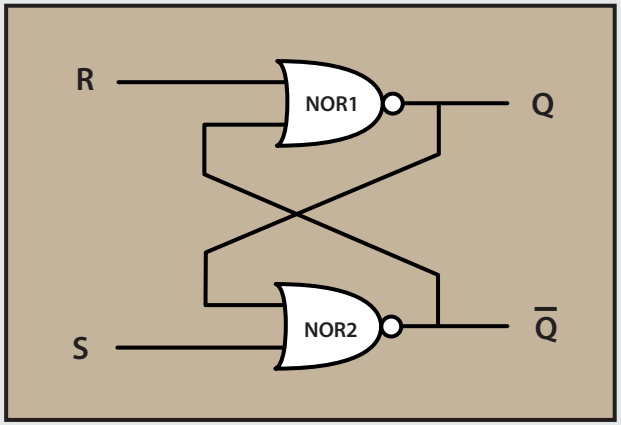
Computer Architecture

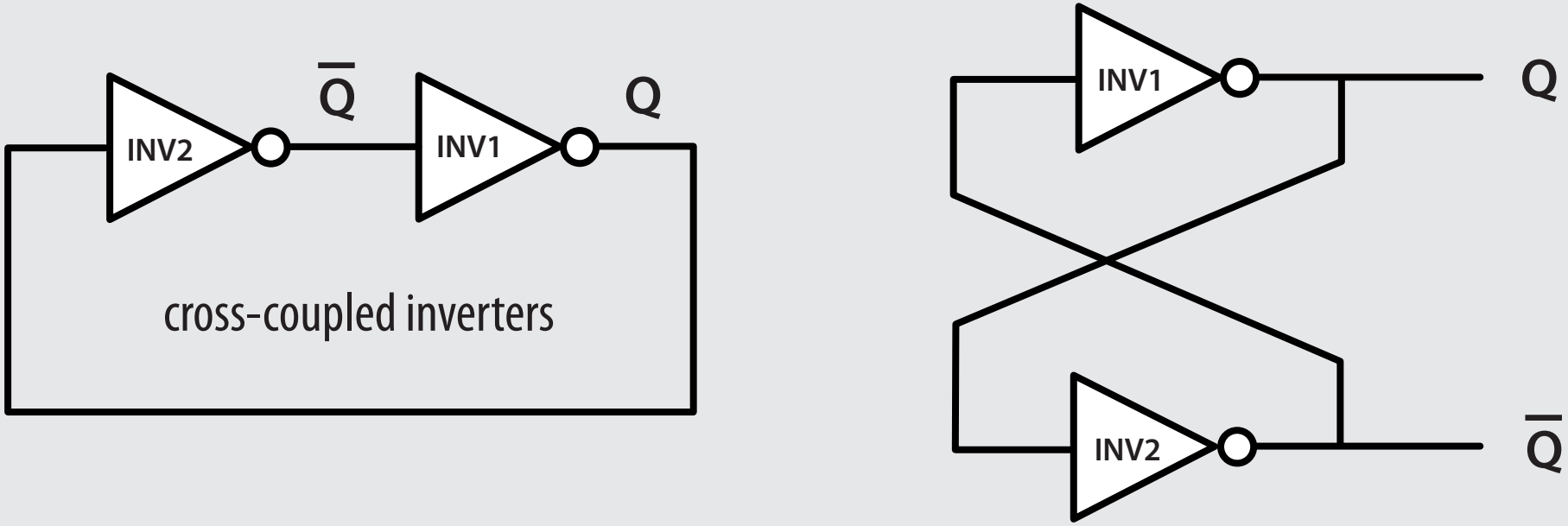
Paul Mellies

Lecture 10 : Latches & Flip-Flops



Latches and flip-flops

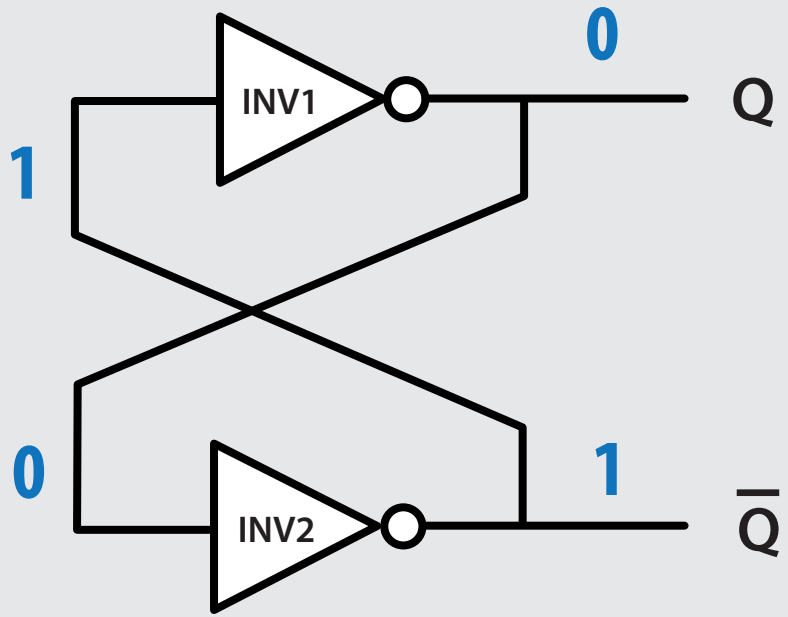
The fundamental building block of memory is a ***bistable*** element, that is : an element with two stable states.



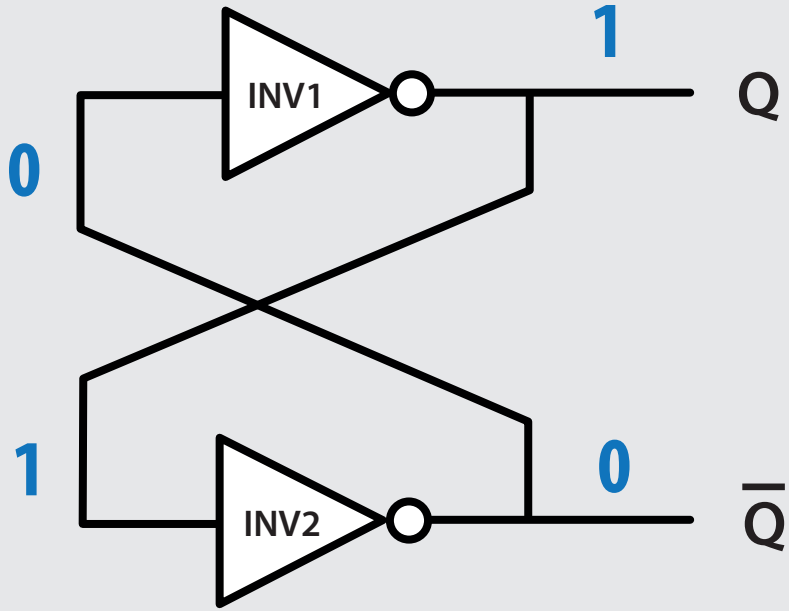
First stable state : $Q = 0$ and $\overline{Q} = 1$

Second stable state : $Q = 1$ and $\overline{Q} = 0$

The two stable states



First stable state

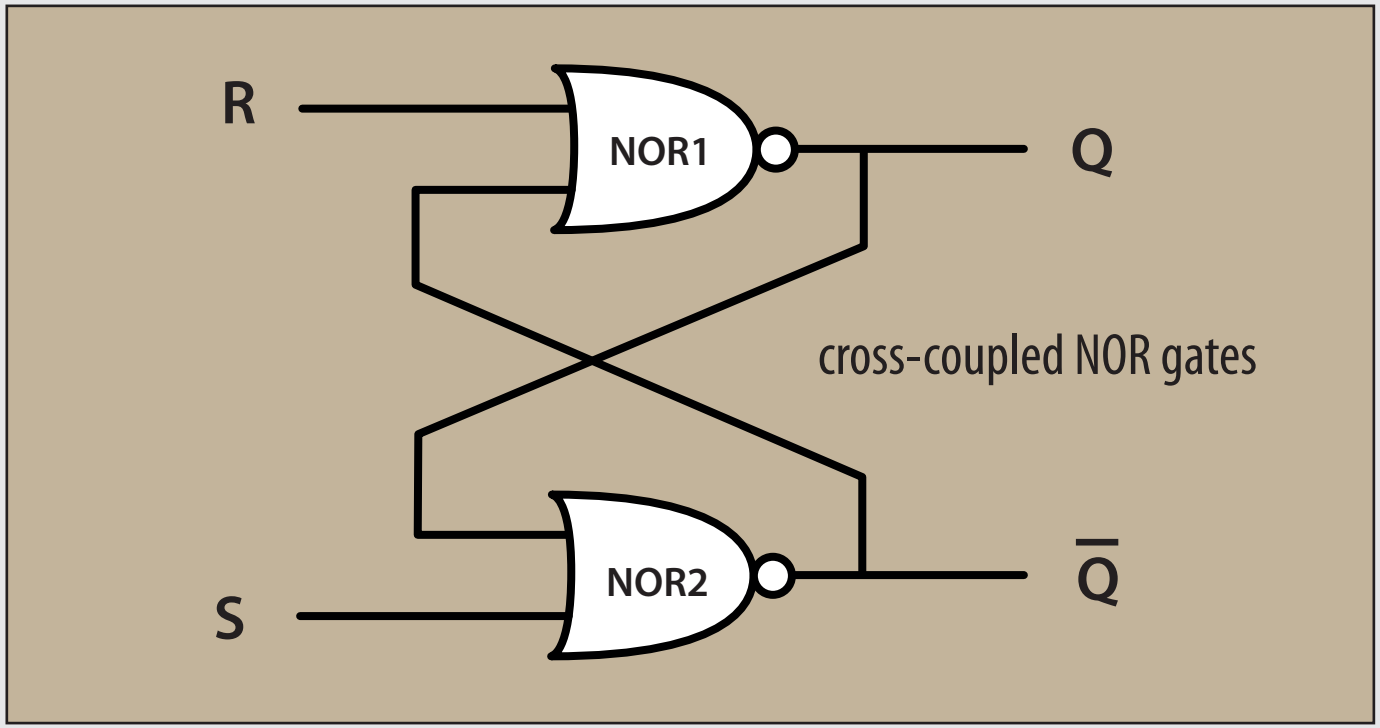


Second stable state

SR latches

SR-latches

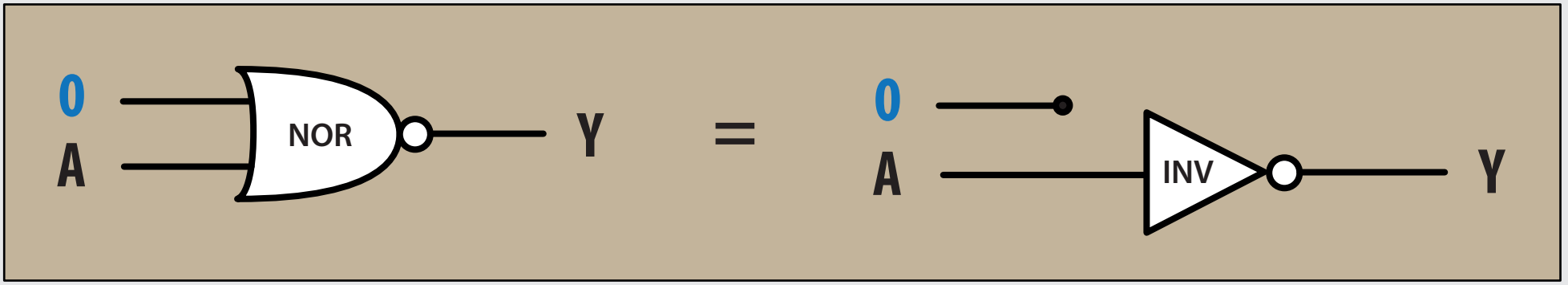
The SR-latch is one of the simplest sequential circuits consisting of two cross-coupled NOR gates :

