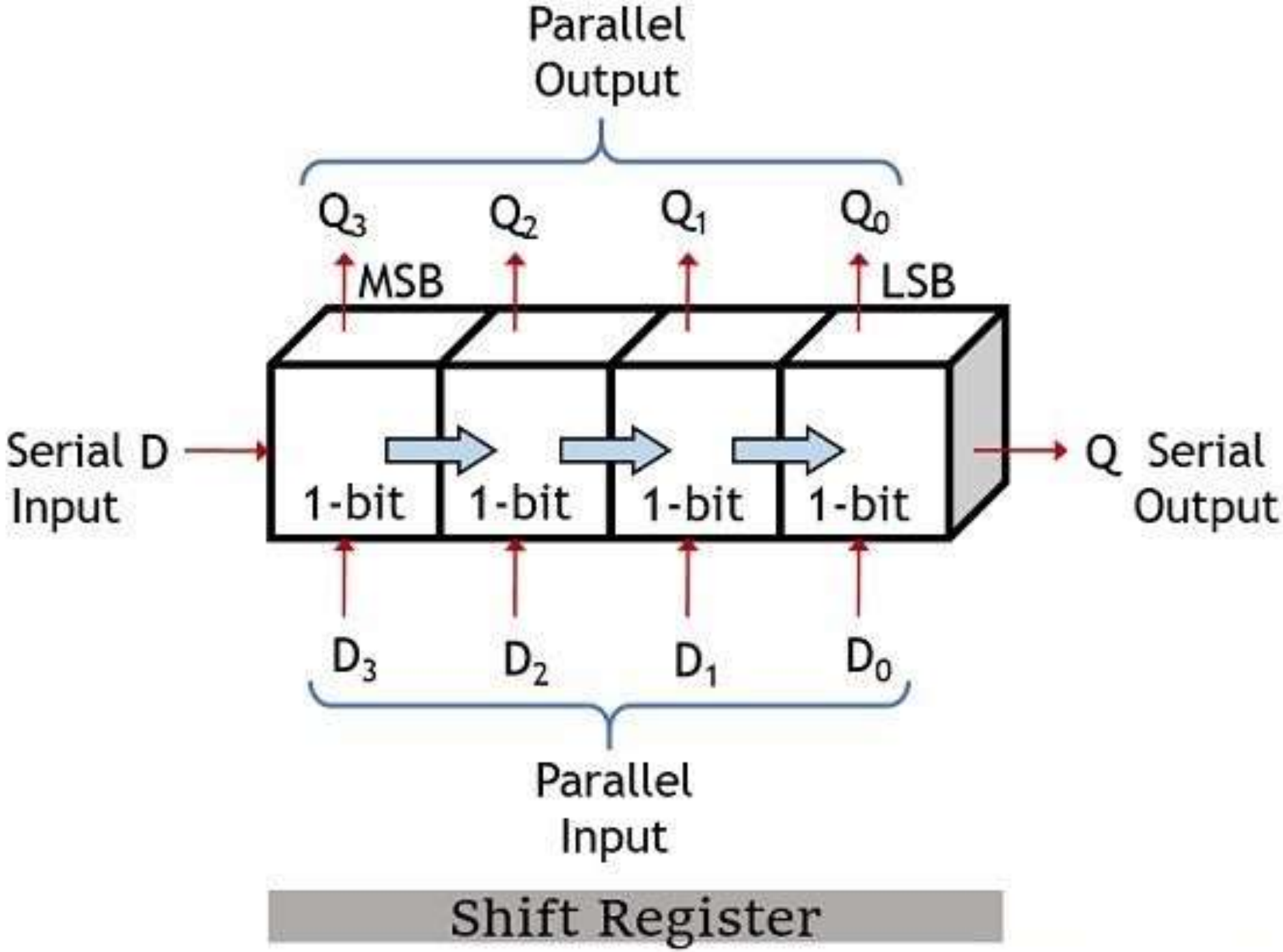
The group of flip-flops, which are used to hold store the binary data is known as the **register**.

A **shift register** is a sequential logic circuit that acts as a unit to store and transfer binary data.

Basically shift registers are bidirectional FIFO circuit that shifts every single bit of the data present in its input towards its output on each clock pulse.

Shift registers are formed by the serial combination of D flip-flops, where each flip-flop in the arrangement holds single data bit.

