S-R LATCH

Digital Logic Design (CC131)

BSCS 2nd Semester Spring-2026

Lecture: Week-8-a

By

Syed Zulfiqar Ali Shah,

Associate Professor (CS) / HoD

Govt. College of Management Sciences, Kohat

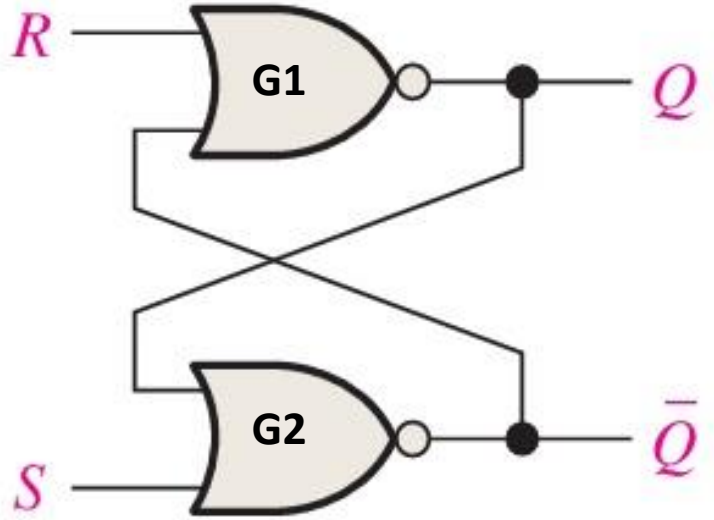
www.linkedin.com/in/syedzulfiqaralishah

SET-RESET Latch

- A Latch is a bistable logic device that can retain its state until it is changed.

- An active-HIGH input S-R latch is created with two NOR gates G1 and G2 in such a way that the output of G1 is connected as input to G2 and output of G2 is connected as input to G1.

- The outputs (Q , $\overline{Q}$) of these two NOR gates are Complement of each other.



SET-RESET Latch

- The cross output/input connection creates the regenerative feedback.

- Unlike combinational logic, which only reacts to current inputs, regenerative feedback allows a circuit to "remember" its previous state.

- This feedback is used to Lock or Retain its state.

  Once you tell it to be a 1, it stays a 1 until you tell it to be a Zero.

- The value of output Q is actually the data/state of this latch.

  Q = 1 means Latch is in SET state

  Q = 0 means Latch is in RESET state

- Example:

  Standard Doorbell has a single ON/OFF switch.

  S-R Latch is like a doorbell which has Two separate buttons for ON and OFF.

SET-RESET Latch

- For creating the S-R Latch, two types of digital logic can be used.
  - Active-HIGH
  - Active-LOW

- By Active-High or Active Low, we are strictly referring to the inputs (S and R).

- Ac-H/Ac-L shows what signal level is required to **activate** the logic.

SET-RESET Latch

- **Active-HIGH:**

  The latch is waiting for a 1 to do something.

  If you give it a 0, it stays idle (Hold).

- **Active-LOW:**

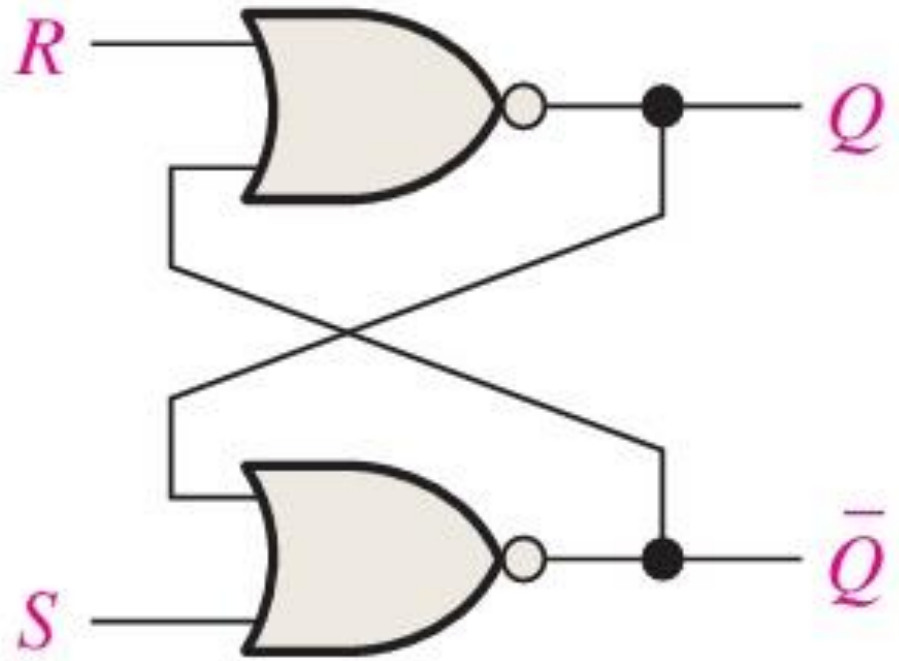
  The latch is waiting for a 0 to do something.

  If you give it a 1. it stays idle (Hold).

SET-RESET Latch

- In Active-High circuit, a 1 (High voltage) will be used for setting the state of the SR Latch and a 1 will be used for Resetting its state.

  (NOR Gate implementation)

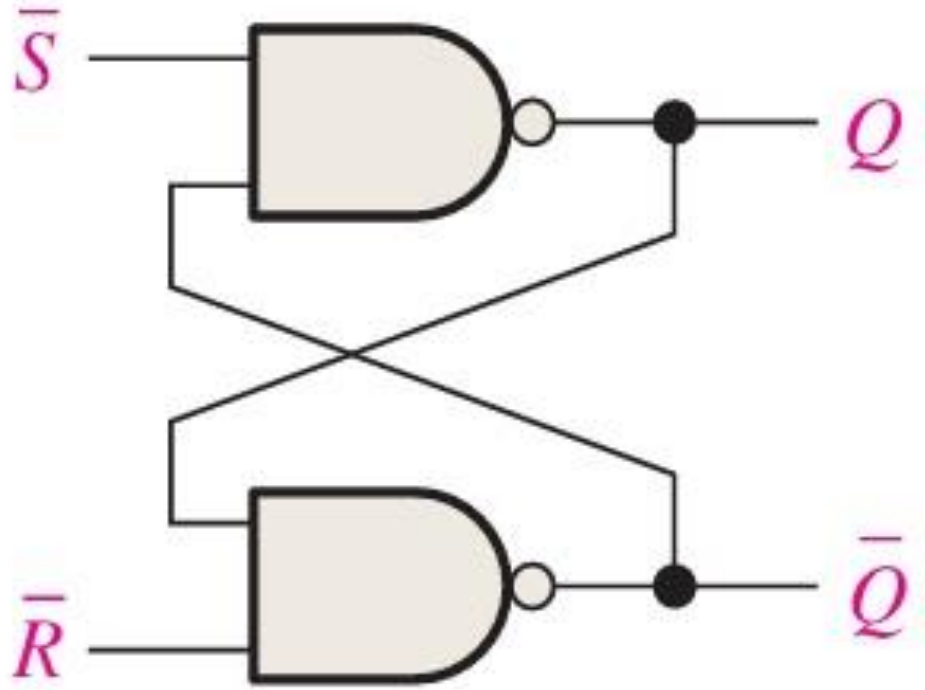


SET-RESET Latch

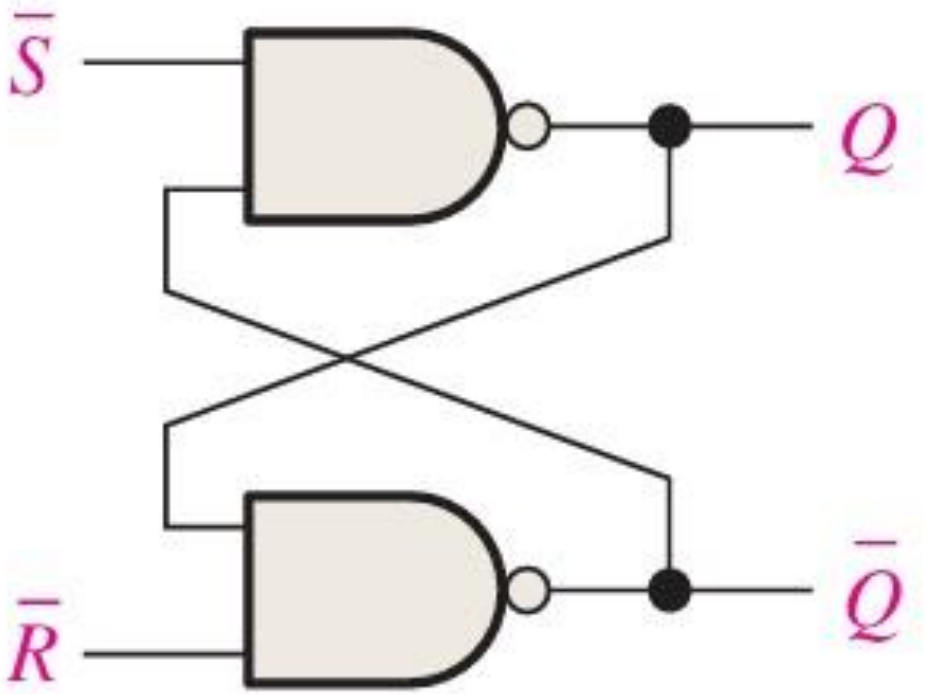
- In Active-Low circuit, a 0 (Low voltage) will be used for Setting or Resetting the state of the SR Latch.

