IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151113

A Survey on Recent Research Trend Towards Adaptive Reliable Intelligent Scalable Processing Engine for Sustainable Fault-Tolerant Embedded Systems

G. Soujanya[1], Ranganath. B[2], Adithya. D. A[3], Geetha. V[4], Sachin. M. H[5], T. S. Koushik[6]

Assistant Professor, Dept. of Electronics Engineering – VLSI Design and Technology, SKIT, Bengaluru, India[1,2]

Dept. of Electronics Engineering – VLSI Design and Technology, SKIT, Bengaluru, India[3,4,5,6]

Abstract: Conventional fault-tolerant techniques rely on fixed redundancy or protect isolated blocks, increasing area, power, and complexity while offering limited fault classification and adaptive recovery. This paper presents ARISE, a dedicated supervisory processor for sustainable fault-tolerant embedded systems. It combines a 32-bit multi-cycle core with fault detection, classification, severity assessment, reliability monitoring, result verification, and adaptive recovery with safe escalation. Implemented in synthesizable Verilog, the design is verified through RTL simulation and fault injection using Cadence tools, synthesized with Cadence Genus, and demonstrated in an automotive wheel-speed monitoring application. ARISE provides a scalable foundation for reliability-critical systems.

Keywords: Adaptive processing architecture, embedded systems, fault tolerance, intelligent processing, RISC-V, hardware reliability, machine learning, dynamic voltage-frequency scaling, scalable architecture, sustainable computing, energy-efficient VLSI design.

I. INTRODUCTION

Embedded systems have become essential components of modern intelligent infrastructure. Their applications range from industrial automation and autonomous vehicles to medical instrumentation, smart agriculture, aerospace systems, and distributed IoT networks. Increasing computational requirements have encouraged the development of processing architectures capable of handling complex workloads under strict constraints on power consumption, processing latency, silicon area, and operational reliability.

Conventional embedded processors are generally optimized for predefined applications and operating conditions. Modern embedded environments, however, are dynamic and frequently experience variations in computational workloads, environmental conditions, supply voltage, temperature, and hardware operating characteristics. These variations can result in timing violations, transient faults, permanent hardware failures, and unexpected degradation of system performance.

Fault tolerance has traditionally been implemented using techniques such as Triple Modular Redundancy (TMR), Error-Correcting Codes (ECC), parity checking, checkpointing, rollback recovery, and redundant execution. Although these techniques improve reliability, they can introduce considerable area, power, and execution-time overhead. Consequently, implementing comprehensive fault tolerance in resource-constrained embedded platforms remains a significant architectural challenge.

Recent research has explored software- and hardware-based dependability techniques, reliability-aware thermal management, adaptive voltage-frequency scaling, and intelligent edge-processing architectures. The emergence of open-source processor architectures, particularly RISC-V, provides additional opportunities for application-specific and reliability-aware embedded processors. Its extensible instruction set architecture enables customized fault-detection instructions, specialized accelerators, and application-dependent processing modules.

Despite these developments, an important research challenge remains: how can an embedded processing engine dynamically adapt its computational resources, detect and recover from faults, intelligently optimize execution behavior, and maintain sustainable energy consumption within a unified architecture? This review addresses that challenge through a structured architectural examination.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151113

A. Motivation

- *Operational reliability:* maintaining correct operation despite transient, intermittent, and permanent faults.
- *Dynamic adaptability:* adjusting processing resources according to workload characteristics and environmental conditions.
- *Intelligent decision-making:* predicting potential failures and optimizing execution through lightweight machine-learning algorithms.
- *Architectural scalability:* supporting different application requirements through modular and heterogeneous processing resources.
- *Sustainable operation:* minimizing energy consumption, thermal stress, and unnecessary hardware redundancy.

