Chapter-6

Synchronous and Asynchronous Sequential Logic

Introduction to sequential circuit

We study combinational circuits in which the outputs at any instant of time are entirely dependent upon the inputs present at that time. Although every digital system is likely to have combinational circuits, most systems encountered in practice also include memory elements, which require that the system be described in terms of sequential logic.

Memory elements are devices capable of storing binary information within them. The binary information stored in the memory elements at any given time defines the state of the sequential circuit. Block diagram shows external outputs in a sequential circuit are a function not only of external inputs, but also of the present state of the memory elements. Thus, a sequential circuit is specified by a time sequence of inputs, outputs, and internal states.

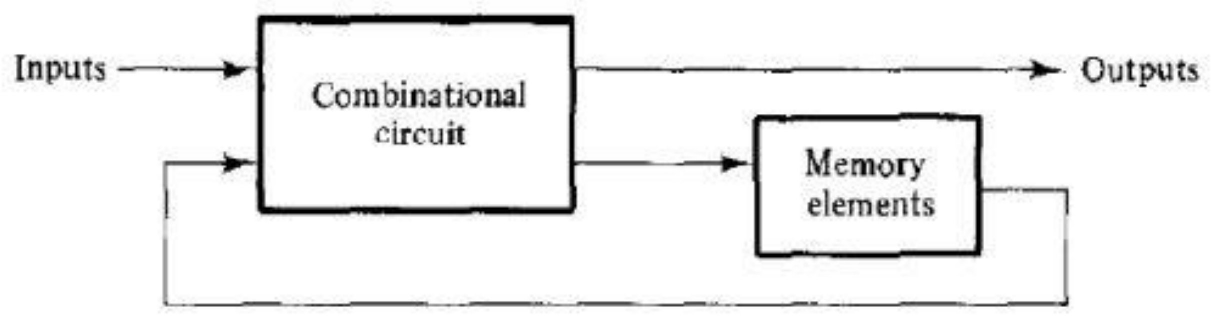


Fig: Block diagram of sequential circuit

Introduction sequential circuit

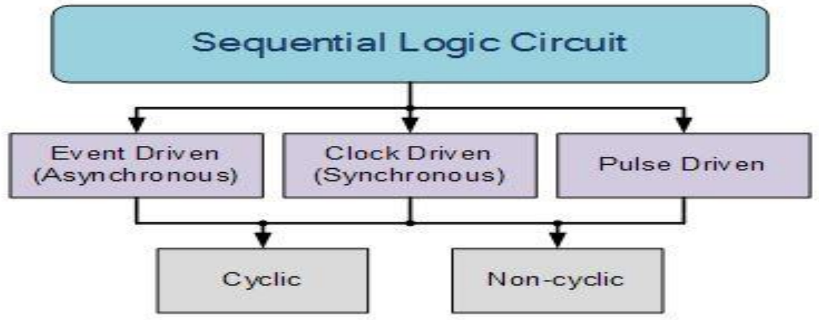
The word “Sequential” means that things happen in a “sequence”, one after another and in Sequential Logic circuits, the actual clock signal determines when things will happen next. Simple sequential logic circuits can be constructed from standard Bistable circuits such as: Flip-flops, Latches and Counters and which themselves can be made by simply connecting together universal NAND Gates and/or NOR Gates in a particular combinational way to produce the required sequential circuit.

Classification of Sequential Logic

1. Event Driven – asynchronous circuits that change state immediately when enabled.

2. Clock Driven – synchronous circuits that are synchronized to a specific clock signal.

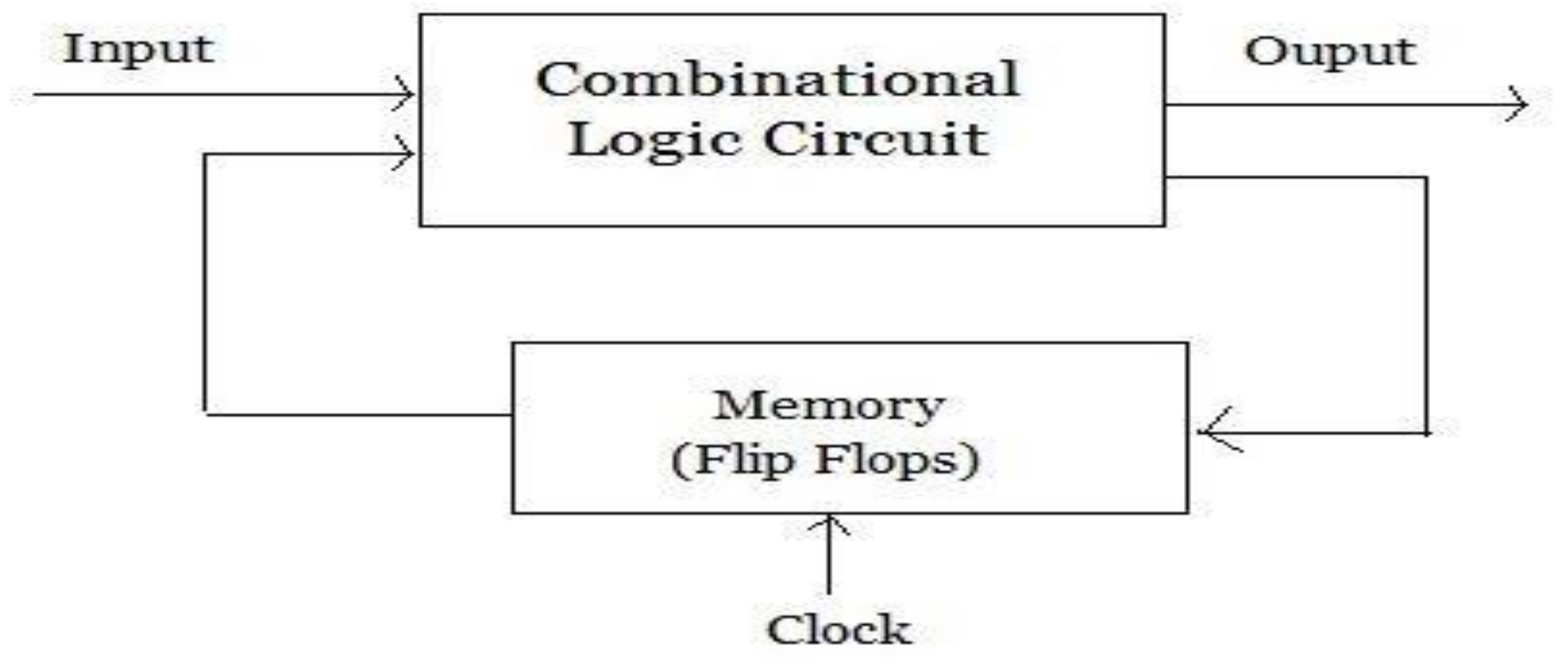
3. Pulse Driven – which is a combination of the two that responds to triggering pulses.



Types of sequential circuit

Synchronous sequential circuit:

A sequential circuit whose output behavior depends on the input at a discrete-time is called synchronous sequential circuits. Synchronous sequential circuits that use clock pulses in the inputs of memory elements are called clocked sequential circuits.

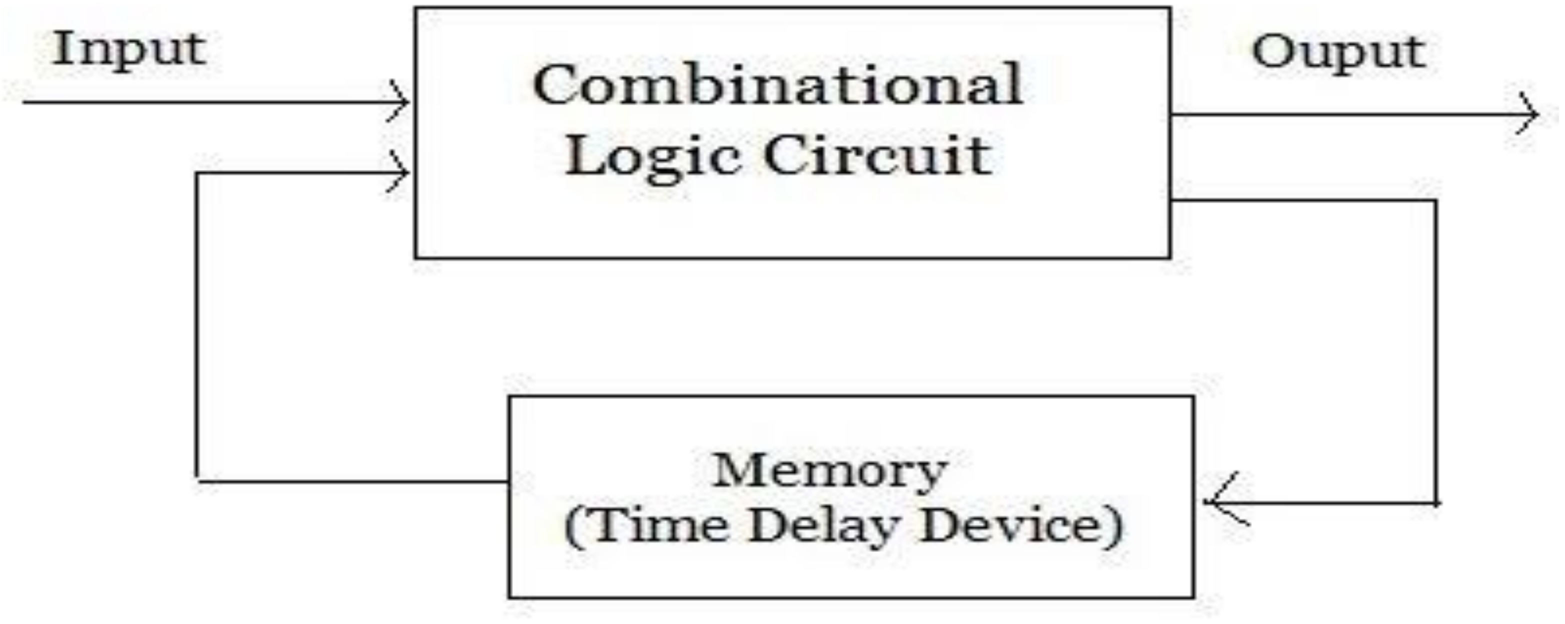


Synchronous sequential circuit

Types of sequential circuit

Asynchronous sequential circuit:

The sequential circuit whose output depends on the sequence in which the input changes are called asynchronous sequential circuits.



Asynchronous sequential circuit

Difference between synchronous and asynchronous

Synchronous	Asynchronous
Synchronous sequential circuits are digital circuits governed by clock signals	Asynchronous sequential circuits are digital circuits that are not driven by clock. They can be called as self-timed circuits.
Output behavior depends on the input at discrete time.	Output depends on the sequence in which the input changes.
In synchronous sequential circuits memory elements are used like clocked flip flop.	In asynchronous sequential circuits un-clocked memory elements are used like un-clocked flip flop or time delay elements.
Due to the presence of clock pulse the operating speed of synchronous sequential circuits is low.	Because of absence of clock pulse, it can be operate faster than Synchronous sequential circuits
In these circuits change in state occurs in response to clock pulse.	In these circuits state change occurs whenever input variable change.
These are more expensive.	These are economical.

Difference between combinational and sequential circuit

Combinational	Sequential
Combinational Circuit is the type of circuit in which output is independent of time and only relies on the input present at that particular instant.	On other hand Sequential circuit is the type of circuit where output not only relies on the current input but also depends on the previous output.
In Combinational circuit as output does not depend on the time instant, no feedback is required for its next output generation.	On other hand in case of Sequential circuit output relies on its previous feedback so output of previous input is being transferred as feedback used with input for next output generation.
As the input of current instant is only required in case of Combinational circuit, it is faster and better in performance as compared to that of Sequential circuit.	On other hand Sequential circuit are comparatively slower and has low performance as compared to that of Combinational circuit.
Elementary building blocks for combinational circuit are logic gates.	On other hand building blocks for sequential circuit are flip flops..
Combinational circuit are mainly used for arithmetic as well as Boolean operations	On other hand Sequential circuit is mainly used for storing data.
Block diagram?	Block diagram?

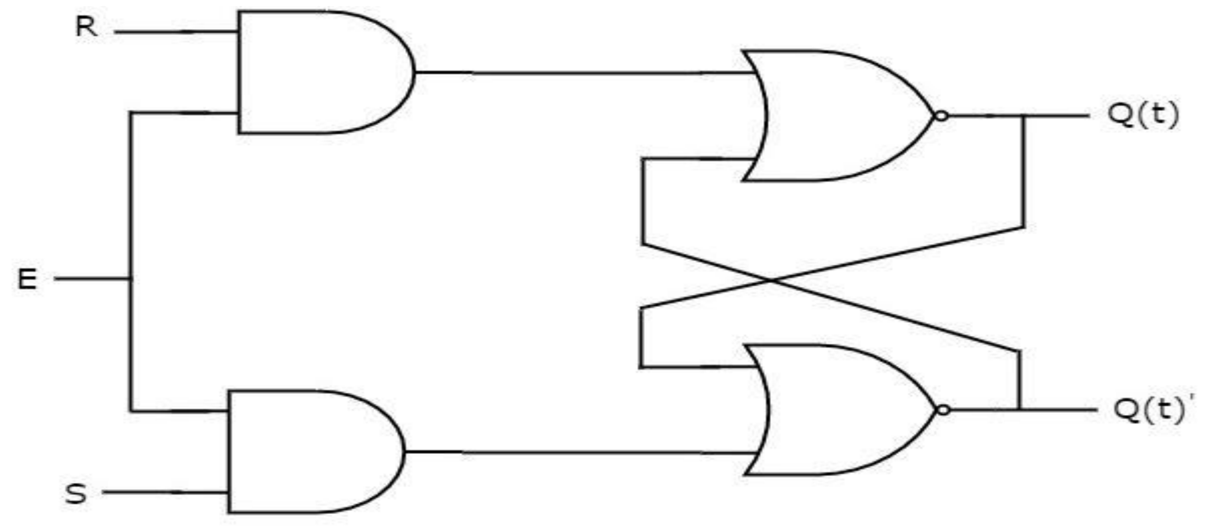
Latch and flip-flop

Latch:

Latch is an electronic device, which changes its output immediately based on the applied input. It is used to store either 1 or 0 at any specified time. It consists of two inputs namely "SET" and RESET and two outputs, which are complement to each other.

SR Latch:

SR Latch is also called as Set Reset Latch. This latch affects the outputs as long as the enable, E is maintained at '1'. The circuit diagram of SR Latch is shown in the following figure.



SR Latch

This circuit has two inputs S & R and two outputs Qt & Qt’. The upper NOR gate has two inputs R & complement of present state, Qt’ and produces next state, Q(t+1) when enable, E is ‘1’.

Similarly, the lower NOR gate has two inputs S & present state, Qt and produces complement of next state, Q(t+1) when enable, E is ‘1’.

We know that a 2-input NOR gate produces an output, which is the complement of another input when one of the input is ‘0’. Similarly, it produces ‘0’ output, when one of the input is ‘1’

If S = 1, then next state Q(t+1) will be equal to ‘1’ irrespective of present state, Qt values.

If R = 1, then next state Q(t+1) will be equal to ‘0’ irrespective of present state, Qt values.

At any time, only of those two inputs should be ‘1’. If both inputs are ‘1’, then the next state Q(t+1) value is undefined.

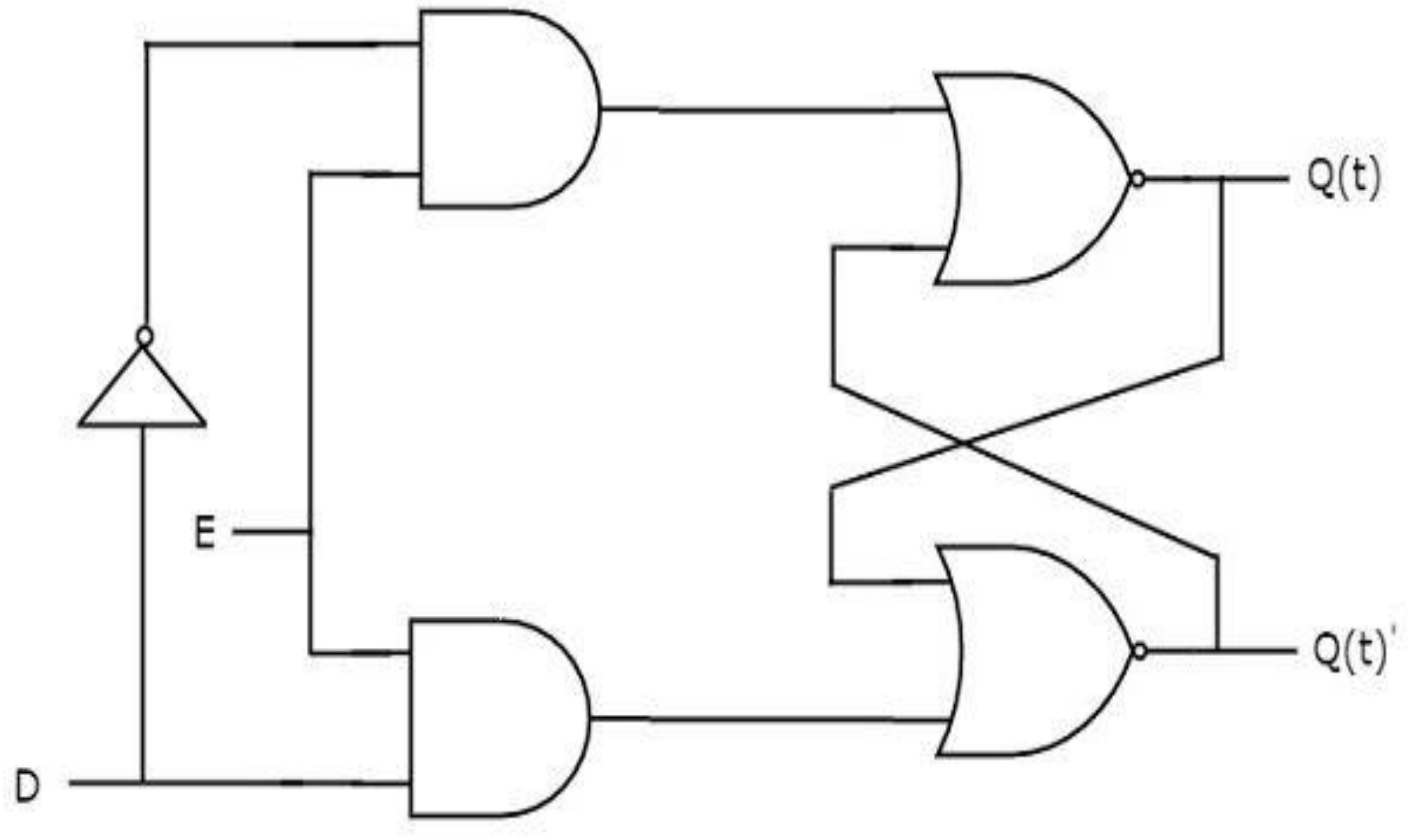
State table

S	R	Q(t+1)
0	0	Qt(hold)
0	1	0
1	0	1
1	1	-

Therefore, SR Latch performs three types of functions such as Hold, Set & Reset based on the input conditions.

