University at Buffalo
The State University of New York

cse@buffalo

CSE115 / CSE503
Introduction to Computer Science I

Dr. Carl Alphonce

343 Davis Hall

alphonce@buffalo.edu

Office hours:

Tuesday 10:00 AM – 12:00 PM*

Wednesday 4:00 PM – 5:00 PM

Friday 11:00 AM – 12:00 PM

OR request appointment via e-mail

**Tuesday adjustments: 11:00 AM – 1:00 PM on 10/11, 11/1 and 12/6*

ROADMAP

Today

- Announcements: Scientista & ACM / Notetaker
- Polls
- Gates
- Memory
- Basic computer organization

Coming up

- Instruction decoding
- Fetch/Decode/Execute cycle
- Low-level and high-level languages
- Expressions and objects

Scientista meeting

Th @ 4:30

Davis 1st floor lounge

facebook.com/ubscientista

FREE FOOD!!

ACM meeting
Th @ 5:30
Davis 2nd floor lounge
ubacm.org
No free food, BUT BBQ Fri!

Notetaker request:

Thanks to everyone: no more volunteers needed!

POLL

PROFESSIONALISM

Until we do an interactive exercise or a poll, turn off and put away electronics:

- cell phones
- pagers
- laptops
- tablets
- etc.

REVIEW

physical vs. logical perspectives

Physical reality:

WIRE

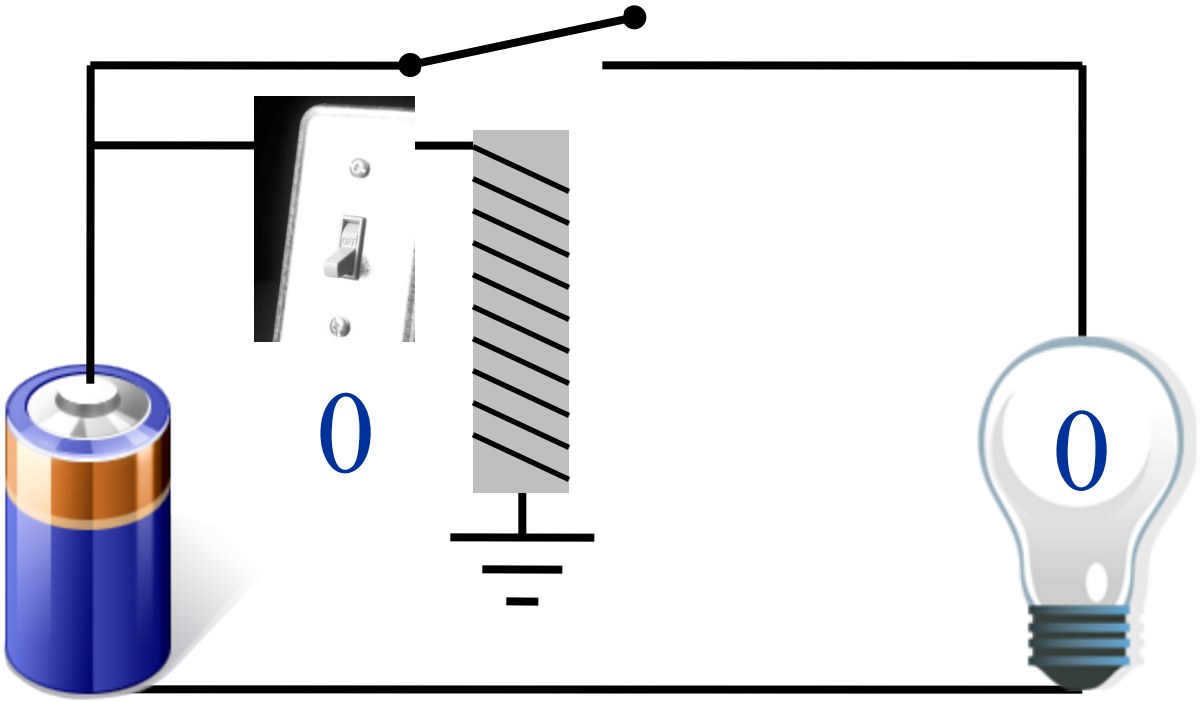
Carries a HIGH voltage or a LOW voltage

Logical view:

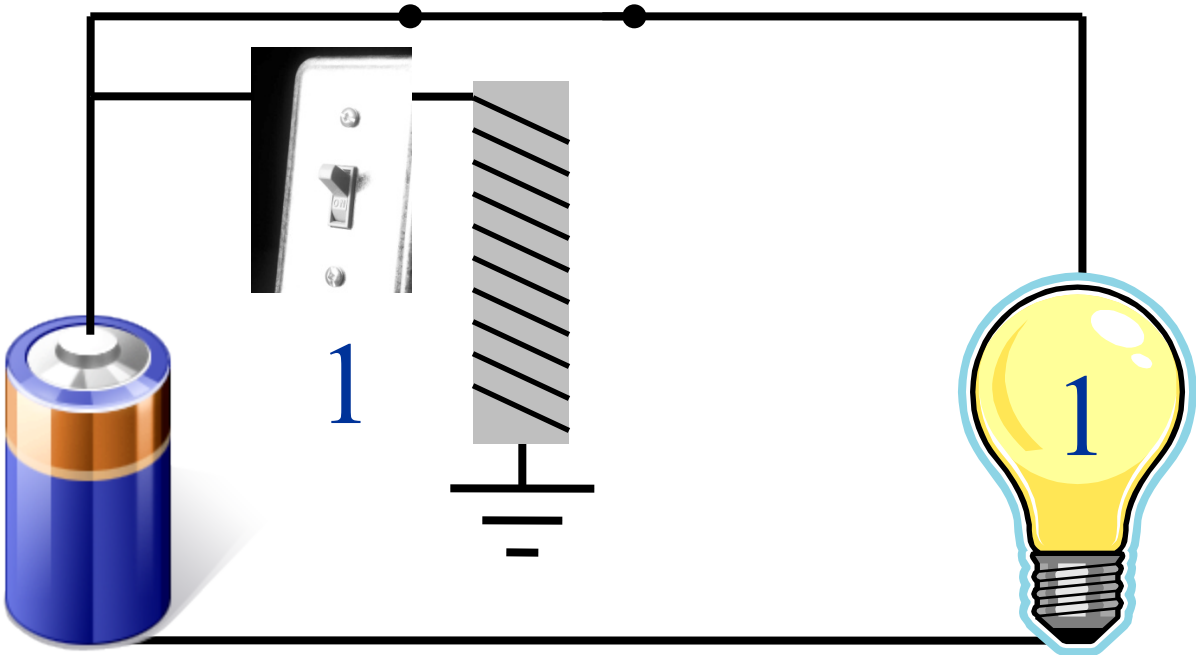
WIRE

Carries a 1 or a 0

Controlling flow (normally open)



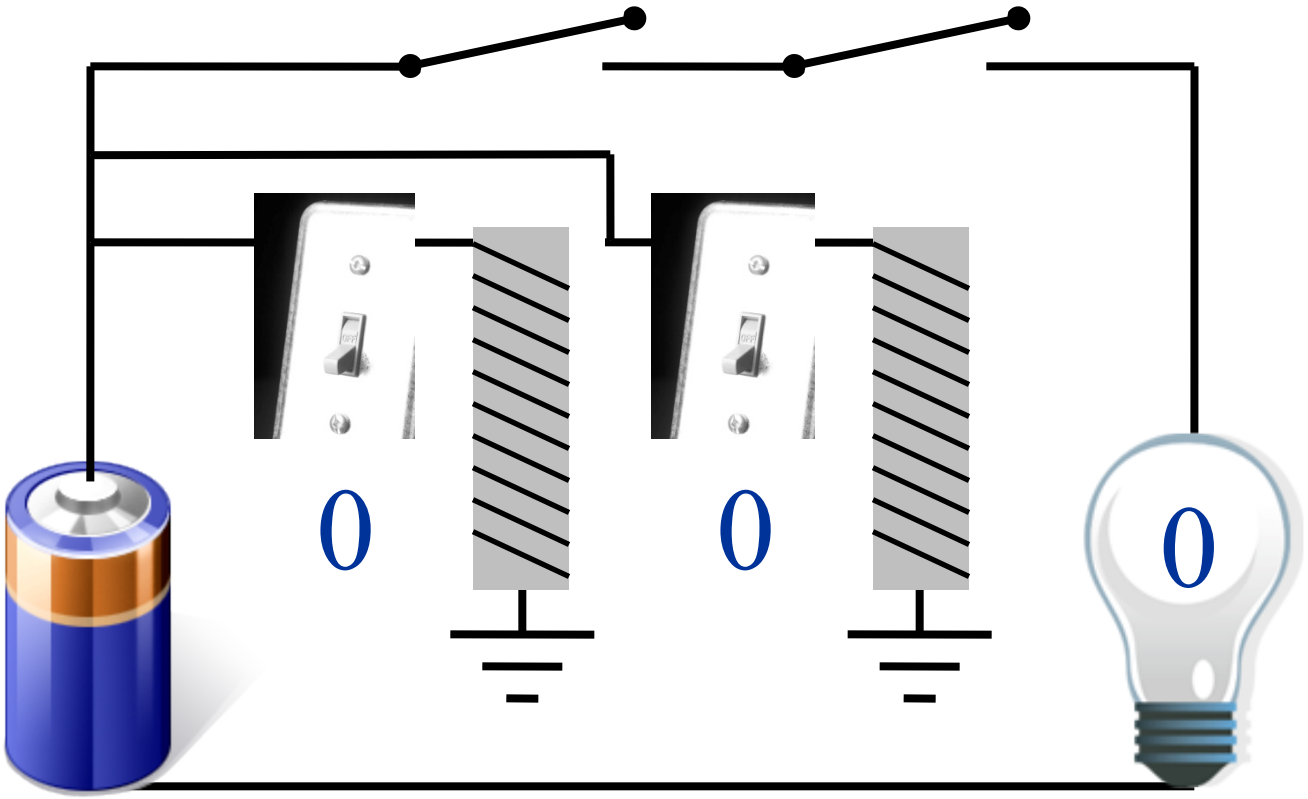
Controlling flow (normally open)



