Sequential logic circuit

Digital Electronics (DE)

Information and Communication Technology

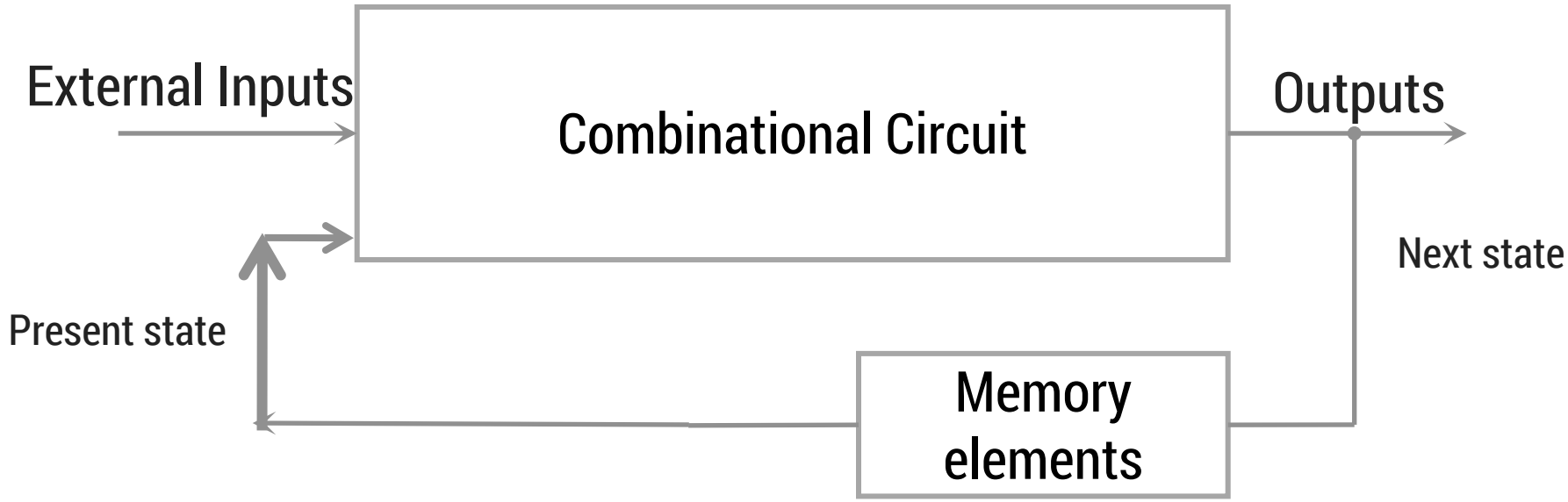
Prepared by Prof. jaydip siyara

Outline

- R-S Flip-Flop D Flip-flops
- Level-Triggered and Edge-Triggered Flip-Flops
- J-K Flip-Flop T Flip-flops
- Synchronous and Asynchronous Inputs
- Applications of FF
- Ripple/Asynchronous Counters
  - Ripple up-down, Modulo or Modulus counter
- Synchronous Counters
  - Up counter, Down counter using different FFs
- Decade and BCD counter
- Presettable Counters
- Decoding a Counter
- Designing Counters with Arbitrary Sequences
- Registers
  - Buffer, Shift, Bidirectional, Universal
  - applications of shift registers,
    - serial to parallel converter, parallel to serial converter
    - ring counter, sequence generator,
- IEEE/ANSI Symbology for Registers and Counters

Sequential Circuits

- Its output levels at any instant of time are dependent on the present inputs at that time and on the state of the circuit, i.e., on the prior input level conditions (i.e. on its past inputs)
- The past history is provided by feedback from the output back to the input.
- Combinational circuits + memory elements = Sequential Circuits.
- Eg. Counters, shift registers, etc.



Sequential Switching Circuits

- **Present state(State input)**: The data stored by memory element at given instant of time.

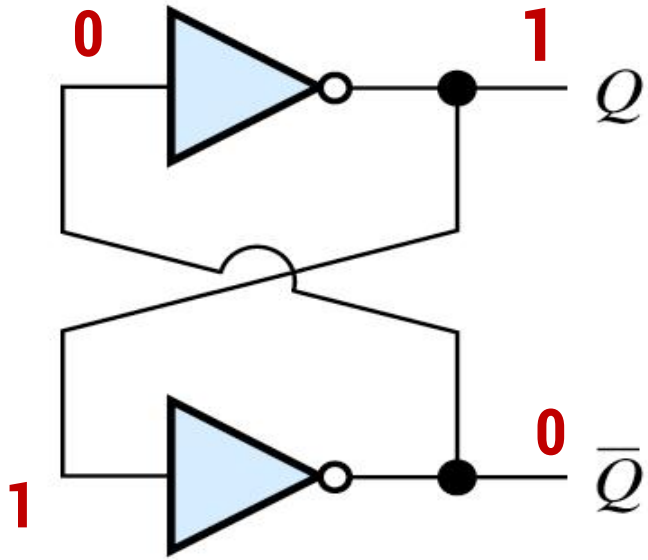
- **Next state:** Combinational circuit have two input 1) External input and 2) present state then new output will be stored in memory called Next state.
- Input: Signal from outside ckt.
- Output: Signals to the outside ckt.

Sequential Circuits v/s Combinational Circuits

Sequential Circuits	Combinational Circuits
The output variables at any instant of time are dependent on the **present input** variables and on the present state, i.e., on the **past history** of the system.	In combinational circuits, the output variables at any instant of time are **dependent only on the present input** variables.
Memory unit is required to store the past history of the input variables in sequential circuits.	Memory unit is **not required** in combinational circuits.
Slower than combinational circuits.	**Faster** because the delay between the input and the output is due to propagation delay of gates only.
Comparatively **harder to design.**	**Easy to design.**
Clock is required	**Clock not required.**
Example:FF,FSM,Registers,Counter	Example: Half- Full adder/Subtractor, Mux-demux Encoder/Decoder,Code converter
Block diagram	Block diagram

1-bit memory and the circuit properties of Bistable latch

- Electronic circuit that has **two stable states** and is capable to store one bit of memory.
- A latch bi-stable multivibrator.
- It can remain in either of the states indefinitely.
- Latch is used for **certain flip-flop** which are *non-clocked.*
- These 'latch on' to a 1 or a 0 immediately upon receiving the input pulse called SET or RESET.
- It has two stable states of complementary output values.
- As an example:

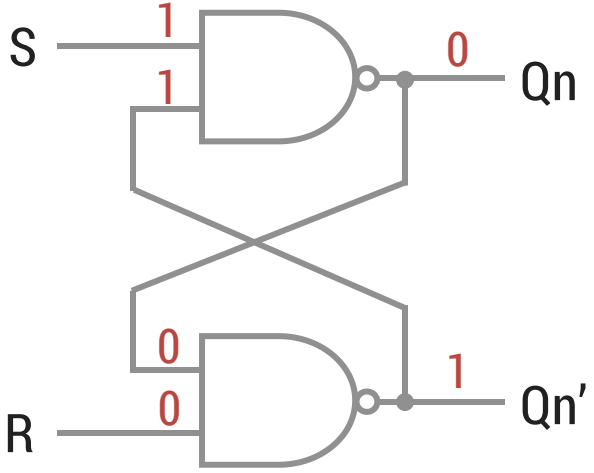


Bistable Latch/ 1 Bit Memory Circuit/S-R Latch

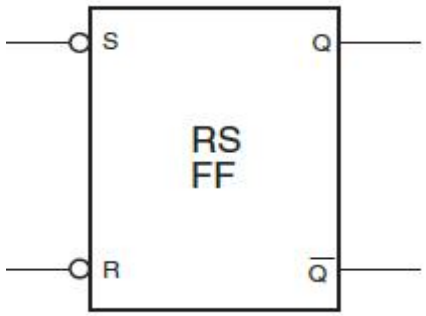
- The simplest type of flip-flop is called an S-R latch.
- It has two outputs labelled Q and Q' and two inputs labelled S and R.
- The state of the latch corresponds to the level of Q (HIGH or LOW, 1 or 0) and Q' is the complement of that state.
- It can be constructed using either **two cross-coupled NAND gates** or **two cross-coupled NOR gates.**
- Using two NOR gates, an active-HIGH S-R latch can be constructed and using two NAND gates an active-LOW S-R latch can be constructed(Cross coupled inverter).
- The name of the latch, S-R or **SET-RESET**, is derived from the names of its inputs.