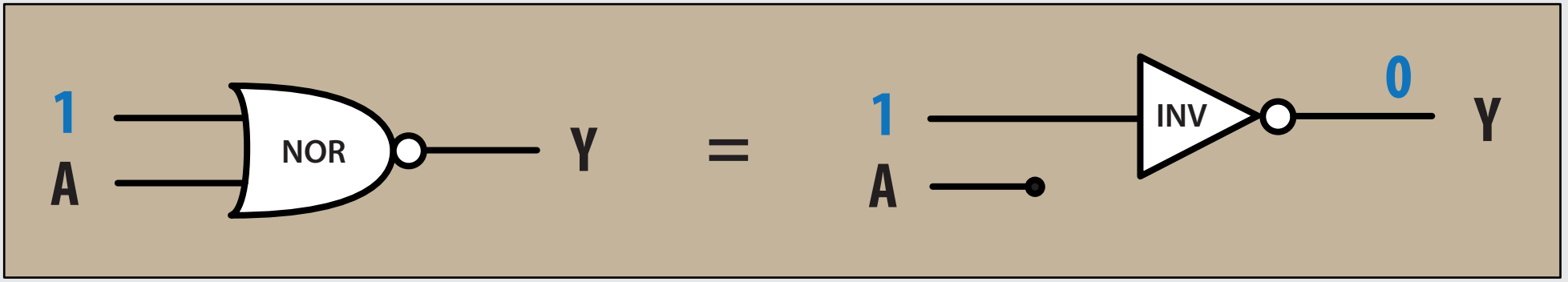


The SR-latch is similar to the cross-coupled inverters, except that its internal state can be controlled by the S and R inputs :

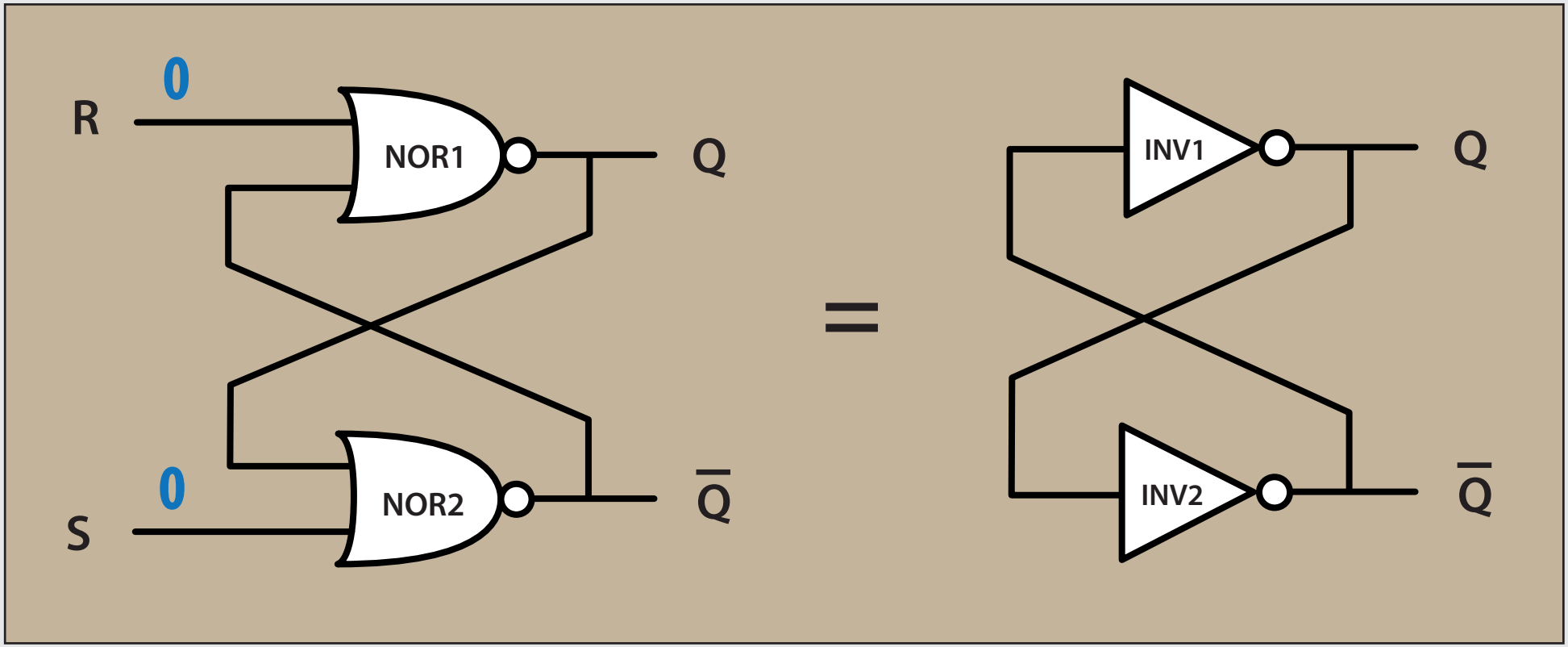
- S stands for ***set***
- R stands for ***reset***

Two useful equalities to remember



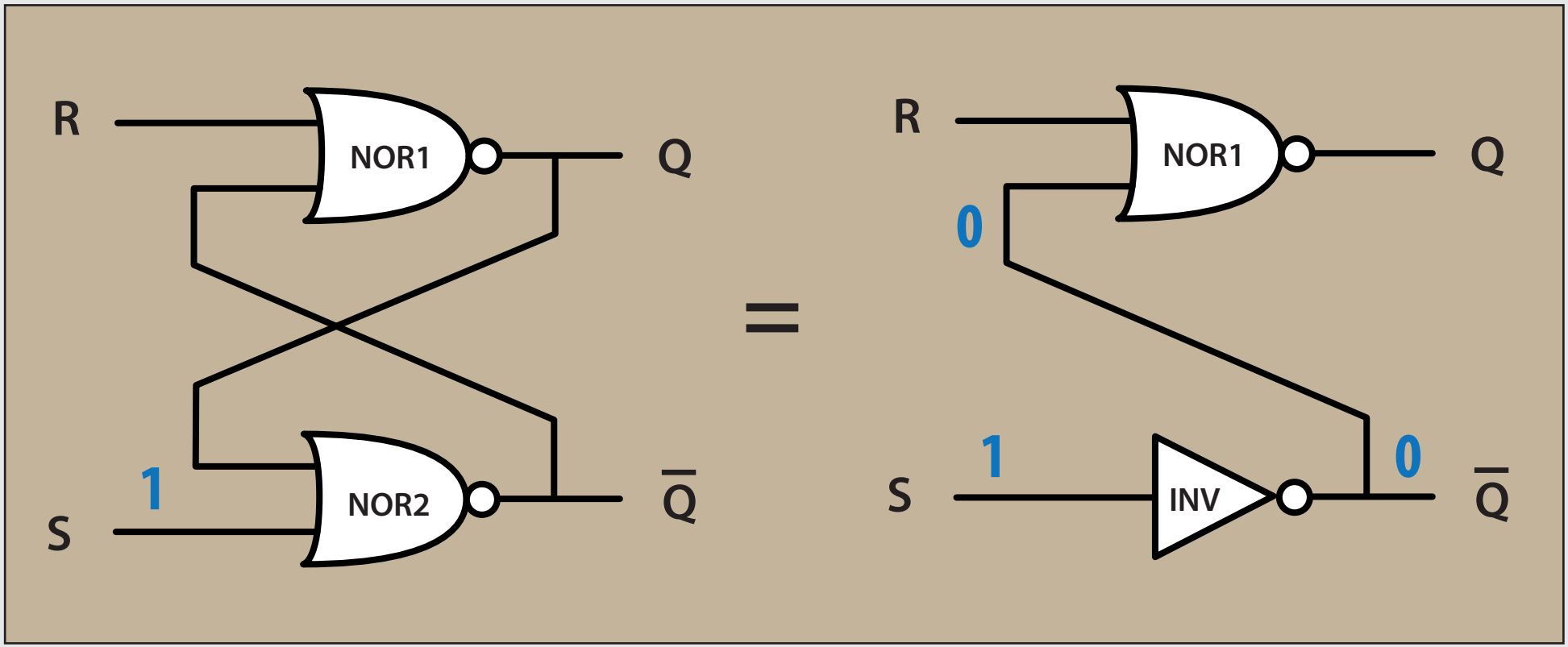


Application to the SR-latch
Set = 0 & Reset = 0

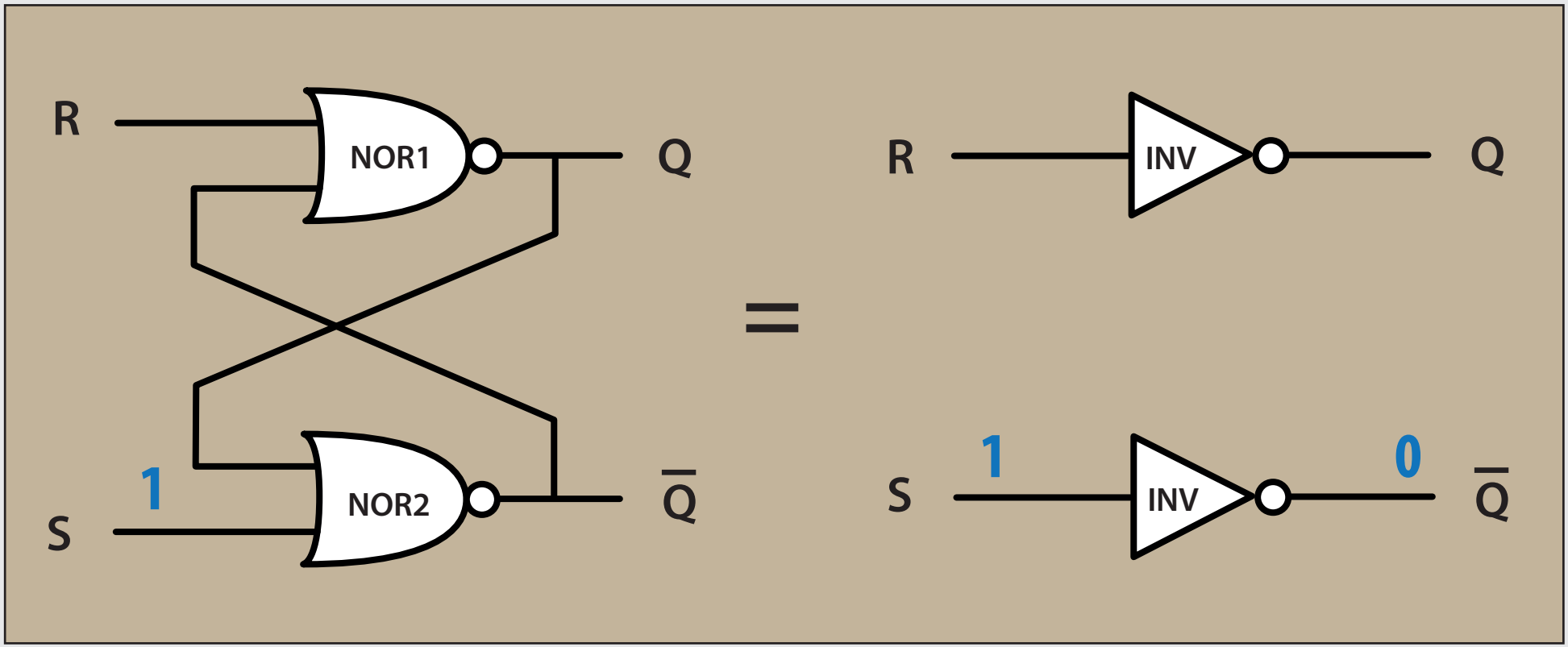


The SR-latch behaves as a pair of cross-coupled inverters.

