An Overview : Dynamically Reconfigurable Fault Tolerant FPGA Architecture

Basheer Ahmed Chagun Basha*, Sebastien Pillement†
*University of Rennes1, IETR, Rue Christian Pauc, 44306 Nantes, France
†LUNAM University, University of Nantes, IETR, Rue Christian Pauc, 44306 Nantes, France
*chagun-basha.basheerahmed@univ-nantes.fr, †sebastien.pillement@univ-nantes.fr

I. Introduction

Field Programmable Gate Arrays (FPGAs) are employed extensively in many applications to implement adaptable systems that provide high density functionality. However, the challenge that must be tackled during system design is their high susceptibility to radiation induced Single Event Upsets (SEUs), when they are designed for hostile environment such as space. Reprogrammability of the FPGA is a major asset for innovative reconfigurable or adaptable system designs. The use of dynamically reconfigurable resources allows the definition of fault-tolerant autonomous systems. This concept refers to a system that is able to configure its own resources in the presence of faults (transient or permanent) to maintain its functionality. Unfortunately, as these circuits are build around SRAM memory cells, they are particularly sensitive to electromagnetic radiation. Therefore, special care must be taken in the design and implementation to face all these adversaries effects.

There are two general known approaches to ensure higher dependability of a computing system [1], [2]: (i) the fault avoidance aiming at reducing the fault rate; a good example of this approach are rad-hard components; and (ii) the fault toleranceaiming at implementing a system still capable to function correctly, even in the presence of a fault be in a degraded mode; good examples of this approach are the triple modular redundancy (that allows to tolerate any faults of one module) and reconfiguration (that allows to passivate faulty components). This article aims to present the proposed dynamically reconfigurable fault tolerant FPGA architecture.

II. Architecture Definition

A. Radiation Tolerance in Multilayer System

Almost all hardware fault tolerant techniques proposed in the literature (such as Triple Modular Redundancy(TMR), Duplication With Comparison(DWC), etc.) rely on redundant or additional hardware. There are further approaches regarding fault tolerance on software level without the need of further hardware, but with increased runtime [3]. One aspect of providing software based fault tolerance to an embedded system could be to regularly perform a context save of all registers, the memory and the program counter. the latest snap shot can be restored, if an uncorrectable error occurs [4]. An other approach is redundant execution of software. Any command could be executed identically twice or with shifted operants to detect malfunction in the hardware. One approach in this area using redundant execution in combination with validation instruction is published as $software\ implemented\ fault\ tolerance(SWIFT)$ [5].

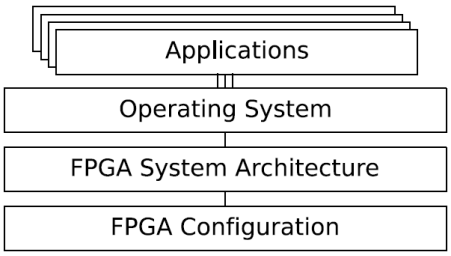

Fig. 1. Layer structure of a FPGA architecture : In order to provide radiation tolerance in such a multilayer system, any layer can give a share by applying layer specific mitigation techniques.

The ultimate aim of this work is to apply fault tolerant techniques to SRAM based reconfigurable architectures not just by blindly triplicating all the logic, but by exploiting mitigation techniques on different layers (refer Fig 1) to keep the area overhead and hardware complexity at a moderate level.

B. A Complete Framework

Indeed, according to the desired application, a radiation hardened SRAM FPGA, a flash based FPGA

