Exploring the Impact of Soft Errors on the Reliability of Real-Time Embedded Operating Systems

electronics

MDPI

Article

Exploring the Impact of Soft Errors on the Reliability of Real-Time Embedded Operating Systems

Sarah Azimi [1,*], Corrado De Sio [1], Andrea Portaluri [1], Daniele Rizzieri [1], Eleonora Vacca [1], Luca Sterpone [1] and David Merodio Codinachs [2]

[1] Dipartimento di Automatica e Informatica (DAUIN), Politecnico di Torino, 10129 Turin, Italy
[2] European Space Research and Technology Centre (ESTEC), European Space Agency (ESA), 2200 AZ Noordwijk, The Netherlands
* Correspondence: sarah.azimi@polito.it

Abstract: The continuous scaling of electronic components has led to the development of high-performance microprocessors that are suitable even for safety-critical applications where radiation-induced errors such as Single Event Effects (SEEs) can have a significant impact on the performance and reliability of the system. This work is dedicated to investigating the reliability of systems based on programmable hardware and Real-time operating Systems (RTOS) in the presence of architectural faults induced by soft errors in the configuration memory of the programmable hardware. We performed a proton radiation test campaigned at PSI radiation facility to identify the fault model affecting the configuration memory of Xilinx Zynq-7020 reconfigurable AP-Soc Device. The identified fault model in terms of SEU and MBU clusters has been used to evaluate the impact of proton-induced faults on applications running within FreeRTOS on a Microblaze soft processor. A Single Event Multiple Upset fault model resulting from a proton test is presented, focusing on characteristics such as shape, size, and frequency of observed cluster of errors. We conduct two fault injection campaigns and analyze the results to assess the effect of cluster size on system reliability. Moreover, we discuss software exceptions caused by faults that can affect the hardware structure of the soft processor.

Keywords: MBU; proton radiation test; Real-Time Operating System; SEU; soft processor

Citation: Azimi, S.; De Sio, C.; Portaluri, A.; Rizzieri, D.; Vacca, E.; Sterpone, L.; Merodio Codinachs, D. Exploring the Impact of Soft Errors on the Reliability of Real-Time Embedded Operating Systems. *Electronics* **2023**, *12*, 169. https://doi.org/10.3390/electronics12010169

Academic Editor: Alexander Barkalov

Received: 15 December 2022
Revised: 23 December 2022
Accepted: 28 December 2022
Published: 30 December 2022

1. Introduction

Field-Programmable Gate Arrays (FPGAs) are becoming increasingly important for space and avionic applications due to their high flexibility and performance. Satellite lifetimes have increased far beyond 10 years, making hardware reconfigurability in-flight a demanded requirement [1]. The increasing complexity in applications and tasks has pushed Soft-core processors to be one of the cores commonly implemented using the programmable logic of the FPGAs. A soft-core processor as an IP-Core provides an easy way to combine the flexibility granted by the software with the performance of dedicated hardware accelerations [2].

Among the available solutions, Microblaze is an industry leader in FPGA-based soft processing solutions. The highly flexible architecture and configuration options make it very suitable for embedded applications due to the few resources required to be implemented on programmable hardware.

The increasing task complexity required for embedded systems has also led to the decrease of bare-metal applications and the adoption of Real-Time Operating Systems (RTOSs), which provide an efficient solution for meeting stringent real-time requirements [3].

The Free Real Time Operating System (FreeRTOS) has been established as an excellent choice when there are multiple tasks to be executed in an organized and predictable fashion. It has deterministic and predictable task scheduling, which allows the more critical tasks to

meet hard real-time executions. Scheduling tasks can be instrumented by priority and time slicing and it is designed to run on small microprocessors that need to perform multiple tasks deterministically [4].

However, when using a soft microprocessor in mission-critical applications, the reliability issues deriving from the exposure of the devices to ionizing radiation, such as Single Event Upsets (SEUs), should be considered [5–7]. Differently from hardwired microprocessors, the netlist of soft microprocessors such as Microblaze is implemented in the programmable hardware using the configuration memory (CRAM) of the FPGA. This memory can be corrupted by SEU, leading to micro-architectural faults in the hardware that can propagate to the application layer and, in the case of the usage of a microprocessor supporting an operating system, can lead to catastrophic results, especially in mission-critical applications [8].

The main contribution of this work is dedicated to performing a detailed evaluation of the impact of radiation-induced architectural faults affecting the application benchmarks running on the FreeRTOS of the Microblaze embedded soft processor. The analysis has been performed through a fault injection campaign while an accurate fault model consisting of different cluster patterns of Multiple Bit Upset (MBU) has been identified through proton radiation performed at Paul Scherrer Institute (PSI) radiation facility. We evaluated the effect of faults during the execution of different software applications on FreeRTOS supported by Microblaze implemented on Zynq-7020 FPGA while we performed a deep investigation on the outcome of the software application.

Please notice that the developed platform is not targeting the software-level fault injections, but is instead targeting the hardware faults and their impact on the execution of the software running in the operating system. In detail, the platform emulates SEUs and MBUs by injecting bitflips in the configuration memory of the FPGA, creating a hardware architectural fault that might propagate to the output of the system and create a wrong functionality.

The current work is organized as follows. Section 2 is dedicated to elaborating on radiation-induced effects in reconfigurable hardware. Section 3 is dedicated to related works. Section 4 describes the methodology and the fault models deriving from the radiation test experiment. Section 5 illustrates the proposed radiation analysis workflow. Section 6 reports the results of the experimental analysis. Finally, Section 7 draws conclusions and proposes further steps.

