HAL open science

FLODAM: Cross-Layer Reliability Analysis Flow for Complex Hardware Designs

Angeliki Kritikakou, Olivier Sentieys, Guillaume Hubert, Youri Helen, Jean-Francois Coulon, Patrice Deroux-Dauphin

To cite this version:

Angeliki Kritikakou, Olivier Sentieys, Guillaume Hubert, Youri Helen, Jean-Francois Coulon, et al.. FLODAM: Cross-Layer Reliability Analysis Flow for Complex Hardware Designs. DATE 2022 - 25th IEEE/ACM Design, Automation and Test in Europe, Mar 2022, Antwerp, Belgium. pp.1-6. hal-03485386

HAL Id: hal-03485386

https://hal.science/hal-03485386v1

Submitted on 17 Dec 2021

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.

Distributed under a Creative Commons Attribution 4.0 International License

FLODAM: Cross-Layer Reliability Analysis Flow for Complex Hardware Designs

Angeliki Kritikakou*, Olivier Sentieys*, Guillaume Hubert†, Youri Helen‡
Jean-Francois Coulon§ and Patrice Deroux-Dauphin§
*Univ. Rennes, IRISA, Inria, France, †ONERA, France, ‡DGA, France, § Temento, France

***Abstract*—Modern technologies make hardware designs more and more sensitive to radiation particles and related faults. As a result, analysing the behavior of a system under radiation-induced faults has become an essential part of the system design process. Existing approaches either focus on analysing the radiation impact at the lower hardware design layers, without further propagating any radiation-induced fault to the system execution, or analyse system reliability at higher hardware or application layers, based on fault models that are agnostic of the fabrication technology and the radiation environment. FLODAM combines the benefits of existing approaches by providing a novel cross-layer reliability analysis from the semiconductor layer up to the application layer, able to quantify the risks of faults under a given context, taking into account the environmental conditions, the physical hardware design and the application under study.**

***Index Terms*—Soft Errors, Radiation, Cross-Layer Reliability Analysis, Single/Multiple-Bit Faults, Fabrication Technology.**

I. INTRODUCTION

Critical embedded applications, typically found in avionics, automotive, railway, defense, nuclear and medical domains, have to provide guarantees in terms of operating safety and reliability. However, due to the increased level of chip integration, the reduced transistor sizes and the lower supply voltages of modern technologies [1], [2], the hardware designs are becoming more and more sensitive to environmental sources [3], such as radiation. Radiation-induced faults can be caused by particles from the atmosphere and their impact on the reliability must be carefully assessed [4]. Ionizing particles can create charges in semiconductor and their density depends on the ion species and their energy, characterized by the Linear Energy Transfer (LET) [5]. The transferred energy from particles can (i) corrupt the state of sequential logic, by flipping bits stored in a memory cell or a flip-flop, and (ii) impact the combinational logic, by creating current-voltage transients, known as Single-Event-Transient (SET). The SET is propagated in the forward cone of the impacted combinational cell and it can be eventually latched by sequential logic [6]. As a result, one or several bits are modified leading to Single-Event-Upset (SEU) or Multiple-Event-Upset (MEU). Such errors can jeopardize the system execution, since their appearance in hardware can lead to failures in the application.

Evaluating the behavior of a system under radiation-induced faults is an essential part of the system design process. Two main trends exist, i.e., the first category characterises the induced faults by analysing the radiation impact at the lower hardware design layers, i.e., Technology and Circuit (T&C) layers, while the second category is applied at higher hardware and application layers to characterize the system execution under transient faults (or soft errors). Table I summarizes the State-of-the-Art (SoA) vulnerability approaches, and compare them with regards to the layers and fault models they consider.

TABLE I: Comparison with representative SoA approaches.

Ref.	Layer					Fault model		
	T&C	Gate	RTL	Microarch.	Application	SEU	MEU	SET
[5], [7]	✓	-	-	-	-	✓	✓	✓
[8]	-	-	-	-	✓	✓	✓	-
[9], [10]	-	-	-	✓	✓	✓	-	-
[11], [12]	-	-	-	✓	✓	✓	✓	-
[13]	-	✓	-	-	✓	✓	✓	✓
[14]	-	✓	✓	-	✓	✓	✓	✓
FLODAM	✓	✓	-	✓	✓	✓	✓	✓

Radiation analysis at technology and circuit layers can take into account the circuit layout, the fabrication technology, the radiation and operational environments. For instance, soft errors due to neutron strikes through a SPICE simulator are characterized in [7] and Monte-Carlo simulations are used to compute neutron, proton, heavy ion and α emitter contamination in [5]. Although these approaches accurately characterize the impact of radiation on the physical circuit, they remain at low hardware design layers and do not analyse the propagation of such faults to the system execution.

