Computer Architecture

Lecture 05

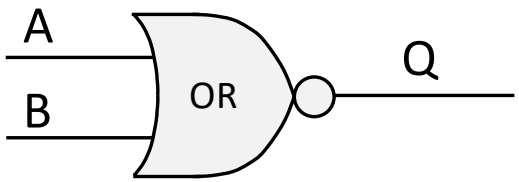
Set/Reset Latches

Artem Burmyakov, Alexander Tormasov

September 23, 2021

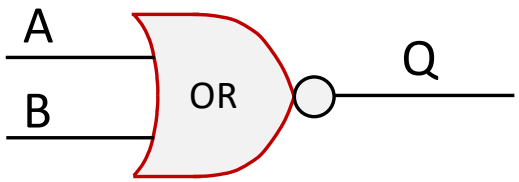
INNOPOLIS UNIVERSITY

Recap: NOR gate



NOR = OR + NOT

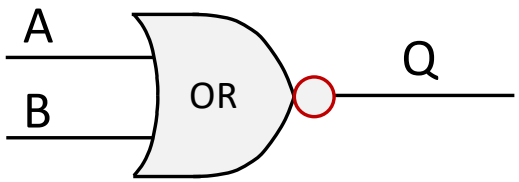
Recap: NOR gate



A	B	OR
1	1	1
1	0	1
0	1	1
0	0	0

NOR = OR + NOT

Recap: NOR gate

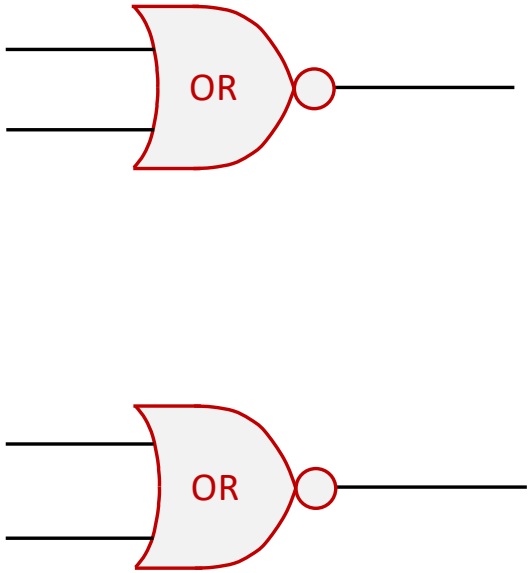


NOR = OR + NOT

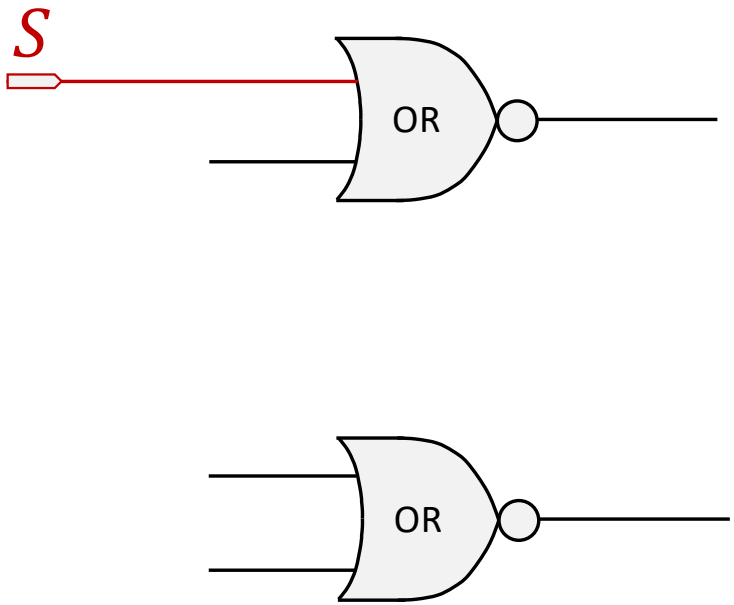
A	B	OR	NOR
1	1	1	0
1	0	1	0
0	1	1	0
0	0	0	1

If at least one input is “1”, NOR outputs “0”

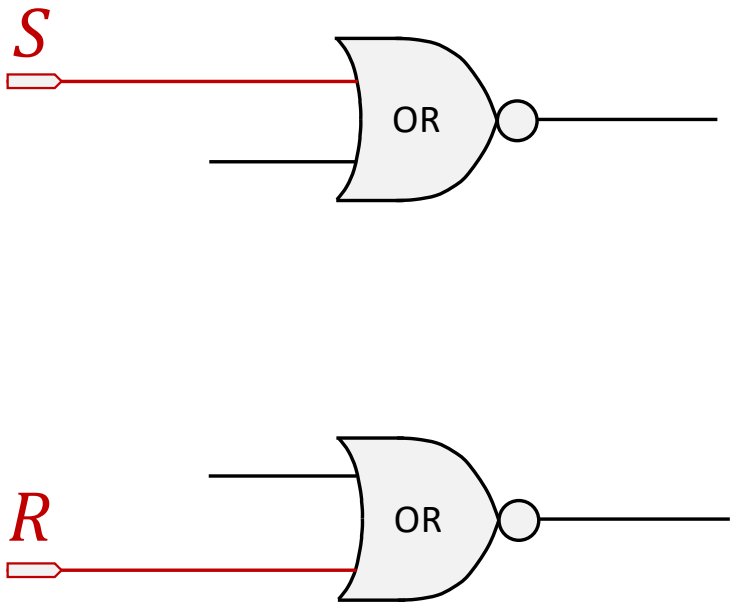
Circuit Construction: 2 NOR gates



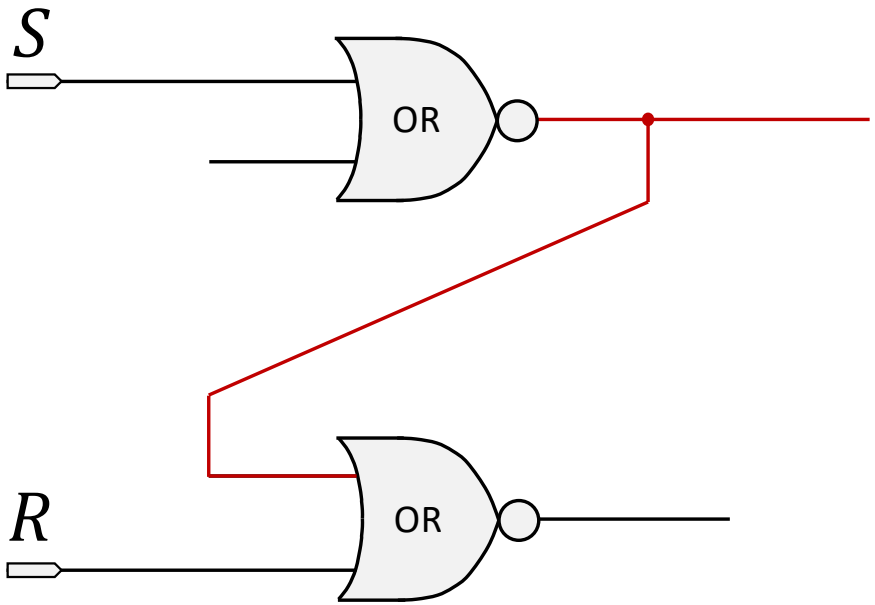
Circuit Construction: Input Pin S



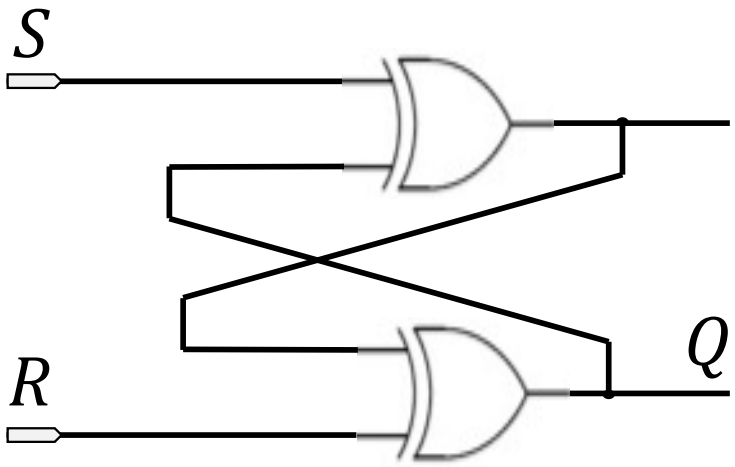
Circuit Construction: Input Pins S and R



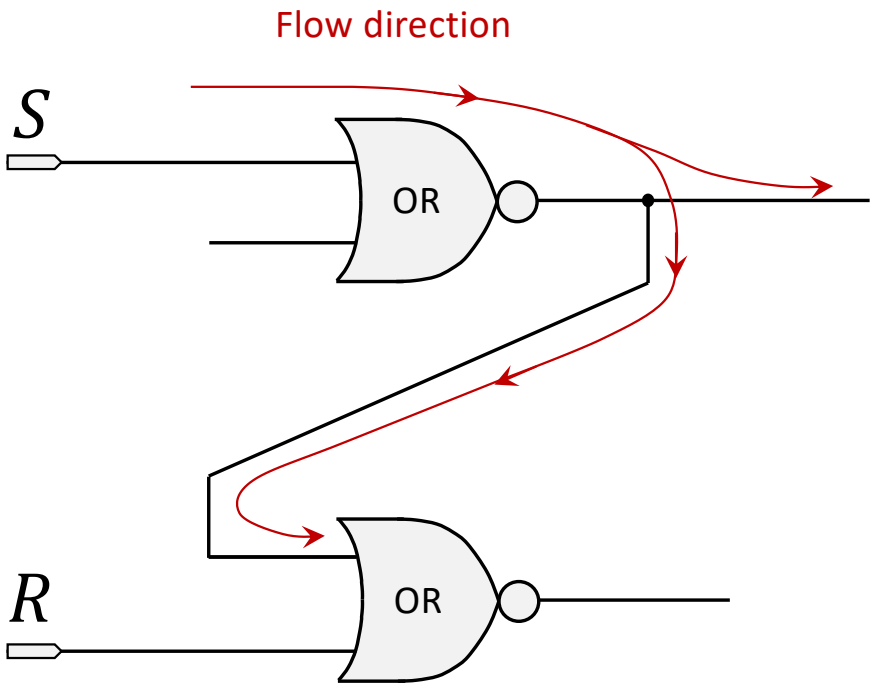
Circuit Construction: Cross-Coupling of NOR Gates



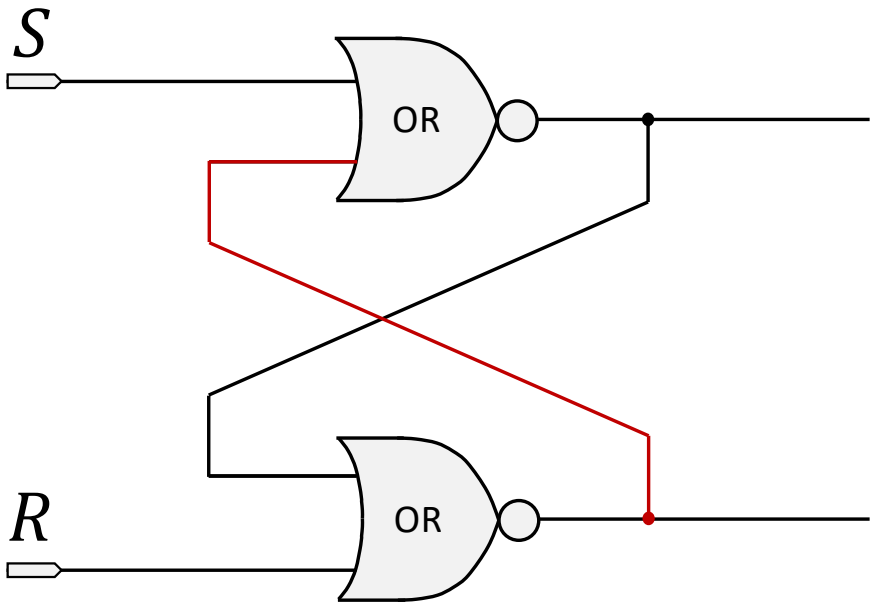
Circuit Construction: Cross-Coupling of NOR Gates



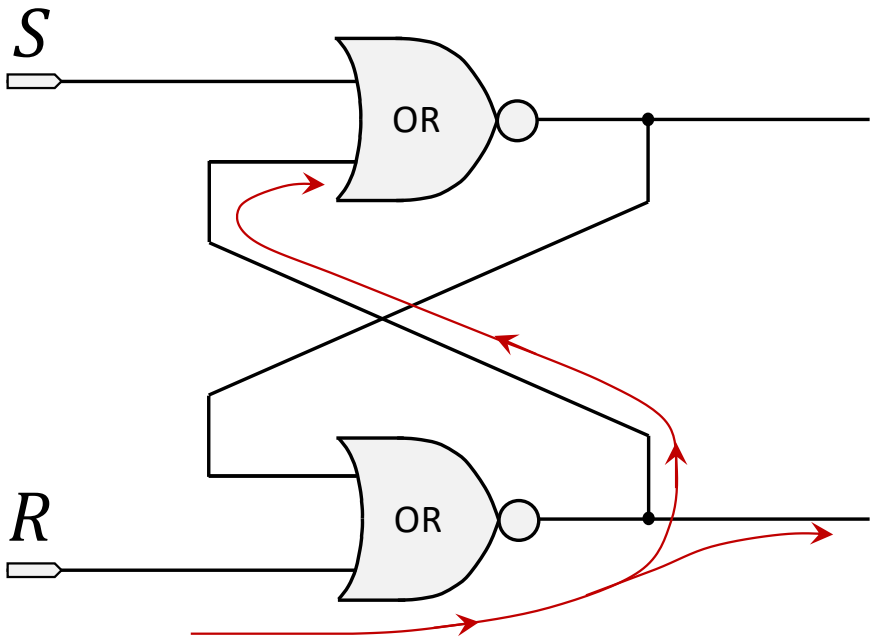
Circuit Construction: Cross-Coupling of NOR Gates



Circuit Construction: Cross-Coupling of NOR Gates

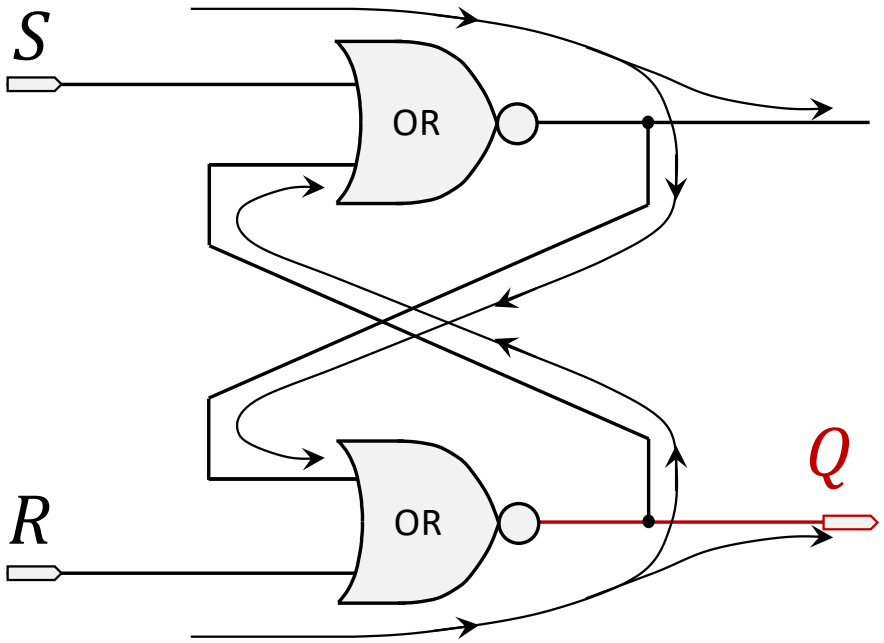


Circuit Construction: Cross-Coupling of NOR Gates

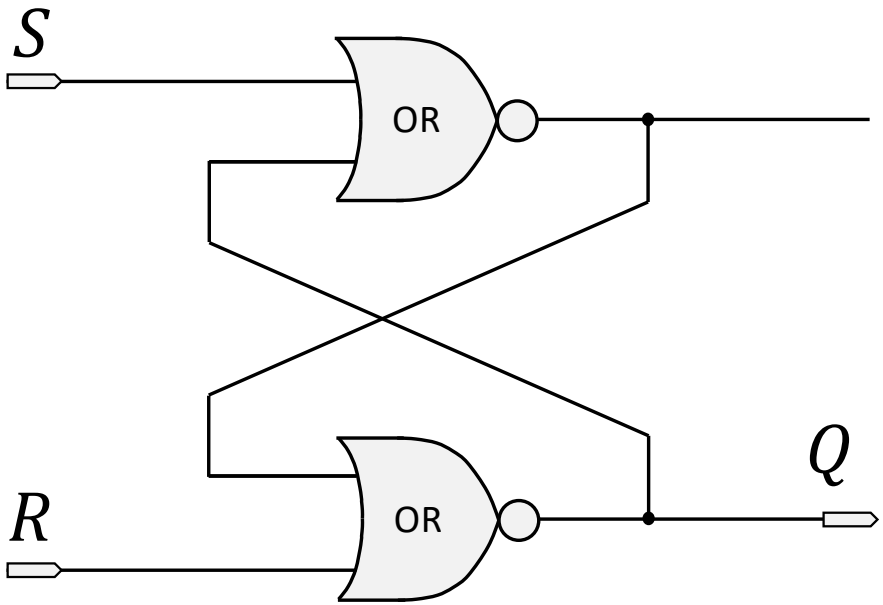


Flow direction

Circuit Construction: Output Pin Q



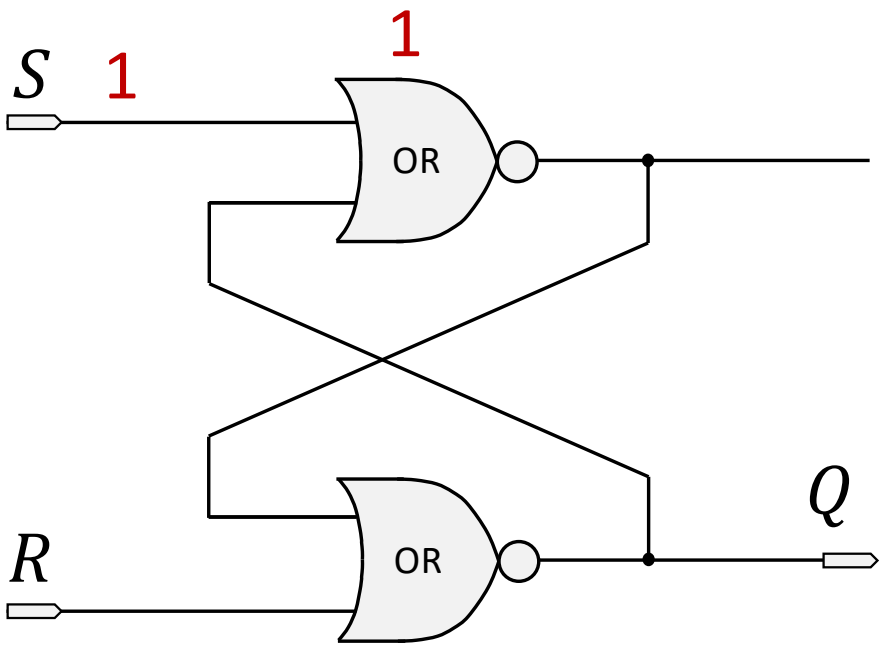
Circuit Construction: Output Pin Q



This pin might lead to nowhere

Circuit Analysis

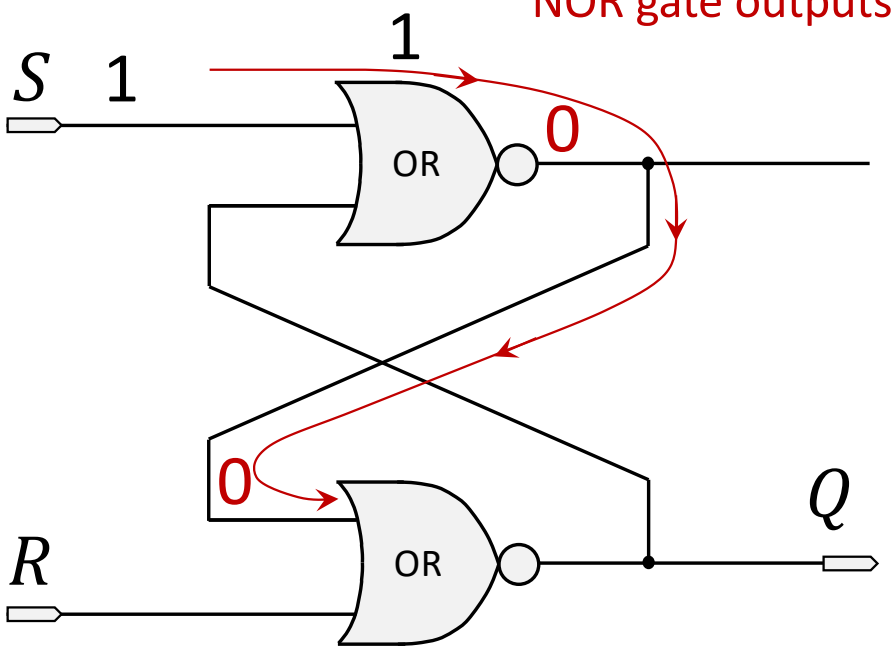
OR gate outputs “1” if at least one input is “1”



Circuit Analysis

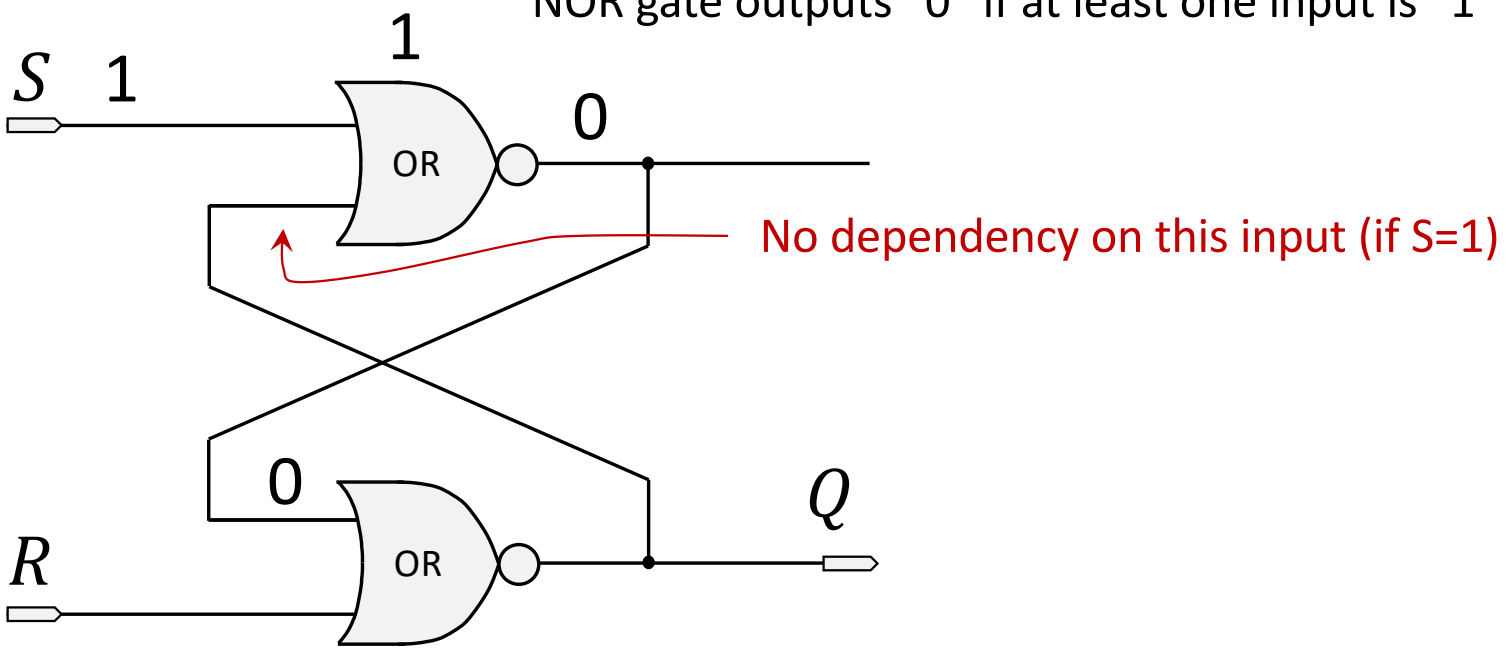
OR gate outputs “1” if at least one input is “1”

