

# DIGITAL ELECTRONICS



Dr. Pradip Kumar Sahu  
Associate Professor,  
Department of Information Technology

# Lecture of Module 4

## **Sequential Circuits (Latch/Flip Flop)**

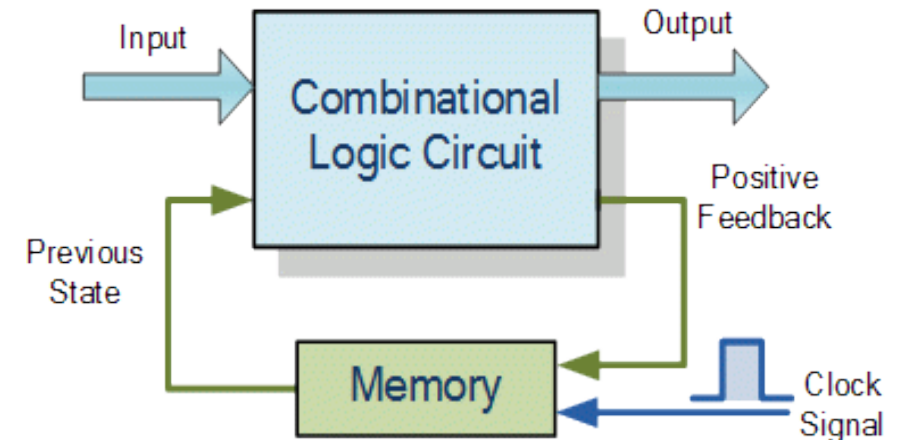
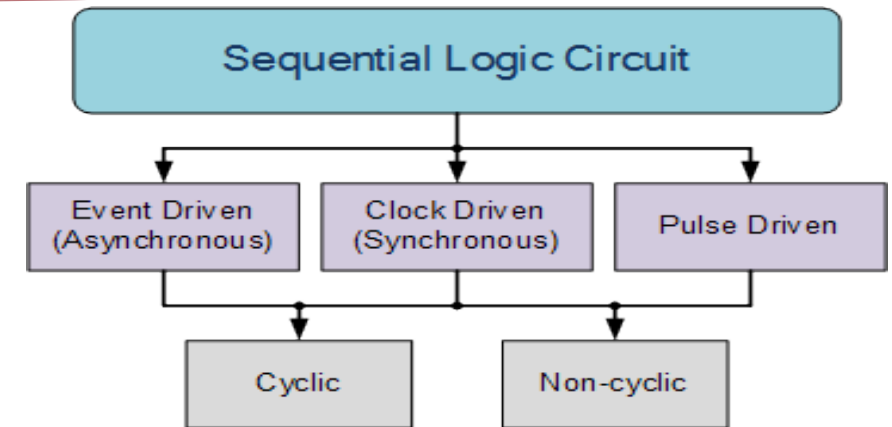
# Overview

- ▶ **Introduction**
- ▶ **Latch**
- ▶ **Flip Flop**
- ▶ **Triggering of Flip Flop**
- ▶ **SR, D, JK, T Flip Flop**
- ▶ **Master Slave Flip Flop**
- ▶ **Characteristic Table and Characteristic Equation**
- ▶ **Excitation Table**

# Sequential Logic Circuits

The output state of a “sequential logic circuit” is a function of the following states, the “present input”, the “past input” and/or the “past output”. *Sequential Logic circuits* remember these conditions and stay fixed in their current state until the next clock signal changes.

Sequential logic circuits are generally termed as *two state* or **Bi-stable devices**. Outputs set in one of two basic states, a logic level “1” or a logic level “0” and will remain “latched” (hence the name is latch) indefinitely in this current state or condition until some other input, that is, trigger pulse or signal is applied which will cause the bi-stable to change its state once again.



# Sequential Logic: Concept

- ▶ Sequential Logic circuits remember past inputs and/or past circuit state.
- ▶ Outputs from the system are “fed back” as new inputs.
- ▶ The storage elements are circuits that are capable of storing binary information: Memory.

The basic sequential circuit elements can be divided in two categories:

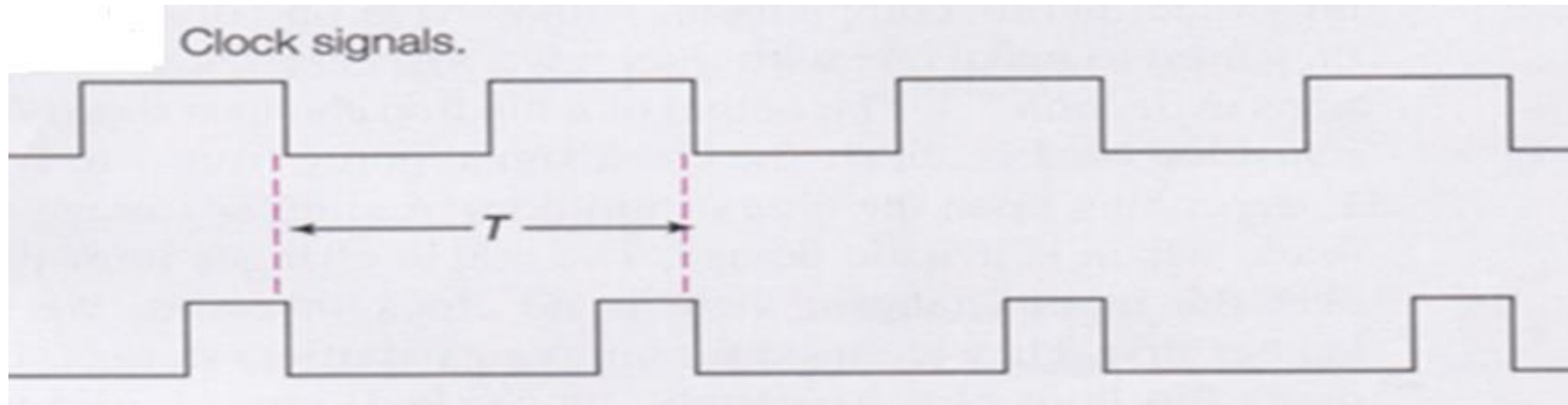
- ▶ Level-sensitive (Latches)
  - ▶ High-level sensitive
  - ▶ Low-level sensitive
- ▶ Edge-triggered (Flip-flops)
  - ▶ Rising (positive) edge triggered
  - ▶ Falling (negative) edge triggered
  - ▶ Dual-edge triggered

# Clock

Sequential circuits can be **Asynchronous** or **Synchronous**.

**Asynchronous** sequential circuits change their states and output values whenever a change in input values occurs. Circuit output can change at **any** time (**clock less**).

**Synchronous** sequential circuits change their states and output values at fixed points of time. This type of circuits achieves synchronization by using a timing signal called the **clock**.



Clock generator: Periodic train of clock pulses

# Memory Devices

**Latches:** A *latch* is a memory element whose excitation signals control the state of the device. A latch has two stages *set* and *reset*. *Set* stage sets the output to 1. *Reset* stage set the output to 0.

**Latches** are also called **level triggered** flip flops, because the change on the outputs will follow the changes of the inputs as long as the Enable input is set.

❖ This causes synchronization problems.

Solution: use latches to create flip-flops that can respond (update) only on specific times (instead of any time).

**Flip-flops:** A *flip-flop* is a memory device that has clock signals control the state of the device.

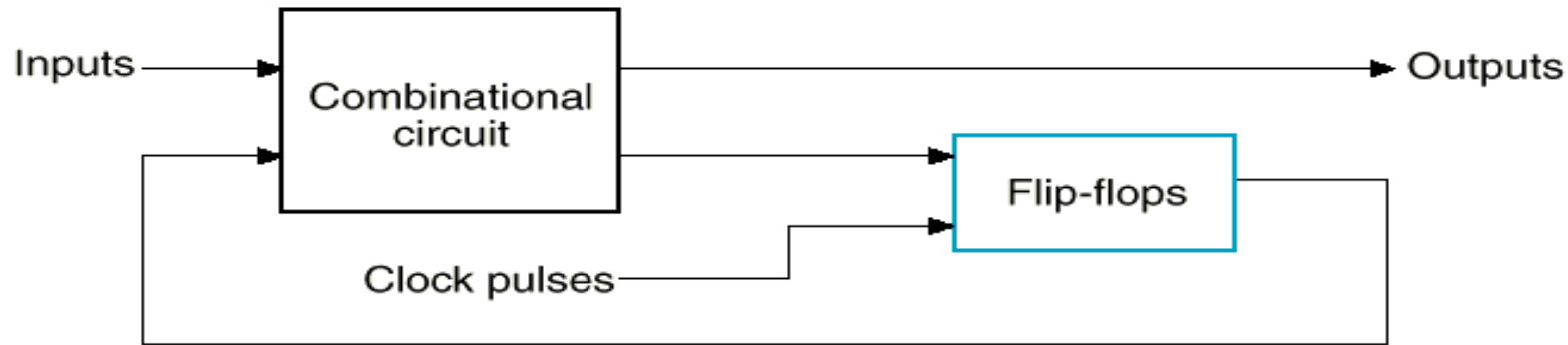
Flip Flops are **Edge triggered** that change there outputs only at the transition of the clock signal.

# Latch Vs. Flip Flop

Latches	Flip Flops
Latches are building blocks of sequential circuits and these can be built from logic gates	Flip flops are also building blocks of sequential circuits. But, these can be built from the latches.
Latch continuously checks its inputs and changes its output correspondingly.	Flip flop continuously checks its inputs and changes its output correspondingly only at times determined by clocking signal
The latch is sensitive to the duration of the pulse and can send or receive the data when the switch is on	Flipflop is sensitive to a signal change. They can transfer data only at the single instant and data cannot be changed until next signal change. Flip flops are used as a register.
It is based on the enable function input	It works on the basis of clock pulses
It is a level triggered, it means that the output of the present state and input of the next state depends on the level that is binary input 1 or 0.	It is an edge triggered, it means that the output and the next state input changes when there is a change in clock pulse whether it may a +ve or -ve clock pulse.



