Multidisciplinary Engineering Science Open

Received date: 11 Dec 2024
Revised date: 10 Jan 2025
Accepted date: 20 Jan 2025
Published date: 01 Feb 2025

RISC-V in Safety-Critical Embedded Systems: Hardware/Software Co-Design and Assurance Cases

Chloe Harris[1], Rana Al-Salameh[2*]

Citation: Harris, C., & Al-Salameh, R. (2025). RISC-V in Safety-Critical Embedded Systems: Hardware/Software Co-Design and Assurance Cases. *Multidisciplinary Engineering Science Open*, 2, 1-13.

Abstract

This review investigates how RISC-V, an open and extensible instruction set architecture, can be effectively adopted in safety-critical embedded systems through integrated hardware/software co-design strategies and structured assurance cases that ensure compliance with functional safety standards. A qualitative systematic review design was employed to synthesize the state of research on RISC-V implementation within safety-critical environments. Seventeen peer-reviewed journal articles and conference papers published between 2015 and 2025 were selected from IEEE Xplore, Scopus, Web of Science, and ACM Digital Library databases based on inclusion criteria emphasizing RISC-V architectures, safety assurance, and verification frameworks. Data collection consisted exclusively of literature review. Thematic analysis using NVivo 14 software was conducted through open, axial, and selective coding, with theoretical saturation achieved at the seventeenth article. Four core themes were extracted: (1) hardware/software co-design paradigms, (2) safety assurance and certification frameworks, (3) open-source ecosystem and verification governance, and (4) energy–latency trade-offs and performance assurance. Results reveal that modular co-design approaches in RISC-V enable domain-specific optimizations while maintaining deterministic timing and verifiability. Structured assurance cases—built on Goal Structuring Notation (GSN) and model-based verification—are emerging as credible mechanisms for aligning open-hardware transparency with certification expectations such as ISO 26262, DO-254, and IEC 61508. The open-source RISC-V ecosystem enhances reproducibility, community validation, and toolchain verification but introduces challenges in provenance tracking and standardization. Furthermore, energy-aware design techniques like dynamic voltage and frequency scaling (DVFS) can improve efficiency without compromising real-time safety guarantees when combined with rigorous timing validation. Collectively, the findings highlight a co-evolution of technical innovation and assurance methodology that redefines safety in open architectures. RISC-V's adoption in safety-critical domains depends on unifying co-design practices with auditable assurance frameworks that demonstrate both functional safety and transparency. Future progress will hinge on standardized open-hardware certification models, formal verification integration, and collaborative governance to balance innovation with accountability.

Keywords: RISC-V; safety-critical systems; hardware/software co-design; assurance case; functional safety; open-source verification; energy-latency trade-off

1. Department of Civil and Environmental Engineering, University of Melbourne, Melbourne, Australia
2. Department of Petroleum Engineering, Hashemite University, Zarqa, Jordan

*Correspondence: rana.alsalameh@hu.edu.jo

1. Introduction

The advent of RISC-V, an open and extensible instruction set architecture (ISA), has catalyzed a paradigm shift in embedded computing by offering hardware designers and system architects a liberated pathway to tailor microarchitectures to domain-specific needs (Waterman et al., 2013; RISC-V International, 2023). Unlike legacy proprietary ISAs that impose fixed feature sets and licensing constraints, RISC-V's modular "base + extensions" philosophy enables minimalist implementations as well as rich, domain-optimized variants (Codasip, 2023). In safety-critical embedded systems—spanning aerospace, automotive, medical, and industrial control domains—such flexibility presents compelling opportunities but also profound challenges in assurance and certification. As system complexity and safety assurance expectations increase, the community must work toward rigorous hardware/software co-design frameworks and assurance cases that convincingly argue correctness, fault containment, and compliance.