S-R latch (Active Low) using NAND Gate

A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0



Logic diagram

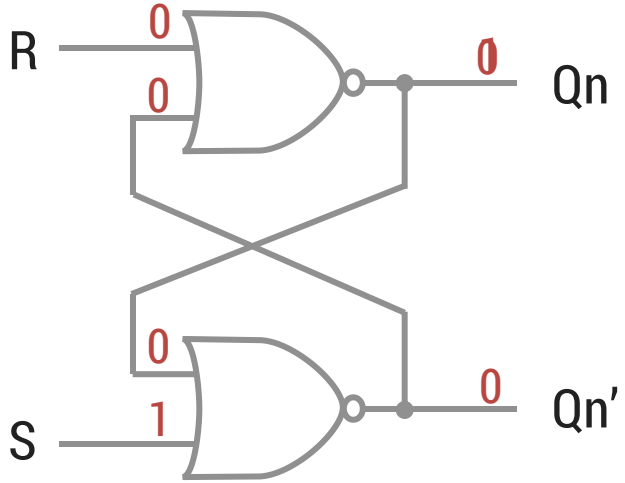


S	R	Q_n	Q_{n+1}	State

S-R latch (Active High) using NOR Gate

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Logic Symbol(Block diagram)



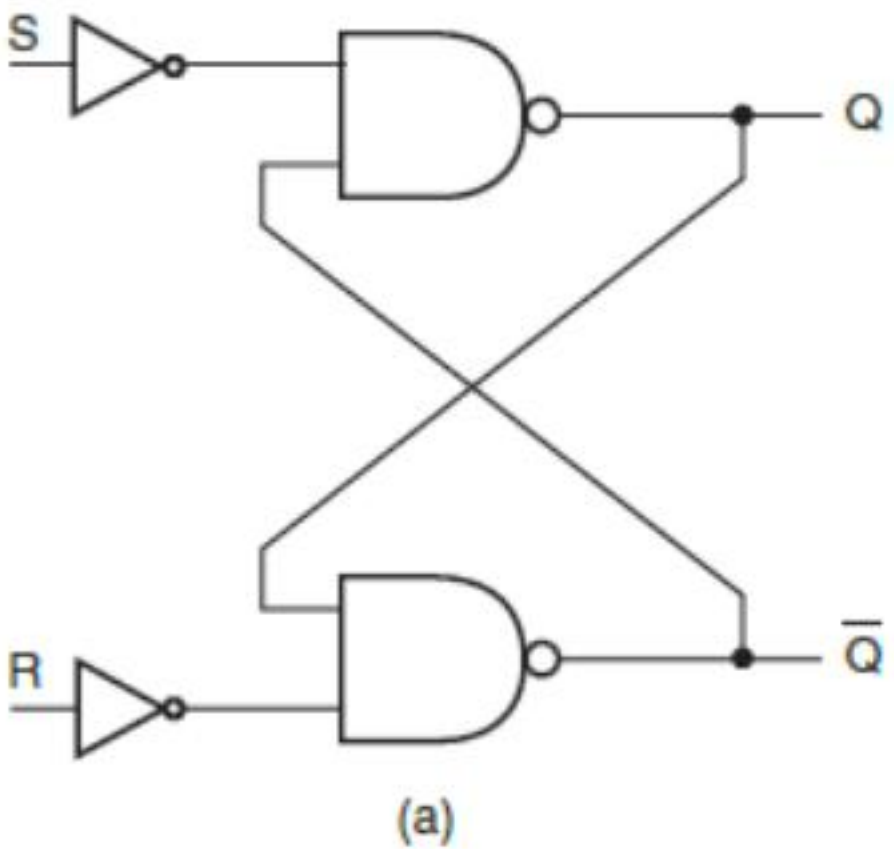
Logic diagram

S	R	$Q_{n(PS)}$	Q_{n+1NS}	State

Characteristic Table

Combination of input and present state and output is next state.

S-R latch (Active High) using NANDGate



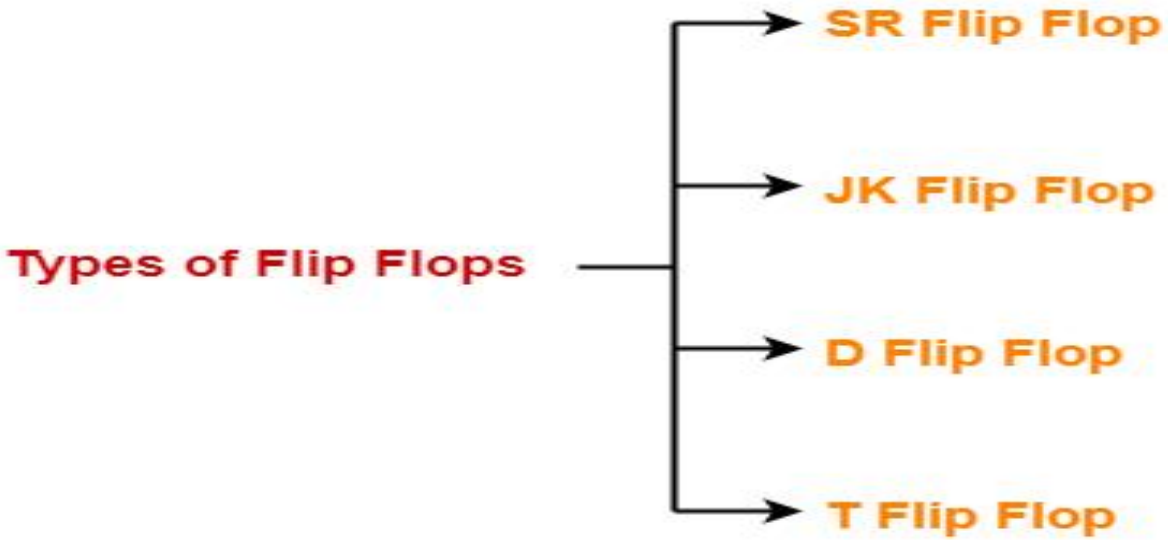
(a)

Operation Mode	S	R	Q_{n+1}
No change	0	0	Q_n
SET	1	0	1
RESET	0	1	0
Forbidden	1	1	—