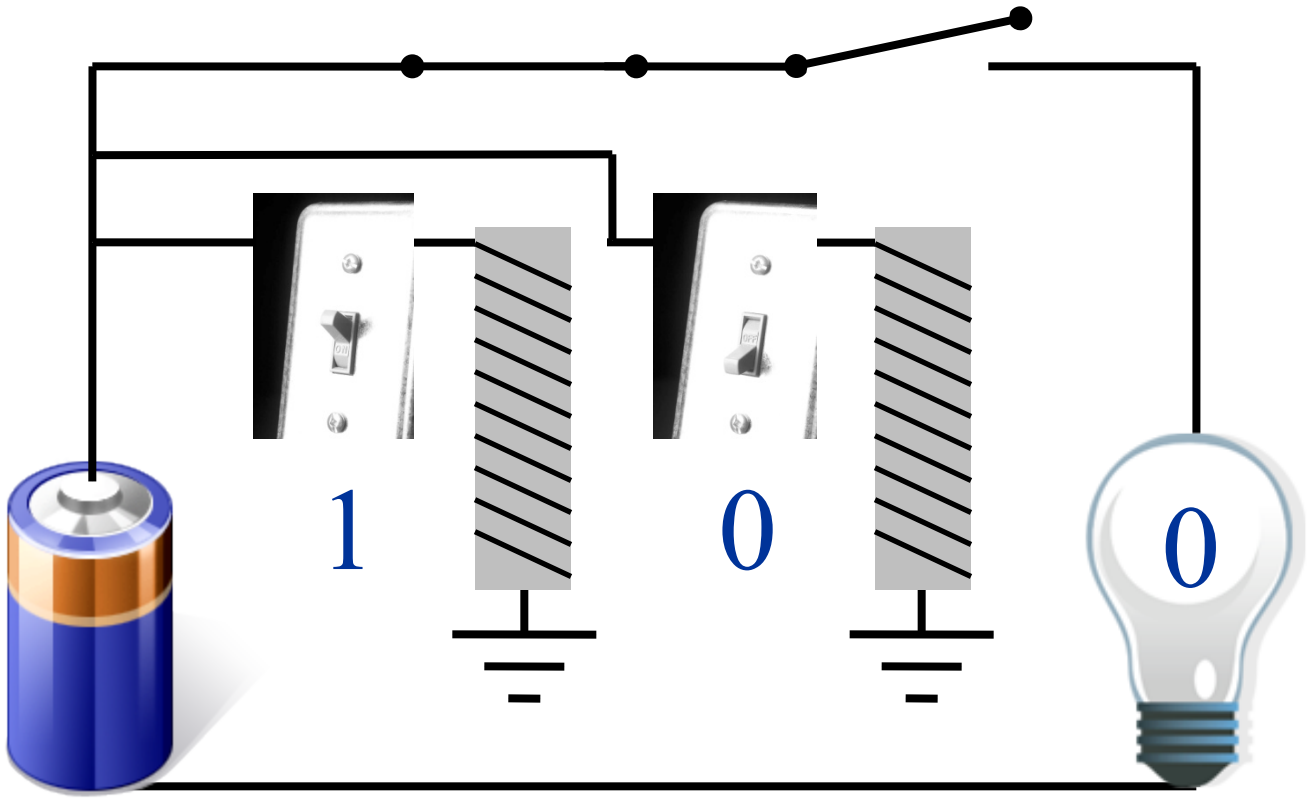
Interesting links

http://history-computer.com/ModernComputer/Basis/relay.html

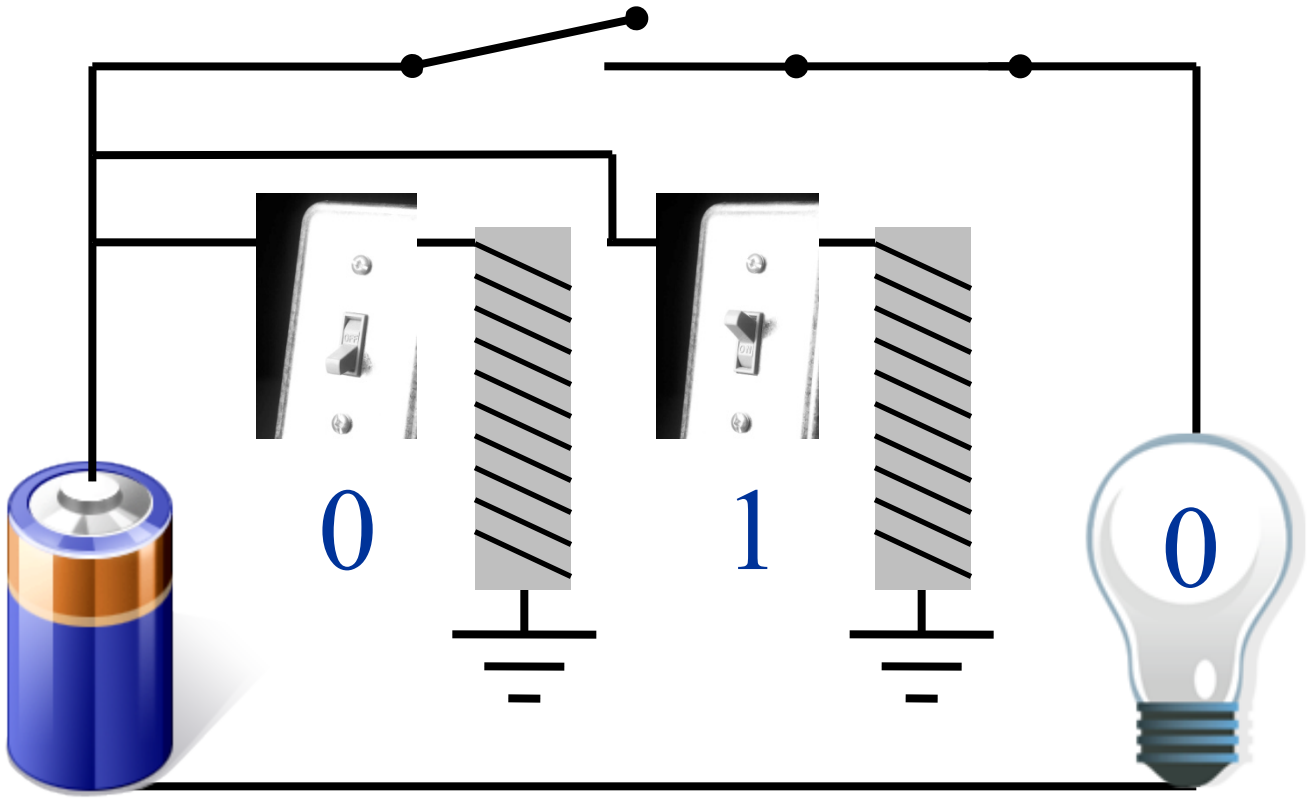
Two in a row (in series)?



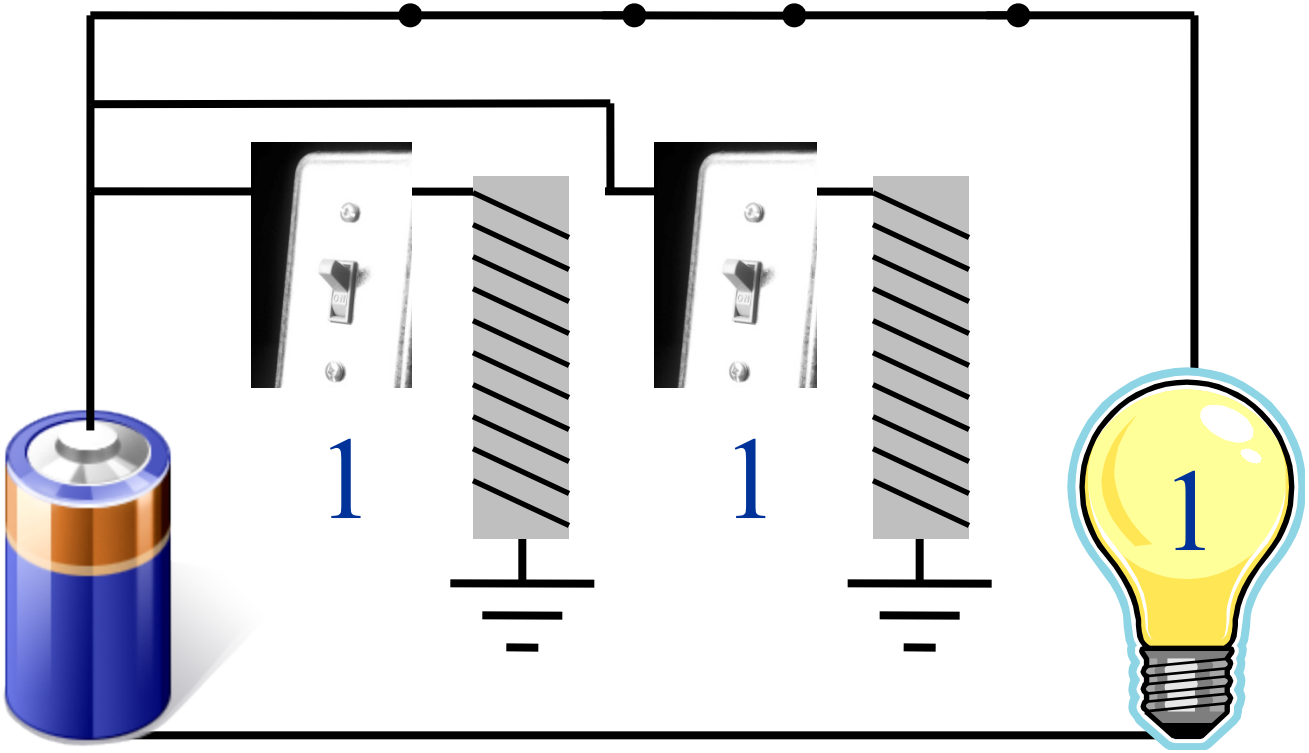
Two in a row (in series)?



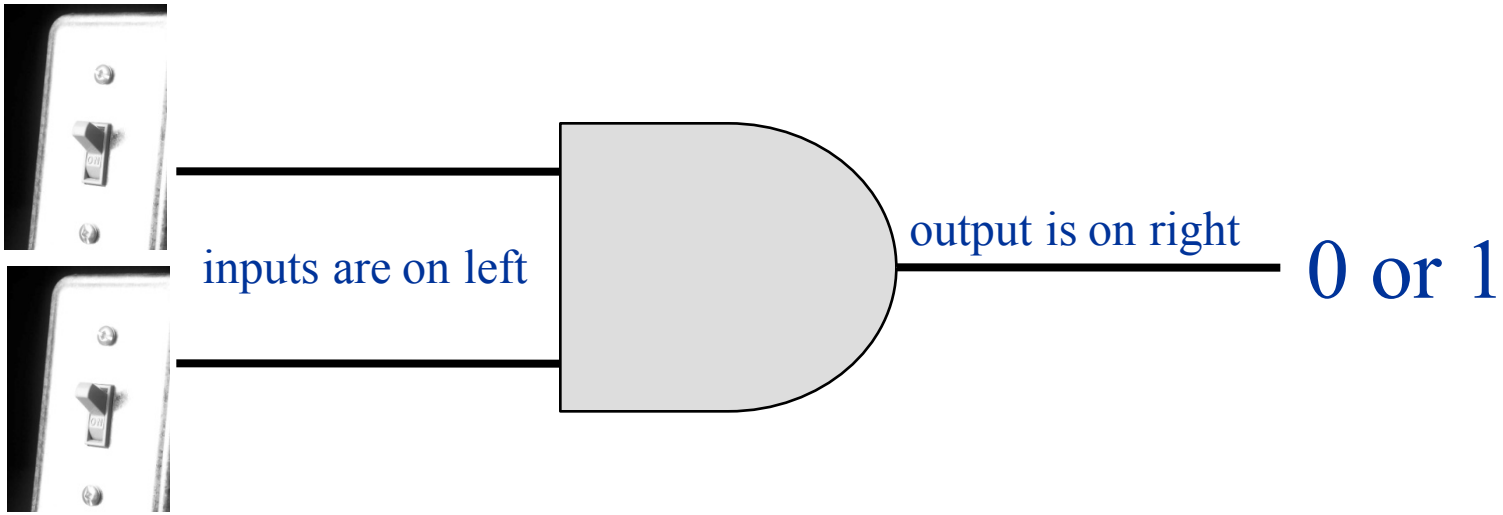
Two in a row (in series)?



