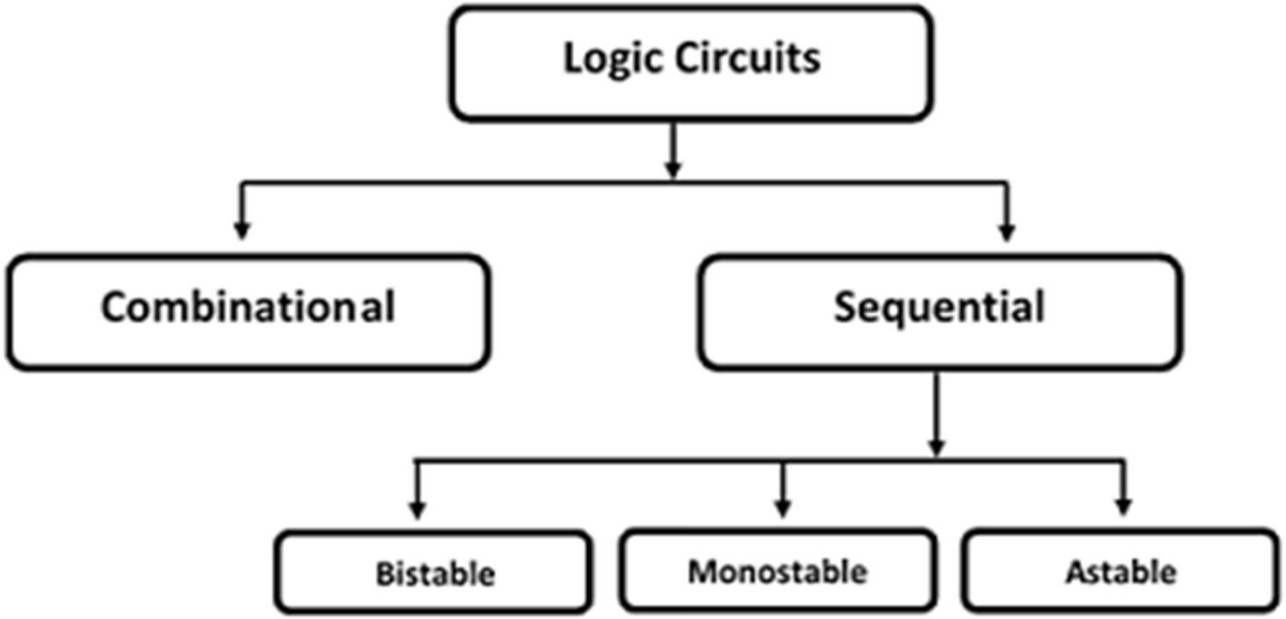
Unit 5

Sequential Circuit Design



Sequential circuits are of three types –
Bistable Bistable circuits have two stable operating points and will be in either of the states. Example Memory cells, latches, flip-flops and registers.
Monostable Monostable circuits have only one stable operating point and even if they are temporarily perturbed to the opposite state, they will return in time to their stable operating point. Example: Timers, pulse generators.
Astable – circuits have no stable operating point and oscillate between several states. Example Ring oscillator.

Combinational 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

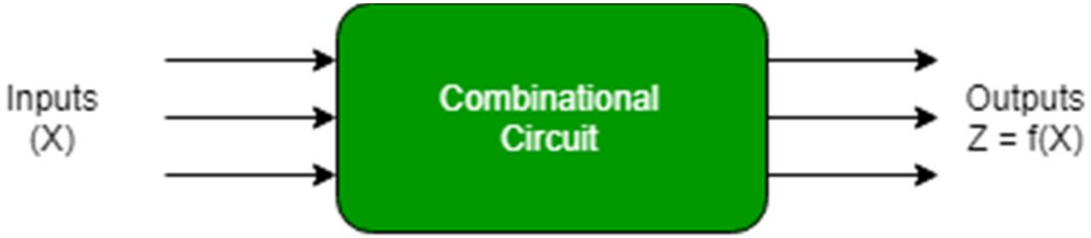


Figure: Combinational Circuits

Examples – Encoder, Decoder, Multiplexer, Demultiplexer

Sequential Circuit

 Sequential circuits are those which are dependent on clock cycles and depends on present as well as past inputs to generate any output.

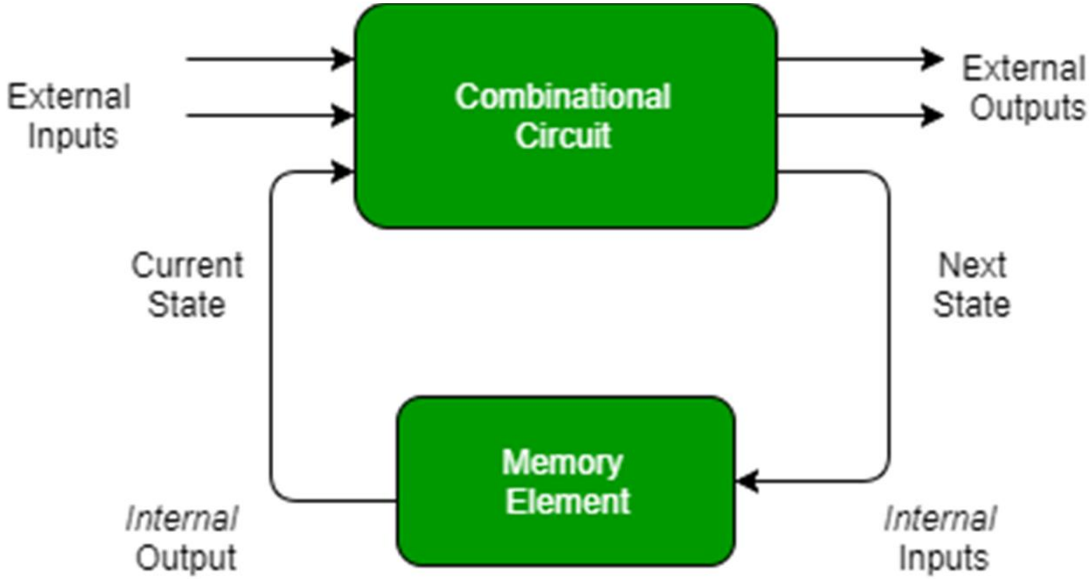


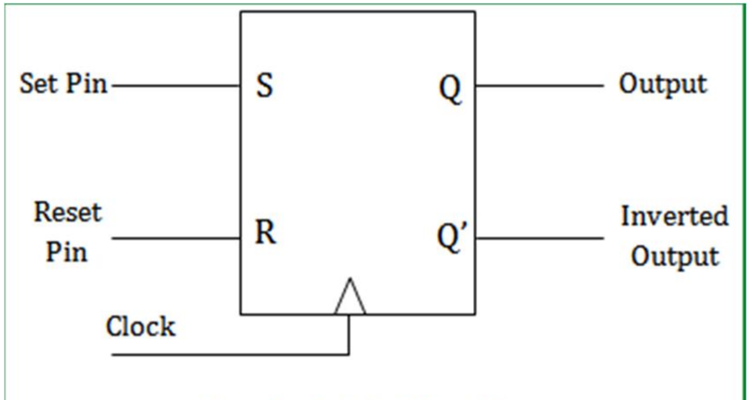
Figure: Sequential Circuit

Example: Flipflop ,FSM, pipeline

Key	Combinational Circuit	Sequential Circuit
Definition	A Combinational Circuit is a type of circuit in which the output is independent of time and only relies on the input present at that particular instant.	A Sequential circuit is a type of circuit where output not only relies on the current input but also depends on the previous output.
Feedback	Since output does not depend on the time instant, no feedback is required for its next output generation.	The output relies on its previous feedback so output of previous input is being transferred as feedback used with input for next output generation.
Performance	As the input of current instant is only required in case of Combinational circuit, it is faster and better in performance as compared to that of Sequential circuit.	Sequential circuits are comparatively slower and has low performance as compared to that of Combinational circuit.
Complexity	No implementation of feedback makes the combinational circuit less complex as compared to sequential circuit.	The implementation of feedback makes sequential circuit more complex as compared to combinational circuit.
Elementary Blocks	The elementary building blocks of a combinational circuit are its logic gates.	The building blocks of a sequential circuit are the logic gates along with flip flops.
Operation	Combinational circuits are mainly used for arithmetic as well as Boolean operations.	Sequential circuits are mainly used for storing data.

SR Flip-flop

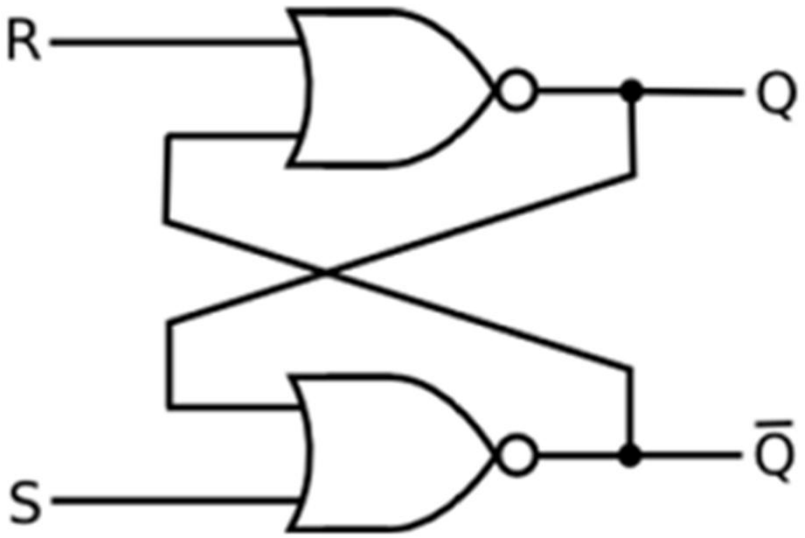
- SR Flip-flops were used in common applications like MP3 players, Home theatres, Portable audio docks, and etc. But now-a-days JK and D flip-flops are used instead, due to versatility. SR latch can be built with NAND gate or with NOR gate.
- Whenever the clock signal is LOW, the inputs S and R are never going to affect the output. The clock has to be high for the inputs to get active. Thus, SR flip-flop is a controlled Bi-stable latch where the clock signal is the control signal.