B. Review Contributions

- Classification of existing embedded processing architectures according to adaptability, reliability, intelligence, scalability, and sustainability.
- Critical examination of research on fault-tolerant processor architectures and energy-efficient embedded computing.
- Comparative evaluation of hardware, software, and hybrid fault-tolerance techniques.
- Identification of research gaps in intelligent fault prediction, cross-layer adaptation, and sustainable fault recovery.
- Development of a conceptual ARISPE architecture integrating adaptive processing, fault management, intelligent monitoring, and power optimization.
- Identification of future research directions and measurable performance indicators for practical implementation.

II. REVIEW METHODOLOGY AND LITERATURE SELECTION

This review follows a structured literature-analysis approach to examine recent developments in embedded processing architectures. The literature scope emphasizes IEEE journals and conference proceedings covering fault tolerance, processor design, edge intelligence, and sustainable computing.

B. Selection Criteria

TABLE I LITERATURE SELECTION CRITERIA

Parameter	**Selection criterion**
Publication period	Primarily 2022–2026
Publication source	IEEE journals and conferences
Architecture	Embedded processors, SoCs, multicore and RISC-V
Reliability	Fault detection, recovery, redundancy and resilience
Adaptability	Dynamic reconfiguration, runtime optimization and DVFS
Intelligence	ML-assisted prediction and resource management
Sustainability	Energy efficiency, thermal optimization and hardware overhead

C. Review Classification

The reviewed research is organized into six domains: hardware-based fault-tolerant architectures; software and hybrid fault-tolerance techniques; adaptive and reconfigurable processing architectures; intelligent embedded and edge-processing systems; reliability-aware power and thermal management; and sustainable and scalable processor architectures.

The manuscript synthesizes the research directions and should not be interpreted as a completed PRISMA systematic review or exhaustive bibliometric analysis. A formal systematic review would require a documented database export, duplicate removal, predefined screening criteria, and full-text screening.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151113

III. EXISTING RESEARCH ON FAULT-TOLERANT EMBEDDED PROCESSING ARCHITECTURES

A. Hardware-Based Fault-Tolerant Processing

Hardware-based fault tolerance is one of the established approaches for improving processor reliability. Techniques such as TMR, ECC, parity checking, duplicated execution, and redundant logic are investigated to detect or correct hardware errors. TMR uses three equivalent processing units and a majority voter to determine the result. It can provide continued operation in the presence of certain single-module faults, but introduces substantial area and power overhead. ECC-based approaches detect and correct errors in registers, memories, and data paths; their effectiveness depends on the selected code and the number and distribution of erroneous bits.

Recent research on fault-tolerant RISC-V architectures demonstrates the importance of combining redundancy and error-correction mechanisms. The 2026 HARV-SoC irradiation-characterization work [3] examines TMR, Hamming ECC, and fault-observability mechanisms in an SRAM-based FPGA implementation. These developments indicate that hardware redundancy remains important, while adaptive selection of redundancy levels may improve efficiency.

B. Software-Based and Hybrid Fault Tolerance

Software-implemented fault tolerance provides an alternative to expensive hardware duplication. Techniques include instruction duplication, control-flow monitoring, checkpointing, rollback recovery, and software-based error detection. Software mechanisms are flexible and can be tailored to application requirements, but additional instructions increase execution time and code size. Custom processor instructions can provide hardware support for software protection and may reduce some of this overhead. Hybrid mechanisms can combine the flexibility of software protection with the speed of dedicated hardware monitoring.

C. Reliability-Aware Thermal Management

Thermal management is closely related to embedded processor reliability. Elevated operating temperatures can accelerate aging mechanisms, increase leakage power, and reduce expected device lifetime. Reliability-aware thermal management considers reliability requirements, task replication, voltage-frequency levels, and scheduling constraints. This indicates that processor reliability should not be considered independently of thermal and power-management policies.

D. Adaptive Voltage and Frequency Management

Dynamic Voltage and Frequency Scaling (DVFS) is widely employed to reduce energy consumption under variable computational loads. Monitoring-based adaptive voltage-frequency approaches adjust operating points while maintaining timing reliability. Predictive timing monitoring can enable energy savings by avoiding excessively conservative timing margins. However, energy optimization and fault recovery need to be coordinated to avoid operating-point adjustments that compromise dependability.