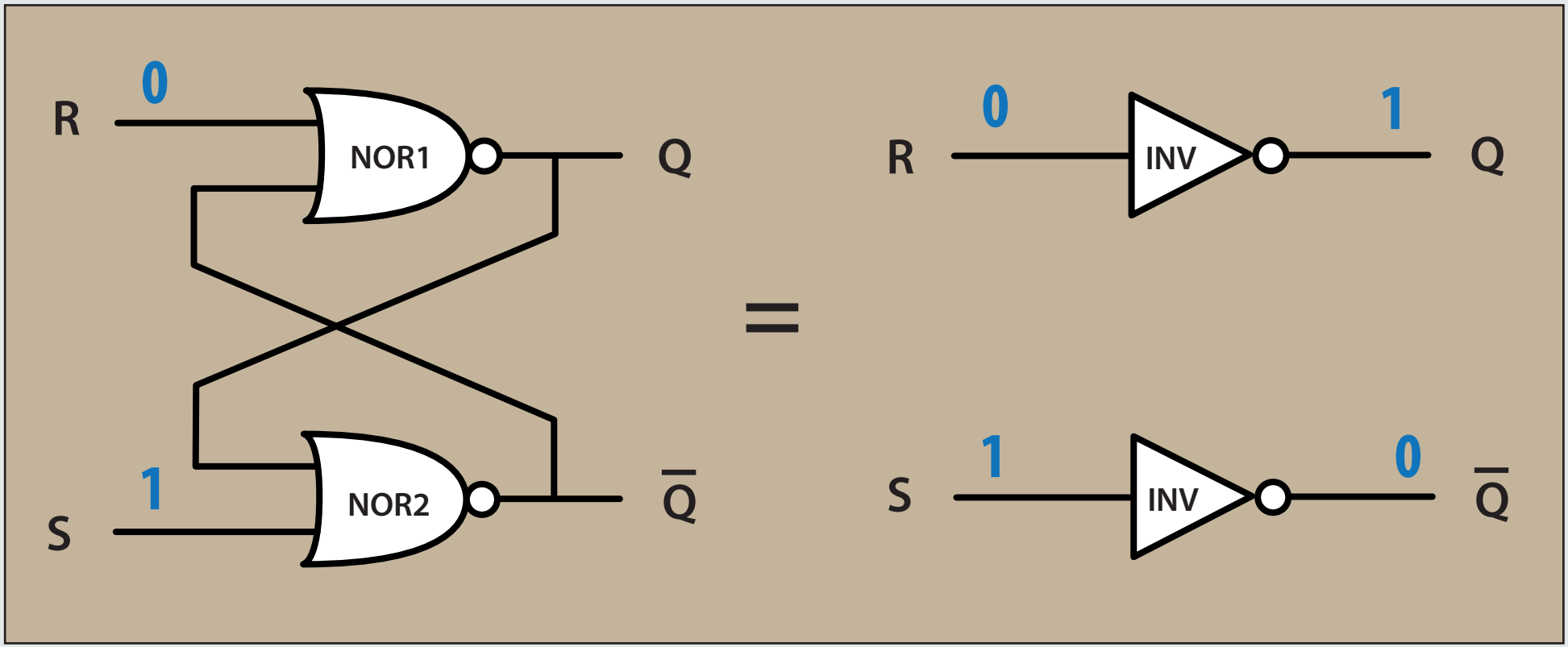
Application to the SR-latch
Set = 1



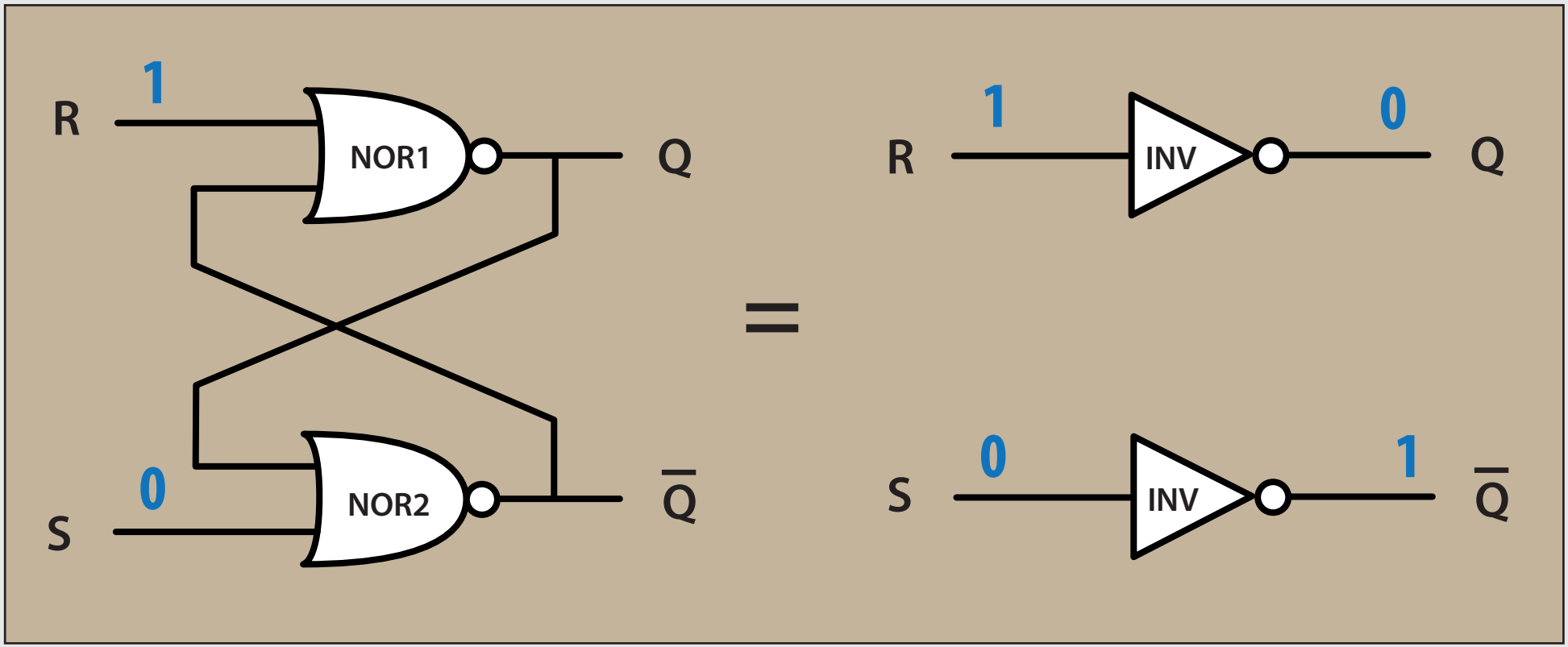
Application to the SR-latch
Set = 1



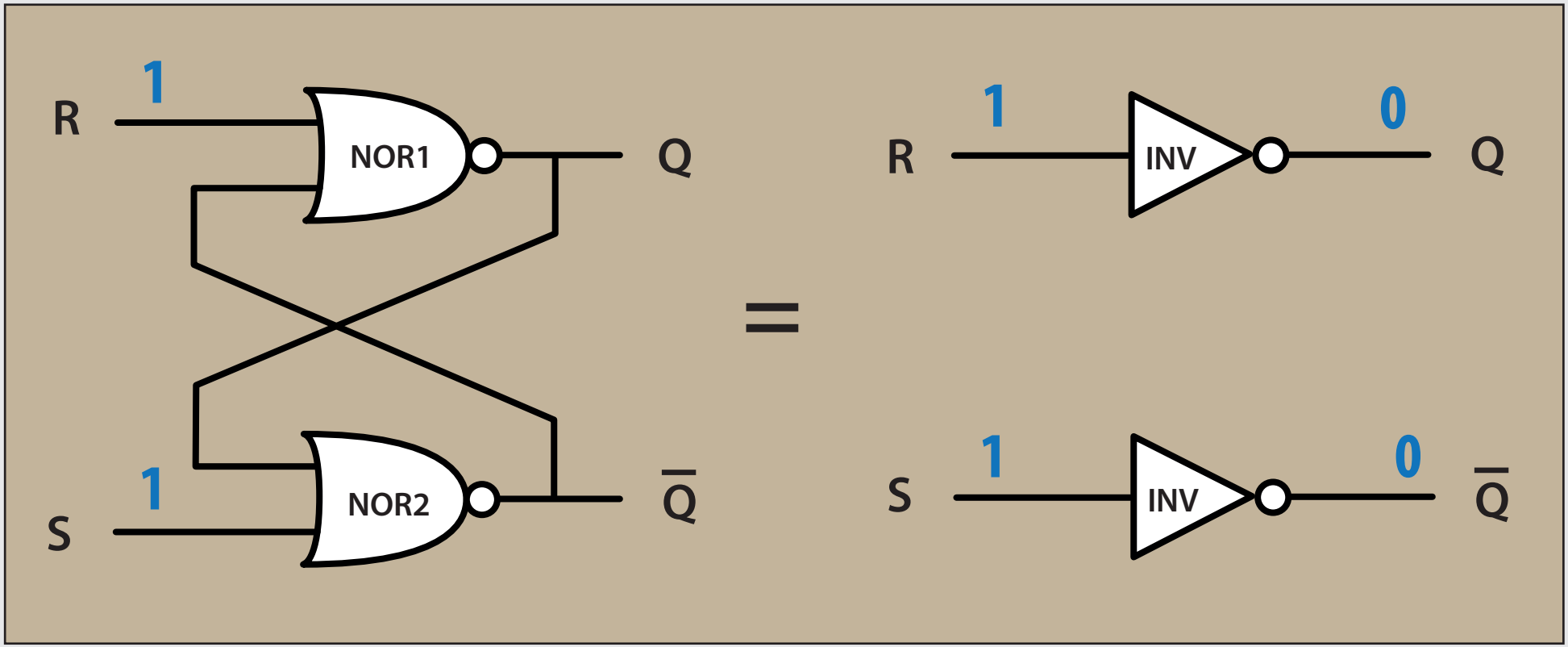
Application to the SR-latch
Set = 1 & Reset = 0



Application to the SR-latch
Set = 0 & Reset = 1



Application to the SR-latch
Set = 1 & Reset = 1

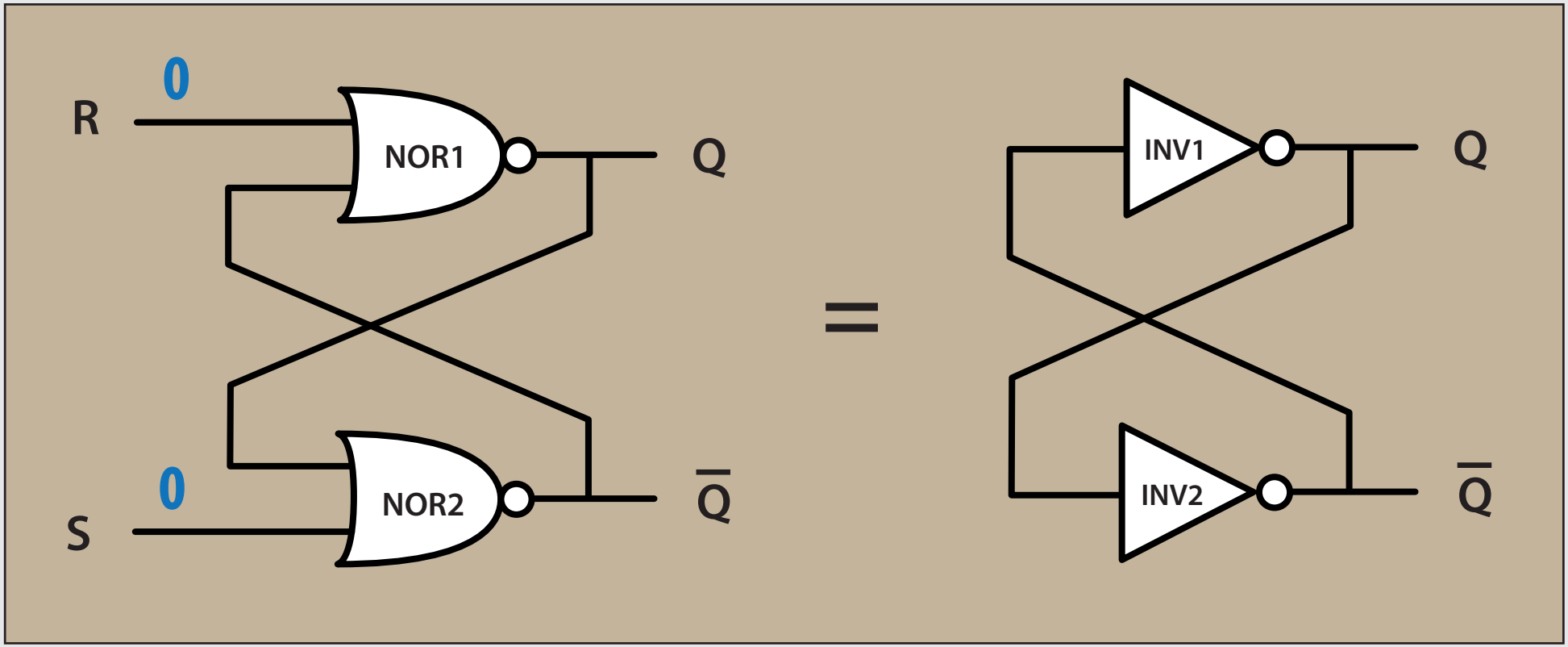


This configuration does not make much sense since :

