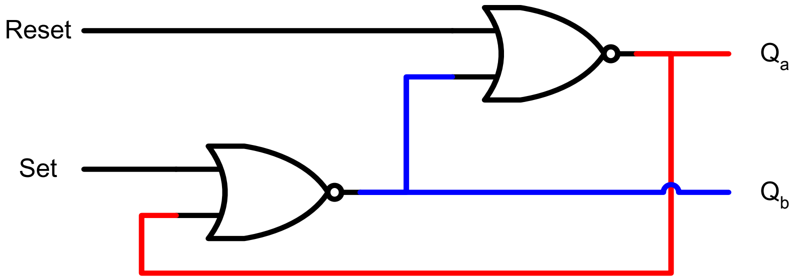
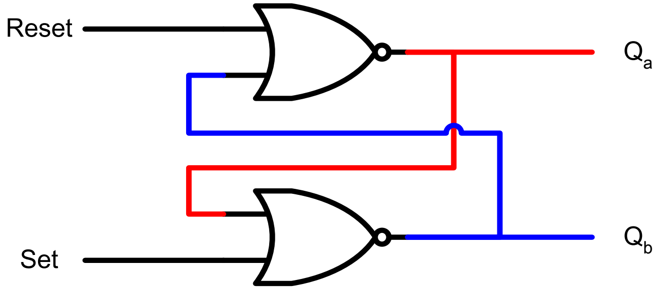
# *CME 2003 Logic Design*

Latch, Flip-flop

# Basic Latch

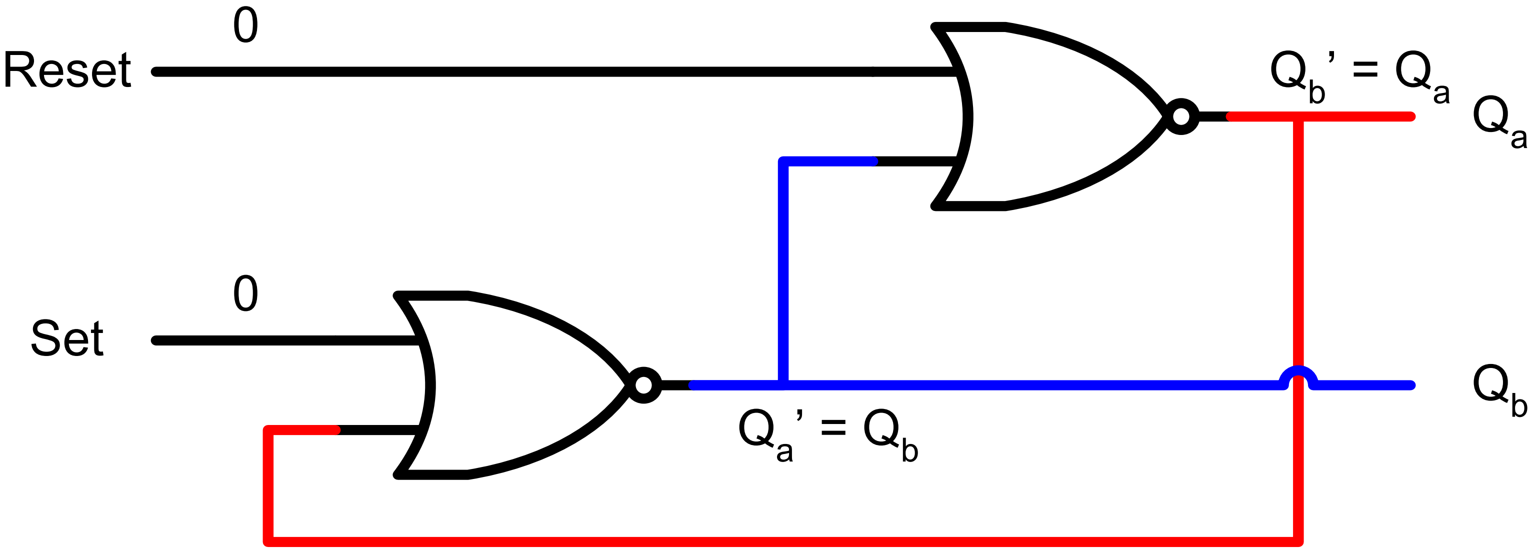
- Both circuit are the same
- The only feedback path is the red line





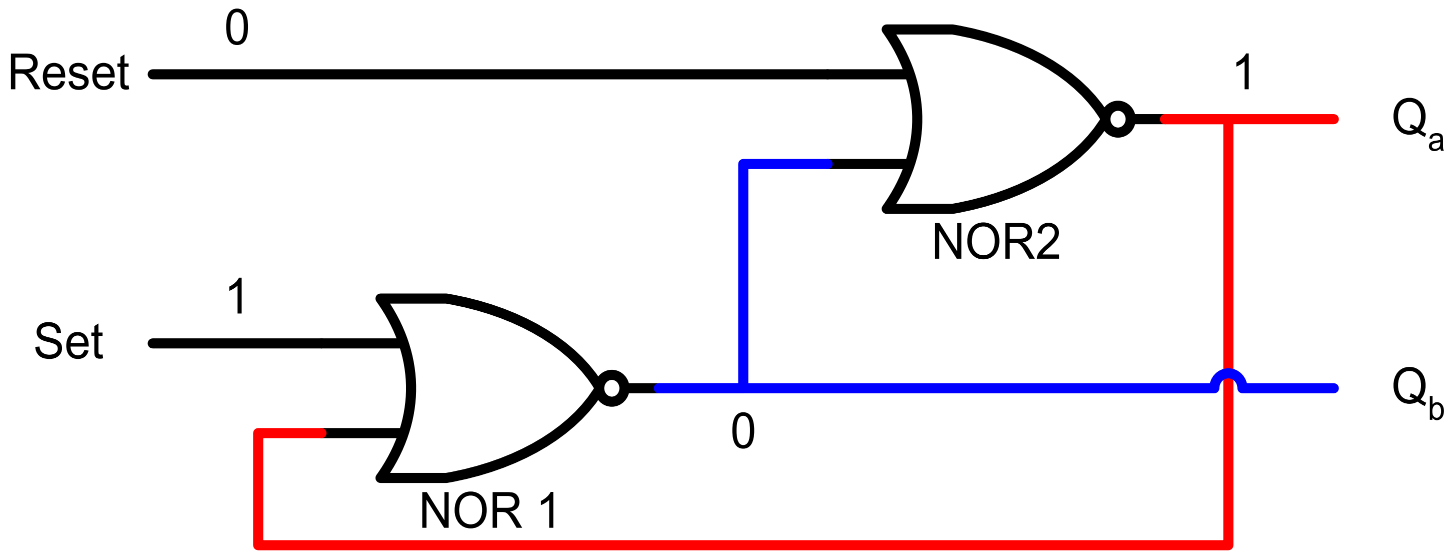
# Basic Latch

- Consider Set = 0, Reset =0



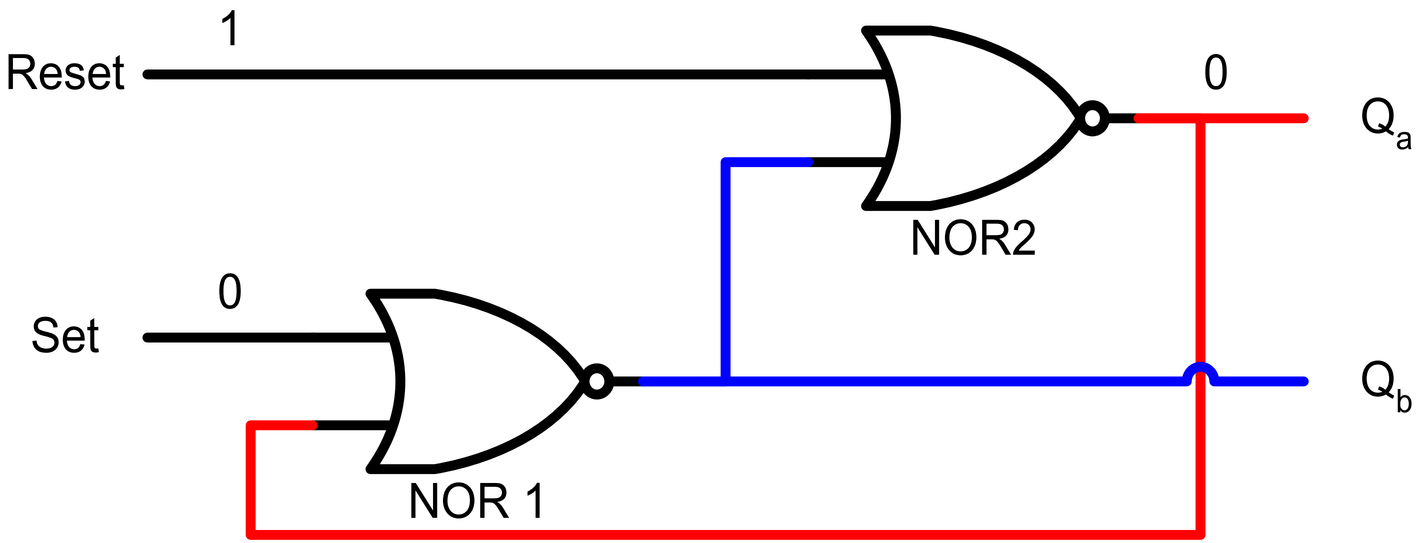
# Basic Latch

- Consider S = 1, R =0
  - As S=1, NOR1 output must be 0
  - As NOR1 ouput = 0 and R =0, NOR2 output must be 1



