POLITECNICO DI TORINO
Repository ISTITUZIONALE

Radiation-Induced Errors in the Software Level of Real-Time Soft Processing System

(Article begins on next page)

29 April 2024

Radiation-Induced Errors in the Software Level of Real-Time Soft Processing System

Corrado De Sio, Daniele Rizzieri, Andrea Portaluri, Salvatore G. La Greca, Sarah Azimi

Dipartimento di Automatica e Informatica (DAUIN)
Politecnico di Torino
Turin, Italy
{corrado.desio, daniele.rizzieri, andrea.portaluri, salvatore.lagreca, sarah.azimi}@polito.it

Abstract— **FPGAs' and programmable hardware's high performance and flexibility have made them a reasonable choice for space-oriented applications, although susceptible to soft errors. This paper proposes a comprehensive analysis of the effects of microarchitectural faults on soft processors due to radiations, identifying the hardware sources of errors and how they propagate to software-level.**

Keywords— *Fault Injection, FreeRTOS, FPGA, Microblaze, RTOS, SEUs, Software Exceptions, Soft Processors*

I. Introduction

In the last few years, Field-Programmable Gate Arrays (FPGAs) have been widely adopted in safety-critical applications, such as space applications thanks to their high performance while keeping costs lower than those required by ASIC solutions. However, exposure to ionizing radiation is a source of many permanent and transient faults, such as Single Event Upsets (SEU) that can affect the device [1] [2] . Real-Time Operating Systems (RTOS) attracted much interest as a suitable solution for meeting stringent real-time predictable execution requirements, especially in the safety-critical domain [3-8]. The study is based on an extensive fault injection campaign emulating radiation-induced SEUs in the configuration memory of a Zynq-7020 MP-SoC. The effects of SEUs are evaluated using a suite of software benchmarks to stimulate different real-time operating systems' basic features. The reliability analysis shows how more than 10% of radiation-induced errors that prevent the program from completing can be detected at the software level by exception-handling mechanisms. Additionally, considerations about the hardware origin of the observed errors are provided by an accurate mapping between configuration memory bits, hardware resources, and software errors.

II. Experimental Analysis

A. Hardware Under Test

To comprehensively analyze hardware faults propagating at the software level, we proposed a thorough reliability evaluation of a soft processor system against microarchitectural faults produced by SEUs in the configuration memory of a 28 nm CMOS AMD Xilinx Zynq-7020 programmable hardware. The MicroBlaze soft processor was implemented in the same hardware and its resource utilization is shown in Table I.

B. Software Benchmark Suite

As software applications set, we chose to rely on the Rhealstone benchmark applications suite [9] running on the real-time operating system FreeRTOS, which included *task*

TABLE I
FPGA RESOURCES USED BY MICROBLAZE SOFT PROCESSORS

Resources	Used [#]	Available
LUTs	2,596	53,200
Logic Slices	966	13,300
Flip-Flops	2,668	106,400
BRAM	32	140

switching, *task preempting*, *semaphore operations*, *deadlock breaking*, and *task communication*.

C. Reliability Analysis Methodology

The proposed reliability analysis methodology is based on emulating SEUs in the configuration memory. The fault injection mechanism is performed through the SEM IP core by Xilinx and the reliability analysis was carried out in parallel, using ten boards. Our in-house developed PyXEL tool [10] has been used to identify the section of the configuration memory associated with the MicroBlaze soft processor. Each benchmark application is deployed in one of the processing systems, and the faults are emulated in the same location for each application using one of the available boards, while an experiment manager running on the host computer controls the reliability evaluation campaign.
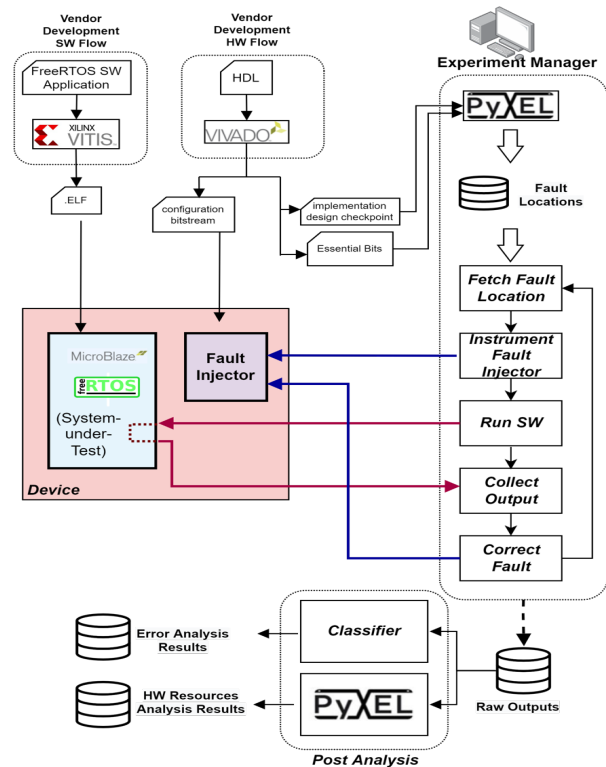

Fig. 1. Fault Injection Campaign Flow.

