

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

MinPool: A 16-core NUMA-L1 Memory RISC-V Processor Cluster for Always-on Image Processing in 65nm CMOS

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:

Riedel, S., Cavalcante, M., Frouzakis, M., Wüthrich, D., Mustafa, E., Billa, A., et al. (2023). MinPool: A 16-core NUMA-L1 Memory RISC-V Processor Cluster for Always-on Image Processing in 65nm CMOS [10.1109/ICECS58634.2023.10382925].

Availability:

This version is available at: <https://hdl.handle.net/11585/958741> since: 2024-02-17

Published:

DOI: <http://doi.org/10.1109/ICECS58634.2023.10382925>

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).
When citing, please refer to the published version.

(Article begins on next page)

MinPool: A 16-core NUMA-L1 Memory RISC-V Processor Cluster for Always-on Image Processing in 65nm CMOS

Samuel Riedel* Matheus Cavalcante* Manos Frouzakis* Domenic Wüthrich*
Enis Mustafa* Arlind Billa* Luca Benini*[†]

*IIS, ETH Zürich [†]DEI, University of Bologna

*{sriedel,matheus,frouzake,domenicw,emustafa,abilla,lbenini}@ethz.ch

Abstract—Always-on image processing is crucial for many applications, such as face and attention detection, and it is usually offloaded to dedicated, energy-efficient image processors. These processors need to be flexible and scalable to follow the rapid evolution of image sensors and always-on image processing workloads. A flexible architecture is the shared memory cluster, where multiple cores are tightly coupled with L1 memory. However, current clusters are not latency tolerant and follow a uniform memory access approach, which limits their frequency and scalability. The MemPool architecture [1] lifts those constraints by combining latency-tolerant cores, pipelined functional processing units, and a non-uniform memory access interconnect. This paper presents *MinPool*, a low-power image processor for always-on functions implemented in TSMC’s 65 nm technology and based on a tailored MemPool architecture. Thanks to an instruction set architecture extension tuned for image processing and the low-leakage process, it achieves excellent utilization results with IPCs of up to 0.98 and an energy efficiency of 65 GOPS/W for key image processing kernels.

Index Terms—ASIC, multicore, manycore, RISC-V

I. INTRODUCTION

In the era of modern IoT devices, the demand for parallel computing power under limited power budgets continues to increase while applications and algorithms evolve quickly. For example, always-on cameras need extremely efficient continuous image processing to enable functionalities like face detection for screen locks, hands-free control of devices, and barcode detection. A common architecture for energy-efficient image processing is a systolic array like Google’s Pixel Visual Core [2]. However, as for all domain-specific accelerators, the rigid interconnect and specialized processing elements (PEs) are designed for specific applications and do not adapt well to new algorithms. Cache-based general-purpose processors are more flexible, but their caches and coherency mechanisms impose significant power overheads and limit scalability.

The low-latency shared-memory cluster architectural template strikes a good balance between performance and flexibility. It consists of a cluster of cores sharing low-latency scratchpad memory (SPM). It is used across a wide range of domains, from GreenWaves’ ultra-low-power GAP8 processor used in Bitcraze’s AI-deck with the Himax ultra-low power camera [3] to NVIDIA’s streaming multiprocessors used as building blocks in general-purpose graphics processing units (GPGPUs) [4].

MemPool is a parametric low-latency shared-memory cluster that scales to hundreds of cores through a hierarchical architecture [1]. It allows independently programmable cores to access a large, distributed L1 SPM in under five cycles. Latency-tolerant RISC-V cores, equipped with custom digital signal processing (DSP) units, ensure high performance. MemPool is designed to be easily programmed, and it can efficiently run not only data-intensive regular workloads but also highly irregular workloads. Through an open-source and parametric design, MemPool can be employed in many use cases.