Clocked latch = FF

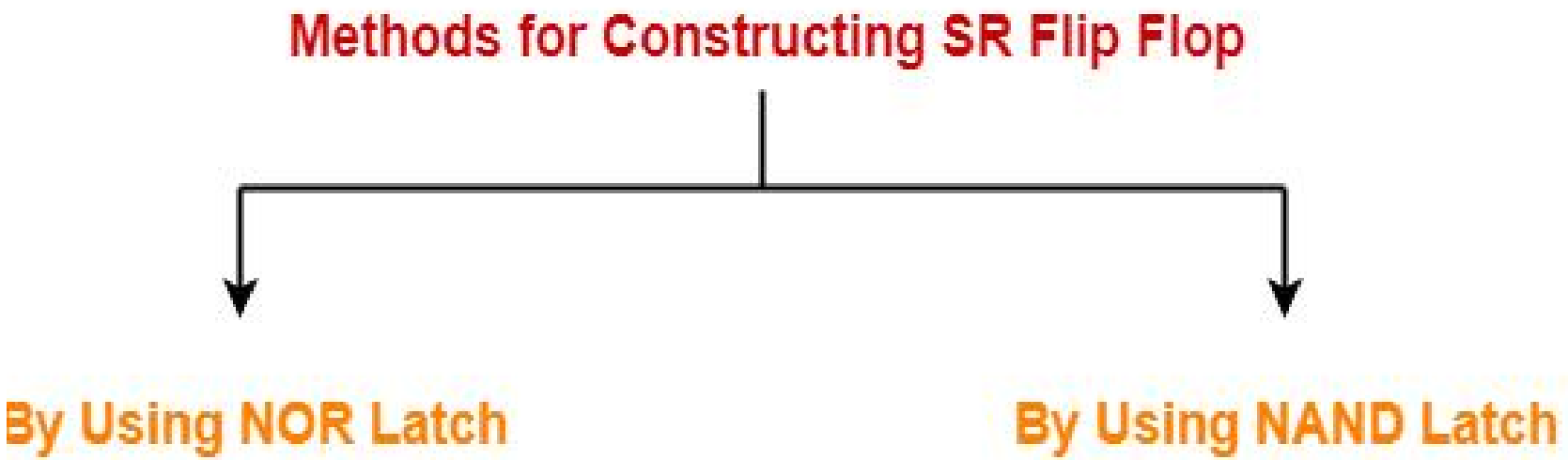
- Output of latch can change any time if input is changed.
- **It is not controllable.**
- To make it controllable, one more input is there called **Enable (CLOCK) signal**.
- A Flip Flop is a **memory element** that is capable of storing one bit of information.
- It is also called as **Bistable Multivibrator** since it has two stable states either 0 or 1.

There are following 4 basic types of flip flops-



Gated S-R Latch/ Clocked S-R Latch (S-R Flip-flop)

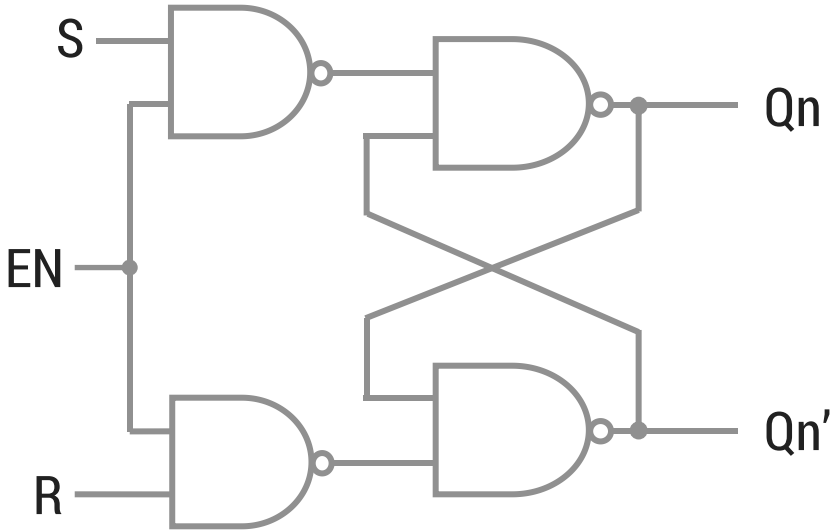
- SR flip flop is the simplest type of flip flops.
- It stands for **Set Reset flip flop.**
- It is a clocked flip flop.



Gated S-R Latch/ Clocked S-R Latch (S-R Flip-flop)

A	B	A·B
0	0	1
0	1	1
1	0	1
1	1	0

Logic Symbol



Logic diagram

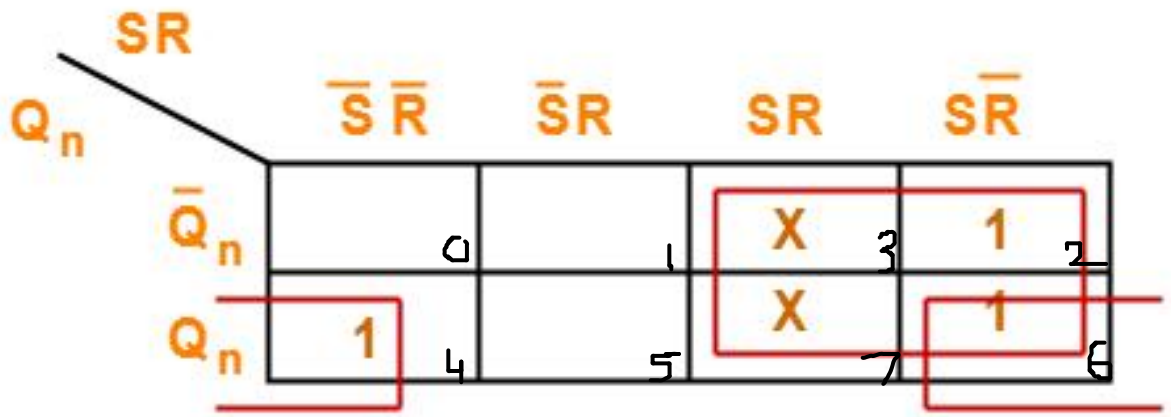
EN	S	R	Q_n	Q_{n+1}	State

Characteristic Table

Characteristic Equation-

Draw a k map using the above truth table-

Q_n	S	R	$Q(n+1)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	UNKNOWN
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	UNKNOWN



