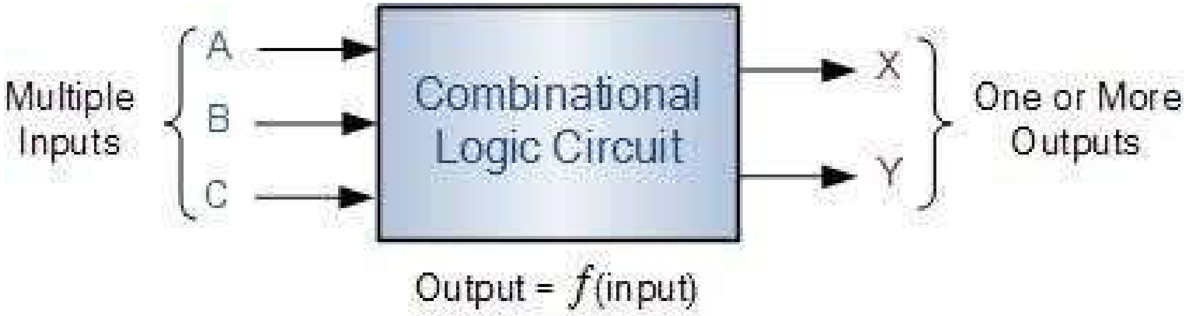
Unit - III

Sequential Circuits

Combinational Logic

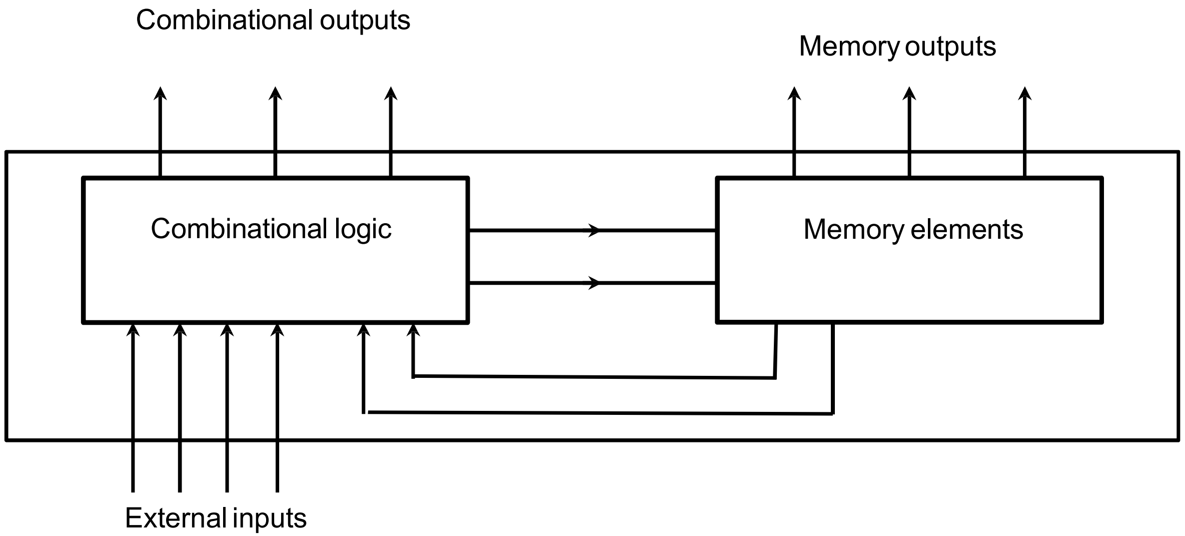
- The outputs depend only on the state of the inputs all of the time. Any change in the state of one of the inputs will ripple through the circuit immediately.
  - Examples of combinational logic are NAND and NOR gates, Inverters, and Buffers. These four logic gates form the basis of almost all combinational logic circuits as well as **flip flops**.



Sequential Logic

- Has **memory;** the circuit stores the result of the previous set of inputs. The current output depends on inputs **in the past as well as present inputs.**
  - The basic element in sequential logic is the **bistable latch** or **flip-flop**, which acts as a memory element for one bit of data.

$$Z = f(A_t, A_{t-1}, \dots, B_t, B_{t-1}, \dots, C_t, C_{t-1}, \dots, D_t, D_{t-1}, \dots,)$$



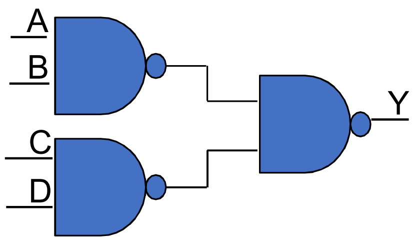
Two types:

- Finite memory system (only couple of previous inputs)
- Infinite memory system (all the past inputs)

Sequential circuit = Combinational logic + Memory Elements

Digital Circuits

Combinational Logic Circuit

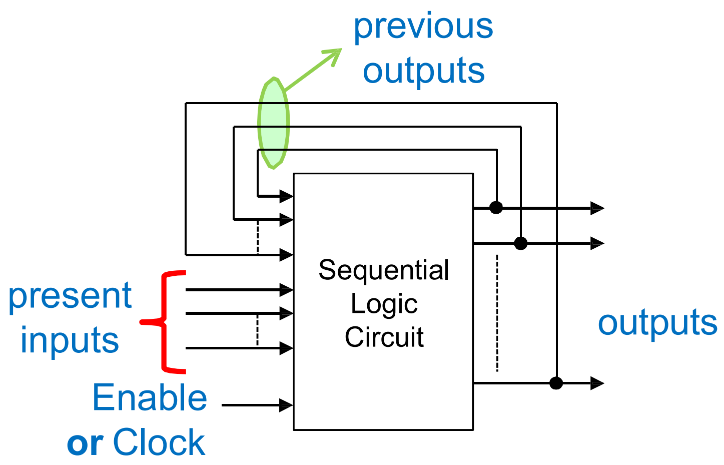


- output depends on present inputs

- consisted of logic gates and combinational logic subsystems (multiplexer, etc)

- Applications: ALU, PLDs

Sequential Logic Circuit



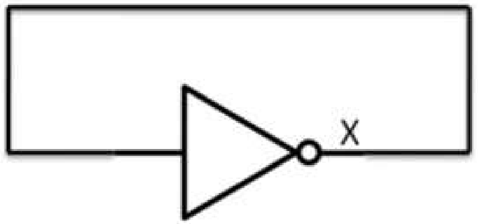
- output depends on present inputs and previous outputs

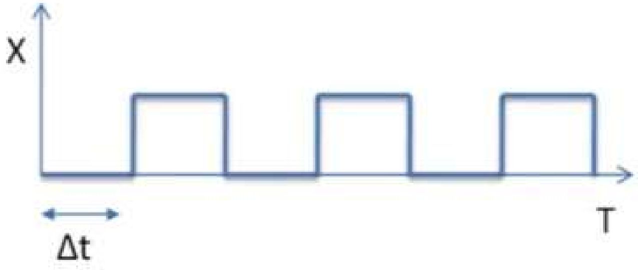
- enable or clock signal is used to synchronize the output change with time

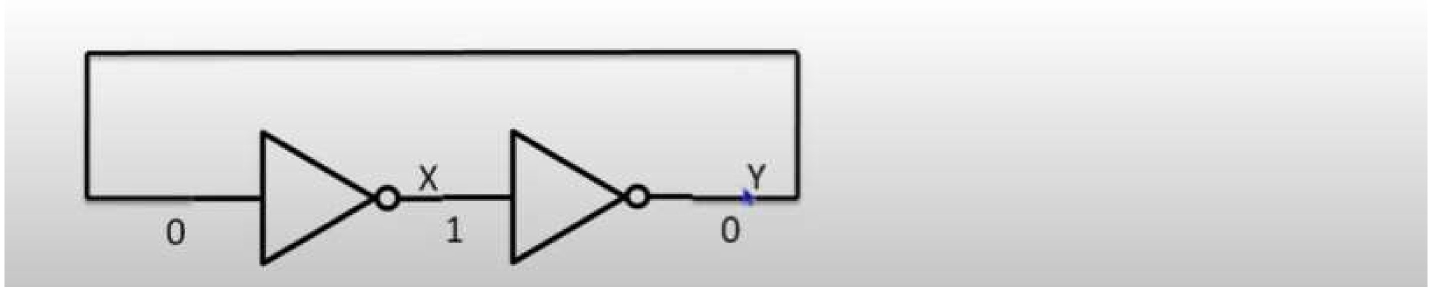
- Applications: Counters, Shift registers

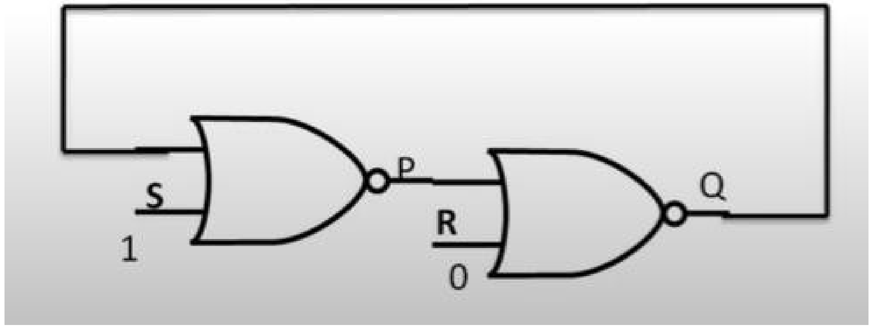
Design of basic storage element

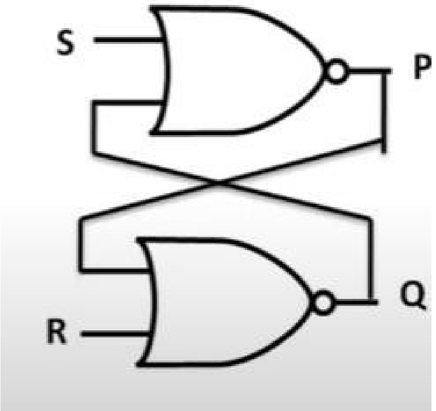
- Digital circuit with feedback





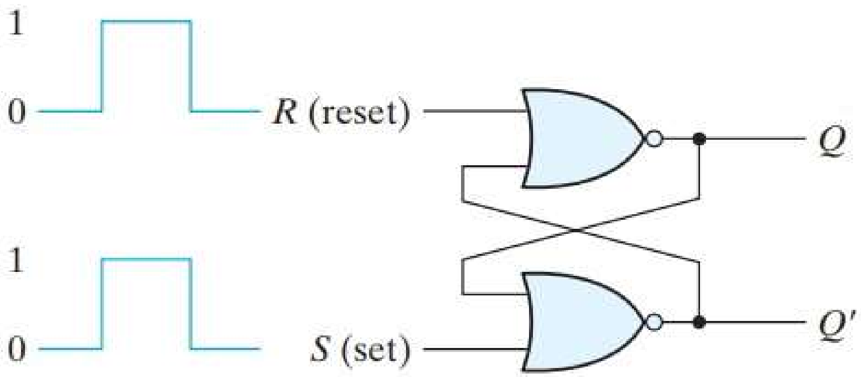






SR Latch (NOR Gates)

The SR latch is a circuit with two cross-coupled NOR gates or two cross-coupled NAND gates. It has two inputs labeled S for set and R for reset.



(a) Logic diagram

S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	(forbidden)

(b) Function table

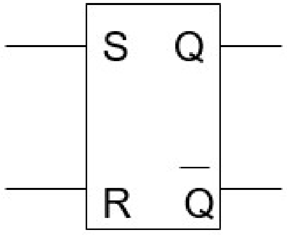


FIGURE 5.3
SR latch with NOR gates

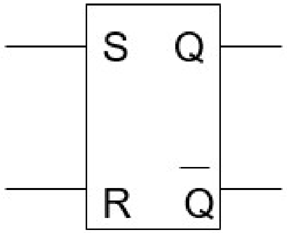
R	S	Q	Q^+
0	0	Q	Q
0	1	X	1
1	0	X	0
1	1	X	Invalid

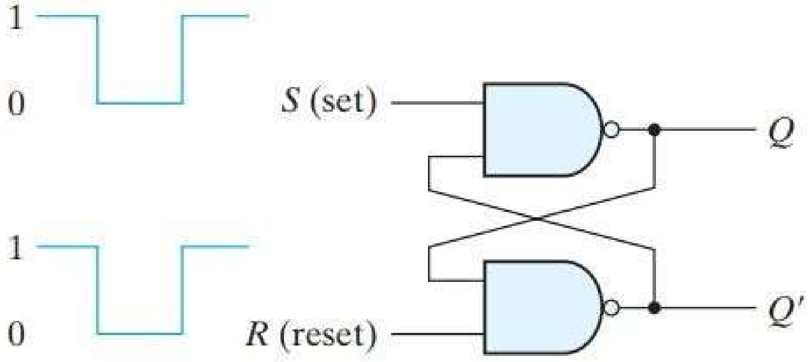
State table

Next state equation:
$Q^+ = S + R'Q$

SR Latch (NAND gates)

The SR latch is a circuit with two cross-coupled NOR gates or two cross-coupled NAND gates. It has two inputs labeled S for set and R for reset.





(a) Logic diagram

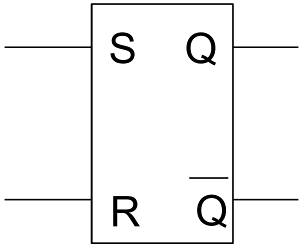
S	R	Q	Q′	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	(forbidden)

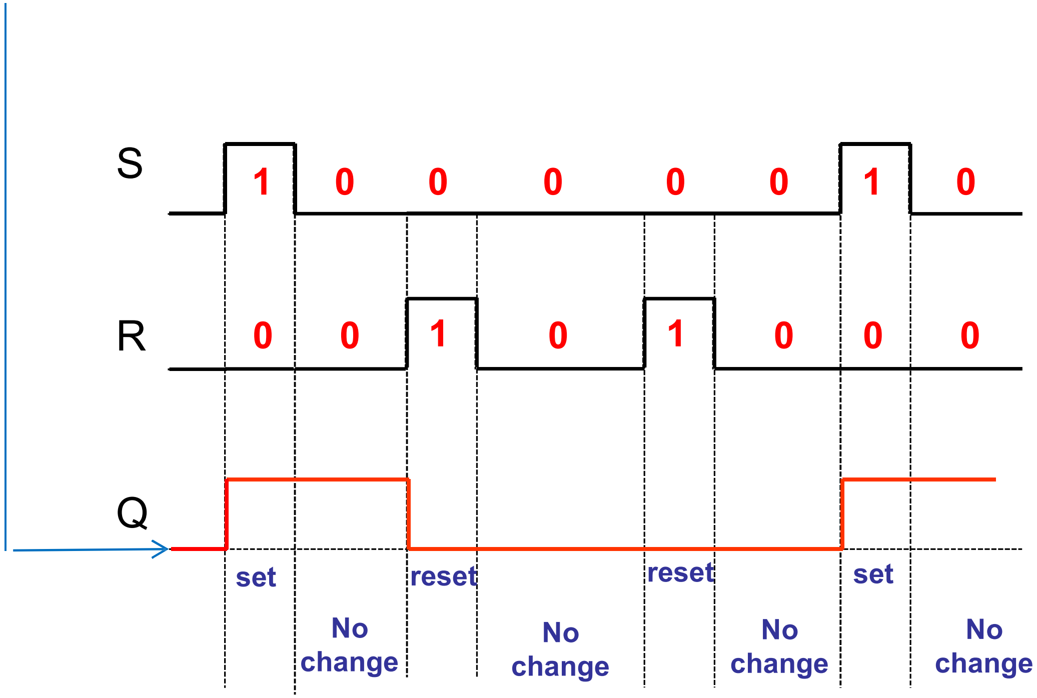
(b) Function table

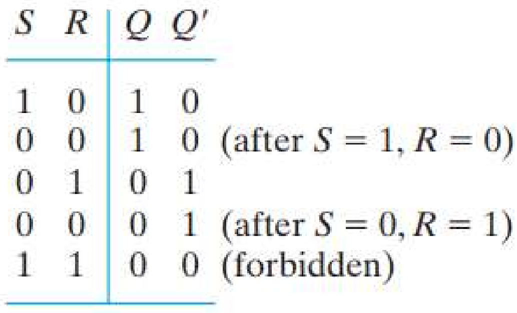
FIGURE 5.4
SR latch with NAND gates

SR Latch timing diagram

- Draw the output waveform for the S, R inputs shown below for an active high input SR latch
- Assume Q=0 initially





