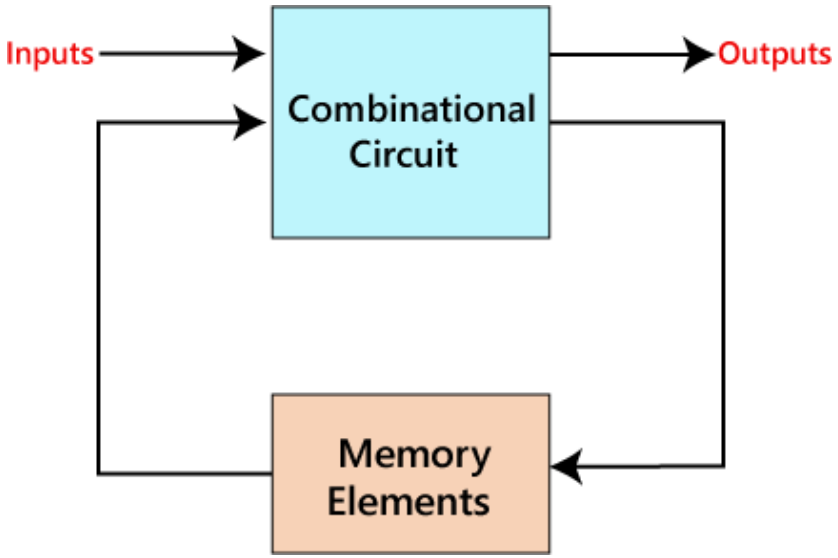
Module 5
Sequential Circuits

Introduction

- Sequential logic circuits are those, whose output depends on both present and past values of their inputs.
- Sequential circuits act as storage elements and have memory.
- They can store, retain, and then retrieve information when needed at a later time.
- It can be considered as combinational circuit with feedback.
- It uses a memory element like flip – flops as feedback circuit in order to store past values. The block diagram of a sequential logic is shown below.



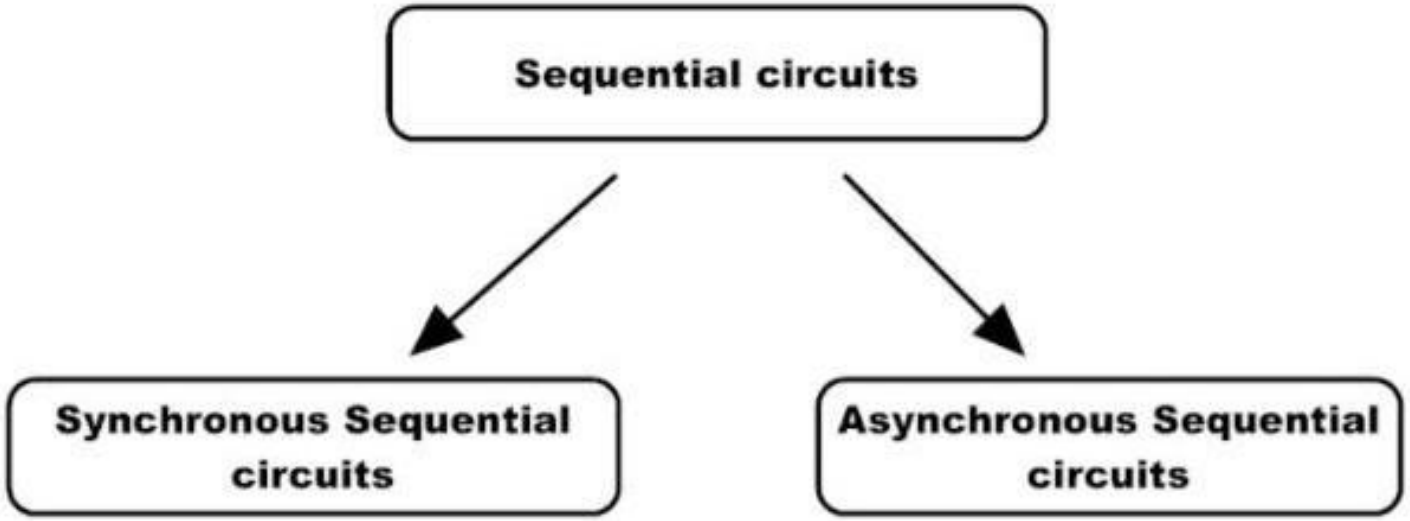
- The block diagram consists of a combinational circuit to which storage elements
  are connected to form a feedback path.
- The storage elements are devices capable of storing binary information.
- The binary information stored in these elements at any given time defines the “State” of the sequential circuit at that time.
- The external inputs also determine the condition for changing the state in the
  storage elements.
- The output in a sequential circuit are a function of the inputs and present state ofthe storage elements.
- The next state of the storage elements is also a function of external inputs and the present state.
- Thus, a Sequential circuit is specified by a time sequence of inputs, outputs, and internal states.

Combinational Circuits vs Sequential Circuits

COMBINATIONAL CIRCUITS	SEQUENTIAL CIRCUITS
Output depends only on the present value of the inputs.	Output depends on both the present and previous state values of the inputs
These circuits will not have any memory as their outputs change with the change in the input value.	Sequential circuits have some sort of memory as their output changes according to the previous and present values.
There are no feedbacks involved.	In a sequential circuit the outputs are connected to it as a feedback path.
Used in basic Boolean operations.	Used in the designing of memory devices.
Implemented in: Half adder circuit, full adder circuit, multiplexers, demultiplexers, decoders and encoders.	Implemented in: RAM, Registers, counters and other state retaining machines.

Classification of Sequential Circuits

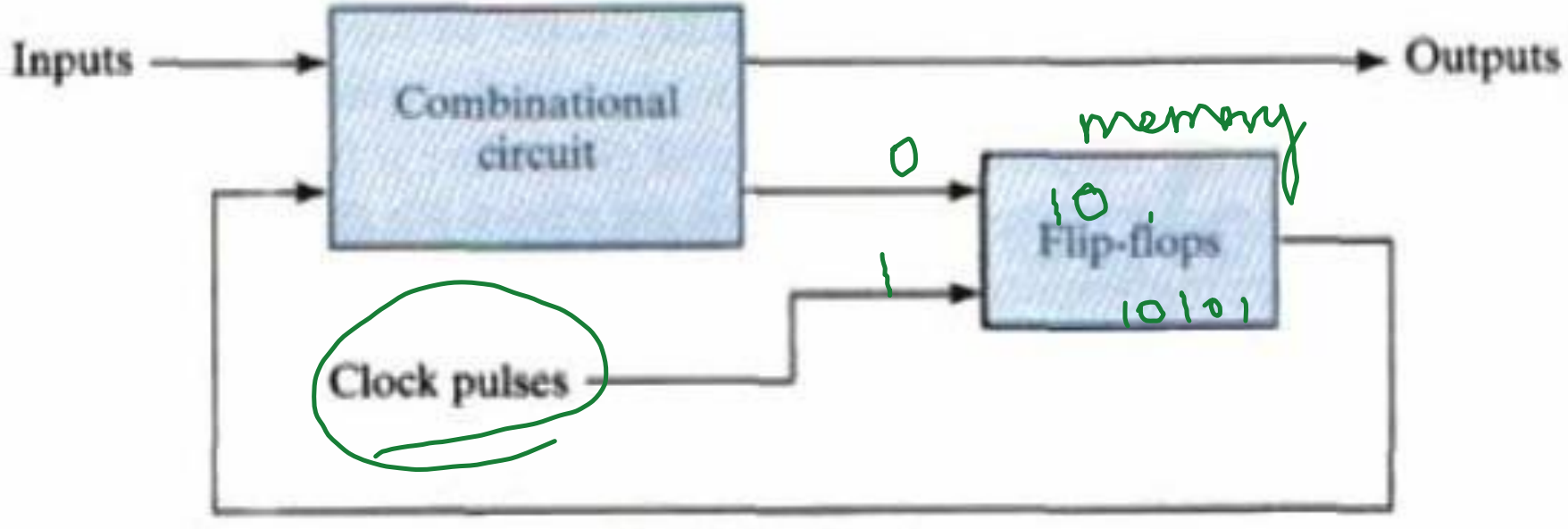
- Based on the clock signal input, the sequential circuits are classified into two types.
- Synchronous sequential circuit
- Asynchronous sequential circuit

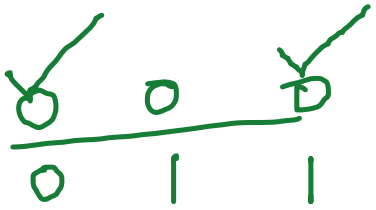


Synchronous sequential circuit

- It is a system whose behaviour can be defined from the knowledge of its signals at discrete instants of time.
- Synchronization is achieved by a timing device called a clock generator, producing clock pulses.
- The output depends on present and previous states of the inputs at the clocked instances.
- The circuits use a memory element to store the previous state. The memory elements in these circuits will have clocks. All these clock signals are driven by the same clock signal.
- Using clock signal, state changes will occur across all storage elements.
- These circuits are bit slower compared to asynchronous because they wait for the next clock pulse to arrive to perform the next operation.
- These circuits can be clocked or pulsed.

- The storage elements used in clocked sequential circuits are called as flip-flops.
- A flip-flop is a binary storage device capable of storing one bit of information.
- The Synchronous sequential circuits that use clock pulses in their inputs are called clocked-sequential circuits. They are very stable.
- The sequential circuits that change their state using the pulse and these are called pulsed or un-clocked sequential circuits.





- **Applications of Synchronous Sequential circuits:**
- Used in the design of MOORE-MEALY state machines.

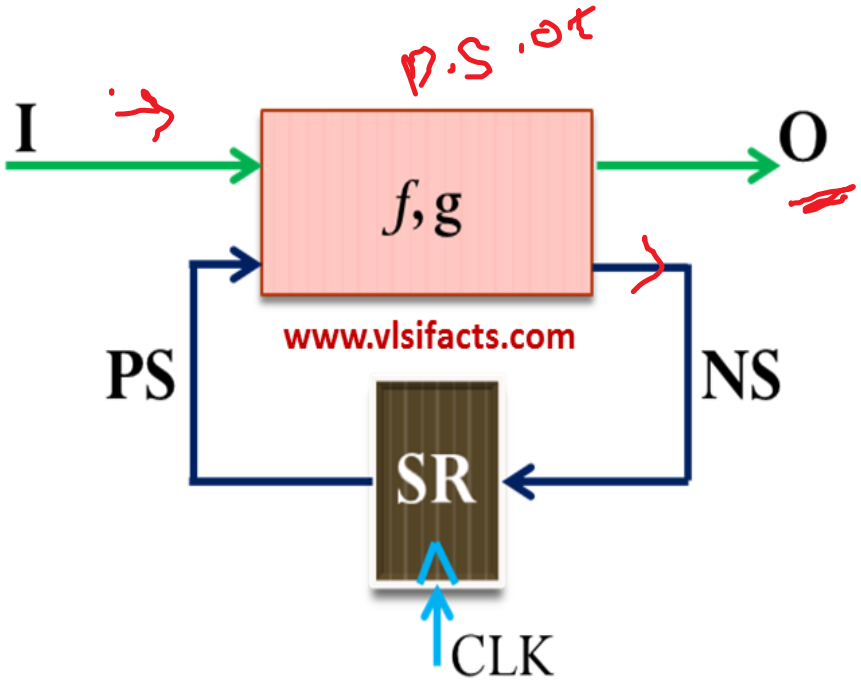
What is a Moore state machine? **A state machine in which the present state depends only on its previous input and previous state, and the present output depends only on the present state.**
A Mealy Machine changes its output on the basis of its present state and current input.

- They are used in synchronous counters, flip flops etc.
- **Limitations of Synchronous Sequential Circuits**
- All the flip – flops in synchronous sequential circuits must be connected to clock signal. Clock signals are very high frequency signals and clock distribution *consumes and dissipates a large amount of heat.*

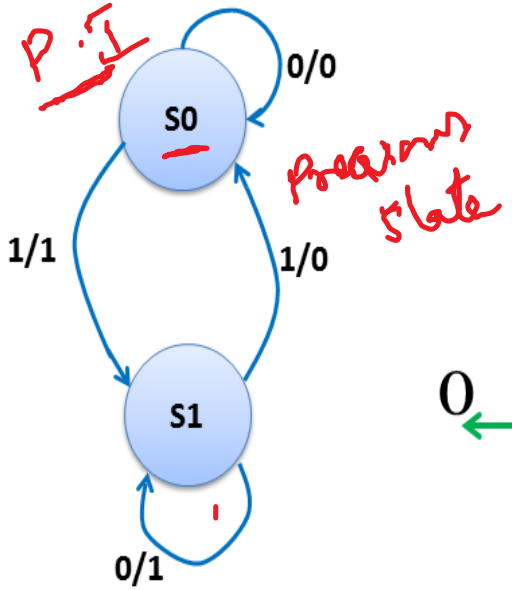
- Critical path or the slowest path determines the maximum possible clock frequency. Hence they are slower than asynchronous circuits.

Mealy Machine

In case of Mealy machine, output is a function of not only the present inputs but also past inputs. In other words we can say; in case of Mealy, both output and the next state depends on the present input and the present state.



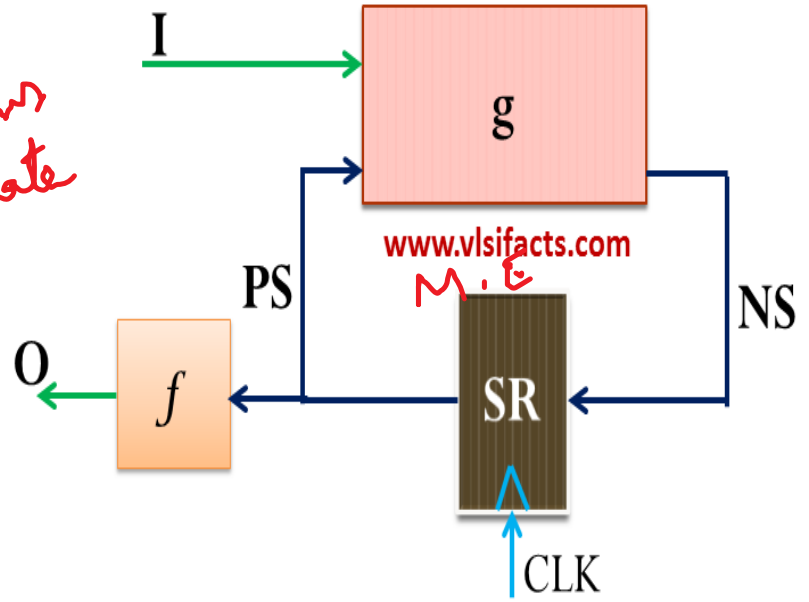
(a)



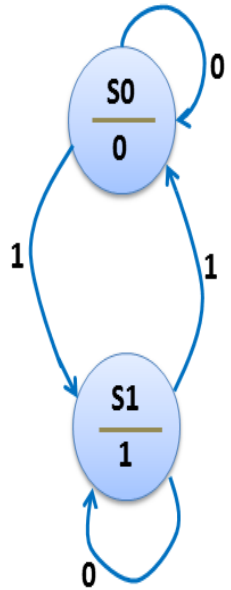
(b)

Moore Machine

In case of Moore machine, present output is not a function of present inputs but is a function of past inputs. The next state is a function of both the present input and the present state.



