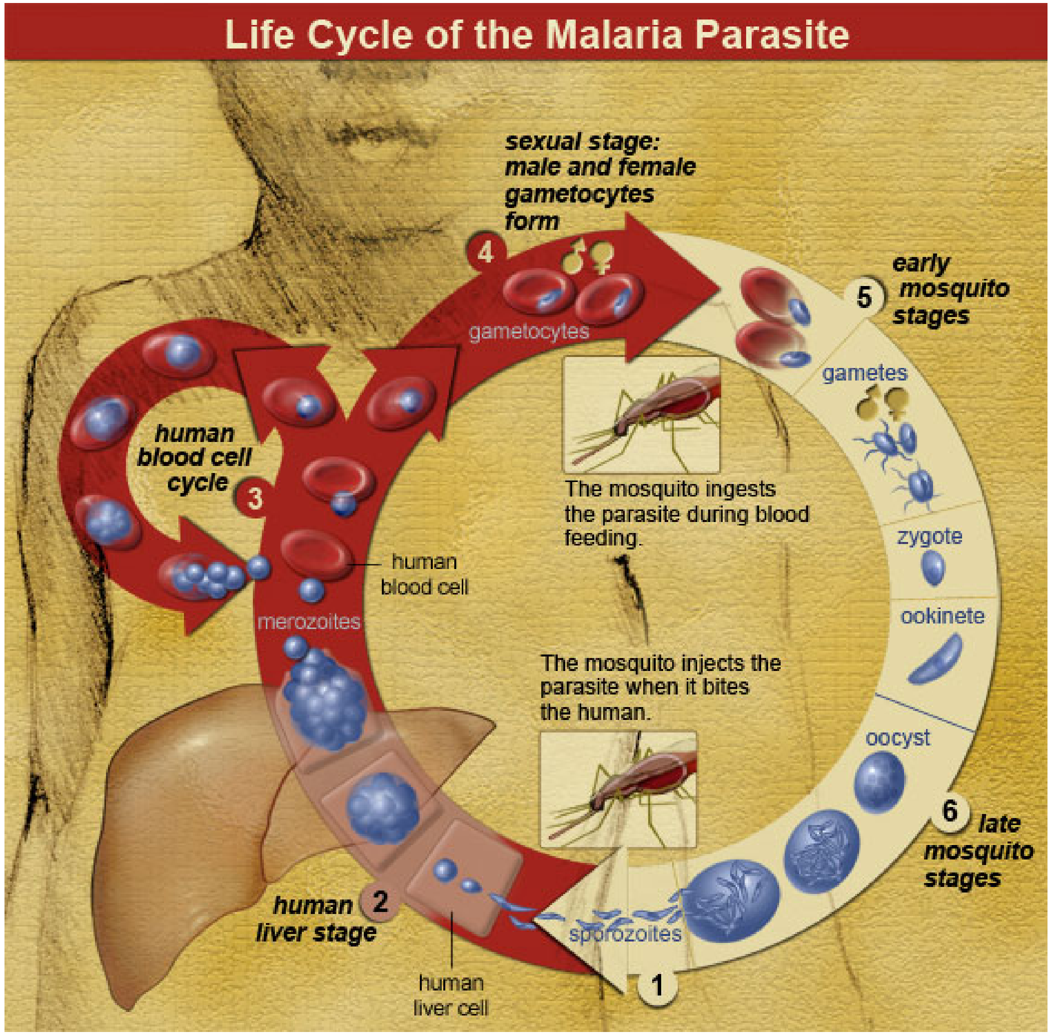
Switching Circuits & Logic Design

Jie-Hong Roland Jiang
江介宏

Department of Electrical Engineering
National Taiwan University

Fall 2012

§11 Latches and Flip-Flops



http://www3.niaid.nih.gov/topics/Malaria/lifecycle.htm

Outline

- Introduction
- Set-reset latch
- Gated D latch
- Edge-triggered D flip-flop
- S-R flip-flop
- J-K flip-flop
- T flip-flop
- Flip-flops with additional inputs
- Summary

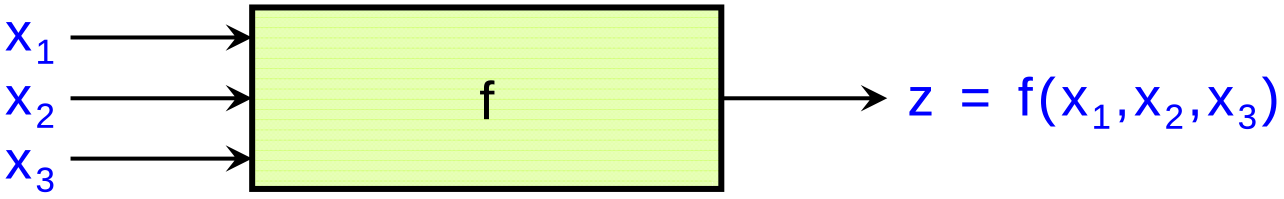
Introduction

- Combinational circuits
  - Output is a *function* depending on the present input, but not past inputs
    - Given an arbitrary input, a combinational circuit produces only one possible output (after certain delay)
  - Not necessarily **acyclic** (without feedback)

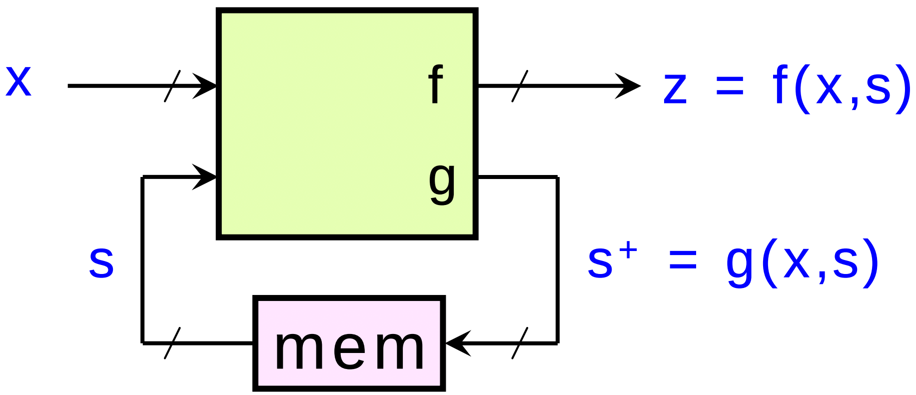
- Sequential circuits
  - Output is a *function* depending on the past sequence of inputs
  - Must be **cyclic** (with feedback)
    - **Synchronous** sequential circuits
      - With memory devices (registers, latches)
    - **Asynchronous** sequential circuits
      - Without memory devices

Introduction

- Combinational circuits (without memory)



- Sequential circuits (with memory)



time index

$z^0 = f(x^0, s^0)$
$z^1 = f(x^1, g(x^0, s^0))$
$z^2 = f(x^2, g(x^1, g(x^0, s^0)))$
$\vdots$

Introduction

- To construct a system (e.g., circuit, neural network, etc.) that "remembers" something about the past history of the inputs
  - Need feedback!
    - Closed loops formed in a circuit connection

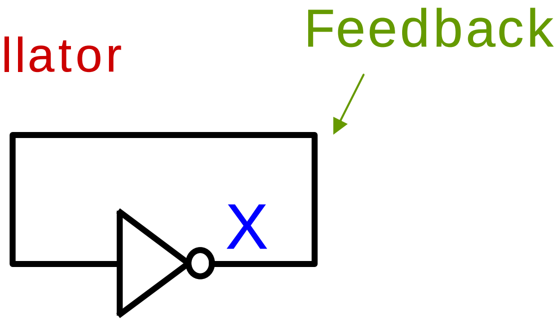
Introduction
Memory devices

- Memory devices
  - **Latches** and **flip-flops** can assume one of two stable output states, and have one or more inputs that can cause the output state to change
    - Latch
      - Have no clock input
    - Flip-flip
      - Change output state in response to a clock input, but not a data input