- Set is designed to set the bit Q to 1
- Reset is designed to reset the bit Q to 0

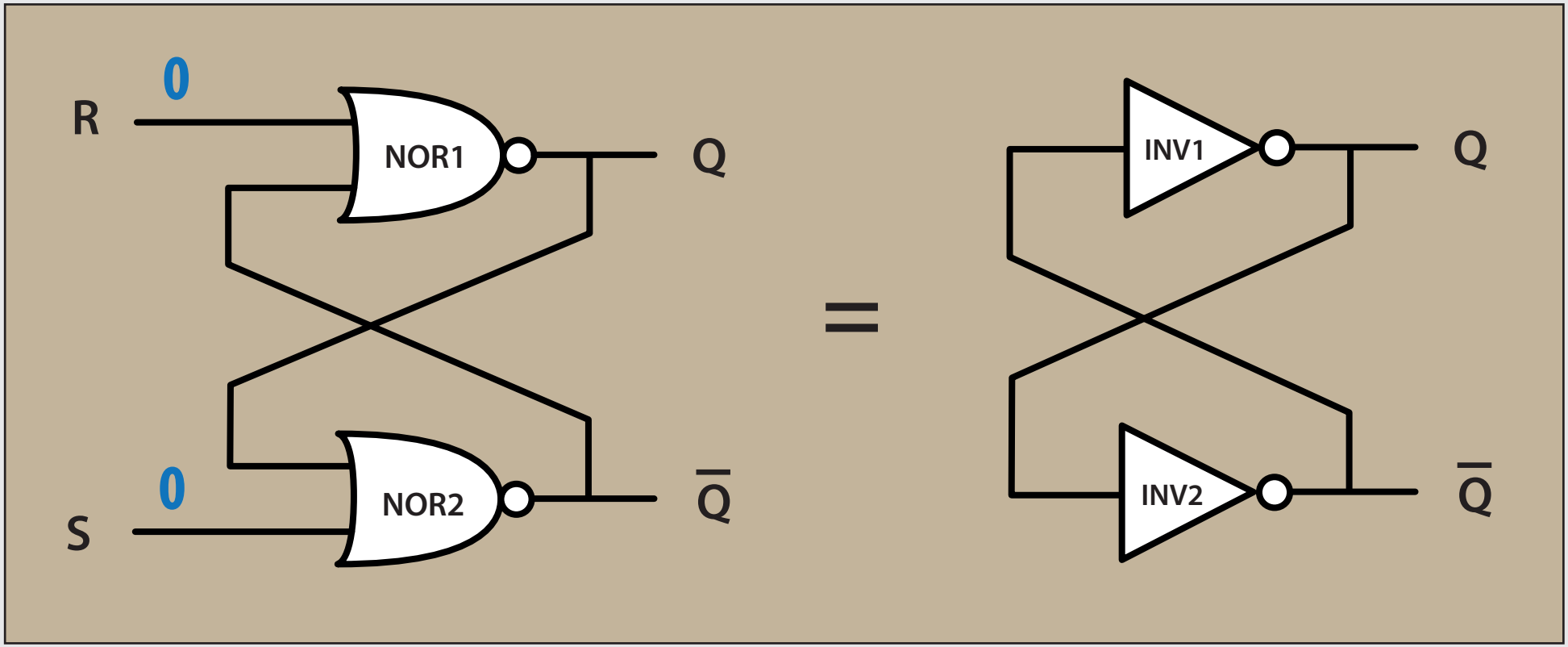
For that reason, this configuration is generally avoided in real-life situation.

Application to the SR-latch
Set = 0 & Reset = 0



The SR-latch behaves as a pair of cross-coupled inverters and ***thus remains in the last encountered stable state.***

Application to the SR-latch
Set = 0 & Reset = 0

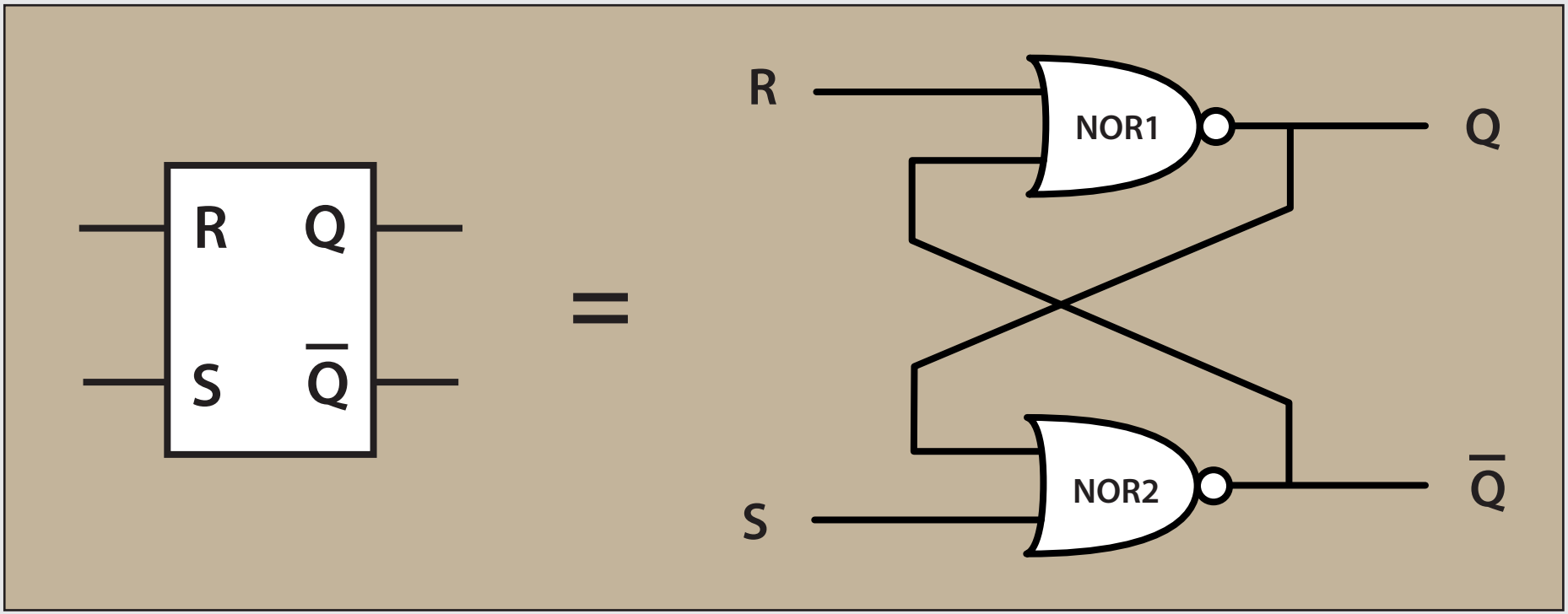


The SR-latch behaves as a pair of cross-coupled inverters and ***thus remains in the last encountered stable state.***

The « truth table » with memory of a SR-latch

S	R	Q	$\overline{Q}$	
0	0	previous state	previous state	keeps memory
0	1	0	1	reset memory bit to zero
1	0	1	0	set memory bit to one
1	1	0	0	usually avoided configuration

The SR latch symbol



The SR latch is such a basic and fundamental circuit that it has a symbol to represent it.

D latches
&&
D flip-flops

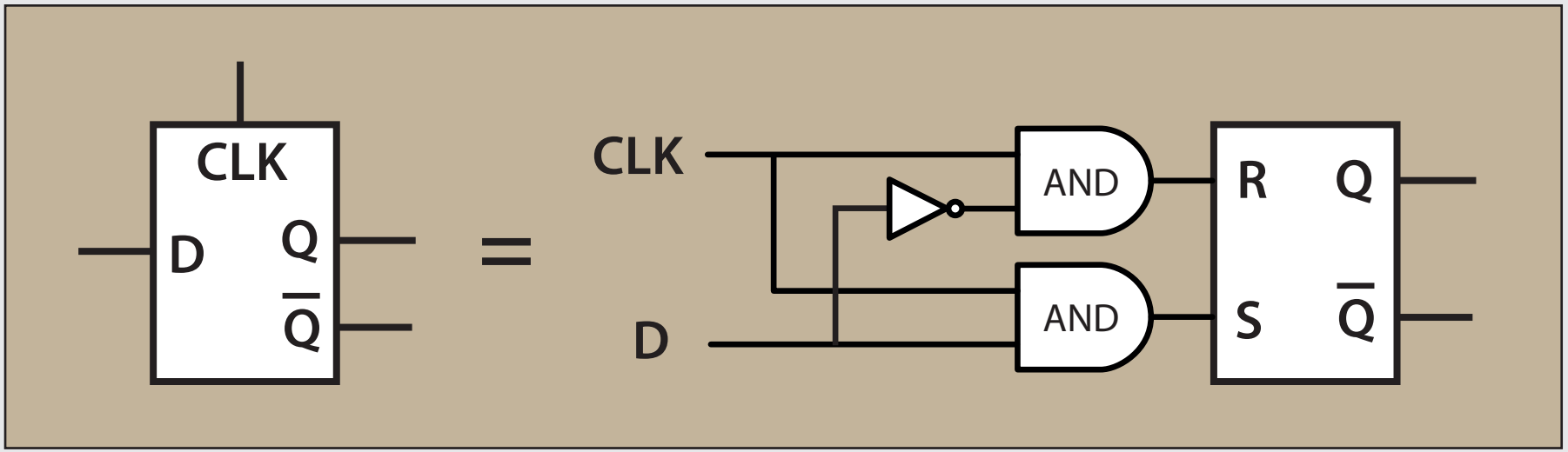
D latches

SR latches are awkward for two reasons :

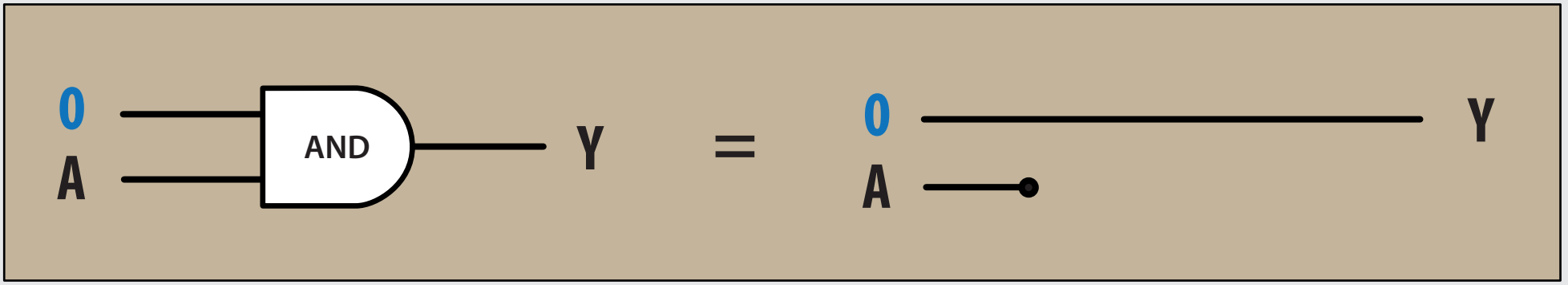
- the behavior when **S** = 1 and **R** = 1 is a bit unexpected
- the S and R inputs conflate the issues of « ***what*** » and « ***when*** » because asserting one of the inputs determines not only ***what*** the state should be but also ***when*** it should change...

