Information Technology for Digital Humanities

Lecture 6

Mario Verdicchio

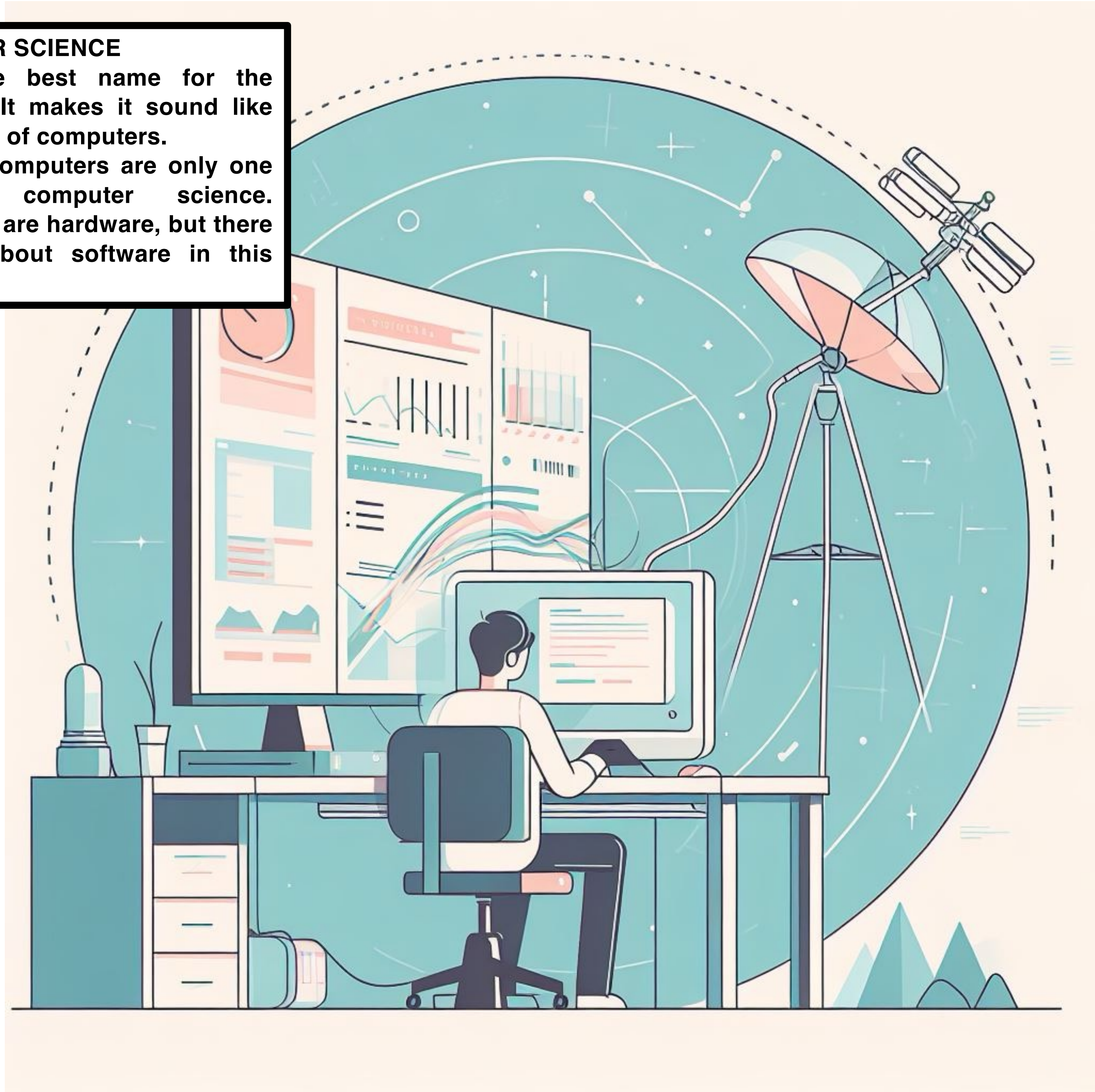
Università degli Studi di Bergamo

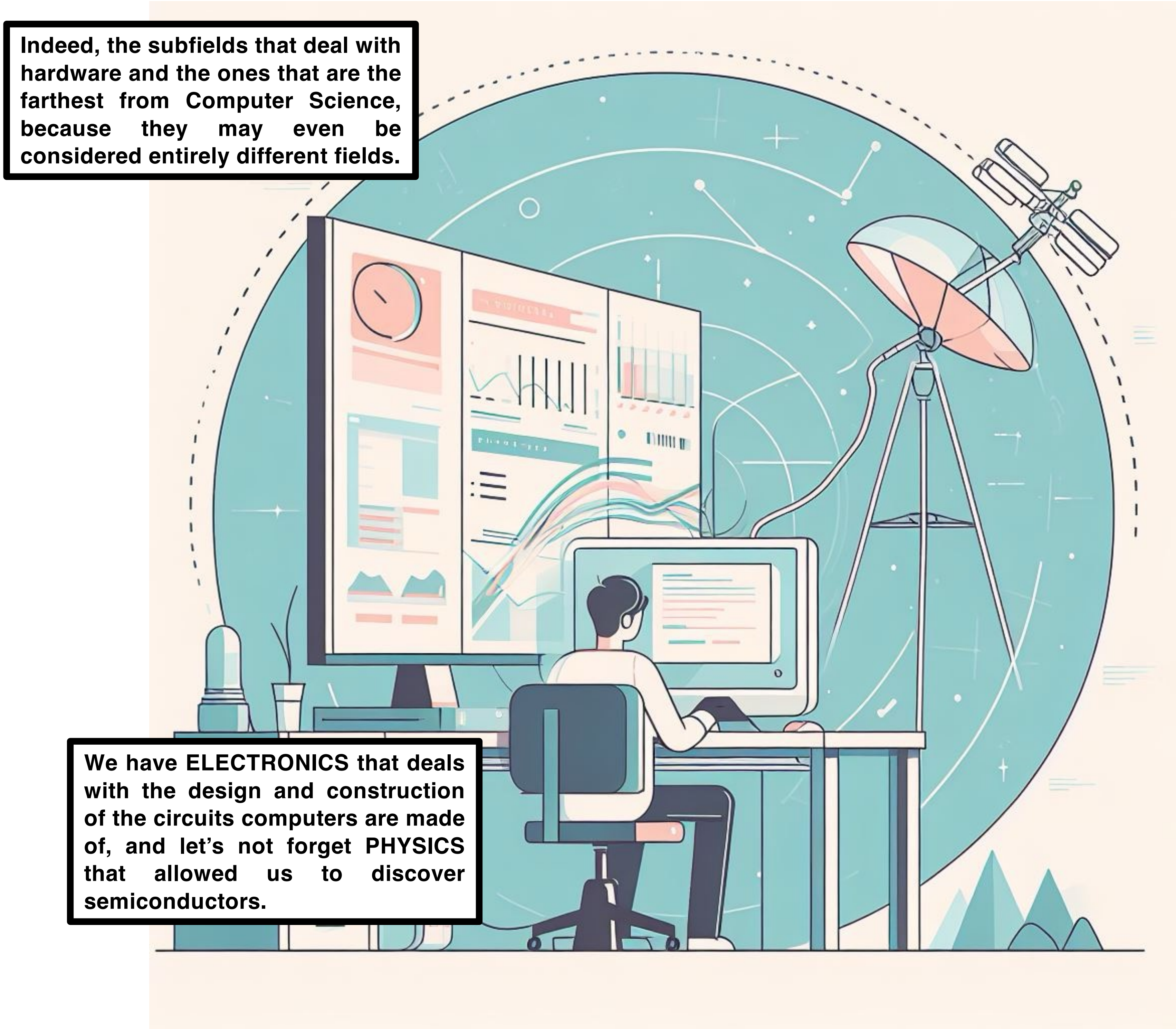
Academic Year 2024-2025

Lecture 6 (October 9 2024)

- Computer science subfields
- Logic gates

COMPUTER SCIENCE
is not the best name for the discipline. It makes it sound like the science of computers.
However, computers are only one part of computer science. Computers are hardware, but there is a lot about software in this discipline.




Indeed, the subfields that deal with hardware and the ones that are the farthest from Computer Science, because they may even be considered entirely different fields.
We have ELECTRONICS that deals with the design and construction of the circuits computers are made of, and let’s not forget PHYSICS that allowed us to discover semiconductors.

PHYSICS is involved also in the construction of the systems that connect computers to one another by means of cables, antennas, satellites, etc.

When physical systems are used for transmission of data over long distances, the discipline is called TELECOMMUNICATIONS (TLC).

A lot of the theory behind encodings comes from the early studies in telecommunications (think of the Morse code).

It is great to build computers, but they exist to be used. SOFTWARE ENGINEERING (SE) is the subfield of computer science that is aimed at the creation of software, in terms of programs and programming languages (artificial languages used to write programs).

