Flip-Flops

Types of Circuit

- Combinational Circuit
  - The outputs depend only on the current input values
  - It uses only logic gates
- Sequential Circuit
  - The outputs depend on the current and past input values
  - It uses logic gates and storage elements
  - Example
    - Vending machine
  - They are referred as finite state machines since they have a finite number of states

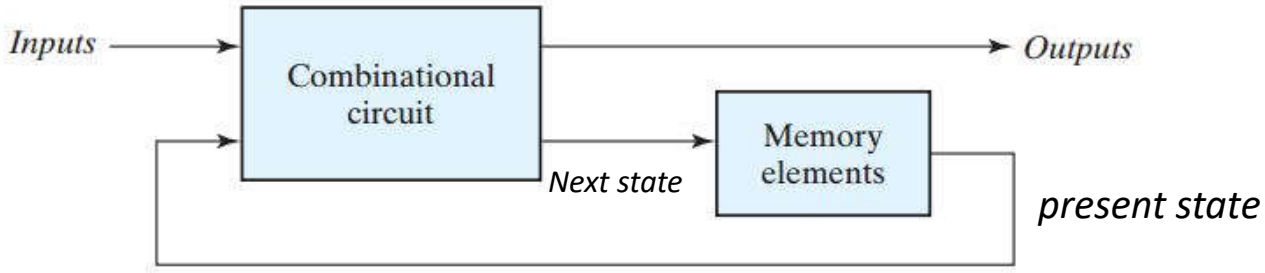


Figure: Block Diagram of Sequential Circuit

- Memory/Storage elements are bistable device i.e. devices capable of storing binary information i.e. 1 or 0. it has two stable states.
- Binary information stored in these elements at the given time determine the **state** of the sequential circuit at that time.
- **Output** is the function of present input and present state of storage element.
- Also the next state is the function of external inputs and the present state
- Thus a sequential circuit is specified by a time sequence of inputs, outputs, and internal states.

Types of Sequential Circuit

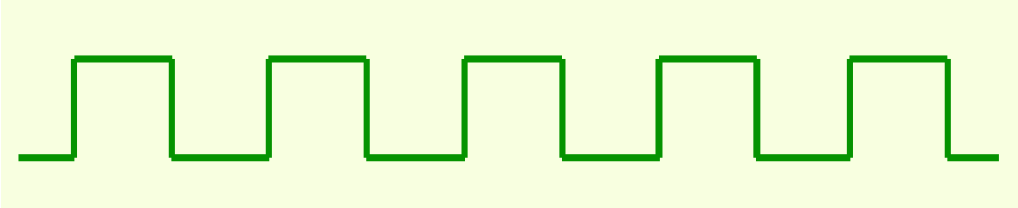
- Synchronous Sequential Circuit
- Asynchronous Sequential Circuit

Synchronous Sequential Circuit

- The circuit **behavior** is *determined by the signals at discrete instants of time.*
- The *memory elements are affected only at discrete instants of time*.
- A **clock** is used for **synchronization**.
  - Memory elements are affected only with the arrival of a clock pulse.
  - If memory elements use clock pulses in their inputs, the circuit is called **clocked sequential circuit.**
  - The memory elements used in clocked sequential circuits are called **flip-flops**.

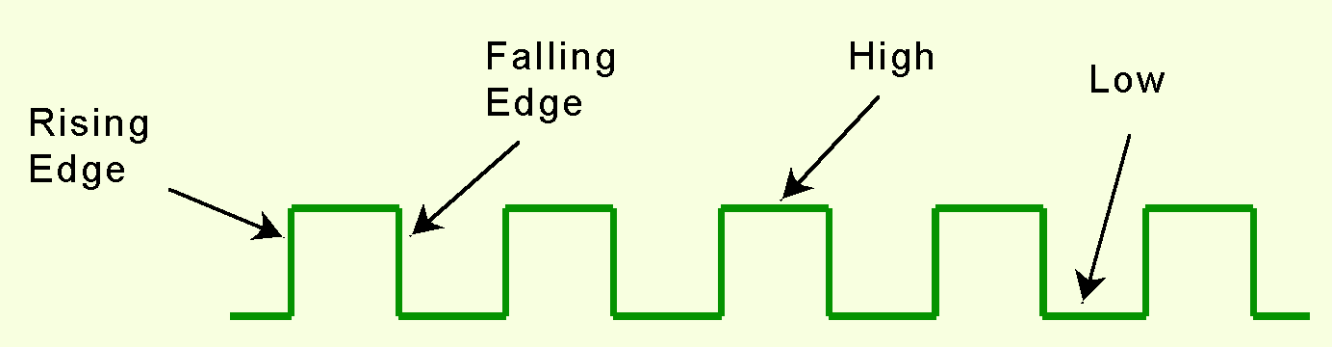
Clock and Synchronization

- **Synchronization** is achieved by a timing device called a **master-clock generator**, which ***generates a periodic train of*** **pulses.**
- A “clock” is a special circuit that sends electrical pulses through a circuit.
- Clocks produce electrical waveforms such as the one shown below.



- It emits a series of pulses with a precise **pulse width** and precise interval between consecutive pulses
- Timing interval between the corresponding edges of two consecutive pulses is known as the **clock cycle time**, or **period**

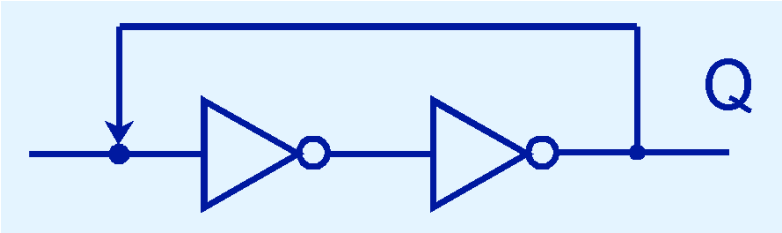
- Circuits that change state on the rising edge, or falling edge of the clock pulse are called *edge-triggered.*
- *Level-triggered circuits* change state when the clock voltage reaches its highest or lowest level.



Feedback

- To retain their state values, sequential circuits rely on *feedback*.
- Feedback in digital circuits occurs when an output is looped back to the input.
- A simple example of this concept is shown below.
  - If Q is 0 it will always be 0, if it is 1, it will always be 1.

  Why? (double-inverter)
  - A one-bit buffer (memory)



Asynchronous Sequential Circuit

- The circuit behavior is determined by the order the inputs change and can be affected by the signals at any instants of time.
- The memory elements used in asynchronous sequential circuits are time-delay devices, which constitute logic gates and whose propagation delay constitute memory element.
- Thus, an asynchronous sequential circuit may be regarded as combinational circuits with feedback.
- Because of this feedback, it causes instability due to which this asynchronous are not much used by designer.

Flip-Flops

- These are memory elements used in clocked sequential circuits.
- They are capable of storing one bit of information at a time.
- It has two outputs, one for *normal* value and *complement* value of the bit stored in it.
- The **state remains indefinitely until** and unless the clock pulse is applied. Then on application of input, it changes its state depending upon present inputs and present state.

Types

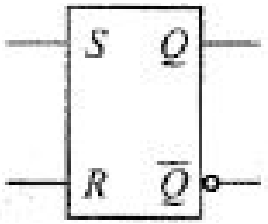
- **RS flip flop**
- **JK flip flop**
- **D flip flop**
- **T flip flop**

Latches

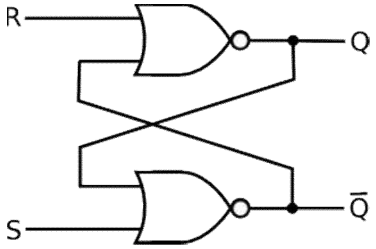
- Transparent→ the output Q immediately follows any change of state at the input.
- Memory element without clock
- Used for asynchronous sequential circuit.
- Level-sensitive device.
- Flip-flops are constructed from latches.

RS Latch with NOR

- The SR Latch is a circuit with two cross-coupled NOR gates or two cross-coupled NAND gates, and the two inputs are labelled S for SET and *R for RESET*.



a) Logical Symbol



b) Logical Diagram

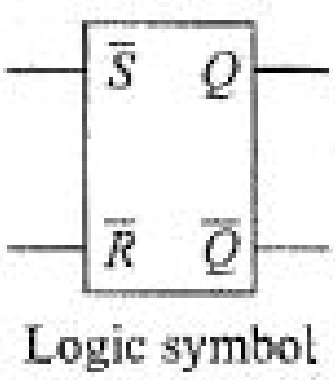
R	S	Q	Action
0	0	Last State	No Change
0	1	1	SET
1	0	0	RESET
1	1	?	Forbidden

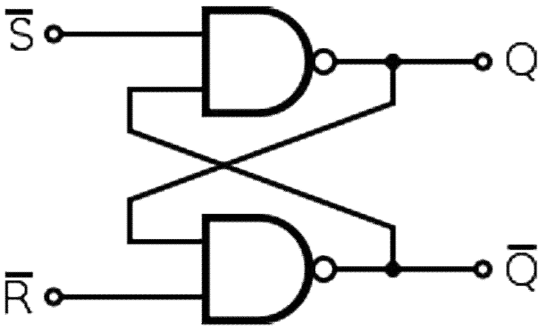
c) Truth Table

Q=1 and Q' = 0 → SET state
Q=0 and Q' = 1 → RESET State
S=1 and S=1 → undefined state, Q and Q' are both 0.