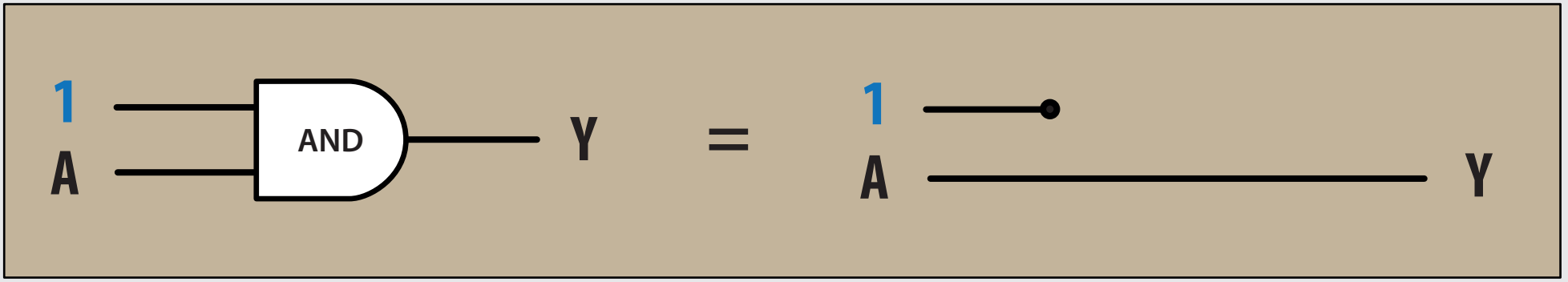
In order to solve these problems, the D latch has two inputs :

- the ***data*** input **D** controls ***what*** the next state should be
- the ***clock*** input **CLK** controls ***when*** the state should change

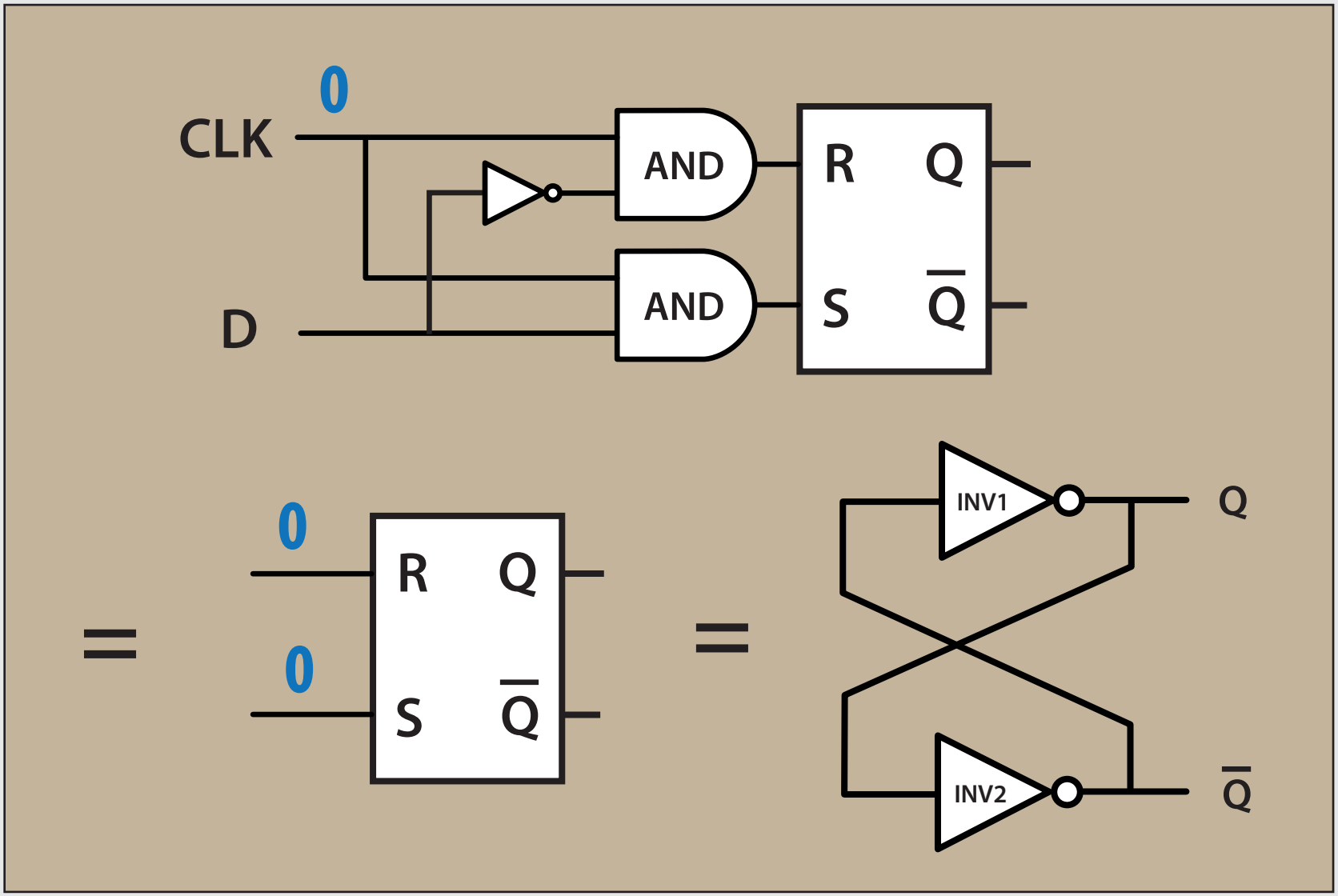


Two other useful equalities to remember





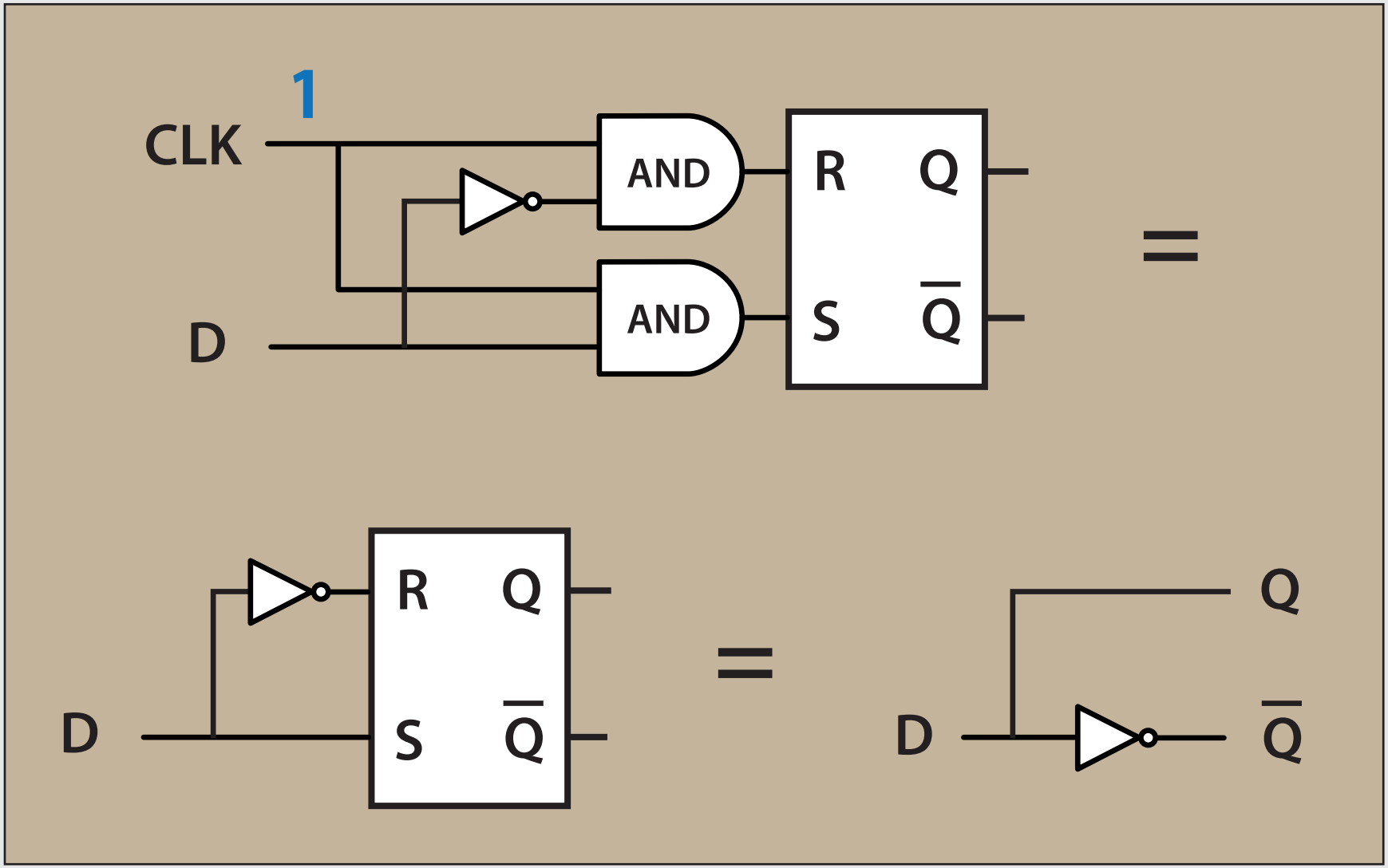
Application to the D latch
CLK = 0



The D latch thus behaves as pair of cross-coupled inverters when CLK = 0

One says in that case that the D latch is « opaque » since it retains the old value of Q

Application to the D latch
CLK = 1



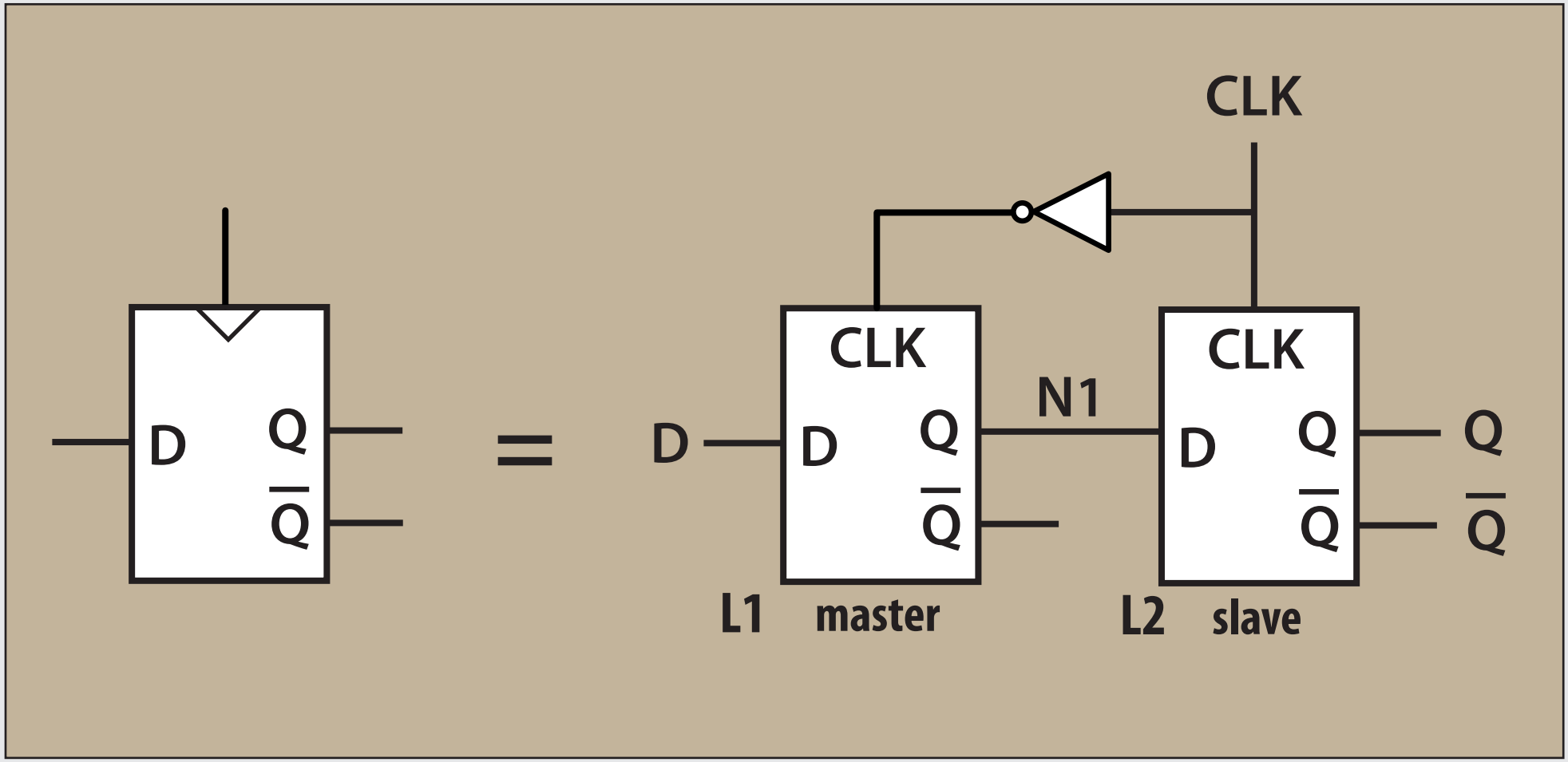
The D latch thus behaves as the « identity » when CLK = 1

