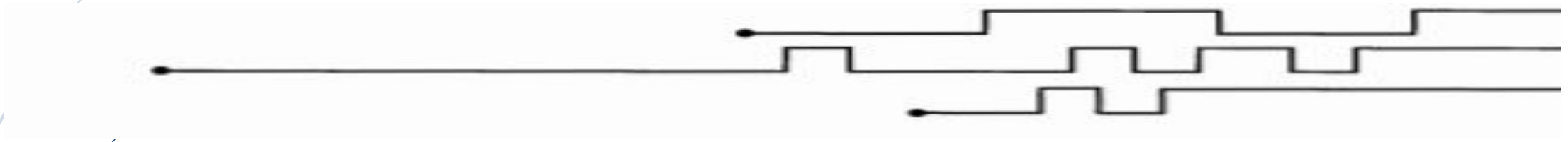
ensia

Digital systems



Chapter 4

Sequential Circuits

1

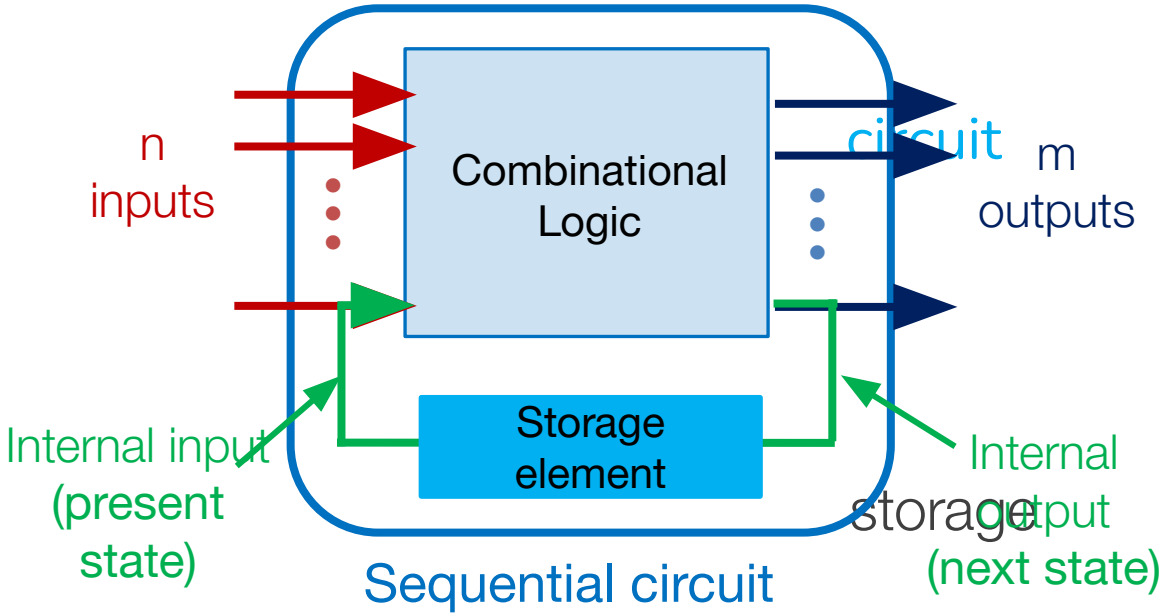
Dr. Iness NEDJI MILAT (Lecturer)
iness.nedji@ensia.edu.dz

Pr. Nasreddine LAGRAA
Nasredine,lagraa@ensia.edu.dz

First year – 2023/2024

Sequential vs. combinational circuit

- Combinational circuit output depends only on current inputs
- How can we design a circuit that stores information?
  - Use a feedback
- **Sequential circuit** is a combinational with feedbacks (storage elements).
  - EX : **Counters** and **Shift registers**.
- Some sequential circuits may not contain combinational circuits, but only elements
  - Ex : Registers (set of bits).



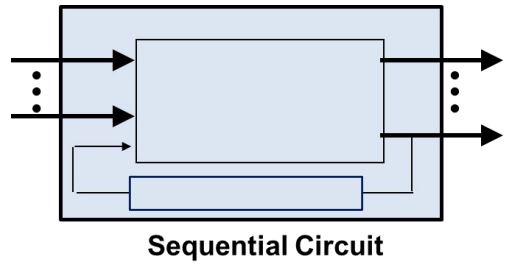
- The **output** of a sequential circuit depends not only on **current inputs** but also on the **previous outputs (states)**.

Storage elements

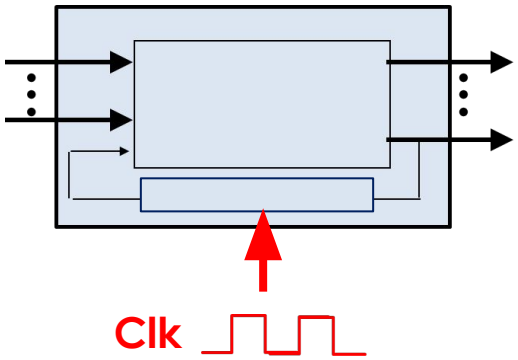
- Internal outputs of a sequential circuit are fed back via storage elements
- The **storage elements** (memory) are circuits that are capable of storing binary information.
- The binary information stored in these memory elements at any given time defines the state of the sequential circuit at that time.
- Each storage element memorize one bit with two states (0 or 1)

Types of sequential circuits

- Sequential circuits can be Asynchronous or Synchronous.
  - Asynchronous sequential circuits
    - change their states and output values whenever a change in input values occurs.

