PhD in Information Technology and Electrical Engineering
Università degli Studi di Napoli Federico II

PhD Student: Vincenzo Maisto

Cycle: XXXVII

Training and Research Activities Report

Academic year: 2022-23 - PhD Year: Second

Vincenzo Maisto

Tutor: prof. Alessandro Cilardo

Alessandro Cilardo

Date: December 14th, 2023

1. Information:

- **PhD student:** Vincenzo Maisto **PhD Cycle:** XXXVII
- **DR number:** DR995868
- **Date of birth:** 24/09/1996
- **Master Science degree:** Computer Engineering**; University:** University of Naples Federico II
- **Scholarship type:** MUR PON
- **Tutor:** prof. Alessandro Cilardo

2. Study and training activities:

Activity	Type[1]	Hours	Credits	Dates	Organizer	Certificate[2]
Statistical data analysis for science and engineering research	Course	24	4	09/05/2023	Prof. Roberto Pietrantuono	Y
Unleashing the Power of LLMs: A Historical Perspective on Generative AI	Seminar	1	0.2	02/03/2023	Prof. Carlo Sansone, Dr. Stefano Marrone	Y
The state of the art of AI and Physics-Based Simulations in drug discovery	Seminar	1	0.2	17/03/2023	Prof. Michele Ceccarelli	Y
How to Publish Under the CARE-CRUI Open Access Agreement with IEEE	Seminar	1.5	0.3	05/04/2023	CARE-CRUI and IEEE	Y
Enhancing qubit readout with Bayesian Learning	Seminar	1	0.2	05/04/2023	Procolo Lucignano, Domenico Montemurro, Davide Massarotti, Vincenzo D'Ambrosio, Filippo Cardano and Martina Esposito	Y
Integrated Systems Seminars	Seminar	3.75	0.75	08,15,22/05/2023	IIS – Integrated Systems	Y

					Laboratory (ETH Zurich)	
Traffic Engineering with Segment Routing optimally dealing with most popular use-case	Seminar	1	0.2	23/06/2023	Valerio Persico	Y
Exploring Advanced Aerial Robotics: A Journey into Cutting-Edge Projects and Neural Control	Seminar	1	0.2	29/06/2023	Julien Mellet	Y
DaeMon: Architectural Support for Efficient Data Movement in Disaggregated Memory Systems	Seminar	1	0.2	06/07/2023	SAFARI (ETH Zurich)	Y
A RISC-V Vector-Processor for High-throughput Multidimensional Sensor Data Processing	Seminar	2	0.4	17/08/2023	ETH Future Computing Laboratory (EFCL)	Y
Economic Fitness Concepts, Methods and Applications	Seminar	1.5	0.3	09/11/2023	Scuola Superiore Meridionale	Y
Deep Learning for Railway Safety and Maintenance: Methodologies and Applications	Seminar	1.5	0.3	27/11/2023	Prof. Valeria Vittorini	Y

1) Courses, Seminar, Doctoral School, Research, Tutorship
2) Choose: Y or N

2.1. Study and training activities - credits earned

	Courses	Seminars	Research	Tutorship	Total
Bimonth 1	**0**	**0**	**9**	**0**	**9**
Bimonth 2	**0**	**0.9**	**9**	**0**	**9.9**
Bimonth 3	**4**	**1.15**	**5**	**0**	**10.15**
Bimonth 4	**0**	**0.6**	**9**	**0**	**9.6**
Bimonth 5	**0**	**0**	**10**	**0**	**10**
Bimonth 6	**0**	**0.6**	**7**	**0**	**7.6**
Total	**4**	**3.45**	**49**	**0**	**57.25**
Expected	**30 - 70**	**10 - 30**	**80 - 140**	**0 – 4.8**	

2.2. Studying and training activities – Months reported w.r.t. DM 1061

My scholarship is, partly, funded by PON "*Ricerca e Innovazione 2014-2020, Azione IV.5*", Ministerial Decree n. 1061 of the Italian Ministry of University and Research (MUR). The months reported as *in industry* refer to the effort spent in collaboration with the chosen company as partner in the PON project, i.e., A3cube Inc..
In the following table, the effort per month is detailed as reported to the MUR.

