Championship of Branch Prediction (CBP - ISCA 2025)

This year, the 6th edition of the international competition of branch prediction is being held!

In this project, **you will design a branch predictor** as per the competition's rules (max 192KB of data structures).

You will be given a C++ simulator and will implement a "predictor" class, no worries about the language, some fluency with C will get you far enough. The winning predictor from the previous competition is also provided as a reference point.

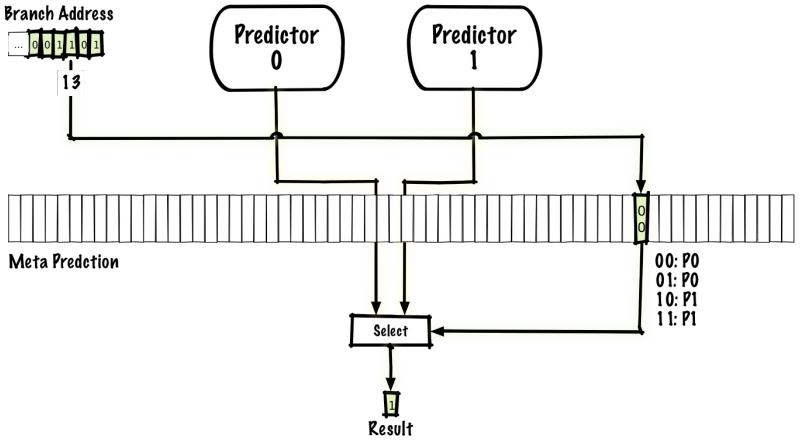
Then, test your predictor on the official set of traces (recorded sequences of instructions) and measure its accuracy.

Iterate your design until you are satisfied with the results!

And let's be ambitious: if you manage to beat the previous competition's winner before the beginning of May, we will help you submit your design to ISCA 2025 (that is held in Tokyo 😉)!

Official website and simulator:

- https://ericrotenberg.wordpress.ncsu.edu/cbp2025/
- https://github.com/ramisheikh/cbp2025?tab=readme-ov-file



Supervisors:

- Ronzani Marco – PhD – marco.ronzani@polimi.it
- Tommaso Spagnolo – PhD – tommaso.spagnolo@polimi.it

Championship of Branch Prediction (CBP - ISCA 2025)

This project is spot-on for the ACA course! Hence, we will provide you some futher help:

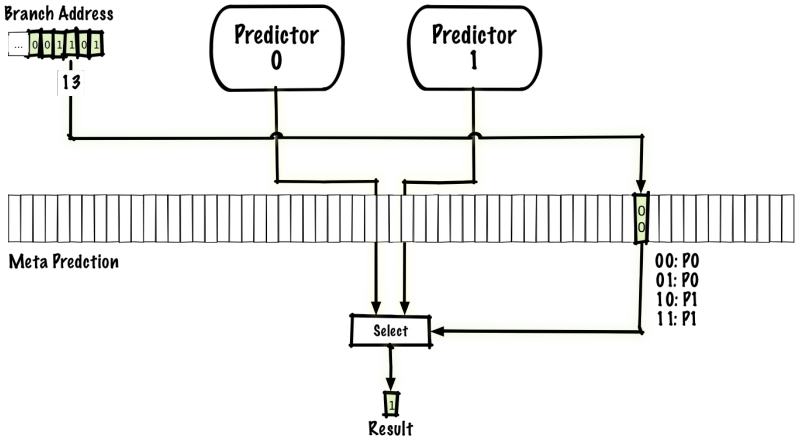
For everyone interested in this project, an optional 2h lecture will be held on **advanced topics in branch prediction**!

You will be implementing some predictors too, with the goal to beat on some randomly chosen traces from the competition in three steps:

1. Beat a backward-taken-forward-not-taken static predictor;
2. Beat a two-bit Smith predictor;
3. Beat a Yeh and Patt two-level PA predictor.

Curious about how the PA predictor works? Check this (old) article out:

https://websrv.cecs.uci.edu/~papers/mpr/MPR/ARTICLES/090405.pdf



Supervisors:

- Ronzani Marco – PhD – marco.ronzani@polimi.it
- Tommaso Spagnolo – PhD – tommaso.spagnolo@polimi.it

Championship of Branch Prediction (CBP - ISCA 2025)

Sure, the international competition is great, but your group will also **compete against each others**:

We will be ranking all your predictors by their accuracy on some random traces from the competition!