  (NAND Gate implementation)

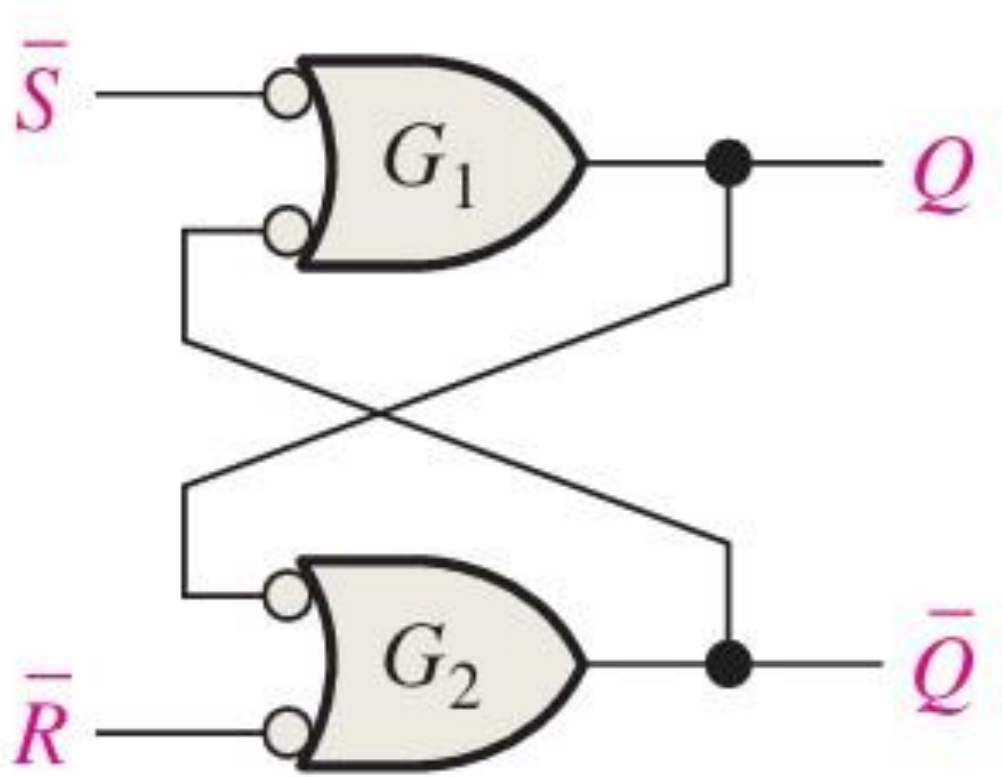
  (bar on the inputs ,i.e. $\overline{S}$, $\overline{R}$)



SET-RESET Latch



NAND gate $\overline{S}$ - $\overline{R}$ Latch



Negative-OR equivalent of the NAND gate $\overline{S}$ - $\overline{R}$ Latch

OPERATION of SR Latch

- To explain the operation of SR latch, we use the Active-LOW NAND gate $\overline{S}$ - $\overline{R}$ Latch.

- This latch performs an action when input drops to 0 (LOW).

- When both inputs ($\overline{S}$ - $\overline{R}$) are at 1 (HIGH), it simply stays in its current state.

- It means that the latch remembers whatever it was doing last; if it was SET, it stays SET; if it was RESET, it stays RESET.

1. SET Operation

GOAL: To make the Output Q = 1

ACTION: A LOW pulse is applied to $\overline{S}$ input while $\overline{R}$ is held HIGH.

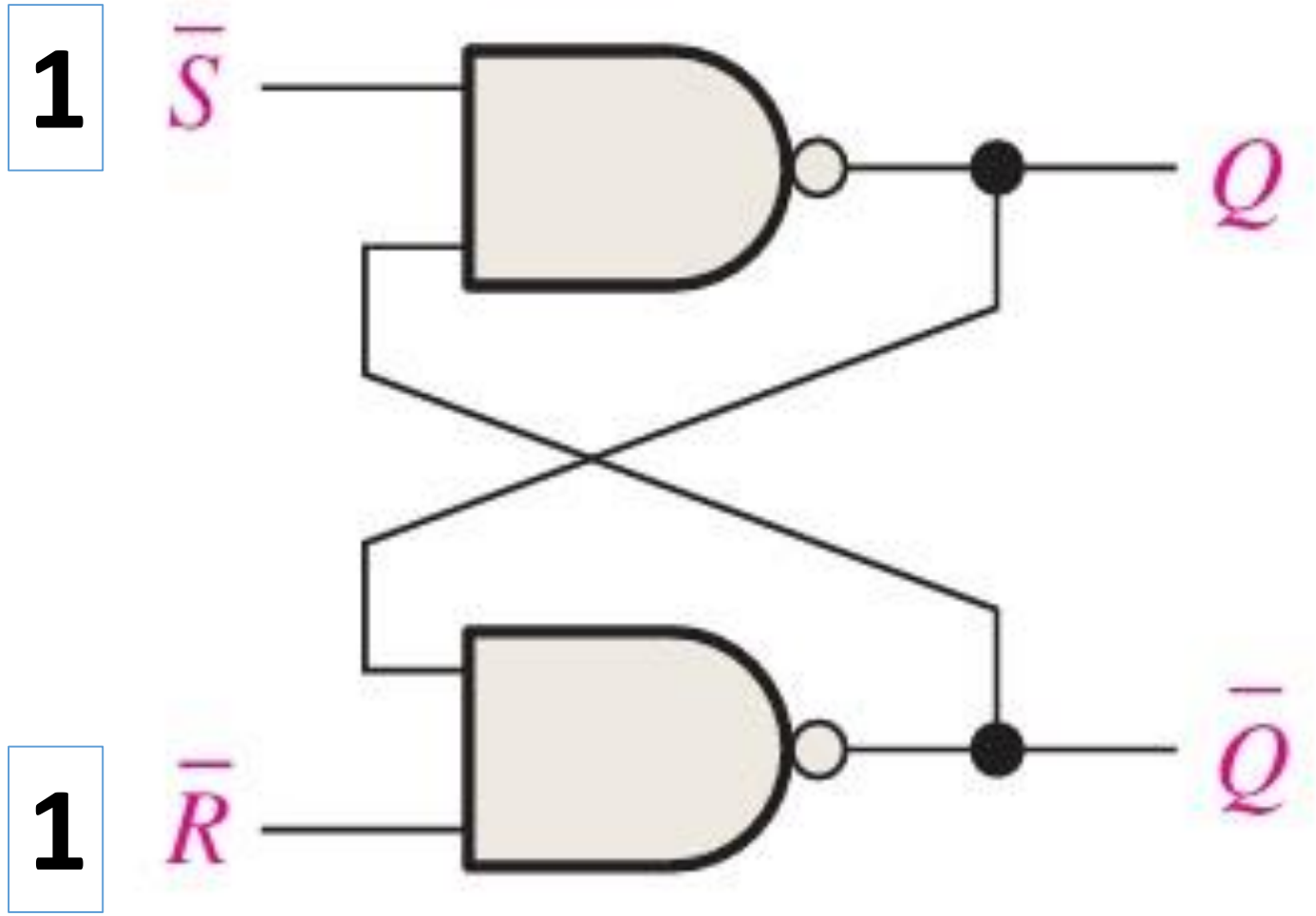
RESULT: If Q was 0, it switches to 1.
If Q was already 1, nothing changes.

MEMORY: Now if $\overline{S}$ goes back to HIGH, the output Q remains at 1.

