Lecture 36 : Sequential Circuits

(36-37)

D, JK, T Flipflops

Clock Triggering

Recap:

SR FlipFlops

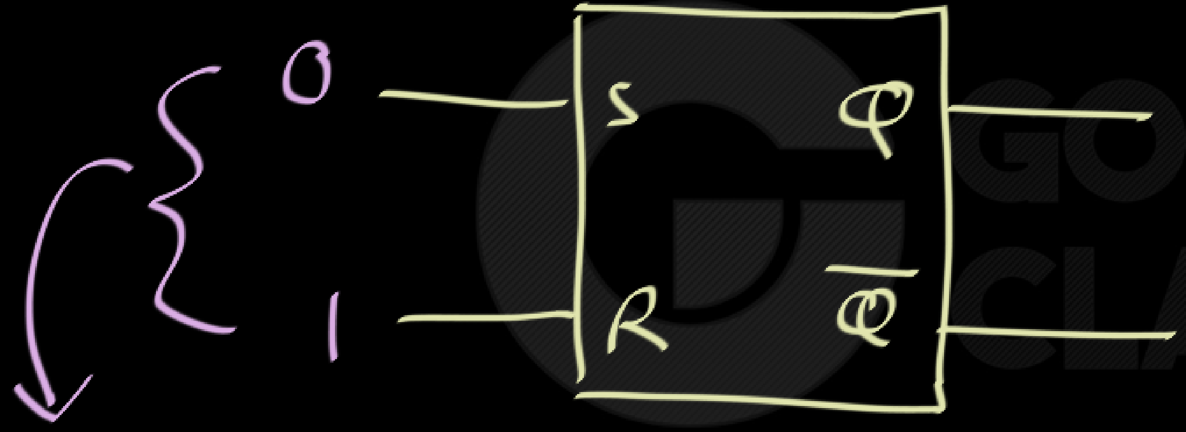
Store a bit

one bit Storage Device

Store "0" :

Store 0 ≡ Reset ≡ o/p 0

≡ State 0

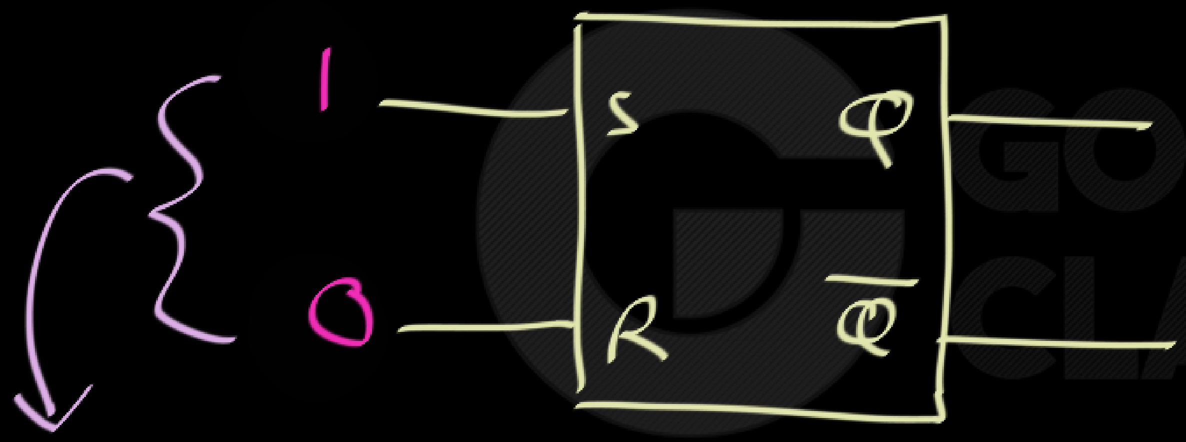


two signals changed to store a bit.

Store "1" :

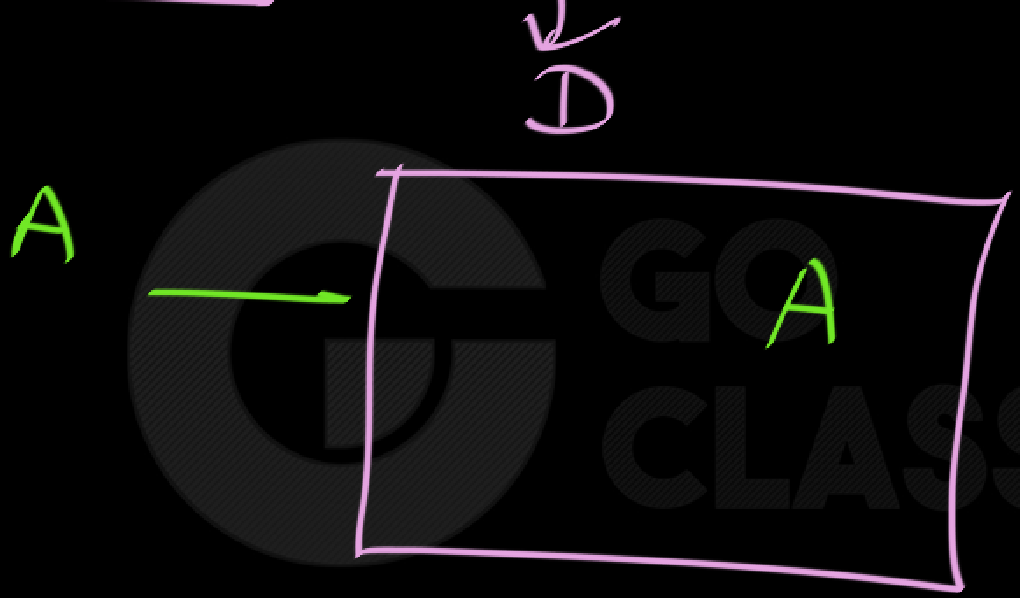
Store 1 $\equiv$ Set $\equiv$ o/p 1

$\equiv$ State 1



two signals changed to store a bit.

Google Drive:



To Store Photo A,
Just input Photo A.

||

Same idea for D flip flop.

Next Topic:

2. D Flipflop

D Flipflop = Data(storage) Flipflop

Idea of D flipflop :

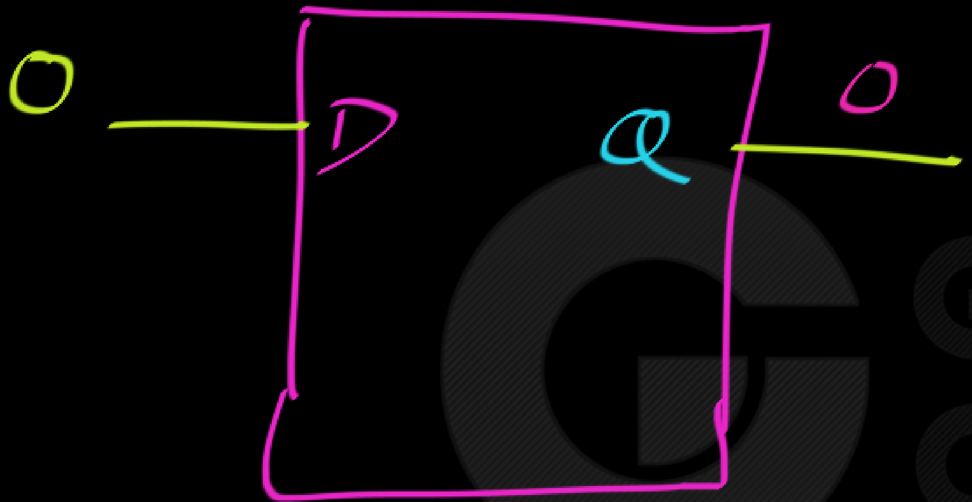
To store a bit, Just provide that bit on input, we will store it.

No need of changing two signals like SR.

Store 0 :

Store 0 $\equiv$ Reset $\equiv$ o/p 0

$\equiv$ State 0

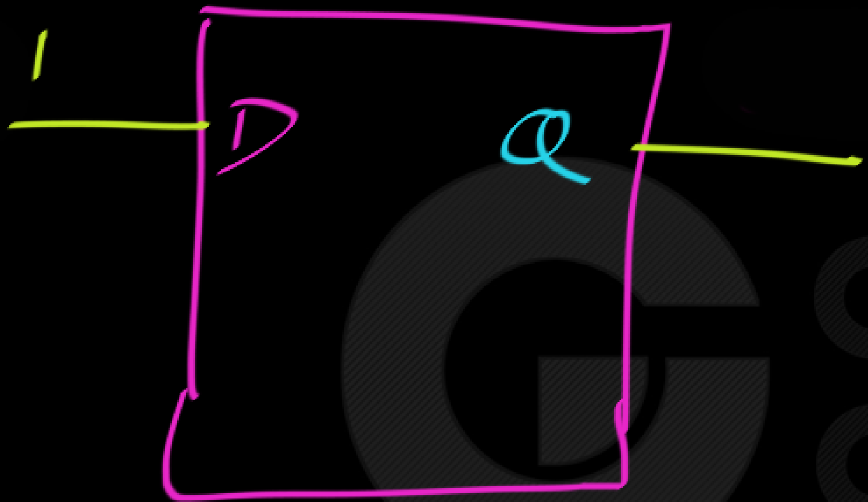


To make $Q_{next} = 0$; Present input $D = 0$

Store 1 :

Store 1 ≡ Set ≡ O/P 1

≡ State 1



To make $Q_{next} = 1$; Present input $D = 1$

Data flipflop / D ff

Data storage.

Slogan of D ff :

Nothing fancy, Just Store Your Data.

One bit Data Storage

SR

S	R	Q_{t+1}	
0	1	0	Reset
1	0	1	Set
0	0	Q_t	Retain
1	1	x	forbidden

D - ff → Data

D	Q_{t+1}
0	0
1	1

Characteristic Table : / State Table :

Clock	D	Q_{t+1}
1	0	0
1	1	1
0	X	Q_t

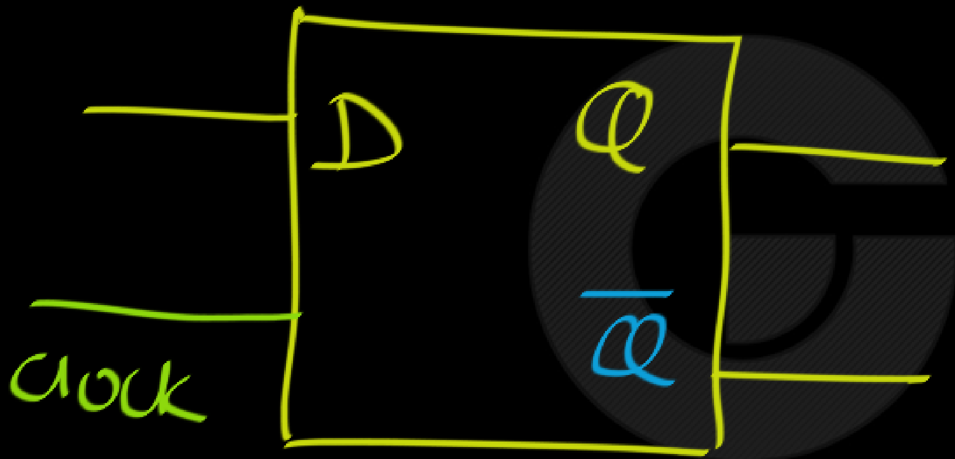
Q_{t+1} → next o/p

When clock = 0

||

ff Does not Respond to input change.

Block Diagram:



State equation / Characteristic equation:

$$Q_{t+1} = f(D, Q_t)$$

$$Q_n = f(D, Q)$$

Q_n: Next o/p

D: Present input

Q: Present o/p

Char. equation:

$$\boxed{Q_{t+1} = D}$$ ✓

$\Rightarrow$ whatever input you Apply now, will be the next o/p (Present state Does not matter)

Truth Table: $Q_n = f(D, Q)$

D	Q	Q_n
0	0	0
0	1	0
1	0	1
1	1	1

Dummy Variable in D-ff

$\boxed{Q_n = D}$

$\boxed{Q_n = D}$

Excitation Table : (from o/p point of view)

State / o/p → State / o/p

which input
Combination needed

Excitation Table: (from o/p point of view)

Q_t	Q_{t+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

Whatever next state you want, Just Apply that as input.

Analogy:

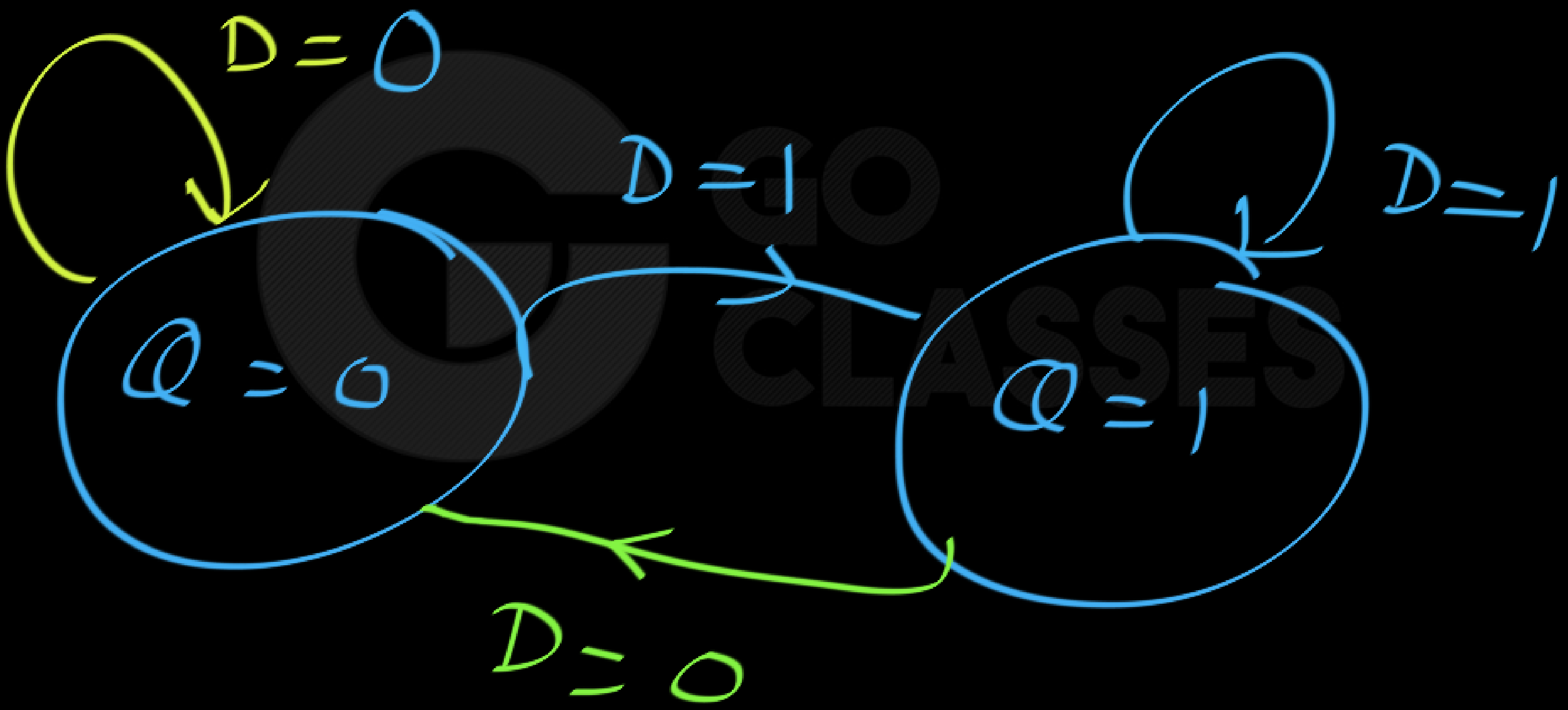
instagram:

username

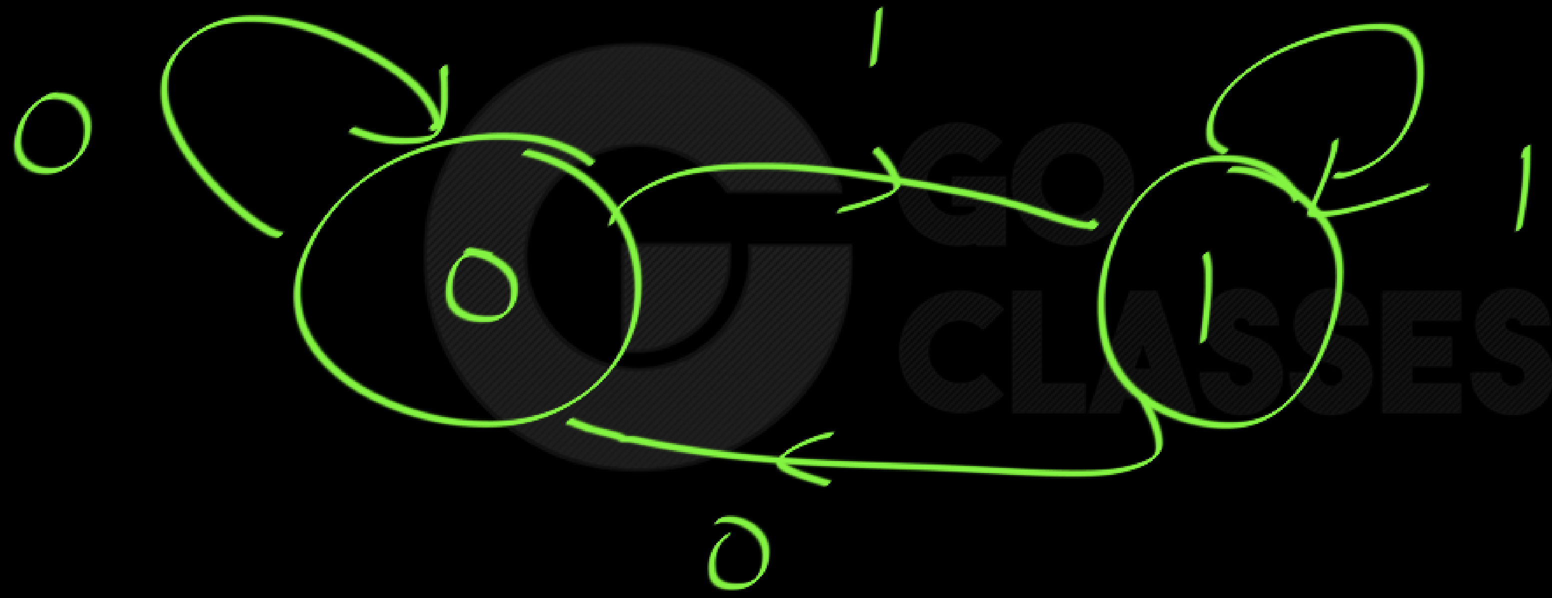
present — Q_t $\boxed{abc\_123}$ $D = xyz\_456$

next Q_{t+1} $\boxed{xyz\_456}$

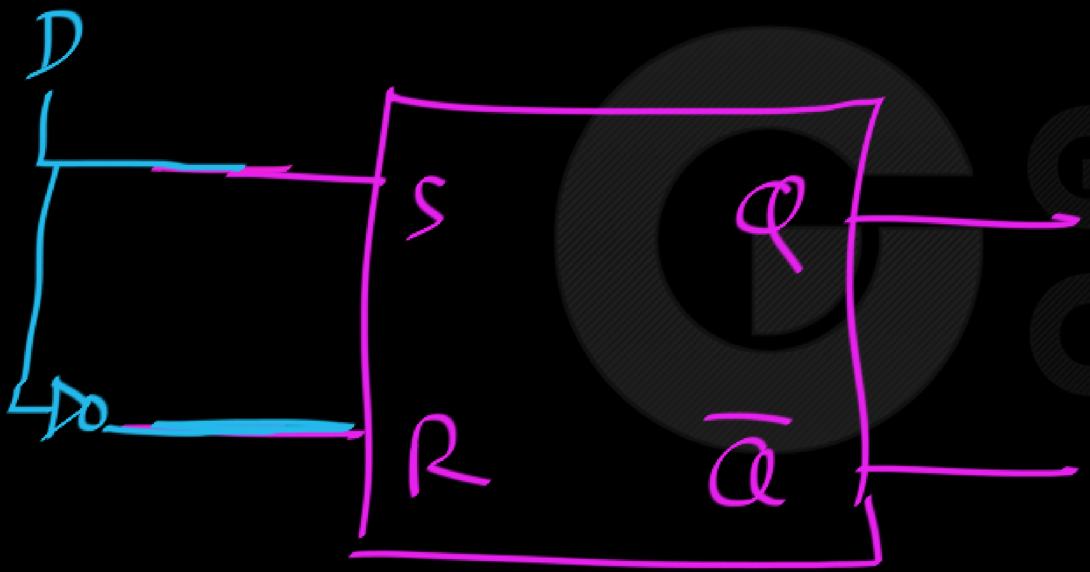
State Diagram: (State ≡ O/p ≡ Stored value)



State Diagram: (State ≡ O/p ≡ Stored value)

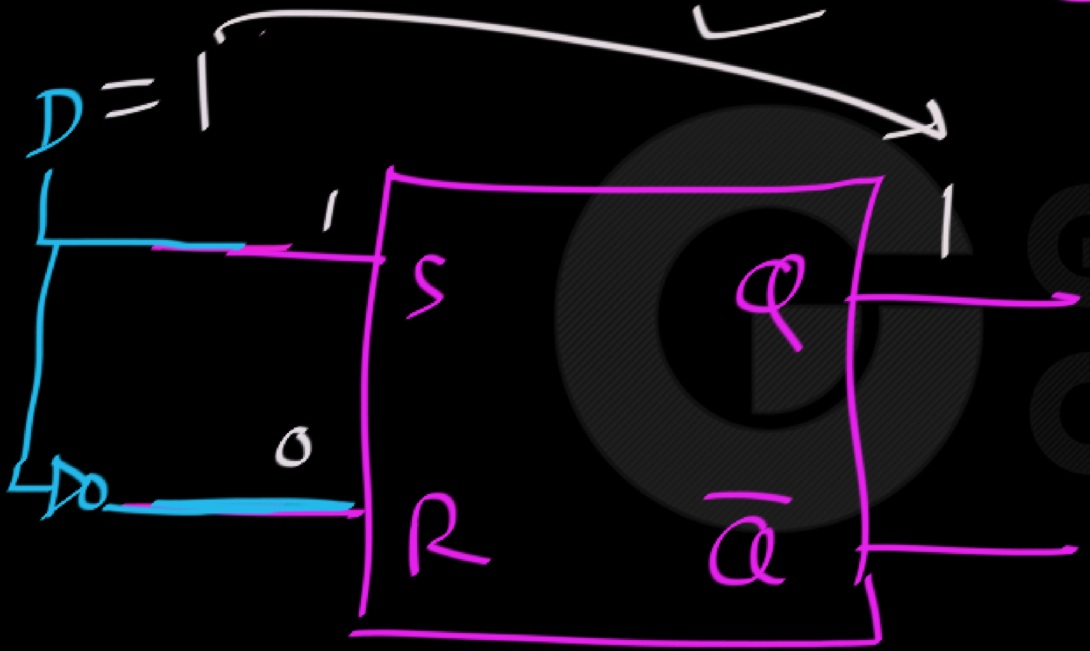


Implementation: D using SR ff:



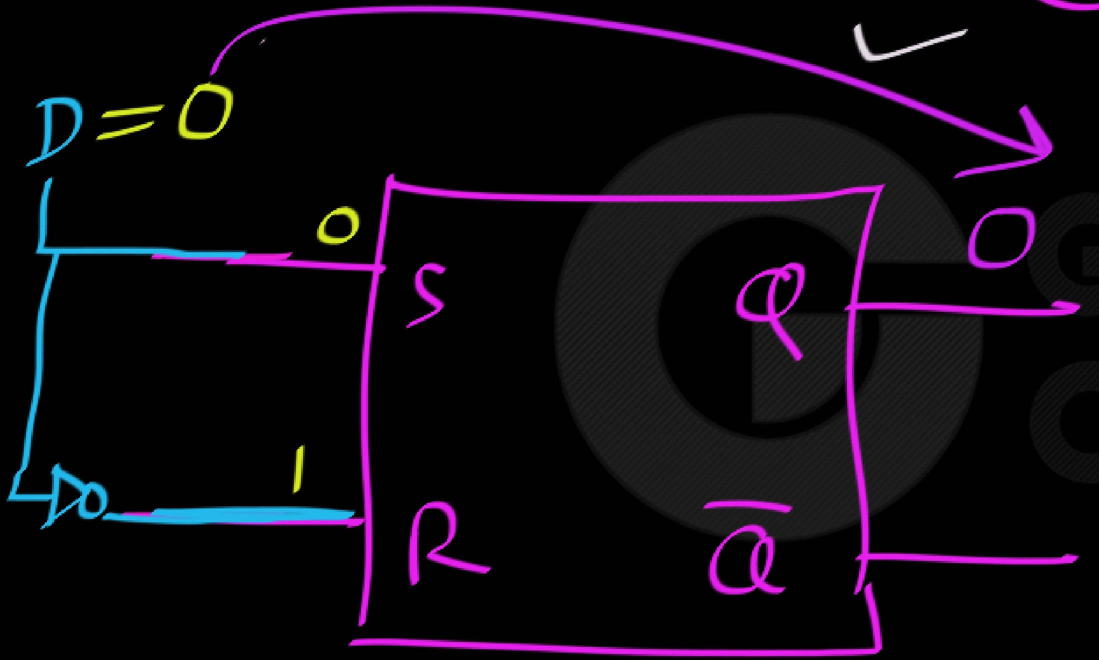
D	Q_{t+1}
0	0 ✓
1	1

Implementation : D using SR ff :



D	Q_{t+1}
0	0
1	1

Implementation : D using SR ff :



D	Q_{t+1}
0	0
1	1

Implementation :

S	R	Q_{t+1}		D
0	1	0	Store 0	0
1	0	1	Store 1	1

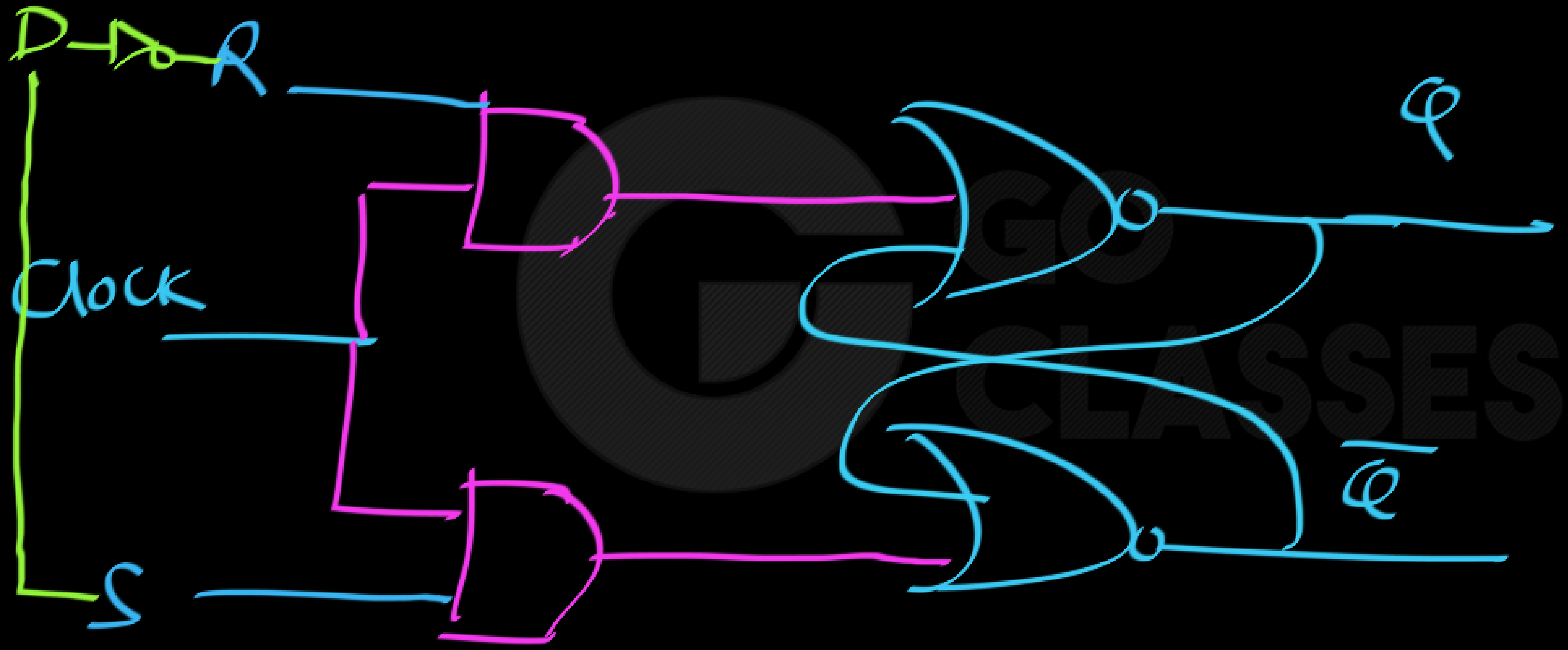
Implementation :

D using SR :

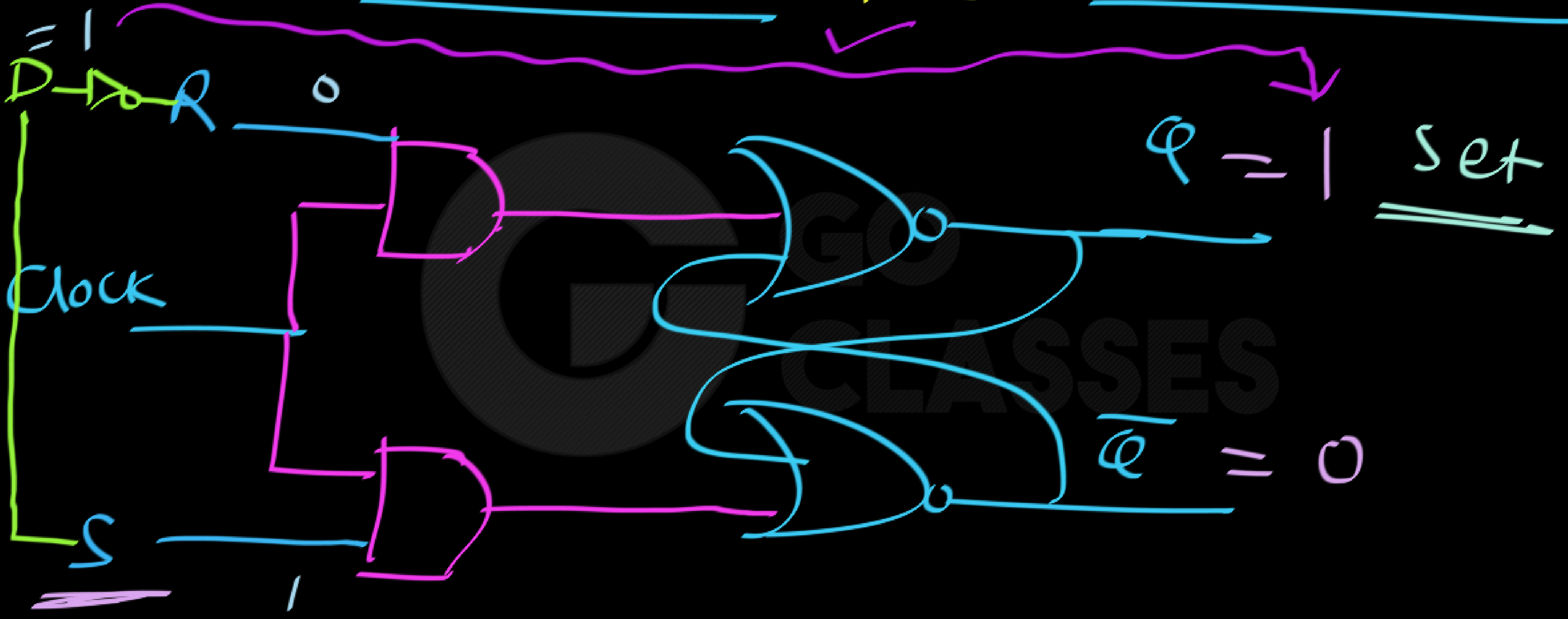
$D \longrightarrow S$

$D' \longrightarrow R$

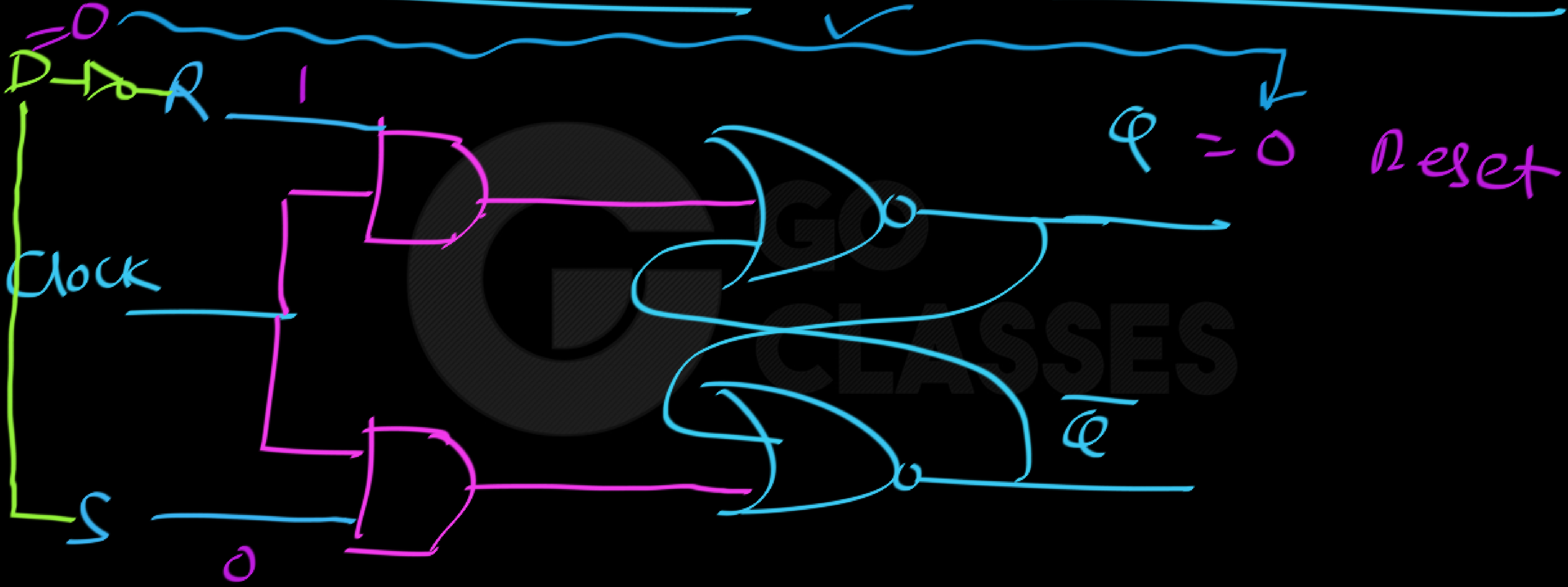
D using SR flip flop (of NOR implementation)



