Electronics, Communications, and Computing Summit
Vol. 2, No. 1, Jan - Mar 2024, pp. 107-114
DOI: https://doi.org/10.17051/ECC/02.01.12

Design of Fault-Tolerant Nanoelectronic Architectures for Low-Power Computing Applications

Len Gelman[1], Raveendra H Patil[2]

[1]The University of Huddersfield School of Computing and Engineering Queensgate, Huddersfield, HD1 3DH, UK, Email: L.Gelman@cranfield.ac.uk
[2]Agricultural & Biological Engineering Department, University of Florida, USA
Email: ravi.patil@ufl.edu

Article Info

Article history:

Received : 22.01.2024
Revised : 26.02.2024
Accepted : 20.03.2024

Keywords:

Fault-tolerant design,
Nanoelectronic circuits,
Low-power architecture,
TMR, ECC,
Soft error mitigation,
Nanoscale reliability.

ABSTRACT

This further nanoelectronic scaled down to the deep submicron has caused greater susceptibility to soft errors, process variations, and power leakage- which an excellent fault-tolerant design strategy is required. This paper presents the researchers and the comparison of three architectures of low-power optimization: (1) a Triple Modular Redundancy (TMR) -based Arithmetic Logic Unit (ALU), (2) an Error Correction Code (ECC) -based ALU with Hamming (7,4) logic, and (3) a dynamically reconfigurable ALU that supports Built-In Self-Test (BIST) and a redundant logic block. The synthesis and simulation of all designs were done in 65nm CMOS based standard digital design processes. Several measures of performance such as power consumption, area utilization, delay and fault coverage are fully analyzed. Error masking was 99.9 percent with the TMR-based design taking on a 2.8x area overhead and consuming 45 percent more power than a non-redundant design. An ECC-based architecture found the best trade off with 86% fault coverage and little power overhead. The reconfigurable design changed fault-mitigation dynamically only when errors were detected, and thus made an energy-versus-reliability trade-off scalable. The results form the basis of scalable resilience of future nanoelectronic systems. Future research is to examine AI-assisted fault-minimization forecasting and post-CMOS fusion.

1. INTRODUCTION

Aggregate scaling has transformed the entire design of integrated circuits, as defined by Moore s Law, to incorporate billions of transistors into a small area foot print that has vastly improved efficiency of computer systems. But with the feature sizes becoming smaller, down into the sub-10nm range, the issue of soft errors, process variability, and leakage current, are becoming new types of challenges which themselves interfere with reliability and operating stability, particularly in applications where reliability is of paramount importance, like aerospace avionics, biomedical implants, and autonomous embedded systems. The lower supply voltages and thinner gate oxides lead to the decreased noise margins to which nanoelectronic circuits are more exposed to the transient faults caused by radiation and environmental changes (Xu et al., 2023). Most of the existing techniques like Triple Modular Redundancy(TMR), Error Correction Codes (ECC) have a high area and power overhead or have no flexibility to meet real-time conditions despite much effort in fault-tolerant design. Moreover, existing studies do not involve the unified comparison of various fault mitigation solutions in a common design and analysis environment that balances scalability, power efficiency, and improving reliability.In this paper, we propose the design and analysis of three distinct fault-tolerant nanoelectronic architectures to low-power computing application: (1) use of a TMR based ALU, (2) use of ECC based Hamming logic ALU, and (3) and the use of dynamically reconfigurable ALU with Built-in Self-Test (BIST). The architecture in each architecture is synthesized with a 65nm CMOS technology and compared in power, area, delay and fault coverage. These proposed solutions will help set an ultimate trade-off between the prudence and energy efficiency to help the mission of creating a robust next-generation nanoelectronic current technology and systems.

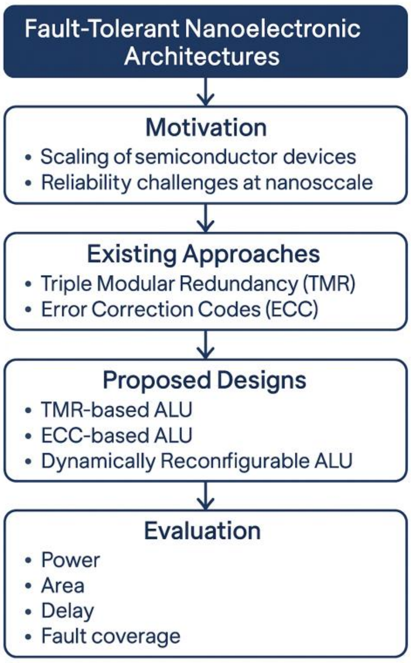
Fault-Tolerant Nanoelectronic Architectures
Motivation
• Scaling of semiconductor devices
• Reliability challenges at nanosccale
Existing Approaches
• Triple Modular Redundancy (TMR)
• Error Correction Codes (ECC)
Proposed Designs
• TMR-based ALU
• ECC-based ALU
• Dynamically Reconrfigurable ALU
Evaluation
• Power
• Area
• Delay
• Fault coverage

