

Sargantana: An Open RISC-V Processor for HW/SW Co-Design of Domain-Specific Accelerators

Narcís Rodas^{1,*}, Max Doblas¹, Víctor Soria-Pardos¹, Santiago Marco-Sola^{1,2} and Miquel Moretó^{1,3}

¹Barcelona Supercomputing Center, Spain.

²Department of Computer Science, Universitat Politècnica de Catalunya, Spain.

³Departament d'Arquitectura de Computadors, Universitat Politècnica de Catalunya, Spain.

Abstract

Domain-specific computing systems are becoming essential to meet the performance and efficiency demands of emerging workloads such as AI and genomics. However, developing and integrating full-fledged RTL-level accelerators into modern processors remains a challenging and time-consuming process. Existing platforms either focus on complex, high-performance cores with limited flexibility or provide simple educational environments that lack the ability to evaluate accelerators under realistic conditions. In this paper, we present Sargantana, an open-source RISC-V Linux-capable processor designed as a research platform for agile design and development of domain-specific accelerators. Its modular architecture and simple integration path lower the barrier for researchers to prototype and evaluate new extensions. We demonstrate its capabilities through three successful accelerator case studies: Mix-GEMM for deep neural network inference, GMX for bioinformatics sequence alignment, and WFAasic for genome sequence analysis. Together, these case studies show how Sargantana helps bridge the gap between academic exploration and real-world domain-specific applications. Find the code in the Sargantana GitHub repository <https://github.com/bsc-loc/sargantana/tree/main>

1 Introduction

Over the past decades, the computing industry has relied on Moore's Law and Dennard scaling to sustain performance growth through increased transistor density and reduced power per device. As process scaling approaches its physical and power limits, power density has emerged as a dominant constraint, and performance improvements from general-purpose processors have become limited. To overcome this challenge, computer architects have increasingly turned toward domain-specific architectures (DSAs) that tailor hardware to the characteristics of particular applications, achieving higher efficiency and performance within limited power and area budgets.

Domain-specific accelerators have become key to sustaining performance and efficiency in modern HPC [1]. Emerging applications in fields such as Artificial Intelligence (AI) and genomics demand performance improvements that general-purpose processors struggle to provide. However, developing RTL-level prototypes of accelerators and integrating them into modern processor cores is a complex and time-consuming task. Existing research and academic platforms either offer complex high-performance cores that make accelerator integration challenging or simple teaching environments that are unable to perform realistic evaluation. This lack of suitable research platforms limits research in domain-specific hardware and software co-design.

RISC-V is an open-source Instruction Set Architecture (ISA) that removes licensing barriers for research and has seen rapid adoption in both academic and commercial domains. Its modular base-plus-extensions design and support for custom instructions enable flexible exploration of new architectural concepts and accelerator interfaces. By exposing a well-defined extension space, RISC-V allows researchers to prototype domain-specific instructions, evaluate tightly coupled accelerators integrated within the pipeline, or attach loosely coupled co-processors through

standardized interfaces such as AXI. In contrast to proprietary ISAs, this openness simplifies toolchain integration, verification, and FPGA prototyping, providing a practical framework for hardware/software co-design. These features make RISC-V an ideal basis for developing, testing, and benchmarking novel domain-specific accelerators (DSAs).

In this work, we present Sargantana [2, 3], an open-source RISC-V processor designed as a research platform for developing domain-specific accelerators. Sargantana is a Linux-capable, edge-class core that combines simplicity of design with modularity, enabling easy integration of new hardware ISA extensions and accelerators. We illustrate its capabilities through three case studies: (1) Mix-GEMM [4, 5] for deep neural network (DNN) inference, (2) GMX [6] for efficient sequence alignment, and (3) WFAasic [7] for genome analysis. These examples demonstrate how Sargantana lowers the barrier for prototyping accelerators and serves as a practical bridge between academic exploration and domain-specific real-world applications.

2 Sargantana RISC-V Core

Sargantana [2, 3] is an open-source [8], Linux-capable RISC-V core designed for research and agile prototyping. It implements the RV64GBV ISA and features a 7-stage, in-order, single-issue pipeline with register renaming and out-of-order writeback. The core integrates a 128-bit SIMD unit, a 16 KiB L1 instruction cache, and a 32 KiB high-performance HPDCache [9] data cache. Synthesized at 22 nm, Sargantana reaches 1.2 GHz and achieves 2.44 Coremark/MHz in the Coremark benchmark.

