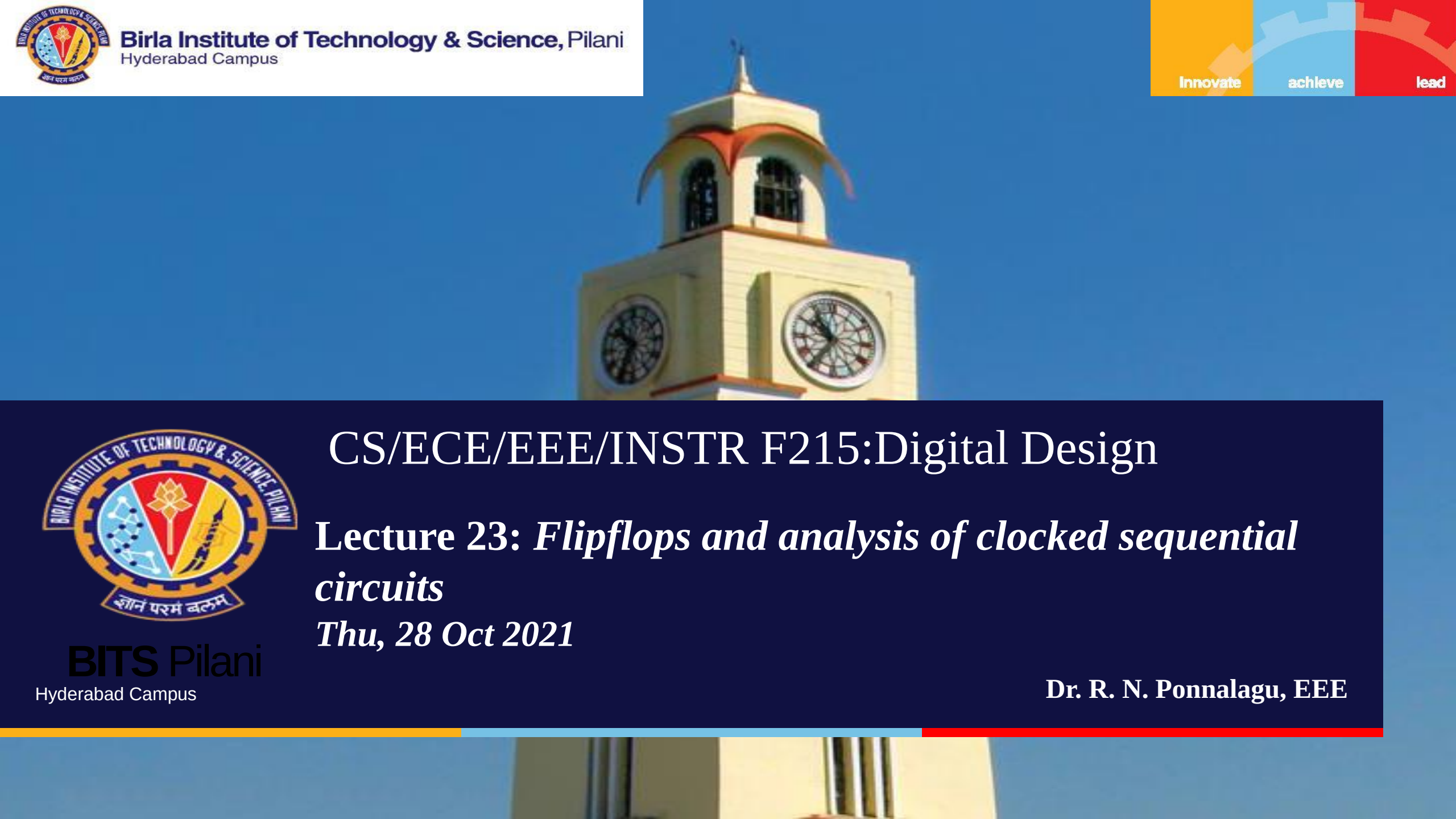


CS/ECE/EEE/INSTR F215:Digital Design

Lecture 23: *Flipflops and analysis of clocked sequential circuits*

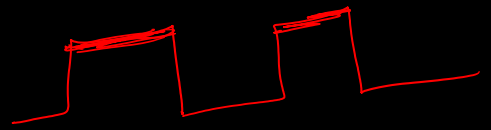
Thu, 28 Oct 2021

BITS Pilani
Hyderabad Campus

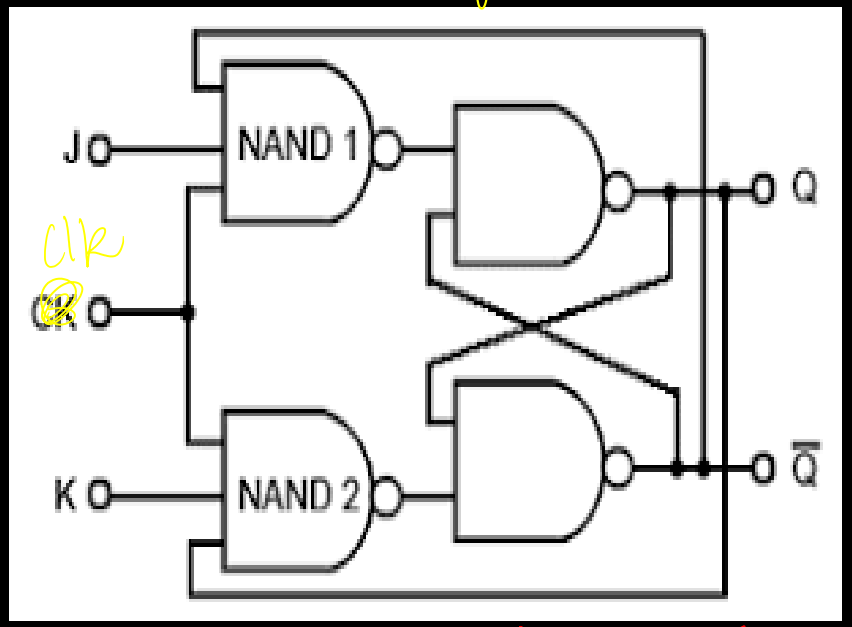
Dr. R. N. Ponnalagu, EEE

The strongest factor for success is
Self-esteem:
Believing you can do it,
Believing You deserve it and
Believing you will get it

- Latches are Level Sensitive (level triggered)
- When the control/clk input is 1, latches change state as and when the input applied to latches changes.
- Unreliable operation. In JK latch **race around condition** occurs
- Synchronization is not possible when output of latch is applied directly or through combinational logic circuit to input of the same or another latch and all latches are controlled by same clock
- Solution: Modify latch to work as flip-flop
- Flip-flops trigger only during clock transition
- Positive transition (+ve edge: clock changes from 0 to 1)
- Negative transition(–ve edge: clock changes from 1 to 0)

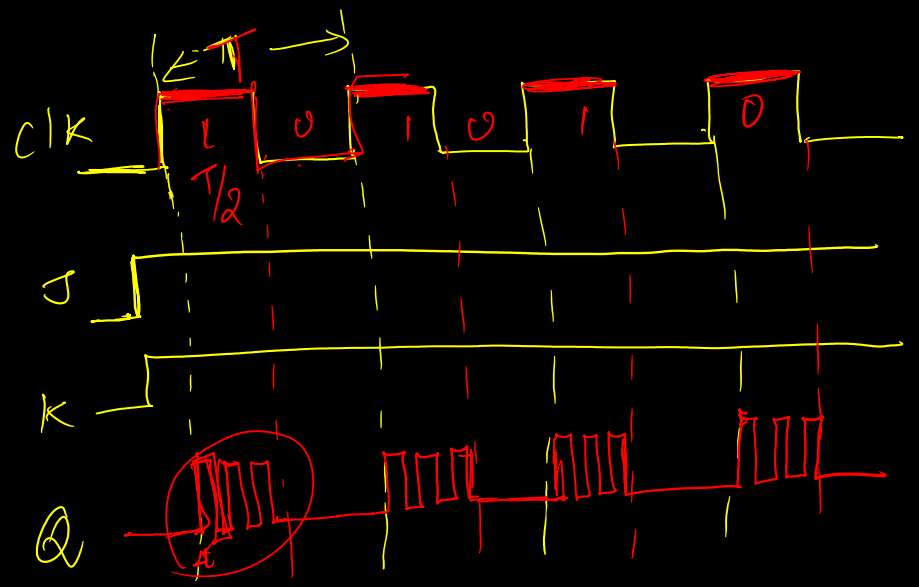


level triggered JK Latch



Characteristic table

J	K	Q_{t+1}
0	0	Q_t ⇒ NC No change
0	1	0 ⇒ reset
1	0	1 ⇒ set
1	1	Toggle ⇒

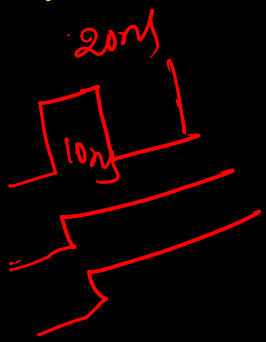


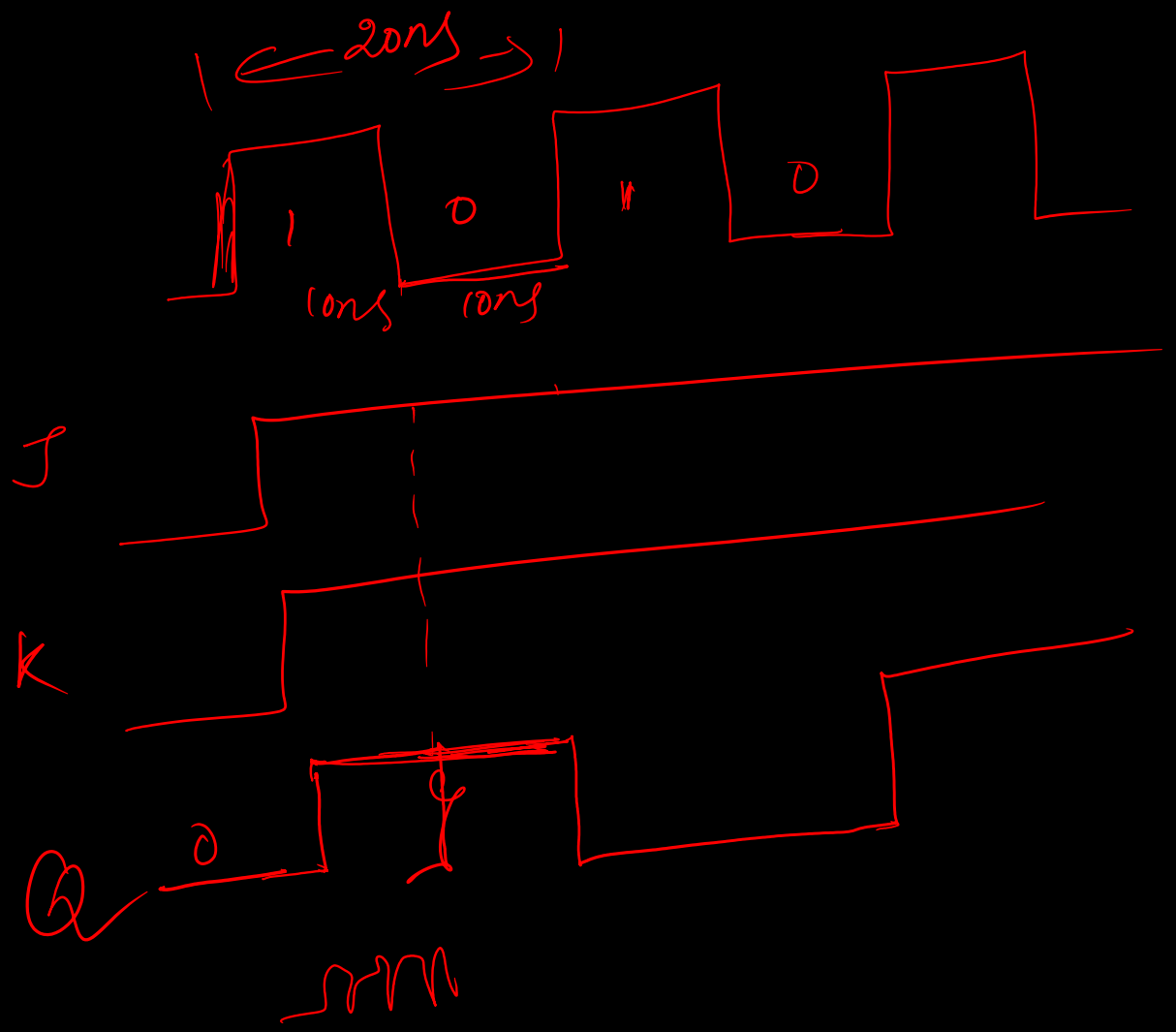
$T/2 = 5$ ns

Race around condition (or) Racing

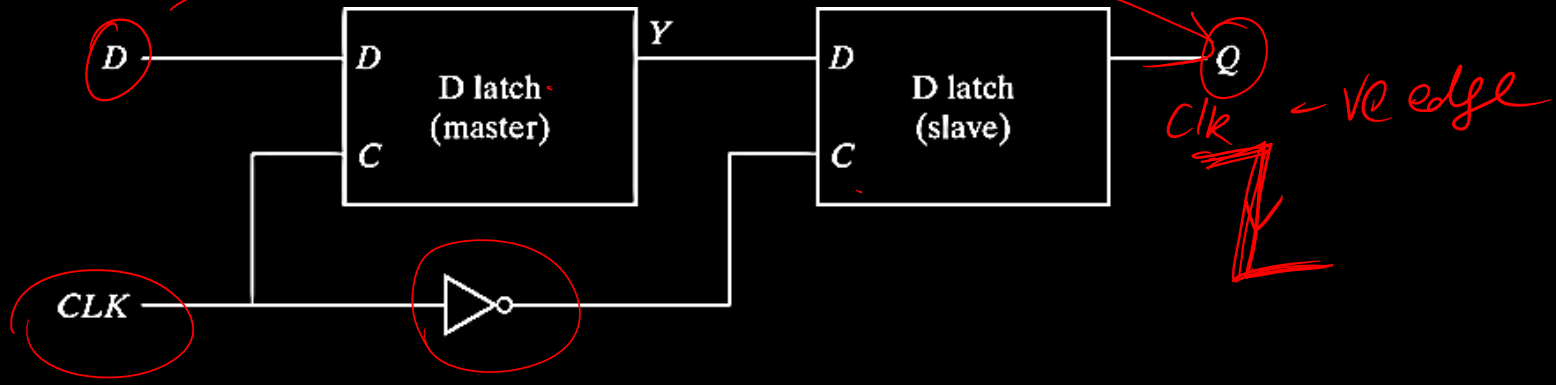
To prevent race around Condition

1) $T/2$ < Propagation delay of gate 10ns
2) Edge triggering instead of level triggering
3) master slave operation

