



HAL
open science

Development and evaluation of a flexible instrumentation layer for system-level testing of radiation effects

Israel C Lopes, Vincent Pouget, Frédéric Wrobel, Frédéric Saigné, Antoine Touboul, Ketil Roed

► **To cite this version:**

Israel C Lopes, Vincent Pouget, Frédéric Wrobel, Frédéric Saigné, Antoine Touboul, et al.. Development and evaluation of a flexible instrumentation layer for system-level testing of radiation effects. IEEE Latin American Test Symposium (LATS) 2020, Mar 2020, Maceio, Brazil. pp.1-6, 10.1109/LATS49555.2020.9093681 . hal-03251533

HAL Id: hal-03251533

<https://hal.science/hal-03251533v1>

Submitted on 16 May 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Development and evaluation of a flexible instrumentation layer for system-level testing of radiation effects

Israel C. Lopes¹, Vincent Pouget¹, Frederic Wrobel¹, Frederic Saigne¹, Antoine Touboul¹, Ketil Røed²

1. IES, University of Montpellier, CNRS, Montpellier, France

2. Department of Physics, University of Oslo, Oslo, Norway

Abstract—In the context of the growing interest for system-level testing for radiation effects, this paper presents the development of different levels of software and firmware instrumentation with limited overhead that can be statically or dynamically added to the application under test in order to provide different levels of observability of radiation-induced single-event effects and total ionizing dose effects, as well as to improve the component-level diagnosis of failures observed at system-level. The instrumentation levels are described and applied to a digital control-loop application implemented on a Zynq7000 based commercial system-on-module. A first set of results obtained under high energy protons is presented and discussed.

Keywords—Radiation effects, System-level testing, System-on-chip testing, styling and Fault diagnosis.

I. INTRODUCTION

Particles interacting with integrated circuits may generate different kinds of effects including single-event effects (SEE), i.e. effects induced by a single particle, and total ionizing dose (TID) effects related to the accumulation of the degradation induced by many interactions. The conventional method to predict the reliability of digital systems exposed to radiation environments is a bottom-up approach based on testing individually each component of the system under radiation, then estimating the system's reliability from those test results with significant margins [1][2]. This classical Radiation Hardness Assurance (RHA) process takes time and suffers from the significant costs associated to multiple test campaigns. Such process is hardly compatible with low-cost systems based on components-off-the-shelf (COTS) and limits the integration of new technologies [3][4][5]. In addition, the application of the component level approach is considerably more complex on non-memory devices, such as Field-Programmable Gate-Array (FPGAs), System-on-Chip (SoC) and microprocessors, for which the radiation-induced failure rates can vary significantly according to the system's implementation and usage of the components, making reusability of the component-level results difficult [6].

Those reasons motivate the development of the system-level testing approach, in which a whole system is exposed to radiation and characterized at once. This approach is mostly used for non-critical missions [8], or small [3] and cube satellites [5][7][8][9]. In this work, we consider the particular case of a digital sub-system (typically a single board) including its application firmware and software. By testing the final application embedded in the processing core (FPGA, SoC etc.), or “*testing as you fly*” [7], it is possible to have a more direct obtention of the system functional

failures. Among the limitations of the system-level approach is the lack of detailed observability of the root causes of the system-level failures. Indeed, errors observed at system-level outputs are not easily related to one particular component, which limits the value of the test data for system-level hardening and for extrapolating the test results to the mission environment.

In this work, we present the development and the first evaluation of a set of instrumentation layers that can be added to an application to improve the observability of system-level failures and facilitate their component-level diagnosis. In the first section, radiation effects on digital systems and state-of-the art characterization methodologies are introduced, afterwards the motivation of this work is presented in section II. The target hardware and case study as well as the instrumentation layer are described in section III. The radiation experiment including experimental setup and facility parameters are presented in section IV. Finally, the radiation results are presented and discussed in the section V, and the overall conclusions are done in section VI.

II. MOTIVATION

So far, there is no standard or common practices regarding system level data reporting, rather application-specific testing and reporting are done. In [3], for instance, a system level failure happens when the system generates an incorrect GPS output and non-critical or soft-error occurs when system crash or loss of positioning happens, whereas in [4], a system failure occurs when bugs are reported by a customized operational system or configuration bit upsets occur. This approach makes the component level information gathering more difficult and avoids the reusability of a system level test data.