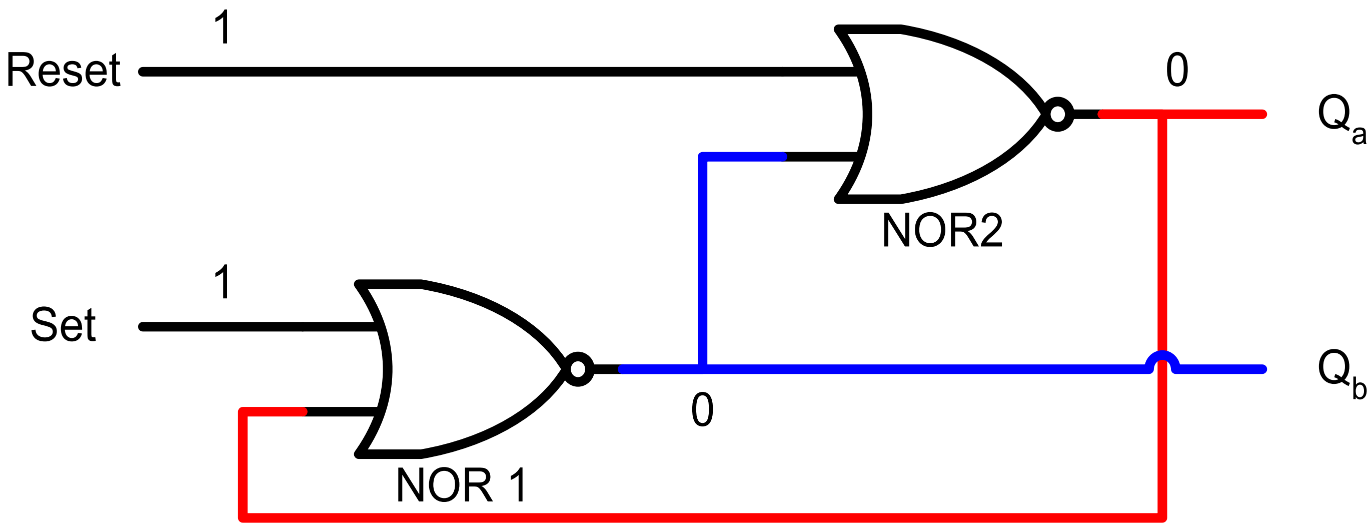
# Basic Latch

- Consider S = 0, R =1
  - As R = 1, NOR2 output must be 0



# Basic Latch

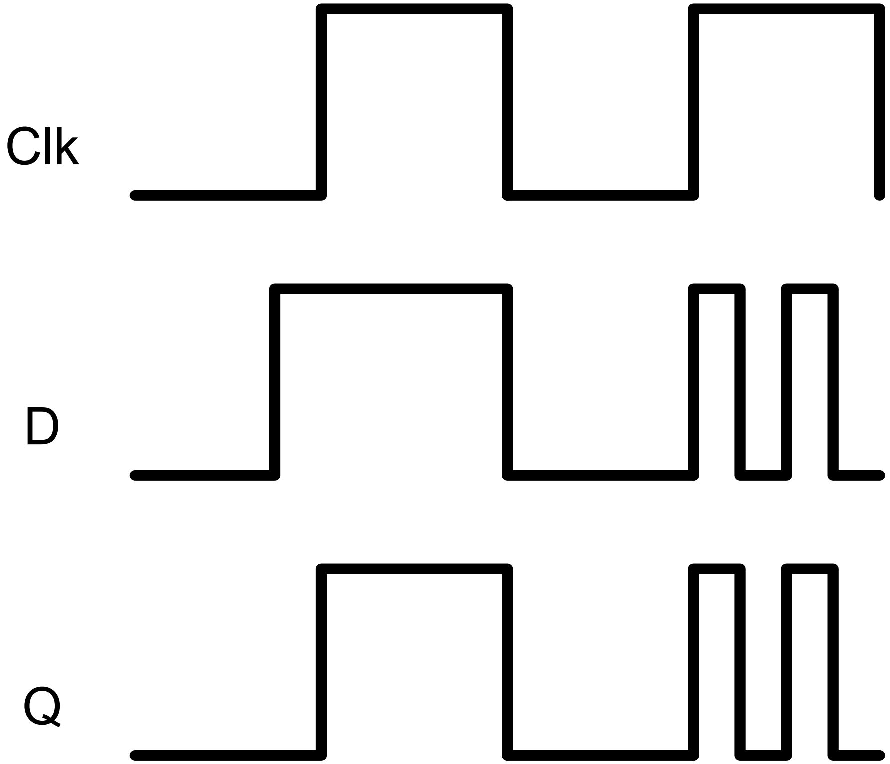
- Consider R = 1 and S =1
  - As R = 1, NOR2 output must be 0
  - As S = 1, NOR1 output must be 0



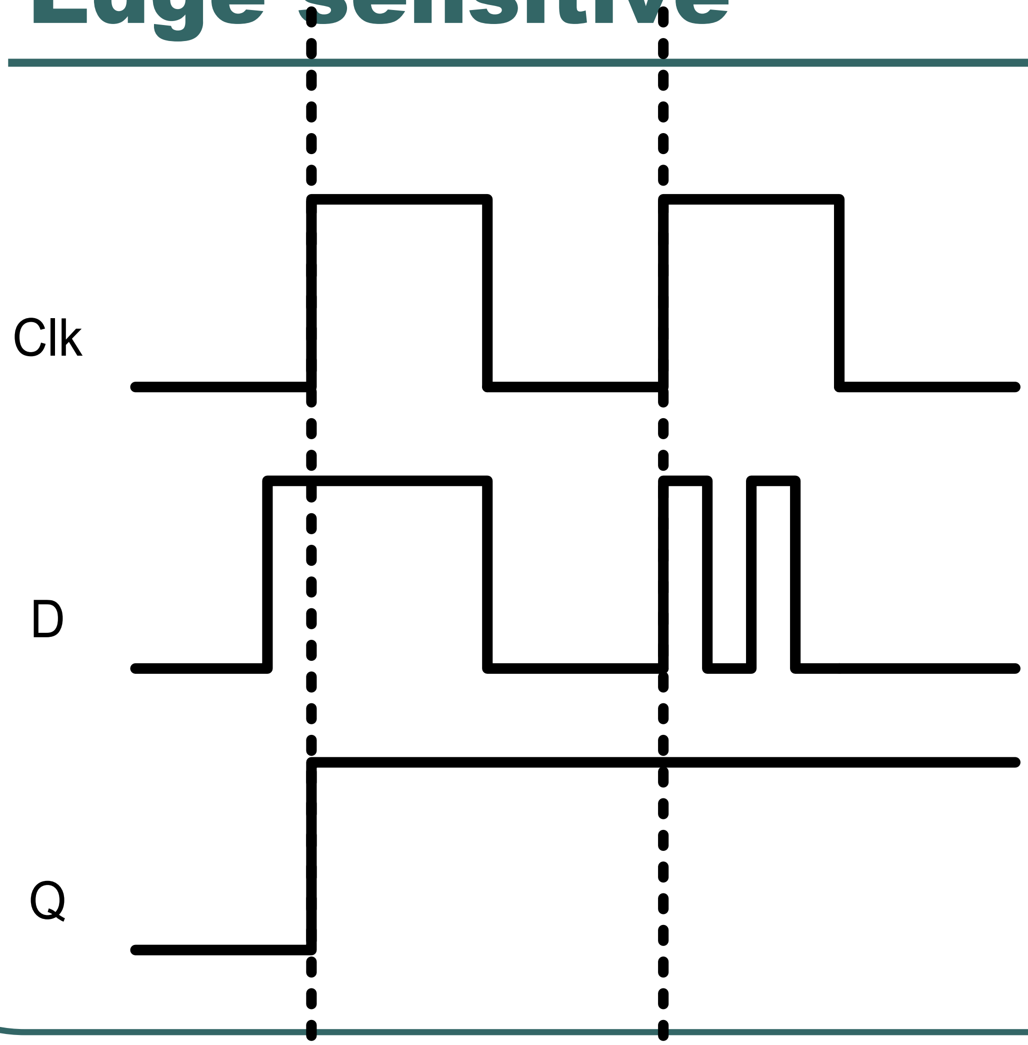
# Level sensitive and edge sensitive

- For a latch and flip-flop (FF), it can be level sensitive or edge sensitive
  - Level sensitive means the latch / FF will copy input D to output Q when Clk = 1
  - Edge sensitive means that the latch / FF will only copy input D to output Q when Clk change from 0 -> 1 (positive edge trigger) / 1 -> 0 (negative edge trigger)