Symbol: SR Flip-flop

CMOS Logic Circuits

- SR flip-flop based on NOR Gate



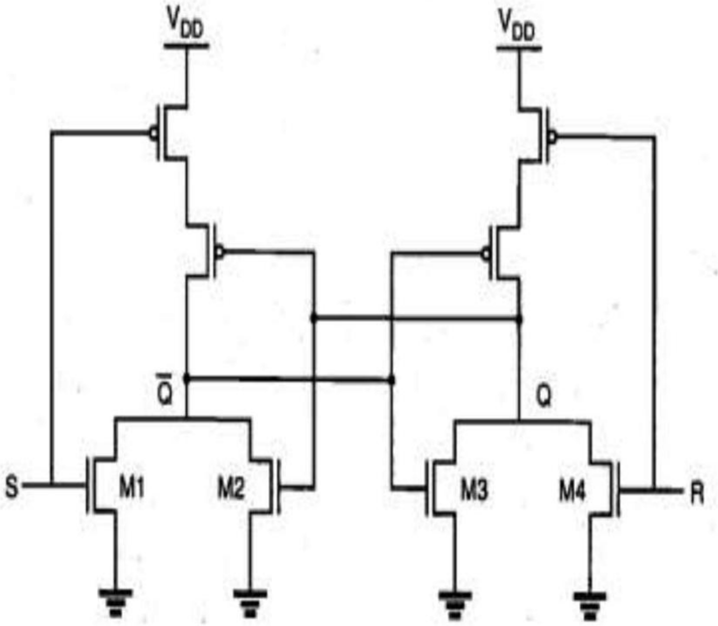
Truth table of NOR based SR Latch is given in table.

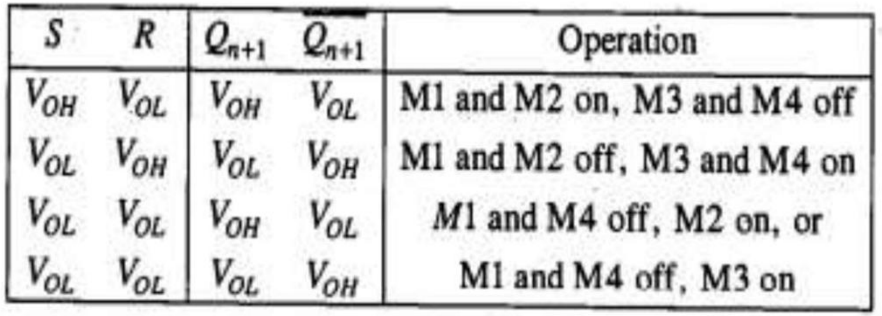
S	**R**	**Q**	$\overline{Q}$	**Operation**
0	0	Q	$\overline{Q}$	Hold
1	0	1	0	Set
0	1	0	1	Reset
1	1	0	0	Not allowed

- If the set input (S) is equal to logic "1" and the reset input is equal to logic "0." then the output Q will be forced to logic "1". While $\bar{Q}$ is forced to logic "0". This means the SR latch will be set, irrespective of its previous state.
- Similarly, if S is equal to "0" and R is equal to "1" then the output Q will be forced to "0" while $\bar{Q}$ is forced to "1". This means the latch is reset, regardless of its previously held state. Finally, if both of the inputs S and R are equal to logic "1" then both output will be forced to logic "0" which conflicts with the complementarity of Q and $\bar{Q}$.
- Therefore, this input combination is not allowed during normal operation.

CMOS SR flip-flop based on NOR gate

SR Latch





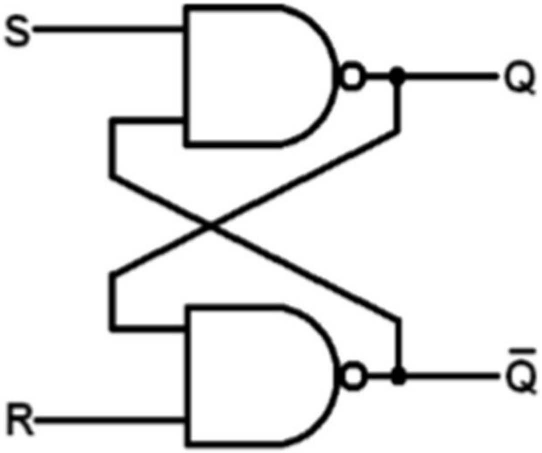
S	R	Q_{n+1}	$\overline{Q}_{n+1}$	Operation
V_{OH}	V_{OL}	V_{OH}	V_{OL}	M1 and M2 on, M3 and M4 off
V_{OL}	V_{OH}	V_{OL}	V_{OH}	M1 and M2 off, M3 and M4 on
V_{OL}	V_{OL}	V_{OH}	V_{OL}	M1 and M4 off, M2 on, or
V_{OL}	V_{OL}	V_{OL}	V_{OH}	M1 and M4 off, M3 on

If the set input (S) is equal to VOH and the reset input (R) is equal to VOL, both of the parallel-connected transistors Ml and M2 will be on. Consequently, the voltage on node Q will assume a logic-low level of VOL = 0.

At the same time, both M3 and M4 are turned off, which results in a logic-high voltage VOH at node Q. If the reset input (R) is equal to VOH and the set input (S) is equal to VOL, the situation will be reversed (Ml and M2 turned off and M3 and M4 turned on).

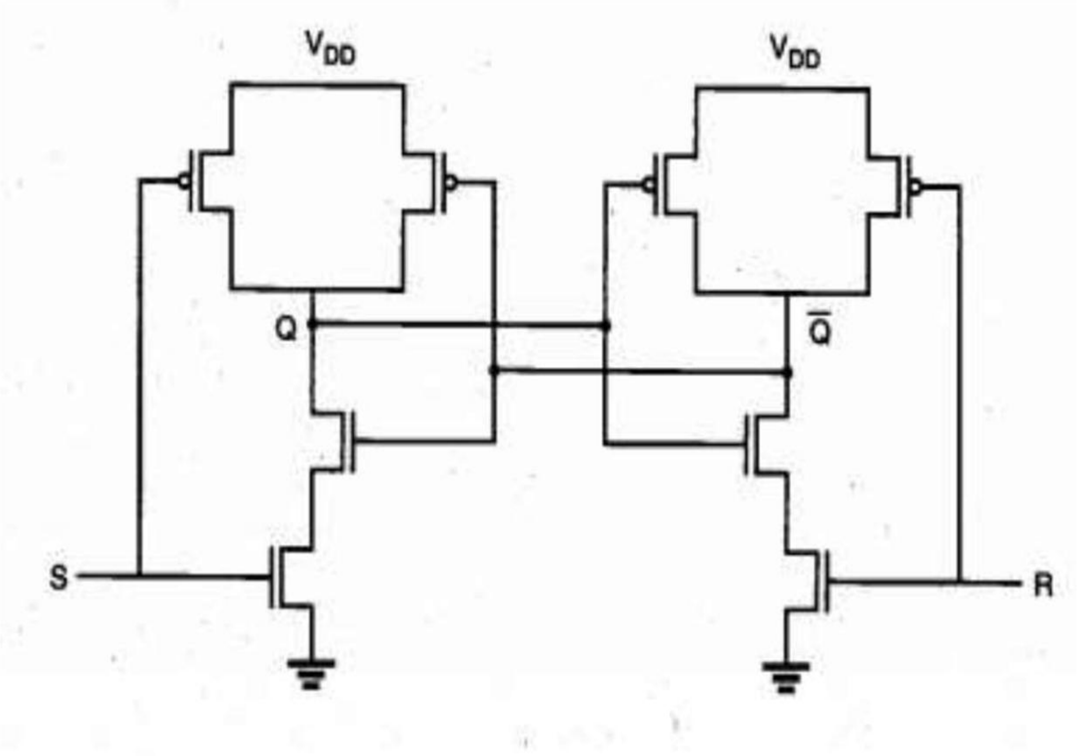
When both of the input voltages are equal to VOL' on the other hand, there are two possibilities. Depending on the previous state of the SR latch, either M2 or M3 will be on, while both of the trigger transistors MI and M4 are off. This will generate a logic-low level of VOL = O at one of the output nodes, while the complementary output node is at VOH.

SR flip-flop based on NAND Gate



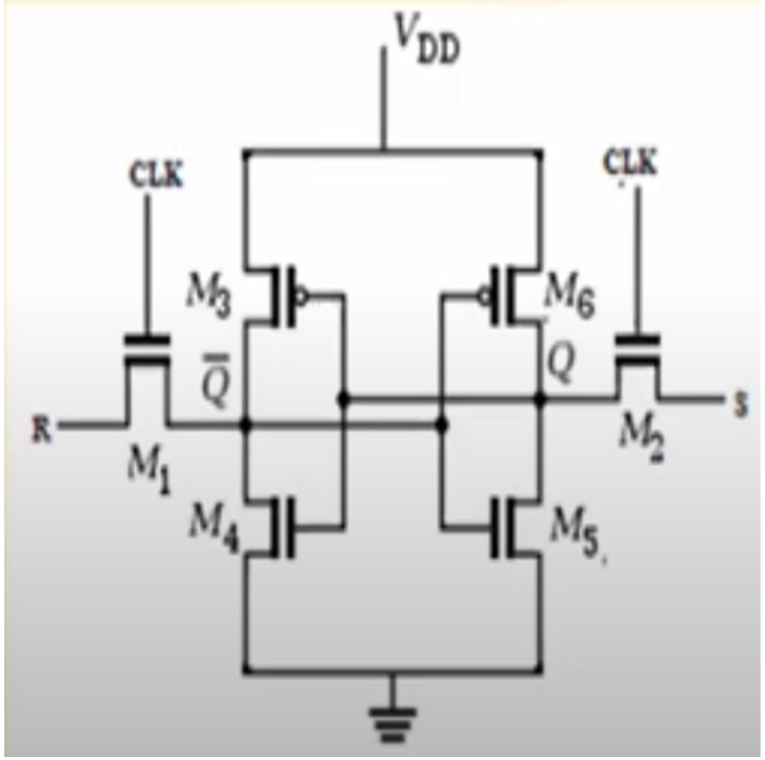
S	R	Q	Q	
0	0	NC	NC	No change. Latch remained in present state.
1	0	1	0	Latch SET.
0	1	0	1	Latch RESET.
1	1	0	0	Invalid condition.