This paper presents MinPool, an image processor based on MemPool’s architecture. MinPool is tuned for energy-efficient processing of always-on, low-power camera data and implemented in TSMC’s low-leakage 65 nm technology. It shows the advantage of combining small, latency-tolerant cores with non-uniform memory access (NUMA) memory and custom DSP extensions for flexible, ultra-low power edge processing. We present the following contributions:

- The design of a full image processing system-on-chip (SoC) containing a 16-core parametrization of the MemPool cluster, equipped with DSP units and 64 KiB of L1 SPM; 64 KiB of L2 on-chip memory; and peripherals like UART, JTAG, GPIO, and a serial link interface for always-on cameras.
- The MinPool chip fabricated in TSMC’s 65 nm technology.
- A comprehensive evaluation of the MinPool chip in terms of performance, power, and area. We present chip measurements for key image processing applications at different frequencies and voltage points, showing the advantage of MemPool and its DSP extension for edge processing.

For critical image processing kernels, MinPool reaches high performances of up to 9 GOPS, which corresponds to 0.94 instructions per cycle (IPC) or a compute unit utilization of 70%. Consuming only 20 mW, it also achieves a remarkable energy efficiency of 65 GOPS/W, which is competitive even with chips using more aggressively scaled technology nodes.

II. ARCHITECTURE

MinPool implements the MemPool cluster, which is an open-source¹, flexible, and parametric manycore architecture that scales to hundreds of individually programmable cores sharing a low-latency L1 SPM [1]. The first building block of MemPool’s hierarchy is the *tile*, which combines a few cores, the first levels of the instruction cache, and a subset of the shared L1 memory banks connected with a fully-connected crossbar to the cores. Each tile also has four *remote* ports, allowing the cores to make requests into remote tiles’ SPM. Analogously, each tile has incoming request ports to serve memory requests from remote tiles. Incoming and outgoing request ports are pipelined to allow scaling while maintaining a high frequency at the cost of latency. A *group* combines multiple tiles and connects their remote ports to the target groups and their tiles’ SPM. Finally, four groups make up the full MemPool cluster with connections between themselves connecting the remote interconnects. Additional pipeline stages at this interconnect level keep the critical path short. MemPool’s hierarchical design and pipelined SPM interconnect allow all cores to share a single, large L1 memory while keeping the access latency below five cycles in the absence of contention.

MemPool’s PEs are Snitch cores, which are small, single-stage 32-bit cores supporting RISC-V’s RV32IMA instruction set architecture (ISA) [5]. Furthermore, each core is equipped with a pipelined image processing unit (IPU) unit enabling support for the Xpulping ISA extension [6], with instructions like multiply–accumulate (MAC), post-increment load and stores, and single instruction, multiple data (SIMD) support. A key feature of Snitch is its ability to handle multiple outstanding instructions. It makes using a pipelined DSP unit and L1 interconnect in MemPool possible by hiding the latencies of those instructions and thereby sustaining high throughput.

A tree-like Advanced eXtensible Interface (AXI) interconnect connects the tiles with the system-level AXI crossbar, where the peripherals and long-latency L2 memory reside. The boot ROM and control registers are also connected to this crossbar.

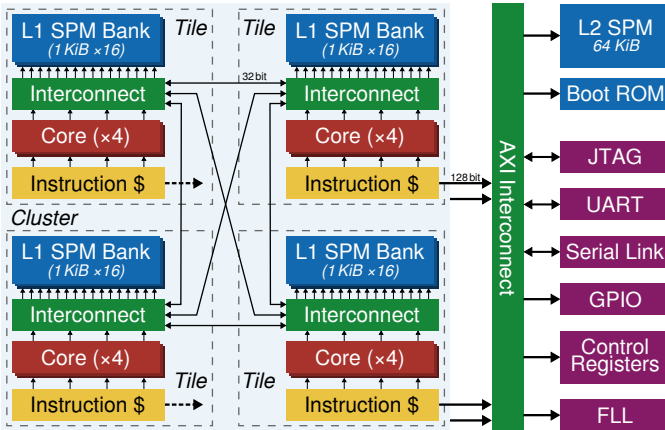


Fig. 1. Block diagram of the MinPool SoC featuring four tiles with four cores and 16 SPM banks each.

