# EE 224: INTRODUCTION TO DIGITAL CIRCUITS & COMPUTER DESIGN

## Lecture 4: Sequential Logic – 1 Latches & Flip Flops

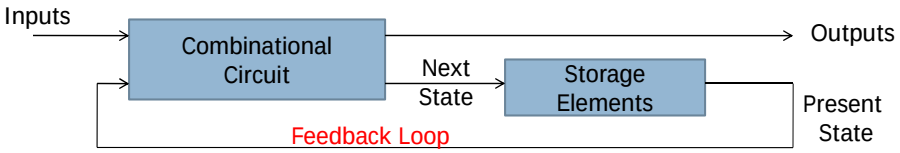
4/19/2010

Avinash Kodi, kodi@ohio.edu

## Introduction to Sequential Logic

2

- In sequential logic circuits the output depends not only on the present input but also on the past sequence of inputs. Thus, these circuits must 'remember' the past history of the inputs to evaluate the present output.

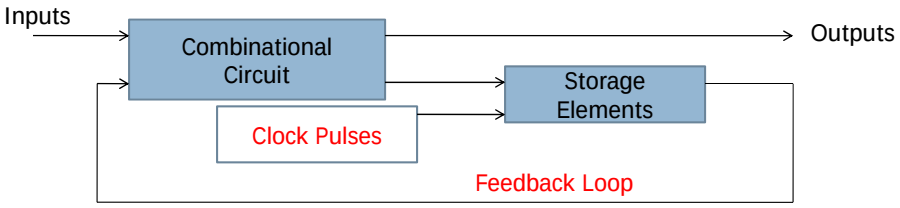



- Latches and flip-flops (FF) are commonly used storage (memory) elements in sequential circuits. Basically, latches and FFs have one or more inputs that can cause the output state to assume one of the two binary levels.

# Basic Concepts

3

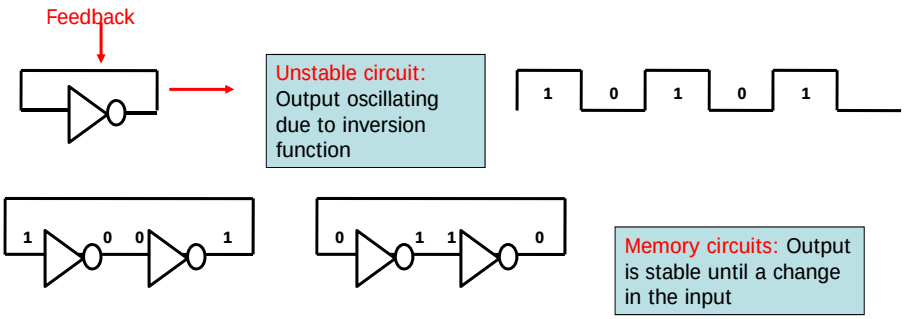
- In sequential logic, time must be divided to specific units/intervals to order the output events or states. According to times at which the inputs are evaluated and output state changes, they may be called
  - Synchronous sequential circuits: all inputs ready and outputs are evaluated at discrete instants of time dictated by the clock signal (pulses).
  - Asynchronous sequential circuit: the inputs arrive at different times and outputs evaluated continuously




# Feedback and Timing

4

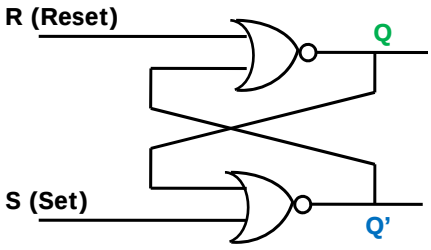
- Feedback: output is fed into input to impact the circuit's (next) output. 'Next' is the following clock pulse in synchronous circuits and as soon as the input is available (stable) in asynchronous circuits
  - Stability: The feedback can create unstable conditions
- Timing: Delays (time need for output evaluation) between input and outputs are important in sequential logic circuits.



