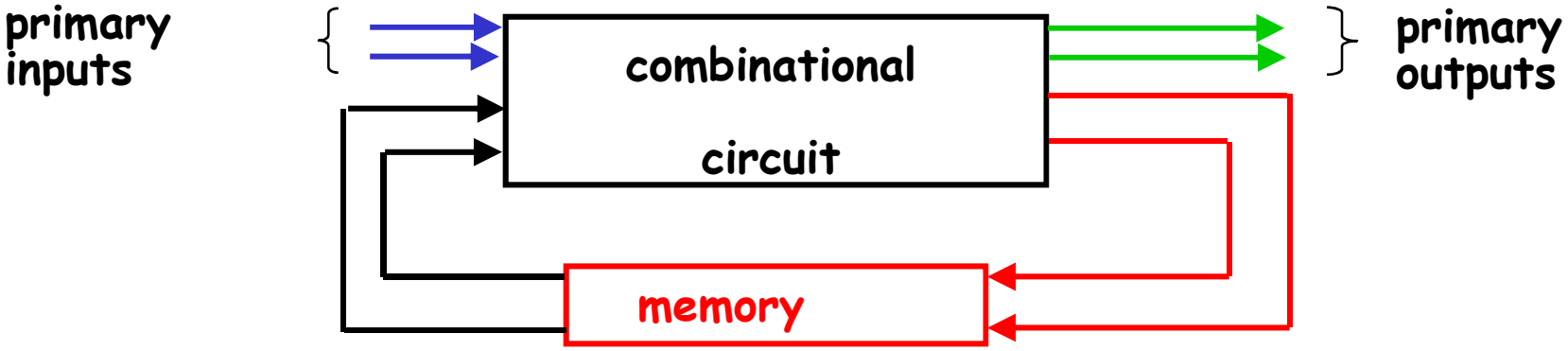
12 Introduction to Sequential Circuits

©Department of Computer Science
University of Victoria

SEQUENTIAL CIRCUITS

We now add memory!

- output depends on both current AND past inputs
- feedback



- the binary information stored in memory elements defines the STATE of the sequential circuit
- TIMING: if things are not in perfect sequence, we might be looking at the incorrect output
- NEXT STATE: depends on current state plus current inputs

Synchronous and Asynchronous

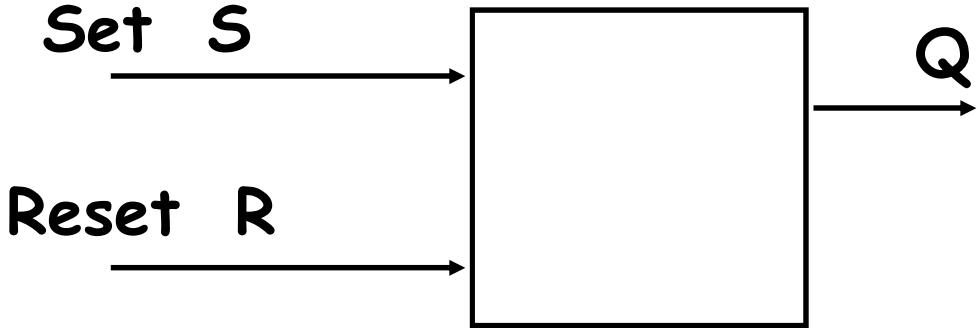
- **synchronous**
  - changes in output occur at discrete instants of time
  - there is a clock which synchronizes all circuit elements

- **asynchronous**
  - behaviour depends on order of input changes
  - does not wait for clock pulses
  - wanting speed, cannot afford to wait for clock signal, control signals from interrupts, etc.

Memory Elements

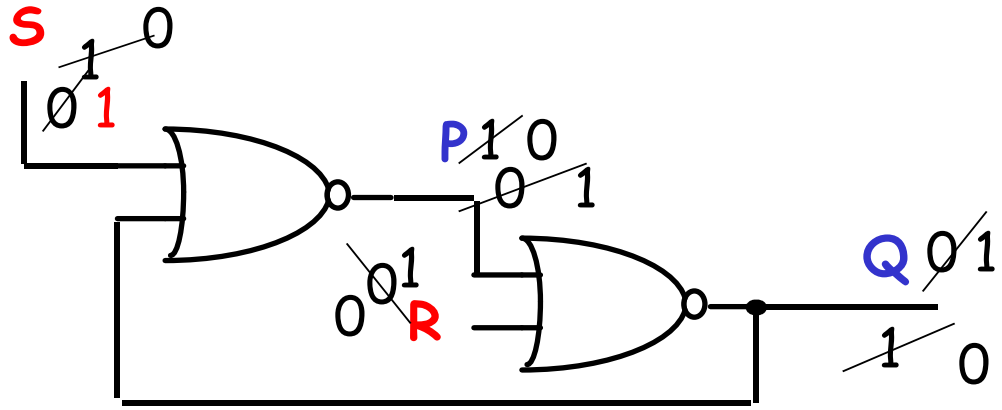
Flip-Flops, Latches,

➔ single bit storage devices



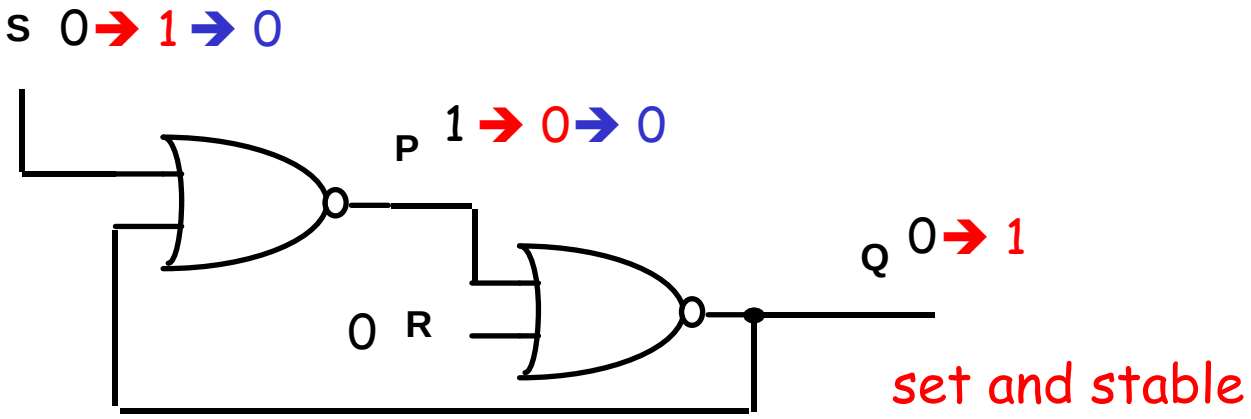
S	R	Q
0	0	unchanged
0	1	goes to 0
1	0	goes to 1

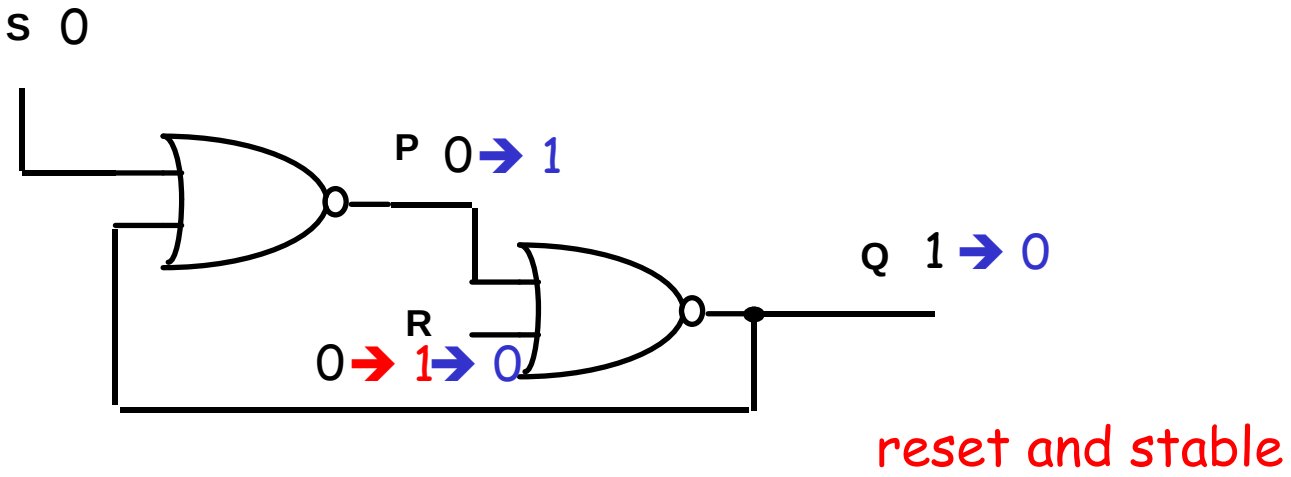
SET / RESET (R - S) Latch



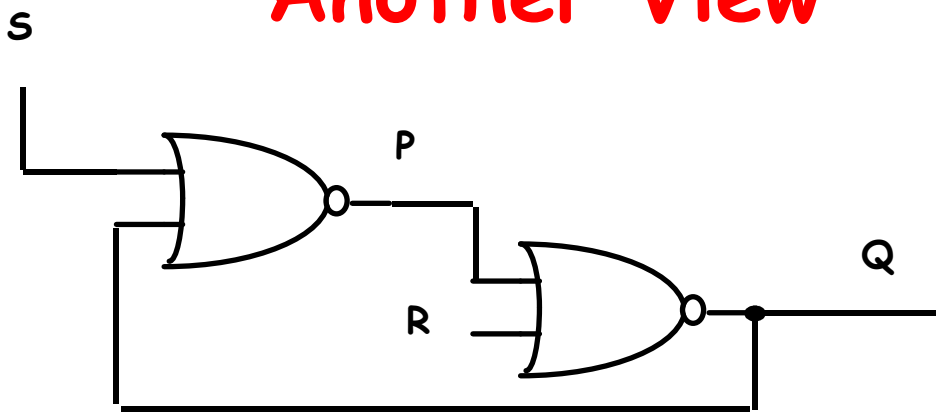
		nor
0	0	1
0	1	0
1	0	0
1	1	0

S	R	P	Q	
0	0	1	0	stable: Q = 0
1	0	0	1	set
0	0	0	1	stable: Q = 1
0	1	1	0	reset
0	0	1	0	stable: Q = 0