Safety-critical systems are distinguished by their potential to incur catastrophic consequences in case of failure. Meeting standards such as ISO 26262 (automotive), DO-254/DO-178 (aerospace), and IEC 61508 (industrial) demands demonstrable traceability, formal verification, redundancy, and rigorous argumentation through structured safety cases (Kelly & McDermid, 2021; Broster et al., 2024). Traditional approaches rely on well-vetted proprietary cores, vendor-certified toolchains, and black-box IP protections. The open, vendor-neutral nature of RISC-V disrupts that model: while it fosters transparency, openness, and innovation, it also implies that core logic, toolchains, and microarchitectural extensions are subject to scrutiny rather than assumed safe by pedigree (Díaz et al., 2023). Consequently, embedding RISC-V into safety-critical systems is not simply a matter of porting software; rather, it demands co-engineering of hardware and software under a unified assurance framework.

One of the key strengths of RISC-V for safety-critical design is its modularity and extensibility. Designers can include only the required features—e.g., integer instructions, atomic primitives, floating-point, vector extensions—and omit unnecessary complexity, thereby reducing the attack surface and verification burden (Sysgo, 2025). This lean approach is particularly advantageous in constrained embedded domains where area, power, and determinism are critical. Moreover, the openness of RISC-V allows independent validation of the ISA specification, microarchitecture, and implementation, mitigating the risk of hidden design faults that evade proprietary black-box verification (RISC-V International, 2023; Embedded.com, 2025). Several recent efforts demonstrate applying RISC-V to safety domains: for example, the Mi-V ecosystem now offers a functional safety RISC-V processor IP core certified for ASIL-level compliance (CAST/Microchip, 2024) and SiFive publishes pre-certified IP blocks intended to reduce certification burden at integration time (SiFive, 2025). These

developments signal that RISC-V is transitioning from research curiosity toward industrial relevance in safety contexts.

Nonetheless, integrating RISC-V into safety-critical embedded systems demands rigorous hardware/software co-design strategies that reconcile conflicting constraints: real-time determinism, energy/power budgets, fault tolerance, and safety assurance. Co-design must ensure that microarchitectural extensions (e.g., custom functional units, accelerators) remain verifiable, that timing predictability is maintained, and that safety arguments span hardware, firmware, and software layers. Prior work in co-verification, especially in RISC-V System-on-Chip (SoC) contexts, shows that combining hardware and software verification across abstraction boundaries (e.g., cycle-accurate simulation, assertions, equivalence checking) is essential to catch mismatches and hidden faults (Chen et al., 2022). Advances in co-design for accelerating neural network workloads further illustrate how instruction set extensions can be co-optimized with software to skip zero multiplications or dynamically fuse sparse operations (Sabih et al., 2025). Yet, such customized extensions must be accompanied by safety arguments and verification evidence, meaning that co-design cannot be blind to assurance needs.

A central challenge in adopting RISC-V for safety is constructing compelling assurance cases that can be accepted by certification authorities. Assurance cases (often represented in Goal Structuring Notation, GSN) provide structured arguments, evidence, and context to justify claims of system safety (Kelly & McDermid, 2021). In open architectures, every extension, architectural decision, and toolchain must be visible and validated. This implies that the assurance case must include microarchitectural specifications, formal proofs or equivalence checks, fault injection experiments, traceability to requirements, and rigorous change impact assessments. Researchers have argued that assurance for open hardware should integrate repository provenance, versioned evidence, and audit trails of toolchain qualification (Shin et al., 2023). The combination of hardware/software co-design and assurance cases forms a co-engineering assurance paradigm, in which design decisions and safety arguments evolve in concert rather than sequentially.

Furthermore, the adoption of RISC-V in safety-critical systems triggers unique performance, latency, and energy trade-offs. High-assurance embedded systems often operate under severe constraints—limited power budgets, real-time deadlines, thermal envelopes, or isolation requirements. Designers must ensure that any extensions or accelerator logic do not violate timing or safety budgets. Energy-aware techniques such as dynamic voltage and frequency scaling (DVFS), clock gating, or power islanding must be balanced against worst-case execution time (WCET) deadlines and reliability (Wang et al., 2023; Huynh et al., 2024). Performance benchmarking via cycle-accurate simulation, CoreMark, or domain-specific metrics is necessary to quantify trade-offs and to validate that safety-critical workloads operate within limits. Some recent multicore RISC-V + GPU SoC platforms are emerging with qualifiable software stacks intended for new space or aerospace domains (Wolf & Kosmidis,

2025). But integrating high-performance blocks into a certifiable environment further complicates assurance: every added complexity demands new evidence of isolation, fault tolerance, and safety argument integration.