Months	Reported	Expected
In department	13.5	24
In industry	4.5	6
Abroad	6	6
Total	24	36

3. Research activity:

3.1. Topic

My research activity involves the quest for sustainable acceleration of industrial and scientific workloads with the adoption of cutting-edge technologies. Sustainability is addressed in terms of power consumption, energy efficiency and scalability to large systems, alongside the improvement of timing performance. In this sense, I focus on modern hardware computing architectures, namely Multi-Processors System-on-Chips (MPSoCs) and Field Programmable Gate Arrays (FPGAs).

Target objectives of my research activity and main experimental tracks are:

- Acceleration of industrial and scientific computing workloads.
- Energy efficient and low-power computing architectures.
- Technologically heterogeneous and scalable systems.

3.1.1. Workload and Use Cases

Applicative workloads and use cases have been identified among those of greater interest in the scientific community, i.e., vector and matrix processing, deep learning, computer vision and artificial intelligence. The collaboration with the company A3cube Inc. provided insights into an industrial framework oriented to large-scale high-performance computing (HPC) systems. The collaboration is centered around the seemingless acceleration of compute-intensive tasks in distributed file systems for datacenter infrastructures, namely erasure coding-based error correction and data compression.

3.2. Methodology

3.2.1. Heterogeneous MPSoCs

Modern hardware acceleration platforms leverage the heterogeneity of pre-existing computing architectures, technologies, and design methodologies. In MPSoC platforms, multi-core processing systems are often integrated on-chip with large Field Programmable Gate Array (FPGA) fabric and high-

performance peripherals [1-3]. Deployment of such platforms can vary from edge computing scenarios, as OS-capable single nodes in a distributed computing system, to HPC clusters as high-performance PCIe accelerators.

MPSoC devices offer the possibility to perform advanced hardware-software co-design and carry out fine-grain engineering and aggressive optimizations for target workloads and application use cases. Furthermore, tight on-chip integration of CPUs and FPGA fabric greatly reduces power-consumption by reducing the need for off-chip data-movement.

3.2.2. Energy Efficiency in Large-scale Systems

Energy efficiency is a key aspect and requirement for the design of modern, large-scale, and heterogeneous systems. Hardware/software co-design, task acceleration and offloading are, increasingly often, required for the reduction of energy consumption costs, more than for absolute speedup. Low-power SoC architecture and special-purpose accelerator design are essential to this aim, but often, not enough to cover the complex requirements of large-scale systems.

Energy efficiency is threatened at multiple levels of abstraction and complexity in such scenarios. Performance bottlenecks, resource underutilization, non-optimal system design and/or integration, stalls and starvation of expensive and energy-hungry resources are among the main causes of the waste of computing and electrical power.

Designing industry-level, high-performance, energy efficient, and large-scale systems requires detailed attention to these aspects. Expert and knowledgeable engineering is necessary to optimize timing performance, power consumption, and energy efficiency from a wide set of angles and with a system-wide perspective.

3.2.3. Advanced Hardware/Software Co-design

In order to perform state-of-the-art research and engineering on MPSoC platforms and large-scale systems, reaching a deep technological and cross-vendor understanding is necessary. I experiment and test advanced FPGA design technologies, namely dynamic partial reconfiguration, floorplanning, high-level design methodologies. Alongside the hardware aspect, an experienced and thorough software engineering perspective is also mandatory. Such experience was acquired on the hardware platforms and software environments of both leading FPGA vendors, namely AMD, formerly Xilinx, and Intel FPGA, formerly Altera, but also on Linux-based kernel and device drivers development.

Basic hardware/software co-design consists in the parallel, iterative, and continuous engineering and development of integrated hardware and software architectures. This approach requires engineering skills for both hardware and software design. On the other hand, complex and large-scale systems, such as datacenters and HPC clusters, require a more advanced methodology.

Advanced hardware/software co-design adds a more vertical skill set to the integrated and iterative design process. The additional complexity stems from the distributed nature of large-scale HPC systems and the robustness hardware accelerators require in such platforms. On one hand, complex software middlewares, e.g., Hadoop[4], are required for the managing a rich and heterogeneous set of resources and services, e.g., distributed file systems, MapReduce[5] schedulers, etc. On the other hand, such middlewares exercise very demanding workloads on the underlying hardware platforms. It follows that,

in order to properly respond to the large-scale and HPC workflow needs, the hardware infrastructure requires more advanced hardware features and abstraction layers to be exposed in a seemingless and robust framework to the client middlewares and applications.

3.3. Results

During this second year, I experimented on MPSoC device on three different levels of abstraction, namely:

- MPSoC architectural and microarchitectural design.
- Acceleration on resource-limited edge-class platforms.
- Acceleration on powerful server/HPC-class accelerator cards.

