HAAP

Hardware Accelerators and Accelerated Programming

Special Track
ADVCOMP 2024

Chair

Dr. Biagio Peccerillo

University of Siena
Italy

e-mail: peccerillo@diism.unisi.it

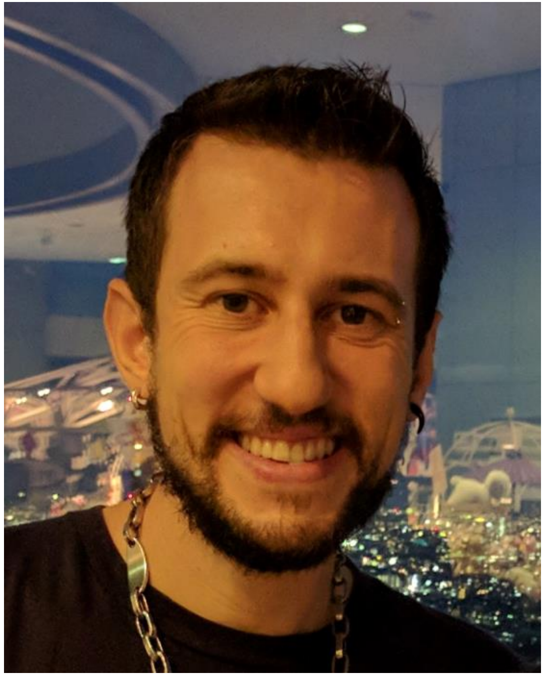
Dr. Biagio Peccerillo

Postdoctoral Researcher at the Department of Information Engineering and Mathematics at the University of Siena

Main research topics:

- hardware accelerators
- heterogeneous architectures
- productivity-oriented high-level abstraction mechanisms
- parallel algorithms

He participated in various R&D projects involving high-productivity solutions to program heterogeneous architectures*, hardware accelerators, haptic algorithms in virtual/augmented reality scenarios