OPERATION of SR Latch

1. SET Operation

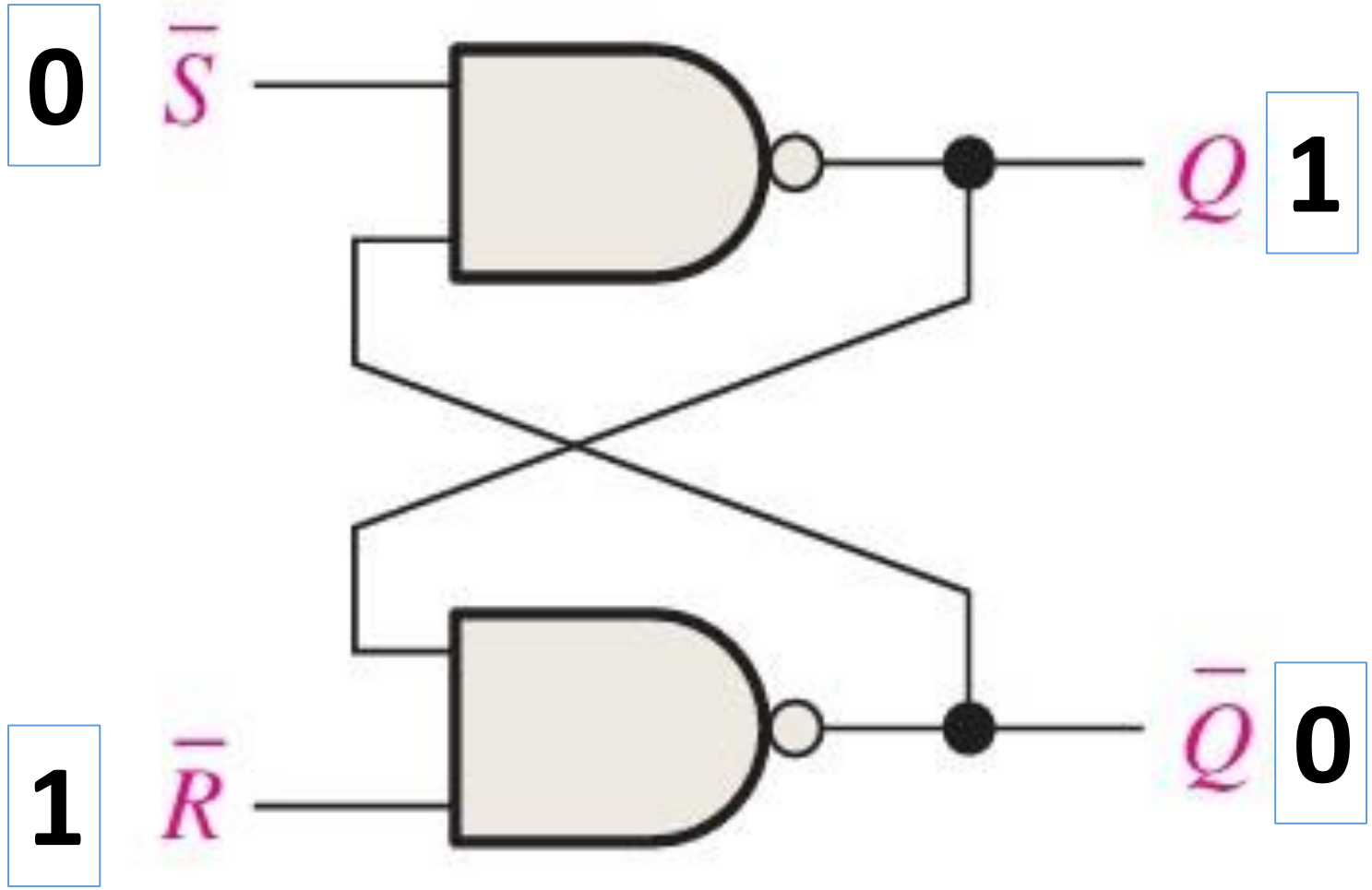
Initial stage when both inputs are High



OPERATION of SR Latch

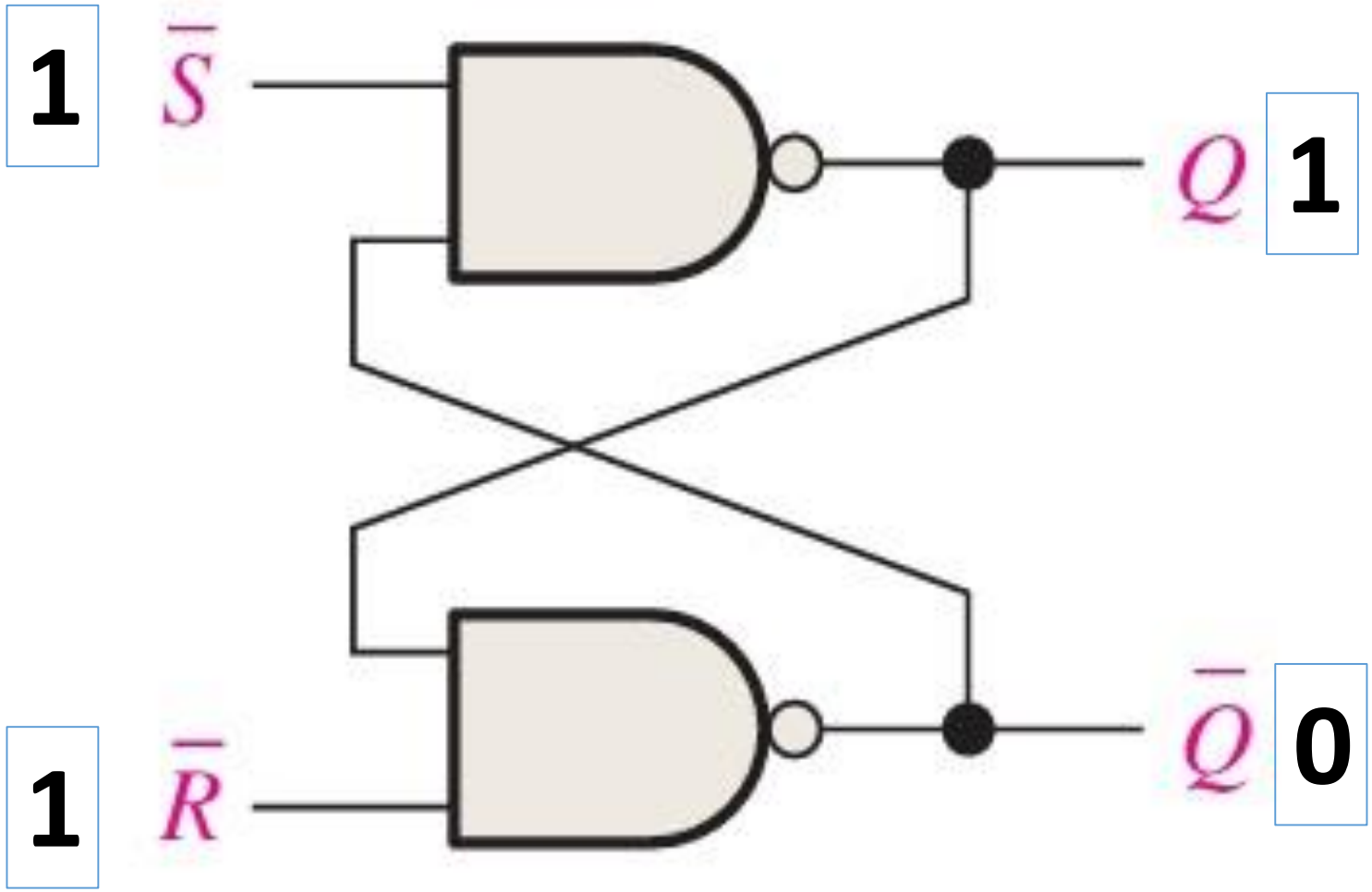
1. SET Operation

Output **Q** goes HIGH when input $\overline{S}$ is applied LOW



OPERATION of SR Latch

1. SET Operation



No Change in Output **Q** when $\bar{S}$ goes back to HIGH.

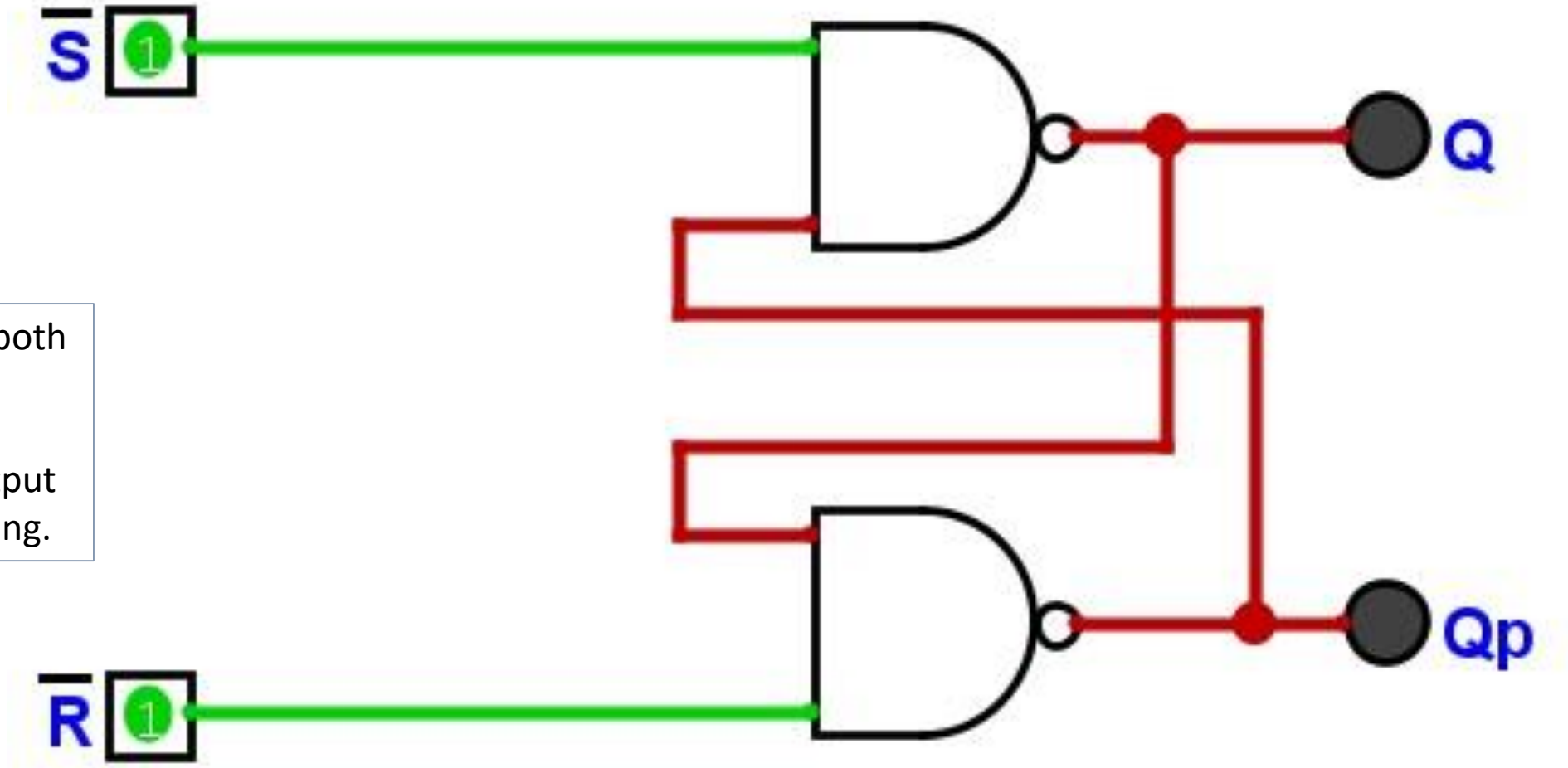
Current state is retained or remembered.

MEMORY

OPERATION of SR Latch

1. SET Operation

PRACTICAL: Logisim-evolution Screen Capture



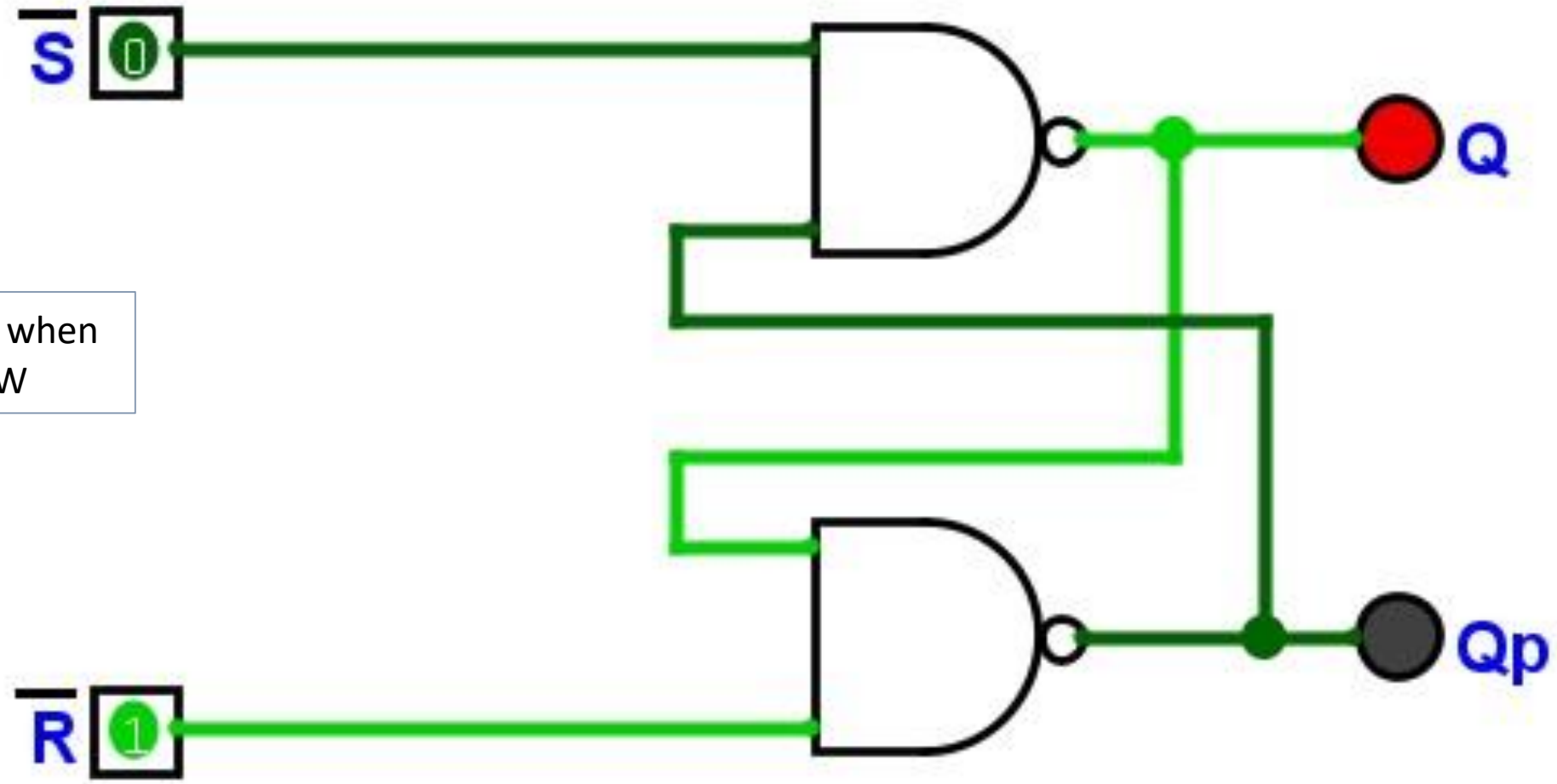
Initial stage when both inputs are High.