2.1 Custom Instructions in Sargantana

Although Sargantana does not match the peak performance of mainstream out-of-order RISC-V cores, its simplicity and modular design make it well-suited for developing

* Corresponding author: narcis.rodaquirola@bsc.es

Received: August 26, 2025

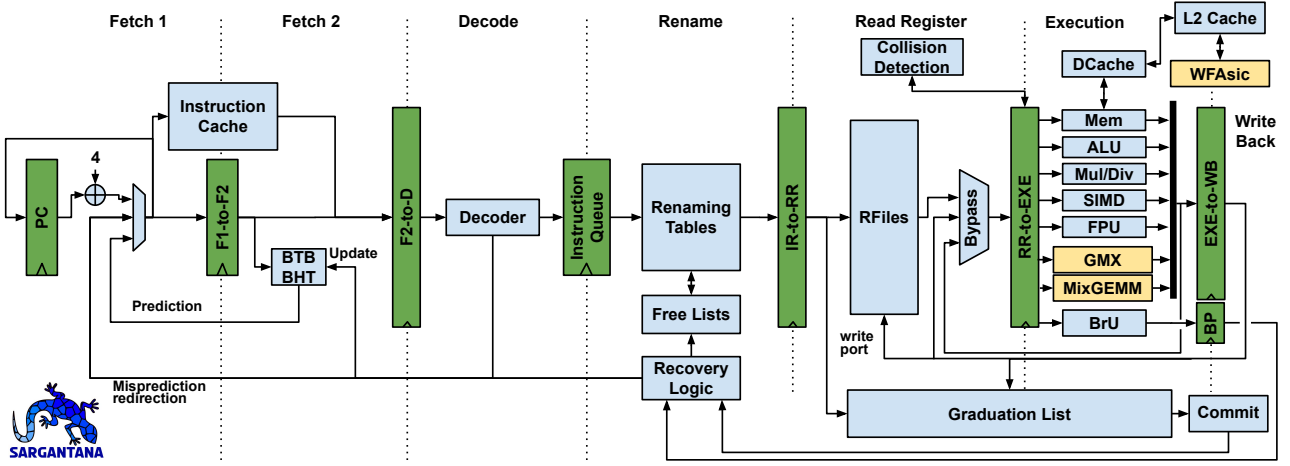


Figure 1: Microarchitecture of the Sargantana processor. Accelerators (GMX, Mix-GEMM, and WFAasic) are highlighted in yellow.

domain-specific accelerators. Sargantana pipeline enables fast custom instruction extension, requiring only minor modifications. Furthermore, some of the advanced features integrated into Sargantana enable high performance out of the box. For instance, register renaming and out-of-order commit enable the integration of multicycle instructions without performance loss or the need for hand-made code optimization. Moreover, memory-intensive applications can exploit the load-store unit to mask the latency of memory access and enable the pipelining of custom instructions, achieving high throughput with minimal effort.

The process of integrating a new custom instruction in Sargantana is straightforward. Every new instruction requires the modification of the decode and execution stages. For tightly coupled accelerators that introduce single-cycle instructions, updating these two modules is enough. Additionally, multi-cycle instructions with fixed latency need to modify the core’s scoreboard mechanism. As a result, custom instructions can be integrated by modifying only three modules at most. This combination of simplicity and modularity makes Sargantana a practical platform for agile prototyping and evaluation of new domain-specific accelerator designs.

2.2 Sargantana Interface with Accelerators

Notwithstanding, loosely coupled accelerators may introduce complex data flow behavior or issue memory access that cannot be integrated within the pipeline. Hence, such accelerators must be integrated within the memory hierarchy or through I/O interfaces such as AXI [10]. In the case of Sargantana, we integrate most accelerators via OpenPiton [11]. OpenPiton not only offers a simple way of integrating accelerators, but also allows scaling designs to multi-core configurations of Sargantana [12]. Furthermore, OpenPiton enables accelerators to access the downstream levels of the cache hierarchy directly (i.e., level 2 and 3). Such levels of the cache hierarchy have larger storage and are more difficult to be polluted by the integrated accelerator.

As a result, there are three possible ways of communicating the core with the accelerator: (i) using memory-mapped registers, (ii) using custom instructions, or (iii) through the Cohort interface [13]. The first option is the most straight-

forward because it does not require modifying the core, but uses memory accesses that can be expensive for frequent communication. The second option is more direct, but requires creating a specific interface that connects the core and the engine. Finally, the third option utilizes common concepts, such as software shared-memory queues, to communicate between the core and the accelerator.

