Characterization of HARV-SoC for Reliable Avionics Applications

Wesley Grignani, Douglas A Santos, Carolina Imianosky, Maria Kastriotou, Carlo Cazzaniga, Luigi Dilillo

Title: Characterization of HARV-SoC for Reliable Avionics Applications

Author(s): Wesley Grignani, Douglas A. Santos, Carolina Imianosky, Maria Kastriotou, Carlo Cazzaniga, and Luigi Dilillo

Document version: Pre-print version (Final draft)

Please cite the original version:
Wesley Grignani, Douglas A. Santos, Carolina Imianosky, Maria Kastriotou, Carlo Cazzaniga, and Luigi Dilillo, ”Characterization of HARV-SoC for Reliable Avionics Applications”. 1st Workshop on REliable and SeCUrE RISC-V architectures - RESCUER, May. 2025, Tallinn, Estonia.

Characterization of HARV-SoC for Reliable Avionics Applications

Wesley Grignani*, Douglas A. Santos*, Carolina Imianosky*, Maria Kastriotou†,
Carlo Cazzaniga†, and Luigi Dilillo*
*IES, University of Montpellier, CNRS, Montpellier, France
†ISIS Facility, STFC, Rutherford Appleton Laboratory, Oxfordshire, UK
{wesley.grignani, douglas.santos, carolina.imianosky, luigi.dilillo}@umontpellier.fr
{maria.kastriotou, carlo.cazzaniga}@stfc.ac.uk

Abstract

Advancements in avionic systems technology demand enhanced resilience against environmental hazards such as radiation, which can compromise the reliability of electronic components through different effects. Among these effects, SEUs (Single-Event Upsets) represent a critical concern, as they induce unintended bit flips in memory elements, affecting data integrity and the system's reliability. The Hardened RISC-V System-on-Chip (HARV-SoC) is a soft-core implementation and was designed to ensure fault tolerance in such conditions through Error Correcting Code (ECC) and redundancy mechanisms, such as Triple Modular Redundancy (TMR). This work presents the HARV-SoC, exploring more features that enhance error analysis and improvements than previous implementations. A fault injection campaign was performed at the ChipIr neutron radiation facility to assess the reliability of the circuit. The results of HARV-SoC are present considering resource utilization and detailed analysis of the observed radiation-induced faults. The system demonstrated robustness through the error handling mechanisms mitigating SEUs and enhanced capabilities for reporting errors, which is essential to ensure processor integrity and minimize failures. These findings highlight the suitability of HARV-SoC for safety-critical avionics applications and provide insights for future enhancements in fault-tolerant RISC-V architectures.

Index Terms

Systems-on-Chip, Fault Tolerance, RISC-V, Radiation Effects, Single-Event Effects, Single-Event Upsets.

I. Introduction

Avionics systems operate in safety-critical environments that demand extreme dependability and resilience against faults. Thus, these systems, often comprising powerful processing units, must be fail-safe and ensure continuous service delivery. To achieve this, each application sector must adhere to specific standards and guidelines to meet these requirements, including the automotive [1], aerospace [2], [3], and military industries [4]. With the emergence of the RISC-V architecture, many projects have explored its use across different domains. Its open and modular design has made it particularly attractive for high-reliability and safety-critical applications, allowing fault-tolerance strategies to meet system demands [5].

However, one of the biggest challenges to reliability in avionics is the adverse radiation environment encountered at high altitudes [6], [7]. In the atmosphere, high-energy neutrons and other ionizing particles generated from cosmic rays can induce Single-Event Effects (SEEs) in electronic components, inducing flipping bits or disrupting logic and thus threatening stable operation [8]. Without proper mitigation, these radiation-induced transient or permanent faults can corrupt data, cause application errors, and lead to unexpected system crashes or hangs [9]. Ensuring reliability, therefore, requires both robust fault-tolerant design and thorough validation of system behavior under such conditions.

To address these challenges, processors that are used in safety-critical applications incorporate various fault-tolerance techniques. In practice, they often employ spatial, temporal, and informational redundancy to detect and correct errors, using, for example, error-correcting codes, watchdog timers, and dual/triple modular redundancy [10]. Developers validate the effectiveness of these techniques via fault injection campaigns (injecting errors in simulations or FPGA prototypes) or by exposing hardware to particle accelerators. Such testing helps to ensure that the system can tolerate fault conditions before deployment.