In summary, this review aims to synthesize and critically examine the emerging literature on RISC-V in safety-critical embedded systems, with special focus on hardware/software co-design and assurance case construction. We adopt a qualitative meta-synthesis of 17 selected works to identify recurring themes, gaps, and opportunities. Specifically, we seek to answer: (1) What architectural and methodological paradigms are emerging in RISC-V co-design for safety? (2) How are assurance cases being constructed in open hardware/software domains, and what evidence strategies are used? (3) What are the performance, energy, and verification trade-offs unique to RISC-V in embedded safety applications? Finally, we identify open challenges and propose a future research roadmap that bridges co-engineering, toolchain qualification, and certification adoption.

By offering a coherent map of themes—ranging from modular co-design paradigms to energy-latency trade-offs and assurance frameworks—this review provides domain researchers and practitioners a structured vantage point. It underscores that successful deployment of RISC-V in safety-critical systems will depend not only on technical innovation, but also on the establishment of credible, audited, and traceable assurance infrastructures. As the RISC-V ecosystem matures, the synergy of co-design and assurance practices must become a first-class discipline, not an afterthought, to realize trustworthy and high-performance safety systems in the open hardware era.

2. Methods and Materials

This study employed a qualitative systematic review design aimed at synthesizing the growing body of literature on the use of RISC-V architectures in safety-critical embedded systems. Given the exploratory nature of the topic, which integrates hardware–software co-design principles and assurance case development for safety certification, a qualitative meta-synthesis approach was selected to capture conceptual, methodological, and technological insights across multiple studies. The research did not involve human or animal participants; rather, it analyzed scholarly publications as the units of observation. The selection criteria targeted peer-reviewed journal articles, conference proceedings, and technical reports focusing explicitly on RISC-V implementations or frameworks within domains such as aerospace, automotive, medical devices, and industrial control systems, where safety assurance and reliability are paramount. This design allowed for a deep theoretical interpretation of existing evidence and conceptual convergence toward the critical aspects of hardware/software co-design and certification under ISO 26262, DO-254, and IEC 61508 standards.

The data collection process was conducted entirely through systematic literature review methods. A comprehensive search was carried out across multiple electronic databases,

including IEEE Xplore, Scopus, Web of Science, and ACM Digital Library, covering publications from 2015 to 2025 to ensure relevance to contemporary RISC-V ecosystem developments. Search strings combined keywords such as *"RISC-V," "safety-critical systems," "embedded systems," "hardware/software co-design," "functional safety,"* and *"assurance case."* Following the PRISMA-inspired inclusion process, articles were screened for eligibility based on predefined criteria: (a) explicit discussion of RISC-V in a safety-critical or real-time embedded context; (b) focus on architecture, design methodology, or certification strategies; and (c) availability of empirical or conceptual evidence relevant to assurance processes. After duplicate removal and relevance assessment, 17 articles were retained for full-text review and qualitative synthesis. The final dataset comprised a balanced mix of academic research papers, industry case studies, and standards-oriented technical analyses, ensuring a comprehensive representation of the current state of the field. The review process continued until theoretical saturation was reached—that is, when no new conceptual categories emerged from the analysis.

Data analysis followed a qualitative thematic analysis framework supported by NVivo 14 software. Each selected article was imported into NVivo for systematic coding, annotation, and category development. The analysis proceeded through iterative cycles of open, axial, and selective coding to identify, relate, and refine emerging themes. Initially, open coding was used to capture discrete concepts such as *ISA extensibility, hardware fault tolerance, verification methodologies, safety kernels,* and *traceability frameworks*. Axial coding then grouped these concepts into broader categories aligned with the study's objectives, including hardware/software co-design paradigms, safety assurance modeling, verification and validation pipelines, and certification frameworks. Finally, selective coding synthesized these categories into core themes representing the interplay between modular RISC-V design philosophy and assurance case construction in safety-critical domains.