NOR gate outputs “0” if at least one input is “1”



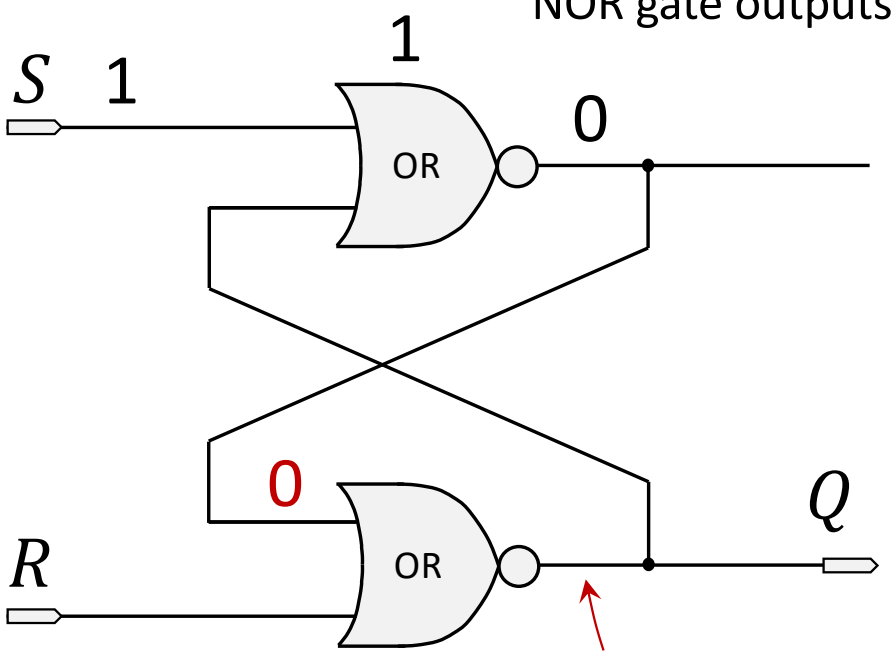
Circuit Analysis

OR gate outputs “1” if at least one input is “1”
NOR gate outputs “0” if at least one input is “1”



Circuit Analysis

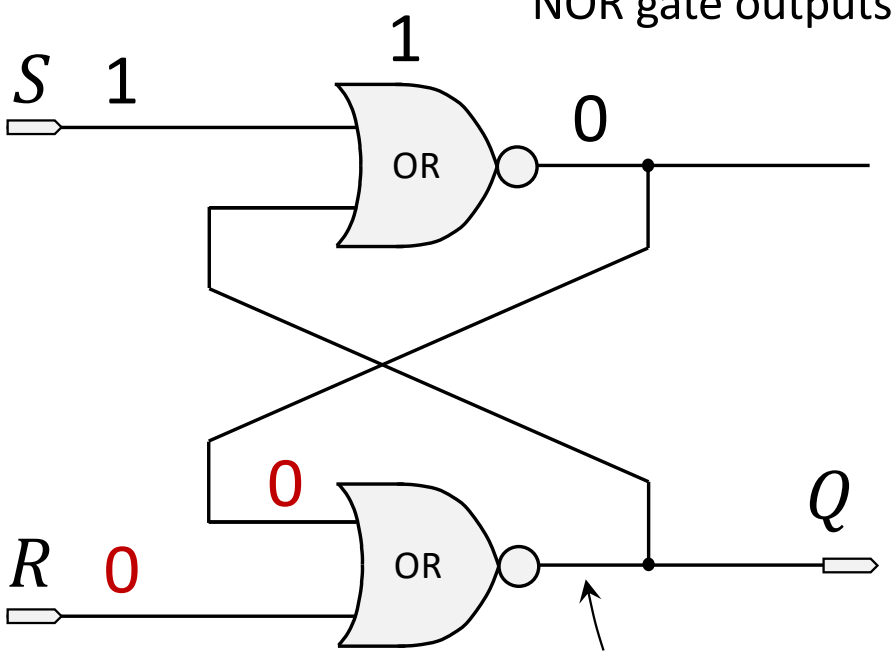
OR gate outputs “1” if at least one input is “1”
NOR gate outputs “0” if at least one input is “1”



This output depends on R value (if S=1)

Circuit Analysis

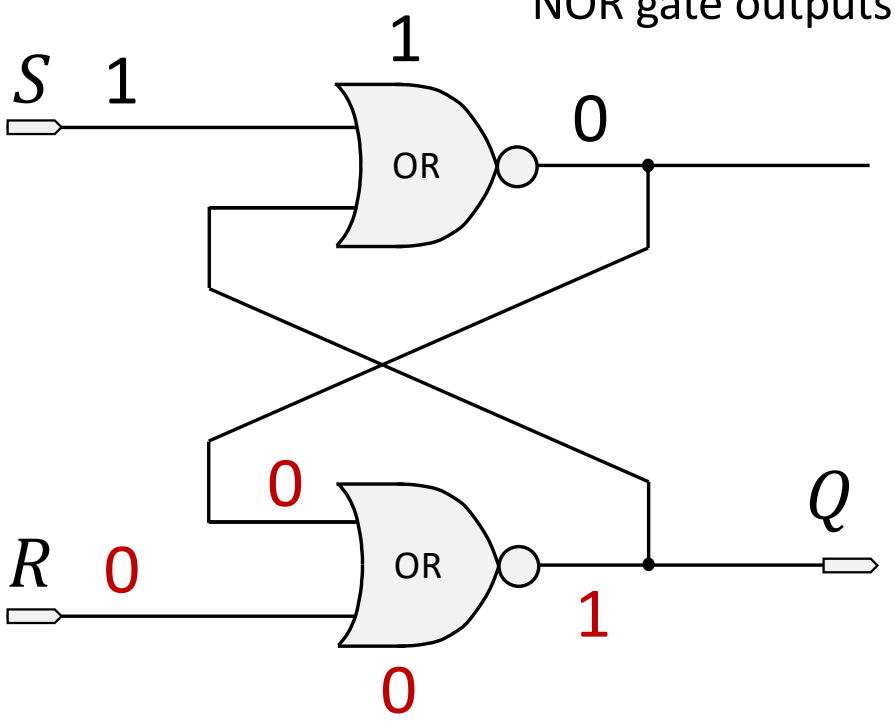
OR gate outputs “1” if at least one input is “1”
NOR gate outputs “0” if at least one input is “1”



This output depends on R value (if S=1)

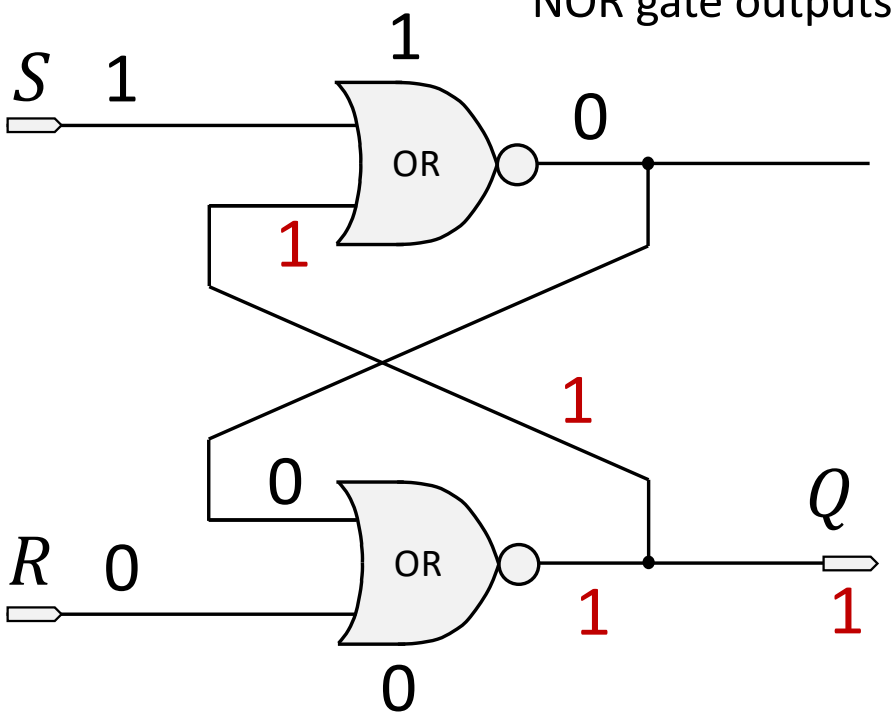
Circuit Analysis

OR gate outputs “1” if at least one input is “1”
NOR gate outputs “0” if at least one input is “1”



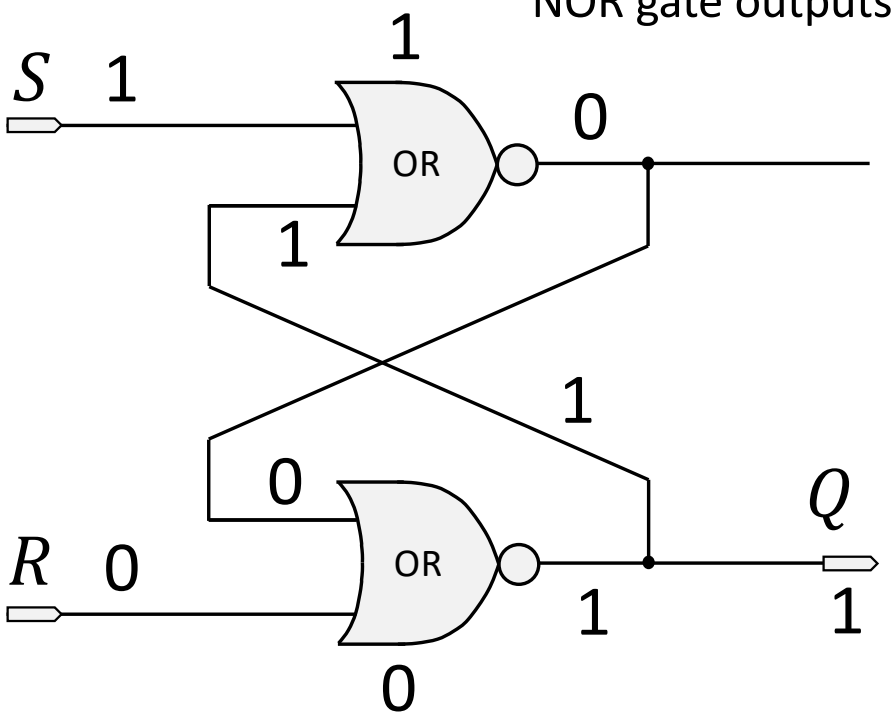
Circuit Analysis

OR gate outputs “1” if at least one input is “1”
NOR gate outputs “0” if at least one input is “1”



Circuit Analysis

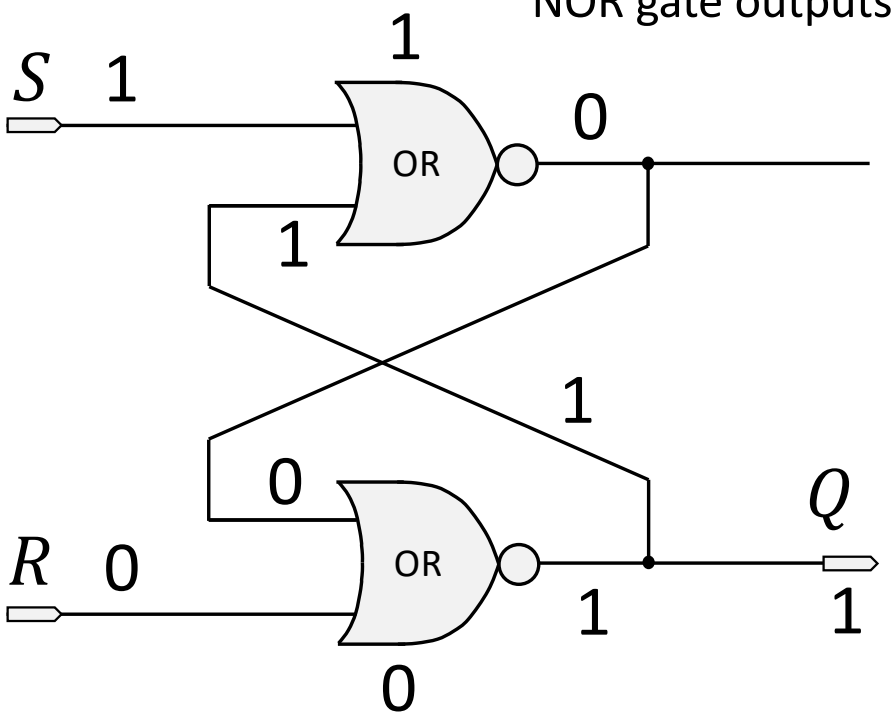
OR gate outputs “1” if at least one input is “1”
NOR gate outputs “0” if at least one input is “1”



Does Q depend on the order of S and R arrival (e.g. S arrives first, and R arrives next) ?

Circuit Analysis

OR gate outputs “1” if at least one input is “1”
NOR gate outputs “0” if at least one input is “1”



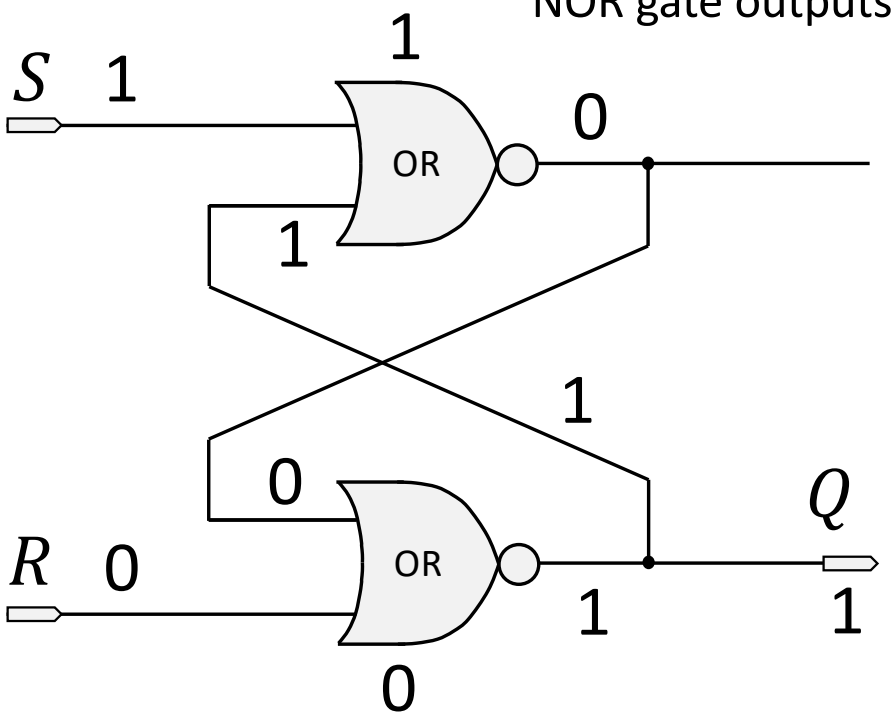
Does Q depend on the order of S and R arrival (e.g. S arrives first, and R arrives next?)

No; you can check it by hand

Circuit Analysis

Assume that S changes from “1” to “0”

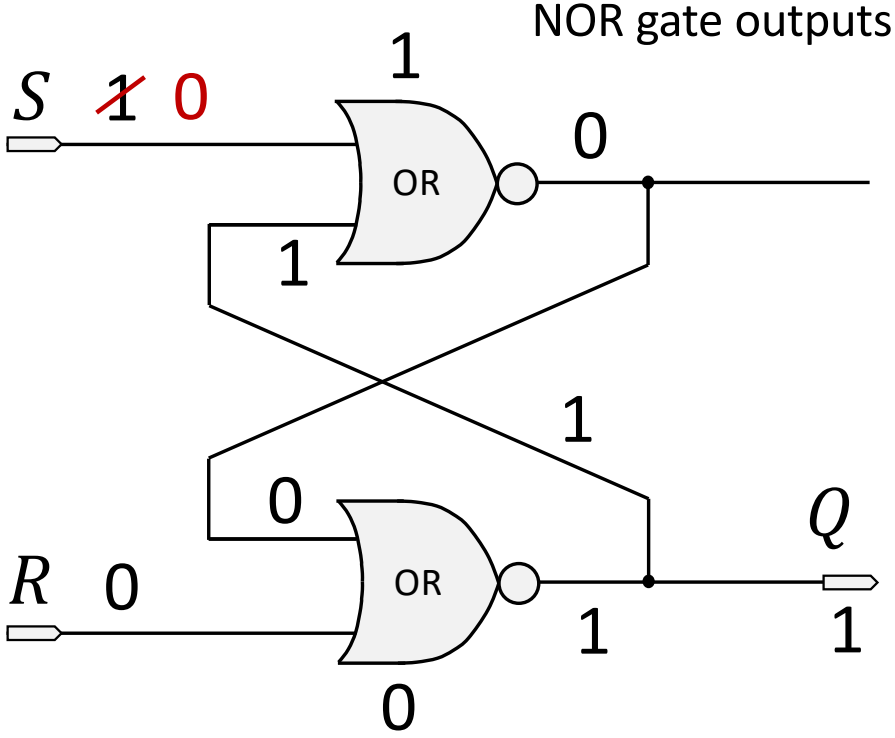
NOR gate outputs “0” if at least one input is “1”



Circuit Analysis

Assume that S changes from “1” to “0”

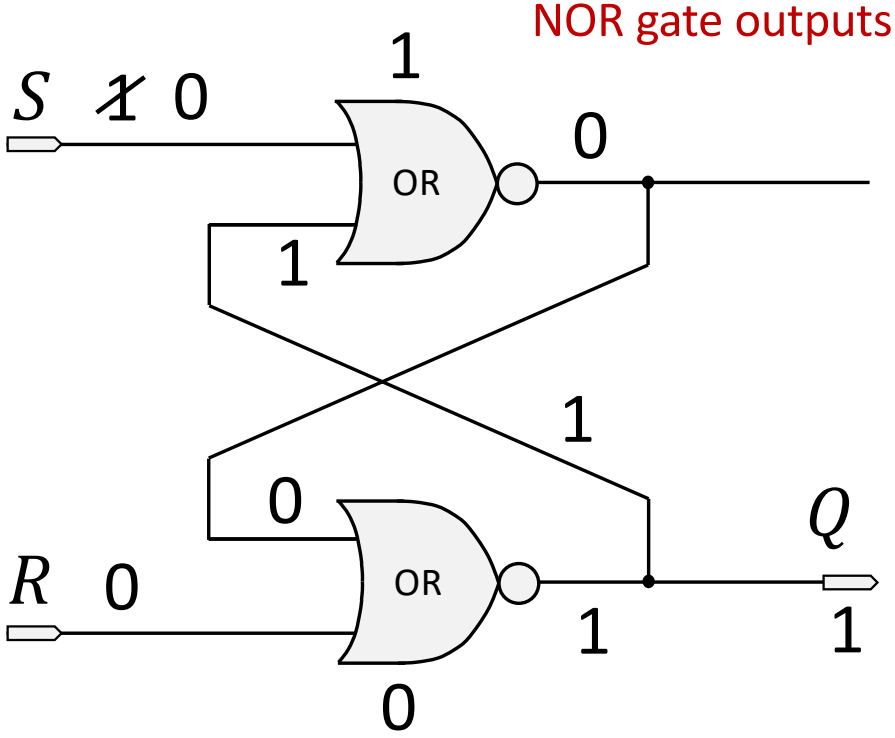
NOR gate outputs “0” if at least one input is “1”



Circuit Analysis

Assume that S changes from “1” to “0”

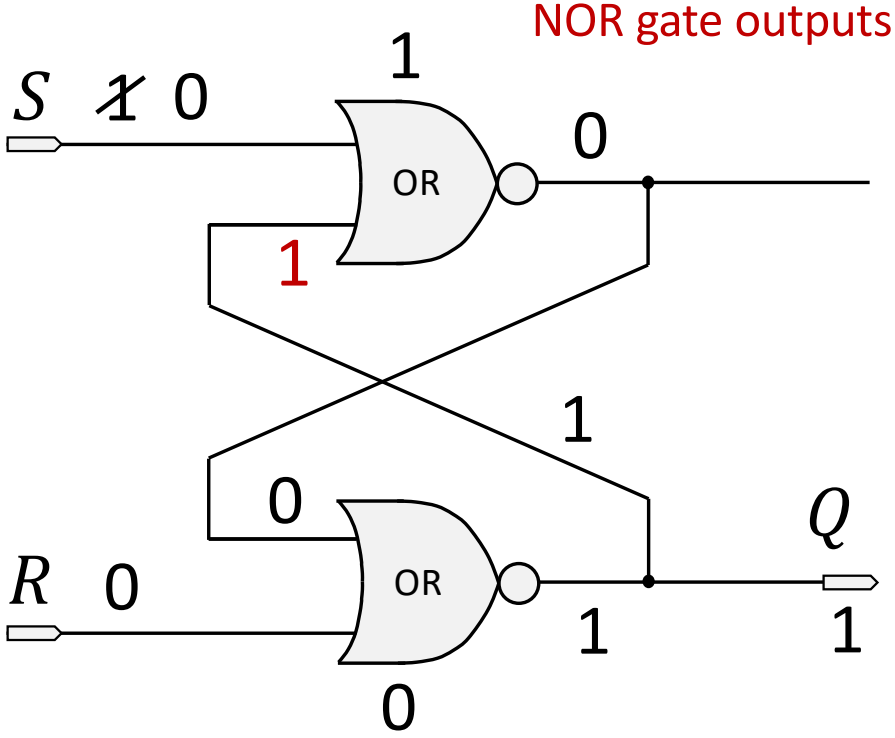
NOR gate outputs “0” if at least one input is “1”



