CSD1100

Logic Gates

Vadim Surov

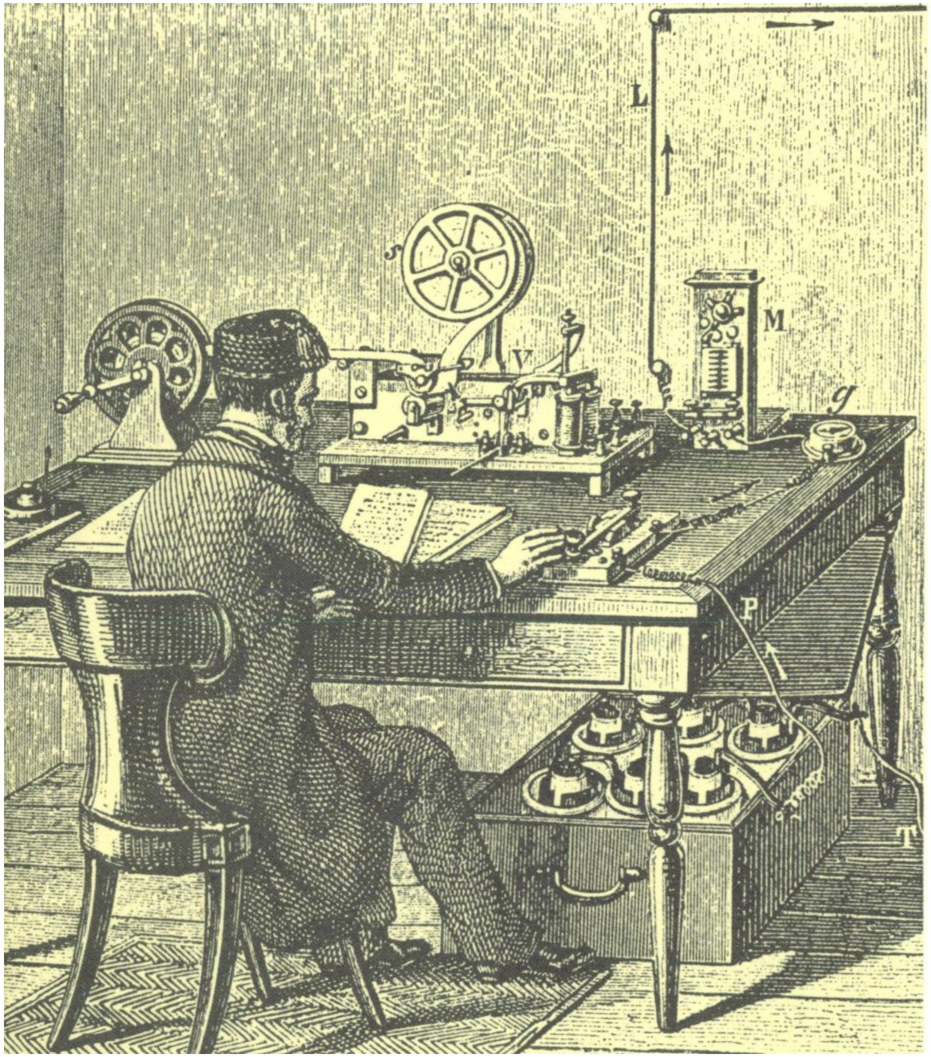
Introduction

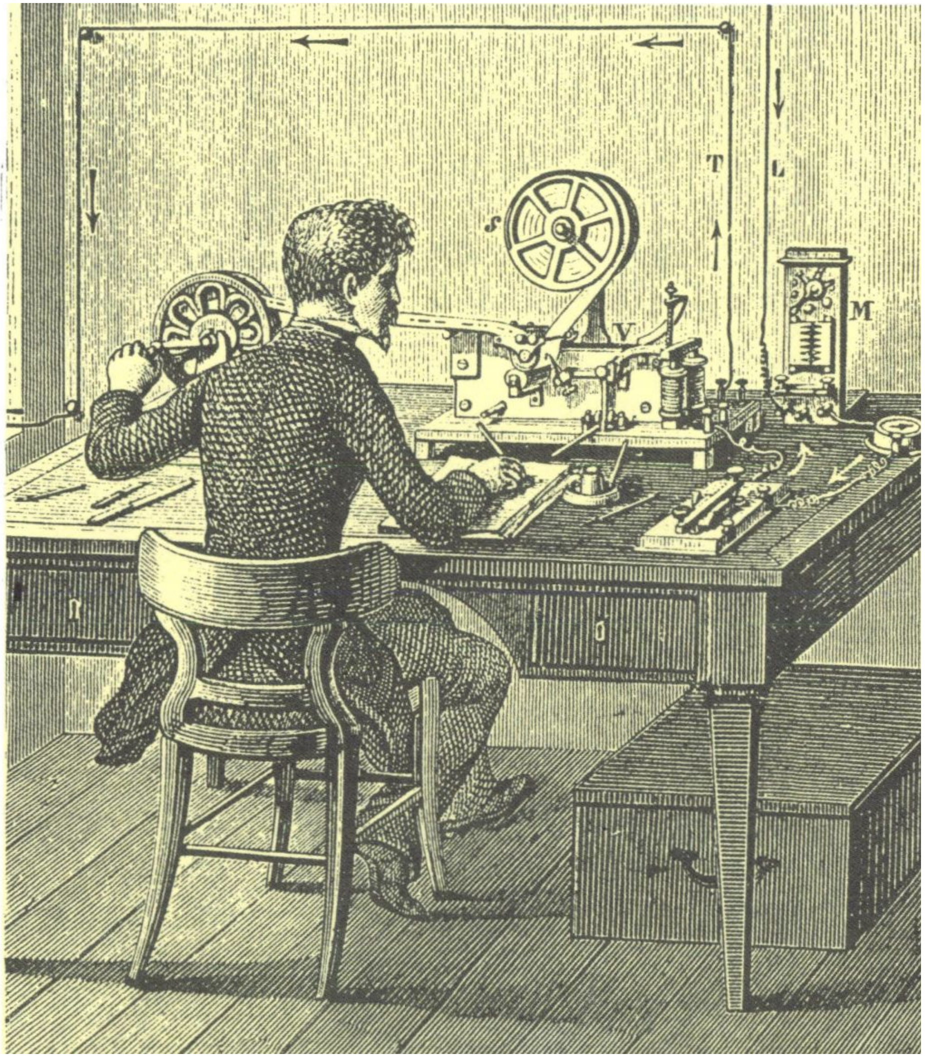
- We have looked at Boolean expressions in abstract terms.
- In this presentation, we see that Boolean functions are implemented in digital computer circuits called gates.
- A gate is an electronic device that produces a result based on two or more input values.
  - In reality, gates consist of one to six transistors, but digital designers think of them as a single unit.
  - Integrated circuits contain collections of gates suited to a particular purpose.

Telegraph

- In 1836, Samuel F. Morse (born 1791 in Massachusetts) invented the telegraph (which means “far writing”).
- The idea is fairly simple: you do something at one end of a wire (apply a current) that causes something to happen (flashing a light) at the other end of the wire.
- At the core of a telegraph is a **relay**; its function relies on electromagnetism.