K Map

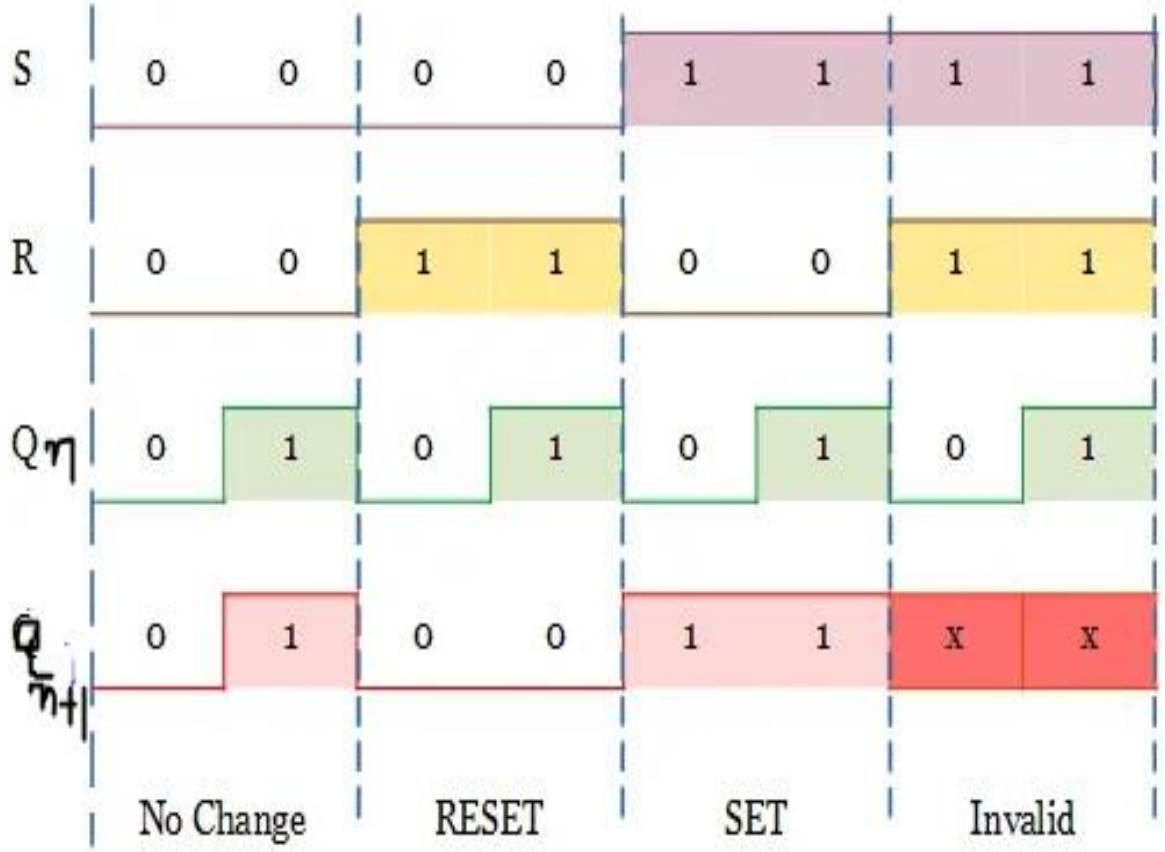
$$Q_{n+1} = S + Q_n R'$$

Switching or Timing diagram of SR FF

S-R Flip-Flop

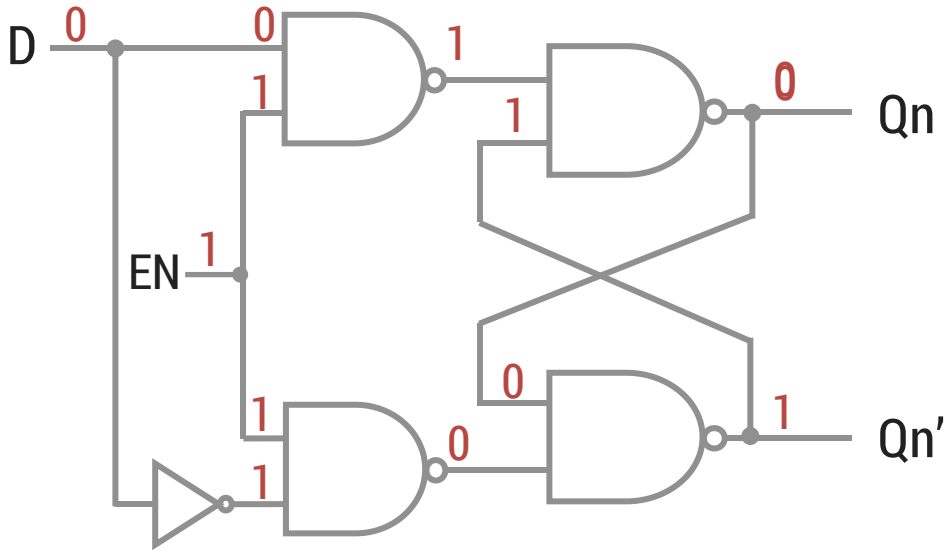
S	R	Q(n+1)
0	0	Q(n)
0	1	0
1	0	1
1	1	Indeterminate

The following figure shows the switching diagram of clocked SR flip flop.



Gated D-Latch or Clocked D-Latch (D Flip-flop)

Logic Symbol



Logic diagram

Characteristic Table

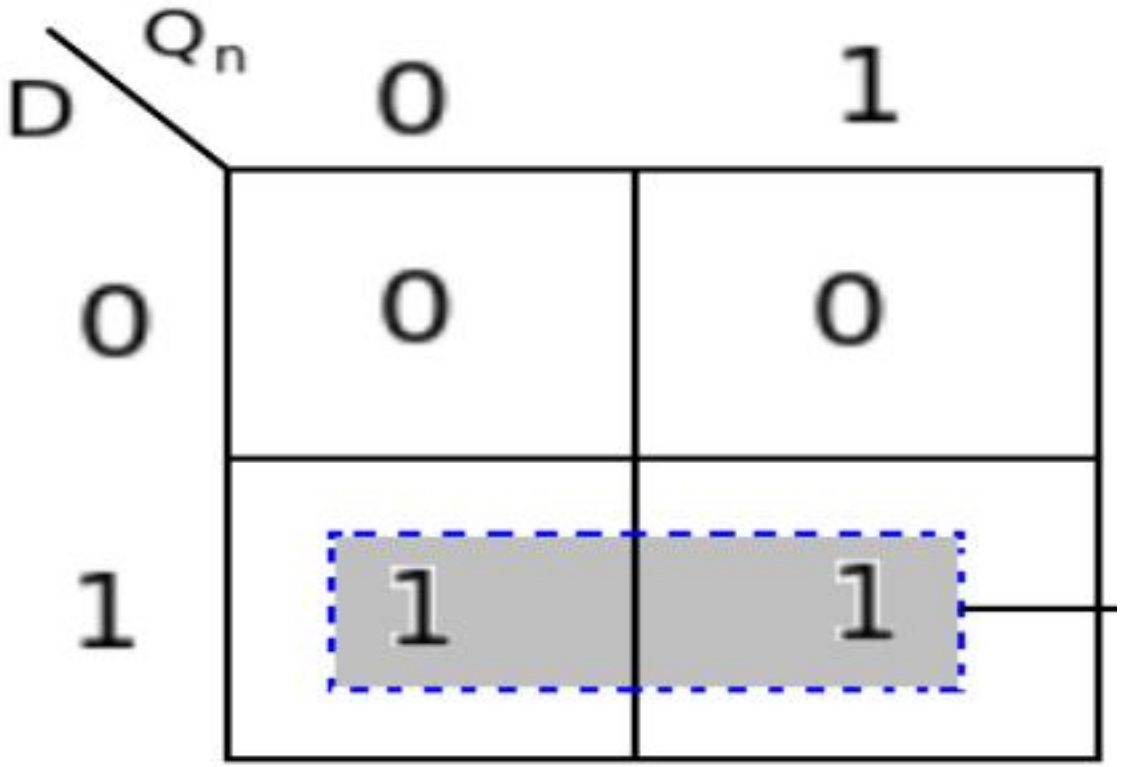
EN	D	Q_n	Q_{n+1}	State

- Here Output follows input , called Transparent latch .
- IC7475 Quad D- FF

Characteristic equation of D Flip-flop

D	Present state Q_n	Next state Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

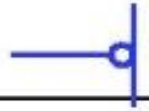
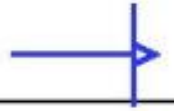
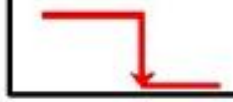
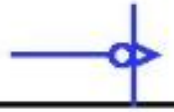
Truth table of D flip flop



