Lecture 10 Flip-Flops/Latches

- Sequential switching network
  - Output depends on present input and past sequence of inputs.
  - Need to remember past history.
  - Flip-flop (latch) is a memory that has a pair of complementary outputs.

Gate Delay

- Propagation delay
- Timing diagram

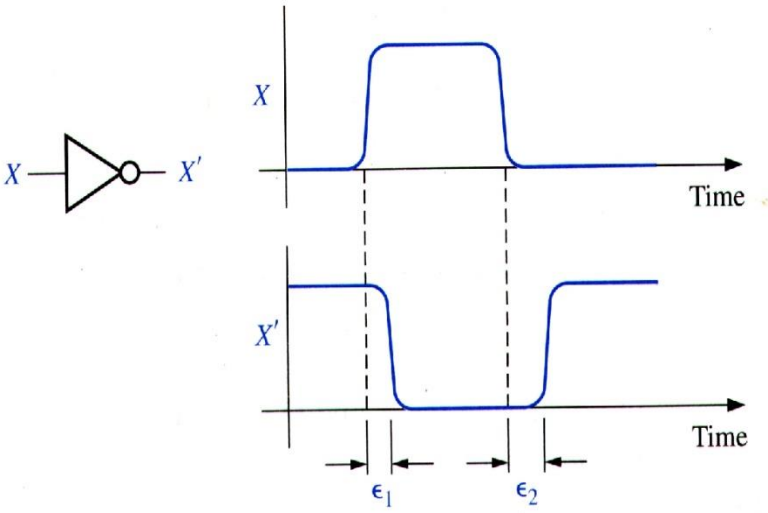
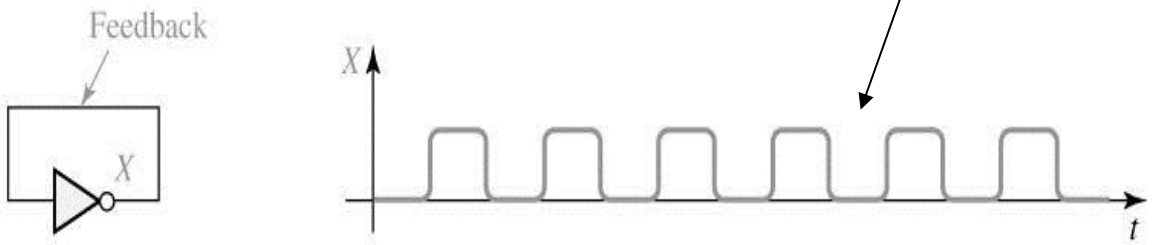


Figure 11–1
Propagation Delay in an Inverter

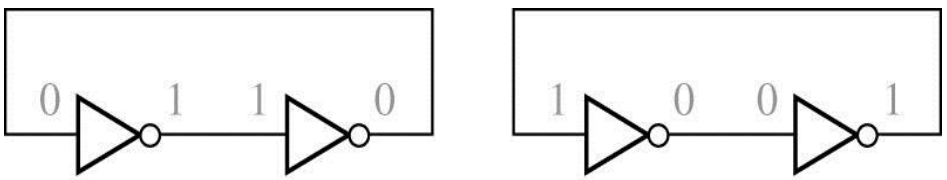
Network with Feedback

- Inverter with feedback.
  - Propagation delay (d = ½ period of CK)
  - Oscillate between 1 and 0.



(a) Inverter with feedback (b) Oscillation at inverter output

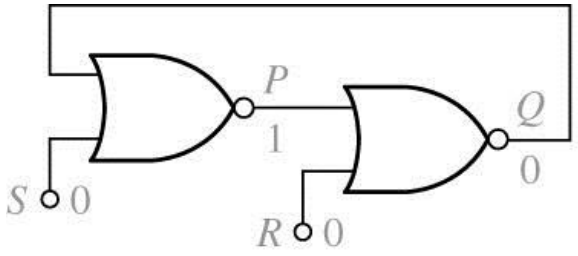
- Stable state



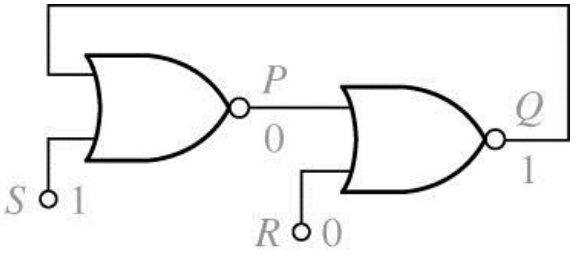
(a) (b)

S-R Latch

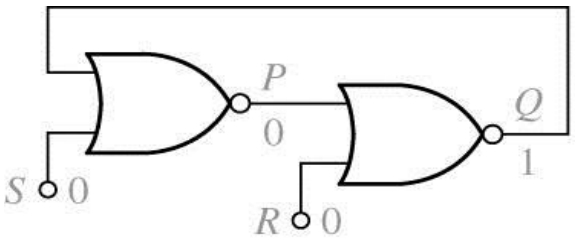
- Set-reset latch
  - Use NOR gate to construct a stable state network



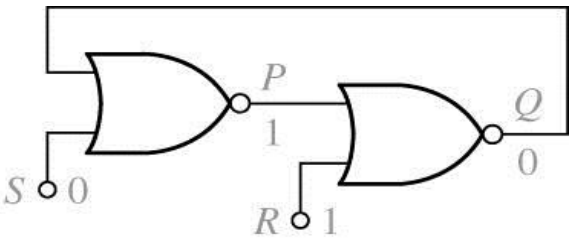
(a)



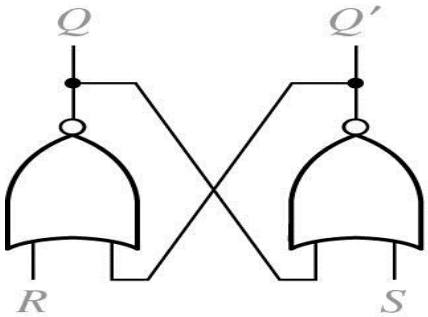
(b)



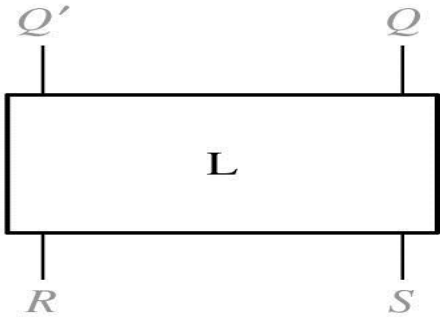
(a)



(b)



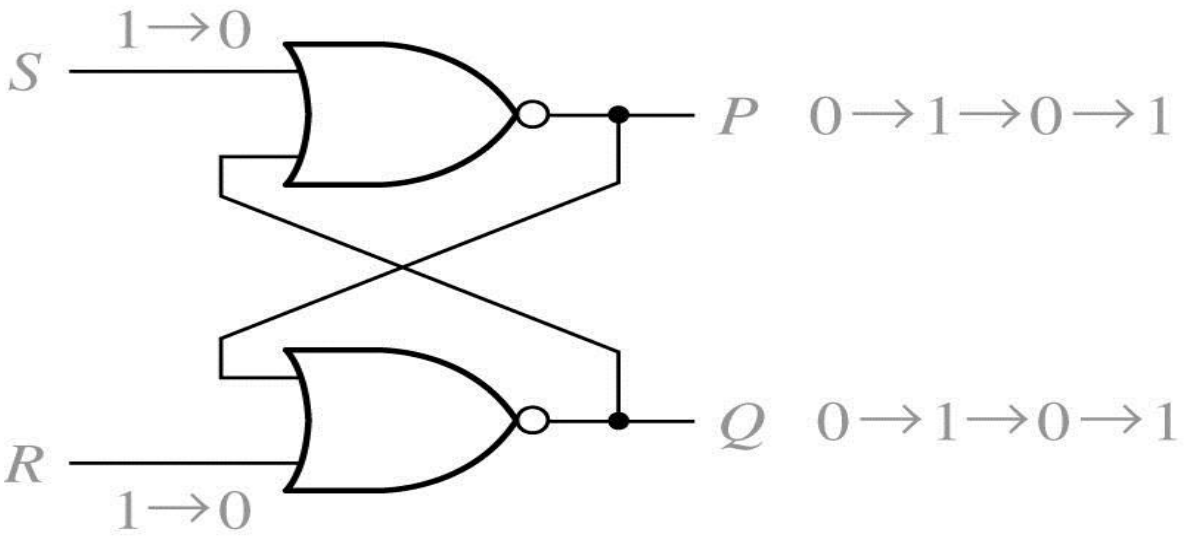
(a)



(b)

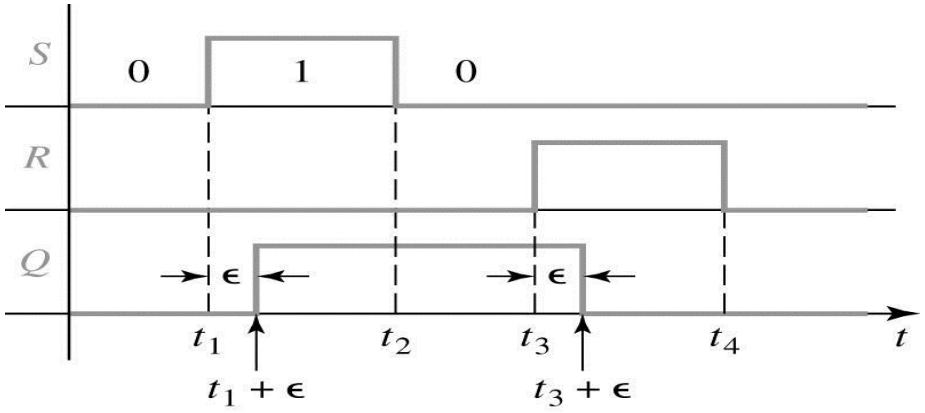
S-R Latch (cont.)