Two in a row (in series)?



AND gate



For which input values is output 1?

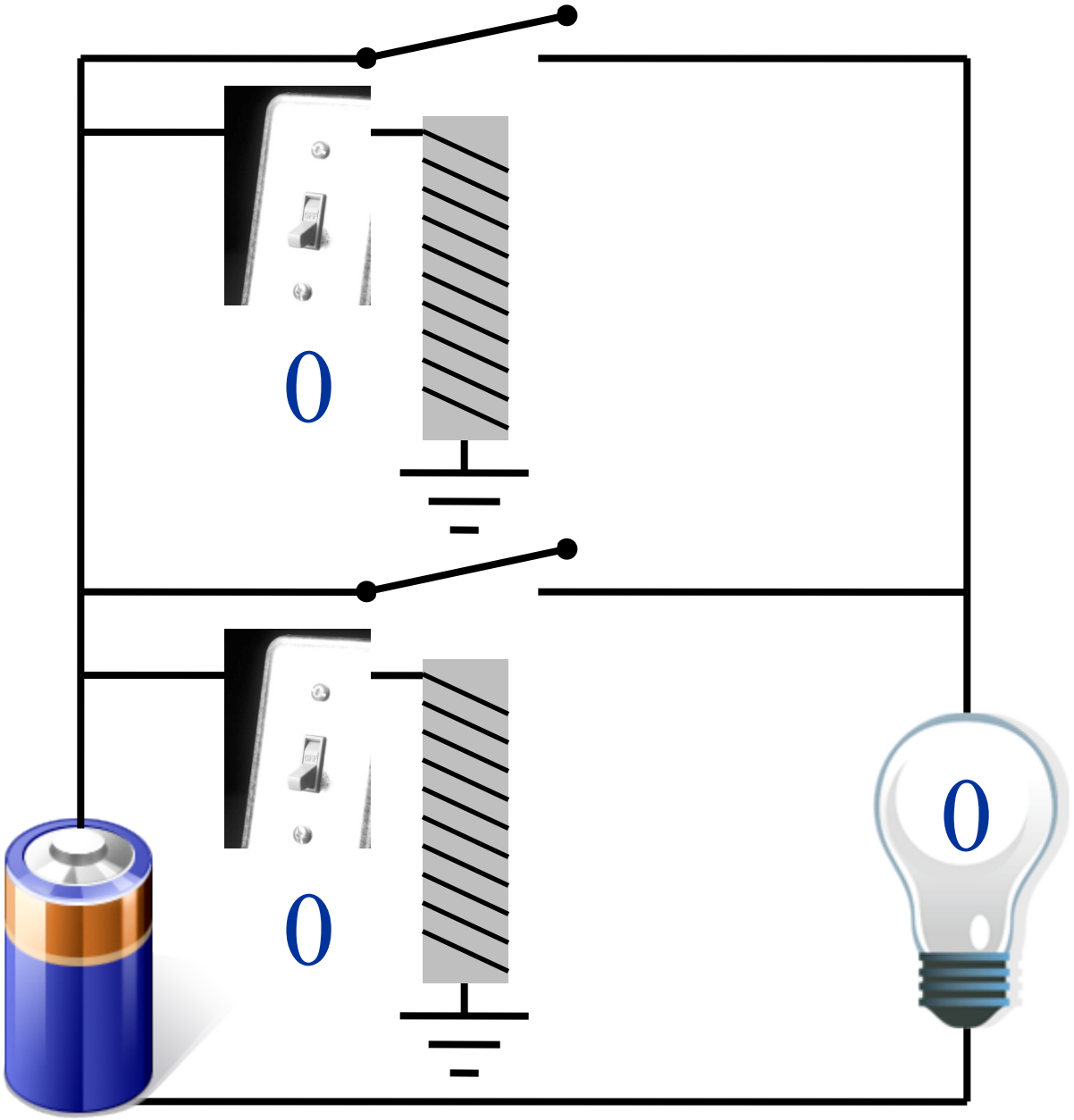
For which input values is output 0?

Truth table for AND

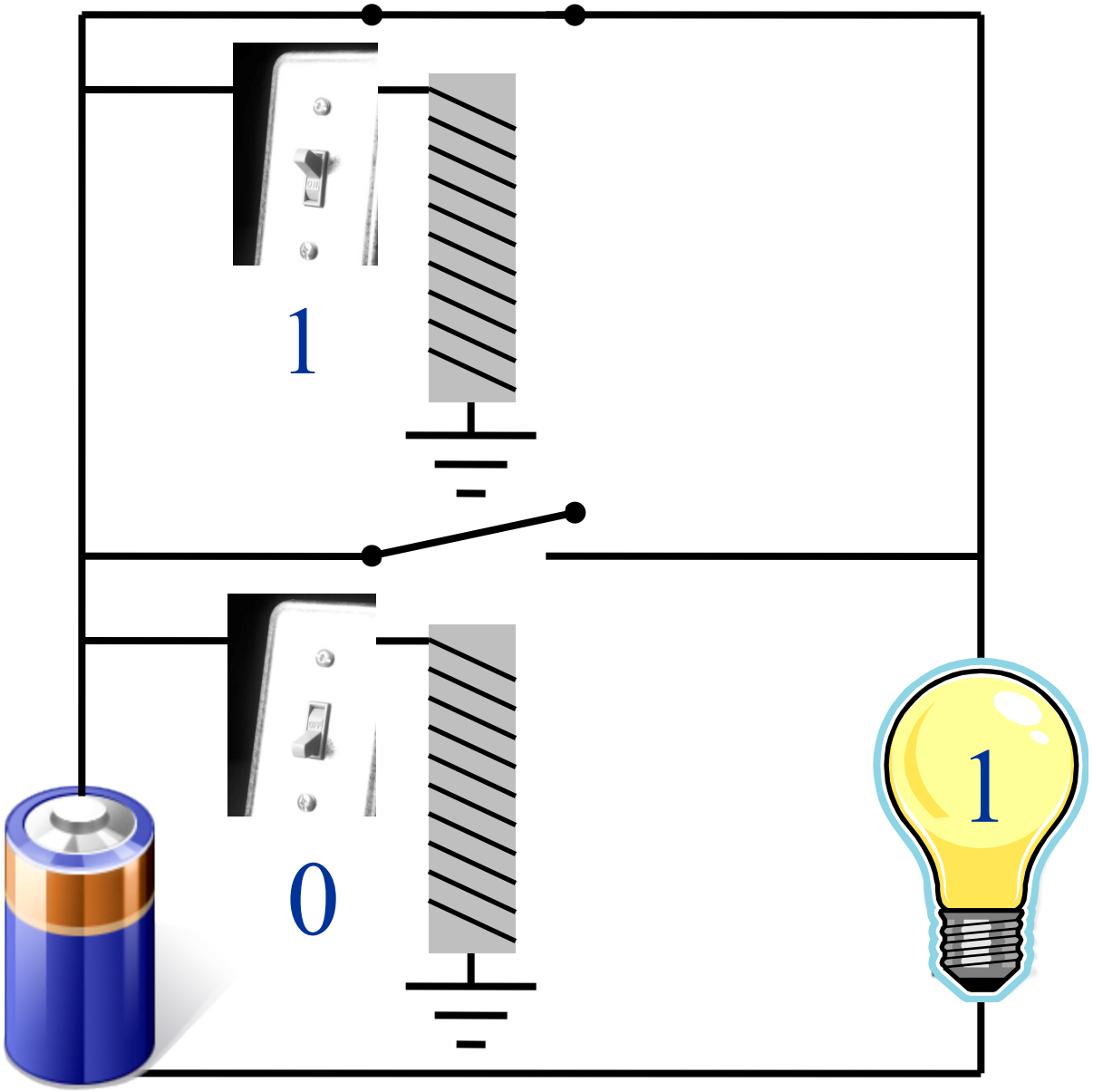
Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

MOVING ON

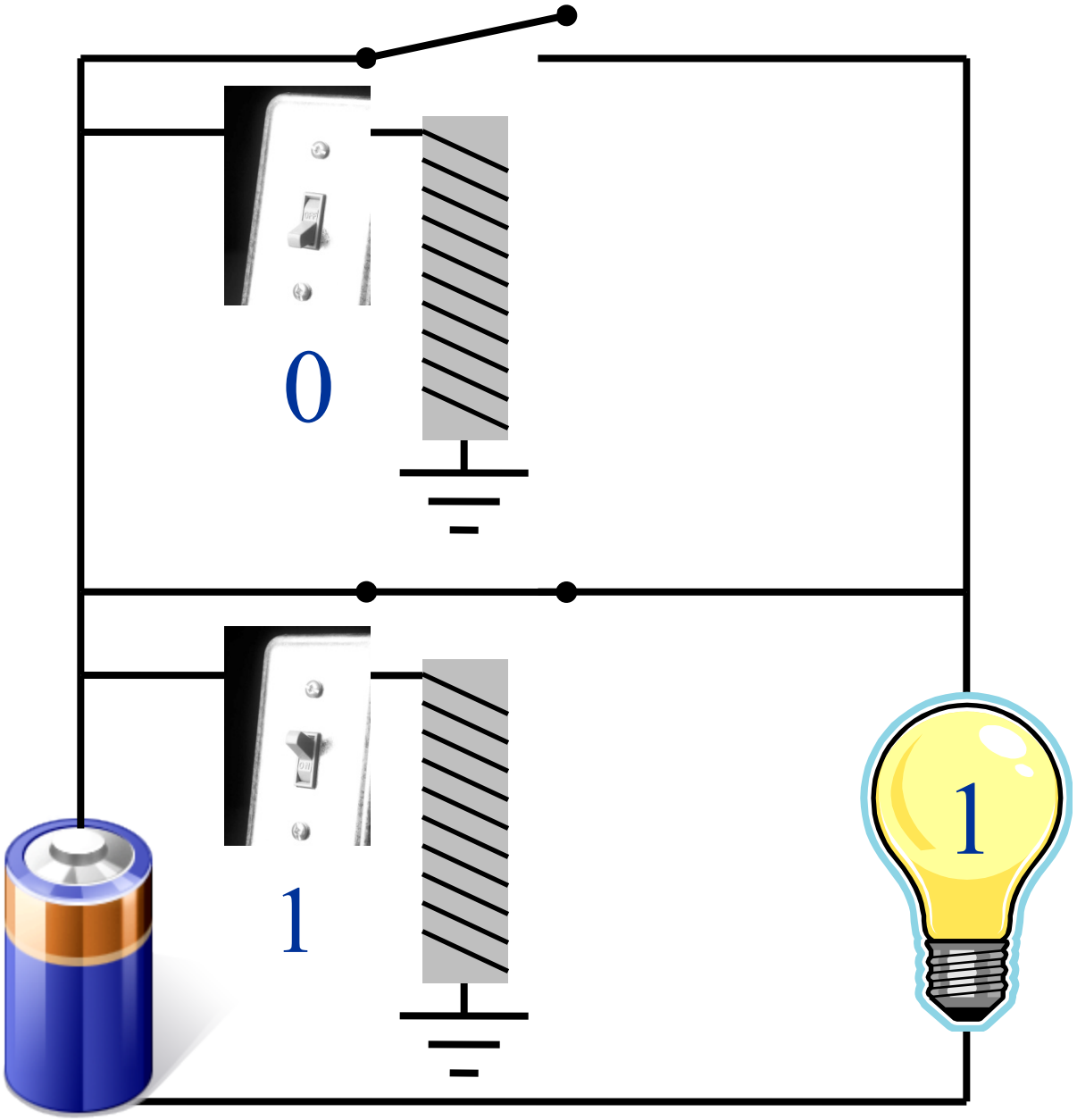
Two side by side (in parallel)?



