



POLITECNICO
MILANO 1863

DIPARTIMENTO DI ELETTRONICA
INFORMAZIONE E BIOINGEGNERIA

Advanced Computer Architectures

Course Introduction

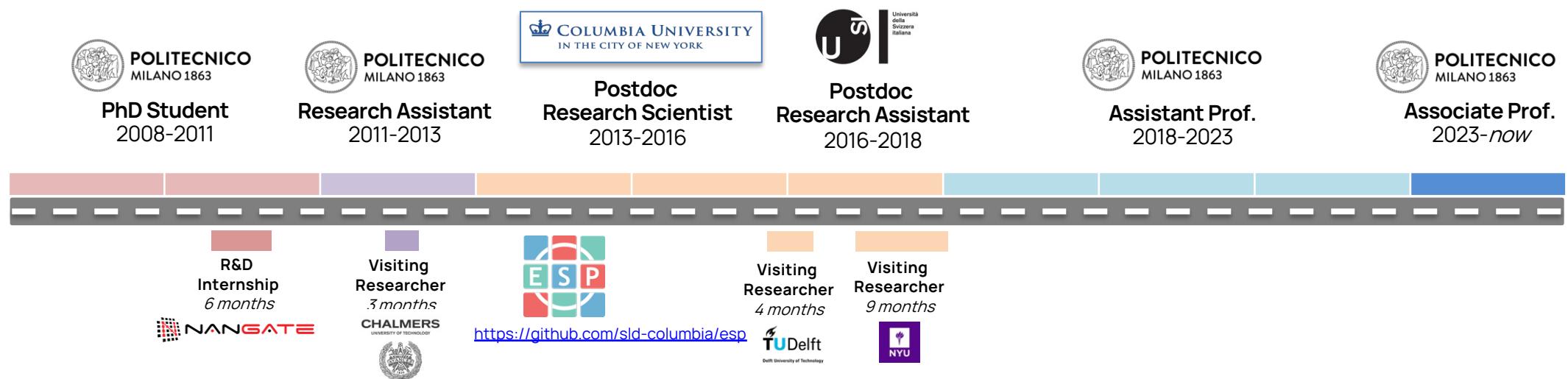
A.Y. 2024/2025 | Christian Pilato (christian.pilato@polimi.it)

About me

Associate Professor - DEIB

Office: DEIB (Building 20 - 1st floor)

Website: <http://pilato.faculty.polimi.it>



R&D Projects

FP6 HARTES

FP7 FASTER

DARPA PERFECT

H2020 CERBERO

H2020 EVEREST

FP7 SYNAPTIC

AMBRA

My research topics

Research focus on

- Accelerators and heterogeneous SoCs
- High-level synthesis
- FPGA prototyping
- Memory design and optimization
- Hardware and hardware-assisted security
- LLM fine-tuning and other AI methods for assisting augmentative and alternative communication



International collaborations (incl. visits and joint projects/theses)



(and more...)

Course objectives

- Have each student to familiarize with computer architectures
 - SoC, Multicore systems
 - Heterogeneous architectures
 - GPU, FPGA
 - Programming models
 - Technology factors
 - Performance, Costs
 - Design of computer architectures
 - etc..
- Envision where/how/why to effectively **use computer architectures in research**