Circuit Analysis

Assume that S changes from “1” to “0”

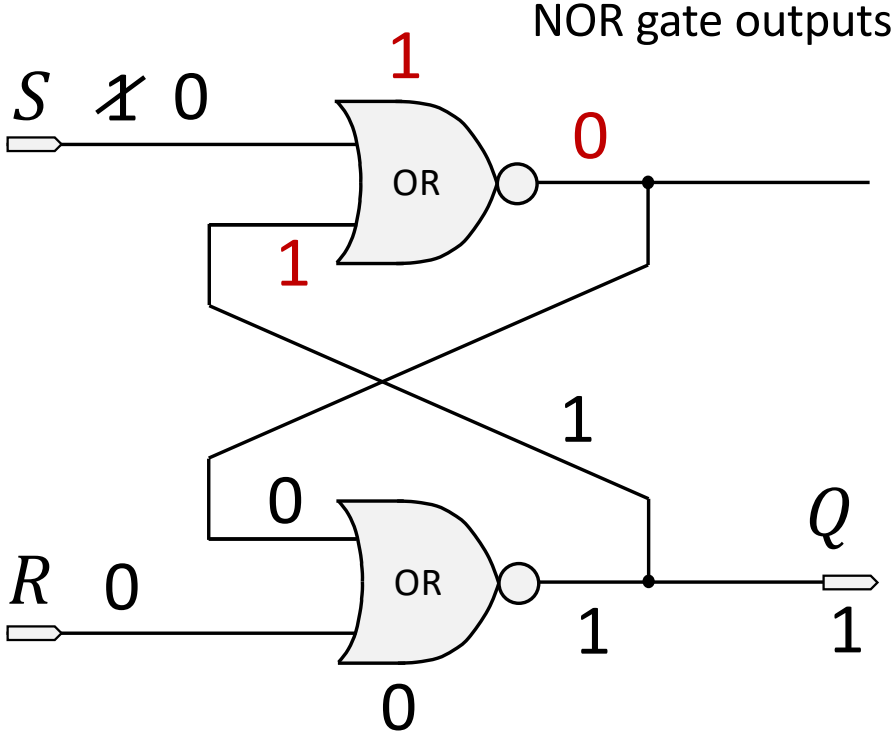
NOR gate outputs “0” if at least one input is “1”



Circuit Analysis

Assume that S changes from “1” to “0”

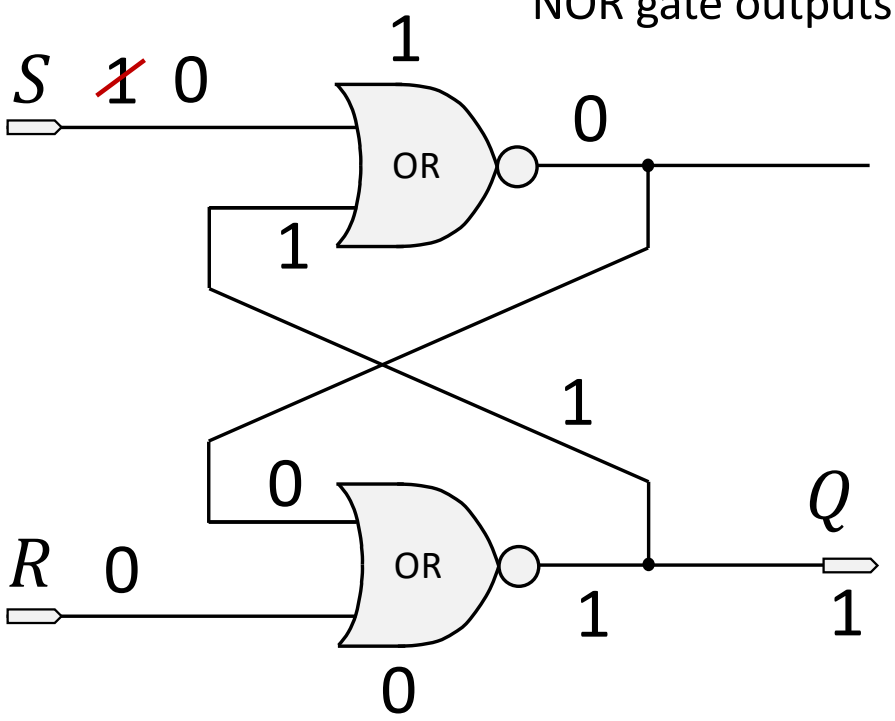
NOR gate outputs “0” if at least one input is “1”



Circuit Analysis

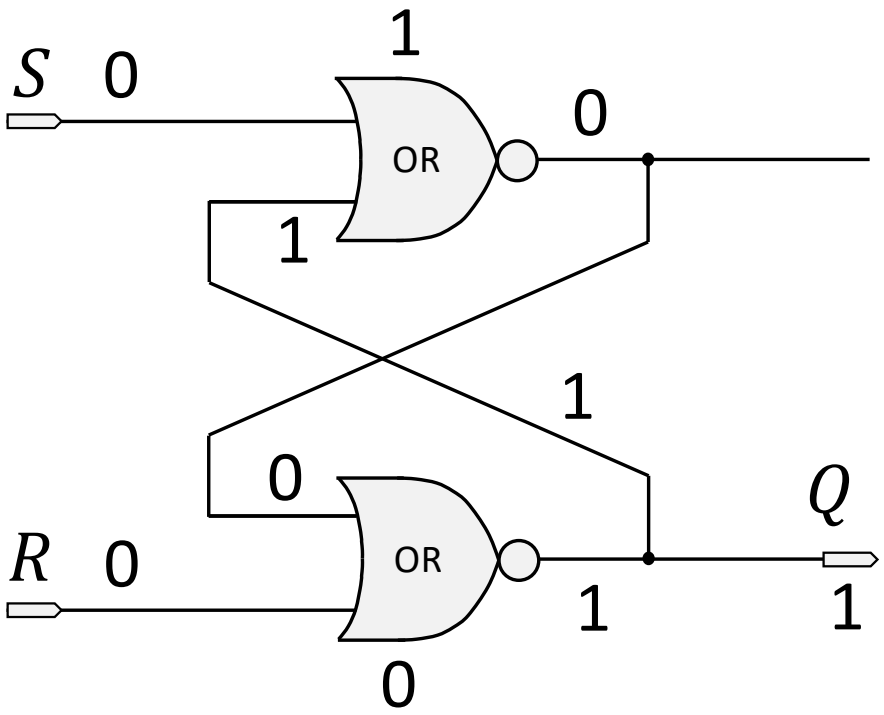
Assume that S changes from “1” to “0”

NOR gate outputs “0” if at least one input is “1”



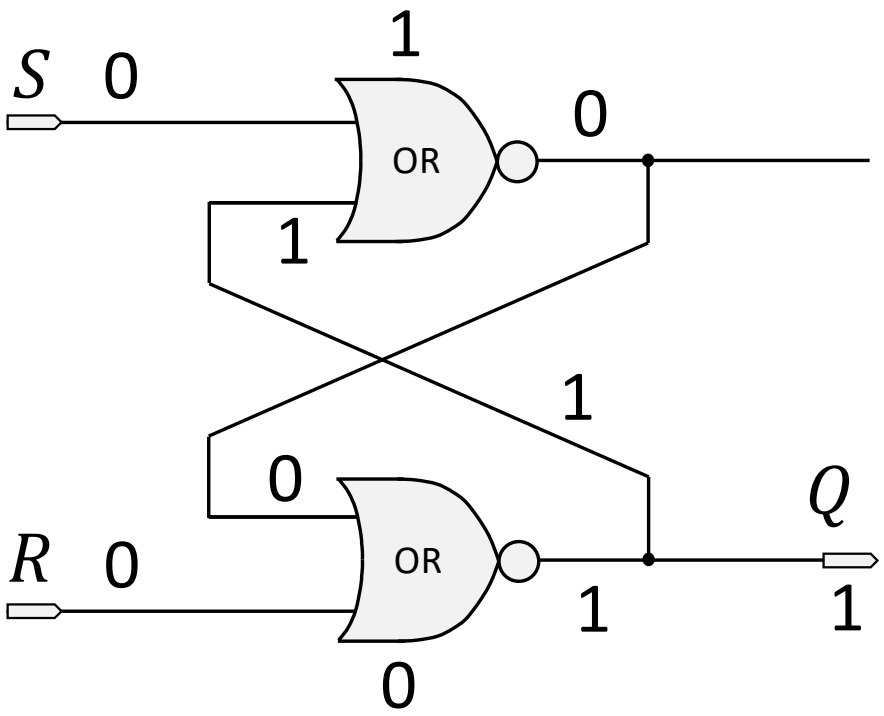
Q does not change and remains “1”

Circuit Analysis



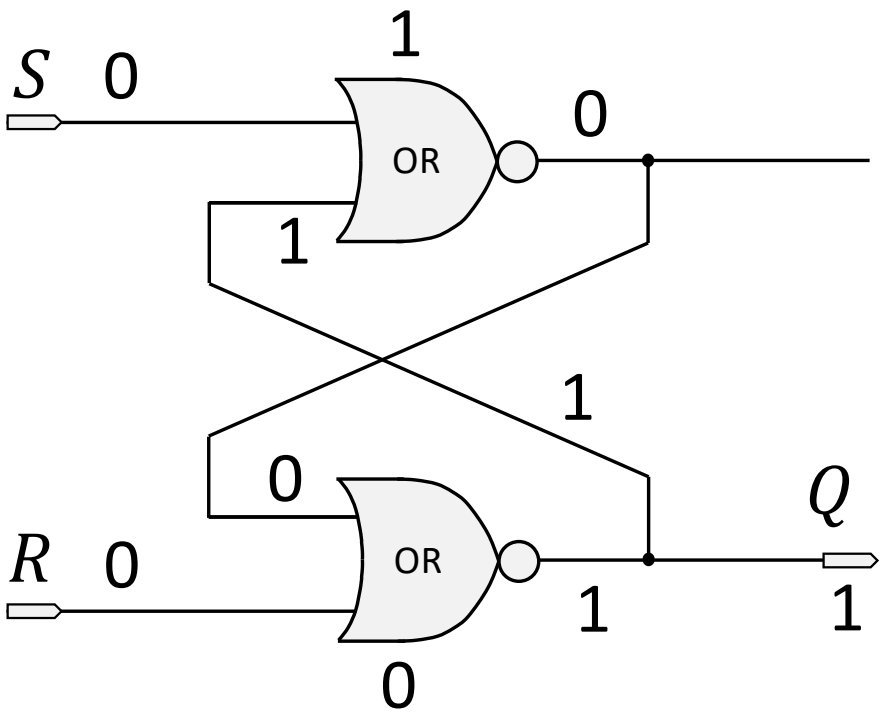
Circuit Analysis

Next, assume that R changes from “0” to “1”



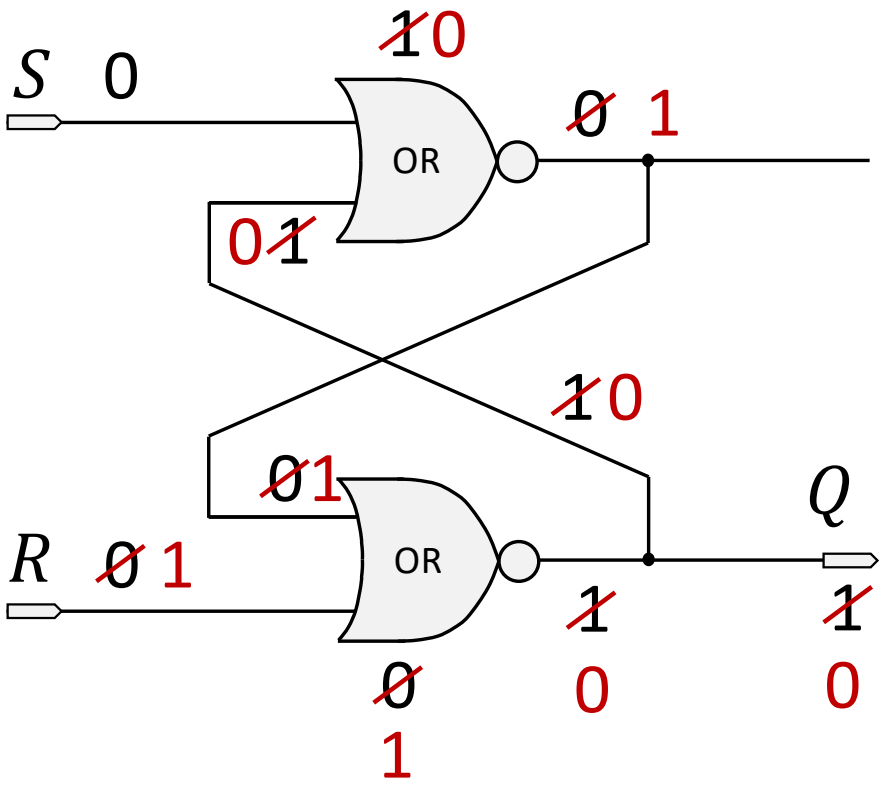
Circuit Analysis

Next, assume that R changes from “0” to “1”



Circuit Analysis

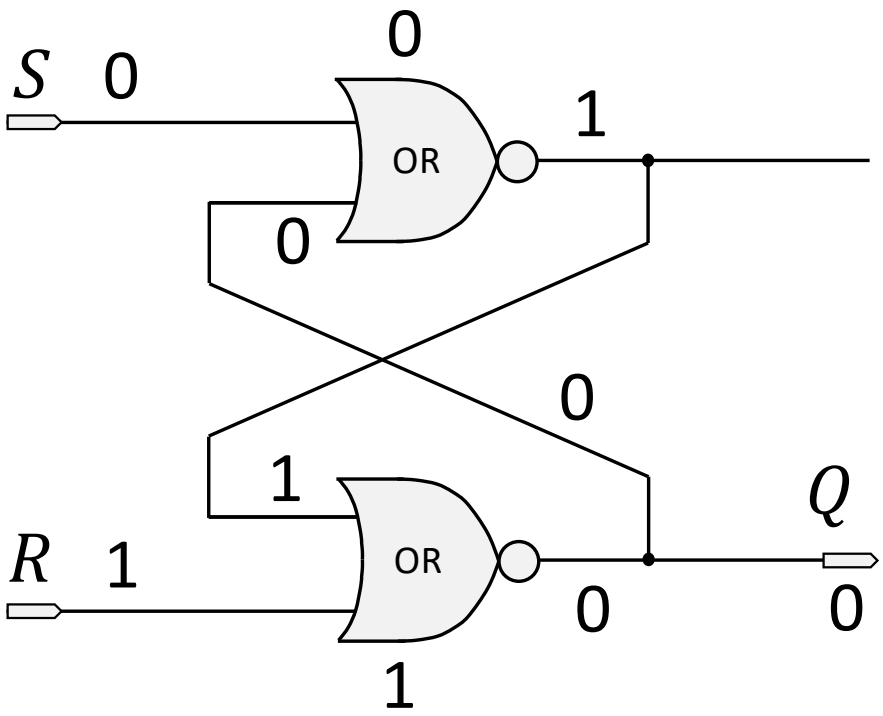
Next, assume that R changes from “0” to “1”



Q changes to “0”

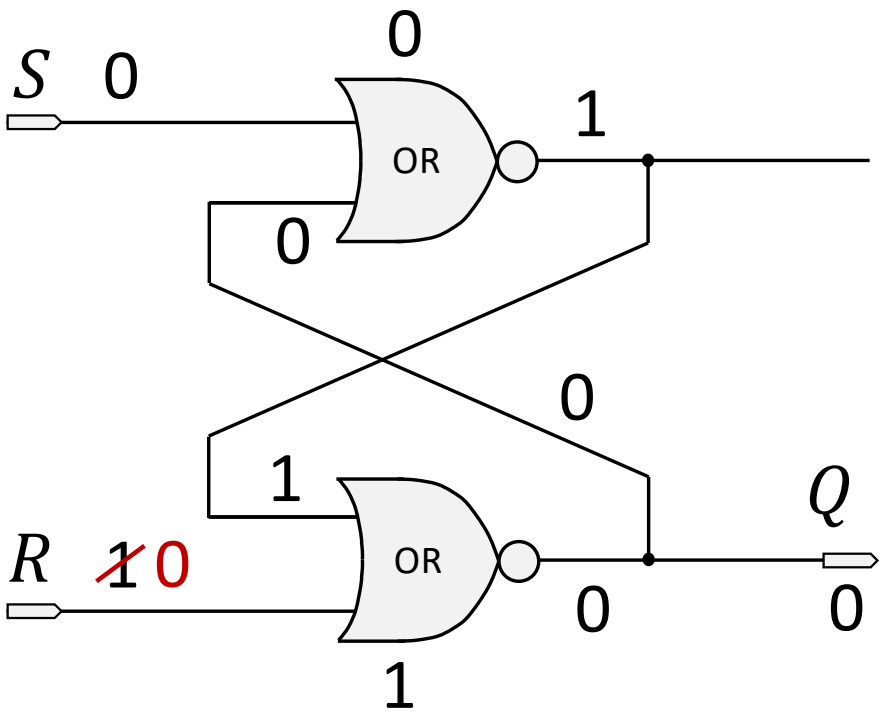
Circuit Analysis

Finally, let R to change from “1” back to “0”



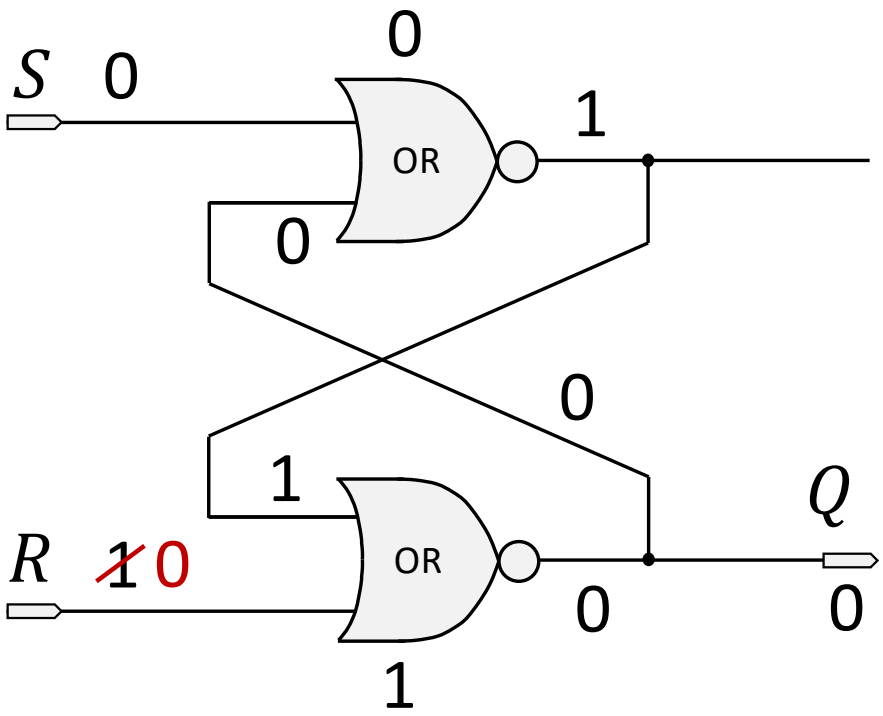
Circuit Analysis

Finally, let R to change from “1” back to “0”



Circuit Analysis

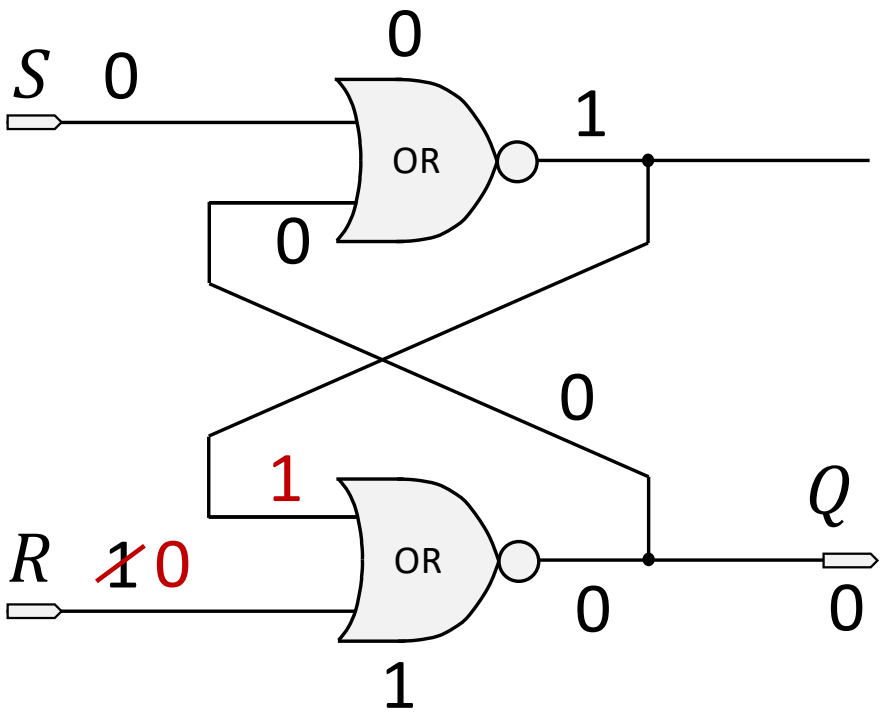
Finally, let R to change from “1” back to “0”



NOR gate outputs “0” if at least one input is “1”

Circuit Analysis

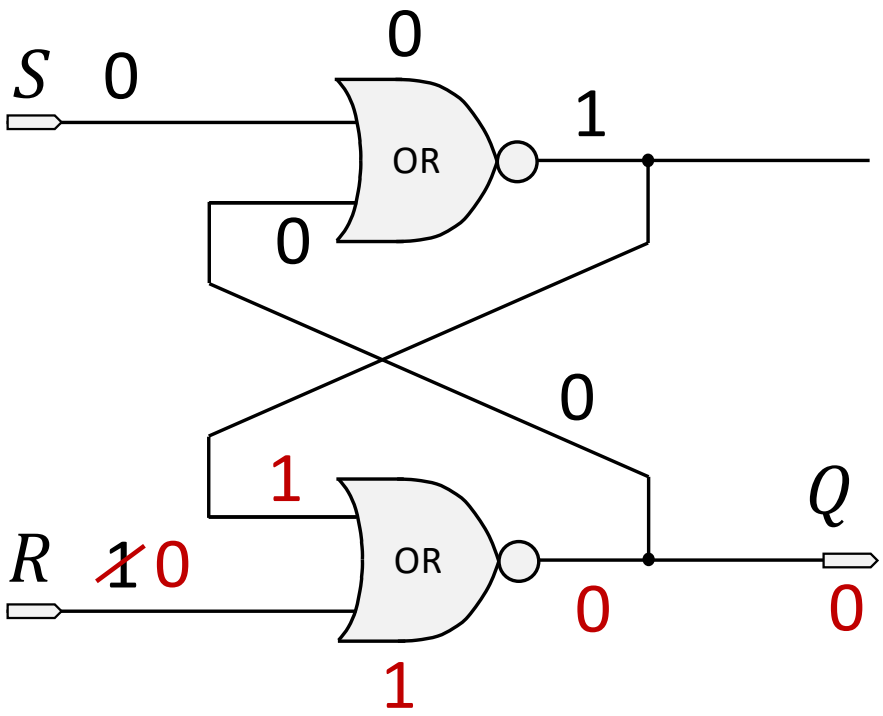
Finally, let R to change from “1” back to “0”



