Microprocessor Design Bootcamp: Gentle Intro to Digital Circuits

December 16, 2025

Nursultan Kabylkas

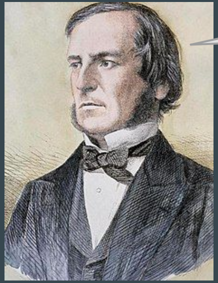
Agenda

- Small lecture
- Labs!

How do computers make decision?

At the core of all digital circuits is a system of logic, which allows computers to perform complex tasks by making simple, binary decisions

At the core of all digital circuits is a **system of logic**, which allows computers to perform complex tasks by making simple, **binary decisions**



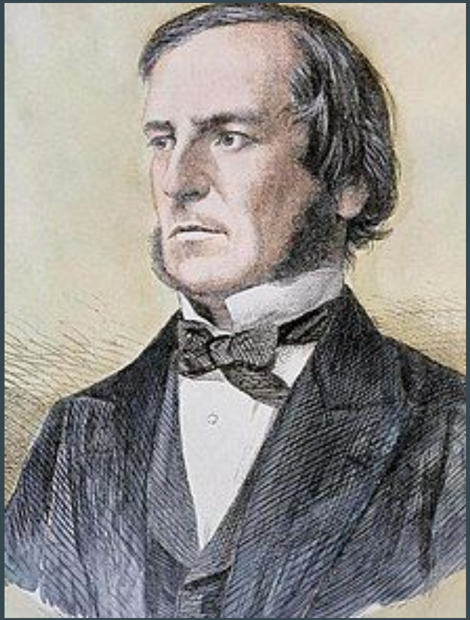
I wonder what is this system of logic?

Boolean algebra

- Boolean algebra is a branch of mathematics that deals with operations on logical values: true and false.

Boolean algebra

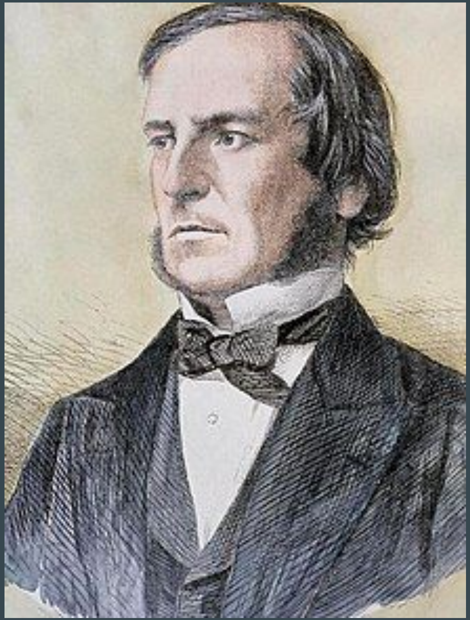
- Boolean algebra is a branch of mathematics that deals with operations on logical values: true and false.
- Introduced by George Boole, in the article “The Laws of Thought” in 1854.



George Boole
1815 - 1864

Boolean algebra

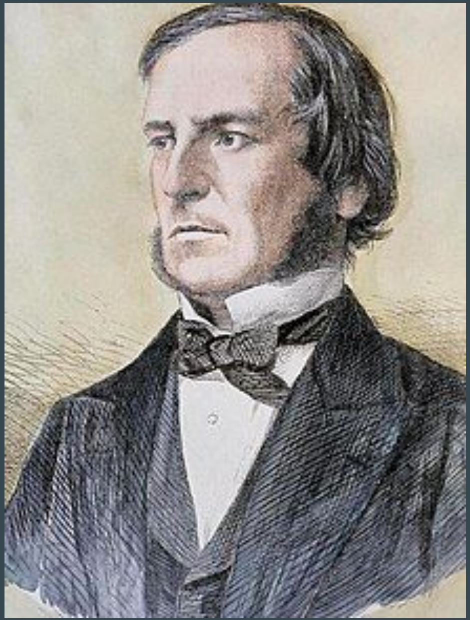
- Boolean algebra is a branch of mathematics that deals with operations on logical values: true and false.
- Introduced by George Boole, in the article "The Laws of Thought" in 1854.
- He basically said that we can express logic with algebraic expressions.