# Synchronous Sequential Circuits: Flip flops as state memory



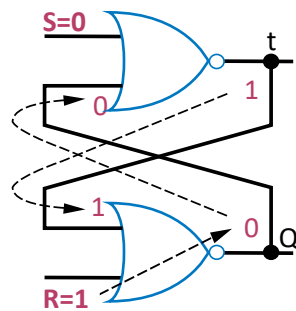
(a) Block diagram



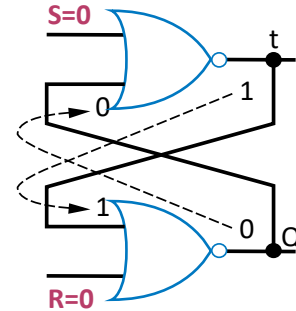
(b) Timing diagram of clock pulses

- The flip-flops receive their inputs from the combinational circuit and also from a clock signal with pulses that occur at fixed intervals of time, as shown in the timing diagram.

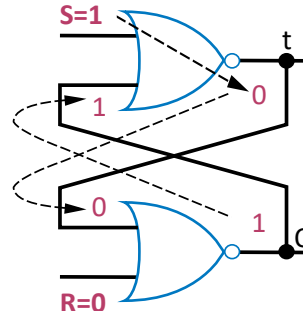
# S-R Latch(NOR version)



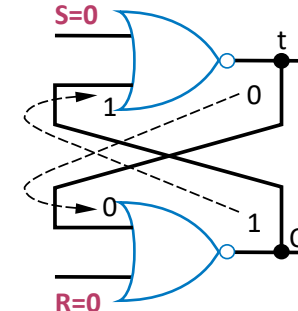
Reset



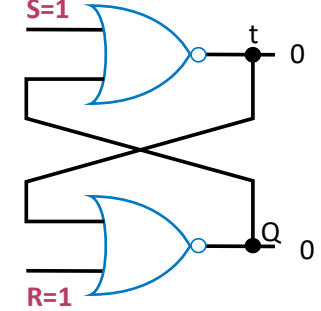
Last State



Set

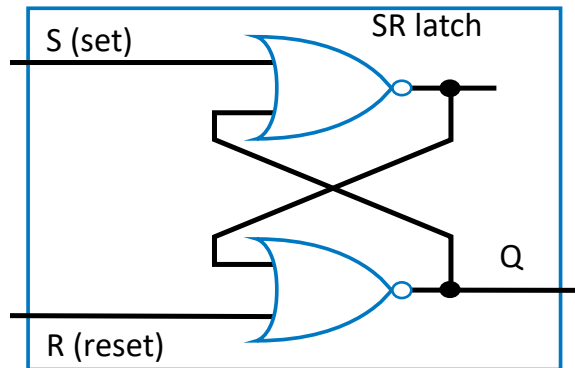
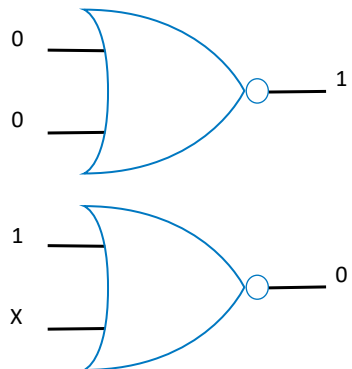


Last State



Forbidden

Recall...

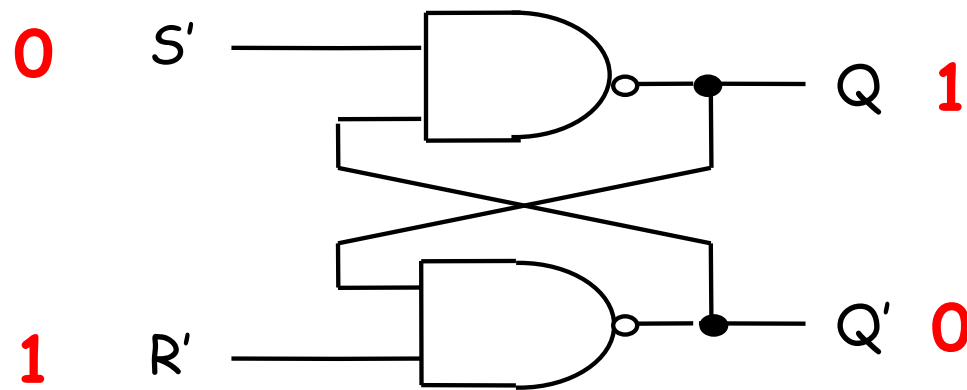


Time



R	S	Q	$\overline{Q}$	Comment
0	0	?	?	Stored state unknown
0	1	1	0	"Set" Q to 1
0	0	1	0	Now Q "remembers" 1
1	0	0	1	"Reset" Q to 0
0	0	0	1	Now Q "remembers" 0
1	1	0	0	Both go low
0	0	?	?	Unstable!

# S-R Latch(NAND version)



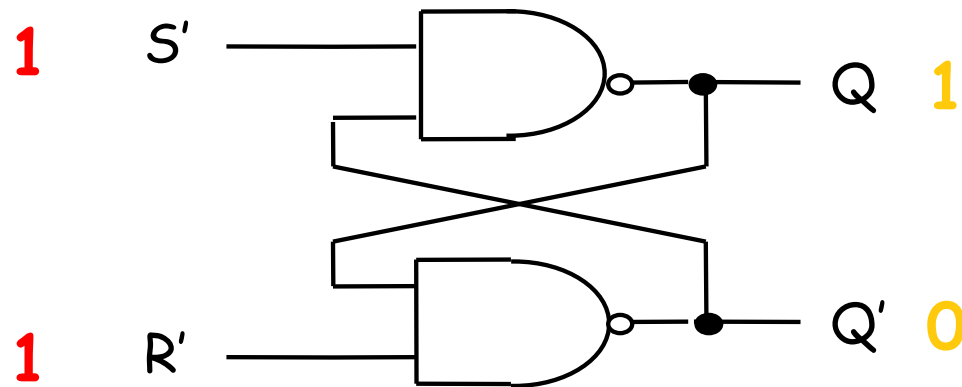
$S'$	$R'$	$Q$	$Q'$
0	0		
0	1	1	0
1	0		
1	1		

Set

Characteristic Table

X	Y	NAND
0	0	1
0	1	1
1	0	1
1	1	0

# S-R Latch(NAND version)

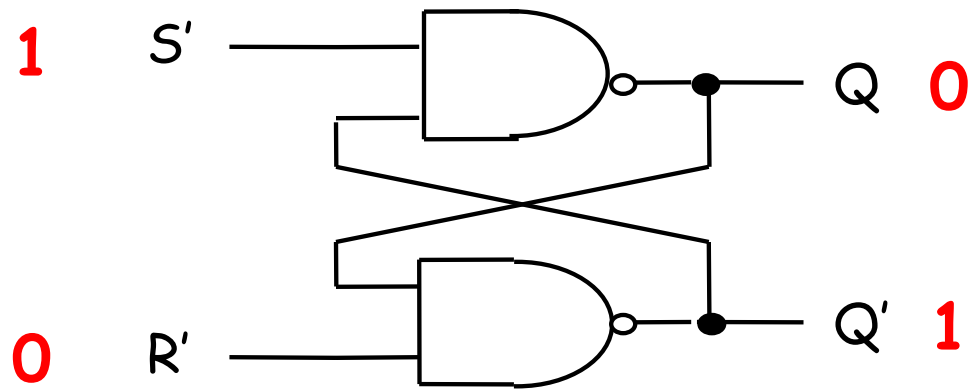


$S'$	$R'$	$Q$	$Q'$	
0	0			
0	1	1	0	Set
1	0			
1	1	1	0	Last State

Characteristic Table

$X$	$Y$	NAND
0	0	1
0	1	1
1	0	1
1	1	0

# S-R Latch(NAND version)

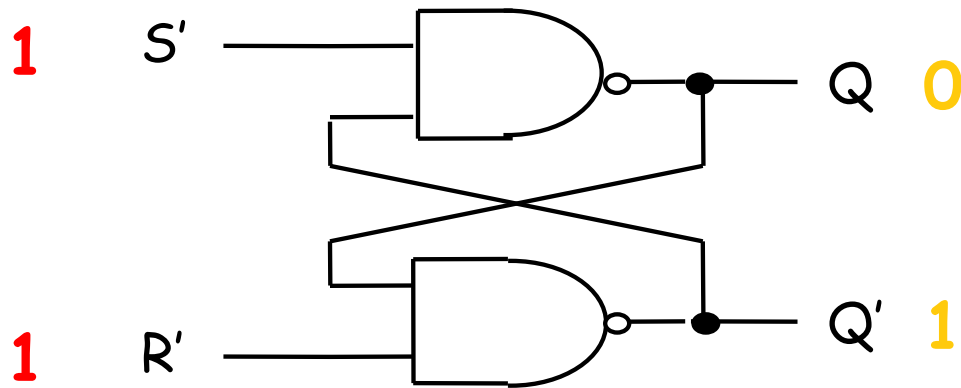


$S'$	$R'$	$Q$	$Q'$	
0	0			
0	1	1	0	Set
1	0	0	1	Reset
1	1	1	0	Last State

Characteristic Table

X	Y	NAND
0	0	1
0	1	1
1	0	1
1	1	0

# S-R Latch Flop(NAND version)

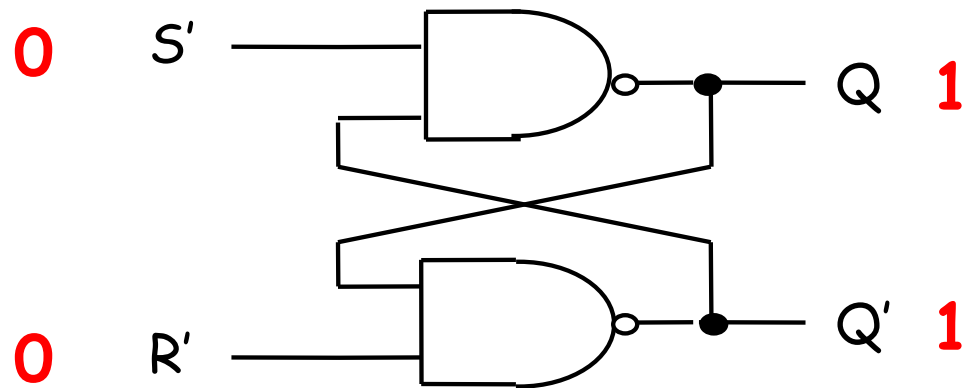


$S'$	$R'$	$Q$	$Q'$	
0	0			
0	1	1	0	Set
1	0	0	1	Reset
1	1	1	0	Last State
		0	1	Last State