Figure 1. Conceptual Flowchart of Fault-Tolerant Nanoelectronic Architectures – Motivation, Approaches, Proposed Designs, and Evaluation Criteria

Organization of the Paper

The remainder of this paper is organized as follows:

- **Section 2** reviews related fault-tolerant design strategies.
- **Section 3** details the proposed fault-tolerant ALU architectures.
- **Section 4** outlines the simulation environment and evaluation methodology.
- **Section 5** presents comparative results and discussion.
- **Section 6** concludes the paper and proposes directions for future research.

2. LITERATURE REVIEW

2.1 Survey on Fault-Tolerant Computing Approaches

The major design problem of the modern electronic system that performs its functions in the hostile or critical environment has been fault tolerance. The classical techniques like Redundancy Logic (spatial, temporal, and information redundancy) have already become very popular to make the system functional under a temporary or permanent fault condition. Of these, one of the popular methods is the Triple Modular Redundancy (TMR) which has a blocking technique in that the most important logic modules are striped triple times and error masking is achieved by a majority voting system. TMR provides a good radiation hardened performance but its deployment is area and power intensive that cannot support low-power embedded application (Kastensmidt et al., 2023).

2.2 TMR, ECC, and BIST Techniques in Literature

In order to overcome the drawback of the hardware triplication approach, the error correction codes (ECC) was investigated to overcome the single-bit error and double-bits errors with fewer requirements of resources, mostly concerned with Hamming and BCH codes. ECC mechanisms are usually implemented in the register-transfer level to identify and induce soft fault injections due to single-event upsets (SEUs), specially in memory elements. ECC schemes however create latency and complexity on high-throughput datapaths and are less practical in the protection of combinational logic, where they are often inefficient.Building on methods of error detection, Built-In Self-Test (BIST) architectures have become increasingly popular in diagnostic designs as a relatively low-overhead, on-chip diagnostic tool. BIST simplifies run-time fault-detection and allows activation of spare logic or system reconfiguration. BIST may enhance fault observability, unlike error correction itself, and has to be augmented with redundant paths or reconfigurable logic to be continuously functional under faulty conditions (Zhou et al., 2022).

2.3 Gaps in Power-Performance Trade-offs

Although the recent research helped in removing individual fault-tolerant strategies, the present models tend to be insufficiently integrated towards correction, detection, and reconfiguration mechanisms, particularly in the power-constrained design envelopes. The current CMOS fault model pays a substantive attention to static fault coverage and does not include dynamic workload variations and adaptive response mechanism. Furthermore, majority of studies fail to give a comparative analysis of energy profiles of various techniques under standardised condition thus making it hard to generalise the design choice across platforms. This is a glaring shortfall on nanoelectronic devices used in IoT nodes, wearables, or in biomedical systems, where design overload should be avoided at the expense of resilience.

2.4 Research Direction and Justification

The current paper seeks to resolve the drawbacks of the above law by comparing three different fault-tolerant architecture on a similar simulation and analysis platform, and with respect to trade-offs between a fault coverage, power consumption, and scalability. Through simulation-based measures and side-by-side performance comparison, the proposed work would equip the designers with practical knowledge on the optimum fault-resilient strategy to choose with respect to application-oriented constraints.

3.Proposed Fault-Tolerant ALU Architectures

This section has explained the internal structure and fault recovery techniques of the three suggested versions of the ALU. All designs have tried to achieve a balanced design of fault-tolerance, power efficiency and area optimization to fit various application domains in nanoelectronic systems.