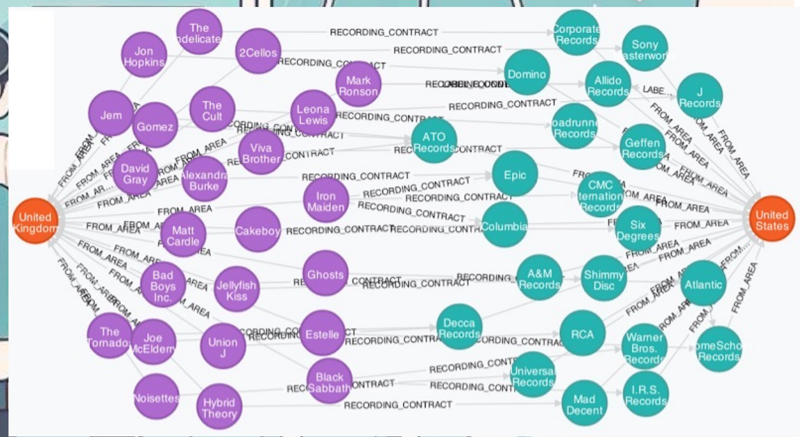

Programs make the computer world go around, but we know that programs are based on algorithms, and that algorithms are much more abstract concepts that can be described in mathematical terms or even natural languages like English, Italian or Japanese. THEORETICAL COMPUTER SCIENCE is the subfield that deals with algorithms: to understand what kind of real life problems can be tackled by means of algorithms and computation, and to understand what kind of resources (in terms of time and memory space) they would need to be brought to completion.

The connectivity ensured by telecommunications and the Internet has notably increased the quantity of data that computers have to deal with. DATABASE MANAGEMENT (DBMS) is a traditional subfield that is aimed at the organization and analysis of the data inside a computer. Typically, data are organized in the form of tables (also known as relational databases), but the more and more dynamic nature of data coming from the Internet is pushing for more flexibility, and in many systems a graph-like structured is preferred over tables (non-relational) for databases. Searches (like when you google stuff you don't know) and big data (massive amounts of data that are collected not only through the Internet but also by means of digital devices like smartphones, smartwatches, credit cards) are strongly emerging as key topics in this context, up to the point hat they might become independent subfields themselves.