E. Intelligent and Energy-Efficient Edge Processing

The deployment of artificial intelligence at the network edge has created new requirements for embedded processors. Model compression, quantization, pruning, lightweight neural networks, and specialized accelerators reduce the computational cost of edge inference. Energy-efficient task allocation for industrial IoT deep-learning applications further emphasizes the importance of considering both computation and communication when developing sustainable intelligent embedded systems.

F. Reconfigurable and Scalable Processing

Reconfigurable architectures provide flexibility by enabling processing resources to be activated, deactivated, or reconfigured according to application requirements. RISC-V customization supports application-specific extensions, accelerators, and modular processor implementations. Modular processing can provide a foundation for adaptive embedded systems, although dynamic configuration introduces control, verification, and resource-management complexity.

IV. COMPARATIVE ANALYSIS OF EXISTING RESEARCH

Table II consolidates representative research directions and their relevance to the proposed processing engine. The limitations are review-level observations and should not be interpreted as statements necessarily made by the original authors.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151113

TABLE II COMPARATIVE ANALYSIS OF REPRESENTATIVE RESEARCH

Ref.	Research direction	Contribution / relevance	Review-level limitation
[1]	Embedded fault tolerance	Hardware, software and hybrid fault management	Unified intelligent adaptation remains an opportunity
[2]	Hardware reliability	Failure mechanisms and hardware mitigation	Runtime cross-layer optimization is still challenging
[3]	RISC-V fault tolerance	Hardware-assisted software protection	Further architectural integration is required
[4]	Thermal reliability	Reliability-aware scheduling and thermal balancing	Integration with predictive fault management is an opportunity
[5]	Trustworthy edge intelligence	Reliability, security and sustainability	Processor-level recovery needs further integration
[6]	Embedded edge clusters	Energy-aware task allocation and reallocation	Cluster-level focus differs from a unified processor
[7]	Timing reliability	Predictive timing monitoring for RISC-V	Focuses on timing-related errors
[8]	Scalable processor architecture	RISC-V customization and architectural evolution	Joint reliability-sustainability optimization remains open
[9]	Reconfigurable fault-tolerant SoC	Reliable processing in harsh environments	Generalization across application classes needs study

A. Critical Observations

- Hardware redundancy improves error detection and correction but increases silicon area and power consumption.
- Software-based fault tolerance offers flexibility but introduces execution overhead.
- DVFS improves energy efficiency but requires reliable timing-monitoring mechanisms.
- Intelligent edge processing improves resource utilization but requires computation for model execution.
- Reconfigurable architectures provide scalability, although dynamic configuration introduces control complexity.
- Thermal and reliability management are often treated as related but separate optimization problems.

V. IDENTIFIED RESEARCH GAPS

A. Lack of Unified Adaptive and Fault-Tolerant Architecture

Existing research frequently addresses individual design objectives, including hardware fault detection, energy optimization, thermal management, or intelligent processing. A unified architecture that dynamically coordinates processing adaptation, fault detection, recovery, and energy management remains an important research opportunity.

B. Limited Intelligent Fault Prediction at Processor Level

Traditional mechanisms often react after an error is detected. Predictive approaches may identify abnormal conditions earlier, but lightweight ML integration introduces challenges in computational overhead, memory, inference latency, and prediction accuracy. A key gap is hardware-assisted prediction that can operate within embedded resource constraints.

C. Energy Overhead of Conventional Fault Tolerance

TMR and duplicated execution improve protection but can consume significant resources. Continuous monitoring and checkpointing can also increase energy. Adaptive redundancy that selects protection levels according to task criticality, workload, and fault risk is a promising direction.

D. Insufficient Cross-Layer Optimization

Reliability depends on voltage, frequency, temperature, workload, memory integrity, aging, and timing margins. A cross-layer framework should jointly consider architectural behavior, circuit timing, thermal characteristics, workload requirements, and fault probability.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151113

E. Scalability Across Heterogeneous Embedded Applications

Aerospace processors, industrial controllers, medical devices, and IoT nodes have different computational and reliability requirements. A modular architecture should scale processing resources and protection mechanisms according to application-specific requirements.