# Latches – SR Latch

5

- A storage element can maintain a binary state indefinitely (as long as power is ON) or until inputs change. The most basic storage elements are latches, from which more complex FFs are usually constructed.
- The simplest latch is a tri-stable Set-Reset (SR) circuit built from two NOR gates in cross feedback

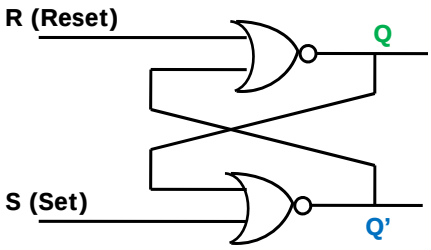



| S | R | Q | Q' | |
|---|---|---|---|---|
| 1 | 0 | 1 | 0 | Set State |
| 0 | 0 | 1 | 0 | |
| 0 | 1 | 0 | 1 | Reset State |
| 0 | 0 | 0 | 1 | |
| 1 | 1 | 0 | 0 | Undefined |

# SR Latch

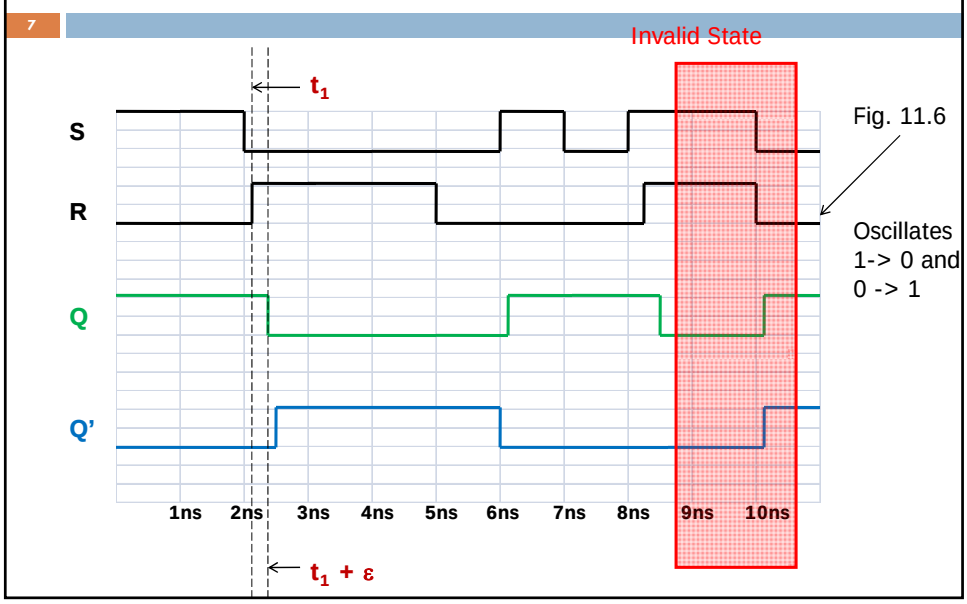
6

- Set=1 and R=0 ⇒SETs the Q (1) output and RESETs Q' (0) (Write)
- Set=0 and R=1 ⇒RESETs the Q output (0) and SETs Q' (1) (Erase)
- Set=0 and R=0 ⇒Retains the states of Q and Q' (No change: Storage)
- Set=1 and R=1 ⇒invalid or forbidden state Q=Q'=0! (No use)