In this context, the Hardened RISC-V System-on-Chip (HARV-SoC), a soft-core fault-tolerant RISC-V System-on-Chip (SoC) designed for high reliability, combines multiple redundancy mechanisms to mitigate faults at various levels of the architecture. This work presents a detailed characterization of HARV-SoC in the context of reliable avionics applications, focusing on its behavior under neutron irradiation tests. In particular, we report results from a neutron

This work was supported in part by the Region d'Occitanie and the École Doctorale I2S from the University of Montpellier (contract no. 00137932/22009671), the EU project RADNEXT – Horizon 2020 (Grant 101008126), and Project HARV (project PE24PR01) in the framework of the action "Accelerateur d'innovation" of the University of Montpellier.

beam test campaign performed at the ChipIr facility, which provides a high-flux beam representative of the atmospheric neutron spectrum. Through these experiments, we assess HARV-SoC's resilience to radiation-induced faults and evaluate the effectiveness of its built-in fault-mitigation techniques in meeting stringent avionics reliability requirements.

The remainder of this paper is structured as follows. Section II introduces the HARV-SoC architecture and its fault-tolerant mechanisms. Section III details the neutron irradiation experiment setup, including the testing methodology and system configuration. Section IV presents the experimental results and analyzes the observed radiation-induced faults. Finally, Section V provides concluding remarks and discusses potential future directions for enhancing the reliability of the HARV-SoC.

II. HARV-SoC

The HARV-SoC is a soft-core implementation of an SoC composed of the HARV (Hardened RISC-V) core, bus interconnections, and peripherals. The HARV core has a multi-cycle implementation of the RV32I base instruction set [11] and supports the Control-Status Registers (CSRs) modules, compressed instructions, multiplication, and vector.

The main focus of this SoC is to provide a reliable processor implementation that can be used in harsh radiation environments without suffering failures that would impact the correct functioning of the application. The HARV core applies the Error Correcting Code (ECC) and Triple Modular Redundancy (TMR) techniques in its internal structures to detect and correct errors. More specifically, the register file, the program counter, and the instruction register are hardened using ECC with Single Error Correction, Double Error Detection (SECDED) capabilities. Meanwhile, the control unit, ALU, and critical CSRs use TMR [12].

Besides the hardening, HARV provides fault awareness through error reporting [13]. These error reports were implemented as custom RISC-V exceptions and provide detailed information about the error through memory-mapped registers. The event identifier flags the identified error in the processor, the one that triggered the exception, and the other registers provide further information, such as the error type, application context, and corrupted data.

In addition to the base extension, the CSRs are essential for more complex applications, such as real-time operating systems. Real-time operating systems rely on exceptions and interrupts to deal with the context-level switching between tasks, such as task schedulers. These traps in the RISC-V specification are configured by CSRs, including the `mtvec`, `mie`, and `mstatus`. Therefore, we hardened these CSRs with TMR with self-correction, ensuring no failures would affect these essential features of the processor.

HARV-SoC comprises the HARV processor core and various peripherals (e.g., communication interfaces and memory controllers) integrated into an architecture that targets reliability with flexibility and low hardening overheads. This reliability is achieved by hardening critical internal components and providing recovery schemes, such as data memory with ECC, bus access timeout, reset controller, watchdog timer reset, and application checksum.

In addition to the error reports in the processor core, the SoC also provides error reporting for the bus and peripherals, which are integrated through a HARV's external port. The default implementation of the HARV-SoC includes the external error reports for peripheral timeout and memory access. The error reports of the memory access are based on ECC and include simple single- and double-bit upsets and stuck-at single- and double-bit upsets with SECDED capabilities.

For the data memory, the SoC uses an external memory accessed through a customized controller that enables SECDED for each 32-bit word. The bus controller provides a timeout flag for access to peripherals, triggering an exception when a peripheral is not responding. With the bus timeout, the application receives information about the faulty peripheral. The application may use this information to perform actions to restore the functionality of a peripheral. For that, the reset controller is used, which enables resetting specific peripherals. Besides the timeout from the bus, the system also provides a Watchdog Timer (WDT) that resets the SoC when a system timeout is reached, detecting and recovering hang processor failures.

HARV-SoC was developed to be flexible to use in different FPGAs (Field-Programmable Gate Arrays), which may use other technologies that cause different effects when exposed to radiation. While flash-based FPGAs have a more resilient configuration memory, SRAM-based FPGA's most common errors are in its configuration memory. Therefore, HARV-SoC implements an interface with a primitive structure of Xilinx devices that reports the errors in the configuration memory through a built-in scrubbing. The reports provided by this structure are integrated with a specific error handler, which also interfaces with HARV to report these errors to the application.

