Unit 6
Synchronous and Asynchronous Sequential Circuit

Er. Sachita Nand Mishra
M.E. In Computer and Electronics Engineering

What We Study:

- Flip Flop
- Triggering of Flip Flop
- Design with state equations and state reduction table
- Design procedure
- Introduction to Asynchronous circuit

Combinational Logic

- Combinational Logic:
  - Output depends only on current input
  - Has no memory

Sequential Logic

- Every digital system is likely to have combinational circuits, most systems encountered in practice also include storage elements, which require that the system be described in term of sequential logic.

Sequential Logic

- Sequential Logic:
  - Output depends not only on current input but also on past input values, e.g., design a counter
  - Need some type of memory to remember the past input values

Sequential Logic: Concept

- Sequential Logic circuits remember past inputs and past circuit state.
- Outputs from the system are “feedback” as new inputs
  - With gate delay and wire delay
- The storage elements are circuits that are capable of storing binary information: memory

Sequential circuits

- The logic circuit whose outputs at any instant of time depend not only on the present inputs but also on past outputs are called Sequential circuits.
- A sequential circuit 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.
- Sequential circuit is slower in operation than combinational circuit
- It may or may not contain clock input.

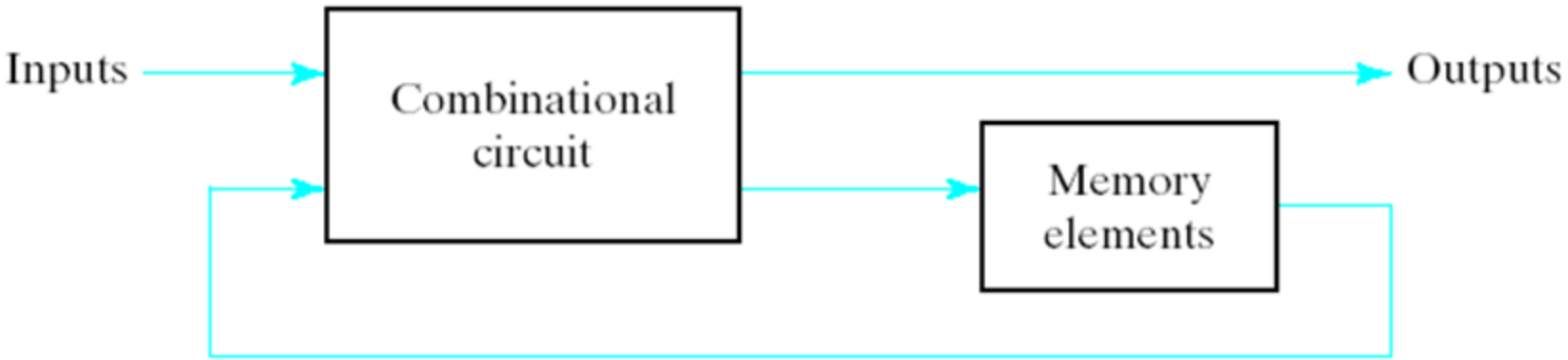


Fig. Block Diagram of Sequential Circuit

Synchronous *vs*. Asynchronous

There are two types of sequential circuits:

- **Synchronous** sequential circuit:
  - ✓ It is a system whose behavior can be defined from the knowledge of its signals at discrete instant of time.
  - ✓ Circuit output changes only at some discrete instants of time.
  - ✓ This type of circuits achieves synchronization by using a timing signal called the *clock*.
- **Asynchronous** sequential circuit:
  - ✓ It is a system whose behavior depends in the order in which its input signals change and can be affected at any instant of time.
  - ✓ circuit output can change at **any** time (clock less).

Difference between synchronous and asynchronous circuit

- The behavior of a synchronous sequential circuit can be defined from the knowledge of its signals at discrete instants of time.
- The behavior of an asynchronous sequential circuit depends upon the input signals at any instant of time and the order in which the inputs change signals at any instant of time and the order in which the inputs change.
- The storage elements commonly used in asynchronous sequential circuits are time-delay devices. Thus, an asynchronous sequential circuit may be regarded as a combinational circuit with feedback (no actual storage elements used).
- Asynchronous sequential circuit may become unstable at times, imposing many difficulties on the designer.

Difference between synchronous and asynchronous circuit

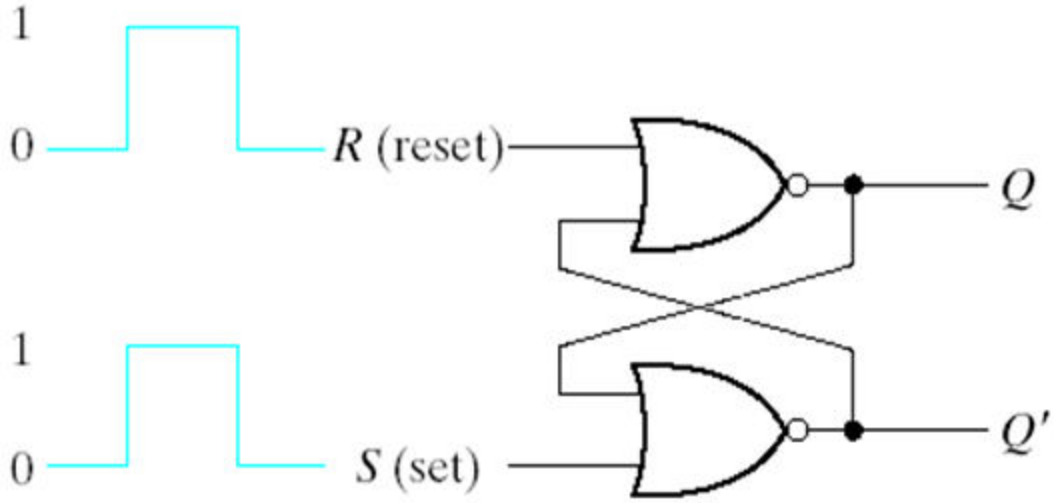
Synchronous Sequential Circuit	Asynchronous Sequential Circuit
• It is easy to design.	• It is difficult to design.
• A clocked flip flop acts as memory element.	• An unclocked flip flop or time delay is used as memory element.
• They are slower as clock is involved.	• They are comparatively faster as no clock is used here.
• The states of memory element is affected only at active edge of clock, if input is changed.	• The states of memory element will change any time as soon as input is changed.

Flip Flop

- The memory elements used in clocked sequential circuits are called *flip-flops*.
- These circuits are binary cells capable of storing one bit of information.
- A flip-flop circuit has two outputs, one for the normal value and one for the complement value of the bit stored in it.
- A flip-flop circuit can maintain a binary state indefinitely (as long as power is delivered to the circuit) until directed by an input signal to switch states.
- The major differences among various types of flip-flops are in the number of inputs they possess and in the manner in which the inputs affect the binary state.

Basic flip-flop circuit (*direct-coupled RS* flip-flop or *SR latch*)

- The SR latch is a circuit with two cross-coupled NOR gates or two cross-coupled NAND gates.
- Each flip-flop has **two outputs**, Q and Q', and **two inputs**, *set* and *reset*
- The cross-coupled connection from the output of one gate to the input of the other gate constitutes a feedback path.
- For this reason, the circuits are classified as asynchronous sequential circuits.



(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	

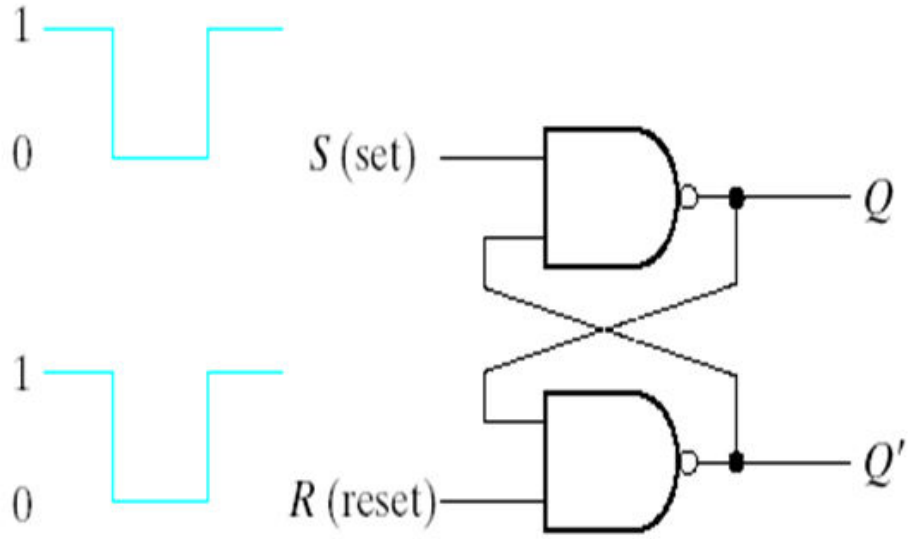
(b) Function table

Fig. *SR* Latch with NOR Gates

Basic flip-flop circuit (*direct-coupled RS* flip-flop or *SR latch*)

- A flip-flop has two useful states.
- **Set state**: When Q = 1 and Q' = 0, (or 1-state),
- **Reset state**: When Q = 0 and Q' = 1, (or 0-state)
- Output of a NOR gate is 0 if any input is 1, and that the output is 1 only when all inputs are 0.
- First, assume that the set input is 1 and the reset input is 0. Since gate-2 has an input of 1, its output Q' must be 0, which puts both inputs of gate-1 at 0, so that output Q is 1. When the set input is returned to 0, the outputs remain the same i.e. output Q' stay at 0, which leaves both inputs of gate-1 at 0, so that output Q is 1.
- Similarly, 1 in the reset input changes output Q to 0 and Q' to 1. When the reset input returns to 0, the outputs do not change.
- When a 1 is applied to both the set and the reset inputs, both Q and Q' outputs go to 0. This condition violates the fact that outputs Q and Q' are the complements of each other.

Basic flip-flop circuit: SR Latch with NAND Gates



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	

(a) Logic diagram (b) Function table

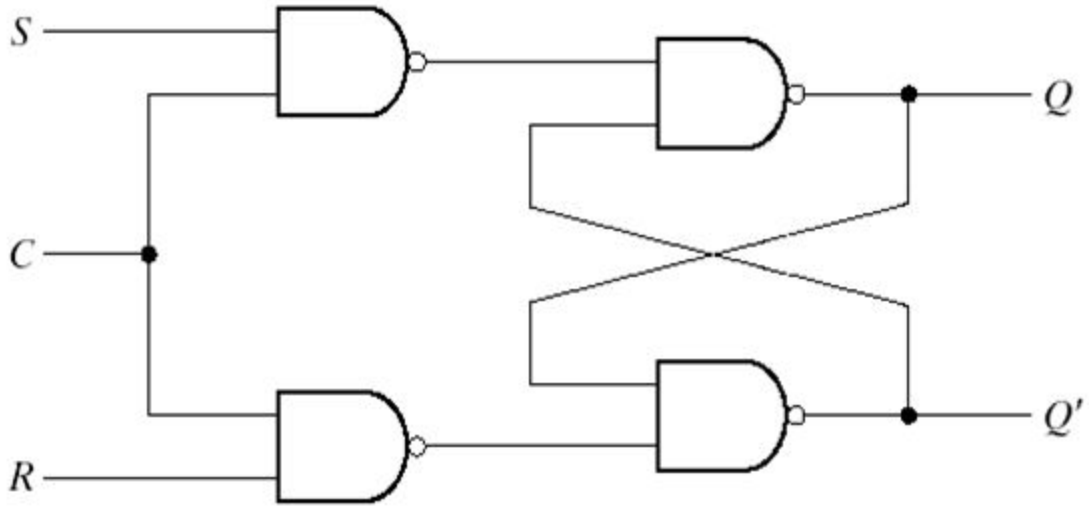
Fig. *SR* Latch with NAND Gates

Basic flip-flop circuit: SR Latch with NAND Gates

- The NAND basic flip-flop circuit operates with both inputs normally at 1 unless the state of the flip-flop has to be changed.
- The application of a momentary 0 to the set input causes output Q to go to 1 and Q' to go to 0, thus putting the flip-flop into the set state
- After the set input returns to 1, a momentary 0 to the reset input causes a transition to the clear state.
- When both inputs go to 0, both outputs go to 1- a condition avoided in normal flip-flop operation.

SR Latch with Control Input

- 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., it consists of the basic SR latch and two additional NAND gates.



(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	Indeterminate

(b) Function table

Fig. SR Latch with Control Input

Latch and Flip Flops

The primary difference between a flip-flop and latch is the EN/CLOCK input.

- ✓ The flip-flop's CLOCK input is edge sensitive, meaning the flip-flop's output changes on the edge (rising or falling) of the CLOCK input.
- ✓ The latch's EN input is level sensitive, meaning the latch's output changes on the level (high or low) of the EN input.

- Flip-flops is a sequential circuit that changes its output based on present input as well as clocking signal.

  Example: SR Flip-flop, JK Flip-flop, D Flip-Flop, T Flip-Flop

- Latch is a sequential circuit changes its output whenever there is change in the input and enable pin, independent of clock hence Latch is called as non-clocked Flip-Flop.

  Examples: SR Latch, JK Latch, D Latch, T Latch

Difference between Latch and Flip Flops

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.

Clocked RS Flip-Flop: NOR gate

- It consists of a basic flip-flop circuit and two additional AND gates along with clock pulse (CP) input.
- The pulse input acts as an enable signal for the other two inputs.

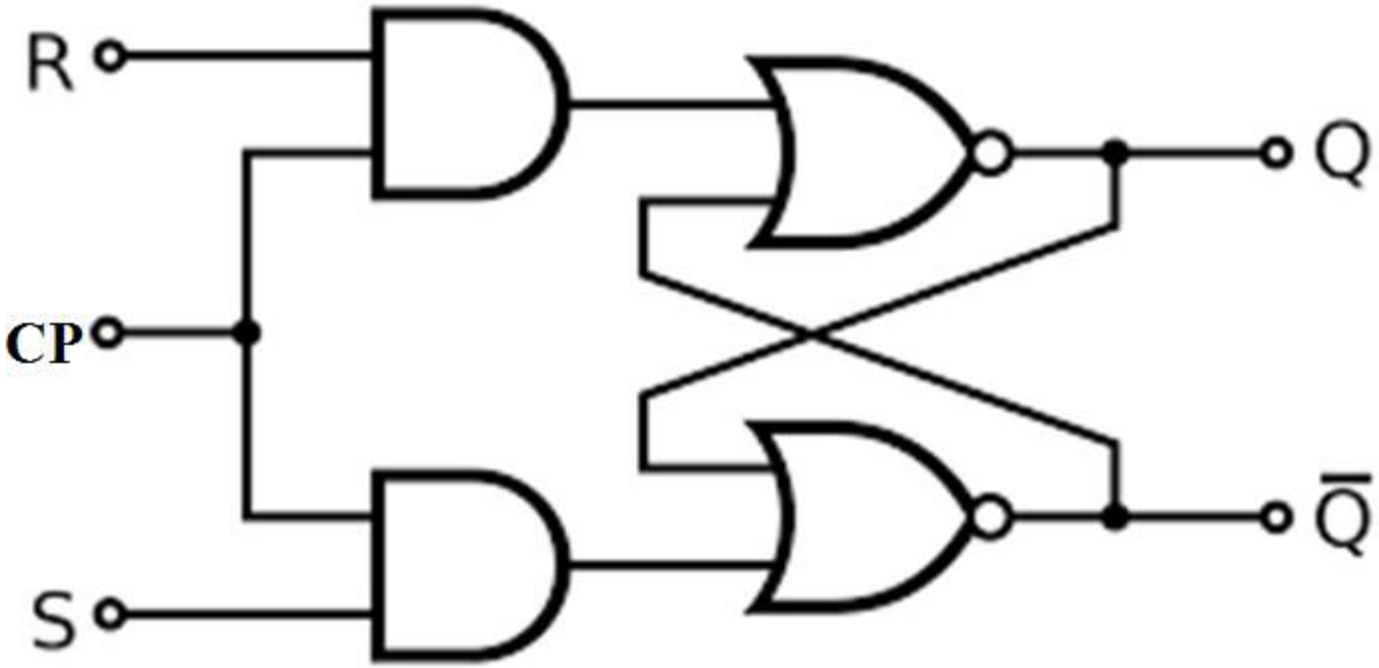


Fig: Logic Diagram

Clocked RS Flip-Flop: NAND gate

- It consists of a basic flip-flop circuit and two additional NAND gates along with clock pulse (CP) input.
- The pulse input acts as an enable signal for the other two inputs.

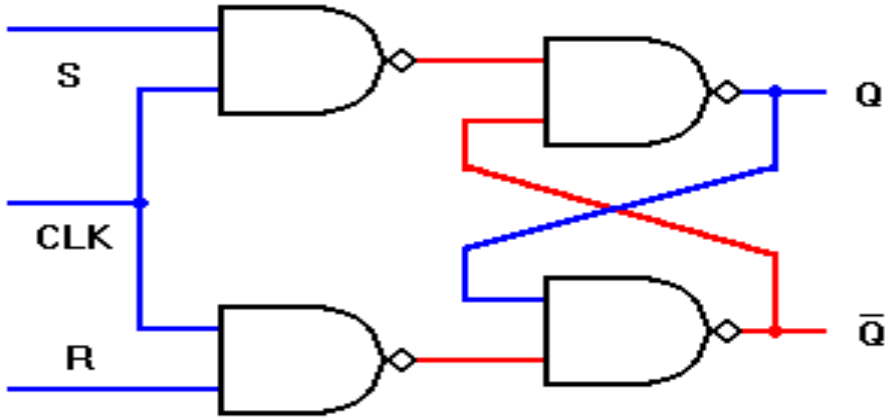


Figure : Clocked RS Flip Flop

Clocked RS Flip-Flop

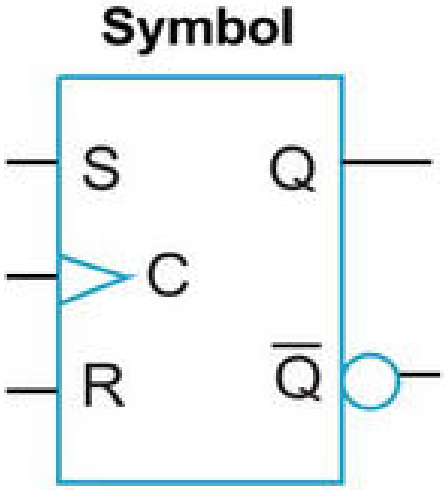
- When the pulse input goes to 1, information from the S or *R* input is allowed to reach the output.
- Set state: S = 1, *R* = 0, and *CP* = 1.
- Reset state: S = 0, *R* = 1, and *CP* = 1.
- In either case, when *CP* returns to 0, the circuit remains in its previous state.
- When *CP* = 1 and both the S and *R* inputs are equal to 0, the state of the circuit does not change.

