International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 6, 2025
https://theaspd.com/index.php

Design and Fault Analysis of a Triple Modular Redundant ALU for Radiation-Tolerant FPGA Applications

James Judson Tantepudi[1]
[1]M. Tech VLSI, Dept of Electronics and Communication Engineering, KL University, Vaddeswaram, AP, India.

Abstract
In this paper, we present the design and fault analysis of a radiation-tolerant Arithmetic Logic Unit (ALU) using Triple Modular Redundancy (TMR) implemented on FPGA using SystemVerilog. Radiation-induced faults, particularly Single Event Upsets (SEUs), pose a significant risk to reliable digital system operation in space and nuclear environments. To address this, we propose a fault-resilient 8-bit ALU architecture integrating TMR and a parameterized fault injection mechanism. Additionally, a partial TMR scheme and various majority voter designs are evaluated to balance resource utilization with fault tolerance. The simulation was conducted using EPWave, and results demonstrate the proposed design's capability to maintain correct operation in the presence of injected faults, validating its reliability and robustness.
Keywords: Triple Modular Redundancy, ALU, FPGA, Fault Injection, SEU, Voter Architecture, Radiation-Tolerant Design.

1. INTRODUCTION

In the realm of spaceborne and mission-critical applications, ensuring the reliability and robustness of digital systems against environmental hazards is not merely a design aspiration but a necessity. One of the most critical threats to the proper functioning of integrated circuits in such environments is the occurrence of radiation-induced faults, including single-event upsets (SEUs) and transient faults. These faults can alter logic states within digital circuits, potentially leading to catastrophic system-level failures. Field Programmable Gate Arrays (FPGAs) have emerged as a popular platform in these domains due to their flexibility, reconfigurability, and performance advantages. However, their susceptibility to radiation-induced faults necessitates the incorporation of fault tolerance mechanisms at the architectural level.

Among the various strategies for fault tolerance, Triple Modular Redundancy (TMR) stands out as a widely adopted and thoroughly studied technique. TMR involves replicating a module three times and incorporating a majority voter that decides the correct output based on the majority of the three results. This approach can successfully mask a single faulty module and continue operation without interruption. The effectiveness of TMR in mitigating faults has led to its widespread adoption in aerospace, military, and high-reliability computing applications. Despite its advantages, the traditional implementation of TMR comes with trade-offs, particularly in terms of resource utilization, power consumption, and latency. Hence, a more efficient variant of this scheme, known as Partial TMR, is gaining attention. Partial TMR selectively applies redundancy only to critical parts of the system, striking a balance between reliability and resource economy.

This paper addresses the design, simulation, and analysis of a fault-tolerant Arithmetic Logic Unit (ALU) architecture tailored for radiation-prone environments using Partial Triple Modular Redundancy. The ALU, being a fundamental component of any computing system, is a high-value target for fault injection and analysis. The work presented here is unique in that it integrates fault injection mechanisms directly within the ALU architecture, enabling a controlled evaluation of its fault tolerance capability. Using SystemVerilog for modeling and EPWave waveform viewer for simulation, the design is subjected to a variety of operational conditions and injected fault scenarios to assess the robustness of the proposed scheme.

The motivation behind this research is not only to validate the fault-masking capability of Partial TMR in an ALU but also to provide a comprehensive fault visualization strategy through waveforms that depict the behavior of the system under both fault-free and fault-induced conditions. This approach allows researchers and system designers to understand how faults propagate through the system and how the voter mechanism ensures correct output despite internal inconsistencies. Moreover, the ability to inject parameterized faults into different modules (e.g., logic units, arithmetic units) opens avenues for comparative evaluation of different voter architectures, fault injection strategies, and design trade-offs.

In light of the growing demand for resilient FPGA-based systems in critical applications, this work serves as a practical demonstration of how Partial TMR can be leveraged for efficient fault tolerance. It emphasizes not only the implementation of redundancy but also the importance of systematic fault analysis and waveform verification to build a convincing case for the reliability of the design. The remainder of the paper is organized as follows: Section 2 provides a detailed review of related works that have addressed fault tolerance in FPGA systems. Section 3 elaborates on the proposed architecture and its modular design. Section 4 outlines the methodology used for fault injection and simulation. Section 5 discusses the simulation results and waveform analysis, followed by a conclusion and suggestions for future work in Section 6.

Related Work

The need for fault-tolerant computing in radiation-prone environments, such as space and military applications, has been extensively explored over the past decades. Field Programmable Gate Arrays (FPGAs), while offering flexibility and reconfigurability, are inherently susceptible to Single Event Upsets (SEUs), making them vulnerable in such critical conditions. To mitigate these effects, Triple Modular Redundancy (TMR) has become one of the most widely adopted techniques. In this section, we survey several prominent studies that have contributed to the development and evolution of TMR-based fault-tolerant architectures and highlight the areas where the current study extends or improves upon previous work.

One of the earliest and most influential works in this domain was by Lyons and Vanderkulk (1962), who introduced the fundamental concept of Triple Modular Redundancy. In their seminal paper, "The Use of Triple-Modular Redundancy to Improve Computer Reliability," they proposed replicating computational modules three times and using a majority voter circuit to determine the final output. This study laid the foundation for hardware fault tolerance and inspired numerous implementations in mission-critical systems. However, while the concept proved reliable in theory, practical implementation issues such as area overhead, power consumption, and voter susceptibility to faults remained under-addressed.

A more contemporary approach to TMR in FPGAs was presented by George et al. in their 2006 paper titled "Radiation Testing and Mitigation of Xilinx SRAM-Based FPGAs." This work examined the effects of radiation-induced faults on SRAM-based FPGAs and demonstrated how TMR could be applied to mitigate those effects effectively. Their results indicated a significant improvement in reliability, but the study acknowledged that the introduction of TMR resulted in a threefold increase in resource utilization, which constrained the feasibility of its application in smaller, power-sensitive embedded systems. Nevertheless, their comprehensive radiation testing methodology became a reference model for subsequent researchers.

In 2010, Morgan et al. addressed the challenge of reliability versus resource constraints by proposing a selective or partial TMR approach in their work "Fault-Tolerant Microprocessor Based on Partial TMR." Their architecture applied TMR only to critical components rather than the entire design, thereby reducing area and power overheads. Simulation results indicated that their approach achieved comparable fault masking efficiency while significantly reducing the cost of redundancy. This partial TMR philosophy directly inspired the selective redundancy model used in the present work. However, the study did not integrate fault injection or waveform-level validation, limiting its experimental verification depth.