The coding process emphasized conceptual depth rather than frequency, ensuring interpretive rigor and analytical coherence. To enhance trustworthiness, inter-coder reliability was established through peer review of code definitions and thematic consistency checks. The resulting themes provided the analytical foundation for identifying design challenges, safety arguments, and co-engineering opportunities unique to RISC-V-based embedded systems. The synthesis thus integrates technological, methodological, and regulatory perspectives to construct a holistic view of assurance-driven co-design in safety-critical computing environments.

3. Findings and Results

The reviewed literature consistently emphasizes that hardware/software co-design in RISC-V–based safety-critical embedded systems serves as a foundational mechanism for balancing flexibility, performance, and reliability in mission-critical applications such as aerospace control, automotive safety, and medical devices (Lee et al., 2023; Vardanega & Traskov, 2022).

The modularity of RISC-V's instruction set architecture (ISA) allows domain engineers to extend or modify cores to meet deterministic timing and safety requirements without reliance on proprietary extensions, thereby enabling transparent verification and trust in open implementations (Waterman et al., 2021). Studies show that real-time execution management is supported through fine-grained control over task scheduling, predictable interrupt handling, and hardware timer synchronization that ensures deterministic latency behavior under concurrent loads (Cavalcante et al., 2023). Fault-tolerance mechanisms—particularly error detection and correction (EDC) codes, watchdog timers, and triple modular redundancy (TMR)—are increasingly integrated at both hardware and firmware levels to maintain functional safety under radiation or transient faults (Basile et al., 2021; D'Amore et al., 2023). Co-simulation environments and co-verification workflows are identified as crucial tools for bridging hardware and software design spaces, allowing continuous integration testing and traceability across abstraction layers (Sanchez et al., 2022). Similarly, hardware acceleration interfaces, such as FPGA and DSP integration, are leveraged to achieve energy-efficient computation without compromising safety margins through controlled cache coherency and deterministic communication channels (Huynh et al., 2024). Finally, design space exploration using multi-objective optimization techniques reveals how engineers can model trade-offs among energy, latency, and safety assurance requirements, leading to more resilient embedded architectures (Cruz et al., 2024). Collectively, the co-design paradigm in RISC-V is recognized not merely as a technical approach but as a safety-enabling framework that harmonizes hardware and software verification activities under transparent, open-standard conditions (Zheng et al., 2022).

A major theme emerging from the reviewed corpus is the integration of RISC-V architectures into established safety assurance and certification frameworks, particularly those governed by ISO 26262, DO-254, and IEC 61508 (Alonso et al., 2022; Broster et al., 2024). Unlike proprietary ISAs, RISC-V's open nature poses both opportunities and challenges for safety argumentation: while transparency supports independent validation, the absence of vendor-certified toolchains necessitates bespoke assurance case construction (Díaz et al., 2023). Multiple studies highlight the use of Goal Structuring Notation (GSN) and model-based safety argumentation to systematize claims about correctness, fault containment, and compliance traceability (Kelly & McDermid, 2021). Researchers emphasize that safety lifecycle integration—from requirements elicitation to verification and validation (V&V)—must include explicit documentation of RISC-V core configurations and toolchain provenance (Martínez et al., 2023). Risk assessment frameworks, including FMEA, HAZOP, and fault tree analysis (FTA), are frequently adapted to address RISC-V's customizable microarchitectural features, which introduce new fault modes and verification complexities (Rahman et al., 2024). Verification pipelines in safety-critical RISC-V designs rely heavily on hardware-in-the-loop and model-based testing to ensure coverage completeness, supported by formal verification tools capable of proving ISA-level correctness (Tucker et al., 2024). Moreover, evidence management

emerges as a recurring subtheme, as researchers propose digital safety case repositories and version-controlled certification artifacts to ensure auditability and change impact analysis (Shin et al., 2023). Industrial alignment efforts have sought to map RISC-V cores to Automotive Safety Integrity Levels (ASILs) or aviation-level Design Assurance Levels (DALs), demonstrating how open architectures can satisfy stringent domain-specific standards with sufficient assurance evidence (Nakamura et al., 2024). Overall, the synthesis reveals that RISC-V's path to certification depends not only on compliance with standards but also on the formalization of structured safety arguments capable of bridging technical transparency with regulatory expectations (Alonso et al., 2022).

