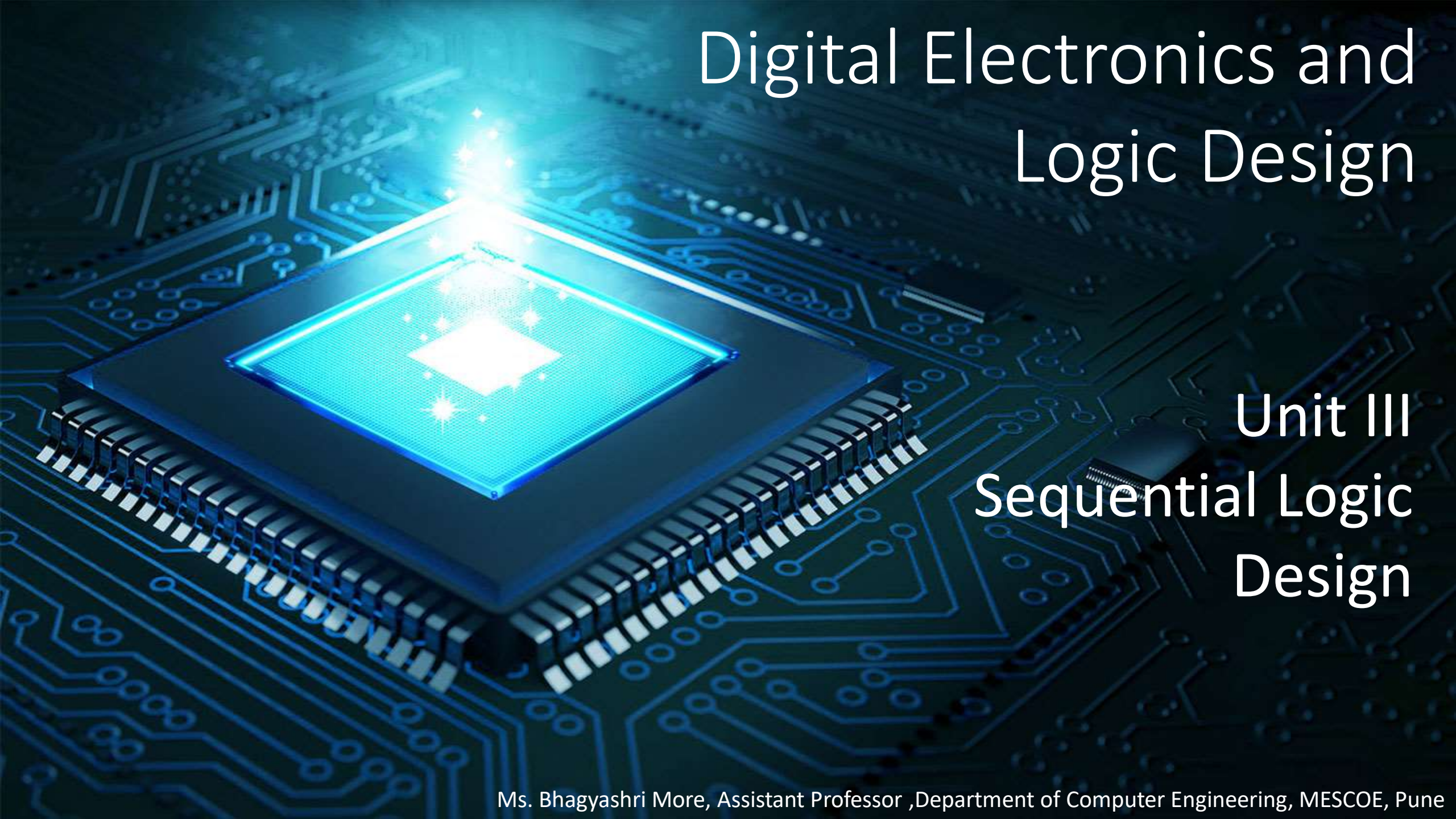


Digital Electronics and Logic Design

Unit III
Sequential Logic Design

Ms. Bhagyashri More, Assistant Professor ,Department of Computer Engineering, MESCOE, Pune

Agenda

01 Flip-Flop: SR, JK,D,T, Preset and clear, Master Slave JK FF, Truth Tables and Excitation Tables, Conversion of FF

02 Registers: SISO, SIPO, PISO, PIPO, Shift registers, Bidirectional shift register, Universal shift register.

03 Counters: Asynchronous counter, synchronous counter, BCD counter, Ring Counter, Johnson Counter, Modulus of counter (IC 7490)

04 Synchronous Sequential Circuit Design: Models- Moore and Mealy, State diagram and state table, Design Procedure, Sequence geneartor and detector.

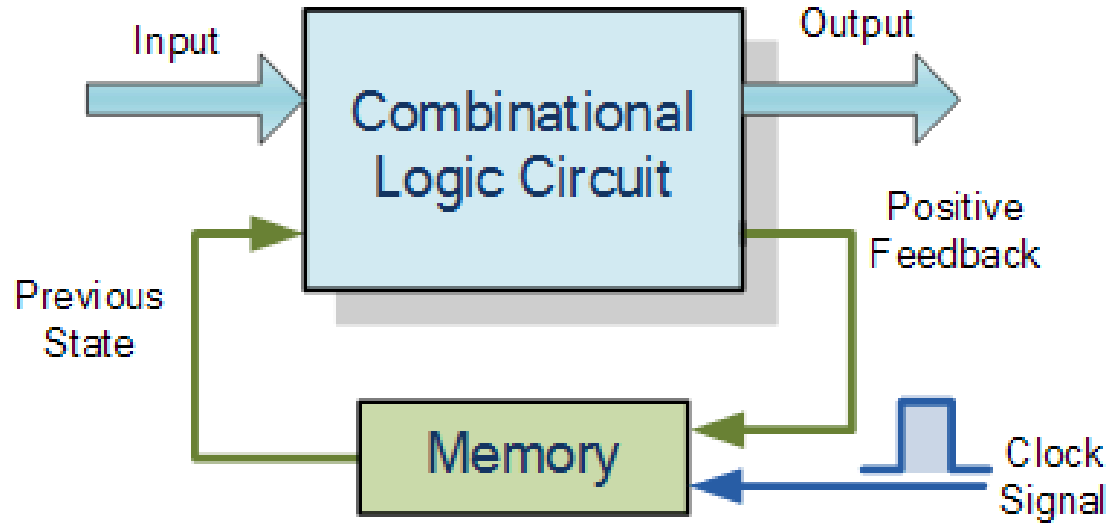
Sequential Circuit

Combinational Circuit:

The combinational circuits have set of outputs, which depends only on the present combination of inputs.

Sequential Circuit:

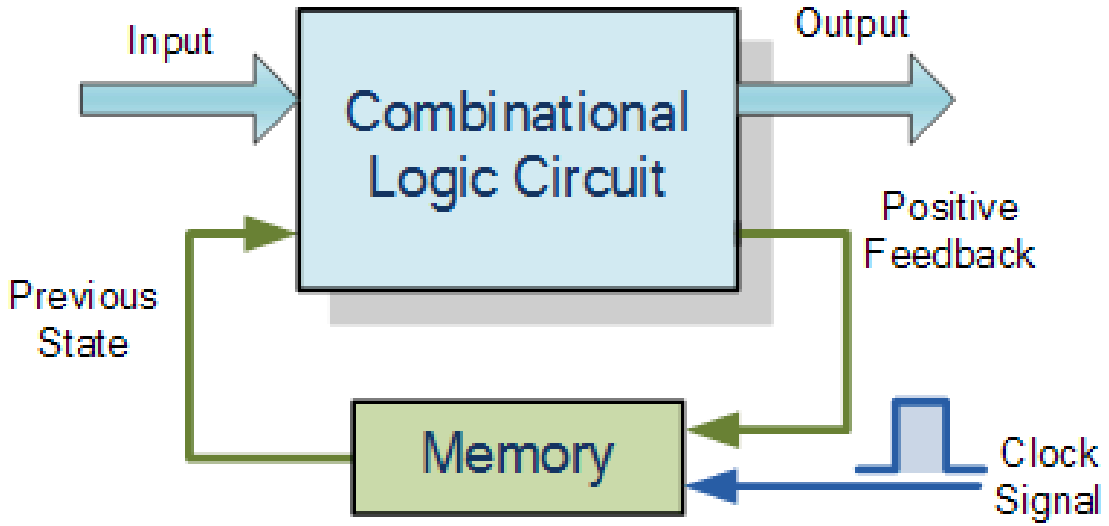
1. The sequential circuit is a special type of circuit that has a series of inputs and outputs. The outputs of the sequential circuits depend on both the combination of present inputs and previous outputs.
2. The previous output is treated as the present state. So, the sequential circuit contains the combinational circuit and its memory storage elements.
3. A sequential circuit doesn't need to always contain a combinational circuit. So, the sequential circuit can contain only the memory element.



Sequential Circuit

Present state: (Qn)
The present state designates the state of flip-flops before the occurrence of a clock pulse.

Next State: (Qn_{+1})
The next state shows the states of flip-flops after the clock pulse, and the output section lists the value of the output variables during the present state.

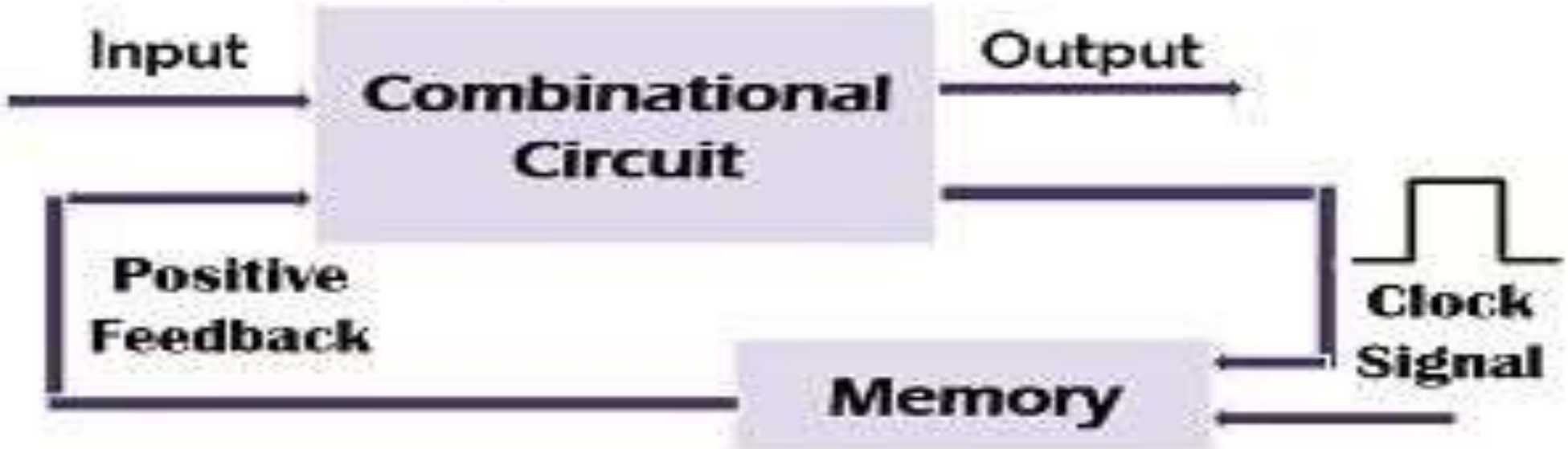


Sr. No.	Key	Combinational Circuit	Sequential Circuit
1	Definition	Combinational Circuit is the type of circuit in which output is independent of time and only relies on the input present at that particular instant.	On other hand Sequential circuit is the type of circuit where output not only relies on the current input but also depends on the previous output.
2	Feedback	In Combinational circuit as output does not depend on the time instant, no feedback is required for its next output generation.	On other hand in case of Sequential circuit output relies on its previous feedback so output of previous input is being transferred as feedback used with input for next output generation.
3	Performance	As the input of current instant is only required in case of Combinational circuit, it is faster and better in performance as compared to that of Sequential circuit.	On other hand Sequential circuit are comparatively slower and has low performance as compared to that of Combinational circuit.
4	Complexity	No implementation of feedback makes the combinational circuit less complex as compared to sequential circuit.	However on other hand implementation of feedback makes sequential circuit more complex as compared to combinational circuit.
5	Elementary Blocks	Elementary building blocks for combinational circuit are logic gates.	On other hand building blocks for sequential circuit are flip flops..
6	Operation	Combinational circuit are mainly used for arithmetic as well as Boolean operations.	On other hand Sequential circuit is mainly used for storing data.
7	Examples	Adder, Subtractor, MUX, DEMUX, Comaparator	Flip Flop, Registers, Counters

Combinational Circuit

Vs

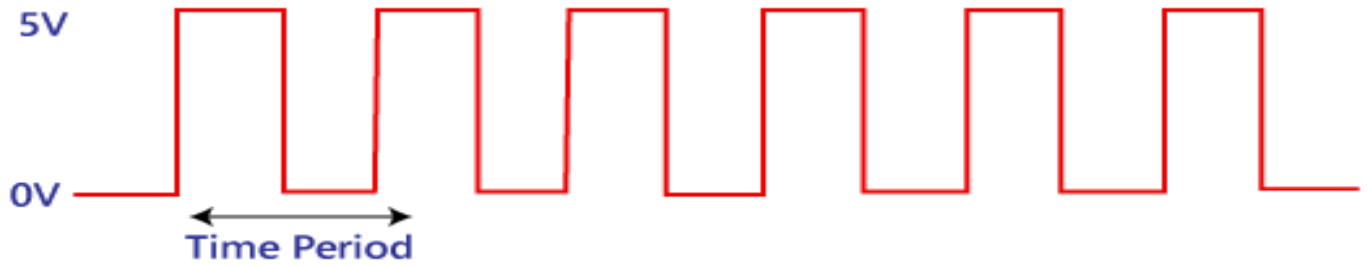
Sequential Circuit



Clock Signal

Clock signal

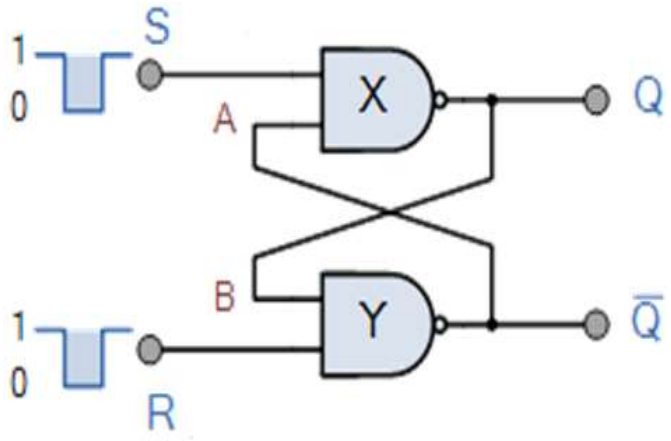
A clock signal is a periodic signal in which ON time and OFF time need not be the same. When ON time and OFF time of the clock signal are the same, a square wave is used to represent the clock signal. Below is a diagram which represents the clock signal:

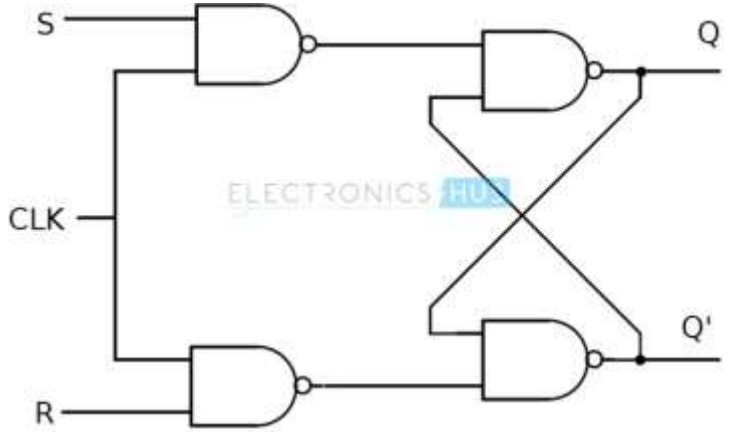


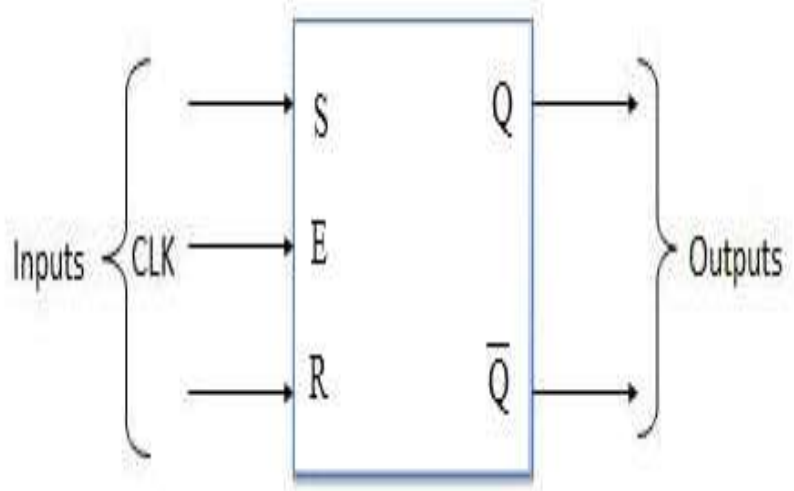
A clock signal is considered as the square wave. Sometimes, the signal stays at logic, either high 5V or low 0V, to an equal amount of time. It repeats with a certain time period, which will be equal to twice the 'ON time' or 'OFF time'.

Basics of Flip Flop

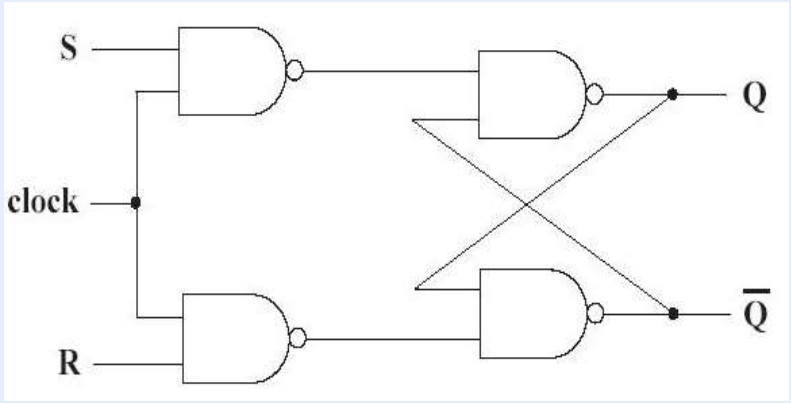
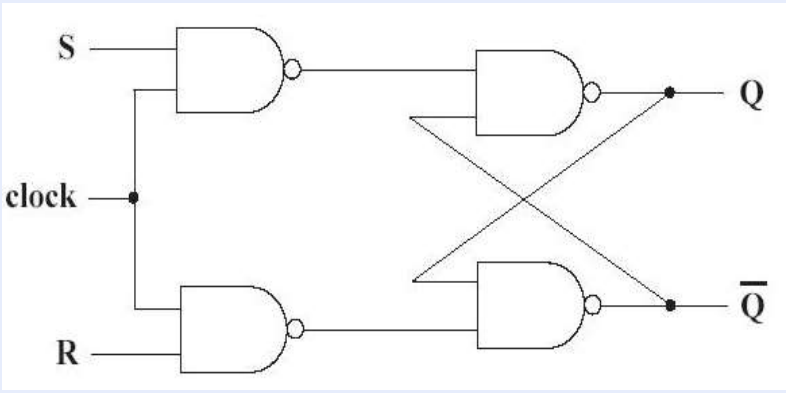
1. A circuit that has two stable states is treated as a **flip flop**. These stable states are used to store binary data that can be changed by applying varying inputs.
2. The flip flops are the fundamental building blocks of the digital system. Flip flops and latches are examples of data storage elements. In the sequential logical circuit, the flip flop is the basic storage element.
3. The latches and flip flops are the basic storage elements but different in working.





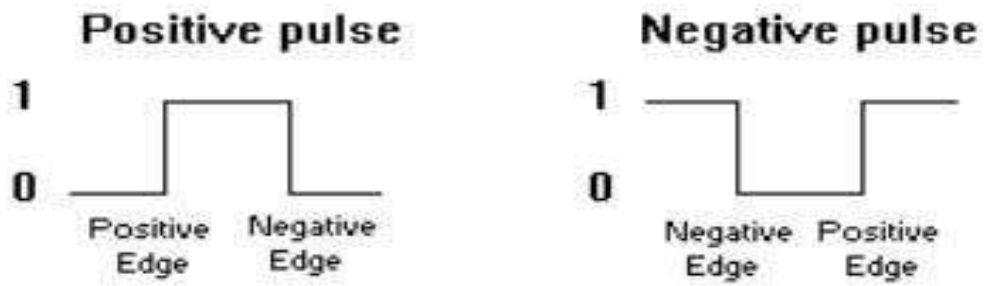


Signed binary number examples

Set state	Reset State
 1 0	 0 1
Q=1 and Q'=0 SET STATE	Q=0 and Q'=1 RESET STATE

Clock Triggering

Clock Pulse Transition:
The movement of a trigger pulse is always from a 0 to 1 and then 1 to 0 of a signal. Thus it takes two transitions in a single signal. When it moves from 0 to 1 it is called a positive transition and when it moves from 1 to 0 it is called a negative transition.



Definition of clock pulse transition

Triggering Method
1. **Level Trigger**
2. **Edge Trigger**

Level Trigger

Level Triggering

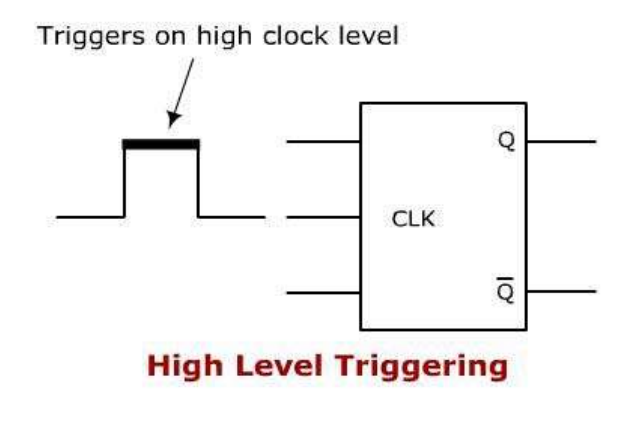
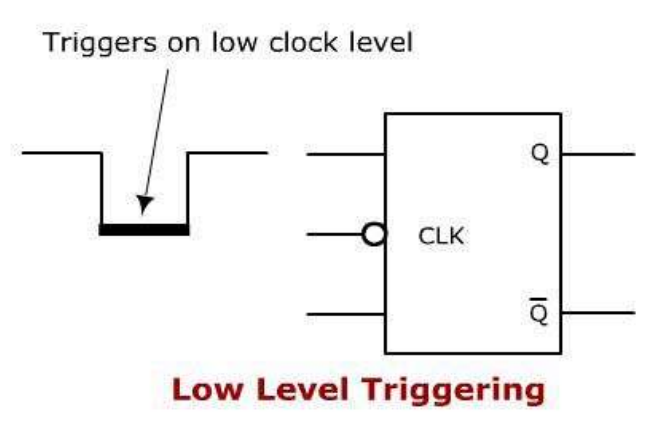
The logic High and logic Low are the two levels in the clock signal. In level triggering, when the clock pulse is at a particular level, only then the circuit is activated. There are the following types of level triggering:

1. **Positive Level Triggering:**

The logic High and logic Low are the two levels in the clock signal. In level triggering, when the clock pulse is at a particular level, only then the circuit is activated.

2. **Negative Level Triggering:**

In negative level triggering, the signal with Logic Low occurs. So, in this triggering, the circuit is operated with such type of clock signal.

Positive Level Triggering	Negative Level Triggering
	

Edge Trigger

Edge Triggering

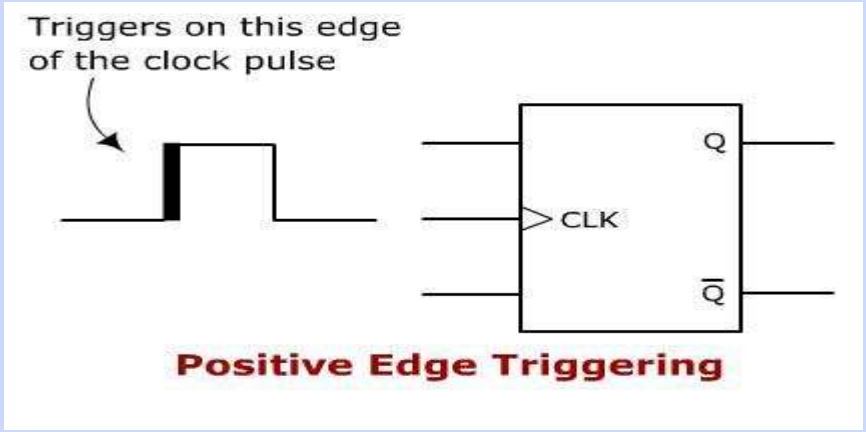
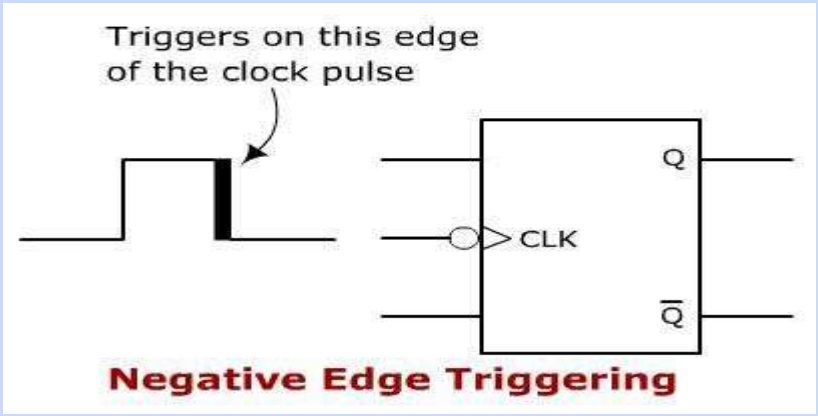
In clock signal of edge triggering, two types of transitions occur, i.e., transition either from Logic Low to Logic High or Logic High to Logic Low. Based on the transitions of the clock signal, there are the following types of edge triggering:

1. **Positive Edge Triggering:**

The transition from Logic Low to Logic High occurs in the clock signal of positive edge triggering. So, in positive edge triggering, the circuit is operated with such type of clock signal.

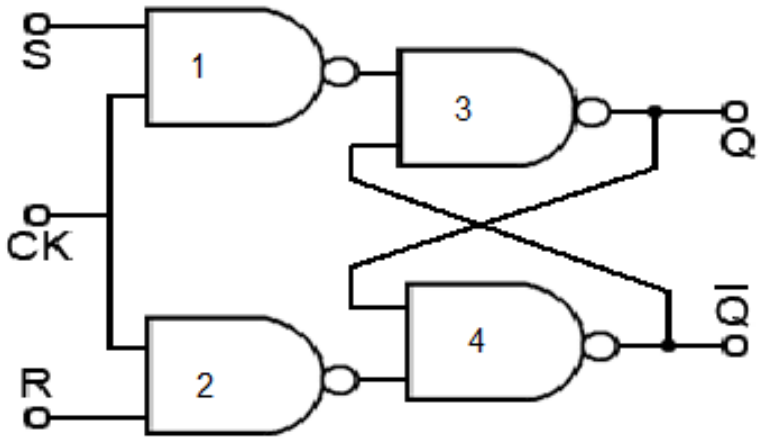
2. **Negative Edge Triggering:**

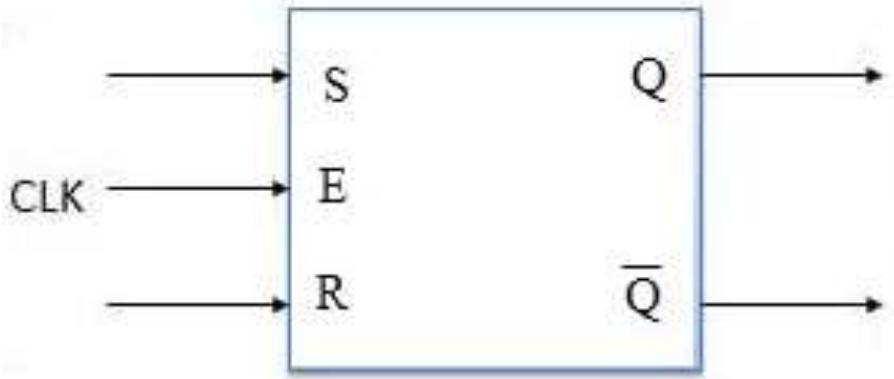
The transition from Logic High to Logic low occurs in the clock signal of negative edge triggering. So, in negative edge triggering, the circuit is operated with such type of clock signal.

Positive Edge Triggering	Negative Edge Triggering
	

SR Flip Flop

SR flip-flop operates with only positive clock transitions or negative clock transitions. Whereas, SR latch operates with enable signal. The **circuit diagram** of SR flip-flop is shown in the following figure.



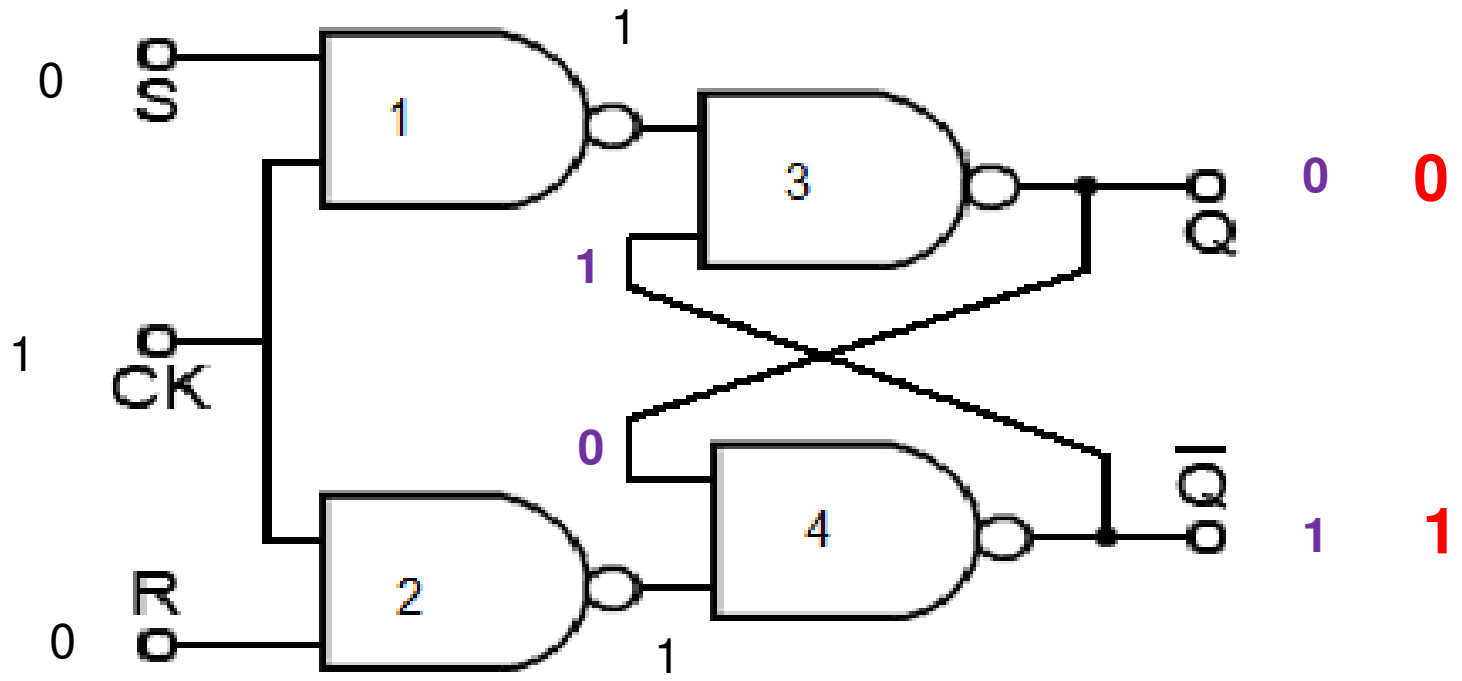


This circuit has two inputs S & R and two outputs Qtt & Qtt'. The operation of SR flipflop is similar to SR Latch. But, this flip-flop affects the outputs only when positive transition of the clock signal is applied instead of active enable.