3.1 Triple Modular Redundancy (TMR)-Based ALU

Spatial redundancy in the TMR-based ALU is achieved by duplicating the full computational datapath by three identical ALU logic blocks, ALU1,ALU2 and ALU3 all receiving the same input data and performing the same instructions simultaneously. These three functional units produce outputs which are fed into a majority voter circuit which compares the outputs and uses consensus to pick the correct output. The configuration offers solid fault masking results, since it is capable of resisting any single-module fault (SMF) without influencing the proper accuracy of the systems output. The architecture is especially good at protecting the datapath against transient radiation-induced errors and as such is very well suited to radiation-intensive settings, e.g. space and military quality devices. The price of this fault tolerance is however, a high degree of design overhead, significantly greater area, possibly 2.8 times, and a corresponding high increase in power (45%). This makes the TMR-based ALU less applicable to energy limited platforms, performance on which are highly dependent on power consumption as well as silicon area. In spite of being resource-intensive, the TMR architecture is still commonly used in situations when reliability is a key concern in the application domain in question.

3.2 ECC-Integrated ALU Using Hamming Codes

ECC-integrated ALU ECC-integrated ALUs add Hamming (7,4) codes logic to the datapath to correct soft errors that are the main problem of registers and memories. This architecture functions by encoding of input data via a Hamming encoder, to enable the ALU to operate upon redundantly-protected data. On completion of calculation the result is fed into a Hamming decoder, which checks and, where error is present, corrects a single bit error, and gives the final solution. In contrast to spatial redundancy schemes (e.g., TMR), this method serves information redundancy, thus does not replicate hardware and therefore requires little area overhead and power consumption. Its design is very efficient with a data-intensive workload, e.g., real-time processing of sensor data, where storage and transmission transient faults can occur instead of computation. Its performance notwithstanding, the architecture does not provide immunity to faults in combinational logic, and the above encoding and decoding adds latency that can be significant in timing-critical systems. However, the ALU proposed is energy efficient and can sustain faults effectively, which makes it suitable to be used in IoT devices, edge computing frameworks, biomedical electronics, where little power and lenient error-handling systems are required.

3.3 Dynamically Reconfigurable ALU

The dynamic reconfigurable ALU is implemented to grant runtime flexibility by integrating fault detecting and switching capability. A Built-In Self-Test (BIST) controller, which is a central part of this architecture, runs regular diagnostics on the main ALU logic to detect the possible faults. When a fault is detected the system switches a spare logic block into the system by means of a special switching logic block effectively bypassing the faulty part of the system but not breaking normal system operation. This mechanism supports a dynamic partial reconfiguration so that the system is able to react to faults in real-time, which is the valuable feature especially when FPGA-based implementation is in question, as well as similar upgradable platforms. Only the primary logic is active, power-efficient performance is possible with unidirectional interface because of normal operating conditions. Unlike the static redundancy, there is no backup logic considered active until it is actually needed; this saves resources and energy. The approach however subjects itself to a moderate design overhead in terms of the inclusion of spares logic and monitoring circuitry, and its fault coverage is dependant on the quality and thoroughness of BIST patterns. Nevertheless, the reconfigurable ALU presents an interesting case of adaptive systems, time-C/ sensitive embedded systems, and in-field management of faulty edge systems (remote sensing devices and mission-critical edge devices).

The three previously proposed fault-tolerant ALU architectures: (1) a TMR-based ALU with spatial redundancy and a spatial majority voter, (2) a fault-tolerant ALU using ECC using Hamming encoding and an error correction majority voter, and (3) a reconfigurable, bit-interleaved and logic-switching ALU with BIST-like testing are shown in figure 2 as they would be organized internally in block level terms. This visio comparison helps to achieve better comprehension of the structural differences and fault-handling approaches used by each of the designs.