2. Background on Radiation Effects in Reconfigurable Hardware

Radiation can significantly impact the reliability of electronic systems [9]. This is a major issue in space and avionics applications, which can be exposed to high fluxes of high energy particles, and it is also one of the main concerns for the reliability of safety-critical systems working at sea level, where cascades of secondary particles due to solar activity interacting with the upper atmosphere can be the source of radiation-induced disturbance in the devices [10].

A single event upset (SEU) occurs when a particle, traversing an electronic device, interacts with a sensitive node and causes data corruption that may lead to different outcomes; for example, it can cause the termination of the application, create wrong outcomes (silent data corruption), system reset, and other malfunctions. The error is a transient effect since the functionality of the affected memory cell is usually not disrupted, but only its content has been corrupted. As a result, the value of the memory cell can be overwritten later by normal system execution, usually leading only to a temporary error.

The increasing technology scaling and the higher working frequency as well as the reduction of operating voltages have exacerbated this phenomenon, especially in CMOS technology [11]. In particular, Single Event Multiple Upsets (SEMUs) refer to when as a result of a single event, more than one memory element is corrupted. It can be the result of a particle interacting with multiple memory cells, which are usually physically close.

These events are a potential issue in reconfigurable hardware such as FPGA. Indeed, reconfigurable hardware is designed to be reprogrammed by writing binary content in its configuration memory. SEUs and SEMUs can cause errors in the configuration memory content that results in errors in the hardware architecture implemented in the device. This phenomenon is schematized in Figure 1 at different levels of abstraction. In addition, configuration memory is usually overwritten only during device programming or reconfiguration. This implies that a microarchitectural error due to failure in configuration memory may affect the system for a long time and cause more critical faults by directly affecting the hardware configuration of the device [12].

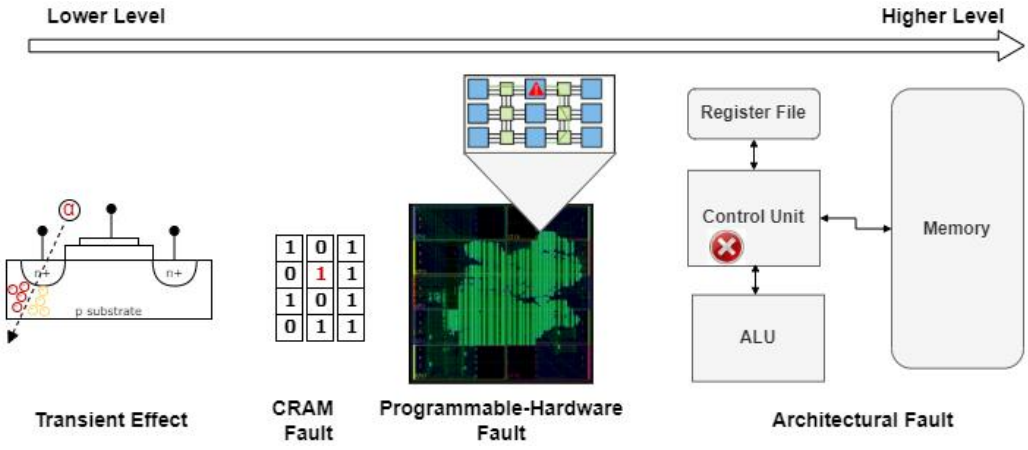

Figure 1. Effects of radiation-induced faults at different levels of abstractions.

3. Related Works

Several works elaborate on the software-level techniques for evaluating the sensitivity to Single Event Upsets (SEUs) of the embedded operating system. Commonly, these approaches are based on modifying the original kernel of the embedded operating system or altering either the memory that the OS uses or the parameters of system calls.

In Ref [13], the vulnerability of FreeRTOS has been evaluated through a software-based fault injection method that targets the most relevant variables and data structures of the OS. An automatic method for fault injection into program and data memory is presented in [14]. The authors of [15] proposed a detailed analysis and hardening architecture based on lockstep synchronization supporting FreeRTOS. The heavy ion irradiation test presented in [16] targets the SRAM and the special purpose registers of an ARM microcontroller to evaluate the impact of the radiation-induced SEU.

However, software application-level methods do not take into account the impact of faults occurring at the architectural level of the soft-core processor on the functionality of the operating system.

On other hand, works such as [17,18] focused on the reliability evaluation of soft cores, in particular RISC-V cores, implemented in programmable hardware against radiation effects. However, these analyses do not consider real-time operating systems such as FreeRTOS.

Other approaches are based on the simulation of the HDL description of microprocessors [19]. The advantage of these methods is the feasibility of injecting upsets into any CPU register and structure at any time; however, these methods are time consuming.

As far as the Single Event Multiple Upset is concerned, there is little literature available because most of the work, especially works focusing on reconfigurable hardware reliability, are dedicated to Single Event Upset evaluation that for less recent technologies largely contributes to the majority of errors. However, due to technology scaling and lower operating voltages, SEMUs are attracting more interest, partly because of their threat to redundant systems [20,21].

4. Proton Radiation Test-Based Fault Model

In order to perform the reliability analysis by fault injection campaigns with an accurate fault model, we have performed a proton radiation test at the Paul Scherrer Institute (PSI) Proton Facility in Switzerland.

4.1. Radiation Test Setup

A Zynq-7020 device has been irradiated with proton beams with energies between 29 and 200 MeV. The board under the test has been mounted on an adjustable frame in the irradiation room to point the beam to Zynq SoC embedded in the board, as shown in Figure 2. The serial and power cables reached out to the control room in order to enable reconfiguration, data acquisition, and power cycle of the board.