SR flip-flop based on NAND Gate

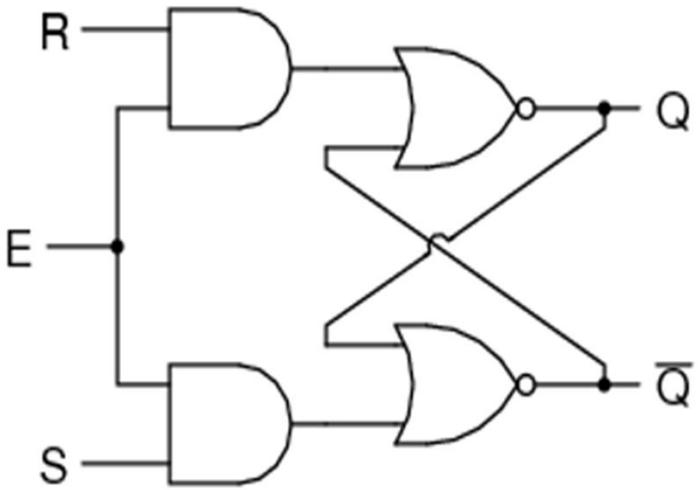


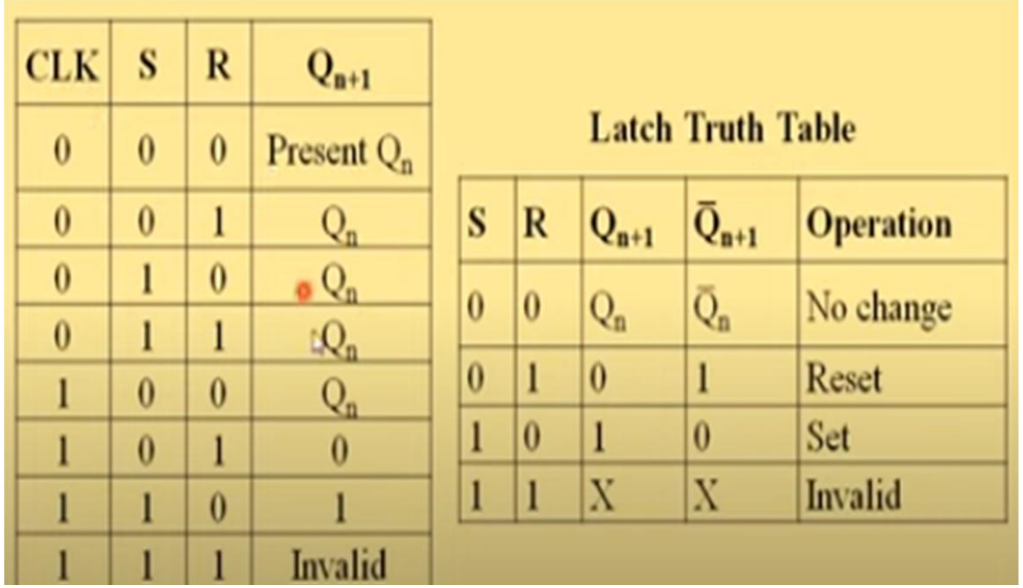
S-Rflip-flop using 6 transistor

- When clk=0,M1 & M2 will be in off condition
- When clk =1, S=1,M2 on and Q=1.This Q will be input to M3 M4 transistor, So M3 will be off, M4 will be on Thus Q’= 0
- When clk =1, R=1,M1 On and Q’=1.This Q’ will be input to M5 M6 transistor, So M6 will be off, M5 will be on Thus Q = 0
- When clk =1, S=1,R=1,M1 and M2 will turn On, Both Q=1 and Q’=1, this is not possible so invalid state.



Clocked SR flip-flop



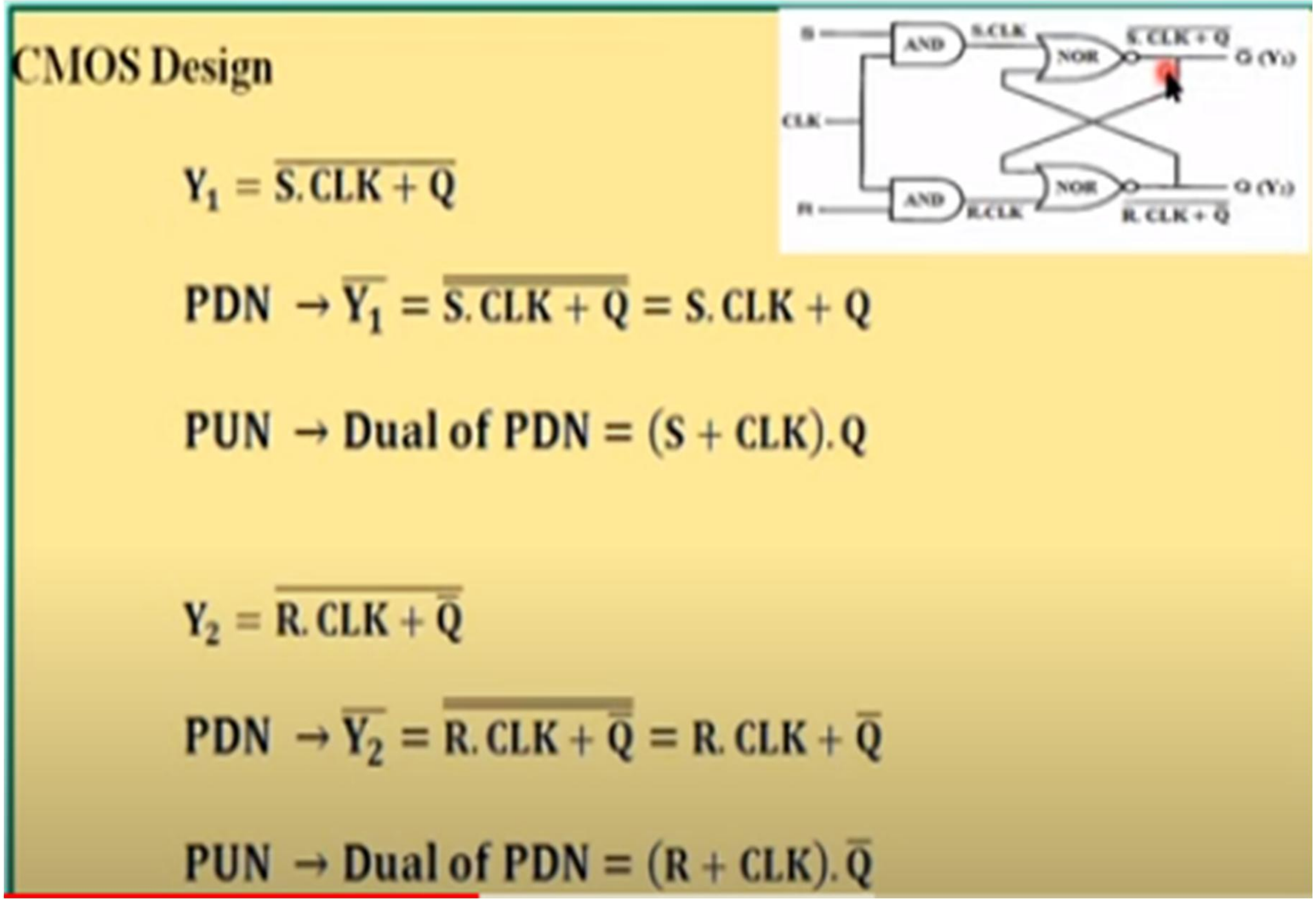


CLK	S	R	Q_{n+1}
0	0	0	Present Q_n
0	0	1	Q_n
0	1	0	Q_n
0	1	1	Q_n
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	Invalid

Latch Truth Table

S	R	Q_{n+1}	$\bar{Q}_{n+1}$	Operation
0	0	Q_n	$\bar{Q}_n$	No change
0	1	0	1	Reset
1	0	1	0	Set
1	1	X	X	Invalid

When CLK is low, two series terminals in N tree N are open and two parallel transistors in tree P are ON, thus retaining state in the memory cell.
When clock is high, the circuit becomes simply a NOR based CMOS latch which will respond to input S and R.