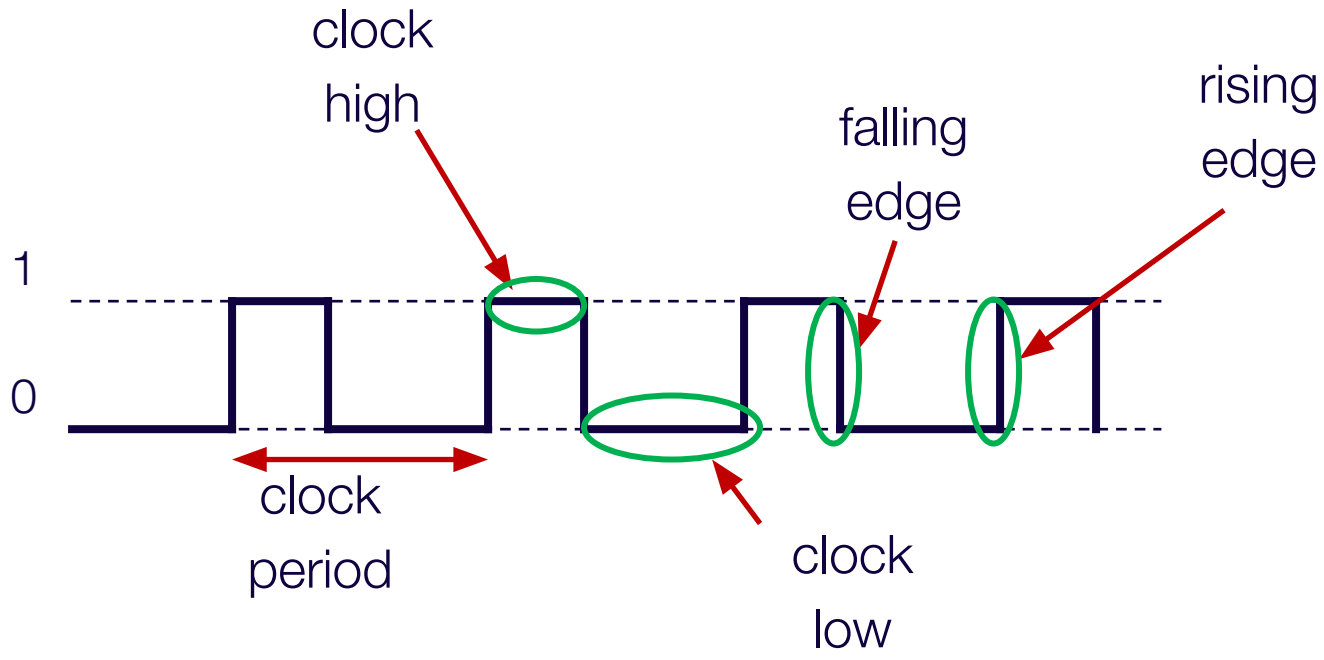


  - Synchronous sequential circuits
    - change their states and output values at fixed points of time, i.e. clock signals. (Clk = 1)



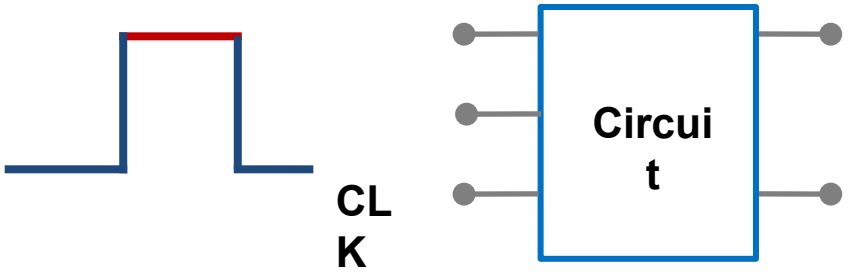
The clock

- In synchronous circuits, state values are controlled by clock signals.
  - A “clock” is a special circuit that sends electrical pulses through a circuit.
- Clocks produce electrical waveforms such as the one shown below.



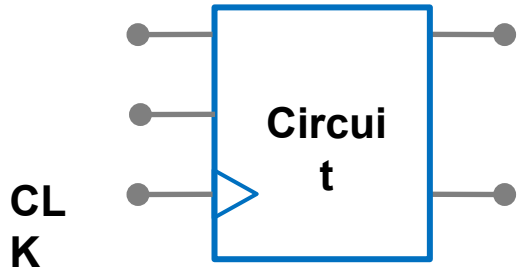
Clock Disciplines

- State changes occur in sequential circuits only when the clock ticks.
- Level sensitive
  - State changes when clock is high (or low)

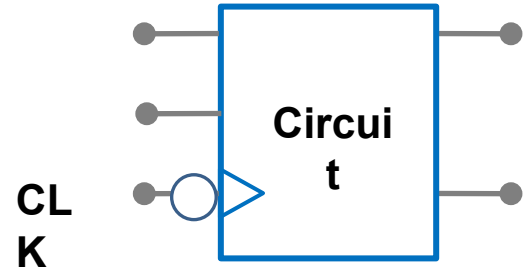


- Edge triggered
  - State changes at clock edge

positive edge-triggered



negative edge-triggered



Basic Storage Element

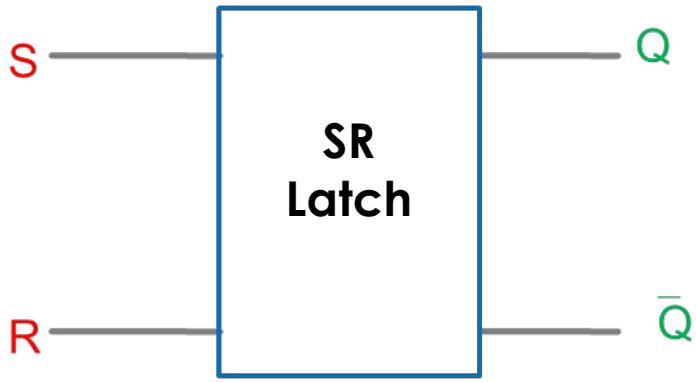
Basic storage elements

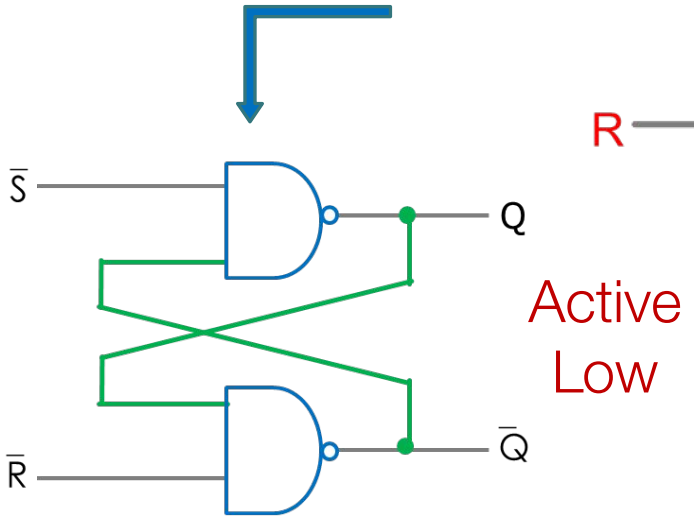
Generally, there are two types of storage elements used: *Latches*, and *Flip-Flops*.