Characteristic Table

$X$	$Y$	NAND
0	0	1
0	1	1
1	0	1
1	1	0

# S-R Latch(NAND version)



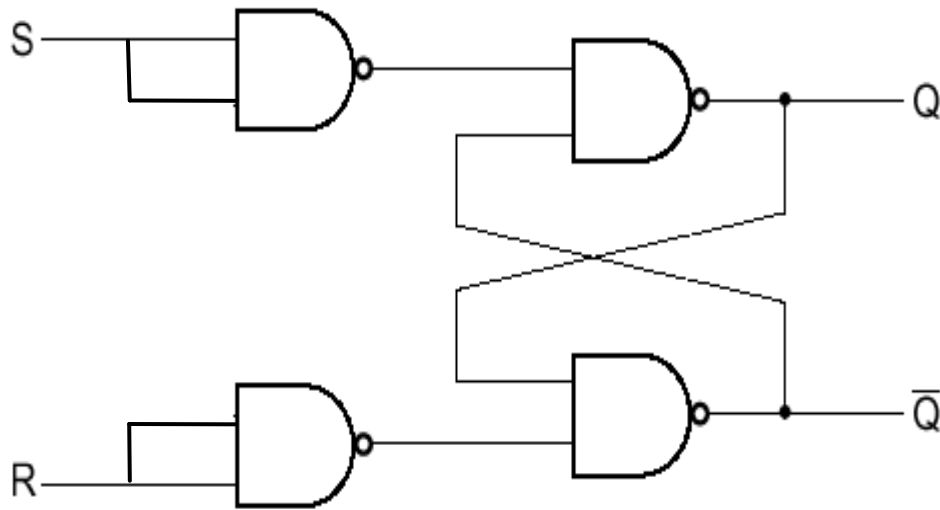
S'	R'	Q	Q'	
0	0	1	1	Forbidden
0	1	1	0	Set
1	0	0	1	Reset
1	1	1	0	Last State
		0	1	Last State

Characteristic Table

X	Y	NAND
0	0	1
0	1	1
1	0	1
1	1	0

S	R	Next state of Q
0	0	No change
0	1	Q = 0; Reset state
1	0	Q = 1; Set state
1	1	Undefined

# S-R Latch(NAND version)



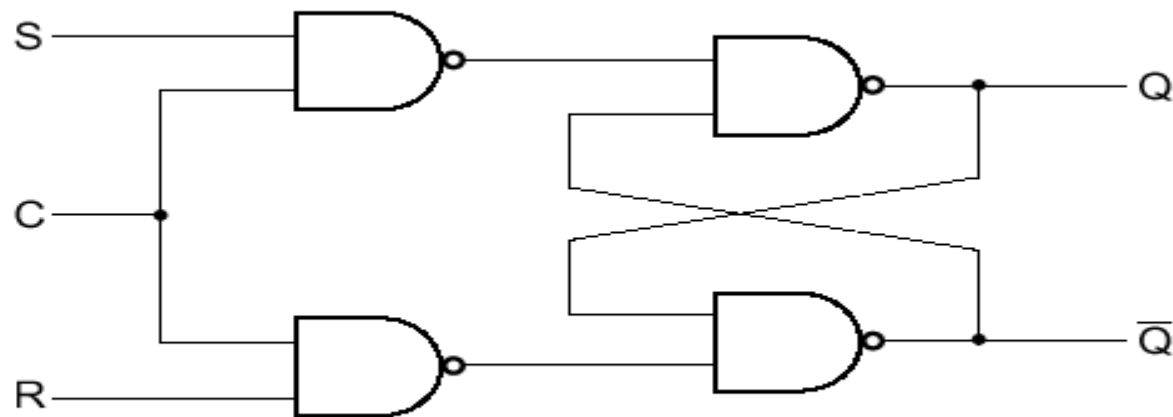
Logic diagram

S	R	Next state of Q
0	0	No change
0	1	Q = 0; Reset state
1	0	Q = 1; Set state
1	1	Undefined

Function table



# S-R Flip Flop with Clock signal



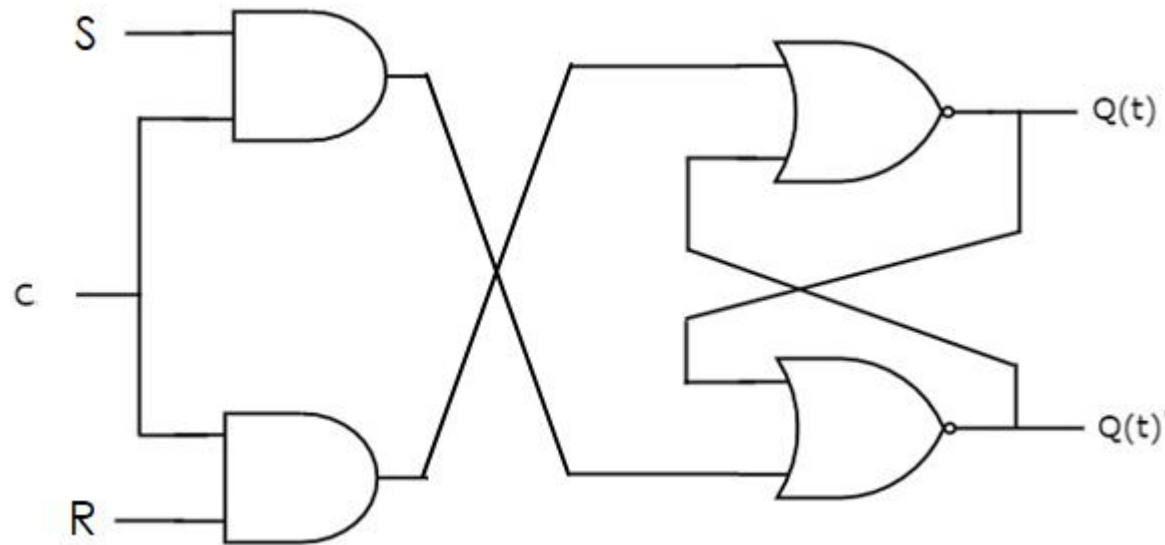
(a) Logic diagram

C	S	R	Next state of Q
0	X	X	No change
1	0	0	No change
1	0	1	Q = 0; Reset state
1	1	0	Q = 1; Set state
1	1	1	Undefined

(b) Function table

Latch is sensitive to input changes ONLY when C=1

# S-R Flip Flop with Clock signal



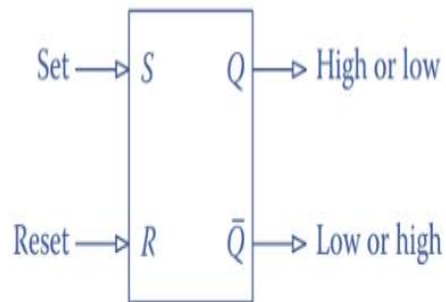
Logic diagram

C	S	R	Next state of Q
0	X	X	No change
1	0	0	No change
1	0	1	Q = 0; Reset state
1	1	0	Q = 1; Set state
1	1	1	Undefined

Function table

Latch is sensitive to input changes ONLY when C=1

# Characteristic Equation of S-R Flip Flop

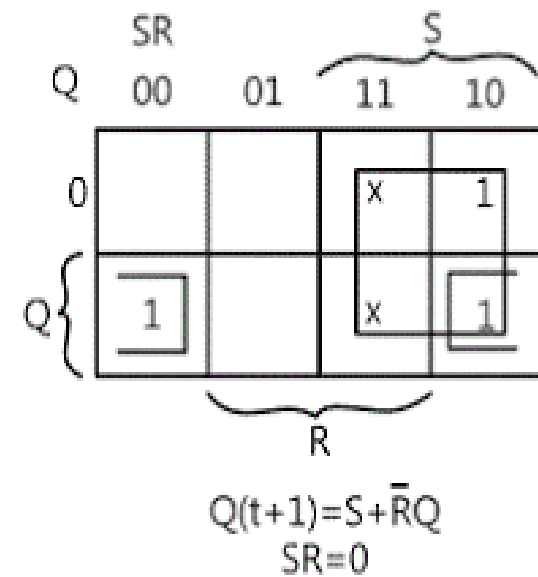


$R$	$S$	$Q$	$\bar{Q}$
0	0	$Q_0$	$\bar{Q}_0$
0	1	1	0
1	0	0	1
1	1	Not allowed	

$Q_0$  and  $\bar{Q}_0$   
are initial conditions  
(not defined)

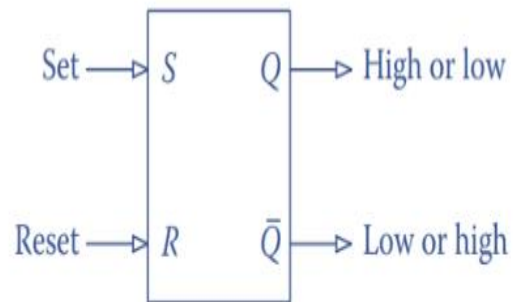
$Q$	$S$	$R$	$Q(t+1)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	Negative Status
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	Negative Status

Characteristic Table



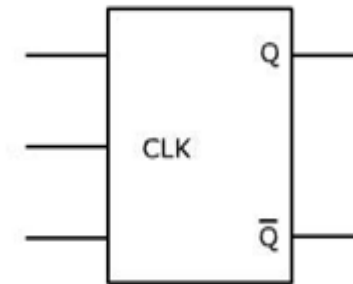
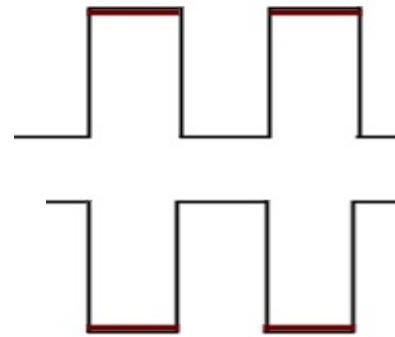
Characteristic Equation

# Triggering of Flip Flop



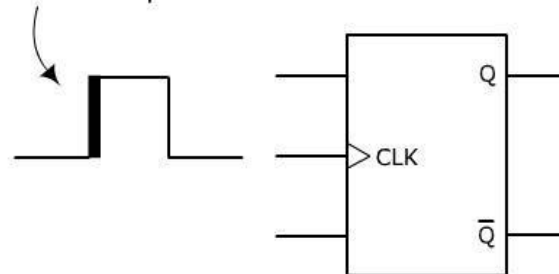
$R$	$S$	$Q$	$\bar{Q}$
0	0	$Q_0$	$\bar{Q}_0$
0	1	1	0
1	0	0	1
1	1	Not allowed	