NOR gate outputs “0” if at least one input is “1”

Circuit Analysis

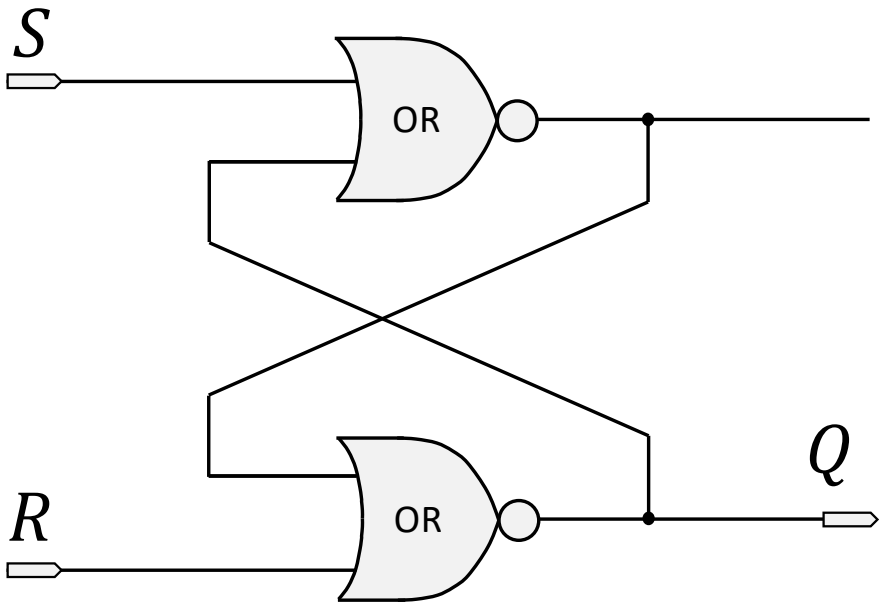
Finally, let R to change from “1” back to “0”