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

Truth table

S	*R*	*CLK*	*Q(t+1)*	Comments
0	0	X	*Q(t)*	No change
0	1	↑	0	Reset
1	0	↑	1	Set
1	1	↑	?	Invalid

Clocked RS Flip-Flop

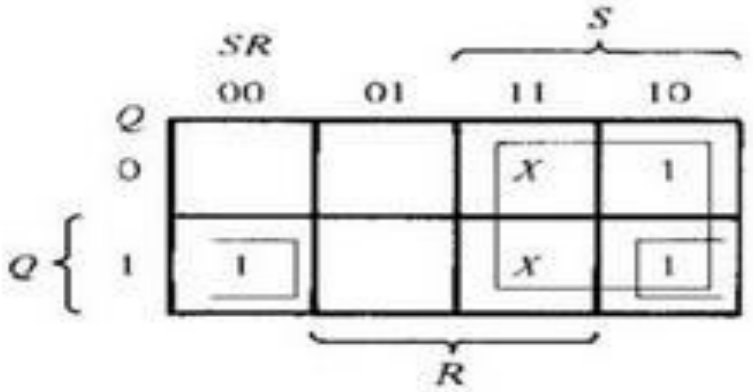


Graphic symbol

The graphic symbol of the RS flip-flop consists of a rectangular-shape block with inputs S, R, and C. The outputs are Q and Q', where Q' is the complement of Q (except in the indeterminate state).

Characteristic equation

The characteristic equation of the flip-flop specifies the value of the next state as a function of the present state and the inputs.

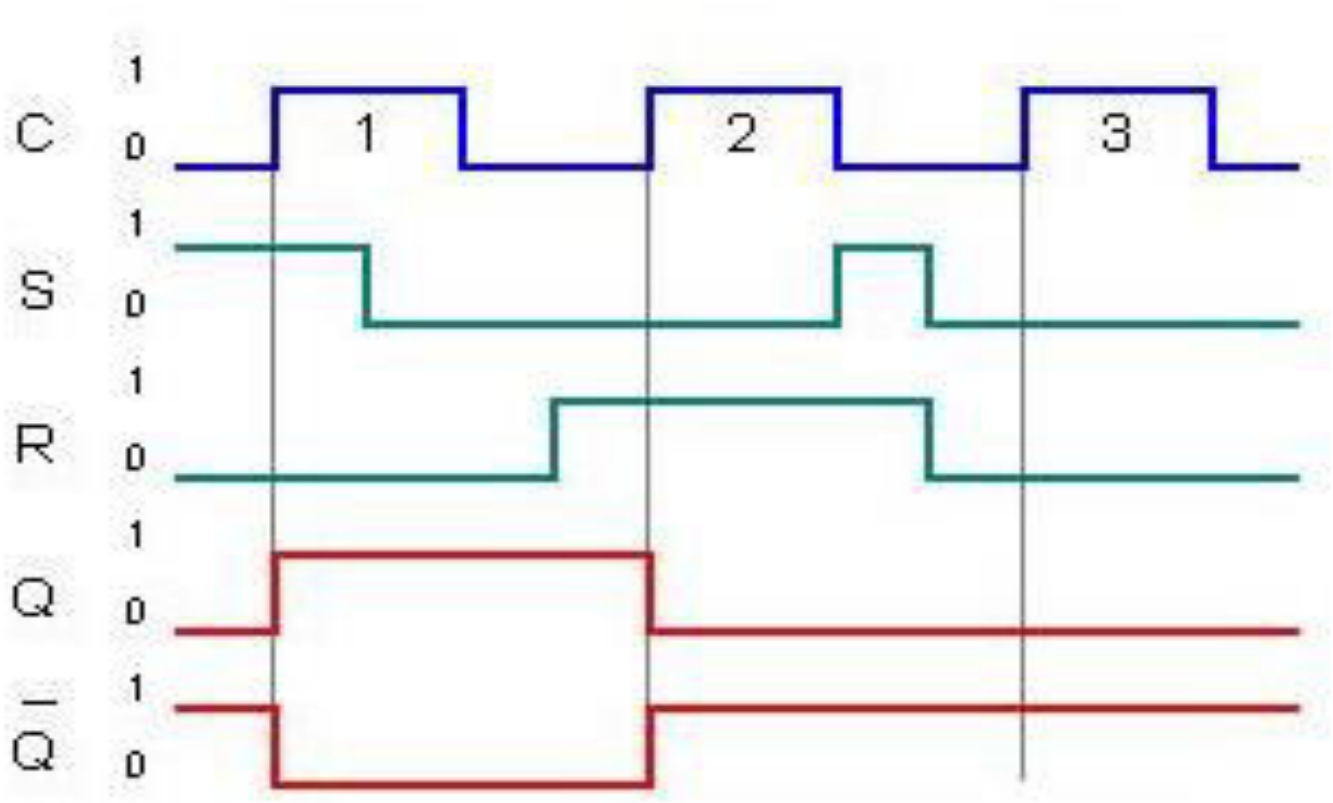


- **Characteristic Equation**

  $Q(t+1) = S + \overline{R}\,Q(t)$

  $S\,R = 0$ (S and R cannot be 1 simultaneously)

Clocked RS Flip-Flop: Timing diagram



Clocked JK Flip-Flop

- A *JK* flip-flop is a refinement of the *RS* flip-flop in that the indeterminate state of the *RS* type is defined in the *JK* type.
- Inputs *J* and *K* behave like inputs *S* and *R* to set and clear the flip-flop, respectively.
- The input marked *J* is for *set* and the input marked *K* is for *reset*.
- When both inputs *J* and *K* are equal to 1, the flip-flop switches to its complement state, that is, if $Q = 1$, it switches to $Q = 0$, and vice versa.
- A *JK* flip-flop constructed with two cross-coupled NOR gates and two AND gates is shown in Fig. below:

Clocked JK Flip-Flop

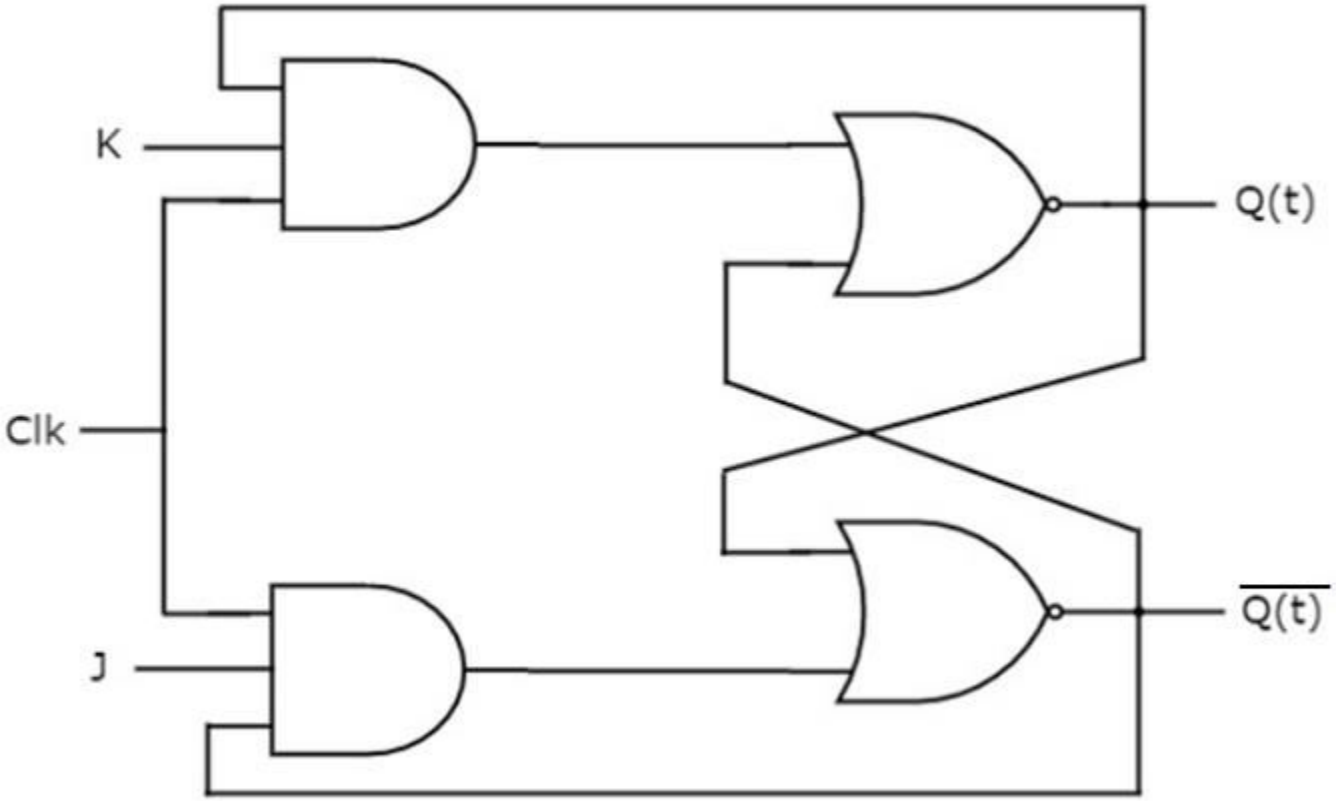


Fig: Logic diagram

Clocked JK Flip-Flop

- A *JK* flip-flop constructed with two cross-coupled NOR gates and two AND gates.
- Output *Q* is ANDed with *K* and *CP* inputs so that the flip-flop is cleared during a clock pulse only if *Q* was previously 1.
- Similarly, output *Q′* is ANDed with J and *CP* inputs so that the flop-flop is set with a clock pulse only when *Q′* was previously 1.
- Because of the feedback connection in the *JK* flipflop, a *CP* pulse that remains in the 1 state while both *J* and *K* are equal to 1 will cause the output to complement again and repeat complementing until the pulse goes back to 0.

