Microprocessors and Microsystems 39 (2015) 1215–1230

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

The evaluation platform for testing fault-tolerance methodologies in electro-mechanical applications

Jakub Podivinsky *, Ondrej Cekan, Marcela Simkova, Zdenek Kotasek

Brno University of Technology, Faculty of Information Technology, Bozetechova 2, 612 66 Brno, Czech Republic

ARTICLE INFO

Article history:
Available online 30 May 2015

Keywords:
Fault-tolerance
Electro-mechanical systems
Fault injection
Single event upset
Functional verification

ABSTRACT

The aim of this paper is to present a new platform for estimating the fault-tolerance quality of electro-mechanical (EM) systems based on FPGAs. We demonstrate one working example of such an EM system that was evaluated using our platform: the mechanical robot and its electronic controller in an FPGA. Different building blocks of the electronic robot controller allow us to model different effects of faults on the whole mission of the robot (searching a path in a maze). In the experiments, the mechanical robot is simulated in a simulation environment, where the effects of faults artificially injected into its controller can be seen. In this way, it is possible to differentiate between the fault that causes the failure of the system and the fault that only decreases its performance. Further extensions of the platform focus on the interconnection of the platform with the functional verification environment working directly in FPGA that allows for the automation and speed-up for checking the correctness of the system after the injection of faults.

1. Introduction

In several areas, such as aerospace and space applications or automotive safety–critical applications, fault-tolerant electro-mechanical (EM) systems are highly desirable. In these systems, the mechanical part is controlled by its electronic controller. Currently, the trend is to add even more electronics into EM systems. For example, in aerospace, extending of the electronic part results in a lower weight that helps to reduce operating costs [1] [2]. The situation is similar in other sectors, such as automotive sg. [3].

It is obvious that the fault-tolerance methodologies are targeted mainly to the electronic components because they perform the actual computation. However, as the electronics can be realized on different hardware platforms (ASICs, FPGAs, etc.), specific fault-tolerance techniques dedicated for these platforms must be developed.

The previous activities of the team at our department specialized on fault tolerant systems design are described in [4]. In that paper, the fault tolerant methodology for the SRAM based FPGA based on the use of Partial Dynamic Reconfiguration and the Generic Partial Dynamic Reconfiguration Controller inside the FPGA were presented.

The goal of our present research is to develop a platform for the verification of EM systems resilience against faults which occur in an electronic component controlling the system, the component is designed as fault tolerant. Besides from this main activity, the use of functional verification for the automated evaluation of fault impacts is described. The goals are available in details in Section 3.

Our research is targeted to *Field Programmable Gate Arrays* (FPGAs) [5] as they present many advantages from the industrial point of view. They can compute many problems hundreds times faster than modern microprocessors while their reconfigurability allows the same flexibility as microprocessors. FPGAs can be either programmed before their use or reconfigured during program runtime of circuit. Partial dynamic reconfiguration can be also used when programming is performed only on a part of the circuit, while the rest of the circuit is working. The programmability of FPGA differs from Application Specific Integrated Circuit (ASIC) to which the required function was configured in its production cycle. FPGAs are becoming increasingly popular and are used in many applications, mainly due to their programmability, ease of design, flexibility, decreasing power consumption and price. The robot manipulator presented in [6], or the FPGA-based robot arm controller presented in [7], can serve as an example. Moreover, the National Instruments company presents their power train controls which also use FPGAs on their web [8]. They are used mainly in the applications where it is necessary to produce small series and design of ASIC and solution with microprocessor is inappropriate.

* Corresponding author.
E-mail addresses: ipodivinsky@fit.vutbr.cz (J. Podivinsky), icekan@fit.vutbr.cz (O. Cekan), isimkova@fit.vutbr.cz (M. Simkova), kotasek@fit.vutbr.cz (Z. Kotasek).

http://dx.doi.org/10.1016/j.micpro.2015.05.011
0141-9331/

FPGAs can be used advantageously for prototyping complex custom devices. Programmability can also be used to change the behavior of the circuit by a customer which allows to correct errors in design or to add new features to circuit already in use.

FPGAs are composed of *Configurable Logic Blocks* (CLBs) that are interconnected by a programmable interconnection net. Every CLB consists of *Look-Up Table* (LUT) that realizes the logic function, a multiplexer and a flip-flop. The structure of FPGA and CLB is shown in Fig. 1. The configuration of CLBs and of the interconnection net is stored in the SRAM memory. Except CLBs, FPGA contains advanced circuits and other elements, such as *Block Memory* (BRAM), fast multipliers or *Digital Signal Processors* (DSPs). *Input/Output Block*s (IOBs) can be used as the FPGA communication interface.

The problem from the reliability point of view is that FPGAs are quite sensitive to faults caused by charged particles [9]. These particles can induce an inversion of a bit in the configuration SRAM memory of an FPGA (or directly to its internal flip-flops) and this may lead to a change in its behavior. Affecting SRAM or directly the flip-flops can be seen as equivalent in possible consequences. This event is called the *Single Event Upset* (SEU). That is the reason why so many fault-tolerance methodologies inclined to FPGAs have been developed and new ones are under investigation which is mentioned in Section 2.

