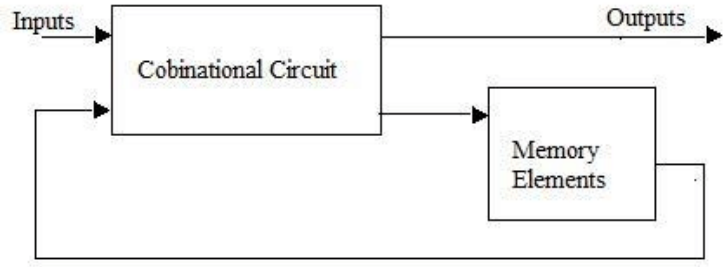
Unit-5
Sequential Logic

Unit-5
Sequential Logic

Sequential Circuit

A **sequential circuit** is one whose outputs depend not only on its current inputs but also on the past sequence of inputs.



The block diagram demonstrates that the external outputs in a sequential circuit are a function not only of external inputs but also of the present state of the memory element. The next state of the memory elements is also a function of external inputs and the present state. Thus, a sequential circuit is specified by a time sequence of inputs, outputs and internal states.

There are two main types of sequential circuits based on the timing of their signals-
1) Synchronous Sequential circuit.
2) Asynchronous Sequential Circuit.

A **Synchronous Sequential circuit** is a system whose behavior can be defined from the knowledge of its signals at discrete instants of time. It must employ signals that affect the memory elements only at discrete instants of time. One way of achieving this goal is to use clock pulses. The clock pulses are distributed throughout the system in such a way that memory elements are affected only with the arrival of the synchronization pulse. Synchronous sequential circuits that use clock pulses in the inputs of memory elements are called clocked sequential circuits.

The behavior of an **asynchronous sequential circuit** depends upon the order in which its input signals change and can be effected at any instant of time.

Flip-Flops

The memory elements used in a clocked sequential circuits are called flip-flops. These circuits are binary cells capable of storing one bit 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 can maintain a binary state indefinitely (as long as power delivered to the circuit) until directed by an input signals 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.

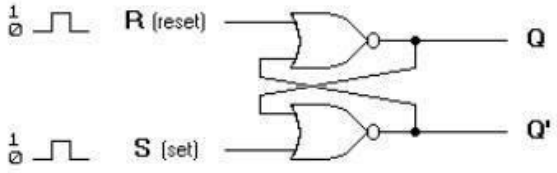
- The most common types of flip-flops are:
  a) RS Flip-Flop
  b) D Flip-Flop
  c) JK Flip-Flop
  d) T Flip-Flop

Basic Flip-Flop Circuit by NOR and NAND Gate

A flip-flop circuit can be constructed from two NAND gates or two NOR gates.

- 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. This types of flip-flop is sometimes called a **direct-coupled RS flip-flop** or **SR latch**.

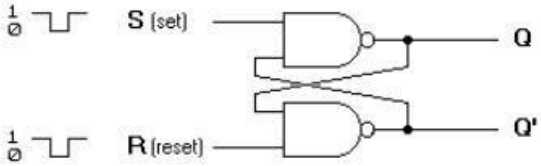


(a) Logic diagram

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

Basic flip-flop circuit with NOR gates



(a) Logic diagram

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

(b) Truth table

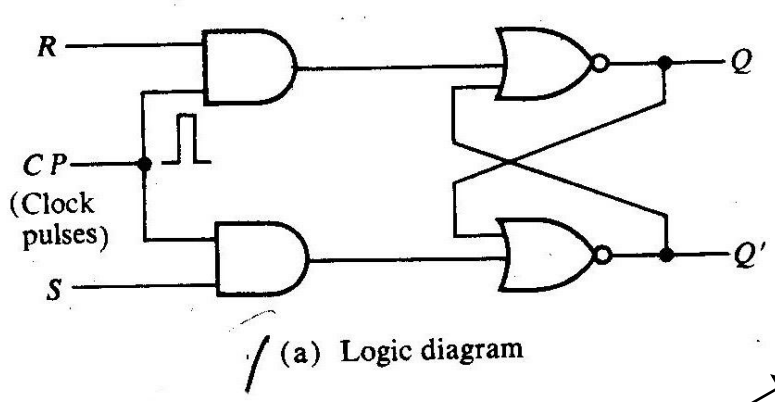
Basic flip-flop circuit with NAND gates

Note:
A flip-flop has two useful states. When $Q = 1$ and $Q' = 0$, it is in the set state (or 1-state) and when $Q = 0$ and $Q' = 1$, it is the clear state (or, 0-state).

Inputs	Output(Flip-flop with NOR gates)	Output(Flip-flop with NAND gates)
S=1 and R=0	goes set state	goes reset state
S=0 and R=1	goes reset state	goes set state
S=0 and R=0	remain unchanged	goes indeterminate (undefined) state
S=1 and R=1	goes indeterminate (undefined) state	remain unchanged

Clocked RS Flip-Flop

It consists of a basic NOR 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.