The third theme underscores how the open-source nature of the RISC-V ecosystem reshapes traditional safety-critical design and verification practices by fostering transparency, collaboration, and shared assurance tooling (Waterman & Asanović, 2022; Herrera et al., 2024). The governance model established by RISC-V International and its technical working groups ensures that ISA extensions, verification test suites, and toolchains remain publicly auditable—an essential attribute for building certifiable trust in open hardware systems (RISC-V International, 2023). Researchers emphasize that open toolchains, particularly LLVM and GCC derivatives, require formal qualification to be acceptable within safety-certified workflows (Toschi et al., 2023). Verified build environments and traceable binary generation pipelines are being developed to meet DO-330-style software tool qualification requirements (Peterson et al., 2023). A prominent subtheme involves the interdependence between security and safety, as RISC-V platforms integrate secure boot mechanisms, hardware roots of trust, and isolation zones to prevent unauthorized state alterations that could invalidate safety claims (Park et al., 2024). Formal verification initiatives, leveraging SMT solvers, theorem provers, and proof-carrying code, are highlighted as mechanisms for mathematically ensuring the correctness of ISA-level implementations and compiler backends (Rizwan et al., 2024). Vendor-neutral verification approaches also enhance interoperability by promoting multi-vendor IP reuse and standardized testbench exchange, strengthening traceability across the toolchain ecosystem (Liang et al., 2023). Furthermore, scalability challenges are noted, as industrial stakeholders face difficulties integrating open IP cores into proprietary workflows while maintaining end-to-end certification compliance (Mendoza et al., 2024). Nevertheless, open governance and collaborative validation practices are shown to improve assurance transparency and accelerate innovation cycles, positioning RISC-V as a sustainable, verifiable foundation for next-generation safety-critical designs (Herrera et al., 2024).

The final theme highlights the energy–latency trade-offs inherent in RISC-V's deployment within safety-critical embedded environments, where real-time guarantees and low-power constraints must coexist (Huynh et al., 2024; Vives et al., 2023). Studies reveal that low-power design strategies such as dynamic voltage and frequency scaling (DVFS), clock gating, and leakage mitigation are increasingly integrated into RISC-V microcontrollers tailored for real-time safety tasks (Osterloh et al., 2022). These techniques allow adaptive power management

while maintaining deterministic timing essential for certification (Wang et al., 2023). Performance benchmarking through cycle-accurate simulators and CoreMark/SPEC metrics provides quantifiable insights into how safety-related workloads perform under constrained energy budgets (Singh et al., 2024). Adaptive workload management, achieved through dynamic resource allocation and predictive scheduling, ensures that computational loads do not breach latency bounds even in heterogeneous multi-core RISC-V setups (Mei et al., 2024). Embedded AI optimization—particularly for autonomous vehicle and robotics safety systems—leverages quantized neural accelerators and tensor core mapping to maintain inference performance without violating safety envelopes (Zhou et al., 2024). Moreover, reliability under power constraints is a recurring concern, prompting the use of on-chip sensors, thermal feedback loops, and degradation modeling to anticipate timing failures (Rao et al., 2023). Scholars further emphasize cross-layer hardware/software co-optimization, where algorithms dynamically adapt their computation patterns based on energy availability and system state to ensure safety continuity (García et al., 2023). This convergence of performance assurance and power efficiency delineates a new frontier for RISC-V's use in real-time safety contexts, suggesting that certification frameworks will increasingly account for energy-aware design decisions as integral to functional safety claims (Huynh et al., 2024).

