Proceedings of CONF-SEML 2026 Symposium: Computational Analysis and Modeling in Complex Intelligent Systems
DOI: 10.54254/2755-2721/2026.GU33410

# *RISC-V Hardware-Software Co-design for Edge AI Acceleration*

**Zhengyan Wang[1], Zhihan Wang[2*], Lancheng Yin[3]**

*[1]School of Electronic Science and Engineering, Nanjing University, Nanjing, China*
*[2]School of Communication & Information Engineering, Shanghai University, Shanghai, China*
*[3]BELL Honors School, Nanjing University of Posts and Telecommunications, Nanjing, China*
**Corresponding Author. Email: wangzhihan@shu.edu.cn*

***Abstract.*** Because of the development of edge intelligence, general-purpose processors have had difficulty meeting the computing power requirements under the constraints of power consumption and storage. Meanwhile, RISC-V, with its open-source, modular architecture, offers a clear path to break through the traditional barriers between software and hardware. Therefore, this article gives a systematic review of RISC-V hardware-software co-design technology for edge AI acceleration. The paper gives a clear and systematic presentation of the three main collaborations: hardware-led (standard extensions, custom instructions, heterogeneous NPU), software-led (model compression, compilation optimization), and a full-stack automated generation framework, and then uses Ara and Gemmini as typical examples to analyze the trade - off mechanisms between different approaches regarding generalization and energy efficiency. The paper also looks ahead to future development directions: sparse computing, Chiplet interconnection, and endogenous security, and gives a very clear, well-organized review of the optimization path from hardware-led to software-led, and finally to full-stack automated design, comparing the classic cases of each path. Based on this analysis, it naturally and convincingly concludes that hardware-led is highly specialized, with strong optimization capabilities but high design cost; software-led is good for rapid iteration but has limited room for improvement; and full-stack automated design strikes the best balance between the two.

***Keywords:*** RISC-V, Edge AI, hardware-software co-design, heterogeneous computing, full-stack automation

## 1. Introduction

The development of Web of Things (WOT) and Edge Computing has naturally led to the demand for edge intelligence, which in turn has created a strong need to deploy complex neural networks on embedded devices, thereby raising the bar for energy efficiency of such devices. Since Moore's Law has slowed down in recent years, general-purpose processors have found it increasingly difficult to meet the energy efficiency requirements, so there has been a fundamental shift in computing architectures from general-purpose processors to domain-specific architectures (DSAs). Against this backdrop, the open-source RISC-V instruction set has been widely adopted in the customization of

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.GU33410

edge AI accelerators, not merely because of its performance, but more importantly due to its modularity, scalability, and license-free characteristics.

Edge AI acceleration technology has naturally evolved through three well-defined stages, and the early stage of software optimization had a very clear and outstanding example: Garofalo et al.'s PULP-NN, which was designed for the parallel ultra-low power (PULP) multi-core cluster. By making optimal use of RISC-V SIMD instructions and fine-grained memory management, it obtained a 19. 6x speedup and a 14. 1x energy efficiency gain over ARM CMSIS - NN on GAP – 8 [1,2]. The hardware extension phase was naturally and logically devoted to ISA customization, with Cavalcante's Ara vector processor reducing instruction overhead by using long vectors (VLEN=2048 bits), hence achieving high GFLOPS/W [3]. Sen's SparCE extended dynamic skipping of invalid computations in a very elegant way using sparse-aware units [4]. More recently, the instruction introduced in the Xiangshan processor gave a 4x speedup for GPT-2 inference, making it a very promising candidate for LLM edge deployment [5].

Compilers and full-stack integration work together very naturally to achieve system-level collaboration: the TVM automated compiler connects high-level frameworks with hardware backends via graph-level and operator-level optimizations, while the Gemmini framework allows full-stack evaluation from RTL up to software libraries, and the Chipyard environment makes it exceedingly easy to simulate and implement custom SoCs, thus greatly lowering the development threshold [6-8]. From the recent proceedings of top conferences such as DAC and ICCAD, it is quite clear that software-hardware collaboration has now become mainstream. Specifically, ICCAD 2024 reports on a RISC-V mixed-precision (INT4/INT8/FP16) co-design with hardware implementing multi-pump soft SIMD, and the compiler automatically chooses the optimal precision [9]. DAC 2024 presents a RISC-V Chiplet heterogeneous architecture that allows very natural, flexible expansion via UCIe interconnect [10].