At this 0-stage, output state can be anything.

OPERATION of SR Latch

1. SET Operation

PRACTICAL: Logisim-evolution Screen Capture

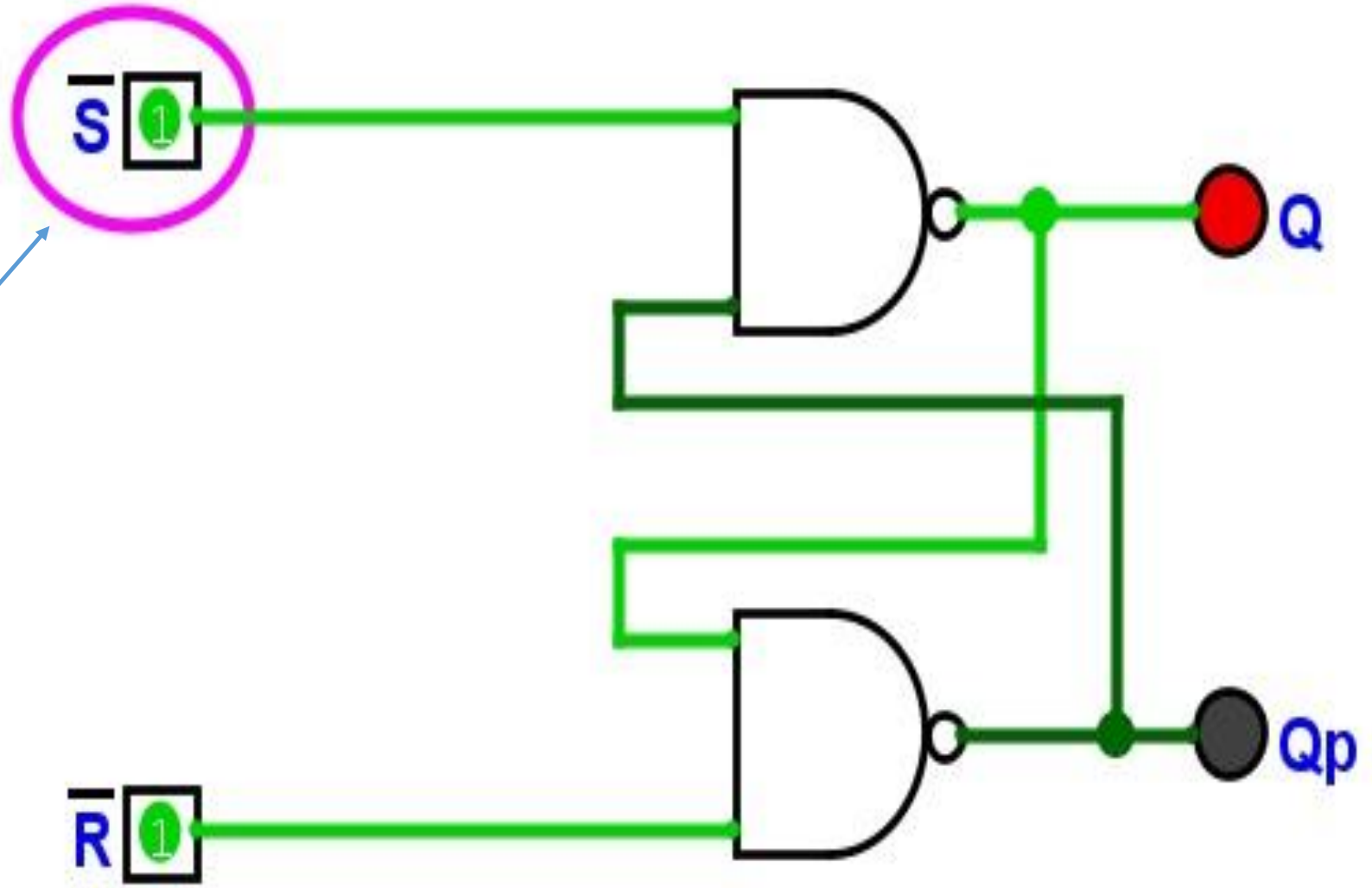


Output **Q** goes HIGH when input $\overline{S}$ is applied LOW

OPERATION of SR Latch

1. SET Operation

PRACTICAL: Logisim-evolution Screen Capture



No Change in Output **Q** when $\overline{S}$ goes back to HIGH.

Current state is retained or remembered.

MEMORY

2. RESET Operation

GOAL: To make the Output Q = 0

ACTION: A LOW pulse is applied to input $\overline{R}$ while $\overline{S}$ is held HIGH.

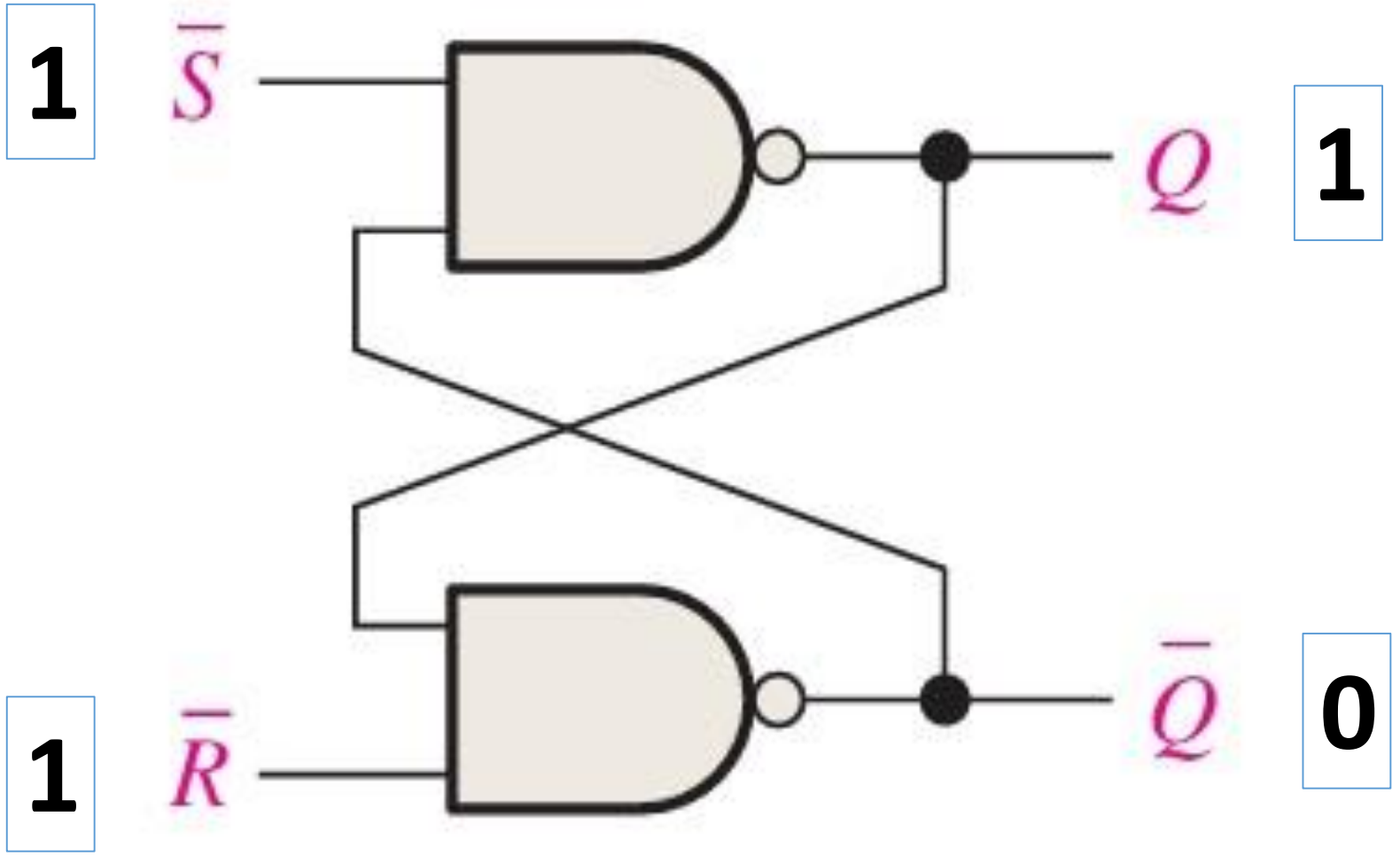
RESULT: If Q was 1, it switches to 0.
If Q was already 0, it stays at 0.

MEMORY: Now if $\overline{R}$ goes back to HIGH, the output Q remains at 0.