In addition to the hardening strategies, the startup routine performs an application integrity verification using a CRC-32 (Cyclic Redundancy Check 32) algorithm. This enables the detection of errors while loading the program from external non-volatile memories. Furthermore, the error monitor's reporting capabilities allow the application to take action according to the detected error. If the error is not correctable, the application uses the available information to decide, for example, to ignore it and continue the execution or even force a system reset to avoid faulty system behavior.

In this work, we present a follow-up neutron irradiation experiment presented in [13]. The present work explores more features regarding the error analysis and some improvements in the HARV-SoC implementation. The enhancements include a more detailed report of the external memory ECC unit, the addition of stuck-at bit errors in the external memory ECC, the implementation of TMR in critical CSRs, and improved handling of the failure-inducing exceptions by the application. These modifications enhance the system's fault tolerance and error detection capabilities, contributing to a more robust and resilient design.

III. Radiation Experiment

This section introduces the implemented system using the HARV-SoC for the physical test in the ChipIr neutron irradiation facility. Next, the board setup and details of the application benchmark used to evaluate the HARV-SoC are described.

A. ChipIr Neutron Irradiation Facility

The ChipIr beamline is part of the ISIS Neutron and Muon Source at the Rutherford Appleton Laboratory, United Kingdom. The ChipIr instrument is representative of the atmospheric neutron spectrum, which causes SEEs on electronic devices [14]. This instrument irradiates the DUT (Device Under Test) with a flux up to 10^9 times greater than the typical natural radiation environment [15]. This high flux enables accelerated testing of electronic devices. The average flux provided by this beamline is 5.6×10^6 n/cm^2/s for energies above 10 MeV.

B. System Design

The system was designed using the HARV-SoC with some modifications for the experiment. Fig. 1 shows an overview of the system developed for the experiment using the SmartFusion2 FPGA.

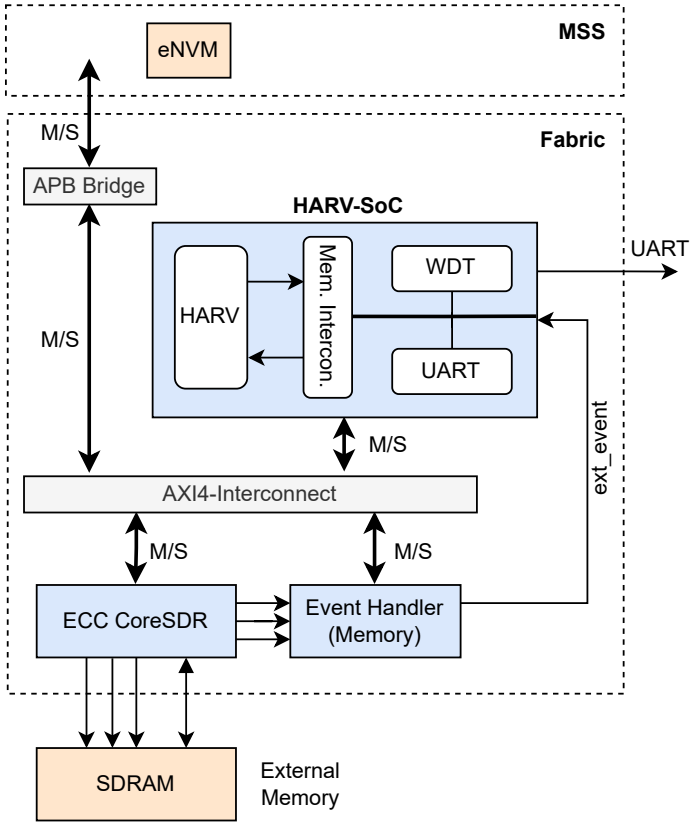

Fig. 1. System design for the experiment using HARV-SoC.

The HARV-SoC comprises the HARV processor connected via the internal bus with components such as the Watchdog timer and UART. Externally, an AXI4-Lite interconnection for Manager and Subordinate (M/S) was used to communicate between the HARV-SoC and other components. In addition, we provided an APB bridge to convert communication between AXI4-Lite and APB protocol to access the Embedded Non-volatile Memory (eNVM) on the Microcontroller Subsystem (MSS) side. This memory stores the program to be executed by the processor.