George Boole
1815 - 1864

Boolean algebra

- Boolean algebra is a branch of mathematics that deals with operations on logical values: true and false.
- Introduced by George Boole, in the article “The Laws of Thought” in 1854.
- He basically said that we can express logic with algebraic expressions.
- Defined operations such as: AND, OR, NOT.



George Boole
1815 - 1864

Example

- A: "I am tired."
- B: "I will take a nap."
- C: "I will drink coffee."

Let’s express as an equation: (A and B) or (not A and B). Interpretation: *If I am tired, I will take a nap OR if I am not tired, I will drink coffee.*

Example

Apply operations from boolean algebra to simplify:

Distributive law

$$A \cdot B + \neg A \cdot C = (A + \neg A) \cdot (B + C)$$

Complement law

$$(A + \neg A) \cdot (B + C) = 1 \cdot (B + C)$$

Example

Apply operations from boolean algebra to simplify:

Distributive law

$$A \cdot B + \neg A \cdot C = (A + \neg A) \cdot (B + C)$$

Complement law

$$(A + \neg A) \cdot (B + C) = 1 \cdot (B + C)$$

$$B + C$$

Interpretation: "I will either take a nap OR drink coffee."

Example

Apply operations from boolean algebra to simplify:

Distributive law

Complement law

$A \cdot B + \neg A \cdot C$

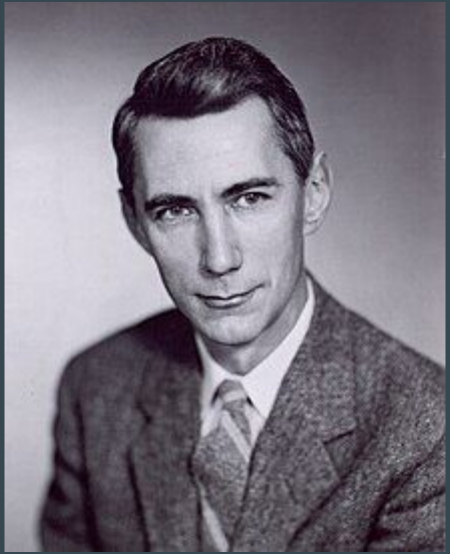
Key idea: “You can apply math on logic!”

$\Rightarrow B + C$

Interpretation: "I will either take a nap OR drink coffee."

Boolean algebra for digital circuits

- Claude Shannon was the first person to introduce the concepts of boolean algebra into electrical circuits.

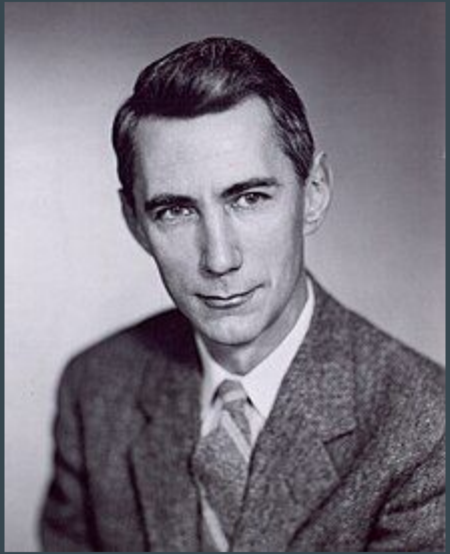


Claude Shannon
1916 - 2001

Boolean algebra for digital circuits

- Claude Shannon was the first person to introduce the concepts of boolean algebra into electrical circuits.

Probably the most impactful Master Thesis in history!
"A Symbolic Analysis of Relay and Switching Circuits". Less than 30 pages long.