3.3.1. Low-power MPSoC Microarchitectures and Open Hardware

During my visit at ETH Zurich, I continued my work[6] on RISC-V and vector processing, hosted by the Parallel Ultra-Low Power (PULP) group of the Integrated Systems Laboratory (IIS, i.e., *Institut für Integrierte Systeme*). I worked on virtual memory support for their open-source vector co-processor. The technical and scientific details are provided in Sections **6.2** and **6.3Technical Activities**.

The adopted methodologies included:

- Study of the RISC-V ISA privileged[7], unprivileged[8], and vector[9] specifications.
- Architectural requirement analysis for virtual memory support in tightly coupled vector co-processors.
- Hardware engineering and implementation at both microarchitectural and SoC architectural levels.
- Embedded software engineering for Linux support on application-class SoC.

3.3.2. Edge MPSoCs

During the first year of my PhD, most of the experimental effort targeted edge-class MPSoCs. In particular, I experimented with Xilinx Zynq UltraScale+ devices and submitted my work on EdgeAI acceleration of CNNs with Xilinx Vitis-AI Deep Learning processing unit (DPU) to the TECS journal at the end of October 2022. The work was, finally, published [10] after a major review, where we used the proposed methodology to refine and extend the multiuser workload analysis. Starting from feedback from the reviewers, we added the following contributions:

- In order to better characterize the target platform for multiuser workloads, we proposed and implemented an efficient multiuser/multitenant task scheduling and dispatching architecture. Our scheduler integrates the platform AI runtime and implements uniform and fair assignment of software threads to hardware DPU threads/cores. The effect of such dispatching uniformity and scheduling fairness resulted in the maximization of the utilization of the DPU cores, increased the energy efficiency of the system and improved inference latency performance for users AI applications.
- We performed load and capacity analysis of the edge platform, observed the trade-offs between hardware multithreading and system load. Our findings were not trivial, since we showed a cross-point effect, i.e., multiple hardware cores do not yield the best performance for reduced loads. At such loads, single cores proved to be more efficient than a multithreaded hardware design. We showed how

hardware multithreading can overtake its overhead only once the system is subject to a higher rate of requests.