OPERATION of SR Latch

2. RESET Operation

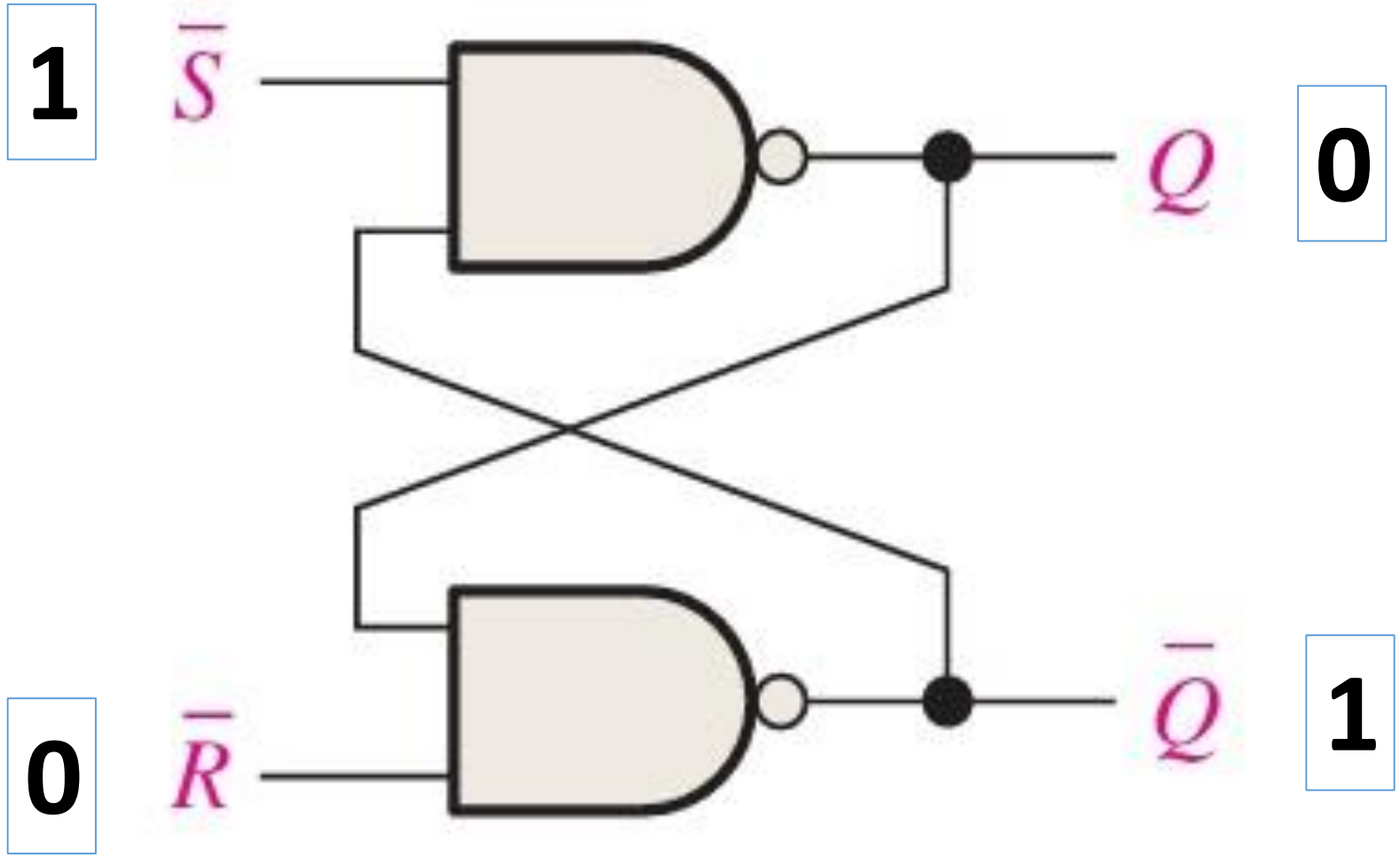
Previous stage when both inputs were High and No change in Output.



OPERATION of SR Latch

2. RESET Operation

Output **Q** goes LOW when input $\overline{R}$ is applied LOW while keeping $\overline{S}$ HIGH.



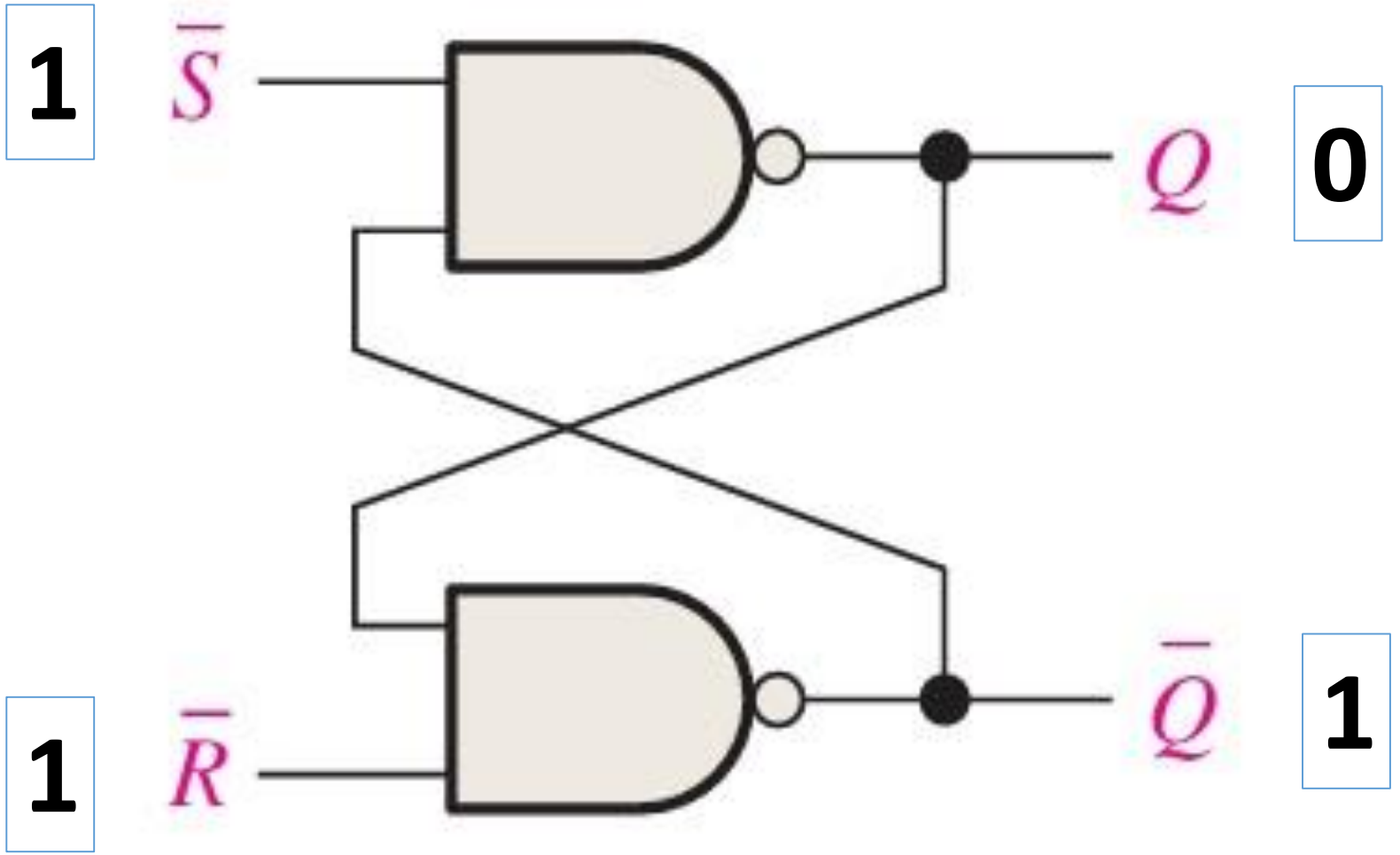
OPERATION of SR Latch

2. RESET Operation

No Change in Output **Q** when $\overline{R}$ goes back to HIGH.

Current state is retained or remembered.