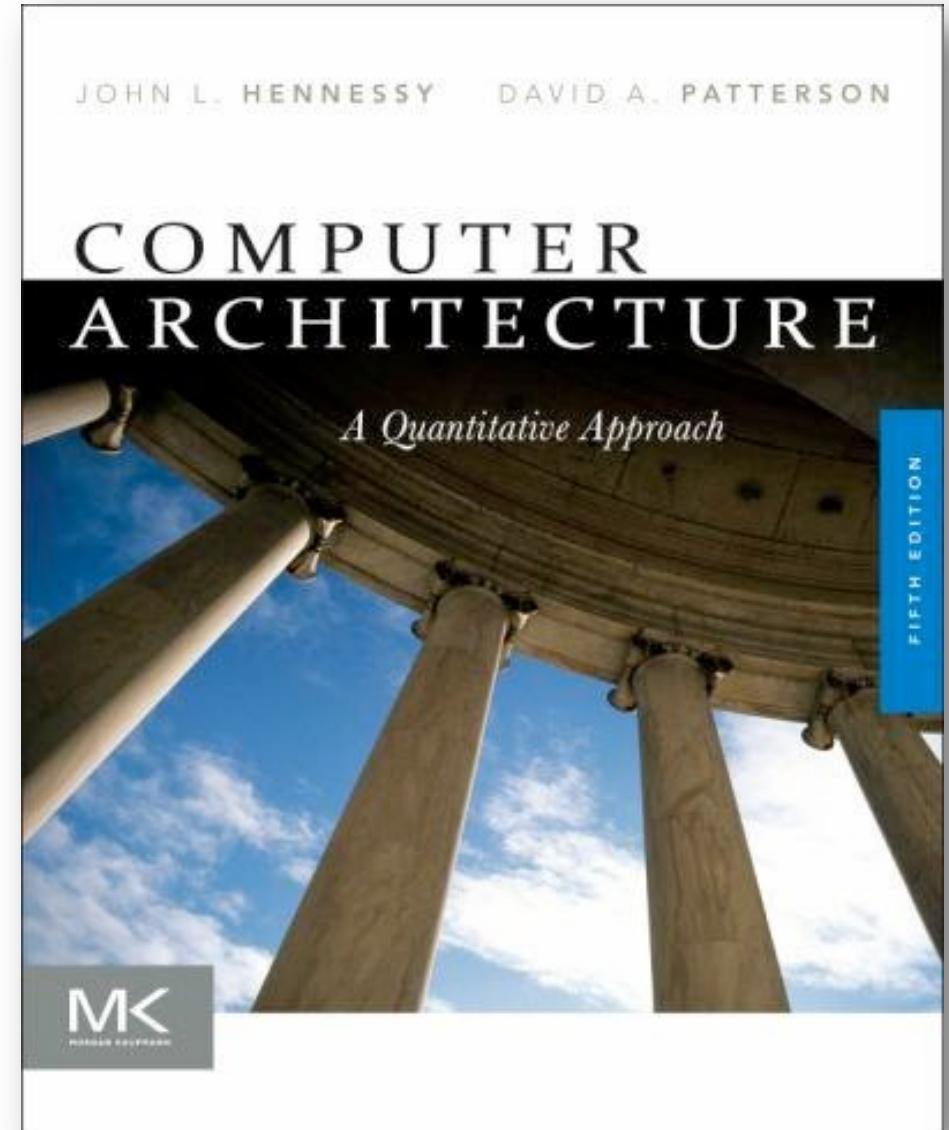
Topics

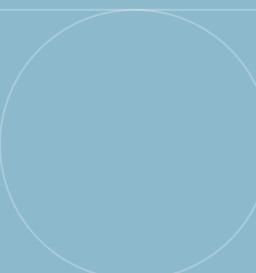
- Measuring performance: What are the driving measures?
 - Performance (area, time, frequency.....), Power, Cost
- Internal Parallelism in processors:
 - Pipelining
 - Instruction level parallelism inside processors
- Memory
- Going beyond ILP
- Multiprocessors and multicore systems: taxonomy, topologies, communication management, memory management, cache coherency protocols, example of architectures
- Heterogeneous architectures: Vector processors; Graphic Processors, GPGPUs, FPGAs



Topic coverage and (online) material

- Textbook: Hennessy and Patterson, Computer Architecture: A Quantitative Approach
- ACA website: WeBeep
 - Slides
 - Videos
 - Calendar