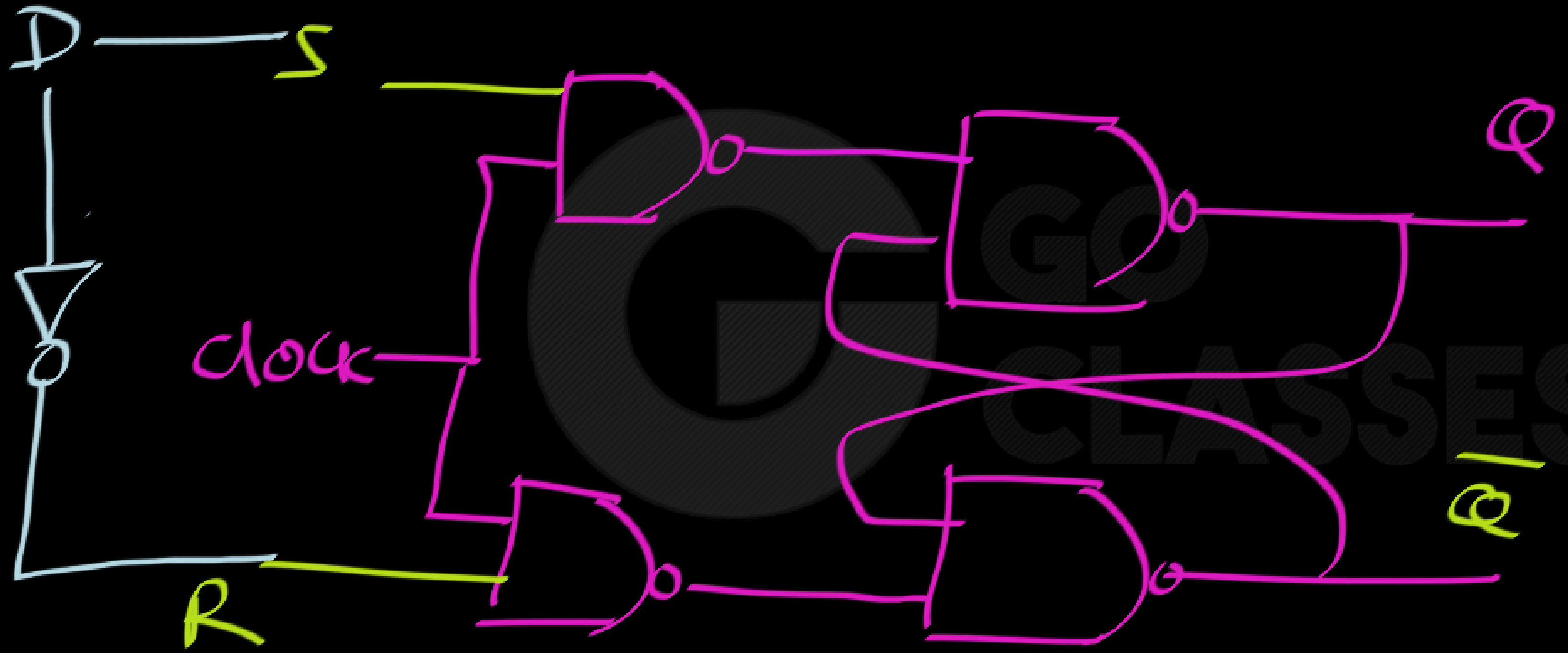
D using SR flip flop (of NOR implementation)



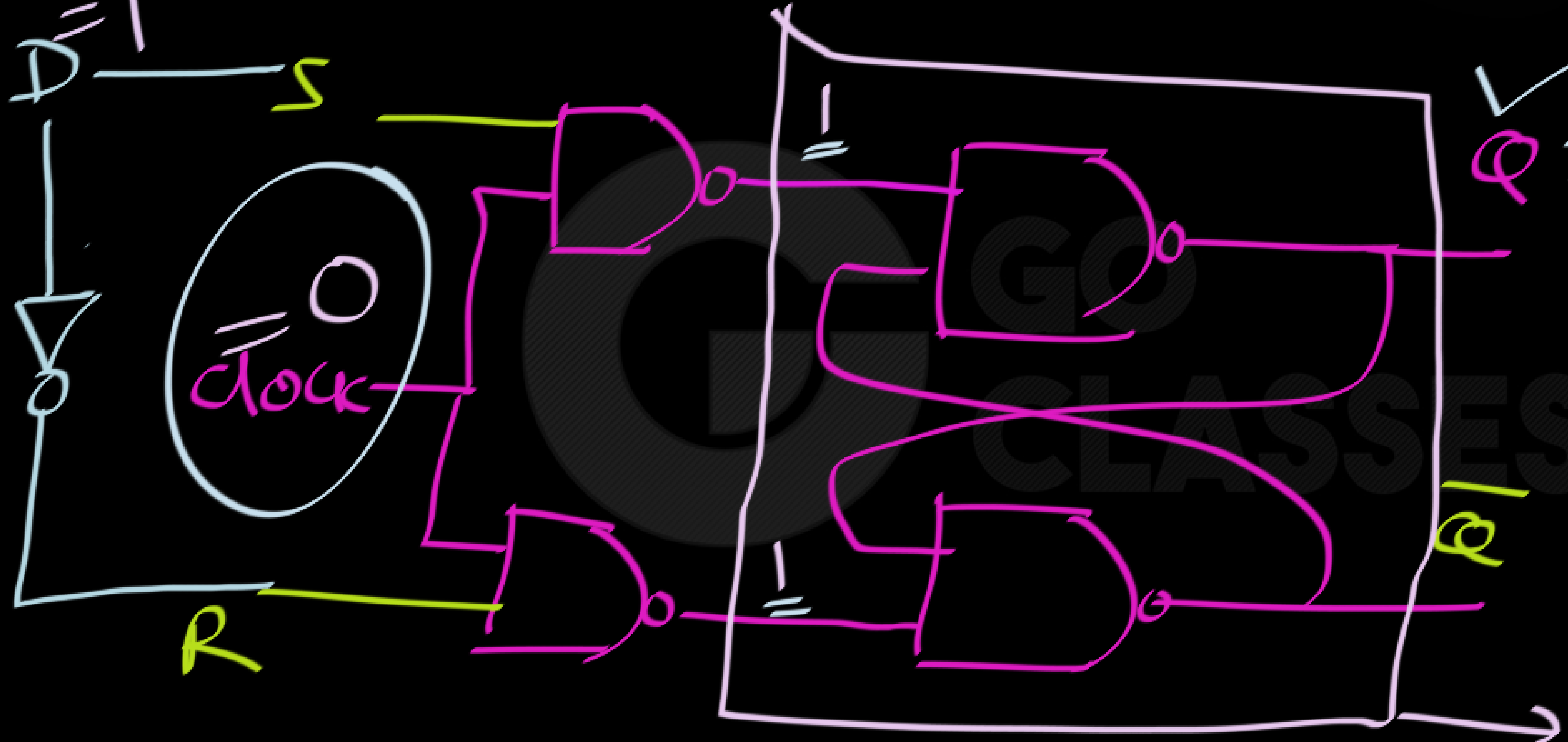
D using SR flip flop (of NOR implementation)



D using SR flip flop (of NAND implementation)



D using SR flip flop (of NAND implementation)

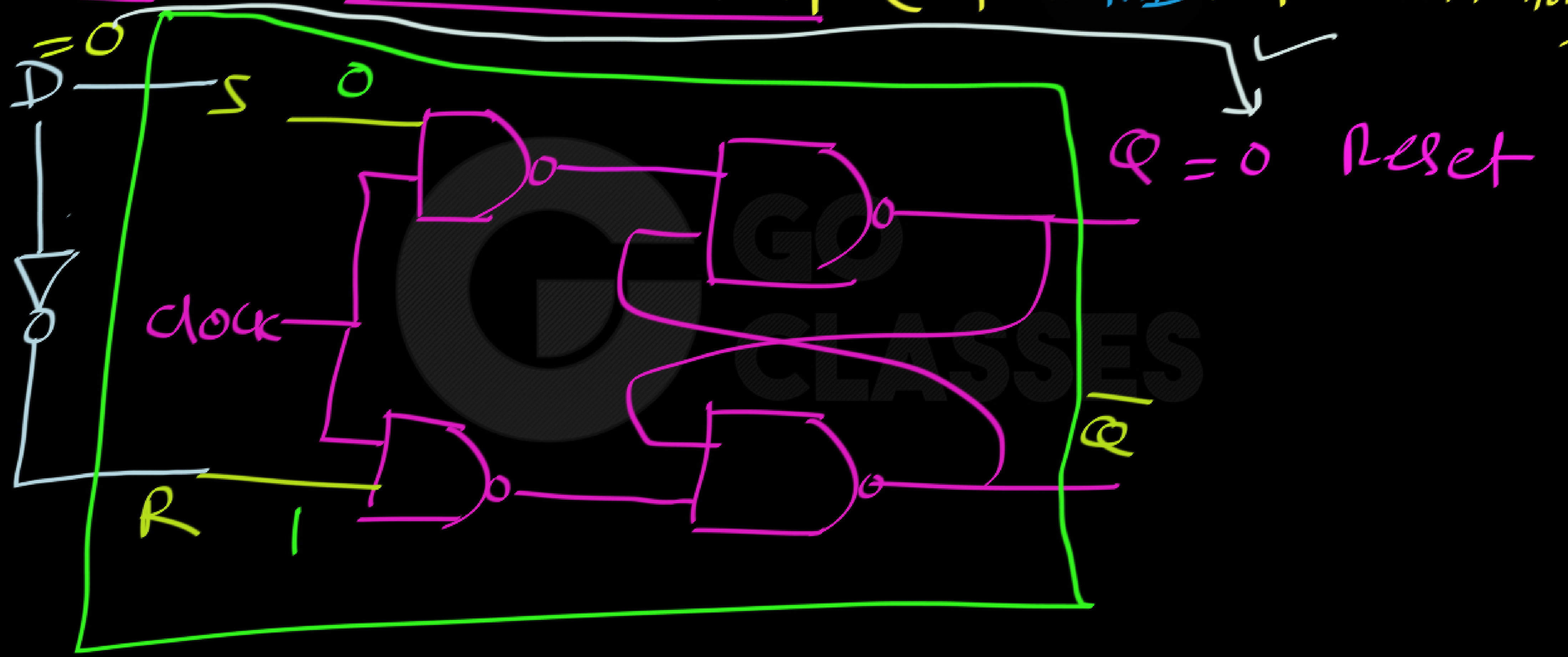


Q = Unchanged

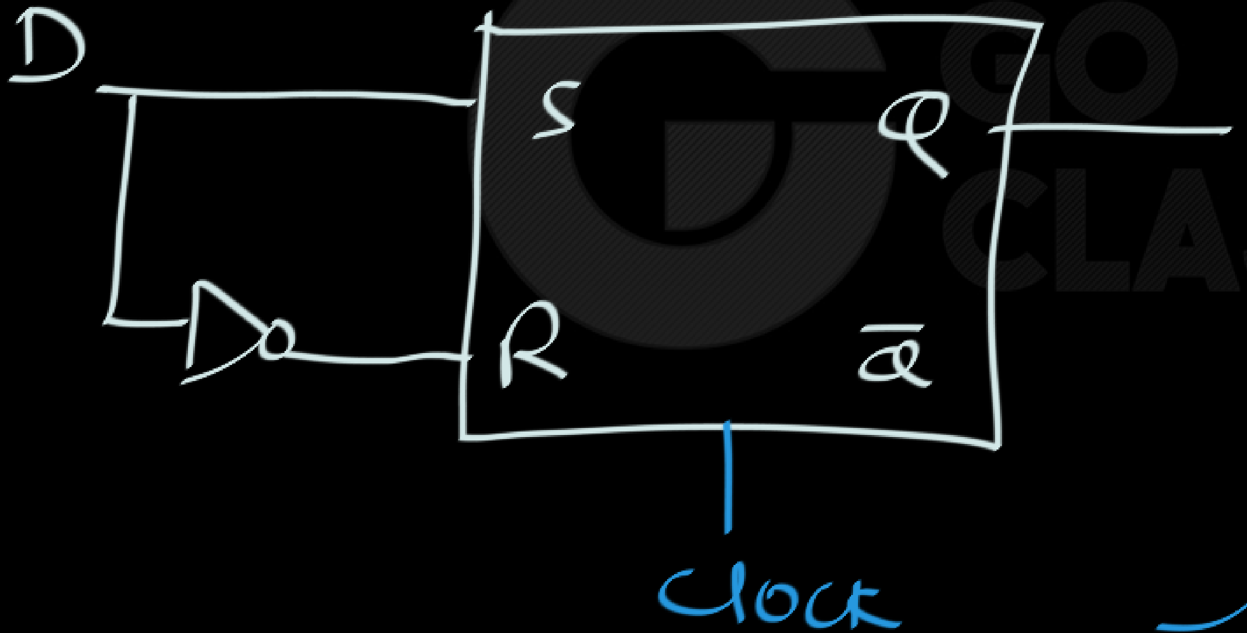
whatever D applied

$\bar{S}\bar{R}$ latch

D using SR flip flop (of NAND implementation)



D using SR :



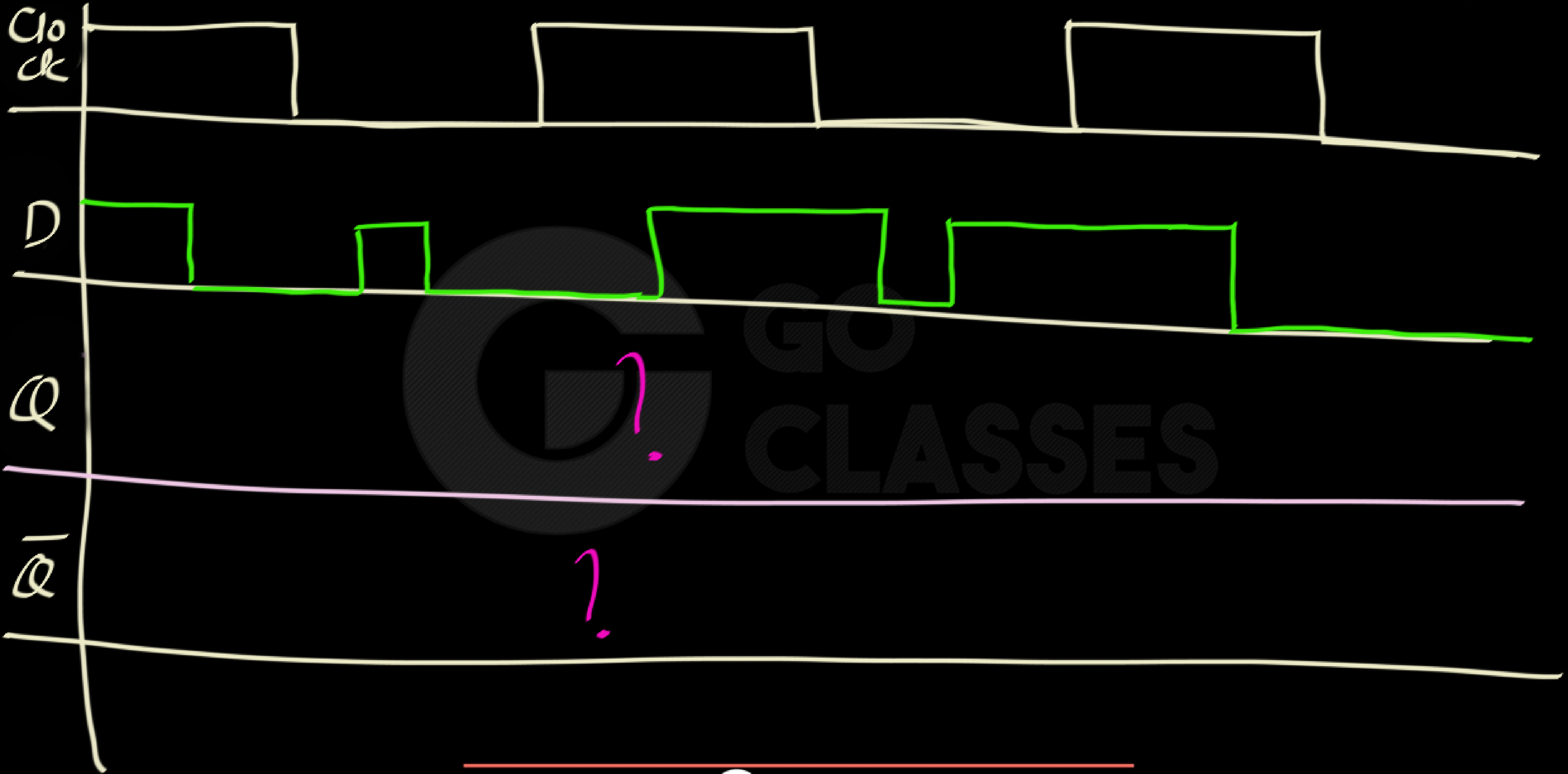
S = R Never Possible Now.

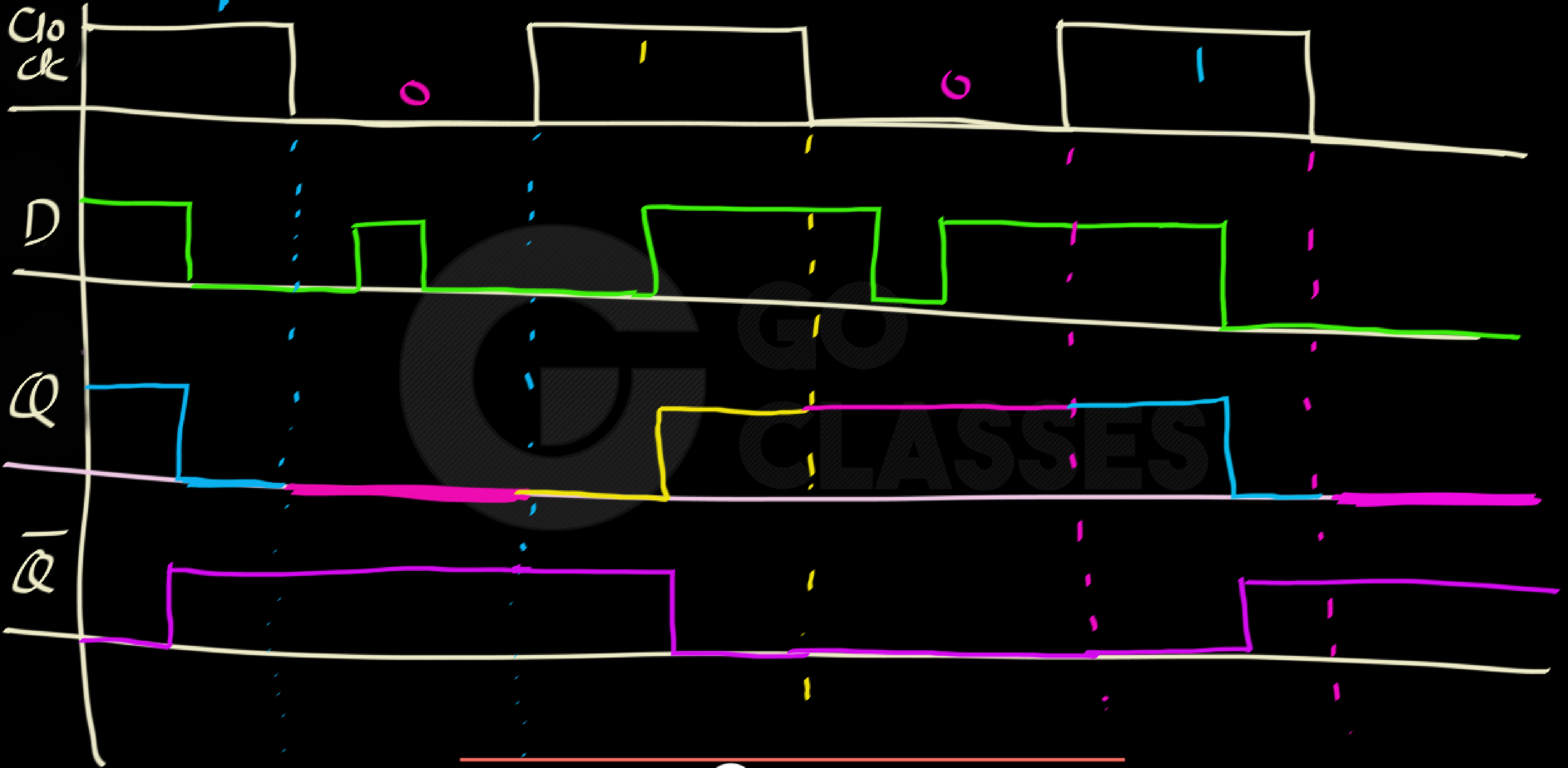
When we implement D using SR.

Next Topic:

2. D Flipflop

Timing Diagram


Clock
D
Q
?
$\overline{Q}$
?


Clock
D
Q
$\overline{Q}$
0
1
0
1

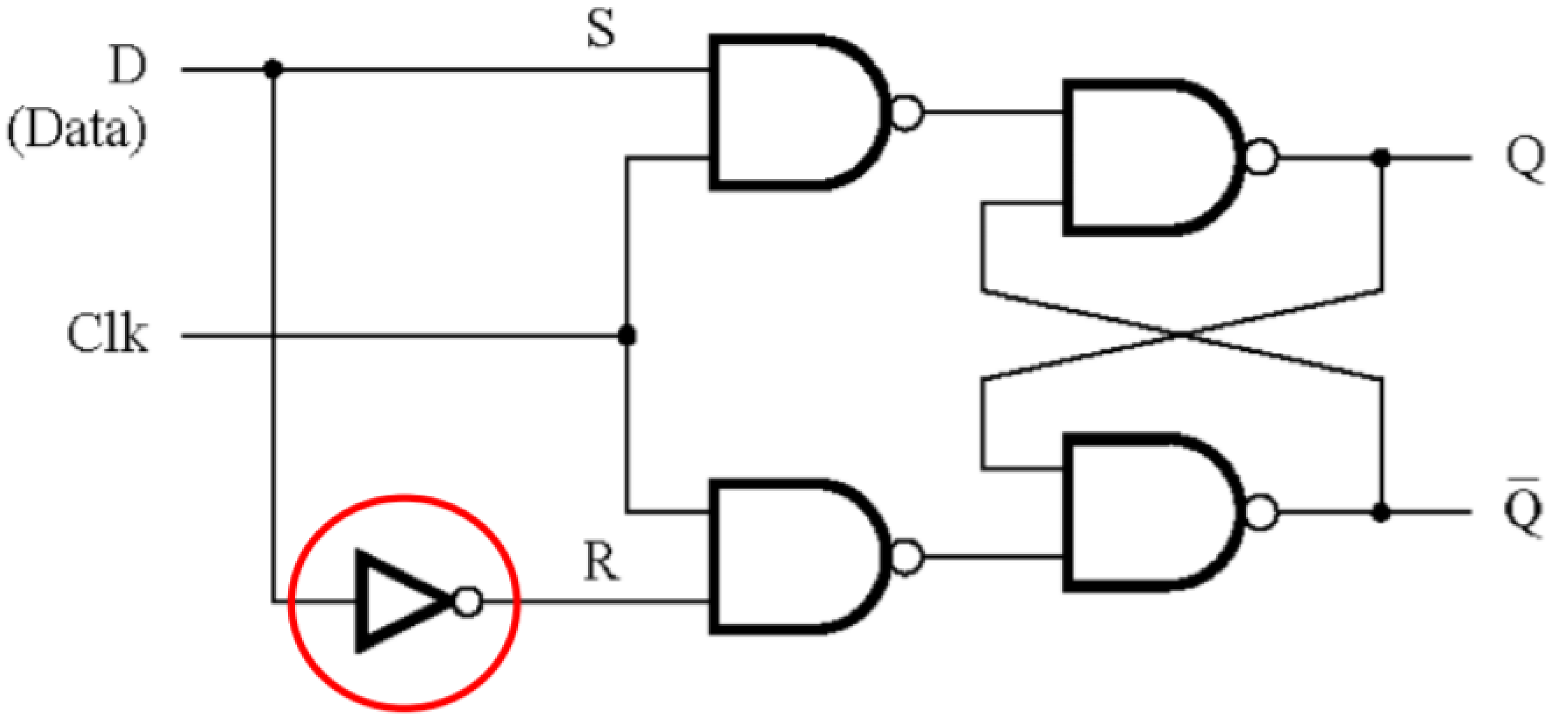
Motivation

- Dealing with two inputs (S and R) could be messy. For example, we may have to reset the latch before some operations in order to store a specific value but the reset may not be necessary depending on the current state of the latch.

- Why not have just one input and call it D.

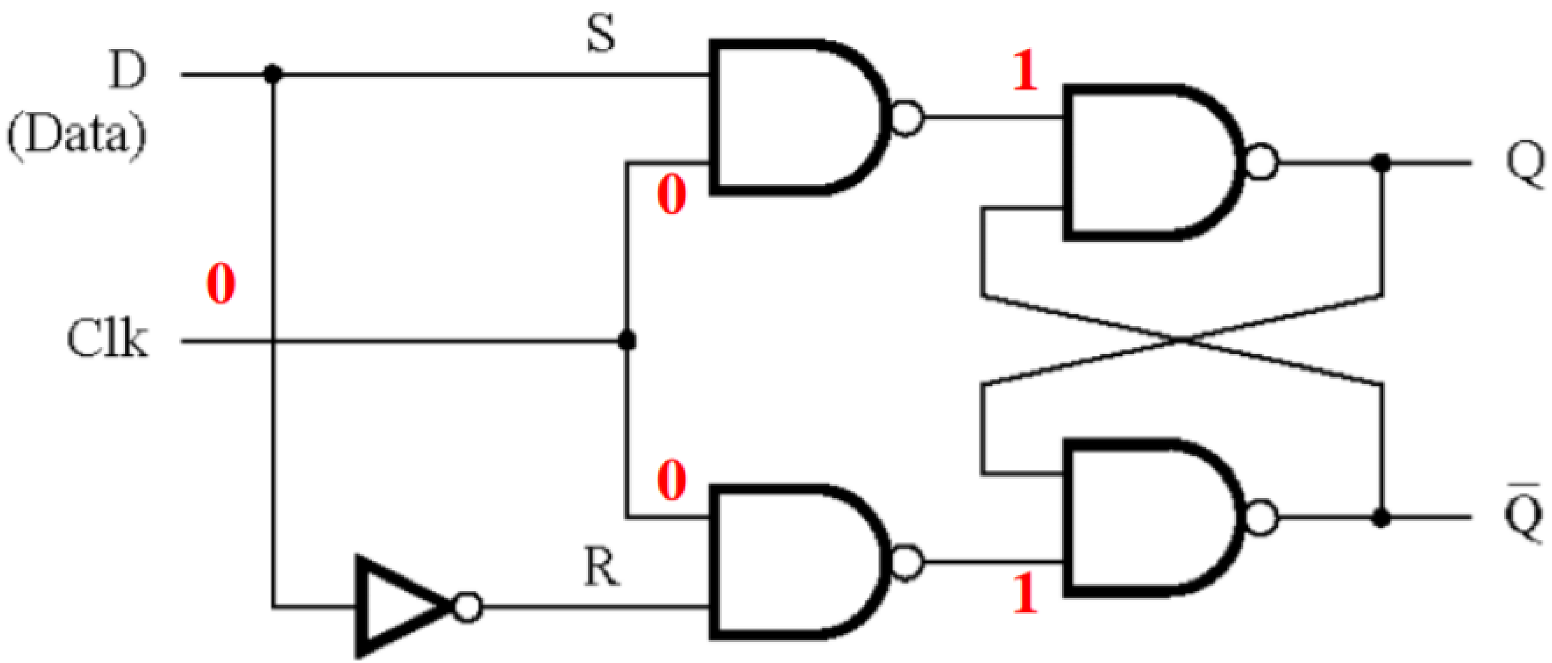
- The D latch can be constructed using a simple modification of the SR latch.

Circuit Diagram for the Gated D Latch



This is the only
new thing here.

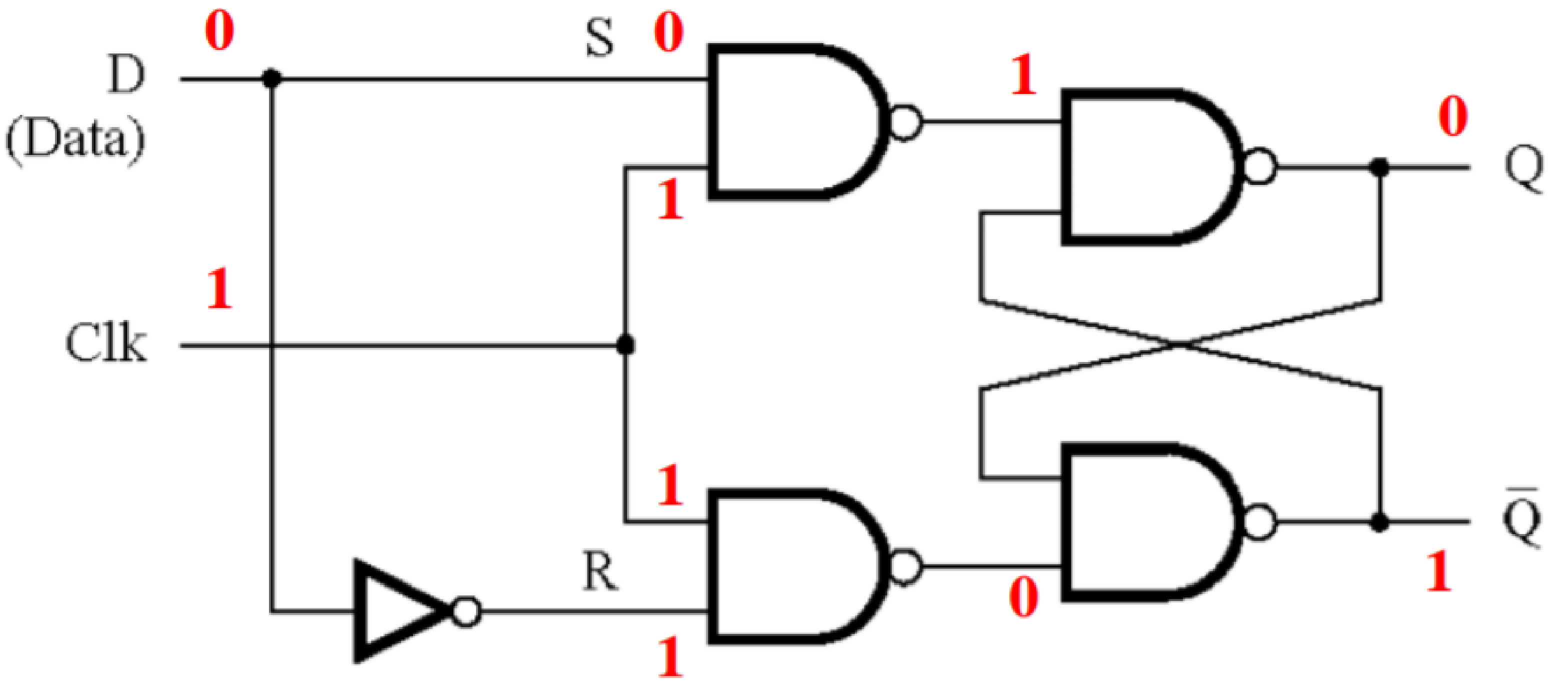
Circuit Diagram for the Gated D Latch



$\overline{S}$	$\overline{R}$	Q_a	Q_b	
0	0	1	1	
0	1	1	0	
1	0	0	1	
1	1	$\boxed{0/1}$	$\boxed{1/0}$	(no change)

[Figure 5.7a from the textbook

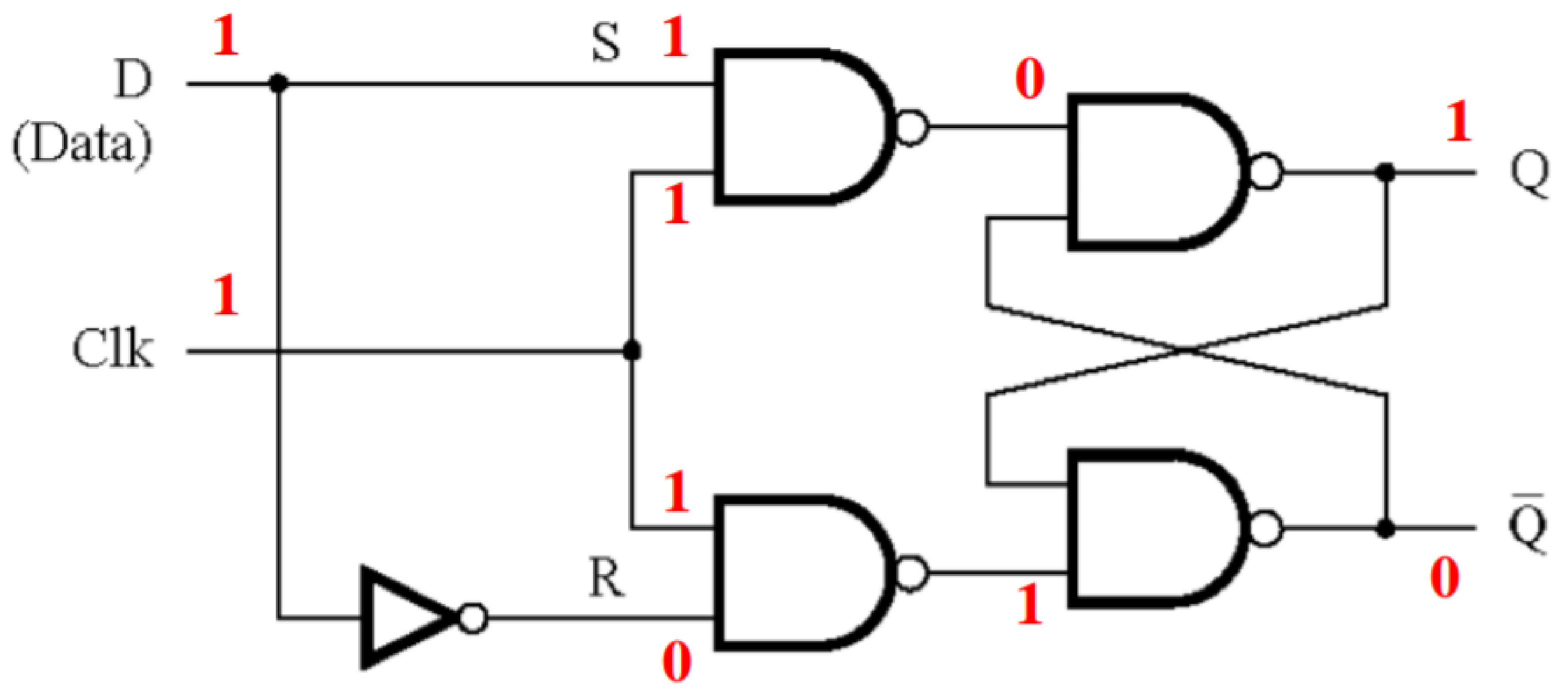
Circuit Diagram for the Gated D Latch



$\overline{S}$	$\overline{R}$	Q_a	Q_b	
0	0	1	1	
0	1	1	0	
1	0	0	1	
1	1	0/1	1/0	(no change)

[Figure 5.7a from the textbook]

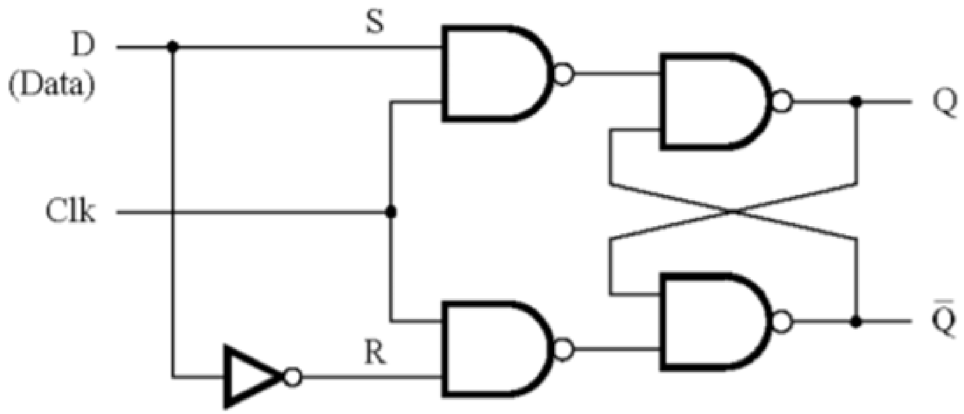
Circuit Diagram for the Gated D Latch



$\overline{S}$	$\overline{R}$	Q_a	Q_b	
0	0	1	1	
0	1	1	0	
1	0	0	1	
1	1	0/1	1/0	(no change)

[Figure 5.7a from the textbook

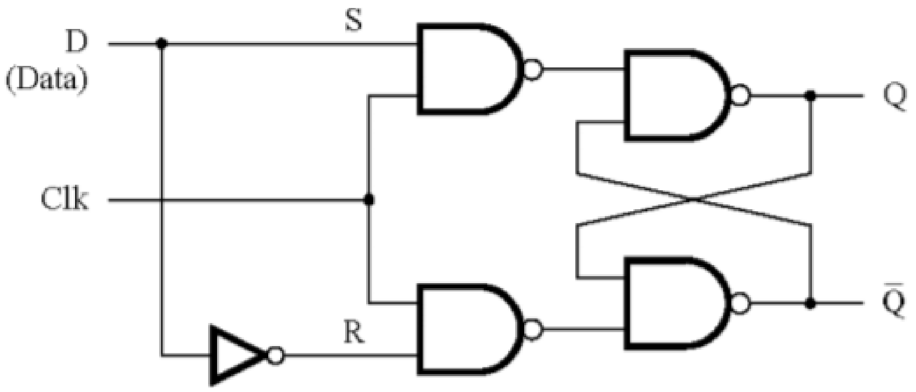
Circuit Diagram and Characteristic Table for the Gated D Latch



Clk	D	$Q(t+1)$
0	x	$Q(t)$
1	0	0
1	1	1

Note that it is now impossible to have S=R=1.

