CDA 3200 Digital Systems

Instructor: Dr. Janusz Zalewski
Developed by: Dr. Dahai Guo
Spring 2012

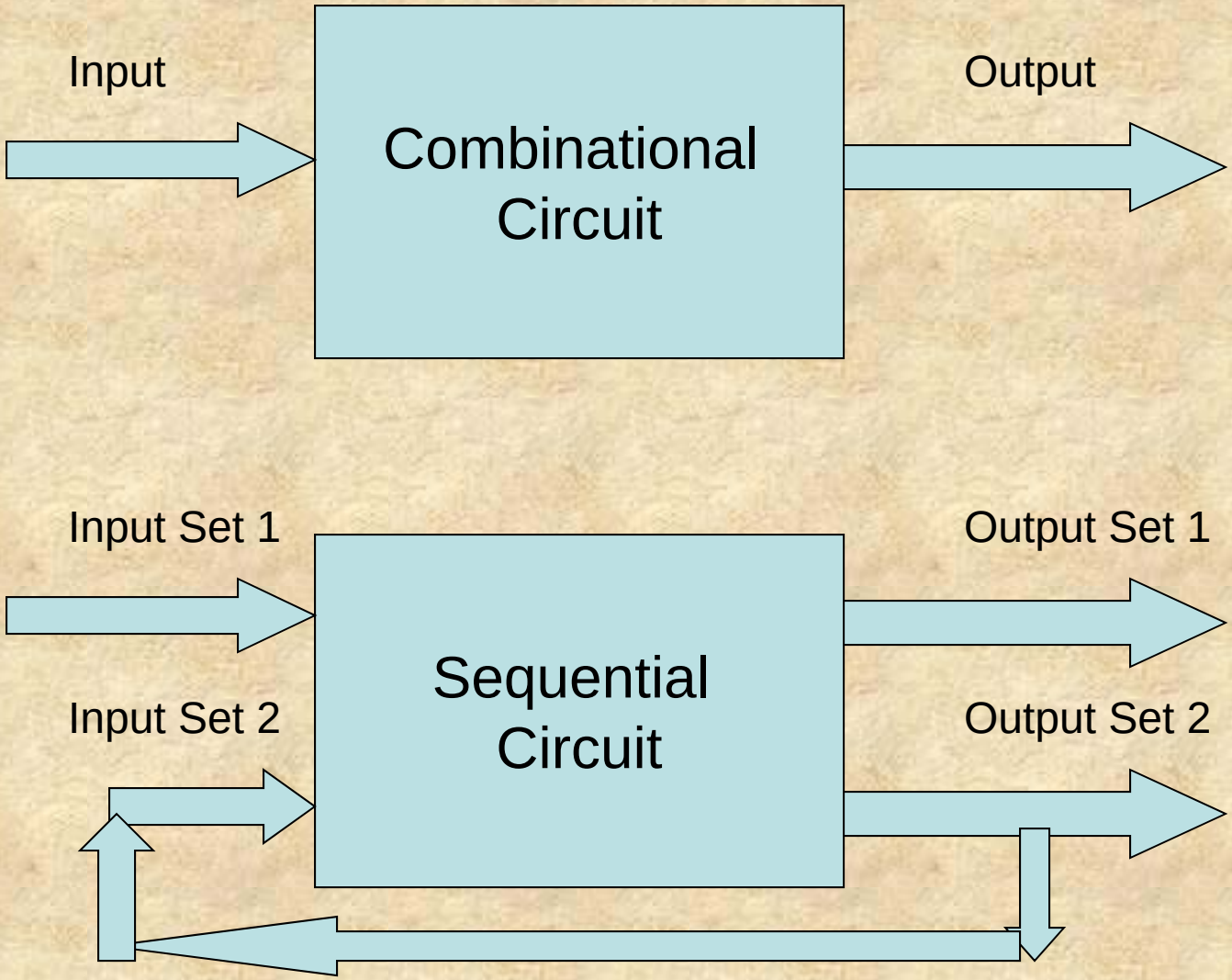
Outline

- SR Latch
- D Latch
- Edge-Triggered D Flip-Flop (FF)
- S-R Flip-Flop (FF)
- J-K Flip-Flop (FF)
- T Flip-Flop (FF)
- Flip-Flops (FFs) with Additional Inputs

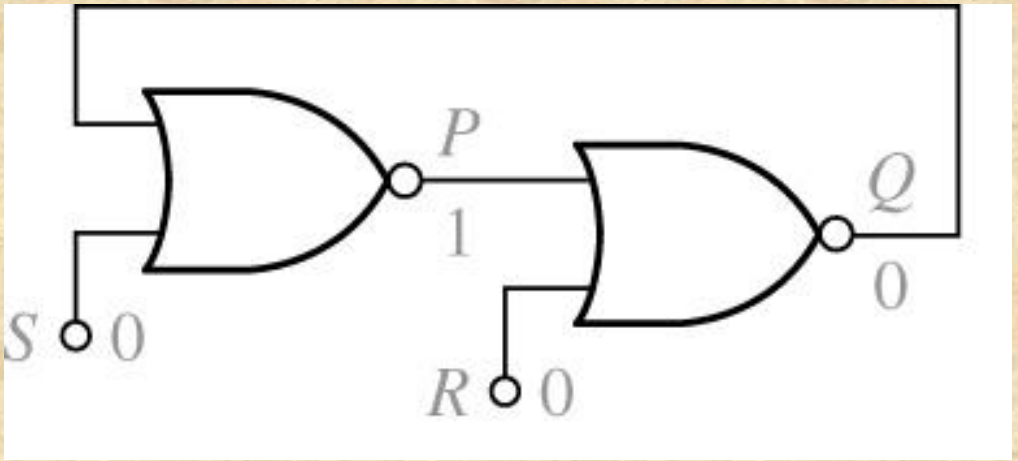
SR Latch (1/17)

- Combinational circuits
  - Outputs depend on present inputs
- Sequential circuits
  - Outputs depend on both present and the past sequence of inputs.
  - Have memory.

SR Latch (2/17)


Input
Combinational Circuit
Output
Input Set 1
Input Set 2
Sequential Circuit
Output Set 1
Output Set 2

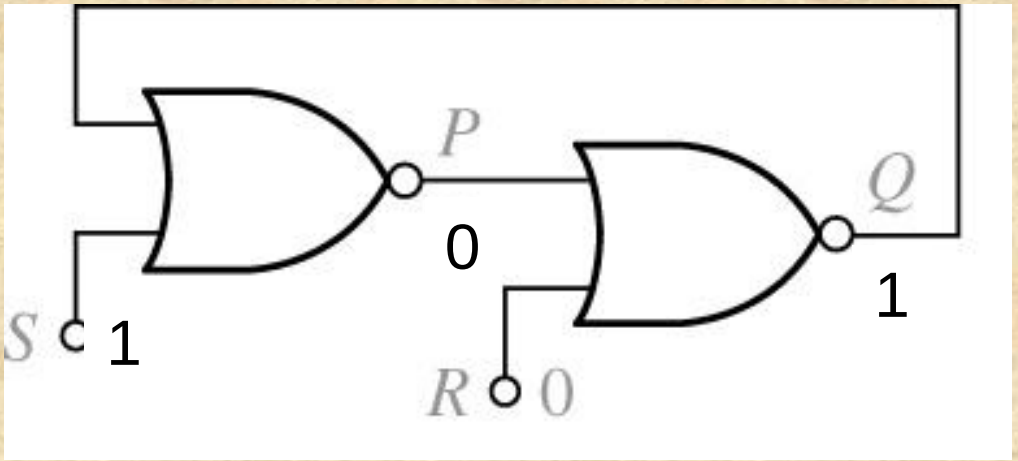
SR Latch (3/17)



X	Y	NOR
0	0	1
0	1	0
1	0	0
1	1	0

The circuit can assume an initial and stable state: SR/PQ=00/10.

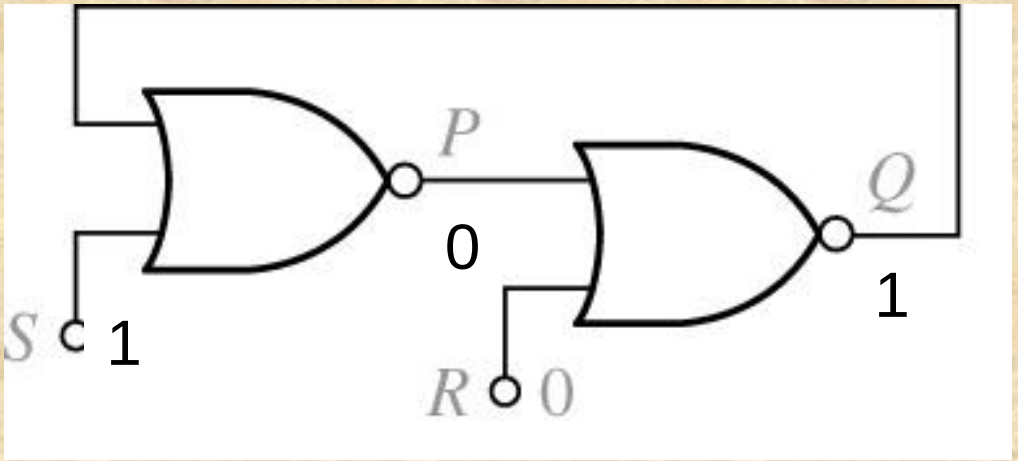
SR Latch (4/17)



X	Y	NOR
0	0	1
0	1	0
1	0	0
1	1	0

- SR/PQ=10/01 is also stable.