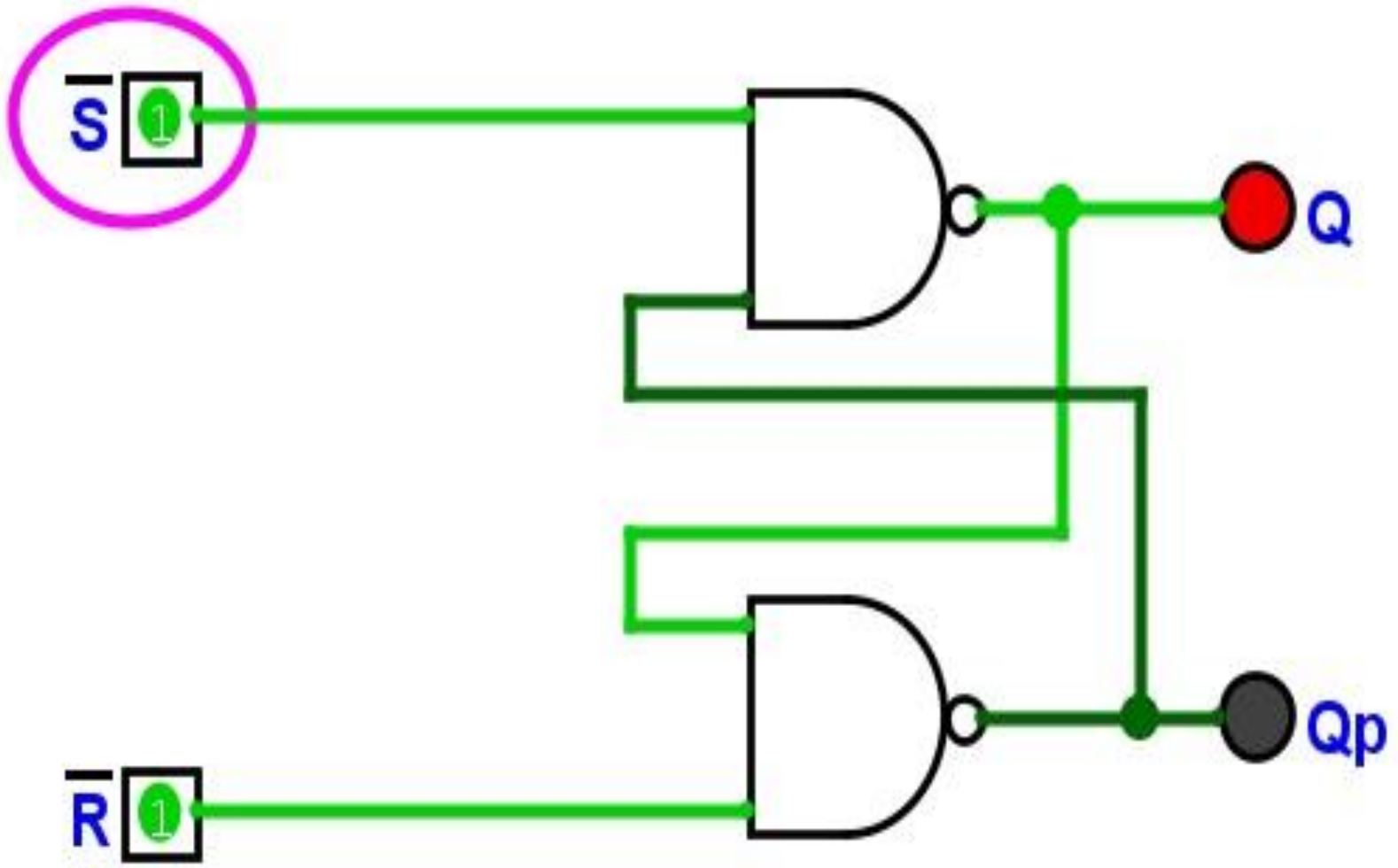
MEMORY



OPERATION of SR Latch

2. RESET Operation

PRACTICAL: Logisim-evolution Screen Capture



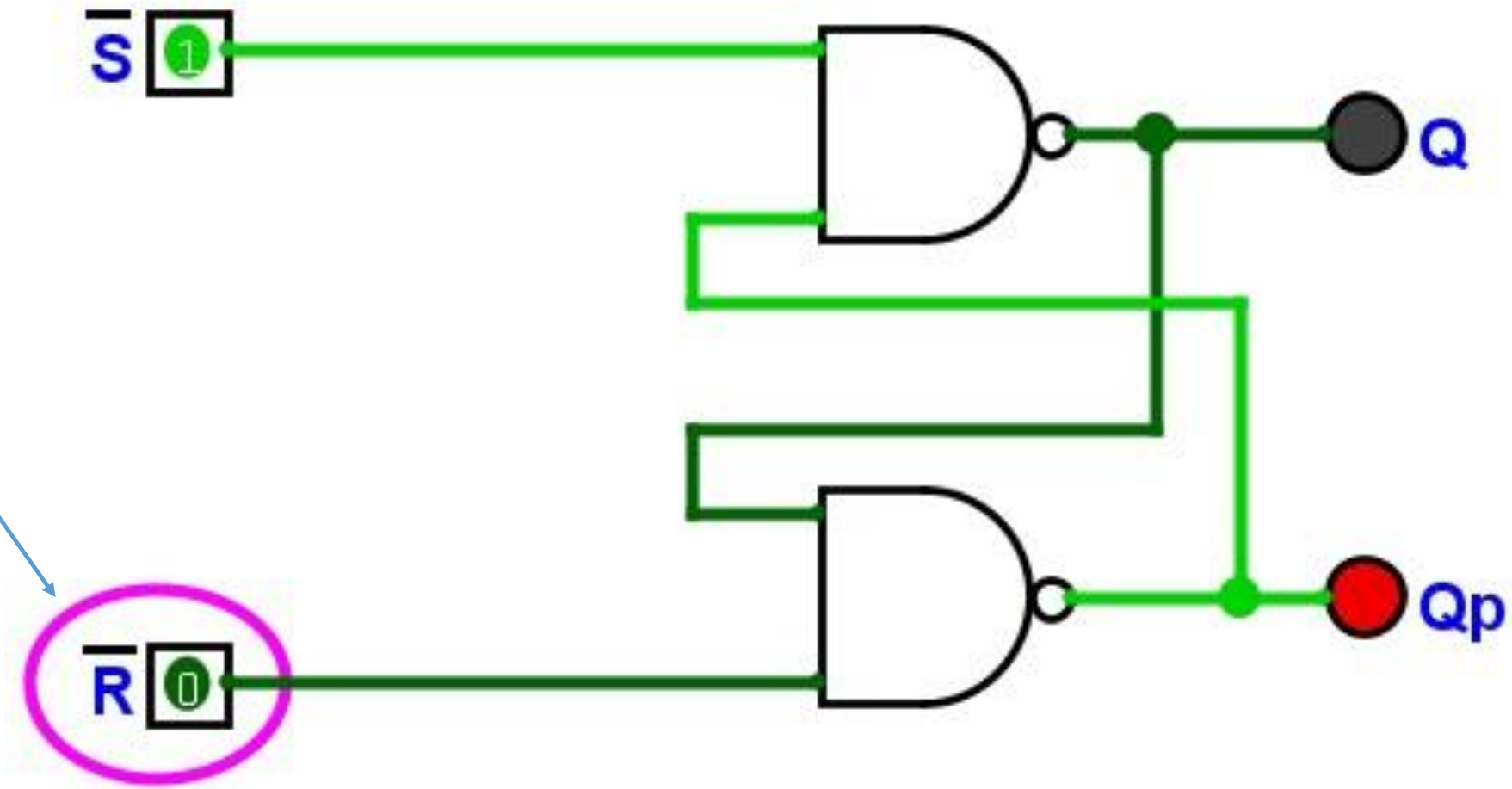
Previous stage when both inputs were High and No change in Output.

OPERATION of SR Latch

2. RESET Operation

PRACTICAL: Logisim-evolution Screen Capture

Output **Q** goes LOW when input $\overline{R}$ is applied LOW while keeping $\overline{S}$ HIGH.



OPERATION of SR Latch

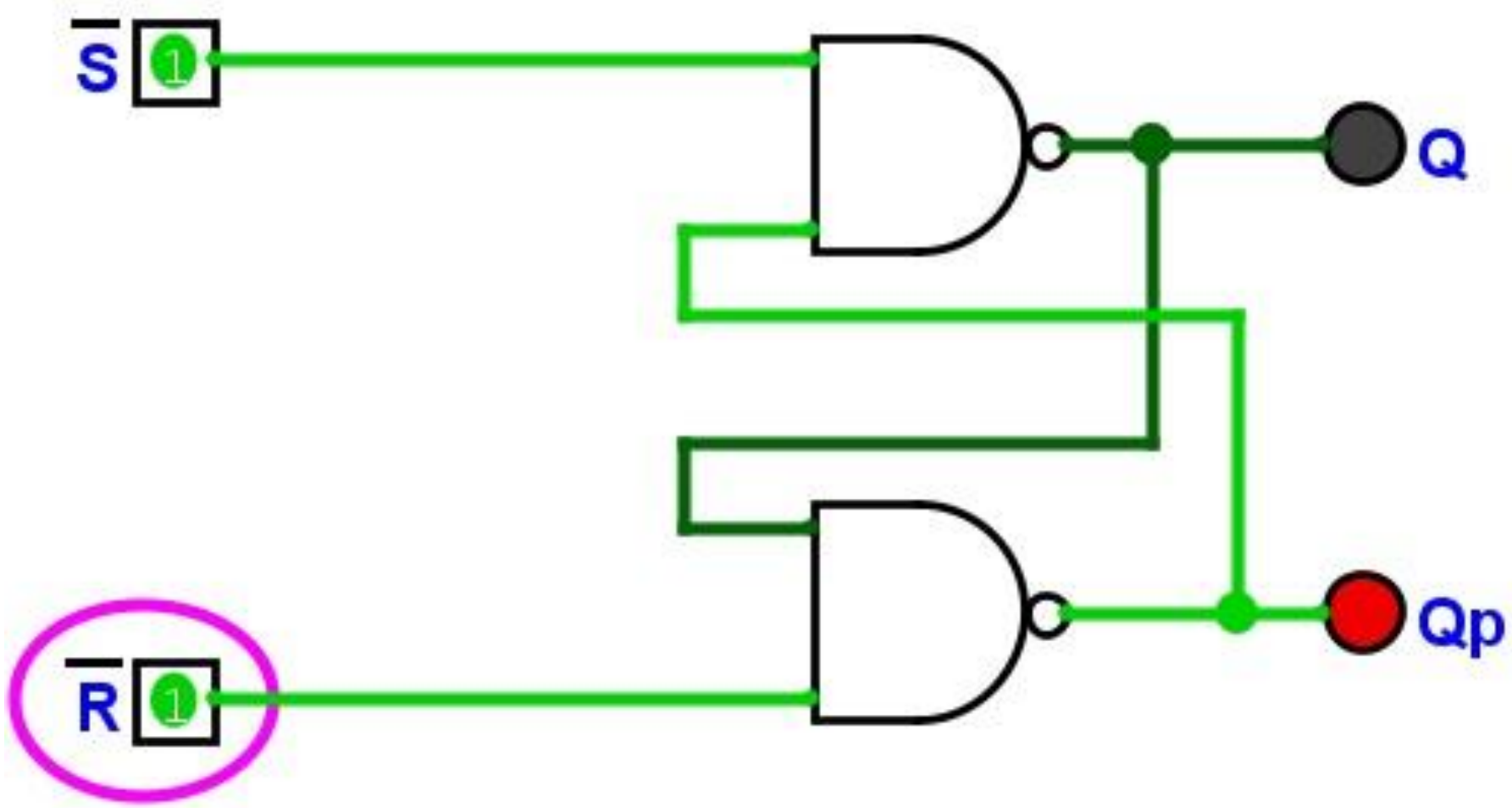
2. RESET Operation

PRACTICAL: Logisim-evolution Screen Capture

No Change in Output **Q** when $\overline{R}$ goes back to HIGH.

Current state is retained or remembered.

MEMORY



3. NO Change Condition

This is the "**resting**" state of the Latch.

CONDITION: Both inputs $\overline{S}$ and $\overline{R}$ are held HIGH.

RESULT: The latch remembers whatever it was doing.

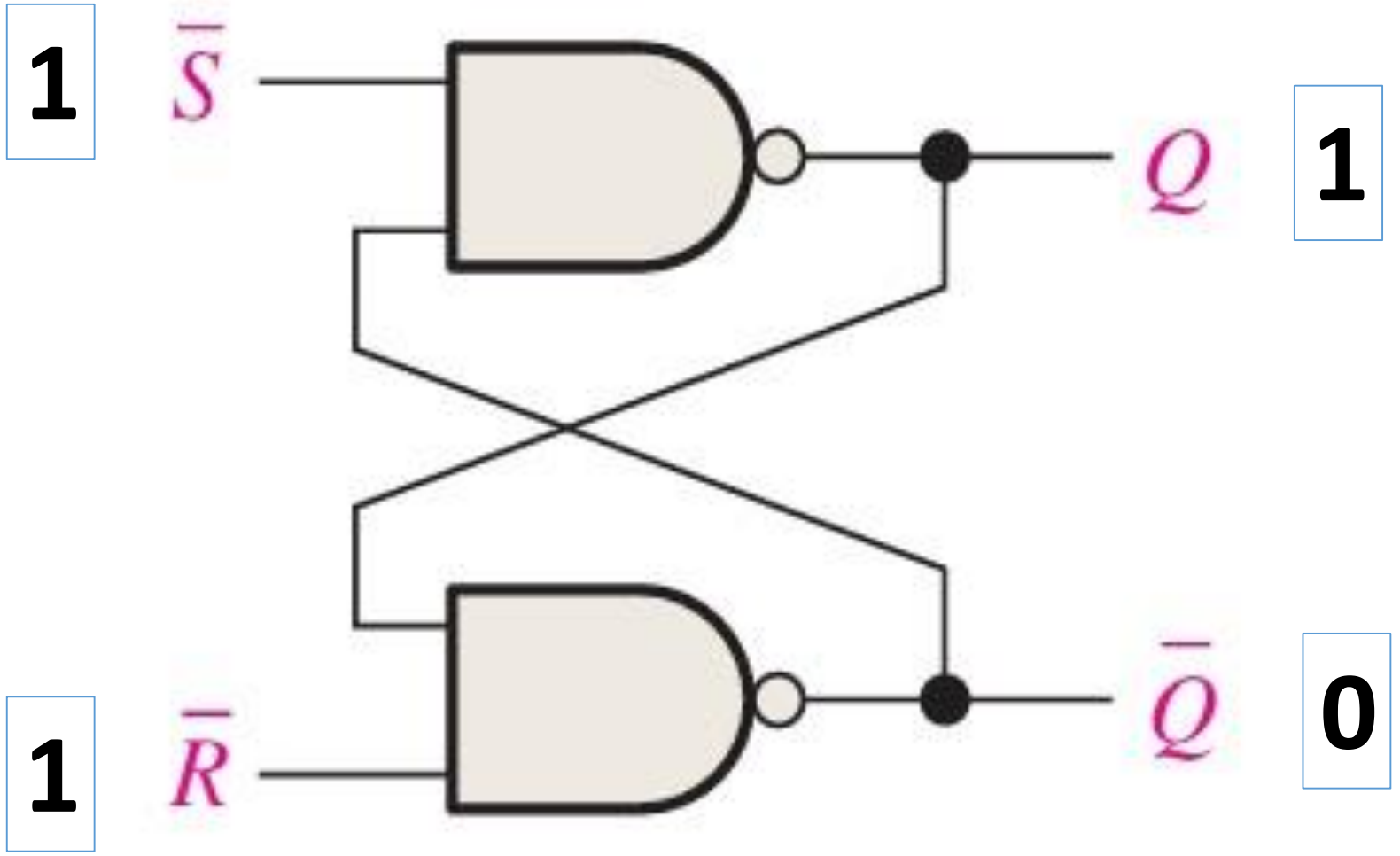
If it was SET, stays at SET.