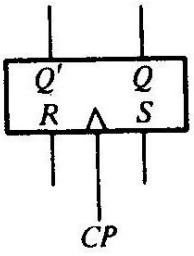


(a) Logic diagram

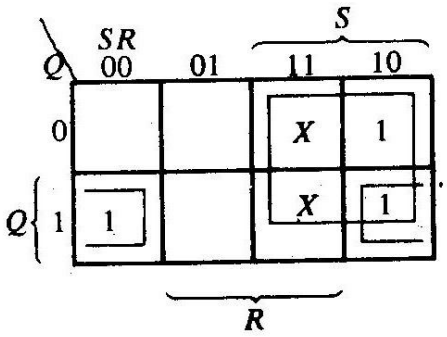
Q	S	R	Q(t + 1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	indeterminate
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	indeterminate

(c) Characteristic table

S	R	Q(t+1)
0	0	No change
0	1	0
1	0	1
1	1	Invalid



(b) Graphic symbol



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

$SR = 0$

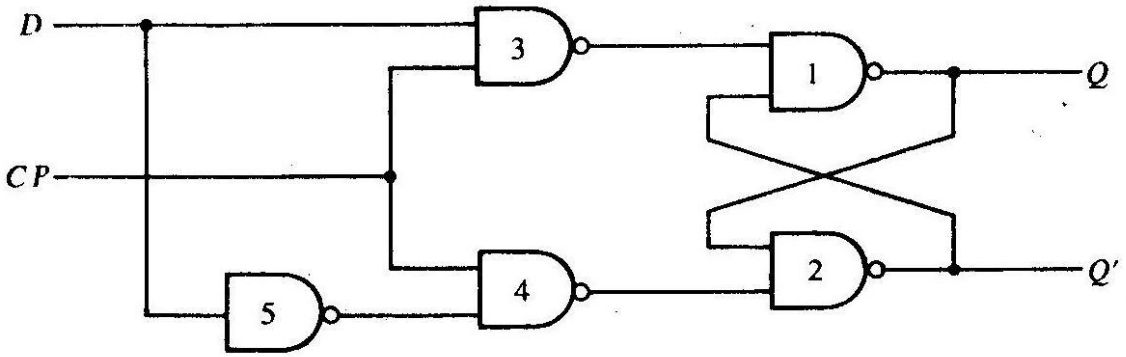
(d) Characteristic equation

Where, S and R are inputs. Q is the output and Q (t+1) shows the next state.

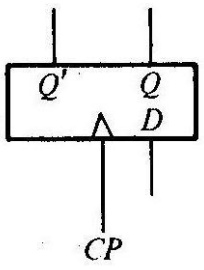
The two AND gates output remains at 0 as long as the clock pulse clocked is 0, regardless of the input values S & R. When clock pulse goes to 1, the information from inputs S & R is allowed to reach the basic flip flop.

- Set state: S=1, R=0 and CP=1 => Q=1 & Q'=0.
- Reset state: S=0, R=1, CP=1 => Q=0 & Q'=1
- When CP=1, inputs S=0 & R=0, that is when both the inputs are 0, the state of the circuit does not change.
- When CP=1, S=1 & R=1, an indeterminate condition occurs. Because both the outputs Q and Q' remain at 1. This is not possible because both the outputs are complementary to each other. So it is better to avoid this condition during practice.

D Flip-Flop



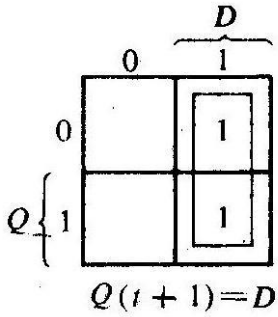
(a) Logic diagram with NAND gates



(b) Graphic symbol

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

(c) Characteristic table



(d) Characteristic equation

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

The D flip-flop is a modification of clocked RS flip-flop. NAND gates 1 and 2 form a basic flip-flop and gates 3 and 4 modify it into a clocked RS flip-flop.

The *D* flip-flop has only two inputs: *D* and *CP*. The D input of the flip-flop is directly given to S. And the complement of this value is given as the R input.

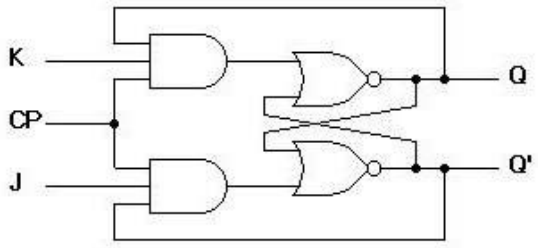
- As long as CP is 0, the outputs of gates 3 and 4 are at the 1 level and the circuit cannot change state regardless of the value of *D*.
- When CP changes to 1, the value of D is sampled and the information is passed to the output.
  - ✓ If D=1 & CP=1, the output of gate 3 goes to 0 and this makes the output Q=1, Q'=0. This is the set state of the D flip-flop.
  - ✓ If D=0 & CP=1, the output Q'=1 and Q=0, which is the reset state of the D Flip-flop.

JK Flip-Flop

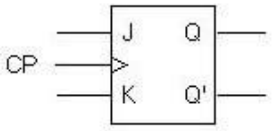
A *JK* flip-flop is a refinement of the *RS* flip-flop in that the indeterminate state of the *RS* type is defined in the *JK* type. Inputs *J* and *K* behave like inputs *S* and *R* to set and clear the flip-flop, respectively. The input marked *J* is for *set* and the input marked *K* is for *reset.*
When both the inputs J and K have a HIGH state (J=K=1), the flip-flop switches to the complement state. So, for a value of Q = 1, it switches to Q=0 and for a value of Q = 0, it switches to Q=1.