Figure: SR Latch with NOR gates

RS Latch with NAND





$\overline{R}$	$\overline{S}$	Q
1	1	Last state
1	0	1
0	1	0
0	0	? (Forbidden)

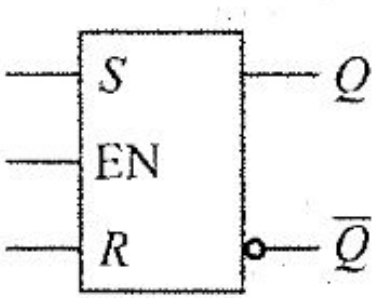
a) Logical Symbol

b) NAND gate latch

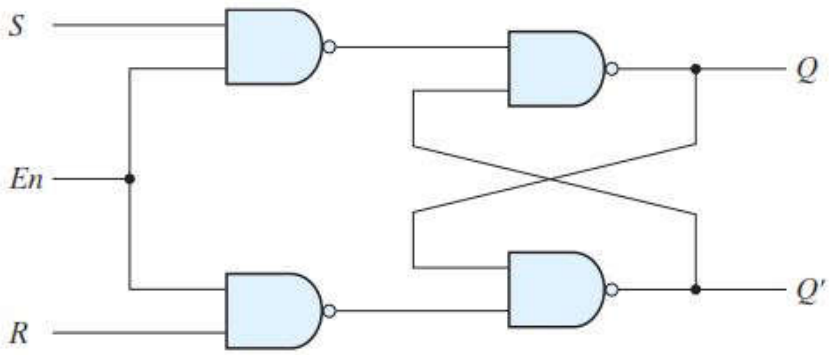
c) Truth Table

Figure: RS Latch with NAND gates ($\overline{R}\overline{S}$ Latch)

SR Latch with Control Input/Gated RS latch



a) Logical symbol



b) Logical Diagram

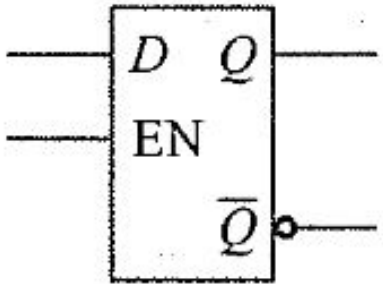
EN	S	R	Q_{n+1}
1	0	0	Q_n (no change)
1	0	1	0
1	1	0	1
1	1	1	? (Illegal)
0	X	X	Q_n (no change)

c) Truth Table

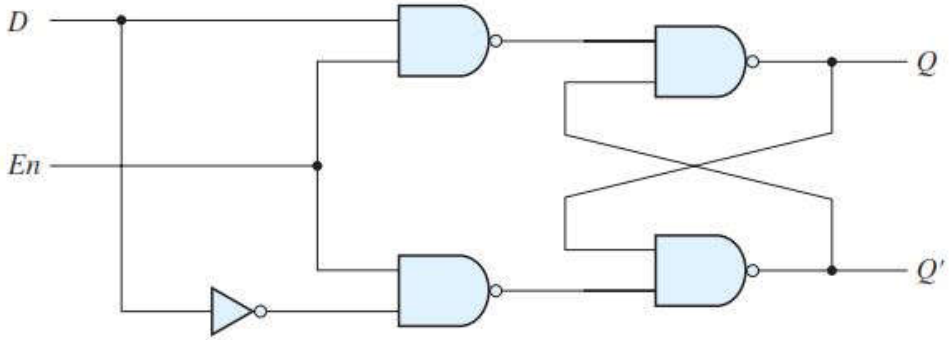
Figure: Rs Latch with Control Input/ Gates RS Latch

D Latch (Transparent Latch)/Gated D Latch

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



a) Logical Symbol



b) Logical Diagram

EN	D	Q_{n+1}
0	X	Q_n (last state)
1	0	0
1	1	1

c) Truth Table

Figure: Gated D Latch

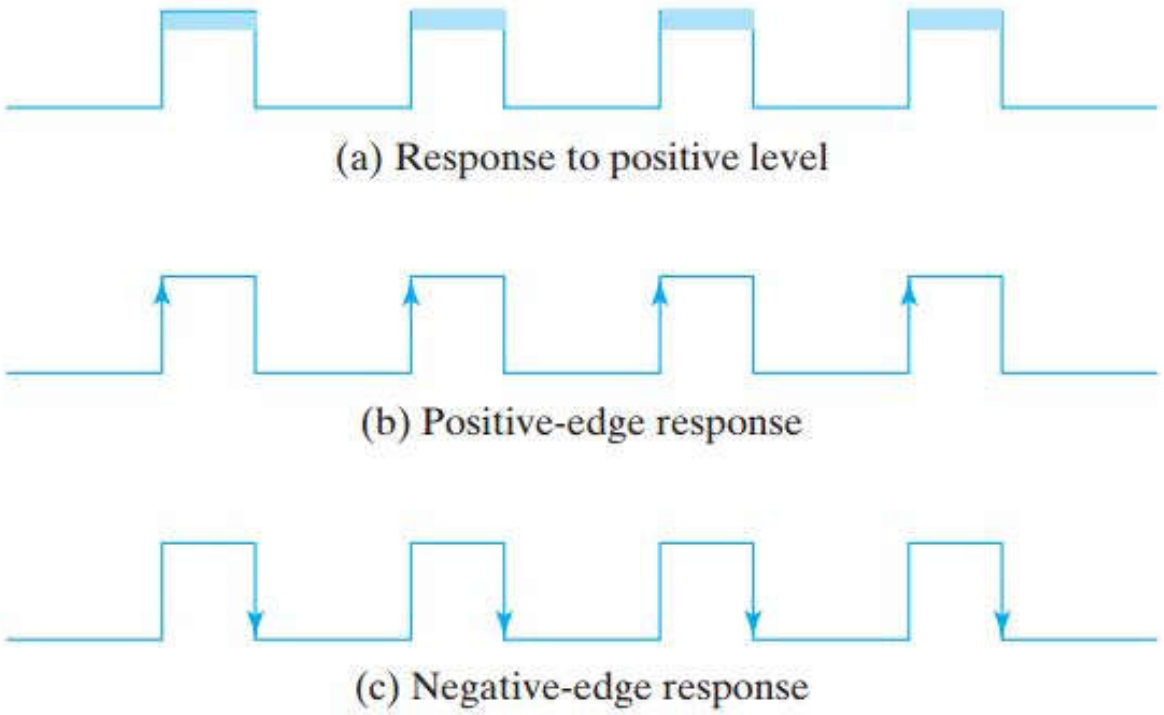


Figure: Clock Response in latch and flip-flop

Edge Triggering Flip-Flops

- Flip flops are synchronous bistable devices, also known *as bistable mutivibrators.* In this case, they are term synchronous means that the output changes state only at a specified point on a triggering input called the *clock* (CLK), which is designated as a control input.

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

- Each type of flip flops can be either positive edge triggered (no bubble at CLK input) or negative edge-triggering (bubble at C input). They key to identifying an edge-triggered flip-flop by its logic symbol is the small triangle inside the block at the clock (C) input. This triangle is called the *dynamic input indicator,* ▷ .

SR Flip-flop

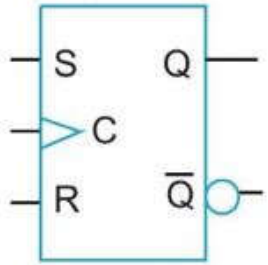


Figure: Graphical symbol of positive edge triggered SR flip flop

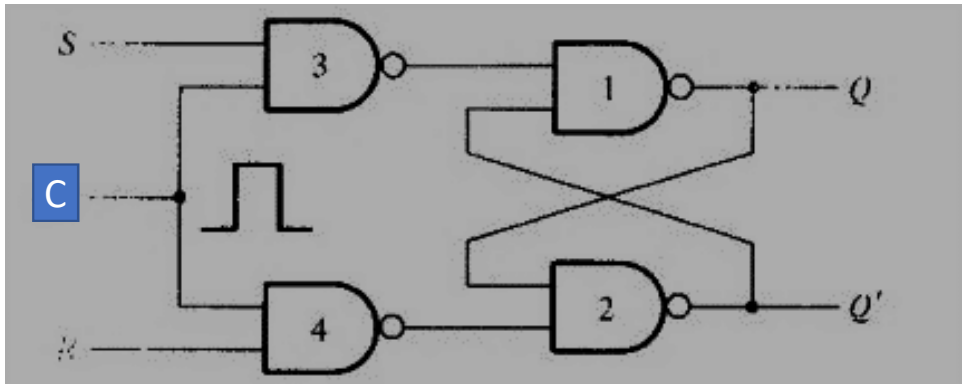


Figure: Logical Diagram

Figure: Logical Diagram

C	S	R	Q(t+1)	Action
↑	0	0	Q(t)	No Change
↑	0	1	0	RESET
↑	1	0	1	SET
↑	1	1	?	Illegal

Figure:Truth Table

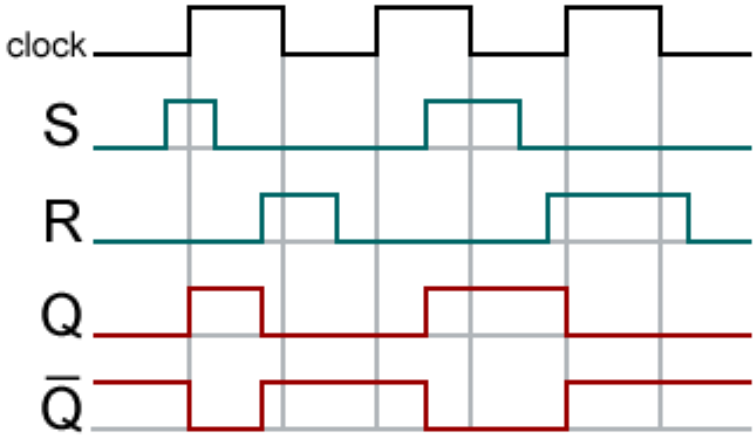


Figure: Timing Diagram

Q(*t*)	S	R	Q(*t+1*)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	?
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	?

Figure: Characteristic Table

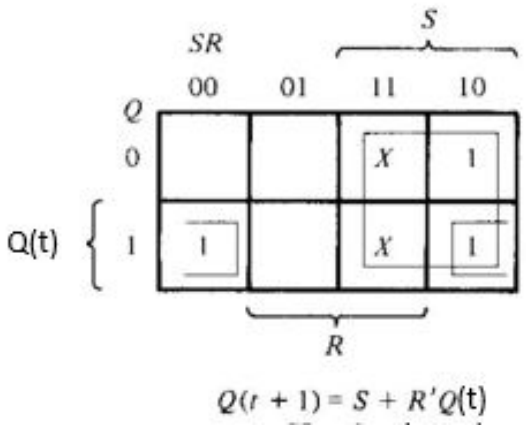


Figure: Characteristic Equation

Excitation Table

Q(t)	Q(t+1)	S	R	Operation
0	0	0	X	No change
0	1	1	0	Set
1	0	0	1	Reset
1	1	X	0	No change

