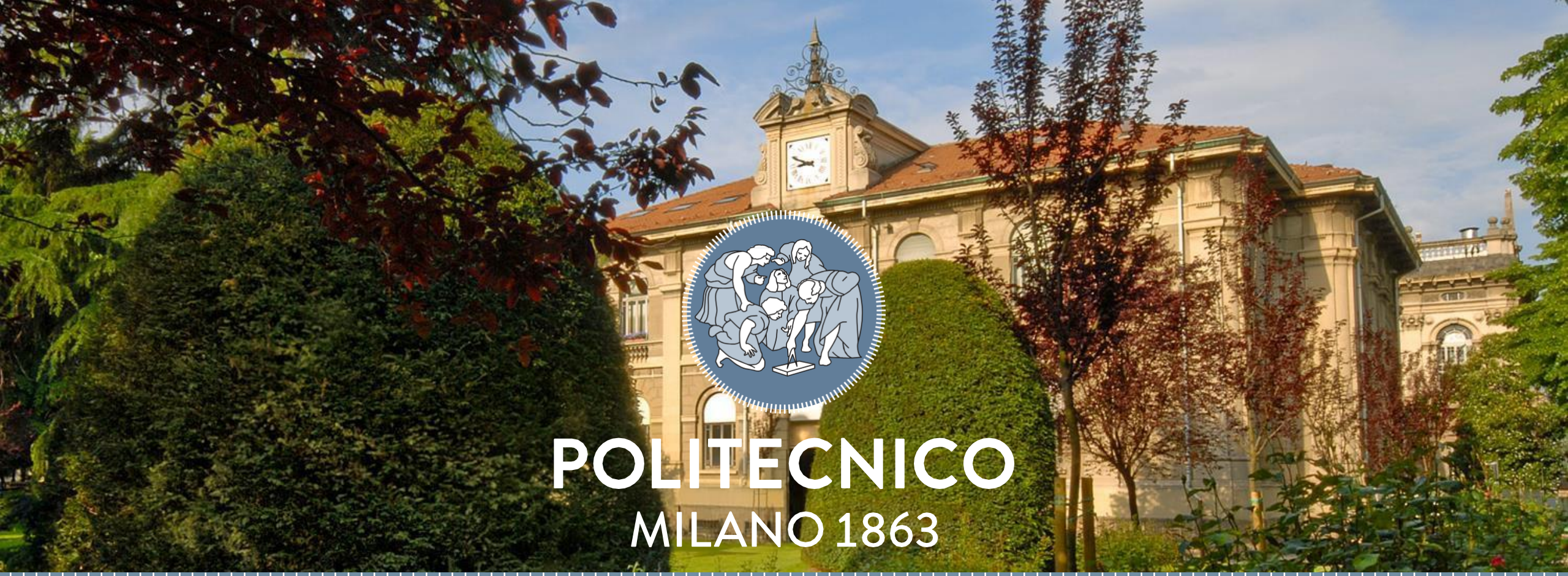

Embedded Systems

Projects (Hardware topics) – AY 2023/2024

Prof. William Fornaciari (william.fornaciari@polimi.it)
Prof. Davide Zoni (davide.zoni@polimi.it)

November 22, 2023

General information

Contacts

Prof. William Fornaciari	william.fornaciari@polimi.it
Prof. Davide Zoni	davide.zoni@polimi.it
Dr. Andrea Galimberti	andrea.galimberti@polimi.it
Davide Galli (PhD Student)	davide.galli@polimi.it
Gabriele Montanaro (PhD Student)	gabriele.montanaro@polimi.it
Giuseppe Chiari (PhD Student)	giuseppe.chiari@polimi.it
Michele Piccoli (Dott. Ing.)	michele.piccoli@polimi.it

MSc thesis proposals: directly ask to Prof. William Fornaciari and Prof. Davide Zoni

Note: For bureaucracy/score registration questions please email to Prof. William Fornaciari

Project rules

- Send email to show interest for project(s) to:
  - Prof. Zoni and the TAs supervising the project
- Get assigned a project and the corresponding supervisors
- Students receive starting material, what they need to do, etc.
- Work on the project: we do schedule one or two intermediate meetings
- When it’s done, send code and report to your supervisors.
- Finally, presentation of the project with demonstration of the work carried out
- Grades 0-10 for project and 0-7 for bibliographical research
- Validity for AOS and/or PhD course: ask for each project
  - In general, the student must drop an email to prof. Zaccaria asking if the project she/he is willing to take is suitable for AOS

[SOC-1] Design an interface between *gem5* and SystemVerilog

Interfacing *gem5* with SystemVerilog

1. Use *gem5* to create multi-thread simulation using out-of-order CPU models
2. Implement a DPI SystemVerilog interface to collect and answer to load and store requests from *gem5*

gem5 system-level and processor simulator: https://www.gem5.org

Prerequisites	C++, SystemVerilog
Score	Up to 10 points
Supervisor	Andrea Galimberti, Davide Zoni
Number of students	1-2
Extends to MSc thesis	Yes
Artifacts	Demo + Code + Report
Tags	**Complex**