CMOS Design

$$Y_1 = \overline{S.CLK + Q}$$

$$\text{PDN} \rightarrow \overline{Y_1} = \overline{\overline{S.CLK + Q}} = S.CLK + Q$$

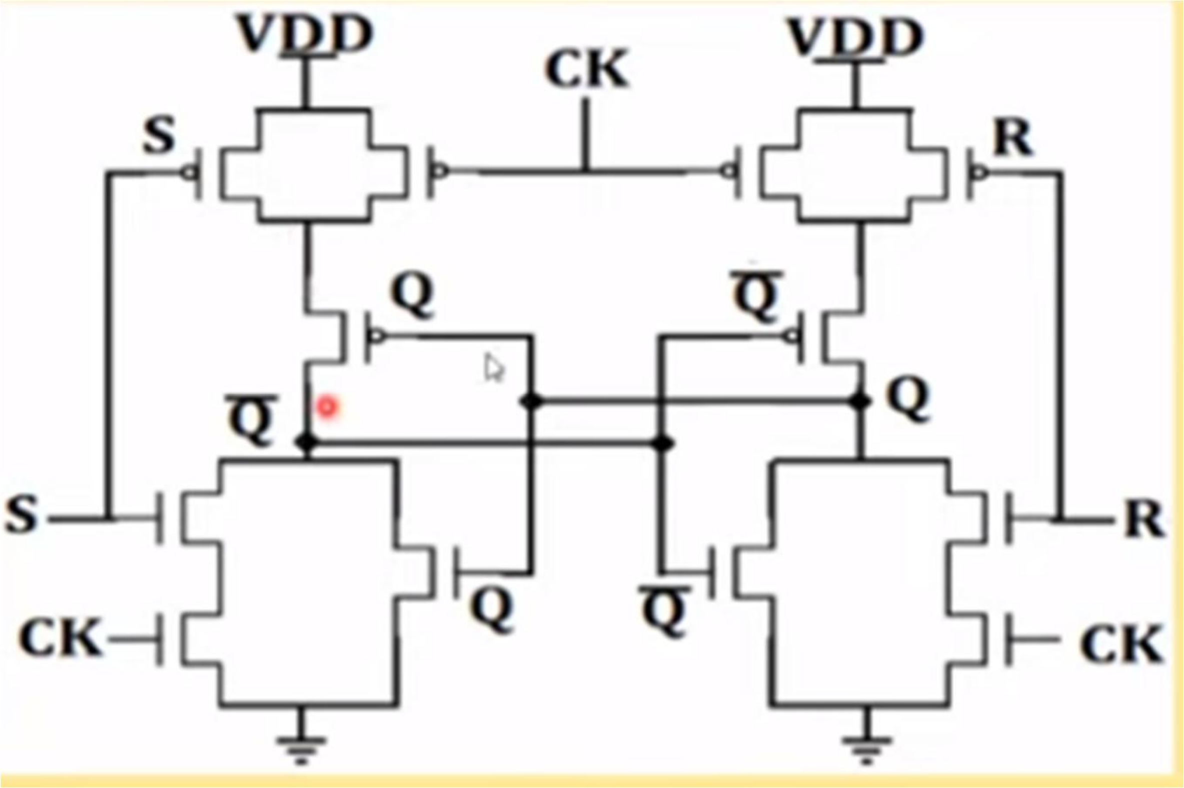
$$\text{PUN} \rightarrow \text{Dual of PDN} = (S + CLK).Q$$

$$Y_2 = \overline{R.CLK + \bar{Q}}$$

$$\text{PDN} \rightarrow \overline{Y_2} = \overline{\overline{R.CLK + \bar{Q}}} = R.CLK + \bar{Q}$$

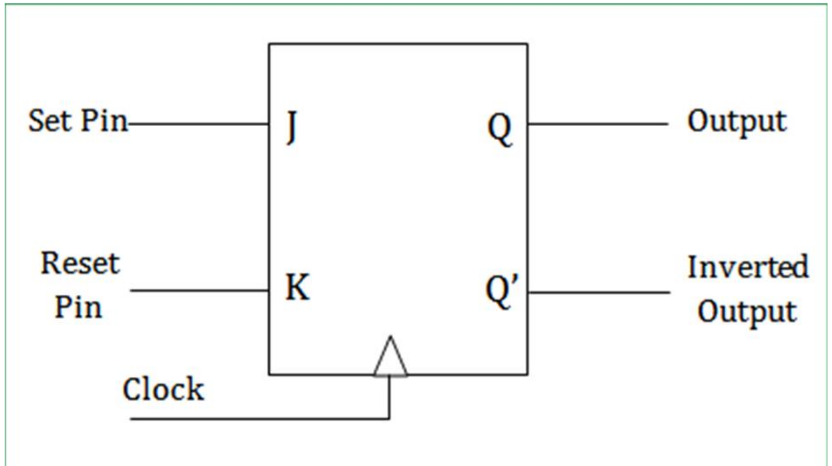
$$\text{PUN} \rightarrow \text{Dual of PDN} = (R + CLK).\bar{Q}$$

Clocked SR flip-flop



JK flip-flop

The name JK flip-flop is termed from the inventor Jack Kilby from texas instruments. Due to its versatility they are available as IC packages. The major applications of JK flip-flop are Shift registers, storage registers, counters and control circuits. Inspite of the simple wiring of D type flip-flop, JK flip-flop has a toggling nature



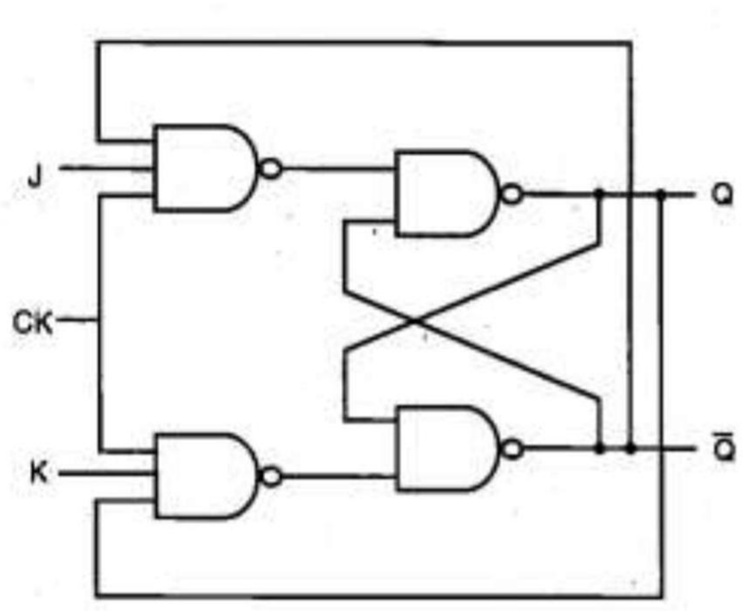
Symbol: JK Flip-flop

Whenever the clock signal is LOW, the input is never going to affect the output state. The clock has to be high for the inputs to get active. Thus, JK flip-flop is a controlled Bi-stable latch where the clock signal is the control signal.

JK flip-flop using NAND gate

Truth table of JK Flip Flop:

Clock	INPUT			OUTPUT	
	RESET	J	K	Q	Q’
X	LOW	X	X	0	1
HIGH	HIGH	0	0	No Change	
HIGH	HIGH	0	1	0	1
HIGH	HIGH	1	0	1	0
HIGH	HIGH	1	1	Toggle	
LOW	HIGH	X	X	No Change	
HIGH	HIGH	X	X	No Change	
HIGH	HIGH	X	X	No Change	