Clocked JK Flip-Flop

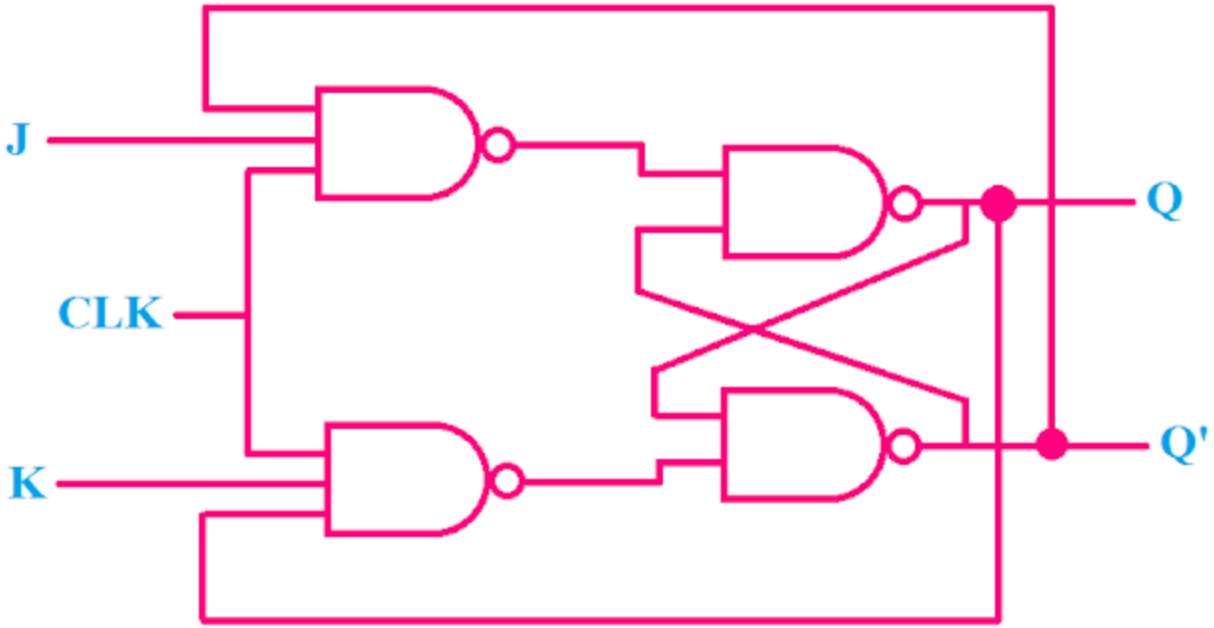


Fig: Logic diagram

Clocked JK Flip-Flop

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

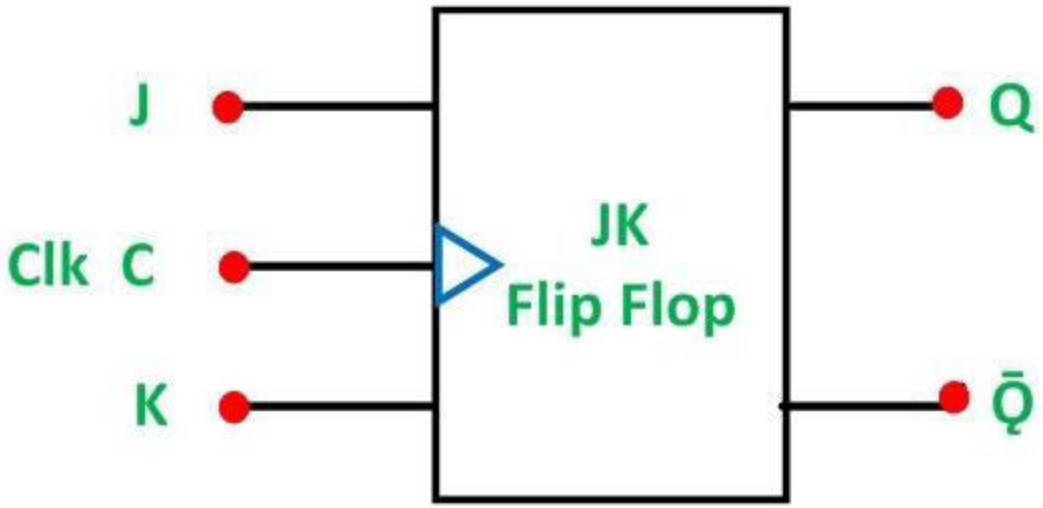


Fig: Graphic Symbol

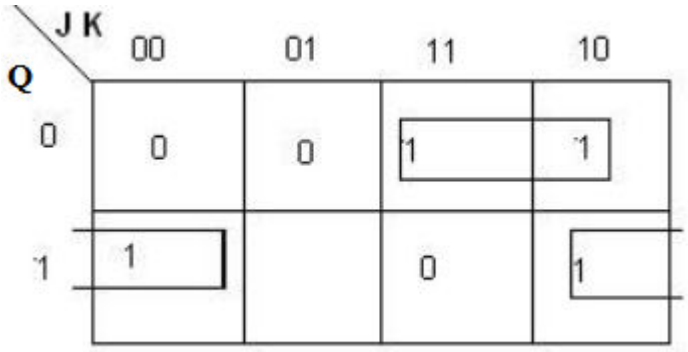
Flip-Flop Characteristic Tables

***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

Clocked JK Flip-Flop

Characteristic equation

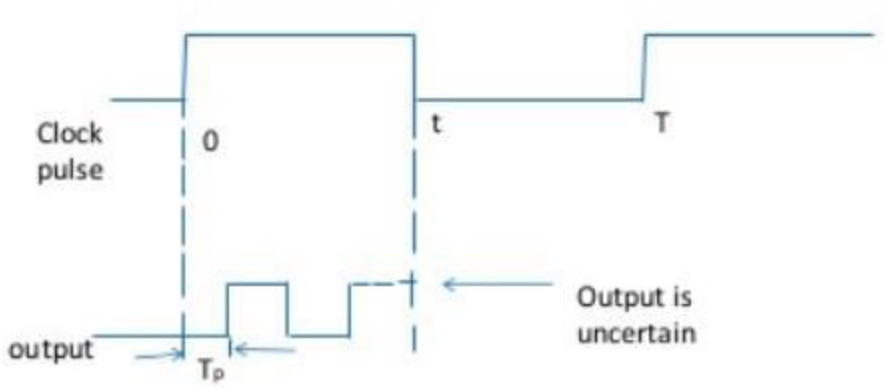


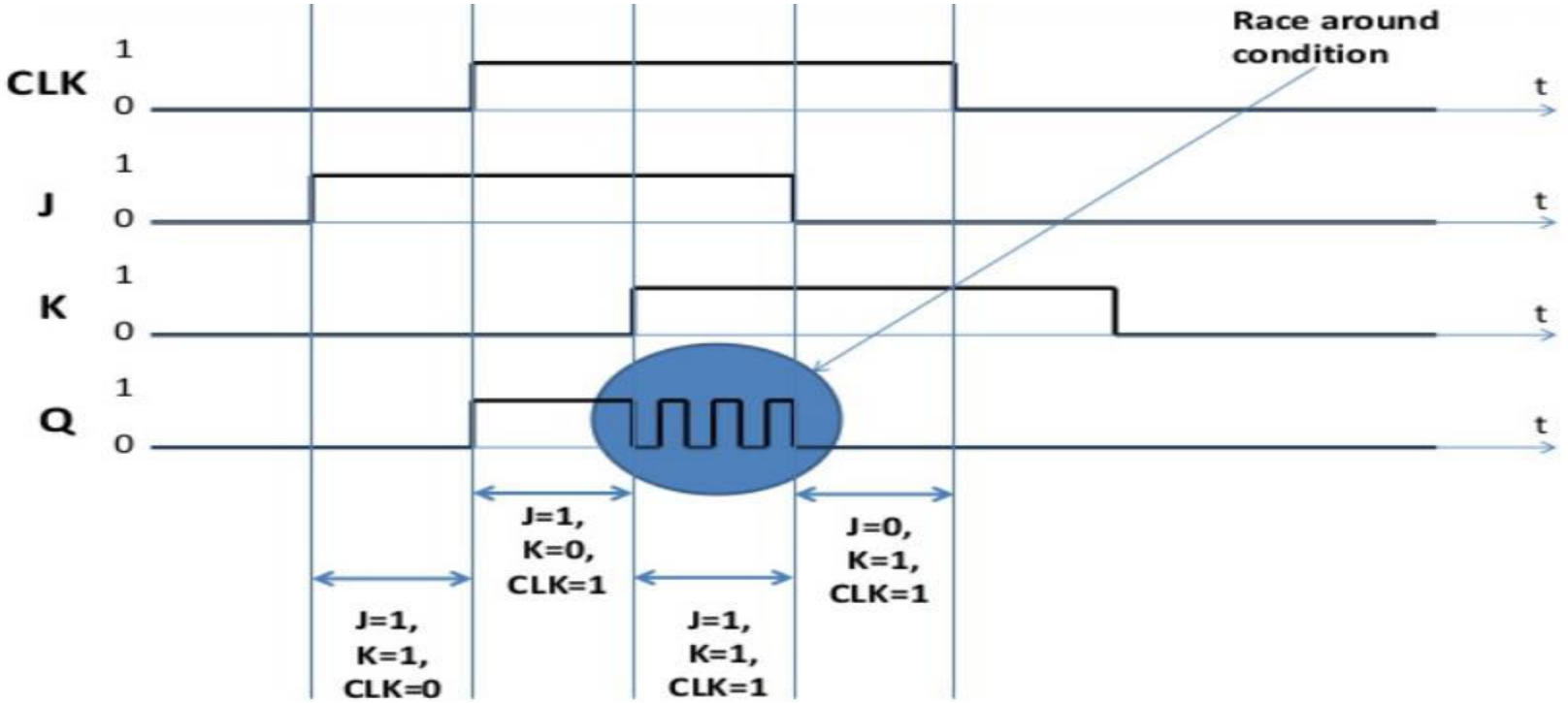
$$Q(t+1)=JQ'+K'Q$$

Clocked JK Flip-Flop: Race around condition

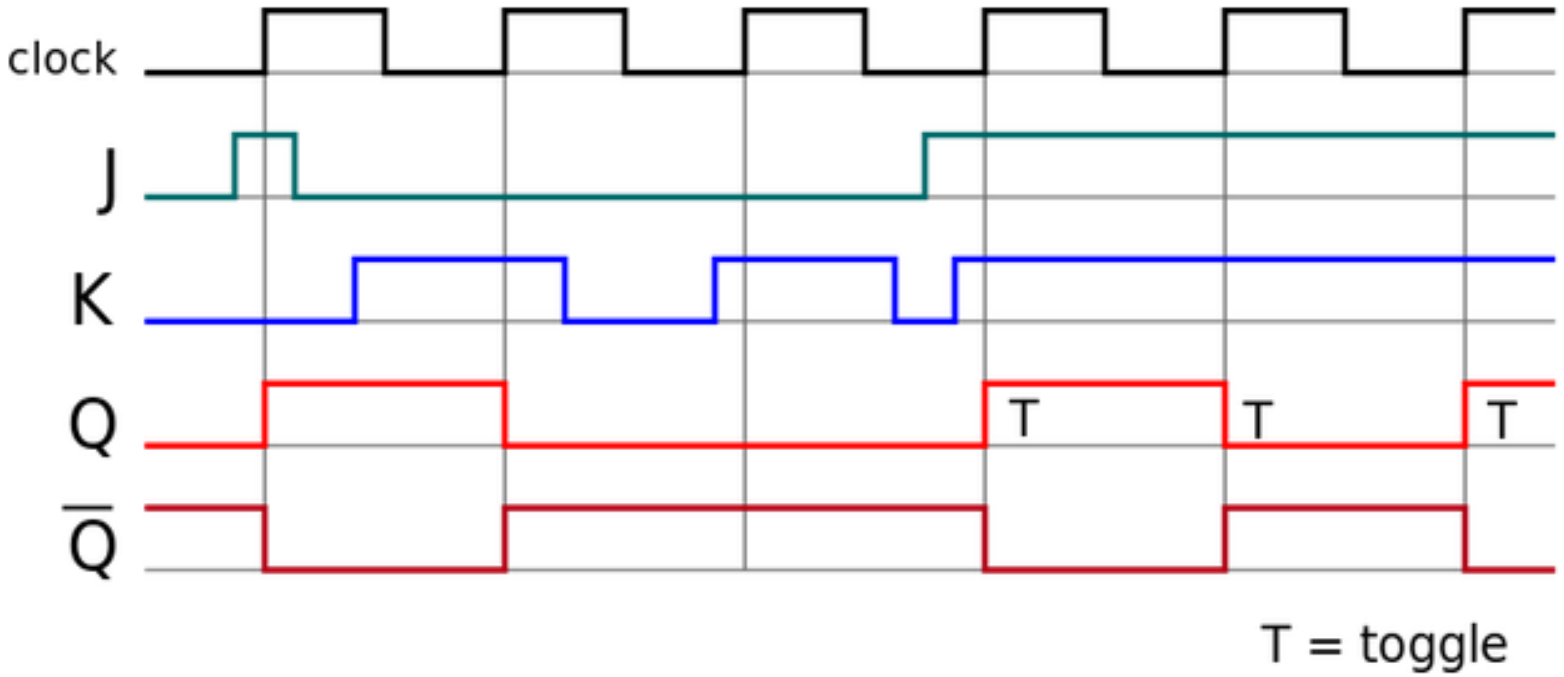
- If the inputs of JK flip Flop are J=K=1 and Q=0 and clock pulse as shown in fig., After a time interval tp equal to propagation delay of NAND gates, the output will change to Q=1.Now we have J=1,K=1 and Q=1.If duration of clock pulse(T) is greater than propagation delay t_p, after another time interval of t_p the output will change back to Q=0,hence the output will oscillate back and forth between 0 and 1.The output is uncertain at the end of clock pulse if flip flop is level trigger. This situation is called race around condition.
- The race around condition can be avoided if clock pulse is reduced than the propagation delay of the flip flop $t_p > T$ but this is not practically feasible.
- A more practical method for this is use of master slave flip flop or edge triggered JK flip flop.

Clocked JK Flip-Flop: Race around condition





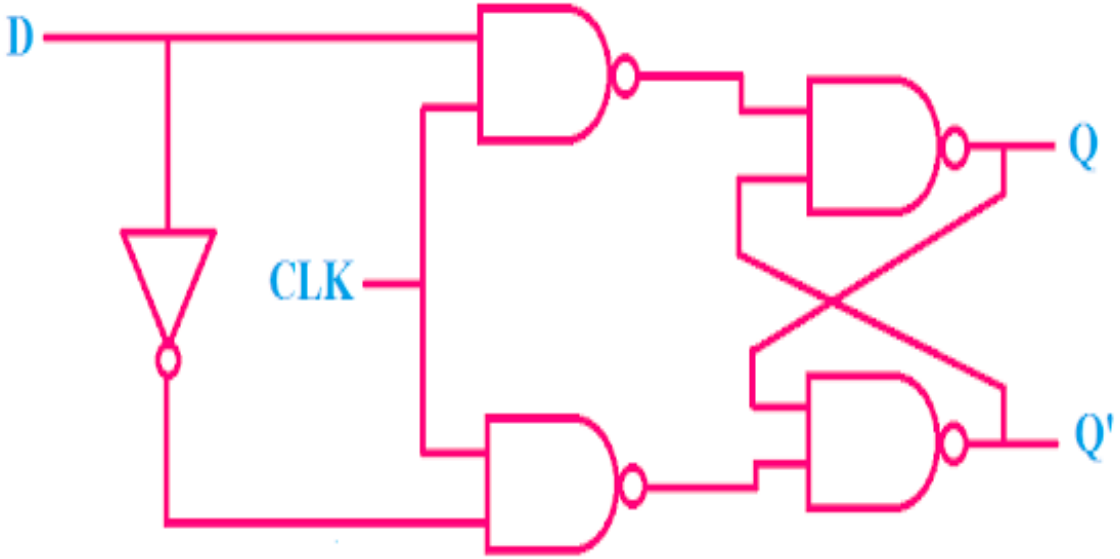
Clocked JK Flip-Flop: Timing diagram



D Flip-Flop

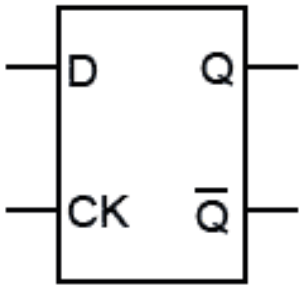
- One way to eliminate the undesirable condition of the indeterminate state in the *RS* flip-flop is to ensure that inputs S and *R* are never equal to 1 at the same time.
  - This is done in the *D* flip-flop
- The *D* flip-flop has only two inputs: *D* and *CP*.
- The *D* input goes directly to the S input and its complement is applied to the *R* input.
- As long as CP is 0, the circuit cannot change state regardless of the value of *D*.
- The *D* input is sampled when *CP* = 1.
  - If *D* is 1, the *Q* output goes to 1, placing the circuit in the **set state**.
  - If *D* is 0, output *Q* goes to 0 and the circuit switches to the **clear state**.

D Flip-Flop



Logic diagram

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

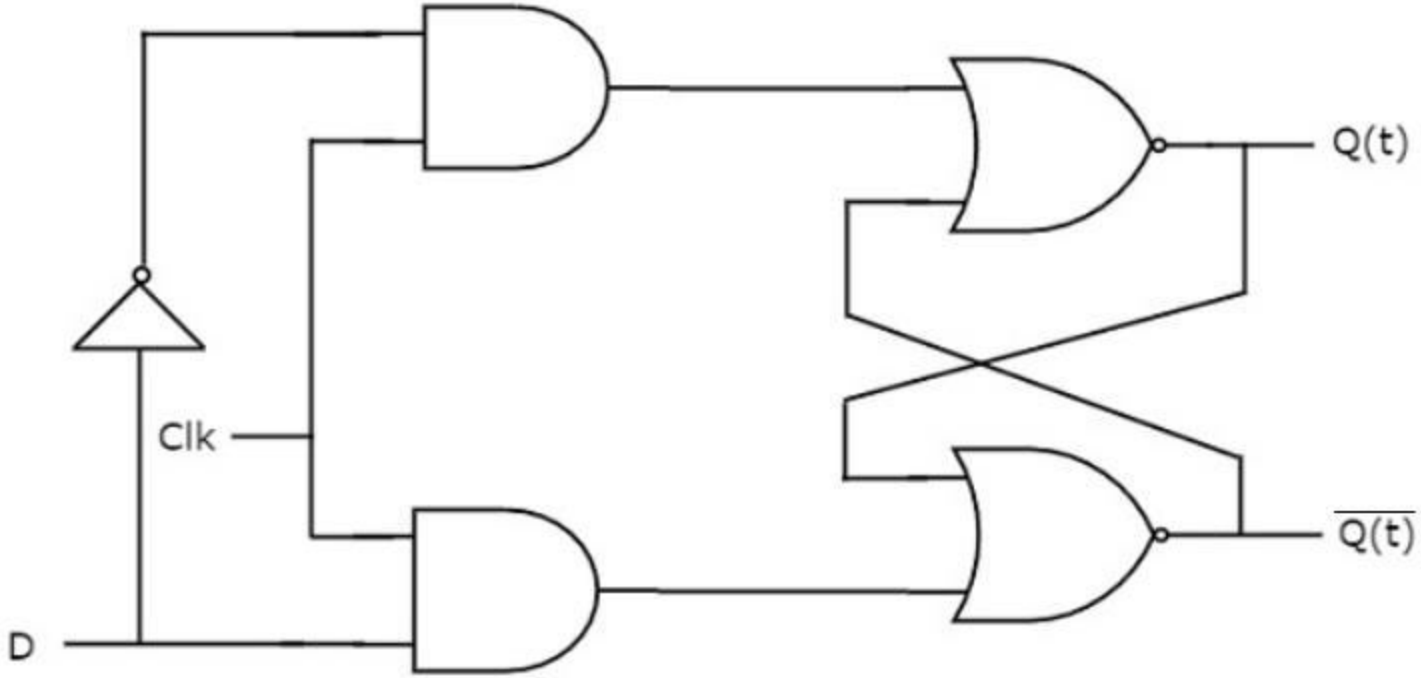


Graphic symbol

Inputs		Outputs	
CK	D	Q	$\overline{Q}$
0	X	No change	
1	0	0	1
1	1	1	0

Characteristic table

D Flip-Flop



Logic diagram

D Flip-Flop: timing diagram and characteristic equation

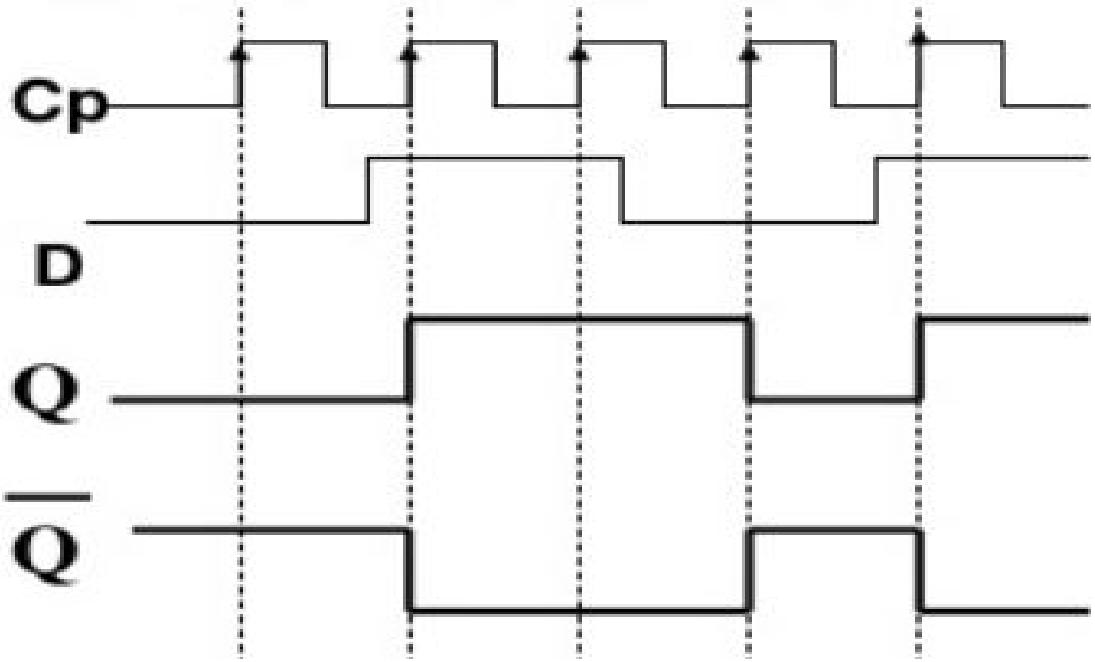
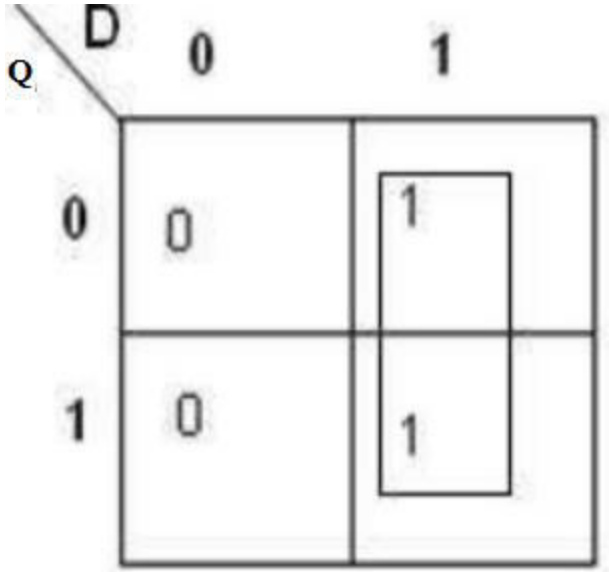


Fig: Timing diagram



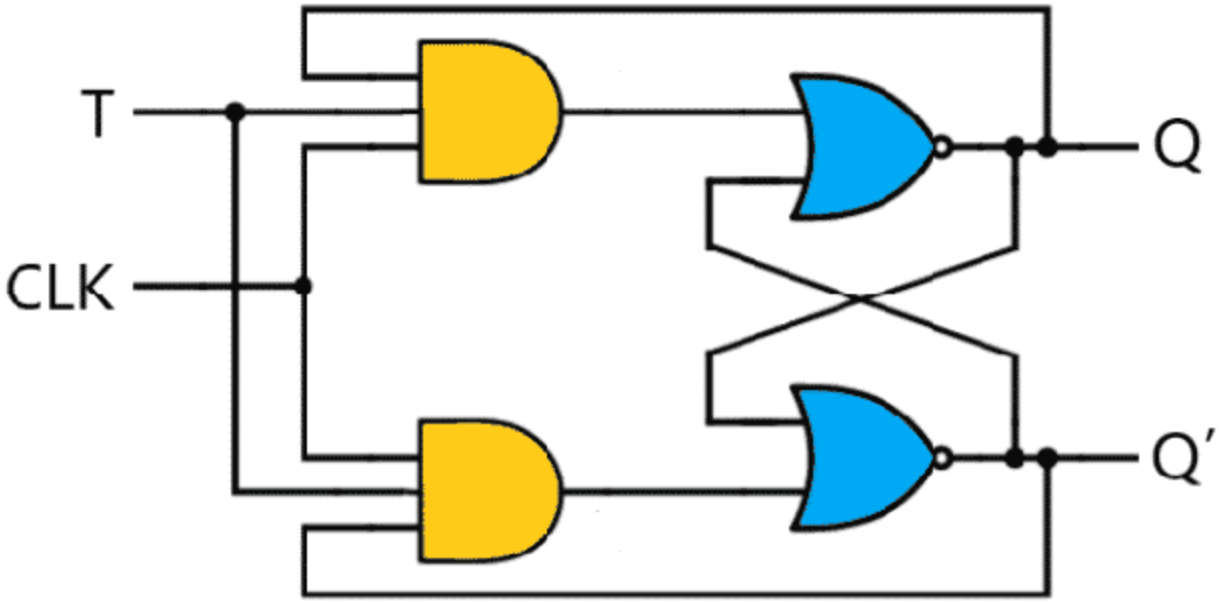
Characteristic Equation

Q(t+1) = D

T Flip-Flop

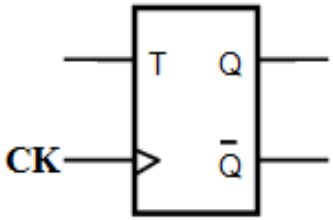
- The *T* flip-flop is a single-input version of the *JK* flip-flop and is obtained from the *JK* flip-flop when both inputs are tied together.
- The designation *T* comes from the ability of the flip-flop to "toggle," or complement, its state.
- Regardless of the present state, the flip-flop complements its output when the clock pulse occurs while input *T* is 1.
- The characteristic table and characteristic equation show that:
  - When $T = 0$, $Q(t + 1) = Q$, that is, the next state is the same as the present state and no change occurs.
  - When $T = 1$, then $Q\ (t + 1) = Q'$, and the state of the flip-flop is complemented.