# Level sensitive



# Edge sensitive



# Review on basic latch

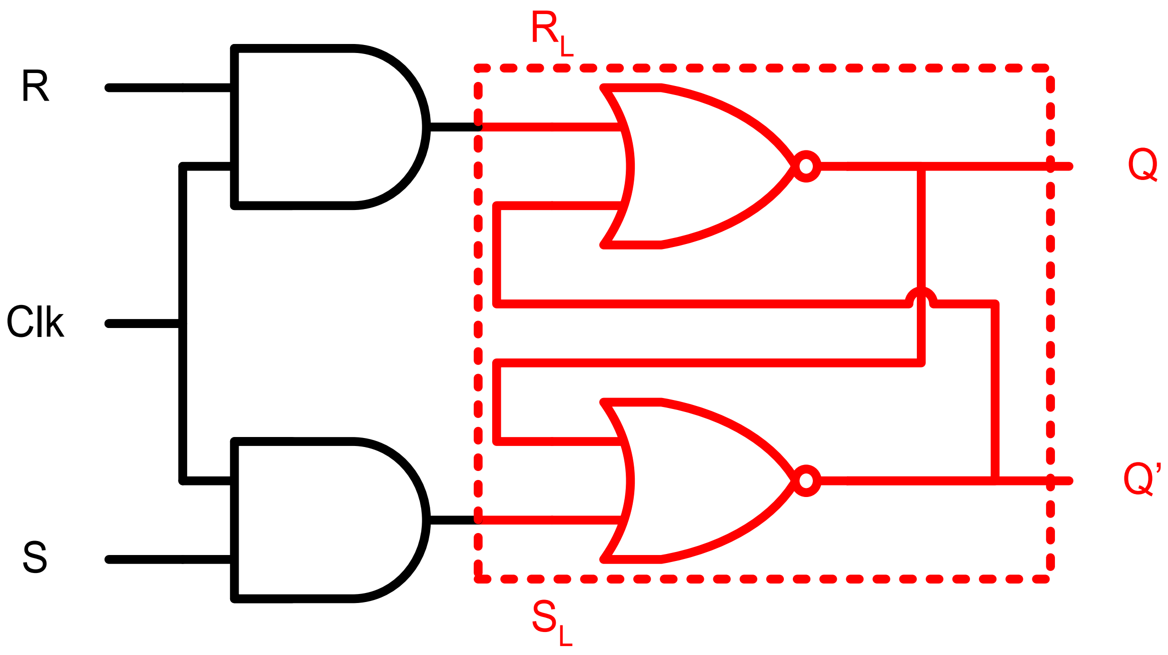
- Truth Table for basic latch

| S | R | Qa | Qb |
|---|---|---|---|
| 0 | 0 | no change | |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

- Also if input changed from S=1 R=1 to S=0 R=0, unstable case occur

# Gated SR latch

- The red part is exactly the basic latch



# Gated SR latch

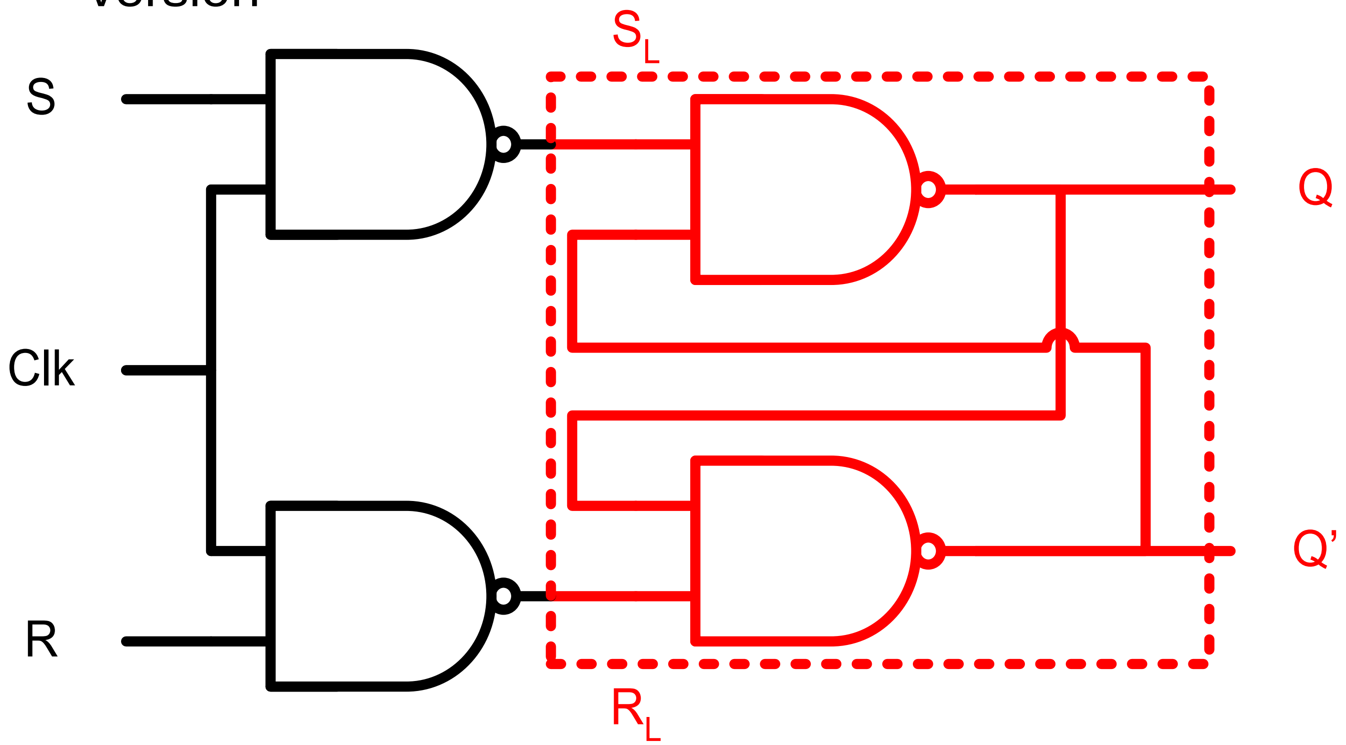
- Gated SR latch has truth table like that

| Clk | S | R | Q(t+1) | => | $S_L$ | $R_L$ |
|---|---|---|---|---|---|---|
| 0 | x | x | Q(t) | => | 0 | 0 |
| 1 | 0 | 0 | Q(t) | => | 0 | 0 |
| 1 | 0 | 1 | 0 | => | 0 | 1 |
| 1 | 1 | 0 | 1 | => | 1 | 0 |
| 1 | 1 | 1 | x | => | x | x |

- $S_L$ = Clk AND S
- $R_L$ = Clk AND R

# Gated SR latch

- Gated SR latch that only use NAND gate
  - Noted that S is at top instead of R compared to NOR version



# Gated D latch

- The follow truth table is the function we want for Gated D latch

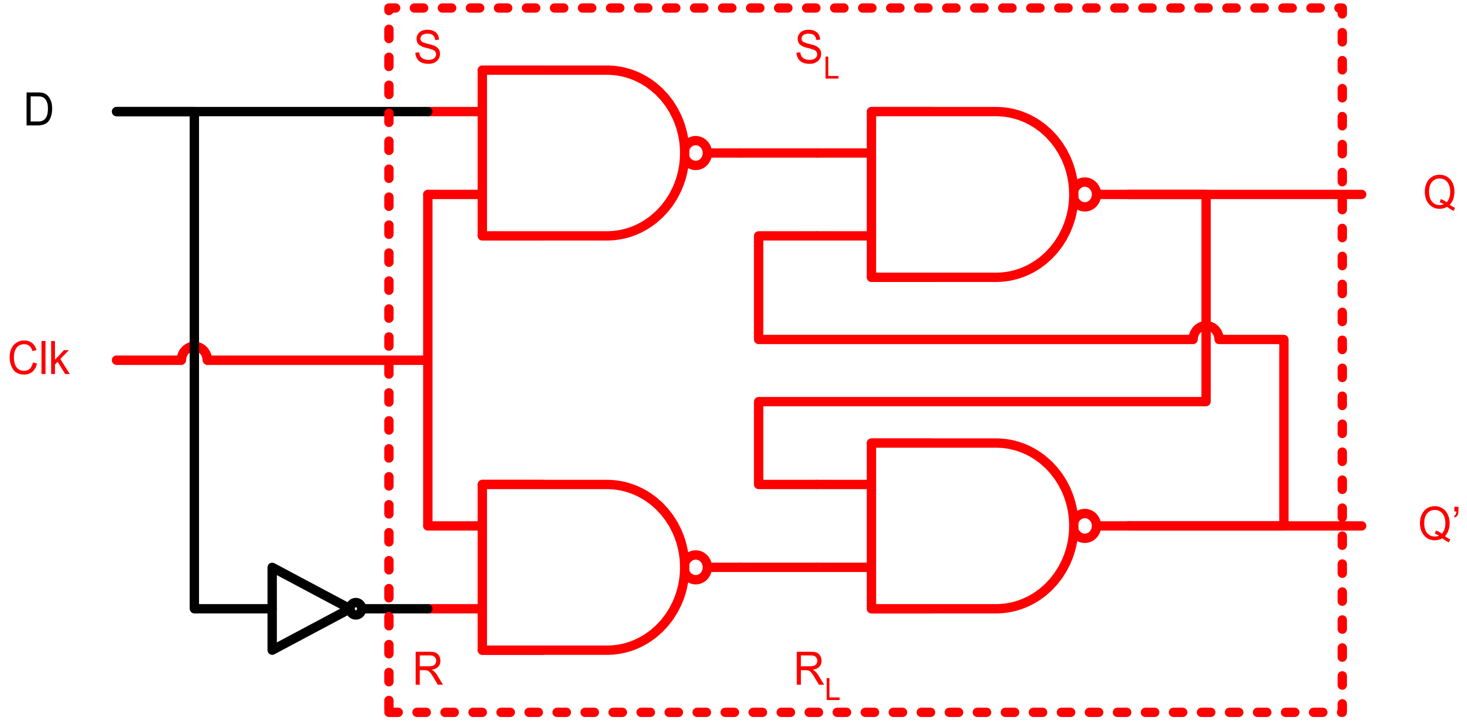
| Clk | D | Q(t+1) |
|---|---|---|
| 0 | x | Q(t) |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