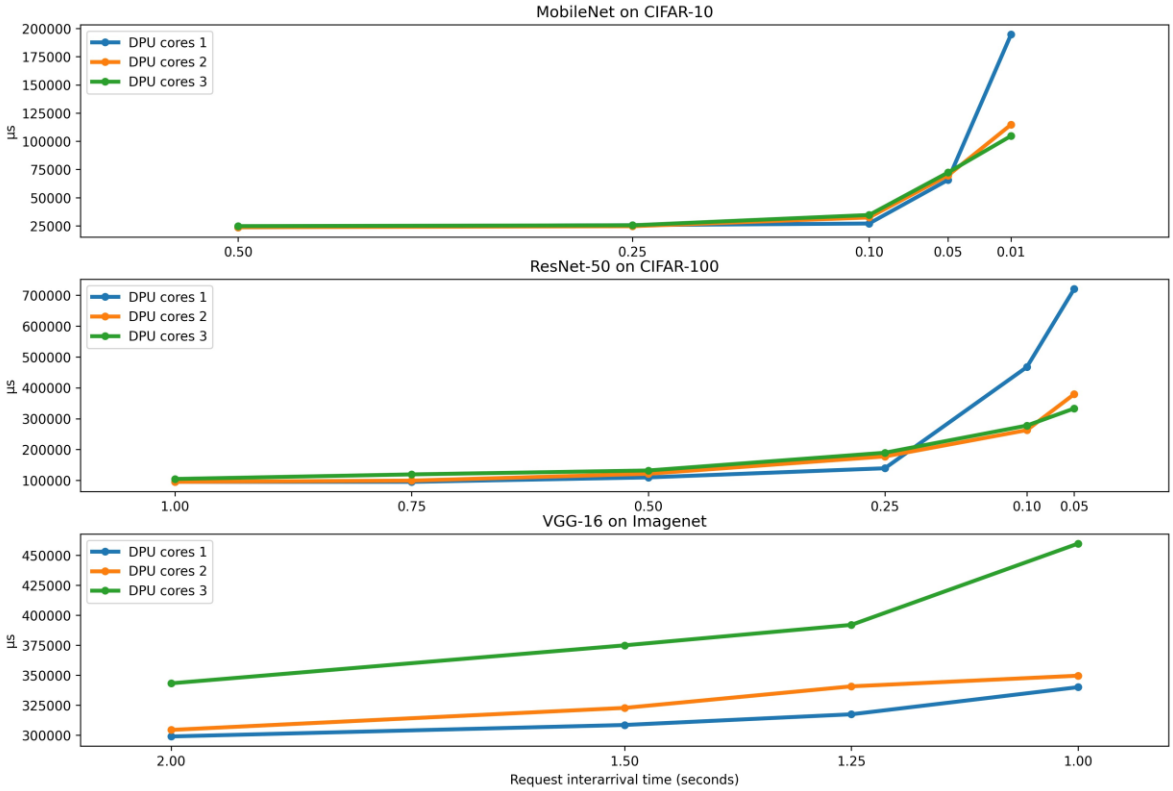

Figure 1. Multiuser load analysis from [10]

My effort on edge-class MPSoCs and AI acceleration continued in collaboration with my colleagues. I am, currently, involved in the energy consumption and accuracy trade-off analysis of edge platforms for hyperparameters of advanced model compression techniques, such as knowledge distillation. Such work is in preparation and is going to be submitted to IEEE Transactions on Sustainable Computing.

3.3.3. HPC MPSoCs

During this second year, I spent a considerable experimental effort on HPC and server-class MPSoCs. The targets MPSoCs were Intel datacenter Programmable Acceleration Cards (PACs). In particular, I experimented on the acceleration of Hadoop[4] distributed file system (HDFS) workloads, namely Reed-Solomon (RS) erasure codes[11], with Intel Rush Creek architectures featuring Arria10 FPGAs[2].

I leveraged Intel's Open Programmable Accelerator Engine (OPAE) framework and used high-level synthesis (HLS) design flow to design the target accelerator IP. The IP core was, then, integrated in OPAE's framework as an acceleration functional unit (AFU). The HLS IP had a very reduced resource utilization on the Arria10 PAC, spanning from 4% to 12% depending on the RS configuration. Furthermore, in isolation, it could reach a throughput of 2.3 to 7.8 GB/s. Once integrated into the system, the system bandwidth was bound by the peak read/write bandwidth of the PCIe Gen3 interface of the

hosting PAC. A simple throughput analysis showed how, once the accelerator load, i.e., the *cell_length* in Figure 2, could overcome the overhead caused by the latency of the PCIe bus, the PAC performance was comparable to the state-of-the-art performance of Intel's ISA-L software acceleration library[12]. In the following figures, *proto1.9*, indicates the performance of the overall system on RS kernels.