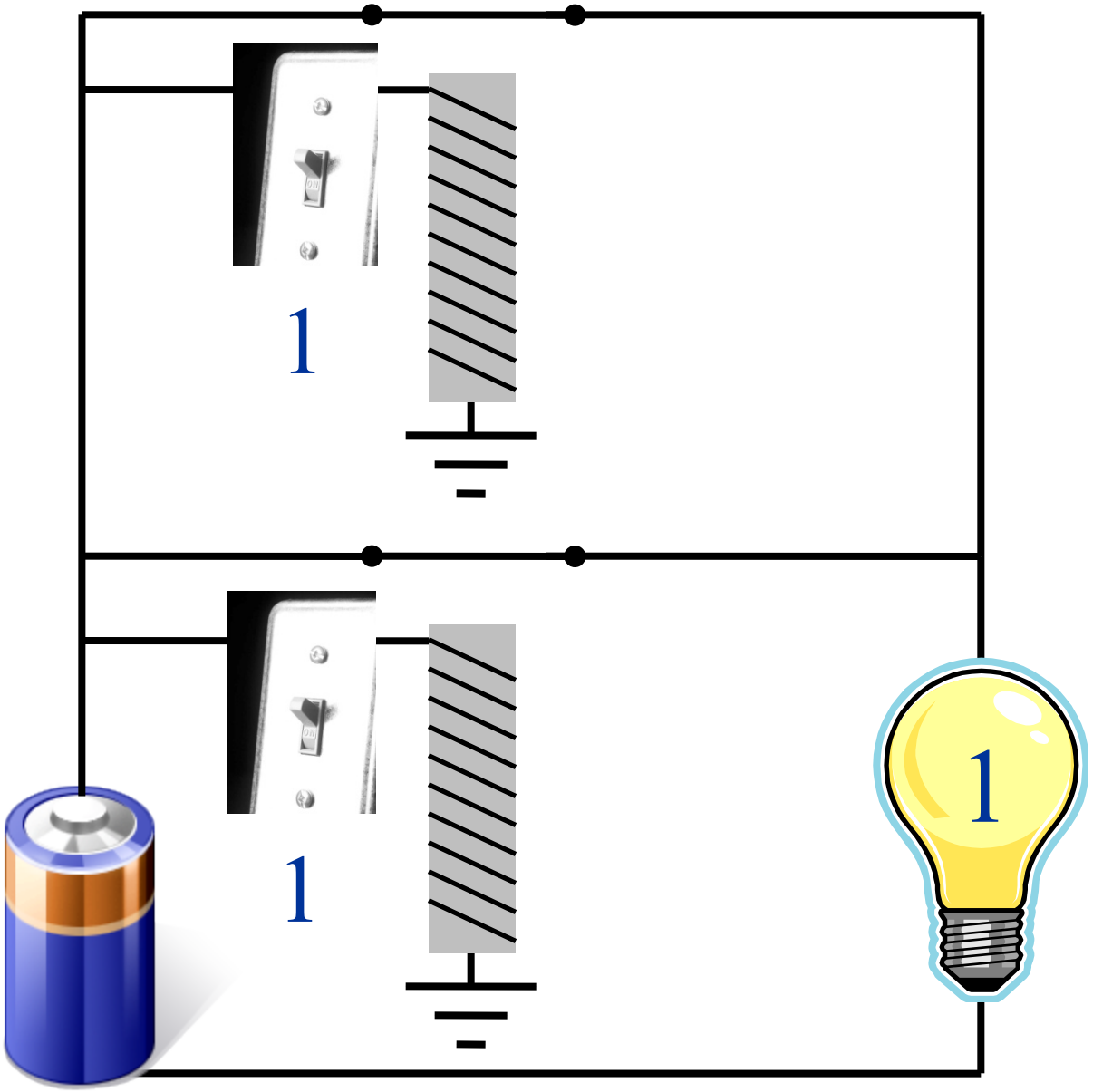
Two side by side (in parallel)?



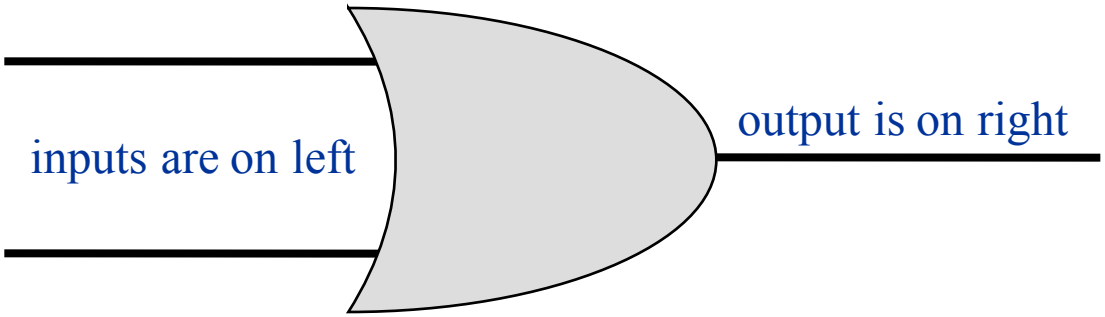
Two side by side (in parallel)?



Two side by side (in parallel)?



OR gate



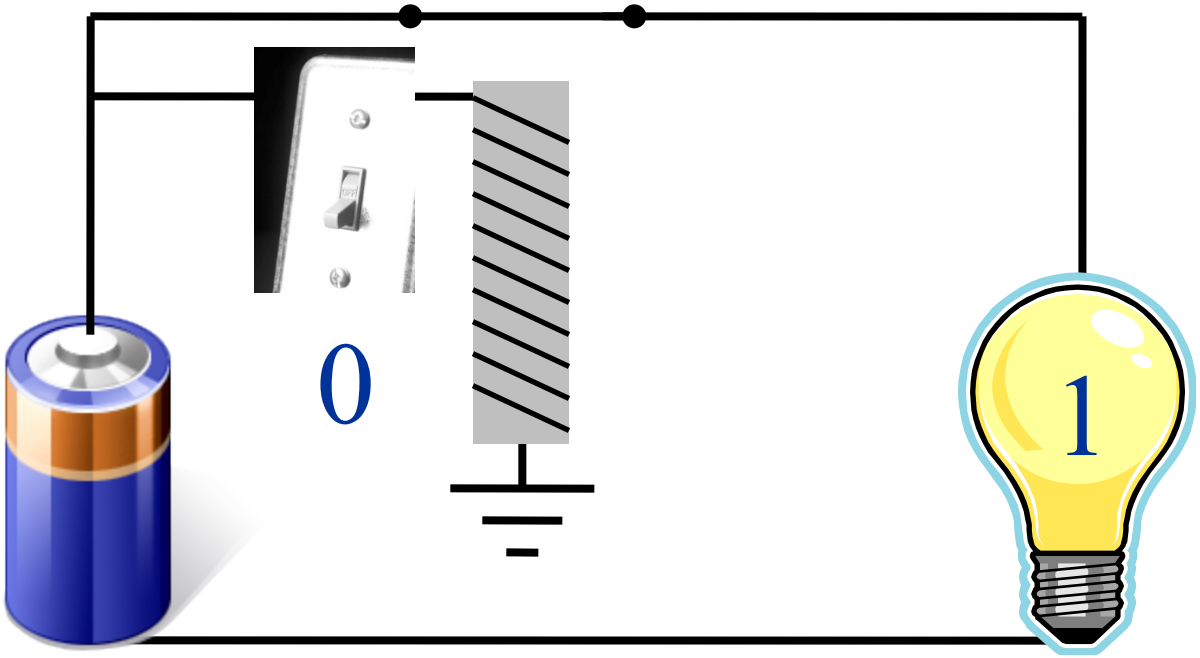
For which input values is output 1?

For which input values is output 0?