(a)



(b)

Asynchronous Sequential Circuits

- The Sequential circuits which do not operate by clock signals are called “Asynchronous sequential circuits”.
- The behaviour of the circuit depends upon the input signals at any instant of time and the order in which the inputs change.
- The storage elements used are time delay devices. (latches)
- They do not operate in pulse mode.
- They have better performance but hard to design due to timing problems.
- Mostly we use the asynchronous circuits when we require the low power operations.
- They are faster than synchronous sequential circuits as they do not need to wait for any clock signal.

- **Applications of Asynchronous Sequential Circuits**
- These are used when speed of operation is important. As they are independent of internal clock pulse, they operate quickly. so they are used in Quick response circuits.
- Used in the communication between two units having their own independent clocks.
- Used when we require the better external input handling.

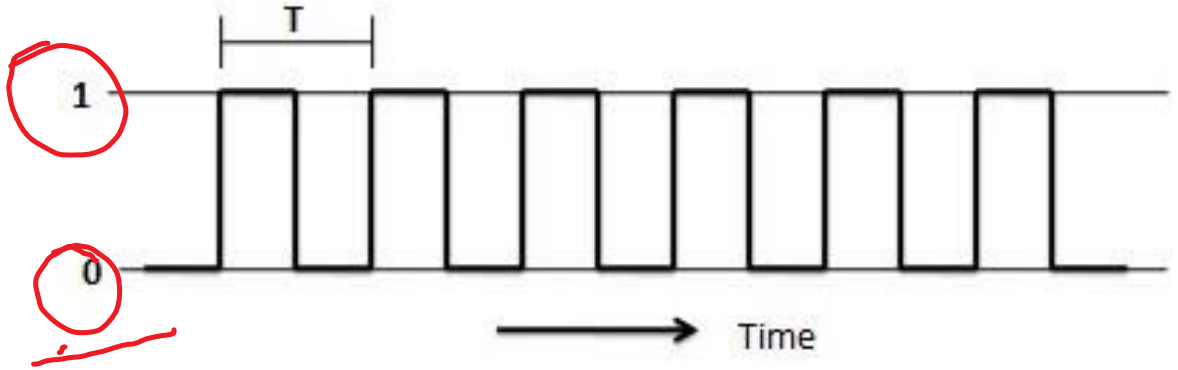
- **Limitations of Asynchronous Sequential Circuits**
- Asynchronous sequential circuits are more difficult to design.
- Though they have a faster performance, their output is uncertain.

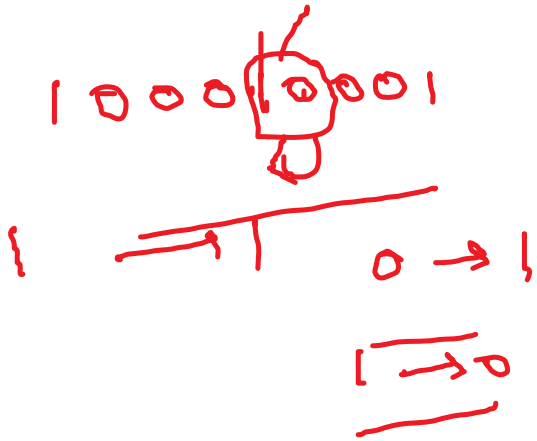
Difference between Flip-flop and Latch :

Flip-flop	Latch
Flip-flop is a bistable device i.e., it has two stable states that are represented as 0 and 1.	Latch is also a bistable device whose states are also represented as 0 and 1.
It checks the inputs but changes the output only at times defined by the clock signal or any other control signal.	It checks the inputs continuously and responds to the changes in inputs immediately.
It is a edge triggered device.	It is a level triggered device.
Gates like NOR, NOT, AND, NAND are building blocks of flip flops.	These are also made up of gates.
They are classified into asynchronous or synchronous flipflops.	There is no such classification in latches.
It forms the building blocks of many sequential circuits like counters.	These can be used for the designing of sequential circuits but are not generally preferred.
a, Flip-flop always have a clock signal	Latches doesn't have a clock signal
Flip-flop can be build from Latches	Latches can be build from gates
ex:D Flip-flop, JK Flip-flop	ex:SR Latch, D Latch

Clock Signal in Sequential Circuits

- A clock is a signal, which oscillates between logic level 0 and logic level 1, repeatedly.
- Square wave with constant frequency is the most common form of clock signal.
- A clock signal has “edges”. These are the instants at which the clock changes from 0 to 1 (a positive edge) or from 1 to 0 (a negative edge).





- Clock signals control the outputs of the sequential circuit . It determines when and how the memory elements change their outputs .

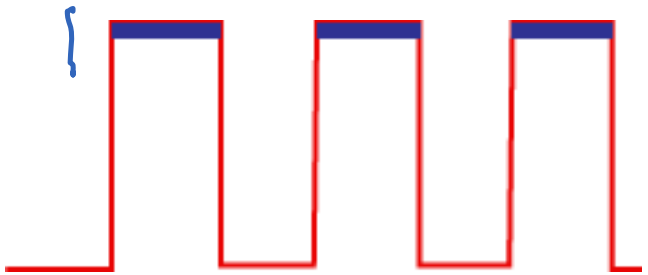
Triggering

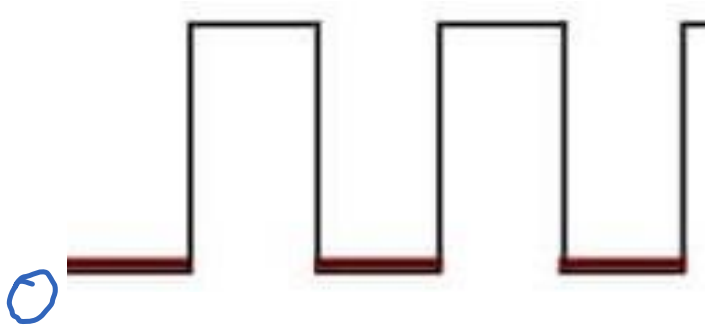
- The change in output of a flip flop can be done by bringing a small change in the input signal. This small change can be done with the help of a clock pulse. This clock pulse is known as a Trigger pulse.
- A flip – flop is said to be “Triggered”, when a trigger pulse is applied to the input that brings changes in the output.
- Flip – flops are basic components in registers and counters, which store data in the form of multi – bit numbers.
- Number of flip – flops are connected to form a sequential circuit and all these flip – flops require trigger pulse.
- The number of trigger pulses applied to the input determines the number in a counter.

There are two type of triggering: Level Triggering and Edge Triggering

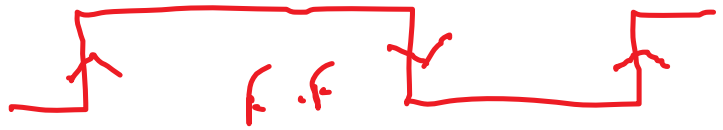
Level Triggering

- The triggering process in which the change in the output state is according to the active level of inputs is called “Level Triggering”.
- Level triggering is of two types - High level and Low level triggering.
- In High Level Triggering, the output of the flip – flop changes only when its enable input is at a high state i.e. logic high or logic 1.
- In Low Level Triggering, the output of the flip – flop changes only when its enable input is at a low state i.e. logic low or logic 0.

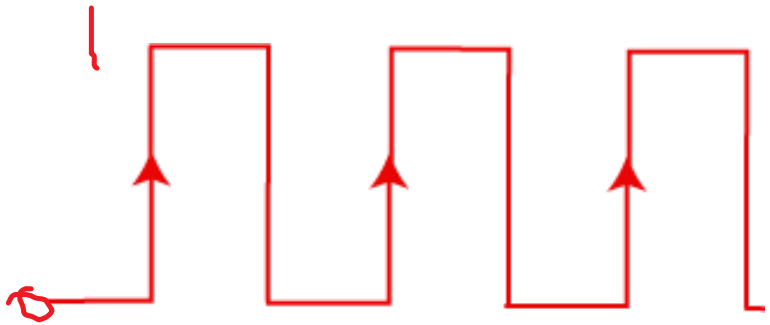


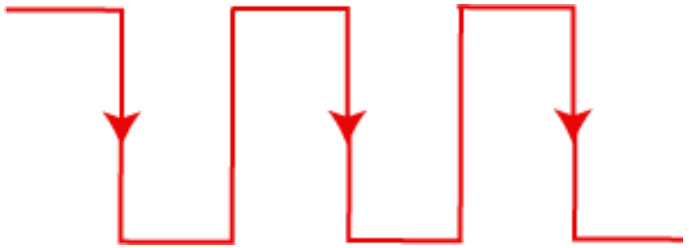


Edge triggering



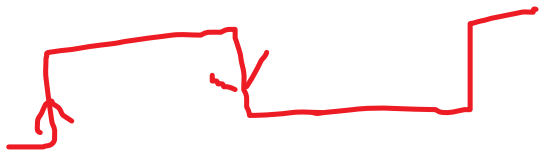
- In Edge Triggering, the output changes only when the inputs are present at either of the transitions of the clock pulse i.e. either from low to high (0 to 1) or from high to low (1 to 0).
- Edge triggering is of two types - Positive and Negative edge triggering.
- In Positive Edge Triggering, the output changes only when the input is at the positive edge of the clock pulse input i.e. a transition from low to high (0 to 1).
- In Negative Edge Triggering, the output changes only when the input is at the negative edge of the clock pulse input i.e. a transition from high to low (1 to 0).





Storage Elements

- A storage element in a digital circuit can maintain a binary state indefinitely, until directed by an input signal to switch states.
- Storage elements that operate with signal levels are referred to as latches.
- Storage elements controlled by a clock transition are referred to as flip flops.
- Latches are said to be level sensitive devices
- Flip flops are edge sensitive devices.



- Latches are the basic building blocks for the construction of flip flops.
- Latches are useful for storing binary information and for the design of asynchronous circuits.

Latches

- Latch is an electronic logic circuit with two stable states i.e. it is a bistable multivibrator.
- Latch has a feedback path to retain the information. Hence a latch can be a memory device.
- Latch can store one bit of information as long as the device is powered on. When enable is asserted, latch immediately changes the stored information when the input is changed i.e. they are level triggered devices.
- It continuously samples the inputs when the enable signal is on.
- Latch circuits can work in two states depending on the triggering signal being high or low: Active – High or Active – Low.
- In case of Active – High latch circuits, normally both the inputs are low. The circuit is triggered by a momentary high on either of the inputs.
- In case of Active – Low latch circuits, normally both the inputs are high. The circuit is triggered by a momentary low on either of the inputs.

SR Latch

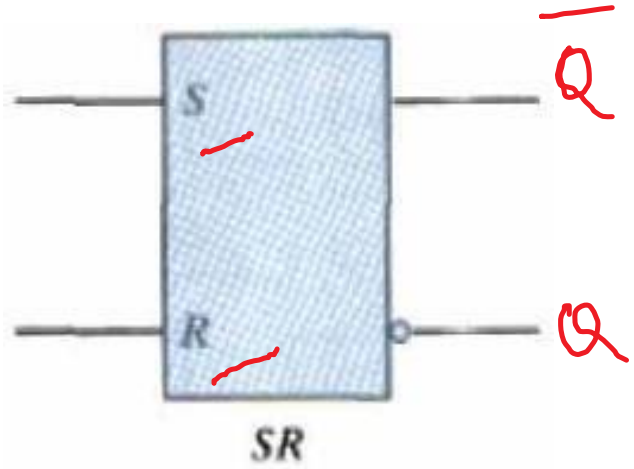
- **SR (Set-Reset) Latch –** SR Latch is a circuit with:

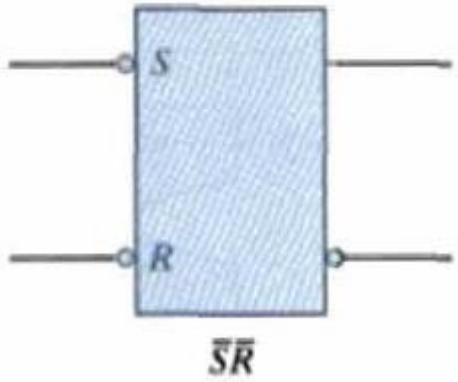
  (i) 2 cross-coupled NOR gate or 2 cross-coupled NAND gate.

  (ii) 2 input S for SET and R for RESET.

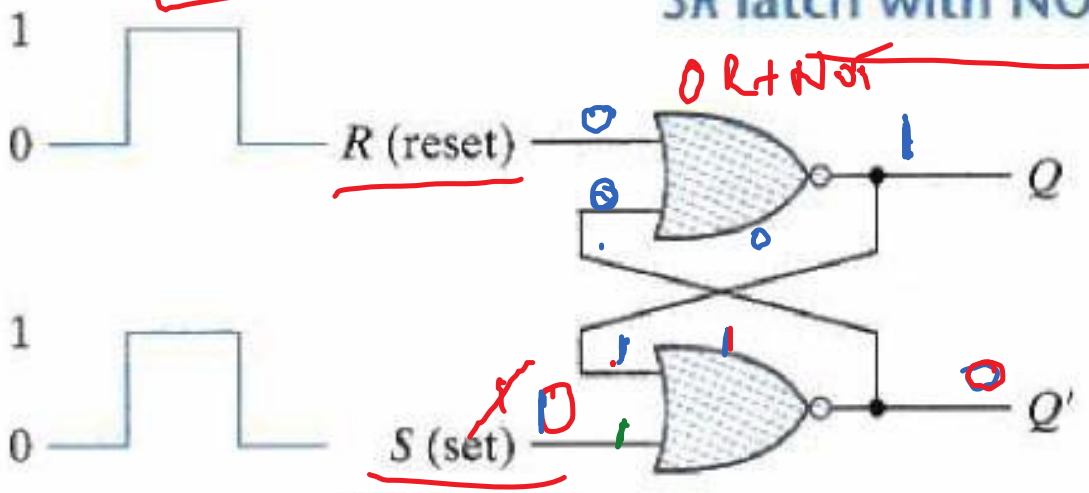
  (iii) 2 output Q, Q'.

