Study and development of fault tolerant operating systems for aerospace applications

Summary

Candidate:
Salvatore Gabriele La Greca
Supervisors:
prof. Luca Sterpone
ing. Daniele Rizzieri
ing. Sarah Azimi

July 2022

In the last few years, the number of missions devoted to universe exploration has increased. Predictions show that the amount of missions in the current decade is expected to be almost three times the amount of missions in the previous one, without considering low-cost and low-weight missions, like the ones including CubeSats. Therefore, the total quantity of electronic devices and the job complexity assigned to them is increasing as well.

Electronic devices must be tailored to work reliably, whatever the purpose of a spacecraft, from the smallest one to a complete rover exploring another planet. Particularly, this concept holds in a complex environment like space, where there are many disturbances such as diverse temperature variations or radiations. The latter is one of the most common causes of failure in spacecrafts and the greatest enemy of electronic components. Thus, a system needs to be as dependable as possible. The dependability of a system is mainly affected by aspects like reliability, availability and safety, especially for space applications.

Nowadays, FPGAs are increasingly being used in aerospace applications due to their flexibility: this is a key aspect of the success of missions because of their high costs, high duration and high complexity. As an example, the Mars Perseverance Rover is almost based on FPGAs. As an example, in the rover's architecture, an FPGA can be found in the automatic entry unit. This unit is responsible for the automatic entry, descent and landing on Mars. Once the rover is landed, it would be useless and would become dead hardware. However, it is based on FPGA hardware so it has been reprogrammed by NASA engineers from Earth to handle computer vision tasks.

Consequently, this thesis aims to study and develop some techniques to mitigate errors induced in soft-cores by "Single Event Upset" faults, which are very common, especially in FPGAs. This area of interest is particularly crucial because complex software, like operating systems, running on top of this hardware, that may be faulty, can create uncoverable and dangerous situations.

Before going deep into the argument, the thesis starts by explaining what an FPGA is, its differences from ASICs and the reason why the space industry is moving towards this technology. Furthermore, Chapter 2 introduces some concepts about radiations on electronic devices, how

are they classified and what effects they can cause on a system. In addition to that, Chapter 3 introduces, with a great level of detail, all the tools and techniques used. The main purpose behind this chapter is to give to the reader a deeper knowledge of the arguments treated in this thesis work and to be able to recreate the proposed solution in the future, in almost a straightforward way.

After that, the reader should be capable of understanding all the concepts that are mentioned in the main chapter of this thesis. Indeed, Chapter 4 analyzes different solutions that have been taken into consideration. The advantages and disadvantages of each solution are also discussed. Without entering into many details, the proposed solution aims to detect faults caused by SEUs in the Xilinx Microblaze CPU by using a custom peripheral. The custom peripheral has been developed in order to be fault-tolerant itself thanks to a Triple Module Redundancy design.

Finally, when a fault is detected, a partial reconfiguration of the FPGA is triggered. This action consists of a partial scrubbing of the configuration memory of the FGPA, in the area where the MicroBlaze physically is. This is achieved by the usage of a partial bitstream, to restore the original behavior. The partial reconfiguration allows for achieving a faster downtime, and consequently a higher availability of the system. This process is entirely managed by the DFX (Dynamic Function Exchange) Controller IP, offered by Xilinx. The DFX Controller loads the configuration file from the memory and sends it to the configuration port of the FPGA.

Moreover, a custom workflow has been developed to allow partial reconfiguration of a MicroBlaze in an older version of Vivado. Additionally, a custom script has been developed based on this workflow, thus providing designers and developers an easy and most automatized way to convert an existing Xilinx design into a design that supports the partial reconfiguration of the Microblaze.

To conclude, Chapter 5 and Chapter 6 are devoted to the analysis of the results coming from this work and the analysis of the possible implications, applications and future work that can be done addressing further research interests.