



CprE 281: Digital Logic

Instructor: Alexander Stoytchev

<http://www.ece.iastate.edu/~alexs/classes/>

T Flip-Flops & JK Flip-Flops

*CprE 281: Digital Logic
Iowa State University, Ames, IA
Copyright © Alexander Stoytchev*

Administrative Stuff

- **Homework 7 is due on Monday Oct 12 @ 4pm.**
- **Homework 8 is due on Monday Oct 19 @ 4pm.**
- **The second midterm exam is in 2 weeks (Friday Oct 23).**

Administrative Stuff

- **Midterm Exam #2**
- **When: Friday October 23 @ 4:20pm.**
- **Where: WebEx**
- **What: Chapters 1, 2, 3, 4 and 5**
- **The exam will be closed book but open notes (you can bring up to 3 pages of handwritten notes).**

Midterm 2: Format

- **The exam will be out of 130 points**
- **You need 95 points to get an A for this exam**
- **It will be great if you can score more than 100 points.**
 - **but you can't roll over your extra points 😞**

Midterm 2: Topics

- **K-maps for 2, 3, and 4 variables**
- **Binary Numbers and Hexadecimal Numbers**
- **1's complement and 2's complement representation**
- **Addition and subtraction of binary numbers**
- **Circuits for adders and fast adders, delay calculation**

- **Single and Double precision IEEE floating point formats**
- **Converting a real number to the IEEE format**
- **Converting a floating point number to base 10**

- **Multiplexers (circuits and function)**
- **Synthesis of logic functions using multiplexers**
- **Shannon's Expansion Theorem**

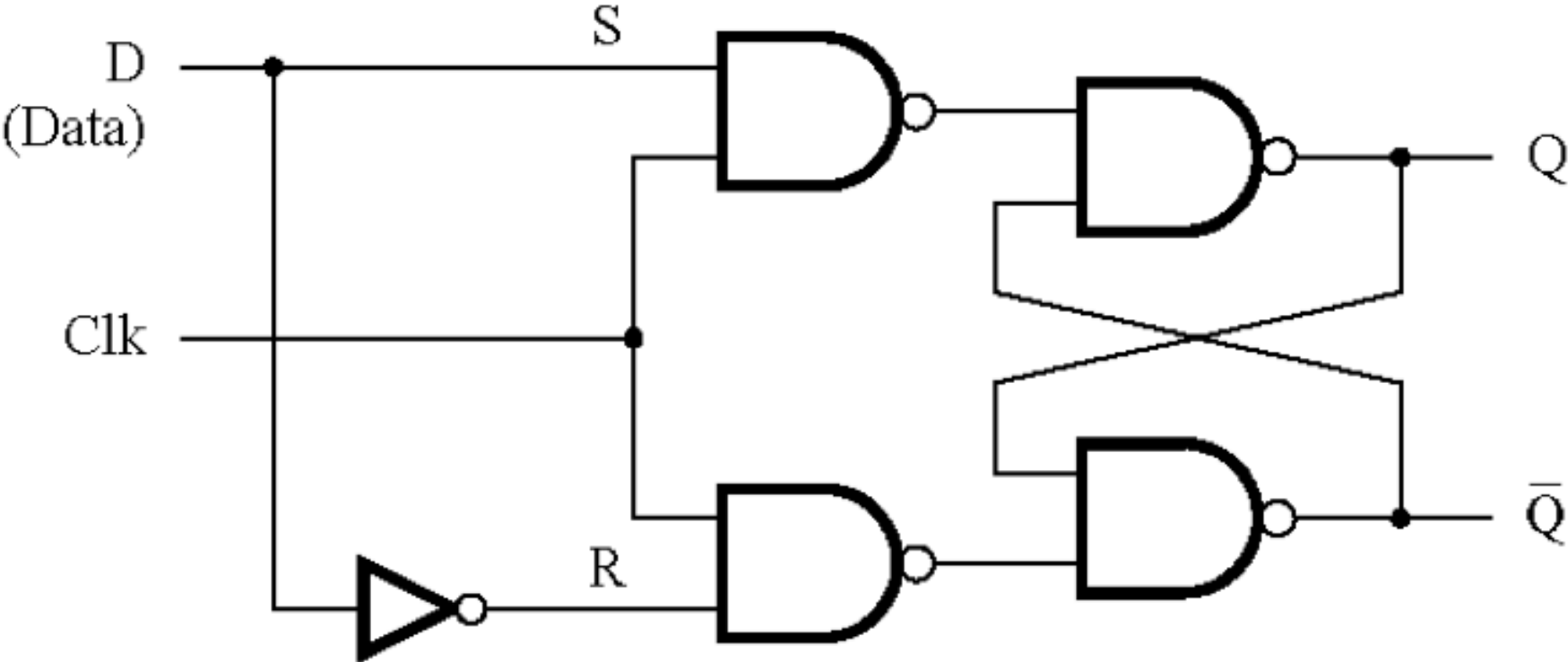
Midterm 2: Topics

- **Decoders (circuits and function)**
- **Demultiplexers**
- **Encoders (binary and priority)**
- **Code Converters and Comparison Circuits**
- **Synthesis of logic circuits using adders, multiplexers, encoders, decoders, and basic logic gates**
- **Synthesis of logic circuits given constraints on the available building blocks that you can use**
- **Latches (circuits, behavior, timing diagrams)**
- **Flip-Flops (circuits, behavior, timing diagrams)**
- **Registers and Register Files**
- **Counters**

Quick Review

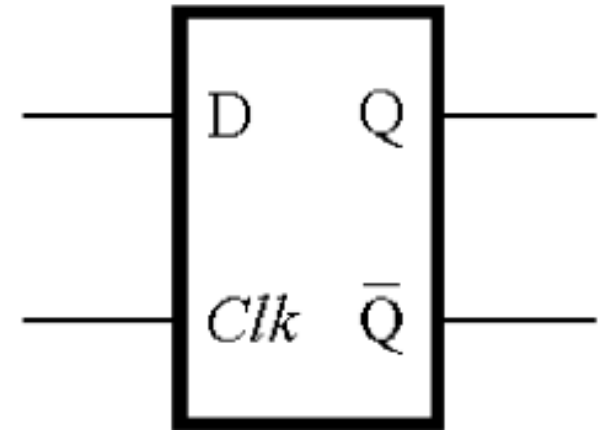
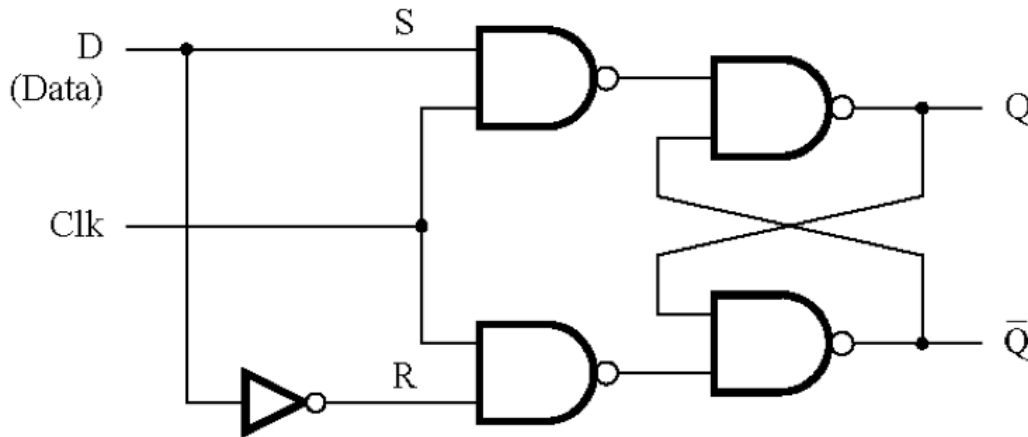
Gated D Latch

Circuit Diagram for the Gated D Latch

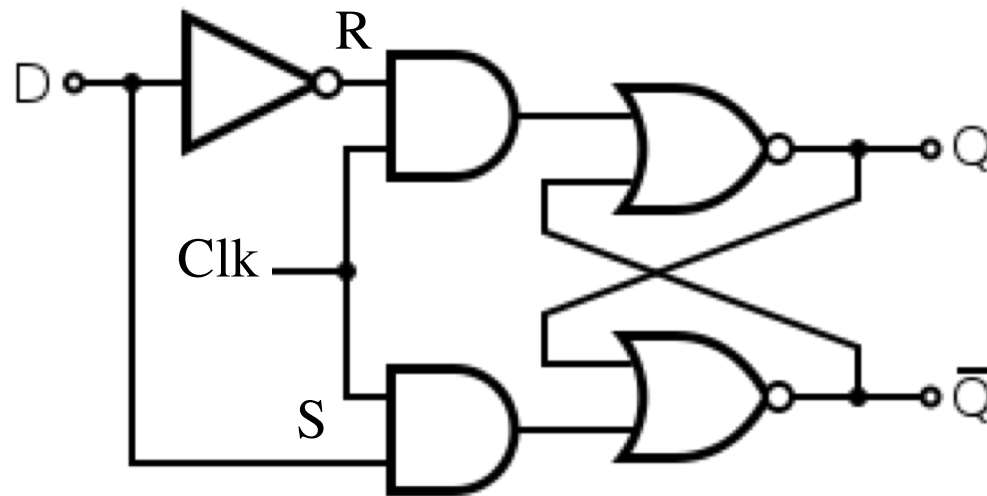


[Figure 5.7a from the textbook]

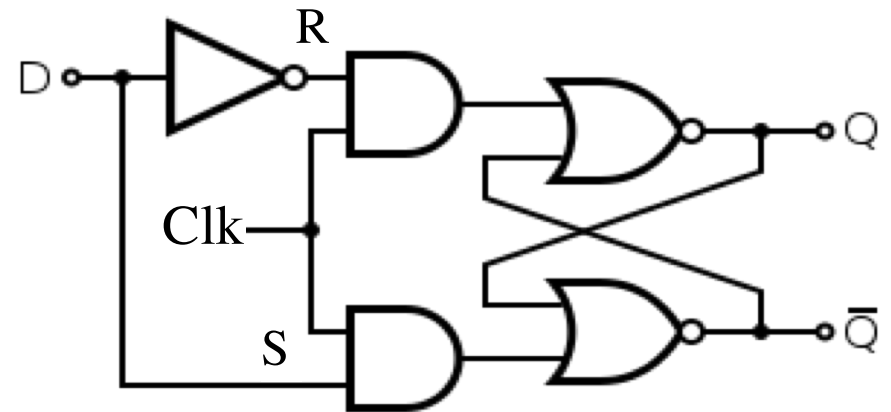
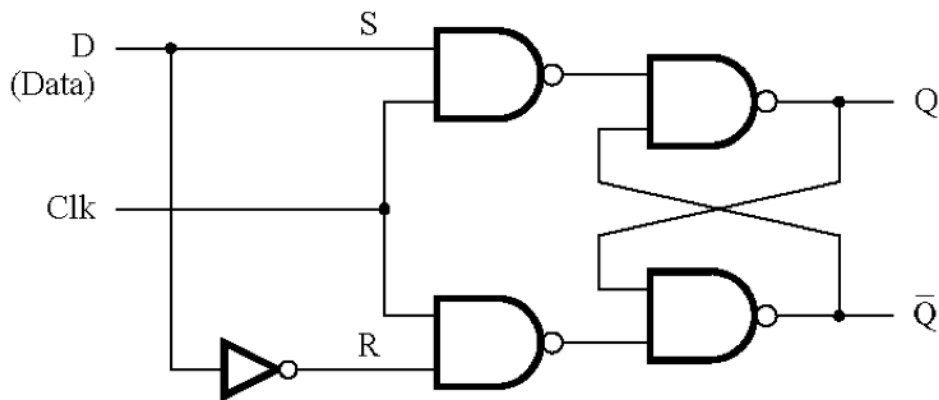
Circuit Diagram and Graphical Symbol for the Gated D Latch



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

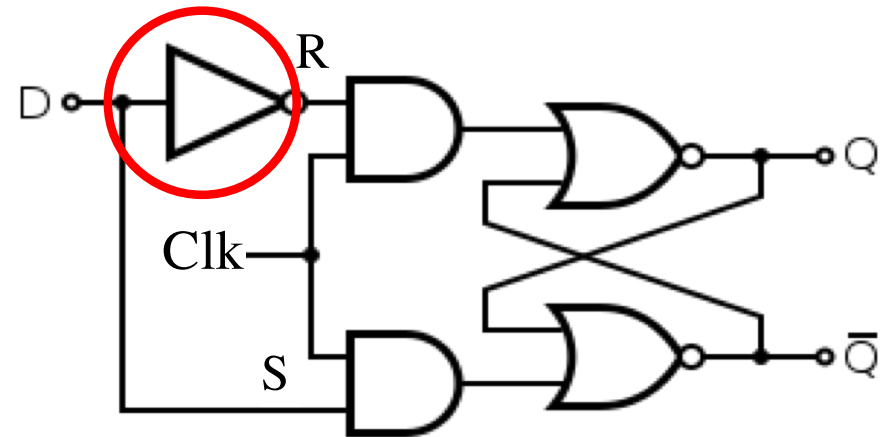
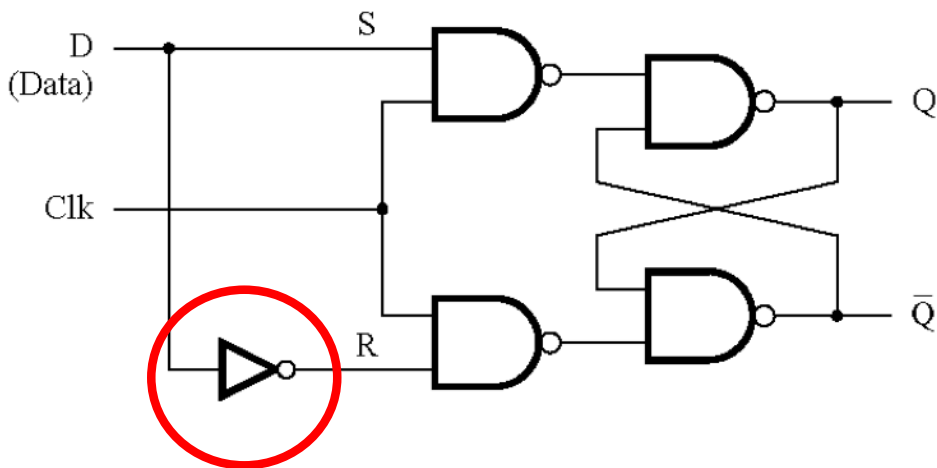


Circuit Diagram for the Gated D Latch (with the latch implemented using NORs)



[Figure 5.7a from the textbook]

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

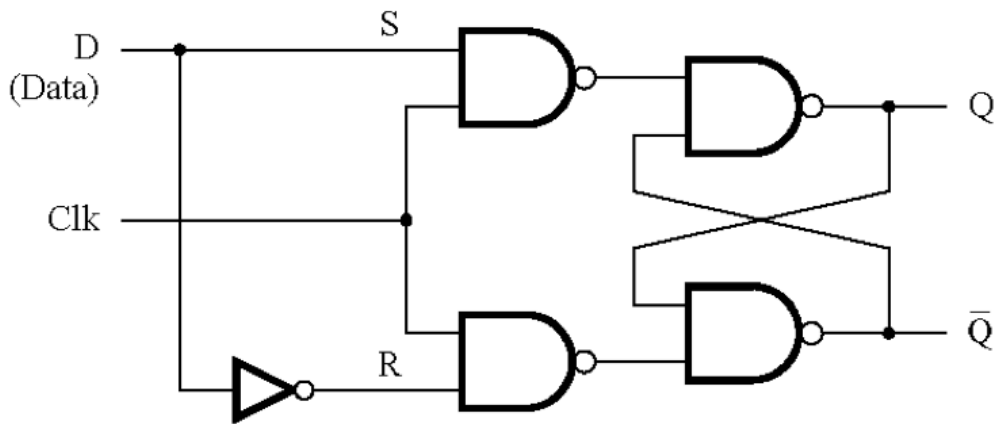


The NOT gate is now in a different place.
Also, S and R are swapped.

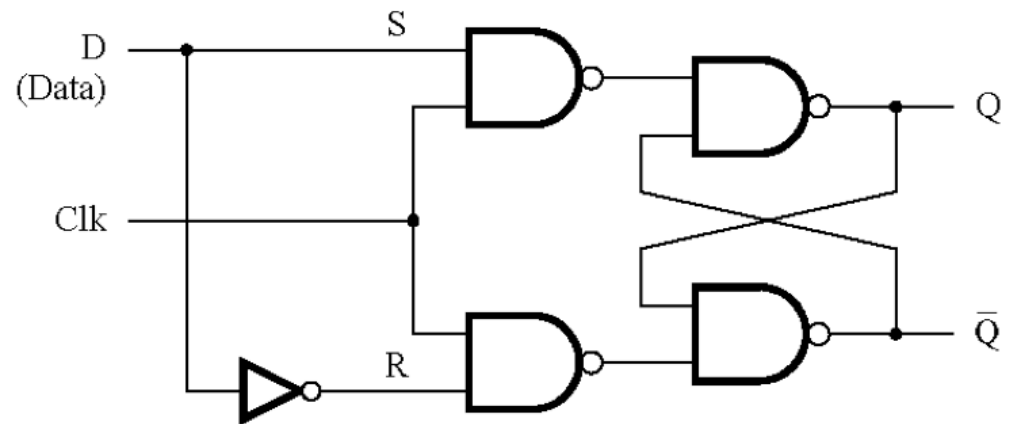
Master-Slave D Flip-Flop

Constructing a Master-Slave D Flip-Flop From Two D Latches

Master



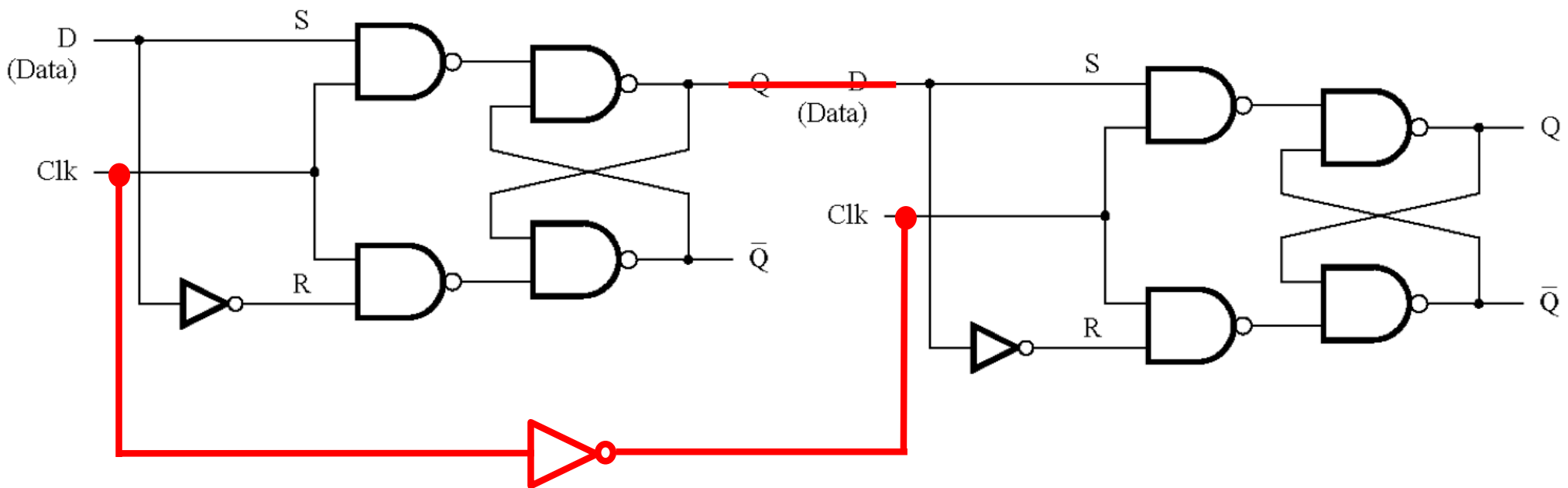
Slave



Constructing a Master-Slave D Flip-Flop From Two D Latches

Master

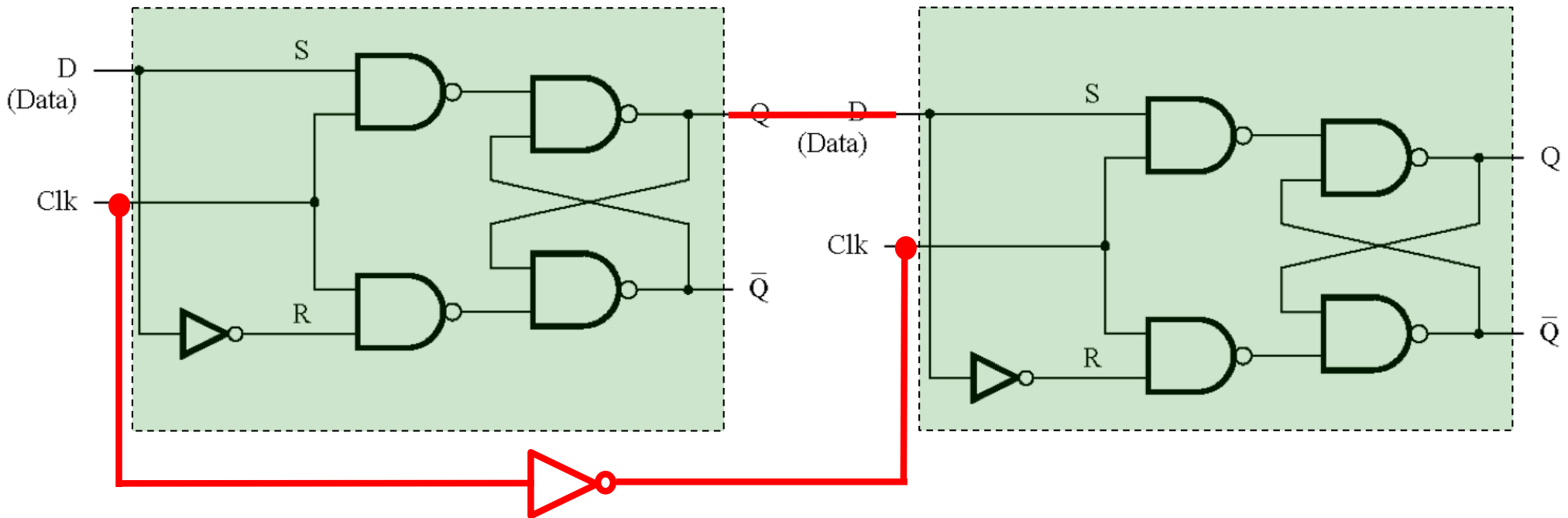
Slave



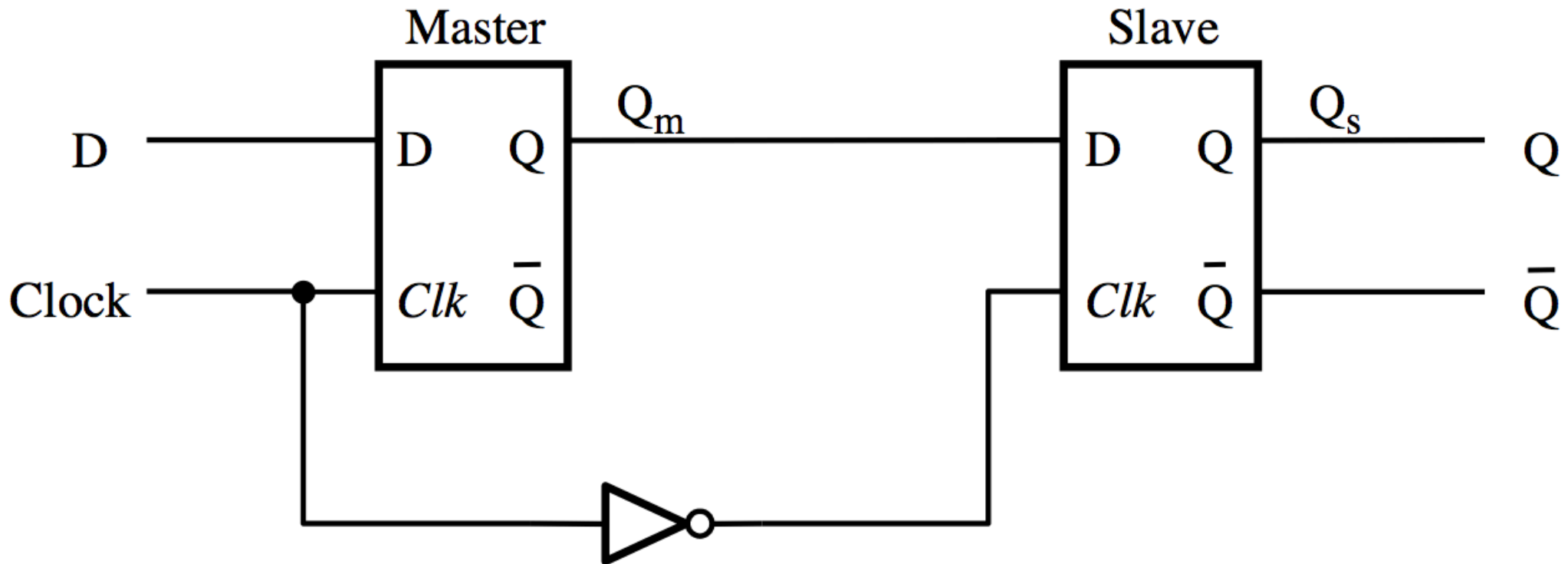
Constructing a Master-Slave D Flip-Flop From Two D Latches

Master

Slave

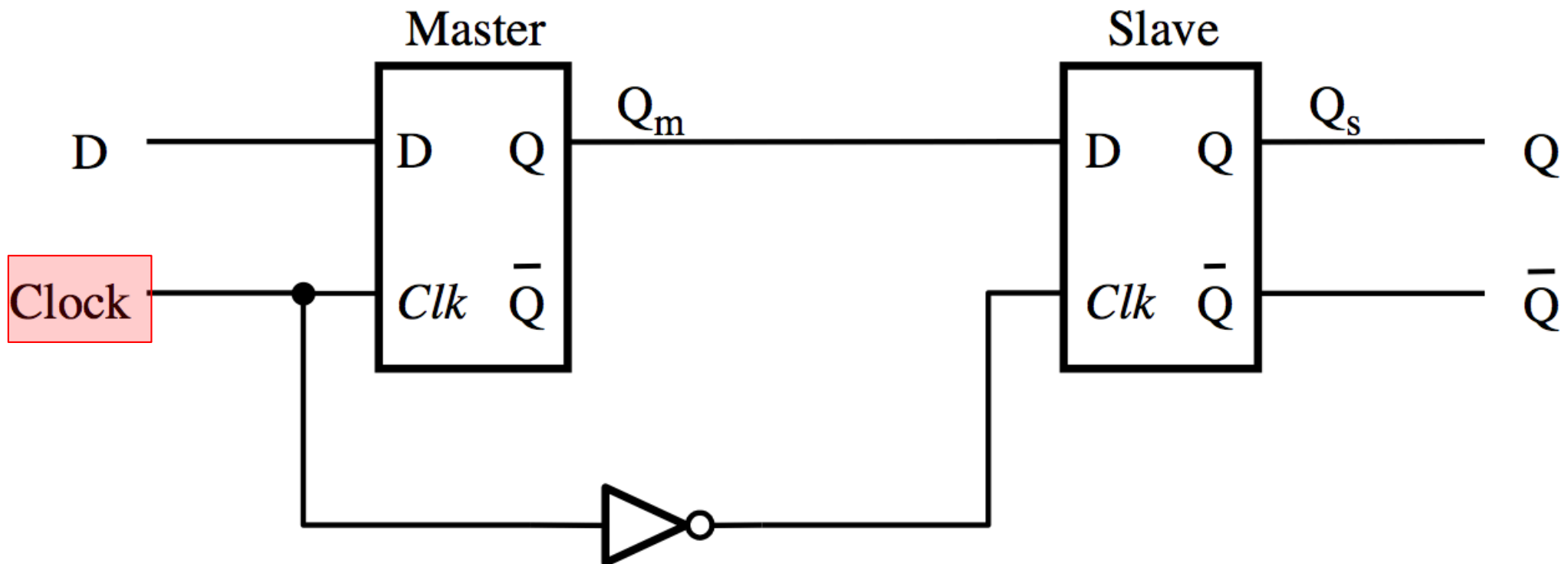


Constructing a Master-Slave D Flip-Flop From Two D Latches



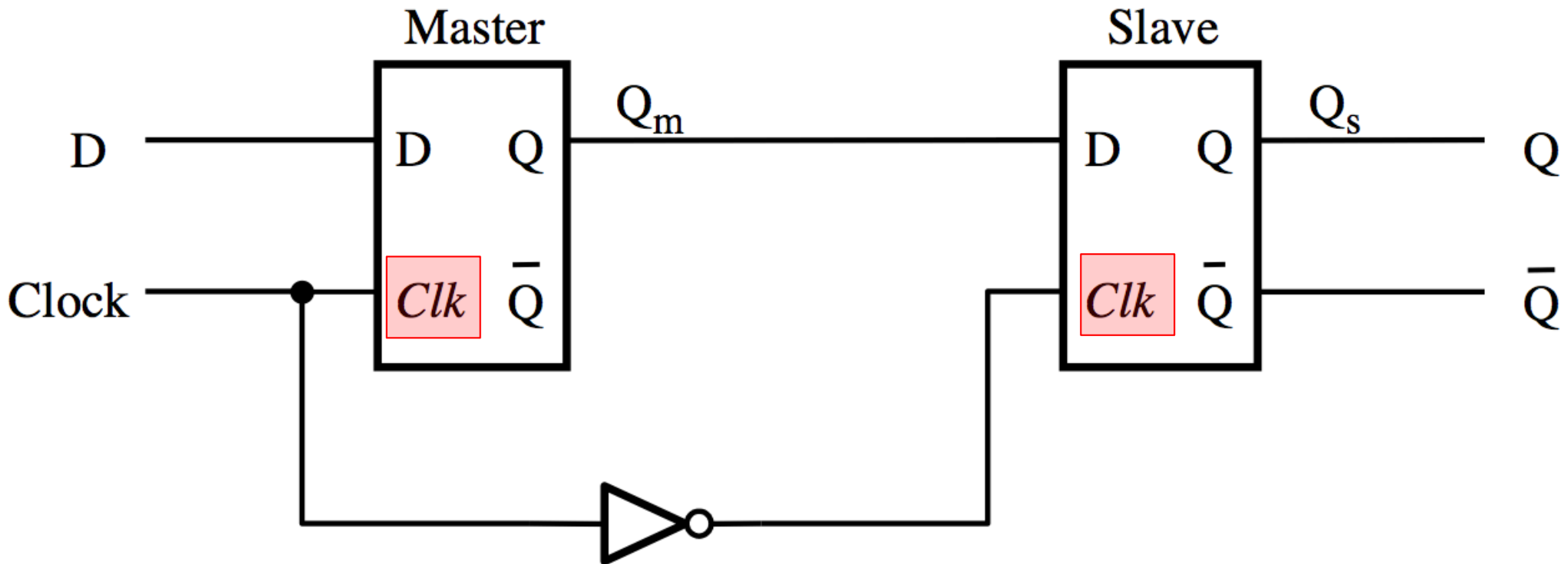
[Figure 5.9a from the textbook]

Clock is used for the D Flip-Flop



[Figure 5.9a from the textbook]

Clock is used for the D Flip-Flop, but Clk is used for each D Latch

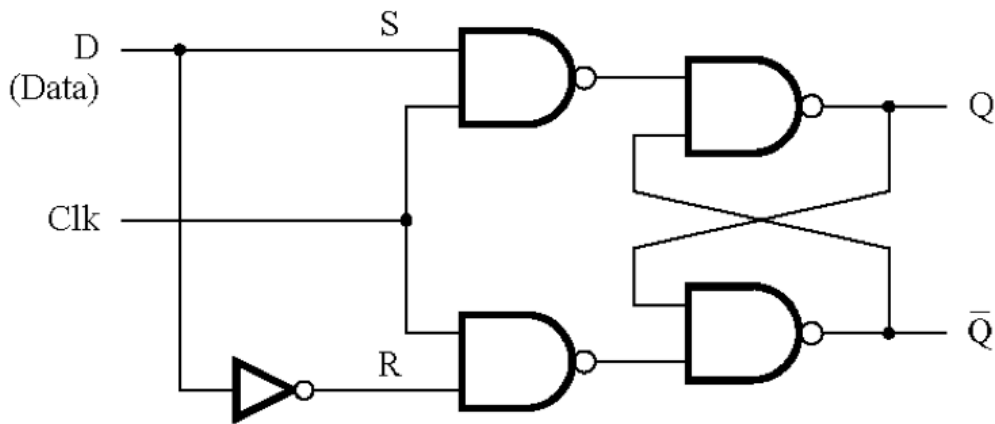


[Figure 5.9a from the textbook]

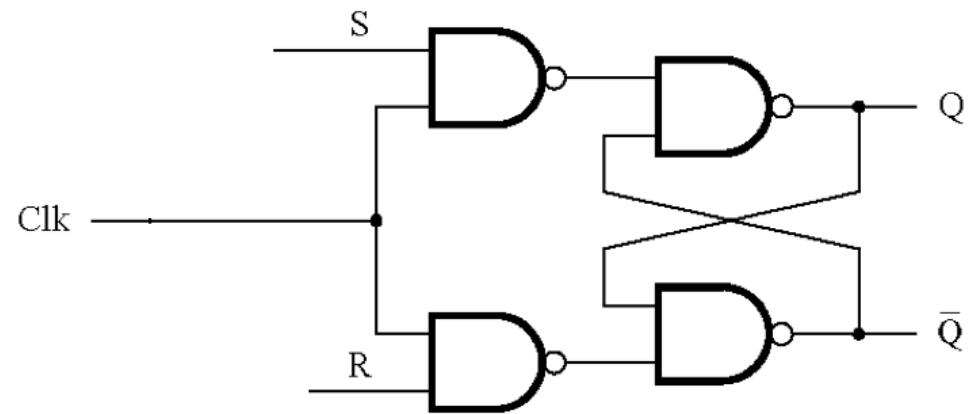
Constructing a Master-Slave D Flip-Flop From one D Latch and one Gated SR Latch

(This version uses one less NOT gate)

Master



Slave

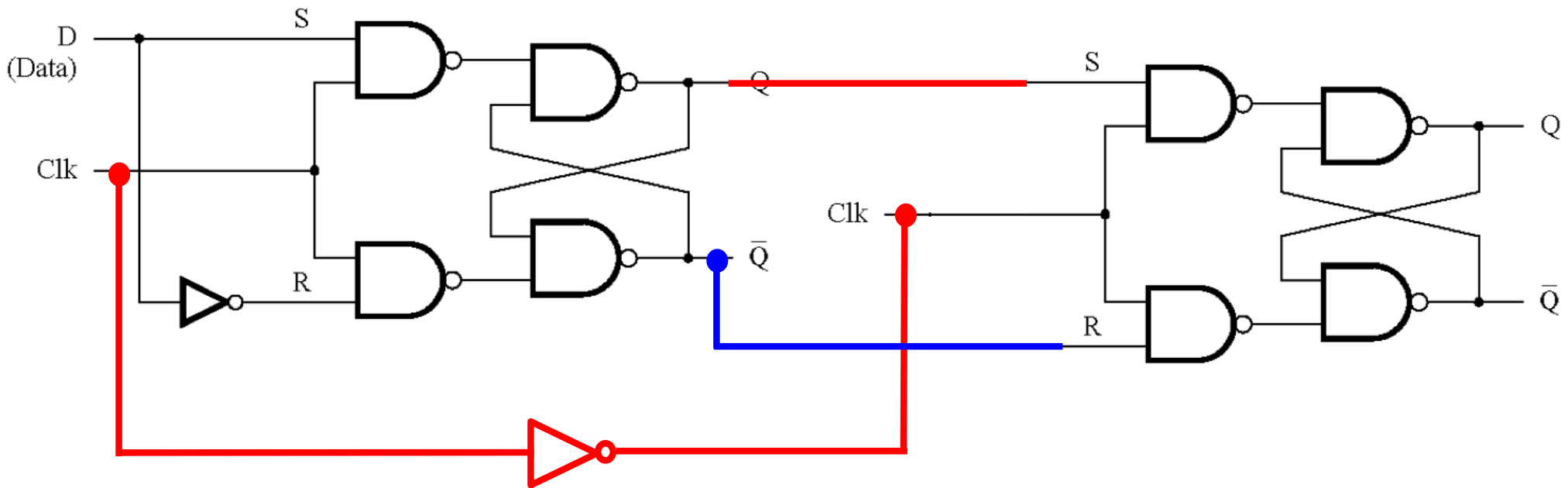


Constructing a Master-Slave D Flip-Flop From one D Latch and one Gated SR Latch

(This version uses one less NOT gate)