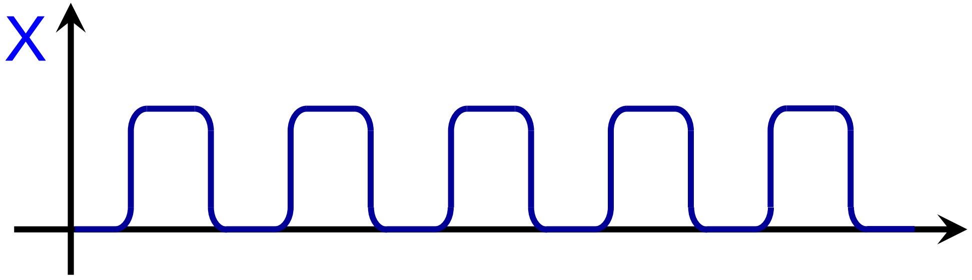
Introduction
Feedback

- Unstable

Oscillator



Inverter with feedback



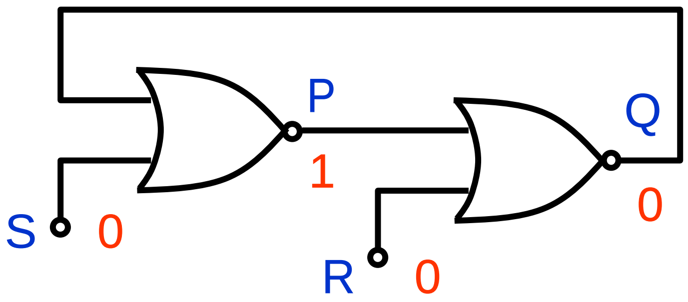
Oscillation at inverter output

- Stable

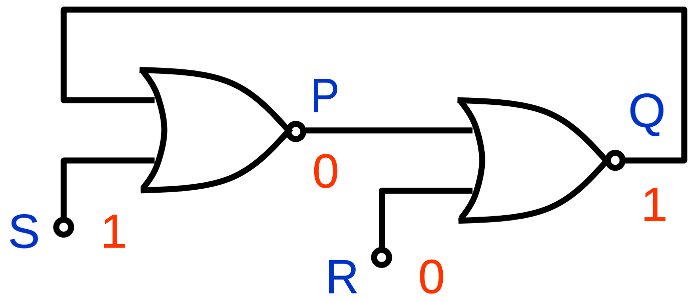
Memory (1-bit)

Set-Reset Latch

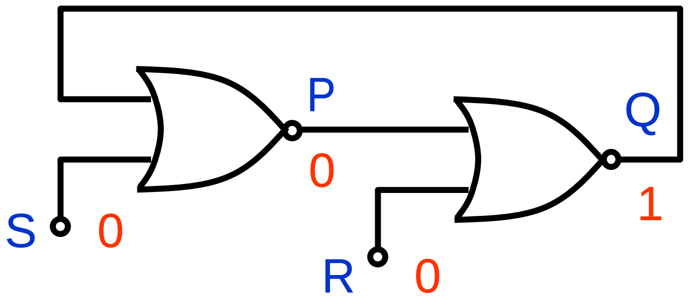
S-R latch



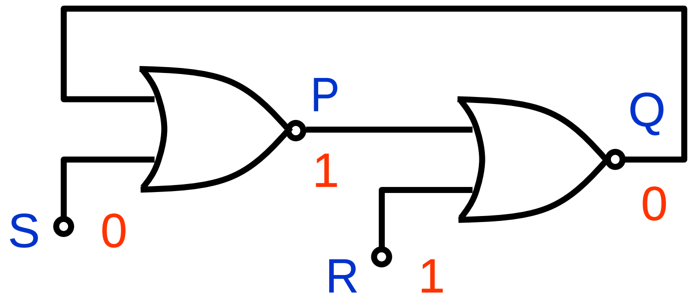
(a) Stable: Q=0



(b) Set: S: 0→1 ⇒ Q: 0→1



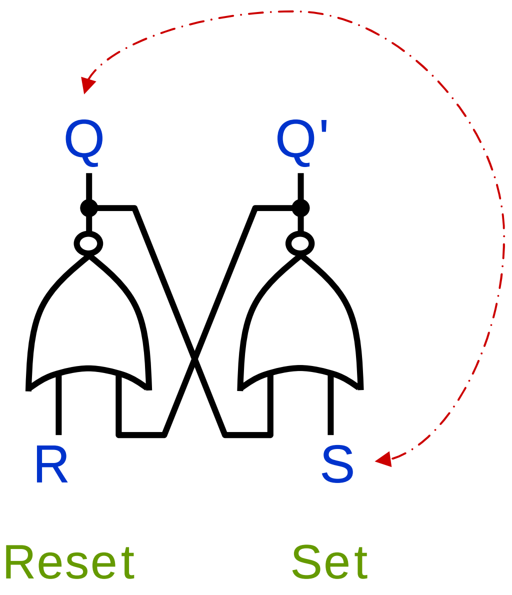
(a) Stable: Q=1



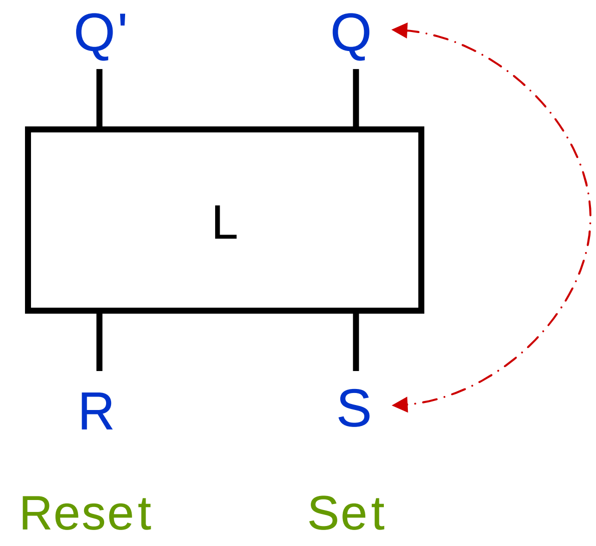
(b) Reset: R: 0→1 ⇒ Q: 1→0

Set-Reset Latch

- Cross-coupled form



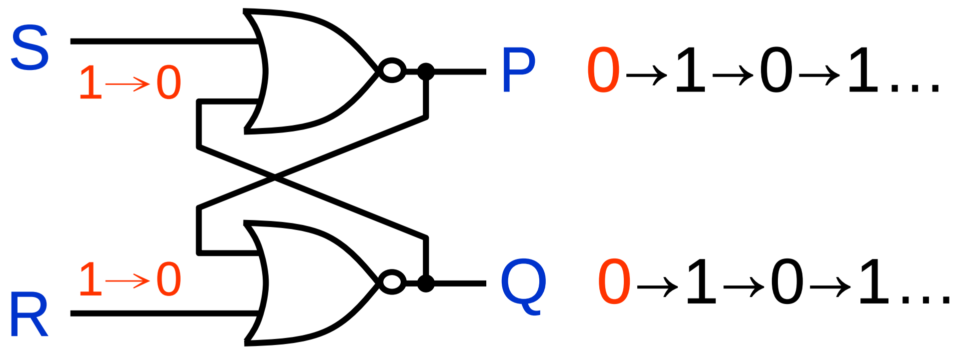
- S-R latch symbol



Q directly above S (different from the cross-coupled form)