# Gated D latch

- Our gated D latch is build from gated SR latch
- Mapping the truth table from gated D latch to Gated SR latch
  - Clk = 0, D = x => Clk = 0, S = x, R = x
  - Clk = 1, D = 0 => Clk = 1, S = 0, R = 1
  - Clk = 1, D = 1 => Clk = 1, S = 1, R = 0
- From the above mapping, we found that S = D and R = NOT D
  - We can build the circuit accordingly

# Gated D Latch

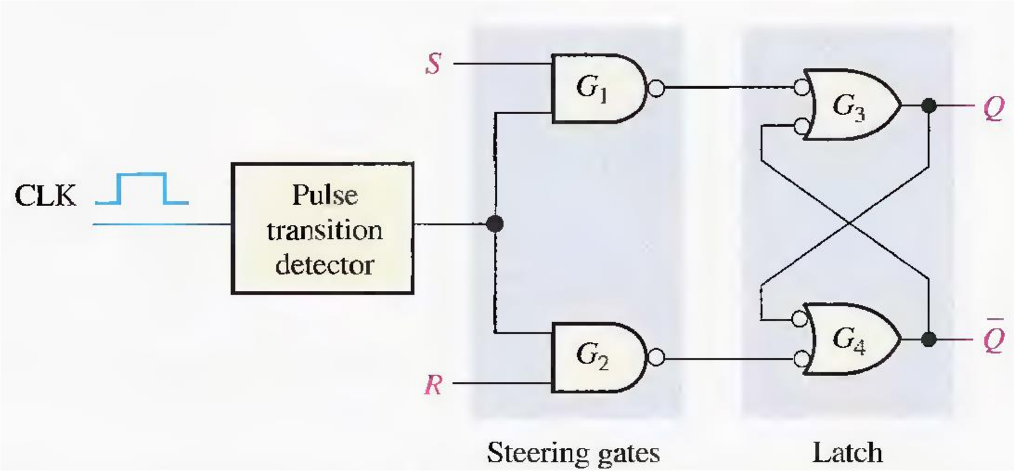
- Red color part is gated SR latch
- Noted that it is level sensitive



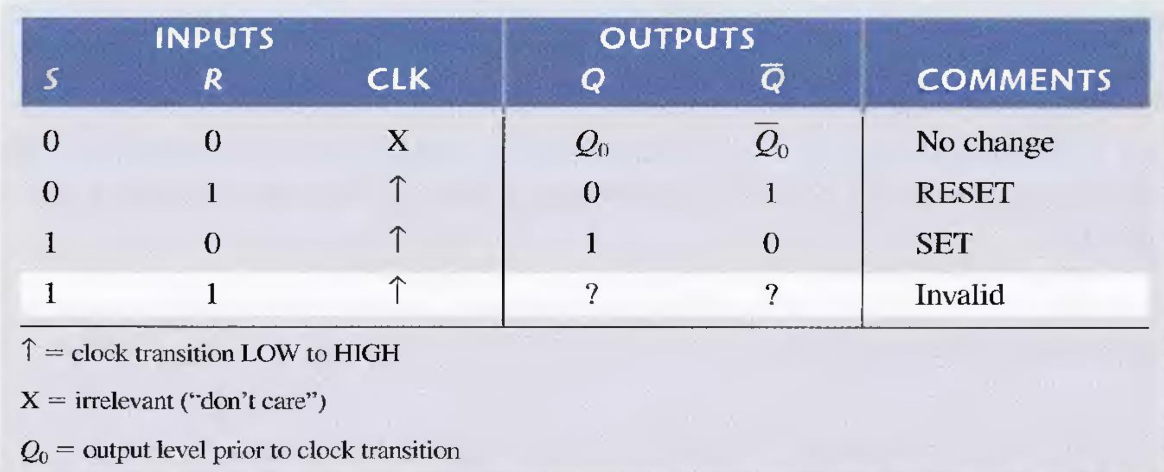
# Flip-Flop

- Flip-flop is a storage device that response to change from input exactly at the edge of clock signal
- The output will be stabilized within clock cycle
- Clock is an important input for electronic device
  - Circuit do not have concept of date and time, it count how many clock cycle pass
  - Clock signal need to accurate in the period

# Edge-Triggered Flip-Flops



The Edge-Triggered S-R Flip-Flop



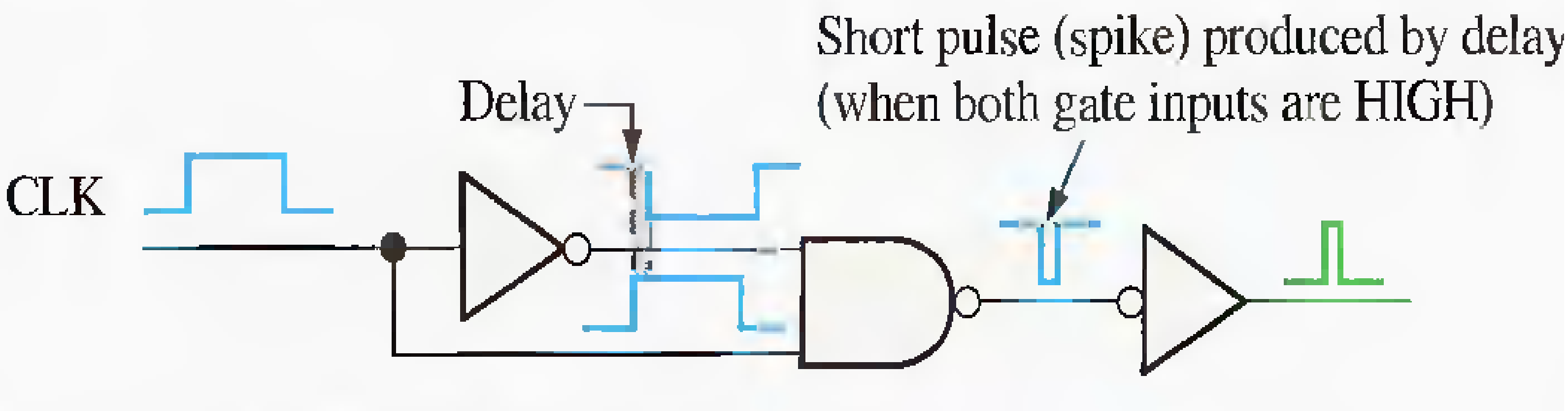
| INPUTS | | | OUTPUTS | | |
|---|---|---|---|---|---|
| *S* | *R* | CLK | *Q* | $\overline{Q}$ | COMMENTS |
| 0 | 0 | X | $Q_0$ | $\overline{Q}_0$ | No change |
| 0 | 1 | ↑ | 0 | 1 | RESET |
| 1 | 0 | ↑ | 1 | 0 | SET |
| 1 | 1 | ↑ | ? | ? | Invalid |

↑ = clock transition LOW to HIGH

X = irrelevant ("don't care")