¹<https://github.com/pulp-platform/mempool>

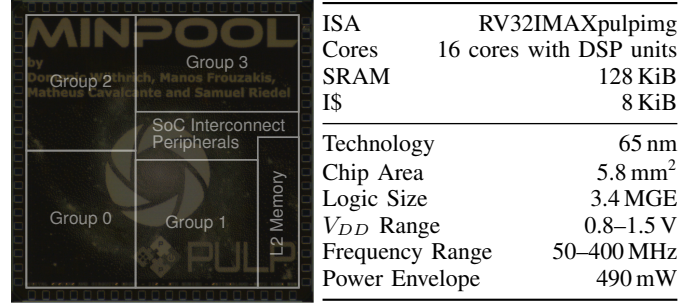


Fig. 2. Annotated die shot of MinPool with key metrics.

III. IMPLEMENTATION

MinPool is designed for low-power edge computing. Therefore, it is implemented in TSMC’s low-leakage 65 nm technology and consists of a smaller MemPool cluster with 16 cores. They are distributed into 4 tiles with 4 cores each, with each tile containing 16 KiB of SPM distributed across 16 banks and 2 KiB of L1 instruction cache, as shown in Fig. 1. Hence, the cluster consists of 4 groups with one tile each and a total of 64 KiB of L1 SPM and 8 KiB of L1 instruction cache. MinPool implements 64 KiB of L2 memory, and the AXI crossbar on the SoC level was configured to be 128 bit wide, leading to 16 B/cycle of L2 bandwidth. MinPool contains a JTAG debugging module, a UART interface, a serial link, and GPIOs for off-chip communication, such as a camera. MinPool leverages the MemPool hierarchical interconnect to ensure a short critical path, thereby achieving high operating frequency without compromising energy efficiency.

We taped out MinPool in TSMC’s 65 nm technology with a 9-metal stack. Figure 2 shows a die shot and the floorplan of MinPool. The chip has an area of 2.4 mm×2.4 mm, where 2.1 mm×2.1 mm are used for logic, which corresponds to 3.4 MGE. It is packaged in a QFN56 package featuring 48 IO and 8 power pins. Its pad supply voltage lies at 2.5 V, while the whole logic area shares one power domain with a nominal voltage of 1.2 V. MinPool targets a frequency of 200 MHz in the worst-case corner (SS/1.08 V/125 °C) and was synthesized with Synopsys DC Compiler 2019.03 and place-and-routed with Cadence Innovus 19.10.

IV. EVALUATION

We evaluate MinPool on the Advantest SOCV93000 ASIC tester, which allows precise measurements at specified voltages and frequencies. For benchmarks, we implemented various kernels and applications targeting DSP and image processing. We choose the largest problem size that still fits into L1 for each of the following kernels, and we evaluate a version with and without our custom DSP extension:

matmul: The multiplication of two 64×64 matrices.

2dconv: The convolution of a 64×96 image with a 3×3 filter.

dct: The 2D discrete cosine transform (DCT) from the JPEG algorithm. It operates on 8×8 blocks on a 64×96 image.

axpy: The $\alpha\vec{x} + \vec{y}$ on two vectors of 6144 elements.

dotp: The dot product $\vec{x} \cdot \vec{y}$ of two vectors of 6144 elements.

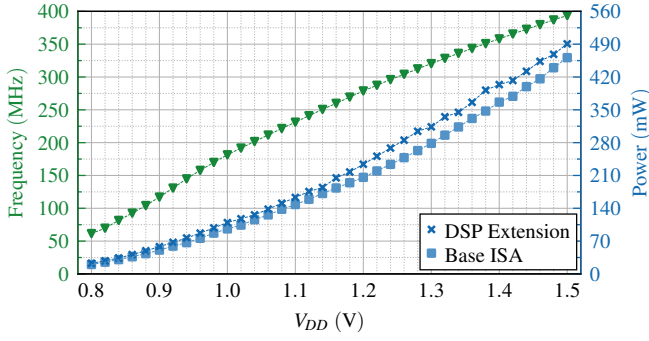


Fig. 3. Maximum operating frequency and power consumption of a 32-bit *matmul* as a function of the core supply voltage V_{DD} .

