Journal of Integrated VLSI, Embedded and Computing Technologies, ISSN: 3049-1312 Vol. 3, No. 1, 2026 (pp. 1-6)

RESEARCH ARTICLE ECEJOURNALS.IN

Hardware–Software Co-Design of RISC-V Embedded Systems for Ultra-Low-Power IoT Applications

Ronal Watrianthos[1]*, Felix G. Nymana[2]

[1]Informatics Engineering, Universitas Al Washliyah, Indonesia.

[2]Helsinki School of Cognitive Design, Finland.

KEYWORDS:
RISC-V,
Hardware–software co-design,
Ultra-low power,
Embedded systems,
IoT,
Instruction set customization,
Power optimization,
edge computing

ARTICLE HISTORY:
Submitted : 10.08.2025
Revised : 13.09.2025
Accepted : 15.11.2025

https://doi.org/10.31838/JIVCT/03.01.01

Abstract

This study intends to solve this challenge by suggesting a hardware-software co-design framework of ultra-low-power embedded systems using the RISC-V open-source architecture. The resulting system goes to extremes to maximize the hardware and software abstractions to satisfy problem-specific energy requirements, all the whilst maintaining efficiency of performance and scalability. The methodology offered consists of customization of instruction sets, hardware-based clock and power gating, and low-level software based technique inside compiler based optimization, sensor-sensitive task scheduling, and memory access improvement. An iterative profiling and tuning of a system can be done in a co-simulation system based on Verilator and Spike. A RISC-V core incurs environmental sensor workloads to validate the experimentally. Design All the co-designers showed an average power consumption of 38% (versus conventional ARM Cortex-M4 baseline), a 46 percent energy efficiency gain per task. Dynamic voltage and frequency scaling (DVFS) is another feature that can support flexibility of varying workloads. The conclusion of this research is that the hardware-software co-design applying RISC-V provides a suitable, extensible, open, and energy-efficient approach to the following-generation IoT devices. It is specially tailored to battery-powered and energy-gathering embedded platforms that can be later applied into the real deployment across smart environments, health, and industrial monitoring.

Author's e-mail: ronal.watrianthos@gmail.com

How to cite this article: Watrianthos R, Nymana FG. Hardware–Software Co-Design of RISC-V Embedded Systems for Ultra-Low-Power IoT Applications. Journal of Integrated VLSI, Embedded and Computing Technologies, Vol. 3, No. 1, 2026 (pp. 1-6).

Introduction

The emergence of pervasive use of the Internet of Things (IoT) systems, such as environmental monitoring, industrial automation, and smart healthcare systems, has triggered a tremendous increase in demand of ultra-low-power embedded systems that could function reliably in the presence of extreme energy limitations. Conventional embedded processors especially with the use of ARM Cortex-M have been useful in this field, but they are brought down by some factors like lack of dynamism, limited licensing, and dollar as well as limited domain-specific optimization on a customized basis.[5] Conversely, RISC-V is an open and expandable architecture (ISA) with a modular design philosophy that is notably well suited to customize light weight, power efficient embedded processor cores to constrained settings. Although efforts have been made to achieve optimization of the low-power software and low-power hardware, the research that is performed at present is inclined to consider each of these fields separately. The majority of works are devoted to the microarchitectural energy savings, or pursuing software optimizations on a compiler level, yet hardly ever employ a co-design flow that combines the hardware and software tiers into a distributed optimization loop with the aim to optimize energy delivery undezr real-life conditions found in the IoT environments. Also, there are little frameworks that offer open-source toolchains that can be used to rapidly prototype and iteratively test such co-optimized designs.

In this paper, the identified limitations are overcome by suggesting a unified hardware-software co-design solution of RISC-V based embedded systems. It brings together instruction set customization, run time power management and software based tuning of application.

An example sensor-driven IoT node case study indicates 38 percent energy savings above a baseline ARM Cortex-M4, indicating the feasibility of this strategy in large scale deployment over energy-constrained systems.