The output Q of the flip flop is returned back as a feedback to the input of the AND along with other inputs like K and clock pulse [CP]. So, if the value of CP is '1', the flip flop gets a CLEAR signal and with the condition that the value of Q was earlier 1. Similarly output Q' of the flip flop is given as a feedback to the input of the AND along with other inputs like J and clock pulse [CP]. So the output becomes SET when the value of CP is 1 only if the value of Q' was earlier 1.
Because of the feedback connection in the *JK* flip-flop, 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.

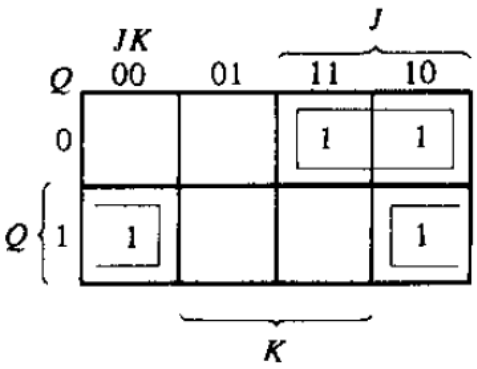
A *JK* flip-flop constructed with two cross-coupled NOR gates and two AND gates is shown in Fig. below:



(a) Logic diagram



(b) Graphical symbol



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

(c) Characteristic equation

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

(c) Transition table

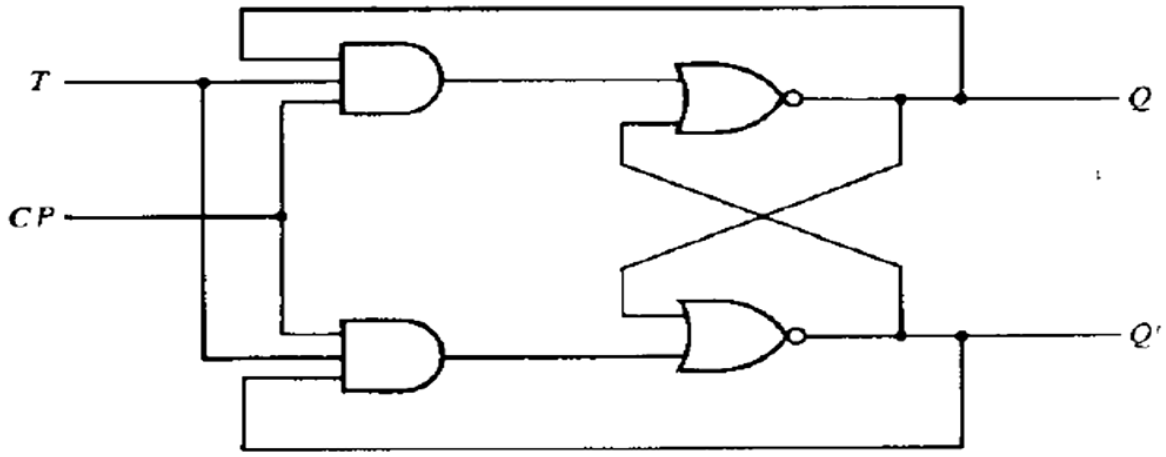
CP	J	K	Q(t+1)
0	X	X	X
1	0	0	Q(t)
1	0	1	0
1	1	0	1
1	1	1	$\overline{Q(t)}$

Clocked JK flip-flop

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:

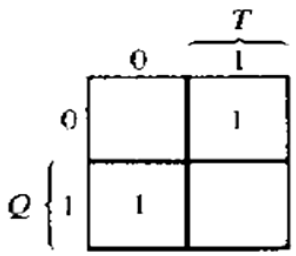
- When $T = 0$, $Q(t + 1) = Q$, that is, the next state is the same as the present state and no change occurs.
- When $T = 1$, then $Q(t + 1) = Q'$, and the state of the flip-flop is complemented.



(a) Logic diagram

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

CP	T	Q(t+1)
0	X	Q(t)
1	0	Q(t)
1	1	Q'(t)

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.

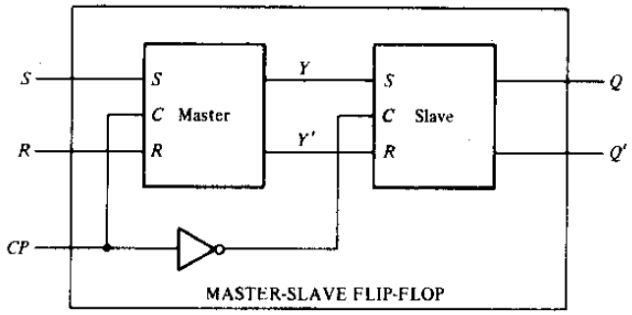


Fig: Logic diagram of Master-Slave flip-flop

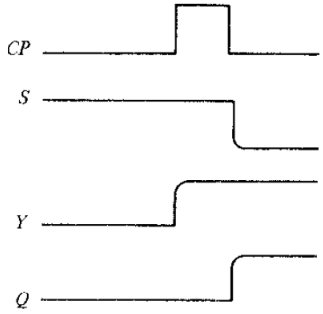


Fig: Timing Relationships Master-Slave flip-flop

Truth Table:

S	R	CP	Q(t+1)	$\overline{Q(t+1)}$
0	0	1	Q(t)	$\overline{Q(t)}$
0	1	1	0	1
1	0	1	1	0
1	1	1	Undefined	Undefined
X	X	0	$Q(t)$	$\overline{Q(t)}$

- **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.
The clock input is normally 0, which keeps the outputs of gates 1 & 2 at the 1 level. This 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, the output of gate 9 is 1. So, output $Q = Y$, 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, because the output of gate 9 provides a 1 to both inputs of the NAND basic flip-flop of gates 7 and 8. 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.

J	**K**	**C**	**Q(t+1)**	$\overline{\mathbf{Q(t+1)}}$
0	0	1	Q(t)	$\overline{Q(t)}$
0	1	1	0	1
1	0	1	1	0
1	1	1	$\overline{Q(t)}$	Q(t)
X	X	0	Q(t)	$\overline{Q(t)}$

Truth table

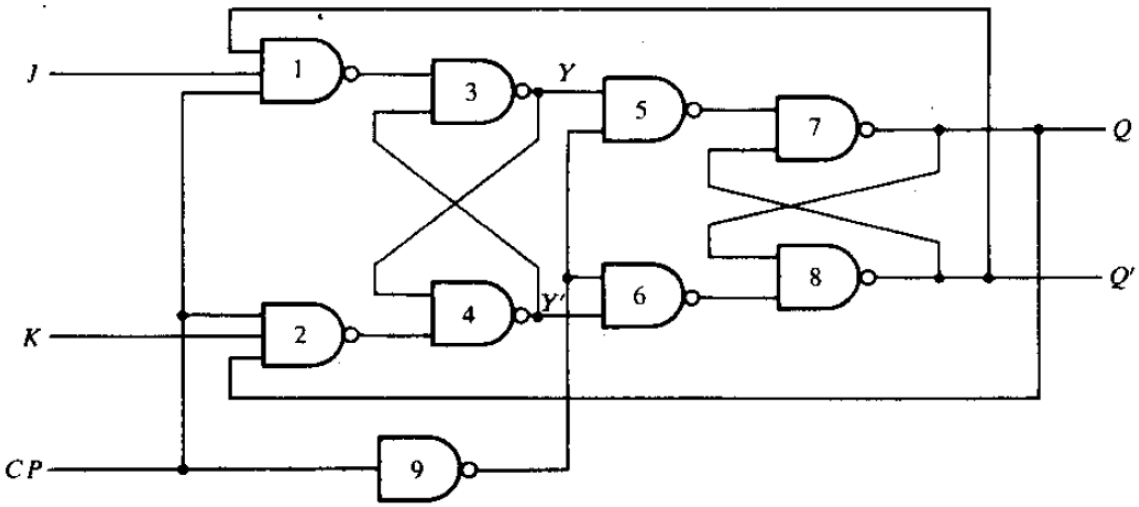


Fig: Clocked Master-Slave JK flip-flop

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.*

- This definition applies also to negative pulses.

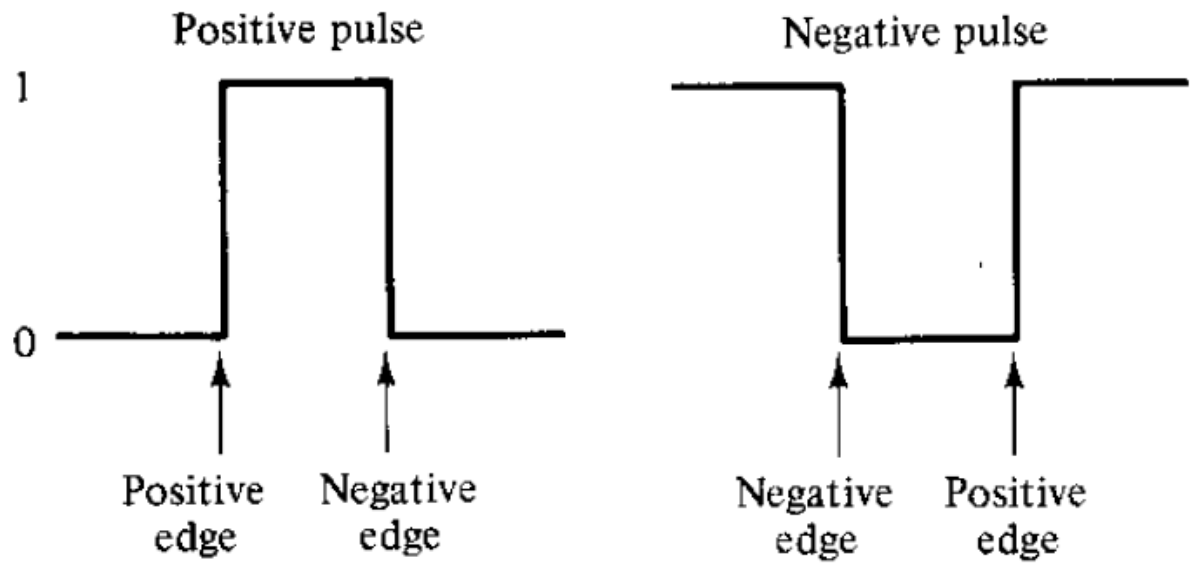


Fig: Definition of clock pulse transition.