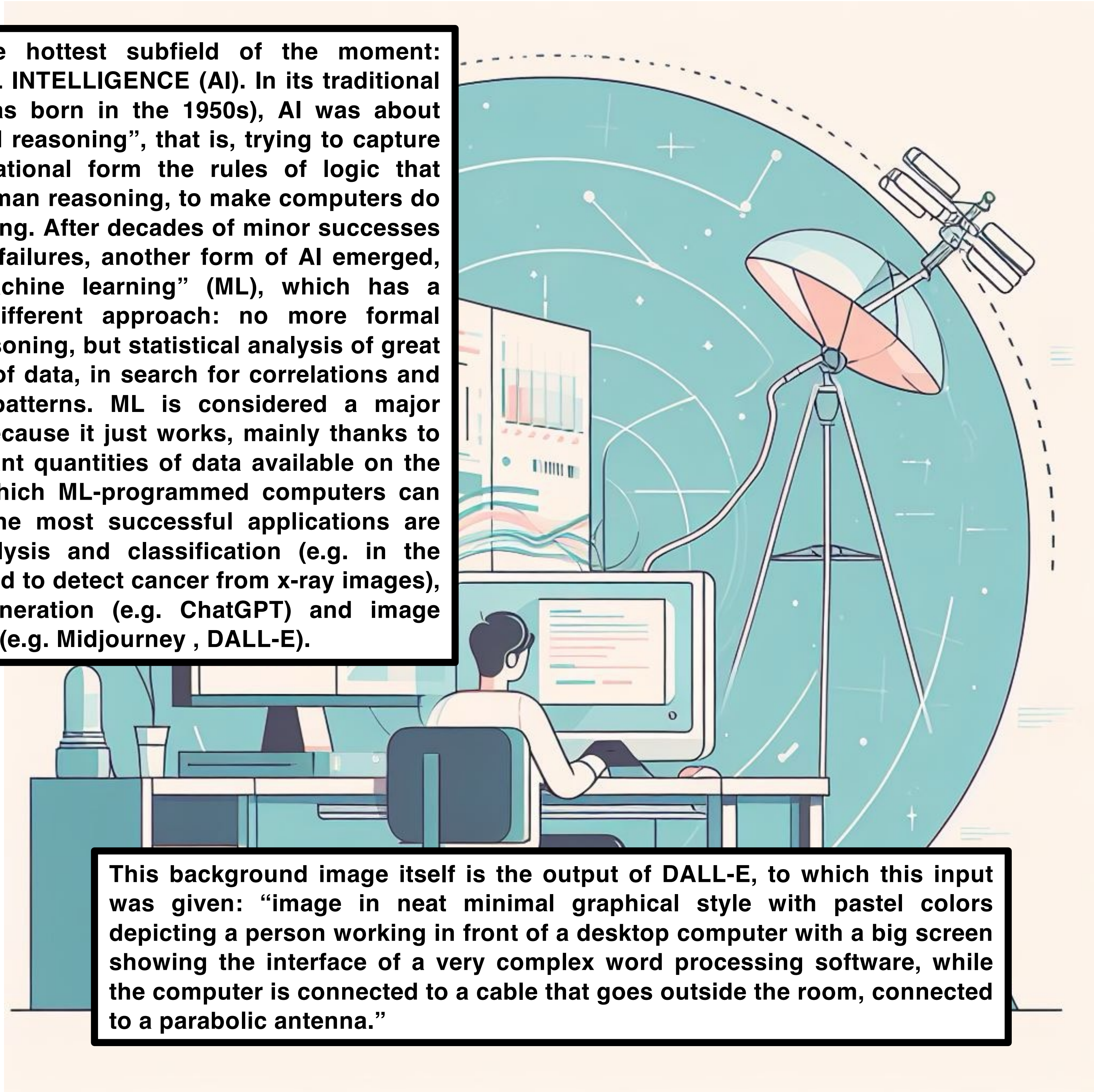
Table in a database

First Name	Last Name	Address	City	Age
Mickey	Mouse	123 Fantasy Way	Anaheim	73
Bat	Man	321 Cavern Ave	Gotham	54
Wonder	Woman	987 Truth Way	Paradise	39
Donald	Duck	555 Quack Street	Mallard	65
Bugs	Bunny	567 Carrot Street	Rascal	58
Wiley	Coyote	999 Acme Way	Canyon	61
Cat	Woman	234 Purrfect Street	Hairball	32
Tweety	Bird	543	Itotltaw	28

Graph-based database



Whatever is happening inside a computer, for us human users it is fundamental that its outside or, more precisely, that part of a computer's outside with which we interact works well. The point (or rather surface) of contact between a computer and its user is called "interface", and an appealing, captivating, easy-to-use interface needs to be designed to ensure a successful IT product. There are many disciplines that gravitate in this area (User Experience, Interaction Design, Graphic Design, etc.), which is genrally known as **HUMAN-COMPUTER INTERACTION (HCI).**



Finally, the hottest subfield of the moment: ARTIFICIAL INTELLIGENCE (AI). In its traditional form (it was born in the 1950s), AI was about “automated reasoning”, that is, trying to capture in computational form the rules of logic that sustain human reasoning, to make computers do the reasoning. After decades of minor successes and major failures, another form of AI emerged, called “machine learning” (ML), which has a radically different approach: no more formal logical reasoning, but statistical analysis of great quantities of data, in search for correlations and recurrent patterns. ML is considered a major success because it just works, mainly thanks to the abundant quantities of data available on the Internet, which ML-programmed computers can analyse. The most successful applications are image analysis and classification (e.g. in the medical field to detect cancer from x-ray images), or text generation (e.g. ChatGPT) and image generation (e.g. Midjourney , DALL-E).

This background image itself is the output of DALL-E, to which this input was given: “image in neat minimal graphical style with pastel colors depicting a person working in front of a desktop computer with a big screen showing the interface of a very complex word processing software, while the computer is connected to a cable that goes outside the room, connected to a parabolic antenna.”