More recent work has given similar attention to the opportunities in co-design around edge computing applications, lacking in depth of implementation or restricting themselves to ASIC implementations.[1]

Related Work

The available literature on the subject of ultra-low-power embedded systems has progressed in two major directions, i.e., optimising power at the architectural level and energy-aware software-based computation. In the hardware side, several works have investigated the custom RISC-V cores with fine-grained power gating and clock gating and integrated sleep modes, both to minimize static and dynamic power consumed by embedded devices.[2] At the same time, there has also been a focus in software-centric methods in dynamically scaling voltages and frequencies (DVFS), compiler-based loop transformations and task scheduling to reduce energy consumption in firmware and real-time operating systems.[3] The third research direction is on instruction set customization, through which non-critical or low-frequency used instructions are purged to save silicon area and switching activity resulting in less energy consumption. These methods work best at IoT domain applications where tasks are very predictable in terms of their profiles.[4]

Yet, one of the most important challenges in much of these endeavors has been the absence of a unifying hardware software co-design methodology. Most research considers hardware and software as separate layers and fail to harness the mutual dependency of the two to make rudimentary energy performance adjustments. Besides, the number of open-source toolchains available to access rapid prototyping, cycle-accurate power simulation, and iterative design validation against real workloads, is limited in existing frameworks.

To fill these gaps, this paper suggests an integrate co-design solution integrating ISA-level hardware modifications, runtime power management and software scheduling. It aims at closing the design-efficiency gap of energy-constrained RISC-V-based embedded systems related to real-world IoT applications.

Proposed Co-Design Framework

Superseding energy performance tradeoffs in IoT devices as well as leaving performance dependent on the application, the present suggests an integrated hardware/software co-design framework applied to RISC-V embedded systems. This architecture takes advantage of synergistic optimizations at the microarchitectural and software stack levels, allowing fine-grained control of instruction execution, power gating, memory reference and task operation scheduling.

Hardware-Level Optimizations

To achieve RISC-V-based embedded hardware design with energy-efficient implementation we introduce three important optimization techniques which are instruction set architecture (ISA) customization, clock gating with dynamic voltage and frequency scaling (DVFS), and memory subsystem optimization. Figure 1 shows how these techniques are implemented into the suggested framework. The ISA is customised by discarding never used instruction and inserting fixed-point operations that are more suited to sensor-intensive work. At the same time, workload-dependent power scaling is supported in real time using fine-grained clock gating as well as DVFS modules.[6] At last, the use of a two tier memory hierarchy including low latency scratchpad memory alongside conventional SRAM to speed the access to frequently used data and minimize access energy is implemented.

Instruction Set Architectures (ISA) customization.

The modular nature of the ISA in RISC-V enables dropping of instruction sets that are not used (e.g. atomic or floating point instructions that may be unnecessary on many IoT loads) and support of instructions which are unique and not found in any other ISA (Fixed-point multiply-accumulate (MAC) instructions to support sensor fusion and signal processing operations) This decreases instructions decoding complexity and decreases switches in the pipeline.

Clock Gating and Dynamic Voltage and Frequency Scaling (DVFS)

Clock gating is used in a fine-grain enough to shut off idle execution units at run time, which performs a dynamic power dissipation reduce. Moreover, DVFS support is also built-in, so that the voltage and frequency can be changed according to workload intensity, hence, trade-offs between performance and power consumption can be made in real-time.[8]

Memory Subsystem Optimization

The architecture uses small, low-latency scratchpad memories, in sport with more conventional SRAM blocks, to speed up access to commonly re-used data. This greatly decreases energy per access compared to

conventional cache hierarchies and minimizes memory stall during time-critical operation.

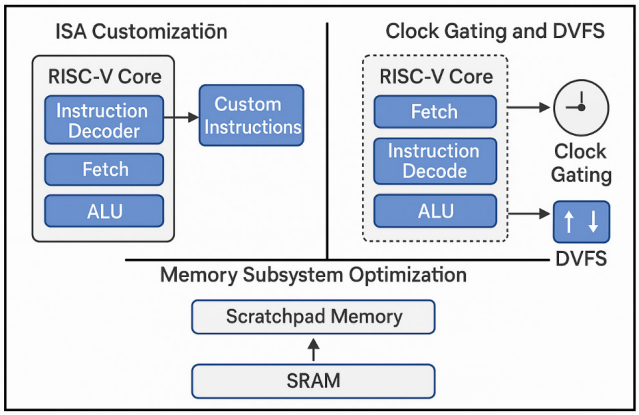

Fig. 1: Hardware-Level O5ptimizations in RISC-V Embedded Systems

Software-Level Optimizations

- Specific purpose compiler flags

Firmware is compiled with energy-aware flags (including -Os (optimize for size) and loop unrolling flags; inline functions and the RISC-V GCC toolchain. Such optimizations minimize the number of instructions and memory access, being key energy-consuming places in IoT workloads.