S 0 → 1 → 0
P 1 → 0 → 0
Q 0 → 1
0 R
set and stable


S 0
P 0 → 1
Q 1 → 0
R
0 → 1 → 0
reset and stable

Another View



IF S = R = 0, then

0 to a NOR input makes it an inverter

==> we have 2 inverters in a closed loop

==> whatever is in the system, stays there,

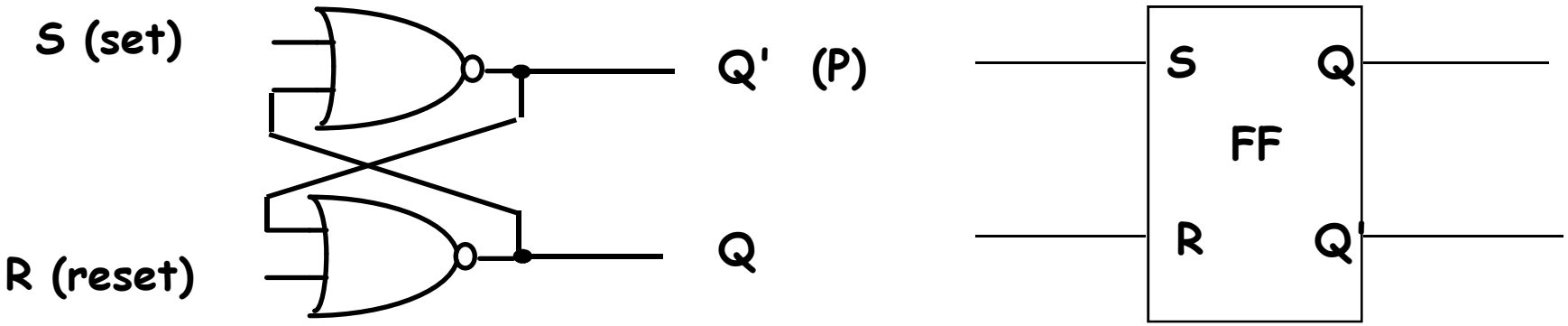
==> keeps circulating

==> we have memory

Details

restrict the inputs so that R=S=1 is not allowed (unstable),

➔ then P, Q are always complements of each other



S=1, R=0 , ➔ Q' to 0, ➔ Q to 1, ➔ FF is set

back to S =R = 0, keeps Q'= 0 and Q = 1

change R to 1 (S=0), ➔ Q to 0, ➔Q' to 1, ➔ FF is reset

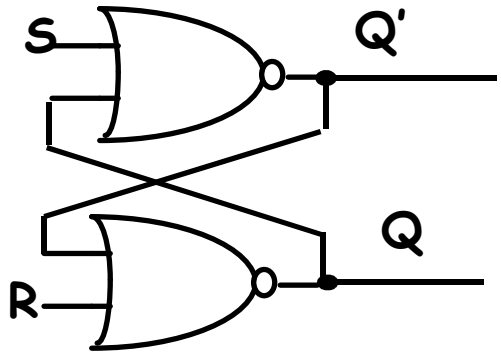
back to S =R = 0, keeps Q = 0 and Q'= 1

State Table

present state

next state

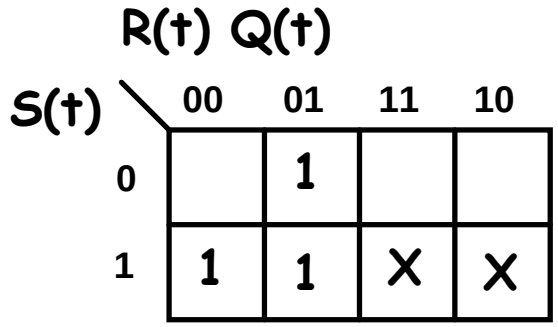
S(t)	R(t)	Q(t)	Q(t + 1) or Q+	
0	0	0	0	} stable
0	0	1	1	
0	1	0	0	} reset
0	1	1	0	
1	0	0	1	} set
1	0	1	1	
1	1	0	X	### NOT
1	1	1	X	### USED



Characteristic Equation

S(t)	R(t)	Q(t)	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

map for Q (t + 1) or Q+



$$Q(t+1) = S(t) + R'(t)\,Q(t)$$

Characteristic Equation is:

$$Q^+ = S + R'\,Q \qquad S\,R = 0$$

What does the characteristic equation mean?

Characteristic Equation

$$Q^+ = Q(t+1) = S + R'Q \qquad SR = 0$$

it means :

(1) ***next state = 1*** IF

(a) it is set to 1 by putting S to 1

or

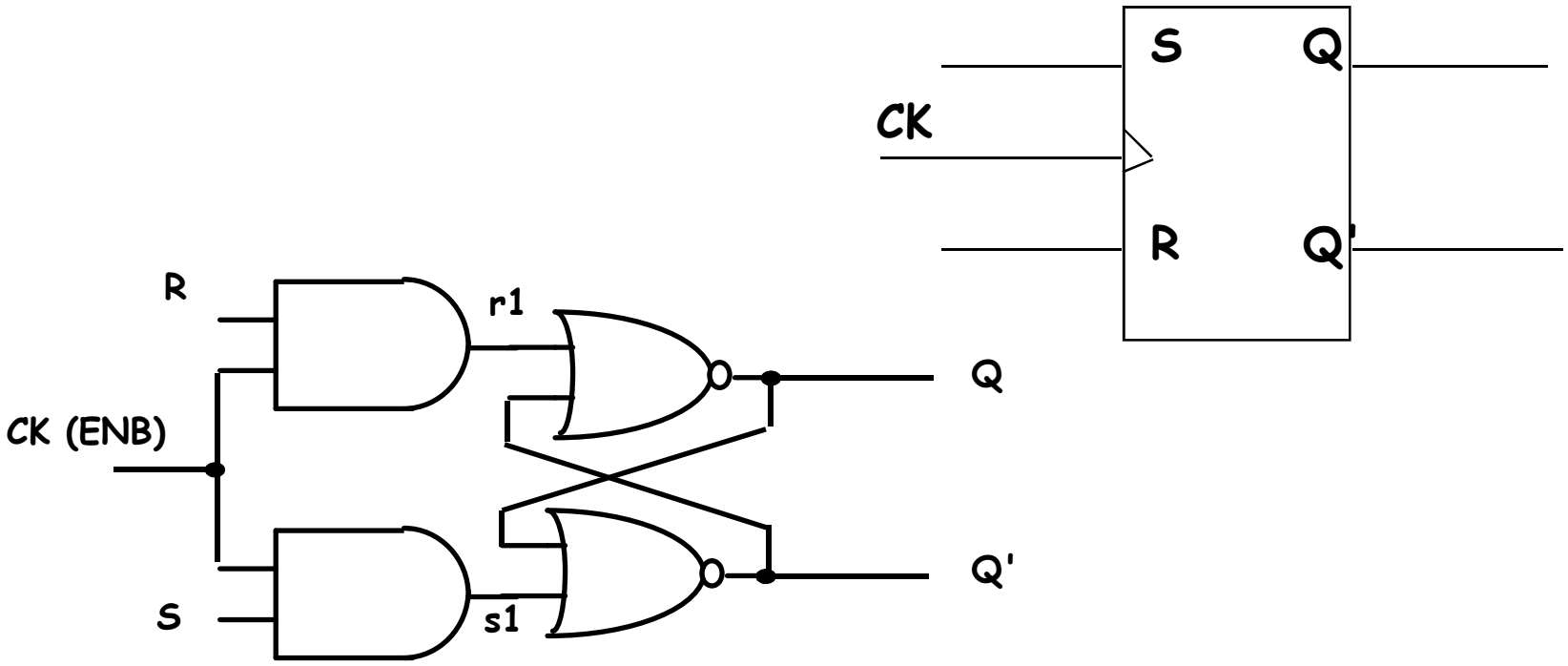
(b) present state =1 and it is not reset

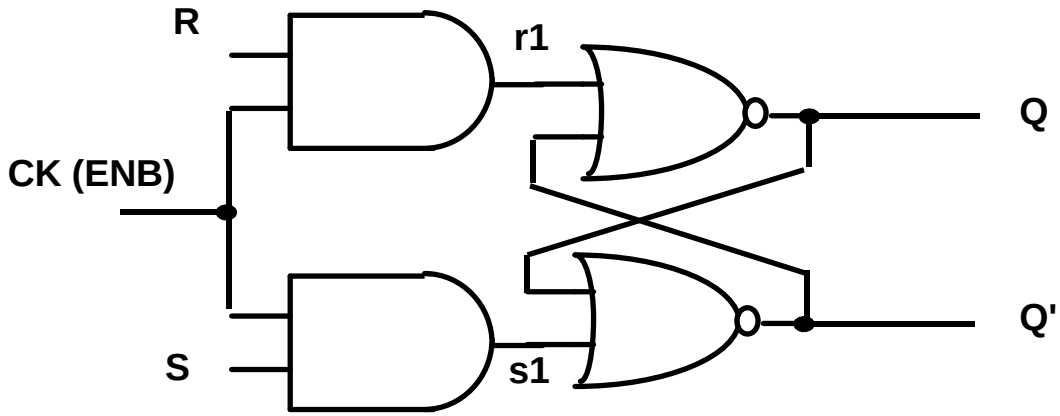
(2) SR=0 means S=R=1 is not allowed

CLOCKED R - S Latch

Level-Sensitive R - S Latch

- a **clock** or **enable** is necessary to maintain the circuit allowing changes at predetermined discrete moments