Qn+1 = D

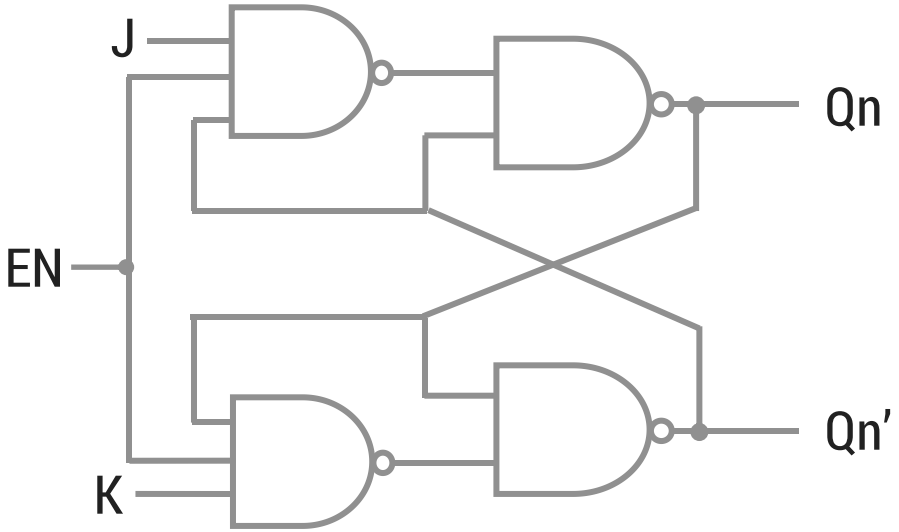
Level-Triggered and Edge-Triggered Flip-Flops

- *Level-triggered* : the output gives response to the inputs during the clock pulse **level** is HIGH or LOW.
- *Edge-triggered* : the output gives response to the inputs only on LOW-to-HIGH(positive edge triggered) or HIGH-to-LOW(negative edge triggered) transition of the clock signal.

Type		Symbol
	Positive Level	
	Negative Level	
	Positive Edge Triggered	
	Negative Edge Triggered	

J-K Flip-flop

Logic Symbol



Logic diagram

Characteristic Table

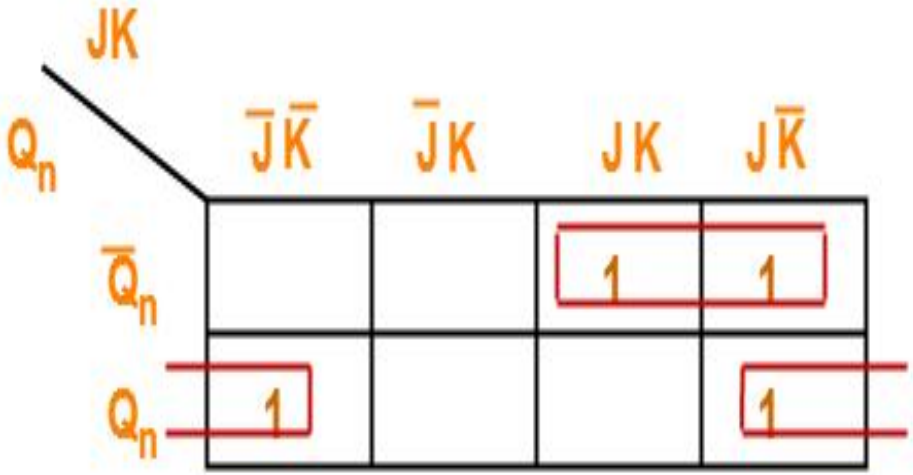
EN	J	K	Q_n	Q_{n+1}	State

Characteristic equation J-K Flip-flop

Draw a k map using the above truth table-

J	K	Present state Q_n	Next state 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	1
1	1	1	0