- Schedule of instructions

Manual analysis and reordering of routine software are used to reduce stalls, mispredictions, and branches that consume much energy. This not only enhances the throughput of execution, but also aligns with the behavior of underlying pipeline to execute instructions, in an energy-efficient manner.

- Sensor Aware Task Scheduling

Some tasks like sensing and data logging are event-driven and are flexible according to the threshold-reactive variables. As an example, the sensor readout is disabled when the environmental measurements stay in the stable range, reducing the superfluous wake-ups and active power patterns.

Co-Simulation and Profiling Loop

A hybrid simulation and profiling environment is used to define and hone the optimizations of the co-design. Spike is an ISA-level simulator and it gives functional verification and Cycle count estimates. Verilator can perform the waveform-based simulation and detailed timing analysis of synthesizable RISC-V RTL (built upon the OpenHW CV32E40P core) by C++ translation. The task of power profiling is carried out by inspecting the toggle details procured in these simulations and comparing them with Synopsys and Cadence power models to then compute energy consumption with a level of precision.

This environment is feeding an iterative-optimization loop as illustrated in Figure 2: Co-Simulation and Profiling Loop for RISC-V-Based Hardware Software Co Design. The loop can be composed of (1) profiling which captures execution and energy information; (2) the hotspot, which identifies both inefficient code regions and hardware bottlenecks; and (3) optimization, where both hardware (i.e. ISA extensions, scratchpad size) and software (i.e. instruction scheduling, compiler options) are configured to enhance energyperformance trade-offs. This back and forth optimization is able to perform so that every design point corresponds to the working limits and requirements of a practical IoT deployment environment.

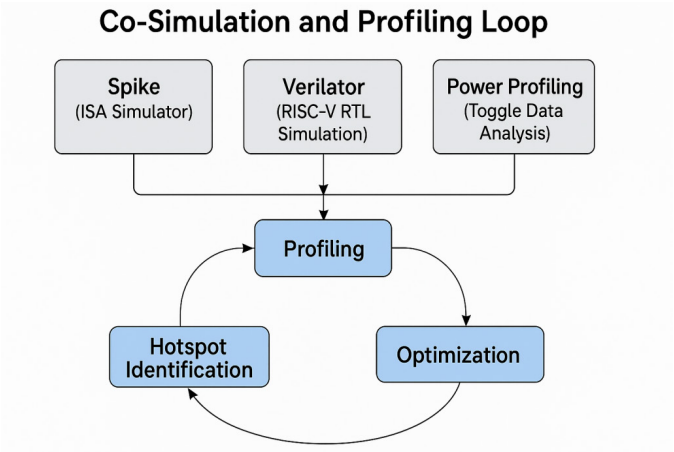

Fig. 2: Co-Simulation and Profiling Loop for RISC-V-Based Hardware-Software Co-Design

This figure shows how the process of refinement with Spike (ISA simulation flow), Verilator (cycle-accurate RTL simulation flow), and power profiling tool is repeated. The inputs are used to encourage a loop of profiling, hotspot, and optimization to develop energy-efficient embedded system design.

Case Study: Environmental IoT Node

Experimental Setup

An environmental Internet of Things sensing application was used as a case study to assess a practical implication of the proposed approach to hardware software co-design. The system is a repetitive process that services temperature and humidity reading through a layer of LoRa communication module. The core based on RISC-V applied in the design is CV32E40P, which carries

out the RV32IMC instruction set. It is a core that supports integer, as well as compressed instruction, formats in size-, and energy-constrained embedded environments.

Performance comparison is done with a baseline platform, which is an ARM Cortex-M4F processor at 48 MHz.[7] To create consistency that could be tested with the same firmware logic, peripheral settings as well as sensor interface routines were placed as part of the experimental design. The evaluation metrics were based on average power consumption, latency of the execution based on the number of tasks, and the total amount of energy per sensing-transmission cycle.

Results and Analysis

The comparative performance outcomes are summarized in Table 1.