Figure: Excitation Table

D flip-flop

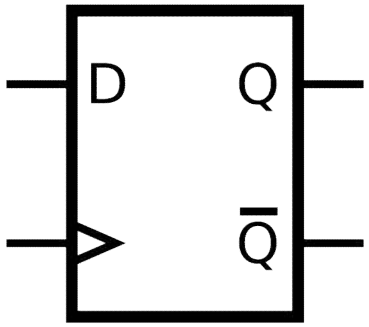


Figure: Graphical Symbol

C	D	Q(t+1)	Action
0	X	Q(t)	Last State
↑	0	0	RESET
↑	1	1	SET

Figure: Truth Table

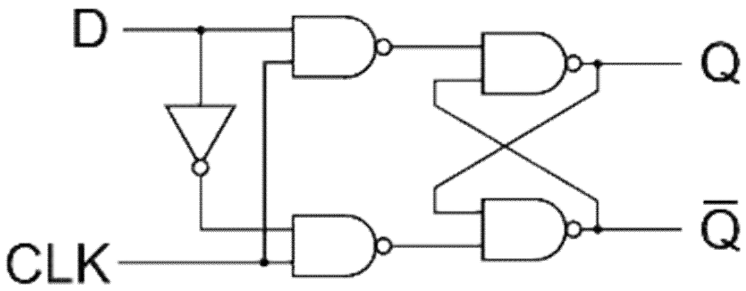


Figure: Logical Diagram

Figure: Timing Diagram

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

Figure: Characteristic Table

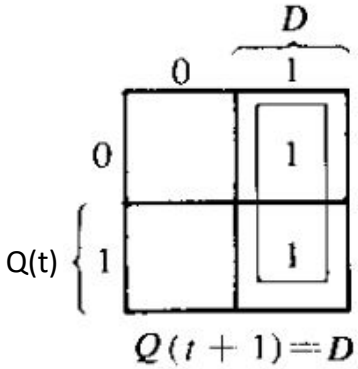
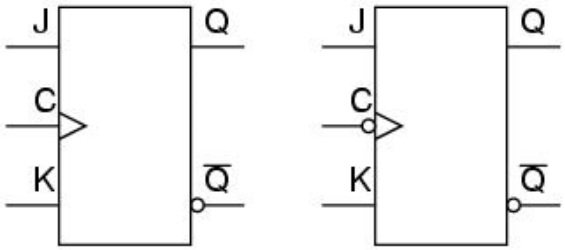


Figure: Characteristic Equation

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

Figure: Excitation Table

JK Flip-flop



Positive Edge Triggering

Negative Edge Triggering

Figure: Graphical Symbol

C	J	K	Q(t+1)	Action
↑	0	0	Q(t)	No Change
↑	0	1	0	RESET
↑	1	0	1	SET
↑	1	1	Q(t)'	Toggle

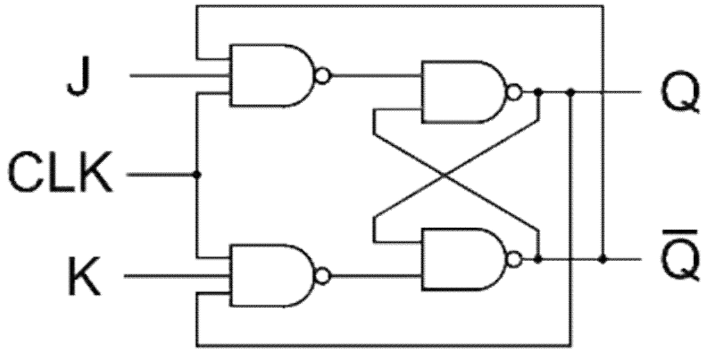
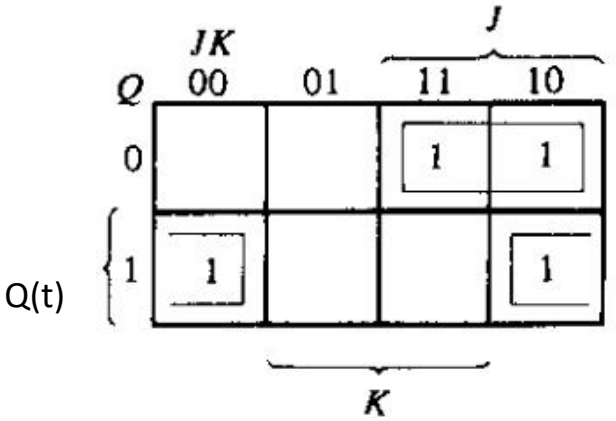


Figure: Logical Diagram

Figure: Timing Diagram

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

Figure: Characteristic Table



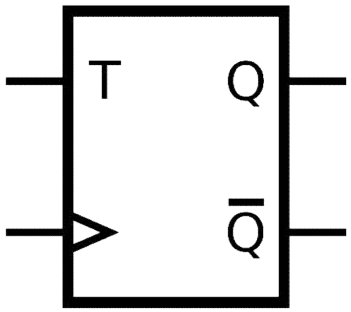
Q(t+1)= JQ(t)' + K'Q(t)

Figure: Characteristic Equation

Q(t)	Q(t+1)	J	k
0	0	0	X
0	1	1	X
1	0	X	1
1	1	x	0

Figure: Excitation Table

T flip-flop



Positive Edge Triggering

Figure: Graphical Symbol

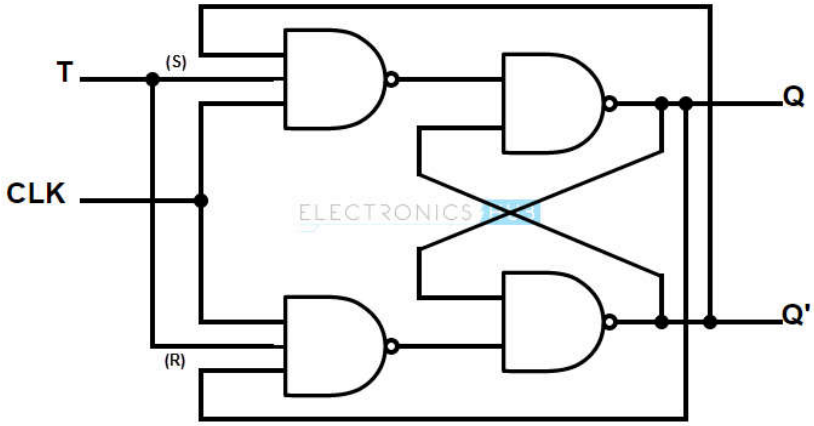


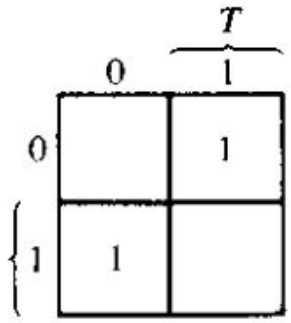
Figure: Logical Diagram

C	T	Q(t+1)	Action
0	X	Q(t) Last State	No Change
↑	0	Q(t)	No Change
↑	1	Q(t)'	Toggle

Figure: Timing Diagram

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

Figure: Characteristic Table



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

Figure: Characteristic Equation

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

Figure: Excitation Table

Characteristic Tables for flip-flops

- The characteristic tables for flip-flops indicate the outputs resulting from each input combination.
- It defines the state of a flip-flop as a function of its inputs and previous state.
- Characteristic table defines the logical property of the flip-flop and completely characterizes its operation.
- Characteristic table allows us to analyze circuits containing flip-flops.

Excitation tables for flip-flops

- While designing sequential circuits, we are given the outputs signals required and have to find the inputs necessary to create these outputs.
- The excitation table of flip-flop provides the necessary inputs to create required outputs (or required change of state).
- Let the variable **Q(t)** represents the Q output prior to the assertion of the clock input and **Q(t+1)** is the output after clock has been asserted.

Excitation Tables for flip-flop

(a) *JK* Flip-Flop

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

(b) *SR* Flip-Flop

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

(c) *D* Flip-Flop

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

(d) *T* Flip-Flop

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

Edge-Triggering D Flip-flop (Master-Slave)

- The construction of D flip-flop with two D latches and inverter is shown below.
- First latch is called the **master** and the second is called the **slave**.