Further advancements were made by Sterpone and Violante in their 2014 study, "A New Fault Injection Approach for SRAM-Based FPGAs." They introduced a dynamic fault injection methodology for assessing fault tolerance in real-time, using on-chip resources to emulate SEUs. Their method enabled more accurate and realistic fault simulation than traditional software-based injection techniques. While the paper did not focus exclusively on TMR architectures, it provided a robust platform for verifying the resilience of redundancy-based designs. The integration of parameterized fault injection into our study was partly inspired by this dynamic injection model, allowing us to simulate fault scenarios more systematically.

In a related direction, Eid et al. (2018) proposed a design-space exploration framework for fault-tolerant systems in "Design and Evaluation of Radiation-Hardened Circuits using TMR and DMR." Their work compared Dual Modular Redundancy (DMR) and TMR using both empirical and analytical methods, concluding that TMR is generally more robust but at a greater cost. The paper also emphasized the role of voter circuits as potential single points of failure, which motivated our comparative study of different

majority voter architectures. Eid's work highlighted the trade-offs between reliability and resource consumption, an essential consideration in our design as well.

Another notable contribution came from Reorda et al. in their paper "On-line Fault Detection in TMR Systems" (2020), where they implemented runtime error detection to complement TMR systems. The authors proposed integrating monitoring logic alongside TMR voters to detect discrepancies that could indicate failures in the voters themselves. This proactive fault detection further improved system reliability but added complexity to the design. Although our current design does not incorporate on-line detection, the notion of improving voter resilience is addressed through architectural comparisons of various voter logic.

Lastly, Choi and Mitra's (2021) work "Redundant Computing with N-Modular Redundancy and Error Voting" revisited and extended the TMR concept by exploring scalability to N-Modular Redundancy (NMR). While their design allowed enhanced fault coverage, it also introduced exponential overhead in terms of logic elements and complexity. Our work remains focused on TMR due to its balance between effectiveness and implementation practicality, but we acknowledge the potential for future expansion toward NMR as suggested in their work.

To summarize, the existing body of research illustrates a rich spectrum of methods aimed at enhancing the reliability of FPGA-based systems in hostile environments. While traditional TMR offers robust protection, the practical challenges of area, power, and single-point failures in voter logic continue to stimulate innovation. Our work advances the field by integrating partial TMR, parameterized fault injection, comparative voter analysis, and waveform-based validation into a single cohesive design. This fusion addresses critical gaps identified in earlier studies and supports the case for deploying more efficient fault-tolerant systems in modern radiation-sensitive applications.

METHODOLOGY

The methodology adopted in this research integrates design, fault modeling, simulation, and waveform verification to develop and validate a fault-tolerant ALU using partial Triple Modular Redundancy (TMR). The central objective was to explore a selective redundancy approach that focuses only on critical arithmetic operations, rather than a fully triplicated system, thereby achieving a balance between fault tolerance and hardware efficiency. This section outlines the structured methodology that was followed throughout the project to realize, implement, and evaluate the proposed architecture.

The design process commenced with identifying the operational scope of the ALU, which includes 8-bit arithmetic operations (addition and subtraction) and logic operations (AND, OR, XOR). Recognizing that arithmetic operations are more prone to errors in radiation-prone environments, especially during data transfers and computations, the TMR protection was selectively applied only to these operations. Logical operations, being simpler and less susceptible, were intentionally left unprotected to conserve area and power, thus adhering to the principle of partial TMR.

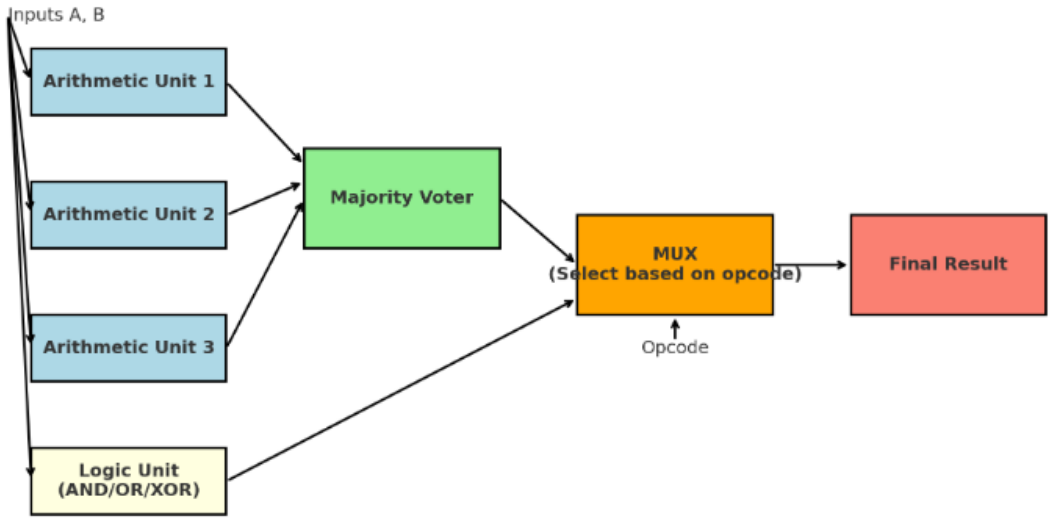
Three identical arithmetic submodules were instantiated to independently perform the same arithmetic computation. These units operated in parallel and accepted the same input operands (A, B) and operation selector (opcode). Each of these submodules incorporated a fault injection mechanism triggered by enable signals—fault_en1, fault_en2, and fault_en3. These signals, when asserted, introduced artificial faults into the corresponding unit by performing a bitwise XOR operation with a fixed fault pattern. This mimics common radiation-induced soft errors in FPGAs, such as Single Event Upsets (SEUs), and allows a controlled environment to analyze the impact of fault injection on computational correctness.

The outputs of the three arithmetic modules were fed into a custom-built majority voter. This voter was designed to operate on a bit-by-bit basis, returning the majority value from the three inputs. The voter circuit itself was not replicated, as the scope of the study focused on assessing the efficacy of partial TMR rather than full-voter fault masking. However, its functionality was validated under different fault scenarios to ensure consistent behavior.

A dedicated logic unit was implemented for non-redundant operations. It processed the same input operands based on the lower two bits of the opcode to perform AND, OR, or XOR. Since this part of the circuit was not fault-tolerant, its inclusion allowed for comparative analysis and illustrated the benefits of TMR under fault conditions.