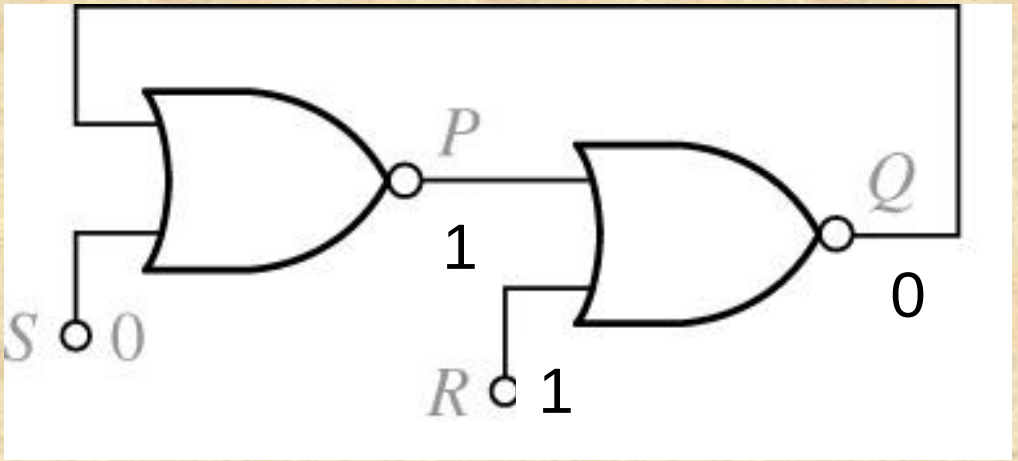
SR Latch (5/17)



X	Y	NOR
0	0	1
0	1	0
1	0	0
1	1	0

- SR/PQ=00/01 is also stable.

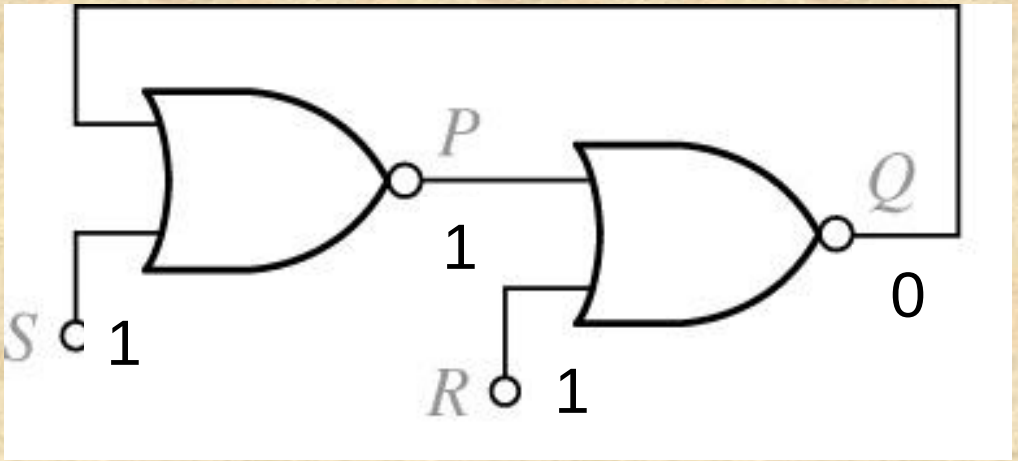
SR Latch (6/17)



X	Y	NOR
0	0	1
0	1	0
1	0	0
1	1	0

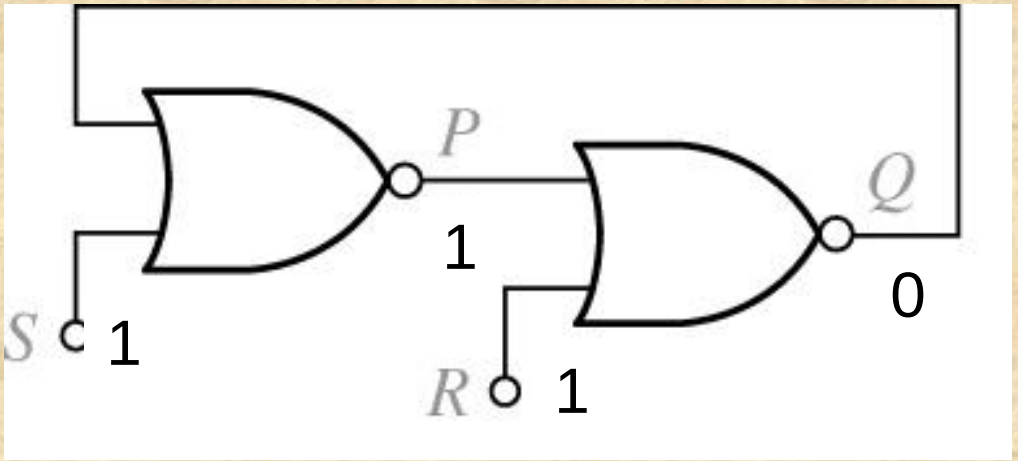
- SR/PQ=01/10 is also stable.
- SR/PQ=00/10 is also stable.

SR Latch (7/17)



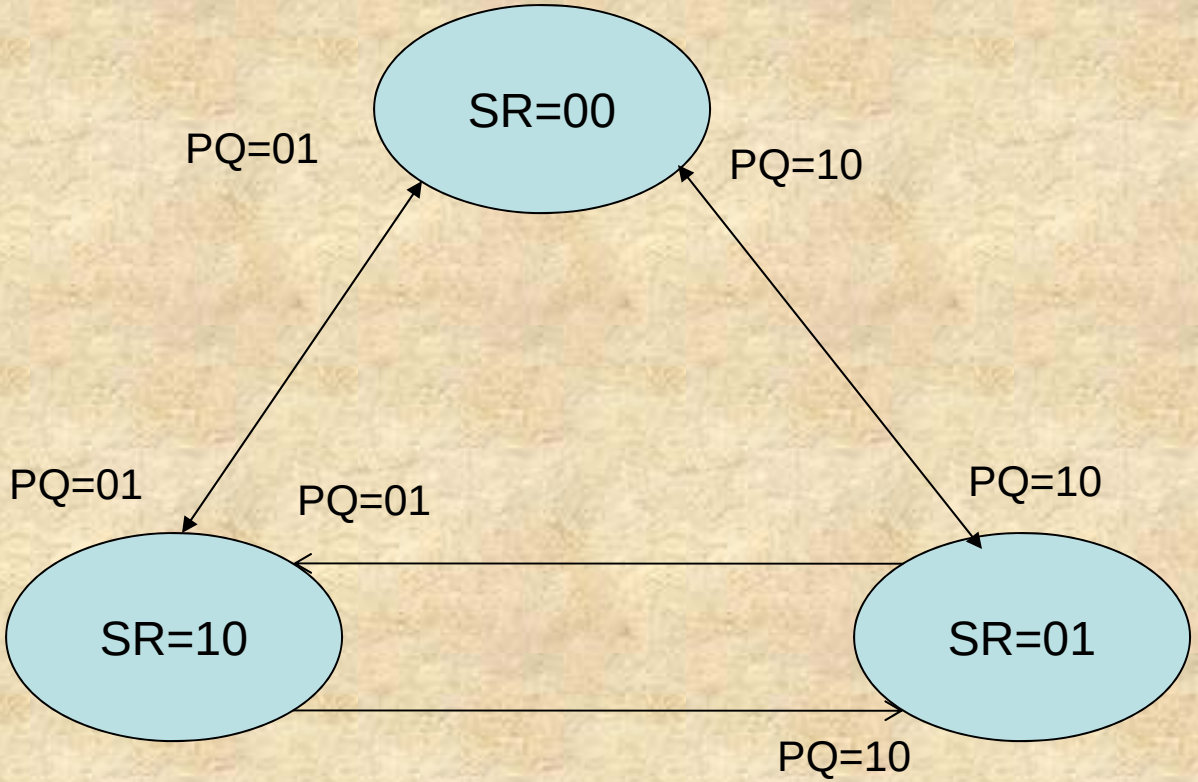
X	Y	NOR
0	0	1
0	1	0
1	0	0
1	1	0

SR Latch (8/17)



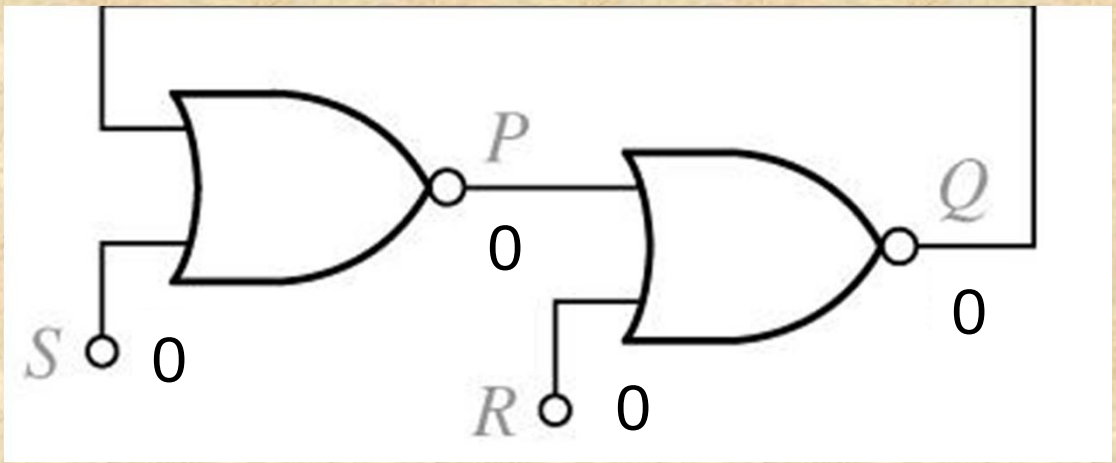
X	Y	NOR
0	0	1
0	1	0
1	0	0
1	1	0

SR Latch (9/17)



The change between any two of 00, 10, 01 will reach a stable state.

SR Latch (10/17)



X	Y	NOR
0	0	1
0	1	0
1	0	0
1	1	0

- What is PQ when the circuit is stable?