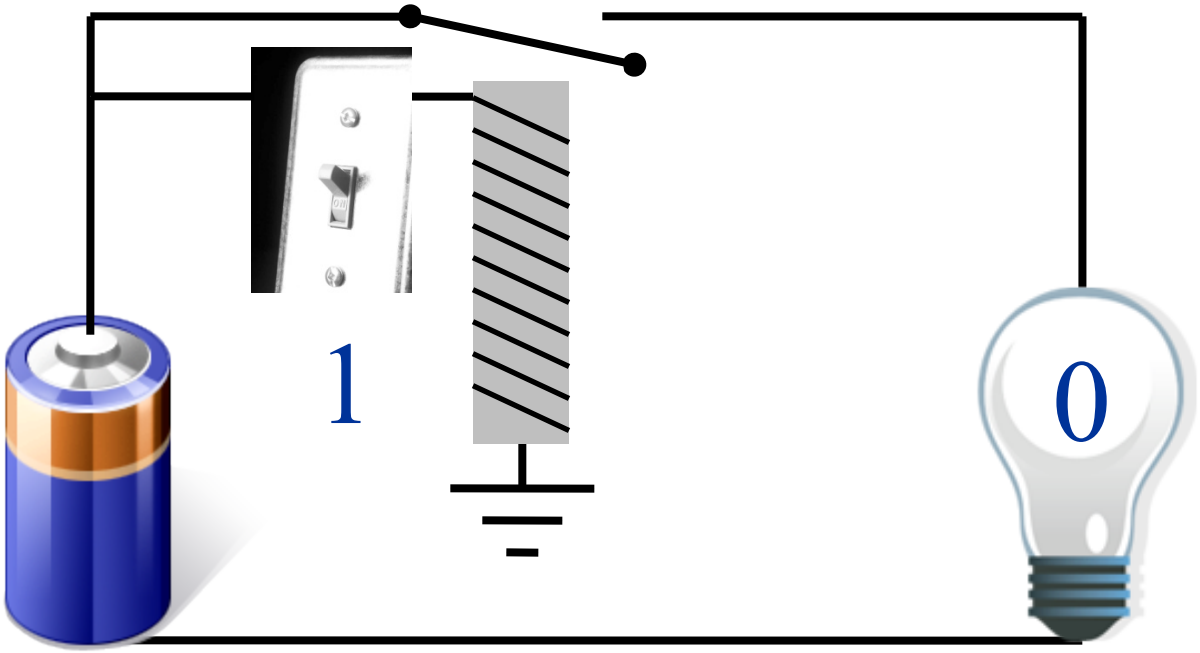
Truth table for OR

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1

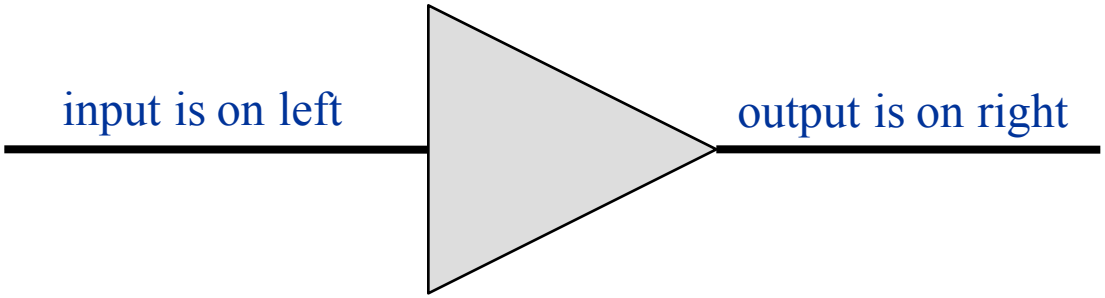
Controlling flow (normally closed)



Controlling flow (normally closed)



NOT gate



For which input value is output 1?

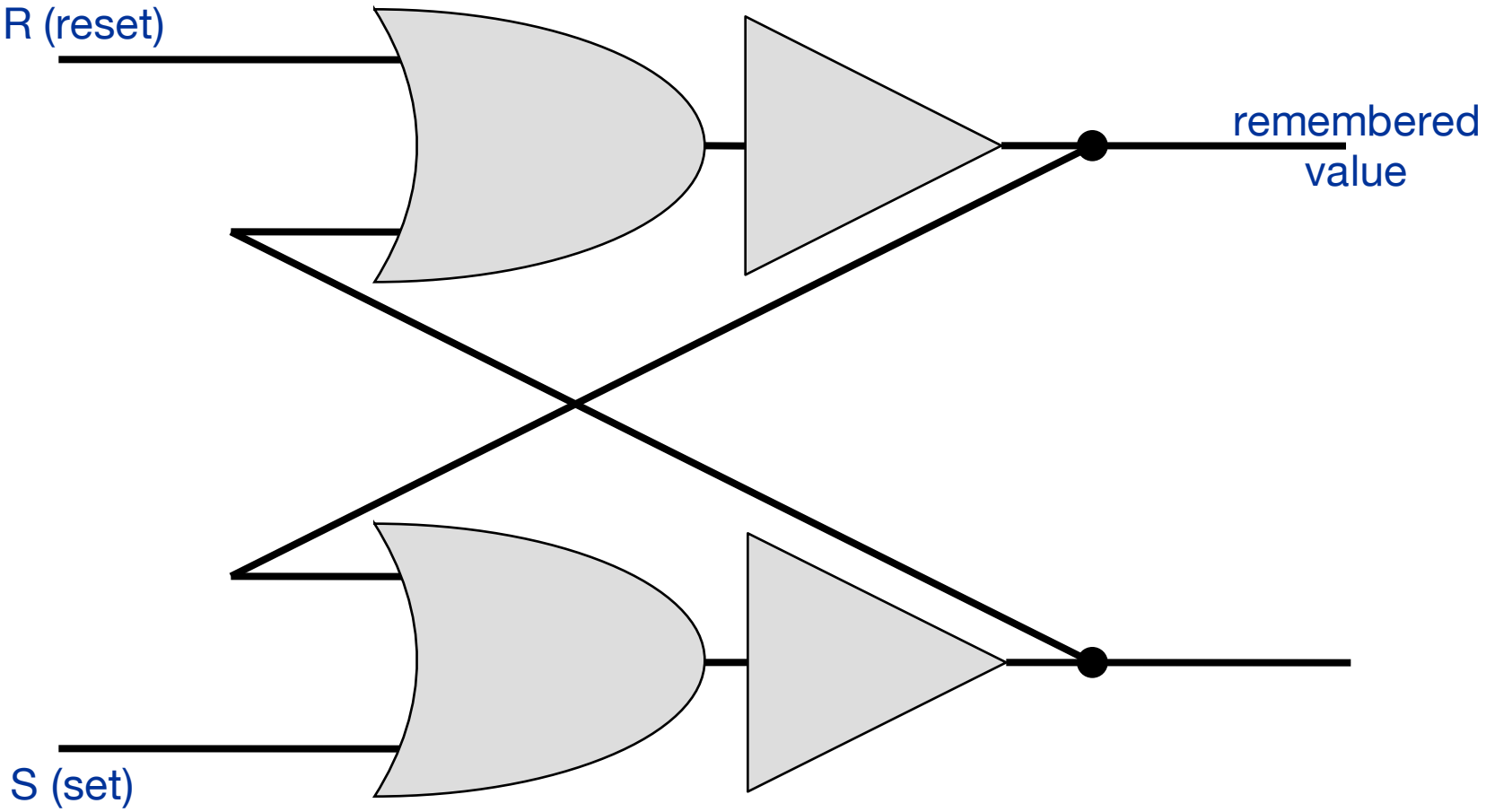
For which input value is output 0?

Truth table for NOT

Input	Output
0	1
1	0

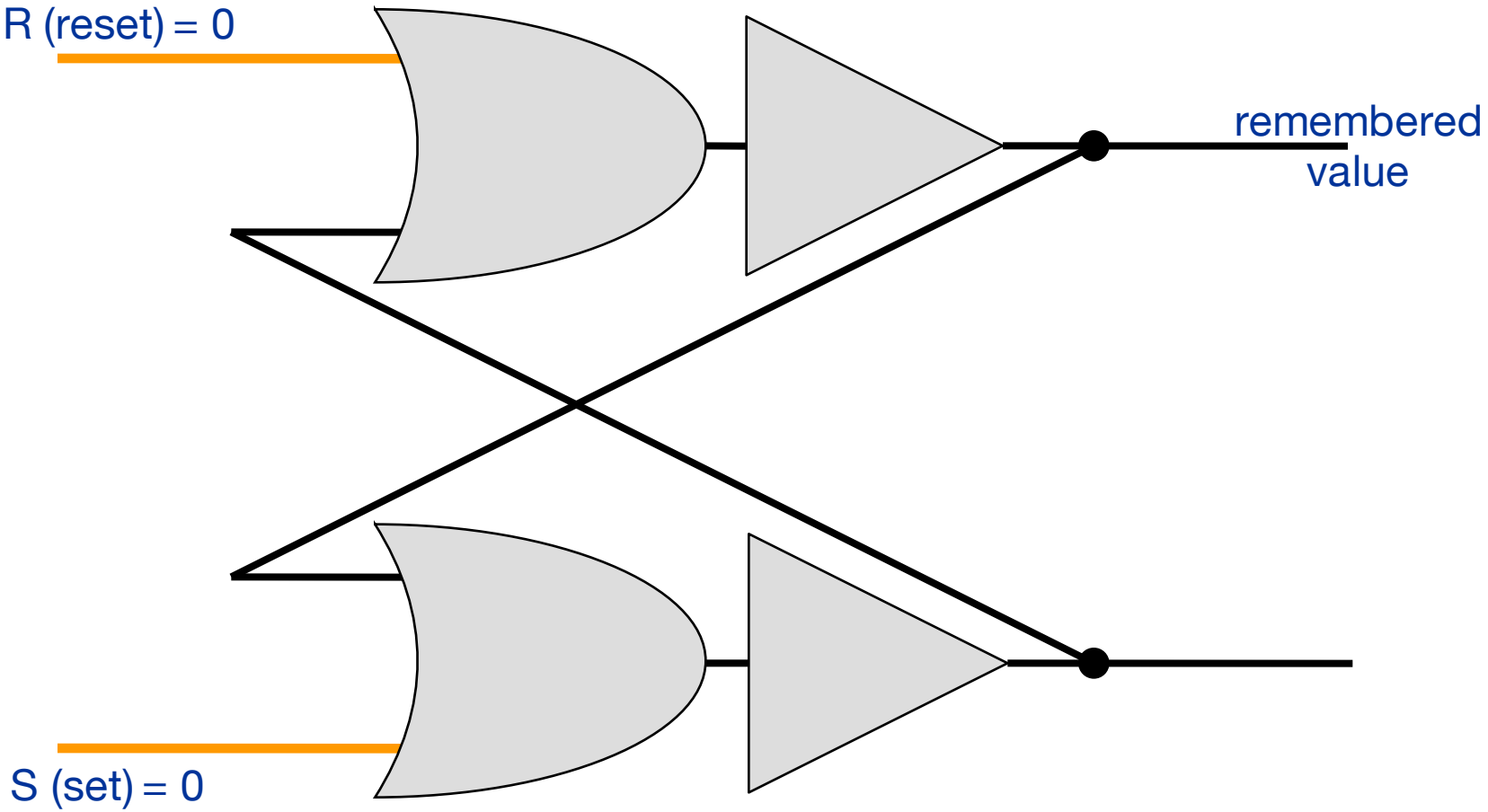
MEMORY

Flip-flop (a bit of memory!)