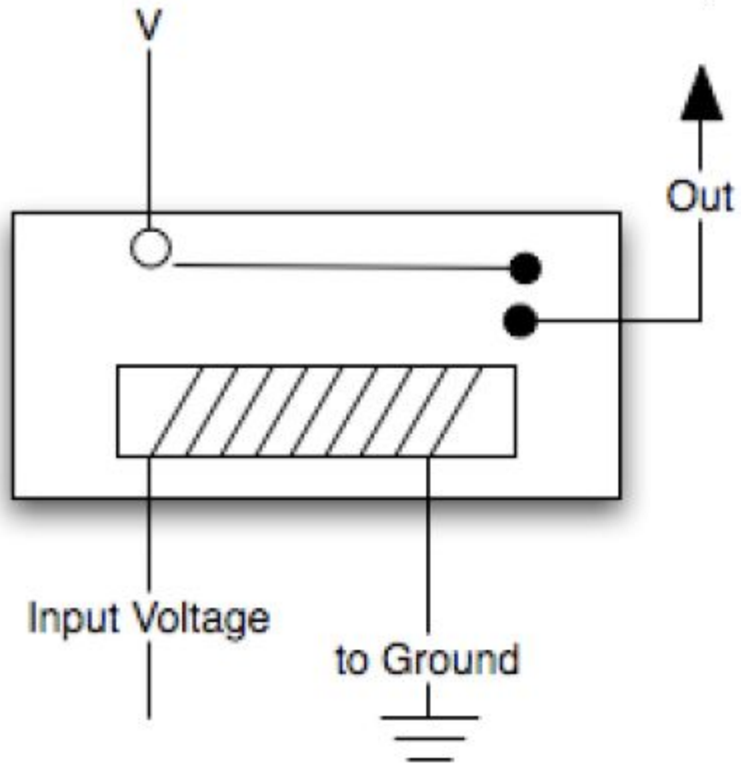
Telegraph





Relay

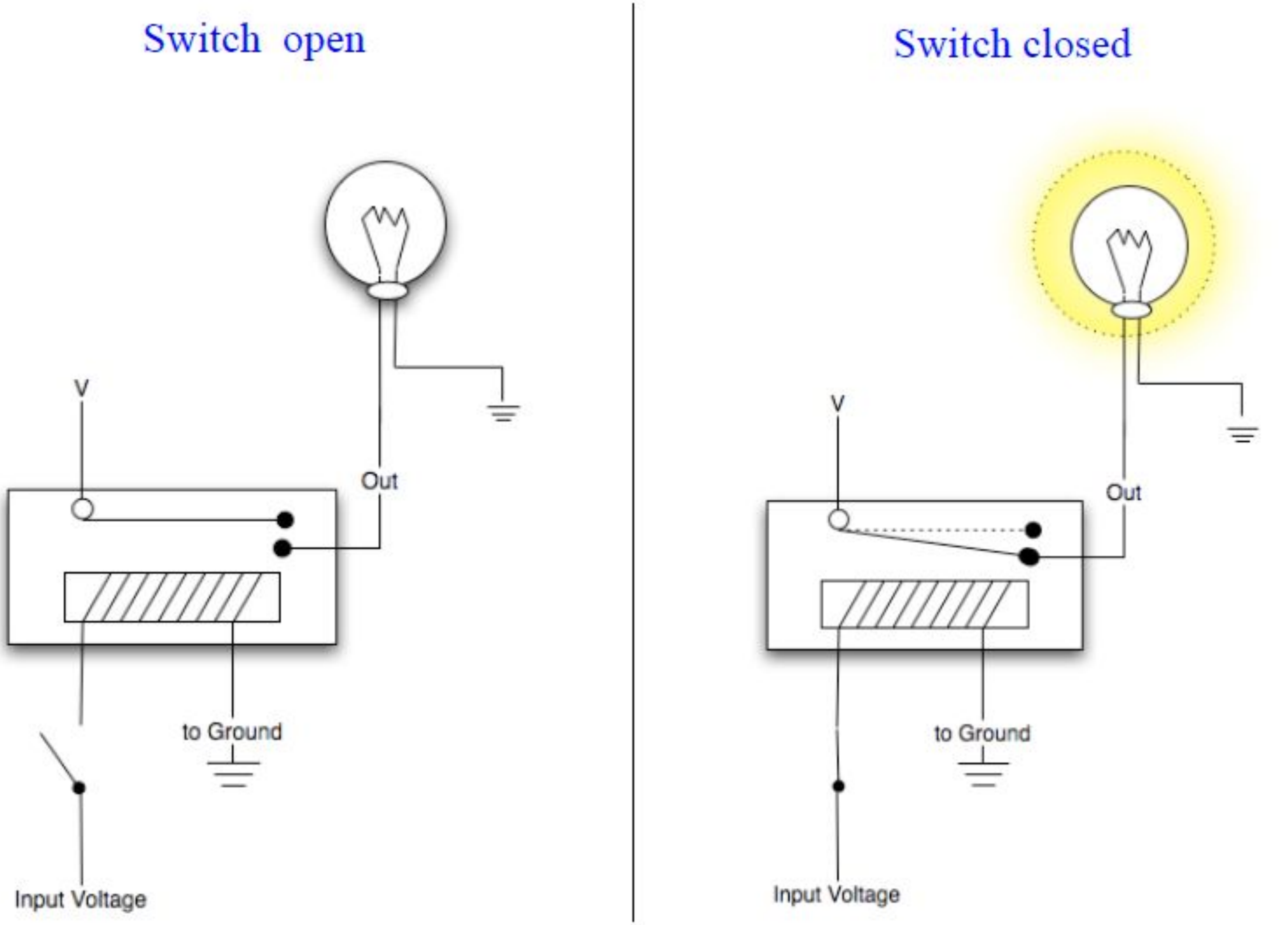
- A relay is an electrically controlled switch.
- Current flowing through the coil of the relay creates a magnetic field which attracts a lever and changes the switch contacts.
- The coil current can be on or off, so a relay has two switch positions.



Relay

- Relays allow one circuit to switch a second circuit, which can be completely and remotely separate from the first.
- For example a low voltage battery circuit can use a relay to switch an 110V circuit.
- There is no electrical connection inside the relay between the two circuits; the link is both magnetic and mechanical.

Relay



Switch open

Switch closed

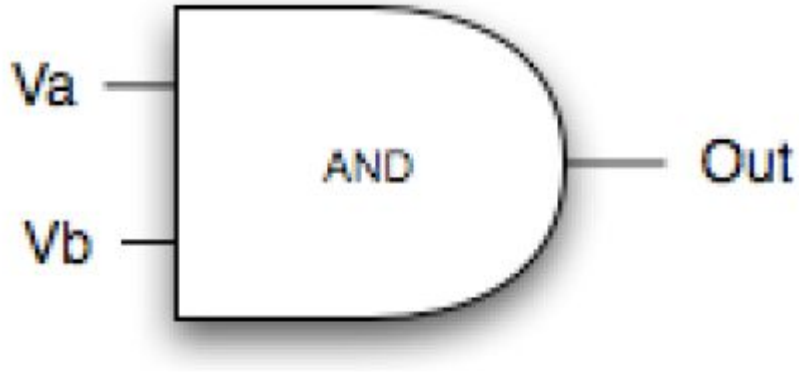
Building logic gates

- Using relays the following simple “machines” (called gates) could be manufactured:
  - AND gate
  - OR gate
  - NOT gate or Inverter
- More complex gates can be manufactured by combining simple gates. The following gates could be manufactured:
  - NAND gate
  - NOR gate
  - XOR gate
  - XNOR gate

AND gate

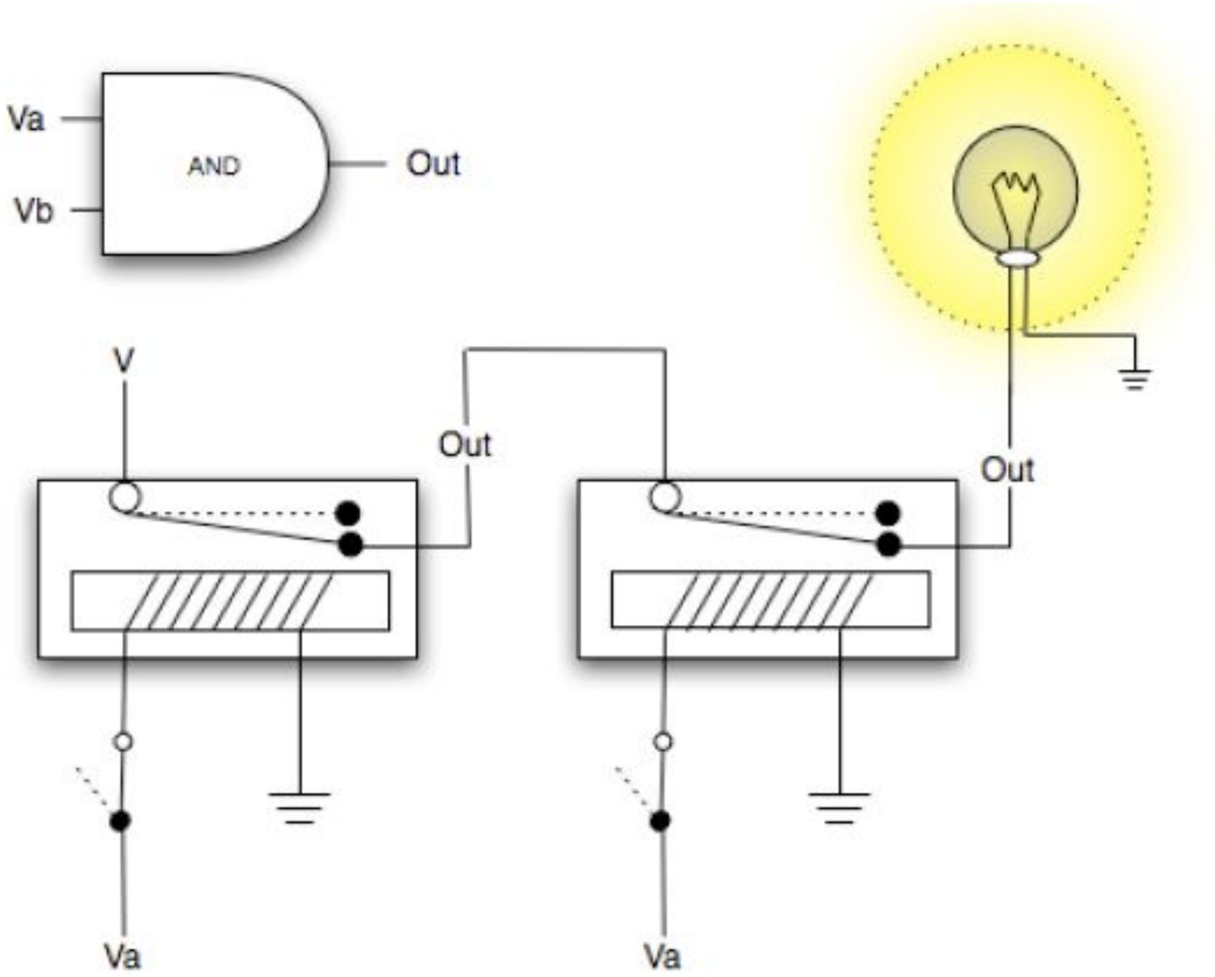
- The basic logical AND gate has has 2 inputs and 1 output.
    - Whenever at least one of the inputs is low, the output voltage will be low.
    - If both inputs are high, then the output voltage will be high.
- Some AND gates have 3 or more inputs.