SystemVerilog design of synthetic traffic generator

1. Consider the traffic generators within *gem5*
2. Implement a synthesizable traffic generator in SystemVerilog

gem5 system-level and processor simulator: https://www.gem5.org

Prerequisites	C++, SystemVerilog
Score	Up to 10 points
Supervisor	Andrea Galimberti, Davide Zoni
Number of students	1-2
Extends to MSc thesis	no
Artifacts	Demo + Code + Report
Tags	**Complex**

Bibliographical research on *AMBA 5 Coherent Hub Interface (CHI)*

1. Discussion of interconnect architecture, coherence protocol, transactions scheme of *AMBA 5 CHI* standard
2. Comparison between *AMBA 5 CHI* and *AMBA AXI+ACE*: differences, advantages and disadvantages

AMBA 5 CHI specification: https://developer.arm.com/documentation/ihi0050/latest

Prerequisites	C, SystemVerilog
Score	Up to 7 points
Supervisor	Andrea Galimberti, Davide Zoni
Number of students	1
Extends to MSc thesis	yes
Artifacts	Report + Presentation
Tags	**Complex**

[SOC-4] RISC-V CVA6: Atomic instructions

Implementation of the atomic instructions in the RISC-V CVA6 CPU

1. Analyze the SystemVerilog implementation of the CVA6 to support *ll-sc* and *rmw*
2. Execute representative benchmarks showing the effectiveness of the implementation

RISC-V ISA spec: https://riscv.org/technical/specifications/

CVA6 manual: https://docs.openhwgroup.org/projects/cva6-user-manual/

Prerequisites	C, SystemVerilog
Score	Up to 10 points
Supervisor	Andrea Galimberti, Davide Zoni
Number of students	2
Extends to MSc thesis	yes
Artifacts	Demo + Code + Report
Tags	**Complex**

[SOC-5] Gem5: Atomic instructions for the ARM CPU model

Implementation of the atomic instructions in the ARM CPU models in GEM5

1. Analyze the gem5 implementation of the ARM CPUs supporting *ll-sc* and *rmw*
2. Execute representative multi-threaded benchmarks showing the effectiveness of the implementation (multicore platform)

gem5 system-level and processor simulator: https://www.gem5.org

Prerequisites	C++, Python
Score	Up to 10 points
Supervisor	Andrea Galimberti, Davide Zoni
Number of students	2
Extends to MSc thesis	yes
Artifacts	Demo + Code + Report
Tags	**Complex**

[SOC-6] OpenNIC

OpenNIC project (Xilinx)

1. Create a verification infrastructure including at least a packet generator and the corresponding scoreboard and functional checker for the OpenNIC DUT
2. Use TCL or Python

Xilinx OpenNIC: https://github.com/Xilinx/open-nic

Prerequisites	C, SystemVerilog, Python, TCL
Score	Up to 10 points
Supervisor	Andrea Galimberti, Davide Zoni
Number of students	2
Extends to MSc thesis	yes
Artifacts	Demo + Code + Report
Tags	**Complex**

[SOC-7] Gem5/CVA6: MMU architecture and microarchitecture (NEW)

Implementation of the memory management unit (MMU) in GEM5 for ARM CPUs

- Analyze the gem5 implementation of the MMU for the ARM CPUs in gem5
- Analyze the implementation of the MMU in the CVA6 RISC-V core
- Provide representative execution examples for both implementations

gem5 system-level and processor simulator: https://www.gem5.org
RISC-V ISA spec: https://riscv.org/technical/specifications/
CVA6 manual: https://docs.openhwgroup.org/projects/cva6-user-manual/

Prerequisites	C++, Python, SystemVerilog
Score	Up to 10 points
Supervisor	Andrea Galimberti, Davide Zoni
Number of students	2
Extends to MSc thesis	yes
Artifacts	Demo + Code + Report
Tags	**Complex**

[SOC-8] uDMA: explore the architecture of the uDMA by PULP (NEW)

Implementation of the memory management unit (MMU) in GEM5 for ARM CPUs

- Analyze the architecture of the uDMA core
- Provide representative execution examples for both implementations

uDMA core: *https://github.com/pulp-platform/udma_core*
CVA6 manual: https://docs.openhwgroup.org/projects/cva6-user-manual/

Prerequisites	C++, Python, SystemVerilog
Score	Up to 10 points
Supervisor	Andrea Galimberti, Davide Zoni
Number of students	2
Extends to MSc thesis	yes
Artifacts	Demo + Code + Report
Tags	**Complex**