System level failures can be generalized by classifying them into: permanent loss of functionality, degradation of functionality, hard and soft loss of functionality, data integrity loss and performance degradation. Those classifications are possible when different systems share an error reporting methodology. A flexible instrumentation layer is required so that different applications can use it and tailor it to their requirements. Such instrumentation layer allows the generation of system's fault signature catalogue that can be partially reused by different applications.

III. SYSTEM UNDER TEST

A. Target System-on-Module (SoM)

The target digital system is a commercial industrial-grade SoC-based System-on-Module (SoM), as illustrated in Figure 1. It mainly comprises a 28nm CMOS Zynq7000 SoC [10], a 4Gb DDR3L SDRAM, a 512Mb QSPI FLASH, Ethernet and USB transceivers as well as DCDC converters.

B. Case study Software and Firmware

As a case study, a control loop application was implemented in order to be representative of space and aviation applications. The application was implemented in the Core 1 of the Zynq7000 Processing System (PS), and the Core 0 was used as a self-test controller. The QPSI FLASH was used to emulate sensor data by storing encrypted data and the decryption key. The control loop is based on decryption, filtering, controlling and actuation. The Advanced Encryption Standard (AES) decryption is performed in the PS, then decrypted data are sent to a Finite-Impulse Response (FIR) filter implemented in the Programmable Logic (PL) through the Advanced eXtensible Interconnect (AXI) bus and FIFOs. Afterwards, the FIR output is read by the PS and the Proportional-Integral-Derivative (PID) controlling data is computed and sent to a Pulse Width Modulation (PWM) actuator implemented in the PL through the AXI bus as well. Finally, the PWM output is read by the PS and the loop continues.

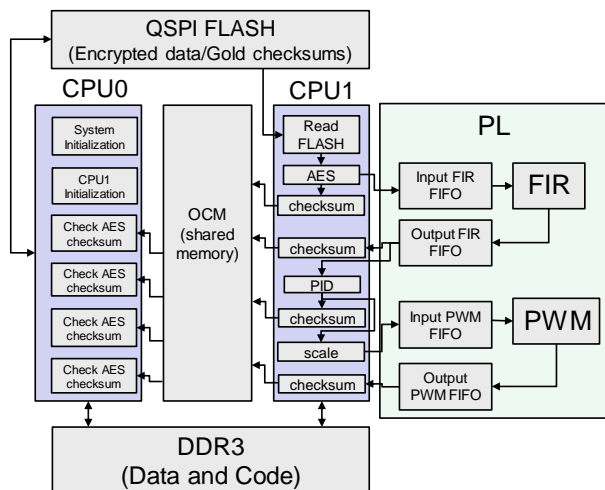


Figure 2. Control-loop application software and firmware structure implemented on the SoM under test.

The AES, FIR, PID and PWM data chunk checksum is computed in the Core 1 and sent to the Core 0 by using the Zynq7000 On-Chip-Memory (OCM) as a shared memory. The Core 0 is responsible for initializing the system, writing the encrypted data, AES key and gold checksums in the FLASH, comparing the checksums and managing the user communication by using the Ethernet protocol. The DDR is used as instruction and data memory for both cores. The overall case study Co-design is illustrated in Figure 2. The AXI bus in the PL operates at 100MHz and the FIR and PWM IP cores operate at 50MHz. A data chunk size of 500 values was used for checksum calculation.

C. Instrumentation levels

The concept of instrumentation in the scope of this work is to add IP or code elements to improve system's

observability and failure diagnosis. The instrumentation was

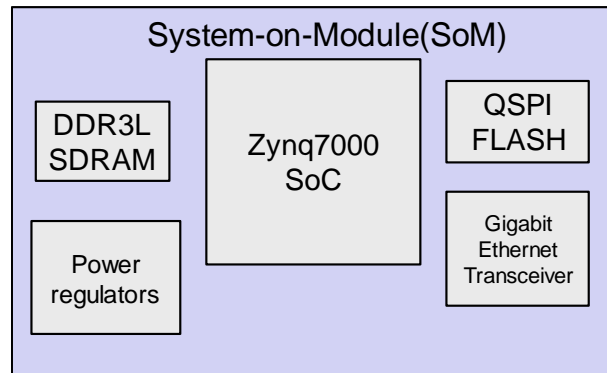


Figure 1. Commercial industrial-grade System on Module under test

divided in different levels that can be easily changed at compiling time. The different Instrumentations Levels (IL) are presented in TABLE I.