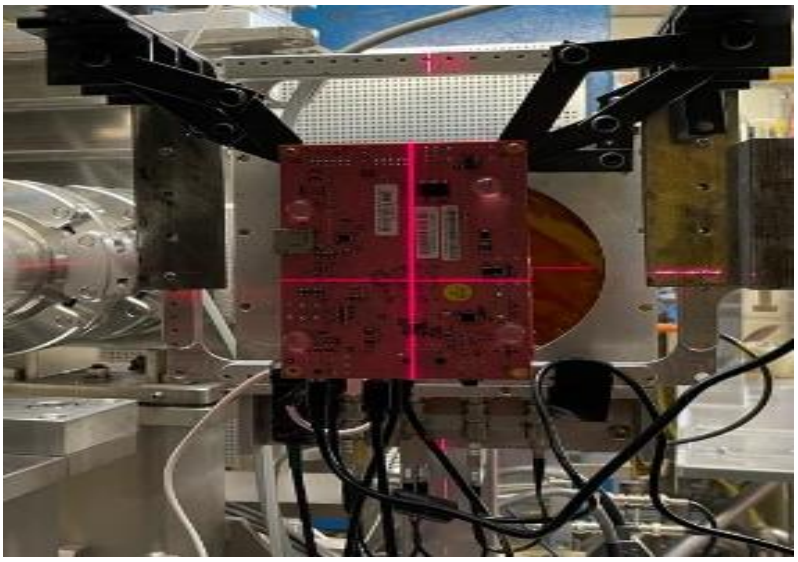

Figure 2. The Board under test is mounted on an adjustable frame facing the proton beam in the irradiation room.

A host computer was placed in the control room. It was connected through a serial connection to the system under test for acquiring the configuration memory content by readback. The monitoring system was autonomously capable of sending a soft reset signal to the device in the irradiation room if the readback system stopped working. In the case that a software reset could not solve the issue, a power switch was available in the control room to perform a manual power cycle if needed.

The experiment was carried on for 8 h in December 2022 at the PSI proton facility. During the experiment, the configuration memory of the device has been continuously monitored through a periodic reading of the content every 5 s. The flux of the particles has been tuned to keep a few bitflips in configuration memory in each snapshot. Table 1 shows the value of energies and fluxes used during the radiation test experiment.

Table 1. Radiation Test Conditions: Energy, Flux, and Fluence.

Energy [MeV]	Flux [$cm^{-2}s^{-1}$]	Fluence [cm^{-2}]
29.31	4.124×10^7	9.173×10^{10}
50.80	4.024×10^7	6.064×10^{10}
69.71	4.110×10^7	2.124×10^{10}
101.34	4.319×10^7	2.415×10^{10}
151.18	4.094×10^7	1.226×10^{10}
200	4.144×10^7	3.942×10^{10}

4.2. Single Event Multiple Upset Fault Model

The snapshots of the configuration memory content have then been analyzed to detect the occurrence of Multiple Bit Upsets (MBUs) between two consecutive readbacks.

The few bitflips and the large size of the configuration memory (more than 10^8 bits) allowed us to detect groups of SEUs with a strong correlation both in time and space. Due to the huge dimension of configuration memory and the possibility to observe a close snapshot of the configuration memory data, it has been possible to evaluate a cluster of bits with a high probability to have occurred as a result of a Single Event Multiple Upsets (SEMUs). The most recurring cluster shapes are depicted in Figure 3 for the different cluster sizes. Two bitflips are defined as close when their Euclidean distance in frame-bit coordinated in the configuration memory is less than $\sqrt{2}$ [22].

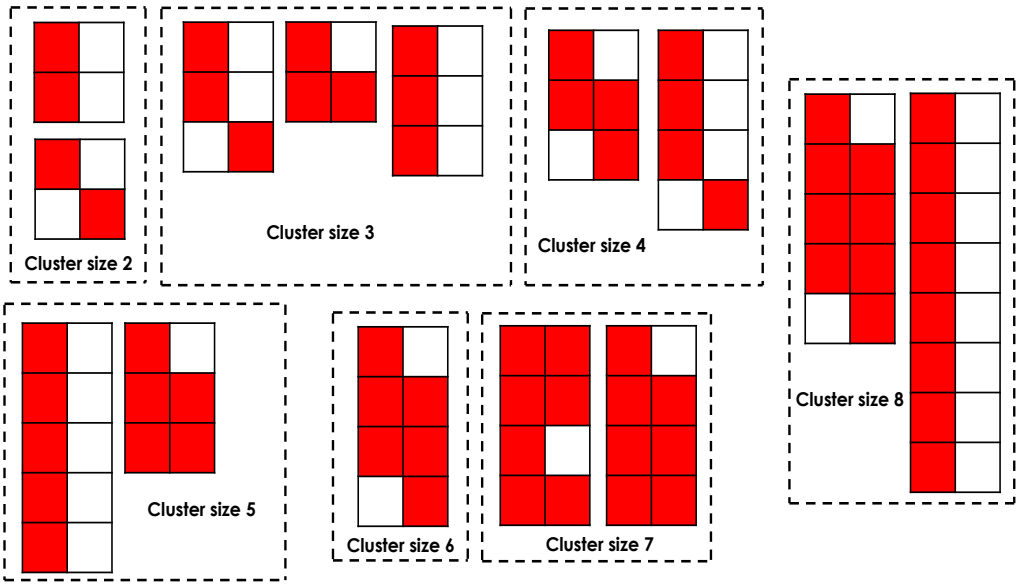

Figure 3. Detected cluster sizes and shapes during the Zynq-7020 proton test.

As was expected, not all the MBUs have been observed with the same frequency. Figure 4 shows the cross-section per particle of each cluster size observed during the whole radiation test. The cross-section is calculated as the number of clusters of a certain size divided by the number of particles passed through the device across all runs for the performed proton test. It can be seen that the contribution of SEMUs is not negligible compared to SEUs. Indeed, more than 40% of the detected events have been SEMUs, which might create a huge impact on the total sensitivity of the system toward radiation-induced errors.

