Unit-3 Clocks and Flip Flops

Clock waveforms, TTL clock, RS Flip Flops, Gated flip-flops, Edge triggered RS Flip-Flops, Edge triggered D Flip-Flops, and Edge triggered JK Flip-Flops, JK master slave Flip Flops, various representations of Flip Flops.

Types of Digital Circuits

- Digital logic circuits are often known as switching circuits, because in digital circuits the voltage levels are assumed to be switched from one value to another value instantaneously. These circuits are termed as logic circuits, as their operation obeys a definite set of logic rules.
- **Combinational logic circuit**
- **Sequential digital logic circuits**

Definition of Combinational & Sequential circuits

- **Combinational circuits** are defined as the time independent circuits which do not depends upon previous inputs to generate any output are termed as combinational circuits. **Sequential circuits** are those which are dependent on clock cycles and depends on present as well as past inputs to generate any output.
- **Examples –** Encoder, Decoder, Multiplexer, Demultiplexer
- **Sequential circuits** are those which are dependent on clock cycles and depends on present as well as past inputs to generate any output.
- Flip-flops, Counters

Characteristics of Combinational logic circuits

- In this output depends only upon present input.
- Speed is fast.
- It is designed easy.
- There is no feedback between input and output.
- This is time independent.
- Elementary building blocks: Logic gates
- Used for arithmetic as well as boolean operations.
- Combinational circuits don’t have capability to store any state.
- As combinational circuits don’t have clock, they don’t require triggering.
- These circuits do not have any memory element.
- It is easy to use and handle.

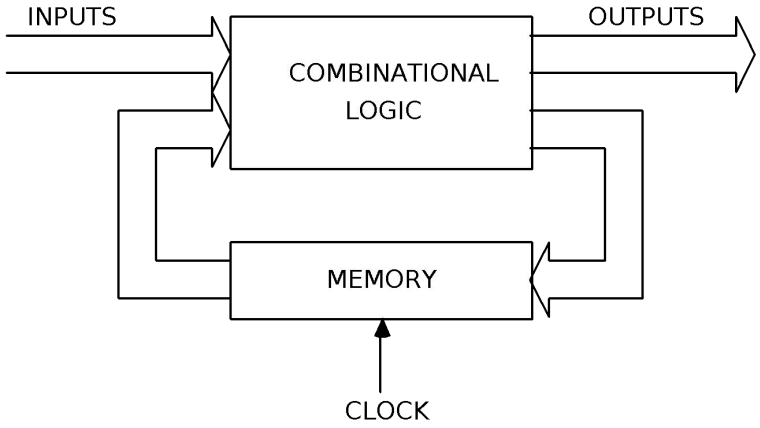
Characteristics of Sequentialal logic circuits

- In this output depends upon present as well as past input.
- Speed is slow.
- It is designed tough as compared to combinational circuits.
- There exists a feedback path between input and output.
- This is time dependent.
- Elementary building blocks: Flip-flops
- Mainly used for storing data.
- Sequential circuits have capability to store any state or to retain earlier state.
- As sequential circuits are clock dependent they need triggering.
- These circuits have memory element.
- It is not easy to use and handle.

Sequential Logic Circuits

- Any sequential logic system can be broken down into two sections:
- Memory part
- combinational part

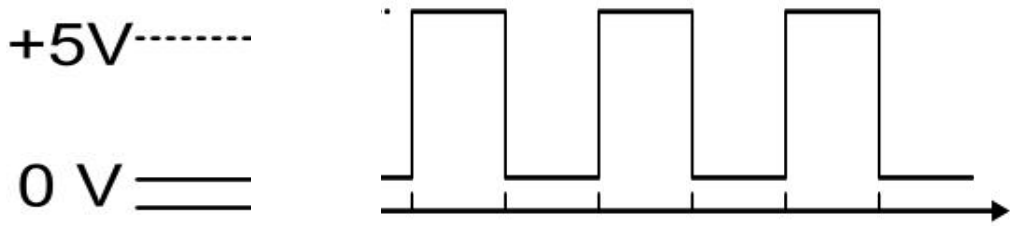
- The combinational part will comprise the usual AND, OR gates etc.
- The memory part is usually implemented with bistable devices.
- Two basic types: latches and flip-flops.



Clock Waveforms

- The heart of every digital system is the system clock. The digital system signals are not static digital logic The clock signal advances the sys tem logic through sequence of steps.
- Ideal Clock Waveform

The ideal clock waveform is as shown below:



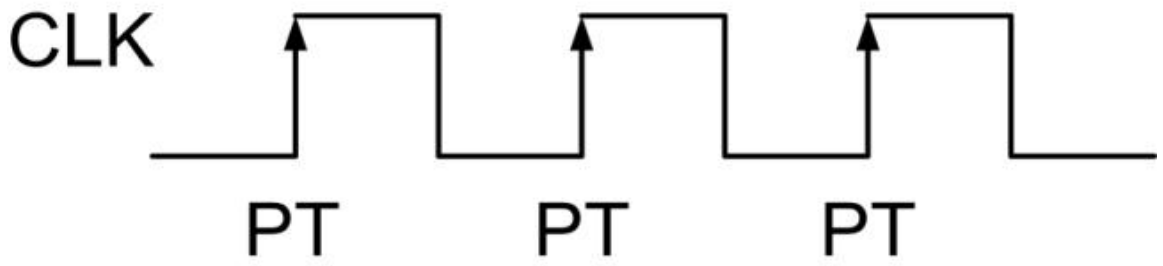
Synchronous Operation

- Most of the digital systems change states in synchronism with the system clock. A change of state will either occur as the clock makes a tr ansition from LOW to HIGH or as it makes a transition from HIGH to LOW.
- Positive Transition

The LOW-to-HIGH transition is called the positive transition (PT). The PT is given emphasis by drawing a small arrow on the rising edge of th e clock waveform as shown in figure.

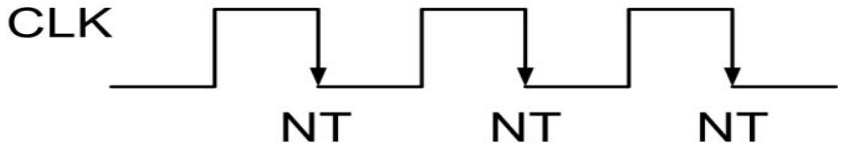
Positive Transitions

- **The LOW-to-HIGH transition is called the positive transition (PT). The PT is given emphasis by drawing a small arrow on the rising edge of the clock waveform as shown in figure below:**
- **A circuit that changes state at this time is said to be positive-edge-triggered.**



Negative Transition

- **The HIGH-to-LOW transition is called the negative transition (NT). The NT is given emphasis by drawing a small arrow on the falling edge of the clock waveform as shown in figure below:**