TABLE I. INSTRUMENTATION LEVELS

IL	Feature
0	Application output (PWM) checksum
	AXI FIFO and FLASH status
	Watchdog counter
1	AES, FIR, PID checksum
	FLASH and DDR ECC
2	OCM Parity check and FIFO ECC
	APU System Register dump

The idea of the **IL 0** is to have an observability of the application as if it were a black box and to provide minimum application verification. At this level it is possible to identify software crashes and functional failures. In the **IL 1**, the observability of components outside (FLASH and DDR3) the SoC is acquired, as well as checksum of intermediate steps (AES, FIR and PID), being possible to identify error propagation and failing components. Finally, in the **IL 2**, the observability of components inside the SoC (the most complex component in the system) is obtained, such as OCM parity check, FIFO ECC and Application Processing Unit (APU) register dumping. The instrumentation levels are cumulative, it means that the **IL1** contains the **IL0** and so on. The Error Correction Code (ECC) verification process is similar for all the components, the Single-Bit Upset (SBU) are corrected and transparent to the user while Multi-Bit Upsets (MBU) are visible to the application and can trigger a software interruption. Obviously, adding code instrumentation to the application will also add code size and execution time overhead that can have an impact on the system reliability. The code-size, execution time and resource overhead per each IL are presented in TABLE II.

TABLE II. INSTRUMENTATION LEVEL OVERHEAD

IL	Code size (bytes)	Code size overhead (%)	Exec. Time (ms)	Exec. time overhead (%)	Resource overhead
-	3350524	0	639.88	-	-

IL	Code size (bytes)	Code size overhead (%)	Exec. Time (ms)	Exec. time overhead (%)	Resource overhead
0	3352252	0.052	739.71	15.60	CORE0
1	3352884	0.070	796.43	24.47	CORE0
2	3353028	0.075	818.15	27.86	CORE0

IV. TEST METHODOLOGY

A. Facility parameters

The proton experiment was performed by using the AGOR cyclotron of the KVI Advanced Center of Technology (KVI-CART). The beam traverses 3m of air before reaching the System Under Test (SUT) where the energy is lowered from 5 to 10MeV depending on the scatter system. The maximum proton energy was chosen reaching 184MeV at the SUT position and a flux range of $1-3E+06$ p/cm²/s was selected. A 10cmx10cm field collimator was used in which the beam homogeneity is better than $\pm 10\%$ [11].

B. Experimental Setup

During the systems level experiments two SoMs were tested in parallel. The first SoM (SUT #1) was completely irradiated and the second one (SUT #2) was partially irradiated leaving the DDR out of the beam. An illustration of the experiment layout is presented in Figure 3 and a picture of the boards mounted in front of the beam line is



Figure 4. Picture of the SoMs in front of the beam line

shown in Figure 4. The same instrumentation level was applied on both SoMs and two different versions of the software were tested, one with IL 2 enabled and the other one with only the IL 0 enabled in order to see the influence of the overhead and the observability increase. The summary of the irradiations runs is described in TABLE III. The beam time was limited due to limited beam time provided by

the facility, lack of beam controlling (configuration done when the beam was on) and test setup issues. This way, dead time was subtracted from the total fluence to provide the effective fluence.

TABLE III. IRRADIATION RUNS

RUN #	Energy (MeV)	Flux (p/cm ² /s)	#SUT ID	Inst. level	Effective fluence (p/cm ²)
1	184	$1-3.00E+06$	#1	IL2	4.20E+09
			#2		3.78E+09
2			#1	IL0	1.08E+09
			#2		5.40E+08

V. RESULTS AND DISCUSSION

As previously mentioned, two different versions of the Co-design were tested, one embedding the IL0 and another one embedding IL2. Even with the fact that it is possible to get IL0 information in the IL2, it is important to test different ILs to see the impact of the IL overhead on system reliability. The event cross-sections of the application embedding the IL2 are plotted in Figure 5. The application