For reliability analysis at higher layers, a trade-off exists between the accuracy and the time for analysing complex hardware designs. Fault injection at the application level is fast, but less accurate. It is agnostic of the hardware state, fault injection occurs only between instructions and in application variables [8]. To improve accuracy, the underlying hardware should be taken into account. Fault injection approaches at the hardware level typically use random fault models following uniform distributions [12]. The majority of existing approaches focus on single-bit faults in sequential logic, e.g., the fault model is based on a random single bit-flip in the microarchitectural state [9] and a bit-flip on the value of one storage element [10]. Some approaches consider multiple-bit flips in sequential logic, e.g., multiple architectural vulnerability analysis is computed for faults affecting a number of contiguous bits in SRAMs [11], [12]. Last, approaches consider faults occurring both in the sequential and the combinational logic. For instance, random single and multiple faults, occurring to instructions that use arithmetic units, are analysed in [13]. An Instruction-Set Simulator (ISS) executes the application

and invokes a gate-level simulator to inject and propagate the fault. A hybrid fault injection framework combines Register Transfer Level (RTL) and gate-level simulation, injecting an SET of one clock cycle following a uniform probability [14]. Existing approaches characterise the impact of transient faults on the system execution, typically considering random faults and uniform distributions.

Although these two categories are complementary, few works consider their combination in order to accurately analyse the impact of the radiation from the environment on the hardware design and application workload under study. A recent cross-layer approach considers technology, circuit, hardware and application layers based on Bayesian models for single-bit faults in memory components [15]. However, multiple-bit faults and combinational logic faults cannot be neglected and should be included in the reliability analysis.

The FLODAM project[1] addresses this limitation by providing a novel cross-layer reliability analysis from the semiconductor layer up to the application layer. The proposed method and associated flow can quantify the risks of faults under a given context, taking into account the characteristics of the environmental radiation, the physical hardware design and the application. The main contributions of FLODAM cross-layer reliability flow are:

- Realistic modelling of radiation-induced faults at the circuit level, considering the environmental conditions, the types and LET of particles, the fabrication technology and the hardware design.
- Efficient gate-level analysis, based on an approach that combines single-cycle simulation with statistical analysis with high confidence level.
- Fast and accurate microarchitecture-level analysis, based on both Cycle-Accurate-Bit-Accurate (CABA) simulation and FPGA emulation, taking into account the application workload.
- An open-source RISC-V processor is used as the case study for the FLODAM flow. The obtained results describe the fault models at the circuit level, the error patterns at the gate level and the vulnerability metrics at the microarchitecture level.

FLODAM is a national collaborative research project between industry and academia, funded by the French Ministry of Defence (Direction générale de l'Armement - DGA), at its final phase. FLODAM provides to semiconductor manufacturers and embedded system designers a methodology and design automation tools to quantify, in an accurate and fast way, the reliability of hardware designs against transient faults, caused by single radiation particles of natural origin, as defined in the IEC 62396 standard.

The rest of this paper is organized as follows. Section II describes FLODAM cross-layer reliability analysis flow. Section III presents representative results obtained by applying FLODAM flow on the open-source RISC-V processor. Section IV presents the conclusions and the lessons learnt from this project.

[1]https://flodam.fr

II. FLODAM CROSS-LAYER RELIABILITY ANALYSIS

A. Overview

The overview of the cross-layer reliability analysis flow of FLODAM is depicted in Figure 1. The first step to obtain an accurate analysis is to use fault models that reflect the reality of the system's environment, e.g., actual occurring faults on a given hardware design during a flight from Paris to Los Angeles due to single energy particles. To achieve that, we model radiation-induced faults, taking into account both the specific environmental conditions and the characteristics of the specific hardware. To accomplish that, FLODAM cross-layer reliability analysis flow uses a technology and circuit layer simulation tool that characterises the impact of ionizing particles, under different scenarios, on the hardware design under study. The tool is based on a multi-physics approach that is able to analyse the impact of radiation on the physical design. The output is a set of databases that model the distribution of the transient faults at the circuit layer, depending on the cell type, size, inputs and radiation scenario. The transient faults are Single Event Upsets (SEU) and Multiple Event Upsets (MEU) for sequential cells and Single Event Transients (SET) for combinational cells. Such technology- and circuit-layer analysis is required once per fabrication technology and radiation scenario.