Claude Shannon
1916 - 2001

AND operation

Theoretically

A	B	AB
F	F	?
F	T	?
T	F	?
T	T	?

AND operation

Theoretically

A	B	AB
F	F	F
F	T	F
T	F	F
T	T	T

AND operation

Theoretically

A	B	AB
F	F	F
F	T	F
T	F	F
T	T	T

Information wise

A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

AND operation

Theoretically

A	B	AB
F	F	F
F	T	F
T	F	F
T	T	T

Information wise

A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

Physically

A	B	AB

AND operation

Theoretically

A	B	AB
F	F	F
F	T	F
T	F	F
T	T	T

Information wise

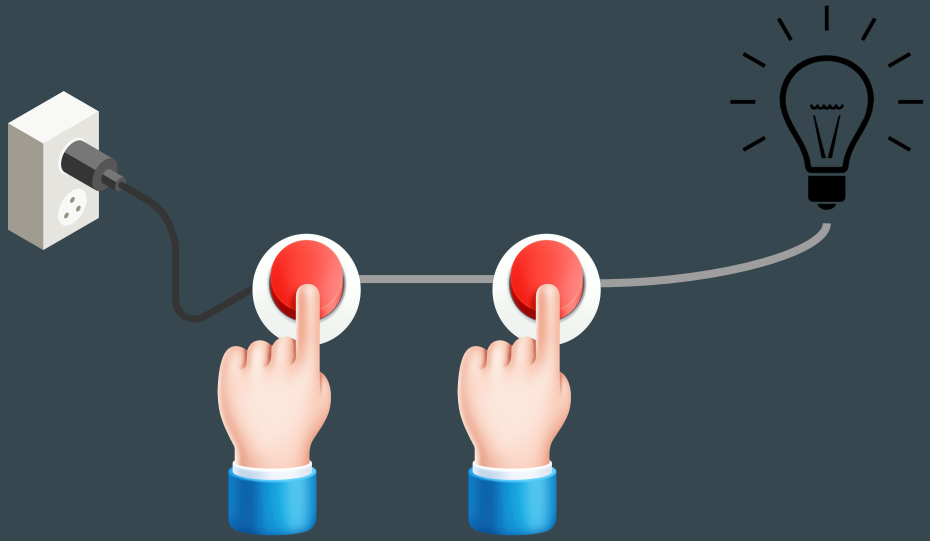
A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

Physically

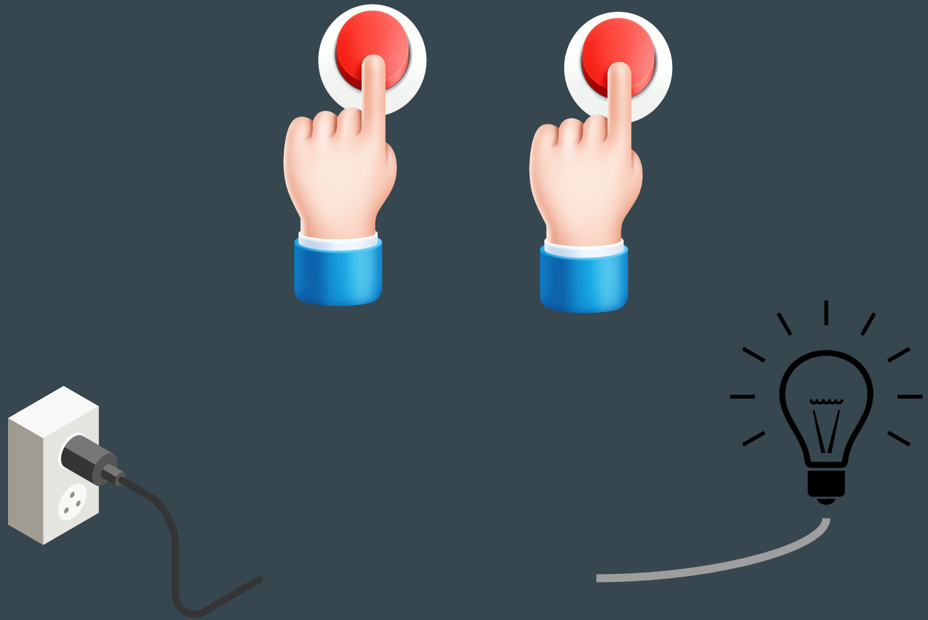
A	B	AB
0V	0V	0V
0V	5V	0V
5V	0V	0V
5V	5V	5V

AND operation

Pushing the button closes the circuit.