T Flip-Flop



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

Fig: Logic diagram



Graphical symbol

***T* Flip-Flop**

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

Table: Function table

T Flip-Flop: Characteristic equation and timing diagram

Q \ T	0	1
0	0	1
1	1	0

Characteristic equation

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

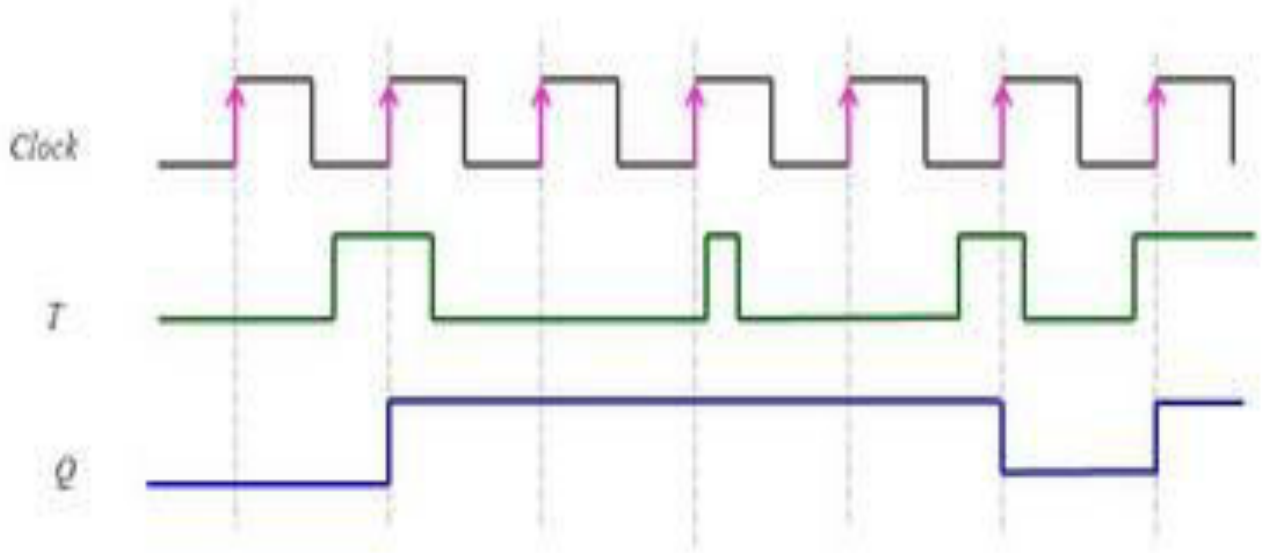


Fig: Timing diagram

Master-Slave Flip-Flop

- A master-slave flip-flop is constructed from two separate flip-flops.
  - One circuit serves as a master and the other as a slave, and the overall circuit is referred to as a master slave flip-flop.

RS Master-Slave Flip-flop

- It consists of a master flip-flop, a slave flip-flop, and an inverter.
- When clock pulse CP is 0, the output of the inverter is 1.
- Since the clock input of the slave is 1, the flip-flop is enabled and output Q is equal to Y, while Q' is equal to Y'.
- The master flip-flop is disabled because CP = 0.
- When the pulse becomes 1, the information then at the external R and S inputs is transmitted to the master flip-flop. The slave flip-flop, however, is isolated as long as the pulse is at its 1 level, because the output of the inverter is 0.
- When the pulse returns to 0, the master flip-flop is isolated; this prevents the external inputs from affecting it. The slave flip-flop then goes to the same state as the master flip-flop.

SR Master-Slave FF Operation

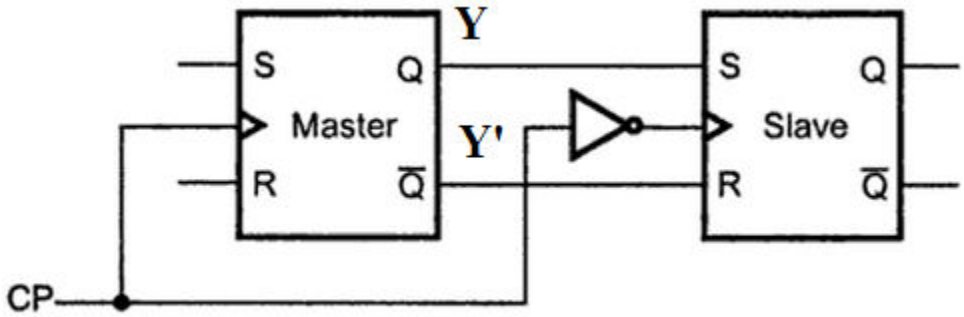


Fig. Master-slave SR flip-flop

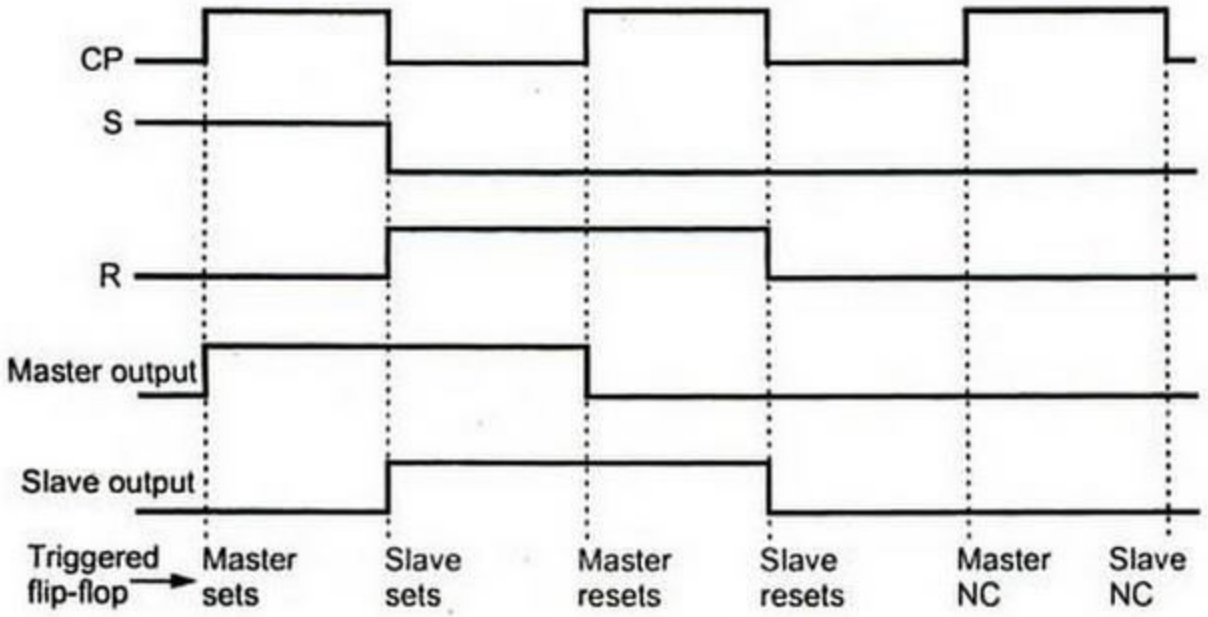
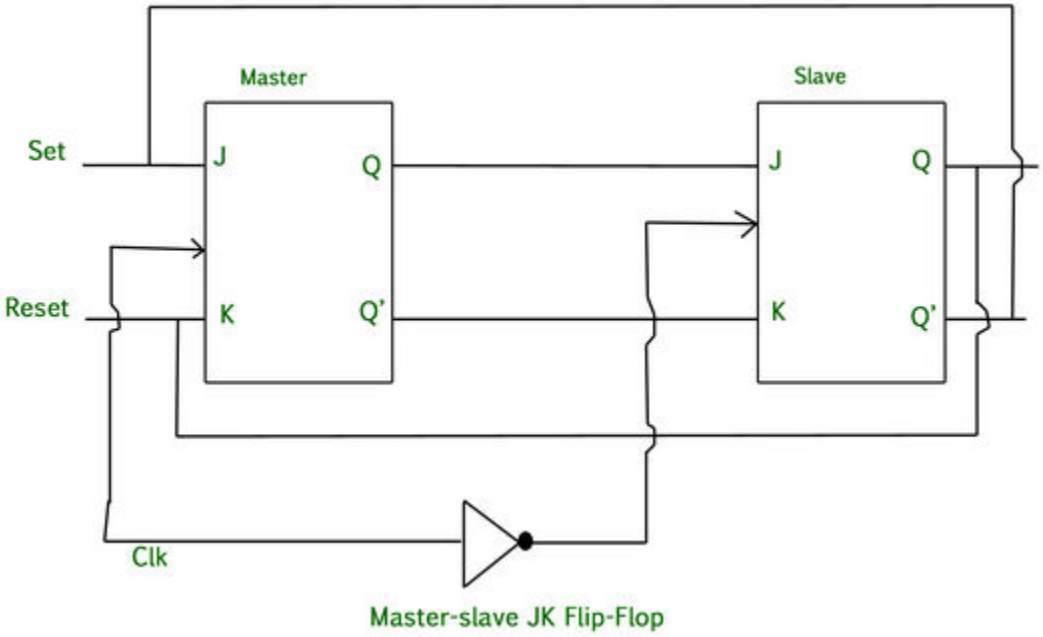


Fig. Input and output waveforms for master-slave flip-flop

JK Master-slave Flip-Flop

- Master-slave JK flip-flop constructed with NAND gates is shown in Fig.
- It consists of two flip-flops; the master flip-flop, and the slave flip-flop.
- The information present at the J and K inputs is transmitted to the master flip-flop on the positive edge of a clock pulse and is held there until the negative edge of the clock pulse occurs, after which it is allowed to pass through to the slave flip-flop.

JK Master-slave Flip-Flop



JK Master-slave Flip-Flop

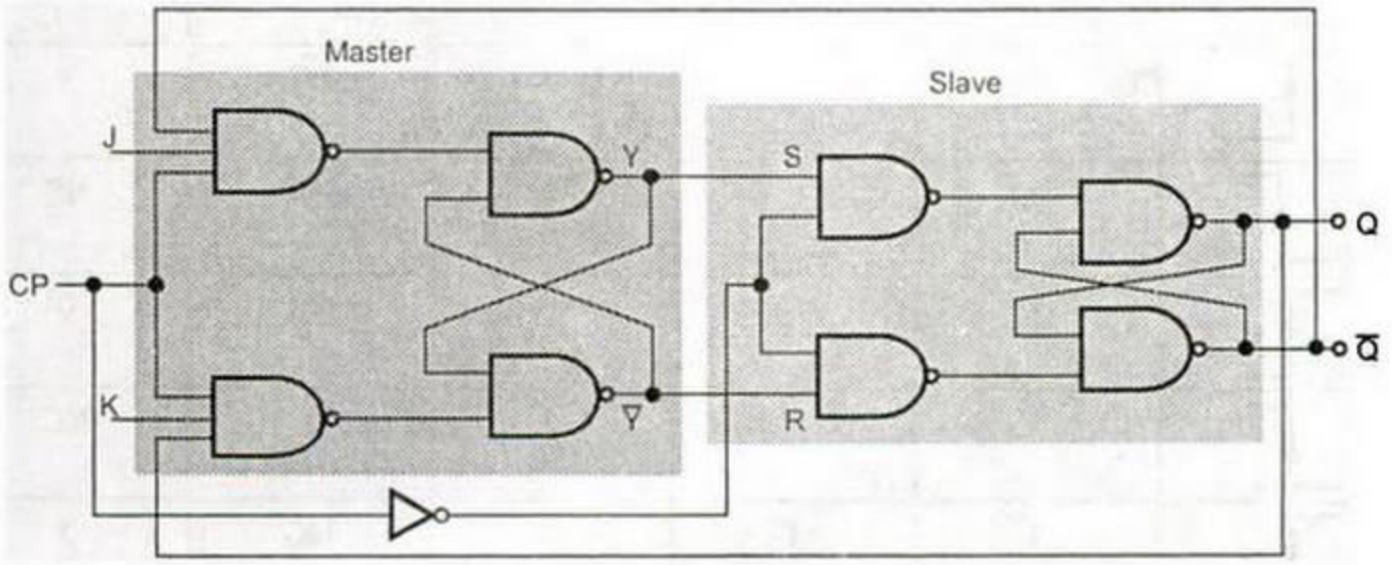


Fig. Master-slave JK flip-flo

- The following is truth table of master slave flip flop.

Case	Inputs			Outputs		Remark
	CLK	J	K	Q_{n+1}	$\bar{Q}_{n+1}$	
I	×	0	0	Q_n	$\bar{Q}_n$	No change
II	⎍(1)	0	0	Q_n	$\bar{Q}_n$	No change
III	⎍(1)	0	1	0	1	Reset
IV	⎍(1)	1	0	1	0	Set
V	⎍(1)	1	1	$\bar{Q}_n$	Q_n	Toggle

Fig . Truth table of Master slave JK FF

JK Master-slave Flip-Flop

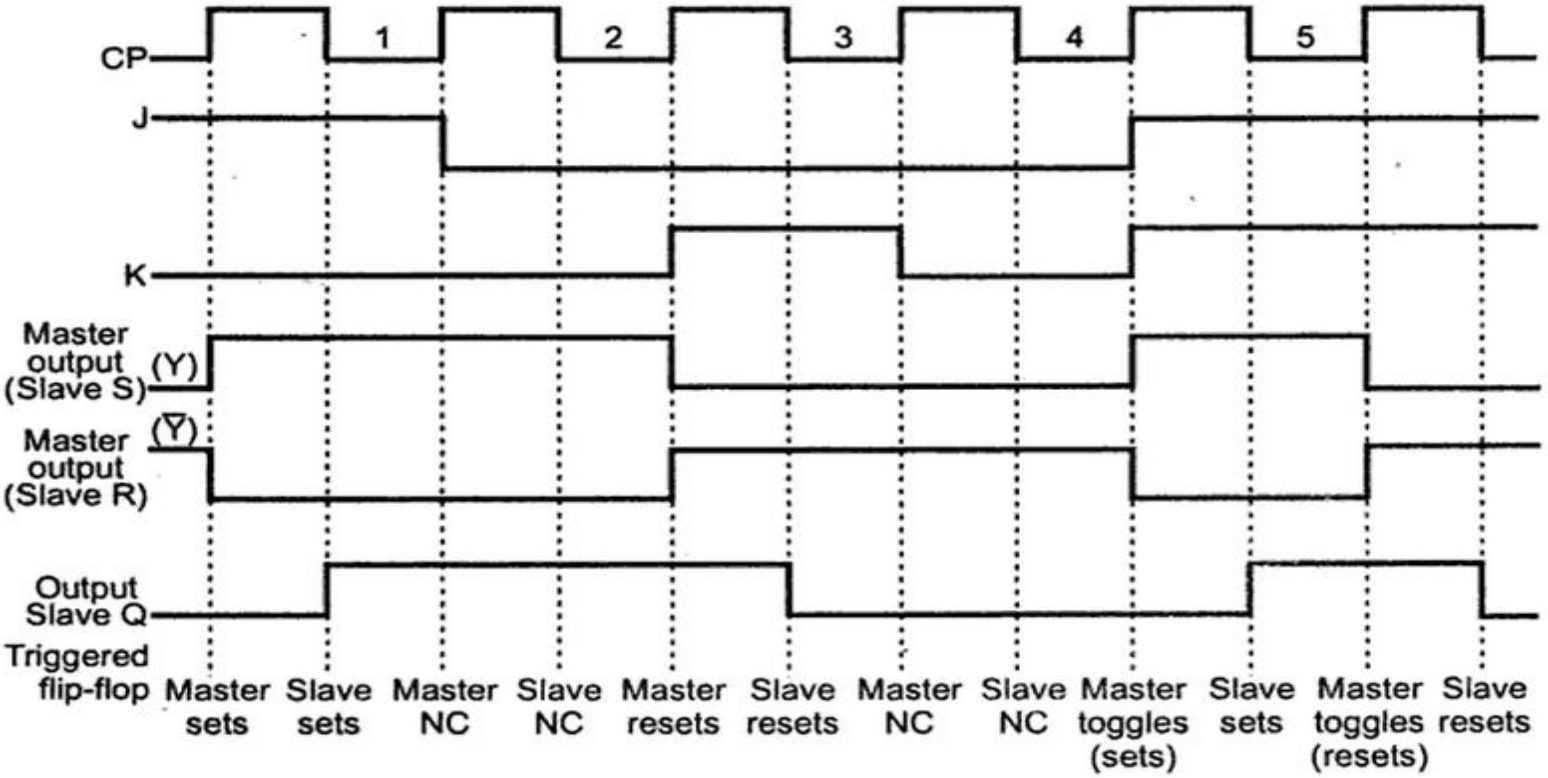


Fig. Input and output waveforms of master-slave JK flip-flop

JK Master-slave Flip-Flop

- Operation:
- The clock input is normally 0, which prevents the J and K inputs from affecting the master flip-flop.
- The slave flip-flop is a clocked RS type, with the master flip-flop supplying the inputs and the clock input being inverted by NOT gate
- When the clock is 0, Q = Y, and Q' = Y'.
- When the positive edge of a clock pulse occurs, the master flip-flop is affected and may switch states.
- The slave flip-flop is isolated as long as the clock is at the 1 level
- When the clock input returns to 0, the master flip-flop is isolated from the J and K inputs and the slave flip-flop goes to the same state as the master flip-flop.