The second step is the gate-level analysis, based on statistical fault injection through a single-cycle simulator, in order to characterise with significant statistical confidence the propagation of radiation-induced faults (modelled at the circuit level during the first step). This simulation is able to analyse logical masking and latching window masking, since the hardware design frequency, the area, the delay, the type of cells and the netlist of the hardware design are taken into account. The output is a set of databases describing single-bit and multi-bit error patterns that avoided masking and finally latched in the hardware design registers. Such gate-level analysis is applied once per hardware design.

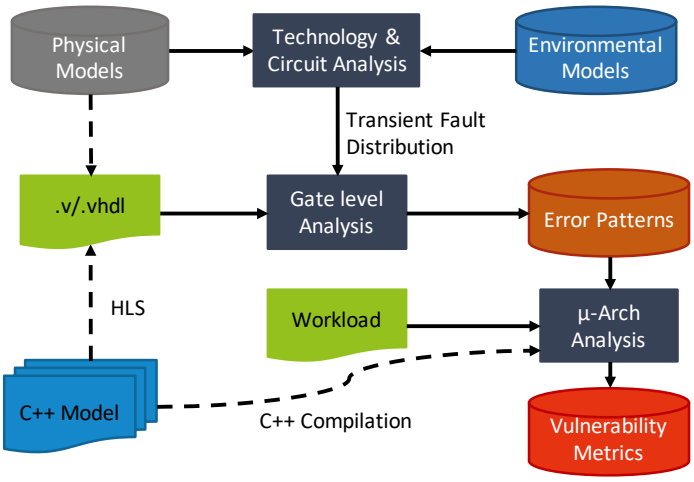


Fig. 1: FLODAM cross-layer reliability analysis flow.

The third step analyses the impact of single-bit and multi-bit error patterns, occurring at microarchitecture layer, on the system execution, taking into account the application workload. This is achieved initially through fault injection using a fast CABA simulator executing the application code. To further improve the analysis time at this step, FLODAM develops a fault injection emulation tool running on an FPGA platform. The microarchitecture-level analysis is applied per application workload.

B. Technology and Circuit Analysis

The FLODAM flow leverages MUSCA-SEP3 [5] and AT-MORAD [16] towards the development of the DiaQuant tool in order to analyse the physical impact of radiation to the hardware circuit, therefore considering the environment and the physical circuit of the hardware design. As an output, DiaQuant creates a set of databases of fault models per technology cell, along with their probability to occur depending on the cell type, size and inputs. Figure 2 depicts the main steps of DiaQuant tool.

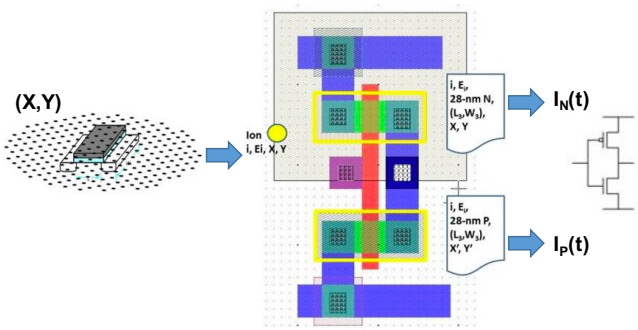


Fig. 2: DiaQuant technology and circuit layer analysis tool.

The environmental models are based on generating environment databases according to the considered missions (e.g., aircraft trajectories, ground trajectories, orbits) and space weather conditions (e.g., solar activity, solar flare). The physical models correspond to the device description, including the circuit layout, fabrication technology, semiconductor active zones, passivation, metallization layers and package. The technology and circuit level analysis is based on Monte-Carlo simulation using a sequential model of the various physical mechanisms occurring when a particle hits a circuit, and potentially leading to a radiation-induced fault. The model describes (i) the particle transport in materials (modifications of the particle primary characteristics via the interaction with the structure, shielding, package and over layers), (ii) the electron-hole pairs and charge generation in semiconductor, (iii) the transport and collection of the charges when electron/holes reach the electrodes, (iv) the transient pulses induced on the different electrodes (drains, sources and taps), and (v) the final effects at the circuit layer. When the circuit is impacted by a particle i of energy E_i, the first step is to identify the cells and transistors potentially impacted. Depending on i, E_i, the size of the transistor and the positioning of the particle with respect to the Drain-Grid-Source topology, the most relevant induced current is identified. After this analysis, DiaQuant selects a current from an $I(t)$ database, depending on the characteristics of the transistor to be impacted (size, type), the particle and its characteristics, to be injected at the circuit layer. Following the aforementioned approach, a set of databases is generated with fault models per cell from the technology library used for the considered design, and for the different particles encountered in atmospheric and space environments.