Truth table of JK flip flop

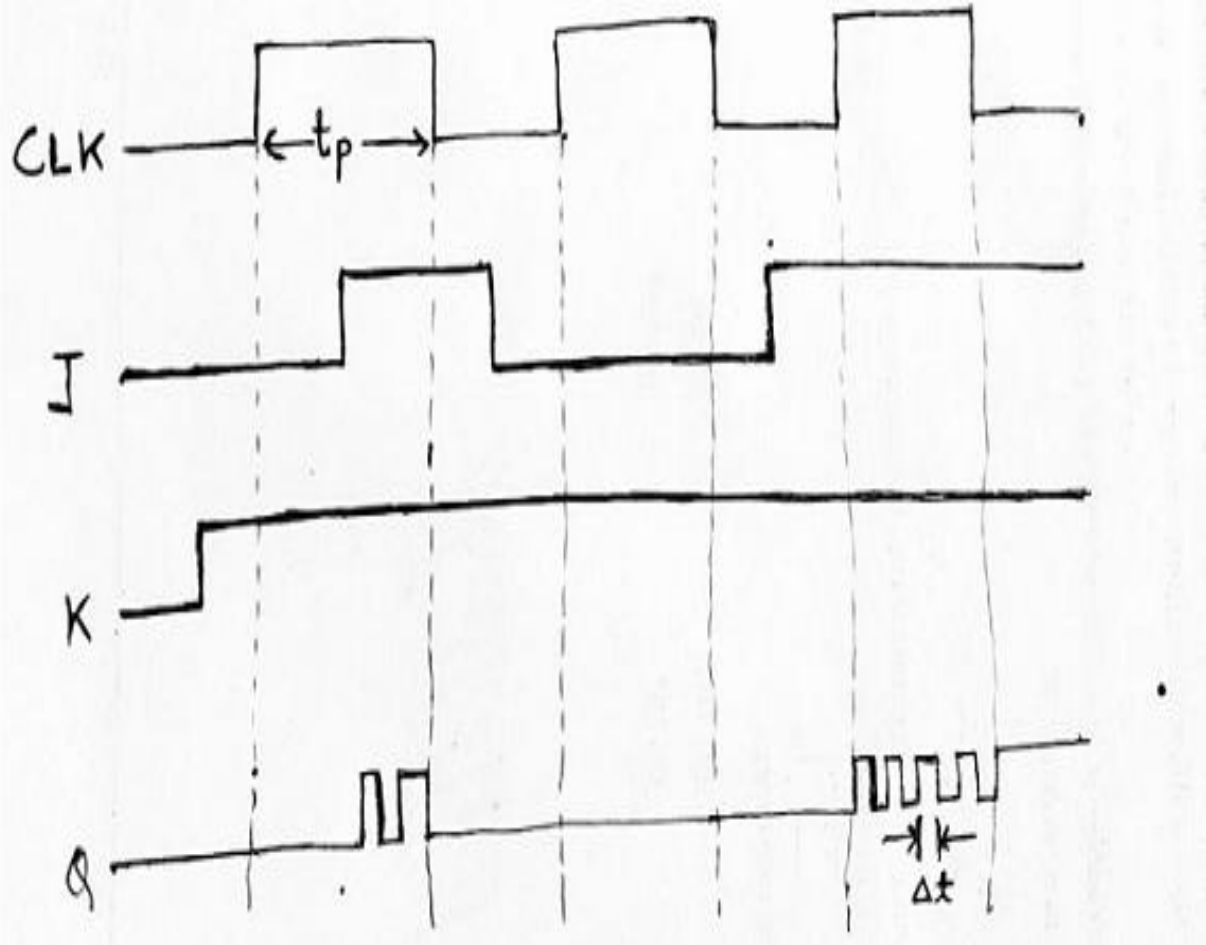


K Map

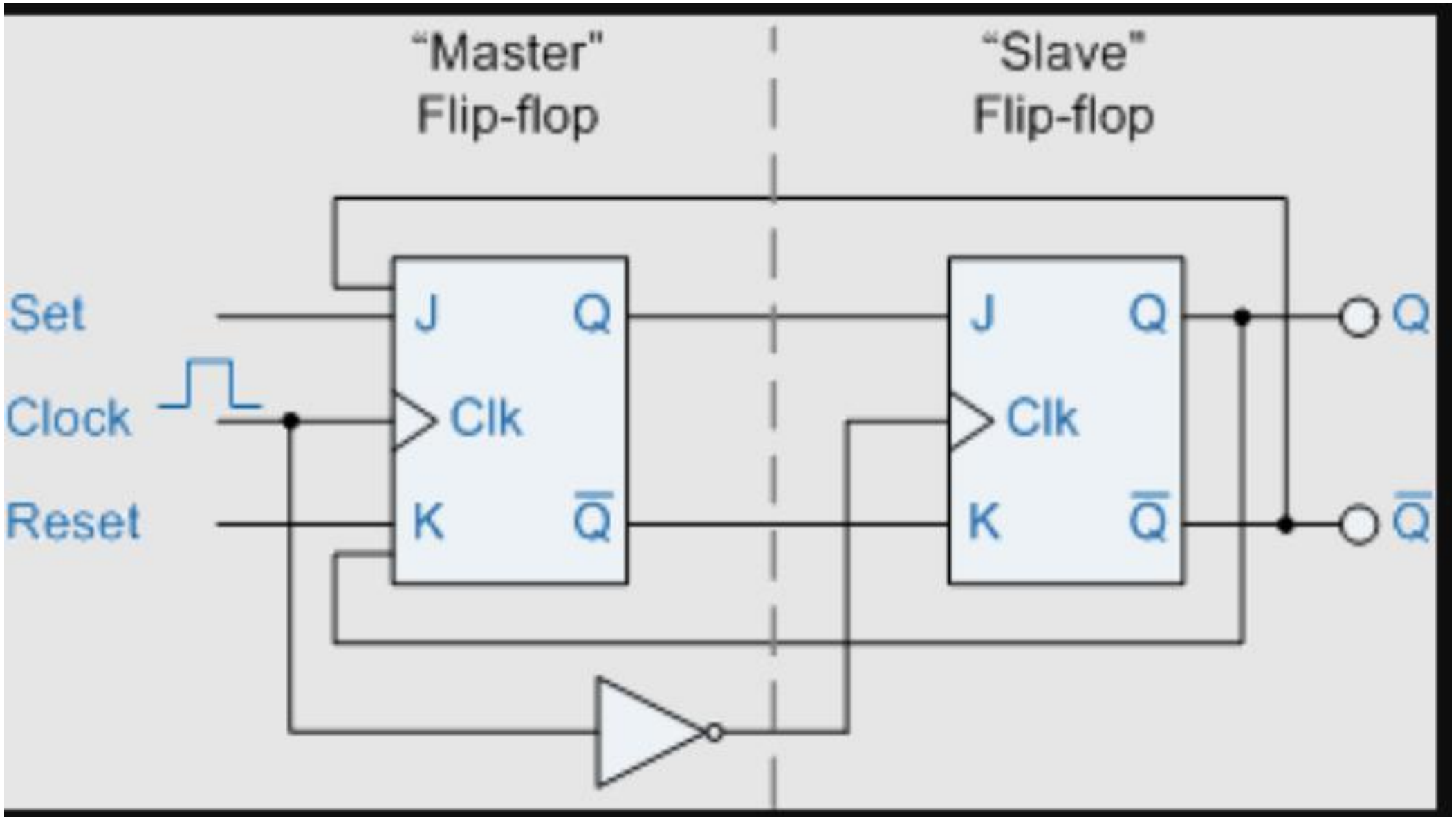
$$Q_{n+1} = Q'_n J + Q_n K'$$

Race around condition of J-K Flip-flop

- For J-K flip-flop, if J=K=1, and if **clk=1 for a long period of time**, then Q output will toggle as long as CLK is high, which makes the output of the flip-flop **unstable or uncertain**.
- This problem is called **race around** condition.
- This problem can be avoided by ensuring
  1. Edge triggering of clock
  2. clock input is at logic "1" only for a very short time.
  3. **Master Slave JK** flip flop.



Master-slave JK flip-flop



Master-slave JK flip-flop

- A combination of two JK flip-flops connected together in a series configuration.
- Out of these, one acts as the **“master”** and the other as a **“slave”**.
- The output from the master flip flop is connected to the two inputs of the slave flip flop whose output is fed back to inputs of the master flip flop.
- The inverter is connected to clock pulse in such a way that the inverted clock pulse is given to the slave flip-flop.
-

Master-Slave FF Working

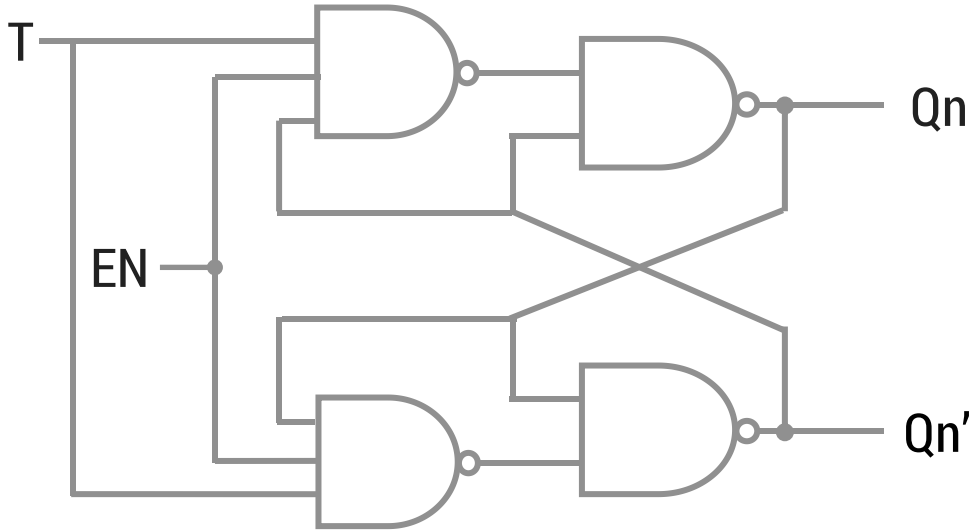
- When CLK = 1, then the **slave can be separated**; the inputs like J & K **affects on master only.**
- CLK = 0, then the data can be transmitted from the master to slave FF and the final **o/p can be obtained.**

- To write more about this FF refer below link,

  https://www.geeksforgeeks.org/master-slave-jk-flip-flop/?ref=lbp

T Flip-flop /Toggle FF

Logic Symbol



Logic diagram

Characteristic Table

EN	T	Q_n	Q_{n+1}	State

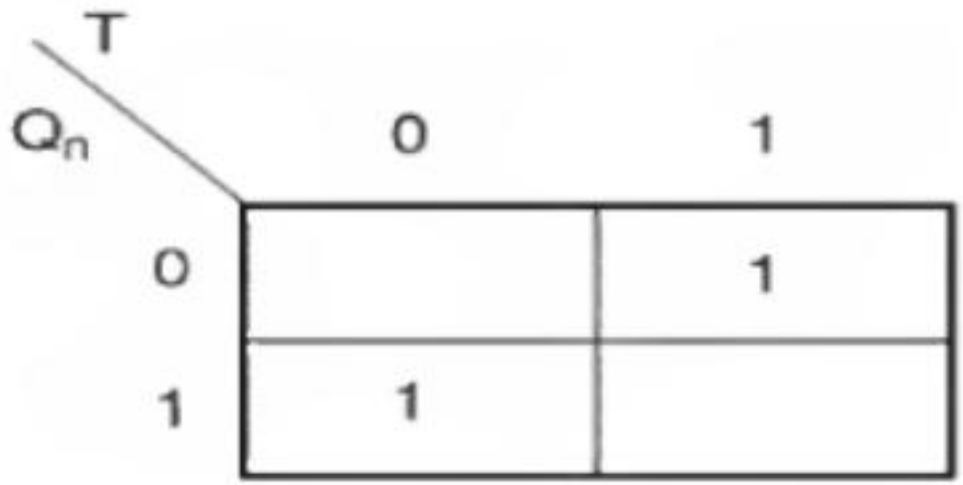
En= 1,
When T = 1 Output = Toggle
When T= 0 Output = remain unchanged.