- **Latches :** A latch is an **asynchronous** memory element whose excitation signals (inputs) control the state of the circuit.
  - A latch has two stages ***Set*** and ***Reset***. Set input sets the output to 1. Reset input sets the output to 0.
- **Flip-flops :** A flip-flop is a **synchronous** memory element that has clock signals control the state of the circuit.
  - Flip-flop is a latch with clock input signal

SR Latch

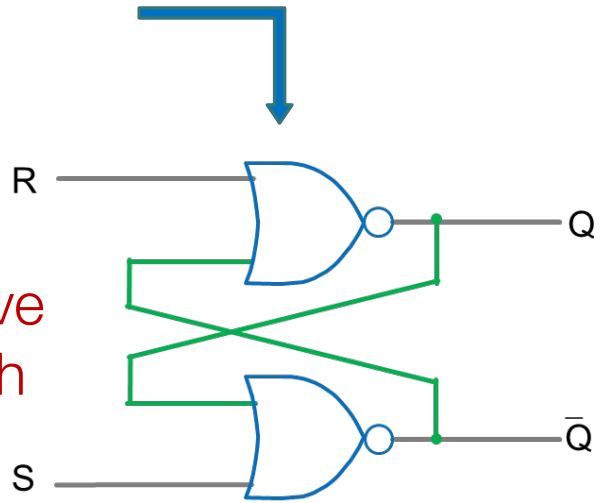
- The SR latch is the basic memory element. It has two input signals (S and R) and two outputs (Q and Q').
- Basic Latches can be constructed using two NAND gates or using two NOR gates





Active Low

Two cross-coupled NAND gates



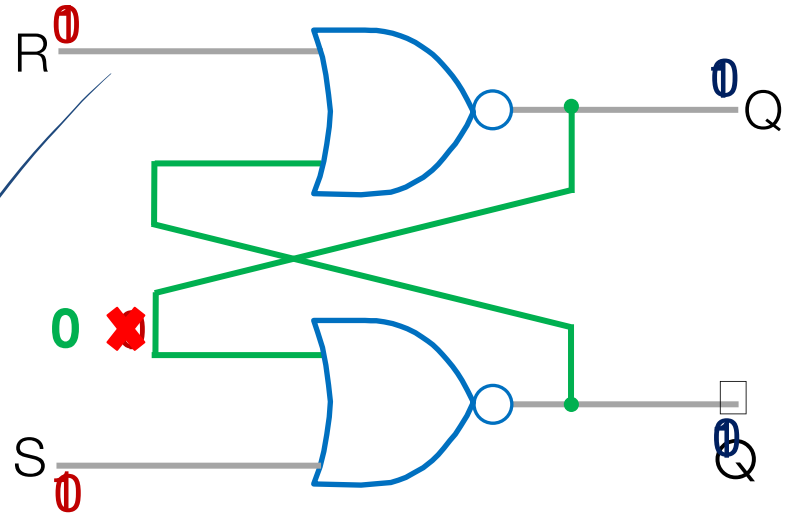
Active High

Two cross-coupled NOR gates

SR Latch

A	B	Or	Nor
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Active high



Present state			Next state			
S	R	Q_0	Q	Q'	Function	
0	0	0	0	1	$Q = Q_0$	Store (hold)
0	0	1	1	0	$Q = Q_0$	
0	1	0	0	1	$Q = 0$	Reset
0	1	1	0	1	$Q = 0$	
1	0	0	1	0	$Q = 1$	Set
1	0	1	1	0	$Q = 1$	
1	1	0	0	0	$Q = Q'$	Forbidden
1	1	1	0	0	$Q = Q'$	