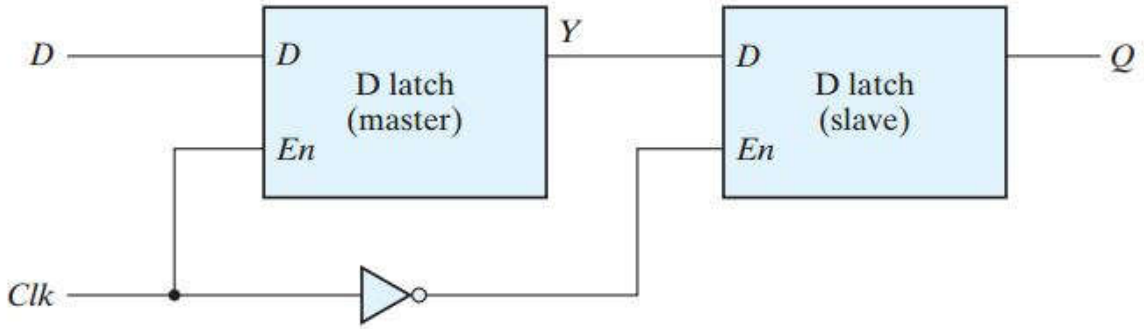


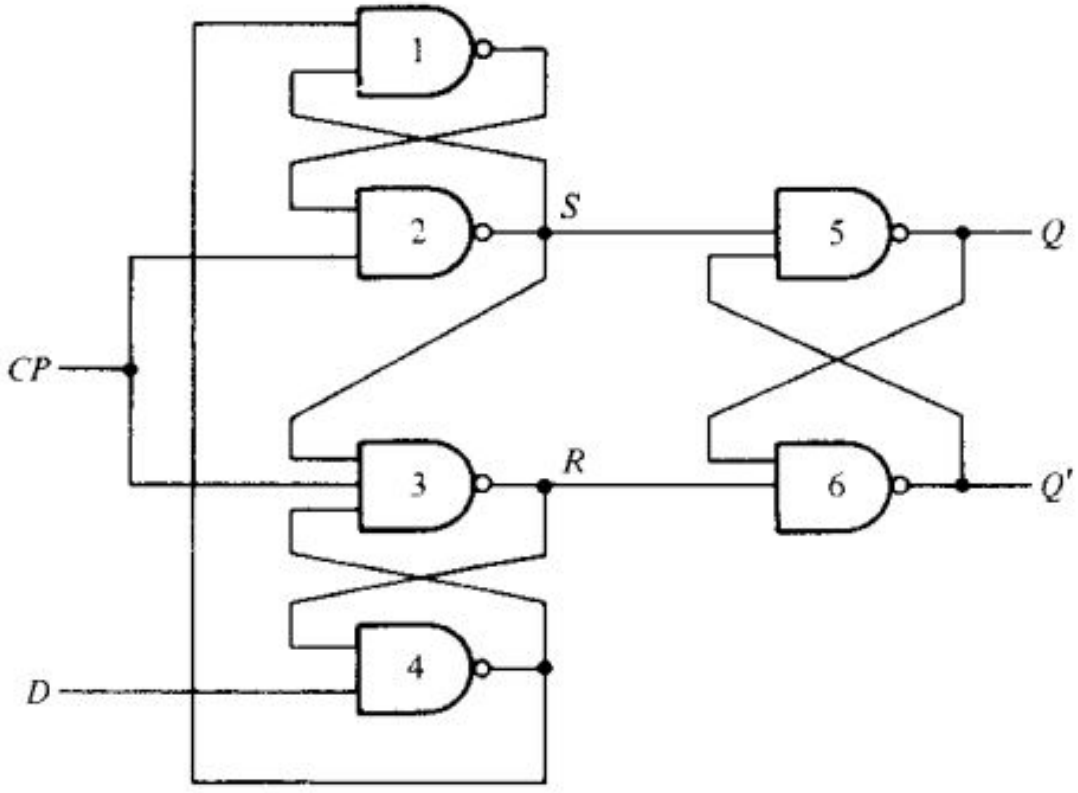
Figure: Master-Slave D flip-flop

Operation

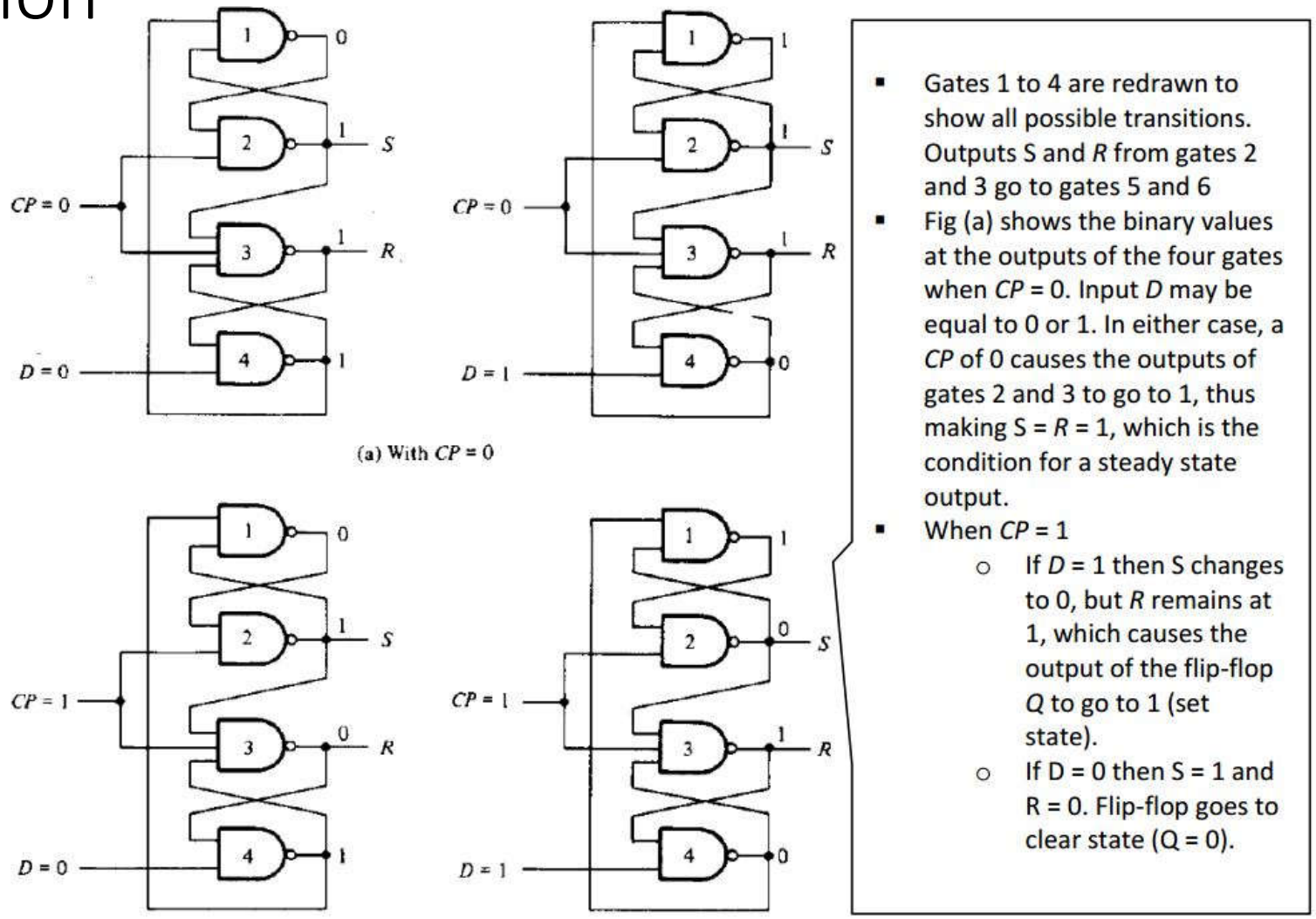
- When the clock is 0, the output of the inverter is 1. The slave latch is enabled, and its output Q is equal to the master output Y. The master latch is disabled because *Clk* =0.
- When the clock pulse changes to logic-1 level, the data from the external D input remains are transferred to the master. The slave, however, is disabled as long as the clock remains at the 1 level.
- *The change in the output of the flop-flop can be triggered only by and during the transition of the clock from 1 to 0. (i.e. triggered by negative edge of clock)*

Alternative method of constructing edge-triggered D flip-flop

- By using three SR latches as shown in figure:



Operation



- 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).

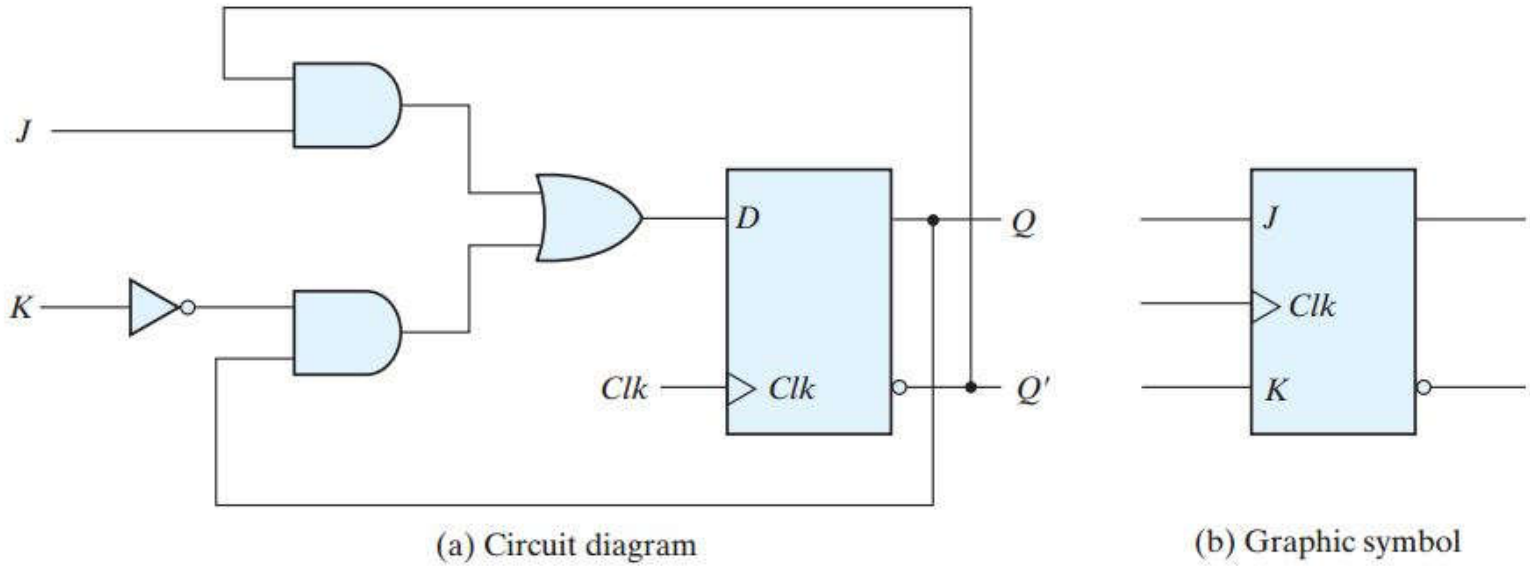
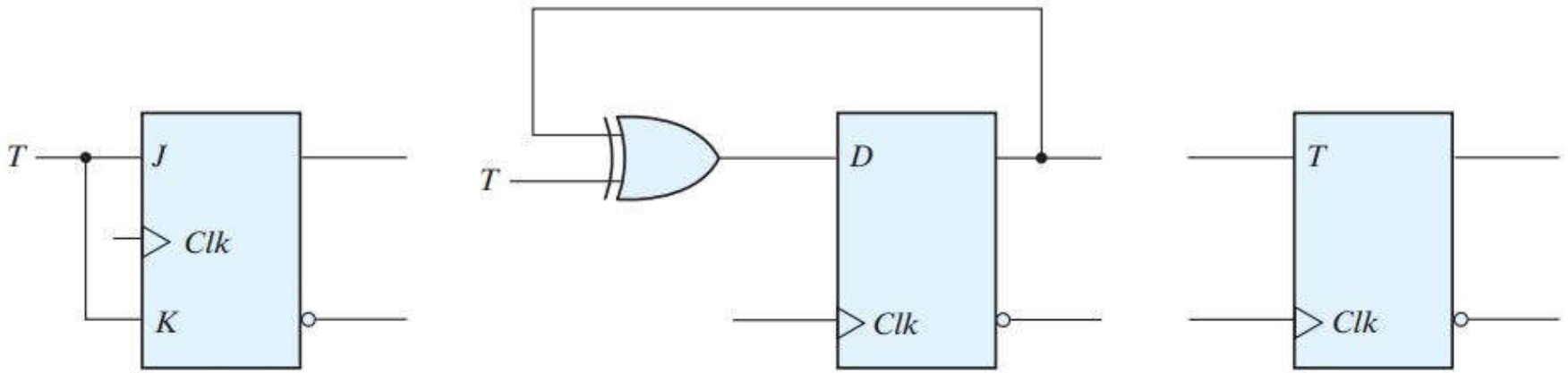


Figure: JK Flip-flop



(a) From *JK* flip-flop (b) From *D* flip-flop (c) Graphic symbol

Figure: T flip-flop

Direct Inputs/Asynchronous Inputs

- Some flip-flops have asynchronous inputs that are used to force the flop-flop to a particular state independently of the clock.
- The input that sets the flip-flop to 1 is called ***preset or direct set***.
- The input that clears the flip-flop to 0 is called ***clear or direct reset***.
- *When power is turned on in a digital system, the state of the flip-flop is unknown. The direct inputs are useful for bringing all flip-flops to a known staring state prior to the clocked operation.*