One says in that case that the D latch is transparent

The D flip-flop

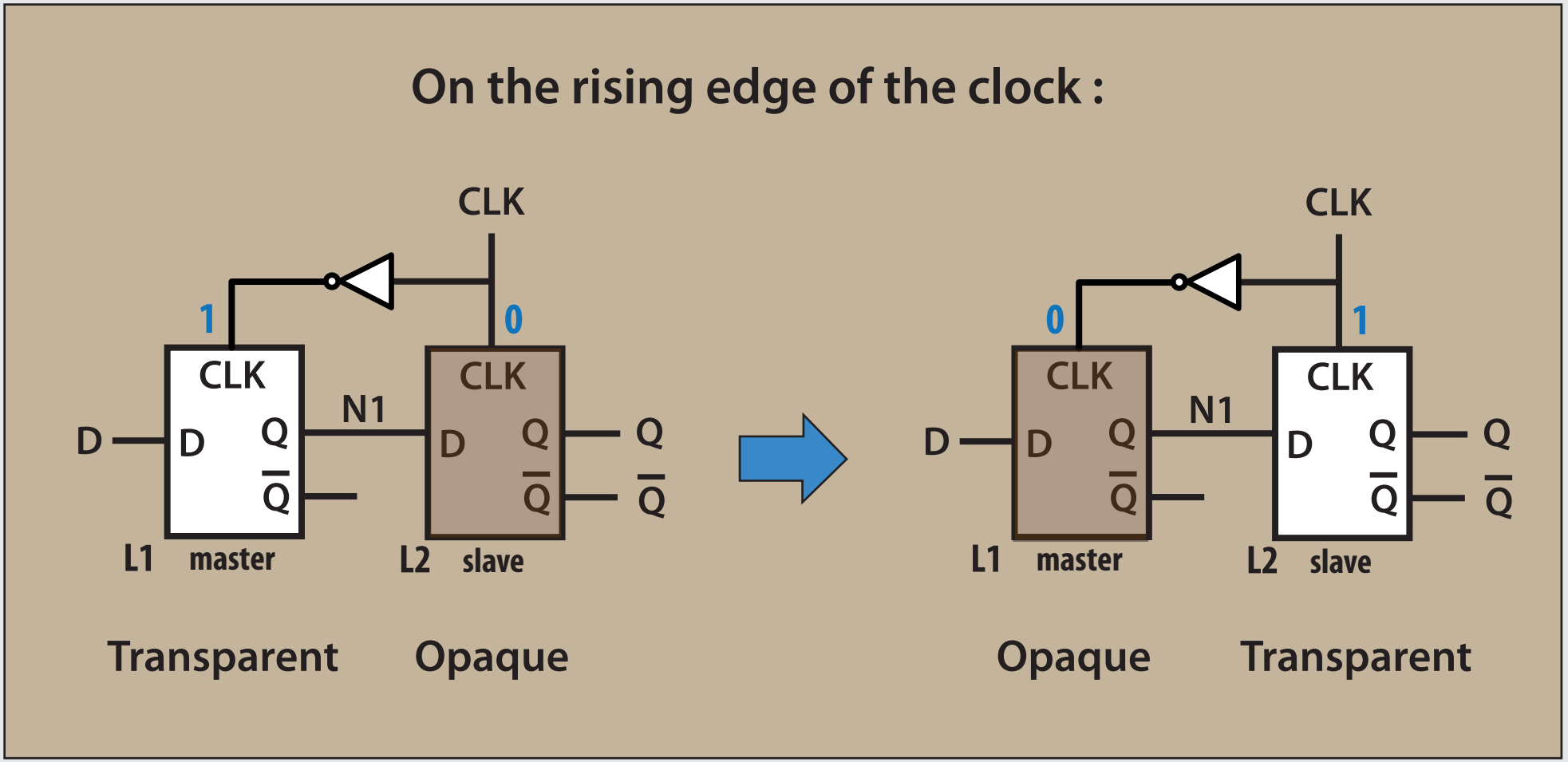
A D flip-flop can be built from two back-to-back D latches controlled by complementary clocks :

- the ***master*** latch **L1** transparent when **CLK** = 0
- the ***slave*** latch **L2** transparent when **CLK** = 1



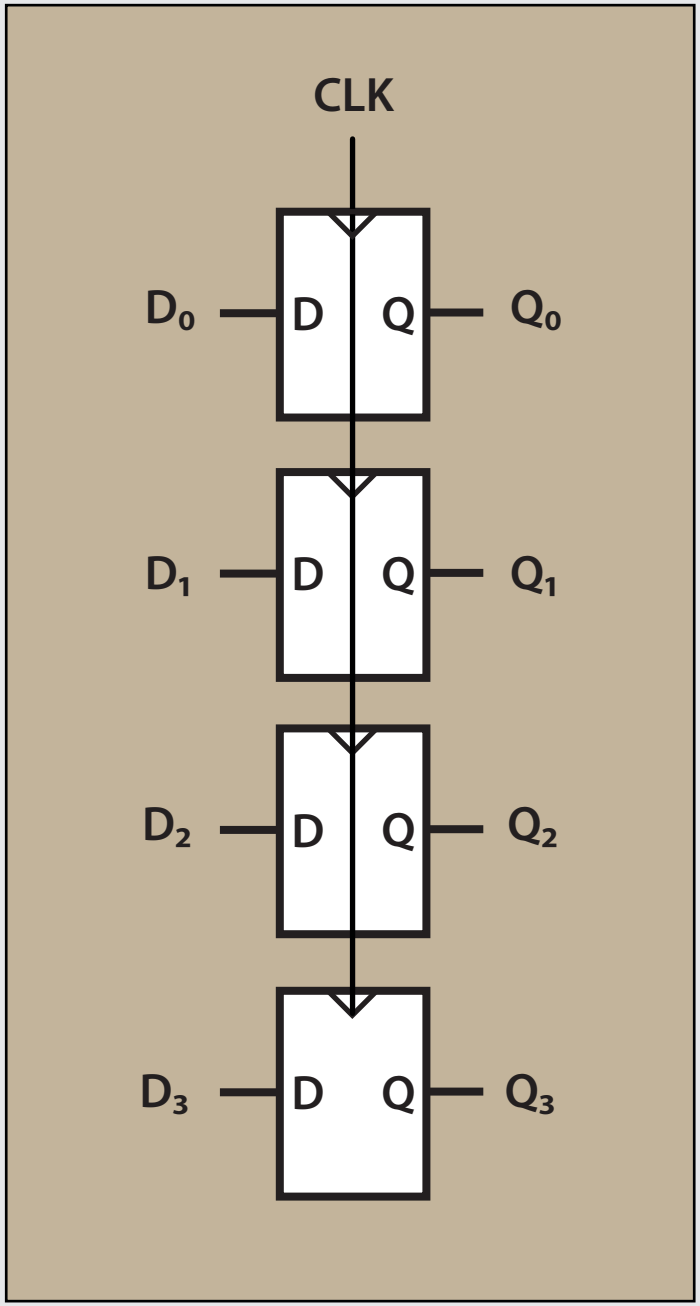
The D flip-flop

A D flip-flop copies **D** to **Q** on the rising edge of the clock and remembers its state at all other times.

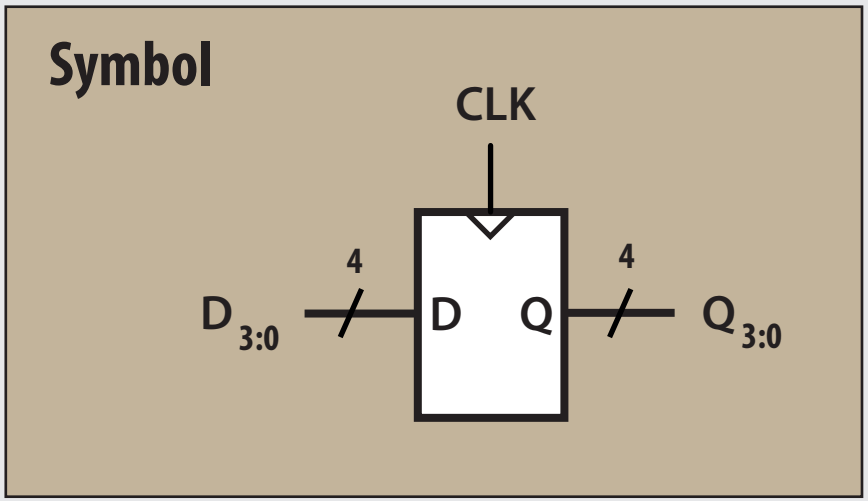


The D flip-flop is also called ***master-slave*** flip-flop or ***positive-edge triggered*** flip-flop

The N-bit Register



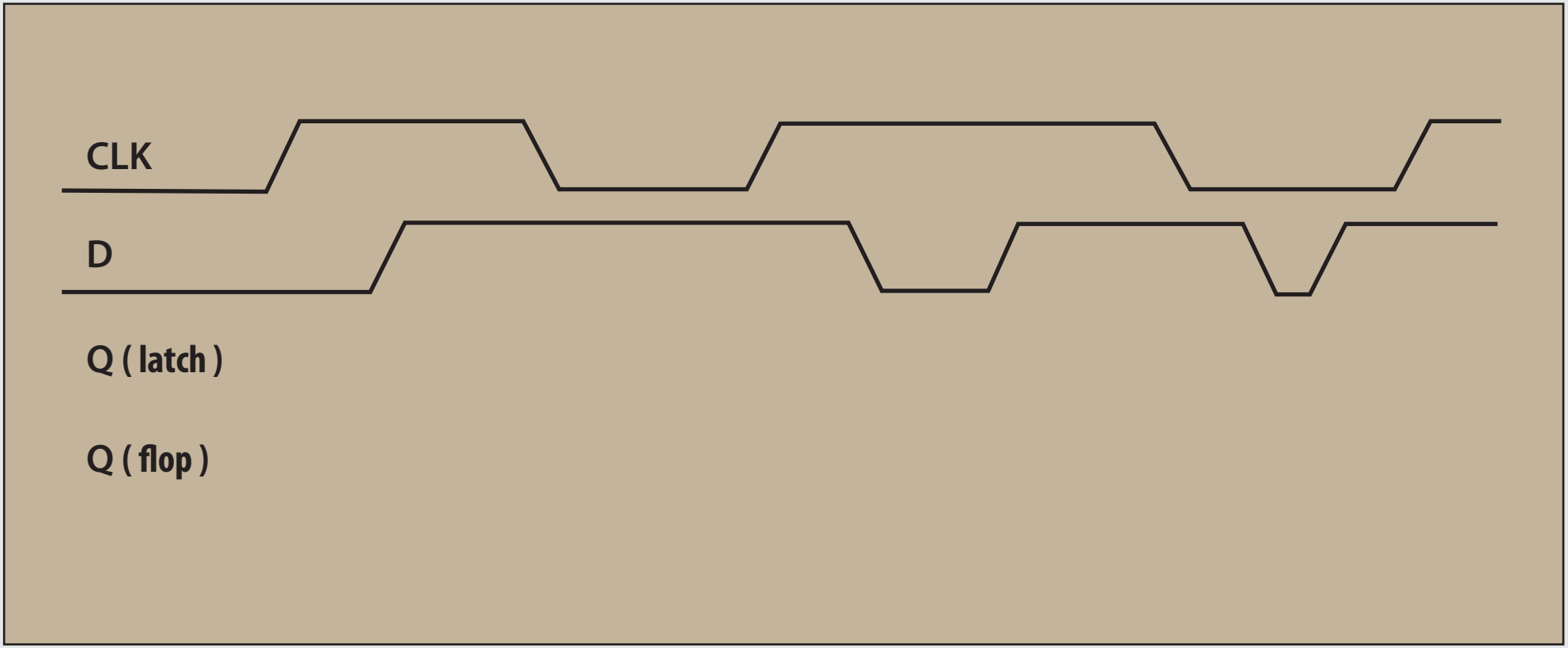
Simply defined as a series of N positive-edge triggered D flip-flops all sharing the same input CLK