The serial arrangement permits the output of one flip-flop to act as input to other and this allows the shifting of data bit inside the register.



Shift Register

If the register is capable of shifting bits either towards right hand side or towards left hand side is known as shift register.

An ‘N’ bit shift register contains ‘N’ flip-flops.

Following are the four types of shift registers based on applying inputs and accessing of outputs.

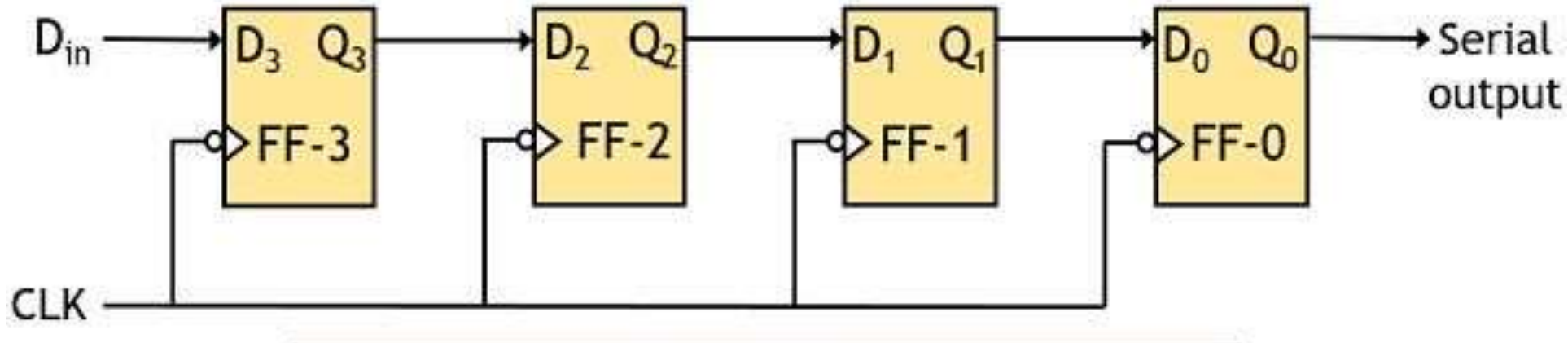
- Serial In − Serial Out shift register
- Serial In − Parallel Out shift register
- Parallel In − Serial Out shift register
- Parallel In − Parallel Out shift register

- **SISO:** Serial-in Serial-out: It permits the insertion of data serially and taking the output also in a serial manner.
- **SIPO:** Serial-in Parallel-out: Here the data is inserted serially either from the left or right direction. But the output is taken parallely.
- **PISO:** Parallel-in Serial-out: This type of shift register allows the parallel input of data bit, but the output is taken serially.
- **PIPO:** Parallel-in Parallel-out: PIPO shift register permits both in and out of data bit in a parallel manner.

Operation of SISO Shift Register

- As we have already discussed that a SISO is a type of shift register in which the input is fed serially and output is also taken in serial manner.

- So consider a connection of 4 D flip-flops D0 to D3 as shown in the figure below:



4-bit SISO Shift Register

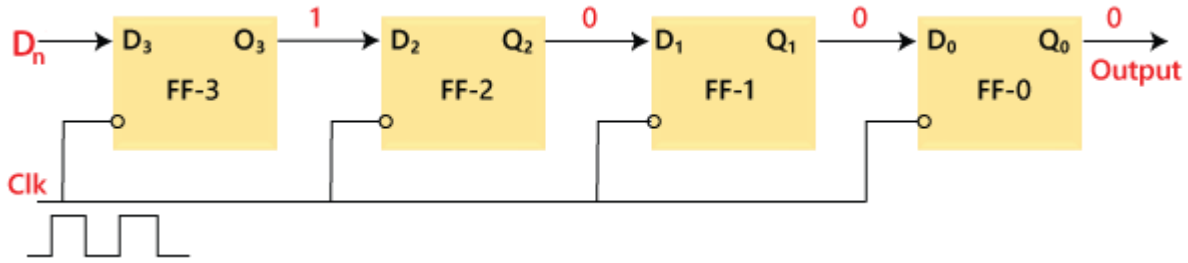
Initially, we consider all the flip-flops are at reset mode. Thus the output of each flip-flop in the arrangement is logic low i.e., 0.