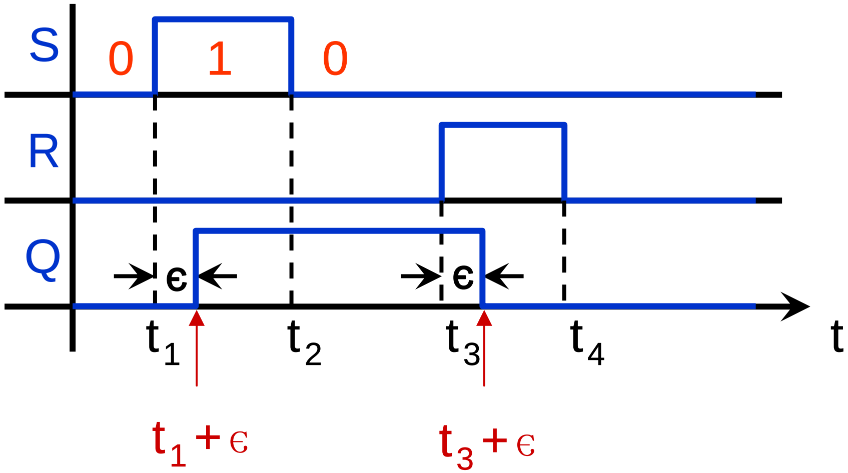
Set-Reset Latch

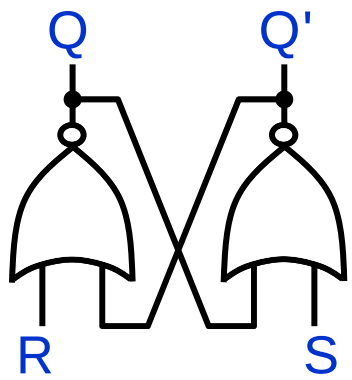
- Improper S-R latch operation
  - When S = R = 1, the circuit is unstable
  - Disallow S = R = 1 for S-R latch



Set-Reset Latch

Timing diagram





ε: two NOR-gate delay

The duration of the S (or R) input pulse must normally be no less than ε in order for a change in the state of Q to occur

Set-Reset Latch Operation

- **Next-state equation** (or **characteristic equation**):

  $Q^+ = S + R'Q$ ($SR = 0$, i.e., $S = R = 1$ disallowed)

  - **Present** (or **current**) **state** Q
    - The state of the Q output of the latch or flip-flop at the time the input signals are applied (or changed)
  - **Next state** Q^+
    - The state of the Q output after the latch or flip-flop has reacted to these input signals

S(t)	R(t)	Q(t)	Q(t+ε)	
0	0	0	0	hold
0	0	1	1	
0	1	0	0	reset
0	1	1	0	
1	0	0	1	set
1	0	1	1	
1	1	0	-	prohibited
1	1	1	-	

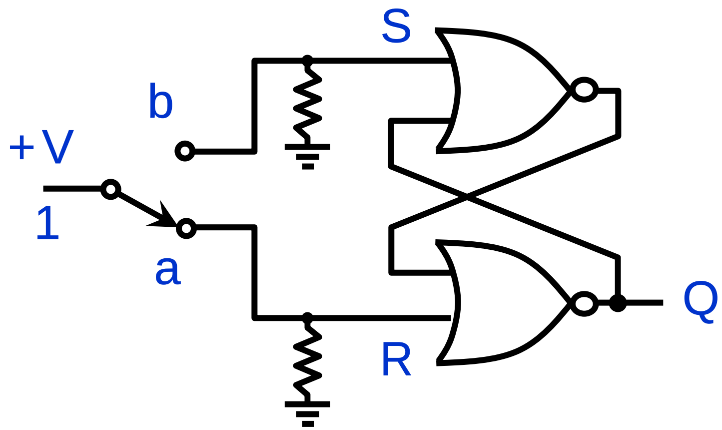
R(t) Q(t) \ S(t)	0	1
00	0	1
01	1	1
11	0	×
10	0	×

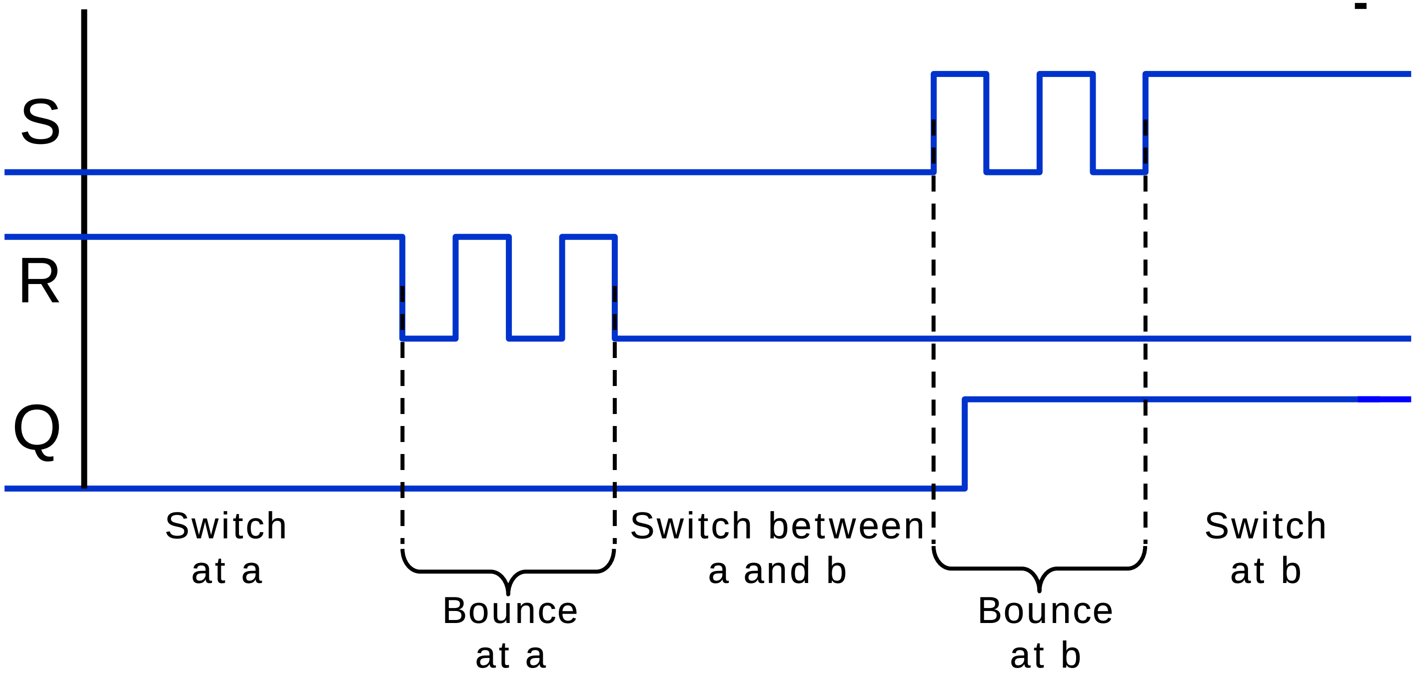
$Q(t+\epsilon) = S(t) + R'(t)Q(t)$

Set-Reset Latch Application

- Switch debouncing

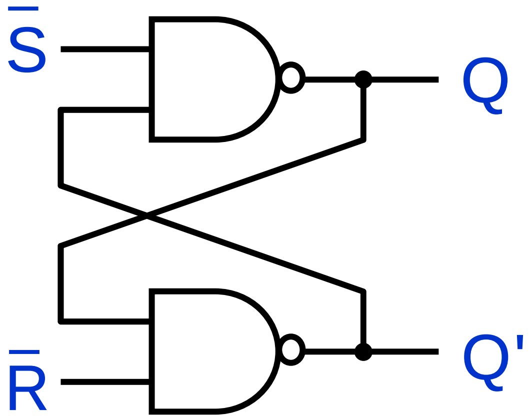
  Note: only work for a double throw switch, switching between two contacts (but not for a single throw switch) why?