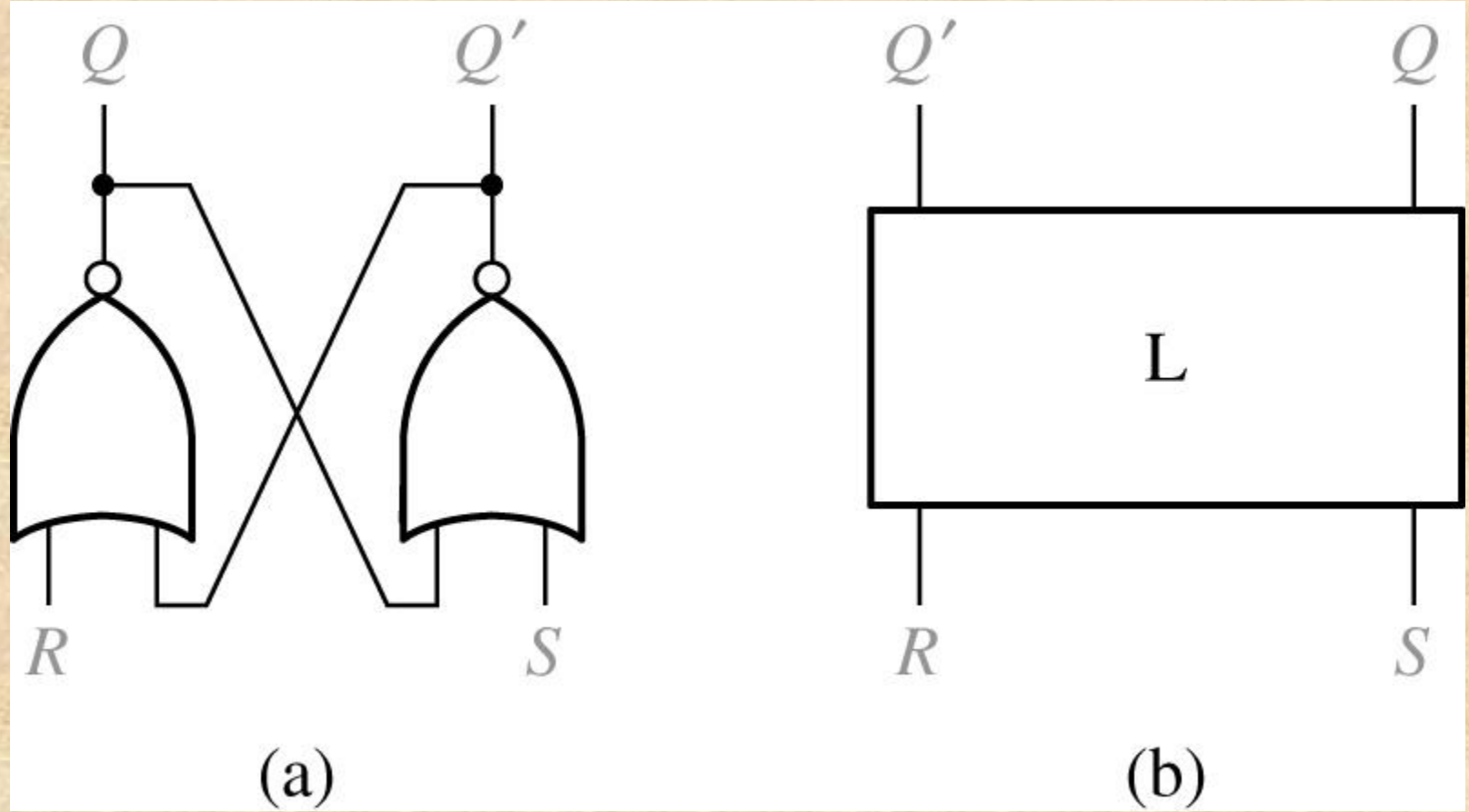
SR Latch (11/17)

- SR=11 is restricted in SR latch.
- PQ cannot be both 1.

SR Latch (12/17)

- When SR=10, PQ=01 is stable.
- When SR=01, PQ=10 is stable.
- When SR=00, both PQ=10 and PQ=01 are stable.
- Note
  - In the stable states, P=Q’
  - Any change to SR=00 will not change PQ.
  - SR=00 is used to keep states (remember what happened.)

SR Latch (13/17)



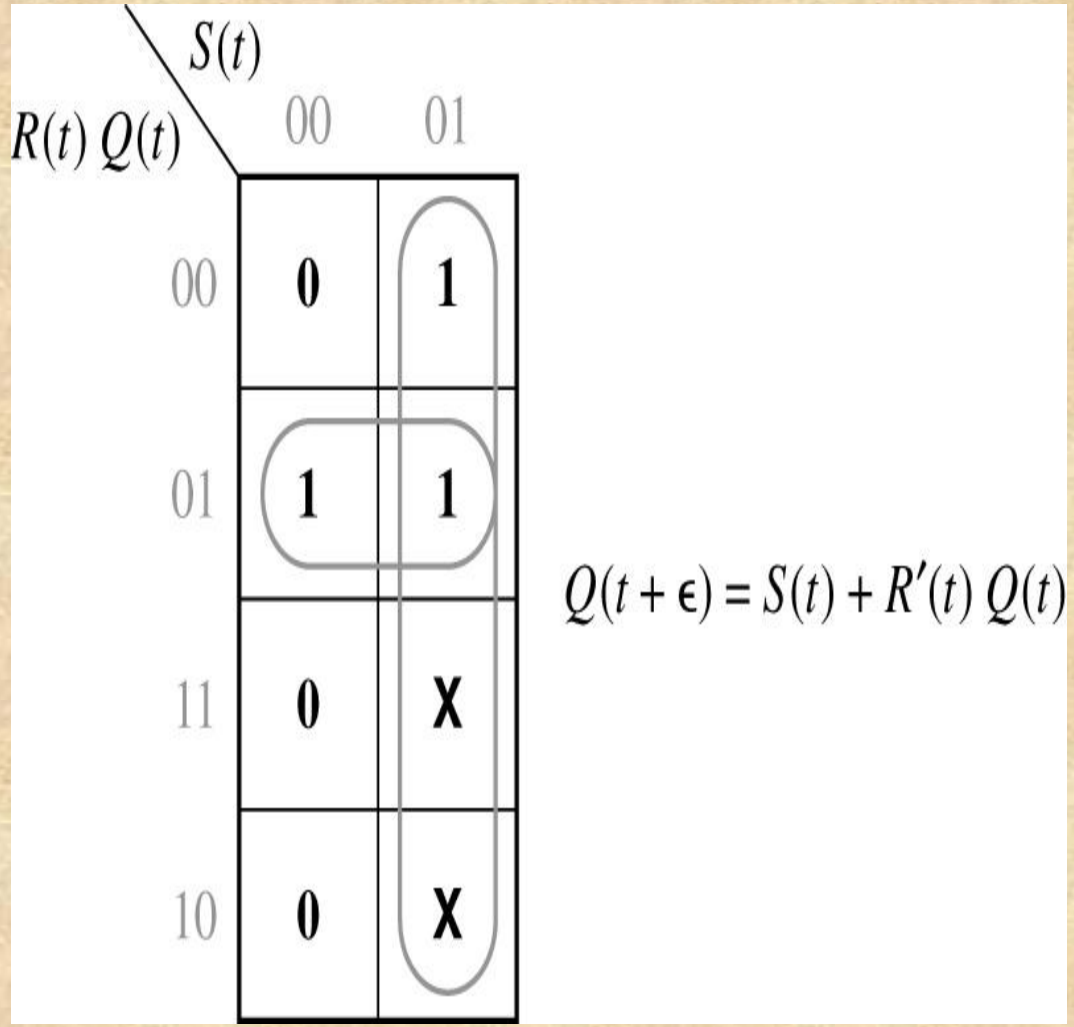
S: Set Q
R: Reset Q

SR Latch (14/17)

- How to draw a truth table for an SR latch?
  - Input?
  - Output?

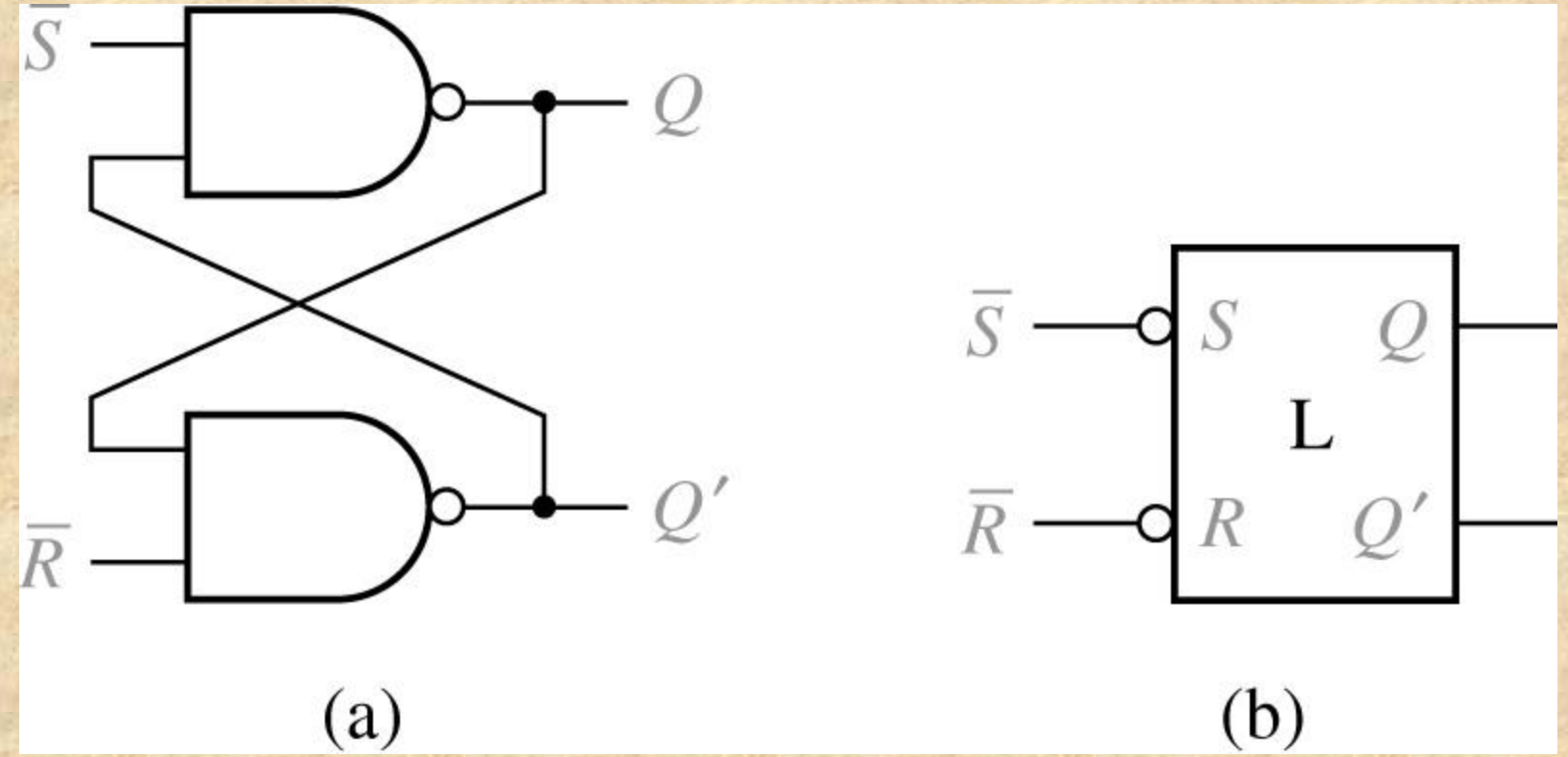
SR Latch (15/17)