- When S=R=1, the S-R latch will not operate properly. (Is it a stable state, if S=R=1?)
  - Q an P are not complementary.
  - If S=R=1 changed to S=R=0, then the network will oscillate assuming both gates have the same delay. (Critical race occurs)



S-R Latch Timing and State

- S duration > delay time

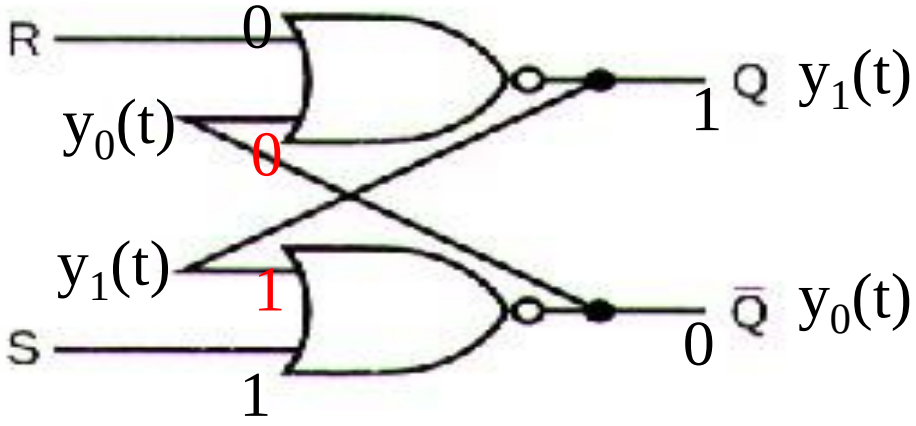


- S-R latch behavior
  - Present state
    - The state of Q output at the time the input signals are applied.
  - Next state
    - The state of Q output after the latch has reacted to the input signals.

$S(t)$	$R(t)$	$Q(t)$	$Q(t+\epsilon)$
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	— } inputs not allowed
1	1	1	— }

S R latch Analysis

- Total state table
- If next state = present state, stable

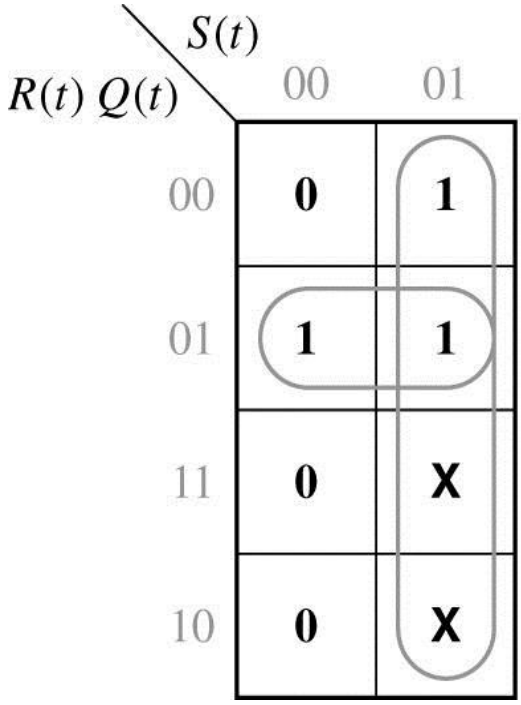


Present state $(y_0(t), y_1(t))$	Current input $(S(t), R(t))$ (0, 0)	(0, 1)	(1, 0)	(1, 1)
(0, 0)	(1, 1)	(1, 0)	(0, 1)	(0, 0)
(0, 1)	(0, 1)	(0, 0)	(0, 1)	(0, 0)
(1, 0)	(1, 0)	(1, 0)	(0, 0)	(0, 0)
(1, 1)	(0, 0)	(0, 0)	(0, 0)	(0, 0)

Next state $(y_0(t + t_{pd}), y_1(t + t_{pd}))$

K-map for Q(t+ε)

- $Q^+ = S + R'Q$ (SR=0)
  - S and R can not be 1 at the same time.
  - Q: present state
  - Q^+: next state
  - Next state equation or characteristic equation.

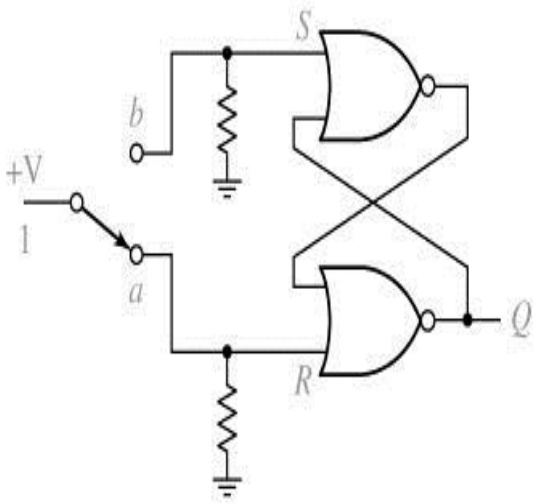


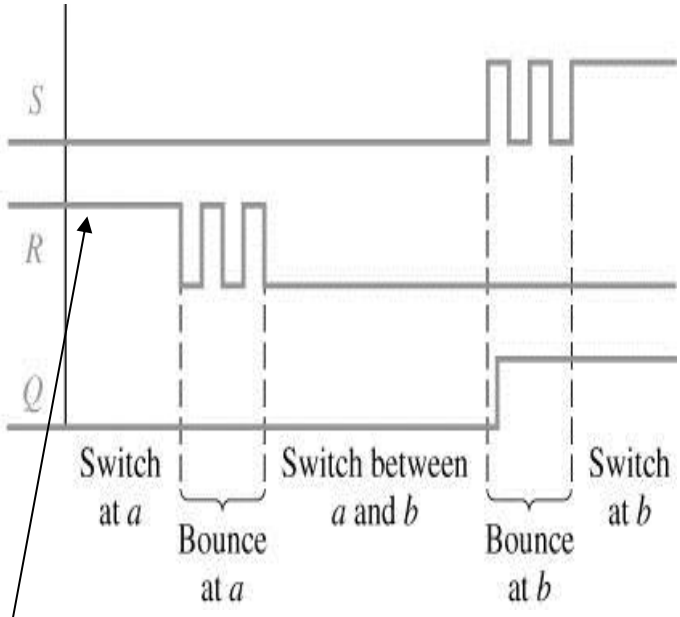
$$Q(t+\epsilon) = S(t) + R'(t)\,Q(t)$$

Debouncing Circuit

- Use S-R latch for debouncing.
- Pull-down resistors
- a switch to b.

S= 1 set Q
S = 0, (R=0)
Q remained



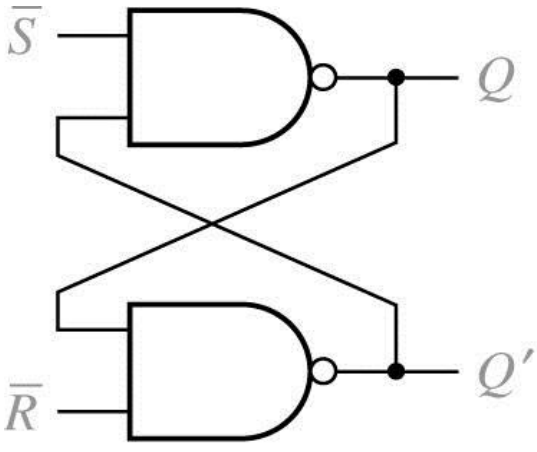


R = 1 set Q = 0

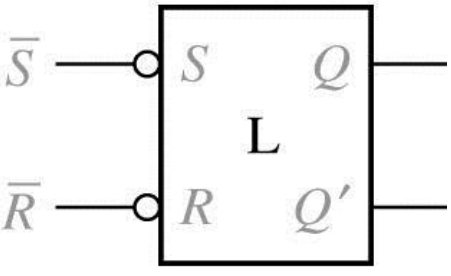
S-R Latch using NAND gates

- S#-R# Latch, when S#= 0 sets Q = 1 and R#=0 resets Q = 0

S#	R#	Q	Q+
1	1	0	0
1	1	1	1
1	0	0	0
1	0	1	0
0	1	0	1
0	1	1	1
0	0	0	-
0	0	1	-



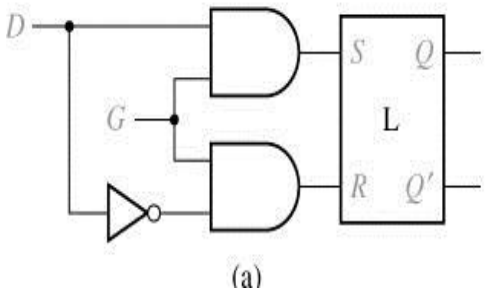
(a)