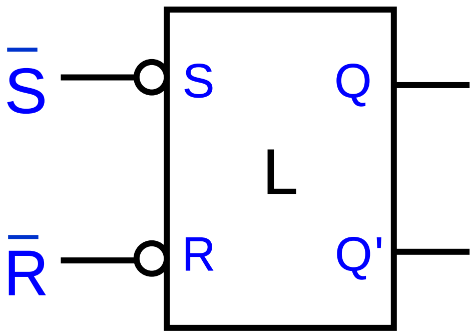




Set-Reset Latch
Alternative Implementation

- $\overline{S}$-$\overline{R}$ latch
  - S-R latch using NAND gates



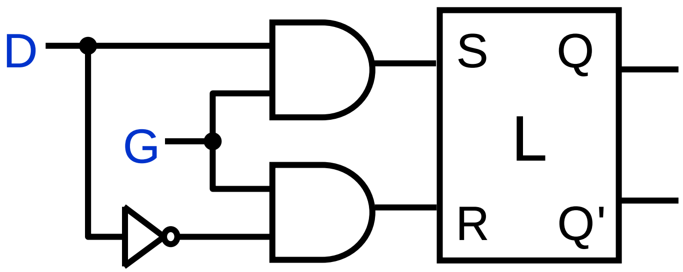


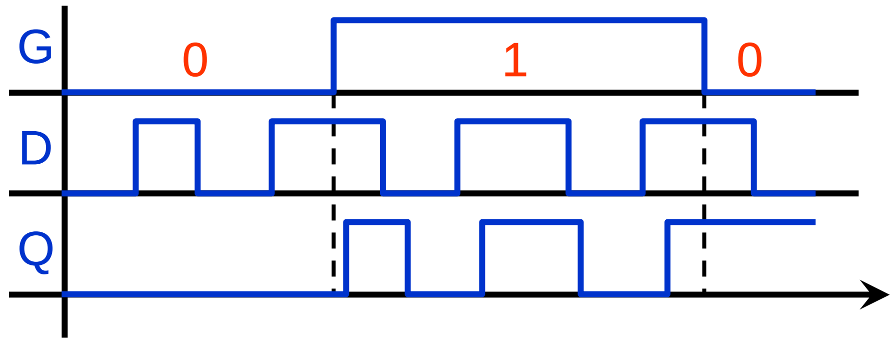
$\overline{S}$	$\overline{R}$	Q	Q^+	
1	1	0	0	hold
1	1	1	1	
1	0	0	0	reset
1	0	1	0	
0	1	0	1	set
0	1	1	1	
0	0	0	-	prohibited
0	0	1	-	

Inputs S and R are **active low**

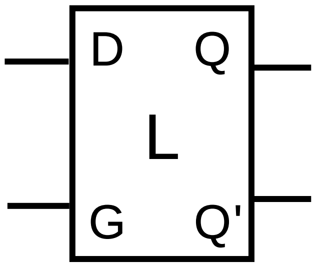
Gated D Latch

Gated D latch





Symbol



Truth table

G D Q	Q^+	
0 0 0	0	hold ($Q^+ = Q$)
0 0 1	1	
0 1 0	0	
0 1 1	1	
1 0 0	0	transparent ($Q^+ = D$)
1 0 1	0	
1 1 0	1	
1 1 1	1	

Q \ GD	00	01	11	10
0	0	0	1	0
1	1	1	1	0

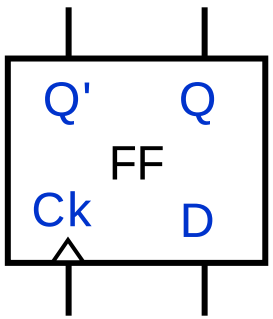
$$Q^+ = G'Q + GD$$

Edge-Triggered D Flip-Flop

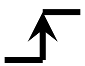
- Unlike D latch, D flip-flip output changes only in response to the clock, not to a change in D

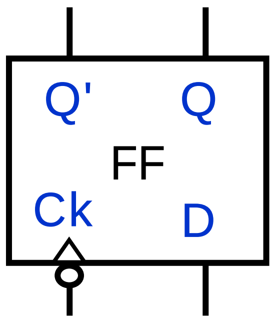
rising (or **positive**) **edge** triggered (0-to-1 transition on clock)

falling (or **negative**) **edge** triggered (1-to-0 transition on clock)

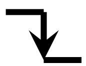


Rising-edge trigger





Falling-edge trigger



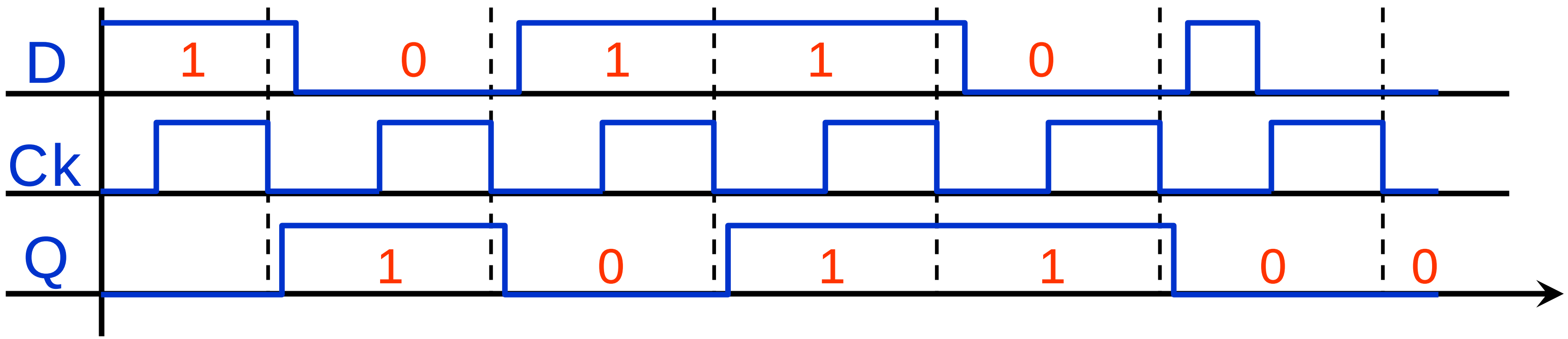
D	Q	Q^+
0	0	0
0	1	0
1	0	1
1	1	1

Truth table

$$Q^+ = D$$

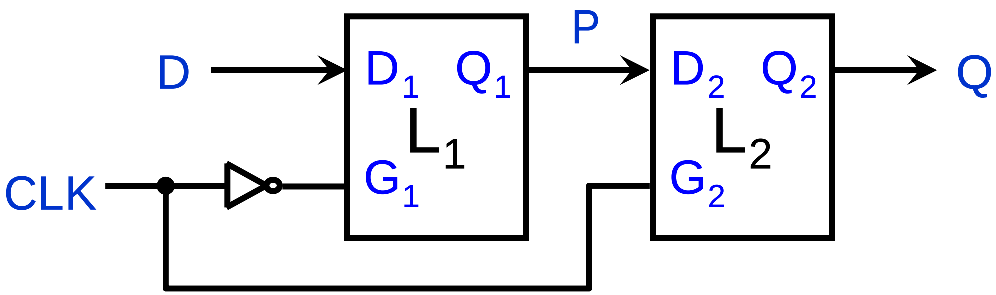
Edge-Triggered D Flip-Flop

- Timing diagram
  - (falling-edge trigger)