The proposed cluster of faults has been used as the fault model in the fault injection campaign. In order to obtain an analysis as close as possible to reality, the occurrence rate of the different clusters during the fault emulation has been weighted on the cross-section reported in Figure 4. Due to the high occurrence rate of SEMUs, evaluating only an SEU fault model will result in a loose approximation of the observed events.

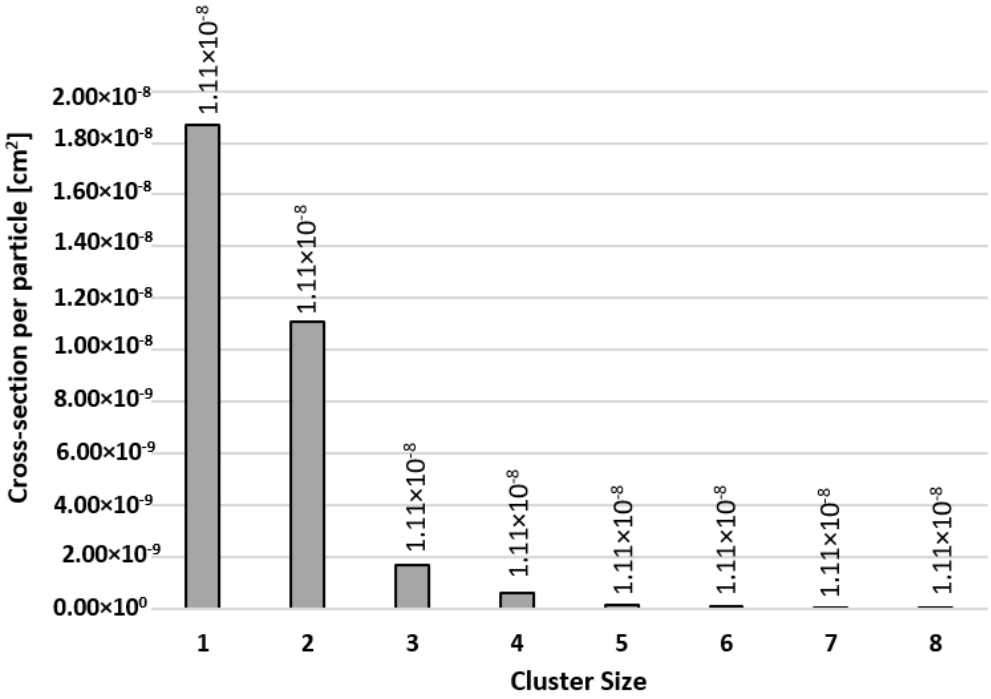

Figure 4. Cross-section of different cluster sizes.

5. Radiation Analysis Workflow

The fault model collected through the proton test has been used to perform an accurate radiation analysis on the impact of radiation-induced faults on the Microblaze-embedded soft processor porting FreeRTOS through fault injection campaigns.

5.1. The Implemented Hardware/Software Platform

The current section elaborates on the implemented hardware platform, Microblaze soft-core, supporting the FreeRTOS and software benchmark applications.

5.1.1. Hardware Platform

The Microblaze embedded soft processor is a Reduced Instruction Set Computer (RISC) optimized for FPGA deployment. It is highly configurable, allowing for the selection of a specific set of features required by the design. Therefore, it has been chosen as the platform for supporting FreeRTOS. FreeRTOS, as a deterministic Real-time operating system, allows concurrency among several tasks with different priority levels, supporting a preemption mechanism to switch between tasks' execution [23].

Another important feature that characterizes the Microblaze porting of FreeRTOS is the possibility to instantiate exception handlers to cope with the standard exception conditions defined by Microblaze soft-core [24].

A Xilinx 28 nm CMOS Zynq-7020 FPGA is chosen as a target hardware device.

Figure 5 represents the implemented hardware while Table 2 reports the device utilization when implementing a Microblaze porting FreeRTOS. As it is can be observed from the table, the implemented design used few resources from the FPGA. Therefore, fault injection campaigns are performed selectively to target only a subset of the whole configuration memory of the FPGA where the circuit is implemented.

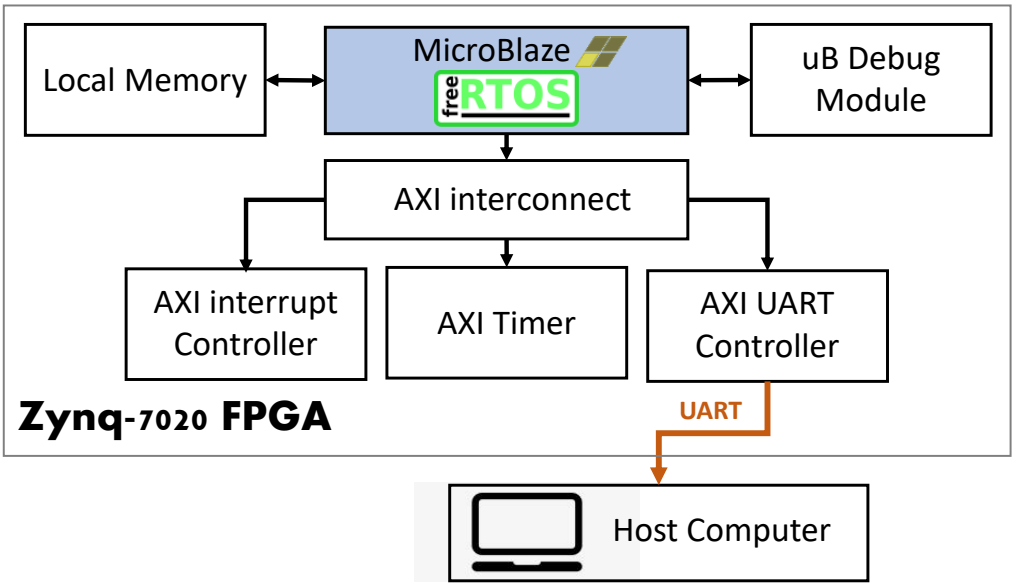

Figure 5. The implemented hardware platform.

