Available projects

“Accelerating Data Processing in the Post-Moore Era” course

Last update: February 25, 2025

1 – Analysis and design of a RISC-V vector unit to support ML tasks

Vector processing emerged as a viable solution to efficiently support the computation of ML tasks. While several RISC-V-based implementations of vector functional units are publicly available, the actual use of such solutions in complete and production-grade computing is still an open problem, e.g. limited compiler support, seminal integration with the main CPU. First, the project requires to analyze ARA, the vector unit delivered by PULP’s project, to understand the current state of the art. Second, the project requires to integrate a customized version of ARA into an in-house RISC-V multicore computing platform targeting FPGA for datacenters.

2 – RISC-V atomic instructions in multi-core processor

Multicores are the de-facto solution to support the computation of ML tasks. However, the need to coordinate the efficient execution of the ML task on multiple cores requires the design of multi-cores with support for atomic instructions. First, the project requires to analyze the implementation of the atomic RISC-V ISA extension within CVA6 that is a processor developed in the PULP project. Second, the project requires to extend the atomic support at hardware level to a dual core architecture, where the cores and the baseline SoC can be taken from the PULP’s codebase.

3 – Efficient Virtual Memory Management in RISC-V Multi-Core Processors

Managing virtual memory in multi-core processors requires efficient translation lookaside buffer (TLB) coherence and page table handling. When memory mappings change, TLB shootdowns across cores cause latency due to inter-processor interrupts (IPIs). Page table walks also introduce stalls, especially in deeply nested memory hierarchies. Reducing these overheads is key to improving performance and scalability.

This project focuses on evaluating solutions for effective virtual memory management in RISC-V multi-core CPUs and designing SystemVerilog RTL components that implement such functionalities, e.g., a hardware-assisted TLB shootdown unit to minimize IPI overhead with a centralized invalidation controller and/or a speculative page table walker that prefetches translations, reducing access delays.

4 – Analysis and Design of Direct Memory Access Engine for RISC-V SoC

Efficient data transfer is crucial in modern computing systems, especially as memory access patterns become more complex and latency impacts performance. Direct memory access (DMA) engines are key hardware components tasked with offloading data transfer from the processor to improve throughput and reduce latency in various system environments.

This project involves analyzing the state of the art in DMA engine design, e.g., the open-source PULP project’s iDMA, and implementing an RTL solution for a RISC-V SoC in SystemVerilog. The focus will be on designing a highly efficient DMA engine that improves data transfer performance while addressing challenges like latency and throughput in modern systems.

5 – HLS-Based FPGA Acceleration of Machine Learning Models with hls4ml

Quick generation of efficient FPGA-based inference accelerators from machine learning (ML) models requires optimized high-level synthesis (HLS) workflows. hls4ml is a Python package designed for this purpose, supporting Keras, PyTorch, and ONNX frameworks while integrating with AMD Vitis HLS for deployment on AMD FPGAs. By leveraging HLS, developers can rapidly explore design trade-offs to balance performance, resource utilization, and power efficiency.

This project aims to use hls4ml to generate FPGA accelerators for various ML applications and evaluate their efficiency in terms of performance, area, and other key quality metrics. Through systematic analysis of different design choices, the goal is to optimize inference acceleration while ensuring practical hardware feasibility.