* Check www.phast-library.com

Research Group

Computer Architecture Group

DIISM, University of Siena

Professor Sandro Bartolini

Researcher Dr. Biagio Peccerillo

Researcher Dr. Alessio Medaglini

PhD Student Mirco Mannino

PhD Student Davide Privitera

BS Student Emanuele Angiolilli

BS Student Gabriele Pica

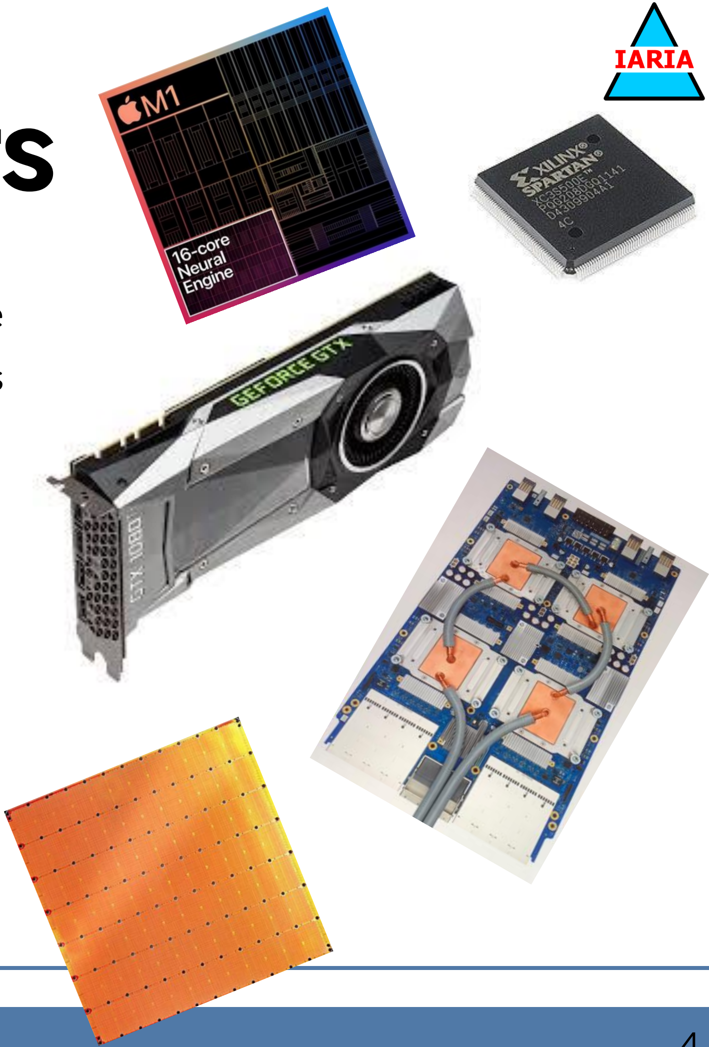
Hardware Accelerators

A **Hardware Accelerator** is a separate architectural substructure used in synergy with a general-purpose CPU that orchestrates **execution** and **task-offloading** on it

They are not meant to *enable* a particular computation, but are a fundamental way to improve **non-functional requirements**, such as:

- ✵ Throughput / latency
- ✵ Energy / power efficiency

This, in turn, enables **application scenarios**



B. Peccerillo, M. Mannino, A. Mondelli, S. Bartolini, “A survey on hardware accelerators: Taxonomy, trends, challenges, and perspectives”, JSA, 2022

Example: LLM Inference

A 3 billion parameter LLM takes around 350ms to produce a token on an NVIDIA A100 GPU

NVIDIA A100: **624 TFLOPs** (FP16) → **350 ms/token**

NVIDIA H100: **1979 TFLOPs** (FP16) → **110 ms/token**

Example: LLM Inference

A **3 billion parameter** LLM takes around 350ms to produce a token on an **NVIDIA A100 GPU**

NVIDIA A100: **624 TFLOPs** (FP16) → **350 ms/token**

NVIDIA H100: **1979 TFLOPs** (FP16) → **110 ms/token**

Intel Xeon 6980P (Q3'24): $\underbrace{3.9}_{\text{GHz (peak frequency)}} \times \underbrace{128}_{\text{cores}} \times \underbrace{32}_{\text{FP16 IPC = 512 bit / 16 bit (AVX512 instructions)}} \times \underbrace{2}_{\text{AVX512 units per core}} \times \underbrace{2}_{\text{1FMA = 2FP instructions}}$ **= 63'897.6 GFLOPs = 63.90 TFLOPs**

Example: LLM Inference

A 3 billion parameter LLM takes around 350ms to produce a token on an NVIDIA A100 GPU

NVIDIA A100: **624 TFLOPs** (FP16) → **350 ms/token**

NVIDIA H100: **1979 TFLOPs** (FP16) → **110 ms/token**

Intel Xeon 6980P: **63.90 TFLOPs** (FP16) → **3.418 s/token**

Example: LLM Inference

A 3 billion parameter LLM takes around 350ms to produce a token on an NVIDIA A100 GPU

NVIDIA A100: **624 TFLOPs** (FP16) → **350 ms/token**

NVIDIA H100: **1979 TFLOPs** (FP16) → **110 ms/token**

Intel Xeon 6980P: **63.90 TFLOPs** (FP16) → **3.418 s/token**

What about energy?

NVIDIA A100: **400W × 350 ms/token = 140 J/token**

NVIDIA H100: **700W × 110 ms/token = 77 J/token**

Intel Xeon 6980P: **500W × 3.418 s/token = 1.71 KJ/token**

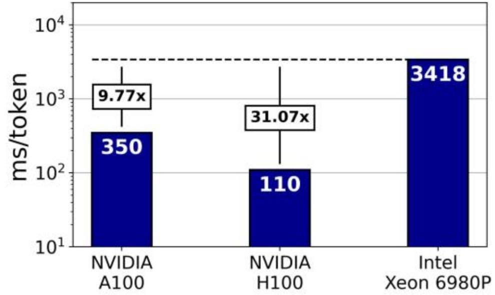
Example: LLM Inference

A **3 billion parameter** LLM takes around 350ms to produce a token on an **NVIDIA A100 GPU**