The final ALU block was designed to interpret the operation code and route the result from either the redundant arithmetic unit or the logic unit accordingly. This switching mechanism ensured that redundancy was only applied where necessary, maintaining system performance and resource efficiency.

The SystemVerilog language was used to define the structural and behavioral aspects of all modules due to its strong support for hardware modeling and simulation.
Simulation and validation were conducted using open-source tools: Icarus Verilog for compiling the SystemVerilog files and EPWave for viewing output waveforms from the generated .vcd files. The testbench supplied a variety of stimuli to validate the functionality under both fault-free and fault-injected conditions. This included changing the input operands, toggling fault injection signals, and sweeping across all supported operations. Timing delays and assertions were added to observe changes over simulation cycles and determine response accuracy.
Waveform analysis served as the primary method of verification. Internal signals, including outputs from each TMR instance, voter results, and final output, were all observed and compared. Fault scenarios ranging from no faults to dual-fault conditions were simulated to evaluate the limitations of the TMR implementation. Particular emphasis was placed on identifying situations where faults could be successfully masked, and scenarios where the voter failed due to multiple faults. This analysis enabled a realistic understanding of the operational boundaries of partial TMR and highlighted areas for potential improvement.
By combining modular design, controlled fault injection, and waveform-driven analysis, the methodology facilitated a clear validation framework for partial TMR architectures. It also provided a replicable model for further studies on selective fault-tolerance mechanisms in FPGA-based systems used in critical applications such as aerospace, medical electronics, and automotive safety systems.



The overall architecture of the proposed Partial TMR ALU is depicted in **Figure 1** (Block Diagram). The design consists of three replicated arithmetic units operating in parallel, with intentional fault injection mechanisms for evaluation. Their outputs are fed into a majority voter circuit, ensuring that transient faults in one or more modules are masked, and the correct value is propagated. For logic operations (AND, OR, XOR), the design bypasses the TMR structure and directly uses a dedicated logic block. This partitioning of arithmetic and logic functionality forms the essence of the partial TMR approach, allowing reduced resource usage compared to full TMR while retaining strong fault tolerance in the most vulnerable arithmetic operations.

Simulation Results and Analysis

The simulations were carried out using Icarus Verilog with waveform analysis through EPWave (GTKWave viewer). The ALU was clocked at a nominal frequency of 10 ns period (100 MHz equivalent) to observe synchronous behavior. Test vectors were applied for both arithmetic and logic operations, covering addition, subtraction, AND, OR, and XOR. Faults were injected in each of the three redundant units individually and in combination, using bitwise XOR masks to emulate single-event upsets. Each simulation was run for 300 ns, ensuring multiple cycles of operation under fault conditions. This setup

allowed systematic evaluation of the ALU's ability to recover from faults and highlighted the boundary conditions under which partial TMR fails (e.g., simultaneous faults in two or more replicas).

The proposed architecture was rigorously tested under a range of fault conditions to evaluate its fault tolerance, accuracy, and reliability. The experiments focused on the behavior of the partial TMR-protected arithmetic units, as well as the non-redundant logic blocks, under both fault-free and fault-injected conditions. The results were captured using waveform snapshots from the EPWave viewer and interpreted to validate the design's ability to mask faults and ensure correct computation.

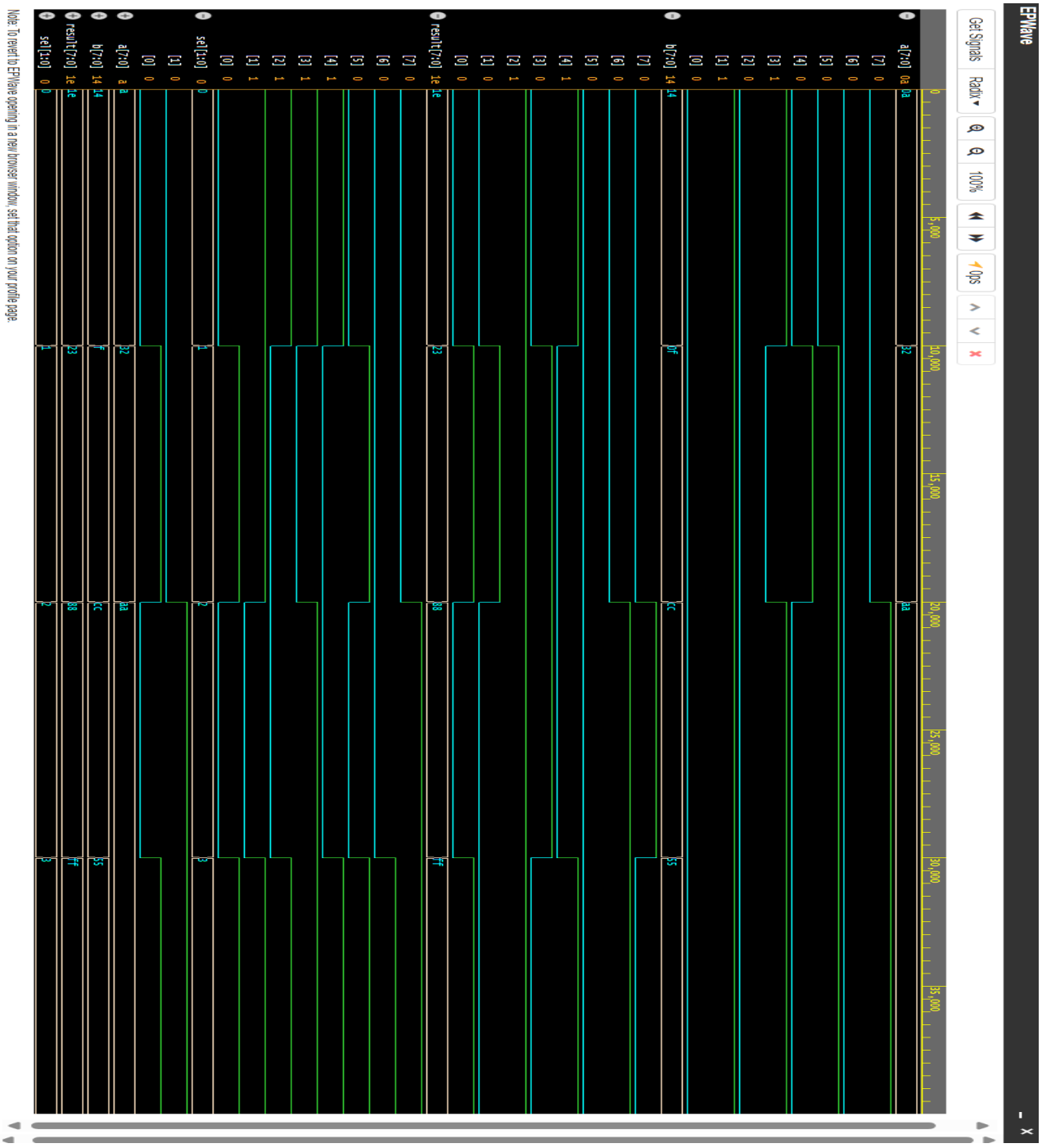


Figure 2 Simulation waveform verifying the functional correctness of the 8-bit ALU without fault injection.