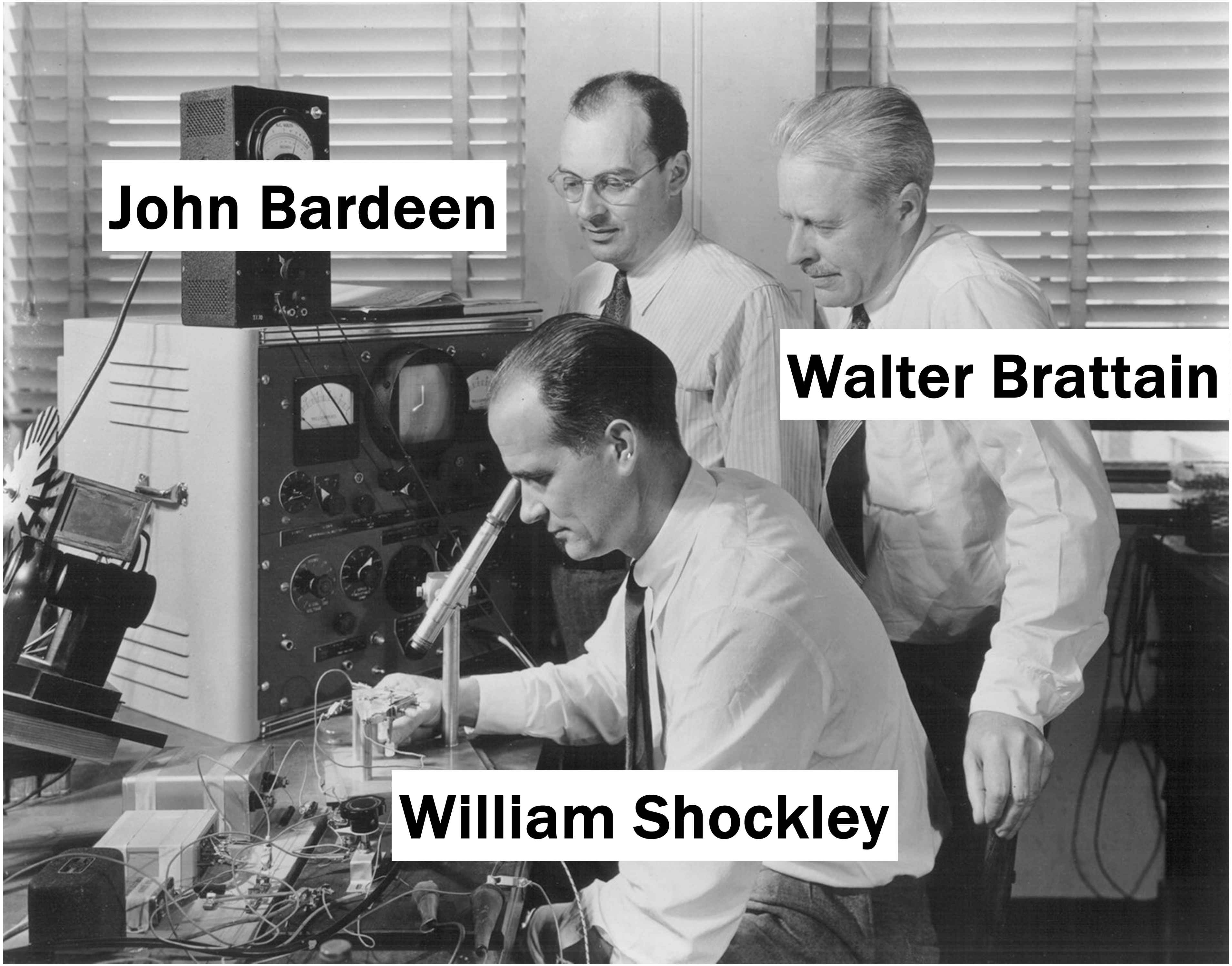
Where do DIGITAL HUMANITIES (DH) stand in all this? Are DH a subfield of Computer Science? They are not: they are rather an interdisciplinary effort, where problems in the humanities are analysed to understand whether they can be solved by means of a computer, and where enhancements offered by digital technology are explored to see whether they can impact traditional methodologies and operations in the humanities. It is a two-way street that is not confined within the realm of Computer Science, but it crosses it and takes us around in still unexplored and not entirely meaningful lands.

To understand how human choice maps onto parts of an electronic circuit, we must take a dive into the basic electronic components that constitute a computer hardware.

This technology dates back to the 1950s, when american scientists Bardeen, Brattain and Shockley invented the transistor, a small device based on a special property of semiconductive materials.

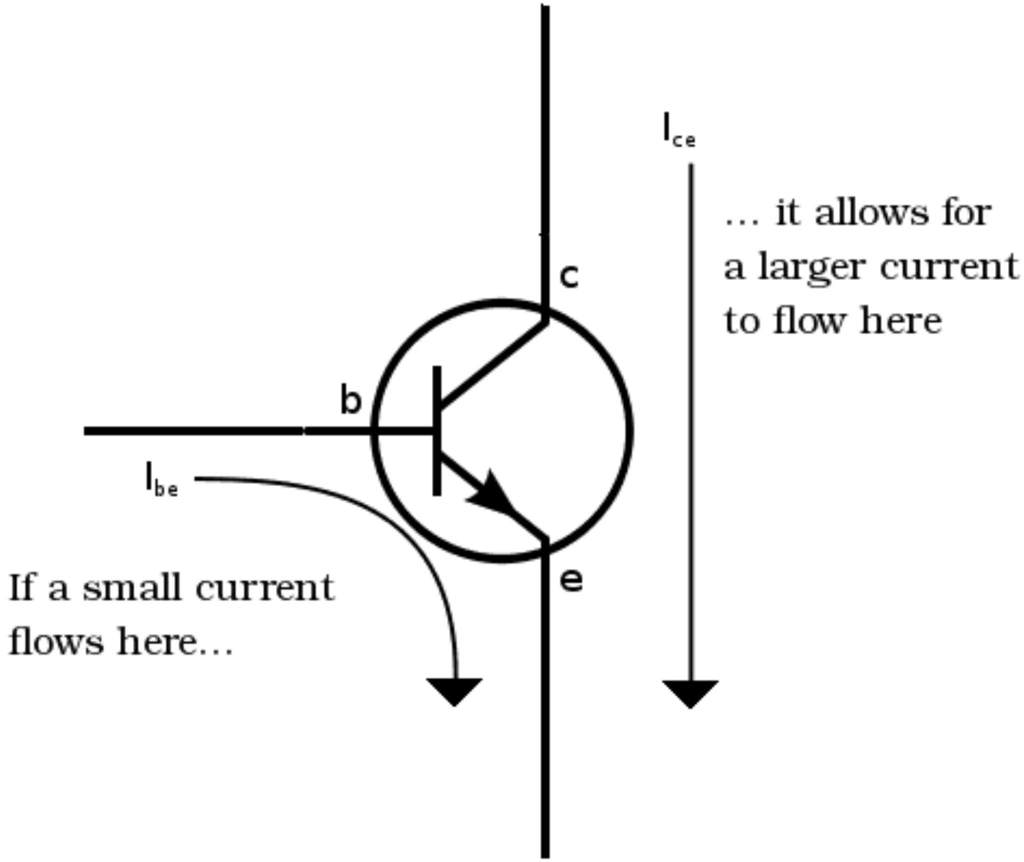
Semiconductive materials are particular: at rest, they do not allow electricity to flow, but if electrically stimulated they become conductive.

Hence, transistors are a special kind of switches: very small and electronic, that is, not mechanical but controlled by means of electricity.