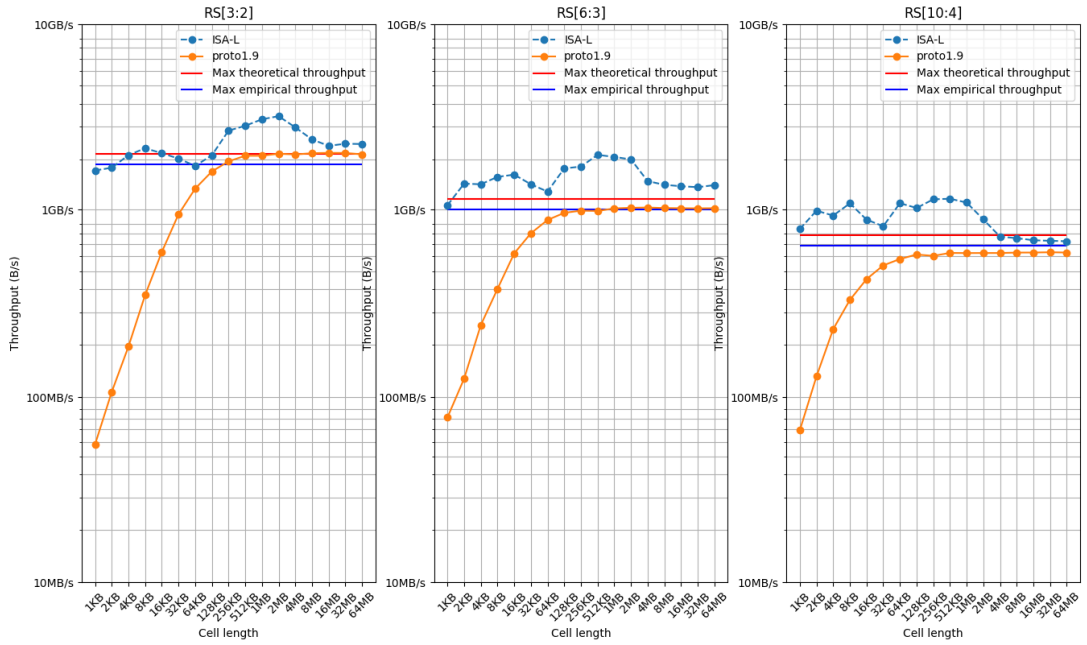

Figure 2(a). System throughput with Intel Arria10 PAC and PCIe Gen3

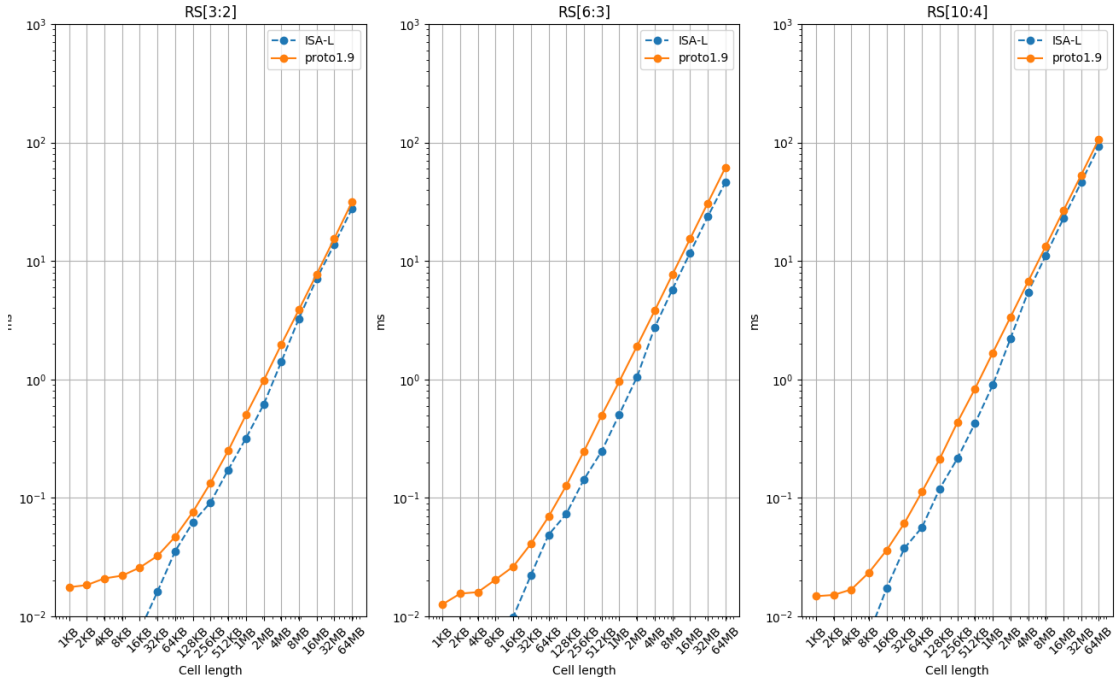

Figure 2 (b). System latency with Intel Arria10 PAC and PCIe Gen3

I integrated the OPAE runtime with HDFS and deployed the system on an 8 nodes cluster, each one equipped with Arria10 PAC. HDFS tasks are dispatched on single nodes across the local network and combined following the MapReduce computing paradigm. This results in an aggressively multithreaded workload. Intel OPAE's runtime was not designed for multithreading and thread-safety, therefore could not sustain such workload without malfunctions and faults. As a result, HDFS could not fully exploit the throughput of the multiple RS accelerators in the cluster. At this stage of the project, basic hardware/software co-design failed to deliver on the requirements of such a complex system.

The experience on the Arria10 PAC showed the need for more sophisticated thread-safety requirements in both the hardware and software platforms. Since the Arria10 support, Intel OPAE evolved considerably including advanced PCIe features, such as SR-IOV. SR-IOV virtual functions (VFs) are leveraged to expose multiple accelerator functional units (AFUs) to the host and can satisfy the thread-safety requirement through hardware isolation and Linux VFIO drivers. The performance and safety of single VFs are isolated up to the saturation of the PCIe bus bandwidth and the system crossbar interconnection. The new hardware/software framework was released and open-sourced during 2023 and I am currently working on it with the novel Intel Agilex PCIe acceleration platforms[3]. Unfortunately, the software library still lacks robust multithreading support. On the other hand, the open-source nature of the framework allows for more advanced hardware/software co-design opportunities, to satisfy the project's challenging requirements both on performance and system robustness.