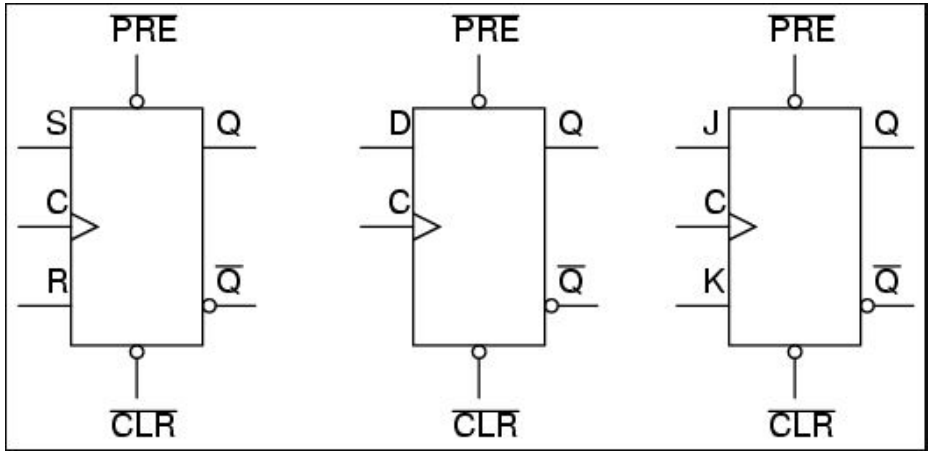


Figure: Direct Inputs Flop-flops Graphical Symbols

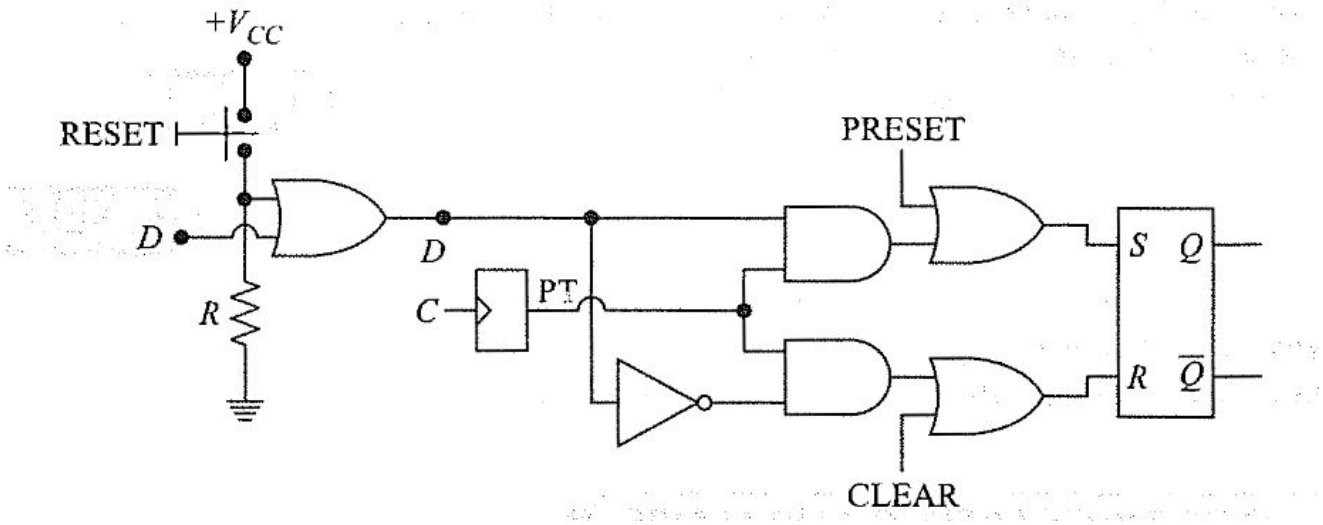


Figure: PRESET and CLEAR Functions

Flip-Flop Timing

- Diodes and transistor cannot switch states immediately. It always takes a small amount of time to turn a diode on or off. Likewise, it takes time for a transistor to switch from saturation to cutoff, and vice versa.
- Switching time is the main cause of **propagation delay** t_p. This represent the amount of time it takes for the output of a gate or flip flop to change states after the input changes. For instance, if the $\mathbf{t_p}$=10 ns for D flip flop, it takes about 10 ns for Q to change states after D has been samples by clock edge. This propagation delay time has been used to construct the "pulse-forming circuit" used with edge-triggered flip-flops.
- The **setup time** t_{setup} is the minimum amount of time that the data bit must be present before the clock edge hits. For instance, if a D flip-flop has a setup time of 15 ns, the data bit to be stored must be at the D input at least 15 ns before the clock edge arrives. Otherwise, the manufacturing does not guarantee correct sampling and storing.
- Furthermore, data bit D has to be held long enough for the internal transistors to switch states. Only after the transistor is assured we can allow data bit D to change. **Hold time** t_{hold} is the minimum amount of time that data bit D must be present after the clock. For example, if t_{setup} = 15 ns, and t_{hold}= 5 ns, the data bit has to be at the D input at least 15 ns before the clock edge arrives and held at least 5 ns after the clock pulse transition.

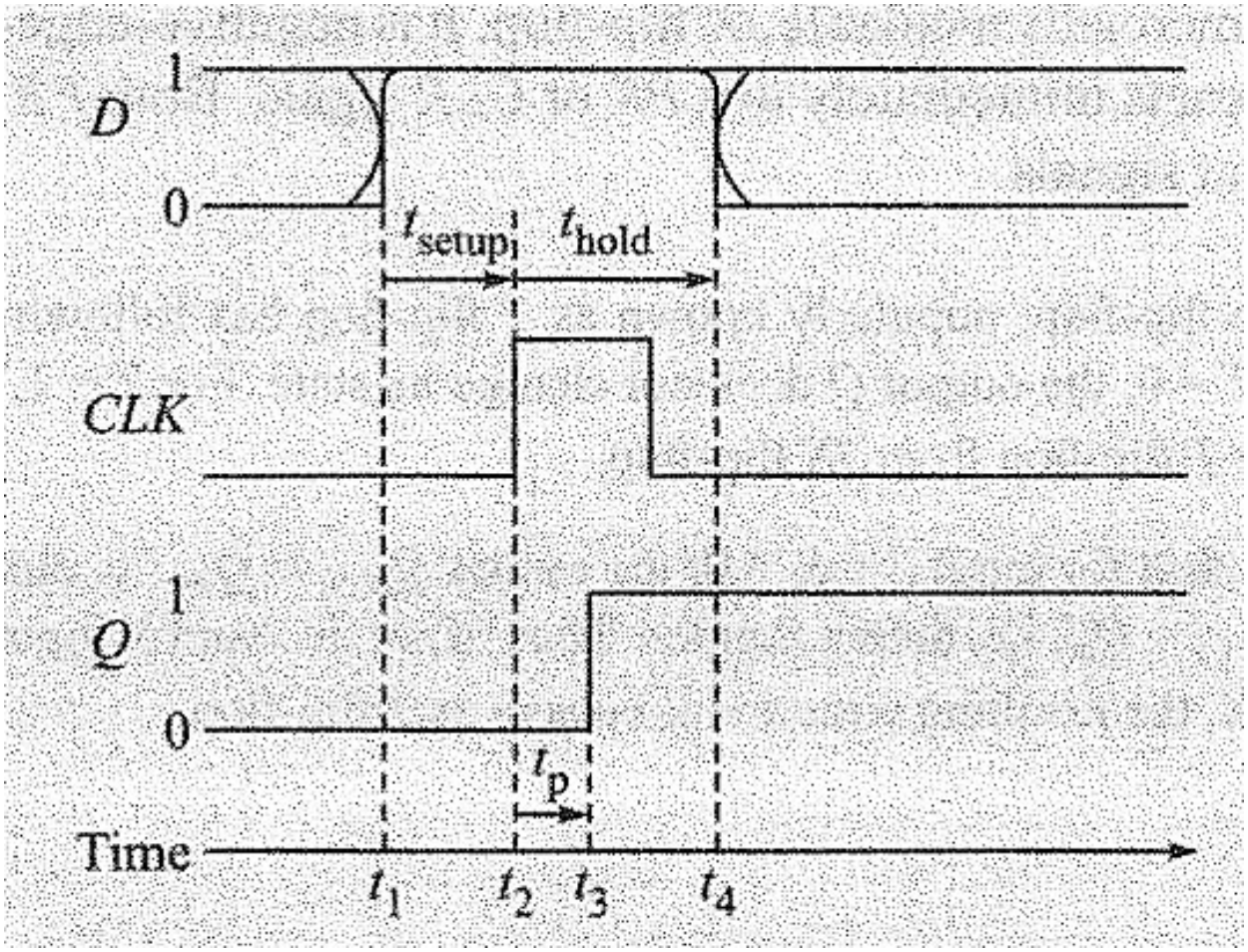


Figure: Flip-flop Timing

JK Master Slave Flip flops

- The master is positive-level triggered and the slave is negative-level triggered. Therefore, the master responds to its J and K inputs before the slave. If J=1 and K=0, the master sets on the positive clock transition. The high Q output of the master drives the J input of the slave, so on the negative clock transition, the slave sets, so on the action of the master.

- If J=0 and K=1, the master resets on the positive transition(PT) of the clock. The high Q output of the master goes to the K input of the slave. Therefore, the negative transition(NT) of the clock forces the slaves to reset. Again, the slave has copied the master.

- If the master's J and K inputs are high, it toggles on the PT of the clock and the slave then toggles on the clock NT. Regardless of what the master does, therefore, the slave copies it; if the master sets, the slaves sets; if the master resets, the slave resets.

- If J=K=0, the flip flop is disabled and Q remains unchanged.