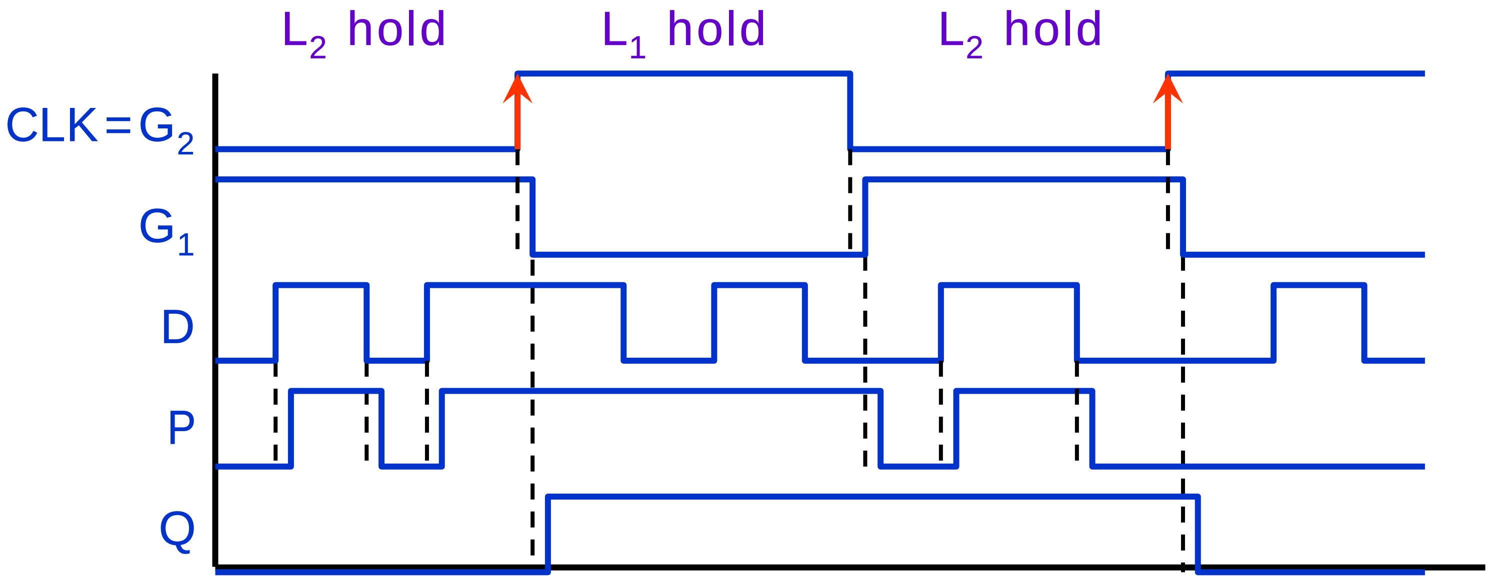
Edge-Triggered D Flip-Flop Implementation

- D flip-flop (rising-edge trigger)
  - Composed of two gated D latches



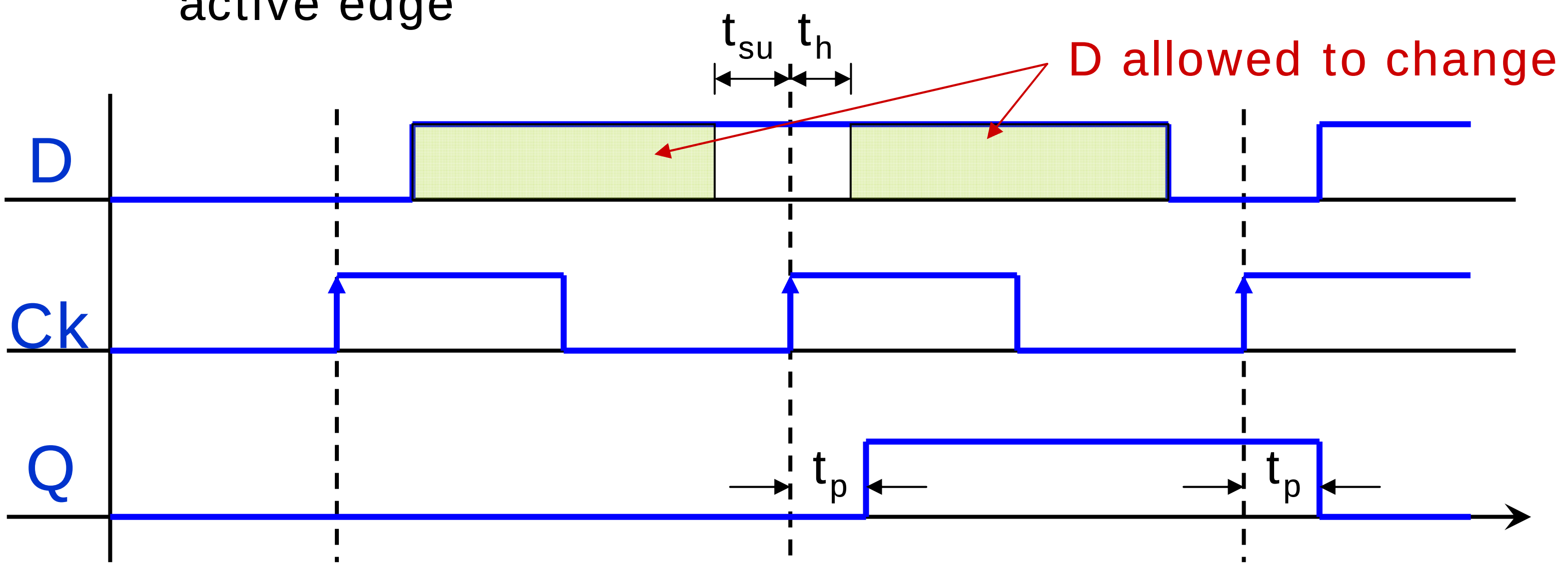
If L_1 starts following D before L_2 takes on P, the FF will not function properly

Time analysis



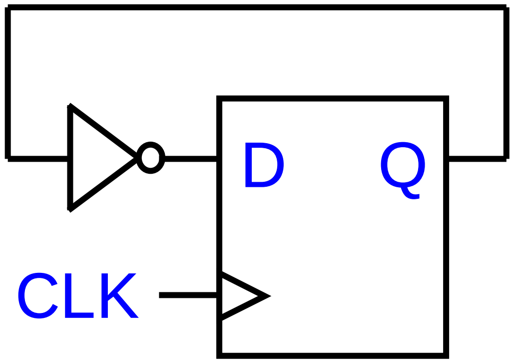
Edge-Triggered D Flip-Flop
Setup Time and Hold Time