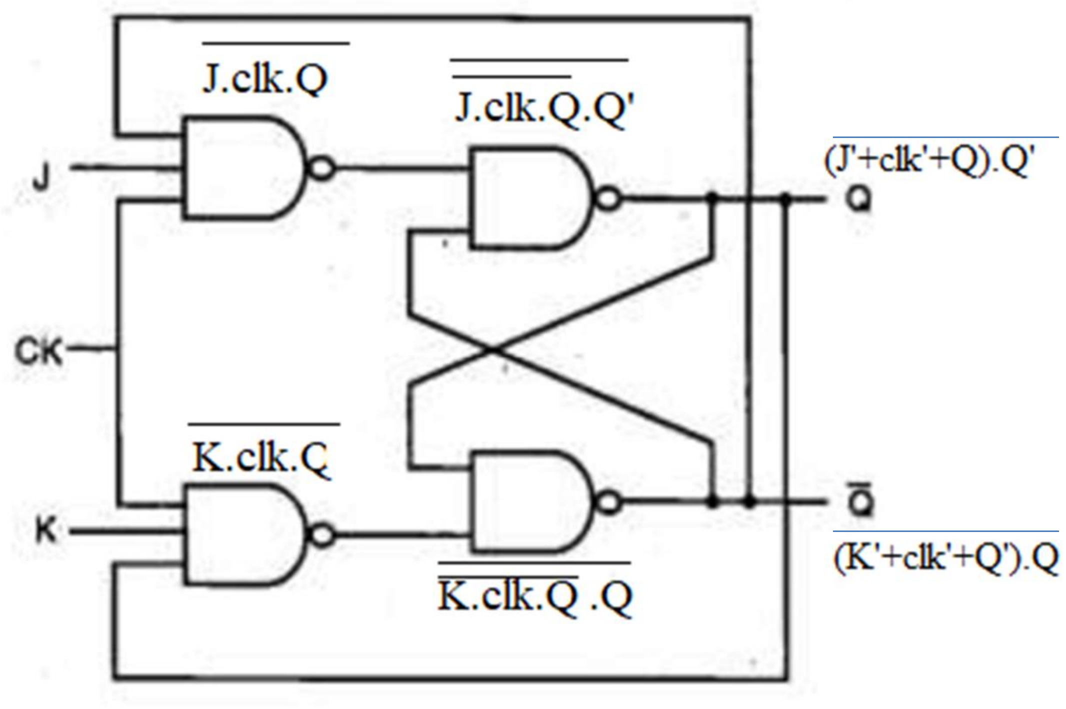


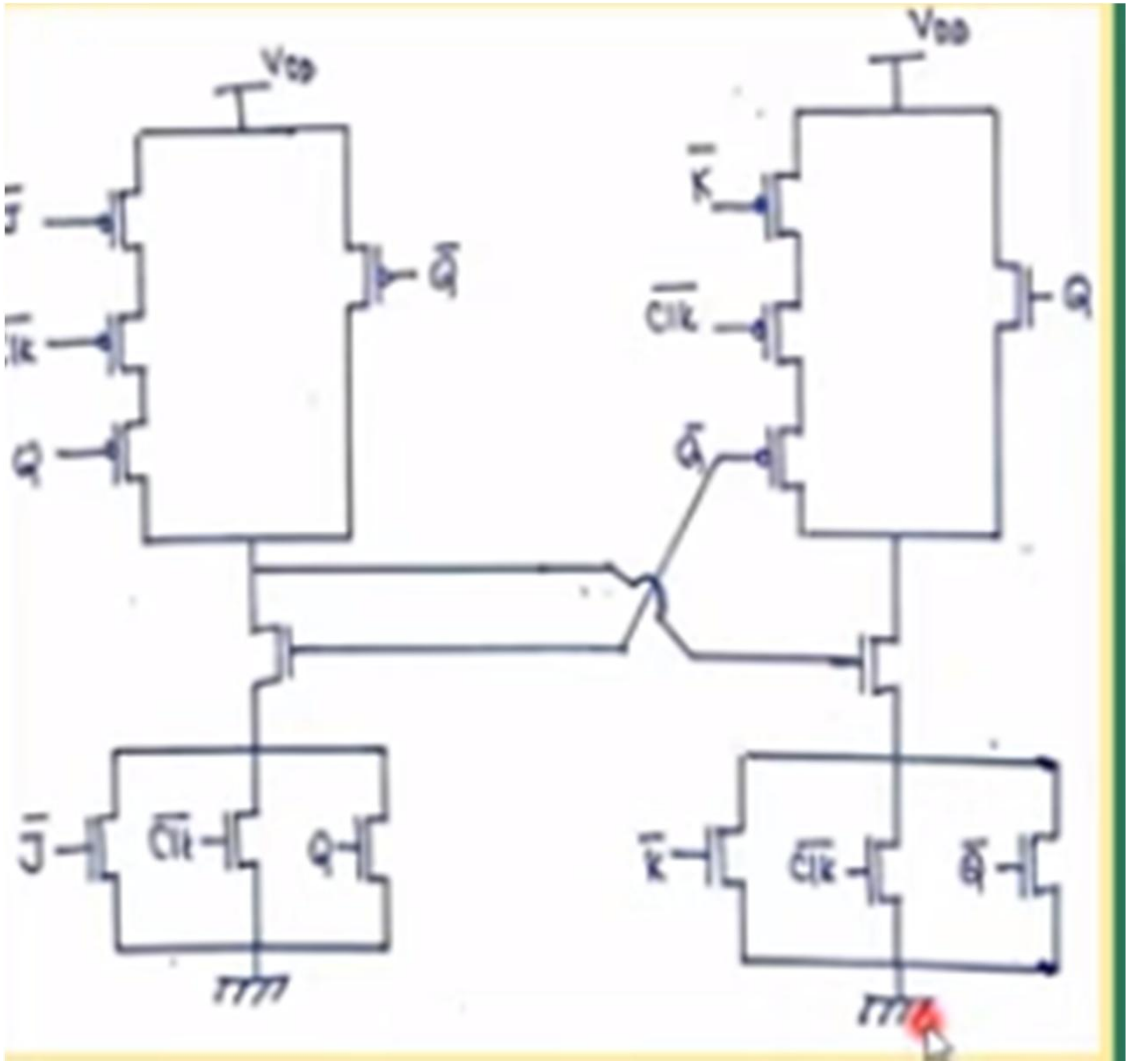
Input NAND gate – Truth Table:

INPUTS			OUTPUT
Input 1	Input 2	Input 13	Output 12
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

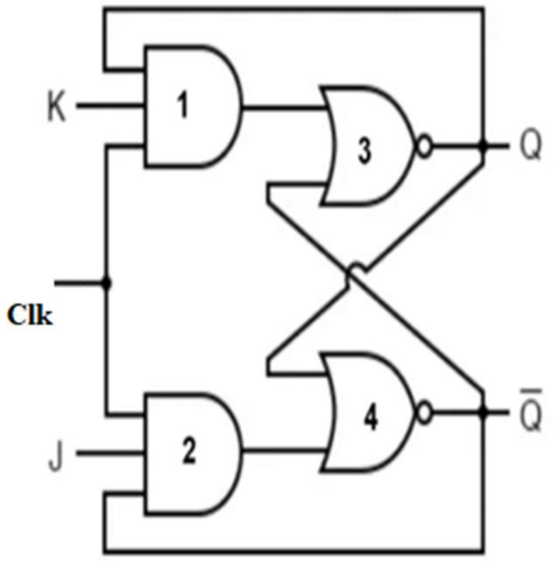
Input NAND gate – Truth Table:

INPUT		OUTPUT
Input 1	Input 2	Output 3
0	0	1
0	1	1
1	0	1
1	1	0


$\overline{J.clk.Q}$
$\overline{\overline{J.clk.Q}.Q'}$
$\overline{(J'+clk'+Q).Q'}$
J
Q
CK
$\overline{K.clk.Q}$
K
$\overline{Q}$
$\overline{(K'+clk'+Q').Q}$
$\overline{\overline{K.clk.Q}.Q}$


VDD
J̄
C̄lk
Q
Q̄
J̄
C̄lk
Q
VDD
K̄
C̄lk
Q̄
Q
K̄
C̄lk
Q̄

JK flip-flop using NOR gate



Case 1: J=0, K=0
Gate1 = 0, Gate2 = 0, Gate3/Q(n+1) = Q,
Gate4/Q(n+1)’ = Q’

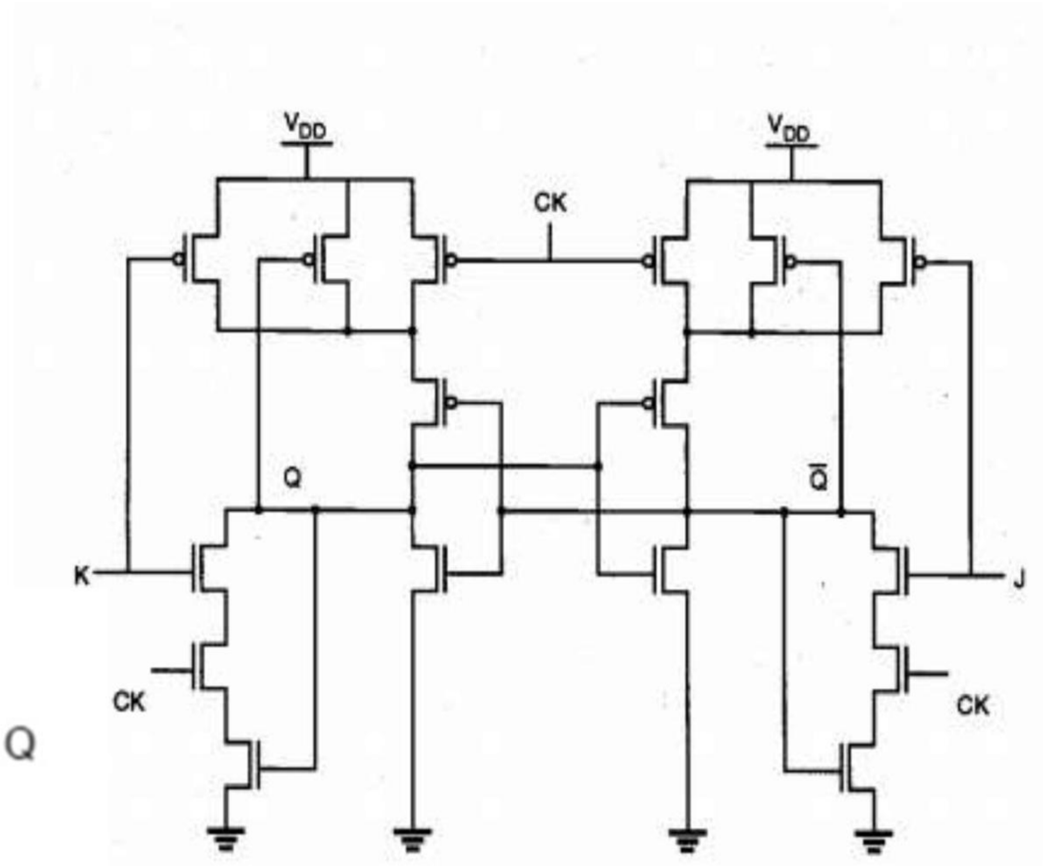
Case 2: J=0, K=1
Gate1 = Q, Gate2 = 0, Gate3/Q(n+1) = 0,
Gate4/Q(n+1)’ = 1

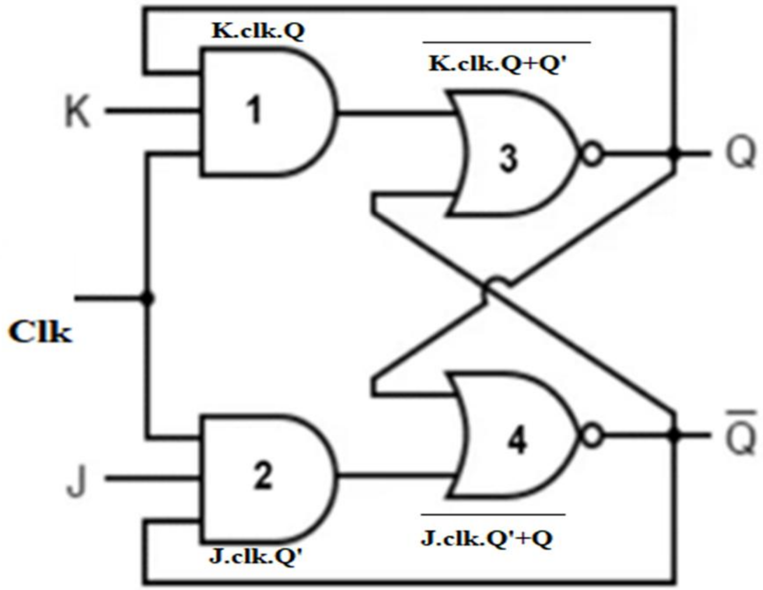
Case 3: J=1, K=0
Gate1 = 0, Gate2 = Q’, Gate4/Q(n+1)’ = 0,
Gate3/Q(n+1) = 1