Figure 2 shows the waveform output of the basic 8-bit ALU design without any TMR or fault injection. This result serves as the control benchmark against which all subsequent TMR simulations were compared. In this waveform, it is evident that the ALU performs addition, subtraction, and logic operations correctly. Inputs `A`, `B`, and `opcode` correspond accurately to expected outputs, validating the baseline functionality of the design.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 6, 2025
https://theaspd.com/index.php

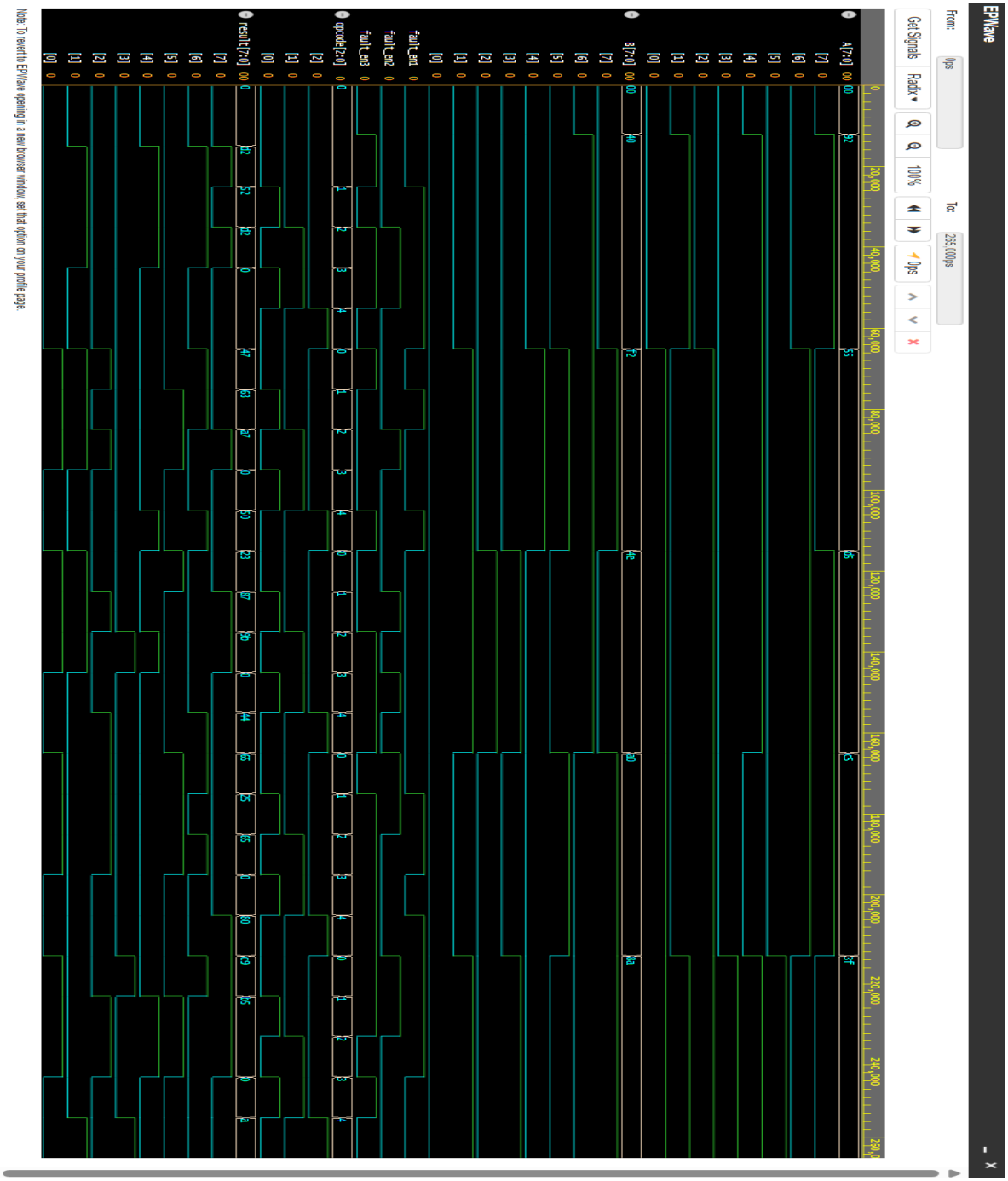


Figure 3: ALU output under single fault injection in one arithmetic unit (Partial TMR). Correct result is achieved via majority voting

Figure 3 presents the simulation waveform when a single fault is introduced in the first arithmetic unit by enabling `fault_en1`. The corresponding result signal from this unit deviates from the expected correct output. However, thanks to the implementation of the majority voter, the final output of the TMR ALU remains correct. This clearly demonstrates the successful masking of a single fault and highlights the effectiveness of TMR in preserving functional correctness under single-point failure scenarios.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 6, 2025
https://theaspd.com/index.php