SR Flip Flop

Operation:

1. S=0 R=0 CLK=1



S= R=0

Consider, Qn = 0 & Qn'=1

Therefor, **Qn+1 = 0**

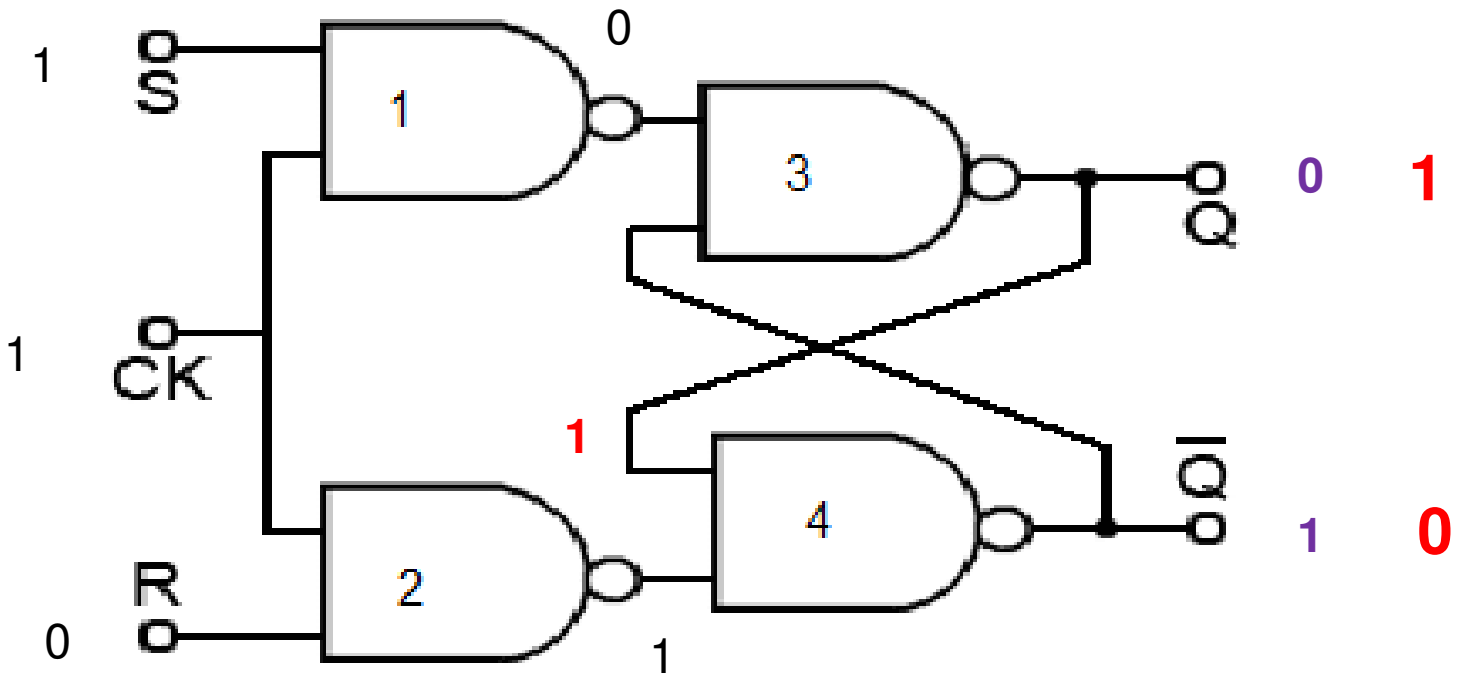
Qn+1' = 1

Present state and next state remain same therefore NO CHAGE n outputs.

SR Flip Flop

Operation:

1. S=1 R=0 CLK=1



S=1 R=0

Consider, Qn = 0 & Qn'=1

Therefore, **Qn+1 = 1**

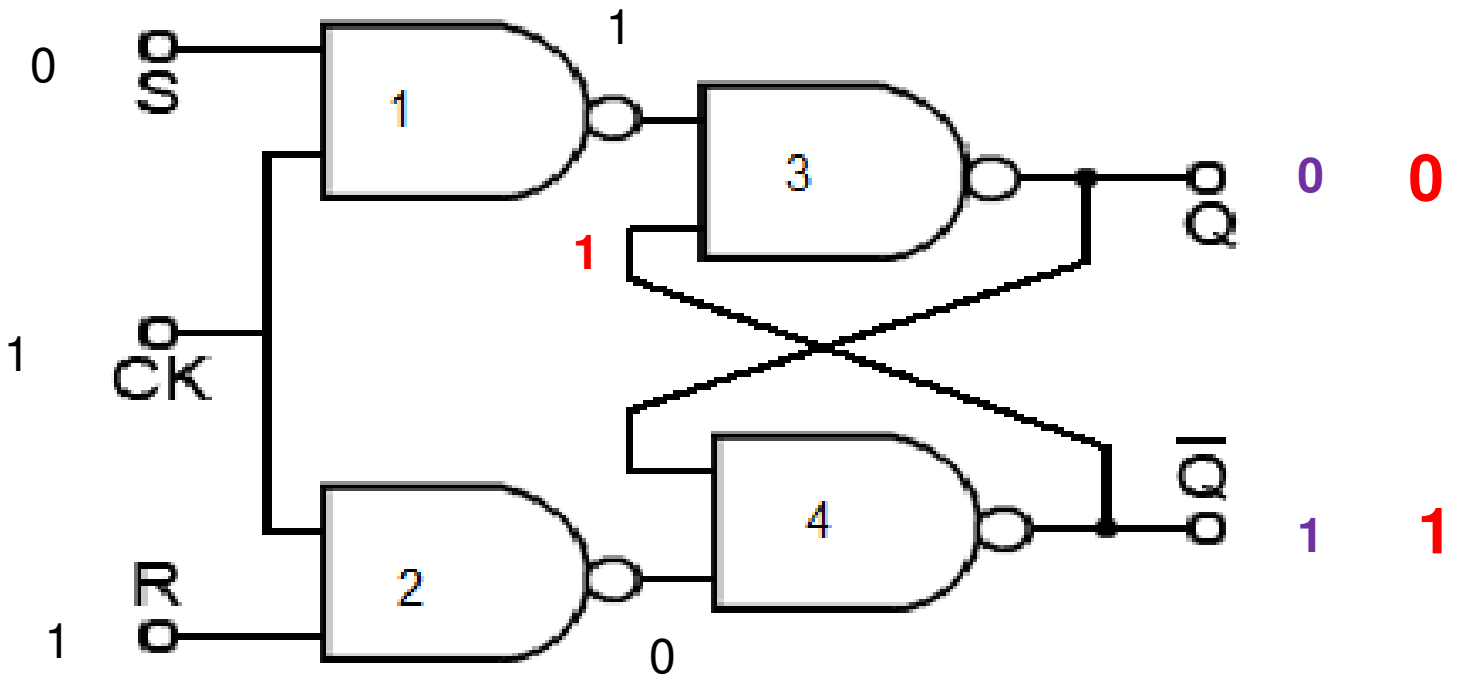
Qn+1' = 0

When S=1 & R=0 then output is in SET STATE.

SR Flip Flop

Operation:

1. S=0 R=1 CLK=1



S=0 R=1

Consider, Qn = 0 & Qn'=1

Therefore, **Qn+1 = 0**

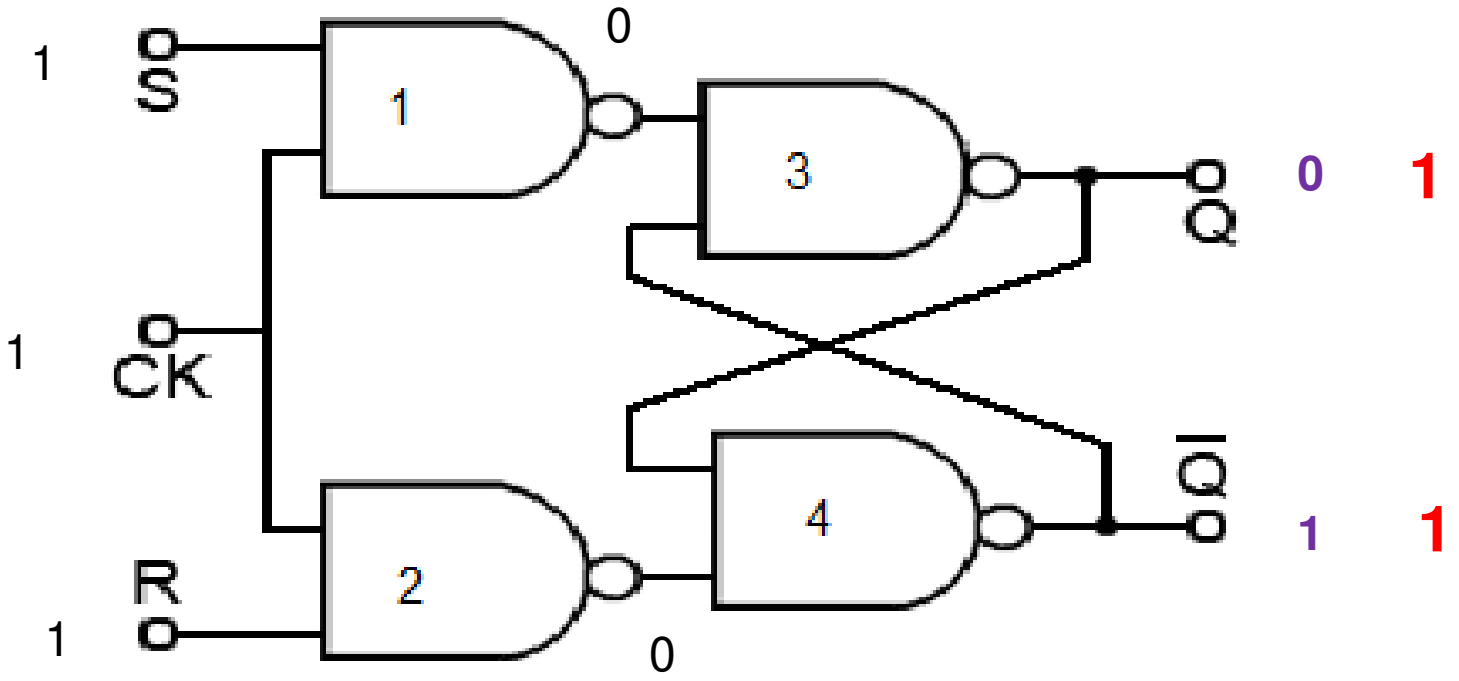
Qn+1' = 1

When S=1 & R=0 then output is in RESET STATE.

SR Flip Flop

Operation:

1. S=1 R=1 CLK=1



OUTPUTS ARE NOT ACCEPTABLE

S=1 R=1

Consider, Qn = 0 & Qn'=1

Here, NAND gate 3 and 4 try to become 1 but it is not acceptable.

Therefore,

It is INVALID or INDETERMINATE condition.

SR Flip Flop

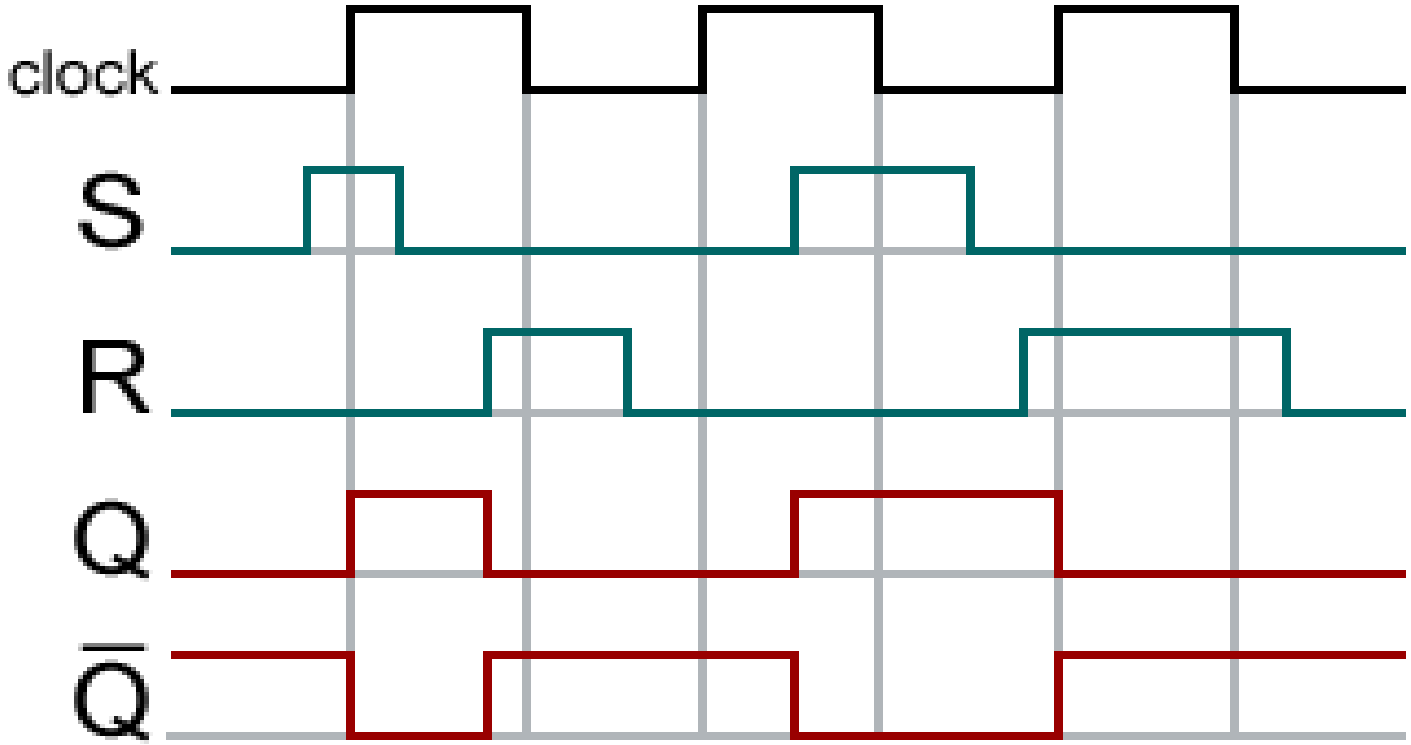
Truth Table:

CLK	S	R	Qn+1	Status
1	0	0	Qn	No change
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	X	Invalid

S	R	Qn	Qn'	Qn+1	Qn+1'	State
		Present state		Next state		
0	0	0	1	0	1	No change
0	0	1	0	1	0	
0	1	0	1	0	1	RESET
0	1	1	0	0	1	
1	0	0	1	1	0	SET
1	0	1	0	1	0	
1	1	0	1	X	X	INVALID
1	1	1	0	X	X	

SR Flip Flop

TIMING DIAGRAM:



Excitation table of SR FF

Truth Table:

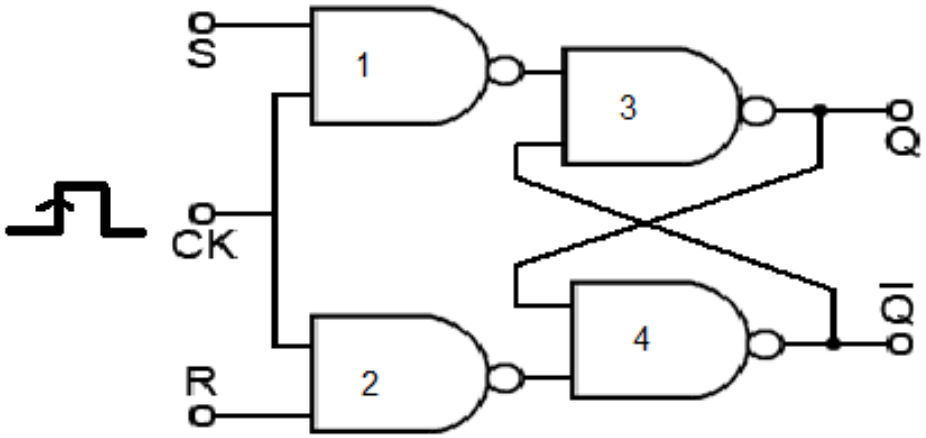
CLK	S	R	Qn+1	Status
1	0	0	Qn	No change
1	0	1	0	Reset
1	1	0	1	Set
1	1	0	X	Invalid

Excitation Table:

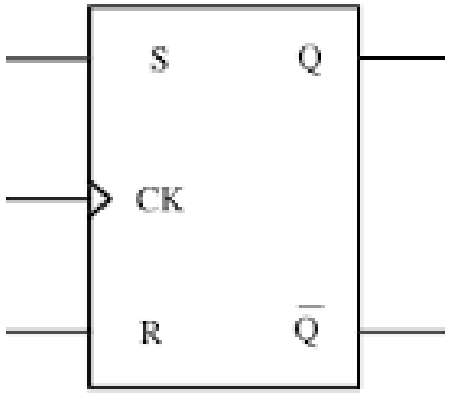
Present state	Next State	Required input	
Qn	Qn+1	S	R
0	0		
0	1		
1	0		
1	1		

Positive Edge Triggered SR FF

SR Flip Flop:



Block Diagram:

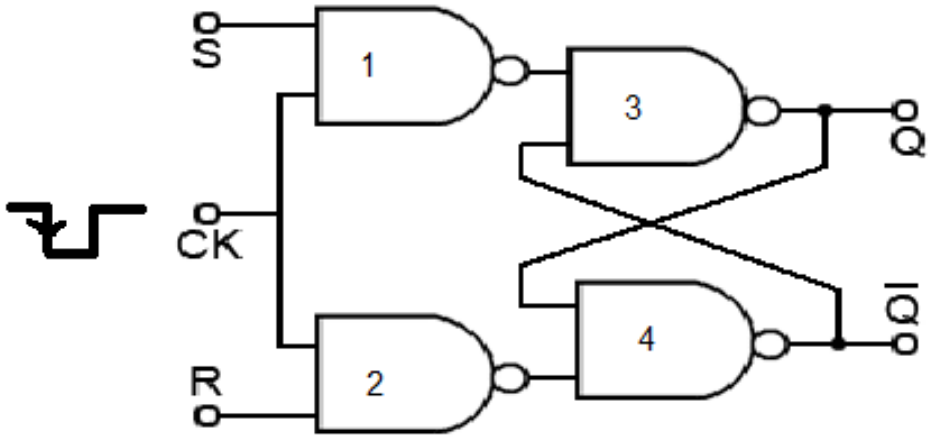


Truth Table:

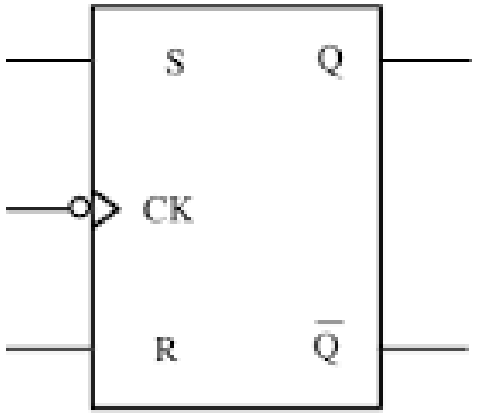
CLK	S	R	Qn+1	Status
↑	0	0	Qn	No change
↑	0	1	0	Reset
↑	1	0	1	Set
↑	1	1	X	Invalid

Negative Edge Triggered SR FF

SR Flip Flop:



Block Diagram:

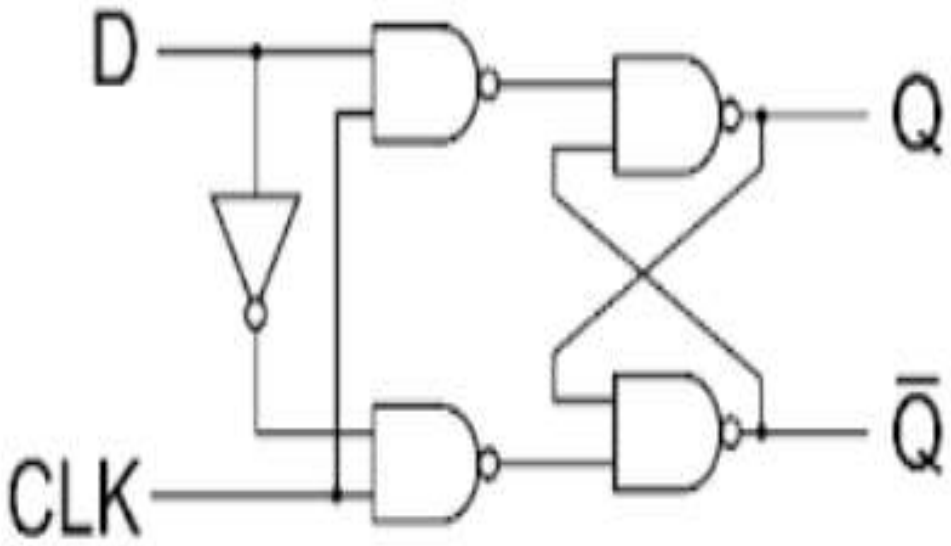


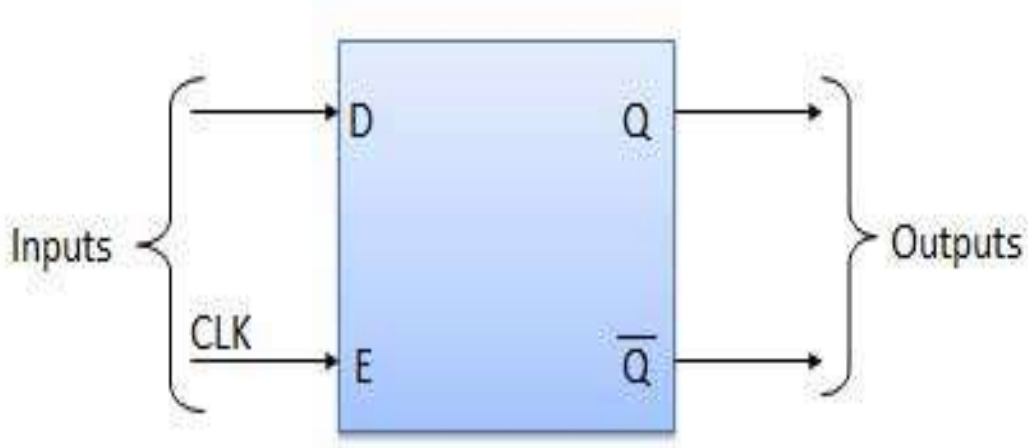
Truth Table:

CLK	S	R	Qn+1	Status
↓	0	0	Qn	No change
↓	0	1	0	Reset
↓	1	0	1	Set
↓	1	1	X	Invalid

D Flip Flop

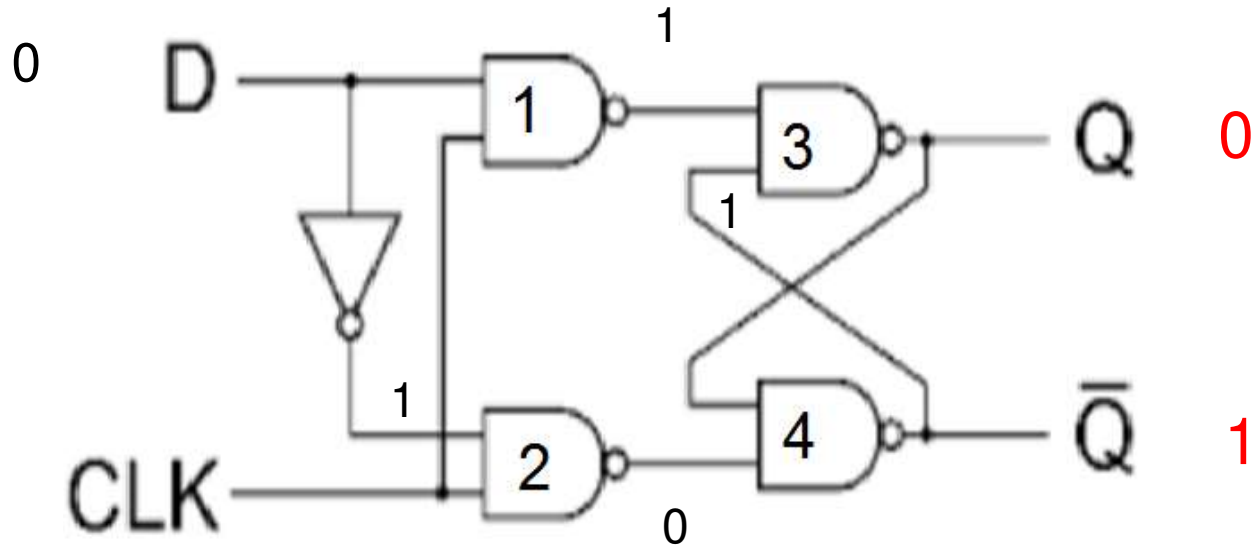
1. D Flip-flops are used as a part of memory storage elements and data processors as well.
2. D flip-flop can be built using NAND gate or with NOR gate. Due to its versatility they are available as IC packages.
3. The major applications of D flip-flop are to introduce delay in timing circuit, as a buffer, sampling data at specific intervals.





D Flip Flop

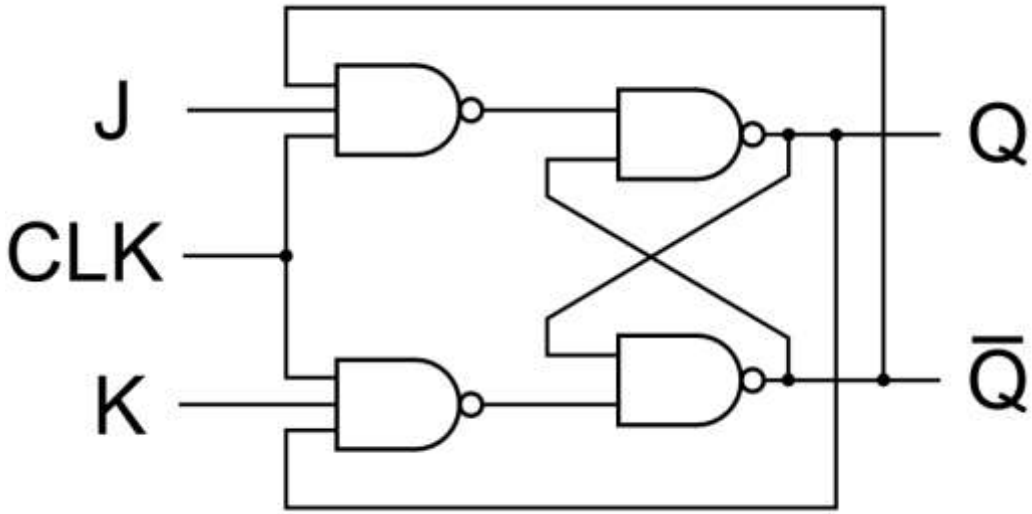
Operation:
D = 0

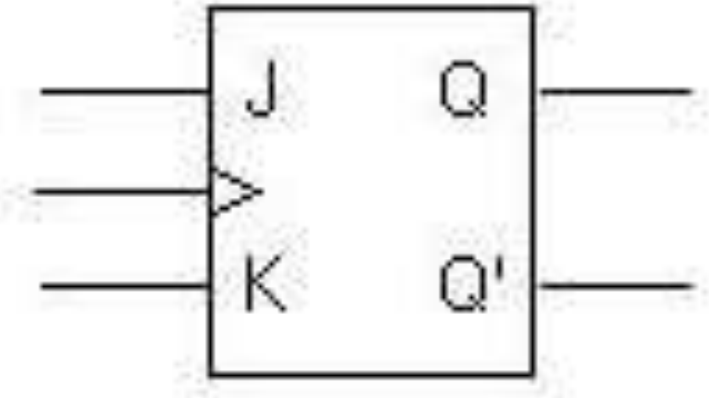


D=0
Therefore NAND gate 1 generate output=1 and NAND gate 2 generate output=0
So, output of NAND gate4 = 1 and NAND gate3 = 1.
Qn+1 = 0 and Qn+1' = 1 **"RESET STATE"**

JK Flip Flop

J K flip-flop is the modified version of SR flip-flop. It operates with only positive clock transitions or negative clock transitions.

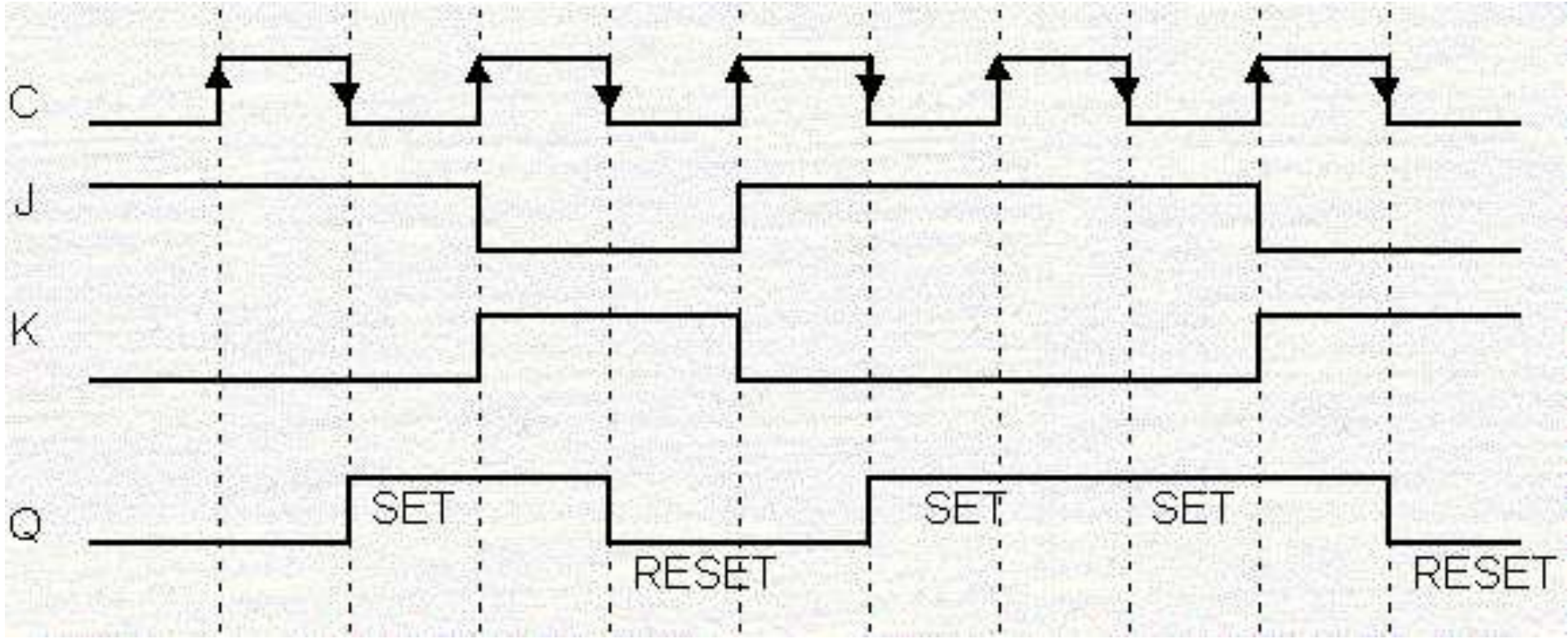




The **JK flip flop** is basically a gated SR flip-flop with the addition of a clock input circuitry that prevents the illegal or invalid output condition that can occur when both inputs S and R are equal to logic level “1”.

