Trikarenos

Design and Experimental Characterization of a Fault-Tolerant 28-nm RISC-V-Based SoC

Rogenmoser, Michael; Wiese, Philip; Forlin, Bruno Endres; Gurkaynak, Frank K.; Rech, Paolo; Menicucci, Alessandra; Ottavi, Marco; Benini, Luca

DOI
10.1109/TNS.2025.3564739

Publication date
2025

Document Version
Final published version

Published in
IEEE Transactions on Nuclear Science

Trikarenos: Design and Experimental Characterization of a Fault-Tolerant 28-nm RISC-V-Based SoC

Michael Rogenmoser, *Graduate Student Member, IEEE*, Philip Wiese, *Graduate Student Member, IEEE*, Bruno Endres Forlin, Frank K. Gürkaynak, Paolo Rech, *Senior Member, IEEE*, Alessandra Menicucci, Marco Ottavi, *Senior Member, IEEE*, and Luca Benini, *Fellow, IEEE*

***Abstract*— RISC-V-based fault-tolerant system-on-chip (SoC) designs are critical for the new generation of automotive and space SoC architectures. However, reliability assessment requires characterization under controlled radiation doses to accurately quantify the fault tolerance of the fabricated designs. This work analyzes the Trikarenos design, an SoC implemented in TSMC 28 nm, for single event upset (SEU) vulnerability under atmospheric neutron and 200-MeV proton radiation, comparing these results to simulation-based fault injection. All faults in error correction codes (ECCs) protected memory are corrected by a scrubber, showing an estimated cross section per bit of up to 1.09×10^{-14} cm^{2}bit^{-1}. Furthermore, the triple-core lockstep (TCLS) mechanism implemented in Trikarenos is validated and is shown to correct errors affecting a cross section up to 3.23×10^{-11} cm^2, with the remaining uncorrectable vulnerability below 5.36×10^{-12} cm^2. When augmenting the experimental analysis of fabricated chips with gate-level fault injection in simulation, 99.10% of injections into the SoC produced correct results, while 100% of injections in the TCLS-protected cores were handled correctly. With 12.28% of all injected faults leading to a TCLS recovery, this indicates an approximate effective flip-flop (FF) cross section of up to 1.28×10^{-14} cm^2/FF.**

***Index Terms*— Fault injection, fault tolerance, integrated circuit reliability, neutron radiation effects, proton radiation effects, radiation effects in ICs, radiation hardening by design (RHBD), reliability analysis, RISC-V, single event upset (SEU).**

I. Introduction

In THE automotive and space domains, the reliability of semiconductor chips and systems-on-chip (SoCs) is critical. The space sector faces challenges due to high radiation levels causing single event effects (SEEs), which can disrupt the normal functioning of electronic components. Reliability is also a major intrinsic concern for the automotive industry since atmospheric neutrons can cause single event upsets (SEUs), leading to crashes or unexpected behaviors. Various strategies are employed to mitigate these risks. At the technology level, radiation-hardened technologies and specialized cells can enhance resilience [1]. Alternatively, architectural solutions such as dual modular redundancy (DMR), triple modular redundancy (TMR) [2], or error correction codes (ECCs) [3] can be implemented to ensure a safe operation in critical applications.

The RISC-V architecture is rapidly gaining traction in safety-critical applications, thanks to the openness of the instruction set architecture (ISA) and the availability of open processor intellectual property (IP) [4]. As a key advantage, the flexibility of RISC-V allows customization and improvement of the ISA and the underlying hardware, tuning them to meet specific (reliability) requirements. Consequently, the RISC-V ecosystem is increasingly adopted in applications where system reliability is non-negotiable [5].

Within this context, previous investigations have tested the reliability of various RISC-V processors, including processors with architectural protection, under neutron and proton radiation [2], [6], [7], [8]. However, these investigations have been limited to field-programmable gate array (FPGA) implementations of these processors, leaving the configuration memory vulnerable to SEUs.

This article focuses on the experimental evaluation of *Trikarenos* [9], an SoC designed to address the reliability requirements of both automotive and space applications. Trikarenos leverages the flexibility of the RISC-V architecture,

Received 28 January 2025; revised 14 March 2025; accepted 22 April 2025. Date of publication 28 April 2025; date of current version 18 August 2025. This work was supported in part through the TRISTAN project that has received funding from Chips Joint Undertaking (CHIPS-JU) under grant agreement no. 101095947. CHIPS JU receives support from the European Union's Horizon Europe's research and innovation programme and Austria, Belgium, Bulgaria, Croatia, Cyprus, Czechia, Germany, Denmark, Estonia, Greece, Spain, Finland, France, Hungary, Ireland, Israel, Iceland, Italy, Lithuania, Luxembourg, Latvia, Malta, Netherlands, Norway, Poland, Portugal, Romania, Sweden, Slovenia, Slovakia, and Turkey. *(Michael Rogenmoser and Philip Wiese contributed equally to this work.) (Corresponding author: Michael Rogenmoser.)*

Michael Rogenmoser, Philip Wiese, and Frank K. Gürkaynak are with the Integrated Systems Laboratory, ETH Zürich, 8092 Zürich, Switzerland (e-mail: michaero@iis.ee.ethz.ch; wiesep@iis.ee.ethz.ch; kgf@iis.ee.ethz.ch).

Bruno Endres Forlin is with the Computer Architecture for Embedded Systems Group, University of Twente, 7522 Enschede, The Netherlands (e-mail: b.endresforlin@utwente.nl).

Paolo Rech is with the Department of Industrial Engineering, University of Trento, Trento, Italy (e-mail: paolo.rech@unitn.it).

Alessandra Menicucci is with the Department of Space Engineering, Delft University of Technology, 2629 Delft, The Netherlands (e-mail: a.menicucci@tudelft.nl).

Marco Ottavi is with the Computer Architecture for Embedded Systems Group, University of Twente, 7522 Enschede, The Netherlands, and also with the Department of Electronic Engineering, University of Rome Tor Vergata, 00133 Rome, Italy (e-mail: m.ottavi@utwente.nl).

Luca Benini is with the Integrated Systems Laboratory, ETH Zürich, 8092 Zürich, Switzerland, and also with the Department of Electrical, Electronic, and Information Engineering, University of Bologna, 40126 Bologna, Italy (e-mail: lbenini@iis.ee.ethz.ch).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNS.2025.3564739.

Digital Object Identifier 10.1109/TNS.2025.3564739

integrating a triple-core lockstep (TCLS) system alongside ECC and scrubbing in memory, implemented in TSMC 28-nm technology. This work investigates the architectural protection features of Trikarenos under neutron and proton radiation, as well as gate-level simulations, evaluating its resilience for automotive and space contexts. Specifically, our contributions include the following:

1) the measurement of static random access memory (SRAM) vulnerability within Trikarenos, focusing on the effectiveness of ECC protection and scrubbing techniques;
2) an evaluation of the TCLS mechanism, demonstrating its capacity to maintain the functionality of the processor in the event of a fault;
3) a performance assessment of Trikarenos in various radiation environments, including an atmospheric setting dominated by neutrons and an orbital environment characterized by protons;
4) evaluation of Trikarenos' vulnerability to SEUs using simulation-based fault injection, comparing to experimental results.

