REMPRO: Reconfigurable Modular Processor

REMPRO: Reconfigurable Modular Processor

Invited Paper

Eleonora Vacca
Dipartimento di Automatica e Informatica
Politecnico di Torino
Turin, Italy
eleonora.vacca@polito.it

Elio Strollo
Ingegneria Elettronica Sistemi
s.r.l (IES)
Anzio, Italy
elio.strollo@iessrl.it

Sarah Azimi
Dipartimento di Automatica e Informatica
Politecnico di Torino
Turin, Italy
sarah.azimi@polito.it

ABSTRACT

In space applications, the electronic systems that drive mission-critical operations face unique challenges due to harsh environments characterized by temperature variations and cosmic radiation. The reliability and resilience of electronic systems become paramount in ensuring the success of space missions, where the slightest hardware failure can have profound consequences. To address these challenges, this paper introduces REMPRO, a robust processing system designed specifically for space applications. REMPRO's architecture embodies modularity at both the system and device levels, utilizing multiple chained Field-Programmable Gate Arrays (FPGAs), each equipped with a triple modular redundancy (TMR) processor to enhance reliability and fault tolerance. The system is distinguished by having only one active device at a time and the ability to reconfigure autonomously in case of faults, transferring the current state to the subsequent processor to ensure continuity in mission execution and consequently enhance overall system availability.

CCS CONCEPTS

Hardware~Robustness~Failure recovery, maintenance, and self-repair

KEYWORDS

Space Application, FPGA, TMR, Fault Tolerance, Processing System, Radiation Effects, Radiation Mitigation

ACM Reference format:

Eleonora Vacca, Elio Strollo, Sarah Azimi. 2024. Rempro: Reconfigurable Modular Processor. In *Proceedings of 21st ACM International Conference on Computing Frontiers Workshops and Special Sessions (CF'24 Companion), May 7-9,2024, Ischia, Italy. ACM, New York, NY, USA, 6 pages.* https://doi.org/10.1145/3637543.3652977

1 Introduction

Space missions represent some of humanity's most ambitious and technically demanding tasks. Whether exploring distant planets, conducting scientific research in orbit, or deploying satellite networks for communication and observation, space missions require precise planning, advanced technology, and unwavering reliability. The criticality of space missions arises from their inherent complexity and the extreme conditions of the space environment. The success of these missions depends on the functioning of various systems and components that must ensure their operability under adverse conditions, as even the smallest malfunction can jeopardize the entire mission. Therefore, in the design phase, it is necessary to anticipate and mitigate all possible undesirable effects due to the space environment. Specifically, electronic components are exposed to various forms of radiation, including cosmic rays and solar radiation, which can provoke serious consequences on component behavior. Ionizing radiation can cause single-event effects (SEEs)[1], which can manifest in various forms. Single-event upsets (SEUs) [2] are non-destructive failures. They typically manifest as transient pulses in logic or as bitflips in memory cells or registers. Various types of potentially destructive hard errors can occur, including Single Event Latch-up (SEL) [3], which results in a high operating current surpassing device specifications and requiring a power cycle to restore. Prolonged exposure may lead to cumulative damage and degradation, affecting the overall reliability of electronic systems. Specifically, radiation dose accumulated over time [4] may lead to threshold shifts, increased device leakage (i.e., power consumption), timing changes, and decreased functionality. Mitigating these radiation effects is crucial for ensuring space missions' long-term functionality and success. Robust designs [5], radiation-hardened materials [6], and fault-tolerant architectures [7][8] are employed to safeguard electronic components against the deleterious effects of radiation, thereby enhancing the resilience of systems.

To address these challenges, this work presents REMPRO, a processing system for space applications. REMPRO's objective is to guarantee continuity and reliability in the workload execution, even in the presence of radiation-induced faults. To achieve this objective, REMPRO is designed as a modular system consisting of interconnected identical computational nodes. Only one computational node is active at a time and is equipped with a self-check and self-correction mechanism in the event of radiation-induced silent data corruption (SDC). The self-check and correction mechanism triggers the substitution of the active module if the failure persists, indicating a degradation of logic induced by accumulated dose rather than a SEE. Through a substitution routine, the next module is activated, the current state of the application to be executed is transferred, and the faulty module is shut down – all without the intervention of external modules or user input. This mechanism ensures continuity and reliability in

CF '24 Companion, May 7–9, 2024, Ischia, Italy

ACM ISBN 979-8-4007-0492-5/24/05.
https://doi.org/10.1145/3637543.3652977

workload execution, and through the cyclic shutdown of computational nodes, recovery from accumulated dose is guaranteed.

