COMPUTER ARCHITECTURE AND SOFTWARE EXECUTION PROCESS

COMPUTER ARCHITECTURE: INTRODUCTION

🎓 Bachelor in Artificial Intelligence, Data and Management Sciences
🏛 CentraleSupelec and ESSEC Business School- 2023/2024

Idir AIT SADOUNE
idir.aitsadoune@centralesupelec.fr

OUTLINE

Back to the outline - Back to the begin

OUTLINE

Back to the outline - Back to the begin

IDIR AIT SADOUNE

- **PhD thesis in Computer Science** graduated from ENSMA in 2010.
  - PhD thesis about **formal modelling** and **verifying** Services compositions.
- **Associate-Professor** at the Computer Science Department of CentraleSupelec.
- **Researcher** at the **Model** and **Proof** Teams of LMF - Formal Methods Laboratory.

SYLLABUS

Chapter	Lecture	TD	Lab
Computer Architecture: Introduction	1h30		
Internal Architecture of Microprocessors	1h30	1h30	3h00
Data representation	1h30	1h30	
Operating systems	1h30		
Parallel programming & synchronization	1h30	1h30	
Memory management	1h30	1h30	

LEARNING OUTCOMES AND ASSESSMENT

Learning outcomes covered on the course

- Understanding how a computer system works.
- Knowing the role of an operating system in a computer system.
- Understanding how a computer system executes a program.
- Resolving management problems of concurrent processes sharing resources.
- Understanding how the memory works.
- ...

Assessment of learning outcomes

- The practical exam during the Lab (**30%**)
- The final exam (**70%**)

QUESTIONS ?

- During the lesson
- After the lesson
- Outside of the lesson
  - idir.aitsadoune@centralesupelec.fr
  - **MS TEAMS**

COMPUTER SCIENCE

- Computer science is the science of **automatic information processing**.
- Automatic processing of information is done with **programs** executed by **machines**.
  - **programs** (software) describe the treatment to be carried out,
  - **machines** (hardware) run **programs**.

THE CONCEPT OF COMPUTER

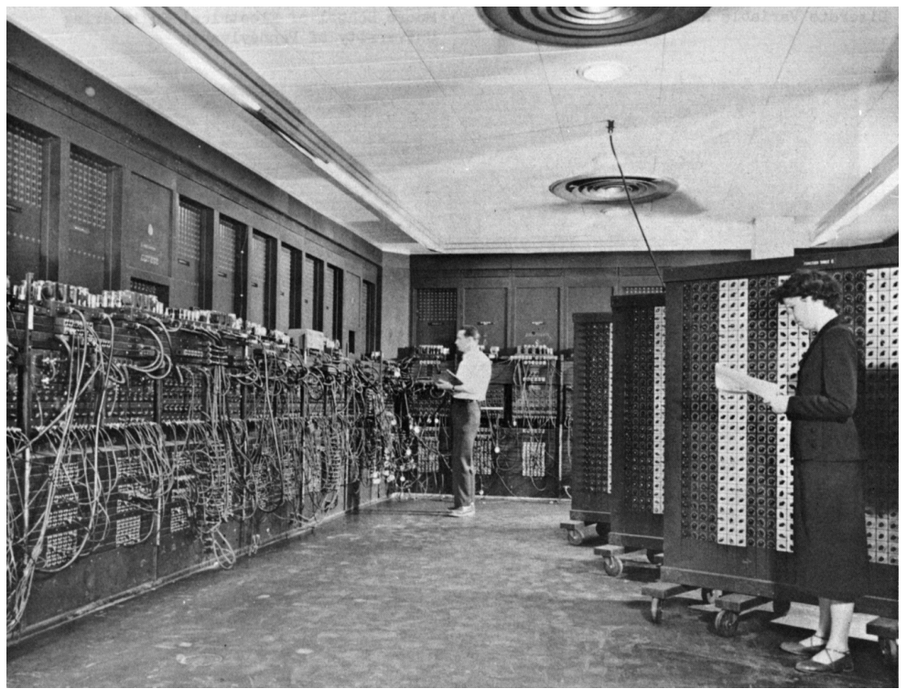
- Computer is a machine that can be programmed to carry out sequences of operations **automatically**.
  - we give him **program** (software)
  - we give him **data** (information)
  - it transforms the data

COMPUTER ARCHITECTURE ?