3 Sargantana as Accelerator Integration Platform

Sargantana was developed within the framework of the DRAC project [14] that aimed to design RISC-V-based accelerators. As part of the DRAC project, several ISA Extensions and Accelerators have been developed, verified, and evaluated using Sargantana as the development platform. Among the ISA extensions integrated in Sargantana, we find GMX [6] for efficient sequence alignment, SMX [15], Mix-GEMM [4, 5] for deep neural network (DNN) inference, and CAMP [16] for matrix multiplication.

Similarly, we find an extensive set of hardware accelerators that have been integrated along with Sargantana: SAURIA [17] for automotive applications, WFAasic [7] for genomic analysis, Classic McEliece [18] for postquantum cryptography, Picos [19, 20] for task scheduling, and the TMU [21, 22] for sparse tensor traversal.

In this work, we illustrate three of these use cases to demonstrate the capability of Sargantana as a development platform for domain-specific accelerators. The selected case studies are: (1) Mix-GEMM, (2) GMX, and (3) WFAasic.

3.1 GMX for Sequence Alignment

Sequence alignment identifies similarities and differences between a pair of sequences and is a fundamental problem in genomics [23, 24, 25, 26] and other fields [27, 28]. Classical dynamic programming (DP) algorithms such as Needleman-Wunsch [29] and Smith-Waterman [30] scale quadratically in both execution time and memory, quickly becoming bottlenecks as sequencing technologies generate longer and noisier reads. While SIMD [31, 32, 33], GPU [34, 35], FPGA [36, 37, 38], PIM [39], and DSAs [40, 41] have been explored, general-purpose processors still struggle due to high memory bandwidth requirements, irregular

data dependencies, and limited scalability across diverse workloads.

GMX (Genome alignMent eXtensions) [6] is a set of RISC-V ISA extensions designed to accelerate DP-based sequence alignment. It introduces specialized instructions (`gmx.v`, `gmx.h`, and `gmx.tb`) that compute entire tiles of the DP matrix and perform traceback in hardware. By operating on tiles, GMX reduces the instruction count quadratically with tile size and minimizes the memory footprint by storing only tile-edge values. This co-design enables the efficient and flexible acceleration of multiple alignment strategies and algorithms, integrating seamlessly into standard software pipelines.

GMX has been integrated into the Sargantana core as an ISA extension, introducing three new instructions and five new CSRs. First, to integrate GMX into the Sargantana core, the decode stage is extended to recognize the new instructions. Then, the execution stage is modified to include a new functional unit that implements the GMX instructions. This unit contains a systolic array for computing the DP tiles and a traceback engine for reconstructing the optimal alignment path. The functional unit receives the two source registers from the register file in the same way as other ALU operations. The results are written back to the register file using the existing writeback ports. In this case, the `gmx.v` and `gmx.h` instructions use the ALU writeback port since both have one-cycle latency, while the `gmx.tb` instruction uses the multiplier writeback port as it has a multi-cycle latency. Although Sargantana has multi-cycle instructions, it does not require any additional logic to handle them, as the core’s register renaming and graduation list mechanism already supports the reordering of the instructions. Finally, five new Control and Status Registers (CSRs) are added to configure the GMX unit, including parameters such as tile size, scoring scheme, and gap penalties. Since the new GMX CSRs follow the same specifications as the existing ones, the core’s CSR read/write logic is updated to expand access to these new registers.

When integrated in the Sargantana core, GMX achieved significant performance improvements over the baselines. GMX improved throughput by up to $45\times$ compared to state-of-the-art software implementations, with even higher benefits when aligning long sequences due to reduced memory pressure. Compared to DSAs like GenASM [41] and Darwin [40], a single GMX-enabled Sargantana core achieved $1.3\text{--}1.9\times$ better throughput than GenASM and $7.2\text{--}16.2\times$ better than Darwin, while requiring an order of magnitude less additional area.

3.2 Mix-GEMM for DNN Inference

Deep Neural Networks (DNNs) have become the standard approach for computer vision [42] and artificial intelligence [43] tasks. However, their deployment on edge devices is limited by stringent constraints in energy, area, and memory. The computation of DNN layers is dominated by General Matrix Multiplication (GEMM), which must be executed efficiently to enable low-latency inference. For that, quantization techniques [44, 45] reduce memory and compute requirements by representing activations and weights with narrow integer precisions (8- to 2-bit), often in mixed-precision form. While Quantization-Aware Training (QAT) [46] enables high accuracy even with sub-byte data sizes, current CPUs and ISAs do not fully support such granularities. Most architectures lack support below

8-bit or require costly decompression and bit manipulation, leaving a gap for efficient execution of quantized DNNs on edge processors.