We used an external Synchronous Dynamic Random Access Memory (SDRAM) available on the board as a data memory for the processor. This memory was used as it provides more storage space and is more resilient to radiation than the conventional SRAM in the FPGA. We used the CoreSDR IP, a controller that provides a high-performance interface to Single-Data-Rate SDRAM devices to interface with the SDRAM. We enhanced memory reliability by applying Hamming ECC for SECDED. Therefore, we incorporated additional logic with CoreSDR IP to store the data in memory with ECC. When an error is detected, this component outputs information about single- and double-bit upsets to the memory Event Handler.

The memory Event Handler forwards exceptions to the HARV error handler via external event input. It detects bus access timeouts and single- or double-bit memory upsets. Additionally, it provides details such as the address of the memory error, associated ECC information, encoded data with the uncorrected word, and the previous error address.

C. Boards Setup

In this experiment, we used two SmartFusion2 FPGAs running the implemented system. Fig. 2 shows the position of the FPGA boards, highlighted as B1 and B2, where the other boards belong to different experiments. The two boards were used with separate and monitored power supply channels. For logging the experimental data, we used serial connections to transmit the logs generated to a host computer outside the irradiation room.

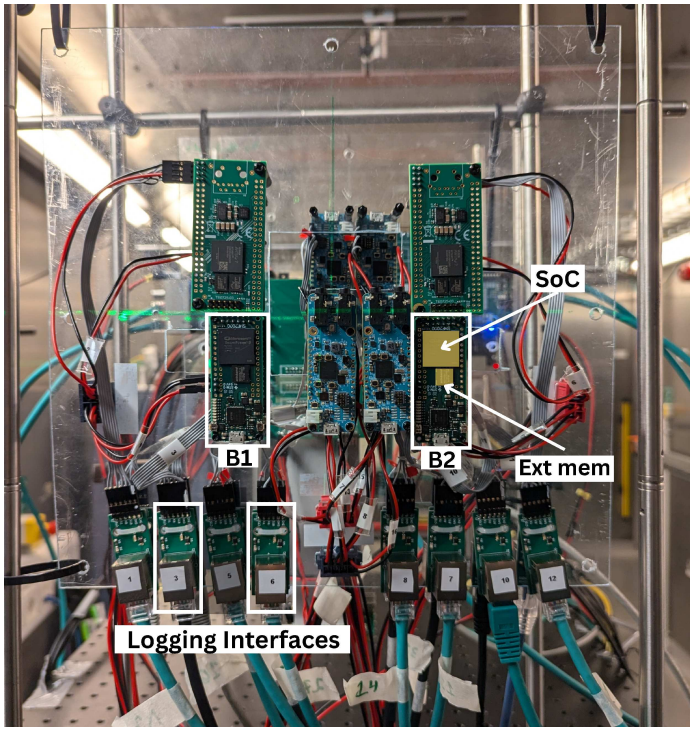

Fig. 2. Experiment setup composed of two SmartFusion2 FPGA boards.

The FPGA used has a flash-based configuration memory. Unlike volatile SRAM-based FPGAs, flash-based configuration cells were demonstrated to be highly resistant to neutron radiation [16], improving the availability of the platform for the experiment. In this setup, we supply power to the board through specific pins instead of using the conventional USB port. In this case, the FTDI chip of the board is not powered, which could prevent errors in the FPGA caused by radiation-induced events in this component.

D. Benchmark

We used the CoreMark [17] workload to sensitize the SoC during the experiment. This benchmark comprises list processing, matrix manipulation, state machines, and CRC algorithms. The error handler policy and exception triggers enable the application to manage radiation-induced events and continue execution accordingly. In the case of an uncorrectable error, the application may proceed, restart, or attempt recovery before failure occurs. We focused on reporting the errors in the experiment, so the benchmark was cyclically executed until errors were detected. The

exception handling routine reports all the available information, corrects or ignores errors based on the hardening configuration, and resumes normal execution.

Each FPGA hosted the HARV-SoC running CoreMark, which cycles between two system configurations: HARV with error correction disabled (ECD) and HARV with error correction enabled (ECE). These configurations comprise the same implemented circuit with the fault-tolerant techniques, but error correction is disabled in the HARV-SoC (ECD) through a configuration register so that errors are only detected and not corrected. This configuration is not applied to the SDRAM, meaning both versions used external memory protected by ECC. Thus, the CoreMark application runs sequentially on each board, switching between versions.