$Q_0$  and  $\bar{Q}_0$   
are initial conditions  
(not defined)



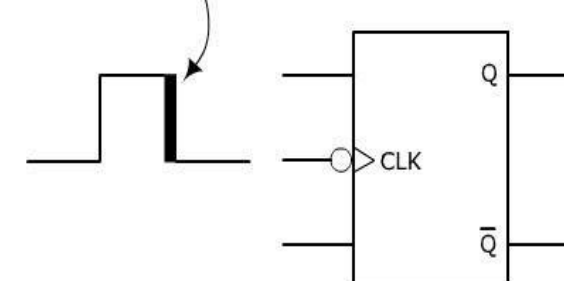
**Level Triggering**

Triggers on this edge  
of the clock pulse



**Positive Edge Triggering**

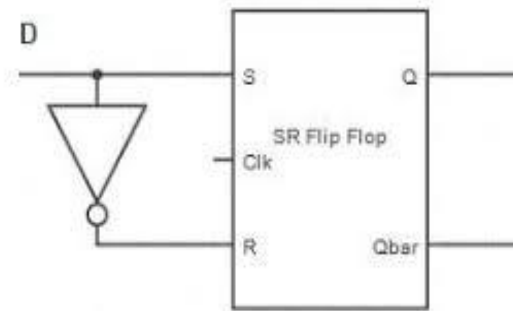
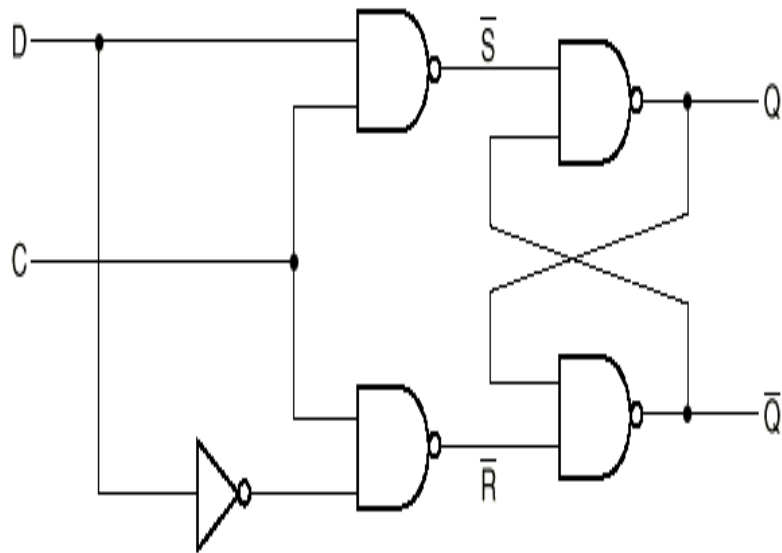
Triggers on this edge  
of the clock pulse



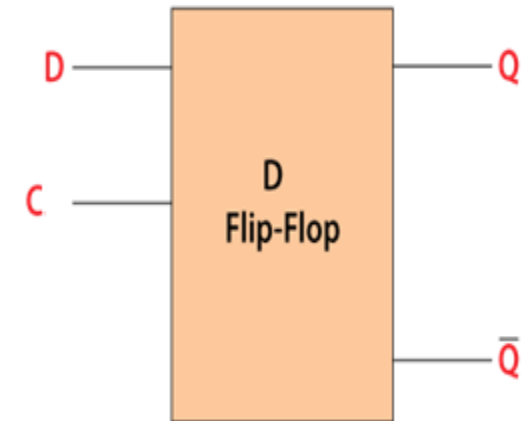
**Negative Edge Triggering**

# D Flip Flop

- One way to eliminate the undesirable indeterminate state in the RS flip flop is to ensure that inputs S and R are never 1 simultaneously.



C	D	Next state of Q
0	X	No change
1	0	Q = 0; Reset state
1	1	Q = 1; Set state



# Characteristic Equation of D Flip Flop

**Characteristics table**

Q <sub>n</sub>	D	Q(n+1)
0	0	0
0	1	1
1	0	0
1	1	1

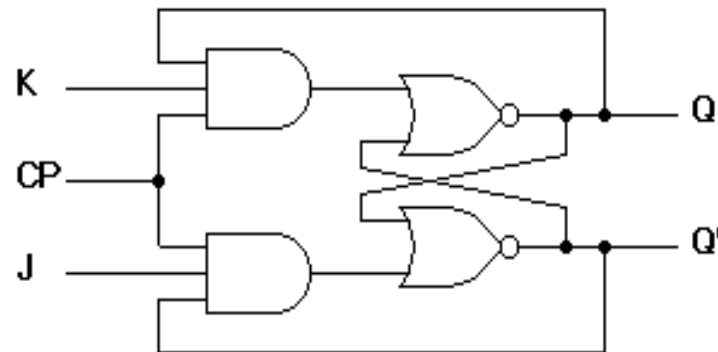
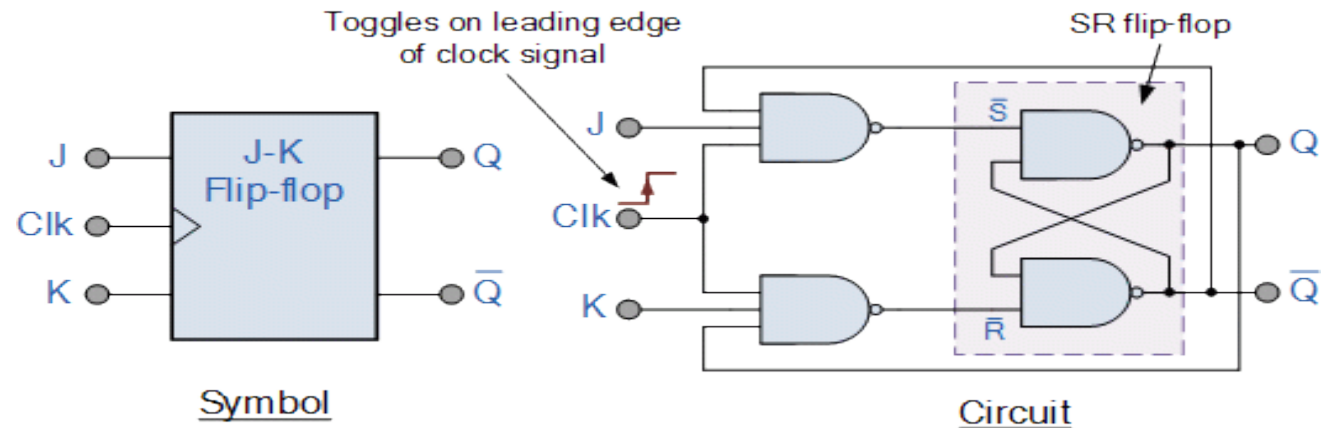
Q	D	
	0	1
0	0	1
1	0	1

$$Q(n+1) = D$$

Characteristic Equation

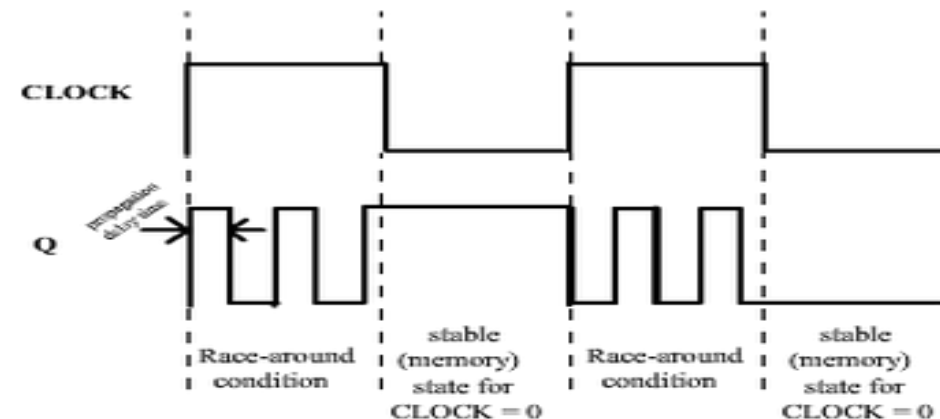
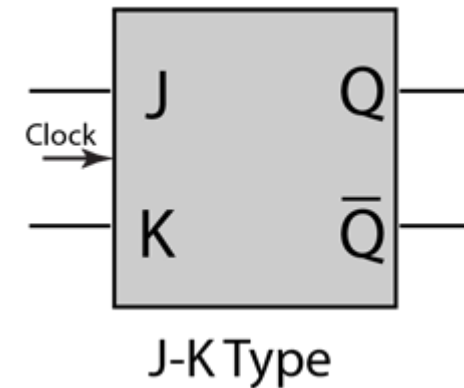
# J-K Flip Flop

- ▶ In SR Flip Flop  $S=R=1$  should be avoided.
- ▶ To overcome that JK Flip Flop developed.
- ▶ Both the S and the R inputs of the previous SR bistable have now been replaced by two inputs called the J and K inputs respectively after its inventor **Jack Kilby**. Then this may equates to:  $J = S$  and  $K = R$ .
- ▶ When  $J=0, K=0$ , no change in state.
- ▶ When  $J=0, K=1$ , Q will reset.
- ▶ When  $J=1, K=0$ , Q will set.
- ▶ When  $J=1, K=1$ , Toggle *i.e*  $Q'_n$



Clk	J	K	Q	Q'	State
1	0	0	Q	Q'	No change in state
1	0	1	0	1	Resets Q to 0
1	1	0	1	0	Sets Q to 1
1	1	1	-	-	Toggles