Q	Q'	STATE
1	0	Set
0	1	Reset





SR latch with NOR gates

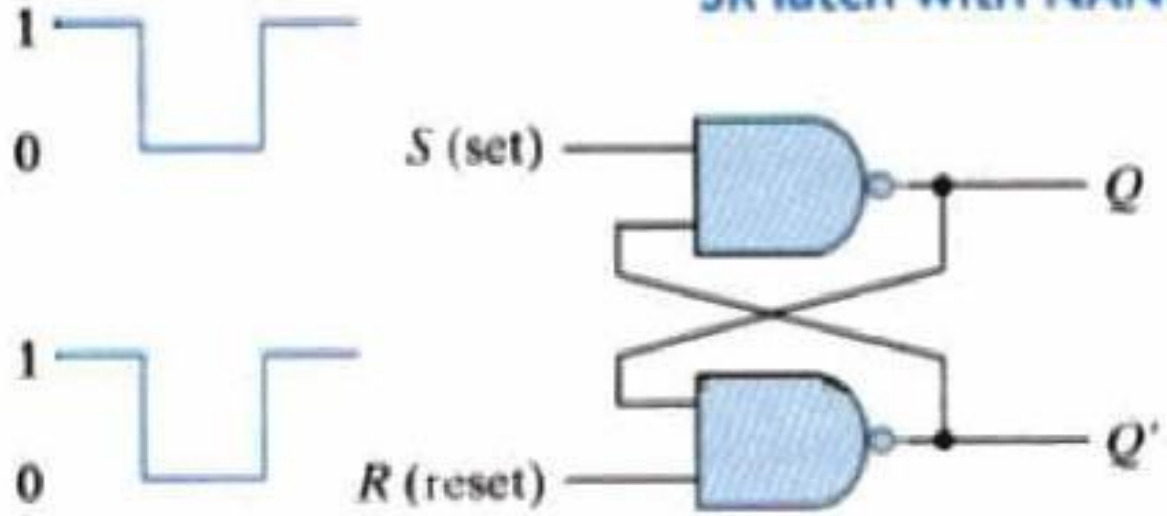


(a) Logic diagram

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

SR latch with NAND gates



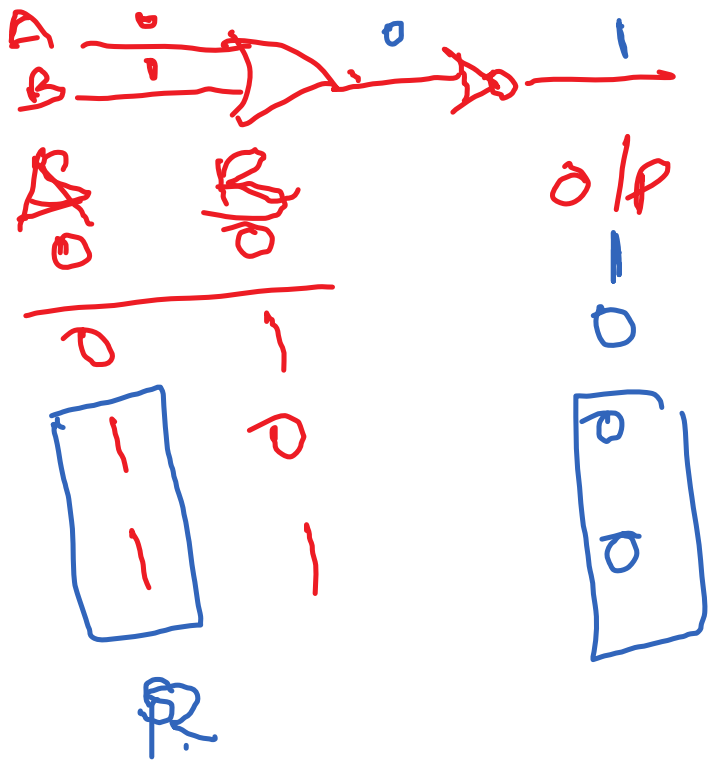
(a) Logic diagram

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)

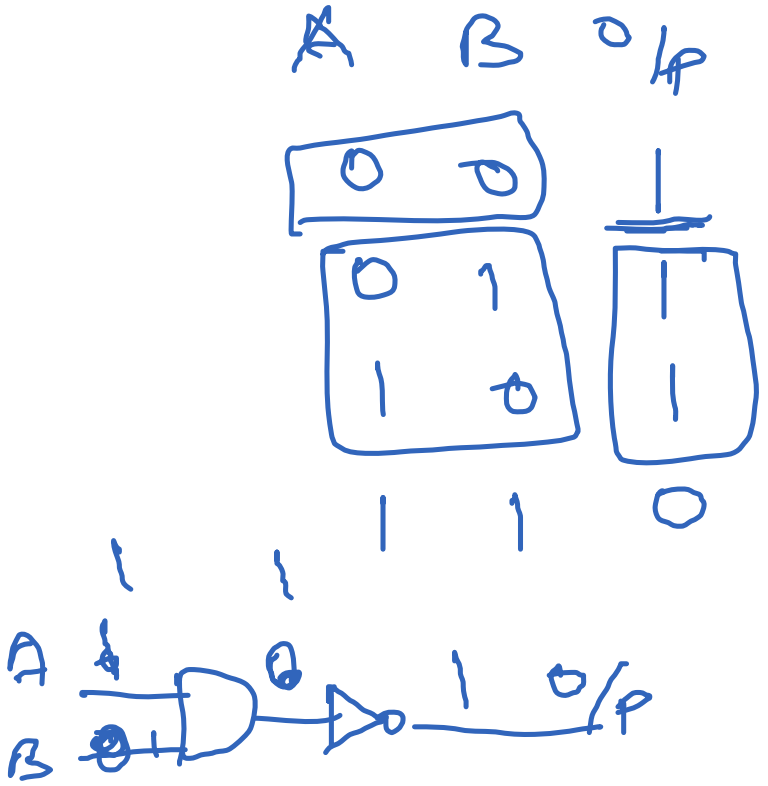
(b) Function table

S. R catcher.

NOR

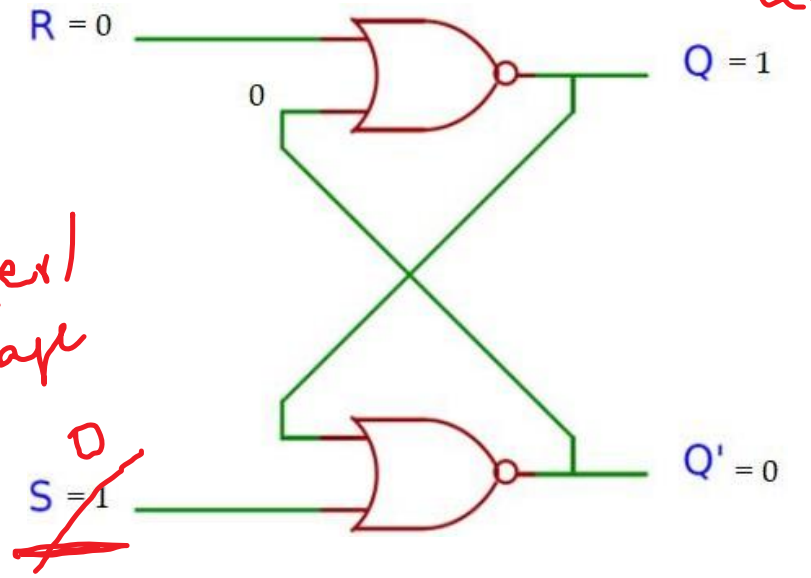


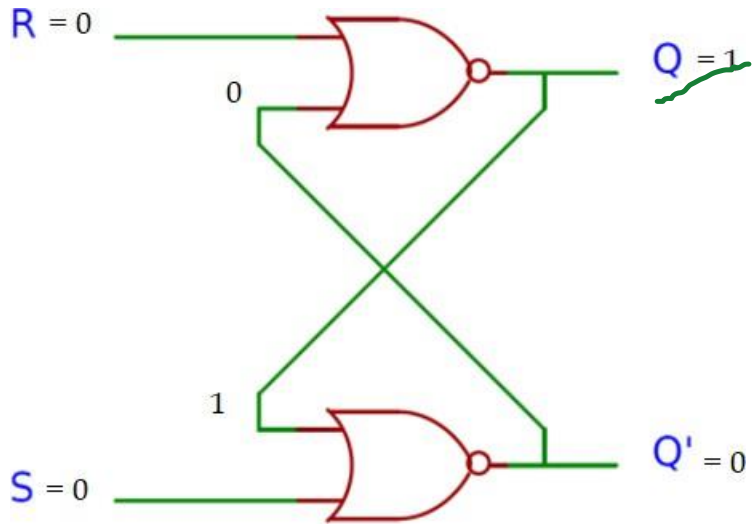
NAND gate



SR latch using NOR gates- Operation

- Under normal conditions, both the input remains 0.
- Case (i): R = 0, S = 1, Q=? and Q' =?





Truth table of NOR gate:

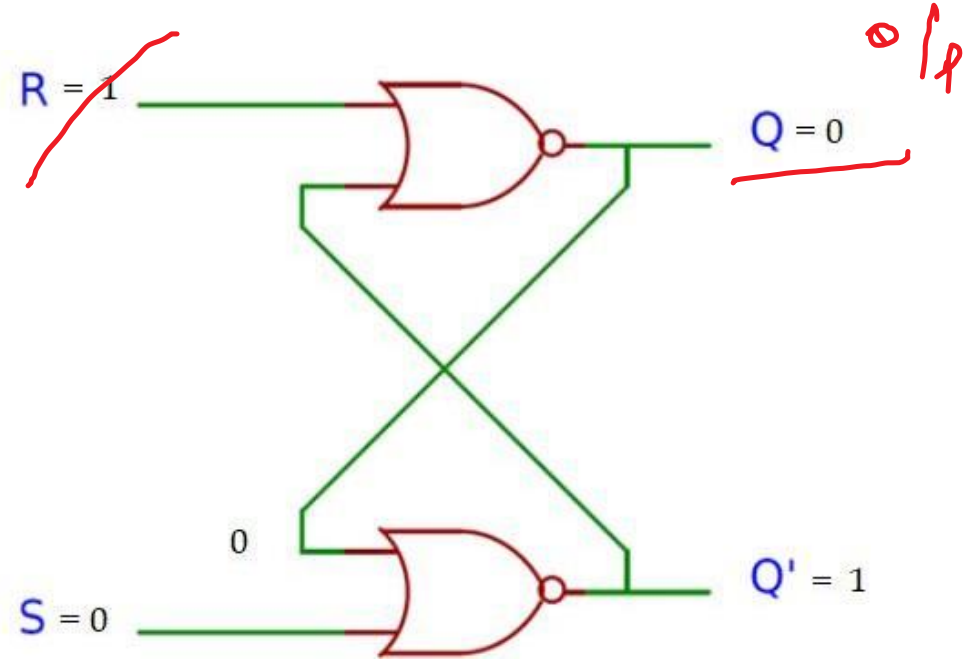
x	y	F
0	0	1
0	1	0
1	0	0
1	1	0

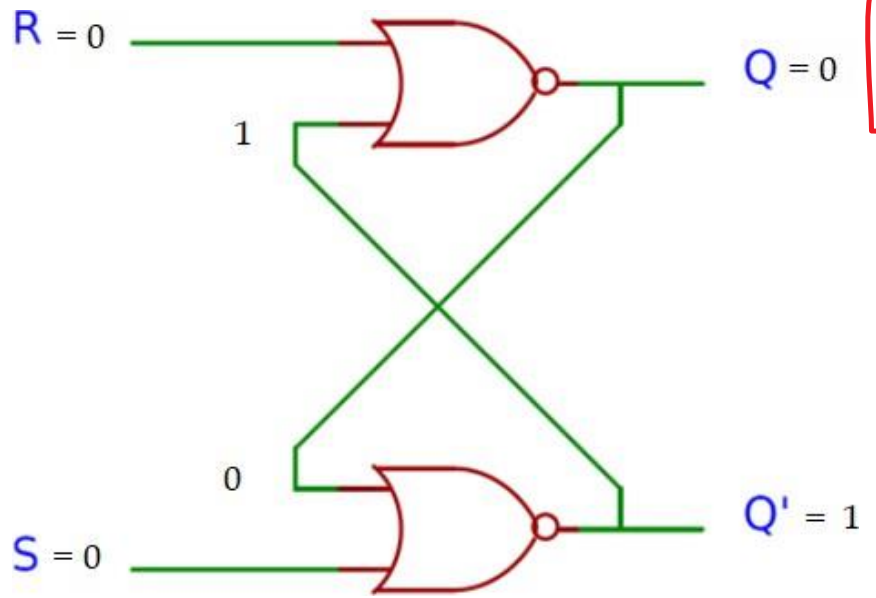
If any one of the input is 1, NOR gate output is 0. So, R = 0, S = 1, Q =1 And Q' = 0.

If the input S = 1 is removed, then S = 0, R = 0, Q =1 and Q' = 0.

No change in the output. It acts as a memory element.

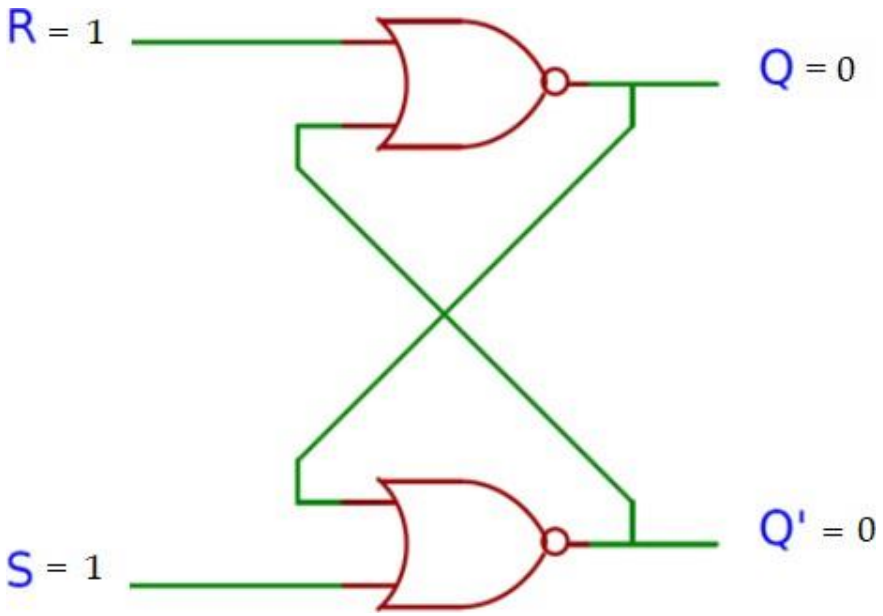
- Case (ii): R = 1, S = 0, Q=? and Q’ =?





- If any one of the input is 1, NOR gate output is 0. So, R = 1, S = 0, Q =0 And Q’ = 1.
- If the input R = 1 is removed, then R = 0, S = 0, Q = 0 and Q’ = 1.
- No change in the output. It acts as a memory element.

- Case (iii): R = 1, S = 1, Q=? and Q’ =?



S	R	Q	Q’
0	0	Memory	Element
0	1	0	1
1	0	1	0
1	1	Not	Used

RESET

SET

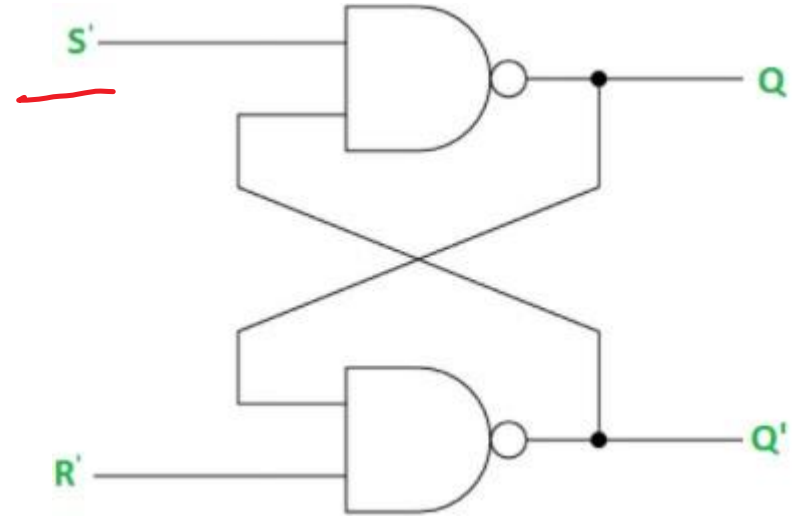
If any one of the input is 1, NOR gate output is 0. So, R = 1, S = 1, Q = 0 And Q’ = 0 (Not valid- Undefined state)

If both the inputs are removed, then R = 0, S = 0, Q = 0 and Q’ = 0.