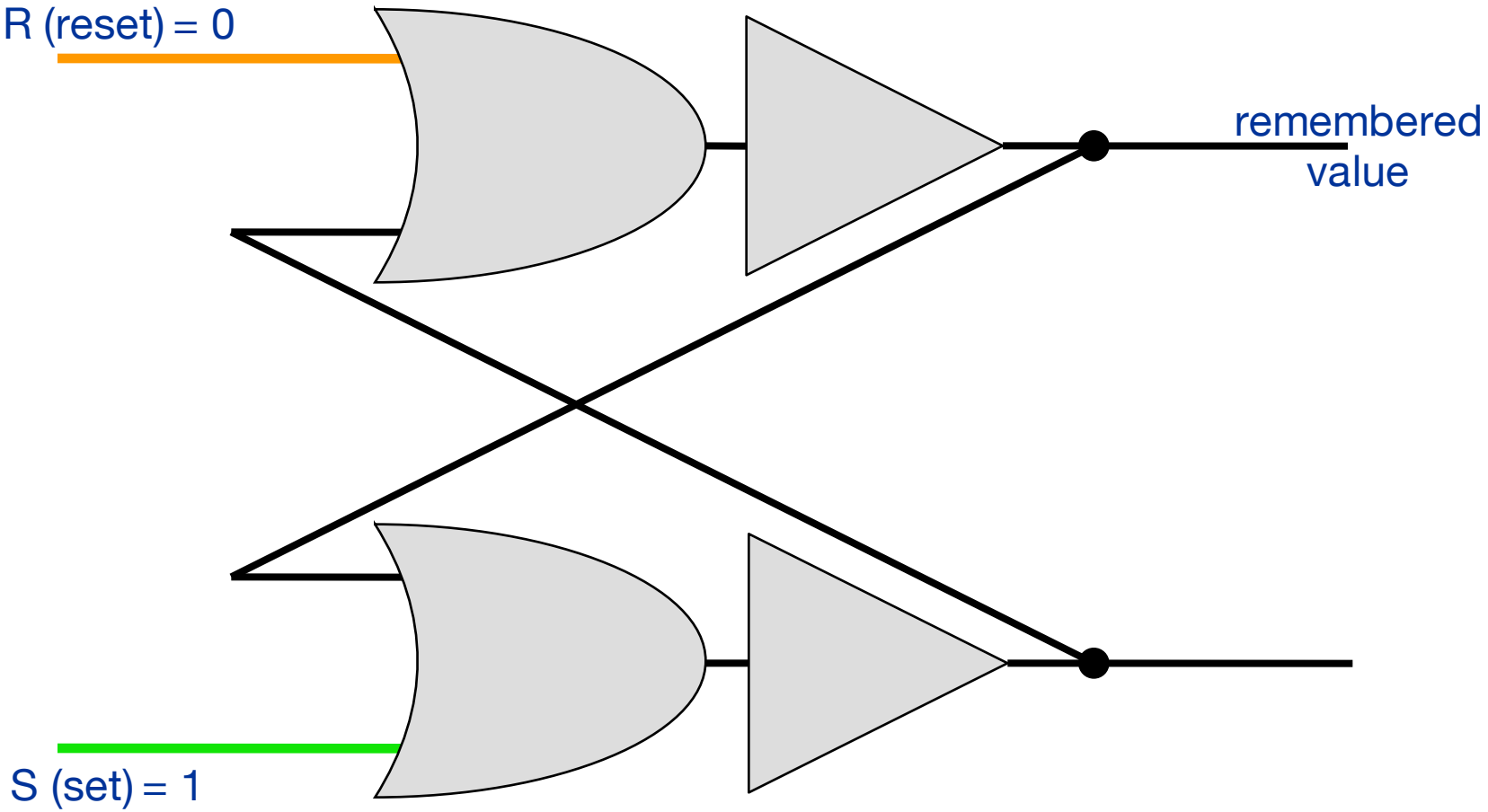
Setting the flip-flop.

The normal value of both R and S is zero



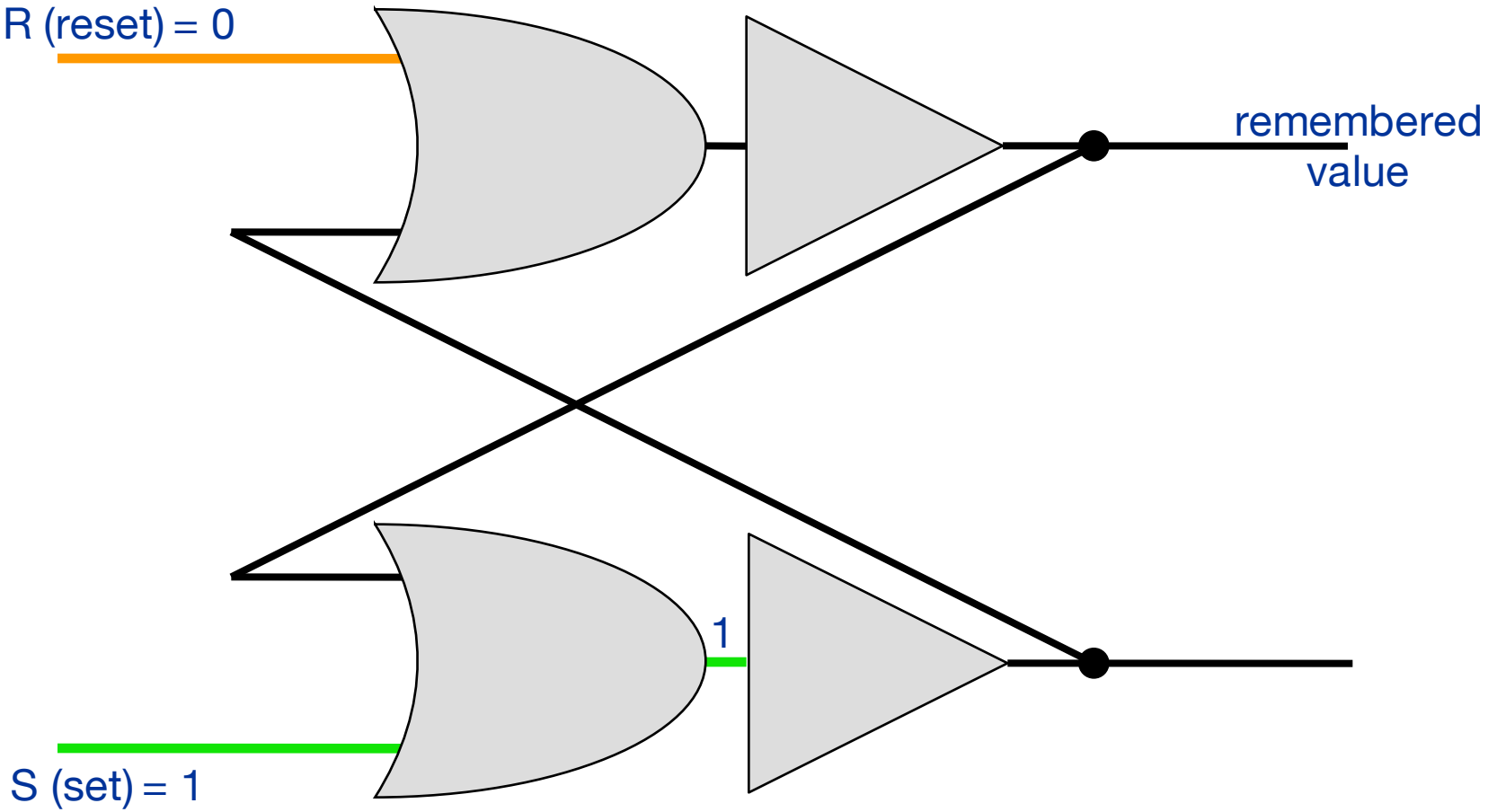
Setting the flip-flop…

To store 1 in the flip-flop, we "raise" S to 1



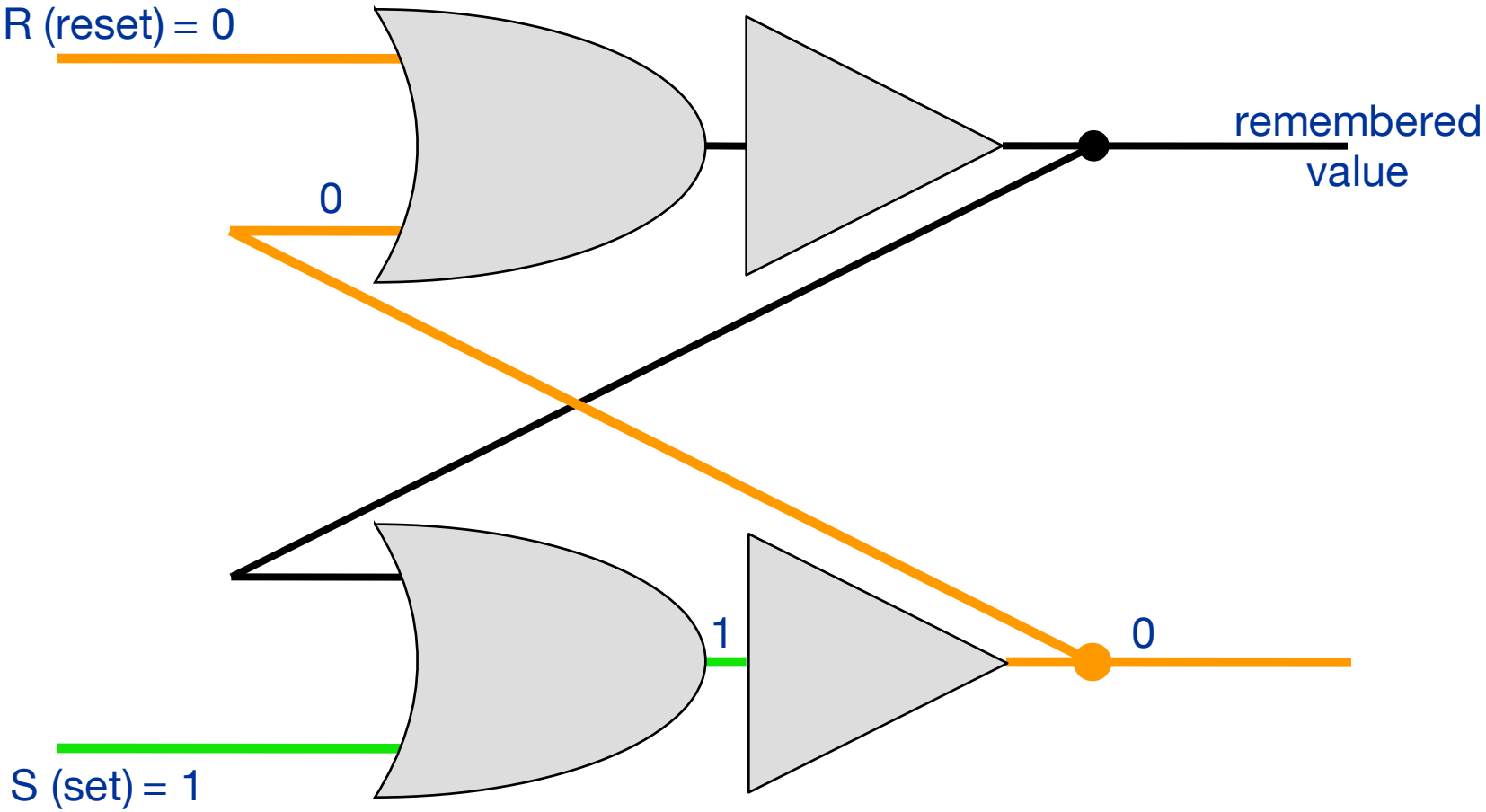
Setting the flip-flop.