S	R	Q	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	X
1	1	1	X



SR Latch (16/17)

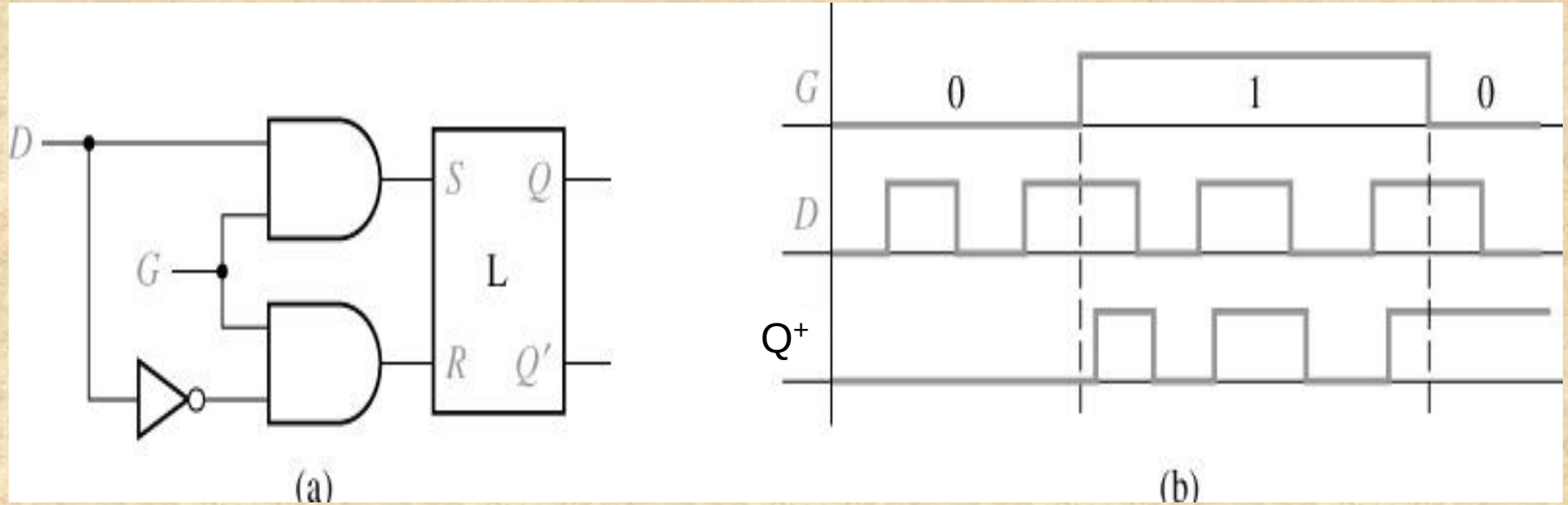
- Alternatively, an SR latch can be realized using NAND gates.



SR Latch (17/17)

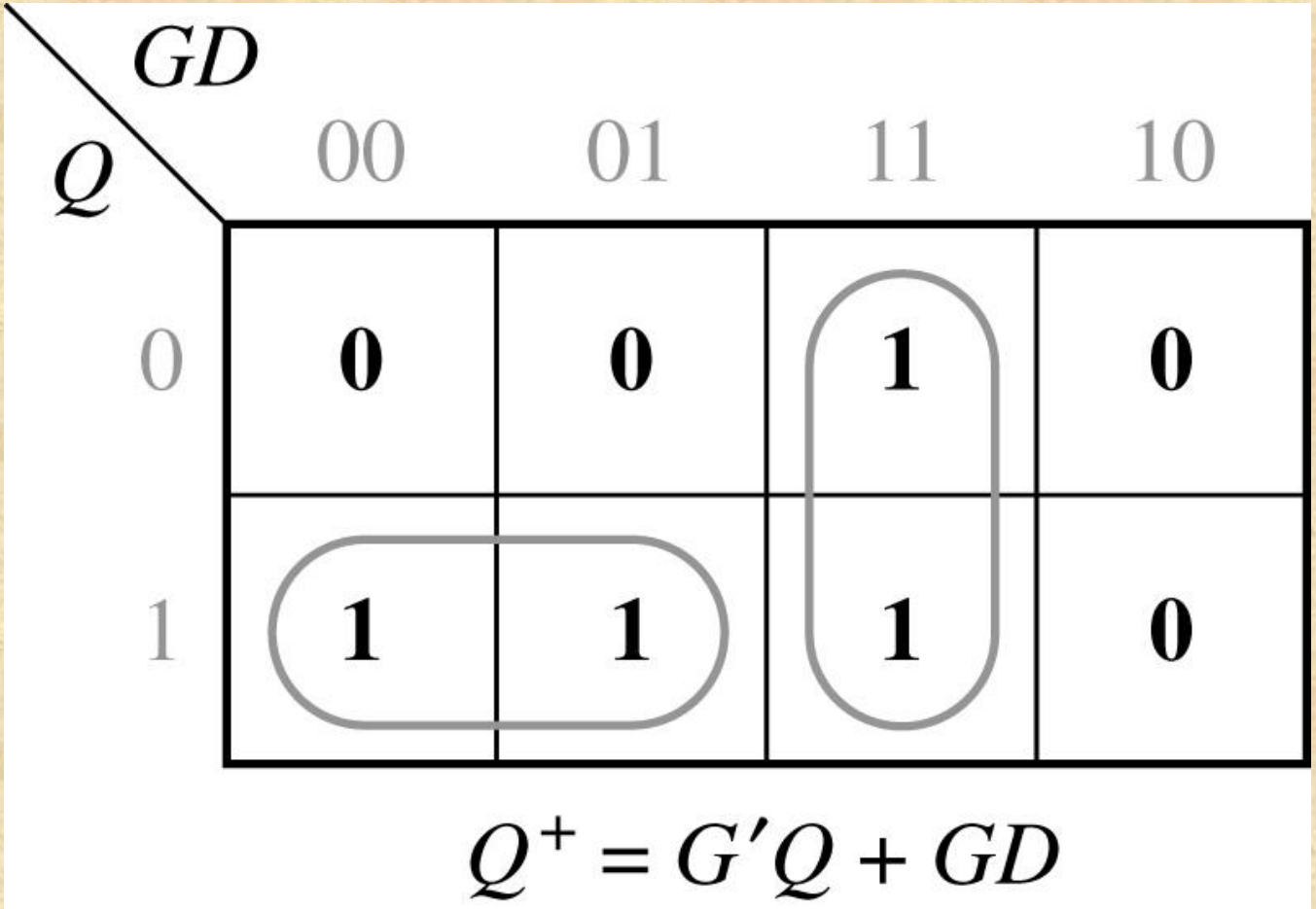
- Alternatively, an SR latch can be realized using NAND gates.
  - S-bar R-bar Q Q^+
  - 1 1 0 0
  - 1 1 1 1
  - 1 0 0 0
  - 1 0 1 0
  - 0 1 0 1
  - 0 1 1 1
  - 0 0 0 X
  - 0 0 1 X

Gated D Latch (1/3)



- What are S and R when G=0?
- G=0 keeps states: Q^+=D.
- Can SR=11 ever occur?
- Q^+=D when G=1.

Gated D Latch (2/3)



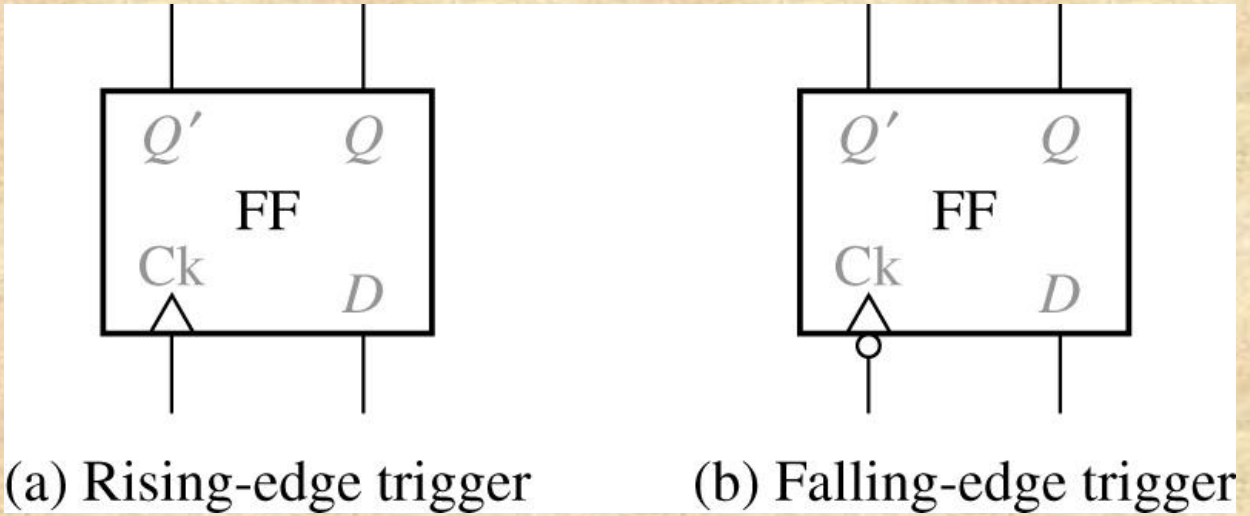
- No “do not care” terms.