AND gate

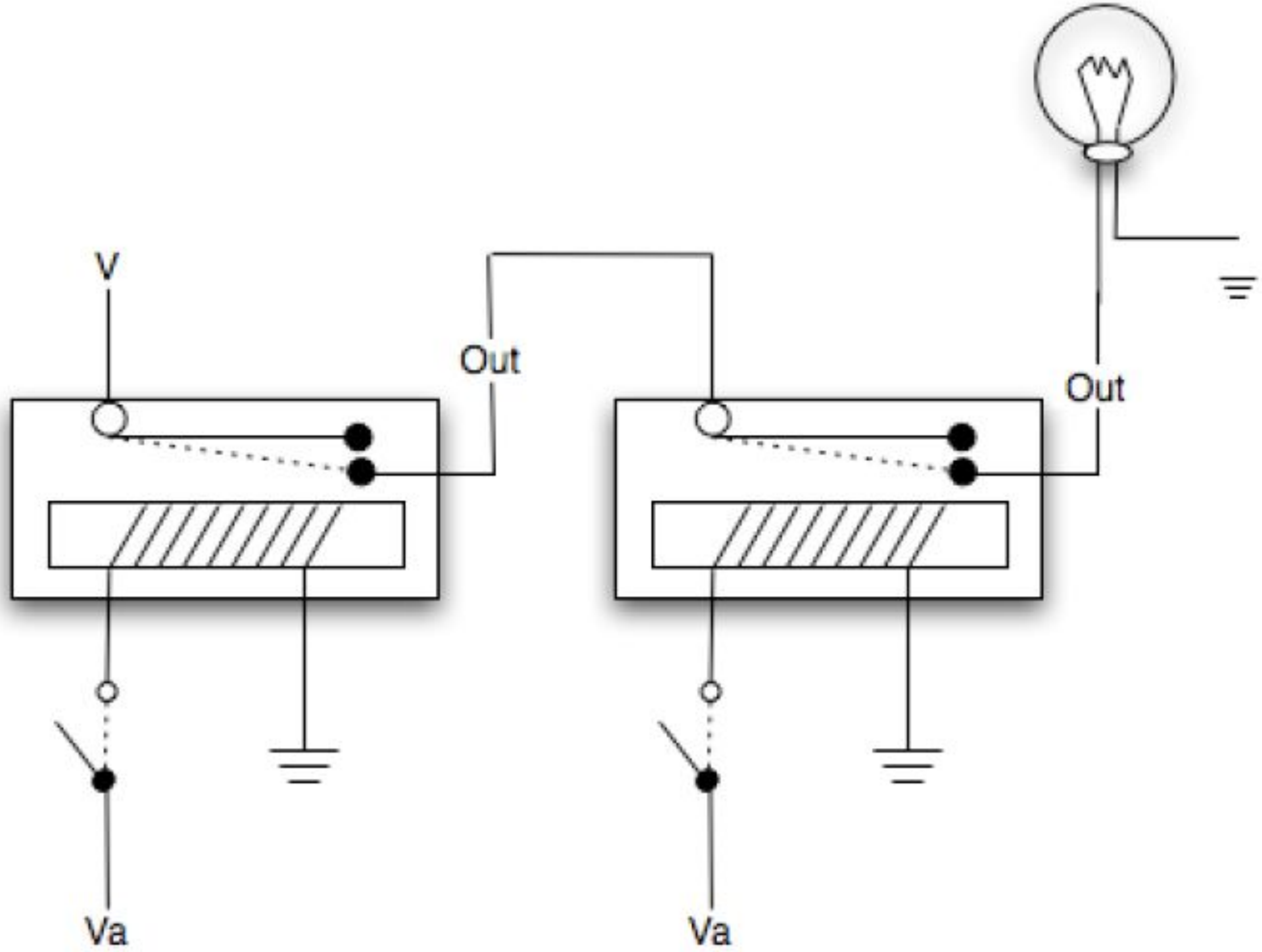


A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

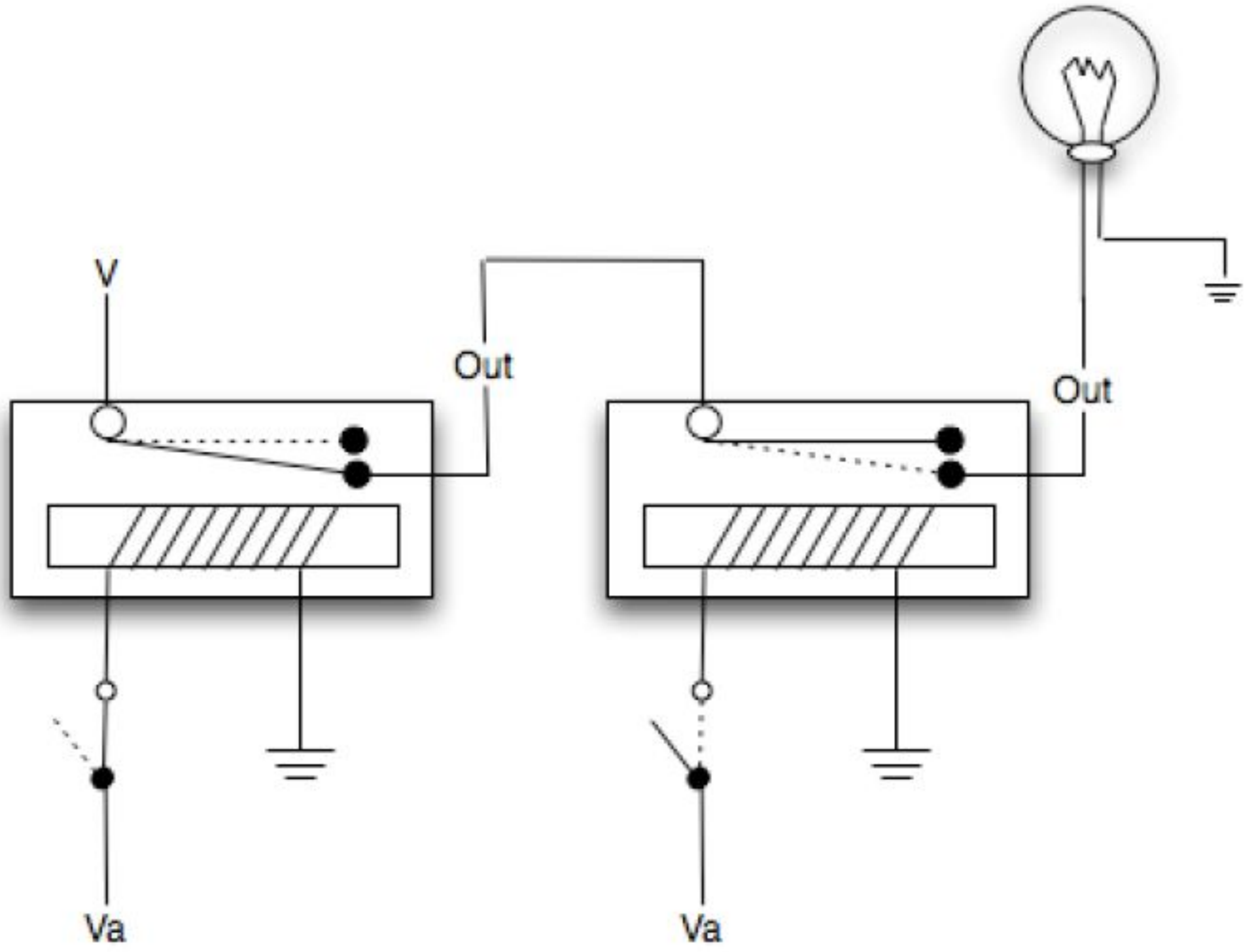
AND gate



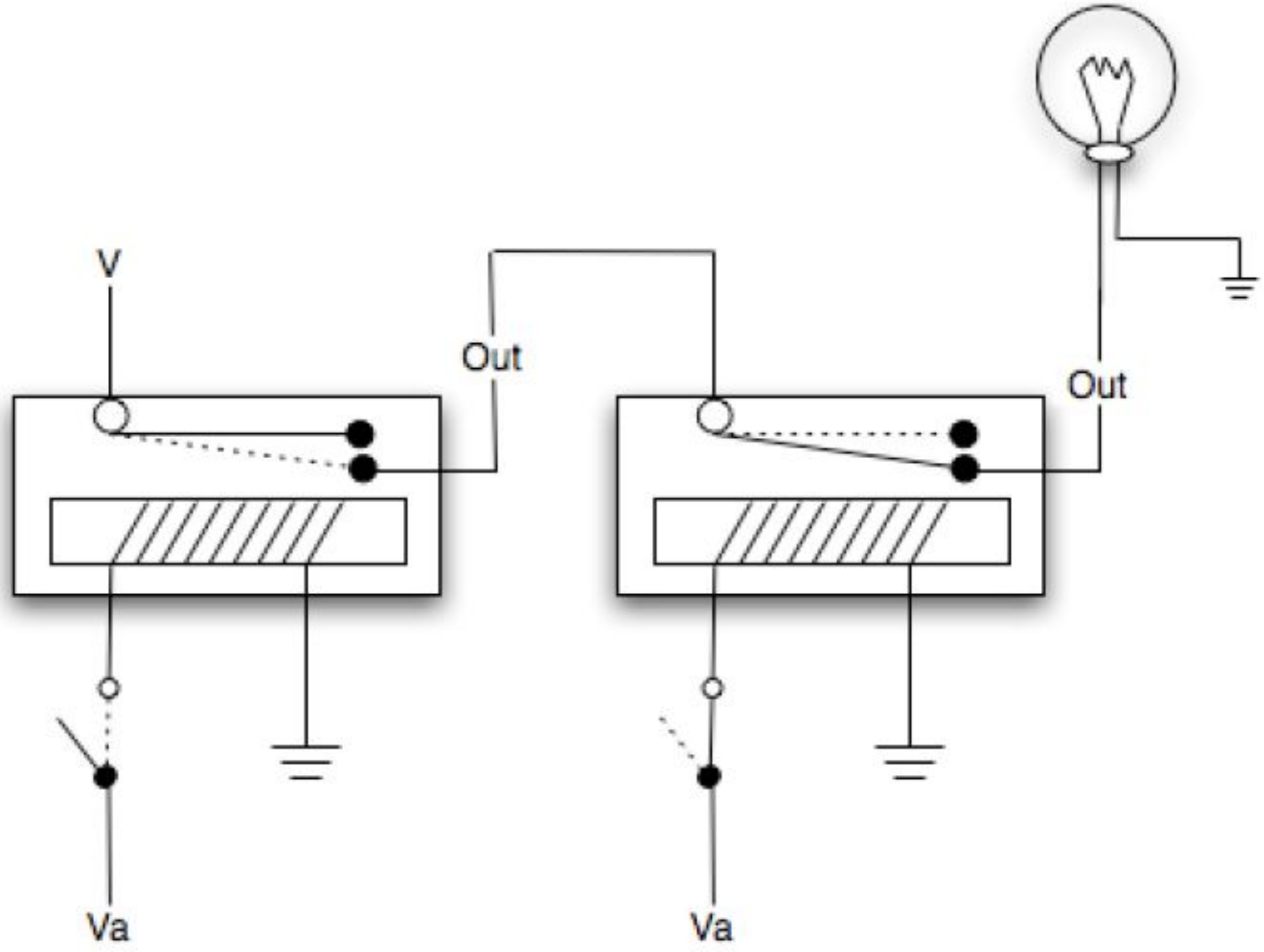
AND gate



AND gate



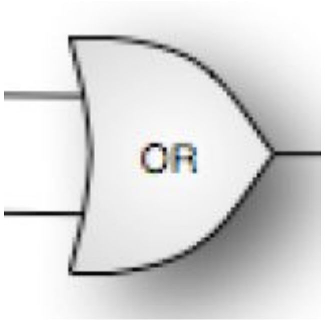
AND gate



OR gate

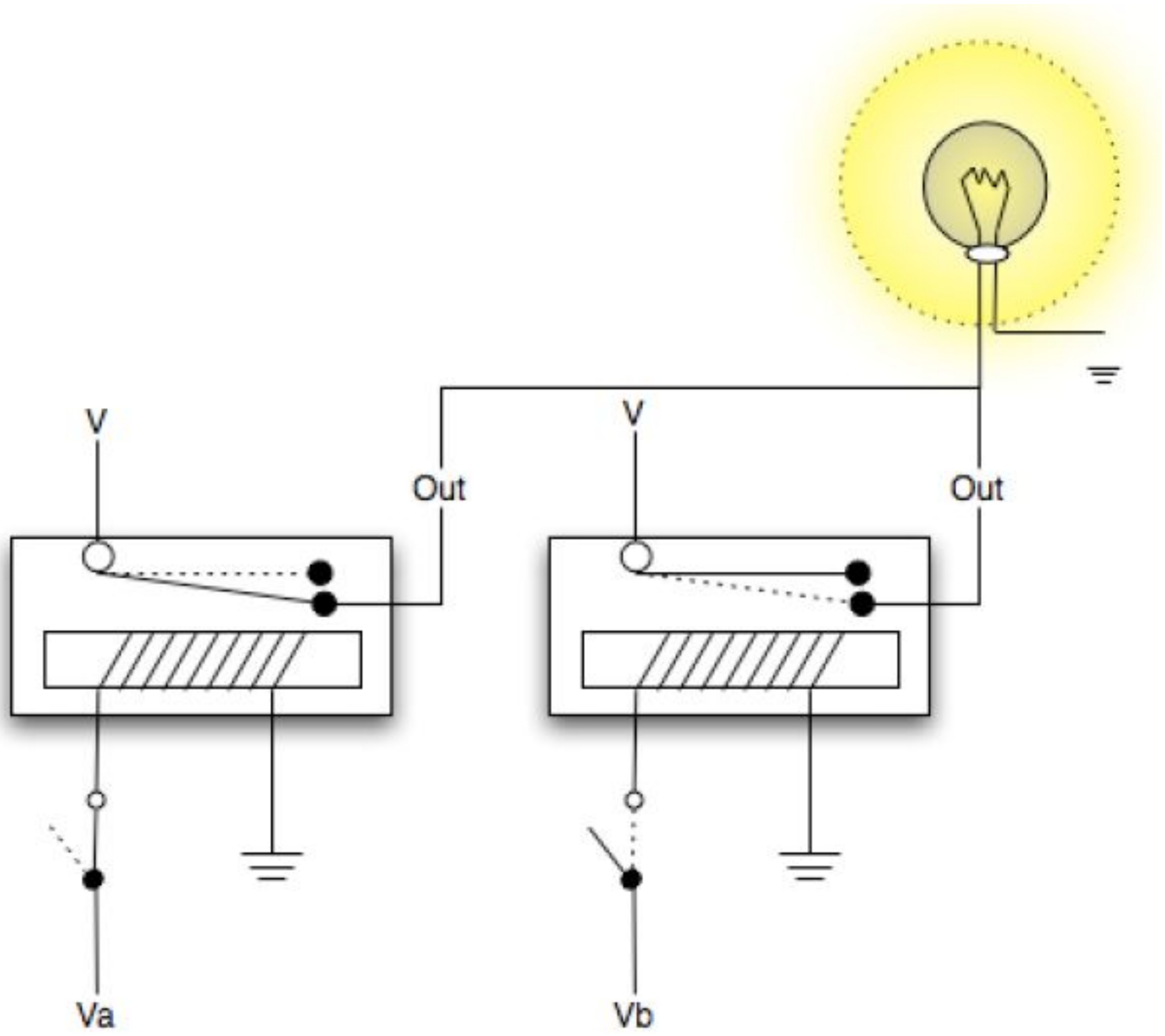
- The basic logical OR gate has 2 inputs and 1 output.
    - Whenever at least one of the inputs is high, the output voltage will be high.
    - If both inputs are low, then the output voltage will be low.
- Some OR gates have 3 or more inputs.
- Note that this is an inclusive OR function.