Mix-GEMM [4, 5] addresses this challenge by co-designing hardware and software to efficiently compute quantized GEMM kernels at arbitrary precisions from 8-down to 2-bit, including mixed-precision activations and weights. The key novelty of Mix-GEMM is its ability to scale performance with decreasing data sizes, while maintaining high accuracy through integration with QAT-trained models. At its core, Mix-GEMM introduces a hardware microarchitecture called the μ -engine, built on the binary segmentation technique, which reuses the existing scalar multiplier to compute multiple Multiply-Accumulate (MAC) operations per cycle with negligible area overhead.

Mix-GEMM is integrated into Sargantana as a new ISA extension. Similar to GMX, Mix-GEMM integrates three new RISC-V instructions (`bs.set`, `bs.ip`, `bs.get`) that configure precision parameters, issue inner-product computations, and retrieve results. For their integration, the decode stage is extended to recognize these new instructions. Moreover, the execution stage is modified to include a new functional unit that implements the μ -engine. The new unit receives the two source registers from the register file in the same way as other ALU operations. In this case, the μ -engine is connected to a multiplexed port of the multiplier unit, as it reuses the existing scalar multiplier to perform the MAC operations. The results are written back to the register file using the existing ALU writeback port, as the extraction operation has a one-cycle latency. Thus, the integration of Mix-GEMM into Sargantana requires minimal changes to the core, mainly in the decode and execution stages, demonstrating the core’s modularity and suitability for accelerator prototyping.

To maximize efficiency, the Mix-GEMM software library extends the BLIS framework to handle compressed narrow-precision data, maintaining cache-friendly data movement across the memory hierarchy. Activations and weights are kept compressed in main memory, reducing bandwidth requirements and enabling deployment of larger DNN models within edge constraints. An internal Accumulator Memory (AccMem) within the μ -engine stores intermediate results locally, improving reuse and reducing instruction count. The design supports both symmetric and asymmetric quantization configurations, seamlessly integrating with ONNX Runtime for deployment.

Mix-GEMM is evaluated on representative CNNs (AlexNet [47], VGG-16 [48], ResNet-18 [49], MobileNet-V1 [50]). In these scenarios, Mix-GEMM achieves a throughput of between 4.8 and 13.6 GOPS, with energy efficiency of up to 1.3 TOPS/W. Compared to software baselines (OpenBLAS on a commercial RISC-V core), Mix-GEMM improves performance by $5.3\times\text{--}15.1\times$, while maintaining accuracy losses below 1.5% for data sizes $\geq 4\text{-bit}$.

3.3 WFAasic for Genome Analysis

In modern genome sequence analysis, sequence alignment remains a fundamental problem and a major computational bottleneck. While GMX achieves ultrafast edit distance computation, it comes with some limitations. First, it only supports edit distance, which is insufficient for many real-world applications that require gap-affine scoring models. Second, its tile granularity reduces effectiveness when aligning sequences with very low error rates ($<1\%$). In these scenarios, the Wavefront Alignment (WFA) [32] algo-

rithm is better suited, as it supports gap-affine scoring and scales with the error rates ($O(ns)$) rather than quadratically with sequence length ($O(n^2)$).

WFAasic [7] is the first ASIC accelerator for exact pairwise alignment based on the WFA algorithm, integrated into the Sargantana core. Unlike GMX’s tile-based DP approach, WFAasic implements the complete WFA algorithm to align two sequences in hardware, enabling efficient gap-affine alignment of long sequences. WFAasic supports DNA sequences of up to 10 Kbps with error rates up to 10%, providing scalability for third-generation sequencing technologies. By exploiting the diagonal sparsity of WFA, the accelerator achieves high performance while drastically reducing the number of computed DP cells.