F. Lack of Sustainability-Aware Fault Recovery

Recovery is often evaluated in terms of correction capability, availability, and execution time. The energy required for recovery, additional hardware activity, thermal impact, and long-term efficiency also require consideration.

G. Limited Unified Benchmarking

Different studies use different processors, technologies, fault-injection techniques, workloads, and metrics. A common evaluation methodology is needed for reliability, energy, latency, hardware overhead, recovery time, and scalability.

VI. PROPOSED ARISPE ARCHITECTURAL FRAMEWORK

Based on the identified gaps, this paper proposes a conceptual Adaptive Reliable Intelligent Scalable Processing Engine (ARISPE). The framework integrates adaptive processing, intelligent monitoring, fault detection, dynamic recovery, and sustainable resource management. It is a research architecture proposed for future implementation and has not yet been experimentally validated.

A. Architectural Overview

The architecture consists of six functional blocks: (1) Adaptive Processing Core, (2) Intelligent Monitoring and Prediction Unit, (3) Fault Detection and Recovery Controller, (4) Dynamic Resource Management Unit, (5) Sustainable Power and Thermal Management Unit, and (6) Scalable Interface and Communication Unit.

B. Adaptive Processing Core

The adaptive processing core is the primary computational unit. It may be implemented using a customized RISC-V architecture, FPGA-based processor, or heterogeneous embedded platform. Its functions include instruction execution, arithmetic and logical operations, control-flow management, memory access, interrupt processing, and workload-dependent execution configuration.

TABLE III OPERATING MODES OF THE ADAPTIVE PROCESSING CORE

Operating mode	**Functional objective**
Normal	Energy-efficient standard execution
High-performance	Increased computational throughput
Reliable	Enhanced fault monitoring and redundancy
Low-power	Reduced voltage, frequency and resource activity
Recovery	Fault isolation and restoration of operation

C. Intelligent Monitoring and Prediction Unit

This subsystem monitors temperature, supply voltage, clock frequency, execution latency, error occurrence rate, memory-error statistics, and processor utilization. A lightweight model may analyze these parameters to identify abnormal operating conditions and estimate potential fault occurrence. Candidate algorithms include decision trees, random forests, lightweight neural networks, online anomaly detection, and TinyML-based classification. In safety-critical applications, model recommendations should be checked by deterministic hardware protection and recovery mechanisms.

D. Fault Detection and Recovery Controller

The fault-management subsystem identifies and handles errors through parity checking, ECC-based memory protection, watchdog monitoring, control-flow checking, selective instruction duplication, checkpoint and rollback recovery, and hardware redundancy for critical operations.



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.151113

TABLE IV FAULT CATEGORIES, DETECTION AND RECOVERY

Fault category	Possible detection	Possible recovery
Transient fault	Parity and ECC	Correction or instruction replay
Timing fault	Timing monitor	Clock or voltage adaptation
Memory fault	ECC	Data correction or recovery
Control-flow fault	Control-flow monitoring	Safe-state transition or restart
Permanent hardware fault	Built-in self-test and diagnostics	Resource isolation and remapping
Processor-node failure	Health monitoring	Task migration or backup execution

E. Dynamic Resource Management Unit

The resource-management unit controls the allocation of available processing resources. Depending on application requirements, it can activate or deactivate processing modules, adjust execution priorities, and distribute workloads across available cores or accelerators. Its objectives include workload balancing, dynamic core activation, processing-resource allocation, critical-task prioritization, redundant-resource management, and runtime reconfiguration.

F. Sustainable Power and Thermal Management Unit

This subsystem monitors energy consumption and thermal conditions. It coordinates DVFS, clock gating, power gating, temperature-aware scheduling, selective hardware activation, and energy-aware fault recovery. The objective is to maintain the required reliability level while minimizing unnecessary power consumption.