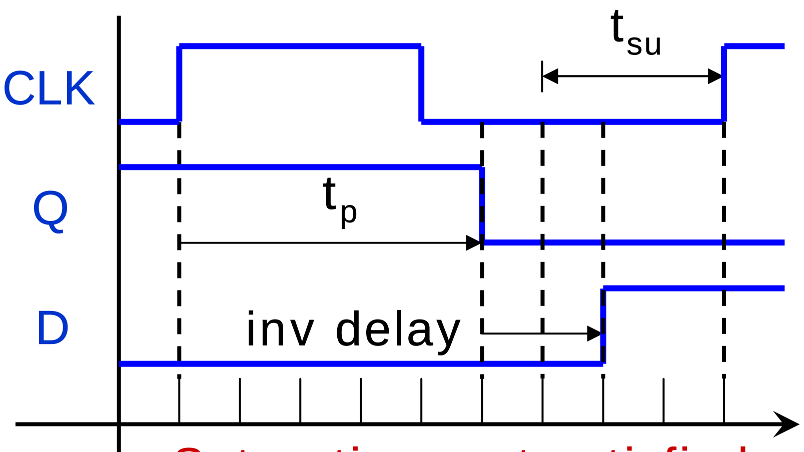
- **Propagation delay**: t_p
  - The time between the active edge of the clock and the resulting change in the output
- **Setup time**: t_{su}
  - The amount of time D must be stable before the active edge
- **Hold time**: t_h
  - The amount of time D must hold the same value after the active edge



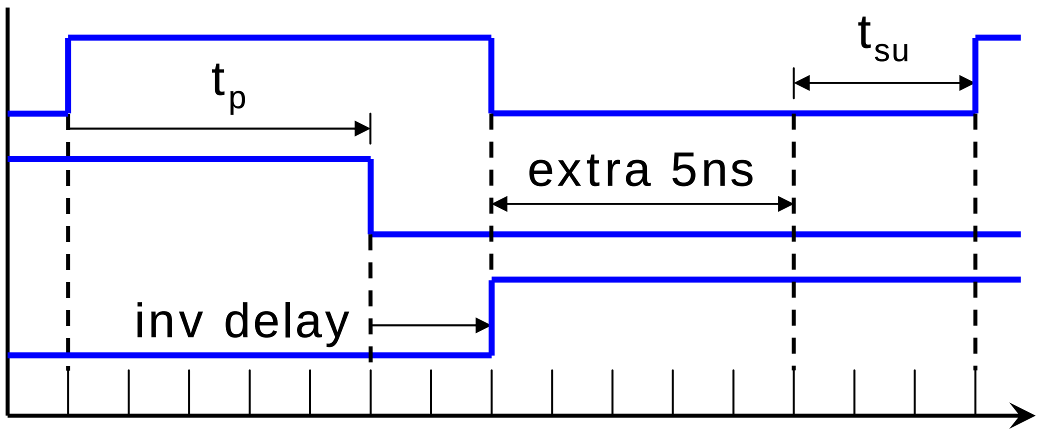
Edge-Triggered D Flip-Flop
Determine Minimum Clock Period

- Simple flip-flop circuit example
  (t_p 5ns, t_{su} 3ns, inverter delay 2ns)

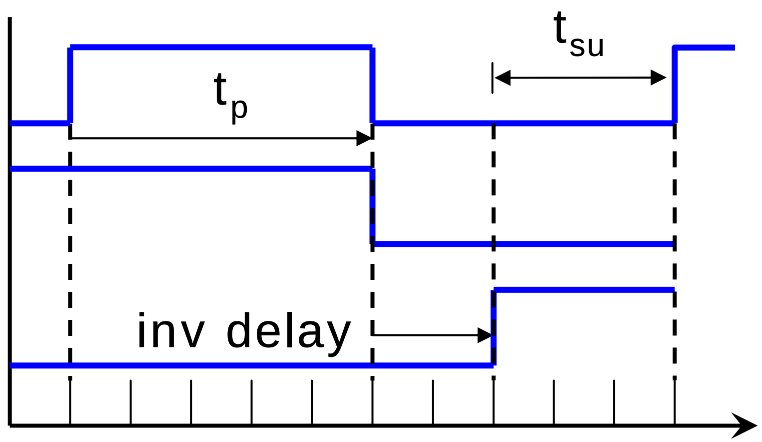




Setup time not satisfied



Setup time satisfied

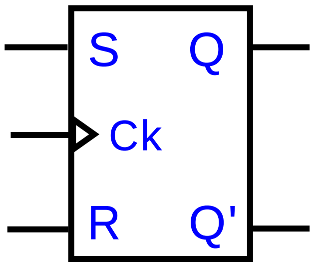


Minimum clock period

S-R Flip-Flop

- Similar to S-R latch but with clock input
  - Same truth table and characteristic equation
  - Interpretation of Q^+ is different
    - Latch: Q^+ is the value of Q after the propagation delay through the latch
    - FF: Q^+ is the value that Q assumes after the active clock edge

S-R flip-flop



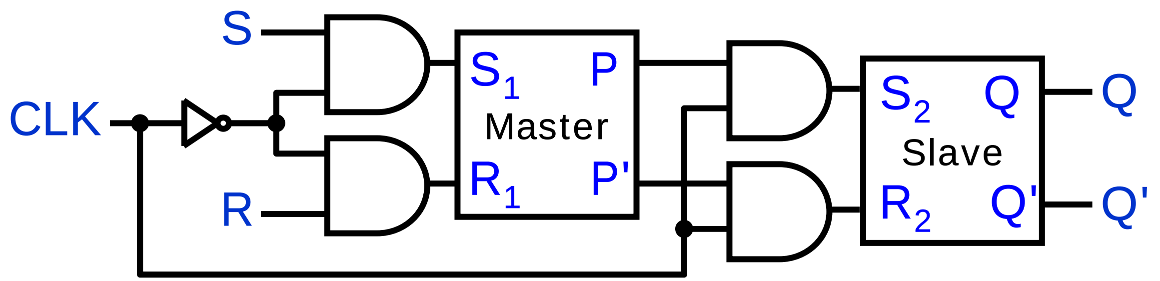
Q changes at clock edges

Operation summary:

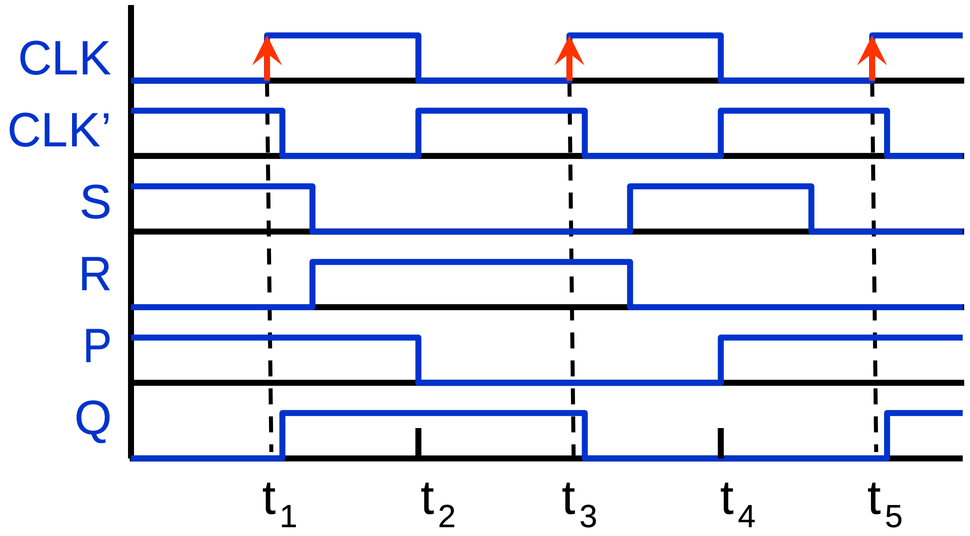
S=R=0	no state change
S=1,R=0	set Q to 1 (after active Ck edge)
S=0,R=1	reset Q to 0 (after active Ck edge)
S=R=1	not allowed

S-R Flip-Flop Implementation

- S-R flip-flop (master-slave flip-flop)
  - Composed of two S-R latches
  - Only allow the S and R inputs to change while CLK is high