OR operation



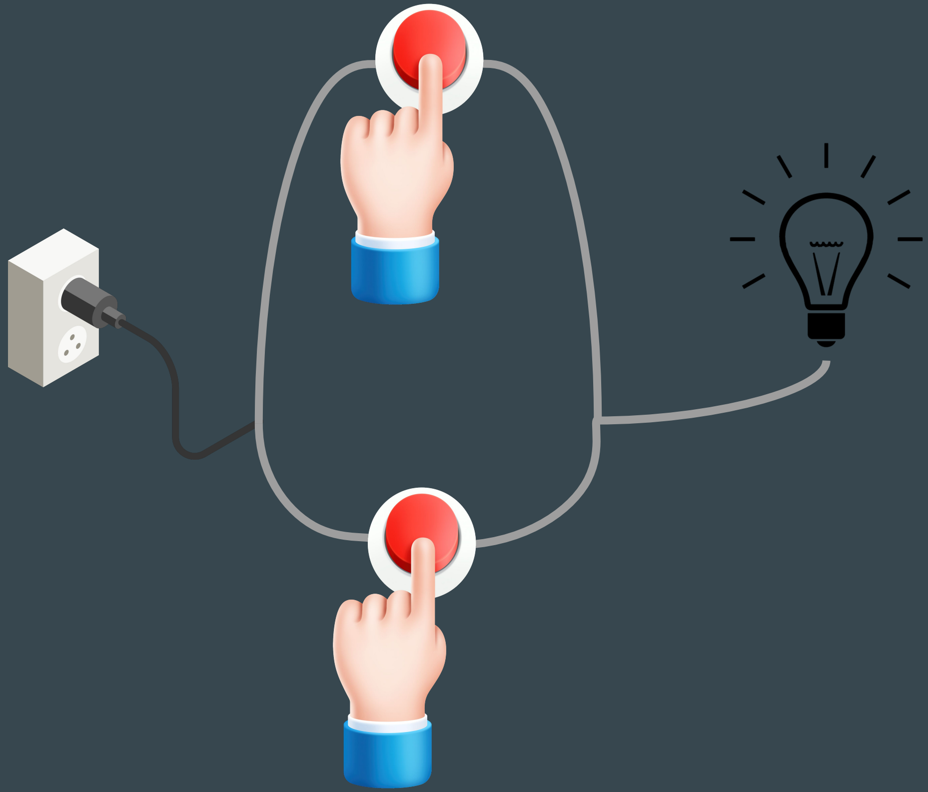
Pushing the button closes the circuit.

How would you place buttons to model OR operation?

