

## Master Thesis

### Design-Space Exploration of Fault-Tolerant Processors

**Keywords:** processor architecture, fault tolerance, error detection and correction, design space exploration, optimization methods, high-level synthesis, FPGA design, RISC-V

**Laboratory:** INRIA – IRISA, Rennes, France

**Team:** CAIRN <<https://team.inria.fr/cairn>>

**Place:** Rennes or Lannion

**Supervisors:** Olivier Sentieys, Angeliki Kritikakou

**Contact:** [Olivier.Sentieys@inria.fr](mailto:Olivier.Sentieys@inria.fr)

As we approach the limit of CMOS scaling, it becomes increasingly unlikely for a computing device to be fully functional due to various sources of faults. Thus, techniques to maintain efficiency in the presence of faults will be important. Generally applicable techniques, such as replication, come with significant overheads. Developing more efficient techniques will be necessary for computing contexts where reliability is critical and is the topic of this thesis. In this direction, we will explore techniques to protect architectures against faults, which have not only a low overhead in terms of area, performance, and energy, but also a significant impact on improving the resilience of the architecture under consideration. For this we will propose optimization methods for exploring automatically the design space of fault-tolerant architectures that provide architectural solutions under reliability constraints to improve the robustness of processor architectures.

#### Context

**Impact of faults.** With advanced technology nodes and the emergence of new devices pressured by the end of Moore's law, manufacturing problems and process variations strongly influence electrical parameters of circuits and architectures [1], leading to dramatically reduced yield rates [2]. Transient errors caused by particles or radiations will also more and more often occur during execution [3,4], and process variability will prevent predicting chip performance (e.g., frequency, power, leakage) without a self-characterization at run time.

Computing architectures become more and more sensible to the environmental impacts [5], such as ionizing particles and high-energy radiation. Such stimuli trigger violations on the system impacting the normal system functionality and creating faults during its operation [6]. If this has been an issue for a long time for space, this now also a problem for avionics and even at ground level.

**Architectures in critical systems.** The consumer market has shifted towards multicore architectures, since the clock speeds of the single processors could not be further increased due to power consumption and heat dissipation limits [4]. To provide correct system functionality, the reliability of multicore architectures has become a very essential aspect. Several different fault tolerant approaches have been proposed in the literature to

improve the system reliability. However, no general solution can exist to provide the required reliability in low cost for all the problems under study. The promising fault tolerant method is determined by the real faults occurring during execution, the application and the platform of each problem under study.

Moreover, there is a clear trend for mission-critical systems to be increasingly based on Commercial-Off-The-Shelf (COTS) unhardened components (e.g., SRAM FPGAs have a much lower cost than developing rad-hard multicores). These systems will also include more and more processing capabilities while operating in harsh space environments. Moving towards rad-hard by designing multicores implemented in FPGAs is clearly challenging. This requires not only for the processor core to be resilient, but also to take advantage of the redundancy offered by multicores, especially at runtime (both at the task level and at the computing resource level). We also expect reconfigurable hardware and soft accelerators to play an important role in the reliability improvement required by such systems.

**Fault-tolerance.** There exist numerous fault-tolerant techniques in the literature [16] among which we can distinguish two main classes: redundancy techniques that apply duplication (for error detection) or triplication (for error mitigation) in space (hardware redundancy) or time (software redundancy), and error detection and correction techniques that encode, e.g. with error correcting codes (ECC), data and operations to verify the correctness of the dataflow inside the processor. Combinations of both techniques are also possible, e.g., adding redundancy in the computation using variants of ECC to avoid full duplication of the hardware [8].

## Objectives

This Master thesis focuses on fault tolerant processor architectures and has as main goal to design and develop a novel method to explore the design space of the promising set of fault tolerant techniques. We will focus this work on the architecture of one core, the management of multicore being the goal of a PhD thesis that will follow this work.

During previous work, we studied the impact of faults on the basic components of a processor architecture, i.e. memory, pipeline, execution units, registers [15]. This analysis was performed on in-house designed VLIW [14] and RISC-V [17] cores specified at the C/C++ level through high-level synthesis (HLS) and designed with a 28nm FDSOI technology. We not only evaluated the probability of Single-Event-Upset (SEU), but also highlighted the impact of Multiple-Bit Upsets (MBUs) on the vulnerability of the processor.