Case 4: J=1, K=1
Gate1 = Q, Gate2 = Q’, Gate4/Q(n+1)’ = 0,
Gate3/Q(n+1) = Q’

JK Flip-Flop Truth Table

J	K	Q(n+1)	State
0	0	Qn	No Change
0	1	0	RESET
1	0	1	SET
1	1	Qn’	TOGGLE

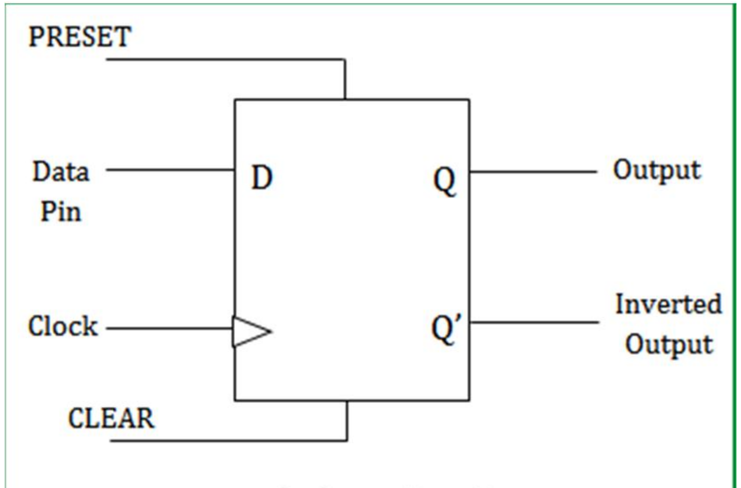




D Flip-flops

D Flip-flops are used as a part of memory storage elements and data processors as well. D flip-flop can be built using NAND gate or with NOR gate. Due to its versatility they are available as IC packages. The major applications of D flip-flop are to introduce delay in timing circuit, as a buffer, sampling data at specific intervals. D flip-flop is simpler in terms of wiring connection compared to JK flip-flop

Clock	INPUT	OUTPUT	
	D	Q	Q’
LOW	x	0	1
HIGH	0	0	1
HIGH	1	1	0



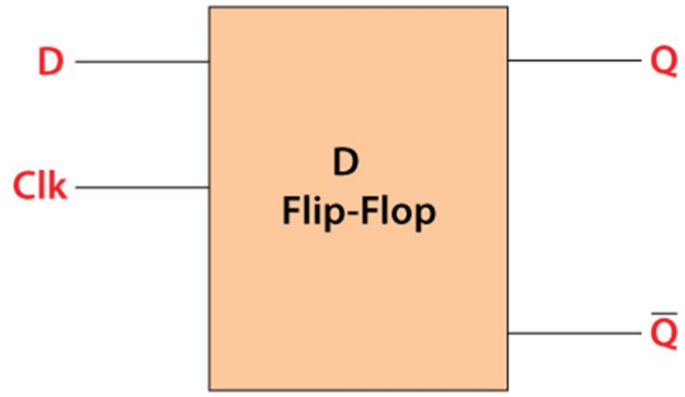
Symbol: D Flip-flop

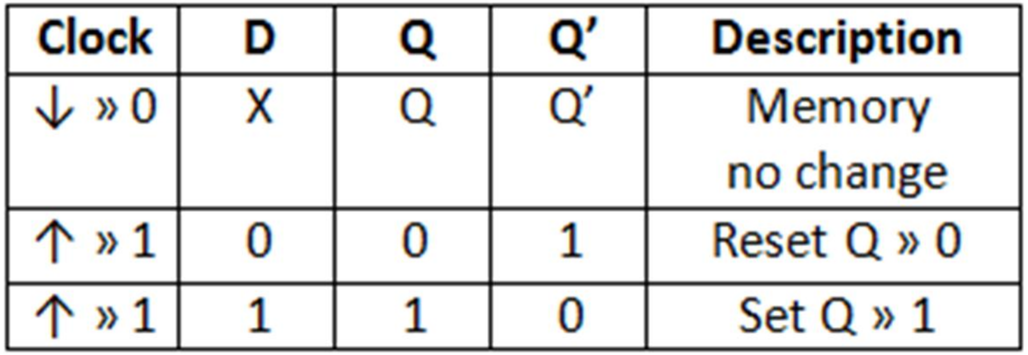
D Flip-flops

- In an active high SR Flip Flop is when S (Set) and R (Reset) both are 0, there will be no change in output of latch, and when both S and R are 1 output of latch is totally unpredictable (Invalid).
- So if both inputs of flip-flop are same there will either be a *No Change* or *Invalid* output condition.
- To avoid these conditions of inputs, there will be either SET or RESET conditions.
- There are many applications, where only SET and RESET conditions of latch are required.
- In these applications, use inputs (S and R) which are always complement of each other.
- This can be designed by a single input (S) to latch and R input achieved by inverting this S.
- This single input is called data input and it is labeled with D.
- Then this Data input, D is used in place of “Set” signal, and inverter is used to generate complementary “Reset” input thereby making a level-sensitive D-type flip-flop from a level-sensitive SR-latch as now S = D and R = not D as shown.

- D Flip Flop is most important of all clocked flip-flops.

- In D flip flop, single input "D" is referred to as "Data" input.
- When data input is set to 1, flip flop is set, and when it is set to 0, flip flop is reset.
- But, output of flip flop would always change on every pulse applied to this data input.
- "CLOCK" or "ENABLE" input is used to avoid this for isolating data input from flip flop's latching circuitry.
- When clock input is set to true, D input condition is only copied to output Q.
- Truth Table for the D-type Flip Flop





Clock	D	Q	Q'	Description
↓ » 0	X	Q	Q'	Memory no change
↑ » 1	0	0	1	Reset Q » 0
↑ » 1	1	1	0	Set Q » 1

Symbols and indicates the direction of the clock pulse. D-type flip flop assumed these symbols as edge-triggers.

$Y_1 = \overline{D.CLK + Q}$

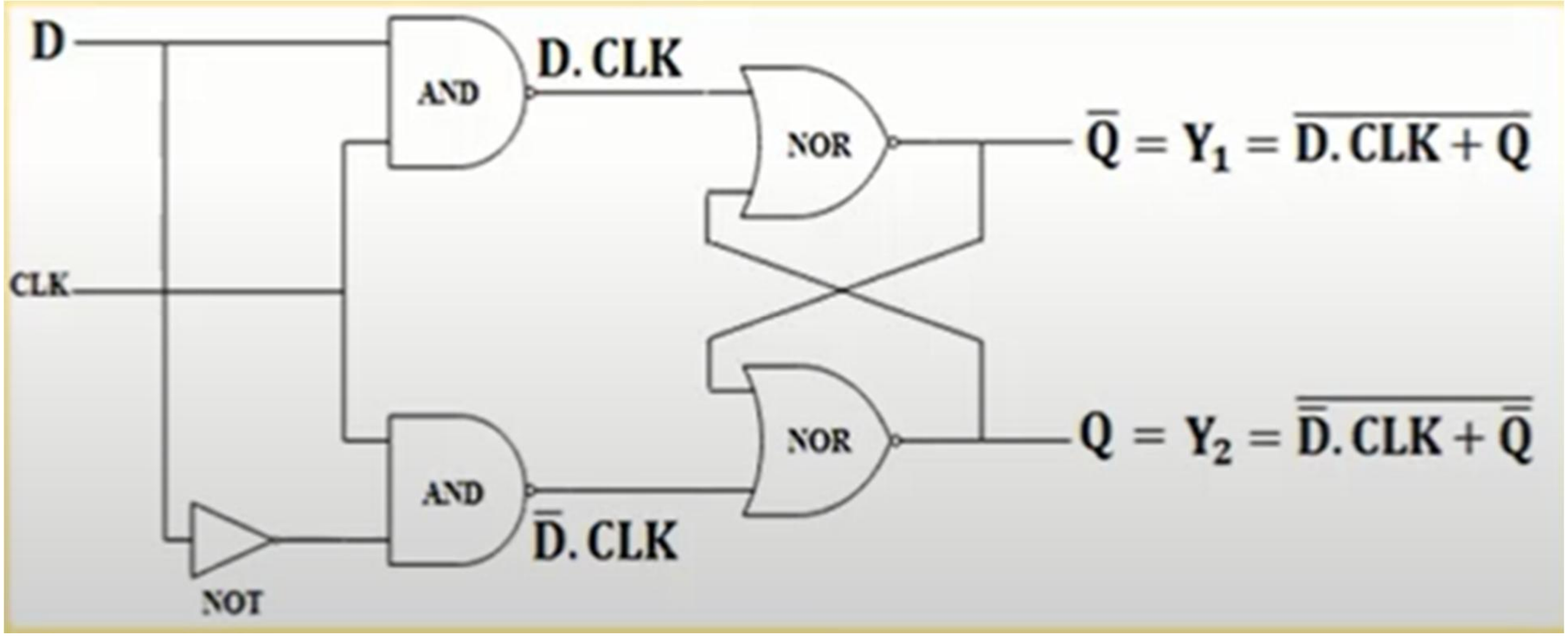
PDN $\rightarrow \overline{Y_1} = \overline{\overline{D.CLK + Q}} = D.CLK + Q$