Gated D Latch (3/3)

- if(G==1){
- $Q^+ = D$;
- }else{
- $Q^+ = Q$;
- }

Edge-Triggered D Flip-Flop (FF) (1/3)

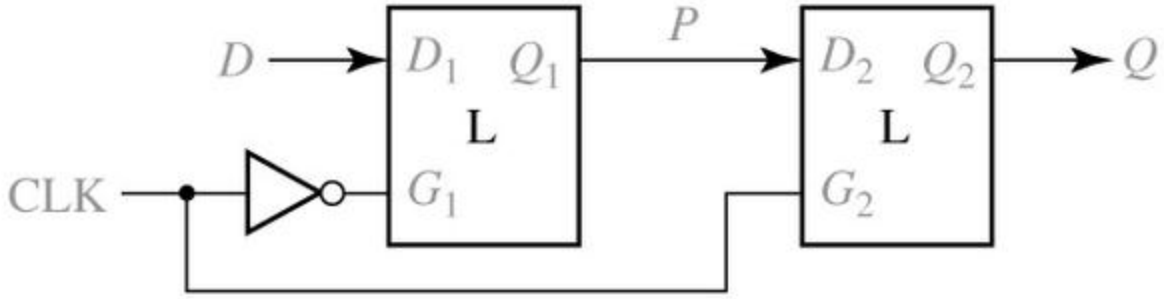
- If the G signal in the D latch is connected to a clock input, the output changes only in response to the clock, not to a change in D.
- And we call this latch a D Flip-Flop (FF).



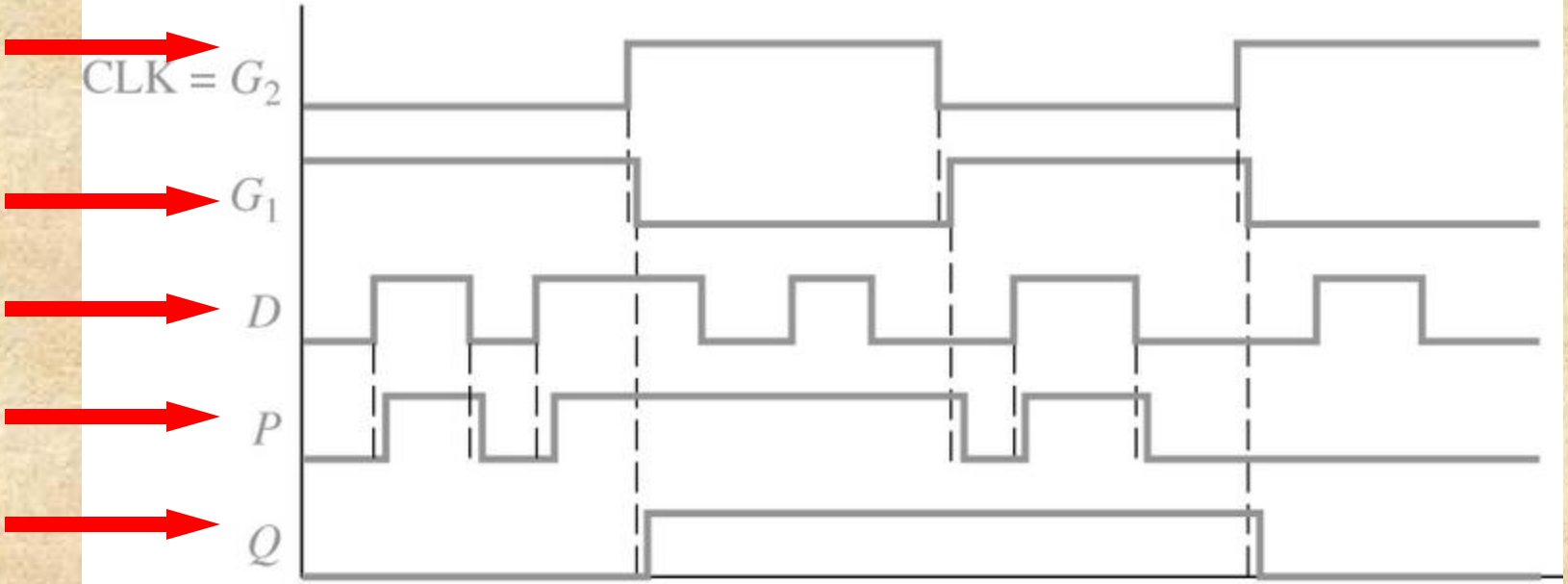
(a) Rising-edge trigger (b) Falling-edge trigger

$Q^+=D$

Edge-Triggered D Flip-Flop (FF) (2/3)



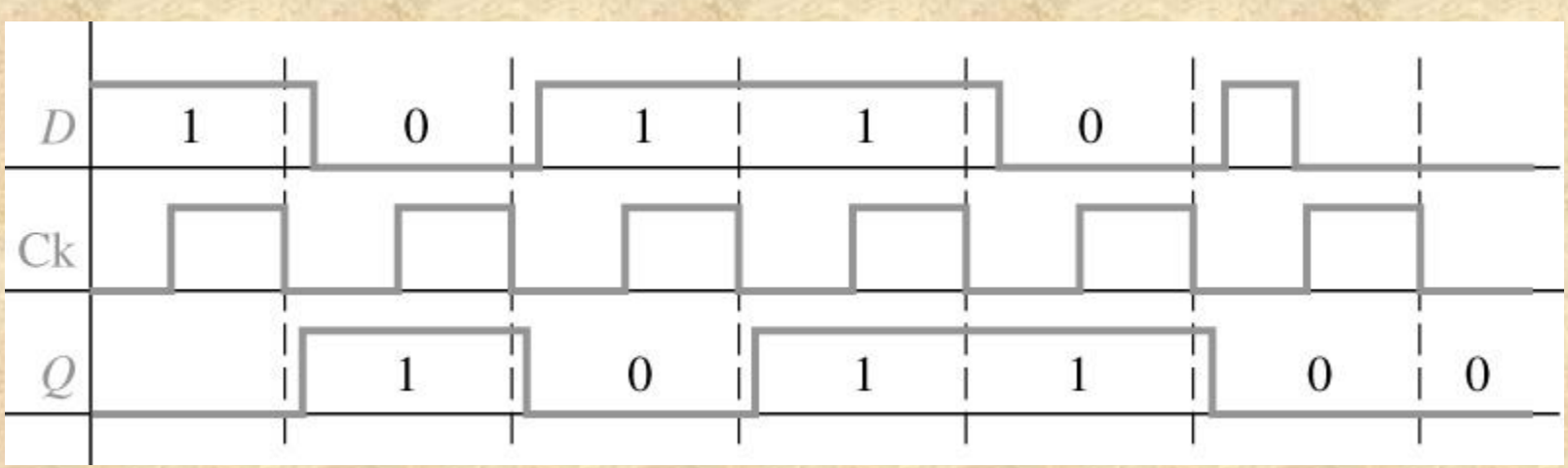
(a) Construction from two gated D latches



(b) Time analysis

Edge-Triggered D Flip-Flop (FF) (3/3)

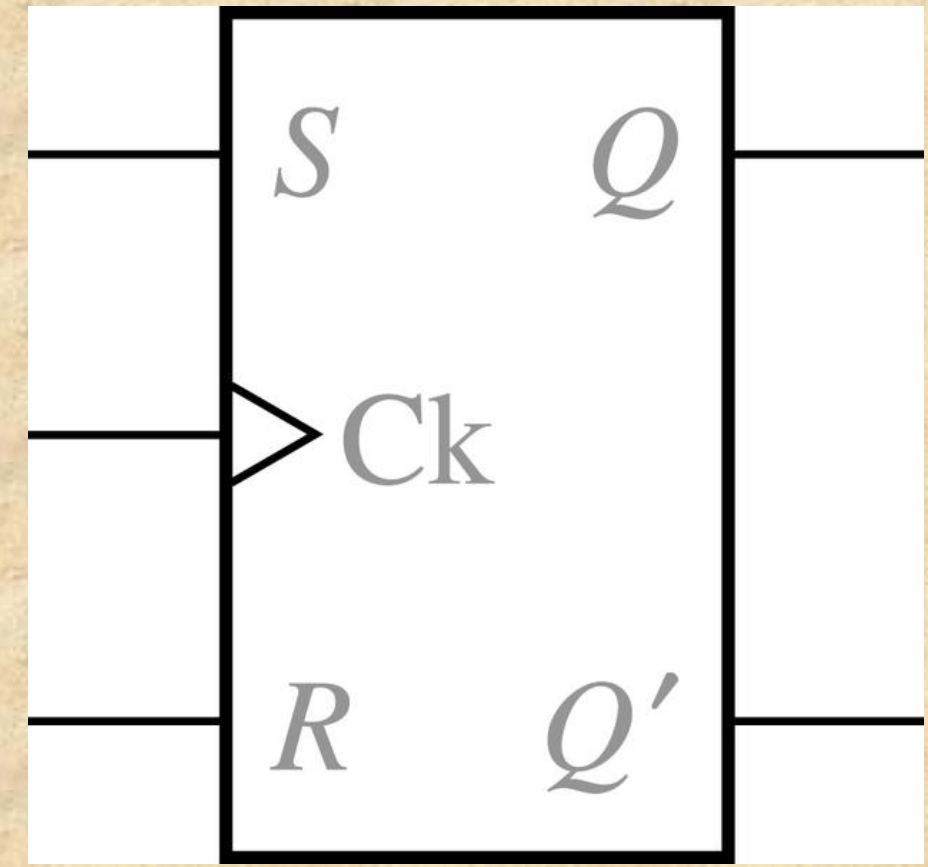
Note Q does not change during Ck=0.