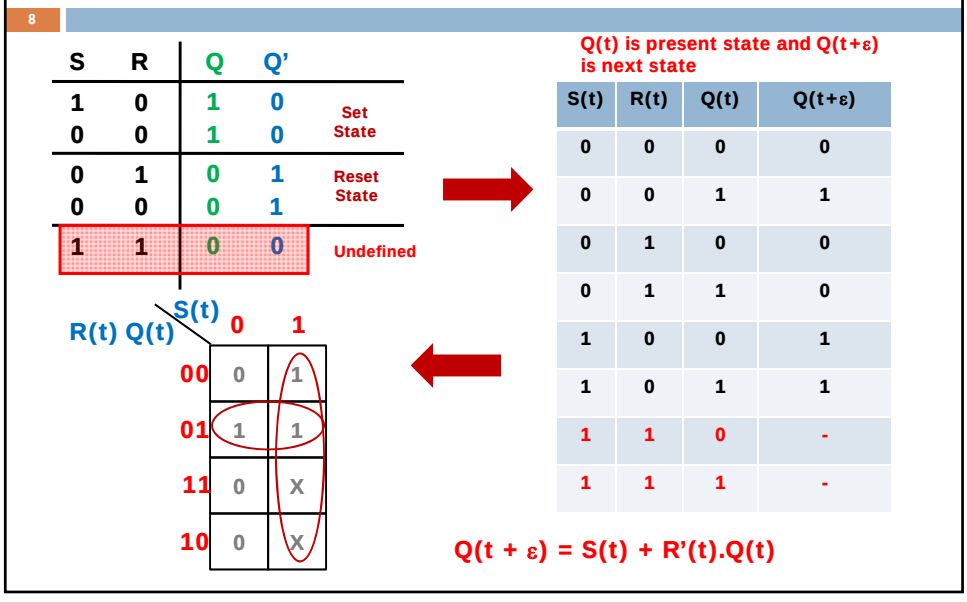



| S | R | Q | Q' | |
|---|---|---|---|---|
| 1 | 0 | 1 | 0 | Set State |
| 0 | 0 | 1 | 0 | |
| 0 | 1 | 0 | 1 | Reset State |
| 0 | 0 | 0 | 1 | |
| 1 | 1 | 0 | 0 | Undefined |

# SR Latch – Timing Diagram



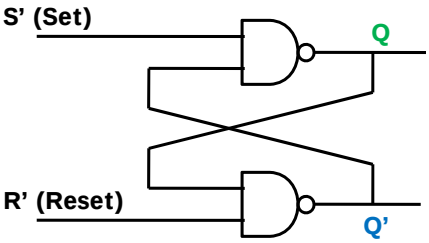

# SR Latch – Truth Table and Characteristic Equation



# S'R' Latch

9

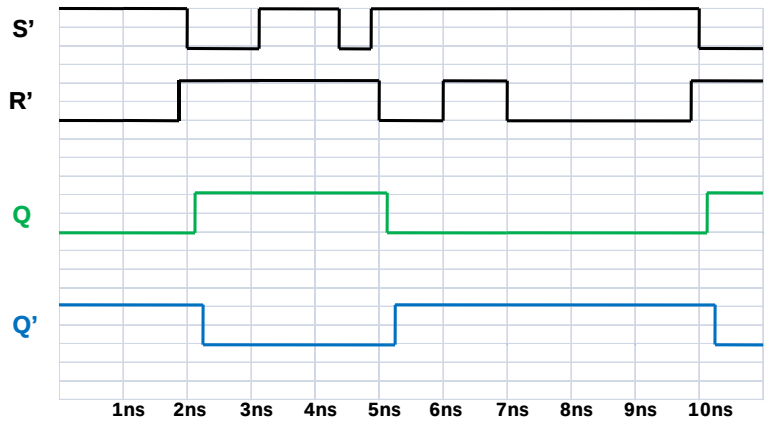
- S'R' Latch is similar to SR, but built using NAND gates
  - Set=0 and R=1 ⇒SETs the Q (1) output and RESETs Q' (0) (Write)
  - Set=1 and R=0 ⇒RESETs the Q output (0) and SETs Q' (1) (Erase)
  - Set=1 and R=1 ⇒Retains the states of Q and Q' (No change: Storage)
  - Set=0 and R=0 ⇒invalid or forbidden state Q=Q'=0! (No use)




| S' | R' | Q | Q' | |
|---|---|---|---|---|
| 0 | 1 | 1 | 0 | Set State |
| 1 | 1 | 1 | 0 | |
| 1 | 0 | 0 | 1 | Reset State |
| 1 | 1 | 0 | 1 | |
| 0 | 0 | 1 | 1 | Undefined |