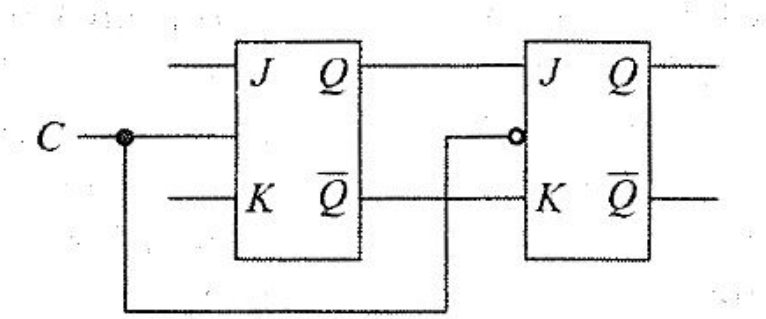
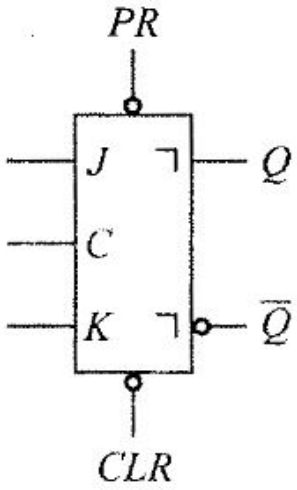


Figure: Master slave JK Flip-Flop



a) Symbol

C	J	K	Q(t+1)	Action
⎍	0	0	Q(t)	No Change
⎍	0	1	L	RESET
⎍	1	0	H	SET
⎍	1	1	$\overline{Q}$(t)	Toggle

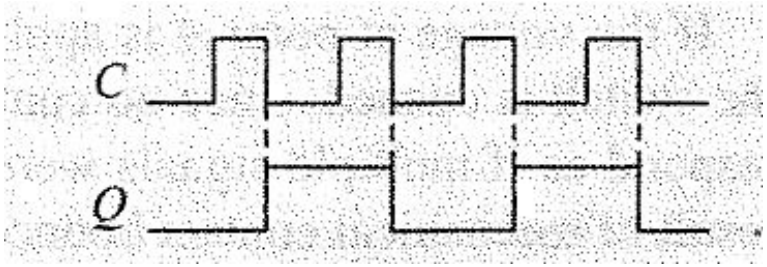
b) Truth Table

Figure: 7476 master JK flip-flop

- Here, C is not edge-triggered. i.e. it is pulse triggered. The master does in fact change state when C goes high. However, while the clock is high, any change in J or K will immediately affect the master flip-flip.

- The symbol ⌝ appearing next to the Q and the Q outputs is the IEEE designation for a ***postponed output***. In this case, it means Q does not change state until the clock makes an NT. In other words, the contents of the master are shifting into the slave on the clock NT, and at this time Q changes state.

Q. The JK master slave flip-flop has its J and K inputs tied to $+V_{cc}$ and a series of pulses (actually a square wave) are applied to its C inputs. Describe the waveform at Q.



- Since J=K=1, the flip-flop simply toggles each time the clock goes low. The waveform at Q has a period twice that of the C waveform. In other words, the frequency of Q is only one-half that of C. This circuit acts as a frequency divider- the output frequency is equal to the input frequency divided by 2.
- The Q changes state on NTs of the clock.

Switch Contact Bounce Circuits

- In nearly every digital system there will be occasion to use mechanical contacts for the purpose of conveying an electrical signal; example of this are the switched used on the keyboard of a computer system,
- In each case, the intent is to apply a high logic level (usually +5 Vdc) or a low logic level (0 Vdc). The single-pole-single-throw (SPST) switch is one such example.

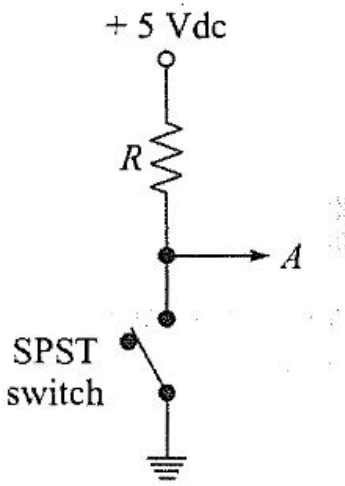


Figure: SPST Switch

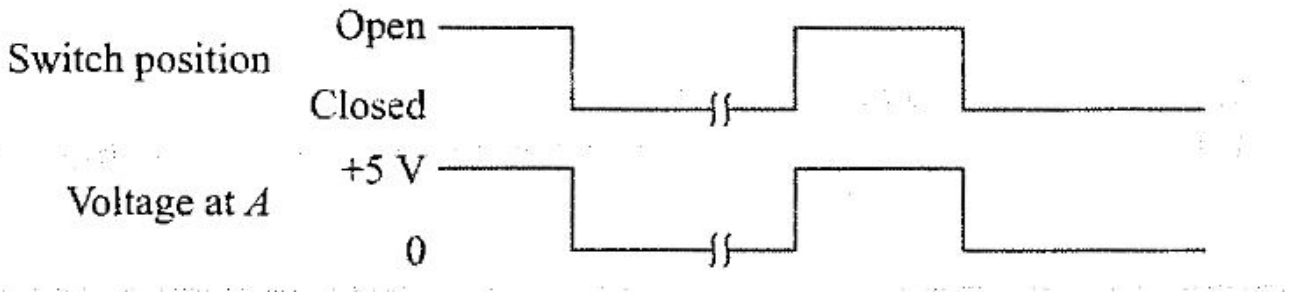


Figure: Ideal Waveform at A when switch is moved form open to closed, or vice versa

- In actuality, the waveform at point A appear more or less as shown in figure below as a result of a phenomenon known as **contact bounce**.
- Any mechanical switching device consists of moving contact arm restrained by some sort of a spring system. As a result, when the arm is moved from one stable position to the other, the arm bounces, much as hard ball bounces when dropped on a hard surface.
- The number of bounces that occur and the period of the bounce differ for each switching device.
- Contact bounce appears in the voltage level at point A only when the switch is closed.
- Need some sort of electronic circuit to eliminate the contact bounce problem

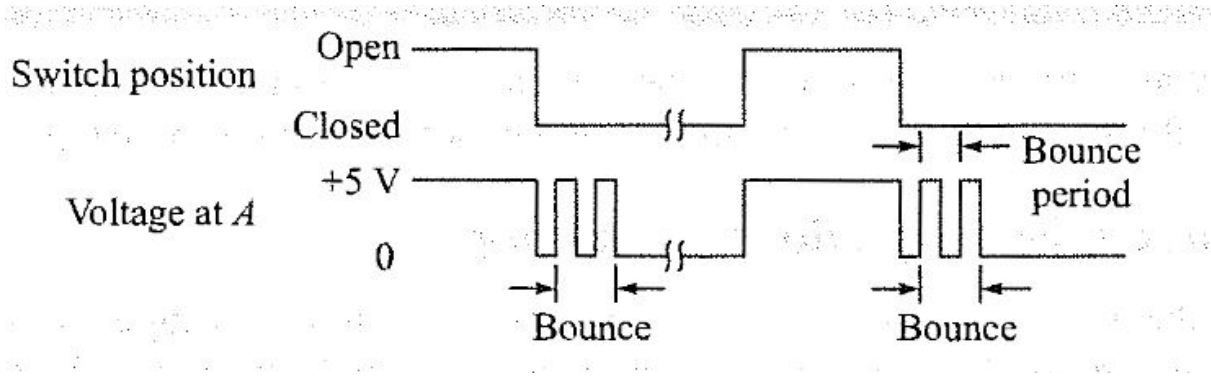
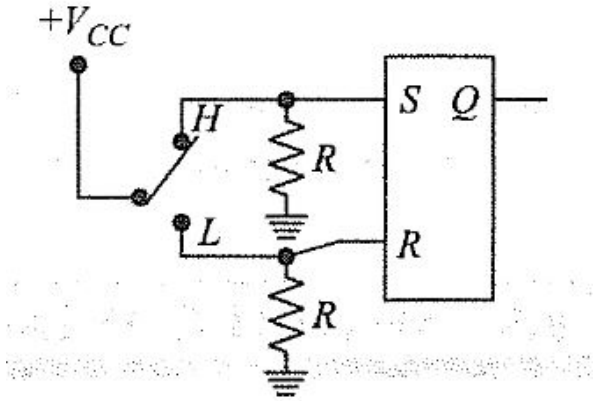


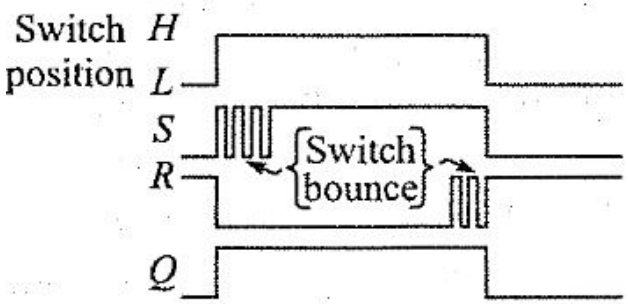
Figure: Voltage at A showing contact bounce

A simple RS latch Debounce Circuit

- The RS latch will remove any contact bounce due to the switch. The output, Q is used to generate the desired switch signal.
- When the switch is moved to position H, R=0 and S=1. bouncing occurs at the S input due to the switch. The flip-flop "sees" this as a series of high and low inputs, setting with a high level. The flip-flop will immediately be set with Q=1 at the first high level on S. When the switch bounces, losing contact, the input signals are R=S=0, therefor the flip-flop remains set (Q=1). When the switch regains contact, R=0 and S=1; this causes an attempt to again the set the flip-flop. But since the flip-flop is already set, no changes occur at Q. The result is that the flip-flop responds to the first, and only to the first, high level at its S input, resulting in a "clean" low-to-high signal at its output (Q).
- When the switch is moved to position L, S=0 and R=1. Bouncing occurs at the R input due the switch. Again, the flip-flop "sees" this as a series of high and low inputs. It simply responds to the first high level, and ignores all following transitions. The result is a "clean" high-to-low signal at the flip-flop output.



a) : Switch contact bounce eliminator



b) Switch bounce

Figure: Debounce Circuit

Various Representation of Flip-Flops

- Characteristic Equations of Flip-flops
- Flip-flops as Finite State Machine
- Flip-flop Excitation Table

Flip-flop as Finite State Machine

- In sequential logic circuits, the value of all the memory elements at a given time defines the state of that circuit at that time.
- FSM concept offer a better alternative to truth table in understanding progress of sequential logic with time.
- For a complex circuit, a truth table is difficult to read as its size becomes too large.
- In FSM, functional behavior of the circuit is explained using finite number of states.
- State transition diagram is a very convenient tool to describe an FSM.