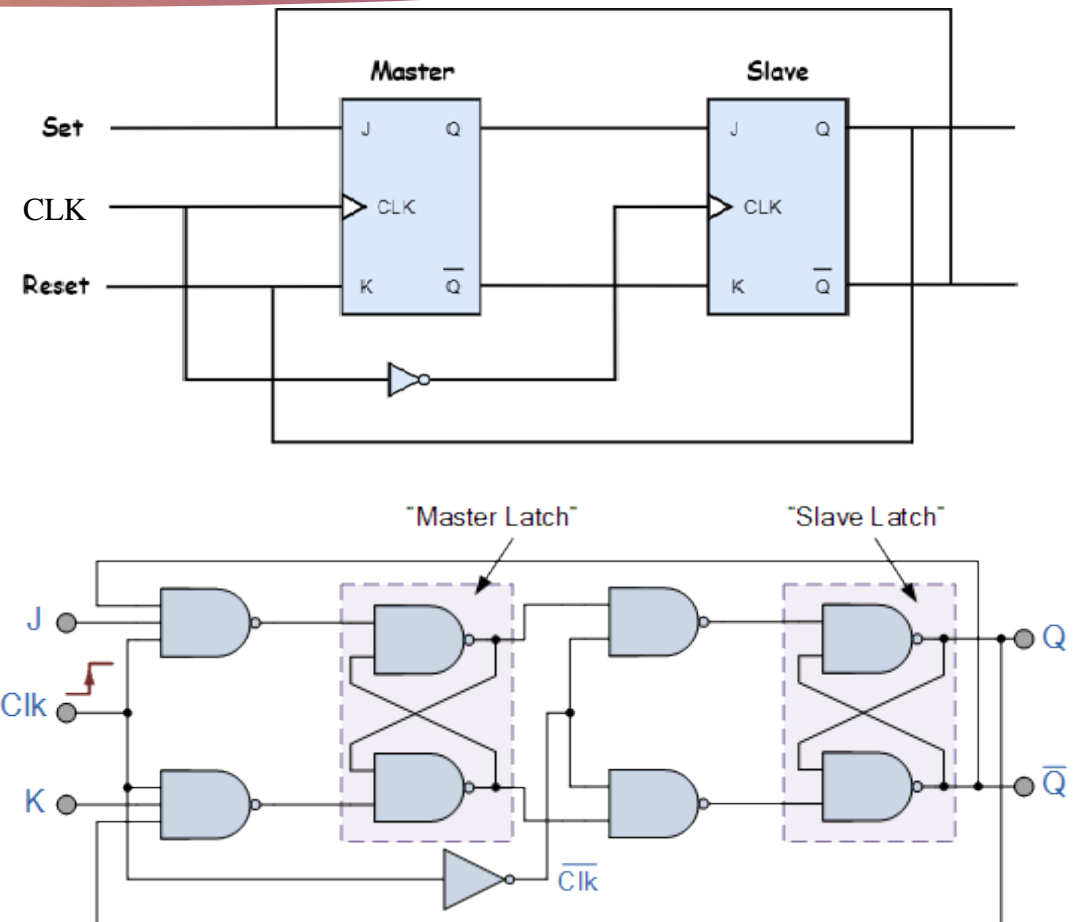
- ▶ When  $J=1$ ,  $K=1$ , Toggle *i.e*  $Q'_n$
- ▶ For JK flip-flop if J, K and Clock are equal to 1 the state of flip-flop keeps on toggling which leads to uncertainty in determining the output of the flip-flop. This problem is called **Race around condition**.
- ▶ This can be avoided by
  - ❖ Using Edge triggering of JK Flip Flop
  - ❖ Enhancing the propagation delay
  - ❖ Using Master-Slave Flip Flop





# Master-Slave Flip Flop

- ▶ Master-slave flip flop is designed using two separate flip flops. Out of these, one acts as the master and the other as a slave.
- ▶ The J-K flip flops are presented in a series connection.
- ▶ The output of the master J-K flip flop is fed to the input of the slave J-K flip flop.
- ▶ The output of the slave J-K flip flop is given as a feedback to the input of the master J-K flip flop.
- ▶ The clock pulse [Clk] is given to the master J-K flip flop and it is sent through a NOT Gate and thus inverted before passing it to the slave J-K flip flop.
- ▶ It avoids the race around condition of J-K Flip Flop

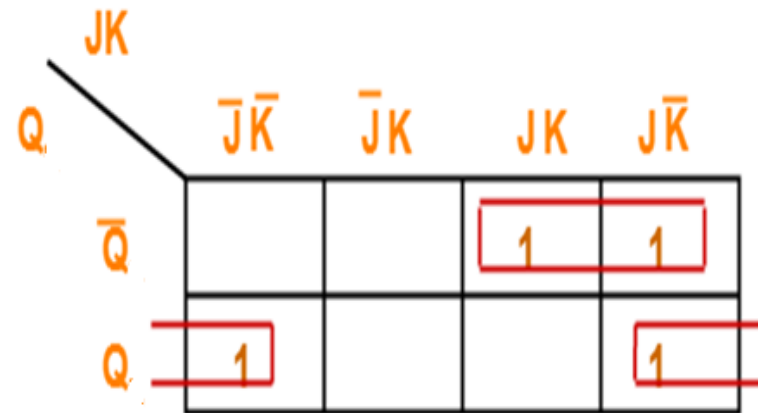


# Characteristic Equation of J-K Flip Flop

Characteristic Table

Q	J	K	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

JK flip-flop



K Map

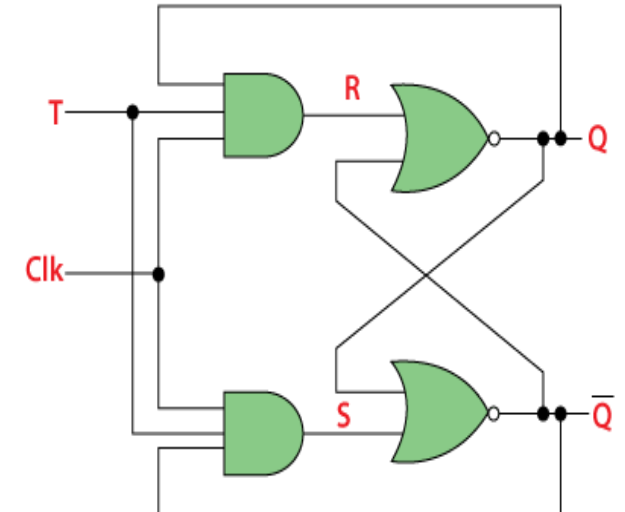
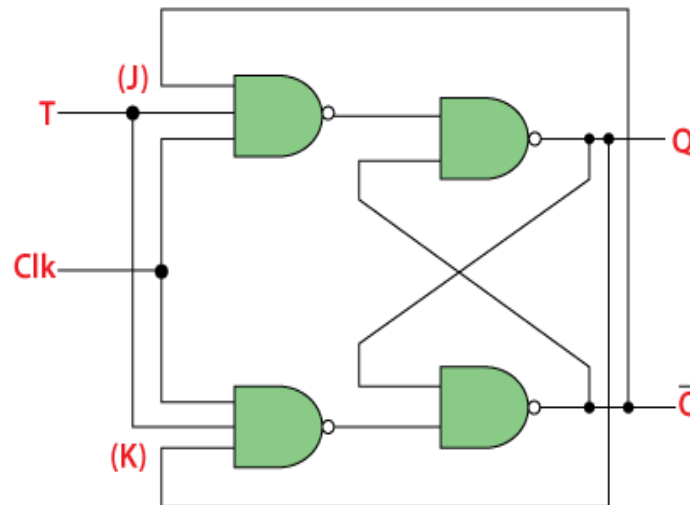
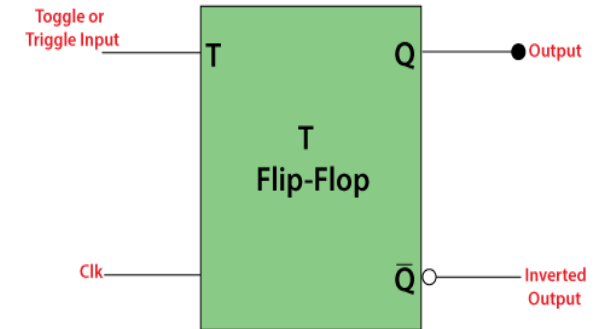
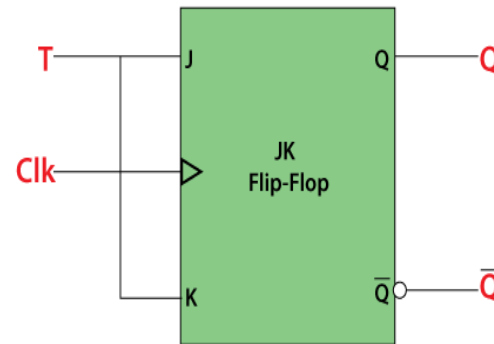
$$Q(t+1) = JQ' + K'Q$$

Characteristic Equation

# T Flip Flop

- ▶ We can construct the "T Flip Flop" by making changes in the "JK Flip Flop".
- ▶ The "T Flip Flop" has only one input, which is constructed by connecting the input of JK Flip Flop.
- ▶ This single input is called T.
- ▶ Sometimes the "T Flip Flop" is referred to as single input "JK Flip Flop".
- ▶ In T flip flop, "T" defines the term "Toggle"

Q	T	Q (T+1)
0	0	0
0	1	1
1	0	1
1	1	0



# Characteristic Equation of T Flip Flop

Characteristic Table

Q	T	Q(t+1)
0	0	0
0	1	1
1	0	1
1	1	0

T flip-flop

Q \ T	0	1
	0	1
0	0	1
1	1	0

$$Q(t+1) = TQ' + T'Q$$

Characteristic Equation

# Excitation Table

- ▶ The characteristic table is useful for analysis and for defining the operation of flip flop.
- ▶ It specifies the next state when the inputs and present state are known.
- ▶ During design process we usually know the transition from present state to next state.
- ▶ So, we want to know the flip flop input conditions that will cause the required transition.
- ▶ Therefore, we need a table that lists the required inputs for a given change of states.
- ▶ Such table is called as Excitation Table.