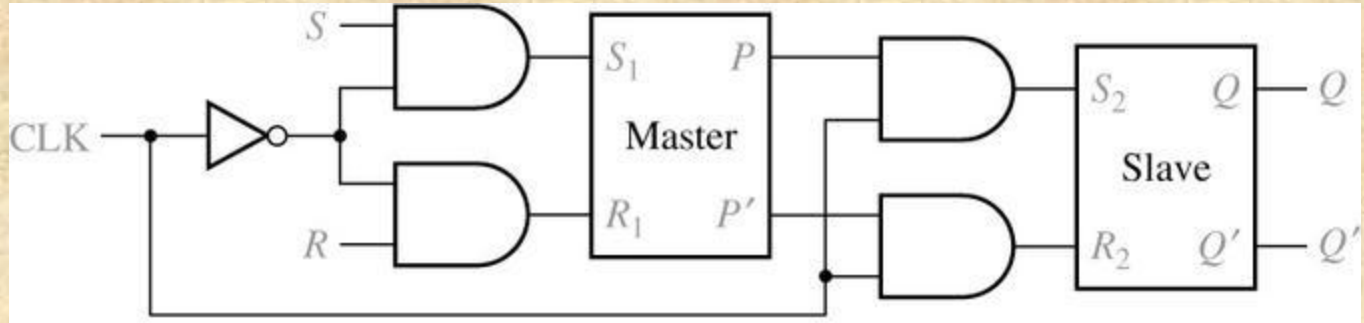
•Timing for D Flip-Flops (FF) (Falling-Edge Trigger)

S-R Flip-Flop (FF) (1/3)

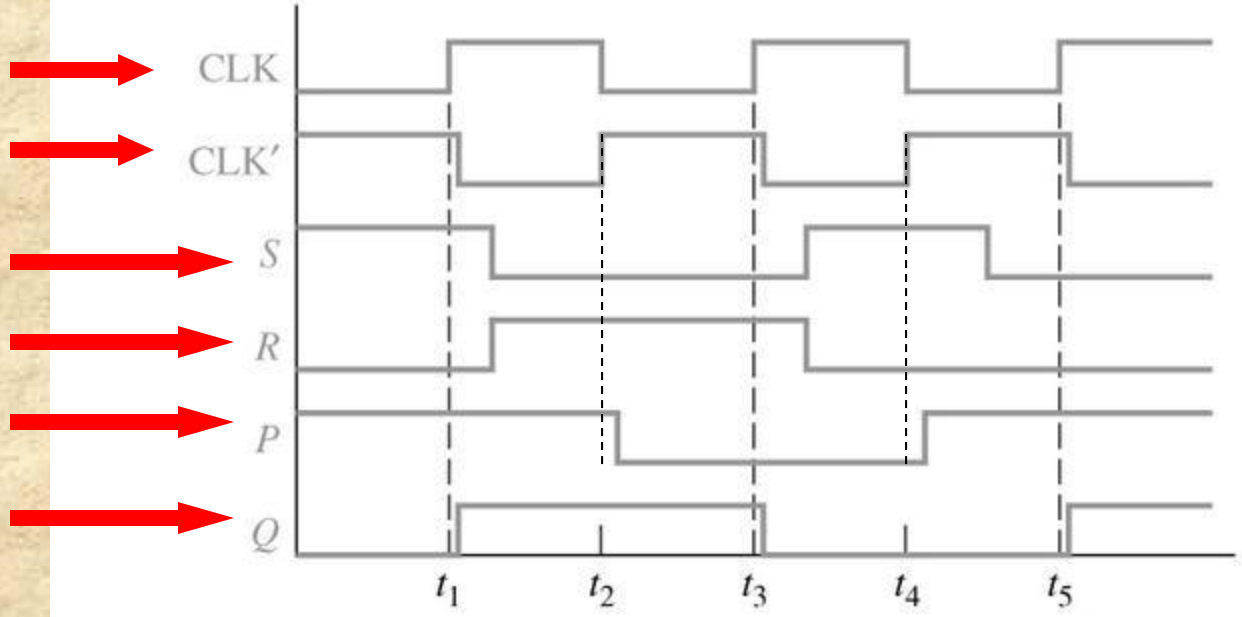
- Q^+ changes in response to the clock signal.



S-R Flip-Flop (FF) (2/3)



(a) Implementation with two latches



(b) Timing analysis

- When CLK=0
  - $P^+=S_1+R_1'P$
- When CLK=1
  - $Q^+=P$

S-R Flip-Flop (FF) (3/3)

- Why master-slave Flip-Flops (FFs)?
  - The master Flip-Flop (FF) holds the output for in the first half clock cycle.
  - When the slave Flip-Flop (FF) updates and outputs, the master is closed.
  - This mechanism guarantees that the final output changes only once in a clock cycle.

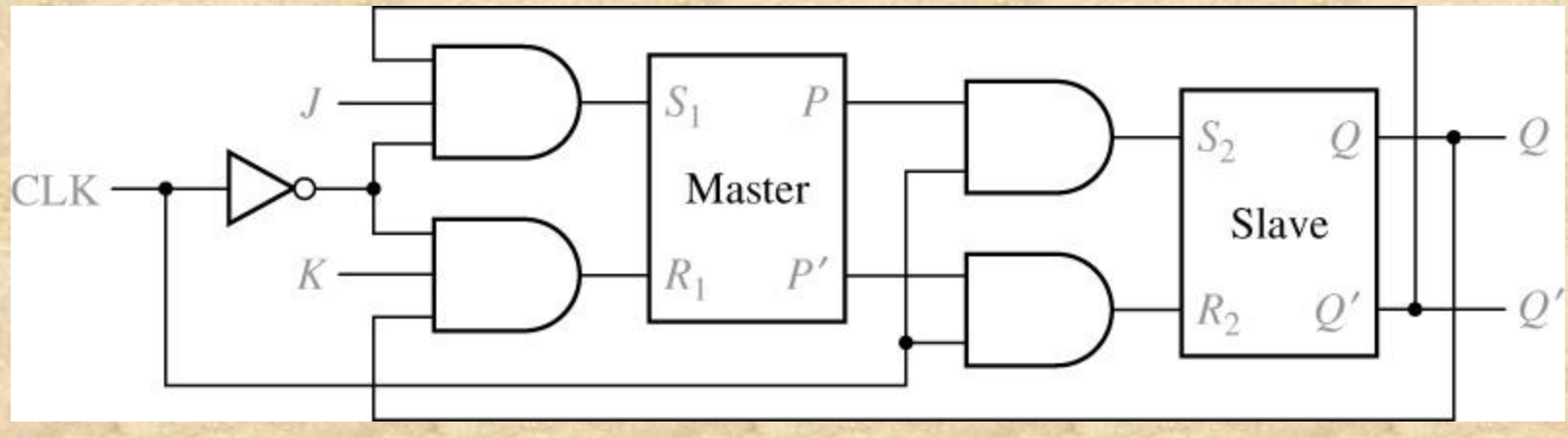
JK Flip-Flop (FF) (1/2)

- An extended version of the SR Flip-Flop (FF)
  - J corresponds to S
  - K corresponds to R

J	K	Q	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

JK can be 11. This configuration changes the state of Q.

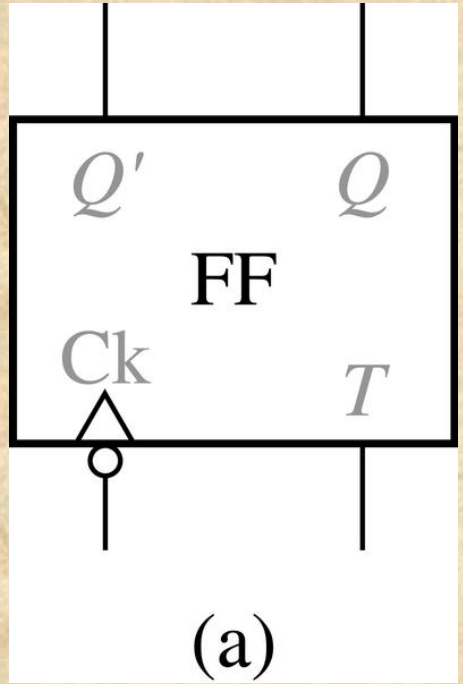
JK Flip-Flop (FF) (2/2)



- S1=J * Q' * CLK'
- R1=K * Q * CLK'
- S1 and R1 cannot be 1 at the same time.

T Flip-Flop (FF) (1/2)