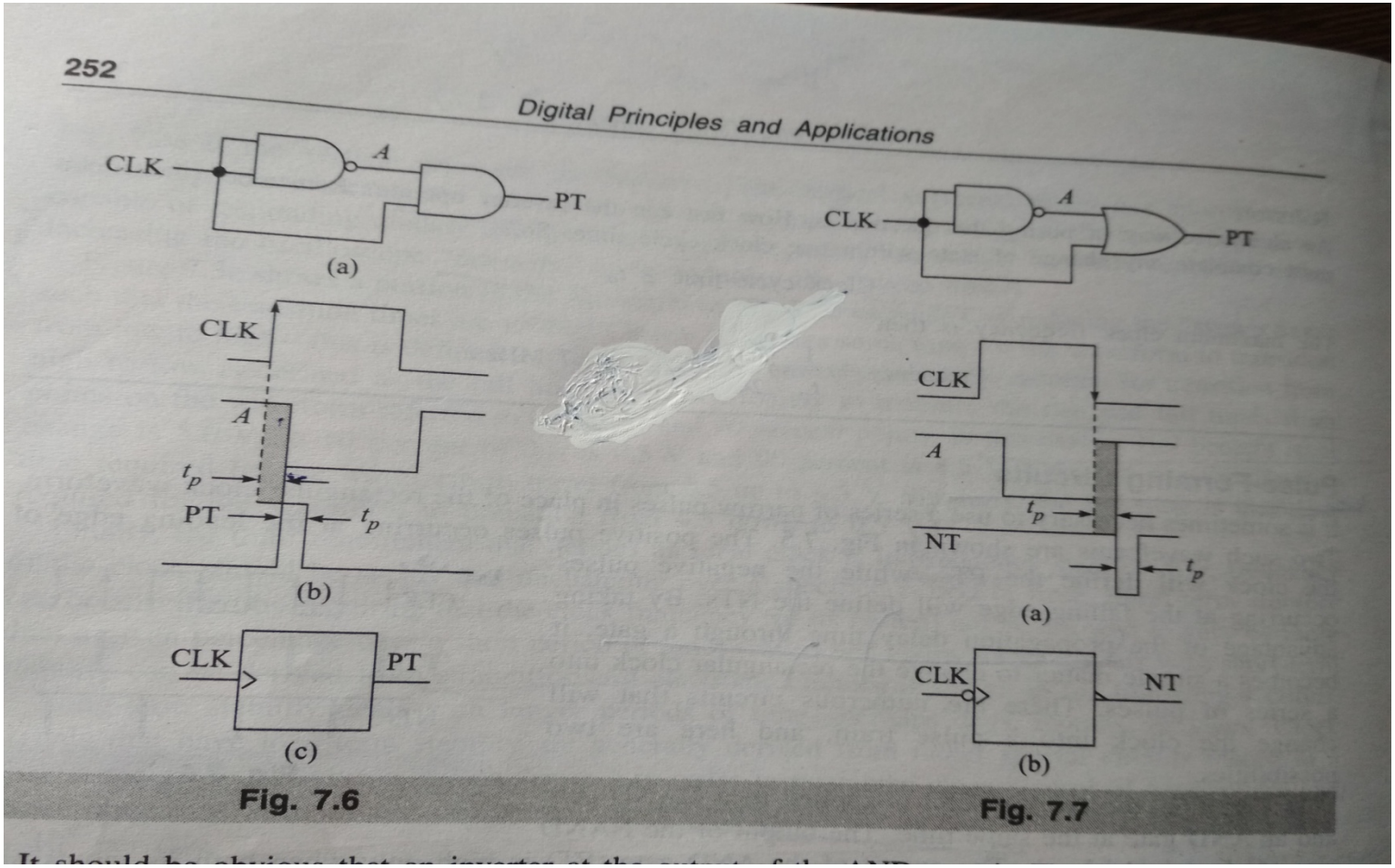
- **A circuit that changes state at this time is said to be negative-edge-triggered.**
- **Example: What is the clock cycle time for a system that uses a 1 MHz clock?**

Solution: The clock cycle time $T = 1/f$

$$1/(1\times10^{6}) = 1\times10^{-6} \text{ sec}$$

$$1\ \mu s$$

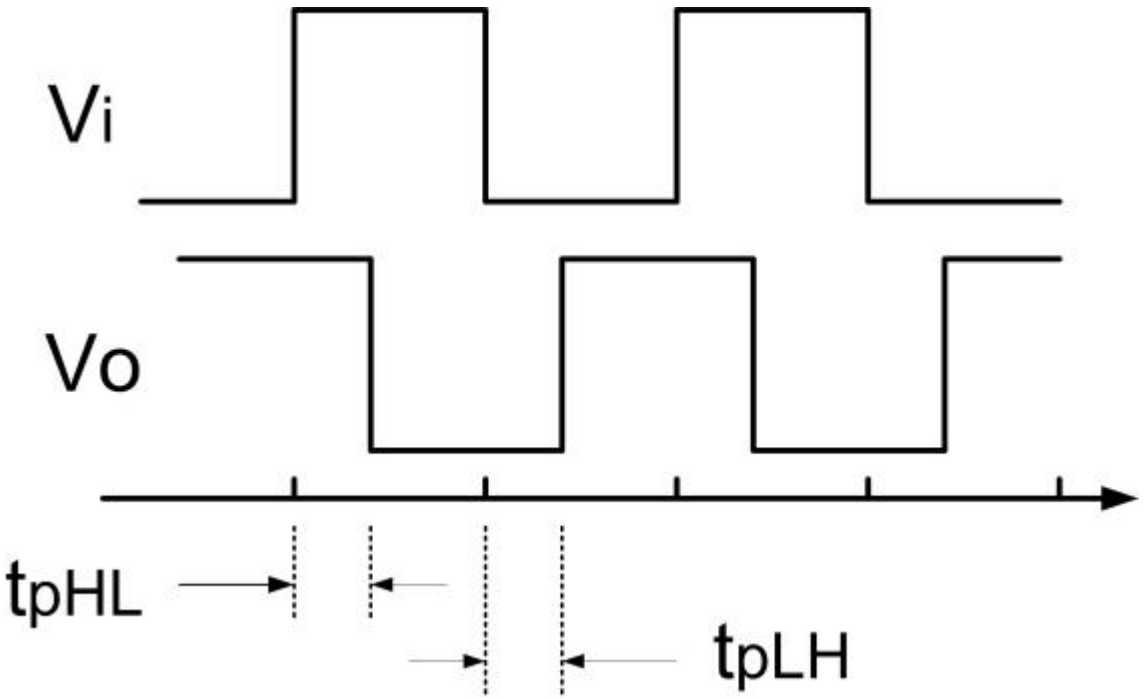
Pulse Forming Circuits



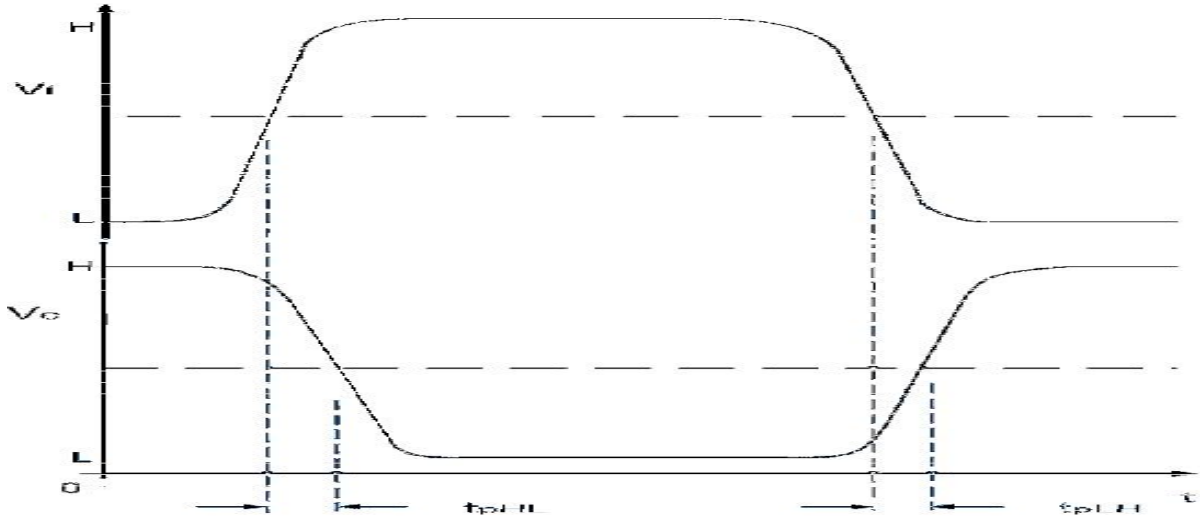
Characteristics of an Ideal Clock

- **Clock levels must be absolutely stable.** When the clock is HIGH, the level must hold a steady value of +5 V. When the clock is LOW, the level must be an unchanging 0 V.
- **Clock should have zero transition time.** The frequency of the clock should be steady and unchanging over a specified period of time.
- **Propagation Delay Time :** Propagation delay t_p is the time between a positive transition (PT) / negative transition (NT) at the input of a digital circuit and the resulting change at the output. As shown in figures below, t_{pHL} is the delay time when the output is making a transition from HIGH to LOW. t_{pLH} is the delay time when the output is making a transition from LOW to HIGH. Usually $t_{pHL} > t_{pLH}$. We can assume $t_p = t_{pHL} = t_{pLH}$ or $t_p = (t_{pHL} + t_{pLH}) / 2$.