D Latch

There is one drawback of SR Latch. That is the next state value can't be predicted when both the inputs S & R are one. So, we can overcome this difficulty by D Latch. It is also called as Delay Latch. The circuit diagram of D Latch is shown in the following figure.



D Latch

This circuit has single input D and two outputs Qt & Qt’. D Latch is obtained from SR Latch by placing an inverter between S & R inputs . That means we eliminated the combinations of S & R are of same value.

If D = 0 → S = 0 & R = 1, then next state Q(t+1) will be equal to ‘0’ irrespective of present state, Qt values. This is corresponding to the second row of SR Latch state table.

If D = 1 → S = 1 & R = 0, then next state Q(t+1) will be equal to ‘1’ irrespective of present state, Qt values. This is corresponding to the third row of SR Latch state table.

D	Q(t+1)
0	0
1	1

Therefore, D Latch Hold the information that is available on data input, D. That means the output of D Latch is sensitive to the changes in the input, D as long as the enable is High.

Flip Flop

A flip flop is an electronic circuit with two stable states that can be used to store binary data. The stored data can be changed by applying varying inputs. Flip-flops and latches are fundamental building blocks of digital electronics systems used in computers, communications, and many other types of systems. Both are used as data storage elements. It is the basic storage element in sequential logic.

The output is obtained in a sequential circuit from combinational circuit or flip-flop or both. The state of flip-flop changes at active state of clock pulses and remains unaffected when the clock pulse is not active. In particular, clocked flip flops serve as memory elements in synchronous sequential Circuits and unclocked flip-flops (i.e., latches) serve as memory elements in asynchronous sequential circuits.

The most common types of flip flop are:

1. JK flip flop
2. RS flip flop
3. D flip flop
4. T flip flop

Differences Between flip flop and latch

Flip-flop	Latch
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.
They are classified into asynchronous or synchronous flipflops.	There is no such classification in latches.
Flip-flop always have a clock signal	latch doesn’t have a clock signal
Flip-flop can be build from Latches	Latches can be build from gates
E.g. D Flip-flop, JK Flip-flop	E.g. SR Latch, D Latch

Basic flip-flop circuit

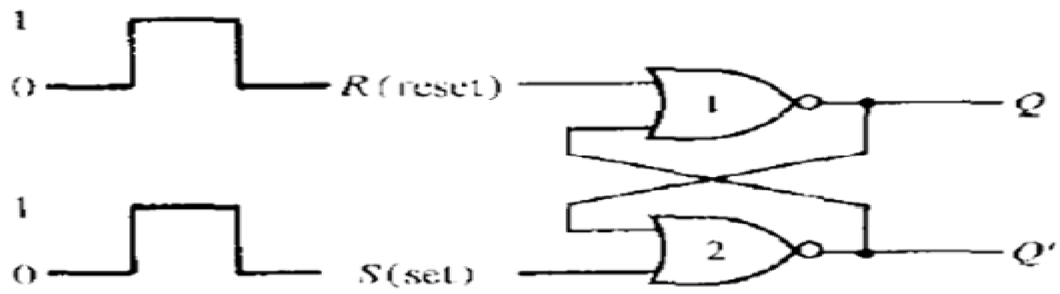
A basic Flip-Flop circuit can be constructed in two ways.

- Using two NOR gates
- Using two NAND gates

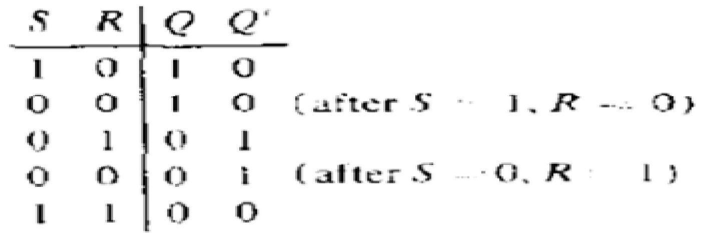
We know that a flip-flop circuit consists of two inputs set(S) and reset(R), two outputs Q and Q’. A cross coupled connection is given between output of one gate and the input of the other gate. Such type of cross coupled connection constitutes the feedback path for the flip-flops. These flip-flops are called direct coupled RS Flip flops (or) SR latch.

A flip-flop circuit can be constructed from two NAND gates or two NOR gates. These constructions are shown in the logic diagrams below. Each circuit forms a basic flip-flop upon which other more complicated types can be built. 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. Each flip-flop has two outputs, Q and Q', and two inputs, set and reset.

Basic flip-flop circuit



(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) Truth table

Fig: basic flip-flop circuit with NOR gates

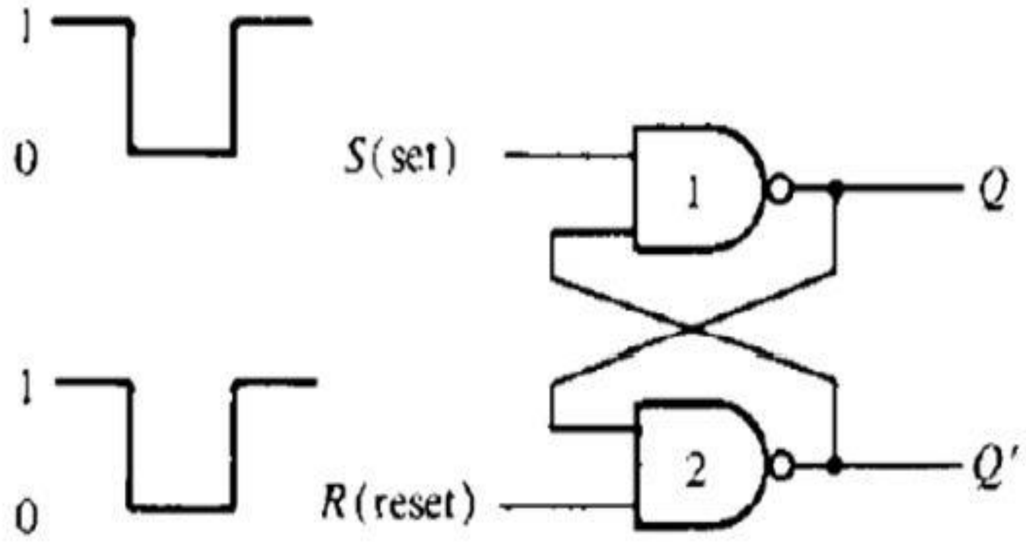
Output of a NOR gate is 0 if any input is 1, and that the output is 1 only when all inputs are 0.

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.

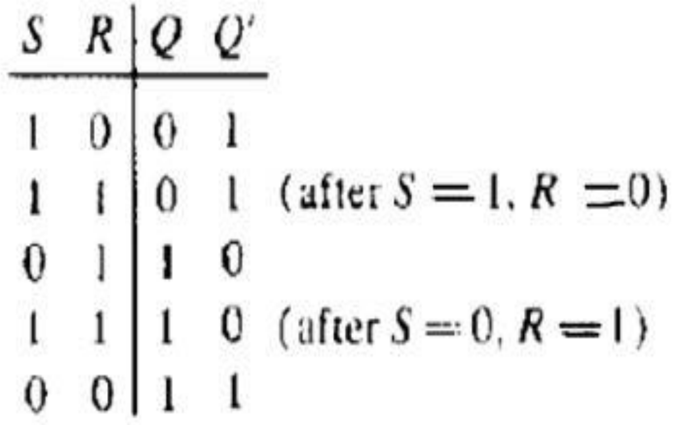
Similarly, 1 in the reset input changes output Q to 0 and Q' to 1. When the reset input returns to 0, the outputs do not change.

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. In normal operation, this condition must be avoided by making sure that 1's are not applied to both inputs simultaneously.

Basic flip-flop circuit



(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	

(b) Truth table

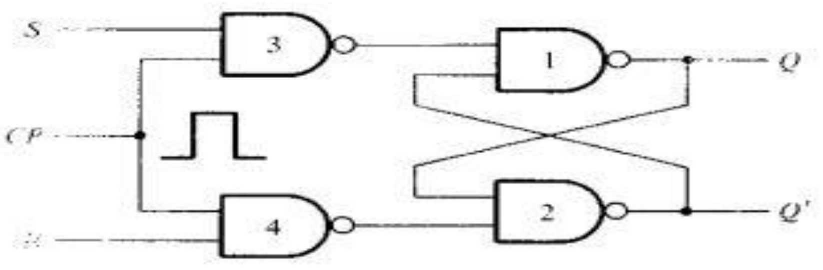
Fig: Basic flip-flop circuit with NAND gates

The operation of the basic flip-flop can be modified by providing an additional control input that determines when the state of the circuit is to be changed. This fact arises 4 common types of flip-flops and are discussed

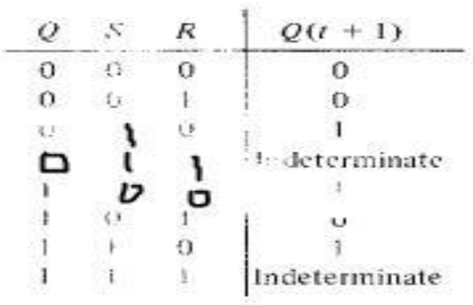
SR Flip Flop

The SR flip flop is a 1-bit memory bistable device having two inputs, i.e., SET and RESET. The SET input 'S' set the device or produce the output 1, and the RESET input 'R' reset the device or produce the output 0. The SET and RESET inputs are labeled as S and R, respectively.

The Basic SR Flip-flop

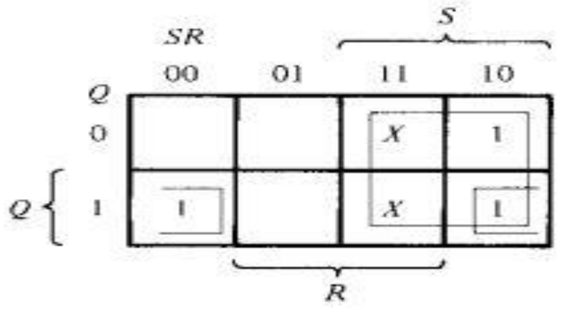


(a) Logic diagram



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

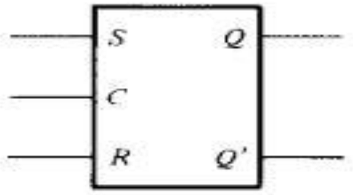
(b) Characteristic table



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

$SR = 0$

(c) Characteristic equation



(d) Graphic symbol

SR Flip Flop

The Set State

Consider the circuit shown above. If the input R is at logic level “0” (R = 0) and input S is at logic level “1” (S = 1), the NAND gate Y has at least one of its inputs at logic “0” therefore, its output Q’ must be at a logic level “1” (NAND Gate principles). Output Q’ is also fed back to input “A” and so both inputs to NAND gate X are at logic level “1”, and therefore its output Q must be at logic level “0”.

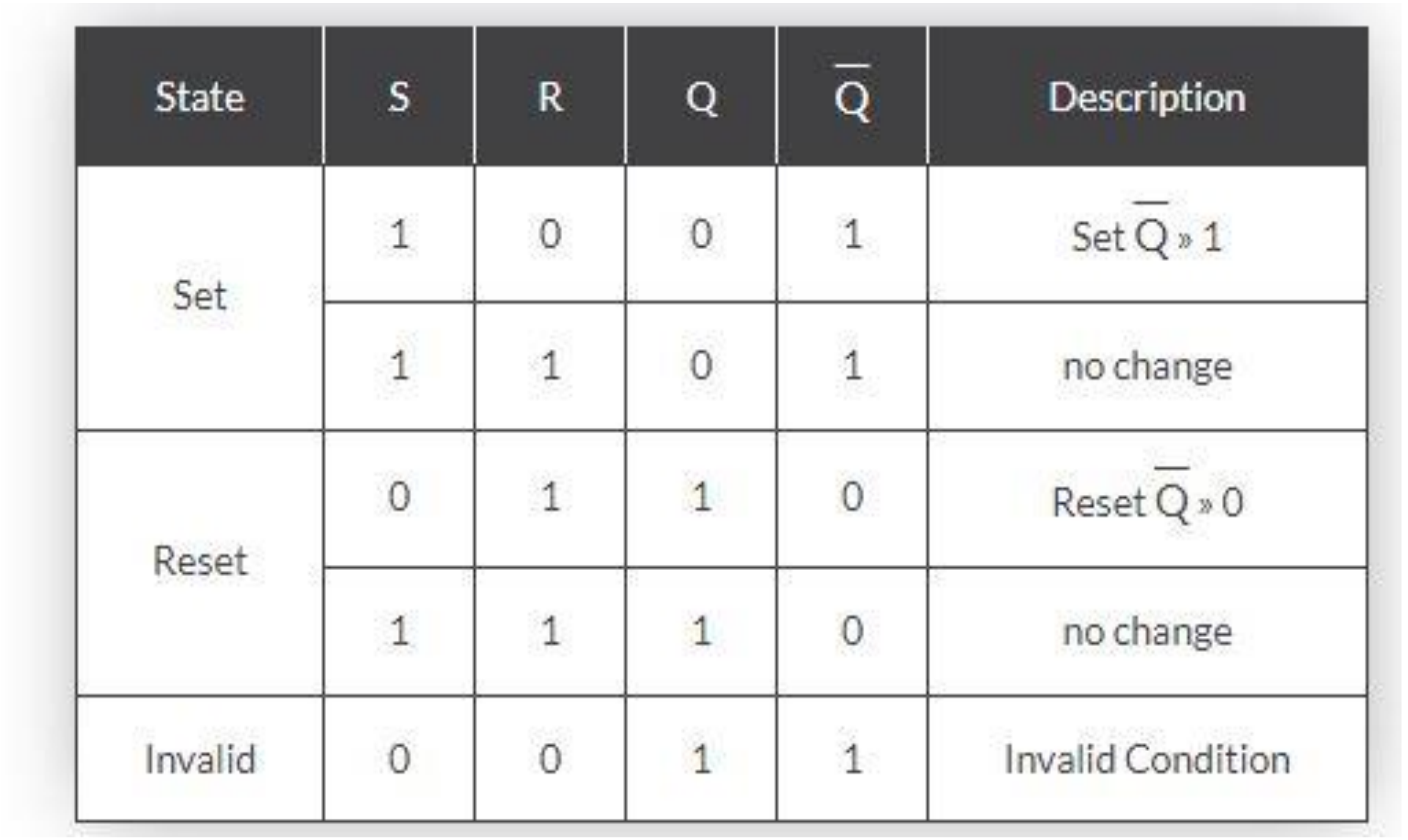
Again NAND gate principals. If the reset input R changes state, and goes HIGH to logic “1” with S remaining HIGH also at logic level “1”, NAND gate Y inputs are now R = “1” and B = “0”. Since one of its inputs is still at logic level “0” the output at Q’ still remains HIGH at logic level “1” and there is no change of state. Therefore, the flip-flop circuit is said to be “Latched” or “Set” with Q’ = “1” and Q = “0”.

Reset State

In this second stable state, Q’ is at logic level “0”, (not Q = “0”) its inverse output at Q is at logic level “1”, (Q = “1”), and is given by R = “1” and S = “0”.

As gate X has one of its inputs at logic “0” its output Q must equal logic level “1” (again NAND gate principles). Output Q is fed back to input “B”, so both inputs to NAND gate Y are at logic “1”, therefore, Q’ = “0”.If the set input, S now changes state to logic “1” with input R remaining at logic “1”, output Q’ still remains LOW at logic level “0” and there is no change of state. Therefore, the flip-flop circuits “Reset” state has also been latched and we can define this “set/reset” action in the following truth table.

SR Flip Flop



State	S	R	Q	$\overline{Q}$	Description
Set	1	0	0	1	Set $\overline{Q}$ » 1
	1	1	0	1	no change
Reset	0	1	1	0	Reset $\overline{Q}$ » 0
	1	1	1	0	no change
Invalid	0	0	1	1	Invalid Condition

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 shown in Fig. below. 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.

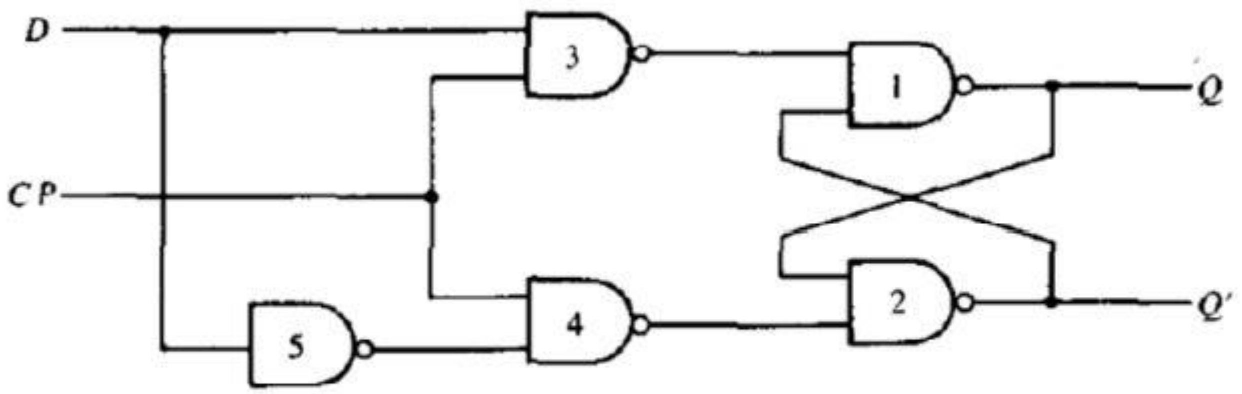
As long as CP is 0, the outputs of gates 3 and 4 are at the 1 level and the circuit cannot change state regardless of the value of D.