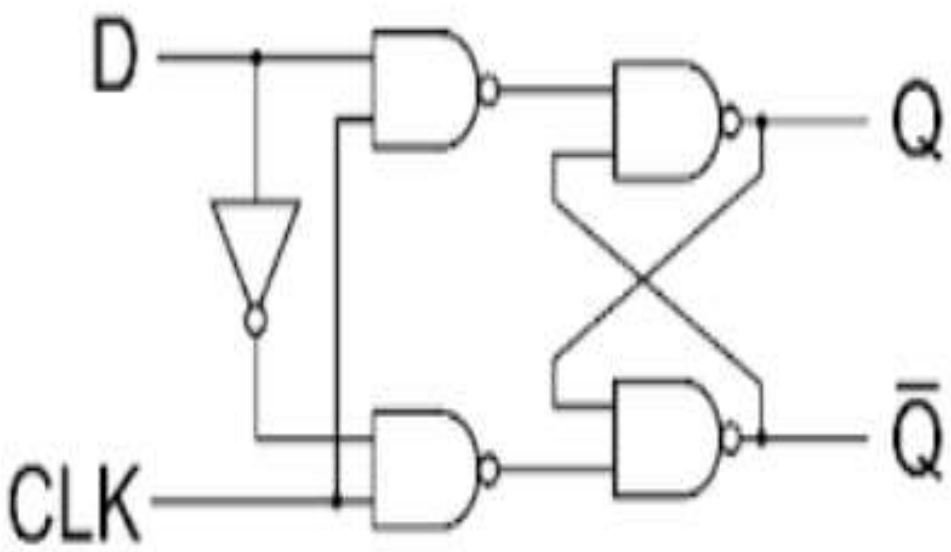
SR Flip Flop

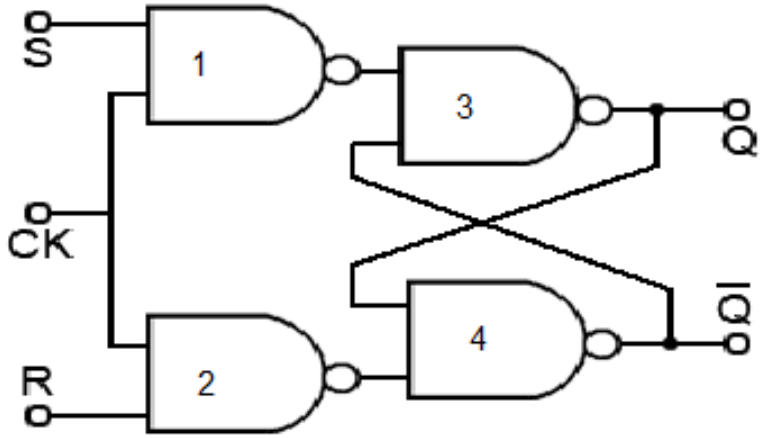
TIMING DIAGRAM:

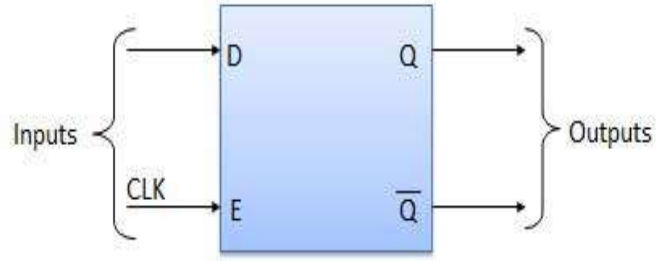


D Flip Flop

1. D Flip-flops are used as a part of memory storage elements and data processors as well.
2. D flip-flop can be built using NAND gate or with NOR gate. Due to its versatility they are available as IC packages.
3. The major applications of D flip-flop are to introduce delay in timing circuit, as a buffer, sampling data at specific intervals.

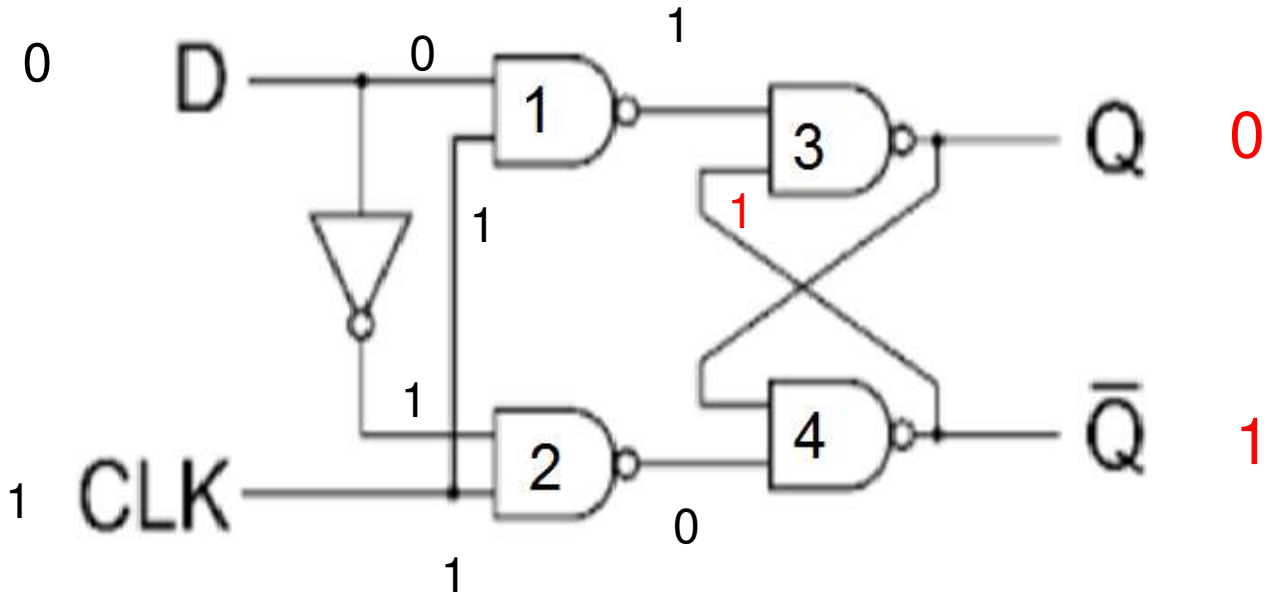






D Flip Flop

Operation:
D = 0

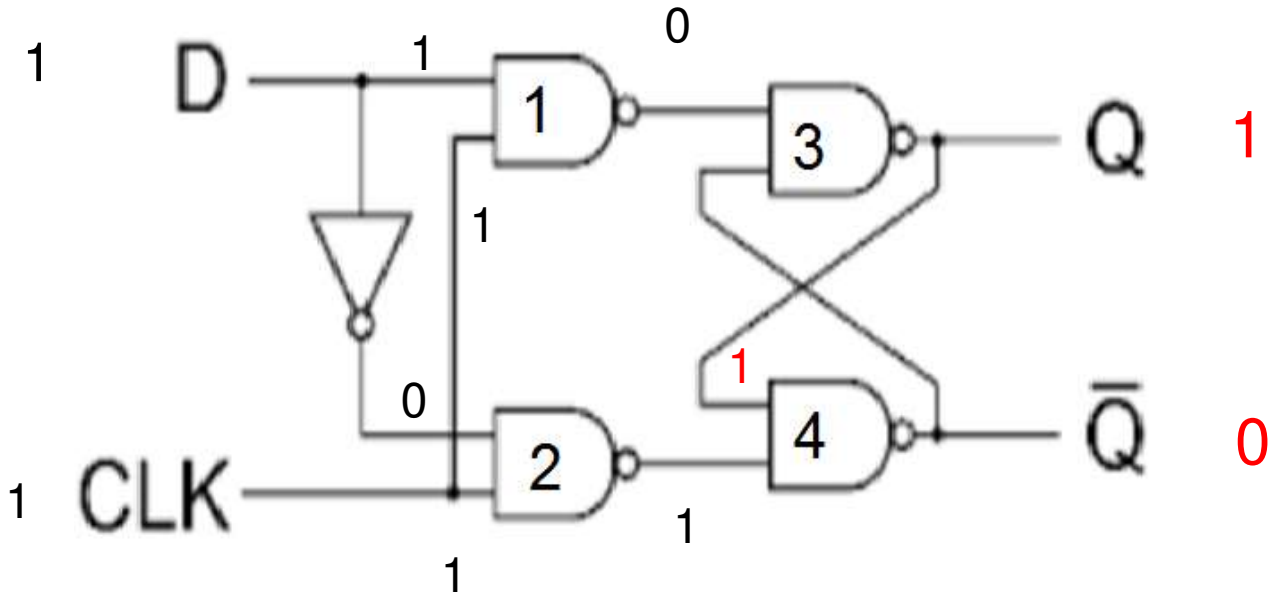


D=0
Therefore NAND gate 1 generate output=1 and NAND gate 2 generate output=0
So, output of NAND gate4 = 1 and NAND gate3 = 0.
Qn+1 = 0 and Qn+1' = 1 **"RESET STATE"**

D Flip Flop

Operation:

D = 1



D=1

Therefore NAND gate 1 generate output=0 and NAND gate 2 generate output=1

So, output of NAND gate3 = 1 and NAND gate4 = 0.

Qn+1 = 0 and Qn+1' = 1 **"SET STATE"**

D Flip Flop

Truth Table:

CLK	D	Qn+1	Status
1	0	0	RESET
1	1	1	SET

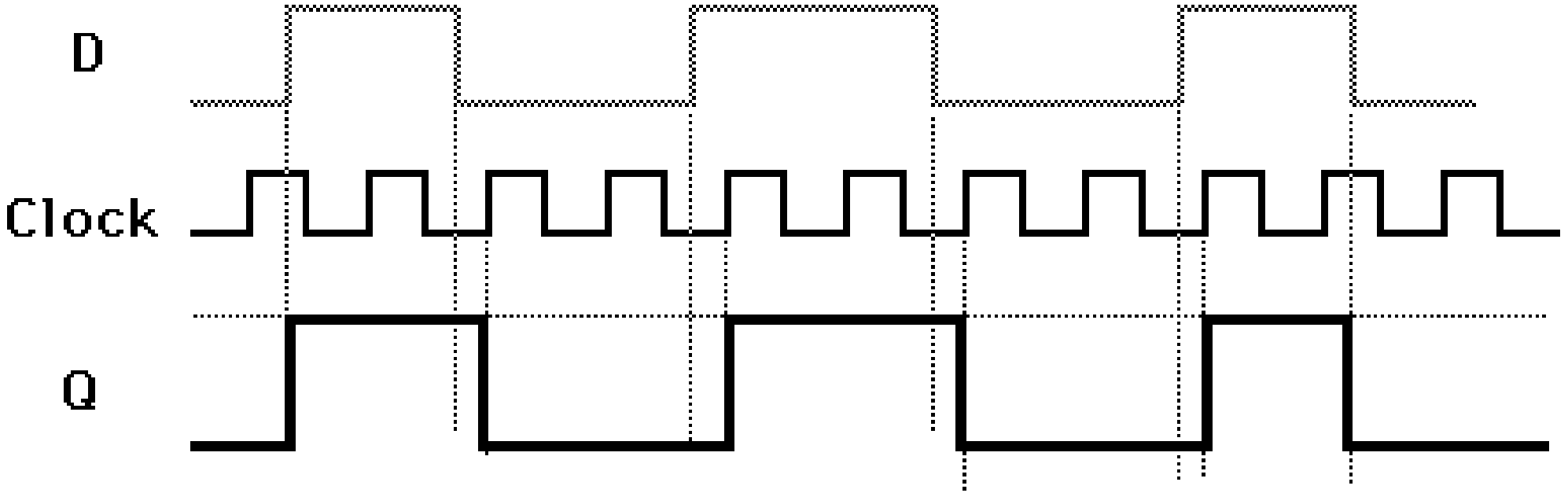
- **Output will follow the D input**.
-However output follows input after some propagation delay therefore it is called as **Delay FF.**

Excitation Table:

Present State	Next State	D
0	0	0
0	1	1
1	0	0
1	1	1

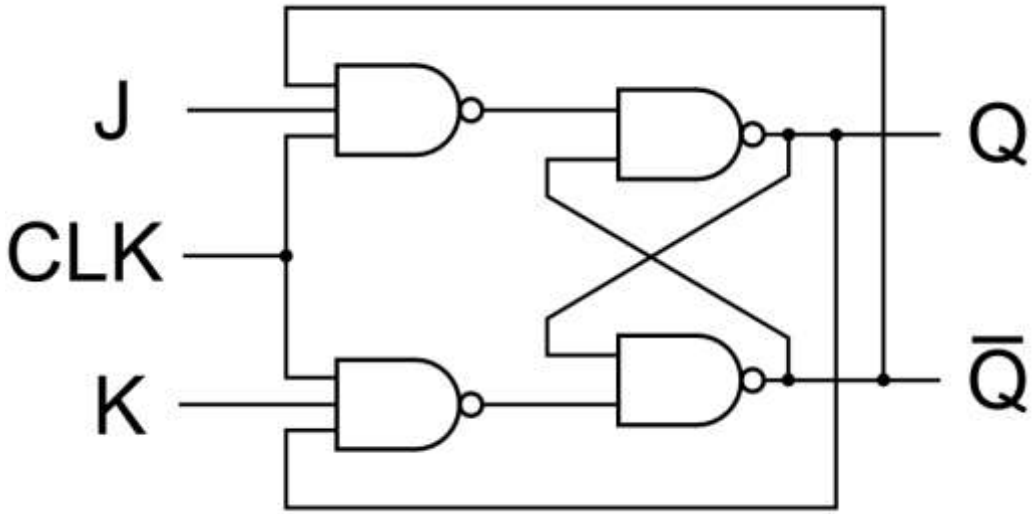
D Flip Flop

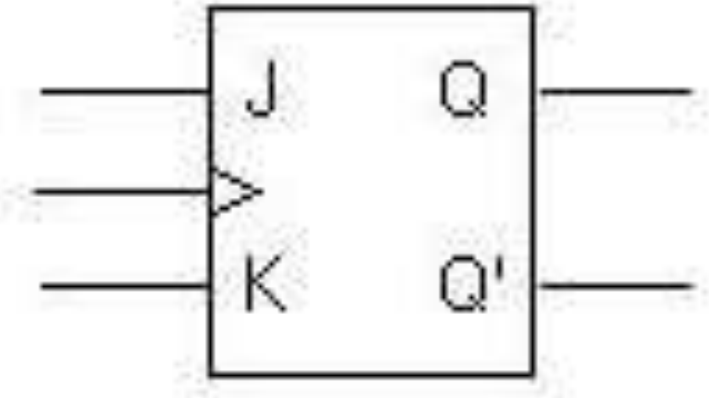
Timing Diagram:



JK Flip Flop

J K flip-flop is the modified version of SR flip-flop. It operates with only positive clock transitions or negative clock transitions.

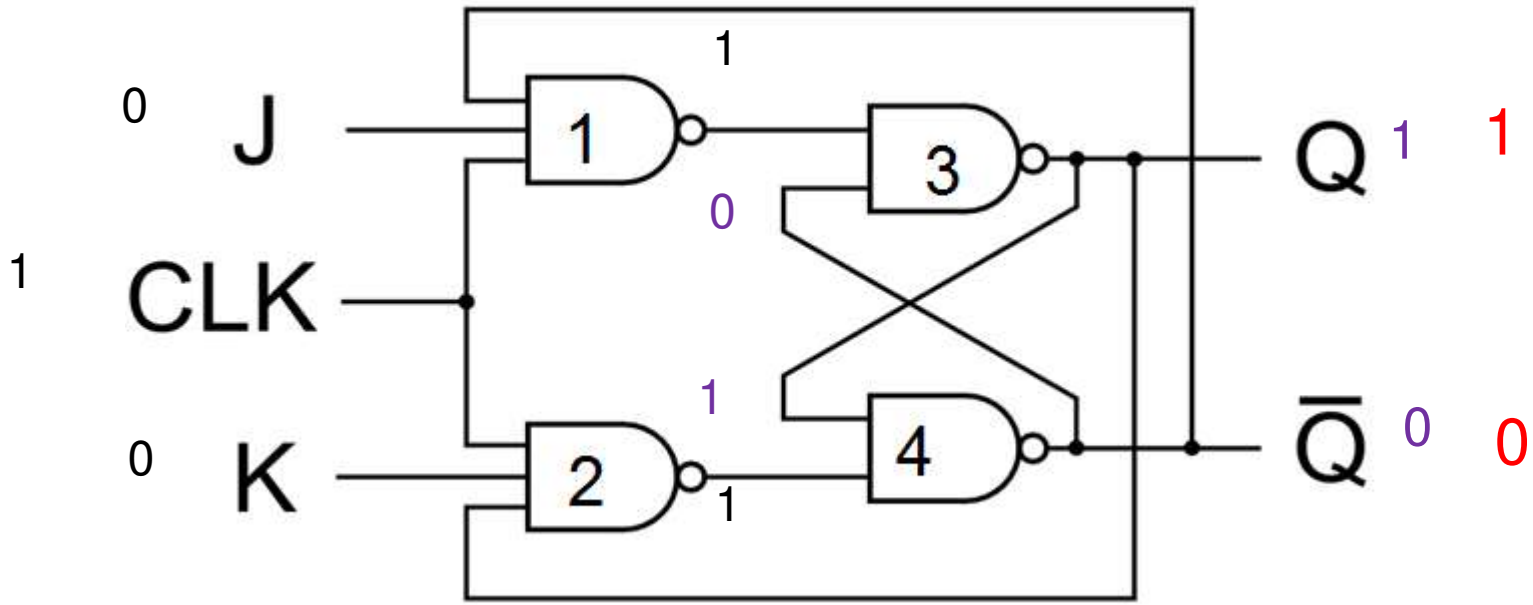




The **JK flip flop** is basically a gated SR flip-flop with the addition of a clock input circuitry that prevents the illegal or invalid output condition that can occur when both inputs S and R are equal to logic level “1”.

JK Flip Flop

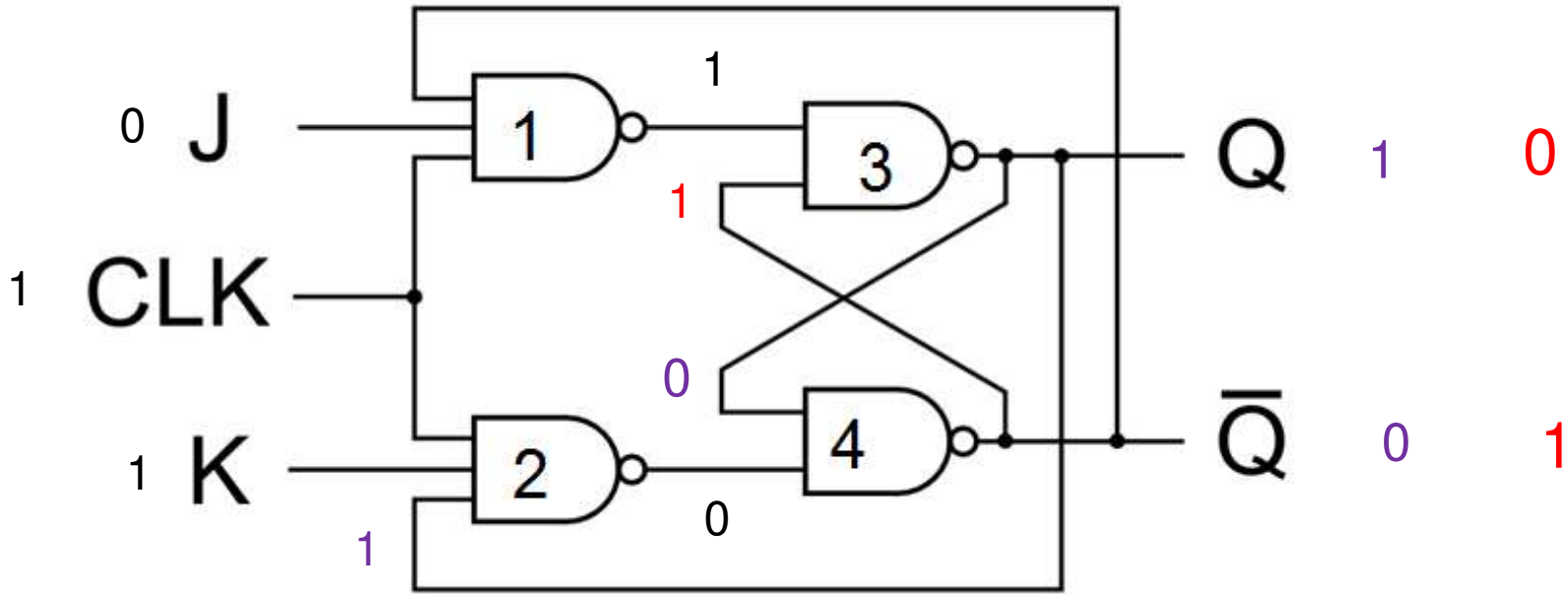
Operation:
J=0 K=0 CLK=1



J=K=0
NAND gate 1 & NAND gate 2 output =1.
Consider, Qn=0 & Qn'=1
Therefore, Qn+1 =0 & Qn+1' =1
Present state and next state remain same therefore NO CHAGE n outputs.

JK Flip Flop

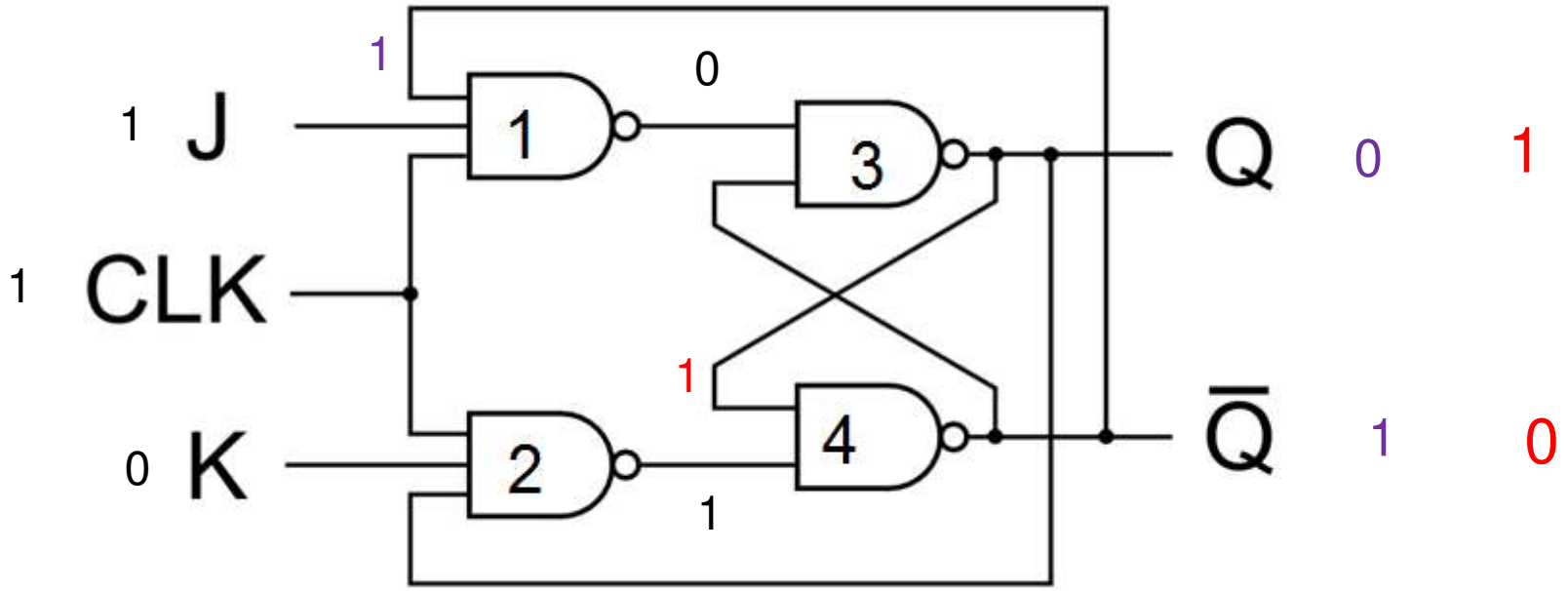
Operation:
J=0 K=1 CLK=1



J=0 K=1
NAND gate 1 OUTPUT= 1 & NAND gate 2 output =1.
Consider, Qn=0 & Qn'=1
Therefore, Qn+1 =0 & Qn+1' =1
When J=0 & K=1 then FF is in RESET (Q=0 & Q'=1) state.

JK Flip Flop

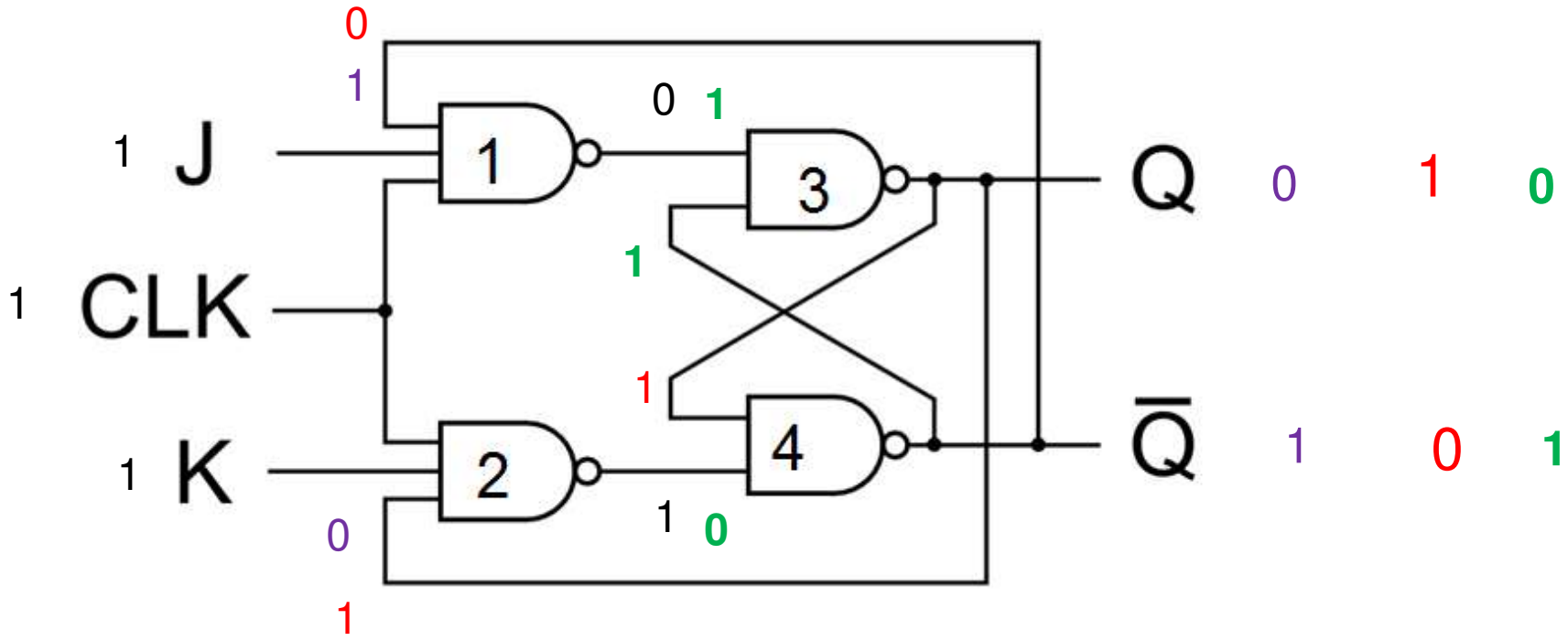
Operation:
J=1 K=0 CLK=1



J=1 K=0
Consider, Qn=0 & Qn'=1
NAND gate 1 OUTPUT= 0 & NAND gate 2 output =1.
Therefore, Qn+1 =1 & Qn+1' =0
When J=1 & K=0 then FF is in SET (Q=1 & Q'=0) state.

JK Flip Flop

Operation:
J=1 K=1 CLK=1



J=1 K=1
Consider, Qn=0 & Qn'=1
NAND gate 1 OUTPUT= 0 & NAND gate 2 output =1.
Therefore, Qn+1 =1 & Qn+1' =0; again this output provided to NAND gate 1 & 2.
Then Qn+1 =0 & Qn+1' =1
When J=1 & K=1 then FF is in Toggle state.