II. Background

A. Radiation Hardening Techniques

Radiation hardening employs a range of strategies to enhance the resilience of systems against radiation-induced faults, particularly in demanding environments such as space. These strategies can be broadly categorized into software-based fault tolerance techniques, radiation-hardened technologies, and design-level mitigation approaches [10], [11]. Trikarenos applies radiation hardening by design (RHBD) techniques at the physical layout, circuit architecture, or system levels to mitigate the effects of radiation on integrated circuits. One widely used RHBD technique is TMR, which provides fault tolerance by replicating hardware modules three times and using a majority voter to determine the correct output. While highly effective at mitigating faults, TMR imposes overheads in terms of power and area, making it best suited for applications requiring high reliability. ECC offers another effective approach, particularly for protecting data stored in memories and registers. By appending parity bits to the primary data, ECC enables the detection and correction of single-bit errors and, in certain implementations, multi-bit errors. Compared to replication-based methods, ECC provides efficient static data protection with minimal resource overhead. By combining these techniques, radiation-hardened systems can meet the high-reliability requirements of space missions and other high-radiation environments while minimizing penalties in performance and cost.

B. RISC-V Processors

In recent years, the adoption of RISC-V has globally advanced, including the space domain driven by mandates in Europe and high-profile projects such as NASA's High-Performance Spaceflight Computing (HPSC), which incorporates RISC-V cores designed by SiFive [12]. Unlike proprietary ISAs such as x86 and ARM, RISC-V offers a completely open and extensible architecture, enabling designers and researchers to implement custom modifications and specialized designs without requiring licensing fees. In comparison to other ISAs such as scalable processor architecture (SPARC) and PowerPC, RISC-V delivers a modern, modular architecture with extensibility at its core. This design philosophy allows for the inclusion of custom instructions tailored to specialized workloads, addressing the needs of evolving computational demands, as well as custom modification to the designs, such as those targeting reliability.

One example of the practical potential of RISC-V is PULPissimo [13], an open-source microcontroller architecture developed at ETH Zürich. It is designed for flexibility and accessibility and offers a configurable processing core, low-latency interconnect, and energy-efficient peripherals. Its modular design enables the integration of custom components, making it adaptable to a wide range of applications and specialized use cases.

C. Reliable Microprocessors

A variety of radiation-tolerant SoCs are available commercially. Processors such as the RAD750 [14] and newer RAD5500 [15] feature a PowerPC architecture with a proven track record operating in space. Frontgrade Gaisler's GR716A [16] and GR716B offer a radiation-tolerant microcontroller platform using the SPARC ISA, with newer high performance designs such as the GR765 and NOEL-V SoC [17] offering RISC-V cores.

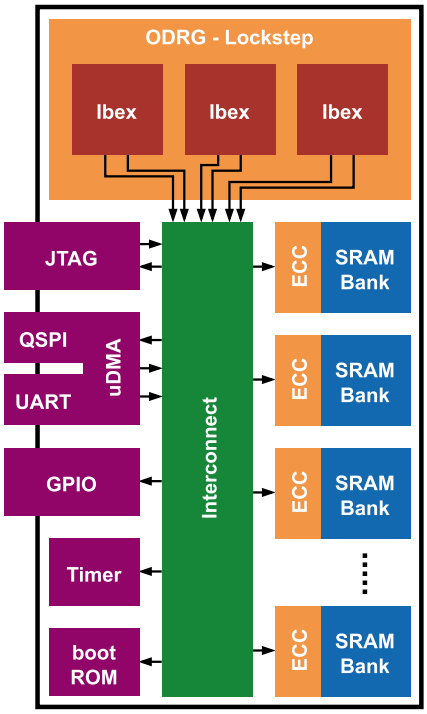
While these systems often report tolerance to radiation, this reporting is limited to total ionizing dose (TID) and single event latchup (SEL) linear energy transfer (LET). They do not report the tolerance to SEUs or the cross section thereof under proton or neutron radiation, making it difficult to appropriately compare to the results in this work.

III. SoC Architecture Description

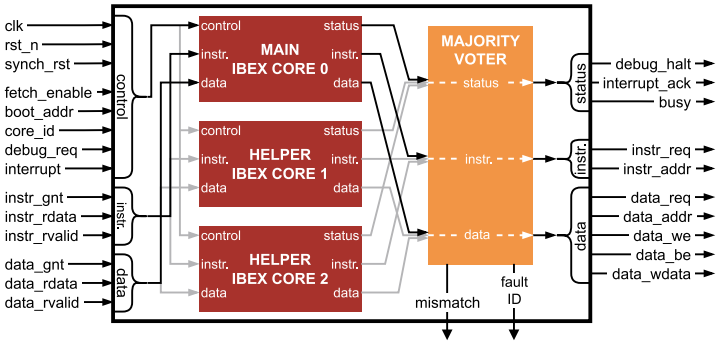
The Trikarenos architecture, presented in [9], implements an adaptation of PULPissimo, an RISC-V-based microcontroller design. As shown in Fig. 1(a), Trikarenos includes three physically separated Ibex cores [18] connected to a low latency interconnect, exposing 256 KiB of SRAM memory. Furthermore, the SoC features a boot memory, timers, peripherals, such as general purpose input/outputs (GPIOs), universal asynchronous receiver-transmitter (UART), and quad serial peripheral interface (QSPI) with direct memory access (DMA), and a JTAG debug unit for programming.

A. Processing Core Reliability

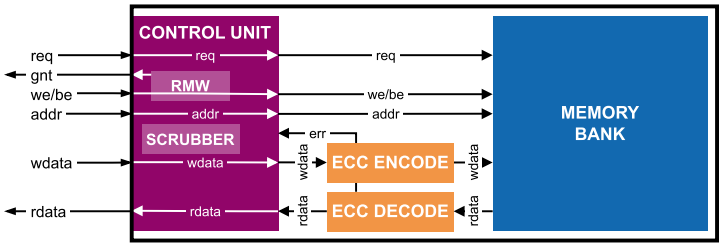
To boost tolerance to SEUs at an architectural level, Trikarenos' cores are augmented with on-demand redundancy grouping (ODRG) [11], a configurable TCLS mechanism. In the locked mode, the three cores' outputs, including instruction fetch and data request ports, are voted. The cores' inputs, including the instruction, data response, and interrupt requests, are identical. This design, illustrated in Fig. 1(b), ensures the cores all receive the same data, and any error propagating outside the cores is detected and corrected. Once an error is detected at the interface, a recovery routine is started, ensuring



(a) Block diagram of Trikarenos SoC, containing three Ibex cores for fault-tolerant operation, ECC-protected memory, and peripherals, connected using a low latency interconnect.



(b) Diagram of the triple-core lockstep (TCLS) mechanism with three Ibex cores. The three core's input signals, split into control, instruction, and data, are identical, with the output signals voted, with additional information indicating an error and the responsible core.