Master

Slave

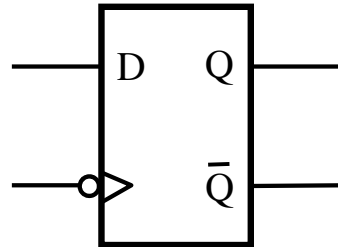


Edge-Triggered D Flip-Flops

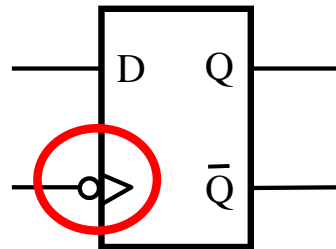
Motivation

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

Graphical Symbol for the Master-Slave D Flip-Flop



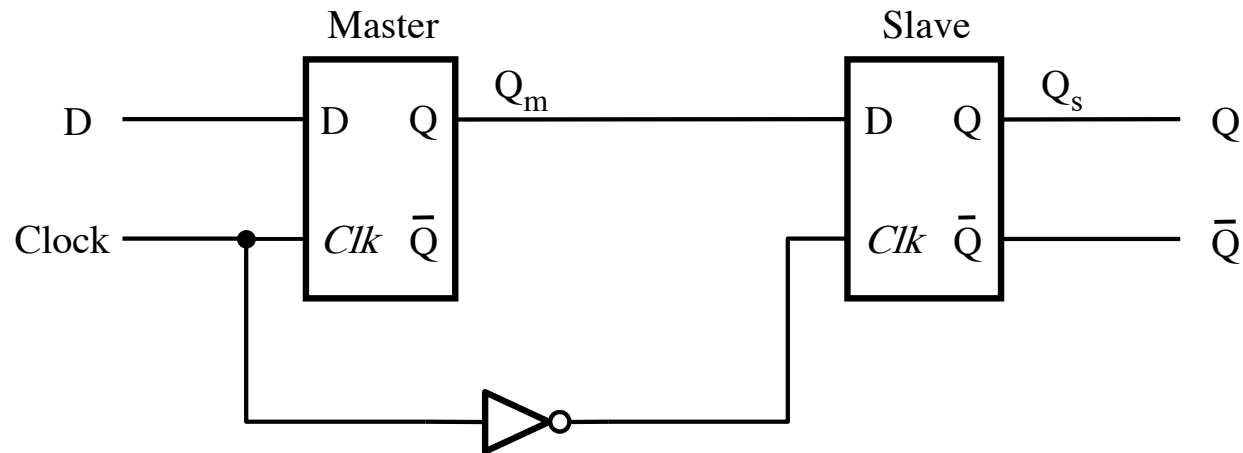
Graphical Symbol for the Master-Slave D Flip-Flop



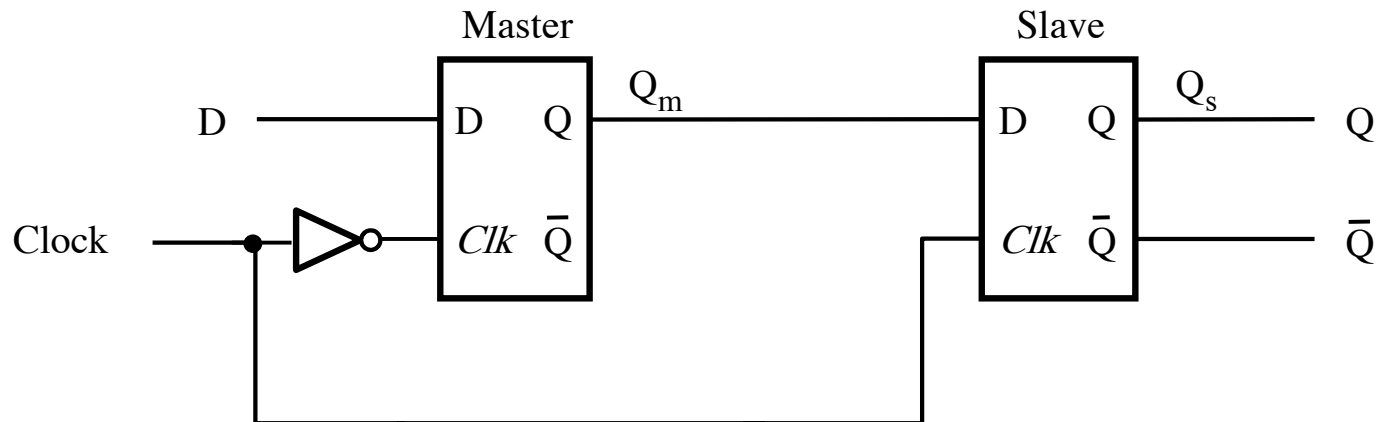
The $>$ means that this is edge-triggered

The small circle means that it is the negative edge

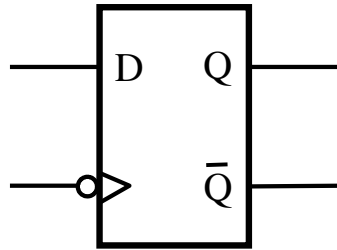
Negative-Edge-Triggered Master-Slave D Flip-Flop



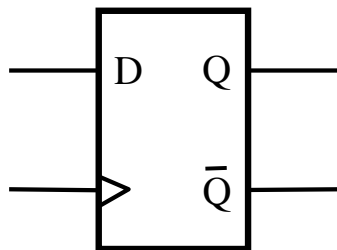
Positive-Edge-Triggered Master-Slave D Flip-Flop



Negative-Edge-Triggered Master-Slave D Flip-Flop

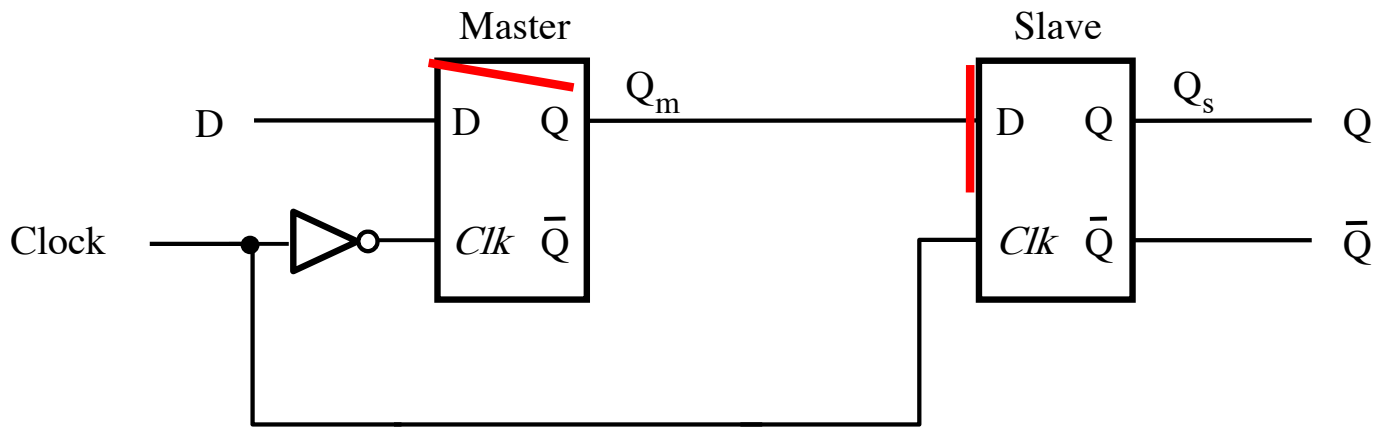


Positive-Edge-Triggered Master-Slave D Flip-Flop

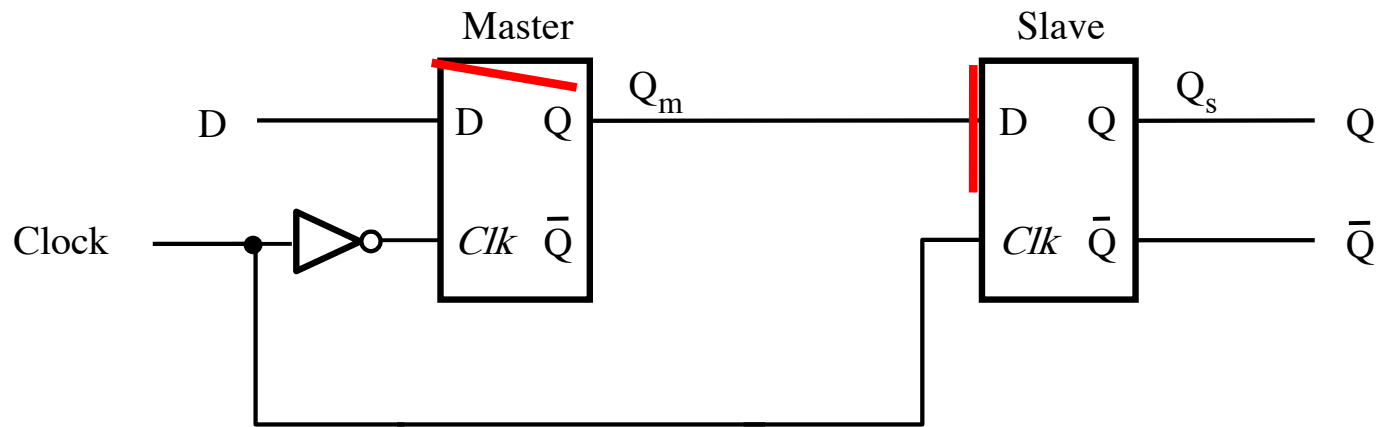


D Flip-Flop: A Double Door Analogy

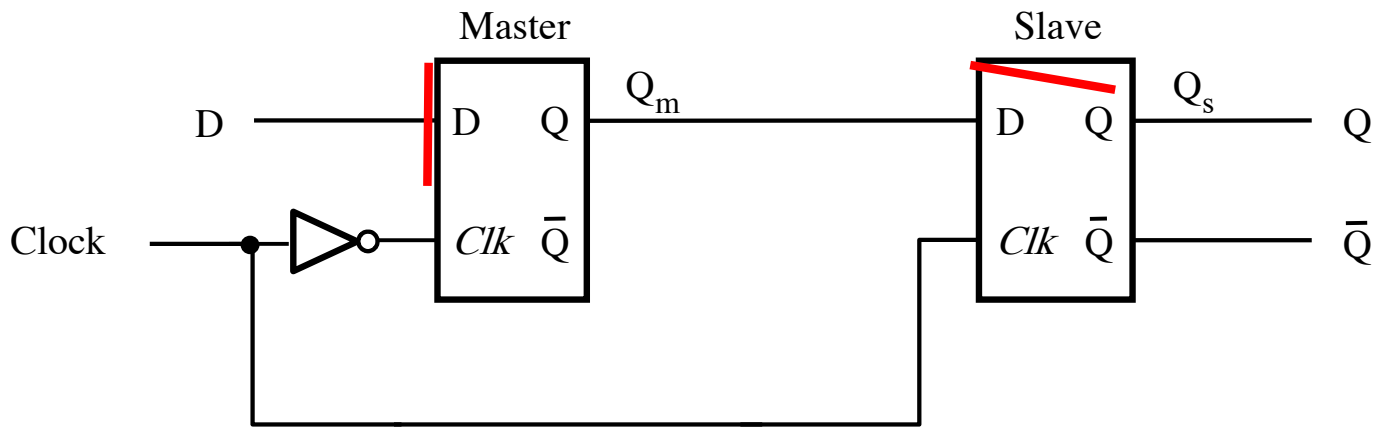
Positive-Edge-Triggered Master-Slave D Flip-Flop



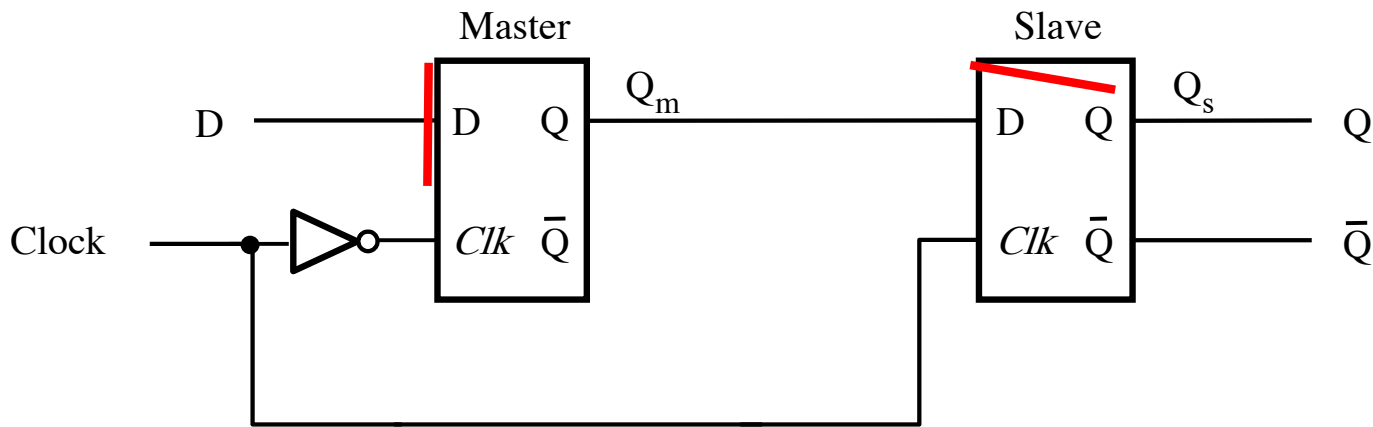
Positive-Edge-Triggered Master-Slave D Flip-Flop



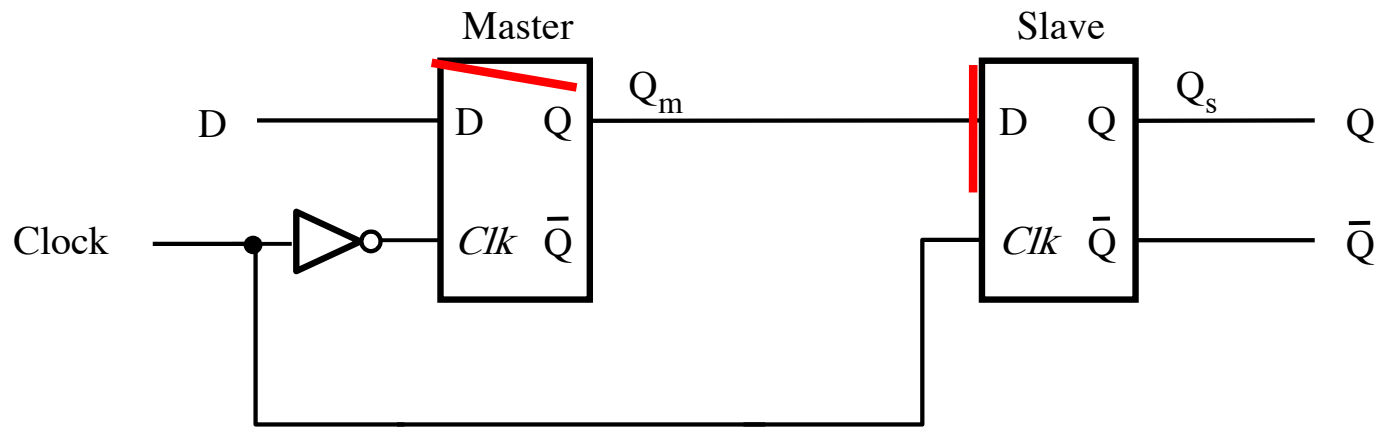
Positive-Edge-Triggered Master-Slave D Flip-Flop



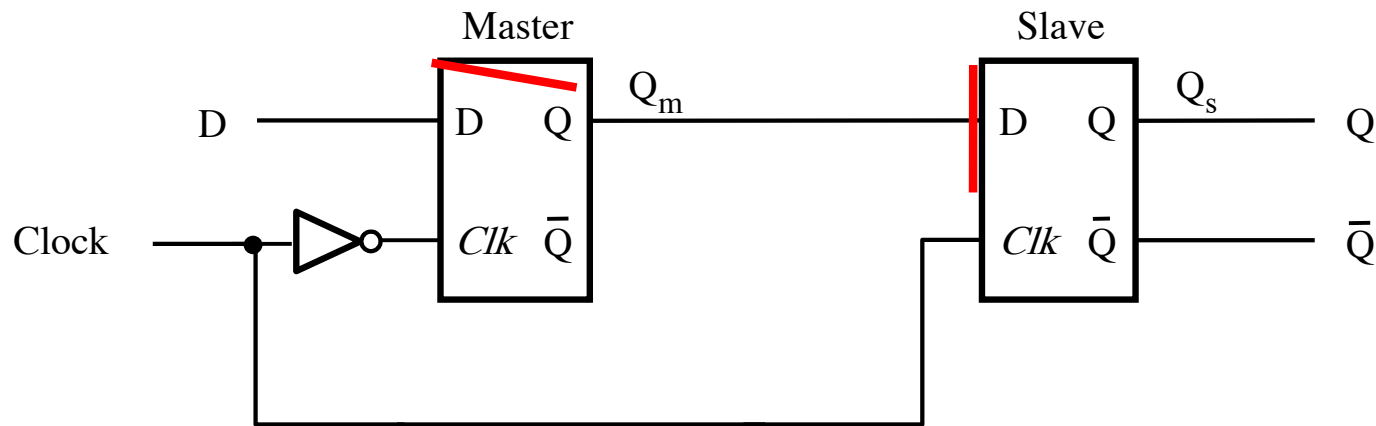
Positive-Edge-Triggered Master-Slave D Flip-Flop



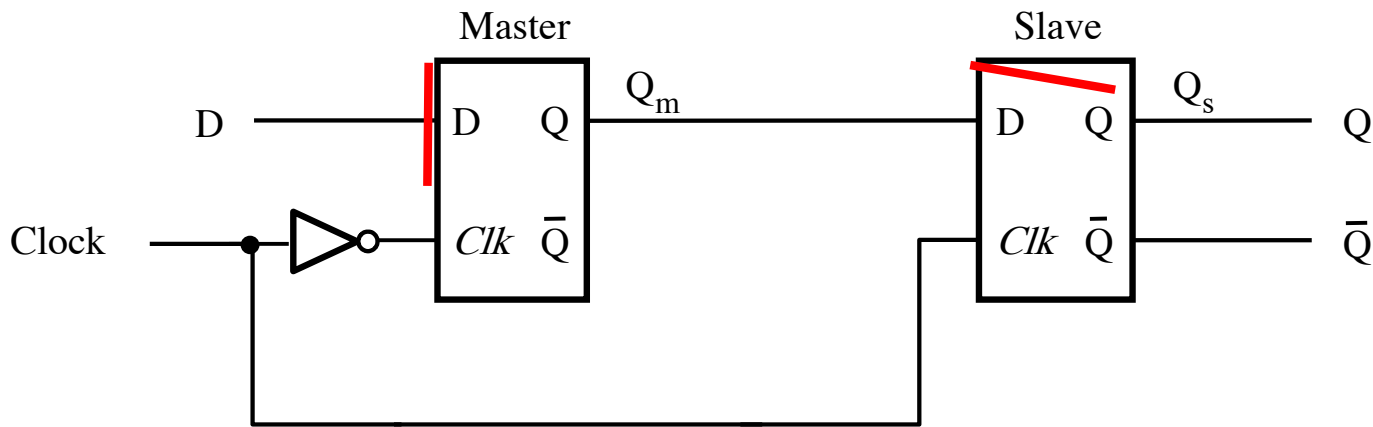
Positive-Edge-Triggered Master-Slave D Flip-Flop



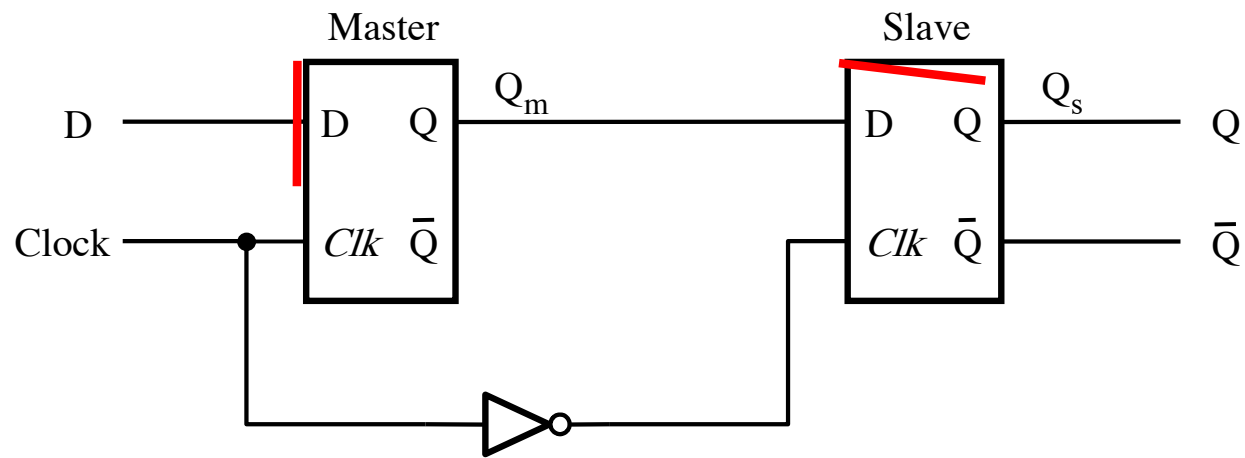
Positive-Edge-Triggered Master-Slave D Flip-Flop



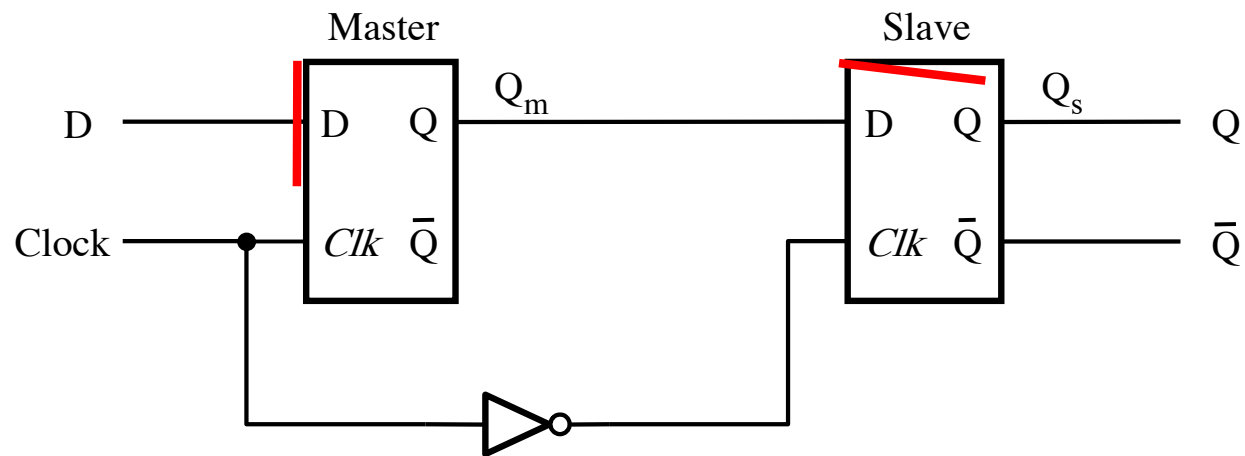
Positive-Edge-Triggered Master-Slave D Flip-Flop



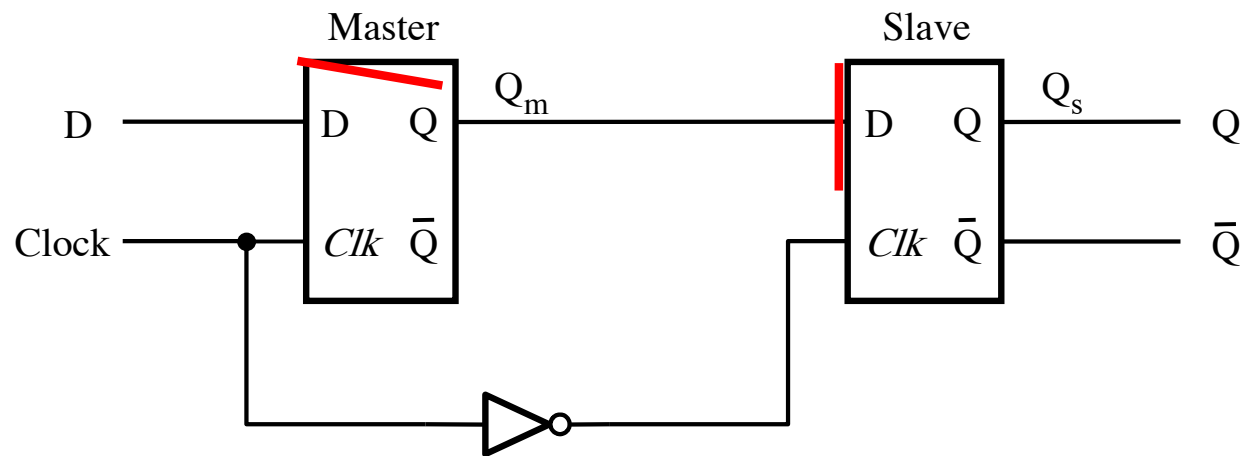
Negative-Edge-Triggered Master-Slave D Flip-Flop



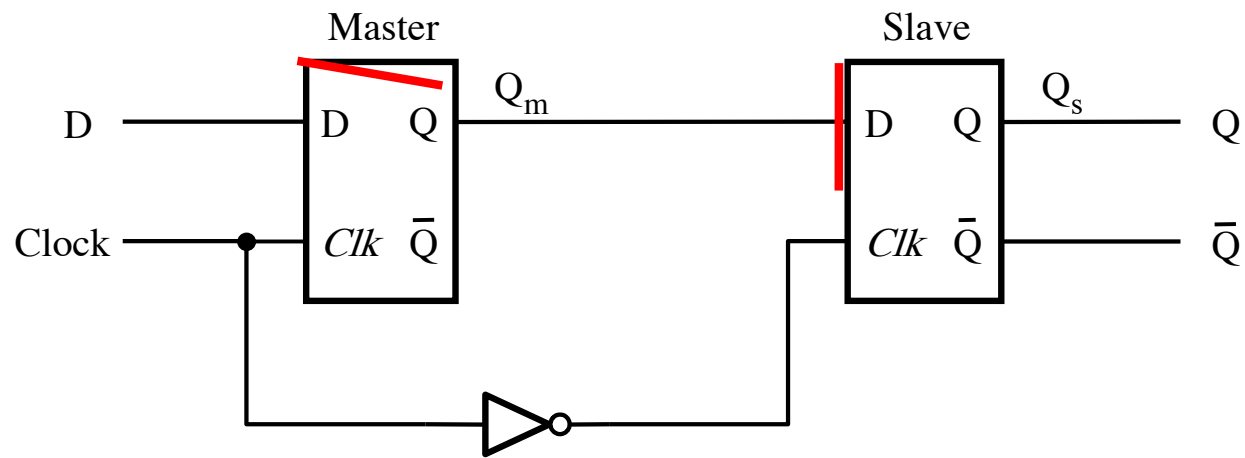
Negative-Edge-Triggered Master-Slave D Flip-Flop



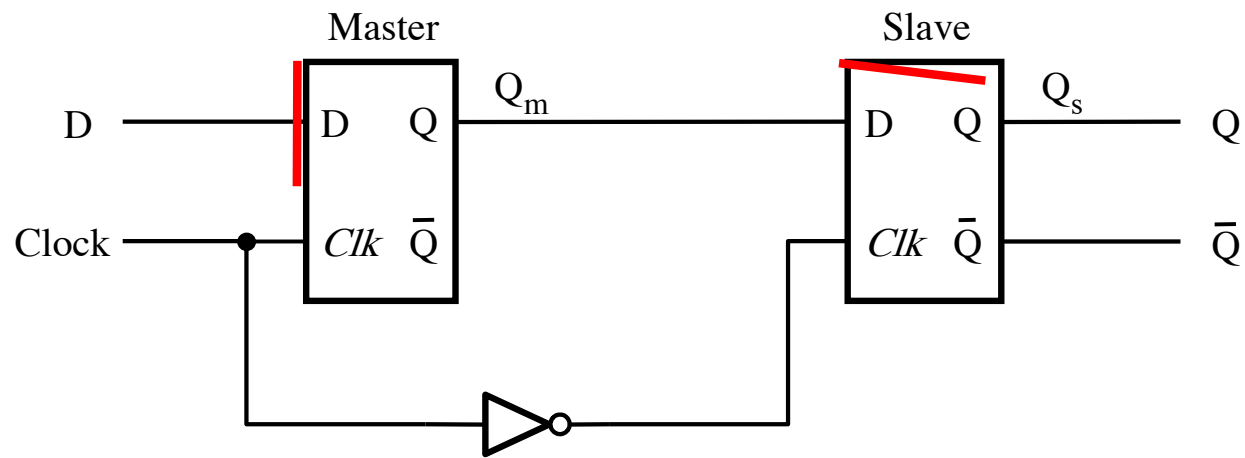
Negative-Edge-Triggered Master-Slave D Flip-Flop



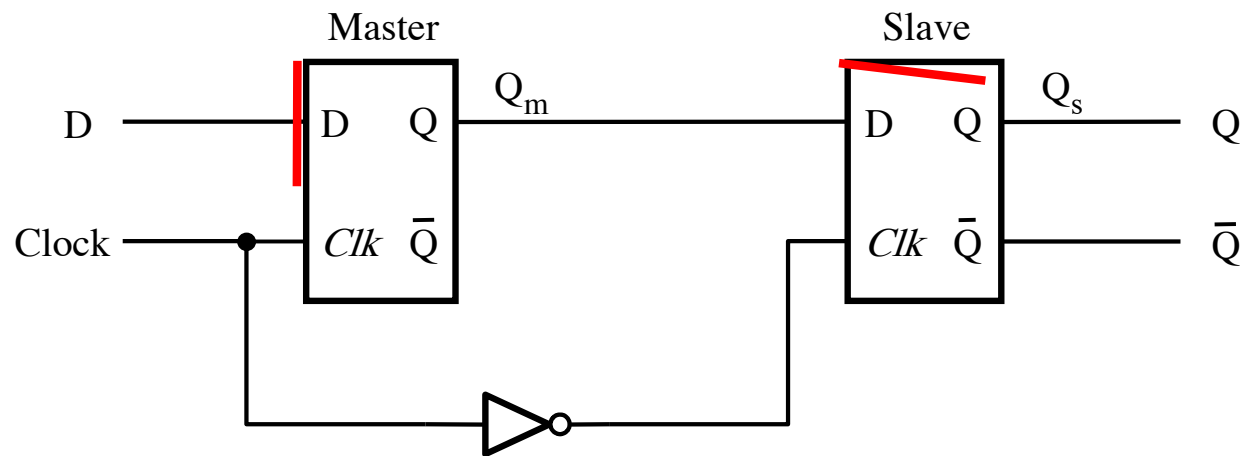
Negative-Edge-Triggered Master-Slave D Flip-Flop



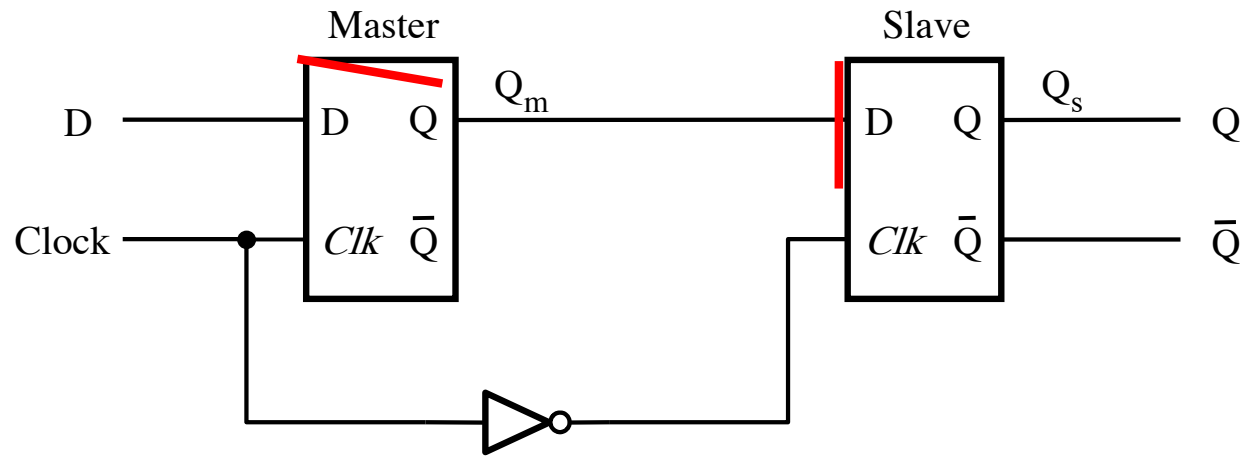
Negative-Edge-Triggered Master-Slave D Flip-Flop



Negative-Edge-Triggered Master-Slave D Flip-Flop



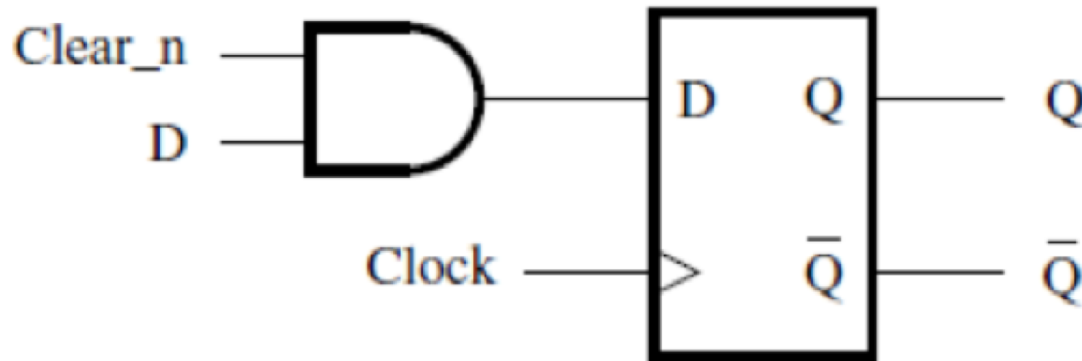
Negative-Edge-Triggered Master-Slave D Flip-Flop



Positive-edge-triggered D flip-flop with Clear and Preset

Positive-edge-triggered D flip-flop with Clear_n and Preset_n

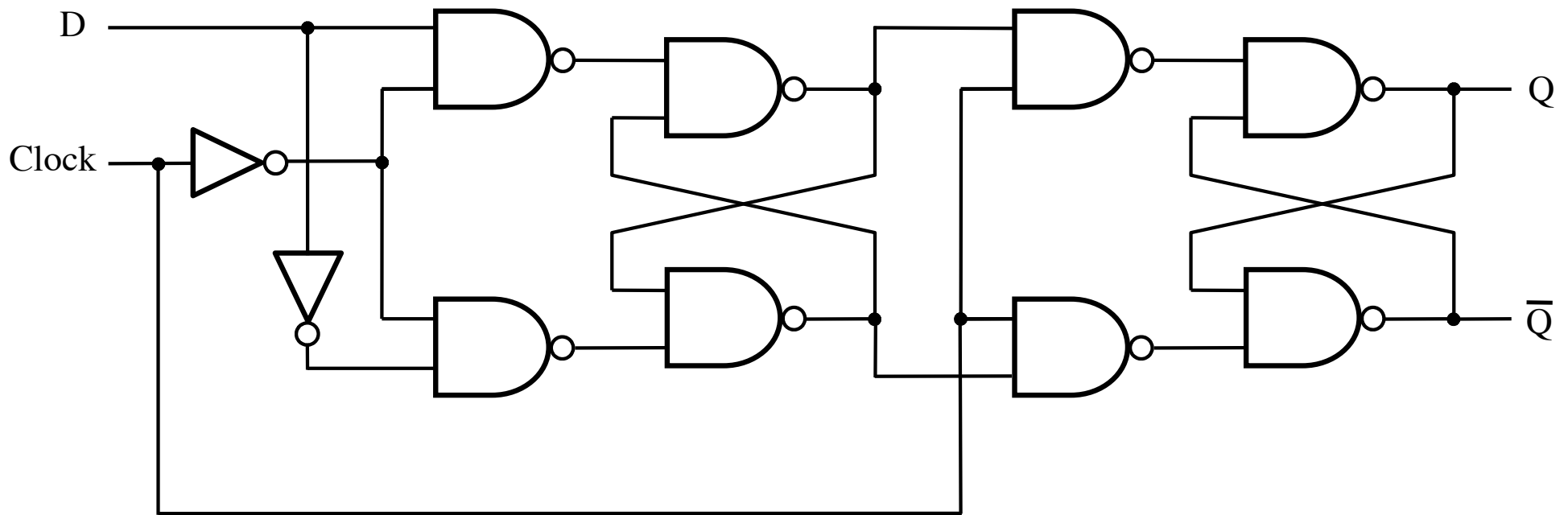
Positive-edge-triggered D flip-flop with **Synchronous** Clear



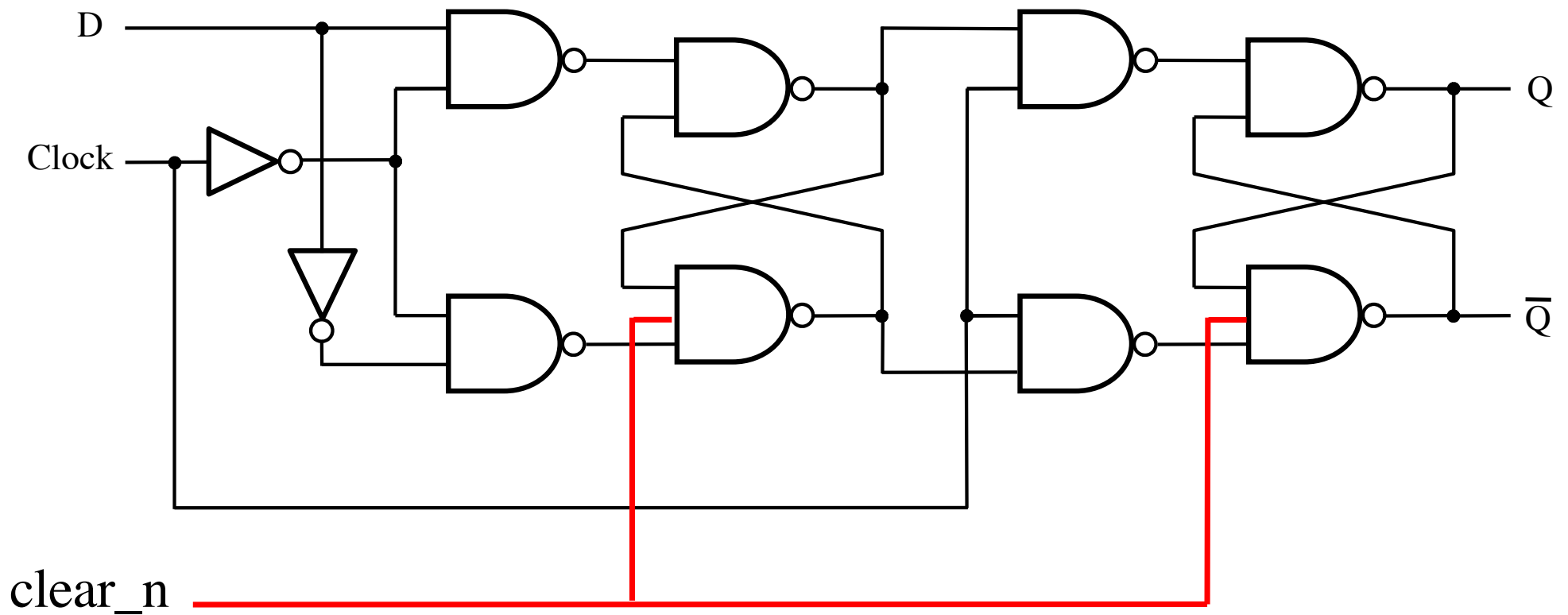
(c) Adding a synchronous clear

The output Q can be cleared only on the positive clock edge.