JK Flip Flop

Truth Table:

CLK	J	K	Qn+1	Qn+1’	Status
1	0	0	Qn	Qn’	No change
1	0	1	0	1	RESET
1	1	0	1	0	SET
1	1	1	Qn’	Qn	Toggle

J	K	Qn	Qn’	Qn+1	Qn+1’	State
		Present state		Next state		
0	0	0	1	0	1	NC
0	0	1	0	1	0	
0	1	**0**	**1**	0	1	RESET
0	1	**1**	**0**	0	1	
1	0	0	1	1	0	SET
1	0	1	0	1	0	
1	1	0	1	1	0	TOGGLE
1	1	1	0	0	1	

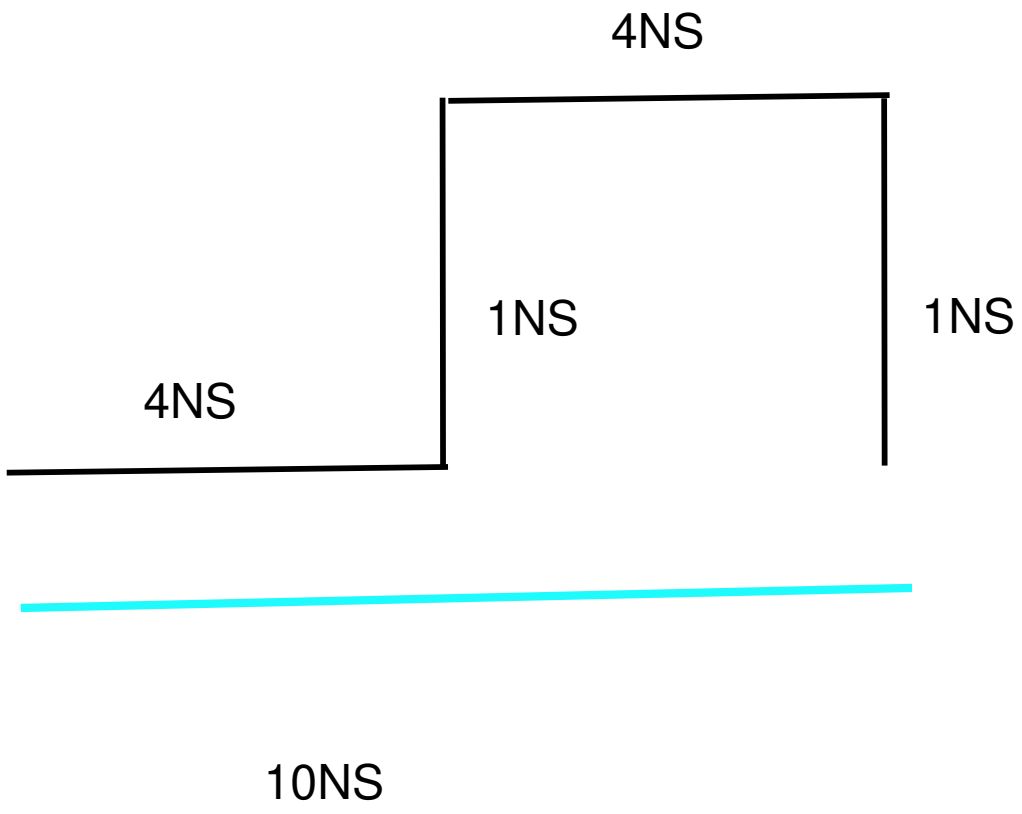
JK Flip Flop

Truth Table:

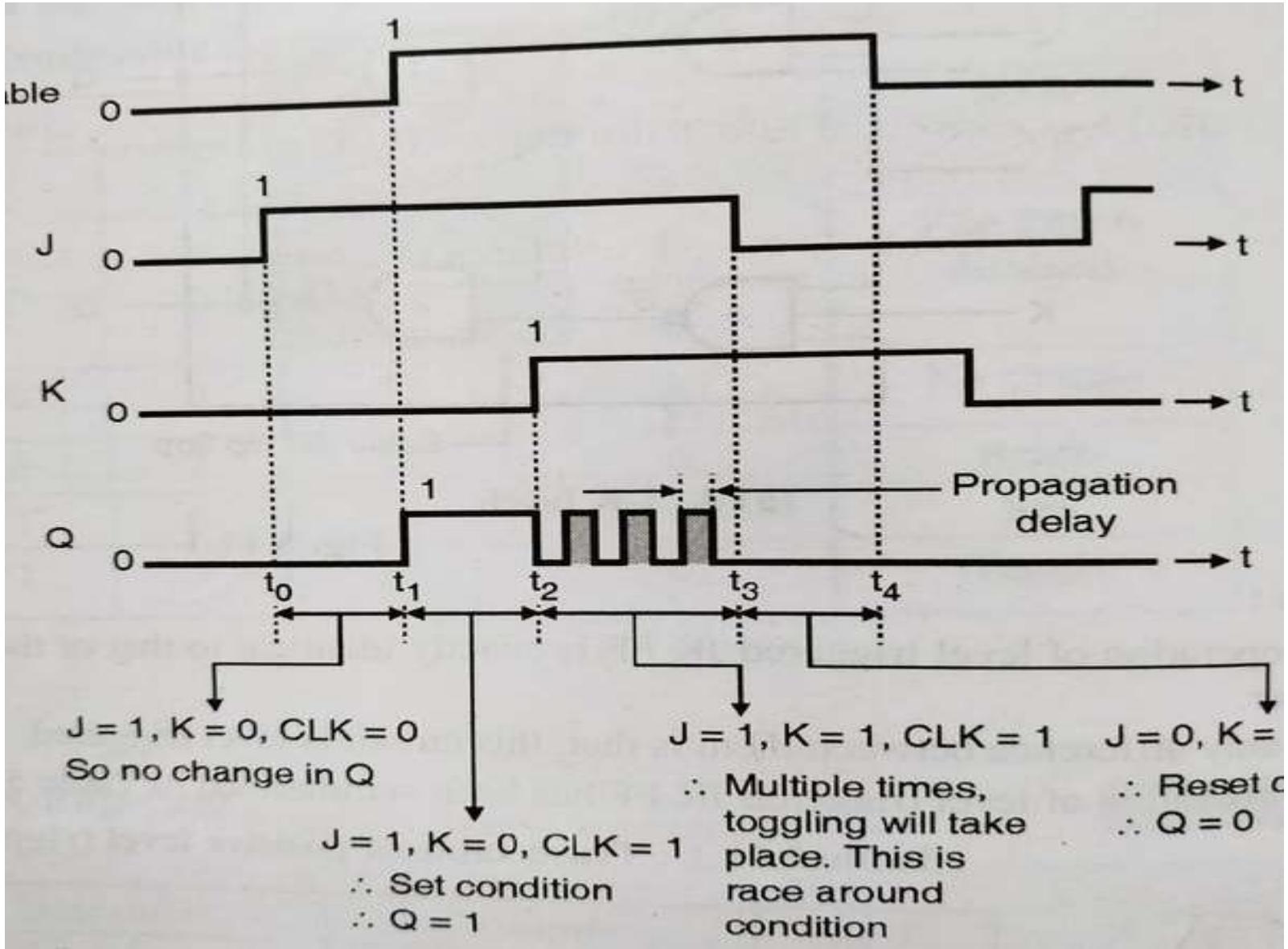
CLK	J	K	Qn+1	Qn+1’	Status
1	0	0	Qn	Qn’	No change
1	0	1	0	1	RESET
1	1	0	1	0	SET
1	1	1	Qn’	Qn	Toggle

Present State	Next State	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

JK Flip Flop



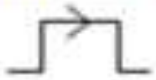
JK Flip Flop Race Around Condition



JK Flip Flop Race Around Condition

How to avoid race around condition in JK FF?
By using,

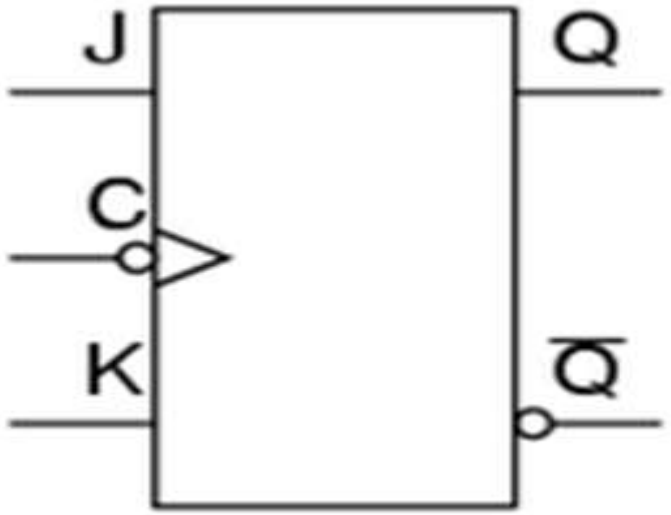
1. **Edge triggered FF**
2. **Master Slave JK FF**

Latch	Flip-flop
It refers to non-clocked flip-flops, because these flip-lops "latch on" to a "1" or a "0" immediately upon receiving the input pulse called SET or RESET.	It samples the inputs only at a clock event.
Latches are level sensitive. 	Flip-flops are edge sensitive. 
Latches are not dependent on the clock signal for their operation, which means a latch is a sequential device that checks all its inputs continuously and changes its outputs accordingly at any time independent of clock signal.	Flip-flop depends on clock signal and continuously checks its inputs and corresponding changes its output only at times determined by clock signal.
It works based on enable function input.	It works on the basis of clock pulse.

Edge Triggered JK Flip Flop

Negative Edge Triggered JK FF

Block Diagram:

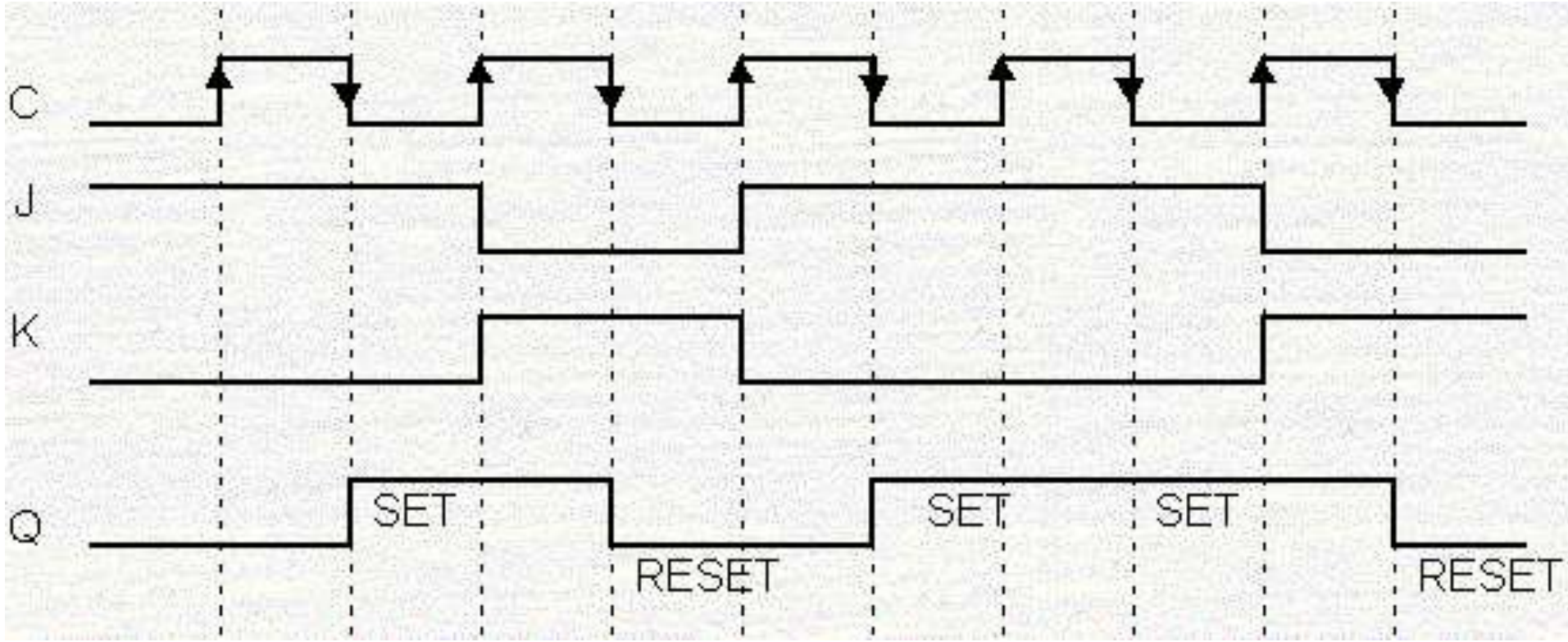


Truth Table:

CLK	J	K	Qn+1	Qn+1’
0	X	X	Qn	Qn’
1	X	X	Qn	Qn’
↑	X	X	Qn	Qn’
↓	0	0	Qn	Qn’
↓	0	1	0	1
↓	1	0	1	0
↓	1	1	Qn’	Qn

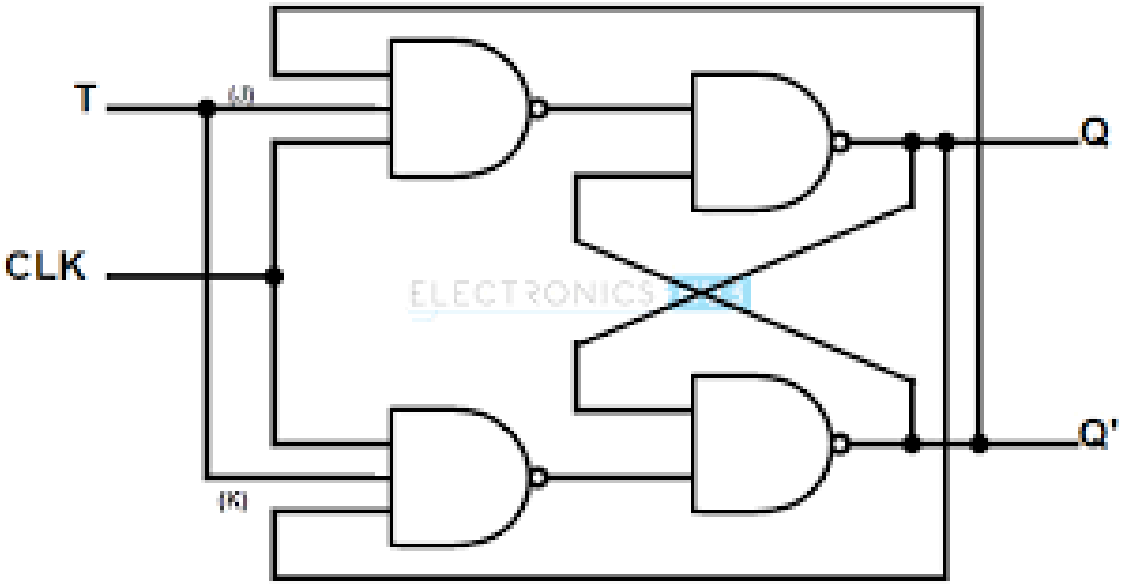
JK Flip Flop

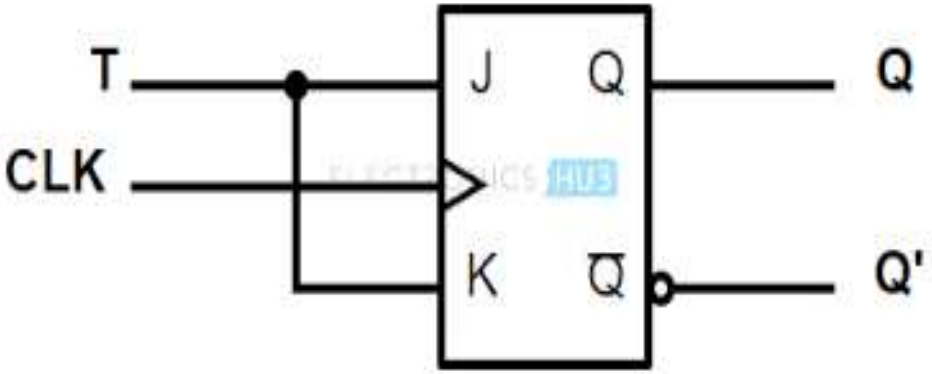
TIMING DIAGRAM:



T Flip Flop

1. T flip – flop is also known as “Toggle Flip – flop”.
2. Toggling means ‘Changing the next state output to complement of the present state output’.
3. The T flip – flop is a single input device and hence by connecting J and K inputs together and giving them with single input called T we can convert a JK flip – flop into T flip – flop.

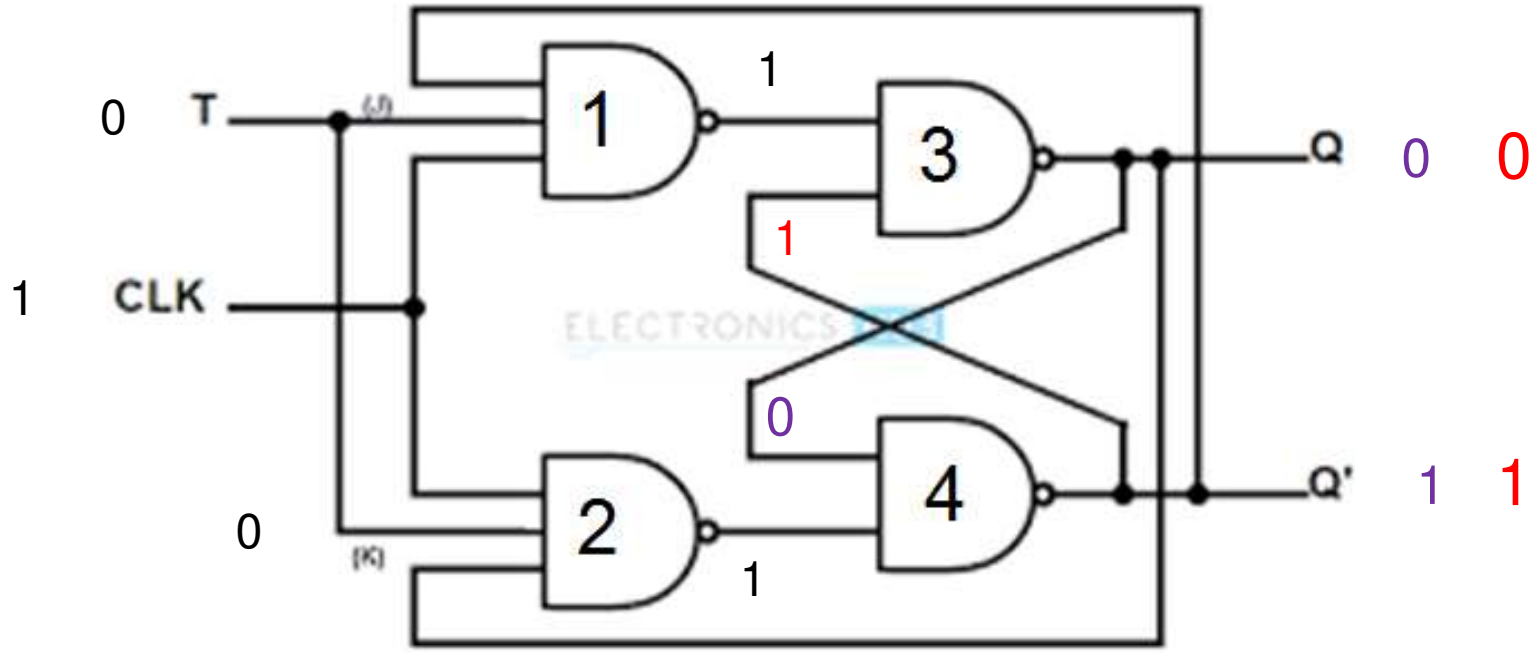




T Flip Flop

Operation:

T = 0



T=0

Therefore NAND gate 1 generate output=1 and NAND gate 2 generate output=1

Consider, Qn =0 and Qn'=1

So, output of NAND gate4 = 1 and NAND gate3 = 0.

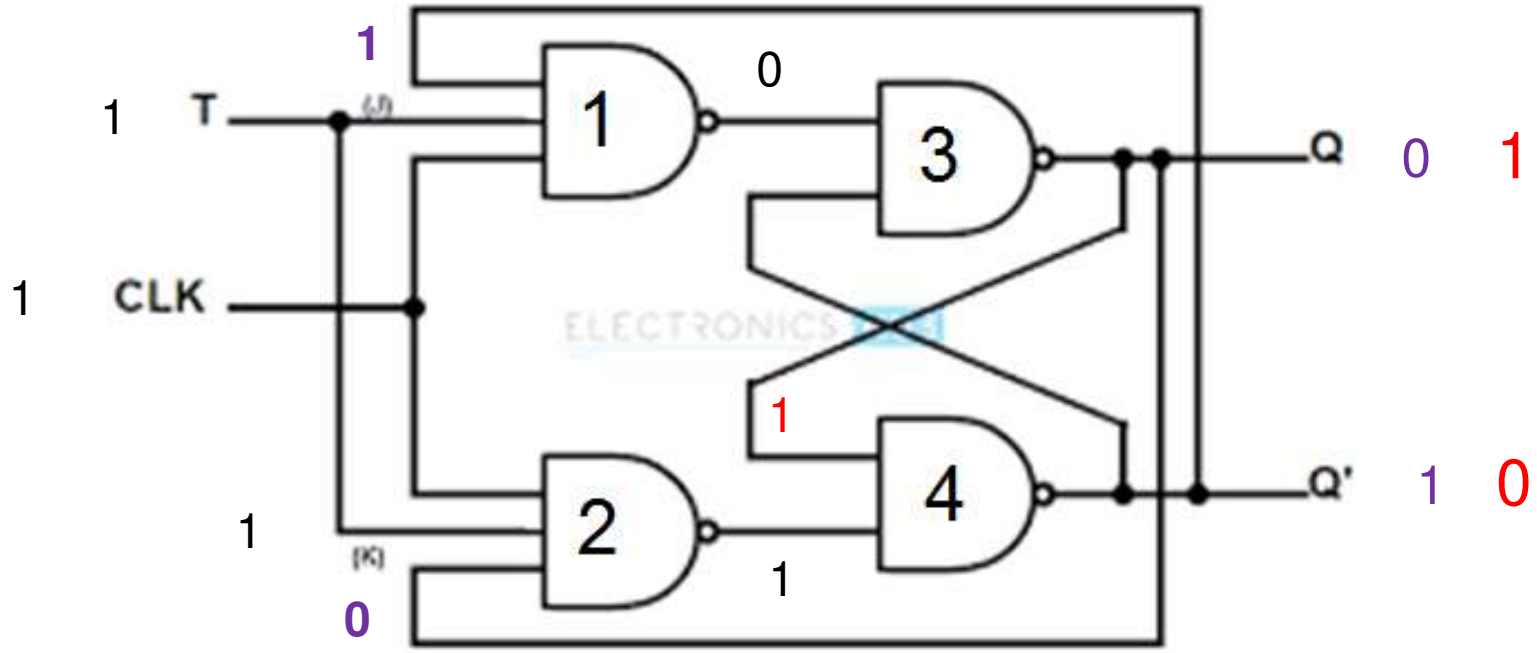
Qn+1 = 0 and Qn+1' = 1

Present State and Next state remains same therefore "NO CHANGE" in outputs.

T Flip Flop

Operation:

T = 1



T=1

Consider, Qn =0 and Qn'=1

Therefore NAND gate 1 generate output=0 and NAND gate 2 generate output=1

So, output of NAND gate3 = 1 and NAND gate4 = 0.

Qn+1 = 1 and Qn+1' = 0

Present State and Next state are complement to each other. "TOGGLE"

T Flip Flop

Truth Table:

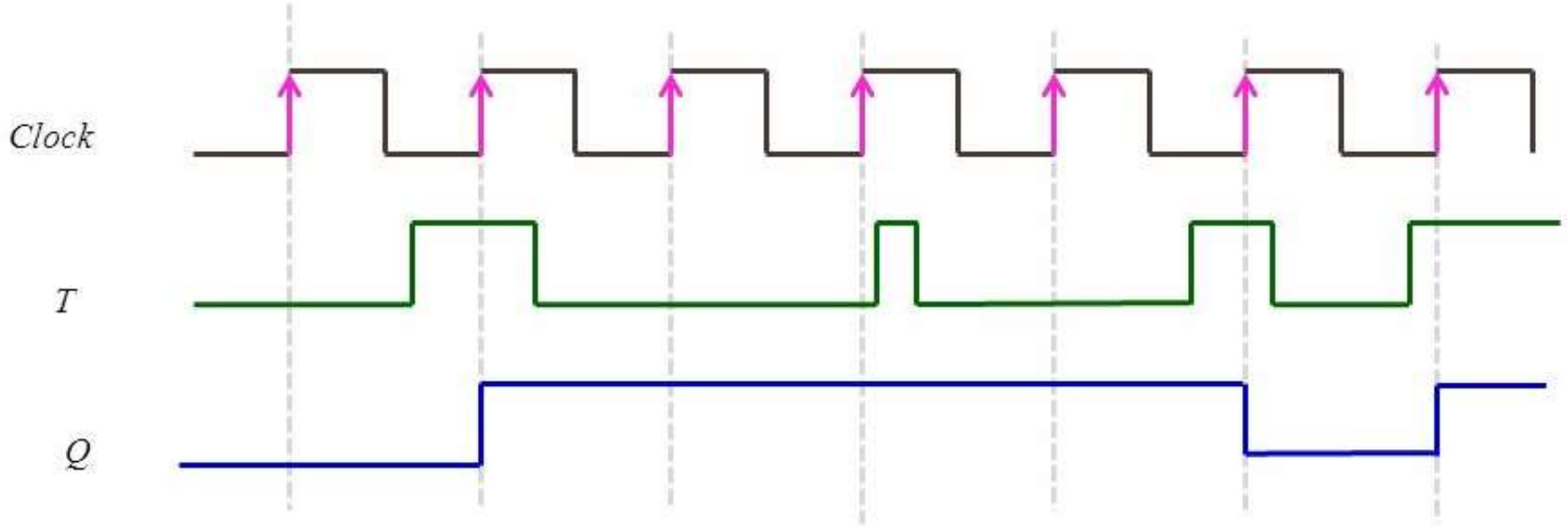
CLK	T	Qn+1	Qn+1’	Status
1/0	X	Qn	Qn’	No change
↓	x	Qn	Qn’	No change
↑	**0**	**Qn**	**Qn’**	**No change**
↑	**1**	**Qn’**	**Qn**	**Toggle**

Excitation Table:

Present State	Next State	T
0	0	0
0	1	1
1	0	1
1	1	0

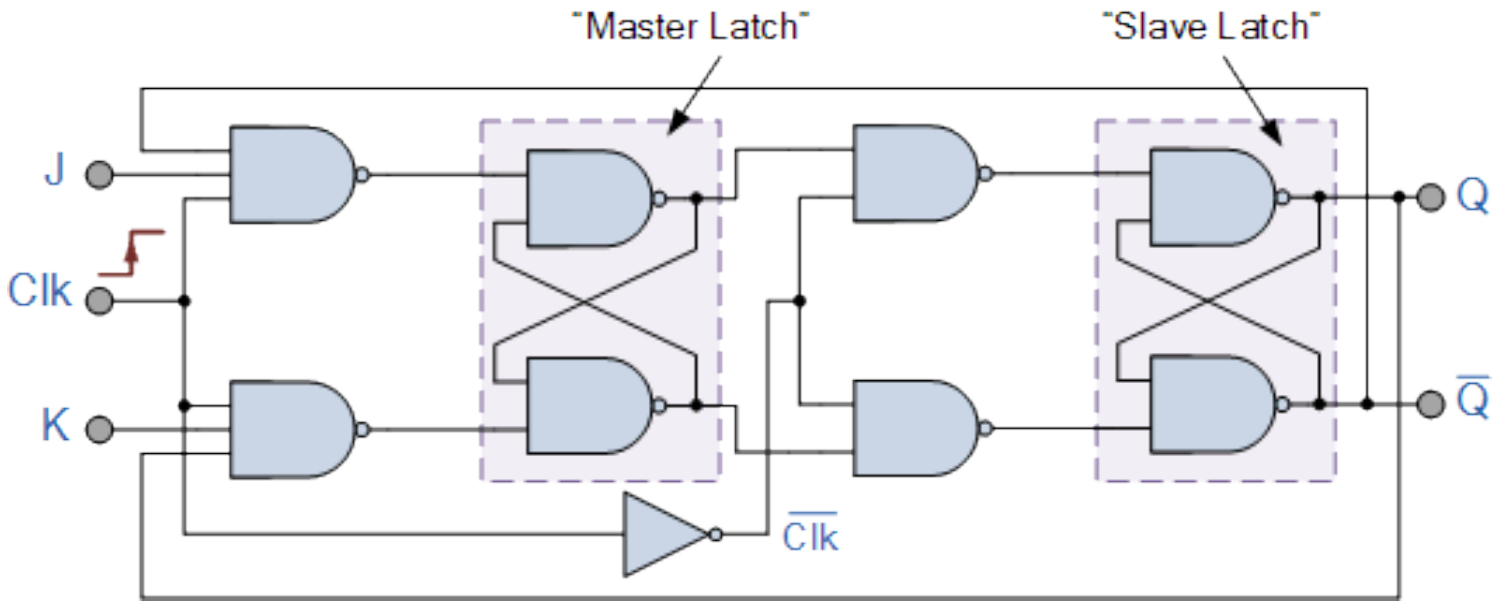
T Flip Flop

Timing Diagram:



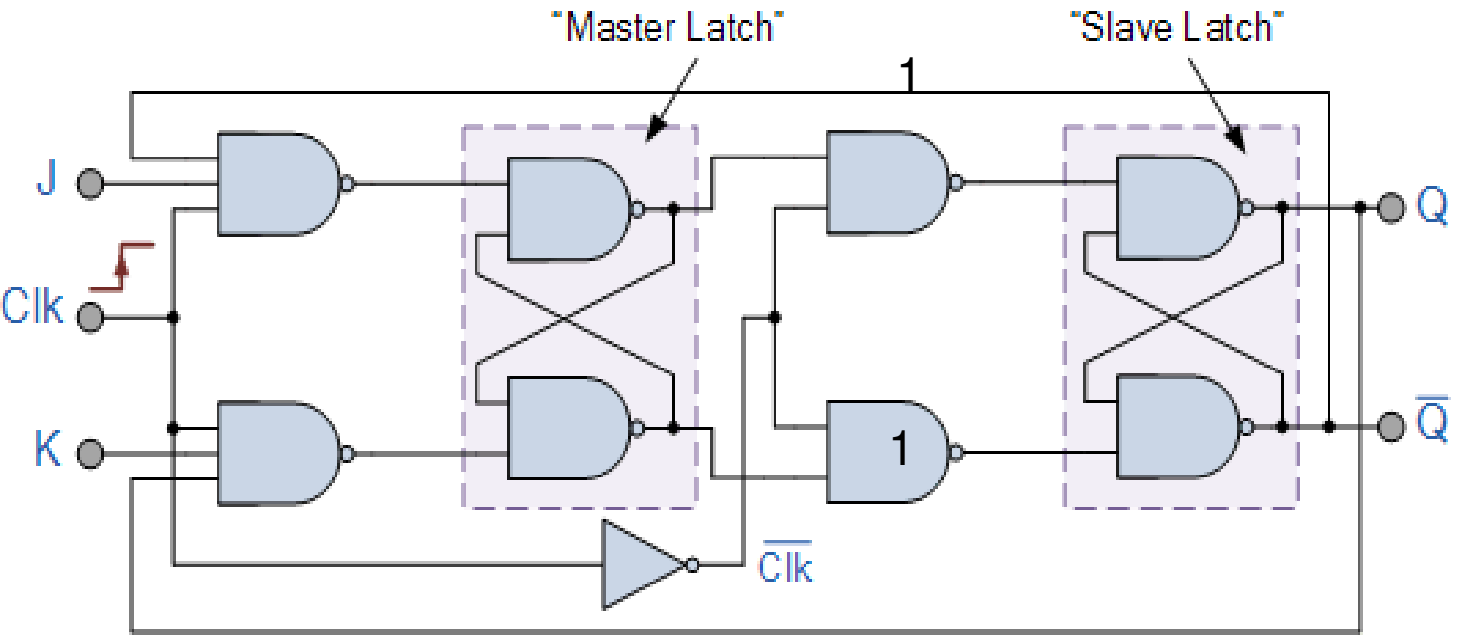
Master Slave JK Flip Flop

1. The master-slave flip-flop eliminates all the timing problems by using two SR flip-flops connected together in a series configuration.
2. One flip-flop acts as the “Master” circuit, which triggers on the leading edge of the clock pulse while the other acts as the “Slave” circuit, which triggers on the falling edge of the clock pulse.
3. This results in the two sections, the master section and the slave section being enabled during opposite half-cycles of the clock signal.