(c) Diagram of the SRAM error correction scheme. Each 32-bit word is stored in 39 bits with a single error correction, double error detection Hsiao code. A scrubber continuously checks the memory for errors, ensuring real-time correction without system interruptions.

Fig. 1. Trikarenos SoC architecture, including (a) overall system block diagram, (b) TCLS mechanism for core fault tolerance, and (c) SRAM error correction and scrubbing system. The design is optimized for fault tolerance in automotive and space applications.

the internal state of the cores is saved to memory, corrected through the voters, and rewritten to ensure an identical starting point for lockstep operation. The TCLS recovery mechanism adds up to 700 cycles overhead for the recovery routine in case a mismatch is detected within the cores. In case a critical code section is being executed, this recovery can be delayed. Functionally, these three cores operate as a single core within the system.

As the TCLS protection is applied outside the processor cores at their interface, only individual voter cells are added in these paths, with the remaining logic NOT affecting the critical path. Even if the critical path for the design is through this interface, the impact on frequency remains minimal and was not observed to affect the target frequency for Trikarenos.

The system allows the three cores to operate independently in parallel for increased performance or in locked reliability mode, where the cores work together to detect and correct mismatches for fault tolerance. The mode switching is managed at runtime via system control registers. This investigation focuses on the locked reliability mode, focusing on its capability of detecting and correcting any mismatch in the cores' operation.

B. Memory Reliability

The on-chip SRAM memory, occupying most of the design area, is split into eight word-interleaved 32-bit memory banks, enabling low contention and supporting byte-wise writes. For reliability, each 32-bit word is stored in 39 bits with a single error correction, double error detection (SECDED) Hsiao code, resulting in effectively 2 555 904 bits for the 256-kB memory.

As shown in Fig. 1(c), a control unit is connected to the encoders and decoders for each memory bank, supporting an efficient read–modify–write architecture for sub-word writes, immediately accepting the operation to allow the system to continue, delaying the subsequent access in case the bank is accessed in the following cycle to re-encode the stored data properly. With the word-interleaved banks in Trikarenos, repeated access to the same bank is reduced significantly, and as only a very small amount of accesses to memory are writes of less than 32 bit, the performance impact is negligible.

The ECC encoding and decoding logic add some additional latency on the data path before the memory, which could have an impact on the maximum achievable frequency. However, the use of an efficient ECC such as the Hsiao code limits the logic levels added to the path, and as other signals, such as handshake signals and address, have longer critical paths than the logic in the dataplane, this increase does not affect the achievable frequency.

To ensure that latent errors do not accumulate within the memory, which would result in uncorrectable data, each

memory bank is equipped with a scrubber, continuously scanning the memory bank contents at a configurable rate. Furthermore, the scrubber delays its memory bank access if the system requests data from the memory bank in the same cycle, thus having no impact on the system's performance. In case the scrubber detects an error, it is directly corrected, and the correction event is logged.

C. SoC Design

Along with the protected core and memory banks, Trikarenos' SoC features a low-latency interconnect between the core, memory banks, and additional SoC components such as debug module and peripherals including UART, serial peripheral interface (SPI), and GPIOs. While the interconnect is critical to operation, it is far smaller in the area it occupies within the system. In an integrated system, the peripherals and debug module are used far less extensively than the processing core; however, they have the ability to both send requests into and receive requests from other components through the interconnect. Interconnect, debug module, and other peripherals are not modified from the base PULPissimo design to harden against radiation effects.

To track the SoC's status, the architecture features a variety of telemetry registers counting errors in the system. This allows efficient readout of errors in the memory banks detected on access and detected by the scrubber, both for correctable and uncorrectable faults. Along with the interconnect, peripherals, and debug module, the TCLS voting logic and ECC encoders and decoders are not hardened for reliability. Faults in these areas can lead to functional errors such as timeouts, exceptions, or incorrect results. However, the voting logic and ECC encoders and decoders represent a relatively small fraction of the overall area compared to the cores and memory banks.

D. Implementation

Trikarenos is implemented in TSMC 28HPC+, a 28-nm bulk process that has been shown to be tolerant to TID effects [19]. Furthermore, TSMC 28HPC+ offers a good balance between performance, power efficiency, and accessibility, making it a practical choice for developing radiation-tolerant designs without requiring custom radiation-hardened processes. The design was implemented relying on standard components and cells without hardened registers, adding physical separation between the processor cores but not adding any additional consideration for clock and reset tree hardening. The implementation targets operating frequencies up to 250 MHz (at 0.9 V), occupying 0.17 mm^2 for logic and 0.56 mm^2 for SRAM in the 2-mm^2 design [9] shown in Fig. 2. In the experimental setup, Trikarenos operates at 125 MHz and 0.9 V.

Fig. 3 provides a detailed breakdown of the area and flip-flop (FF) distribution across key components of the design, including the cores, interconnect, peripherals, memory, and debug modules. To provide context for the errors seen in experiments, one core contains 2398 flip-flops (FFs), whereas the SoC contains 20 921 FFs.

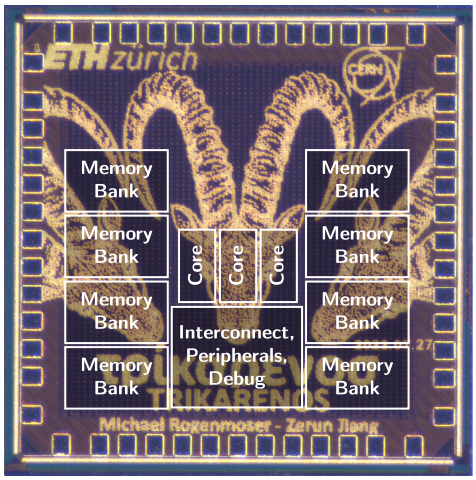


Fig. 2. Annotated die shot of Trikarenos, highlighting the spatial distribution of the three Ibex cores, memory banks, interconnect, and key peripherals within the 2-mm^2 die.

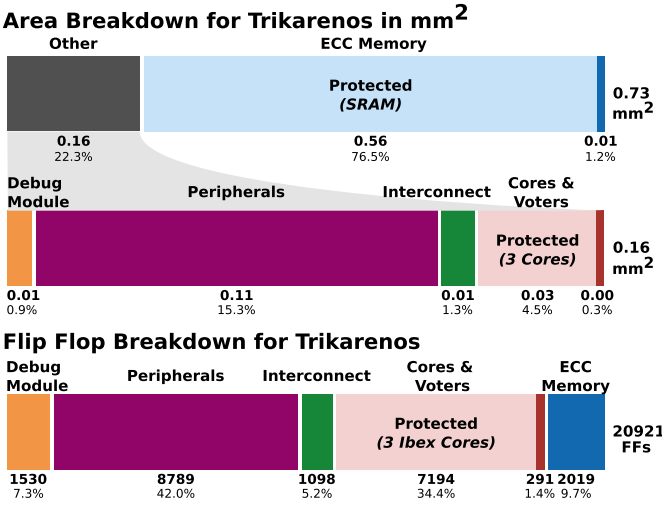


Fig. 3. Area and FF breakdown for the Trikarenos SoC. The three Ibex cores and memory banks are protected, while the remaining components remain unprotected.