# S'R' Latch – Timing Diagram

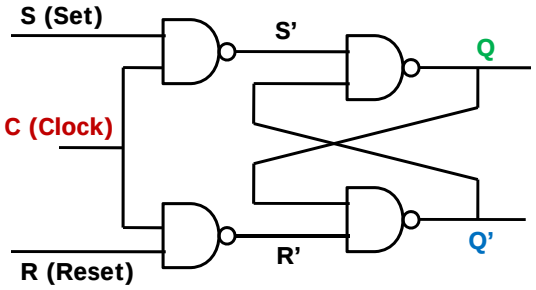
10



# Clocked SR Latch

11

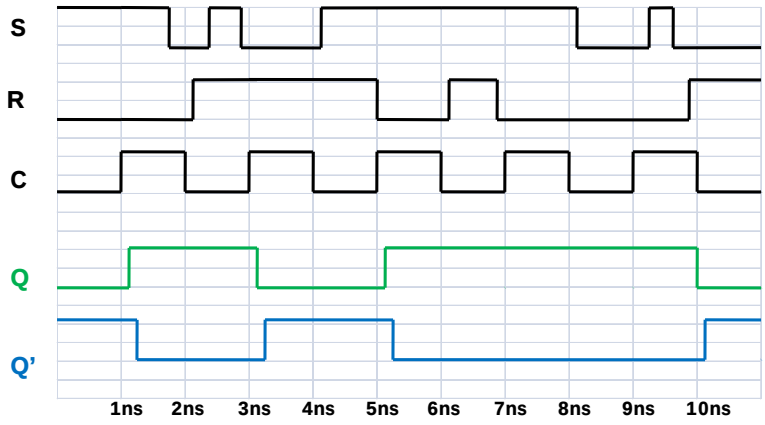
- Apply SR inputs through individual NAND gates with the other inputs coming from a clock input (C)
  - Will evaluate SR inputs and thus Latch only when C=1
    - **When C=0 => (C.S)'=1 => so S'R'=11 (No change)**
    - **When C=1 => (C.S)'=S' => inputs to S'R' latch**
  - Due to clocked inputs S'R' Latch becomes SR Latch
- A way to ensure timing of many latches in the system




| C | S | R | Next State |
|---|---|---|---|
| 0 | X | X | No Change |
| 1 | 0 | 0 | No Change |
| 1 | 0 | 1 | Q = 0, Reset |
| 1 | 1 | 0 | Q = 1, Set |
| 1 | 1 | 1 | Undefined |

# Clocked SR Latch – Timing Diagram

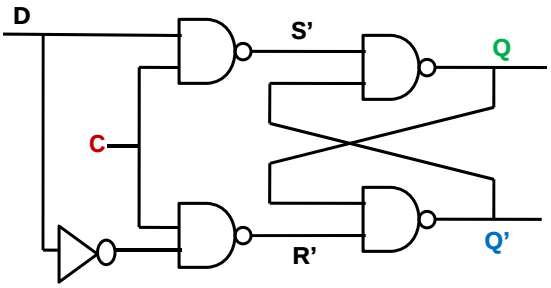
12



# Gated D Latch

13

- One way to eliminate the undesirable state in SR latch is to ensure that S and R are never equal to 1 at the same time



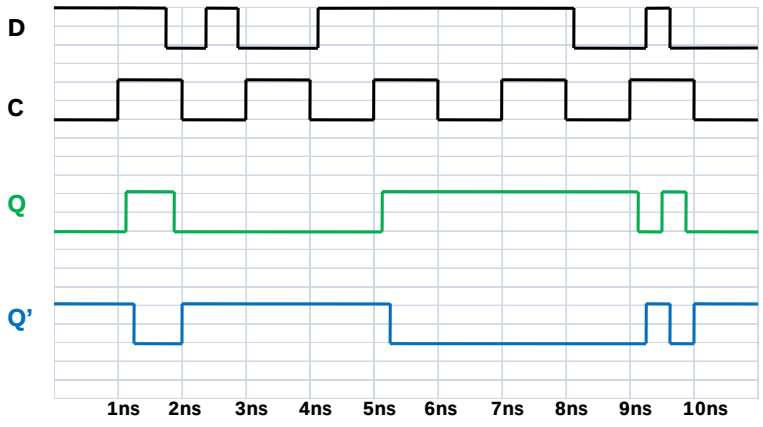

| C | D | Next state ($Q^+$) |
|---|---|---|
| 0 | X | No Change |
| 1 | 0 | Q = 0, Reset |
| 1 | 1 | Q = 1, Set |

| C | D | Q | $Q^+$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

$Q^+ = C'Q + CD$
(Characteristic Equation

# Gated D Latch – Timing Diagram

14



## Flip-Flops

15

- Latches suffer from transparency problem as well as serious timing issues
  - If the clock has a relatively long pulse, the inputs (from other latches) may change more than once causing the output to change
- The solution is to use **Flip-Flops**: These are latches augmented with additional circuits to switch by a momentary change in the input control