- **Architecture**
  - a general term to describe buildings and other physical structures
  - the art and **science** of designing structures
- **Computer**
  - a machine that can be programmed to carry out sequences of operations **automatically**.
    - ➠ personal computer (PC), computer on board an airplane, TV, martphone, ...
- **Computer Architecture**
  - a description of the structure of a computer system.

ENIAC - 1945

- was designed in 1945 by John Mauchley and John Eckert at the University of Pennsylvania.
- was the first programmable, **electronic**, general-purpose digital computer.
- was a large, modular computer with individual panels performing different arithmetic functions.



HP 3000 - 1972

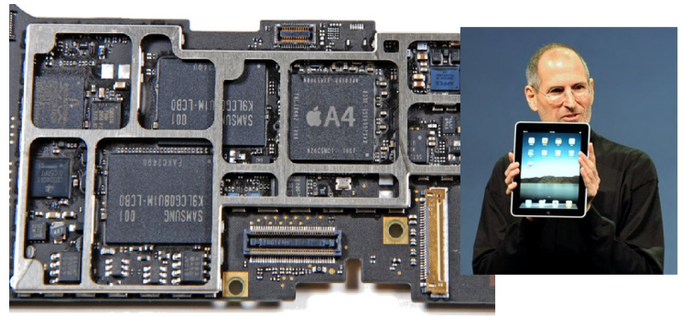
- was designed to be the first **minicomputer** with full support for time-sharing.
- first implemented with Transistor-transistor logic.
- integrating integrated circuits on a large scale led to the development of **microprocessors**.

APPLE II - 1977

- one of the first highly successful mass-produced **microcomputer** products.
- designed by Steve Wozniak, and launched in 1977 by **Apple**.

TODAY'S COMPUTERS

- System on a Chip (**SOC**) : a complete system embedded in a chip (**integrated circuits**).
- An **integrated circuits** can contain:
  - one or more microprocessors,
  - memory,
  - interface devices,
  - or any other component