John Bardeen
Walter Brattain
William Shockley

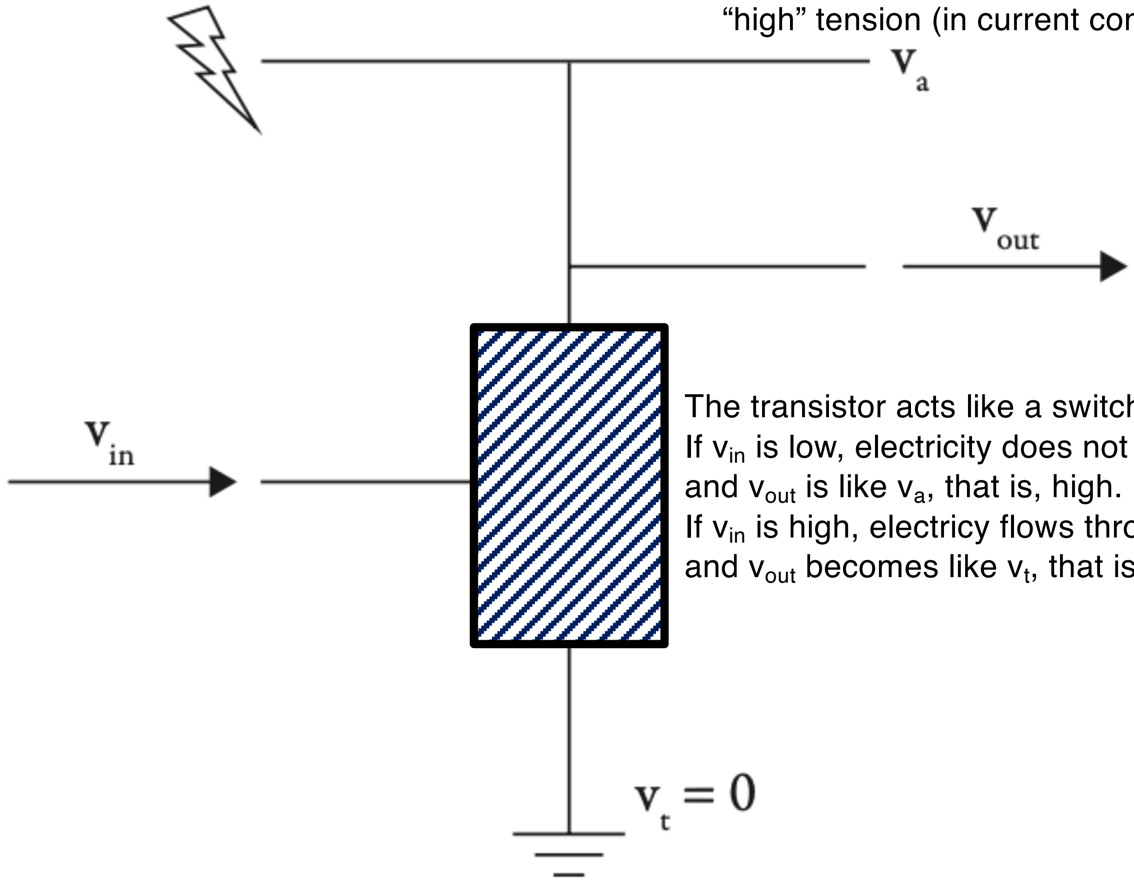
a transistor from the 1950s

how it works



A more abstract description of a transistor

“high” tension (in current computers 5V or 3V)



The transistor acts like a switch controlled by v_{in}.
If v_{in} is low, electricity does not flow through the transistor and v_{out} is like v_a, that is, high.
If v_{in} is high, electricy flows through the transistor and v_{out} becomes like v_t, that is, low.

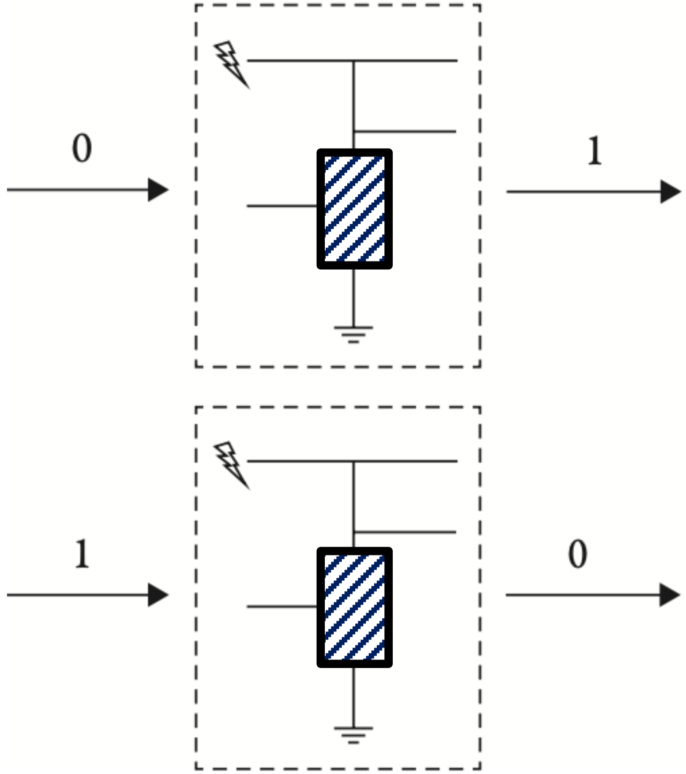
Ground: no tension here. Electricity flows from points of high tension to points of low tension.