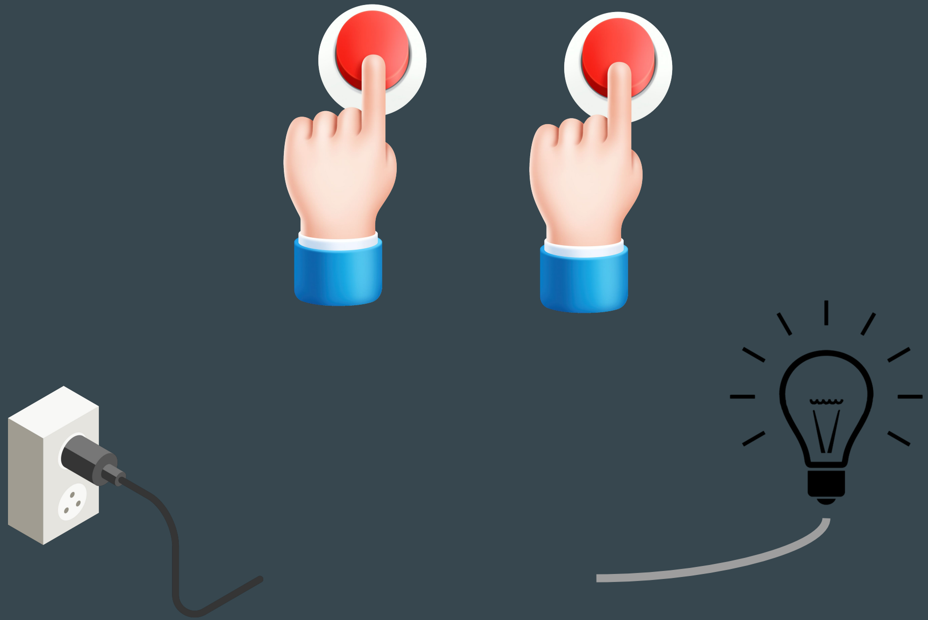
OR operation

Pushing the button closes the circuit.

How would you place buttons to model OR operation?



NOT operation

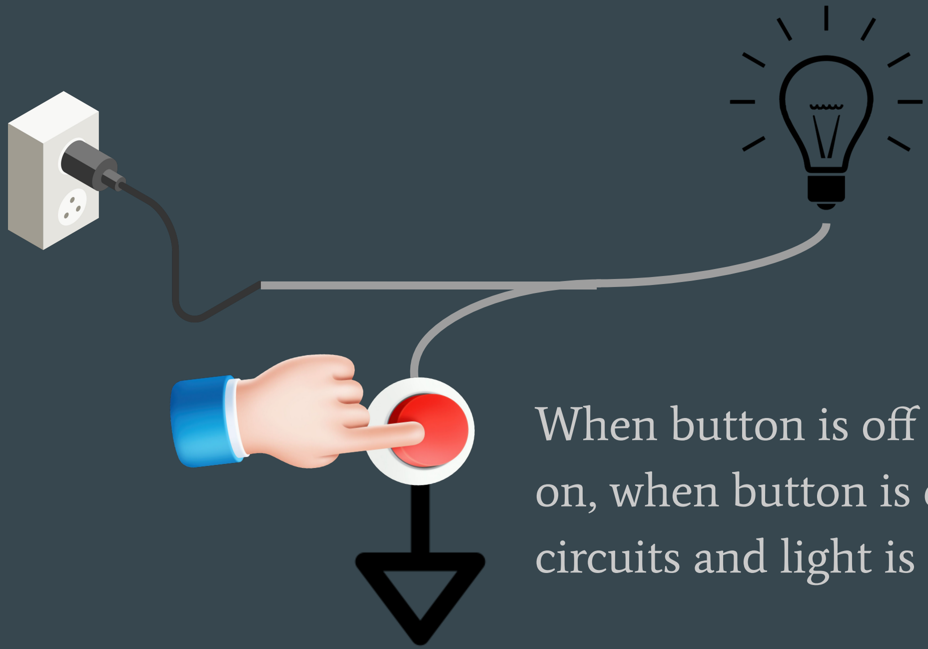


Pushing the button closes the circuit.

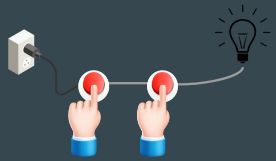
How would you place buttons to model NOT operation?

NOT operation

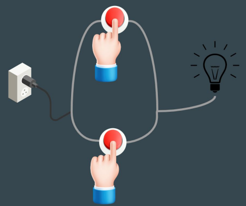
Pushing the button closes the circuit.