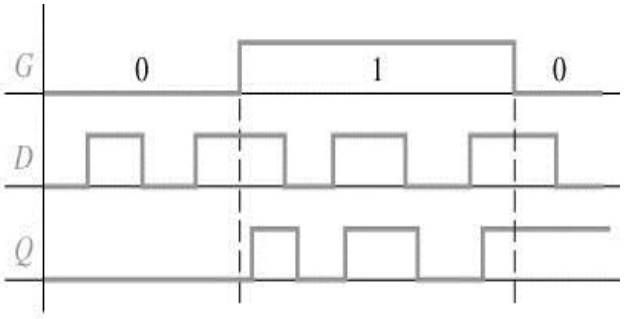
(b)

Gated D Latch

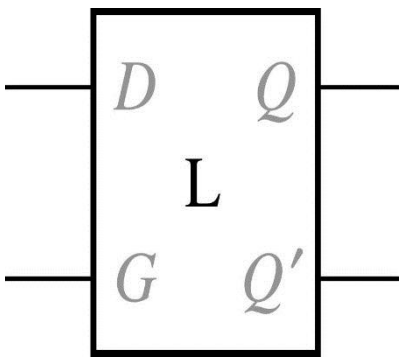
- Gate input G
  - Transparent latch (when G= 1, Q = D)



(a)



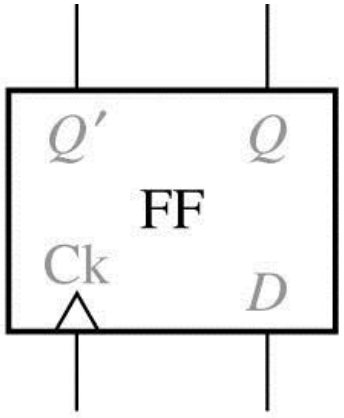
(b)



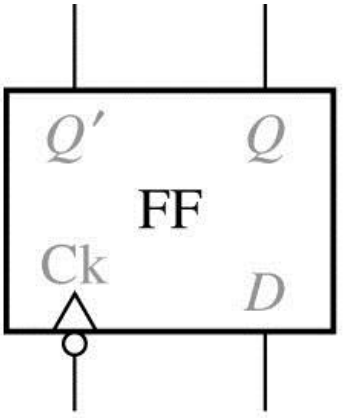
G	D	Q	Q+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Edge-Triggered D Flip-Flop

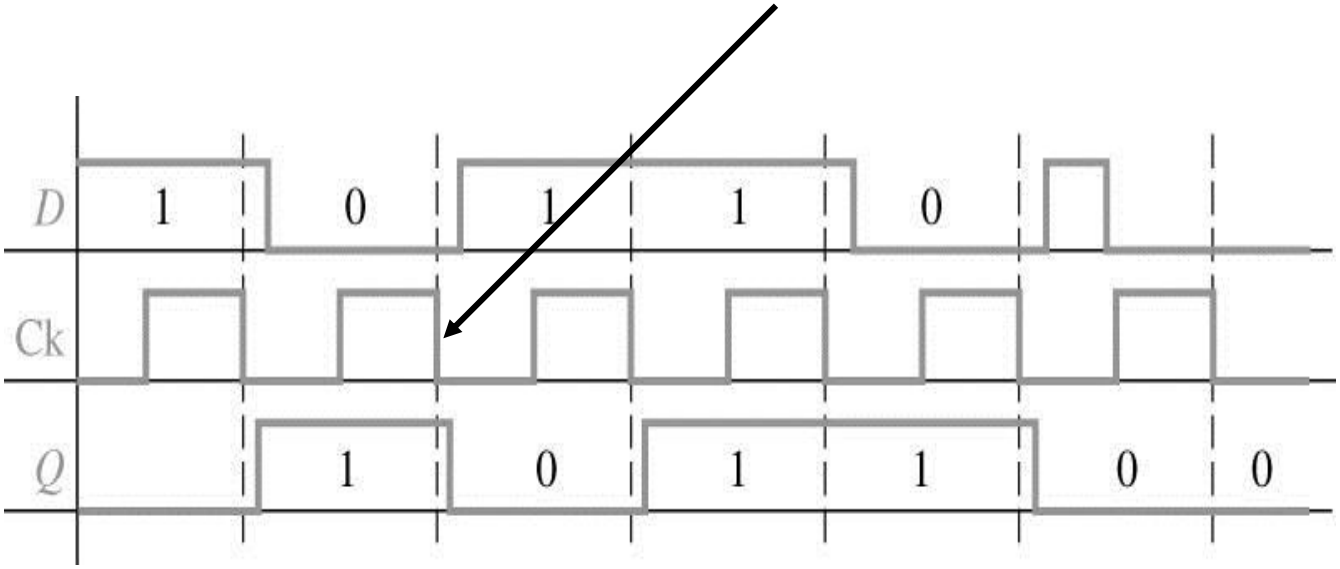
- Output changes in response to clock edge



(a) Rising-edge trigger

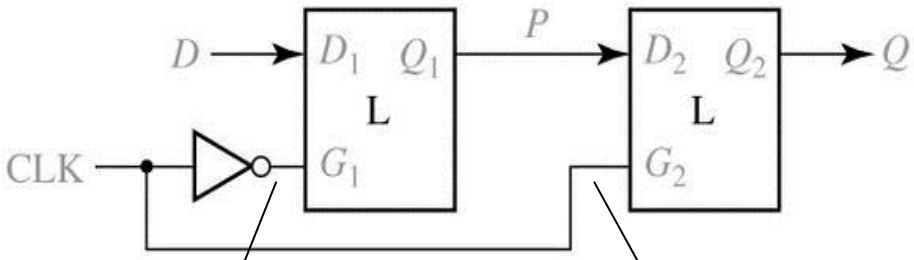


(b) Falling-edge trigger

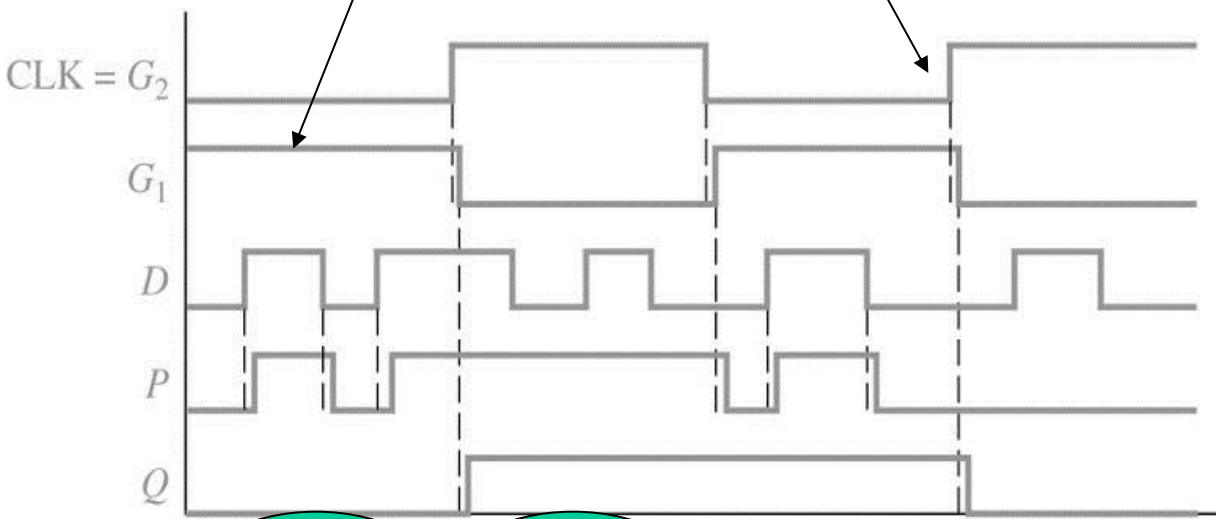


D Flip-Flop

- Using two gated D latches
- Output changes occur at the rising edge
  - CK = H, output follows input.
  - CK = L, output remains