V. RESULTS

We determine MinPool’s maximum frequency over the range of 0.8 V to 1.5 V, as shown in Fig. 3. The frequency increases linearly with the voltage up to a maximum of 400 MHz at 1.5 V, which corresponds to only 30 fanout-of-4 gate delays. We also show the power consumption for each voltage point while running a 32-bit *matmul* at the corresponding maximum frequency. Even at the highest voltage and frequency, MinPool only consumes 490 mW. The lightweight Snitch cores and the pipelined interconnect allow MinPool to reach a very high frequency at very little power, considering its technology node.

To evaluate MinPool’s application performance, we measure all our kernels’ runtime, performance, and energy efficiency. We define an operation as either a 32-bit multiplication or addition. Hence, a MAC is two operations, and MinPool’s maximum theoretic throughput is 32 OP/cycle. This paper focuses on 32-bit operations and does not consider MinPool’s SIMD capabilities for lower precisions. Table I shows the performance of all kernels with MinPool’s DSP extension enabled. MinPool consistently achieves IPCs very close to 1, meaning there are only very few architectural stalls. This translates into a very high DSP unit utilization, considering that MinPool implements a load-store architecture. Even the memory-bound kernels are close to reaching their optimal DSP unit utilization, considering the number of loads and stores they must manage. This shows that the MemPool architecture, with its NUMA interconnect and DSP unit, can achieve very high utilization without being limited by the latencies. In fact, MemPool is designed to scale while keeping the latencies low and hiding them, enabling high performance with a high core count.

To visualize the benefit of the DSP extension, we show all our kernels’ performance and energy efficiency at different voltage points and the corresponding maximum achievable frequency in Fig. 4. MinPool achieves a peak performance of 9 GOPS when computing a *2dconv* at 1.5 V and 400 MHz. The most energy-efficient point is at 0.8 V and 60 MHz, where we reach 65 GOPS/W. For the *2dconv*, the DSP extension improves the peak performance by $3.7\times$ and energy efficiency by $3.5\times$ since it allows implementing a kernel with more data reuse by fusing operations and freeing up registers. The other kernels

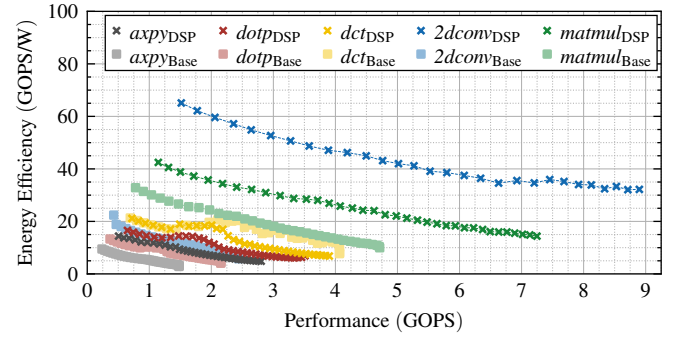


Fig. 4. Performance and energy efficiency of MinPool at the maximum frequency supported over a voltage sweep.

TABLE I
PERFORMANCE AND UTILIZATION OF KERNELS WITH THE DSP EXTENSION.

Kernel	Performance (OP/cycle)	IPC	DSP Util.	Peak Performance (GOPS)	Peak Efficiency (GOPS/W)
<i>matmul</i>	18.3	0.92	57%	7.2@400 MHz	43@0.8 V
<i>2dconv</i>	22.5	0.94	70%	9.0@400 MHz	65@0.8 V
<i>dct</i>	10.4	0.93	66%	3.9@400 MHz	21@0.8 V
<i>axpy</i>	7.3	0.98	23%	2.8@400 MHz	14@0.8 V
<i>dotp</i>	9.3	0.97	29%	3.5@400 MHz	17@0.8 V

also benefit from the DSP instructions, consistently gaining between 50% to 90% in performance and 30% to 55% in energy efficiency. Most of those gains come from fusing instructions like the MAC, where the pipelined implementation and latency tolerance guarantee a high frequency and throughput. The only exception is the *dct* kernel. Since it is mostly additions and subtractions, its benefits of the DSP extension are limited.