Put 0 in bit

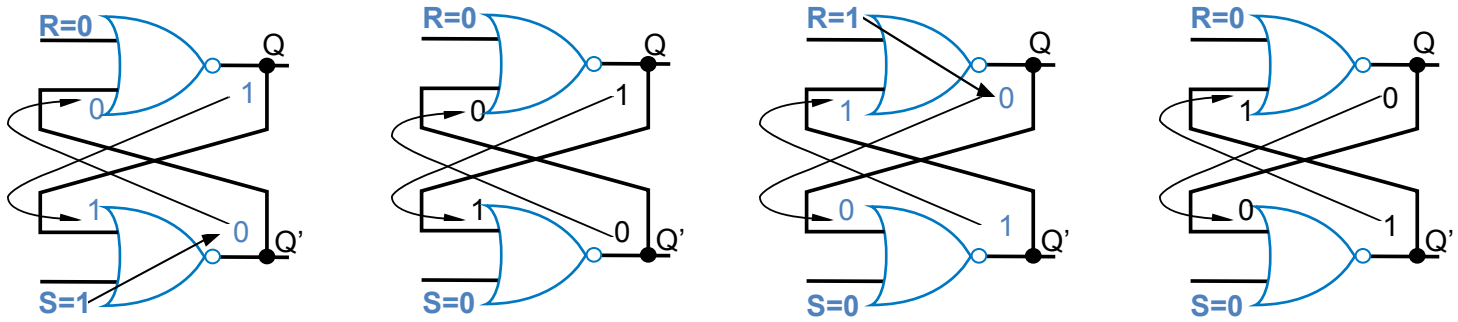
Put 1 in bit

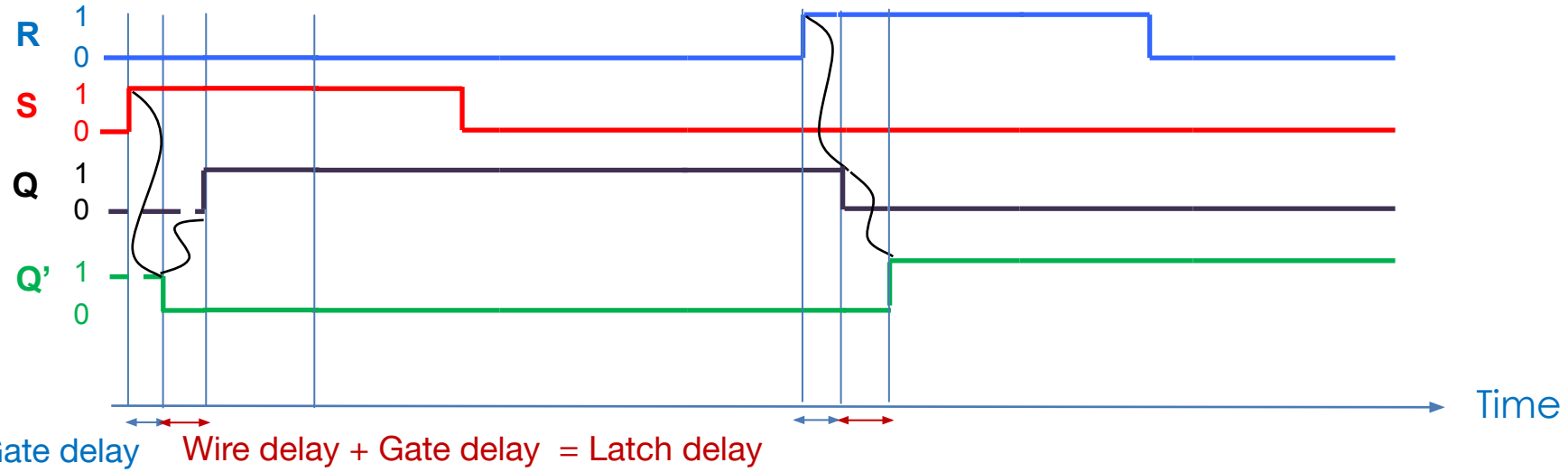
To avoid

Timing Diagram Representation

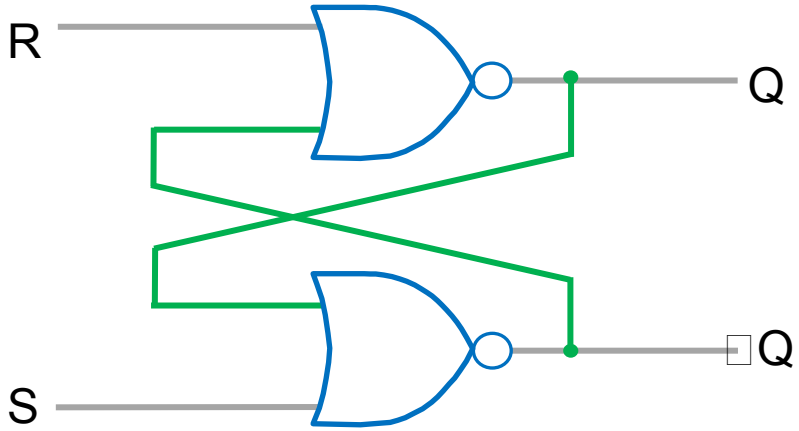
A	B	Or	Nor
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Active high





SR Latch with NOR



$Q(t)$ \ SR	00	01	11	10
0	0	0	X	1
1	1	0	X	1

S R Q_0	Q	Q'
0 0 0	0	1
0 0 1	1	0
0 1 0	0	1
0 1 1	0	1
1 0 0	1	0
1 0 1	1	0
1 1 0	x	x
1 1 1	x	x

characteristic equation

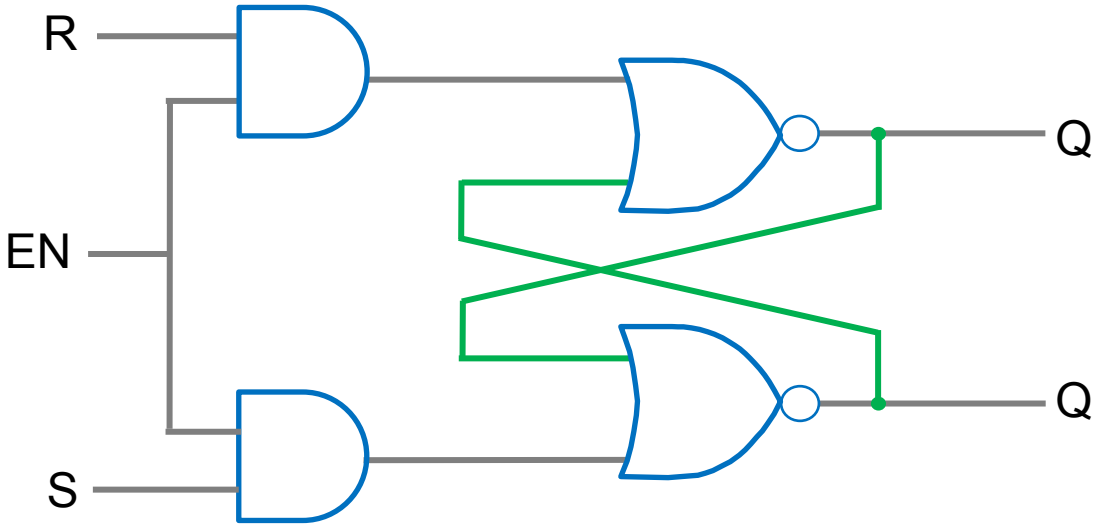
$$Q(t+1) = S + R' Q(t)$$

Next state Q+

Present state Q

SR gated Latch

- Similar to the SR latch but with the extra control input EN which enables or disables the operation of the S and R inputs.
  - When EN=1, the gated SR latch operates as an SR latch.
  - When EN=0, S and R are disabled and the circuit persists in the preceding state.



EN	S	R	Q	Q^+
0	x	x	0	0
0	x	x	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	x	0
1	1	0	x	1
1	1	1	x	NA

Takeaway

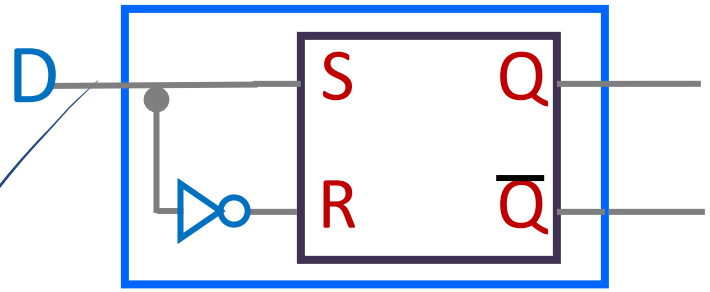
- The fact:
  - Set-Reset (SR) Latch can store one bit and we can change the value of the stored bit. But,
- The problem:
  - SR Latch has a forbidden state.
- Question:
  - How do we avoid the forbidden state of SR Latch?
- Response:
  - Ensure that inputs S and R are never equal to 1 at the same time,

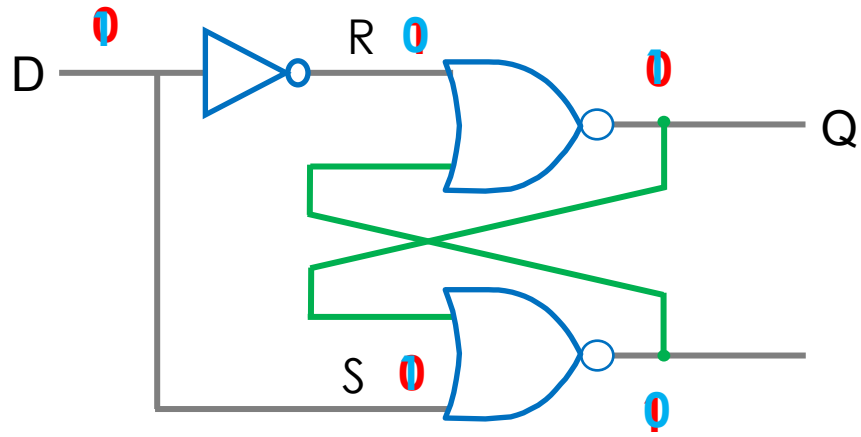
How ?