The clocked flip-flops introduced earlier are triggered during the positive edge of the pulse, and the state transition starts as soon as the pulse reaches the logic-1 level. The new state of the flip-flop may appear at the output terminals while the input pulse is still 1. If the other inputs of the flip-flop change while the clock is still 1, the flip-flop will start responding to these new values and a new output state may occur.

One of the way to achieve edge triggering is to use a **master-slave or edge-triggered 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).

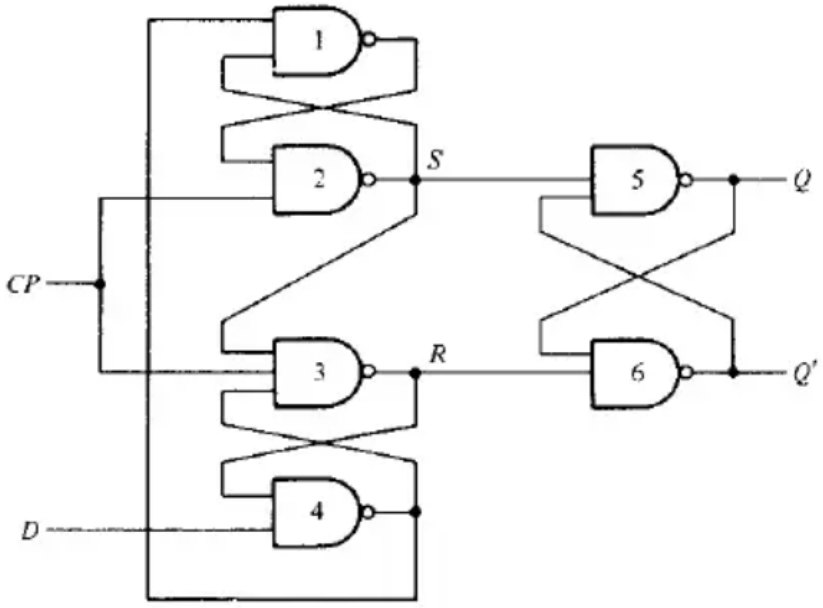
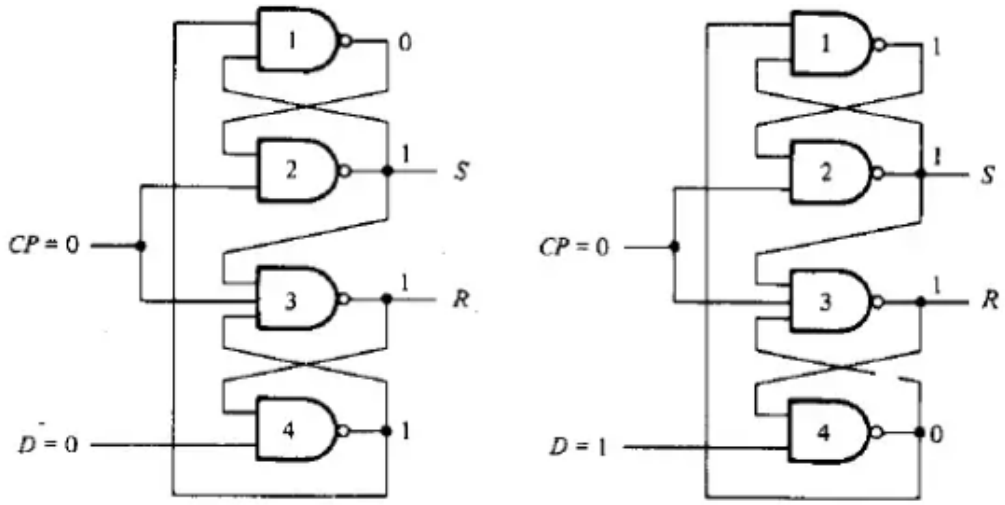
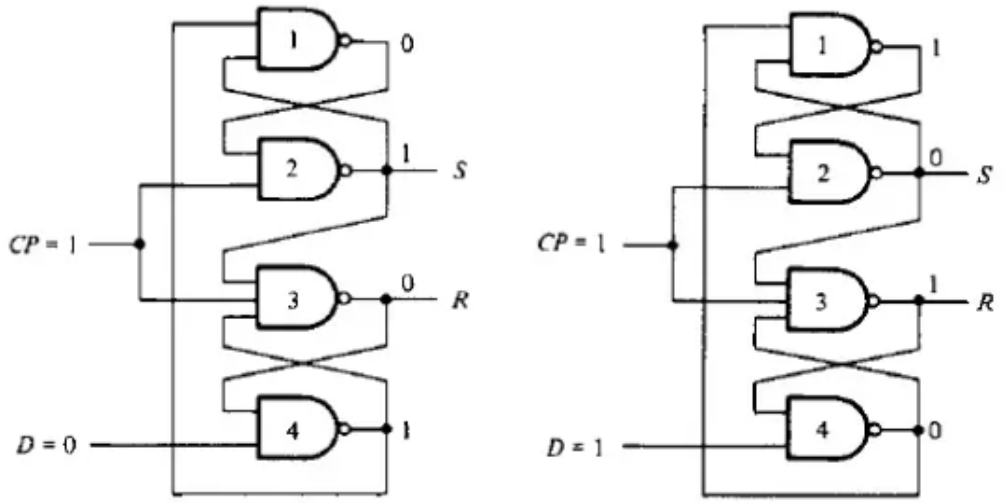


