RISC-V Summit Europe 2025

LEN5: an Out-Of-Order RISC-V Microprocessor

Vincenzo Petrolo, Flavia Guella, Michele Caon

Supervisors: Prof. Maurizio Martina, Prof. Guido Masera

Introduction

Computers are dead Aside from the undeniable yet expensive benefits brought by the advancements in the semiconductor manufacturing process, integrated computer systems **have not drastically changed** since their introduction in the 70s.

The combined action of a shift in the programming paradigm towards **data-driven algorithms** and the ever-increasing **difficulties and costs** involved when trying to keep pace with Moore's Law exposed the limitations of traditional computer architectures:

- Researchers have pushed the state of the art in existing techniques (i.e., branch predictors, prefetching strategies) to the point that performance improvements hardly ever reach even **a few percentage points**.
- Traditional Von Neumann architectures **struggle to feed the hardware computing engines** with the massive amount of data required by modern computationally expensive workloads (did anyone mention machine learning yet?).

Long live the computers In this environment, the **open and extensible** RISC-V Instruction Set Architecture (ISA) established itself as the perfect **architectural and microarchitectural playground**, paving the way to a so-called "Renaissance" in computer architecture towards the definition of next-generation **heterogeneous** computer systems.

Open hardware for all! While the consolidation of RISC-V as the ISA of choice in both industry and academy is an excellent achievement for the affirmation of "open hardware", it comes with the side-effect of **fragmentation**:

- Several RISC-V implementations (PULP, among the others) ⇒ **disperse** the scientific effort ⇒ **steep learning curve** for new adopters
- Inconsistent software and OS support among frameworks (e.g., EEI).
- Several commonly used implementations are impractical to modify.

Need for specialization Most RISC-V implementations fit in the microcontroller range of products, however:

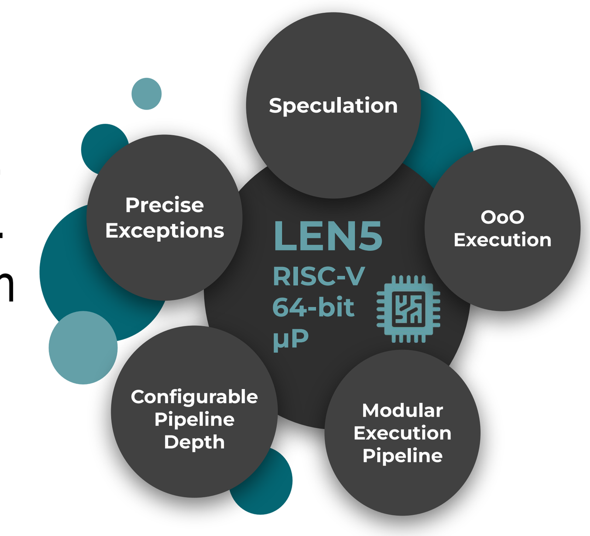
- Modern application software requires hardware support for **computationally-intensive workloads**, even on **edge devices** relying on limited power sources.
- In these systems, memory transfers often represent a bottleneck for performance and an unacceptable overhead for energy efficiency.

Introducing LEN5 Starting from the assumptions above, the **LEN5** RISC-V microprocessor is being developed as a **modular** platform designed to be as simple as possible to configure, extend, and modify.

Methodology

LEN5 (**Fig. 2**) is a modular, highly-configurable, 64-bit, out-of-order, speculative RISC-V microprocessor, described in **SystemVerilog**.

Features



- ✓ RV64I ISA with "Zicsr", "M" and "F" [3] extension.
- ✓ Small **prefetch queue** to compensate for memory latency.
- ✓ Global **gshare branch predictor** and branch target buffer.
- ✓ **Dynamically scheduled**, **out-of-order** (OoO), execution pipeline.

Modularity

- ✓ System-wide AXI-like (valid-ready) **handshaking** protocol.
  - Execution Unit (EU) **latency-independent** execution pipeline.
  - Easy to expand with **custom tightly-coupled EUs**.
- ✓ **Reservation stations** for each EU, allowing for self-contained control.
- ✓ **Store Buffer** acts as a level-zero cache by holding committed store instructions.
- ✓ **Common Data Bus** for fast result forwarding and broadcasting between EUs.
- ✓ **ReOrder Buffer** to store produced results waiting for commit.

Software

- ✓ "PULP-like" bootstrap code and bare-metal system calls.
- ✓ Automated memory image creation (from SW) and simulation.
- ✓ **make-based** HW/SW build system.
- ✓ High-level, modular **emulator** for **fast prototyping** and **verification**.

Configurability

- ✓ **Configuration files**
  - Enable/disable features and extensions.
  - Modify buffers size, pipeline depth, number of reservation stations, etc.