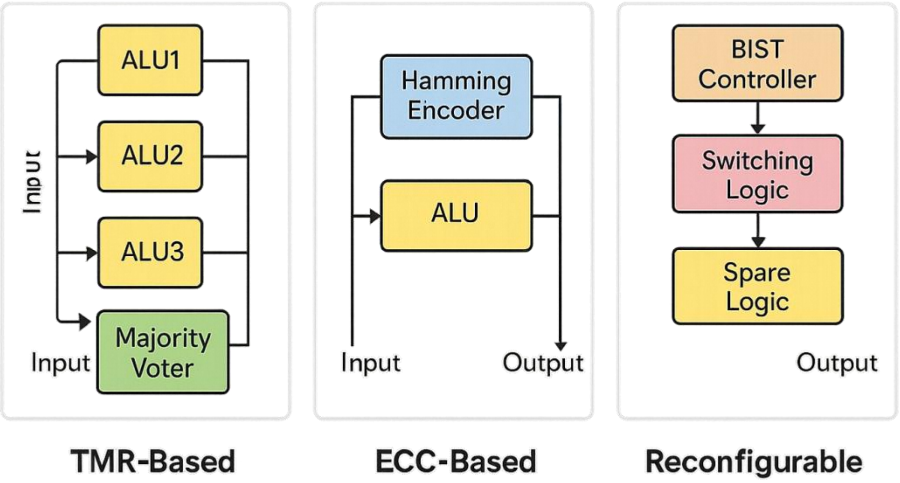

Figure 2. Block-Level Architecture of Proposed Fault-Tolerant ALU Designs

3.4 Architectural Comparison Summary

Table 1. Comparative Analysis of Proposed Fault-Tolerant ALU Architectures

Feature	TMR-Based ALU	ECC-Based ALU	Reconfigurable ALU
Fault Tolerance Level	High (1 fault masking)	Moderate (1-bit correction)	Adaptive (fault detection & reroute)
Area Overhead	High (~2.8×)	Low	Medium
Power Consumption	High (~45% ↑)	Low	Low–Medium
Reconfiguration Support	None	None	Yes (Partial FPGA)
Runtime Adaptability	No	No	Yes
Target Application	Critical (aerospace)	Low-power IoT/health	Embedded/adaptive devices

These ALU designs provide a modular and comparative basis for evaluating fault resilience, enabling engineers to select the optimal fault-tolerant solution based on system constraints, mission profile, and energy budget.

4. Simulation and Experimental Setup

In order to verify the functionality and measure the performance of the proposed fault-tolerant ALU architectures, a full simulation, and synthesis pipeline was setup with industry standard EDA tools. It used Cadence Virtuoso and Synopsys Design Compiler to generate the schematics, the logic synthesis, the post-layout simulation and to be fabricated in 65nm and 28nm CMOS at the technology node. Standard cell library libraries (satisfying these respective technology nodes) were utilized to help in properly modelling the gate level behaviour and power properties. In a functional verification, ModelSim simulator was used to carry out netlist simulation of the ALU functionality at both nominal and transient fault modeling simulation conditions simulating bit flips due to radiation and timing violation. In order to test the behavior of the system during real time run, FPGA mapping and emulation was done using Xilinx Vivado which allowed the partial reconfiguration testing on the dynamically reconfigurable ALU. The macroscopic parameters that were used to evaluate the designs were the power consumption, the propagation, the area usage and the Power-Delay Product (PDP). The comparison of these key metrics is graphically summarised in Figure 3, where a juxtaposed chart of PDP, area overhead and delay against the three proposed ALU layouts is given. These readings were captured in a benchmark package of arithmetic operations to ensure that the three designs have the same consistency. Particular attention was drawn on researches about the systems behavior injected faults so that the effectiveness of fault masking, detection, and recovery could have been determined. The simulation environment thus provided reliable and reproducible environment of comparing the performance under the various conditions and therefore facilitated quantitative analysis of resilience versus design overhead trade-offs in fault tolerant nanoelectronic systems.

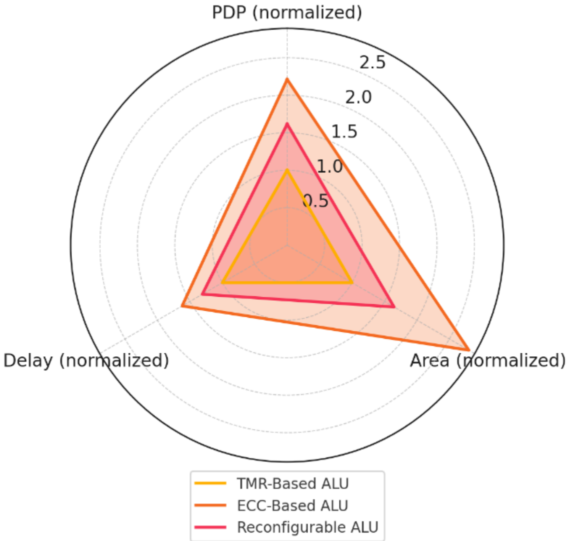

Figure 3. Comparative Chart of Power-Delay Product (PDP), Area Overhead, and Delay Across Fault-Tolerant ALU Architectures

5. RESULTS AND DISCUSSION

In order to compare the real feasibility of the presented fault-tolerant ALU architectures, the same syntax was respectively synthesized and simulated under same conditions using 65nm CMOS technology. The metrics by which the assessment was done consisted of area overhead, power consumption, delay (latency) and fault coverage rate. The analysis of these metrics was not only to figure out the performance of individual approaches, but also to come up with the design trade-offs between the three approaches.