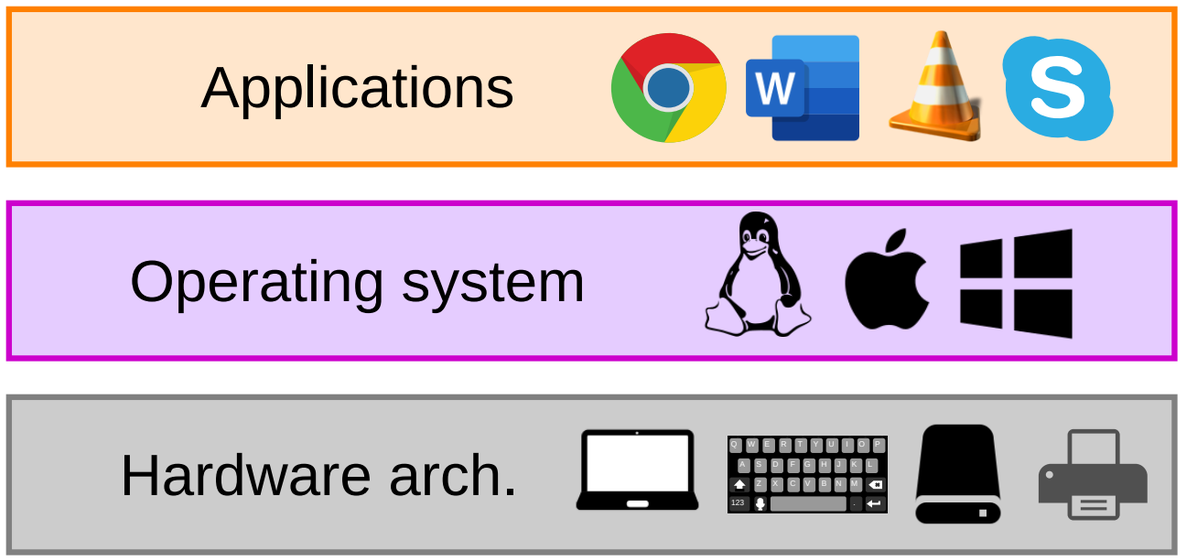


LAYERED ORGANIZATION

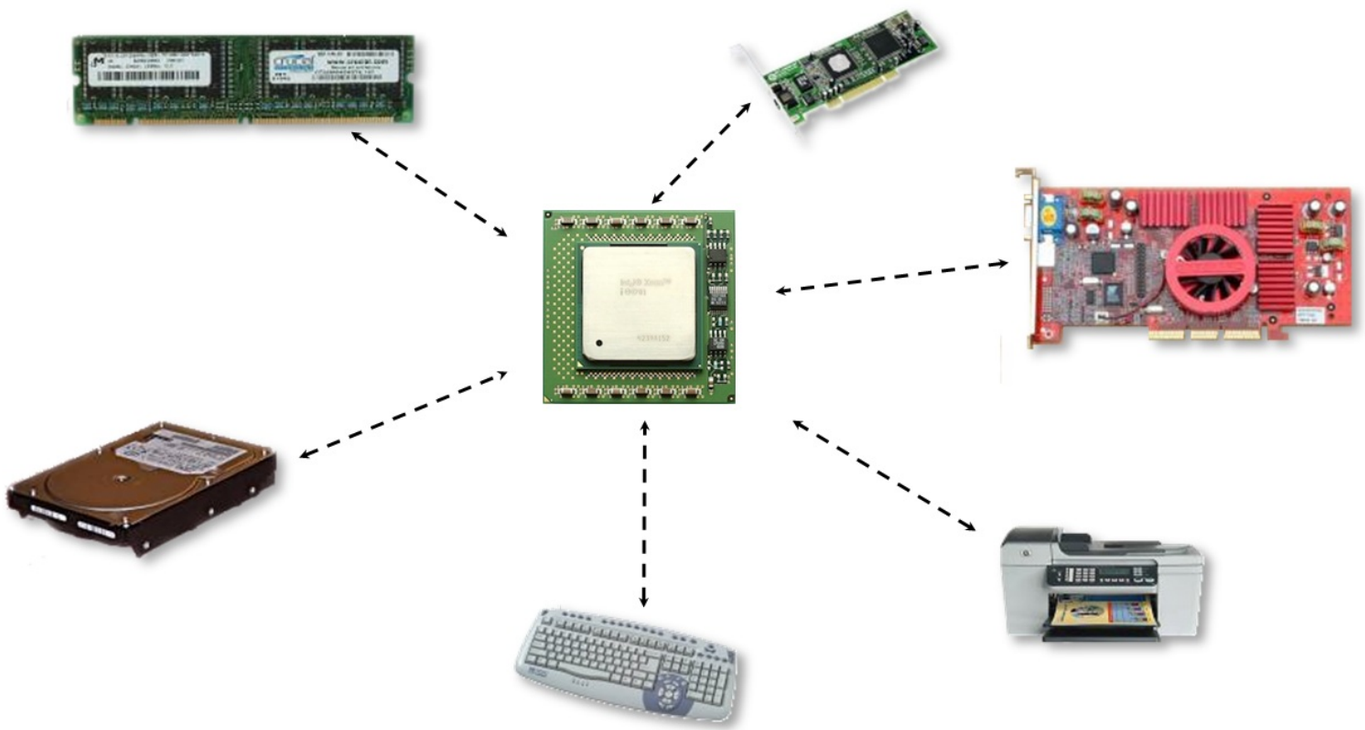
These IT systems are built on a **single model**:

- a hardware architecture
- a set of peripherals
- an operating system
- apps

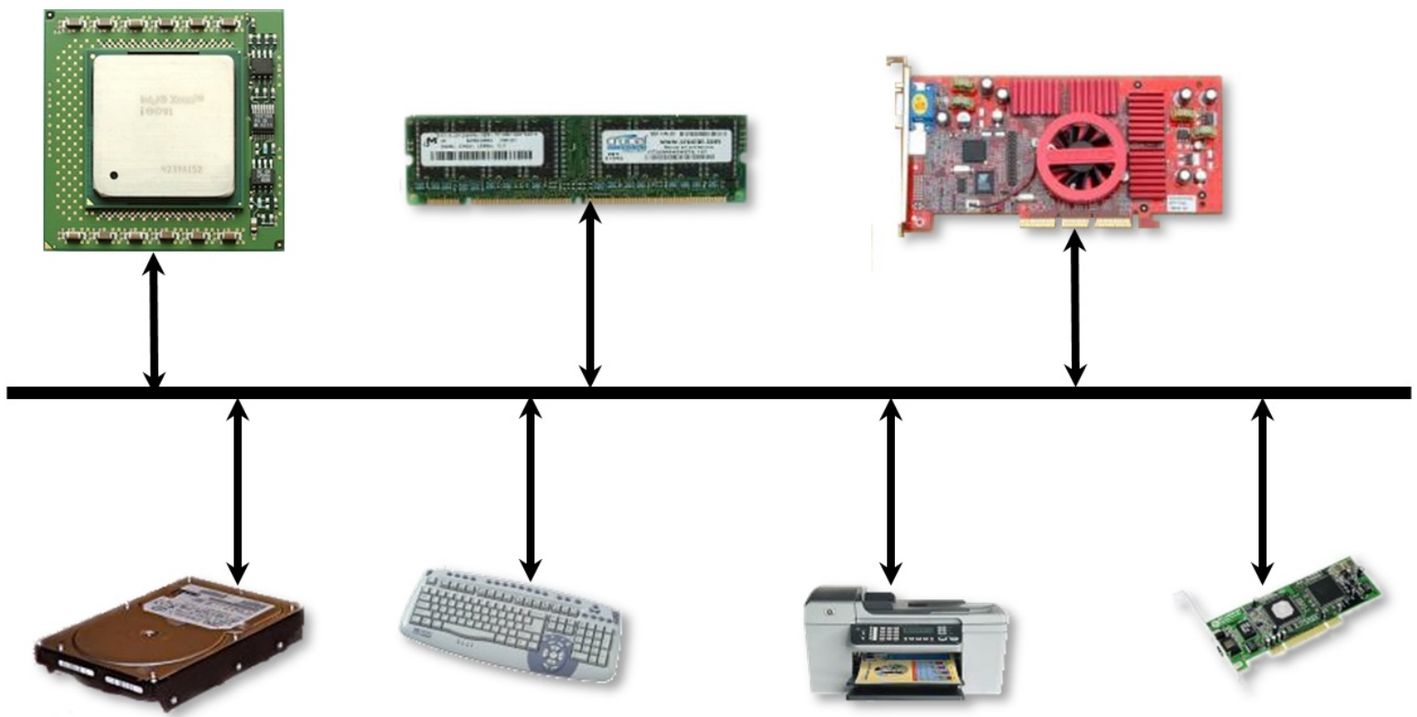
LAYERED ORGANIZATION



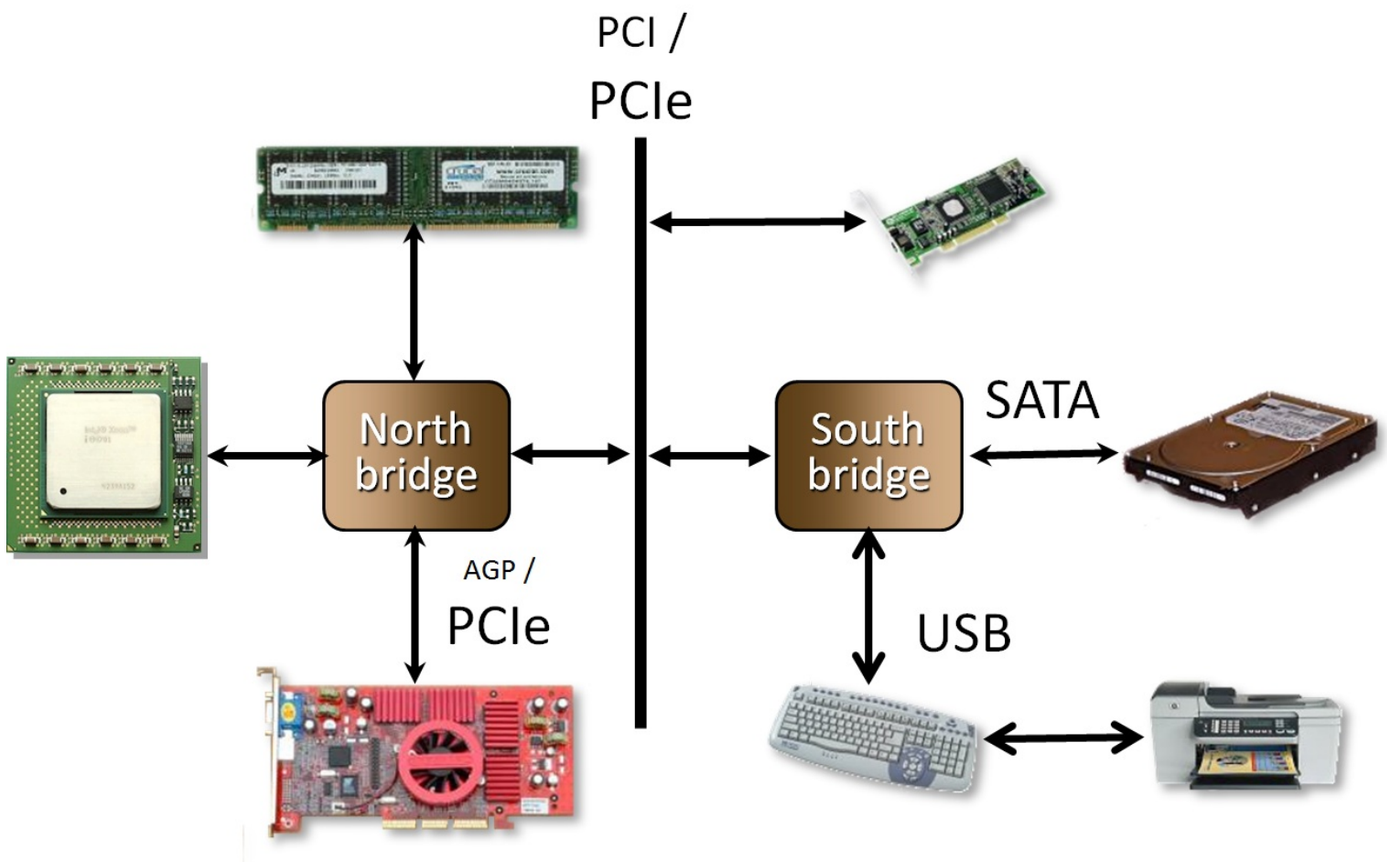
A HARDWARE ARCHITECTURE



A HARDWARE ARCHITECTURE



A HARDWARE ARCHITECTURE



OUTLINE

Back to the outline - Back to the begin

BINARY

- 2 electrical levels (0V, 5V) $\rightarrow$ 2 digits (0 and 1)
  - Base 2 (**binary system**)
- **Bit** = **B**inary digi**t**
- Useful powers of 2 ...
  - $2^0 = 1$
  - $2^1 = 2$
  - $2^2 = 4$
  - $2^3 = 8$
  - $2^4 = 16$
  - $2^5 = 32$
  - $2^6 = 64$
  - $2^7 = 128$
  - $2^8 = 256$
  - $2^9 = 512$
  - $2^{10} = 1024$
  - ...