4. Research products:

1) Cilardo, A., Maisto, V., Mazzocca, N., & Rocco di Torrepadula, F. (2023). An approach to the systematic characterization of multitask accelerated CNN inference in edge MPSoCs. ACM Transactions on Embedded Computing Systems. DOI: https://doi.org/10.1145/3611015 [*accepted*]
2) Cilardo, A., Maisto, V., Mazzocca, N., Rocco di Torrepadula, F. Knowledge Distillation for EdgeAI: A Systematic Evaluation of the Energy Efficiency and Accuracy Trade-off. [*in preparation for IEEE Transactions on Sustainable Computing*]
3) Draft GitHub pull requests:
    a. For Ara:
        i. [Draft] Refactoring hw source code
        ii. [Draft] Extend sw build flow for Linux environment
        iii. [Draft] Bug fixes and vstart CSR support
        iv. [Draft] Introduce virtual memory support in Ara
    b. For CVA6, CVA6-SDK and Cheshire [*in preparation*]
4) Maisto, V., Perotti, M., Cilardo, A., Benini, L. .Virtual Memory Support for RISC-V Vector Extension: A Quantitative Evaluation of Runtime Performance and Energy Efficiency. *[in preparation]*
5) Maisto, V., Cilardo, A., Billi, E. . Datacenter-scale Acceleration of Distributed Filesystems: Erasure Codes FPGA Offloading with SYCL High-Level Design and PCIe SR-IOV. *[in preparation]*

5. Conferences and Seminars Attended

Participation to the "10 Years of PULP" workshop in Lugano, Switzerland 5th-6th June 2023. https://pulp-platform.org/10years/ .

6. Periods abroad and/or in international research institutions

6.1. Hosting Institution and Visiting Period

I was hosted for six months as academic guest ETH Zurich by the PULP group in the Integrated Systems Laboratory (IIS), under the supervision of prof. Luca Benini. The visit period started the 1st May 2023 and ended 31st October 2023.

6.2. Technical Activities

I was involved in the implementation of virtual memory support for Ara v2.0[13], PULP's vector co-processor for the CVA6[14] application-class core of OpenHW. The Ara co-processor had never previously targeted Linux workloads and was not initially designed for virtual memory support.

While hosted by the PULP group, I had the opportunity to study and discuss many of their most advanced systems, both from the FPGA and front-end SoC engineering perspective. I was involved in several projects, namely CVA6, Ara, Chesire[15] and CVA6-SDK[16]. Moreover, I also took interest in other designs, namely Occamy[17-18], HERO[19], Shasheen[20] and VEGA[21]. I had the opportunity to study and analyze their low-power computing design and analysis methodologies. I acquired the tools for energy efficient system design and analysis, which are going to be fundamental skills during next year and the development of my thesis.

My technical activities involved:

- Implementing missing features from the RISC-V vector specification, which Ara was also not completely compliant with. I targeted crucial features for virtual memory and OS support, i.e., precise RISC-V CSR support and complete implementation, precise exception generation and handling, MMU and TLB interaction.
- Contextually, I fixed several pre-existing bugs in the Ara project.
- Integrate Ara in a Linux-capable SoC. The choice of the target system fell on Cheshire, PULP's new CVA6-based SoC. I supported the team in the development and debugging of the FPGA emulation prototype of Cheshire and Linux boot on a new FPGA platform, namely Xilinx Virtex US+ VCU128, which was necessary to host large hardware designs such as Ara.
- I updated and extended the CVA6-SDK to support the RISC-V vector extension. I successfully booted Linux on CVA6 in Cheshire with vector instructions support and Ara integration.

Furthermore, the vision of the PULP group is pivoted around the openness of hardware specifications, architectures, and implementations. They promote the use of open and collaborative workflows and methodologies of open-source software, namely GitHub and GitLab, also to the traditionally closed world of hardware. Together with the group, I participated in the collaborative efforts of development, bug reporting[22-23], verification, and integration[24-27] of an open-source community in the relatively more complex world of hardware design.