$Q_0$ = output level prior to clock transition

# Pulse Transition Detector


Short pulse (spike) produced by delay
(when both gate inputs are HIGH)
Delay
CLK

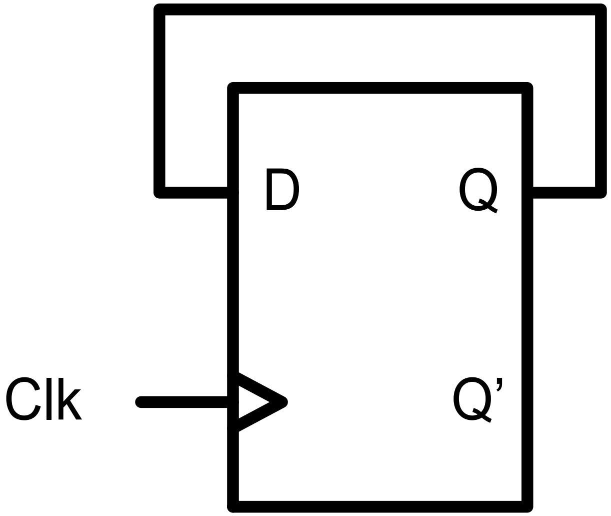
b) A type of pulse transition detector

# T Flip-Flop

- Truth Table for T FF

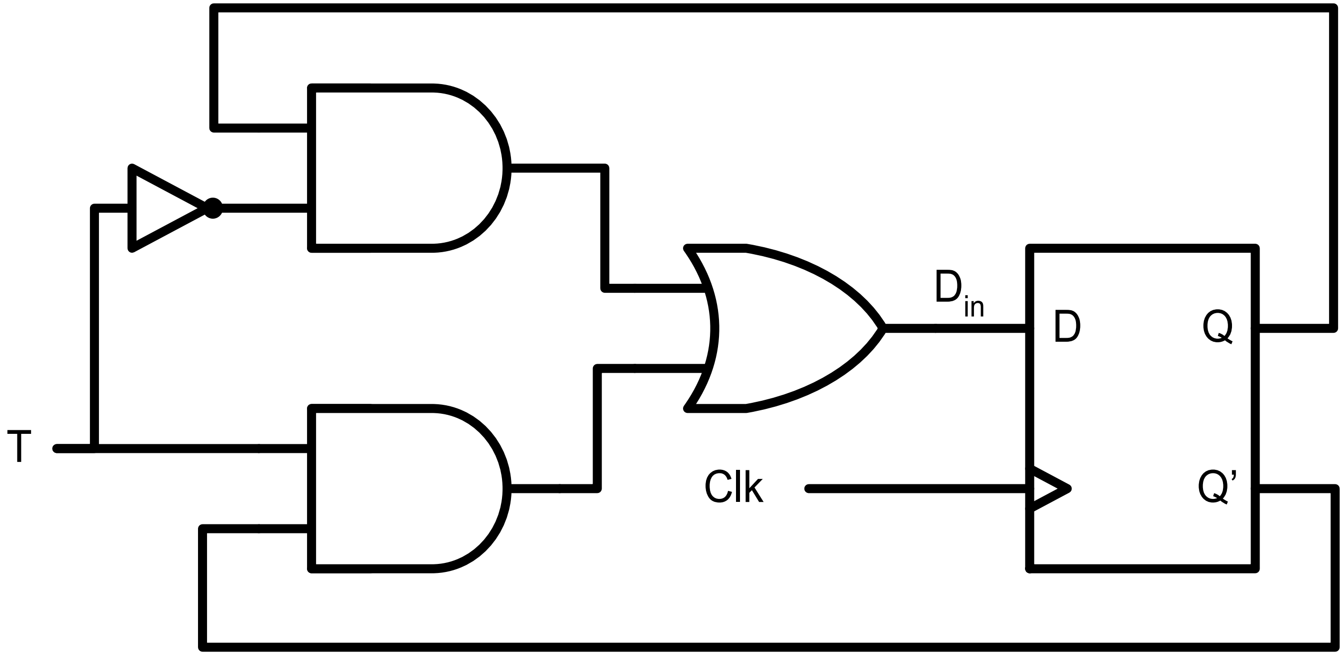
| T | Q(t+1) |
|---|---|
| 0 | Q(t) |
| 1 | Q(t)’ |

- We build T FF from D FF
- The circuit on the right do the following thing
  - Q(t+1) = Q(t)

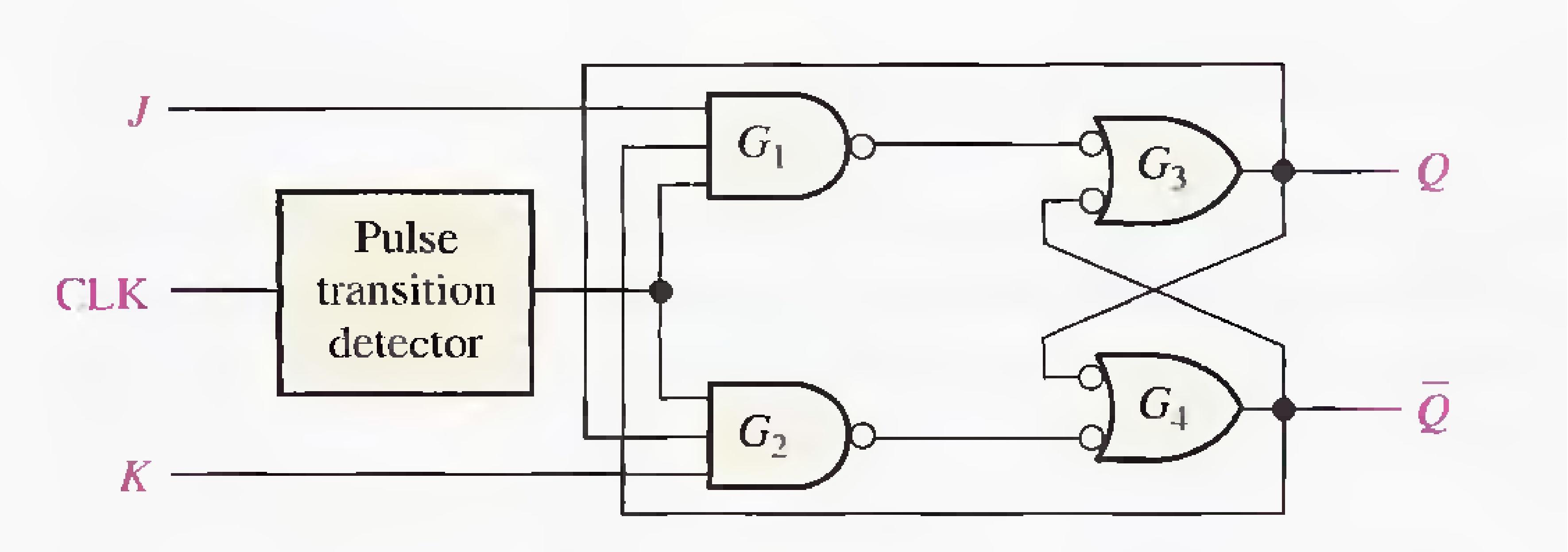


# T Flip-Flop

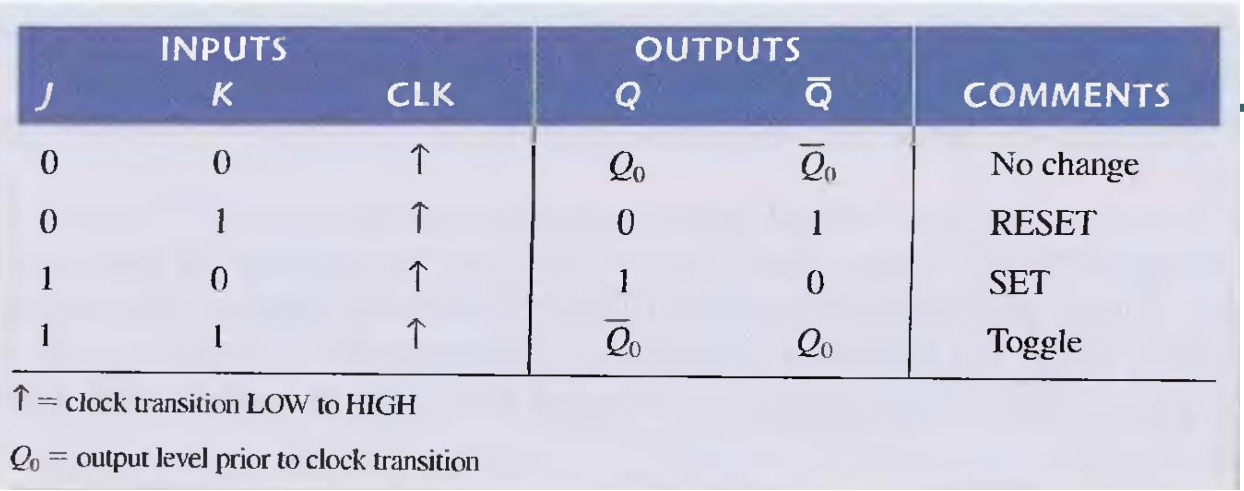
- Base on the circuit on last slide
  - Din = T'Q + TQ'
  - Q(t+1) = T'Q(t) + TQ'(t)



# JK Flip-Flop



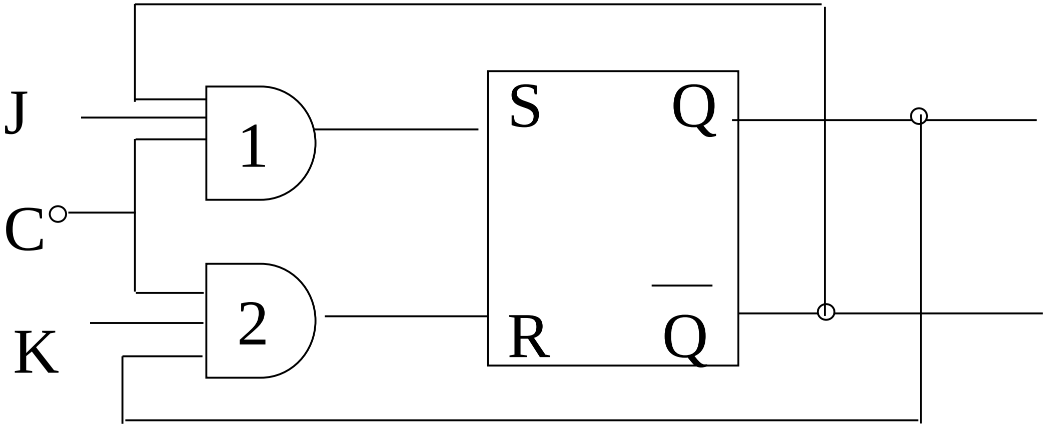
# JK Flip-Flop by using SR Flip-Flop



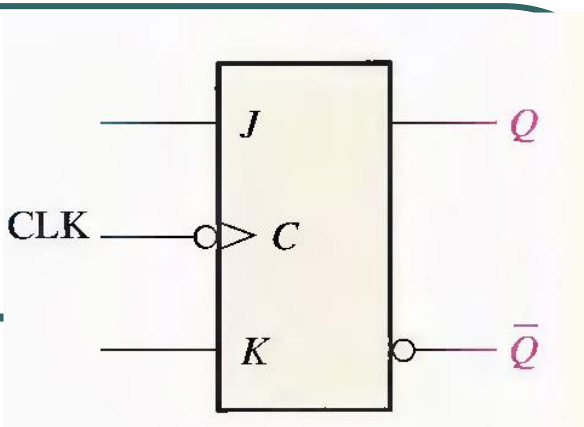
| INPUTS | | | OUTPUTS | | |
|---|---|---|---|---|---|
| *J* | *K* | CLK | *Q* | $\overline{Q}$ | COMMENTS |
| 0 | 0 | ↑ | $Q_0$ | $\overline{Q}_0$ | No change |
| 0 | 1 | ↑ | 0 | 1 | RESET |
| 1 | 0 | ↑ | 1 | 0 | SET |
| 1 | 1 | ↑ | $\overline{Q}_0$ | $Q_0$ | Toggle |