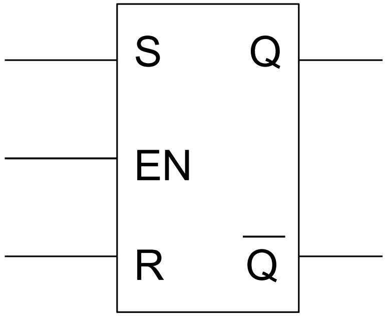


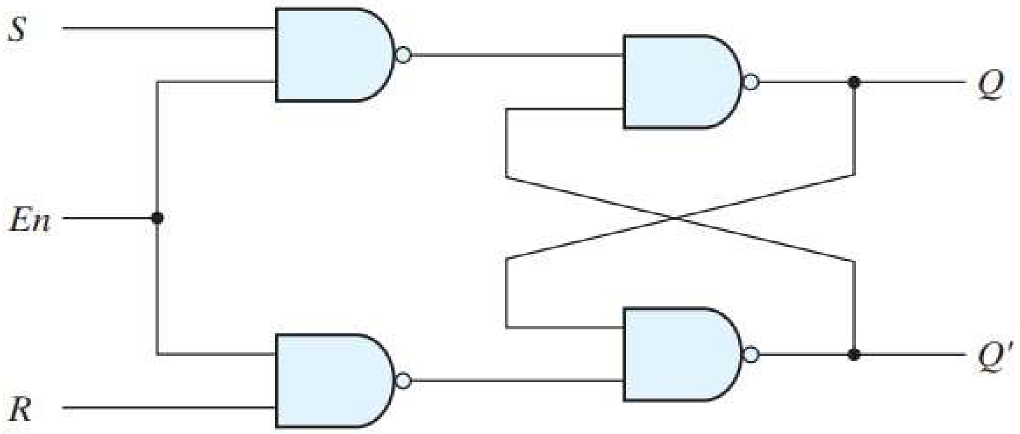
S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	(forbidden)

(b) Function table

Gated SR Latch

- The operation of the basic SR latch can be modified by providing an additional control input that determines when the state of the latch can be changed. In Fig. 5-5, it consists of the basic SR latch and two additional NAND gates.





(a) Logic diagram

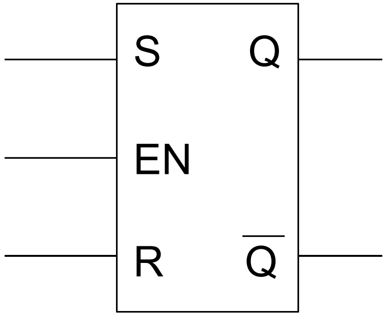
En	*S*	*R*	Next state of *Q*
0	X	X	No change
1	0	0	No change
1	0	1	*Q* = 0; reset state
1	1	0	*Q* = 1; set state
1	1	1	Indeterminate

(b) Function table

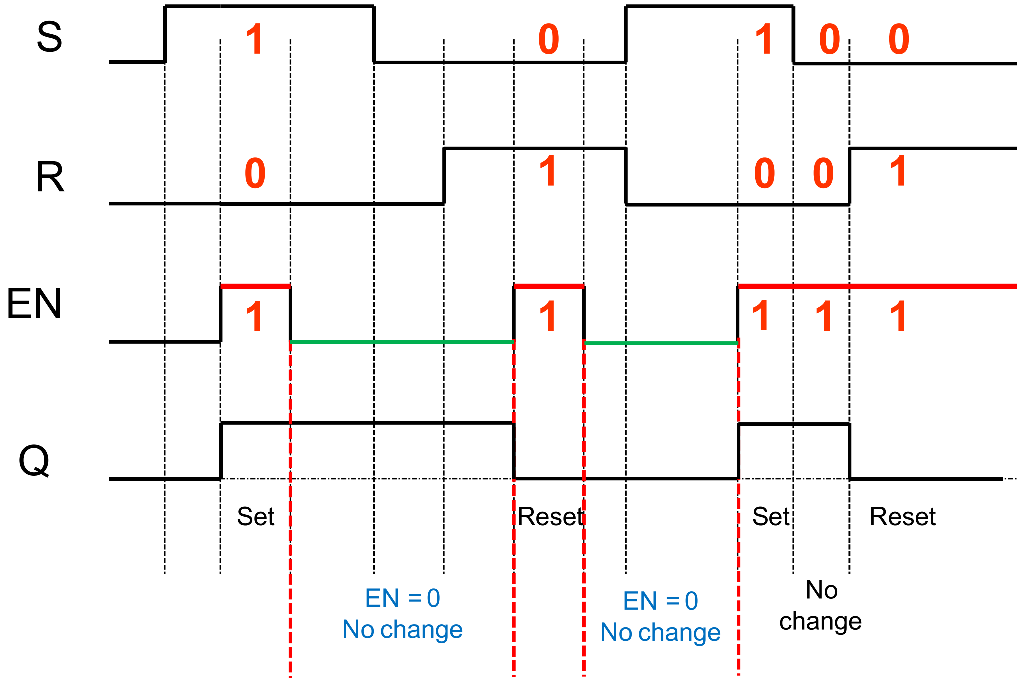
FIGURE 5.5
SR latch with control input

Gated SR Latch Timing diagram

1. Gated SR Latch

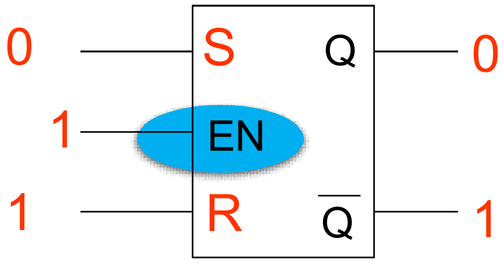


EN	S	R	Q
0	x	x	No change
1	0	0	No change
1	0	1	0 (Reset)
1	1	0	1 (Set)
1	1	1	Invalid



Assume Q=0 initially

Gated Latch

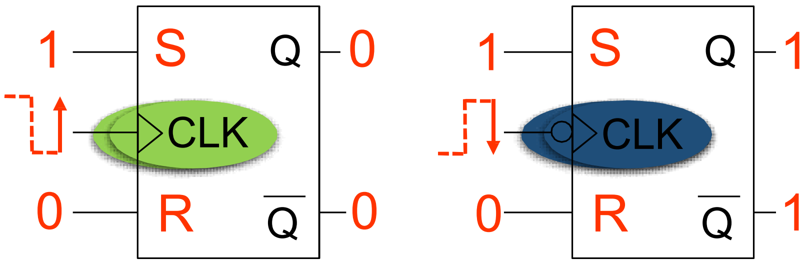


Level enabled

EN = 1, normal latch operation

EN = 0, no change in output even
when inputs change

Flip-flop

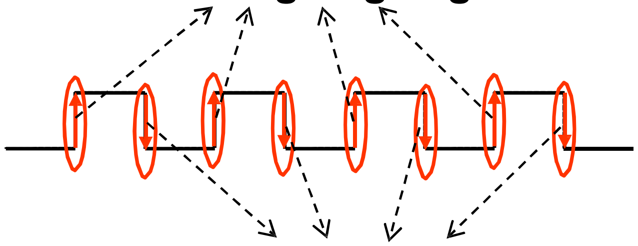


PGT FF **NGT FF**

Edge-triggered

CLOCK (CLK) signal

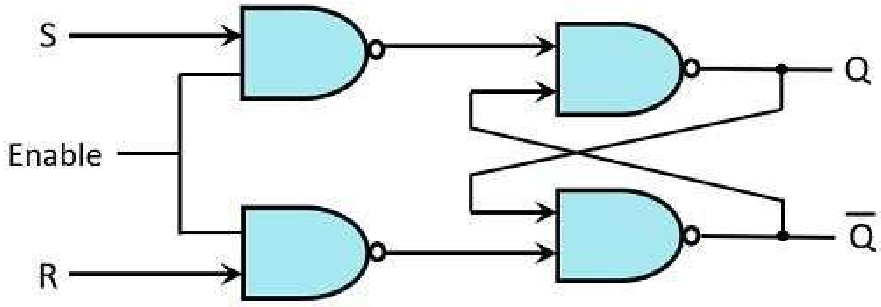
Positive-going edge



Negative-going edge

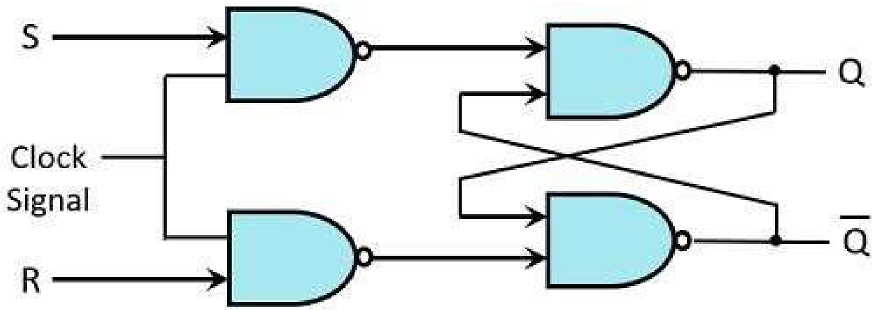
Flip Flops Vs Latches

- A flip flop is an electronic circuit with two stable states that can be used to store binary data.
- The stored data can be changed by applying varying inputs.
- **Latches and flip-flops** are circuit elements whose output depends not only on the current inputs, but also on previous inputs and outputs.
- The **difference between** a **latch** and a **flip-flop** is that a **latch** does not have a clock signal, whereas a **flip-flop** always does.



SR Latch

Circuit Globe



SR Flip Flop

Circuit Globe

Flipflops

❑ A Flip Flop is a memory element that is capable of storing one bit of information.

❖A flip flop can maintain a binary state for an unlimited period of time as long as-

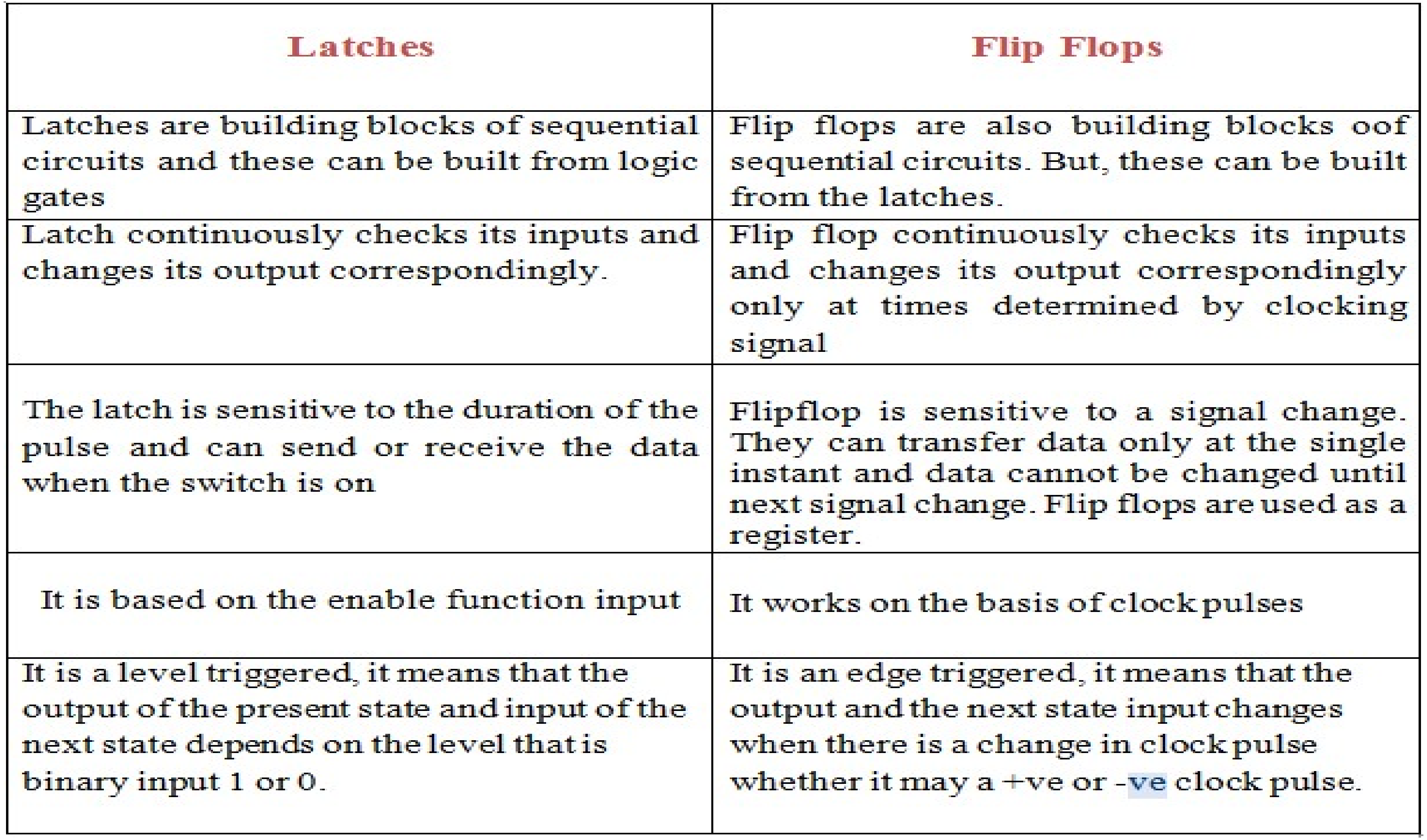
(i) Power is supplied to the circuit.

(ii) or until it is directed by an input signal to switch states

Types of Flipflops

- SR Flip Flop
- JK Flip Flop
- D Flip Flop
- T Flip Flop

Latch VS Flipflop

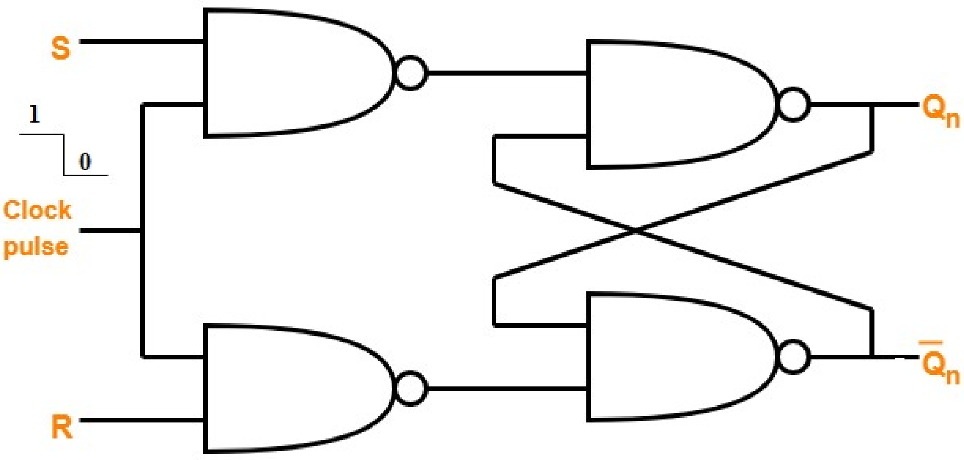


Latches	Flip Flops
Latches are building blocks of sequential circuits and these can be built from logic gates	Flip flops are also building blocks oof sequential circuits. But, these can be built from the latches.
Latch continuously checks its inputs and changes its output correspondingly.	Flip flop continuously 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.