HEXADECIMAL

- **Base 16** System
- Numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
- Numbers prefixed with 0x or $
- **binary** $\Longleftrightarrow$ **hexadecimal** immediate translation
  - $2^4 = 16^1$
  - 4 binary digits $\Longleftrightarrow$ 1 hexadecimal digit

		2810
1010	**1111**	**1010**
A	**F**	**A**

link to a converter

DATA MEASUREMENT UNIT

- The **byte** is the unit of measurement for data
  - 1 byte = 8 bits (binary digits)
- Multiples of the **byte**
  - 1 Kilobyte (KB) = 2^{10} bytes = 1 024 bytes $\approx 10^3$ bytes
  - 1 Megabyte (MB) = 2^{20} bytes = 1 024 KB = 1 048 576 bytes $\approx 10^6$ bytes
  - 1 Gigabyte (GB) = 2^{30} bytes = 1 024 MB = 1 073 741 824 bytes $\approx 10^9$ bytes
  - 1 Terabyte (TB) = 2^{40} bytes = 1 024 GB = 1 099 511 627 776 bytes $\approx 10^{12}$ bytes

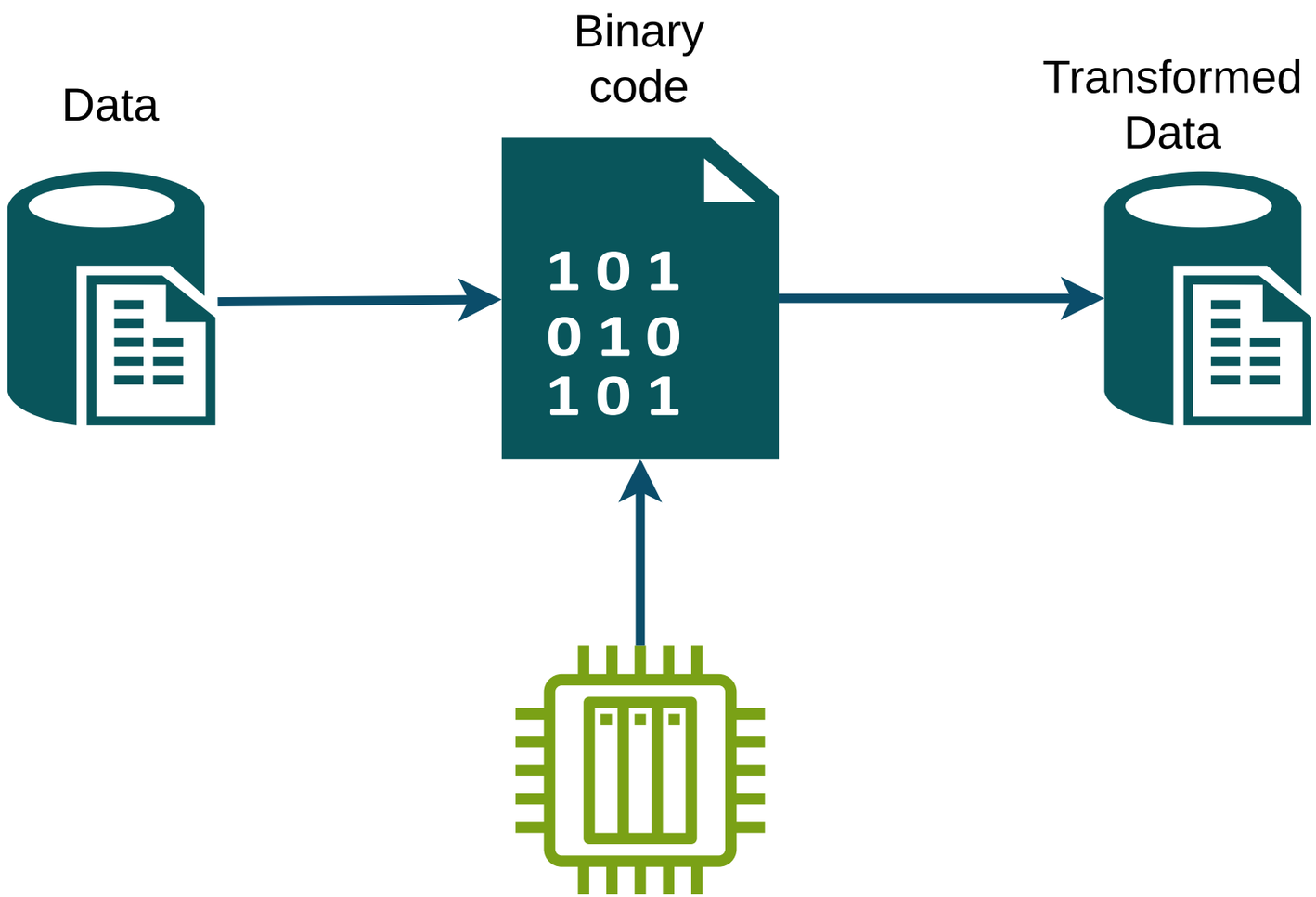
OUTLINE

Back to the outline - Back to the begin

MACHINE LANGUAGE

- What can a computer do?
  - copy values between storage units
  - perform logical/arithmetic operations between stored values