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



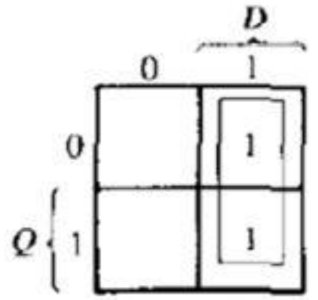
(a) Logic diagram

D Flip-Flop

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

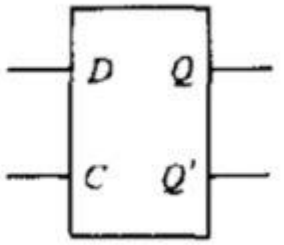
Q	*D*	*Q*(*t* + 1)
0	0	0
0	1	1
1	0	0
1	1	1

(b) Characteristic table



$Q(t+1) = D$

(c) Characteristic equation



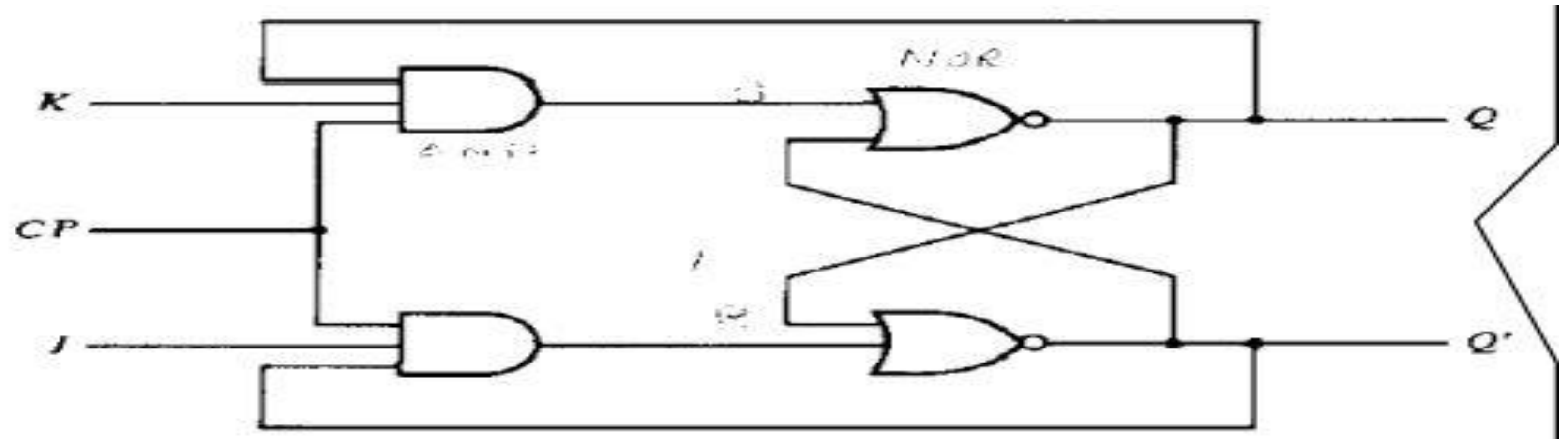
(d) Graphic symbol

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:

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. To avoid this undesirable operation, the clock pulse must have a time duration that is shorter than the propagation delay time of the flip-flop. This is a restrictive requirement of JK which is eliminated with a master-slave or edge-triggered construct

JK Flip-Flop



(a) Logic diagram

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

(b) Characteristic table

Q \ JK	00	01	11	10
0			1	1
1	1			1

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

(c) Characteristic equation

JK Flip-Flop

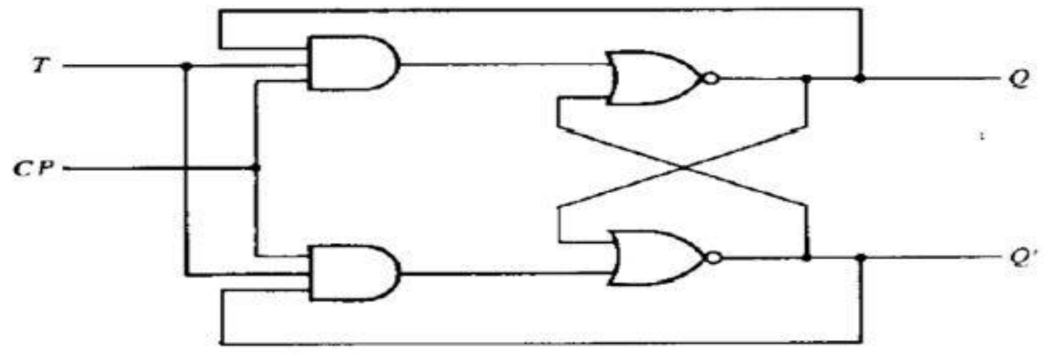
J	K	Q(t + 1)	
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	Complement

T Flip-Flop

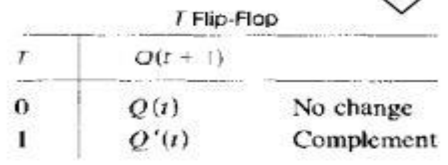
The T flip-flop is a single-input version of the JK flip-flop and is obtained from the JK flip-flop when both inputs are tied together. The designation T comes from the ability of the flip-flop to "toggle," or complement, its state. Regardless of the present state, the flip-flop complements its output 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.

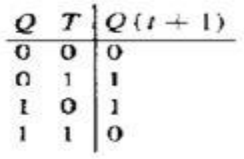


(a) Logic diagram



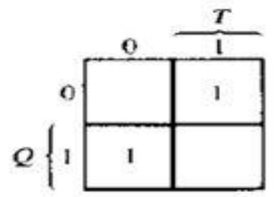
T Flip-Flop

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



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

(b) Characteristic table

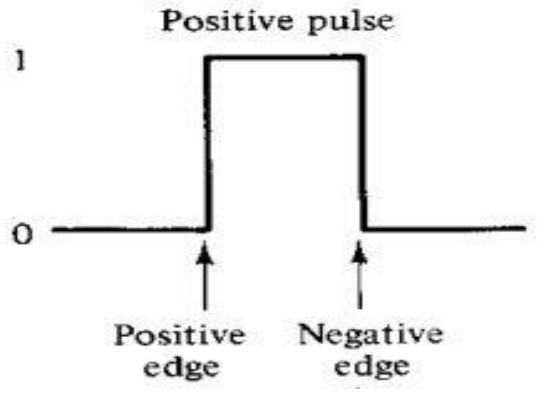


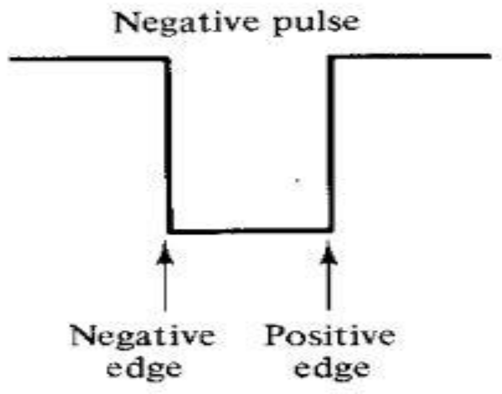
$Q(t+1) = TQ' + T'Q$

(c) Characteristic equation

Triggering of Flip-Flops

The state of a flip-flop is switched by a momentary change in the input signal. This momentary change is called a trigger and the transition it causes is said to trigger the flip-flop. 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 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. The pulse goes through two signal transitions: from 0 to 1 and the return from 1 to 0. As shown in Fig. below, the positive transition is defined as the positive edge and the negative transition as the negative edge.



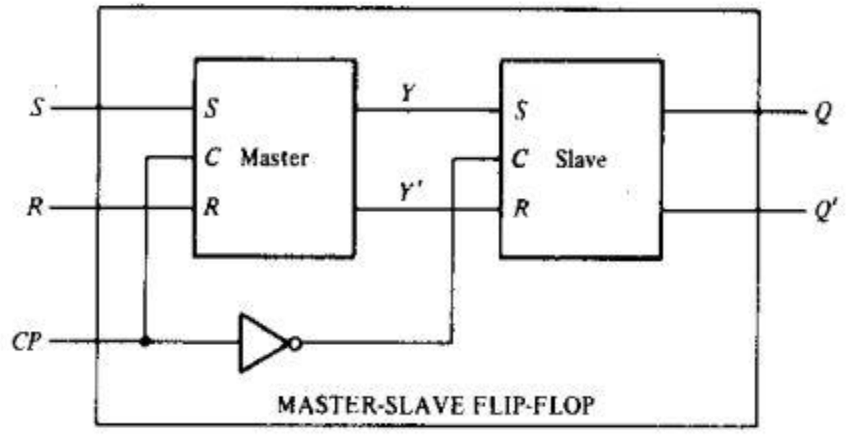


Master-slave Flip-Flop

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.



JK Master-slave Flip-Flop

Master-slave JK flip-flop constructed with NAND gates is shown in Fig. below. It consists of two flip flops; gates 1 through 4 form the master flip-flop, and gates 5 through 8 form 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.

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 gate 9.
- 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

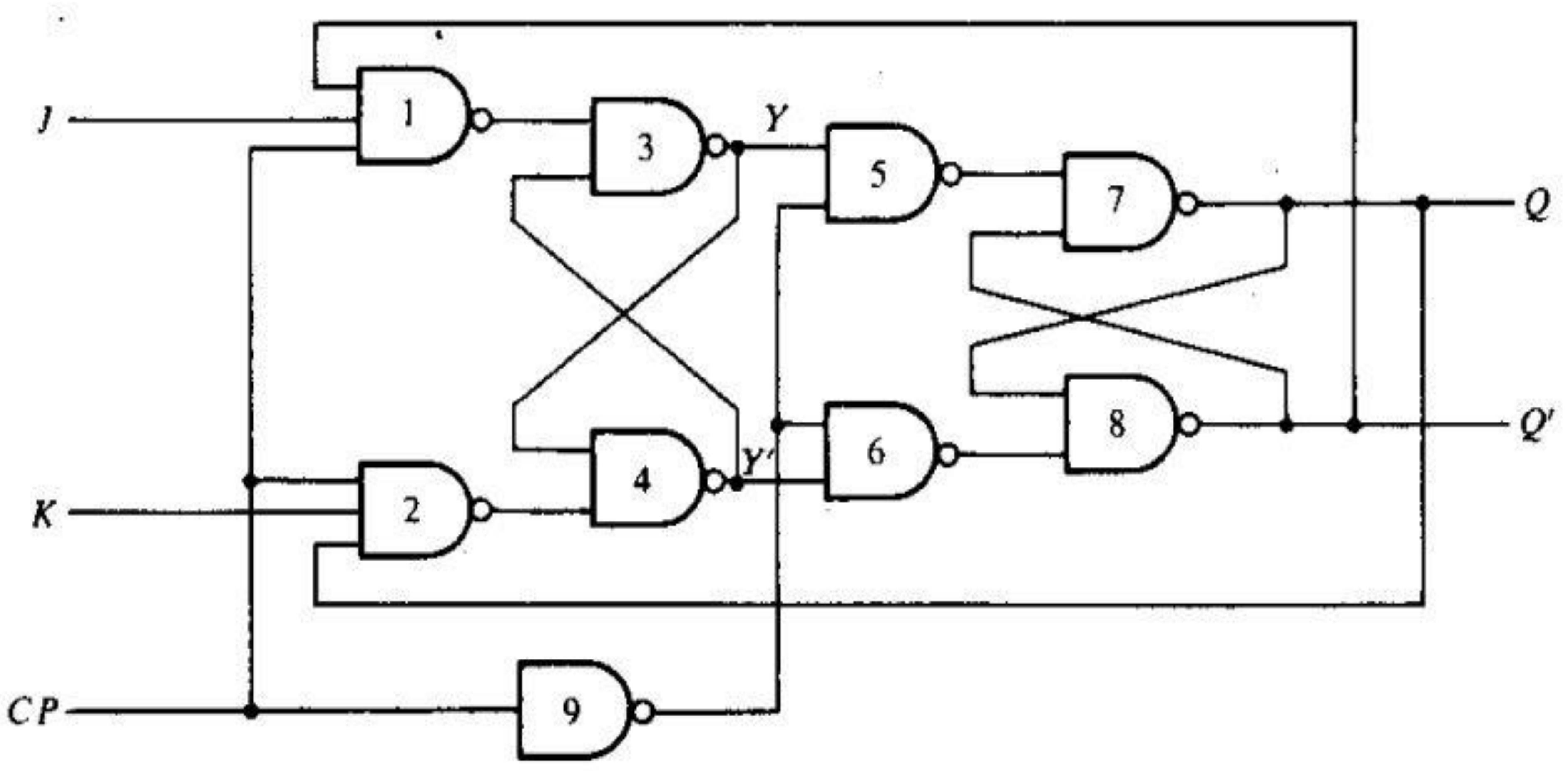


FIGURE 6-11

Clocked master–slave *JK* flip-flop

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.

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

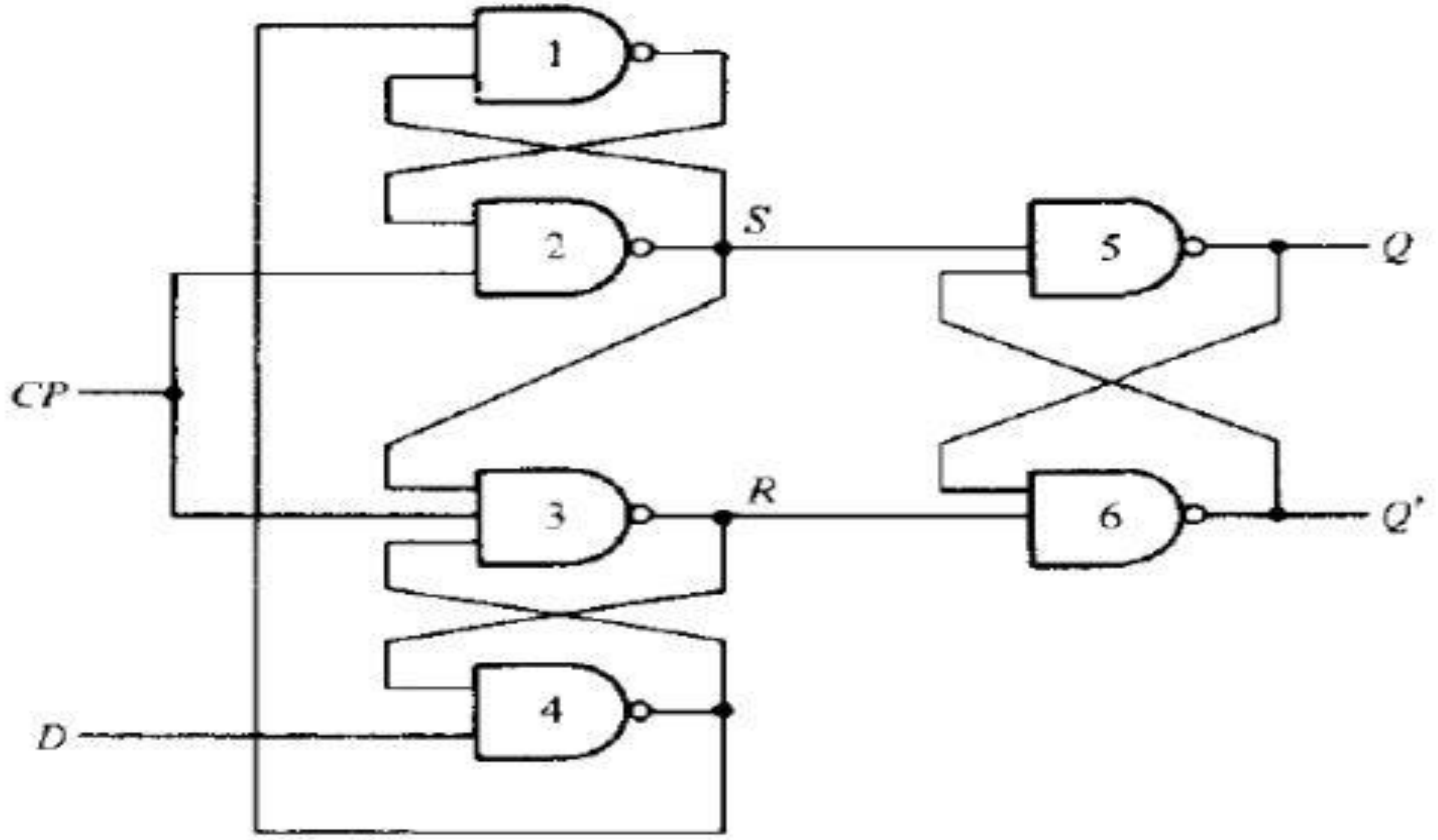
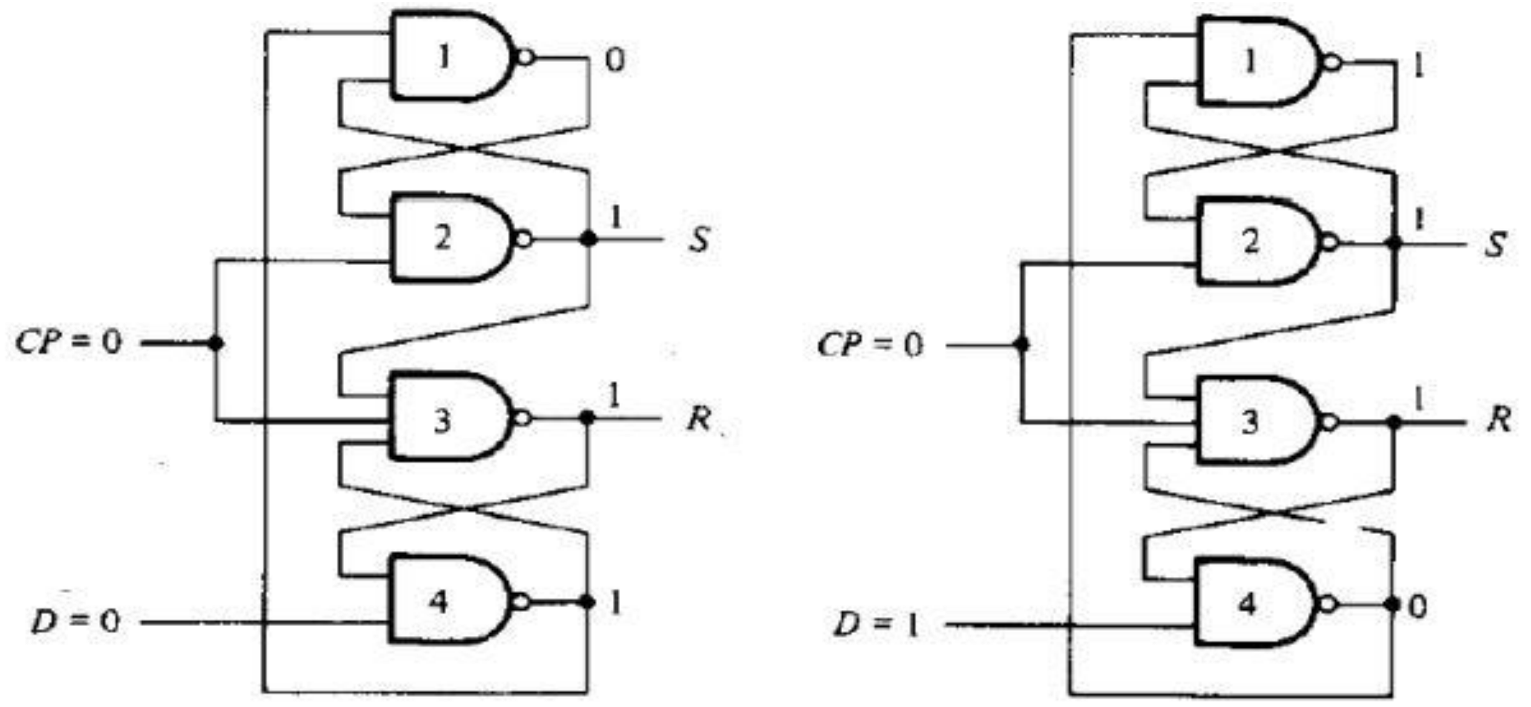


Fig: D-type positive-edge-triggered flip-flop

Edge-Triggered Flip-Flop

Operation:

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.