Both the outputs are not complementing each other. This condition is not used in latches.

SR latch using NAND gates- Operation

- It operates with both inputs normally at 1.



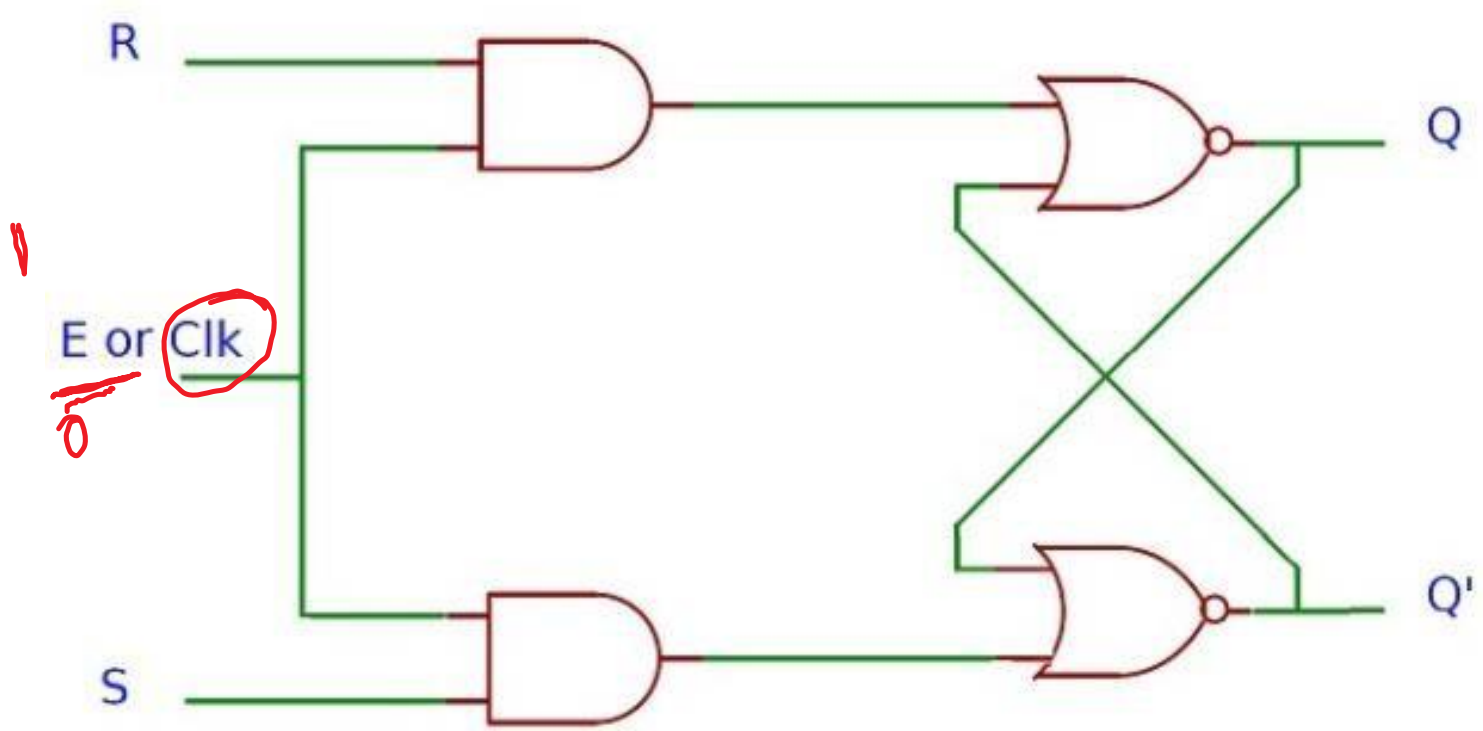
S’	R’	Q	Q’
0	0	Not	Used
0	1	1	0
1	0	0	1
1	1	Memory	Element

Note: In comparing the NAND with the NOR latch, the input signals for the NAND require the complement of those values used for the NOR latch.

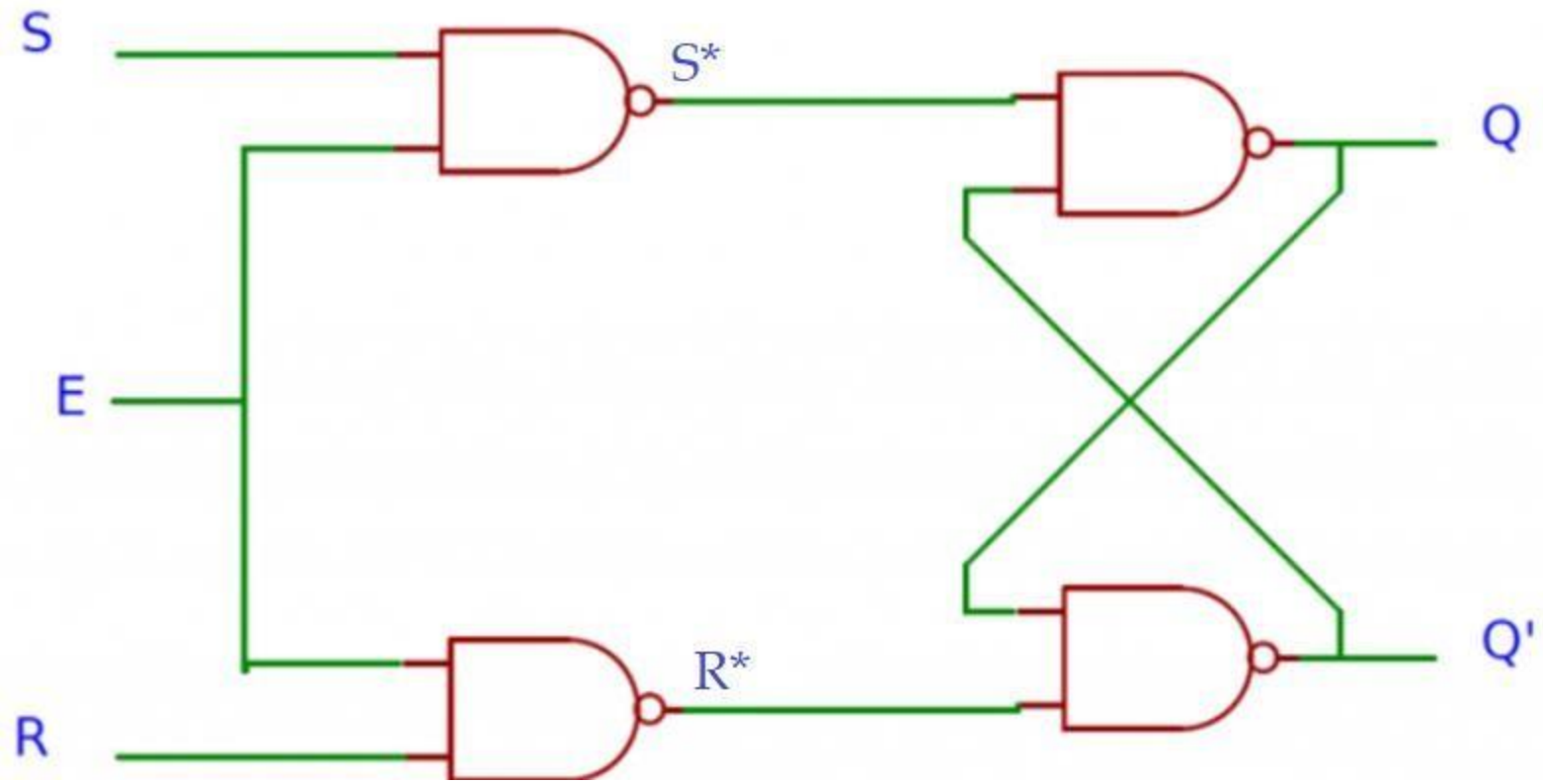
GATED S-R LATCH

- Generally, latches are transparent i.e. the output changes immediately when there is a change in the input.
- But for many applications, it is desirable to have an isolated period where the output doesn't change even when there is a change in the input. During this period, the outputs are said to be truly 'latched'.
- This can be achieved with the use of an extra input (enable or clock or gate). If the enable (or clock or gate) signal is not asserted, the inputs are ignored and the outputs are latched to the previous values.
- In order to use this extra signal, additional logic should be added. These circuits are called Gated or Clocked Latches.
- Gated SR latch can be made in two ways: by adding second level of AND gates to SR latch or by adding second level of NAND gates to S'R' latch (Inverted SR latch).

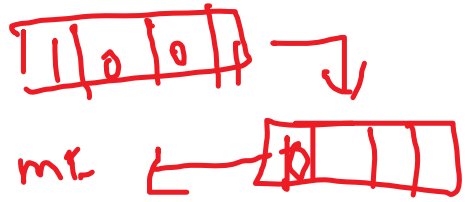
Gated SR latch constructed from NOR gates



Gated SR latch constructed from NAND gates



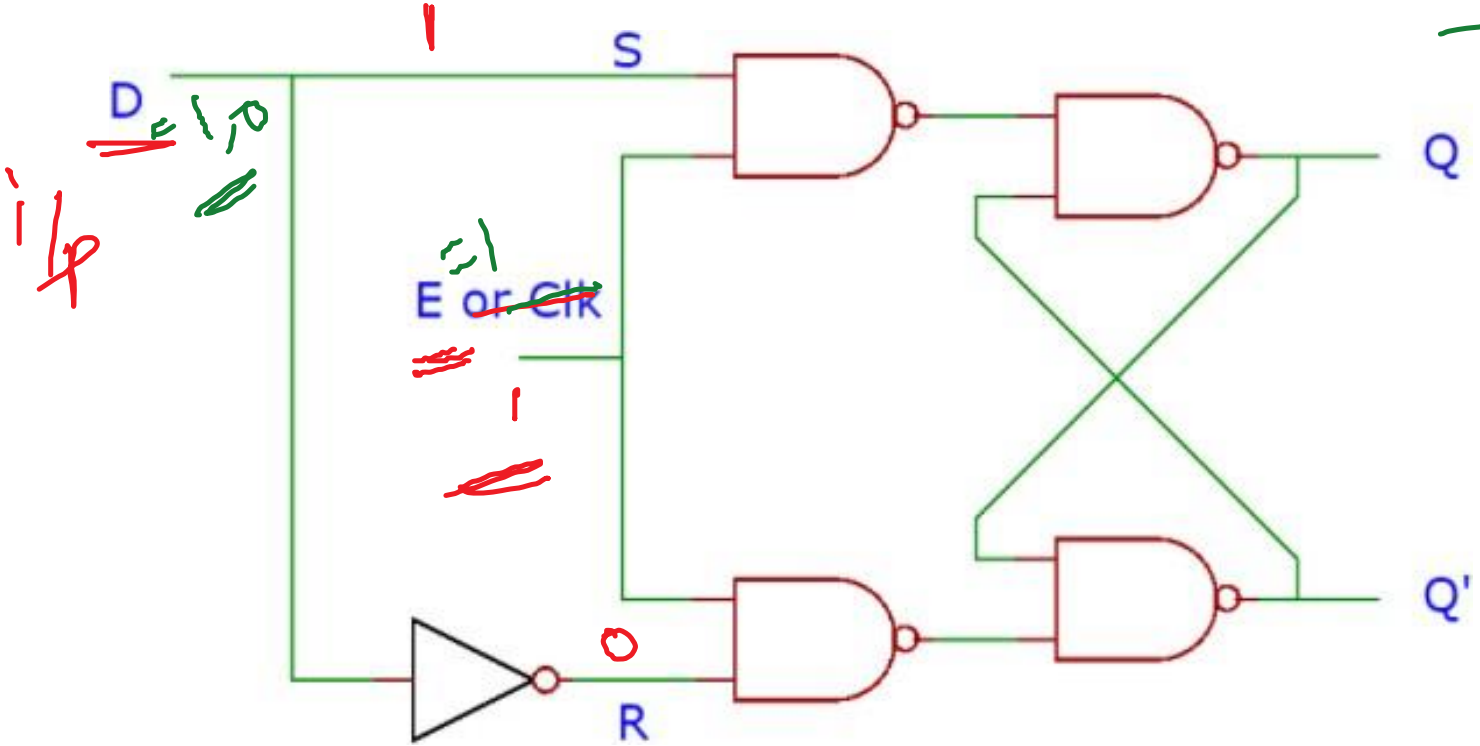
- The output S* and R* can be written as
- $S^* = (S.E)' = S' + E'$ and $R^* = (R.E)' = R' + E'$
- When E = 0, S* and R* becomes 1 and is dependent only on enable signal irrespective of the inputs. So, there is no change in the output. This is the quiescent condition for the SR latch.
- When E = 1, input from S or R affects the output of latch.
- When all the inputs are 1, output is in indeterminate state.
- This indeterminate state makes it difficult to use in practical case, but can be used for constructing other flip flops.

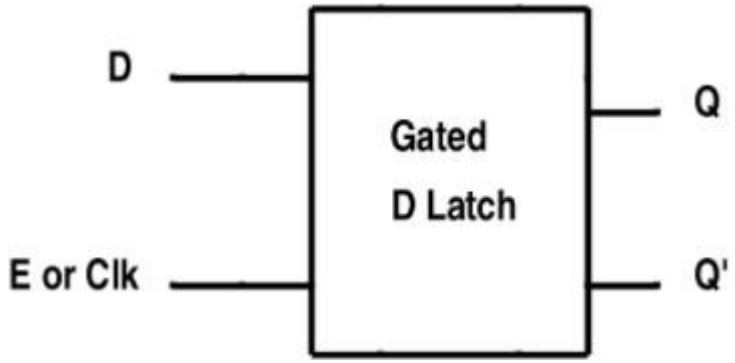


E	S	R	Next state of Q
0	X	X	No change
1	0	0	No change
1	0	1	Q = 0; RESET
1	1	0	Q = 1; SET
1	1	1	Indeterminate

D Latch (Transparent Latch)

- One way to eliminate the undesirable condition of the indeterminate state in the SR latch is to ensure that S and R are never equal to 1 at the same time.
- This is done in the D latch.
- This latch has only two inputs: D (data) and E (Enable).





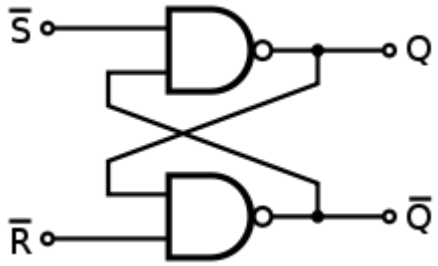
- The D input goes directly to the S input, and its complement is applied to the R input.
- When E = 0, the input to SR latch is 1 level and the circuit cannot change regardless of the value of D.
- When E= 1, and D = 1, the Q output goes to 1 (SET)
- When E= 1, and D = 0, the Q output goes to 0 (RESET), hence named as transparent latch.
- **Function table:**

E	D	Next state of Q
0	X	No change
1	0	Q = 0, RESET
1	1	Q = 1, SET

Difference between both….?