The WFAasic accelerator is integrated into the Sargantana RISC-V CPU core as a loosely coupled accelerator via an AXI-based interface. Sargantana’s memory hierarchy includes multiple AXI crossbars on both the I/O and memory channels, facilitating seamless integration. The CPU configures WFAasic through memory-mapped registers on the I/O channel, setting parameters such as maximum read length, DMA transfer addresses, and backtrace enable flags. Once configured, the accelerator fetches input sequences directly from main memory using the AXI interface, executes the extend and compute operations in hardware, and writes the alignment results back to memory. To address on-chip storage limitations, WFAasic generates backtrace data on-chip but streams it directly to main memory, where the CPU completes the traceback reconstruction. This design effectively reduces memory bottlenecks while ensuring correctness and scalability.

When evaluated aligning sequence pairs of up to 10 Kbp, WFAasic achieves speedups of up to $1076\times$ over a scalar CPU WFA implementation, and up to $344\times$ even when including full backtrace computation. Compared to GPU and FPGA implementations, WFAasic demonstrates both superior energy efficiency and up to $9.7\times$ higher GCUPS/mm², making it a highly competitive solution for accelerating long-read alignment in genome analysis pipelines.

4 Area and power results

A comprehensive experimental evaluation was conducted on the Mix-GEMM, GMX, and WFAasic accelerators, which were fully implemented in RTL and integrated into the Sargantana SoC. The SoC infrastructure offers a simple and modular approach to extend the design, emulate it on an FPGA, and perform physical synthesis in 22 nm or 7 nm. For instance, Figure 2 shows the physical layout of the Sargantana SoC with Mix-GEMM highlighted in green. Our setup allows each accelerator to be evaluated by running its corresponding benchmarks on the FPGA and analyzing the resulting power and area metrics from synthesis.

For the FPGA emulation, Mix-GEMM was done on a Xilinx Alveo U55C, while GMX and WFAasic were performed in a Xilinx Alveo U280. The physical synthesis of all three accelerators was done using GlobalFoundries 22nm technology node with a target frequency of 1 GHz in the typical corner. In particular, Cadence Genus was used for logical synthesis, and Cadence Innovus was used for place-and-route.

Table 1 shows a summary of the area and power results from the core and the accelerators. Mix-GEMM occupies an area of 0.01 mm² and has an average power consumption

Table 1: Area and power of the Sargantana core (including L1 and L2 caches) and the Mix-GEMM, GMX, and WFAasic accelerators.

Design	Area (mm ²)	Power (mW)
Sargantana	1.24	397.44
Mix-GEMM 1-issue	0.01	3.60
Mix-GEMM 2-issue	0.03	8.60
GMX	0.02	8.47
WFAasic	1.60	312.00

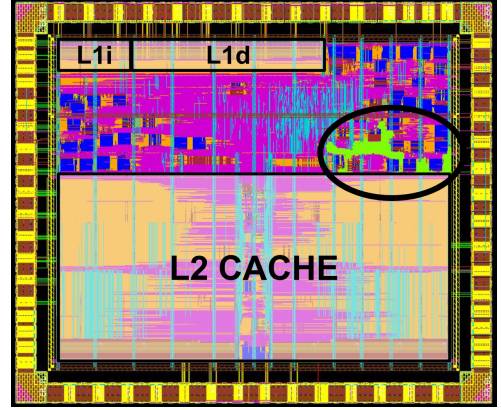


Figure 2: Physical layout of the Sargantana SoC with Mix-GEMM integrated. Source: [4]

of 3.60 mW in its single-issue configuration (0.03 mm² and 8.60 mW, respectively, in the dual-issue version). The area of the GMX accelerator is 0.02 mm², with an 8.47 mW power consumption. In contrast, WFAasic fits in 1.60 mm² and consumes 312.00 mW. Note the difference in results between the tightly-coupled accelerators (Mix-GEMM and GMX) and the loosely-coupled WFAasic. While the first two require a small area and power increase with respect to Sargantana, the latter shows values comparable to the full SoC.

5 Conclusions

Moore’s Law projected that transistor density in integrated circuits would double approximately every two years, while Dennard scaling assumed constant power density across generations. Together, these trends enabled decades of performance growth through transistor miniaturization. However, Dennard scaling has already broken down, and Moore’s Law is approaching its physical and economic limits. As a result, the focus has shifted toward specialized architectures that improve performance and efficiency for specific applications. Heterogeneous systems integrating DSAs have emerged as a key approach to sustain performance scaling in modern computing.