Ideal waveforms:



- **For all practical purposes, the time difference between 50% level of the input and corresponding output waveforms is used to calculate propagation delay.**



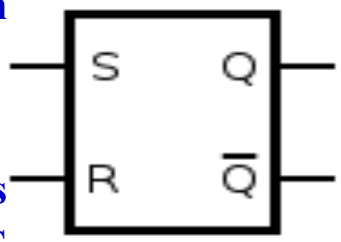
Latches

- **A latch changes state when the input states change. This property is known as transparency. Data is stored by the latch when an enable input (if provided) is valid.**

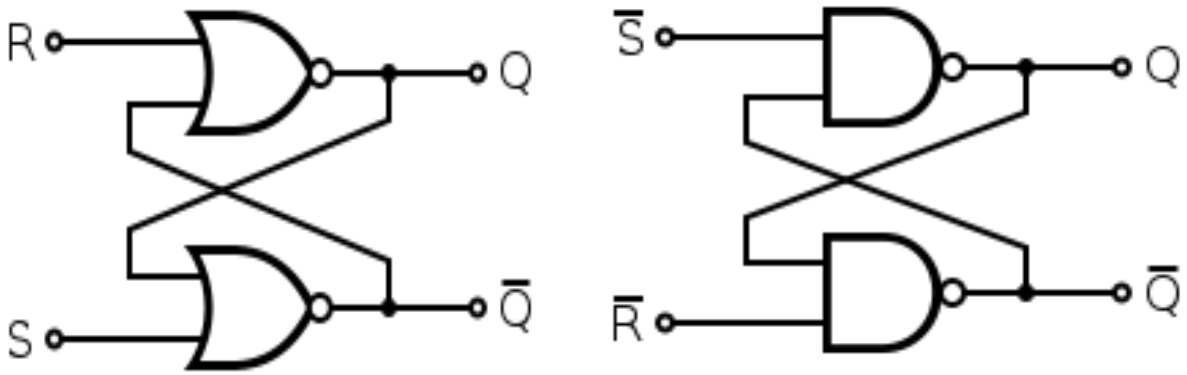
- **A flip-flop however can only change state on the application of a clock pulse.**
- **A purely sequential system is one in which the output is determined by the order in which the input signals are applied.**
- **Sequential circuits are classified as** asynchronous **(unclocked) or** synchronous **(clocked).**

The Basic Latch

- The basic latch has two stable states that correspond to a logic 1 and logic 0. The output changes from one state to another depending on the inputs.
- Consider the simple latch shown below.
- An SR latch (Set/Reset) is an asynchronous device: it works independently of control signals and relies only on the state of the S and R inputs. In the image we can see that an SR latch can be created with two NOR gates that have a cross-feedback loop. SR latches can also be made from NAND gates, but the inputs are swapped and negated. In this case, it is sometimes called an SR latch.



An SR latch made from two NOR and two NAND gates.



S	R	Q	State
0	0	Previous State	No change
0	1	0	Reset
1	0	1	Set
1	1	?	Forbidden

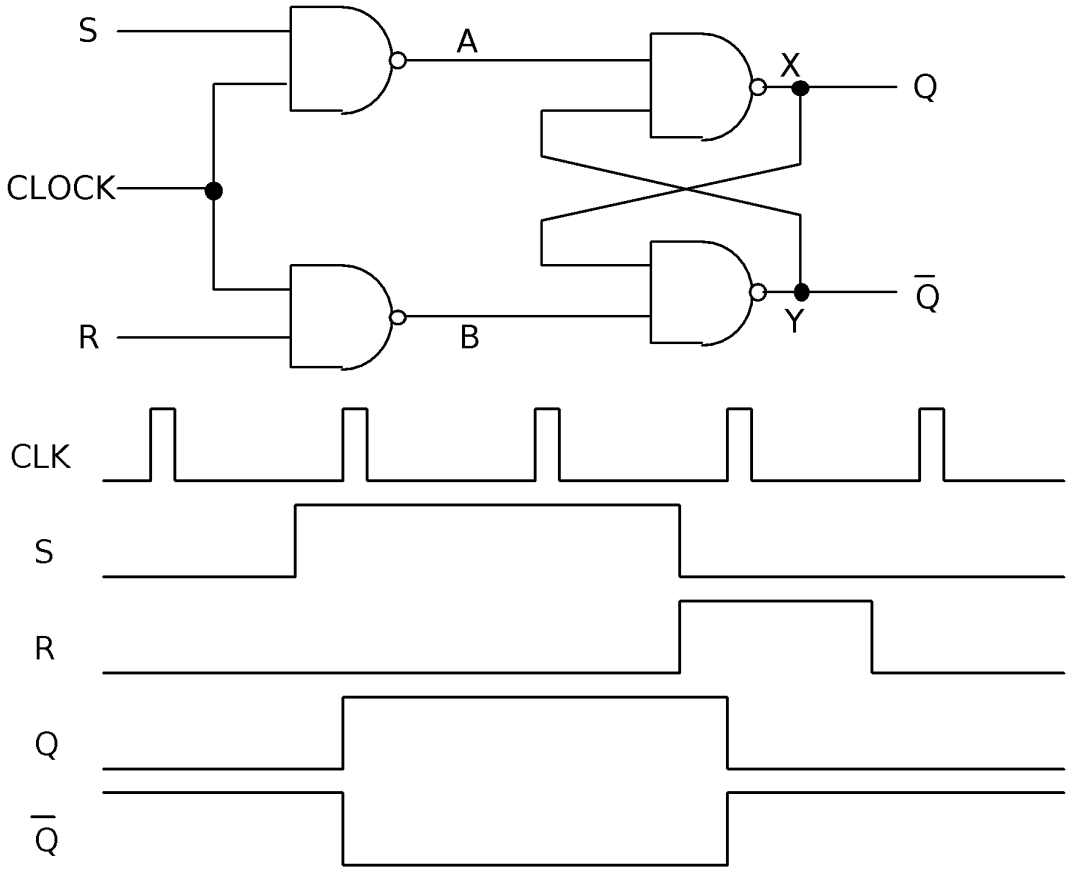
Continued…

- When a high is applied to the *Set* line of an SR latch, the *Q* output goes high (and Q low). The feedback mechanism, however, means that the *Q* output will remain high, even when the *S* input goes low again. This is how the latch serves as a memory device. Conversely, a high input on the *Reset* line will drive the *Q* output low (and *Q* high), effectively resetting the latch's "memory". When both inputs are low, the latch "latches" – it remains in its previously set or reset state.
- When both inputs are high at once, however, there is a problem: it is being told to simultaneously produce a high *Q* and a low *Q*. This produces a "race condition" within the circuit - whichever flip flop succeeds in changing first will feedback to the other and assert itself. Ideally, both gates are identical and this is "metastable", and the device will be in an undefined state for an indefinite period. In real life, due to manufacturing methods, one gate will always win, but it's impossible to tell which it will be for a particular device from an assembly line. The state of $S = R = 1$ is therefore "illegal" and should never be entered.

Flip-Flops(1 bit memory)

- Sequential logic is a form of binary circuit design that employs one or more inputs and one or more outputs, whose states are related by defined rules that depend, in part, on previous states. Each of the inputs and output(s) can attain either of two states: logic 0 (low) or logic 1 (high).
- A common example of a circuit employing sequential logic is the flip-flop, also called a bistable gate. A simple flip-flop has two stable states. The flip-flop maintains its states indefinitely until an input pulse called a trigger is received. If a trigger is received, the flip-flop outputs change their states according to defined rules, and remain in those states until another trigger is received.
- Flip-flops, also called bistable gates, are digital logic circuits that can be in one of two states. Flip-flops maintain their state indefinitely until an input pulse called a trigger is received. When a trigger is received, the flip-flop outputs change state according to defined rules and remain in those states until another trigger is received. Flip-flop circuits are interconnected to form the logic gates for the digital integrated circuits (IC s) used in memory chips and microprocessors. Flip-flops can be used to store one bit, or binary digit, of data. The data may represent the state of a sequencer, the value of a counter, an ASCII character in a computer's memory or any other piece of information.