JK Master-slave Flip-Flop

When J =1 and K = 0, the master sets on the positive clock. The high Y output of the master drives the S input of the slave, so at negative clock, slave sets, copying the action of the master.

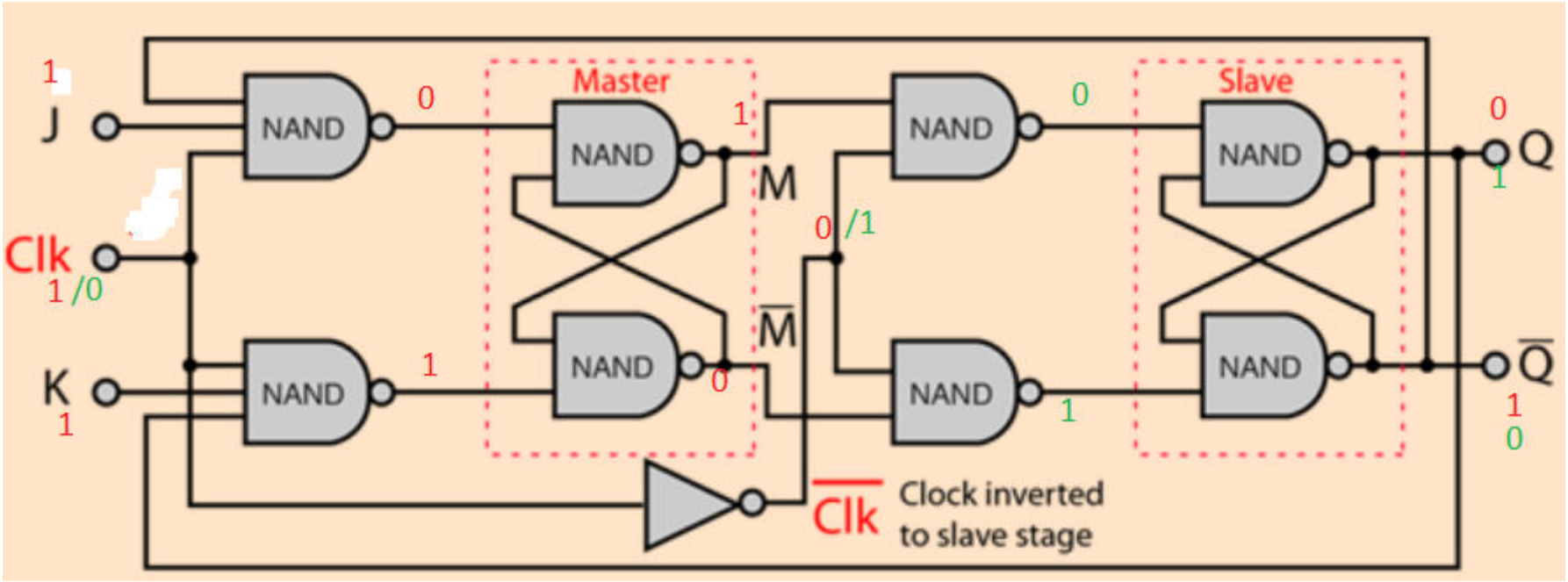
When J = 0 and K = 1, the master resets on the positive clock. The high $\overline{Y}$ output of the master goes to the R input of the slave. Therefore, at the negative clock slave resets, again copying the action of the master.

When J = 1 and K = 1, master toggles on the positive clock and slave then copies the output of master on the negative clock. At this instant, feedback inputs to the master flip-flop are complemented but as it is negative half of the clock pulse master flip-flop is inactive. This prevents race around condition. Fig. shows input and output waveforms of master-slave JK flip-flop.

Master Slave flip flop: Avoidance of Race Around Condition of JK Flip Flop

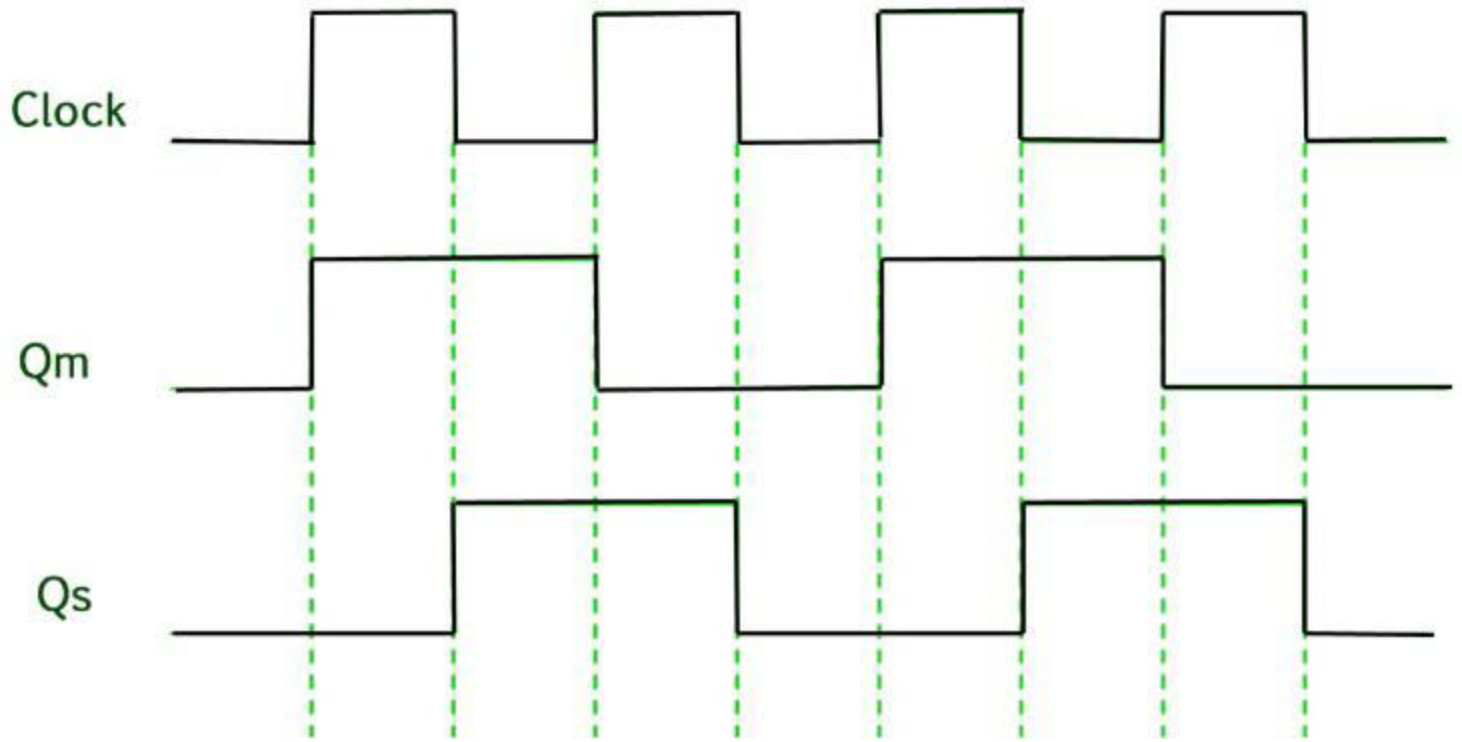
- **Race Around Condition In JK Flip-flop** – For J-K flip-flop, if J=K=1, and if clk=1 for a long period of time, then Q output will toggle as long as CLK is high, which makes the output of the flip-flop unstable or uncertain. This problem is called race around condition in J-K flip-flop.
- This problem (Race Around Condition) can be avoided by ensuring that the clock input is at logic "1" only for a very short time. This introduced the concept of **Master Slave JK** flip flop.
- If J=1 and K=1, it toggles on the positive transition of the clock and thus the slave toggles on the negative transition of the clock.

Master Slave flip flop: Avoidance of Race Around Condition of JK Flip Flop



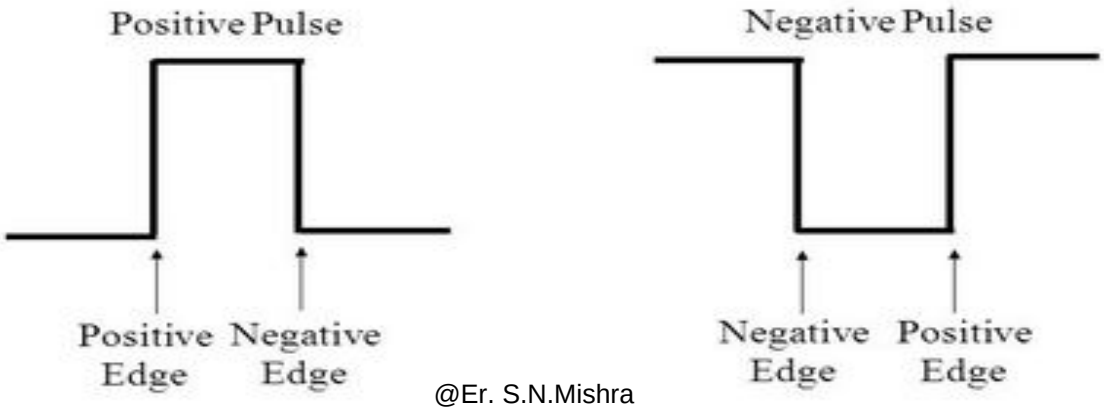
Master Slave flip flop: Avoidance of Race Around Condition of JK Flip Flop

Timing Diagram of a Master flip flop When J=1 and K=1



Triggering of Flip Flop

- The state of a flip-flop is switched by a momentary change in the input signal. This momentary change is called a ***trigger.***
- Clocked flip-flops are triggered by ***pulses***.
- A pulse starts from an initial value of 0, goes momentarily to 1, and after a short time, returns to its initial 0 value.
- A clock signal is a periodic square wave that indefinitely switches values from 0 to 1 and 1 to 0 at fixed intervals.
- A clock pulse may be either positive or negative.
  - A positive clock source remains at o during the interval between pulses and goes to 1 during the occurrence of a pulse.



Triggering of Flip Flop

- An edge triggered flip flop changes state either at positive edge(rising edge) or at negative edge(falling edge) of the clock pulse and is sensitive to its input only at this transition of the clock.
- A pulse triggered flip flop changes state either at the positive pulse (positive level of the pulse) or at negative pulse(negative level of pulse)of the applied clock pulse.

Level Triggering	Edge Triggering
• Outputs change based on inputs whenever *clock* is high • Memory will be considered to be level triggered (for cost reasons)	• Outputs change based on inputs only when clock transitions • Positive edge triggered logic when leading edge cause triggering • Negative edge triggered when trailing edge causes triggering

Triggering of Flip Flop

These circuits respond to their inputs on either the **rising** or **falling** edge of the clock — a precise point in time rather than an interval.

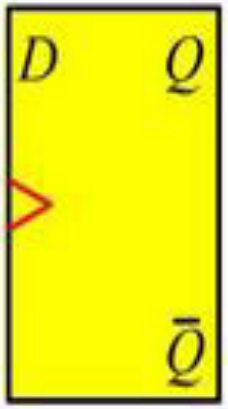
Positive edge triggered

rising edge of clock

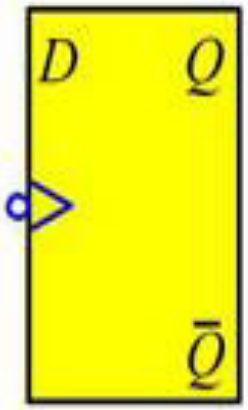
Negative edge triggered

falling edge of clock

wedge shows positive edge triggering



additional of a circle means that there is negative edge triggering



Edge-Triggered Flip-Flop

- Edge-triggered flip-flop (alternative to master-slave) synchronizes the state changes during clock-pulse transitions.
- In this type of flip-flop, output transitions occur at a specific level of the clock pulse.
- When the pulse input level exceeds this threshold level, the inputs are locked out and the flip-flop is therefore unresponsive to further changes in inputs until the clock pulse returns to 0 and another pulse occurs.
- Some edge-triggered flip-flops cause a transition on the positive edge of the pulse, and others cause a transition on the negative edge of the pulse.
- The logic diagram of a D-type positive-edge-triggered flip-flop is shown below.
- It consists of three basic flip-flops. NAND gates 1 and 2 make up one basic flip-flop and gates 3 and 4 another. The third basic flip-flop comprising gates 5 and 6 provides the outputs to the circuit. Inputs S and R of the third basic flip-flop must be maintained at logic-1 for the outputs to remain in their steady state values.

Edge-Triggered Flip-Flop

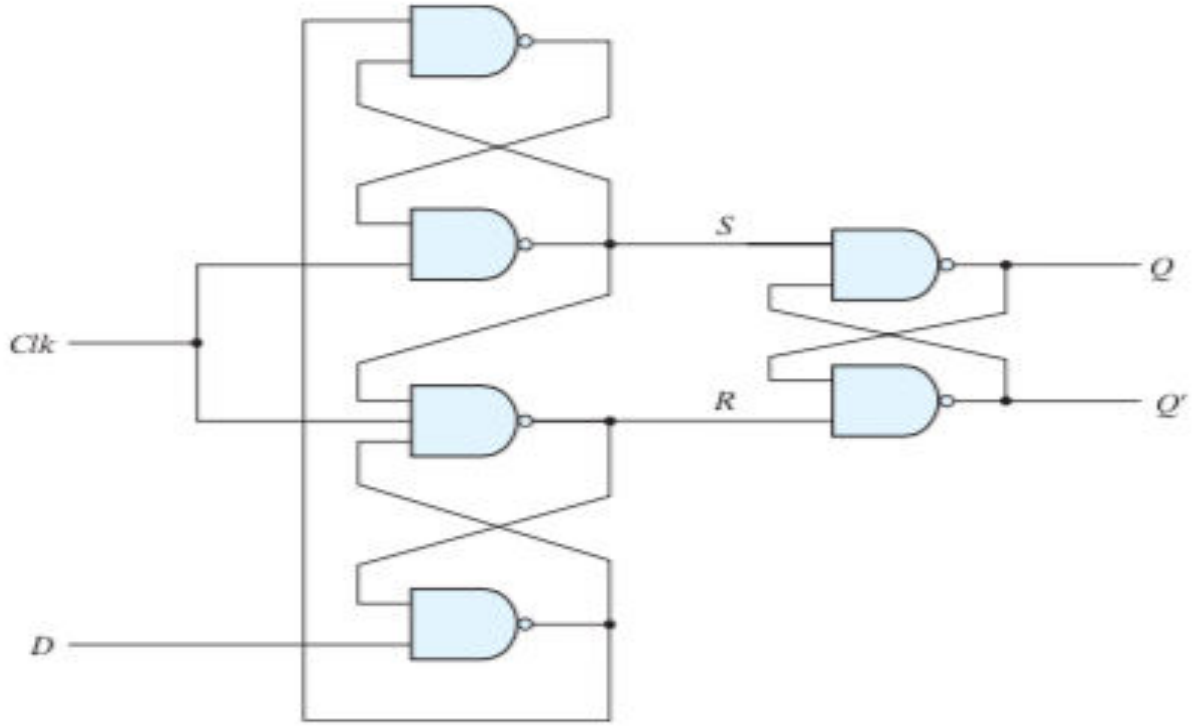


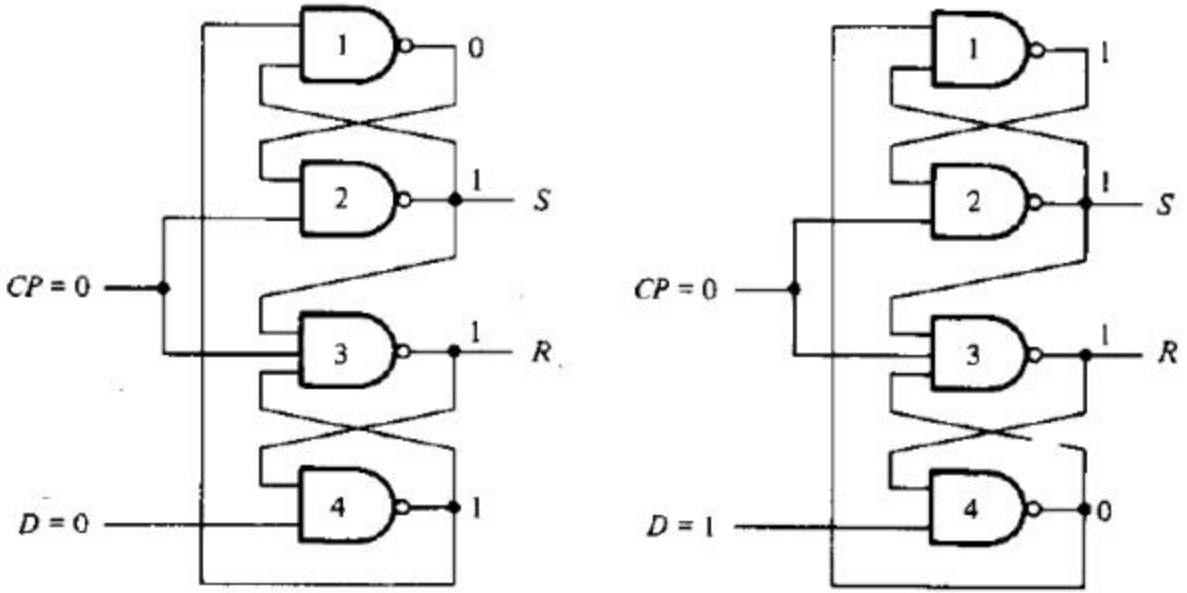
FIGURE
D-type positive-edge-triggered flip-flop

Edge-Triggered Flip-Flop

- When S = 0 and R = 1, the output goes to the set state with Q = 1.
- When S = 1 and R = 0, the output goes to the clear state with Q = 0.
- Inputs S and R are determined from the states of the other two basic flip-flops. These two basic flip-flops respond to the external inputs D (data) and CP (clock pulse).