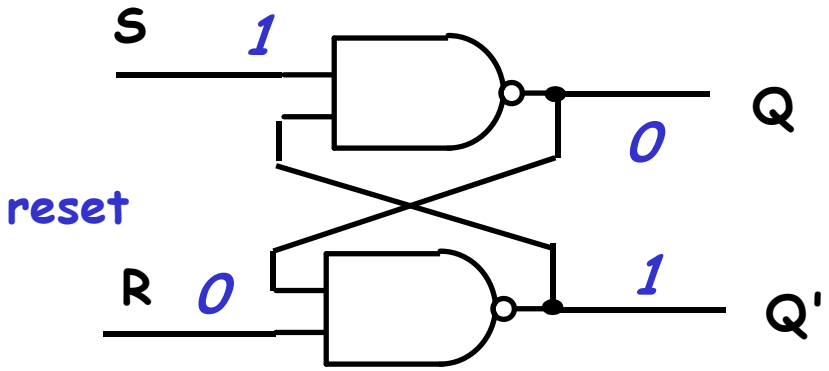


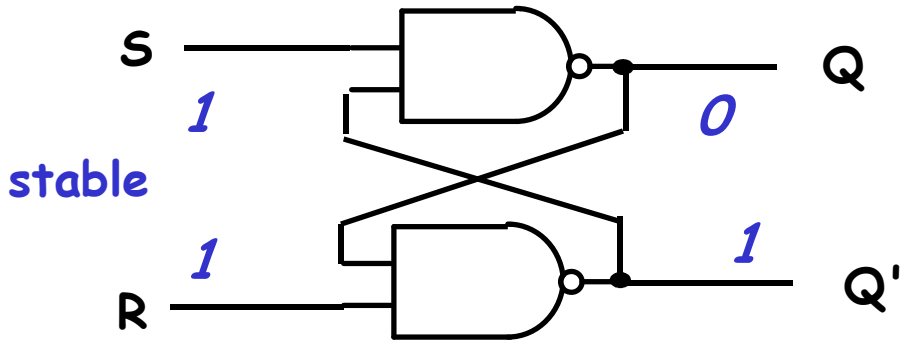


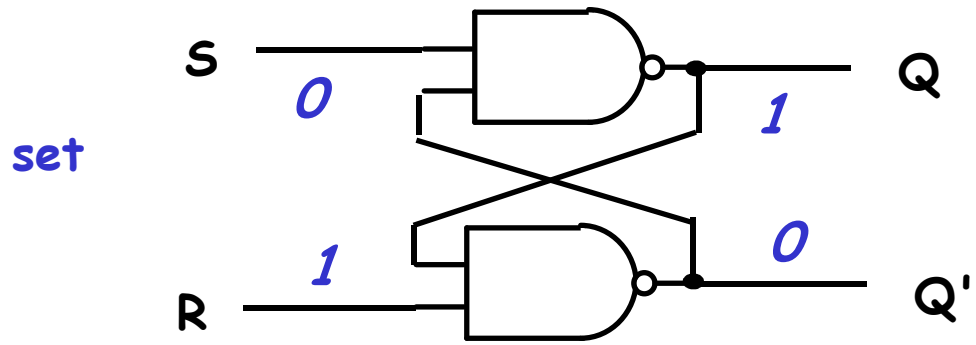
- outputs r1 and s1 are 0 as long as enable is 0
  - ➔ SR latch stable
- when enable goes to 1, the S and R inputs are allowed through to set/reset the basic latch
- S,R inputs can change while enable is 0, but latch is only activated when enable becomes 1

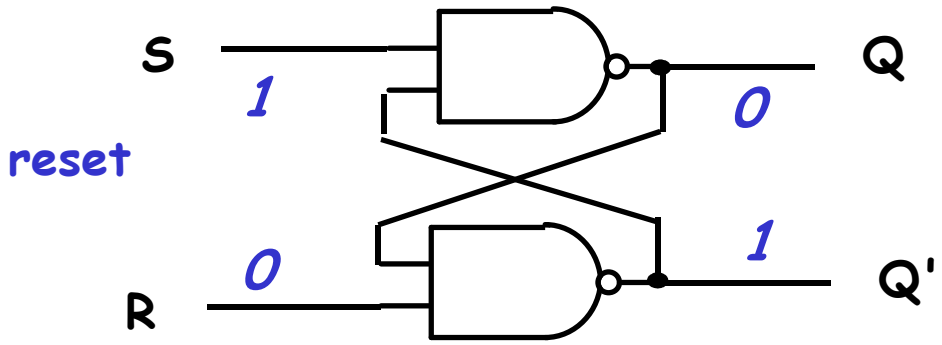
NAND implementation for SR

nand	0	1
0	1	1
1	1	0









opposite of NOR: LOW CONTROL

$\overline{S}\,\overline{R}$ Latch

SR TABLE (Nand)

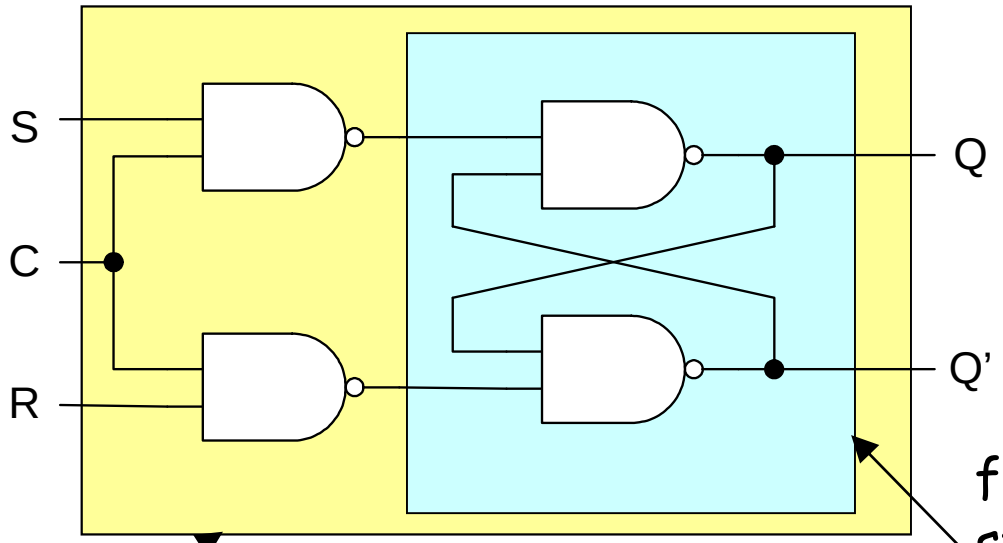
present state

next state

S(t)	R(t)	Q(t)	Q(t+1)	
0	0	0	X	not allowed
0	0	1	X	
0	1	0	1	set
0	1	1	1	
1	0	0	0	reset
1	0	1	0	
1	1	0	0	stable
1	1	1	1	

Look carefully at CLOCKED NAND FF and the characteristic equation

Common circuit, as used in the lab

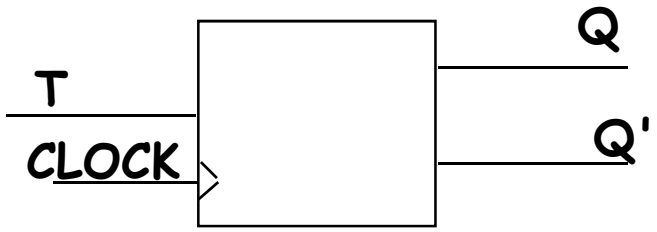


follows NAND state table

follows primary SR table

C	S	R	Q+
0	x	x	No change
1	0	0	No Change
1	0	1	Q=0
1	1	0	Q=1
1	1	1	Undefined

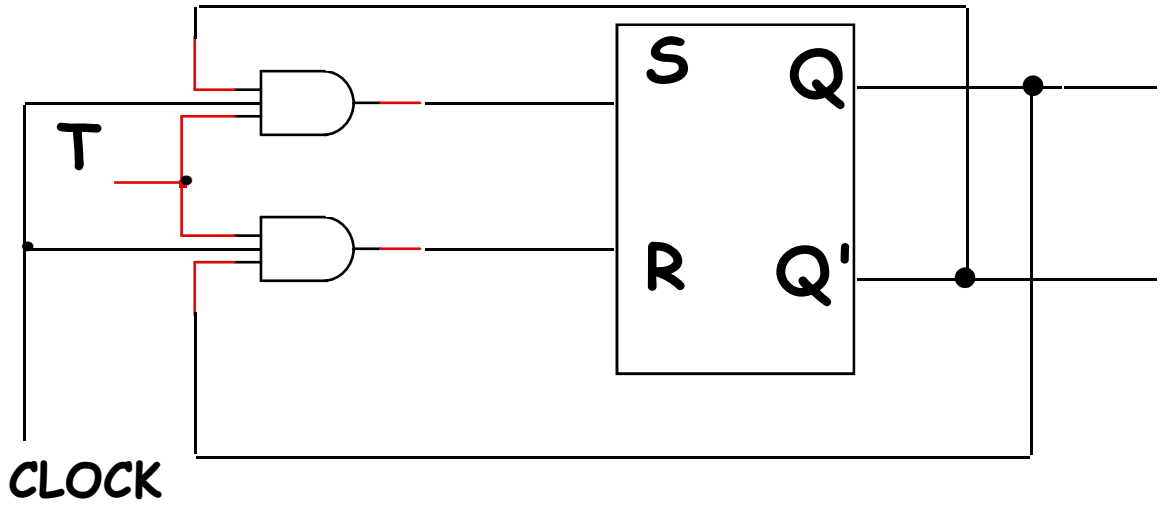
TOGGLE (T) Flip Flop



T=0, no change

T=1, flip state

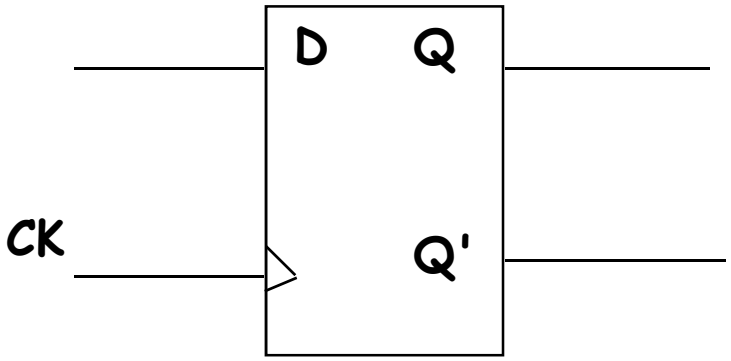
T : 0->1



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

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

D Flip Flop



D for DATA

➔transferring data into a FF (typical memory element)