The article gives a clear, concise summary of the achievements and present difficulties of the previously discussed research directions, then systematically reviews the three main optimization approaches of hardware-led, software-led, and software-hardware collaboration in full-stack automated design, before discussing future trends.

## 2. Hardware-centric co-design techniques (HCCT)

Since the HCCT design starts from the underlying architecture, it is natural and proper to construct it using the open-source, modular, and scalable features of the RISC-V instruction set.

### 2.1. Enhancement of computing power based on standard expansion

The vector extensions defined by RISC-V (such as RVV) provide standard hardware support at the hardware level for neural network workloads. Compared with traditional Single Instruction, Multiple Data (SIMD), the programming model of RVV is independent of vector length, achieving good compatibility, and meanwhile significantly improving throughput.

Ara is an example of the optimization route under discussion, namely that Ara is an open-source vector coprocessor designed primarily for high-performance computing and AI inference, hence it implements the RISC-V vector extension standard and uses that standard to cleanly demonstrate its potential: it supports parallel computing well and has very efficient floating-point operations [3]. Moreover, thanks to its advanced process technology, Ara can reach a main frequency of 1GHz+. This indicates that by implementing standardized RVV extensions, RISC-V processors can achieve significant performance and energy efficiency improvements in core AI operators such as matrix

DOI: 10.54254/2755-2721/2026.GU33410

multiplication and convolution without relying on custom modifications. Experimental data show that when configured with 16 Lanes, Ara achieves a floating-point unit (FPU) utilization rate of up to 97% when running a 256×256 double-precision matrix multiplication, with peak computing power reaching 33 DP-GFLOPS and an energy efficiency ratio of up to 41 DP-GFLOPS/W.

## 2.2. Optimization of custom instruction sets for AI

Although the expansion-oriented design gives RISC-V general acceleration capabilities, it is clear that the standard instruction set is too redundant when handling specific AI models, the natural and effective solution is to use dedicated instructions and fuse complex operator sequences into a single instruction.

XpulpNN is a very natural and clear representative of the optimization approach under consideration [6]. Since the edge-side implementation uses low-precision quantized neural networks, XpulpNN extends the RISC - V basic instruction set with dedicated instructions, introduces SIMD dot product instructions that directly support low - precision data types INT8/INT4, and cleverly designs a fused instruction with " multiply - add - shift - truncate - activation " data path. This significantly reduces the additional instruction overhead for data packaging/unpackaging in the traditional architecture. The quantization evaluation results show that the expansion of XpulpNN brings only 1.8% of hardware power overhead. However, when performing convolution layers, compared to the baseline processor that only supports 8-bit SIMD, its 4-bit and 2-bit convolution operations are respectively 5.3 times and 8.9 times faster. Meanwhile, at the 22nm process node, it achieves a peak energy efficiency of 279 GMAC/s/W, with an overall energy efficiency that is 9 times higher than the baseline, thus making it very well suited for efficient neural network inference on low-power IoT devices. In recent years, instruction automation generation frameworks based on agile development (Agile Development) have indeed become a hot topic in research. Notably, Prakash et al. presented the CFU Playground framework in 2022, which elegantly connects the TensorFlow Lite Micro software layer to the RISC-V hardware layer [11]..

## 2.3. Heterogeneous accelerator architecture design

Because neural network models with large computational requirements cannot be adequately handled by the CPU alone, it is natural and sensible to adopt a heterogeneous computing architecture: the RISC - V core is used as the controller to manage data transfer and process control, while the actual matrix operations are offloaded to dedicated neural network processors (NPU) or deep learning accelerators (DLA).