which makes the output of the OR gate 1



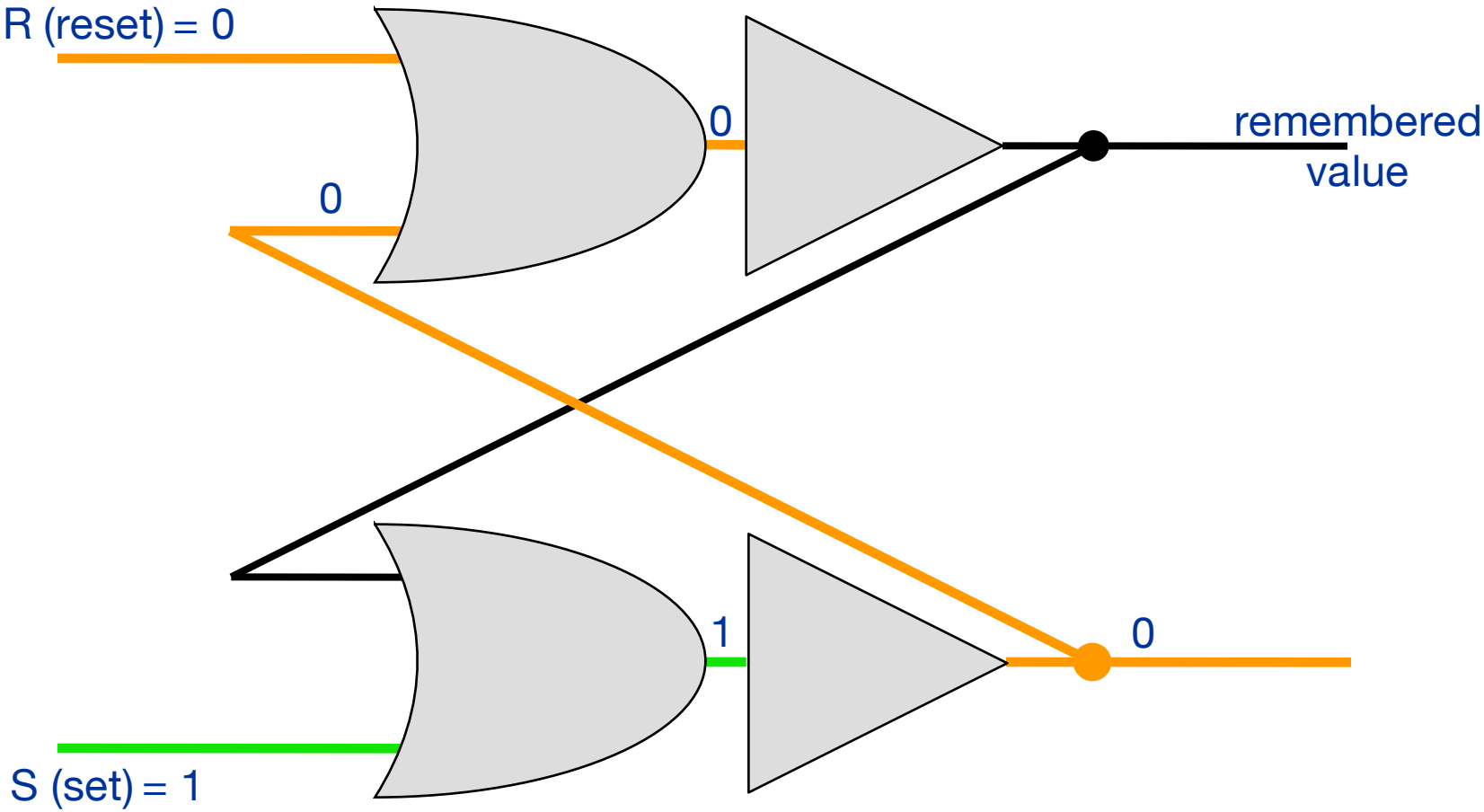
Setting the flip-flop

The NOT gate inverts this 1 value to 0, which becomes the second input to the upper OR gate.

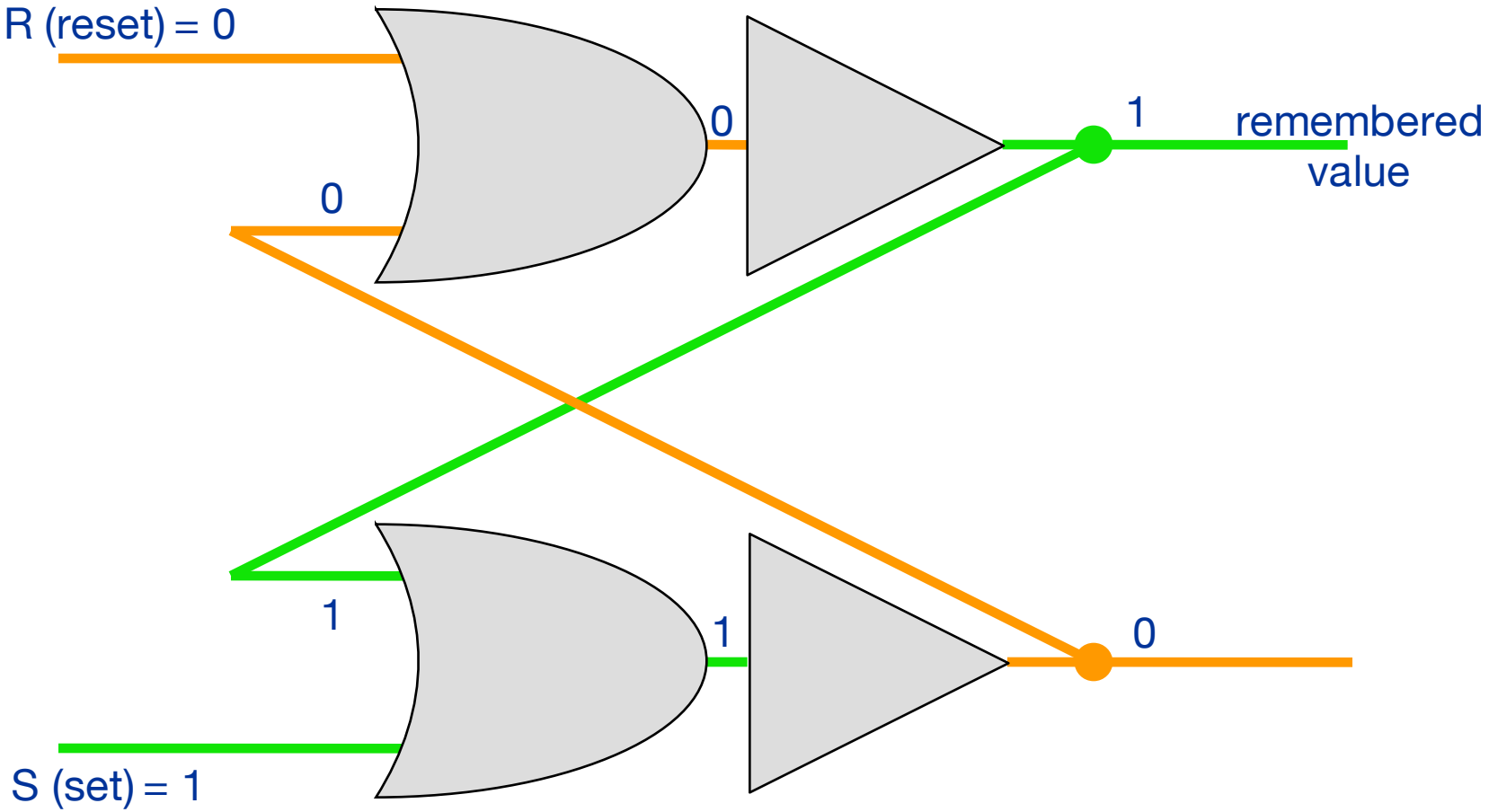


Setting the flip-flop

Since both inputs of the upper OR gate are zero, its output is zero.

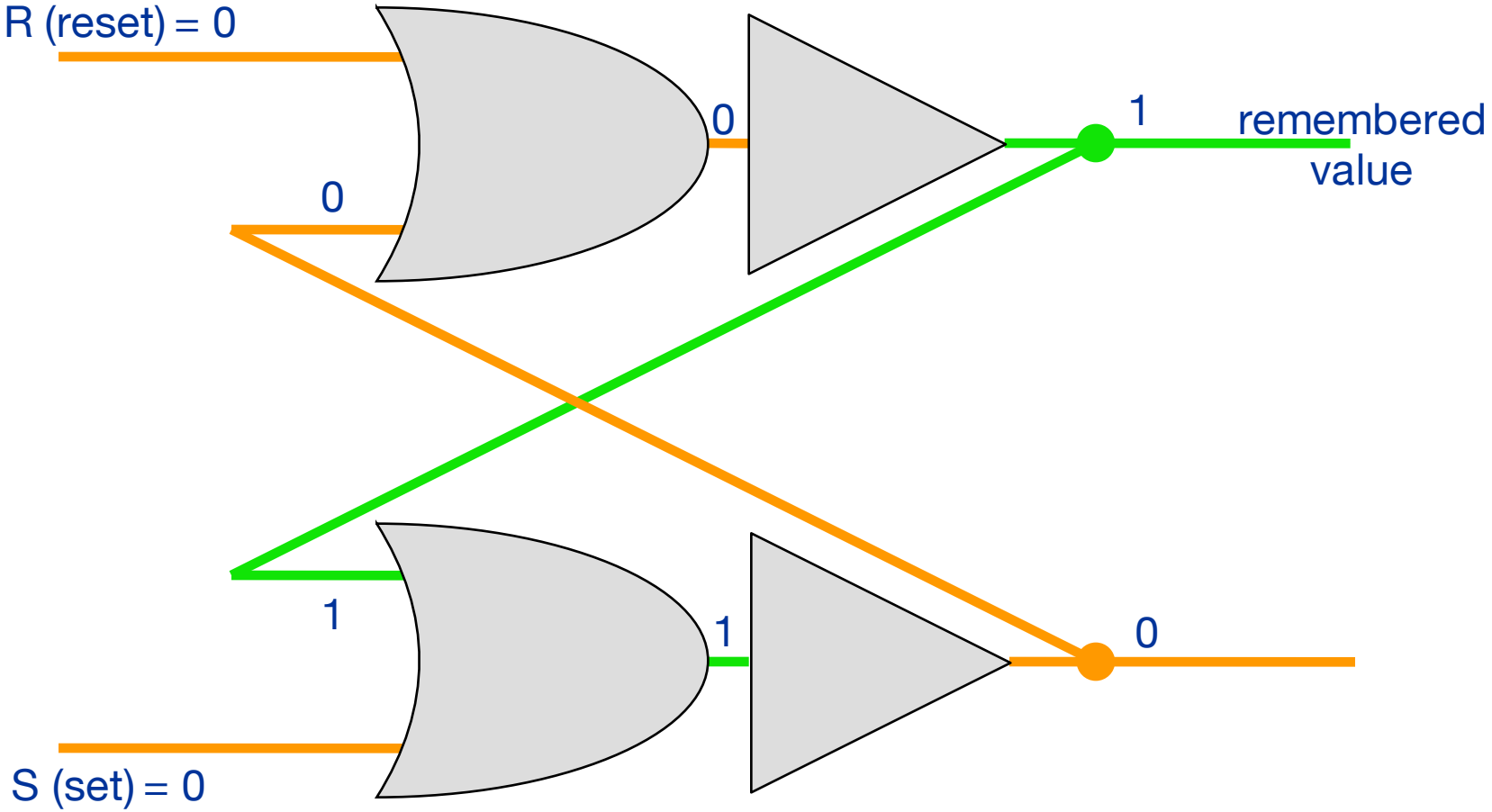


The NOT gate inverts this 0 to a 1; this value becomes the second input to the bottom OR.