The experiment manager also controls the SEM and the MicroBlaze by memory-mapped registers and signals, respectively. The fault injection campaign flow consists of various steps represented for a single board in Figure 1. The Essential Bits set is targeted for the corruption, representing a subset of the whole configuration memory that Xilinx identifies as more critical for that specific design and, therefore, to be evaluated for robustness analysis. The raw results are then post-analyzed for detecting and classifying errors by comparison with the golden result generated by a golden run. Additionally, using PyXEL, the fault injection locations that until now have been processed only as configuration memory cells are mapped with the corresponding hardware resource.

III. EXPERIMENTAL RESULTS

A. Error Classification

The results obtained by the fault injection campaign have been categorized based on the errors observed in the software application results. Classification is performed by comparing the output resulting from the system with an emulated fault with the one generated by the system without faults. The outcomes have been listed as:

- *Masked*: the system executed the application correctly even if a fault was injected.

- *Silent Data Corruption (SDC)*: the system executed the application, but the results do not match the expected output.

- *Halt*: the software application does not complete the execution. It can be due to different causes, such as infinite loops or application timeout.

- *Exception*: an exception is generated at the software level due to a fault affecting the Microblaze netlist.

B. Experimental Results and Discussion

The analysis campaigns produced interesting results. Table II ve the remoreports the overall error rates for the analyzed software applications, while Figure 2 reports the error categories distribution for each application. The distribution of the various groups of errors also stands steady among applications confirming hardware faults produced similar outcomes for all the software applications. The distribution of exception types shows how about 50% of them are related to using Advance eXtended Interface (AXI) Bus. The criticality of this module has already been observed by other works [11-12]. Relating fault location and targeted hardware resources with the observed software errors showed additional significant insights. Figure 3 shows the percentage of corrupted locations according to the resource type and Figure 4 reports the rate of software errors for each application associated with the resources targeted.

IV. CONCLUSIONS AND FUTURE WORKS

We presented a comprehensive reliability analysis of a soft processor against SEUs in the configuration memory, analyzing errors propagating from the hardware faults to the software level. The results highlighted the critical role of specific resources in the occurrence of particular software errors, such as the criticality of the programmable interconnections, especially to Halt conditions and Exceptions. Moreover, the criticality of the programmable interconnections should be considered. In particular, custom place and route

solutions are oriented to mitigate errors in interconnection, through net isolation and redundancy are good candidates to improve the robustness of the evaluated soft processor.

TABLE II. OVERALL ERROR RATE FOR SOFTWARE APPLICATIONS

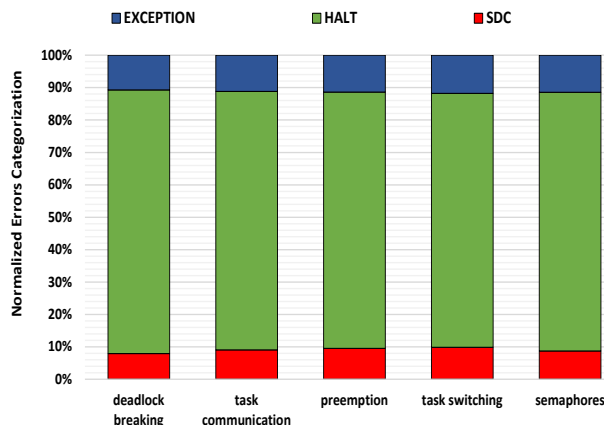| Application	Error Rate	Injected Faults [#]
deadlock breaking	7.11%	50,000
task communication	6.58%	50,000
task preempting	8.03%	50,000
task switching	7.55%	50,000
semaphore operations	7.97%	50,000

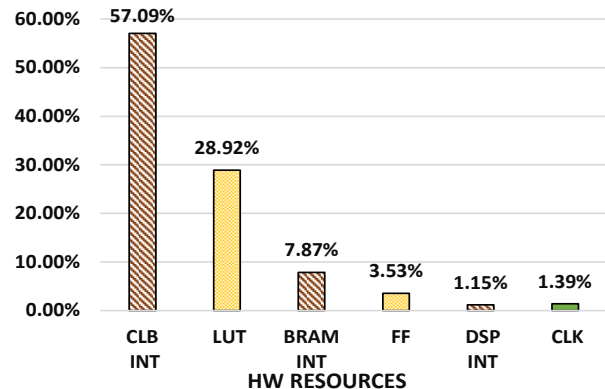Fig. 2. Percentage of the Error Categories within the faulty outcomes.