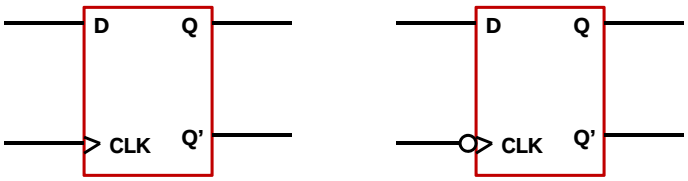
## Flip Flops

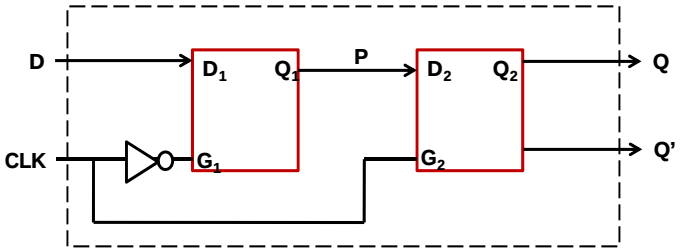
16

- FFs have a clock input and the output changes only in response to a rising or falling edge of the clock
- Edge-Triggered: Triggers only during a signal transition from 0 to 1 (or from 1 to 0) on the clock and disabled at all other times (even when C = 1)
- A 2-stage master (recorder)-slave (reporter) FF with edge triggering

# Edge-Triggered D-Flip Flop

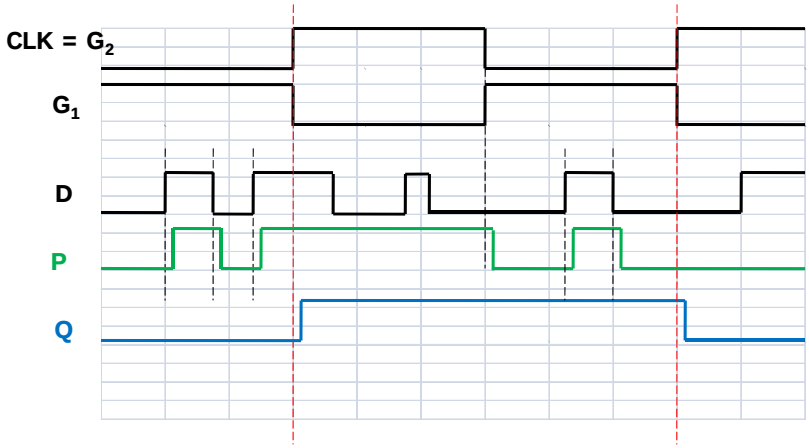
17



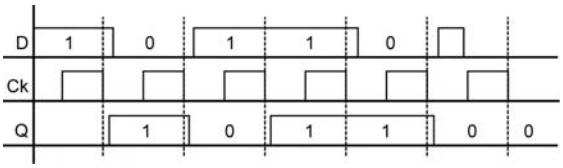

# Rising Edge-Triggered D-Flip Flop

18



# Falling Edge Triggered D-Flip Flop

19

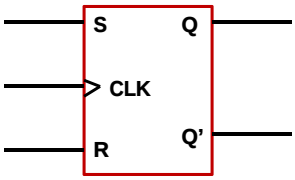



Note that Output (Q) follows input (D) after each falling edge of the Clock (Ck) in the timing diagram