C. Gate-Level Analysis

As exhaustive fault injection is not possible for complex hardware designs, the aim of the gate-level analysis is to create statistically representative models at this layer for radiation-induced faults occurring at both combinational and sequential cells of the hardware design. To obtain such fault models with statistical confidence and within reasonable time, the gate-level analysis is performed through statistical fault injection per pipeline stage. This is achieved through single-cycle gate-level simulation using the fault models obtained by the technology and circuit analysis. The number of faults N to be injected is defined from the required confidence level in the statistical analysis, i.e., $N = \frac{t^2 \times p \times (1-p)}{e^2}$, where t is the critical value related to the statistical confidence interval, e the error margin, and p the percentage of the possible fault population individuals that are assumed to lead to errors [17].

In order to be able to inject faults, the netlist of the hardware design is extended with an injection block inserted at the output of each cell. With these injection blocks, we can insert SEUs, MEUs and SETs anywhere, at any time and with any duration in the netlist. Note that, in order not to affect the timing characteristics of the hardware design, the propagation delays of all the injection blocks are set to zero. The inputs of the hardware design are randomly generated, e.g., for the execution stage of a processor the inputs are instructions randomly generated from the Instruction Set Architecture (ISA) using random operands. For each such input, a fault-free cycle is executed to obtain the golden reference, i.e., the fault-free output. Based on the technology and circuit layer analysis, the higher the area of a technology cell, the higher its probability to be affected by radiation. Therefore, the selection of the cell, where the fault is injected, it driven by the area of the cell. If the selected cell is of sequential type, the fault is injected directly to the pipeline register. If the selected cell is of combinational type, an SET is inserted into the netlist. The time offset, when the SET is inserted, is randomly chosen within a clock cycle, since radiation may affect the hardware at whatever time instance. The SET duration is given from the databases obtained during the technology and circuit layer analysis. Then, the SET is injected and the output is latched by register at the output of the design stage. If the result in the register is different than the golden reference, the injected fault has led to a single-bit or multiple-bit error. The number and the position of faulty bits are logged, in order to create a set of databases with error patterns.

D. Microarchitecture-Level Analysis

The microarchitecture-level analysis is based on fault injection that is able to evaluate the masking due to the microarchitecture and the application. The fault injection is driven by the error patterns, modelled during the gate-level analysis and describe the single-bit and multiple-bit faults to be injected at the hardware design registers. Prior to any fault injection, we execute the application under study, without faults, in order to obtain a set of golden references: (i) the application output, (ii) the system state (memory and registers), and (iii) the number of cycles required for the execution of the application. Then, the fault injection tool executes the application and injects faults to the hardware design registers, while the application runs. The cycle to inject the faults is chosen randomly between the first cycle and the total number of cycles needed for the fault-free execution, since radiation may impact the system any time. The location, where the faults are injected, is driven by the size of the combinational and sequential logic of the overall hardware design. The larger the area, the higher its probability to be selected. The characteristics of the fault (i.e., how many and which register bits are affected), to be injected in the register of the selected hardware pipeline stage, are provided by the gate-level error patterns. The more times a specific error has appeared, the higher is its probability to be injected, during the microarchitecture analysis. After the fault injection and upon application termination, the results are compared to the golden references, categorising the impact of faults as:

- *Hang (H):* The execution time has exceeded a threshold, and thus, it is assumed that it has entered an infinite loop.
- *Crash (C):* The execution of the application has terminated unexpectedly and an exception has been thrown (out of bound memory access, misaligned PC, hardware trap, etc.)
- *Application Output Mismatch (AOM):* The application output is different than the golden reference.
- *Internal State Mismatch (ISM):* The system state (memory and registers) are different than the golden reference.
- *Functionally Masked (FM):* The application has finished execution, with no AOM and no ISM.