Here we have assumed a shift right mode circuit as the data input is present at the left end while the stored bit is getting shifted towards the right so as to provide serial output.

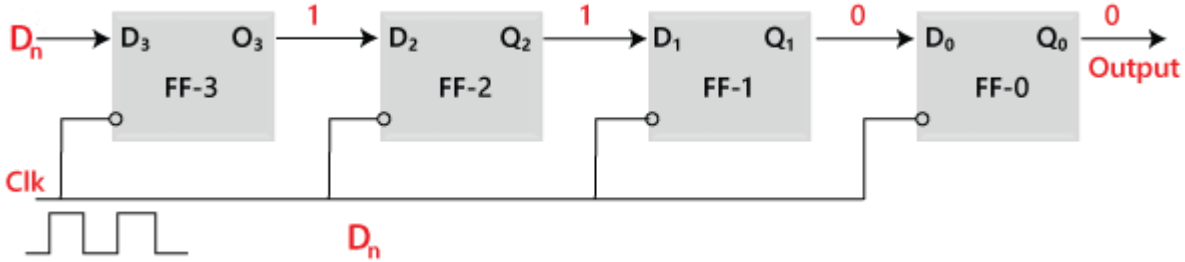
Let us now understand how data is stored in a shift register. Suppose we have to insert ‘1111’ inside the shift register.

Initially, as the device is in reset mode thus the output of each register will be low, thereby providing output of all the 4 registers as 0000.

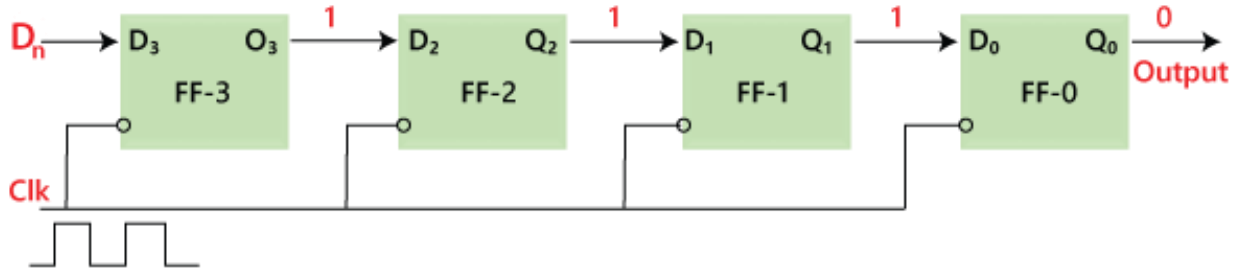
- Now beginning with left of the data to be inserted so 1 is provided as input to the circuit i.e., D3 = 1. But as initially the output of all the flip-flops were 0. Therefore, D2, D1 and D0 will be 0. While input D3 = 1 will cause Q3 to be 1. Thus the overall output will be 1000.



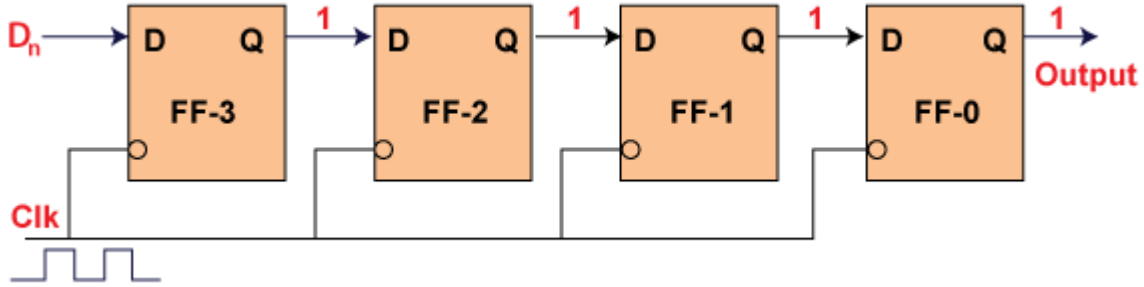
- Further, when another data input bit i.e., 1 is provided at D3. Then again this will cause Q3 to be 1, but as Q3 is provided as input to D2. Therefore, this will cause Q2 to be 1, while rest all other outputs will be 0. Thus for a second falling edge, we will get 11 at the stored bit inside the register, thereby giving the overall output as 1100.