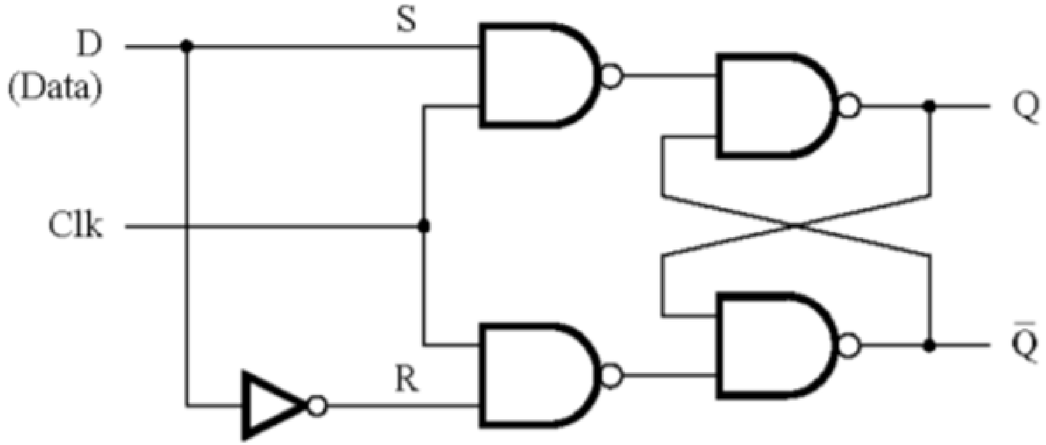
Circuit Diagram and Characteristic Table for the Gated D Latch

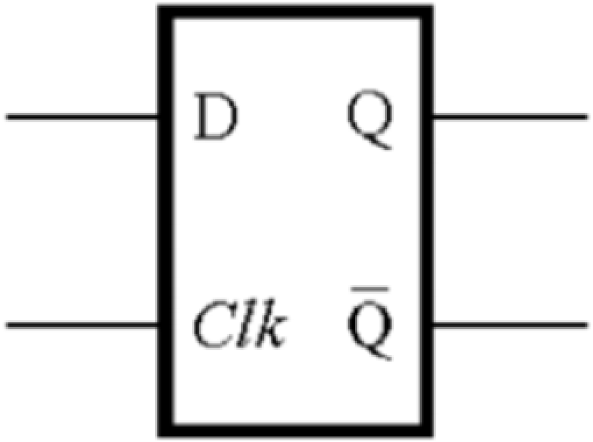


Clk	D	$Q(t+1)$
0	x	$Q(t)$
1	0	0
1	1	1

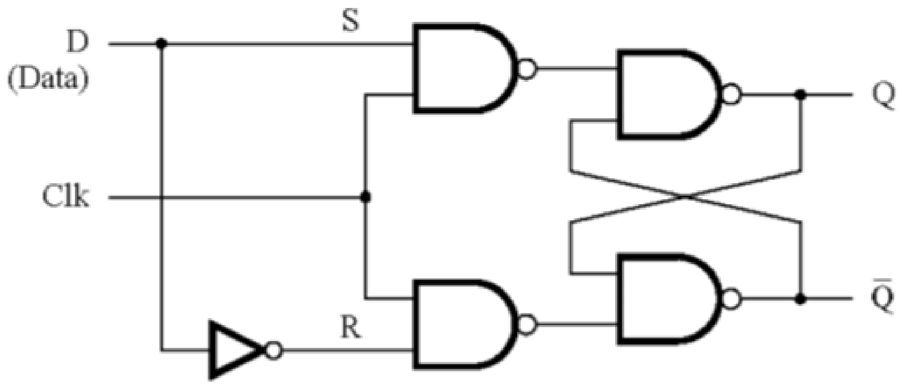
When Clk=1 the output follows the D input.
When Clk=0 the output cannot be changed.

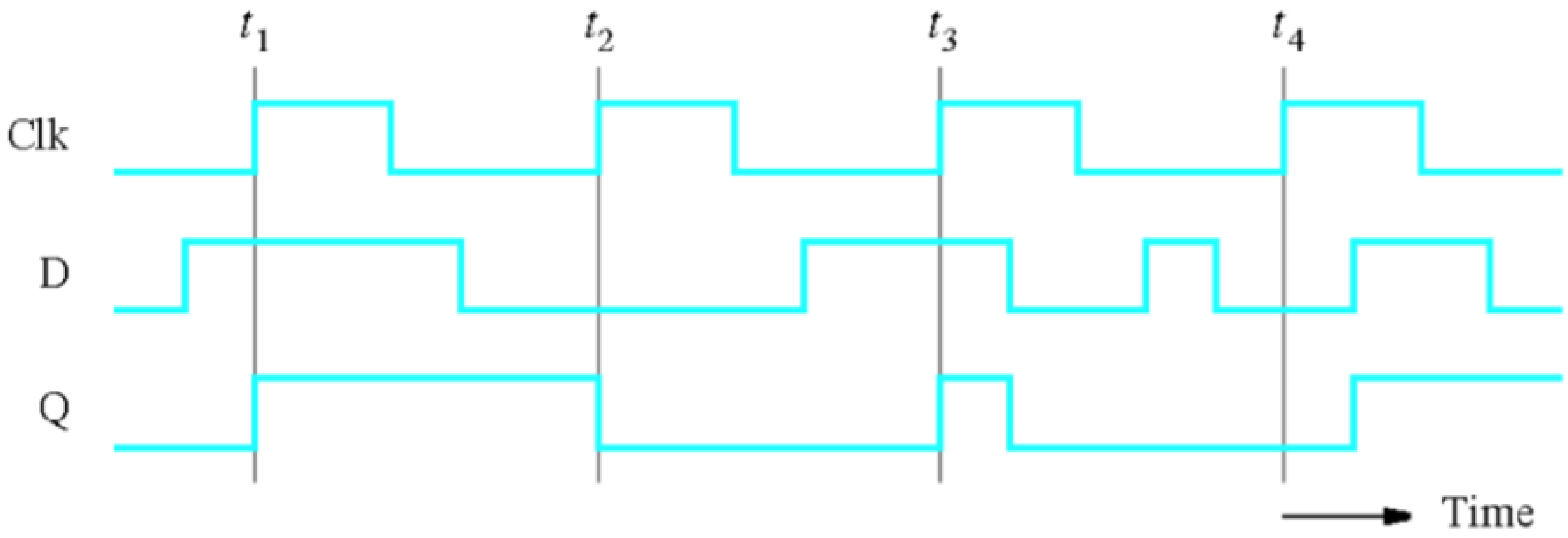
Circuit Diagram and Graphical Symbol for the Gated D Latch



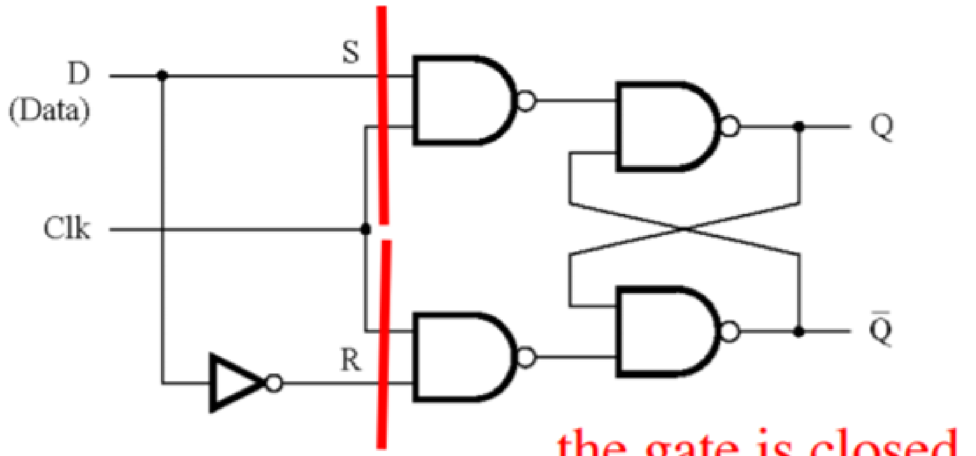


Timing Diagram for the Gated D Latch

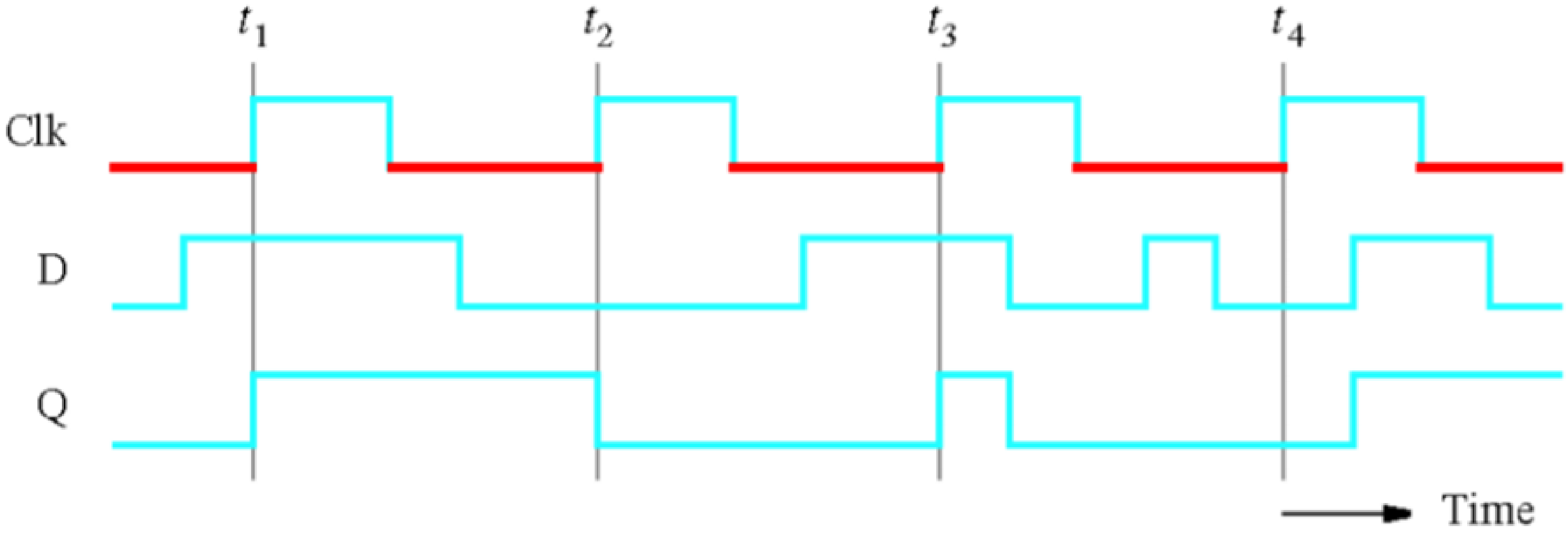




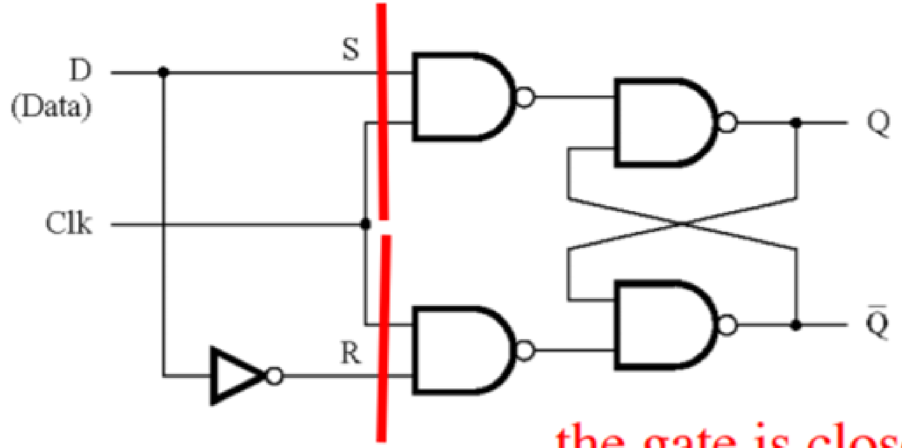
Timing Diagram for the Gated D Latch



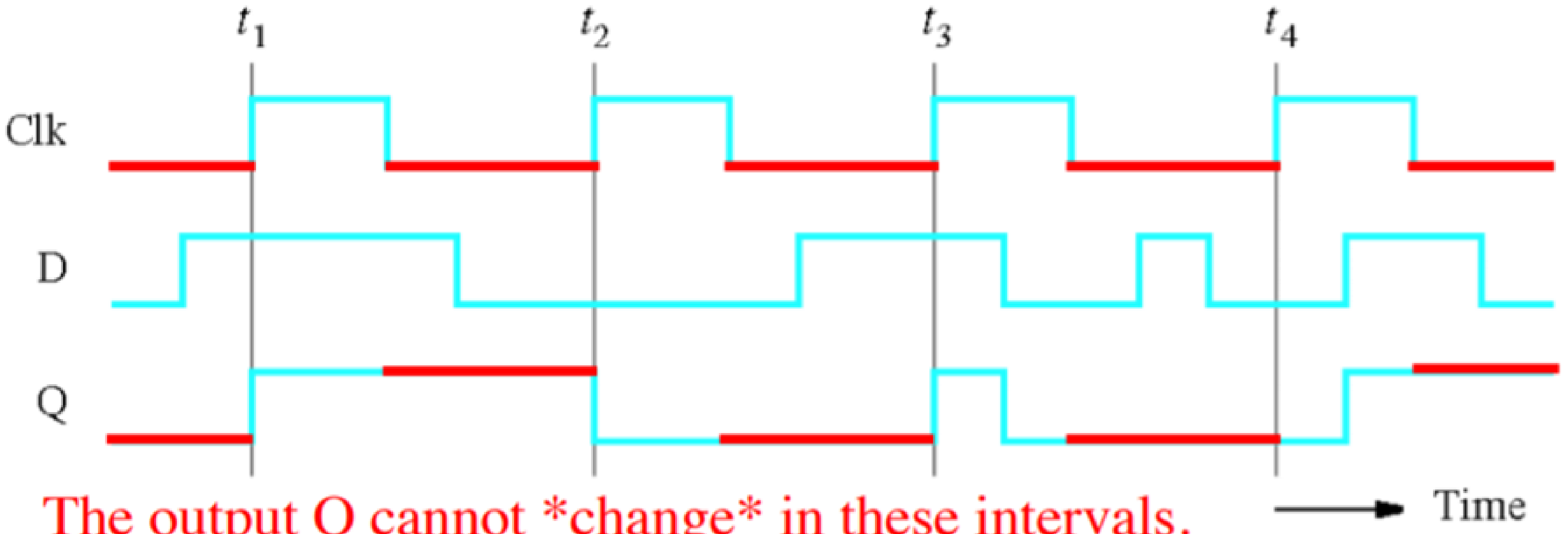
Clk=0



Timing Diagram for the Gated D Latch

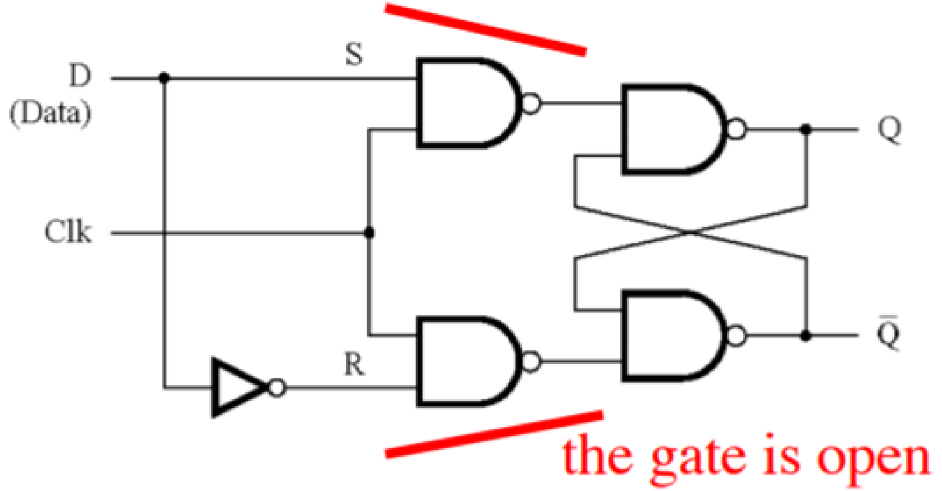


Clk=0

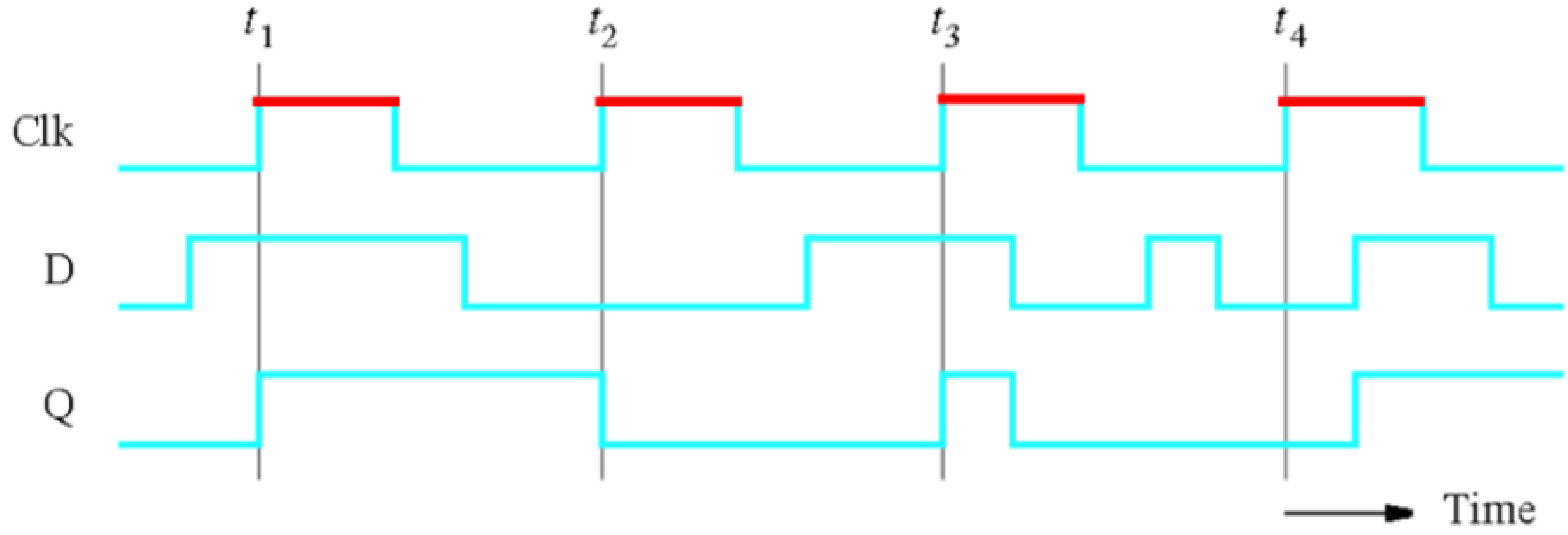


The output Q cannot *change* in these intervals.

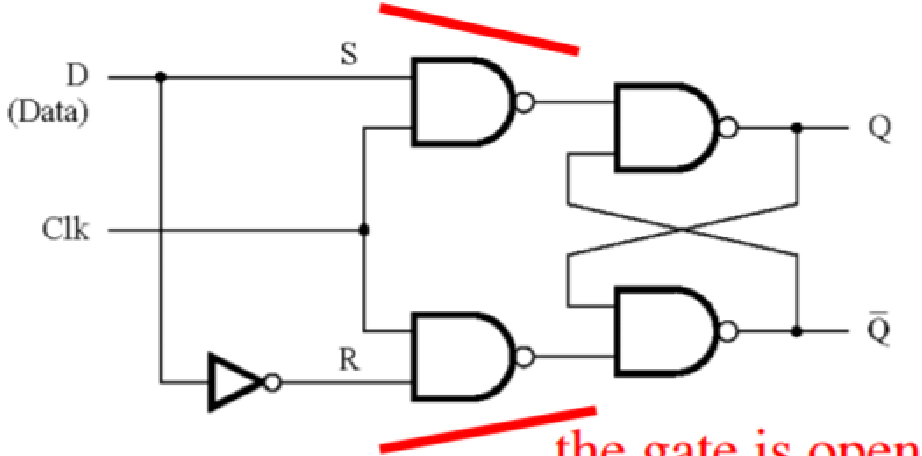
Timing Diagram for the Gated D Latch



Clk=1

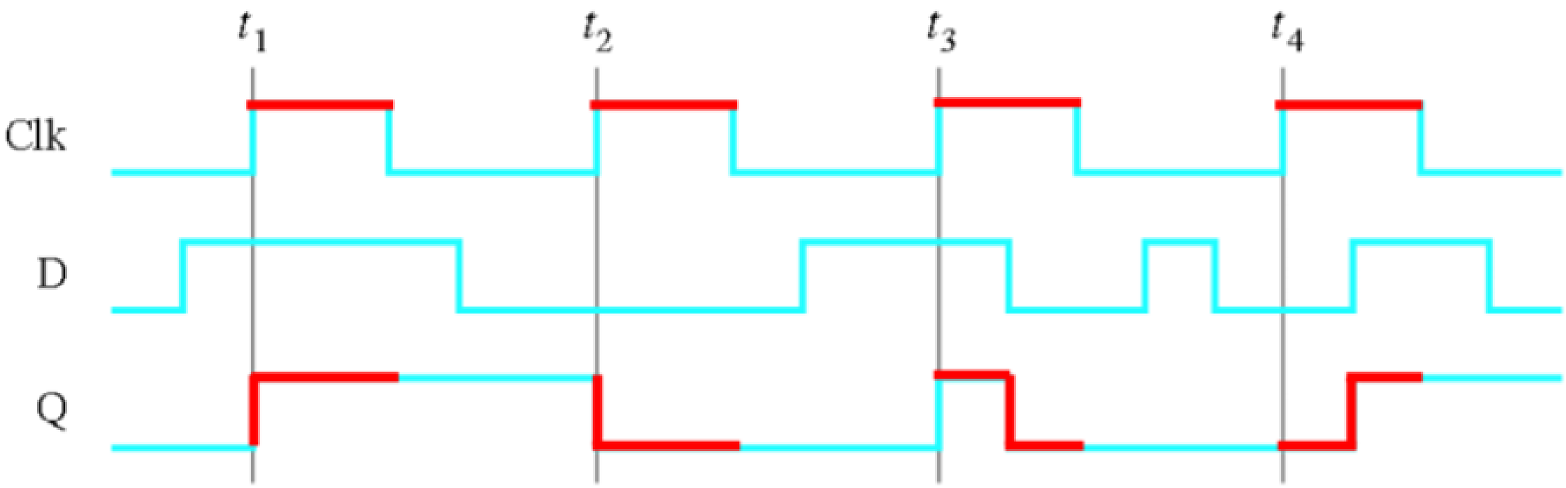


Timing Diagram for the Gated D Latch



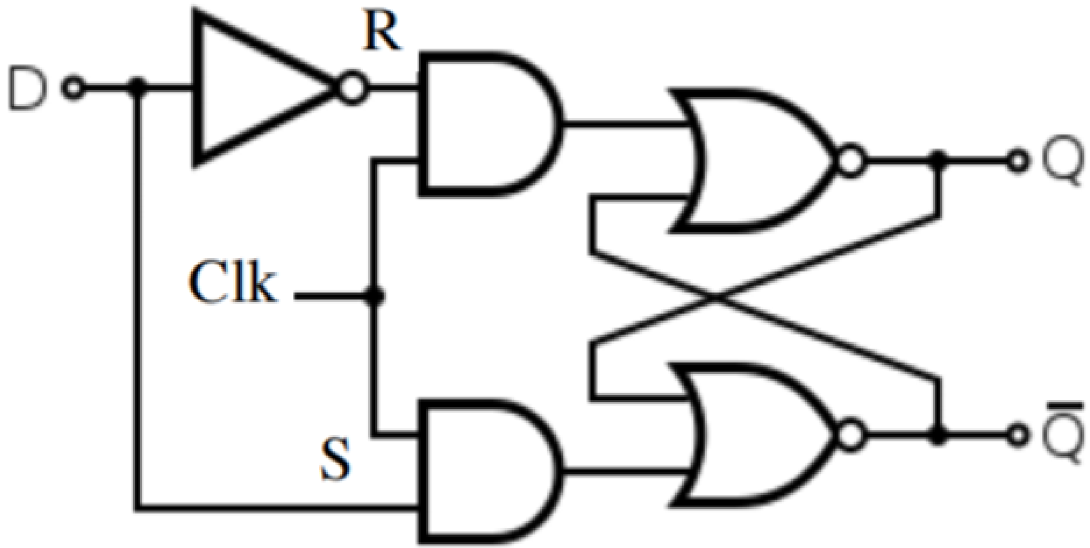
the gate is open

Clk=1

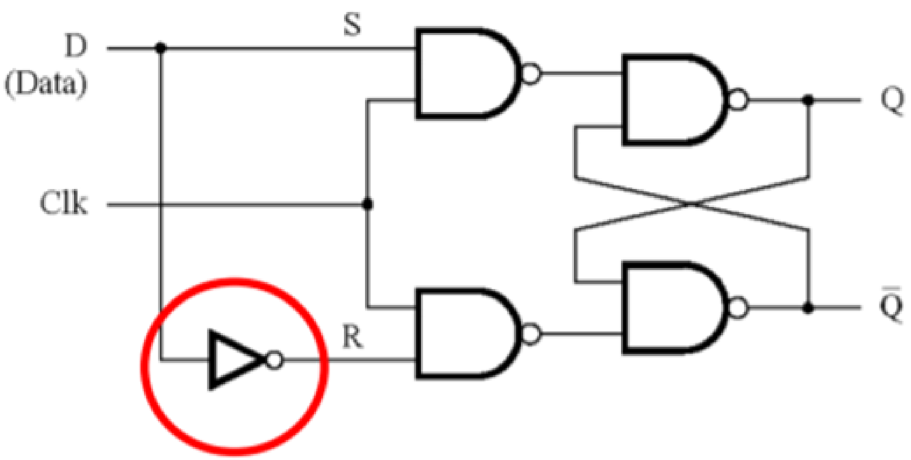


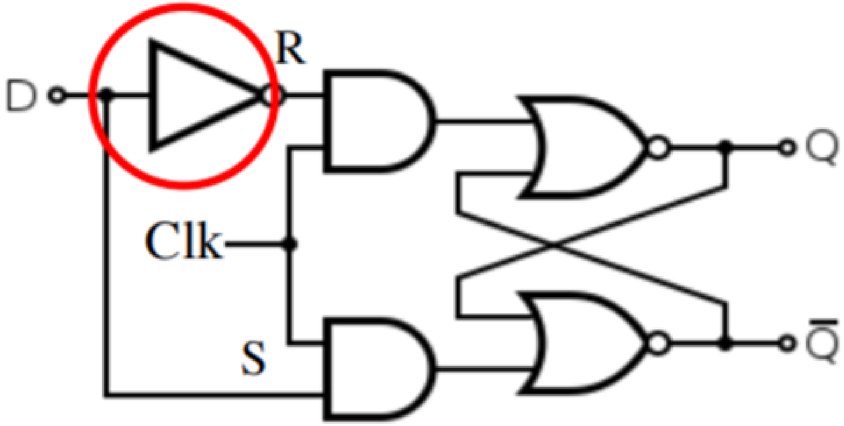
The output Q is equal to D in these intervals.

Circuit Diagram for the Gated D Latch (with the latch implemented using NORs)



Circuit Diagram for the Gated D Latch (with the latch implemented using NORs)





The NOT gate is now in a different place.
Also, S and R are swapped.