The Gemmini framework developed by UC Berkeley is a very clear, well-documented representative of the optimization approach under discussion, since it connects to the RISC-V CPU via the RoCC (Rocket Custom Coprocessor) interface of the Rocket Chip, uses a spatial architecture tailor-made for matrix multiplication (GEMM), and therefore naturally leads to a rigorous evaluation of the RISC-V+NPU heterogeneous system [7]. The new JSSC 2024 research (Siracusa) verified the 'At-MRAM' heterogeneous architecture under the 16nm process [12]. Its measured peak energy efficiency reached as high as 8.84 TOPS/W, which was 3 times higher than the end-to-end energy efficiency of the traditional storage architecture. More importantly, the 2022 paper DIANA presented a very elegant and well-executed design of an end-to-end SoC with RISC-V main core as the control center, incorporating a digital coprocessor and an analog SRAM computing storage integration macro [13]. The resulting heterogeneous chip achieved system-level energy efficiency of

DOI: 10.54254/2755-2721/2026.GU33410

hundreds of TOPS/W for deep learning inference tasks, thereby naturally and convincingly demonstrating the potential of RISC-V in scheduling new non-Fourier-Norman architectures.

## 3. Software-led collaborative design technology

Software-Centric Co-Design emphasizes the active adaptation of underlying hardware characteristics from the algorithm and compiler levels to achieve top-down systematic optimization. Different from the traditional "hardware first, software later" decentralized development paradigm, this approach takes RISC-V microarchitecture constraints, including register number, extended instruction set support, and cache capacity as design boundaries, and maximizes chip area utilization and energy efficiency ratio through hardware-aware model compression, targeted compilation optimization, and lightweight runtime adaptation.

### 3.1. Hardware-aware model compression

Model compression is the core technical path to reduce the complexity of edge AI inference computation. However, the compression strategy must closely match the data path characteristics of RISC-V; the software overhead introduced by data format conversion will offset the benefits of hardware acceleration. The XpulpNN framework and its accompanying custom instruction set extensions are typical cases of software-hardware collaborative quantization [14]. When performing INT4 or INT2 sub-byte quantized inference on a standard RISC-V scalar architecture (RV32IMC), due to the lack of native bitwidth data paths, the software must extract weights with shift and mask instructions, resulting in significant register and ALU resource overhead. XpulpNN adopts the following software and hardware collaboration strategy to solve the above problems: at the software level, multiple 2-bit or 4-bit weights and activation values are packed into a single 32-bit register to reduce memory bandwidth pressure; At the hardware level, a dedicated dot product instruction (e.g., pv.sdotsp.b) is introduced, which can perform subbyte operand unpacking and multiplication (MAC) calculations in parallel directly from 32-bit registers in a single clock cycle [14]. Experimental results show that the deep coupling of the above software packaging and hardware-native unpacking/computation improves the inference energy efficiency of the 4-bit quantitative network by about 10 times compared with the standard RISC-V implementation [14].

### 3.2. Compiler optimization and code generation

The compiler is a critical abstraction layer that connects high-level AI frameworks with diverse RISC-V hardware. RISC-V Vector Extension (RVV) supports variable vector length (VLEN), and the length of physical vector registers in different implementations can range from 128 to 1024 bits; the cache specifications vary between implementations. Traditional methods that rely on manual inline assembly optimization struggle to cope with this hardware diversity. The automatic search and code generation mechanism based on the Apache TVM deep learning compiler is the mainstream solution to this problem [6].

TVM abstracts operators such as convolution, matrix multiplication, and so on into a multi-layer nested loop structure through its underlying TensorIR (TIR) intermediate representation. At the core of its collaborative optimization mechanism lies the AutoTVM auto-tuning engine: during the compilation phase, AutoTVM automatically searches for the optimal tile size and unroll factor in the scheduling space by using the microarchitecture parameters of the target RISC-V core, including VLEN, SRAM capacity, and multiplier latency. In addition, TVM is able to sense RVV's unique