JK Flip Flop

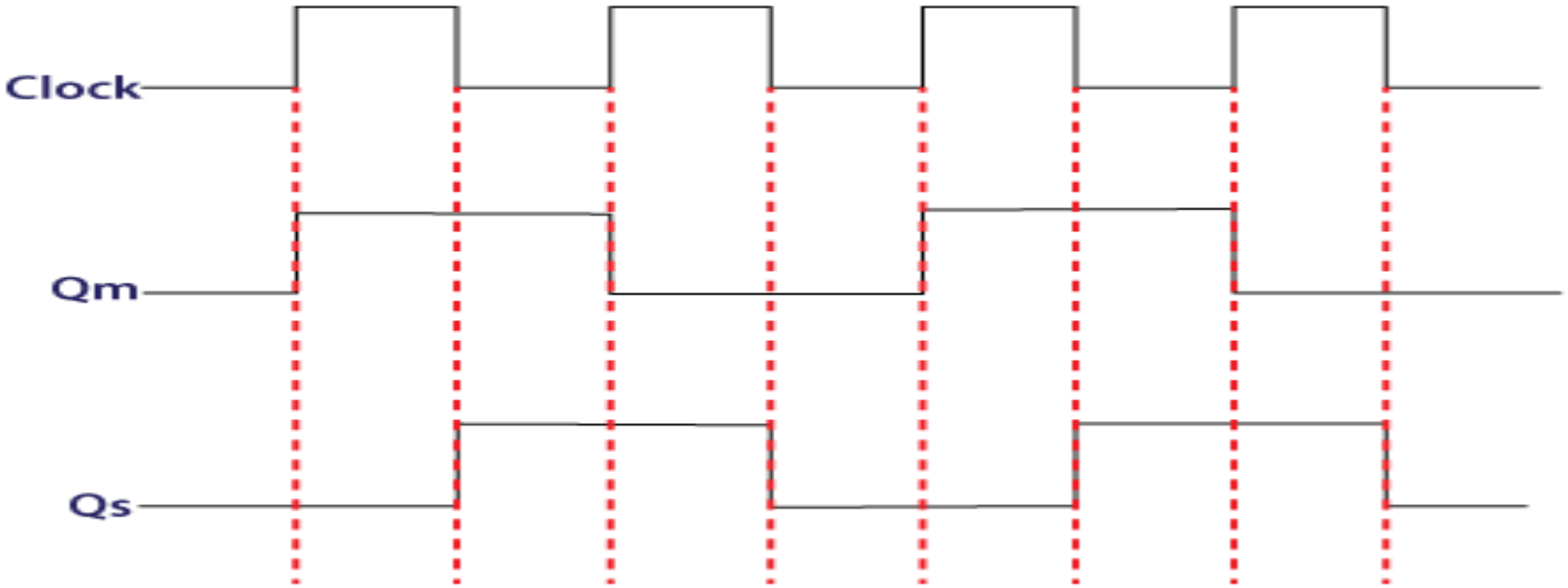
Operation:



CLK	J	K	Qn+1	State
⎍	0	0	Qn	NC
⎍	0	1	0	RESET
⎍	1	0	1	SET
⎍	1	1	Qn’	Toggle

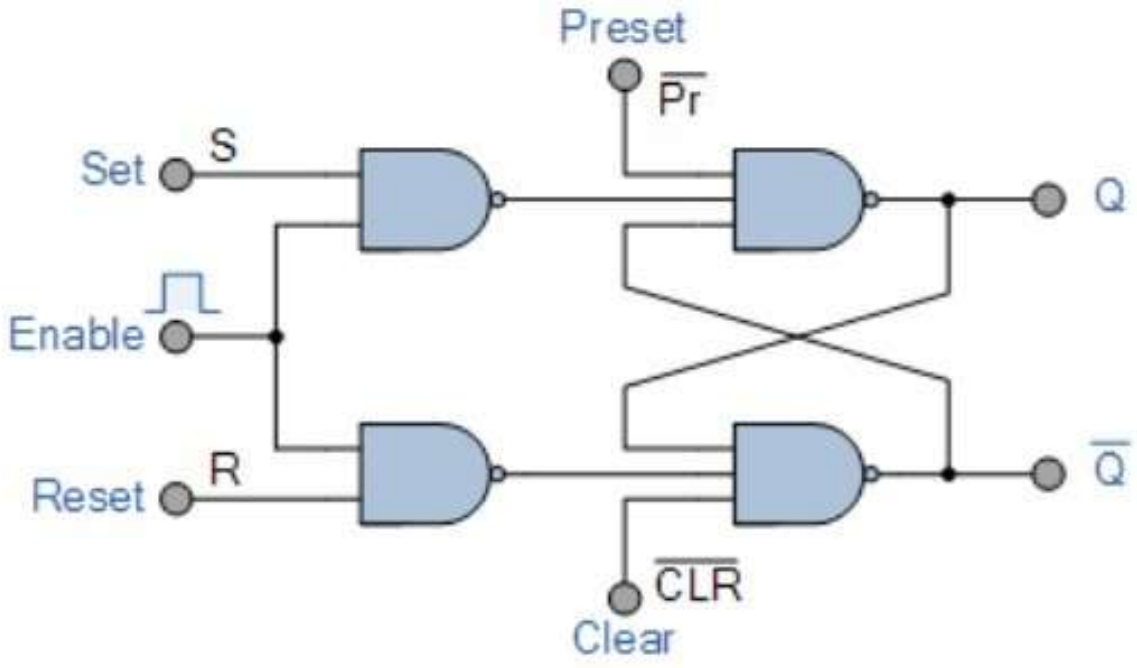
Master Slave JK Flip Flop

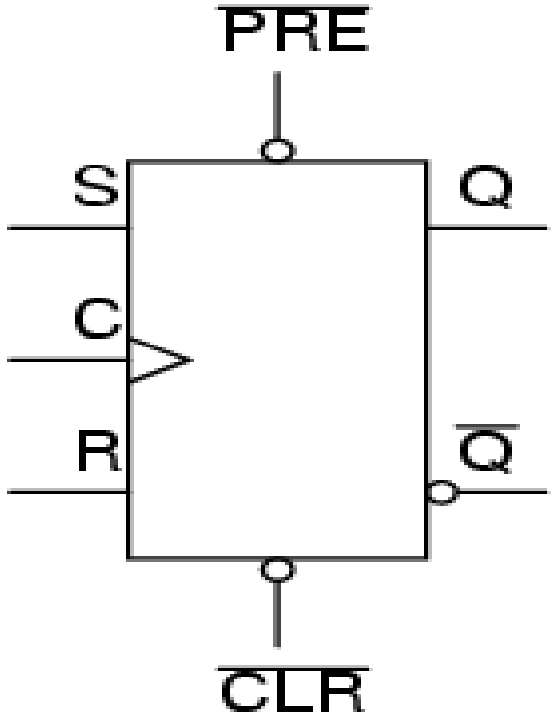
Timing Diagram:



Preset & Clear Input

1. The PRESET and CLEAR inputs of the JK Flip-Flop are asynchronous, which means that they will have an immediate effect on the Q and Q' outputs regardless of the state of the clock and / or the J and K inputs.
2. It is important NOT to simultaneously activate the CLEAR and PRESET inputs.
3. The Flip-Flop have a small bubble in the PRESET or CLEAR inputs which indicate that they are active low.

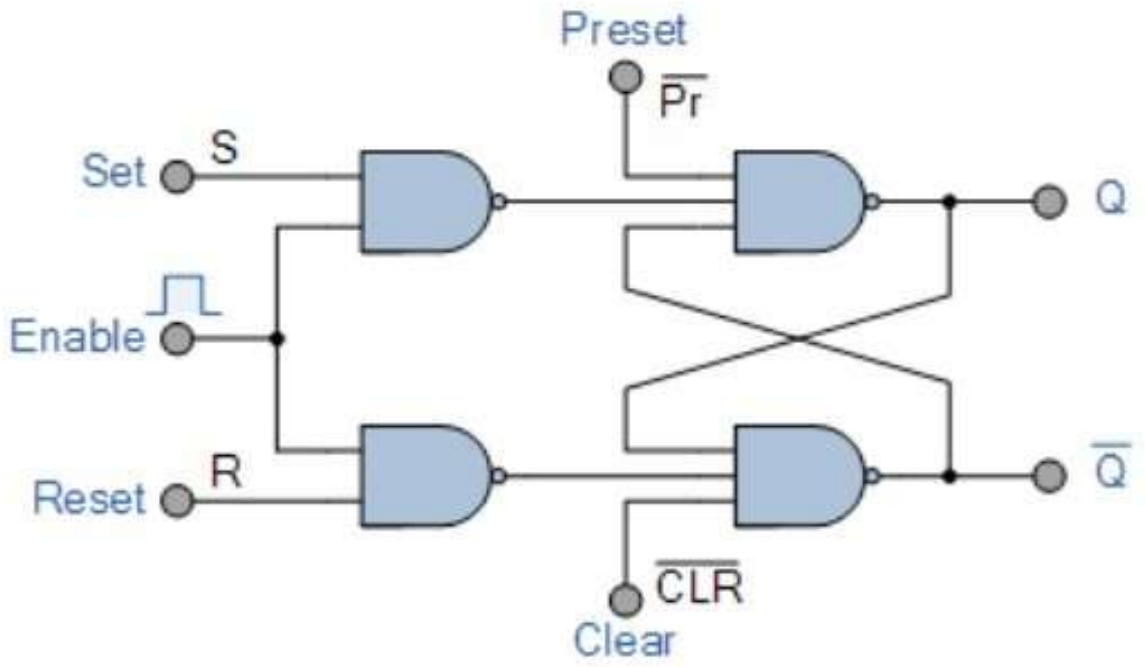




Preset & Clear Input

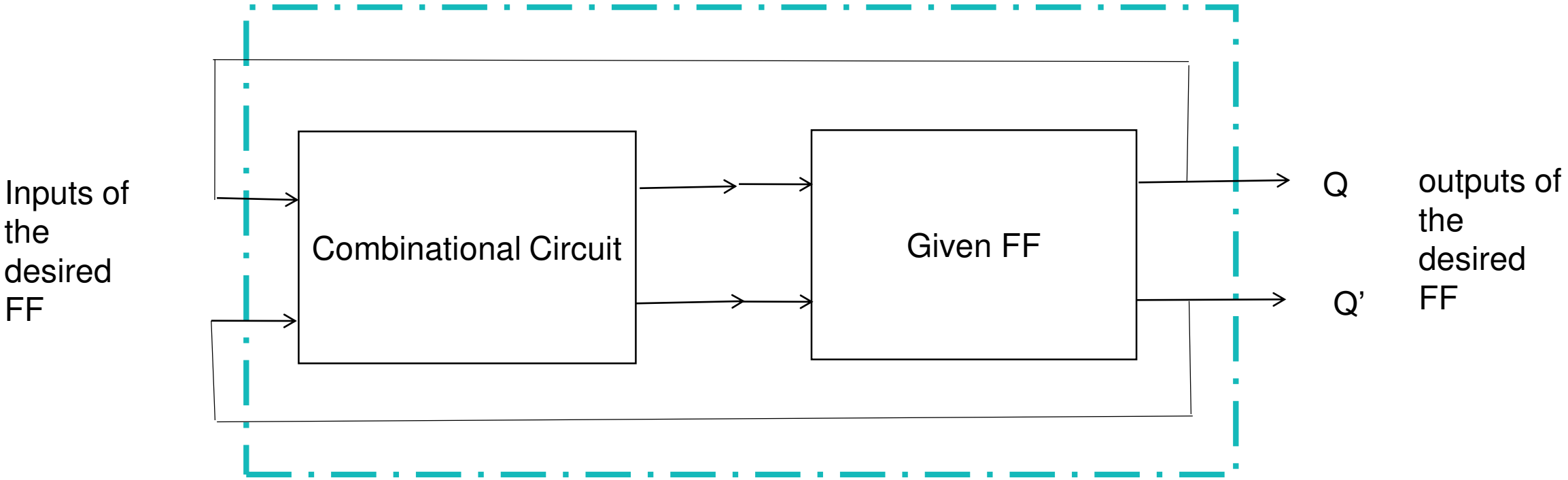
Truth Table:

CLK	PR	CLR	Output (Q)	Operation
1	1	1	Qn+1	Normal FF
x	0	1	1	FF SET
x	1	0	0	FF RESET



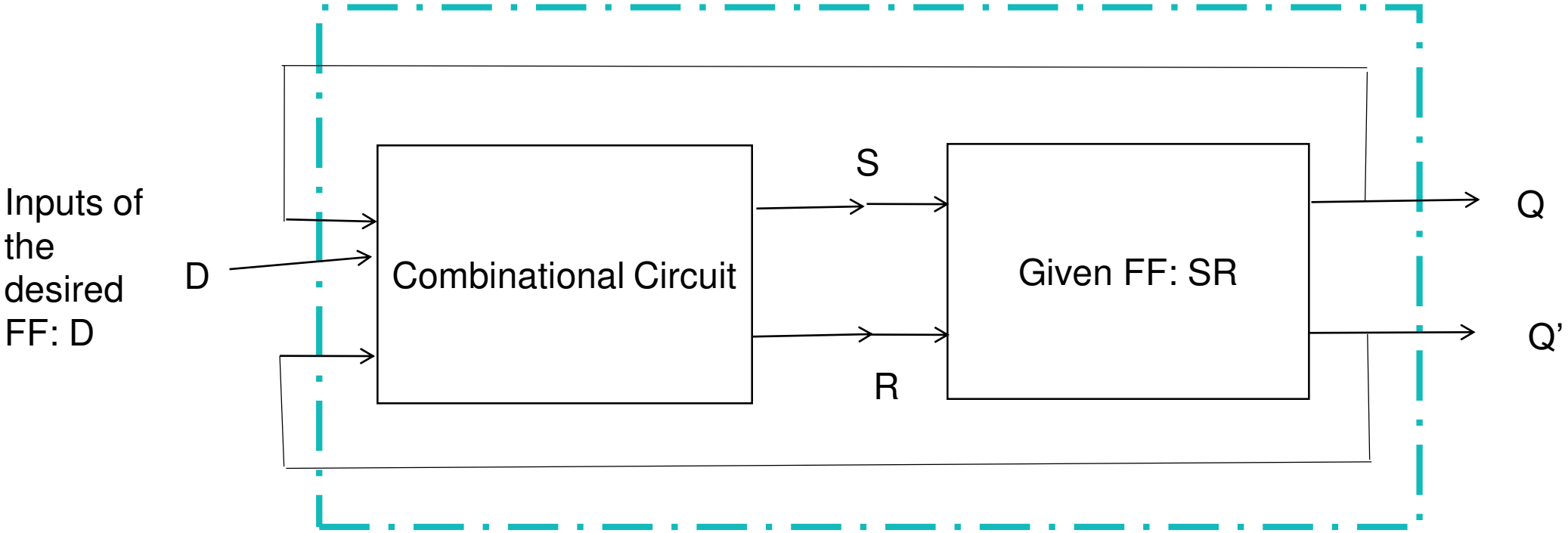
Flip Flop Conversion

1. a combinational circuit has to be designed first. If a JK Flip Flop is required, the inputs are given to the combinational circuit and the output of the combinational circuit is connected to the inputs of the actual flip flop. Thus, the output of the actual flip flop is the output of the required flip flop.



Flip Flop Conversion

Example 1:
Convert the following flip-flop: SR to D
-> Given FF SR
Expected FF D



Flip Flop Conversion

Example 1:
Convert the following flip-flop: SR to D
-> Given FF SR
Expected FF D

Excitation Table of SR FF:

P.S.	N.S.	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Excitation Table of D:

P.S.	N.S.	D
0	0	0
0	1	1
1	0	0
1	1	1

Conversion Table:

D	P.S. (Qn)	N.S (Qn+1)	S	R
0	0	0	0	X
1	0	1	1	0
0	1	0	0	1
1	1	1	X	0

Flip Flop Conversion

Kmap:
Kmap Simplification for S & R outputs,

D	P.S. (Qn)	N.S. (Qn+1)	S	R
0	0	0	0	X
1	0	1	1	0
0	1	0	0	1
1	1	1	X	0

Kmap for S

D

:

Qn \ D	0	1
0	0	0
1	1	X

S=D

Kmap for R

D

Qn \ D	0	1
0	X	1
1	0	0

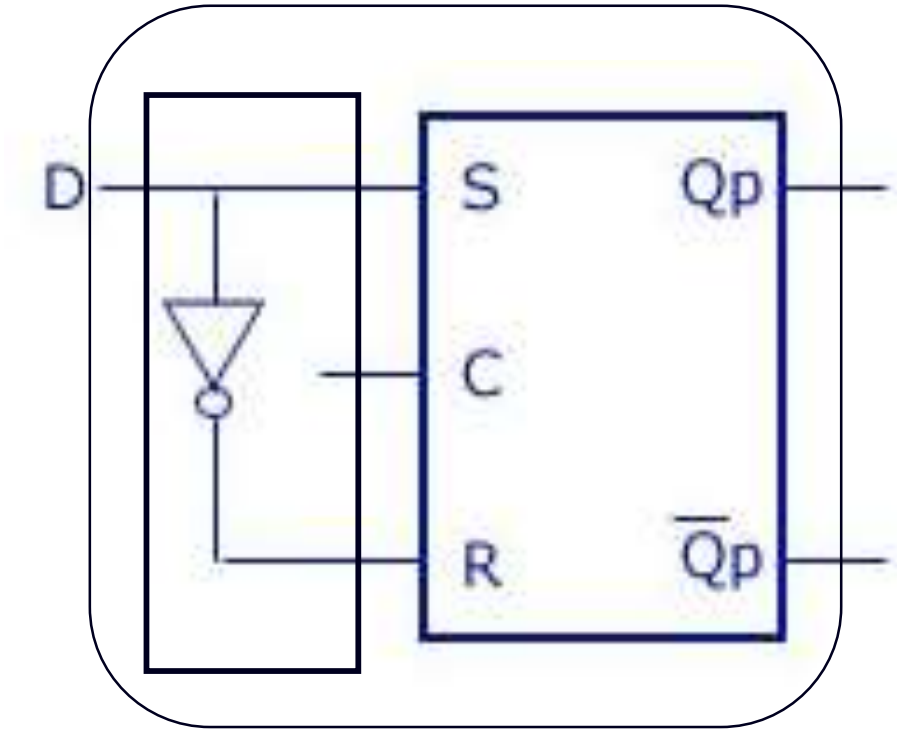
R=D'

Flip Flop Conversion

Logic Diagram,

S= D

R= D’



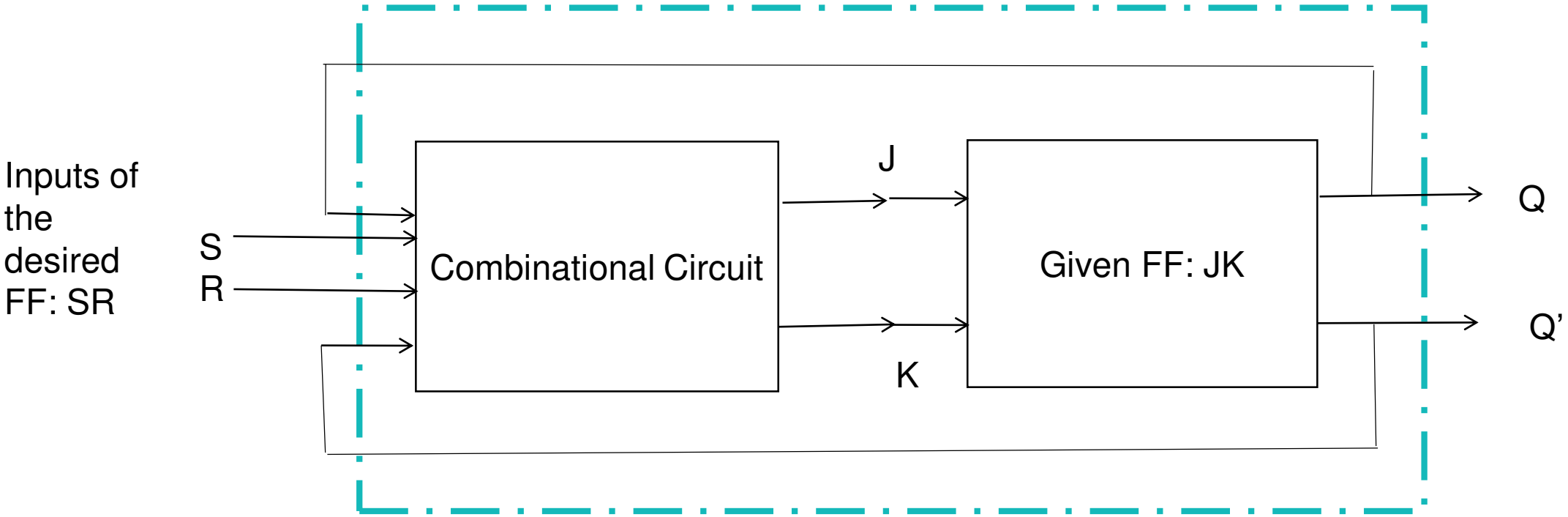
Flip Flop Conversion

Example 2:

Convert the following flip-flop: JK to SR

-> Given FF JK

Expected FF SR



Flip Flop Conversion

Example 1:
Convert the following flip-flop: JK TO SR
-> Given FF JK
Expected FF SR

Excitation Table of SR FF:

P.S.	N.S.	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Excitation Table of JK:

P.S.	N.S.	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Conversion Table:

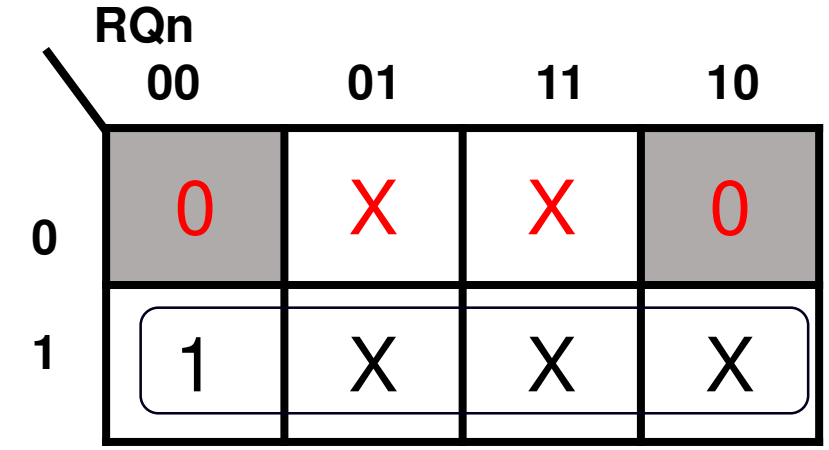
S	R	P.S.(Q)	N.S	J	K
0	X	0	0	0	X
1	0	0	1	1	X
0	1	1	0	X	1
X	0	1	1	X	0

Flip Flop Conversion

Kmap:
Kmap Simplification for J & K outputs,

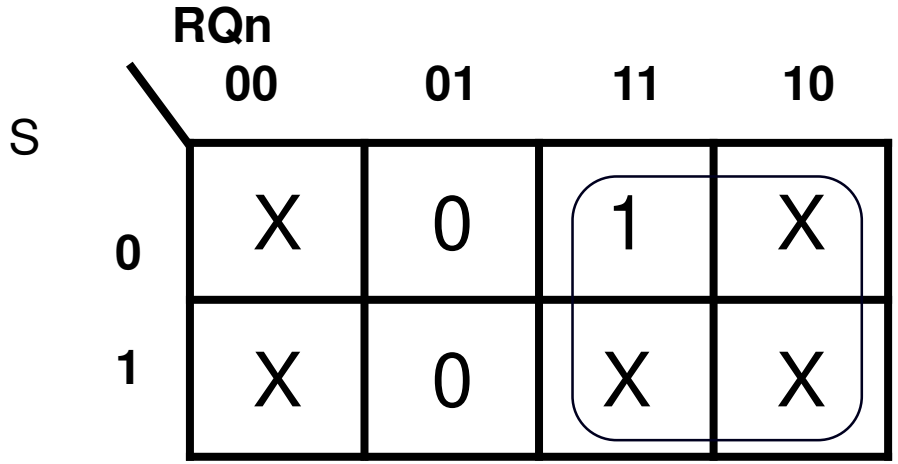
S	R	P.S.	N.S	J	K
0	X	0	0	0	X
1	0	0	1	1	X
0	1	1	0	X	1
X	0	1	1	X	0

Kmap for J



J=S

Kmap for K

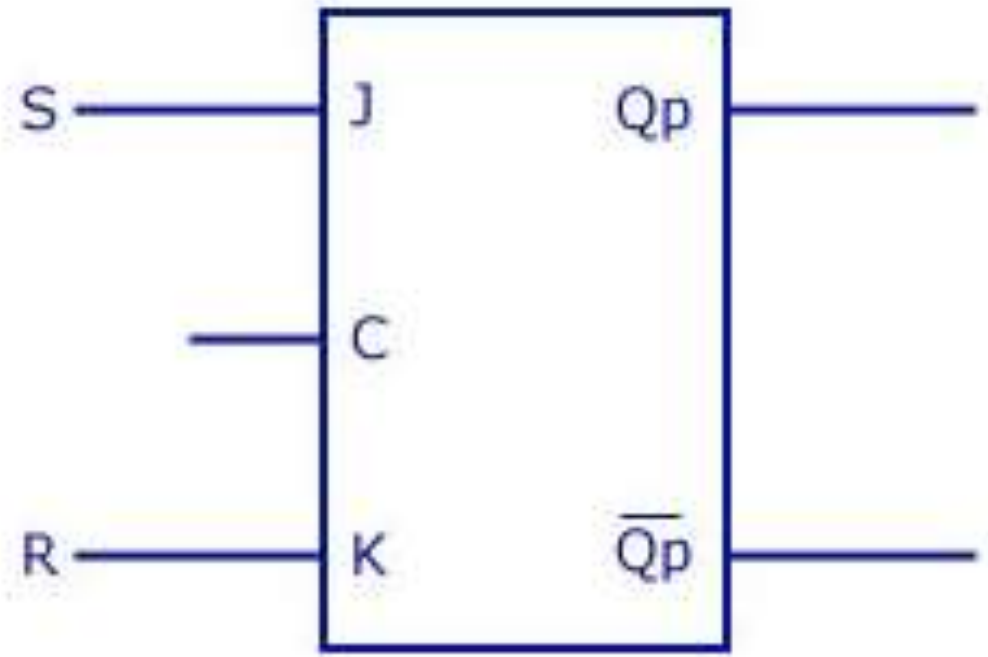


K=R

Flip Flop Conversion

Logic Diagram:

J= S



K= R

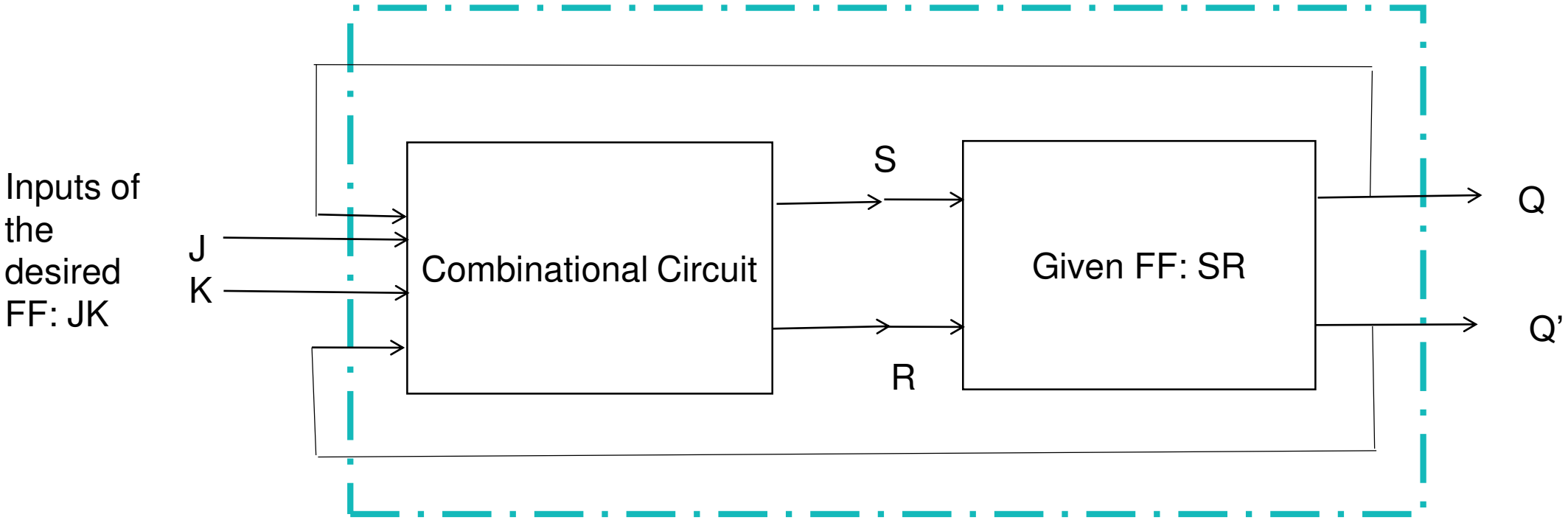
Flip Flop Conversion

Example 3:

Convert the following flip-flop: SR to JK

-> Given FF SR

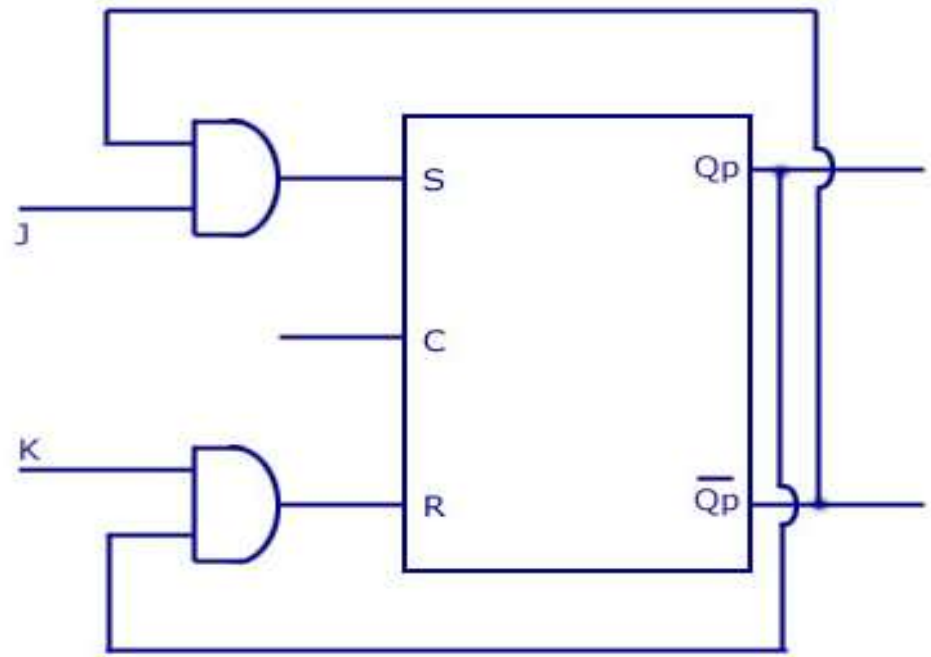
Expected FF JK

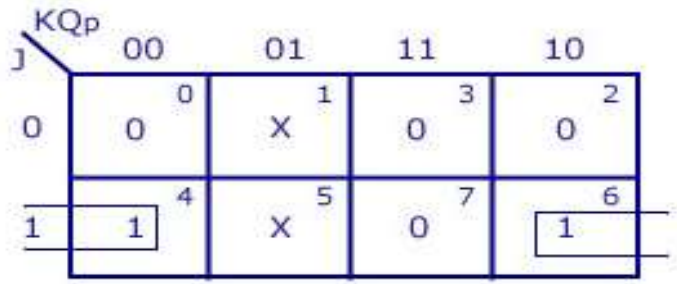


Conversion Table

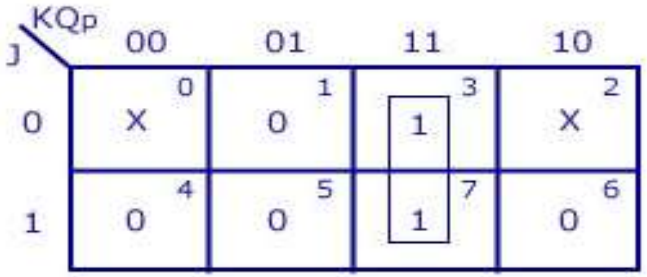
J-K Inputs		Outputs		S-R Inputs	
J	K	Qp	Qp+1	S	R
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	X	0
1	1	0	1	1	0
1	1	1	0	0	1

Logic Diagram





$S = \overline{J}Qp$



$R = KQp$

K-Map

Exsercise

Example 1: Design T FF using D FF

Example 2: Design D FF using T FF

Example 3: Design SR FF using T FF

Example 4: Design JK FF using D FF

:

Agenda

01

Flip-Flop: SR, JK,D,T, Preset and clear, Master Slave JK FF, Truth Tables and Excitation Tables, Conversion of FF

02

Registers: SISO, SIPO, PISO, PIPO, Shift registers, Bidirectional shift register, Universal shift register.