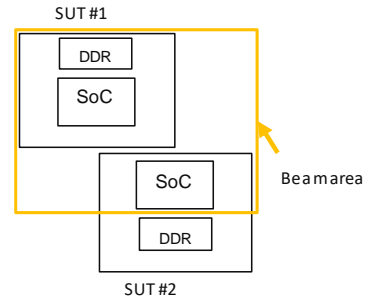


Figure 3. Experiment Layout

hangs (left side of Figure 5) were classified in two different events: application crashes and control flow errors. When no byte is received by the UART communication, it is characterized as an application crash, whereas when the UART keeps receiving application information but the application is looping in an undesirable region of the code, it is characterized as a control flow error. Application crashes can be generated by different components on the system. It can be generated by upsets in the PS cache memories, AXI bus transaction errors, and upsets on the DDR region used to store data and instruction. All of those events will generate data, prefetch or unexpected aborts that will trigger software interruptions that if not handled, will be generically seen as application crashes. Application crashes can also happen due to hard Single-Event Functional Interrupt (SEFIs) on the control logic of the FLASH and DDR memories, as the software will wait indefinitely for those components until jumping to the next instruction. In addition to those possible causes, application crashes can also be generated due to Single Event Latch-up (SEL) and micro SEL on the components of the systems.

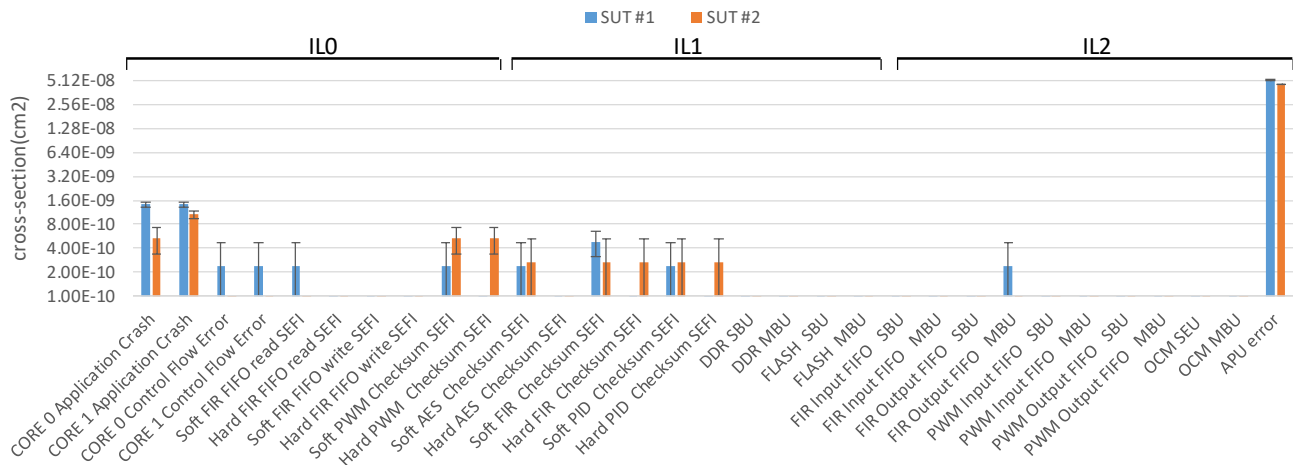


Figure 5. IL2 event cross-sections

On the other hand, control flow errors can be caused by upsets on the DDR region used to store the instructions, instruction caches and PS registers. As it can be seen in the Figure 5, a higher number of applications crashes were observed in the SoM that had the DDR exposed (SUT #1), and no control flow error were observed when the DDR was not exposed (SUT #2), it can indicate that most of the those events were generated by the DDR. Nonetheless, a low number of errors were observed and a higher fluence has to be reached to improve the statistics and make concrete conclusions.

Errors returned by the FLASH (not plotted) and AXI FIFO status checking were classified as hard and soft SEFIs. A soft SEFI happens when the error is corrected in the next iteration while a hard SEFI happens when the SoC has to be reconfigured to correct the error. No FLASH status error was observed and only one Soft FIR FIFO Read SEFI was observed for the SoM with DDR exposed (SUT #1). This error could be generated by an error on the DDR position used to store this status flag, an error on an PL register used to store this status value or an error in the state machine of the AXI FIFO. Additional instrumentation, such as bitstream readback could be used to find the root cause of SEFI, however this approach is so timing consuming and this criticality has to be taken into account.

