

# MemPool Flavors: Between Versatility and Specialization in a RISC-V Manycore Cluster

Sergio Mazzola<sup>1\*</sup>, Yichao Zhang<sup>1</sup>, Marco Bertuletti<sup>1</sup>, Diyou Shen<sup>1</sup>, and Luca Benini<sup>1,2</sup>

<sup>1</sup>Integrated Systems Laboratory (IIS), Swiss Federal Institute of Technology (ETH Zürich), Switzerland

<sup>2</sup>Department of Electrical, Electronic and Information Engineering (DEI), University of Bologna, Italy

## Abstract

*As computational paradigms evolve, applications such as attention-based models, wireless telecommunications, and computer vision impose increasingly challenging requirements on computer architectures: significant memory footprints and computing resources are demanded while maintaining flexibility and programmability at a low power budget. Thanks to their advantageous trade-offs, shared-L1-memory clusters have become a common building block of massively parallel computing architectures tackling these issues. MemPool is an open-source, RISC-V-based manycore cluster scaling up to 1024 processing elements (PEs). MemPool offers a scalable, extensible, and programmable solution to the challenges of shared-L1 clusters, establishing itself as an open-source research platform for architectural variants covering a wide trade-off space between versatility and performance. As a demonstration, this paper compares the three main MemPool flavors, Baseline MemPool, Systolic MemPool, and Vectorial MemPool, detailing their architecture, targets, and achieved trade-offs.*

## Introduction

Manycore systems are a class of parallel computer architectures ranging from tens to thousands of PEs. They cover a wide trade-off space among versatility, performance, and energy efficiency, where their position is determined by the specialization of the PEs, their interconnect topology, and the nature of the memory subsystem. Their massive parallelism makes manycore systems particularly suitable for many transformative technologies such as telecommunications, with Beyond-5G (B5G) and 6G, artificial intelligence, with large language models (LLMs), and computer vision, with extended reality (XR). Along with the high computational demands of these target applications, however, massive parallelism also introduces challenges such as complex programming models and communication bottlenecks.

MemPool is an open-source, RISC-V-based manycore cluster scaling up to 1024 PEs sharing a pool of multi-banked L1 scratchpad memory (SPM) through a hierarchical, low-latency interconnect [1]. MemPool's PEs are small, memory-latency-tolerant 32-bit RISC-V cores supporting custom instruction set architecture (ISA) extensions [2]. MemPool strikes a new trade-off between general-purpose, inefficient systems and fully specialized, high-performance architectures. The simple, full-fledged cores allow programmability and versatility, with their flexible datapath enhancing performance. Moreover, the hierarchical interconnect provides the cores with low-latency and energy-efficient access to any address of the shared memory, decoupling the system from domain-specific dataflows.

Since its first developments in 2020, MemPool's flex-

ibility has proven to extend beyond just architectural versatility. Not only is it an efficient, general-purpose system, but it has also evolved into a modular and adaptable open-source development platform. Its ease of experimentation and extensibility, together with the wide range of available software runtimes and computational kernel libraries, supported its usage as a research tool. This led to the emergence of multiple MemPool *flavors* and, to date, to the tapeout of two chips<sup>1</sup>. Such continuous evolution has further expanded the range of trade-offs covered by MemPool while maintaining efficiency and general-purpose applicability within the domain of highly parallel manycore systems.

## MemPool Flavors

Supported by MemPool's open-source, RISC-V-based nature, research into the topic of manycore systems led to the development of many MemPool flavors. While MemPool features 256, *TeraPool* scales up to 1024 cores with floating-point support, targeting 6G base-band processing. *ITA MemPool* (Integer Transformer Accelerator MemPool) focuses on transformer models processing, and *CachePool* explores the usage of a cache hierarchy in place of the shared L1 SPM. In the following, we compare the main 256-PE variations of MemPool, analyzing their architecture and trade-offs.

**Baseline MemPool** In its baseline version, MemPool features 256 32-bit Snitch cores with integer arithmetic, atomic memory operations (AMOs), and support for custom digital signal processing (DSP) instructions. The 256 cores share 1 MiB of L1 SPM and

\*Corresponding author: [smazzola@iis.ee.ethz.ch](mailto:smazzola@iis.ee.ethz.ch)

<sup>1</sup> IIS Chip Gallery (ETH Zürich). *Minpool* (2021), <http://asic.ethz.ch/2021/Minpool.html>. *Heartstream* (2024), <http://asic.ethz.ch/2024/Heartstream.html>.