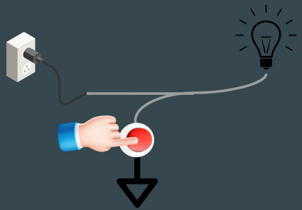
When button is off light is on, when button is on, short circuits and light is off.



AND

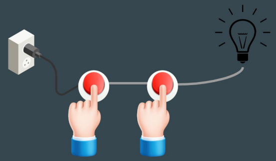


OR

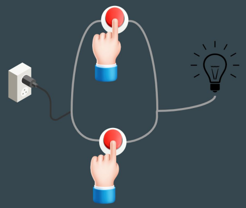


NOT

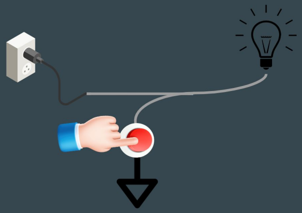
Obviously these configurations are far from the reality but the principle is true. We model the behaviour with electrical circuits.



AND

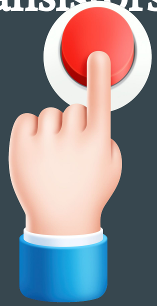


OR

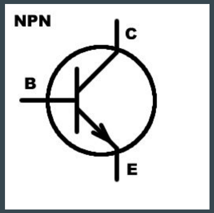


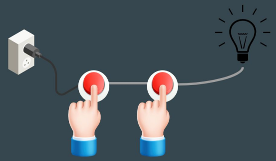
NOT

Obviously these configurations are far from the reality but the principle is true. We model the behaviour with electrical circuits. In modern processors, buttons/switches are transistors.

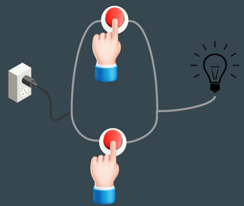


=

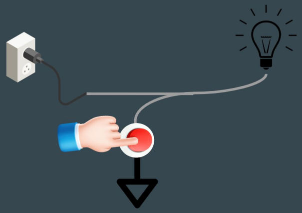




AND

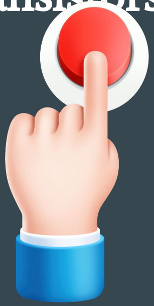


OR

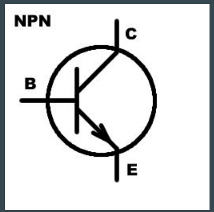


NOT

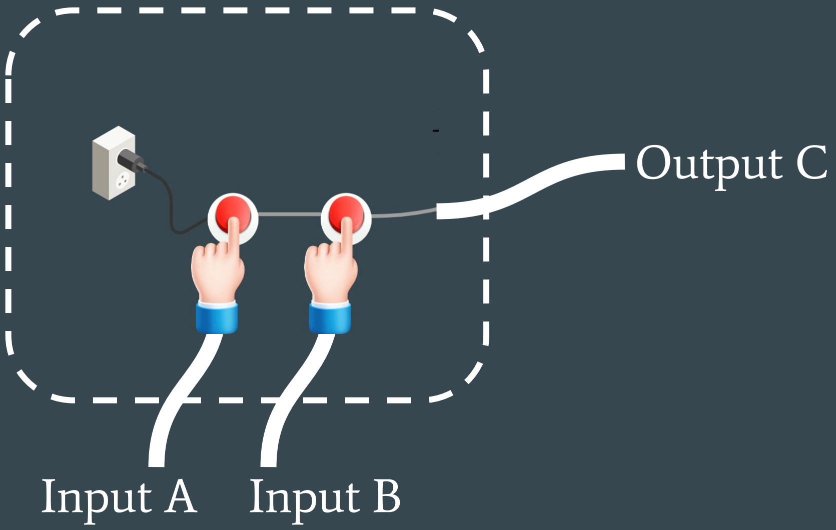
Obviously these configurations are far from the reality but the principle is true. We model the behaviour with electrical circuits. In modern processors, buttons/switches are transistors.