Characteristic equation of T Flip-flop /Toggle FF

T	Present state Q_n	Next state Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Truth table of T flip flop

K map for T Flip Flop:



$$Q_{n+1} = T.\overline{Q_n} + \overline{T}.Q_n$$

Excitation Tables

Based on Present State and Next State required input are found called Excitation Table.

S	R	Present state Q_n	Next state 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

Truth table of SR flip flop

PS	NS	Required inputs	
Q_n	Q_{n+1}	S	R
0	0		
0	1		
1	0		
1	1		

S-R FF Excitation Tables

Excitation Tables

J	K	Present state Q_n	Next state 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	1
1	1	1	0

Truth table of JK flip flop

PS	NS	Required inputs	
Q_n	Q_{n+1}	J	K
0	0		
0	1		
1	0		
1	1		

J-K FF

Excitation Tables

PS	NS	Required inputs
Q_n	Q_{n+1}	D
0	0	
0	1	
1	0	
1	1	

D FF

PS	NS	Required inputs
Q_n	Q_{n+1}	T
0	0	
0	1	
1	0	
1	1	

T FF

FLIP FLOP CONVERSIONS

- SR to D
- SR to JK
- SR to T
- JK to T
- JK to D
- JK to SR
- D to T
- D to SR
- T to D

Conversion of Flipflop –conversion Steps

Draw the block diagram of the target flip flop from the given problem.

Write truth table for the target flip-flop.

Write excitation table for the available flip-flop.

Write Conversion Table for available flip-flop from Target Flip-flop's Present and Next state.

Using K-map solve equation and draw final diagram.

Conversion of Flipflop –Example 1

Convert SR FF to D FF.

- Here D FF is target FF and SR is given FF.

Apply steps.

Block diagram of D FF

Conversion of Flipflop –Example 1

D flip-flop input	Present State	Next State
D	Q t	Q $t+1$
0	0	0
0	1	0
1	0	1
1	1	1

Truth table of D FF

Present State	Next State	SR flip-flop inputs	
Q t	Q $t+1$	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

Excitation table of SR FF

Conversion of Flipflop –Example 1

D flip-flop input	Present State	Next State	SR flip-flop inputs	
D	Q t	Q $t+1$	S	R
0	0	0	0	x
0	1	0	0	1
1	0	1	1	0
1	1	1	x	0

Conversion Table

Conversion of Flipflop –Example 1

Use Kamp.

K-Map for S

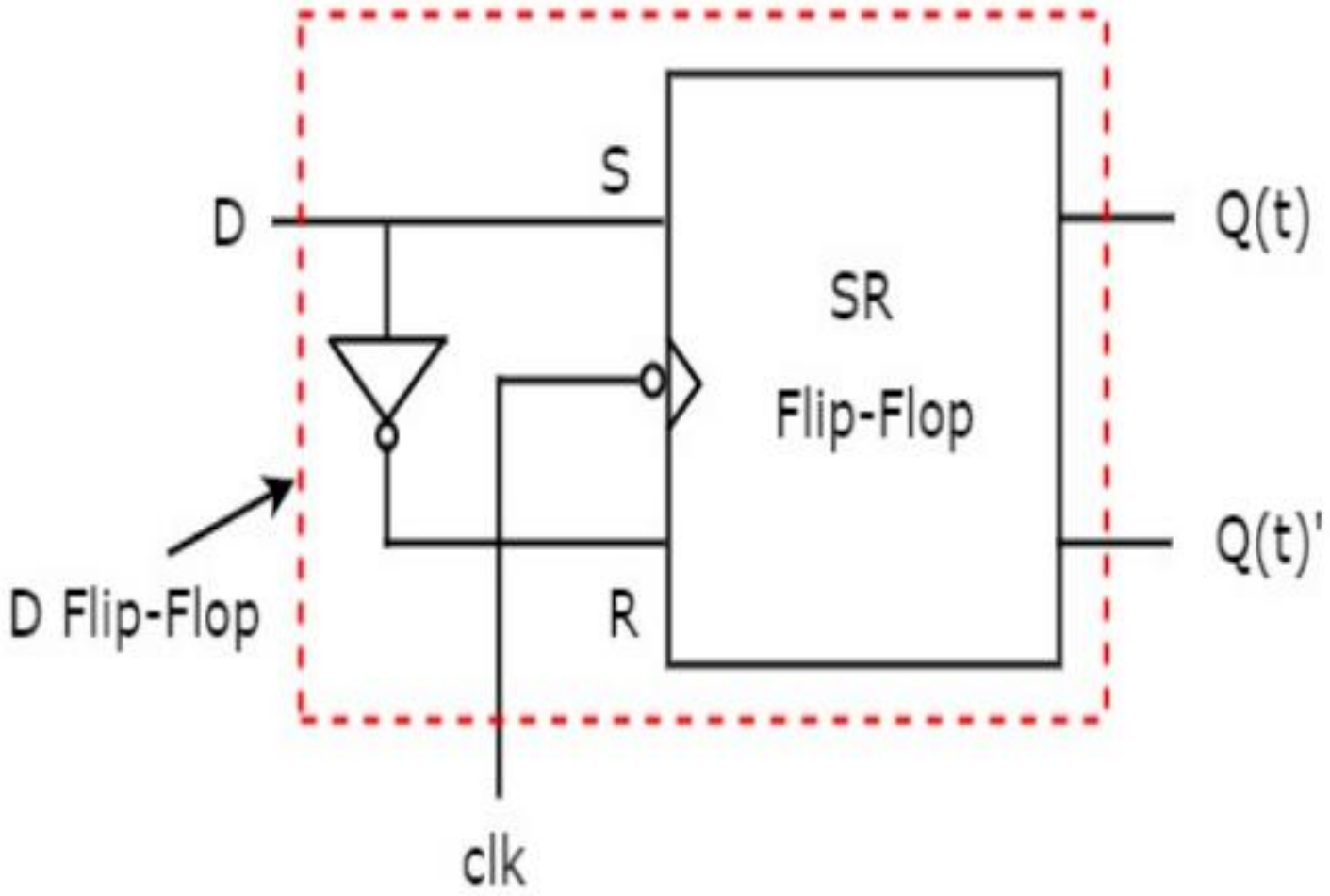
D \ Q(t)	0	1
0		
1	1	x

D

K-Map for R

D \ Q(t)	0	1
0	x	1
1		

D'



Conversion of Flipflop

For all example refer site as below link.