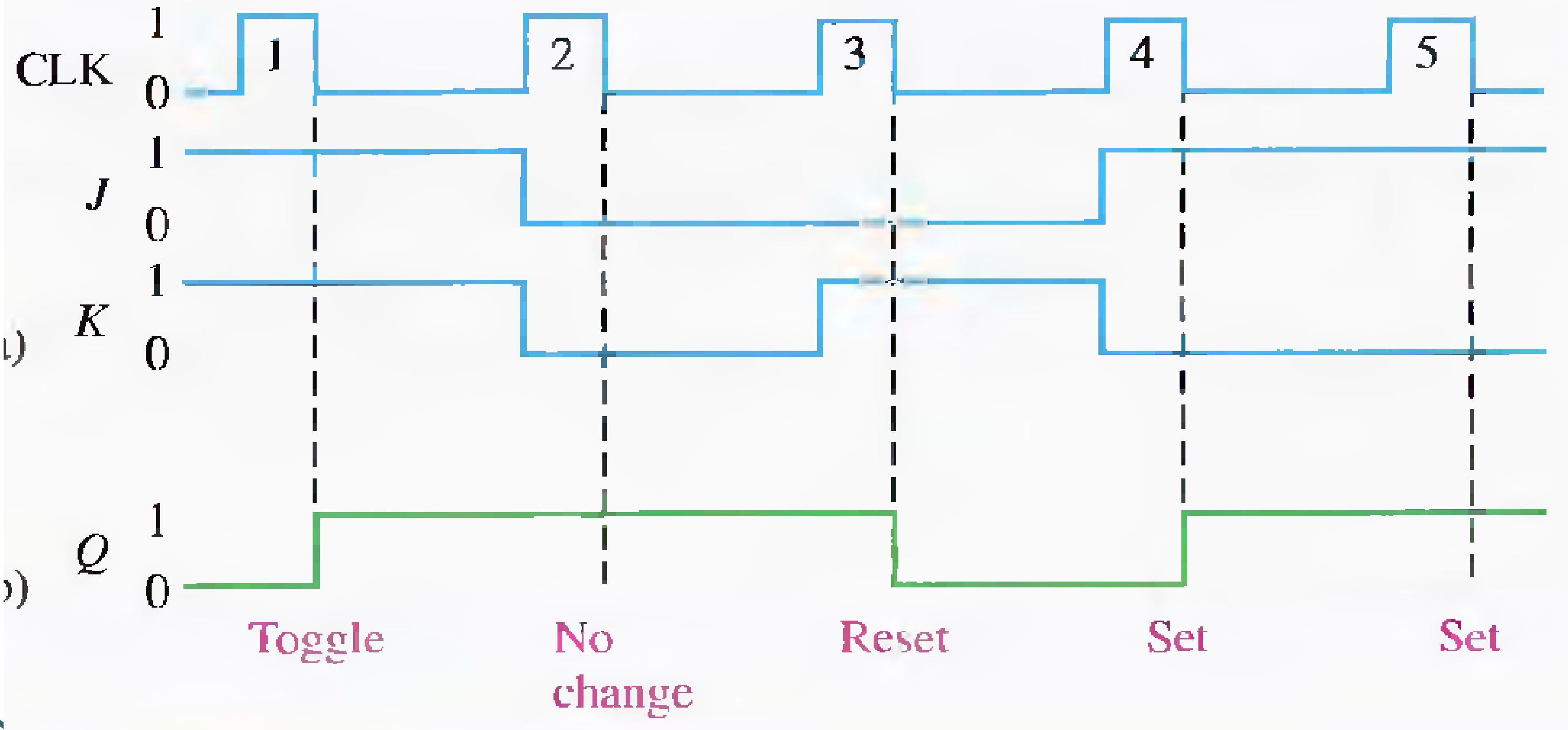
↑ = clock transition LOW to HIGH

$Q_0$ = output level prior to clock transition

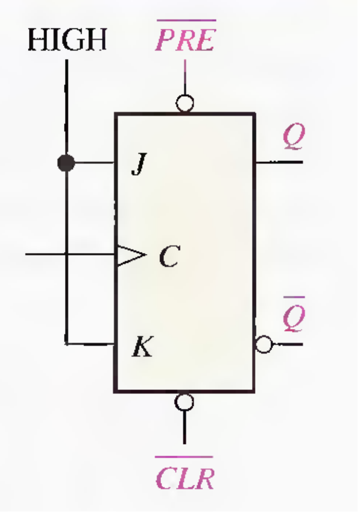


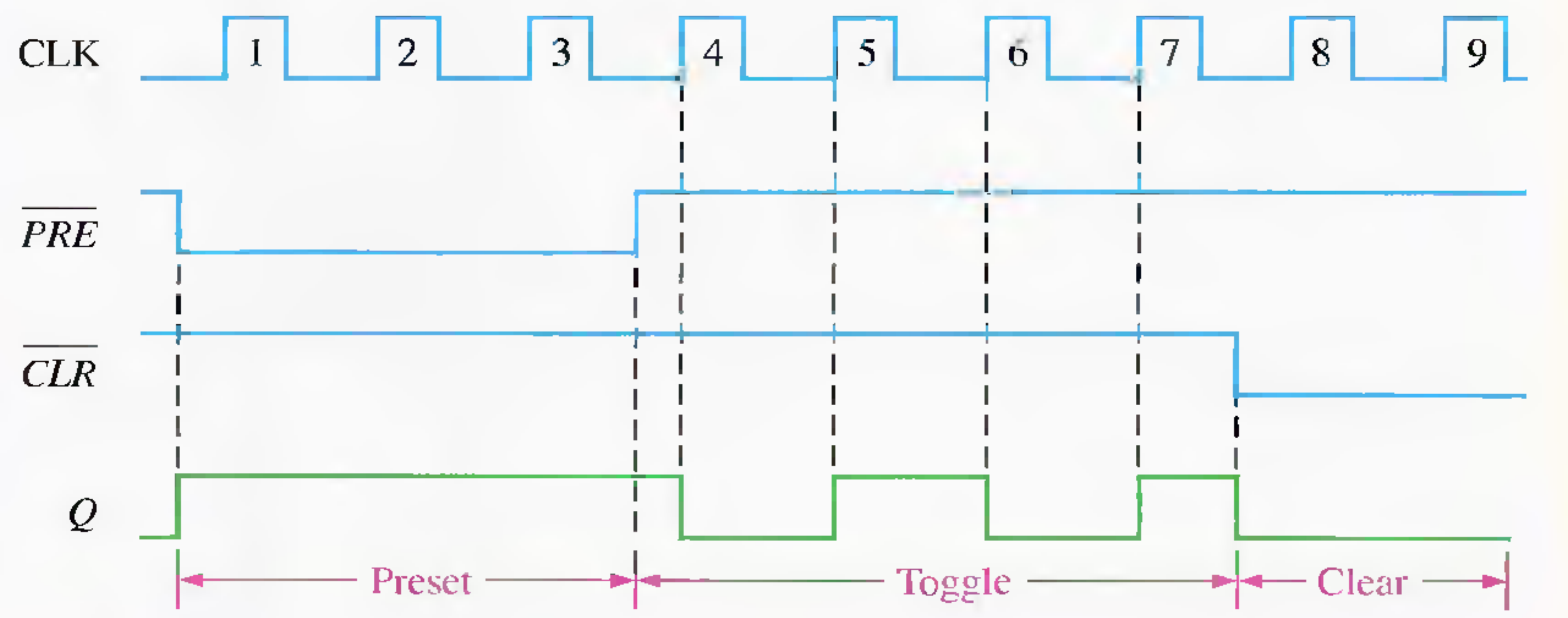
# Example


J
CLK
C
K
Q
$\overline{Q}$


CLK
1
0
1
2
3
4
5
J
1
0
K
1
0
a)
Q
1
0
b)
Toggle
No change
Reset
Set
Set

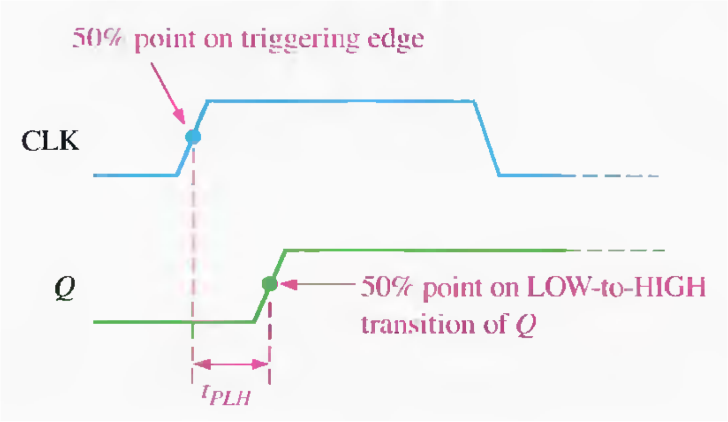
# Asynchronous Preset and Clear Inputs



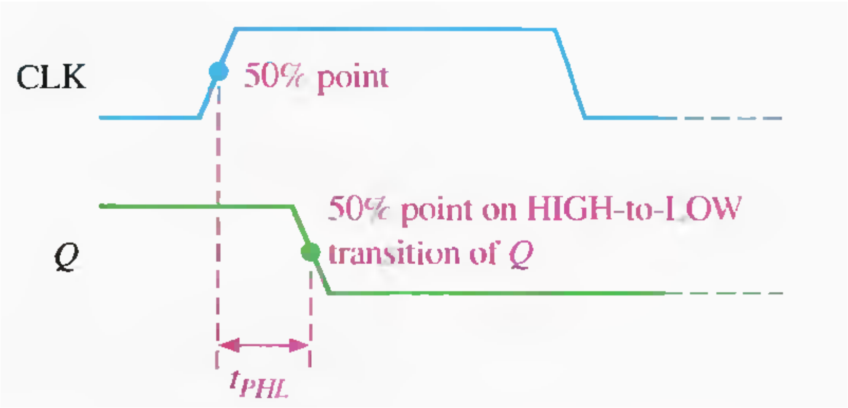


# Propagation Delay Times ( tp )

- the interval of time required after an input signal has been applied for the resulting output change to occur.