The fundamental encoding inside a computer

v_t ------- 0

v_a ------- 1

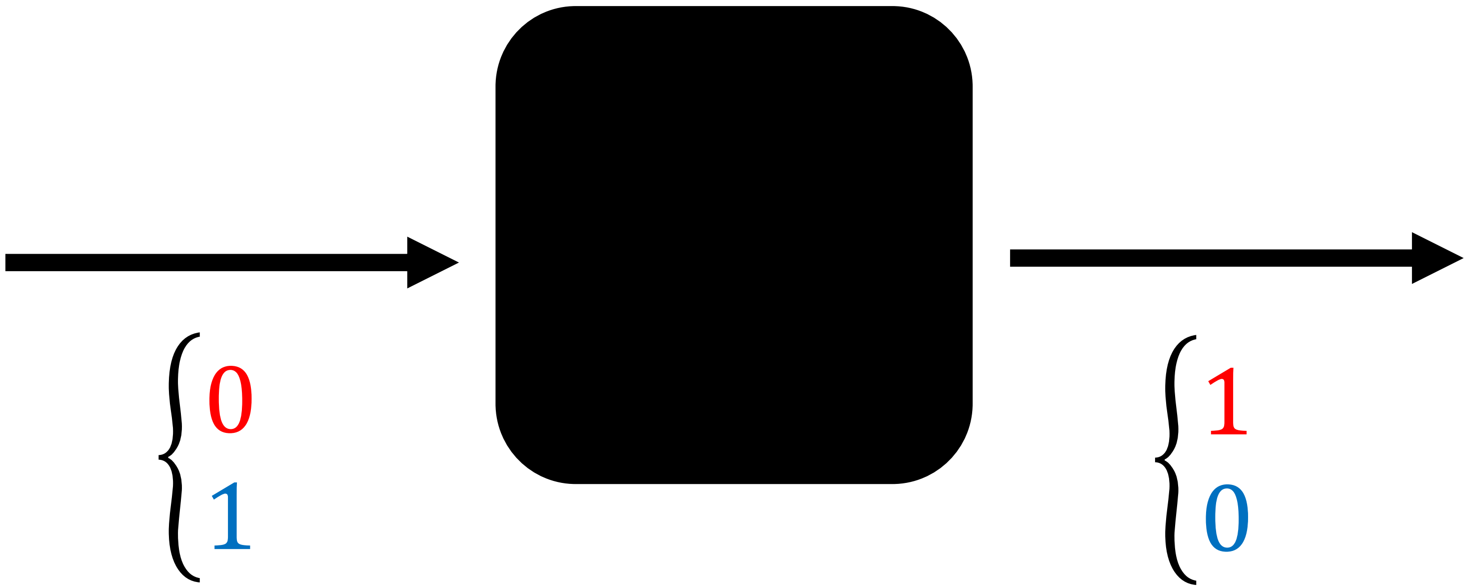
We interpret the high tension inside a computer like a “1”, and the low tension like a “0”.
This is an encoding: it maps two entities in the real world onto a set of natural numbers (0 and 1).



With such encoding in mind, a single transistor acts like a system which replies with “1” when we give it “0”, and viceversa.

$$\begin{cases} \color{red}{case\ 1} \\ \color{blue}{case\ 2} \end{cases}$$

We will use this graphical organization of multiple cases.
Case 1 is when we give the system “0” in input and obtain “1” in output.
Case 2 is when we give “1” in input and obtain “0” in output.



What to do with a system with this behavior?

v_t ------ 0 ---------------- **false**

v_a ------ 1 ---------------- **true**

If we use '0' to encode the concept of "false", and '1' to encode "true", we are moving from arithmetic with numbers to logic with truth values.

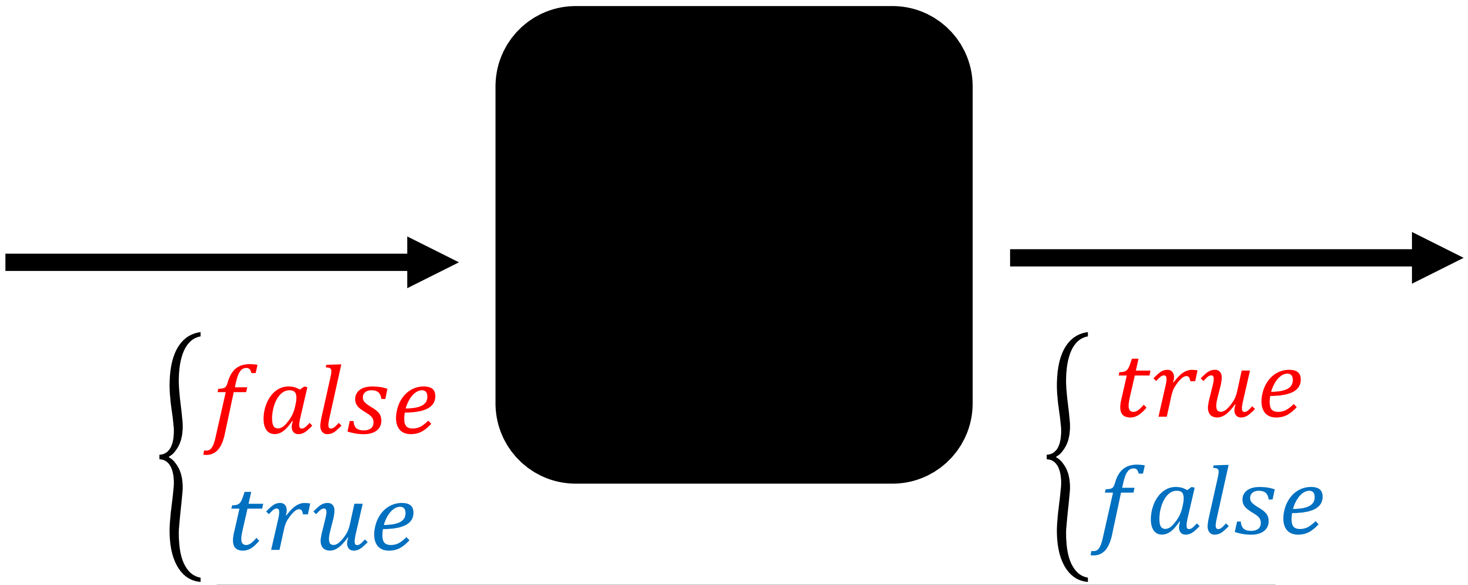
Let's introduce another encoding on top of the fundamental one.

Logic is the discipline that formalizes reasoning, that is, aims at making reasoning (e.g. "all men are mortal; Socrates is a man; hence, Socrates is mortal") rigorous by transforming sentences in sequences of symbols (called formulas) that are manipulated by means of rules.

"False" and "true" are "truth values" that we assign to formulas. Logic does not help us establish the truth values of hypotheses from the real world (e.g. "does God exist?") but it supports us in checking whether a certain way of reasoning is correct or not.