[ML] Hardware acceleration of machine learning tasks

ScaleHLS framework for HLS-based generation of accelerators from ML models

1. Analyze *ScaleHLS*: focus on 1) graph-, loop-, and directive-level optimizations on MLIR representation, 2) MLIR-to-C/C++ conversion (C/C++ suitable for HLS)
2. Demonstrate using *ScaleHLS* to obtain, through *Vivado/Vitis HLS*, the hardware accelerator for a *PyTorch* CNN/RNN model

ScaleHLS repository: https://github.com/hanchenye/scalehls

Prerequisites	Python, C++
Score	Up to 10 points
Supervisor	Gabriele Montanaro, Andrea Galimberti, and Davide Zoni
Number of students	1-2
Extends to MSc thesis	Yes
Artifacts	Demo + Code + Report
Tags	**Complex**

CVA6 (formerly Ariane) 64-bit, 6-stage, single issue, in-order CPU

1. Instantiate *CVA6* cores (open source, written in SystemVerilog)
2. Port Linux/FreeRTOS to and execute C applications on *CVA6*

GROUP OPENHW PROVEN PROCESSOR IP

CVA6 user manual: https://docs.openhwgroup.org/projects/cva6-user-manual/

FreeRTOS website: https://www.freertos.org/

Prerequisites	C, SystemVerilog
Score	Up to 10 points
Supervisor	Davide Galli, Davide Zoni
Number of students	1-2
Extends to MSc thesis	Yes
Artifacts	Demo + Code + Report
Tags	**Complex**

Ibex 32-bit RISC-V CPU core

1. Instantiate *Ibex* (open source, written in SystemVerilog)
2. Port Linux/FreeRTOS to and execute C applications on *Ibex*

Ibex repository: https://github.com/lowRISC/ibex
Ibex documentation: https://ibex-core.readthedocs.io/
FreeRTOS website: https://www.freertos.org/

Prerequisites	C, SystemVerilog
Score	Up to 10 points
Supervisor	Davide Galli, Davide Zoni
Number of students	1-2
Extends to MSc thesis	Yes
Artifacts	Demo + Code + Report
Tags	**Complex**

Rocket in-order single-issue RISC-V core

1. Instantiate *Rocket* (open source, written in SystemVerilog)
2. Port Linux/FreeRTOS to and execute C applications on *Rocket*

Rocket repository: https://github.com/chipsalliance/rocket-chip

FreeRTOS website: https://www.freertos.org/

Prerequisites	C, SystemVerilog
Score	Up to 10 points
Supervisor	Davide Galli, Davide Zoni
Number of students	1-2
Extends to MSc thesis	Yes
Artifacts	Demo + Code + Report
Tags	**Complex**

BOOM out-of-order multi-issue RISC-V core

1. Instantiate *BOOM* (open source, written in SystemVerilog)
2. Port Linux/FreeRTOS to execute C applications on *BOOM*

BOOM repository: https://github.com/riscv-boom/riscv-boom

FreeRTOS website: https://www.freertos.org/

Prerequisites	C, SystemVerilog
Score	Up to 10 points
Supervisor	Davide Galli, Davide Zoni
Number of students	1-2
Extends to MSc thesis	Yes
Artifacts	Demo + Code + Report
Tags	**Complex**

[CPU-5] Vortex a RISC-V GPGPU core

Vortex RISC-V gp-gpu core

1. Instantiate Vortex into a Vivado project (open source, written in SystemVerilog)
2. Analyze resource utilization for different parametrizations
3. Analyze the performance by executing few representative benchmarks

Paper: https://dl.acm.org/doi/pdf/10.1145/3466752.3480128

Repository: https://github.com/vortexgpgpu/vortex

Prerequisites	C, SystemVerilog
Score	Up to 10 points
Supervisor	Davide Galli, Andrea Galimberti, and Davide Zoni
Number of students	1-2
Extends to MSc thesis	Yes
Artifacts	Demo + Code + Report
Tags	**Complex**

PyTorch porting of *Keras*-based CNN automatic generation framework

1. Port from Keras to Pythorch Lighting of the framework
2. Run a demo to attack a pre-define side-channel dataset

Small variant: Port from Keras 2.14 to Keras 3.0 using Python 3.11

Link to paper: Reinforcement Learning for Hyperparameter Tuning in DL-based SCA

Link to repository: GitHub - AISyLab/Reinforcement-Learning-for-SCA

Prerequisites	Python
Score	Up to 10 points
Supervisor	Davide Galli, Giuseppe Chiari, and Davide Zoni
Number of students	1-2
Extends to MSc thesis	TBD
Artifacts	Demo + Code + Report
Tags	**Complex**

Repo: https://github.com/Theommy01/ESLampertiNaja

Contacts

Prof. William Fornaciari william.fornaciari@polimi.it
Prof. Davide Zoni davide.zoni@polimi.it