SR Flip-flop				D Flip-flop		
Q(t)	Q(t+1)	S	R	Q(t)	Q(t+1)	D
0	0	0	X	0	0	0
0	1	1	0	0	1	1
1	0	0	1	1	0	0
1	1	X	0	1	1	1

JK flip-flop				T flip-flop		
Q(t)	Q(t+1)	J	K	Q(t)	Q(t+1)	T
0	0	0	x	0	0	0
0	1	1	x	0	1	1
1	0	x	1	1	0	1
1	1	x	0	1	1	0

Excitation Table for different Flip Flops

# Sequential Circuit Design

Example:

Count Sequences			Flip Flop Inputs		
A	B	C	TA	TB	TC
0	0	0	0	0	1
0	0	1	0	1	1
0	1	0	0	0	1
0	1	1	1	1	1
1	0	0	0	0	1
1	0	1	0	1	1
1	1	0	0	0	1
1	1	1	1	1	1

	B'C'	B'C	BC	BC'
A'			1	
A			1	

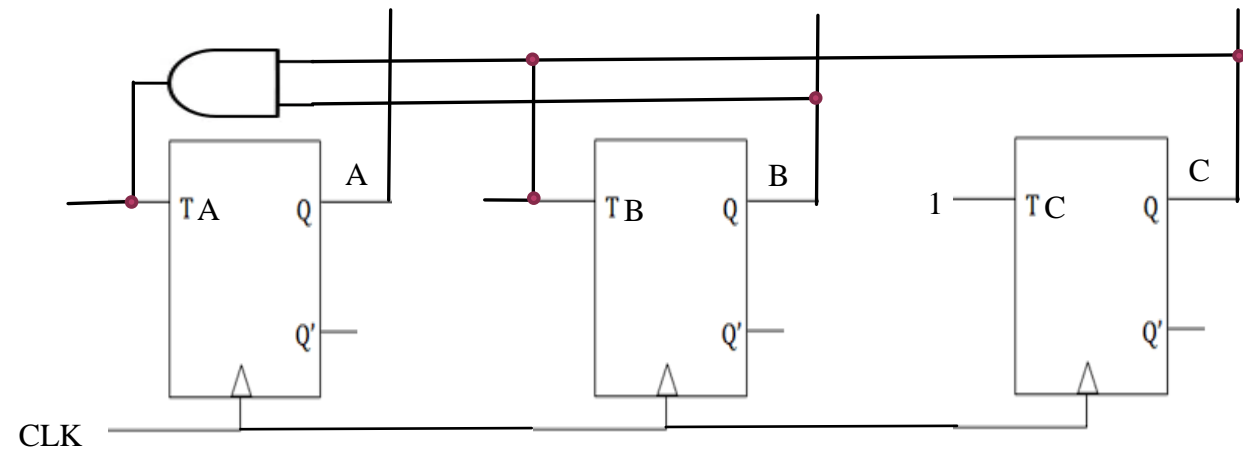
$$TA = BC$$

	B'C'	B'C	BC	BC'
A'		1	1	
A		1	1	

$$TB = C$$

	B'C'	B'C	BC	BC'
A'	1	1	1	1
A	1	1	1	1

$$TC = 1$$



# Sequential Circuit Design

## Example:

Let the state equations are:

$$A(t+1) = A'B'CD + A'B'C + ACD + AC'D'$$

$$B(t+1) = A'C + CD' + A'BC'$$

$$C(t+1) = B$$

$$D(t+1) = D'$$

The above equation can be rearranged in the form of characteristic equation of J-K flip flop.

Characteristic equation of J-K flip flop is

$$Q(t+1) = JQ' + K'Q$$

$$\begin{aligned} A(t+1) &= A'B'CD + A'B'C + ACD + AC'D' \\ &= (B'CD + B'C)A' + (CD + C'D')A \end{aligned}$$

$$\text{So, } J = B'CD + B'C = B'C$$

$$K = (CD + C'D')' = C'D + CD'$$

$$B(t+1) = A'C + CD' + A'BC'$$

$$= (A'C + CD')(B+B') + A'BC'$$

$$= (A'C + CD')B' + (A'C + CD')B + A'BC'$$

$$= (A'C + CD')B' + (A'C + CD' + A'C')B$$

$$\text{So, } J = A'C + CD'$$

$$K = (A'C + CD' + A'C')' = AC' + AD$$

$$C(t+1) = B = B(C+C') = BC' + BC$$

$$\text{So, } J = B$$

$$K = B'$$

$$D(t+1) = D' = (1)D' + (0)D$$

$$\text{So, } J = 1$$

$$K = 1$$

So, finally

$$JA = B'C$$

$$KA = C'D + CD'$$

$$JB = A'C + CD'$$

$$KB = AC' + AD$$

$$JC = B$$

$$KC = B'$$

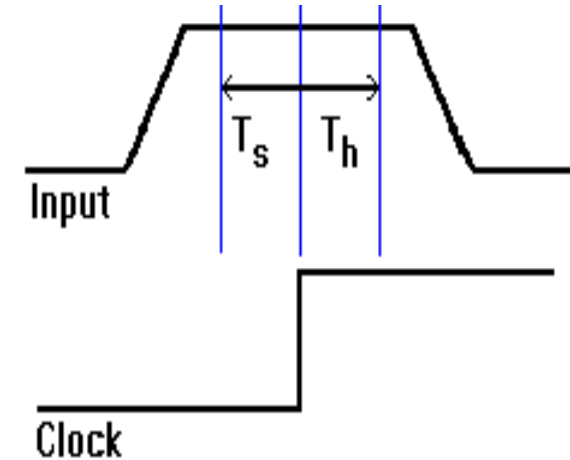
$$JD = 1$$

$$KD = 1$$

# Setup time and Hold time

The Clocking event can be either from low to high or from high to low. The input signal around clocking event must remain unchanged in order to have a correct effect on the outcome of the new state.

- ▶  $T_s$ : the minimum time interval preceding the clocking event the input signal must remain available and unchanged.
- ▶  $T_h$ : the minimum time interval after edge of the clocking event, the input signals must remain unchanged





# Applications

Flip flops will find their use in many of the fields in digital electronics. Flip flops are the main components of sequential circuits. Particularly, edge triggered flip flops are very resourceful devices that can be used in wide range of applications like storing of binary data and transferring binary data from one location to other etc. Some of the most common applications of flip flops are

- ▶ Shift Register
- ▶ Counter
- ▶ Storage Registers
- ▶ Memory
- ▶ Data transfer
- ▶ Frequency Dividers etc.

# Lecture of Module 4

## **Sequential Circuits (Shift Register)**

# Overview

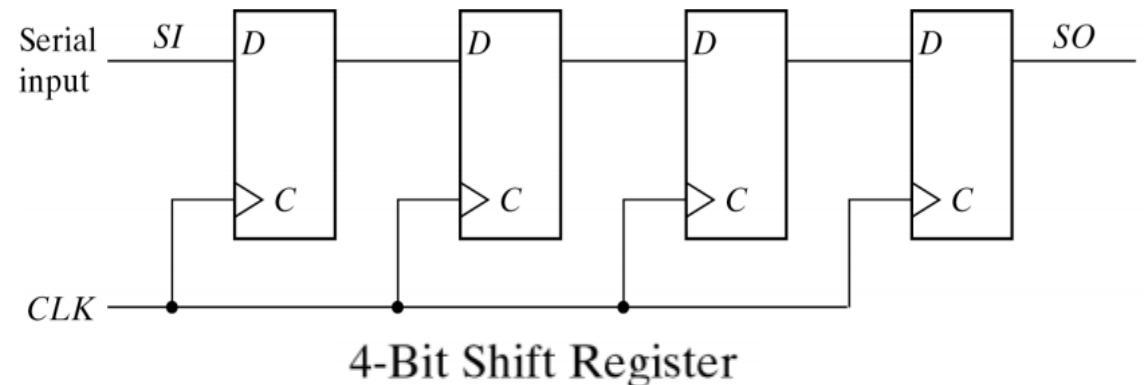
- ▶ **Register**
- ▶ **Shift Register**
- ▶ **Types of Shift Register**
- ▶ **Bidirectional Shift Register**
- ▶ **Universal Shift Register**
- ▶ **Typical ICs for Shift register**

# Register

- ▶ The Flip Flops are essential component in clocked sequential circuits.
- ▶ Circuits that include flip flops are usually classified by the function they perform.
- ▶ Two such circuits are **Registers** and **Counters**.
- ▶ An n-bit register consists of a group of n flip flops capable of storing n bits of binary information.
- ▶ So, **Register** is a collection of flip flops.
- ▶ A flip flop is used to store single bit **digital** data. For storing a large number of bits, the storage capacity is increased by grouping more than one flip flops.
- ▶ It is used to perform simple data **storage, movement, manipulation** and **processing** operations (e.g. **load, increment, shift, add**, etc.)
- ▶ The computer processes data by performing operations on registers, e.g. **ADD A, B** where A and B are the **registers**.
- ▶ A register capable of shifting its binary information in one or both direction is called a **Shift register**.
- ▶ All flip-flops receive common clock pulses, which activate the shift from one stage to the next.

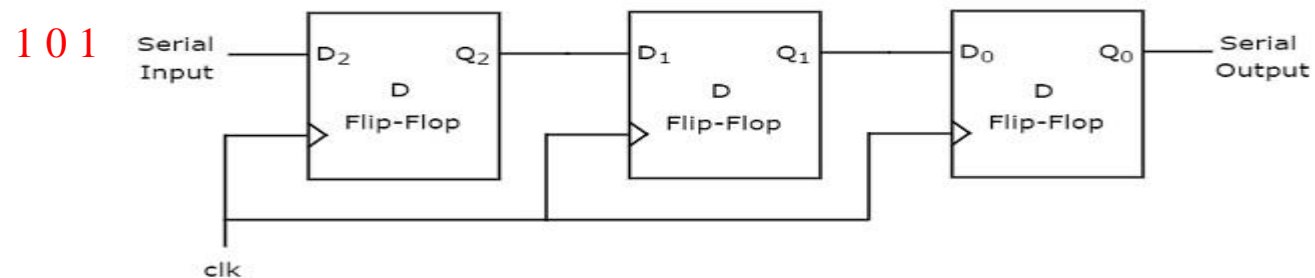
# Shift Register

- ▶ The simplest possible shift register is one that uses only flip-flops, as shown in Fig.
- ▶ Each clock pulse shifts the contents of the register one bit position to the right.
- ▶ The serial input determines what goes into the leftmost flip-flop during the shift.
- ▶ The serial output is taken from the output of the rightmost flip-flop.
- ▶ Following are the four types of shift registers based on applying inputs and accessing of outputs.
- ▶ Serial In – Serial Out (**SISO**) shift register
- ▶ Serial In – Parallel Out (**SIPO**) shift register
- ▶ Parallel In – Serial Out (**PISO**) shift register
- ▶ Parallel In – Parallel Out (**PIPO**) shift register