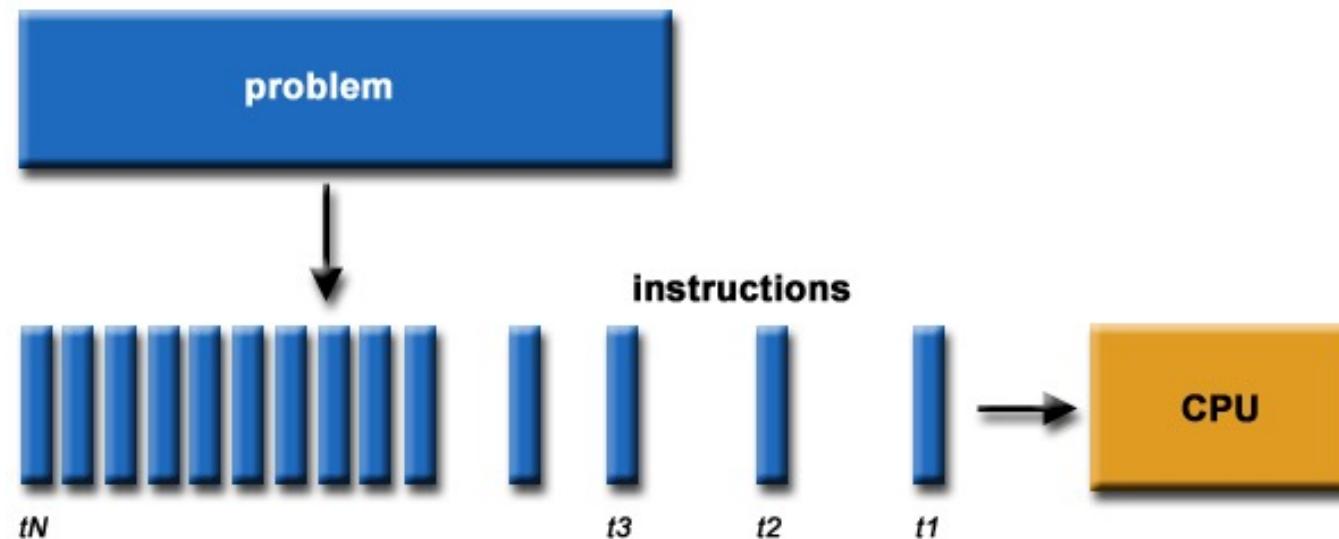


What are we talking about?

"Traditional" computation

Software is written for **serial computation**

- It has to be executed on a single computer having a single **Central Processing Unit (CPU)**
- A problem is broken into a discrete series of **instructions**
 - Instructions are executed one after another
- **Only one instruction may execute at any moment in time**



But the world is "parallel"

Events are happening simultaneously

Many complex, interrelated events happening at the same time, yet within a sequence

Some examples:

Galaxy formation

Planetary movement

Tectonic plate drift

Rush hour traffic

Automobile assembly line

Building a space shuttle

Ordering a hamburger at the drive-through



Beyond traditional computation – Flynn taxonomy (1966)

SISD - Single Instruction Single Data

Uniprocessor systems

MISD - Multiple Instruction Single Data

[multiple functions on the same data](#)

No practical configuration or commercial systems

SIMD - Single Instruction Multiple Data

[functionality is replicated on different data](#)

Simple programming model, low overhead, flexibility, custom integrated circuits

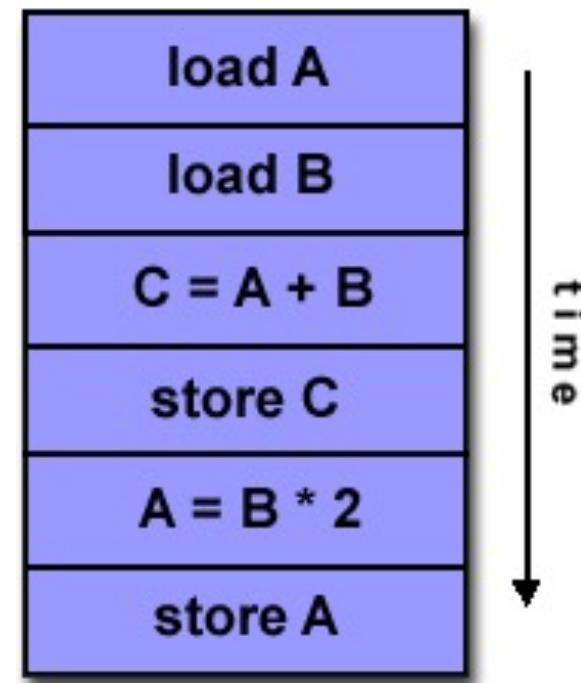
[es: video processing, image is splitted in blocks and on each block the functionality is applied](#)