During this thesis, we will define the set of relevant fault tolerant techniques within our domain and classify and characterize them with respect to the reliability that they can offer and the overhead that they impose on the design (performance, area, energy). In the next step we will focus on defining a novel design space exploration methodology and designing the corresponding tools in order to efficiently explore the different fault tolerance design options. The methodology will be based on pruning methods over the classification and optimizations strategies. The results of the proposed methodology are the set of the most promising fault tolerant approaches under given fault scenarios and platform characteristics that reduce the system cost, while providing reliability and real-time guarantees. The C++ RISC-V processor architecture will be used to perform the evaluation of the proposed methodology. Adding fault tolerance to the core is simply achieved by adding code to its C++ specification and resynthesizing the architecture with HLS. Vulnerability is analysed with the in-house tools developed in our previous work [15].

**Note on the bibliography:** for the bibliography part of this Master thesis, a small set of fault-tolerant techniques (duplication, triplication, some ECCs) will be studied from the literature and the project will consist in the analysis of their impact on reliability (e.g., how much the probability of a crash of the processor is reduced by the technique when the system runs at high flight altitude during a solar event) and their overhead (area, power, clock frequency).

## References

- [1] E. Ibe 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 Elect. Dev.*, 57(7):1527–1538, 2010.
- [2] S. Rehman et al. *Reliable Software for Unreliable Hardware: A Cross Layer Perspective*. Springer, 1st edition, 2016.
- [3] N. Seifert et al. Soft Error Susceptibilities of 22 nm Tri-Gate Devices. *IEEE Trans. on Nuclear Science*, 59:2666–2673, 2012.
- [4] V. Vargas et al. Radiation experiments on a 28 nm single-chip many-core processor and seu error-rate prediction. *IEEE Trans. on Nuclear Science*, 64(1):483–490, 2017.
- [5] D. Gizopoulos, M. Psarakis, S.V. Adve, et al. “Architectures for Online Error Detection and Recovery in Multicore Processors”, in DATE, 2011
- [6] D. P. Siewiorek and P. Narasimhan, “Fault-tolerant architectures for space and avionics applications”, technical report, Carnegie Mellon University, 2008.
- [7] Di Carlo, P. Prinetto, D. Rolfo, and P. Trotta, “A fault injection methodology and infrastructure for fast single event upsets emulation on xilinx sram-based fpgas,” in *IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*. 2014, pp. 159–164.
- [8] S. Jafri, J. Piestrak, S. O. Sentieys, and S. Pillement, “Design of the coarse-grained reconfigurable architecture DART with on-line error detection,” *Microprocessors and Microsystems*, vol. 38, pp. 124–136, Mar. 2014.
- [9] S. Jafri, S. J. Piestrak, O. Sentieys, and S. Pillement, “Design of a fault-tolerant coarse-grained reconfigurable architecture: A case study,” in *Proc. of the 11th IEEE International Symposium on Quality Electronic Design (ISQED 2010)*, (San Diego, CA, USA), p. 6 pages, IEEE, Mar. 2010.
- [10] M. Gatti, “Development and certification of avionics platforms on multi-core processors,” in *Tutorial Mixed-Criticality Systems: Design and Certification Challenges*, ESWeek, (Montreal, Canada), 2013.
- [11] The RISC-V Instruction Set Architecture, <http://riscv.org>, 2016.
- [12] R. Psiakis, A. Kritikakou and O. Sentieys, NEDA: NOP Exploitation with Dependency Awareness for Reliable VLIW Processors, *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, July 3-5, 2017.
- [13] G. Hubert et al. A generic platform for remote accelerated tests and high altitude SEU experiments on advanced ICs: Correlation with MUSCA SEP3 calculations. In *IEEE Int. On-Line Testing Symp.*, pages 180–180, June 2009.
- [14] R. Psiakis, A. Kritikakou, and O. Sentieys. Fine-Grained Hardware Mitigation for Multiple Long-Duration Transients on VLIW Function Units. In *22nd IEEE/ACM Design, Automation and Test in Europe (DATE)*, pages 976–979, March 2019.
- [15] J. Paturel, A. Kritikakou, and O. Sentieys. Fast Cross-Layer Vulnerability Analysis of Complex Hardware Designs. In *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 1–6. IEEE, July 2020.
- [16] A. Kritikakou, R. Psiakis, F. Catthoor, and O. Sentieys. Binary Classification Tree of Rigid Error Detection and Correction Techniques. *ACM Computing Surveys*, 2020.
- [17] S. Rokicki, D. Pala, J. Paturel, and O. Sentieys. What You Simulate Is What You Synthesize: Designing a Processor Core from C++ Specifications. In *38th IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–8, Nov. 2019.