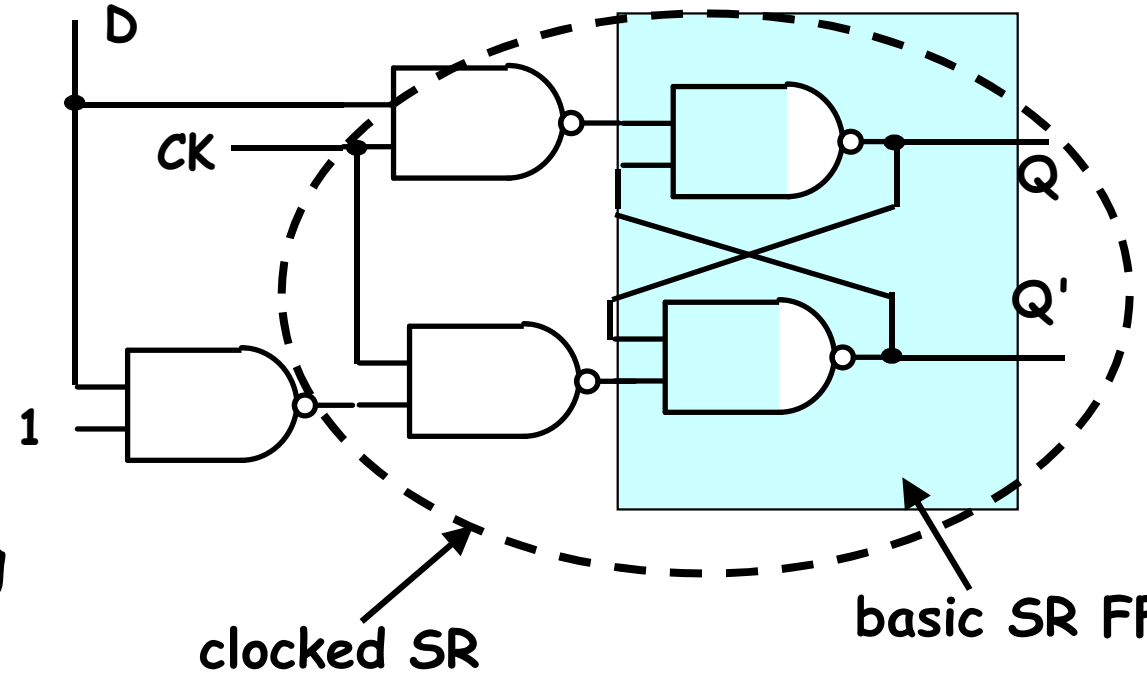
D	Q	Q^+
0	0	0
0	1	0
1	0	1
1	1	1

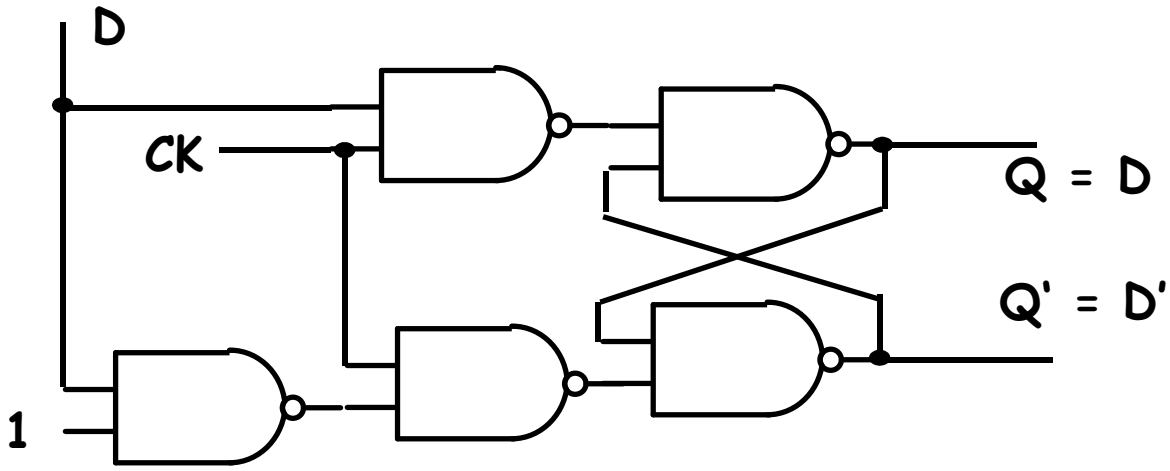
state Q^+ of FF after clock pulse is equal to the input D before the clock pulse

$$Q^+ = D$$

next state equation

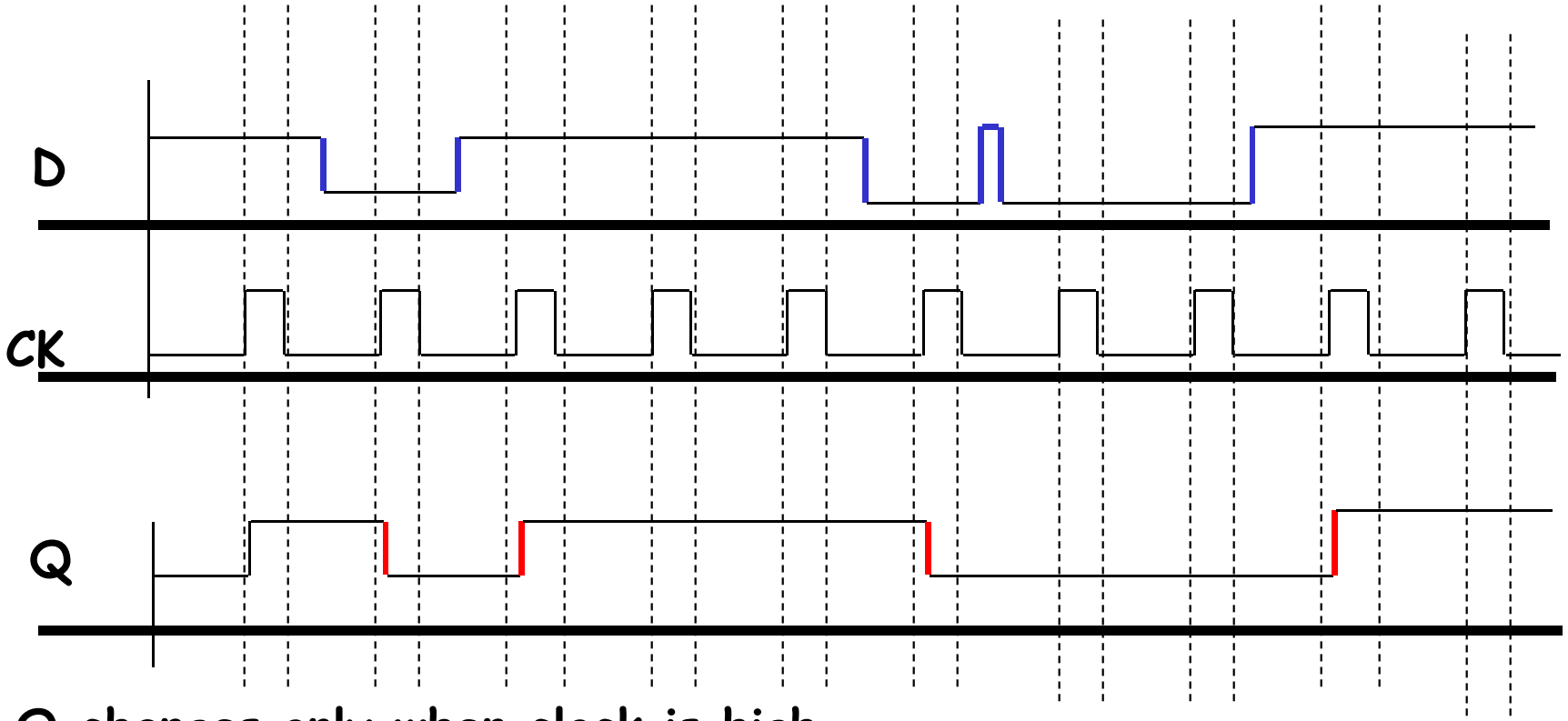
Removes the undefined state of S=R=1 by using inverter!!





$Q^+ = D$

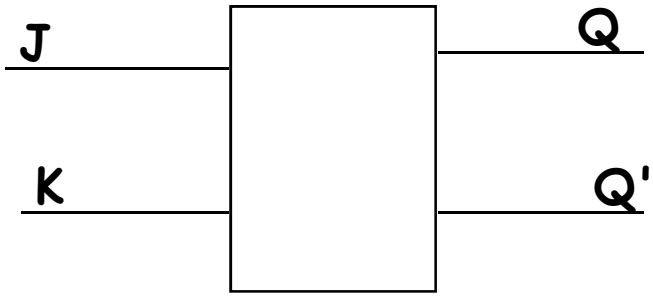
STABLE STATE



- Q changes only when clock is high
- here it shows that it changes after the clock has been high
- D gets changed when clock is low and can be changed many times without any effect (while clock is low)

J K Flip Flops

➔ they combine SR and T, eliminating invalid inputs



- when both J and K are 0, stable, no change
- when both J and K are 1, it acts as T (toggles)
- when either J or K are 1, but not both, it acts as SR

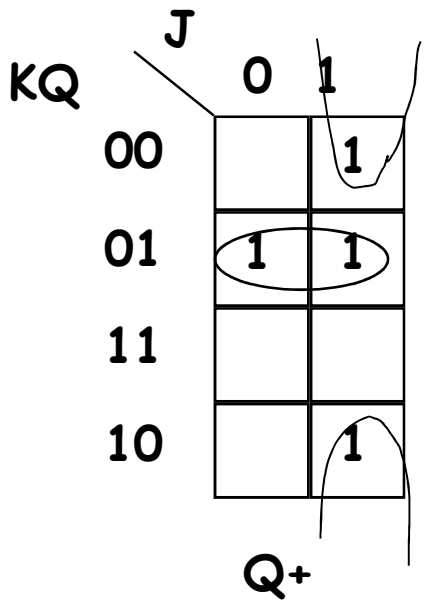
STATE TABLE FOR J K

present state

next state

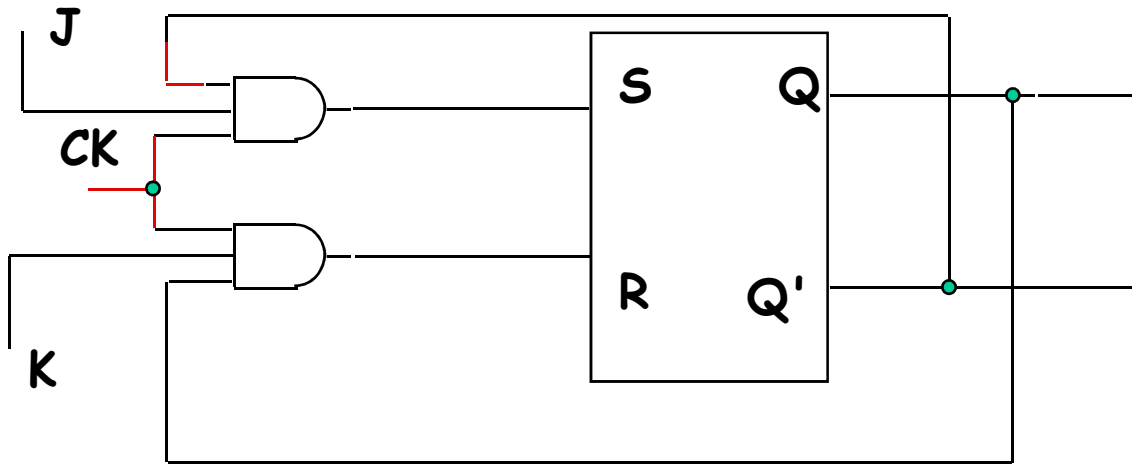
J	K	Q(t)	Q(t+1)	
0	0	0	0	stable
0	0	1	1	
0	1	0	0	reset
0	1	1	0	
1	0	0	1	set
1	0	1	1	
1	1	0	1	toggle
1	1	1	0	