(a) Construction from two gated D latches



P follows D

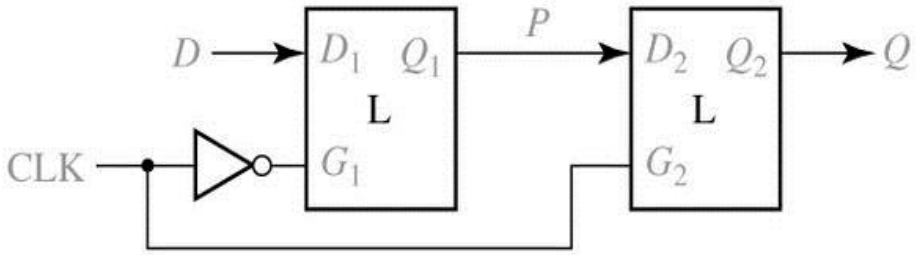
Q follows P

P follows D

Q follows P

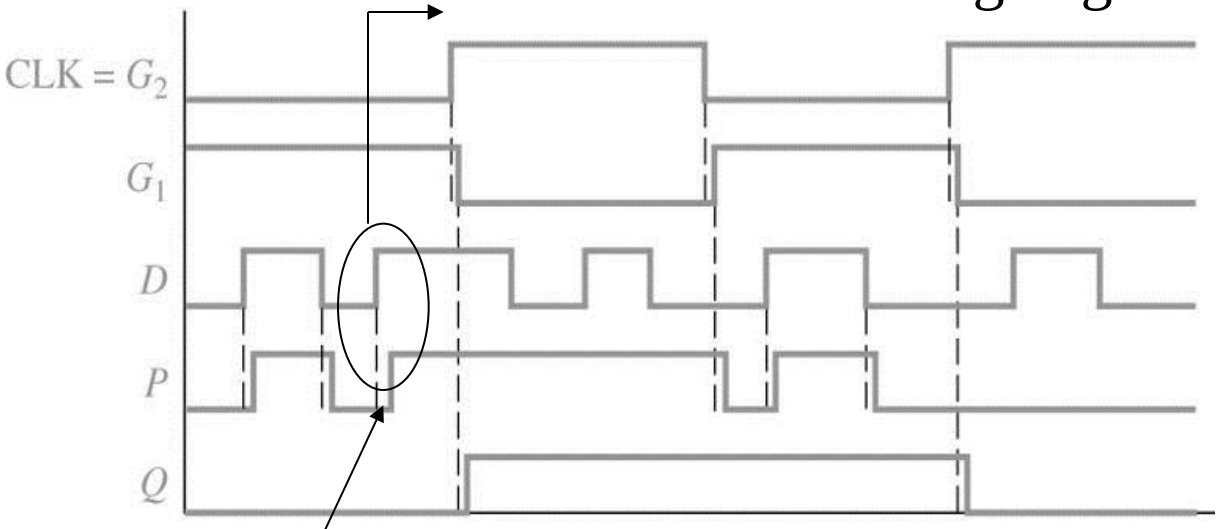
D Flip-Flop

- Output changes occur at the rising edge



(a) Construction from two gated D latches

Move towards rising edge of G_2

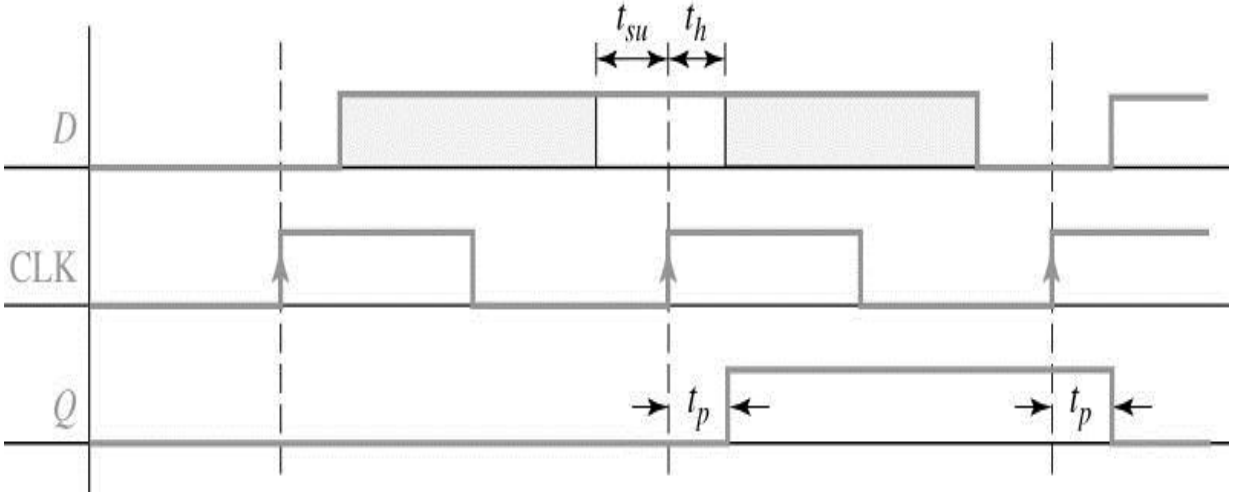


(b) Time analysis

Δt = D1 latch's propagation delay
This is the setup time for this FF.

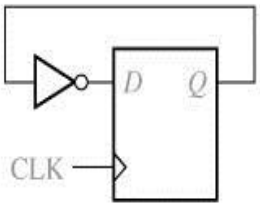
Setup and Hold Time

- Edge-Triggered D FF
  - Propagation delay of a FF is the time btw the active edge of the clock and resulting change in the output

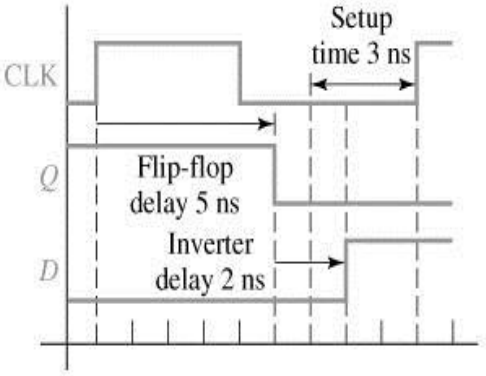


Minimum Clock Period

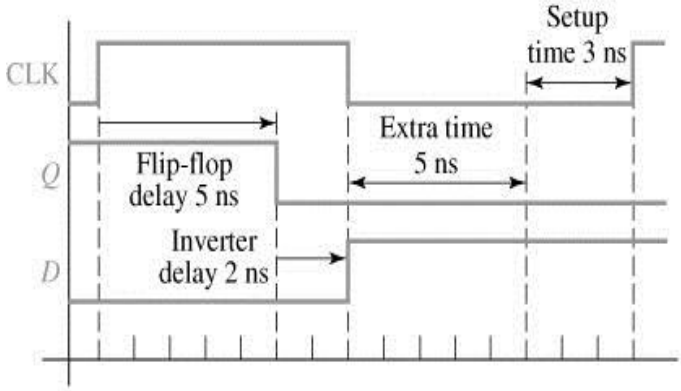
- tp
  - inverter = 2 ns,
  - FF = 5ns
- Setup time 3 ns



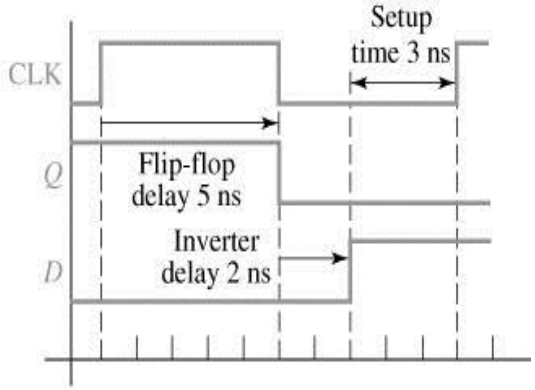
(a) Simple flip-flop circuit



(b) Setup time not satisfied



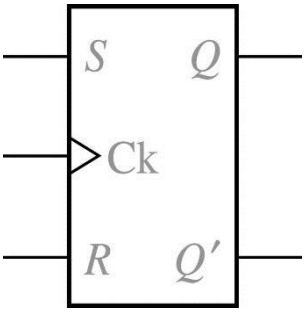
(c) Setup time satisfied

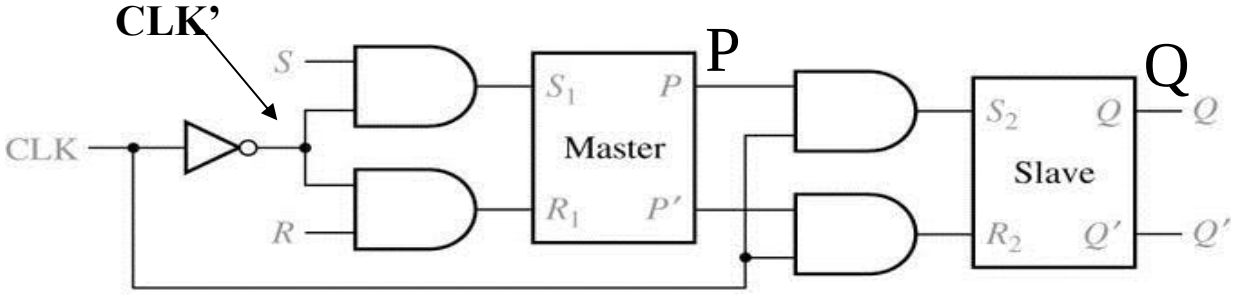


(d) Minimum clock period

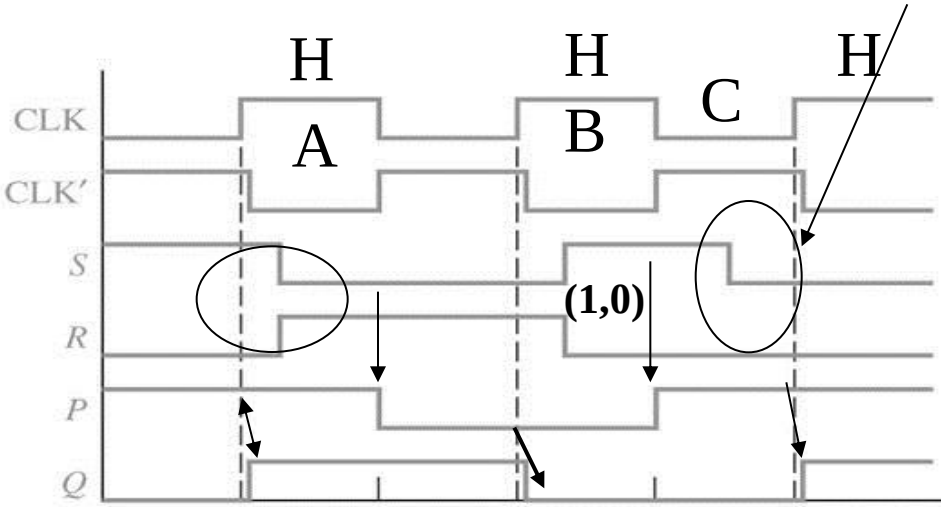
Master-Slave S-R Flip-Flop

- **FF has a clock input.**
- **Change state after rising edge**
- **This figure shows a case that (S, R) are changed at CLK = H (A,B), then at CLK=L for C.**
  - **What if (S,R) becomes (1,0) nears t5? (OK)**
- **Actually (S,R) should be set to change (for set or reset) at the time near the end of M on and before S on. (to avoid miss-interpretation at t5)**





(a) Implementation with two latches



(b) Timing analysis

Input changed at CLK low to (0,0) has no effect to P due to S-R latch feature.