We illustrate MinPool’s capability of running full image processing pipelines by running a histogram equalization algorithm and Bayer pattern demosaicing with bilinear interpolation. The debayering achieves a performance of up to 380 Mpixel/s at 400 MHz or 2.7 Gpixel/s/W at 60 MHz. Despite its sequential parts, the histogram equalization achieves a performance of 262 Mpixel/s at 400 MHz or 2 Gpixel/s/W at 60 MHz.

VI. RELATED WORK

Table II summarizes the state-of-the-art (SoA) shared-memory cluster chips from the past years. While all related work features 8 to 10 cores, MinPool is the only chip that enables 16 cores to share a low-latency L1 memory. This is due to the MemPool architecture’s hierarchical L1 interconnect and lightweight cores. In contrast to MemPool, other work uses more complex cores with deeper pipeline stages and without support for outstanding instructions. Consequently, they must rely on a fully connected interconnect between all cores and banks since they cannot tolerate congestion or latency. However, this interconnect limits scaling. MemPool can trade off some bandwidth for scalability and, by hiding the latency of consequential conflicts and pipeline stages, keep performance unaffected while the design can run at a higher frequency.

The additional cores and efficient architecture are noticeable in the performance. MinPool’s *matmul* performance outper-

TABLE II

METRICS AND RESULTS OF MINPOOL COMPARED TO STATE-OF-THE-ART. THE 32-BIT MATMUL IS USED FOR COMPARING PERFORMANCE BETWEEN CHIPS.

	Thestral [7]	Vega [8]	Mr. Wolf [9]	Darkside [10]	MinPool
Year	2021	2022	2018	2022	2023
ISA	RV32IMAFD	RV32IMFXpulp	RV32IMFX	RV32IMFXpulpNN2	RV32IMAXpulpimg
Technology	22 nm	22 nm	40 nm	65 nm	65 nm
Chip Area	1.6 mm ²	12 mm ²	10 mm ²	12 mm ²	5.8 mm ²
Voltage	0.6 V to 0.9 V	0.5 V to 0.8 V	0.8 V to 1.1 V	0.75 V to 1.2 V	0.8 V to 1.5 V
Max Frequency	910 MHz	450 MHz	450 MHz	290 MHz	400 MHz
#Cores	10	10	1 + 8	8	16
Int formats	32 bit	8, 16, 32 bit	8, 16, 32 bit	8, 16, 32 bit	8, 16, 32 bit
Peak Performance ¹ (Int32 ²)	6.8 GOPS	4 GOPS	1.7 GOPS	1.2 GOPS	7.2 GOPS
Peak Efficiency ¹ (Int32 ²)	-	177 GOPS/W	30 GOPS/W	12 GOPS/W	43 GOPS/W
DSP utilization	37%	44%	30%	26%	57%
Normalized Performance	7.5 OP/cycle	8.9 OP/cycle	3.8 OP/cycle	4.1 OP/cycle	18.3 OP/cycle
Normalized Efficiency	-	44 GOPS/W	19 GOPS/W	6.75 GOPS/W	28.7 GOPS/W
Normalized Area	7.9 MGE	60.3 MGE	13.9 MGE	9.4 MGE	3.4 MGE
Normalized Area Efficiency	0.95 OP/cycle/MGE	0.15 OP/cycle/MGE	0.27 OP/cycle/MGE	0.44 OP/cycle/MGE	5.38 OP/cycle/MGE

1. The performance of the *matmul* kernel is used for comparison, even if the *2dconv* kernel achieves better results. 2. One OP denotes one 32-bit addition or multiplication.

forms all chips by a factor of 1.1–6×. It even outperforms Thestral [7], which is implemented in a newer technology and runs at 2.5× the frequency. Similarly, MinPool achieves high energy efficiency and is up to 3.6× more efficient than similar architectures using the same technology. Only Vega [8] is more efficient due to its advanced technology node.