IV. Evaluation Methodology

Two methods were used to evaluate the Trikarenos SoC. Primarily, beam experiments were used to determine the vulnerable cross section in different parts of the design. Fault injection in a simulation environment was also performed, relating the experimental cross section to FF vulnerability.

A. Testing Facilities

We conducted experiments at the ChipIR and HollandPTC facilities to assess the sensitivity of Trikarenos to neutron and proton radiation. The ChipIR beamline, located at the Rutherford Appleton Laboratories, U.K., is designed to replicate the energy spectrum of atmospheric neutrons [20]. On the other hand, HollandPTC, located in Delft, The Netherlands, is equipped with a fixed horizontal proton beamline, delivering a beam with nominal energies ranging from 70 up to 240 MeV. Its Cyclotron can deliver nominal beam currents from 1 up to 800 nA and a maximum flux of 1.13×10^9 p cm^{-2} s^{-1} [21].

TABLE I
OBSERVABLE ERROR TYPES AND CLASSIFICATION

Observed Error	Class
Single SRAM Errors (scrubber)	Correctable
Multiple SRAM Errors (on access & scrubber)	Functional Error
TCLS Events	Correctable
Miscalculations (silent data corruption (SDC))	Functional Error
System Crashes	Functional Error
SEL	Total Loss

B. Error Model

Our experiments focus on analyzing SEEs in the RISC-V-based Trikarenos SoC. For the TSMC 28-nm technology, other research has already investigated the TID effects [19], showing extreme tolerance levels.

We investigate the error mechanisms in Table I, which are traceable with the SoC telemetry registers and given test applications.

While most error sources can be traced back to SEUs or multiple/accumulation thereof, system crashes can have different causes. They can be traced back to a single event functional interrupt (SEFI) or an SEU affecting an unprotected, critical component, such as the system's interconnect. While some GPIO signals and routines are designed to assist in differentiating these faults, exact cause identification remains difficult.

Finally, most data corruption will be detected by one of the two mitigation mechanisms. ECC protection in the SRAM memory will catch an error affecting stored data, and the TCLS mechanism will detect data corruption in the cores. For a particle strike in the cores, it is more than likely that only a single core will be affected due to their separation, thus resulting in different outputs from the cores and an error in the voter while storing the result back in memory.

C. Experimental Setup

The test setup, shown in Fig. 4, comprised a dedicated testing printed circuit board (PCB) containing the device under test (DUT) and a test harness, a monitoring device with an interface board. A Raspberry Pi (RPi) single-board computer served as the primary control and monitoring system for the entire test suite. The presence of a Linux operating system on the RPi enabled the development of flexible and modular software in Python, which facilitated comprehensive monitoring and control capabilities. The interface board supplied the necessary 1.8-V I/O and 0.9-V core voltages for the Trikarenos SoC, in addition to incorporating a level shifter, dual clock generation circuits, and several status LEDs for real-time feedback and diagnostics. This setup controlled the power supply to the DUT, managed binary loading into the SRAM via JTAG, and monitored GPIO pins while interfacing with the DUT via JTAG and UART for data acquisition.

The test application running on the DUT was carefully designed to emulate a realistic application and maximize the propagation probability and observability of bit-flips inside the cores and memory. To emulate a realistic application, Trikarenos executed the Coremark benchmark, implementing a variety of workloads to test a processor core. The final version for the proton tests augmented this with a section continuously executing simple NOT operations on the processor core's registers filled with a checkerboard pattern and periodically saving these values to memory. Saving the register contents uses the TCLS voters to detect any errors. This maximized the exposure time of the vulnerable components, allowing for a possible mismatch between the cores in TCLS to trigger an error.

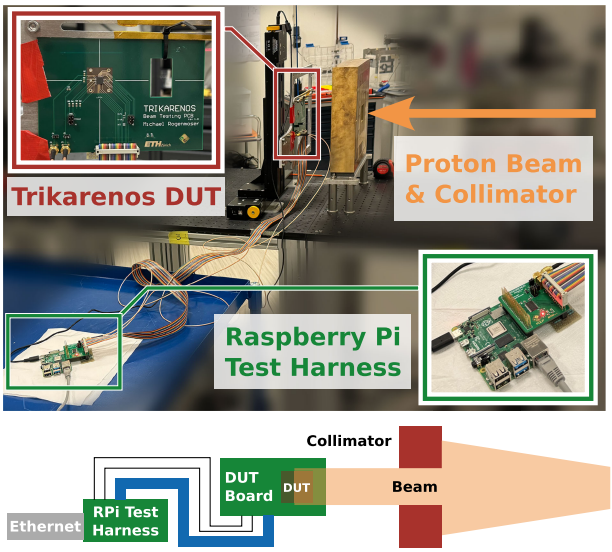


Fig. 4. Annotated image of testing setup at HollandPTC (top) and schematic of the testing setup (bottom), showing the proton beam, the DUT, and the test harness.

For testing, the DUT application facilitated real-time error monitoring by periodically transmitting the status of memory-mapped registers that count errors through the UART and JTAG interfaces. These registers include error counters that log recoverable TCLS events detected by majority voting, ECC access errors that track memory read/write operations with invalid ECC bits and scrubber counters that distinguish between corrected and uncorrectable errors in the memory. The test harness acquired this transmitted data, additionally monitoring other GPIO signals designed to indicate other system failures. Finally, in the event of critical, non-recoverable errors, the monitoring system would halt the cores and dump all relevant registers and memory areas for subsequent analysis. This comprehensive approach, combined with multiple redundant recovery mechanisms and extensive logging strategies, equipped the setup for effective long-term and unsupervised radiation testing.

Data integrity and logging were critical concerns addressed in our setup. All test data were logged locally on NAND flash memory using a secure digital (SD) card. To ensure the test data is properly backed up in case of SD card corruption, the data and the RPi's system logs (which can assist in tracing errors) were transmitted over ethernet to a computer outside the beam room for later processing. To detect latch-up events, a current monitor and a current-limited power switch were added to the test harness to prevent permanent damage to the DUT.

D. Simulation-Based Fault Injection

To determine the vulnerability of the design to SEUs in a more targeted manner, we performed extensive fault injection simulations on the post-layout gate-level netlist. The fault injection simulations were performed using *Siemens QuestaSim 2023.4* with the help of custom scripts.[1] These scripts initially launch a normal simulation of the design without any modification, using this as a golden result to compare to, saving an initial checkpoint after setup has been completed, and storing the final state of all registers as a reference. Once this reference is available, the scripts re-run the final part of the simulation from the checkpoint, selecting a random signal and flipping it at a random timestep in a given window using the `force` command. The final result is then compared to the golden reference, differentiating correct results, functional errors, and timeouts, also keeping track of latent errors still in the design's registers. These simulation runs are then repeated with different randomization seeds, storing the results for different tests.