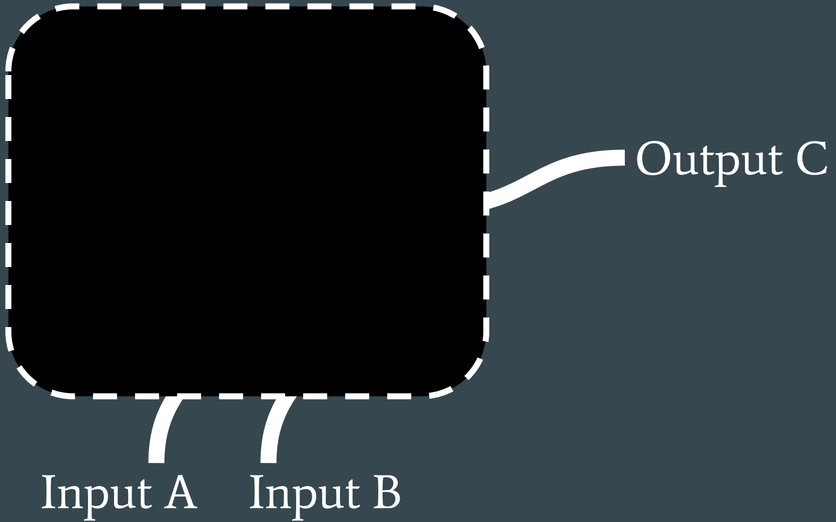
=



Let’s abstract

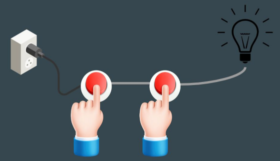


Let’s abstract

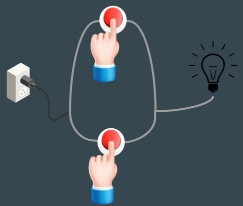


=

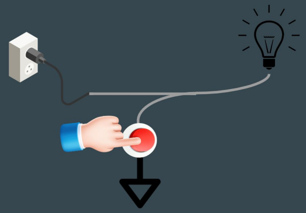
Let’s create a special symbol



AND



OR



NOT

Problem Statement

You need to design a basic home security system that activates an alarm when certain conditions are met. The system has three inputs:

- Window Sensor (W): True (1) if the window is open, False (0) if closed.
- Door Sensor (D): True (1) if the door is open, False (0) if closed.
- Motion Detector (M): True (1) if motion is detected, False (0) if no motion is detected.

The alarm (A) should sound (output True or 1) if The window is open or the door is open and motion is detected.

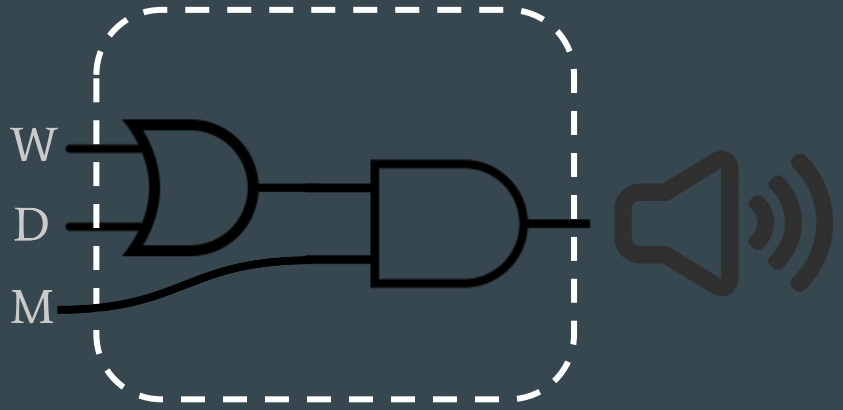
Solution

Expression = (W or D) and M

Solution

Expression = (W or D) and M

Let’s abstract again!



Let’s abstract again!

Let’s abstract again!

Abstraction is a key to the technological progress!

Let’s abstract again!

Abstraction is a key to the technological progress!