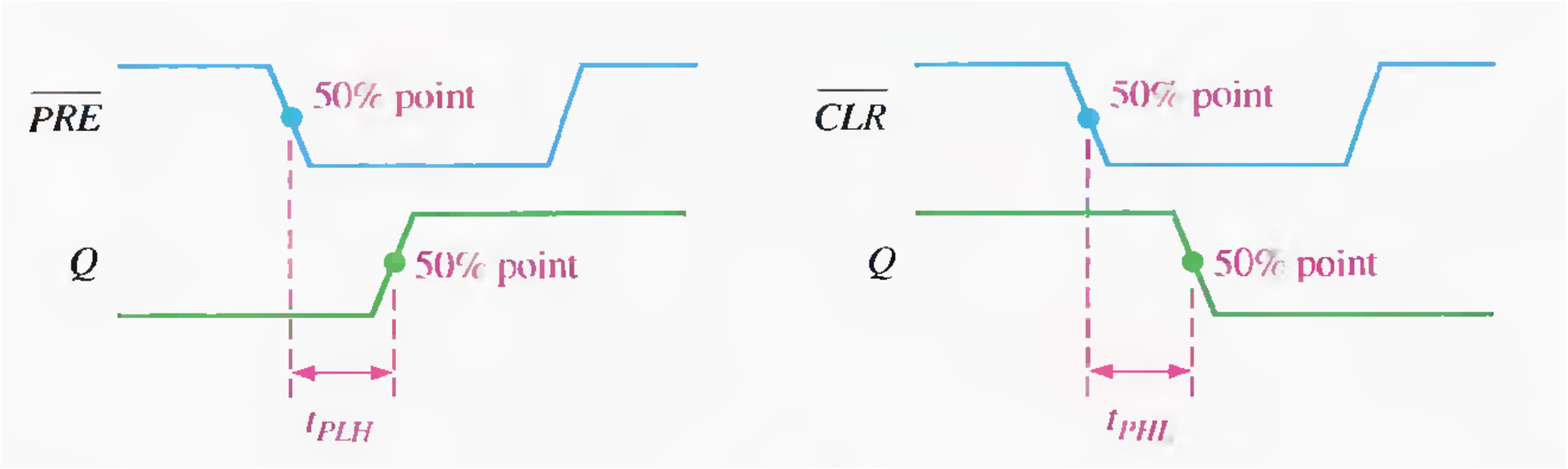

50% point on triggering edge
CLK
Q
50% point on LOW-to-HIGH transition of Q
$t_{PLH}$

(a)



(b)

# Propagation Delay Times …

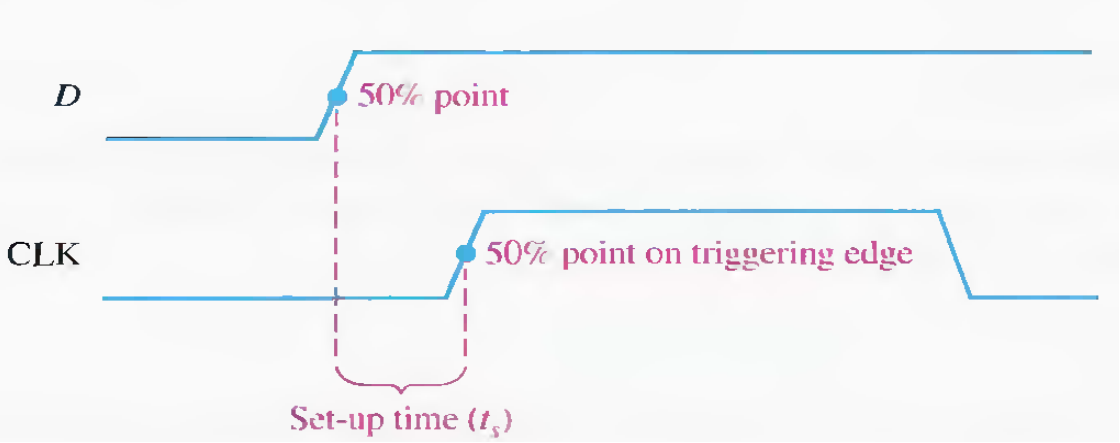


(c)

(d)

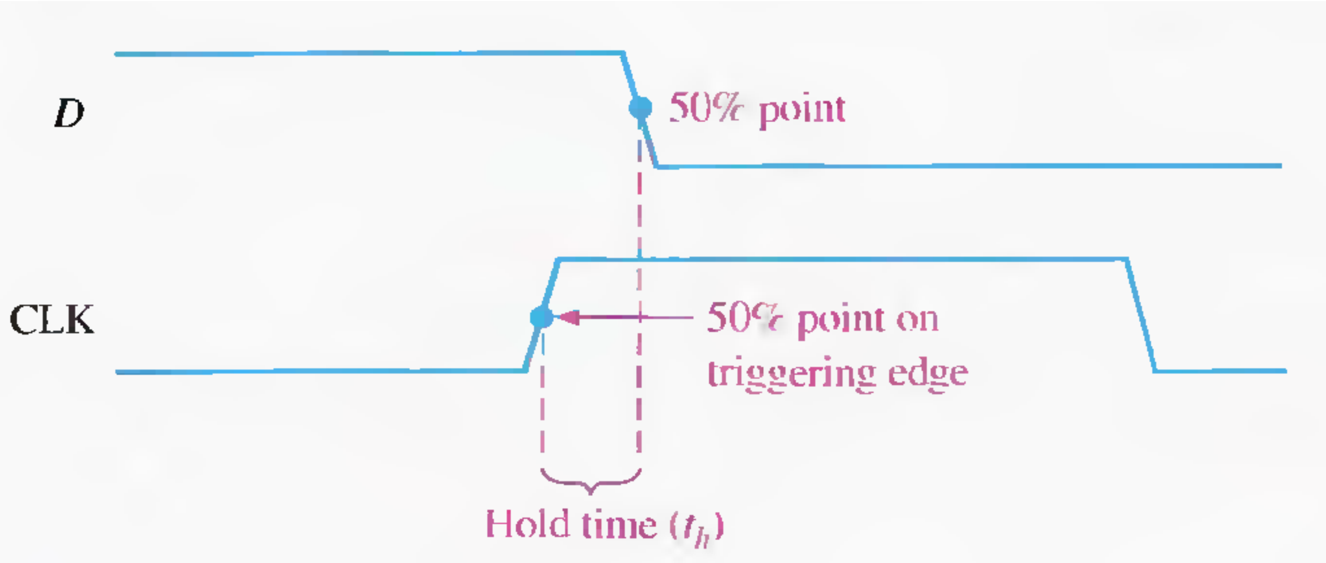
# Set-up Time ( $t_s$ )

- the minimum interval required for the logic levels to be maintained constantly on the inputs (J and K, or Sand R, or D) prior to the triggering edge of the clock  pulse in order for the levels to be reliably clocked into the flip-flop.