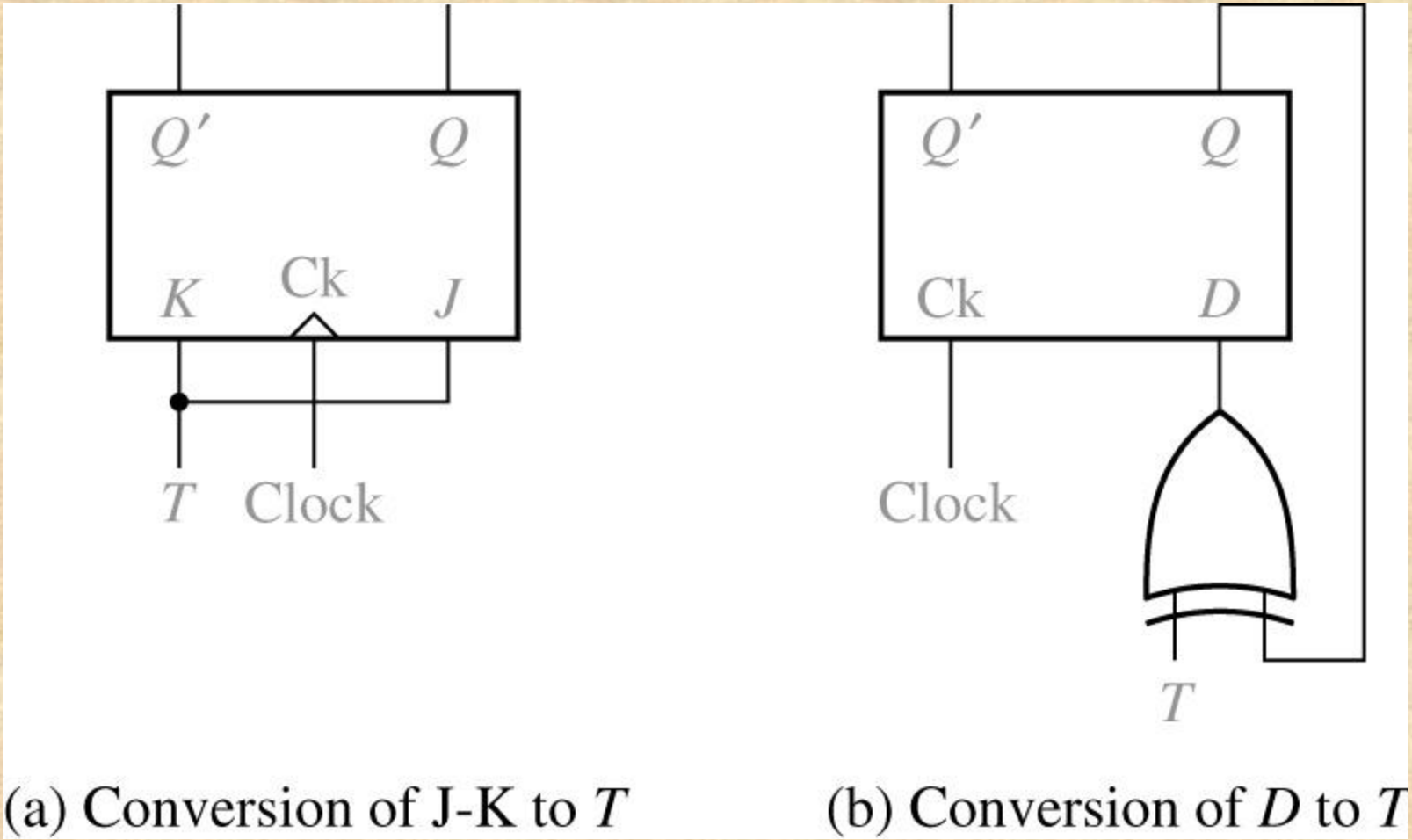
- T=0 → no state change
- T=1 → state changes



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

Q^+=T xor Q

T Flip-Flop (FF) (2/2)



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

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

Summary (1/8)

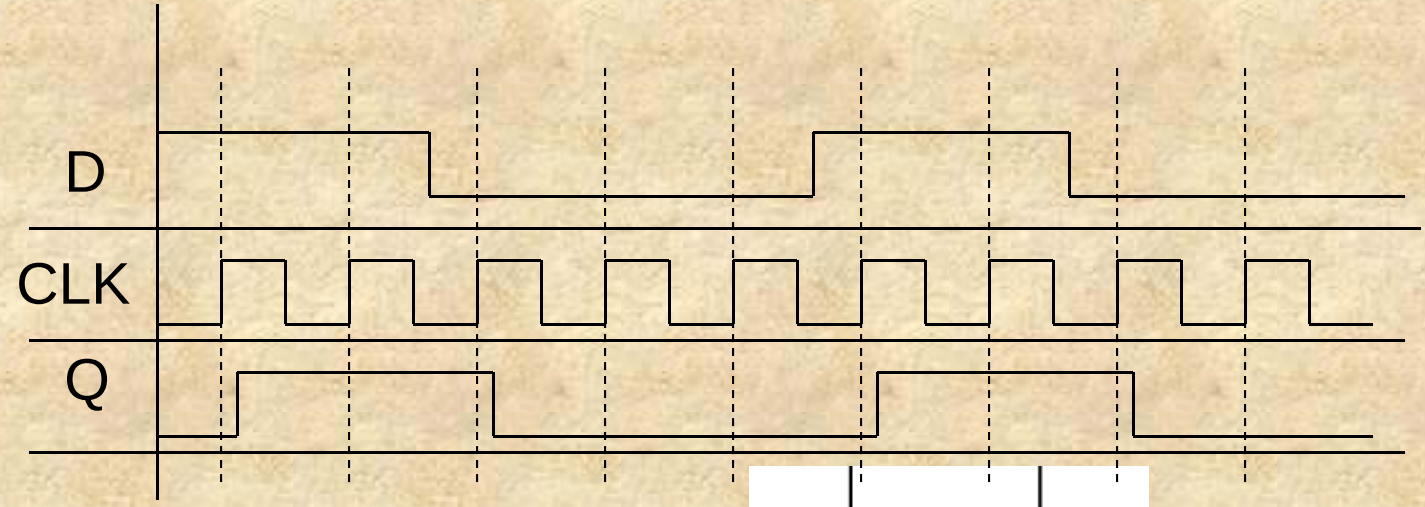
- All the Flip-Flops (FFs) and D latch are based on SR latch.
- SR latch can be described using
  - $Q^+=S+R'Q$
  - S=1 sets Q
  - R=1 resets Q
  - SR=00 keeps states (Q does not change.)

Summary (2/8) $Q^+=S+R'Q$

- Gated D latch
  - S=DG
  - R=D'G
  - When G=0, SR==00 ➔ state kept.
  - When G=1, Q=D
    - When D=0, SR=01 ➔ reset Q ➔ Q=0
    - When D=1, SR=10 ➔ set Q ➔ Q=1

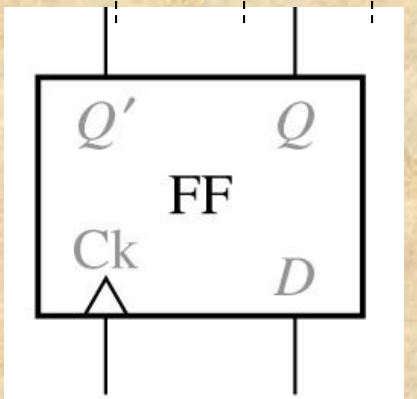
Summary (3/8) $Q^+=S+R'Q$

- When G is a clock signal, two gated D latches comprise an edge-triggered D Flip-Flop (FF)



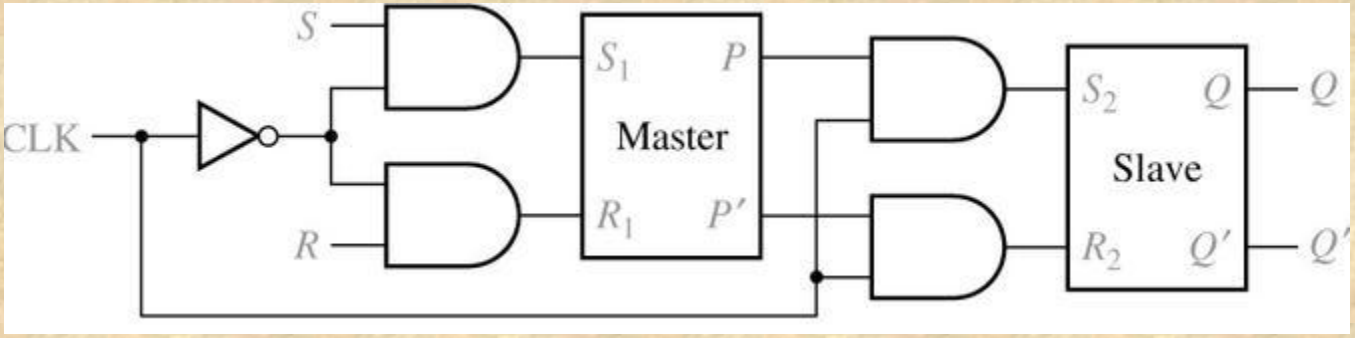
if(CLK is low)
Q does not change
else
Q becomes D shortly after the rising edge

- Note Q's waveform is not exactly the same with D's.



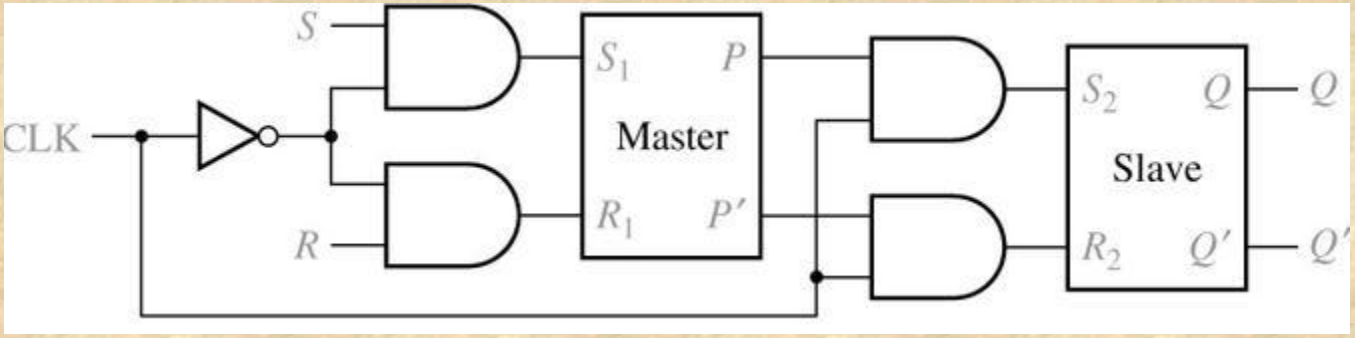
Summary (4/8) $Q^+=S+R'Q$

- SR Flip-Flop (FF)
  - Master and slave
  - When the master receives the input and updates, the slave is close.
  - When the slave outputs, the master does not respond to any input change.