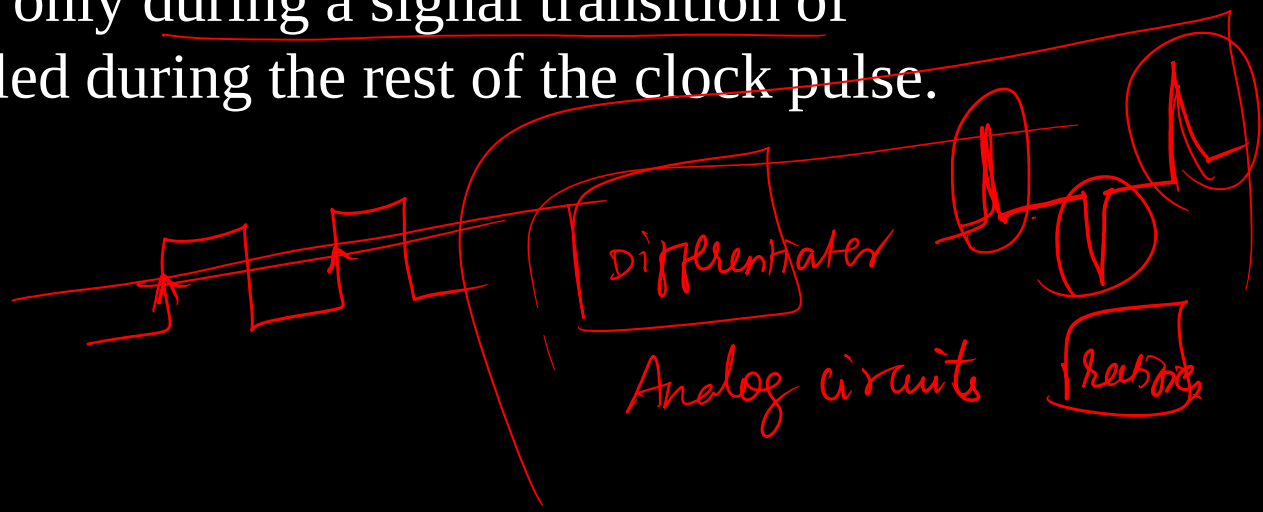



20ns
1
0
1
0
10ns
10ns
J
K
Q
0

- Modifying Latches to Flipflops
  - Two latches connected in a special configuration in such a way that output is isolated and prevented from being affected when the input to the flip flop is changing.
    - Master- Slave Configuration

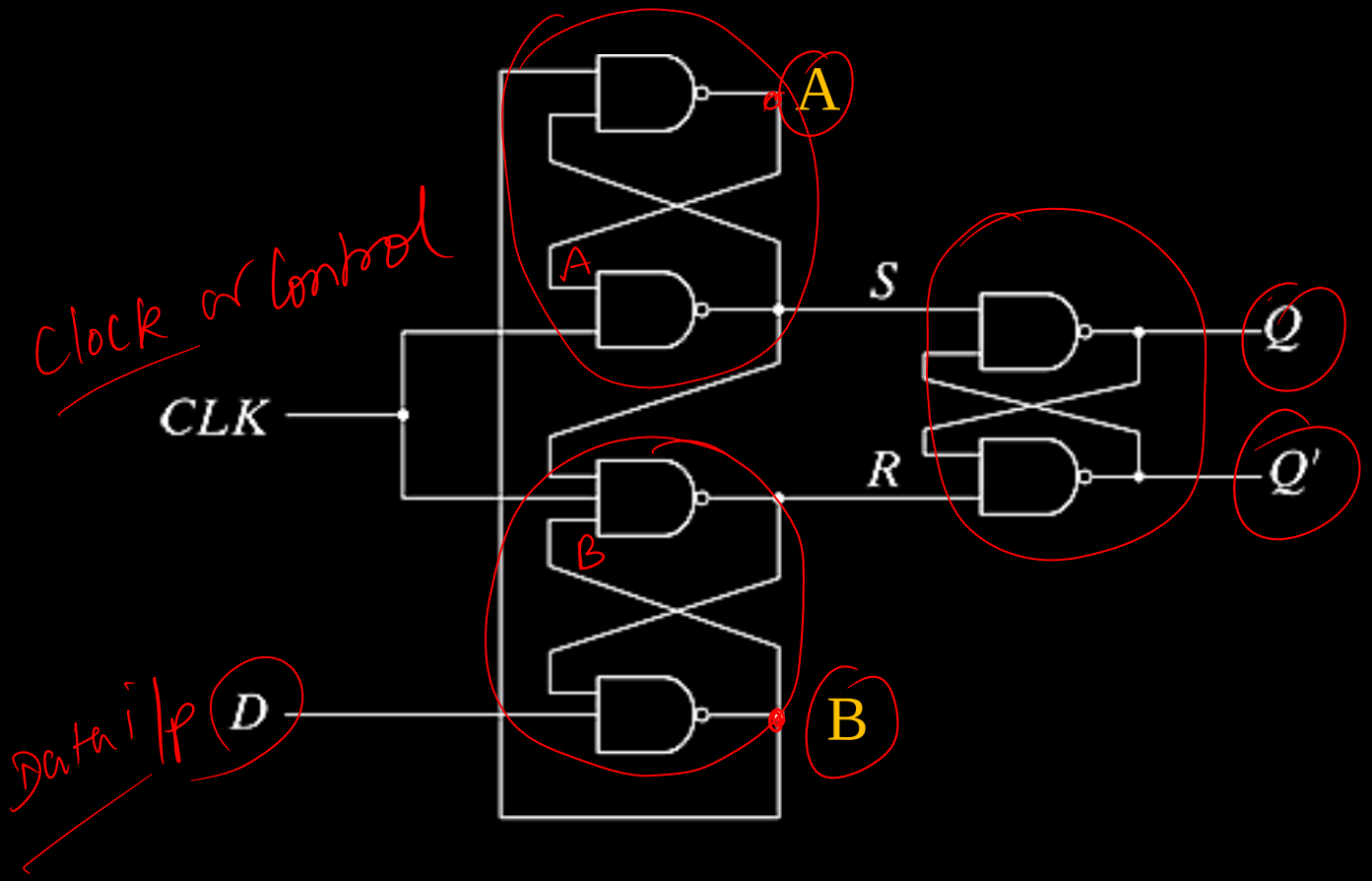


  - Gate based design (Latches) which trigger only during a signal transition of synchronizing (Clock) signal and is disabled during the rest of the clock pulse.



Edge Triggered D Flip-Flops – Another Construction

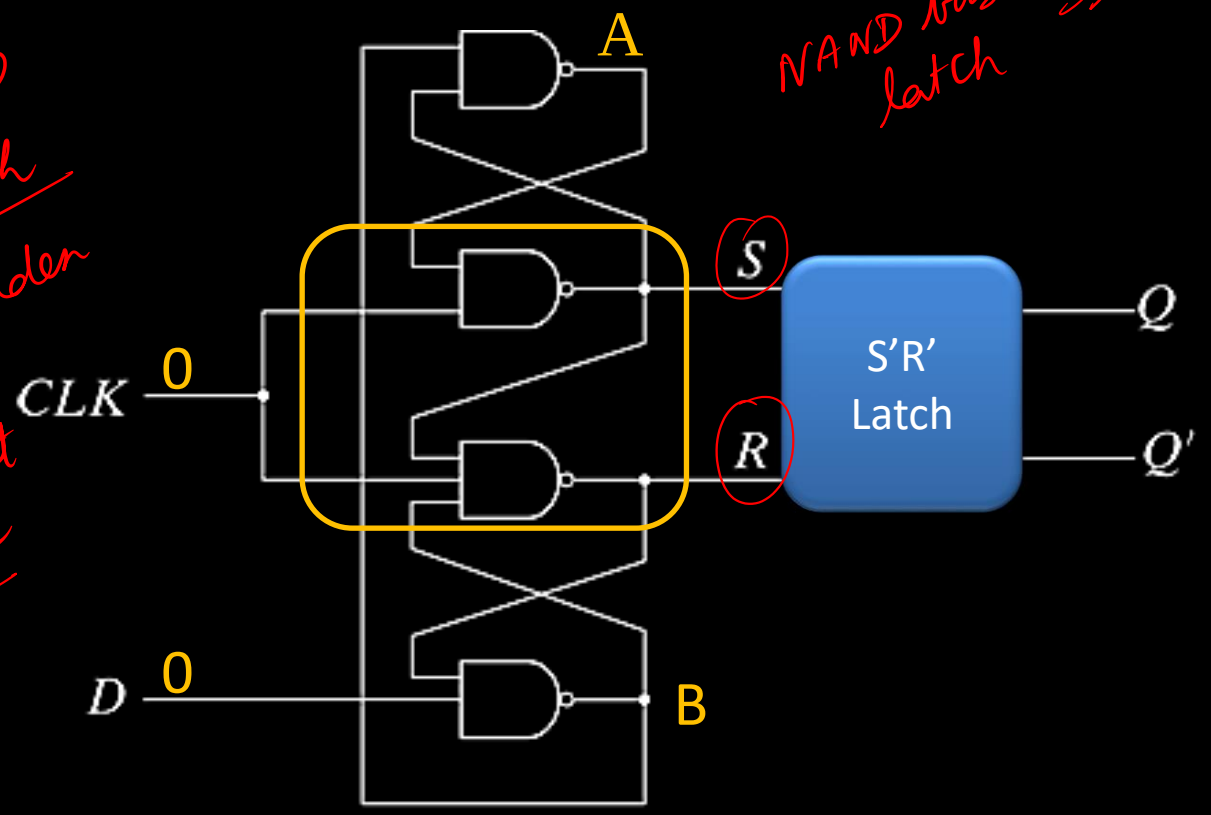
D-type positive edge triggered FF



Uses 3 latches out of which
Two latches respond to
external D (data) input
and clock (clk) inputs
Third latch provides the
output of the flip-flop

Edge Triggered D Flip-Flops

D-type positive edge triggered FF



Case I:

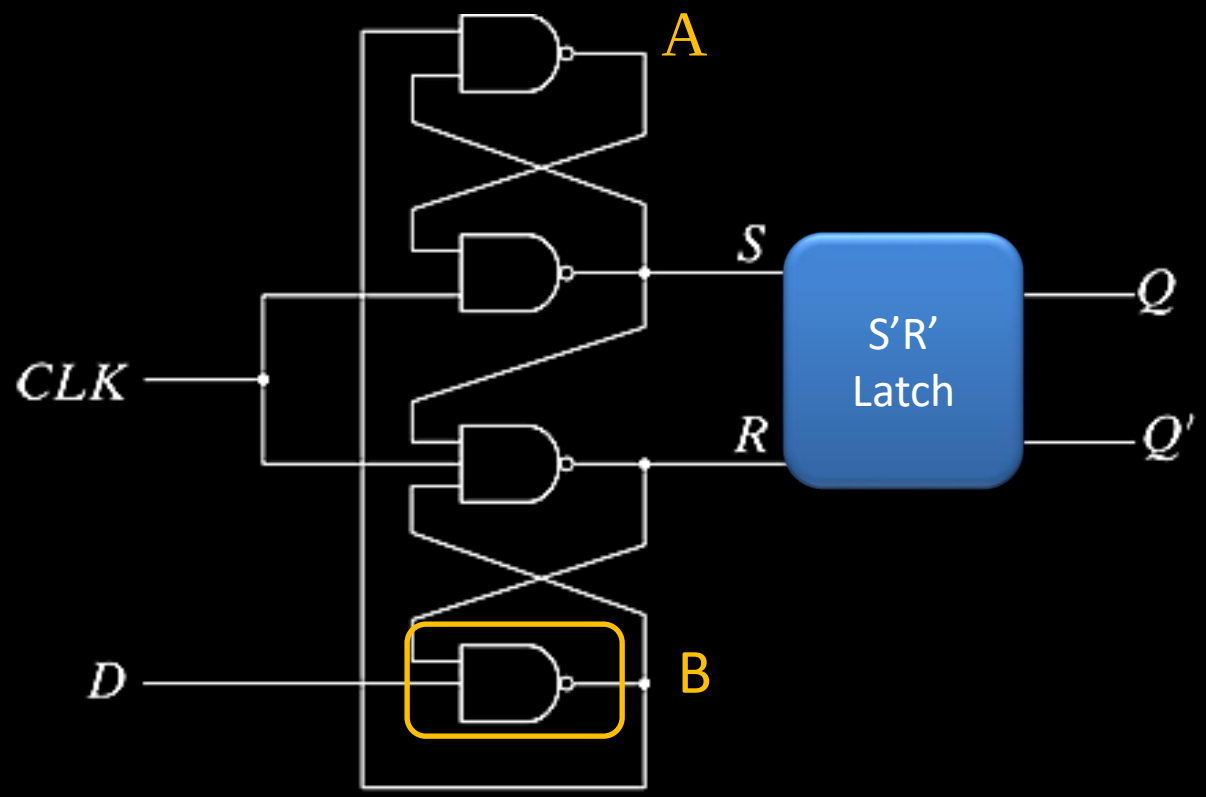
D = 0, CLK = 0

CLK = 0 => S=1, R=1

CKV

Edge Triggered D Flip-Flops

D-type positive edge triggered FF



Case I:

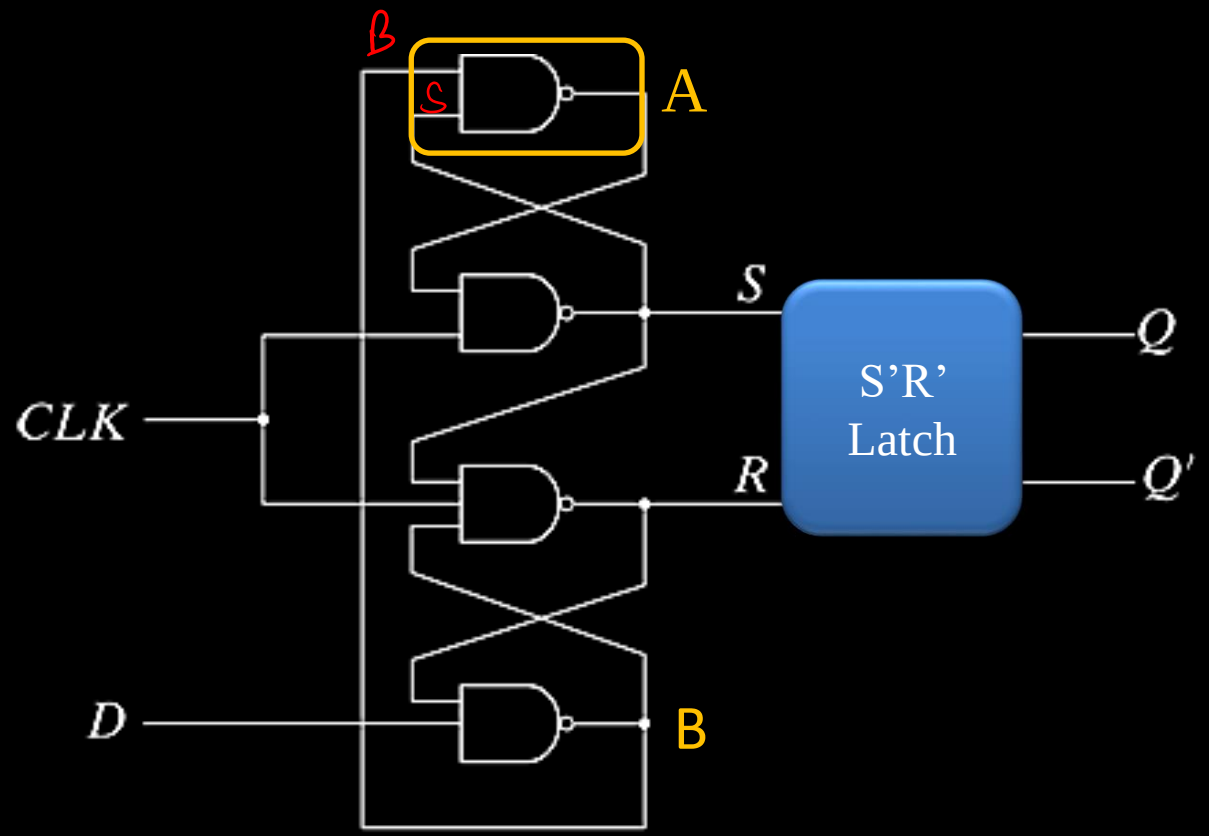
D = 0, CLK = 0

CLK = 0 => S=1, R=1

D = 0 => B=1

Edge Triggered D Flip-Flops

D-type positive edge triggered FF



Case I:

D = 0, CLK = 0

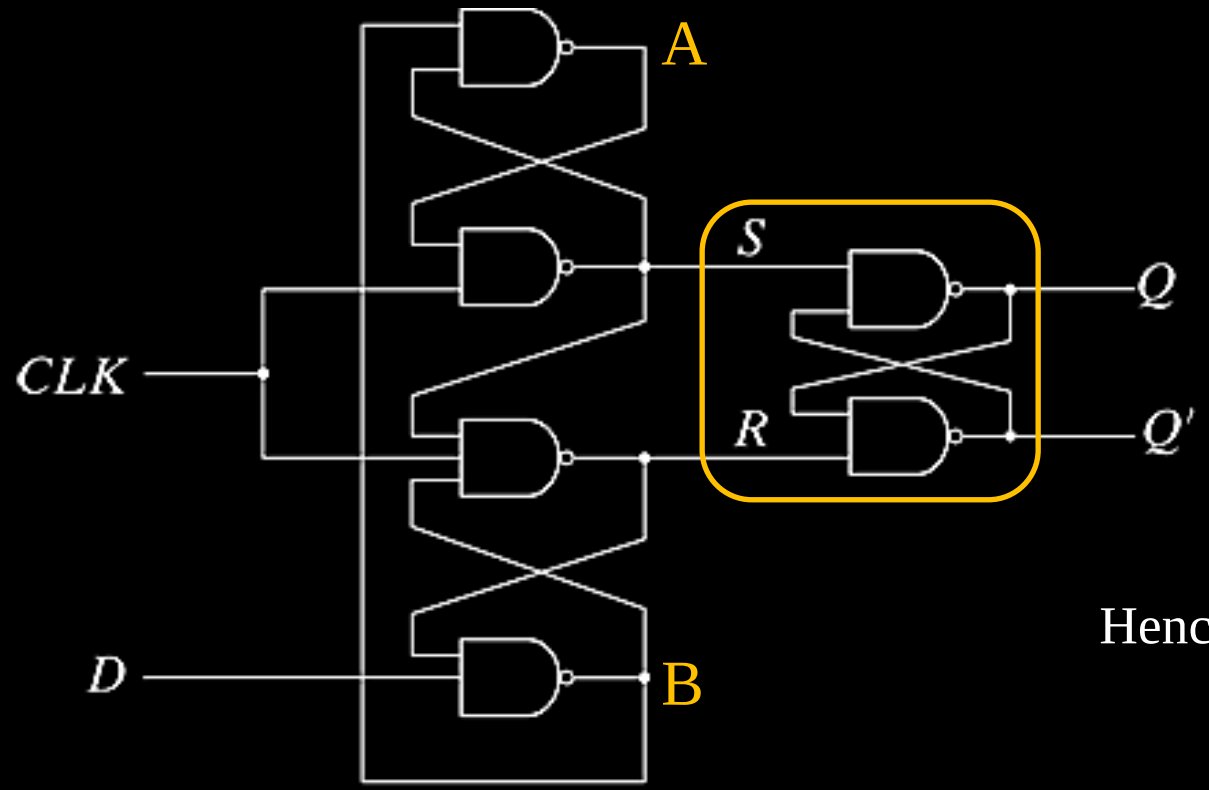
CLK = 0 => S=1, R=1

D = 0 => B=1

B=1, S=1 => A=0

Edge Triggered D Flip-Flops

D-type positive edge triggered FF



Case I:

D = 0, CLK = 0

CLK = 0 => S=1, R=1

D = 0 => B=1

B=1, S=1 => A=0

S = 1, R = 1 for S'R' latch

No change Q & Q̄

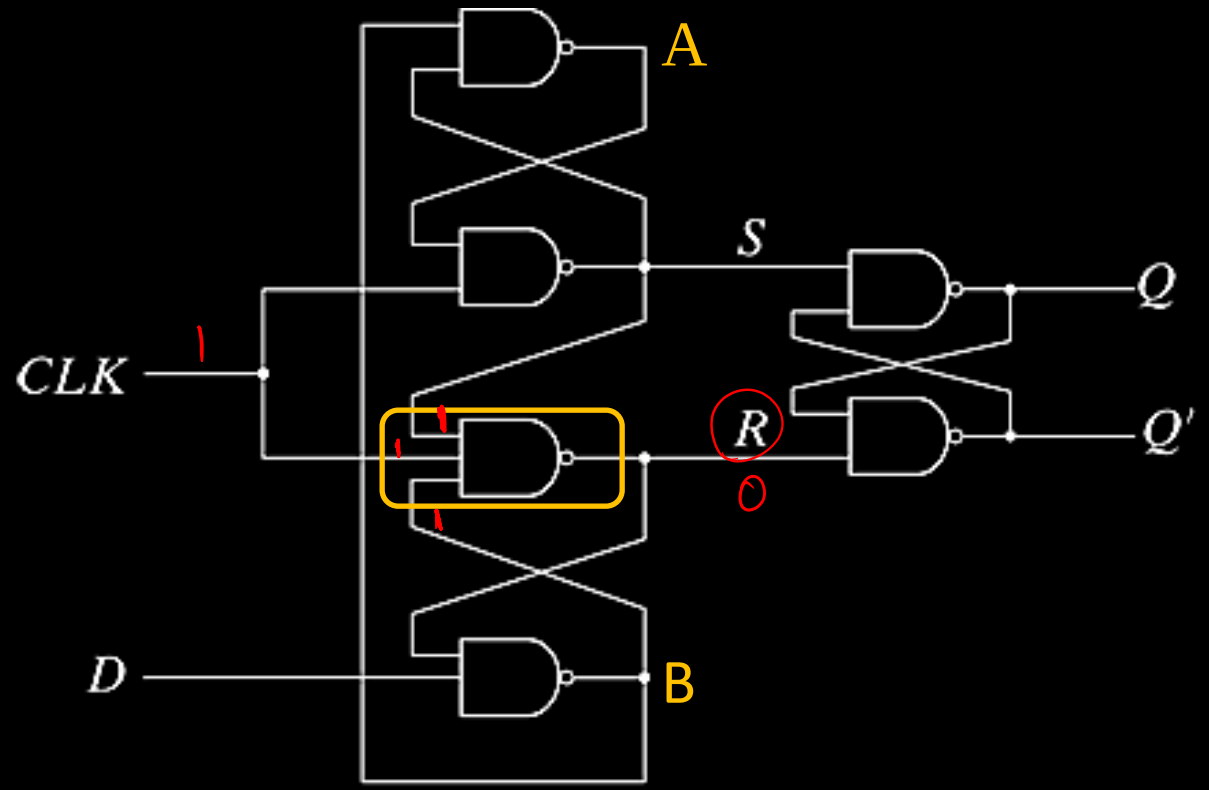
Hence Previous state is maintained

A = 0, S = 1, R = 1, B = 1

CKV

Edge Triggered D Flip-Flops

D-type positive edge triggered FF



In case I: make CLK = 1

i.e D=0, A = 0, S = 1, R = 1, B = 1

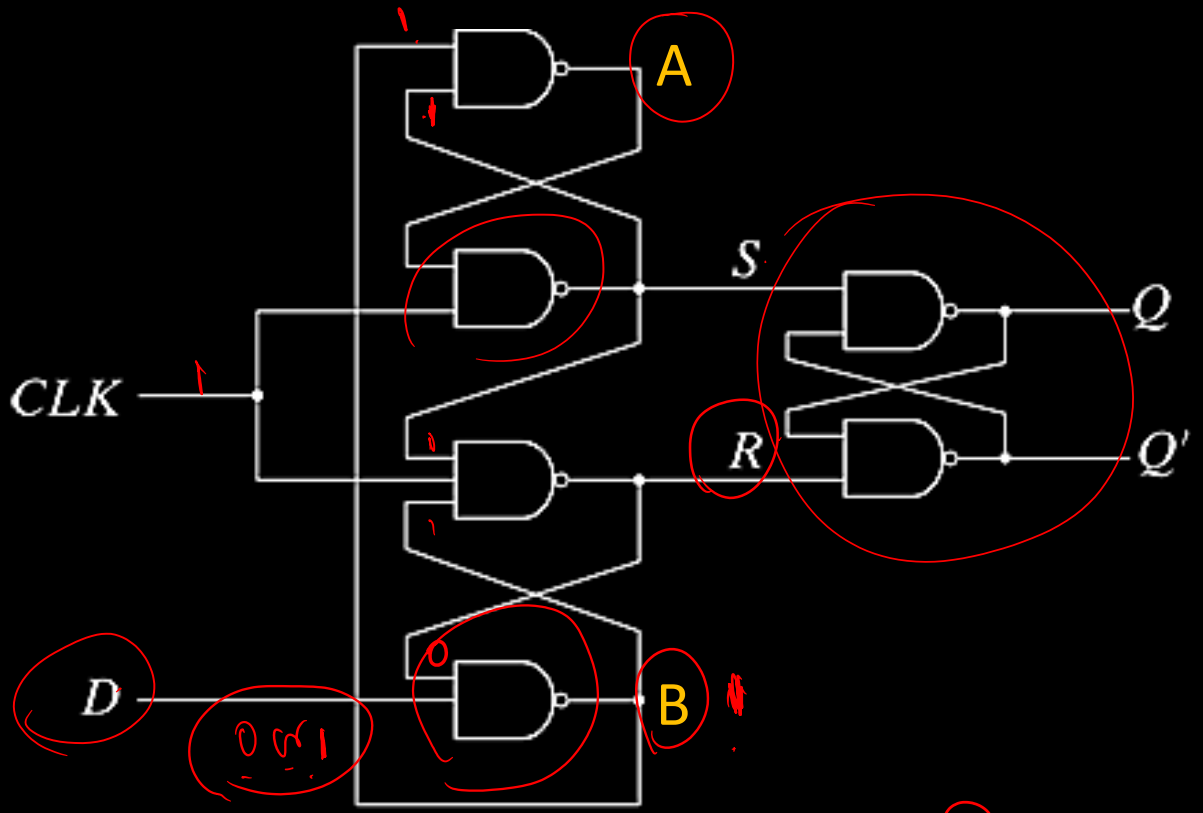
When CLK = 1, S = 1, B=1 => R = 0

D=0

CKV

Edge Triggered D Flip-Flops

D-type positive edge triggered FF



A = 0, S = 1, R = 1, B = 1 D = 0, CLK = 1

CLK = 1, S = 1, B=1 => R = 0

D = 0, R = 0 => B = 1

B = 1, S = 1 => A = 0

Only R changes to 0 ⇒ R Changes here after one gate delay

A = 0, S = 1, R = 0, B = 1

S = 1, R = 0 for S'R' latch

Hence Q = 0 and Q' =1 Reset state

If D is changed now, keeping CLK =1??