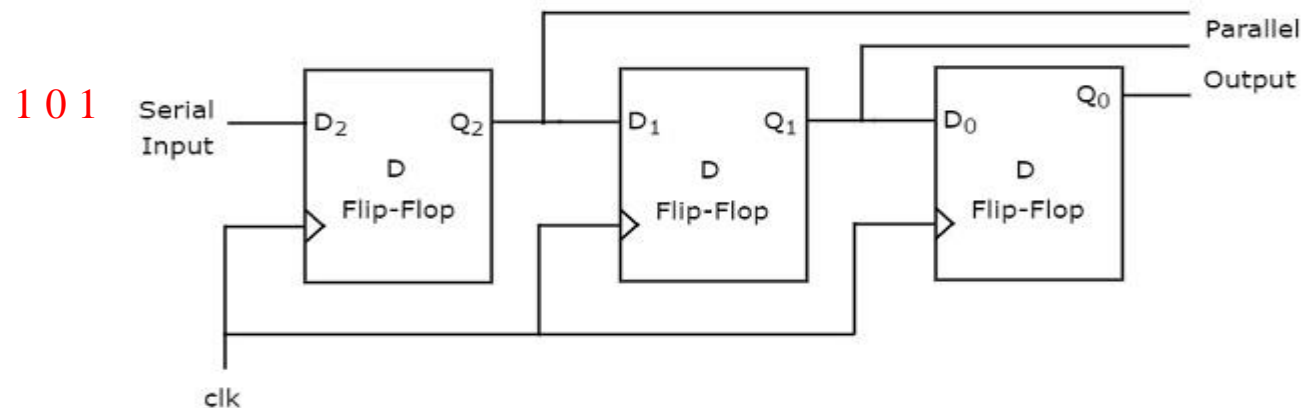
# Serial In – Serial Out (SISO) shift register

- ▶ The shift register, which allows serial input and produces serial output is known as Serial In – Serial Out (SISO) shift register.
- ▶ This block diagram consists of three D flip-flops, which are **cascaded**. That means, output of one D flip-flop is connected as the input of next D flip-flop.
- ▶ All these flip-flops are synchronous with each other since, the same clock signal is applied to each one.
- ▶ In this shift register, we can send the bits serially from the input of left most D flip-flop. Hence, this input is also called as **serial input**.
- ▶ For every positive edge triggering of clock signal, the data shifts from one stage to the next. So, we can receive the bits serially from the output of right most D flip-flop. Hence, this output is also called as **serial output**.



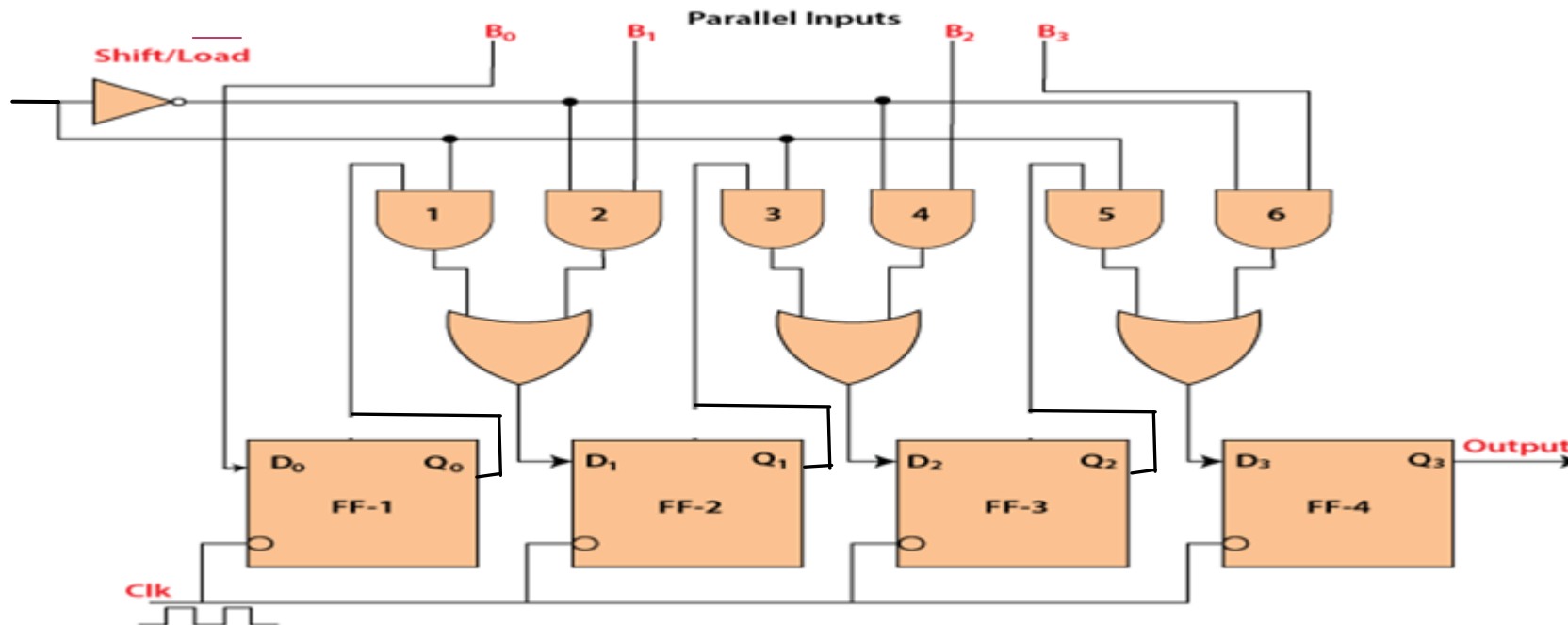
# Serial In – Parallel Out (SIPO) shift register

- ▶ The shift register, which allows serial input and produces parallel output is known as Serial In – Parallel Out (SIPO) shift register.
- ▶ In this shift register, we can send the bits serially from the input of left most D flip-flop. Hence, this input is also called as **serial input**.
- ▶ For every positive edge triggering of clock signal, the data shifts from one stage to the next.
- ▶ In this case, we can access the outputs of each D flip-flop in parallel. So, we will get **parallel outputs** from this shift register.



# Parallel In – Serial Out (PISO) shift register

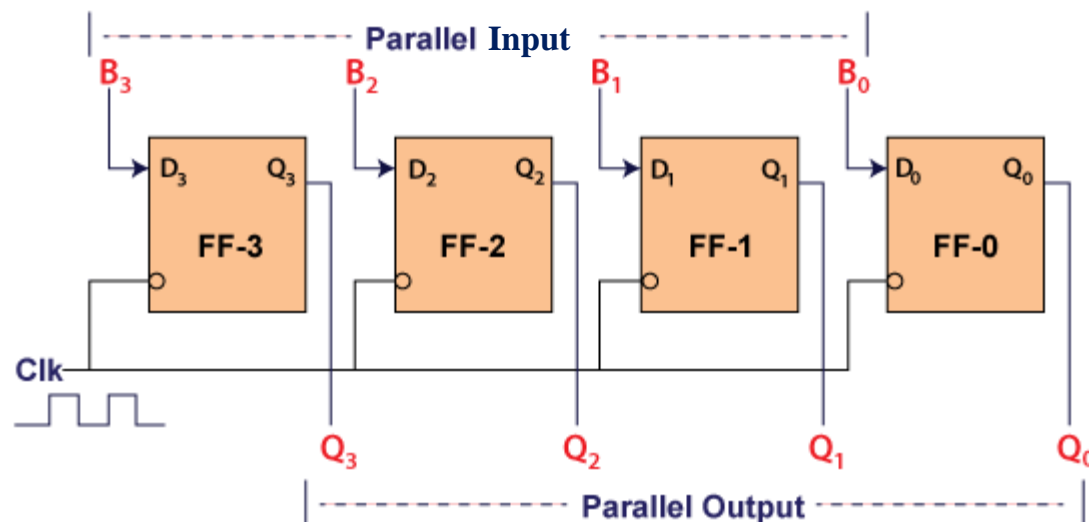
- ▶ In the "**Parallel In Serial Out**" register, the data is entered in a parallel way, and the outcome comes serially.
- ▶ The **shift mode** and the **load mode** are the two modes in which the "**PISO**" circuit works.



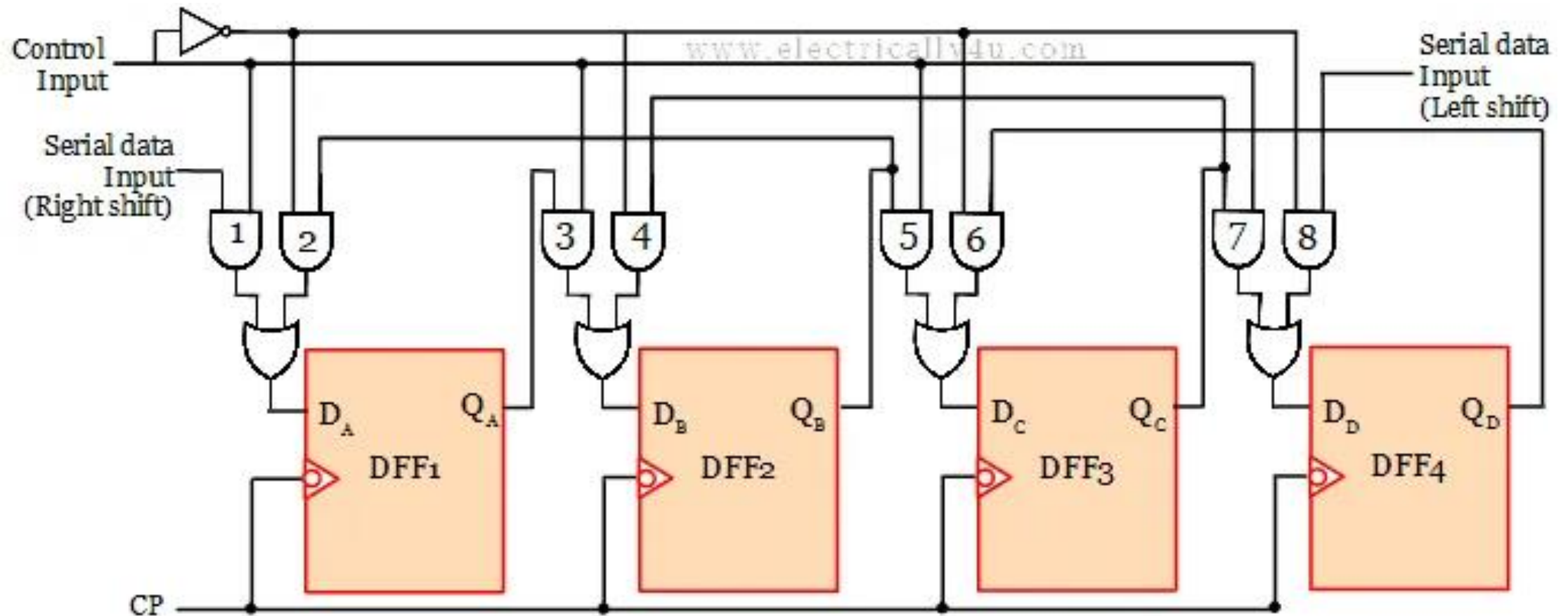


# Parallel In – Parallel Out (PIPO) shift register

- ▶ In "**Parallel In Parallel Out**", the inputs and the outputs come in a parallel way in the register.
- ▶ The inputs  $B_0$ ,  $B_1$ ,  $B_2$ , and  $B_3$ , are directly passed to the data inputs  $D_0$ ,  $D_1$ ,  $D_2$ , and  $D_3$  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.