Taking the technology differences into account, we provide normalized, voltage-independent numbers. Performance is normalized by dividing by the frequency to get OP/cycle. We scale the efficiency numbers by the voltage with the following formula since the power is proportional to the square of V_{DD} .

$$\text{NormEff.} = \text{PeakEff}(V_{DD}) \times V_{DD}^2 \quad (1)$$

Comparing the normalized numbers makes the benefit of the MemPool architecture even more apparent because it uses the least advanced technology. With a DSP unit utilization of 57%, MinPool spends most of its time computing, which leads to a normalized performance that is 2.1–4.8× better than the SoA. Similarly, MinPool’s energy efficiency can compete even with Vega when we compensate for the technology, and it outperforms the others by 1.5–4.3×. Finally, due to its small cores and high performance, MinPool is about 5.7–36× more area efficient than the SoA.

VII. CONCLUSION

We presented MinPool, a low-power image processor for always-on image processing derived from the MemPool architecture. It shows the advantage of combining lightweight RISC-V cores that can tolerate outstanding transactions with a pipelined, hierarchical, NUMA L1 interconnect, which trades off some bandwidth and latency for scalability and high clock speeds. Similarly, these cores can take advantage of pipelined, specialized DSP units tuned for image processing to maximize performance while sustaining high IPC counts. Furthermore, the scalable architecture allows MinPool to pack twice as many cores as SoA. MinPool outperforms related work by up to 6× while remaining energy efficient, being up to 3.6× more energy

efficient than designs in the same technology. We implement key image processing kernels and pipelines, which achieve performances of up to 380 Mpixel/s, showing the potential of NUMA clusters for ultra-low power image processing.

ACKNOWLEDGMENT

This work is funded in part by the COREnext project supported by the EU Horizon Europe research and innovation programme under grant agreement No. 101092598.

REFERENCES

- [1] S. Riedel, M. Cavalcante, R. Andri, and L. Benini, “MemPool: A scalable manycore architecture with a low-latency shared L1 memory,” *IEEE Trans. Comput.*, 2023, early access.
- [2] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, 6th ed. San Francisco, CA, USA: Morgan Kaufmann, 2017.
- [3] Bitcraze, “AI-deck 1.1 – Bitcraze store,” 2023. [Online]. Available: <https://store.bitcraze.io/products/ai-deck-1-1>
- [4] NVIDIA Corporation, “NVIDIA H100 tensor core GPU architecture,” NVIDIA Corp., Tech. Rep., 2022. [Online]. Available: <https://www.nvidia.com/en-us/data-center/h100/>
- [5] F. Zaruba, F. Schuiki, T. Hoefler, and L. Benini, “Snitch: A tiny pseudo dual-issue processor for area and energy efficient execution of floating-point intensive workloads,” *IEEE Trans. Comput.*, vol. 70, no. 11, pp. 1845–1860, Feb. 2021.
- [6] S. Mazzola, S. Riedel, M. Cavalcante, L. Benini, and A. Macii, “ISA extensions in the Snitch processor for signal processing,” *Politecnico di Torino, Tech. Rep.*, Apr. 2021.
- [7] T. Benz, L. Bertaccini, F. Zaruba, F. Schuiki, F. K. Gurkaynak, and L. Benini, “A 10-core soc with 20 fine-grain power domains for energy-proportional data-parallel processing over a wide voltage and temperature range,” *ESSCIRC 2021 - IEEE 47th Eur. Solid State Circuits Conf. Proc.*, pp. 263–266, Sep. 2021.
- [8] 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,” *IEEE J. Solid-State Circuits*, vol. 57, no. 1, pp. 127–139, Jan. 2022.
- [9] A. Pullini, D. Rossi, I. Loi, A. Di Mauro, and L. Benini, “Mr. wolf: A 1 gflop/s energy-proportional parallel ultra low power soc for iot edge processing,” *ESSCIRC 2018 - IEEE 44th Eur. Solid State Circuits Conf.*, pp. 130–133, Oct. 2018.
- [10] A. Garofalo *et al.*, “Darkside: 2.6gflops, 8.7mw heterogeneous risc-v cluster for extreme-edge on-chip dnn inference and training,” *ESSCIRC 2022 - IEEE 48th Eur. Solid State Circuits Conf. Proc.*, pp. 273–276, 2022.