Small Detour:

Clock Triggering

for Sequential Circuits(Flipflops)

Level-Sensitive v.s. Edge-Triggered

flip-flop :

Two ways to Design

- ff Responds to Level of Clock
  ⇓
  Level-Triggered flipflop
- ff Respondes to Edge of Clock
  ⇓
  Edge-Triggered Flipflop

flip-flop :

Two ways to Design

→ ff Responds to Level of Clock
1. Positive level : +ve level Triggered ff
2. Negative level : -ve " " ff

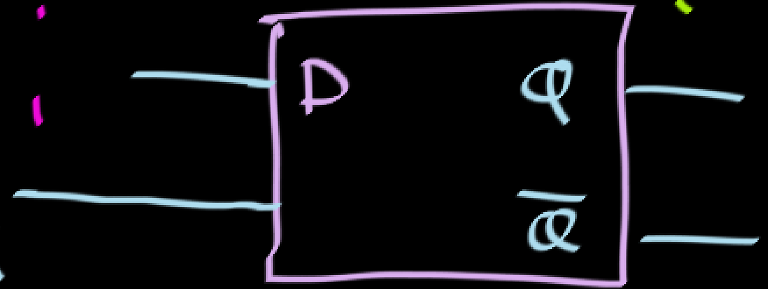
→ ff Respondes to Edge of Clock
1. Rising Edge : +ve Edge Triggered ff
2. falling Edge : -ve " " "

flip-flop :

4 ways to Design

1. Positive level Triggered flip flop
2. Negative level " "
3. +ve Edge " "
4. -ve Edge " "

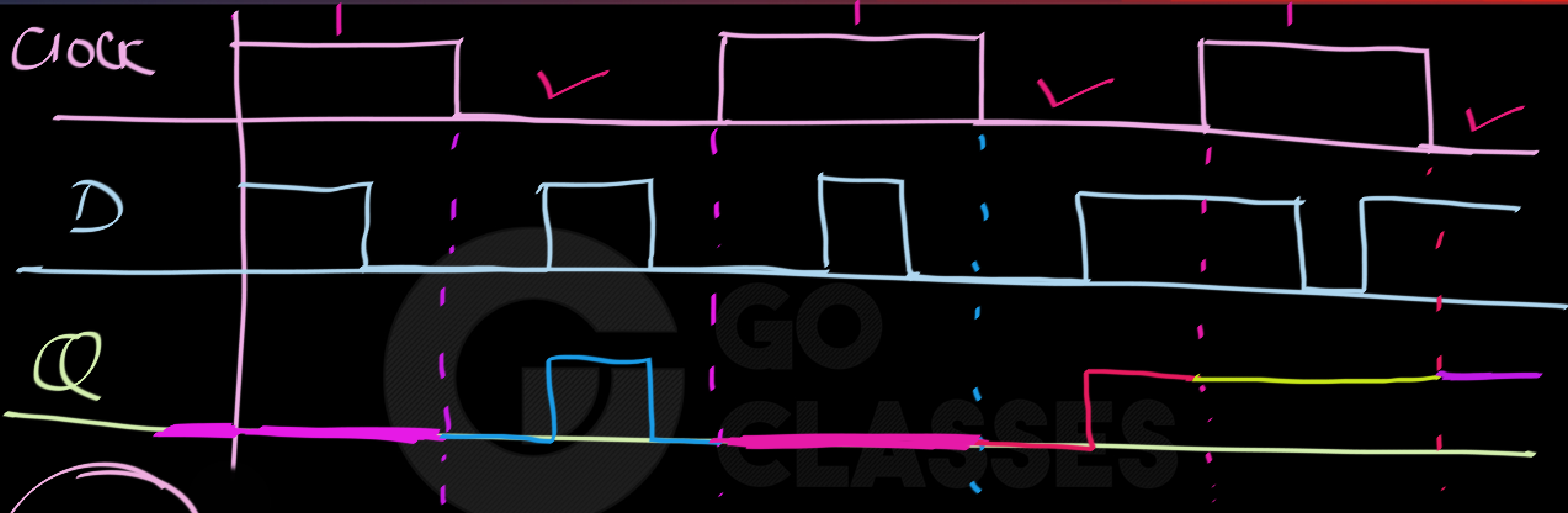
If D-FF is +ve level Triggered



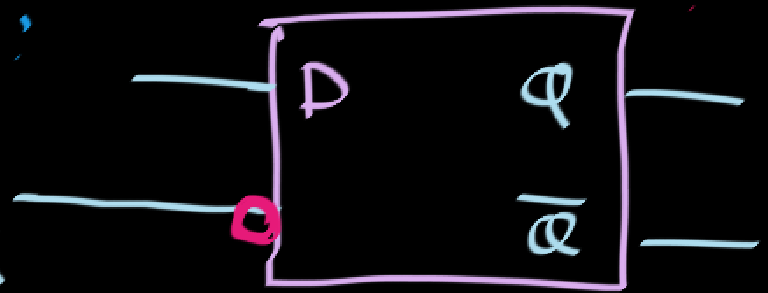
+ve level Triggered D-ff :

D-ff works when clock level = 1

Else D-ff Doesnot work (State or o/p Does not change)



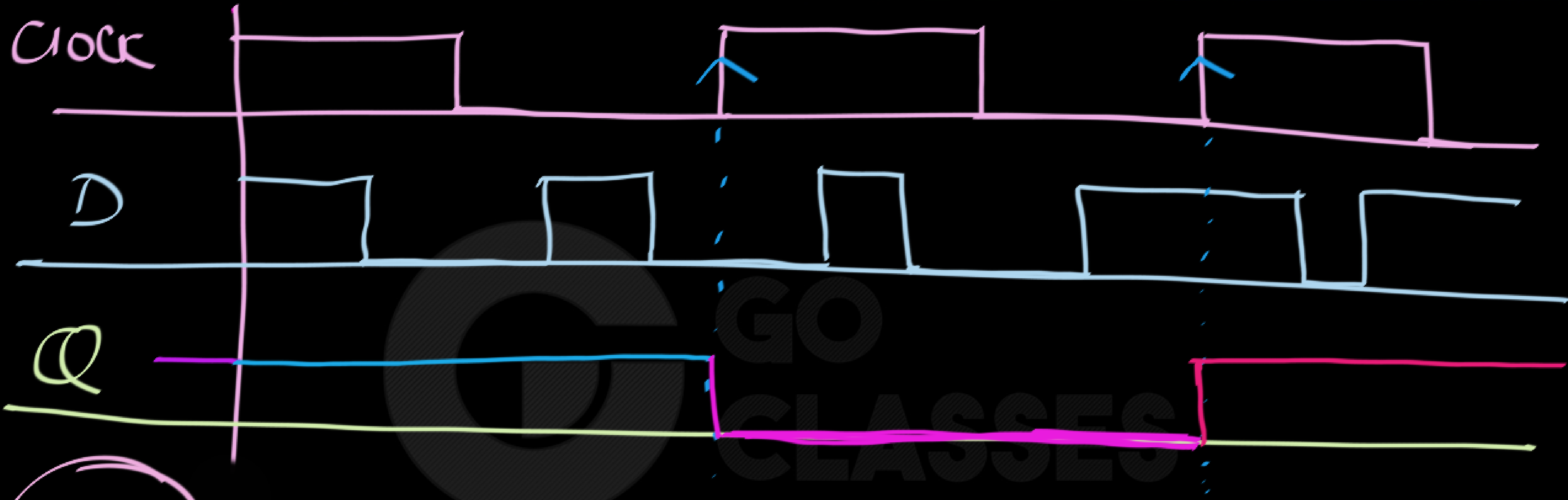
If D-ff is
-ve level Triggered



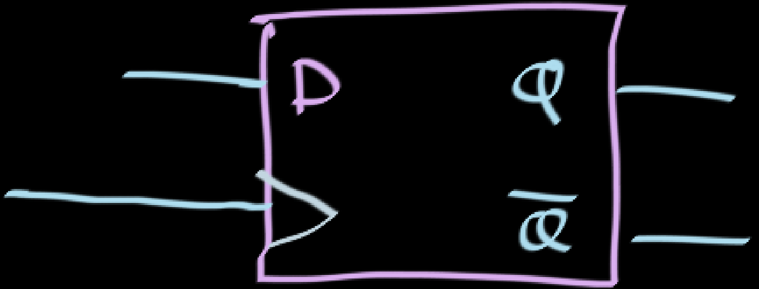
(-ve) level Triggered D-ff :

D-ff works when clock level = 0

Else D-ff Doesnot work (State or o/p Does not change)



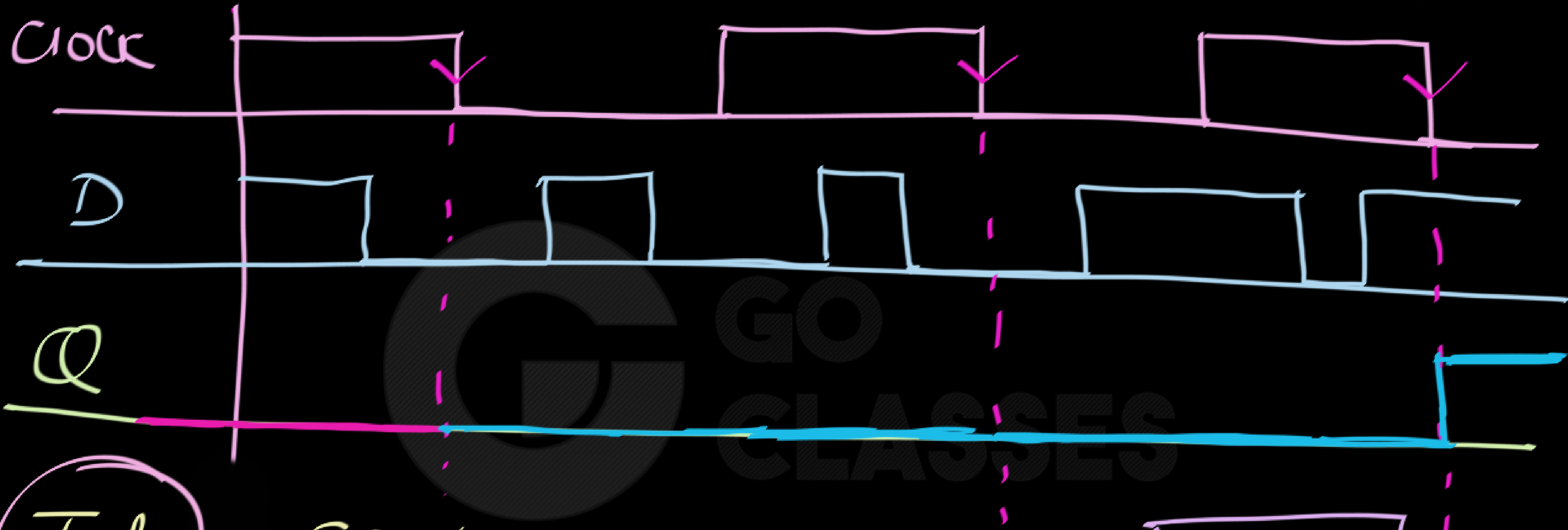
If D-ff is +ve Edge Triggered



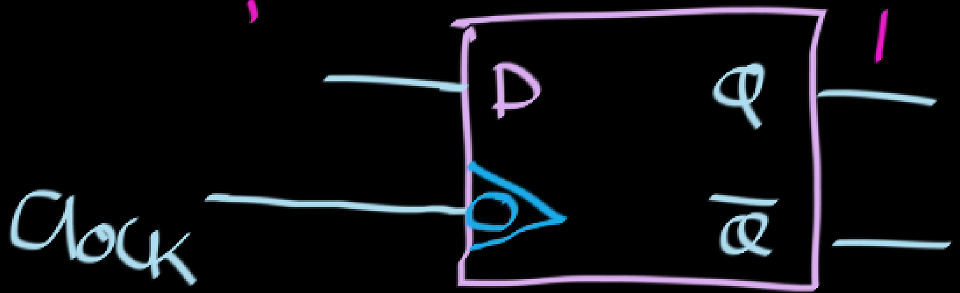
+ve Edge Triggered D-ff :

D-ff works only on Positive Edge

Else D-ff Doesnot work (State or o/p Does not change)



If D-FF is
-ve level Triggered



-ve Edge Triggered D-ff :

D-ff works only on Negative Edge

Else D-ff Doesnot work (State or o/p Does not change)

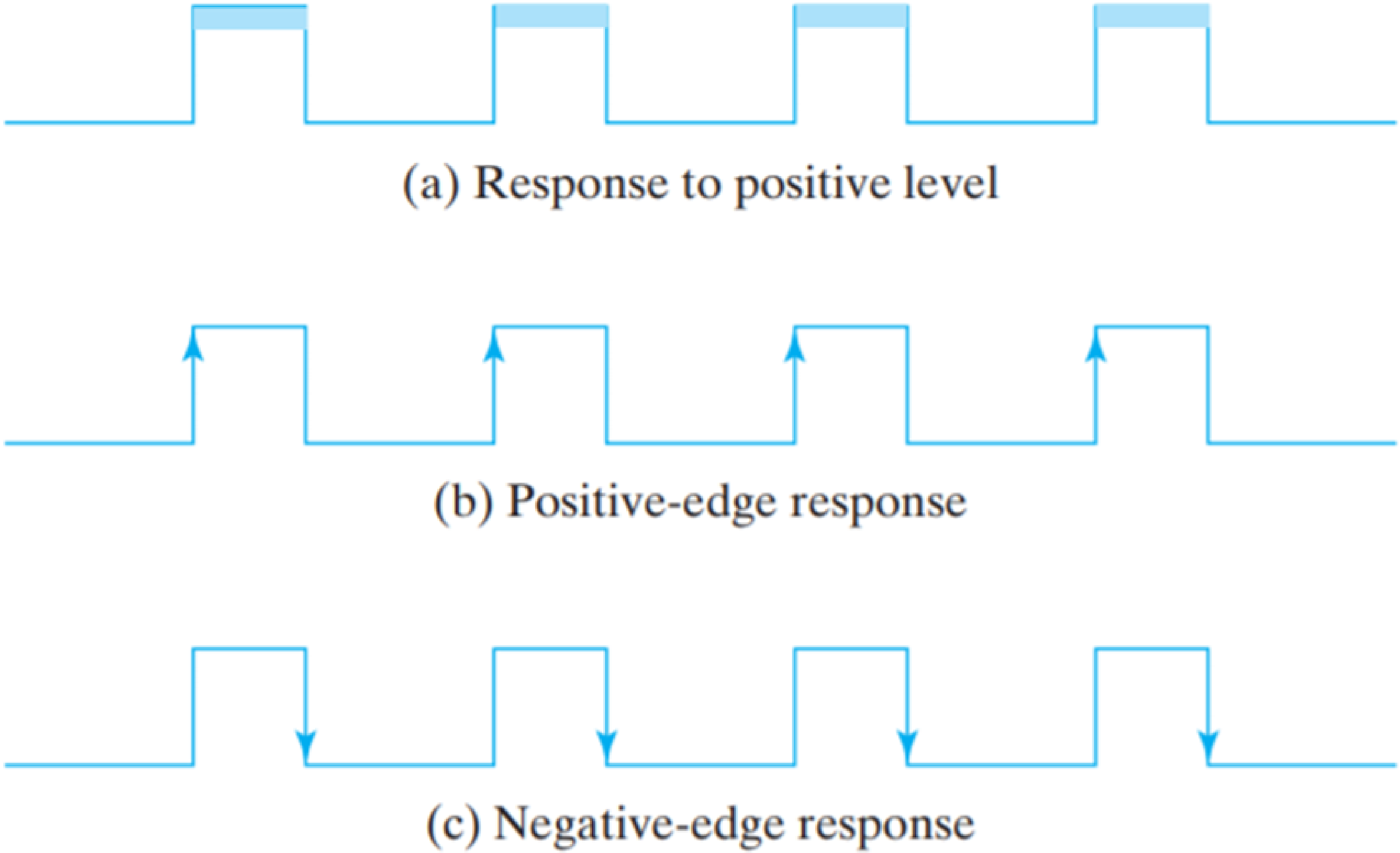
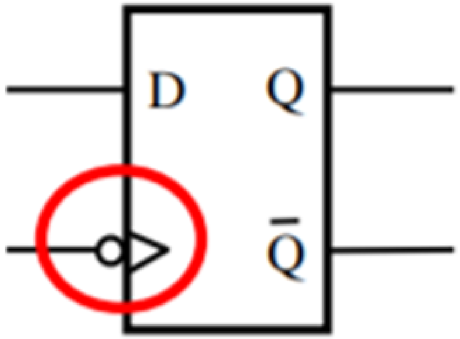


FIGURE 5.8
Clock response in latch and flip-flop

Motivation

In some cases we need to use a memory storage device that can change its state no more than once during each clock cycle.

Edge-Triggered D Flip-Flops



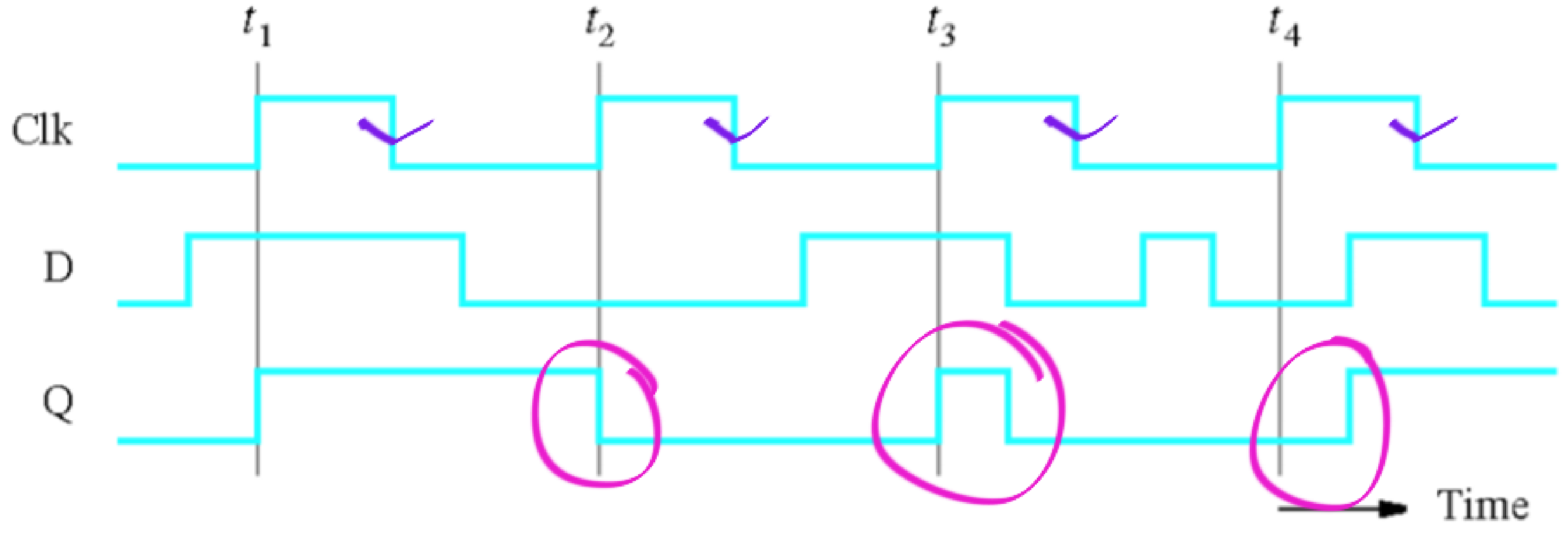
The > means that this is edge-triggered
The small circle means that is is the negative edge

Q: In the following Timing Diagram of D-FF, find the Clock Triggering of D-FF?

Timing Diagram for the Gated D Latch

−Ve Edge Triggered?

NO

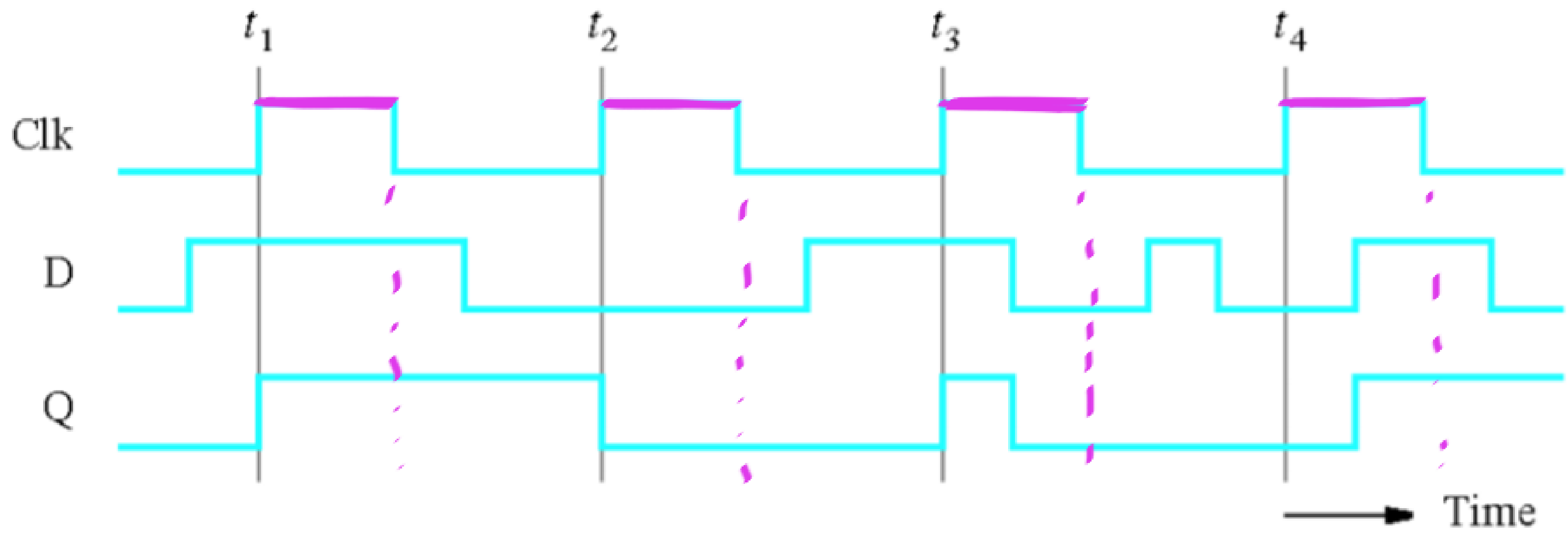


Timing Diagram for the Gated D Latch

GO Classes

[redacted]

+ve level Triggered.



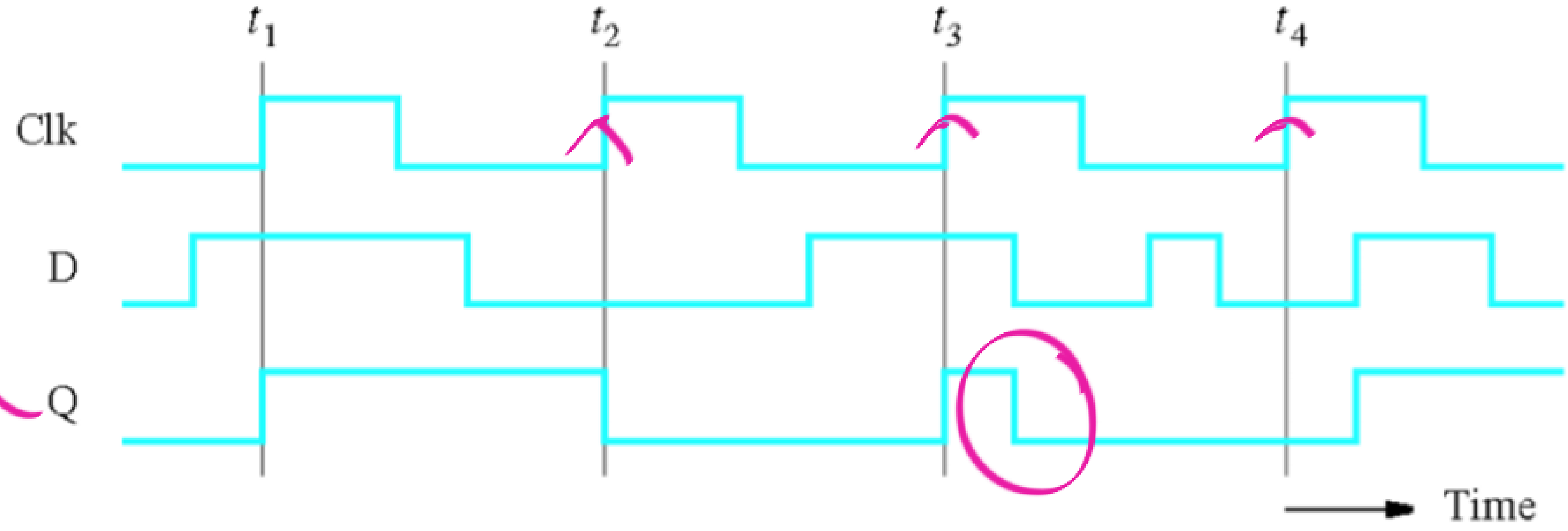
Timing Diagram for the Gated D Latch

GO Classes

+ve Edge Triggered?

No

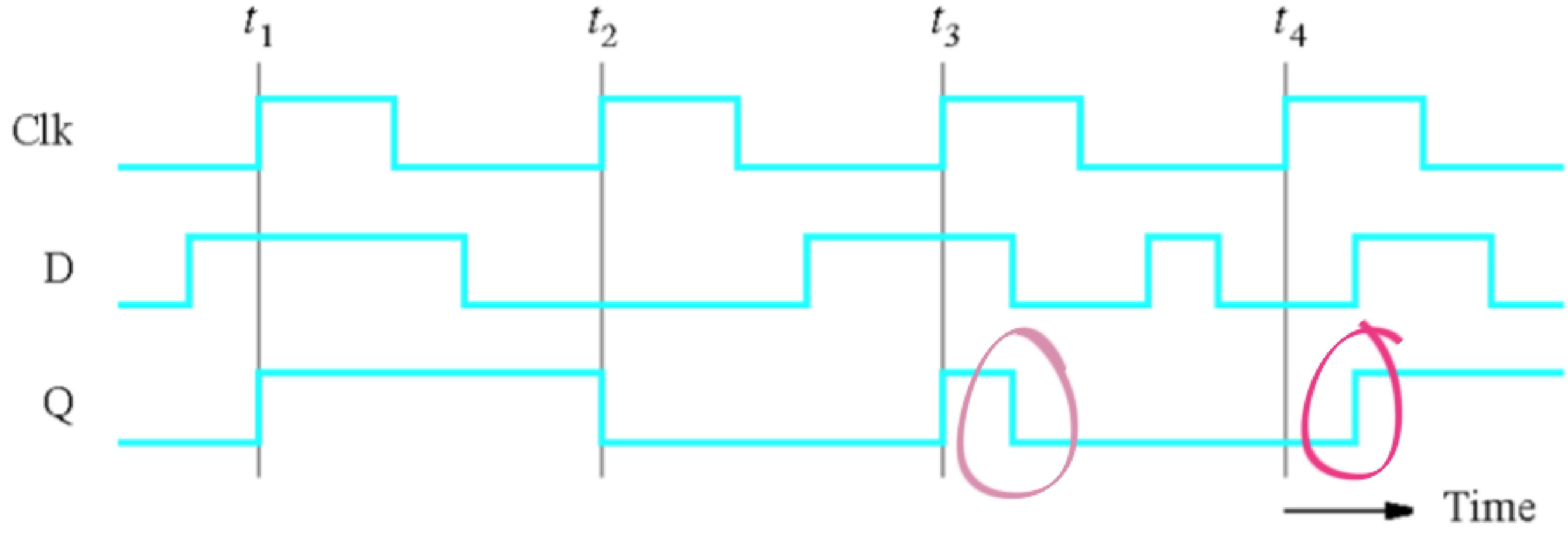
State



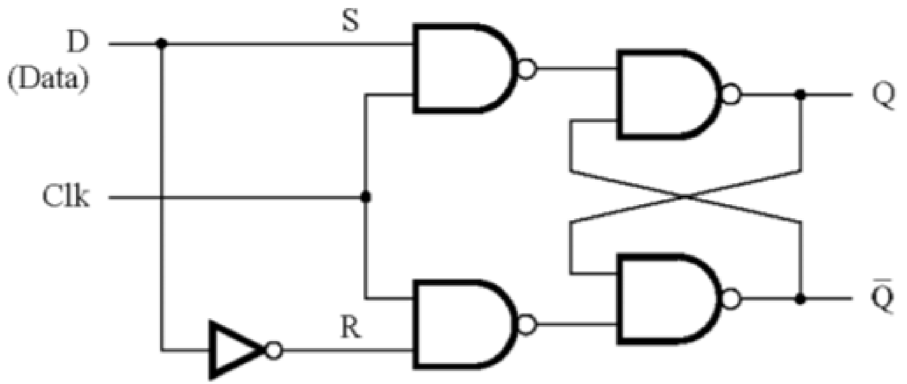
Timing Diagram for the Gated D Latch

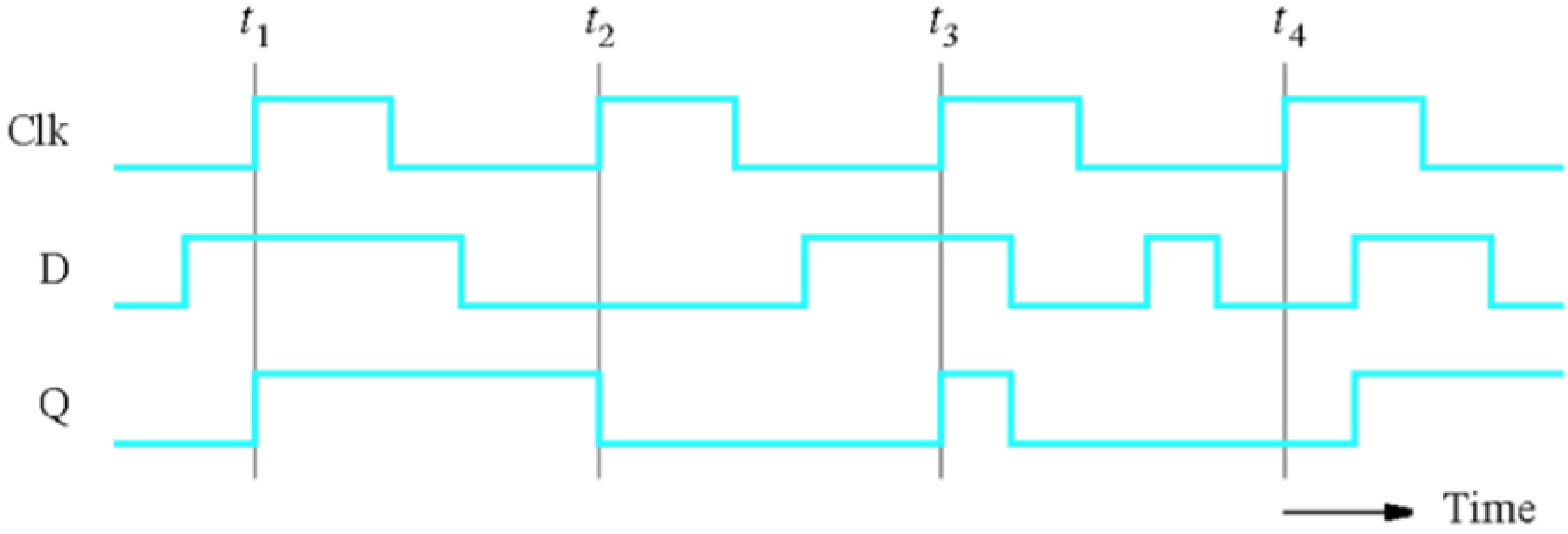
-ve level Triggered?

No



Timing Diagram for the Gated D Latch





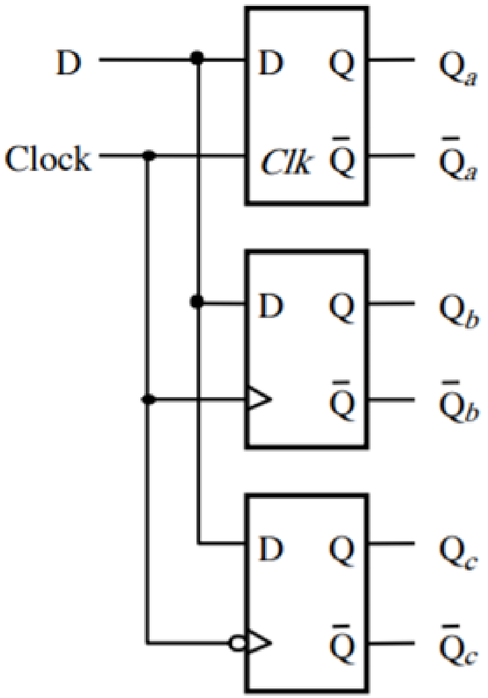
Note:

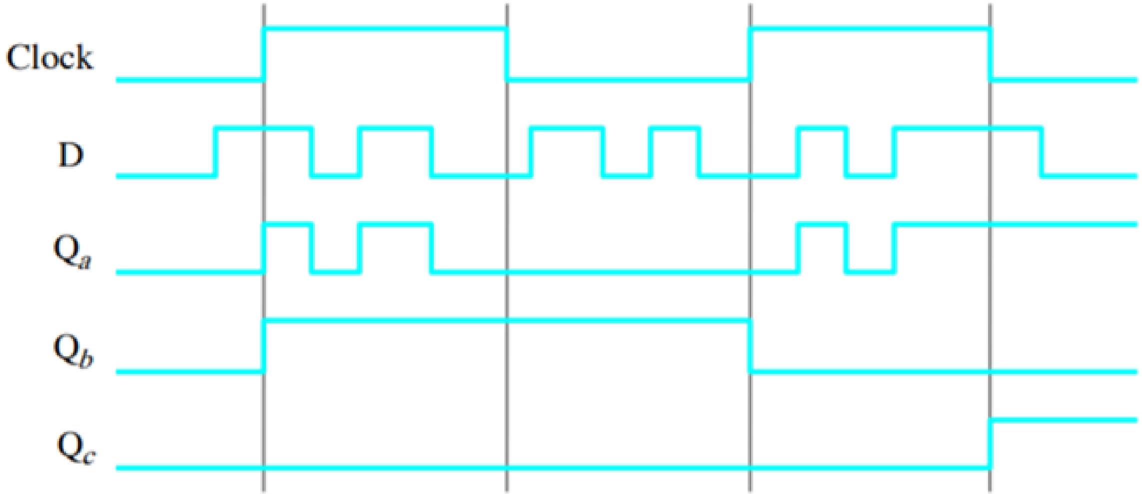
provided by user

Input can change anytime.

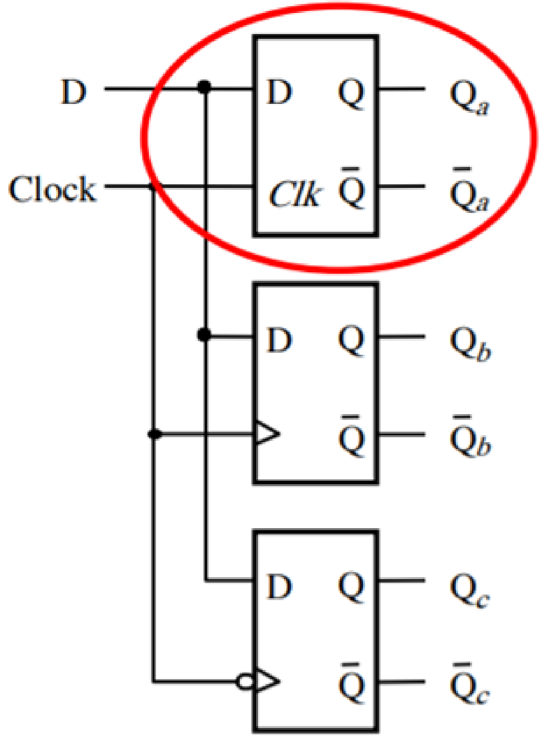
Triggering is related to o/p (state) change.