Table 2. Resource Utilization of the Hardware Platform.

Resources	Used [#]	Available [#]	Usage [%]
LUTs	2596	53,200	4.88
Logic Slices	966	13,300	7.26
Flip-Flops	2668	106,400	2.51
BRAM	32	140	22.86

5.1.2. Software Platform

As a software application suite, a set of software benchmarks have been chosen to run through FreeRTOS while exploiting its main functionalities. To exploit the capability of FreeRTOS to schedule the execution of different tasks, for each software application, three different tasks (software benchmarks) with the same priority have been instantiated. To elaborate, the three tasks are executing the same application software—thus the same program code—and performing the same calculation, but on different input data, and creating different outcomes. Therefore, three tasks running on FreeRTOS share the same binary code in the instruction memory; however, they are operating on different input data while sharing the processor execution time. The three selected software benchmarks are:

- matmul: matrix multiplication between large matrices of integers.
- matconv: matrix convolution between large matrices of integers.
- dijkstra: computation of shortest path between nodes in a large graph using the Dijkstra algorithm.

5.2. Fault Injection Analysis

Reliability analysis of the applications under test against radiation-induced hardware architectural faults in soft microprocessors has been performed through a fault injection campaign. The PyXEL platform has been used as a supporting fault injection framework [25]. PyXEL is a python-based platform easing the execution of FPGA fault injection campaigns. The platform has been instrumented to inject MBU patterns identified during the proton radiation test into the configuration memory of the FPGA.

The device under test is connected to the host computer running the PyXEL experiment manager through a serial connection, allowing it to run the experiments on the platform and collect results. A timeout mechanism is used to handle the halt or endless loops of the processor due to the injected faults.

The fault injection platform emulates the effects of MBU by changing the configuration memory content. The faulty configuration data are then loaded into the configuration memory via the JTAG interface. The benchmark software uses the serial connection to send the results of the computation. The experiment manager logs the output of each run along with the injected fault model and injected location. Silent Data Corruption is detected by comparison with the expected results, while a Halt is detected via timeout. The rise of exception is notified by the software benchmark to the experiment manager through the serial connection as well. A single experiment on a single board, starting with the fault model and fault location generation and ending with outcome collection and categorization, requires about 7 s. As a reference, a 10,000-fault injection campaign requires about 20 h. However, we sped up the experiments by using more fault injection platforms in parallel.

Two fault injection campaigns have been performed. The former has been carried out based on the distribution of SEUs and MBUs represented in Figure 4. In the latter, each detected cluster has been extensively tested in order to estimate the impact of the different MBU clusters on the application failures.

As we have mentioned previously, during the fault injection campaigns, only part of the whole configuration memory implementing the circuit under test is targeted by the fault injection task in order to reduce the injection space to the resources used by the implemented netlist.

6. Experimental Analysis and Results

This reliability analysis has been carried out by two fault injection campaigns based on fault models collected through a proton radiation test campaign targeting the Zynq-7020 device. Results have been collected, categorized, and discussed.

6.1. Error Classification

As a result of fault injections, different misbehaviors have been observed. Errors are detected by comparison of the outcomes of the fault injection experiments with the outcome of the golden (i.e., without faults) run.

The collected results have been classified into four categories: correct, silent data corruption (SDC), halt, and raising exceptions. They are defined as follows:

- Correct: The FreeRTOS succeeded in executing the application and produced an output that matches the golden one.
- Silent Data Corruption: the task execution on FreeRTOS terminates but the produced output data does not match the golden one.
- Halt: The FreeRTOS does not complete the task. It can be due to different causes, such as infinite loops and application timeout.
- Raising Exceptions: an exception is generated in the FreeRTOS (i.e., at the software level) as a result of a fault affecting Microblaze architecture (i.e., netlist modification due to configuration memory corruption).

Moreover, we have performed a detailed investigation on the cause of each raised exception and classified them as follows:

- FSL_EXCEPTION: data bus error exception.
- UNALIGNED_ACCESS: attempt to perform unsupported unaligned access to memory.
- ILLEGAL_OPCODE: attempt to execute an illegal opcode.
- AXI_D_EXCEPTION: data system bus timeout.

Finally, we have reported the benchmark application error rate which is defined as the percentage of results that deviate from the nominal behavior.

6.2. Experimental Results

We have performed two fault injection campaigns.

The first one is dedicated to evaluating the reliability of the system against the fault models reported in Section 4 observed during radiation testing. It takes into account the clusters' shape, size, and distribution reported in Figures 3 and 4, respectively.

The second one is dedicated to evaluating the criticality and comparison of each type of cluster size to the system's reliability.

6.2.1. First Experimental Campaign

In the first campaign, we performed a total of 10,000 fault injections considering the cluster distribution represented in Figure 4. The observed error rate with respect to the cluster distribution is reported in Figure 6. Data show that the three applications have been impacted differently by the fault injections. In particular, the *matconv* has registered the highest number of total errors (including all four categories) with 2179 corruptions over 10,000 injections while *matmul* and *dijkstra* collected 1028 and 1026 errors, respectively.