Steps to design Flipflops

- Circuit diagram using universal gates
- Truth Table
- Characteristics Equation
- Excitation Table
- State Diagram
- Timing (or) Switching (or) Waveform Diagram

S-R Flip flop



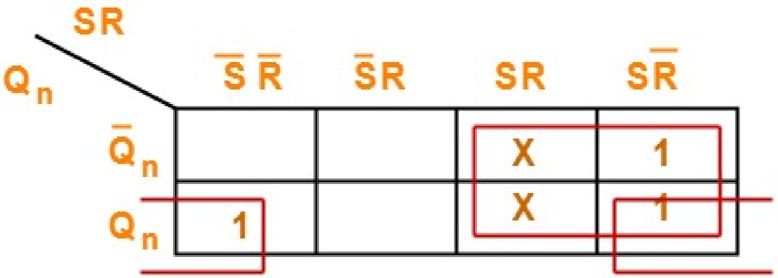
(i) Circuit diagram

(ii) Characteristics Table

C	S	R	Q_n	Q_{n+1}	State of Operation
1	0	0	0	0	hold
1	0	0	1	1	
1	0	1	0	0	clear (reset)
1	0	1	1	0	
1	1	0	0	1	set
1	1	0	1	1	
1	1	1	0	?	indeterminate
1	1	1	1	?	— avoid

(iii) Characteristic Equation

$$\boxed{Q_{n+1} = S + Q_nR'}$$

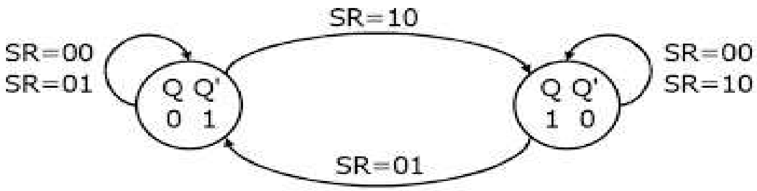


Q_n \ SR	$\bar{S}\bar{R}$	$\bar{S}R$	SR	$S\bar{R}$
$\bar{Q}_n$			X	1
Q_n	1		X	1

(iv) Excitation Table

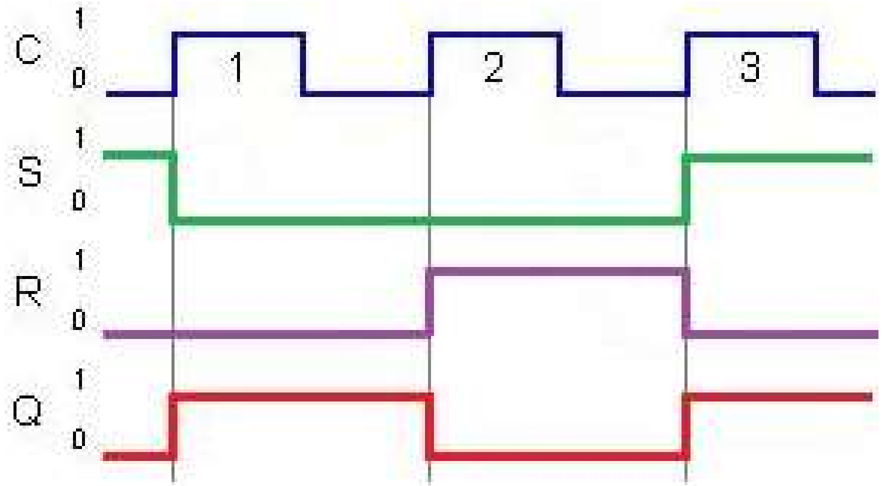
Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

(v) State Diagram



(V) Timing (or) Switching (or) Waveform Diagram

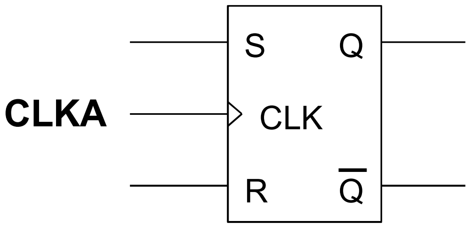
When Qn=1



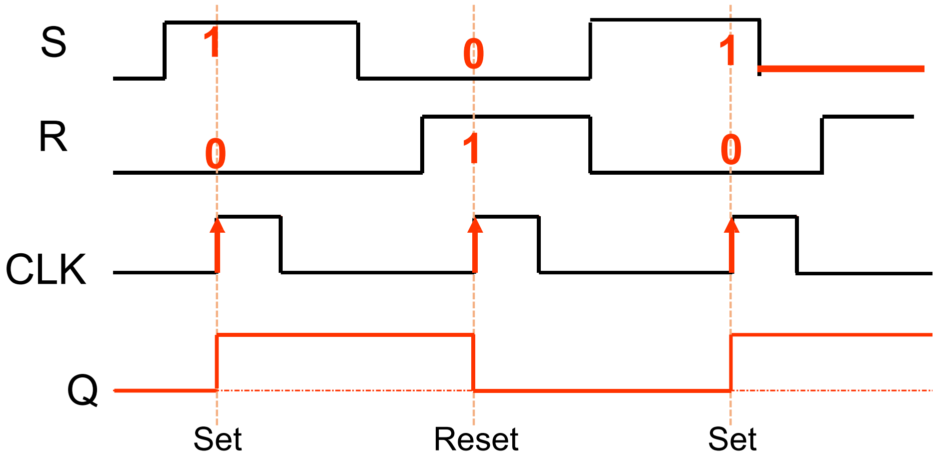
C	S	R	Q_n	Q_{n+1}
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	?
1	1	1	1	?

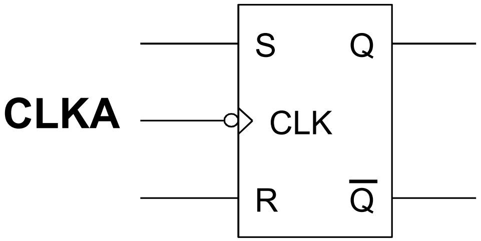
(ii) Characteristics Table

Timing diagrams

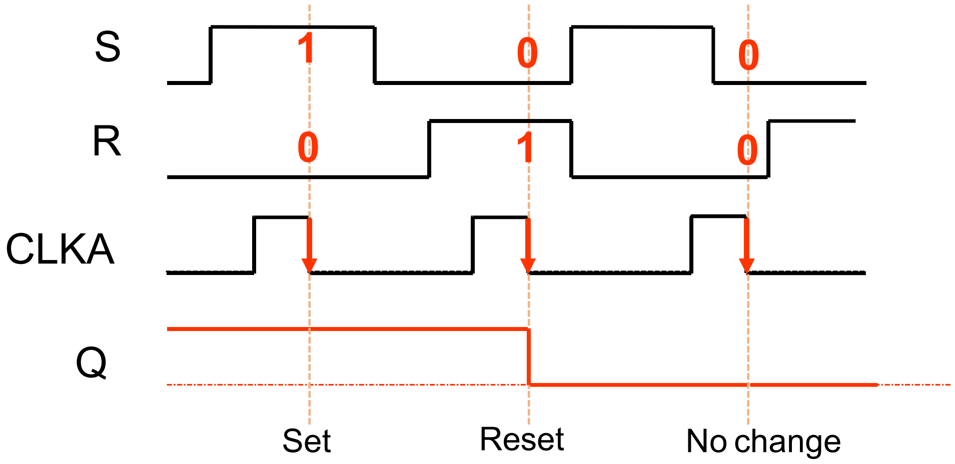


Assume Q=0 initially

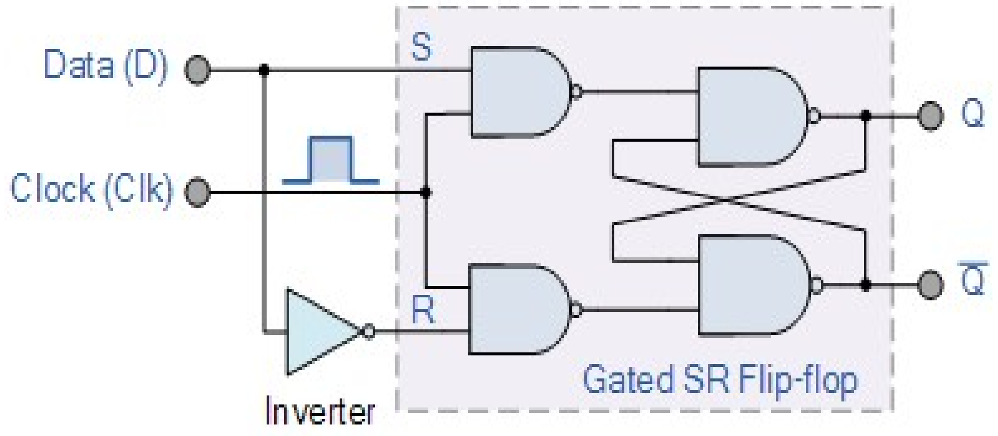




Assume Q=1 initially



D - Flip flop



(i) Circuit diagram

C	D	Q_n	Q_{n+1}	State of Operation
1	0	0	0	Clear (reset)
1	0	1	0	
1	1	0	1	Set
1	1	1	1	

(ii) Characteristics Table

(iii) Characteristic Equation

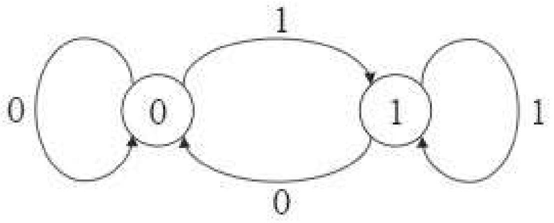
Qn+1 = D

	Qn'	Qn
D'	0	0
D	1	1

(iv) Excitation Table

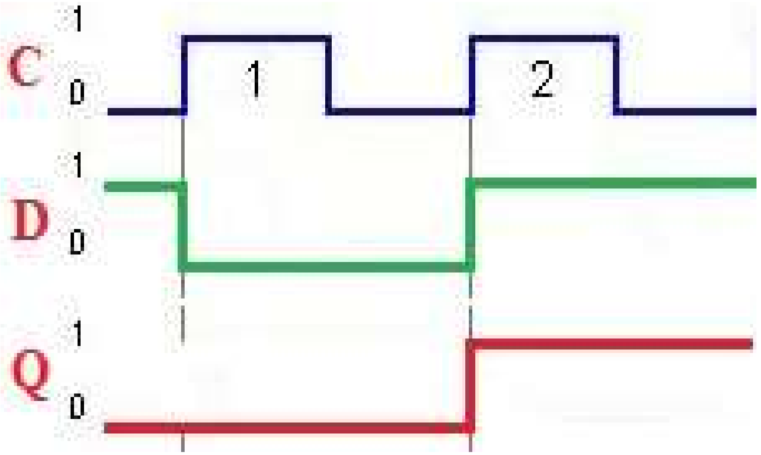
Qn	Qn+1	D
0	0	0
0	1	1
1	0	0
1	1	1

(v) State Diagram



(V) Timing (or) Switching (or) Waveform Diagram

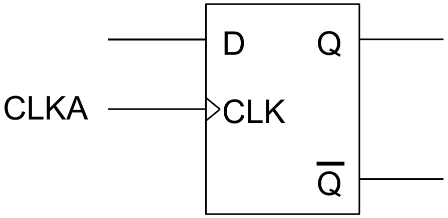
When Qn=1



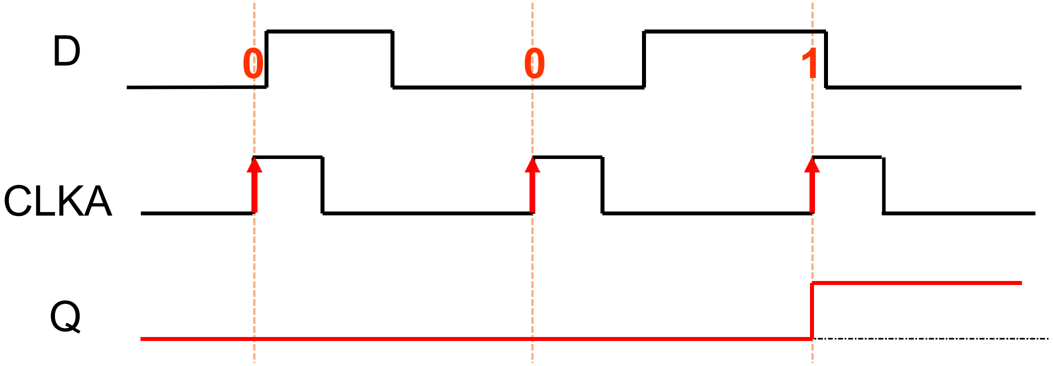
(ii) Characteristics Table

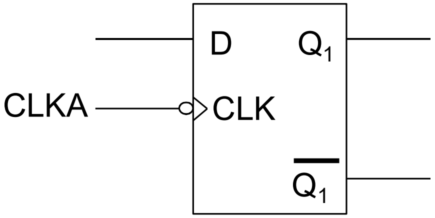
C	D	Q_n	Q_{n+1}	State of Operation
1	0	0	0	Clear (reset)
1	0	1	0	
1	1	0	1	Set
1	1	1	1	

timing diagrams

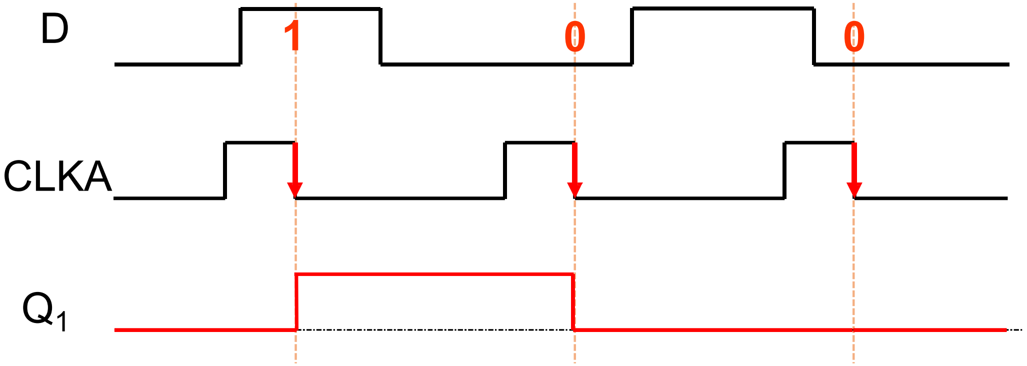


Assume Q=0 initially





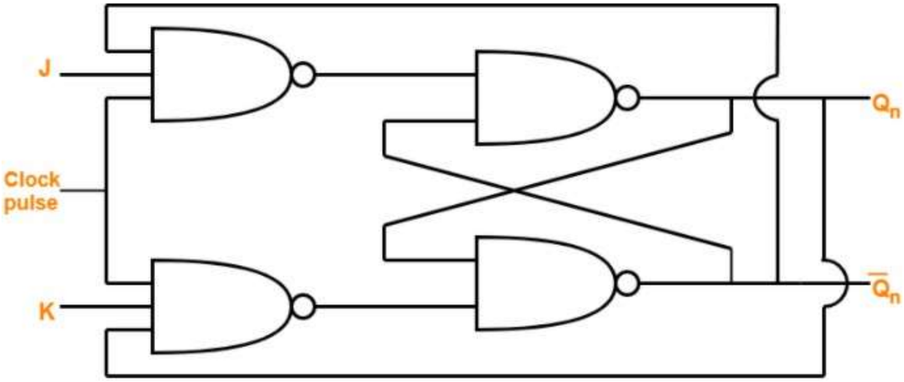
Assume Q1=0 initially



J-K Flip flop

- A J-K flip-flop behaves in the same fashion as an R-S flip-flop except for one of the entries in the function table.
- In the case of an R-S flip-flop, the input combination S = R = 1 are prohibited.
- In the case of a J-K flip-flop, the output of the flip-flop toggles, that is, it goes to the other state, for J = K = 1 .
- Thus, a J-K flip-flop overcomes the problem of a forbidden input combination of the R-S flip-flop.