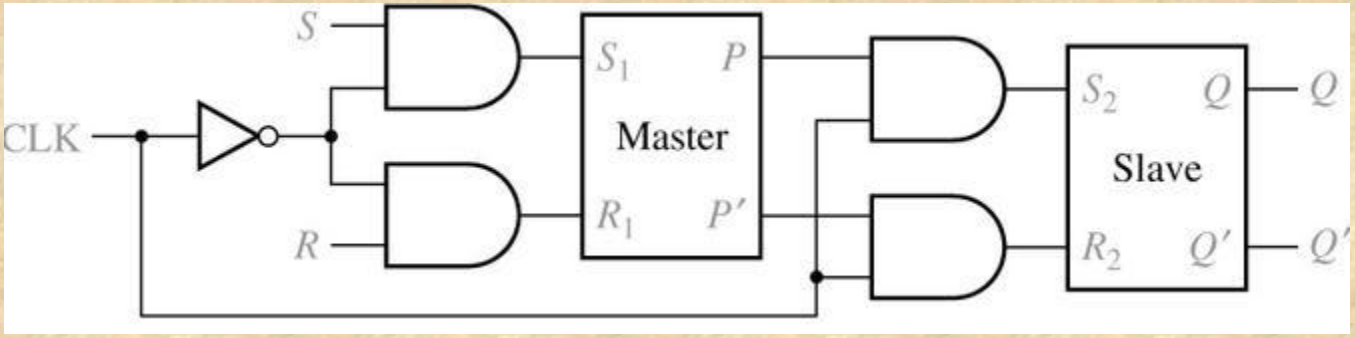
Summary (5/8) $Q^+=S+R'Q$

- SR Flip-Flop (FF) (cont)
  - When CLK is low,
    - $S_1R_1=SR$ ➔ $P^+ = S+R'P$, master is updated
    - $S_2R_2=00$ ➔ Q does not change



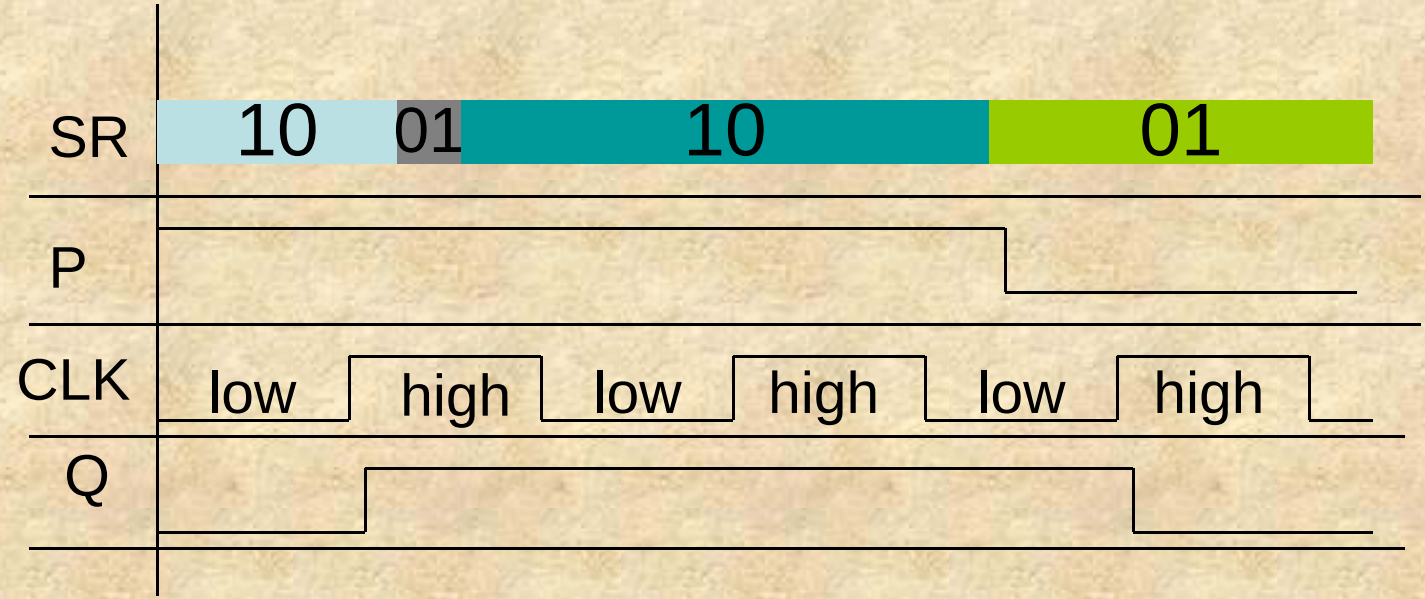
Summary (6/8) $Q^+=S+R'Q$

- SR Flip-Flop (FF) (cont)
  - When CLK is high,
    - $S_1R_1=00$ ➔ P does not change
    - master does not respond to inputs
    - $S_2R_2=PP'$ ➔ $Q^+=P+(P')'Q=P$



Summary (7/8) $Q^{+}=S+R'Q$

- SR Flip-Flop (FF) (cont)



- Master
- Slave

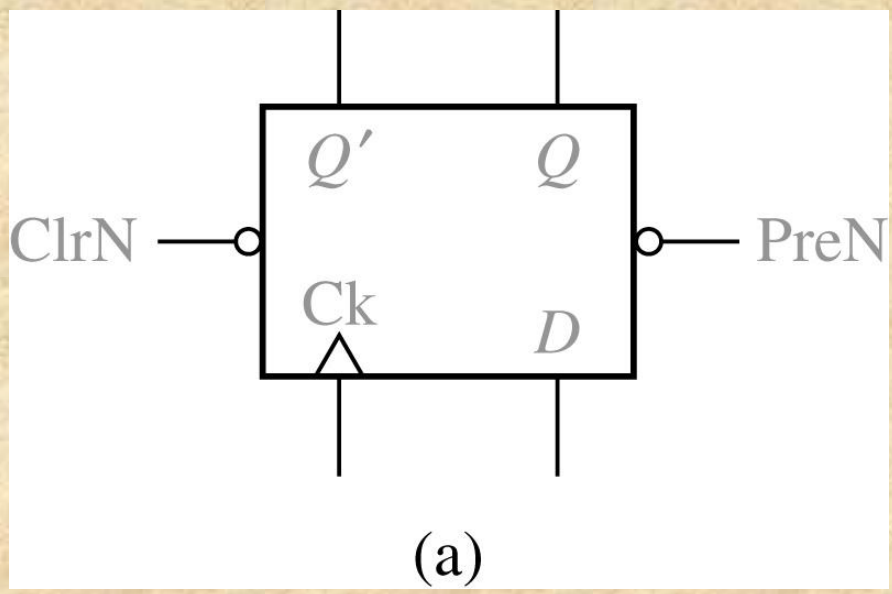
- The final output, Q, was not affected by 01 .

Summary (8/8) $Q^{+}=S+R'Q$

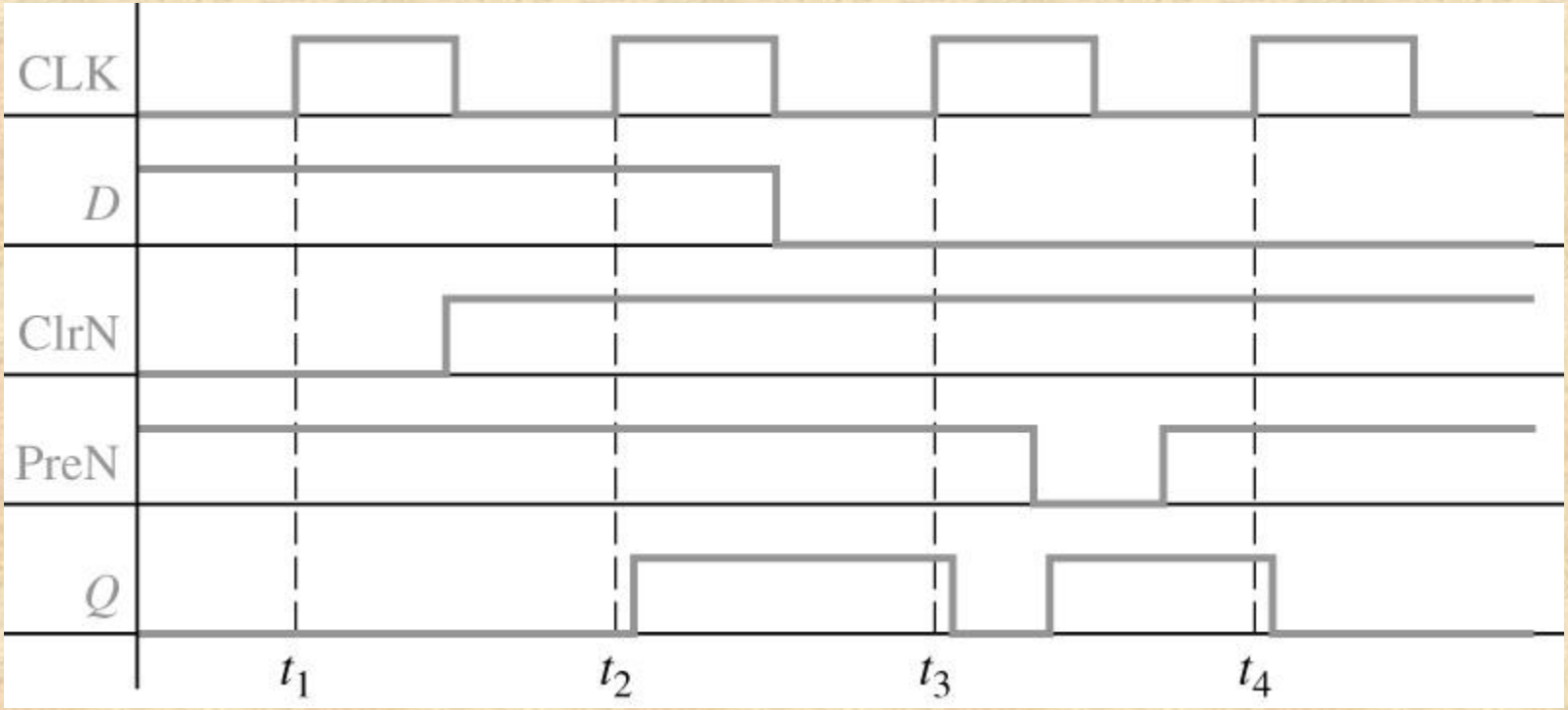
- JK Flip-Flop (FF)
  - Very similar to SR master-slave Flip-Flop (FF)
  - Except JK=11 inverts the output
- T Flip-Flop (FF)
  - T=1 ➔ inverts the output
  - T=0 ➔ keeps the same output

Flip-Flops (FFs) with Additional Inputs (1/3)

- Clear and Preset signals are two asynchronous signals and do not depend on CLK.



Flip-Flops (FFs) with Additional Inputs (2/3)



Flip-Flops (FFs) with Additional Inputs (3/3)

- Clock enable signal