NVIDIA A100: **624 TFLOPs** (FP16) → **350 ms/token**

NVIDIA H100: **1979 TFLOPs** (FP16) → **110 ms/token**

Intel Xeon 6980P: **63.90 TFLOPs** (FP16) → **3.418 s/token**

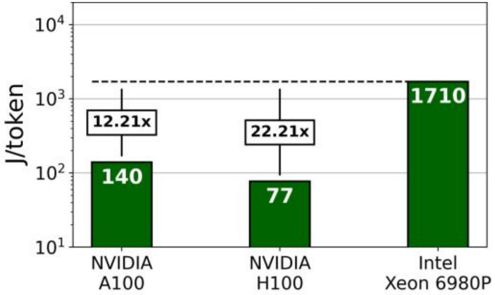


What about energy?

NVIDIA A100: **400W × 350 ms/token = 140 J/token**

NVIDIA H100: **700W × 110 ms/token = 77 J/token**

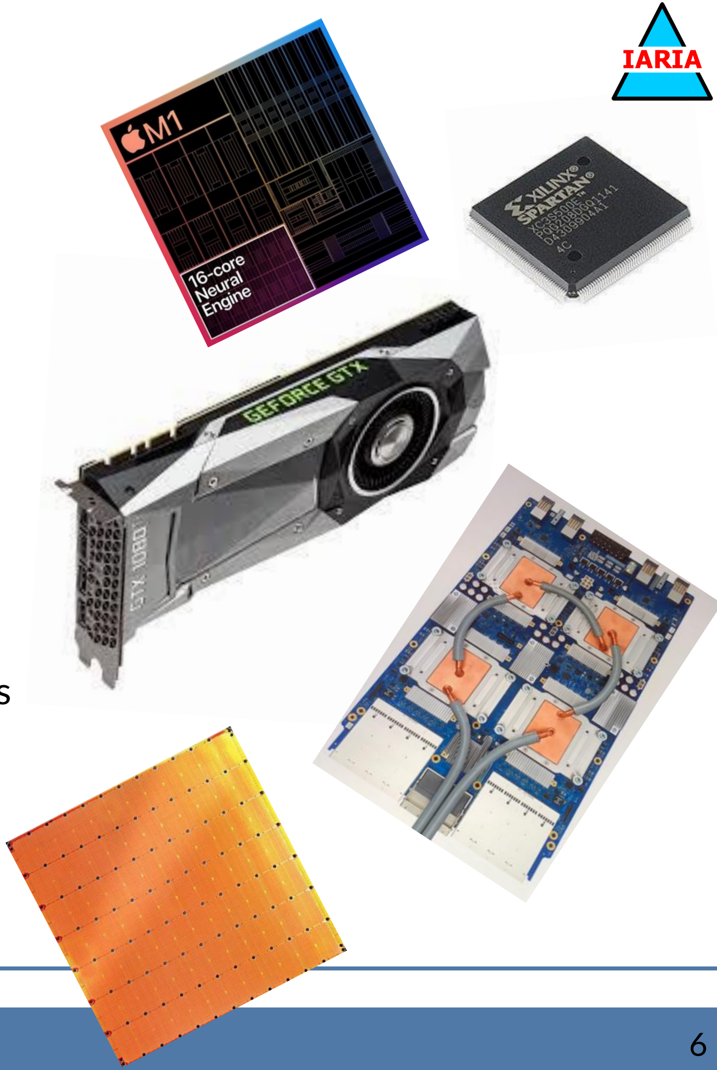
Intel Xeon 6980P: **500W × 3.418 s/token = 1.71 KJ/token**