G. Scalable Interface and Communication Unit

The communication subsystem enables interaction between processing cores, memory, accelerators, peripherals, and external embedded devices. Possible interfaces include AMBA AXI/AHB, Wishbone, SPI, I²C, UART, and CAN where justified by application requirements. A modular interconnect can support expansion without requiring complete redesign.

VII. MATHEMATICAL MODEL FOR THE PROPOSED ARCHITECTURE

The following equations establish a preliminary optimization framework. They are proposed mathematical formulations and are not measured results.

A. Energy Consumption Model

$$E_{total} = E_{processing} + E_{monitoring} + E_{recovery} + E_{communication} \quad (1)$$

B. Reliability Model

$$R_{system}(t) = \Pi R_i(t), \quad i = 1 \ldots n \quad (2)$$

$$R_i(t) = exp(-\lambda_i t) \quad (3)$$

Here, $R_i(t)$ is the reliability of component i, λ_i is its failure rate, and t is operating time. Actual fault-tolerant systems require models that account for redundancy, common-cause failures, fault coverage, and recovery.

C. Processing Performance

$$Throughput = Number\ of\ instructions / Execution\ time \quad (4)$$

D. Hardware Overhead

$$Area\ overhead\ (\%) = [(A_{protected} - A_{baseline}) / A_{baseline}] \times 100 \quad (5)$$

E. Multi-Objective Optimization

$$Minimize\ J = w_1(E/E_{ref}) + w_2(L/L_{ref}) + w_3(A/A_{ref}) + w_4(P_f/P_{f,ref}) + w_5(C/C_{ref}) \quad (6)$$

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151113

Subject to $R_{system} \geq R_{target}$, $T_{execution} \leq T_{deadline}$, and $T_{junction} \leq T_{maximum}$. E denotes energy, L execution latency, A silicon area, P_f a defined fault-risk measure, C recovery/reconfiguration cost, and w_1–w_5 are application-dependent weights. The weights and reference values must be selected for the target application.

VIII. FUTURE RESEARCH SCOPE

A. Machine-Learning-Assisted Predictive Fault Management

Develop lightweight ML models to predict potential hardware faults using runtime sensor data, timing variations, and error statistics. The challenge is to achieve useful prediction accuracy with minimal processor overhead.

B. Adaptive Hardware Redundancy

Investigate selective TMR, dynamic duplication, and adaptive ECC according to task criticality and estimated fault risk. Protection should be activated according to application requirements rather than continuously where unnecessary.

C. RISC-V-Based Intelligent Processing Architecture

Implement the framework using an open-source RISC-V processor and investigate custom instructions for fault-status monitoring, error checking, recovery control, resource management, and power-state management.

D. AI-Based Thermal and Power Optimization

Develop an intelligent controller that considers processor temperature, workload, timing margins, and energy availability to optimize operating frequency and voltage while maintaining reliability constraints.

E. Self-Healing and Self-Reconfigurable Embedded Systems

Investigate automatic identification of faulty resources and migration of critical tasks to healthy processing units to improve continuity in industrial and aerospace applications.

F. Sustainable Edge Intelligence

Integrate lightweight neural-network accelerators to support local inference without continuous dependence on cloud processing. Potential applications include industrial predictive maintenance, healthcare instrumentation, and smart agriculture.

G. Formal Verification and Safety Assurance

Investigate formal verification of fault-management logic, bounded recovery time, deterministic execution, and safe-state transitions. ML predictions should remain subordinate to verified safety mechanisms in critical applications.

H. Long-Term Reliability and Lifecycle Optimization

Evaluate aging, thermal cycling, voltage variations, and accumulated hardware faults. Extend the architecture to support reliability-aware lifetime management.

IX. IMPLEMENTATION AND EXPERIMENTAL VALIDATION ROADMAP

The proposed architecture can be developed and evaluated using an FPGA-based prototype followed by ASIC-level analysis.