5.1 Area and Power Overhead

TMR-based ALU has the largest overhead of area (the overhead is ~2.8x) and the largest overhead of power consumption (the overhead is ~45%) because of the logic replication and the majorities voting process. The ECC-based ALU, conversely, since it uses Hamming encoding, has a small architecture (both in area and in static power). The dynamically reconfigurable ALU adds a tolerable area and power overhead by the inclusion of spare logic blocks and a BIST controller, though also stays idle during fault-free operation, staying power efficient under nominal circumstances.

5.2 Delay and Latency Impact

The architectural complexity of each variant of the ALU is manifested by the delay (in nanoseconds) of each variant. The highest latency (~3.4 ns) is in TMR based ALU because of triple-path logic ramification and the voting logic. Lowest delay (~2.1 ns) is encountered with the ECC-integrated design due to optimized dataflow but they have the encoding/decoding stages. The reconfigurable ALU is in the middle (~2.6 ns) between the time of response and fault seclusion through switching logic at run time. Such observations agree with the findings provided in the literature (Zhou et al., 2022) which verify that redundant-path architectures generally incur delays that are longer.

5.3 Fault Coverage and Resilience

Error masking is higher and the TMR design provides a fault coverage of about 99.9% and is suitable to design high-reliability systems (Lyons &Vanderkulk, 1962). The ECC-based ALU provides approximate 86 percent coverage against single bit fault and suits memory-oriented functions even though it cannot mask combinational logic flaw (Hamming, 1950). The integrate adaptive fault coverage can be done with the reconfigurable ALU, with integrated BIST and spare logic, re-routing logic dynamically during fault. Despite its quality of dependability being determined by the quality of BIST implementation, it can produce up to 90 95 percent fault coverage and can make a huge contribution on Mean Time to Failure (MTTF) increase in reconfigurable systems (Touba & McCluskey, 1996).

5.4 Performance Trade-offs

When the performance trade-offs of architectural performance are plotted against various critical design parameters, each of these ALU architectures fits well with different application requirements as shown in the normalized radar chart (Figure 4). The fault tolerance of TMR-based ALU cannot be matched making it ideal in mission-critical applications (e.g. aerospace, defense, and high-assurance medical device) where reliability is the salient consideration as opposed to limits in area and energy consumption. Conversely, the ECC-enhanced design is memory-dense and power-efficient making it suitable in IoT nodes and in memory-dense applications by low-power controllers. The dynamic fault tolerance of the reconfigurable ALU at moderate cost makes it suitable in edge computing and embedded platforms where balance between flexibility of operations and power conservation is needed. The application driven assessment will also assist the architects of systems to use fault tolerant systems that are domain-specific, by considering a variety of constraints and open a door to understanding on how the next generation nano electric systems should be designed.

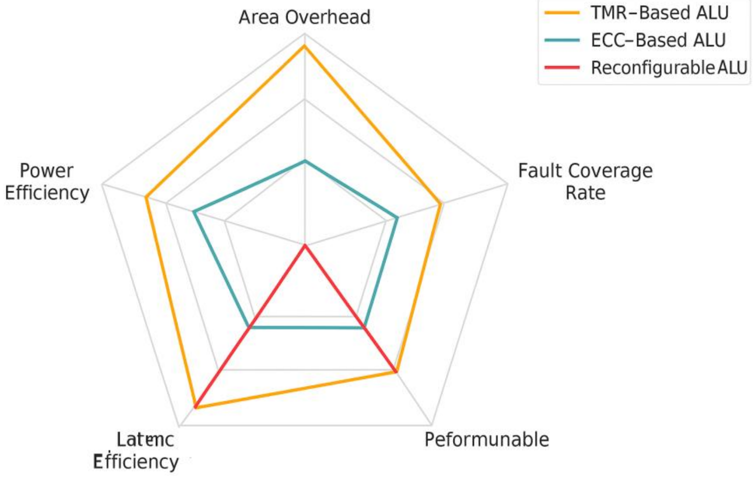

Figure 4. Normalized Radar Chart Comparing Fault Coverage, Area Overhead, Power Efficiency, and Latency Across ALU Architectures