Comparison of level-sensitive and edge-triggered D storage elements



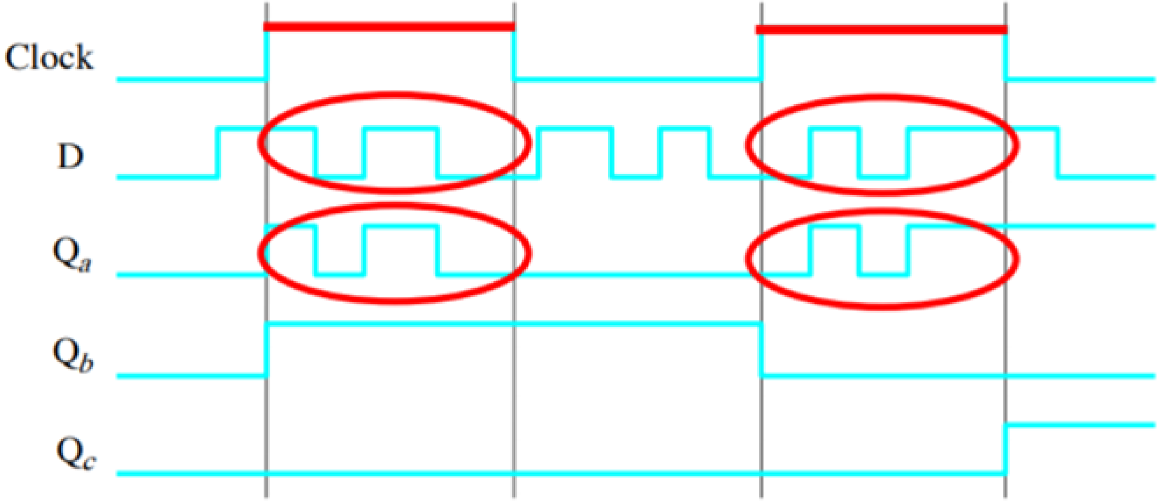


GO Classes

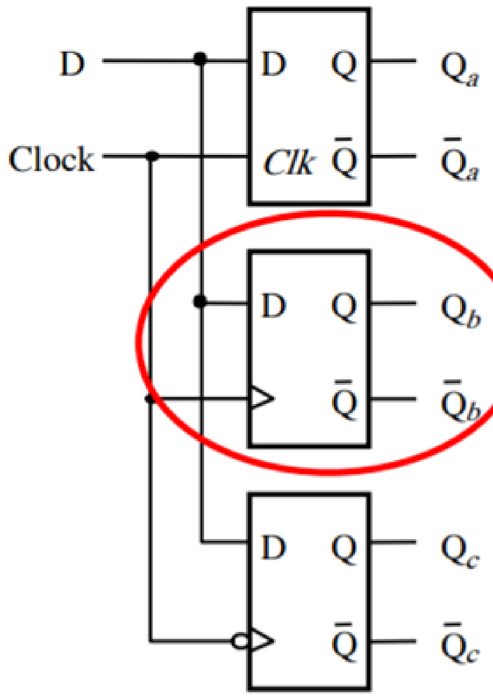


Comparison of level-sensitive and edge-triggered D storage elements

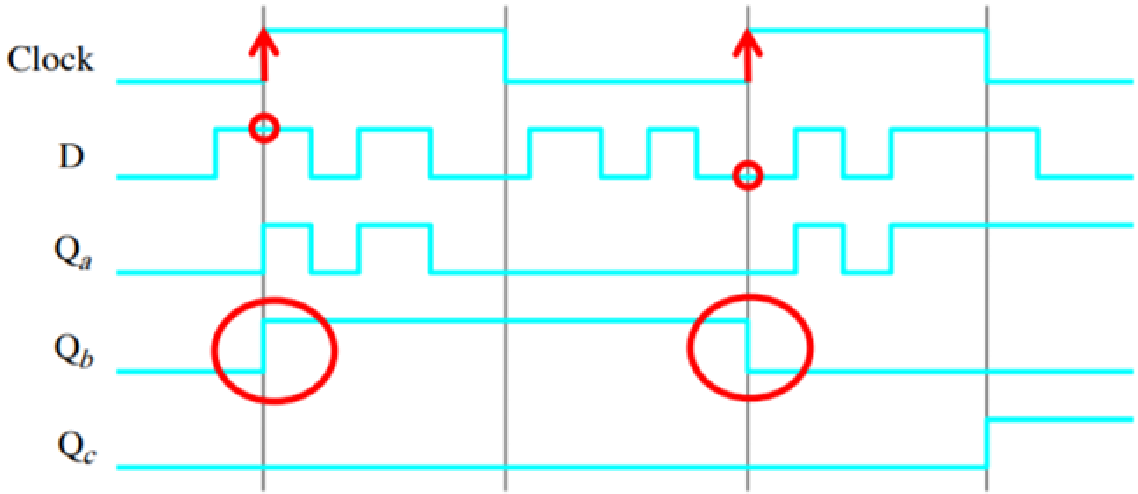
The D Latch is Level-Sensitive
(the output mirrors the D input when Clk=1)



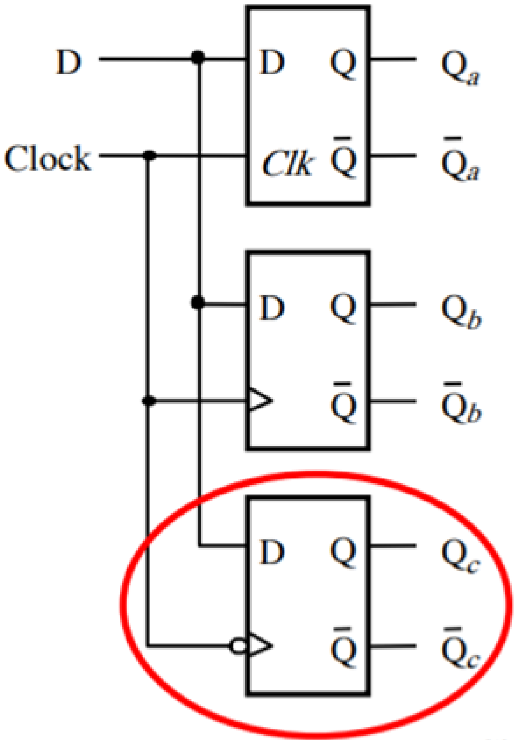
Comparison of level-sensitive and edge-triggered D storage elements



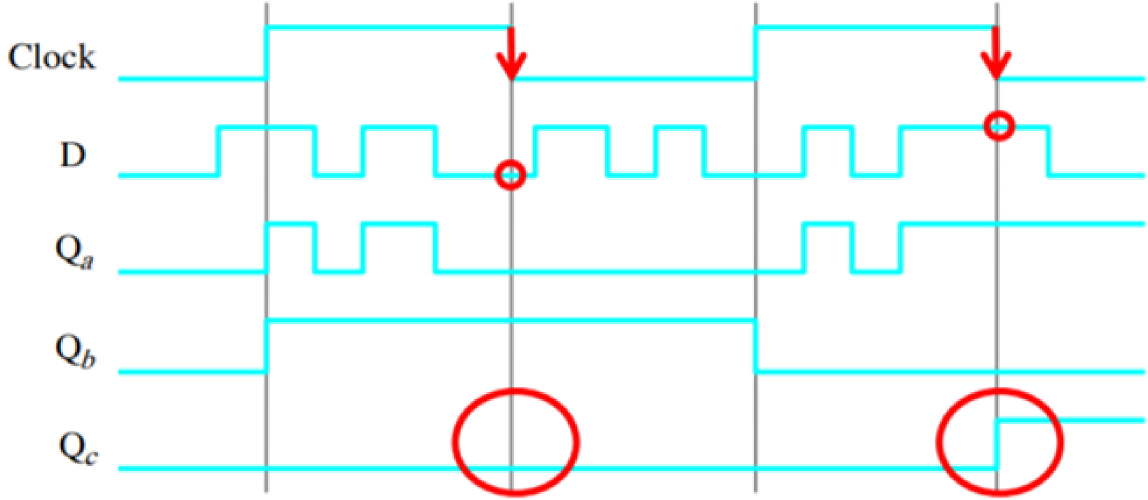
Positive-edge-triggered D Flip-Flop
(the output is equal to the value of D right at the positive edge of the clock signal)



Comparison of level-sensitive and edge-triggered D storage elements



Negative-edge-triggered D Flip-Flop
(the output is equal to the value of D right at the negative edge of the clock signal)



Note:

for any flipflop, "by Default"

Edge Triggering is used.

why? ⇒ within one time period (cycle) of clock, State (o/p) can change only once.

D - ff :

$$Q_{t+1} = f(\text{clock}, D, Q_t)$$

for implementation, equation :

$$Q_{t+1} = f(\cancel{\text{clock}}, D, Q_t)$$

Reason:

D-ff Can have any Triggering.

D-ff (−ve level Triggered)

Clock	D	Q_{t+1}
1	X	Q_t → No Response
0	0	0
0	1	1

} on −ve level D-ff works.

D-ff (+ve level Triggered)

Clock	D	Q_{t+1}
0	X	Q_t
1	0	0
1	1	1

$\rightarrow$ No Response

working

D-ff (+ve Edge Triggered)

Clock	D	Q_{t+1}
0	x	Q_t
1	x	Q_t
↑	0	0
↑	1	1
↓	x	Q_t

No Response

⟹ work

D-ff (-ve Edge Triggered)

Clock	D	Q_{t+1}	
0/1/↑	X	Q_t	No Response
↓	0	0	working
↓	1	1	

Next Topic:

S R
| |
J K

3. JK Flipflop

Similar to SR, but 11 Input Combination usable

Characteristic table / State table :

J — S, K — R

J (S (Set))	K (R (Reset))	Q_{t+1}	
0	0	Q_t	Retain / No change
0	1	0	Reset
1	0	1	Set
1	1	$\overline{Q_t}$	Complement / Toggle

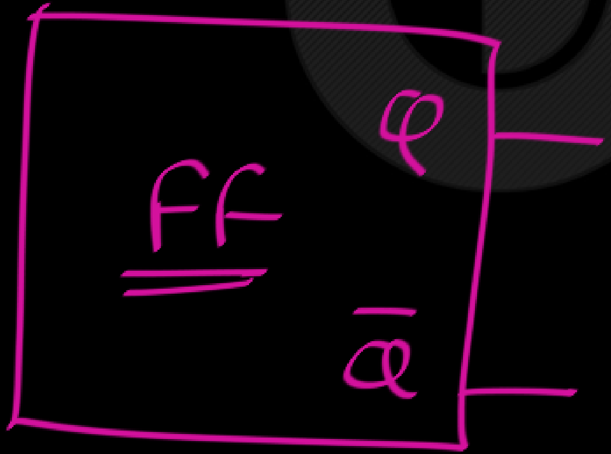
SR

Flipflop: $\Rightarrow$ one bit Storage Device

$\Downarrow$

Clocked latch

- Store 0
- Store 1
- Unchanged / keep / Retain
- Complement / Toggle

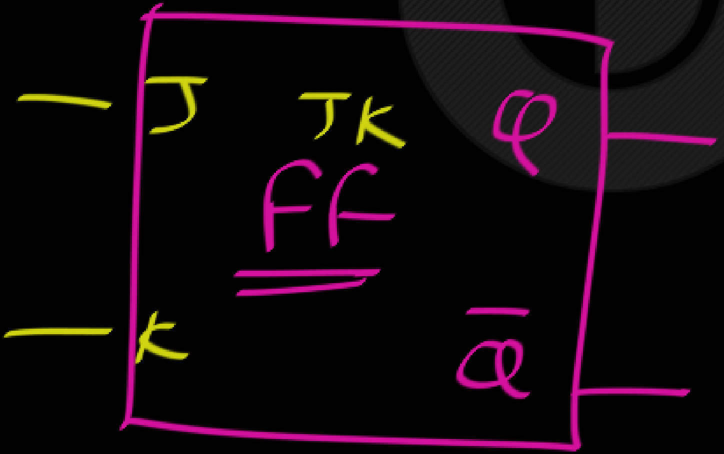


JK Flip flop :- $\Rightarrow$ one bit Storage Device

$\Downarrow$

Clocked latch

	J	K
Store 0	0	1
Store 1	1	0
Unchanged / Keep / Retain	0	0
Complement / Toggle	1	1



Clock Triggering :

any of these:

- +ve level Triggered JK FF
- −ve " " " "
- +ve Edge " " "
- −ve " " " "

by Default

$$Q_{t+1} = f(J, K, Q_t)$$

K-map of Q_{t+1}

$K=1, J=0$

$J=1$

Q_t \ JK				
$Q_t=0$	0	0	1	1
$Q_t=1$	1	0	0	1

$J=0$
$K=0$

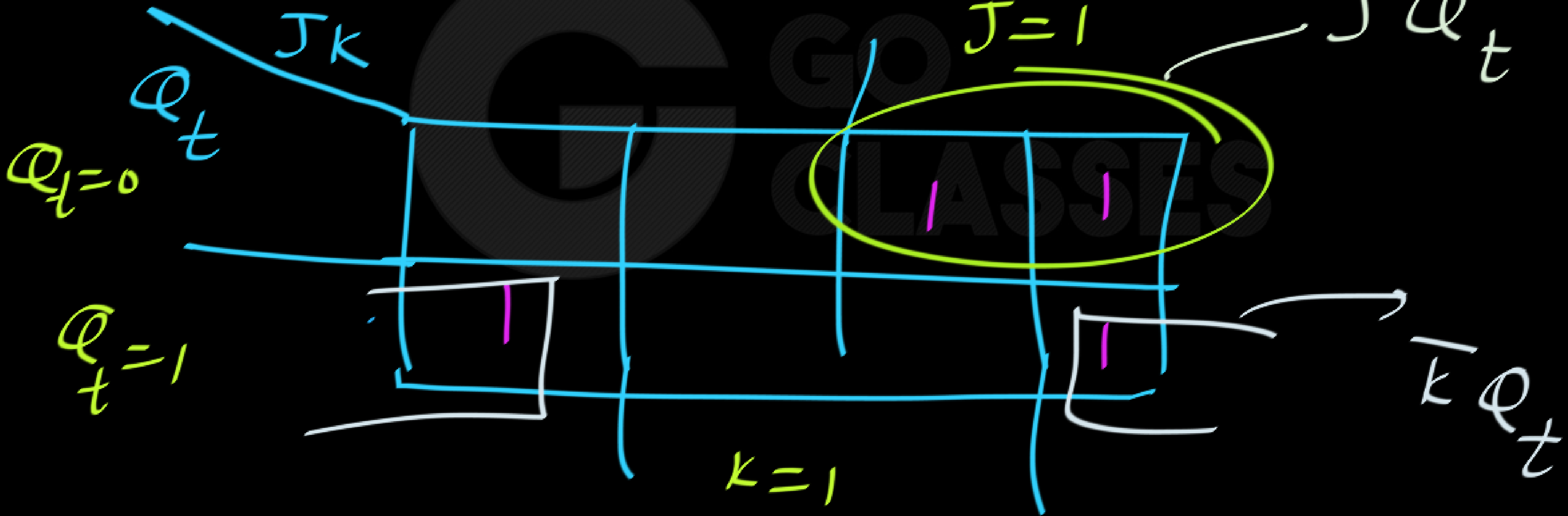
$K=1$

$J=1$
$K=0$

J	K	Q_{t+1}
0	0	Q_t
1	0	1
0	1	0
1	1	$\overline{Q_t}$

$$Q_{t+1} = f(J, K, Q_t)$$

K-map of Q_{t+1}



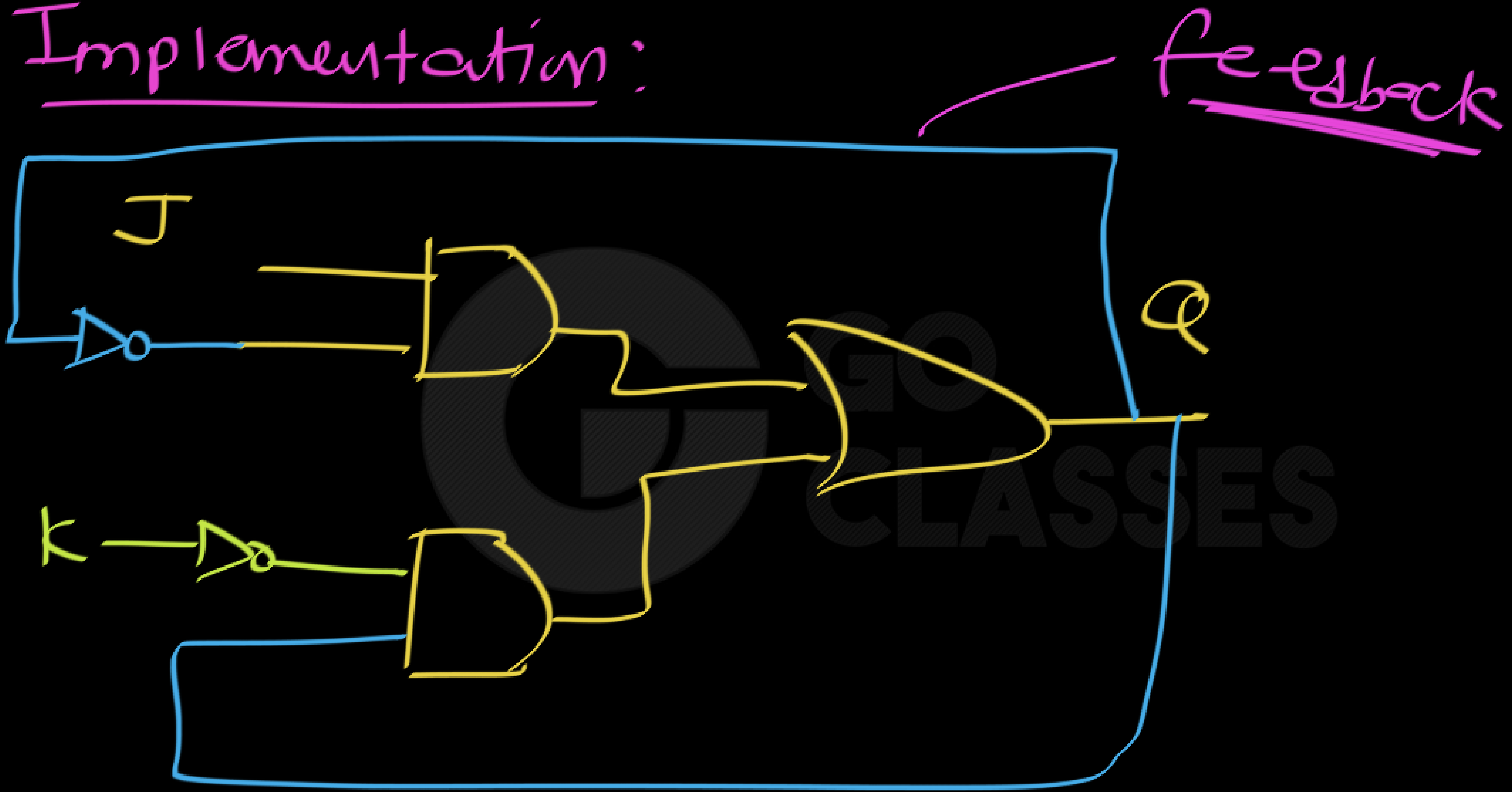
State equation (output equation) :

(characteristic equation)

$$Q_{t+1} = J\overline{Q_t} + \overline{K}Q_t$$

$$Q_n = J\overline{Q} + \overline{K}Q$$

Implementation:



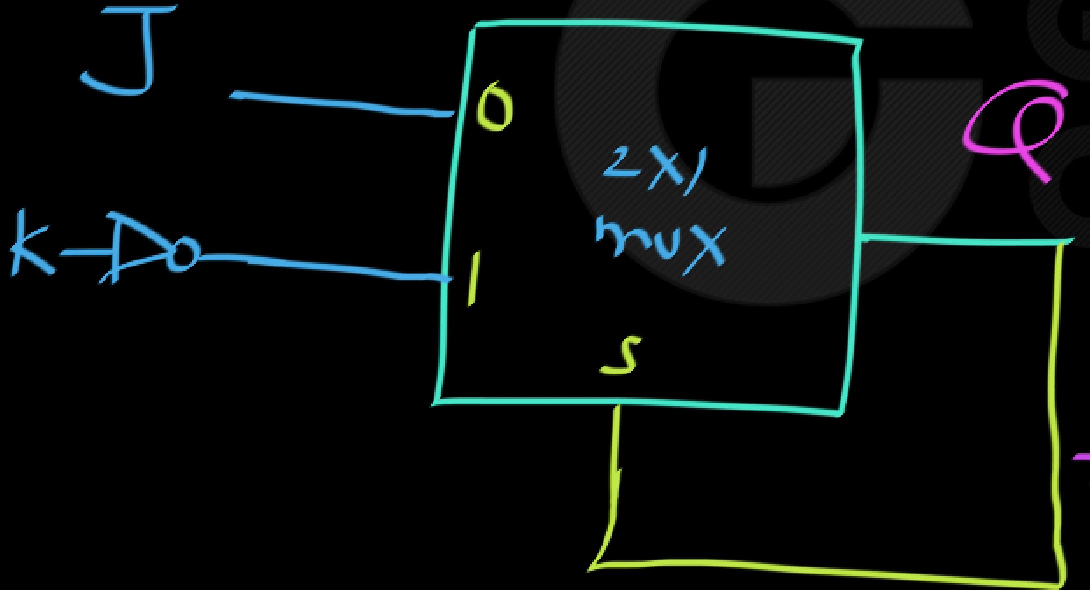
mux equation: $\Rightarrow f = \bar{s}\underline{\underline{a}} + s\underline{\underline{b}}$

Jk ff:

$Q_n = \bar{Q}J + Q\bar{K}$

Implementation: using mux:

$$Q_n = J\overline{Q} + \overline{K}Q$$



⇒ feedback

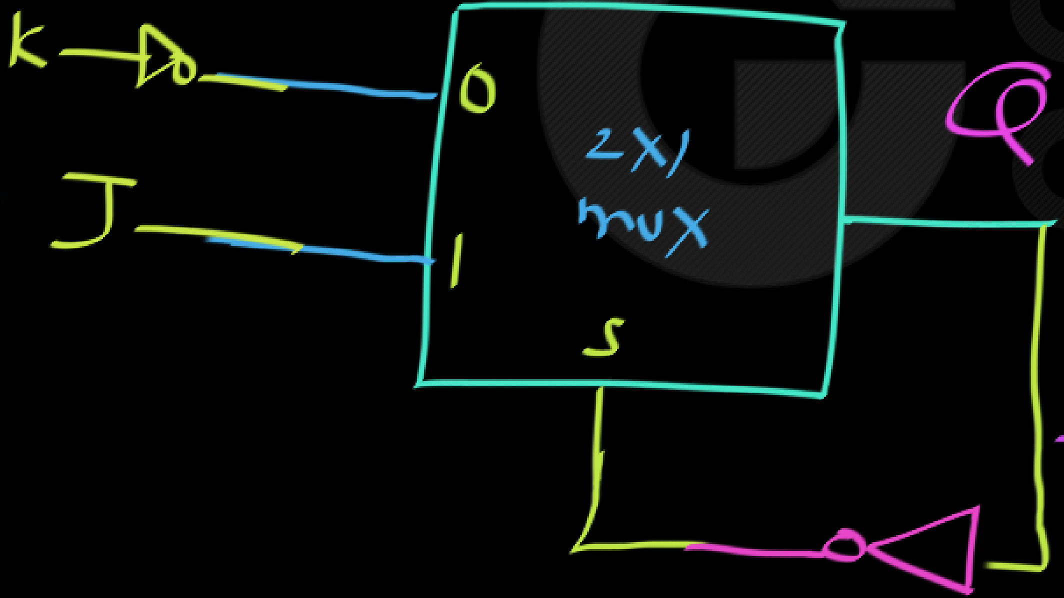
Q_n, Q

Not Diff. o/p lines

→ same o/p lines

Implementation: using mux:

$$Q_n = J\overline{Q} + \overline{K}Q$$



$\Rightarrow$ feedback

Q_n, Q

Not Diff. o/p lines

$\rightarrow$ Same o/p lines

5 year old

You

kid Q_5

Adult Q_{18}

Same person,

Different times

Sequential Circuit: → feedback

⇓

feeds the present o/p back as input.

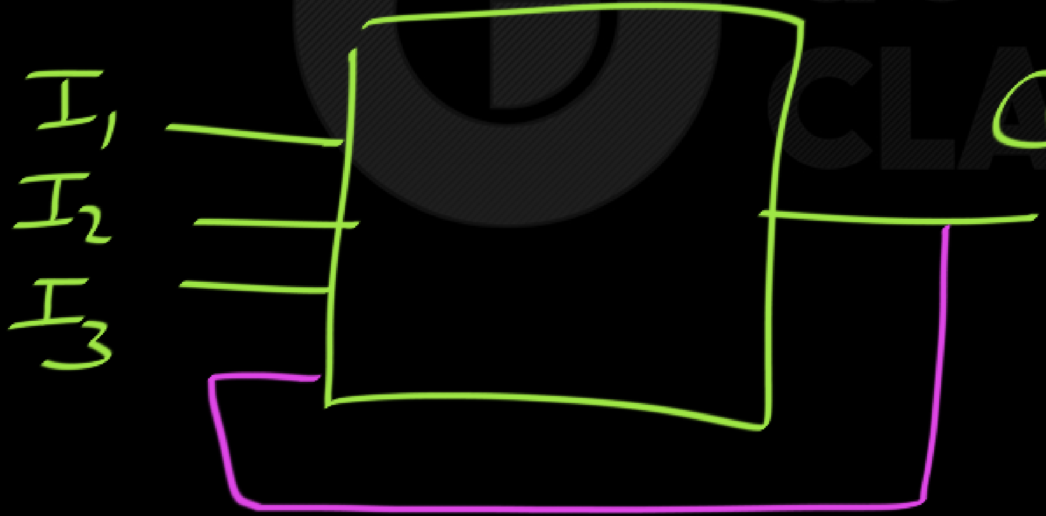
flip flop
Counter
Register
finite state machines

Sequential ckt:

present input

o/p

$$\text{output } Q_n = f(I, Q)$$



$$Q = f(I_1, I_2, I_3, Q)$$

Truth Table:

$Q_n = f(J, K, Q)$

8 Rows

$Q^+ = Q_n = Q_{t+1}$

J	K	Q	$Q^+ = Q_n = Q_{t+1}$	
0	0	0	0	Retain
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	

Excitation Table:

$Q \rightarrow Q_n$	J K
$0 \rightarrow 0$	Retain / Reset
$0 \rightarrow 1$	Set / Toggle
$1 \rightarrow 0$	Reset / Toggle
$1 \rightarrow 1$	Retain / Set

Excitation Table :

Q	Q_n	J	K
0	0	0	0
		0	1
0	1	1	0
		1	1
1	0	0	1
		1	1
1	1	0	0
		1	0

Excitation Table:

Q	Q_n	J	K
0 →	0	0	X
0 →	1	1	X
1 →	0	X	1
1 →	1	X	0

$J = f(Q, Q_n)$

$Q_n \backslash Q$	Q'	Q
Q_n'		X
Q_n	1	X

K-map for J

$$\boxed{J = Q_n}$$

$K = f(Q, Q_n)$

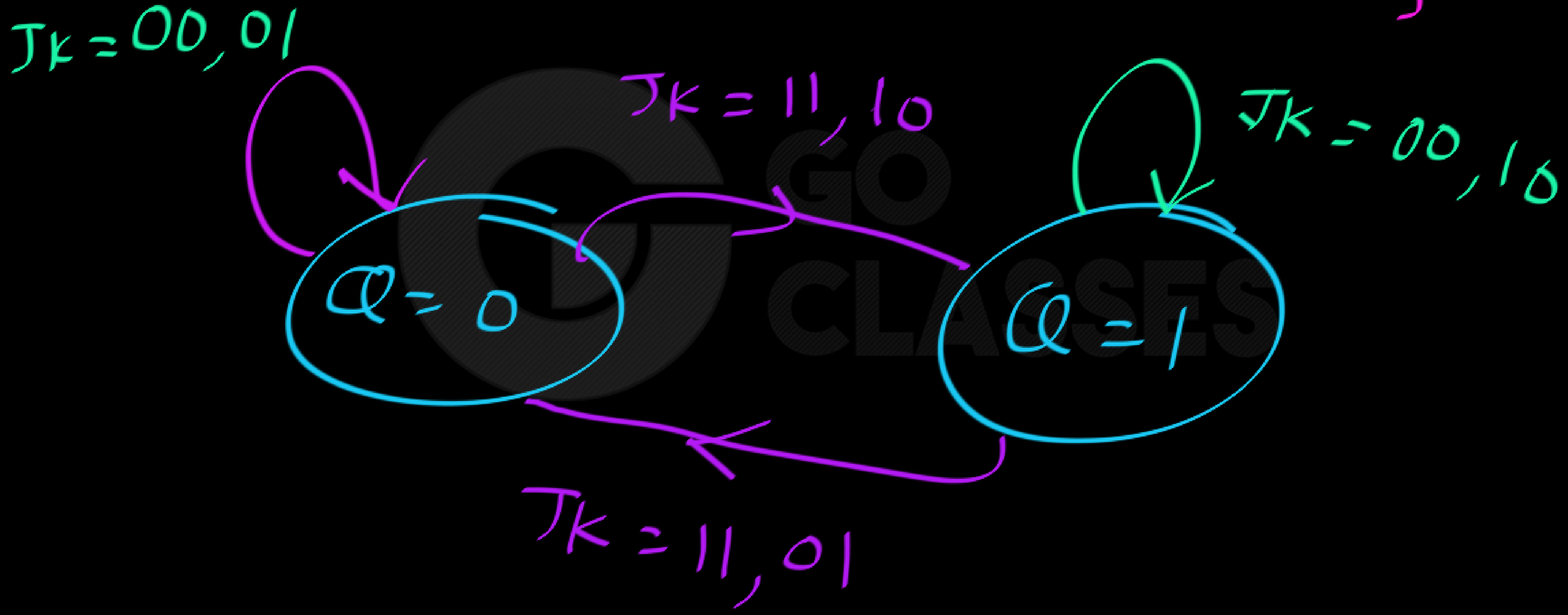
$Q_n \backslash Q$	Q	Q
Q_n	X	1
Q_n	X	

K-map for K

$\boxed{K = \overline{Q_n}}$

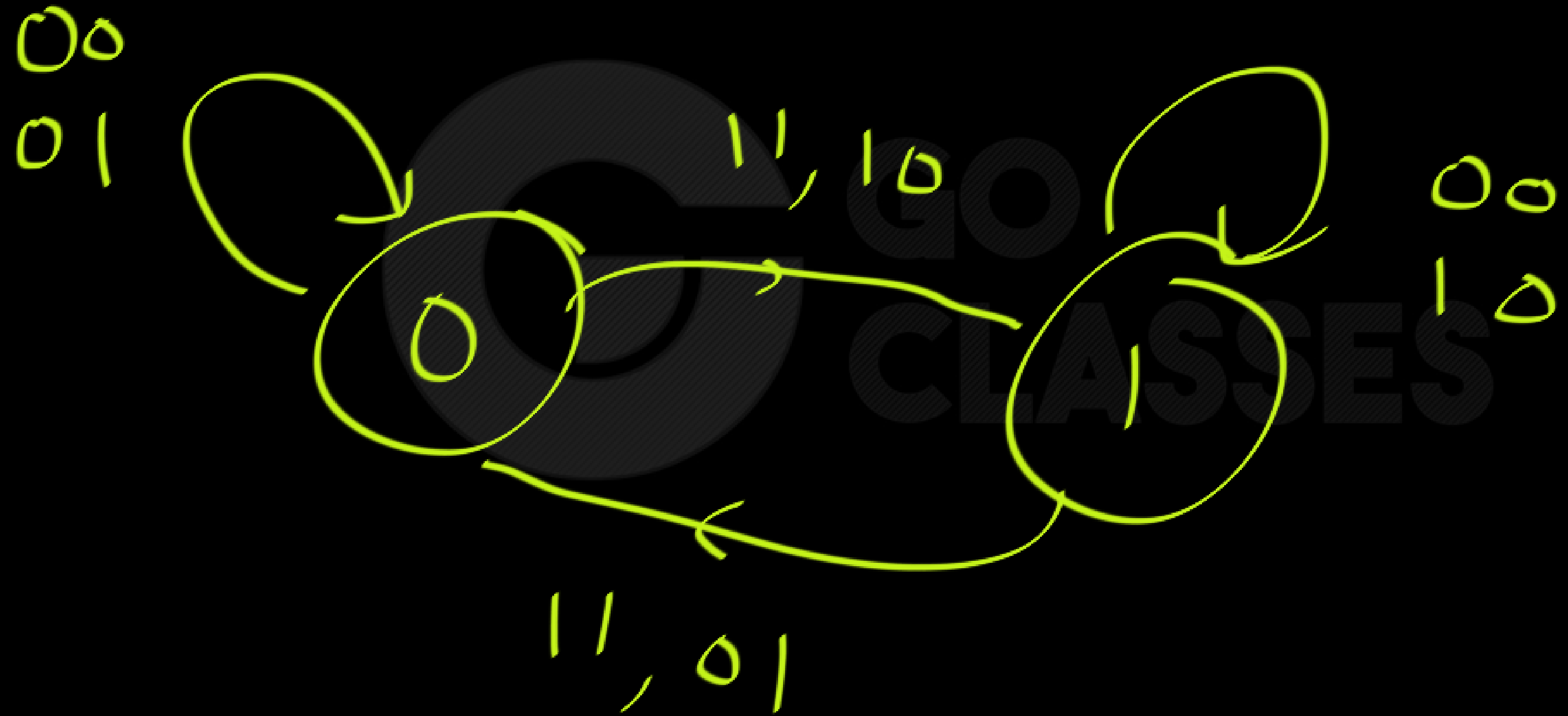
State Diagram:

JK - ff ⇒ Two states
0 , 1



State Diagram:

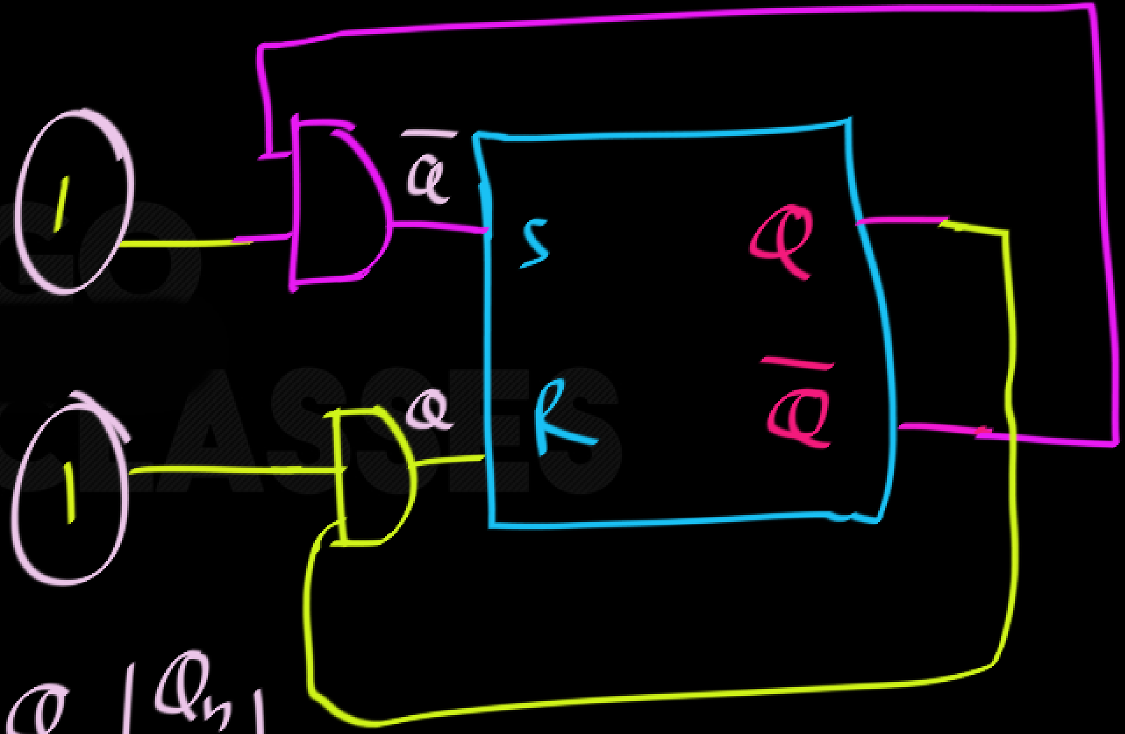
JK - FF ⇒ Two states
0 , 1



Implementation using SR :

Idea:

J	K	Q_n
1	1	$\bar{Q}$



Q	Q_n
0	1
1	0

wanted

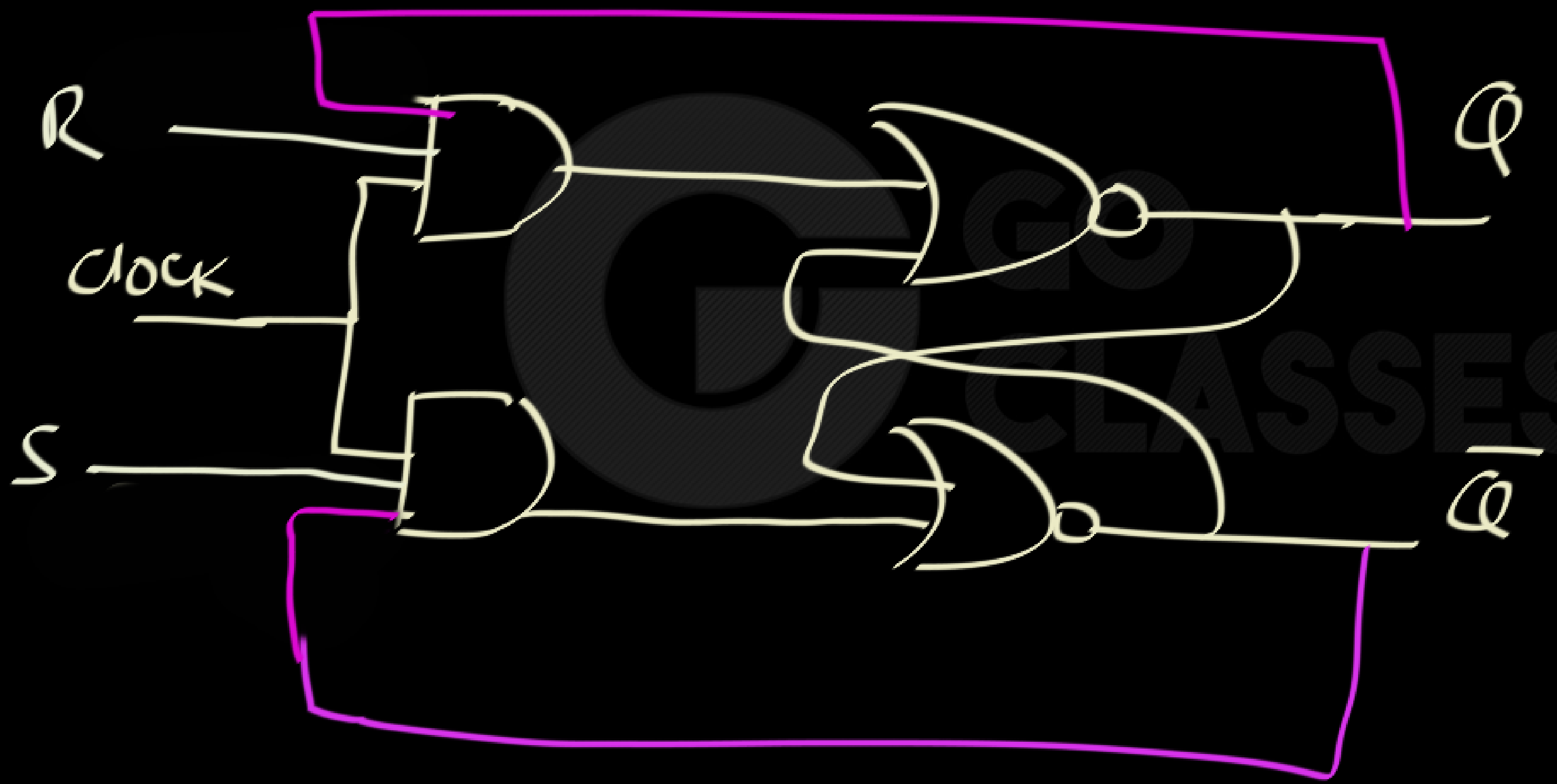
SR to JK :

$Q \longrightarrow$ with R

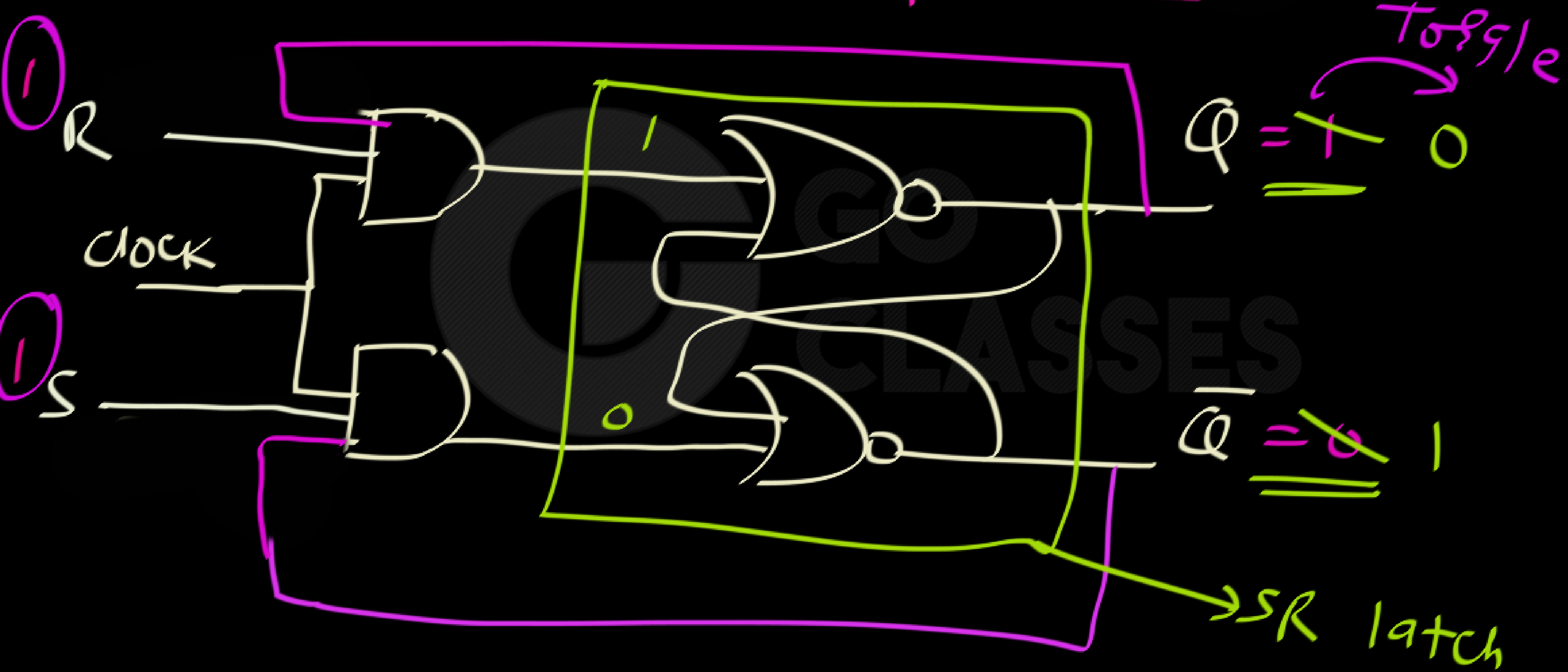
$\overline{Q} \longrightarrow$ with S

SR (of Nor) :

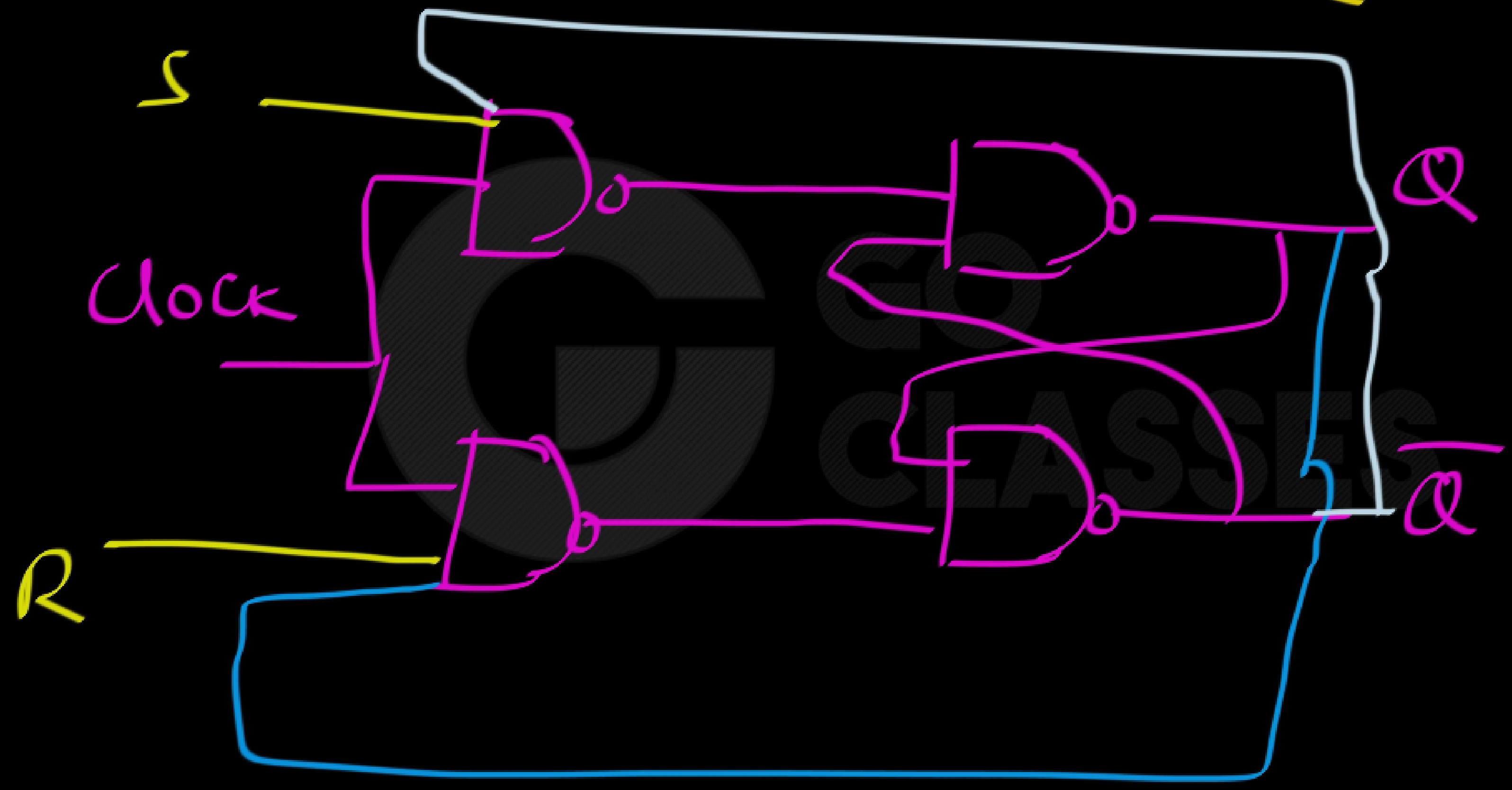
Implementation using SR (of NOR) :



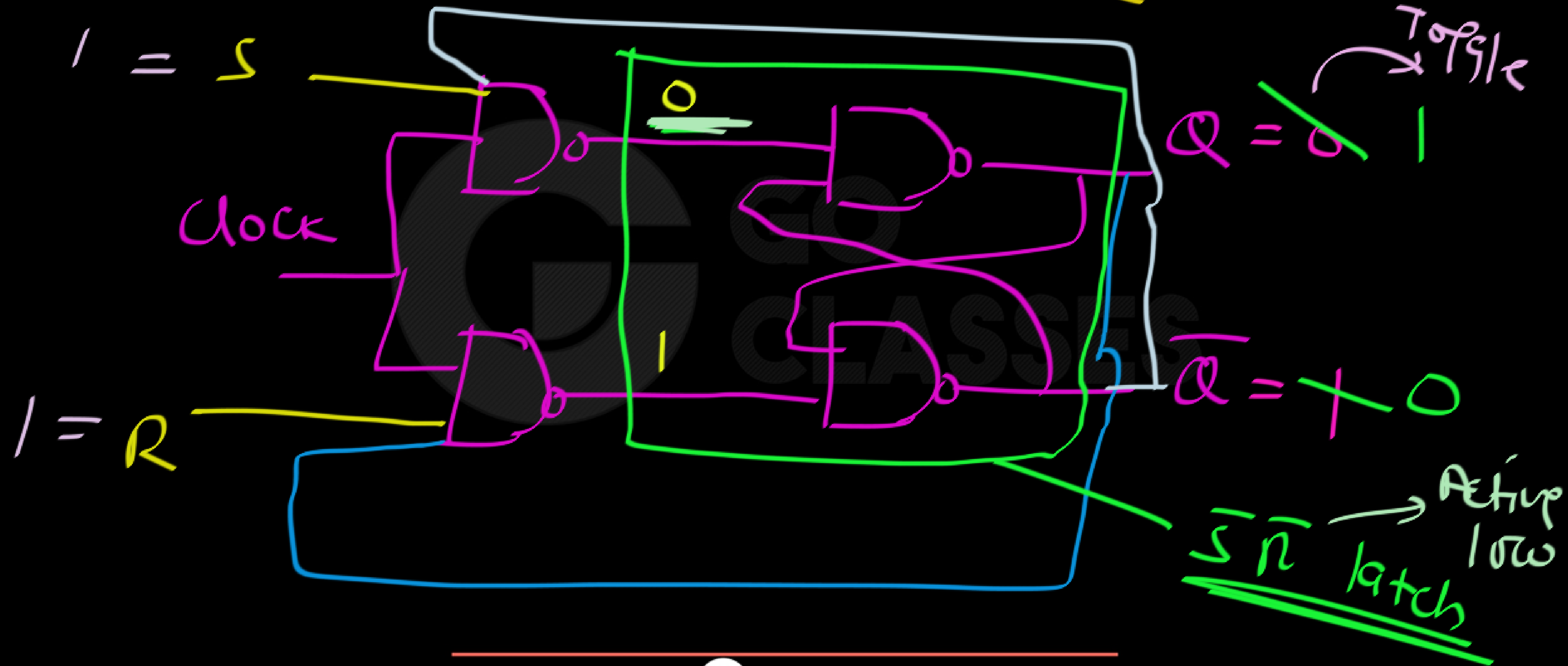
Implementation using SR (of NOR):



JK implementation using SR (of NAND)



JK implementation using SR (of NAND)



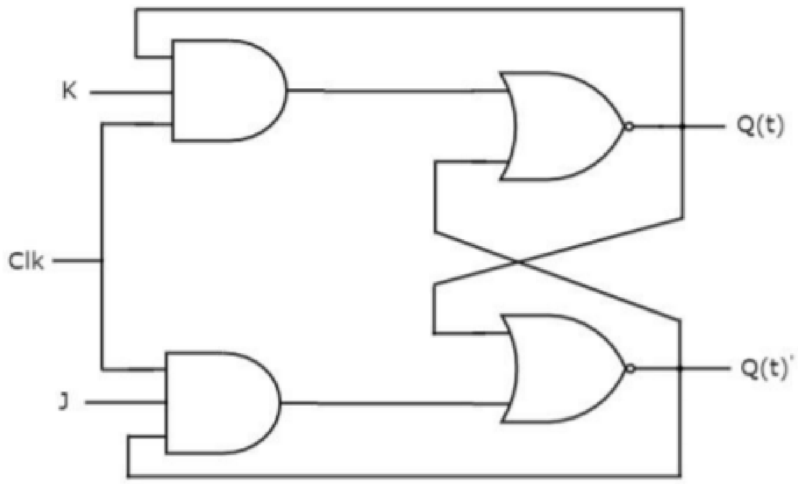
SR to JK ?

$Q \xrightarrow{\text{feedback}}$ with R

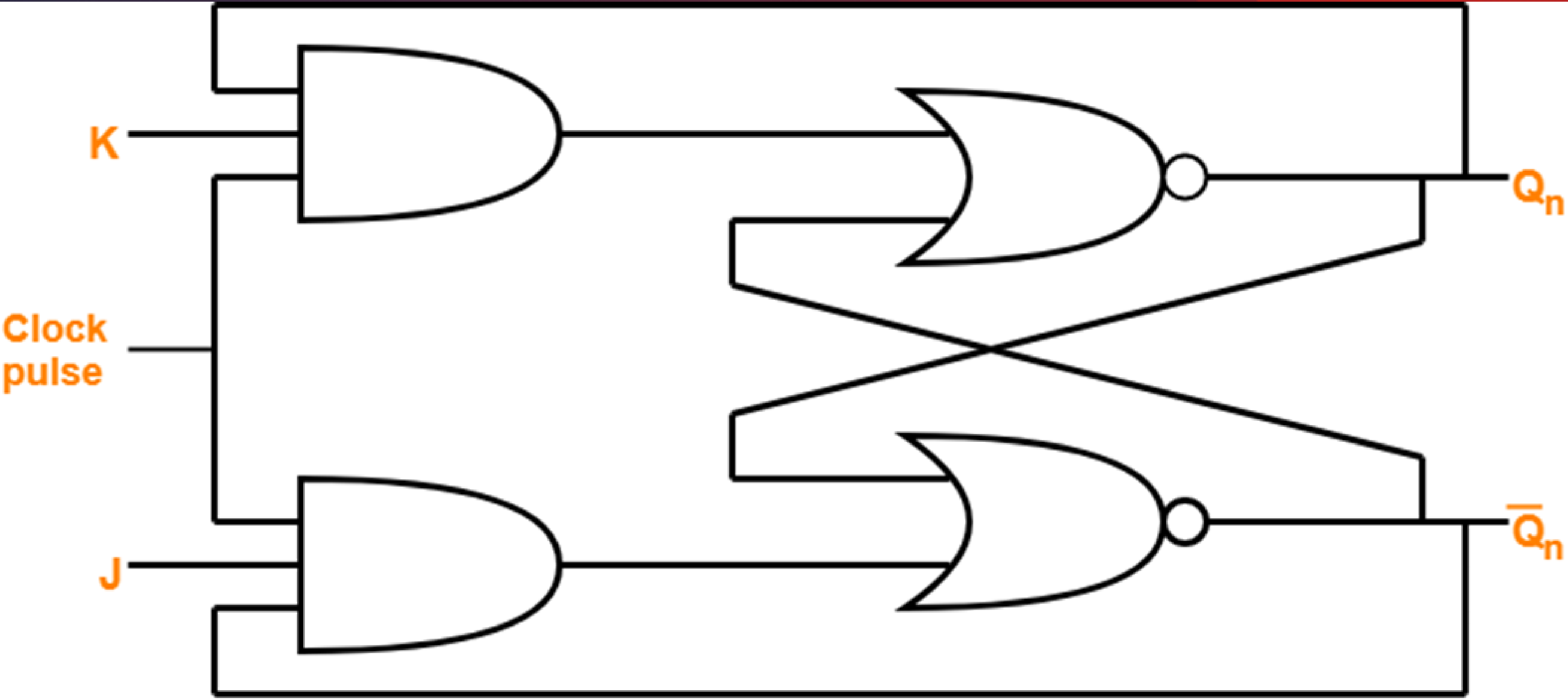
$\overline{Q} \xrightarrow{\text{feedback}}$ with S

J K Flip flop

- Invented by Jack Kilby
- A JK is refinement of RS Flipflop in that the indeterminate state of the RS type is defined in the JK type.
- The inputs J and K behave exactly like inputs S and R to Set and reset Flip flop respectively. When J and K are 1, the Flip flop output is complemented with clock transaction.

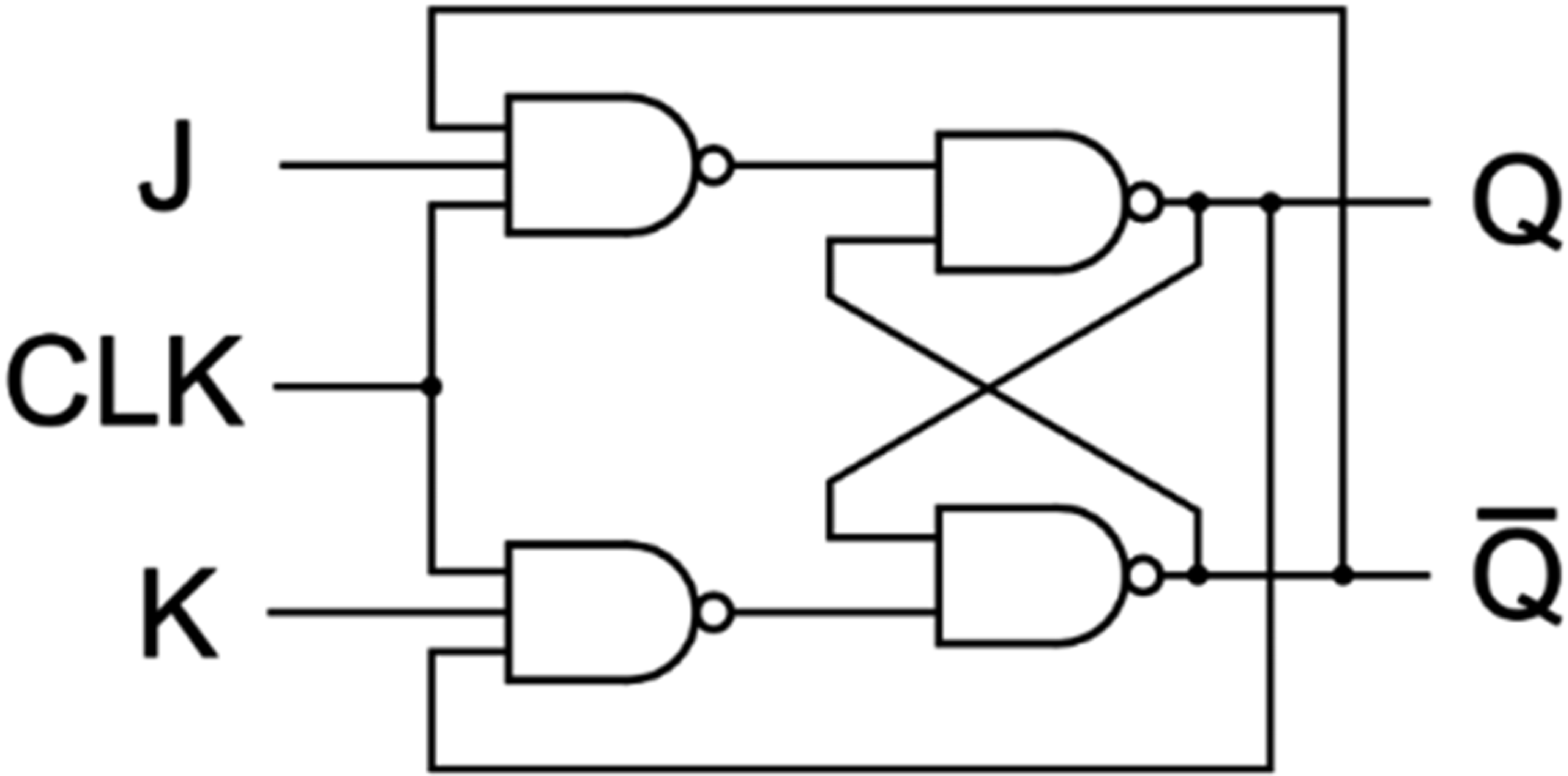


Clk	J	K	Q(t+1)	
0	X	X	Q(t)	No change
1	0	0	Q(t)	No change
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	Q'(t)	Compliment



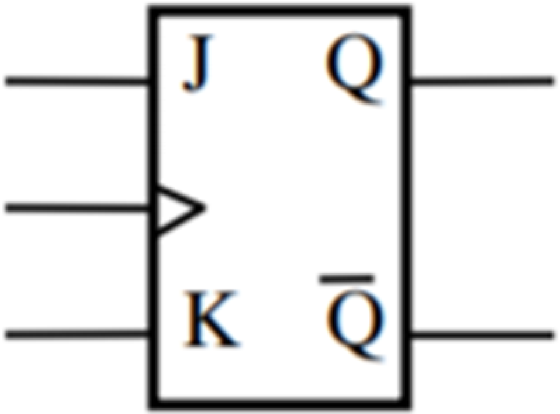
Logic Circuit For JK Flip Flop Using SR Flip Flop

(Constructed From NOR Latch)


J
CLK
K
Q
$\overline{Q}$

J	K	Q(t+1)	
0	0	Q(t)	Hold
0	1	0	Reset
1	0	1	Set
1	1	$\overline{Q}(t)$	Toggle

(b) Truth table



(c) Graphical symbol

JK Flip-Flop
(how it works)

A more versatile flip-flop

If J=0 and K=0 it stays in the same state

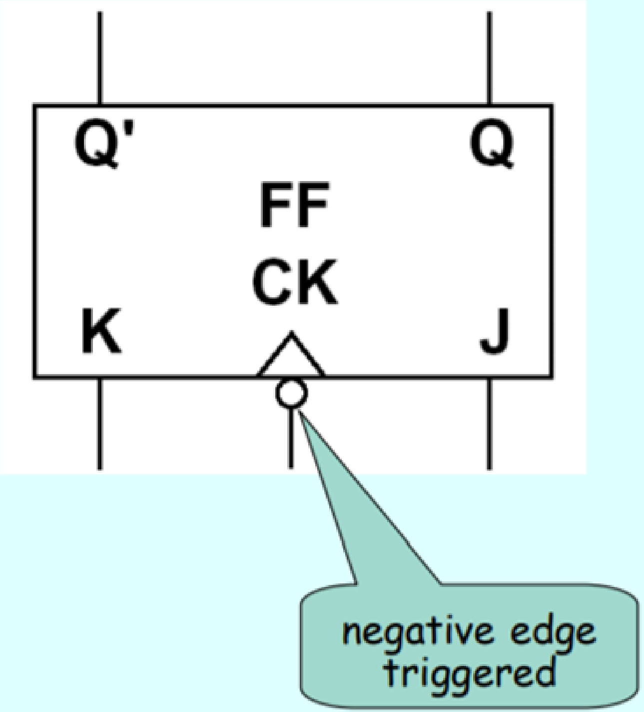
If J=1 and K=0 it sets the output Q to 1

If J=0 and K=1 it resets the output Q to 0

If J=1 and K=1 it toggles the output Q

If J=K then it behaves like a T flip-flop

Clocked JK Flip-Flop



J	K	Q	Q^+	
0	0	0	0	do nothing
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	

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

next state

- JK = 00 => no state change occurs
- JK = 10 => the flip-flop is set to 1, independent of the current state
- JK = 01 => the flip-flop is always reset to 0
- JK = 11 => the flip-flop changes the state $Q^+ = Q'$ (toggle)

Excitation Table

- Operational Characteristics of FF is given by Characteristic table. The characteristic table specifies the next state of flip flop when the inputs and present state are known.

  inputs + Present state → next state

- Excitation Table shows "For a particular transition to take place what should be the inputs" e.g., if present state is 0 and to get next state as 1 what should be the inputs
- Excitation Table for RS Flipflop

Present State	Next State	SR flip-flop inputs	
Q(t)	Q(t+1)	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

Excitation Table

Present State	Next State	JK flip-flop inputs	
Q(t)	Q(t+1)	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

Present State	Next State	D flip-flop input
Q(t)	Q(t+1)	D
0	0	0
0	1	1
1	0	0
1	1	1

Present State	Next State	T flip-flop input
Q(t)	Q(t+1)	T
0	0	0
0	1	1
1	0	1
1	1	0

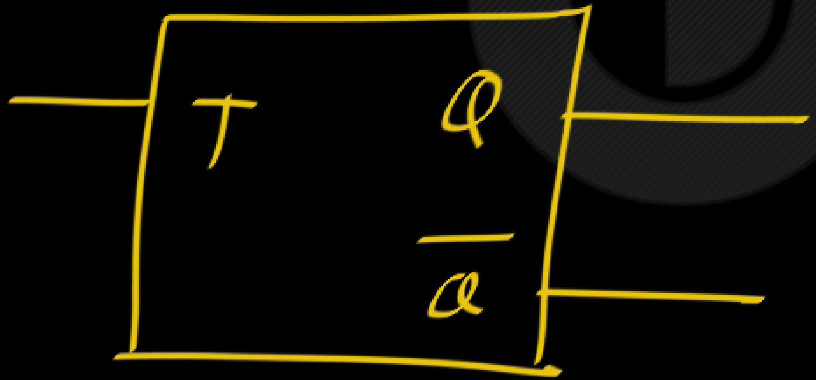
Next Topic:

4. T Flipflop

Toggle FlipFlop

If $T = 1$ $\qquad Q_n = \overline{Q}$

$T = 0$ $\qquad Q_n = Q$



$\boxed{\text{T=1 makes State Toggle / Complement.}}$

Clock	T	Q_{t+1}
Not Triggering	X	Q_t
	0	Q_t Retain
	1	$\overline{Q_t}$ ✓ Toggle

Truth Table:

	T	Q	Q_n
T=0	0	0	0
	0	1	1
T=1	1	0	1
	1	1	0

Retain / No change

State equation:

$$Q_n = T \oplus Q = \boxed{T\bar{Q} + \bar{T}Q}$$

Excitation Table:

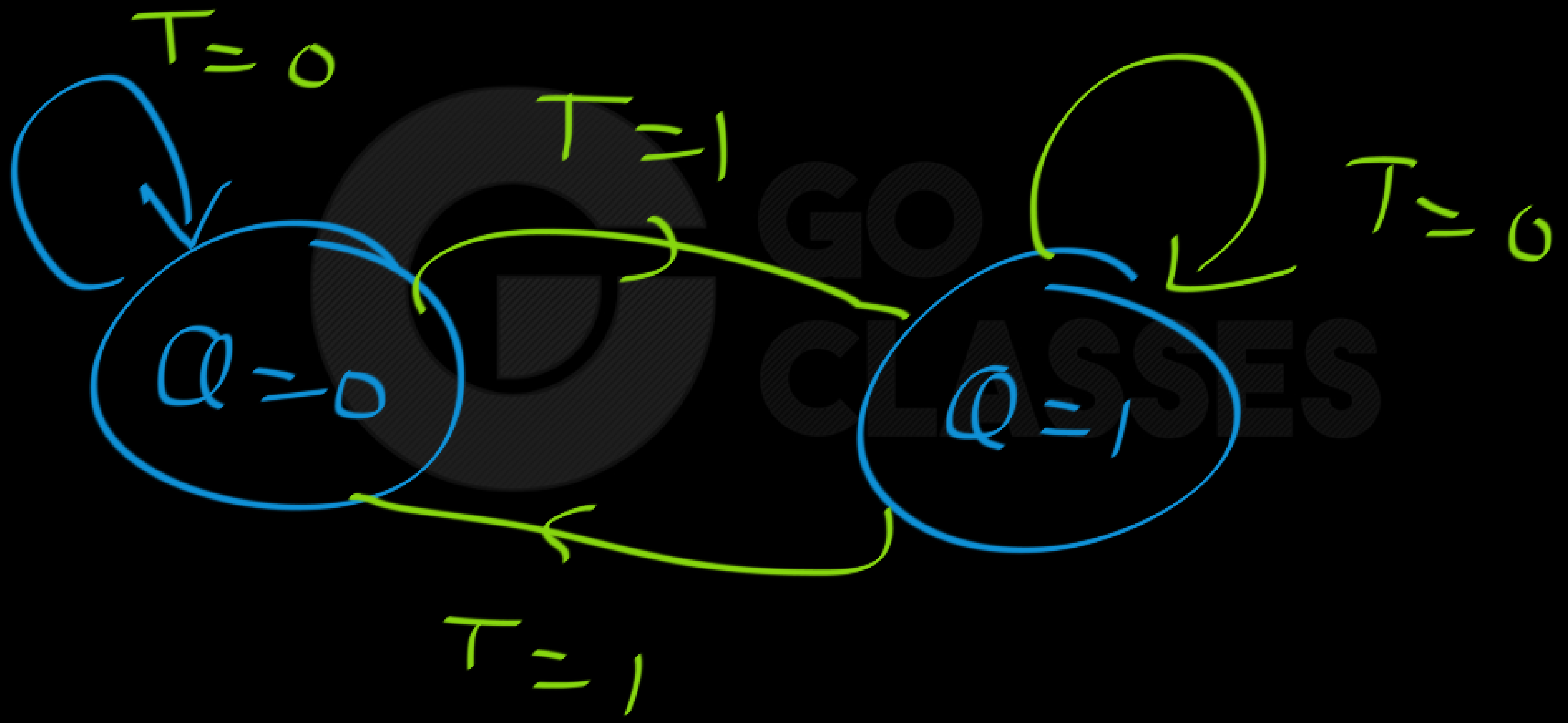
Q	$\rightarrow$	Q_n	T
0	$\rightarrow$	0	0
0	$\rightarrow$	1	1
1	$\rightarrow$	0	1
1	$\rightarrow$	1	0