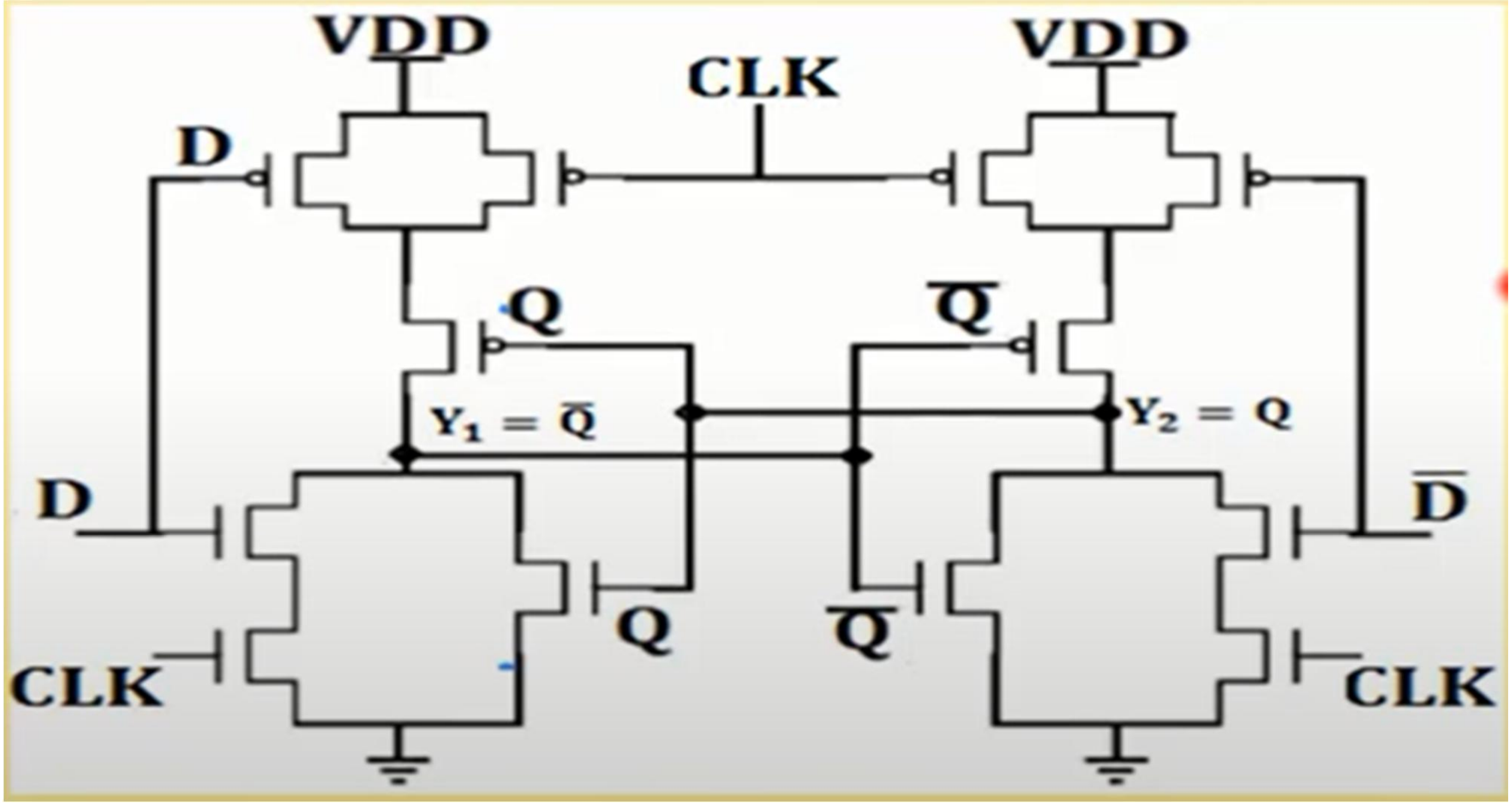
PUN $\rightarrow$ Dual of PDN $= (D + CLK).Q$

$Y_2 = \overline{\bar{D}.CLK + \bar{Q}}$

PDN $\rightarrow \overline{Y_2} = \overline{\overline{\bar{D}.CLK + \bar{Q}}} = \bar{D}.CLK + \bar{Q}$

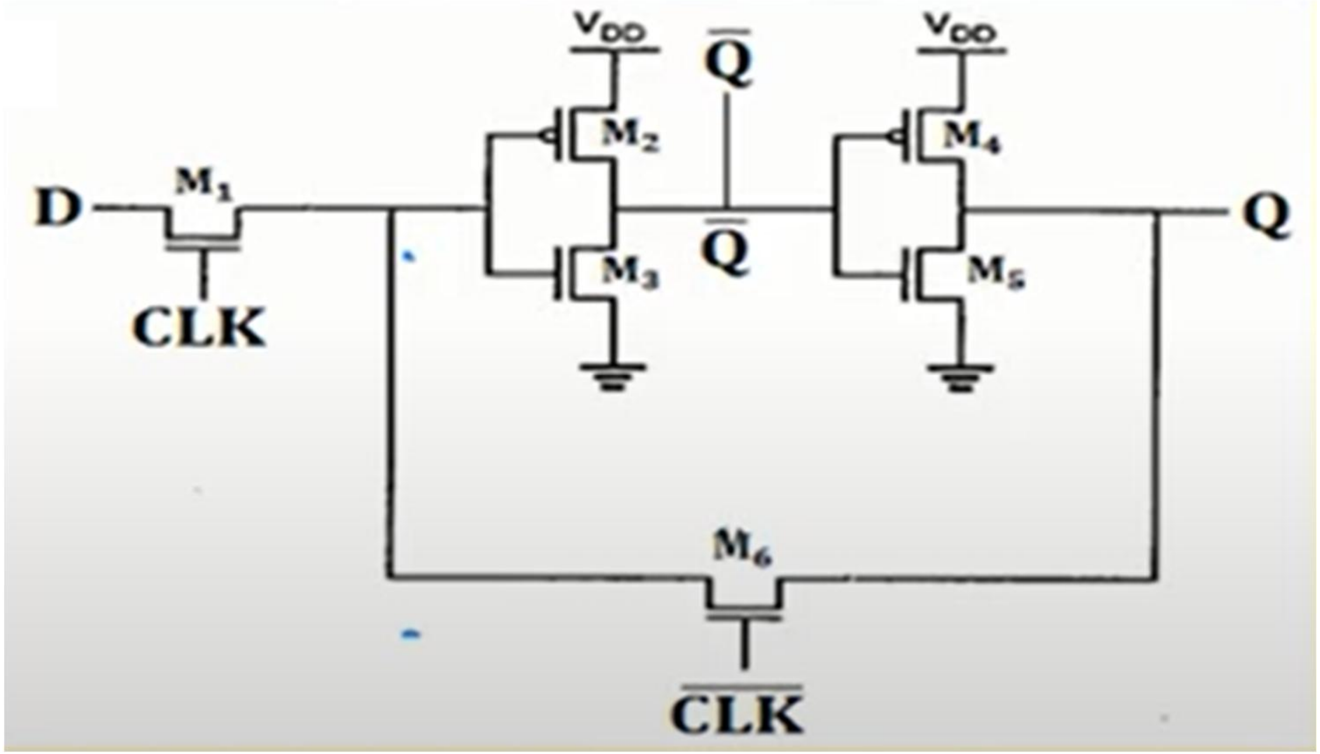
PUN $\rightarrow$ Dual of PDN $= (\bar{D} + CLK).\bar{Q}$


D
CLK
AND
AND
NOT
NOR
NOR
$D.CLK$
$\bar{D}.CLK$
$\bar{Q} = Y_1 = \overline{D.CLK + Q}$
$Q = Y_2 = \overline{\bar{D}.CLK + \bar{Q}}$


VDD
VDD
CLK
D
Q
$\overline{Q}$
$Y_1 = \overline{Q}$
$Y_2 = Q$
D
$\overline{D}$
Q
$\overline{Q}$
CLK
CLK

Simpler Implementation of D flipflop

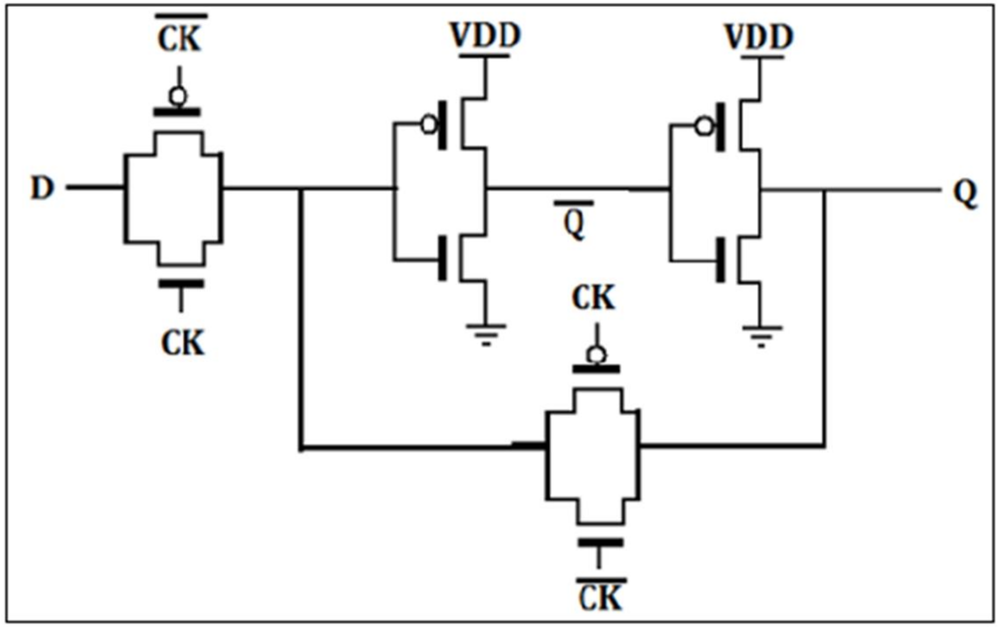
- It has many applications in digital circuit design for temporary storage of data or as a delay element.
- D flipflop can be designed with only 6 MOS transistors as shown below.
- In this circuit two CMOS inverters are connected back to back and clock and inverted clock are connected to two NMOS transistors.
- Output of first inverter is Q’ which is connected as an input to second inverter so that output Q will be complemented.



Dr.Pradnya Zode

- When CLK = 0 and CLK’ = 1, M1 (NMOS) will be OFF and M6 (NMOS) will be ON.
- Therefore the feedback loop from output Q to input of first inverter will be active. In this state is circuit will work as memory or storage device.