During the simulation campaign, single bit-flips were injected into all registers of the SoC. Bit-flips in registers or FFs are intended to represent SEUs, both caused by direct flips within the registers or single event transients (SETs) sampled by an FF disrupting functionality. Trikarenos executed the Coremark benchmark, followed by 100 iterations of NOT operations on the processor core's registers, representing a single iteration of the program loop run during the beam experiments. The injection scripts were adapted to differentiate the correct results further, checking for recoveries with TCLS but ensuring that the Coremark check returned a correct result. Latent errors were filtered into two categories: some errors remain only in the originally injected register, not propagating through the system, indicating that this register has limited functionality or the component injected is not used. While propagating errors may indicate future issues with the device, the result and functionality were not impacted. Functional errors were also differentiated, separating a simulation timeout, indicating an internal issue or infinite loop, an unexpected system exception, or an incorrect result from the test.

V. Experimental Beam Results

Throughout all our tests, we did not observe any uncorrectable SRAM errors, miscalculations, or SELs indicated by a change in power consumption. Therefore, the results below only differentiate correctable SRAM errors, correctable TCLS events, and uncorrectable errors that result in a system crash. Because some errors are rare or possibly even zero, we calculate upper and lower estimates of the cross section assuming a Poisson distribution and use the relationship between the cumulative distribution function of the Possion and the chi-squared distribution [22]. To calculate these estimates σ, we use

$$\frac{1}{2F}\chi^2\left(\frac{\alpha}{2}, 2N\right) < \sigma < \frac{1}{2F}\chi^2\left(1-\frac{\alpha}{2}, 2(N+1)\right) \quad (1)$$

where $\chi^2(p, n)$ is the quantile function of the chi-squared distribution with n degrees of freedom, α describes the $100(1-\alpha)\%$ confidence interval, F is the total exposed fluence, and N is the number of detected errors. All calculations below use a 95% confidence interval.

To differentiate between errors attributed to the test setup and those inherent to the DUT, we conducted a detailed analysis of the system logs from the RPi. Our evaluation identified that most errors originating from the setup involved SD card-related timeouts and kernel segmentation faults.

TABLE II
Cross Section Results of Trikarenos Under Neutron Beam for a Fluence of 6.88×10^{11} Neutrons/cm^2

Component	Error Count	Errors / bit	Cross-Section[†]
Single SRAM Errors	18 786	7.35×10^{-3}	2.77×10^{-8} cm^2
TCLS Events	13	-	3.23×10^{-11} cm^2
System Crashes	0	-	5.36×10^{-12} cm^2

[†] Upper-bound with a 95 % confidence interval & Poisson distribution. This assumes one additional fault, i.e., device failure after the tests.

A. Neutron Tests

At the ChipIR facility [20], Trikarenos was exposed to a total fluence of 6.88×10^{11} n cm^{-2}. While active, the beam produced an average neutron flux of 4.90×10^6 n cm^{-2} s^{-1} with an energy spectrum representing that found for atmospheric neutrons at ground level up to 800 MeV. The measurements are shown in Fig. 5, and the errors and estimated cross sections are summarized in Table II.

1) Memory Errors: During the neutron exposure, we observed 18 786 errors with a median error rate of 492 error h^{-1}. All of these were single errors detected and corrected by the scrubber that checked the hardware ECC encoding. Using (1) allows us to estimate the cross section of the memory to $2.75(2) \times 10^{-8}$ cm^2. With 2 555 904 bits, we measured an average bit error rate of 1.92×10^{-4} error bit^{-1} h^{-1} for a cross section per bit of $1.08(1) \times 10^{-14}$ cm^2 bit^{-1}. This rate is around 11% lower than the findings in [23], which evaluated the 14-MeV neutron sensitivity of an SRAM manufactured in the same technology.

2) Core Errors: During the experiment, we measured 13 correctable TCLS events. We estimate the cross section of recoverable TCLS events to be $2.55(68) \times 10^{-11}$ cm^2. No unrecoverable faults were measured. Thus, we can only estimate the upper-bound cross section of the unrecoverable events to 5.36×10^{-12} cm^2.

To contextualize, we consider a terrestrial environment to provide estimations for mean time to failure (MTTF). Assuming an integral flux density of 20 n cm^{-2} h^{-1} above 1 MeV corresponding to the ground radiation level in New York [24], the minimum MTTF for a terrestrial application with an active TCLS recovery mechanism is above 1.06 million years.

B. Proton Tests

Trikarenos was exposed to a total fluence of 2.91×10^{12} p cm^{-2} over 55 min of active beam time, a notably high value. This value directly results from our

[1] https://github.com/pulp-platform/InjectaFault/

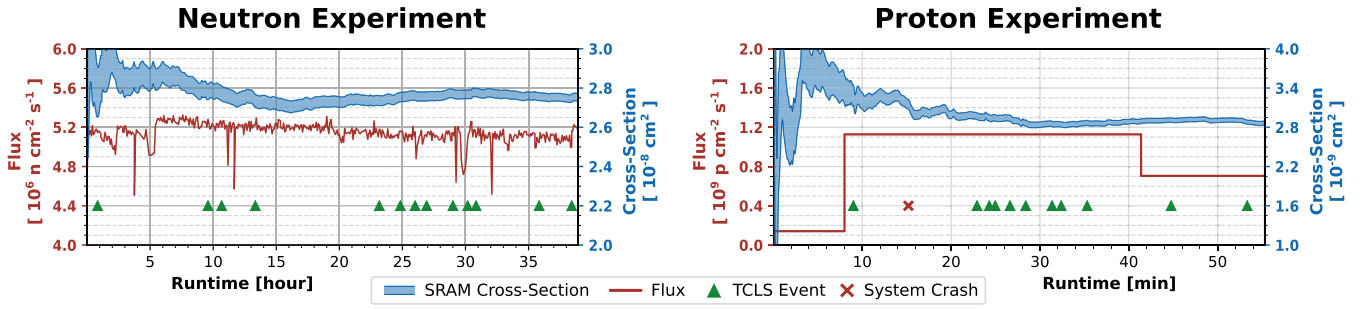


Fig. 5. Visualization of induced errors and flux rates during both radiation tests of Trikarenos. We show the cross sections, flux rates, TCLS events, and system crashes over time. We use an atmospheric-like neutron beam [20] and 200-MeV protons. For the proton tests, no TID-related degradation was observed.

TABLE III
CROSS SECTION RESULTS OF TRIKARENOS UNDER PROTON BEAM FOR A FLUENCE OF 2.91×10^{12} Protons/cm^2

Component	Error count	Errors / bit	Cross-Section†
Single SRAM Errors	8249	3.23×10^{-3}	2.89×10^{-9} cm^2
TCLS Events	11	-	6.76×10^{-12} cm^2
System Crashes	1	-	1.91×10^{-12} cm^2

† Upper-bound with a 95 % confidence interval & Poisson distribution. This assumes one additional fault, i.e., device failure after the tests.