MIMD - Multiple Instruction Multiple Data

Scalable, fault-tolerant, *off-the-shelf* micros

SISD

- A serial (non-parallel) computer
- **Single instruction**: only one instruction stream is being acted on by the CPU during any one clock cycle
- **Single data**: only one data stream is being used as input during any one clock cycle
- Deterministic execution
- This is the oldest and, even today, the most common type of computer



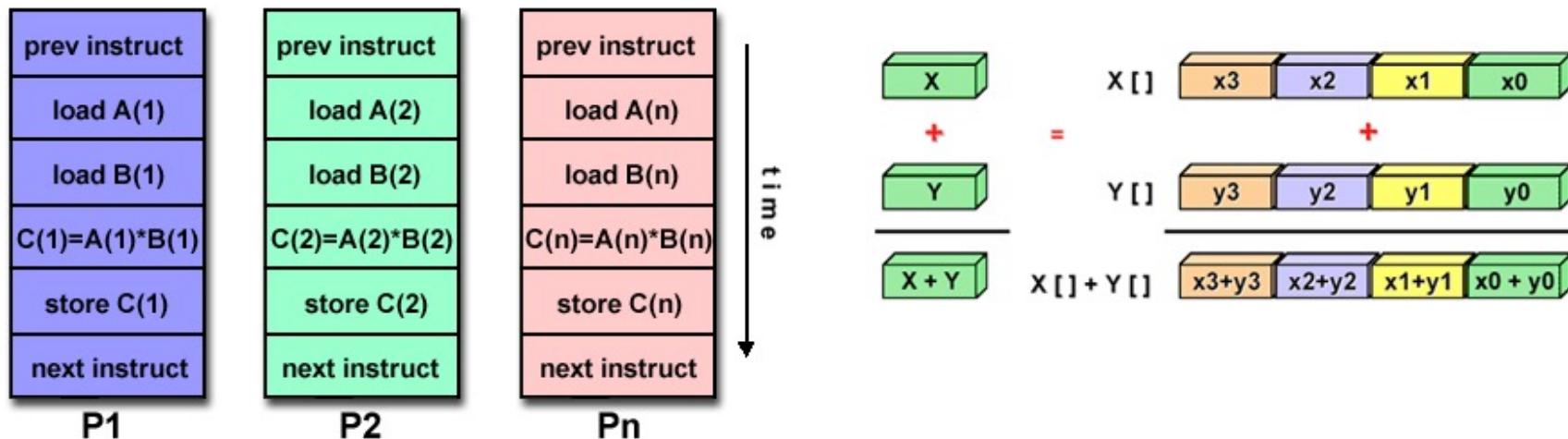
Parallelism? Which kind?

Data-level parallelism (DLP)



SIMD

- A type of parallel computer
- **Single instruction**: all processing units execute the same instruction in any clock cycle
- **Multiple data**: each processing unit can operate on a different data element



Best suited for specialized problems characterized by a **high degree of regularity**, such as graphics/image processing

Parallelism? Which kind?