Toggle

$T = f(Q, Q_n)$

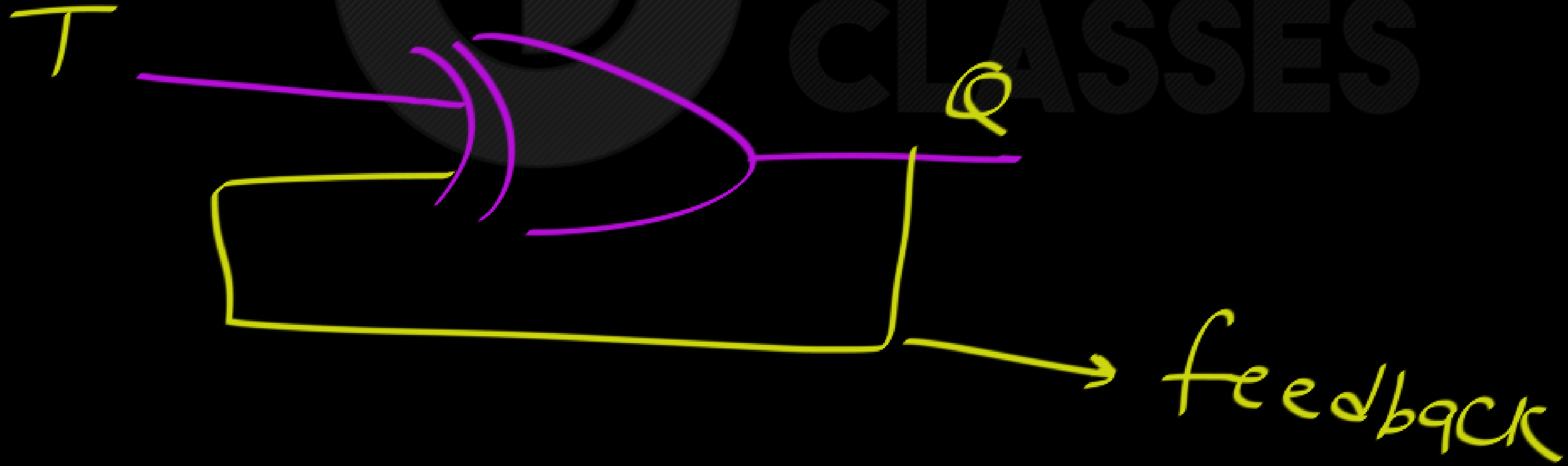
$T = Q \oplus Q_n$

State Diagram:

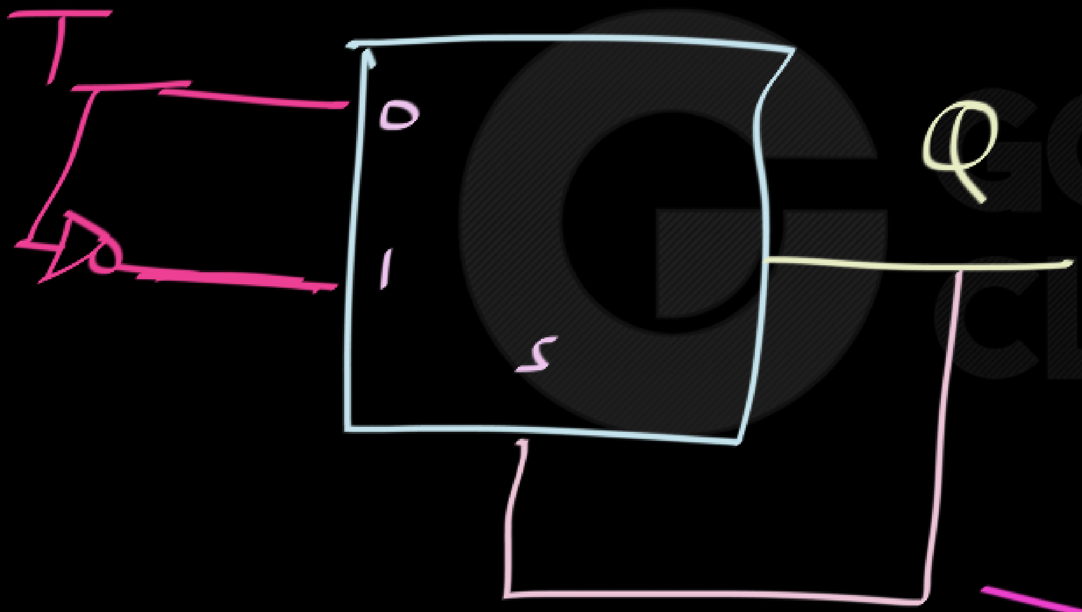


Implementation 1 :

$$Q_n = Q \oplus T = \overline{Q}T + Q\overline{T}$$

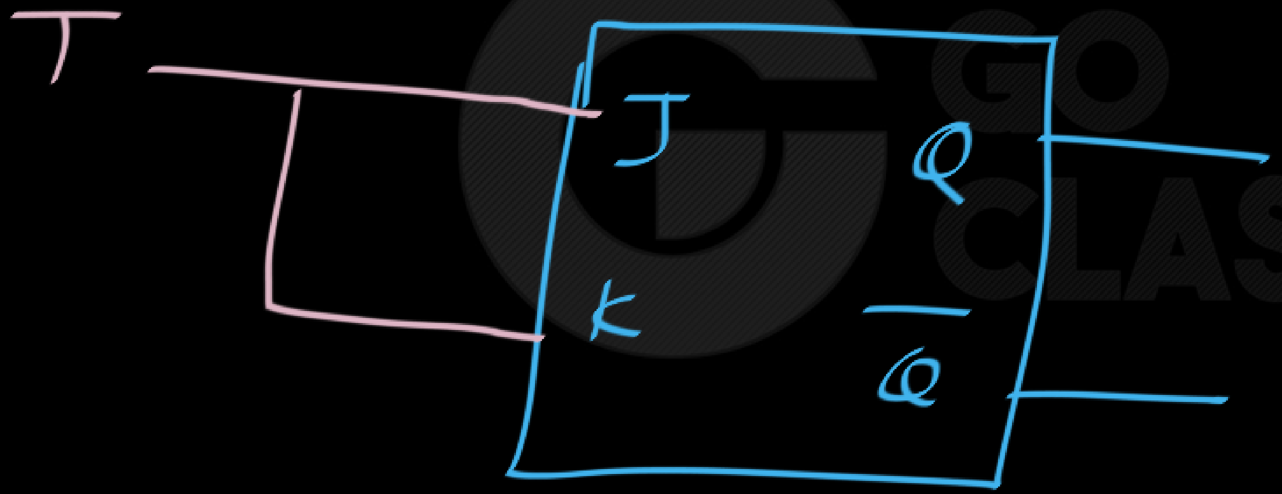


Implementation 2 (using MUX)



→ feedback

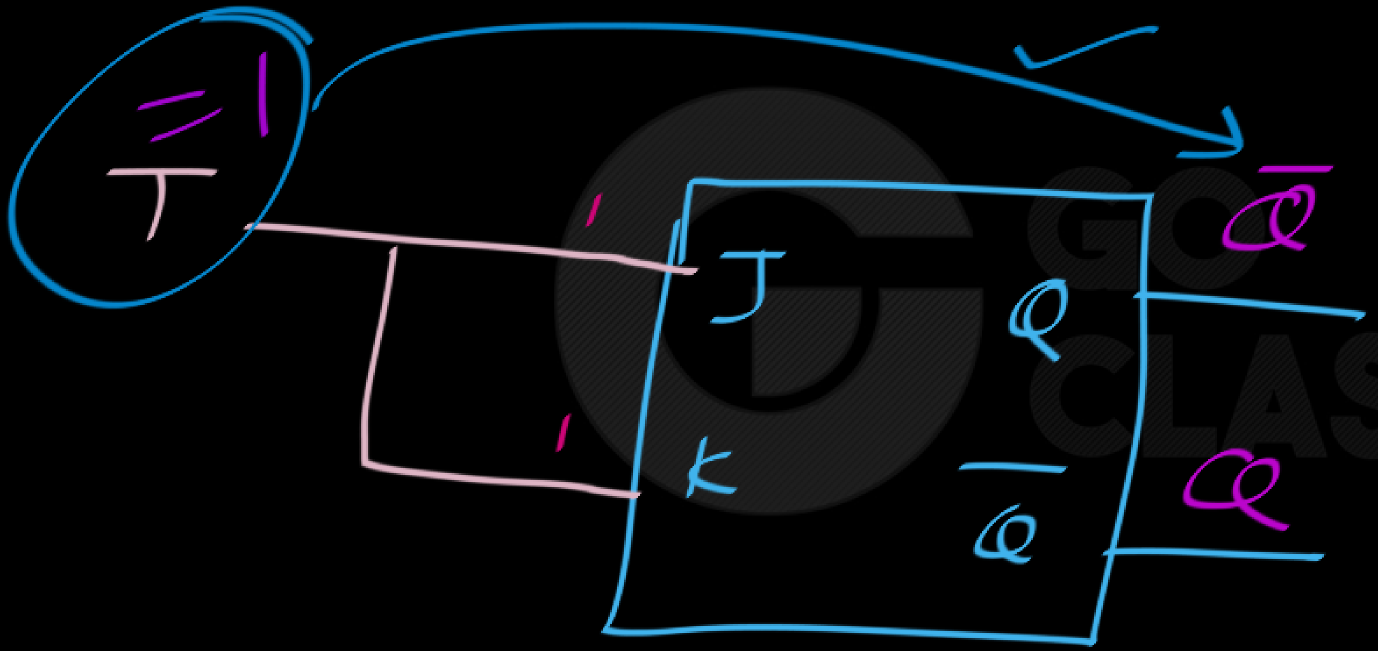
Implement using JK :



J	K	Q_n
0	0	Q
0	1	0
1	0	1
1	1	$\overline{Q}$

Relevant to T

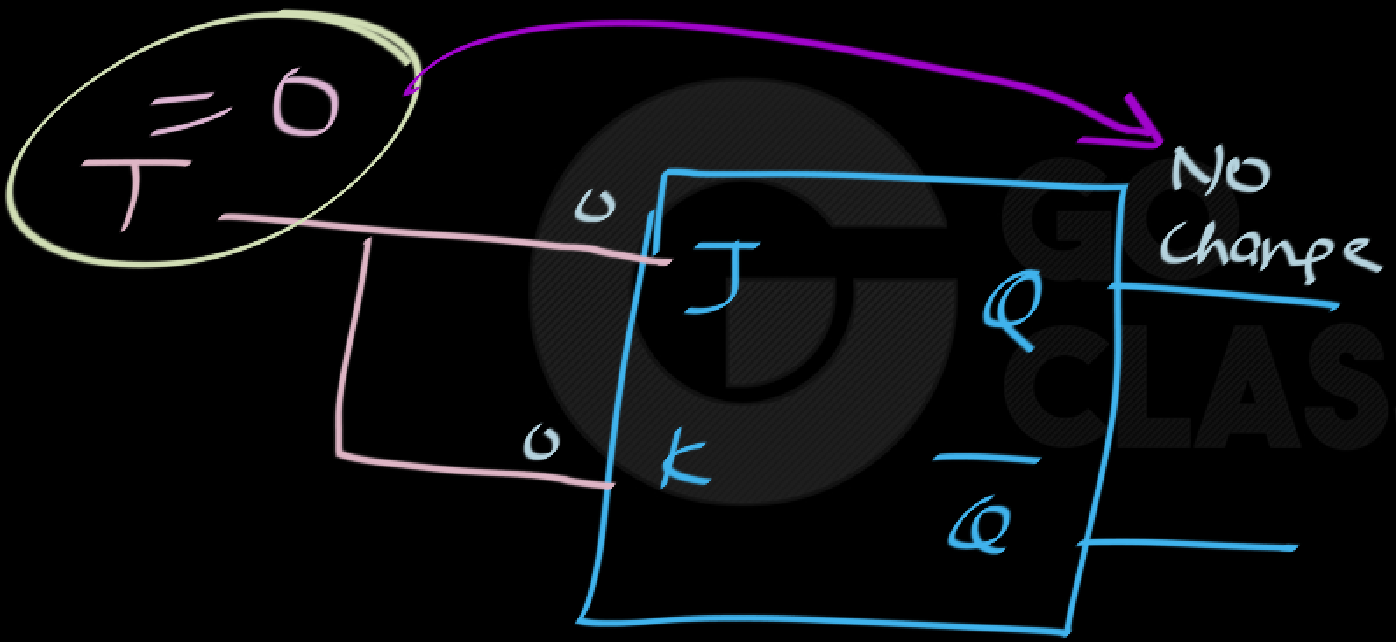
Implement using JK :



J	K	Q_n
0	0	Q
0	1	0
1	0	1
1	1	$\overline{Q}$

Relevant to T

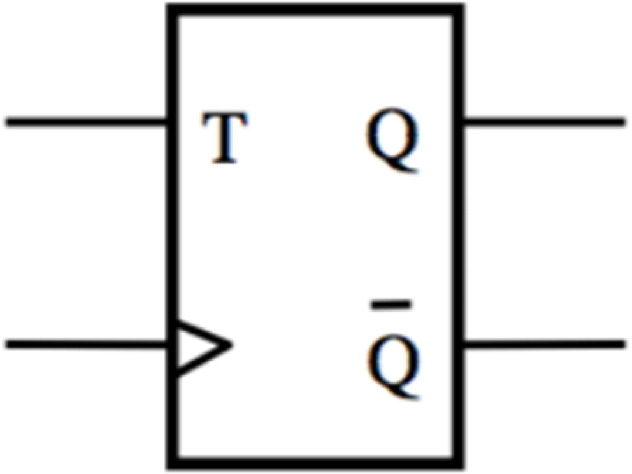
Implement using JK :



J	K	Q_n
0	0	Q
0	1	0
1	0	1
1	1	$\overline{Q}$

Relevant to T

T Flip-Flop
(circuit and truth table)



T Flip-Flop

T	$Q(t+1)$	
0	$Q(t)$	No change
1	$Q'(t)$	Complement

Motivation

A slight modification of the D flip-flop that can be used for some nice applications (e.g., counters).

In this case, T stands for Toggle.

T Flip-Flop
(how it works)

If T=0 then it stays in its current state

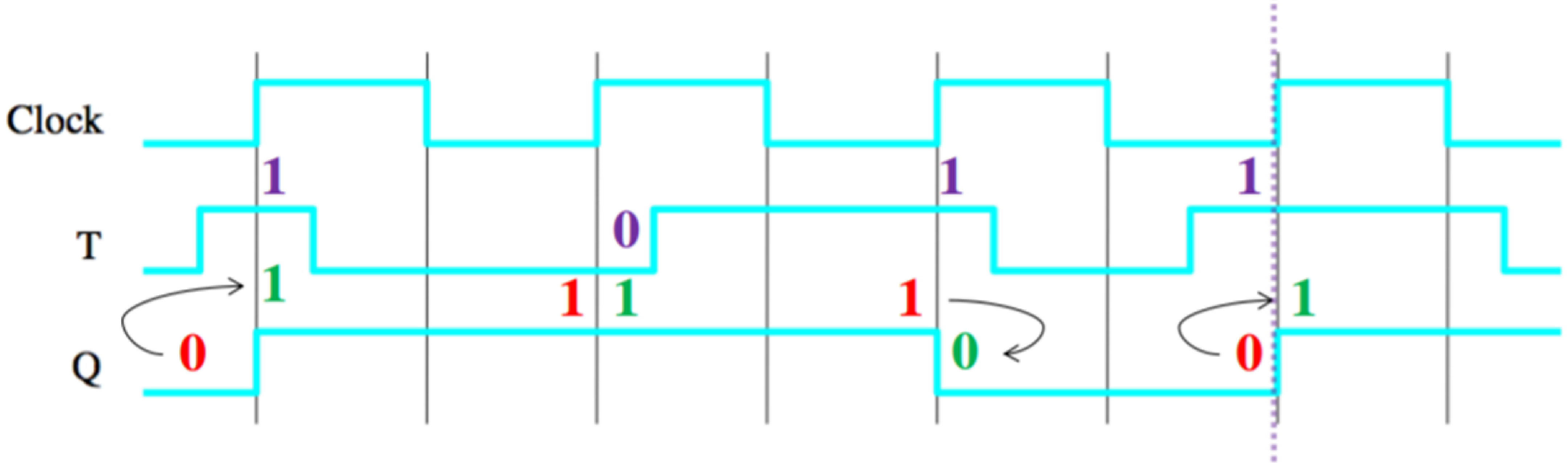
If T=1 then it reverses its current state

In other words the circuit “toggles” its state when T=1. This is why it is called T flip-flop.

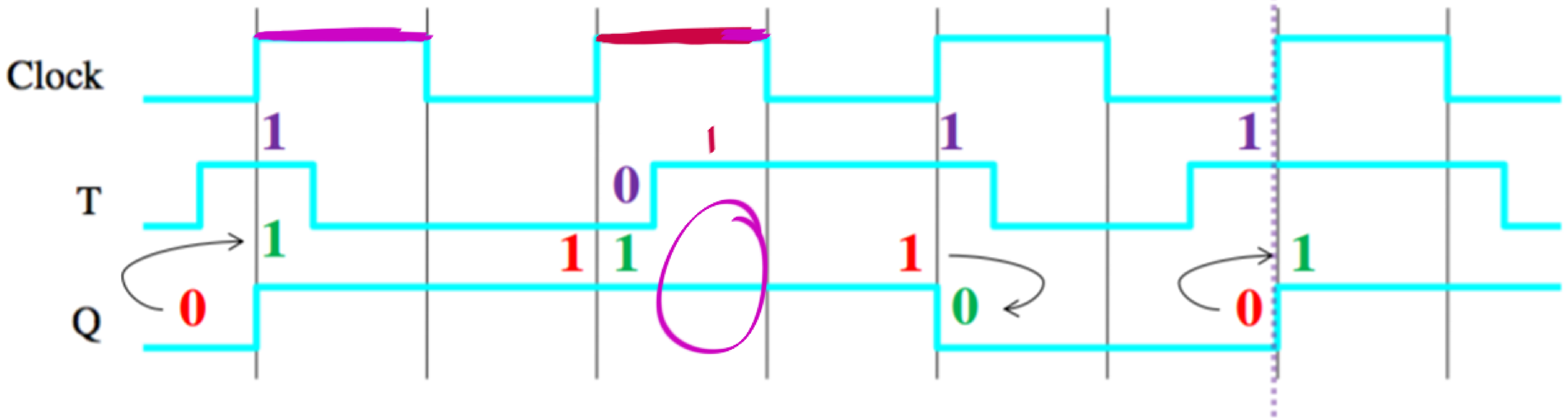
Q :

For the following T flipflop, find out whether this T ff is Edge Or Level Triggered?

T Flip-Flop (Timing Diagram)

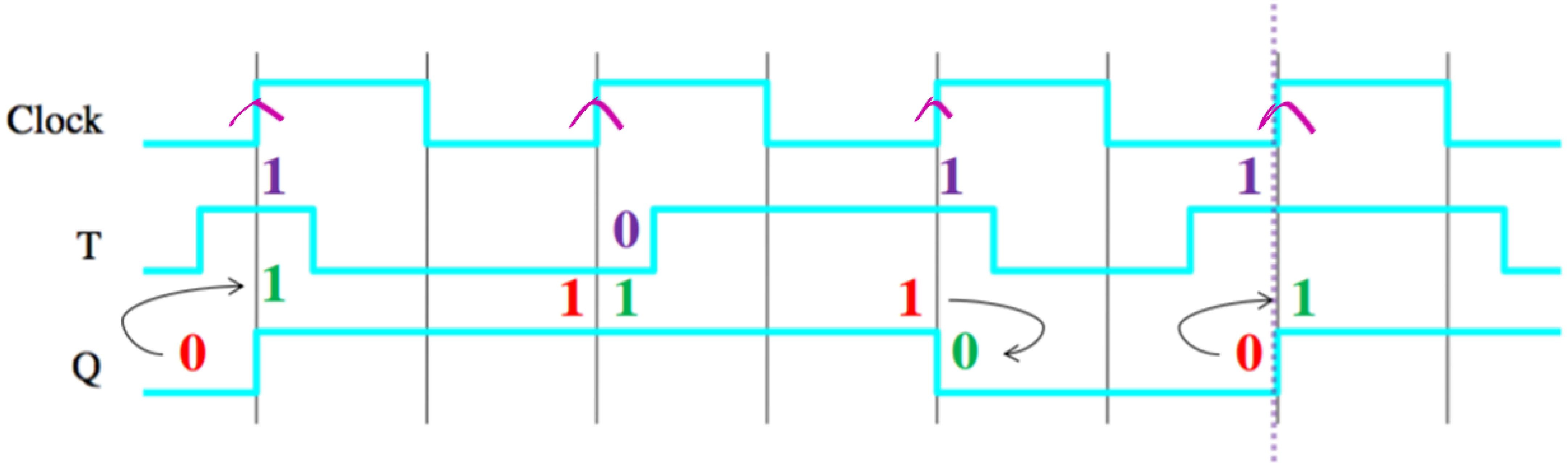


T Flip-Flop (Timing Diagram)

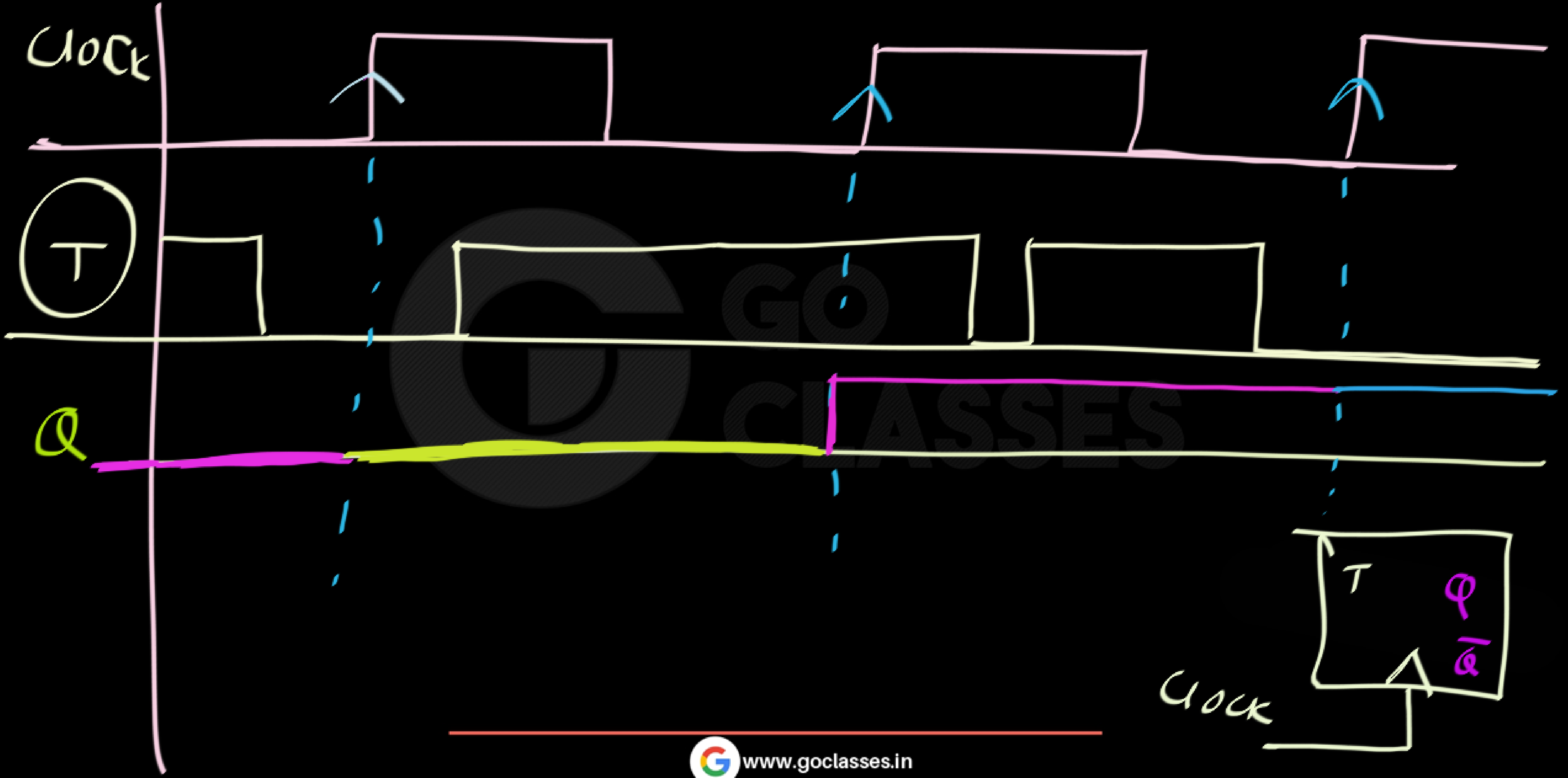


+ve level Triggered ? No

T Flip-Flop (Timing Diagram)



+ve Edge Triggered ?


Clock
T
Q
T
Q
$\bar{Q}$
Clock

7.9.7 State Diagram

A *state diagram* is a graph that shows the flip-flop's operations in terms of how it transitions from one state to another. The nodes are labeled with the states and the directed arcs are labeled with the input signals that cause the transition to go from one state to the next. Figure 21 shows the state diagram for the SR flip-flop. For example, to go from state $Q = 0$ to the state $Q = 1$, the two inputs S and R have to be 1 and 0 respectively. Similarly, if the current state is $Q = 0$ and we want to remain in that state, then SR need to be 00 or 01.

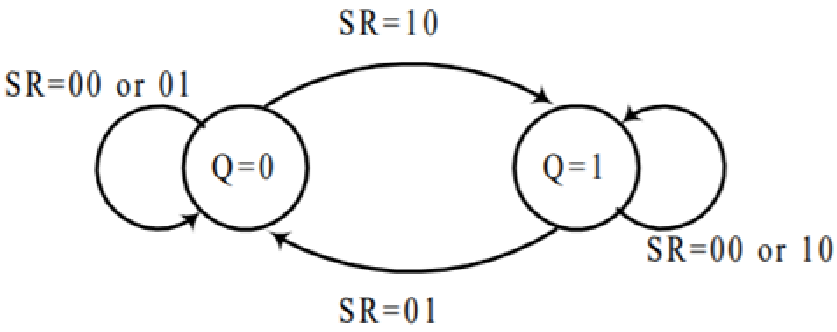


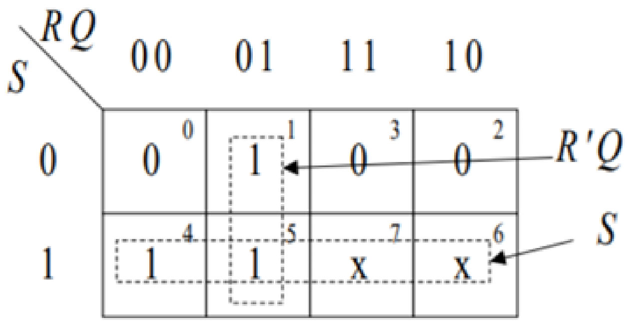
Figure 21. State diagram for the SR flip-flop.

7.9.6 Characteristic Equation

The *characteristic equation* is the functional Boolean equation that is derived from the characteristic table. This equation formally describes the functional behavior of the flip-flop. Like the characteristic table, it specifies the flip-flop's next state as a function of its current state and inputs. For example, the characteristic equation for the JK flip-flop can be derived from the truth table as follows:

$$
\begin{aligned}
Q_{next} &= J'K'Q + JK'Q + JK'Q' + JKQ' \\
&= K'Q(J'+J) + JQ'(K'+K) \\
&= K'Q + JQ'
\end{aligned}
$$

The characteristic equation can also be obtained from the truth table using the K-map method as follows for the SR flip-flop:



Thus, the characteristic equation for the SR flip-flop is

$$Q_{next} = S + R'Q.$$

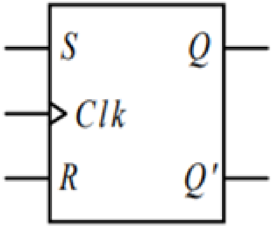
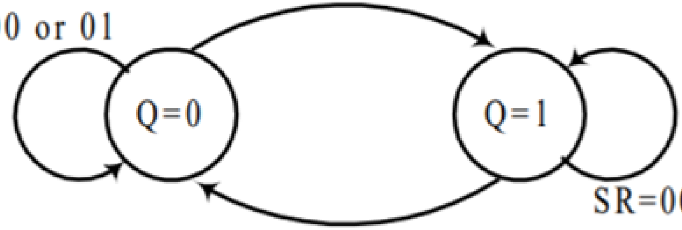
7.9.8 Excitation Table

The *excitation table* gives the value of the flip-flop's inputs that are necessary to change the flip-flop's current state to the desired next state at the next active edge of the clock signal. The excitation table answers the question of what should the inputs be when given the current state that the flip-flop is in and the next state that we want the flip-flop to go to. This table is used in the synthesis of sequential circuits.

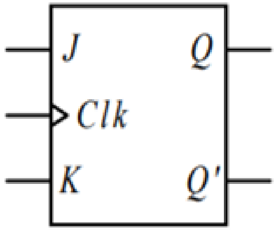
Figure 22 shows the excitation table for the SR flip-flop. As can be seen, this table can be obtained directly from the state diagram. For example, if the current state is $Q = 0$ and we want the next state to be $Q = 1$, then the two inputs must be $SR = 10$.

Q	Q_{next}	S	R
0	0	0	×
0	1	1	0
1	0	0	1
1	1	×	0

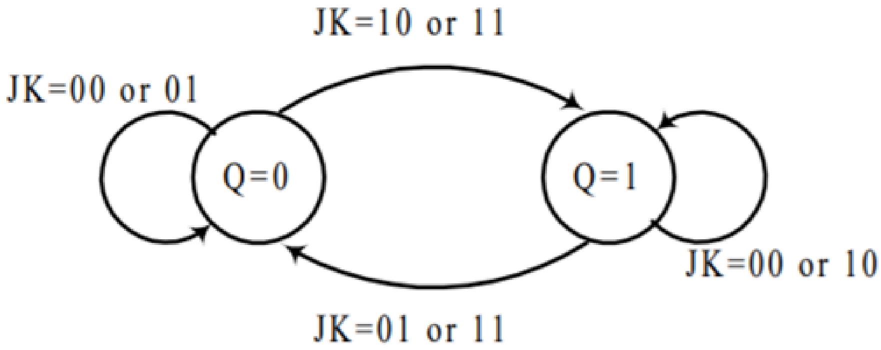
Figure 22. SR flip-flop excitation table.