(i) Circuit diagram

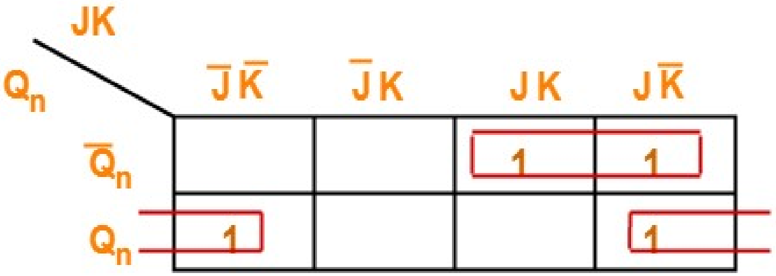


(ii) Characteristics Table

C	J	K	Q_n	Q_{n+1}	State of Operation
1	0	0	0	0	hold
1	0	0	1	1	
1	0	1	0	0	clear (reset)
1	0	1	1	0	
1	1	0	0	1	set
1	1	0	1	1	
1	1	1	0	1	Toggle
1	1	1	1	0	

(iii) Characteristic Equation

$$Q_{n+1} = Q'_nJ + Q_nK'$$

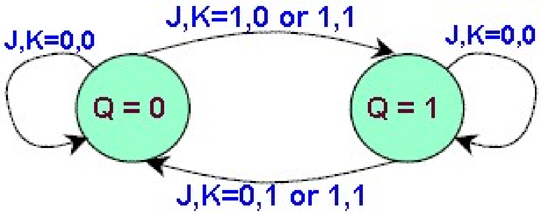


$Q_n \backslash JK$	$\bar{J}\bar{K}$	$\bar{J}K$	JK	$J\bar{K}$
$\bar{Q}_n$			1	1
Q_n	1			1

(iv) Excitation Table

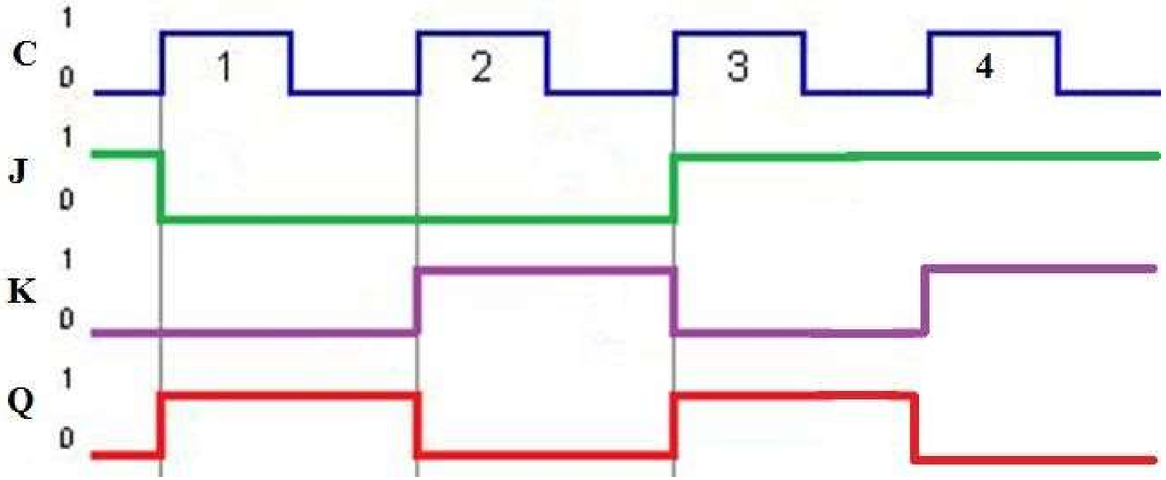
Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

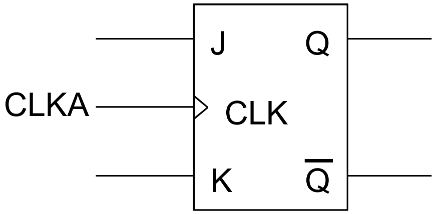
(v) State Diagram



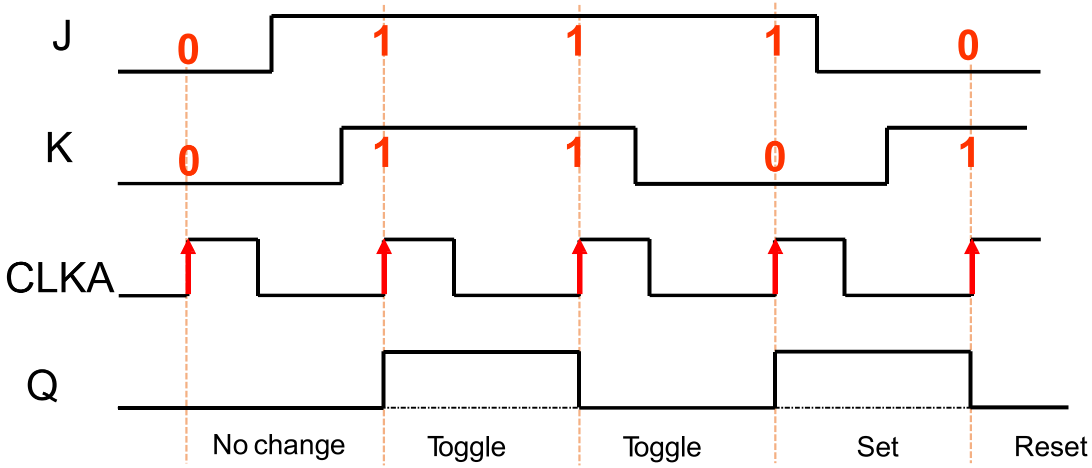
(V) Timing (or) Switching (or) Waveform Diagram

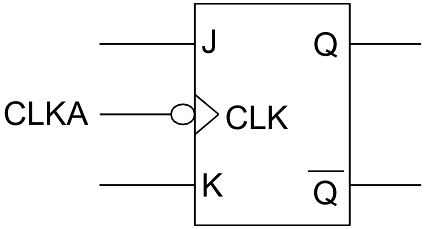
When Qn=1



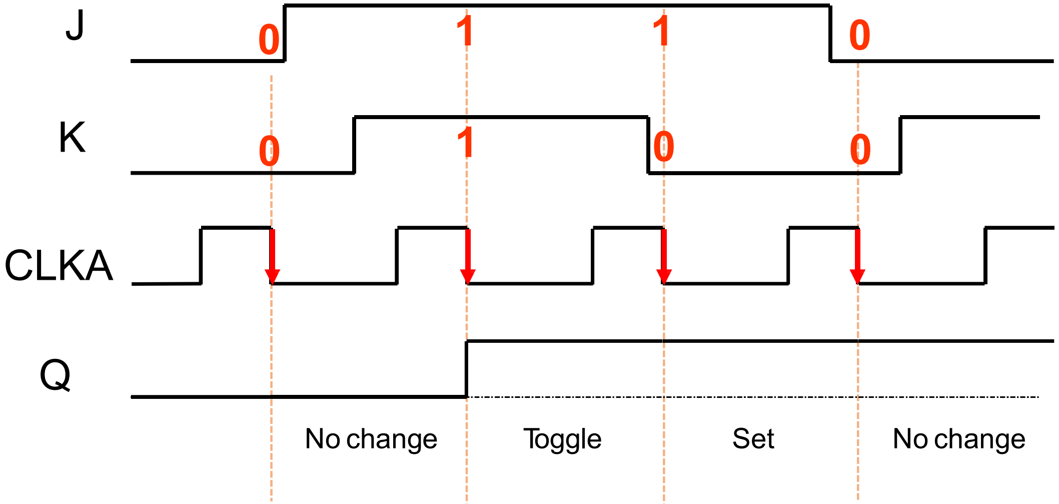


Assume Q=0 initially



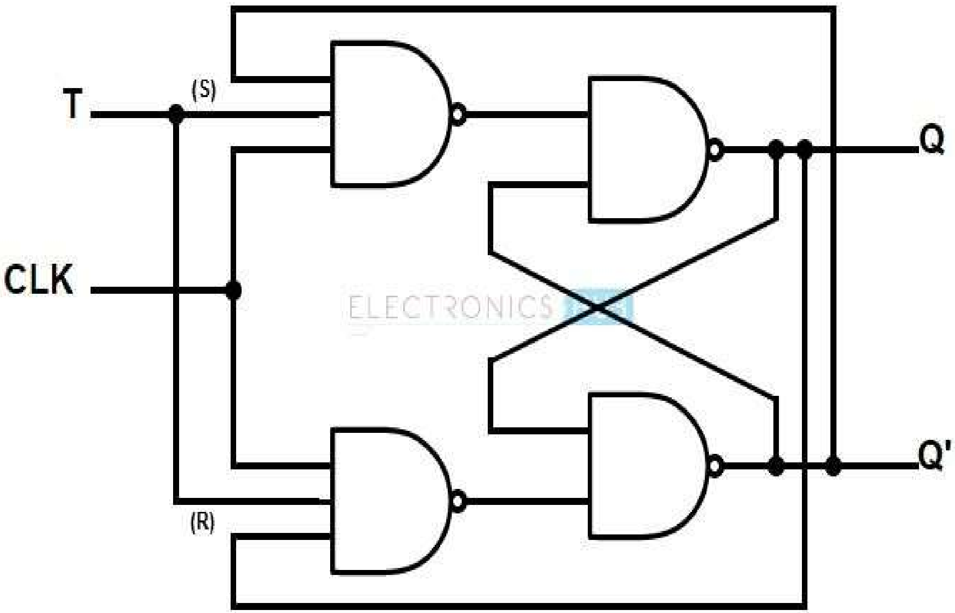


Assume Q=0 initially

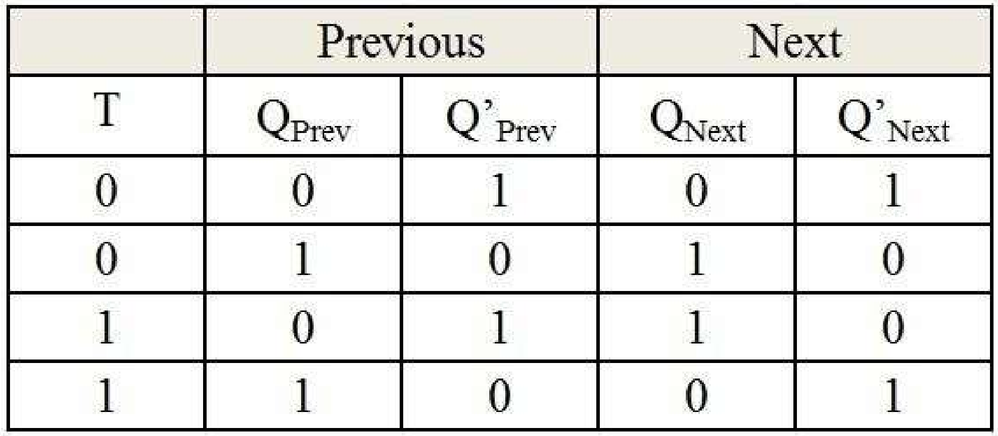


T Flipflop

(i) Circuit Diagram

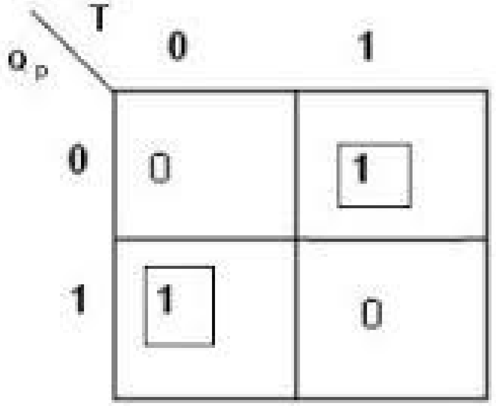


(ii) Characteristics Table



	Previous		Next	
T	Q_{Prev}	Q'_{Prev}	Q_{Next}	Q'_{Next}
0	0	1	0	1
0	1	0	1	0
1	0	1	1	0
1	1	0	0	1

(iii) Characteristic Equation

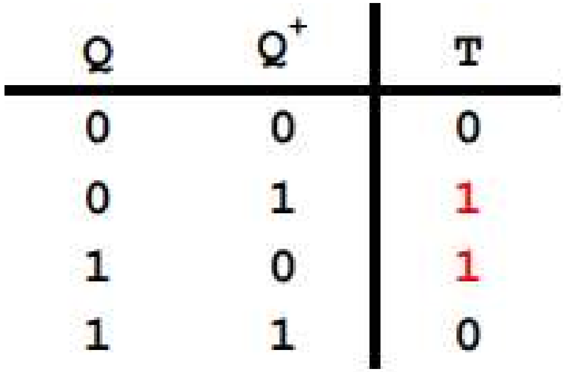


Q_p \ T	0	1
0	0	$\boxed{1}$
1	$\boxed{1}$	0

The Equation we get is

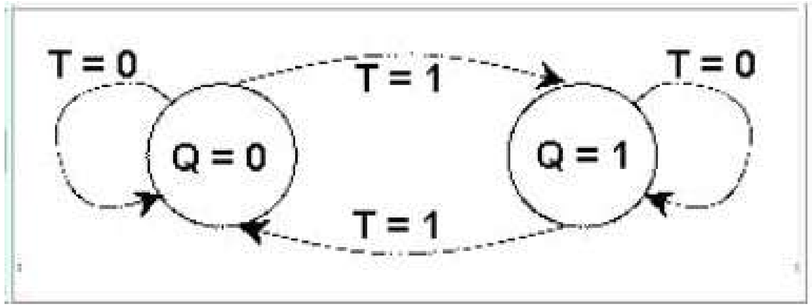
$$Q = T\,Q_p' + T'\,Q_p$$
$$= T \text{ XOR } Q_p$$