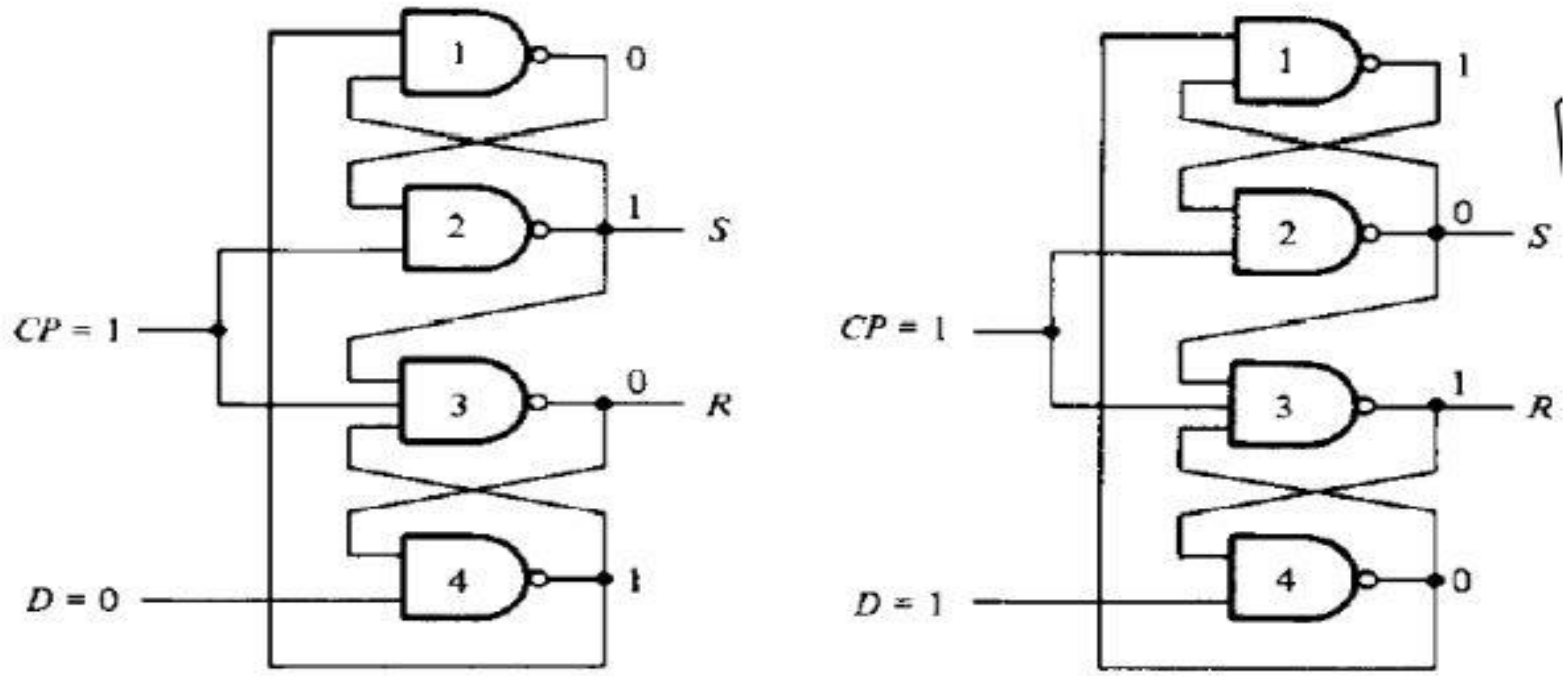


(a) With $CP = 0$

Edge-Triggered Flip-Flop

Operations:

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.

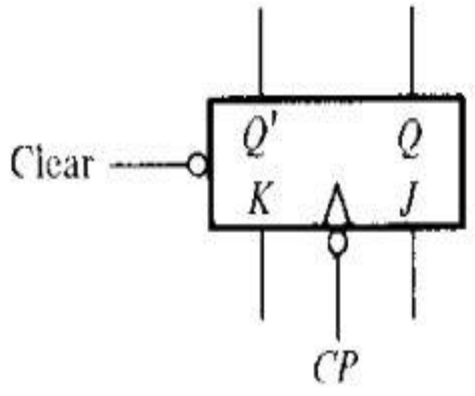
Example: After power is turned on in a digital system, the states of its flip-flops are indeterminate. A clear switch clears all the flip-flops to an initial cleared state and a start switch begins the system's clocked operation. The clear switch must clear all flip-flops asynchronously without the need for a pulse.

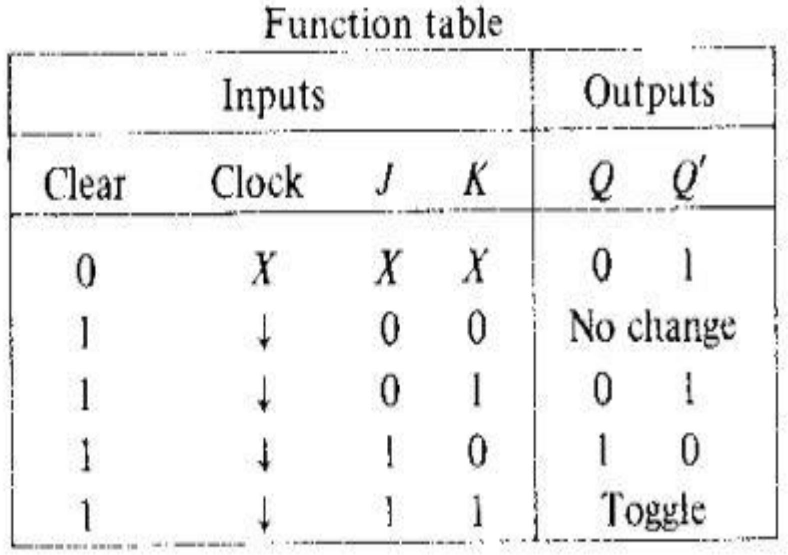
The direct-clear input has a small circle to indicate that, normally, this input must be maintained at 1. If the clear input is maintained at 0, the flip-flop remains cleared, regardless of the other inputs or the clock pulse.

Direct Inputs

The function table specifies the circuit operation. The X's are don't-care conditions, which indicate that a 0 in the direct-clear input disables all other inputs. Only when the clear input is 1 would a negative transition of the clock have an effect on the outputs.

The outputs do not change if J = K = O. The flip-flop toggles, or complements, when J = K = 1. Some flip-flops may also have a direct-preset input, which sets the output Q to 1(and Q' to 0) asynchronously.





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	

State diagram

The state diagram is the pictorial representation of the behavior of sequential circuits. It clearly shows the transition of states from the present state to the next state and output for a corresponding input.

In this diagram, each present state is represented inside a circle.

The transition from the present state to the next state is represented by a directed line connecting the circles.

If the directed line connects the circle itself, which indicates that there is no change in the state(the next state is the same as the present state).

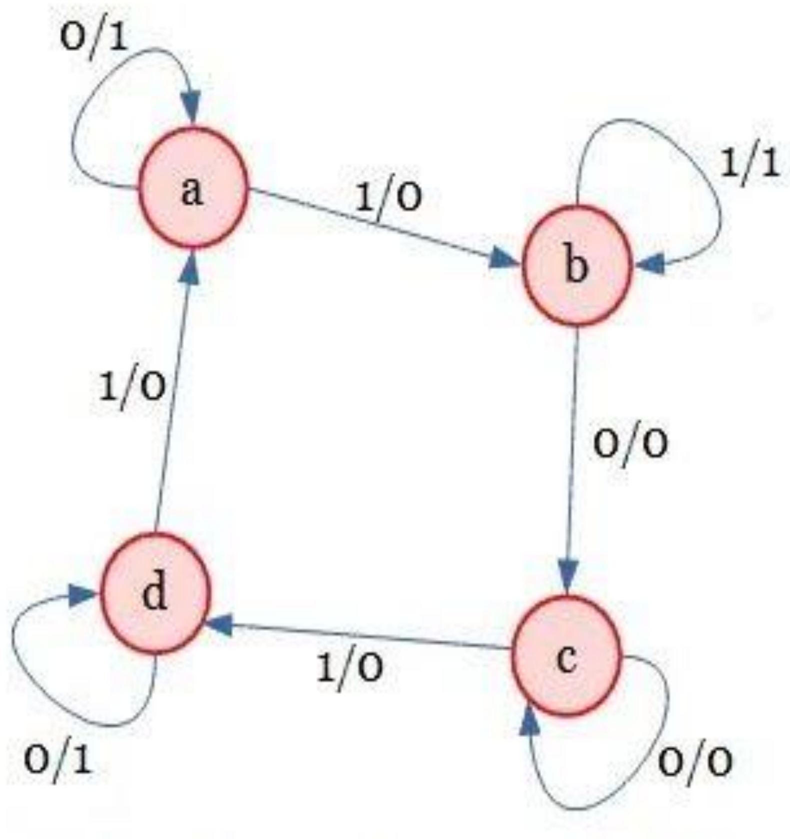
For mealy model, the directed line is labeled with binary numbers separated with ‘/’, as shown in the below diagram.

The input value, which causes the transition to occur is labeled first ‘1/’. The output produced for the corresponding input is labeled second ‘/0’.

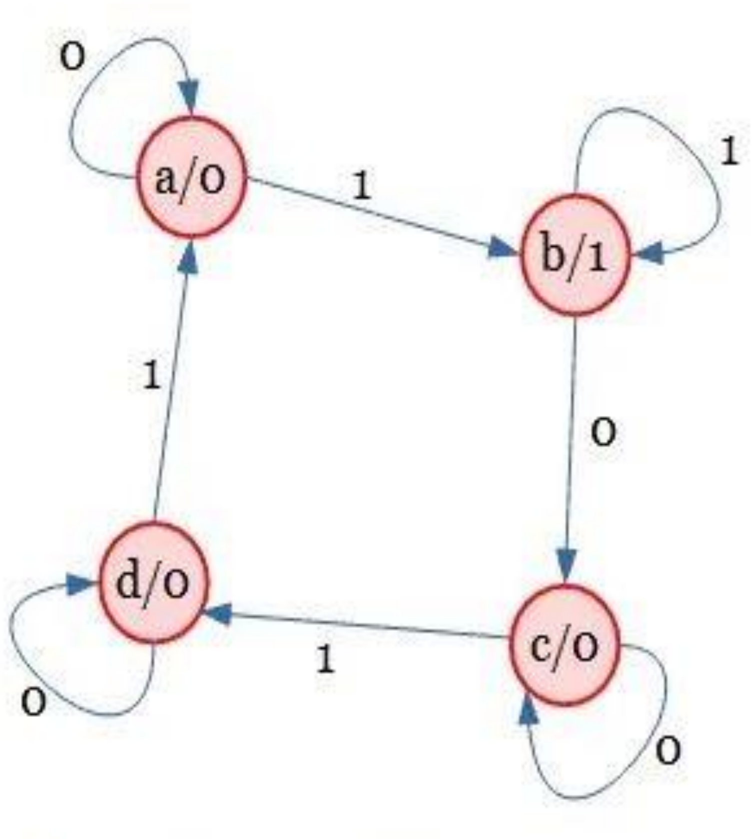
For Moore circuit, the directed lines are labeled with only one binary number. It is nothing but the input value which causes the transition.

The output value is indicated inside the circle below the present state. It is because, in Moore model, the output depends on the present state but not on the input.

State diagram



State diagram for Mealy Model



State diagram for Moore Model

State table

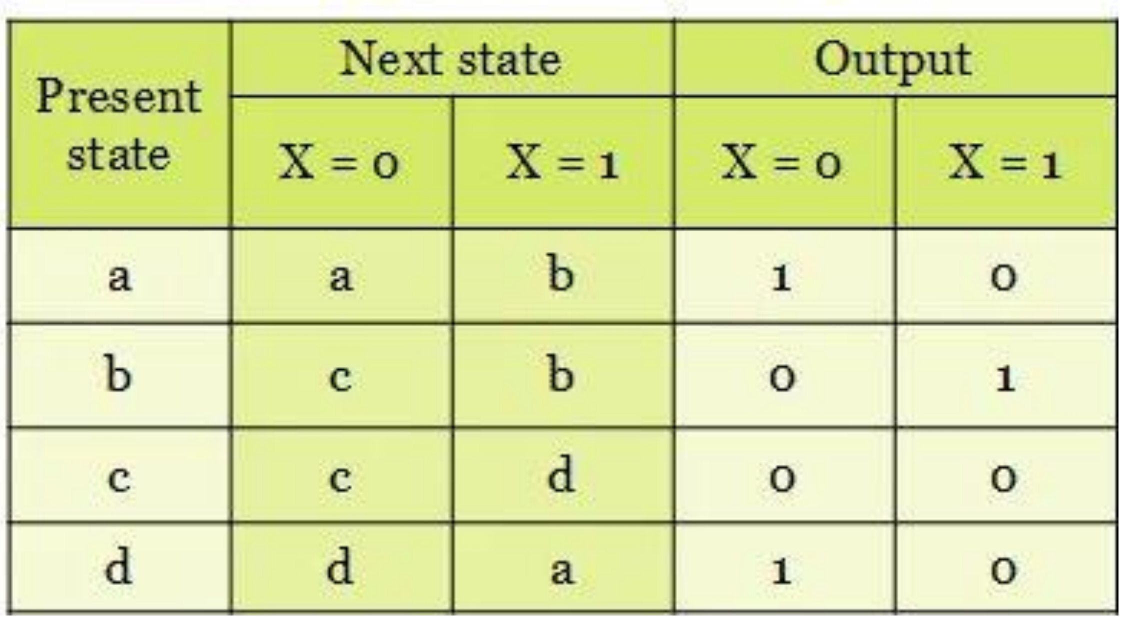
The information contained in the state diagram is transformed into a table called a state table or state synthesis table. Although the state diagram describes the behavior of the sequential circuit, in order to implement it in the circuit, it has to be transformed into the tabular form.

The below table shows the state table for Mealy state machine model. As you can see, it has the present state, next state and output. The present state is the state before the occurrence of the clock pulse.

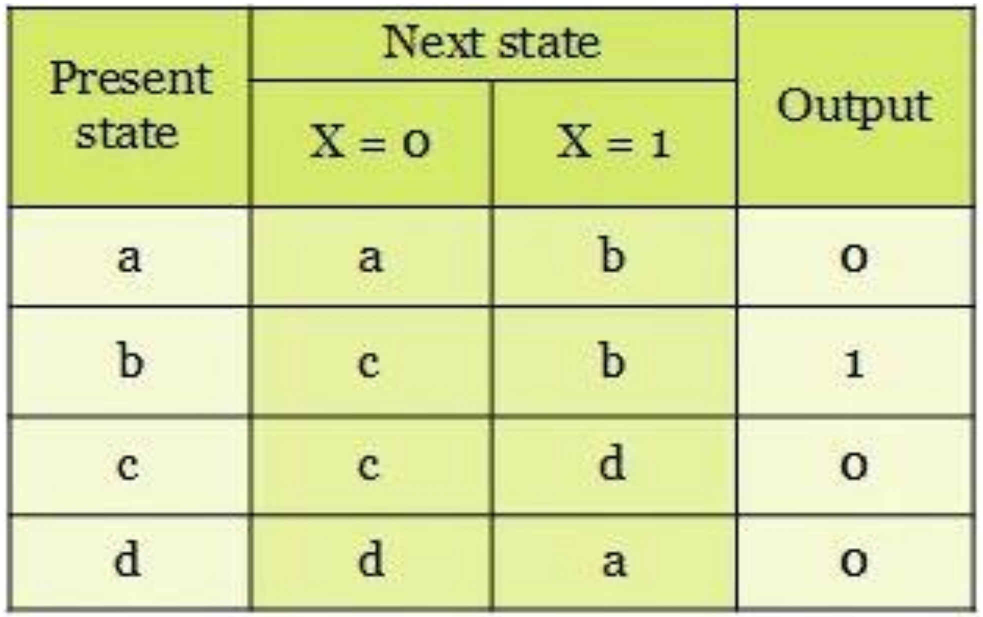
After the application of the clock pulse, depending on the input(X = 0 or 1), the state changes. It is indicated in the ‘next state’ column. The output produced for each input is represented in the last column

The table shown below is the state table for Moore state machine model. Since, in Moore state machine model, the output depends only on the present state, the last column has only output.