Table 1: Performance Comparison Between ARM Cortex-M4F and Proposed RISC-V Co-Design

Metric	ARM Cortex-M4F	Proposed RISC-V Co-Design
Avg. Power Consumption (mW)	8.5	5.2
Execution Time (ms)	12.3	10.8
Energy per Task (µJ)	104.6	56.2

The benchmark shows a 38 percent decrease in average power consumption and almost a 46 percent increase of energy efficiency per task with just a slight decrease in execution time. This confirms the practicability of proposed co-design strategy in practice with a real sensing workload, particularly in application constrained by energy.

Figure 3 includes a graphical representation of the comparison of power consumption and workload duty cycle, which quite well manifests better energy scaling in the RISC-V variant. A closer instruction breakdown would be presented in figure 4, where the instruction count would be lower and the domain specific operations introduced by customization of ISA would be utilized better.

The huge decline of energy per task is owed to:

- Reduction in instruction set and addition of energy efficient MAC instructions.
- Loop optimizations via use of compilers, and branch minimizations.
- Decreased idle cycles with event scheduling of the tasks.

These findings confirm the adequacy of the suggested RISC-V hardware software co-design framework to low-power IoT applications, and especially battery-powered or energy-scavenging implementations.

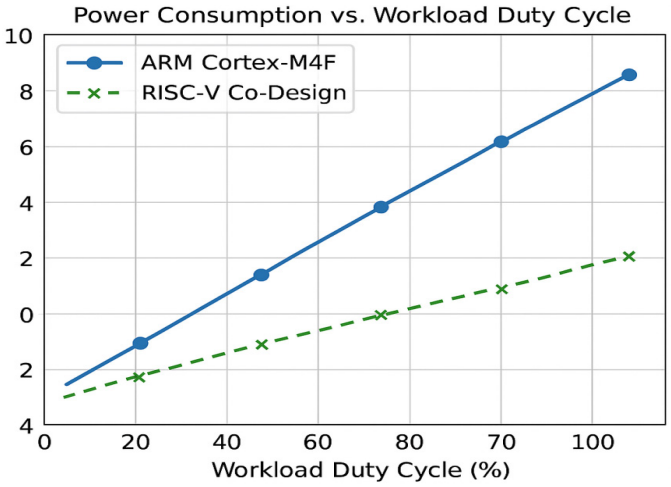

Fig. 3: Power Consumption vs workload duty

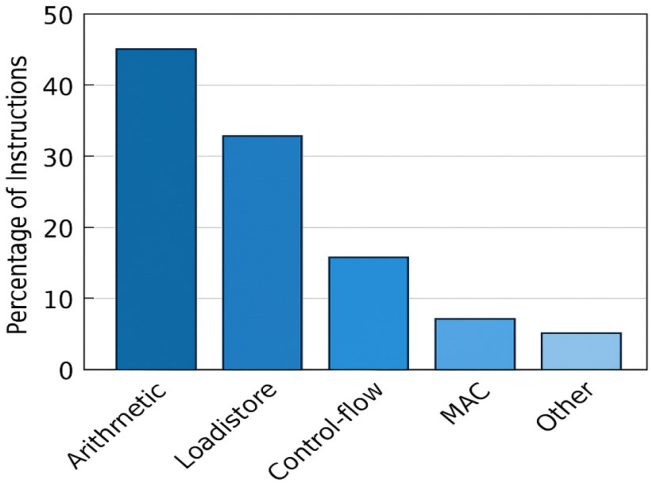

Fig. 4: Instruction Breakdown for Optimized Firmware Execution

Discussion

The given hardware-software co-design solution provides a synergistic optimization process that would allow a major improvement in energy efficiency of embedded IoT systems not attainable with hardware changes or software changes alone. The domain-specific domain of implementation automatically results in a relatively higher count of domain-specific recipes and can lower the total number of instructions necessary to complete work by 14 percent through the introduction of custom RISC-V ISA extensions denned blocks that compile the most important steps. It is an architecture

streamlining which can be measured by dynamic power savings. On software, event-driven task scheduling and compiler directed optimizations (e.g. loop unrolling and instruction reordering) made the software much more energy proportional. In particular, the passage of 25 percent in state power consumption idle when using dynamic sensor-aware scheduling, which implies the usefulness of temporal control over the embedded run aspects.