Nonetheless, integrating RTL implementations of DSA into modern high-performance cores is resource and time-consuming. To perform evaluation and early prototyping of new designs, researchers often require in-depth knowledge of the processor’s specific microarchitecture to integrate their work effectively. This challenge is particularly evident for tightly coupled accelerators, but it also affects loosely coupled co-processors, which must still manage

complex communication with the host CPU. Furthermore, modern processors continue to grow in complexity. Having an N-way out-of-order superscalar core with simultaneous multithreading requires complex logic to control the multiple threads that progress through the pipeline in the same cycle.

In this paper, we present Sargantana, an open-source RISC-V processor designed to facilitate research on domain-specific accelerators. One of the main design goals behind Sargantana is to maintain its simplicity and modularity while extending its features via RISC-V extensions. By integrating their accelerator in Sargantana, researchers can easily prototype their design by emulating it on an FPGA, which is more accurate than cycle-level simulators. It also allows them to obtain area and power results by performing physical synthesis in an advanced technology node. Because Sargantana is a simple in-order single-issue processor, it is easier to expand and reduce the verification effort needed to debug new features. Nevertheless, the performance achieved by the core cannot compete with the full-fledged out-of-order architectures present in modern HPC systems. A similar proposal is Cheshire [51], which also provides a lightweight RISC-V platform for integrating domain-specific accelerators, but focuses exclusively on loosely coupled designs, unlike Sargantana.

Sargantana has been validated using a diverse set of state-of-the-art domain-specific accelerators, demonstrating its flexibility and maturity as a research platform. Through the integration of Mix-GEMM, GMX, and WFAasic, we have shown that the architecture enables rapid prototyping, modular evaluation, and straightforward integration of both tightly and loosely coupled accelerators. These case studies also confirm that Sargantana is a complete and robust framework, supporting the full development flow from RTL design and FPGA emulation to physical synthesis and benchmarking, making it a practical and reliable platform for DSA research.

We believe that Sargantana will enable researchers to more easily test, prototype, and evaluate their designs in an environment that gives them meaningful results in terms of performance and efficiency.

Acknowledgments

This work has been partially funded by the Spanish Ministry of Science and Innovation MCIN/AEI/10.13039/501100011033 and FSE+ [grant number PID2019-107255GB-C21, PID2020-113614RB-C21, TED2021-132634A-I00, and PID2023-146511NB-I00], the Spanish Ministry of Economy, Industry, and Competitiveness through an FPU fellowship [grant number FPU20-04076 to M.D.], the Spanish Ministry for Digital Transformation and Public Service, and the European Union – NextGenerationEU through the Càtedra Chip UPC project [grant number TSI-069100-2023-0015], the Spanish Ministry for Digital Transformation and Public Service, within the framework of the Recovery, Transformation and Resilience Plan - NextGenerationEU [REGAGE22e00058408992], and the Generalitat de Catalunya GenCat-DIUIE (GRR) [grant numbers 2021-SGR-00763, and 2021-SGR-00574].

References

- [1] Mauricio Araya-Polo et al. “Assessing Accelerator-Based HPC Reverse Time Migration”. In: *IEEE Transactions on Parallel and Distributed Systems* 22.1 (2011), pp. 147–162. doi: 10.1109/TPDS.2010.144.
- [2] Víctor Soria-Pardos et al. “Sargantana: A 1 GHz+ in-order RISC-V processor with SIMD vector extensions in 22nm FD-SOI”. In: *2022 25th Euromicro Conference on Digital System Design (DSD)*. IEEE. 2022, pp. 254–261.
- [3] Max Doblas et al. “Sargantana: An Academic SoC RISC-V Processor in 22nm FDSOI Technology”. In: *2023 38th Conference on Design of Circuits and Integrated Systems (DCIS)*. IEEE. 2023, pp. 1–6.
- [4] Enrico Reggiani et al. “Mix-gemm: An efficient hws architecture for mixed-precision quantized deep neural networks inference on edge devices”. In: *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE. 2023, pp. 1085–1098.
- [5] Jordi Fornt et al. “Mix-GEMM: extending RISC-V CPUs for energy-efficient mixed-precision DNN inference using binary segmentation”. In: *IEEE Transactions on Computers* (2024).
- [6] Max Doblas et al. “Gmx: Instruction set extensions for fast, scalable, and efficient genome sequence alignment”. In: *Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture*. 2023, pp. 1466–1480.
- [7] Abbas Haghi et al. “WFAasic: a high-performance ASIC accelerator for DNA sequence alignment on a RISC-V SoC”. In: *Proceedings of the 52nd International Conference on Parallel Processing*. 2023, pp. 392–401.
- [8] *Sargantana GitHub repository*. Accessed: 23 Sep 2025 [Online] <https://github.com/bsc-locasargantana/tree/main>.
- [9] César Fuguet. “HPDcache: Open-source high-performance L1 data cache for RISC-V cores”. In: *Proceedings of the 20th ACM International Conference on Computing Frontiers*. 2023, pp. 377–378.
- [10] AXI AMBA. “Protocol specification”. In: *ARM, June* (2003).
- [11] Jonathan Balkind et al. “OpenPiton: An Open Source Manycore Research Framework”. In: *SIGARCH Comput. Archit. News* 44.2 (Mar. 2016), pp. 217–232. issn: 0163-5964. doi: 10.1145/2980024.2872414. url: <https://doi.org/10.1145/2980024.2872414>.
- [12] Noelia Oliete-Escuín et al. “REPTILES: Repeated Tiles of Sargantana, a RISC-V multicore based on OpenPiton”. In: *RISC-V Summit Europe* (2025).
- [13] Tianrui Wei et al. “Cohort: Software-Oriented Acceleration for Heterogeneous SoCs”. In: *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*. ASPLOS 2023. Vancouver, BC, Canada: Association for Computing Machinery, 2023, pp. 105–117. isbn: 9781450399180. doi: 10.1145/3582016.3582059. url: <https://doi.org/10.1145/3582016.3582059>.
- [14] Drac Project. *Designing RISC-V-based Accelerators for next generation Computers*. [Online; accessed 7-October-2025]. 2024. url: <https://drac.bsc.es/en>.