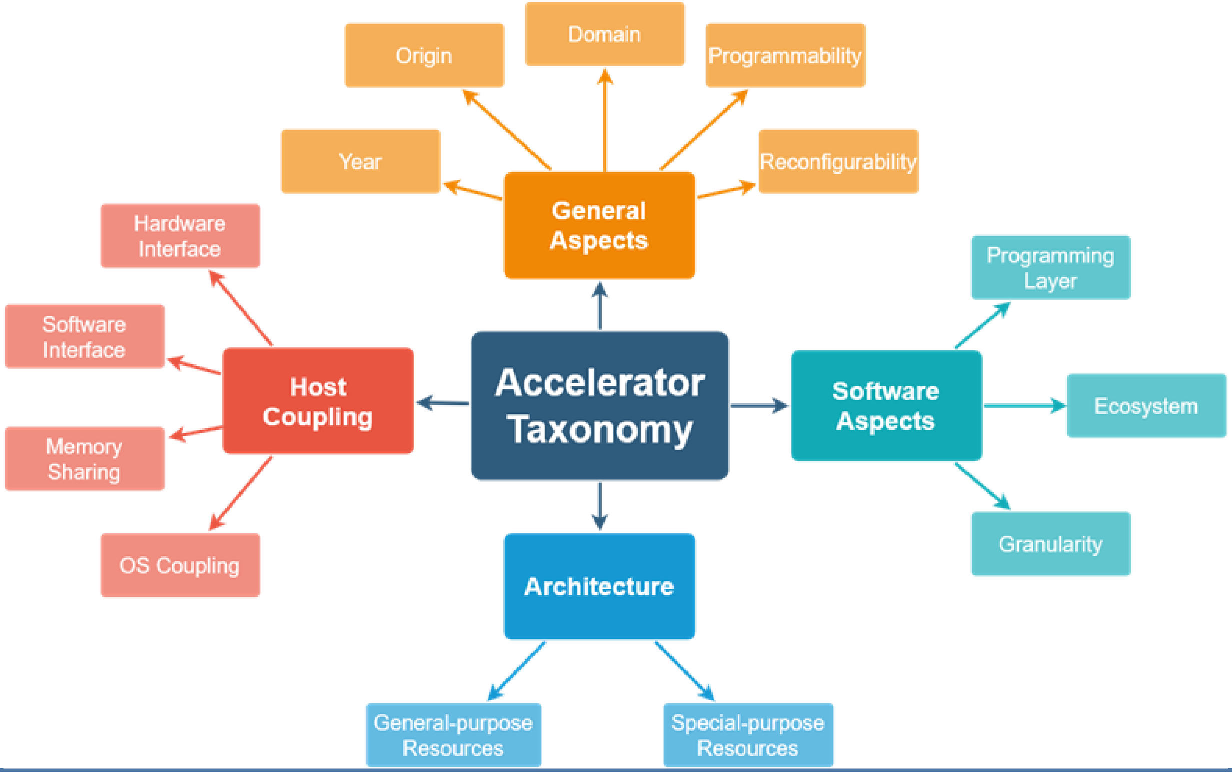
Some examples

A *very* partial list:

- Application Specific Integrated Circuits (since 1967)
- Mathematical coprocessors (Intel 8087, 1980)
- Graphics Processing Units (since NVIDIA GeForce 256, 1999)
- Field Programmable Gate Arrays (since Altera EP300, 1984)
- Tensor Processing Units (since Google TPU, 2015)
- Neural Processing Units (integrated in smartphone’s SoCs since Qualcomm Snapdragon 820, 2015)
- Processing in Memory
- Coarse-Grain Reconfigurable Arrays



Heterogeneity



The main characteristic of Hardware Accelerators is **heterogeneity**

They can be classified along four axes:

General Aspects: very high level of abstraction to quickly contextualize

Host Coupling: detail about the connection strategy to the rest of the system

Architecture: accelerator structure from a hardware standpoint

Software Aspects: software characteristics

B. Peccerillo, M. Mannino, A. Mondelli, S. Bartolini, "A survey on hardware accelerators: Taxonomy, trends, challenges, and perspectives", JSA, 2022

CPU means *stability*

The CPU ecosystem is characterized by **stability**

Hardware is stable:

- A few cores, pipelined, superscalar, three levels of cache, …

Software is stable:

- A few Operating Systems
- Same languages, compilers, libraries for a few ISAs (x86, ARM, *+ RISC-V*)

Generally, **portability** and **performance portability** are not an issue in the CPU ecosystem

- *In the worst case, recompile your program and you're fine!*

Accelerators means *challenges*

CPUs' hardware and software are limited by ***general-purposeness***, which is fundamental!