Next State Equation - Characteristic Equation



$$Q^{+} = \overline{K}\, Q + J\, \overline{Q}$$

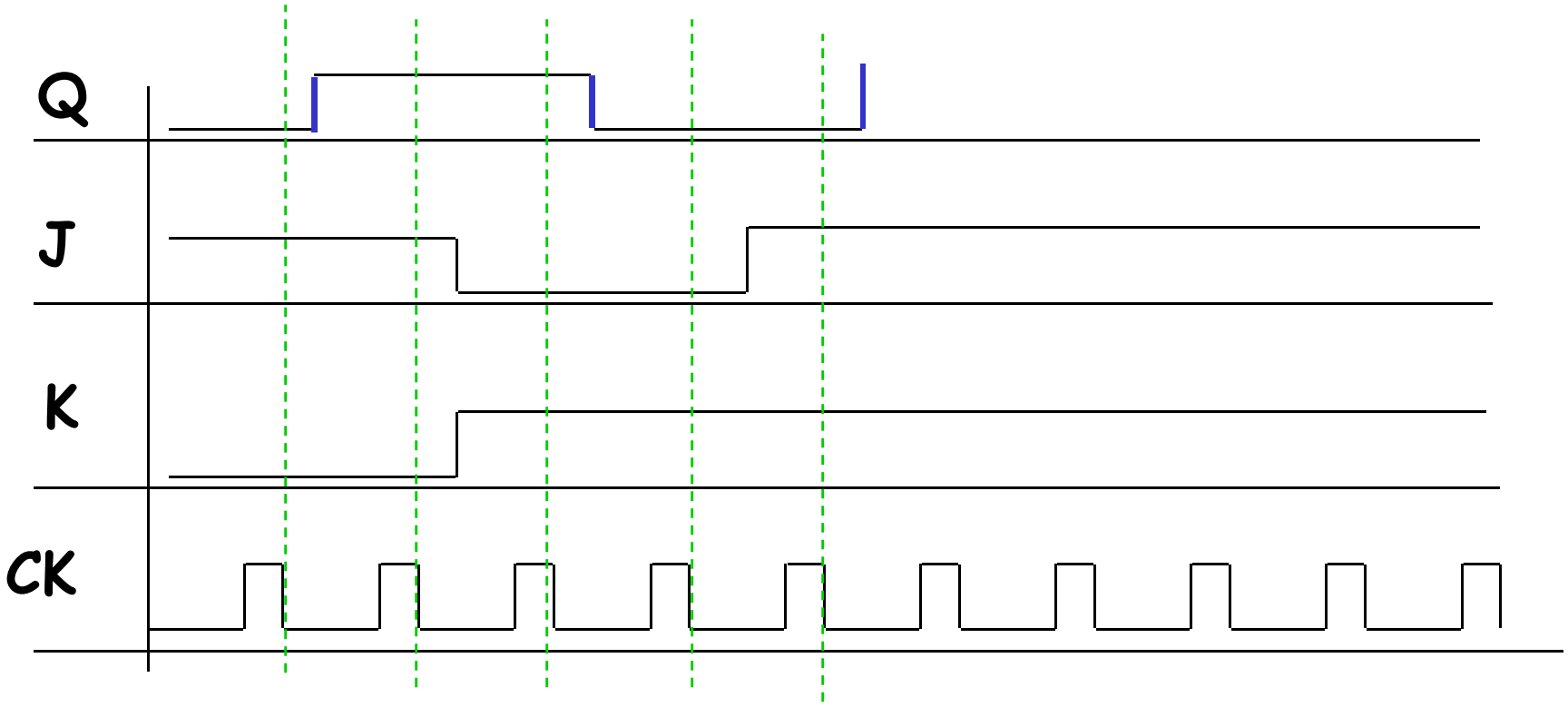
JK Flip Flop from SR FF (example)



- when CK = 0, nothing happens
- when CK=1 and Q = 0, then R = 0

  and S = J ➔ it will set or stay stable
- when CK=1 and Q = 1, then S = 0

  and R = K ➔ it will reset or stay stable
- when J = K = 1, it toggles (flips state)



- state Q changes short time after trailing edge of clock
- change in state always initiated by clock pulse, not by changes in J or K
- if clock remains 1 while J=K=1, it keeps flipping
- clock must have time duration shorter than gates propagation when used in feedback loops (difficult)

  (use master-slave or edge triggered FF - later)

Characteristic Equations

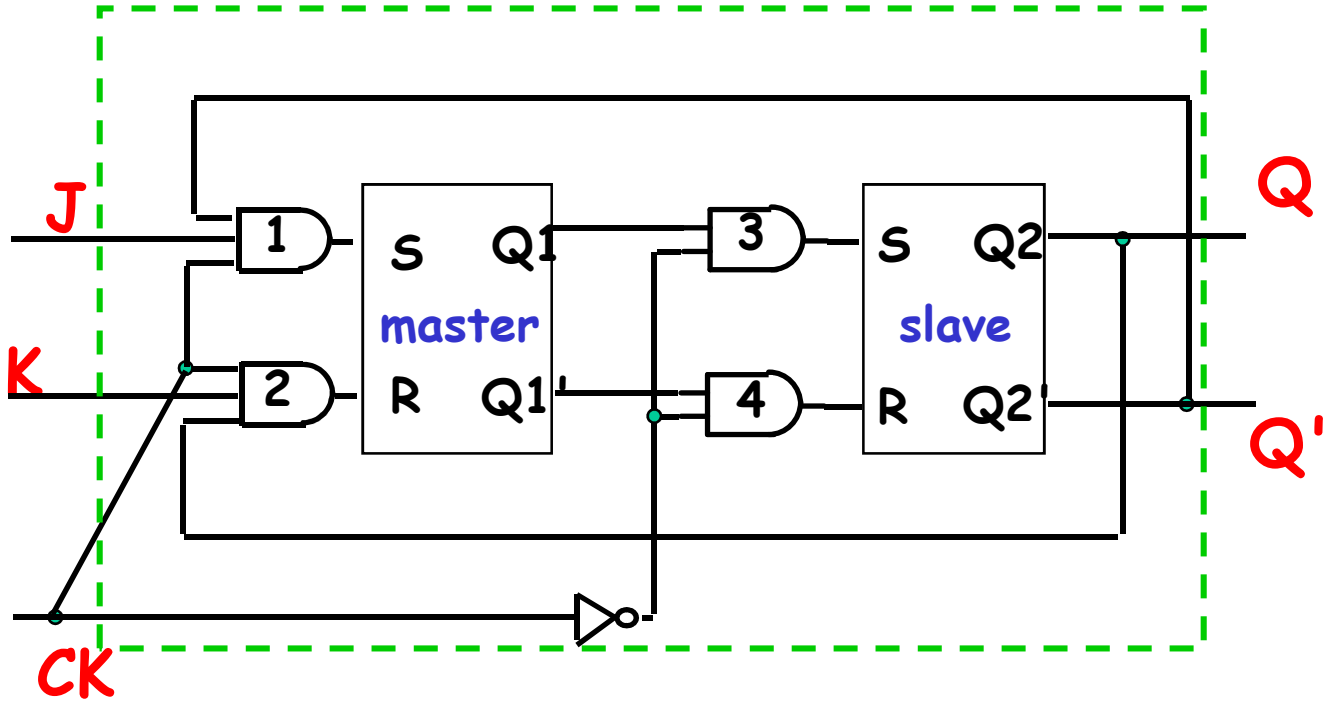
S-R : $Q^+ = \overline{R}\,Q + S \quad (SR = 0)$

T : $Q^+ = \overline{T}\,Q + T\,\overline{Q}$

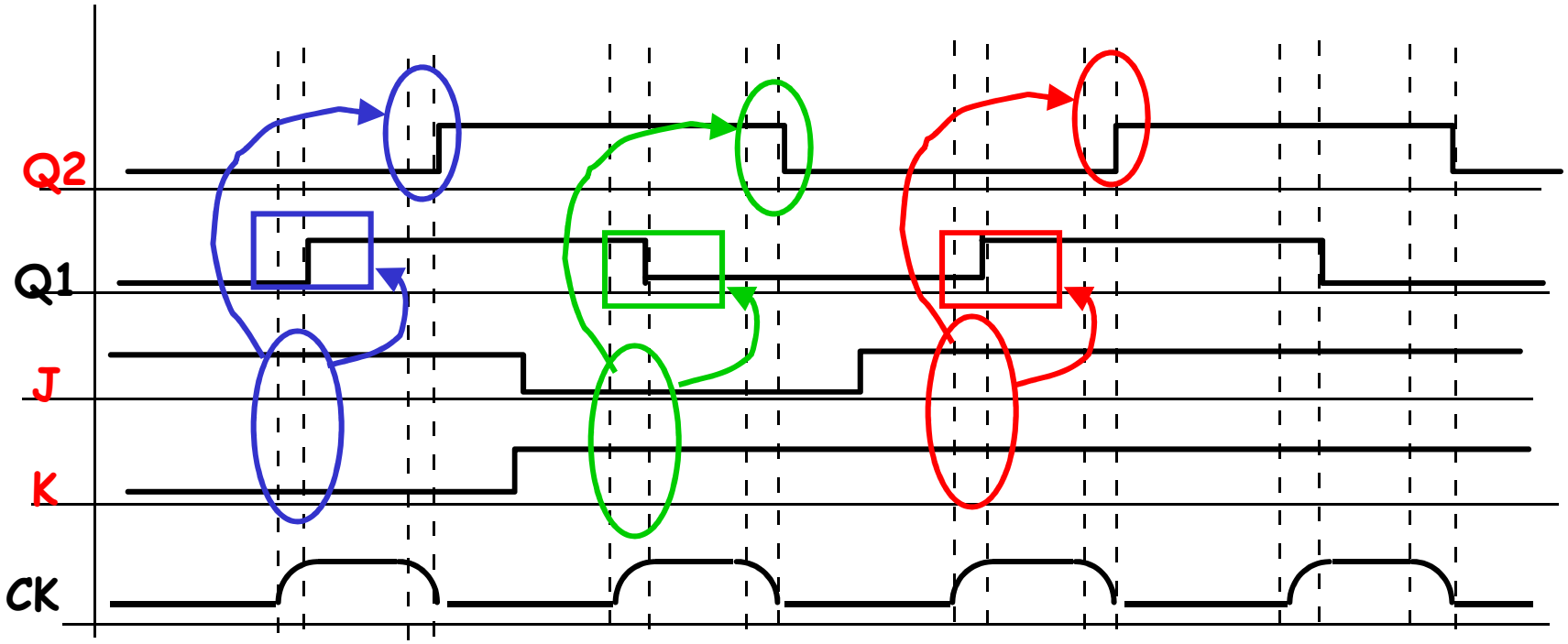
J-K : $Q^+ = \overline{K}\,Q + J\,\overline{Q}$