Thus, remember:

Everything's fair, so long as an LLM didn't write (most of) your code!

But no copying between groups, since this is **a competition!**

To deliver the project you will need to **provide us your code**, alongside a brief **report** detailing how you reached your final implementation and any design decisions taken along the way.

At last, here is a little tip: make your data structures' size parametric. They will need to fit in 192KB, but as long as that holds, you are free to **explore** how to divide that between them!

Supervisors:

- Ronzani Marco – PhD – marco.ronzani@polimi.it
- Tommaso Spagnolo – PhD – tommaso.spagnolo@polimi.it

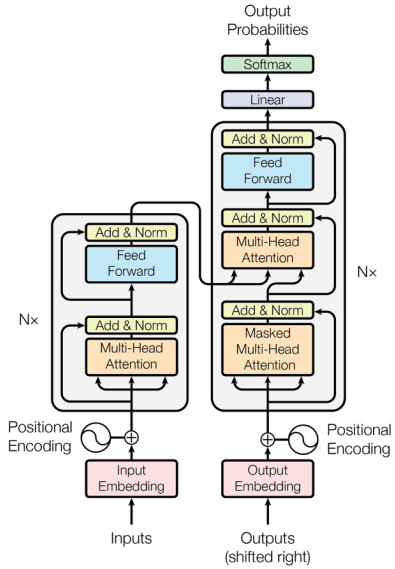
Demonstration of a Simple Transformer Running on the NPU of an STM32N6

Running modern AI on edge devices is challenging!

You need to manage limited resources, dedicated hardware accelerators, and custom toolchains….but it’s worth it!

For instance, an STM32N6 can run an object detection network in less than 1W [1].

Goal: realize a functioning demonstration of Transformer running on the NPU of the latest STM32N6 MCU.



Tools:

- SMT32Cube.AI
- STM32AI Model Zoo
- STM32N6570-DK board

DEPLOY

Supervisor: Ronzani Marco – PhD – marco.ronzani@polimi.it

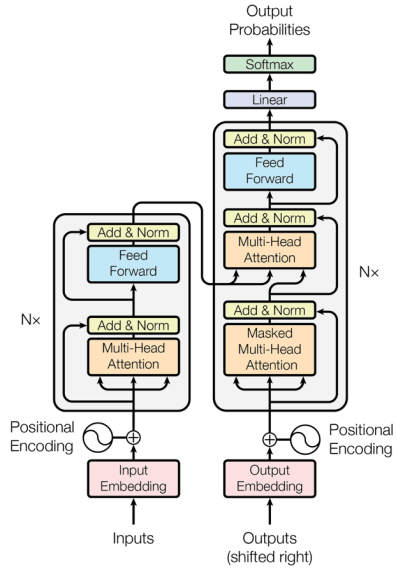
[1]: https://wiki.stmicroelectronics.cn/stm32mcu/wiki/AI:STM32Cube.AI_model_performances

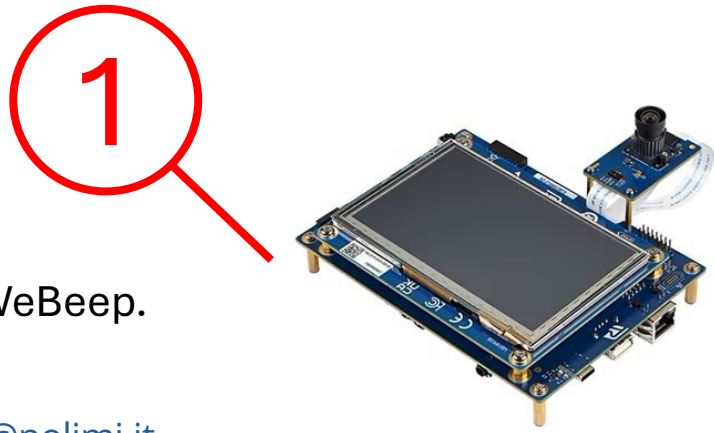
Demonstration of a Simple Transformer Running on the NPU of an STM32N6

WARNING: we only have **one board**, that you will be provided access to under our supervision.