Fig: D-type positive-edge-triggered flip-flop

Operation:



(a) With $CP = 0$



(b) With $CP = 1$

Gates 1 to 4 are redrawn to show all possible transitions. Outputs S and *R* from gates 2 and 3 go to gates 5 and 6. 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.
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).

Analysis of Clocked Sequential Circuit

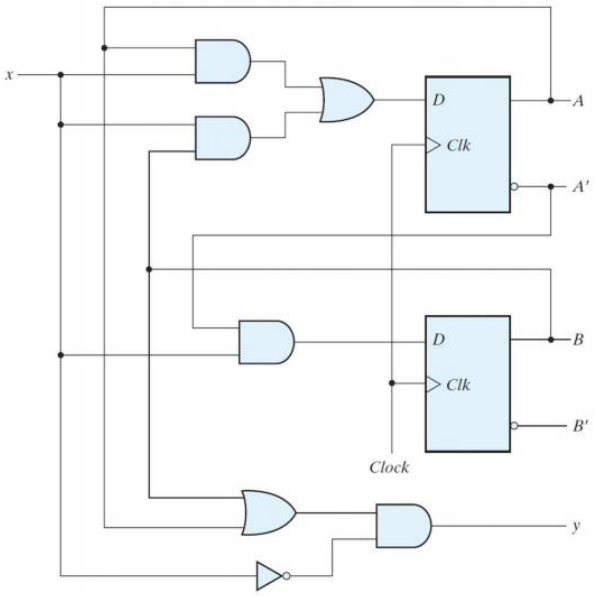
The behavior of a sequential circuit is determined from the inputs, the outputs and the states of its flip-flop. Both the outputs and the next state are a function of the inputs and the present state.
The analysis of sequential circuits consists of obtaining a table (known as state table) or a diagram (known as state diagram) or boolean expression (known as state equation) that describe the behavior of sequential circuits.

State Table: The time sequence of inputs, outputs and flip-flop states is enumerated in a state table. It consists of three sections labeled present state, next state and output. Present state designates the states of flip-flops before clock and next state shows the state after clock. Output section lists the values of the output variables during the present state.

State Diagram: The information available in a state table may be represented graphically in a state diagram.

State Equations: A state equation (also known as an application equation) is an algebraic expression that specifies the conditions for a flip-flop state transition. The left side of the equation denotes the next state of a flip-flop and the right side, a Boolean function that specifies the present state conditions that make the next state equal to 1.

E.g.
Q. Obtain the state table, state diagram and state equations of the following sequential circuit.



Here,
Type of memory: D Flip-Flops
Two state variables: A and B
One input: x
One output: y

State Equations:
The equations on the D inputs of the flip-flops:

$$D_A = Ax + Bx$$
$$D_B = A'x$$

A and Bare the current state: $A(t) = A$ & $B(t) = B$
D_A and D_B are the next state: $A(t+1) = D_A, B(t+1) = D_B$
The value of A and B will be D_A and D_B at the next clock edge.

∴ Next **state equations:** $\boldsymbol{A(t+1) = Ax + Bx}$ and $\boldsymbol{B(t+1) = A'x}$
Output equation: $\boldsymbol{y = (A + B)x'}$

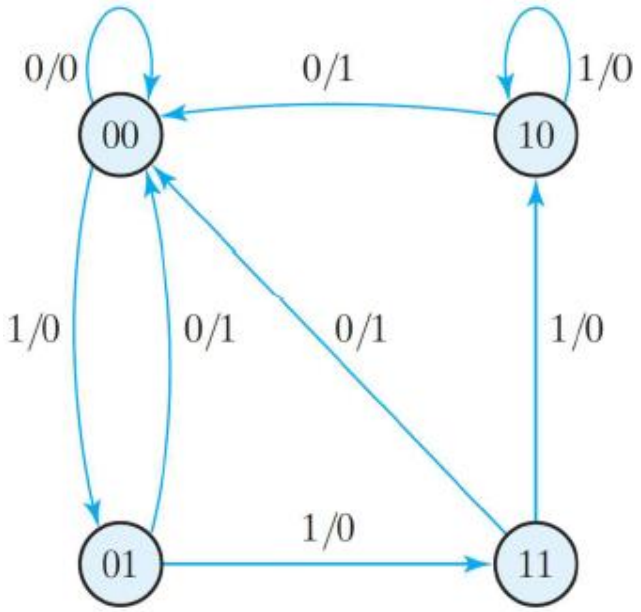
State table:

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

Another 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:

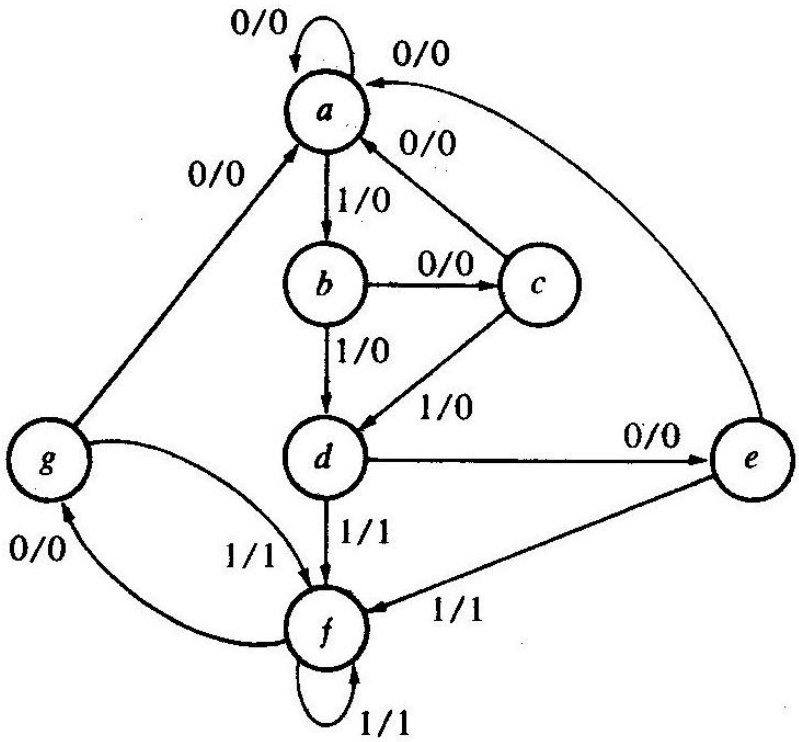


State Reduction

The reduction of number of flip-flops in a sequential circuit is referred to as the state reduction problem. State reduction algorithms are concerned with procedures for reducing the number of states in a state table while keeping the external input-output requirements unchanged.

E.g.

Reduce the number of states of the state diagram shown in fig..



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

Algorithm of state reduction: 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.

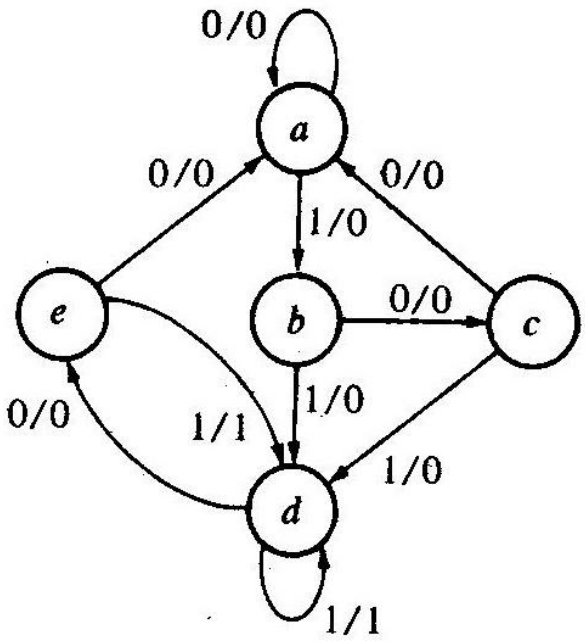
Reducing to the state table:

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

Reduced 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*	*d*	0	1
e	*a*	*d*	0	1

Reduced state diagram:



Flip-Flop Excitation Tables

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

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

(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*

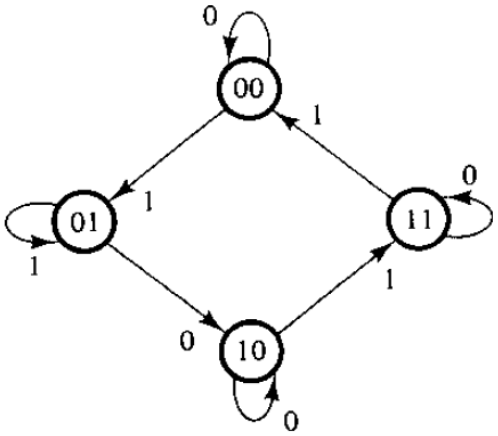
Design Procedure of clocked Sequential Circuit

Steps in design procedure:

1. Description of circuit behavior is stated. This may be accompanied by a state diagram, a timing diagram or other pertinent information.
2. From the given information about the circuit, obtain the state table.
3. The number of states may be reduced by state reduction method.
4. Assigned 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. Number of flip-flops is determined from number of steps.
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.

E.g.

Design a clocked sequential circuit whose state diagram is given in figure.