Important parameters: propagation delay, setup time, hold time

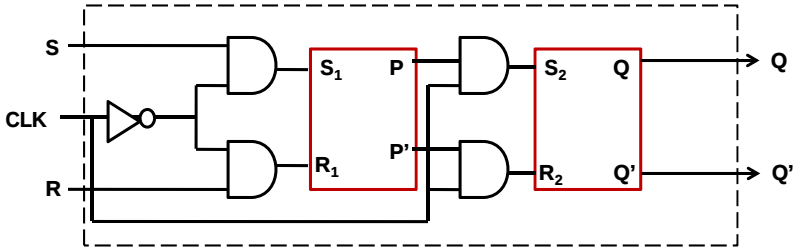
# SR Flip Flop

20



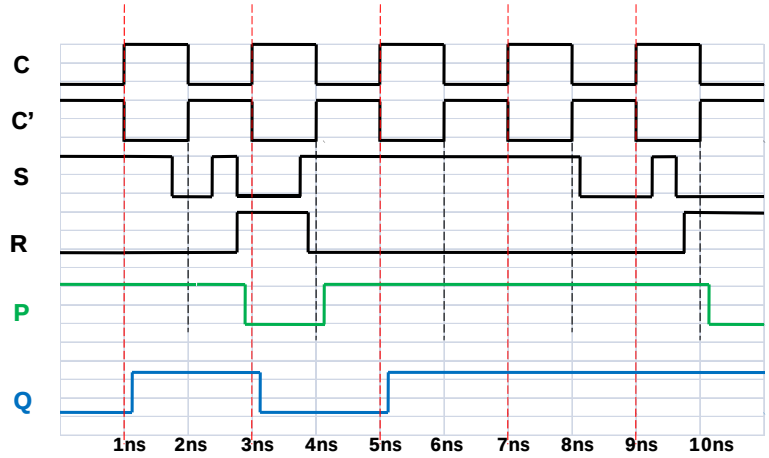

Rising Edge Trigger

| S | R | Next state ($Q^+$) |
|---|---|---|
| 0 | 0 | No Change |
| 1 | 0 | Q = 1, Set (after active CLK edge) |
| 0 | 1 | Q = 0, Reset (after active CLK edge) |
| 1 | 1 | Not allowed |



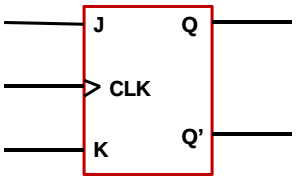
# Rising Edge Triggered SR Flip Flop

21




# JK Flip-Flop (J = S, K = R)

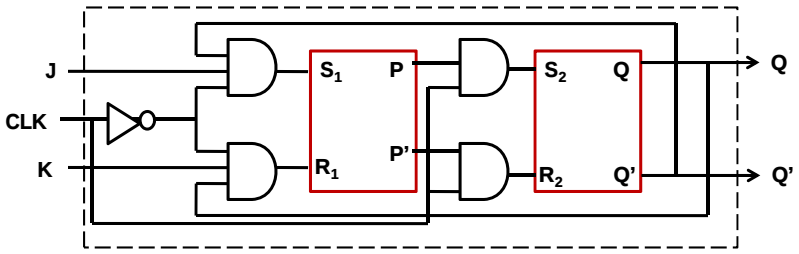
22




Rising Edge Trigger

| J | K | Next state ($Q^+$) |
|---|---|---|
| 0 | 0 | No Change |
| 1 | 0 | Q = 1, Set (after active CLK edge) |
| 0 | 1 | Q = 0, Reset (after active CLK edge) |
| 1 | 1 | Q' |

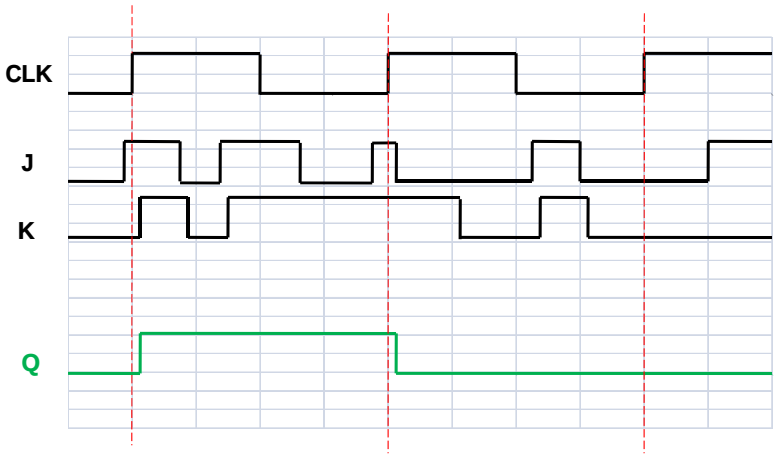
$Q^+ = JQ' + K'Q$
(Characteristic Equation)



# Rising Edge-Triggered JK FF

23

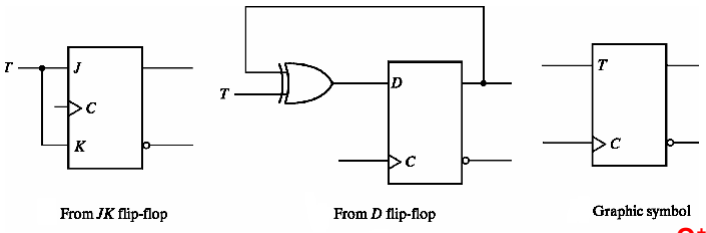
- In JK FF, J (S) = 1 and K (R) = 1 is allowed and causes the output to complement its value