The FLODAM microarchitecture-level analysis is initially implemented by a Cycle-Accurate and Bit-Accurate (CABA) simulator. To further improve the time of fault injection campaigns at this level, a fault injection hardware emulation architecture running on an FPGA has been developed to inject faults at the flip-flops of the hardware design following the patterns issued from gate-level analysis. In this way, fault injection campaigns are performed at the speed of the design on the target FPGA.

III. CASE STUDY: VULNERABILITY ANALYSIS OF A RISC-V PROCESSOR

An open-source 32-bit RISC-V processor [18] is used as a case study, consisting of a standard 5-stage pipeline, including a forwarding mechanism, a hardware multiplier in its execution stage, a Register File (RF) with 32 registers in the write-back stage, and an instruction and data level-one (L1) cache memory. The considered processor is fully

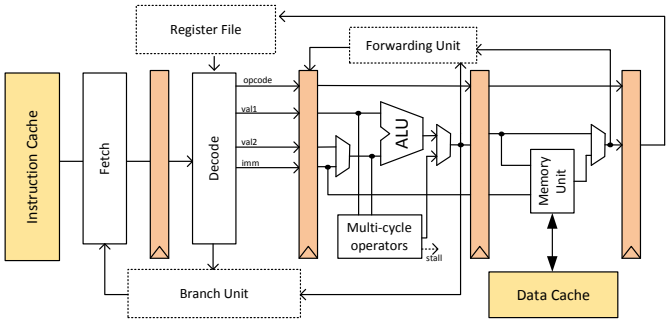


Fig. 3: RISC-V core with 5-stage pipeline [18]

specified using C++ and has been synthesized to the gate-level using Mentor Graphics Catapult High-Level Synthesis and Synopsys Design Compiler with a target frequency of 500 MHz. The target fabrication technology is 28nm FDSOI from ST-Microelectronics using a supply voltage of 1.0V. The sequential logic of the processor (including RF) corresponds to 45.85% of the total area and the combinational logic to 54.15%. Table II depicts the relative area the pipeline stages.

Pipeline stage	Fetch	Decode	Execute	Memory	WriteBack
Area	6.01%	11.02%	35.47%	5.10%	42.41%

TABLE II: Relative area of RISC-V pipeline stages.

A. Technology and Circuit Layer Results

Thanks to the DiaQuant tool, we can obtain fault models for 28nm FDSOI technology cells considering different radiation scenarios. In this use case, we considered several flights departing from Paris to New York, Los Angeles, Johannesburg and Sao Paulo, and vice versa. Each flight is described from a set of realistic flight plans from the Eurocontrol Demand Data Repository (DDR) databases. Each flight plan is characterised by the information regarding the timing (date, time) and the trajectory (latitude, longitude and altitude). Figure 4 represents the set of considered flights. We can observe the diversity of the flight plans for a given flight, e.g., the maximum latitude of a flight from Paris to Los Angeles is between 67° and 45° North, which implies a potential variability in the radiation impacting the aeroplane during the flight.

For each flight plan, the natural radiation environment can be calculated, which describes the differential energy spectrum of neutron, proton and muon for each point of the flight plan trajectory. Analysis takes place considering different particle types, particle energies and technology cells. The transistors potentially impacted when a particle i of energy E_i hits a technology cell and the most relevant induced currents are identified. A current database is created depending on the transistor size and type, the particle and its characteristics. Then, current injection at the circuit level takes place leading to a set databases with fault models characterised with distributions of SET widths in the cell. For instance, Figure 5 summarizes the distribution of the width of the SET pulses created when neutrons hit the 28nm FDSOI cells, normalized to the cell

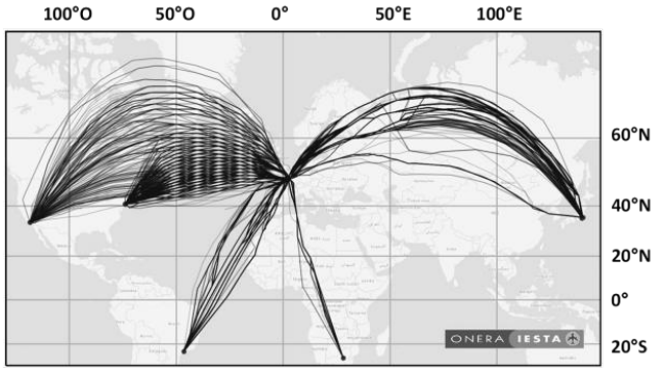


Fig. 4: Flight plans from and to Paris

area and input sizes. In this radiation scenario, we considered an LET equal to $58MeV.cm^2.\mu m^{-1}$, a temperature $25°C$ and an incidence angle of $90°$. Overall, $212,000$ injections took place on 91 logic gates of the fabrication technology and the analysis took XX time. We observe that the majority of SET widths are below $500ps$ (89.457%), while a minority is larger than $1ns$ (0.651%) and than $1.5ns$ (0.13%) for our case study.