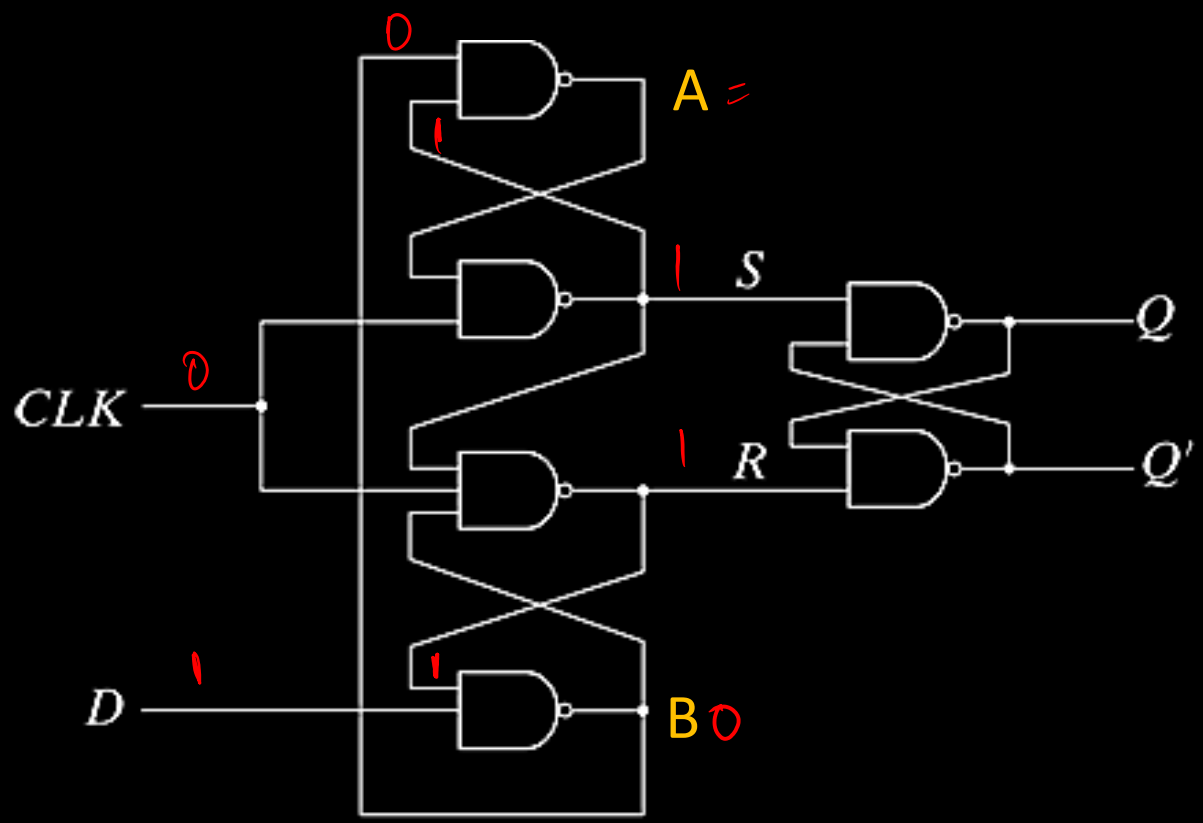
If D changes after R has stabilized there will be no effect on output.

This is Hold Time.

CKV

Edge Triggered D Flip-Flops

D-type positive edge triggered FF

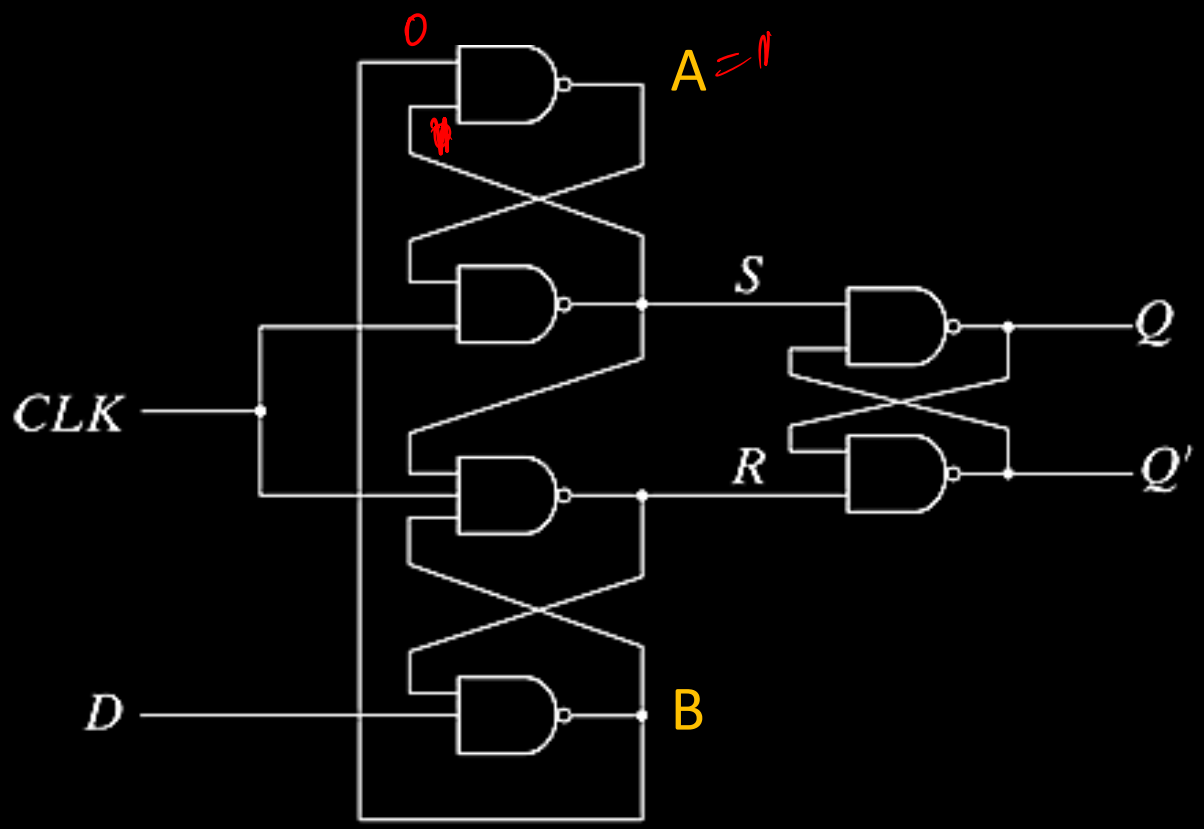


Case II:

- **D = 1, CLK = 0**
- **CLK = 0 => R =1, S = 1**

Edge Triggered D Flip-Flops

D-type positive edge triggered FF

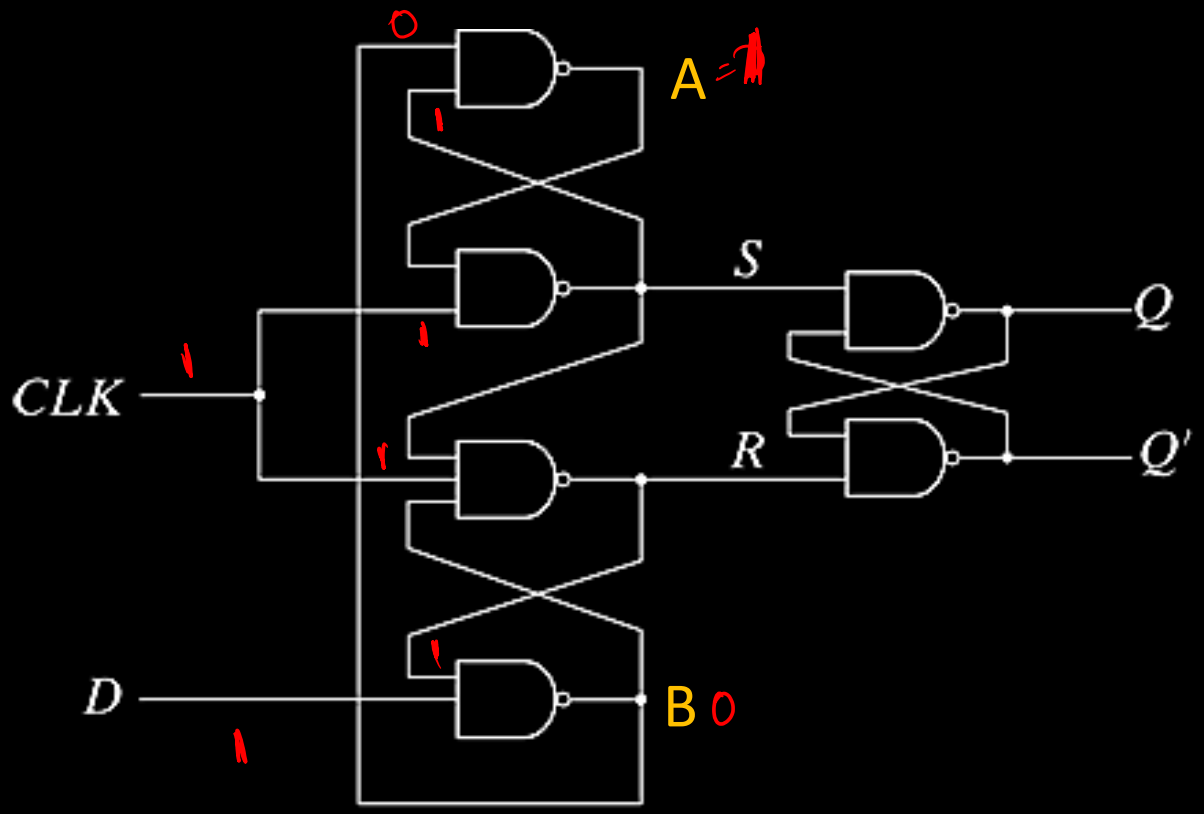


Case II:

- **D = 1, CLK = 0**
- **CLK = 0 => R =1, S = 1**
- **D = 1, R = 1 => B = 0**
- **B = 0, S = 1 => A = 1**
- **ASRB = 1110**
- **SR = 11 => No change**