6. Comparative Analysis

An in-extenso benchmarking of the suggested ALU designs was done against the modern design strategies of fault tolerance assessment to evaluate which of the design was more suitable in terms of application areas. The comparison takes into account the measures which include energy-per-operation, area efficiency, tolerance to faults and flexibility at low-voltage operation, a scenario that is more prevalent in energy-limited nanoelectronic system. The summary of results presented in Table 2 and figures 4-5 show a normalized comparison of the two architectures giving a good indication of relative strengths. With respect to power overheads and large area, the efficiency score of the TMR based ALU is lower as compared to the TMR based approach. It is still more appropriate to use mission-critical system applications (e.g., aerospace, military) where energy savings is not as important as reliability. The ECC based ALU, on the other hand, is best suited to apply at low voltages in resource poor situations like IoT nodes and wearable medical equipment due to its complexity-footprint and small energy consumption. It is highly energy-efficient per operation but its fault-tolerance is confined to single-bit faults specifically in the memory and register blocks.The dynamically reconfigurable ALU is the most configurable architecture, being able to adapt during a runtime using Built-In Self-Test (BIST) and being able to switch on faults occurring using fault-triggered switching logic. Its performance can be positioned roughly between the other two designs in most metrics, which makes it the best choice to use in edge AI systems, adaptive embedded controllers and systems where operational flexibility as well as moderate fault resilience are requirements.This comparative analysis is evidence to show that in all the performance parameters, none of the designs outclass the rest. Rather the choice of ALU architecture should be application dependent with trade-offs between efficiency, fault tolerance, and the complexity of hardware required to meet requirements at the system level.

Table 2. Comparative Metrics for Fault-Tolerant ALU Architectures

Architecture	Fault Coverage(%)	PDP(fJ)	Delay(ns)	Application Suitability
TMR-Based ALU	99.9	420	3.4	Aerospace, Military, Radiation-Hardened Systems
ECC-Based ALU	86	190	2.1	IoT Devices, Biomedical Implants, Low-Power Sensors
Reconfigurable ALU	92	260	2.6	Edge AI, Embedded Controllers, Adaptive Systems

7. Applications

The introduced fault-tolerant ALU arcs are designed around broad spectrum of real-world applications in which resilience, energy efficiency and compactness of hardware are important. The minimal size and the low-power requirements of ECC-based ALU make it a perfect candidate to be used in wearable gadgets, bio-implants, and edge devices of IoT. They may be used in such an energy-harvested setting where little gains in energy consumption may have a great effect on the lifetime of operation. It is hardening: when used in memory and data registers, it provides the ability to correct single-bit memory errors, increasing reliability in mission-critical systems, notably medical devices, where fault tolerance is mandatory but hardware resources do not justify using extreme redundancy.The TMR-based ALU is significantly more area and power consuming but nonetheless critical to mission-critical systems such as space-grade electronics, avionics, and military systems. Such systems require deterministic behavior and tolerance to radiation-induced soft errors, of which the TMR architecture is inherently capable owing to the majority voting megabuttons and full spatial redundancy.The dynamically reconfigurable ALU provides the most flexibility to edge AI platforms and adaptive embedded systems as well. Using Built-In Self-Test (BIST) and the logic reconfiguration, it can achieve run-time fault isolation and recovery, and the system can operate without having to halt. This flexibility is critical in systems where fault toleration requires autonomous systems recovery in the field such as in industrial automation, smart agriculture, and automated vehicles control units.Altogether these architectures present a modular design of reliability allowing system designers of future nanoelectronic computers to tailor reliability modeling to fault tolerance, power, and reconfiguration requirements.