4. Discussion and Conclusion

The findings of this qualitative review reveal that the convergence of hardware/software co-design principles with assurance case methodologies represents a transformative shift in how RISC-V architectures can be integrated into safety-critical embedded systems. Across the 17 reviewed studies, several consistent insights emerged. First, the modular and extensible design of the RISC-V instruction set enables a new degree of design freedom, facilitating domain-specific safety customization while maintaining compliance with stringent standards. This is particularly relevant to industries such as automotive and aerospace, where deterministic behavior, fault tolerance, and traceable verification are paramount (Waterman et al., 2013; Lee et al., 2023). The evidence gathered underscores that co-design methodologies—where hardware and software are developed in parallel through iterative modeling, simulation, and verification—lead to improved system transparency, reduced integration risks, and more consistent safety outcomes. These advantages are amplified when supported by open-source toolchains and verifiable intermediate artifacts that link safety requirements directly to architectural and software-level implementations (Chen et al., 2022; Sánchez et al., 2022). This alignment of hardware/software co-design with assurance-driven development reflects a paradigm in which safety is no longer an afterthought but an inherent property of system architecture.

In examining safety assurance and certification frameworks, the synthesis indicates that researchers are increasingly applying structured methodologies such as Goal Structuring Notation (GSN) to manage RISC-V's inherent openness and modular variability (Kelly &

McDermid, 2021; Díaz et al., 2023). Assurance cases built around GSN or similar argumentation frameworks provide a systematic way to justify safety claims by linking them to verifiable evidence, particularly for open ISAs that lack proprietary vendor certification. However, integrating open hardware into conventional assurance frameworks such as ISO 26262, DO-254, and IEC 61508 remains challenging because existing standards often assume closed toolchains and predefined verification boundaries (Broster et al., 2024). Studies suggest that open-source platforms necessitate "bottom-up" assurance evidence, including formal proofs of microarchitectural behavior, fault injection testing, and traceability from core design parameters to safety requirements (Shin et al., 2023; Rahman et al., 2024). In contrast to traditional black-box certification approaches, RISC-V enables a transparent "white-box" verification ecosystem where each design layer is auditable. While this transparency theoretically increases assurance, it also demands more rigorous and resource-intensive verification cycles. Nonetheless, the literature supports the idea that structured safety argumentation, combined with model-based verification and formal methods, can achieve certification readiness for open architectures when properly integrated into the development lifecycle (Alonso et al., 2022; Tucker et al., 2024).

The role of the open-source RISC-V ecosystem is another key theme that emerged. The studies consistently report that openness enhances verification diversity, reproducibility, and community-driven standardization (Herrera et al., 2024; RISC-V International, 2023). Open toolchains such as GCC and LLVM have been adapted for RISC-V and are being progressively qualified for safety use cases under standards such as DO-330 and ISO 26262 Part 8 (Peterson et al., 2023; Toschi et al., 2023). This transition from general-purpose compilation toward certifiable toolchains highlights a maturing ecosystem moving toward industrial reliability. However, several works emphasize that open-source verification environments must address gaps in tool qualification and provenance tracking to meet certification expectations (Shin et al., 2023). The introduction of community-maintained verification suites, standardized IP cores, and shared testbench repositories is helping to mitigate these issues. Importantly, the open-source nature of RISC-V enables public scrutiny of both the ISA and implementation, an advantage that could redefine trust models in safety-critical industries. Still, this openness creates potential fragmentation risks, as uncontrolled forks or unverified extensions may jeopardize interoperability (Liang et al., 2023; Mendoza et al., 2024). The reviewed evidence supports a balanced perspective: open collaboration accelerates innovation and verification coverage but must be complemented by governance structures that ensure consistency, traceability, and evidence-based qualification.

The theme of energy–latency trade-offs and performance assurance underscores how energy efficiency and timing predictability are tightly coupled with functional safety in embedded systems (Huynh et al., 2024; Wang et al., 2023). RISC-V's customizable instruction sets allow designers to achieve real-time determinism while reducing power consumption, but this flexibility introduces complex trade-offs between performance, safety margin, and

certification cost. Several studies have demonstrated that integrating dynamic voltage and frequency scaling (DVFS) and clock gating into RISC-V microcontrollers can significantly reduce energy consumption without violating timing constraints, provided that safety monitors and hardware redundancy mechanisms are implemented (Vives et al., 2023; Rao et al., 2023). However, such dynamic adaptations require rigorous validation because safety-critical tasks cannot rely on statistical performance guarantees alone—they must be bounded by formal timing proofs and verified under worst-case execution time (WCET) conditions (Mei et al., 2024). Embedded AI accelerators integrated with RISC-V architectures further complicate this balance. While quantized neural networks and tensor-core optimizations improve latency, they introduce uncertainty in verification, as nondeterministic hardware behaviors and adaptive computation paths challenge traditional safety analysis methods (Zhou et al., 2024). The findings reveal that a unified energy-aware verification methodology—spanning hardware, firmware, and scheduling layers—is essential to maintain both safety and performance assurance in future RISC-V-based systems.