Hence, since you will have limited time with the board, make sure to test your code, thoughtfully read the official documentation, and be mentally prepared for a lot of debugging 😊 !

NOTE: this project is more indicated if you plan to continue it with the **Multidisciplinary Project course (5 CFU).**





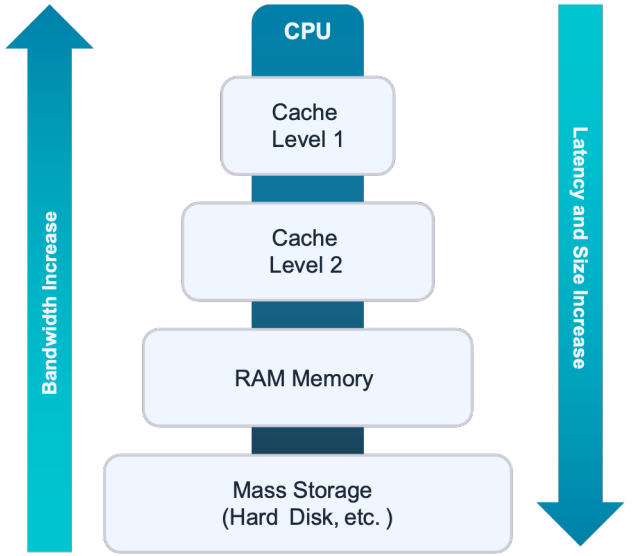
For more details, see the project's pitch on WeBeep.

Supervisor: Ronzani Marco – PhD – marco.ronzani@polimi.it

Exploring Cache Behavior with a Timing Simulator

Caches rule performance in modern processors.

Your job? **Make them work better.**



What you'll do:

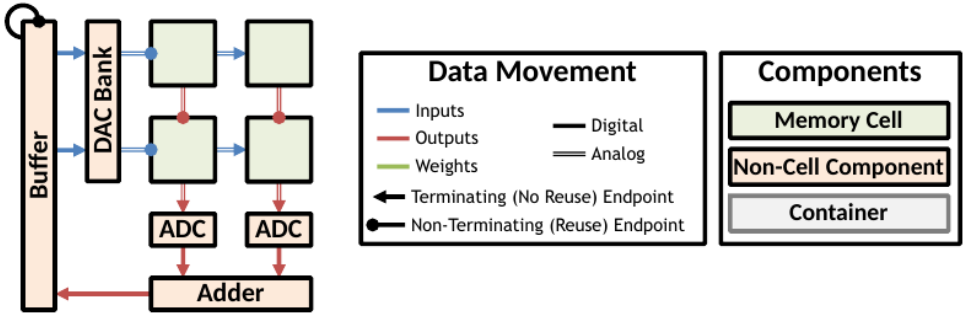
- **Build a cache simulator** in C for a MIPS / RISC-V processor.
- **Implement** a **4-way instruction cache** (8KB) and an **8-way data cache** (64KB).
- **Simulate cache misses and pipeline stalls.**
- **Test different replacement policies** (we're starting with Least Recently Used, but what else works?).
- **Run experiments**—cache size, block size, associativity, policies—see what actually improves performance.

Supervisor: Tommaso Spagnolo – PhD – tommaso.spagnolo@polimi.it

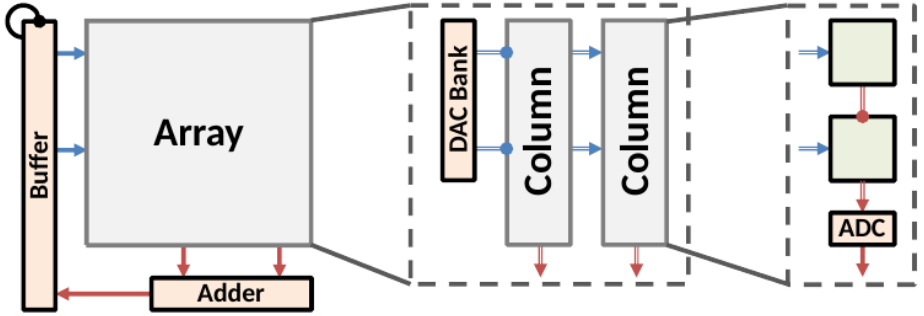
In-memory computing exploration

What's an In-Memory Computing architecture?