  - move within the program (jump), possibly conditionally
- A computer executes a very low level language (**Machine Language**)
  - means of controlling and elementary control of computer electronics
  - the instructions of this language carry out elementary operations
- **Machine language** coding must be simple to decode and optimized
  - this coding is not textual; it is done at bit level (**binary code**)
  - in practice, we represent these codes in hexadecimal or in textual form

AN EXAMPLE



ASSEMBLY LANGUAGE

- Unreasonable to want to write/read binary machine language directly
- Definition of an equivalent **symbolic representation**
  - textual mnemonics for operations
  - textual writing of entire programs
- **Assembly language**
  - very close to the machine, very low level
  - **Assembler**: translator from assembly language to machine language

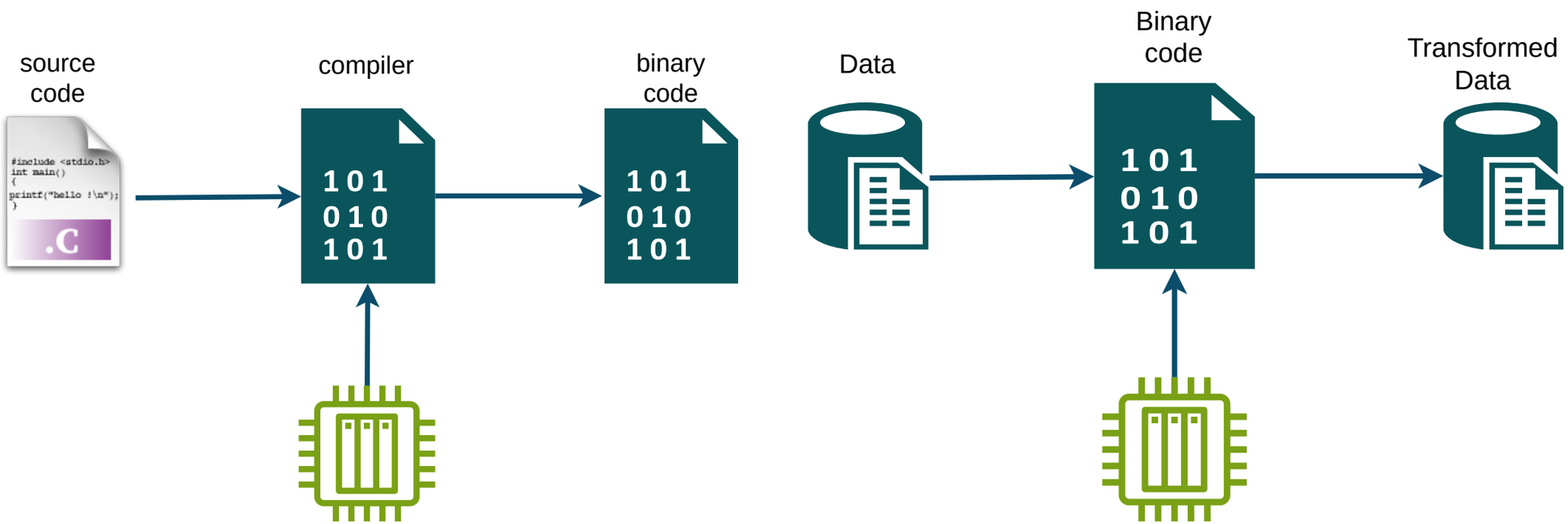
```
1    mov eax, a
2    mov ebx, 2
3    add ecx, eax, ebx
4    mov a, ecx
5    ...
6  @a: memval 3
```

```
1 2B50: 12AE 2B1E
2 2B52: 12AF #0002
3 2B54: 13BE
4 2B55: 12BD 2B1E
5 ...
6 2B1E: 0003
```