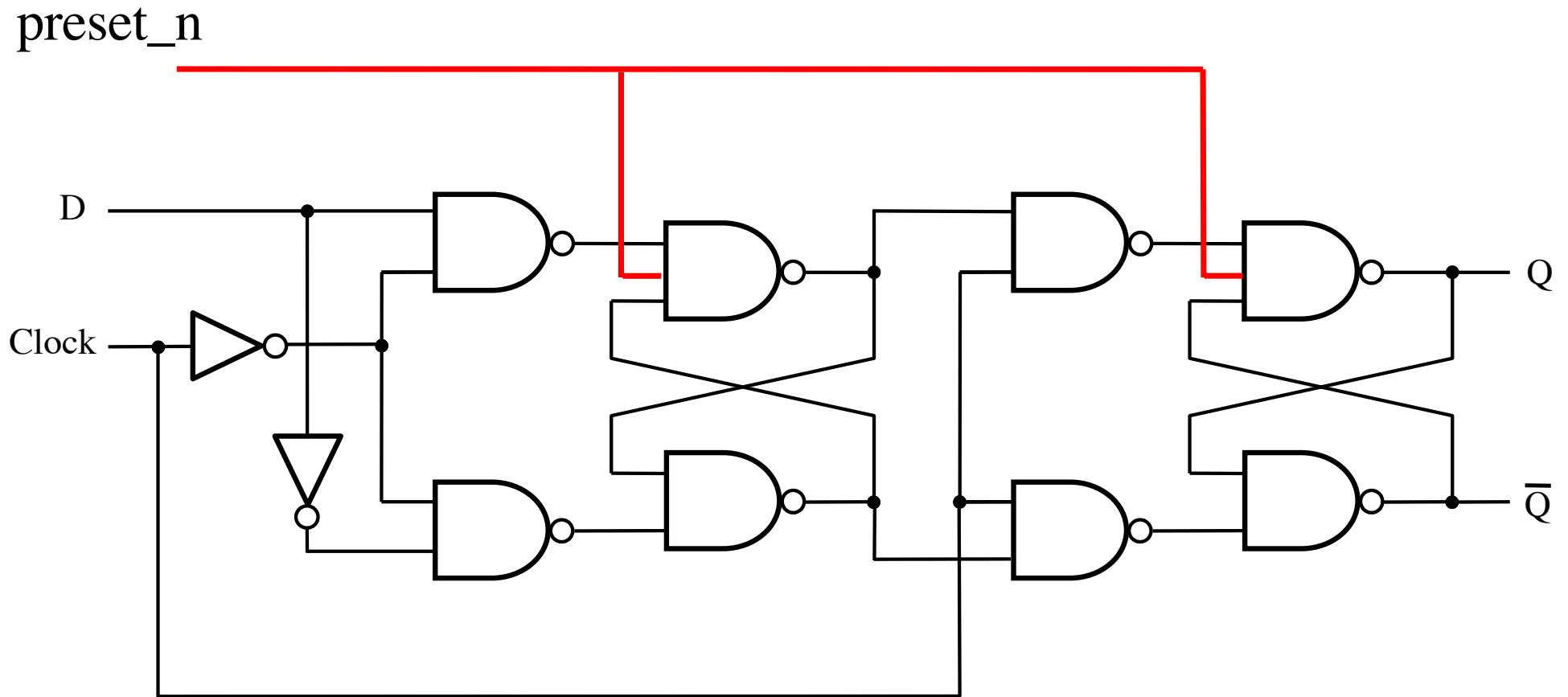
The Complete Wiring Diagram for a Positive-Edge-Triggered D Flip-Flop



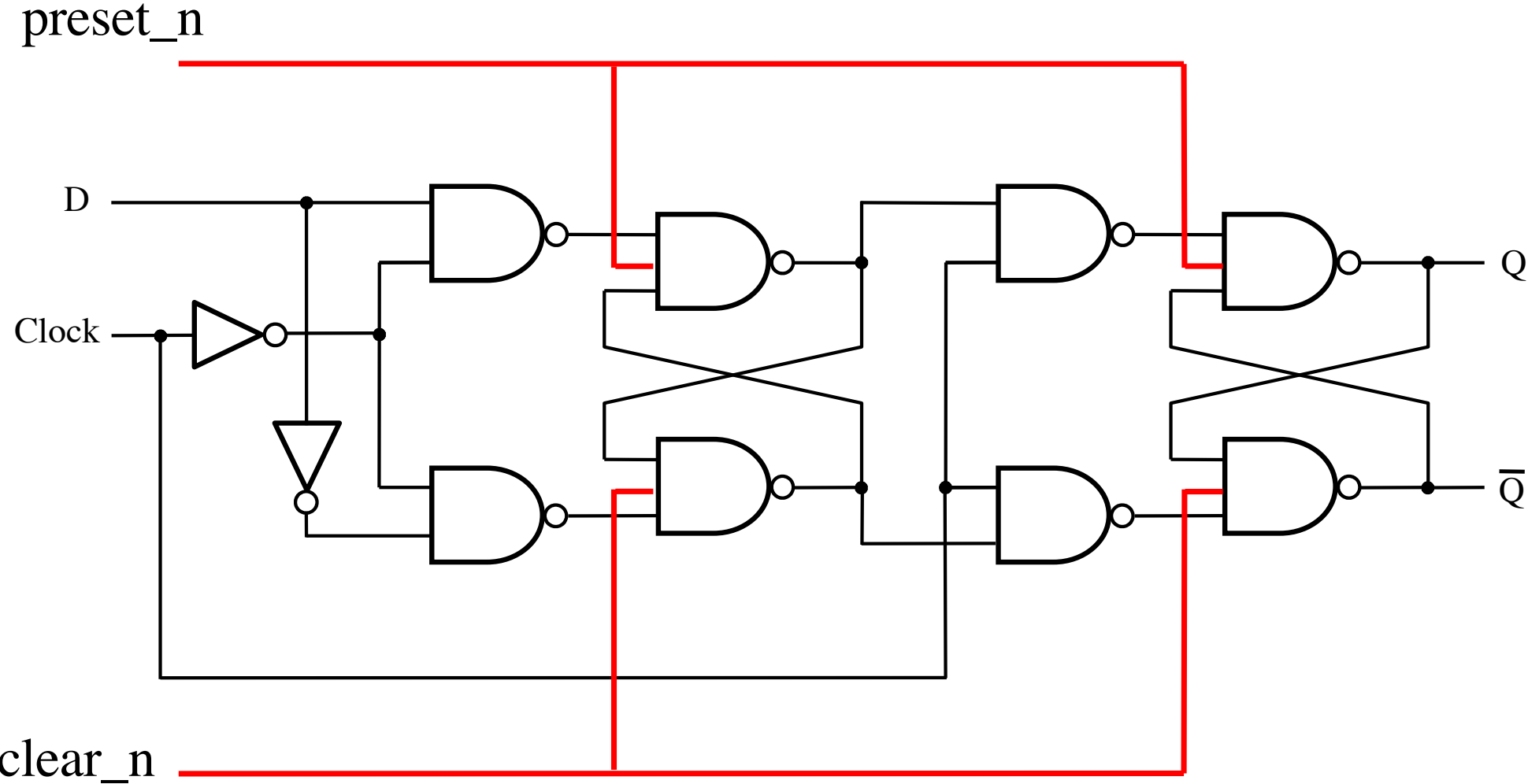
Adding an Asynchronous Clear



Adding an Asynchronous Preset

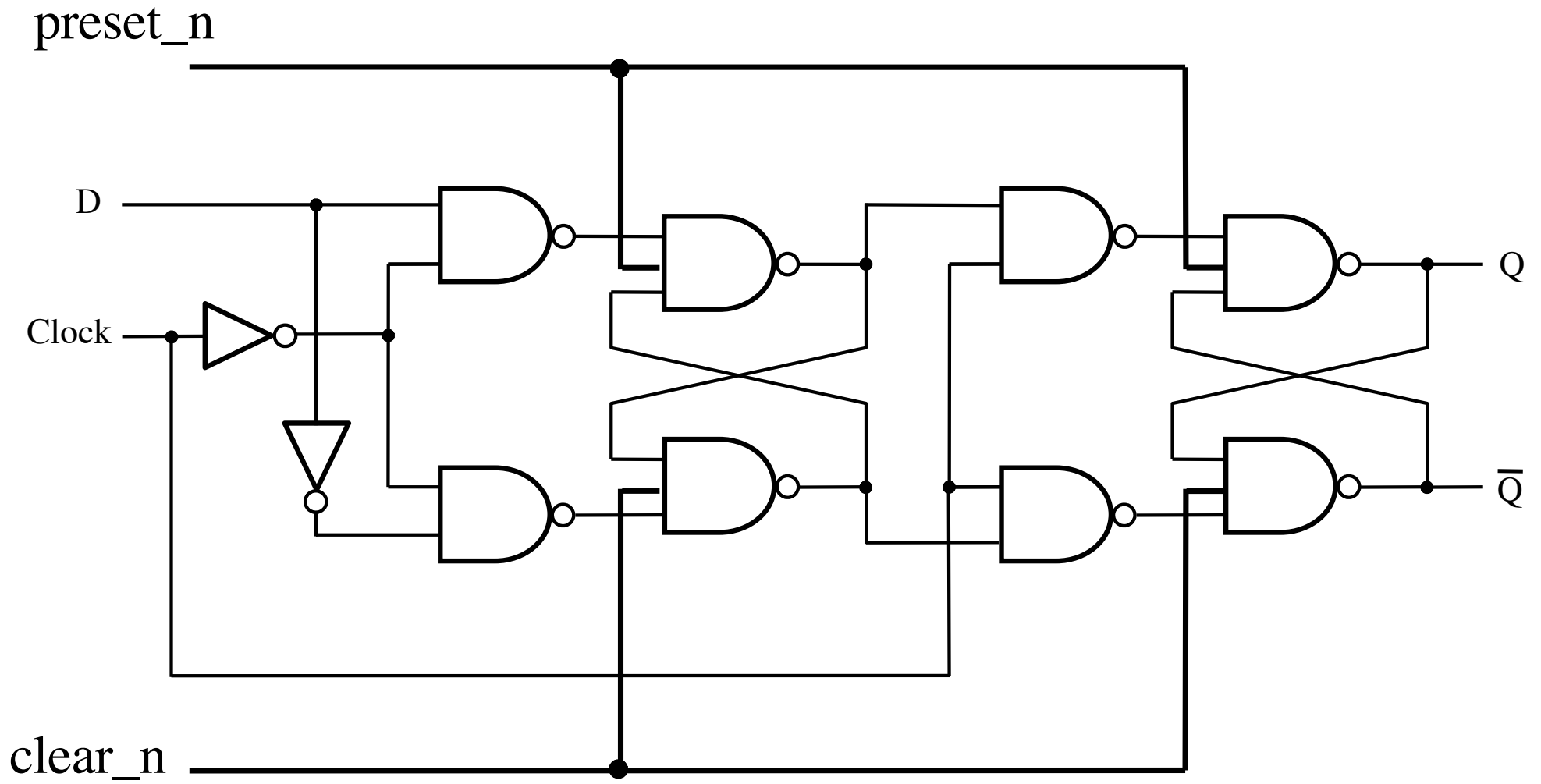


Positive-Edge-Triggered D Flip-Flop with **Asynchronous** Clear and Preset

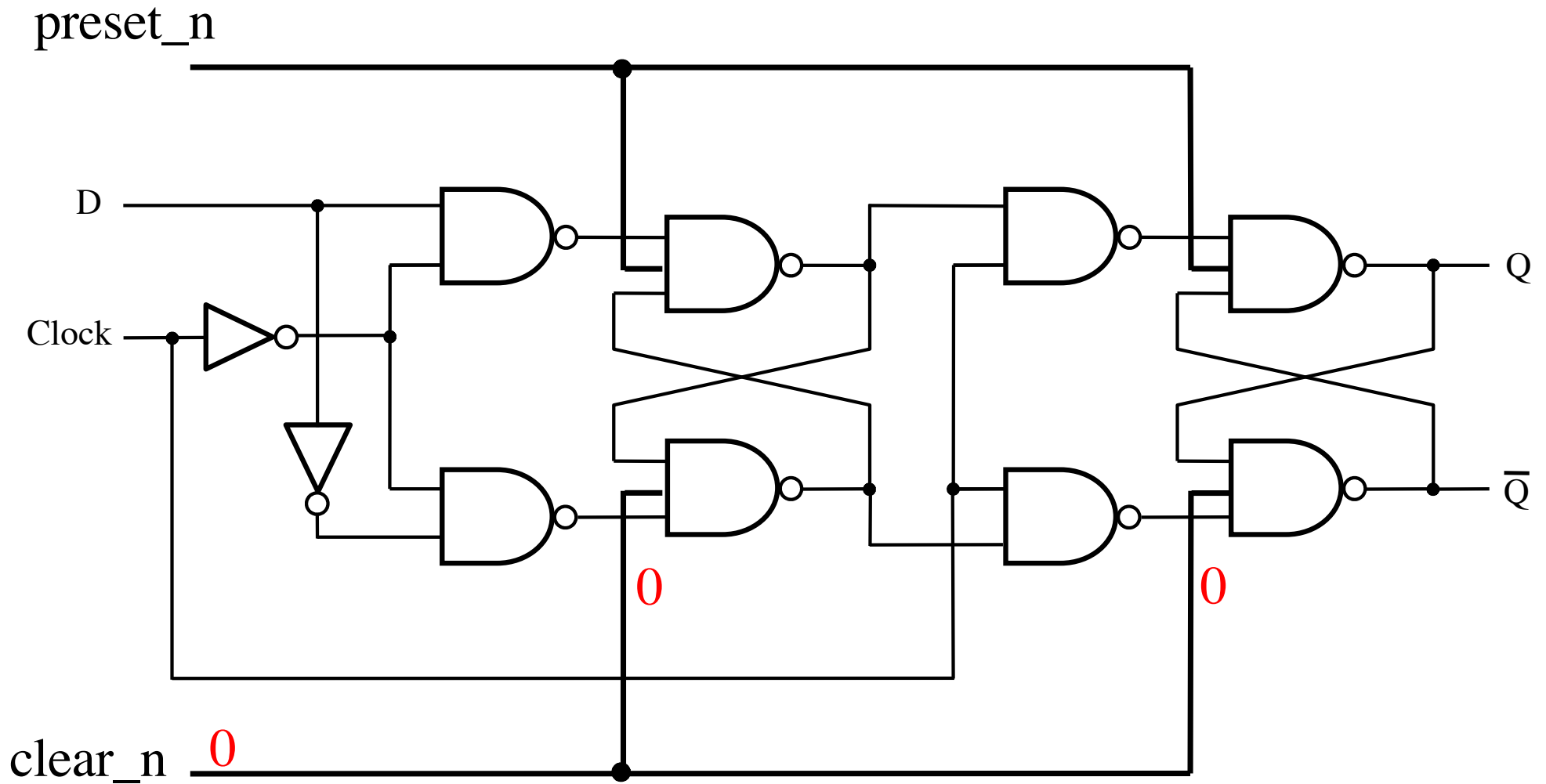


How does clear work?

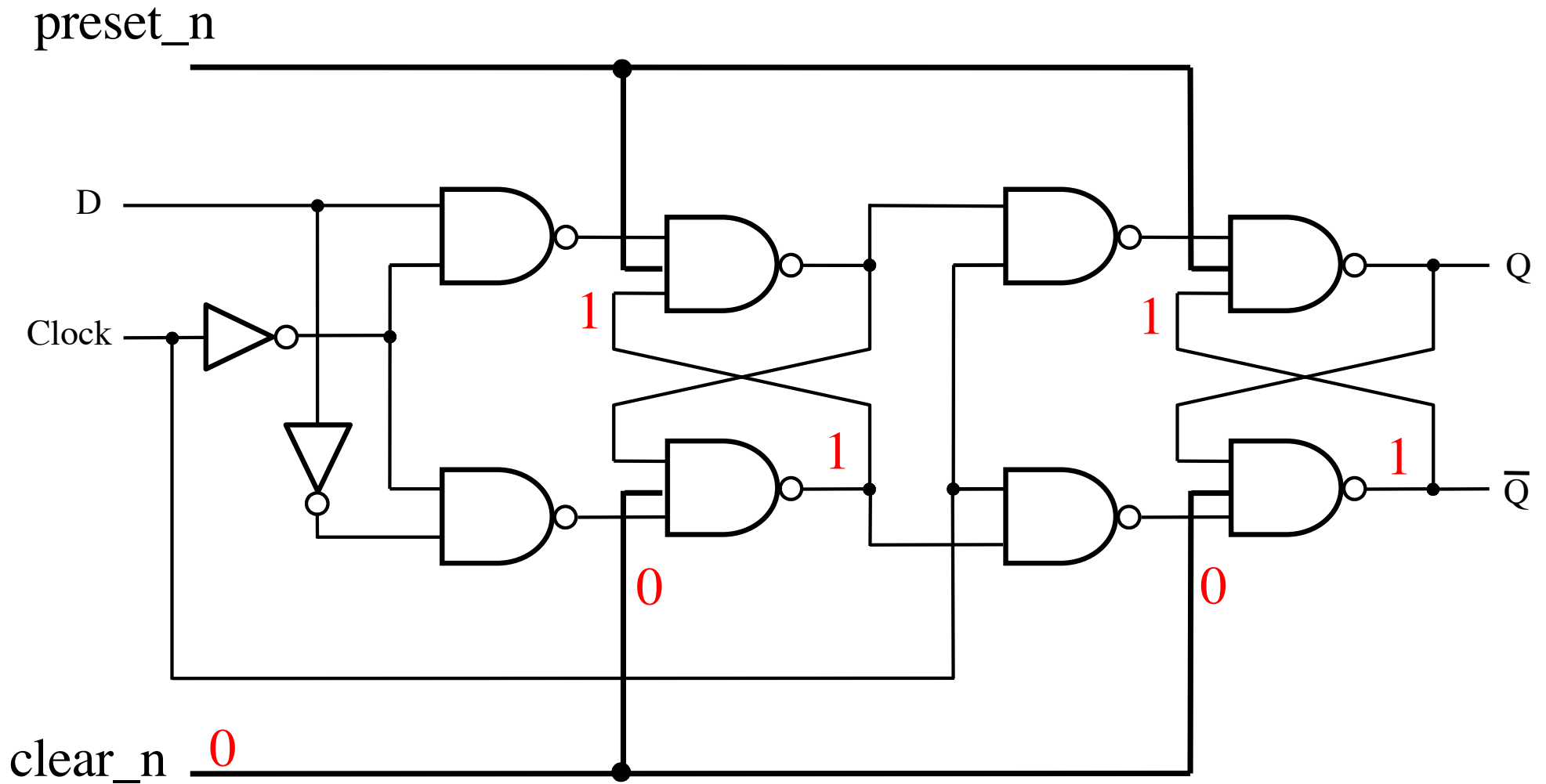
How does clear work?



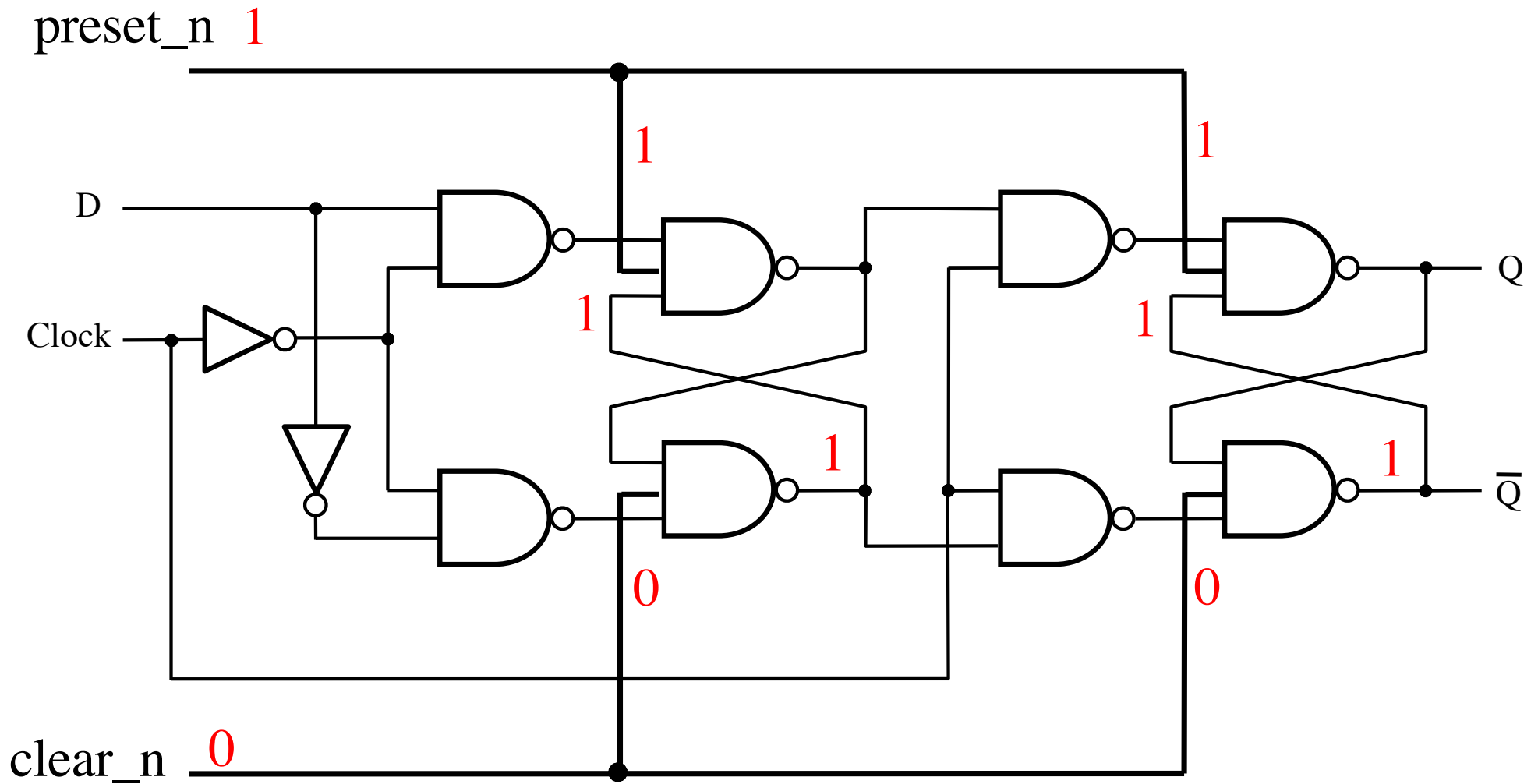
How does clear work?



How does clear work?

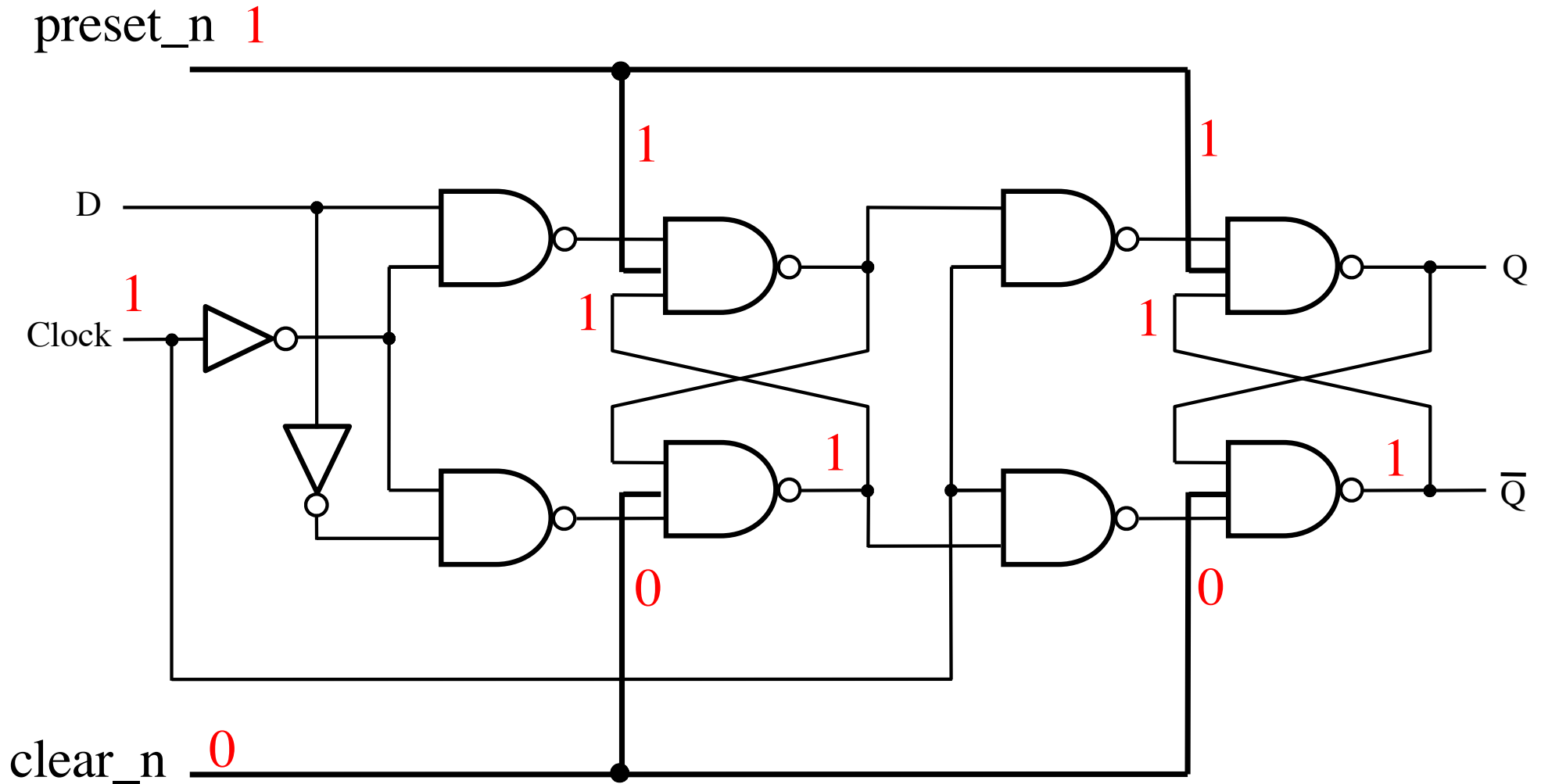


How does clear work?



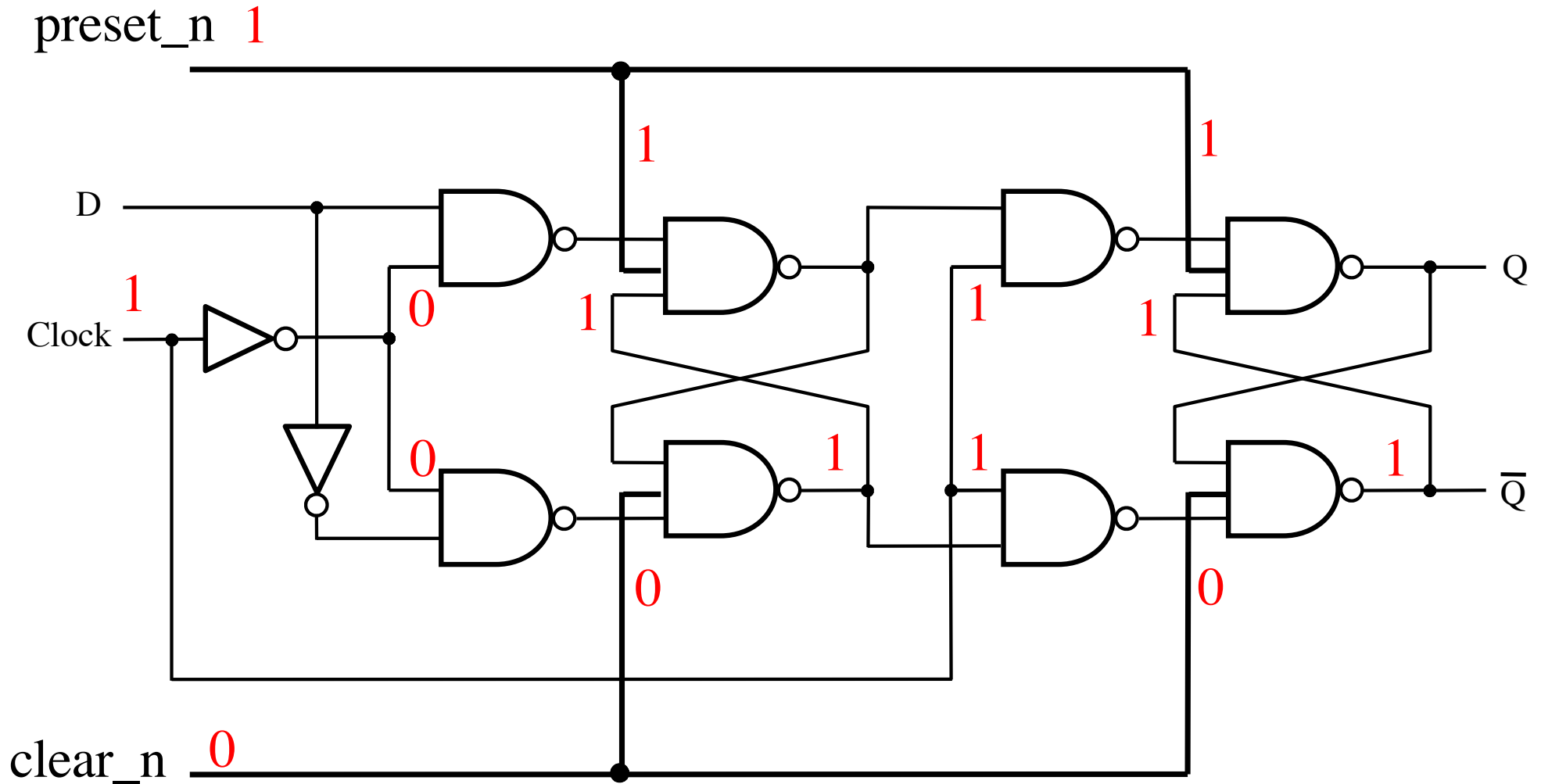
How does clear work?