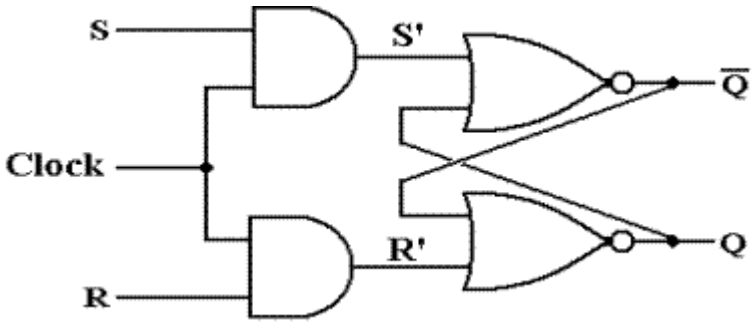
Latches..

- Latches are the building blocks of sequential circuits.
- latches can be built from gates.
- latch does not have a *clock signal.*



Flip Flop..

- flip-flops are also the building blocks of sequential circuits.
- Flip-flops can be built from latches.
- A flip-flop always has a *clock Signal*



What are flip flops

In electronics, a **flip-flop** or **latch** is a circuit that has two stable states and can be used to store state information. A flip-flop is a bistable multivibrator. The circuit can be made to change state by signals applied to one or more control inputs and will have one or two outputs. It is the basic storage element

in sequential logic.

Flip-flops maintain their state indefinitely until an input pulse called a trigger is received. When a trigger is received, the flip-flop outputs change state according to defined rules and remain in those states until another trigger is received.

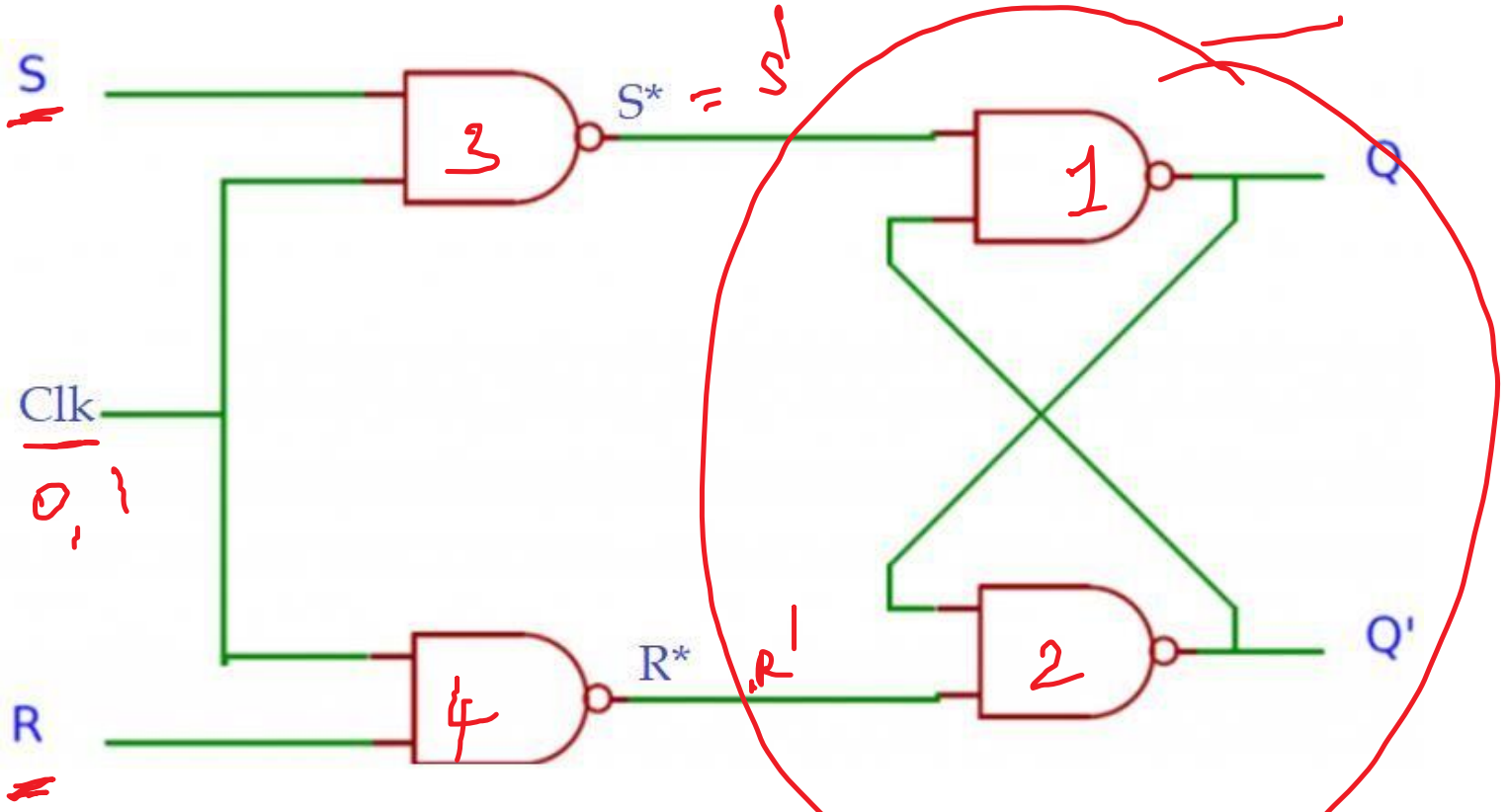
Uses of flip flops.

- Flip flop and latches are the circuits that can store and remember information. They're the kind of circuits that are used in computers to store program information like RAM memory and Registers.
- Flip-flops can be used to store one bit, or binary digit, of data. The data may represent the state of a sequencer, the value of a counter, an ASCII character in a computer's memory or any other piece of information.

Types Of flip flop

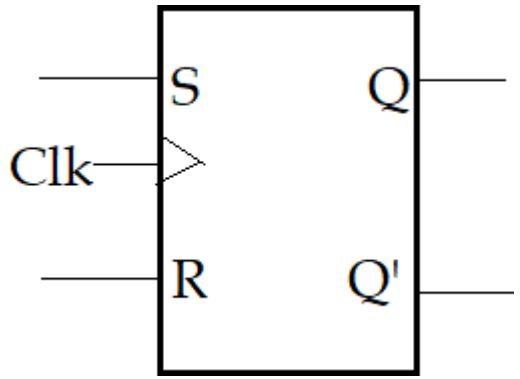
- RS flip-flop circuit = SR FF
- Clocked RS flip flop circuit
- D flip flop circuit
- Jk flip flop circuit
- T flip flop circuit

SR Flip Flop

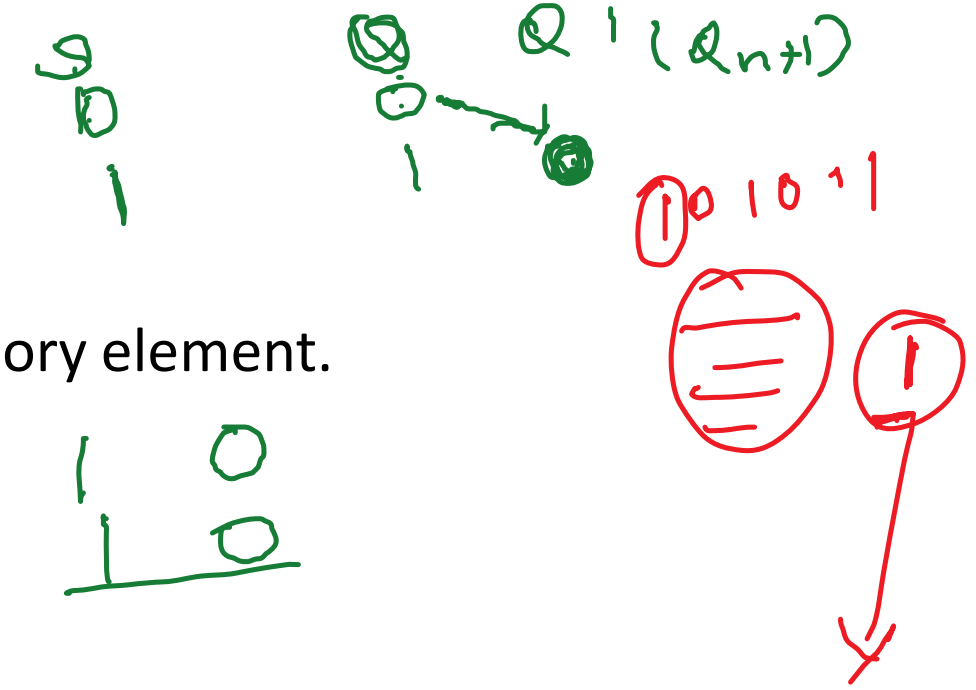


Truth table of SR latch

S*	R*	Q	Q'
0	0	Not	Used
0	1	1	0
1	0	0	1
1	1	Memory	Element



- The output S* and R* can be written as
- S* = (S.Clk)' = S' + Clk' and R* = (R.Clk)' = R' + Clk'
- When Clk = 0, S* = 1 , R* = 1, then SR FF acts like memory element.
- When Clk = 1, S*= S' , R* = R'



- Function Table:

Clk	S	R	Q	Q'
0	X	X	Memory	Element
1	0	0	Memory	Element
1	0	1	0	1
1	1	0	1	0
1	1	1	Not	Used

Truth Table:

Clk	S	R	Q(t+1)
0	X	X	Q(t)
1	0	0	Q(t)
1	0	1	0
1	1	0	1
1	1	1	Invalid

Characteristic Table:

Clk = 1

Q(t)	S	R	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

Excitation Table:

Q(t)	Q(t+1)	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

K map for Q(t+1):

Q(t) \ S R	00	01	11	10
0	0	0	X	1
1	1	0	X	1

$$Q(t+1) = S + Q(t)\,R'$$

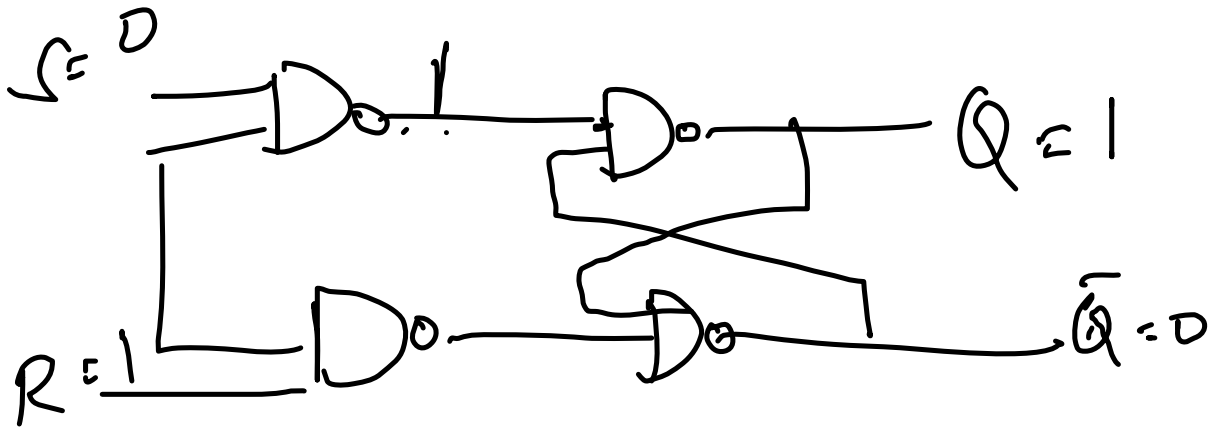
J-K F.F

(Q)
N.S

characteristic table

Clk	S	R	Q_{n+1}
0	X	X	Q(t) M.E
1	0	0	Q(t) M.E.
1	0	1	0
1	1	0	1
1	1	1	invalid

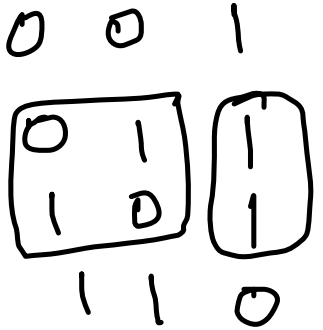
Exc. table



S R D J K

Conversion of F.F.

Counter



D Flip Flop

Truth Table:

Clk	D	Q(t+1)
0	X	Q(t)
1	0	0
1	1	1

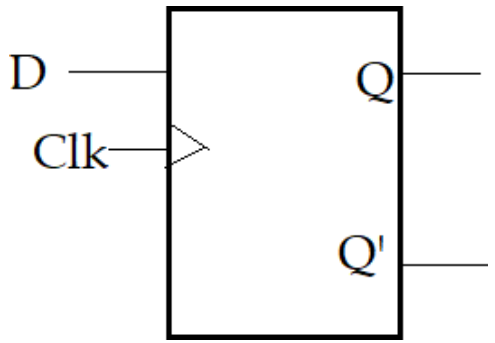
Characteristic Table:

Q(t)	D	Q(t+1)
0	0	0
0	1	1
1	0	0
1	1	1

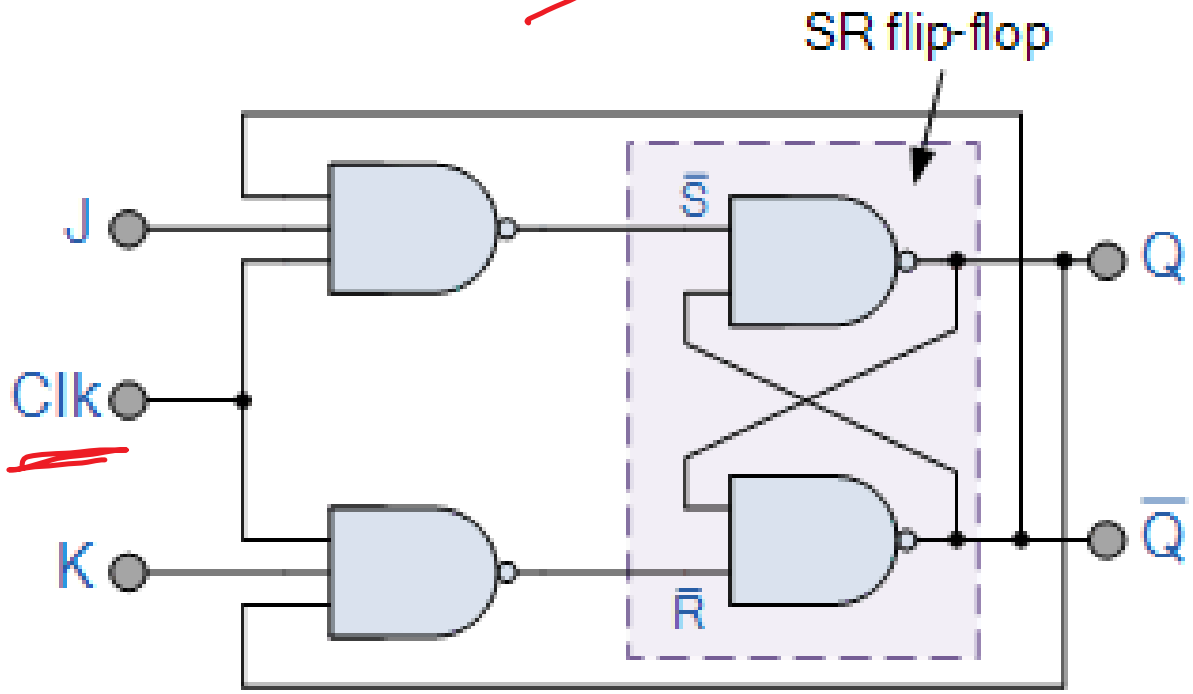
Q(t+1) = D

Excitation Table:

Q(t)	Q(t+1)	D
0	0	0
0	1	1
1	0	0
1	1	1



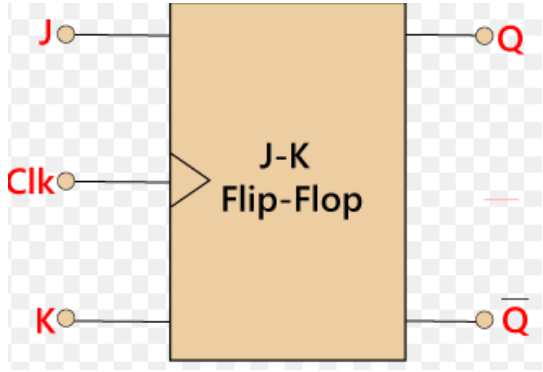
JK Flip Flop



Truth Table:

Clk	J	K	Q(t+1)
0	X	X	Memory; Q(t)
1	0	0	Memory; Q(t)
1	0	1	0
1	1	0	1
1	1	1	Toggle; Q(t)'

- The truth table for SR and JK flip flops are same for the first four combinations.
- In SRFF, S=1 and R=1, Q(t+1) is invalid
- Whereas in JK FF, J=1 and K =1, Q(t+1) = Q(t)'



Truth Table:

Clk	J	K	Q(t+1)
0	X	X	Q(t)
1	0	0	Q(t)
1	0	1	0
1	1	0	1
1	1	1	Q(t)'

Characteristic Table:

Q(t)	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

Excitation Table:

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

K map for J:

Q(t) \ Q(t+1)	0	1
0	0	1
1	X	X

J = Q(t+1)

K map for K:

Q(t) \ Q(t+1)	0	1
0	X	X
1	1	0

K = Q(t+1)'

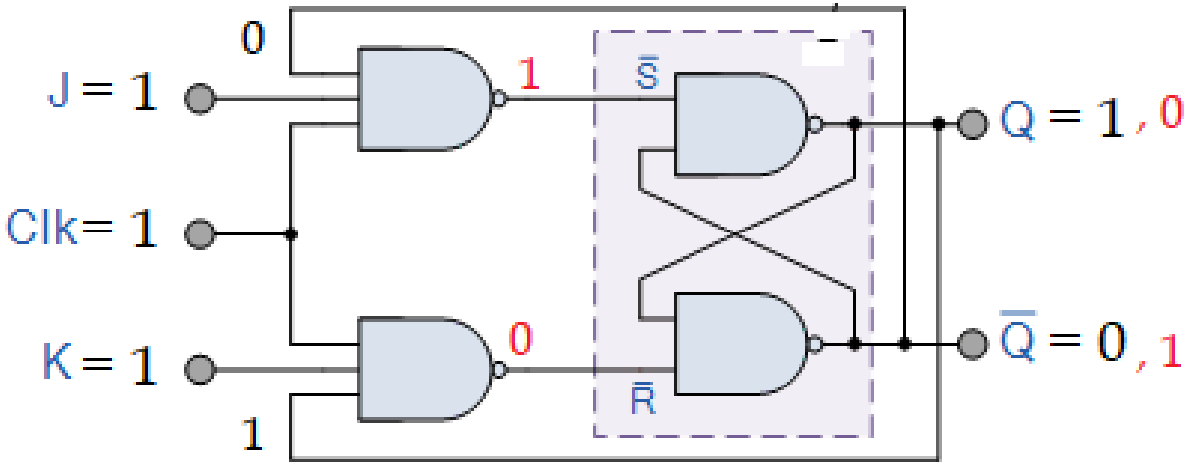
K map for Q(t+1):

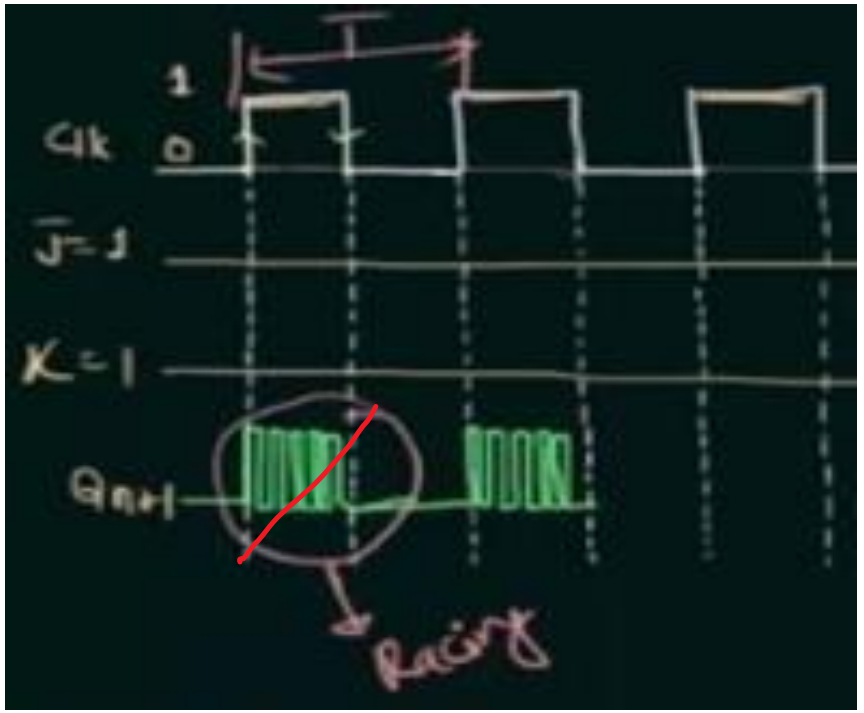
Q(t) \ J K	00	01	11	10
0	0	0	1	1
1	1	0	0	1

Q(t+1) = J.Q(t)' + K'.Q(t)

Clk	J	K	Q(t+1)	Q(t+1)'
0	X	X	Q(t)	Q(t)'
1	0	0	Q(t)	Q(t)'
1	0	1	0	1
1	1	0	1	0
1	1	1	?	?

Race around Condition:



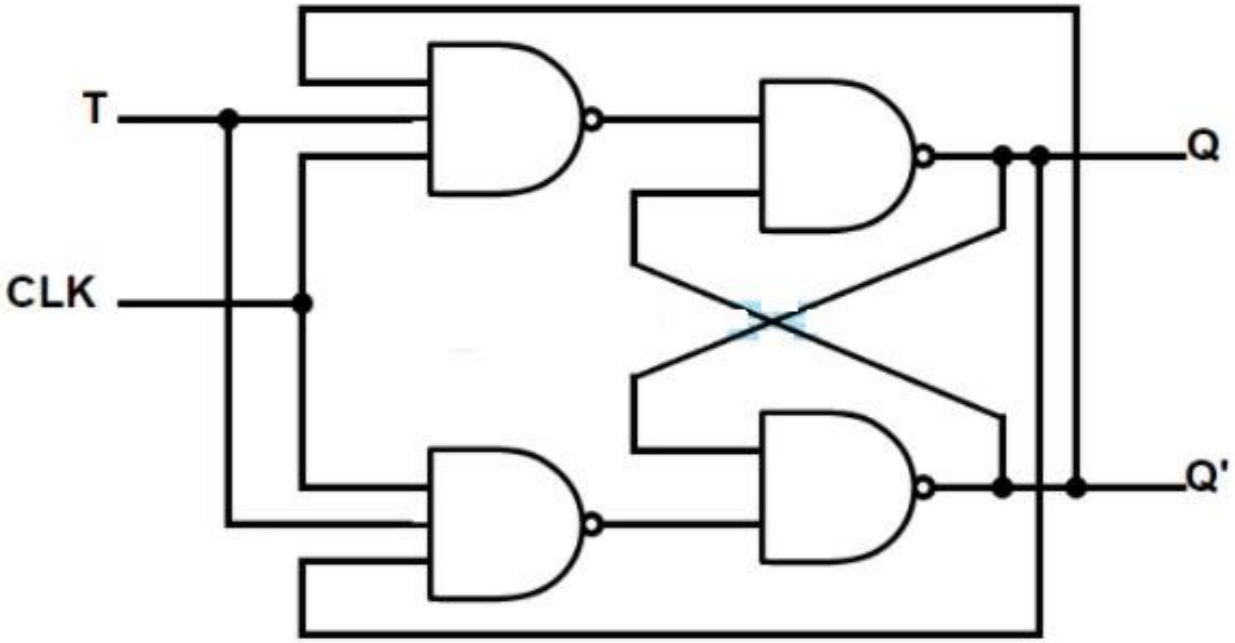


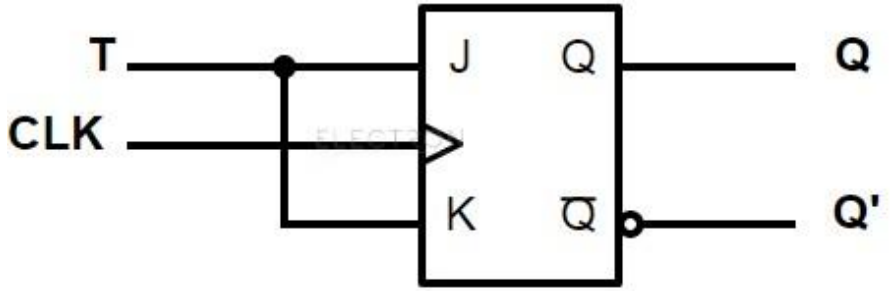
Condition to overcome Racing:

1) T/2 < propagation delay of FF.
2) Use edge triggering
3) Master – Slave operation is same as –ve edge triggering

T Flip Flop

- T flip – flop is also known as “Toggle Flip – flop”.
- To avoid the occurrence of invalid state in SR flip – flop, one input is given to the flip – flop called the Trigger input or Toggle input (T).
- Then the flip – flop acts as a Toggle switch. Toggling means ‘Changing the next state output to complement of the present state output’.





Clk	T	Q(t+1)
0	X	Q(t); Memory
1	0	Q(t); Memory
1	1	Q(t)’; Toggling

Truth Table:

Clk	T	Q(t+1)
0	X	Q(t)
1	0	Q(t)
1	1	Q(t)’

Characteristic Table:

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

Excitation Table:

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

$Q(t+1) = Q(t) \oplus T$

$T = Q(t) \oplus Q(t+1)$

Odd 1’s detector

Flip Flop Conversions

- Steps:

1. Identify the available and required flip flop
2. Make the excitation table for available flip flop
3. Make the characteristic table for required flip flop
4. Write the Boolean expression for available flip flop
5. Draw the circuit.

JK to D Flip Flop

1. Available FF = JK & Required FF = D

2. Excitation table for JK FF:

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

3. Characteristic table for D FF:

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

4. Boolean expression for available JKFF:

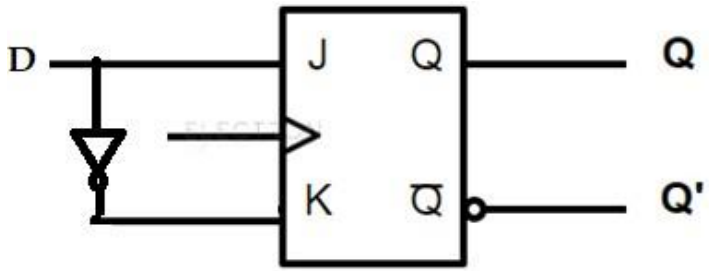
Q(t) \ D	0	1
0	0	1
1	X	X

J = D

Q(t) \ D	0	1
0	X	X
1	1	0

K = D’

5. Circuit Diagram:



SR to JK Flip Flop

1. Available FF = SR FF ; Required FF = JK FF

2. Excitation table of SR FF:

Q(t)	Q(t+1)	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

3. Characteristic table for JK FF:

Q(t)	J	K	Q(t+1)	S	R
0	0	0	0	0	X
0	0	1	0	0	X
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	1	X	0
1	0	1	0	0	1
1	1	0	1	X	0
1	1	1	0	0	1

4. Boolean expression for available SR FF:

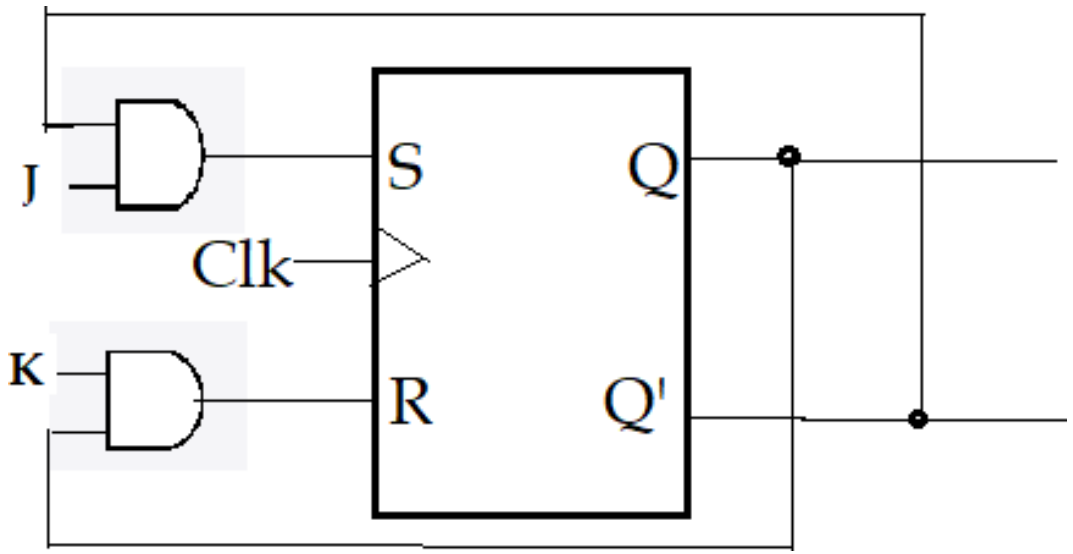
Q(t) \ J K	00	01	11	10
0	0	0	1	1
1	X	0	0	X