DOI: 10.54254/2755-2721/2026.GU33410

vector length multiplier (LMUL) mechanism and logically bind multiple physical vector registers to ultra-wide virtual registers through software configuration, thereby approaching the theoretical throughput limit of the FPU pipeline while avoiding register spills. [Moon2] Measured data show that the binary code size of core tensor operators can be reduced by about 90% compared to traditional manually optimized operator libraries (such as muRISCV-NN) through the above-mentioned automated tuning and code generation based on TVM, and the execution delay of the core tensor operator can be reduced by about 46% compared to GCC automatic vectorization in end-to-end AI model inference [6]. At the same time, thanks to the static memory planning at the microtensor compiler (microTVM) level, the peak memory footprint of the model can be further reduced by about 25%, significantly improving the deployment of complex models on edge bare-metal devices with extremely limited storage [6].

## 4. Automated full-stack co-design framework

The RISC-V automated full-stack co-design framework for edge AI acceleration aims to balance the performance, energy efficiency, and area of AI tasks under resource constraints, and realize end-to-end hardware-software co-optimization from AI model to hardware deployment. The framework adopts a five-layer architecture including model optimization, compilation optimization, hardware customization, simulation verification, and physical implementation, integrating technologies such as model lightweighting, TVM compilation, RISC-V customization, and FireSim/Gem5 simulation. It features full-stack coverage, high automation, flexible scenario adaptation, energy efficiency priority, and open-source compatibility, and can provide a standardized and efficient RISC-V hardware-software co-design solution for edge AI acceleration.

### 4.1. Design space exploration (DSE): automated search for optimal RISC-V parameter configuration to match specific AI models

The implementation of automated Design Space Exploration (DSE) is centered on multi-objective optimization of edge scenario performance, energy efficiency, and resource occupation, relying on the configurable IP generation capability of Chipyard, FireSim FPGA-accelerated simulation, and TVM ML-driven optimization technology. First, the legal parameter space is defined based on the engineering configurable range of the Rocket/BOOM core, and illegal combinations that cause resource waste are excluded [8]. At the verification and simulation level, a hybrid search strategy of "ML prediction + simulation verification" is adopted, and potential optimal configurations are pre-screened through the XGBoost cost model. A model-parameter co-optimization mechanism is integrated: for compute-intensive models (e.g., ResNet50), the focus is on optimizing the issue width and L1 D-Cache size; for memory-intensive models (e.g., MobileNetV2), the L2 Cache configuration is prioritized. Finally, the multi-objective decision module normalizes the indicators, outputs the Pareto optimal configuration in combination with user priorities, and verifies the consistency of the indicators in real hardware or ASIC synthesis (error $\leq$5%), achieving accurate matching between parameter configuration and the computation/memory access characteristics of specific AI models.

### 4.2. Model optimization automation

Model optimization automation is the front-end core module of the full-stack automated co-design framework. Its core goal is to complete model lightweighting and native RISC-V hardware

DOI: 10.54254/2755-2721/2026.GU33410

adaptation through automated technology under the premise of controllable accuracy loss of AI tasks (usually ≤3%), reduce the computation, storage, and power consumption overhead of edge devices, and output key characteristic parameters such as quantization precision, sparse mode, and operator structure for subsequent compilation optimization and hardware customization links, laying the foundation for full-link co-optimization of "model-compilation-hardware".

Focusing on the strong resource constraints of edge AI scenarios, this module builds a hardware-aware automated optimization system for mainstream edge models such as CNN, LSTM, and Transformer, covering core technologies such as automated quantization, structured sparsification, operator fusion, and computational graph optimization. Relying on frameworks such as XpulpNN and CFU Playground, it realizes the collaborative adaptation of model lightweighting strategies with RISC-V instruction sets and custom functional units, and can complete end-to-end automated mapping from model operators to hardware instructions, breaking through the limitation of the disconnection between traditional model optimization and hardware architecture.