IV. Experiment Results

The fault injection campaign occurred at the ChipIr neutron irradiation facility in Oxfordshire, UK, from July 12th to 15th, 2024. This section presents the synthesis results of the implemented system and the reliability results from the fault injection campaign.

A. Synthesis

We used the Libero SoC 2024.1 to implement the system and collect the synthesis results. Table I presents the resource utilization for the designed system targeting the FPGA M2S010-VFG400. Resource utilization is presented in 4-input Look-up-Tables (4LUTs), D-type Flip-Flops (DFFs), and uSRAM for each system component.

TABLE I
Synthesis results for the implemented system

Component	4LUTs	DFFs	uSRAM (1K)
HARV-SoC	8,616	3,817	1
ECC CoreSDR	1,128	566	0
Memory EH	558	553	0
Interconnects	141	110	0
Total	**10,518**	**5,046**	**1**
Available	12,084	12,084	22
Utilization	87.40%	42.32%	4.55%

As shown in Table I, HARV-SoC has the highest resource utilization, corresponding to 82% of the total 4LUTs and 75% of the total DFFs used by the implemented system. The HARV-SoC comprises the HARV core with enabled hardening techniques, which include TMR in the control unit, ALU, and the CSR registers. It also includes the Hamming ECC applied to the register file and the PC and instruction registers in the instruction fetch unit. The maximum frequency reported by the tool for this system was 38.47 MHz.

In the HARV-SoC, it is possible to observe that a uSRAM is used, which is implemented by the UART component. However, this memory corresponds to a FIFO memory in the data-receiving interface of the UART, which was not used in this experiment. The other components account for less resource utilization, including the modified CoreSDR, the Memory Event Handler (EH), and the Interconnects for the AXI4-Lite and APB buses.

B. Realiability Results

During the experiment, a total of 55 events were observed, considering the two boards. Table II shows the number of events for each component observed in the circuit and separated by each board. Within all the observed circuit components, most of the events that occurred were classified by the error handler as external, representing, in this case, single- or double-bit upsets in the external memory (SDRAM).

Some events were observed and identified in the two register file outputs, `rs1` and `rs2`. Considering the two boards, six events occurred in the HARV (ECE) version and 5 in the HARV (ECD) version. Unlike the HARV (ECE) version, the HARV (ECD) version can only detect errors, meaning the five benchmark runs used the uncorrected value in the specific register. In addition, it is essential to mention that all 11 events in the register file were single-bit upsets.

Based on each FPGA board's total number of events and the accumulated fluence, we calculated the cross-section and Failure In Time (FIT) as shown in Table III. We calculated the FIT based on terrestrial and avionics target environment, where the terrestrial refers to New York sea level, and avionics considering a typical cruising altitude about 300$\times$ the sea level flux [18]. Considering different environments, we used this metric to estimate the number of expected events in the circuit components in a billion hours.

TABLE II
Events classification for each observed component

Component	B1		B2	
	HARV (ECD)	HARV (ECE)	HARV (ECD)	HARV (ECE)
pc_sbu	0	0	0	0
pc_dbu	0	0	0	0
ir_sbu	0	0	0	0
ir_dbu	0	0	0	0
regfile_rs1_sbu	0	1	1	0
regfile_rs1_dbu	0	0	0	0
regfile_rs2_sbu	3	3	1	2
regfile_rs2_dbu	0	0	0	0
ctl_err	0	0	0	0
alu_err	0	0	0	0
ext_event	7	10	10	15
periph_timeout	0	1	0	0
Total	**10**	**15**	**12**	**17**

TABLE III
Cross-section and FIT based on the total number of events

Board	Fluence (n/cm^2)	Events	Cross-section (cm^2/device)	FIT_{NYC}
B1	2.291×10^{12}	25	1.091×10^{-11}	1.41
B2	2.288×10^{12}	30	1.311×10^{-11}	1.70

The overall cross-section was calculated to 1.201×10^{-11}cm^2/device considering the results of the two boards. The overall FIT considering the terrestrial and avionics target environments was estimated to be 1.56 and 468.39, respectively. As most of the events observed were in external memory, we consider that most of the expected events based on the FIT metric will be in this memory.

One reason for the difference in the number of events reported between the versions is that most events occur in memory locations (such as the register file and external memory) that are not always accessed. Errors are detected only when these values are read, meaning that events can occur in areas the application is not currently utilizing. This can lead to a discrepancy in the overall number of errors reported.