TABLE IV
SIMULATION RESULTS OF FAULT INJECTION INTO FFS OF TRIKARENOS

	Total		In Cores	
Termination Reason	Amount	Probability	Amount	Probability
Correct Termination	**99 096**	**99.10 %**	**34 524**	**100.00 %**
Correct	18 958	18.96 %	7642	22.14 %
TCLS	12 283	12.28 %	11 806	34.20 %
Latent Non-Propagating	61 514	61.51 %	13 805	39.99 %
Latent Propagating	6341	6.34 %	1271	3.68 %
Functional Error	**904**	**0.90 %**	**0**	**0.00 %**
Timeout	822	0.82 %	0	0.00 %
Exception	42	0.04 %	0	0.00 %
Incorrect	40	0.04 %	0	0.00 %
Total	**100 000**	**100.00 %**	**34 524**	**100.00 %**

stress testing of the system with the highest flux available in the facility. We tested a range of beam currents for a 200-MeV proton with the flux rate extracted from calibration data provided by the facility operator. Fig. 5 displays the measurements, and Table III presents the observed errors alongside estimated cross sections.

1) Memory Errors: We measured 8249 errors for a median error rate of 9180 error h^{-1}, all corrected by the scrubbers, leading to SRAM cross section 2.86(3) $\times$ 10^{-9} cm^2. This results in a cross section per bit of 1.12(1) $\times$ 10^{-15} cm^2 bit^{-1}.

2) Core Errors: During the experiment, we measured 11 recoverable errors and one unrecoverable error. Our data indicate that an unresponsive debug module caused the unrecoverable error, an element of the SoC currently not protected. We estimate the cross section of recoverable TCLS events to 5.25(151) $\times$ 10^{-12} cm^2 and the upper bound for unrecoverable faults to 1.91 $\times$ 10^{-12} cm^2. Using this, we can calculate the minimum mean fluence to failure (MFTF) of the system with the TCLS recovery mechanism to 5.23 $\times$ 10^{11} p cm^{-2}.

3) Other Faults: Trikarenos received a TID of more than 1.77 $\times$ 10^3 Gy throughout the measurements with protons. No degradation in performance or TID-related errors were observed. Furthermore, the observed current consumption remained constant within noise-induced bounds, indicating no SELs.

VI. SIMULATION-BASED FAULT INJECTION

In the simulations, 100 000 faults were injected into randomly selected registers of 20 921 flip − flop(FF) in a randomly selected cycle in 29 516 cycles where the main application loop was active, with a single simulation only having one traceability fault. All registers of the design were injectable and used to check for latent errors, while the SRAM memory was not injected. Table IV shows that over 99% of the injected faults lead to correct application termination. While most of the correct terminations still included latent errors, most of these latent errors did not affect the remaining SoC, indicating that these registers or components are not used in the application tested.

Investigating the confidence of the injection results, based on [25], we assumed a normal distribution for high event counts, such as correct terminations (99 096) and functional errors (904). Using a conservative probability of ($p = 0.5$), we achieved an absolute margin of error of ±0.31% with a 95% confidence interval. This corresponds to lower bounds of 98.79% for correct terminations and an upper bound of 1.21% for functional errors. For rare events such as exceptions (42) and incorrect terminations (40), where the normal approximation is less suitable, we used the Poisson distribution, resulting in intervals of 0.030%–0.055% and 0.028%–0.053%, respectively. Overall, less than 1% of injections resulted in functional errors. Furthermore, all injections inside one of the three cores resulted in the correct termination, and the majority either had no effect or were corrected by the TCLS mechanism. Assuming that a watchdog timer is used to handle timeouts and exceptions of the system, only 0.04% of errors lead to potentially undetected failures.

1) Core Errors: Of the 100 000 faults injected, 34 524 faults were injected into the three processor cores' 7194 FF, indicated by the right-most columns in Table IV. Of these, none caused a functional error, with 34.20% triggering a TCLS recovery event, ensuring correctness. While latent errors still exist for this configuration, any

TABLE V
FUNCTIONAL ERRORS OF FAULT INJECTION INTO FFS OF TRIKARENOS FOR THE DIFFERENT MODULES

Components	Functional Errors			Fault / Injections
	Timeout	Exception	Incorrect	
Interconnect	613	22	26	661 / 5176
Debug Module	164	0	1	165 / 7379
Peripherals	26	12	13	51 / 41 867
Cores & Voters	19	2	0	21 / 35 934
Memory	0	6	0	6 / 9644
Total	**822**	**42**	**40**	**904 / 100 000**

remaining errors in the cores will be corrected if they cause a divergence in a future operation. Alternatively, a re-synchronization routine can be executed to eliminate all latent errors in the cores. This process involves saving the full state of the cores to memory on the main software stack, resetting the faulty core, and restoring its state from the saved data. The programmer can trigger the routine manually or periodically, effectively mitigating the risk of error accumulation over time.

Furthermore, the TCLS mechanism does contain some FFs; none are directly part of the voters. Thus, any error in the voting can be attributed to interconnect or system errors.

2) Functional Errors: A closer look at the remaining functional errors in Table V reveals that 73% of system-level issues are caused by bit-flips in the interconnect, a critical component of the design that currently lacks protection. In addition, 18% of errors originate from faults in the JTAG debug module. While these primarily affect testing and simulation infrastructure, their impact on a real-world application is likely to be minimal but could still benefit from isolation from the rest of the system. In addition, 6% of errors are due to bit-flips in the peripherals, which only accounts for a small portion of the final errors. However, peripherals can still affect the remaining system through interrupt signals or by driving the main memory bus, as is possible with Trikarenos' peripheral subsystem. Unprotected, disabled components could greatly benefit from proper isolation from the rest of the SoC. Although these peripherals are not heavily utilized in the current tests, their role would become far more critical if Trikarenos were deployed in a complete system, such as a satellite, where latent errors in these modules could pose significant risks. Finally, 3% of errors are caused by bit-flips inside the cores in the ODRG wrapper and the ECC modules.

VII. DISCUSSION

A. Memory Errors

Although the application running may impact the TCLS and total fault rate, the SRAM fault rate checked by the scrubber should not be affected. We observe a $> 9\times$ higher cross section of the SRAM for neutrons when comparing neutron and proton fault rates. A precise cause for this requires further investigation but could be attributed to variations in particle types, energy levels, and differences in the devices used across tests. However, this provides an initial foundation for the validity of neutron beam experiments to evaluate the radiation tolerance of new devices, even in other domains.

We can effectively tune the device's scrub rate using SRAM cross sections of $1.08(1) \times 10^{-14}$ cm^2 bit^{-1} for neutrons and $1.12(1) \times 10^{-15}$ cm^2 bit^{-1} for protons. Trikarenos' scrubber was set to the maximum possible rate for the experiments to ensure that all errors were caught without accumulation. This can be reduced to limit power consumption. With individual scrubbers for each of the eight memory banks, each scrubber is responsible for 8192 words. To ensure that the entire memory space is checked between the mean failure rate, the scrubber must complete its memory check every 408 ms for the proton test condition, leading to a word check per bank every 6225 cycles.