- Similarly, when 3rd input bit ‘1’ is provided then previous output Q2 will cause the input D1 to be 1. This will provide output Q3, Q2 and Q1 as 1 while Q0 will still remain 0. Thus the overall output will be 1110.



- Furthermore, when MSB of the data is provided as input, then 1 at Q1 will cause D0 input to be logic high. So, this will cause Q0 to be 1.

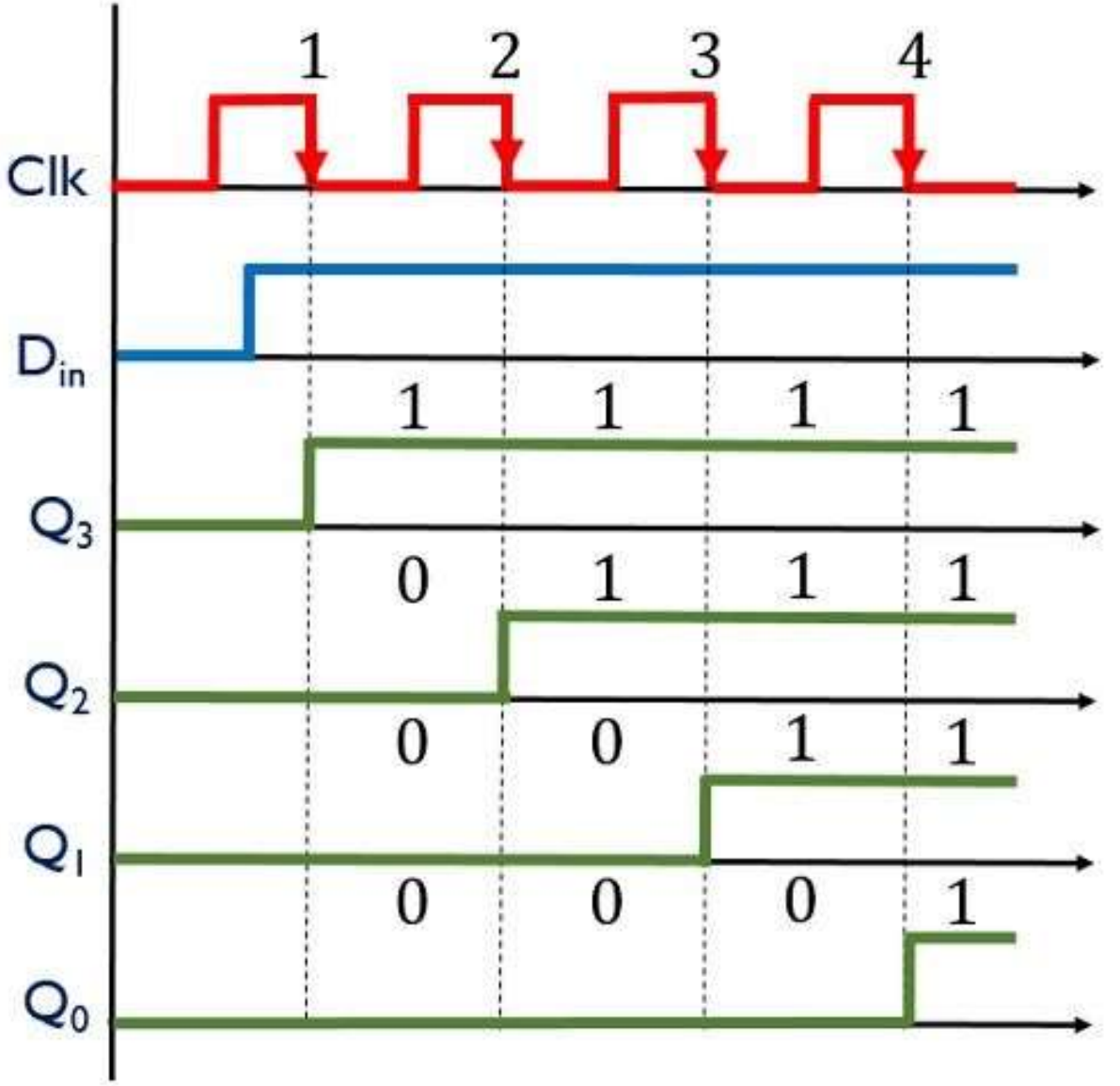


- Thus in this way shift register stores ‘1111‘ thereby showing in the output.