Edge Triggered D Flip-Flops

D-type positive edge triggered FF



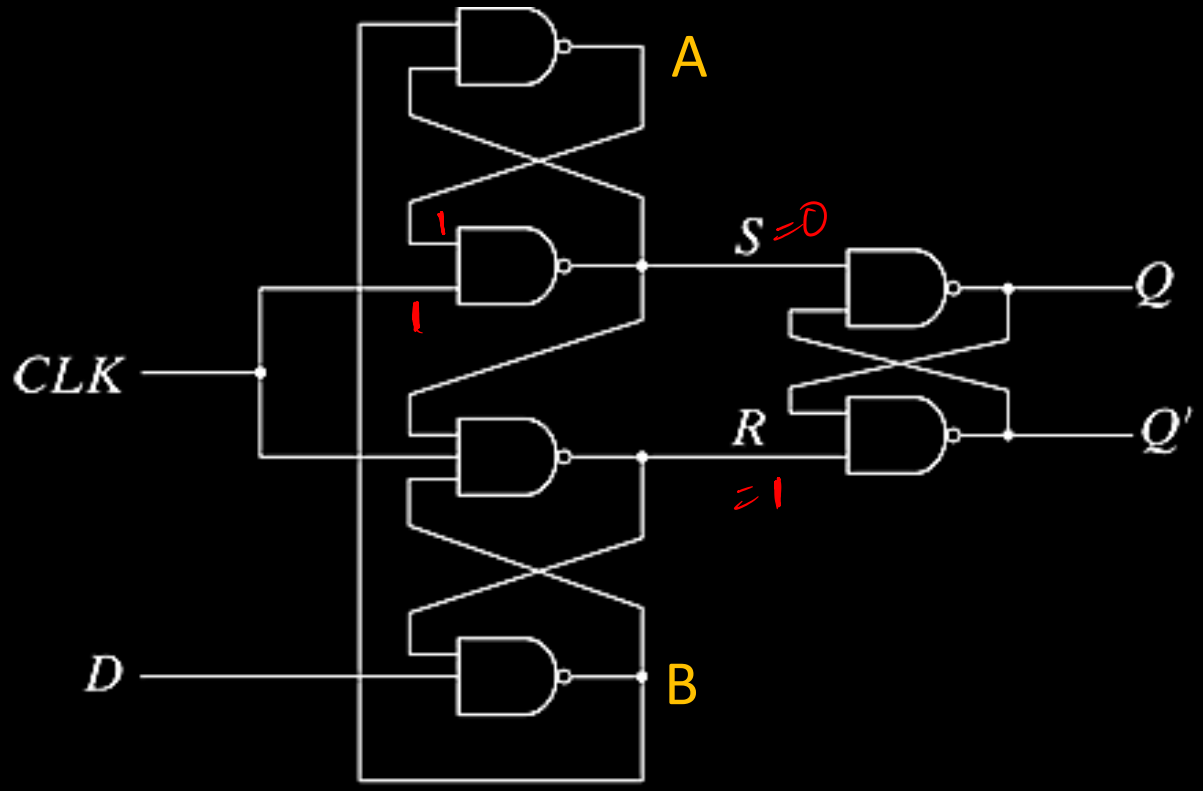
In case II make CLK = 1

- **ASRB = 1110**
- **R=1, D=1 => B=0**
- **B=0, => A=1**

Edge Triggered D Flip-Flops

D=1 Q=1

D-type positive edge triggered FF



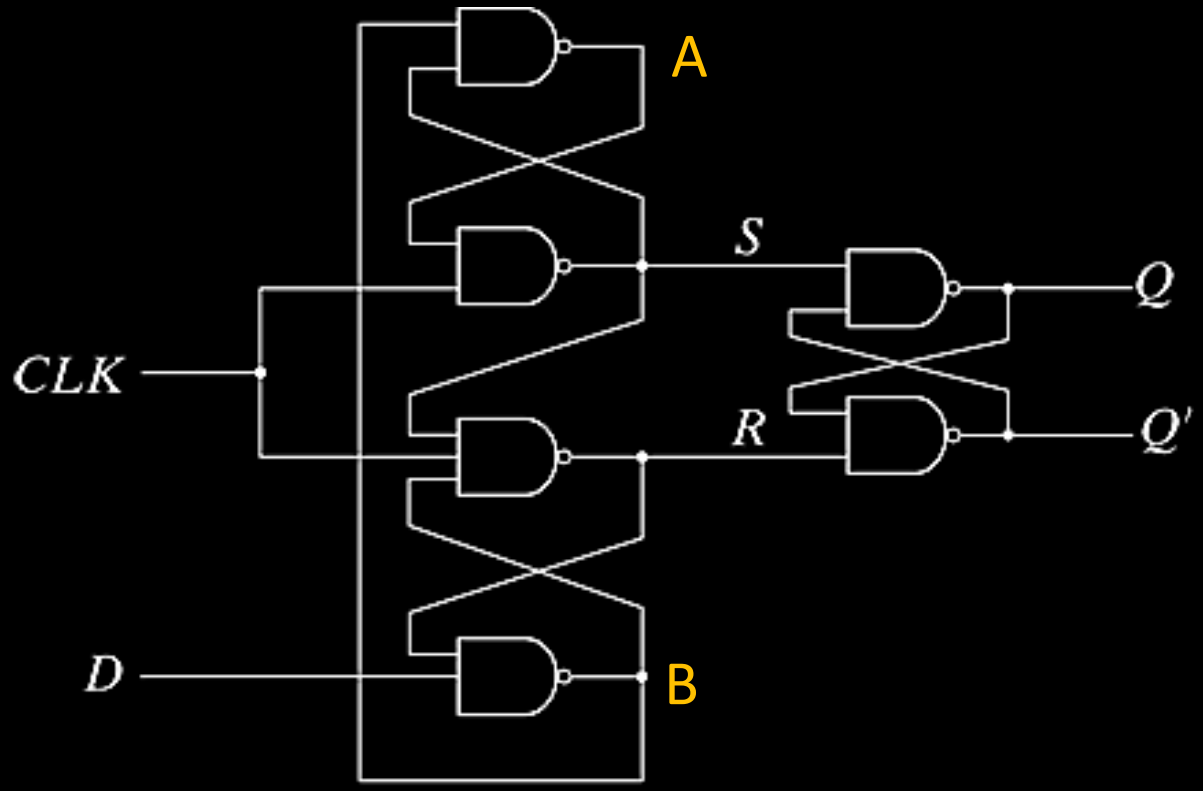
In case II make CLK = 1

- **ASRB = 1110**
- **R=1, D=1 => B=0**
- **B=0, => A=1**
- **A=1,CLK=1 => S=0 =>R=1**
- **SR=01 => SET**
- **ASRB=1010**

=> Q=1

Edge Triggered D Flip-Flops

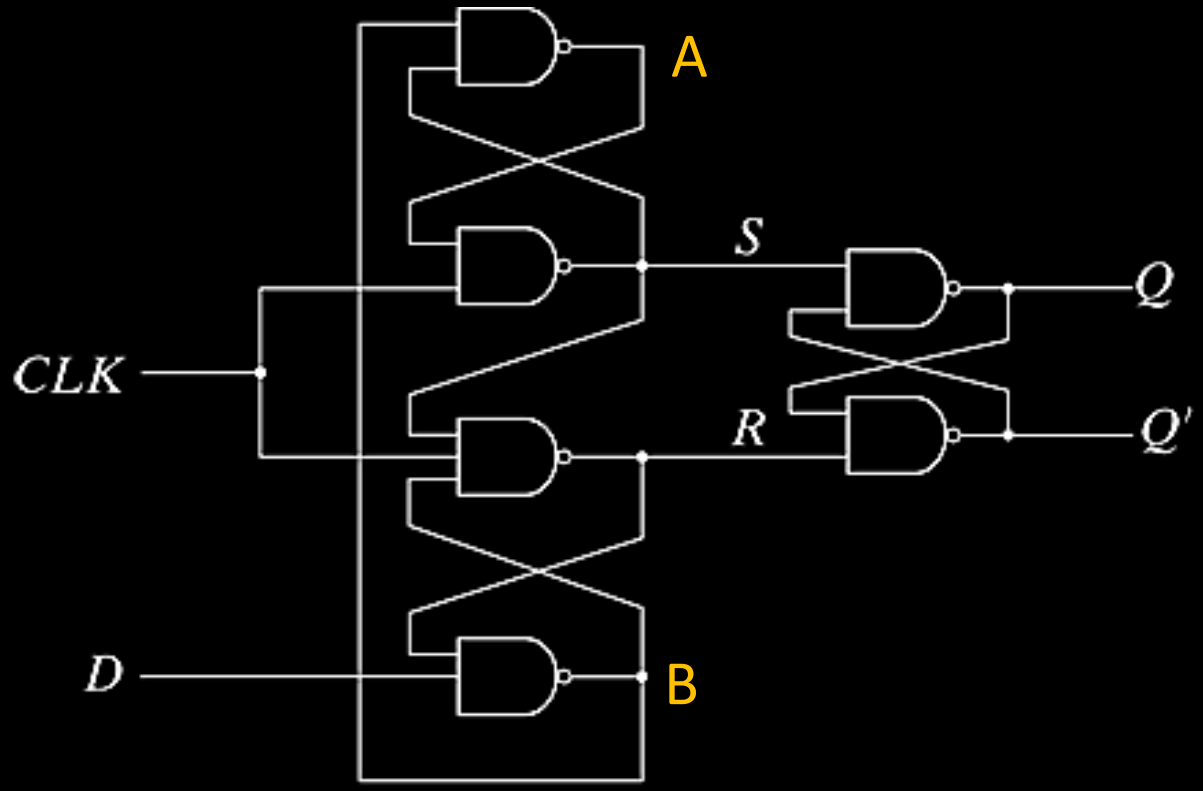
D-type positive edge triggered FF



Going from D=1, CLK=0 to D=0, CLK = 1

Edge Triggered D Flip-Flops

D-type positive edge triggered FF



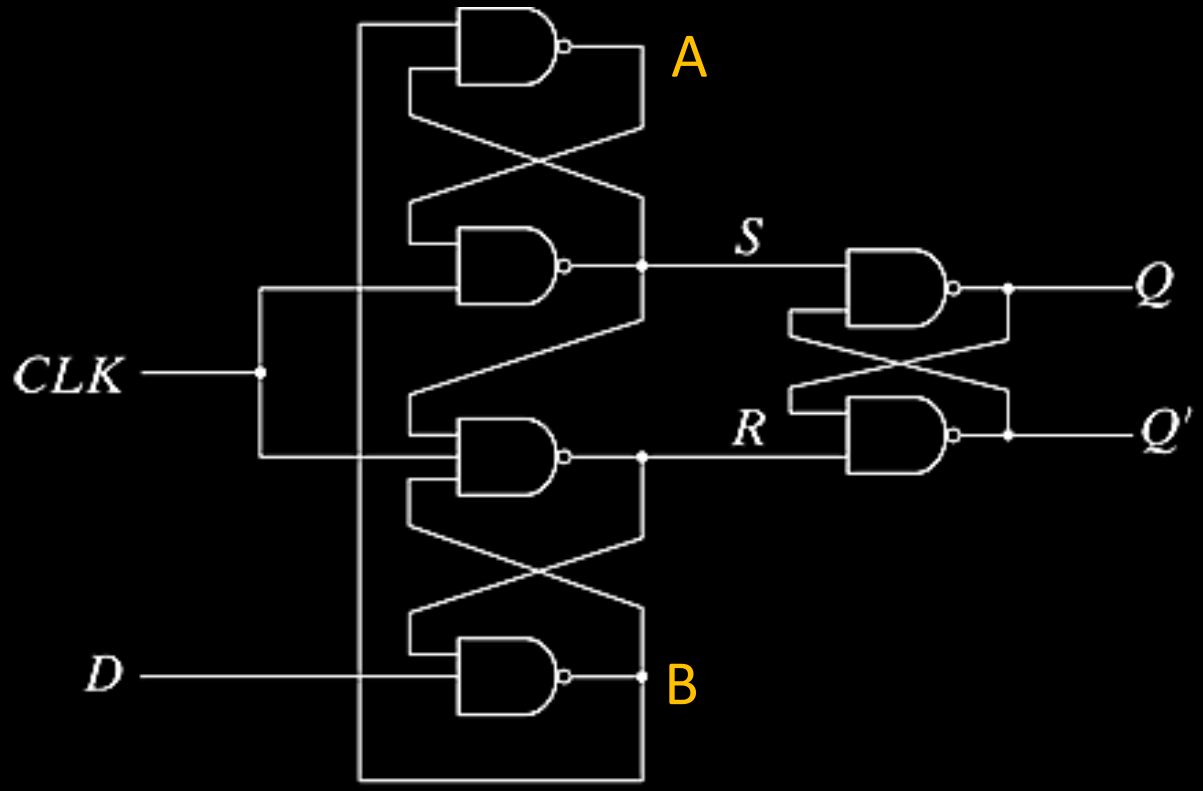
Going from D=1, CLK=0 to D=0, CLK = 1

ASRB=1110

- **D=0 => B=1**
- **CLK=1, A=1 => S=0**

Edge Triggered D Flip-Flops

D-type positive edge triggered FF



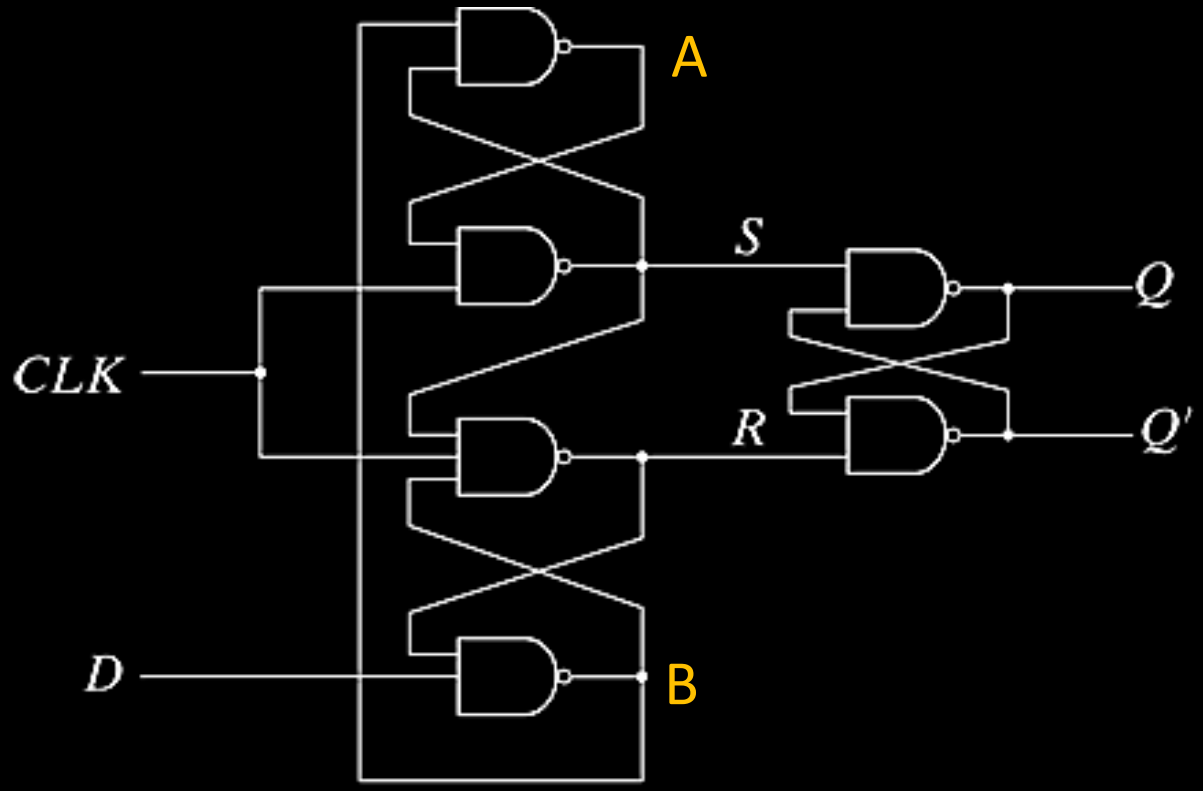
Going from D=1, CLK=0 to D=0, CLK = 1

ASRB=1110

- **D=0 => B=1**
- **CLK=1, A=1 => S=0**
- **S=0,B=1 => A=1**
- **S=0, B=1,CLK=1 => R=1**

Edge Triggered D Flip-Flops

D-type positive edge triggered FF



Going from D=1, CLK=0 to D=0, CLK = 1

ASRB=1110

- **D = 0 => B = 1**
- **CLK = 1, A = 1 => S = 0**
- **S = 0, B = 1 => A = 1**
- **S = 0, B = 1, CLK = 1 => R = 1**
- **SR = 01 => SET**
- **ASRB = 1011**