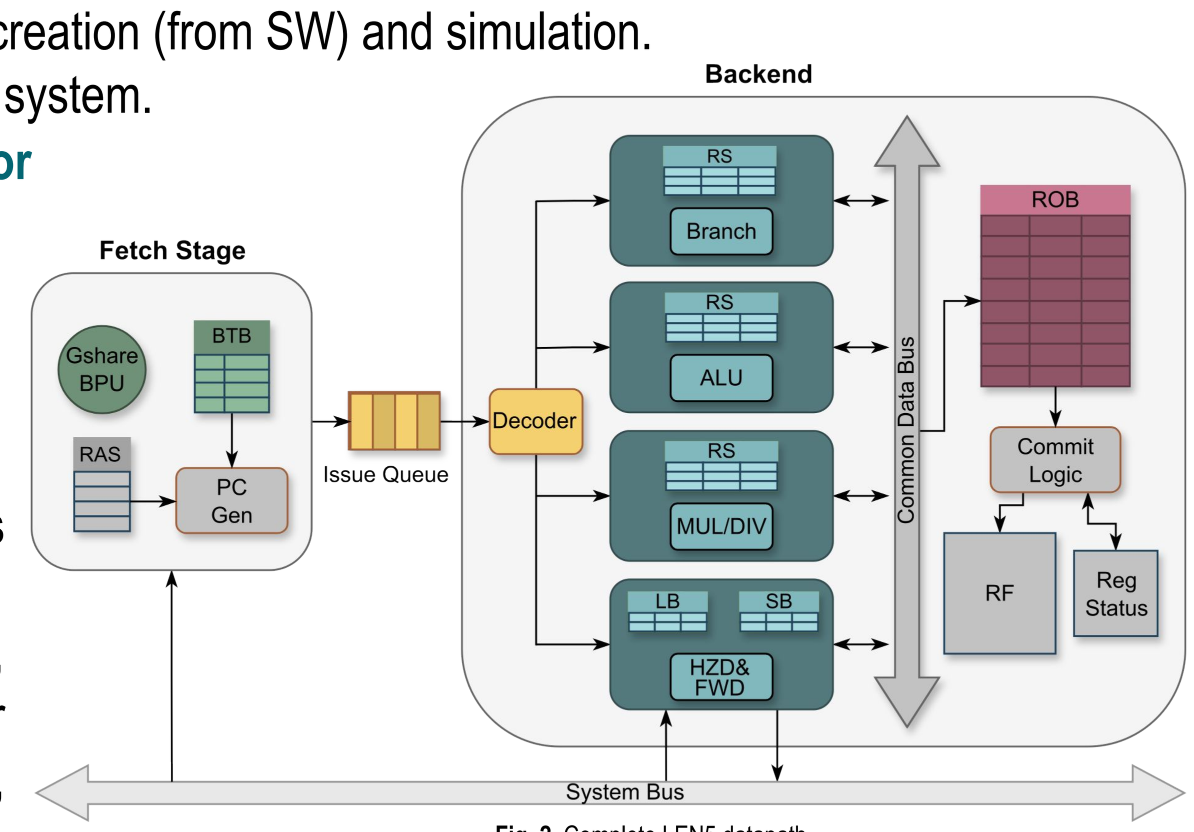


Fig. 2 Complete LEN5 datapath.

Results

Fig.3 compares LEN5 (`rv64im`) *MaxPerf* IPC performance to the `cv32e40p` (`rv32imc`) [2] 32-bit in-order RISC-V when integrated inside the X-HEEP [4] MCU running the **Embench** benchmarks.

- **IPC superiority** manifests in benchmarks with low control-flow orientation: `crc32`, `edn`, `matmult-int`
- **Misprediction** costs due to known BPU/BU synchronization issues negatively impact LEN5 IPC performance on data-dependent benchmarks.
- **Lower execution time** is achieved thanks to the 64-bit ISA that **retires fewer instructions** on particular benchmarks: `aha-mont64` (2.4x), `st` (1.4x), `matmult-int` (1.6x).