2 Background on Radiation Hardening Techniques

The radiation hardening methodology involves a multi-faceted approach encompassing technology, architecture, and software. The key aspect of the technology level involves acting on the manufacturing process, using radiation-hardened semiconductors such as silicon-on-insulator (SOI) [9], where a layer of insulating material is added beneath the silicon, reducing the susceptibility of transistors to radiation-induced charge collection. Another possibility is acting at the layout level for the integrated circuit (IC) design [10], allocating guard rings for SEL protection [11], and guard-gate circuitry for SET protection [12][13]. Furthermore, rad-hard non-volatile memory technologies, such as Ferroelectric RAM (FRAM) [14] and Magnetoresistive RAM (MRAM)[15], offer advantages over traditional memory regarding data retention and radiation tolerance. However, radiation hardening by process's main drawbacks are performance degradation and silicon area increment. A notable current trend is the increasing adoption of Commercial Off-The-Shelf (COTS) products in space missions. The significant cost advantage drives this shift, as COTS products are approximately an order of magnitude less expensive than their radiation-hardened counterparts. To address radiation issues while using the COTS component, redundancy is employed. The most common approach is triple modular redundancy (TMR) [16] which consists of allocating three modules executing in parallel the same task, whose outputs are then provided to a majority voter. The TMR granularity [17] strictly depends on the mission requirements and technology. When dealing with COTS application-specific IC (ASIC), it can be applied at the system level, allocating several chips in parallel, while more flexibility is achieved when using programmable logic as field programmable gate arrays (FPGA), where redundancy can be applied at the architectural level or the gate level [18], since circuit implementation is only constrained by logic resource available on the FPGA device. Moreover, adopting an FPGA device allows the combining of different hardening techniques, such as acting on the physical layout through place and route manipulation[5]. However, strong reliability issues may arise by adopting FPGA for space applications. Indeed, the main drawback is the SEU sensitivity of their configuration memory since a radiation-induced memory bitflip can alter the implemented circuit structure. This aspect is a concern especially when adopting SRAM-based FPGA. On the other hand, despite lower performance and higher time to configure, FLASH-based FPGAs are more robust. FLASH technology requires higher energy deposition to provoke memory cell upset concerning SRAM one. As a consequence, for space missions in the GEO environment and beyond, usually FLASH-based FPGA are adopted [19].

At the software level, methodologies are based on information redundancy through error detection and correction, such as parity, Hamming code, periodic memory scrubbing, or temporal redundancy, executing redundant operations in sequence on the same hardware [20]. Hardware and software redundancy may be coupled to enhance further the robustness of the system. Indeed, typically, when dealing with system-level TMR, software methodologies involving task monitoring and recovery procedures are adopted [21][22].

In this perspective, the REMPRO system merges state-of-the-art methodologies to achieve high levels of robustness. Specifically, it combines low-level techniques targeting the processor architecture with high-level techniques involving software monitoring, memory content scrubbing, and checkpoint recovery routines.

3 REMPRO

The requirements to ensure that a system is robust and, therefore, suitable for mission-critical applications such as space missions can be achieved by addressing different levels of abstraction. This involves adopting broad-spectrum methodologies, including architectural design, system design, protocol, and software. A reasonable level of robustness encompasses the following characteristics:

1. The majority of failures are not inherently destructive and can be recovered, for example, through power cycling or complex recovery procedures.
2. At least key components must be implemented robustly to facilitate recovery.
3. Diagnostic and monitoring procedures must be in place to detect any failures or anomalies in operation.
4. Protection and redundancy mechanisms must be implemented for critical systems, such as power systems, communication systems, and control systems.
5. Backup and restoration procedures must be established for recovering functionality in the event of failures.

The REMPRO system is the outcome of a research collaboration between academia and industry aimed at addressing the aforementioned points. The development process involved thoroughly exploring various trade-offs among state-of-the-art methodologies aiming at developing a robust processing system for space applications. Various alternatives, each characterized by the application of TMR with different granularity, have been evaluated through an iterative process. Furthermore, design choices have been made related to the monitoring system either embedded in the processor Datapath or as an external watchdog system. This iterative exploration ultimately led to the definition of the final version of the system.

The REMPRO system is developed on FLASH-based FPGAs to achieve design flexibility without compromising robustness.