C. Error Analysis

The total number of errors per component is shown in Table IV, along with their respective calculated cross-section and FIT_{NYC}. Each error represents an event on board B1 or B2 summarized in Table III. 72% of the errors correspond to single-bit upsets in external memory, representing the higher cross-section, followed by the register file representing 20% of the total errors. In addition to reporting errors, we check that the application produced the expected result even in the presence of errors that the processor did not correct. In this case, if the application was unable to finish or finished with the incorrect result, it is considered that there has been a propagation of the observed error.

TABLE IV
Cross-section for each component error

Structure	N Errors	Cross-section (cm^2/device)	FIT_{NYC}
Memory SBU	40	8.735×10^{-12}	1.13
Register file SBU	11	2.402×10^{-12}	0.31
Memory DBU	2	4.367×10^{-13}	0.06
Instruction access fault	1	2.183×10^{-13}	0.03
Peripheral timeout	1	2.183×10^{-13}	0.03

The 40 single-bit upsets observed in the external memory did not affect execution since the memory is protected in both the HARV (ECE) and HARV (ECD) configurations. Although two double-bit upsets were detected and remained

uncorrected, the application was completed without errors. This suggests that the affected memory values did not propagate through the circuit in a way that could compromise the execution of the CoreMark.

In the register file, six of the eleven single-bit upsets occurred in the ECE version, eliminating the possibility of compromising the execution of the application. The other five events happened in the ECD version, in which four of them did not propagate, and the application was completed as expected. The other single-bit upset caused the PC value to go to 0, leading the processor to fetch an instruction from an unspecified address and causing an instruction access fault. In this scenario, the processor performed a soft reset through the WDT and return to executing the benchmark again.

Another error observed in the circuit was classified as a peripheral timeout. This type of error represents a communication issue between the HARV-SoC and a peripheral device, such as UART or eNVM memory in this system. Through the error report, we observed that the cause was an instruction access fault, indicating that the problem was probably caused by a fault in some component that interfaces with the eNVM memory, such as the AXI4 interconnection or the protocols conversion carried out on the APB bridge. In this case, the system could not access the memory but could recover by performing a system reset.

The FIT_{NYC} is presented for each component error in Table IV considering the results from the two boards. From the general FIT_{NYC} previously shown in Table III, we can observe in Table IV that the majority correspond to the external memory, followed by the register file.

Implementing protection techniques and observability mechanisms allowed us to gain insights into what has occurred within the circuit and how it has impacted the executed application. Observability is also crucial for assessing the processor's reliability, especially when new components are introduced or modified in the SoC.

D. Failure Characterization

In addition to the events previously mentioned, we also observed some system failures in which no output from the processor was received for more than 6 minutes. In this case, a power cycle of the FPGA was necessary for the processor to return to regular operation. Considering this, we calculated the cross-section of the failure characterization to $1.965 \times 10^{-12} cm^2$/device.

For each run in which this failure was observed, the processor was stuck in the CoreMark run, and no information about the event was received. The processor tried to reset the system but was not able to recover the system. This could mean the system entered this mode due to an event outside the fabric part. During the experiment, we monitored the current in the circuit. We observed that during these failures, there was a drop in current, which remained constant during the 6 minutes until the power cycle. This behavior suggests that the processor remained idle, waiting for a signal to perform an operation such as accessing the memory.

To better understand the cause of the failure, we reproduced the same experiment with a modification to the APB bus access via a pin external to the FPGA. In this way, while the application was running, we restricted access to the APB bus used to read from the eNVM memory. As a result, the same current drop and failure were observed. This may indicate that during the experiment, the faults observed were caused by events on the MSS side, which somehow affected communication with the eNVM memory.

V. Conclusion

In this work, we presented a system designed using the HARV-SoC for a neutron irradiation experiment in a particle accelerator using the SmartFusion2 FPGA. We introduced more features regarding the error analysis and improvements in the HARV-SoC. Furthermore, we presented the resource utilization and performed a reliability analysis of the radiation-induced events in the circuit.

The fault tolerance techniques and observability mechanisms allowed for an in-depth understanding of processor faults and their impact on the application. The reliability results system showed that the external memory suffered the most faults. Additionally, checking error propagation helped us determine when the application could function correctly despite uncorrected single- or double-bit upsets.