D : $Q^+ = D$

Master-Slave J-K FF

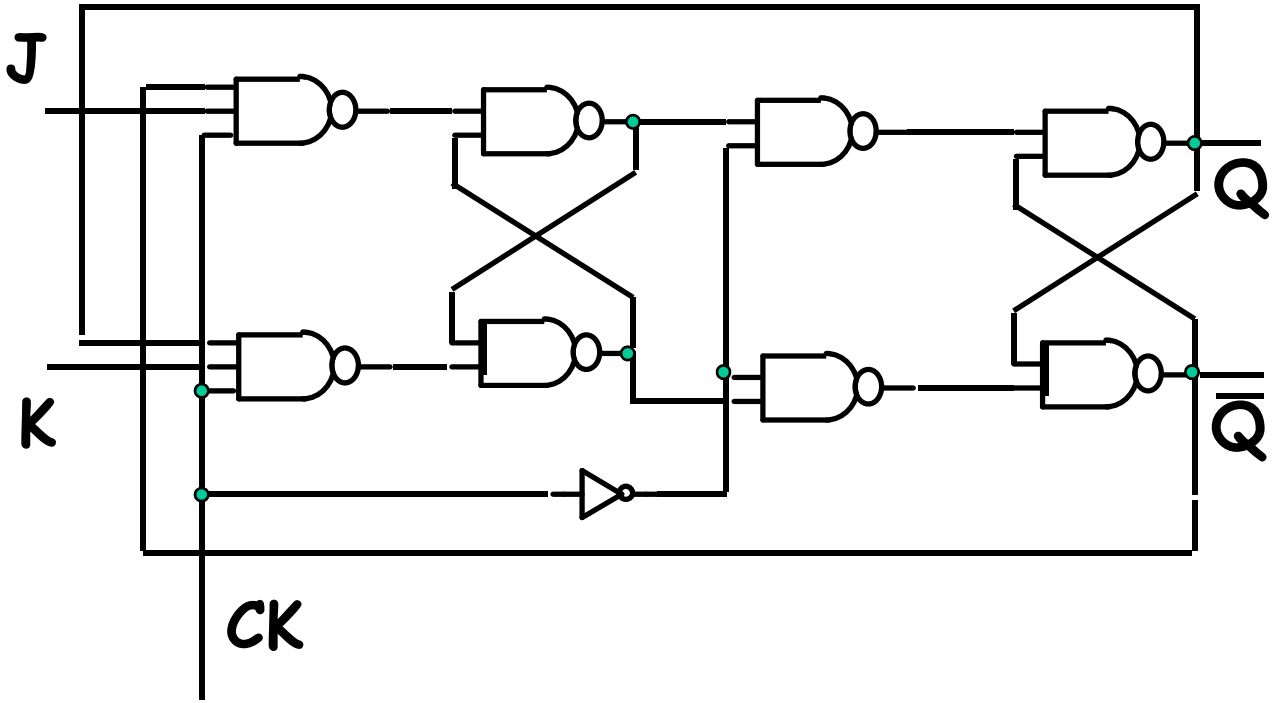


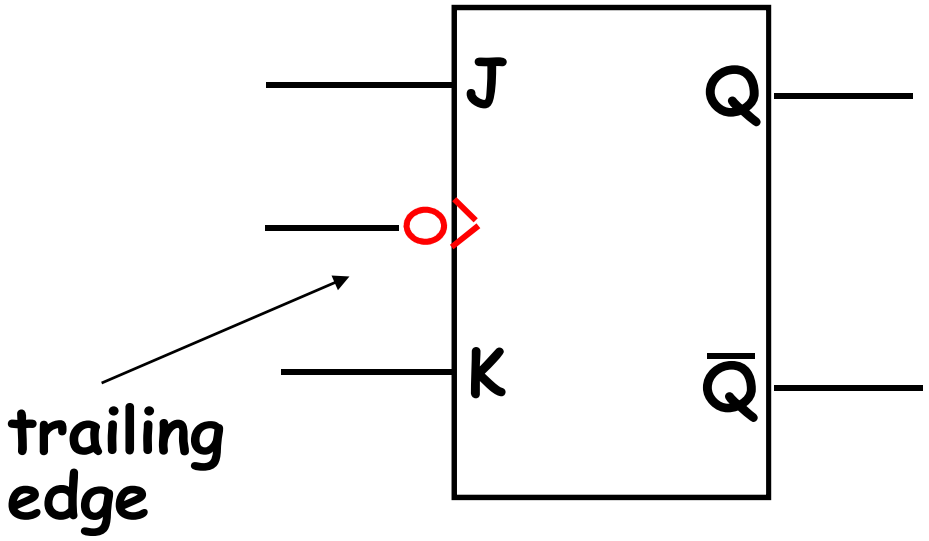
- when CK=1, only gates 1 & 2 operate, setting or resetting master FF (Q1)
- when CK=0, only gates 3 & 4 operate, setting or resetting slave FF (Q2)
- one FF is always isolated and maintains state



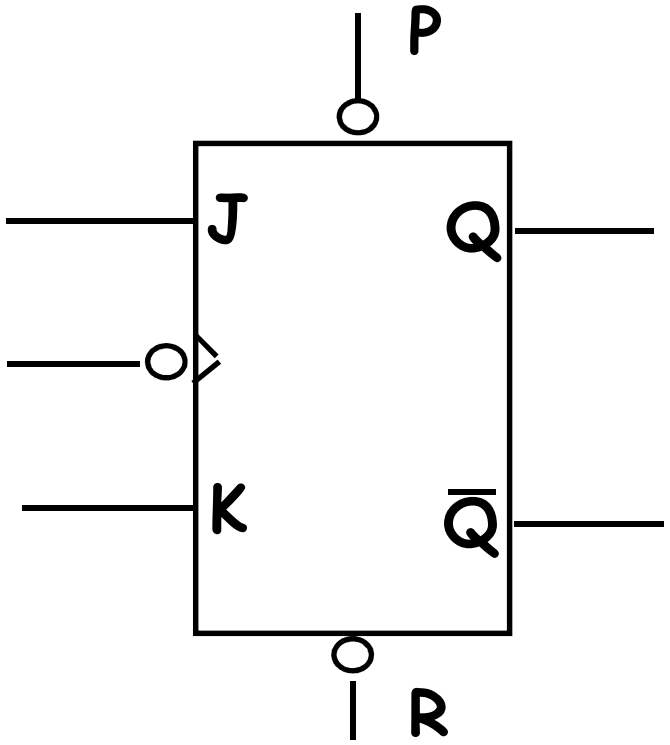
Q1 changes at *leading edge* ,Q2 changes *trailing edge*

- master operates at *leading edge* of clock
- slave operates at *trailing edge* of clock
- any changes in J,K primary inputs must occur when the master is isolated (clock=0)
- outputs are read after trailing edge


J
K
CK
Q
$\overline{Q}$


J
K
Q
$\overline{Q}$
trailing
edge

Direct Inputs



J K flip flop with preset and clear

At power up or at reset, all or part of a sequential circuit usually is initialized to a known state before it begins operation

This initialization is often done outside of the clocked behavior of the circuit, i.e., asynchronously

Latch Problems

- The change in the flip-flop output is delayed by the pulse width which makes the circuit slower
- S and/or R may change while C = 1
  - This behavior is called *1s catching*

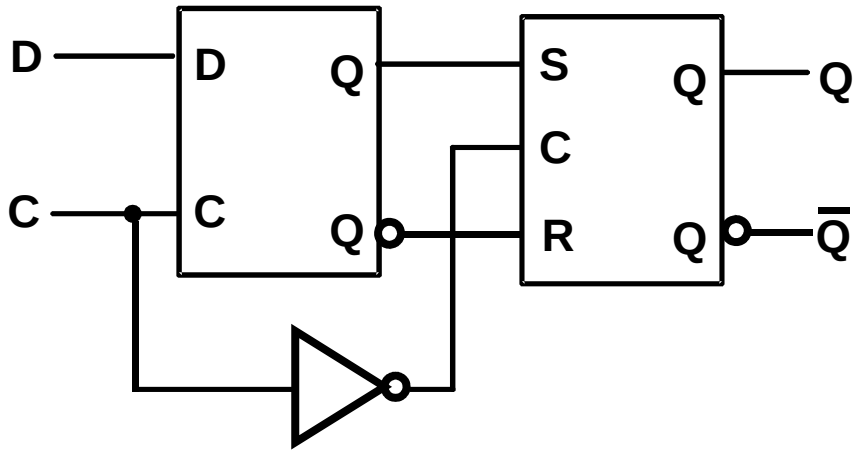
Flip-Flop Solution

- Use edge-triggering instead of latches or master-slave
- An *edge-triggered* flip-flop ignores the pulse while it is at a constant level and triggers only during a *transition* of the clock signal