At this point we need to consider two cases: Clock=1 v.s. Clock =0



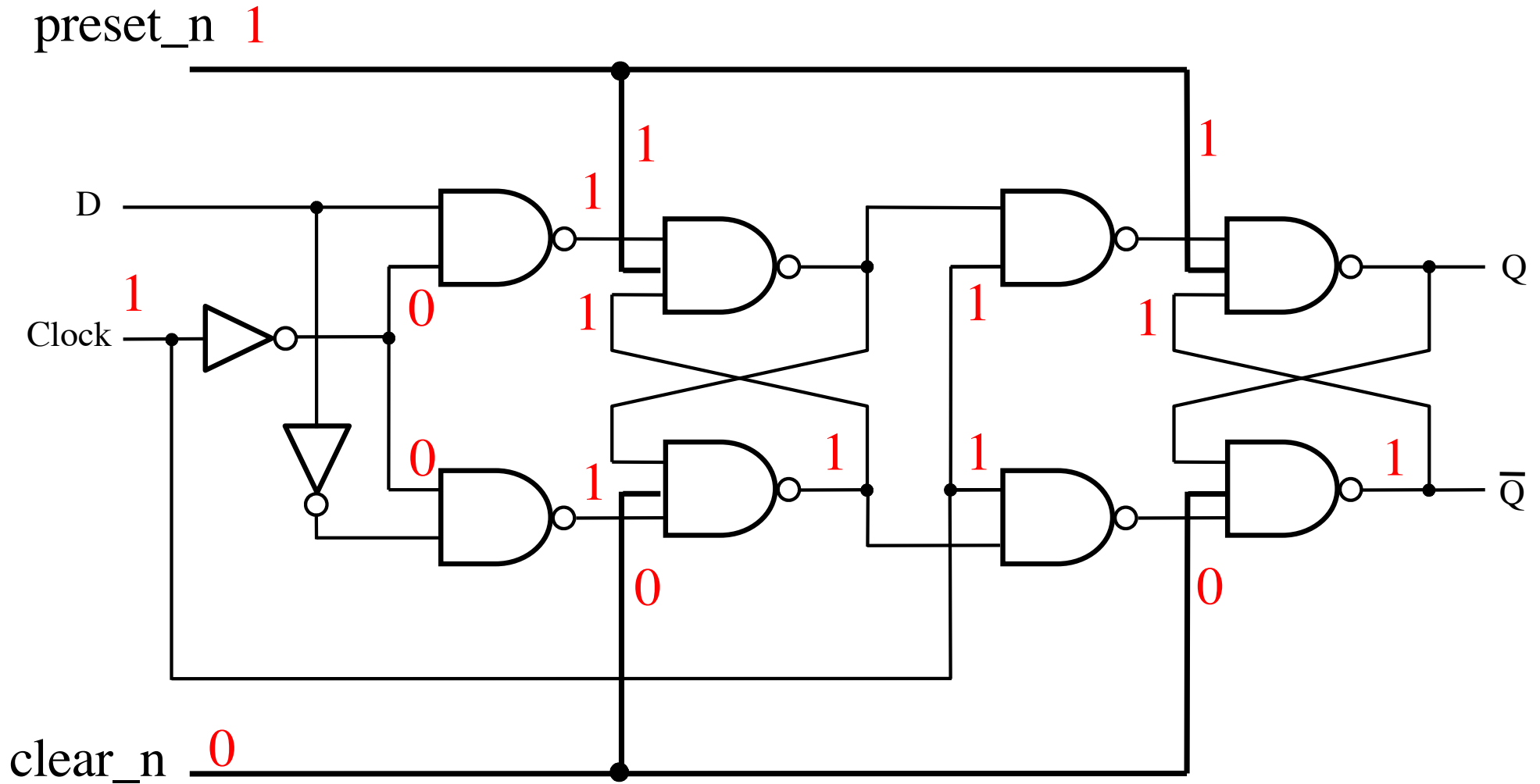
How does clear work?

Clock=1



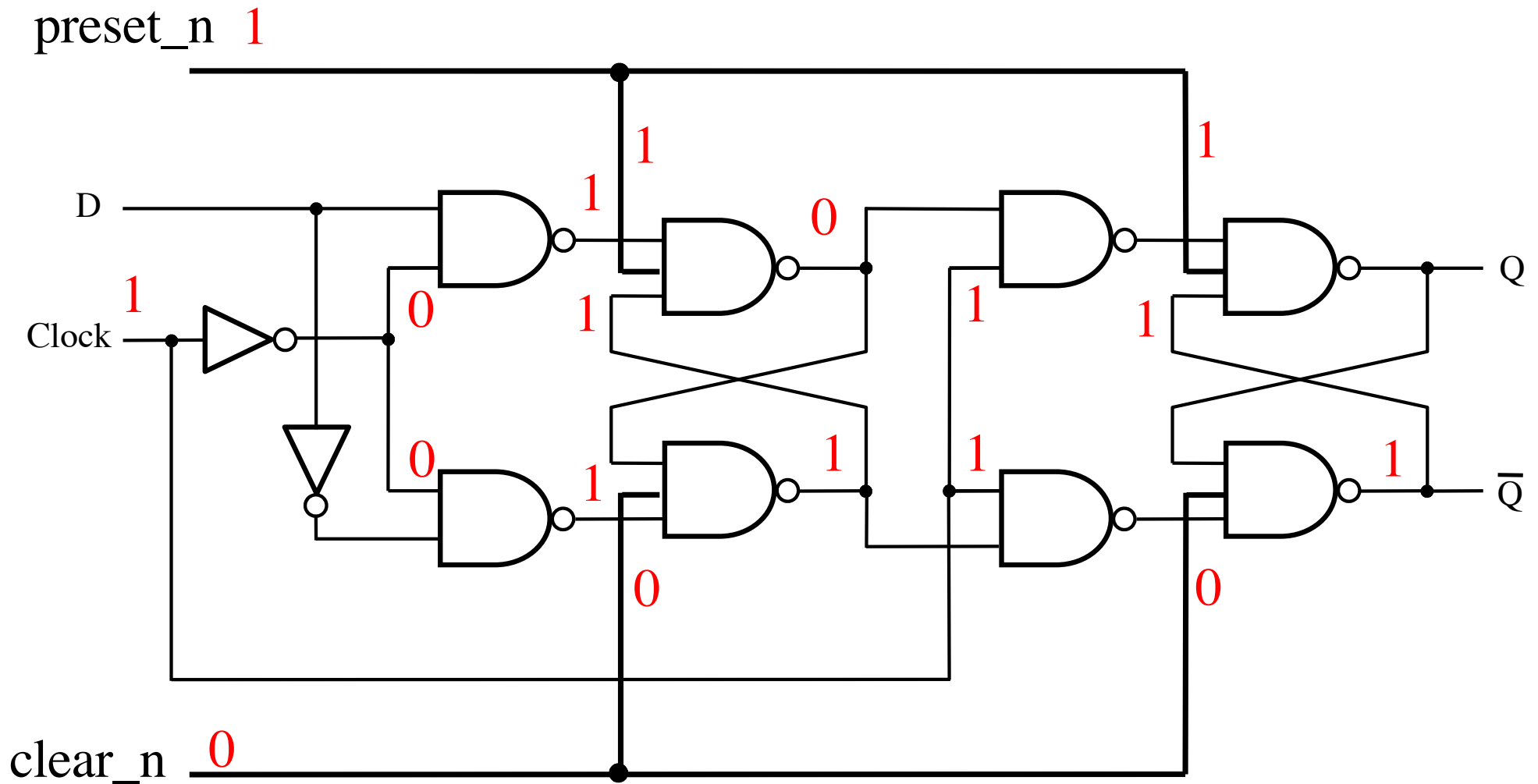
How does clear work?

Clock=1



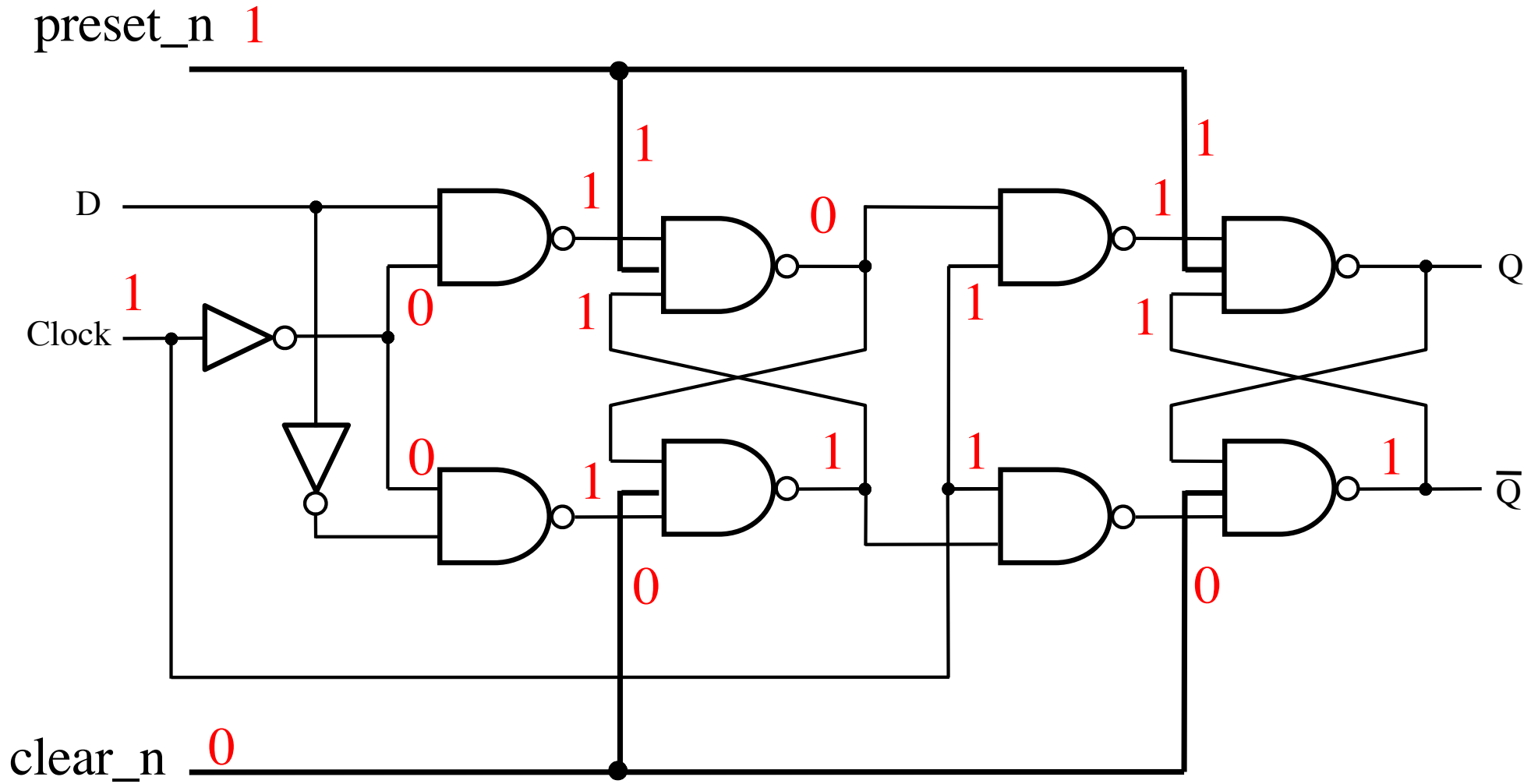
How does clear work?

Clock=1



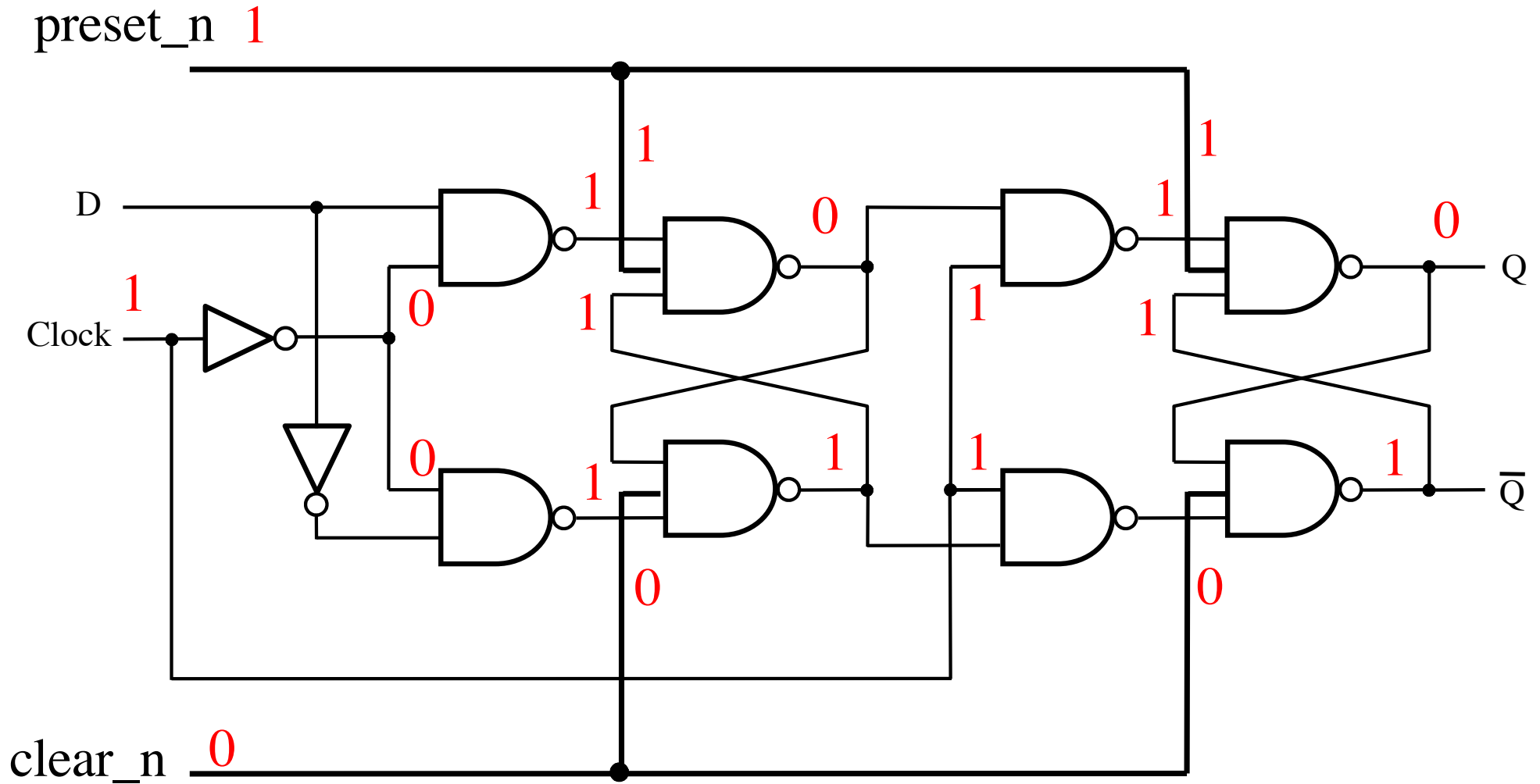
How does clear work?

Clock=1



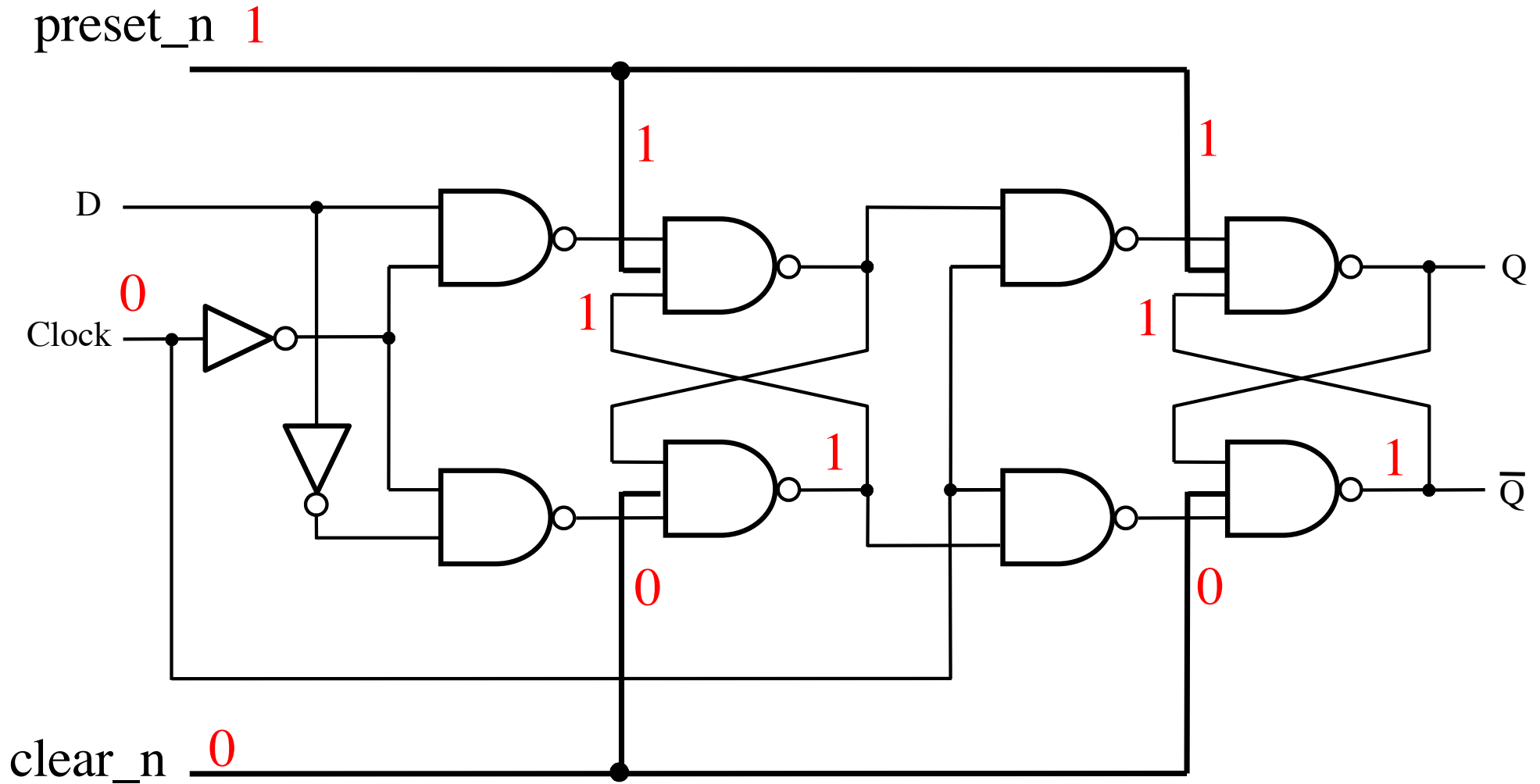
How does clear work?

Clock=1



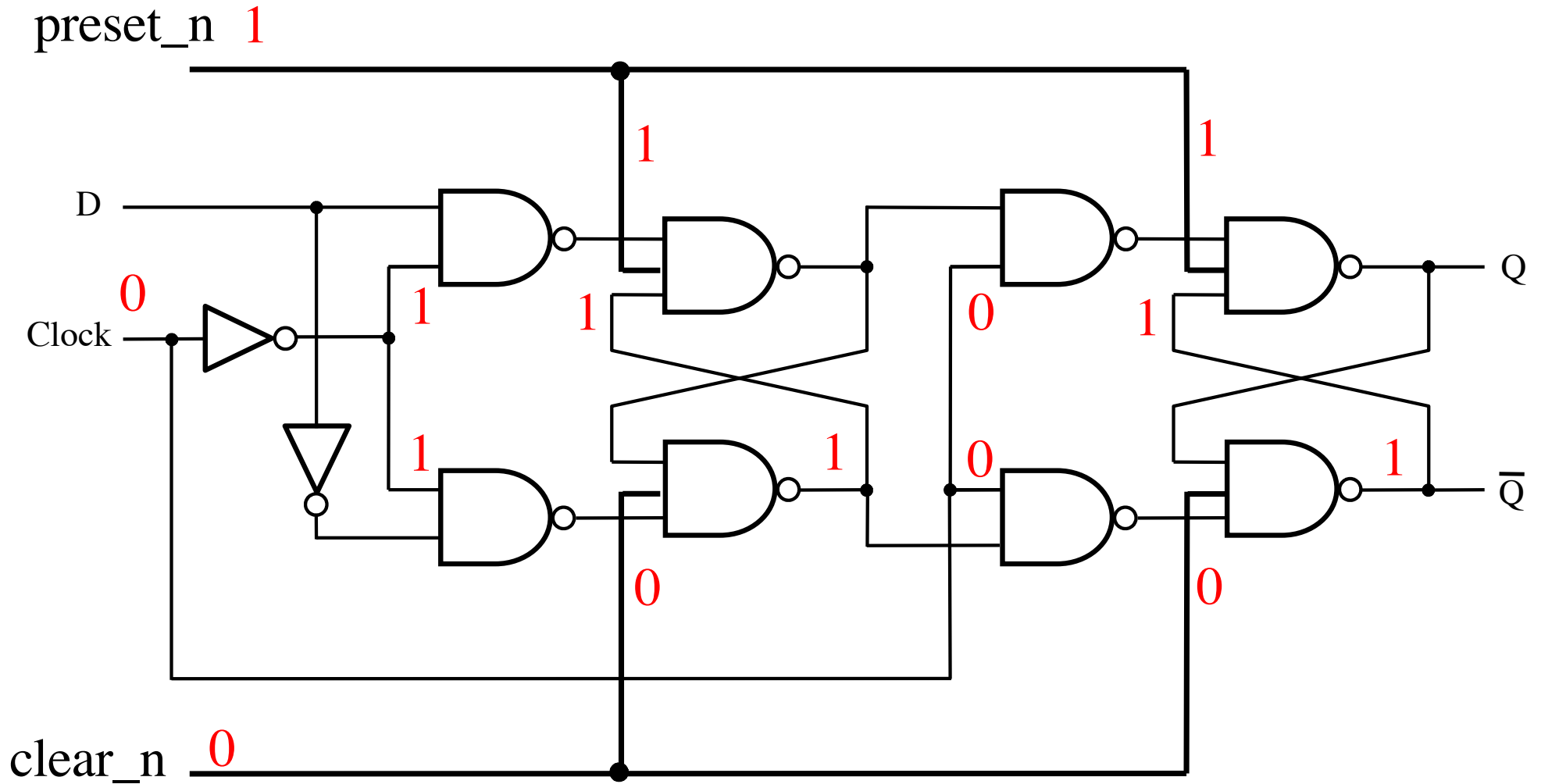
How does clear work?

Clock=0



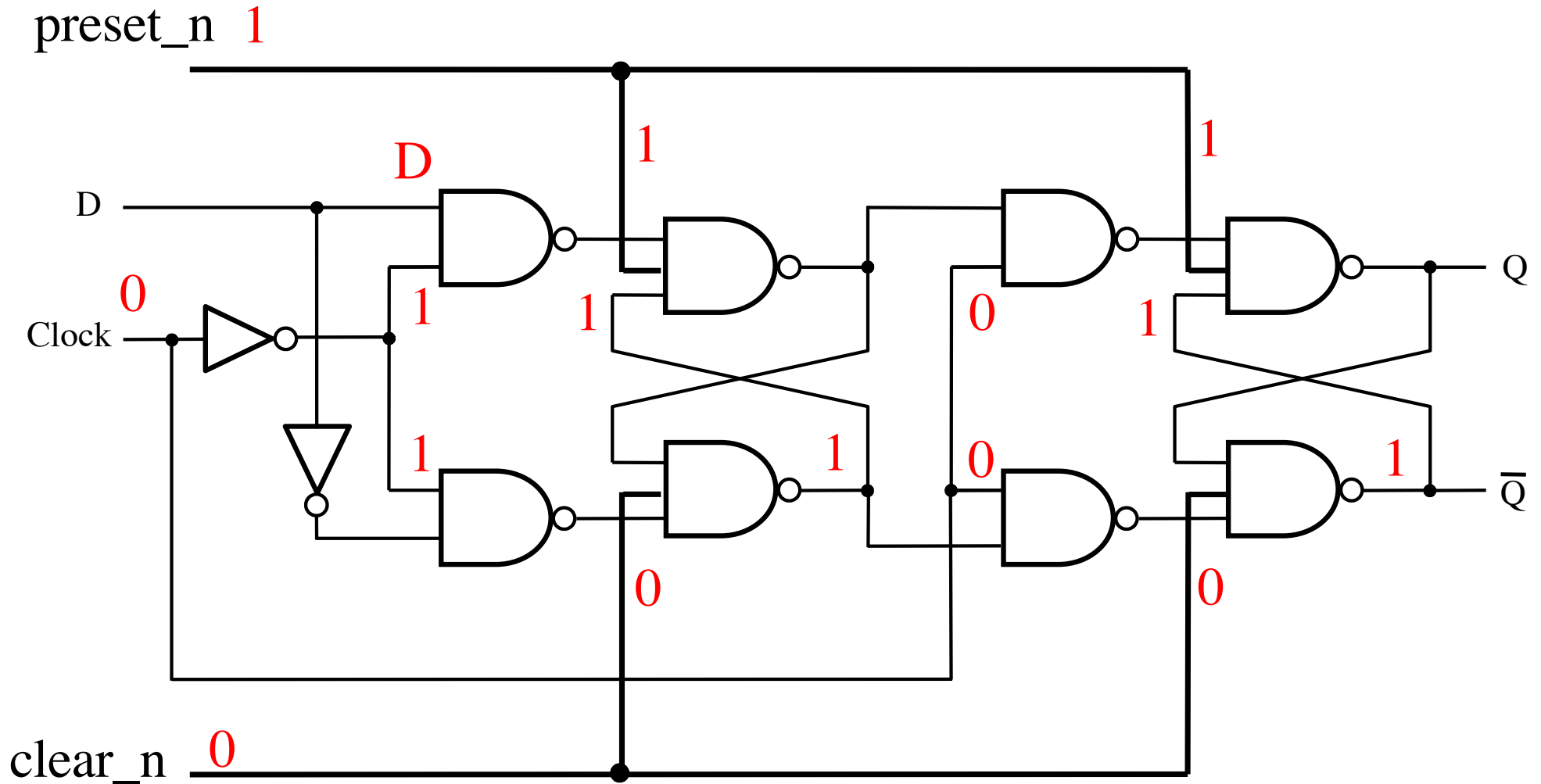
How does clear work?

Clock=0



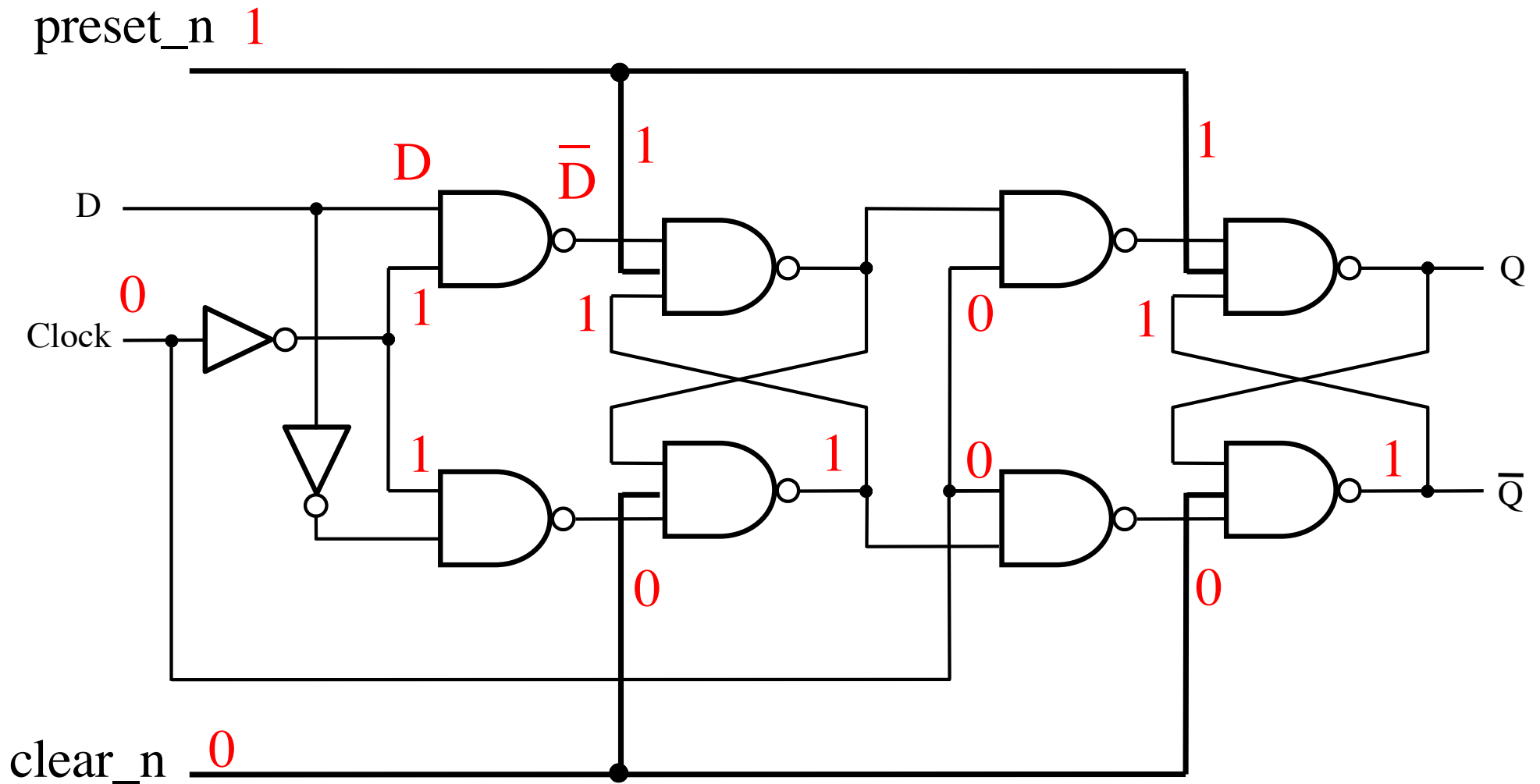
How does clear work?

Clock=0



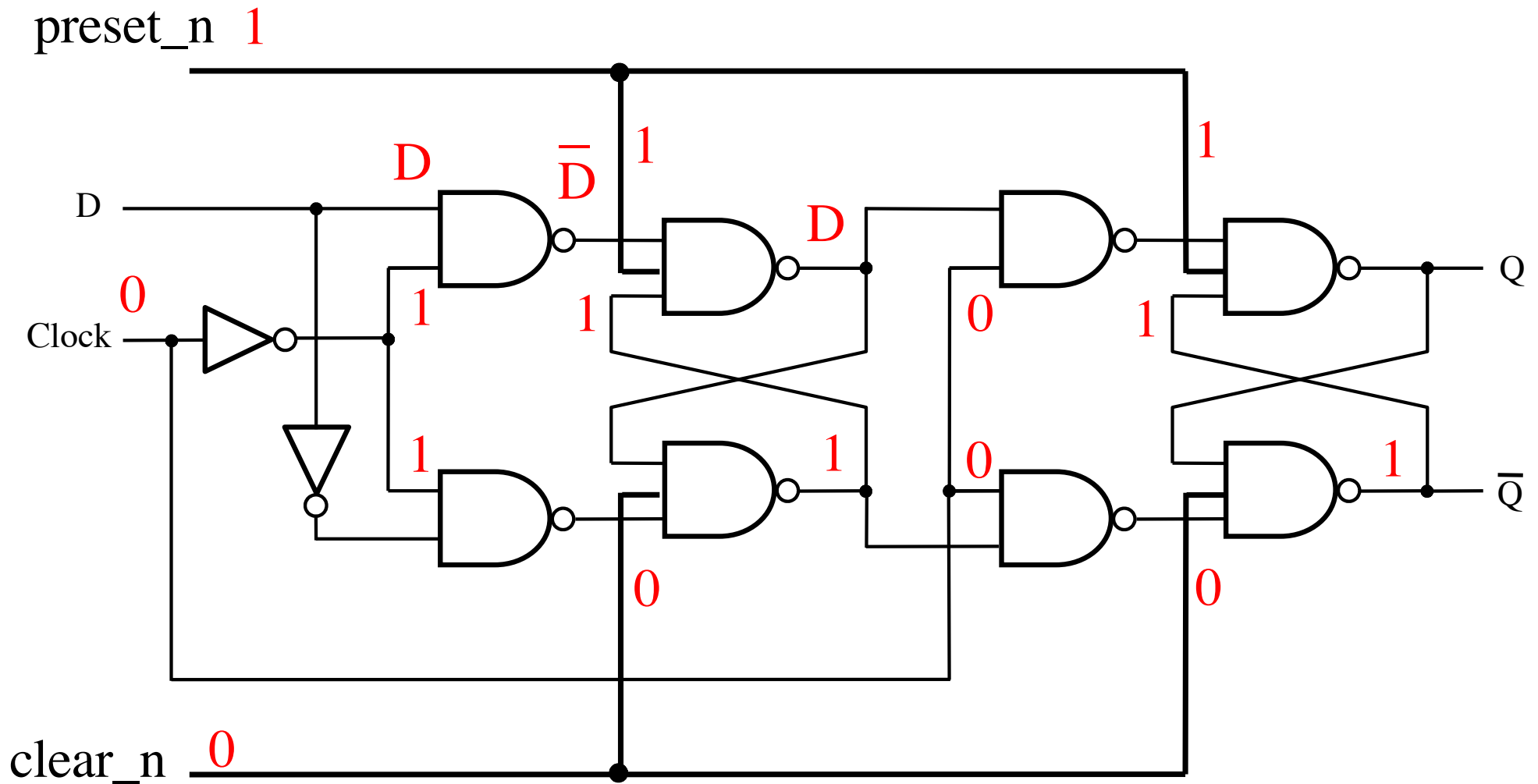
How does clear work?

Clock=0



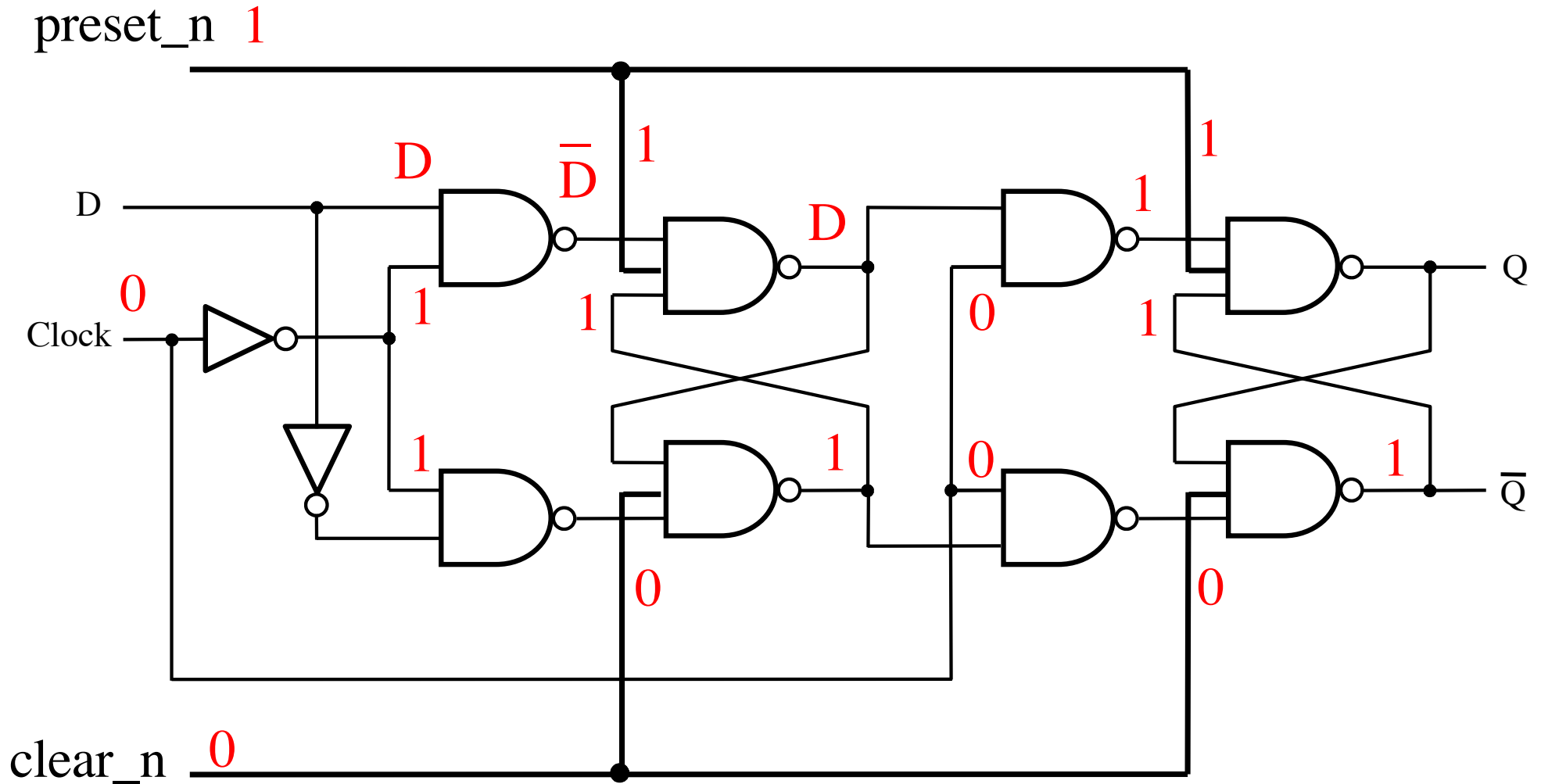
How does clear work?

