The Use of Functional Verification for Monitoring Impact of Faults in SRAM-based FPGAs

Jakub Podivinsky, Zdenek Kotasek
Brno University of Technology, Faculty of Information Technology, Centre of Excellence IT4Innovations
Bozetechova 2, 612 66 Brno, Czech Republic

{ipodivinsky, kotasek}@fit.vutbr.cz

Abstract

Digital systems play an important role in our lives. They are used in industry, medicine and other safety critical sectors. Not only the loss of a huge amount of money, but also the loss of human lives may occur in case of their failure. The current trend is that the complexity rises, which leads to an increased susceptibility to faults. The approach called *Fault tolerance* is the ability of a system to continue performing its correct function even in the presence of unexpected faults. There have been many fault-tolerant methodologies inclined, among others, to *Field Programmable Gate Arrays* (FPGAs) developed and new ones are under investigation, because FPGAs are becoming more popular due to their flexibility and reconfigurability. The second reason why so many techniques are inclined to FPGAs is their sensitivity to faults and ability to be reconfigured in the case of fault occurrence. The configuration of FPGAs is stored as a *bitstream* in SRAM memory. The problem is that FPGAs are quite sensitive to faults caused by charged particles. This particle can induce inversion of a bit in bitstream and this may lead to a change in its behaviour. This event is called *Single Event Upset* (SEU).

The systems implemented as fault-tolerant often consist of two parts - an electronic one and a mechanical one. The mechanical part is controlled by its electronic controller. It can be stated that such areas exist in which electro-mechanical applications are implemented as fault-tolerant - aerospace and space applications can serve as an example. We feel that for electro-mechanical systems it must be possible to check the reactions of the mechanical component if the functionality of its electronic controller is corrupted. Based on experiments we found functional verification as an appropriate technique for evaluation impact of faults. Functional verification checks whether a hardware system satisfies a given specification. The main principle of functional verification is to compare the outputs of verified circuits with those of the reference model. Different coverage metrics are defined in order to assess that the design has been adequately exercised. Our experimental platform is composed of a few components running on a computer or on an FPGA evaluation board: 1) software part of verification environment running on a computer, 2) software simulation environment for robot simulation (Player/Stage) running on a computer, 3) robot controller implemented to FPGA, and 4) external fault injector running on a computer.

The process of the fault impact evaluation is divided into three phases. In the *first phase*, we use the simulation-based functional verification where the VHDL description of the electronic robot controller is verified. In this phase, testing whether the robot controller works correctly according to the specification is done. In this phase we acquire a set of verification scenarios (different mazes with different start and goal positions) that will be used in the subsequent phase. The *second phase* consists of the verification of the robot controller implemented into FPGA with the scenarios obtained during the previous phase and uses a previously implemented fault injector. The analysis of the faults which corrupted the mechanical part is the goal of the *third phase*. The outputs are evaluated verification scenarios supplemented by information about injected faults and its impact on the electronical and mechanical part. Various strategies of fault injection may be used in this phase (e.g. one fault for one verification run or multiple faults in the same functional unit).

The outputs of the *first phase* experiments are: 1) the electronic part without bugs (robot controller),

2) the list of the used verification scenarios, and 3) achieved coverage. Performed experiments show us that the 15x15 cells maze is the proper size for the next phase of fault impact evaluation process. For the *second phase* experiments, fault injection is used. No fault tolerance methodology implemented in the robot controller for these experiments was used and the goals of the experiment are: 1) detailed reliability analysis of the robot controller and its functional units, and 2) demonstration that the evaluation platform can be used for the fault tolerance evaluation. We have decided to perform 50 and 100 verification runs and inject one fault into one functional unit (single fault) during one verification run. The robot controller consists of 16 functional units which leads to 1600 evaluation runs. The results of our experiments are shown in Figure 1. The bar chart expresses a percentage number of faults with their impact on the robot (*third phase*) and its controller (*second phase*), the first bar shows the number of electronic failures and the second bar shows the number of collisions of mechanical part for 50 verification runs. The last two bars show the same, but for 100 verification runs. The faults that cause a failure of electronic, but do not cause any collision of mechanical part are usually leading to the robot stop, which is more safety situation than collision. As can be seen, some anomalies in the results of the experiments exist, some of functional units are more prone to the faults than others, but the average number of faults with the impact on electronic part is around 60%. This is especially important for the future applications of fault-tolerant methodologies. A system designer obtains the information which blocks need more attention from a reliability point of view. The chart also shows that results ale similar for both 50 and 100 verification runs.

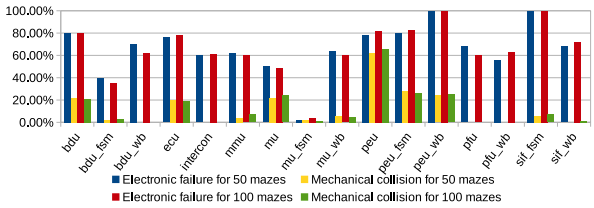

Figure 1: Impact of faults on electro-mechanical system.

The goal of our future work is to apply various fault tolerance methodologies on the robot controller and evaluate them with our evaluation platform. For example, we plan to construct our robot controller as a fault tolerant neural network. We will focus on testing fault tolerance methodologies targeted to FPGAs in the context of electro-mechanical systems and applications which is often the way of using fault-tolerant electronic controllers. As the final result of our reserarch, generally usable principles for testing fault tolerance properties of electromechanical systems will be defined.

Paper origin

The original paper has been accepted and presented at International Conference on Field Programmable Technology (FPT 2016) in China [1].

Acknowledgement

This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II); project IT4Innovations excellence in science - LQ1602, ARTEMIS JU under grant agreement no 621439 (ALMARVI) and BUT project FIT-S-14-2297.

References

[1] J. Podivinsky, O. Cekan, J. Lojda a Z. Kotasek. Functional Verification as a Tool for Monitoring Impact of Faults in SRAM-based FPGAs. In: Proceedings of the 2016 International Conference on Field Programmable Technology. Xi'an: IEEE Computer Society, 2016, s. 289-290. ISBN 978-1-5090-5602-6.