If it was RESET, stays at RESET.

OPERATION of SR Latch

3. NO Change Condition

When both inputs are High there is No change in Output with respect to previous state.

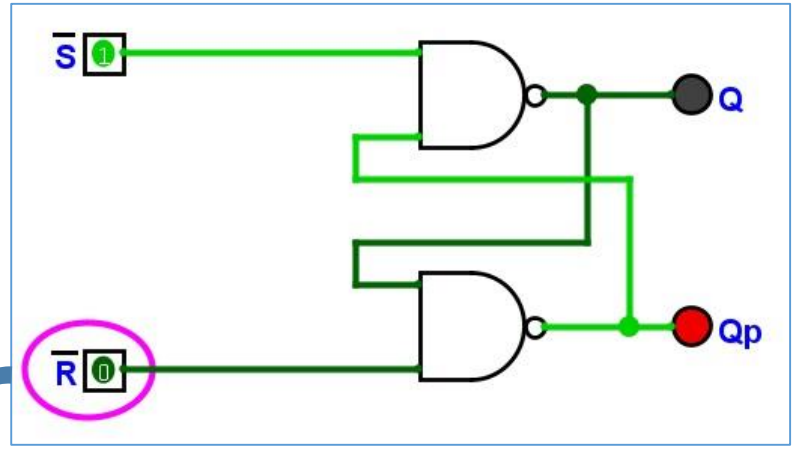


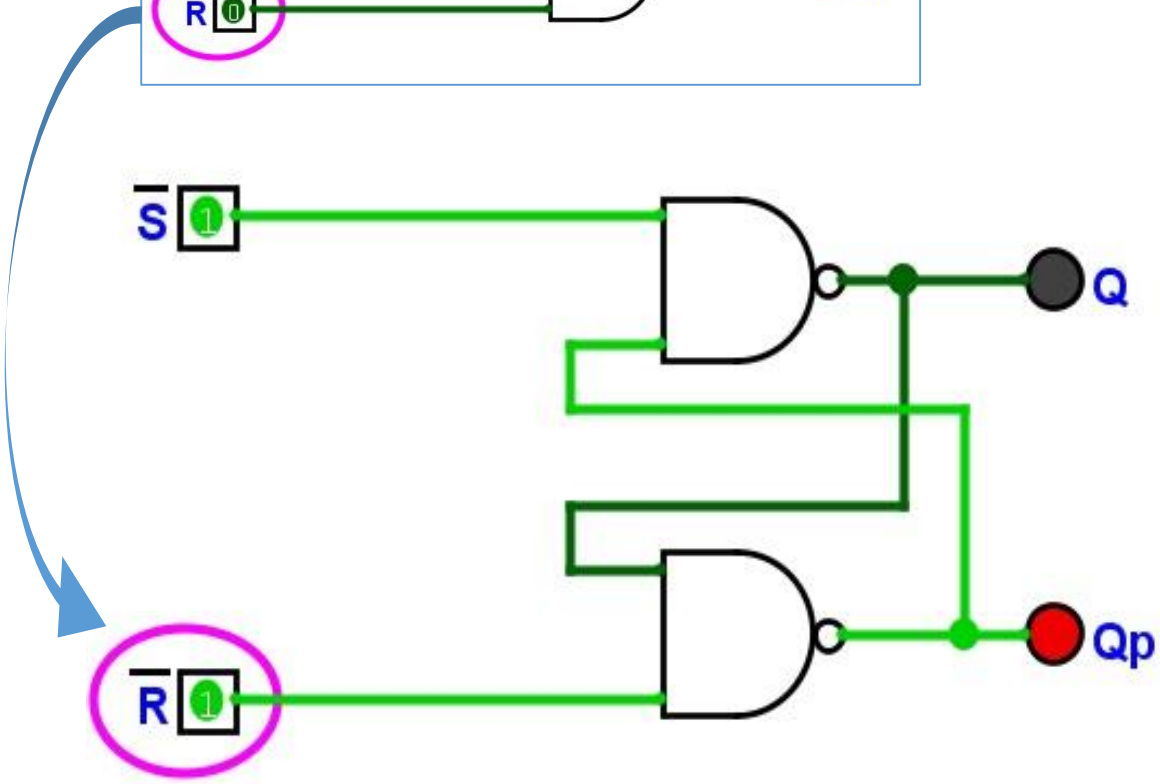
OPERATION of SR Latch

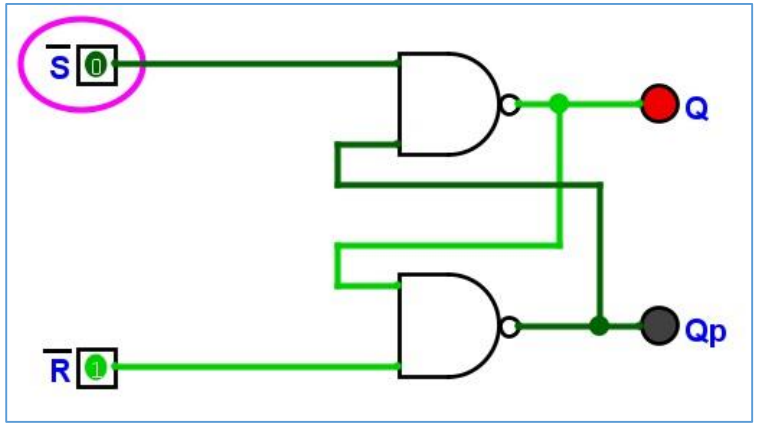
3. NO Change Condition

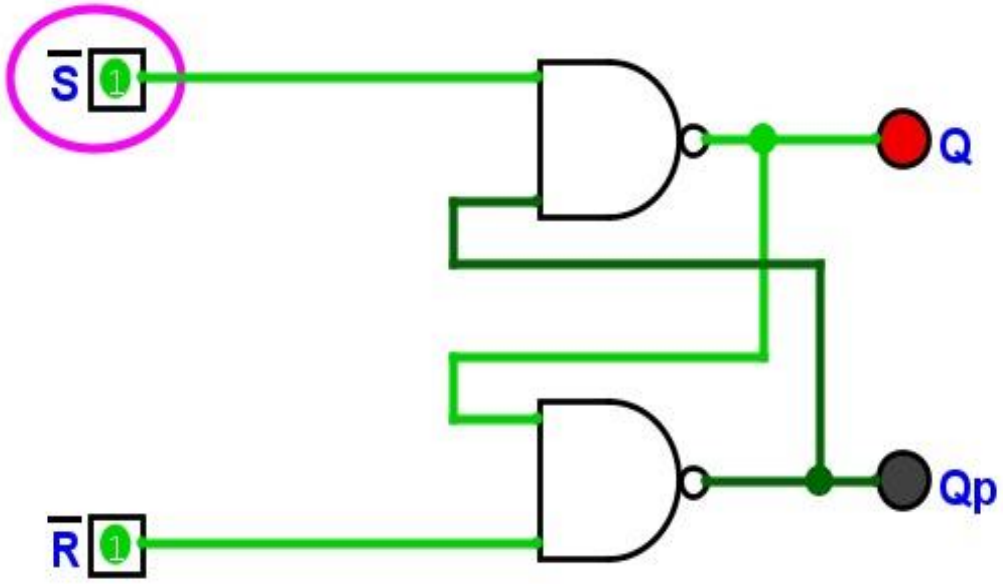
PRACTICAL: Logisim-evolution Screen Capture

No effect by changing any input.









4. The Invalid Condition

This state is generally avoided.

CONDITION: Both inputs $\overline{S}$ and $\overline{R}$ are applied LOW simultaneously.

PROBLEM: This results in both **Q** and $\overline{\mathbf{Q}}$ to go HIGH at the same time which contradicts the logic that $\overline{\mathbf{Q}}$ should be the opposite of **Q**.

The **RACE**: When both inputs $\overline{S}$ and $\overline{R}$ return to HIGH at the same time, the final state of the latch is unpredictable.