6.3. Scientific Output

Together with other members of the hosting group, I aim to leverage the technical and engineering efforts described above as a prototypical platform for a scientific publication. Although such work is in draft at the time of this writing, I can report its key and most interesting aspects with respect to my thesis:

1. First open-source implementation of virtual memory support for RISC-V vector extension.
2. Design-space exploration and trade-off evaluation of TLB architectures for vector processing in application-class processors.
3. Quantitative and empiric evaluation of runtime performance and energy efficiency overhead of virtual memory support, compared with a baseline bare metal execution of vector workloads.

6.4. Total Number of Months

The visit lasted for six months. It began and was concluded within this academic year.

7. Tutorship

None

8. Plan for year three

8.1. Studying Activities

As studying activities, during next year, I plan to attend:

- The course “STRATEGIC ORIENTATION FOR STEM RESEARCH & WRITING”, organized by the ITEE PhD program.
- Seminars from ITEE and other organizers.

Moreover, further studying of literature and the state-of-the-art is going to be necessary on the three tracks introduced in Section 3.1, as detailed in Section 8.4.

8.2. Research periods abroad

No other research periods abroad are planned for next year.

8.3. Courses for tutorship activities

I do not plan courses for tutorship activities.

8.4. Research activities

During next year, I plan to finalize the works started on the three tracks I followed in the last two years, namely:

- MPSoC architectural and microarchitectural design.

- Edge-class MPSoC platforms.
- HPC-class FPGA accelerator cards.

In the following, I report the details of the single activities.

8.4.1. MPSoC architectural and microarchitectural design

The technical plans for low-level MPSoC design are to finalize the collaborative engineering and integration on GitHub with the PULP group [24-27] and to proceed with the empiric evaluation introduced in Section 6.3 on timing performance and energy efficiency.

8.4.2. Edge-class MPSoC platforms

My effort on edge-class MPSoCs and AI acceleration continued in collaboration with my colleagues. I am, currently, involved in the energy consumption and accuracy trade-off analysis of edge platforms for hyperparameters of advanced model compression techniques, such as knowledge distillation. Such work is in preparation and is going to be submitted to IEEE Transactions on Sustainable Computing.

8.4.3. HPC-class FPGA accelerator cards

On the HPC side, I plan to leverage the newly found open-source nature of the hardware/software framework provided by Intel FPGA, i.e., Intel OFS, to design, deploy, and evaluate a complete Hadoop cluster for integrated FPGA-accelerated RS coding. The final system aim is to grant high-performance and energy efficient error correction, as well as thread-safety to all the hardware and software integration and abstraction layers.

The target methodology is going to be advanced hardware/software co-design. Therefore, both hardware and software aspects are going to be addressed in the integrated methodology. On the hardware side, further analysis is necessary to assess the performance and scalability of multiple VF as independent RS accelerators. Possible optimizations are at the RTL and firmware level, i.e., static bitstream of the dynamic partial reconfiguration framework. On the software side, the user runtime needs to be extended to grant thread-safety and isolation also extensively and robustly at the software user level, which the framework still misses.

8.5. Draft topic of the thesis

The thesis will focus on the definition of high-performance and energy efficient solutions in heterogenous and scalable computing architectures. The experience, methodologies and engineering tools acquired through the developments and research in the three above tracks is going to be fundamental for the definition of a complete and transversal methodology. In particular:

- Industry interests and perspectives, such as seemingless scalability, system robustness and ease to deploy and integrate accelerators in large-scale systems, are going to be considered thanks to the collaboration with the company A3cube Inc..
- Evaluation of the energy consumption figures and overall system performance of AI benchmarks in the edge domain is going to offer scientific, critical, and methodological insights on the acceleration of the workloads of the present and the future.

- The collaboration with the PULP group at the IIS laboratory of ETH Zurich is going to provide deeper knowledge and understanding of front-end SoC design and the energy and power costs of high- and low-level system design choices.

Experience at all the levels of MPSoC design and integration in small- and large-scale systems is going to offer a wider perspective for optimal engineering and fine-grain optimization on conflicting requirements like high-performance, energy efficiency and low power computing. Overspecialization and overoptimization of minor aspects of already complex subsystems is going to be avoided, in favor of a more thorough and large-scale perspective and the challenging global performance and energy requirements.