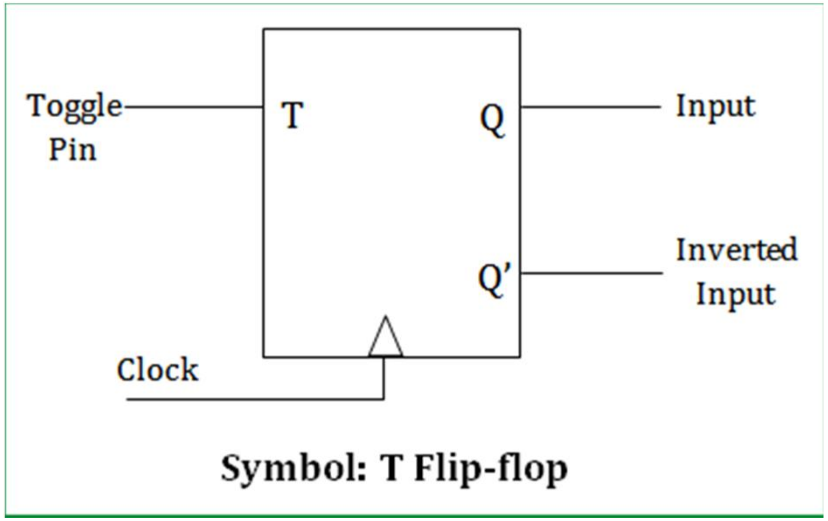
- When CLK = 1 and CLK’ = 0, M1 (NMOS) will be ON and M6 (NMOS) will be OFF.
- Therefore the feedback loop from output Q to input of first inverter will not be active.
- When D = 1 which is connected as an input to first inverter with M2(PMOS) is OFF and M3(NMOS) is ON
- So output Q’ will be 0
- Again Q’ is connected as an input to second inverter with M4(PMOS) is ON and M5(NMOS) is OFF.
- So that output Q will be 1 as get connected with VDD.
- Next when D = 0 which is connected as an input to first inverter with M2(PMOS) is ON and M3(NMOS) is OFF
- So output Q’ will be 1
- As Q’ is connected as an input to second inverter with M4(PMOS) is OFF and M5(NMOS) is ON.
- So that output Q will be 0 as get connected with GND.
- As soon as CLK will become logic 0 and CLK’ logic 1 the feedback loop will get activated and value on D input will get stored.



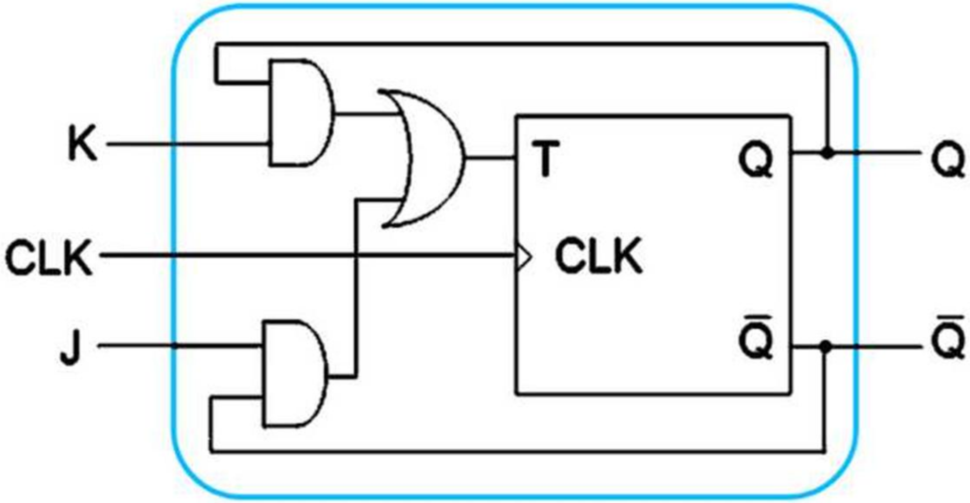
The D latch is normally, implemented with transmission gate (TG) switches as shown in the figure. The input TG is activated with CLK while the latch feedback loop TG is activated with CLK. Input D is accepted when CLK is high. When CLK goes low, the input is opencircuited and the latch is set with the prior data D.

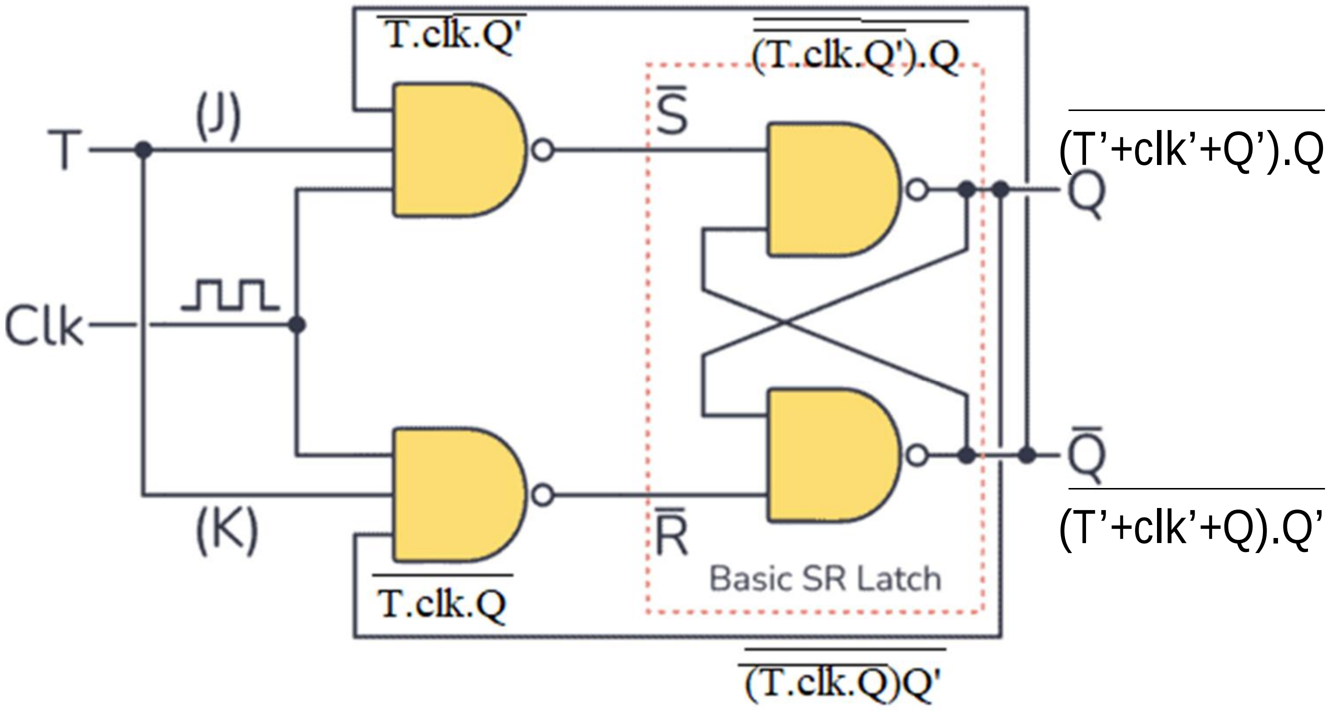
T flip-flop

The name T flip-flop is termed from the nature of toggling operation. The major applications of T flip-flop are counters and control circuits.



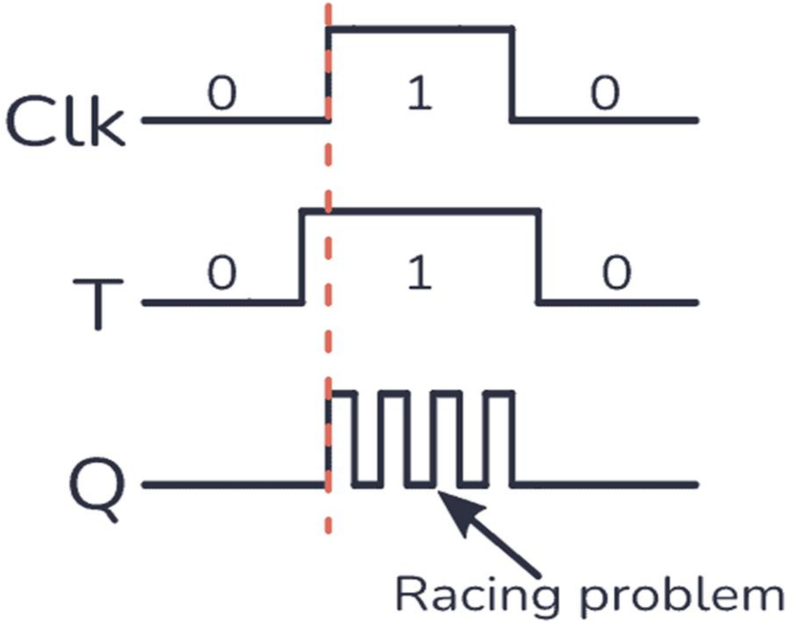
Symbol: T Flip-flop




$\overline{T.clk.Q'}$
$\overline{\overline{(T.clk.Q')}.Q}$
T
(J)
Clk
$\overline{S}$
$\overline{(T'+clk'+Q').Q}$
Q
$\overline{Q}$
$\overline{(T'+clk'+Q).Q'}$
$\overline{R}$
(K)
Basic SR Latch
$\overline{T.clk.Q}$
$\overline{\overline{(T.clk.Q)}Q'}$

Clock	INPUT		OUTPUT	
	RESET	T	Q	Q’
X	LOW	X	0	1
HIGH	HIGH	0	No Change	
HIGH	HIGH	1	Toggle	
LOW	HIGH	X	No Change	

clock pulse needs to go high, then low again before the output (Q) changes state. Otherwise, the Q output will toggle quickly between 1 and 0 during the entire positive pulse duration.


Clk
0
1
0
T
0
1
0
Q
Racing problem