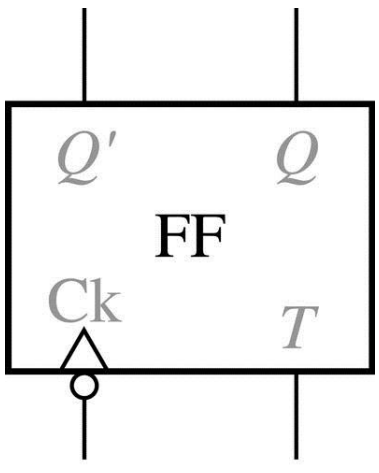
Explaining at CLK = H clock is correct.

Toggle FF

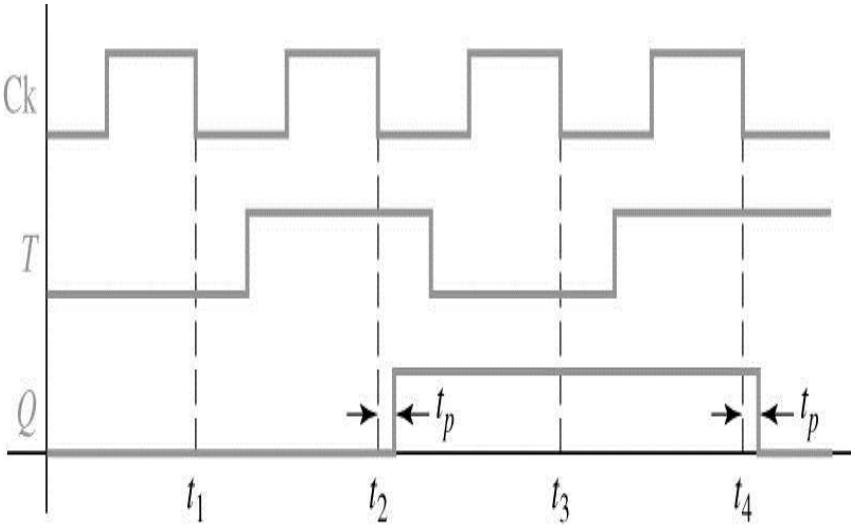
- T flip-flop
  - Single input
  - When T = 1, at clock edge, T FF changes state. If T = 0, no state changes.

T	Q	Q^+
0	0	0
0	1	1
1	0	1
1	1	0

$Q^+ = T'Q + TQ' = T \text{ xor } Q$



(a)

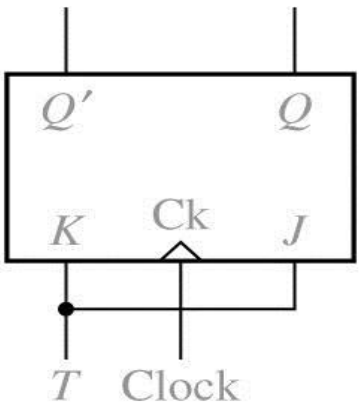


T FF

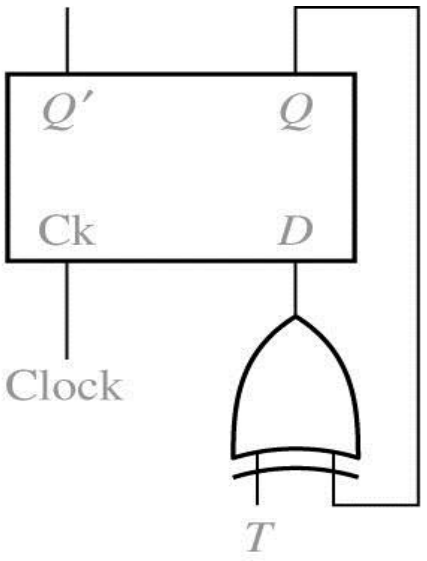
- T flip-flop
  - Converted from D to T
  - $Q^+ = D = Q \text{ xor } T = TQ' + T'Q$

T	Q	Q^+
0	0	0
0	1	1
1	0	1
1	1	0

$Q^+ = T'Q + TQ' = T \text{ xor } Q$



(a) Conversion of J-K to *T*

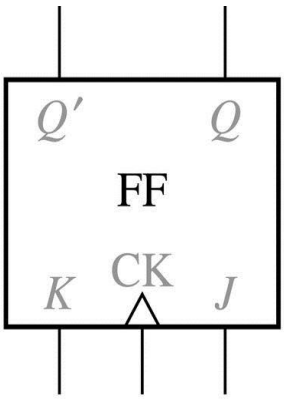


(b) Conversion of *D* to *T*

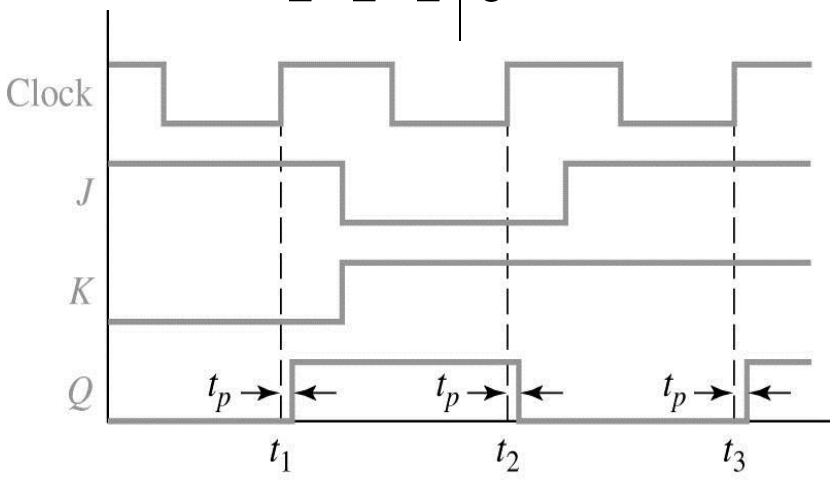
J-K Flip Flop

- J-K FF = S-R FF + T FF.
  - Allow J = K = 1. This case works like a T FF.
  - Split T to J and K