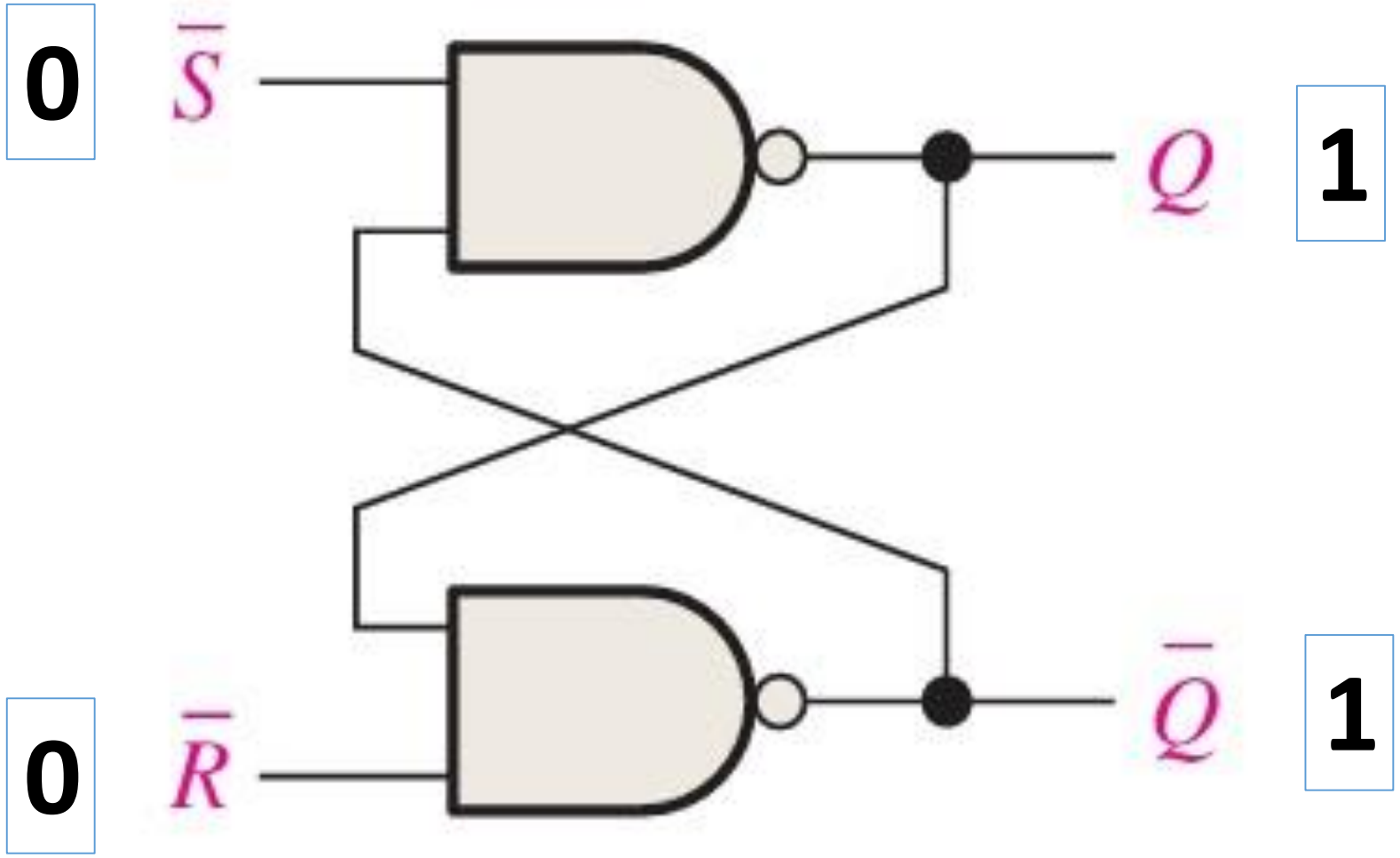
It depends on which Gate is slightly faster.

OPERATION of SR Latch

4. The Invalid Condition

When both inputs are applied LOW at the same time, it is the INVALID state.

Because both Outputs go to HIGH and this is not possible in digital logic.



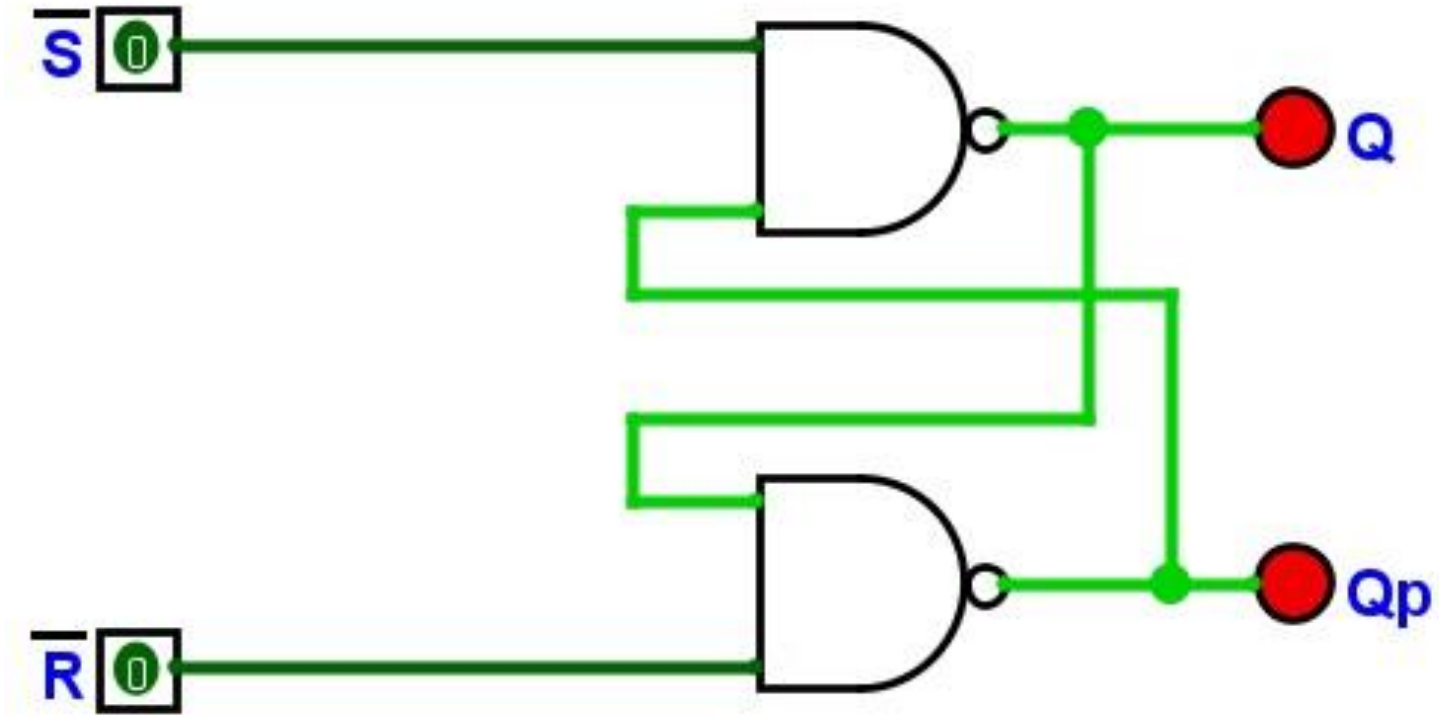
OPERATION of SR Latch

4. The Invalid Condition

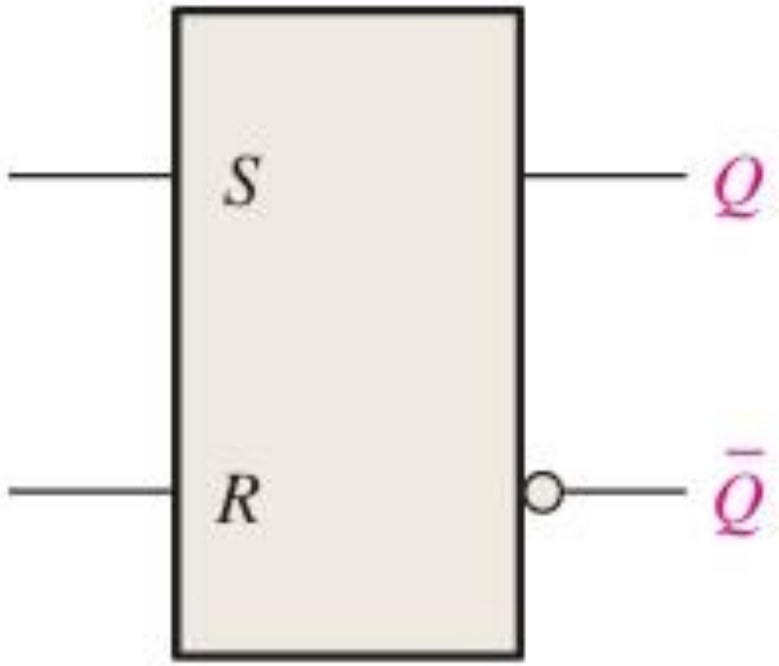
When both inputs are applied LOW at the same time, it is the INVALID state.

Because both Outputs go to HIGH and this is not possible in digital logic.

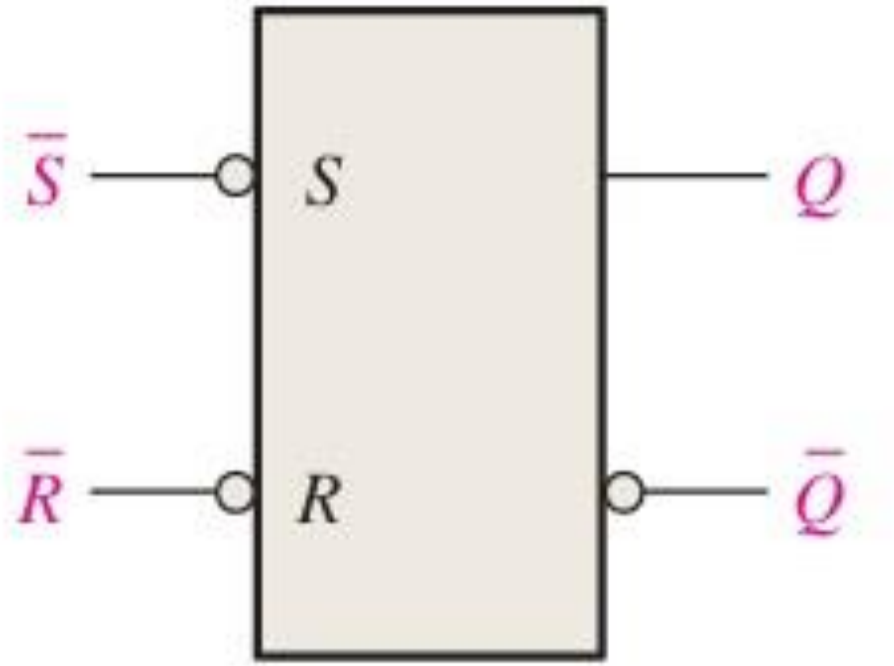
PRACTICAL: Logisim-evolution Screen Capture



SYMBOLS



(a) Active-HIGH input S-R latch



(b) Active-LOW input $\overline{S}$-$\overline{R}$ latch

TRUTH TABLE

Truth table for an active-LOW input $\overline{S}$-$\overline{R}$ latch.

Inputs		Outputs		
$\overline{S}$	$\overline{R}$	Q	$\overline{Q}$	**Comments**
1	1	NC	NC	No change. Latch remains in present state.
0	1	1	0	Latch SET.
1	0	0	1	Latch RESET.
0	0	1	1	Invalid condition

TRUTH TABLE

Truth Table for an Active-HIGH input S-R latch				
Inputs		Outputs		Comments
S	R	Q	$\overline{Q}$	
0	0	NC	NC	No Change. Latch remains in present state.
1	0	1	0	Latch SET
0	1	0	1	Latch RESET
1	1	0	0	Invalid Condition (Forbidden)

Thank You