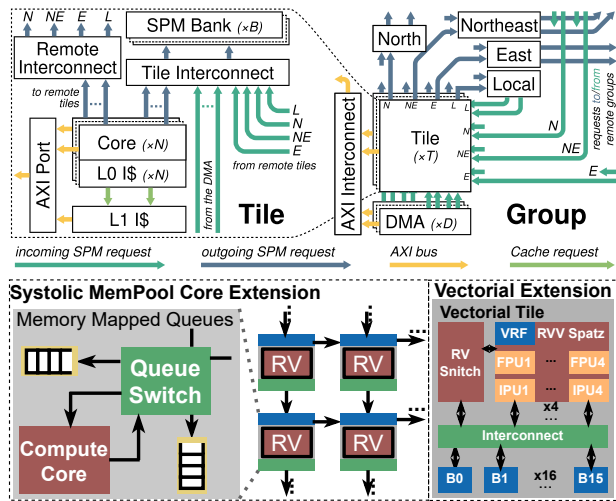
have independent instruction paths. Figure 1 shows MemPool’s hierarchical architecture, where each tile interconnects 4 cores to its 16 SPM banks with single-cycle latency. 16 tiles form a fully connected group, and 4 groups connect together to compose the MemPool cluster. Each hierarchy level adds two cycles of latency to the memory transactions, easily hidden thanks to the cores’ lightweight scoreboard.

**Systolic MemPool** Many parallel workloads common to telecommunications, artificial intelligence, and image processing feature a highly regular dataflow that can be leveraged to boost performance and energy efficiency. Systolic MemPool features a low-overhead ISA extension to implement implicit inter-PE communication and synchronization through L1-mapped queues [3]. This enables MemPool to configure arbitrary systolic topologies among its PEs, without compromising the versatility of the cluster as in systolic-array-based architectures.

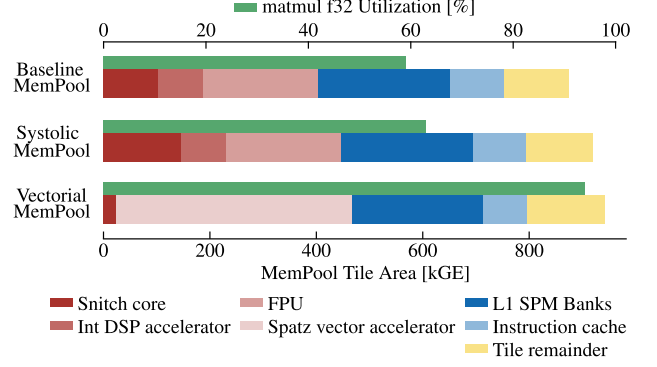
**Vectorial MemPool** Another approach to meeting the intense memory requirement of MemPool’s target applications is vector processing. In Vectorial MemPool, a tile only features one scalar 32-bit Snitch core, coupled with a Spatz vector accelerator based on the RISC-V Vector Extension (RVV) [4]. Each Spatz accelerator includes 4 32-bit floating-point units (FPUs). At the cost of the reduced versatility, Vectorial MemPool leverages the data-level parallelism (DLP) of vectorial workloads with single instruction, multiple data (SIMD) execution, greatly boosting their performance and energy efficiency.

## Flavor Tasting

Figure 2 provides an intuition of the wide trade-off range covered by the flavors of MemPool. The area



**Figure 1:** Architecture of the baseline MemPool, also highlighting its systolic and vectorial extensions.



**Figure 2:** Area breakdown of the MemPool tile in the main MemPool flavors and the corresponding 32-bit floating-point matmul utilization.

measurements refer to the MemPool’s tile implementation in GlobalFoundries’ 12LPP 12 nm advanced FinFET technology. For a proxy of the performance, we employ the utilization of a 32-bit floating-point matrix multiplication kernel.

Baseline MemPool achieves a utilization of 59%, where the remaining time is spent on control instructions and synchronization. Systolic MemPool is able to increase the cluster utilization by reducing the amount of explicit load/store instructions. With an area overhead of 5%, it brings a 7% improvement in matmul’s performance. With its large vector accelerator, Vectorial MemPool reports the largest tile, with an area 8% larger than Baseline MemPool. Additional area spent on hardware specialization negatively impacts other generic workloads with higher power consumption. In addition to this, Vectorial MemPool only features one general-purpose core per tile, which decreases its performance with non-vectorizable workloads. However, at the cost of versatility, it achieves up to 94% utilization with workloads optimized for RVV. All three *flavors* achieve 800MHz@ (TT, 0.8V), delivering a single-precision floating-point performance of up to 204.8 GFLOP/s.

## References

- [1] Samuel Riedel et al. “MemPool: A Scalable Manycore Architecture With a Low-Latency Shared L1 Memory.” In: *IEEE Transactions on Computers* (2023).
- [2] Florian Zaruba et al. “Snitch: A tiny pseudo dual-issue processor for area and energy efficient execution of floating-point intensive workloads.” In: *IEEE Transactions on Computers* (2020).
- [3] Sergio Mazzola, Samuel Riedel, and Luca Benini. “Enabling Efficient Hybrid Systolic Computation in Shared-L1-Memory Manycore Clusters.” In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* (2024).
- [4] Matteo Perotti et al. “Spatz: Clustering Compact RISC-V-Based Vector Units to Maximize Computing Efficiency.” In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2025).