The figure below represents the truth table for the 4-bit SISO shift register:

CLK	Q_3	Q_2	Q_1	Q_0
Initially	0	0	0	0
1st falling edge	1	0	0	0
2nd falling edge	1	1	0	0
3rd falling edge	1	1	1	0
4th falling edge	1	1	1	1

Considering the table let us have a look at the waveform representation of a SISO shift register:



Waveform Representation of SISO Shift Register

The above figure represents the 4 data output of the flip-flops.

Initially, all the outputs were 0, this is clearly shown in the waveform representation. However, the output Q3 changes from 0 to 1 on arrival of first clock pulse. While rest other outputs are still 0.

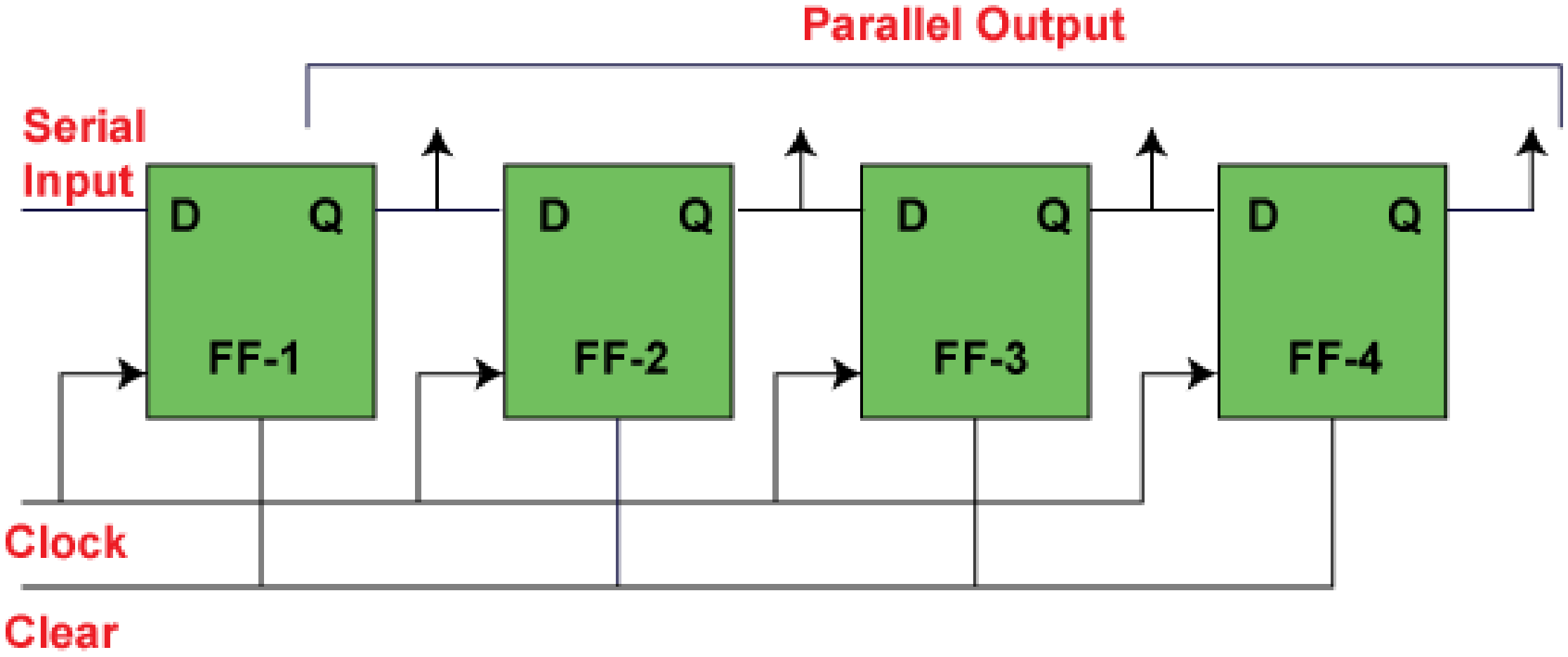
In a similar way, the second clock pulse changes Q2 from 0 to 1. Thus now both Q3 and Q2 are showing logic high in the waveform representation.