Edge-Triggered D Flip-Flop

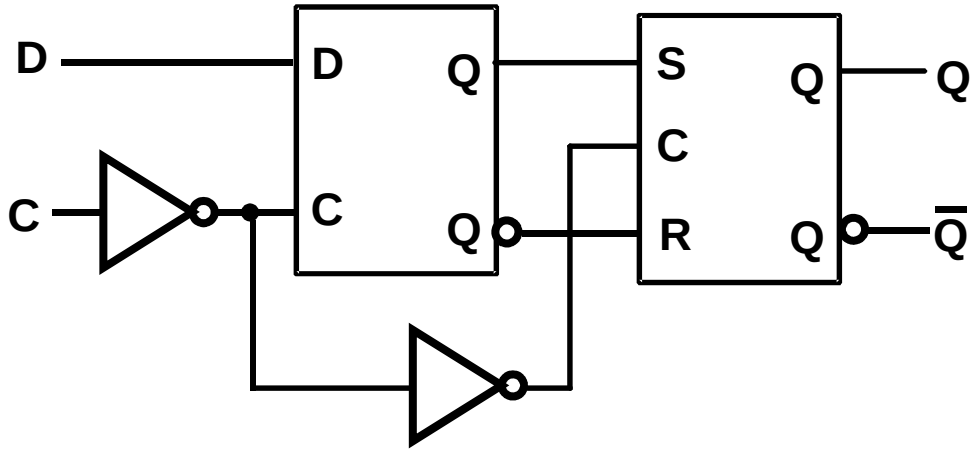
The edge-triggered
D flip-flop is similar to
a master-slave D flip-flop

- It can be formed by:
  - Replacing the first clocked S-R latch with a clocked D latch or
  - Adding a D input and inverter to a master-slave S-R flip-flop



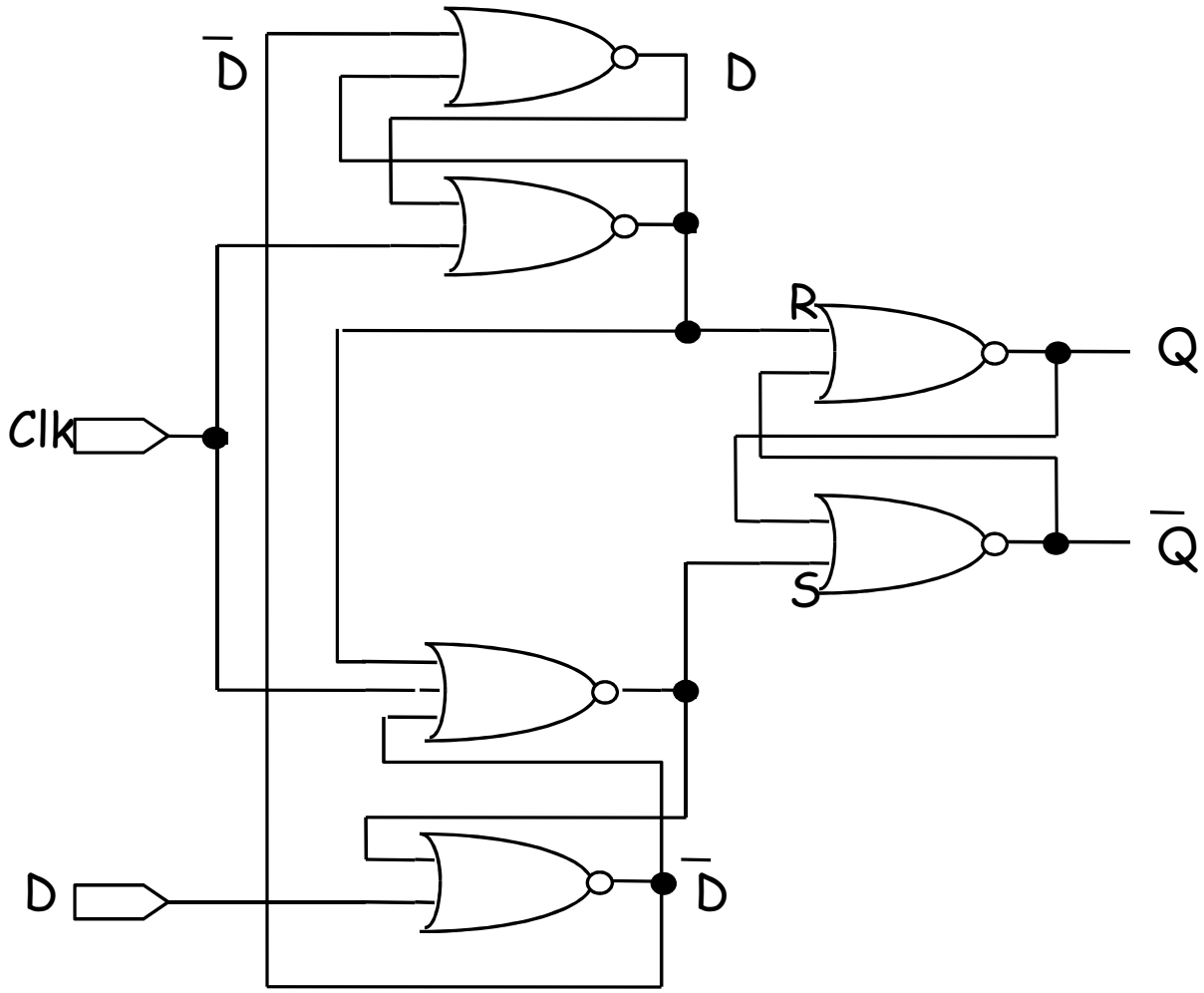
The change of the D flip-flop output is associated with the *negative edge* at the end of the pulse➔ called a *negative-edge triggered* flip-flop