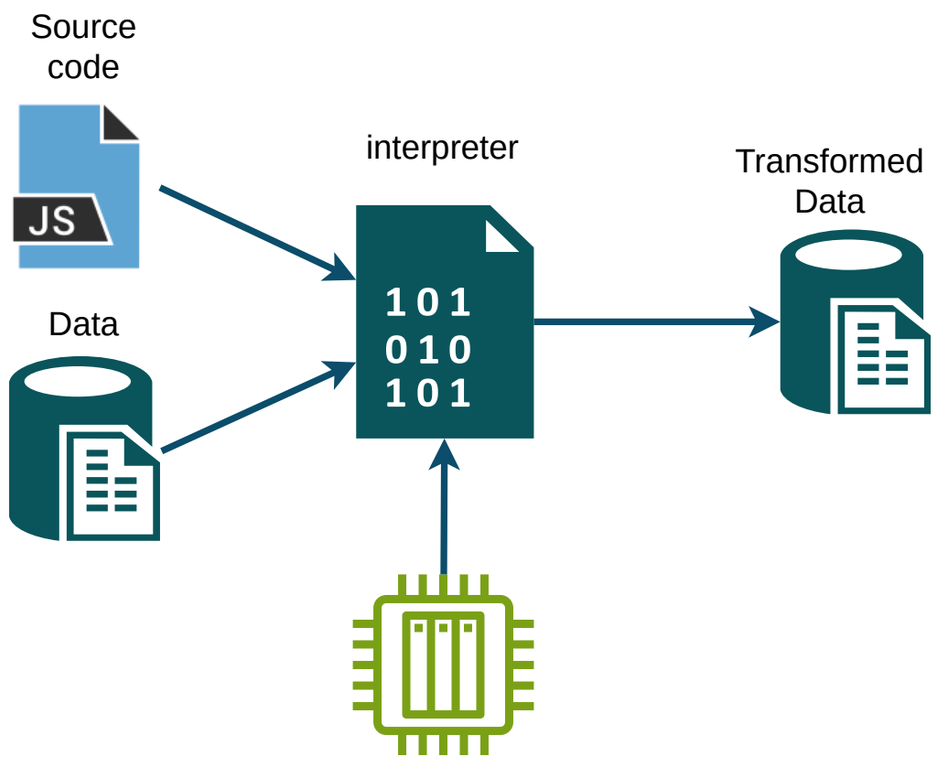
HIGH LEVEL PROGRAMMING LANGUAGE

- Most software is written in a **high-level programming language**.
  - **example:** **Python**, **Java**, **C**, **C++**, **C#**, **Ruby** ...
- How is a program written in a high-level programming language executed?
  1. **compilation**
  2. **interpretation**

THE COMPILATION PROCESS



THE INTERPRETATION PROCESS



THANK YOU

Back to the begin - Back to the outline