Data-level parallelism (DLP)
Thread-level parallelism (TLP)
Request-level parallelism (RLP)



Hardware parallelism

Instruction-Level Parallelism

Exploits data-level parallelism at modest level (through compiler techniques such as pipelining) and medium levels (using speculation)

Vector architectures and Graphic Processor Units

Exploit data-level parallelism by applying a single instruction to a collection of data in parallel

Thread-level parallelism

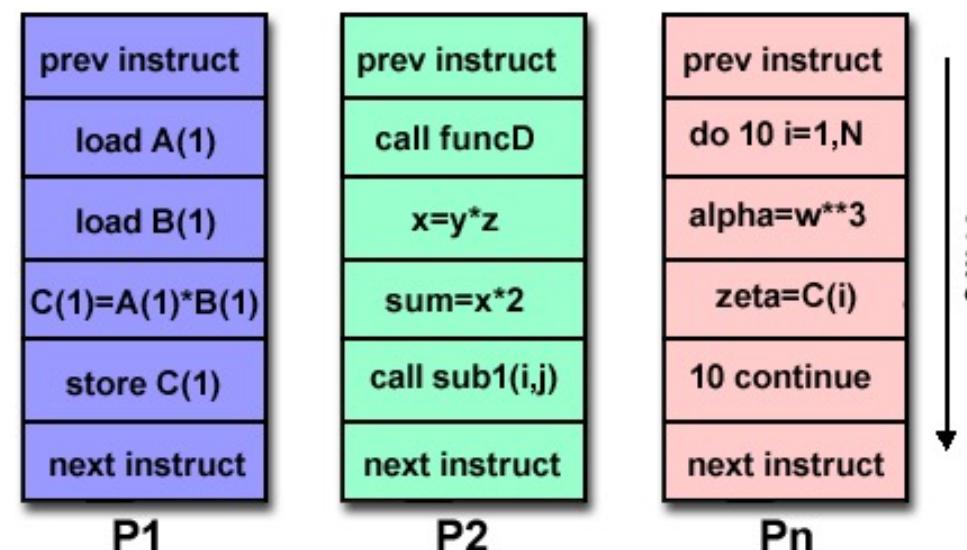
Exploits either data-level parallelism or task-level parallelism in a tightly coupled hw model that allows interaction among threads