State table



Present state	Next state		Output	
	X = 0	X = 1	X = 0	X = 1
a	a	b	1	0
b	c	b	0	1
c	c	d	0	0
d	d	a	1	0



Present state	Next state		Output
	X = 0	X = 1	
a	a	b	0
b	c	b	1
c	c	d	0
d	d	a	0

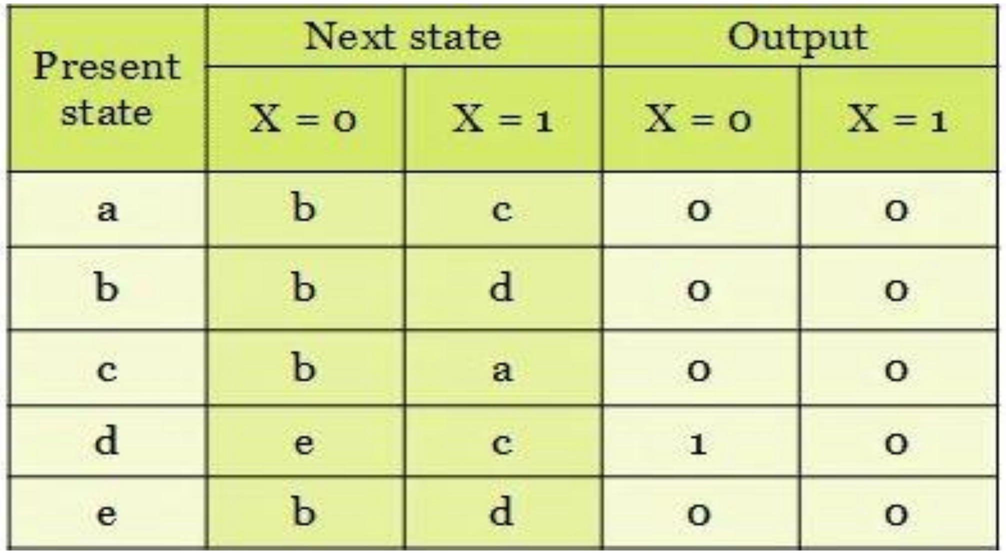
State reduction

While designing a sequential circuit, it is very important to remove the redundant states. The removal of redundant states will reduce the number of flip flops and logic gates, thereby reducing the cost and size of the sequential circuit.

When two states are said to be redundant?

The two states are said to be redundant if the output and the next state produced for each and every input are the same. In that case, one of the redundant states can be removed without altering the input-output relationship. This method is called the state elimination method.

Determine the reduced state table for the given state table.



Present state	Next state		Output	
	X = 0	X = 1	X = 0	X = 1
a	b	c	0	0
b	b	d	0	0
c	b	a	0	0
d	e	c	1	0
e	b	d	0	0

State reduction

The given table contains the present state, next state and output produced for inputs X = 0 and 1. To find the reduced state table, the first step is to find the redundant/equivalent states from the given state table.

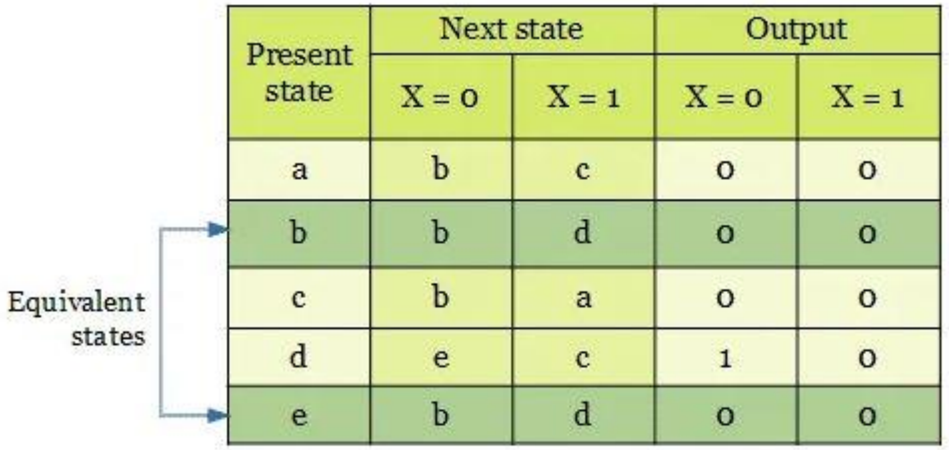
As explained above, any two states are said to be equivalent, if their next state and output are the same. In order to check that, compare each present state with the other.

First, consider the present state ‘a’, compare its next state and output with the other present states one by one. In this comparison, none of the present states is the same as the present state ‘a’.

Now, consider the next present state ‘b’ and compare it with other present states. While doing so, you can find the next state and the output of the present state ‘e’ is the same as that of ‘b’. They are marked as equivalent states as shown below.

Similarly, consider the other present states and compare them with other states for redundancy.

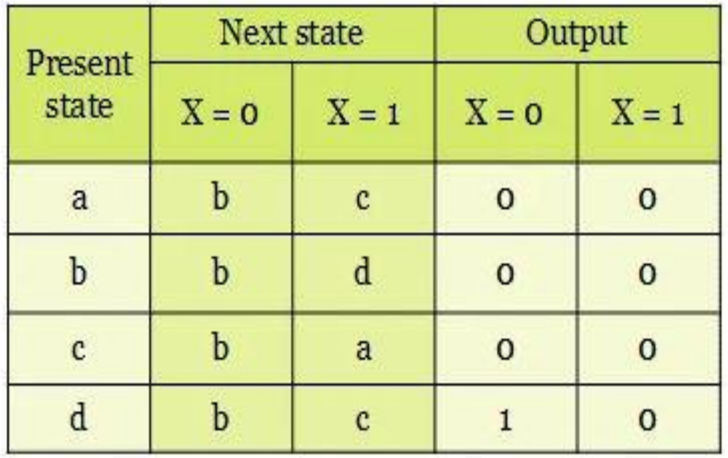
State reduction


Equivalent states

Present state	Next state		Output	
	X = 0	X = 1	X = 0	X = 1
a	b	c	0	0
b	b	d	0	0
c	b	a	0	0
d	e	c	1	0
e	b	d	0	0

The next step is to replace the redundant states with the equivalent state. Here we have found, states b and e are redundant. Replace e by b and remove the state e.

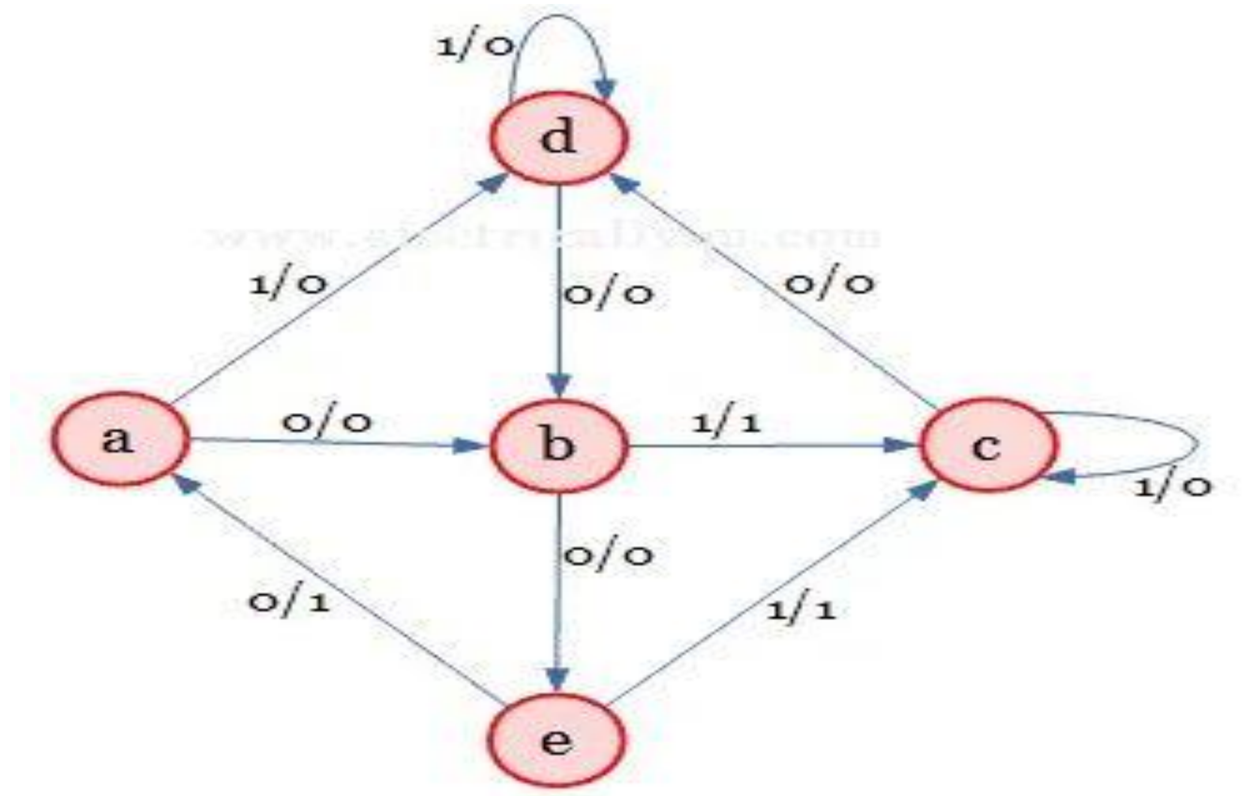
Now, there are no equivalent states and so the reduced state table will become as follows.



Present state	Next state		Output	
	X = 0	X = 1	X = 0	X = 1
a	b	c	0	0
b	b	d	0	0
c	b	a	0	0
d	b	c	1	0

State reduction

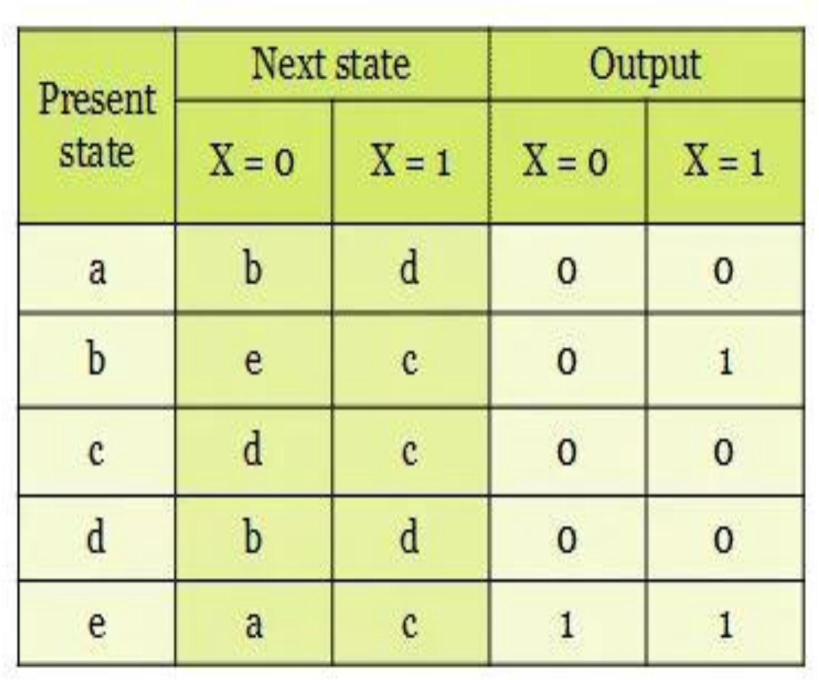
Determine the reduced state diagram for the given state diagram.



State Reduction

To construct the reduced state diagram, first, build the state table for the given state diagram, find the equivalent states, remove the redundant state, draw the reduced state table and finally construct the state diagram.

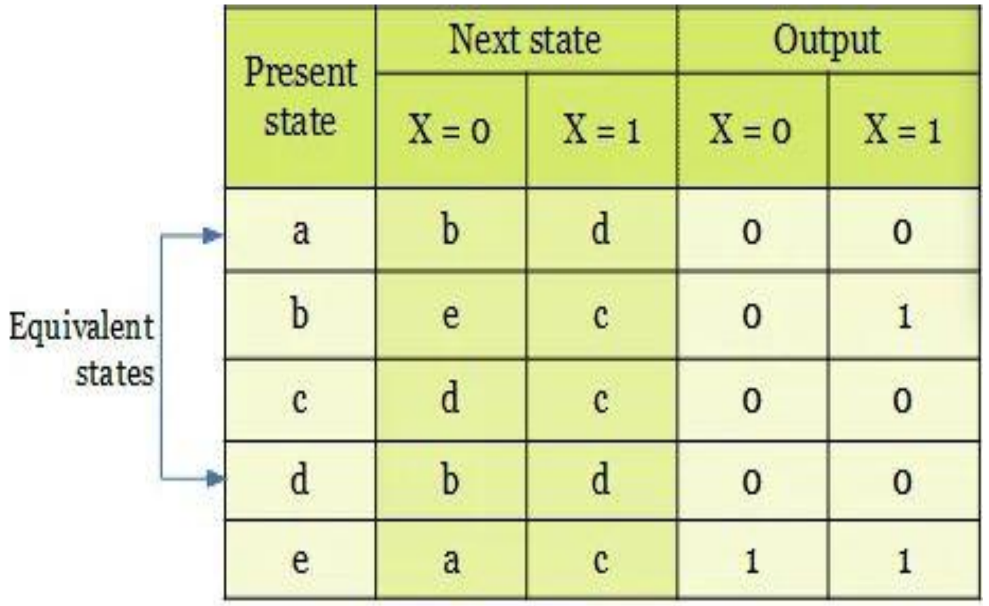
First, the information in the state diagram is transferred into the state table as shown below.



Present state	Next state		Output	
	X = 0	X = 1	X = 0	X = 1
a	b	d	0	0
b	e	c	0	1
c	d	c	0	0
d	b	d	0	0
e	a	c	1	1

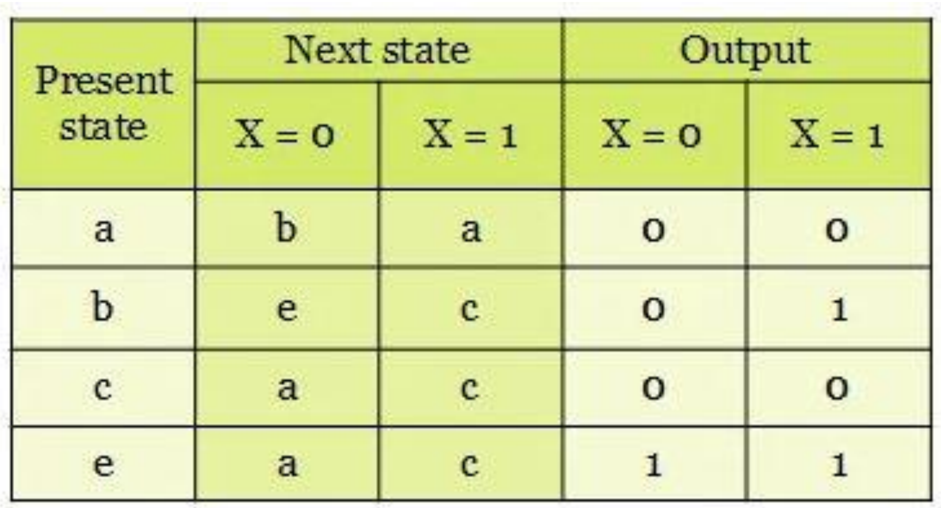
Next, find the equivalent states. From the above table, you can observe that the next state and output of the present states ‘a’ and ‘d’ is found to be the same. It is shown in the below table.

State Reduction


Equivalent states

Present state	Next state		Output	
	X = 0	X = 1	X = 0	X = 1
a	b	d	0	0
b	e	c	0	1
c	d	c	0	0
d	b	d	0	0
e	a	c	1	1

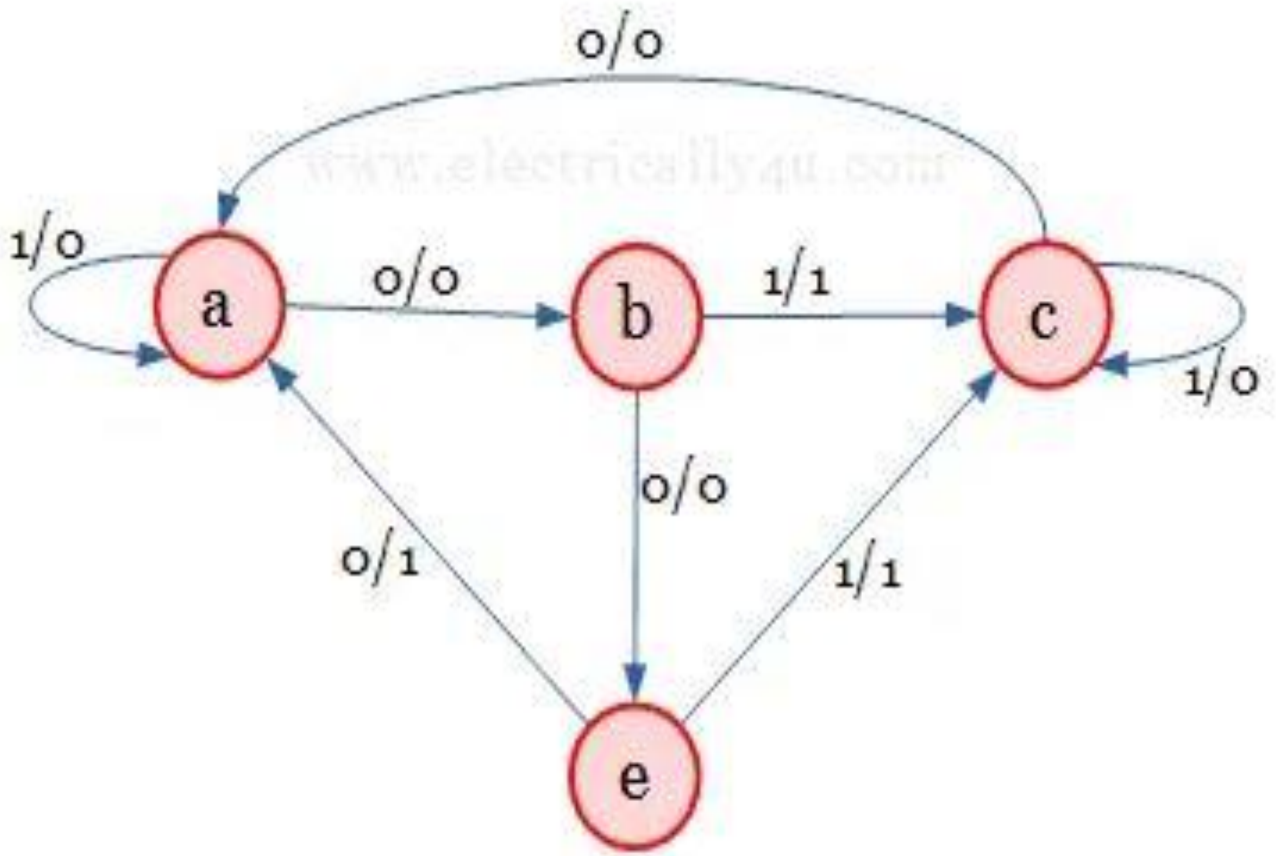
Thus ‘a’ and ‘d’ are found as equivalent states. So, replace ‘d’ by ‘a’ and remove ‘d’. Now, the reduced state table will become as below.