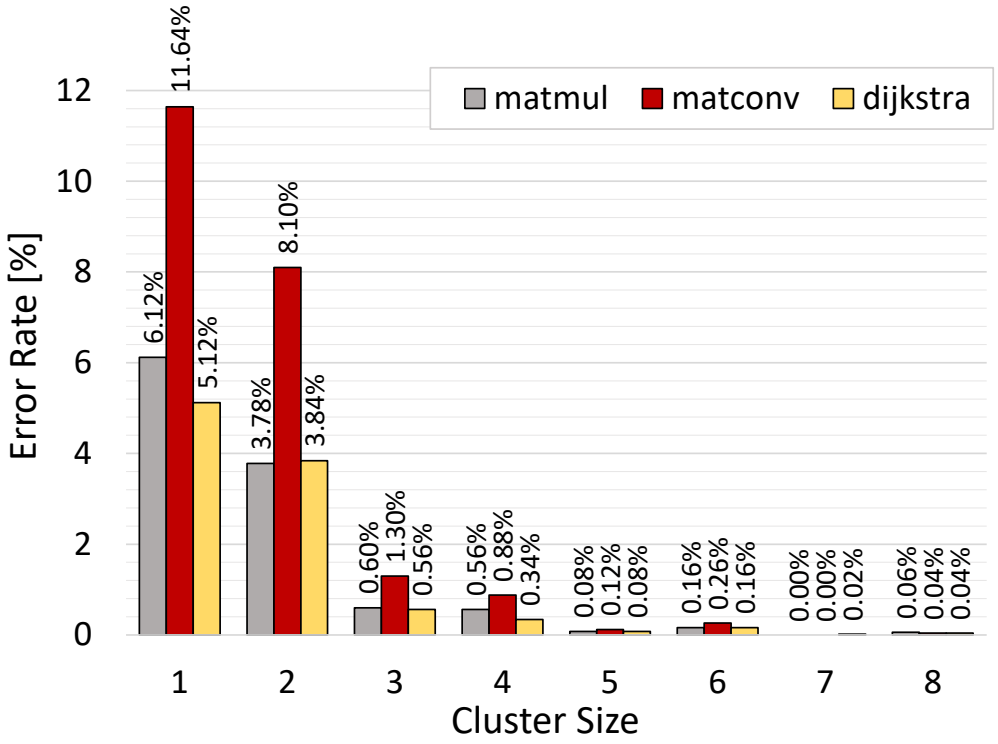

Figure 6. Distribution of errors for each application over cluster sizes.

For each application, the vast majority of the errors are due to the cluster injection of sizes 1 and 2. Indeed, it is easy to observe that the distribution of errors by roughly following the distribution of the clusters. Table 3 reports the classification of the observed errors for each application.

Table 3. Error Rate Classification (# is the number of occurrences).

Application	SDC [#]	Halt [#]	Exception [#]	Total Errors [#]
matmul	227 (19.95%)	882 (77.50%)	29 (2.54%)	1138 (100%)
matconv	1016 (46.63%)	1139 (52.27%)	24 (1.10%)	2179 (100%)
dijkstra	119 (11.58%)	891 (86.67%)	18 (1.75%)	1028 (100%)

It can be noticed that the highest value always belongs to the Halt label, most likely due to the corruption of communication modules within the design.

6.2.2. Second Experimental Campaign

The second campaign is performed considering 5000 fault injections for each cluster size. The results are represented in Figure 7. As can be seen, the cluster size has a marginal effect on the error rate. It may be related to the fact that bits associated with a specific hardware resource are closely located in the configuration memory. If that part of configuration memory is selected as a fault location, the size of the injected cluster will only marginally increase the corruption of the used logic resource.

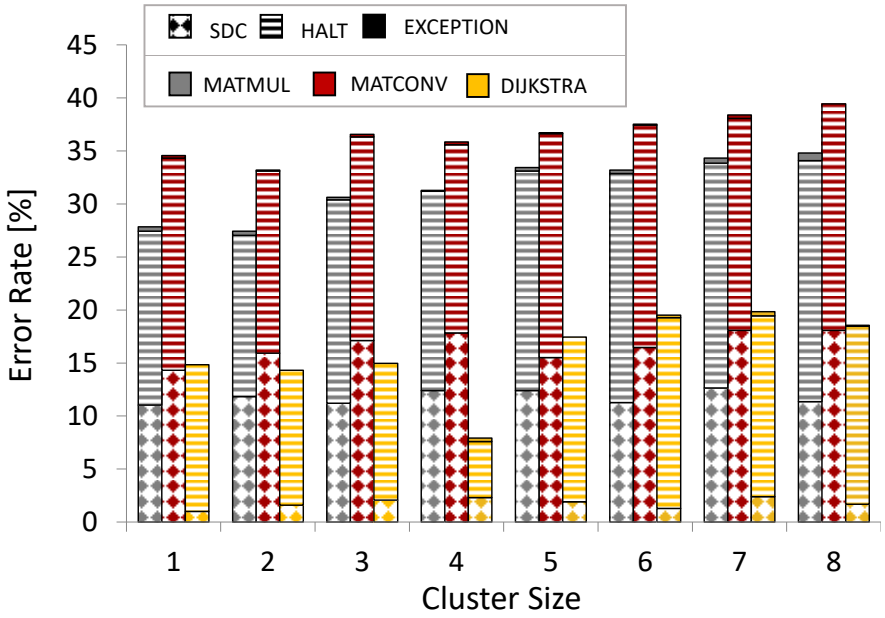

Figure 7. Distribution of errors for each application over cluster sizes.

6.2.3. Software Exceptions induced by Hardware Architectural Faults

A classification of the observed exceptions in the two fault injection campaigns has been performed. The chart in Figure 8 shows the relative frequencies of each exception type.