$S = J\,Q(t)'$

Q(t) \ J K	00	01	11	10
0	X	X	0	0
1	0	1	1	0

$R = K\,Q(t)$

5. Circuit Diagram:



Characteristic Equations and Table

JK Flip-Flop

J	K	Q(t + 1)	
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	Complement

T Flip-Flop

T	Q(t + 1)	
0	$Q(t)$	No change
1	$Q'(t)$	Complement

D Flip-Flop

D	Q(t + 1)	
0	0	Reset
1	1	Set

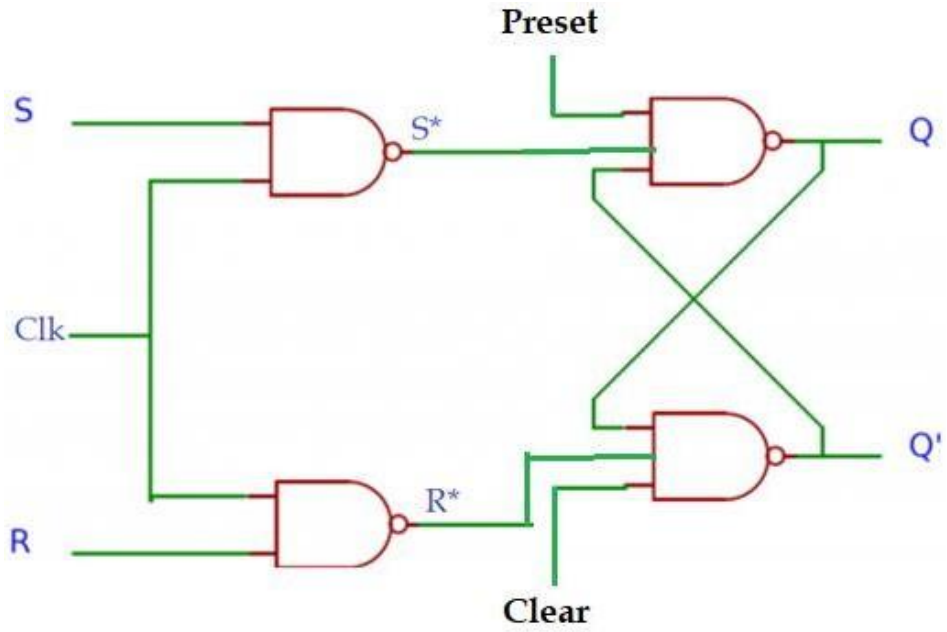
JK FF $$Q(t + 1) = JQ' + K'Q$$

DFF $$Q(t + 1) = D$$

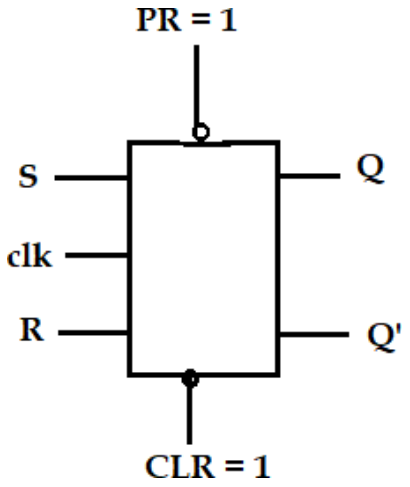
TFF: $$Q(t + 1) = T \oplus Q = TQ' + T'Q$$

Preset and Clear Inputs

- They are the direct inputs or overriding inputs or asynchronous inputs (preset and clear).
- The synchronous inputs are S, R, J , K, D and T.
- Preset = 0 -----→ Q = 1
- Clear = 0 ---------→ Q = 0
- Whatever be the value of clock and synchronous inputs



Preset	Clear	Q(t)
0	0	Not used
0	1	1
1	0	0
1	1	FF perform normally



Analysis of Clocked Sequential Circuits

Introduction

- Analysis describes the behaviour of a clocked sequential circuits under certain operating conditions.
- It can be determined from the inputs, outputs and state of flip flops.
- The outputs and the next state are both a function of the inputs and the present state.
- The analysis of a sequential circuit obtained by state table or a state diagram.
- It is also possible to write Boolean expressions that describes the behaviour of the sequential circuits.
- A logic diagram is recognized as a clocked sequential circuit it includes flip flops with clock inputs.
- The FF may be of any type and the logic diagram may or may not include combinational logic gates.

Introduction to State equation, State table and State diagram

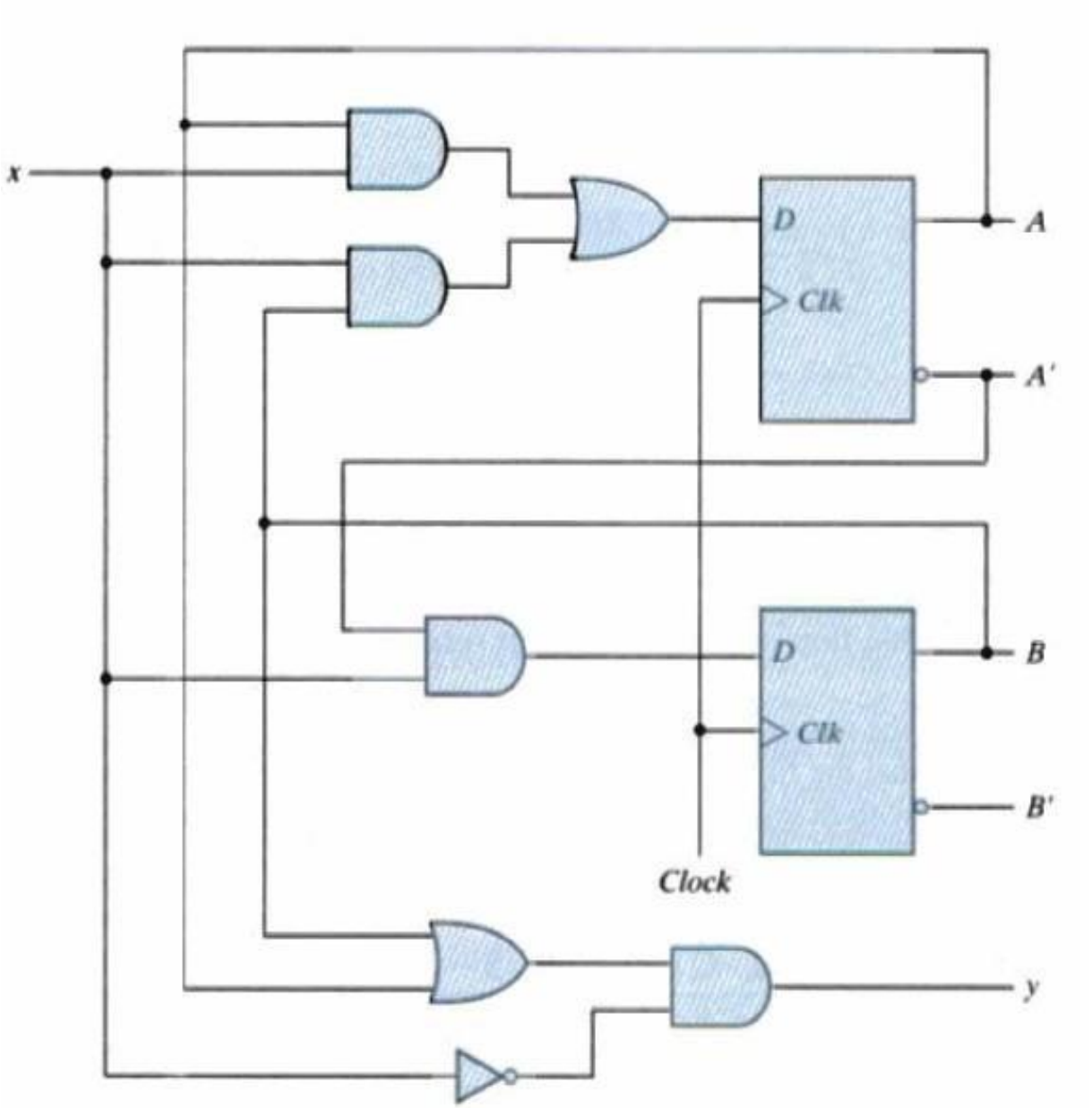
- The behaviour of a clocked sequential circuit can be described by means of state equations.

- A state equation (transition equation) specifies the next state as a function of the present state and inputs.

$$A(t+1) = A(t)x(t) + B(t)x(t)$$
$$B(t+1) = A'(t)x(t)$$
$$y(t) = [A(t) + B(t)]x'(t)$$

- For convenience, omit "t"

$$A^+ = A(t+1) = Ax + Bx$$
$$B^+ = B(t+1) = A'x$$
$$y = (A + B)x'$$



- State Table: The time sequence of inputs, outputs and flip flops can be enumerated in a state table. It consists of four sections such as present state, input, next state, and output.

$$A^+ = A(t+1) = Ax + Bx$$

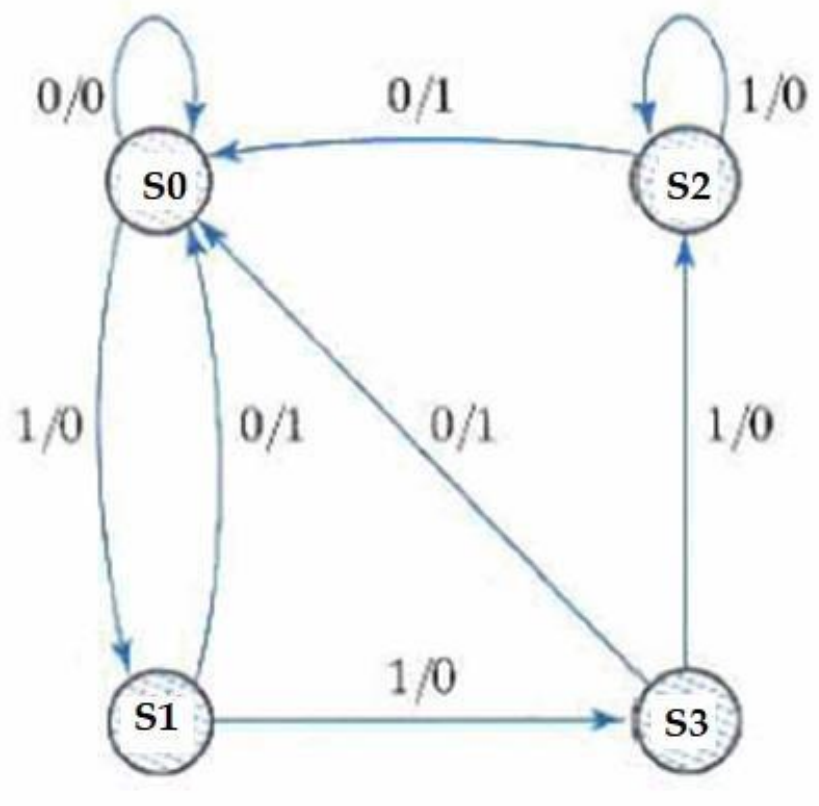
$$B^+ = B(t+1) = A'x$$

$$y = (A + B)x'$$

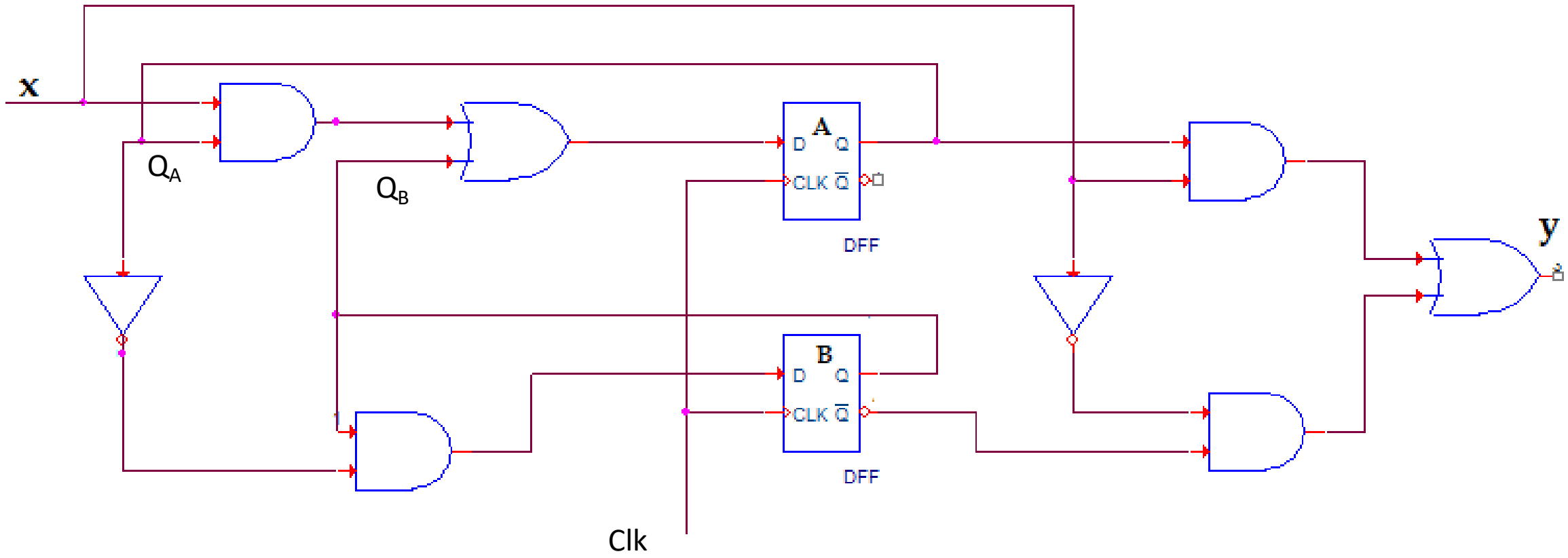
Present State		Input	Next State		Output
A	B	x	A^+	B^+	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

Present State		Next State				Output	
		x = 0		x = 1		x = 0	x = 1
A	B	A^+	B^+	A^+	B^+	y	y
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

- State Diagram: The information available in a state table can be represented graphically in the form of a state diagram.
- A state is represented by a circle and the transitions between states are indicated by directed lines connecting the circles.
- State assignments:
- S0 = 0 0
- S1 = 0 1
- S2 = 1 0
- S3 = 1 1



Analysis of Clocked Sequential Circuit with D Flip Flop



Step 1: Find the input and output equations:

$D_A = X Q_A + Q_B$ $D_B = Q'_A Q_B$

$Y = X Q_A + X' Q'_B$

Characteristic equation:

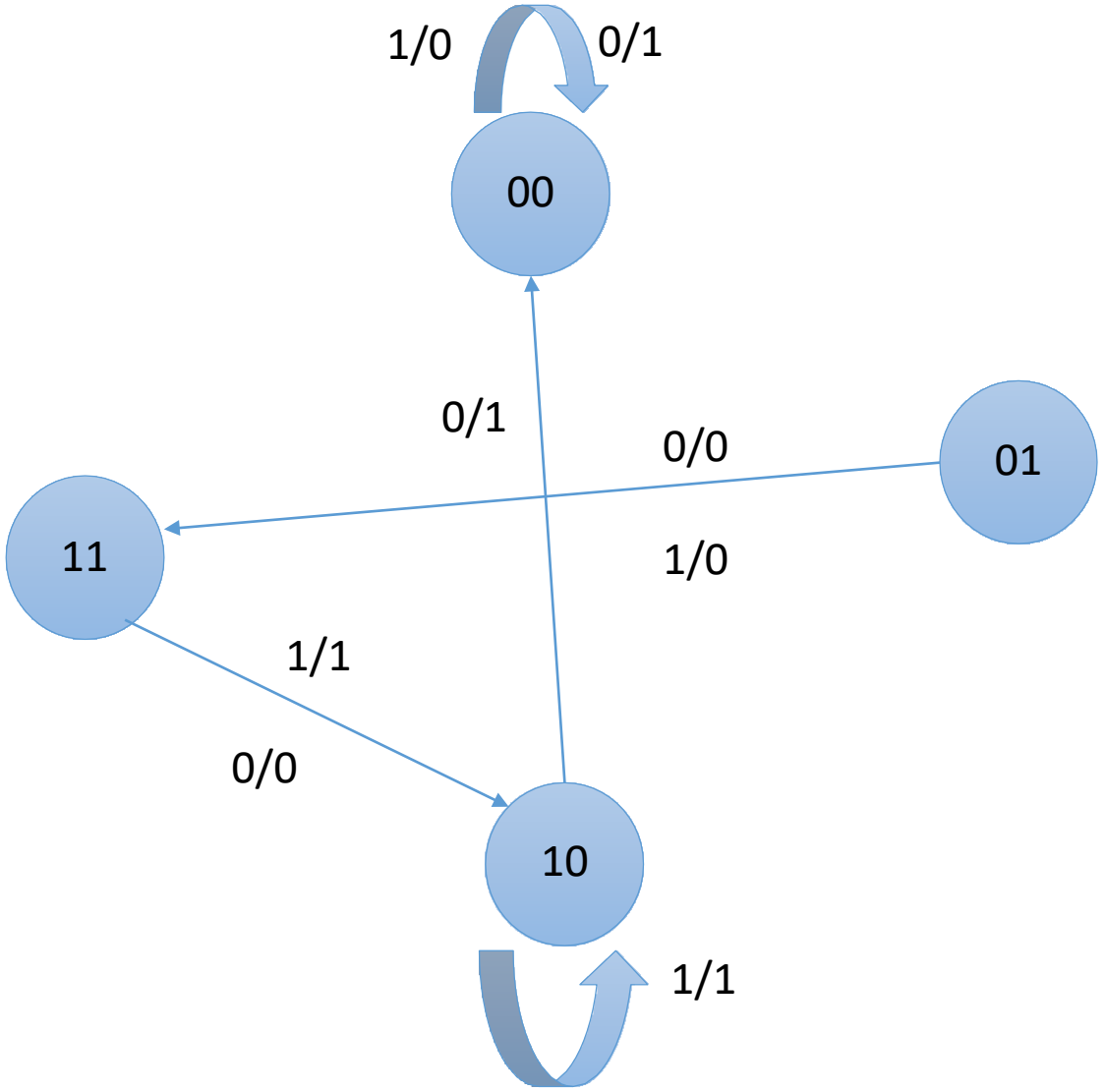
Q(t+1) = D

- State Table:

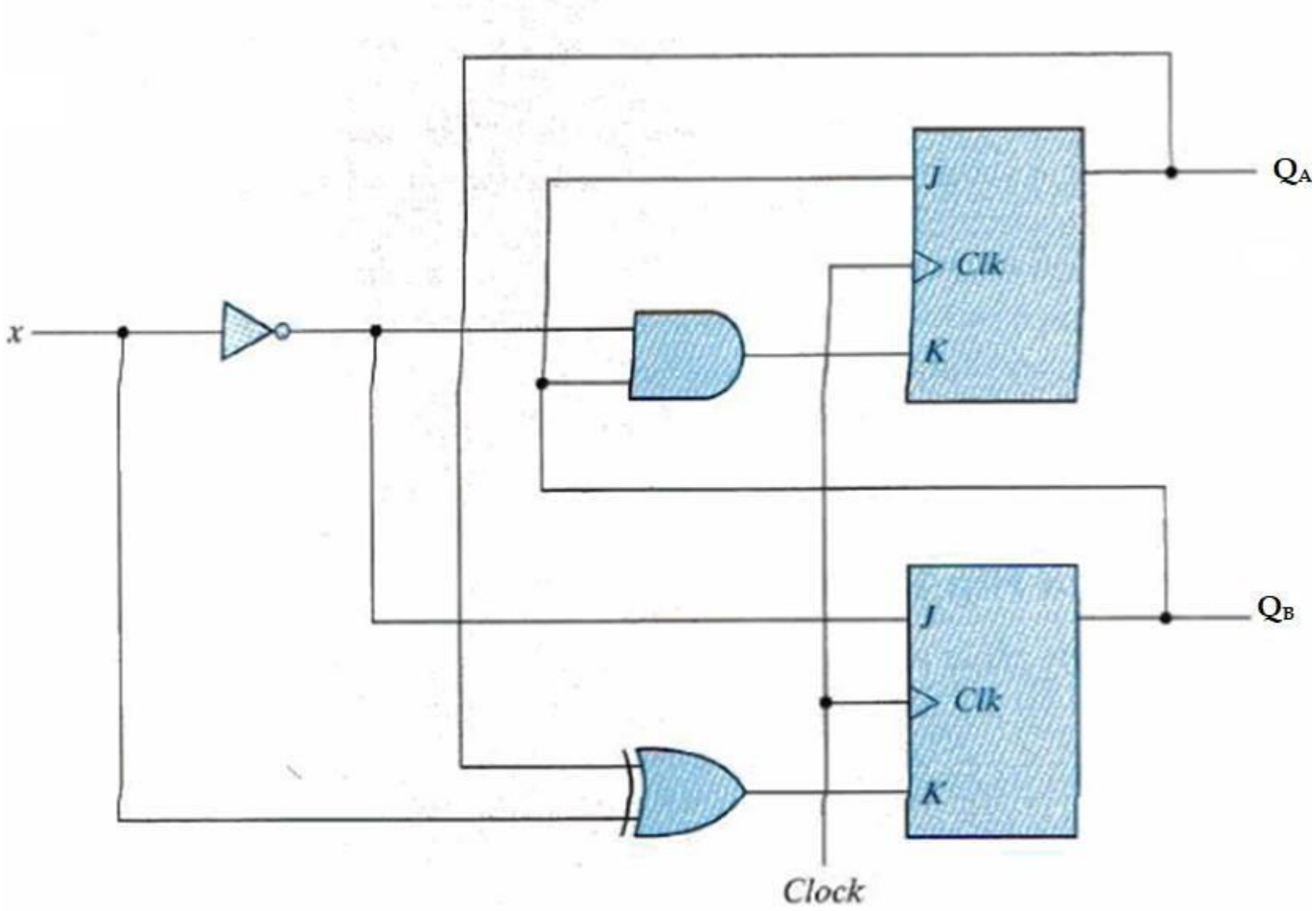
$Q_A+ = D_A = X\,Q_A + Q_B$ $\quad$ $Q_B+ = D_B = Q'_A\,Q_B$ $\quad$ $Y = X\,Q_A + X'\,Q'_B$

Present State		Input	Next State		Output
Q_A	Q_B	X	Q_A+	Q_B+	Y
0	0	0	0	0	1
0	0	1	0	0	0
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	1
1	1	0	1	0	0
1	1	1	1	0	1

- State diagram:
- 2 FF, hence 4 states
- $Q_A Q_B$
- S0 = 0 0
- S1 = 0 1
- S2 = 1 0
- S3 = 1 1



Analysis of Clocked Sequential Circuit with JK Flip Flop



- Step 1: Find the input and output equation.
- There is no output combinational logic, hence no output equation.
- $J_A = Q_B$
- $K_A = X' Q_B = X' J_A$
- $J_B = X'$
- $K_B = X \oplus Q_A$
- Characteristic table and equation:

JK Flip-Flop

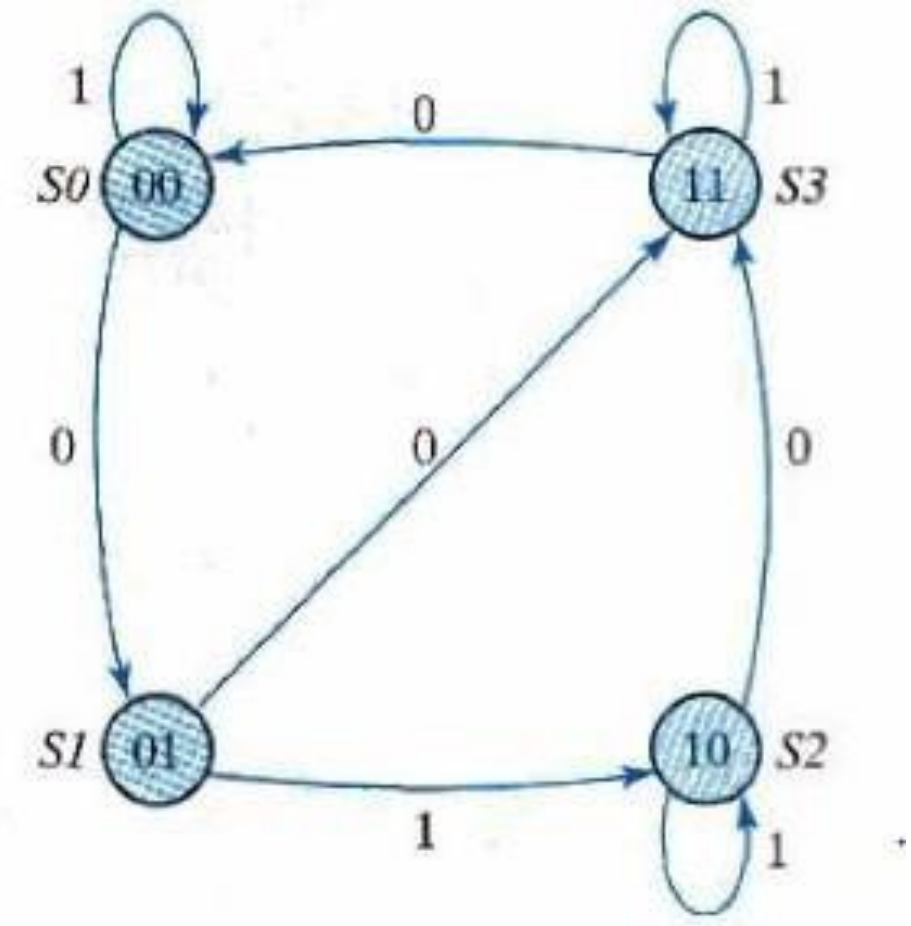
J	K	Q(t + 1)	
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	Complement

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

- Step 2: State Table

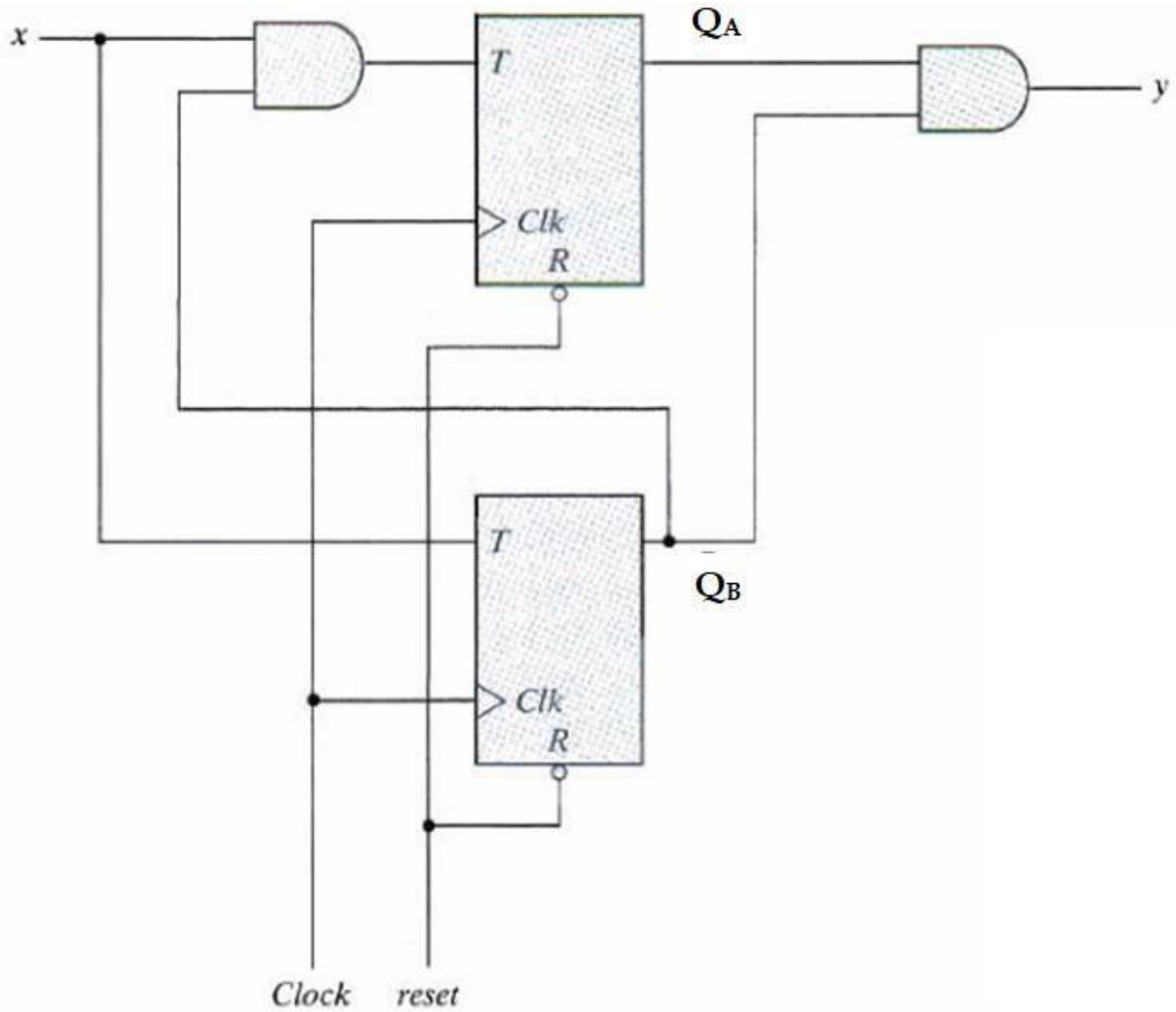
Present State		Input	Flip-Flop Inputs				Next State	
Q_A	Q_B	x	J_A	K_A	J_B	K_B	Q_A^+	Q_B^+
0	0	0	0	0	1	0	0	1
0	0	1	0	0	0	1	0	0
0	1	0	1	1	1	0	1	1
0	1	1	1	0	0	1	1	0
1	0	0	0	0	1	1	1	1
1	0	1	0	0	0	0	1	0
1	1	0	1	1	1	1	0	0
1	1	1	1	0	0	0	1	1

- Step 3: State Diagram.



Present State		Input	Next State	
Q_A	Q_B	x	Q_A^+	Q_B^+
0	0	0	0	1
0	0	1	0	0
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1

Analysis of Clocked Sequential Circuit with T Flip Flop



- Step 1: Find the input and output equation.
- $T_A = X\ Q_B$
- $T_B = X$
- $Y = Q_A\ Q_B$
- Characteristic table and equation:

T Flip-Flop

T	Q(t + 1)	
0	Q(t)	No change
1	Q′(t)	Complement

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

$Q_A^+ = Q_A \oplus T_A$

$Q_B^+ = Q_B \oplus T_B$

- Step 2: State Table.

Present State		Input			Next State		Output
Q_A	Q_B	X	T_A	T_B	Q_A^+	Q_B^+	Y
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	0	0	0	1	0
0	1	1	1	1	1	0	0
1	0	0	0	0	1	0	0
1	0	1	0	1	1	1	0
1	1	0	0	0	1	1	1
1	1	1	1	1	0	0	1

- Step 3: State Diagram.