Present state	Next state		Output	
	X = 0	X = 1	X = 0	X = 1
a	b	a	0	0
b	e	c	0	1
c	a	c	0	0
e	a	c	1	1

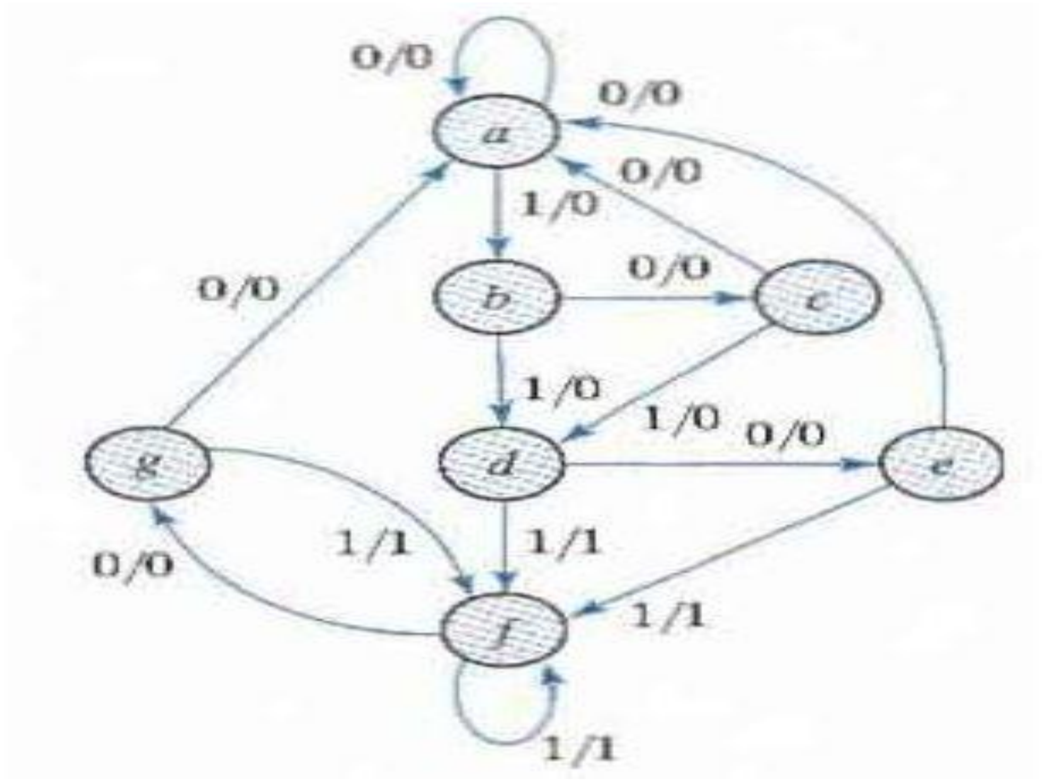
State Reduction

The state diagram is constructed for the reduced state table as shown below.



State Reduction

If possible reduce the state in the following state diagram.



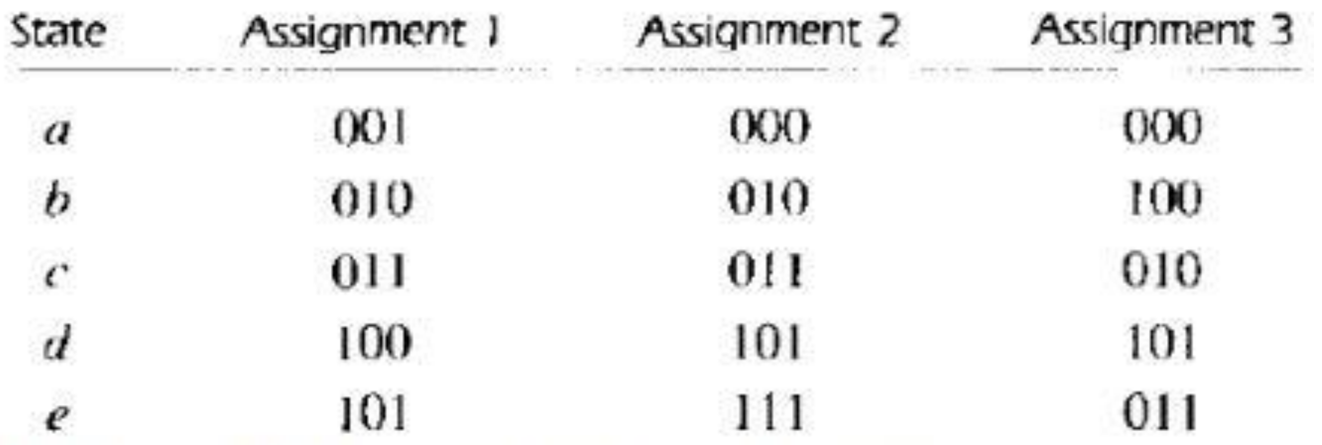
State Assignment

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.

Consider a example shown in state reduction, 3 examples of possible binary assignments are shown in Table below for the five states of the reduced table. Assignment 1 is a straight binary assignment for the sequence of states from a through e. The other two assignments are chosen arbitrarily. In fact, there are 140 different distinct assignments for this circuit.

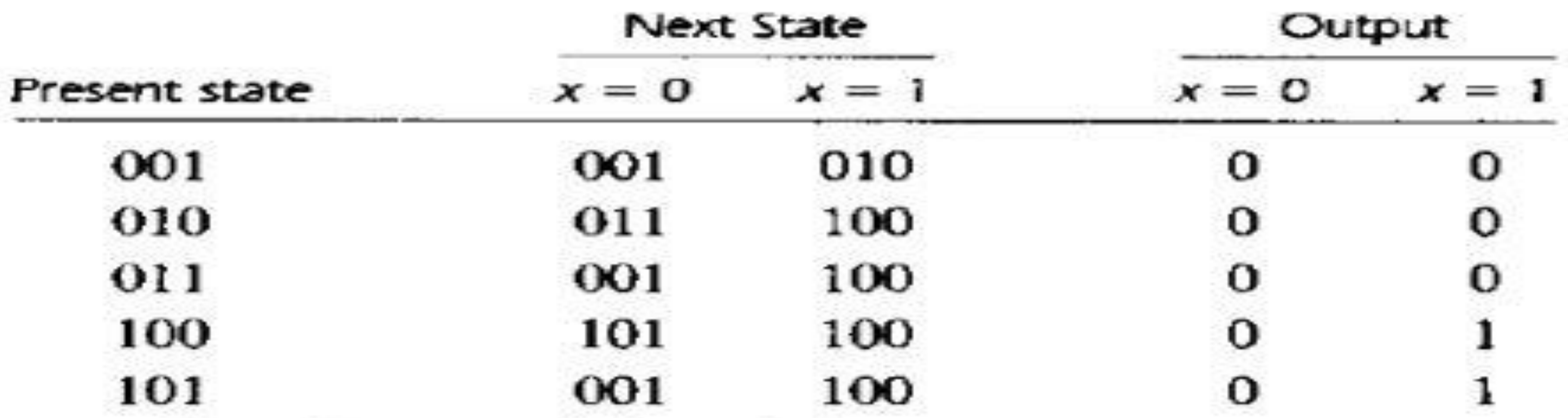
The binary form of the state table is used to derive the combinational-circuit part of the sequential circuit. The complexity of the combinational circuit obtained depends on the binary state assignment chosen.

State Assignment



State	Assignment 1	Assignment 2	Assignment 3
a	001	000	000
b	010	010	100
c	011	011	010
d	100	101	101
e	101	111	011

Fig: 3 possible binary sate assignments



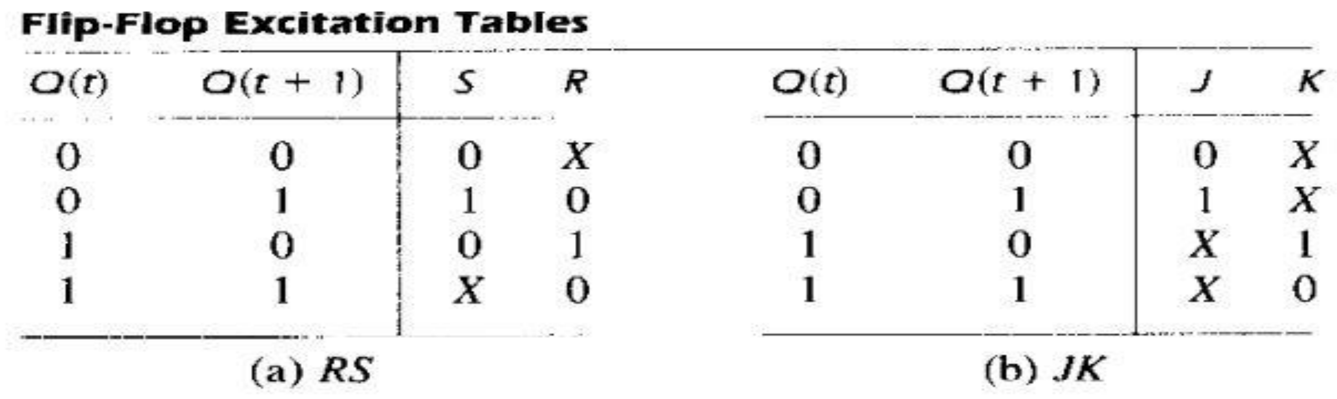
Present state	Next State $x = 0$	Next State $x = 1$	Output $x = 0$	Output $x = 1$
001	001	010	0	0
010	011	100	0	0
011	001	100	0	0
100	101	100	0	1
101	001	100	0	1

Fig: Reduced state table with binary assignment 1

Excitation Tables

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

The required input conditions for each of the four transitions are derived from the information available in the characteristic table. The symbol X in the tables represents a don't-care condition, i.e., it does not matter whether the input is 1 or 0



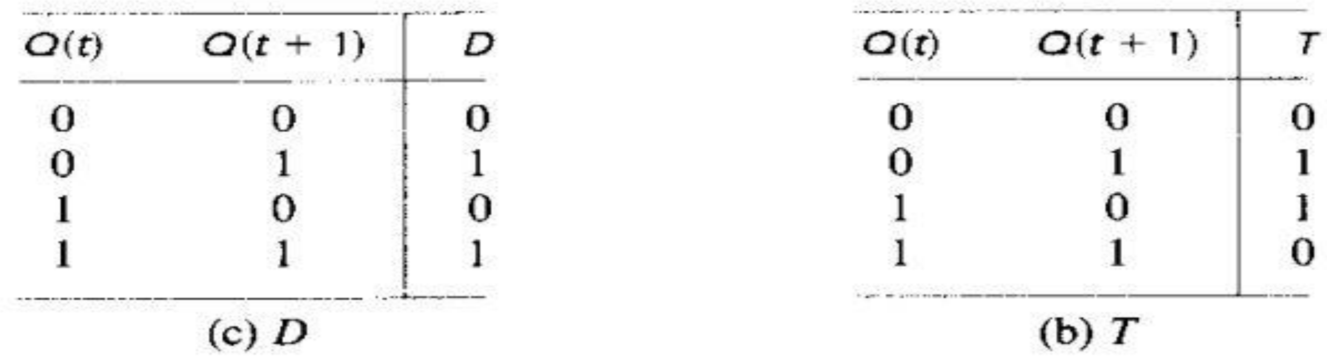
Flip-Flop Excitation Tables

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

(a) *RS*

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

(b) *JK*



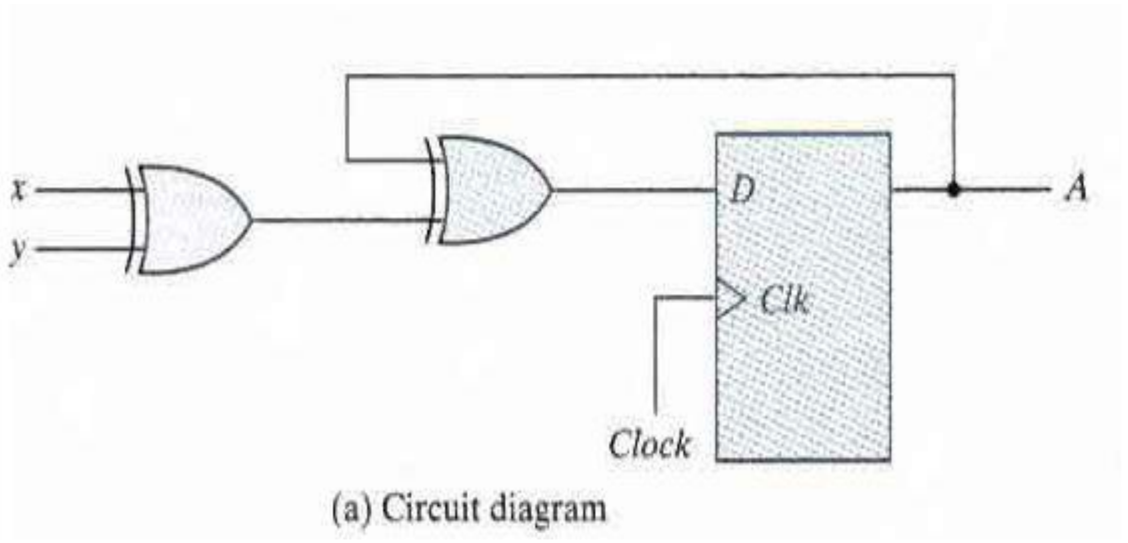
$Q(t)$	$Q(t+1)$	D
0	0	0
0	1	1
1	0	0
1	1	1

(c) *D*

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

(b) *T*

Analysis of sequential circuit



(a) Circuit diagram

The input equation of a D Flip-flop is given by DA = A ⊕ x ⊕ y. DA means a D Flip-flop with output A.

The x and y variables are the inputs to the circuit. No output equations are given, which implies that the output comes from the output of the flip-flop.

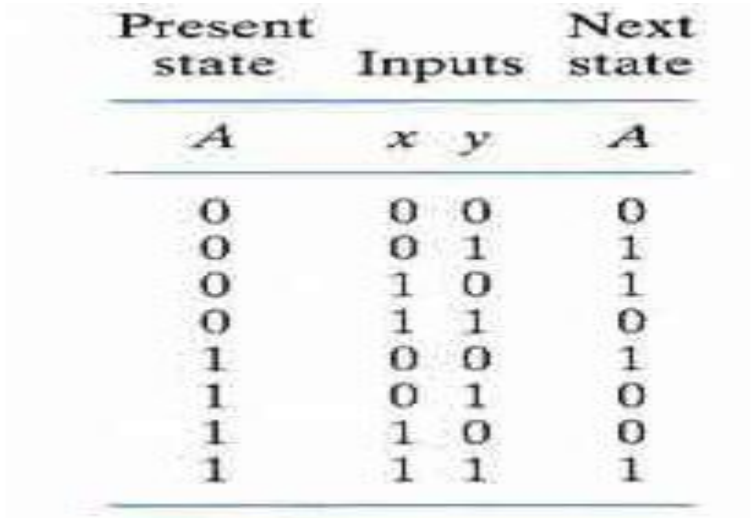
The state table has one column for the present state of flip-flop 'A' two columns for the two inputs, and one column for the next state of A.

The next-state values are obtained from the state equation A(t + 1) = A ⊕ x ⊕ y.

The expression specifies an odd function and is equal to 1 when only one variable is 1 or when all three variables are 1.