Exercise 1

H & H Example 3.2

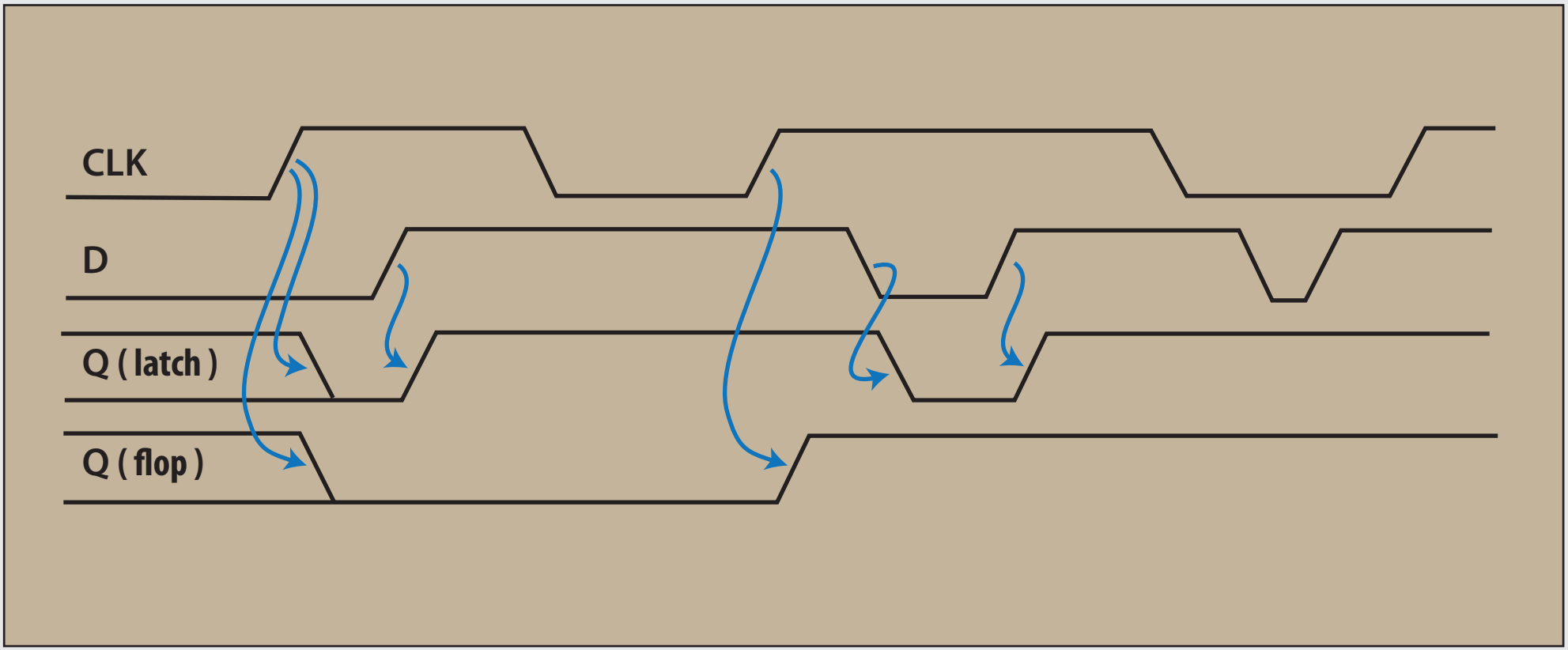
Ben Bitdiddle applies the D and the CLK inputs shown below to a D latch and to a D flip-flop. Help him determine the output Q of each device.



Solution

H & H Example 3.2

Ben Bitdiddle applies the D and the CLK inputs shown below to a D latch and to a D flip-flop. Help him determine the output Q of each device.

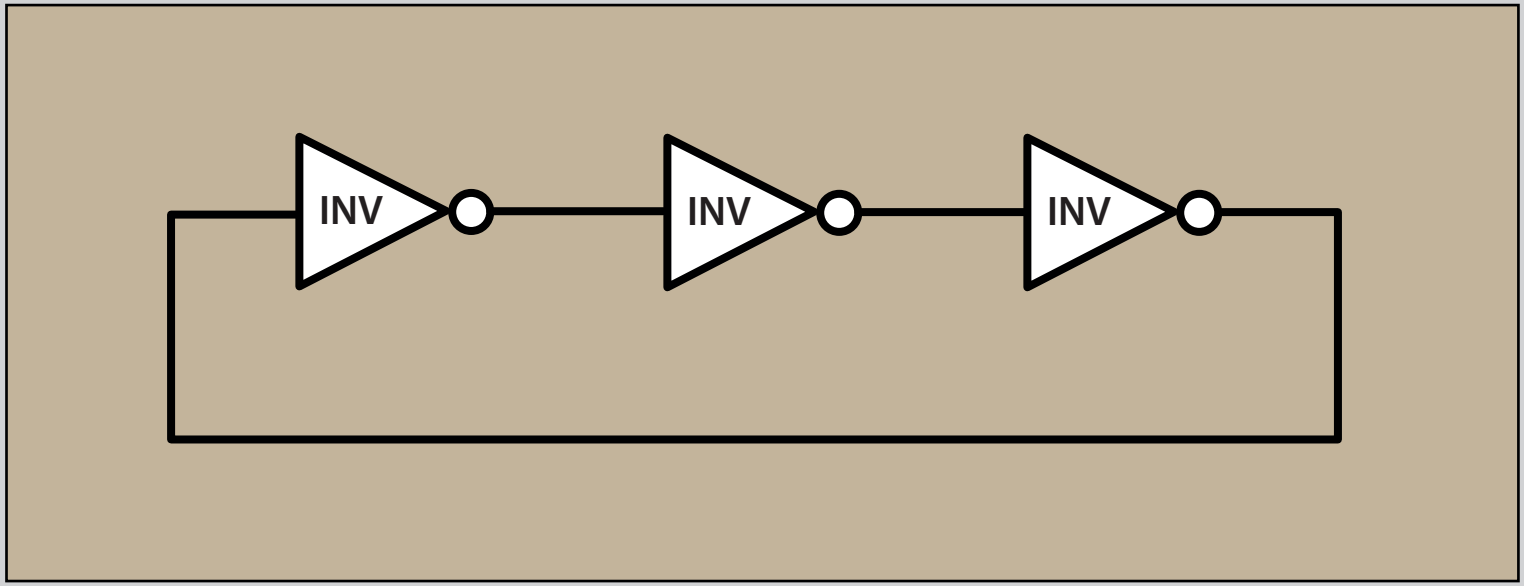


Sequential Circuits

Problematic circuits
1. astable circuits

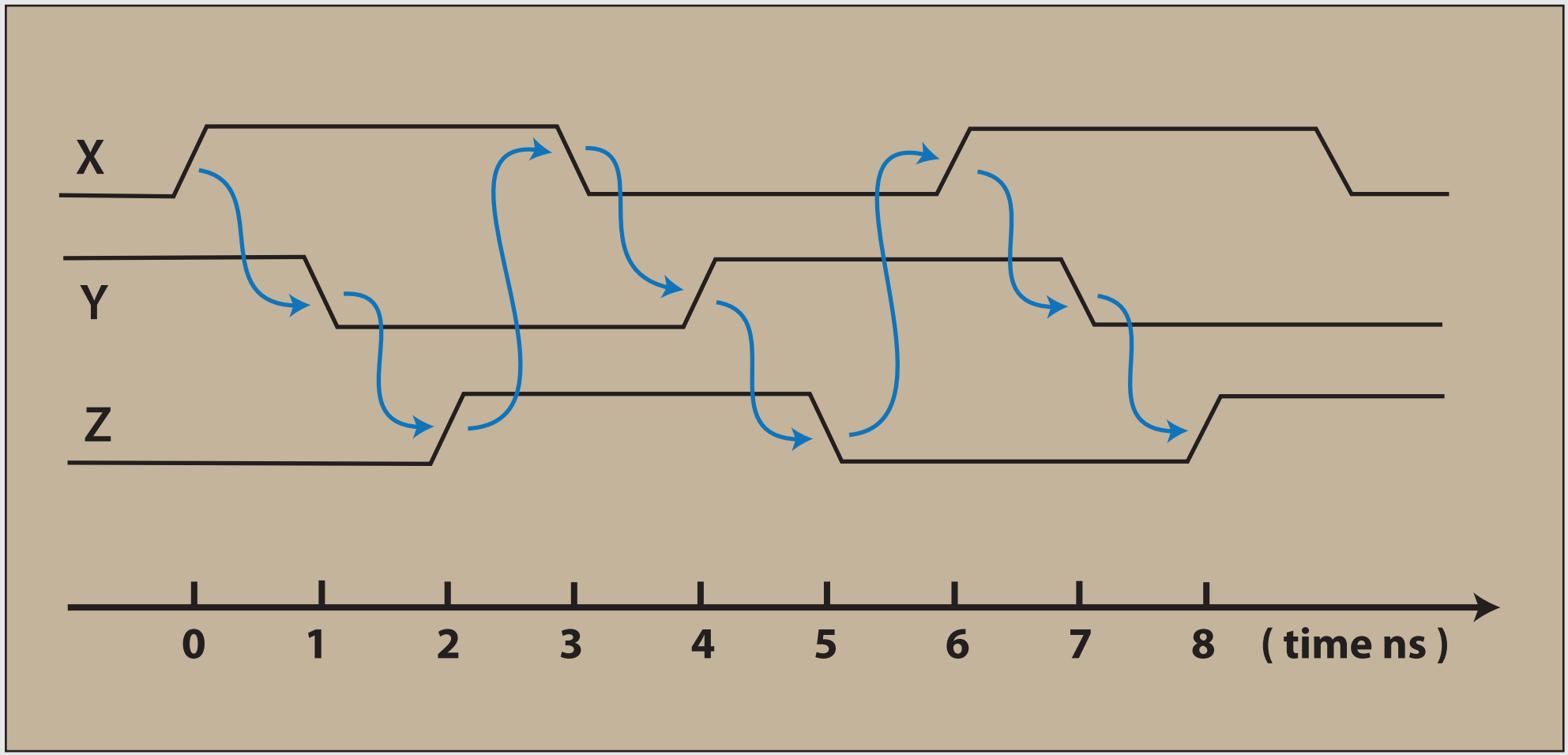
Exercise :

Suppose that each inverter in the circuit below has delay 1 ns and determine what the circuit does.