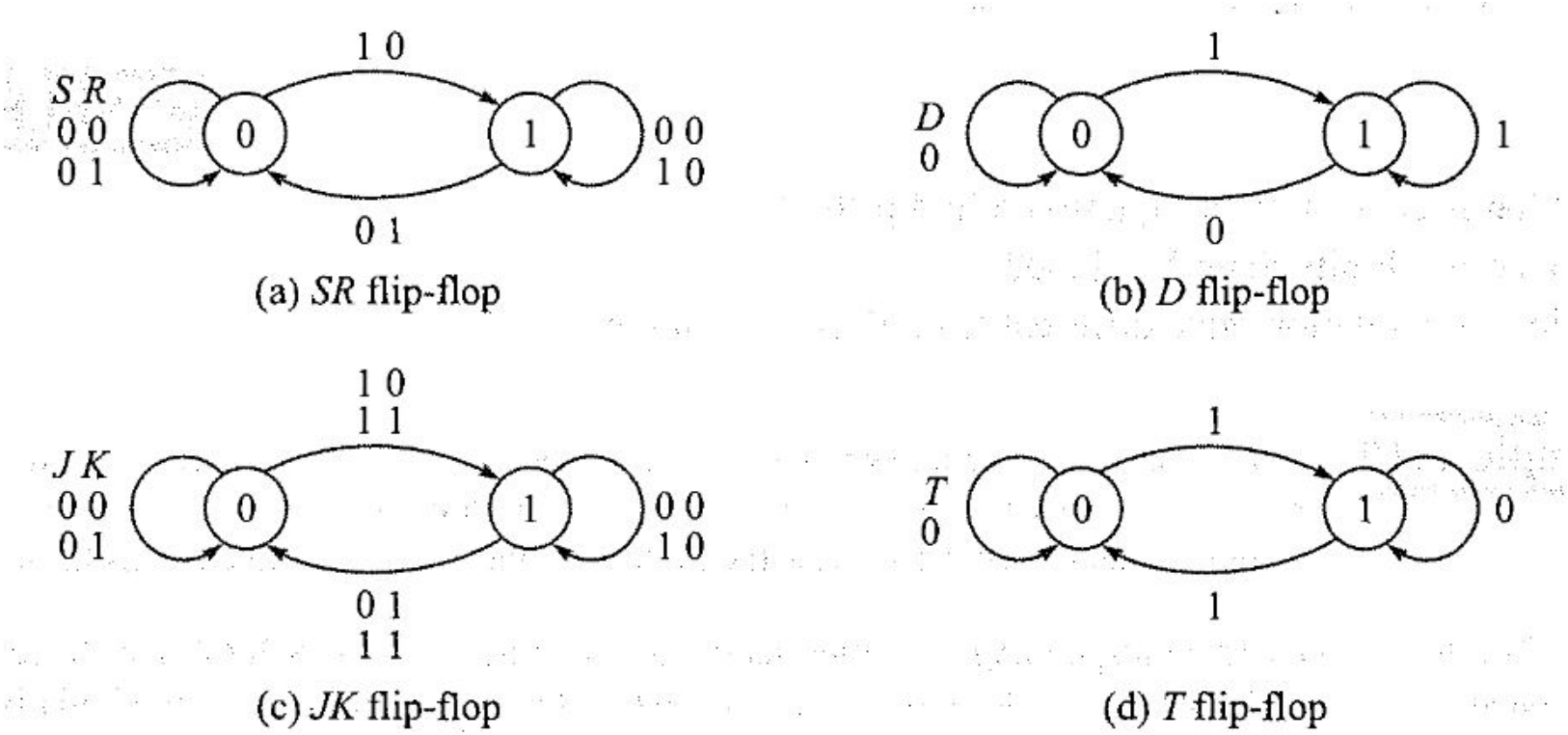


Figure: Representation of flip-flops as FSM through their state diagram

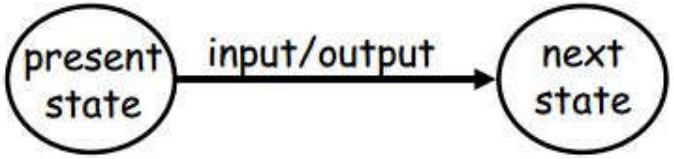
Design Procedure

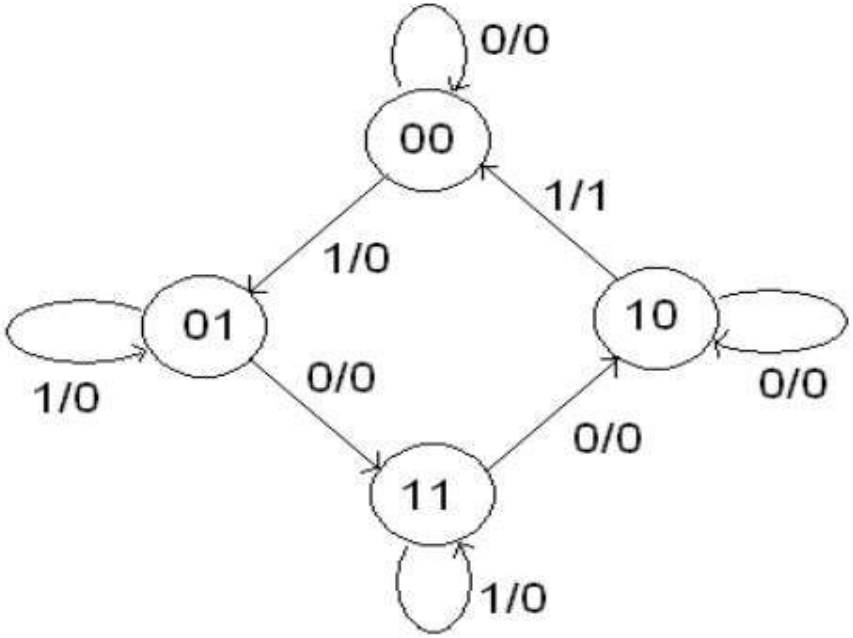
- Derive a state diagram
- Reduce the number of states if applicable
- Assign binary values to the states
- Obtain binary coded state table/transition table
- Choose the type of flip-flop to be used
- Derive simplified flip-flop input equations and output equations
- Draw the logic diagram

State Diagram

- The State transition diagram is a graphical representation of the machine changes from one state to another in reaction to the input(s).
- In state diagram, a state is represented by a circle, and the transition between states is indicated by direct lines connecting the circles.
- The name or value (binary number) inside each circle identifies the state the circle represents.
- The directed lines are labeled with two binary numbers separated by a /.
- The input value that causes the state transition is labelled first; the number after the symbol / gives the value of the output during the present state.

Example of state diagram





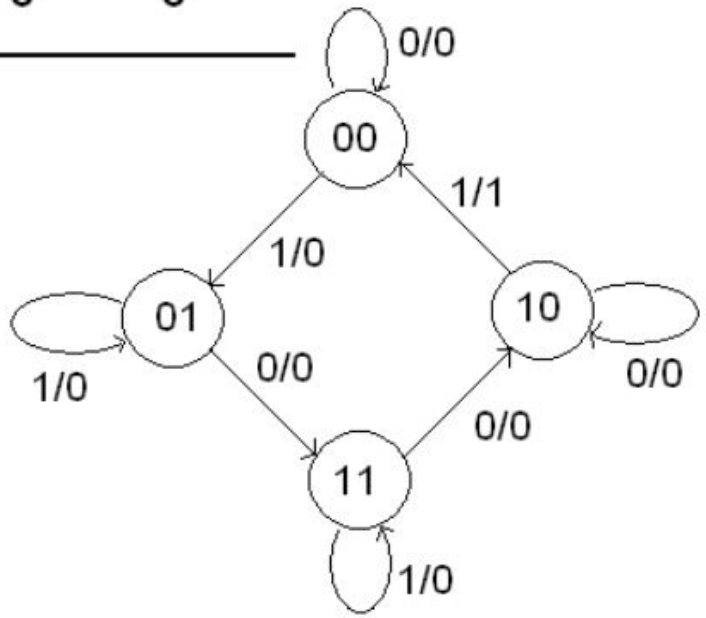
State Table/Transition Table

- The time sequence of inputs, outputs and states can be represented in a tabular form called as a state table or transition table.
- The state table represents the changes of machine from one state to another in reaction to the inputs.
- A state transition table is a systematic method of exploring or analyzing the behavior of a discrete machine or object that exhibits history dependent behavior.
- A state table is essentially a truth table in which inputs are the current state with input lines, and the outputs include the next state along with other outputs.
- The state table provides the same information as the state diagram i.e. there is no difference between a state table and state diagram except in the manner of representation.

Parts of transition table

- **Present state**: the present state designates the states of flip-flops before the occurrence of a clock pulse.
- **Next state**: the next state shows the states of flip-flop after the application of a clock pulse.
- **Output**: output section lists the values of the output variables during the present state.
- Both the next state and output sections have different columns of different values of inputs.

	Next State		Output	
Present State	x=0	x=1	x=0	x=0
AB	AB	AB	y	y
00	00	01	0	0
01	11	01	0	0
10	10	00	0	1
11	10	11	0	0



example

- 3 bit binary up counter.

State Machine Model

1. Mealy Model

- State machine produces an output for each transition
- Output is the function of present state and input

2. Moore Model

- State machine produces an output for each state
- Output is the function of state only.

State Machine

- A state machine, also called, a sequential machine, is a system that can be described in terms of a set of states that the system may enter.
- Once in a particular state, the system should be capable of remaining in that state for some finite period of time even if the system input change.
- The state may be updated at each clock edge otherwise the state remains unchanged.

Analysis of Clocked Sequential Circuits

Process

1. Logical Diagram (given) ⟶
2. Flip-flop input equations
3. Flip-flop Characteristic equations
4. State Equations/Transition Equations
5. State Table
6. State Diagram

- Figure with D flip-flop
- Figure with T flip-flop
- Figure with JK flip-flop
- Figure with SR flip-flop

A **state equation** is an algebraic expression that specifies the condition for a flip-flop state transition.

Example: Analysis with D flip-flop

Class room activities

Steps in obtaining state equation and next state

1. Determine the flip-flop input equations in terms of the present state and input variables
2. Substitute the input equations into the flip-flop characteristic equation to obtain the state equations.
3. Use corresponding state equation to determine the next-state values in the state table