Analysis of sequential circuit

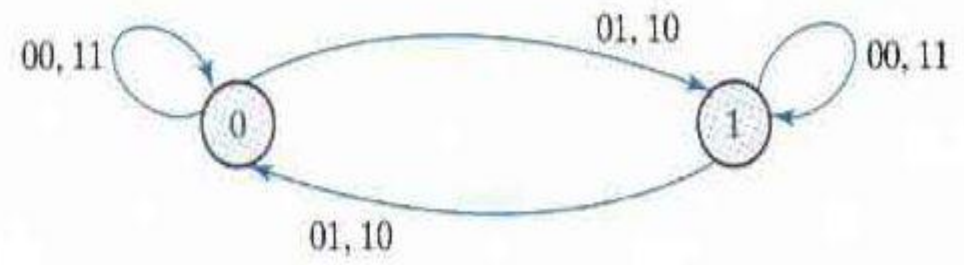
State table based on the above expression



Present state	Inputs		Next state
A	x	y	A
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

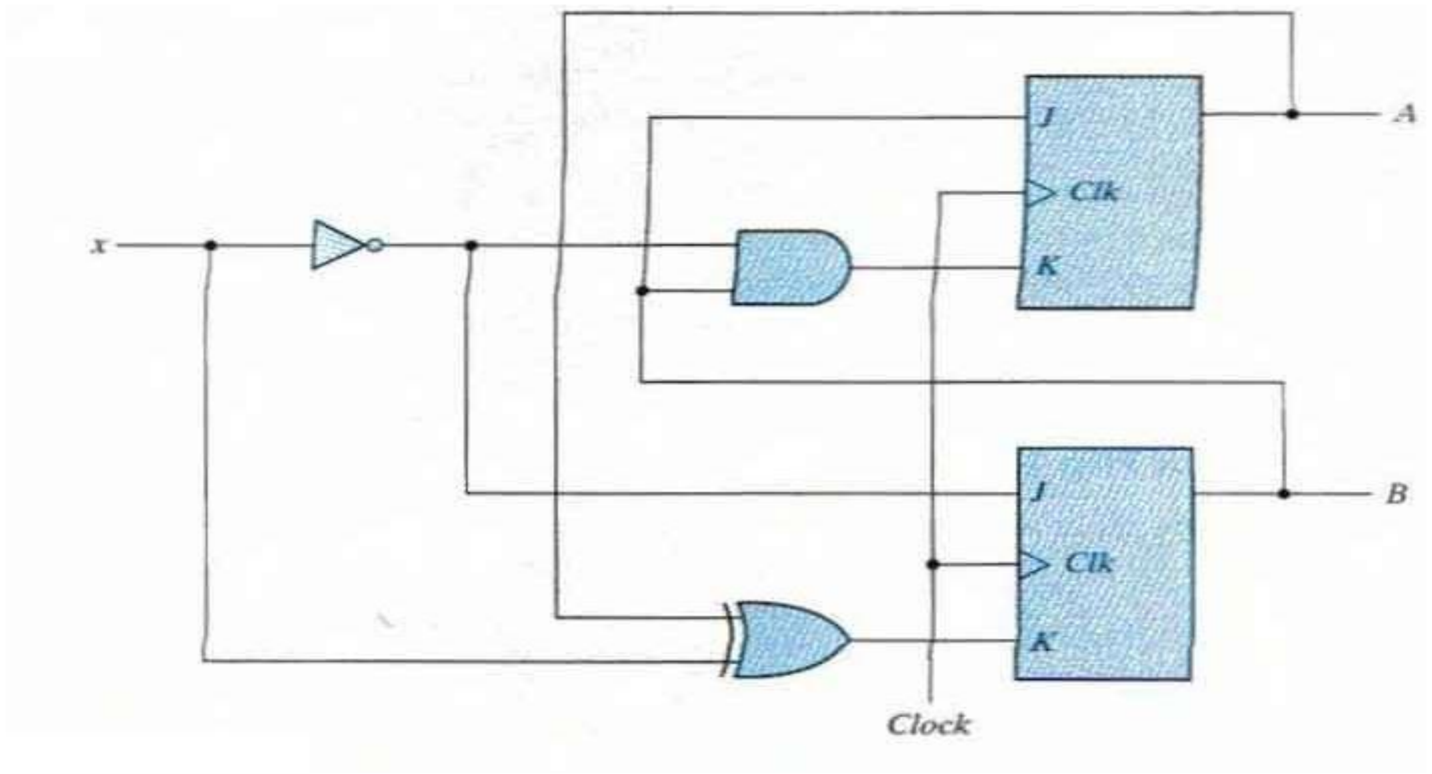
(b) State table

Now, draw state diagram



(c) State diagram

Analysis with JK Flip-Flops



The circuit can be specified by the flip-flop input equations:

JA = B; KA = Bx'

JB = x'; KB = A'x + Ax' = A $\oplus$ x

Analysis with JK Flip-Flops

The next state of each flip-flop is evaluated from the corresponding J and K inputs and the characteristic table of the JK flip-flop listed as:

When J = 1 and K = 0 the next state is 1

When J = 0 and K = 1 the next state is 0

When J = 0 and K = 0 there is no change of state and the next-state value is the same as that of the present state.

When J = K = 1, the next-state bit is the complement of the present-state bit.

The characteristic equations for the flip-flops are

$A(t + 1) = JA' + K'A$

$B(t + 1) = JB' + K'B$

This gives us the state equation of A by substituting the values of JA, KA

$A(t + 1) = BA' + (Bx')'A = A'B + AB' + Ax$

The state equation provides the bit values for the column headed "Next State" for A in the state table. Similarly, the state equation for flip-flop B can be derived from the characteristic equation by substituting the values of JB and KB.:

$B(t + 1) = x'B' + (A \oplus x)'B = B'x' + ABx + A'Bx'$

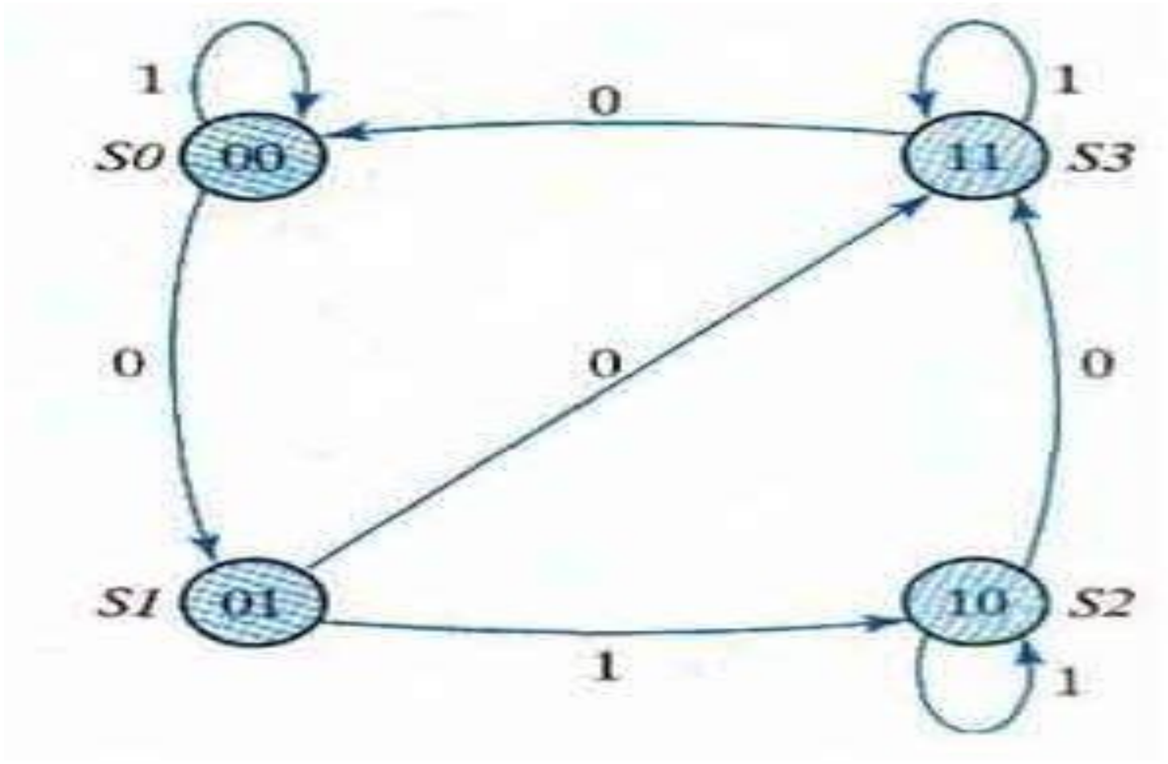
Analysis with JK Flip-Flops

State Table

Present state		Input(x)	Flip flop inputs				Next state	
A	B	X	J_A	K_A	J_B	K_B	A(t+1)	B(t+1)
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

Analysis with JK Flip-Flops

Now, draw state diagram using state table



Analysis with T Flip-Flops

Lets analyze input and output

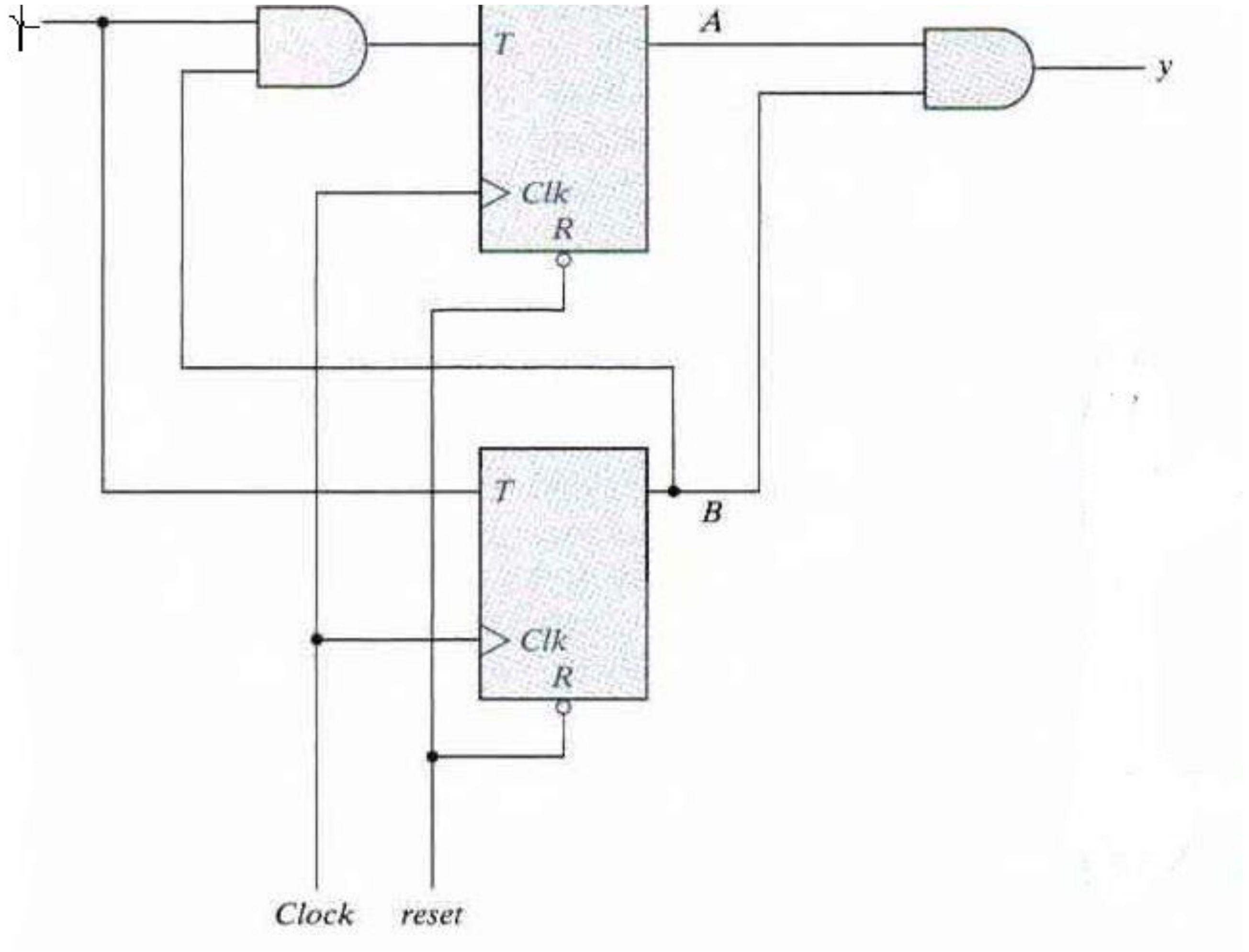
Equations:

T_B=x

T_A=x.B

Output equations:

Y=A. B



Now, based on the above equations

make state table

Analysis with T Flip-Flops

State Table

Present state		Input(x)	Flip flop inputs		Next state		Output(y)
A	B	X	T_A	T_B	A(t+1)	B(t+1)	0
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	1	1	0	1	0
1	1	0	0	0	1	1	1
1	1	1	1	1	0	0	1

Now, Draw state diagram

Design Procedure

The design of a clocked sequential circuit starts from a set of specifications (state table) and ends in a logic diagram or a list of Boolean functions from which the logic diagram can be obtained.

Procedure:

The procedure can be summarized by a list of consecutive recommended steps:

(1) State the word description of the circuit behavior. It may be a state diagram, a timing diagram, or other pertinent information.

(2) From the given information about the circuit, obtain the state table.

(3) Apply state-reduction methods if the sequential circuit can be characterized by input-output relationships independent of the number of states.

(4) Assign binary values to each state if the state table obtained in step 2 or 3 contains letter symbols.

(5) Determine the number of flip-flops needed and assign a letter symbol to each.

(6) Choose the type of flip-flop to be used.

(7) From the state table, derive the circuit excitation and output tables.

(8) Using the map or any other simplification method, derive the circuit output functions and the flip-flop input functions.

(9) Draw the logic diagram.

Design Procedure

Procedure step: (1) and (2)

- ✓ 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

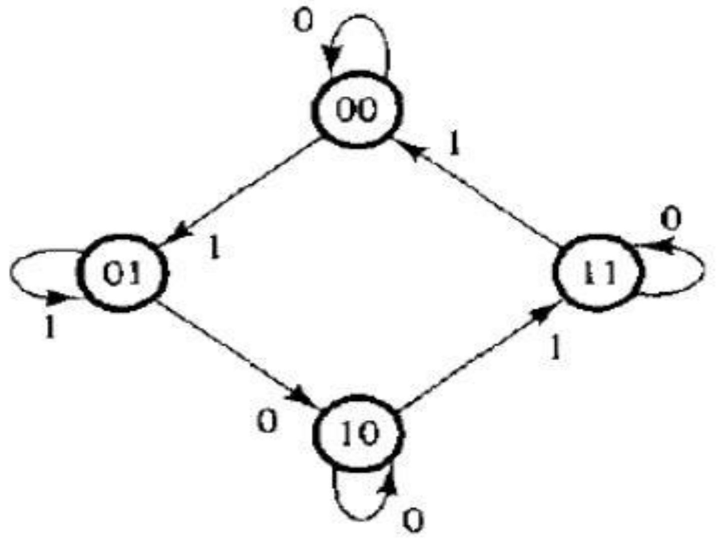
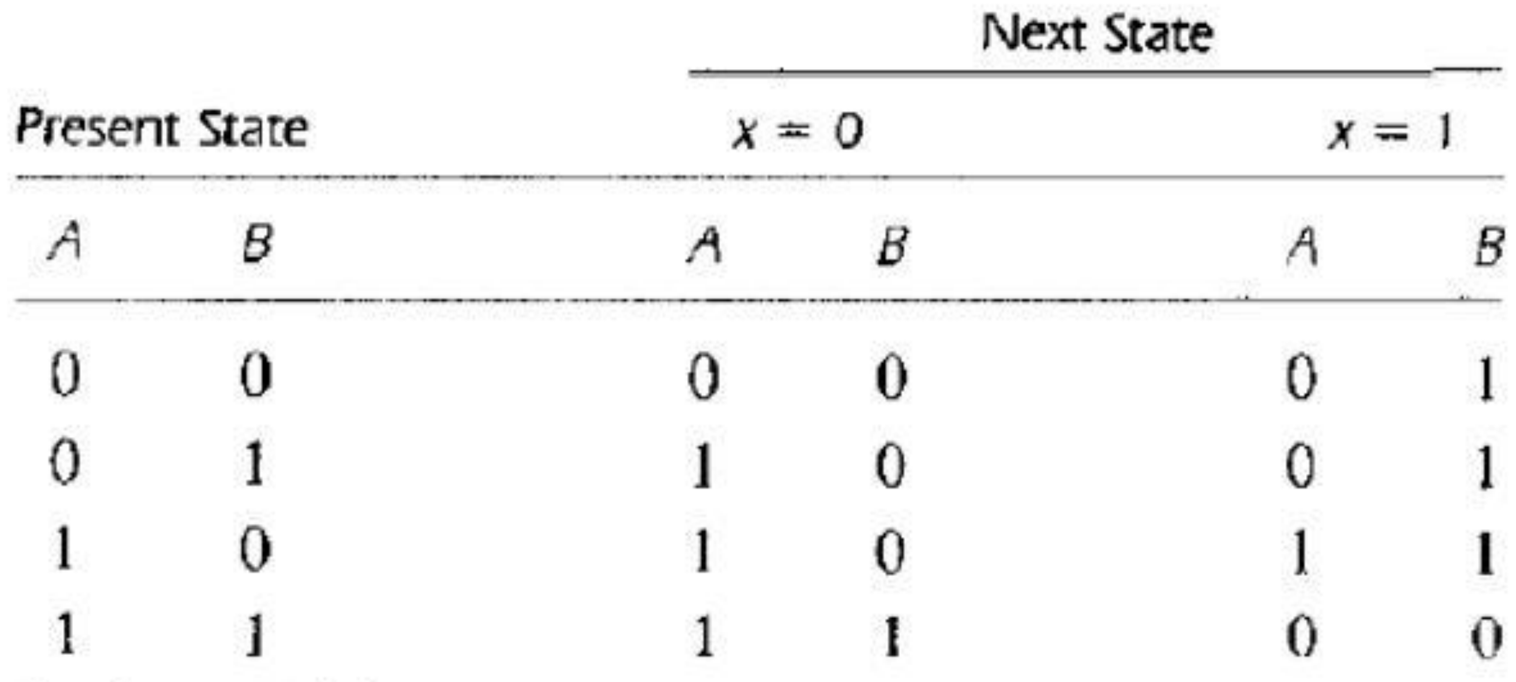


Fig: State-diagram for design example

Design Procedure

Procedure step: (3)

The state table for this circuit, derived from the state diagram. Note that there is no output section for this circuit.