D Latch

D Latch

- D latch ensures that inputs S and R never equal to 1 at the same time.





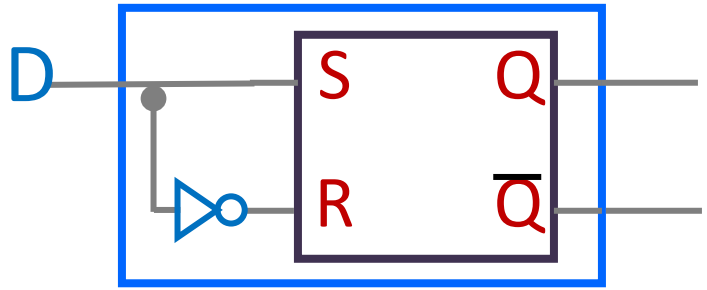
D	Q	
0	0	1
1	1	0

Q+ = Q(t+1) = D

- D Latch can store and change a bit like an SR Latch while avoiding the forbidden state.

D Latch

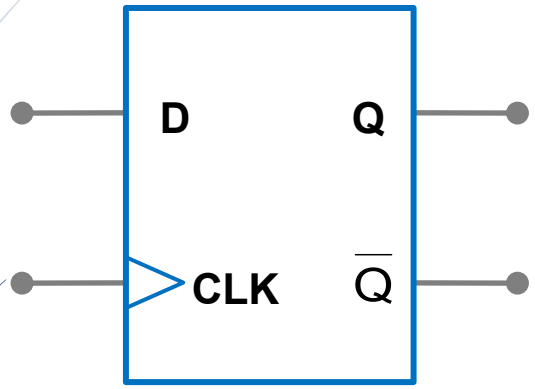
- Data (D) Latch
  - Easier to use than an SR latch
  - No possibility of entering an undefined state

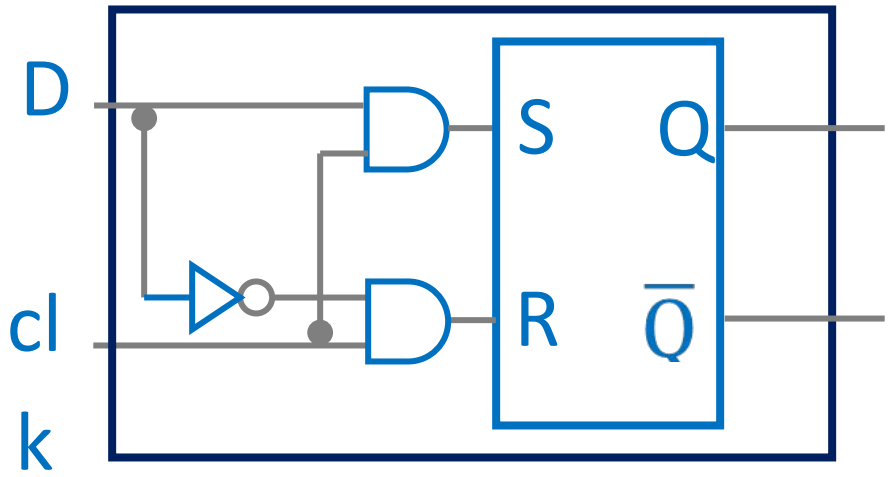


- When D changes, Q changes immediately (…after a delay of 1 NOT gate and 2 NORs gates)

- SR latches are useful in control applications
  - we often think in terms of setting or resetting a **flag** to some condition.
- D latches are useful to store bits of information

D Flip-Flop (D Latch with Clock)