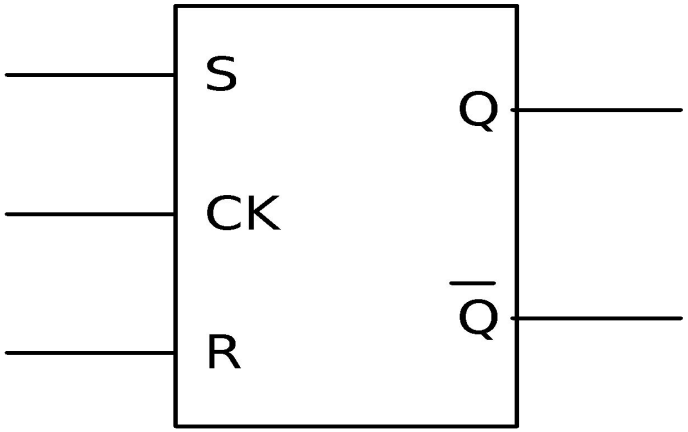
The Gated SR Latch



Symbol & Truth table

The truth table is now:

Symbol



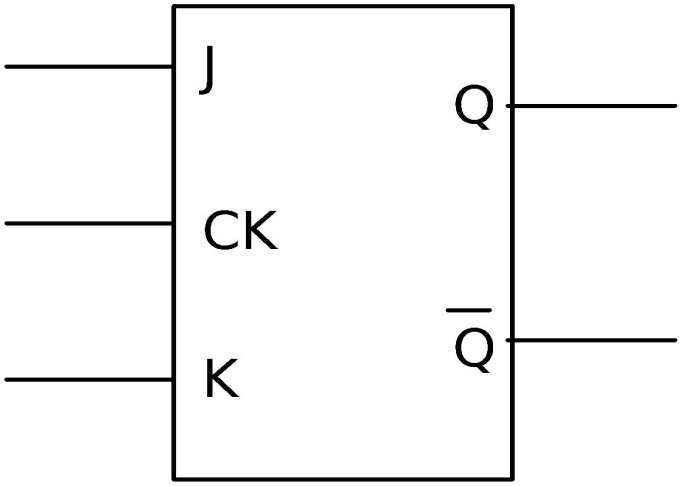
Truth table

R	S	Clk	Q	/Q	
X	X	0	X	X	Same as A = B = 1
0	0	1	X	X	Same as A = B = 1
0	1	1	1	0	
1	0	1	0	1	
1	1	1	1	1	

The JK Flip-Flop

- **The JK flip-flop is basically an SR gated latch with modifications to eliminate the problems of the final indeterminate state. When the two inputs J and K are both 1 then the output Q changes state. It is said to toggle under these conditions.**

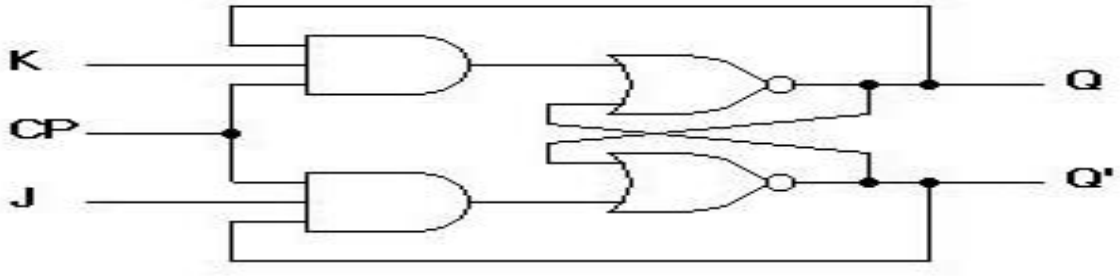
Circuit Symbol:



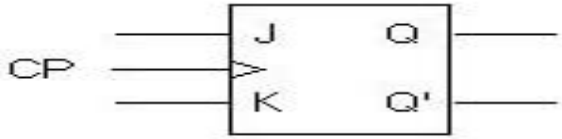
Truth Table

J	K	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	$\overline{Q}_0$ (toggles)

Clocked J-K Flip Flop



(a) Logic diagram



(b) Graphical symbol

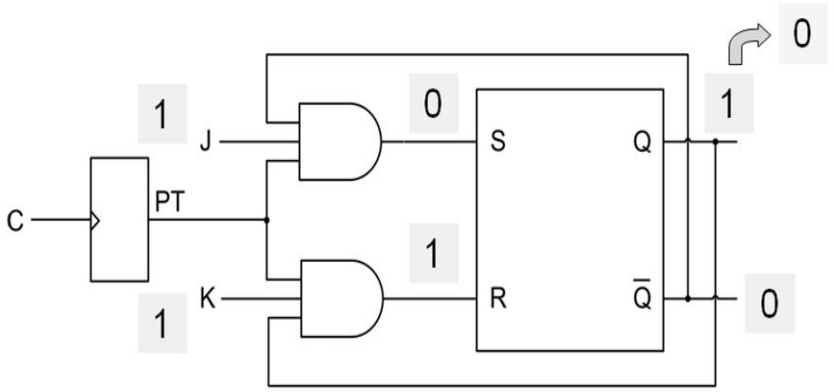
Q	J	K	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

(c) Transition table

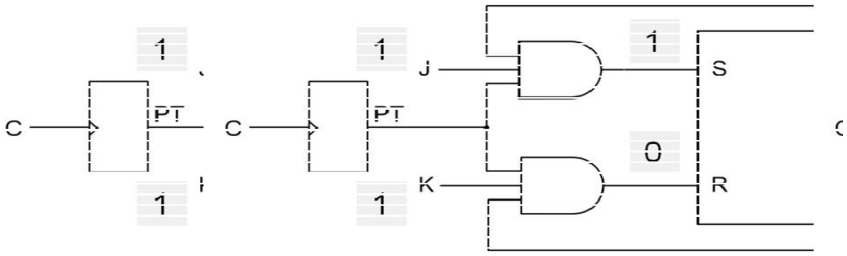
Clocked JK flip-flop

JK Flip-Flop Operation

- **Consider J = K = 1. If Q = 1, the lower AND gate passes a RESET pulse on the next PT and Q changes to 0 as illustrated in the figure below:**



- **Consider J = K = 1. If Q = 0, the upper AND gate passes a SET pulse on the next PT and Q changes to 1 as illustrated in the figure below:**



- **Therefore, J = K = 1 means the flip-flop will toggle (switch to the opposite state) on the next positive clock edge.**

Racing Problem

- **When J = K = 1, the output can continue to toggle as long as the clock is high. This condition is called the race-around condition. Propagation delay prevents the JK flip-flop from racing. The JK flip-flop output changes after the PT of the clock. T he new Q and Q' values are too late to coincide with the PTs driving the AND gate. If PTs are narrower than the propagation delay of the flip-flop, the returning Q and Q' arrive too late to cause false triggering.**

Avoiding Racing Problem

- **A better method to avoid race around condition is to use a master-slave circuit.**