Clock=0



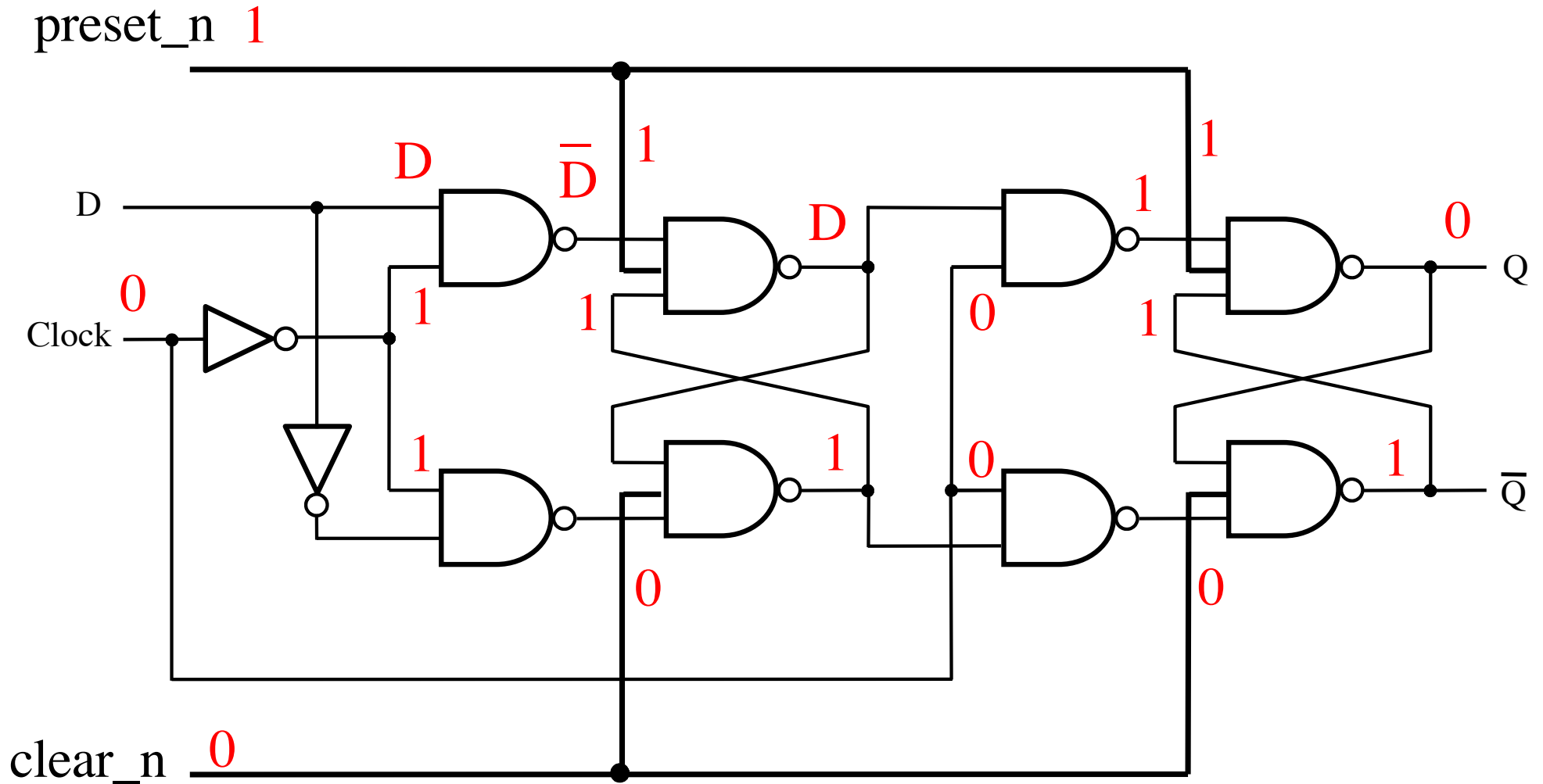
How does clear work?

Clock=0

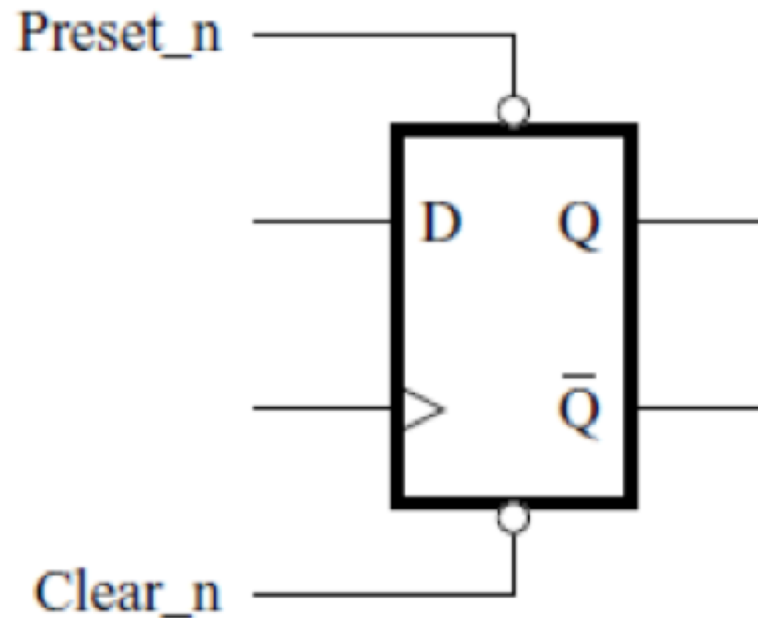


How does clear work?

Clock=0

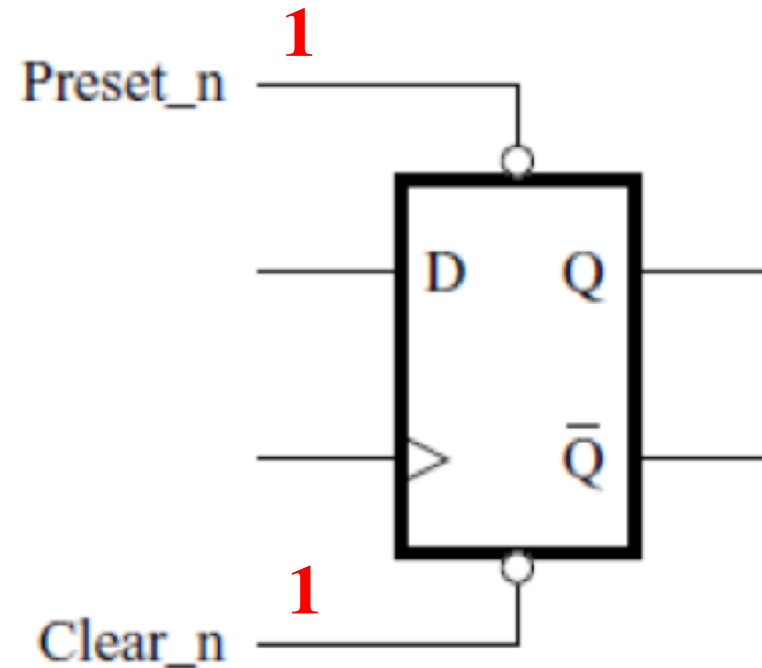


Positive-edge-triggered D flip-flop with asynchronous Clear and Preset



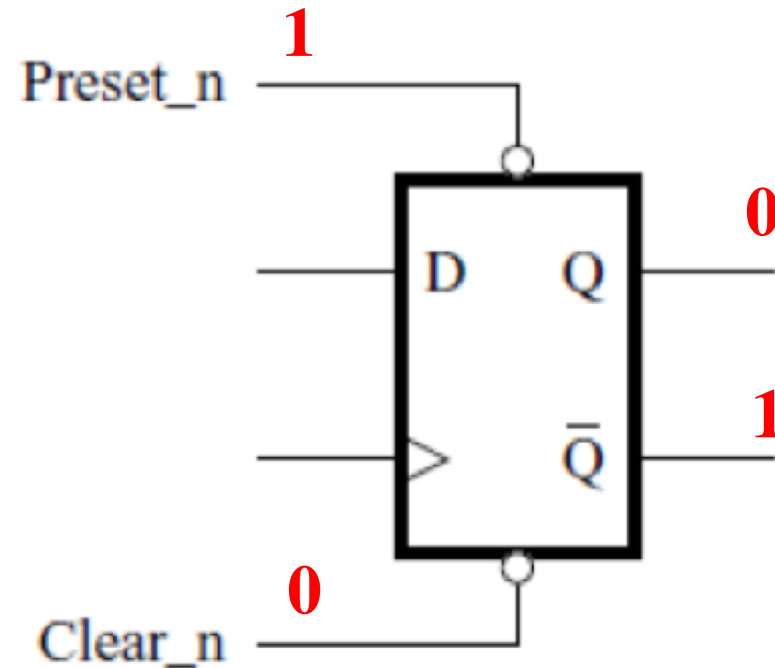
(b) Graphical symbol

For normal operation both must be set to 1



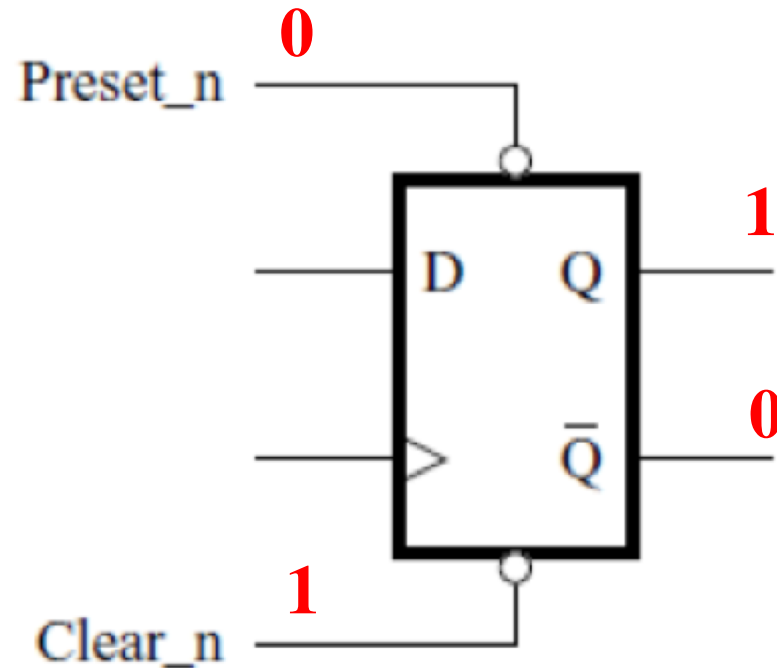
(b) Graphical symbol

A zero on clear_n drives the output Q to zero



(b) Graphical symbol

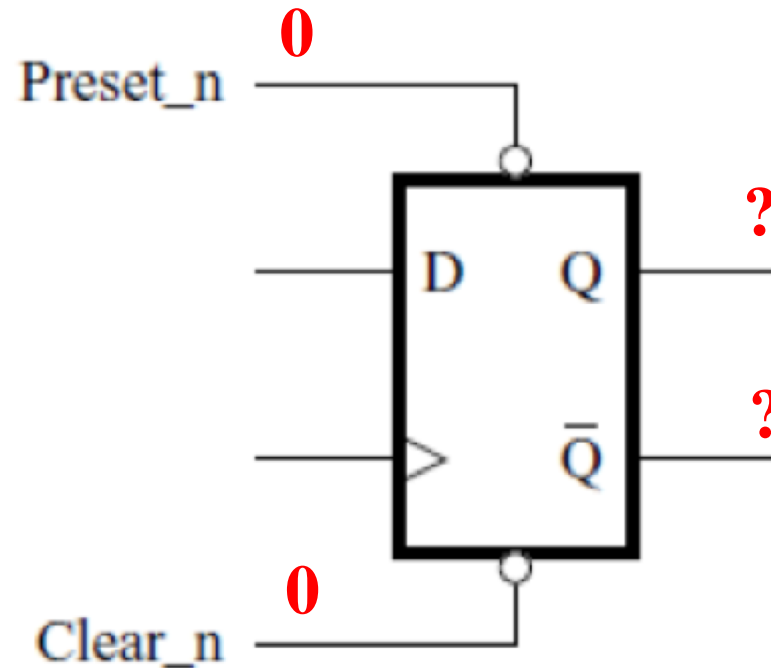
A zero on preset_n drives the output Q to one



(b) Graphical symbol

The output is indeterminate if both are zero

don't ever
use this one



(b) Graphical symbol

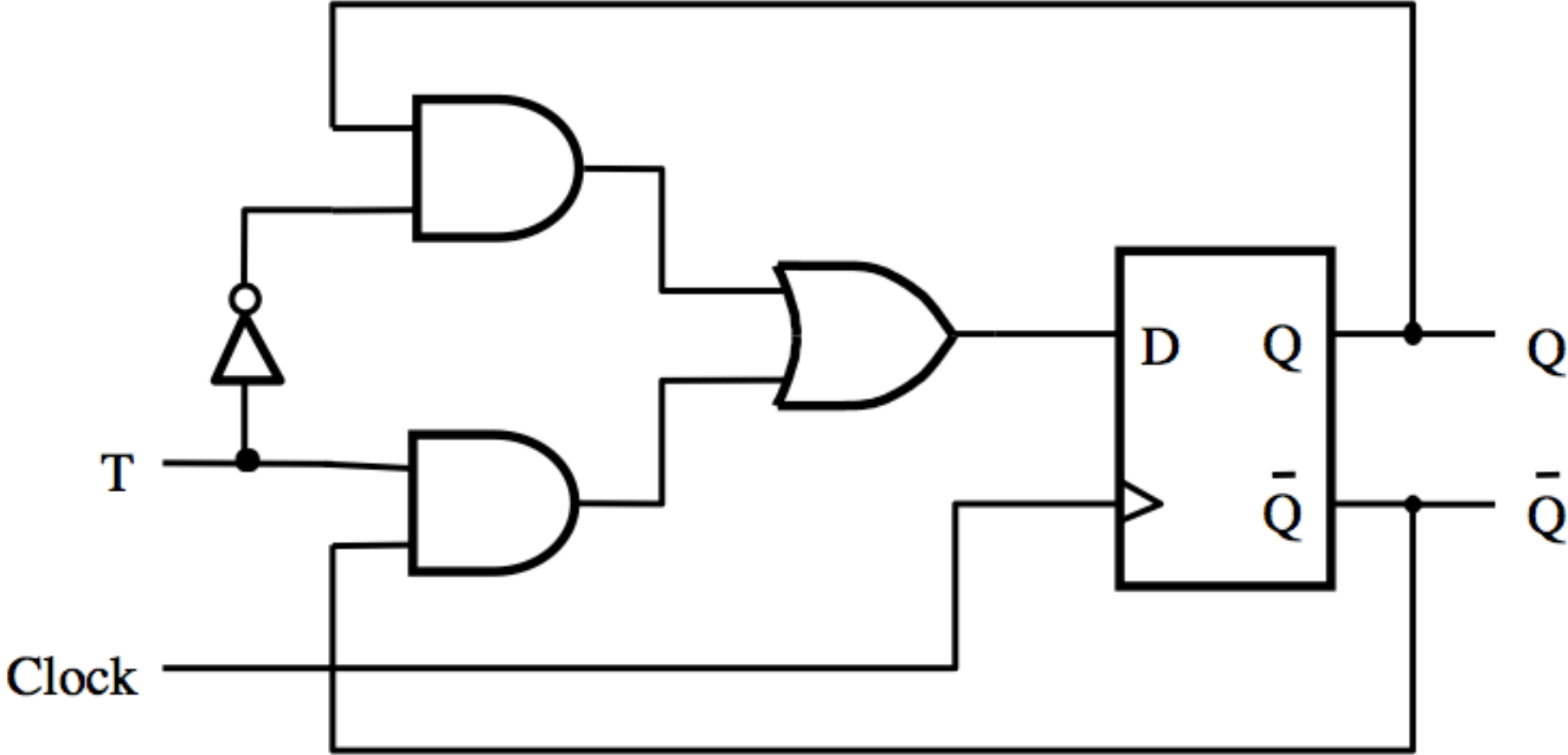
T Flip-Flop

Motivation

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

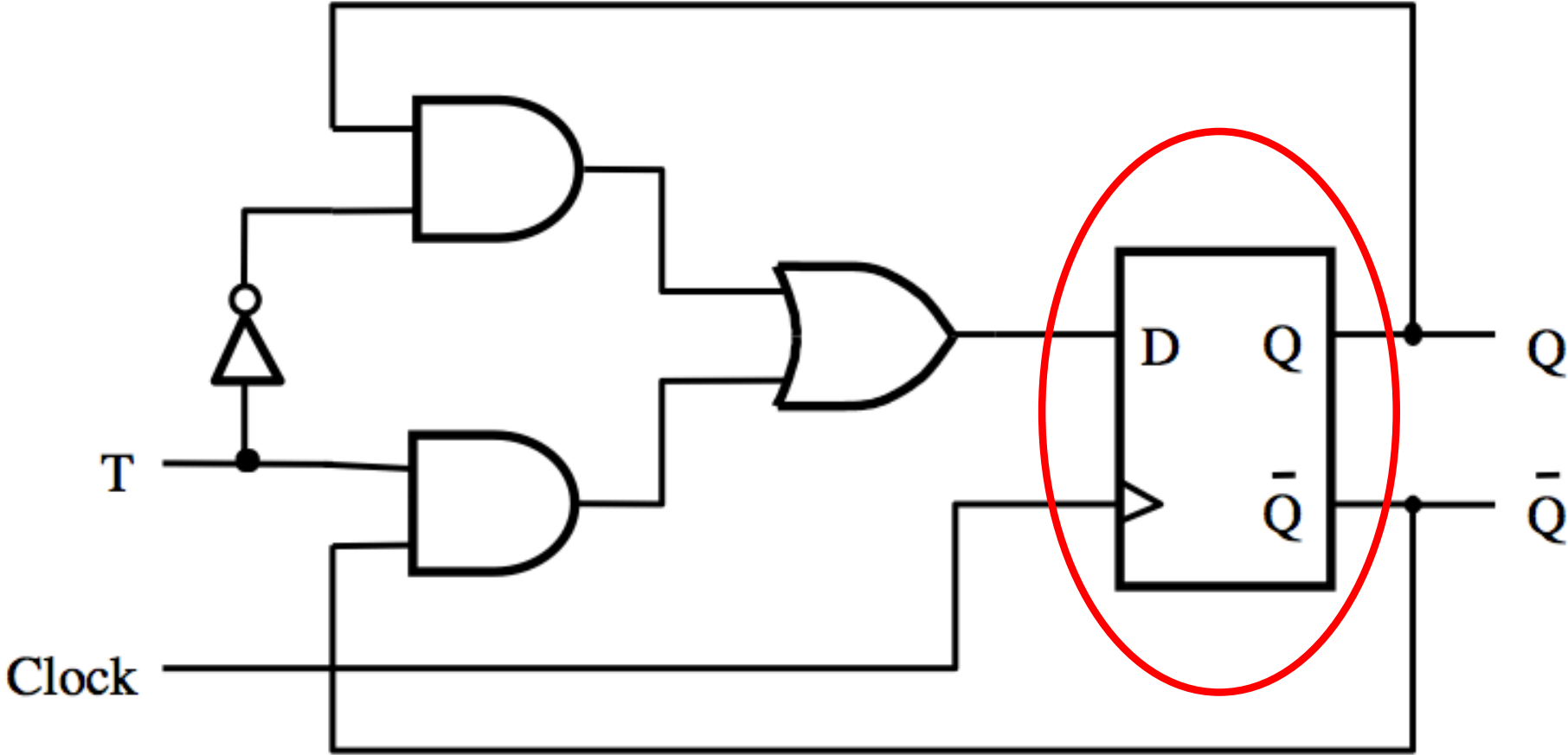
In this case, T stands for Toggle.

T Flip-Flop



[Figure 5.15a from the textbook]

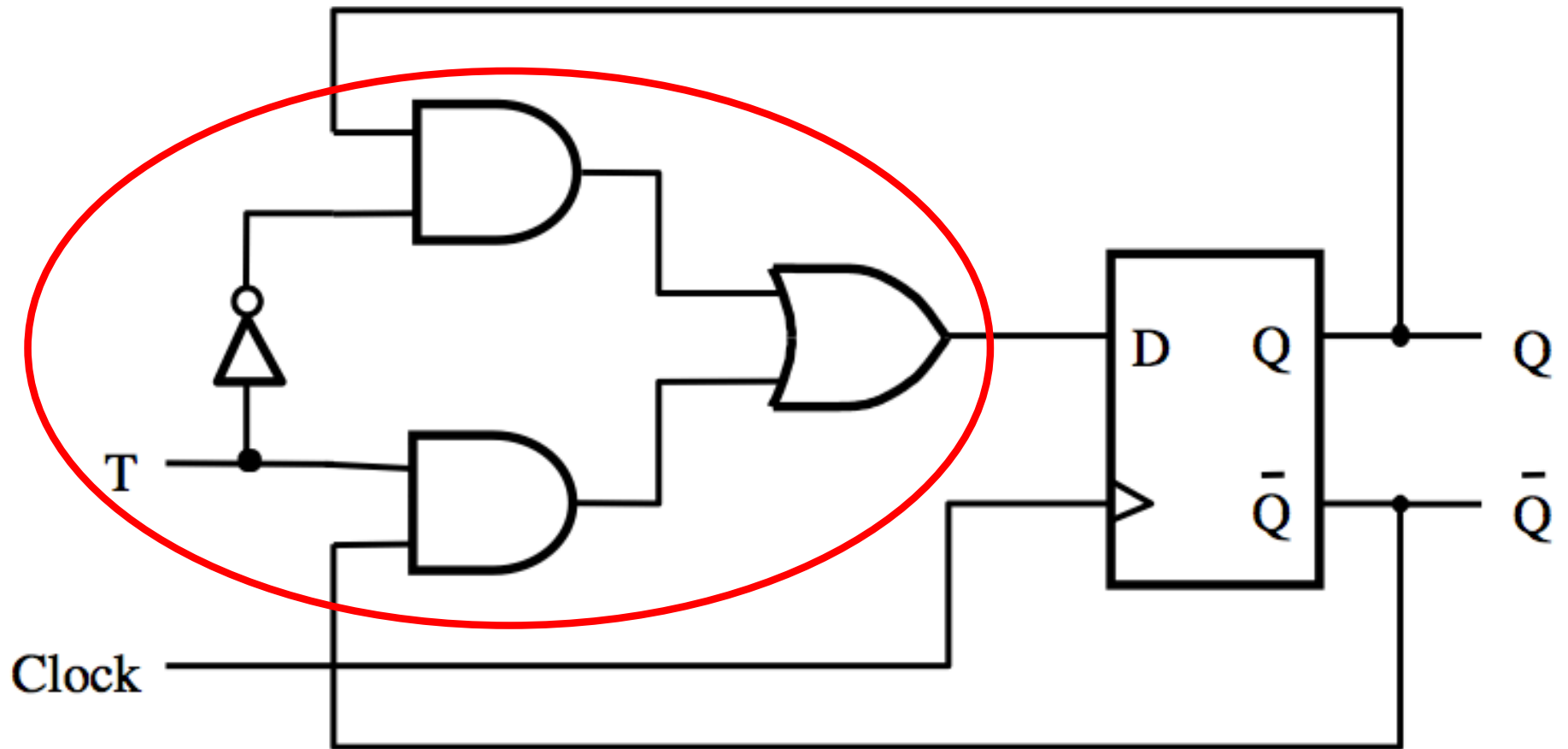
T Flip-Flop



Positive-edge-triggered
D Flip-Flop

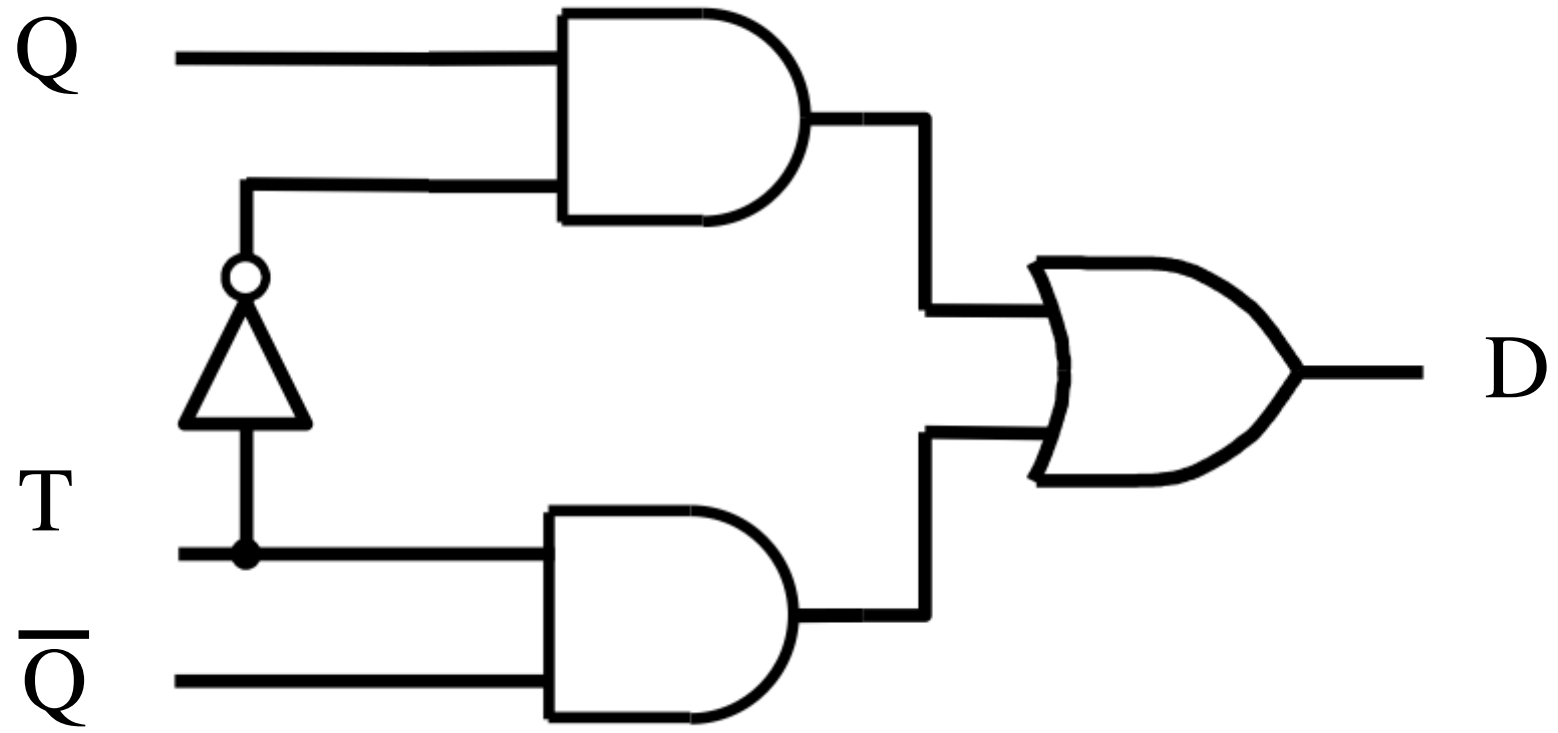
[Figure 5.15a from the textbook]

T Flip-Flop

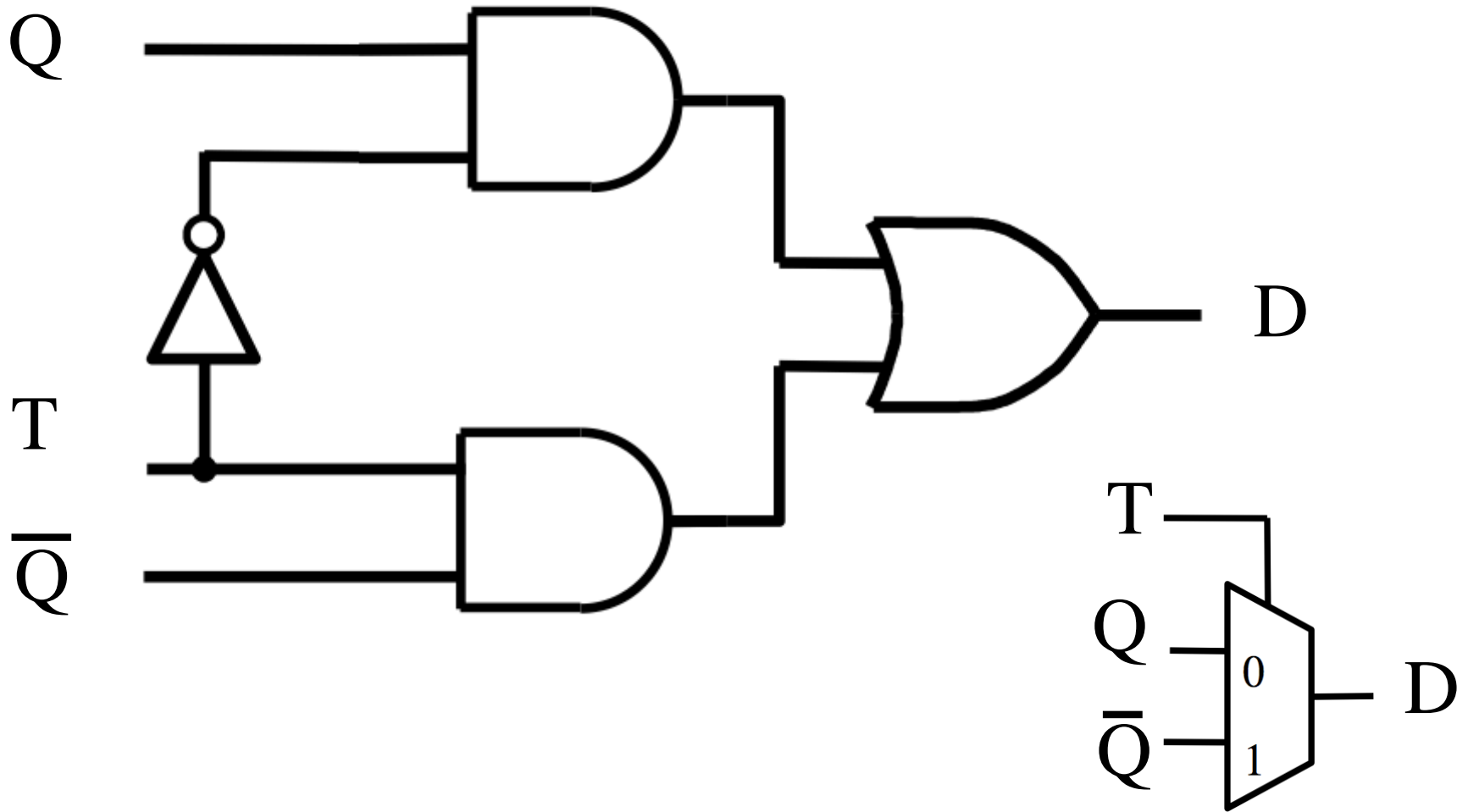


What is this?

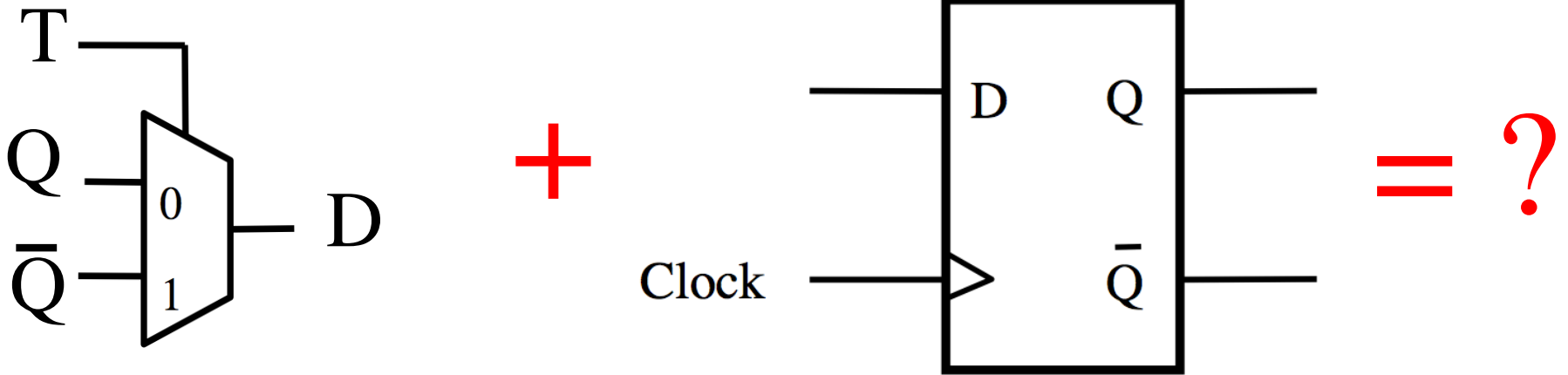
What is this?



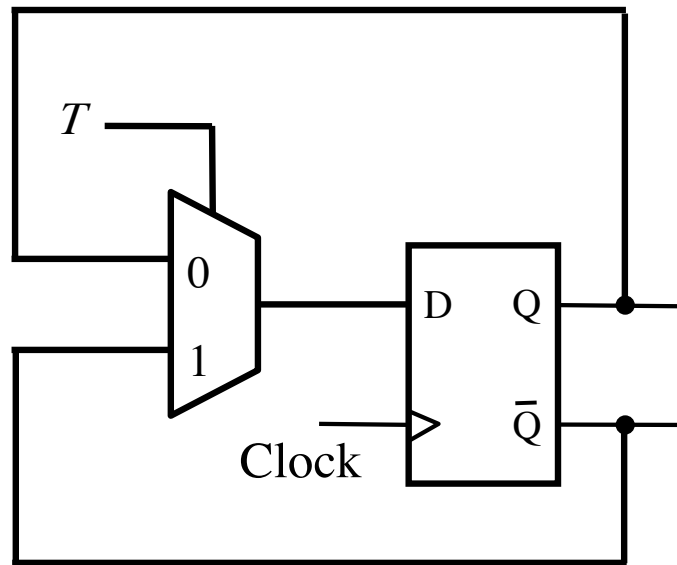
It is a 2-to-1 Multiplexer



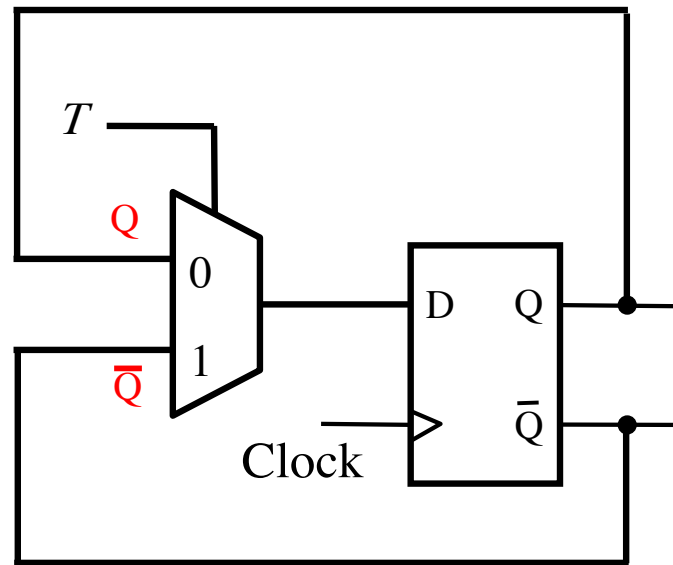
What is this?