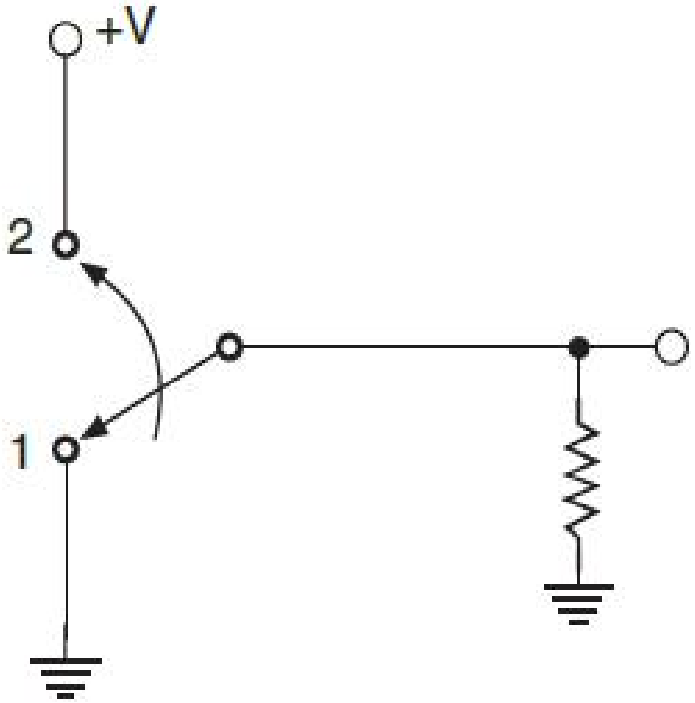
Digital Circuits - Conversion of Flip-Flops (tutorialspoint.com)

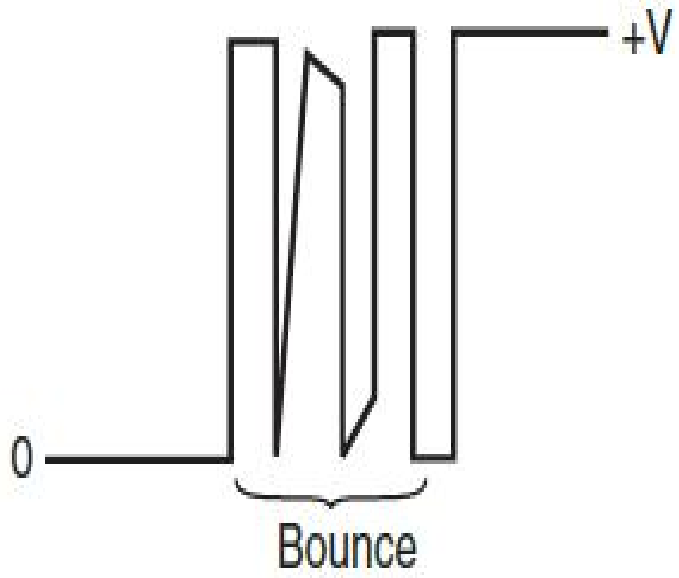
Synchronous and Asynchronous Inputs

- Synchronous inputs are those whose effect on the flip-flop output is synchronized with the clock input. R, S, J, K and D inputs are all synchronous inputs. Asynchronous inputs are those that operate independently of the synchronous inputs and the input clock signal.
- They force the flip-flop output to go to a predefined state irrespective of input.
- i.e. PRESET and CLEAR.
- are not active simultaneously.
- When it is desired that the flip-flop functions as per its synchronous inputs, the asynchronous inputs are kept in their inactive state.

Flip-Flop Applications

Switch De-bouncing

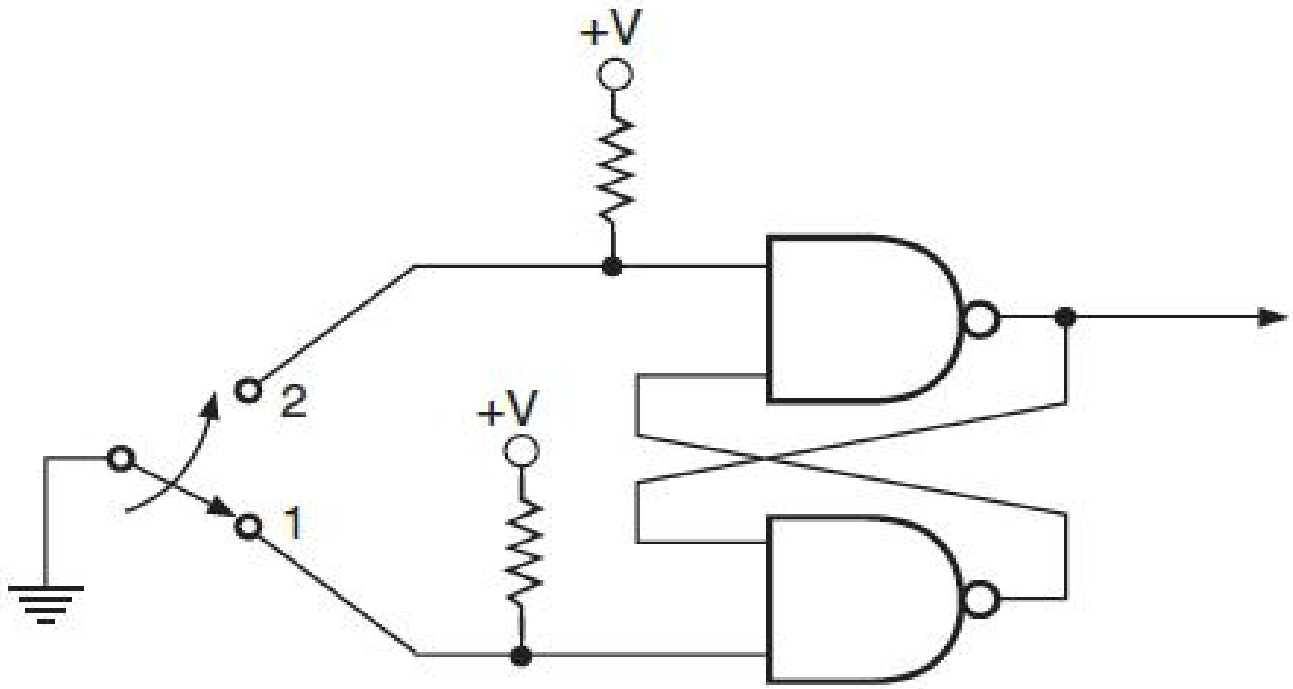


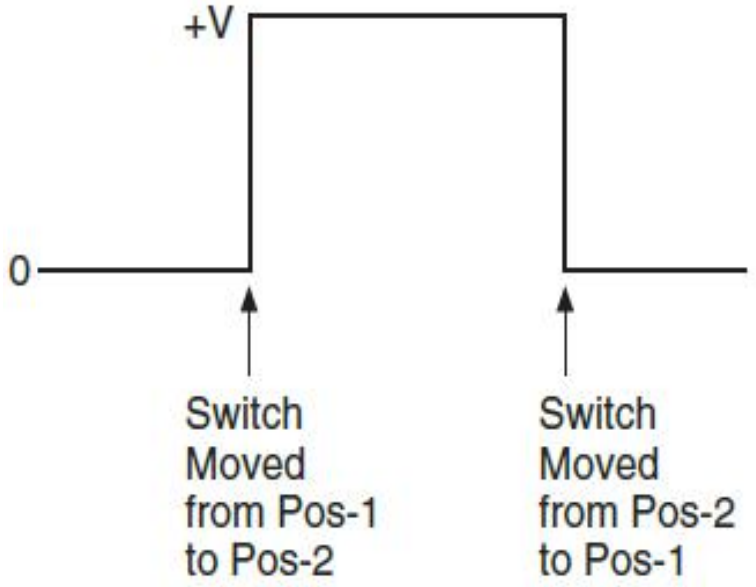


Switch bounce phenomenon→ **It is unacceptable for many digital circuit applications.**

A NAND or a NOR latch can solve this problem and provide a clean output transition.

Flip-Flop Applications → Switch De-bouncing





Switch de-bounce circuit.

Flip-Flop Applications → Switch De-bouncing

- When the switch is in position 1, the output is at a ‘0’ level. When it is moved to position 2, the output goes to a ‘1’ level.