03

Counters: Asynchronous counter, synchronous counter, BCD counter, Ring Counter, Johnson Counter, Modulus of counter (IC 7490)

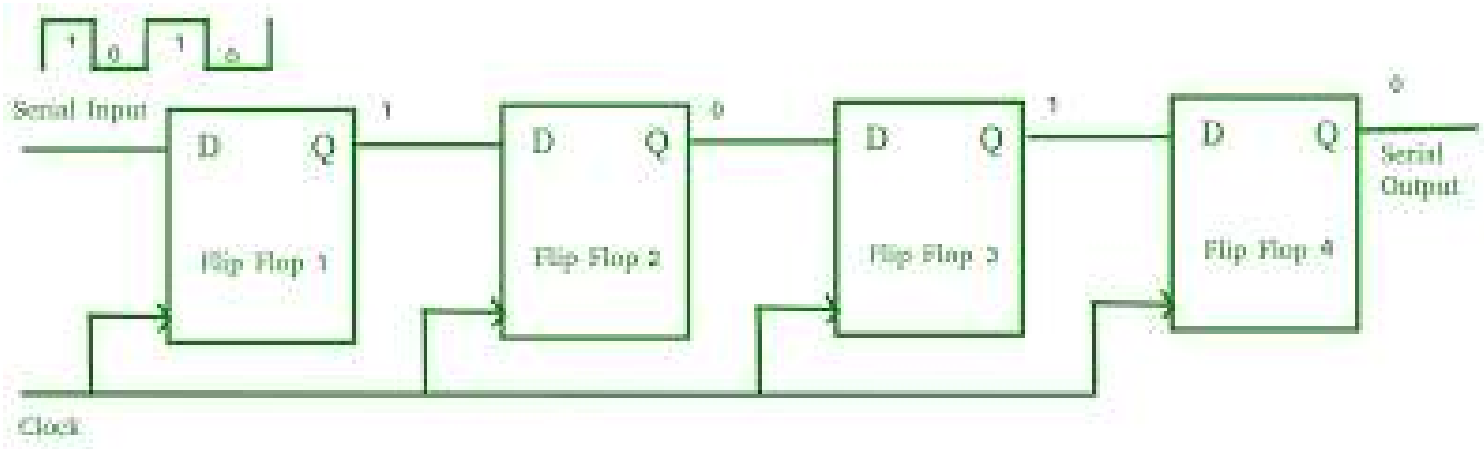
04

Synchronous Sequential Circuit Design: Models- Moore and Mealy, State diagram and state table, Design Procedure, Sequence geneartor and detector.

Registers

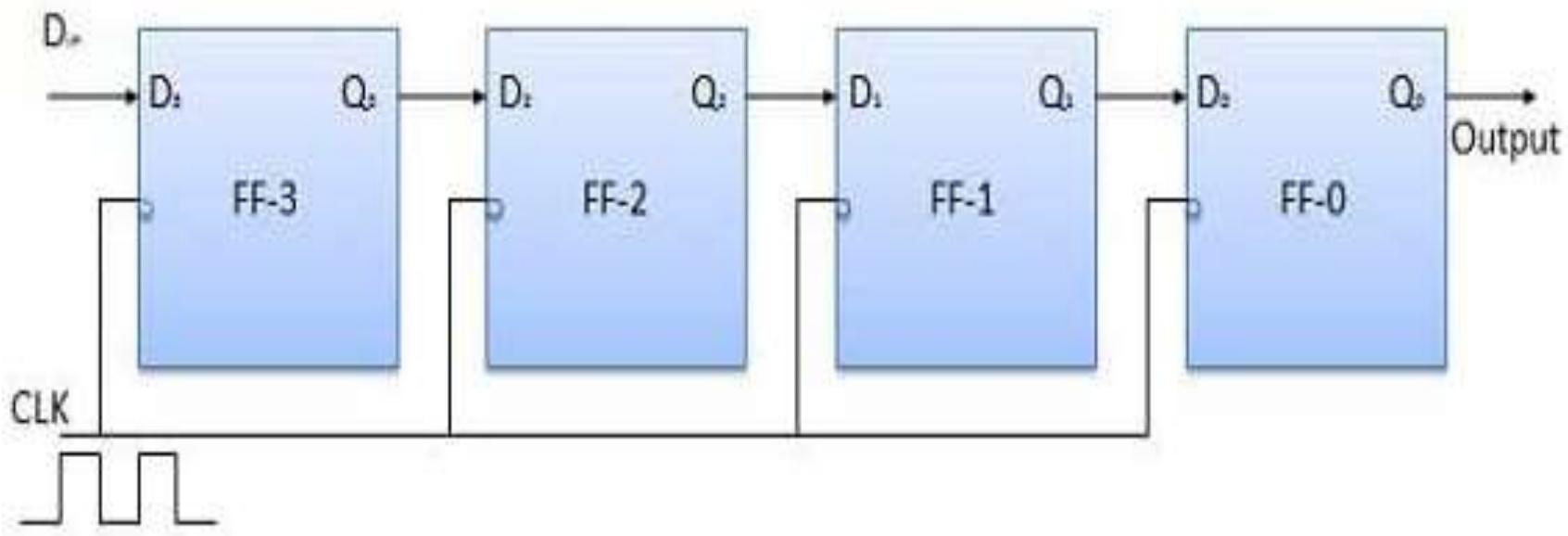
1. Flip-flop is a 1 bit memory cell which can be used for storing the digital data.
2. To increase the storage capacity in terms of number of bits, we have to use a group of flip-flop.
3. Such a group of flip-flop is known as a **Register**.
4. The **n-bit register** will consist of **n** number of flip-flop and it is capable of storing an **n-bit** word.
5. The binary data in a register can be moved within the register from one flip-flop to another. The registers that allow such data transfers are called as **shift registers**. There are four mode of operations of a shift register.

- Serial Input Serial Output
- Serial Input Parallel Output
- Parallel Input Serial Output
- Parallel Input Parallel Output



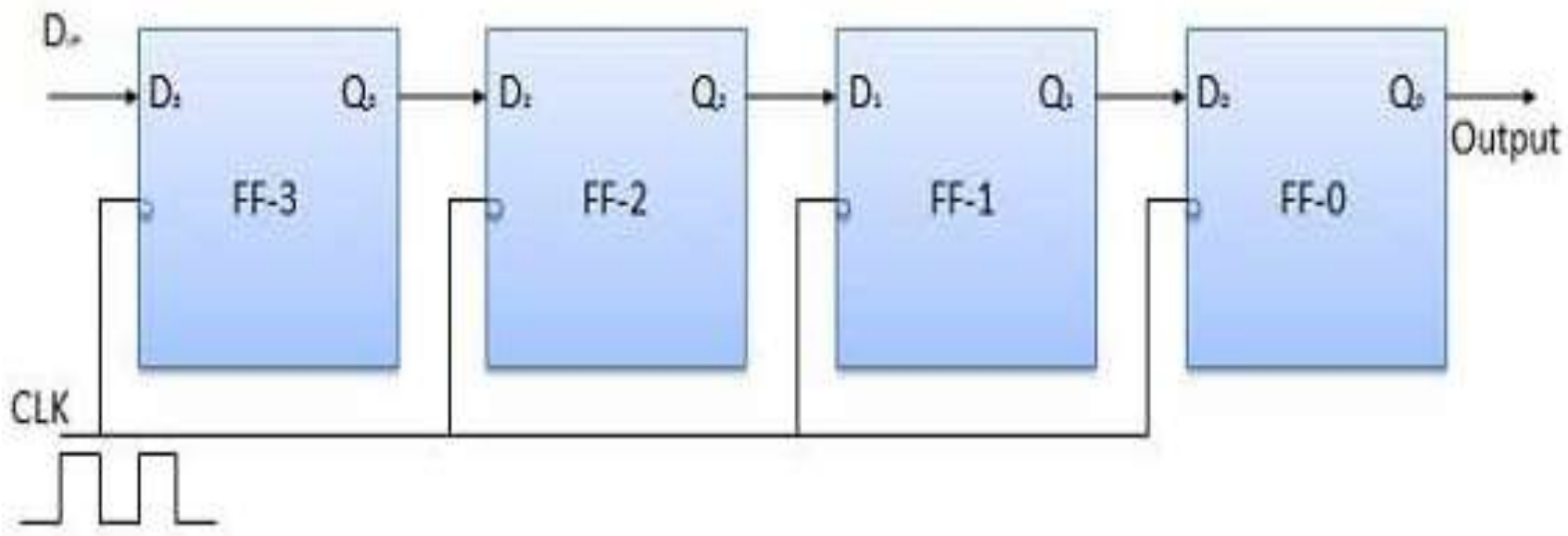
Registers

1. **Serial Input Serial Output**
   Let all the flip-flop be initially in the reset condition i.e. $Q_3 = Q_2 = Q_1 = Q_0 = 0$. If an entry of a four bit binary number 1 1 1 1 is made into the register, this number should be applied to $\mathbf{D_{in}}$ bit with the LSB bit applied first. The D input of FF-3 i.e. D_3 is connected to serial data input $\mathbf{D_{in}}$. Output of FF-3 i.e. Q_3 is connected to the input of the next flip-flop i.e. D_2 and so on.



Registers

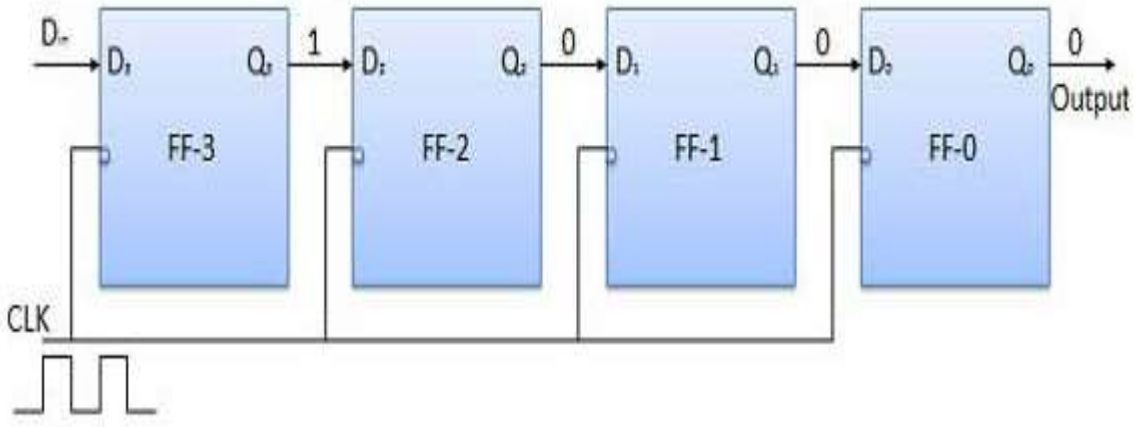
1. **Serial Input Serial Output**
   Let all the flip-flop be initially in the reset condition i.e. $Q_3 = Q_2 = Q_1 = Q_0 = 0$. If an entry of a four bit binary number 1 1 1 1 is made into the register, this number should be applied to $\mathbf{D_{in}}$ bit with the LSB bit applied first. The D input of FF-3 i.e. D_3 is connected to serial data input $\mathbf{D_{in}}$. Output of FF-3 i.e. Q_3 is connected to the input of the next flip-flop i.e. D_2 and so on.



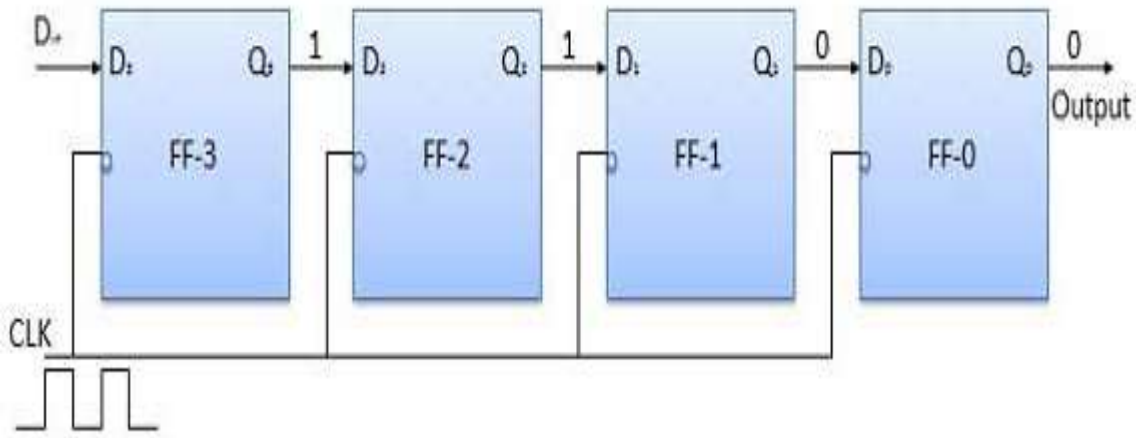
Registers

Operation:

Before application of clock signal, let Q_3 Q_2 Q_1 Q_0 = 0000 and apply LSB bit of the number to be entered to D_{in}. So $D_{in} = D_3 = 1$. Apply the clock. On the first falling edge of clock, the FF-3 is set, and stored word in the register is Q_3 Q_2 Q_1 Q_0 = 1000

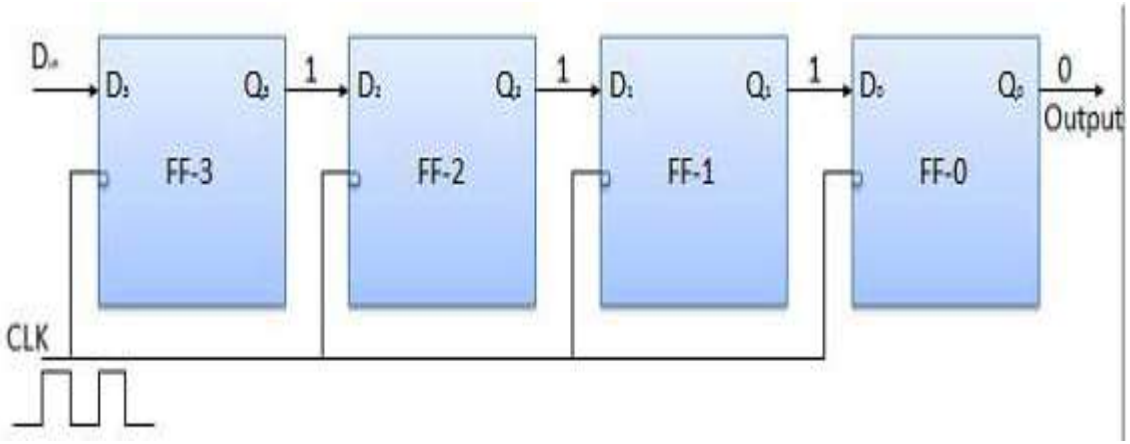


Apply the next bit to D_{in}. So $D_{in} = 1$. As soon as the next negative edge of the clock hits, FF-2 will set and the stored word change to Q_3 Q_2 Q_1 Q_0 = 1100.

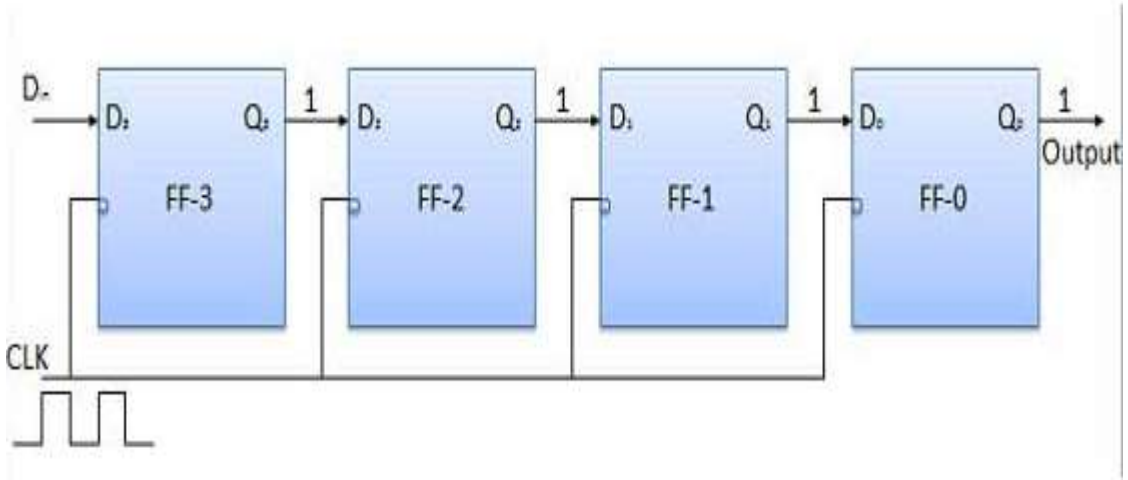


Registers

Apply the next bit to be stored i.e. 1 to D_{in}. Apply the clock pulse. As soon as the third negative clock edge hits, FF-1 will be set and output will be modified to $Q_3\ Q_2\ Q_1\ Q_0 = 1110$.



Similarly with $D_{in} = 1$ and with the fourth negative clock edge arriving, the stored word in the register is $Q_3\ Q_2\ Q_1\ Q_0 = 1111$.



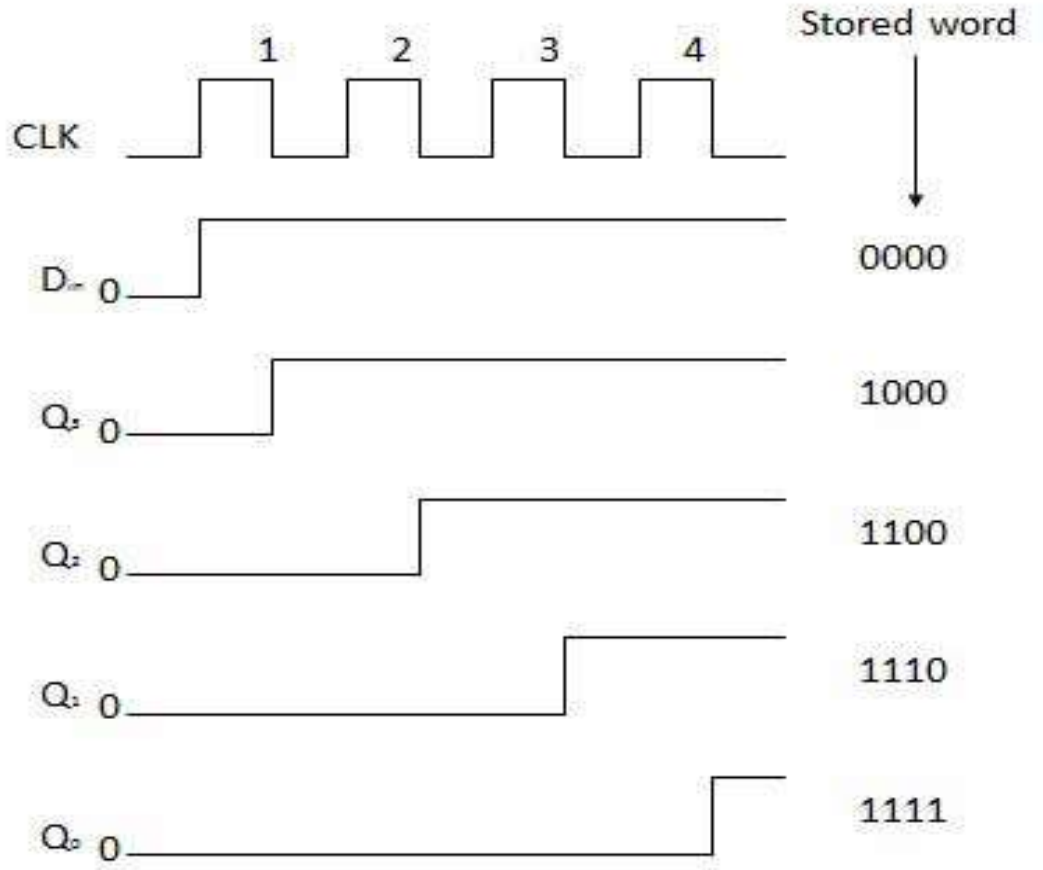
Registers

Truth Table

	CLK	$D_n = Q_3$	$Q_3 = D_2$	$Q_2 = D_1$	$Q_1 = D_0$	Q_0
Initially			0	0	0	0
(i)	↓	1 →	1	0	0	0
(ii)	↓	1 →	1	1	0	0
(iii)	↓	1 →	1	1	1	0
(iv)	↓	1 →	1	1	1	1

→ Direction of data travel

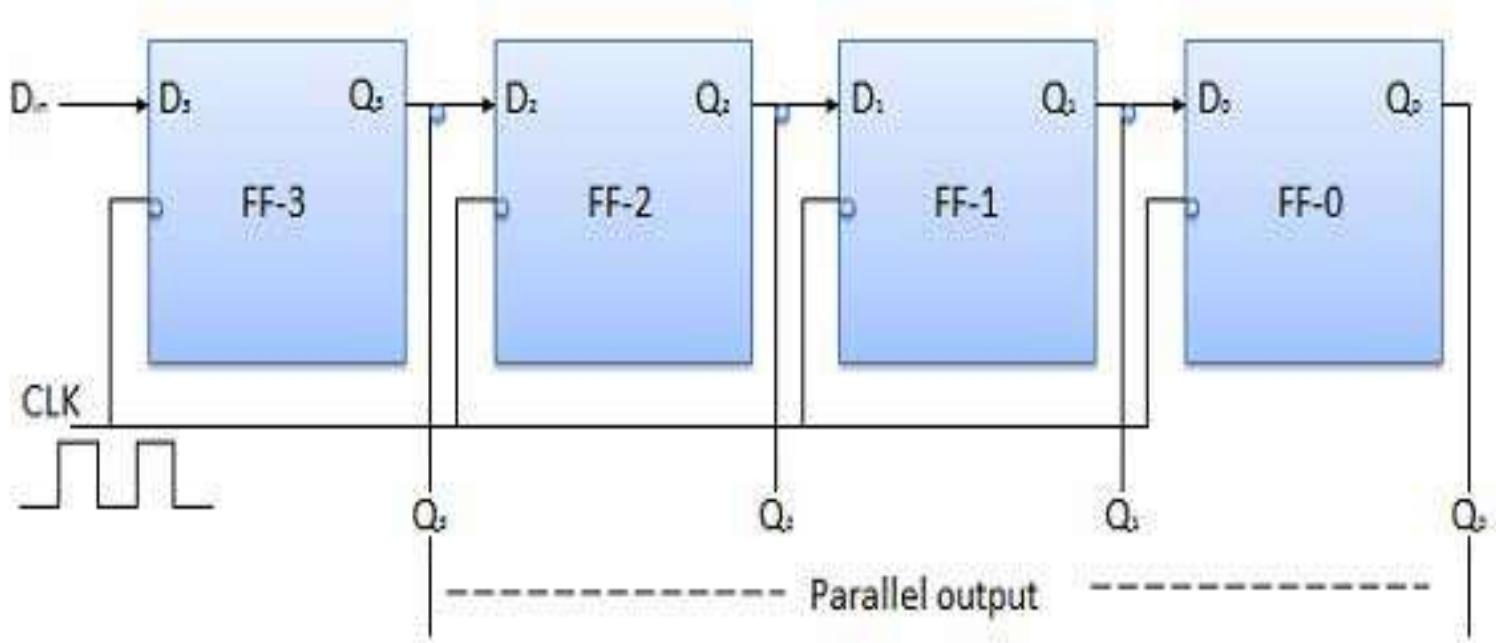
Waveform:



Registers

Serial Input Parallel Output:

1. In such types of operations, the data is entered serially and taken out in parallel fashion.
2. Data is loaded bit by bit. The outputs are disabled as long as the data is loading.
3. As soon as the data loading gets completed, all the flip-flops contain their required data, the outputs are enabled so that all the loaded data is made available over all the output lines at the same time.
4. 4 clock cycles are required to load a four bit word. Hence the speed of operation of SIPO mode is same as that of SISO mode.



Registers

Parallel Input Serial Output (PISO):

1. Data bits are entered in parallel fashion.
2. The circuit shown below is a four bit parallel input serial output register.
3. Output of previous Flip Flop is connected to the input of the next one via a combinational circuit.
4. The binary input word B_0, B_1, B_2, B_3 is applied though the same combinational circuit.
5. There are two modes in which this circuit can work namely - shift mode or load mode.

Registers

Parallel Input Serial Output (PISO):

1. Data bits are entered in parallel fashion.
2. The circuit shown below is a four bit parallel input serial output register.
3. Output of previous Flip Flop is connected to the input of the next one via a combinational circuit.
4. The binary input word B_0, B_1, B_2, B_3 is applied though the same combinational circuit.
5. There are two modes in which this circuit can work namely - shift mode or load mode.

Load mode

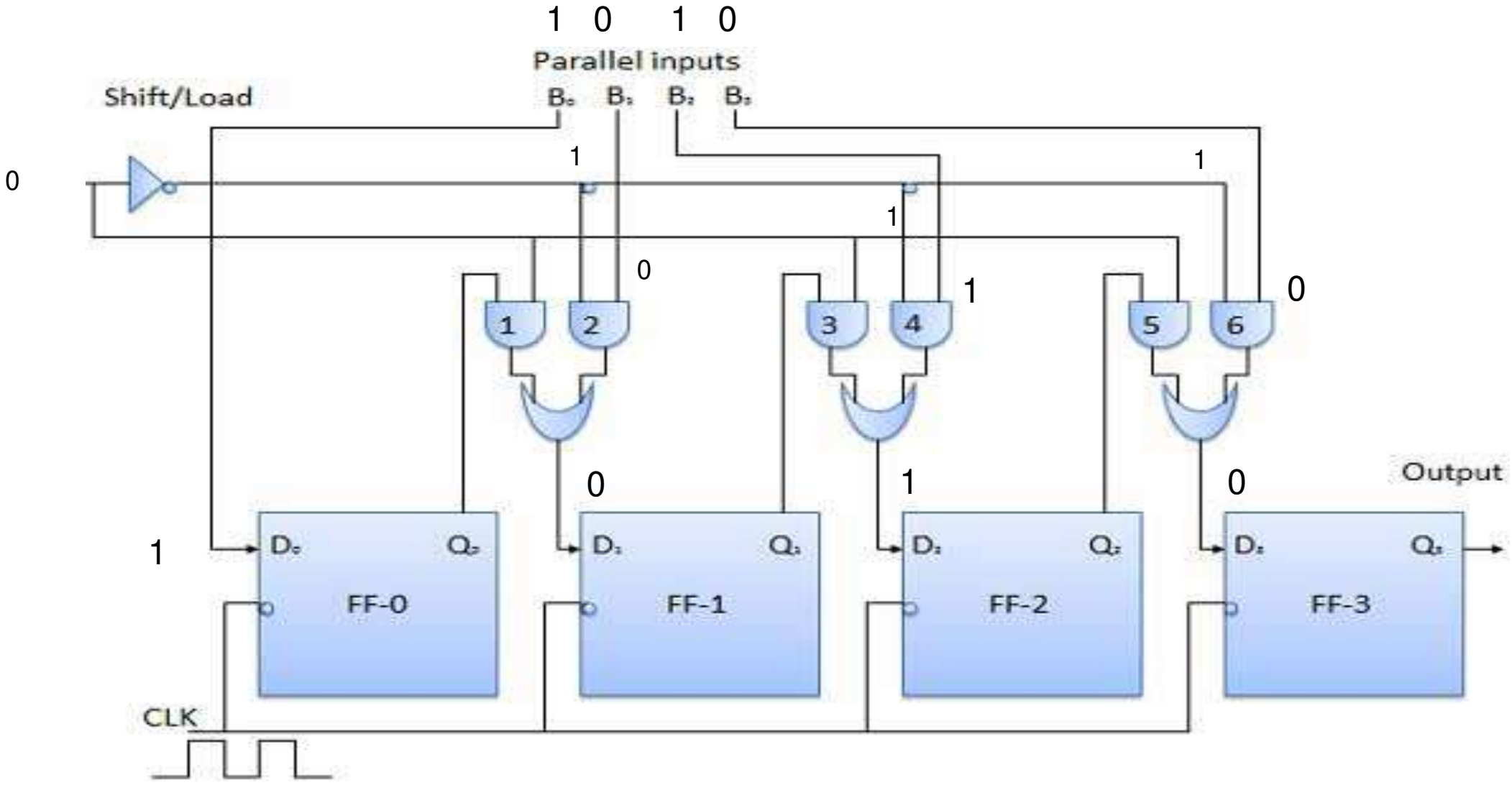
When the shift/load bar line is low (0), the AND gate 2, 4 and 6 become active they will pass B_1, B_2, B_3 bits to the corresponding flip-flops. On the low going edge of clock, the binary input B_0, B_1, B_2, B_3 will get loaded into the corresponding flip-flops. Thus parallel loading takes place.

Shift mode

When the shift/load bar line is low (1), the AND gate 2, 4 and 6 become inactive. Hence the parallel loading of the data becomes impossible. But the AND gate 1,3 and 5 become active. Therefore the shifting of data from left to right bit by bit on application of clock pulses. Thus the parallel in serial out operation takes place.