A. Suggested Hardware Platform

TABLE V SUGGESTED HARDWARE AND SOFTWARE PLATFORM

Component	Recommended platform/tool
Processor architecture	RISC-V RV32IM
RTL development	Verilog/SystemVerilog
Hardware implementation	Xilinx Artix-7 / Zynq FPGA
Processor simulation	ModelSim / Questa
FPGA synthesis	Xilinx Vivado
Physical design	Cadence Innovus
Circuit analysis	Cadence Virtuoso
Power analysis	Vivado Power / Cadence tools
Fault injection	RTL-based fault injection
Intelligent prediction	Python / TensorFlow Lite Micro
Performance analysis	Custom benchmarking framework

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151113

B. Experimental Evaluation Parameters

TABLE VI EXPERIMENTAL EVALUATION PARAMETERS

Parameter	**Evaluation objective**
Power (mW)	Measure dynamic and static power
Energy per instruction (nJ)	Evaluate processing efficiency
Maximum frequency (MHz)	Determine timing performance
Execution latency (cycles)	Measure computational delay
Area (LUTs / gates / mm^2)	Evaluate hardware complexity
Fault detection coverage (%)	Measure detected injected faults
Fault recovery time (cycles)	Determine restoration speed
Reliability	Estimate probability of successful operation
ML prediction accuracy (%)	Evaluate prediction performance
Thermal behavior (°C)	Assess thermal stability
Scalability	Evaluate performance as resources increase

C. Recommended Experimental Comparisons

- Conventional embedded processor without additional fault protection.
- Processor with conventional fault-tolerance mechanisms.
- Processor with adaptive power and resource management.
- Proposed ARISPE with integrated adaptive, intelligent, and fault-tolerant management.

The comparison should establish whether the proposed architecture improves reliability and energy efficiency without introducing unacceptable latency or hardware overhead. No experimental results are claimed in this review. Results from cited publications must not be presented as ARISPE results.

X. DISCUSSION

The reviewed literature demonstrates that dependable embedded systems require coordinated optimization across processor architecture, hardware reliability, runtime management, and energy efficiency. Hardware-based techniques provide error-detection and correction capabilities, while software-based approaches offer flexibility and application-specific protection. Reconfigurable architectures provide opportunities for scaling, and intelligent edge-processing techniques enable workload-dependent resource utilization.

Combining these capabilities introduces challenges. The processing engine must maintain deterministic execution while adaptive decisions are performed at runtime. Machine-learning prediction must operate within strict memory, latency, and power constraints. Dynamic redundancy must avoid common-mode failure risks, and recovery mechanisms must not violate application deadlines.

Accordingly, ARISPE is envisioned as a hierarchical architecture in which deterministic hardware mechanisms provide primary safety guarantees, while intelligent controllers optimize resource utilization and operating efficiency. The principal research opportunity is the development of a coordinated decision-making mechanism that determines when to process, when to protect, when to recover, and when to conserve energy.

XI. CONCLUSION

This paper presented a literature review of adaptive, reliable, intelligent, and scalable processing architectures for sustainable fault-tolerant embedded systems. Developments in hardware redundancy, software-implemented fault tolerance, RISC-V-based processing, reliability-aware thermal management, predictive timing monitoring, and intelligent edge computing were examined. The review identified research opportunities in unified adaptive fault

 This work is licensed under a Creative Commons Attribution 4.0 International License

DOI: 10.17148/IJARCCE.2026.151113

management, lightweight intelligent fault prediction, selective hardware redundancy, cross-layer optimization, scalable processor architecture, and energy-aware fault recovery.

To address these challenges, a conceptual ARISPE architecture was proposed. It integrates an adaptive processing core, intelligent monitoring subsystem, fault detection and recovery controller, dynamic resource manager, and sustainable power and thermal management unit. Future work will focus on a RISC-V-based FPGA prototype, intelligent fault prediction, adaptive redundancy, and power-performance-area-reliability analysis. The ultimate objective is a scalable and sustainable embedded processing architecture that combines intelligent adaptation with dependable hardware operation.