B. Core Errors

In the tested operation mode, Trikarenos detects all corruptions within the cores, both those that may lead to system corruption due to an erroneous branch, corrupted pointer or misdecoded instruction, and data corruption.

Operating in Trikarenos' independent mode, a performance improvement of up to $2.96\times$ with three cores in parallel or a 35% reduction in core power consumption with a single core is possible. However, in this independent mode, erroneous control flow may lead to stalls that a watchdog timer can catch or silent data corruption (SDC) can lead to erroneous data. Assuming that all faults in a core lead to a system crash, the device cross section in parallel operation increases by at least $3.5\times$ from $2.55(68) \times 10^{-11}$ cm^2 for neutrons and by at least $2.14\times$ from $5.64(156) \times 10^{-12}$ cm^2 for protons. For single-core operation, assuming an equal distribution of errors in the cores, the cross section improvement is at least $1.12\times$ from $1.08(48) \times 10^{-11}$ cm^2 for neutrons. Further measurements can help narrow down the improvement, likely showing a significant increase in the reliability factor. Without SRAM protection, the improvement is much more significant, even considering that only a subsection of the memory is used and that some data corruption may not cause a system crash.

The TCLS recovery routine, requiring around 600 cycles or 4.8 μs at 125 MHz, has a negligible impact on performance for the observed error rates.

C. Fault Injection Simulation

The results presented in Table IV show that all faults injected into the protected part of the SoC are corrected by the TCLS mechanism, demonstrating its effectiveness. In addition, most injections result in latent state mismatches that do not propagate outside the cores or do not affect any other component. However, these latent errors may affect future computations, where the TCLS mechanism will detect and correct them.

D. Comparison of the Experiments and Fault Simulation

The effective average FF cross section can be estimated by combining experimental and simulation results. This is done by adjusting the observed TCLS recoveries using their likelihood from the fault injection simulations using the following

expression:

$$\sigma_{\mathrm{FF}} = \frac{\sigma_{\mathrm{TCLS}}}{(P_{\mathrm{TCLS}} * N_{\mathrm{FF}})} \tag{2}$$

where σ_{TCLS} is the measured TCLS cross section, P_{TCLS} is the probability that a bit-flip leads to an TCLS events, and N_{FF} is the total number of FFs in Trikarenos. It is important to note that this cross section reflects the combined effect of raw physical susceptibility and fault masking within the architecture. Individual FFs may exhibit different susceptibilities depending on their function and the execution context, and the timing of fault injection can further influence error occurrence. Consequently, this averaged model inherently includes variability that is not captured explicitly in the calculation. We estimate an average FF cross sections of $0.99(29) \times 10^{-14}$ cm^2/FF for atmospheric neutrons and $2.04(64) \times 10^{-15}$ cm^2/FF for 200-MeV protons. The neutron value aligns with the measurement reported by Fabero et al. [26] for 14 MeV, and the proton value is of the same order of magnitude as that reported by Borghello et al. [27]. Furthermore, the FF cross section during neutron experiments is similar to the SRAM bit-error cross section, while, for proton experiments, the FF cross section appears higher. While the error margins for these estimations remain quite large, further consideration may need to be given to the fault injection methods, as only FF errors were considered in this investigation, leaving out SET in logic blocks affecting the state.

E. Other

The neutron and proton experiments also revealed vulnerabilities in the telemetry registers, showing flips of specific bits reflecting immediate, immense increases in measured error counts. While not critical for performance or operations, such registers or similar SoC control registers highlight latent errors that do not affect the computation result seen in simulation and may be worth protecting in future designs.

In addition, comparing the functional errors in Table V to the area or the FF count shown in Fig. 3 reveals that vulnerability is not directly correlated with either. This observation aligns with the fact that not all components are equally critical or active during operation. For instance, while the interconnect occupies a relatively small portion of the system's FFs, it remains a key source of system-level issues due to its critical role in data transfer, which currently lacks protection mechanisms.

Finally, while unrecoverable errors were very rare in our investigations, it was shown that they are still possible, requiring watchdog circuitry or further protection in future designs. For the Trikarenos SoC, their root cause can be traced back to the interconnect, debug module, and peripherals, which would benefit from fault tolerance in a future design.

VIII. Conclusion

Exposing the Trikarenos SoC to neutrons and protons, we observe most errors affecting the SRAM for a cross section per bit of $1.08(1)$ and $1.12(1) \times 10^{-15}$ cm^2 bit^{-1} for atmospheric neutrons and 200-MeV protons, respectively, protected by ECC, thus not affecting the system's function. Trikarenos' TCLS mechanism corrects most of the remaining errors, reducing the cross section by at least 3.5× from $2.55(68) \times 10^{-11}$ to below 5.36×10^{-12} cm^2, showing TCLS' efficacy and Trikarenos' validity for use in critical environments.

Moreover, the TCLS mechanism completely protects the cores, preventing all functional errors when bit-flips are injected into the core regions. Across the entire SoC, only 0.90% of bit-flips result in functional errors, with the majority occurring in the interconnect, one of the critical components of the design that remains unprotected.

While this work shows that protecting the largest elements of a microprocessor makes it very robust toward SEUs, both induced by particles and in simulation, it further highlights the remaining vulnerabilities within the system. Protecting very active areas such as the interconnect is critical within the design, and enabling proper fault detection and recovery mechanisms or isolation for peripherals merits further research. Finally, the large number of non-propagating latent errors observed during fault injection simulations, 61.51% of injected faults, indicates that significant portions of the SoC remained inactive in the presented experiments. Future work is needed to better understand and decrease the number of latent errors remaining in the design, requiring increased observability and a focused software testing methodology, properly considering a wider spectrum of workloads.

Acknowledgment

The authors thank Gianluca Furano from ESA, Thomas Toet from HollandPTC, Edian Annink, and Elijah Cishugi for their support in testing Trikarenos.

References

[1] B. Narasimham et al., "A hysteresis-based D-flip-flop design in 28 nm CMOS for improved SER hardness at low performance overhead," *IEEE Trans. Nucl. Sci.*, vol. 59, no. 6, pp. 2847–2851, Dec. 2012. [Online]. Available: https://ieeexplore.ieee.org/document/6365409

[2] A. E. Wilson, M. Wirthlin, and N. G. Baker, "Neutron radiation testing of RISC-V TMR soft processors on SRAM-based FPGAs," *IEEE Trans. Nucl. Sci.*, vol. 70, no. 4, pp. 603–610, Apr. 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10012379

[3] A. Neale and M. Sachdev, "Neutron radiation induced soft error rates for an adjacent-ECC protected SRAM in 28 nm CMOS," *IEEE Trans. Nucl. Sci.*, vol. 63, no. 3, pp. 1912–1917, Jun. 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7497732

[4] S. Di Mascio, A. Menicucci, E. Gill, G. Furano, and C. Monteleone, "Open-source IP cores for space: A processor-level perspective on soft errors in the RISC-V era," *Comput. Sci. Rev.*, vol. 39, Feb. 2021, Art. no. 100349. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013720304494