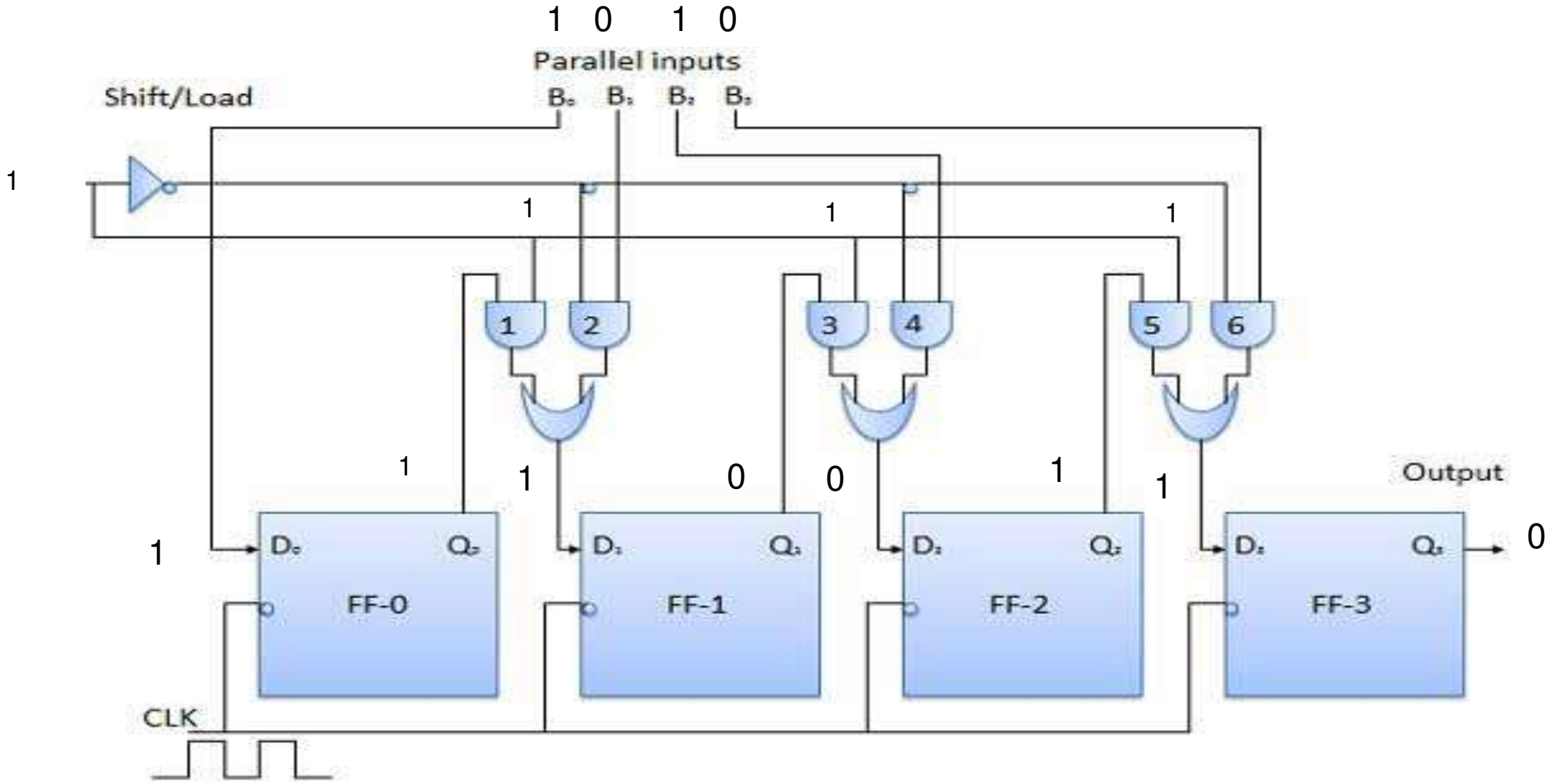
Registers

Load



Registers

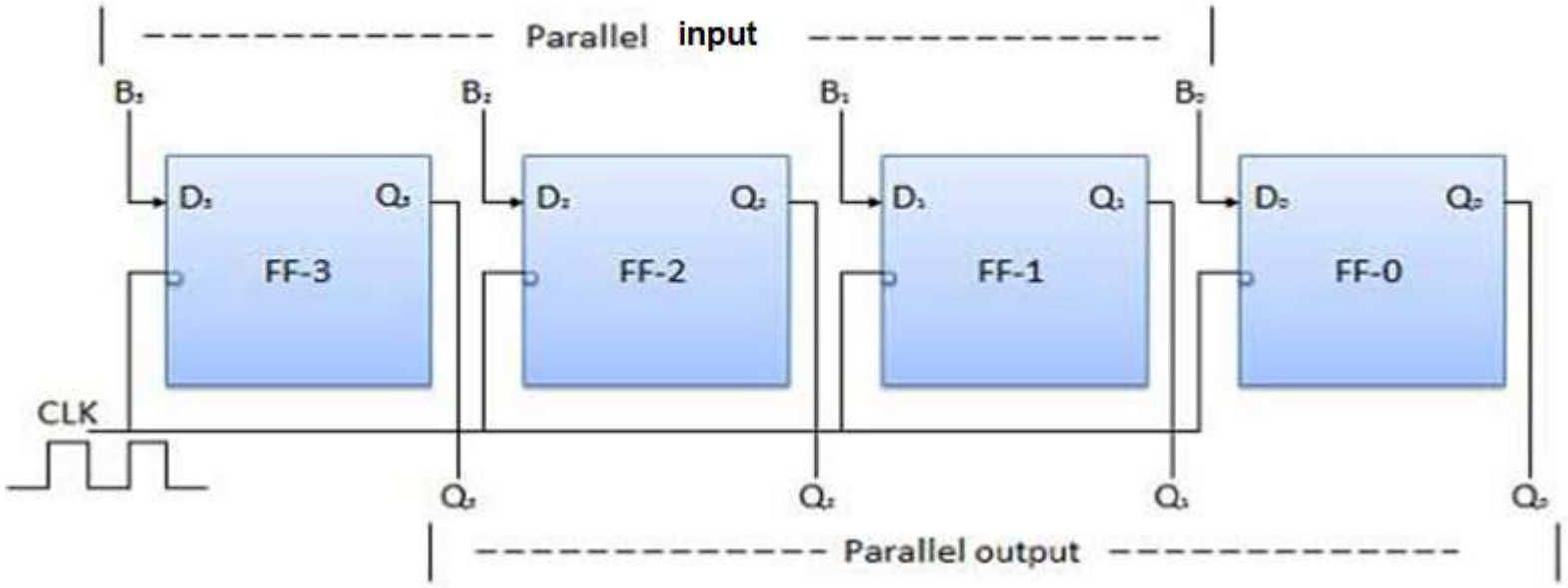
Shift



Registers

Parallel Input Parallel Output (PIPO):

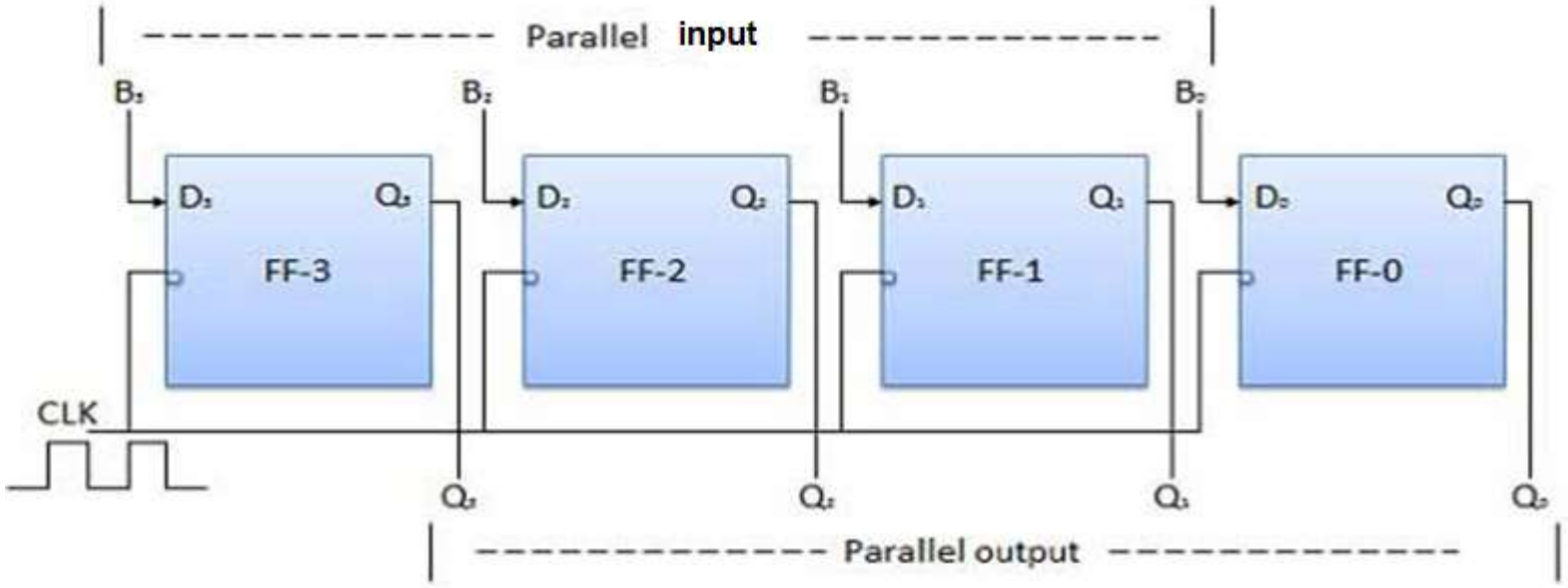
1. In this mode, the 4 bit binary input B_0, B_1, B_2, B_3 is applied to the data inputs D_0, D_1, D_2, D_3 respectively of the four flip-flops.
2. As soon as a negative clock edge is applied, the input binary bits will be loaded into the flip-flops simultaneously.
3. The loaded bits will appear simultaneously to the output side. Only clock pulse is essential to load all the bits.



Registers

Parallel Input Parallel Output (PIPO):

1. In this mode, the 4 bit binary input B_0, B_1, B_2, B_3 is applied to the data inputs D_0, D_1, D_2, D_3 respectively of the four flip-flops.
2. As soon as a negative clock edge is applied, the input binary bits will be loaded into the flip-flops simultaneously.
3. The loaded bits will appear simultaneously to the output side. Only clock pulse is essential to load all the bits.



Shift Register

There are two types of shift registers,

1. Left Shift
2. Right Shift

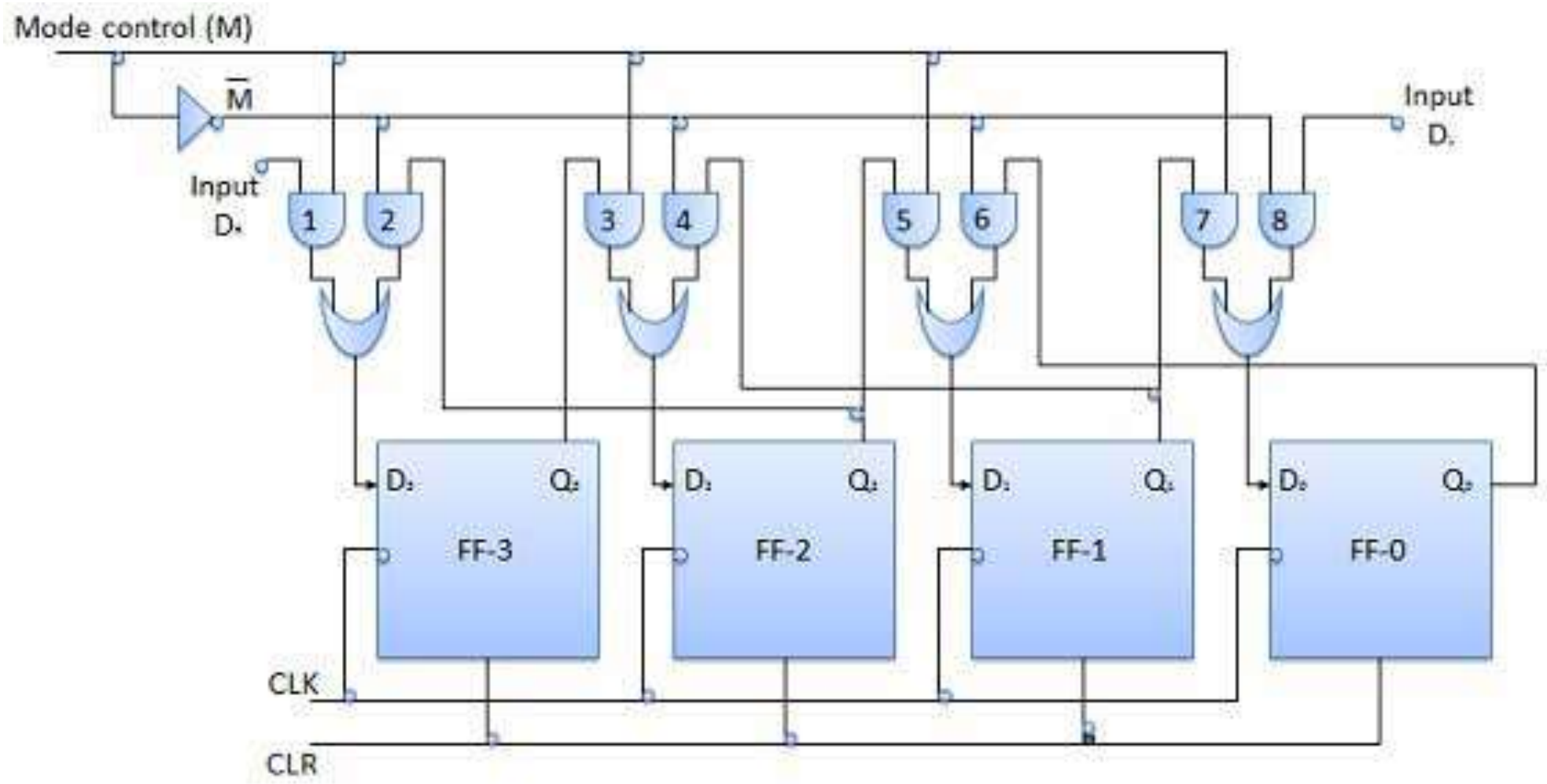
Applications of shift Registers

1. The shift registers are used for temporary data storage.
2. The shift registers are also used for data transfer and data manipulation.
3. The serial-in serial-out and parallel-in parallel-out shift registers are used to produce time delay to digital circuits.
4. 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 line is required.
5. A Parallel in Serial out shift register us used to convert parallel data to serial data.

Bidirectional Shift Register

1. If a binary number is shifted left by one position then it is equivalent to multiplying the original number by 2. Similarly if a binary number is shifted right by one position then it is equivalent to dividing the original number by 2.
2. Hence if we want to use the shift register to multiply and divide the given binary number, then we should be able to move the data in either left or right direction.
3. Such a register is called bi-directional register. A four bit bi-directional shift register is shown in fig.
4. There are two serial inputs namely the serial right shift data input DR, and the serial left shift data input DL along with a mode select input (M).

Bidirectional Shift Register



Bidirectional Shift Register

Operation:

With M = 1 – Shift right operation	With M = 0 – Shift left operation
1. If M = 1, then the AND gates 1, 3, 5 and 7 are enabled whereas the remaining AND gates 2, 4, 6 and 8 will be disabled. 2. The data at D_R is shifted to right bit by bit from FF-3 to FF-0 on the application of clock pulses. Thus with M = 1 we get the serial right shift operation.	1. When the mode control M is connected to 0 then the AND gates 2, 4, 6 and 8 are enabled while 1, 3, 5 and 7 are disabled. 2. The data at D_L is shifted left bit by bit from FF-0 to FF-3 on the application of clock pulses. Thus with M = 0 we get the serial right shift operation.

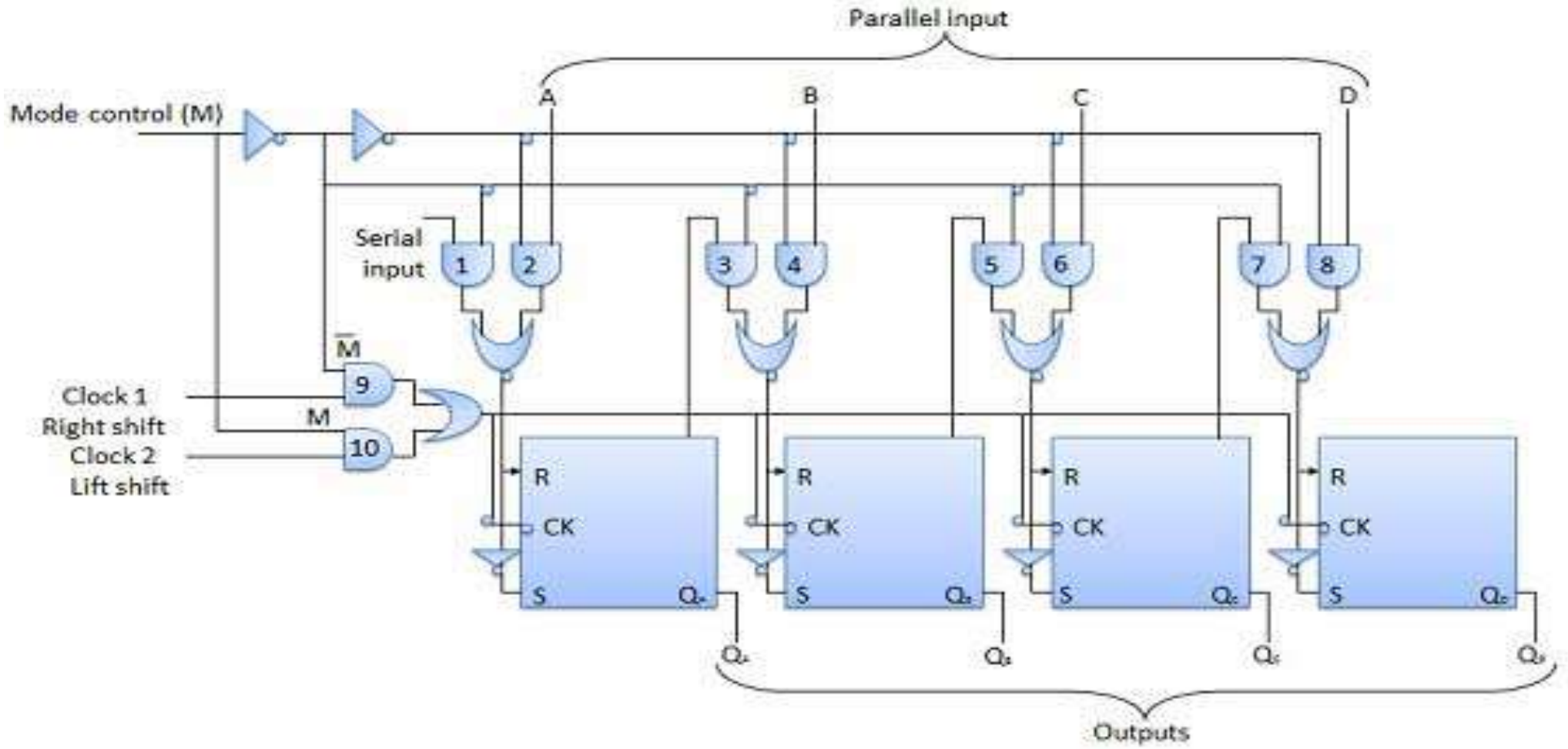
Universal Shift Register

- A shift register which can shift the data in only one direction is called a uni-directional shift register.
- A shift register which can shift the data in both directions is called a bi-directional shift register. Applying the same logic, a shift register which can shift the data in both directions as well as load it parallel, is known as a universal shift register. The shift register is capable of performing the following operation −

1. Parallel loading
2. Left Shifting
3. Right shifting

- The mode control input is connected to logic 1 for parallel loading operation whereas it is connected to 0 for serial shifting.
- With mode control pin connected to ground, the universal shift register acts as a bi-directional register.
- For serial left operation, the input is applied to the serial input which goes to AND gate-1 shown in figure. Whereas for the shift right operation, the serial input is applied to D input.

Universal Shift Register



Agenda

01

Flip-Flop: SR, JK,D,T, Preset and clear, Master Slave JK FF, Truth Tables and Excitation Tables, Conversion of FF

02

Registers: SISO, SIPO, PISO, PIPO, Shift registers, Bidirectional shift register, Universal shift register.

03

Counters: Asynchronous counter, synchronous counter, BCD counter, Ring Counter, Johnson Counter, Modulus of counter (IC 7490)

04

Synchronous Sequential Circuit Design: Models- Moore and Mealy, State diagram and state table, Design Procedure, Sequence geneartor and detector.

Digital 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 of two types.

1. **Asynchronous or ripple counters**.

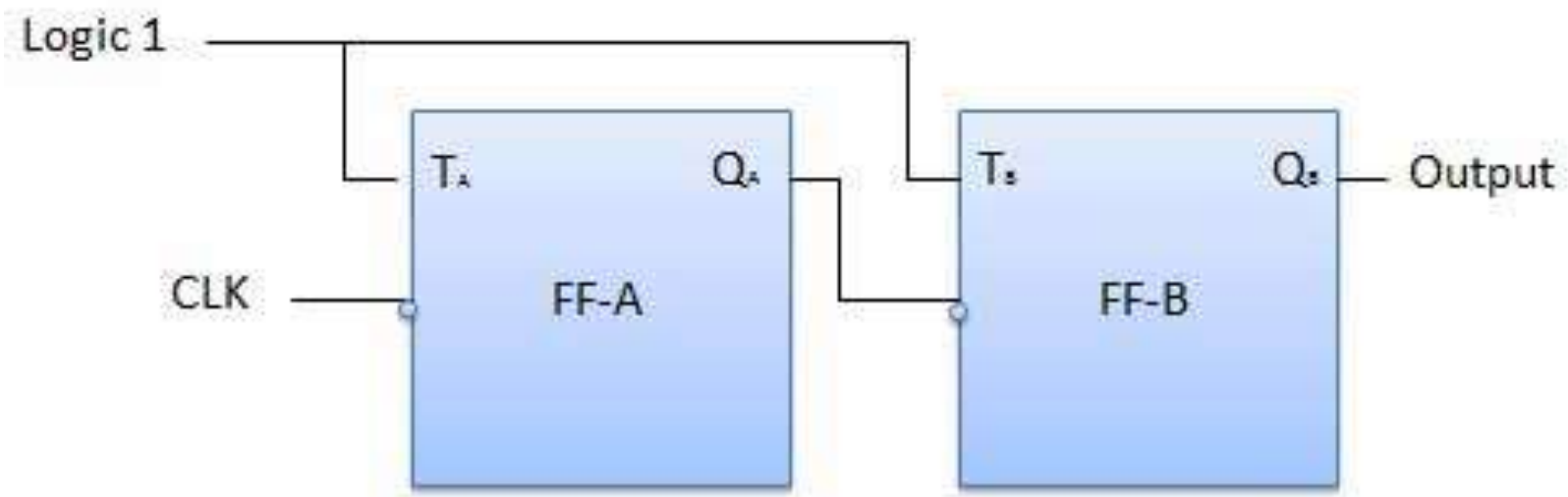
- Asynchronous Binary up counter
- Asynchronous Binary down counter

2. **Synchronous counters.**

- synchronous Binary up counter
- synchronous Binary down counter

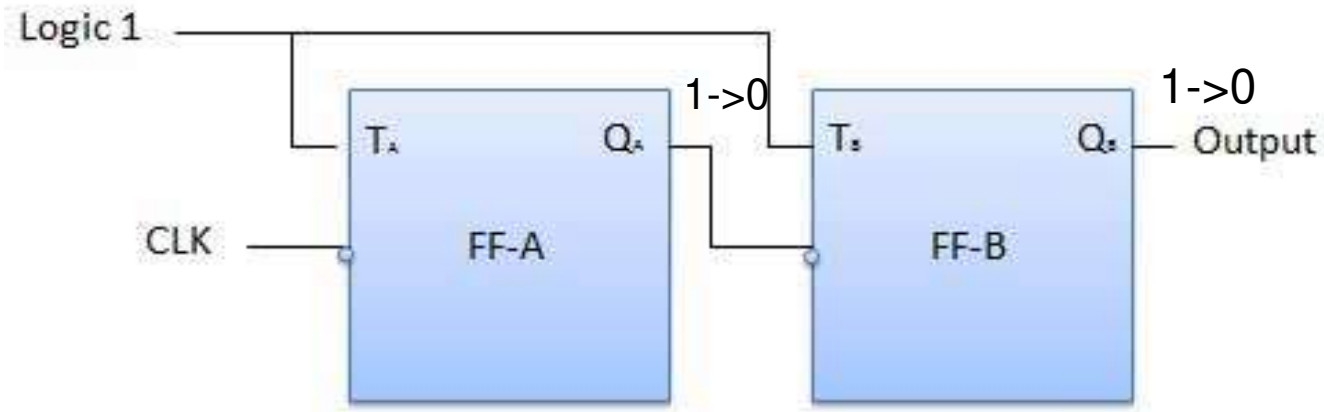
Asynchronous or ripple UP counters

1. The logic diagram of a 2-bit ripple up counter is shown in figure.
2. The toggle (T) flip-flop are being used. But we can use the JK flip-flop also with J and K connected permanently to logic 1. External clock is applied to the clock input of flip-flop A and Q_A output is applied to the clock input of the next flip-flop i.e. FF-B.



Asynchronous or ripple UP counters

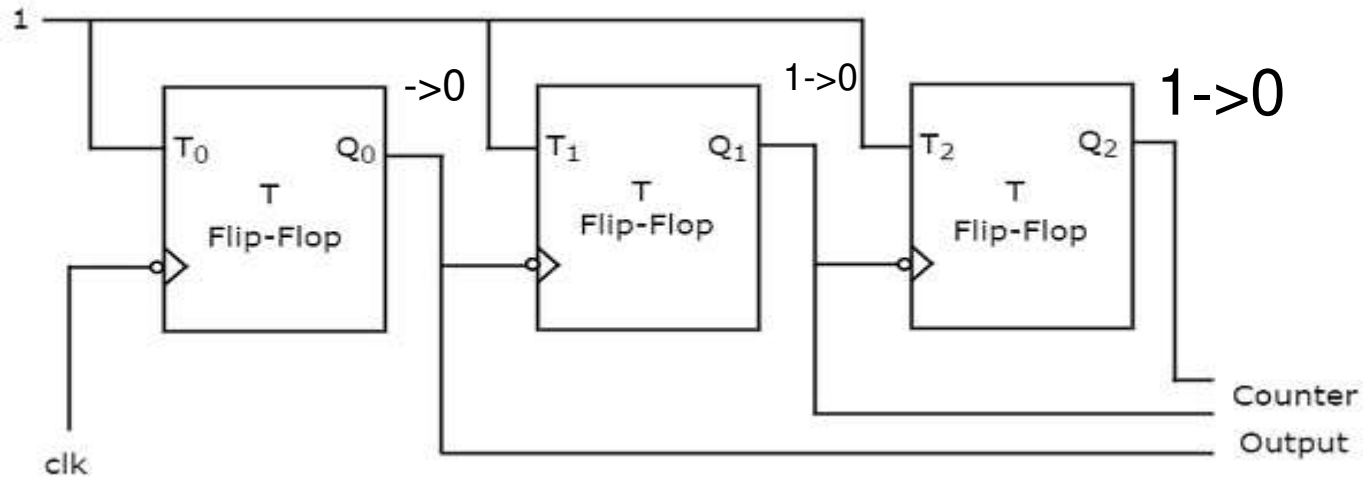
Operation:



CLK	Q_B	Q_A	Decimal Count
INITIALLY	**0**	**0**	0
↓	0	1	1
↓	1	0	2
↓	1	1	3
↓	0	0	0

Asynchronous or ripple UP counters

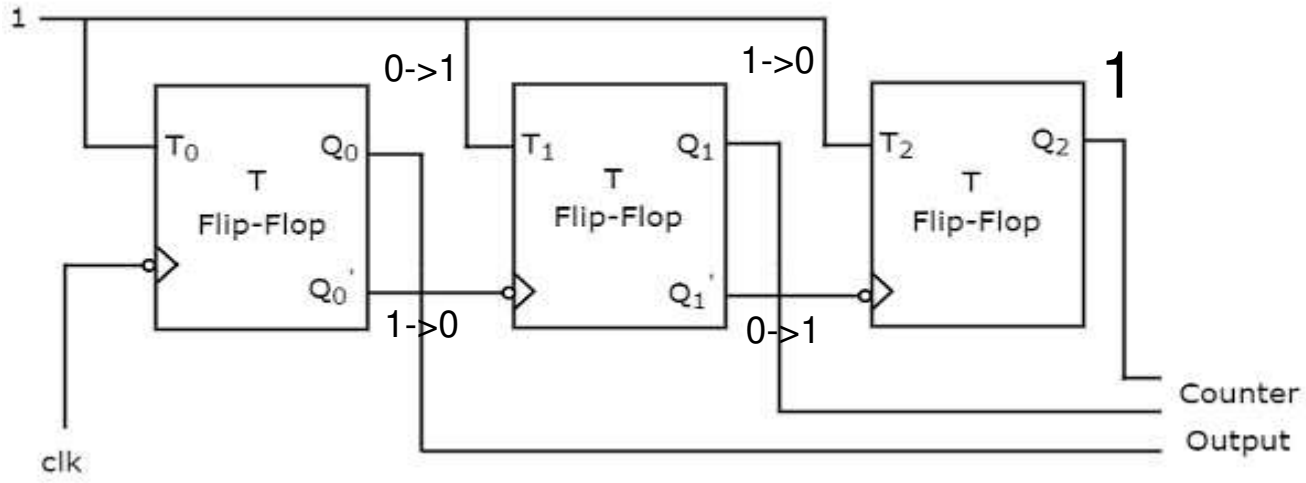
Operation:



CLK	Q_2	Q_1	Q_0	Decimal Count
	0	**0**	**0**	0
↓	0	0	1	1
↓	0	1	0	2
↓	0	1	1	3
↓	1	0	0	4
↓	1	0	1	5
↓	1	1	0	6
↓	1	1	1	7
↓	0	0	0	0

Asynchronous or ripple DOWN counters

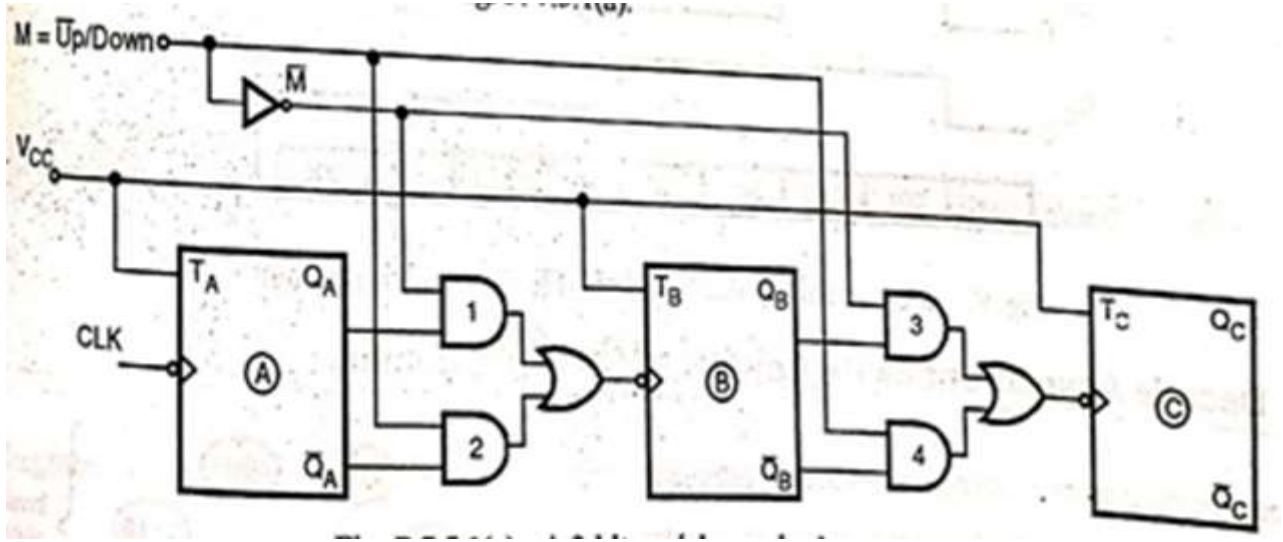
Operation:



CLK +	Q_2	Q_1	Q_0	Decimal Count
	0	**0**	**0**	0
↓	1	1	1	7
↓	1	1	0	6
↓	1	0	1	5
↓	1	0	0	4
↓	0	1	1	3
↓	0	1	0	2
↓	0	0	1	1
↓	0	0	0	0

Asynchronous or ripple UP-DOWN counter

Operation:



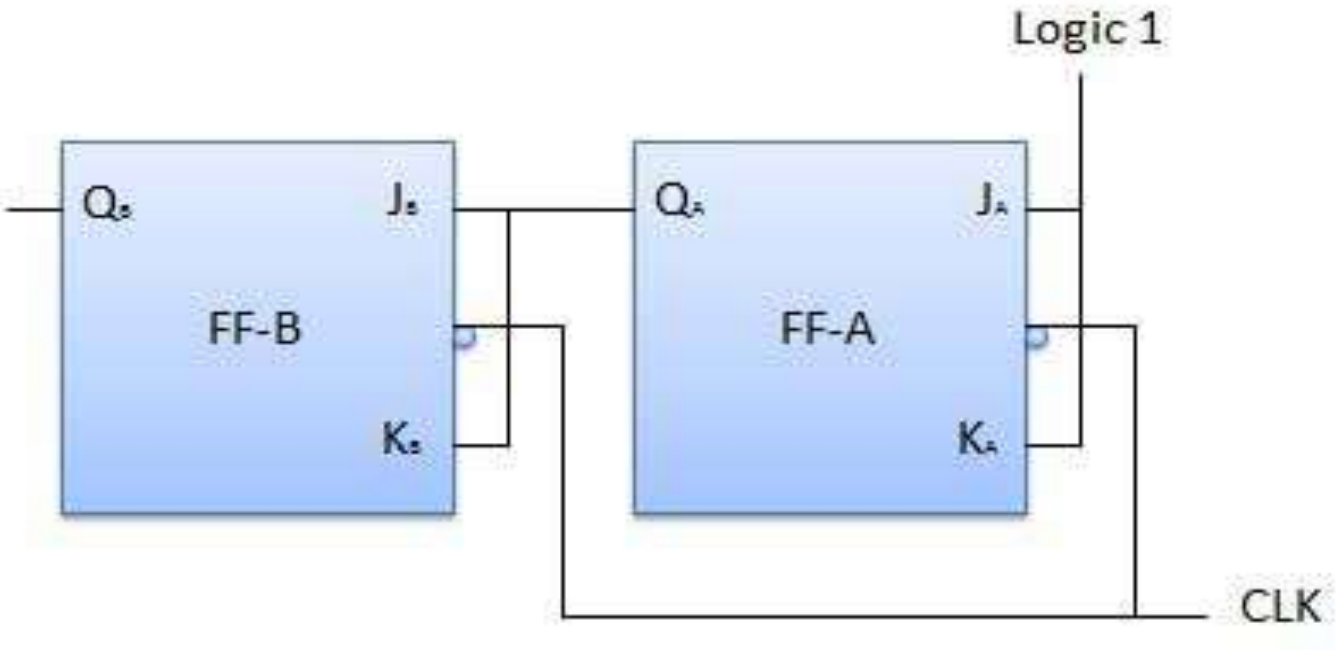
M	CLK	Q_C	Q_B	Q_A	Decimal Count
		0	**0**	**0**	0
0	↓	0	0	1	1
0	↓	0	1	0	2
0	↓	0	1	1	3
0	↓	1	0	0	4
0	↓	1	0	1	5
0	↓	1	1	0	6
0	↓	1	1	1	7
0	↓	0	0	0	0
1	↓	1	1	1	7
1	↓	1	1	0	6
1	↓	1	0	1	5
1	↓	1	0	0	4
1	↓	0	1	1	3

Synchronous UP counters

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

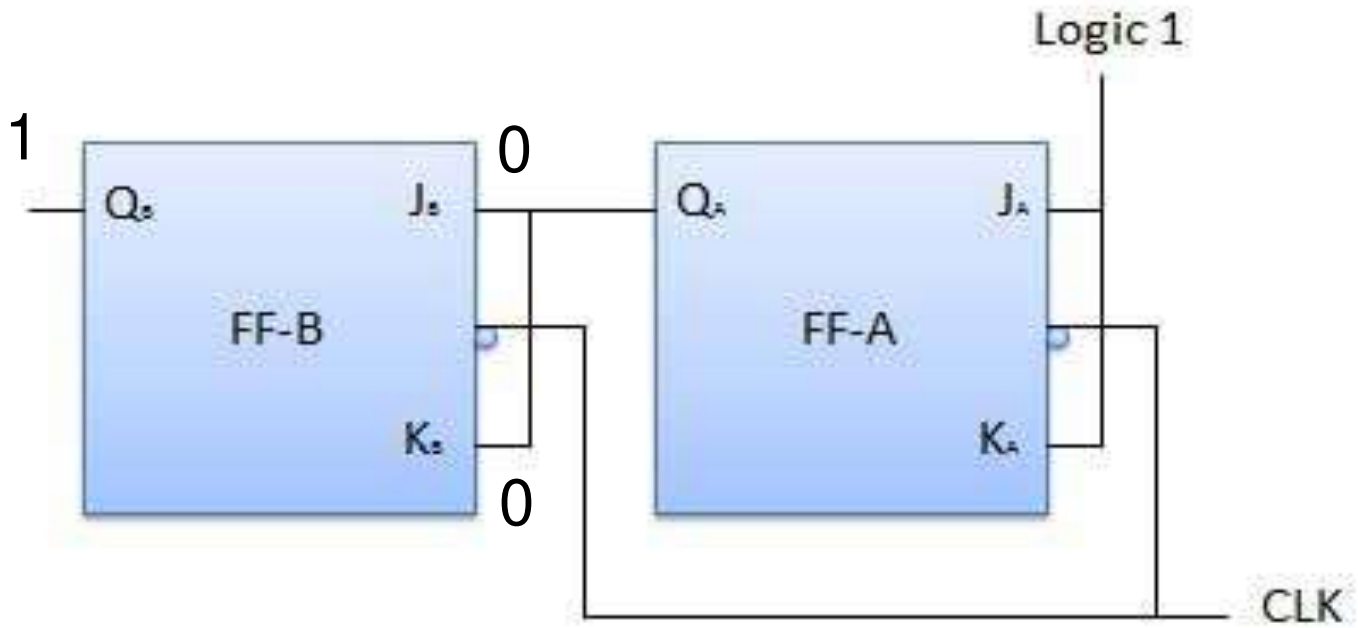
2-bit Synchronous up counter

The J_A and K_A inputs of FF-A are tied to logic 1. So FF-A will work as a toggle flip-flop. The J_B and K_B inputs are connected to Q_A



Synchronous UP counters

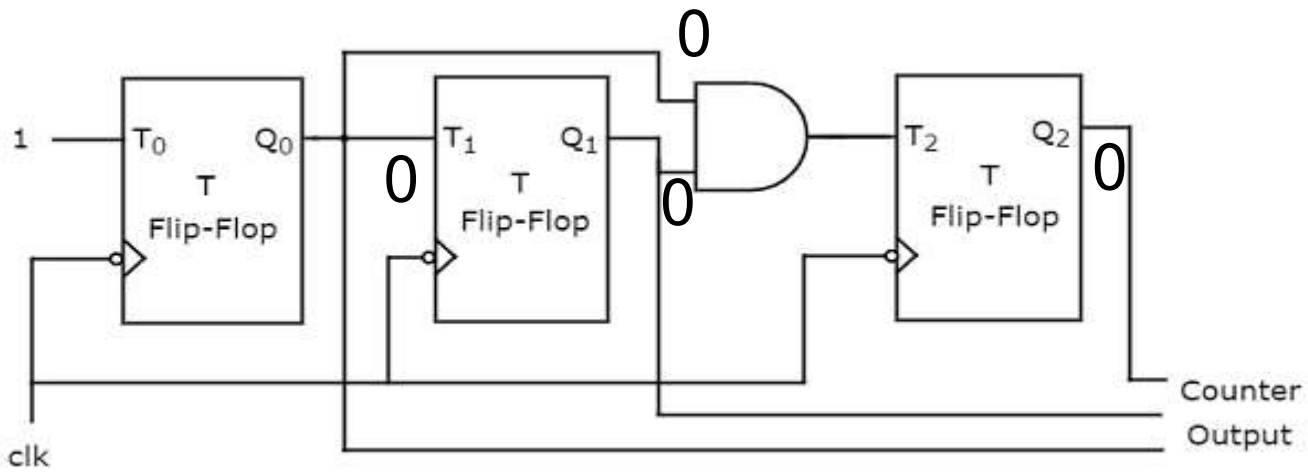
Operation:



CLK	Q_B	Q_A	Decimal Count
	0	**0**	0
↓	0	1	1
↓	1	0	2
↓	1	1	3
↓	0	0	0

Synchronous UP counters

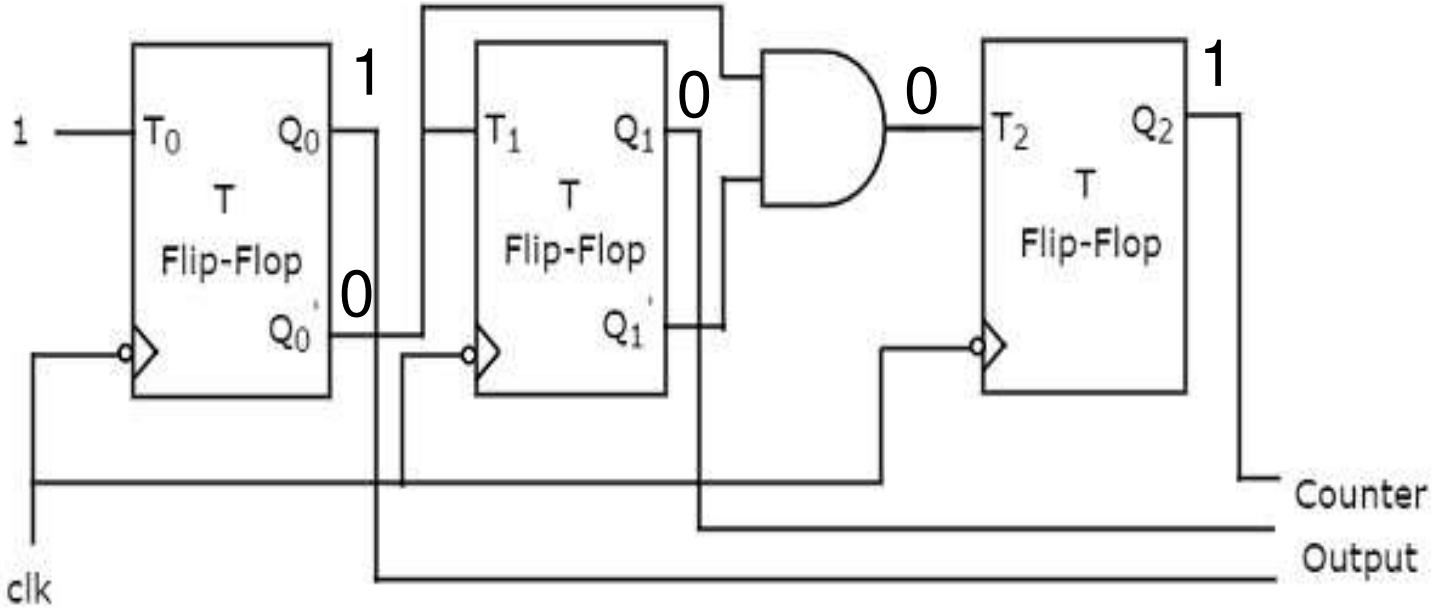
Operation:



CLK	Q_2	Q_1	Q_0	Decimal Count
	0	**0**	**0**	0
↓	0	0	1	1
↓	0	1	0	2
↓	0	**1**	**1**	3
↓	1	0	0	4
↓	1	0	1	5
↓	1	1	0	6
↓	1	**1**	**1**	7
↓	0	0	0	0

Synchronous DOWN counters

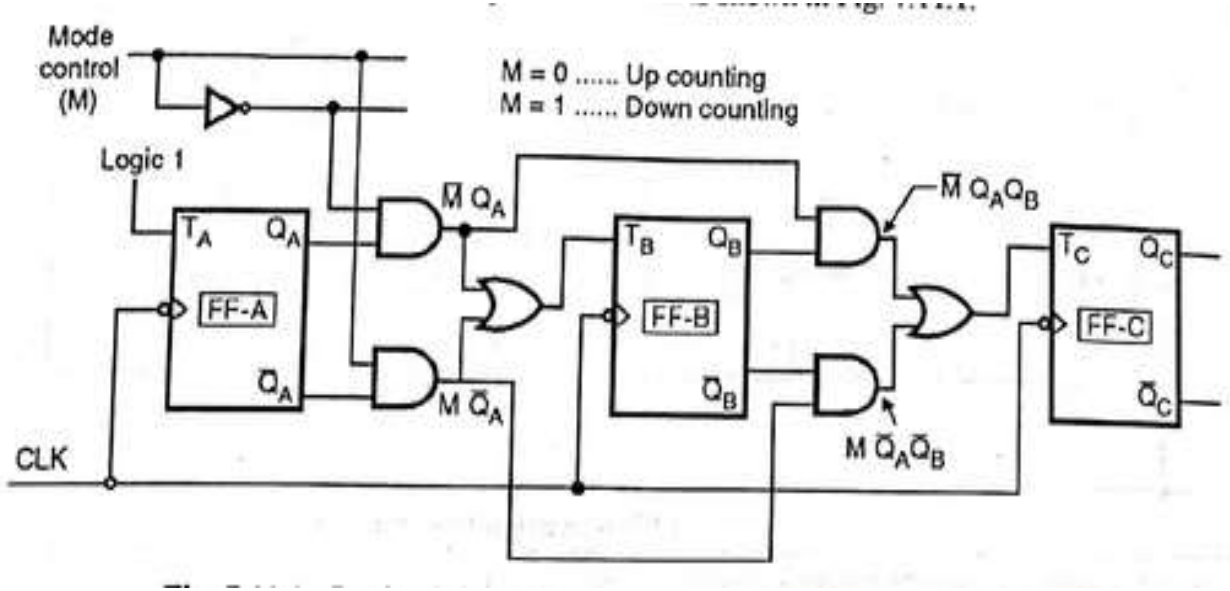
Operation:



CLK	Q_2	Q_1	Q_0	Decimal Count
	0	**0**	**0**	0
↓	1	1	1	
↓	1	1	0	
↓				
↓				
↓				
↓				
↓				
↓				

Synchronous UP-DOWN counters

Operation:



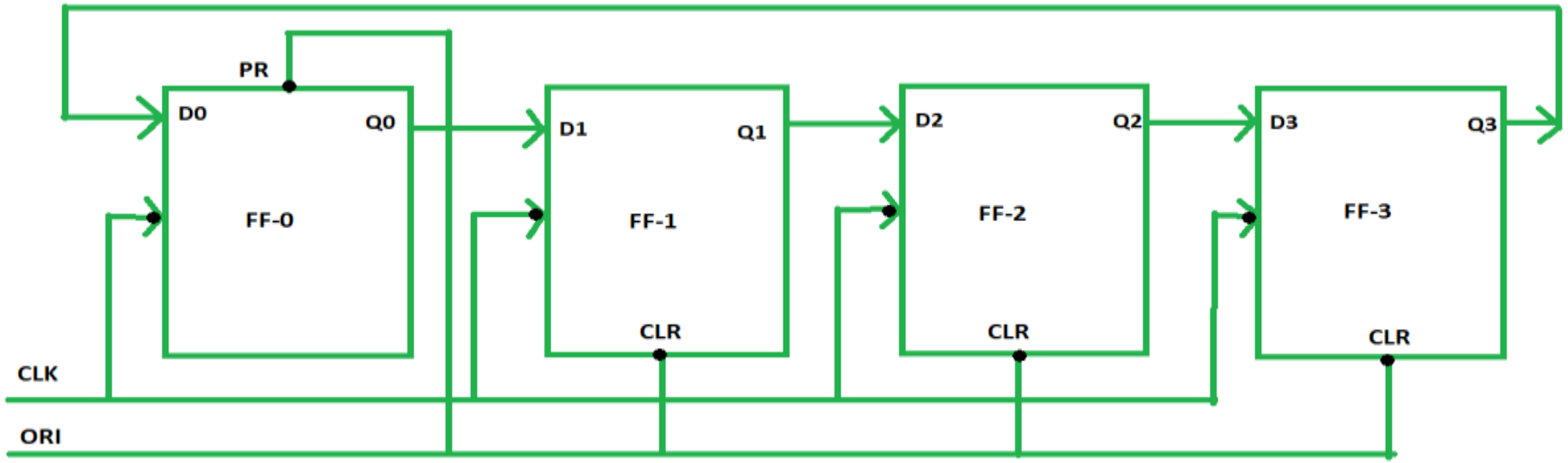
M	CLK	Q_2	Q_1	Q_0	Decimal Count
		0	**0**	**0**	0
0		0	0	1	1
0		0	1	0	2
0		0	1	1	3
0		1	0	0	4
0		1	0	1	5
0		1	1	0	6
0		1	1	1	7
0		0	0	0	0
1		1	1	1	7
1		1	1	0	6
1		1	0	1	5

Ring Counter

1. Ring counter is a typical application of Shift resister. Ring counter is almost same as the shift counter. The only change is that the output of the last flip-flop is connected to the input of the first flip-flop in case of ring counter but in case of shift resister it is taken as output. Except this all the other things are same.

No. of states in Ring counter = No. of flip-flop used

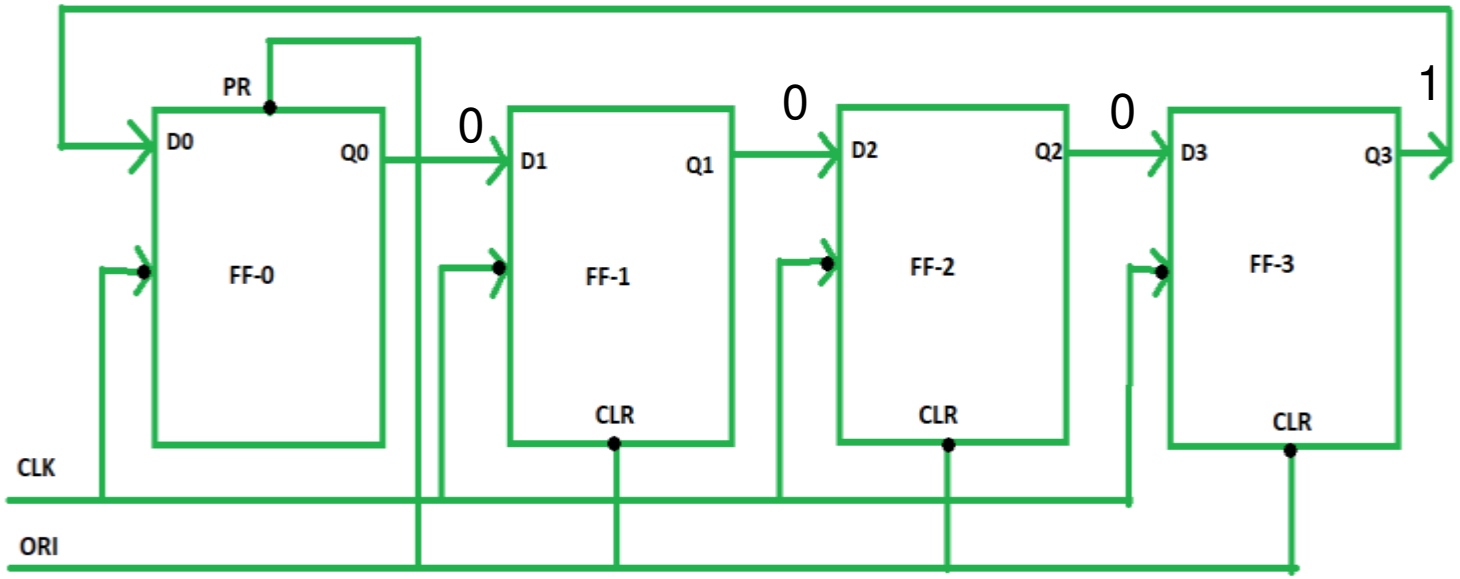
So, for designing 4-bit Ring counter we need 4 flip-flop.



Ring Counter

Ring Counter

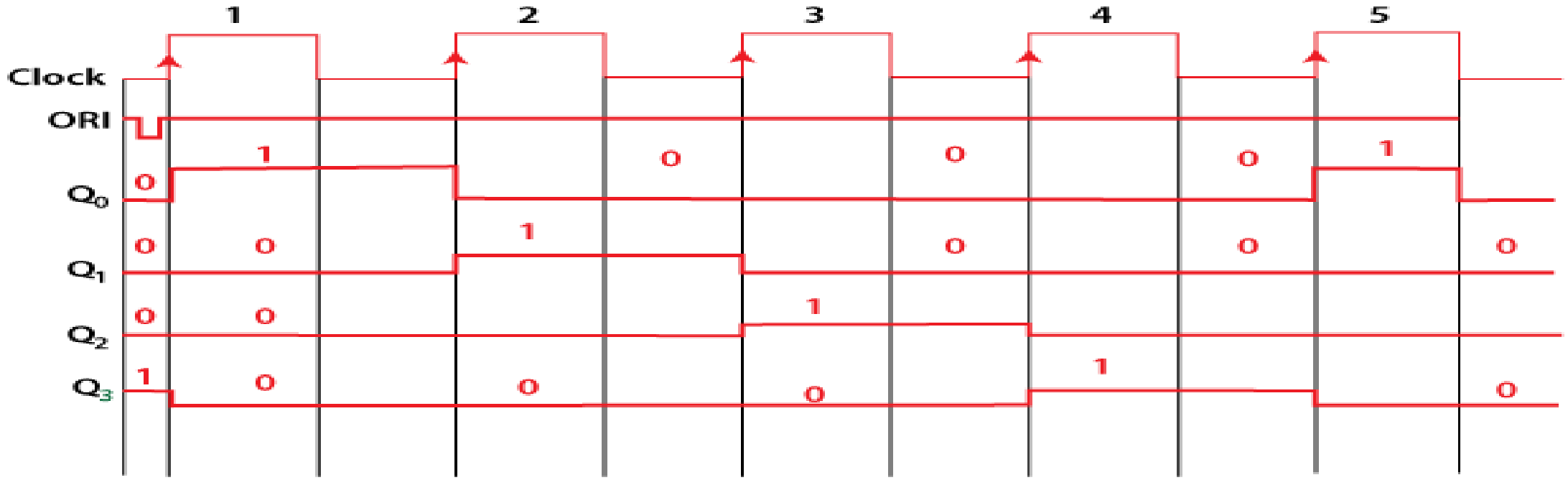
1. When PR is 0, then the output is 1. And when CLR is 0, then the output is 0. Both PR and CLR are active low signal that is always works in value 0.



Ring Counter

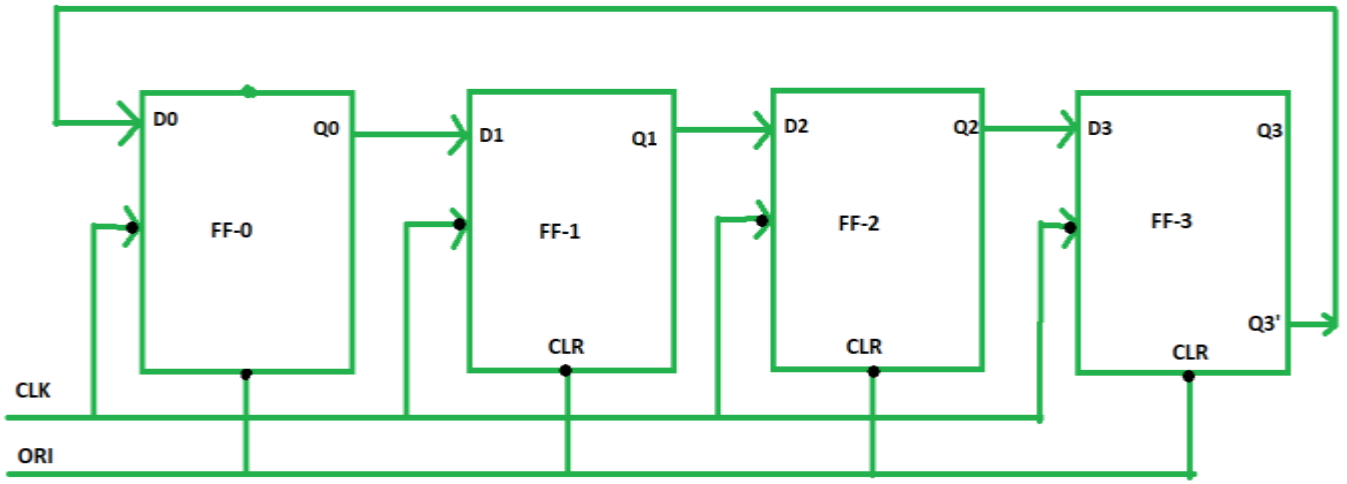
CLK	Q0	Q1	Q2	Q3
ini	1	0	0	0
↓	0	1	0	0
↓	0	0	1	0
↓	0	0	0	1
↓	1	0	0	0
	0	1	0	0

Ring Counter



Johnson Counter

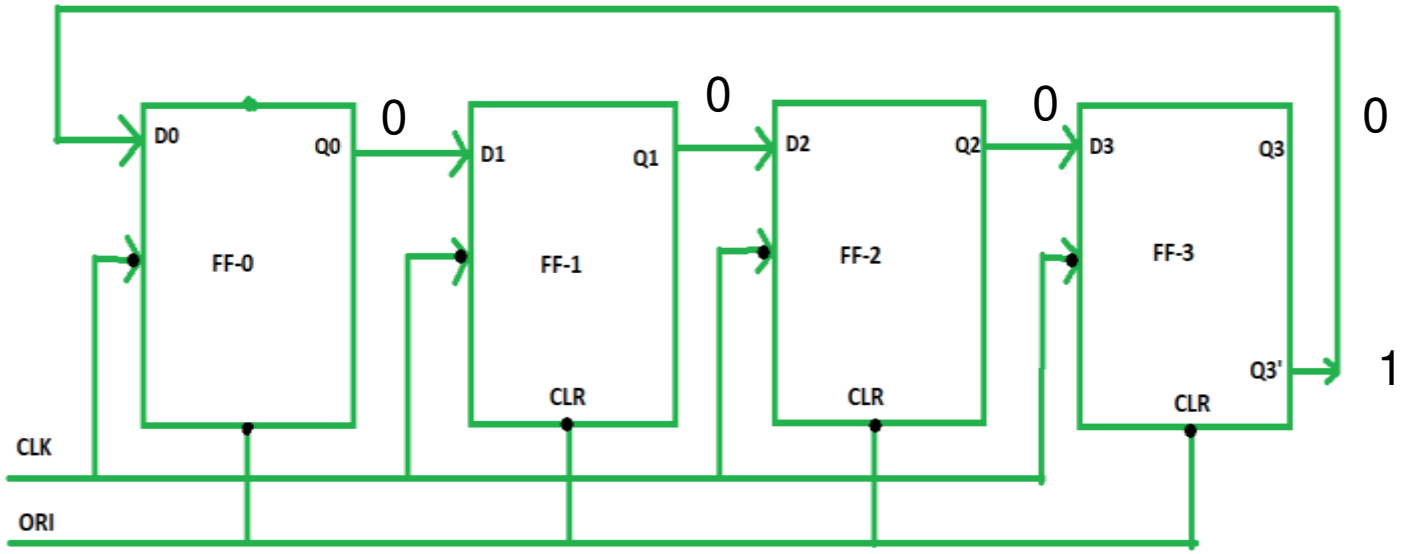
1. Johnson counter also known as creeping counter, is an example of synchronous counter.
2. In Johnson counter, the complemented output of last flip flop is connected to input of first flip flop and to implement n-bit Johnson counter we require n flip-flop.
3. It is one of the most important type of shift register counter. It is formed by the feedback of the output to its own input.
4. Johnson counter is a ring with an inversion. Another name of Johnson counter are: creeping counter, twisted ring counter, walking counter, mobile counter and switch tail counter.



Twisted Ring Counter

Johnson Counter

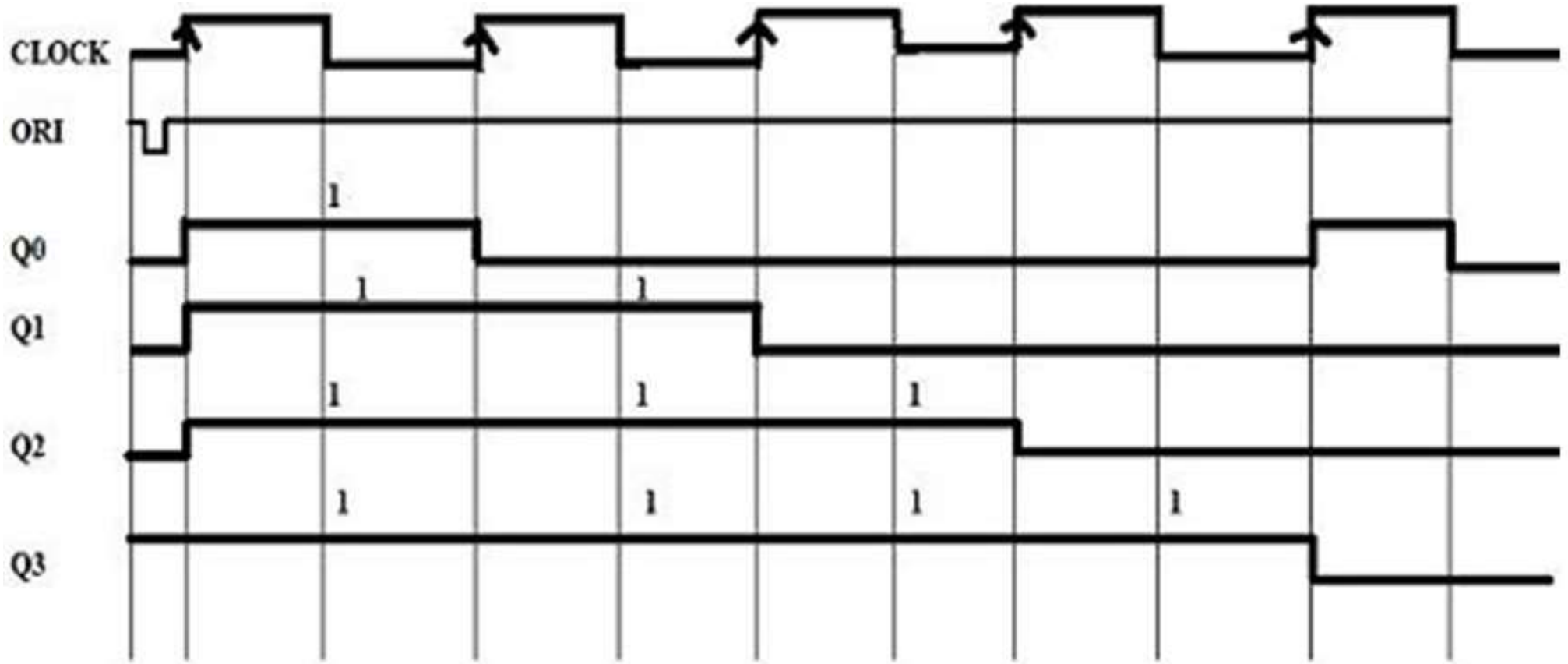
1. All four flip flops are reset.



Twisted Ring Counter

CLK	Q0	Q1	Q2	Q3
	0	0	0	0
↓	1	0	0	0
↓	1	1	0	0
↓	1	1	1	0
↓	1	1	1	1
↓	0	1	1	1
↓	0	0	1	1
↓	0	0	0	1
↓	0	0	0	0
↓	1	0	0	0

Johnson Counter



Johnson Counter

Advantages of Johnson counter:

1. The Johnson counter has same number of flip flop but it can count twice the number of states the ring counter can count.
2. It can be implemented using D and JK flip flop.
3. Johnson ring counter is used to count the data in a continuous loop.
4. Johnson counter is a self-decoding circuit.

Disadvantages of Johnson counter:

1. Johnson counter doesn't count in a binary sequence.
2. In Johnson counter more number of states remain unutilized than the number of states being utilized.
3. The number of flip flops needed is one half the number of timing signals.
4. It can be constructed for any number of timing sequence.

Applications of Johnson counter:

1. Johnson counter is used as a synchronous decade counter or divider circuit.
2. It is used in hardware logic design to create complicated Finite states machine. ex: ASIC and FPGA design.
3. The 3 stage Johnson counter is used as a 3 phase square wave generator which produces 1200 phase shift.
4. It is used to divide the frequency of the clock signal by varying their feedback.