# T Flip Flop

24

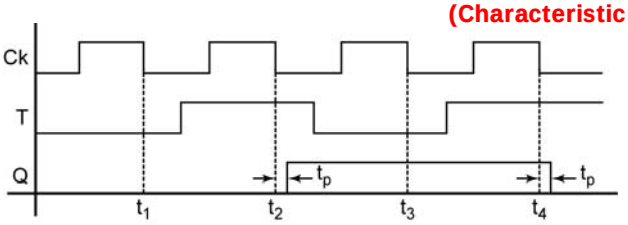
- T Flip Flop, called as Toggle FF is frequently used in building counters




| TQ | $Q^+$ |
|---|---|
| 0 0 | 0 |
| 0 1 | 1 |
| 1 0 | 1 |
| 1 1 | 0 |

$Q^+ = T \oplus Q$
(Characteristic Equation)

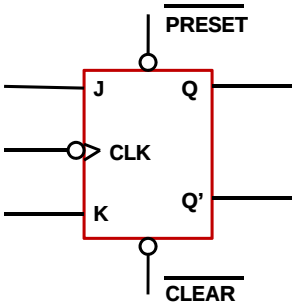
Falling edge triggered



# Asynchronous JK FF

25

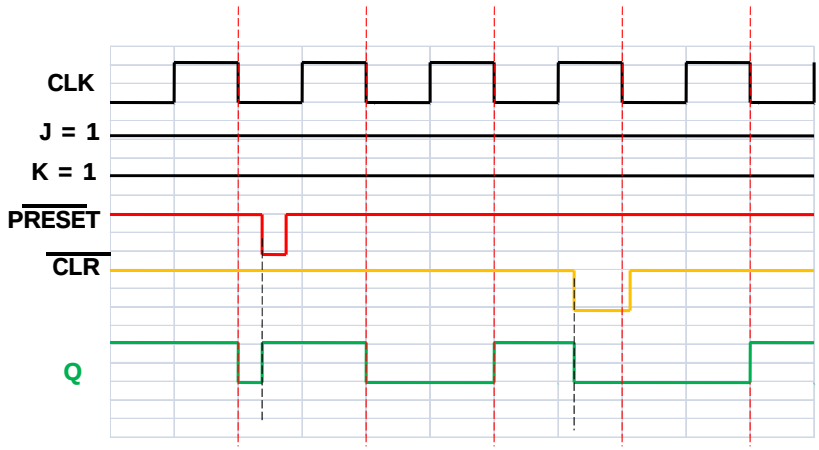
- J, K are synchronous inputs
  - Effect on the output synchronous with CLK input
- In some JK FF, there maybe asynchronous inputs which operate independent of inputs and clock
  - Set the FF to 1/0 state at any time




| $\overline{PRESET}$ | $\overline{CLEAR}$ | FF Response |
|---|---|---|
| 1 | 1 | Clocked Operation |
| 0 | 1 | Q = 1, (regardless of CLK) |
| 1 | 0 | Q = 0, (regardless of CLK) |
| 0 | 0 | Not Used |

# Asynchronous JK FF

26



# FF Summary

27

**Flip-Flop Characteristic Table**

**JK Flip-Flop**

| J | K | $Q^+$ | Operation |
|---|---|---|---|
| 0 | 0 | Q | No Change |
| 0 | 1 | 0 | Reset |
| 1 | 0 | 1 | Set |
| 1 | 1 | Q’ | Complement |

**SR Flip-Flop**

| S | R | $Q^+$ | Operation |
|---|---|---|---|
| 0 | 0 | Q | No Change |
| 0 | 1 | 0 | Reset |
| 1 | 0 | 1 | Set |
| 1 | 1 | ? | Undefined |

**D Flip-Flop**

| D | $Q^+$ | Operation |
|---|---|---|
| 0 | 0 | Reset |
| 1 | 1 | Set |

**T Flip-Flop**

| T | $Q^+$ | Operation |
|---|---|---|
| 0 | Q | No Change |
| 1 | Q’ | Complement |