The REMPRO system is conceived as a modular structure consisting of N interconnected FLASH-based FPGAs. A 16-bit custom processor, Hepro, is implemented in the TMR configuration within each FPGA. This configuration involves replicating not only the computational components but also the memory modules, as illustrated in Figure 1. The processing system is equipped with a global voter for the outputs generated by individual processors. The value produced by this voter serves as the system's overall output. Additionally, for each independent Datapath, a voter V_j is instantiated between processor P_j and its memory bank. Each voter receives, as input, the output read from its corresponding RAM, along with those read from the other two RAMs within the system. The three voters operate in parallel, each providing an independently voted output for its respective processor.

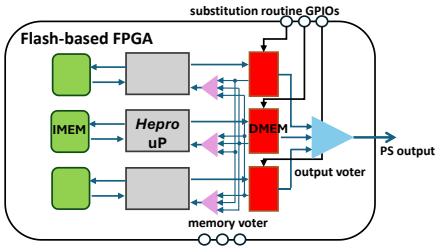


Figure 1. Internal TMR structure for a single FPGA device. Green blocks are the instruction memory, grey blocks are the *Hepro* processing system, and red blocks relate to data memory. The system is equipped with two voting stages.

Each voter is equipped with a data comparison mechanism. In addition to majority voting to provide a correct output, this mechanism activates a *warning* signal in case a difference is detected among the input data. Each processor has its warning signal, W_j. Figure 2 presents the overall modular system architecture, with the daisy chain connection among the FGPAs.

Each FPGA will have a dedicated set of pins for transmitting the contents of the RAM, used during the initialization of a new board. Specifically, the data parallelism is 16 bits. Therefore, 48 pins of the FPGA are allocated for transferring memory content. In the proposed scheme, each RAM_j belonging to $FPGA_i$ transfers data to the corresponding RAM_J in $FPGA_{i+1}$ without any voting mechanism. Upon completion of data transmission, the substitution routine involves reading the memory content, which then passes through the output voters of the RAM and rewriting the data just read and voted on.

Each *Hepro* structure is provided with a special register *Warning* used to store the W_j signals coming from the voters. This register is periodically inspected so that if one of its bits is set to 1, it's likely that one of the system voters detected an error. The choice of using a dedicated register to be inspected instead of the use of the warning signals as processor interrupt is to avoid the flush of current execution, whose correctness is still guaranteed by the voting mechanism.

The main feature of REMPRO is the autonomous handling of turn-on and turn-off of FPGA devices when a persistent fault is detected. Specifically, the faulty FPGA is in charge of power on the following one, while the just turned-on FPGA has to power off the previous FPGA. This behavior is achieved by a power relay controlled by signals coming from the FPGA I/O pins. The power control circuit is a bistable circuit, shown in Figure 3.

For each $FPGA_i$ in the system, its RESET terminal is driven by the $FPGA_{i+1}$ in the chain, while the $FPGA_{i-1}$ drives the SET terminal.

An external serial ferroelectric memory is used to store the program code to enhance system reliability further. This memory undergoes continuous reading to scrub the content of the IMEM for each *Hepro*, implemented in the FPGA through BRAMs and therefore susceptible to SEUs.

The overall design choices are compliant with the requirements defining a robust system. Adopting Flash-based FPGA guarantees no structural faults on Datapath induced by SEU in memory while maintaining design flexibility. Moreover, the adoption of ferroelectric memory, combined with the periodic memory scrubbing methodology, ensures consistent execution of the workload application among the TMR processors. Finally, the power cycling procedure achieved with the programmed relay

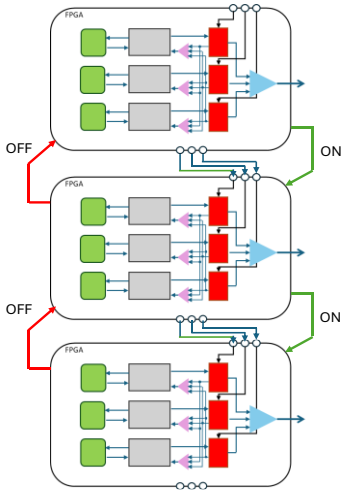


Figure 2. Schematic view of REMPRO system. FPGA devices are connected in a daisy chain, with just one active device at a time. Each FPGA controls its adjacent devices' turn on/turn off mechanism.