(iv) Excitation Table

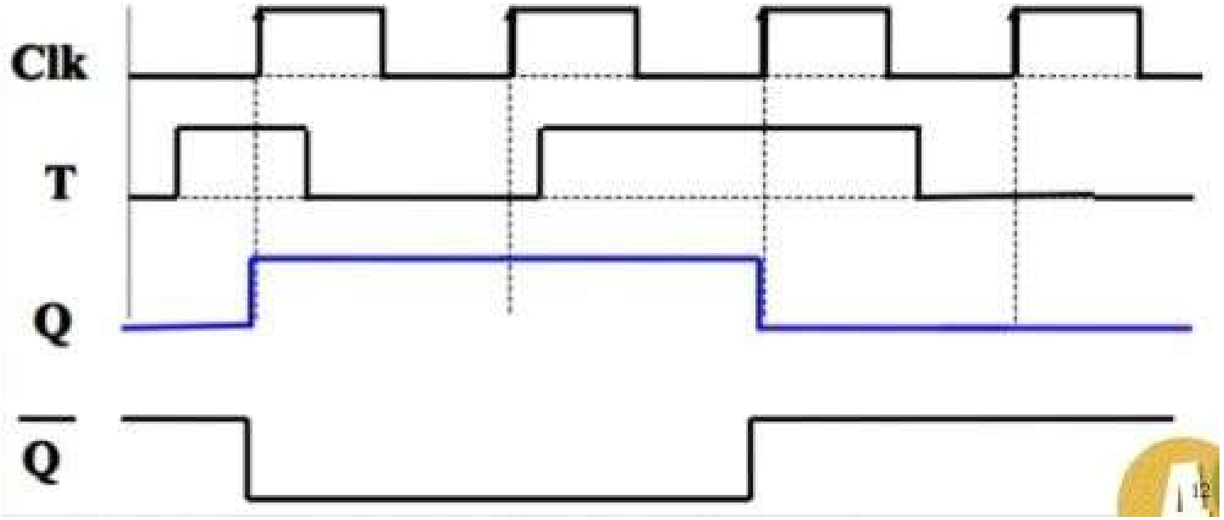


Q	Q^+	T
0	0	0
0	1	1
1	0	1
1	1	0

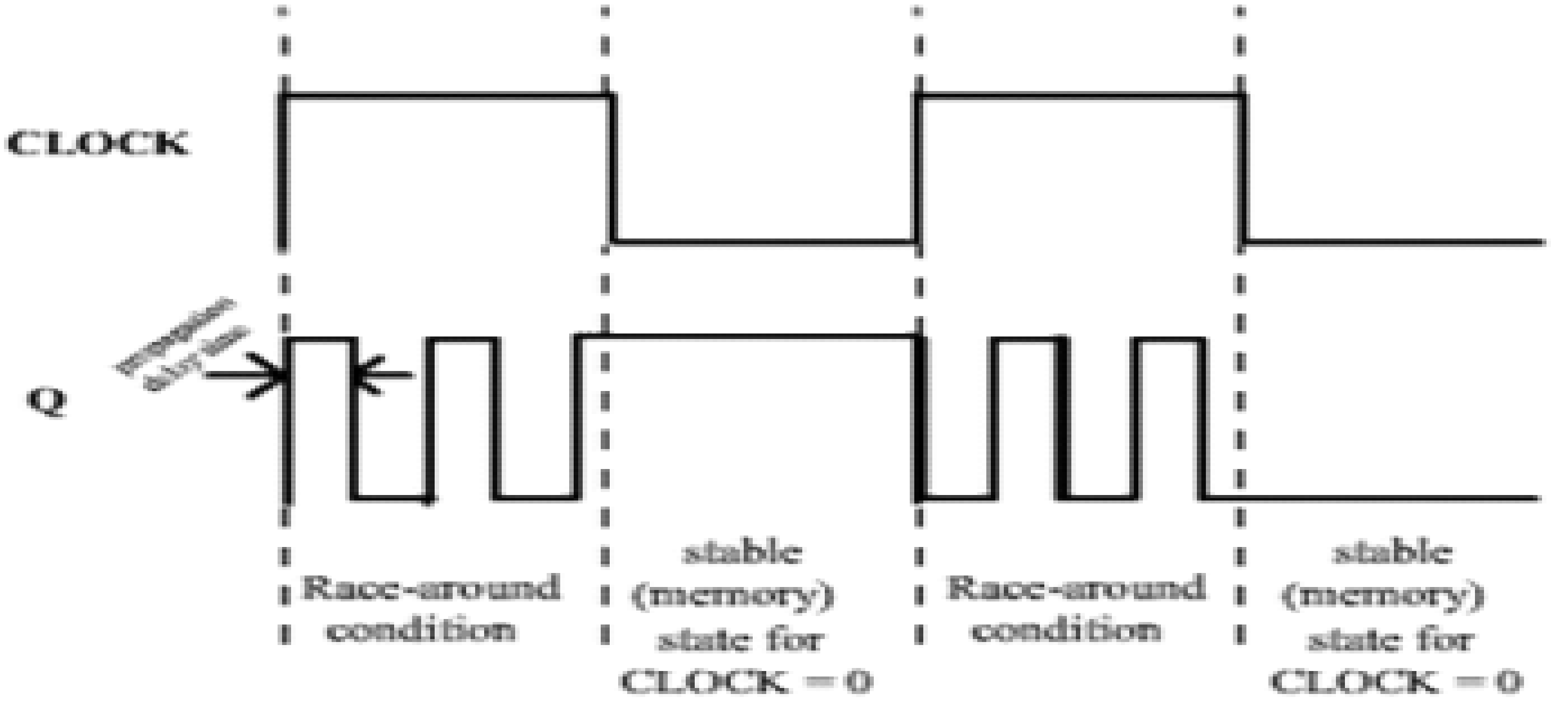
(v) State Diagram



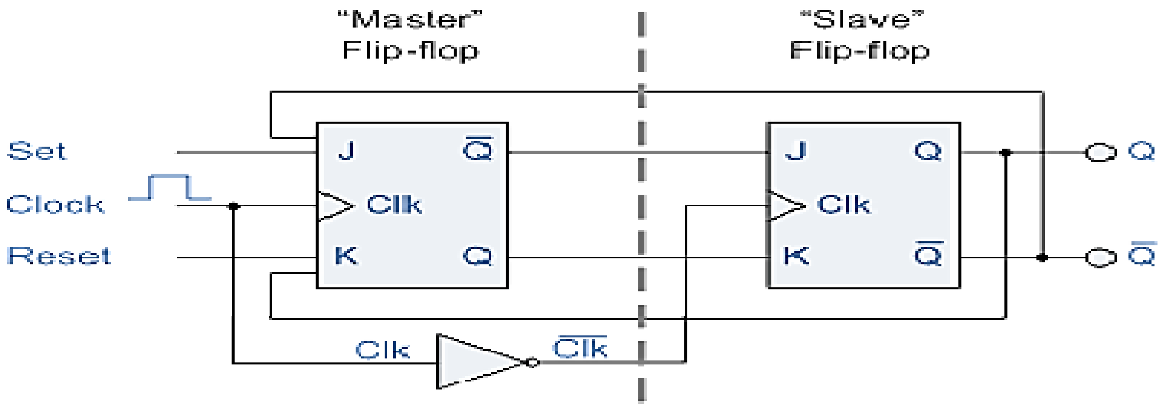
(vi) Timing Diagram

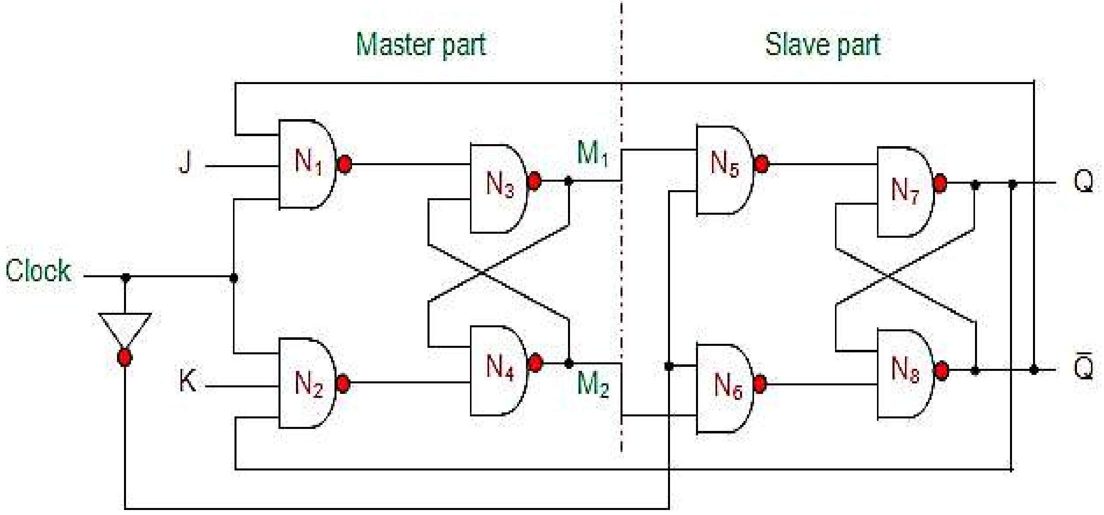


Race around condition in JK FF

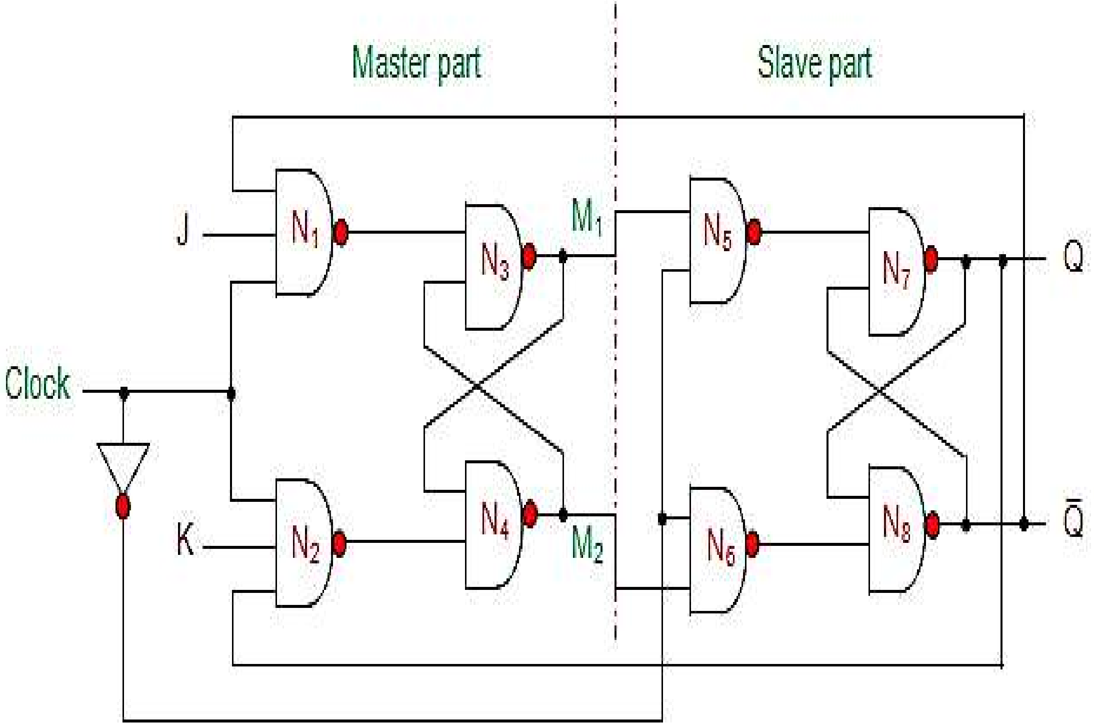

CLOCK
Q
propagation delay time
Race-around condition
stable (memory) state for CLOCK = 0
Race-around condition
stable (memory) state for CLOCK = 0

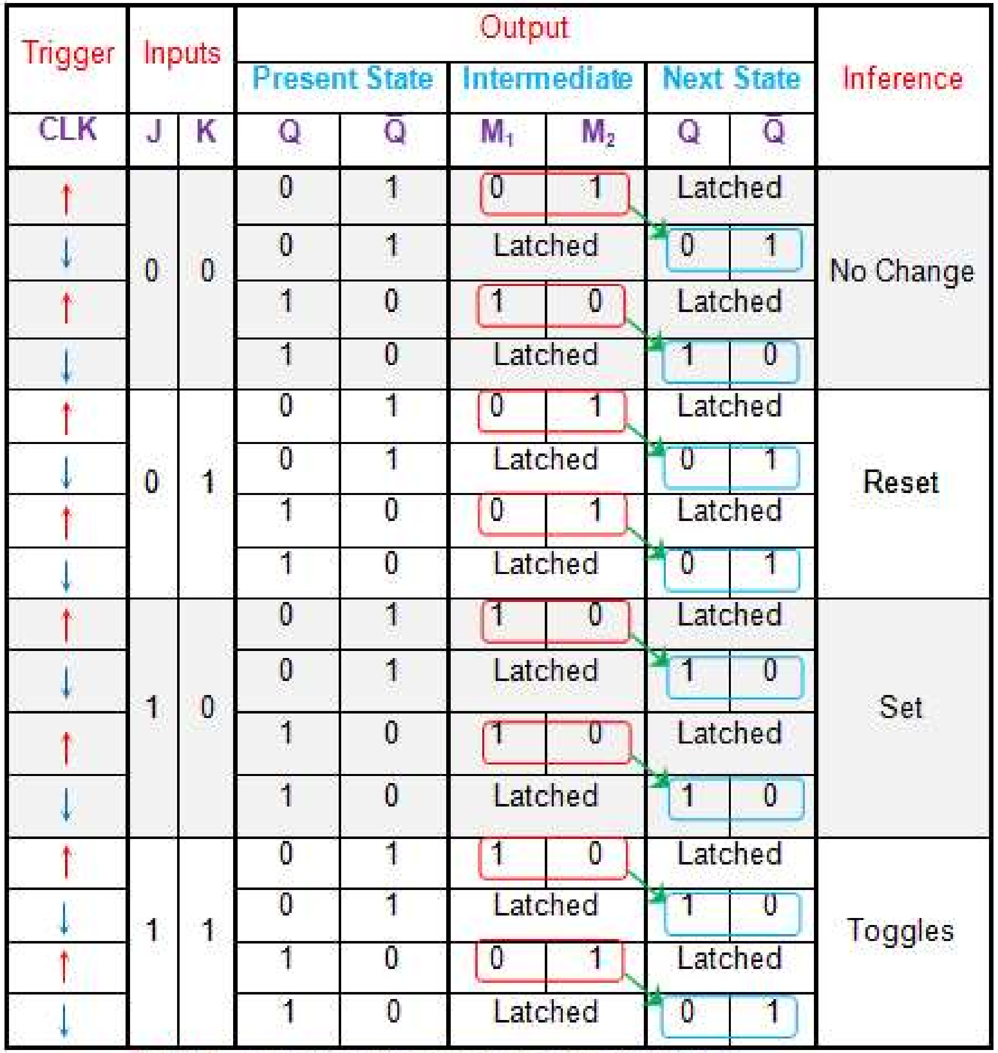
Master slave JK flipflop





Master slave JK flipflop

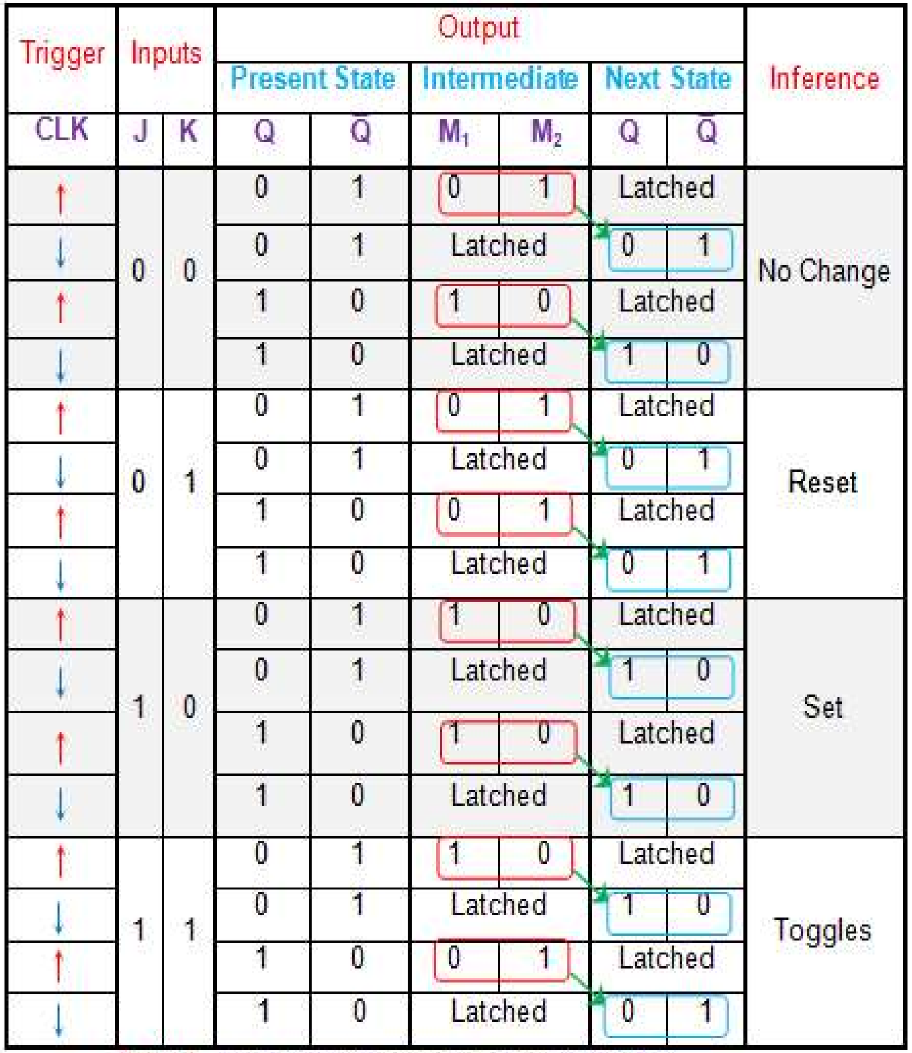




Trigger	Inputs		Output						Inference
			Present State		Intermediate		Next State		
CLK	J	K	Q	$\bar{Q}$	M_1	M_2	Q	$\bar{Q}$	
↑	0	0	0	1	0	1	Latched		No Change
↓			0	1	Latched		0	1	
↑			1	0	1	0	Latched		
↓			1	0	Latched		1	0	
↑	0	1	0	1	0	1	Latched		Reset
↓			0	1	Latched		0	1	
↑			1	0	0	1	Latched		
↓			1	0	Latched		0	1	
↑	1	0	0	1	1	0	Latched		Set
↓			0	1	Latched		1	0	
↑			1	0	1	0	Latched		
↓			1	0	Latched		1	0	
↑	1	1	0	1	1	0	Latched		Toggles
↓			0	1	Latched		1	0	
↑			1	0	0	1	Latched		
↓			1	0	Latched		0	1	