$\begin{cases} case\ 1 \\ case\ 2 \end{cases}$

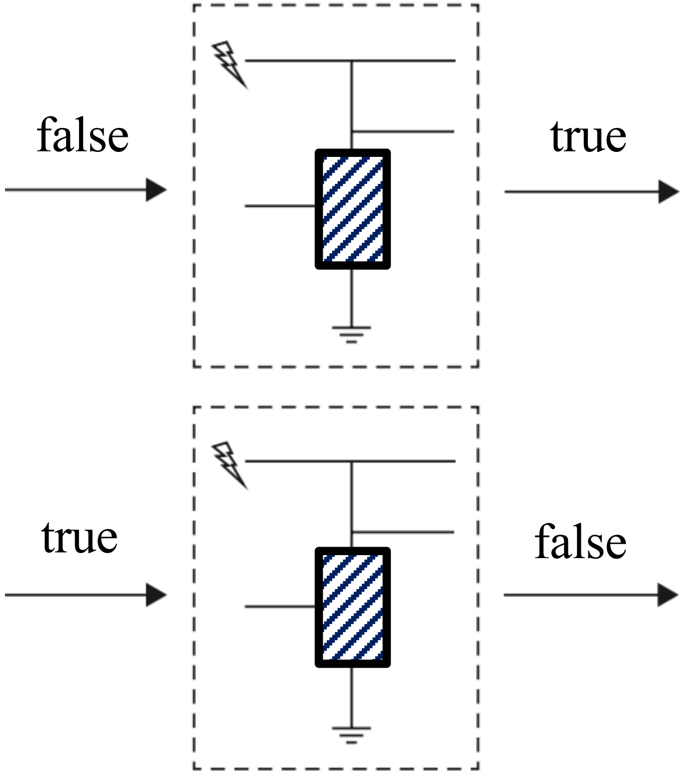
Let's take a look at the system again, with the logic-oriented encoding in mind.

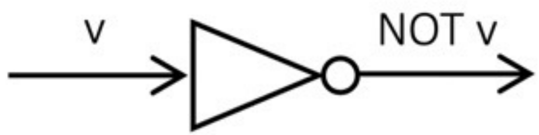


When we give it "false", it replies with "true".
When we give it "true", it replies with "false".

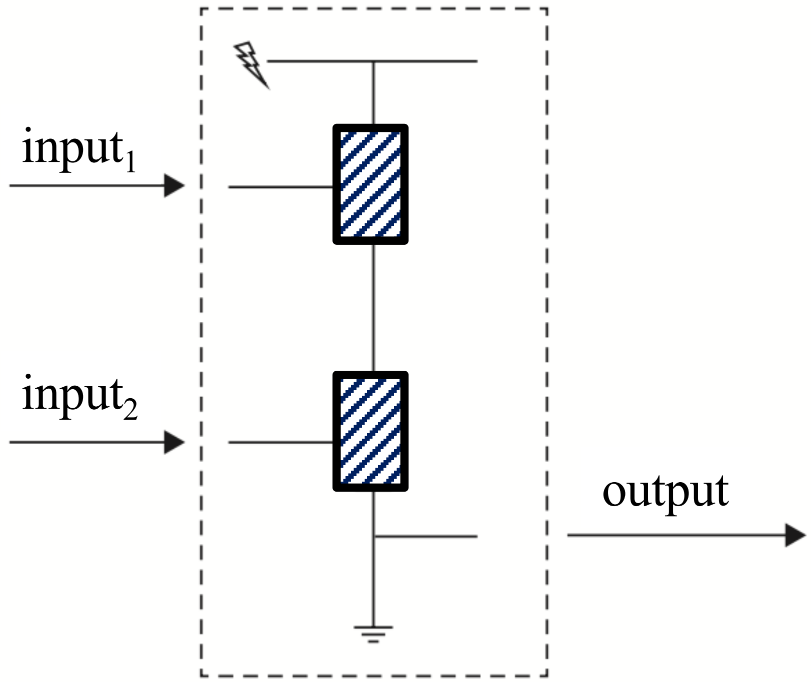
The system acts like a "not". Indeed, "not false" is the same as "true", and "not true" is the same as "false".

A transistor can then be seen as the electronic embodiment of the “negation” operator in logic.





This is how a system that negates, also known as “NOT gate” (”gate” because the electric signal goes through it), is represented in the graphical standard for circuit design.



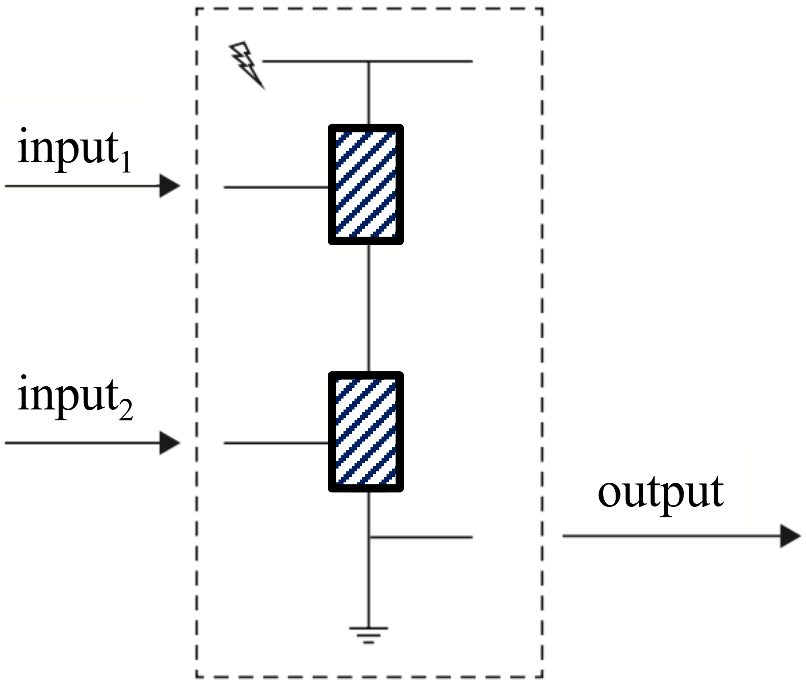
Let's build more complex systems, by using more than one transistor. This system has two inputs and one output.

The transistors are said to be in a "serial" configuration, because they are one after the other (i.e. in a series) between the high tension and the ground.

The tension in the output point is high only when the point is directly connected to the high tension above. This means that both transistors must be conductive, which means that both inputs must be high.