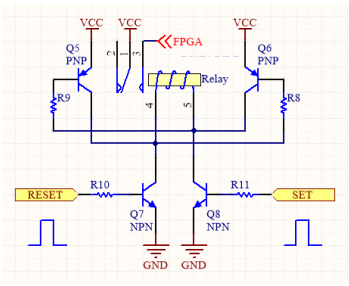


Figure 3. Schematic view of the power-on and power-off circuit of each FPGA device. The $FPGA_{i-1}$ drives the SET terminal to power on the $FPGA_i$, while the $FPGA_{i+1}$ drives the RESET terminal to power off the device.

combined with the substitution routine guarantees continuity in the workload execution and system availability.

4 Substitution Routine

Despite the high level of reliability achieved by adopting TMR at the design level, including two voting stages, radiation-induced effects still may result in system failure. Indeed, TMR mitigates transient faults resulting in wrong computation but in the end, cannot avoid any structural damage induced by long-term exposure to radiation sources. Indeed, the high-energy particles hitting the device over time, deposit energy in the material, typically leading to the creation of electrons-hole pairs. This translates into variations of the technology parameters such as leakage current, threshold voltage, and performance degradation. For this reason, the REMPRO system is thought to be resilient concerning this kind of radiation effect by adopting the replacement mechanism as soon as a permanent fault is detected in the current active FPGA.

As mentioned in Section 3, each *Hepro* is equipped with a special register *Warning* that records the occurrence of a mismatch detected by the voting system. Specifically, each voter output signal is assigned to a specific bit of this register. Inside the main program code, checkpoint instructions are inserted to read the Warning register's content. If one of the register's bits is set to 1,

the current execution is halted, and the program jumps to the substitution routine.

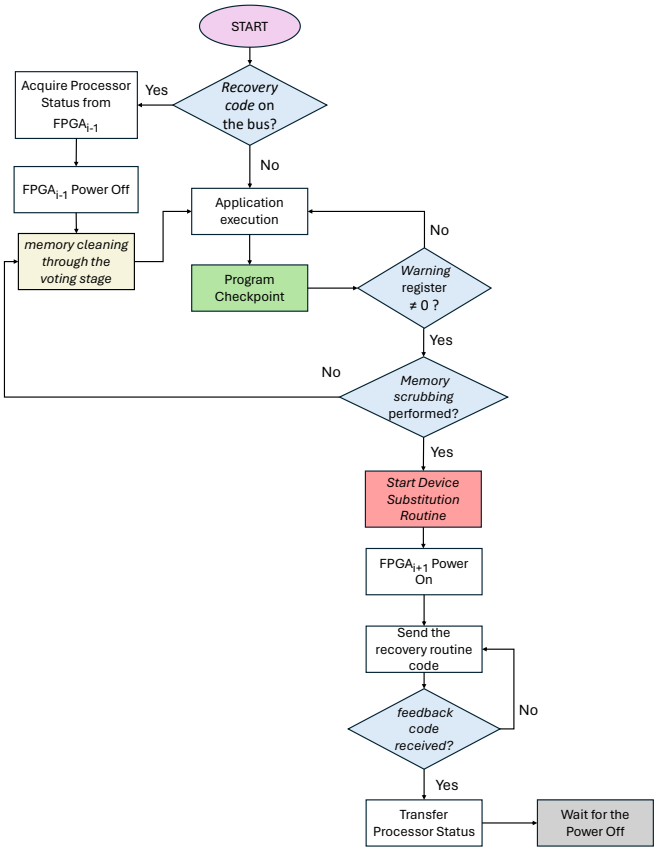


Figure 4. The substitution routine flow.

The substitution routine consists of three main phases. The first phase evaluates the severity of the detected fault. This is done by forcing a *cleaning* operation of the memory content, which is read and re-written through the voting stage located between each *Hepro* and its memory bank. In the case of simple SDC affecting either the logic due to a sampled transient or an SEU in memory, this procedure will remove any radiation-induced mismatch among the processors. Then, the program execution restarts from the latest instructions, and the Warning register is read just after. If its bits are still set to 1, means that a permanent fault is affecting the device, which needs to be substituted. At this point, the second phase starts. The current active FPGA$_i$ forces the turn-on of the following one in the chain, FPGA$_{i+1}$. Then, it sends a special code to the bus through its substitution routine GPIOs. This code is kept on the bus until the FPGA$_{i+1}$ completes its boot with the TMR *Hepro* active. As soon as the second triplet awakes, reads the bus, recognizes the special code transmitted, and understands it to be a recovery device. Hence, its program counter jumps to the context transfer procedure. At this point, the new FPGA$_{i+1}$ sends an acknowledgment code and waits for the FPGA$_i$ to transfer the overall processor status and data memory content. When the FPGA$_i$ completes the transmission, the third phase starts with the FPGA$_{i+1}$ turning off the FPGA$_i$ and executing a cleaning operation before resuming the program execution. Indeed, the memory transfer routine is executed in parallel for all the memory banks, not adopting any voting mechanism. Hence, the memories are read and re-written to ensure consistent data, flowing through the voting stage. Since the program counter is saved in the stack and also transferred during the substitution routine, the new TMR processor starts the program code not from the beginning but from the last executed instruction before the replacement.