Table I Truth table for master-slave JK flip-flop

Master slave JK flipflop



Trigger	Inputs		Output						Inference
			Present State		Intermediate		Next State		
CLK	J	K	Q	$\bar{Q}$	M_1	M_2	Q	$\bar{Q}$	
↑	0	0	0	1	0	1	Latched		No Change
↓			0	1	Latched		0	1	
↑			1	0	1	0	Latched		
↓			1	0	Latched		1	0	
↑	0	1	0	1	0	1	Latched		Reset
↓			0	1	Latched		0	1	
↑			1	0	0	1	Latched		
↓			1	0	Latched		0	1	
↑	1	0	0	1	1	0	Latched		Set
↓			0	1	Latched		1	0	
↑			1	0	1	0	Latched		
↓			1	0	Latched		1	0	
↑	1	1	0	1	1	0	Latched		Toggles
↓			0	1	Latched		1	0	
↑			1	0	0	1	Latched		
↓			1	0	Latched		0	1	

Table I Truth table for master-slave JK flip-flop

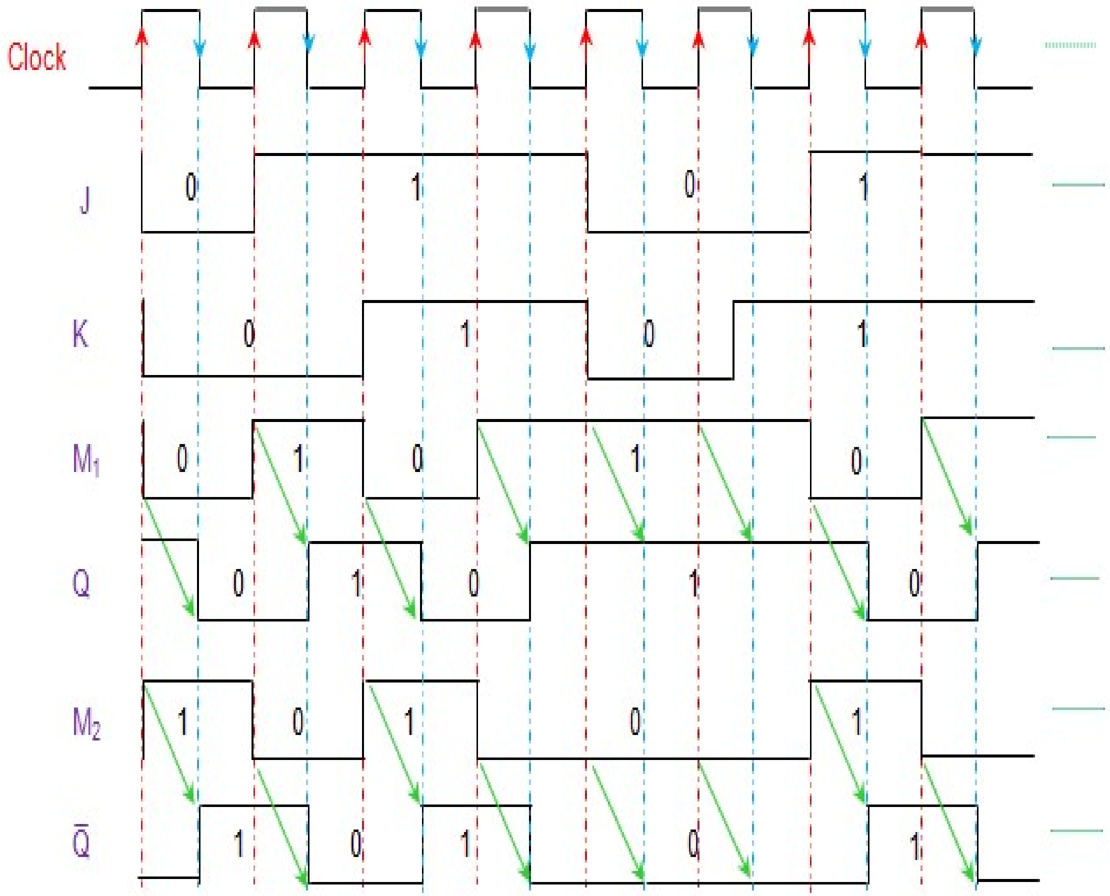


Figure 3 Timing diagram for master-slave JK flip-flop

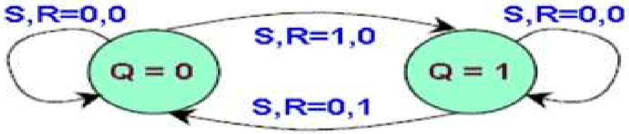
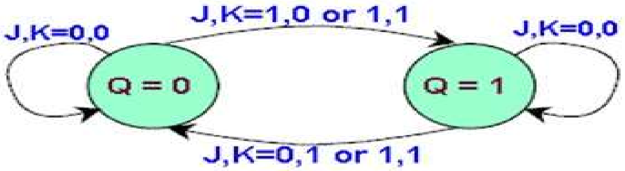
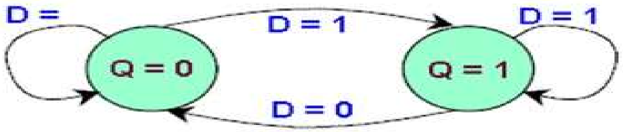
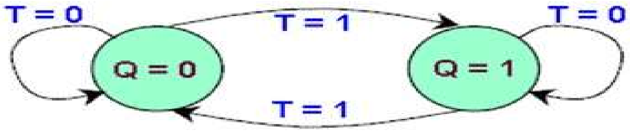
Excitation Tables

RS FLIP FLOP							
Characteristics Table				Excitation Table			
S	R	Q_{n+1}		Qn	Qn+1	S	R
0	0	Qn		0	0	0	X
0	1	0		0	1	1	0
1	0	1		1	0	0	1
1	1	X		1	1	X	0

JK FLIP FLOP							
Characteristics Table				Excitation Table			
J	K	Qn+1		Qn	Qn+1	J	K
0	0	Qn		0	0	0	X
0	1	0		0	1	1	X
1	0	1		1	0	X	1
1	1	Qn’		1	1	X	0

D FLIP FLOP						
Characteristics Table				Excitation Table		
D	Qn	Qn+1		Qn	Qn+1	D
0	0	0		0	0	0
0	1	0		0	1	1
1	0	1		1	0	0
1	1	1		1	1	1

T FLIP FLOP						
Characteristics Table				Excitation Table		
T	Qn	Qn+1		Qn	Qn+1	T
0	0	0		0	0	0
0	1	1		0	1	1
1	0	1		1	0	1
1	1	0		1	1	0

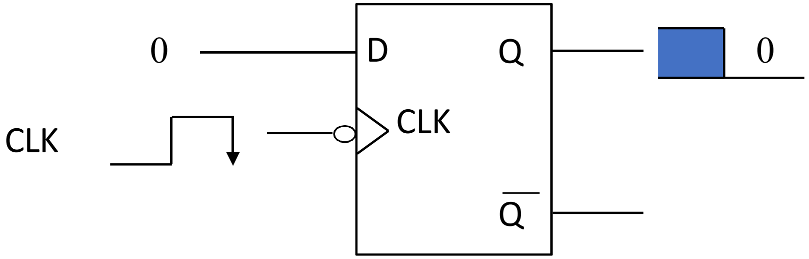
NAME	STATE DIAGRAM
SR	
JK	 J,K=1,0 or 1,1 J,K=0,0 J,K=0,0 Q = 0 Q = 1 J,K=0,1 or 1,1
D	
T	

REGISTERS

Shift registers are types of sequential logic circuits closely related to digital counters used primarily for the storage of data

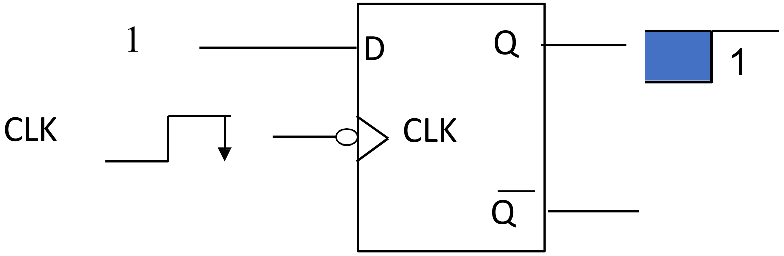
DATA STORAGE

transfer 0 from input to output



Q → 0 at the clock NGT

transfer 1 from input to output



Q → 1 at the clock NGT

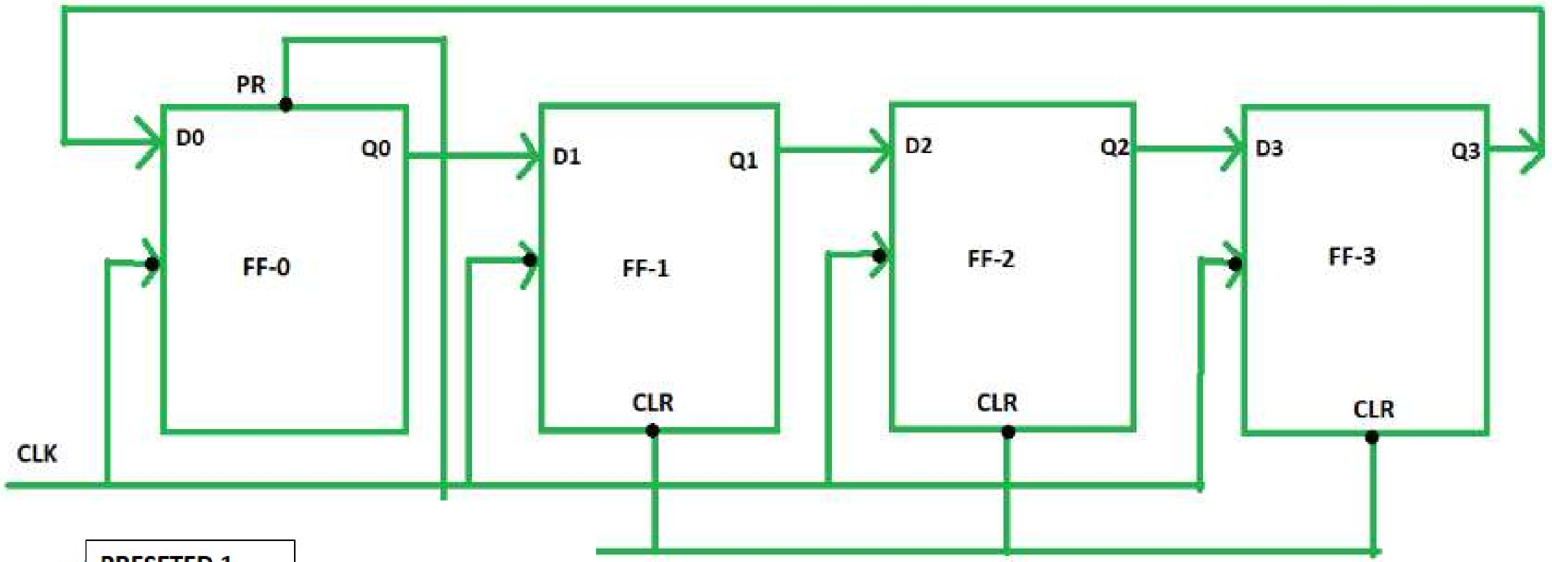
DATA STORAGE AND SHIFT

- Storage capacity (number of bits) of a register is determined by the number of flip-flops
  - One flip flop represents one bit of data storage
  - "n" number of flip-flops can store an **n-bit** data

- The FFs interconnection should facilitate
  - transfer of binary data into and out of the register, and
  - shifting data within the register one or more bit positions left or right, effected upon the application of clock pulses

- If **no clock pulses** are applied, there will be **no data shift** in the register

Ring counter



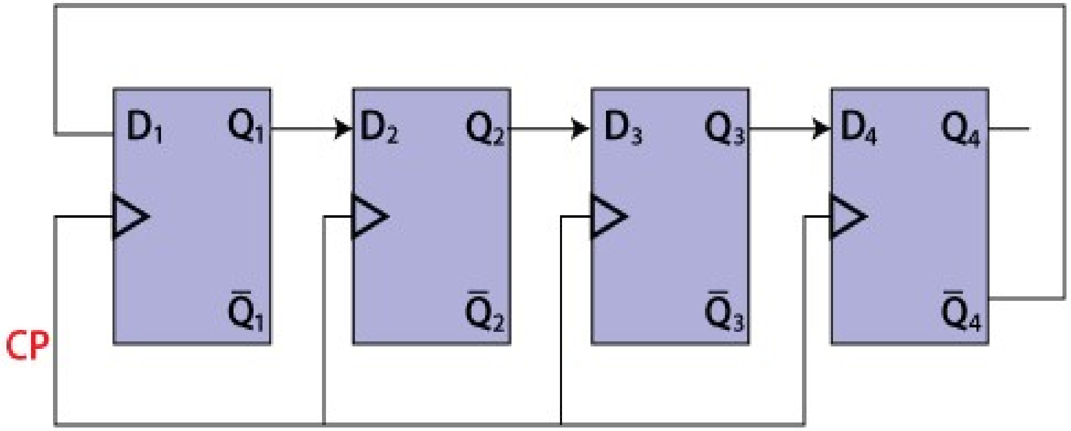
Ring Counter

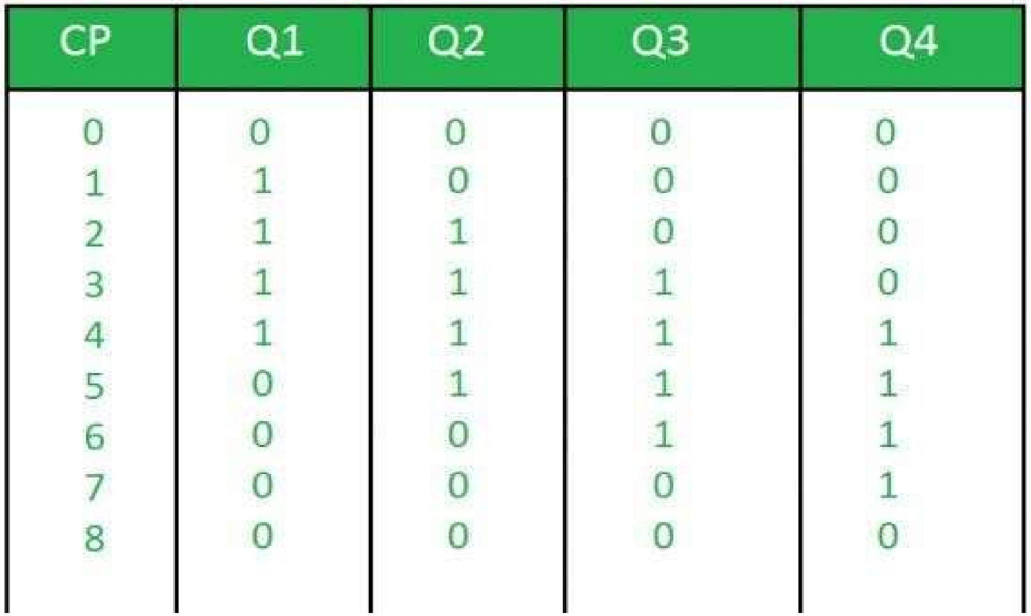
PRESETED 1

ORI	CLK	Q0	Q1	Q2	Q3
low	X	1	0	0	0
high	low	0	1	0	0
high	low	0	0	1	0
high	low	0	0	0	1
high	low	1	0	0	0

- It is a Synchronous Counter.
- Also, here we use Overriding input (ORI) for each flip-flop. Preset (PR) and Clear (CLR) are used as ORI.
- 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 always works in value 0.