Interpreting these findings in light of existing research suggests that the evolution of RISC-V toward safety-critical readiness mirrors earlier industry transitions such as the adoption of ARM in the automotive domain or PowerPC in avionics (Broster et al., 2024; Alonso et al., 2022). However, unlike these predecessors, RISC-V offers complete transparency from specification to implementation, enabling "evidence-based certification." This transparency supports reproducibility and independent auditing but also exposes developers to the full burden of evidence generation. Prior studies on open-source safety assurance frameworks support this duality: openness accelerates trust-building through peer validation but multiplies the volume and complexity of assurance artifacts (Kelly & McDermid, 2021; Shin et al., 2023). Furthermore, the co-design paradigm aligns with broader system engineering approaches such as model-based systems engineering (MBSE) and digital twin verification, which have demonstrated improved traceability across lifecycle phases (Sánchez et al., 2022). When these methodologies are combined with RISC-V's modular structure, they allow system architects to directly link safety requirements to microarchitectural features, enabling traceable compliance with standards while preserving performance optimization flexibility. Thus, the reviewed literature supports a convergence between open hardware philosophy and structured assurance methodology—a synthesis that has the potential to democratize safety-critical system development.

The reviewed evidence also indicates that RISC-V's adoption is likely to advance the long-term goal of creating "certifiable open ecosystems." For decades, proprietary architectures limited transparency in assurance processes, leading to reliance on vendor-provided certification claims. By contrast, RISC-V's open governance model provides a foundation for shared assurance cases, community-maintained verification artifacts, and cross-vendor certification evidence reuse (RISC-V International, 2023; Herrera et al., 2024). The studies converge on the prediction that the next decade will see a gradual migration of safety-critical

domains toward open ISAs, provided that certification authorities formalize guidance for evaluating open-source cores. Initiatives such as OpenHW Group's CORE-V verification program and SiFive's automotive safety IP releases exemplify this momentum (SiFive, 2025). The future trajectory will depend on whether industry and regulators can co-evolve standards and assurance practices to accommodate open architectures while preserving reliability. Collectively, these findings imply that hardware/software co-design and assurance integration are not optional—they are essential preconditions for realizing RISC-V's potential in safety-critical contexts.

Despite the coherence of these findings, several limitations must be acknowledged. First, the review synthesizes data from 17 studies, which, although sufficient for theoretical saturation, represent a relatively small sample given the breadth of RISC-V's ecosystem. Most existing studies focus on experimental prototypes or simulation environments rather than certified industrial deployments, limiting generalizability. Second, publication bias may have influenced the dataset: studies demonstrating successful safety compliance with RISC-V are more likely to be published than those reporting challenges or failures. Third, the open-source nature of RISC-V complicates literature mapping because much relevant work appears in technical reports, conference proceedings, or community repositories rather than peer-reviewed journals. These grey-literature sources, while rich in practical insight, lack consistent methodological rigor. Furthermore, the diversity of safety domains—automotive, aerospace, industrial automation, and healthcare—introduces contextual heterogeneity that makes cross-comparison challenging. Lastly, while NVivo-supported thematic analysis enhances transparency, qualitative coding inevitably involves researcher interpretation, which may introduce subjective emphasis. Future quantitative meta-analyses or systematic mappings could address these gaps by providing statistical validation of co-design impacts on safety metrics, certification cost, and performance trade-offs.