Time analysis



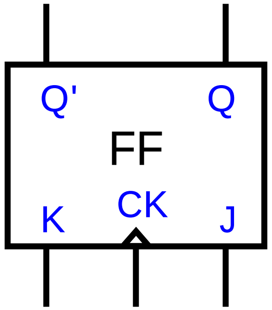
Rising-edge-triggered FF: Inputs can change while CLK is low

Master-slave FF: Incorrect if inputs change while CLK is low

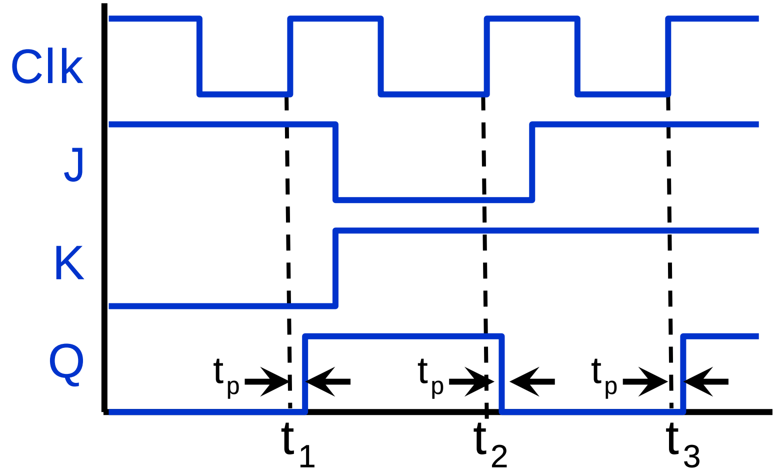
J-K Flip-Flop

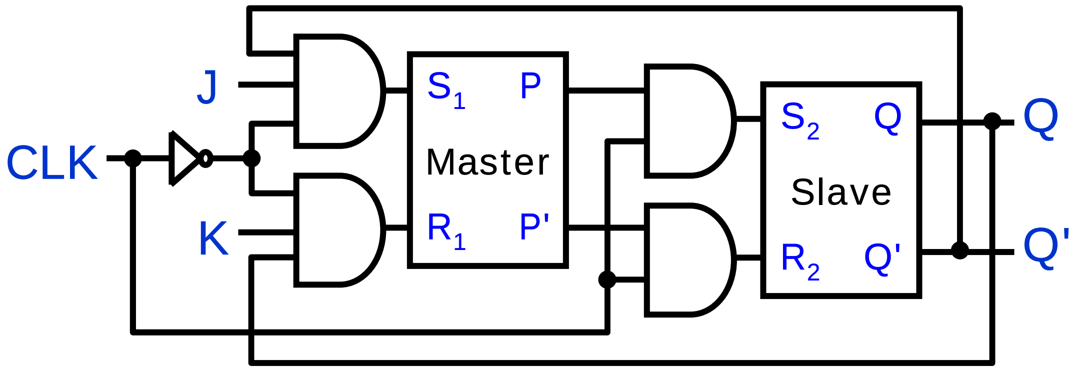
- J-K flip-flop is an extended version of S-R flip-flop
  - $Q^+ = JQ' + K'Q$
  - J corresponds to S (Jump to 1); K corresponds to R (Klear to 0)
  - State toggled when J=K=1

J-K flip-flop

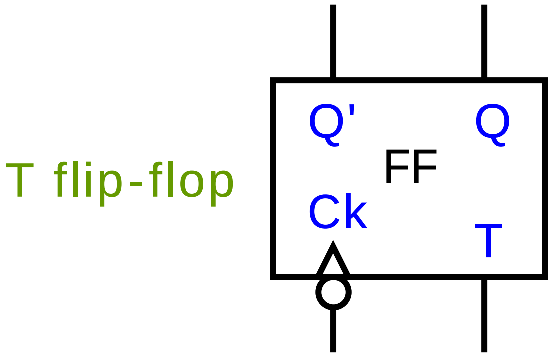


J K Q	Q^+	
0 0 0	0	Hold
0 0 1	1	
0 1 0	0	Clear to 0
0 1 1	0	
1 0 0	1	Jump to 1
1 0 1	1	
1 1 0	1	Toggle
1 1 1	0	





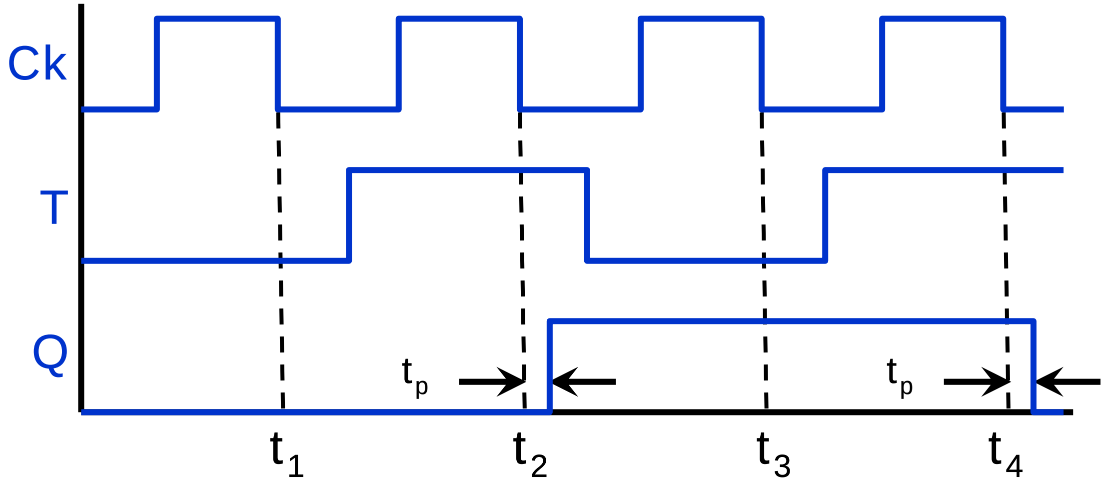
T Flip-Flop



T Q	Q^+	
0 0	0	hold
0 1	1	
1 0	1	toggle
1 1	0	

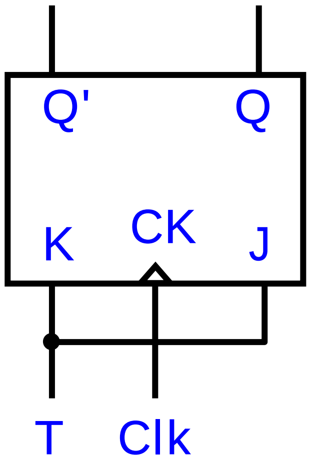
$$Q^+ = T'Q + TQ' = T \oplus Q$$

$$T = 0 \Rightarrow Q^+ = Q$$
$$T = 1 \Rightarrow Q^+ = Q'$$

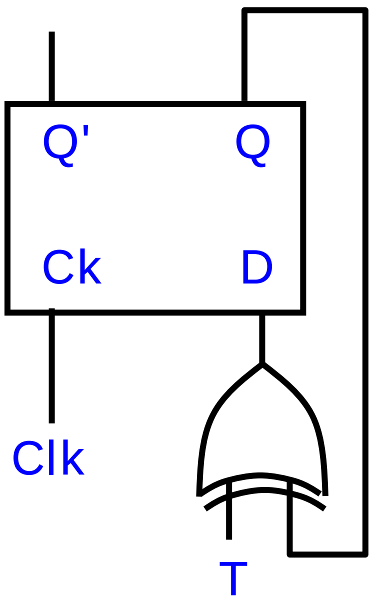


T Flip-Flop Implementation

- Conversion of J-K to T
  - Connect J and K inputs of a J-K FF together
  - $Q^+ = JQ' + K'Q \Rightarrow$
    $Q^+ = TQ' + T'Q$