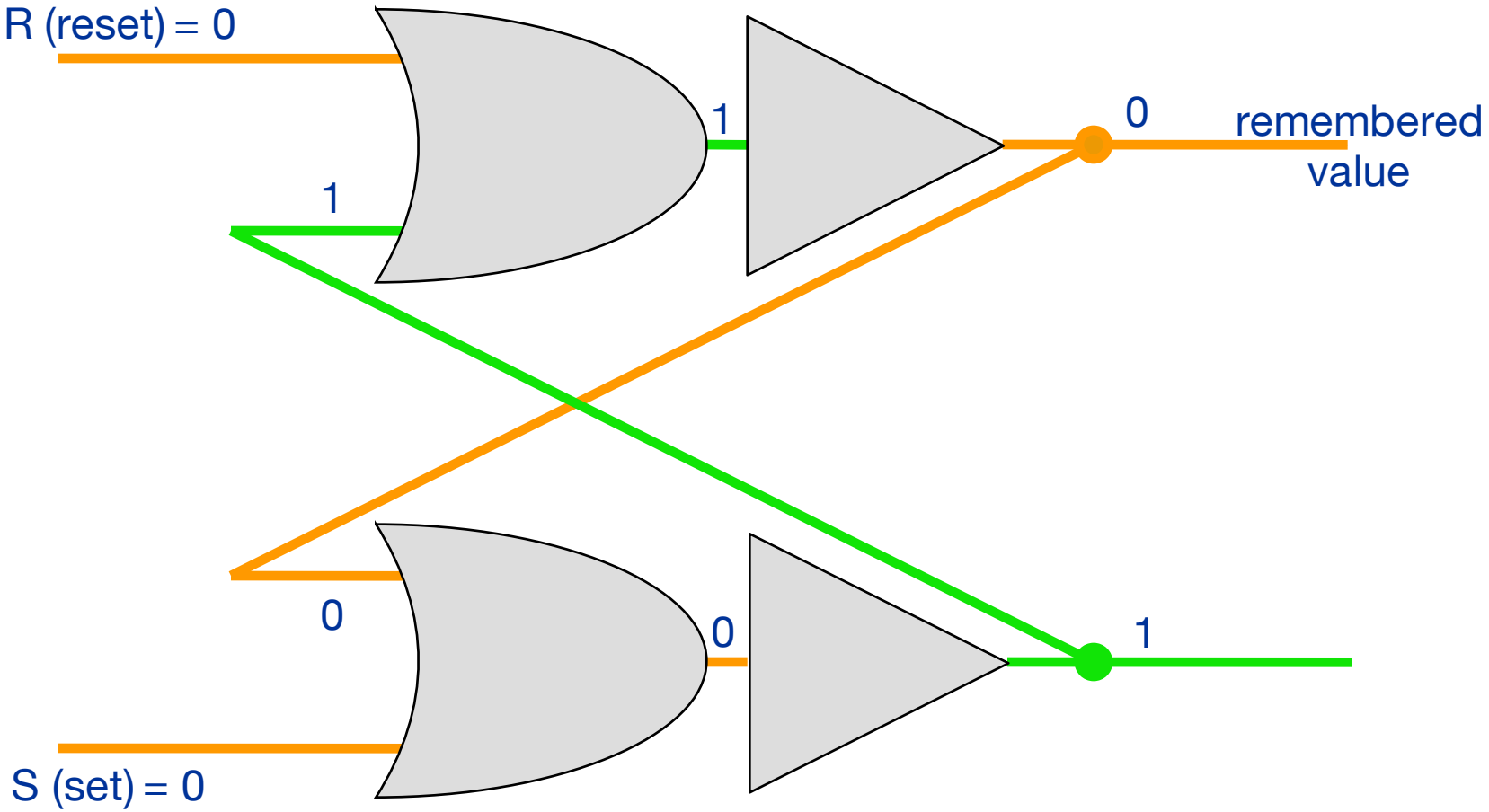
Setting the flip-flop

Because the output of the bottom OR gate will now stay at 1, we can lower S to zero, and the circuit will stay in a stable state, with 1 as the remembered value!



Resetting the flip-flop

Resetting the remembered value to zero is similar, except we raise, then lower, the value on R.



Recap

A sequence of bits (a.k.a. *bit string*) by itself does not carry meaning.

A bit string can be interpreted under a given representation scheme, which allows us to recover the encoded meaning.

Circuits made from simple gates let us store and manipulate bit strings.

1 flip-flop stores 1 bit

1 byte = 8 bits

Primary storage (2^x)

1 KB = 2^{10} bytes
(1,024 bytes)

1 MB = 2^{20} bytes
(1,048,576 bytes)

1 GB = 2^{30} bytes
(1,073,741,824 bytes)

Secondary storage (10^x)

1 kB = 10^3 bytes
(1,000 bytes)

1 MB = 10^6 bytes
(1,000,000 bytes)

1 GB = 10^9 bytes
(1,000,000,000 bytes)

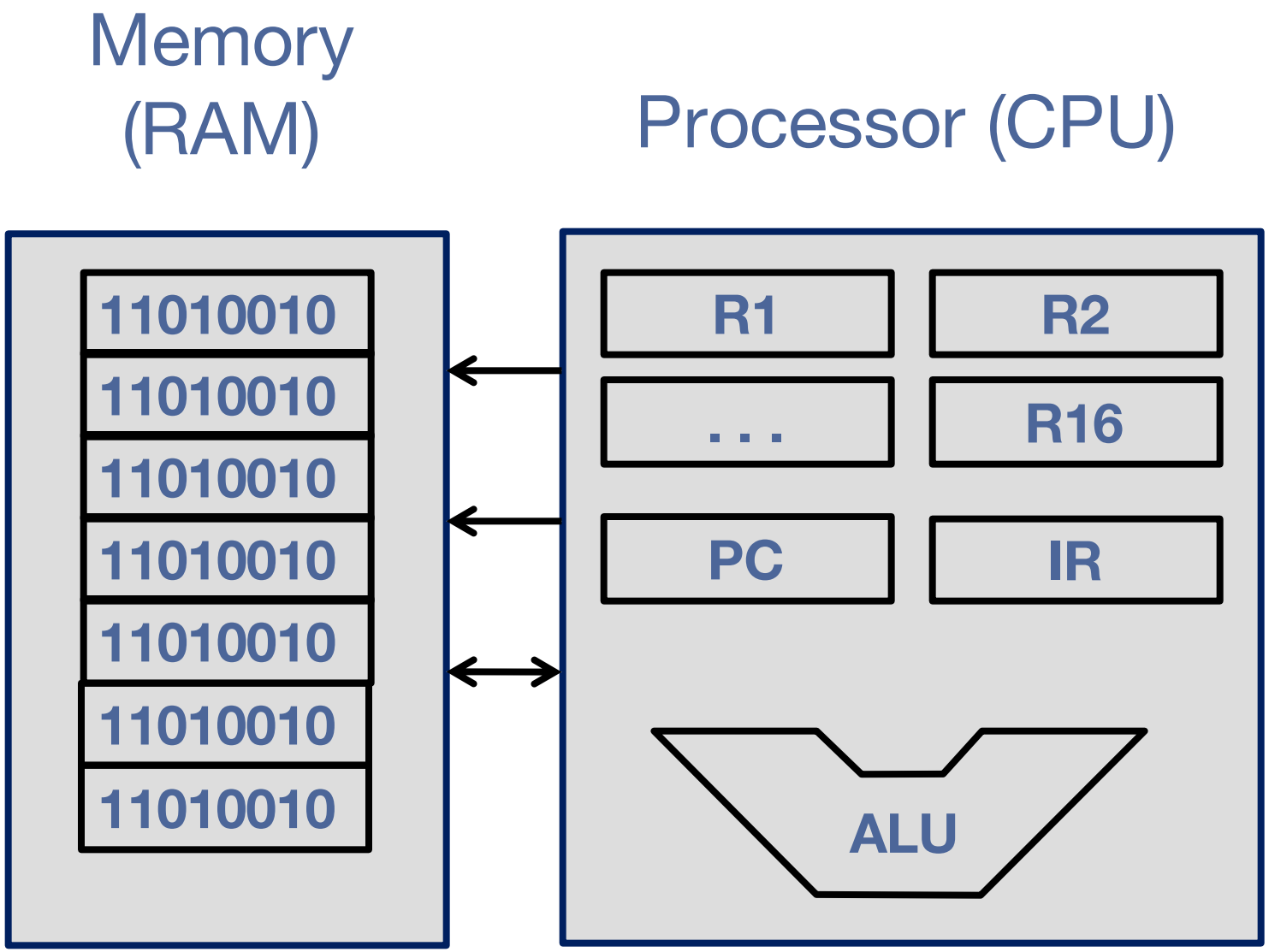
For more info, see https://en.wikipedia.org/wiki/Binary_prefix

INSTRUCTION DECODING

Memory
Central Processing Unit (CPU)

- Memory
- Central Processing Unit (CPU)
  - Arithmetic Logic Unit (ALU)
  - Registers

- Memory
- Central Processing Unit (CPU)
  - Arithmetic Logic Unit (ALU)
  - Registers
    - General purpose (e.g. R1 – R16)
    - Special purpose (e.g. Program Counter and Instruction Register)


Memory
(RAM)
Processor (CPU)
11010010
11010010
11010010
11010010
11010010
11010010
11010010
R1
R2
. . .
R16
PC
IR
ALU