- [15] Max Doblas et al. "SMX: Heterogeneous Architecture for Universal Sequence Alignment Acceleration". In: *Proceedings of the 58th Annual IEEE/ACM International Symposium on Microarchitecture*. 2025. DOI: 10.1145/3725843.3756018.
- [16] Mohammadreza Esmali Nojehdeh et al. *Empowering Vector Architectures for ML: The CAMP Architecture for Matrix Multiplication*. 2025. arXiv: 2504.08137 [cs.AR]. URL: <https://arxiv.org/abs/2504.08137>.
- [17] Jordi Fornt et al. "An Energy-Efficient GeMM-Based Convolution Accelerator With On-the-Fly im2col". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 31.11 (2023), pp. 1874–1878. DOI: 10.1109/TVLSI.2023.3286122.
- [18] Ioannis-Vatistas Kostalampros et al. "Post-quantum cryptography acceleration for next generation computers". In: (2024).
- [19] Fahimeh Yazdanpanah et al. "Picos: A hardware runtime architecture support for OmpSs". In: *Future Generation Computer Systems* 53 (2015), pp. 130–139. ISSN: 0167-739X. DOI: <https://doi.org/10.1016/j.future.2014.12.010>. URL: <https://www.sciencedirect.com/science/article/pii/S0167739X14002702>.
- [20] Xubin Tan et al. "Picos, A Hardware Task-Dependence Manager for Task-Based Dataflow Programming Models". In: *2017 International Conference on High Performance Computing & Simulation (HPCS)*. 2017, pp. 878–880. DOI: 10.1109/HPCS.2017.134.
- [21] Marco Siracusa et al. "A Tensor Marshaling Unit for Sparse Tensor Algebra on General-Purpose Processors". In: *Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture*. MI-CRO '23. Toronto, ON, Canada: Association for Computing Machinery, 2023, pp. 1332–1346. ISBN: 9798400703294. DOI: 10.1145/3613424.3614284. URL: <https://doi.org/10.1145/3613424.3614284>.
- [22] Marco Siracusa et al. *Ember: A Compiler for Efficient Embedding Operations on Decoupled Access-Execute Architectures*. 2025. arXiv: 2504.09870 [cs.AR]. URL: <https://arxiv.org/abs/2504.09870>.
- [23] Heng Li. "Minimap2: pairwise alignment for nucleotide sequences". In: *Bioinformatics* 34.18 (2018), pp. 3094–3100.
- [24] Santiago Marco-Sola et al. "The GEM mapper: fast, accurate and versatile alignment by filtration". In: *Nature Methods* 9.12 (2012), pp. 1185–1188.
- [25] Santiago Marco-Sola and Paolo Ribeca. "Efficient Alignment of Illumina-Like High-Throughput Sequencing Reads with the GENomic Multi-tool (GEM) Mapper". In: *Current Protocols in Bioinformatics* 50.1 (2015), pp. 11–13.
- [26] Lorién López-Villellas et al. "RISC-V for genome data analysis: opportunities and challenges". In: *2023 38th Conference on Design of Circuits and Integrated Systems (DCIS)*. IEEE. 2023, pp. 1–6.
- [27] Maciej Janicki. "Optimizing the weighted sequence alignment algorithm for large-scale text similarity computation". In: *Proceedings of the 2nd International Workshop on Natural Language Processing for Digital Humanities*. 2022, pp. 96–100.
- [28] Gerhard Mehldau and Gene Myers. "A system for pattern matching applications on biosequences". In: *Bioinformatics* 9.3 (1993), pp. 299–314.
- [29] Saul B Needleman and Christian D Wunsch. "A general method applicable to the search for similarities in the amino acid sequence of two proteins". In: *Journal of molecular biology* 48.3 (1970), pp. 443–453.
- [30] Temple F Smith and Michael S Waterman. "Identification of common molecular subsequences". In: *Journal of molecular biology* 147.1 (1981), pp. 195–197.