Name / Symbol	Characteristic (Truth) Table	State Diagram / Characteristic Equations	Excitation Table
SR  	*S R Q Qnext* 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 ×	 $Q_{next} = S + R'Q$ $SR = 0$	*Q Qnext S R* 0 0 0 × 0 1 1 0 1 0 0 1 1 1 × 0

JK



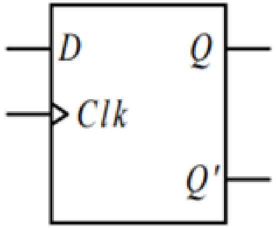
J	K	Q	Q_{next}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



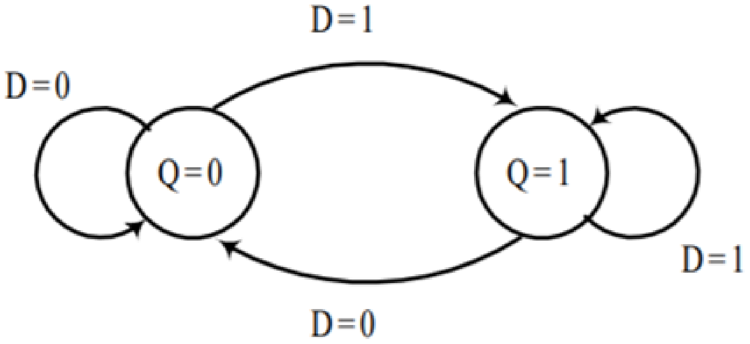
$$
\begin{aligned}
Q_{next} &= J'K'Q + JK' + JKQ' \\
&= J'K'Q + JK'Q + JK'Q' + JKQ' \\
&= K'Q(J'+J) + JQ'(K'+K) \\
&= K'Q + JQ'
\end{aligned}
$$

Q	Q_{next}	J	K
0	0	0	×
0	1	1	×
1	0	×	1
1	1	×	0

D



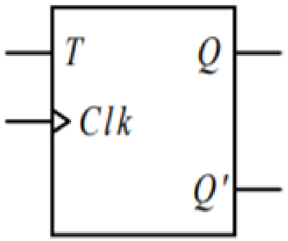
D	Q	Q_{next}
0	×	0
1	×	1



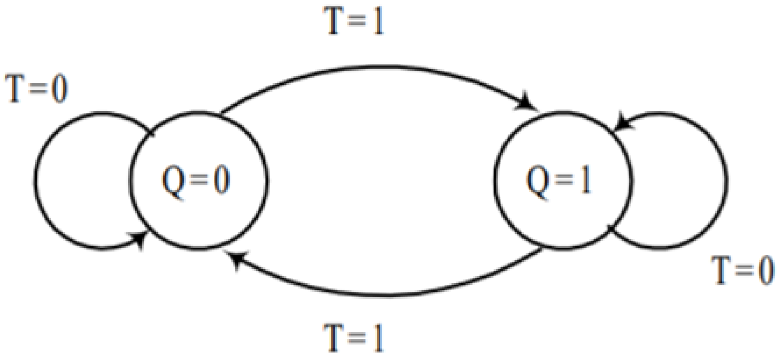
$$Q_{next} = D$$

Q	Q_{next}	D
0	0	0
0	1	1
1	0	0
1	1	1

T

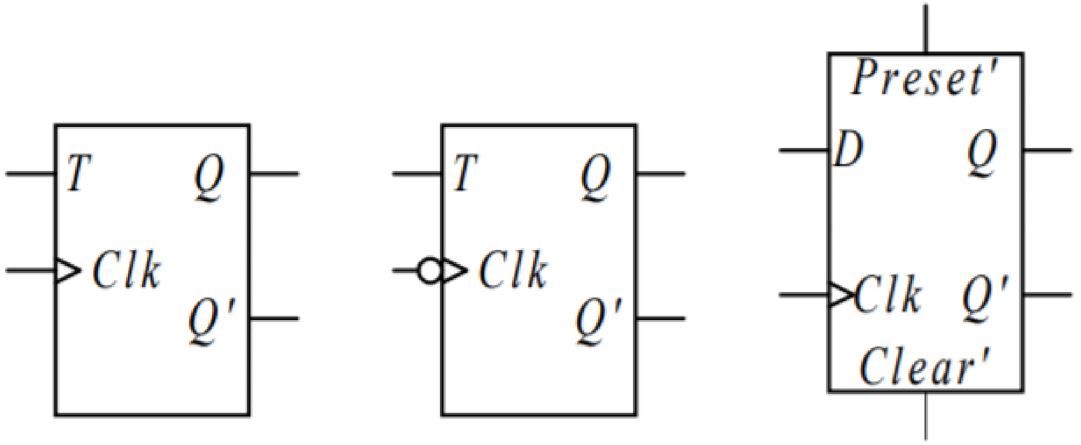


T	Q	Q_{next}
0	0	0
0	1	1
1	0	1
1	1	0



$$Q_{next} = TQ' + T'Q = T \oplus Q$$

Q	Q_{next}	T
0	0	0
0	1	1
1	0	1
1	1	0



(b) (c) (d)

(b) positive-edge-triggered active high T flip-flop;

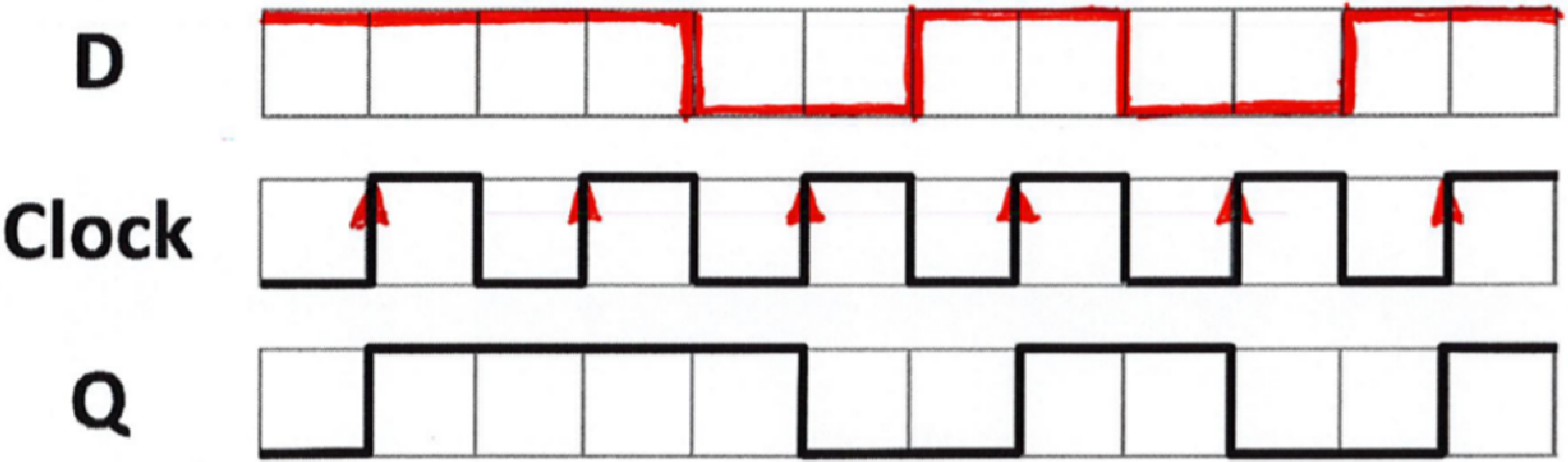
(c) negative-edge-triggered T flip-flop;

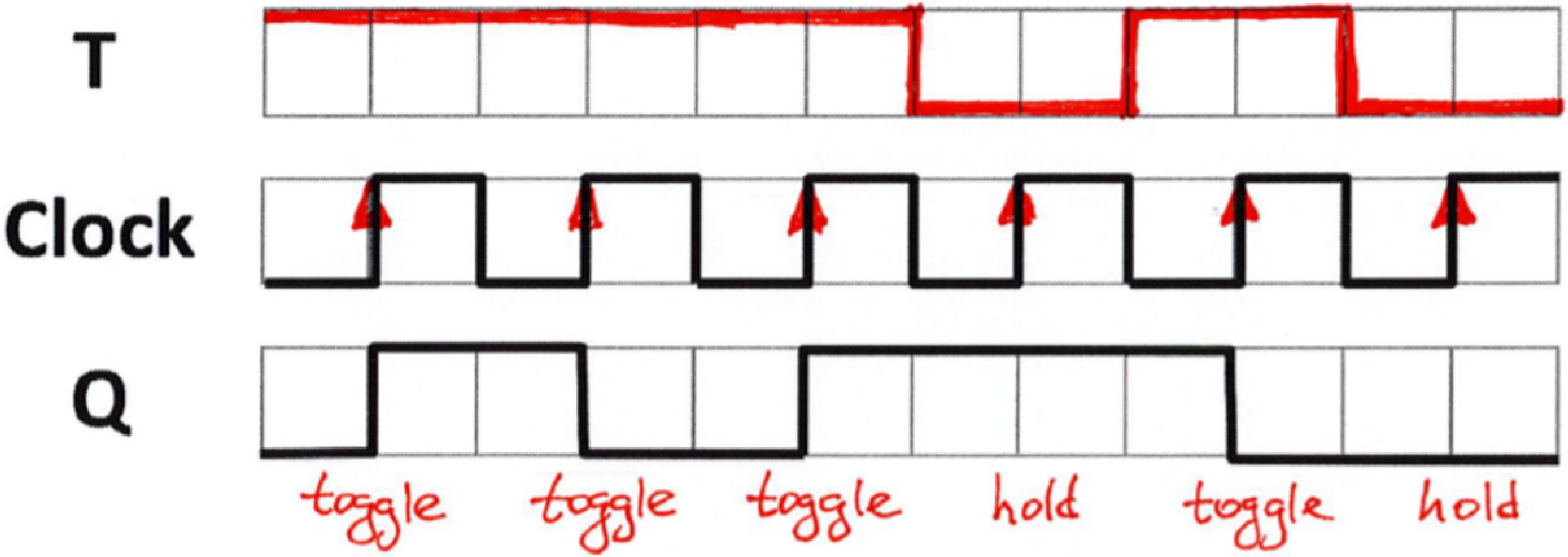
(d) positive-edge-triggered D flip-flop with asynchronous active low preset and clear.

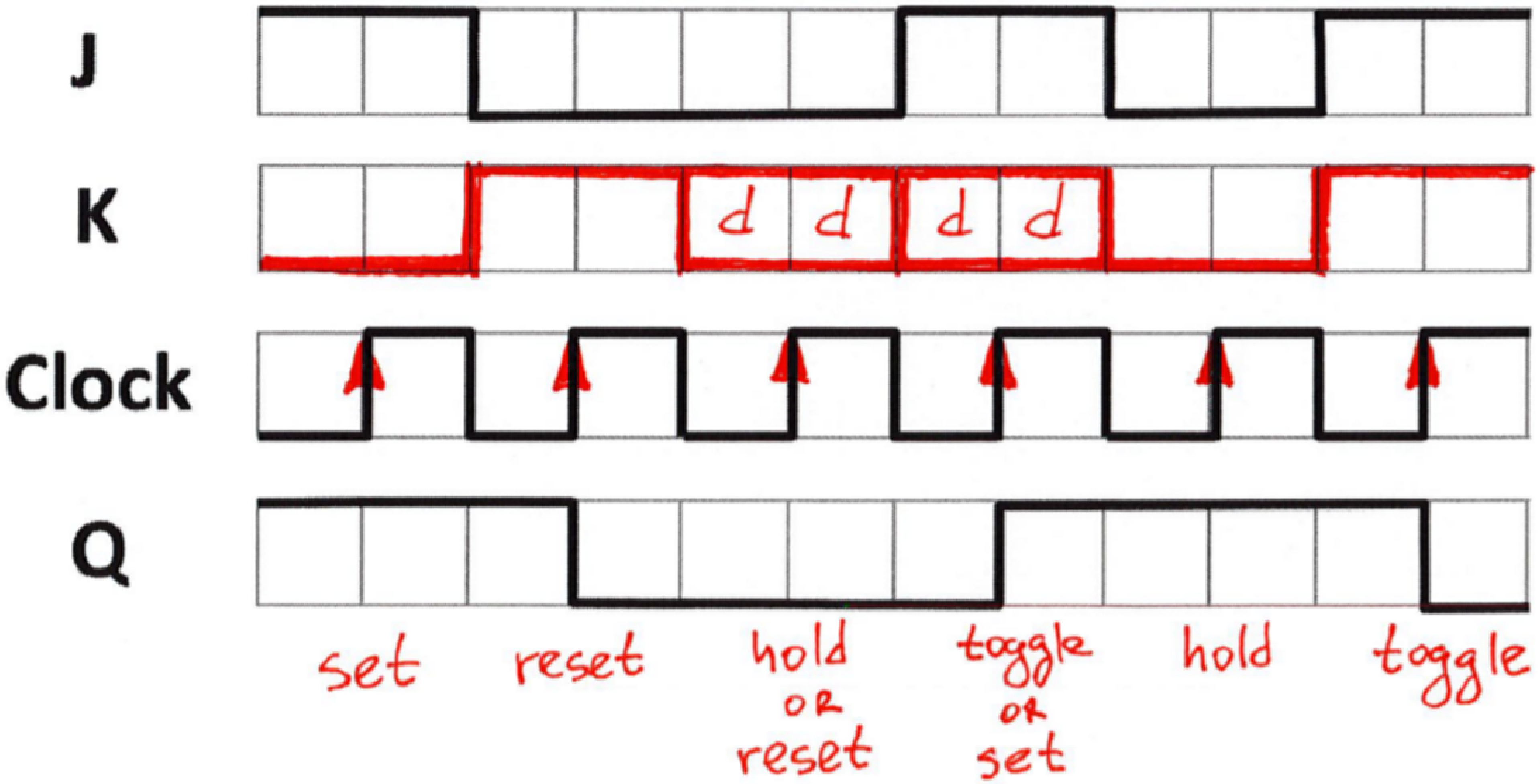
Next lecture

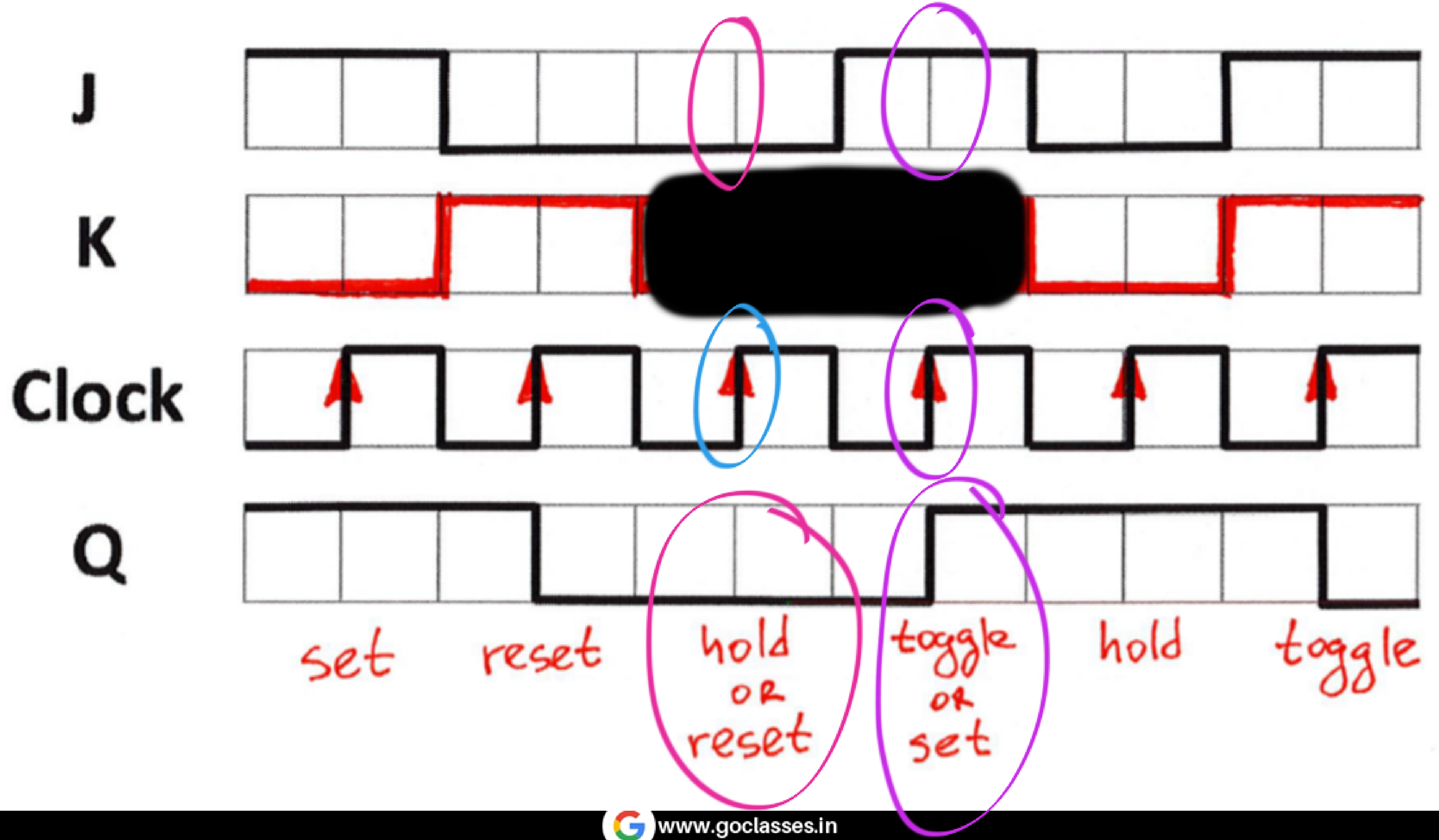
Some Timing Diagrams (for positive-edge-triggered FF) :

HW : Verify them ; Understand them


D
Clock
Q


T
Clock
Q
toggle
toggle
toggle
hold
toggle
hold


J
K
d
d
d
d
Clock
Q
set
reset
hold OR reset
toggle OR set
hold
toggle


J
K
Clock
Q
set
reset
hold
OR
reset
toggle
OR
set
hold
toggle

Q: If $J=1$; $k=$ unknown

$Q_n = ?$

Q: If $J = 1$; k = unknown

$Q_n = ?$ → set / Toggle

$J=1, k=0$; set

$J=1, k=1$ Toggle

Q: If $J=0$; $k=$ unknown

$Q_n = ?$

Q: If $J=0$; $k=$ Unknown

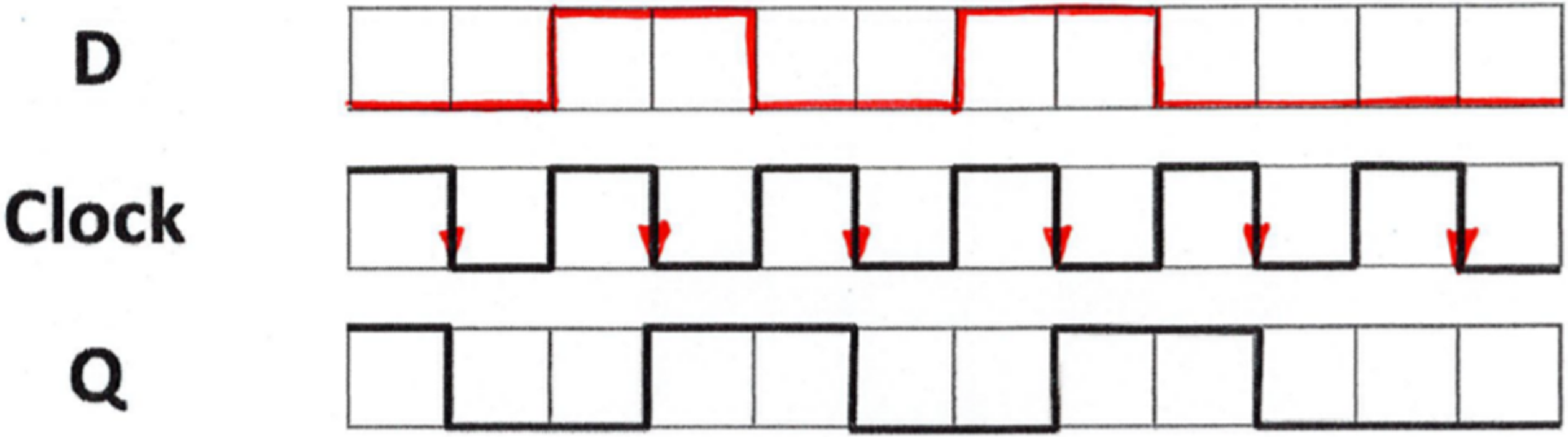
$Q_n = ?$ → Retain / Reset

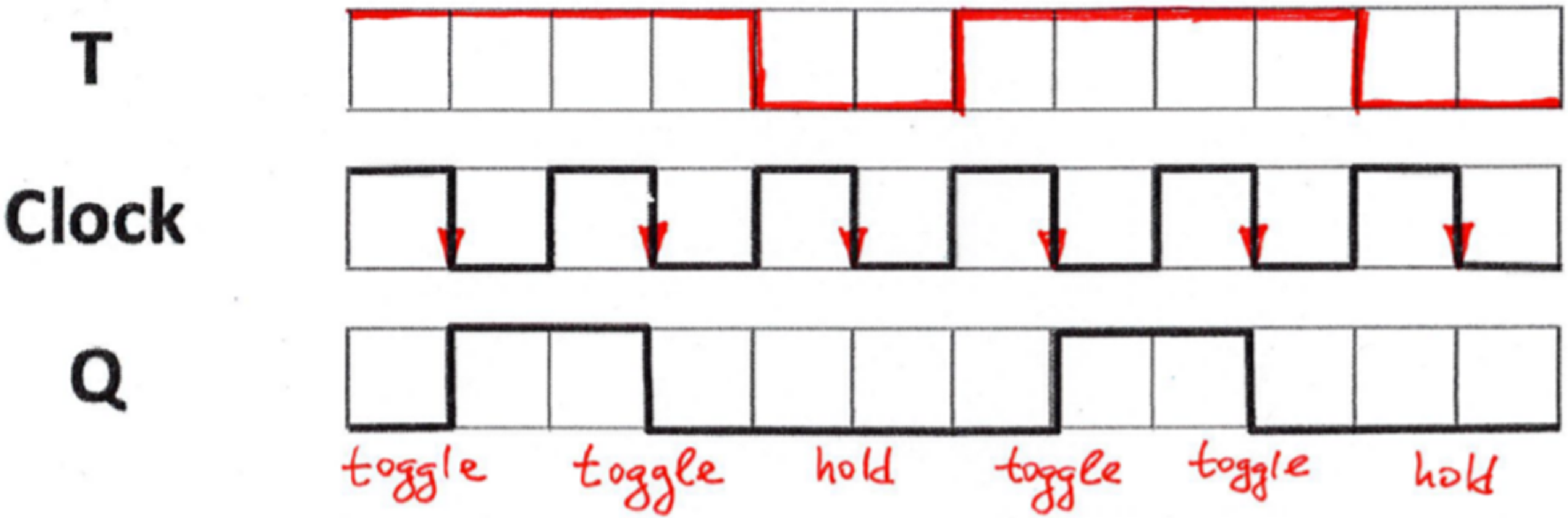
$J=0 \quad k=0$ — $Q_n = Q$

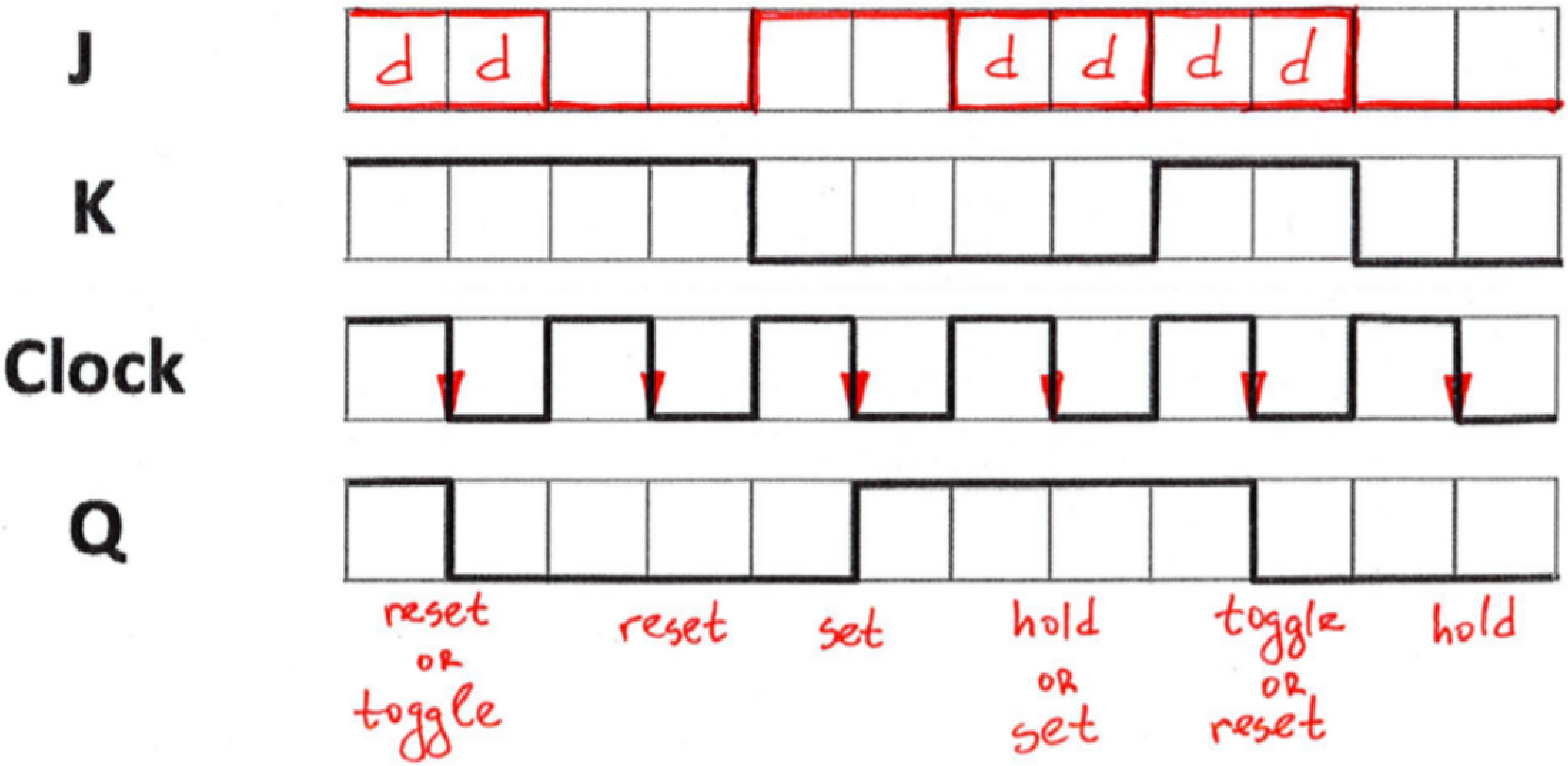
$J=0 \quad k=1$ — $Q_n = 0$

Some Timing Diagrams (for negative-edge-triggered FF) :

HW: Verify, Understand.


D
Clock
Q


T
Clock
Q
toggle
toggle
hold
toggle
toggle
hold

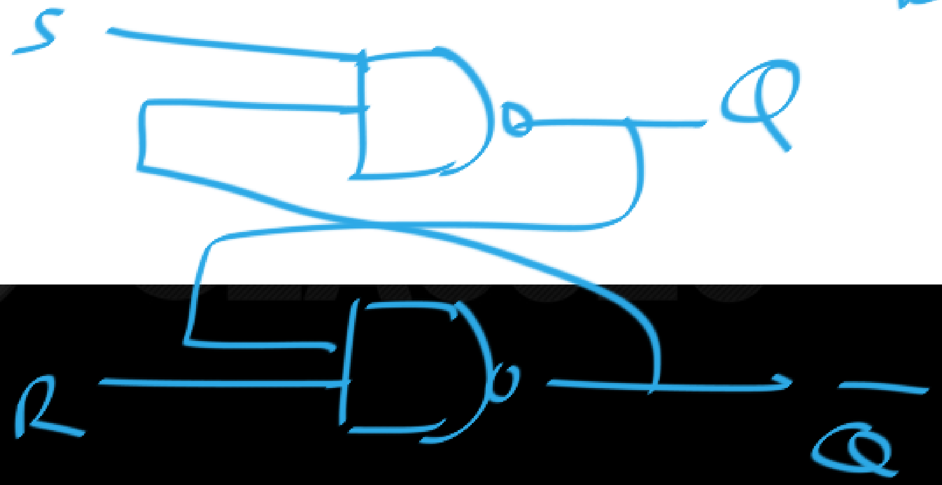

J
K
Clock
Q
d
d
d
d
d
d
reset
OR
toggle
reset
set
hold
OR
set
toggle
OR
reset
hold

Revisit

GATE IT 2007 | Question: 7

Which of the following input sequences for a cross-coupled $R-S$ flip-flop realized with two $NAND$ gates may lead to an oscillation?

A. $11, 00$
B. $01, 10$
C. $10, 01$
D. $00, 11$



Author 1

flipflop : clocked latch

We are following NPTEL (Srinivasan Sir)

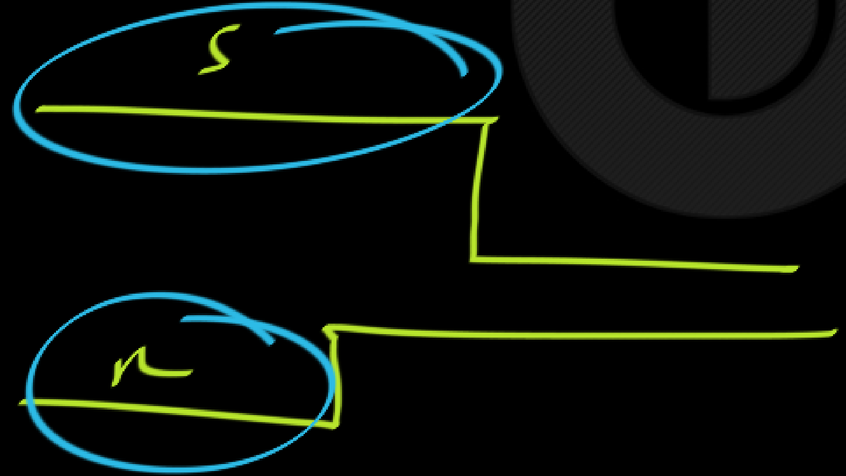
Author 2

"flipflop" : Edge Triggered flipflop

Latch : level-Triggered ff

Latch → No concept of Clock

Level - Triggerd



because as long as inputs are Applied, o/p will keep changing.

Q: Clock Triggering (+ve Edge Triggered ff, +ve level Triggered ff)

is Property of ?

1. Clock

2. Flip Flop

Q: Clock Triggering (+ve Edge Triggered ff, +ve level Triggered ff)

is Property of ?

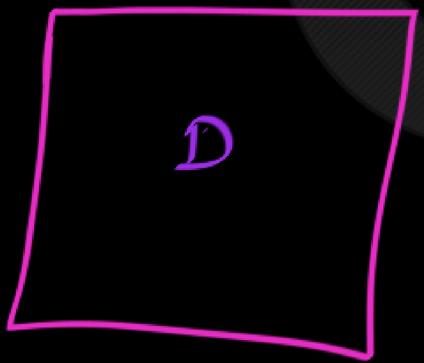
X (1) Clock → fixes for all flipflops

(2) flip flop

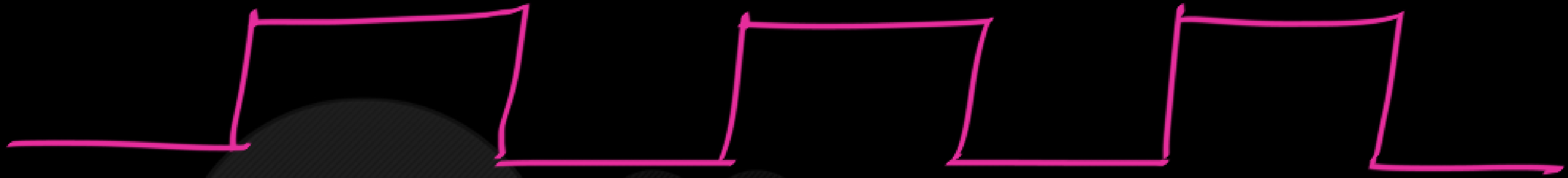
~~+ve Edge Triggered Clock~~

" " " flip flop ✓

ff is Triggered by +ve Edge of Clock.



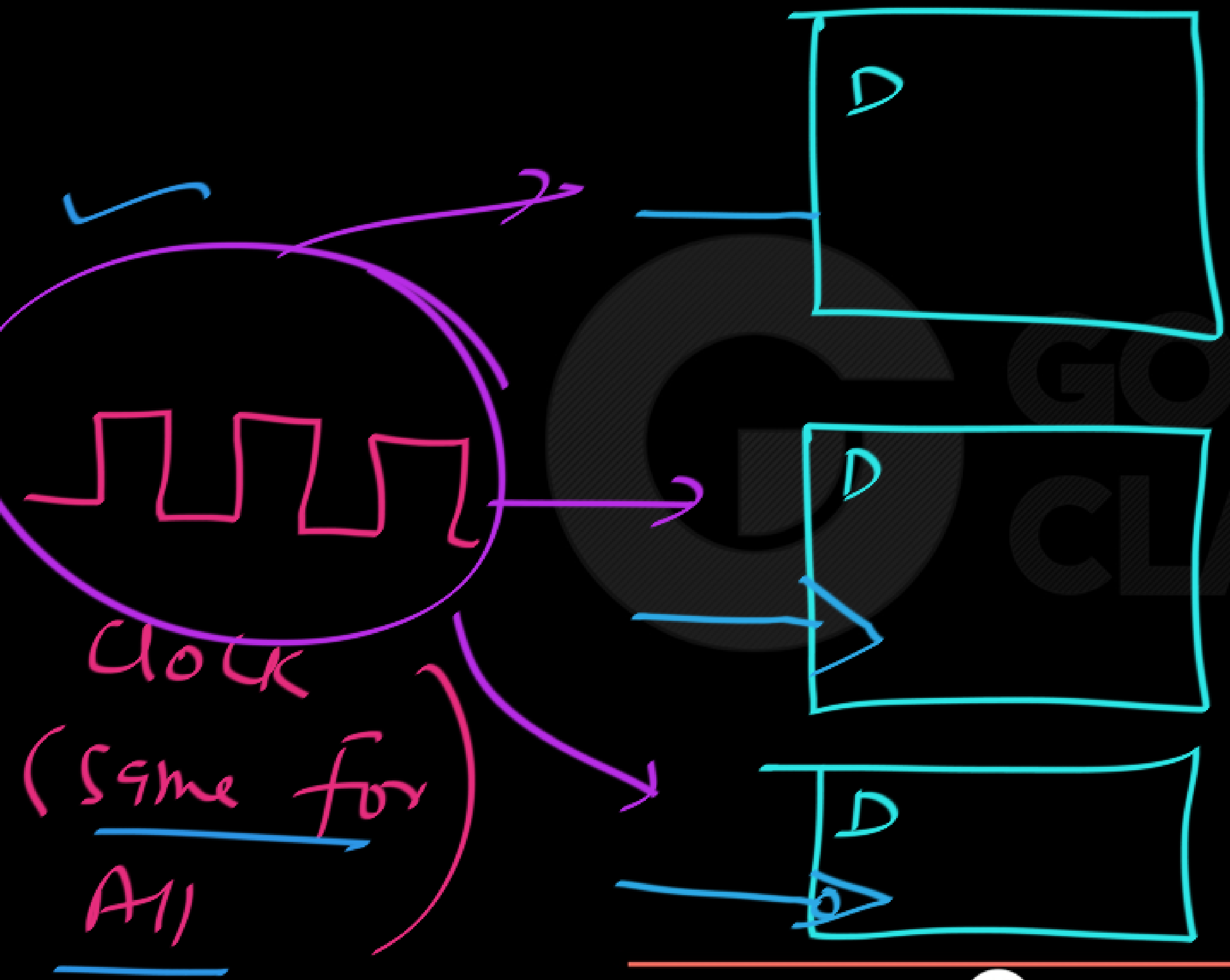
Clock :



+ve level Triggered clock

-ve Edge " "

Non-sense.


D
D
D
clock
(same for
All