It is a T Flip-Flop

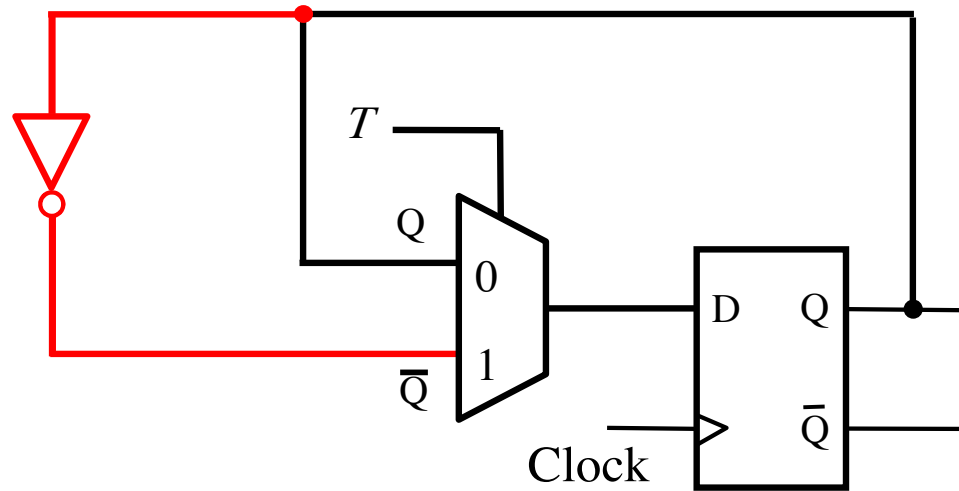


It is a T Flip-Flop

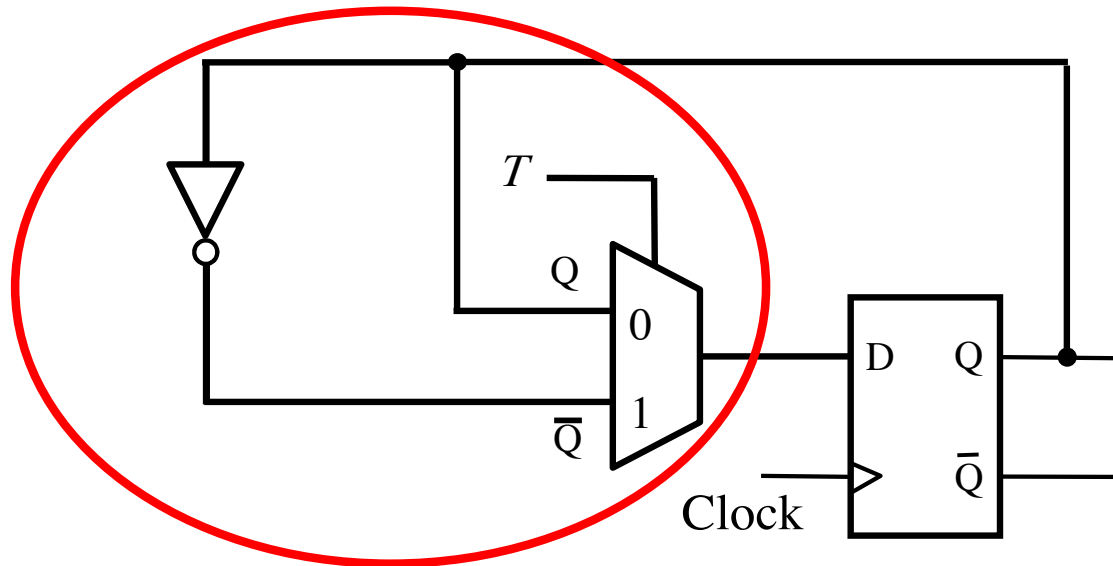


Note that the two inputs to the multiplexer are inverses of each other.

Another Way to Draw This

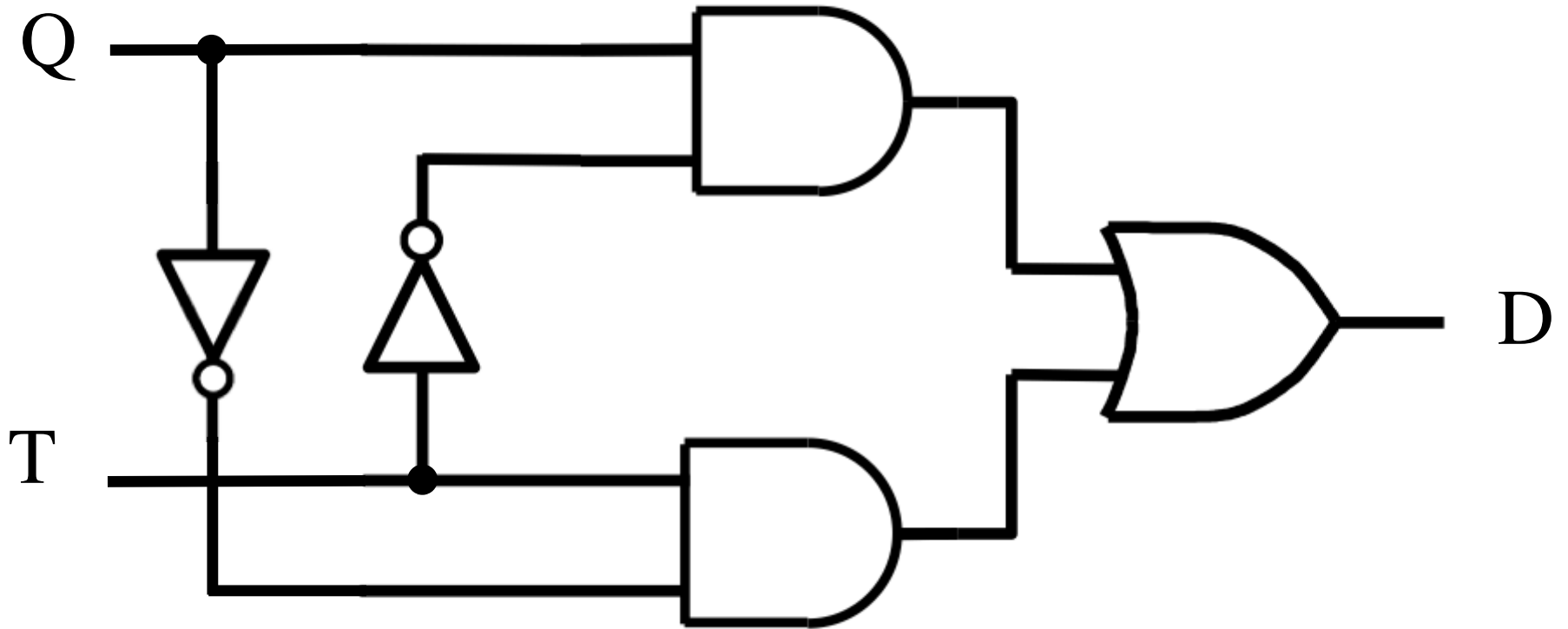


Another Way to Draw This

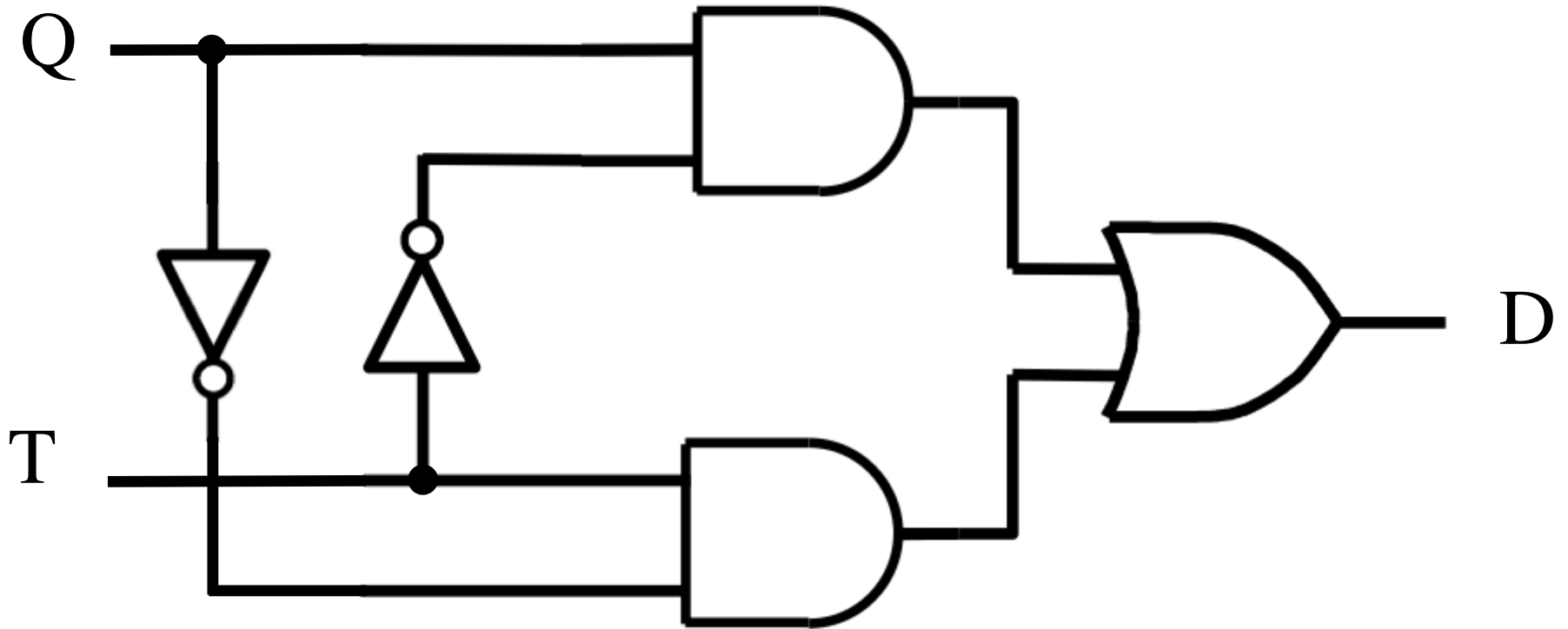


What is this?

What is this?

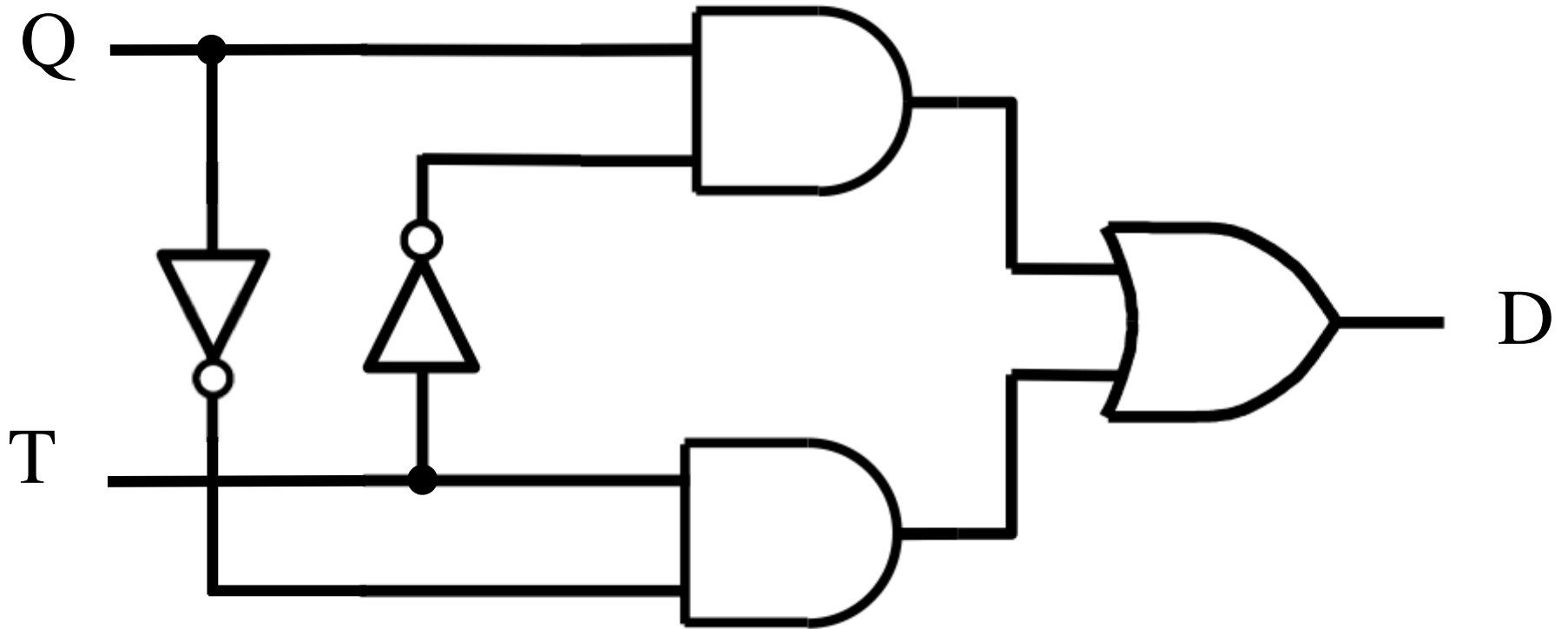


What is this?



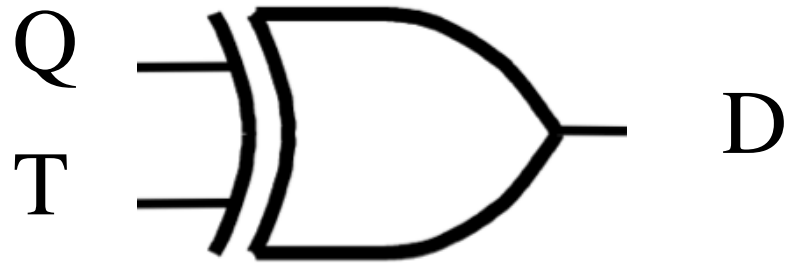
$$D = Q\bar{T} + \bar{Q}T$$

It is an XOR



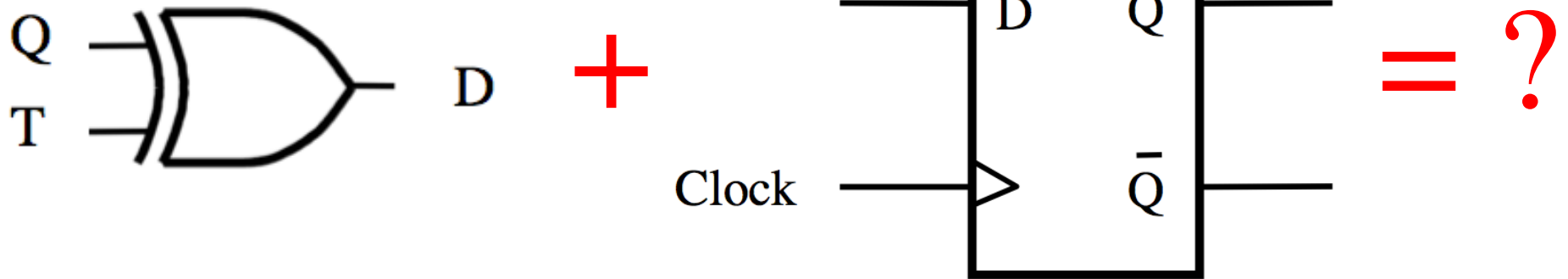
$$D = Q \oplus T$$

It is an XOR

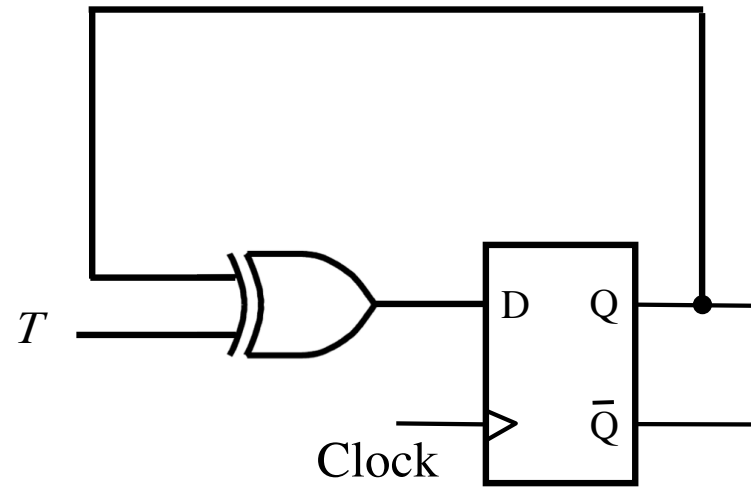


$$D = Q \oplus T$$

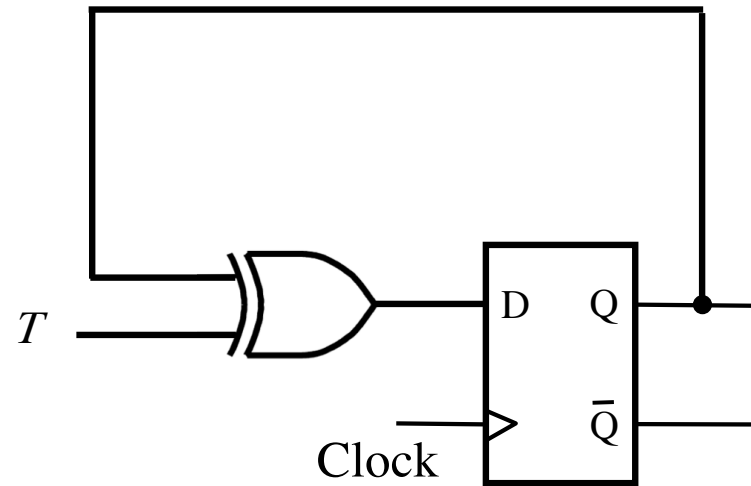
What is this?



It is a T Flip-Flop too

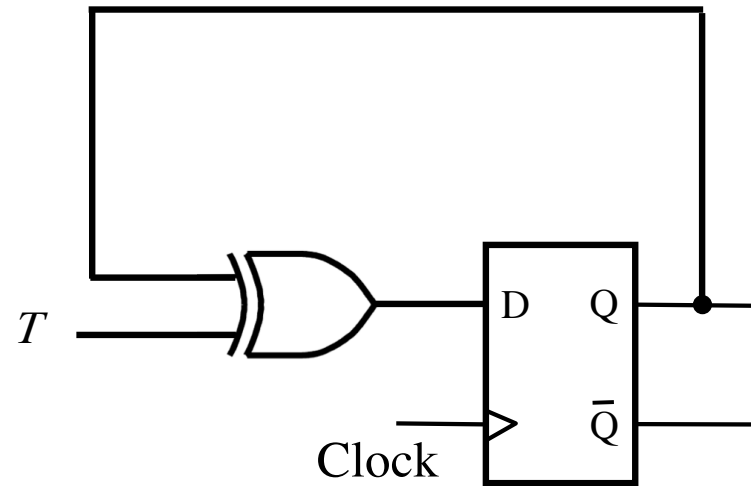


It is a T Flip-Flop too



T	Q	D
0	0	0
0	1	1
1	0	1
1	1	0

It is a T Flip-Flop too



T	Q	D
0	0	0
0	1	1
<hr/>		
1	0	1
1	1	0

Red annotations in the table: A red bracket groups the first two rows (0,0) and (0,1) with a red Q label. Another red bracket groups the last two rows (1,0) and (1,1) with a red \bar{Q} label. A red horizontal line is drawn under the first two rows.

T Flip-Flop

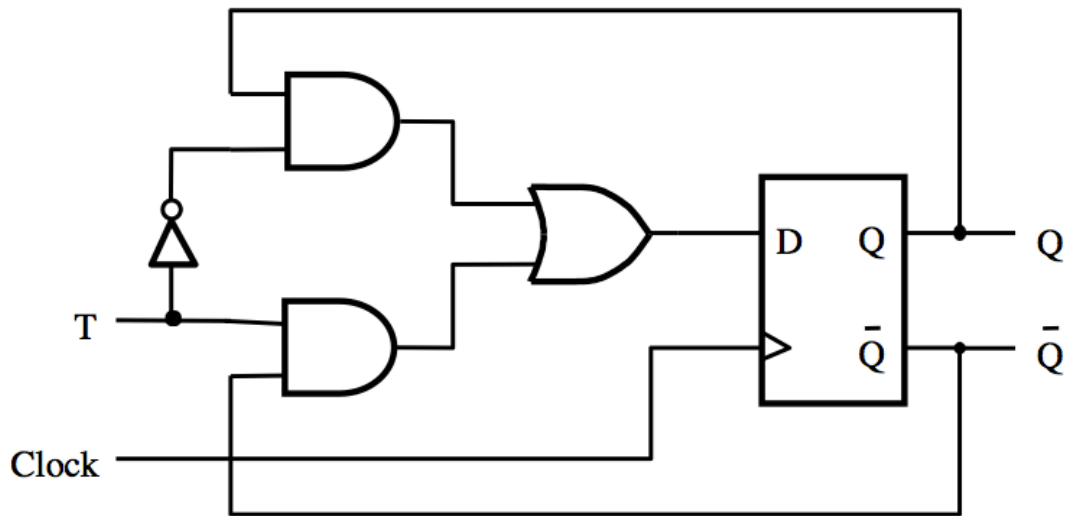
(how it works)

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.

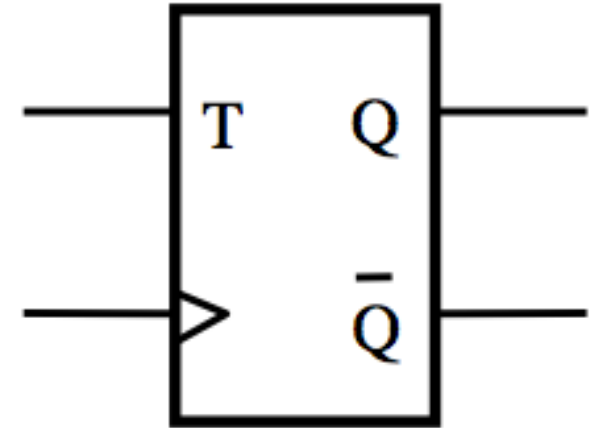
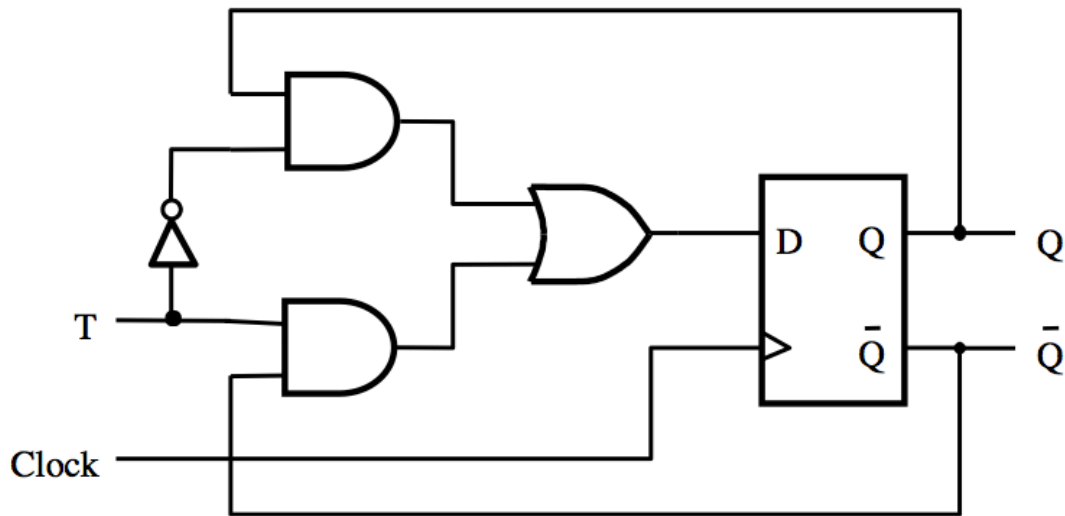
T Flip-Flop (circuit and truth table)



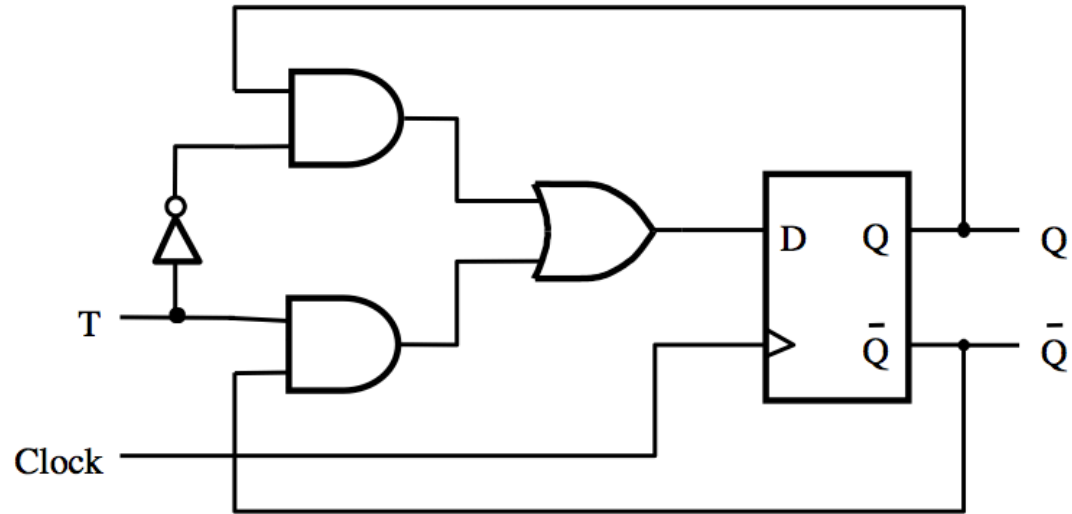
T	$Q(t+1)$
0	$Q(t)$
1	$\bar{Q}(t)$


T Flip-Flop

(circuit and graphical symbol)



T Flip-Flop (Timing Diagram)

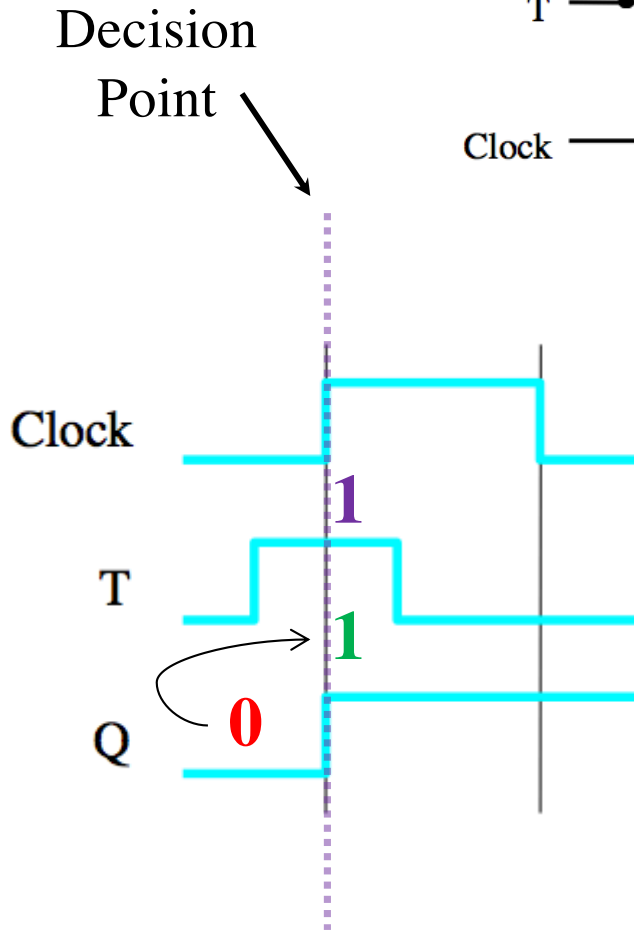
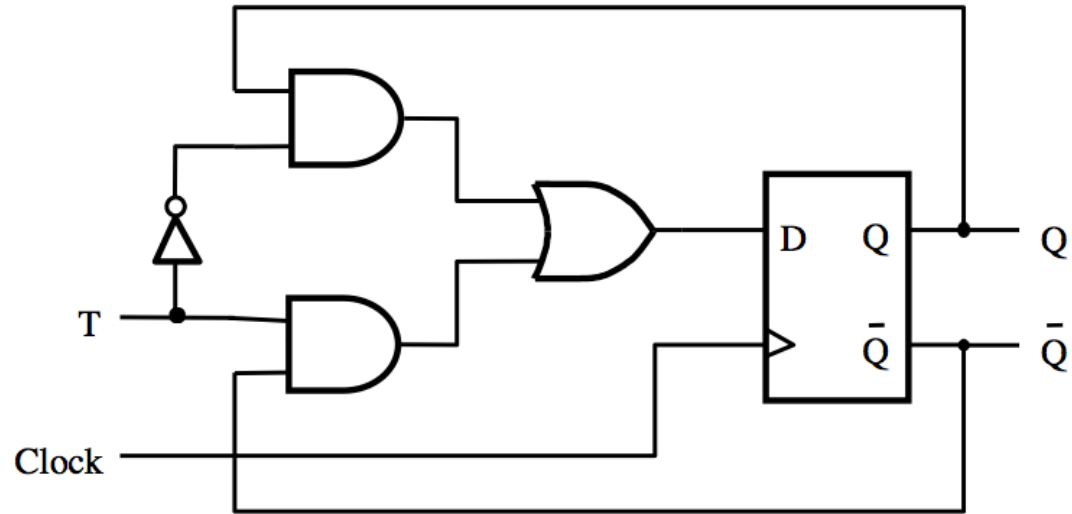


Clock 

T 

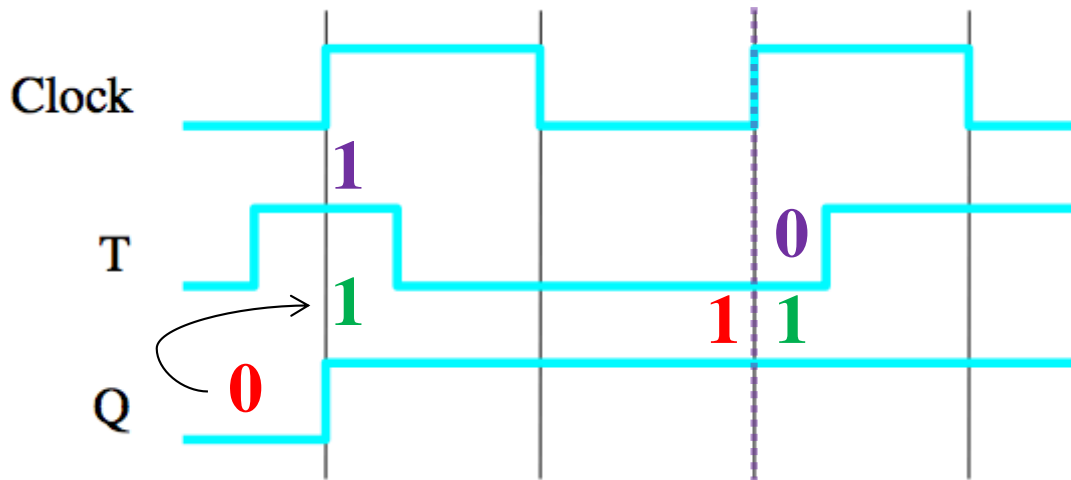
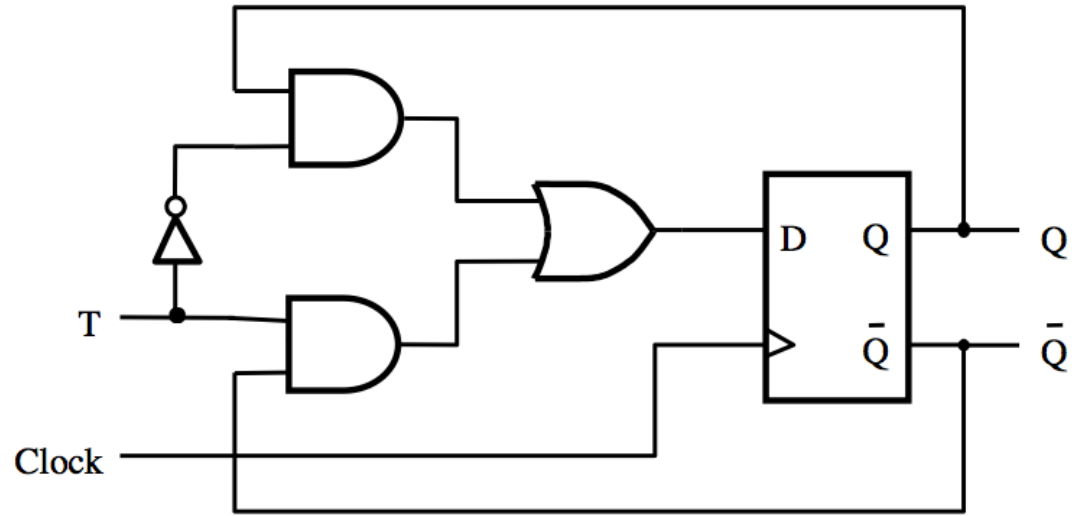
Q 

T Flip-Flop (Timing Diagram)



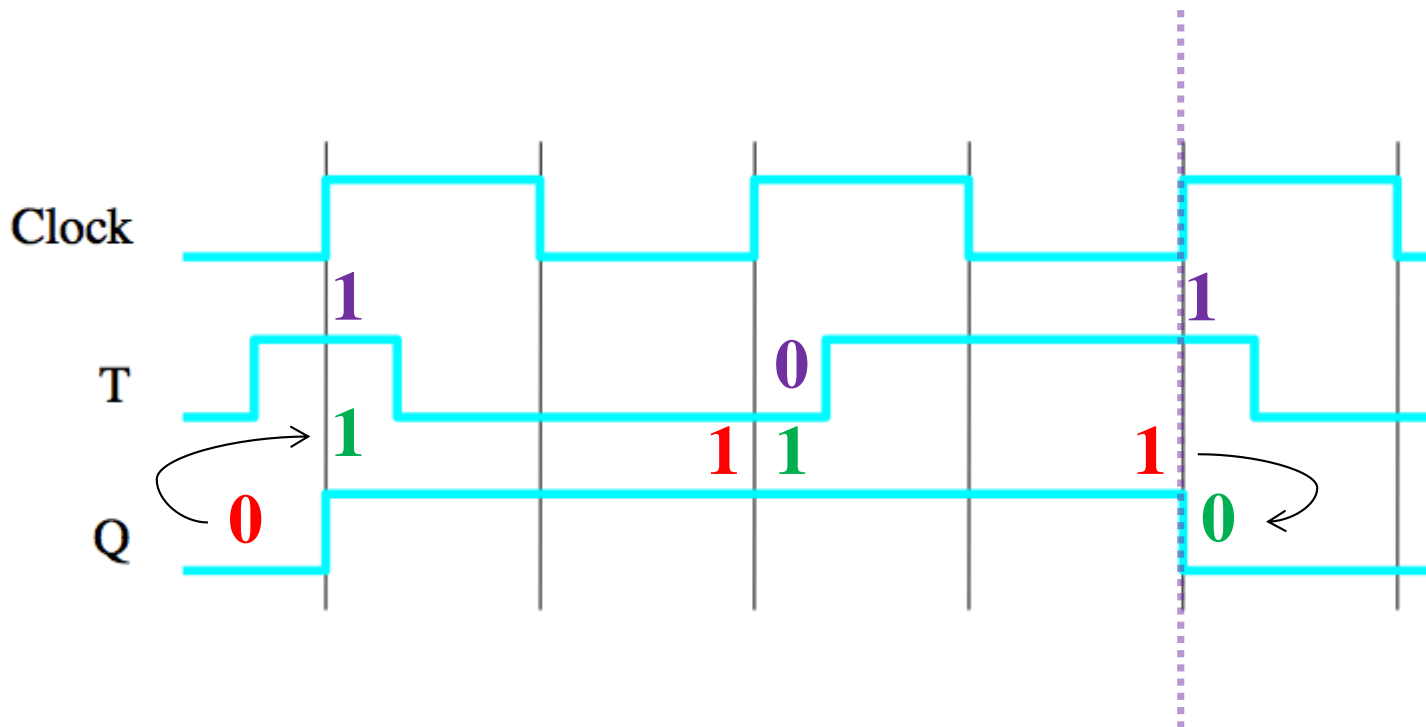
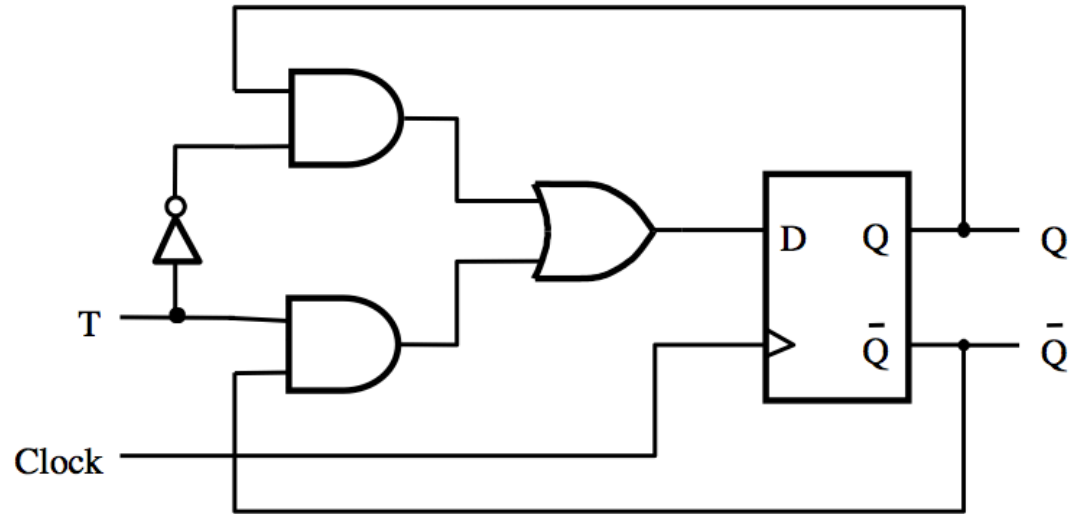
[Figure 5.15d from the textbook]