By turning off the faulty devices and adopting a rotation mechanism in the spare FPGA activation, we ensure a sufficient recovery time from TID effects, as during the off period, the electron-hole pairs recombine, and the radiation-induced variation is absorbed.

5 Experimental Analysis

The proposed TMR Hepro processor has been prototyped on ProASIC3 A3P250 Flash FPGA. Implementation details are reported in Table 1. The design validation involved two main aspects (i) the fault tolerance capabilities of the developed system and (ii) its autonomous reconfiguration. The robustness of the system has been evaluated through the emulation of a Single Event

Table 1. Implementation details of the Hepro processor

	LUTs [%]	FFs [%]	BRAMs	Frequency (MHz)
Hepro	61	6.07	2	16

Transient (SET) affecting the logic. In contrast to SRAM-based implementation, where SEUs in the CRAM are the main concern [2], the implemented circuit is likely resilient to SEU-induced structural faults in the Datapath, given that FLASH technology, used to implement the CRAM, is more robust. Still, the combinational paths are susceptible to radiation-induced Single Event Transient (SET)[23]. High energy particles striking the silicon material may lead to voltage glitches at the output of transistors.

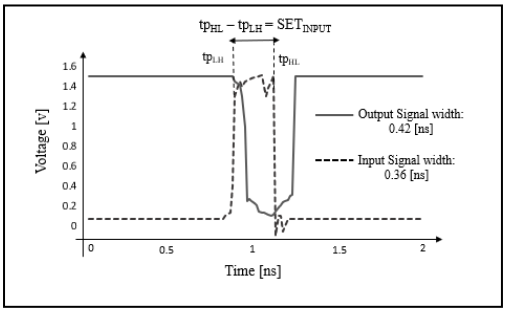


Figure 5. The SET pulse propagates through an Inverter Gate, broadening due to the PIPB effect.

The propagating SET may be sampled by memory elements and cause, in the worst-case scenario, system failure if a control signal is affected, otherwise, SDC. Depending on the combinational path characteristics, such as length and logic gates in the path, the SET width can be broadened or filtered. As a consequence, not all the SETs are transformed into circuit errors. However, the wider the pulse the higher the probability of being sampled by a memory element since its duration may match the set-up and hold time of the FFs, endpoints of the affected combinational path. SET injection campaign was carried out by instrumenting the HDL simulation environment, targeting the post-layout netlist. The SET analysis is performed by defining the injection node, the injection instant, and the transient pulse width.

Targeting different space applications and environment scenarios, different pulse widths have been evaluated, ranging from 450ps to

1ns, hence covering applications from Low Earth Orbit to Deep Space, where the particle, namely Galactic Cosmic Rays have higher energy, inducing higher charge deposition.

The target node can be either a logic resource's inputs or outputs or a junction between connections. However, in post-layout dynamic analysis, FF's description includes in their port description, not only the nominal cell delay but also the delay related to the combinational path driving its ports. Considering this, FFs were selected as target nodes of the injections. Moreover, the single Hepro design, as can be seen from Table I, uses a small amount of FFs (371), which allows us to perform an exhaustive SET injection campaign.

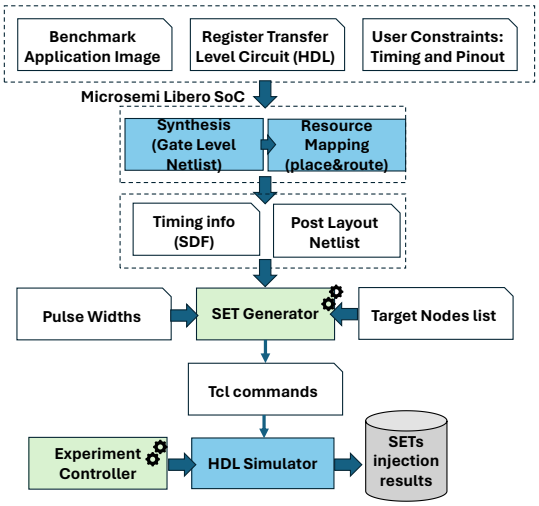


Figure 6. SET injection flow.