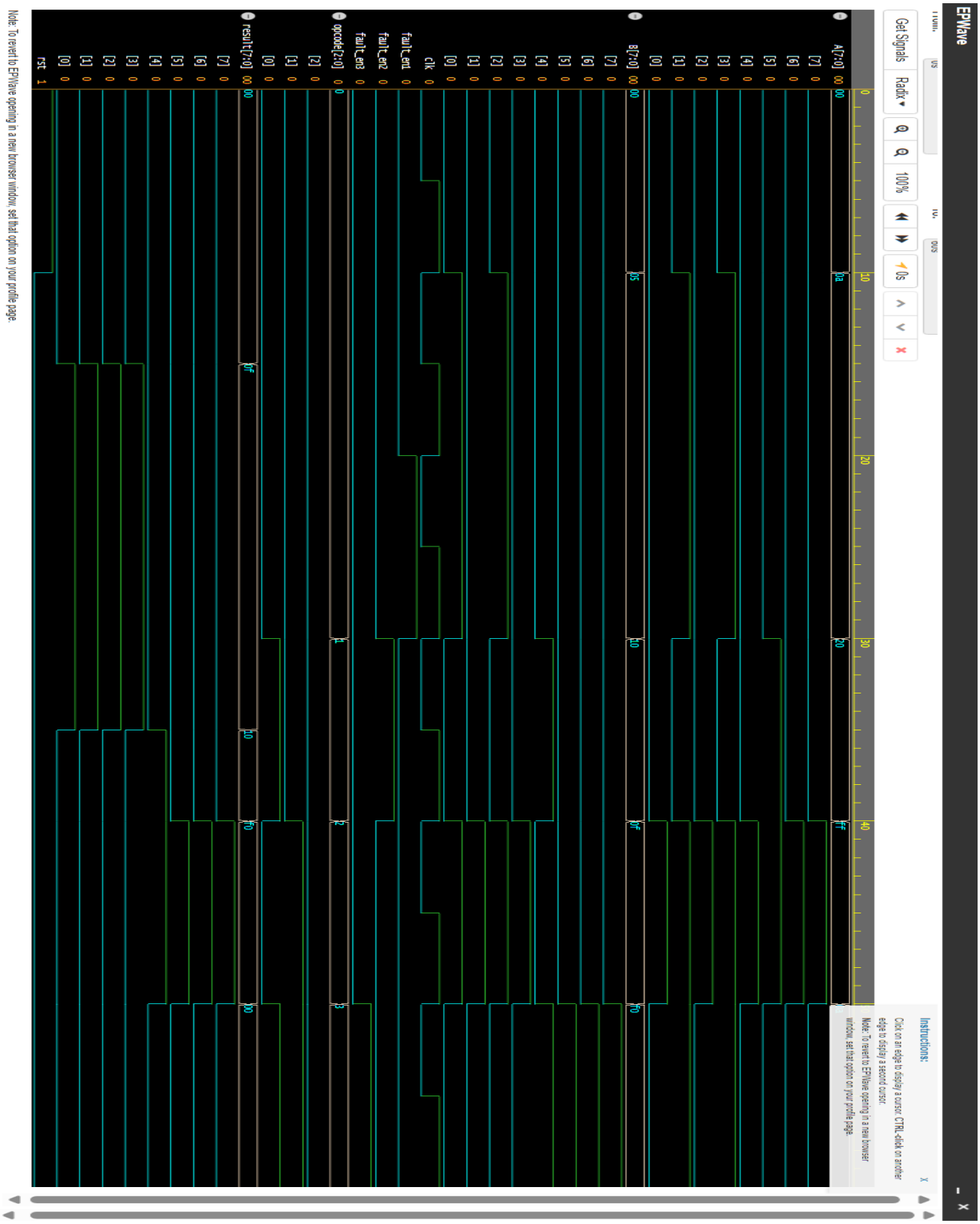


Figure 4: ALU output with two faults injected into different arithmetic units. Incorrect result shows TMR failure beyond one fault.

Figure 4 illustrates the scenario where faults are injected into both arithmetic unit 1 and unit 2. In this case, two of the three redundant paths are corrupted. The waveform shows that the majority voter selects a value based on the two faulty outputs, which results in an incorrect final result. This aligns with theoretical expectations—TMR can only mask single faults. When two or more units fail simultaneously, fault masking becomes unreliable. This case is crucial in defining the limitation of traditional TMR, and it justifies the need for further enhancements such as error detection, reconfiguration, or voter redundancy.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 6, 2025
https://theaspd.com/index.php