Request-level parallelism

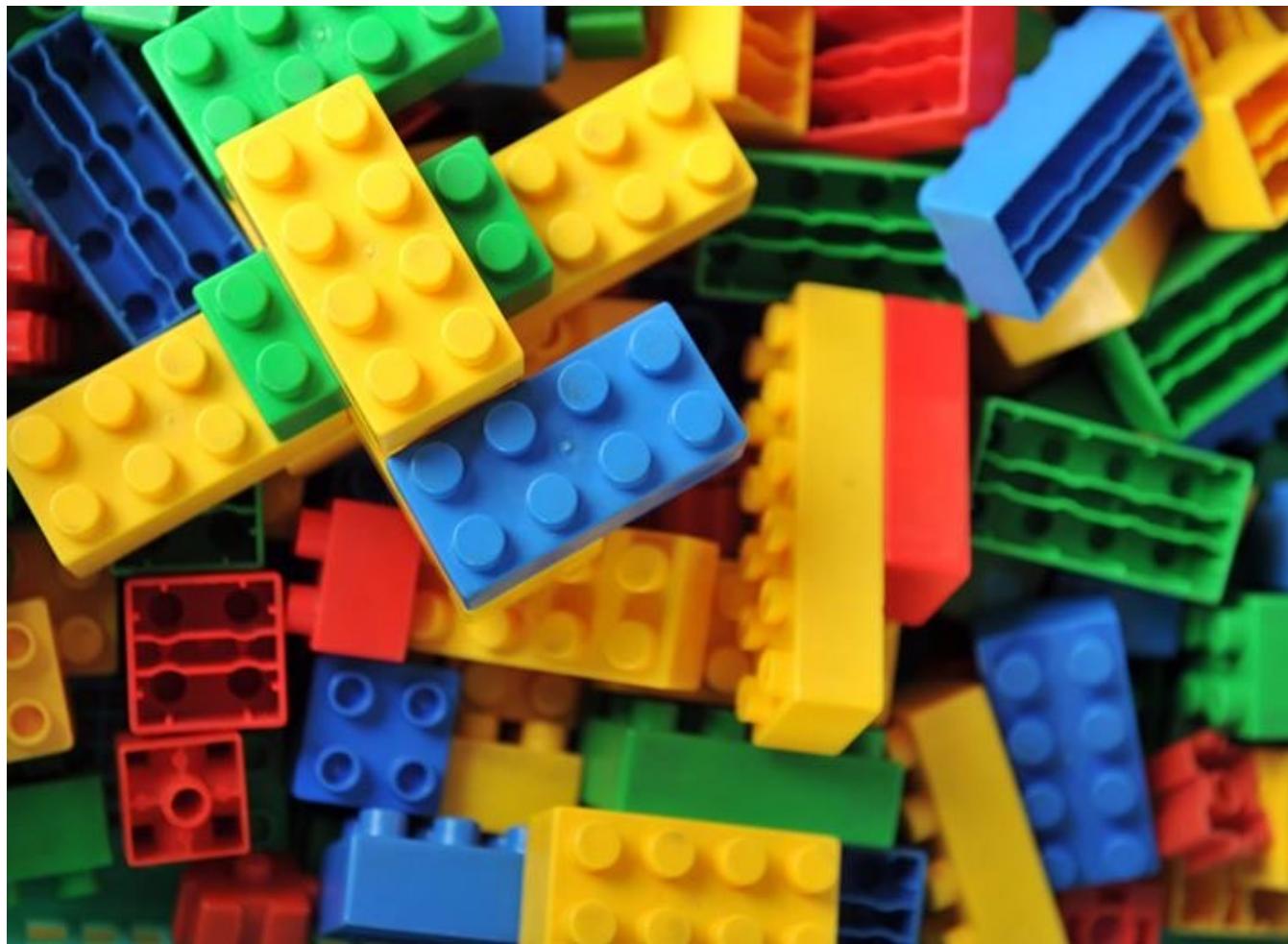
Exploits parallelism among largely decoupled tasks specified by the programmer or the OS

MIMD

- Nowadays, the most common type of **parallel computer**
- Multiple Instruction**: every processor may be executing a different instruction stream
- Multiple Data**: every processor may be working with a different data stream
- Execution can be synchronous or asynchronous, deterministic or non-deterministic



Today... Adaptation and heterogeneity are everywhere...



Heterogeneous system architecture

They all have to deal with **energy and power consumption**



More information available at: <http://hsafoundation.com/>

Class Organization

ACA course organization

- Feb 17- May 31, 2025
 - mix of Theory & Practice [T&P]:
 - Foundational classes (basic ACA101 topics)
 - Theoretical classes (advanced topics)
 - TA hours (exercises, tests)
- Complete calendar will be online **by this week**
- Two options to pass the course: **classic and project-based**

Classic written exam

The final exam consists of a **90-min written exam**

For each written exam (6 problems), a max score of **33 pts (equal to 30L)**

- Exercises part [**Practice**]
 - 3 exercises
- Theoretical part [**Theory and advanced topics**]
 - 3 open questions

To pass the exam:

- at least 18 pts overall
- at least 50% of points in each part (exercises and theory)

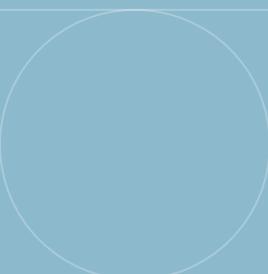
Project-based exam

Max score of **33 pts (equal to 30L)**

- A – one **written test** in the first half of May: **max 15 pts**
ci può essere un altro appello per la parte A nell'appello di luglio
- B – **Project**: **max 18 pts**
 - Team up to 2 students
 - Starting: 28.2.25 - Deadline: 1.7.25
 - Advanced topics, high scientific value (relevant research topic, publication/thesis, ...)

To pass the exam:

- at least 18 pts overall
- at least 8 pts from A (written test)
- at least 10 pts from B (project)



Questions?