Timing Diagram

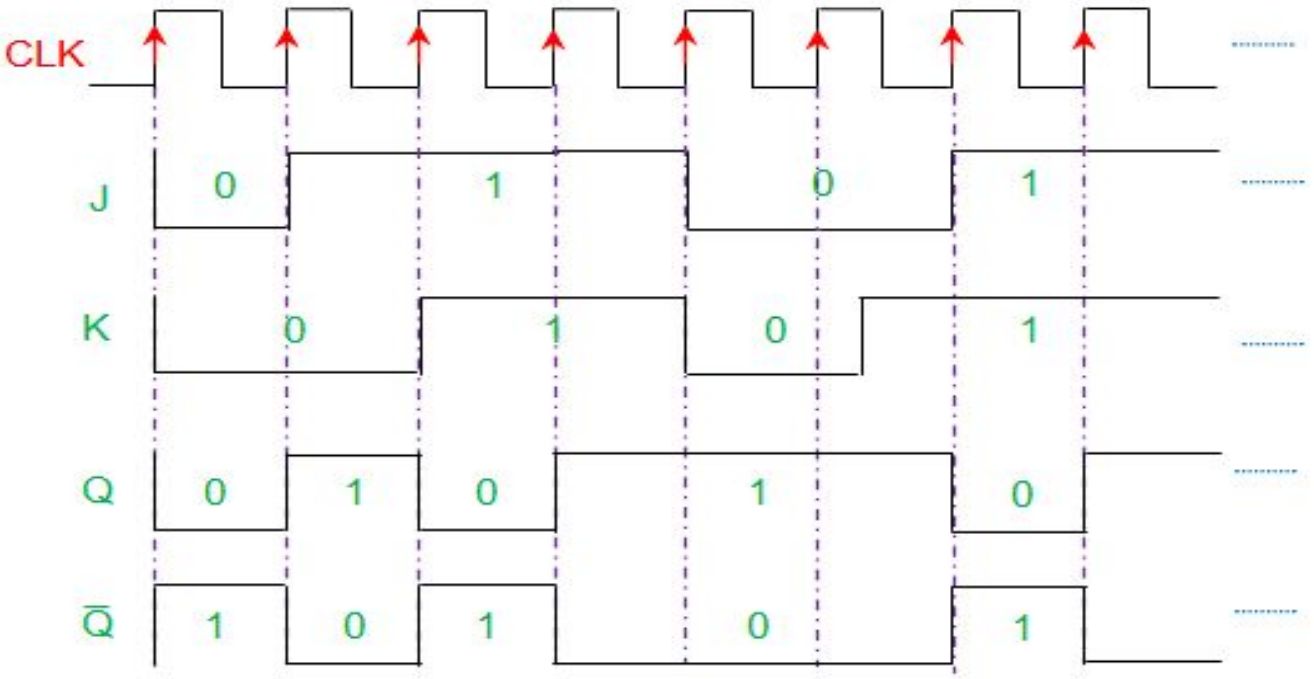
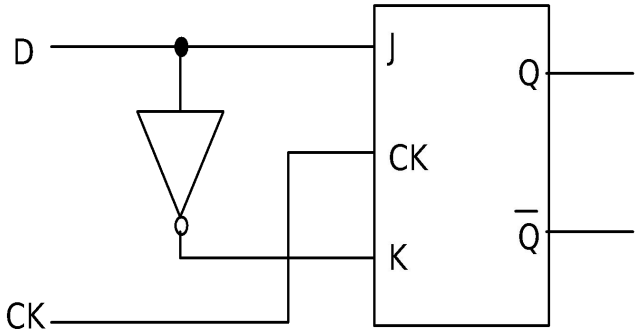


Figure 3 Timing diagram for positive edge-triggered JK flip-flop

The D-Type Flip-Flop

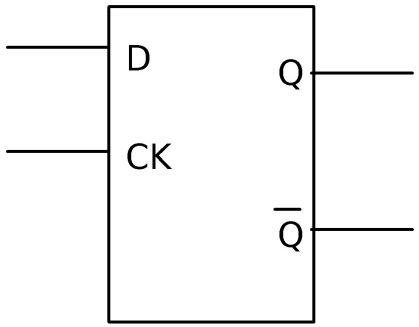
- The D (data) type flip-flop is basically a single-bit storage device. It can be constructed by connecting an inverter between the J and K inputs of a JK flip-flop as shown overleaf.



D	Qn+1
0	0
1	1

- Data appearing on the data input D is simply clocked through to the output Q on the application of a clock pulse.

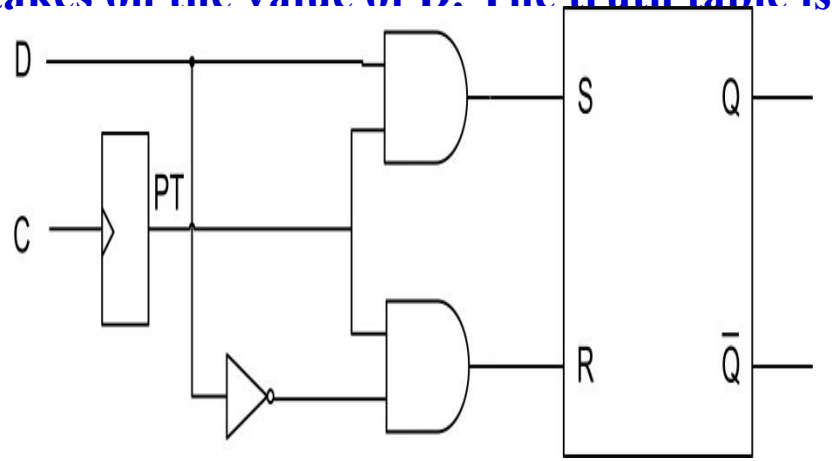
Symbol



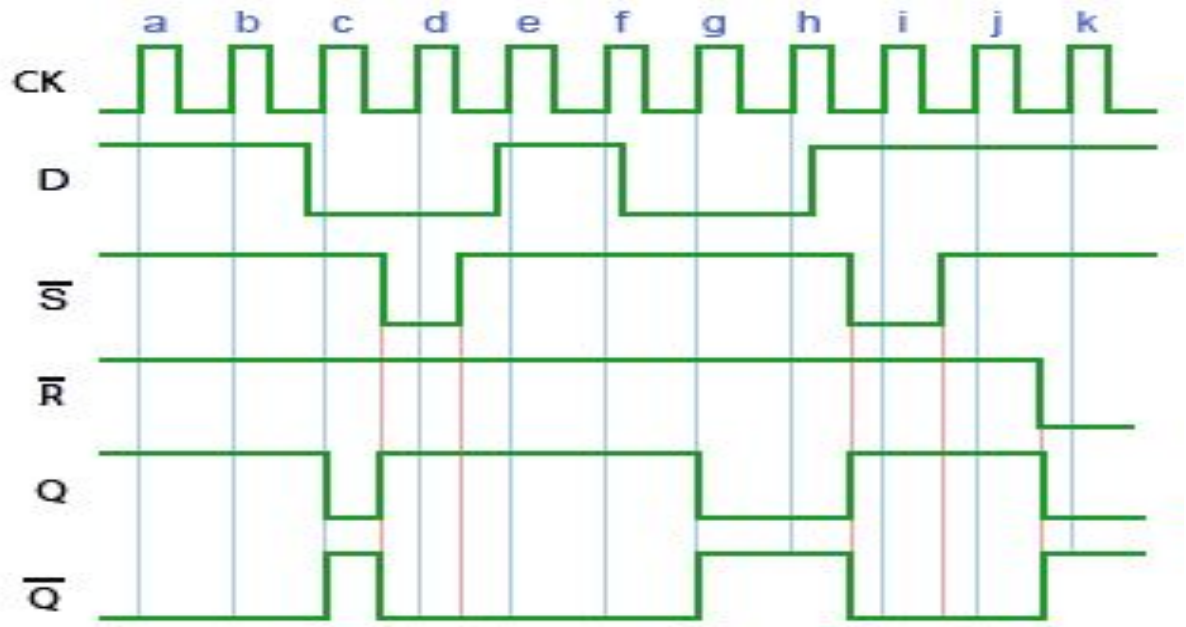
Edge-triggered D Flip-Flop

- The logic circuit of a positive-edge triggered D flip-flop is as shown:

- On the leading edge of the clock (P T), the data bit is loaded into the flip-flop and Q takes on the value of D. The truth table is as shown below:

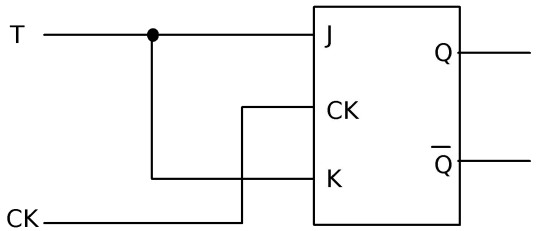


Timing Diagram of D flip flop



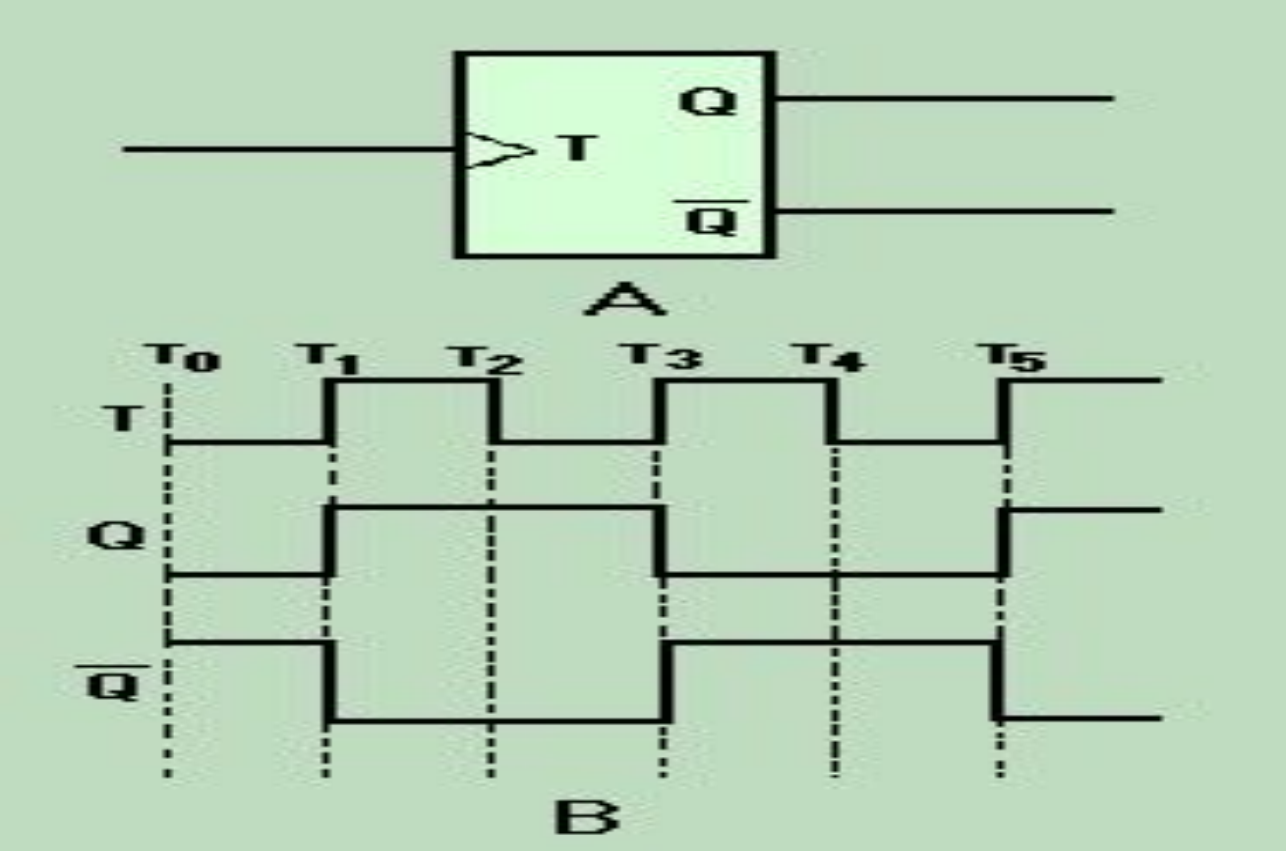
T-Type Flip-Flop

- **The T (toggle) type flip-flop also has a single input. It too can be implemented using a JK flip-flop; the inputs are simply connected together as shown below**



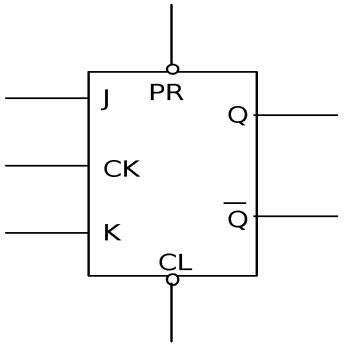
T	Qn+1
0	Qn
1	/Qn

Timing Diagram of T flip flop



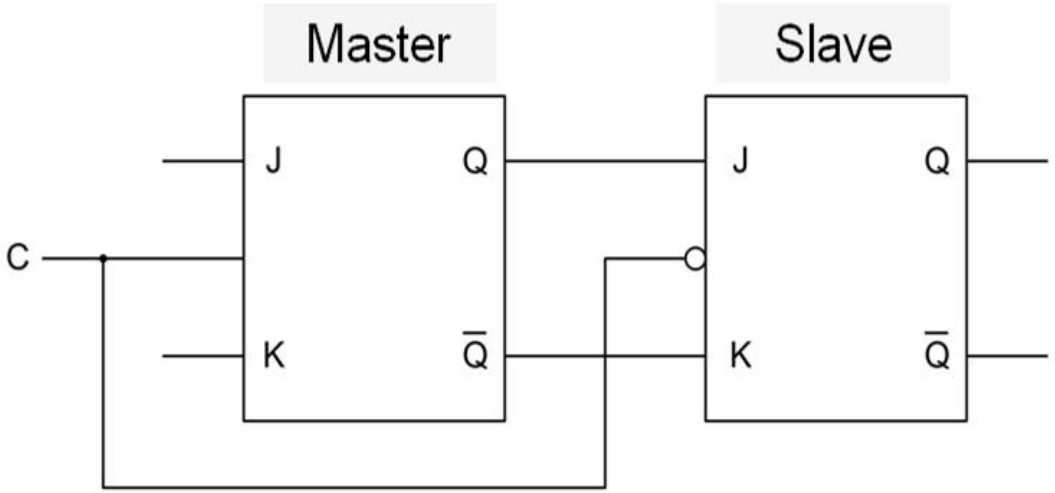
PR ESET and CLEAR inputs

- When power is first applied, flip-flops will have their outputs in random states. We can make the flip-flop to have the desired starting state. Additional PRESET and CLEAR inputs are used. The PRESET and CLEAR inputs are called asynchronous inputs because they activate the flip-flop independently of the clock. The D input is a synchronous input because it has an effect only with the positive transitions of the clock.



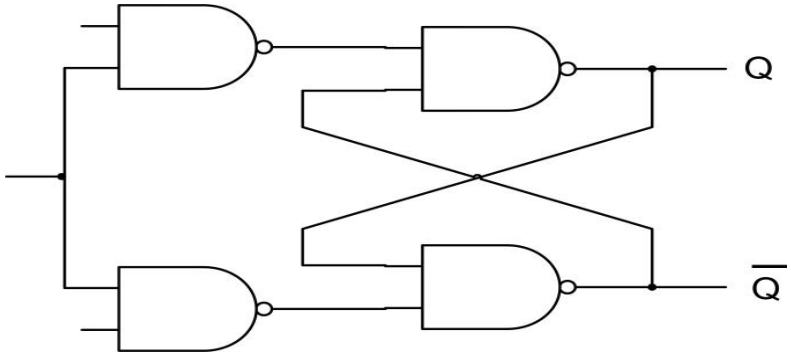
JK Master-Slave Flip-Flop

- The JK master-slave flip-flop configuration is as shown below: The J and K data is processed by the flip-flop after a complete clock pulse. On the positive transition of the clock, the data from the J and K inputs is transferred to the master. On the negative transition of the clock, the data from the master is transferred to the slave. The flip-flop is referred to as pulse-triggered.