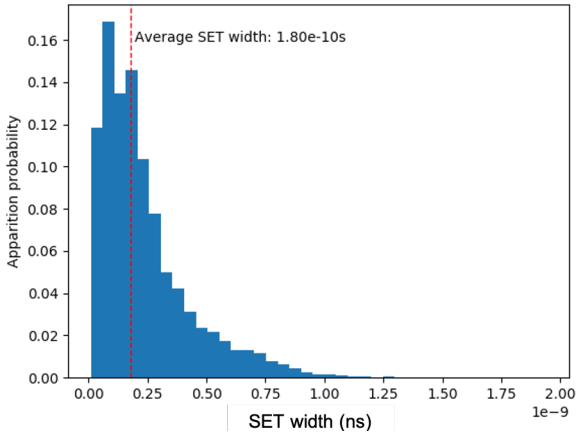
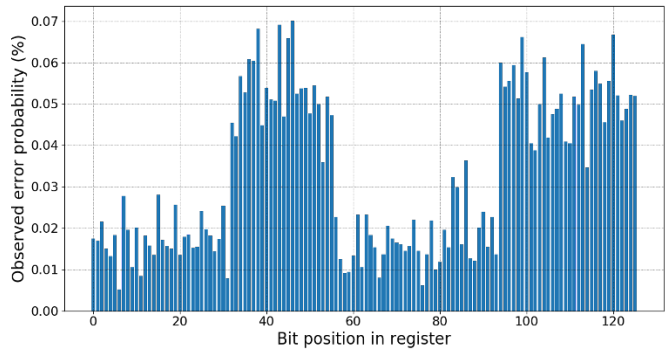


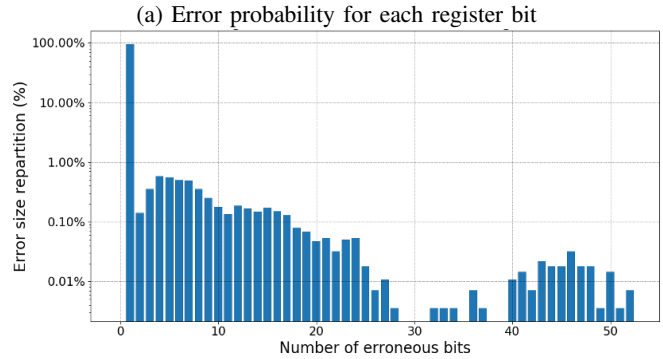
Fig. 5: SET distribution normalized to cell area and input sizes

B. Gate-Level Results

The gate-level analysis provides the error patterns, created due to injected faults at the circuit layer, that managed to avoid logical and timing window masking and latched at the output register. Considering a 99.8% confidence interval with 5% error margin for each input set values, we injected 10^3 for each of the 10^3 different inputs per pipeline stage (10^6 total fault injections per stage). The time required to perform gate-level analysis for the five RISC-V pipeline stages is $1,039s$ with Questa Advanced Simulator 10.7b running on a second-generation Intel Xeon CPU. Figure 6 depicts the gate-level analysis results for the RISC-V execution stage, considering an SET pulse width equal to $400ps$ and an operating frequency of 500MHz. Figure 6a shows the probability of each bit of the output register of the execution stage to be erroneous. Some bits in the register have higher error probabilities, such as the bits corresponding to the ALU output (bit 32 up to bit 63) and data resulting from logical operations on values from Control and Status Registers (CSRs) (bit 94 up to bit 125). Figure 6b shows how many bits were faulty due to a single SET. The higher number of observed erroneous bits for the execution stage is 52 bits, which is 41.3% of the pipeline register size. Experiments for different pulse widths have shown similar results. Regarding SET propagation, for pulse widths $100ps$ and below, the SETs are not often propagated within the latching window of the register, and with a pulse width below $50ps$, no latching of SETs is observed.