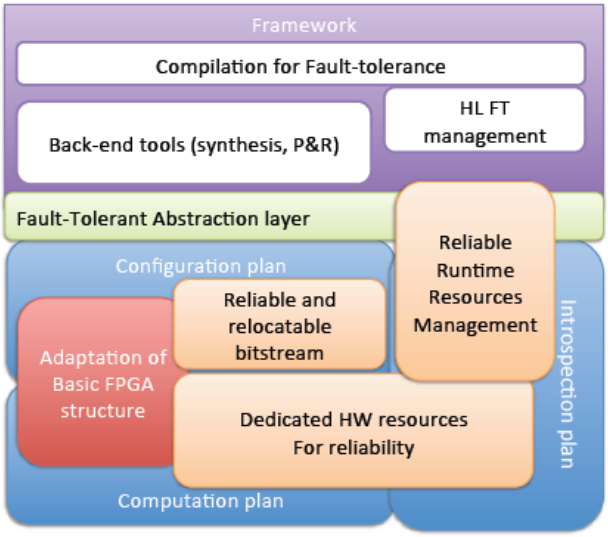

Fig. 2. Dynamically Reconfigurable Fault Tolerant Architecture : A Complete Framwork

or even an antifuse solution may give a significantly better radiation tolerance. The aim of this project is rather to find out a better cost effective SRAM based FPGA architecture. The SRAM FPGA market is continously growing, the flexibility and the possibilities for applications have increased, so why not use them in a radioactive environment, if it is possible with a fair effort?.

As a result of it, a frame work has been developed as shown in Fig 2. In which, a suitable fault mitigation strategy is planned to be deployed at different levels of the architecture from configuration layer to appliaction layer. The purpose of the project is to provide a complete environment for the design of a fault tolerant and self-adaptable reconfigurable platform. It means that the reconfigurable architecture, the associated programming environment and the management methodologies for diagnosis, testability and reliability are all within the scope of the project.

The use of dynamic reconfiguration combined with advanced fault detection and diagnosis techniques will define an adaptive autonomous fault-tolerant architecture (which is self-healing capable to modify its own functioning in the presence of errors). The goal is to define the architecture with built-in basic operators which would enable dynamic management system, including the structure and the efficient distribution of the bitstream. Proposing a new architecture alone, without any supporting CAD tools, would be largely insufficient. Therefore, it is mandatory to set up a complete design framework allowing designers to program and use this architecture.

To decouple development of the architecture and the tools framework, we propose to define the fault tolerant abstraction layer (FTAL). Such a set-up of this layer will facilitate generation of efficient tools using the properties described in the FTAL, whereas the architecture will include dedicated resources to efficiently support the desired properties. The FTAL will then support the newly developed fault-tolerant strategies by defining correct Application Programming Interface (API) methodologies. The tools can then use this API to implement specified algorithms at the compilation level for the uses of the underlying architecture.

This FTAL is a virtual component of the overall flow and will serve as a specification layer for the architecture and an API for the dedicated tools. The modification required in the basic building blocks of the hardware architecture should be specified by the FTAL. All the proposed modifications are intended to improve the reliability of the overall architecture. As an additional feature, an introspection has been planned to adopt in the framework to improve the testability.

III. Conclusion and Prospective Work

We have introduced a framework for dynamically reconfigurable, fault tolerant SRAM based FPGA architecture. The proper sharing of mitigation strategies between different layers of architecture will enable us to have a reliable architectural solution at reasonable cost. The prospective work on this project focuses on functional specification of fault tolerant abstraction layer, which includes, getting the fault detection responces from different hardware modules, runtime resource management, task relocation upon detection of permanent error, partial reconfiguration of faulty hardware module etc., Deploying hardened voters and some dedicated resources for the detection of errors in hardware architecture also being considered.

References

[1] D. K. Pradhan, *Fault-Tolerant Computer System Design.* Prentice-Hall, Upper Saddle River, NJ,, 1996.

[2] D. P. Siewiorek and R. S. Swarz, *Reliable Computer Systems: Design and Evaluation*. Digital Press, Bedford, MA, USA, 2nd edition,, 1992.

[3] H. Engel, "Development of a fault-tolerant softcore cpu for sram based fpgas." Ph.D. dissertation, Kirchhoff Institute for Physics, 2009.

[4] M. Reordq, Violante, "A low-cost see mitigation solution for soft-processcors embedded in systems on programmable chips." in *DATE*, April 2009.

[5] G. A. Reis, "Swift : Software implemented fault-tolerance," 2005.