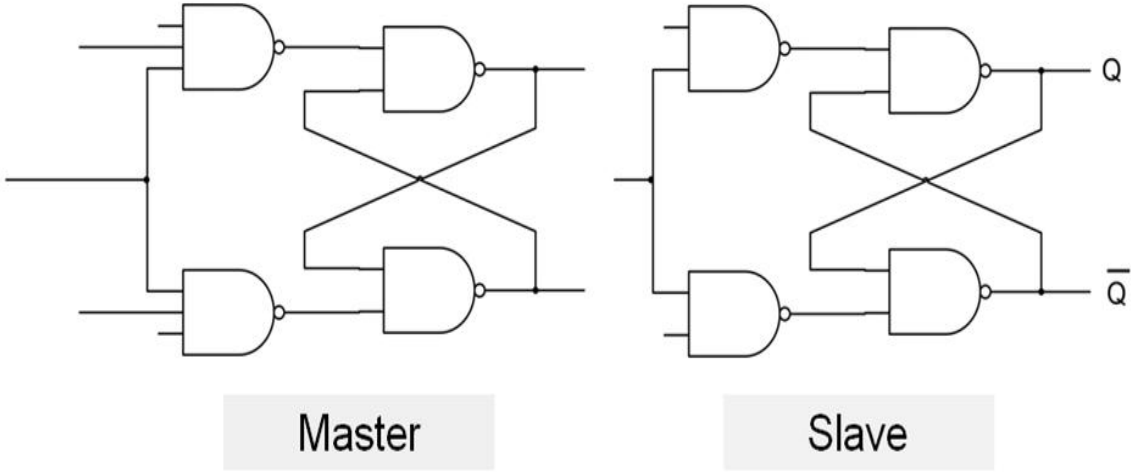


JK Master-Slave Flip-Flop Construction using NAND gates

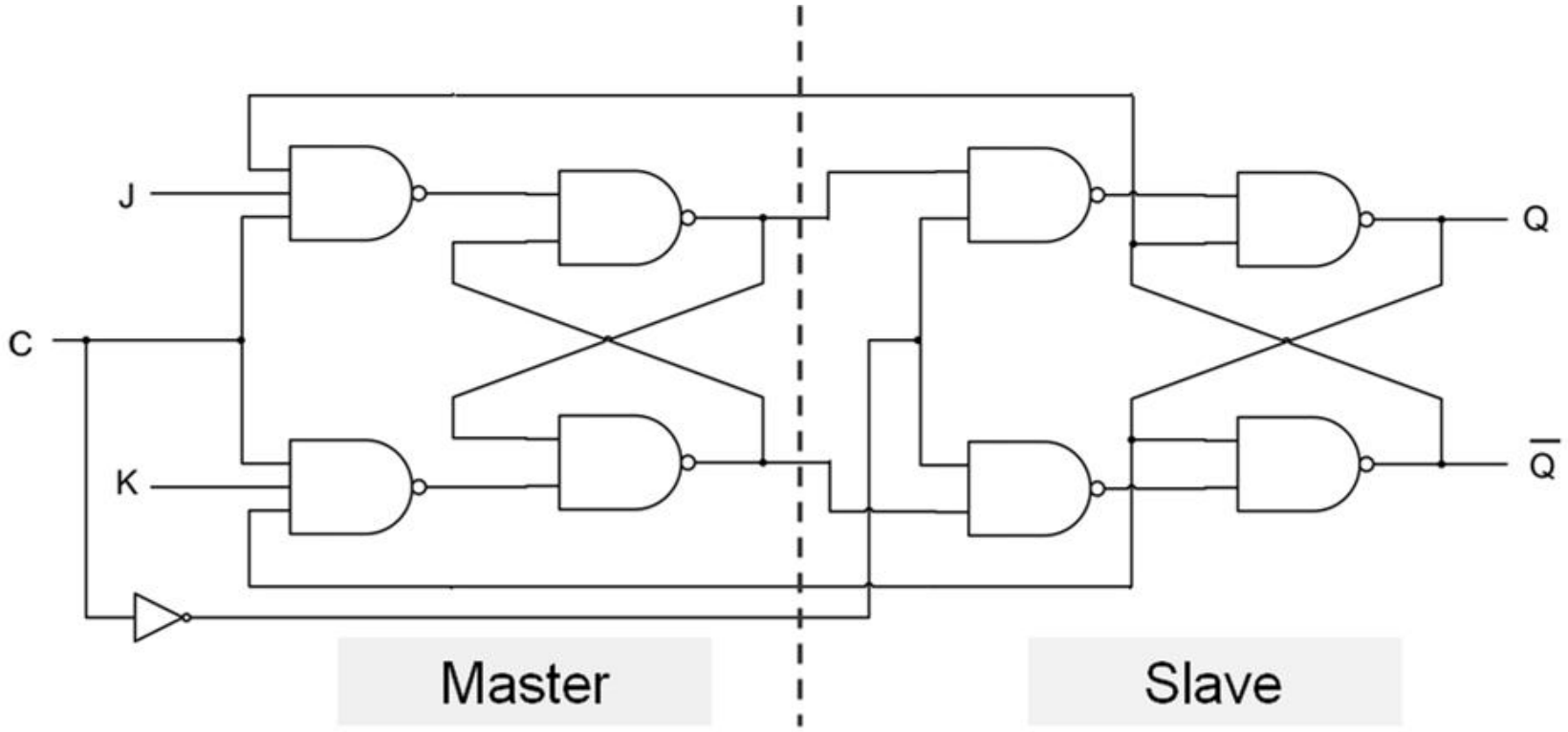
- **Step 1 :**



- **Step 2**



Continued…

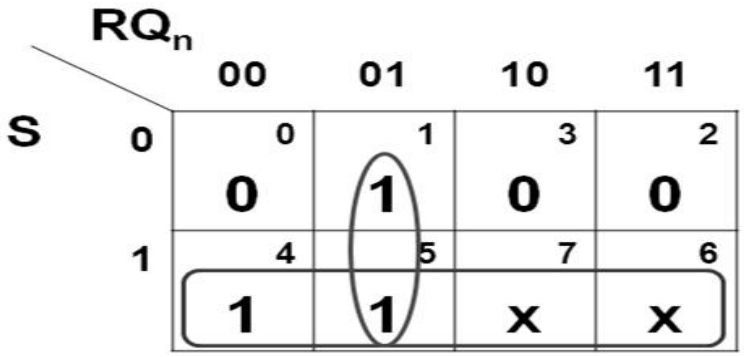


Various Representations of Flip-Flops

- There are various ways a flip-flop c an be represented, each one suitable for certain application. They are:
- Truth Tables or Characteristic Tables
- Characteristic Equations
- Finite State Machine
- Excitation Table

Characteristic Equations

- The characteristic equations of flip-flops are useful in analyzing circuits made of them. Here, next output Q_{n+1} is expressed as a function of present output Q_n and inputs to flip-flop.
- **Characteristic Equation of SR Flip-Flop**
- The characteristic equation of SR flip-flop is obtained as shown below:



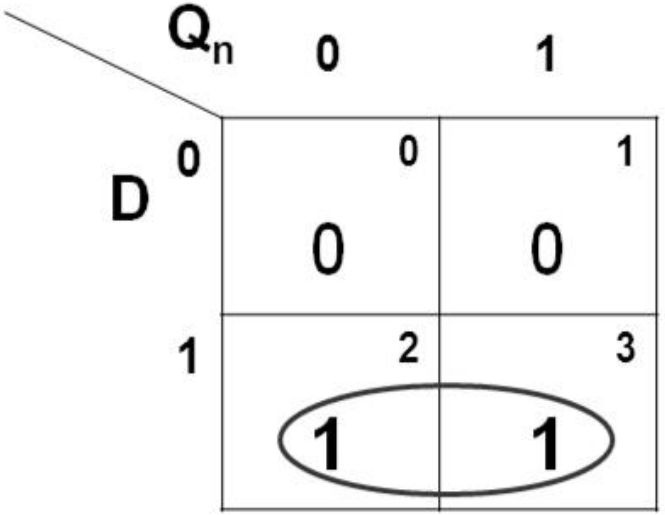
$$Q_{n+1} = S + R'Q_n$$

S	R	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	x
1	1	1	x

Characteristic Equation of D Flip-Flop

- **The characteristic equation of D flip -flop is obtained as shown below.**

D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1



$$Q_{n+1} = D$$

Truth Table

Trigger	Inputs		Output				Inference
			Present State		Next State		
CLK	J	K	Q	$\bar{Q}$	Q	$\bar{Q}$	
✗	x	x	-		-		Latched
↑	0	0	0	1	0	1	No Change
↑			1	0	1	0	
↑	0	1	0	1	0	1	Reset
↑			1	0	0	1	
↑	1	0	0	1	1	0	Set
↑			1	0	1	0	
↑	1	1	0	1	1	0	Toggles
↑			1	0	0	1	