(a) Error probability for each register bit



(b) Size of error patterns.

Fig. 6: Gate-level results of the RISC-V execution stage.

C. Microarchitecture-Level Results

Although the error patterns managed to survive at the microarchitecture-level, they may not necessarily lead to a visible error during the system execution. Using a 99.8% confidence interval and a 1% error margin, we injected $23,874$ faults, using the error patterns obtained from the gate-level analysis. The microarchitecture layer output is the distribution of vulnerability metrics. Figure 7 compares the vulnerability results for four benchmarks (matrix multiplication, qsort, strsearch, blowfish), obtained by FLODAM flow and by typical approaches which inject SEUs in the microarchitecture with a uniform fault occurring probability for all registers. Overall, with the FLODAM flow, we observe that less faults are masked, compared to standard microarchitecture-level analysis.

The CABA simulator performance is 18.2 million instructions per second on average, using an 8th generation Intel i7 CPU running at 2.40 GHz. The time required to perform the

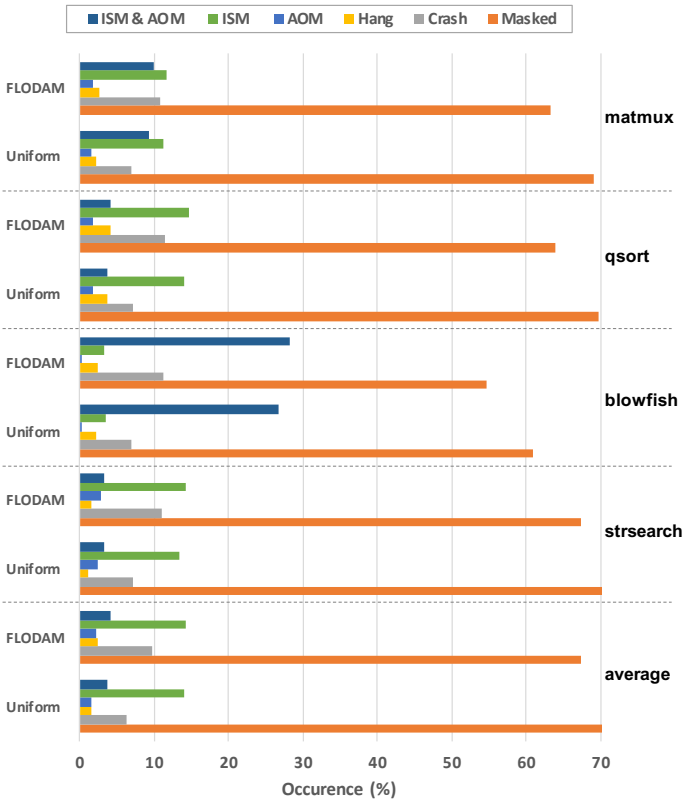


Fig. 7: Vulnerability metrics.

vulnerability analysis at the microarchitecture-level based on the CABA simulator is shown in Table III. The FLODAM fault injection emulation tool, currently without any optimisation, can run at 50 MHz on FPGA Xilinx Zynq XCZ 7045, improving by a factor of 2.5× the time required for the fault injection campaigns. Note that, performing similar analysis using full gate-level methods would require a prohibited time.

Bench.	matmux	qsort	blowfish	strsearch	average
Time	388min 40sec	327min 20sec	466min	427min 20sec	406min 32sec

TABLE III: Time of microarchitecture-level analysis.

IV. Conclusions and Lessons Learnt

FLODAM proposes a cross-layer reliability analysis from the semiconductor layer up to the application layer, able to quantify, in an accurate way, the reliability of hardware designs against soft errors, caused by single radiation particles. The main highlights and insights of FLODAM results are:

- Particles with different characteristics have different impacts when hitting sequential and combinational cells of the design. As a result, the environmental conditions should not be neglected during the reliability analysis of the system in order to obtain accurate results.
- Particles impacting combinational logic can lead to SET of different width pulses. Depending on the characteristics of the particles and the hardware design, SET can be latched in the registers and alter a significant number of bits, leading to a significant number of MEUs.
- The MEUs derived from SETs can be significantly large in size and they not disturb only adjacent bits. Thus, radiation-induced faults should not be modeled only with SEU, but also with (significantly large) MEUs.
- Performing analysis considering the impact of radiation to combinational logic leads to less masked faults at the application layer, compared to typical methods using randomly injected faults based on uniform distribution.
- Combining single-cycle gate-level fault injection and microarchitecture-level fault injection, the reliability analysis is significantly reduced compared to full gate-level fault injection. Further time reduction is obtained by migrating the microarchitecture-level analysis from the CABA simulator to the FLODAM fault injection hardware emulator.