It also possible to note in the Figure 5 that for some cases more intermediate step (PWM, AES, FIR, PID) checksum

SEFIs were observed for the SoM in which the DDR was not irradiated (SUT #2), however the differences are within the errors bars and a higher number of events has to be observed to improve the statistics. Those errors can be caused by any component on the system that is under SEFI or MBU, as explained before SBU are transparent to the application. This way, DDR, FLASH ECC and OCM Parity Check registers were checked when a checksum mismatch occurred in order to track the component that caused a checksum SEFI or to know when a SBU in a component, which was going to be masked, occurred.

No SBU/MBU were observed on the FLASH, DDR and OCM memories. Regarding FLASH memories, previous work on the literature [12] shown that SBU/MBU are so rare in FLASH memories due to its floating-gate structure, rather SEFI on the control logic are more likely. Low cross-sections are also expected in the OCM, as reported in previous laser fault injections [13] on the Zynq7000 OCM. However, high cross-sections are expected on DDR3 due its Dynamic RAM (DRAM) structure (no feedback) and lower critical charge than Static RAMs (SRAMs), as reported on the literature [14][15]. One possible explanation could be the fact the most of the memory is used for code instead of data, this way, upsets on instructions generate application crashes before the application to be able to report the DDR ECC registers information. It is also possible to note in the Figure 5 (right side), that a pretty high number of errors on the APU registers were observed. Those system registers are responsible for different PS configurations including interruption controller configuration, thus an error on those registers can generate an application crash stopping the execution and preventing it to output DDR SBU/MBUs.

Moving to the IL2 events (Last group of events in Figure 5), only one FIR FIFO Output MBU was detected by the FIFO ECC. It was propagated to the FIR, PID and PWM checksum computations, however many bitstream upsets are expected in the PL according to previous experiments [16] and static-cross-section provided by the manufacturer [17], this can be due to a short software lifetime (time until hang) preventing the application to output those kind of events, as mentioned before. As previously mentioned, errors in the PL could be observed by doing simultaneous bitstream readback with the cost of test time overhead.

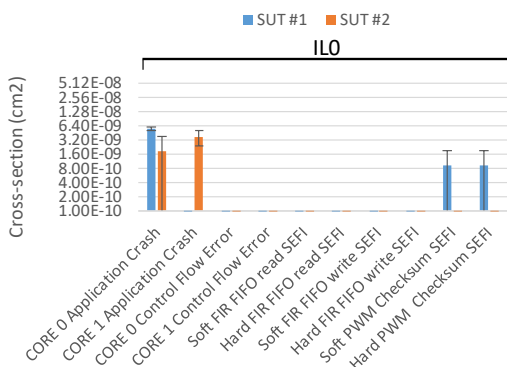


Figure 6. IL0 event cross-sections

Finally, in the Figure 6, the cross-sections of the applications embedding the IL 0 are plotted. In this version of the software it is possible to see that the event cross-sections are even higher than the ones observed in the IL 2, it means that no significant negative impact of IL2 overhead is observed compared to the IL0. A higher number of applications crashes is also observed in this version of the software, confirming that this event has the highest rate. Regarding control flow errors, only one was observed in the non-irradiated DDR SoM (SUT #2), however, as mentioned before, the difference is within the error bars and a higher fluence and error counting has to be provided in order to improve the statistical value of the data. Regarding application checksum SEFIs, only one hard PWM checksum SEFI was observed in the DDR-irradiated SoM (SUT #1), that could be associated to a DDR MBU, however without DDR ECC checking it is not possible to identify the root cause of failure showing the importance of the IL 1 for this kind of analysis.

VI. CONCLUSION

A control loop Co-design embedding an instrumentation layer was developed and irradiated with 184MeV protons. The overhead of the higher instrumentation level did not have an expressive impact on the system reliability. Rather, it allowed the observation of additional events that helped in the first analysis of the root causes of failures and also provided a system's fault signature catalogue. By means of the development of a cross-platform code-instrumentation library, this methodology can be applied to several digital systems in order to normalize the system level test reporting and allow partial reuse of system's fault signature catalogue for different applications and microprocessor architectures.

ACKNOWLEDGMENT

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie-Skolodowska-Curie grant agreement number 721624.