Solution:

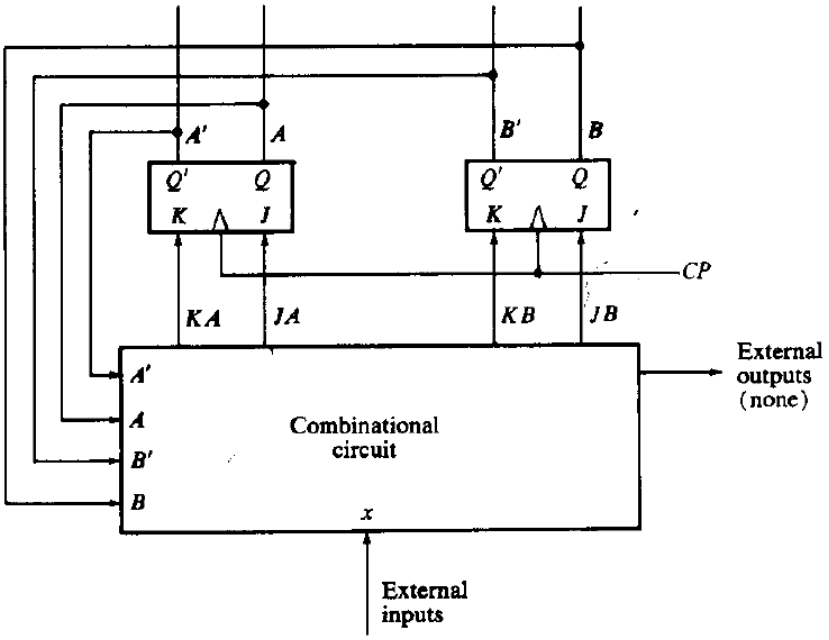
Finding state table:

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

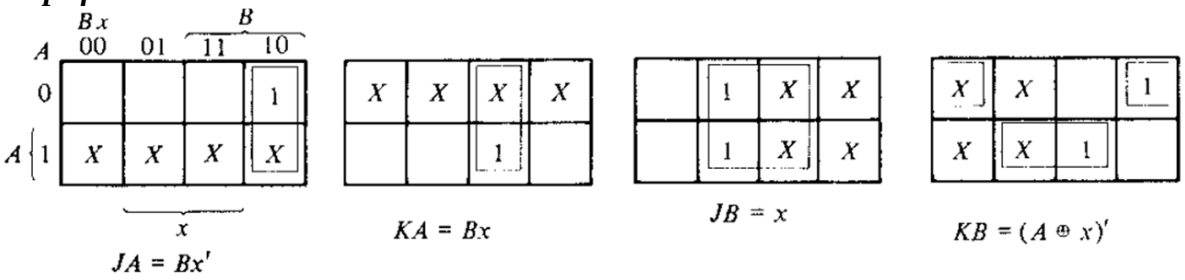
Excitation table:

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

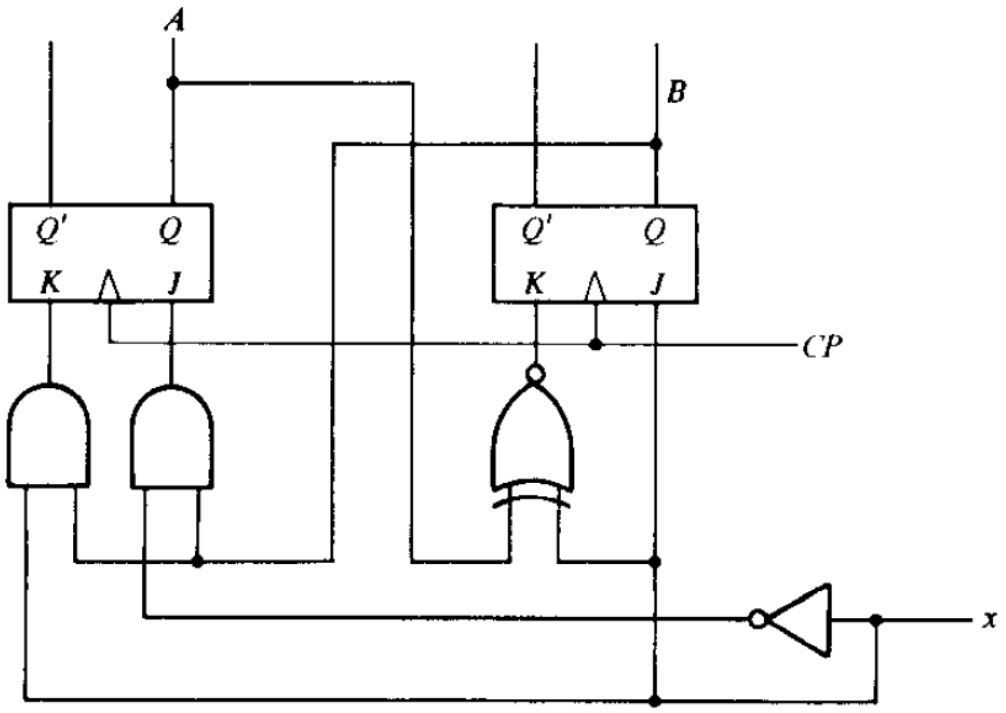
Block diagram of sequential circuit:



Maps for combinational circuit:



Logic diagram of sequential circuit:



References:

- ***M. Morris Mano, "Digital Logic & Computer Design"***