T Flip-Flop (Timing Diagram)



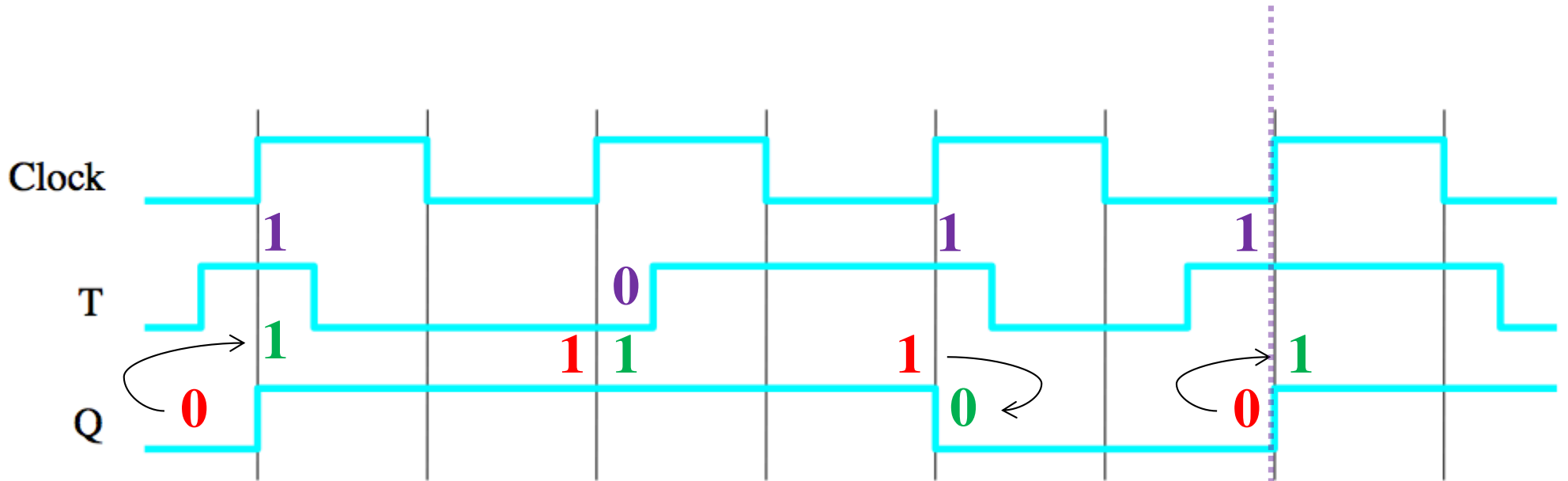
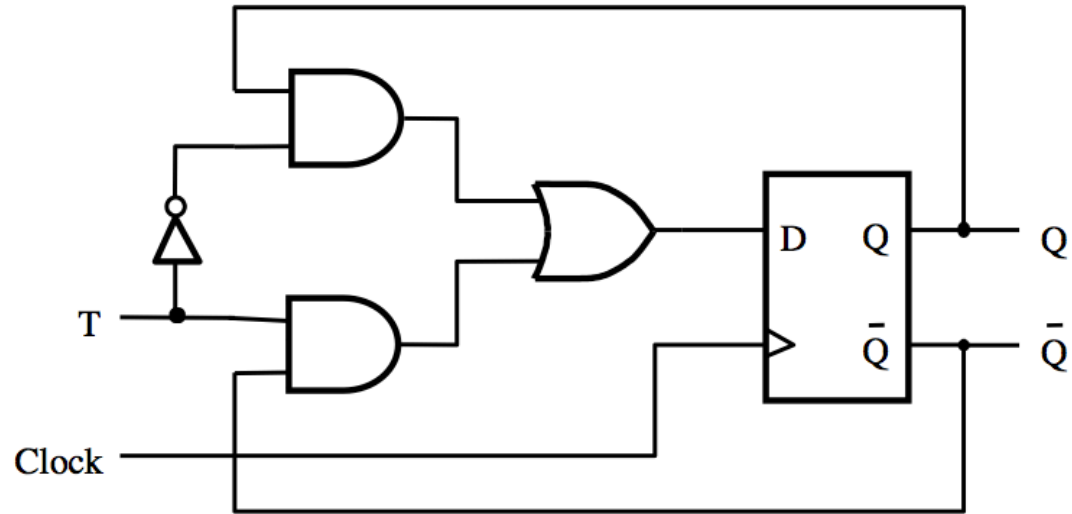
[Figure 5.15d from the textbook]

T Flip-Flop (Timing Diagram)



[Figure 5.15d from the textbook]

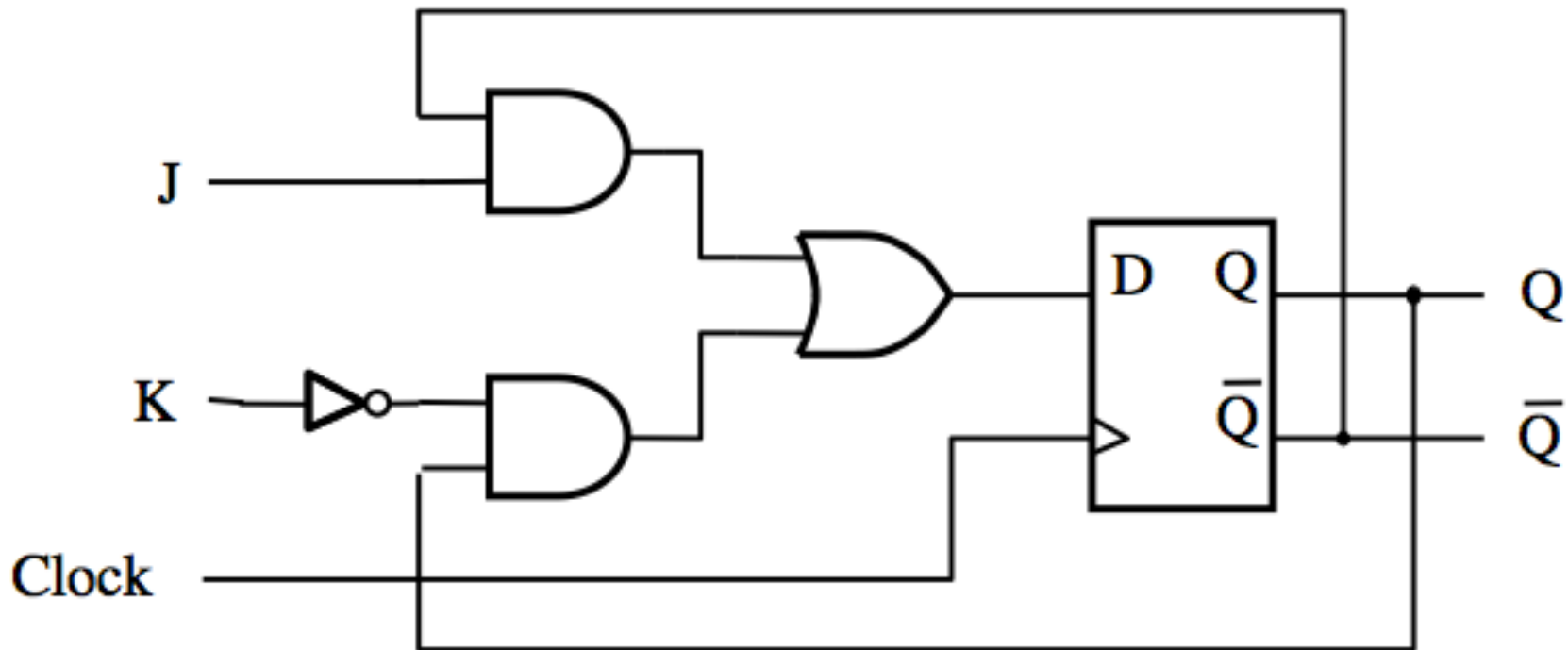
T Flip-Flop (Timing Diagram)



[Figure 5.15d from the textbook]

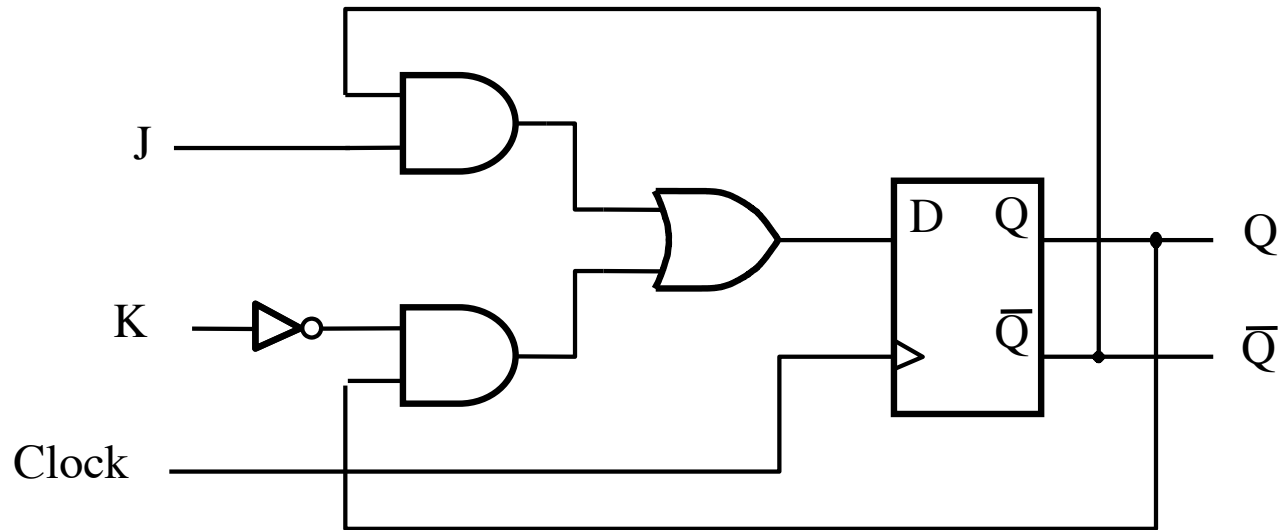
JK Flip-Flop

JK Flip-Flop



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

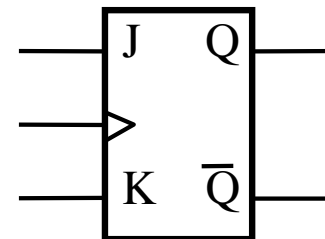
JK Flip-Flop



(a) Circuit

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

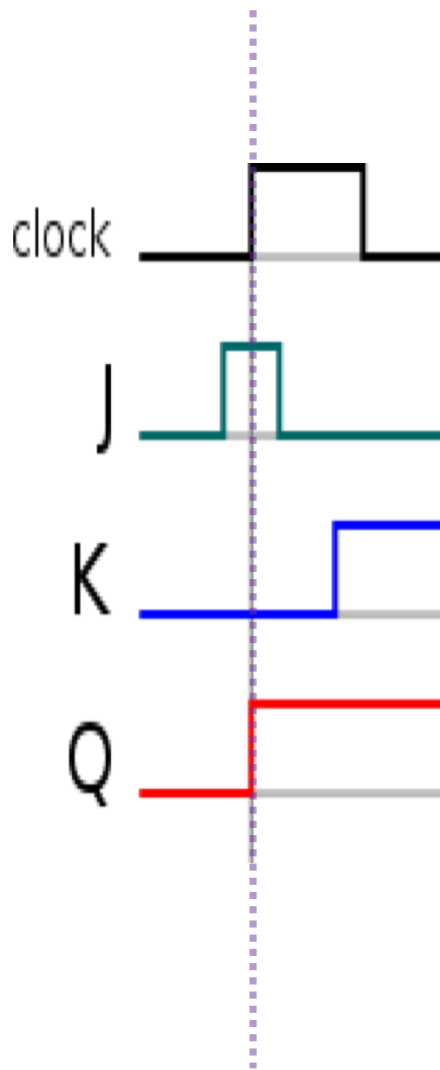
If $J=1$ and $K=0$ it sets the output Q to 1

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

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

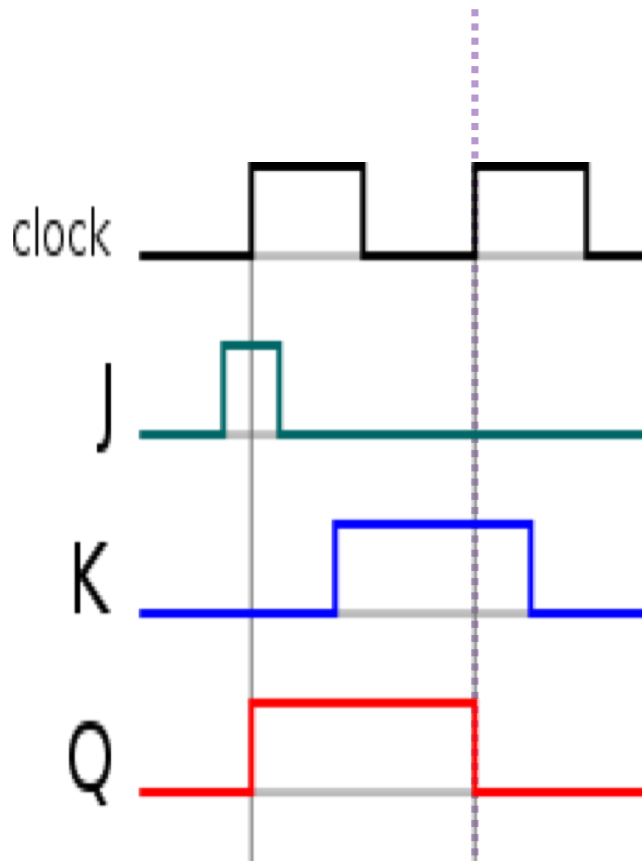
If $J=K$ then it behaves like a T flip-flop

JK Flip-Flop (timing diagram)



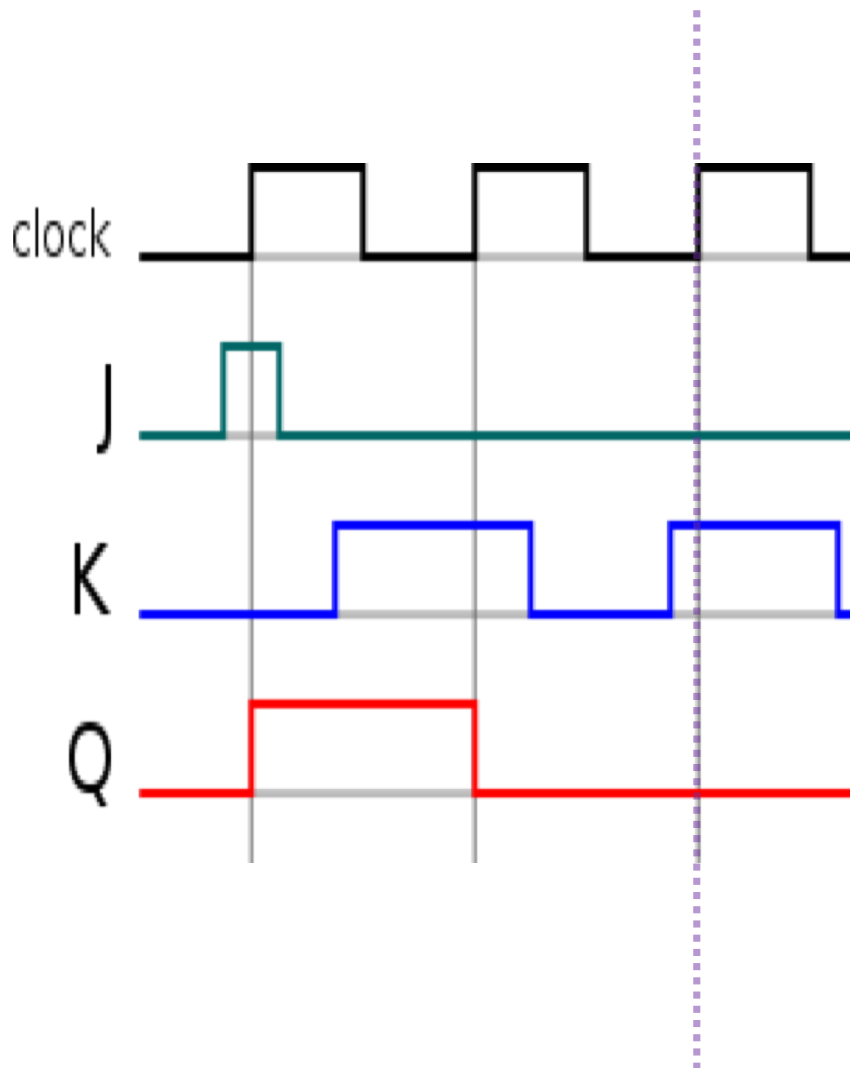
J	K	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$\overline{Q}(t)$

JK Flip-Flop (timing diagram)



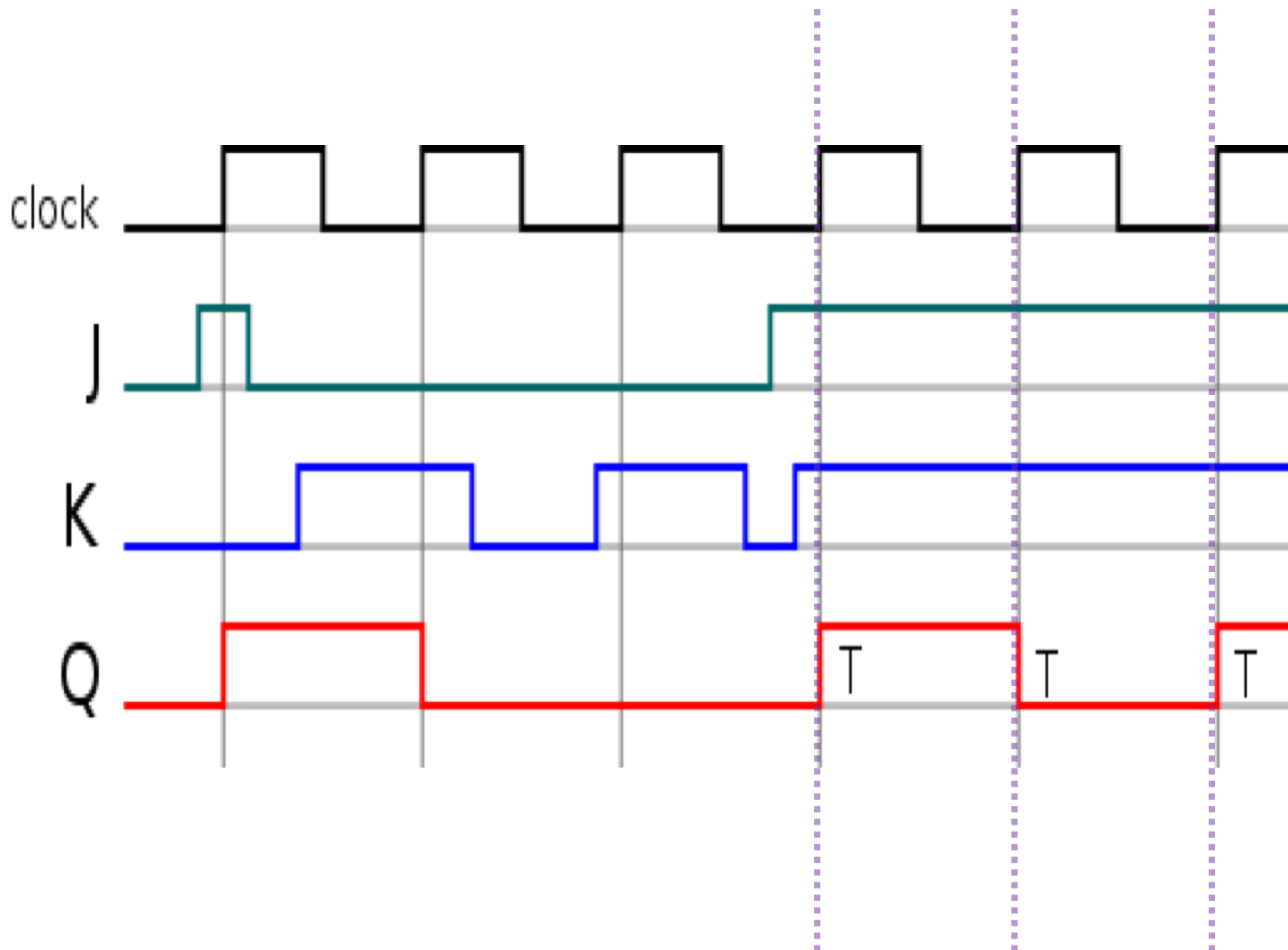
J	K	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$\overline{Q}(t)$

JK Flip-Flop (timing diagram)



J	K	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$\bar{Q}(t)$

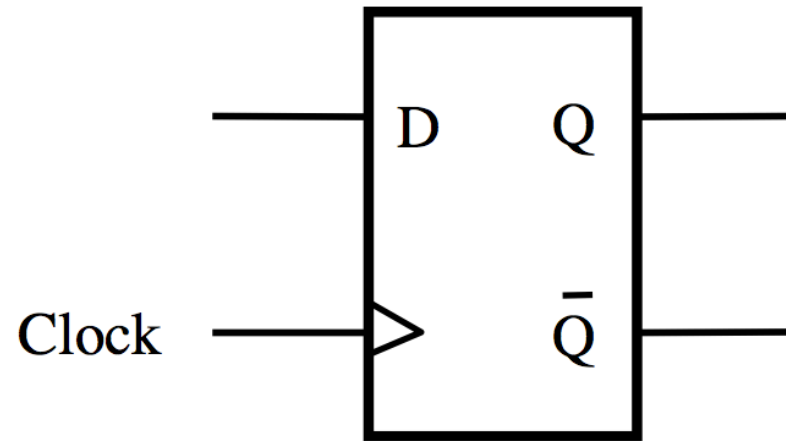
JK Flip-Flop (timing diagram)



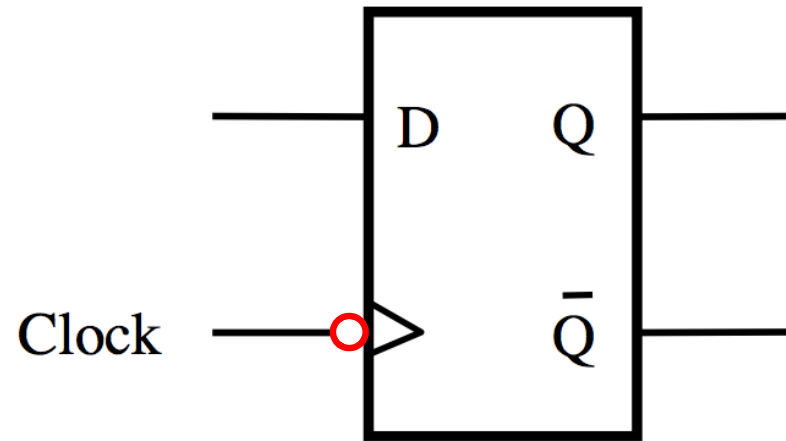
J	K	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$\bar{Q}(t)$

Complete Wiring Diagrams

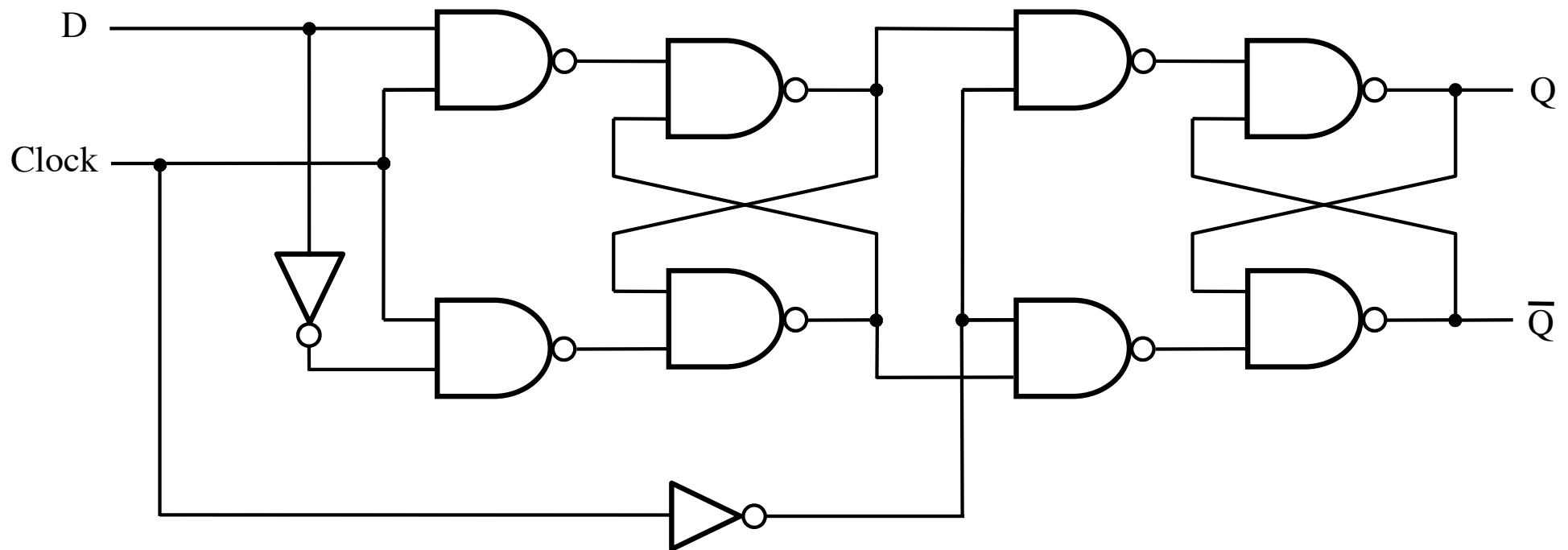
Positive-Edge-Triggered D Flip-Flop



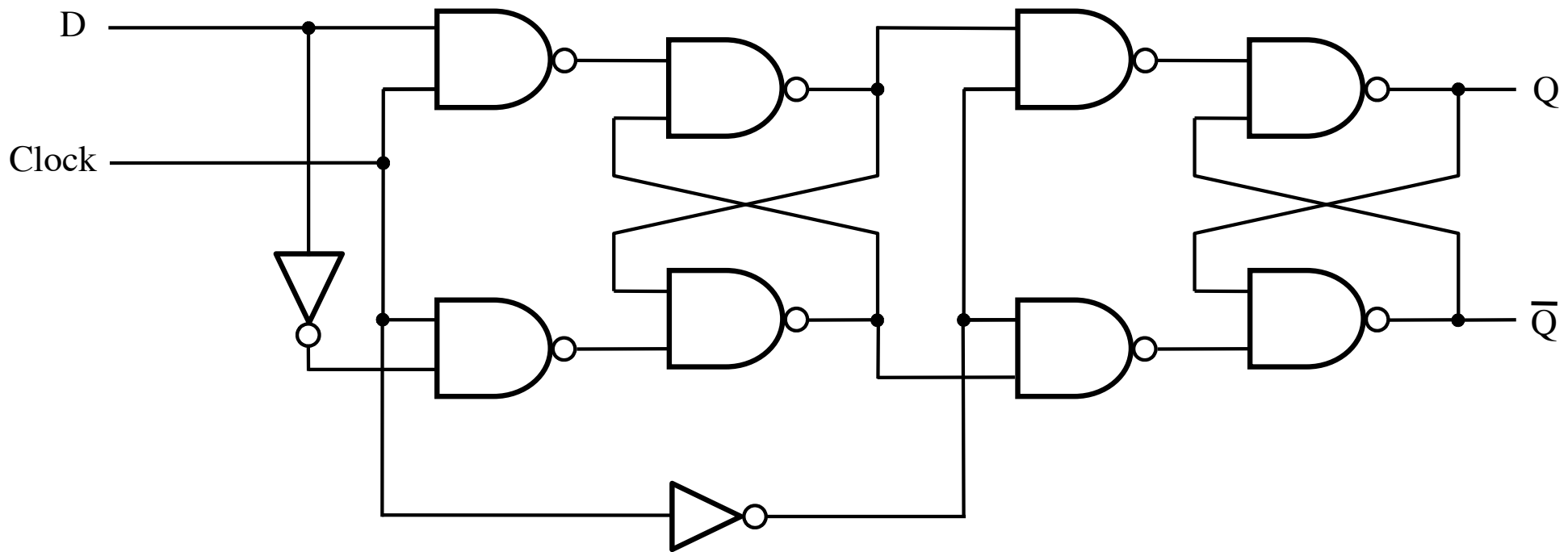
Negative-Edge-Triggered D Flip-Flop



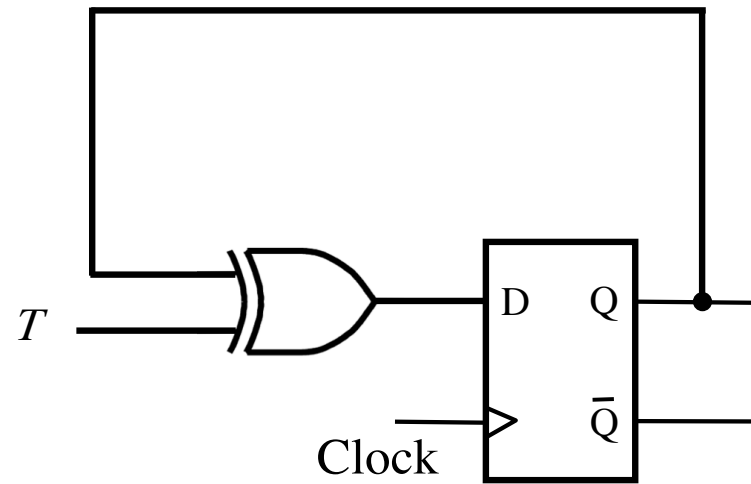
The Complete Wiring Diagram for a **Negative**-Edge-Triggered D Flip-Flop



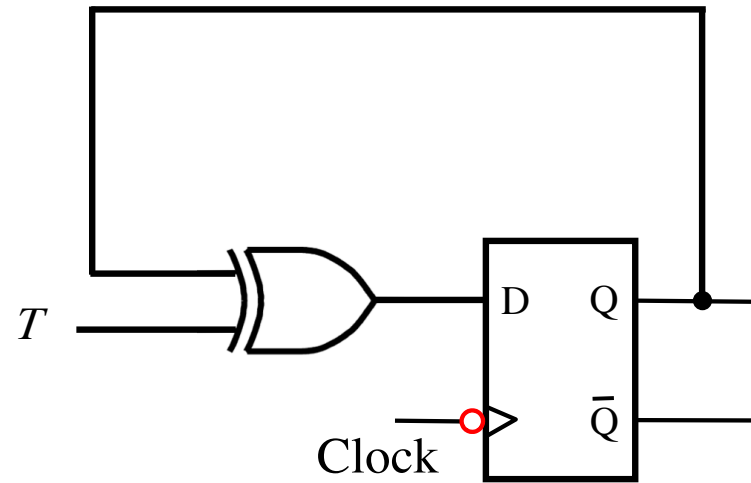
The Complete Wiring Diagram for a **Negative**-Edge-Triggered D Flip-Flop