Q does not change

Summary

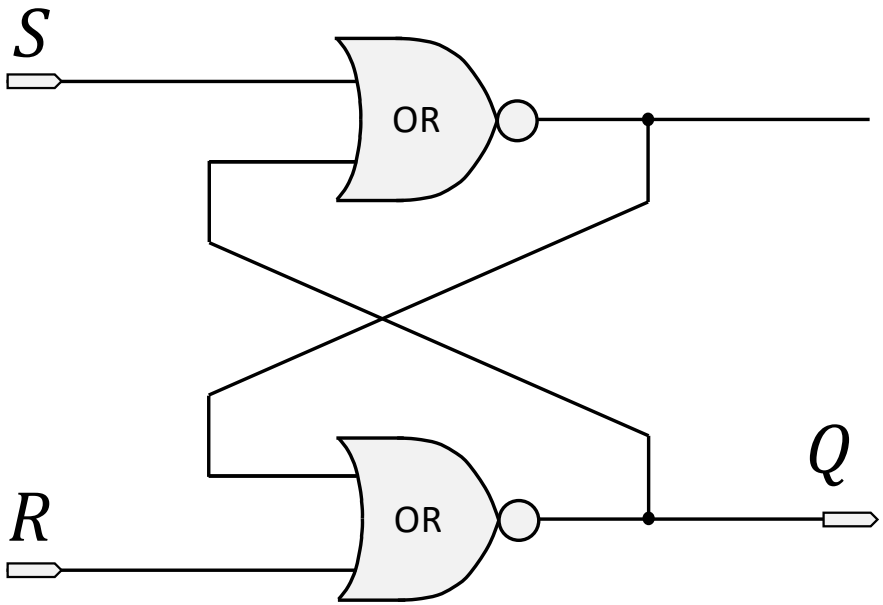
$S = 1, R = 0$: $Q = 1$

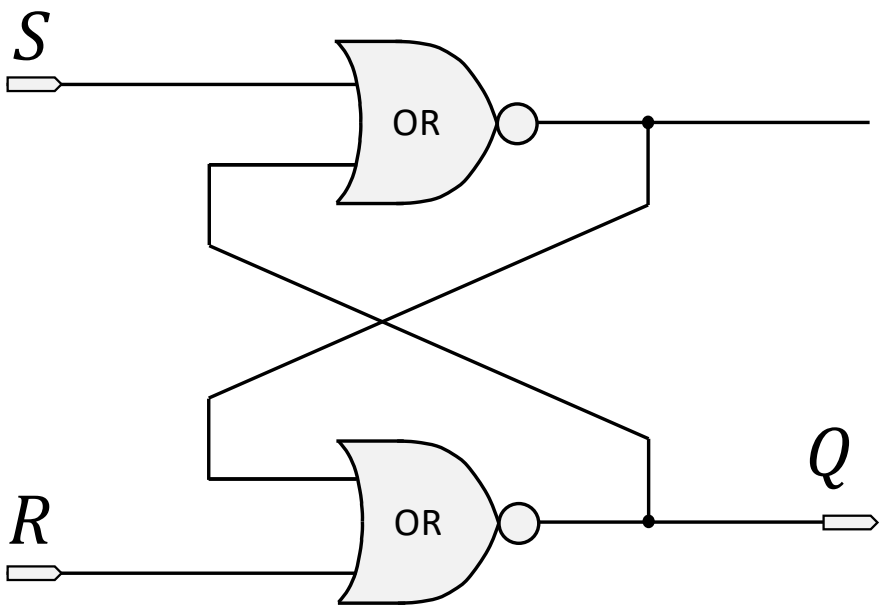


Summary

$S = 1, R = 0$: $Q = 1$

$S = 0, R = 1$: $Q = 0$





Summary

$S = 1, R = 0$: $Q = 1$

$S = 0, R = 1$: $Q = 0$

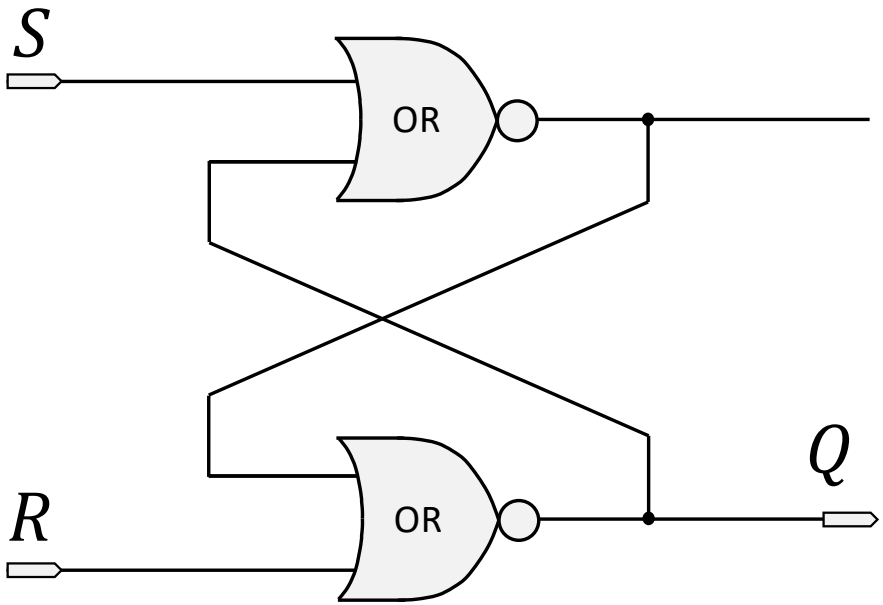
$S = 0, R = 0$: Q is stored by circuit

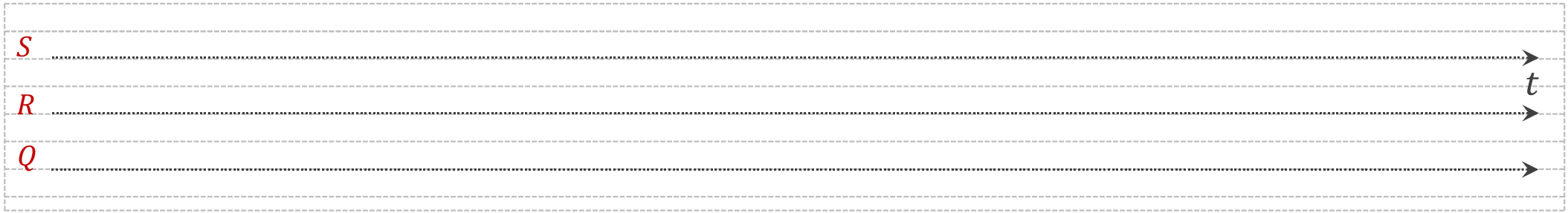
Summary

$S = 1, R = 0$: $Q = 1$

$S = 0, R = 1$: $Q = 0$

$S = 0, R = 0$: Q is stored by circuit



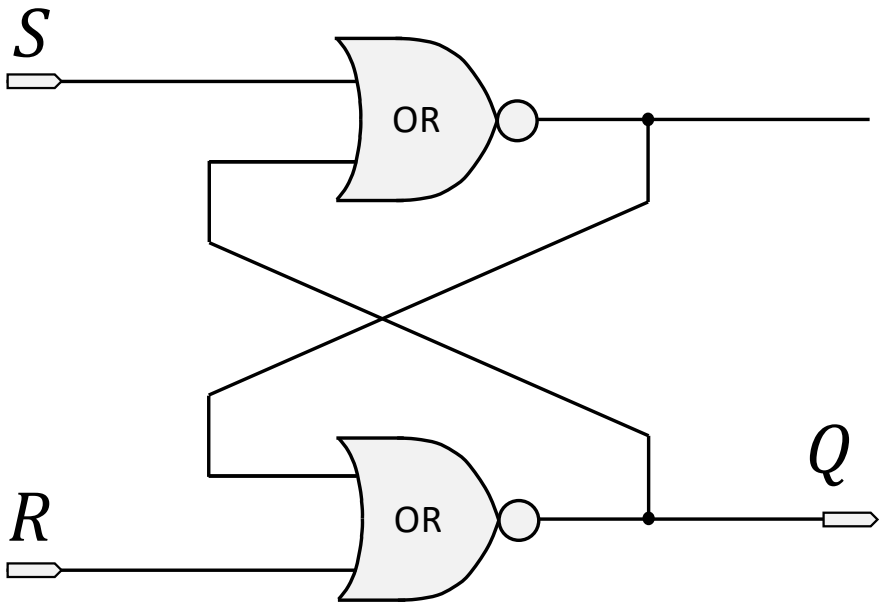


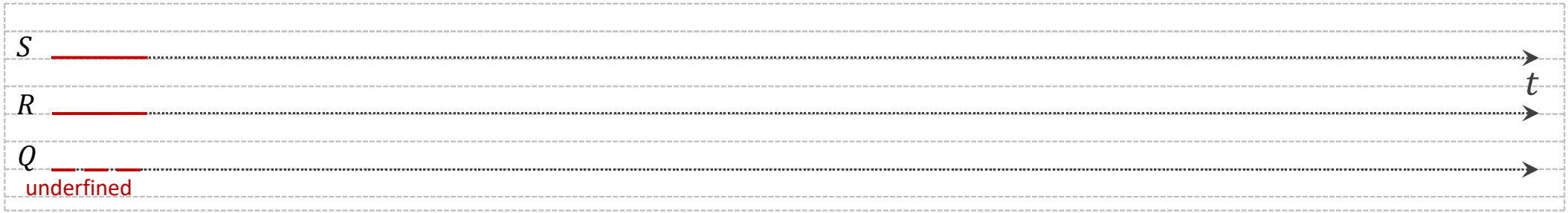
Summary

$S = 1, R = 0$: $Q = 1$

$S = 0, R = 1$: $Q = 0$

$S = 0, R = 0$: Q is stored by circuit



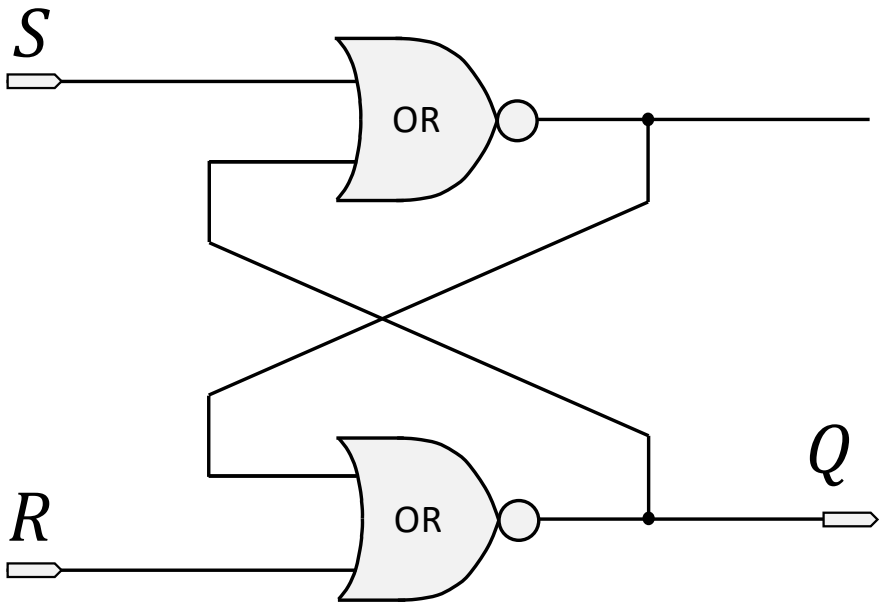


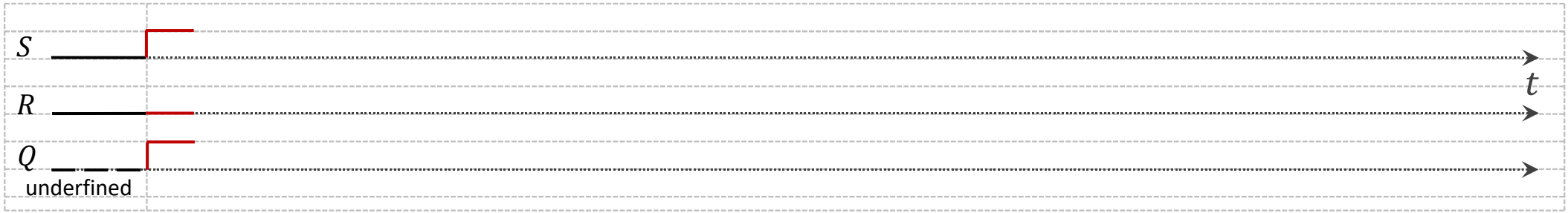
Summary

$S = 1, R = 0$: $Q = 1$

$S = 0, R = 1$: $Q = 0$

$S = 0, R = 0$: Q is stored by circuit



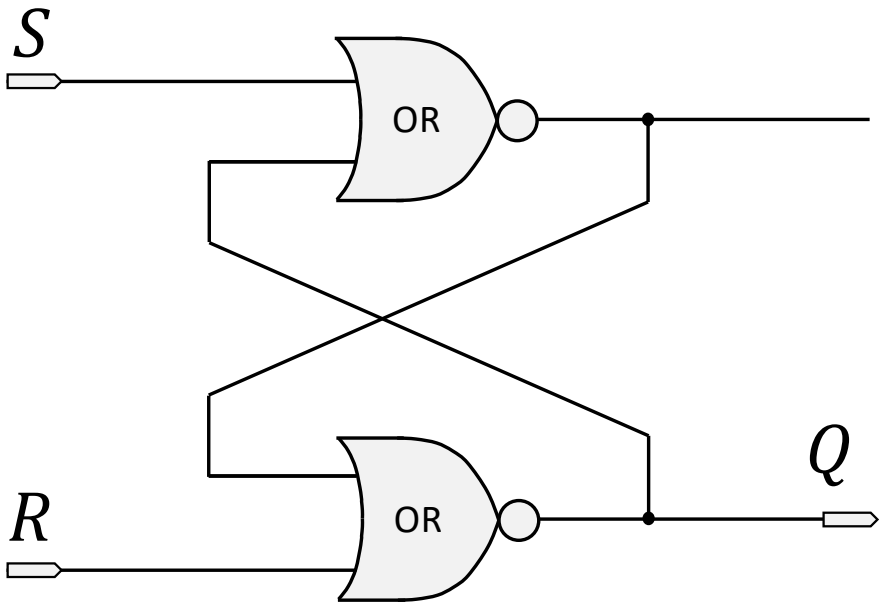


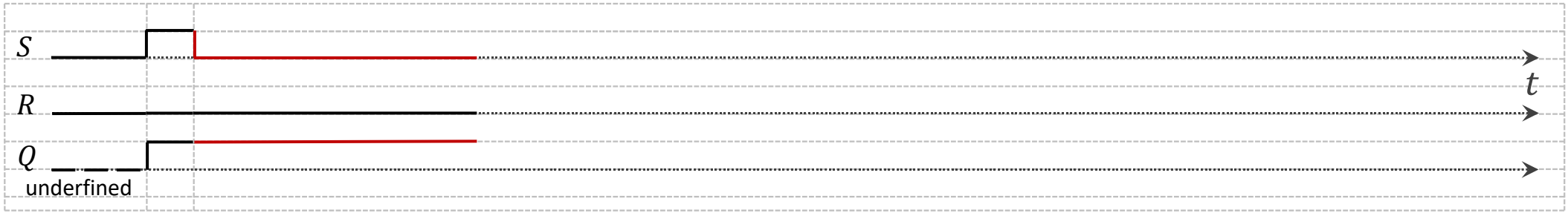
Summary

$S = 1, R = 0$: $Q = 1$

$S = 0, R = 1$: $Q = 0$

$S = 0, R = 0$: Q is stored by circuit



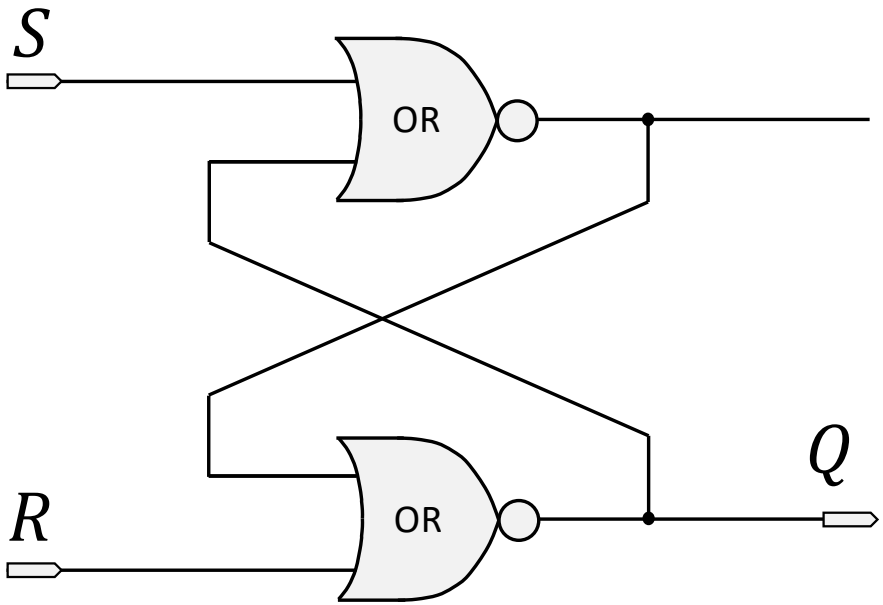


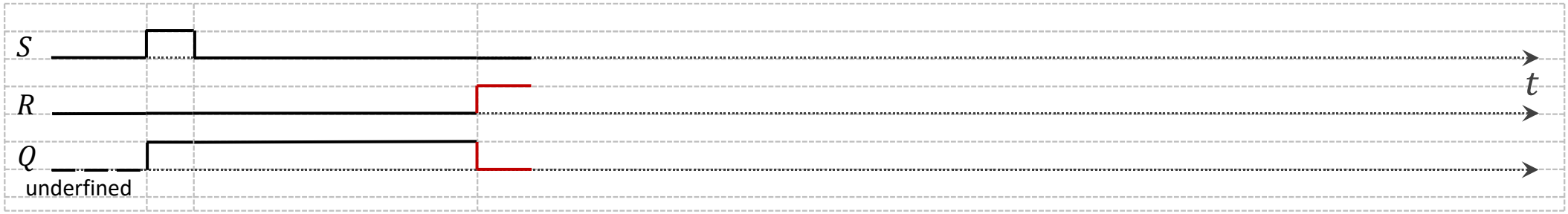
Summary

$S = 1, R = 0$: $Q = 1$

$S = 0, R = 1$: $Q = 0$

$S = 0, R = 0$: Q is stored by circuit



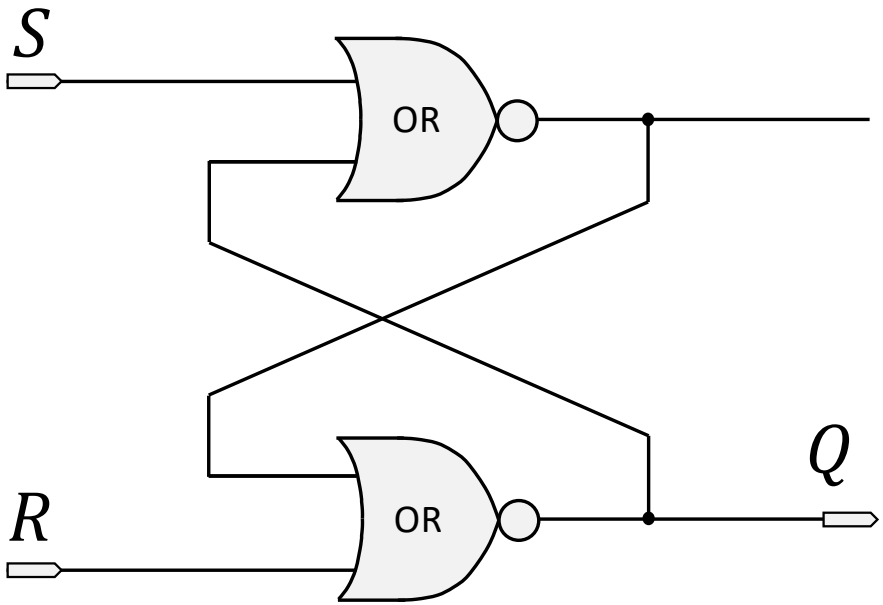


Summary

$S = 1, R = 0$: $Q = 1$

$S = 0, R = 1$: $Q = 0$

$S = 0, R = 0$: Q is stored by circuit

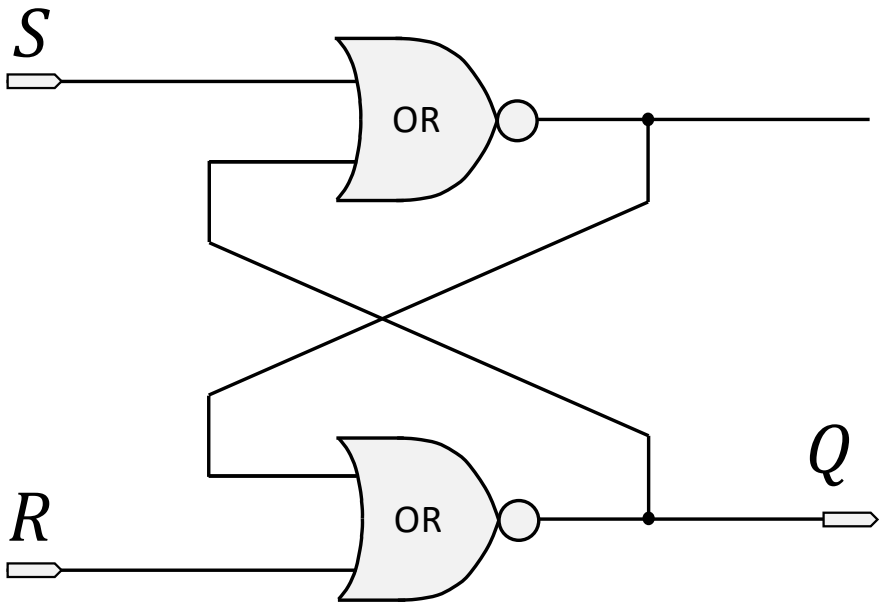


Summary

$S = 1, R = 0$: $Q = 1$

$S = 0, R = 1$: $Q = 0$

$S = 0, R = 0$: Q is stored by circuit

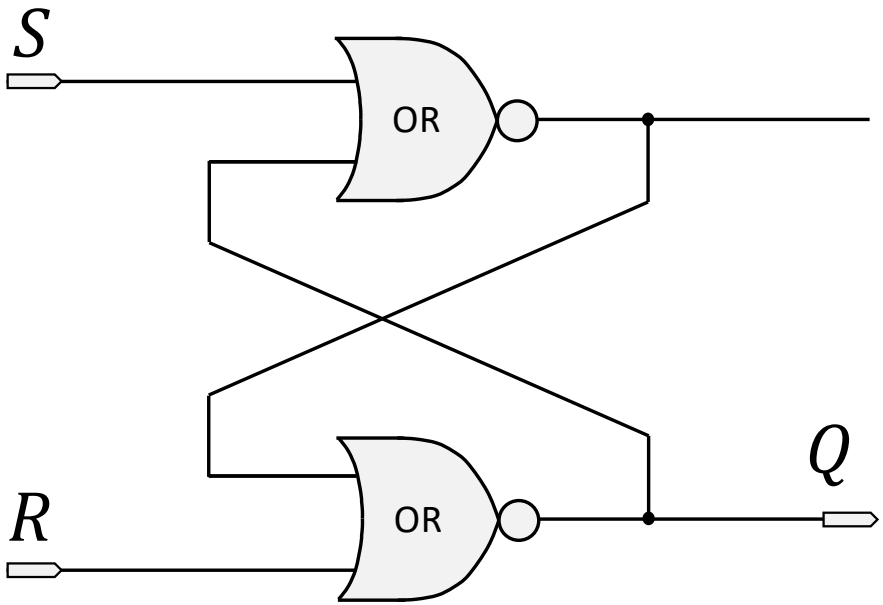


Summary

$S = 1, R = 0$: $Q = 1$

$S = 0, R = 1$: $Q = 0$

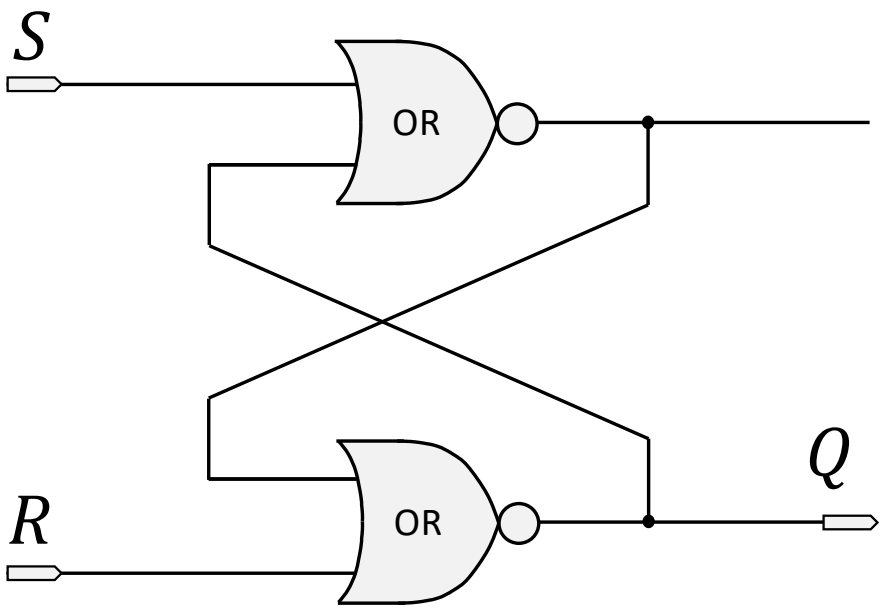
$S = 0, R = 0$: Q is stored by circuit



The value of Q is stored by a circuit for some time
(due to complex physics processes)

S/R Latch Circuit

Stores 1 bit of information

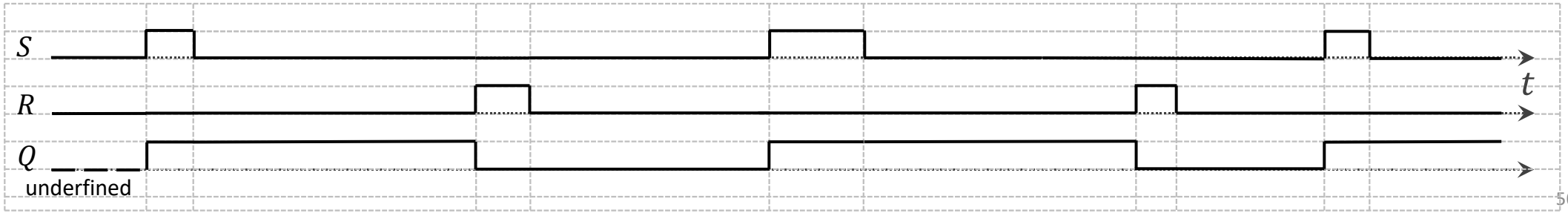


$S = 1, R = 0$: $Q = 1$

$S = 0, R = 1$: $Q = 0$

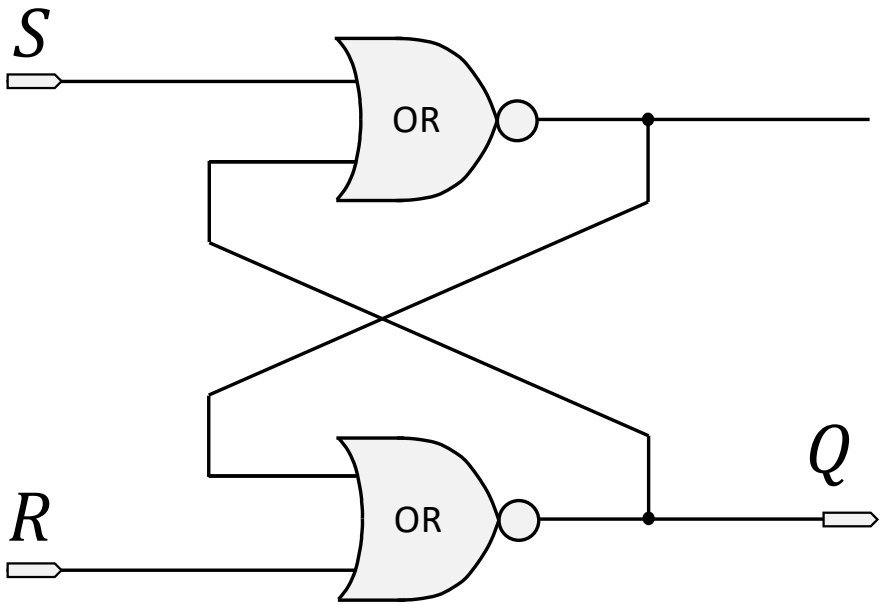
$S = 0, R = 0$: Q is stored by circuit

The value of Q is stored by a circuit for some time
(due to complex physics processes)



S/R Latch Circuit

Stores 1 bit of information



$S = 1, R = 0$: $Q = 1$

$S = 0, R = 1$: $Q = 0$

$S = 0, R = 0$: Q is stored by circuit

Interpretation:

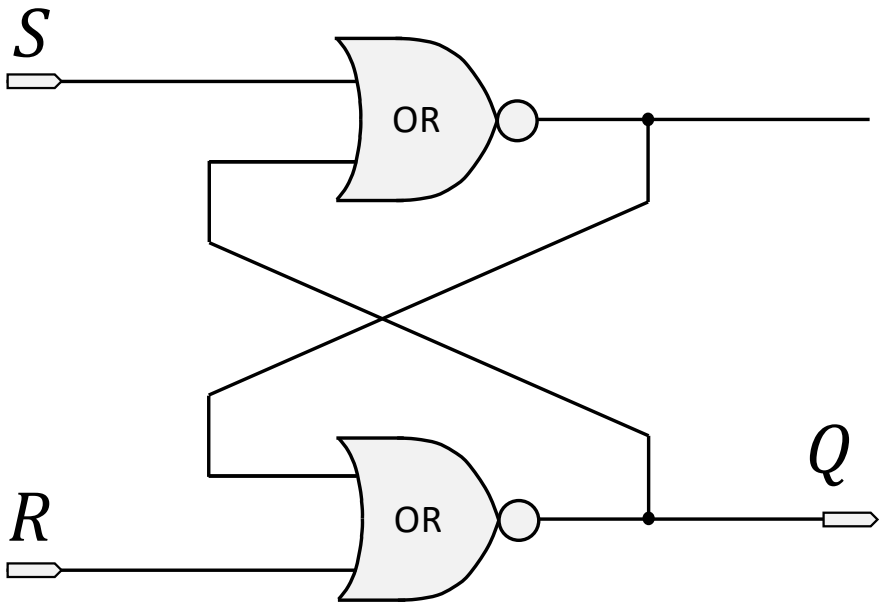
S – "set" pin

R – "reset" pin

Q – stored value

S/R Latch Circuit

Stores 1 bit of information



$S = 1, R = 0$: $Q = 1$

$S = 0, R = 1$: $Q = 0$

$S = 0, R = 0$: Q is stored by circuit

$S = 1, R = 1$: An illegal combination

Interpretation:

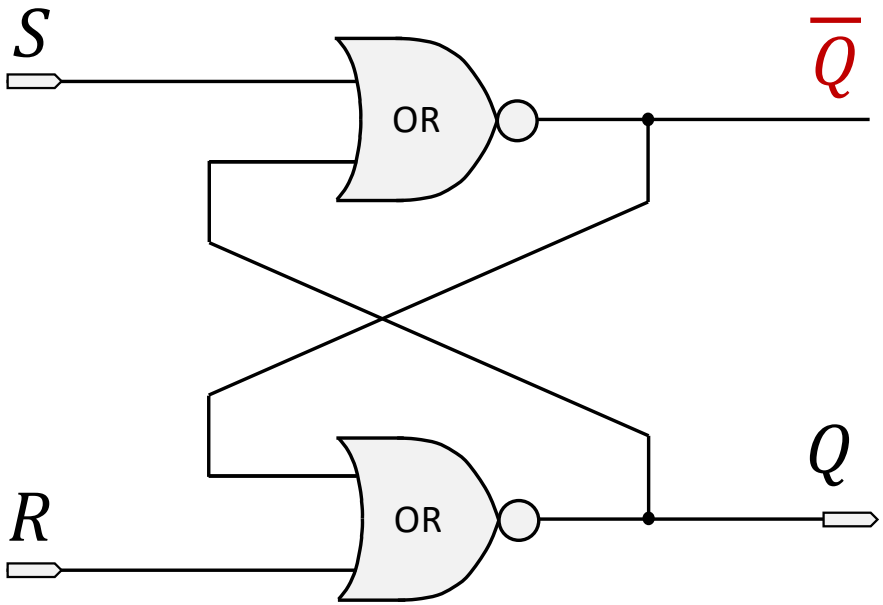
S – "set" pin

R – "reset" pin

Q – stored value

S/R Latch Circuit

Stores 1 bit of information



$S = 1, R = 0$: $Q = 1$

$S = 0, R = 1$: $Q = 0$

$S = 0, R = 0$: Q is stored by circuit

$S = 1, R = 1$: An illegal combination

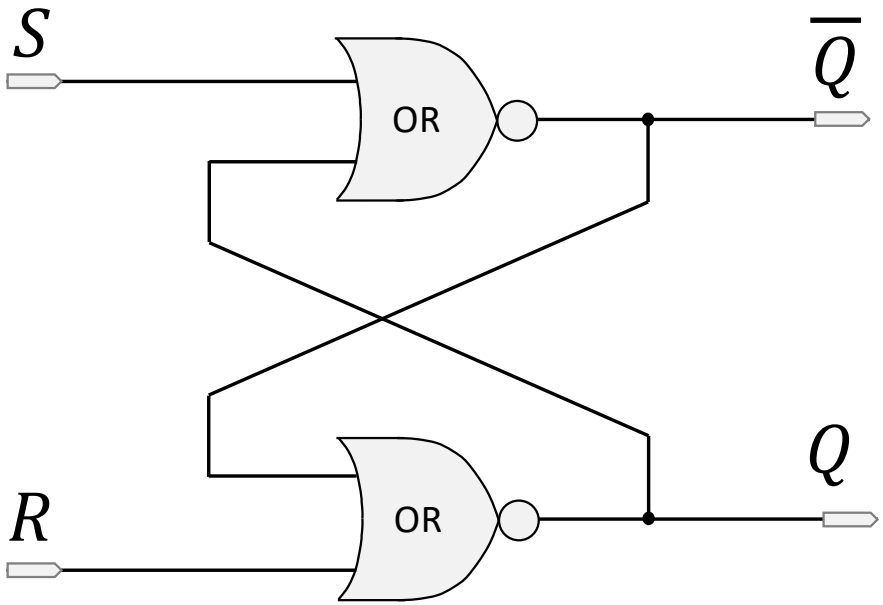
Interpretation:

S – "set" pin

R – "reset" pin

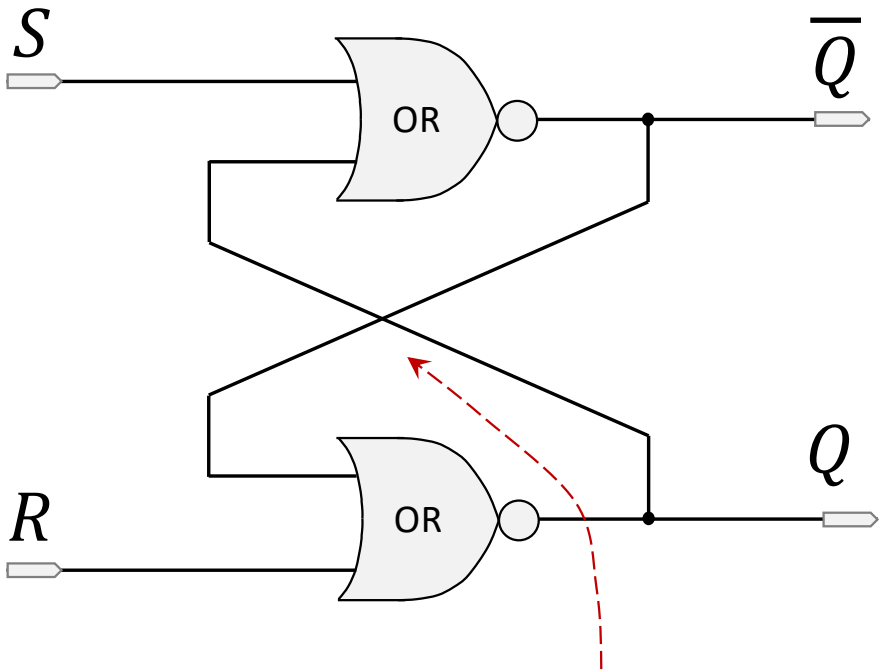
Q – stored value

S/R Latch: Truth Table



S	R	Q
1	0	1
0	1	0
0	0	Q^{prev}
1	1	Illegal inputs

S/R Latch: Truth Table

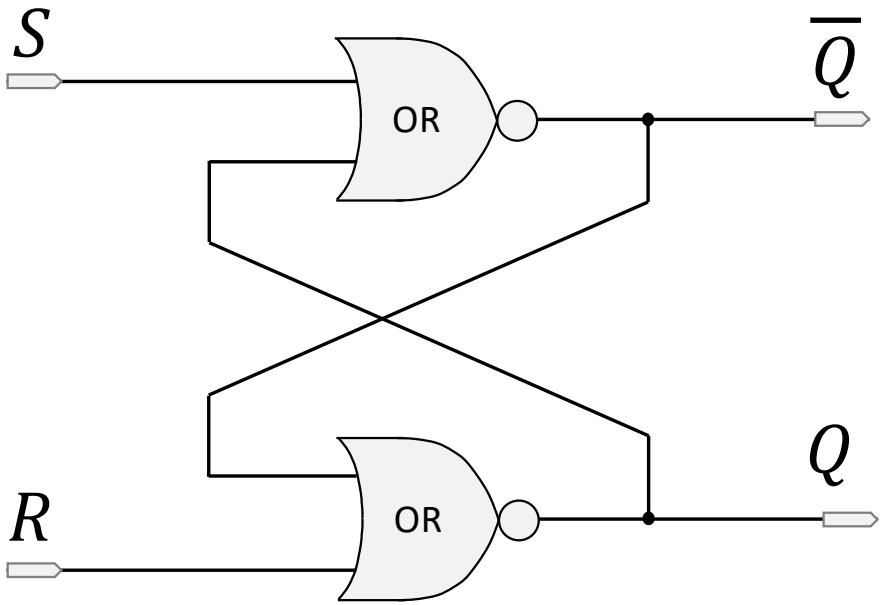


S	R	Q
1	0	1
0	1	0
0	0	Q^{prev}
1	1	Illegal inputs

Cross-coupled connection – the key feature of latches:
the output of one gate serves as an input to another gate

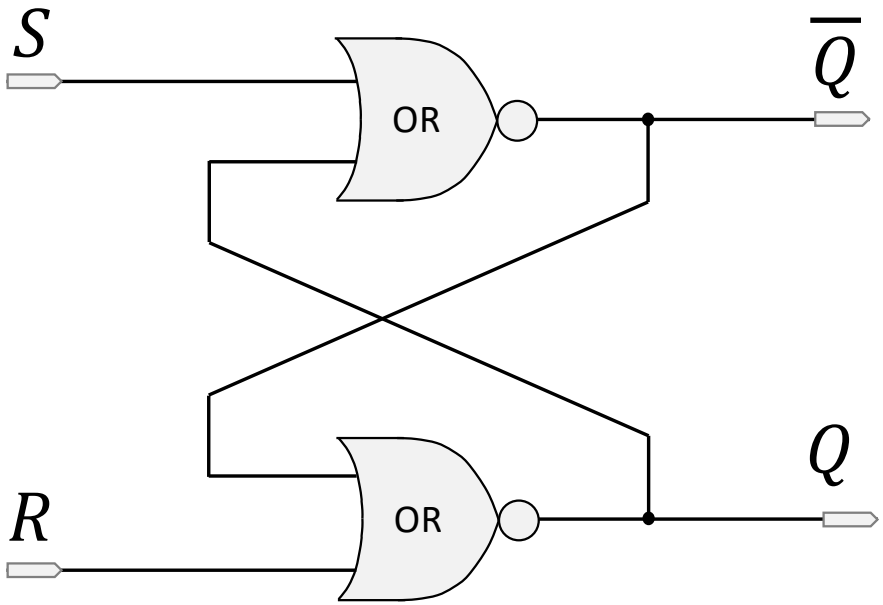
S/R Latch: Multiple Implementations Available

1) Implementation by using NOR logic gates:

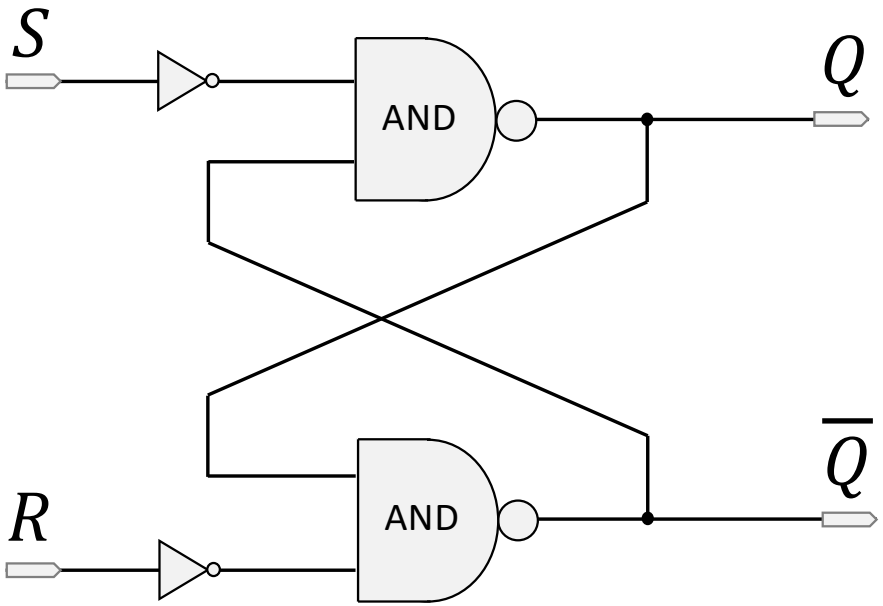


S/R Latch: Multiple Implementations Available

1) Implementation by using NOR logic gates:

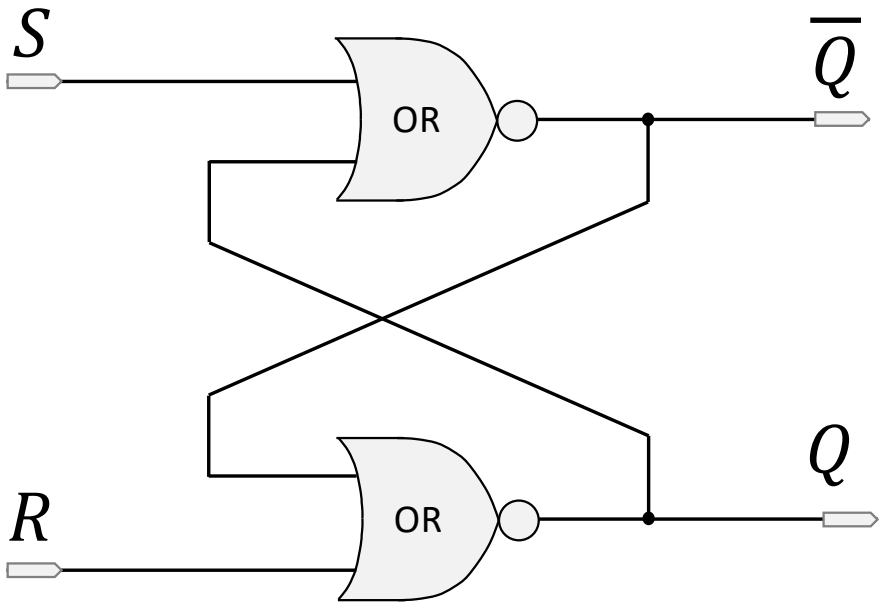


2) by using NAND logic gates:

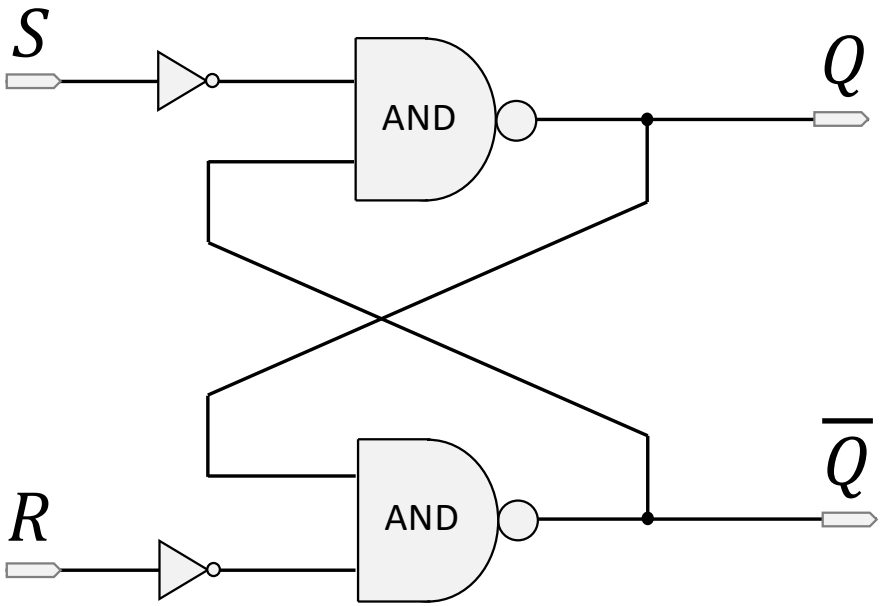


S/R Latch: Multiple Implementations Available

1) Implementation by using NOR logic gates:

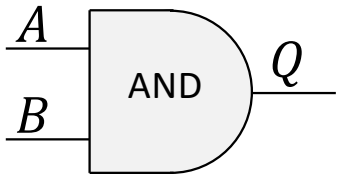
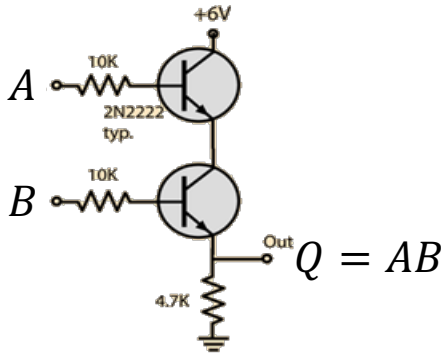
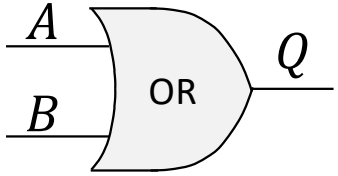
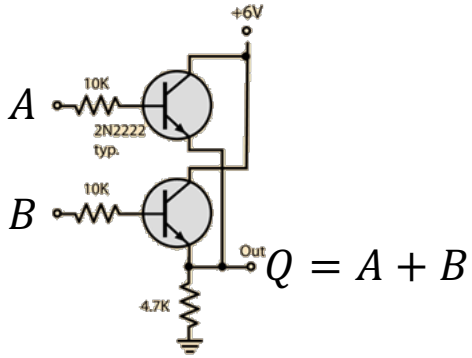
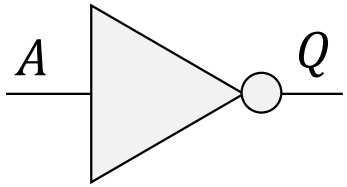
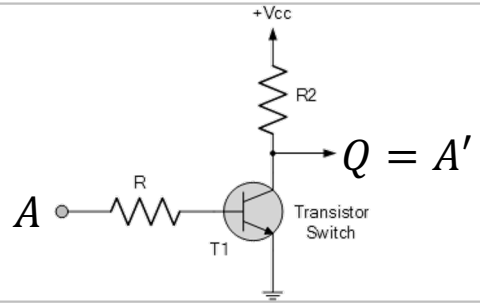


2) by using NAND logic gates:



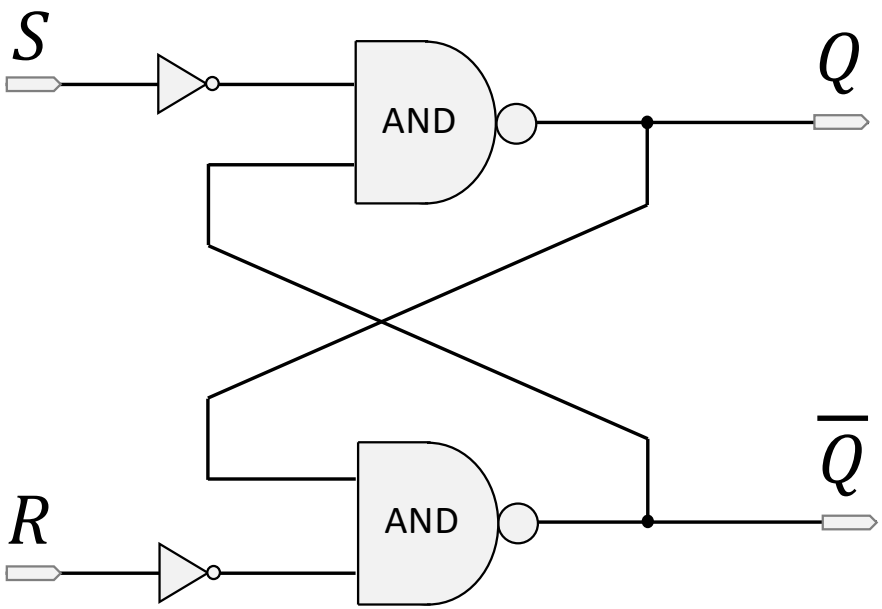
These representations are logical oversimplified representations, hiding many implementation details

Recap: Each logic gate is an electronic circuit, implemented by using transistors

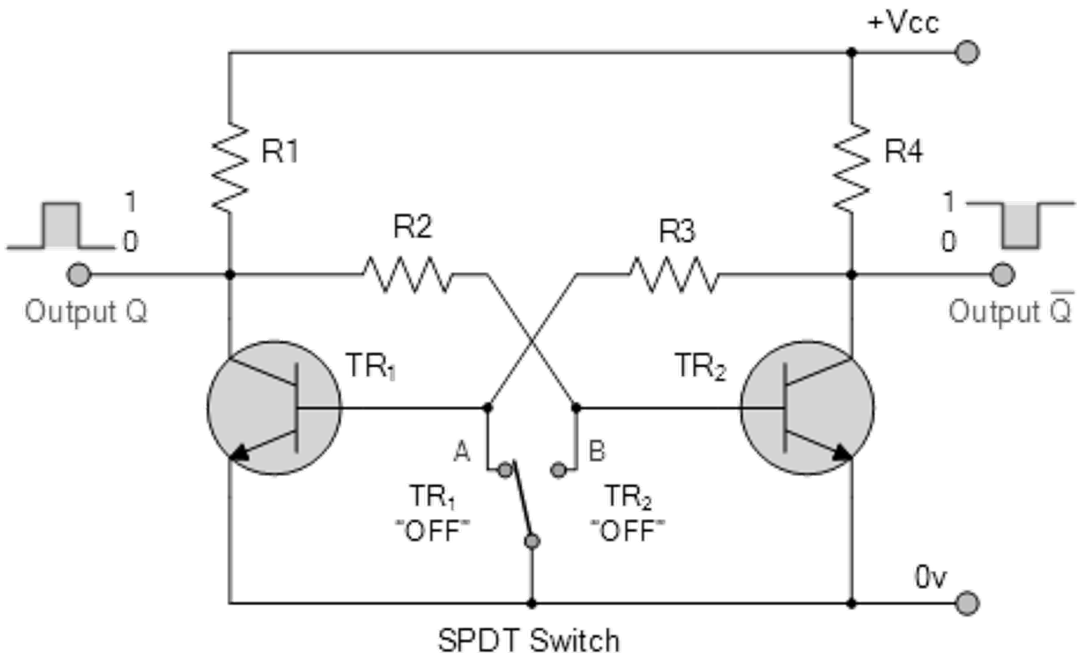
<table>
<tr><th>Logic Gate</th><th>Symbolic Representation</th><th>Truth Table</th><th>Implementation with Transistors</th></tr>
<tr>
<td>AND</td>
<td> 
</td>
<td>
<table>
<tr><th>A</th><th>B</th><th>Q=AB</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>
</td>
<td> 
</td>
</tr>
<tr>
<td>OR</td>
<td> 
</td>
<td>
<table>
<tr><th>A</th><th>B</th><th>Q=A+B</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>
</td>
<td> 
</td>
</tr>
<tr>
<td>NOT</td>
<td> 
</td>
<td>
<table>
<tr><th>A</th><th>Q=A'</th></tr>
<tr><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td></tr>
</table>
</td>
<td> 
</td>
</tr>
</table>

Transistor implementations are taken from http://hyperphysics.phy-astr.gsu.edu/hbase/Electronic/trangate.html#c1 and https://www.electronics-tutorials.ws/logic/logic_4.html

Latch Logical representation:



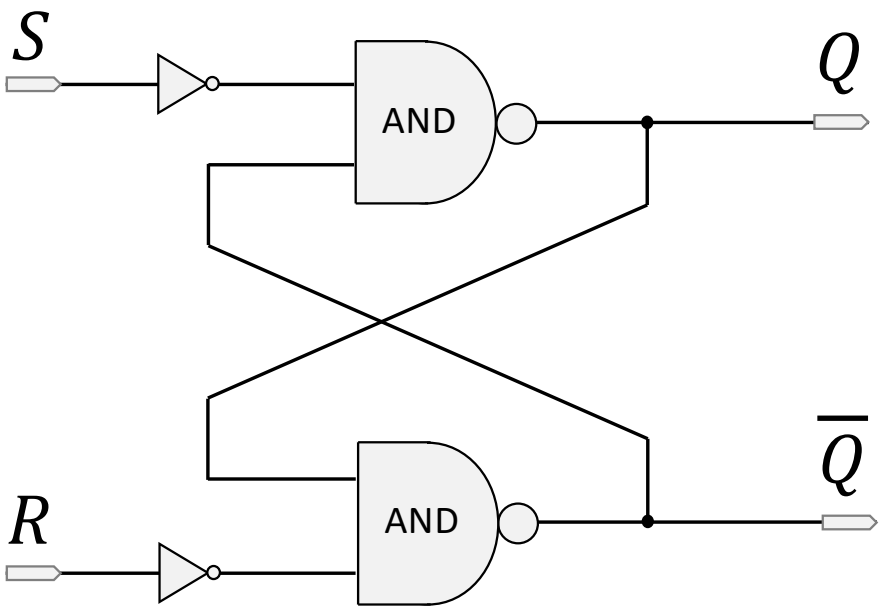
Electrical circuit implementation (one of many):



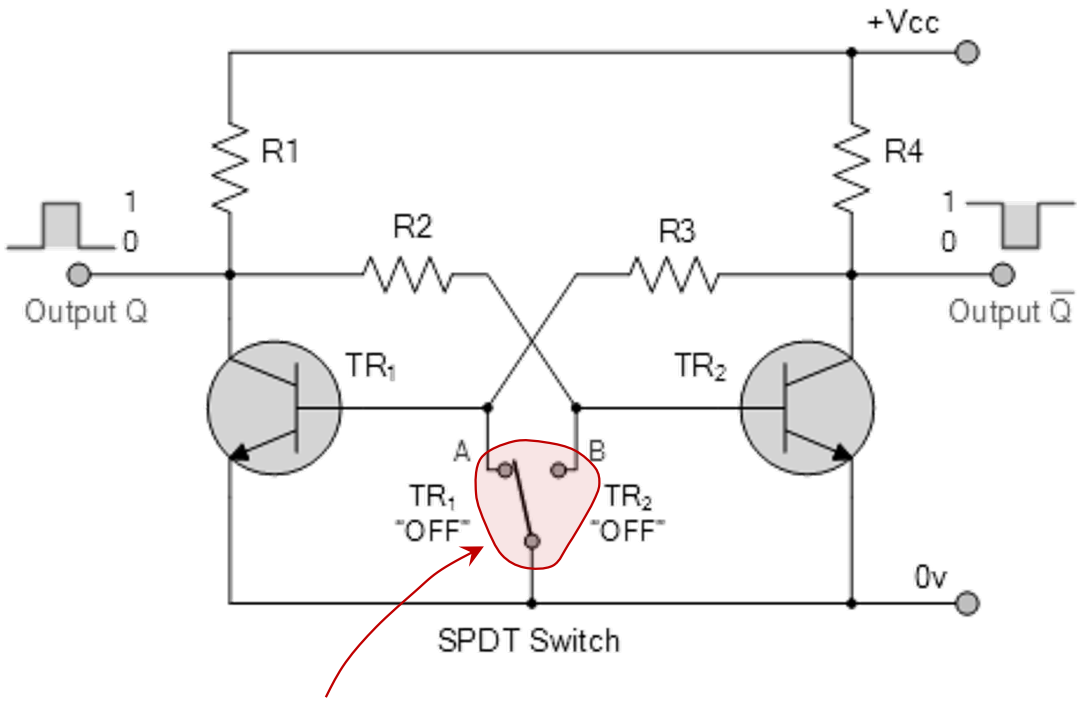
Before usage, latch is initialized by one of these input combinations:
either R = 1 & S = 0, or S = 1 & R = 0;

Image is taken from https://www.electronics-tutorials.ws/waveforms/bistable.html

Latch Logical representation:

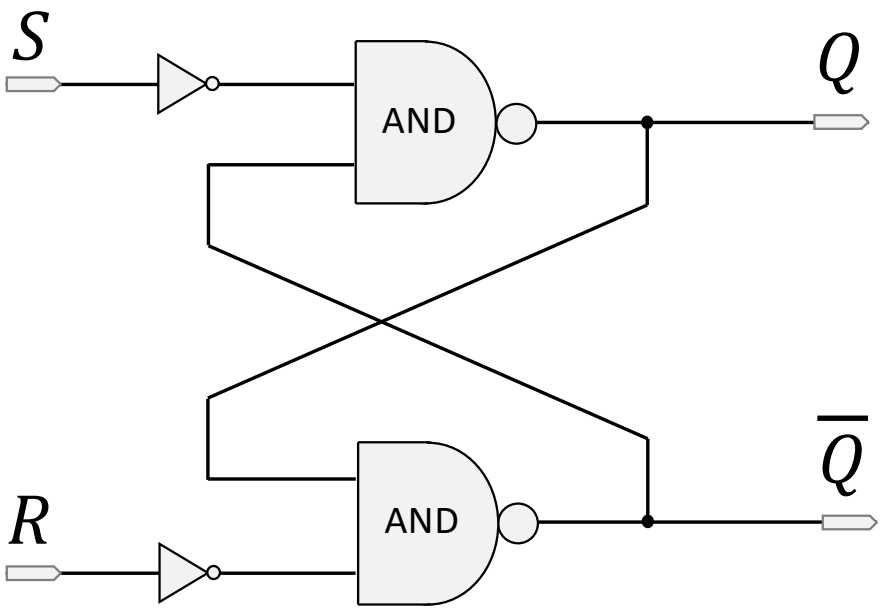


Electrical circuit implementation (one of many):



Switch to control
the input values of *S* and *R*
(assumed to have 3 positions)

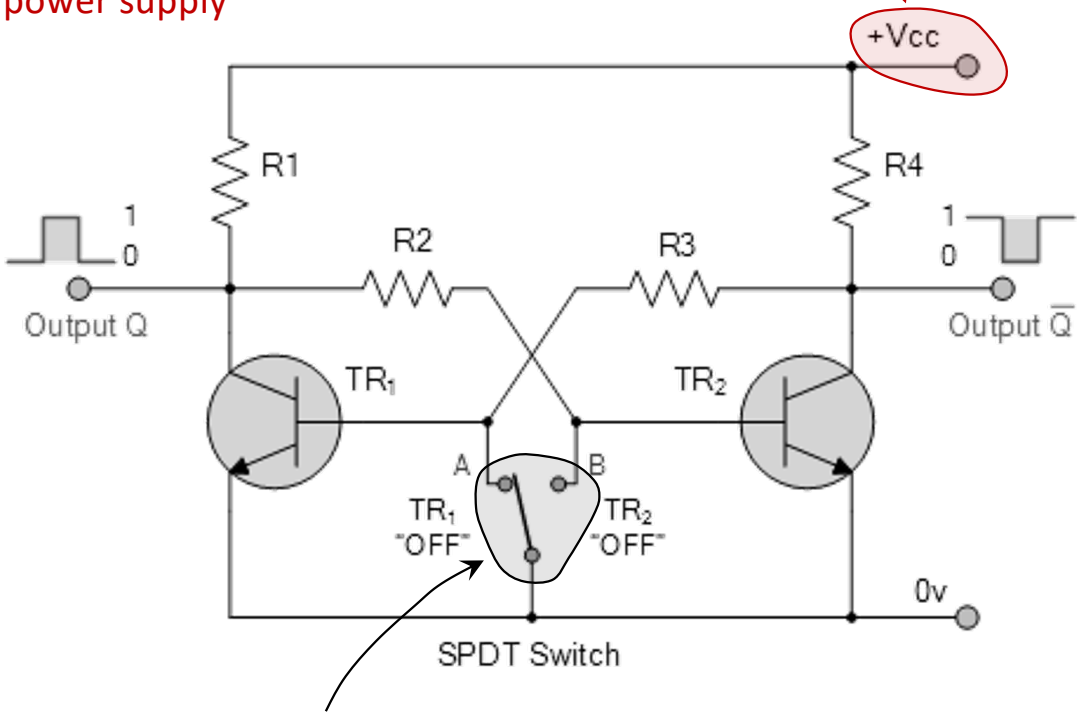
Latch Logical representation:



Electrical circuit implementation (one of many):

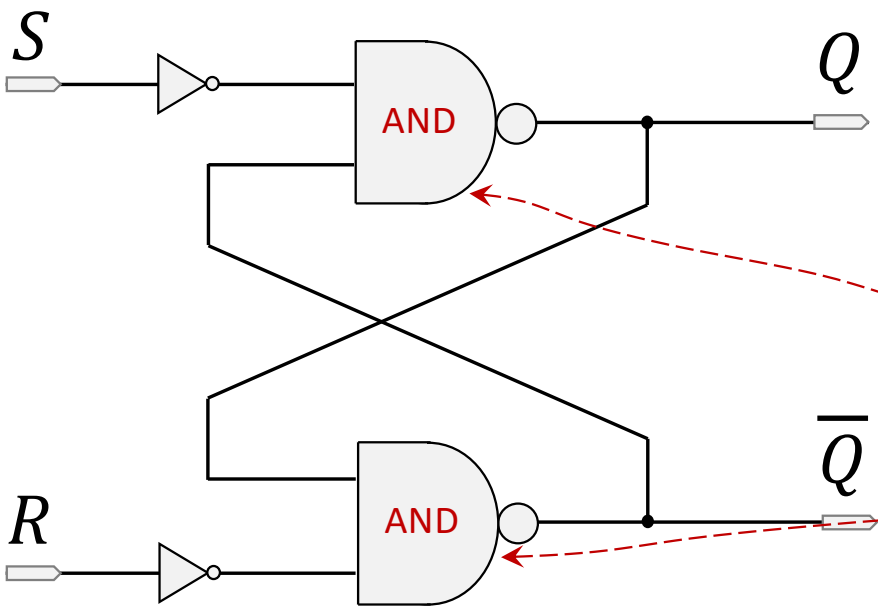
If S and R are “0”s, Q might remain “1”;
Latch is not an “endless engine”, it has a power supply

There is an external power supply, to support latch functioning

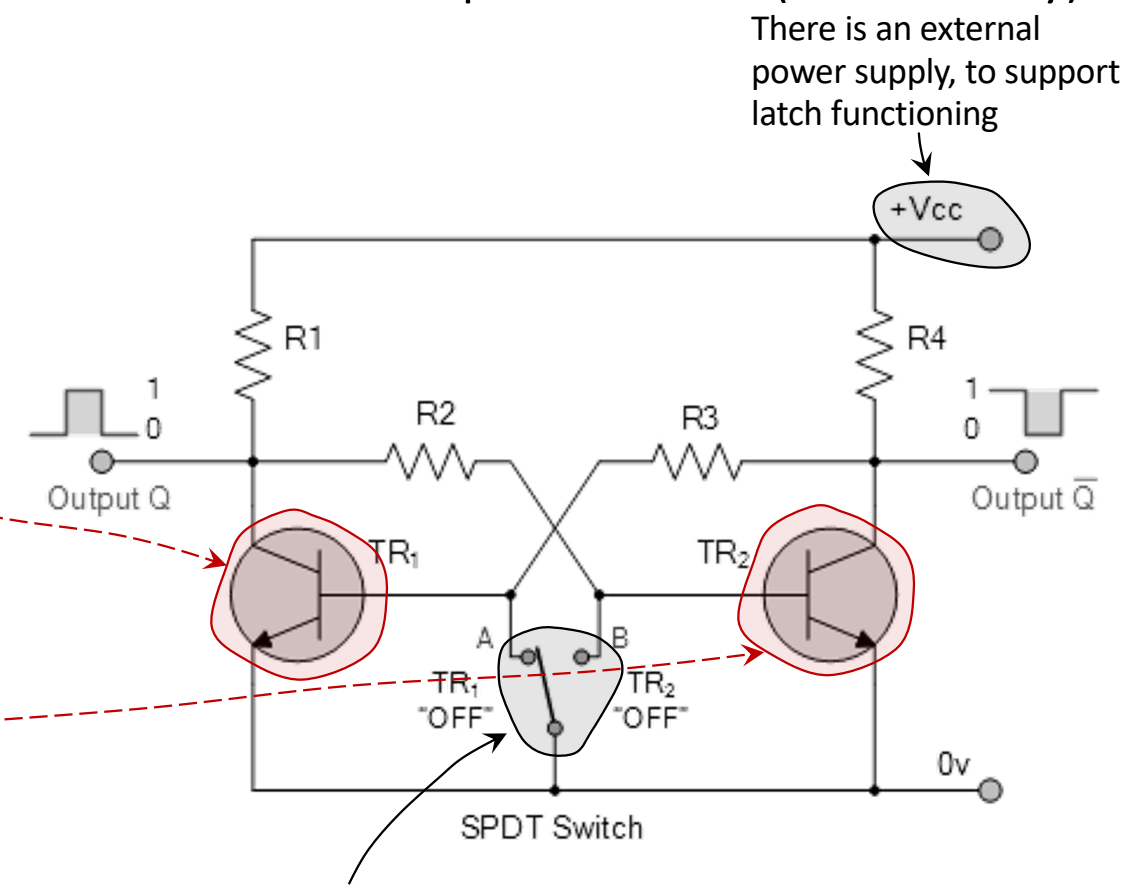


Switch to control
the input values of *S* and *R*
(assumed to have 3 positions)