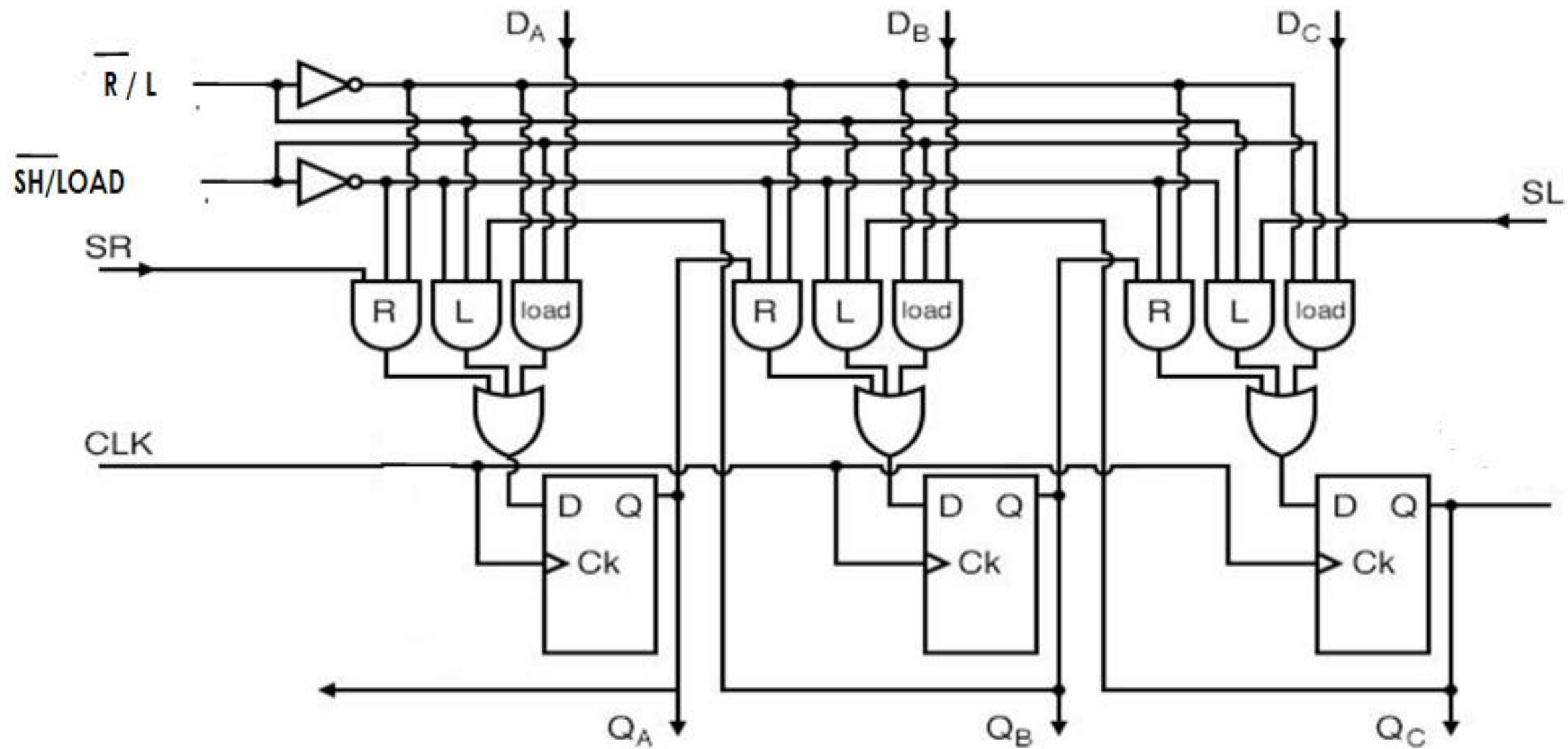


# Bidirectional Shift Register

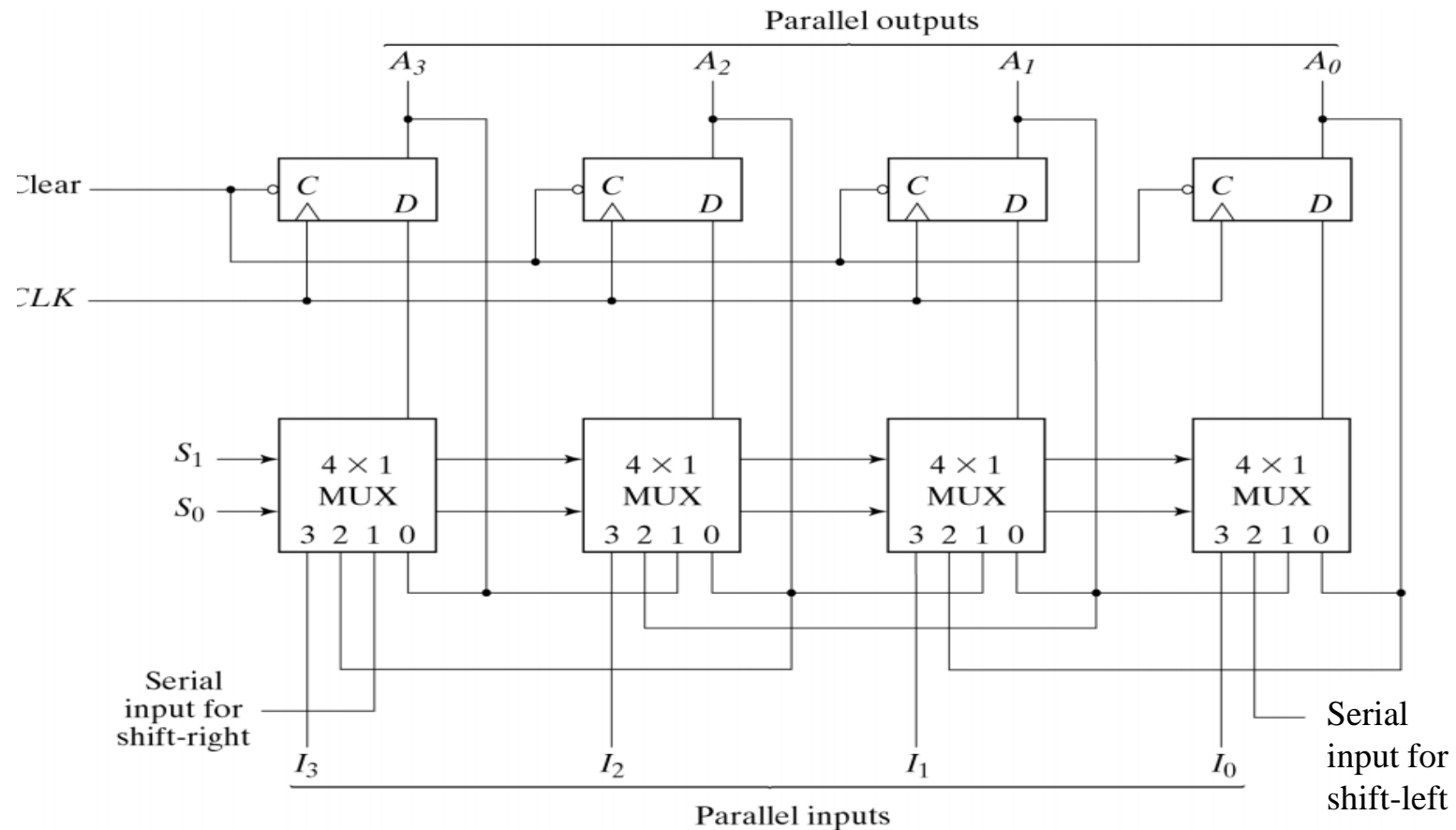


# Universal Shift Register

- ▶ A shift-right control to enable the shift operation and the serial input and output lines associated with the shift right.
- ▶ A shift-left control to enable the shift operation and the serial input and output lines associated with the shift left.
- ▶ A parallel-load control to enable a parallel transfer and the  $n$  input lines associated with the parallel transfer.
- ▶ If the Shift register has the capability of
  - Serial In – Serial Out
  - Serial In – Parallel Out
  - Parallel In – Serial Out
  - Parallel In – Parallel Outand act as Bidirectional shift register is referred as a universal shift register.



# Universal Shift Register using MUX



Mode Control		Register Operation
$S_1$	$S_0$	
0	0	No Change
0	1	Shift right
1	0	Shift Left
1	1	Parallel load

# Typical ICs for Shift register

- ▶ Commonly available SISO IC is 74HC595, which is 8-bit.
- ▶ Commonly available SIPO IC's include the standard 8-bit 74LS164, 74LS594.
- ▶ Commonly available PISO IC is 74HC166, which is 8-bit.
- ▶ Commonly available PIPO IC's include the standard 8-bit M54HC195, M74HC195.
- ▶ Today, there are many high speed bi-directional or “universal” type **Shift Registers** available such as the TTL 74LS194, 74LS195 or the CMOS 4035 which are available as 4-bit multi-function devices that can be used in either serial-to-serial, left shifting, right shifting, serial-to-parallel, parallel-to-serial, or as a parallel-to-parallel multifunction data register, hence their name “Universal”.
- ▶ The 74HC299 is an 8-bit Universal Shift register.
- ▶ The 74S299 is an 8-bit Universal Shift and Storage Register.