The SET is implemented by forcing the FF's driving signal through Tcl commands to logic value high (or low), hence emulating a glitch. The injection time instant is random, and for each FF, the same glitch is injected 100 times targeting different time instants during the application workload execution. The overall fault injection flow is reported in Figure 6. As benchmark applications, three different programs have been implemented aiming at stimulating different areas of the Datapath This allows observing how the same transients, applied to the same nodes, lead to consequences varying in severity depending on the application workload. Specifically, an arithmetic test, matrix multiplication application, and quicksort application.

The arithmetic test executes a series of arithmetic operations, jumps, and branches representing the execution of a generic program, stimulating a condition of balanced workload for the processor. The matrix multiplication benchmark mainly stimulates the ALU and the memory accesses to load and store the results. Finally, the quicksort consists of loops, recursive calls, and read/write operations allowing stress the conditional jumps module, the memory, and the stack. The SET analysis results are provided in Figure 7.

The results are presented as the percentage of injected SETs provoking either a computational mismatch compared to the golden reference or a system timeout marked as an error. Results show that the node's vulnerability is strongly dependent on the application workload. Indeed, recalling that for each SET injected, targeting a node (FF), the three different applications are executed. Considering this, the Quicksort application resulted in a lower failure rate than the other two applications, regardless of the same

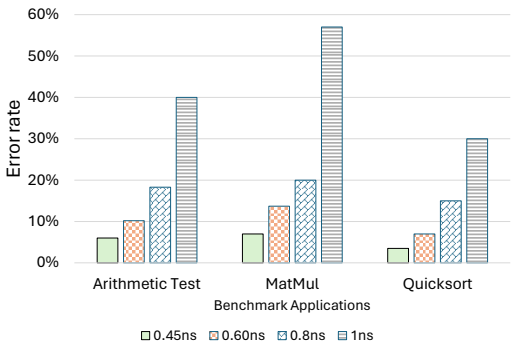
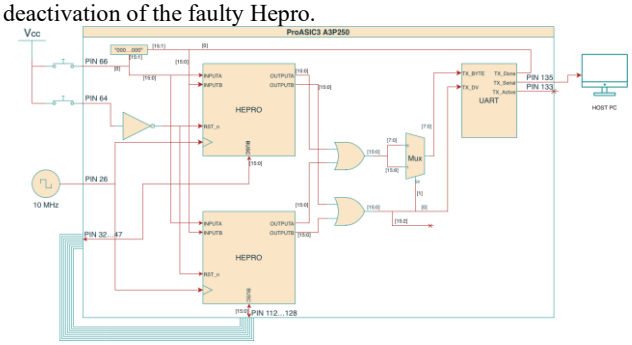


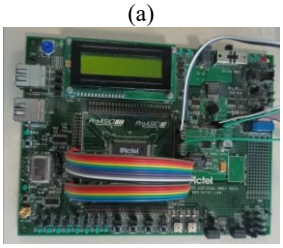
Figure 7. SET analysis results over an exhaustive campaign of 37,100 injections for different SET pulse widths.

injection nodes. Moreover, as the pulse width increases, also the error rate increases since the longer the pulse, the higher the probability of catching the FF sampling window. Overall, the system presents good resilience related to transient with pulses ranging from 450ps to 800ps providing an error rate lower than 40% for the different applications.

Once the robustness of the system has been evaluated, its embedded auto-reconfiguration capabilities have been considered. The full REMPRO system requires chained FPGA devices to exploit in reconfiguration. However, at the time of the project prototyping, just one board was available. To overcome we designed a DUT consisting of two independent Hepro processors to be implemented in the same FPGA. The circuit schematic is proposed in Figure 8a. The two processors communicate through an external bus, which is used during the substitution routine to transfer the current system status to the spare FPGA. The bus has been implemented through jumpers connected to each processor Recovery Port mapped to FPGA GPIO, as shown in Figure 8b. In the final system, each TMR processor must be capable of turning on/off the next and previous processors, respectively. To simulate this behavior, the architecture of each processor has been equipped with a special register *PWC* to control the reset of the other processor, thereby simulating the deactivation of the faulty Hepro.