References

[1] AMD Zynq™ UltraScale+™ MPSoC, https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html
[2] Intel® Arria® 10 FPGA and SoC FPGA, https://www.intel.com/content/www/us/en/products/details/fpga/arria/10.html
[3] Intel Agilex® FPGA Portfolio, https://www.intel.com/content/www/us/en/products/details/fpga/agilex.html
[4] Apache Hadoop, https://hadoop.apache.org/
[5] Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: simplified data processing on large clusters. Commun. ACM 51, 1 (January 2008), 107–113. https://doi.org/10.1145/1327452.1327492
[6] V. Maisto and A. Cilardo, "A Pluggable Vector Unit for RISC-V Vector Extension," 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE), Antwerp, Belgium, 2022, pp. 1143-1148, doi: 10.23919/DATE54114.2022.9774501.
[7] RISC-V Privileged Architecture v1.12, https://github.com/riscv/riscv-isa-manual/releases/tag/Priv-v1.12
[8] Ratified versions of the RV32I and RV64I base ISAs and MAFDQC standard extensions, https://github.com/riscv/riscv-isa-manual/releases/tag/Ratified-IMAFDQC
[9] RISC-V Vector Extension 1.0, https://github.com/riscv/riscv-v-spec/releases/tag/v1.0
[10] Alessandro Cilardo, Vincenzo Maisto, Nicola Mazzocca, and Franca Rocco di Torrepadula. 2023. An approach to the systematic characterization of multitask accelerated CNN inference in edge MPSoCs. ACM Trans. Embed. Comput. Syst. Just Accepted (August 2023). https://doi.org/10.1145/3611015.
[11] I. S. Reed, G. Solomon, “Polynomial Codes Over Certain Finite Fields”, Journal of the Society for Industrial and Applied Mathematics, (1960), doi: 10.1137/0108018
[12] Intel ISA-L GitHub, https://github.com/intel/isa-l
[13] M. Perotti, M. Cavalcante, N. Wistoff, R. Andri, L. Cavigelli and L. Benini, "A “New Ara” for Vector Computing: An Open Source Highly Efficient RISC-V V 1.0 Vector Processor Design," 2022 IEEE 33rd International Conference on Application-specific Systems, Architectures and Processors (ASAP), Gothenburg, Sweden, 2022, pp. 43-51, doi: 10.1109/ASAP54787.2022.00017.
[14] F. Zaruba and L. Benini, "The Cost of Application-Class Processing: Energy and Performance Analysis of a Linux-Ready 1.7-GHz 64-Bit RISC-V Core in 22-nm FDSOI Technology," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 27, no. 11, pp. 2629-2640, Nov. 2019, doi: 10.1109/TVLSI.2019.2926114.
[15] A. Ottaviano , T. Benz, P. Scheffler and L. Benini, “Cheshire: A Lightweight, Linux-Capable RISC-V Host Platform for Domain-Specific Accelerator Plug-In”, https://doi.org/10.48550/arXiv.2305.04760
[16] CVA6-SDK, https://github.com/openhwgroup/cva6-sdk
[17] Occamy slides at DATE 2023, https://pulp-platform.org/docs/date2023/2023-04-19-DATE-3DIC-workshop-v4-pulp-platform.pdf
[18] Occamy GitHub, https://github.com/pulp-platform/occamy

[19] Kurth, Andreas, et al. "HERO: Heterogeneous embedded research platform for exploring RISC-V manycore accelerators on FPGA." arXiv preprint arXiv:1712.06497 (2017).
[20] L. Valente et al., "Shaheen: An Open, Secure, and Scalable RV64 SoC for Autonomous Nano-UAVs," 2023 IEEE Hot Chips 35 Symposium (HCS), Palo Alto, CA, USA, 2023, pp. 1-12, doi: 10.1109/HCS59251.2023.10254698.
[21] D. Rossi et al., "Vega: A Ten-Core SoC for IoT Endnodes With DNN Acceleration and Cognitive Wake-Up From MRAM-Based State-Retentive Sleep Mode," in IEEE Journal of Solid-State Circuits, vol. 57, no. 1, pp. 127-139, Jan. 2022, doi: 10.1109/JSSC.2021.3114881.
[22] GitHub Pull Request: Update mmu.sv for PMP violations tval value
[23] GitHub Pull Request: Update README.md
[24] GitHub Pull Request: [Draft] Refactoring hw source code
[25] GitHub Pull Request: [Draft] Extend sw build flow for Linux environment
[26] GitHub Pull Request: [Draft] Bug fixes and vstart CSR support
[27] GitHub Pull Request: [Draft] Introduce virtual memory support in Ara