Edge-Triggered Flip-Flop

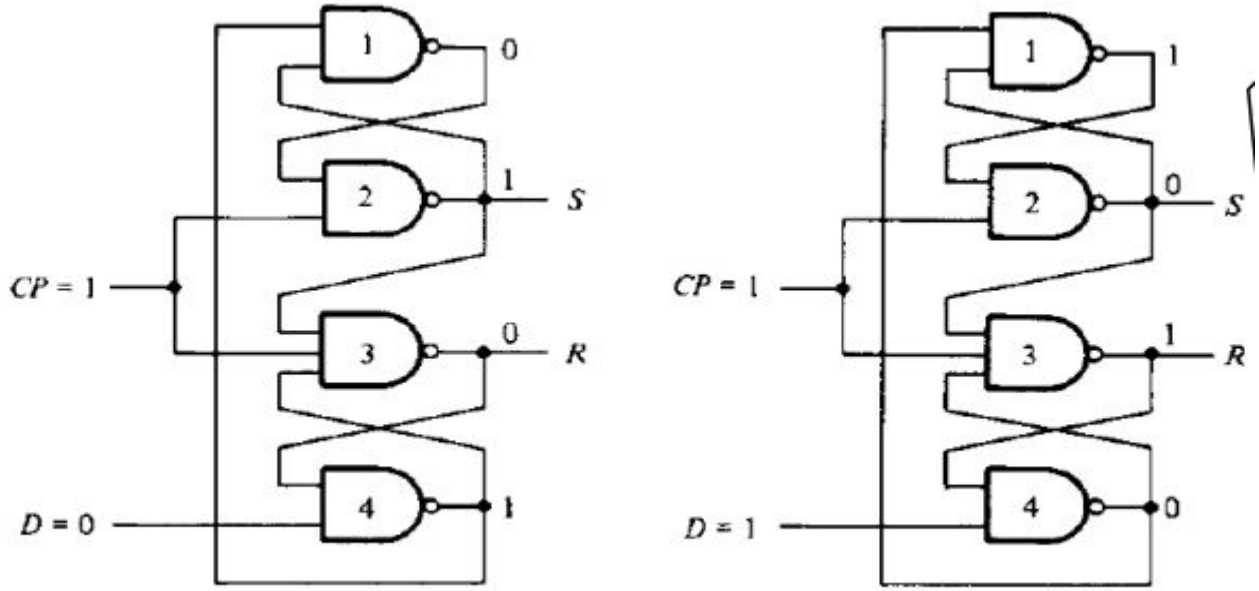
- Fig (a) shows the binary values at the outputs of the four gates when CP = 0. Input D may be equal to 0 or 1. In either case, a CP of 0 causes the outputs of gates 2 and 3 to go to 1, thus making S = R = 1, which is the condition for a steady state output.
- Operation



(a) With $CP = 0$

Edge-Triggered Flip-Flop

- When CP = 1
  - If D = 1 then S changes to 0, but R remains at 1, which causes the output of the flip-flop Q to go to 1 (set state).
  - If D = 0 then S = 1 and R = 0. Flip-flop goes to clear state (Q = 0).

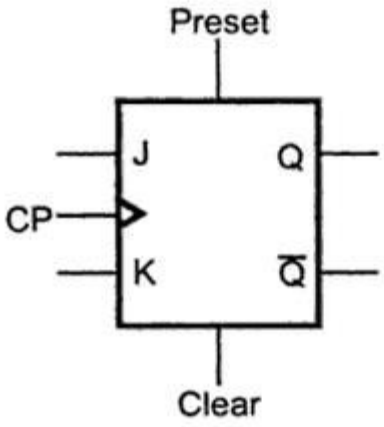


(b) With $CP = 1$

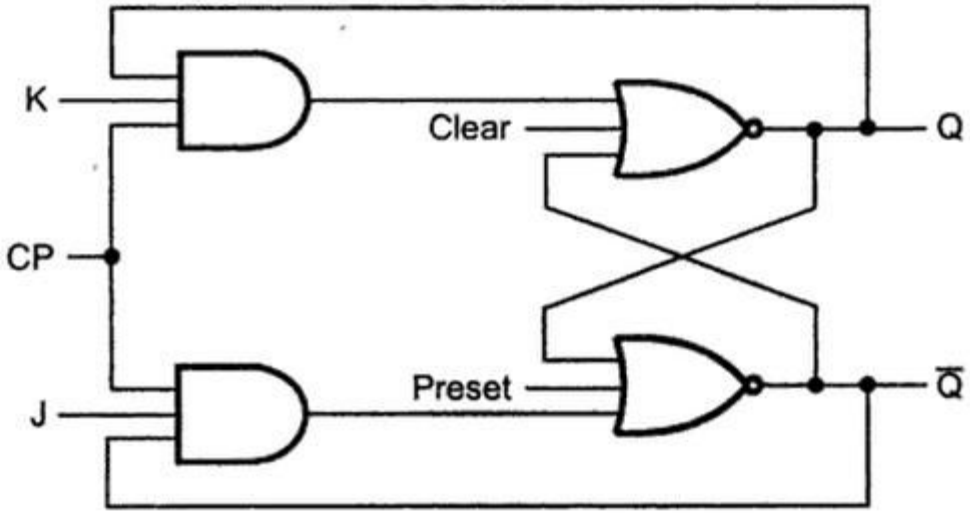
Direct Inputs

- Flip-flops available in IC packages sometimes provide special inputs for setting or clearing the flip-flop asynchronously.
- These inputs are usually called direct preset and direct clear. They affect the flip-flop on a positive (or negative) value of the input signal without the need for a clock pulse.
- These inputs are useful for bringing all flip-flops to an initial state prior to their clocked operation.
- Example: After power is turned on in a digital system, the states of its flip-flops are indeterminate.
- A clear switch clears all the flip-flops to an initial cleared state and a start switch begins the system's clocked operation. The clear switch must clear all flip-flops asynchronously without the need for a pulse.

Direct Inputs

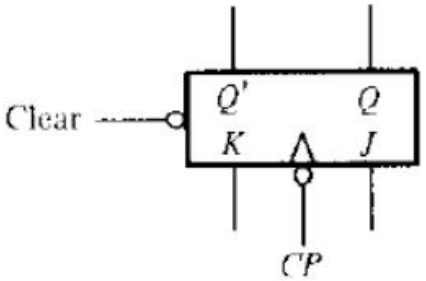


(a) Logic symbol



(b) JK flip-flop with active low preset and clear

Direct Inputs



Function table

Inputs				Outputs	
Clear	Clock	J	K	Q	Q'
0	X	X	X	0	1
1	↓	0	0	No change	
1	↓	0	1	0	1
1	↓	1	0	1	0
1	↓	1	1	Toggle	

Fig: JK flip-flop with direct clear

Analysis of Clocked Sequential Circuit

- The behavior of a clocked sequential circuit is determined from its inputs, outputs and state of the flip-flops (i.e., the output of the flip-flops).
- The analysis of a clocked sequential circuit consists of obtaining a table of a diagram of the time sequences of inputs, outputs and states.
  - E.g., given a current state and current inputs, how will the state and outputs change when the next active clock edge arrives???

Analysis of Clocked Sequential Circuit

- We have a basic procedure for analyzing a clocked sequential circuit:
  - ✓ Write down the equations for the outputs and the flip-flop inputs.
  - ✓ Using these equations, derive a state table which describes the next state.
  - ✓ Obtain a state diagram from the state table.
- It is the state table and/or state diagram that specifies the behavior of the circuit.
- The flip-flop input equations are sometimes called the excitation equations.
- The state table is sometimes called a transition table.

Analysis Example

- Is this a clocked sequential circuit?

  YES!

- What type of Memory?

  D Flip-Flops

- How many state variables?

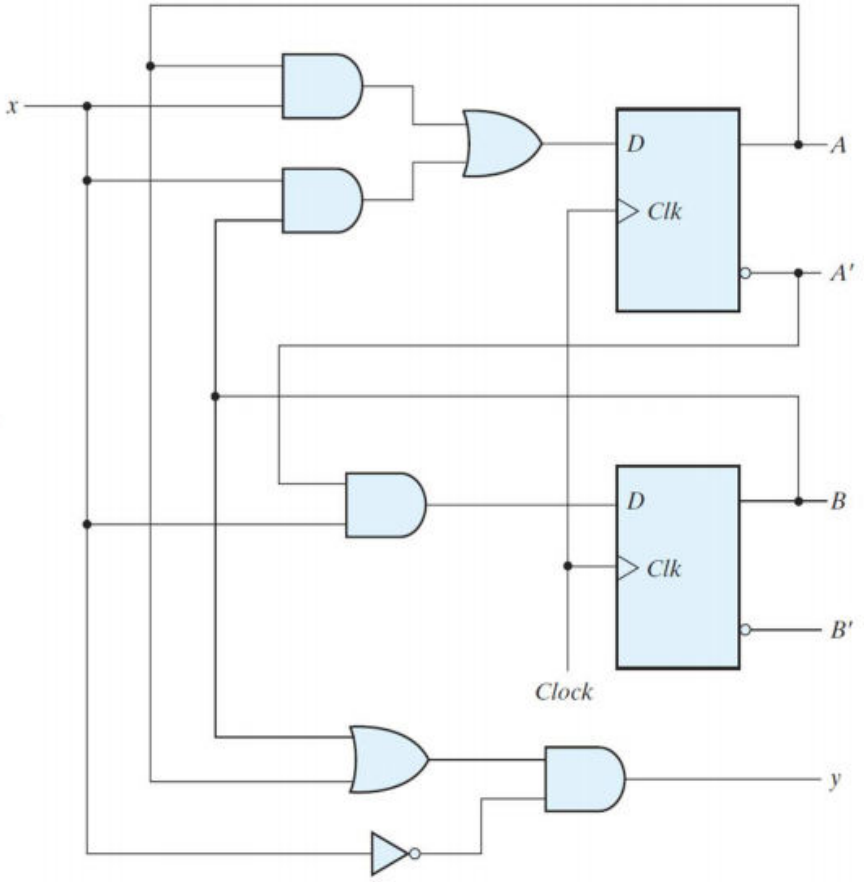
  Two state variables: A and B

- What are the Inputs?

  One Input: x

- What are the Outputs?

  One Output: y



State Equations

- A state equation is an algebraic expression that specifies the condition for a flip-flop state transition.
- The left side of the equation denotes the next state of the flip-flop and the right side of the equation is a Boolean expression that specifies the present state and input conditions that make the next state equal to 1.
- In above example, D inputs determine the flip-flop's next state, so it is possible to write a set of next-state equations for the circuit:

$$A(t + 1) = A(t)x(t) + B(t)x(t)$$

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

- In compact form:

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

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

- Similarly, the present-state value of the output y can be expressed algebraically as : $y(t) = [A(t) + B(t)]x'(t)$
- Removing the symbol (t) for the present state, the output Boolean function: $y = (A + B)x'$

State table

- The time sequence of inputs, outputs, and flip-flop states can be enumerated in a state table.
- The state table for the example circuit above is shown in the table below.

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

The table consists of four sections:
Present state: shows the states of flip-flops A and B at any given time t
Input: gives a value of x for each possible present state
Next state: shows the states of the flip-flops one clock period later at time t + 1.
Output: gives the value of y for each present state.

Next state and output column is derived from the state equations

State table

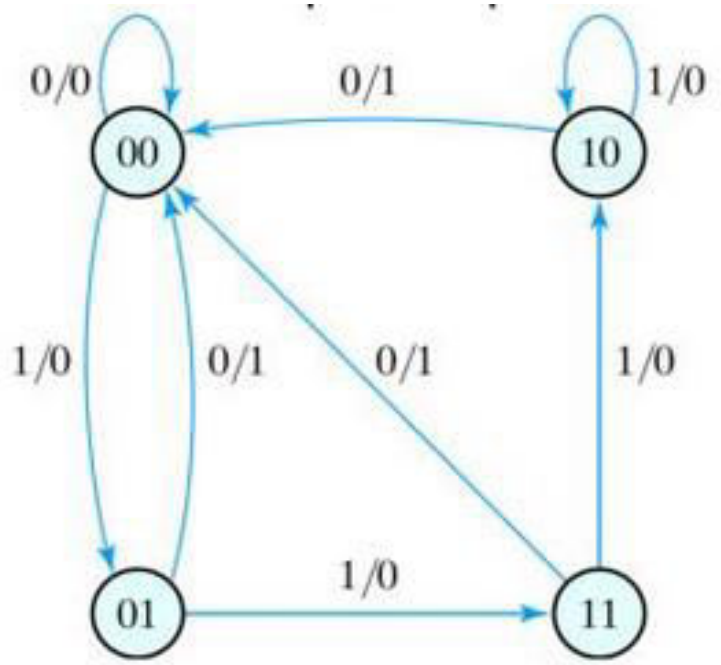
Second Form of the State Table

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 a state diagram.
- In this diagram, a state is represented by a circle, and the transition between states is indicated by directed lines connecting the circles.
- Each directed line is labeled ‘inputs/outputs’

- a logic diagram ⇔ a state table ⇔ a state diagram



Analysis Example 1

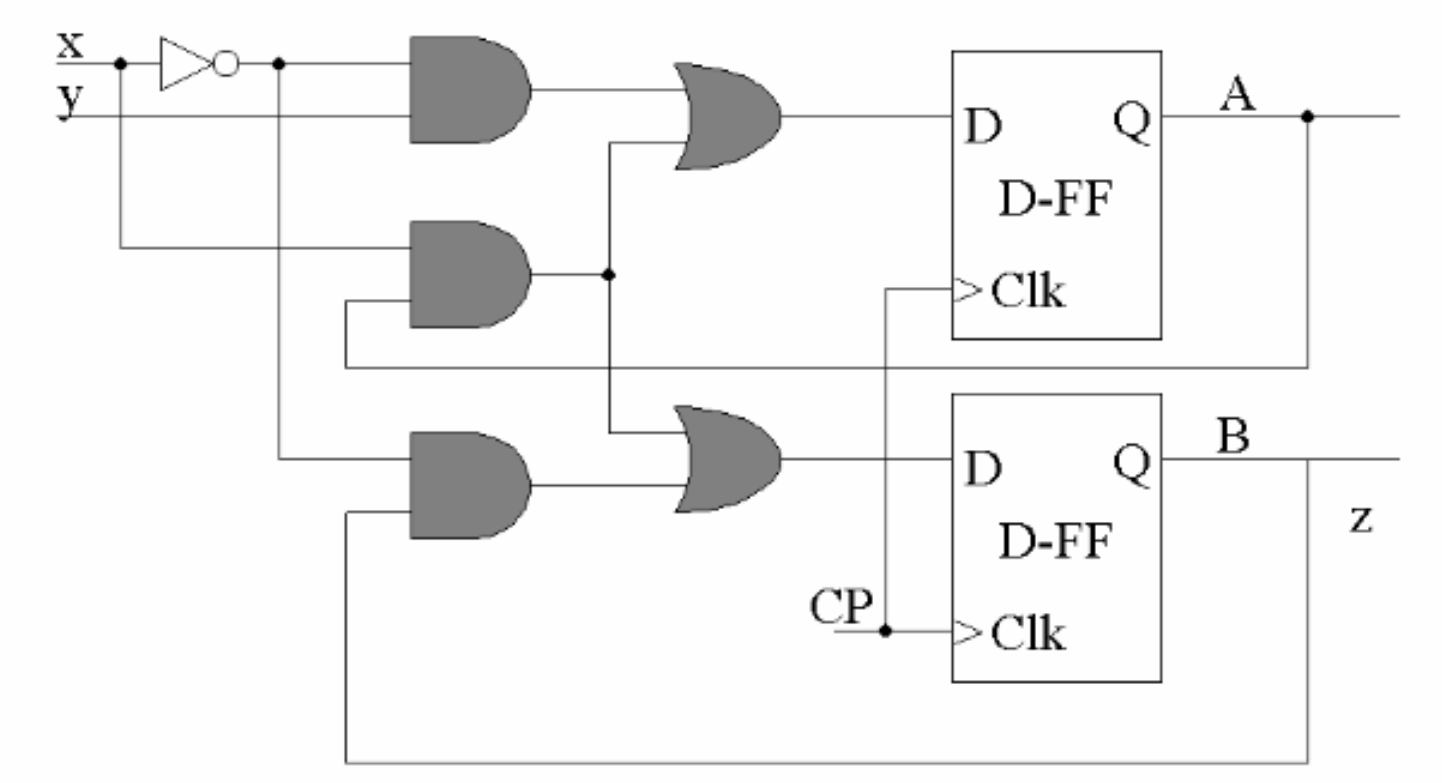
A sequential circuit with two D Flip-Flops, A and B; two inputs, x and y; and one output, z, is specified by the following next-state and output equations:

$$A(t+1) = x'y + xA$$
$$B(t+1) = x'B + xA$$
$$z = B$$

a) Draw the logic diagram of the circuit.
b) List the state table for the sequential circuit.
c) Draw the corresponding state diagram.