It's a solution to accelerate DNNs by addressing the data movement energy and computations required directly inside memory.



Main Objectives: explore, analyze and optimize the energy consumption and efficiency of a given In-Memory Computing architecture



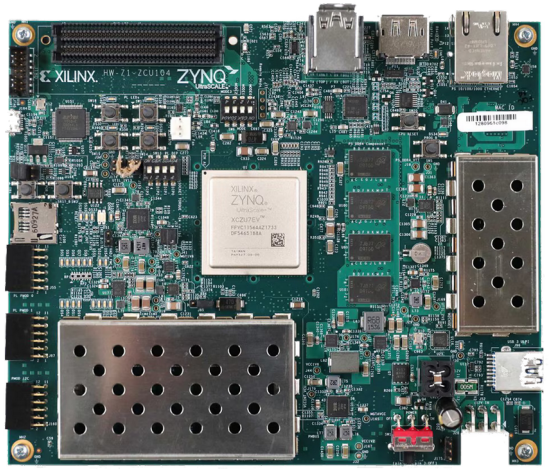
Tools: CiMLoop

Supervisor: Valeria de Gennaro, valeria.degennaro@mail.polimi.it

ACA Project proposal

Goals:

- Integrate existing accelerators into a complete system
- Verify performance against what is published
- Assess development time with different methods



Keywords:
FPGA, simulation, system design

Contact: serena.curzel@polimi.it

ACA Project proposal

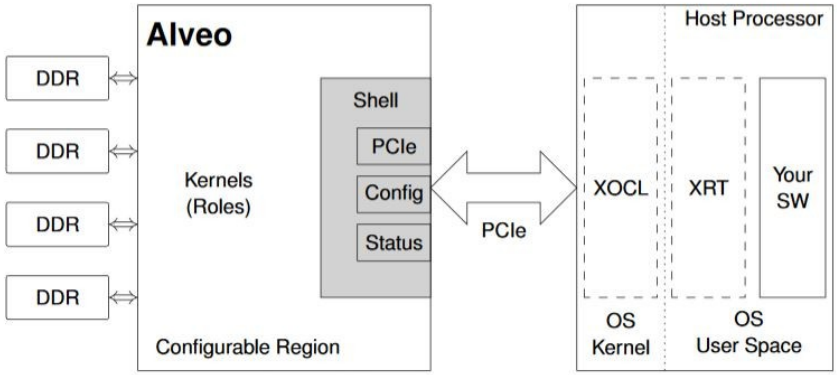
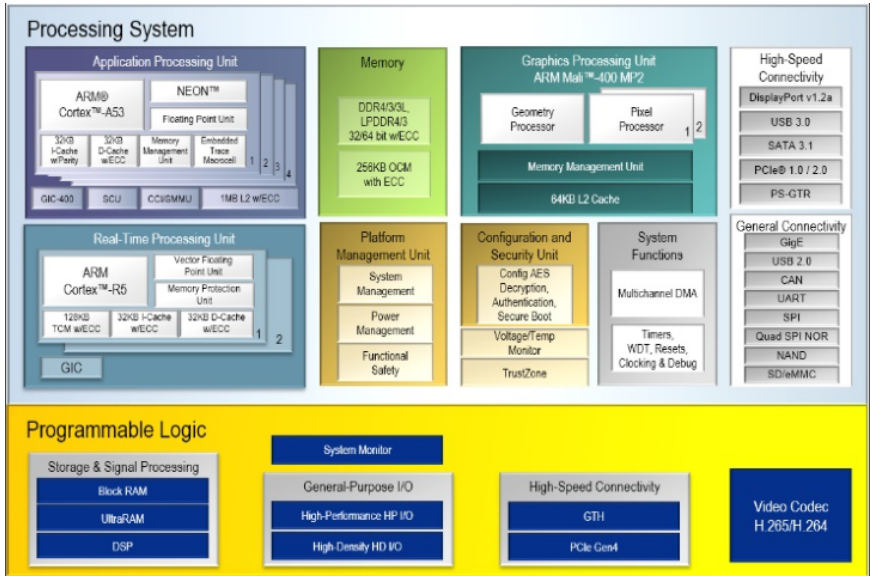