In numerical terms, since we have 2 inputs, we have 4 cases ($4 = 2^2$), and the output is 1 only in one case ($input_1 = 1$ and $input_2 = 1$)

$input_1$	$input_2$	output
0	0	0
0	1	0
1	0	0
1	1	1

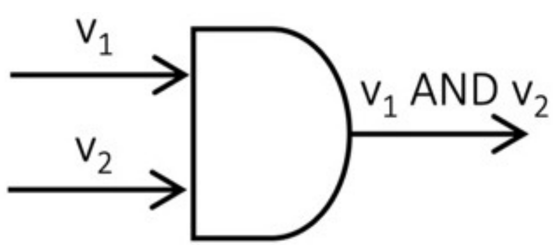


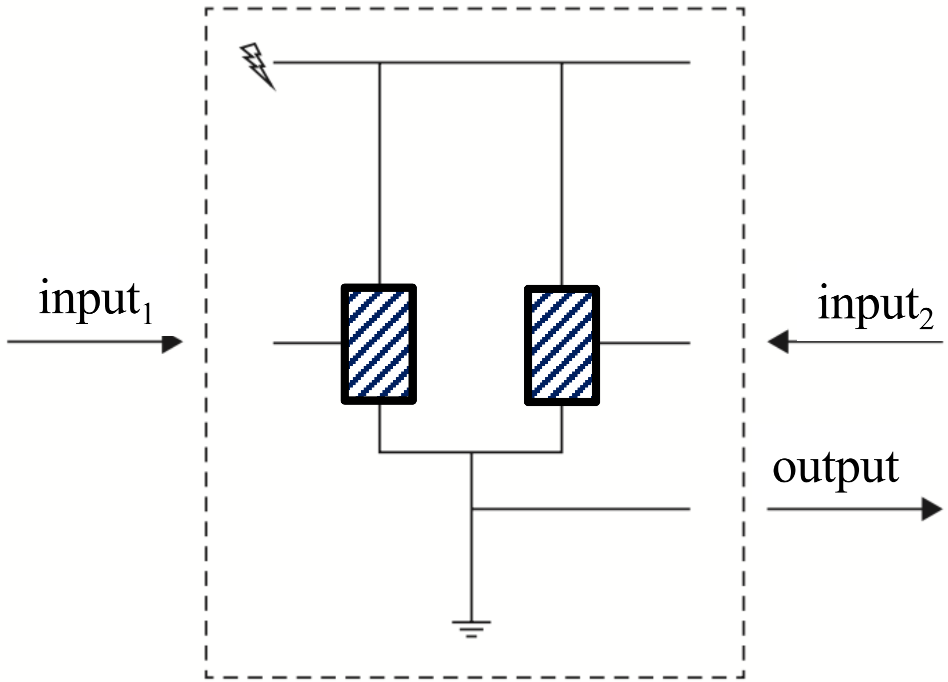
In logical terms, the output is "true" only when both inputs are "false". In other words, just one false input is enough to make the output false.

input$_1$	input$_2$	output
false	false	false
false	true	false
true	false	false
true	true	true

The two inputs are put together in the same way we use the "AND" conjunction: "the Sun is cold AND one plus one is two" is false, whereas "water is a liquid AND seven is an odd number" is true.
This circuit may be seen as the electronic version of the AND logical operator.

This is how an "AND gate" is represented in the graphical standard for circuit design.



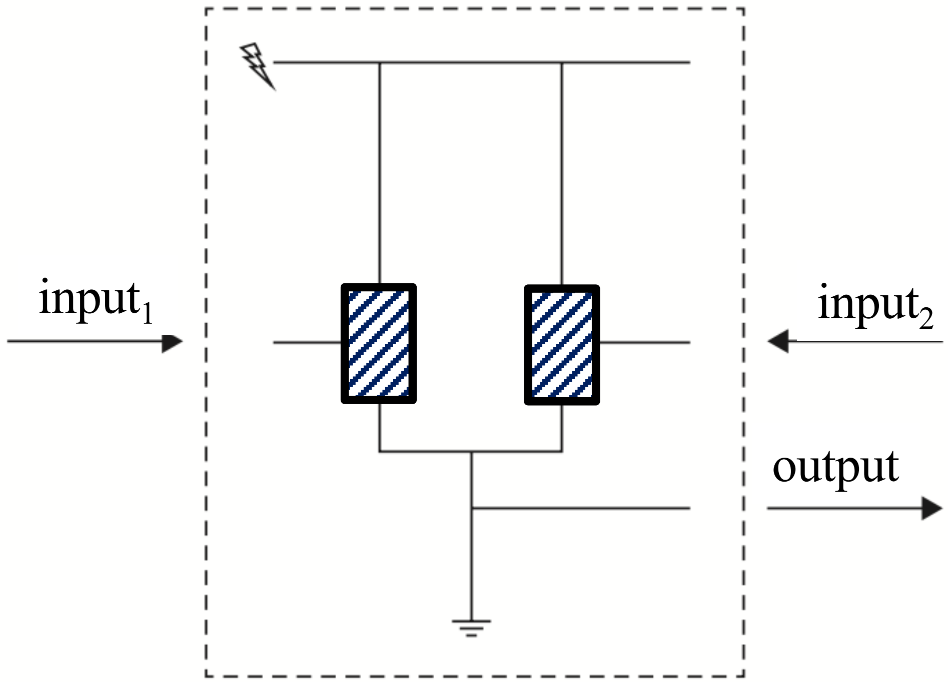


In this other system, the two transistors are said to be in a "parallel" configuration, because they are one next to the other (i.e. on parallel lines) between the high tension and the ground.

The tension in the output point is high in more cases here, because for the output point to be connected to the high tension we just need one transistor to be condictive. This means that only when both transistors are off the output is low tension.

In numerical terms, the output is 0 only in one case (input$_1$ = 0 and input$_2$ = 0)

input$_1$	input$_2$	output
0	0	0
0	1	1
1	0	1
1	1	1



In logical terms, the output is "false" only when both inputs are "false" and "true" in all other cases. We need one true input to make the output true as well.

$input_1$	$input_2$	output
false	false	false
false	true	true
true	false	true
true	true	true

The two inputs are put together in the same way we use the "OR" conjunction (also known as "disjunction"): "the Sun is cold OR one plus one is three" is false, whereas "water is a liquid OR the Earth if flat" is true.
This circuit may be seen as the electronic version of the OR logical operator.

This is how an "OR gate" is represented in the graphical standard for circuit design.