We decided to use FPGAs in our research mainly because of their speed, re-configurability and because we aim to evaluate various fault-tolerant methodologies dedicated to FPGAs. Despite our exemplary system is not so complex as typical FPGA applications are, it serves for evaluating these methodologies connected to the verification environment very well. All our previous research in the area of fault tolerant systems design was oriented to FPGAs and all our tools were developed for this platform. Therefore, the system presented in this paper has been physically also realized on FPGA mainly for our research purposes and not because it cannot be realized on different platforms as well (for instance, on an ASIC or on a microprocessor).

The paper is organized as follows. The basic concepts connected to the FPGA reliability and verification of hardware systems are summarized in Section 2. The goals of our research and the interconnection scheme of the platform for estimating the quality of EM systems can be found in Section 3. The architecture of our experimental design, the robot controller, is provided in Section 4. A detailed description of the fault injection process that is used for artificial injection of faults into the robot controller is described in Section 5.1. Results of the experiments with the robot controller are available in Section 5.2. The future work that includes using *functional verification* for automated evaluation of impacts of faults and the stimuli generation process is presented in Sections 6 and 7. Section 8 presents another use case – the processor, the reliability of which will be checked in our future work. Finally, the paper is concluded in Section 9.

The research was supported by the following European projects: EU COST Action IC1103 - MEDIAN – "Manufacturable and Dependable Multicore Architectures at Nanoscale" and project IT4Innovations Centre of Excellence (ED1.1.00/02.0070).

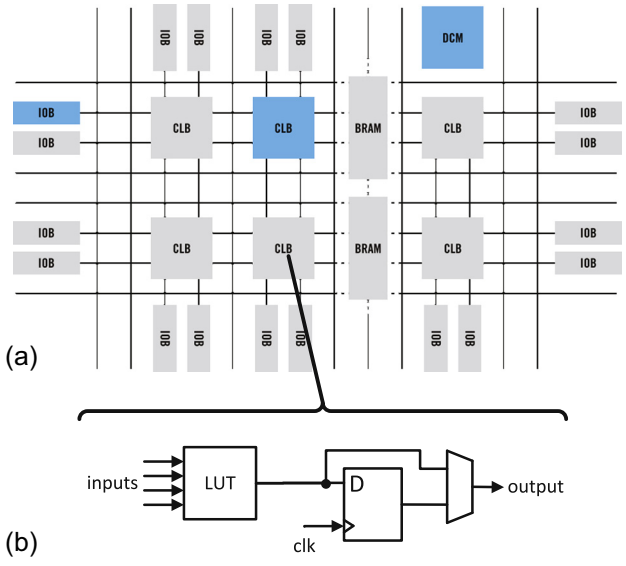

Fig. 1. Structure of (a) *Field Programmable Gate Array* (FPGA) and (b) *Configurable Logic Blocks* (CLB).

2. Related work

Our presented research is unique in a combination of fault-tolerance methodologies and functional verification for improving the reliability of digital systems. For a better understanding, the reader should be familiar with the basic concepts and trends in these two areas. The basic overview is outlined in this section.

2.1. Fault-tolerance methodologies for FPGAs-based systems

Fault-tolerance (FT) is an important feature for many systems, especially for those that aim to be highly reliable. A fault-tolerant system is also able to operate correctly in the presence of faults (SEUs, transient faults, etc.). There are several basic FT architectures that use hardware redundancy such as n-modular redundancy or duplex systems [10]. A special type of n-modular redundancy is *Triple Modular Redundancy* (TMR) which is able to mask a single fault in the system. TMR uses three identical copies of a functional unit (FU) and the unit called Voter. If there is a fault in one FU, Voter chooses the output value using a majority function applied on the primary outputs of the FUs. The TMR architecture is shown in Fig. 2a.

The duplex architecture also provides fault security and is used as the core of many advanced FT architectures. The duplex system can be seen in Fig. 2b. It uses two identical copies of a FU and a comparator (XOR). The output signal *error* informs us about a fault occurrence in the system.

The other type of redundancy, which can be used for hardening against faults, is time redundancy [11]. Time redundancy is based on the repetitive result calculation using the same components but at different time intervals. The obtained results are then compared together. If there are differences, a fault is detected. The scheme of time redundancy is shown in Fig. 3.

The presented hardware redundancy is able to mask a fault occurrence in the FT system. However, the fault localization is needed in order to repair the faulty modules. For these purposes, techniques called *Concurrent Error Detection* (CED) were developed. These techniques encapsulate on-line checkers, self-checking units or parity checkers. A combination of the duplex system with CED that is based on time redundancy is presented in [12]. The duplex system is able to detect a fault occurrence. If a fault is detected, recomputation in the next time slot is able to locate the faulty module. In comparison to the presented TMR architecture, this approach saves some resources. The use of time redundancy as CED leads to less power consumption because the result is recomputed only if a fault is detected. Moreover, this technique reduces the number of input and output pins of the combinational logic.

An important feature of FPGAs, which can be utilized for reliability purposes after a fault (we consider SEUs) is detected, is called *Partial Dynamic Reconfiguration* (PDR) [13]. PDR allows for modifying