Figure 2.1: Alveo Conceptual Topology



Keywords:
FPGA, simulation, system design

The RTL description of the accelerator is provided
(some Verilog knowledge might help, but is not necessary)

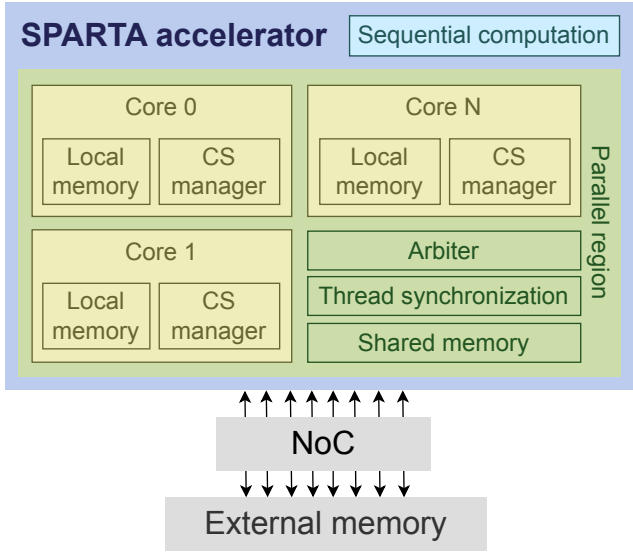
Build connections to the host and to memory

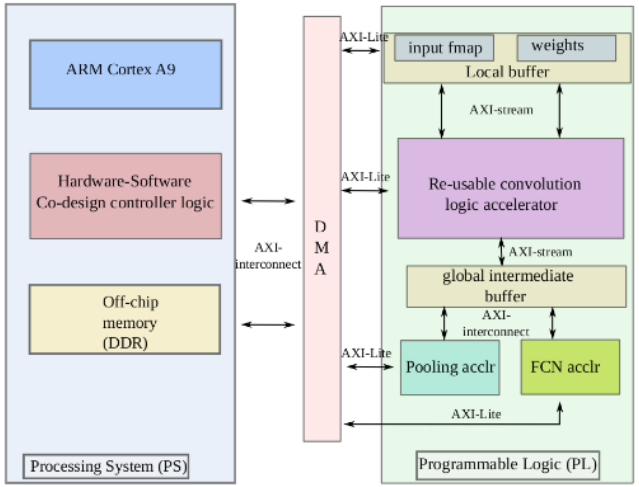
Contact: serena.curzel@polimi.it

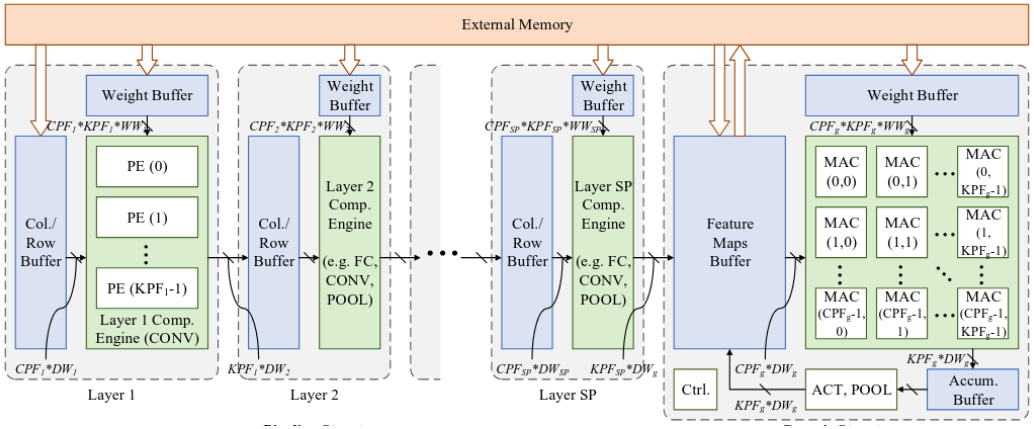
ACA Project proposal

Multiple projects available:

- Graph processing accelerators
- DNN accelerators
- …







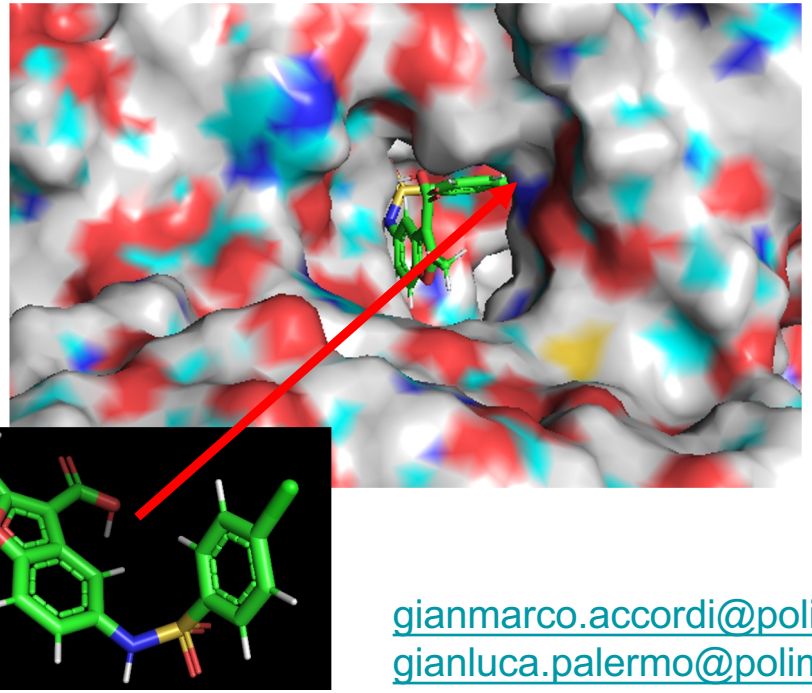
Keywords:
FPGA, simulation, system design

Contact: serena.curzel@polimi.it

Porting of a Kernel of a Molecular Docking mini-app

Porting of computation kernel of a virtual screening mini-application using parallel frameworks for heterogeneous hardware.

(https://github.com/elvispolimi/muDock)

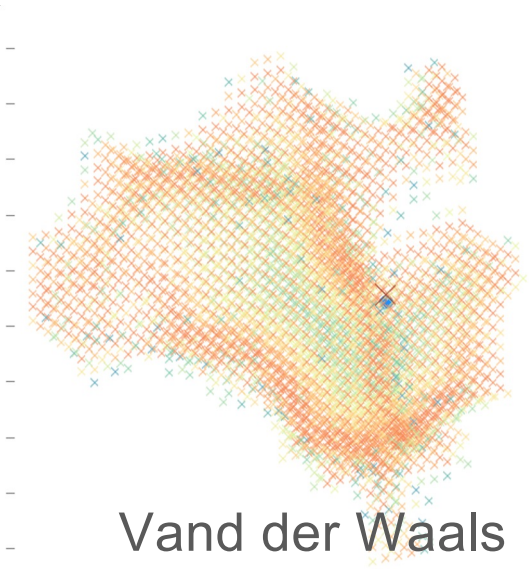


gianmarco.accordi@polimi.it
gianluca.palermo@polimi.it

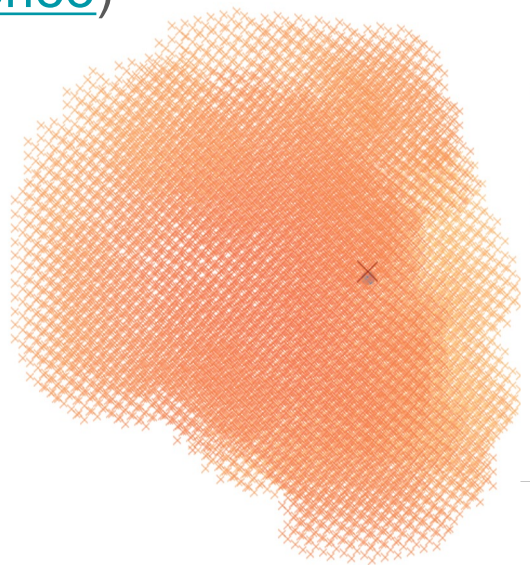
Analysis of a multi-layered map access on GPUs

Design and Implement an efficient data structure
to access a multi-layered map on GPUs.