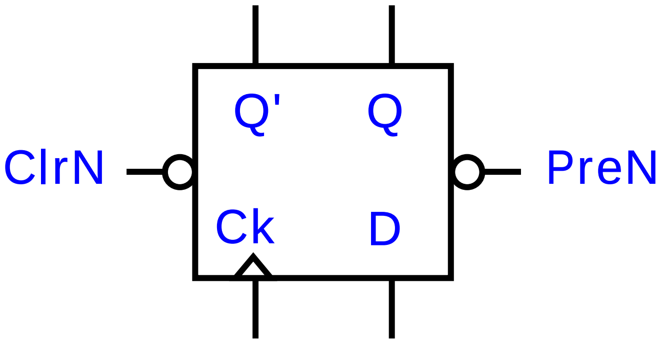


- Conversion of D to T
  - Let $D = Q \oplus T$
  - $Q^+ = D \Rightarrow$
    $Q^+ = Q \oplus T$



Flip-Flops with Additional Inputs
Asynchronous Clear and Preset

- Flip-flops often have additional inputs to set the flip-flops to an initial state independent of the clock

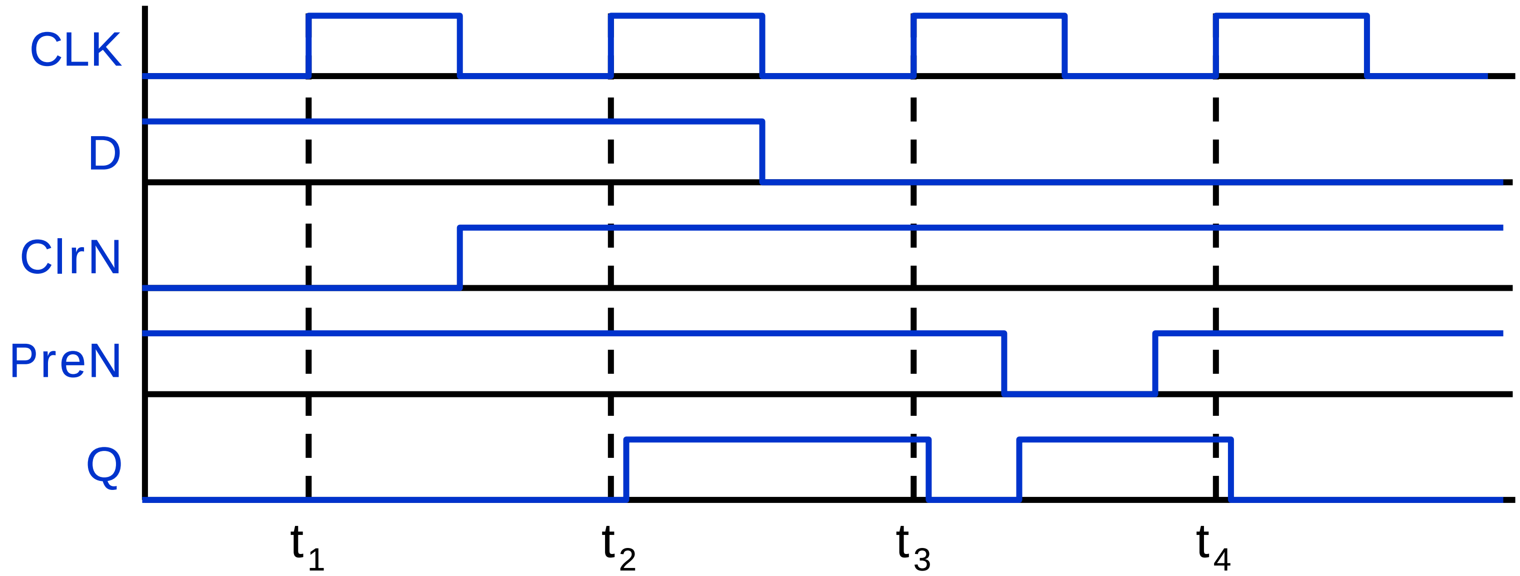


Ck	D	PreN	ClrN	Q^+
x	x	0	0	(not allowed)
x	x	0	1	1
x	x	1	0	0
↑	0	1	1	0
↑	1	1	1	1
0,1,↓	x	1	1	Q (no change)

- ClrN and PreN are **asynchronous** clear and preset inputs (they override the Ck and D inputs)
- ClrN and PreN are **active low** signals
- When ClrN=PreN=1, the FF is in normal operation
- 0 should not be applied to ClrN and PreN simultaneously

Flip-Flops with Additional Inputs
Asynchronous Clear and Preset

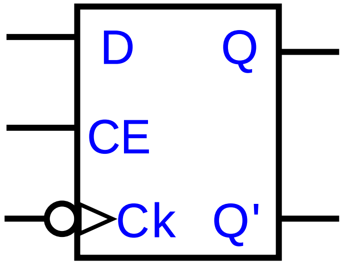
- Timing diagram for D flip-flop with asynchronous clear and preset



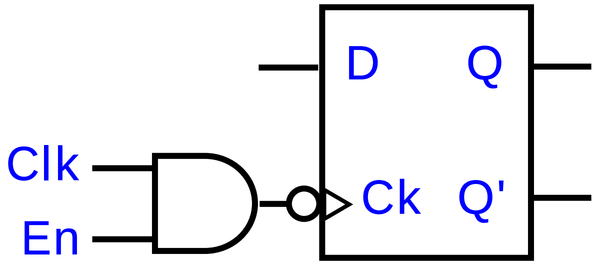
Flip-Flops with Additional Inputs
Clock Enable

- D flip-flop with clock enable (CE)

D-CE symbol

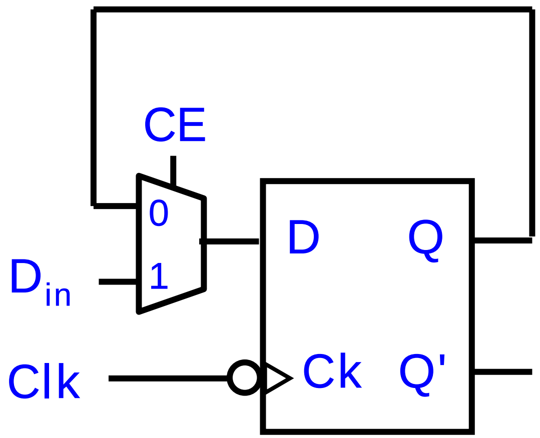


Implementation 1: gating the clock



Loss of synchronization when
1) clock arrive at some FFs at different times
2) En changes at the wrong time

Implementation 2: no clock gating



$$Q^{+} = D = Q\cdot(CE)' + D_{in}\cdot(CE)$$

No synchronization problem

Summary

- Latch (w/o clock input) vs. flip-flop (w/ clock input)
- Propagation delay, setup time, hold time
- Present (current) state, next state
- Characteristic (next-state) equations
  - $Q^+ = S+R'Q$ $(SR=0)$ (S-R latch or flip-flop)
  - $Q^+ = GD+G'Q$ (gated D latch)
  - $Q^+ = D$ (D flip-flop)
  - $Q^+ = D \cdot CE+Q \cdot CE'$ (D-CE flip-flop)
  - $Q^+ = JQ'+K'Q$ (J-K flip-flop)
  - $Q^+ = T \oplus Q = TQ'+T'Q$ (T flip-flop)
- Restrictions
  - For S-R latch/flip-flop, S and R can not be 1 simultaneously
  - For master-slave S-R flip-flop, S and R should not change during the half clock cycle preceding the active edge
  - Setup and hold time constraints