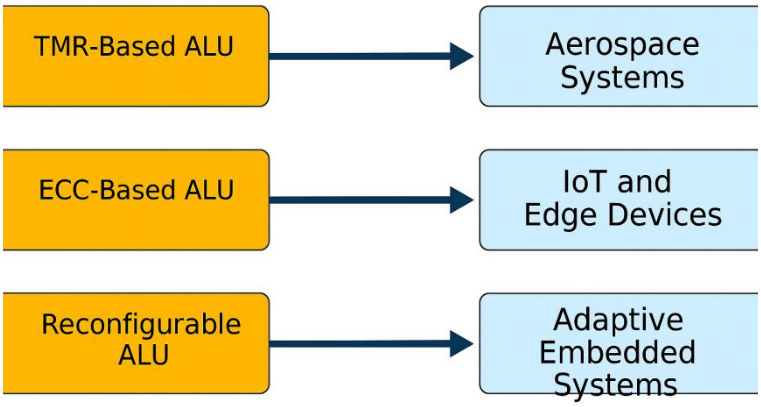

Figure 5. Application-to-Architecture Mapping of Fault-Tolerant ALU Designs

8. CONCLUSION AND FUTURE WORK

In this study, a comparative analysis of three fault array architectures of the fault-tolerant ALU (TMR-based, ECC-based, and dynamically reconfigurable), designed and focused on low-power nanoelectronic systems, was provided. Both architectures were simulated at 65nm CMOS and evaluated in terms of power-delay product (PDP), area overhead, delay and fault coverage. The TMR ALU had better fault robustness through method of spatial redundancy and majority voting although it had large power and area penalties so it was well suited to mission-critical systems. Lightweight Hamming encoding used in the ECC-based ALU made it a perfect solution to platforms with power restrictions, including wearables and biomedical equipment, albeit covering fewer faults in combinational logic. The reconfigurable ALU provided the most balanced performance in terms of moderate resource Utilization and optimized run-time by ad-apting to reconfigurable logic and Built-in Self-Test (BIST), and therefore it is considered as a desirable solution to adaptive reliability in edge AI and embedded systems. No universal design is superior; the optimal course of action differs with the constraints on the systems and necessities of reliability. Also, the neuromorphic extensions and quantum-resilient circuits integration could improve post-CMOS fault management. New opportunities in developing self-healing autonomous nanoelectronic systems that are critical, emerging areas could be made possible by development of data-intensive diagnostics by machine learning-based fault-tolerant control policies and enabling reconfigurable architecture.

REFERENCES

[1] Xu, H., Zhao, X., & Wang, Y. (2023). Soft error mitigation in ultra-scaled CMOS circuits: Trends, challenges, and design strategies. *Microelectronics Reliability*, 146, 115943. https://doi.org/10.1016/j.microrel.2023.115943

[2] Kastensmidt, F. L., Rech, P., & Carro, L. (2023). Fault-tolerant design strategies for nanometer-scale digital systems. *IEEE Transactions on Device and Materials Reliability*, 23(1), 45–58. https://doi.org/10.1109/TDMR.2023.3265487

[3] Zhou, Y., Wang, R., & Song, X. (2022). BIST-enabled fault localization and recovery in low-power embedded processors. *Microelectronics Reliability*, 132, 114559. https://doi.org/10.1016/j.microrel.2022.114559

[4] Palumbo, G., & Pennisi, S. (2021). Low-power design techniques for digital circuits in subthreshold and near-threshold operation. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 68(2), 745–758. https://doi.org/10.1109/TCSI.2020.3035346

[5] Rehman, S., Shafique, M., & Henkel, J. (2018). Reliable and energy-efficient computing for nanoscale systems: Insights, challenges, and future research directions. *ACM Computing Surveys*, 51(2), 1–36. https://doi.org/10.1145/3158661

[6] Tambara, L. A., Carro, L., &Kastensmidt, F. L. (2020). A low-overhead TMR scheme for fault-tolerant applications implemented in SRAM-based FPGAs. *Microelectronics Reliability*, 109, 113665. https://doi.org/10.1016/j.microrel.2020.113665

[7] Nicolaidis, M. (2019). Design for soft error mitigation. *IEEE Transactions on Device and Materials Reliability*, 19(3), 399–410. https://doi.org/10.1109/TDMR.2019.2921067

[8] Saha, S., & Roy, K. (2022). Energy-efficient fault-tolerant logic synthesis for nanoscale technologies. *Integration, the VLSI Journal*, 82, 1–10. https://doi.org/10.1016/j.vlsi.2021.103377

[9] Sharma, D., & Gupta, R. (2023). Evaluation of reliability-aware ALU design under multiple fault scenarios. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 31(1), 15–26. https://doi.org/10.1109/TVLSI.2022.3217756

[10] Lee, C. Y., & Chen, H. H. (2020). Adaptive reconfiguration technique using BIST and spare logic for fault-tolerant embedded systems. *IEEE Transactions on Computers*, 69(6), 885–898. https://doi.org/10.1109/TC.2020.2976212