J	K	Q	$Q^+ = JQ' + K'Q$
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



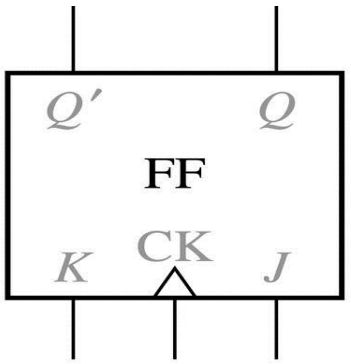
(a) J-K flip-flop



(c) J-K flip-flop timing

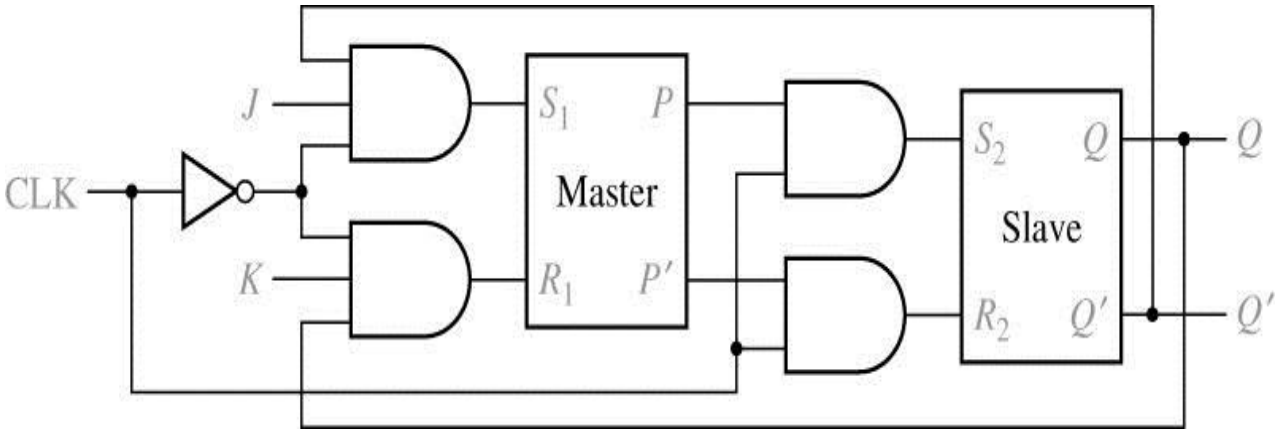
J-K Flip Flop

- J-K FF rising edge trigger



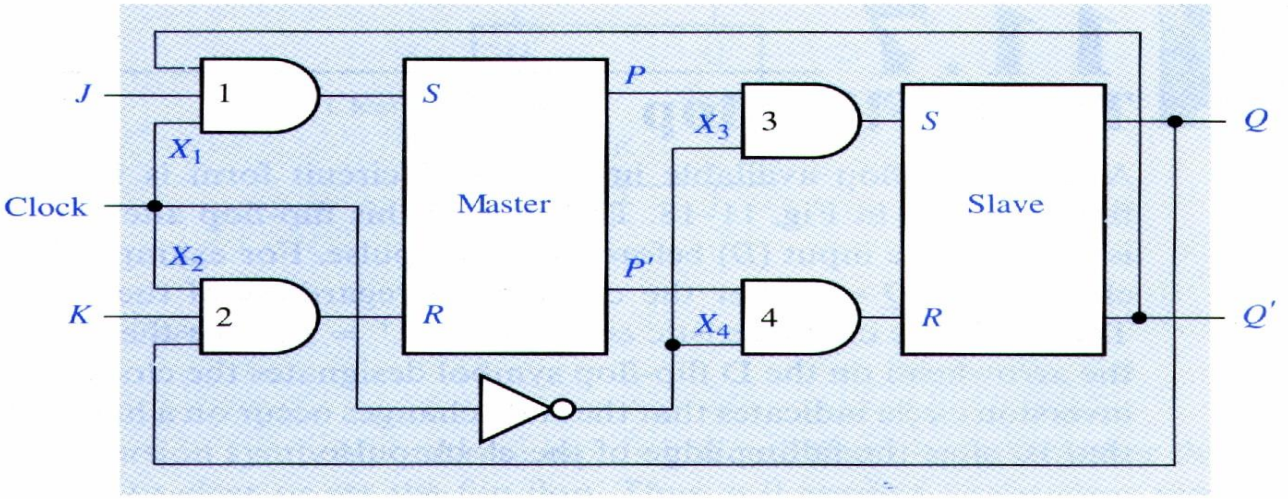
(a) J-K flip-flop

J	K	Q	$Q^+ = JQ' + K'Q$
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

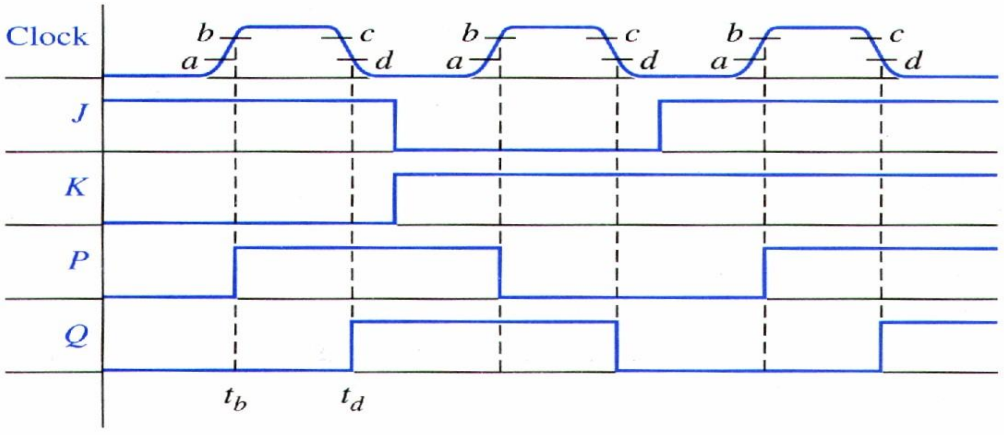


Master-Slave J-K Flip-Flop

- Clocked J-K FF (falling edge)
  - Realization using two S-R latches
  - Note where J and K change.



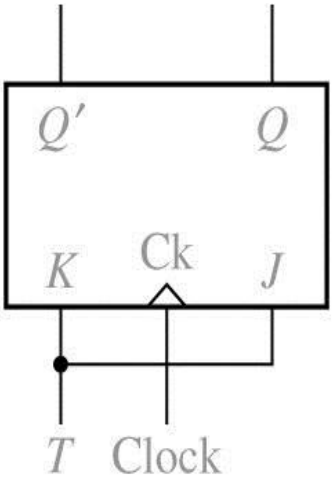
(a) Master-slave J-K flip-flop



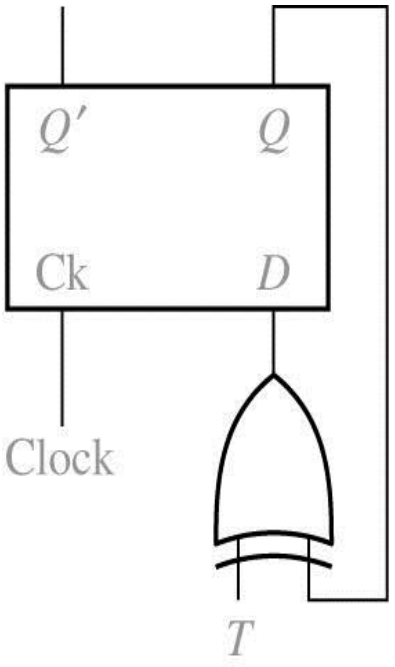
(b) Internal timing diagram for master-slave J-K flip-flop