- [31] Max Doblas et al. "QuickEd: high-performance exact sequence alignment based on bound-and-align". In: *Bioinformatics* 41.3 (2025), btaf112.
- [32] Santiago Marco-Sola et al. "Fast gap-affine pairwise alignment using the wavefront algorithm". In: *Bioinformatics* 37.4 (2021), pp. 456–463.
- [33] Lorién López-Villellas et al. "GenArchBench: A genomics benchmark suite for arm HPC processors". In: *Future Generation Computer Systems* 157 (2024), pp. 313–329.
- [34] Quim Aguado-Puig et al. "WFA-GPU: gap-affine pairwise read-alignment using GPUs". In: *Bioinformatics* 39.12 (Nov. 2023), btad701. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btad701. URL: <https://doi.org/10.1093/bioinformatics/btad701>.
- [35] Quim Aguado-Puig et al. "Accelerating edit-distance sequence alignment on GPU using the wavefront algorithm". In: *IEEE access* 10 (2022), pp. 63782–63796.
- [36] Abbas Haghi et al. "WFA-FPGA: An efficient accelerator of the wavefront algorithm for short and long read genomics alignment". In: *Future Generation Computer Systems* 149 (2023), pp. 39–58.
- [37] David Castells-Rufas et al. "FPGA acceleration of pre-alignment filters for short read mapping with HLS". In: *IEEE Access* 10 (2022), pp. 22079–22100.
- [38] David Castells-Rufas et al. "OpenCL-based FPGA accelerator for semi-global approximate string matching using diagonal bit-vectors". In: *2021 31st International Conference on Field-Programmable Logic and Applications (FPL)*. IEEE. 2021, pp. 174–178.
- [39] Alejandro Alonso-Marín et al. "BIMSA: accelerating long sequence alignment using processing-in-memory". In: *Bioinformatics* 40.11 (2024), btae631.
- [40] Yatish Turakhia, Gill Bejerano, and William J Dally. "Darwin: A genomics co-processor provides up to 15,000 x acceleration on long read assembly". In: *ACM SIGPLAN Notices* 53.2 (2018), pp. 199–213.
- [41] Damla Senol Cali et al. "Genasm: A high-performance, low-power approximate string matching acceleration framework for genome sequence analysis". In: *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE. 2020, pp. 951–966.
- [42] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.
- [43] Ali Bou Nassif et al. "Speech recognition using deep neural networks: A systematic review". In: *IEEE access* 7 (2019), pp. 19143–19165.

- [44] Yuhang Li et al. "Brecq: Pushing the limit of post-training quantization by block reconstruction". In: *arXiv preprint arXiv:2102.05426* (2021).
- [45] Yoni Choukroun et al. "Low-bit quantization of neural networks for efficient inference". In: *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE. 2019, pp. 3009–3018.
- [46] Xiaotian Zhao, Ruge Xu, and Xinfei Guo. "Post-training quantization or quantization-aware training? That is the question". In: *2023 China Semiconductor Technology International Conference (CSTIC)*. IEEE. 2023, pp. 1–3.
- [47] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012).
- [48] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).
- [49] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.
- [50] Andrew G Howard et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861* (2017).
- [51] Alessandro Ottaviano et al. "Cheshire: A lightweight, linux-capable risc-v host platform for domain-specific accelerator plug-in". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 70.10 (2023), pp. 3777–3781.