(a)



(b)

Figure 8. DUT schematic for the autonomous recovery routine testing.

At the system turn on, just one Hepro is active, executing a simple benchmark program. Some program checkpoints are implemented, where the *Warning* register is read. To force the substitution routine, as per the SET analysis, a bit of this register is forced to 1 in runtime, provoking the program counter to jump to the substitution routine code. At this moment, since no TMR is implemented, the routine does not try to recover from the detected error but starts the processor context transfer using the external bus, executing the flow described in Section 4. As soon as the second Hepro finishes acquiring the data coming from the faulty Hepro, it updates its program counter with the content of the one just transferred by the faulty Hepro and the program flow continues from the last instruction before the detected error. The overall correctness of the routine is evaluated by forcing the processors to write to the UART interface, to check whether the computations and instructions fetching flow is correct before and after the substitution routine.

This test has enabled the verification of two crucial components of the REMPRO system. Both the transmission through the external bus and the devised recovery procedure exhibited the expected behavior. The pair of processors successfully halted execution, replaced the active processor, transferred the program state to the next processor, and turned off the first one. The second processor then resumed execution from where the first had been interrupted and managed to run the program to completion accurately. The two processors remained synchronized, and no data losses occurred. This demonstrates the robustness of the initial concept and allows for the continued development of the system.

6 Conclusions

In this work, we introduced the REMPRO system as a fault-tolerant processing system for space applications. We went through the design phase, illustrating all the choices that led to the final structure of the system. Specifically, we deeply analyzed the TMR architecture inside each FPGA device, the inter-board communication, and the system auto-reconfiguration mechanism through the proposed substitution routine. We then evaluated the robustness of the main component, the Hepro processor again SET, and discovered high fault tolerance capabilities. The achieved results show both the feasibility of the proposed system and the injection campaign its robustness, placing the REMPRO system as a prominent solution for space applications.

REFERENCES

[1] D. Kobayashi, "Scaling Trends of Digital Single-Event Effects: A Survey of SEU and SET Parameters and Comparison With Transistor Performance," in IEEE Transactions on Nuclear Science, vol. 68, no. 2, pp. 124-148, Feb. 2021, doi: 10.1109/TNS.2020.3044659.

[2] E. Vacca et al., "Analyzing the SEU-induced Error Propagation in Systolic Array on SRAM-based FPGA", in IEEE European Conference on Radiation and its Effects on Components and Systems (RADECS), Toulouse, France, 2023.

[3] R. Secondo et al., "Analysis of SEL on Commercial SRAM Memories and Mixed Field Characterization of a Latchup Detection Circuit for LEO Space Applications," in IEEE Transactions on Nuclear Science, vol. 64, no. 8, pp. 2107-2114, Aug. 2017, doi: 10.1109/TNS.2017.2691403.

[4] A. C. V. Bôas et al., "Radiation Hardness of GaN HEMTs to TID Effects: COTS for harsh environments," 2019 19th European Conference on Radiation and Its Effects on Components and Systems (RADECS), Montpellier, France, 2019, pp. 1-4, doi: 10.1109/RADECS47380.2019.9745697.

[5] E. Vacca et al., "Layout-oriented radiation effects mitigation in RISC-V soft processor". 2022, In Proceedings of the 19th ACM International Conference on Computing Frontiers (CF '22). Association for Computing Machinery, New York, NY, USA, 215–220. https://doi.org/10.1145/3528416.3530984.

[6] L. Blanquart et al., "Study of proton radiation effects on analog IC designed for high energy physics in a BiCMOS-JFET radhard SOI technology," in IEEE Transactions on Nuclear Science, vol. 41, no. 6, pp. 2525-2529, Dec. 1994, doi: 10.1109/23.340611.

[7] A. Aponte-Moreno et al. "A Low-Overhead Radiation Hardening Approach using Approximate Computing and Selective Fault Tolerance Techniques at the Software Level," 2019 19th European Conference on Radiation and Its Effects on Components and Systems (RADECS), Montpellier, France, 2019, pp. 1-4, doi: 10.1109/RADECS47380.2019.9745663.

[8] E. Vacca et al., "Failure rate analysis of radiation tolerant design techniques on SRAM-based FPGAs", Microelectronics Reliability, Volume 138,2022,114778, ISSN 0026-2714, doi:10.1016/j.microrel.2022.114778.