At the same time, this module can be deeply linked with the TVM compiler stack and the muRISCV-NN lightweight operator library, realize automated deployment and adaptation of the optimized model based on the characteristics of the RISC-V RVV vector extension, and solve the efficiency bottleneck of tail data processing under the fixed vector width architecture. It can also automatically adjust the model computing scheduling and data layout strategy for the heterogeneous acceleration architecture of RISC-V main control + NPU, maximize the energy efficiency benefits of model optimization, and finally realize the full-process automated conversion from the original pre-trained model to edge RISC-V hardware deployment, solving the core pain points of low efficiency and poor hardware adaptability of traditional manual optimization.

## 5. Discussion

### 5.1. Comparative analysis of existing work

As mentioned above, there are three existing solutions for hardware-oriented optimization, and the comparison is shown in Table 1.

Table 1. Comparison of different ways above

| Technical classification | Representative cases | Pros | Cons |
|---|---|---|---|
| based on standard expansion | Ar (ETH Zurich) | Excellent software compatibility and strong versatility. | Low energy efficiency ratio and large decoding overhead |
| Optimization of Custom Instruction | XpulpNN/RI5CY | High area utilization rate control logic simplification | Accelerate the fragmentation of the RISC-V ecosystem |
| Heterogeneous Accelerator Architecture Design | Gemmini (UC Berkeley) | Optimal performance in terms of computing power and energy efficiency ratio | Long development cycle and the poorest flexibility |

### 5.2. Standardization and customization

Traditional RISC-V hardware-software co-design is highly dependent on customized design. Customized solutions centered on Application-Specific Instruction-Set Processors (ASIP) can achieve significant computing power and energy efficiency improvements for specific application

DOI: 10.54254/2755-2721/2026.GU33410

scenarios while retaining processor programmability. Relevant studies have achieved a maximum program execution acceleration of 2.47× and an average running time reduction of 44%. However, the traditional customization process has problems such as excessively long development and verification cycles, and great difficulty in microarchitecture adaptation of custom instructions and compiler chain retargeting due to the small number of IP reuse, making it difficult to cope with the rapid iteration of AI algorithms.

Therefore, it is necessary to shift from standardized development methods to agile development, the core of which is to realize software-defined hardware through the generator mechanism of the automated framework. With the help of generators and end-to-end automation tools, developers can carry out interactive development for software, automatically generate RTL code and supporting drivers, and greatly shorten the SoC design cycle.

The standardized framework also realizes hardware-software decoupling and closed-loop optimization in the architecture design phase. For example, compiler stacks represented by TVM and DORY can automatically perceive the underlying hardware characteristics, realize end-to-end operator scheduling and memory management optimization, ensure that every microarchitecture adjustment can be fully utilized by the software stack, and realize a closed loop of "algorithm-compilation-architecture".

## 6. Conclusion

This paper presents a comprehensive survey of RISC-V hardware-software co-design for edge AI acceleration scenarios. Aiming at the core contradiction between the resource constraints of edge devices and the surging demand for AI computing power, this paper systematically analyzes the technical characteristics, typical cases, and optimization logic of the three major technical routes: hardware-centric, software-centric, and full-stack automated co-design. The study finds that the hardware-centric optimization route relies on the open-source and scalable features of RISC-V to achieve computing power enhancement through standard instruction extensions, custom instruction sets, and heterogeneous architecture design. Although it can obtain significant performance and energy efficiency improvements, it has problems such as high design cost, insufficient flexibility, or ecosystem fragmentation. Software-centric optimization starts with model compression, compilation optimization, and lightweight runtime, and adapts to RISC-V microarchitecture constraints through hardware-aware top-down optimization. It has the advantage of rapid iteration, but it is difficult to achieve a breakthrough in performance. The automated full-stack co-design framework integrates the advantages of the former two, realizes end-to-end optimization from AI model to hardware deployment through a five-layer architecture and three-level collaborative mechanism, and achieves a good balance in automation, scenario adaptability, and energy efficiency performance, becoming an important direction to solve the challenges of edge AI hardware-software co-design.

Overall, single-dimensional optimization has reached the performance bottleneck, and vertically integrated co-design that breaks down the barriers between hardware and software development is the only way for edge AI acceleration. The open-source and modular features of RISC-V provide core support for the innovation of domain-specific architectures. In the future, continuous exploration combined with sparse awareness computing, endogenous security, Chiplet interconnection, and other technologies will further promote the maturity and improvement of the RISC-V hardware-software co-design ecosystem and provide more efficient and flexible solutions for diverse edge AI scenarios.