OR gate

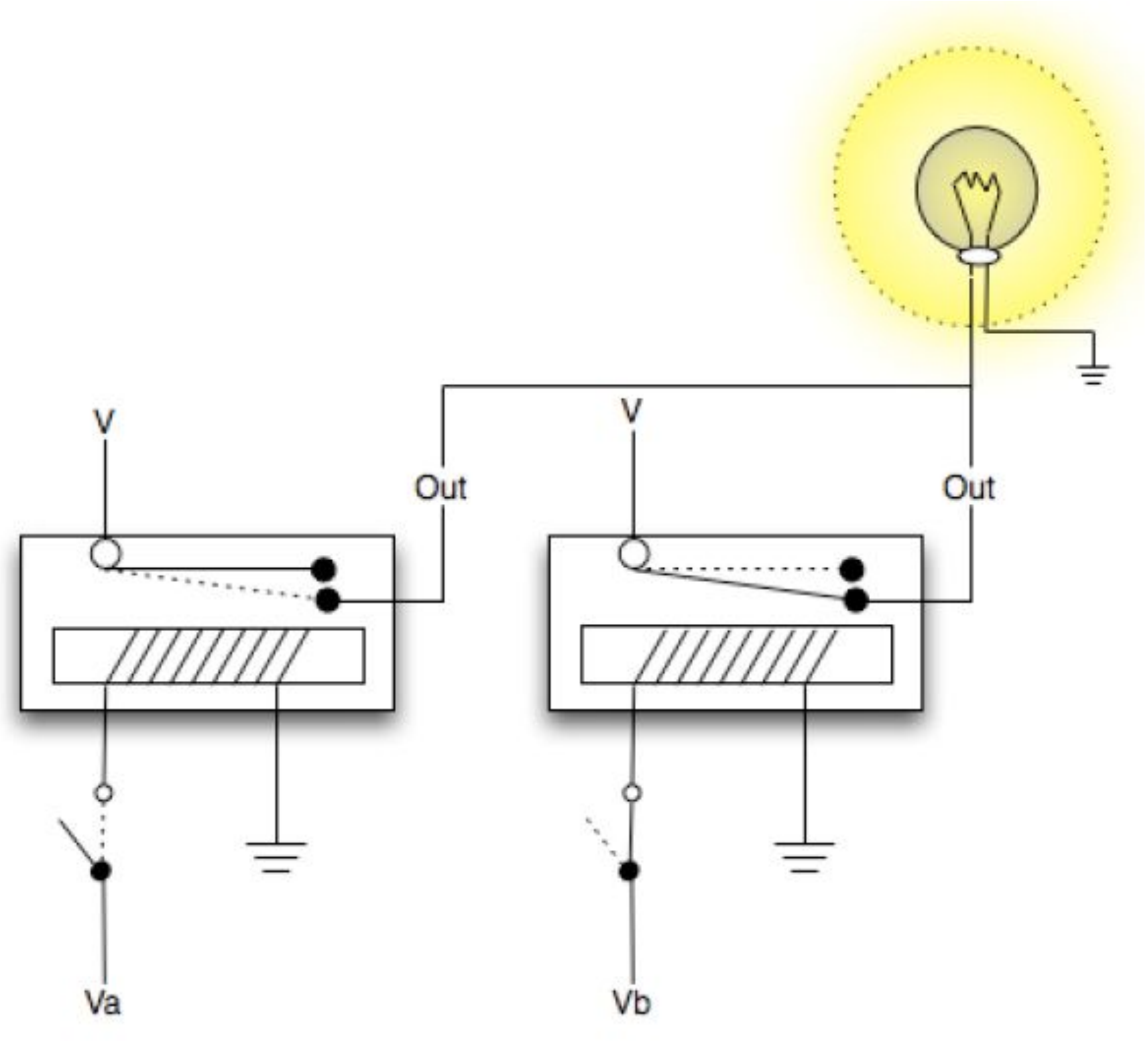


A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

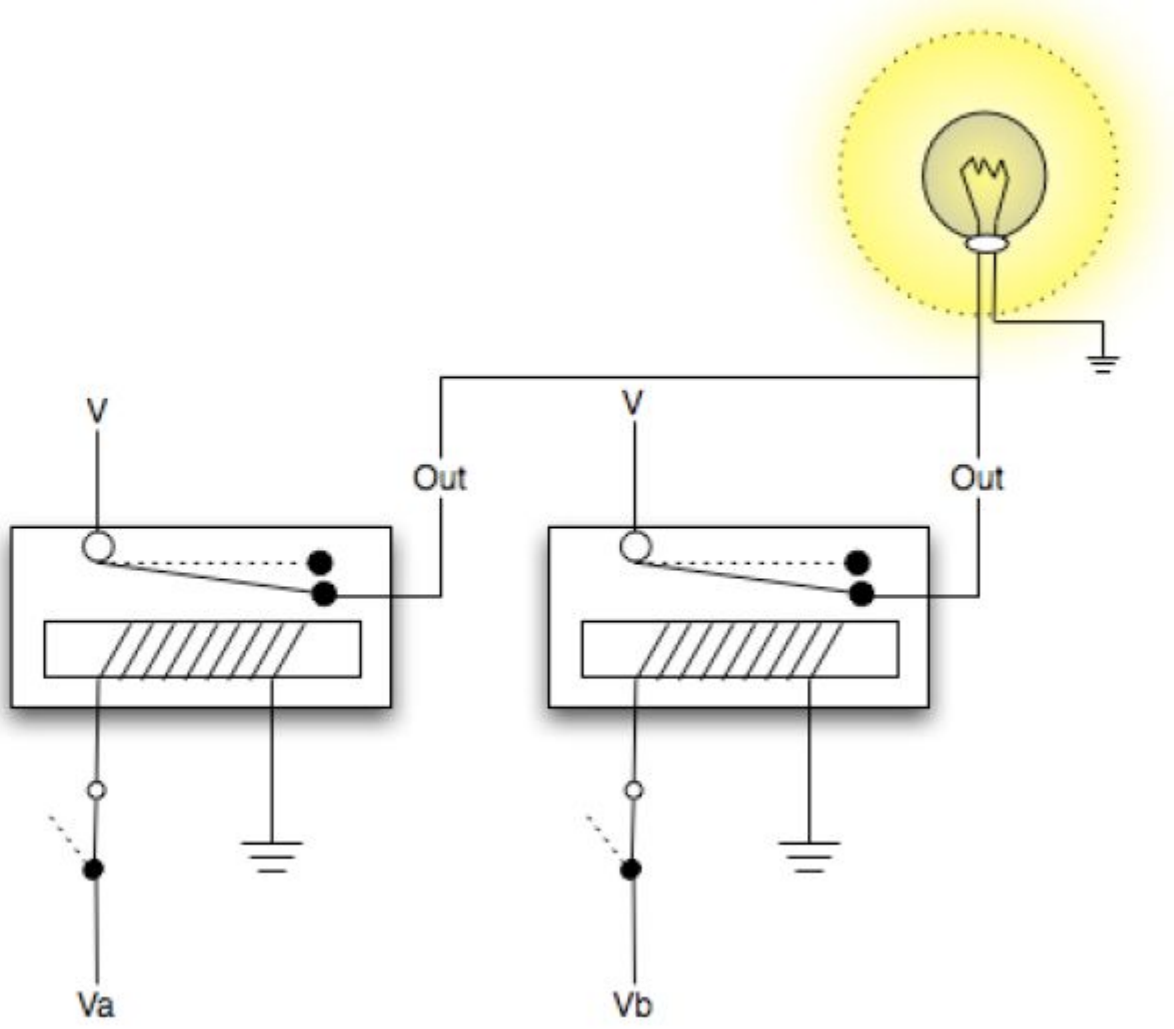
OR gate



OR gate

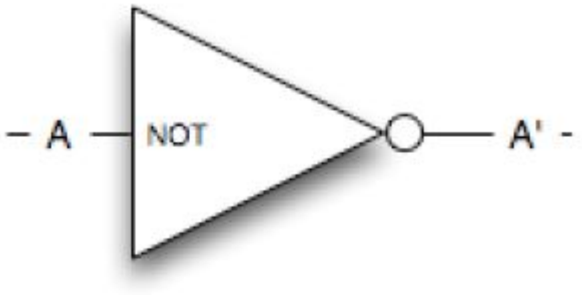


OR gate



NOT gate (Inverter)

- The Inverter is the simplest logic gate.
- The logical circuit has 1 input and 1 output.
- The output is the opposite of the input.



A	NOT A
0	1
1	0

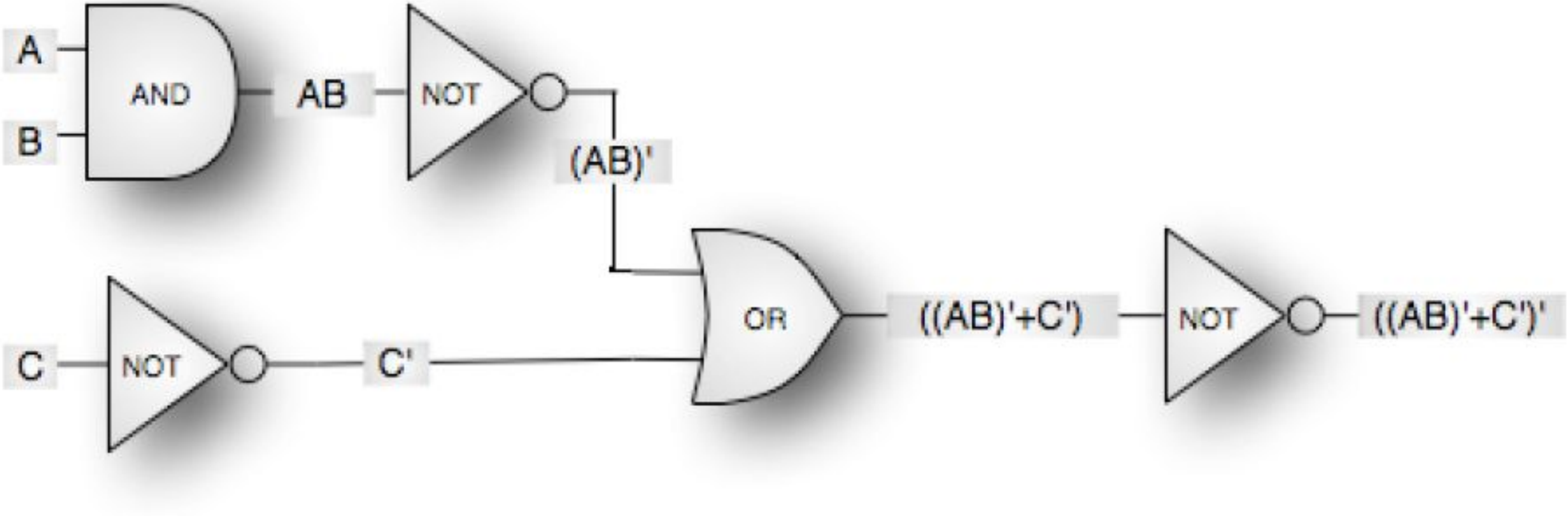
Combinational Circuits

- Logic gates are elementary building blocks.
- They can be connected to build more complex circuits (adder or subtractor)
- Combinational circuits are hardware circuits providing the logical function for electronic systems.
- These circuits are interconnected set of various gates where outputs connected to other inputs.
- Combinational circuits perform a logical function: if you change the inputs, you get a new output.