Excitation Table

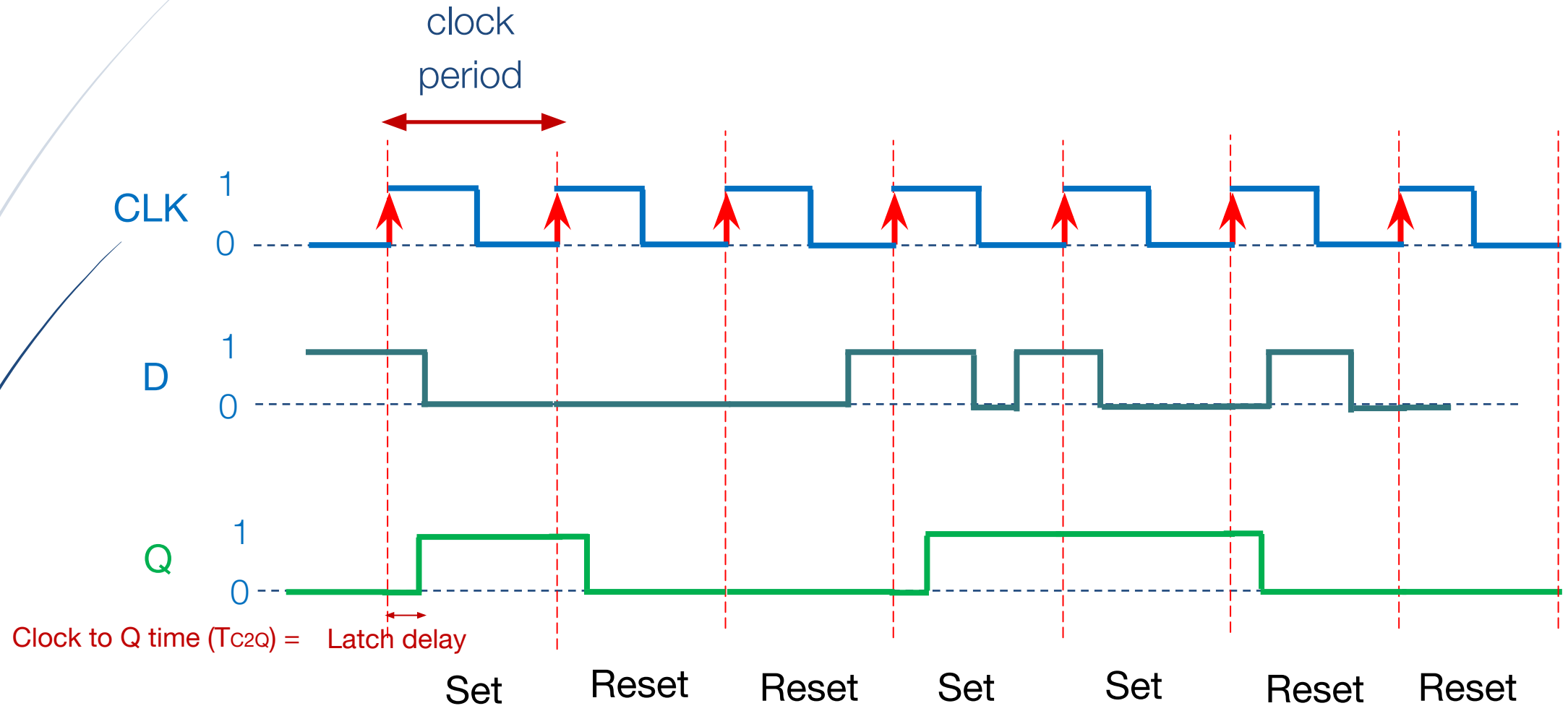
CLK	D	Q	$\overline{Q}$
↑	0	0	1
↑	1	1	0

↑ : Rising Edge of Clock

TruthTable

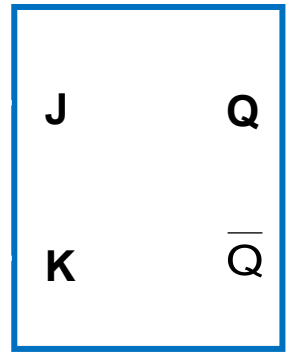
clk	D	Q+	
0	0	Q	Store bit
0	1	Q	
1	0	0	Reset bit
1	1	1	Set bit

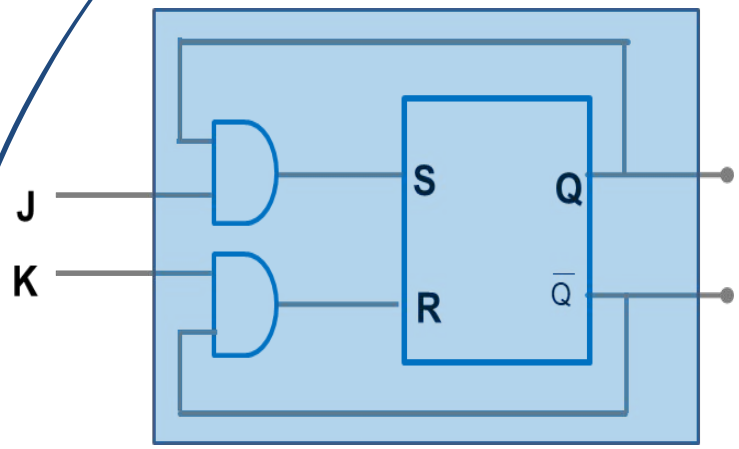
D Flip-Flop: Example Timing Diagram

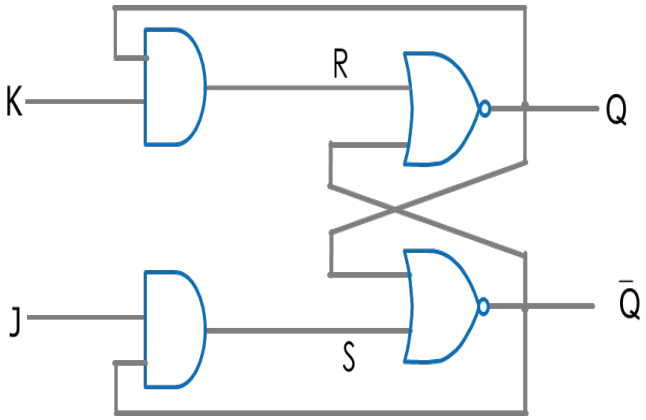


J K Latch

- The SR latch can be modified to provide a stable state when both inputs are 1 : use JK latch.
- The “JK” is called in honor of his inventor Jack Kilby.
- In the JK latch, the inputs J and K behave exactly like inputs S and R, But it provides a stable state when both inputs are 1 (The output is complemented).





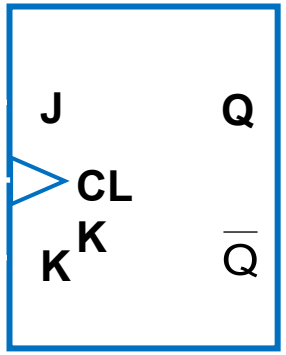


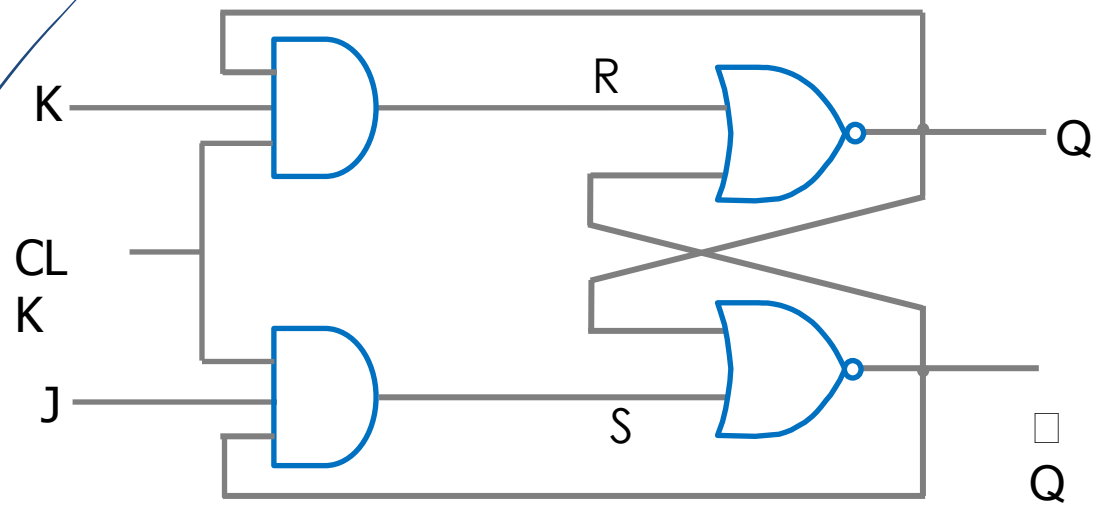
J	K	Q+	
0	0	Q_{prev}	No change (hold /store)
0	1	0	Reset (clear)
1	0	1	Set
1	1	□Q	Toggle