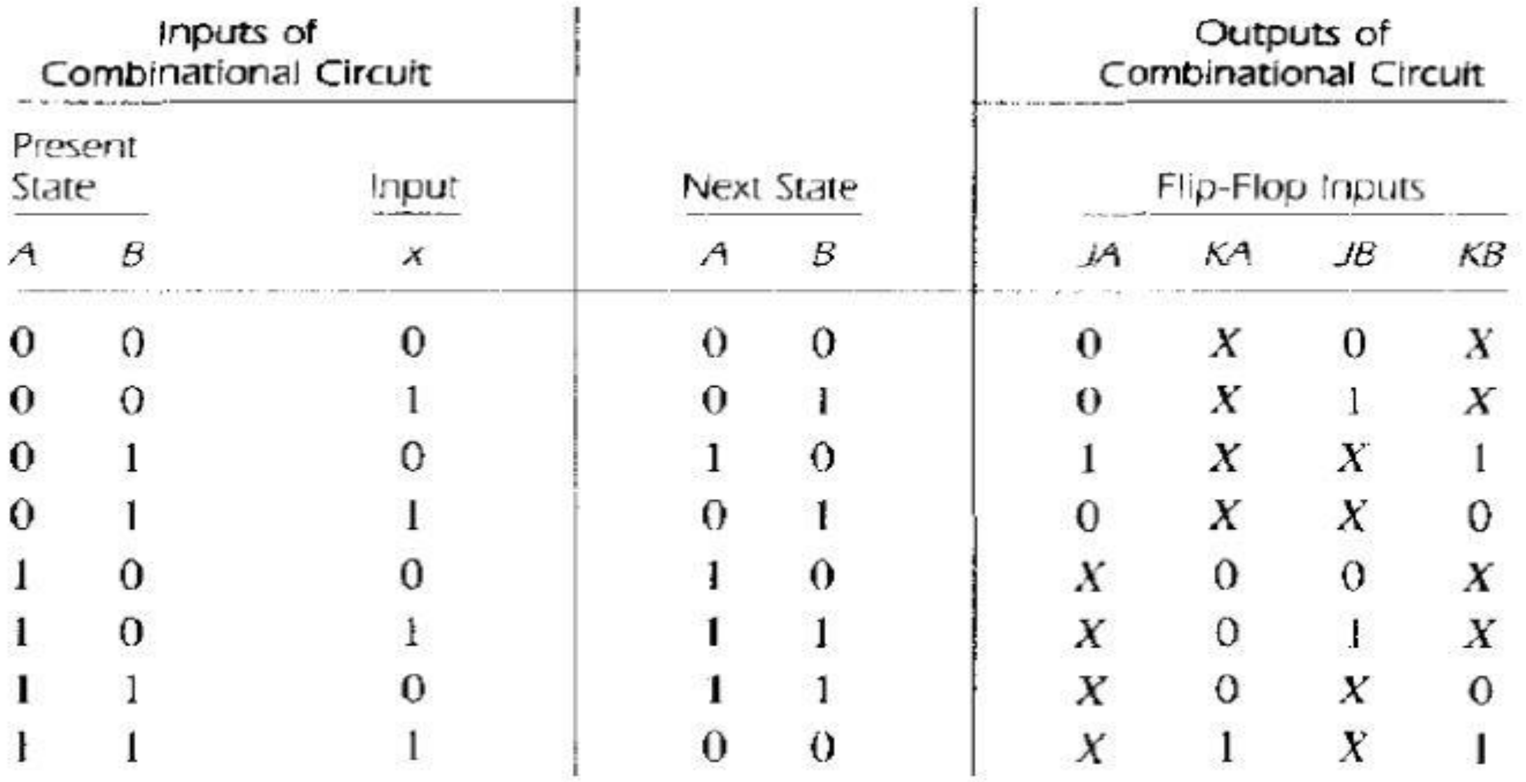
Present State		Next State			
		$x = 0$		$x = 1$	
A	B	A	B	A	B
0	0	0	0	0	1
0	1	1	0	0	1
1	0	1	0	1	1
1	1	1	1	0	0

Fig: State Table

Design Procedure

In the derivation of the excitation table, present state and input variables are arranged in the form of a truth table.

JK type is used here. Since JK flip-flops are used we need columns for the J and K inputs of flip-flops A (denoted by JA and KA) and B(denoted by JB and KB).

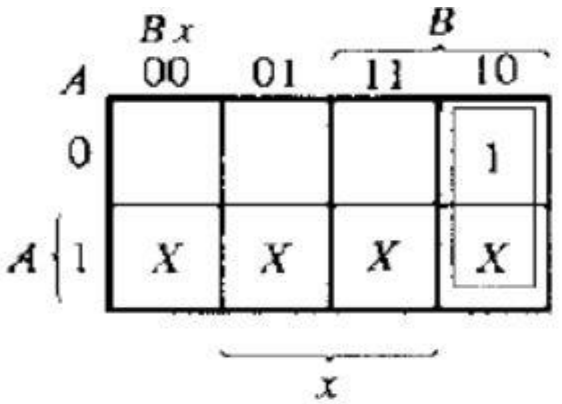


Inputs of Combinational Circuit					Outputs of Combinational Circuit			
Present State		Input	Next State		Flip-Flop Inputs			
A	*B*	*x*	*A*	*B*	*JA*	*KA*	*JB*	*KB*
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

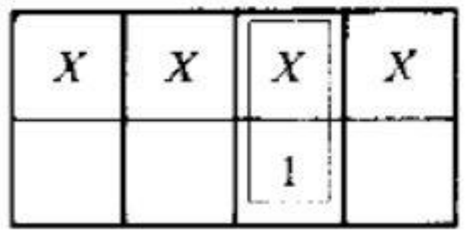
Design Procedure

Derivation of simplified Boolean functions for the combinational circuit. The information from the truth table is transferred into the maps. The inputs are the variables A, B, and x; the outputs are the variables JA, KA, JB, and KB.

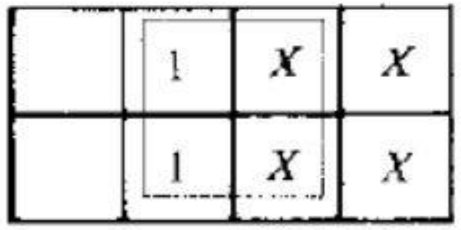
Now draw k map for JA, KA, JB and KB



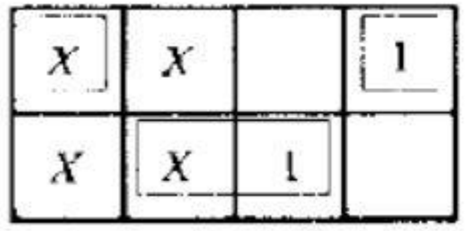
$JA = Bx'$



$KA = Bx$



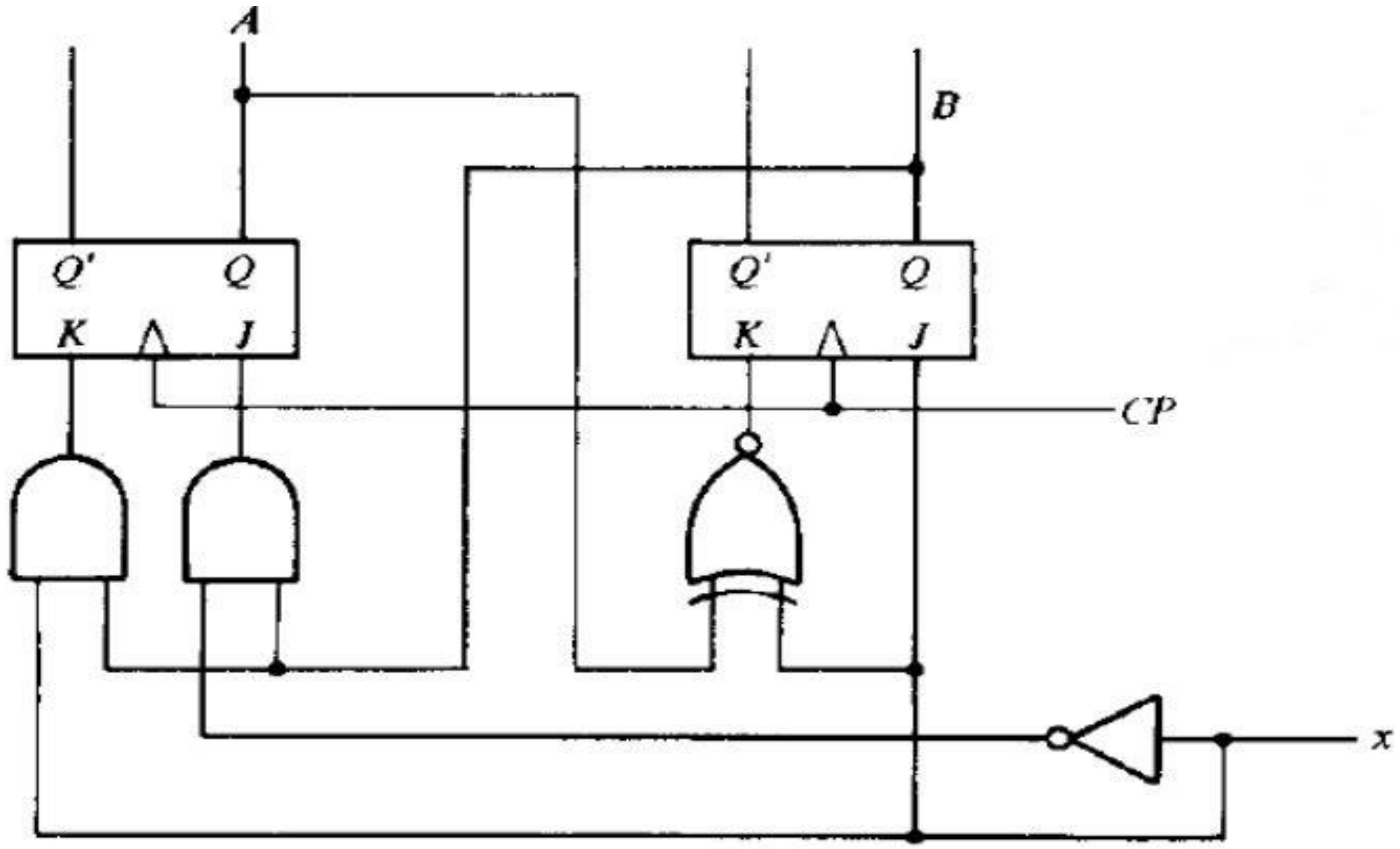
$JB = x$



$KB = (A \oplus x)'$

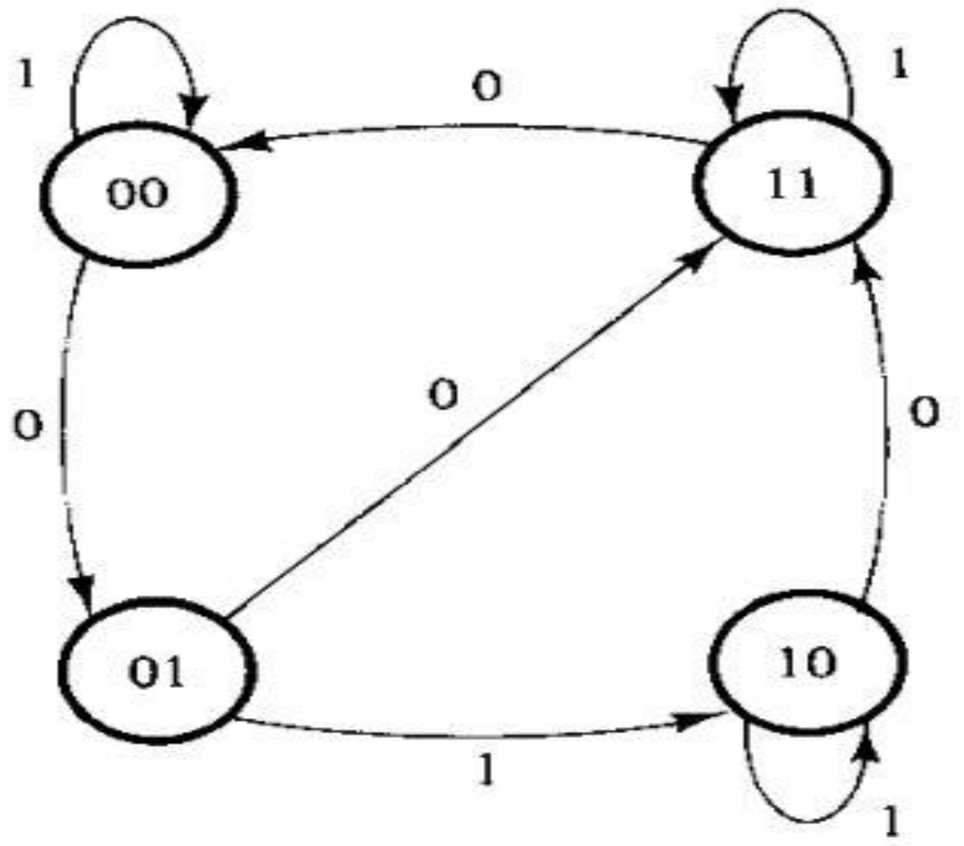
Design Procedure

Now, draw circuit for the above Boolean expression.



Exercise

Design the sequential circuit specified by the following state diagram using T flip-flop.



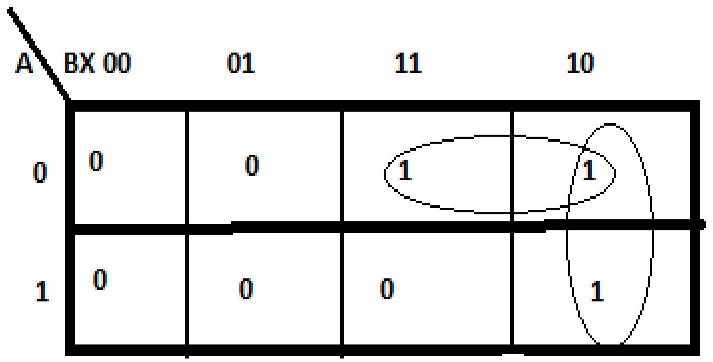
State Table

Present state		Input(x)	Next state		Flip Flop inputs	
A	B	X	A(t+1)	B(t+1)	T_A	T_B
0	0	0	0	1	0	1
0	0	1	0	0	0	0
0	1	0	1	1	1	0
0	1	1	1	0	1	1
1	0	0	1	1	0	1
1	0	1	1	0	0	0
1	1	0	0	0	1	1
1	1	1	1	1	0	0

Now, solve using K-map to find expression for T_A and T_B

K map for flip flop

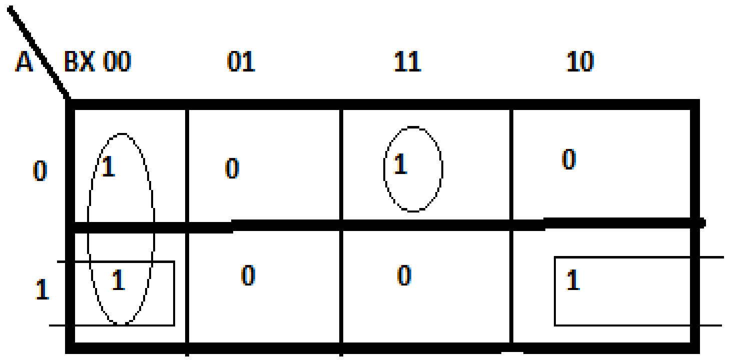
K map for T_A



A \ BX	00	01	11	10
0	0	0	1	1
1	0	0	0	1

$T_A = A'B + Bx'$

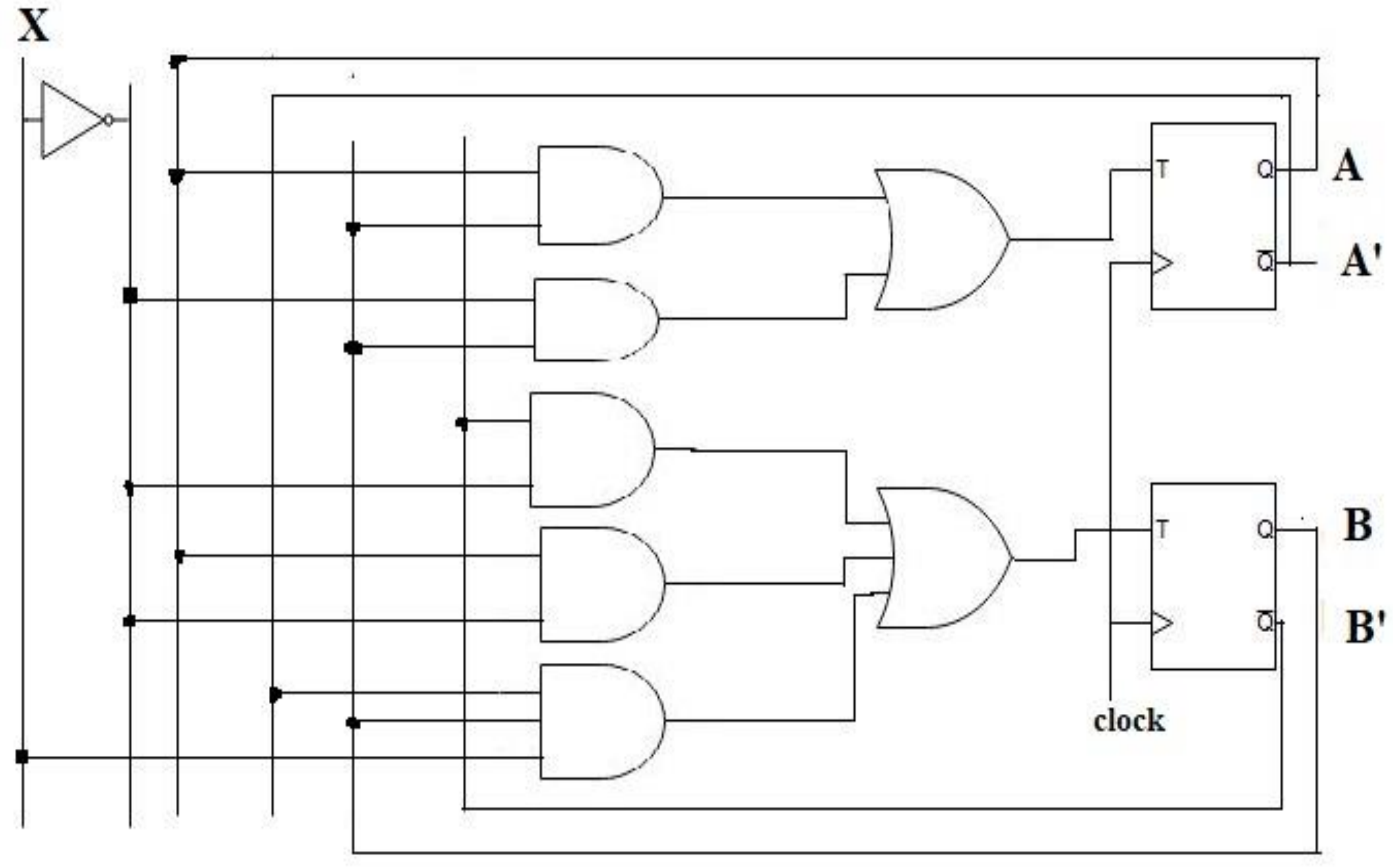
K map for T_B



A \ BX	00	01	11	10
0	1	0	1	0
1	1	0	0	1

$T_B = B'X' + AX' + A'BX$

Logical diagram


X
T
Q
Q
A
A'
T
Q
Q
B
B'
clock