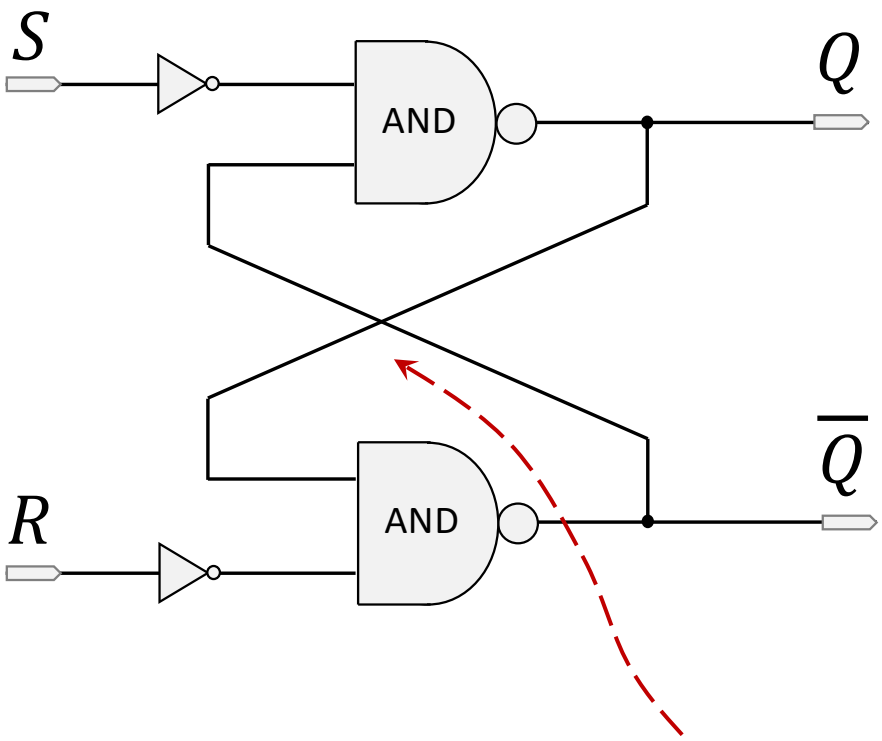
Latch Logical representation:



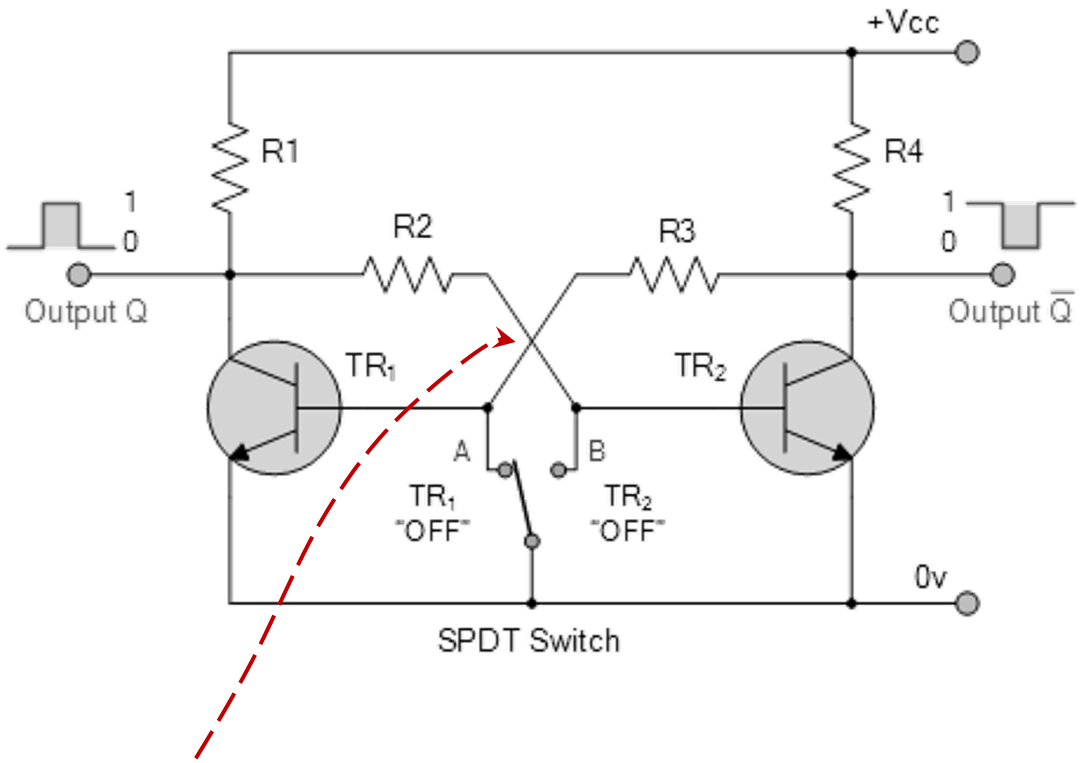
Electrical circuit implementation (one of many):



Latch Logical representation:



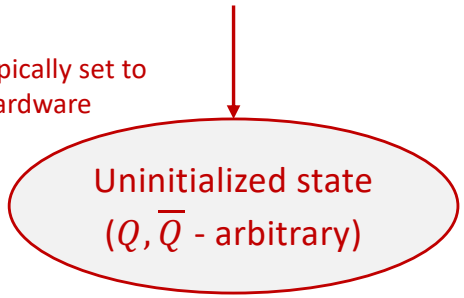
Electrical circuit implementation (one of many):

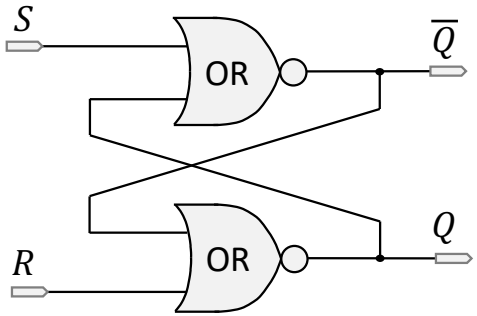


Cross-coupled connections between gates and corresponding transistors

State Transition Diagram for a Latch

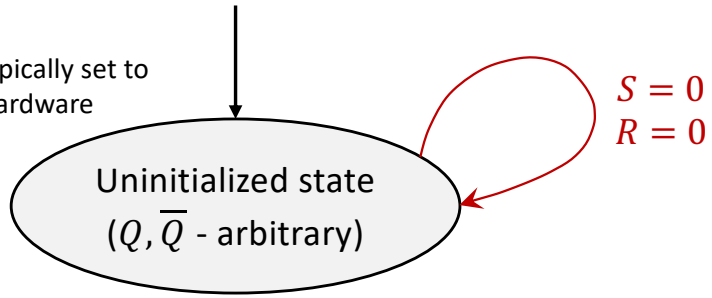
At this state, Q and $\overline{Q}$ are typically set to 0, for most of the availble hardware implementations

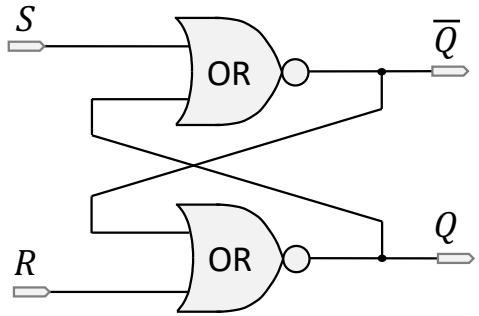




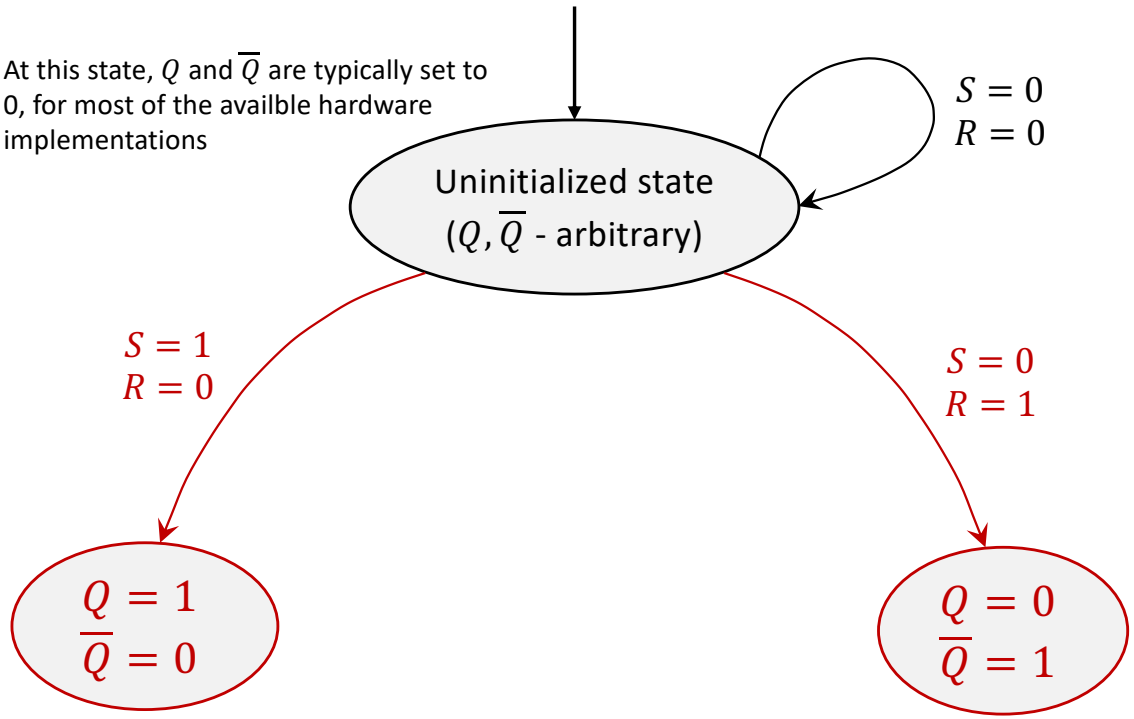
State Transition Diagram for a Latch

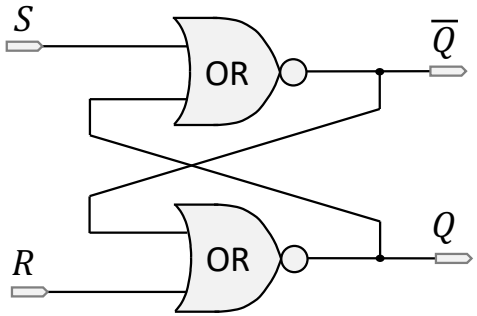
At this state, Q and $\overline{Q}$ are typically set to 0, for most of the availble hardware implementations