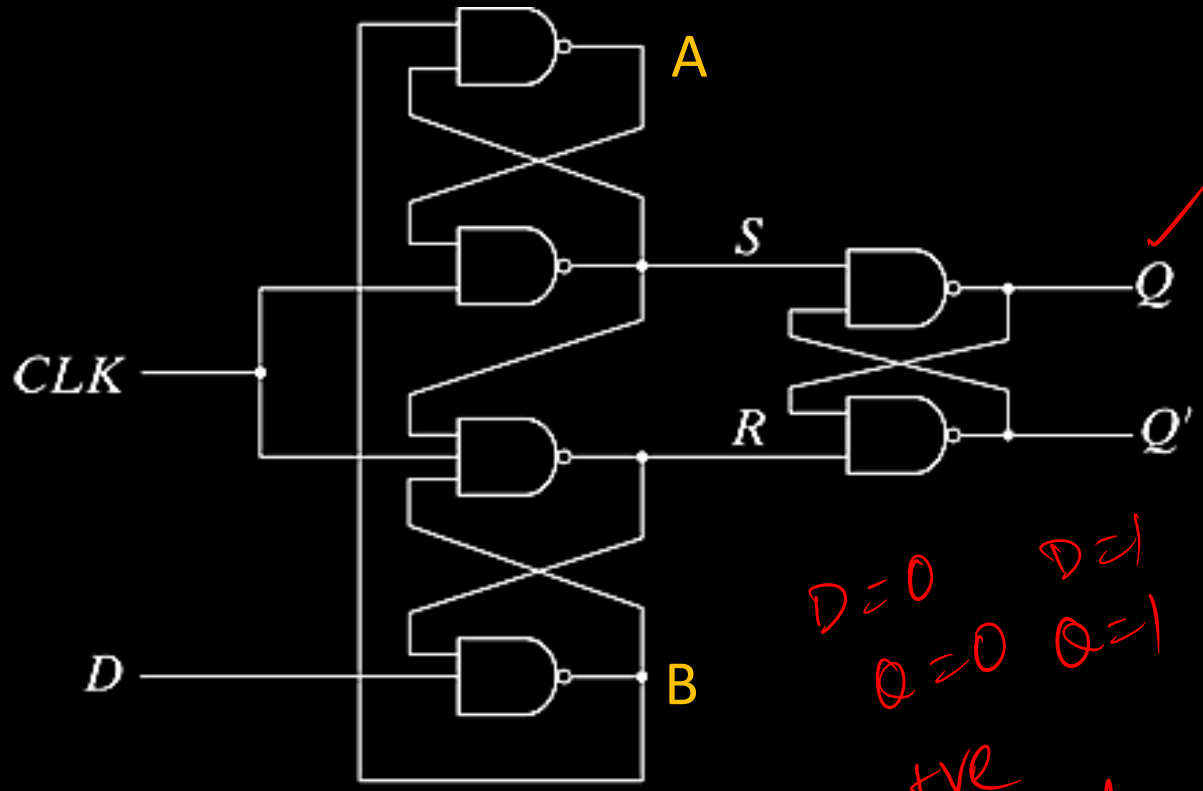
Wrong result

Correct sequence will be:

1. **Go from case II to case I, i.e., let ASRB change from 1110 to 0111. Here A takes two gate delays to change.**

Edge Triggered D Flip-Flops

D-type positive edge triggered FF



D=0 D=1
Q=0 Q=1

Clk => 0 to 1 => +ve triggered

Going from D=1, CLK=0 to D=0, CLK = 1

ASRB=1110

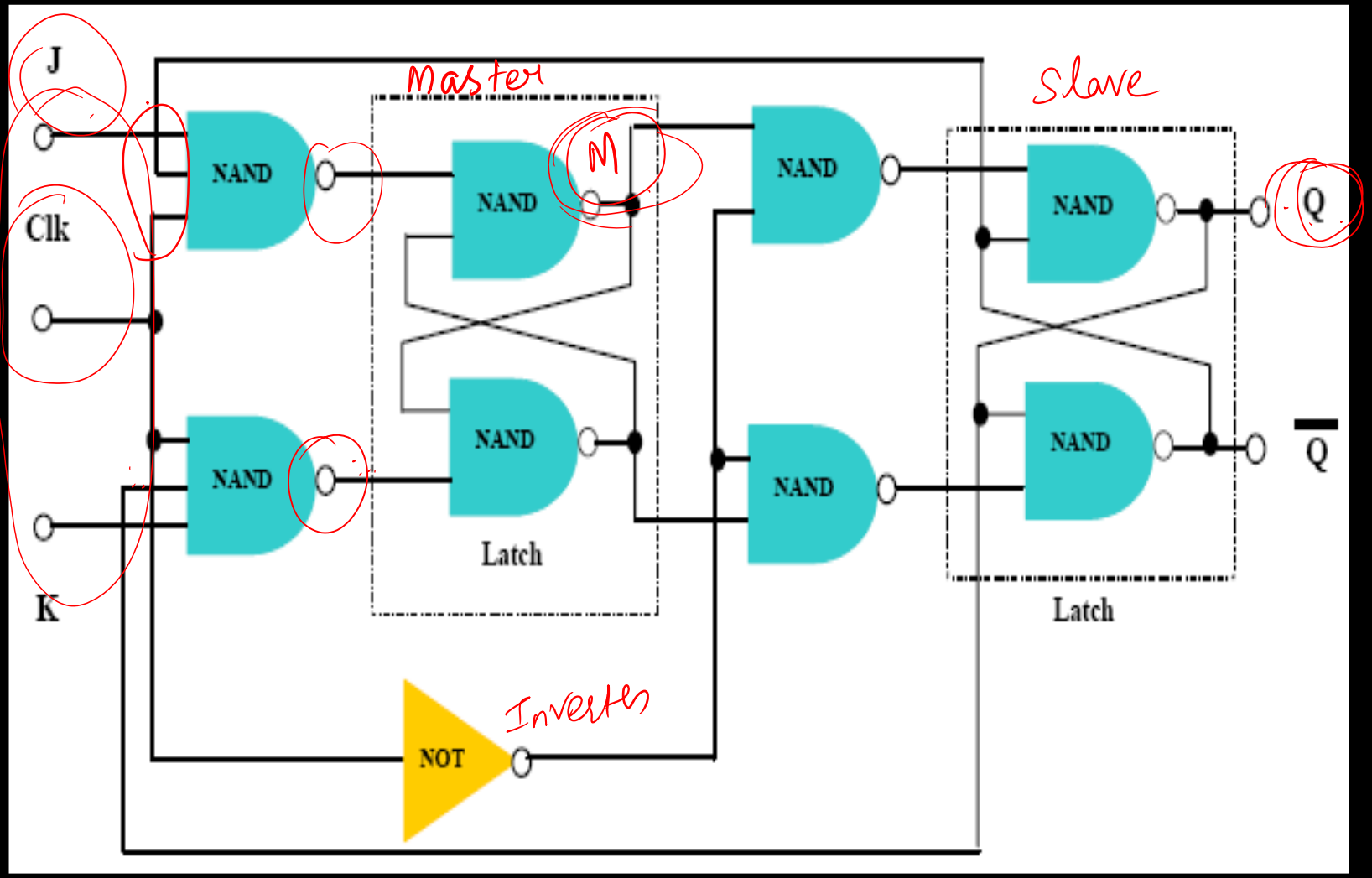
- **D=0 => B=1**
- **CLK=1, A=1 => S=0**
- **S=0,B=1 => A=1**
- **S=0,B=1,CLK=1 => R=1**
- **SR=01 => SET**
- **ASRB=1011**

Wrong result

Correct sequence will be:

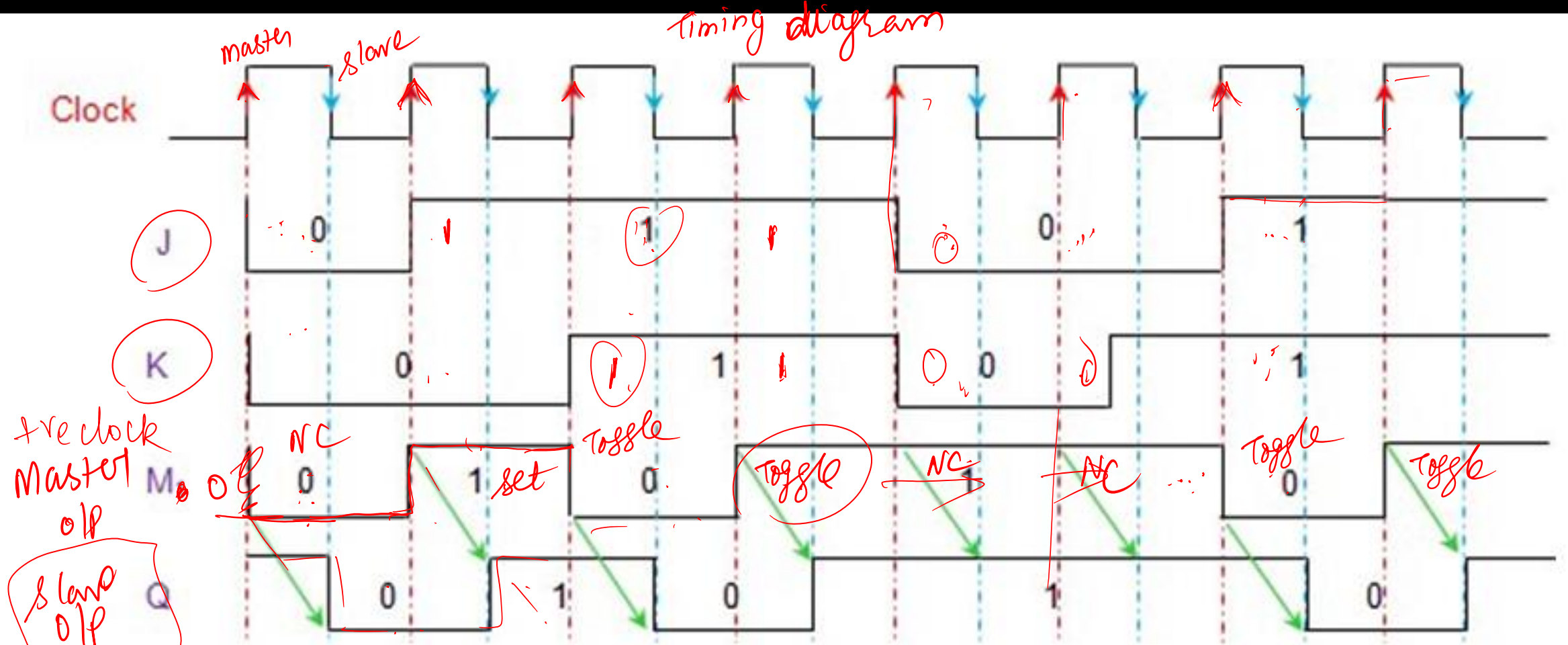
1. **Go from case II to case I, i.e., let ASRB change from 1110 to 0111. Here A takes two gate delays to change. This is Setup time.**
2. **Now go from case I to D=0, CLK=1.**

JK Master - Slave FF



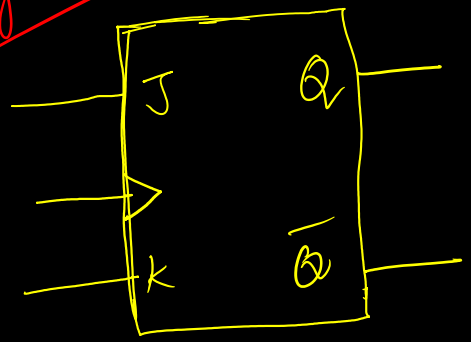
J & K => i/ps are

timing diagram



Edge Triggered JK Flip-Flops

Symbol +ve edge triggered



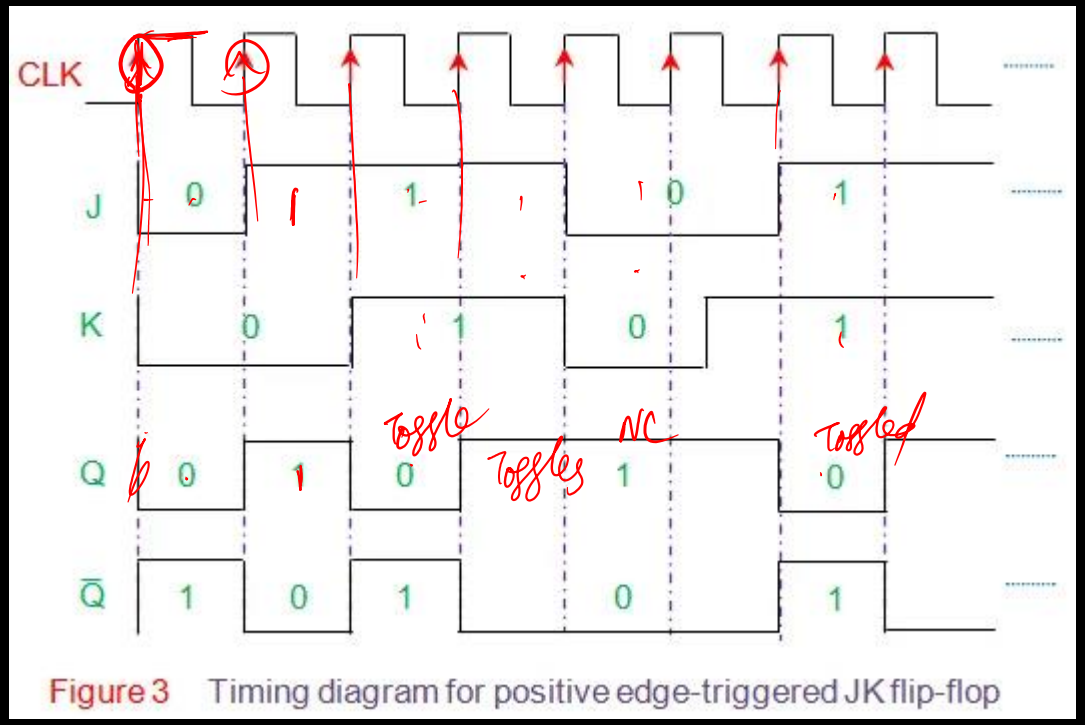
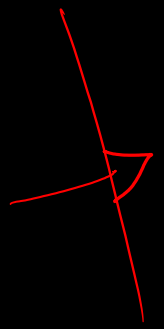
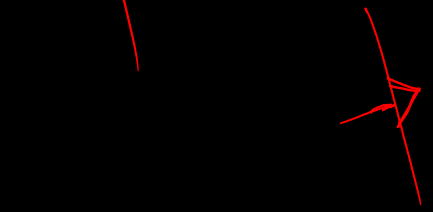
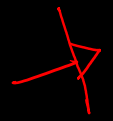