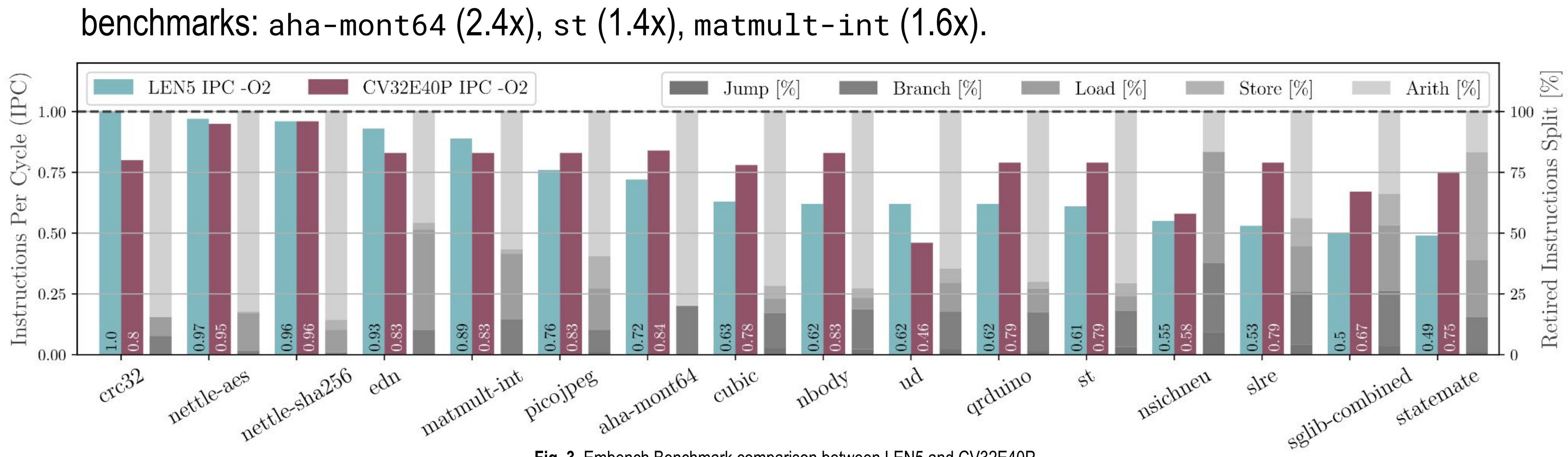


Fig. 3 Embench Benchmark comparison between LEN5 and CV32E40P

Synthesis on the TSMC65LP was performed.

Three different LEN5 presets were chosen as user-ready:

- ***MaxPerf***: Full-fledged LEN5 variant featuring a Multiplier and Divider.
- ***AvgPerf***: Carefully tuned LEN5 variant featuring a Serial Multiplier (**~55 %** of *MaxPerf* complexity).
- ***MinArea***: Basic LEN5 variant for area constrained systems (**~50 %** of *MaxPerf* complexity).

Fig. 4 shows area contributions for the *MaxPerf* variant. Larger contribution comes from the Multiplier followed by the ALU's XL reservation station.

When *MaxPerf* is integrated in X-HEEP, the area **only rises by ~12 %** when compared to the standard X-HEEP system (featuring `cv32e40p`).

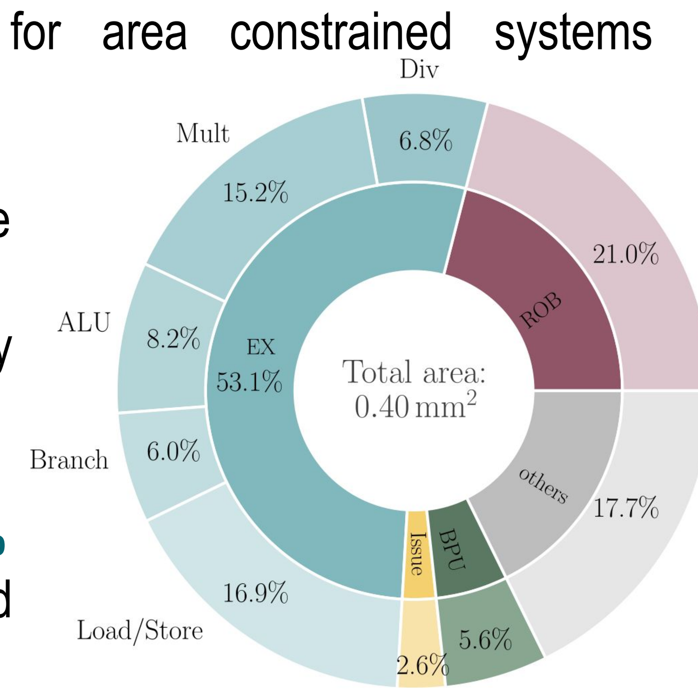


Fig. 4 Area Partition for the *MaxPerf* LEN5 variant

Future work

Work in progress The following milestones are under active development:

- Multiple-issue, to properly take advantage of the out-of-order nature of **LEN5**.
- **Speculative Load** execution to increase reordering opportunities.
- Non-speculative **secure execution mode** to prevent side-channel attacks.

Critical features required to make it a suitable platform for advanced research projects:

- C (compressed) and A (atomic) instruction set extensions.
- Software and hardware interrupt support.

Comparative analyses against leading-edge OoO CPUs.

References

[1] M. Caon, V. Petrolo, F. Guella *et al.*, "Seeing Beyond the Order: a LEN5 to Sharpen Edge Microprocessors with Dynamic Scheduling", *21st ACM International Conference on Computing Frontiers: Workshops and Special Sessions (CF '24)*, 2024, pp. 47–50.

[2] M. Gautschi *et al.*, "Near-Threshold RISC-V Core With DSP Extensions for Scalable IoT Endpoint Devices", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2017, vol. 25, no. 10, pp. 2700-2713.

[3] S. Mach *et al.*, "FPnew: An Open-Source Multiformat Floating-Point Unit Architecture for Energy-Proportional Transprecision Computing", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2021, vol. 29, no. 4, pp. 774-787

[4] Pasquale Davide Schiavone *et al.*, "X-HEEP: An Open-Source, Configurable and Extendible RISC-V Microcontroller", 20th ACM International Conference on Computing Frontiers (CF '23), 2023, pp. 379–380.

Politecnico di Torino
Department of Electronics and Telecommunications