For future work, we intend to characterize the HARV-SoC system presented in this work using a different FPGA technology, specifically the PolarFire SoC FPGA. Comparing resource utilization and reliability is essential for understanding the impacts of radiation-induced events and how they can affect the circuit components of the HARV-SoC.

References

[1] G. Xie, Y. Li, Y. Han, Y. Xie, G. Zeng, and R. Li, "Recent advances and future trends for automotive functional safety design methodologies," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 5629–5642, 2020.

[2] A. Schwierz and H. Forsberg, "Assurance benefits of iso 26262 compliant microcontrollers for safety-critical avionics," in *SAFECOMP 2018: The 37th International Conference on Computer Safety, Reliability and Security*, B. Gallina, A. Skavhaug, and F. Bitsch, Eds. Springer International Publishing, 2018, pp. 27–41.

[3] European Cooperation for Space Standardization, “Description, implementation and general requirement,” European Space Agency, Tech. Rep. ECSS-S-ST-00C Rev.1, jun 2020.
[4] E. C. Gangl, “A case study on u.s. government military standard development,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 28, no. 7, pp. 40–45, 2013.
[5] S. Di Mascio, A. Menicucci, E. Gill, G. Furano, and C. Monteleone, “Leveraging the openness and modularity of risc-v in space,” *Journal of Aerospace Information Systems*, vol. 16, no. 11, pp. 454–472, 2019.
[6] P. Cannon, M. Angling, L. Barclay *et al.*, “Chapter 7 and 9 - Radiation impacts on satellites and Ionising radiation impacts on avionics and ground systems,” in *Extreme space weather: impacts on engineered systems and infrastructure*. Royal Academy of Engineering, 2013, [Online]. Available: https://www.raeng.org.uk/publications/reports/space-weather-full-report.
[7] J.-C. Boudenot, “Radiation space environment,” in *Radiation Effects on Embedded Systems*. Springer, 2007, pp. 1–9.
[8] M. Yang, G. Hua, Y. Feng, and J. Gong, *Fault-tolerance techniques for spacecraft control computers*, 1st ed. Wiley Publishing, 2017.
[9] H. Cho, S. Mirkhani, C.-Y. Cher, J. A. Abraham, and S. Mitra, “Quantitative evaluation of soft error injection techniques for robust system design,” in *Proceedings of the 50th Annual Design Automation Conference*, 2013, pp. 1–10.
[10] D. J. Sorin, *Fault Tolerant Computer Architecture*. Morgan and Claypool Publishers, 2009.
[11] A. Waterman and K. Asanović, *The RISC-V Instruction Set Manual Volume I: Unprivileged ISA*, December 2019.
[12] D. A. Santos, L. M. Luza, C. A. Zeferino, L. Dilillo, and D. R. Melo, “A low-cost fault-tolerant RISC-V processor for space systems,” in *2020 15th Design Technology of Integrated Systems in Nanoscale Era (DTIS)*, 2020, pp. 1–5.
[13] D. A. Santos, A. M. P. Mattos, D. R. Melo, and L. Dilillo, “Enhancing fault awareness and reliability of a fault-tolerant RISC-V system-on-chip,” *Electronics*, vol. 12, no. 12, 2023.
[14] C. Cazzaniga and C. D. Frost, “Progress of the scientific commissioning of a fast neutron beamline for chip irradiation,” *Journal of Physics: Conference Series*, vol. 1021, p. 012037, may 2018.
[15] C. Cazzaniga, M. Bagatin, S. Gerardin, A. Costantino, and C. D. Frost, “First tests of a new facility for device-level, board-level and system-level neutron irradiation of microelectronics,” *IEEE Transactions on Emerging Topics in Computing*, 2018.
[16] D. Dsilva, J.-J. Wang, N. Rezzak, and N. Jat, “Neutron SEE testing of the 65nm SmartFusion2 flash-based FPGA,” in *2015 IEEE Radiation Effects Data Workshop (REDW)*. IEEE, 2015, pp. 1–5.
[17] S. Gal-On and M. Levy, “Exploring coremark a benchmark maximizing simplicity and efficacy,” *The Embedded Microprocessor Benchmark Consortium*, vol. 6, no. 23, p. 87, 2012.
[18] JEDEC, “Measurement and reporting of alpha particle and terrestrial cosmic ray-induced soft errors in semiconductor devices,” Available at: https://www.jedec.org/standards-documents/docs/jesd-89a. Acessed: January 24, 2025, 2021.