Table I Truth table for positive-edge triggered JK flip-flop

Characteristic Equation of JK Flip-Flop

J \ KQ	00	01	11	10
0	0 (0)	1 (1)	0 (3)	0 (2)
1	1 (4)	1 (5)	0 (7)	1 (6)

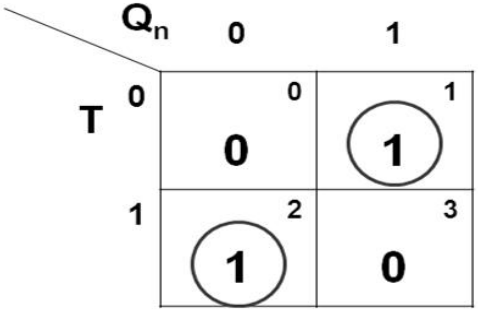
Table II K-map simplification for JK flip-flop output

$$Q_n = J\overline{Q} + \overline{K}Q$$

Characteristic Equation of T Flip-Flop

The characteristic equation of T flip -flop is obtained as shown below:

T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0



$$Q_{n+1} = TQ_n' + T'Q_n$$

- **Drawback of Truth Table**
- **For a complex circuit a truth table is difficult to read as its size becomes too large. In finite state machine (FSM), functional behavior of the circuit is explained using finite number of states.**

Flip-Flop Excitation Table

- **Truth tables are important in analysis problems. Excitation tables show the required input trigger signals required for the desired flip-flop state transition. Excitation tables are very useful in synthesis or design problems. The excitation tables of different flip-flops are as shown:**
- **Excitation Table of SR flip-flop**

S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	?

Truth Table

Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Excitation Table

Excitation Table of SR flip-flop

S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	?

Truth Table

Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Excitation Table

Excitation Table of D flip-flop

D	Q_{n+1}
0	0
1	1

Truth Table

Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

Excitation Table

Excitation Table of JK flip-flop

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	Q_n'

Truth Table

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Excitation Table

Excitation Table of T flip-flop

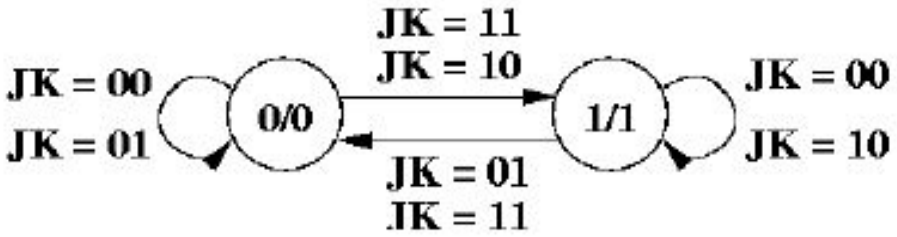
States		Input
Present Qn	Next Qn+1	T
0	0	0
0	1	1
1	0	1
1	1	0

Flip Flops as Finite State Machine

- It is a computation model that can be implemented with hardware or software and can be used to simulate sequential logic and some computer programs.
- Sequential circuits change their states for every positive *or negative* transition of the clock signal based on the input. So, this behavior of synchronous sequential circuits can be represented in the graphical form and it is known as **state diagram**.

State Transition Diagram of JK flip flop

JK flip-flop:

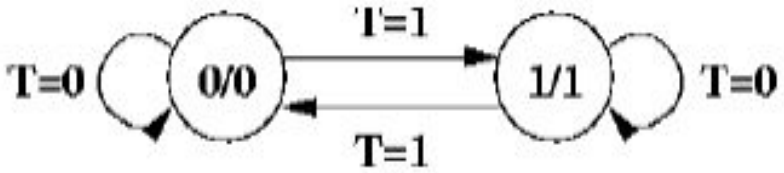


State: 1 bit (q)
Input: 2 bits (JK)
Output: current state (z)

q	J	K	q^+	z	
0	0	0	0	0	hold
0	0	1	0	0	reset
0	1	0	1	0	set
0	1	1	1	0	toggle
1	0	0	1	1	hold
1	0	1	0	1	reset
1	1	0	1	1	set
1	1	1	0	1	toggle

State Transition Diagram of T flip flop

T flip-flop:



State: 1 bit (q)
Input: 1 bit (T)
Output: current state (z)

q	T	q^+	z	D
0	0	0	0	0
0	1	1	0	1
1	0	1	1	1
1	1	0	1	0

In state 0:

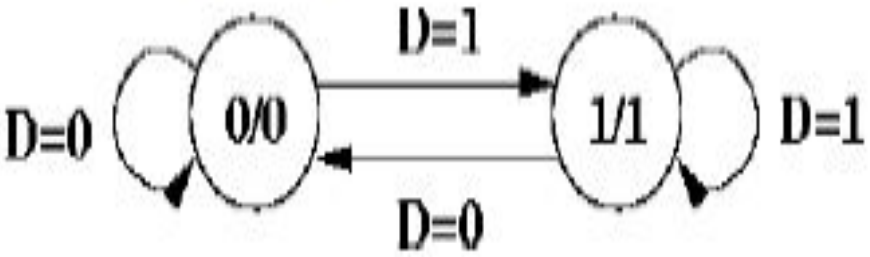
input 0 gives new state 0 (hold)
input 1 gives new state 1 (toggle)

In state 1:

input 0 gives new state 1 (hold)
input 1 gives new state 0 (toggle)

State Transition Diagram of JK flip flop

Flip-flop can be modeled as a finite-state machine



D flip-flop

State: 1 bit (q)

Input: 1 bit (D)

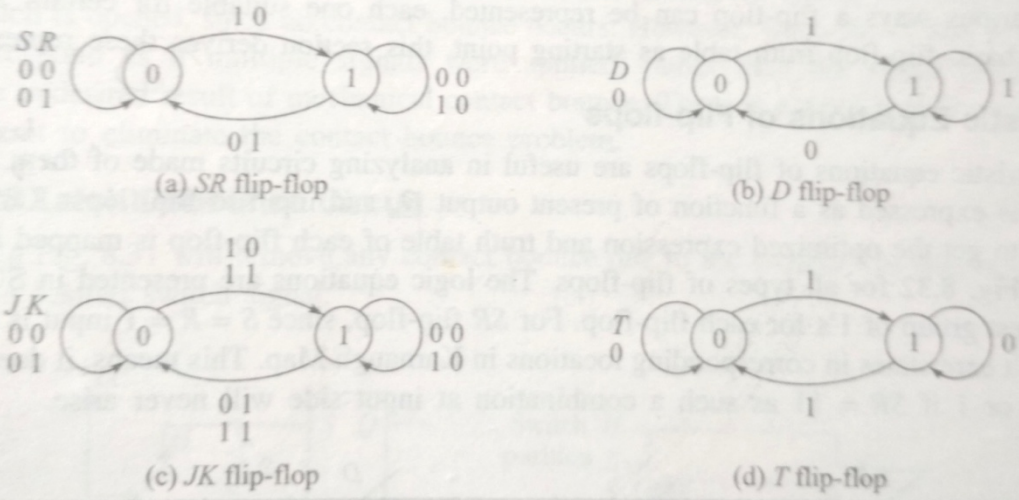
Output: current state (z)

q	D	q^+	z	T
0	0	0	0	0
0	1	1	0	1
1	0	0	1	1
1	1	1	1	0

- **Can You Draw For SR flip flop?**

Flip-Flops as Finite State Machine

In a sequential logic circuit the value of all the memory elements at a given time define the *state* of that circuit at that time. Finite State Machine (FSM) concept offers 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. In Fig. 8.33 all the flip-flops are represented as finite state machine through their state transition diagrams.



(a) *SR* flip-flop (b) *D* flip-flop

(c) *JK* flip-flop (d) *T* flip-flop

Fig. 6.33 *State transition diagram of (a) SR flip-flop. (b) D flip-flop. (c) JK flip-flop. (d) T flip-flop.*

Thank You