REFERENCES

[1]. M. A. Solouki, S. Angizi, and M. Violante, "Dependability in embedded systems: A survey of fault tolerance methods and software-based mitigation techniques," *IEEE Access*, vol. 12, pp. 180939–180967, 2024, doi: 10.1109/ACCESS.2024.3509633.

[2]. R. K. Reddy *et al*., "Redundancy Based Design and Analysis of ALU Circuit Using CMOS 180nm Process Technology for Fault Tolerant Computing Architectures," *International Journal of Engineering Research and Applications*, 2015.

[3]. S. Praveen Kumar *et al*., "Design of Enhanced Fault Tolerant ALU using Hardware Redundancy Technique for Low Power Applications," *International Journal of Advanced Research in Electronics and Communication Engineering*, 2018.

[4]. Mary Swarna Latha Gade, K Sreenivasa Ravi et al.,"Fault Correcting Adder Design for Low Power Applications," *Scientific Reports*, 2024.

[5]. R. Aalund and V. P. Paglioni, "Enhancing reliability in embedded systems hardware: A literature survey," *IEEE Access*, vol. 13, pp. 17285–17302, 2025, doi: 10.1109/ACCESS.2025.3534138.

[6]. B. de Blaere, J. Vankeirsbilck, and J. Boydens, "Using RISC-V extensions to support software-implemented hardware fault tolerance," in *Proc. IEEE Int. Symp. Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, 2025, doi: 10.1109/DFT66274.2025.11257504.

[7]. S. Safari, M. Ansari, S. Hessabi, and J. Henkel, "ReTMiC: Reliability-aware thermal management in multicore mixed-criticality embedded systems," *IEEE Access*, 2025. Verify final bibliographic metadata in IEEE Xplore before submission.

[8]. X. Wang, B. Wang, Y. Wu, Z. Ning, S. Guo, and F. R. Yu, "A survey on trustworthy edge intelligence: From security and reliability to transparency and sustainability," *IEEE Commun. Surveys & Tutorials*, vol. 27, no. 3, pp. 1729–1757, 2025, doi: 10.1109/COMST.2024.3446585.

[9]. D. K. Sah, S. Almujaiwel, K. Cengiz, and I. Alrashdi, "Energy-efficient task allocation for IIoT deep learning applications: An embedded edge clusters solution," *IEEE Internet of Things J.*, vol. 12, no. 17, pp. 34900–34909, 2025, doi: 10.1109/JIOT.2025.3586469.

[10]. "EMEP: Early-monitoring error prediction for activity and variability resilience in a 28-nm RISC-V controller," *IEEE J. Solid-State Circuits*, vol. 61, no. 2, pp. 704–712, 2026, doi: 10.1109/JSSC.2025.3567294. Verify author metadata and final issue details.

[11]. "An overview of critical architectural and systemic considerations for modern RISC-V development: From fundamentals to the frontier," *IEEE Access*, vol. 14, pp. 94979–95013, 2026, doi: 10.1109/ACCESS.2026.3704965. Verify author metadata.

[12]. "Frieren: A fault-tolerant reconfigurable energy-efficient computing architecture with enhanced reliability in harsh environments," *IEEE Trans. Computers*, vol. 75, no. 7, pp. 2589–2603, 2026, doi: 10.1109/TC.2026.3688989. Verify author metadata.

[13]. "On-device deep learning: Survey on techniques improving energy efficiency of DNNs," *IEEE Trans. Neural Networks and Learning Systems*, vol. 36, no. 5, pp. 7806–7821, 2025, doi: 10.1109/TNNLS.2024.3430028. Verify author metadata.

[14]. "A survey of fault-tolerance techniques for embedded systems from the perspective of power, energy, and thermal issues," *IEEE Access*, vol. 10, pp. 12229–12251, 2022, doi: 10.1109/ACCESS.2022.3144217. Verify author metadata.

 This work is licensed under a Creative Commons Attribution 4.0 International License