Figure 3 Timing diagram for positive edge-triggered JK flip-flop



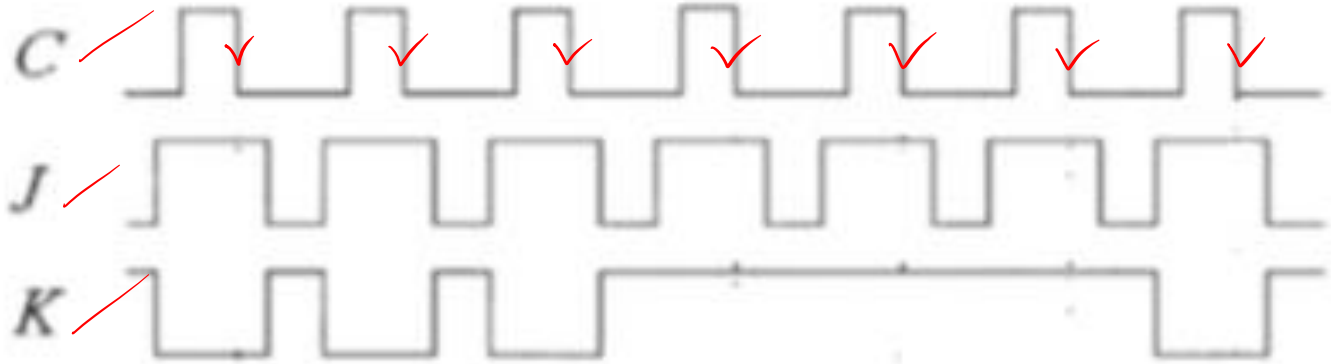
FF



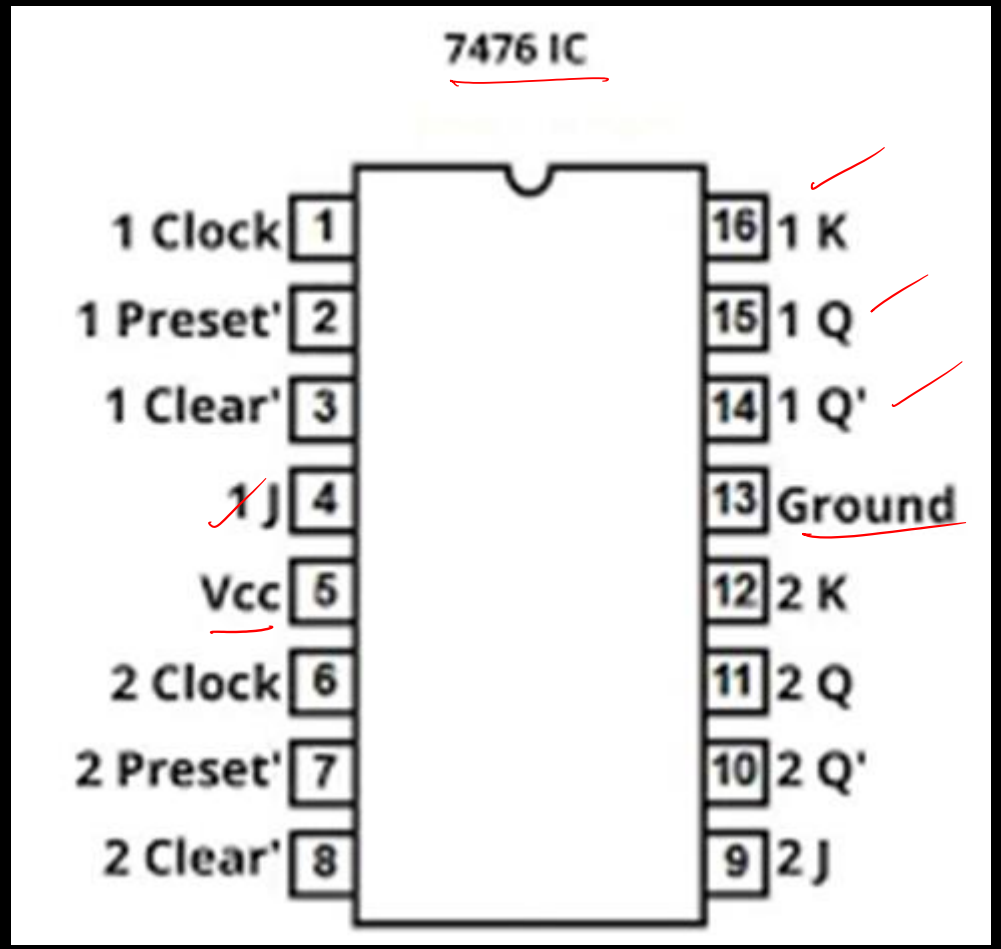


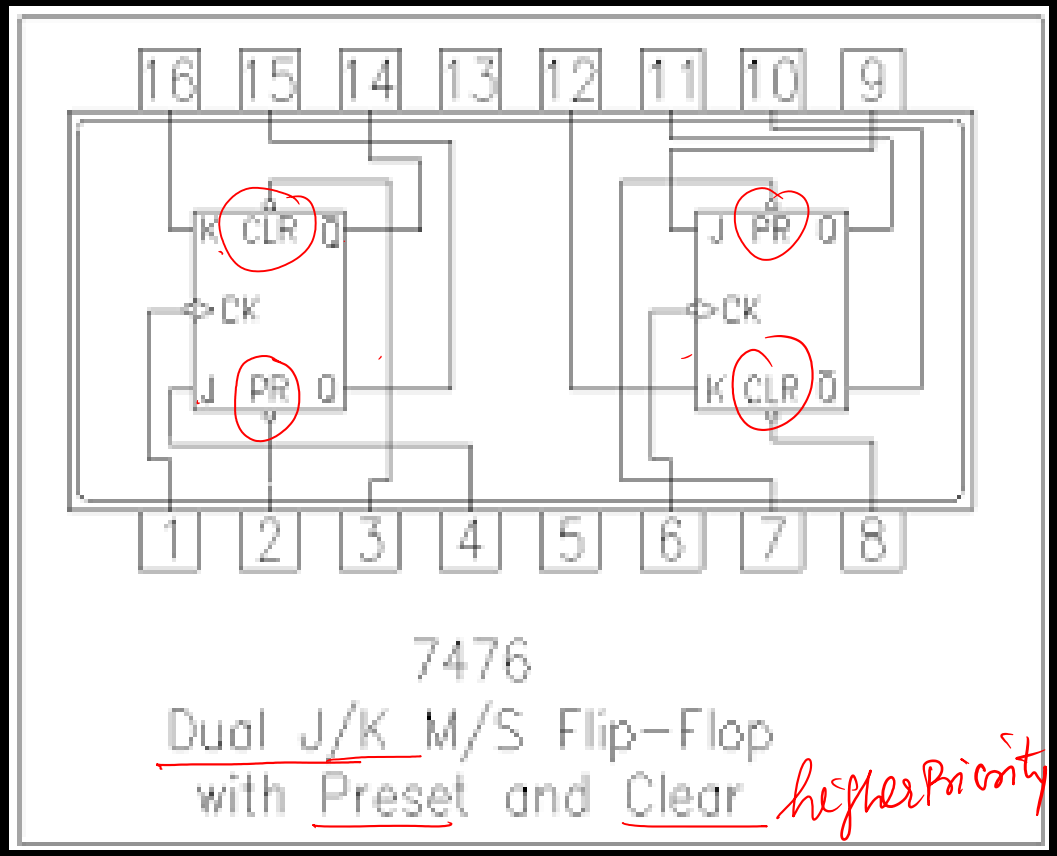
-ve edge triggered

For a negative edge-triggered J-K flip-flop with the inputs shown below, develop the Q output waveform relative to the clock. Assume that Q is initially LOW.