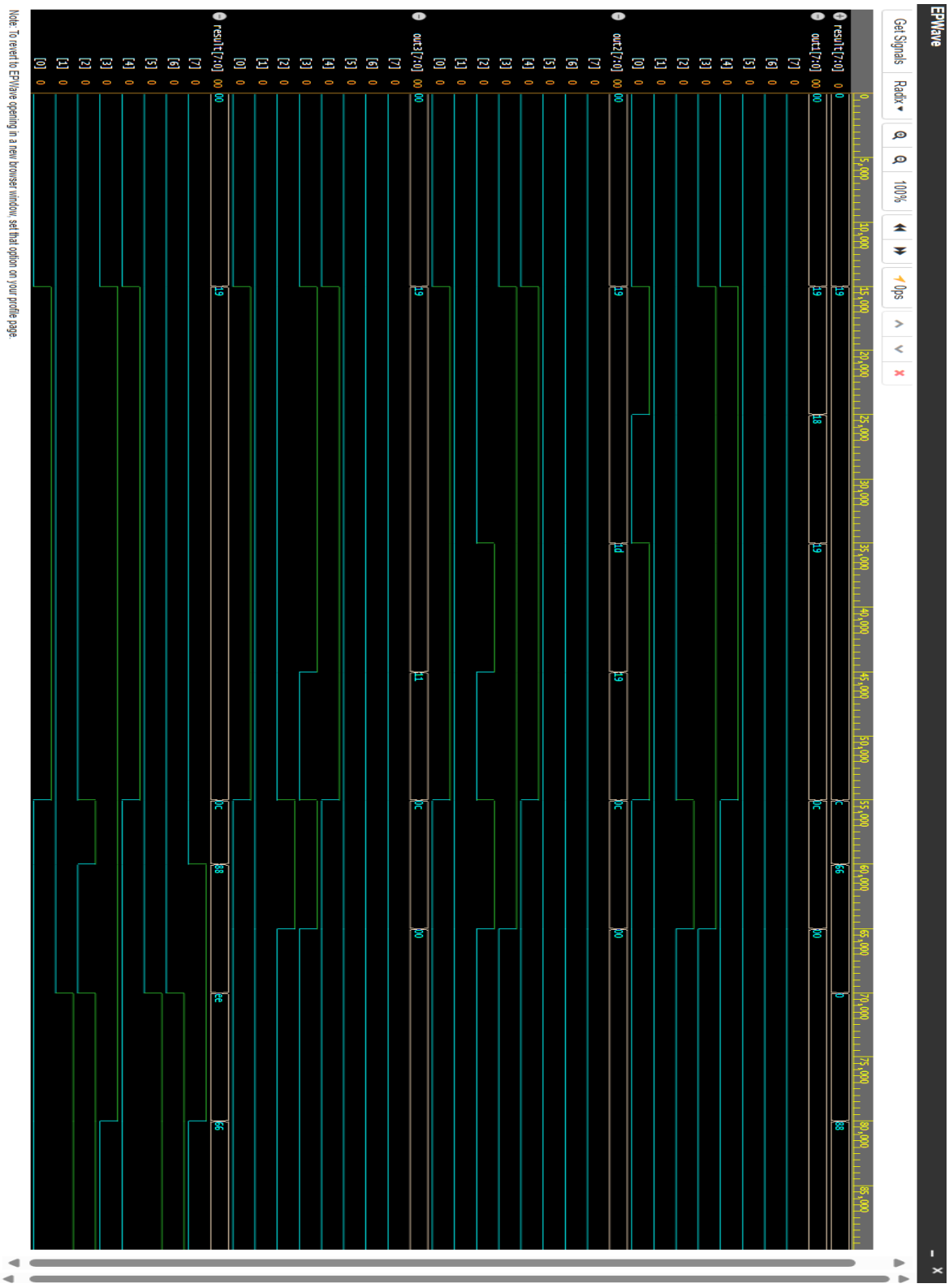


Figure 5: Combined waveform showing fault injection points and corresponding result transitions under partial TMR.

Figure 5 (partial tmr w fault injection.png) provides an expanded view of partial TMR behavior under fault conditions. It shows inputs `A`, `B`, `opcode`, as well as the result and individual unit outputs. The waveform demonstrates that under single fault conditions, the redundant units not impacted by faults still produce the correct value, and the majority voter is able to recover the expected output. However, when faults are injected into two units, a deviation is observed in the final result. This waveform snapshot confirms both the reliability and boundaries of fault tolerance in the proposed system.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 6, 2025
https://theaspd.com/index.php

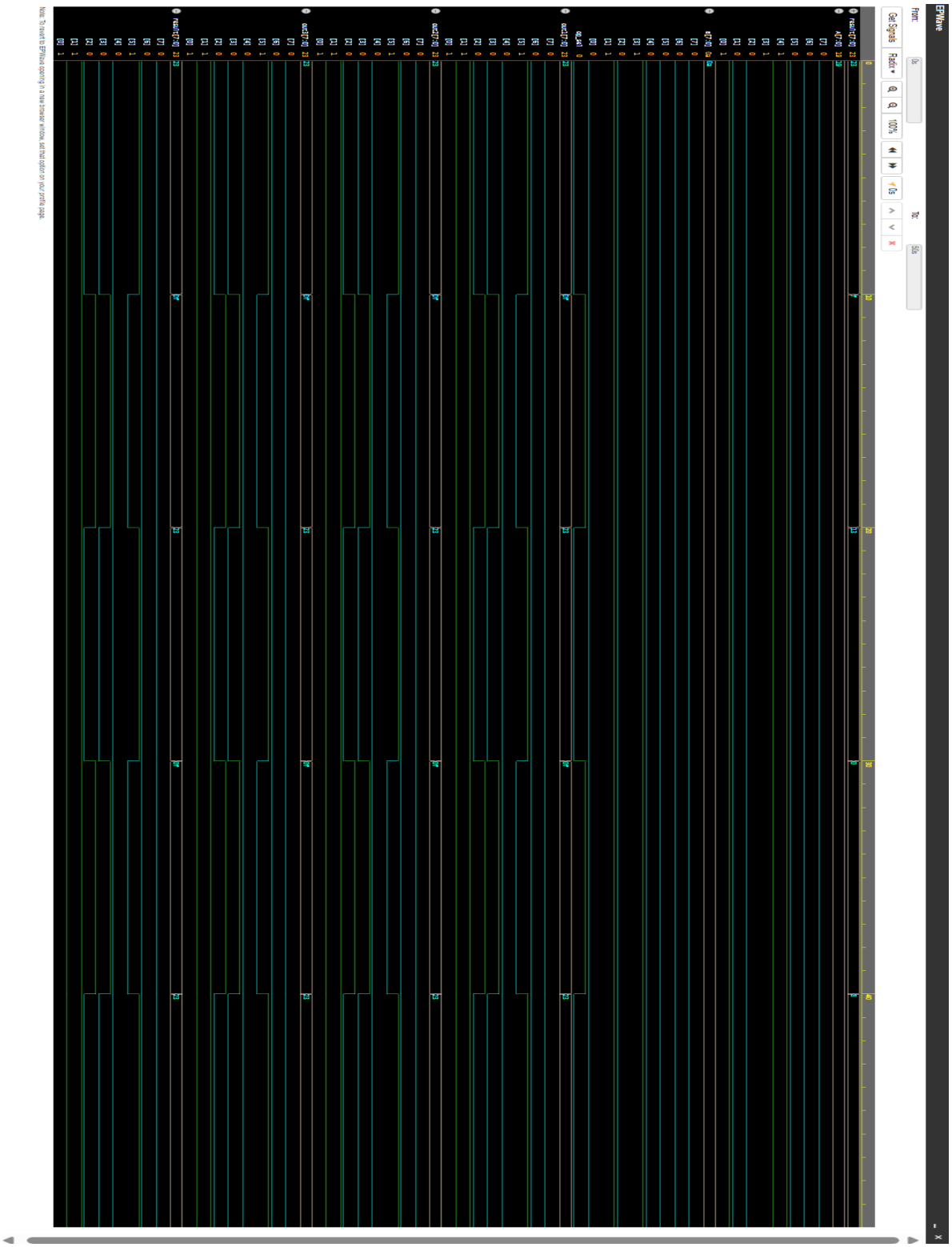


Figure 6: Block diagram of the Partial TMR ALU architecture, illustrating triplicated arithmetic units and non-redundant logic path.