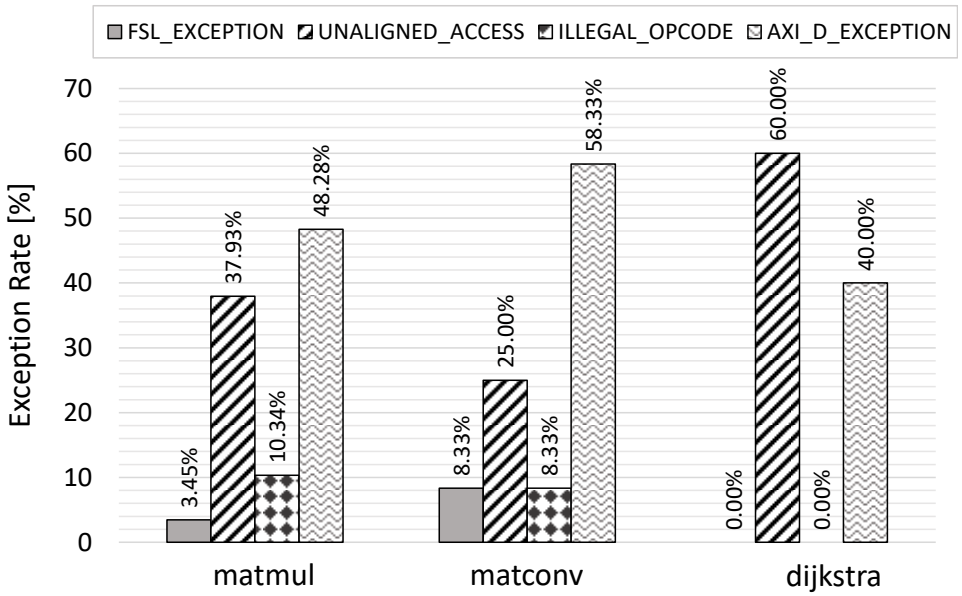

Figure 8. Relative frequency of the exception types resulting from the two fault injection campaigns.

Observed exceptions are all related to reading from the system memory, except in the case of ILLEGAL_OPCODE, which is the less occurring exception.

Although a negligible value with respect to the other errors, the exceptions are a mechanism that can be used to improve the reliability of soft processors and detect the occurrence of a hardware fault.

7. Conclusions and Future Works

In this paper, the impact of radiation-induced architectural faults that have been identified through a proton radiation test have been evaluated, which affected different applications executing on a Microblaze embedded soft-processor implemented on a Zynq-7020 device running FreeRTOS.

The Single Event Multiple Upset fault model resulting from a proton experiment using a range of 200 MeV energies has been presented in detail, with consideration of characteristics such as shape, size, and frequency.

We have performed different fault injection campaigns that consider the identified pattern of MBU fault models, and have investigated the contribution of the proposed fault models to the error rate. Errors have been categorized by typology and observed software exceptions have been categorized further.

In the future, we plan to consider both software and hardware approaches for mitigating radiation-induced errors on applications running within FreeRTOS. For example, we plan to evaluate the impact of selective software redundancy and exploit multiple task features of the FreeRTOS to implement a fault-tolerant system.

Author Contributions: Conceptualization, S.A.; methodology, C.D.S. and S.A.; software, C.D.S. and D.R.; investigation, C.D.S., A.P., D.R. and E.V.; data curation S.A., D.R. and E.V.; writing—original draft preparation, C.D.S., S.A., A.P., D.R. and E.V.; writing—review and editing, L.S. and D.M.C.; visualization, C.D.S. and A.P.; supervision, L.S. and S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors want to thank Wojciech Hajdas and his team at the Proton Irradiation Facility in Villigen, Switzerland for their precious time and support during the radiation experiment.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hofmann, A.; Wansch, R.; Glein, R.; Kollmannthaler, B. An FPGA based on-board processor platform for space application. In Proceedings of the 2012 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), Nuremberg, Germany, 25–28 June 2012; pp. 17–22. [CrossRef]
2. Azimi, S.; De Sio, C.; Rizzieri, D.; Sterpone, L. Analysis of Single Event Effects on Embedded Processor. *Electronics* **2021**, *10*, 3160. [CrossRef]
3. IEEE. *2050–2018-IEEE Standard for a Real-Time Operating System (RTOS) for Small-Scale Embedded Systems*; IEEE: Piscataway, NJ, USA, 2018; pp. 1–333. [CrossRef]
4. Abbas, M.; Ahmed, S. Real-time operating systems for embedded systems: A survey. *Comput. Sci. Inf. Syst.* **2016**, *13*, 1497–1520. [CrossRef]
5. De Sio, C.; Azimi, S.; Portaluri, A.; Sterpone, L. SEU Evaluation of Hardened-by-Replication Software in RISC-V Soft Processor. In Proceedings of the IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), Austin, TX, USA, 21 October 2021; pp. 1–6. [CrossRef]
6. de Oliveira, A.B.; Tambara, L.A.; Benevenuti, F.; Benites, L.A.; Added, N.; Aguiar, V.A.; Medina, N.H.; Silveira, M.A.; Kastensmidt, F.L. Evaluating Soft Core RISC-V Processor in SRAM-Based FPGA under Radiation Effects. *IEEE Trans. Nucl. Sci.* **2020**, *67*, 1503–1510. [CrossRef]