Analysis Example 1: Solution

a)



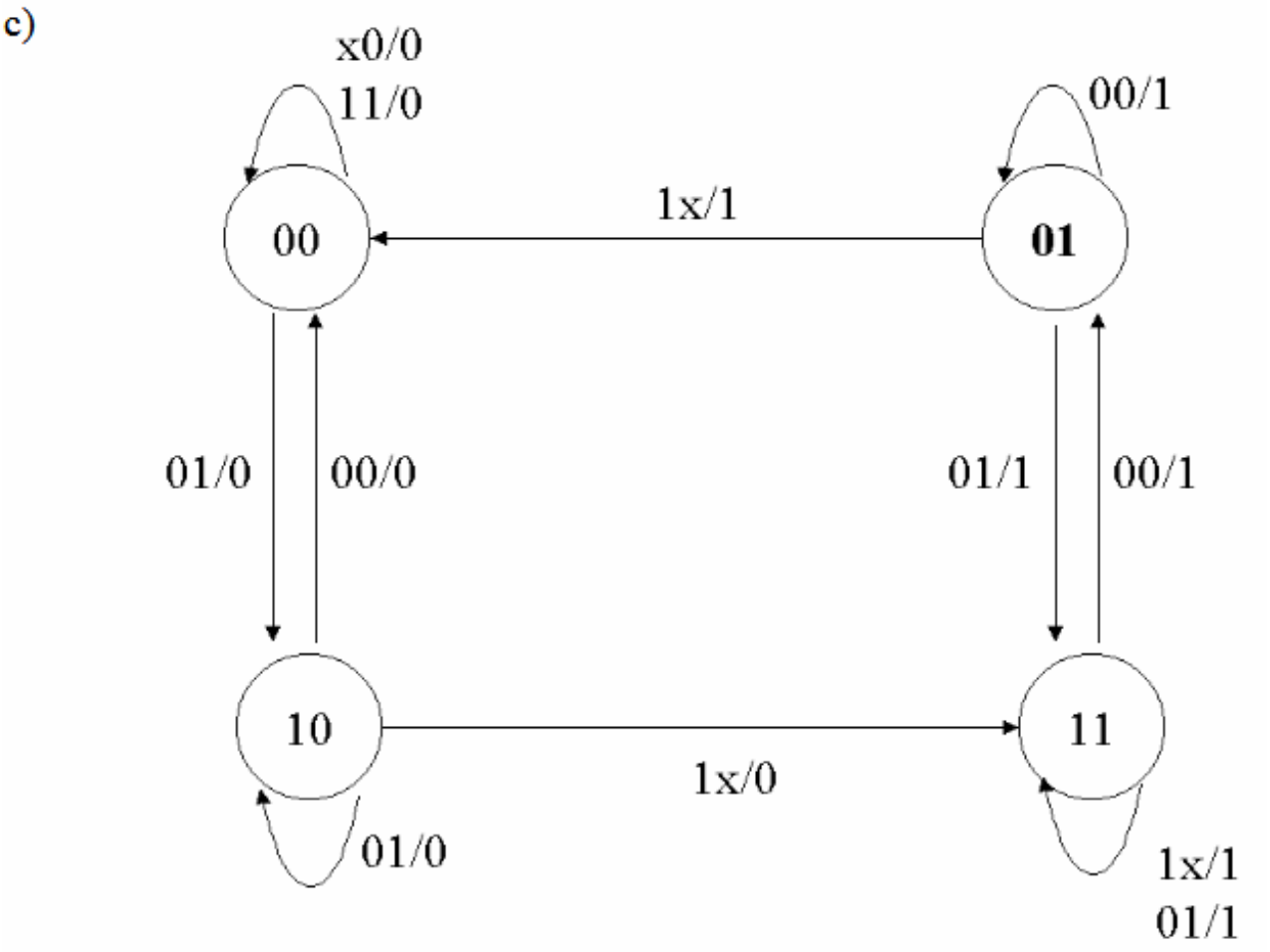
Analysis Example 1: Solution

b)

Present State		Inputs		Next State		Output
A	B	x	y	A	B	z
0	0	0	0	0	0	0
0	0	0	1	1	0	0
0	0	1	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	1	1
0	1	0	1	1	1	1
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	0	0	0
1	0	0	1	1	0	0
1	0	1	0	1	1	0
1	0	1	1	1	1	0
1	1	0	0	0	1	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1

Analysis Example 1: Solution

c)



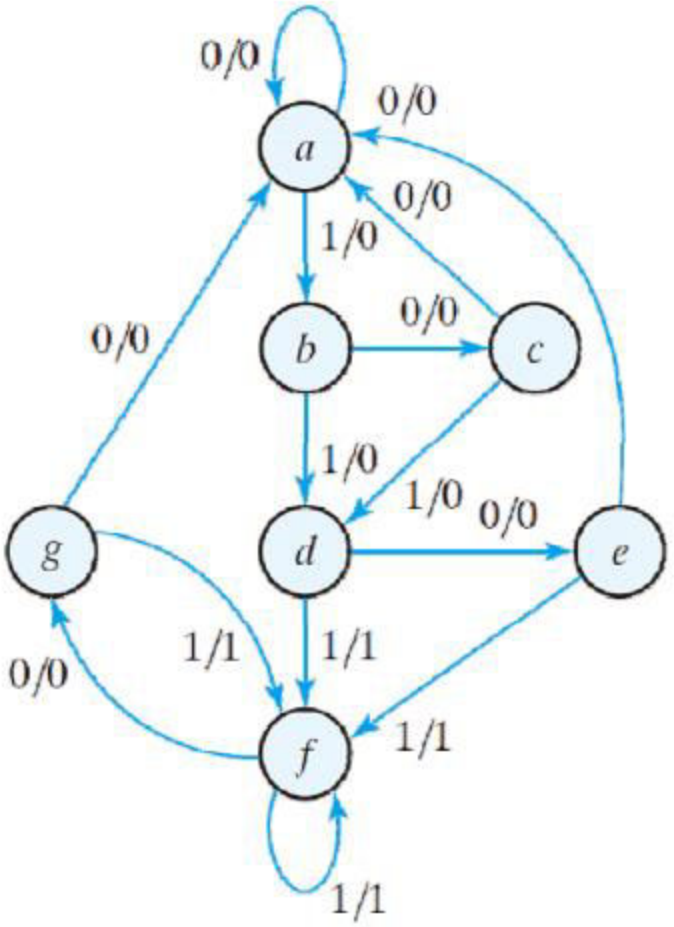
State Reduction and Assignment

State Reduction

- The reduction of the number of flip-flops in a sequential circuit is referred to as the state-reduction problem.
- State-reduction, reducing the number of states in a state table, while keeping the external input–output requirements unchanged, can reduce the number of flip-flops used in a sequential circuit.
- Since m flip-flops produce 2^m states, a reduction in the number of states may (or may not)result in a reduction in the number of flip-flops.
- Reducing the number of flip-flops sometimes results the equivalent circuit with fewer flip-flops but more combinational gates to realize its next state and output fewer flip flops but more combinational gates to realize its next state and output logic.

State Reduction Example

- Two circuits are equivalent if identical input sequences are applied to the two circuits and identical outputs occur for all input sequences, then one may be replaced by the other.
- State reduction reduces the number of states in a sequential circuit without altering the input–output relationships.
- Only the input-output sequences are important in this example.
- Consider the input sequence 01010110100 starting from the initial state a



State Reduction Example

state	*a*	*a*	*b*	*c*	*d*	*e*	*f*	*f*	*g*	*f*	*g*	*a*
input	0	1	0	1	0	1	1	0	1	0	0	
output	0	0	0	0	0	1	1	0	1	0	0	

- State table is more convenient for state reduction than a diagram.
- State reduction algorithm: "Two states are said to be equivalent if, for each member of the set of inputs, they give exactly the same output and send the circuit either to the same state or to an equivalent state."
- When two states are equivalent, one of them can be removed without altering the input–output relationships.

State Reduction Example

Original State Table

Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

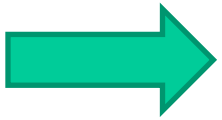
e=g

If next state and output of two present states are same then We can eliminate one state.

State Reduction Example

Reducing the State Table

Present State	Next State $x=0$	Next State $x=1$	Output $x=0$	Output $x=1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	e	f	0	1

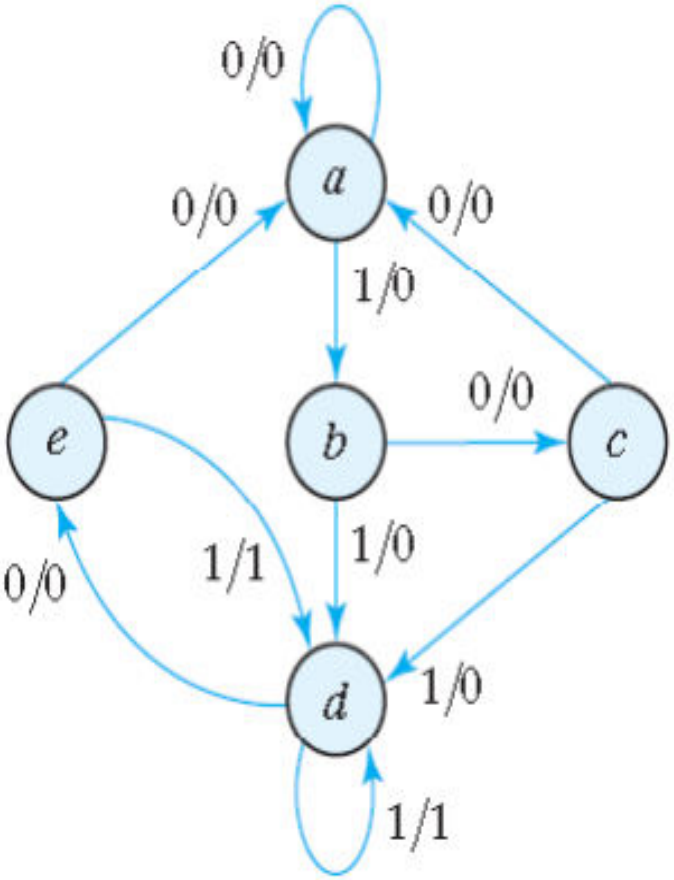


Reduced State Table

Present State	Next State $x=0$	Next State $x=1$	Output $x=0$	Output $x=1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	d	0	1
e	a	d	0	1

State Reduction Example

Reduced State Diagram



State Reduction Example

Reduce the number of states in the following state table and tabulate the reduced state table.

Present State	Next State x =0	Next State x =1	Output x =0	Output x =1
a	f	b	0	0
b	d	c	0	0
c	f	e	0	0
d	g	a	1	0
e	d	c	0	0
f	f	b	1	1
g	g	h	0	1
h	g	a	1	0

State Reduction Example

Present State	Next State x =0	Next State x =1	Output x =0	Output x =1
a	f	b	0	0
b	d	a	0	0
d	g	a	1	0
f	f	b	1	1
g	g	d	0	1

State Assignment

- The cost of the combinational-circuit part of a sequential circuit can be reduced by using the known simplification methods for combinational circuits.
- However, there is another factor, known as the state-assignment problem that comes into play in minimizing the combinational gates.
- State-assignment procedures are concerned with methods for assigning binary values to states in such a way as to reduce the cost of the combinational circuit that drives the flip-flops.
- States must be assigned with unique coded binary values to implement the physical components.
- The simplest way to code states is to use binary counting code or Gray code without guaranteeing a better result.

State Assignment

Three Possible Binary State Assignments

State	Assignment 1, Binary	Assignment 2, Gray Code	Assignment 3, One-Hot
a	000	000	00001
b	001	001	00010
c	010	011	00100
d	011	010	01000
e	100	110	10000

State Assignment-Binary Assignment

- A different assignment will result in a state table with different binary values for the states.
- The binary form of the state table is used to derive the next state and output-forming combinational logic part of the sequential circuit.
- The complexity of the combinational circuit depends on the binary state assignment chosen.
- Sometimes, the name transition table is used for a state table with a binary assignment.

Reduced State Table with Binary Assignment 1

		Next State		Output	
Present State		$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	000	000	001	0	0
b	001	010	011	0	0
c	010	000	011	0	0
d	011	100	011	0	1
e	100	000	011	0	1

A great many possible binary assignments may exist.

Design Procedure of Sequential Circuit

- A synchronous sequential circuit is made up of flip-flops and combinational gates.
- The design of the circuit consists of choosing the flip-flops and then finding a combinational gate structure that, together with the flip-flops, produces a circuit which fulfills the stated specifications.
- The design steps for synchronous sequential circuits can be summarized as:

1. From the word description and specifications of the desired operation, derive a state diagram for the circuit.
2. Obtain the State table
3. Reduce the number of states if necessary.
4. Assign binary values to the states.
5. Determine the number of flip-flops needed and assign a letter symbol to each.
6. Choose the type of flip-flops to be used
7. From the state table, derive the circuit excitation and output tables.
8. Derive the simplified flip-flop input equations and output equations.
9. Draw the logic diagram.

Design Example

- The state diagram consists of four states with binary values already assigned.
- Directed lines contain single binary digit without a slash, we conclude that there is one input variable and no output variables. (The state of the flip-flops may be considered the outputs of the circuit).
- The two flip-flops needed to represent the four states are designated A and B.
- The input variable is designated x.

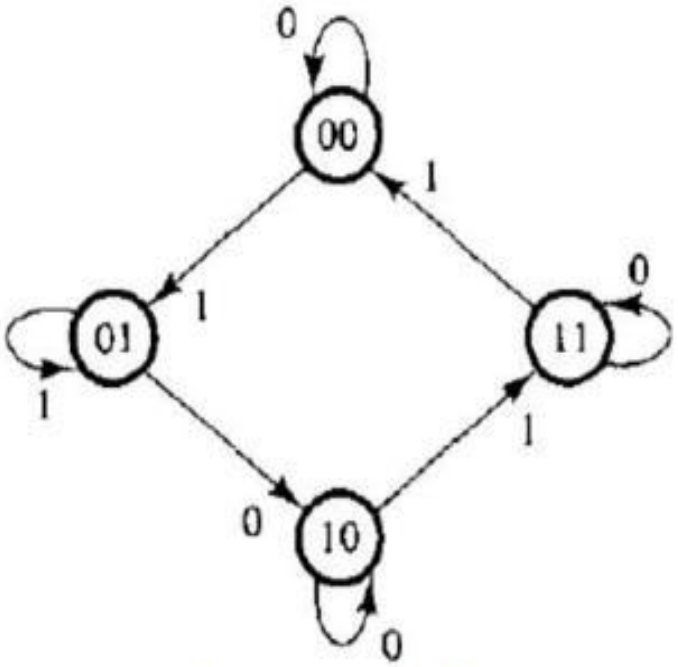


Fig: State-diagram for design example

Design Example

Present State		Next State			
		$x = 0$		$x = 1$	
A	B	A	B	A	B
0	0	0	0	0	1
0	1	1	0	0	1
1	0	1	0	1	1
1	1	1	1	0	0

Fig: State Table

Flip-Flop Excitation Tables

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

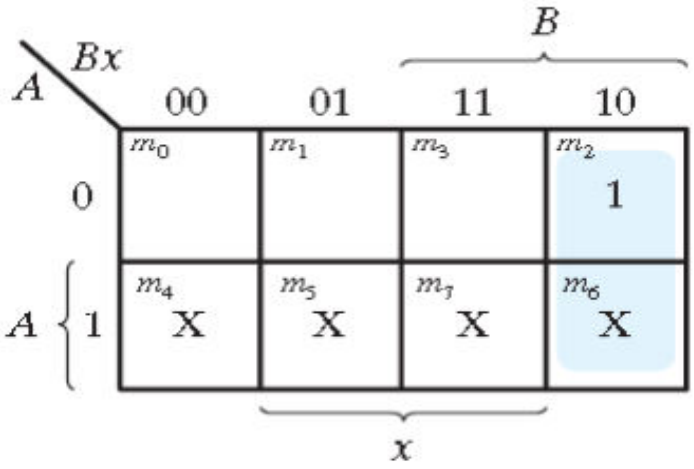
(a) JK Flip-Flop

Design Example Using JK Flip Flop

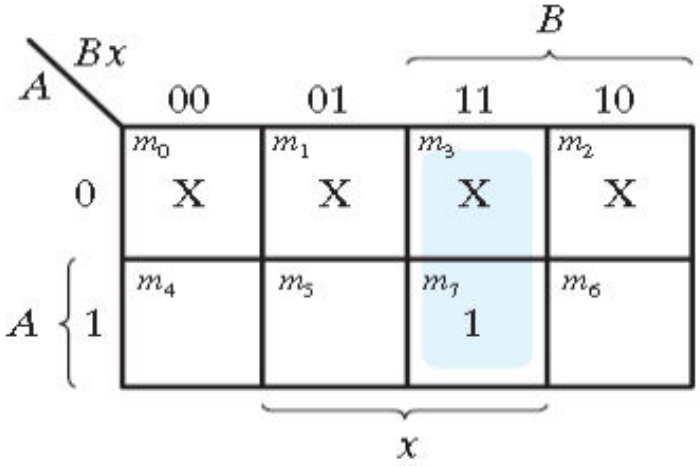
State Table and JK Flip-Flop Inputs

Present State		Input	Next State		Flip-Flop Inputs			
A	**B**	**x**	**A**	**B**	J_A	K_A	J_B	K_B
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