$$Q(t+1) = JQ`(t) + K`Q(t)$$

J K Flip flop

- JK Latch with clock

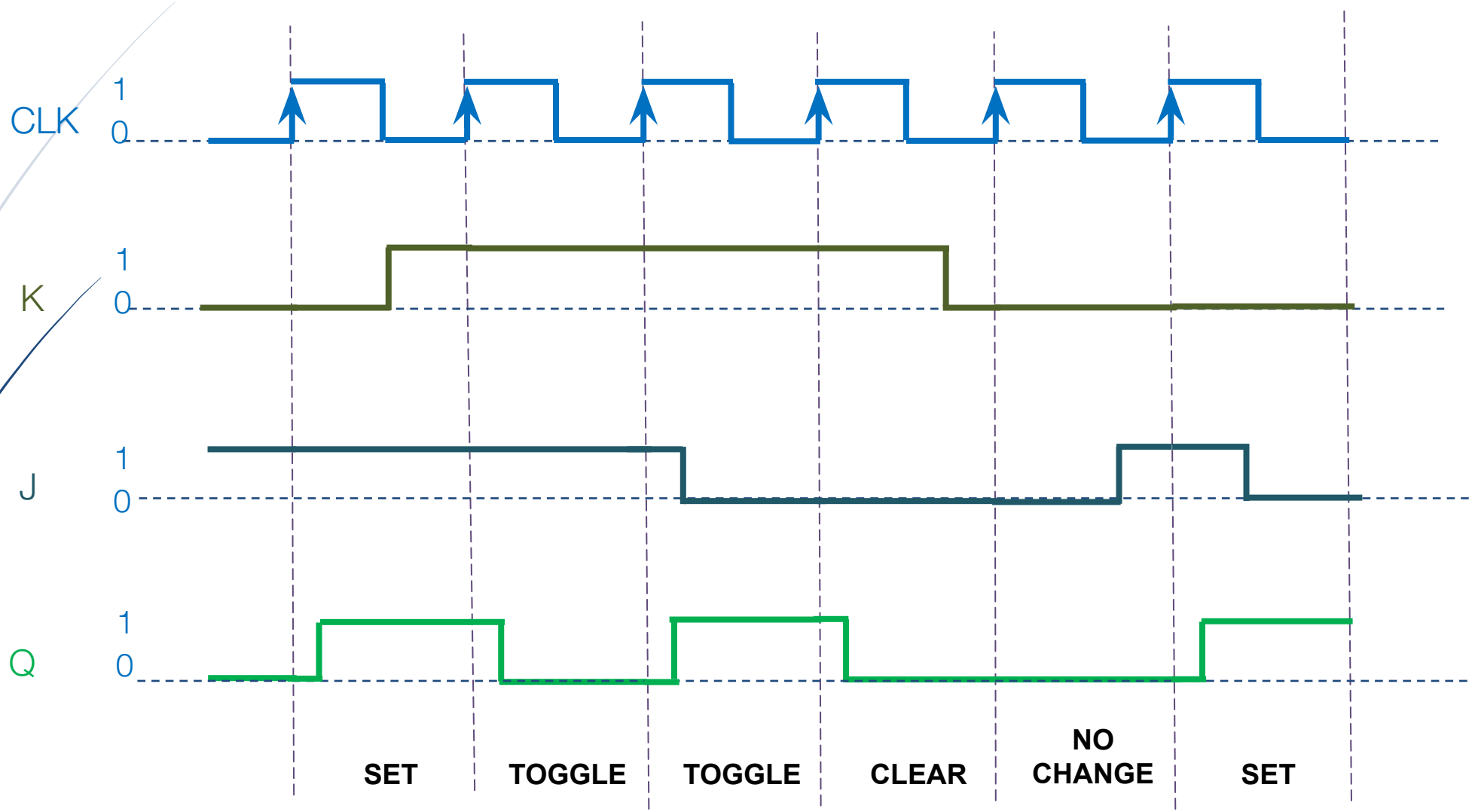




clk	J	K	Q+	
0	x	x	Q	Store
1	0	0	Q	Store
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	Q	Toggle

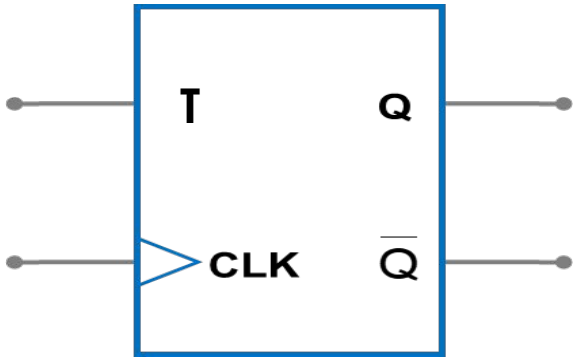
J/K Flip-Flop : Example Timing

J	K	Q+	
0	0	Q_{prev}	No change (hold /store)
0	1	0	Reset (clear)
1	0	1	Set
1	1	$\bar{Q}$	Toggle



T flip-flop

- T (Toggle) Flip-Flop is a complementing synchronous flip-flop.
- It can be designed using :
  - SR (Set-Reset) Flip-Flop
  - D (Data) Flip-Flop
  - JK Flip-Flop