[5] A. Walsemann, M. Karagounis, A. Stanitzki, and D. Tutsch, "STRV—A radiation hard RISC-V microprocessor for high-energy physics applications," *J. Instrum.*, vol. 18, no. 2, Feb. 2023, Art. no. C02032. [Online]. Available: https://iopscience.iop.org/article/10.1088/1748-0221/18/02/C02032

[6] A. B. de Oliveira et al., "Evaluating soft core RISC-V processor in SRAM-based FPGA under radiation effects," *IEEE Trans. Nucl. Sci.*, vol. 67, no. 7, pp. 1503–1510, Jul. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9096401

[7] Á. B. de Oliveira and F. L. Kastensmidt, "Evaluating fault-tolerant techniques on COTS RISC-V NOEL-V processor in Zynq UltraScale+ FPGA under proton testing," *IEEE Trans. Nucl. Sci.*, vol. 70, no. 8, pp. 1708–1715, Aug. 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10138560

[8] D. A. Santos et al., "Hybrid hardening approach for a fault-tolerant RISC-V system-on-chip," *IEEE Trans. Nucl. Sci.*, vol. 71, no. 8, pp. 1722–1730, Aug. 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10540019

[9] M. Rogenmoser and L. Benini, "Trikarenos: A fault-tolerant RISC-V-based microcontroller for CubeSats in 28 nm," in *Proc. 30th IEEE Int. Conf. Electron., Circuits Syst. (ICECS)*, Istanbul, Turkey, Dec. 2023, pp. 1–4. [Online]. Available: https://ieeexplore.ieee.org/document/10382727

[10] ECSS Secretariat, "Space engineering: Engineering techniques for radiation effects mitigation in ASICs and FPGAs handbook," ESA Requirements Standards Sect., Noordwijk, The Netherlands, Tech. Rep. ECSS-E-HB-20-40A, Oct. 2023. [Online]. Available: https://amstel.estec.esa.int/tecedm/website/docs_generic/ECSS-E-HB-20-40A(11October2023).pdf

[11] M. Rogenmoser, Y. Tortorella, D. Rossi, F. Conti, and L. Benini, "Hybrid modular redundancy: Exploring modular redundancy approaches in RISC-V multi-core computing clusters for reliable processing in space," *ACM Trans. Cyber-Phys. Syst.*, vol. 9, no. 1, pp. 1–29, Jan. 2025. [Online]. Available: https://dl.acm.org/doi/10.1145/3635161

[12] S. Leibson, "NASA recruits microchip, SiFive, and RISC-V to develop 12-core processor SoC for autonomous space missions," *EEJournal*, Jan. 2023. [Online]. Available: https://www.eejournal.com/article/nasa-recruits-microchip-sifive-and-risc-v-to-develop-12-core-processor-soc-for-autonomous-space-missions/

[13] P. D. Schiavone, D. Rossi, A. Pullini, A. Di Mauro, F. Conti, and L. Benini, "Quentin: An ultra-low-power PULPissimo SoC in 22nm FDX," in *Proc. IEEE SOI-3D-Subthreshold Microelectron. Technol. Unified Conf. (S3S)*, Burlingame, CA, USA, Oct. 2018, pp. 1–3.

[14] D. Rea, D. Bayles, P. Kapcio, S. Doyle, and D. Stanley, "PowerPC™ RAD750™—A microprocessor for now and the future," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, USA, Mar. 2005, pp. 2390–2394.

[15] R. Berger et al., "Quad-core radiation-hardened system-on-chip power architecture processor," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, USA, Mar. 2015, pp. 1–12.

[16] S. Habinc et al., "Rad-hard microcontroller for space applications," in *Proc. AMICSA DSP*, vol. 732. Gothenburg, Sweden: ESA, Jun. 2015, pp. 3–7. [Online]. Available: https://indico.esa.int/event/102/attachments/22/27/Proceedings_2016_AMICSA.pdf

[17] J. Andersson, "Gaisler NOEL-V SoC applications and ecosystem," ESA ESTEC, Noordwijk, The Netherlands, Tech. Rep. 6, 2022. [Online]. Available: http://microelectronics.esa.int/riscv/rvws2022/presentations/06_ESA_RISC-V_in_Space-NOEL-V.pdf

[18] P. Davide Schiavone et al., "Slow and steady wins the race? A comparison of ultra-low-power RISC-V cores for Internet-of-Things applications," in *Proc. 27th Int. Symp. Power Timing Modeling, Optim. Simulation (PATMOS)*, Thessaloniki, Greece, Sep. 2017, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/8106976

[19] G. Borghello et al., "Total ionizing dose effects on ring-oscillators and SRAMs in a commercial 28 nm CMOS technology," *J. Instrum.*, vol. 18, no. 2, Feb. 2023, Art. no. C02003, doi: 10.1088/1748-0221/18/02/c02003.

[20] C. Cazzaniga and C. D. Frost, "Progress of the scientific commissioning of a fast neutron beamline for chip irradiation," *J. Phys., Conf. Ser.*, vol. 1021, May 2018, Art. no. 012037. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1021/1/012037

[21] M. Rovituso et al., "Characterisation of the HollandPTC R&D proton beamline for physics and radiobiology studies," 2023, *arXiv:2302.09943*.

[22] N. L. Johnson, A. W. Kemp, and S. Kotz, "Poisson distribution," in *Univariate Discrete Distributions*. Hoboken, NJ, USA: Wiley, 2005, p. 176. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/0471715816.ch4

[23] V. Vargas et al., "Radiation experiments on a 28 nm single-chip many-core processor and SEU error-rate prediction," *IEEE Trans. Nucl. Sci.*, vol. 64, no. 1, pp. 483–490, Jan. 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7779088

[24] M. Gordon et al., "Measurement of the flux and energy spectrum of cosmic-ray induced neutrons on the ground," *IEEE Trans. Nucl. Sci.*, vol. 51, no. 6, pp. 3427–3434, Dec. 2004. [Online]. Available: http://ieeexplore.ieee.org/document/1369506/

[25] R. Leveugle, A. Calvez, P. Maistri, and P. Vanhauwaert, "Statistical fault injection: Quantified error and confidence," in *Proc. Design, Autom. Test Eur. Conf. Exhib.*, Apr. 2009, pp. 502–506. [Online]. Available: https://ieeexplore.ieee.org/document/5090716

[26] J. C. Fabero et al., "Single event upsets under 14-MeV neutrons in a 28-nm SRAM-based FPGA in static mode," *IEEE Trans. Nucl. Sci.*, vol. 67, no. 7, pp. 1461–1469, Jul. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9020124

[27] G. Borghello, D. Ceresa, R. Pejasinovic, F. P. Diaz, G. Bergamin, and K. Kloukinas, "Single event effects characterization of a commercial 28 nm CMOS technology," in *Proc. Topical Workshop Electron. Part. Phys. (TWEPP)*, Geremeas, Italy, Oct. 2023, pp. 55–56. [Online]. Available: https://indico.cern.ch/event/1255624/contributions/5443894/