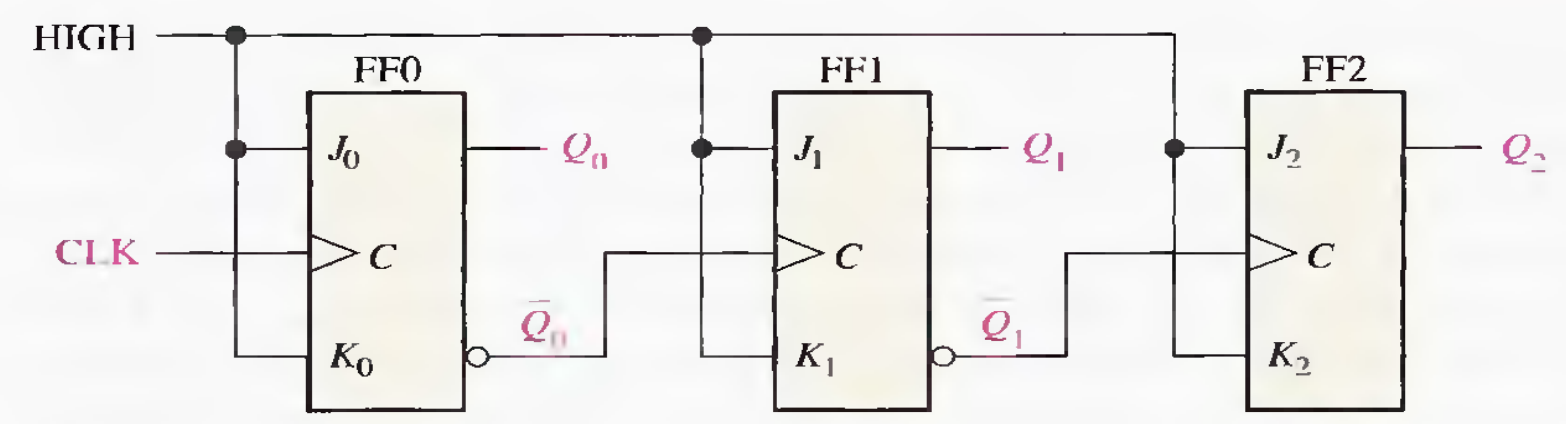


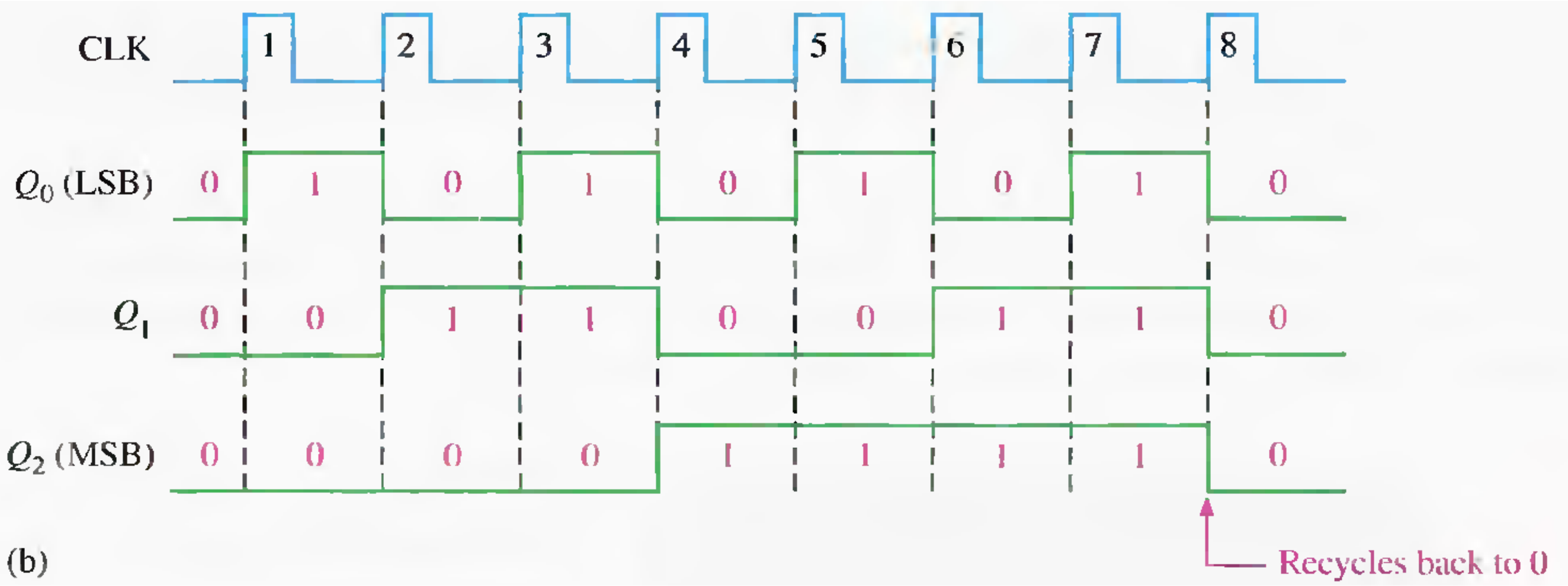
# **hold time** ( $t_h$ )

- the minimum interval required for the logic levels to remain on the inputs after the triggering edge of the clock pulse in order for the levels to be reliably clocked  into the flip-flop.



# Asynchronous Counter

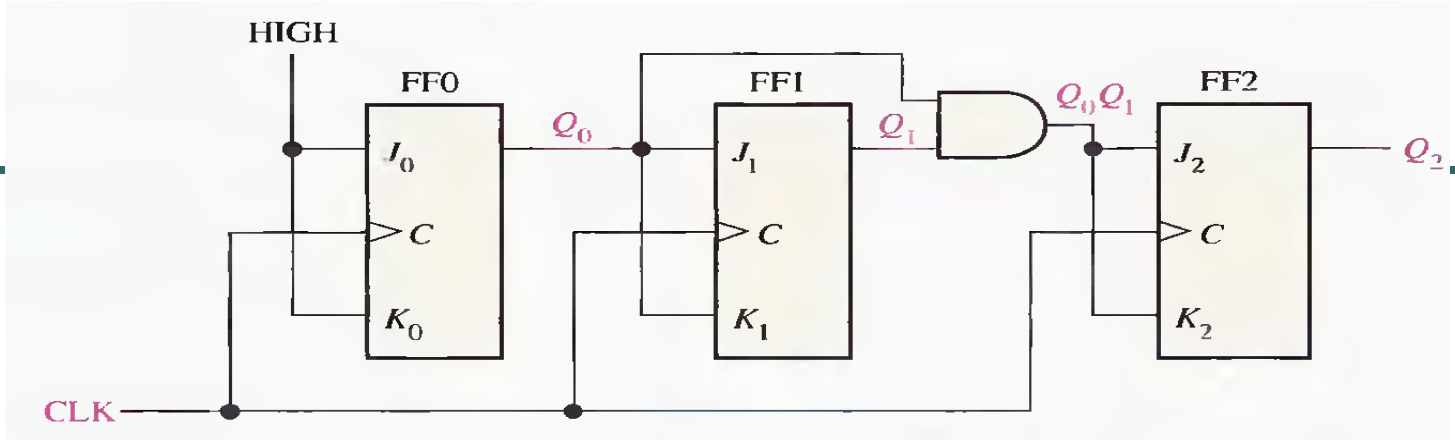


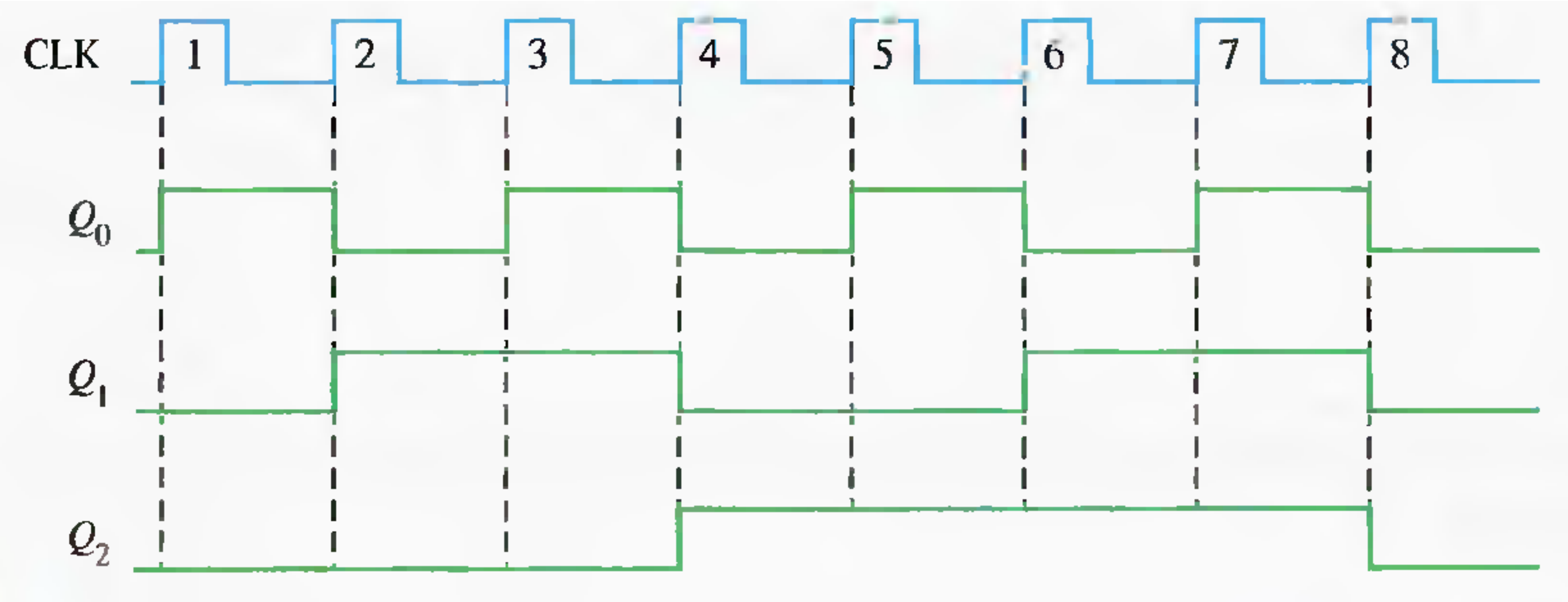


(b)

Recycles back to 0

# Synchronous Counter





# Difference between Asynchronous and Synchronous Counter

| Asynchronous Counter | Synchronous Counter |
|---|---|
| 1. Clock input is applied to LSB FF. The output of first FF is connected as clock to next FF. | 1. Clock input is common to all FF. |
| 2. All Flip-Flops are toggle FF. | 2. Any FF can be used. |
| 3. Speed depends on no. of FF used for n bit . | 3. Speed is independent of no. of FF used. |
| 4. No extra Logic Gates are required. Cost is less. | 4. Logic Gates are required based on design. Cost is more. |