REFERENCES

- [1] Schwank, J. R., Shaneyfelt, M. R., & Dodd, P. E. (2013). Radiation hardness assurance testing of microelectronic devices and integrated circuits: Radiation environments, physical mechanisms, and foundations for hardness assurance. *IEEE Transactions on Nuclear Science*, 60(3), 2074–2100. <https://doi.org/10.1109/TNS.2013.2254722>
- [2] Carvalho, A. Industrial Point of View for Space Applications: Best Practices From Parts Selection To Flight Acronyms & Abbreviations, 1–60. RADECS Short Course, 2017.
- [3] Castillo, G. M., & Ratkevich, B. A. (2011). Single event upset testing of commercial off-the-shelf electronics for launch vehicle applications. *IEEE Aerospace Conference Proceedings*, (January). <https://doi.org/10.1109/AERO.2011.5747254>
- [4] Julien, Connor R., Brock J. LaMeres, and Raymond J. Weber. "An FPGA-based Radiation Tolerant SmallSat Computer System." 2017 IEEE Aerospace Conference. IEEE, 2017.
- [5] Secondo, R., Garcia Alia, R., Peronnard, P., Brugger, M., Masi, A., Danzeca, S., ... Dusseau, L. (2018). System Level Radiation Characterization of a 1U CubeSat Based on CERN Radiation Monitoring Technology. *IEEE Transactions on Nuclear Science*, 65(8), 1694–1699. <https://doi.org/10.1109/TNS.2018.2797319>
- [6] JEDEC. (2006). Measurement and Reporting of Alpha Particle and Terrestrial Cosmic Ray Induced Soft Error in Semiconductor Devices. JEDEC Standard JESD89A, (October), 1–85. Retrieved from <http://www.jedec.org/standards-documents/docs/jesd-89a>
- [7] Guertin, S. M., & Guertin, S. M. (2019). Board Level Proton Testing Book of Knowledge for NASA Electronic Parts and Packaging Program Board Level Proton Testing for NASA Electronic Parts and Packaging Commercial- off-the Shelf Book of Knowledge NASA Electronic Parts and Packaging (NEPP) Program Office of Safety and Mission Assurance.
- [8] Uznanski, S., Alia, R. G., Brugger, M., Moreira, P., & Todd, B. (2000). Qualification of Electronic Components for a Radiation Environment: When Standards do not exist - High Energy Physics.
- [9] Ladbury, R. Strategies for SEE hardness Assurance - From Buy-It-And-Fly-It to Bullet Proof. NSREC Short Course 2017.
- [10] Xilinx Inc. (2018). Zynq-7000 SoC Data Sheet: Overview, 190, 1–21. Retrieved from https://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf
- [11] Van Der Graaf, E. R., Ostendorf, R. W., Van Goethem, M. J., Kiewiet, H. H., Hofstee, M. A., & Brandenburg, S. (2009). AGORFIRM, the AGOR facility for irradiations of materials. *Proceedings of the European Conference on Radiation and Its Effects on Components and Systems, RADECS*, 451–454. <https://doi.org/10.1109/RADECS.2009.5994694>
- [12] Irom, F., Nguyen, D. N., & Allen, G. R. (2013). Single event effect and total ionizing dose results of highly scaled flash memories. *IEEE Radiation Effects Data Workshop*, 13–16. <https://doi.org/10.1109/REDW.2013.6658209>
- [13] Systems, M. R., Rodrigues, G. S., Barros, A., Lopes, I., Pouget, V., Bosio, A., & Fernanda, L. (n.d.). An Approximate Error-Detection Technique for. 1–4.
- [14] Guertin, S. M., & Amrbar, M. (2016). Single event testing of SDRAM, DDR2 and DDR3 memories. *IEEE Radiation Effects Data Workshop*, 0, 1–7. <https://doi.org/10.1109/NSREC.2016.7891742>
- [15] Baumann, R. C. (2005). Radiation-induced soft errors in advanced semiconductor technologies. *IEEE Transactions on Device and Materials Reliability*, 5(3), 305–315. <https://doi.org/10.1109/TDMR.2005.853449>
- [16] Lopes, I. C., Benevenuti, F., Kastensmidt, F. L., Susin, A. A., & Rech, P. (2018). Reliability analysis on case-study traffic sign convolutional neural network on APSoc. 2018 IEEE 19th Latin-American Test Symposium, LATS 2018, 2018-Janua(1), 1–6. <https://doi.org/10.1109/LATW.2018.8347234>
- [17] Xilinx Corporation. (2014). Device Reliability Report. UG116 (V10.1), 116, 1–104. <https://doi.org/Xilinx-ug116>