References

[1] E. Ibe, H. Taniguchi, Y. Yahagi *et al.*, "Impact of Scaling on Neutron-Induced Soft Error in SRAMs From a 250 nm to a 22 nm Design Rule," *IEEE Trans. on Electron Devices*, vol. 57, no. 7, pp. 1527–1538, 2010.

[2] A. Dixit and A. Wood, "The impact of new technology on soft error rates," in *Int. Reliability Physics Symp.*, Apr. 2011, pp. 5B.4.1–5B.4.7.

[3] S. Rehman, M. Shafique, and J. Henkel, *Reliable Software for Unreliable Hardware: A Cross Layer Perspective*. Springer Publishing, 2016.

[4] R. Baumann, "Soft errors in advanced computer systems," *IEEE Design Test of Computers*, vol. 22, no. 3, pp. 258–266, 2005.

[5] G. Hubert, S. Duzellier, F. Bezerra *et al.*, "Musca sep3 contributions to investigate the direct ionization proton upset in 65nm technology for space, atmospheric and ground applications," in *Eur. Conf. Radiation & Its Effects on Components and Systems (RADECS)*, 2009, pp. 179–186.

[6] N. Seifert, B. Gill, S. Jahinuzzaman *et al.*, "Soft Error Susceptibilities of 22 nm Tri-Gate Devices," *IEEE Trans. Nuclear Science*, vol. 59, 2012.

[7] M. Riera, R. Canal, J. Abella *et al.*, "A detailed methodology to compute soft error rates in advanced technologies," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2016, pp. 217–222.

[8] B. O. Mutlu, G. Kestor, J. Manzano *et al.*, "Characterization of the impact of soft errors on iterative methods," in *Int. Conf. on High Performance Computing (HiPC)*, 2018, pp. 203–214.

[9] N. J. Wang, A. Mahesri, and S. J. Patel, "Examining ace analysis reliability estimates using fault-injection," in *Int. Symposium on Computer Architecture (ISCA)*, 2007, p. 460–469.

[10] M. Kaliorakis, S. Tselonis, A. Chatzidimitriou *et al.*, "Differential fault injection on microarchitectural simulators," in *IEEE Int. Symp. Workload Characterization (ISWC)*, 2015, pp. 172–182.

[11] M. Wilkening, V. Sridharan, S. Li *et al.*, "Calculating architectural vulnerability factors for spatial multi-bit transient faults," in *Int. Symposium on Microarchitecture*, 2014, pp. 293–305.

[12] N. George, C. Elks, B. Johnson, and J. Lach, "Transient fault models and avf estimation revisited," in *Int. Conf. Dependable Systems and Networks (DSN)*, 08 2010, pp. 477 – 486.

[13] C.-K. Chang, S. Lym, N. Kelly *et al.*, "Hamartia: A fast and accurate error injection framework," in *Int. Conf. on Dependable Systems and Networks Workshops (DSN-W)*, 2018, pp. 101–108.

[14] A. L. Sartor, P. H. Becker, and A. C. Beck, "A fast and accurate hybrid fault injection platform for transient and permanent faults," 2018.

[15] A. Vallero, A. Savino, A. Chatzidimitriou *et al.*, "Syra: Early system reliability analysis for cross-layer soft errors resilience in memory arrays of microprocessor systems," *IEEE Trans. Computers*, vol. 68, no. 5, pp. 765–783, 2019.

[16] P. L. Cavoli, G. Hubert, and J. Busto, "Study of atmospheric muon interactions in si nanoscale devices," vol. 12, no. 12, pp. P12 021–P12 021, dec 2017.

[17] I. Tuzov, D. de Andrés, and J. Ruiz, "Accurate Robustness Assessment of HDL Models Through Iterative Statistical Fault Injection," in *European Dependable Computing Conf. (EDCC)*, Sep. 2018, pp. 1–8.

[18] S. Rokicki, D. Pala, J. Paturel *et al.*, "What You Simulate Is What You Synthesize: Designing a Processor Core from C++ Specifications," in *IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD)*, Nov. 2019.