Positive-Edge Triggered D Flip-Flop

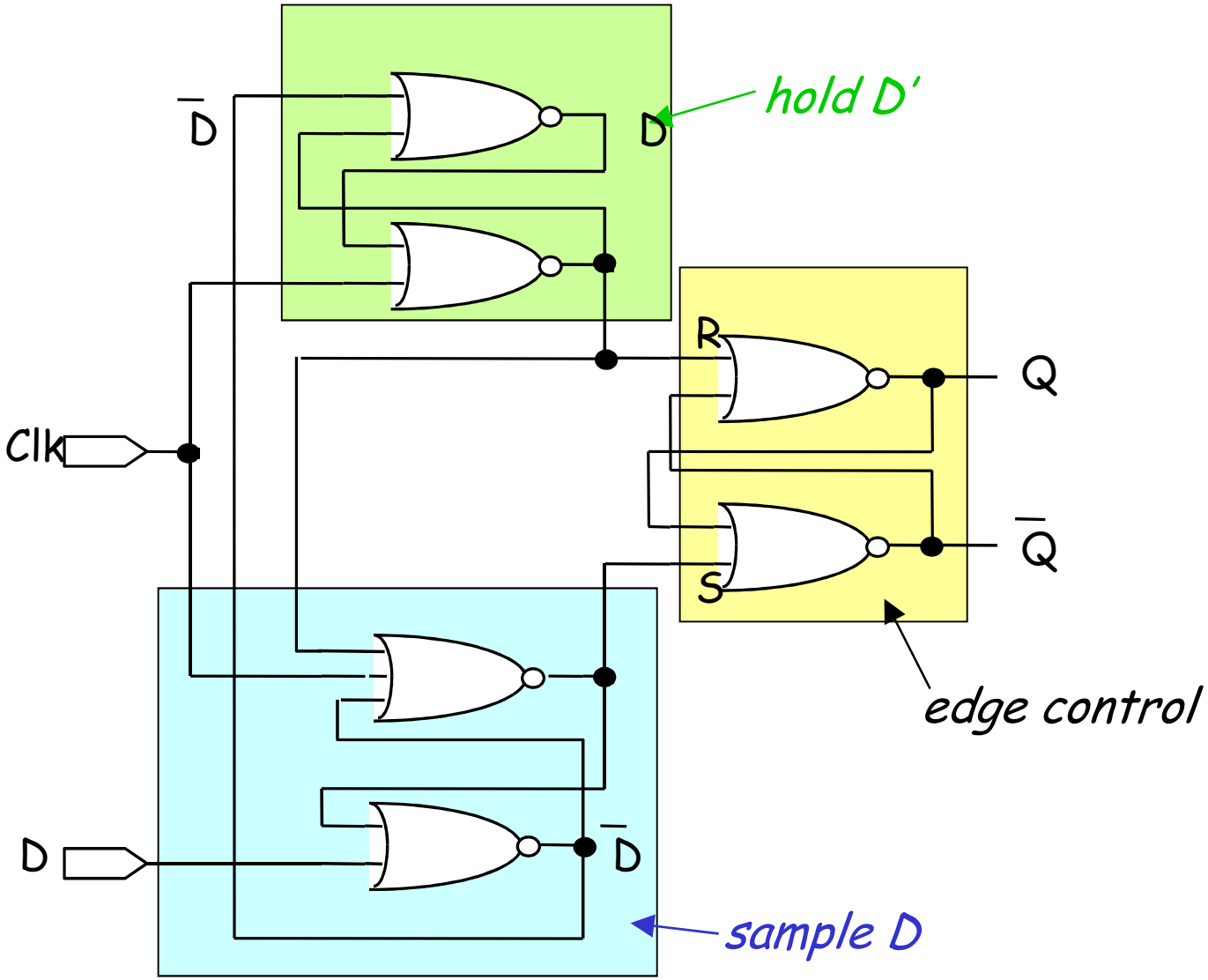


Formed by adding inverter to clock input

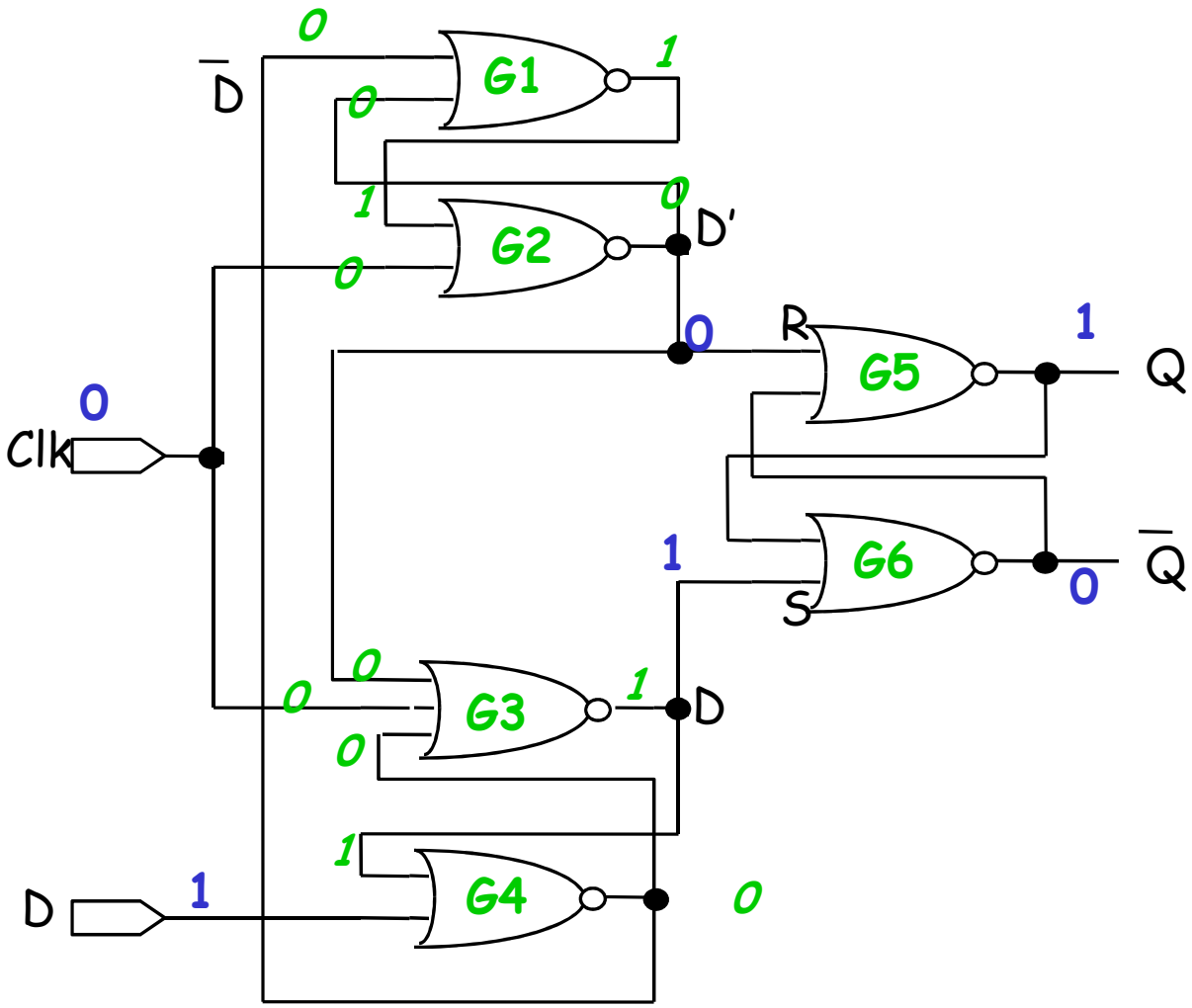
Negative Edge-Triggered D Flip Flop



Negative Edge-Triggered D Flip Flop

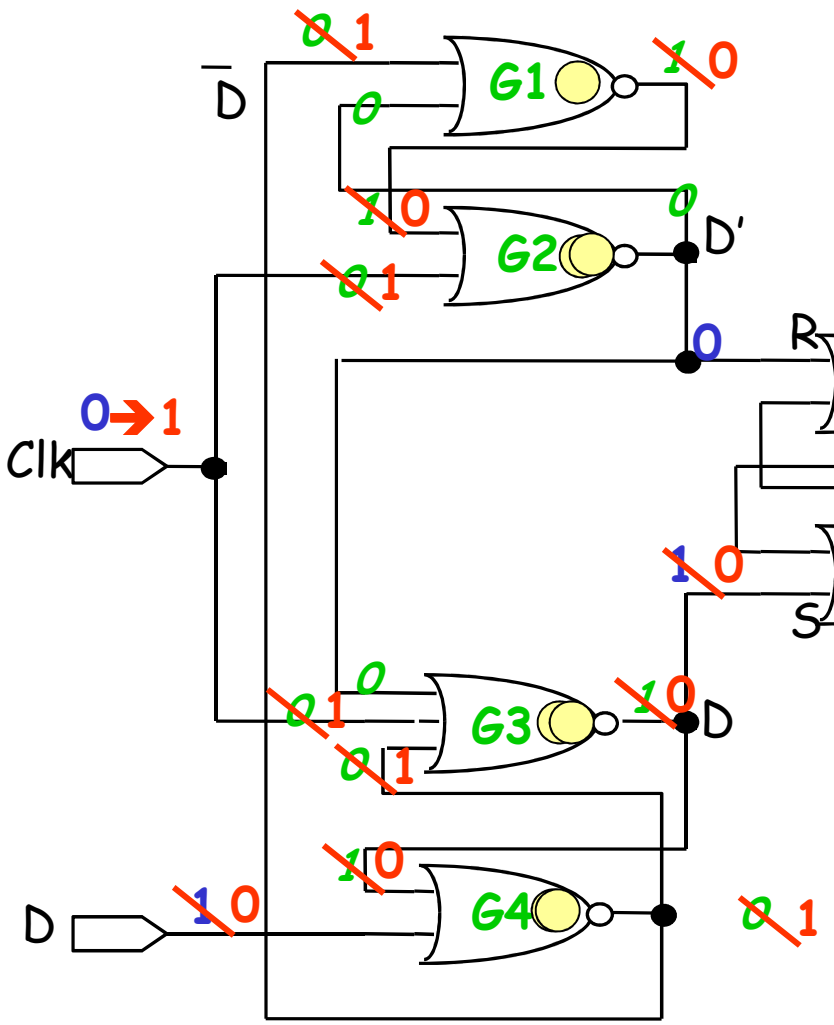


Negative Edge-Triggered D Flip Flop



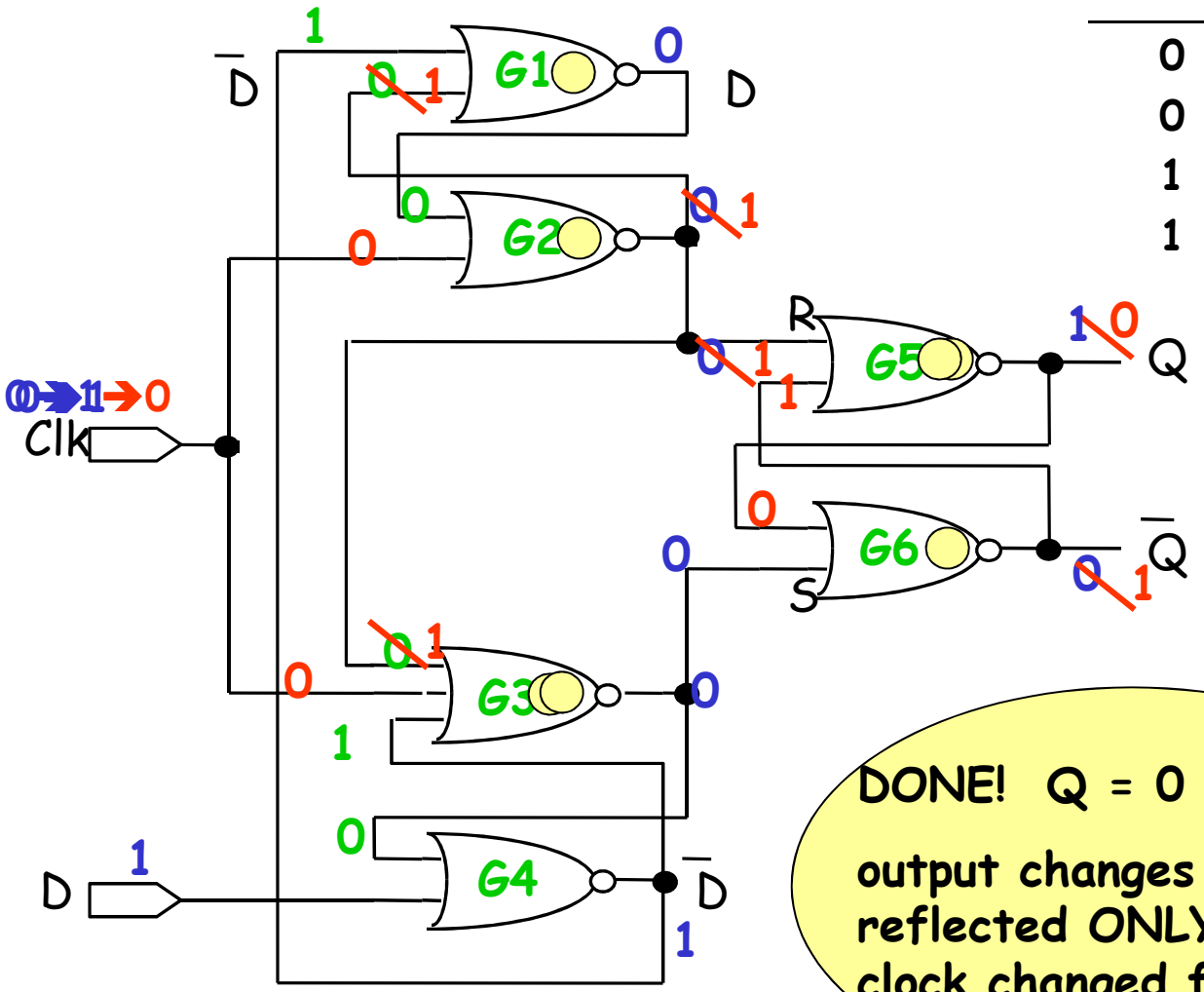
Negative Edge-Triggered D Flip Flop

		nor
0	0	1
0	1	0
1	0	0
1	1	0



DONE!
D has been sampled, but output has not changed yet

Negative Edge-Triggered D Flip Flop



Latches versus Flip-Flops

Type	Inputs Sampled	Outputs Valid
Unclocked latch	always	prop. delay from input change
Level sensitive	clock high	prop. delay from input change
Positive edge flip-flop	Clock Lo-to-Hi	prop. delay from rising edge
Negative edge flip-flop	Clock Hi-to-Lo	prop. delay from trailing edge
Master/slave flip-flop	Clock Hi-to-Lo	prop. delay from trailing edge

Standard Symbols for Storage Elements

- Master-Slave: Postponed output indicators
- Edge-Triggered: Dynamic indicator