7. Santini, T.; Carro, L.; Wagner, F.R.; Rech, P. Reliability Analysis of Operating Systems for Embedded SoC. In Proceedings of the 2015 15th European Conference on Radiation and Its Effects on Components and Systems (RADECS), Moscow, Russian, 14–18 September 2015; pp. 1–5. [CrossRef]
8. Nekrasov, P.V.; Karakozov, A.B.; Bobrovskyi, D.V.; Marfin, V.A. Investigation of Single Event Functional Interrupts in Microcontroller with PIC17 Architecture. In Proceedings of the 2015 15th European Conference on Radiation and Its Effects on Components and Systems (RADECS), Moscow, Russian, 14–18 September 2015; pp. 1–4.
9. Sterpone, L.; Du, B. Azimi, Radiation-induced single event transients modeling and testing on nanometric flash-based technologies. *Microelectron. Reliab.* **2015**, *55*, 2087–2091. [CrossRef]
10. Azimi, S.; Sterpone, L. Digital Design Techniques for Dependable High Performance Computing. In Proceedings of the 2020 IEEE International Test Conference (ITC), Washington, DC, USA, 1–6 November 2020; pp. 1–10. [CrossRef]
11. Azimi, S.; De Sio, C.; Sterpone, L. A Radiation-Hardened CMOS Full-Adder Based on Layout Selective Transistor Duplication. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2021**, *29*, 1596–1600. [CrossRef]
12. Azimi, S.; Sio, C.D.; Sterpone, L. On the Evaluation of SEEs on Open-Source Embedded Static RAMs. In Proceedings of the 2021 IFIP/IEEE 29th International Conference on Very Large Scale Integration (VLSI-SoC), Singapore, 4–7 October 2021; pp. 1–6. [CrossRef]
13. Mamone, D.; Bosio, A.; Savino, A.; Hamdioui, S.; Rebaudengo, M. On the Analysis of Real-time Operating System Reliability in Embedded Systems. In Proceedings of the IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), Frascati, Italy, 19–21 October 2020; pp. 1–6. [CrossRef]
14. Loskutov, I.O.; Kravchenko, N.D.; Marfin, V.A.; Nekrasov, P.V.; Bobrovsky, D.V.; Smolin, A.A.; Yanenko, A.V. Investigation of Operating System Influence on Single Event Functional Interrupts Using Fault Injection and Hardware Error Detection in ARM Microcontroller. In Proceedings of the International Siberian Conference on Control and Communications (SIBCON), Kazan, Russian, 13–15 May 2021; pp. 1–4. [CrossRef]
15. Aviles, P.M.; Lindoso, A.; Belloch, J.A.; Garcia-Valderas, M.; Morilla, Y.; Entrena, L. Radiation Testing of a Multiprocessor Macrosynchronized Lockstep Architecture With FreeRTOS. *IEEE Trans. Nucl. Sci.* **2022**, *69*, 462–469. [CrossRef]
16. Loskutov, I.O.; Nekrasov, P.V.; Shvetsov-Shilovskiy, I.I.; Boychenko, D.V.; Uzhegov, V.M. SEFI cross-section evaluation by fault injection software approach and hardware detection. In Proceedings of the IEEE 30th International Conference on Microelectronics (MIEL), Niš, Serbia, 11–17 October 2017; pp. 251–254. [CrossRef]
17. Aranda, L.A.; Wessman, N.-J.; Santos, L.; Sánchez-Macián, A.; Andersson, J.; Weigand, R.; Maestro, J.A. Analysis of the Critical Bits of a RISC-V Processor Implemented in an SRAM-Based FPGA for Space Applications. *Electronics* **2020**, *9*, 175. [CrossRef]
18. Wilson, A.E.; Wirthlin, M. Neutron Radiation Testing of Fault Tolerant RISC-V Soft Processor on Xilinx SRAM-based FPGAs. In Proceedings of the 2019 IEEE Space Computing Conference (SCC), Pasadena, CA, USA, 30 July–1 August 2019; pp. 25–32. [CrossRef]
19. Mansour, W.; Velazco, R. SEU fault-injection in VHDL-based processors: A case study. *J. Electron. Test. Theory Appl. (JETTA)* **2013**, *29*, 87–94. [CrossRef]
20. Sio, C.D.; Azimi, S.; Sterpone, L.; Codinachs, D.M. Analysis of Proton-induced Single Event Effect in the On-Chip Memory of Embedded Process. In Proceedings of the 2022 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), Austin, Texas, USA, 19–21 October 2022; pp. 1–6. [CrossRef]
21. Cai, C.; Gao, S.; Zhao, P.; Yu, J.; Zhao, K.; Xu, L.; Li, D.; He, Z.; Yang, G.; Liu, T.; et al. SEE Sensitivity Evaluation for Commercial 16 nm SRAM-FPGA. *Electronics* **2019**, *8*, 1531. [CrossRef]
22. FreeRTOS. Xilinx Microblaze Port. Informative Webpage. Available online: https://bit.ly/3r5Y3ph (accessed on 6 April 2022).
23. Du, B.; Sterpone, L.; Azimi, S.; Codinachs, D.M.; Ferlet-Cavrois, V.; Polo, C.B.; Alía, R.G.; Kastriotou, M.; Fernandez-Martínez, P. Ultrahigh Energy Heavy Ion Test Beam on Xilinx Kintex-7 SRAM-Based FPGA. *IEEE Trans. Nucl. Sci.* **2019**, *66*, 1813–1819. [CrossRef]
24. Xilinx. MicroBlaze Processor Reference Guide. UG984 v2021.2, 27 October 2021, pp. 80–89. Available online: https://docs.xilinx.com/v/u/en-US/ug984-vivado-microblaze-ref (accessed on 27 December 2022).
25. Bozzoli, L.; De Sio, C.; Sterpone, L.; Bernardeschi, C. PyXEL: An Integrated Environment for the Analysis of Fault Effects in SRAM-Based FPGA Routing. In Proceedings of the International Symposium on Rapid System Prototyping (RSP), Torino, Italy, 4–5 October 2018; pp. 70–75. [CrossRef]