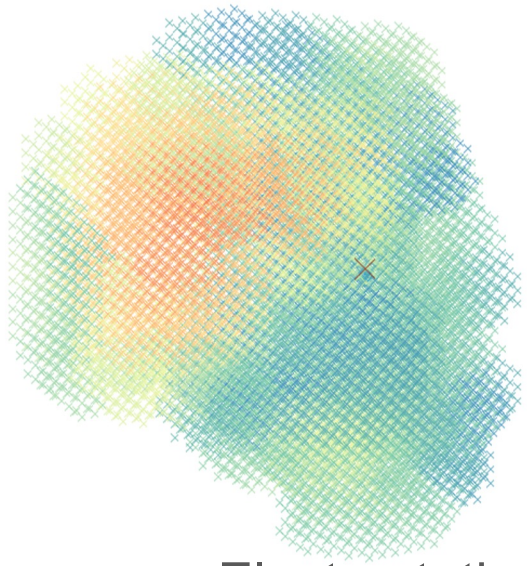
(https://elvispolimi.github.io/projects/geneo)



Vand der Waals



Lipophilicity



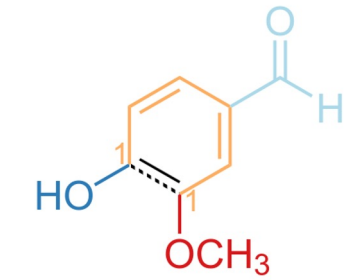
Electrostatic

davide.gadioli@polimi.it
gianluca.palermo@polimi.it

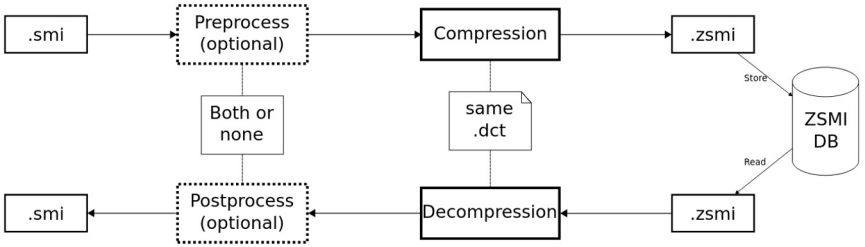
Performance Portability of ZSMILES

Porting and performance portability analysis on GPU/CPU of a compression library for HPC drug discovery applications.

(https://github.com/elvispolimi/ZSMILES)



COc1cc(C=O)ccc1O

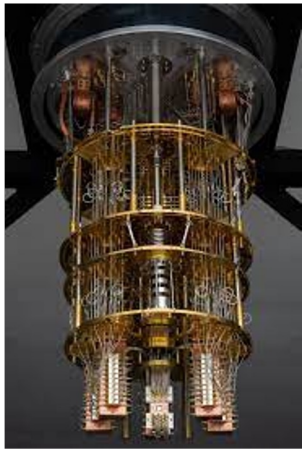


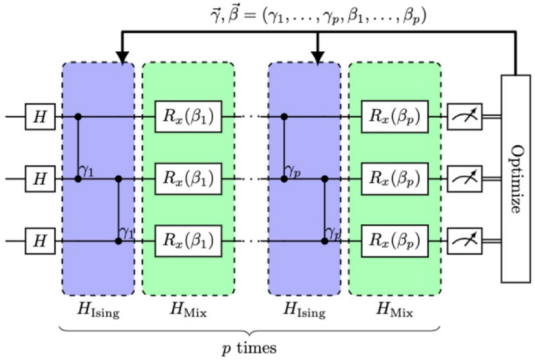
gianmarco.accordi@polimi.it
gianluca.palermo@polimi.it