So, in this way the above figure is clearly showing the SISO operation of the shift register and on the arrival of 4th clock pulse the output of all the 4 registers will be 1. As the storage is performed by shifting each bit on the arrival of each clock pulse thus it is named so.

Serial IN Parallel OUT" shift register

In the "Serial IN Parallel OUT" shift register, the data is passed serially to the flip flop, and outputs are fetched in a parallel way. The data is passed bit by bit in the register, and the output remains disabled until the data is not passed to the data input. When the data is passed to the register, the outputs are enabled, and the flip flops contain their return value

Below is the block diagram of the 4-bit serial in the parallel-out shift register. The circuit having four D flip-flops contains a clear and clock signal to reset these four flip flops. In SIPO, the input of the second flip flop is the output of the first flip flop, and so on. The same clock signal is applied to each flip flop since the flip flops synchronize each other. The parallel outputs are used for communication.


Parallel Output
Serial Input
D
Q
FF-1
D
Q
FF-2
D
Q
FF-3
D
Q
FF-4
Clock
Clear

Parallel IN Serial OUT

In the "Parallel IN Serial OUT" register, the data is entered in a parallel way, and the outcome comes serially.

A four-bit "Parallel IN Serial OUT" register is designed below. The input of the flip flop is the output of the previous Flip Flop.

The input and outputs are connected through the combinational circuit. Through this combinational circuit, the binary input B0, B1, B2, B3 are passed.

The shift mode and the load mode are the two modes in which the "PISO" circuit works.

Load mode

- The bits B0, B1, B2, and B3 are passed to the corresponding flip flops when the second, fourth, and sixth "AND" gates are active.

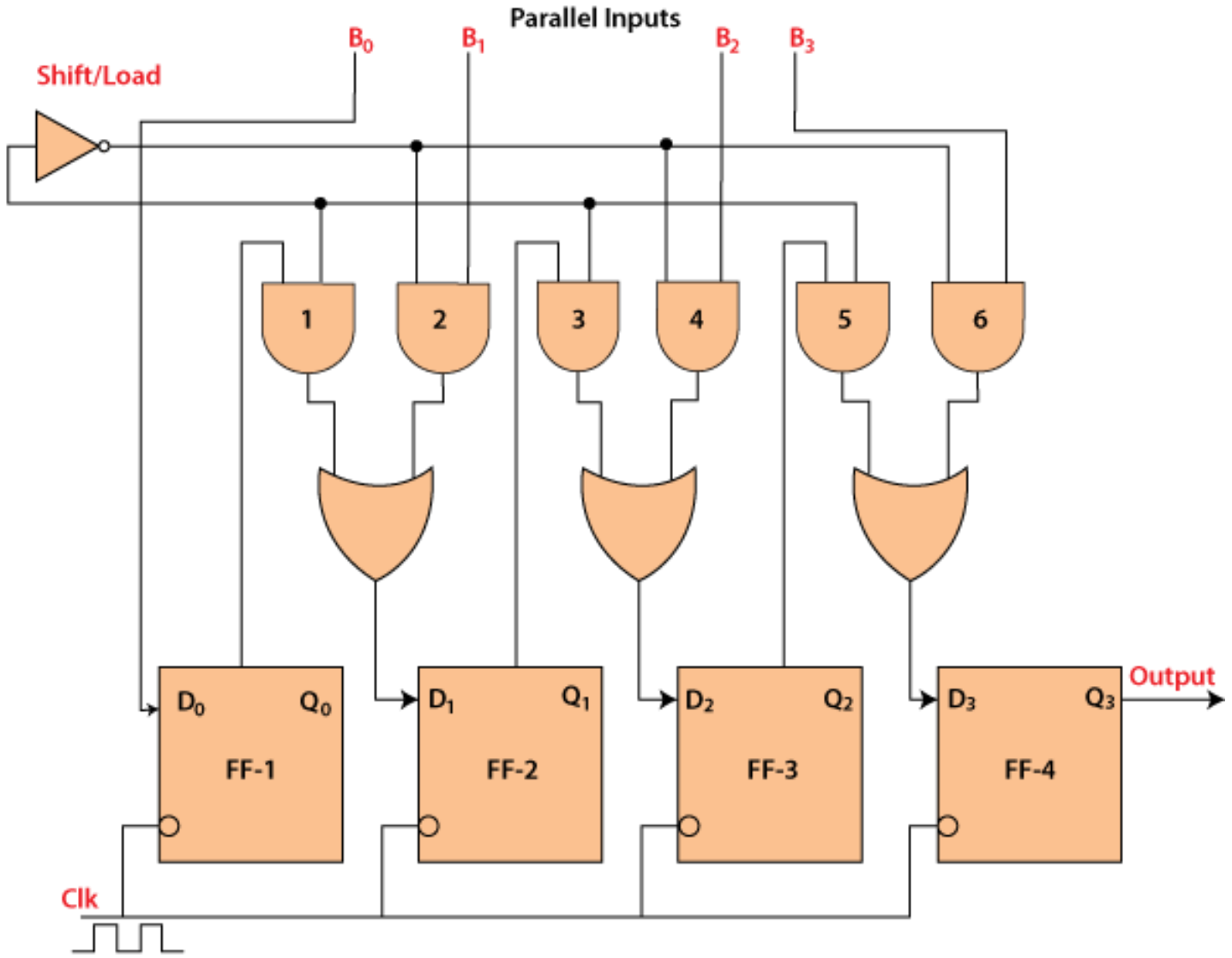
- These gates are active when the shift or load bar line set to 0. The binary inputs B0, B1, B2, and B3 will be loaded into the respective flip-flops when the edge of the clock is low. Thus, parallel loading occurs.