DOI: 10.54254/2755-2721/2026.GU33410

## Authors contribution

All the authors contributed equally, and their names were listed in alphabetical order.

## References

[1] Garofalo, A., Rusci, M., Conti, F., Rossi, D., & Benini, L. (2020). PULP-NN: Accelerating quantized neural networks on parallel ultra-low-power RISC-V processors. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 378(2164), 20190155.

[2] Rossi, D., Conti, F., Marongiu, A., Pullini, A., Loi, I., Gautschi, M., ... & Benini, L. (2015, August). PULP: A parallel ultra low power platform for next generation IoT applications. 2015 IEEE Hot Chips 27 Symposium (HCS), Cupertino, CA, United States.

[3] Cavalcante, M., Schuiki, F., Zaruba, F., Schaffner, M., & Benini, L. (2019). Ara: A 1-GHz+ scalable and energy-efficient RISC-V vector processor with multiprecision floating-point support in 22-nm FD-SOI. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 28(2), 530–543.

[4] Sen, S., Jain, S., Venkataramani, S., & Raghunathan, A. (2018). SparCE: Sparsity aware general-purpose core extensions to accelerate deep neural networks. IEEE Transactions on Computers, 68(6), 912–925.

[5] Beijing Institute of Open Source Chip (BOSC). (2024). XiangShan. GitHub. https: //github.com/OpenXiangShan

[6] Chen, T., Moreau, T., Jiang, Z., Zheng, L., Yan, E., Shen, H., ... & Krishnamurthy, A. (2018). TVM: An automated end-to-end optimizing compiler for deep learning. 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18), Carlsbad, CA, United States.

[7] Genc, H., Kim, S., Amid, A., Haj-Ali, A., Iyer, V., Prakash, P., ... & Shao, Y. S. (2021, December). Gemmini: Enabling systematic deep-learning architecture evaluation via full-stack integration [Paper presentation]. 2021 58th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, United States.

[8] Amid, A., Biancolin, D., Gonzalez, A., Grubb, D., Karandikar, S., Liew, H., ... & Nikolić, B. (2020). Chipyard: Integrated design, simulation, and implementation framework for custom SoCs. IEEE Micro, 40(4), 10–21.

[9] Armeniakos, G., Maras, A., Xydis, S., & Soudris, D. (2024, October). Mixed-precision neural networks on RISC-V cores: ISA extensions for multi-pumped soft SIMD operations. Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design (ICCAD), New Jersey, NJ, United States.

[10] Evans, A., Fuguet, C., & Million, D. (2024, October). OpenSource heterogeneous chiplet-based computing architectures. Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design (ICCAD), New Jersey, NJ, United States.

[11] Prakash, S., Callahan, T., Bushagour, J., Banbury, C., Green, A. V., Warden, P., ... & Reddi, V. J. (2023, April). CFU Playground: Full-stack open-source framework for tiny machine learning (tinyML) acceleration on FPGAs. 2023 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), Raleigh, NC, United States.

[12] Prasad, A. S., Scherer, M., Conti, F., Rossi, D., Di Mauro, A., Eggimann, M., ... & Benini, L. (2024). Siracusa: A 16 nm heterogenous RISC-V SoC for extended reality with at-MRAM neural engine. IEEE Journal of Solid-State Circuits, 59(7), 2055–2069.

[13] Ueyoshi, K., Papistas, I. A., Houshmand, P., Sarda, G. M., Jain, V., Shi, M., ... & Verhelst, M. (2022, February). DIANA: An end-to-end energy-efficient digital and ANAlog hybrid neural network SoC. 2022 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, United States.

[14] Garofalo, A., Tagliavini, G., Conti, F., Rossi, D., & Benini, L. (2020, March). XpulpNN: Accelerating quantized neural networks on RISC-V processors through ISA extensions. 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France.