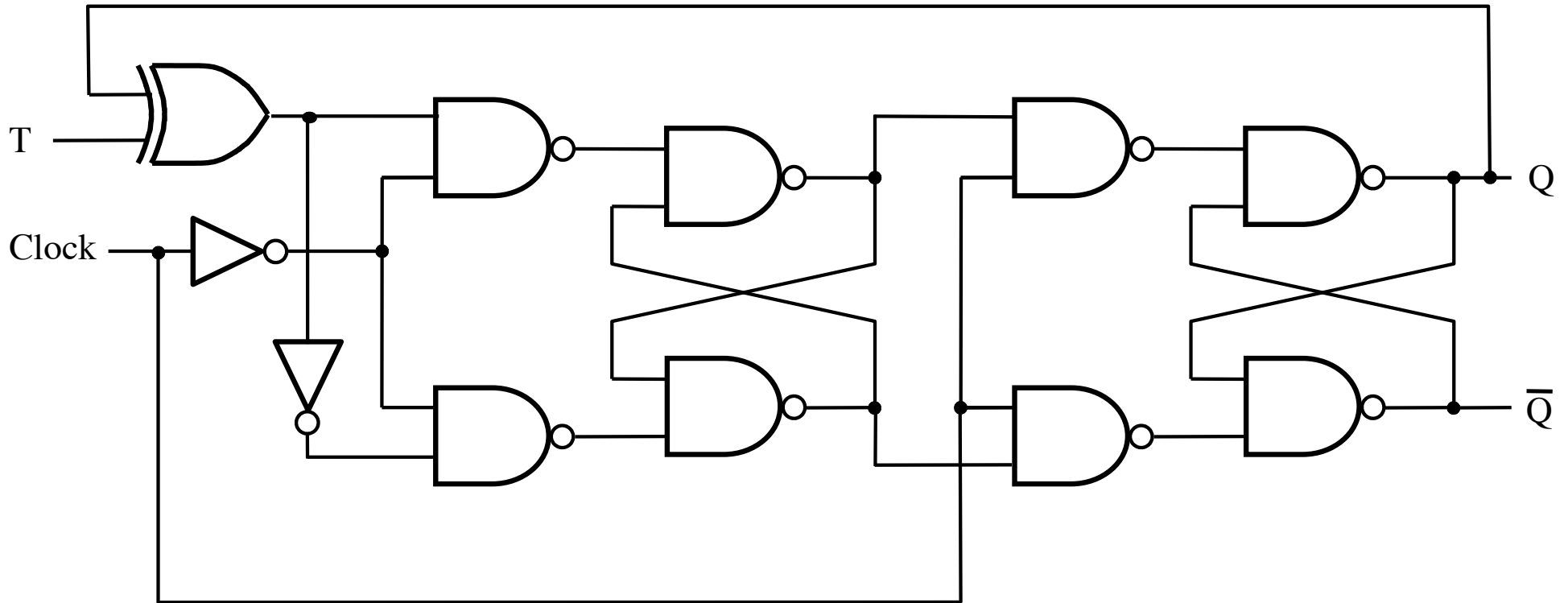
Positive-Edge-Triggered T Flip-Flop



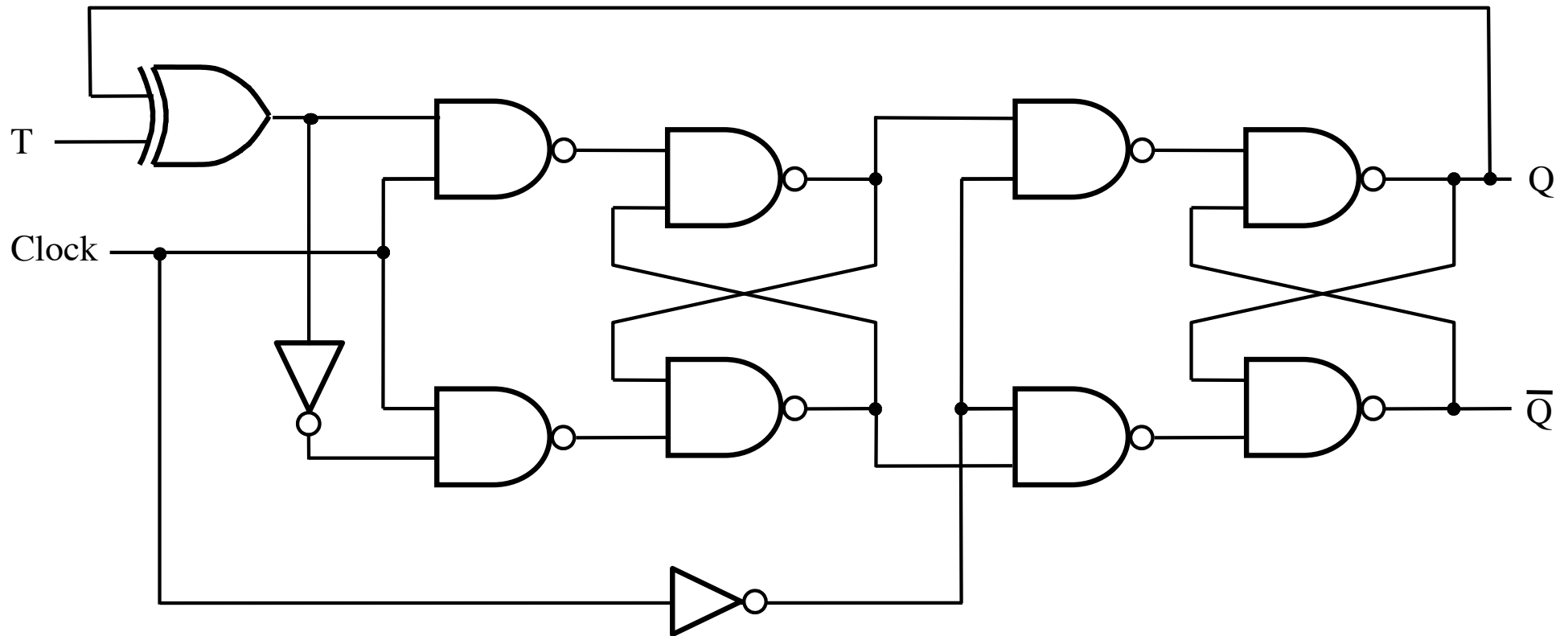
Negative-Edge-Triggered T Flip-Flop



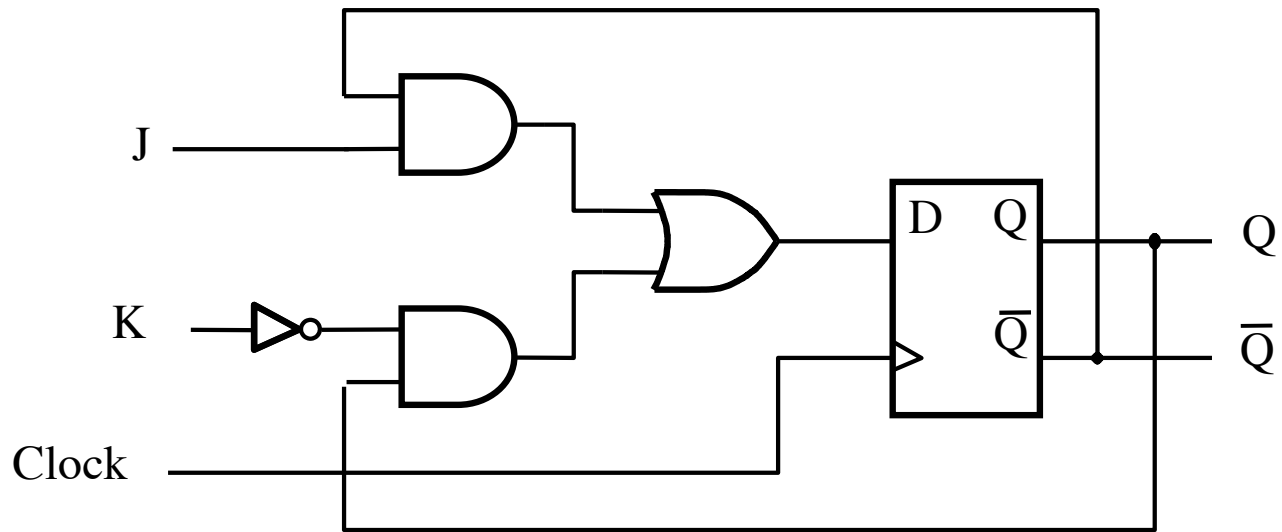
The Complete Wiring Diagram for a Positive-Edge-Triggered T Flip-Flop



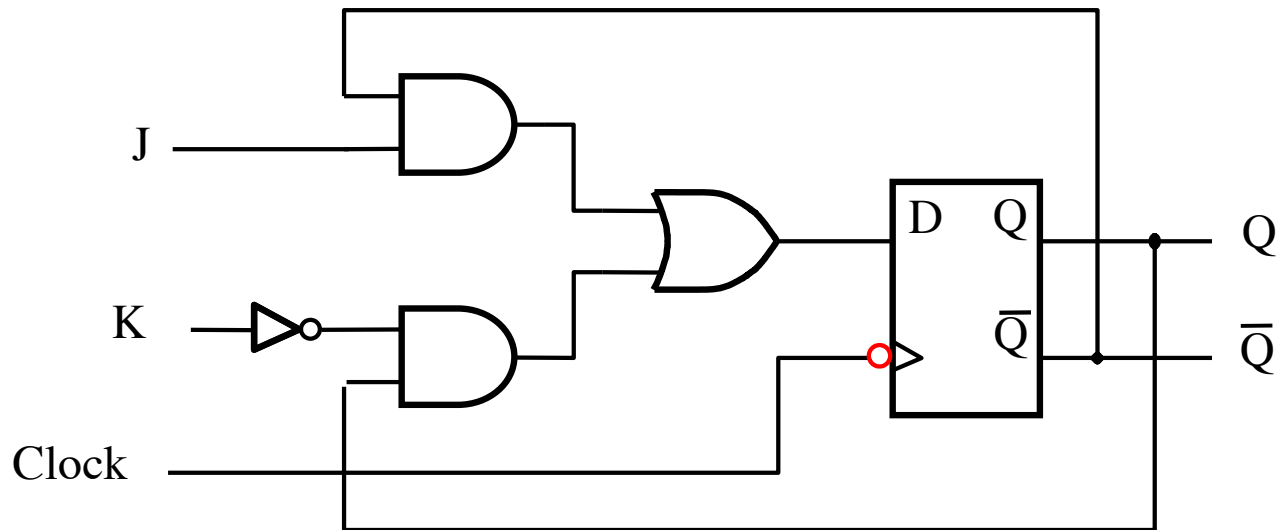
The Complete Wiring Diagram for a **Negative**-Edge-Triggered T Flip-Flop



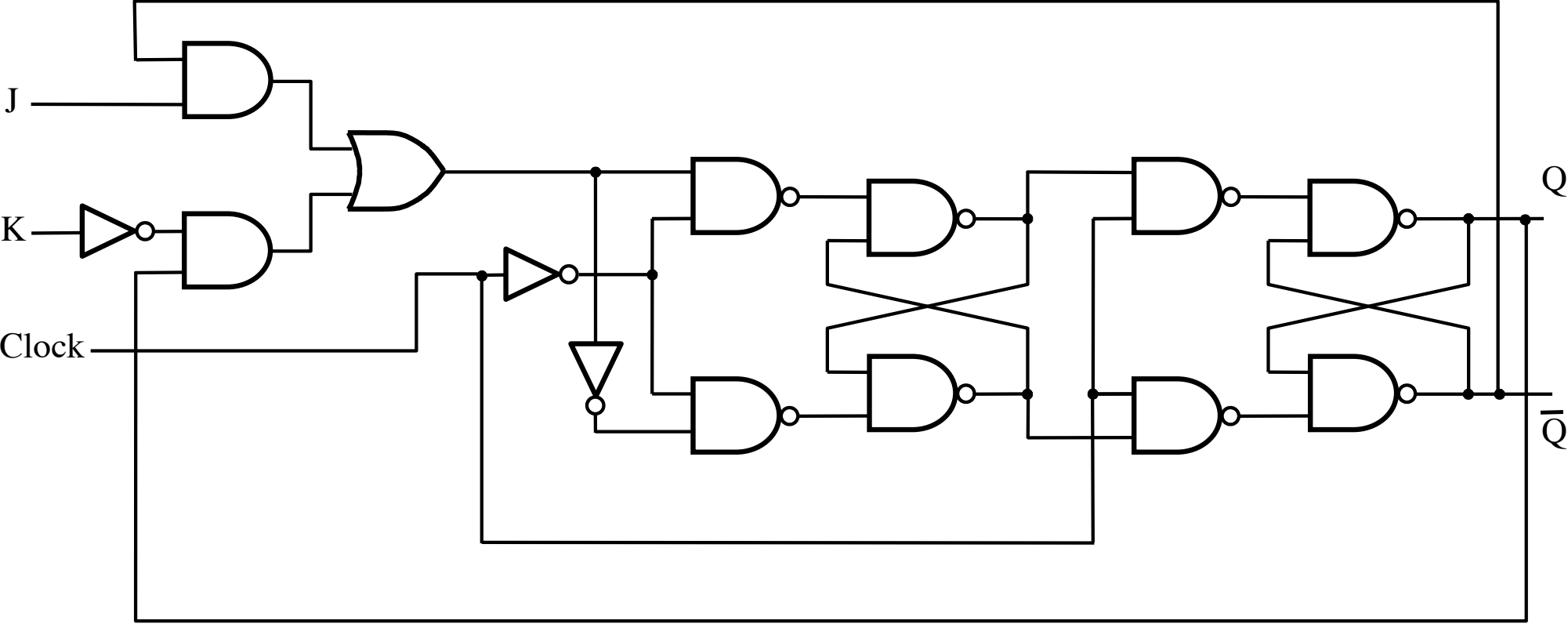
Positive-Edge-Triggered JK Flip-Flop



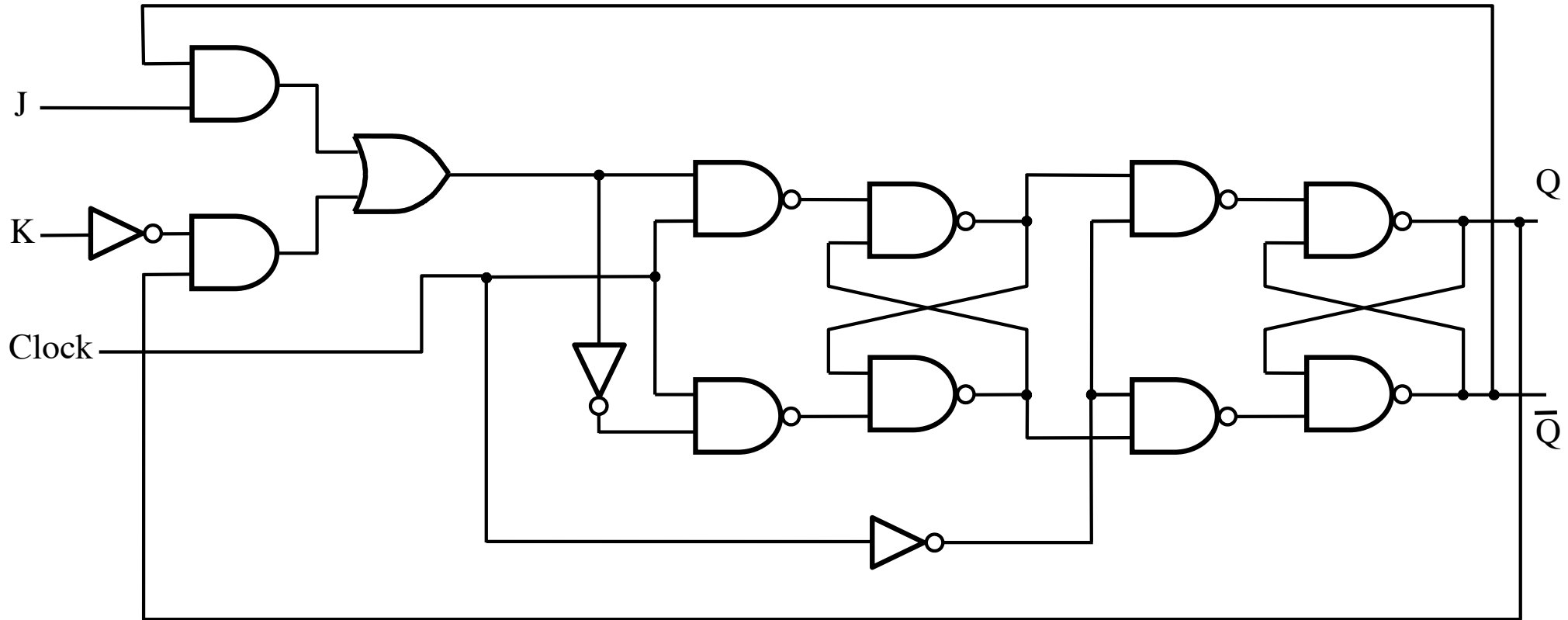
Negative-Edge-Triggered JK Flip-Flop



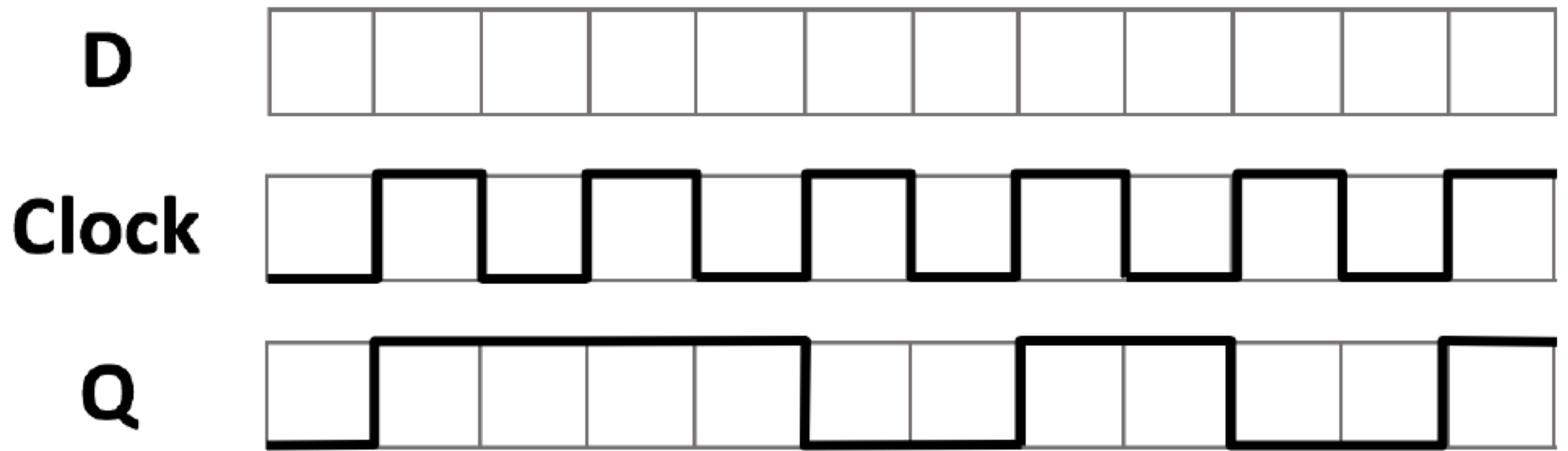
The Complete Wiring Diagram for a Positive-Edge-Triggered JK Flip-Flop

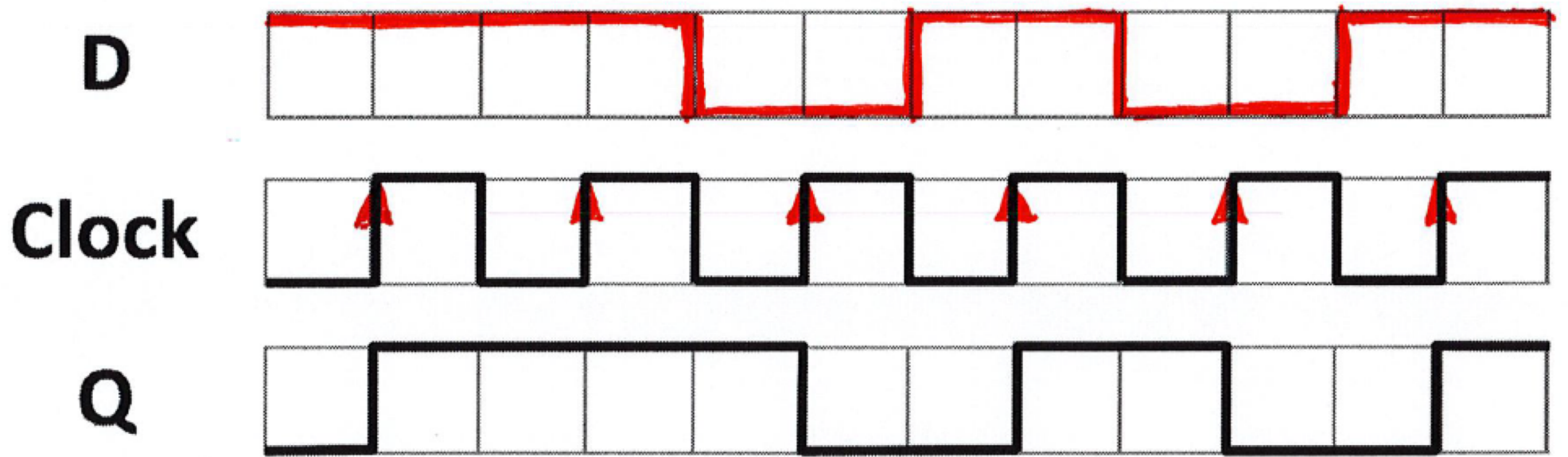


The Complete Wiring Diagram for a **Negative**-Edge-Triggered JK Flip-Flop



**Complete the Timing diagrams
(for positive-edge-triggered F-F)**





T

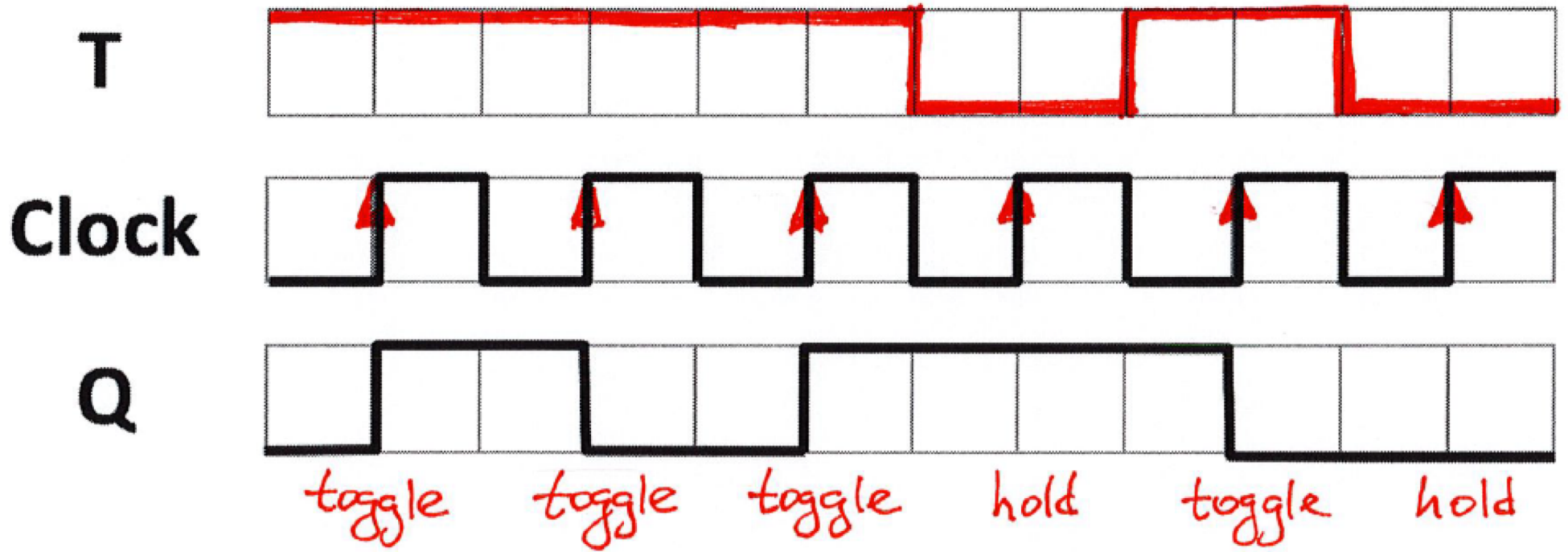


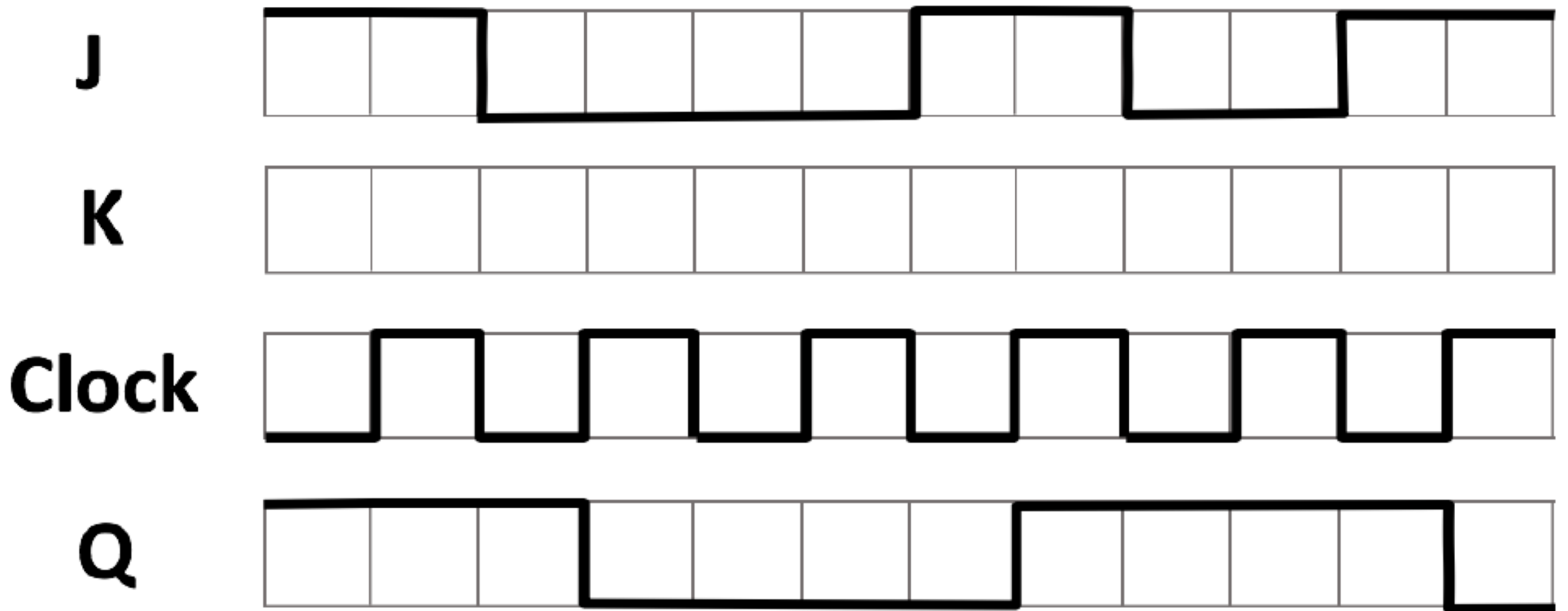
Clock

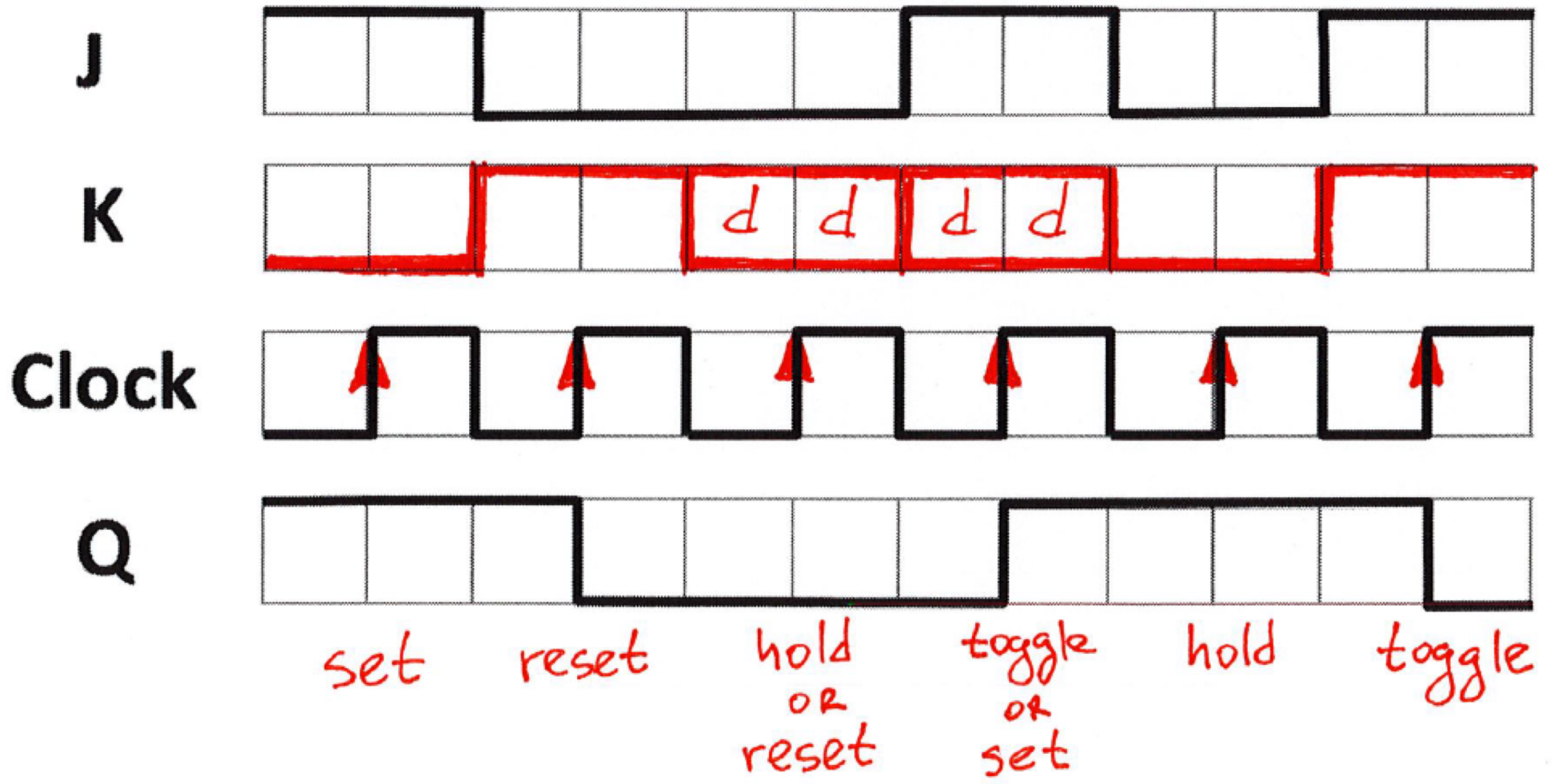


Q









**Complete the Timing diagrams
(for negative-edge-triggered F-F)**

D



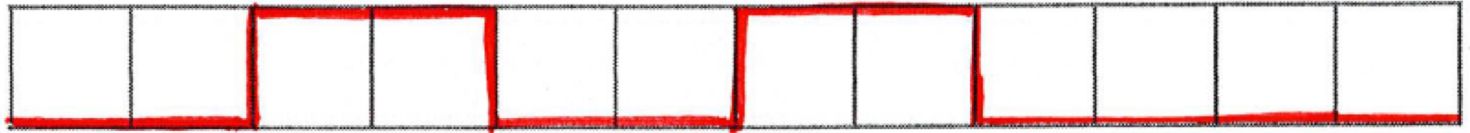
Clock



Q



D



Clock



Q



T



Clock



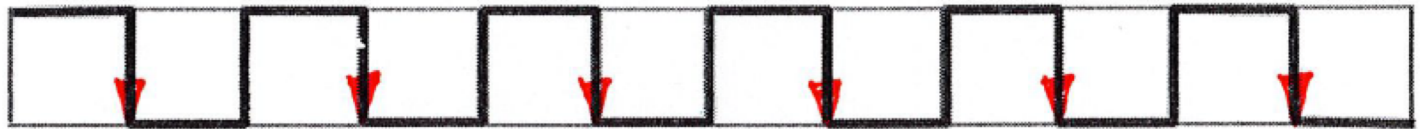
Q



T



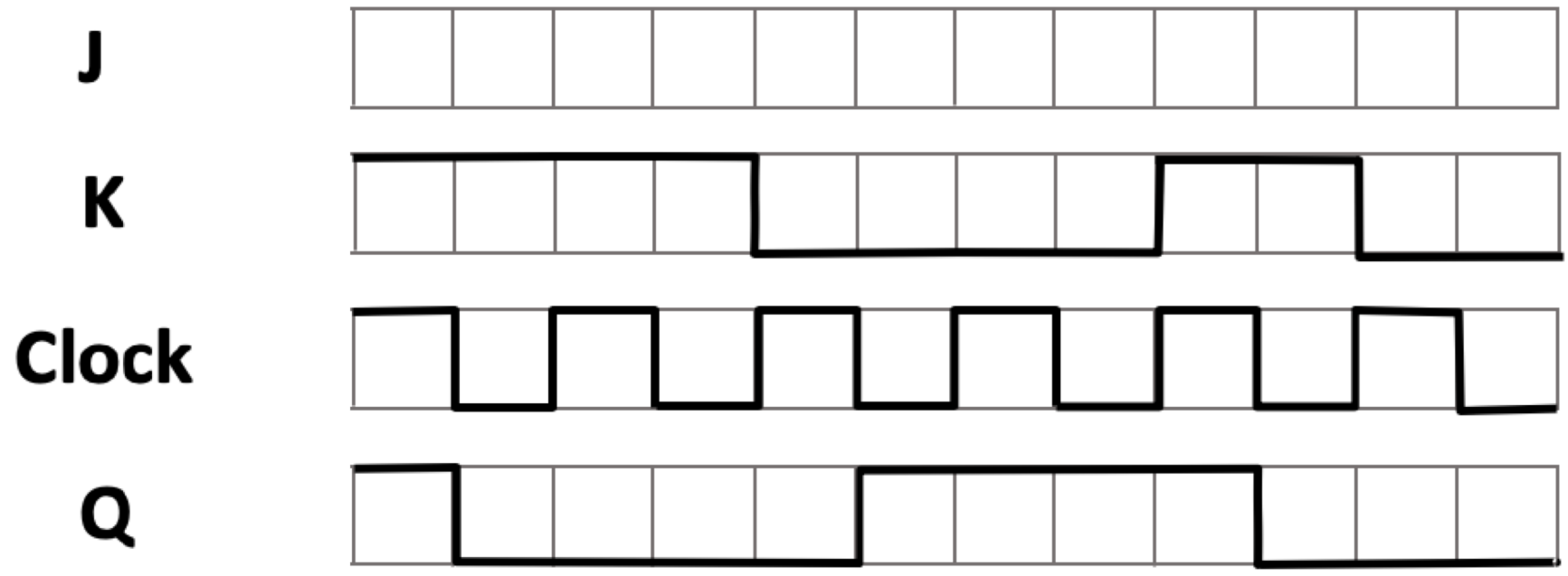
Clock

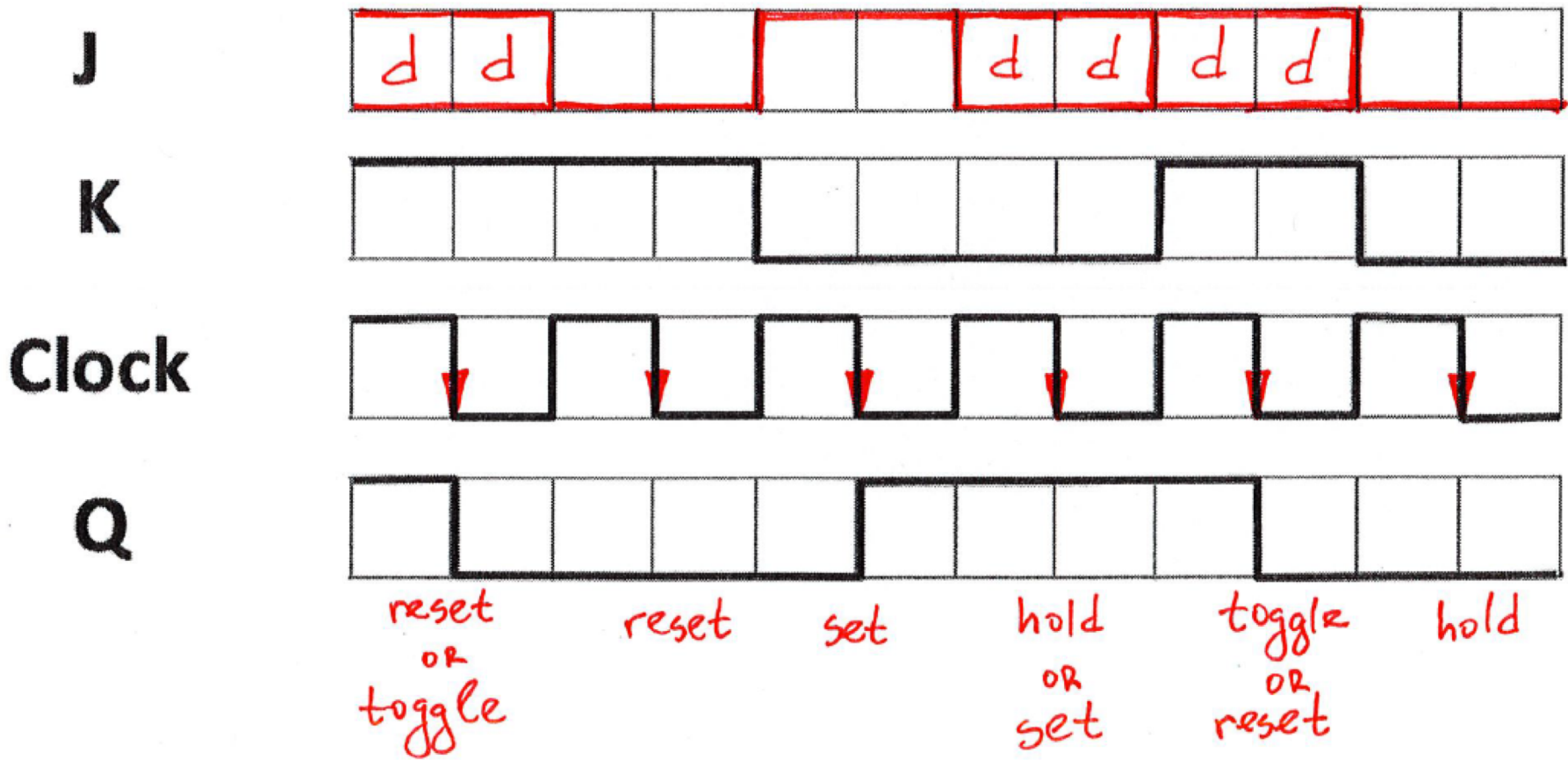


Q



toggle toggle hold toggle toggle hold





Questions?

THE END