Shift mode

- The second, fourth, and sixth gates are inactive when the load and shift line set to 0. So, we are not able to load data in a parallel way.

- At this time, the first, third, and fifth gates will be activated, and the shifting of the data will be left to the right bit. In this way, the "Parallel IN Serial OUT" operation occurs.

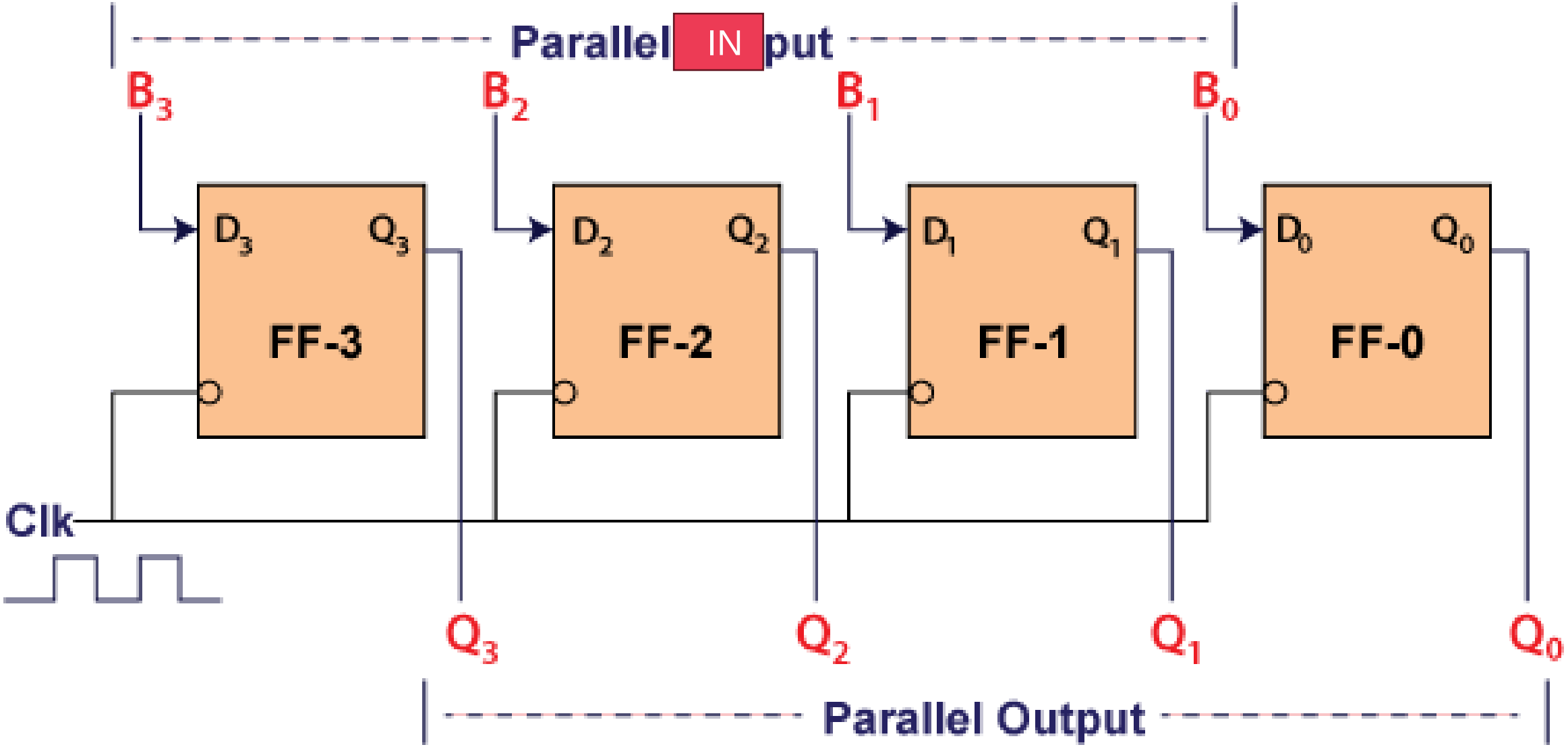
Block Diagram



Parallel IN Parallel OUT

- In "Parallel IN Parallel OUT", the inputs and the outputs come in a parallel way in the register. The inputs A0, A1, A2, and A3, are directly passed to the data inputs D0, D1, D2, and D3 of the respective flip flop.

- The bits of the binary input is loaded to the flip flops when the negative clock edge is applied. The clock pulse is required for loading all the bits. At the output side, the loaded bits appear.


Parallel IN put
B3
B2
B1
B0
D3
Q3
FF-3
D2
Q2
FF-2
D1
Q1
FF-1
D0
Q0
FF-0
Clk
Q3
Q2
Q1
Q0
Parallel Output

Applications of Shift Register

- Shift Register can be used for the temporary storage of data such as buffering and delay circuits.

- The two types of shift registers, parallel-in-parallel-out and serial-in-serial-out are used to produce time delay to digital circuits.

- A shift register can be used to convert data from parallel to shift or vice versa as such a parallel data stream can be converted to a series data stream using a shift register.

- Digital Signal Processing is another application of shift registers like filtering and enhancement. Shift Register is used in the design of finite impulse response.

- Counter Circuits can be implemented by placing the result of flip-flops together to the inputs of others. It counts up or down as per the configuration.

- Liquid Crystal Displays or LCDs use shift registers to convert parallel data to series data to make the segments of LCD function properly.

- Media Compression is the wide-used application of shift registers as algorithms like MPEG and JPEG use shift registers to encode and decode using variable-length codes.