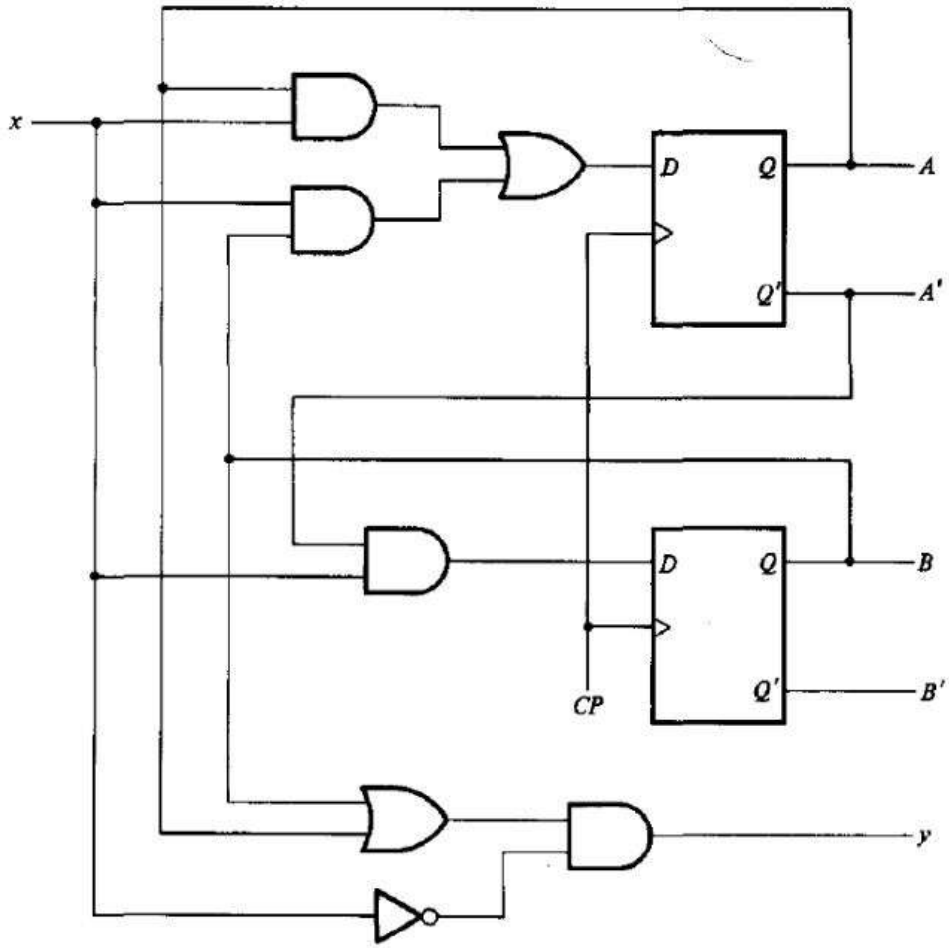
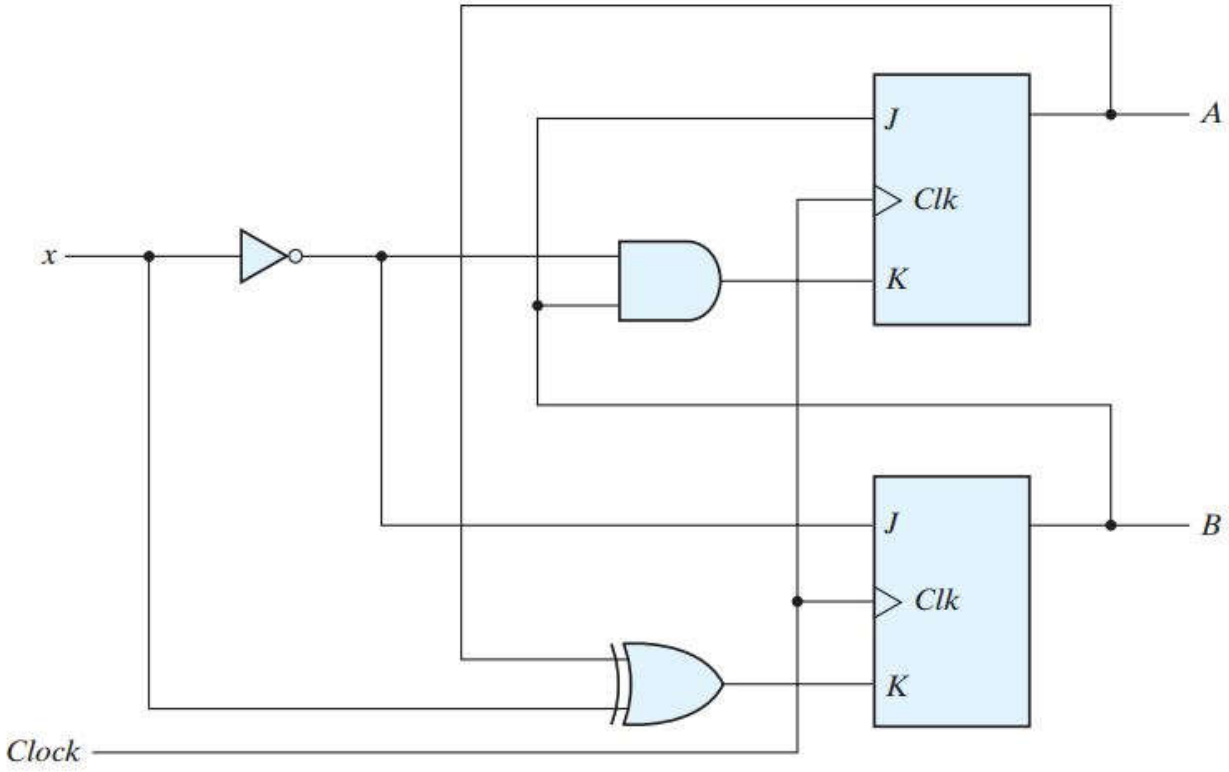
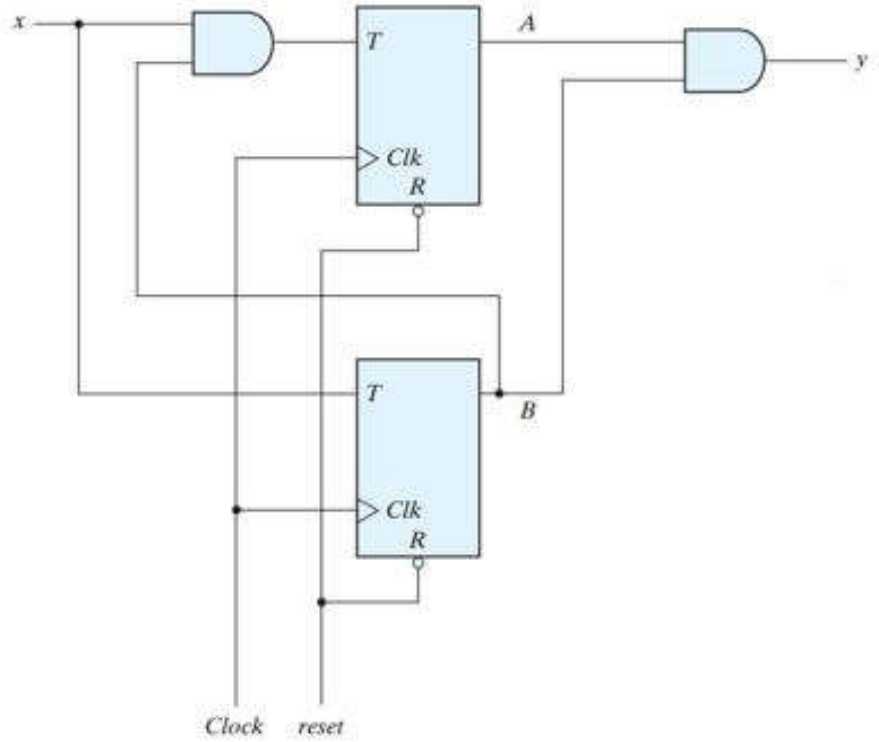


Fig: Example of sequential circuit

Example : Analysis with JK flip-flop



Example: Analysis with T flip-flop



(a) Circuit diagram

State Reduction and Assignment

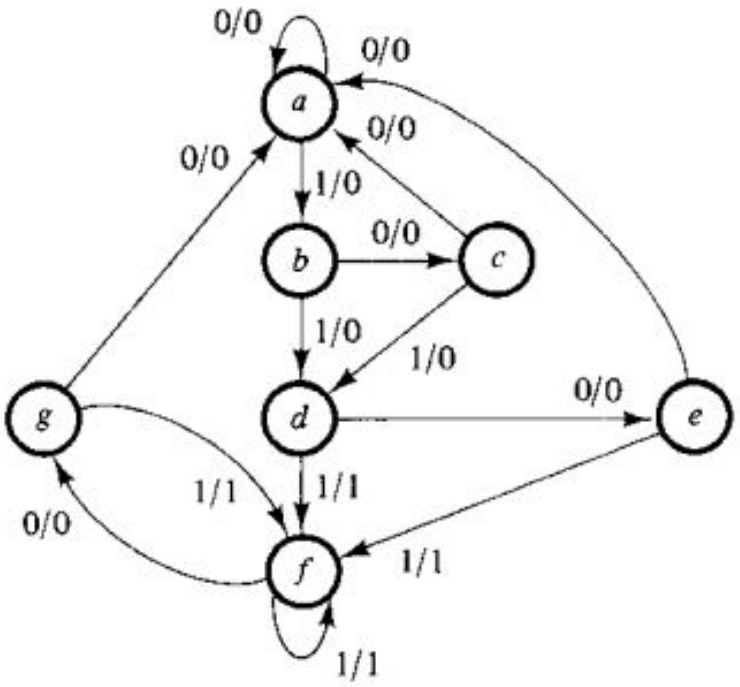


Figure: State Diagram

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	g	f	0	1
g	a	f	0	1

Figure: State Table

- Two states are said to be equivalent, if for each member of the set of inputs, they give exactly the same input and send the circuit either to the same state or to and equivalent state.
- When two states are equivalent, one of them can be removed without altering the input-output relationships.
- In above state table, **e** and **g** states are equivalent; one can be removed. Replace g with e.

Reducing the State Table

	Next State		Output	
Present State	$x = 0$	$x = 1$	$x = 0$	$x = 1$
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

- After replacing **g** with **e** states, **f** and **d** states are equivalent. States **f** is then removed replacing by **d**.

Reduced State Table

	Next state		Output	
Present State	$x = 0$	$x = 1$	$x = 0$	$x = 1$
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 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

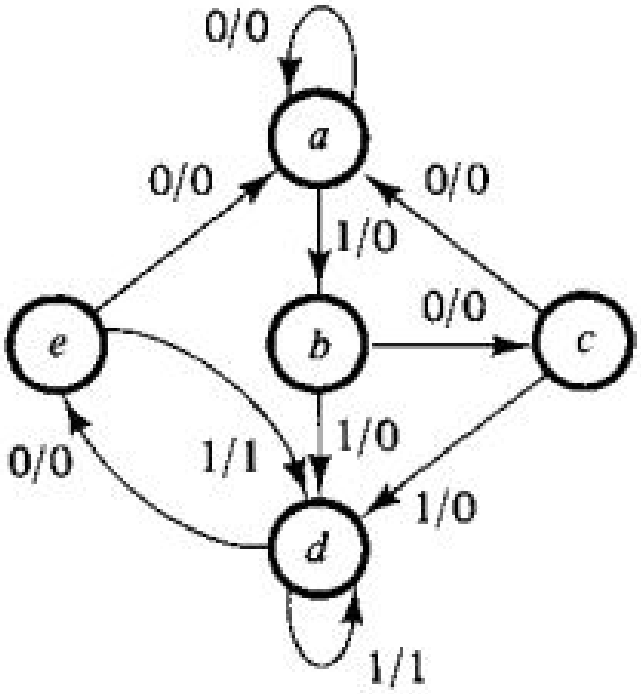


Figure: Reduced State Diagram