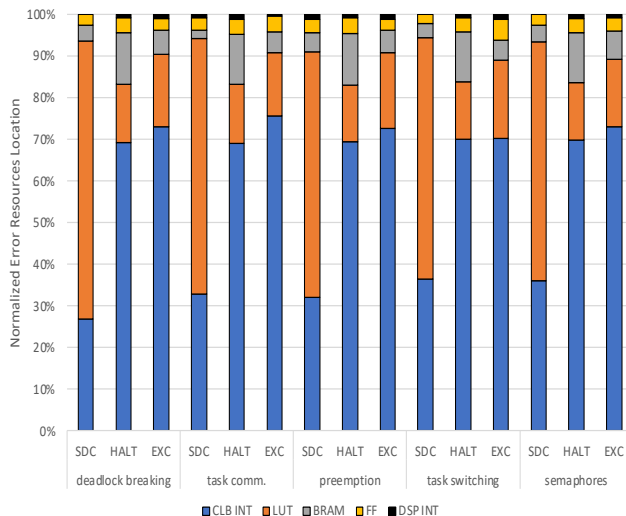Fig. 3. Percentage of the total Fault Injected Locations associated with each HW Resource Type.

Fig. 4. Normalized error rates per resource type over 50,000 injections.

REFERENCES

[1] S. Azimi, C. De Sio, A. Portaluri, D. Rizzieri, L. Sterpone, "A comparative radiation analysis of reconfigurable memory technologies: FinFET versus bulk CMOS," Microelectronics Reliability, Volume 138, 2022, doi.org/10.1016/j.microrel.2022.114733.

[2] H. Quinn, "Radiation effects in reconfigurable FPGAs", in *Semiconductor Science and Technology*, Volume 32, pp 044001, April, 2017. 10.1088/1361-6641/aa57f6.

[3] Á. B. de Oliveira et al., "Evaluating Soft Core RISC-V Processor in SRAM-Based FPGA Under Radiation Effects," in IEEE Transactions on Nuclear Science, vol. 67, no. 7, pp. 1503-1510, July 2020, doi: 10.1109/TNS.2020.2995729.

[4] S. Shukla and K. C. Ray, "A Low-Overhead Reconfigurable RISC-V Quad-Core Processor Architecture for Fault-Tolerant Applications," in IEEE Access, vol. 10, pp. 44136-44146, 2022, doi: 10.1109/ACCESS.2022.3169495.

[5] Portaluri, A., et al. (2022). On the Reliability of Real-Time Operating System on Embedded Soft Processor for Space Applications. In: Schulz, M., Trinitis, C., Papadopoulou, N., Pionteck, T. (eds) Architecture of Computing Systems. ARCS 2022. Lecture Notes in Computer Science, vol 13642. Springer. https://doi.org/10.1007/978-3-031-21867-5_12

[6] 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

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

[8] D. Mamone, A. Bosio, A. Savino, S. Hamdioui and M. Rebaudengo, "On the Analysis of Real-time Operating System Reliability in Embedded Systems," 2020 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), Frascati, Italy, 2020, pp. 1-6.

[9] T. J. Boger, "Rhealstone Benchmarking of FreeRTOS and the Xilinx Zynq Extensible Processing Platform," MS thesis, Dept. Elect. and Com. Eng., Temple Univ., Philadelphia, PA, USA, 2013.

[10] C. De Sio, S. Azimi, L. Sterpone, D. Merodio Codinachs, and Filomena Decuzzi "PyXEL: Exploring Bitstream Analysis to Assess and Enhance the Robustness of Designs on FPGAs," 2023 19th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), Funchal - Madeira Island, Portugal, 2023, pp. 1-4. (Accepted).

[11] C. De Sio, S. Azimi, L. Sterpone, "On the analysis of radiation-induced failures in the AXI interconnect module", Microelectronics Reliability, Volume 114, 2020, 113733, ISSN 0026-2714, https://doi.org/10.1016/j.microrel.2020.113733.

[12] Benevenuti, F., Kastensmidt, F.L. (2018). Analyzing AXI Streaming Interface for Hardware Acceleration in AP-SoC Under Soft Errors. In: Voros, N., Huebner, M., Keramidas, G., Goehringer, D., Antonopoulos, C., Diniz, P. (eds) Applied Reconfigurable Computing. Architectures, Tools, and Applications. ARC 2018. Lecture Notes in Computer Science(), vol 10824. Springer, Cham. https://doi.org/10.1007/978-3-319-78890-6_20