[9] Y. Sun et al., "SOI FinFET Design Optimization for Radiation Hardening and Performance Enhancement," in IEEE Transactions on Device and Materials Reliability, vol. 23, no. 3, pp. 386-394, Sept. 2023, doi: 10.1109/TDMR.2023.3287839

[10] S. Azimi et al., "A Radiation-Hardened CMOS Full-Adder Based on Layout Selective Transistor Duplication," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 29, no. 8, pp. 1596-1600, Aug. 2021, doi: 10.1109/TVLSI.2021.3086897.

[11] N. Rezzak and J. -J. Wang, "Single Event Latch-Up Hardening Using TCAD Simulations in 130 nm and 65 nm Embedded SRAM in Flash-Based FPGAs," in IEEE Transactions on Nuclear Science, vol. 62, no. 4, pp. 1599-1608, Aug. 2015, doi: 10.1109/TNS.2015.2450210.

[12] M. Andjelkovic et al., "SET and SEU Hardened Clock Gating Cell," 2023 38th Conference on Design of Circuits and Integrated Systems (DCIS), Málaga, Spain, 2023, pp. 1-6, doi: 10.1109/DCIS58620.2023.10335985.

[13] E. Vacca et al., "A Comprehensive Analysis of Transient Errors on Systolic Arrays," 2023 26th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS), Tallinn, Estonia, 2023, pp. 175-180, doi: 10.1109/DDECS57882.2023.10139763.

[14] F. Huang et al., "HfO2-Based Highly Stable Radiation-Immune Ferroelectric Memory," in IEEE Electron Device Letters, vol. 38, no. 3, pp. 330-333, March 2017, doi: 10.1109/LED.2017.2653848

[15] B. Wang et al., Soft Error Sensitivity of Magnetic Random Access Memory and Its Radiation Hardening Design," 2021 18th International SoC Design Conference (ISOCC), Jeju Island, Korea, Republic of, 2021, pp. 199-200, doi: 10.1109/ISOCC53507.2021.9613876.

[16] K. Szurman et. al., "Coarse-Grained TMR Soft-Core Processor Fault Tolerance Methods and State Synchronization for Run-Time Fault Recovery," 2019 IEEE Latin American Test Symposium (LATS), Santiago, Chile, 2019, pp. 1-4, doi: 10.1109/LATW.2019.8704639.

[17] U. Kretzschmar, A. Astarloa, J. Lázaro, M. Garay and J. Del Ser, "Robustness of different TMR granularities in shared wishbone architectures on SRAM FPGA," 2012 International Conference on Reconfigurable Computing and FPGAs, Cancun, Mexico, 2012, pp. 1-6, doi: 10.1109/ReConFig.2012.6416785.

[18] L. A. García-Astudillo et al., "Evaluating Reduced Resolution Redundancy for Radiation Hardening in FPGA Designs," in IEEE Transactions on Nuclear Science, vol. 70, no. 8, pp. 2060-2067, Aug. 2023, doi: 10.1109/TNS.2023.3268825.

[19] S. Azimi et al.,"Accurate analysis of SET effects on Flash-based FPGA System-on-a-Chip for satellite applications," 2016 16th European Conference on Radiation and Its Effects on Components and Systems (RADECS), Bremen, Germany, 2016, pp. 1-4, doi: 10.1109/RADECS.2016.8093203.

[20] C. De Sio et al., "SEU Evaluation of Hardened-by-Replication Software in RISC-V Soft Processor," 2021 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), Athens, Greece, 2021, pp. 1-6, doi: 10.1109/DFT52944.2021.9568342.

[21] Á. B. de Oliveira et al., "Analyzing the Influence of using Reconfiguration Memory Scrubber and Hardware Redundancy in a Radiation Hardened FPGA under Heavy Ions," 2018 18th European Conference on Radiation and Its Effects on Components and Systems (RADECS), Goteborg, Sweden, 2018, pp. 1-4, doi: 10.1109/RADECS45761.2018.9328683.

[22] L. Bozzoli et al., "EuFRATE: European FPGA Radiation-hardened Architecture for Telecommunications," 2023 Design, Automation & Test in Europe Conference & Exhibition (DATE), Antwerp, Belgium, 2023, pp. 1-6, doi: 10.23919/DATE56975.2023.10137035.

[23] L. Sterpone et al., "Effective Characterization of Radiation-induced SET on Flash-based FPGAs," 2017 17th European Conference on Radiation and Its Effects on Components and Systems (RADECS), Geneva, Switzerland, 2017, pp. 1-4, doi: 10.1109/RADECS.2017.8696255.