T Flip-Flop Implementation

- Conversion
  - Using a J-K FF
  - Using a D FF



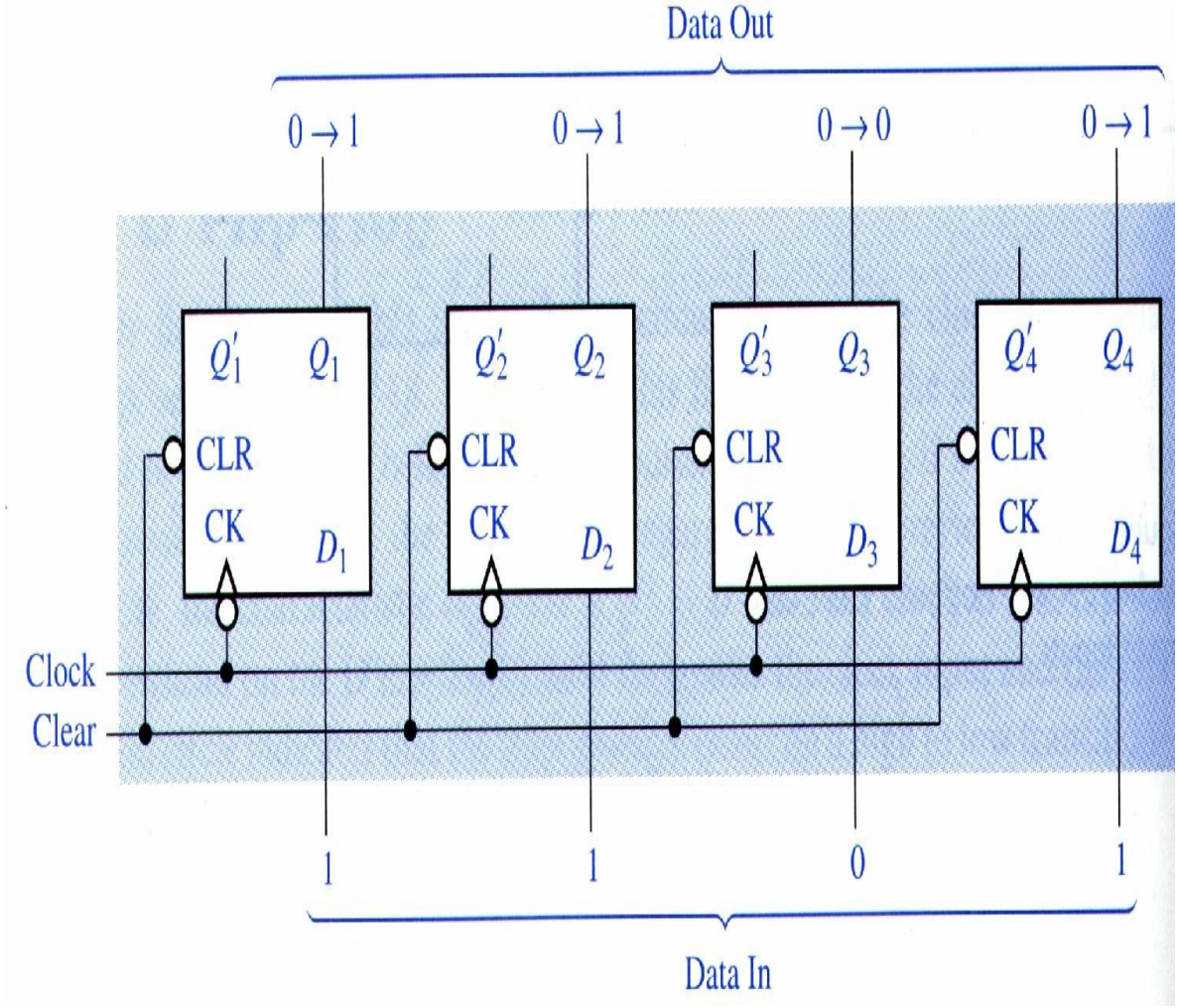
(a) Conversion of J-K to *T*



(b) Conversion of *D* to *T*

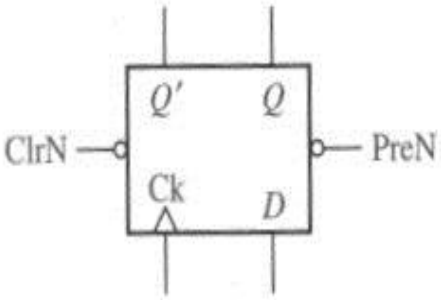
D FF Register

- Register = many clocked D FFs
- $Q^+ = D$



Clear and Presets

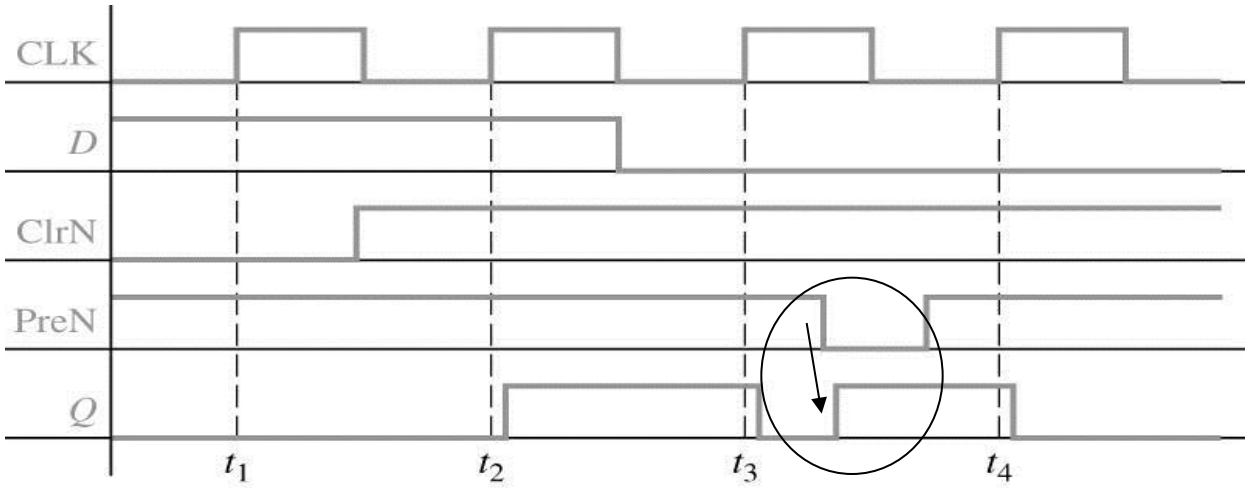
- Active low inputs
- Asynchronous clear and preset



(a)

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)

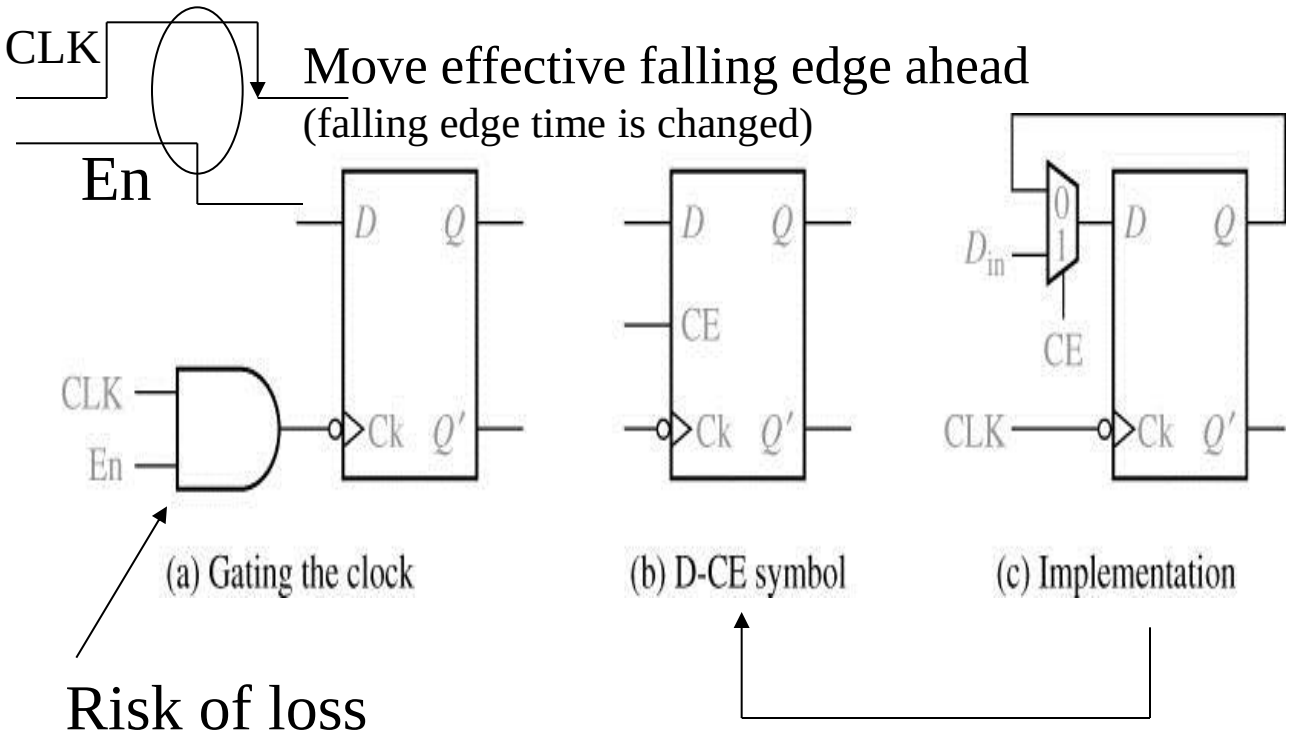
(b)



preset

Clock Enable

- D-CE flip-flop
- Hold existing data even input changes.
- $Q^+ = Q.CE' + D.CE$
- In Fig. c, $Q^+ = D = Q.CE' + D_{in}.CE$
  - No gating in clock line, no synchronization problem.

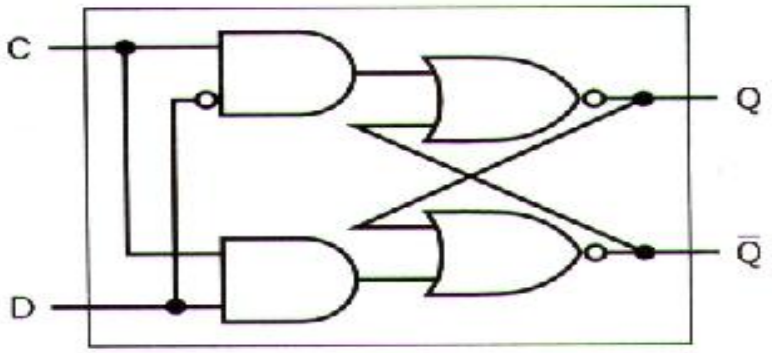


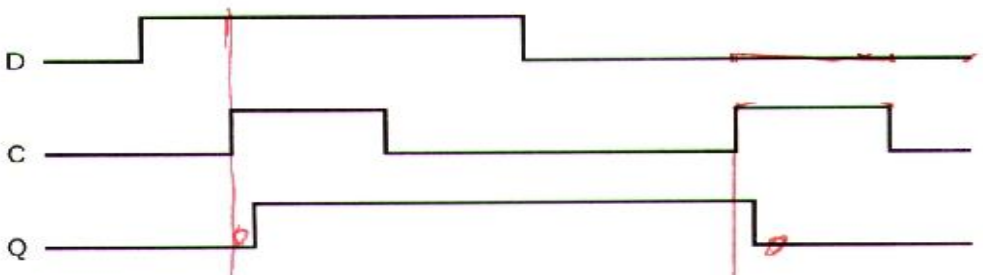
Clocked Latches

Clocked latch:

The state changed whenever the inputs change and the clock is asserted.