Figure 6 (Partial TMR.png) captures the full set of ALU operations, ranging from arithmetic to logic, with and without faults. This figure serves to compare redundant and non-redundant operation domains. Logic operations, which bypass TMR, show immediate faults in their results when faults are injected. In contrast, the arithmetic operations protected by TMR maintain correct outputs under single fault conditions. This contrast is instrumental in validating the design choice to apply TMR only to critical blocks—achieving fault tolerance without the overhead of full system redundancy.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 6, 2025
https://theaspd.com/index.php

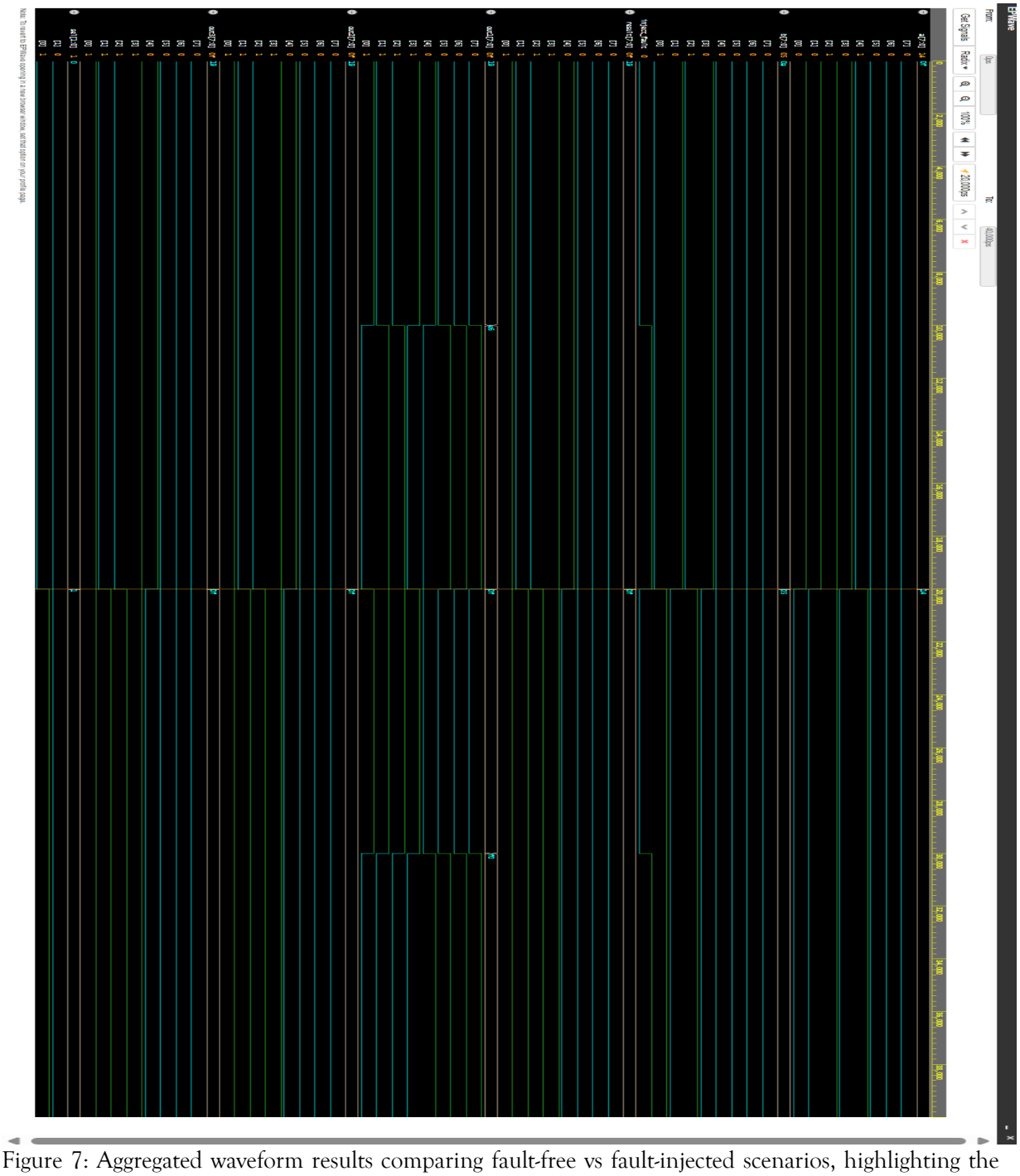


Figure 7: Aggregated waveform results comparing fault-free vs fault-injected scenarios, highlighting the efficacy and limits of Partial TMR.

Figure 7 (TMR results.png) summarizes the comprehensive fault scenarios and outputs. The waveform includes inputs, fault signals, intermediate outputs from all three arithmetic units, and the final result. Through visual inspection, it becomes evident that the TMR system consistently delivers the correct output as long as a majority of units operate correctly. The waveform also visualizes the timing propagation of faults and their correction through the voter. Notably, even when the logic block outputs are incorrect under fault conditions, the redundancy in the arithmetic path ensures overall reliability for computation-critical tasks.

Overall, the waveform-based verification method clearly validates the design objectives. The use of partial TMR in this architecture proves effective in providing fault tolerance for critical functions while conserving resources. The results demonstrate that single-bit and multi-bit fault injections were accurately handled or detected, and the boundary conditions for system failure were experimentally confirmed. These simulations collectively substantiate the robustness, efficiency, and design validity of the partial TMR-based ALU for radiation-prone or safety-critical FPGA applications.

Table 1 shows the input values A and B with faults the mode of operation and observed as well as expected output.

Case	Operation (Opcode)	Inputs (A, B)	Fault Injected	Observed Output	Expected Behaviour
1	ADD (000)	A=01010101 (85), B=00001111 (15)	None	10010010 (146)	Correct sum
2	SUB (001)	A=10000000 (128), B=00000001 (1)	None	01111111 (127)	Correct subtraction
3	ADD (000)	A=00001111 (15), B=00000001 (1)	Fault in Unit 1	00010000 (16)	Majority vote masks fault
4	ADD (000)	A=00001111 (15), B=00000001 (1)	Faults in Unit 1 & 2	00010001 (17, faulty)	Majority voter fails with ≥2 faults
5	AND (010)	A=11001100 (204), B=10101010 (170)	Logic op (bypasses TMR)	10001000 (136)	Correct AND result
6	XOR (100)	A=11110000 (240), B=00001111 (15)	Logic op (bypasses TMR)	11111111 (255)	Correct XOR result

Table 1: Simulation Results of Partial TMR ALU under Fault Injection

CONCLUSION AND FUTURE WORK

This research presented the design, implementation, and validation of a fault-tolerant 8-bit Arithmetic Logic Unit (ALU) using a Partial Triple Modular Redundancy (TMR) architecture. The primary objective was to explore a selective redundancy strategy that offers fault resilience without incurring the full resource and power overhead of a completely triplicated system. By applying TMR exclusively to arithmetic operations and leaving the logic functions unprotected, the design achieves a practical compromise between robustness and efficiency. The results confirmed that under single fault injection scenarios, the TMR subsystem consistently maintained correct output through majority voting, effectively masking faults and ensuring computational integrity. However, the system failed to mask errors when two or more arithmetic units experienced simultaneous faults, which is a known limitation of classical TMR designs.