- Serial Communication uses shift registers for transmitting and receiving data in serial communication protocols like serial peripheral interfaces and inter-integrated circuits.

Short Questions:

1. What is a shift register?

A shift register is an electronic device that can store and shift data in a serial manner. It is a collection of flip-flops connected in series, with the output of one flip-flop connected to the input of the next. The shift register can shift the data either left or right, depending on the design.

2. What are the applications of shift registers?

Shift registers are commonly used in digital circuits for a variety of applications, including data storage, serial-to-parallel and parallel-to-serial data conversion, shift and rotate operations, and frequency division. They are also used in communications systems, such as in serial data transmission and reception.

3. What are the different types of shift registers?

There are several types of shift registers, including parallel-in, serial-out (PISO), serial-in, parallel-out (SIPO), serial-in, serial-out (SISO), and parallel-in, parallel-out (PIPO) shift registers. Each type has its own specific application and use case.

4. How does a shift register work?
A shift register works by storing and shifting data in a serial manner. When data is input to the first flip-flop, it is stored there temporarily. When the shift signal is applied, the data is shifted to the next flip-flop, and so on. The output of the shift register is the last bit shifted out, which can be either the most significant or least significant bit depending on the shift direction.

5. What are the advantages of using shift registers?
Shift registers offer several advantages in digital circuit design, including storing and shifting large amounts of data in a small space, low power consumption, and high-speed operation. They also provide flexibility in data manipulation and conversion, making them useful in a wide range of applications.

What did we learn?