Johnson counter





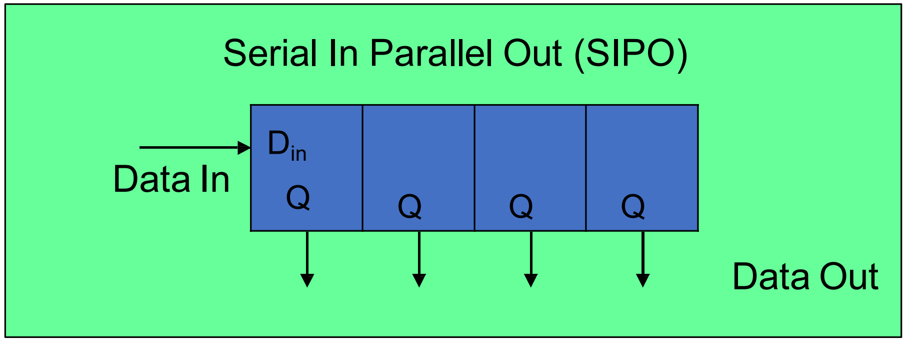
CP	Q1	Q2	Q3	Q4
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
8	0	0	0	0

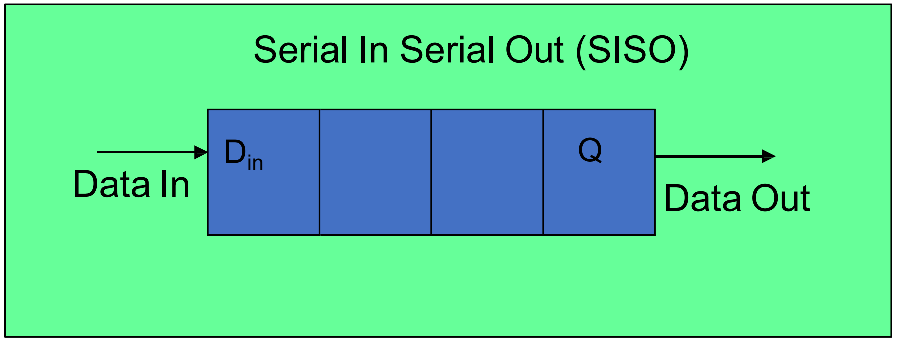
- Johnson counter also known as creeping counter
- In Johnson counter, the complemented output of last flip flop is connected to input of first flip flop and to implement n-bit Johnson counter we require n flip-flop.
- It is one of the most important type of shift register counter.
- Johnson counter is a ring with an inversion.

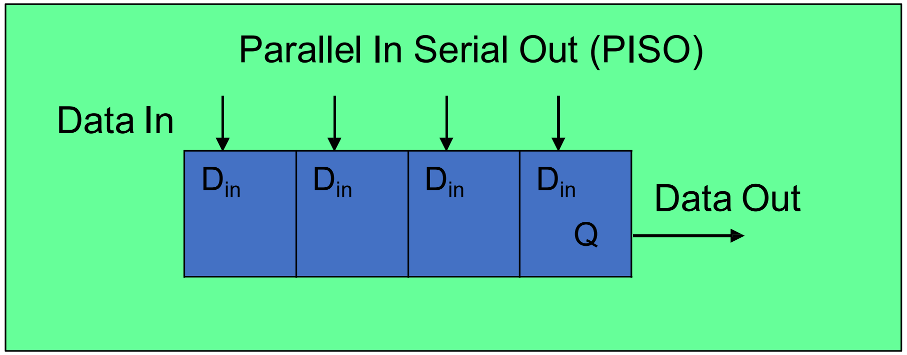
In Johnson counter
1.No. of states in Johnson counter = No. of flip-flop used
2.Number of used states=2n
3.Number of unused states=2n - 2*n

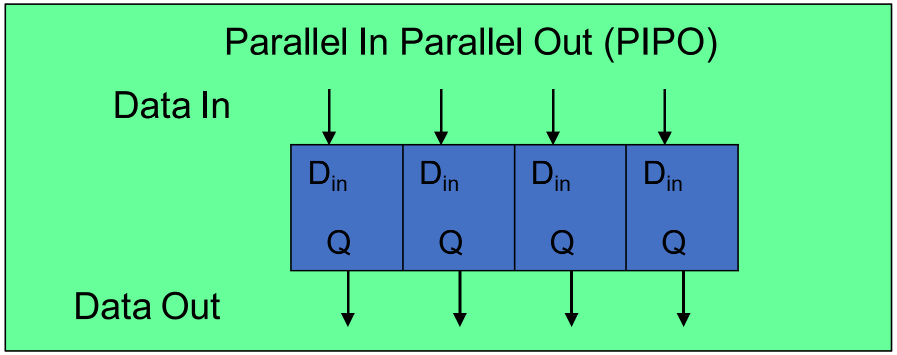
DATA MANIPULATION

- Data can *written in* and *read out* from a register either serially (one bit at a time) or in parallel (all bits simultaneously)


Serial In Parallel Out (SIPO)


Serial In Serial Out (SISO)


Parallel In Serial Out (PISO)


Parallel In Parallel Out (PIPO)

4 TYPES OF DATAMOVEMENT

Type 1 - Serial In Parallel Out (SIPO)

Data bits are entered serially into the shift register.
Each bit appears on its respective output line, and all bits are available simultaneously.

Type 2 - Serial In Serial Out (SISO)

The serial in serial out shift register accepts data serially, i.e. one bit at a time on a single line.
It produces stored information on its output also in serial form.

4 TYPES OF DATA MOVEMENT

Type 3 - Parallel In Serial Out (PISO)

Data bits are entered simultaneously into their respective stages in parallel fashion.
The output is serial i.e. shifting one bit at a time on a single line.

Type 4 - Parallel In Parallel Out (PIPO)

Data bits are entered simultaneously into their respective stages in parallel. All bits are also available at the respective output stage in parallel.

SIPO Register Basic Operation

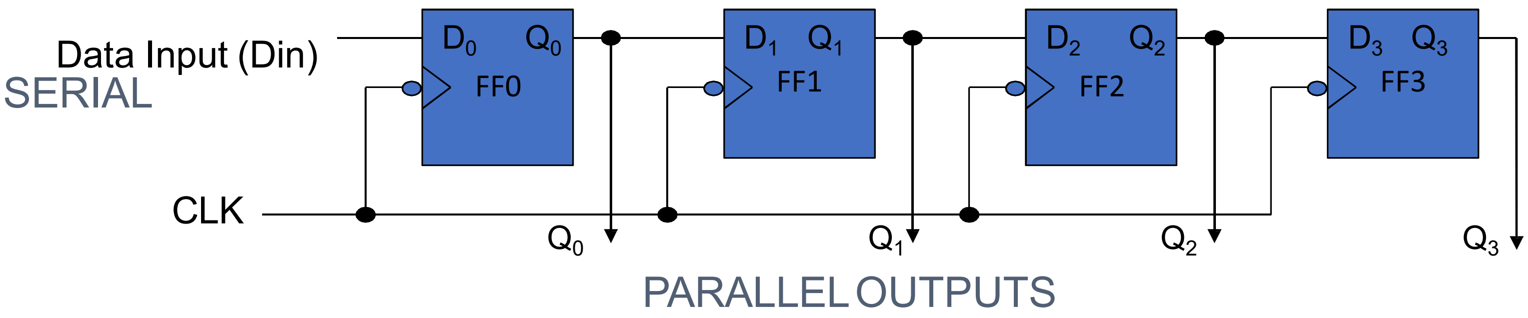
- Accepts data serially on bit by bit basis
- Once data is stored, each data bit appears simultaneously on its respective output line

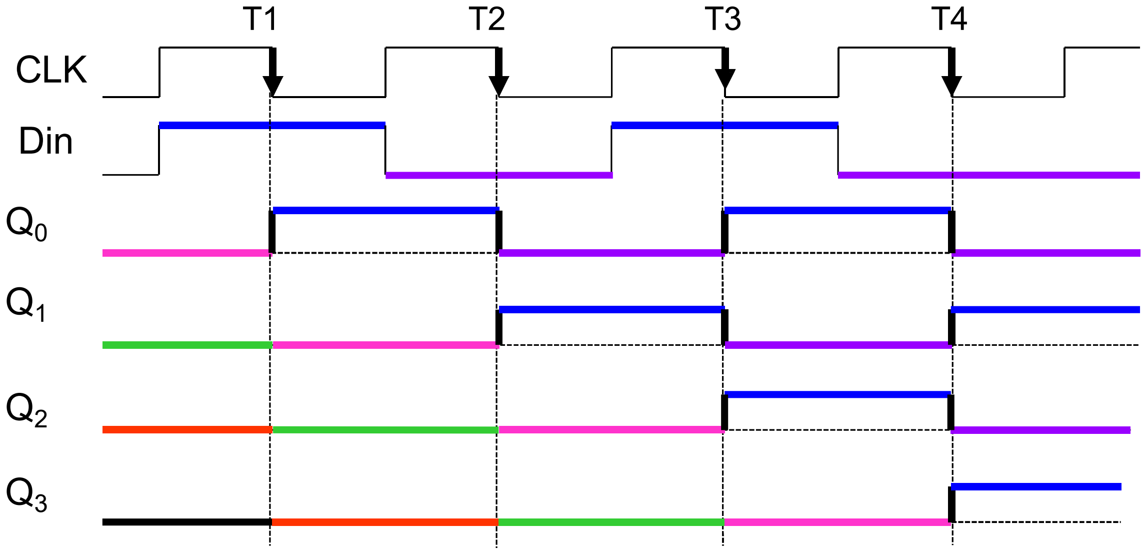
- Application of SIPO:

  - A computer or microprocessor based system commonly processes data in parallel format.

  - Hence, serial incoming data is converted into parallel format through a SIPO register

Example 1: 4-bit Shift SIPO Register





CLK	Din	Q_0	Q_1	Q_2	Q_3
Initial	x	0	0	0	0
T1	1	1	0	0	0
T2	0	0	1	0	0
T3	1	1	0	1	0
T4	0	0	1	0	1

Show the contents of the register for the data input and clock given. Initial register value is 0.

SISO Register Basic Operation

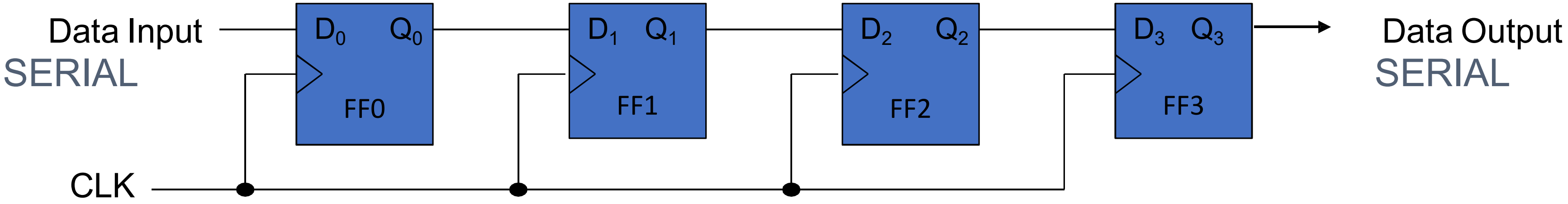
- Data is shifted into the register on bit by bit basis.
- After the data is stored completely, it can be shifted out serially upon further application of the clock pulses

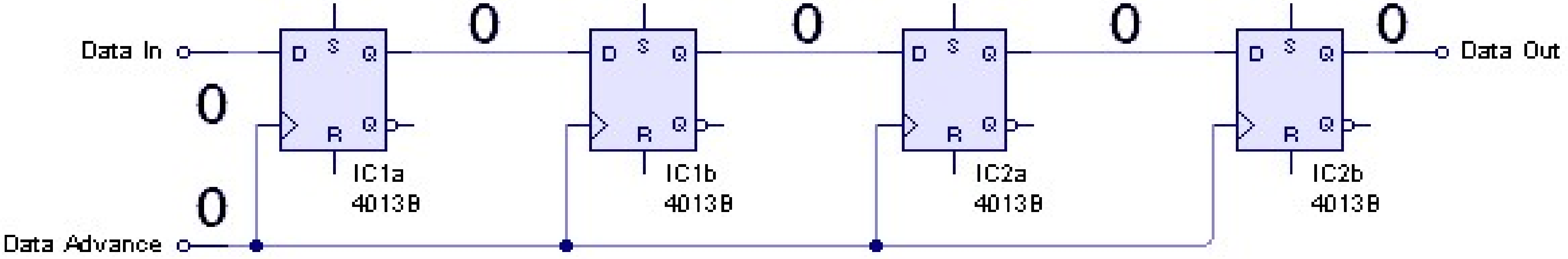
- Application of the SISO Register

  - The SISO shift register can be used to provide a time delay from input to output.

  - This delay is a function of the number of FFs in the register and clock period.