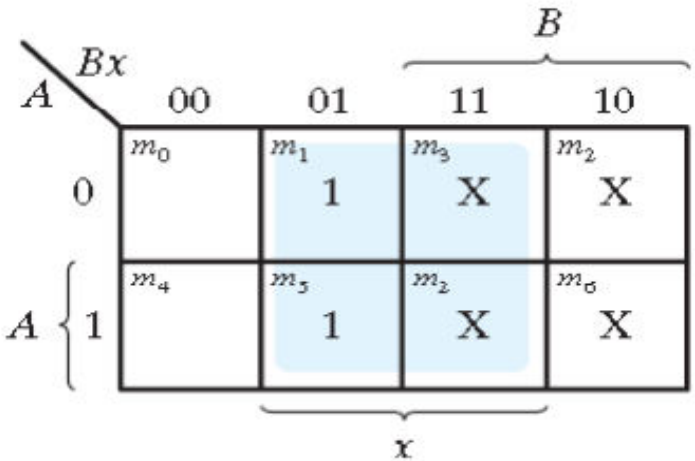
K-Maps for JK Input Equations



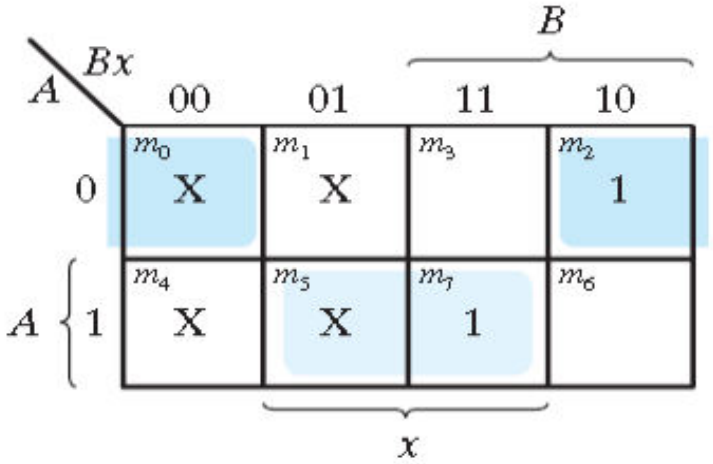
$J_A = Bx'$



$K_A = Bx$



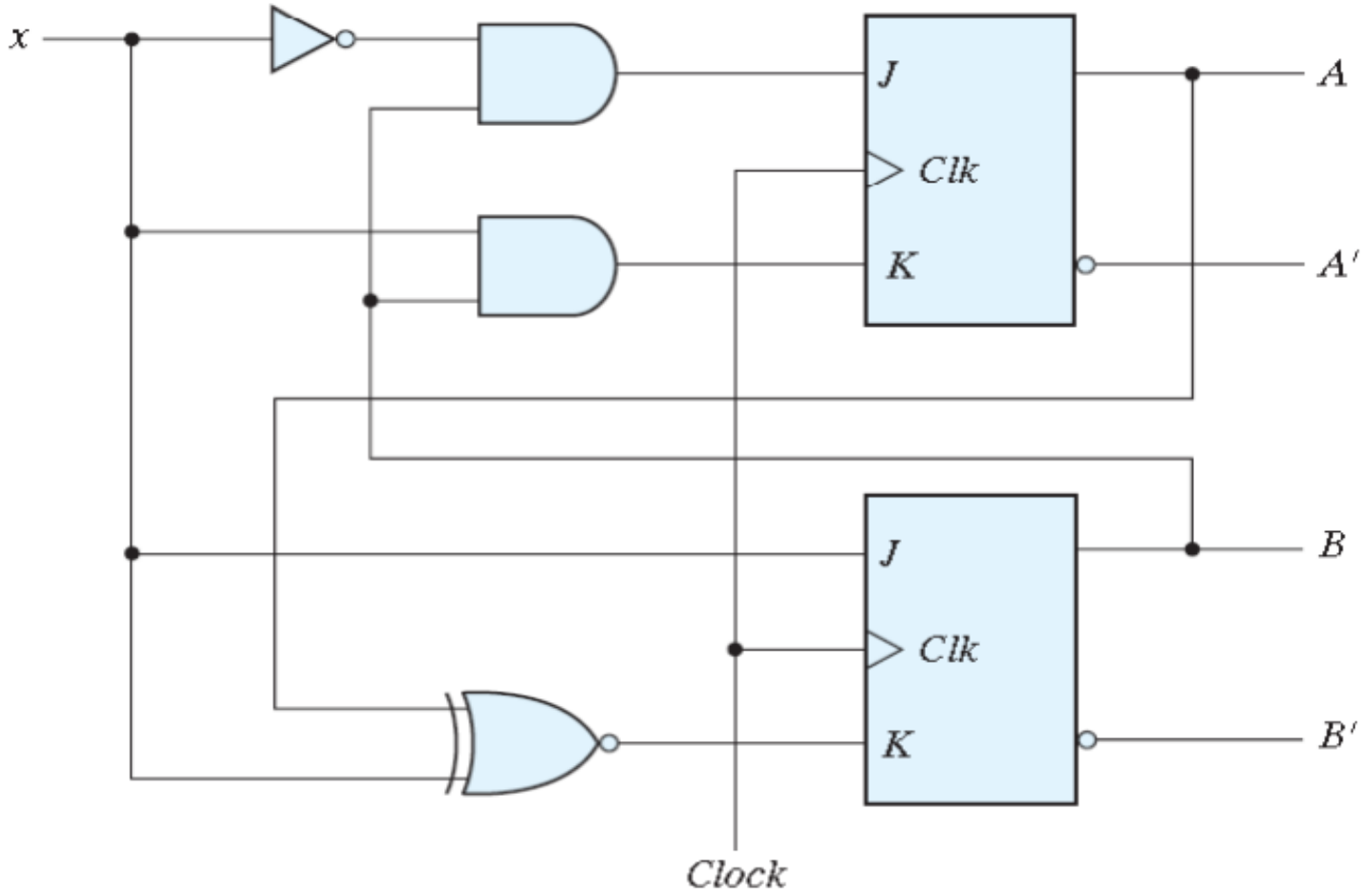
$J_B = x$



$K_B = (A \oplus x)'$

Logic Diagram with JK Flip-flops

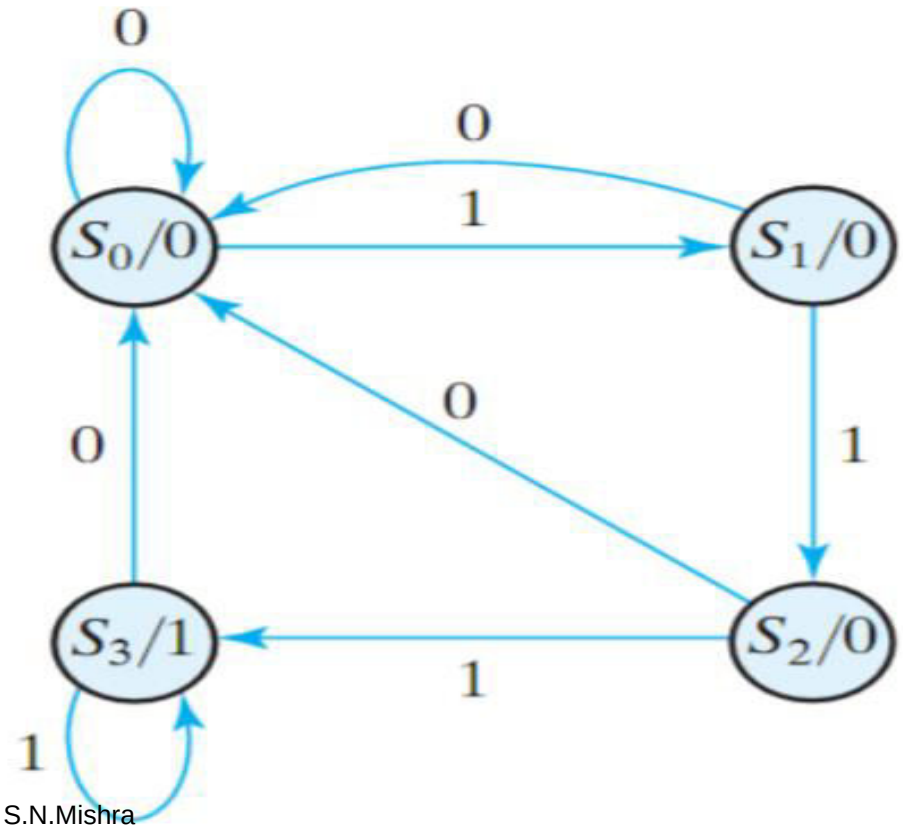
$J_A = Bx'$ $K_A = Bx$ $J_B = x$ $K_B = (A \oplus x)'$



Design Example 2

- Example: Design a circuit that detects a sequence of three consecutive 1's in a string of bits coming through an input line (serial bit stream)

- There are four states S_0, S_1, S_2, and S_3, which represents 0 at input, single 1, two consecutive 1's, or three or more consecutive 1's respectively.
- The output of this detector is 1 when 3 or more consecutive 1's are detected.



➢ Example – D FF implementation

- The four states are assigned binary codes 00,01,10,11 and hence two FF's A and B are needed.
- The state table is:
- The characteristic equations are:

$$A(t+1) = D_A$$
$$B(t+1) = D_B$$

- From the table we find:

$$A(t+1) = D_A(A,B,x) = \sum(3,5,7)$$

$$B(t+1) = D_B(A,B,x) = \sum(1,5,7)$$

Input Equations

State Table for Sequence Detector

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	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	1
1	1	0	0	0	0
1	1	1	1	1	1

➢ *Example – D FF implementation* (continue)

- Output equation is:

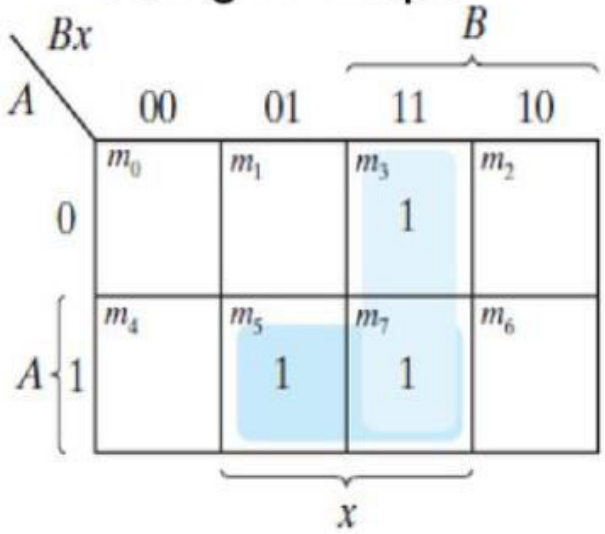
$$y(A, B, x) = \sum(6,7)$$

State Table for Sequence Detector

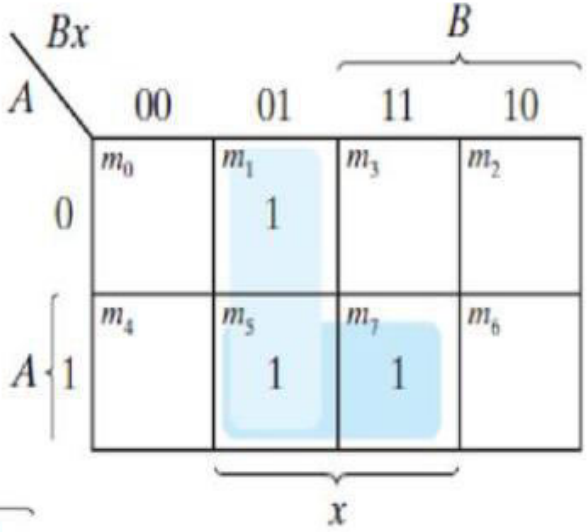
Present State		Input	Next State		Output
A	**B**	**x**	**A**	**B**	**y**
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

➢ *Example – D FF implementation* (continue)

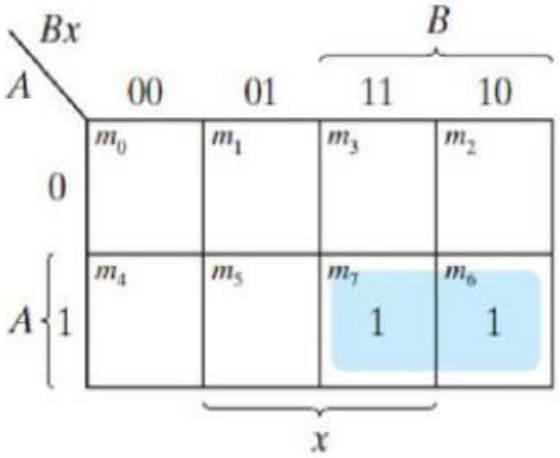
- Since the input and output equations are written down as sum of minterms, therefore the these equation can be simplified using K-maps.



$D_A = Ax + Bx$



$D_B = Ax + B'x$



$y = AB$

➢ Example – D FF implementation (continue)

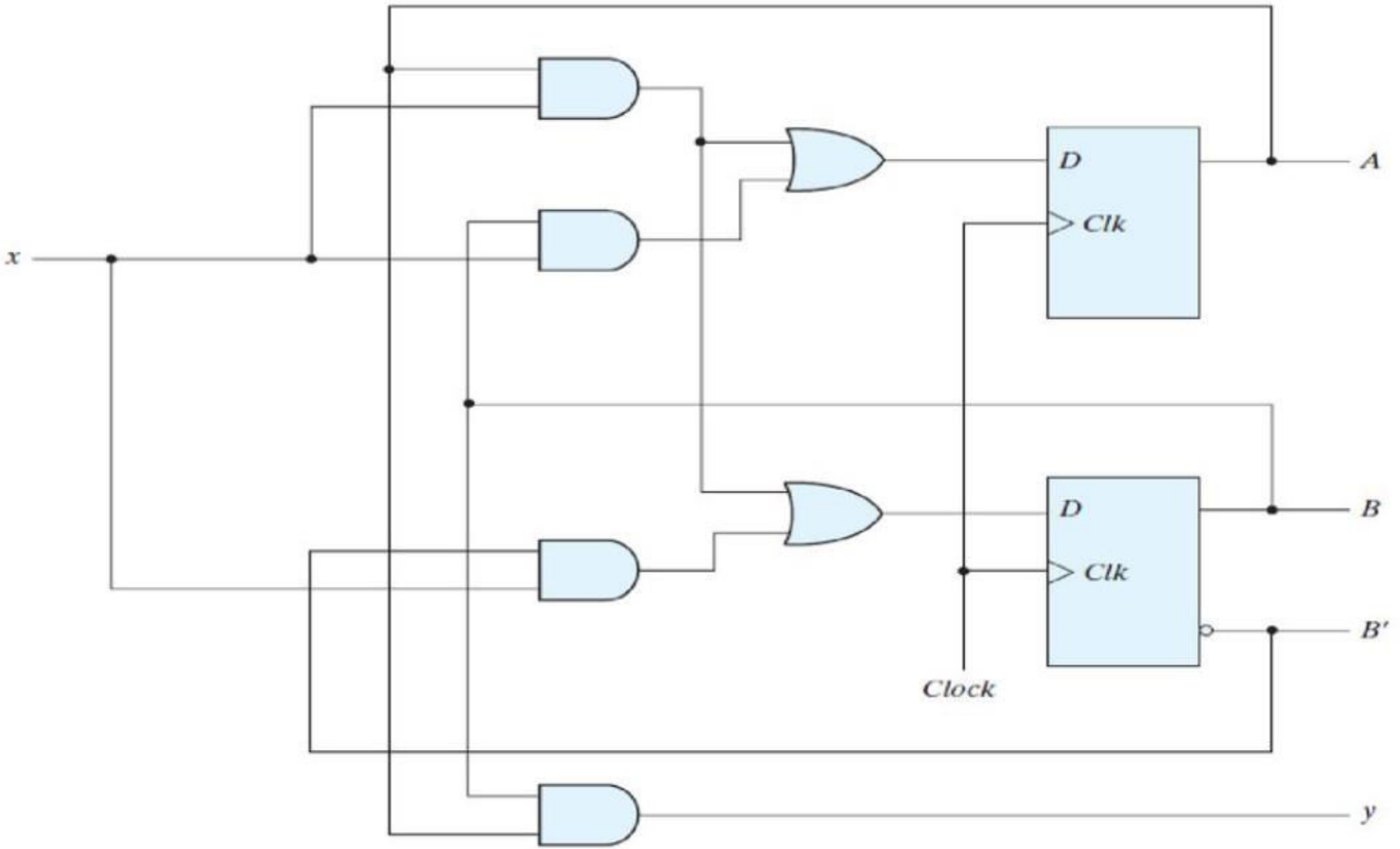


Fig: Logic diagram of sequence detector

Excitation Tables

- A table that lists required inputs for a given change of state (Present to next-state) is called an excitation table.

1. Excitation Table for SR F.F.

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

00 → No Change
01 → Reset

Characteristic Table

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

2. Excitation Table for JK F.F.

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

00
01
10
11
01
11
00
10

Excitation Tables

3. Excitation Table for T F.F.

Q(t)	Q(t + 1)	T	
0	0	0	No Change
0	1	1	Complement
1	0	1	Complement
1	1	0	No Change

4. Excitation Table for D F.F.

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

Assignment Questions

1. Write the characteristic equations for JK and D Flip Flops.
2. What is meant by the term edge triggered?
3. How many ff's are required to design a mod-7 up down counter?
4. Define synchronous sequential circuit
5. Define flip flop.
6. What is race around condition?
7. What is the difference between synchronous and asynchronous Sequential Circuits?
8. What is a state equation?
9. A sequential circuit has 2D ff's A and B an input x and output y is specified by the following next state and output equations.
   A (t+1)= Ax + Bx B (t+1)= A'x Y= (A+B) x'
   (i) Draw the logic diagram of the circuit. (ii) Derive the state table. (iii) Derive the state diagram.

Assignment Questions

1. Explain the Logic diagram of JK flip-flop?
2. Write difference between Combinational & Sequential circuits?
3. Explain the Logic diagram of SR flip-flop?
4. Draw and explain the operation of T Flip-Flop?
5. What is state assignment? Explain with a suitable example?
6. Explain the working of the following i) J-K flip-flop ii) S- R flip-flop iii) D flip-flop
7. What is race-around condition? How does it set eliminate is a Master –slave J-K flip-flop?
8. Write the truth table of clocked T- Flip Flop?
9. Write the differences between latches and flip flops?
10. State the excitation table of JK Flip Flop.

Assignment Questions

- Draw the diagram of T flip flop and discuss its working.
- Give the block diagram of master-slave D flip- flop.
- Write the characteristics table and equation of JK flip flop.
- What is meant by triggering of Flip flop?
- How race around condition can be eliminated?
- What is a state diagram?
- What do you meant by the term state reduction problem?
- Design a sequence detector that detects a sequence of three or more consecutive 1's in a string of bits coming through an input line and produces an output whenever this sequence is detected

Assignment Questions

- A sequential circuit with two D flip-flops A and B, one input x and one output z is specified by the following next-state and output equations: A(t+1)= A′+B, B(t+1)=B′x, z=A+B′

i. Draw the logic diagram of the circuit

ii. Draw the state table

iii. Draw the state diagram of the circuit

iv. Explain the difference between a state table, characteristics table and excitation table.