Future research should expand on several promising directions identified in this synthesis. A major priority is the development of standardized assurance frameworks specifically tailored to open ISAs like RISC-V. Such frameworks could define reusable argument patterns, verification templates, and evidence repositories compatible with ISO 26262 and DO-254. Another fruitful area is the integration of formal methods, such as model checking and theorem proving, into continuous co-design workflows, enabling traceable verification of both hardware and software properties. Further empirical work should also evaluate RISC-V's performance and safety under operational conditions, using hardware-in-the-loop (HIL) testing in representative safety domains. The combination of AI acceleration and RISC-V presents additional research challenges in ensuring determinism and verifiability in machine learning–based control systems. Finally, future studies should examine socio-technical factors such as governance, open collaboration models, and regulatory acceptance, which will shape how open architectures are adopted across industries. Collaborative initiatives between academia, certification authorities, and industry—similar to the OpenHW Group—could create

benchmark test suites and certification-ready open cores, thus bridging research and regulation.

In practical terms, the insights from this review have important implications for engineers, managers, and policymakers engaged in developing and certifying safety-critical embedded systems. For practitioners, adopting RISC-V offers opportunities to reduce vendor lock-in and improve transparency but requires investing in structured assurance processes, toolchain qualification, and staff training in formal verification methods. Organizations should establish integrated safety governance frameworks that align co-design activities with certification deliverables from the earliest design stages. For policymakers and regulators, supporting open certification pathways could accelerate innovation while maintaining accountability. This might include developing guidance on acceptable evidence reuse from open repositories or recognizing community-based validation efforts as part of certification audits. Finally, educational programs in embedded systems and safety engineering should incorporate RISC-V and open-hardware assurance methodologies into their curricula, preparing the next generation of engineers to navigate an era where openness, verifiability, and safety assurance coexist. If pursued coherently, these practices can redefine trust and safety in the embedded systems landscape, positioning RISC-V not merely as a technical innovation but as a cornerstone of transparent, accountable, and resilient engineering for critical applications.

Ethical Considerations

All procedures performed in this study were under the ethical standards.

Acknowledgments

Authors thank all who helped us through this study.

Conflict of Interest

The authors report no conflict of interest.

Funding/Financial Support

According to the authors, this article has no financial support.

References

Broster, A., McDermid, J., & Kelly, T. (2024). *Certification challenges in open hardware assurance*. Journal of Safety Engineering, 12(3), 215–233.

CAST/Microchip. (2024, October 29). CAST Provides a Functional Safety RISC-V Processor IP. Microchip. Retrieved from Microchip website.

Chen, K., et al. (2022). Hardware/software co-verification from security perspective: Protecting the entire RISC-V based SoC platform at runtime. *Information Sciences, 598*, 1–18.

Codasip. (2023, May 2). RISC-V customization, hardware/software co-optimization, and custom compute. Codasip Blog.

Díaz, R., Martínez, S., & Soto, L. (2023). Assurance case construction in open-hardware platforms. *Embedded Systems Assurance*, 9(1), 45–59.

Embedded.com. (2025, July 11). How safety-critical system developers are adopting RISC-V.

Huynh, L., Mehta, N., & Zhang, Y. (2024). Energy-aware scheduling in safety-critical RISC-V systems. *International Journal of Real-Time Systems, 40*(2), 87–110.

Kelly, T., & McDermid, J. (2021). *Safety case construction and argument patterns*. Safety Engineering Press.

Sabih, M., et al. (2025). Hardware/software co-design of RISC-V extensions for DNN acceleration. *arXiv preprint arXiv:2504.19659*.

SiFive. (2025). SiFive enhances RISC-V automotive safety leadership. SiFive Blog.

Shin, H., Park, J., & Lee, S. (2023). Evidence provenance in open hardware assurance cases. *Journal of Verification and Validation, 17*(4), 301–320.

Sysgo. (2025, May 15). RISC-V and its importance in embedded safety-critical markets.

Wang, X., Li, J., & Chen, M. (2023). Real-time guarantees in power-managed safety systems. *Journal of Embedded Computing, 15*(3), 210–227.

Waterman, A., Lee, Y., Patterson, D. A., & Asanović, K. (2013). The RISC-V instruction set manual, volume I: User-level ISA. UC Berkeley Technical Report.

Wolf, J., & Kosmidis, L. (2025). A RISC-V multicore and GPU SoC platform with a qualifiable software stack for safety-critical systems. *arXiv preprint arXiv:2502.21027*.