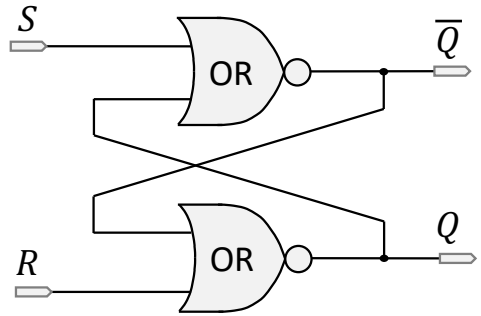


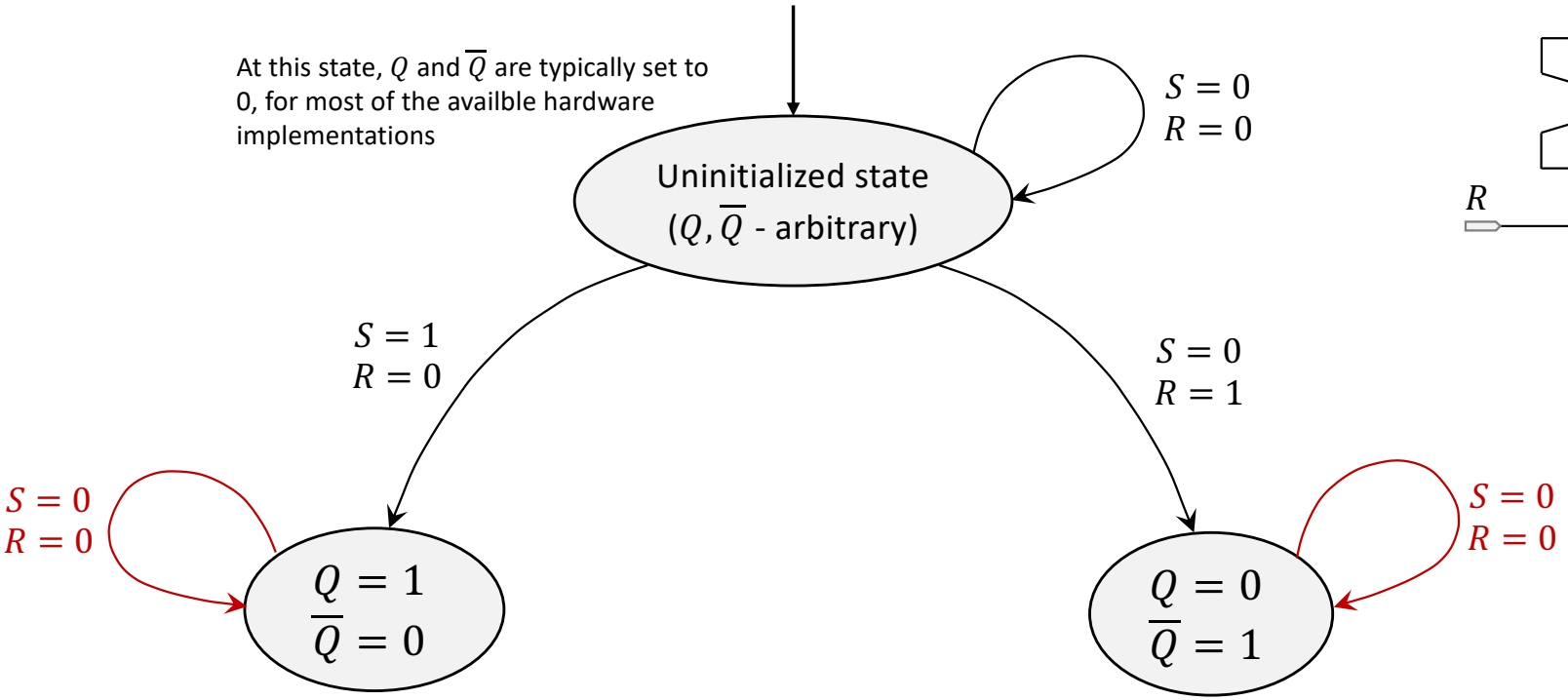
State Transition Diagram for a Latch



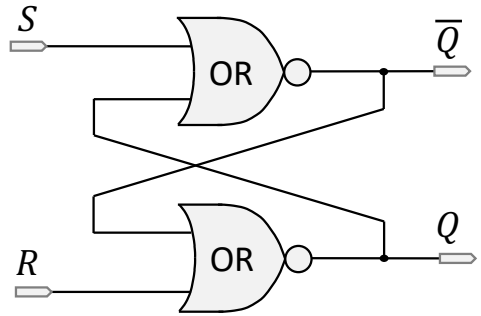


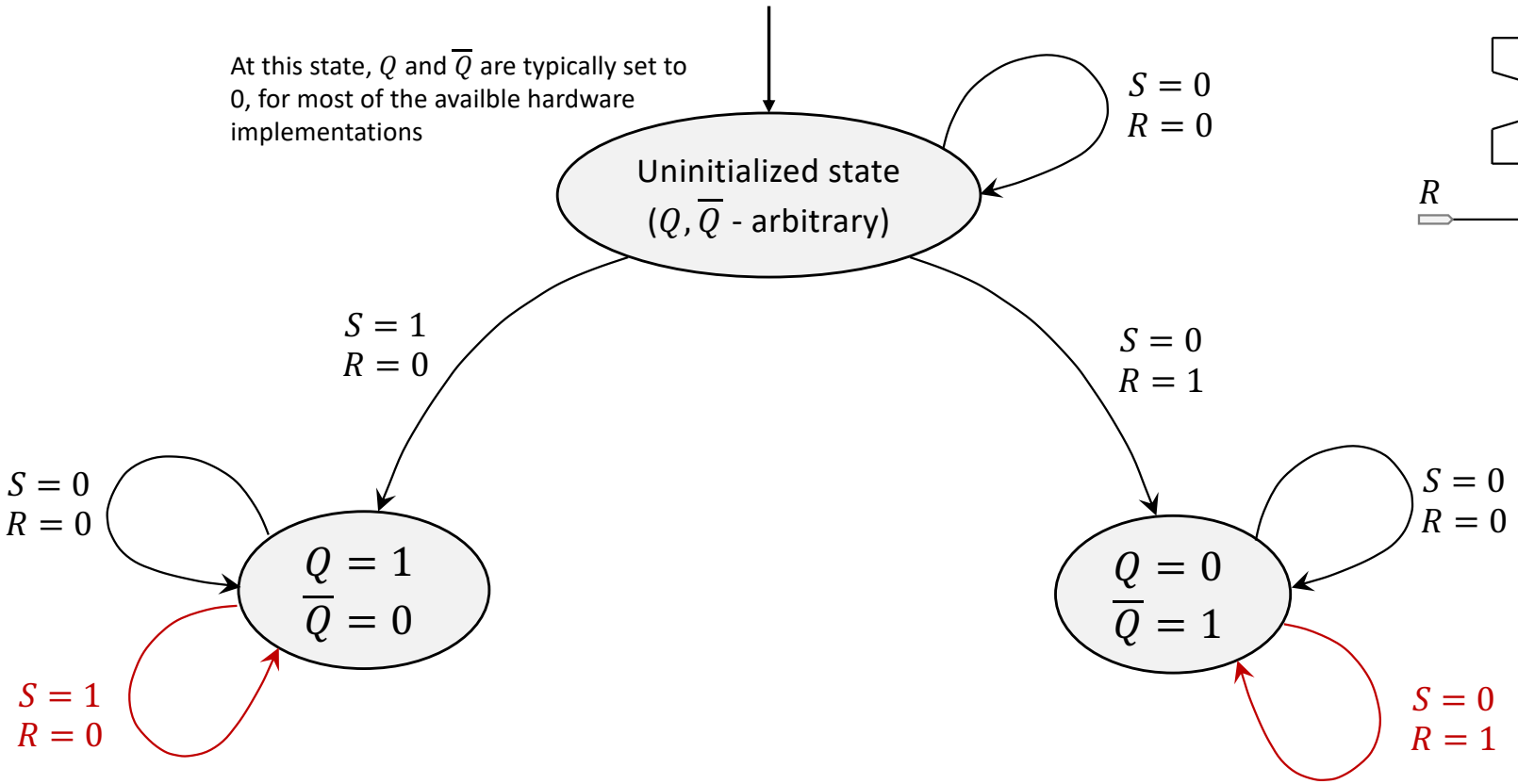
State Transition Diagram for a Latch



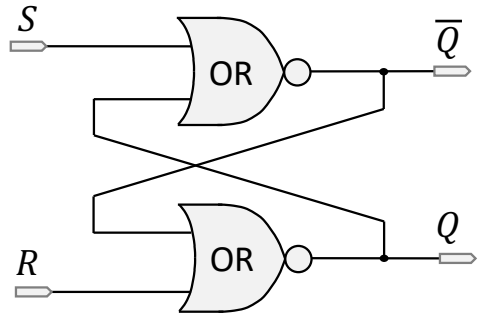


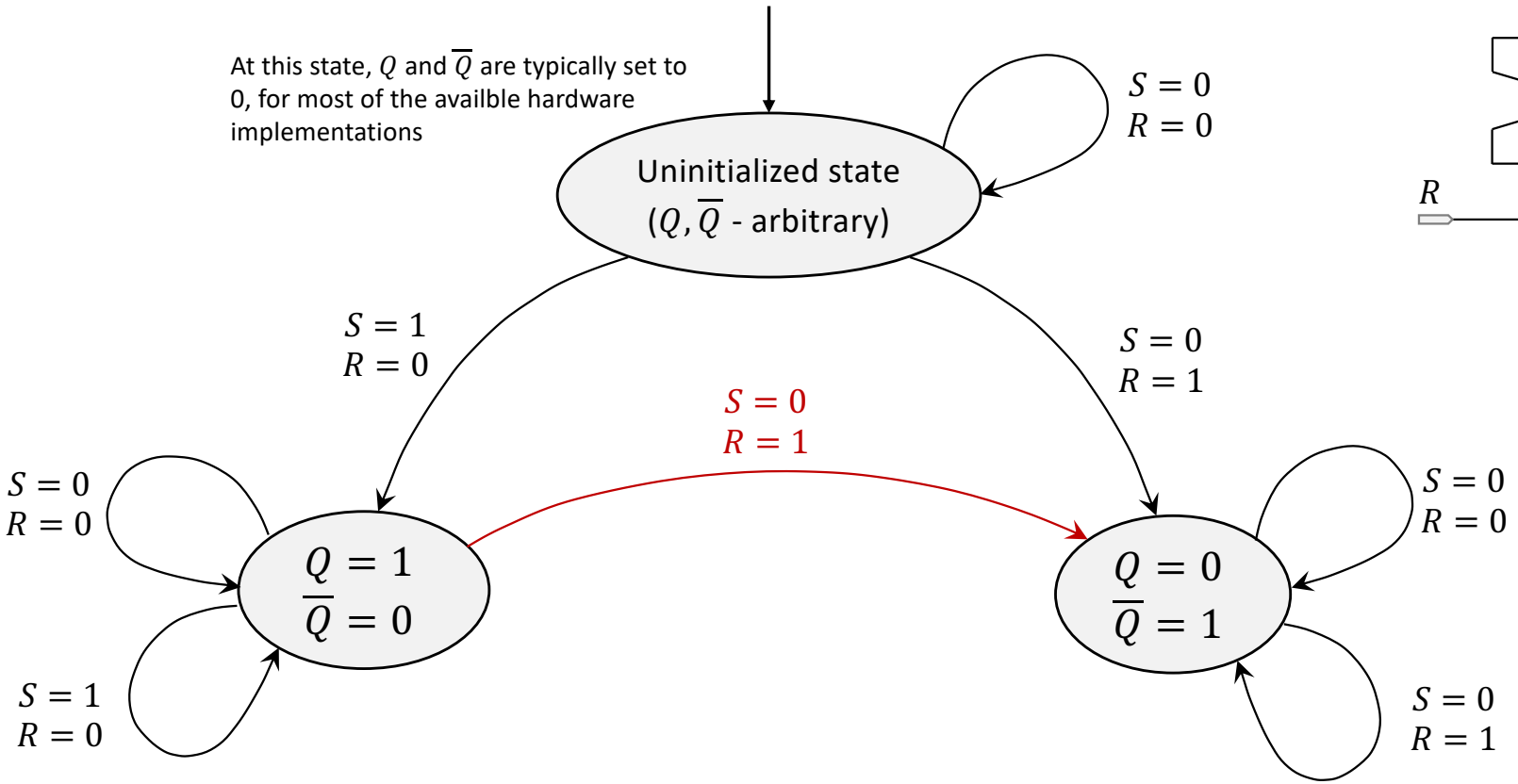
State Transition Diagram for a Latch



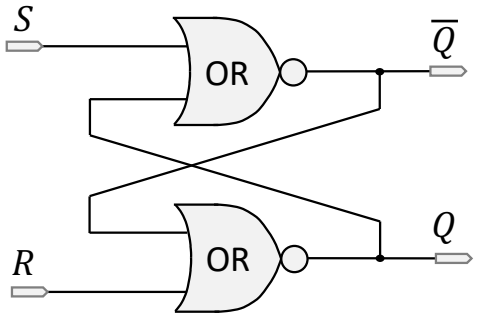


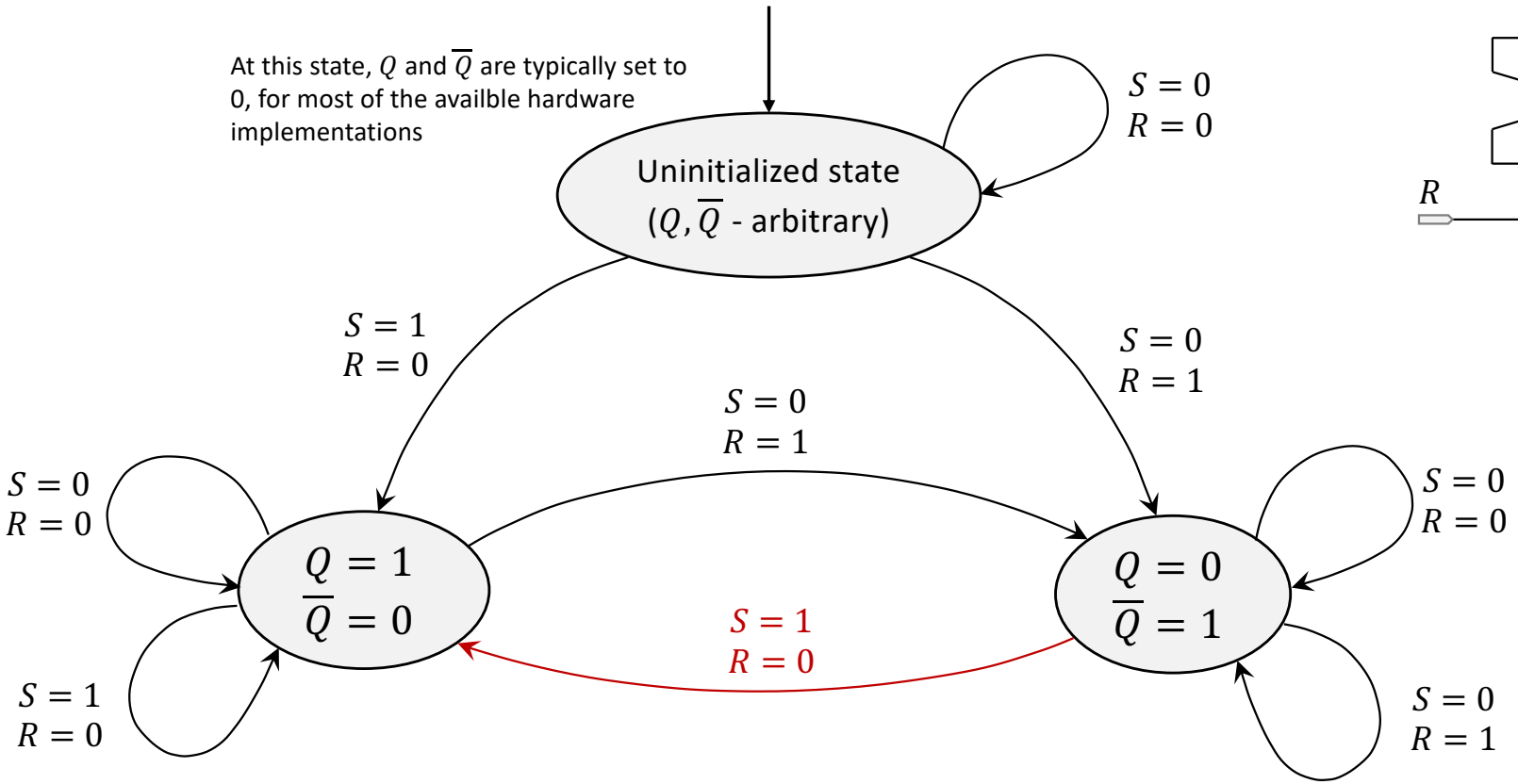
State Transition Diagram for a Latch



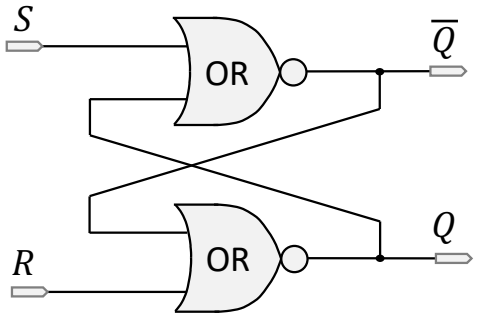


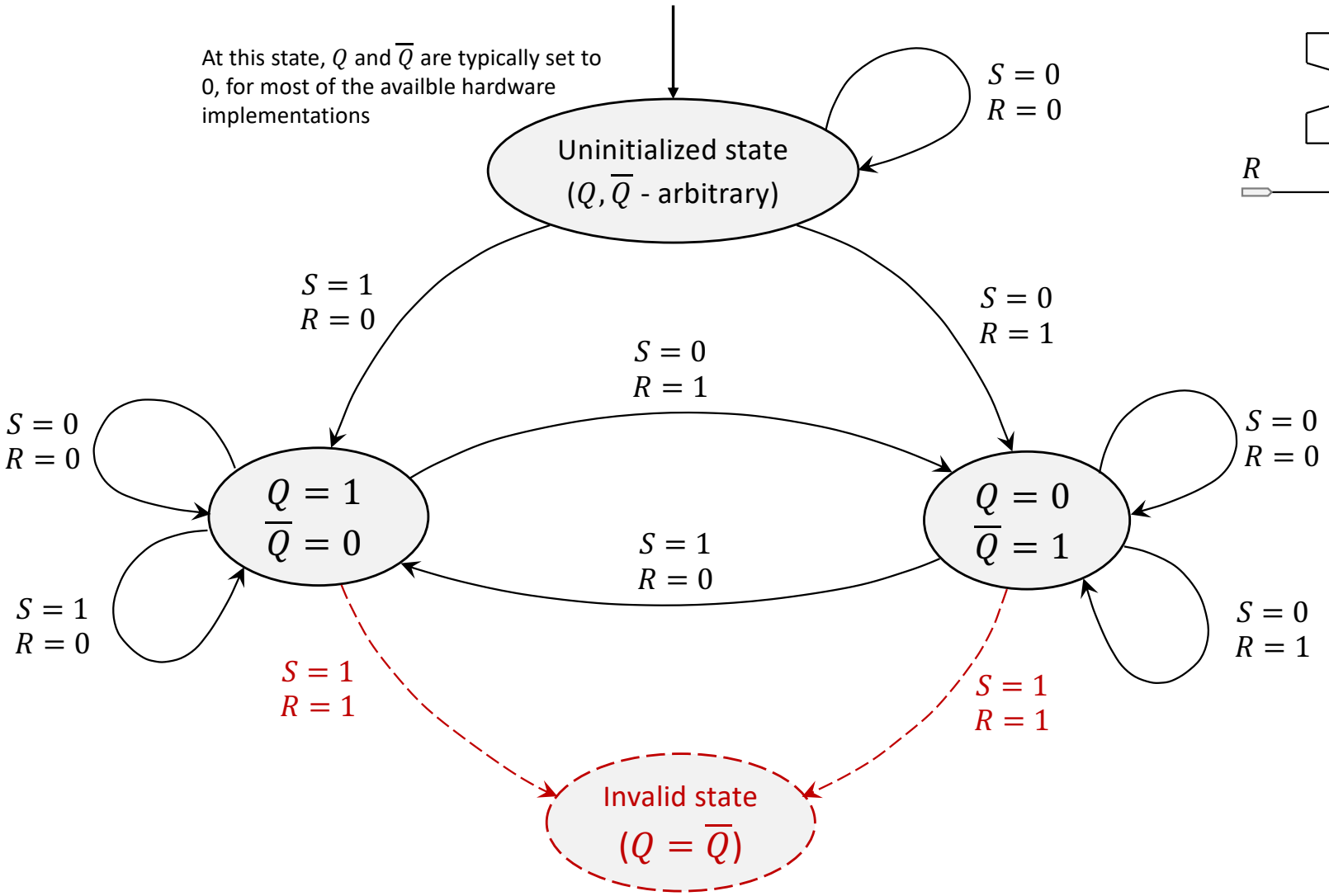
State Transition Diagram for a Latch



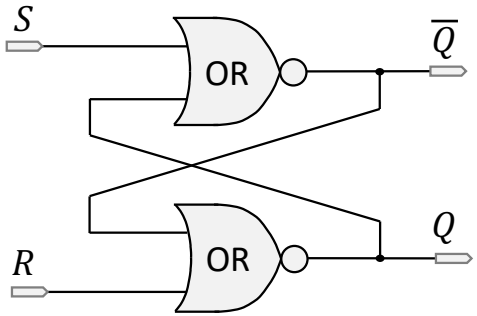


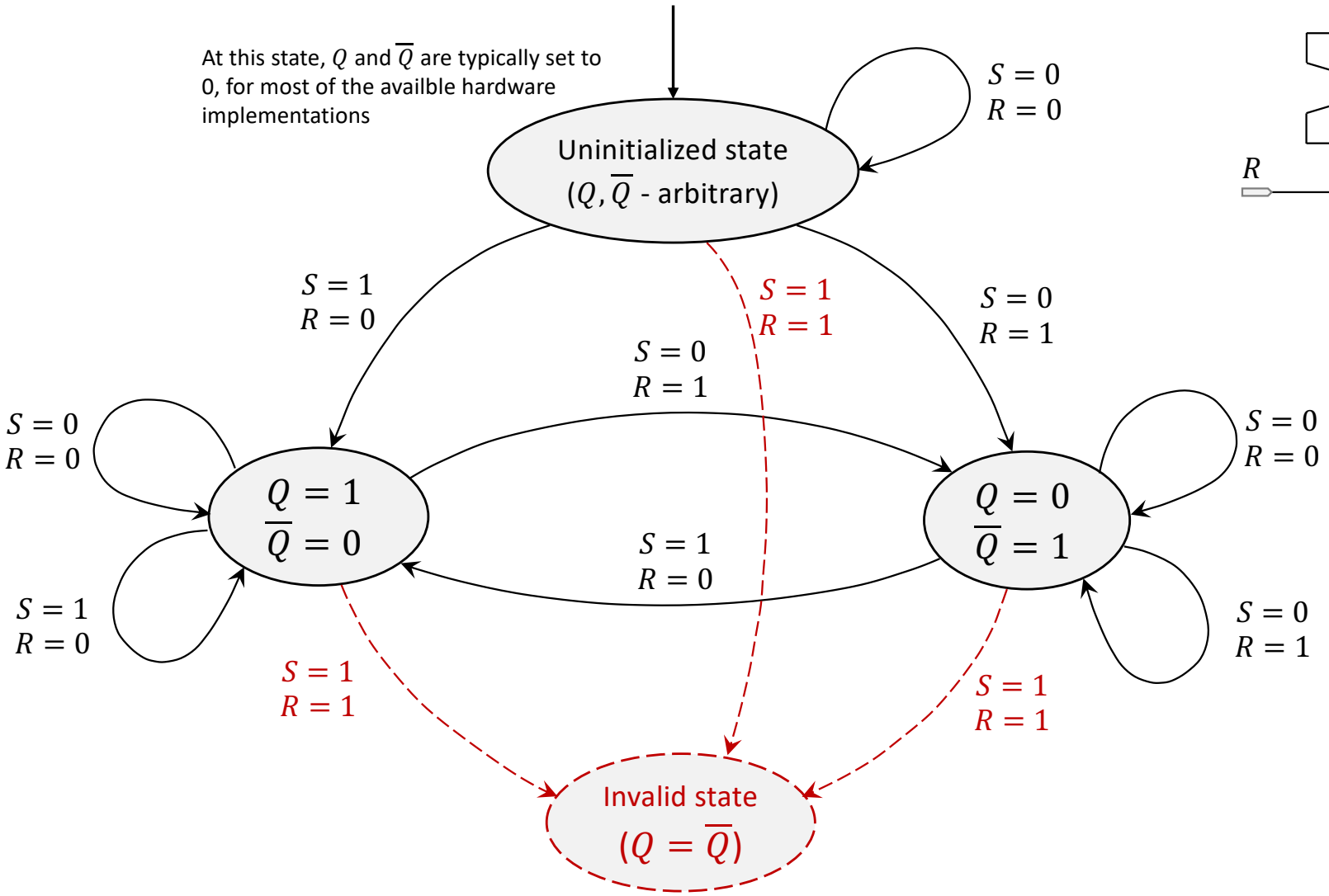
State Transition Diagram for a Latch



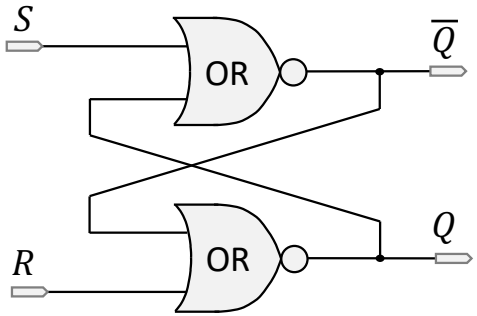


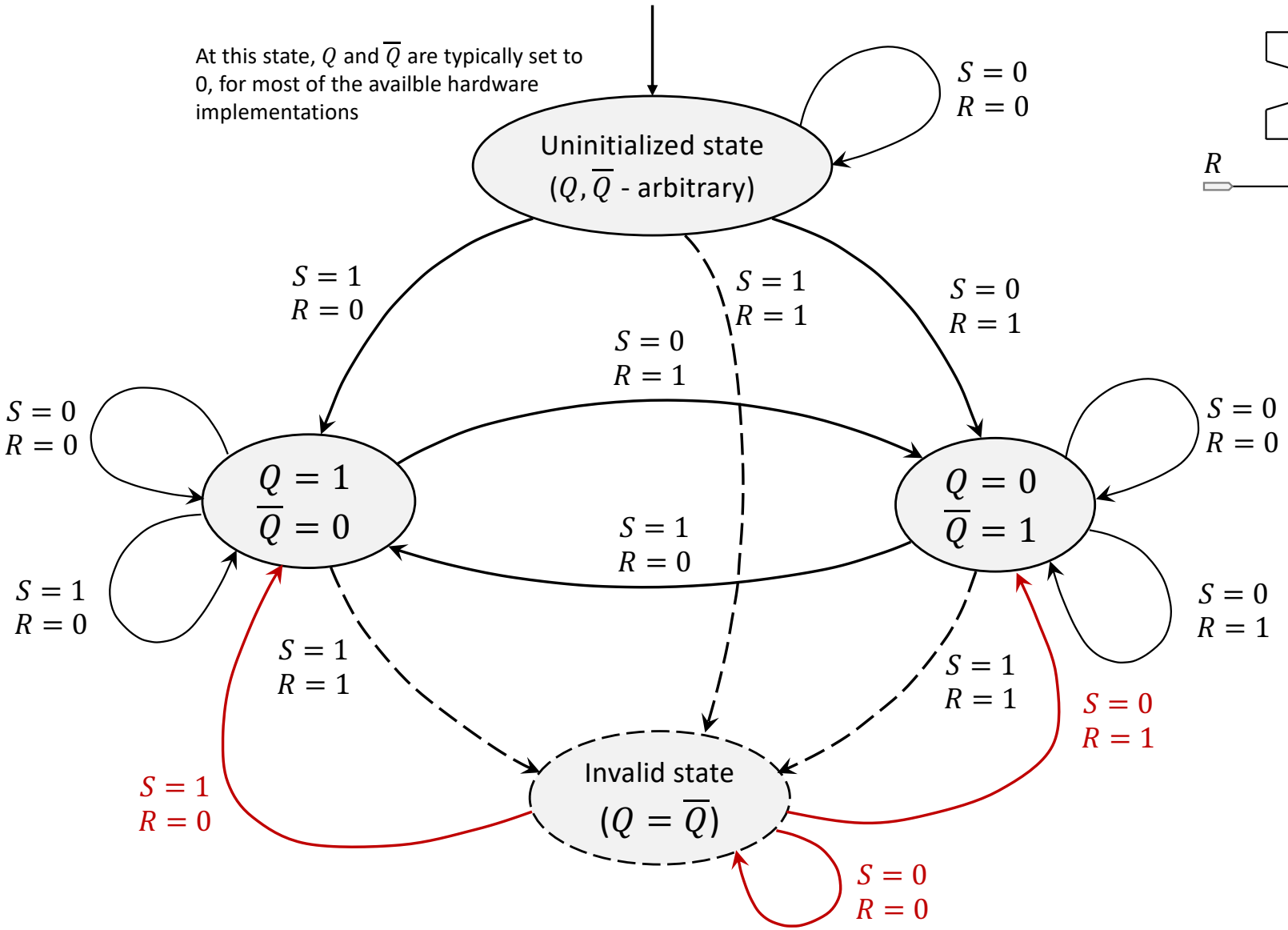
State Transition Diagram for a Latch



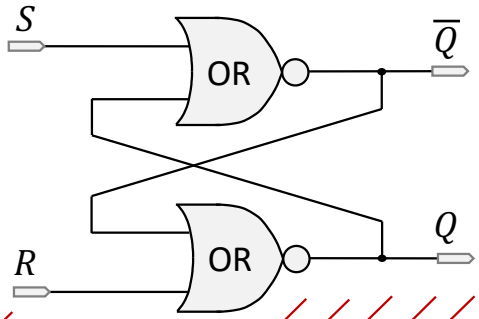


State Transition Diagram for a Latch





State-Transition Diagram for a Latch



At this state, Q and $\overline{Q}$ are typically set to 0, for most of the availble hardware implementations

Uninitialized state
($Q, \overline{Q}$ - arbitrary)

$S = 0$
$R = 0$

$Q, \overline{Q}$ get its meaning only after this point

$S = 1$
$R = 0$

$S = 1$
$R = 1$

$S = 0$
$R = 1$

$S = 0$
$R = 1$

$S = 0$
$R = 0$

$Q = 1$
$\overline{Q} = 0$

$Q = 0$
$\overline{Q} = 1$

$S = 0$
$R = 0$

$S = 1$
$R = 0$

$S = 1$
$R = 0$

$S = 0$
$R = 1$

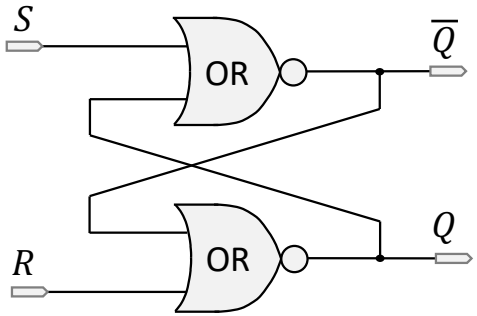
$S = 1$
$R = 1$

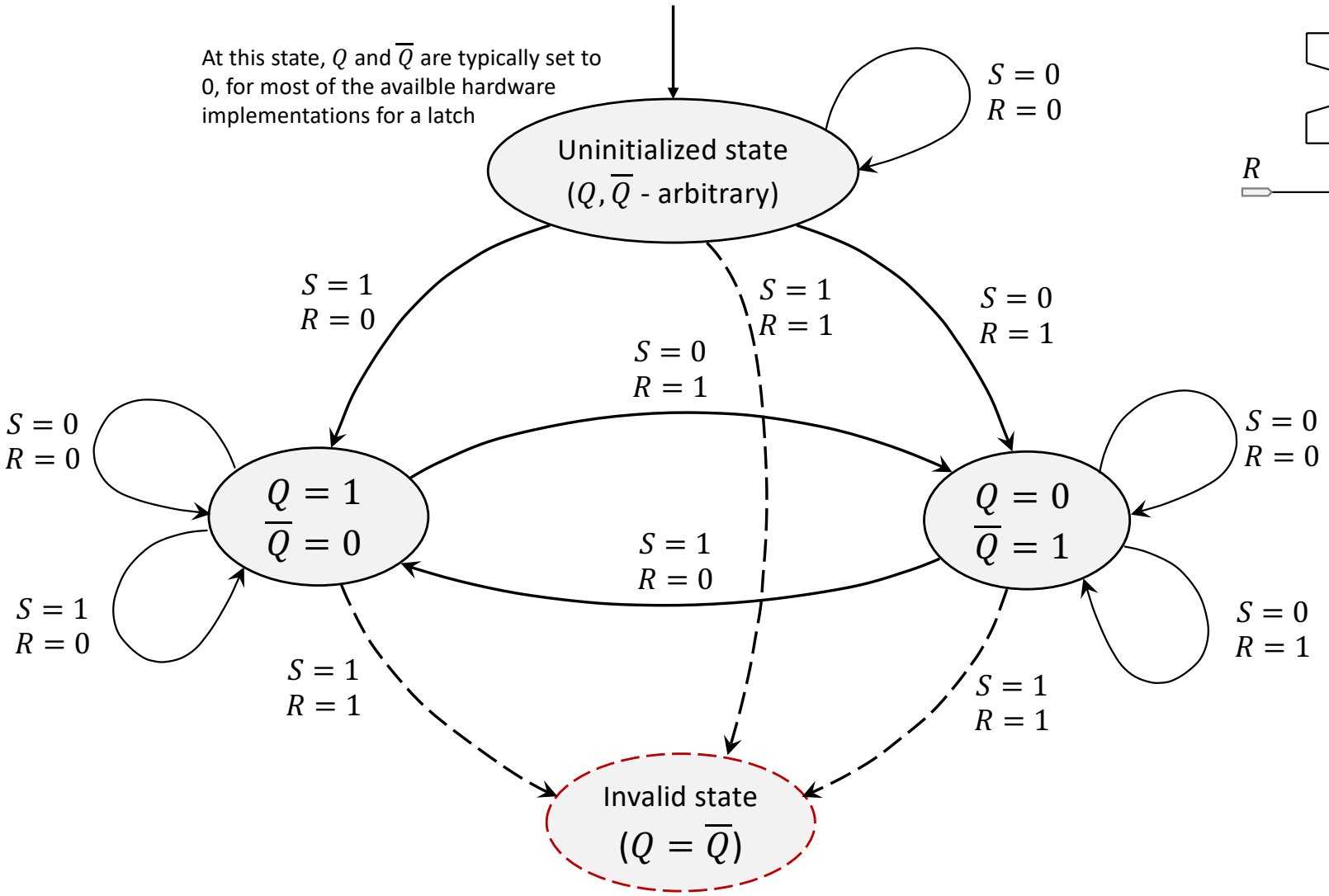
$S = 1$
$R = 1$

... until this point

Invalid state
($Q = \overline{Q}$)

State-Transition Diagram for a Latch





A reversible state?

State-Transition Diagram for a Latch