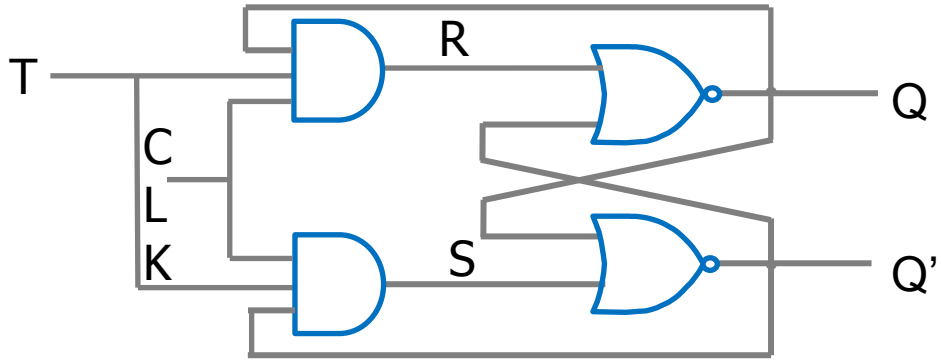


T	Q(t+1)	
0	Q(t)	No change
1	$\overline{Q}$	Complement

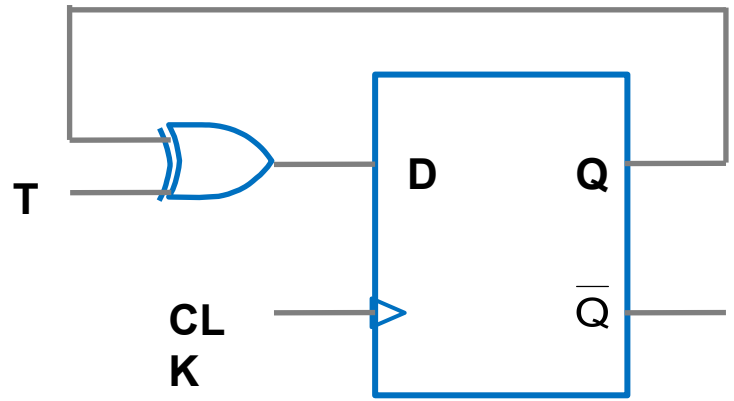
$$Q(t+1) = T \oplus Q = TQ' + T'Q$$

Implementations of T flip flop

- T flip-flop can be implemented using SR flip-flop

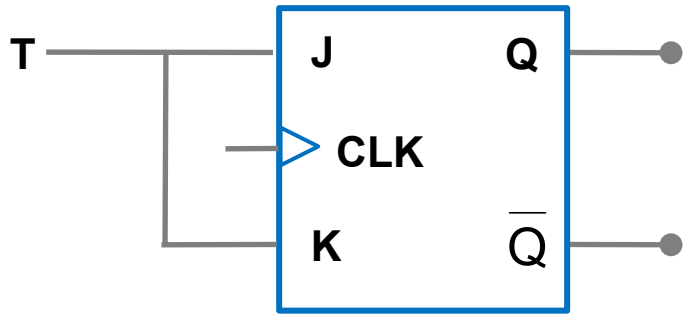


- T flip-flop can be constructed with a D flip-flop and a XOR gate.



Implementations of T flip flop

- The T flip-flop can also be obtained from a JK flip-flop when inputs J and K are tied together.



Characteristic Equations

SR flip-flop Characteristic Equations

$$Q(t + 1) = S + R`Q(t)$$

D flip-flop Characteristic Equations

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

JK flip-flop Characteristic Equations

$$Q(t + 1) = JQ`(t) + K`Q(t)$$

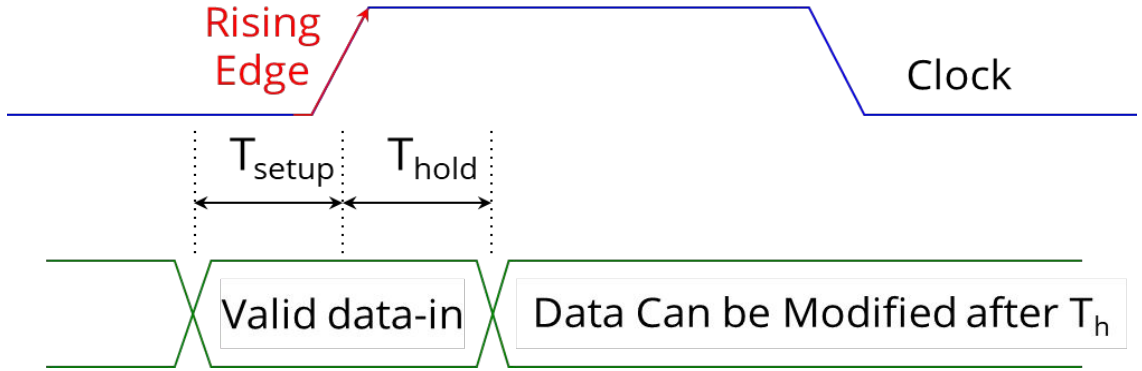
T flip-flop Characteristic Equations

$$Q(t + 1) = T \oplus Q = TQ`(t) + T`Q(t)$$

Sequential circuit timing constraints

- The Flip-Flop operates properly if the imput signal (Ex: D) is stable (that is, not changing) at the time Clk goes from 0 to 1 (if rising edge). But it may lead to unpredictable results if the input signal also changes at this time.

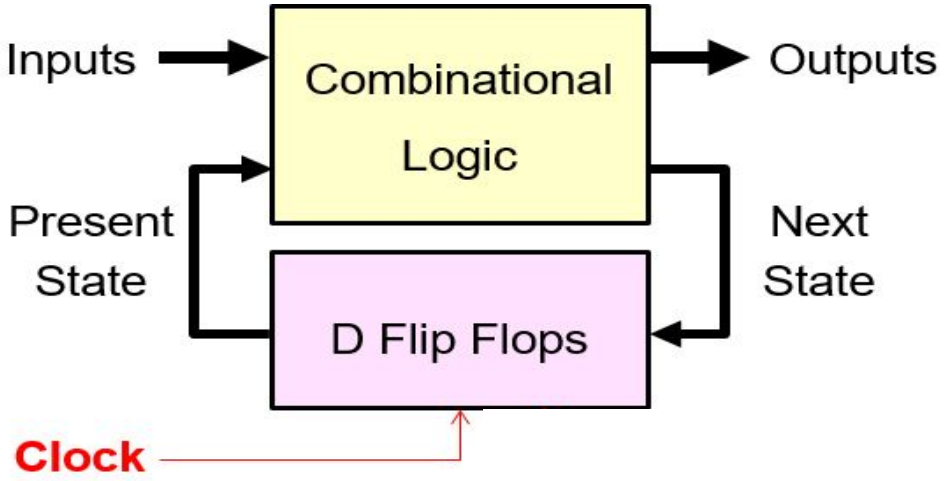
- Constraint 1 : the input signal must be stable between *setup time* and *hold time.*
  - Setup Time (T_{setup}): Time duration for which the data input must be valid and stable before the arrival of the clock edge.
  - Hold Time (T_{hold}): Time duration for which the data input must not be changed after the clock transition occurs.



Sequential circuit timing constraints

Constraint 2 : The following timing constraint is required for correct operation of the sequential circuit :

$$T_{clk} \geq T_{clk2q} + T_{Combinational} + T_{setup}$$



Setup time violation

Timing constraint :

$$T_{clk} \geq T_{clk2q} + T_{Combinational} + T_{setup}$$