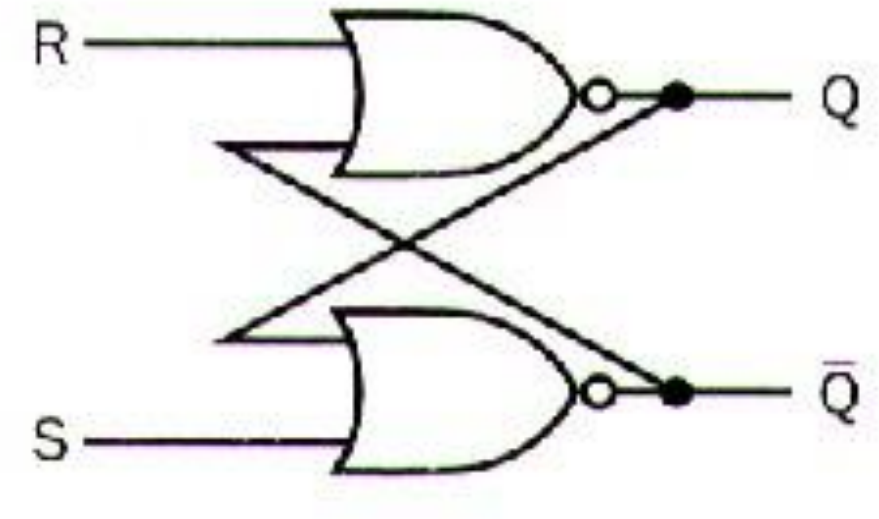
A D latch with NOR gates and clock (level trigger)





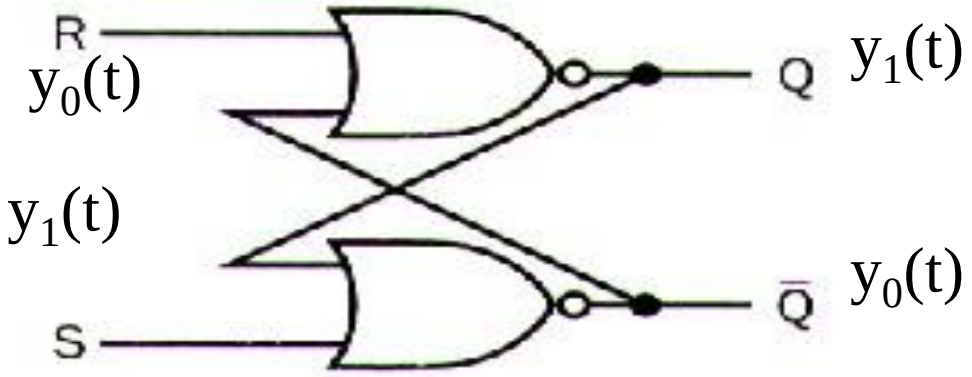
Unclocked Latch

- SR latch
  - State may change if input changes.



S R latch Analysis

- Total state table
- Next internal state

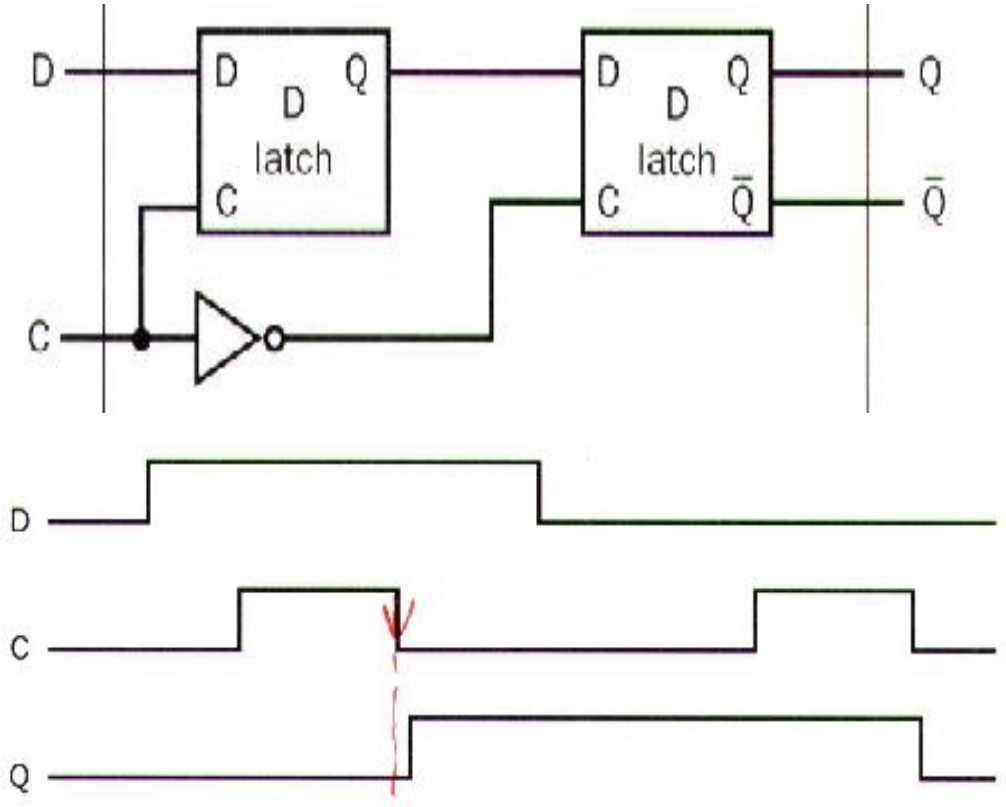


Present state $(y_0(t), y_1(t))$	Current input $(S(t), R(t))$ (0, 0)	(0, 1)	(1, 0)	(1, 1)
(0, 0)	(1, 1)	(1, 0)	(0, 1)	(0, 0)
(0, 1)	(0, 1)	(0, 0)	(0, 1)	(0, 0)
(1, 0)	(1, 0)	(1, 0)	(0, 0)	(0, 0)
(1, 1)	(0, 0)	(0, 0)	(0, 0)	(0, 0)

Next state $(y_0(t + t_{pd}), y_1(t + t_{pd}))$

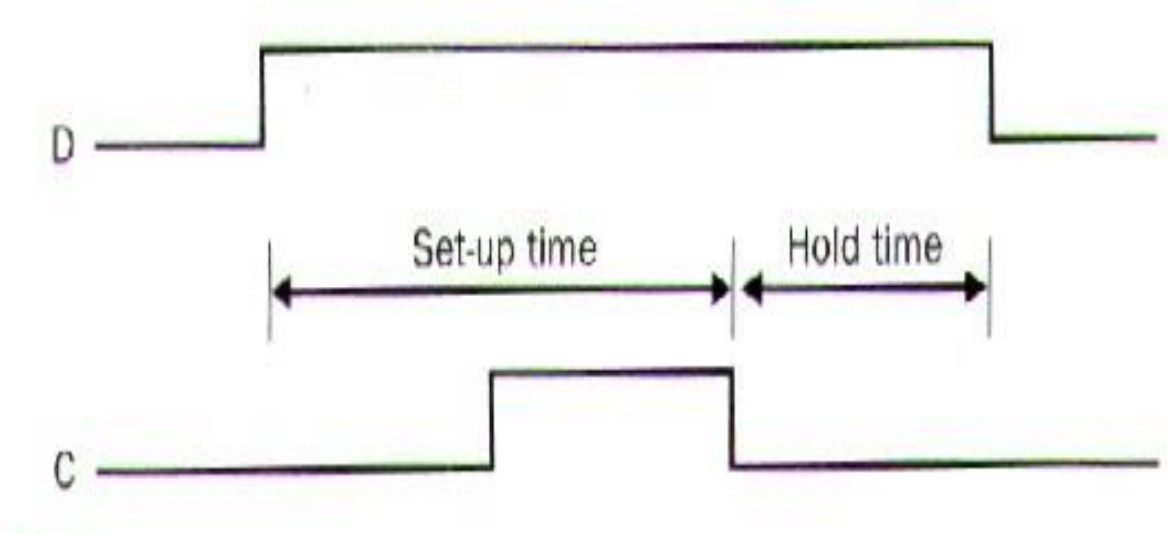
D Flip Flop

- Master latch with a slave latch
  - State changes only at clock edge.
  - Falling edge.



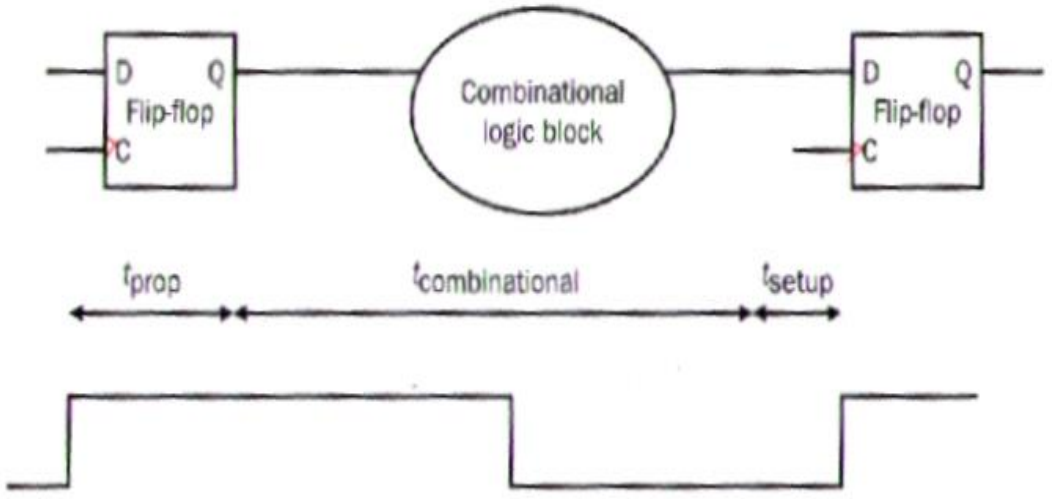
Timing Requirement

- Falling edge trigger
  - Set up time
  - Hold time
    - **Hold time requirement is either 0 or very small.**



Clock Period Requirement

- Clock period requirement
  - t_propagation + t_combinational + t_setup + t_skew
  - t_propagation is the time for FF inputs to FF outputs.
  - t_skew is the time difference when two state elements see a clock edge.



Asynchronous inputs

- Why makes it a synchronous input?
  - Used to change state of a system
  - If not synchronized, the signals may violate the setup time or hold time of a receiving device.
- Metastable behavior
  - State in the middle of 1 and 0.