- A CPU cannot limit the nature of *runnable* programs
- CPUs are optimized to run a vast variety of programs
  - Branch predictors, caches, etc.

With hardware accelerators, designers can release this constraint to pursue ***special-purposeness***

- They target application domains

Everything is legitimate for performance and efficiency → Massive heterogeneity

This means that many aspects that we give for granted when dealing with CPUs are not so when it comes to hardware accelerators

Challenge 1: Programmability

Problem:

- Heterogeneous parallel programming is very hard
- Different programming strategies, hardware knowledge, and leaky abstractions are common

Current solutions:

- Data-parallel: hyper-specific low-level approaches that *may* be used to write high-performance code
- Machine Learning: Very high-level frameworks that try to encompass any programming need
- Other: Typically, *just libraries*

Perspectives:

- Accelerator adoption needs high-level solutions
- Consolidated approaches may be adapted to *sell* novel accelerators, even outside their domain
- For new classes of accelerators, “high-level” is preferable to “consolidated”: familiarity may come later

B. Peccerillo, M. Mannino, A. Mondelli, S. Bartolini, “A survey on hardware accelerators: Taxonomy, trends, challenges, and perspectives”, JSA, 2022

Challenge 2: Reconfigurability

Problem:

- Fine-grained reconfigurability: slow to achieve, complex tools

- Coarse-grained reconfigurability: complex design, interconnect bottlenecks

Current solutions:

- Fine-grained is well-established in FPGAs, mainly used to produce prototypes (*before ASIC*)

- Coarse-grained, as in CGRAs, is *struggling* to enter the market

Perspectives:

- Limited reconfigurable logic with fast reconfigurable time is promising

- A *hybrid* spatial architecture with programmable Processing Elements and a (fast) reconfigurable interconnect may be a breakthrough

B. Peccerillo, M. Mannino, A. Mondelli, S. Bartolini, “A survey on hardware accelerators: Taxonomy, trends, challenges, and perspectives”, JSA, 2022

Challenge 3: Coherency

Problem:

- High-level abstractions such as a *coherent virtual memory space* are usually limited to the CPU
- Managing separate memory spaces is complex and error-prone

Current solutions:

- Accelerators are usually non-coherent, with the coherency burden on the programmer
- Optimal computation-data transfer overlapping

Perspectives:

- Simple accelerators with limited uses may be included in coherency mechanisms
- Complex accelerators may offer different solutions, depending on the application

B. Peccerillo, M. Mannino, A. Mondelli, S. Bartolini, “A survey on hardware accelerators: Taxonomy, trends, challenges, and perspectives”, JSA, 2022

Accelerators are here to stay

Despite these challenges (and others…), **hardware accelerators** are seemingly *unstoppable*

We just can’t give up their promised **performance** and **efficiency** figures

+ We are unable to improve multi-core CPUs as we have done for decades

- End of frequency scaling, dark silicon, diminishing returns from parallelism, …

They are being included in **virtually every computing system**, independently of its form factor

- Server, desktop, mobile, wearable, IoT, …

In the words of Hennessy & Patterson, they brought us in **“a new golden age of computer architecture”**

We need to understand them, do more research, and address the challenges!

J.L. Hennessy, D.A. Patterson, “A new golden age of computer architecture”, Communications of the ACM, 2019

HAAP Program

- MORUS-PRNG: a Hardware Accelerator Based on the MORUS Cipher and the IXIAM Framework

  Alessio Medaglini, Mirco Mannino, Biagio Peccerillo, Sandro Bartolini

  University of Siena

- Accelerating Differential Privacy Based Federated Learning Systems

  Mirco Mannino, Alessio Medaglini, Biagio Peccerillo, Sandro Bartolini

  University of Siena