Q

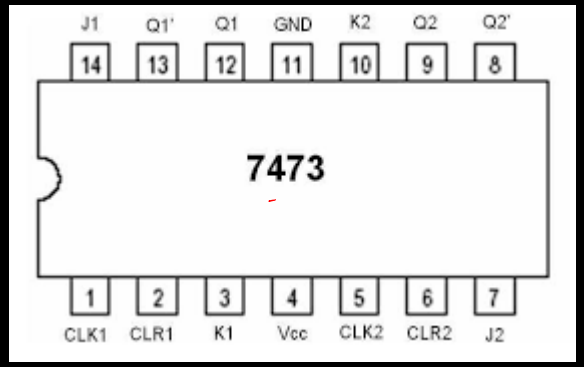


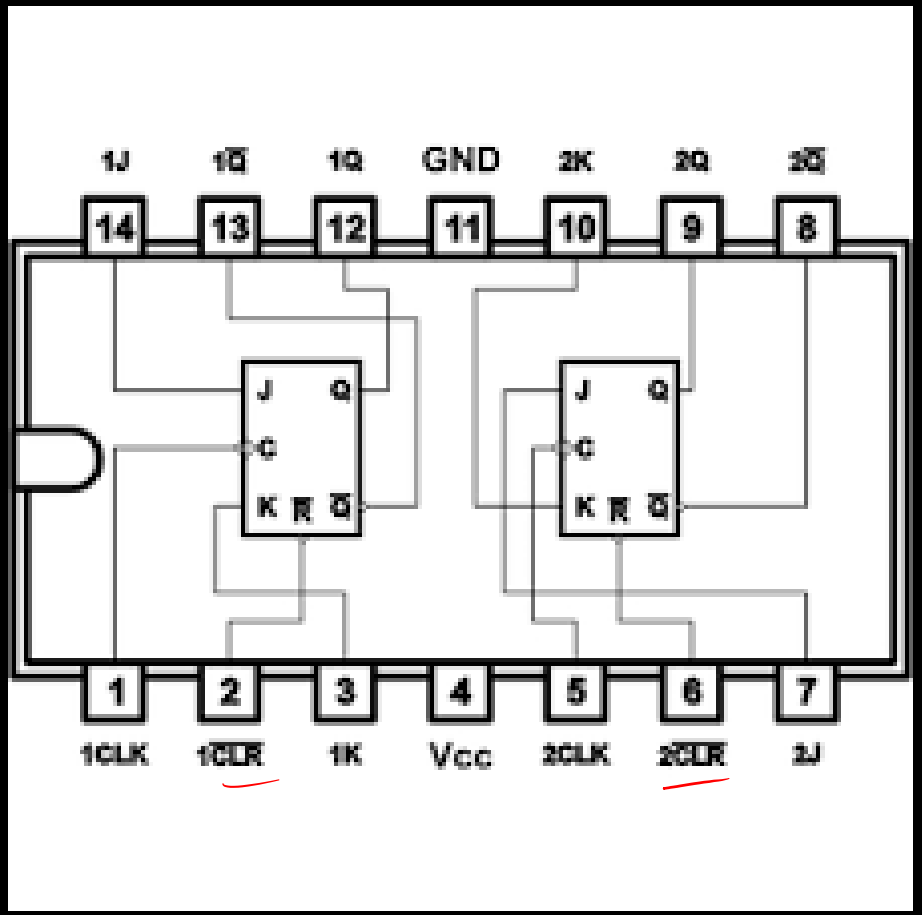


higher Priority

CLR Q=0

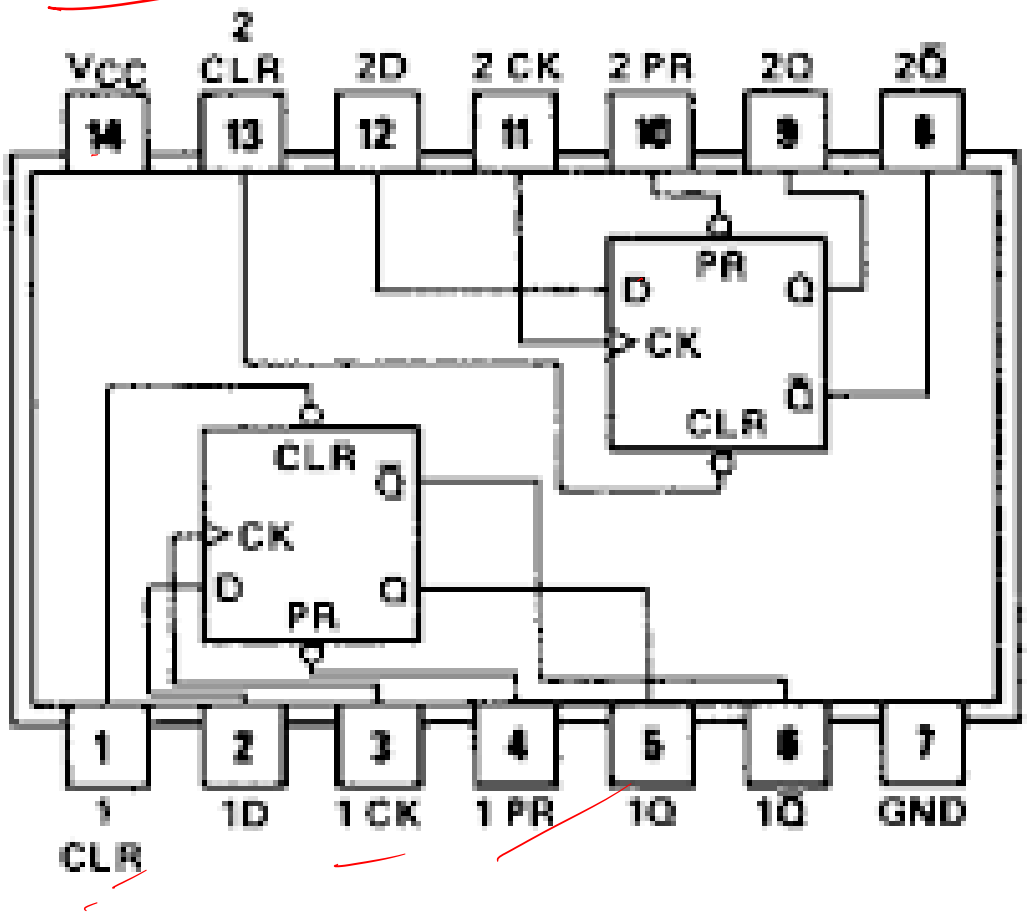
Preset Q=1


J1
Q1'
Q1
GND
K2
Q2
Q2'
14
13
12
11
10
9
8
7473
1
2
3
4
5
6
7
CLK1
CLR1
K1
Vcc
CLK2
CLR2
J2


1J
1Q̄
1Q
GND
2K
2Q
2Q̄
14
13
12
11
10
9
8
J
Q
C
K
R
Q̄
1
2
3
4
5
6
7
1CLK
1CLR
1K
Vcc
2CLK
2CLR
2J

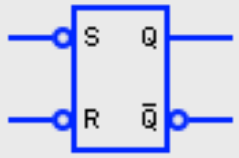
7474

Dff IC

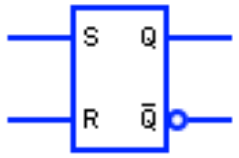

2
VCC
2 CLR
2D
2 CK
2 PR
2Q
2Q̄
14
13
12
11
10
9
8
PR
D
Q
CK
Q̄
CLR
CLR
Q
CK
D
Q
PR
1
2
3
4
5
6
7
1
1D
1 CK
1 PR
1Q
1Q̄
GND
CLR

Flip-Flop Symbols

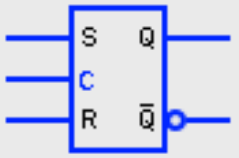
www.electricaltechnology.org



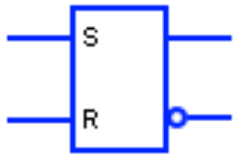
Active Low NAND SR Flipflop



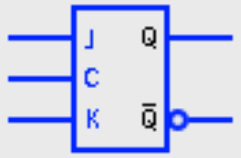
Active High NAND SR Flipflop



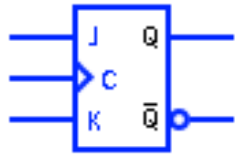
Clocked NAND SR Flipflop



SR Flip-Flop
Set Reset

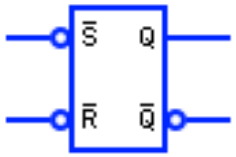


JK Flipflop High level Triggered

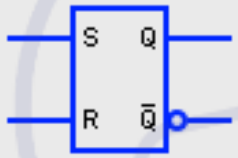


JK Flipflop Rising Edge Triggered

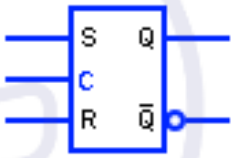
JK Flip-Flop
Activated by
the falling edge



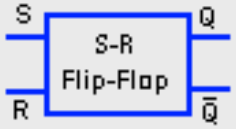
Active Low NOR SR Flipflop



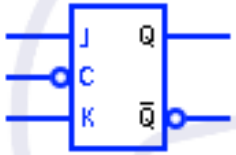
Active High NOR SR Flipflop



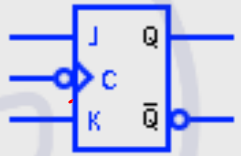
Clocked NOR SR Flipflop



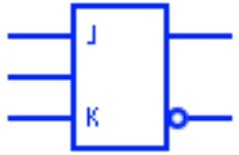
SR Flip-Flop
Set Reset



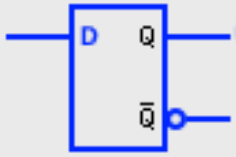
JK Flipflop Low level Triggered



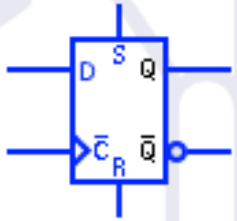
JK Flipflop Falling Edge Triggered



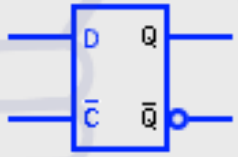
JK Flip-Flop
Generic symbol



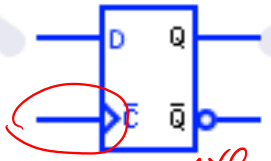
D Flipflop



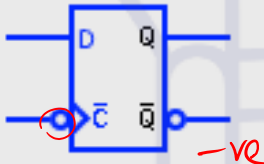
D Flipflop
With Preset-Clear



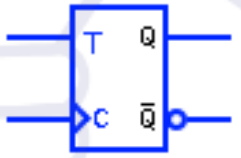
Gated D Flipflop



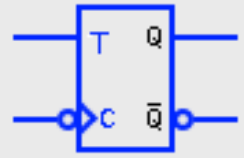
D Flipflop Rising Edge Triggered



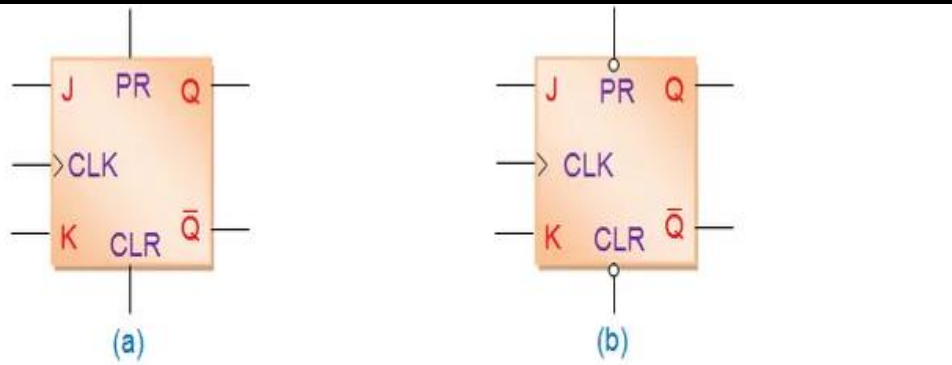
D Flipflop Falling Edge Triggered



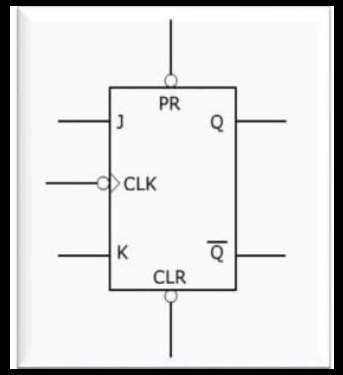
T Flipflop Rising Edge Triggered



T Flipflop Falling Edge Triggered



JK flip-flop with (a) active high preset and clear pins (b) active low preset and clear pins



Clear and Preset ilps have highest Priority

Has 5 inputs named:
J(set),K(reset), PR, CLR, and CLK

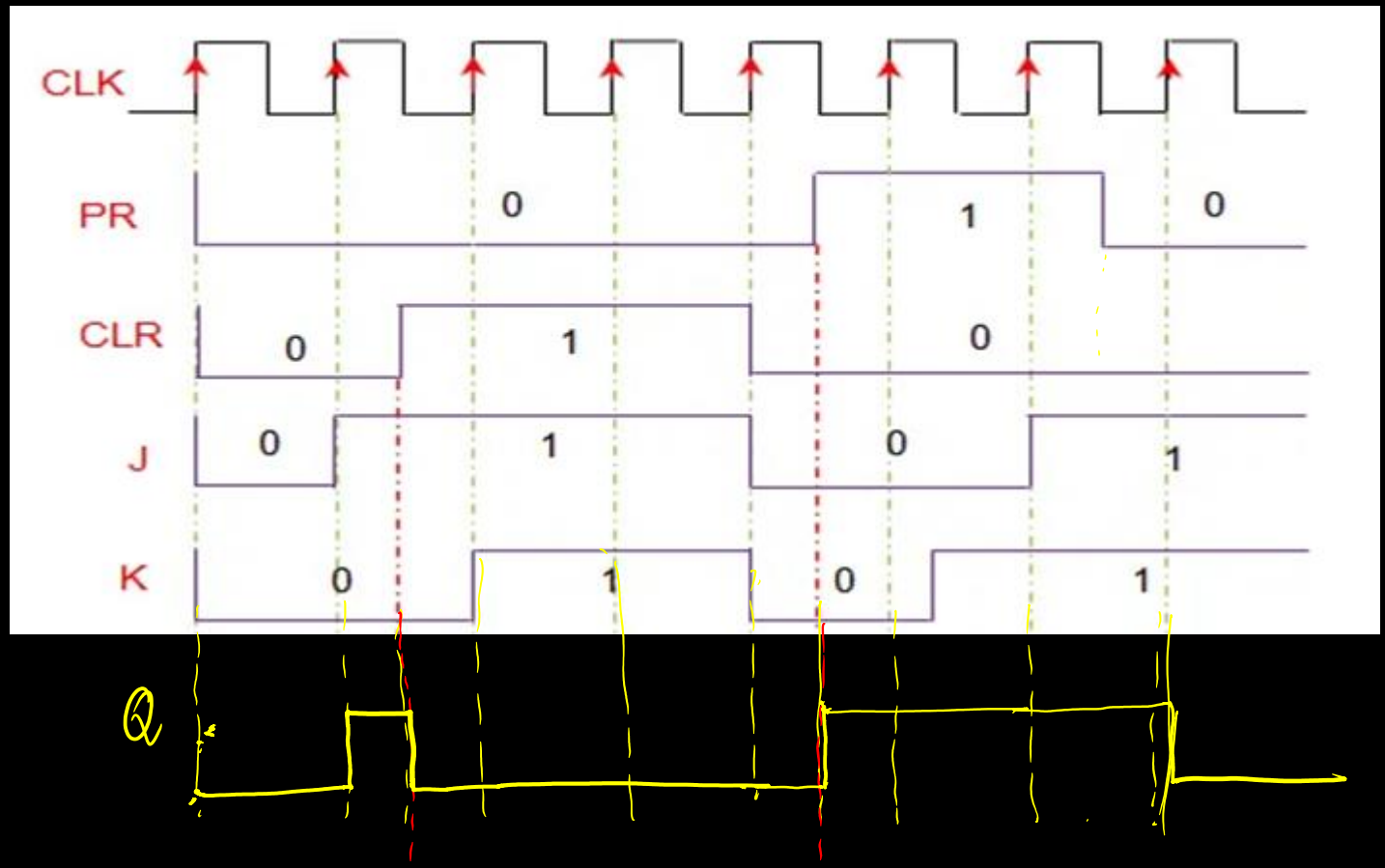
Has 2 outputs: Q and Q'

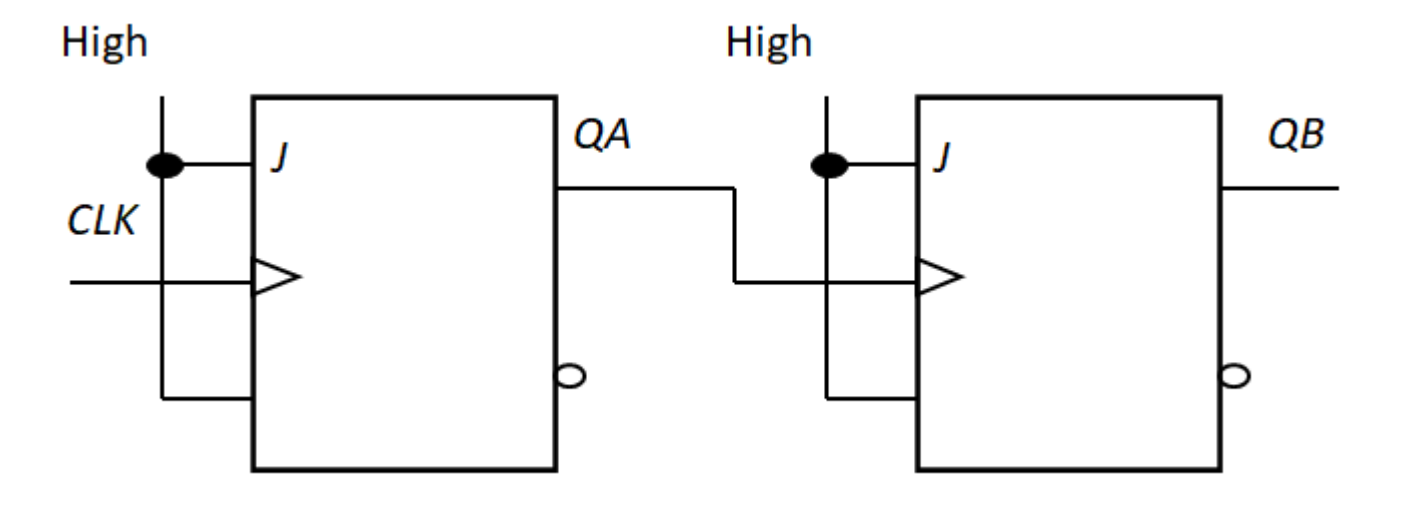
PR = Preset → Q = 1
CLR = Clear → Q = 0
CLK = Clock

Set: when it stores a binary 1

Cleared (reset): when it stores a binary 0

The PR and CLR inputs always **override** the J,K inputs.





Q_A & Q_B = ?

Inference