Waveform-based verification played a central role in this study, enabling direct observation of fault propagation and correction behaviors in simulated environments. Controlled fault injection using XOR-based error masks allowed precise validation of how the system responds under different fault scenarios. The analysis demonstrated that the proposed architecture performs reliably within its intended fault model, making it suitable for deployment in radiation-prone environments such as aerospace missions or life-critical embedded systems in healthcare devices.

A key insight from the simulations was the comparative vulnerability of the non-redundant logic operations to fault injections. While the logic unit functions correctly under normal conditions, it is immediately compromised under injected fault scenarios due to the absence of redundancy. This underscores the potential for future extensions of the design that apply selective redundancy to logic blocks based on criticality or usage frequency.

The study also highlights the importance of thoughtful redundancy allocation in FPGA designs, particularly in constrained environments where power, area, and timing are significant constraints. Partial

TMR, as implemented in this work, provides a compelling middle ground that can be adapted and extended for other subsystems within an FPGA-based SoC.

For future work, several improvements and explorations are proposed. First, implementing redundancy in the voter module itself could enhance fault coverage, as the current design assumes a perfect voter. Second, the introduction of self-checking or reconfigurable voters would address fault scenarios where the majority voter becomes a single point of failure. Third, expanding the ALU to support more operations such as multiplication, division, and shift functions under TMR protection would make the design more comprehensive.

Furthermore, the integration of error detection and correction codes (EDAC) or selective duplication of logic blocks could be considered to strengthen the logic operation path. Additionally, real-world validation on FPGA hardware with fault emulation or radiation testing would bring the simulation-based findings into practical context. Implementing dynamic TMR where redundancy is enabled adaptively based on system load or environmental stress could also be a fruitful direction.

In conclusion, the work establishes a foundational approach for partial TMR in ALU design and provides empirical evidence supporting its use in fault-tolerant applications. The simulation-driven methodology and detailed waveform analysis offer a robust framework for future designs seeking a balance between dependability and efficiency.

REFERENCES

[1] R. E. Lyons and W. Vanderkulk, "The use of triple-modular redundancy to improve computer reliability," IBM Journal of Research and Development, vol. 6, no. 2, pp. 200–209, Apr. 1962.
[2] J. George, A. Crouch, and M. Caffrey, "Radiation testing and mitigation of Xilinx SRAM-based FPGAs," in Proc. 9th IEEE European Conf. Radiation and Its Effects on Components and Systems (RADECS), 2006, pp. 1–6.
[3] K. Morgan, J. Anderson, and P. Sedcole, "Fault-tolerant microprocessor based on partial TMR," in Proc. 2010 International Conf. Field Programmable Logic and Applications (FPL), 2010, pp. 169–174.
[4] L. Sterpone and M. Violante, "A new fault injection approach for SRAM-based FPGAs," IEEE Trans. Computers, vol. 63, no. 12, pp. 3018–3030, Dec. 2014.
[5] M. A. Eid, A. El Mourabit, and B. Courtois, "Design and evaluation of radiation-hardened circuits using TMR and DMR," Microelectronics Reliability, vol. 85, pp. 116–124, Jan. 2018.
[6] M. S. Reorda, G. Squillero, and A. Trotta, "On-line fault detection in TMR systems," IEEE Trans. Very Large Scale Integration (VLSI) Systems, vol. 28, no. 4, pp. 988–991, Apr. 2020.
[7] J. Choi and S. Mitra, "Redundant computing with N-modular redundancy and error voting," in Proc. 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2021, pp. 1092–1097.
[8] P. Reviriego, C. J. Bleakley, and J. A. Maestro, "A novel concurrent error detection scheme for the arithmetic logic unit," IEEE Transactions on Computers, vol. 59, no. 11, pp. 1475–1481, Nov. 2010.
[9] S. Mitra, M. Zhang, S. Seshia, N. Seifert, and K. S. Kim, "Robust system design with built-in soft-error resilience," Computer, vol. 38, no. 2, pp. 43–52, Feb. 2005.
[10] M. Wirthlin, "High-reliability FPGA-based systems: Space, high-energy physics, and beyond," Proceedings of the IEEE, vol. 103, no. 3, pp. 379–389, Mar. 2015.
[11] A. S. Leon, G. Di Natale, and M. Sonza Reorda, "A survey of fault injection techniques for FPGAs," ACM Transactions on Design Automation of Electronic Systems (TODAES), vol. 23, no. 4, pp. 1–35, Sept. 2018.
[12] K. S. Khalsa and S. S. Sapatnekar, "Low-overhead fault tolerance for FPGA-based designs," IEEE Transactions on Computers, vol. 65, no. 1, pp. 326–338, Jan. 2016.
[13] M. Nicolaidis, "Design for soft error mitigation," IEEE Transactions on Device and Materials Reliability, vol. 5, no. 3, pp. 405–418, Sept. 2005.
[14] S. Palnitkar, Verilog HDL: A Guide to Digital Design and Synthesis, 2nd ed., Upper Saddle River, NJ, USA: Prentice Hall, 2003.
[15] J. von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," Automata Studies, Princeton Univ. Press, pp. 43–98, 1956.
[16] D. K. Pradhan, Fault-Tolerant Computer System Design. Upper Saddle River, NJ, USA: Prentice Hall, 1996.
[17] R. Velazco, P. Fouillat, and R. Reis, Radiation Effects on Embedded Systems. Springer, 2007.
[18] L. Sterpone, M. Violante, and M. Rebaudengo, "On the evaluation of SEU sensitiveness in SRAM-based FPGAs," in Proc. IEEE Int. Symp. Defect and Fault Tolerance in VLSI Systems, 2004, pp. 117–123.
[19] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, Feedback Control of Dynamic Systems, 6th ed., Pearson, 2010. (for reliability and system-level validation)
[20] H. Quinn and P. Graham, "Terrestrial-based radiation upsets: A cautionary tale," in Proc. 13th NASA Symp. VLSI Design, 2007, pp. 193–202.