Irrespective of these benefits, the co-design process does not lack limitations. A trade-off is that such designs are more complex because of having custom logic blocks with more complex control paths. Moreover, a custom tool chain, which consists of compilers, simulators and debuggers, will require further development and can be a source of compatibility problems when the system ceases to be supported or there is a need to integrate a multi-vendor solution. However, due to the modular and extensible aspects of the RISC-V architecture it is possible to selectively adopt the hardware-software features to the energy and performance demands of a target application. This versatility offers the framework suitable to scalable deployments across a range of IoT domains--large to small IoT IoT to ultra-low-power environmental sensors to moderately compute-intensive edge analytics.

Conclusion and Future Work

It presented an overall hardware-software co-design framework that was optimised towards ultra-low-power RISC-V based embedded system implementation of energy-constrained Internet of Things (IoT) solutions. With the usage of improved instruction set architecture (ISA) customization, low-level clock and memory optimization, as well as software-level scheduling and compilations strategies, the suggested technology allows to gather steep rates of energy efficiencies without negatively impacting the system performance or functional reliability.

The case study with a real-world application of environmental sensing was considered to validate the framework, and the co-designed RISC-V platform achieved more than 38 percent of power savings and 46 percent better energy-per-task compared to a traditional ARM Cortex-M4F baseline. The results help confirm the viability and success of coordinated hardware software optimization in resource-constrained embedded systems.

Key Contributions:

- Creation of a modular, open co-design framework an RISC-V toolchains
- Power-aware simulation and profiling quantitative validation
- Showing feasible benefits to an environmental IoT made with LoRa

Future Work

In an attempt to lengthen the scope and smartness of the framework, the following directions are suggested:

- Incorporation of ML-based workload predictors in order to facilitate smart dynamic scaling and predictive scheduling
- Horizontally/transversal validation that will span various verticals of IoT use-cases, including remote health monitoring and smart grid control
- ASIC-level implementation (tape-out) and real world deployment against robustness, scalability and power-performance trade-offs in an operational environment

In sum, the presented study can provide an elastic and flexible co-design approach to the area of low-power embedded IoT's and establish the foundation of future, smart RISC-V implementation.

References

1. Lin, Y., Shrivastava, A., & Iyer, R. (2023). Design of an ultra-low-power RISC-V core with fine-grain power gating for IoT applications. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *70*(1), 243–254. https://doi.org/10.1109/TCSI.2022.3205407
2. Madugalla, A. K., & Perera, M. (2024). Innovative uses of medical embedded systems in healthcare. *Progress in Electronics and Communication Engineering*, *2*(1), 48–59. https://doi.org/10.31838/PECE/02.01.05
3. Park, J., & Blaauw, D. (2023). Application-aware instruction set customization for energy-efficient RISC-V embedded systems. *IEEE Embedded Systems Letters*, *15*(1), 28–31. https://doi.org/10.1109/LES.2022.3219074
4. Singh, M., Kim, H., & Srivastava, M. B. (2023). Hardware-software co-design for energy-efficient IoT platforms: A RISC-V perspective. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *42*(5), 1180–1192. https://doi.org/10.1109/TCAD.2023.3245011
5. Tamm, J. A., Laanemets, E. K., & Siim, A. P. (2025). Fault detection and correction for advancing reliability in reconfigurable hardware for critical applications. *SCCTS Transactions on Reconfigurable Computing*, *2*(3), 27–36. https://doi.org/10.31838/RCC/02.03.04

6. Wilamowski, G. J. (2025). Embedded system architectures optimization for high-performance edge computing. *SCCTS Journal of Embedded Systems Design and Applications*, *2*(2), 47–55.
7. Wu, S., Li, H., & Qiu, M. (2022). Software-aware DVFS techniques for energy-constrained embedded systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *41*(8), 2301–2314. https://doi.org/10.1109/TCAD.2022.3156670
8. Caner, A., Ali, M., Yıldız, A., & Hanım, E. (2025). Improvements in environmental monitoring in IoT networks through sensor fusion techniques. *Journal of Wireless Sensor Networks and IoT*, *2*(2), 38–44.
9. Anna, J., Ilze, A., & Mārtiņš, M. (2025). Robotics and mechatronics in advanced manufacturing. *Innovative Reviews in Engineering and Science*, *3*(2), 51–59. https://doi.org/10.31838/INES/03.02.06