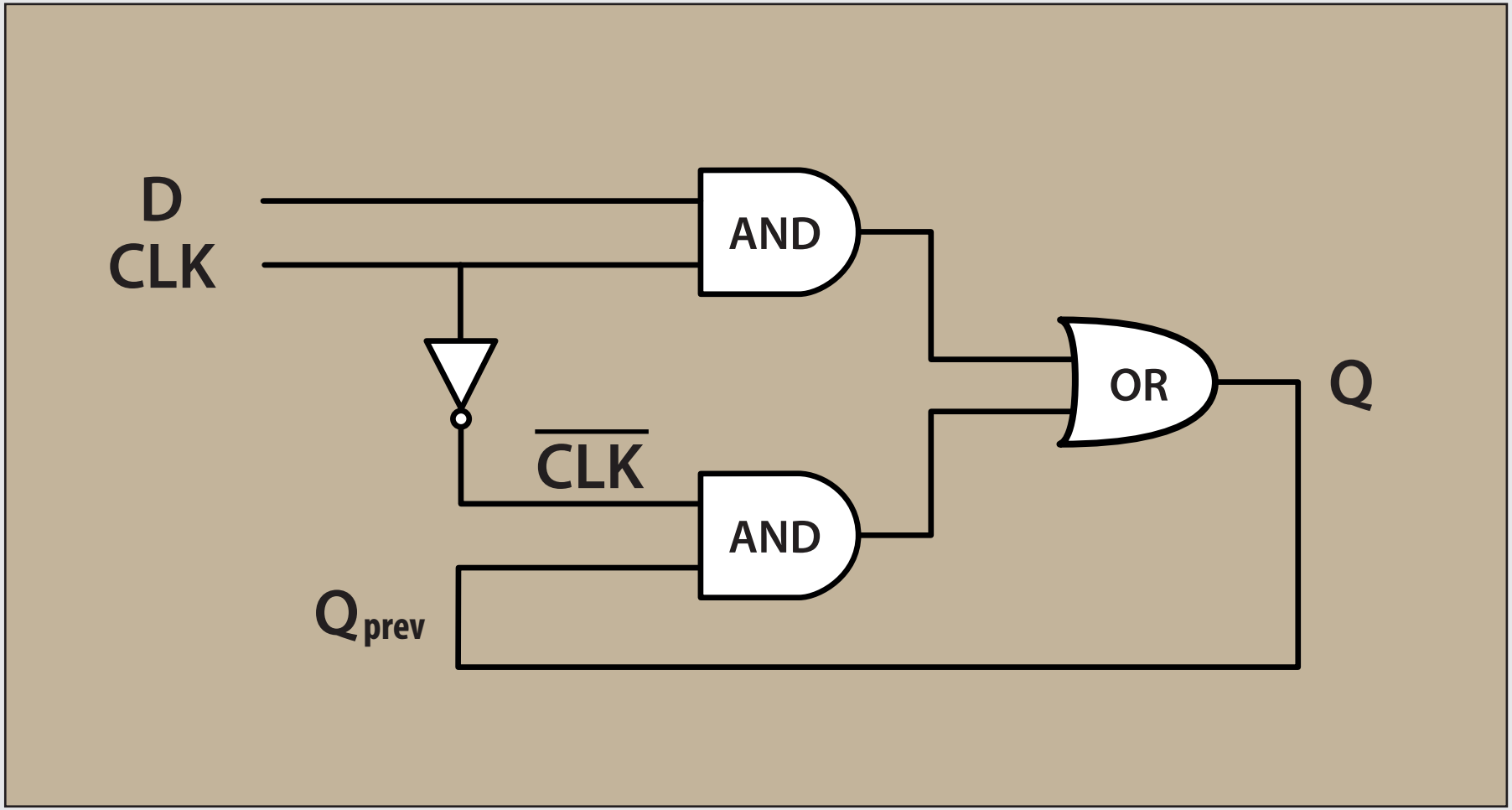


Solution

Suppose that each inverter in the circuit below has delay 1 ns and determine what the circuit does.



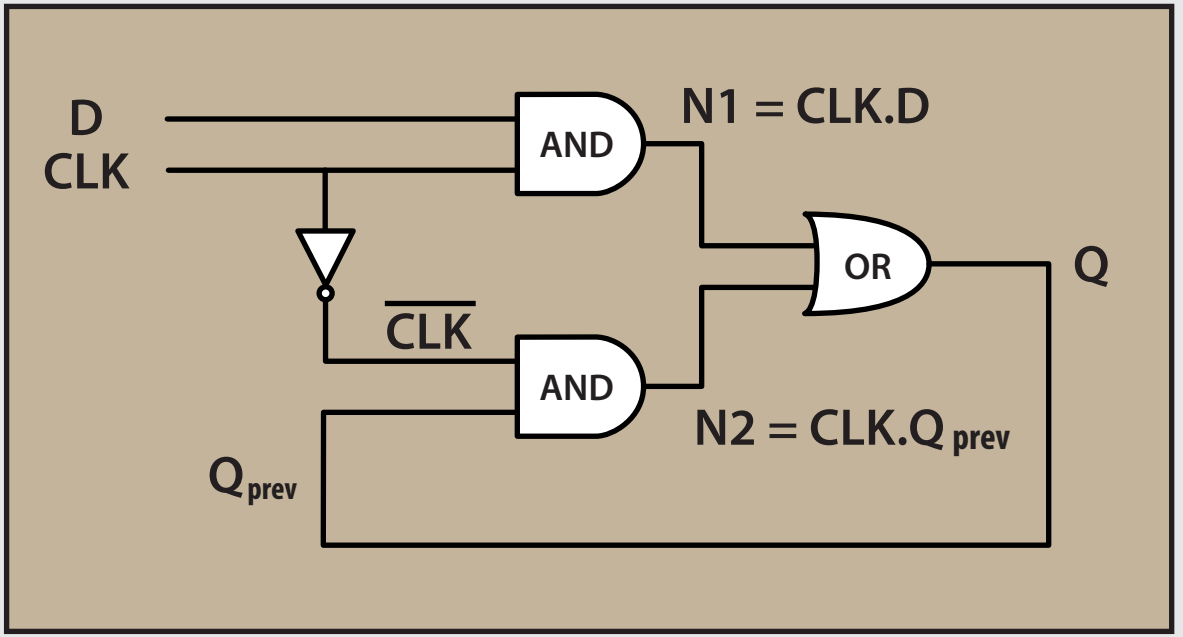
Problematic circuits
2. race conditions



$$Q = CLK \,.\, D + \overline{CLK} \,.\, Q_{prev}$$

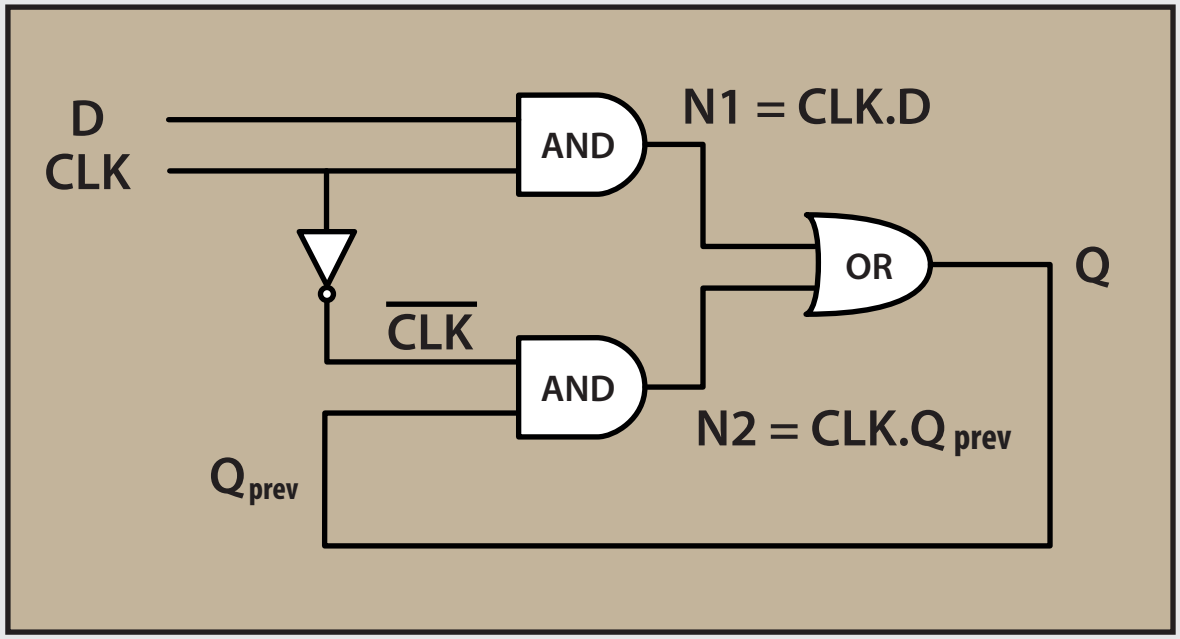
Problematic circuits
2. race conditions

CLK	D	Q_{prev}	Q
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



$$Q = CLK \,.\, D + \overline{CLK} \,.\, Q_{prev}$$

Race conditions



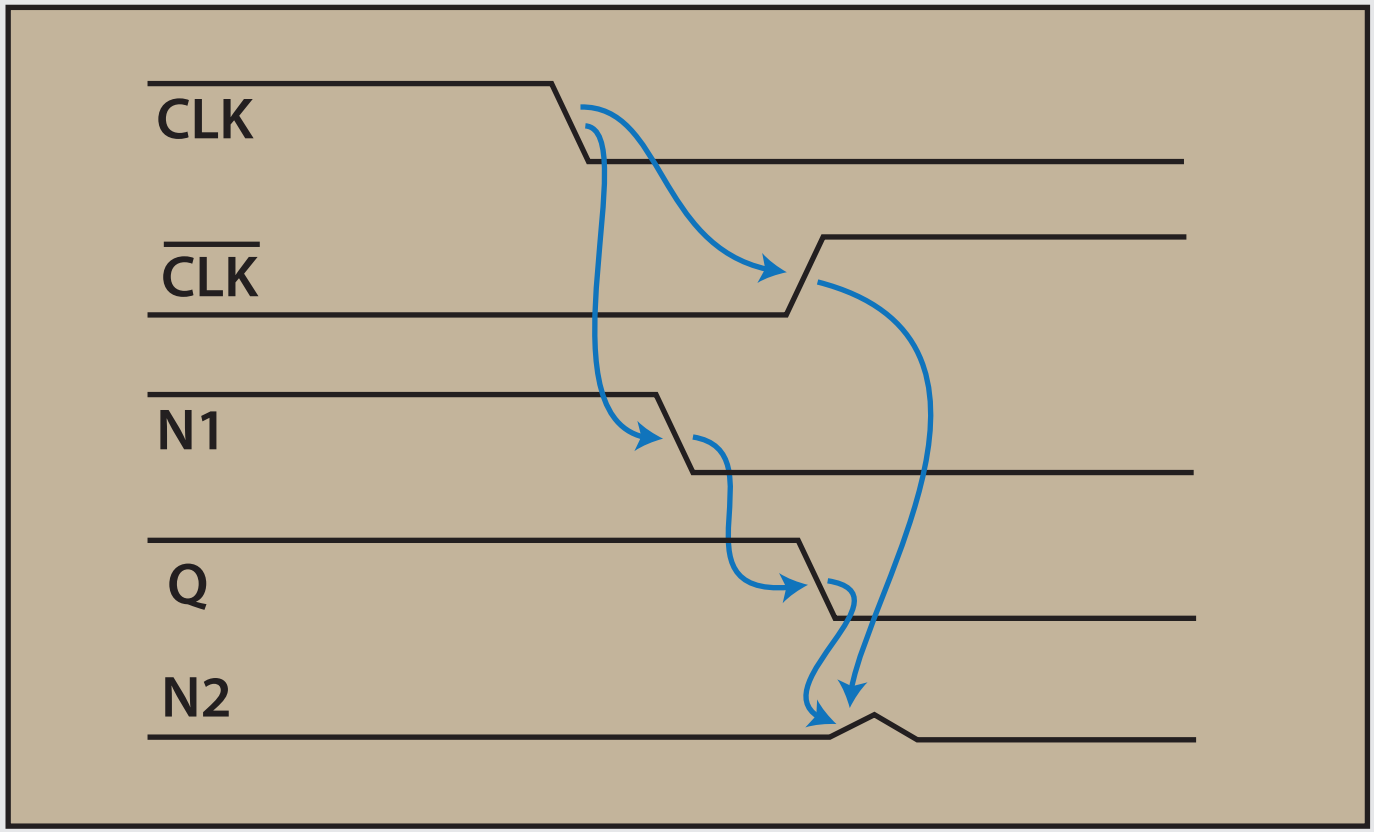
Suppose that D = CLK =1

The latch is transparent
and thus: Q =1.

Now, suppose that CLK falls.

Then, if the delay of the inverter
is too long compared to the delays
of the AND and OR gates,
then N1 and Q may fall
before $\overline{CLK}$ rises.

In that case, N2 will never rise
and Q becomes stuck to 0



Synchronous Sequential Circuits

Synchronous Sequential Circuits

A circuit is ***sequential synchronous*** when :

- every circuit element is either a register or a combinational circuit
- at least one circuit element is a register
- all registers receive the same clock signal
- every cyclic path contains at least one register.

Sequential circuits that are not synchronous are called asynchronous.

Exercise

Which circuits below are ***sequential synchronous*** ?