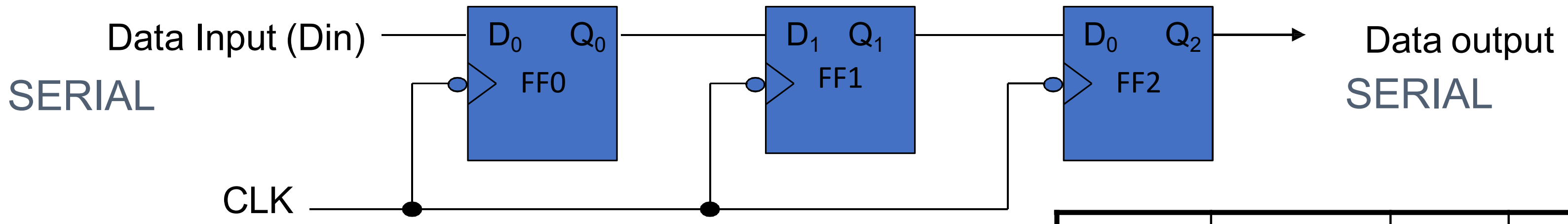
4-bit SISO shift register using D-FFs

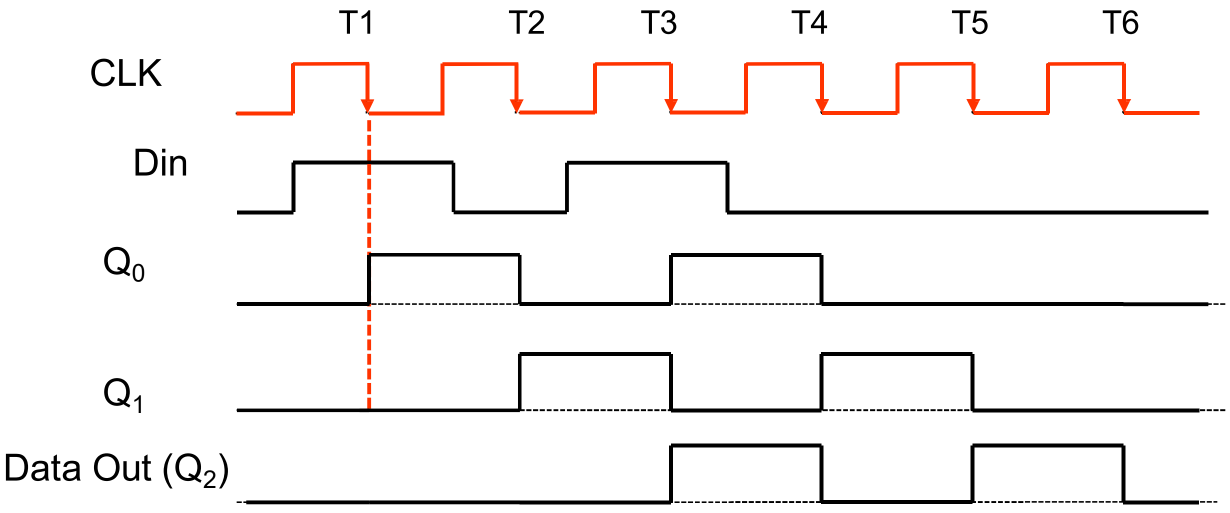




Data Advance (clock) connected to PGT → Data is shifted when clock signal changes from 0 to 1

Example 2: 3-bit shift SISO register





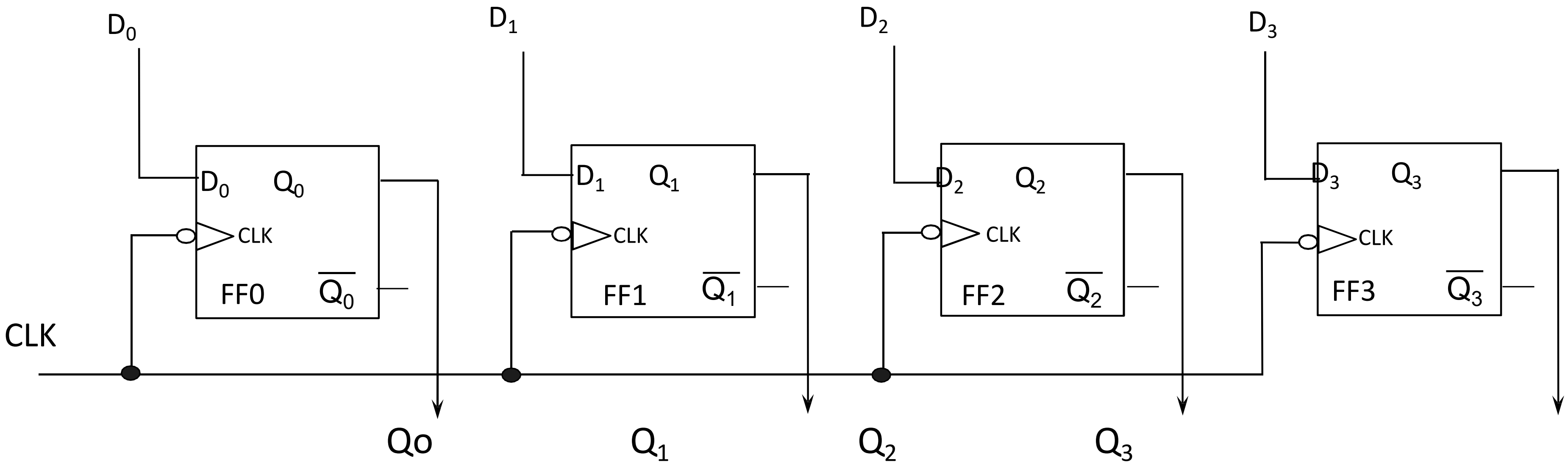
CLK	Din	Q_0	Q_1	Q_2
Initial	x	0	0	0
T1	1	1	0	0
T2	0	0	1	0
T3	1	1	0	1
T4	0	0	1	0
T5	0	0	0	1
T6	0	0	0	0

- Show the state of the output Q of the 3-bit SISO register for the data input and clock given. Initial register value is 0.

Parallel In Parallel Out Shift Register

- All the data bits are entered and available on the output lines simultaneously.

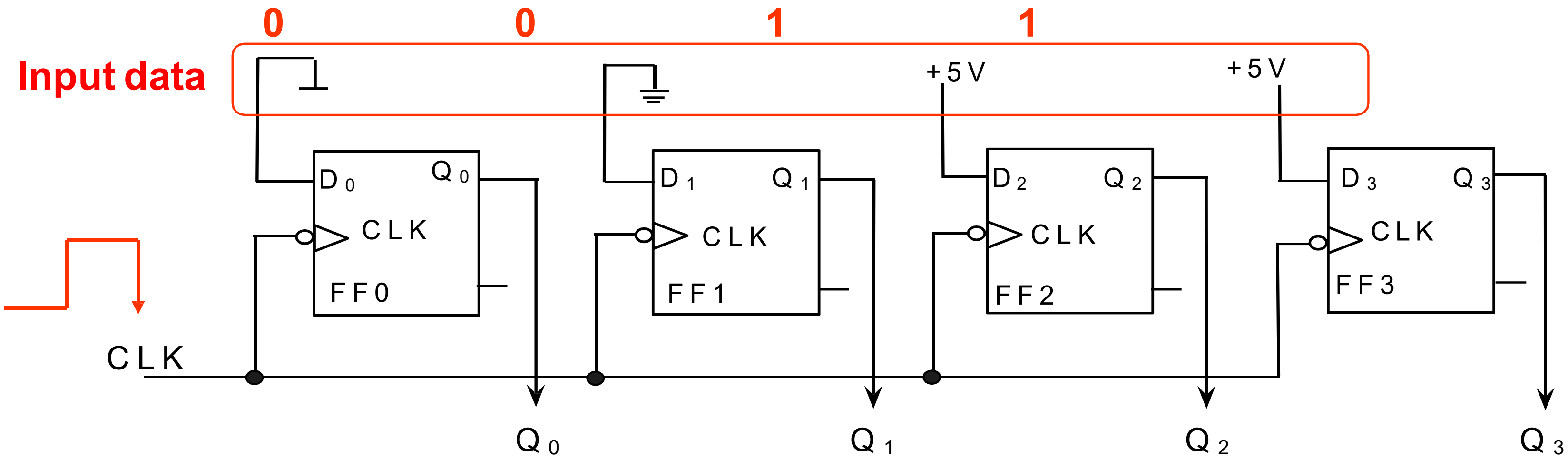
- Application: For synchronization of signals and data transfer



4-bit PIPO shift register using D flip-flops

Example 4

For the 4-bit PIPO register below, the initial states are given $Q_0Q_1Q_2Q_3$= 1000. what are the outputs of registers immediately after 3rd clock pulse?



Outputs of registers will always be 0011 after first and subsequent clock pulses. Why?

Input data at D is fixed

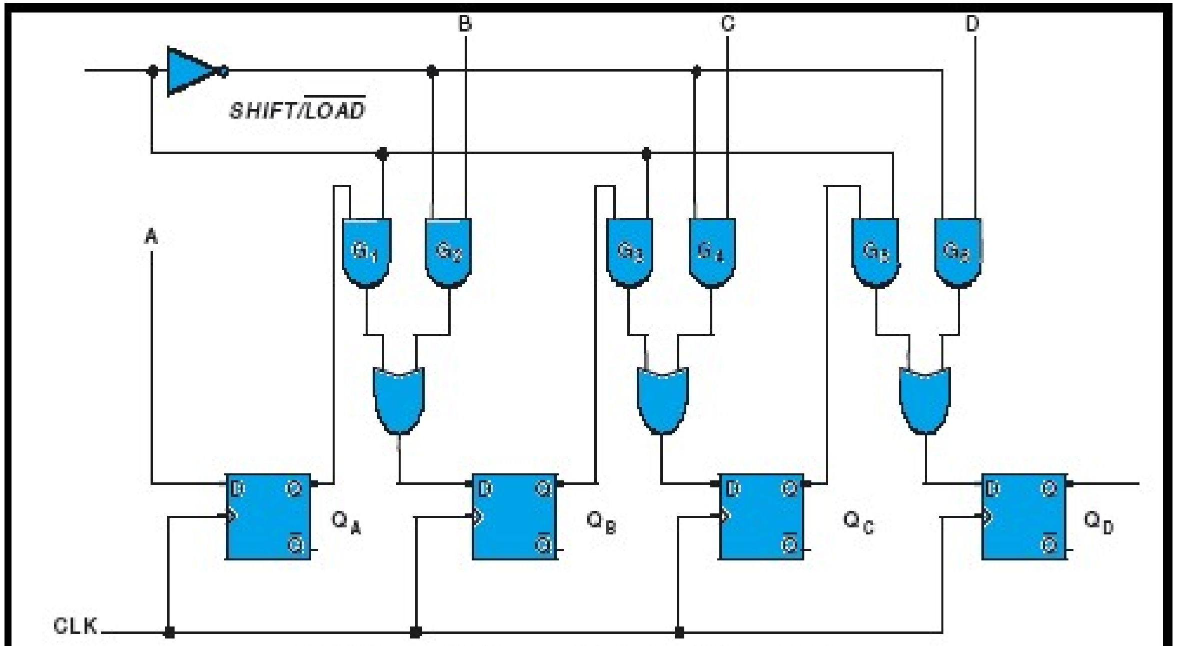
Parallel In Serial Out Shift Register

- Loading: Data bits are loaded simultaneously in parallel to their respective flip-flops. This happens when SH/$\overline{LD}$ = 0 and inputs are taken from parallel data Q_A-Q_H.

- Shifting: Once data is completely stored in the register, it is shifted out serially bit by bit. This happens when SH/$\overline{LD}$ = 1 and input bit for 1st FF is taken from SER input.

Parallel In - Serial Out Shift Registers

- The circuit uses D flip-flops and NAND gates for entering data (ie writing) to the register.
- D0, D1, D2 and D3 are the parallel inputs, where D0 is the most significant bit and D3 is the least significant bit.
- To **write** data in, the **mode control line is taken to LOW** and the data is clocked in.
- The data can be **shifted** when the **mode control line is HIGH** as SHIFT is active high.

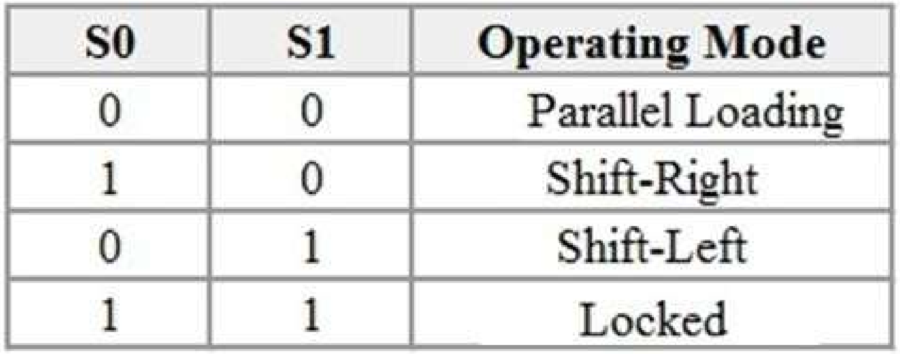


A 4-bit parallel-in–serial-out shift register.

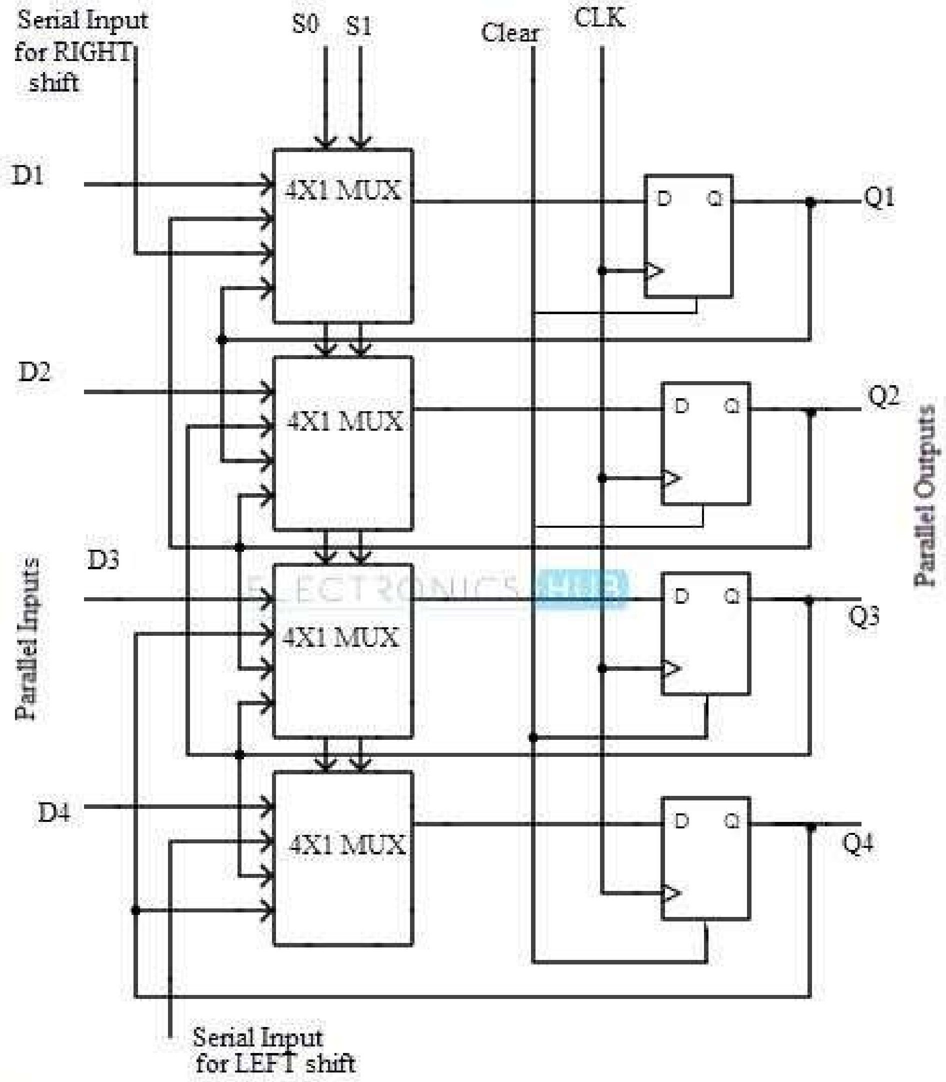
	Q_0	Q_1	Q_2	Q_3	
Clear	0	0	0	0	
Write	1	0	0	1	
Shift	1	0	0	1	
	1	1	0	0	1
	1	1	1	0	01
	1	1	1	1	001
	1	1	1	1	1001

Universal Shift Register

- The universal shift register can be defined as “The register which can be used to shift the data in both the directions like left, right and can load parallel data as well”.
- This register can perform three types of operations, stated below.
  - Parallel loading
  - Shifting left
  - Shifting right.



S0	S1	Operating Mode
0	0	Parallel Loading
1	0	Shift-Right
0	1	Shift-Left
1	1	Locked



Applications of Shift Register

- Delay line
- Serial to parallel converter
- Parallel to serial converter
- Shift register counter
- Sequence generator