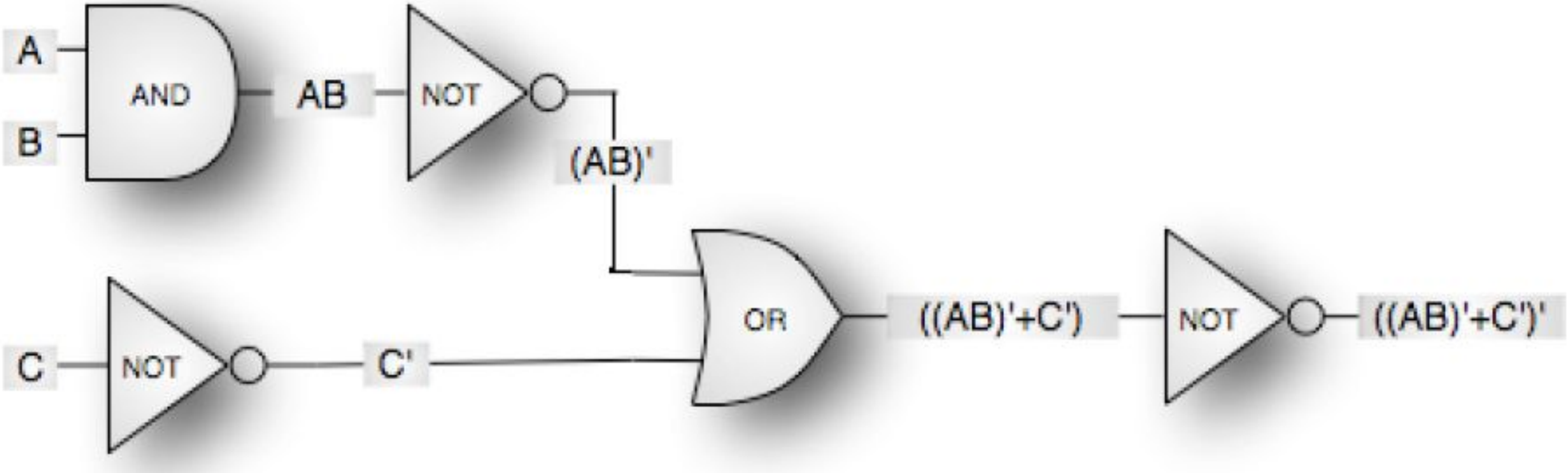
Combinational Circuits

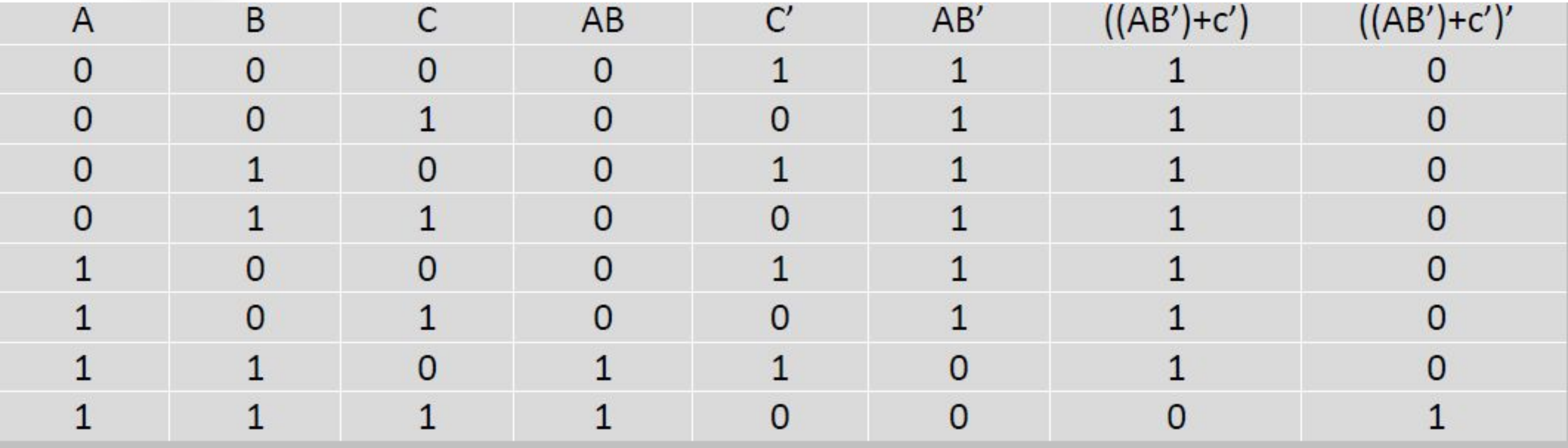
- The circuit below implements the Boolean expression

  ((AB)’+C’)’



Combinational Circuits

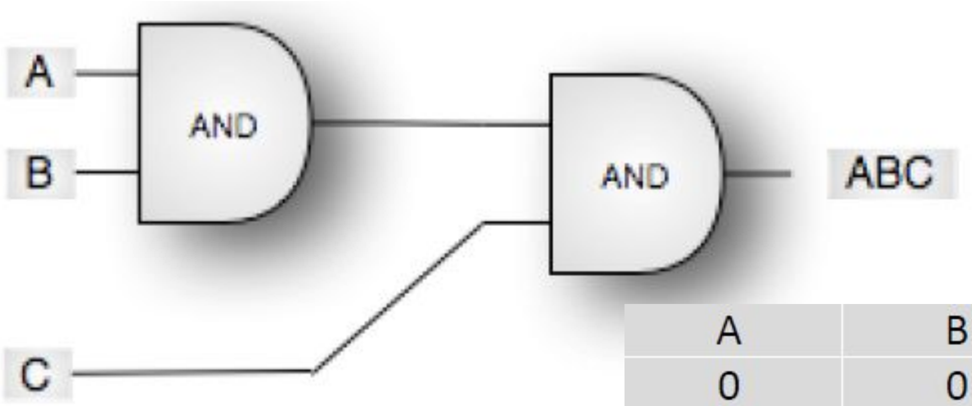




A	B	C	AB	C’	AB’	((AB’)+c’)	((AB’)+c’)’
0	0	0	0	1	1	1	0
0	0	1	0	0	1	1	0
0	1	0	0	1	1	1	0
0	1	1	0	0	1	1	0
1	0	0	0	1	1	1	0
1	0	1	0	0	1	1	0
1	1	0	1	1	0	1	0
1	1	1	1	0	0	0	1

Combinational Circuits

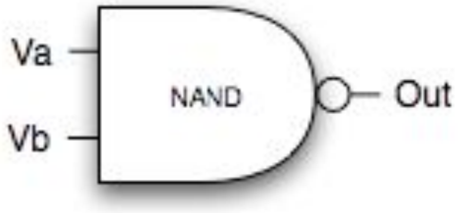
- Compare previous circuit to the following one, you will find that they are equivalent, they produce similar outputs:



A	B	C	AB	ABC
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	1	0
1	1	1	1	1

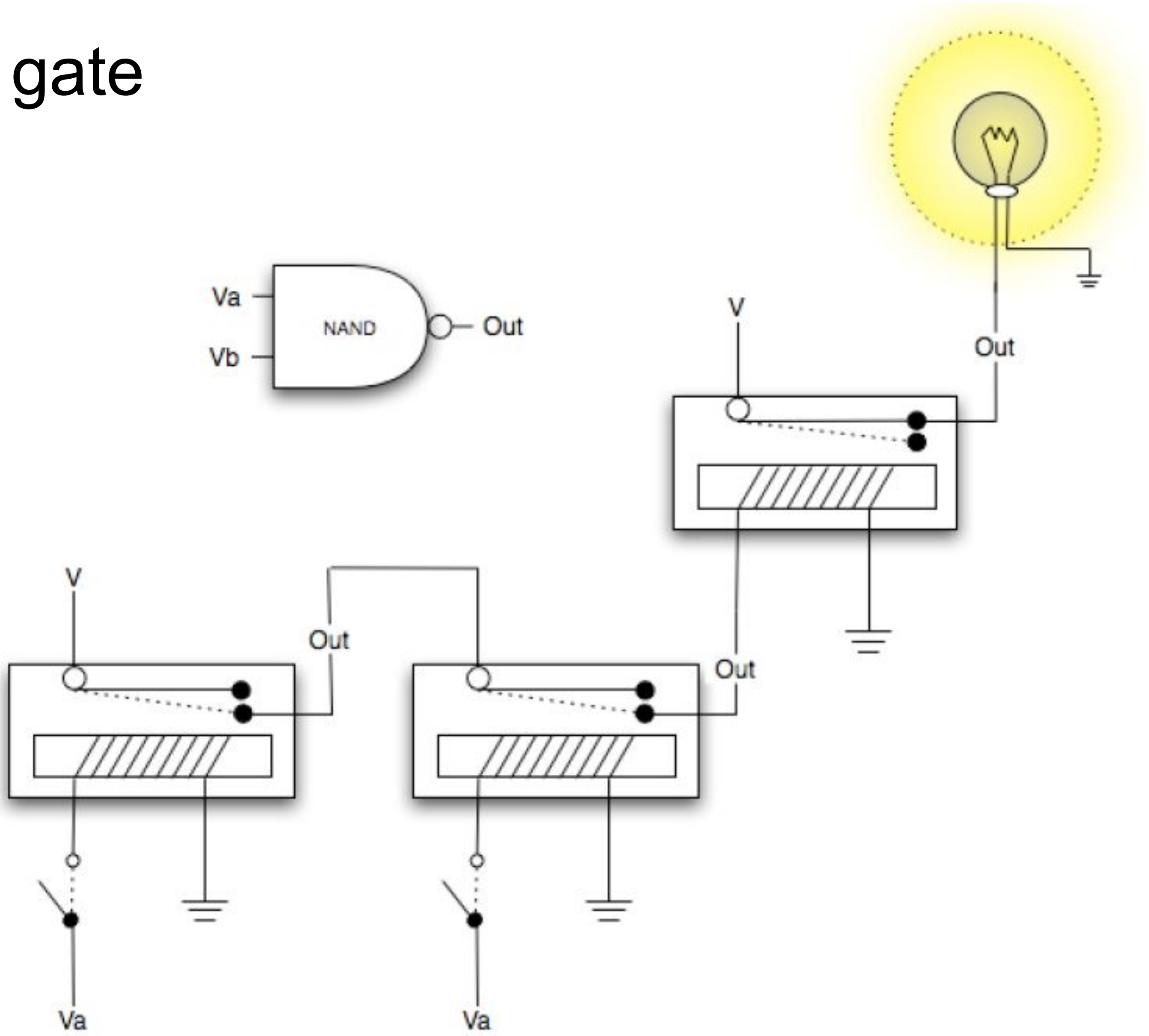
NAND gate

- (AB)’
- The logical NAND gate has 2 inputs and 1 output.
- It is equivalent to an AND gate followed by a NOT gate.
- Whenever at least one of the inputs is low, the output voltage will be high.
- If both inputs are high, then the output voltage will be low.



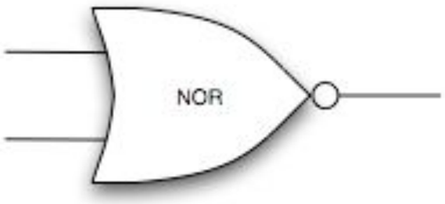
A	B	NAND
0	0	1
0	1	1
1	0	1
1	1	0

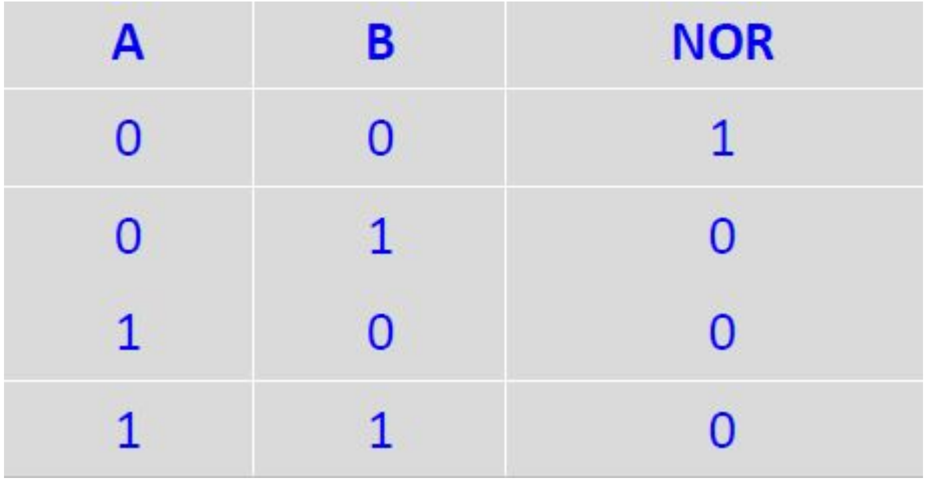
NAND gate



NOR gate

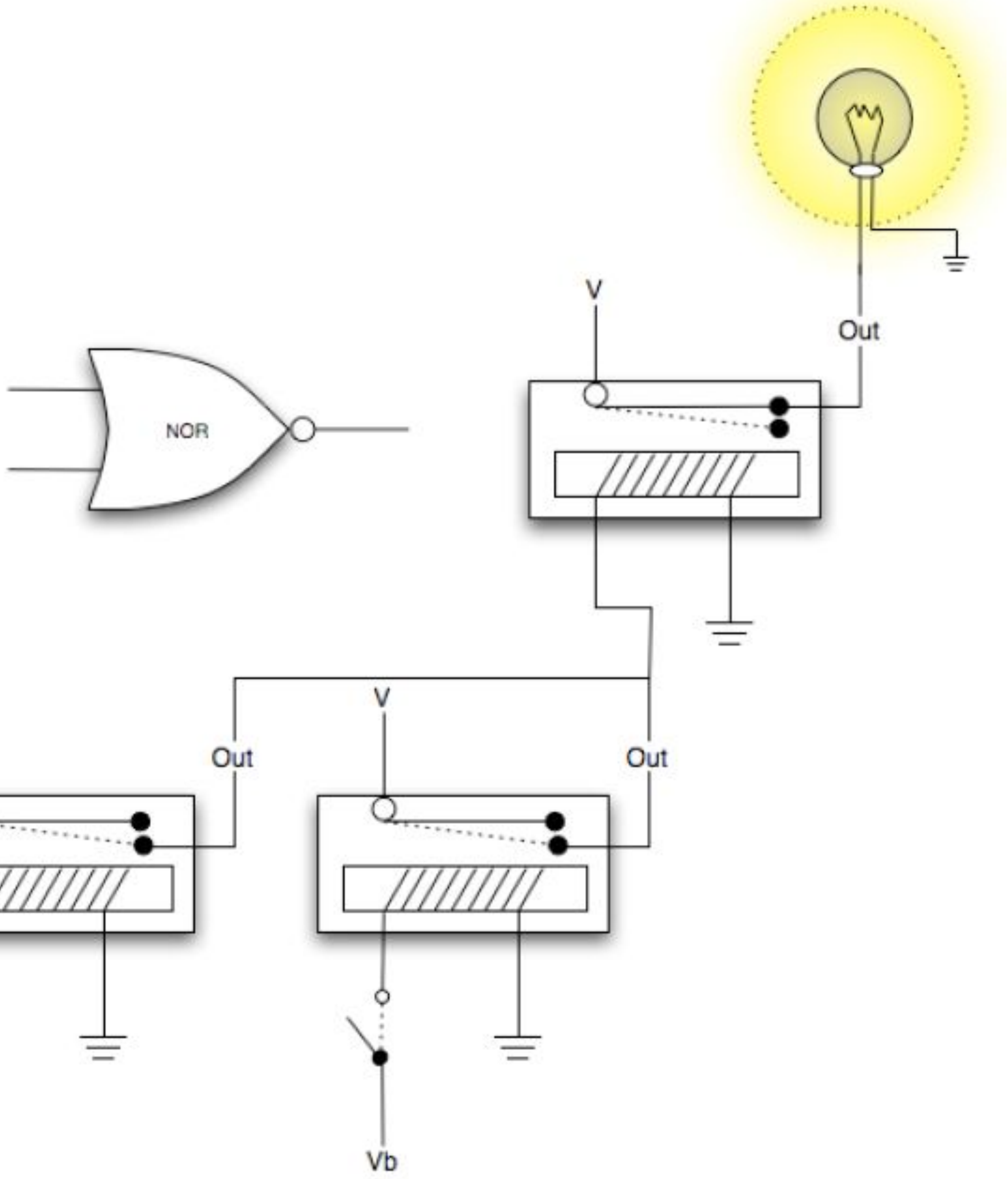
- (A + B)’
- The logical NOR gate has 2 inputs and 1 output.
- It is equivalent to an OR gate followed by a NOT gate.
- Whenever at least one of the inputs is high, the output voltage will be low.
- If both inputs are low, then the output voltage will be high.