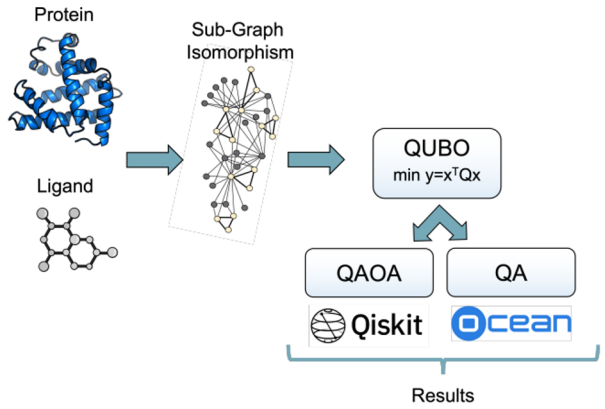
Quantum Computing for Molecular Docking Algorithm

Quantum computing optimization of a virtual screening application by leveraging quantum annealing and gate-based quantum circuits.

(https://ieeexplore.ieee.org/abstract/document/10821325)





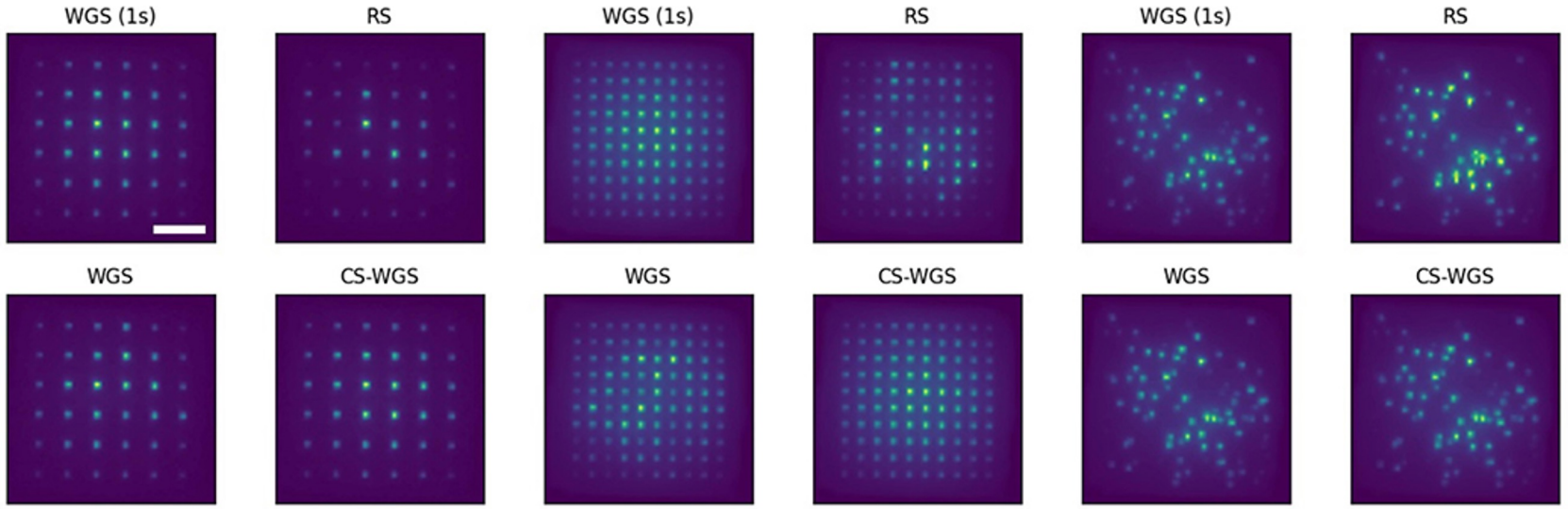


gianmarco.accordi@polimi.it
gianluca.palermo@polimi.it

Porting of an Holographic Application on GPUs

Port the kernel that compute a hologram, as point cloud, on GPU.

(https://elvispolimi.github.io/projects/holographic)



NVIDIA CUDA

SYCL

AMD ROCm

OpenMP

kokkos

HIGHWAY

gianmarco.accordi@polimi.it
gianluca.palermo@polimi.it