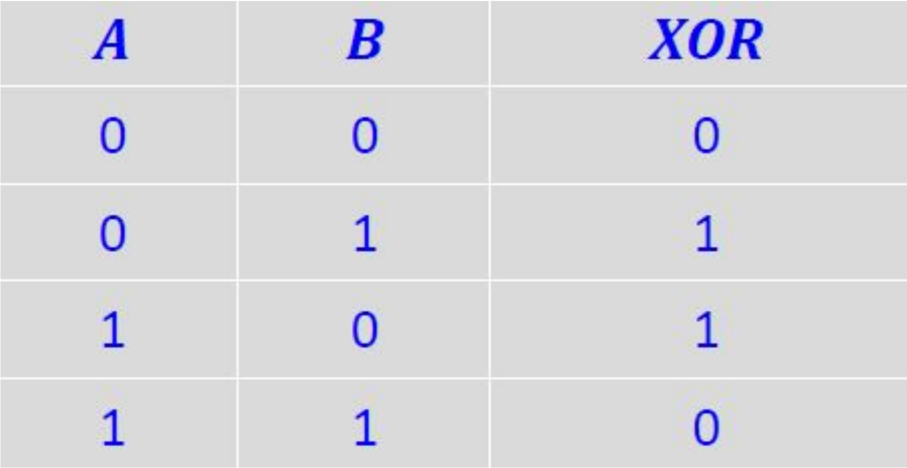
A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

NOR gate



XOR gate

- **eXclusive OR**.
- The logical XOR gate has 2 inputs and 1 output.
- Whenever only one of the inputs is high, the output voltage will be high.
- If both inputs are low, then the output voltage will be low.
- If both inputs are high, then the output voltage will be low.



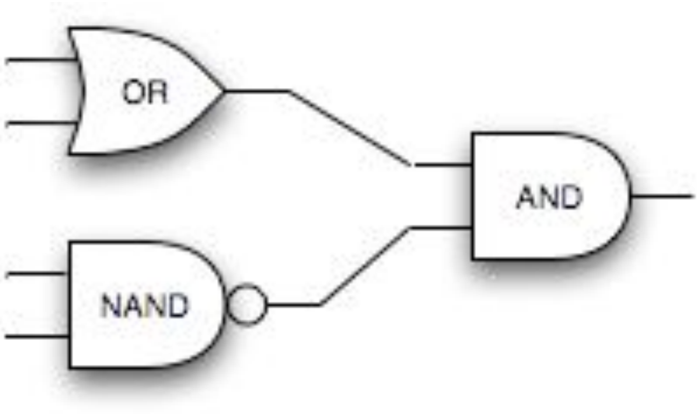
A	*B*	*XOR*
0	0	0
0	1	1
1	0	1
1	1	0

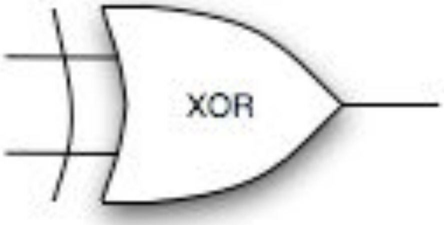
XOR gate

$$A \cdot B' + A' \cdot B$$

$$(A + B) \cdot (A' + B')$$

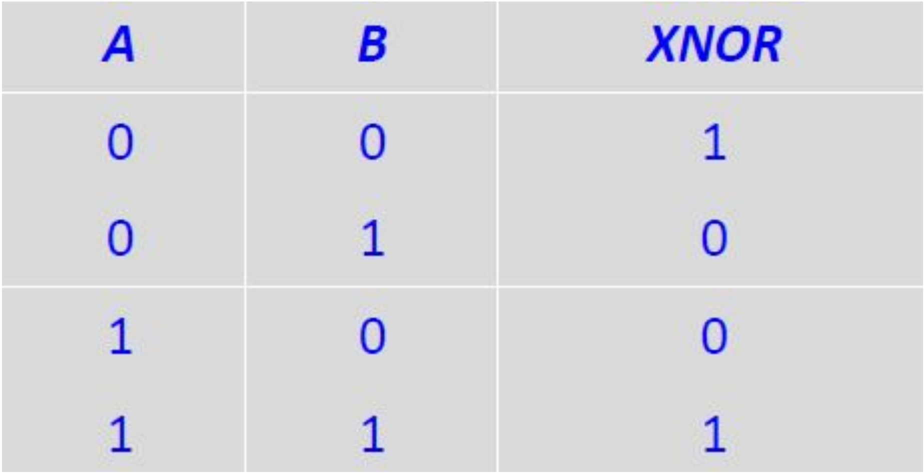
$$(A + B) \cdot (A \cdot B)'$$





XNOR gate

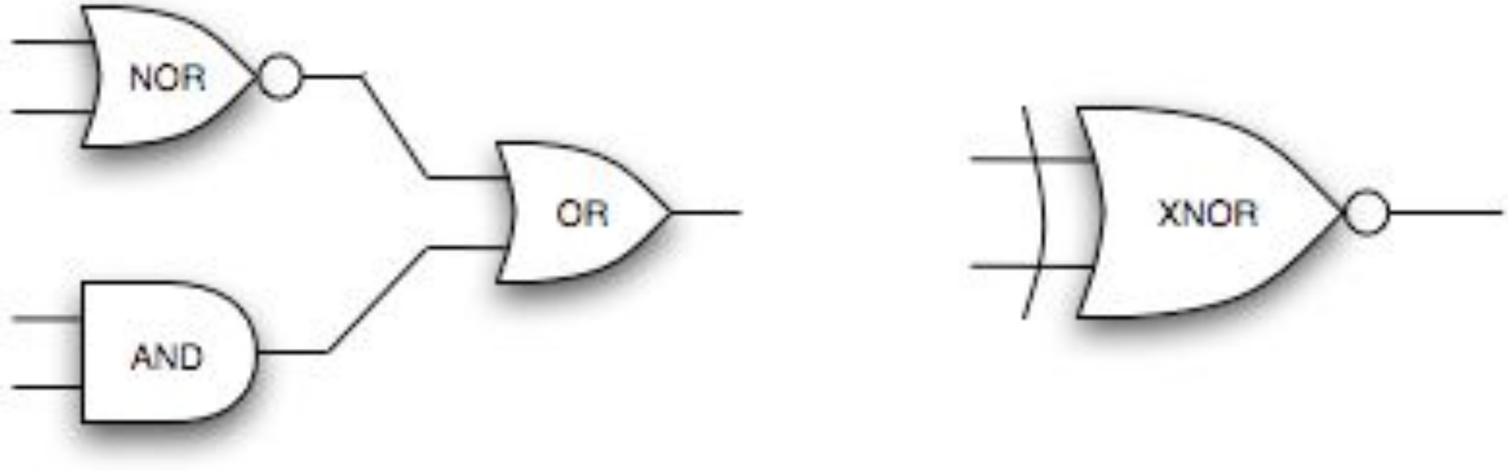
- eXclusive NOR gate
- The logical XNOR gate has 2 inputs and 1 output.
- It is equivalent to an XOR gate followed by a NOT gate.
- Whenever only one of the inputs is high, the output voltage will be low.
- If both inputs are low, then the output voltage will be high.
- If both inputs are high, then the output voltage will be high.



A	*B*	*XNOR*
0	0	1
0	1	0
1	0	0
1	1	1

XNOR gate

$$(A + B)' + (A \cdot B)$$



Conversion From Logical Circuit to a Boolean Expression (1/2)

- Work out the circuit from the leftmost gate (input) and work forward towards the right (output).
- Build the corresponding Boolean expression from the innermost to the outermost level:
  - The leftmost gates (inputs) are the innermost operators of the Boolean expression.
  - The rightmost gate (the output) is the outermost operator of the Boolean expression.

Conversion From Logical Circuit to a Boolean Expression (2/2)

- The inputs of the circuit are assigned to variables of the Boolean expression.
- Write the Boolean operator that correspond to each gate:
  - OR gate translates to a +
  - AND gate translates to a *
  - NOT gate translates to a '
- Add parenthesis as desired, especially around the innermost expressions.
- The order of the expressions does not matter but try to make the expression easy to read.

Simplification of combinational circuits (Gate reduction)

- Combinational circuits can become too large to handle.
- Optimizing the circuit is to eliminate gate redundancy.
- Gates are manufactured and integrated into chips.
- Fewer gates means smaller, less expensive and faster circuit
- Simplification can be:
    - Achieved manually using Boolean theorems & axioms
    - Automated using tabular methods
        - K-map method
        - Quine-McKluskey method

Next Time

- So far we covered binary numbering system, and the electricity leading to logical gates implementing the Boolean algebra.
- Since the binary numbering system can be represented using electricity, in the next class we will build a combinational logic circuit that will take as input two numbers and produce their sum as an output.
- The adder is the core unit of the CPU (Arithmetic unit). Similarly, logical gates are used to build the logical unit.
- Usually, they are called the ALU, the Arithmetic and Logical unit.