High Reliable FPGA Based System Design Methodology

Pavel Kubalík, Hana Kubátová
Department of Computer Science and Engineering
Czech Technical University
Karlovo nam. 13, 121 35 Prague 2
e-mail: xkubalik@fel.cvut.cz, kubatova@fel.cvut.cz

1. Introduction

This paper describes a methodology of the automatic design process for the concurrent error detection (CED) circuits based on FPGAs. Our solution assumes the possibility of dynamical reconfiguration of the faulty part. The most important criterion is the speed of the fault detection and the safety of the whole circuit with respect to the surrounding environment. Our methodology enables cooperation between on-line and off-line BIST for fault detection and localization.

The concurrent error detection (CED) design methodology used to satisfy TSC property has a deep impact on the fault coverage of circuits implemented in FPGAs. Basic methods used for the fault detection in logic circuits are based on simple duplication. This methodology tries to determine the final area overhead before the circuit is duplicated. The duplicate part can be modified to avoid common-mode failures (CMFs). Another approach can be used where the duplicate circuit is modified to decrease the number of outputs of the duplicate part (output parity bits are used instead original outputs). Error detecting codes can be used in this case. Both of these techniques are compared in [4].

There are two main reasons why the CED techniques were not so popular in the past: Very high area overhead and low disposition to temporary faults due to their large feature sizes. Some of the new design methods try to reach smaller area overhead but they achieve low fault detection. For example, only some inputs may be used to ensure the partial self-checking property of a multilevel logic, by using low-cost parity error detecting codes [5].

The next different design methodology ensuring smaller area overhead uses duplication of only some parts of the original circuit. This method is based on the reduction of the number of selected input combinations [2]. Some articles describe methods how to detect the faulty part of an FPGA without stopping its function [1]. These methods test unused parts of the FPGA. When the test is performed, the tested part is exchanged with the used part and the testing process is started again for currently unused area.

2. Used methodology

We have proposed the structure shown in Fig. 1 as a basic model of the totally self-checking (TSC) circuit [3]. In all of our experiments the FPGA platform has been used. The appropriate fault model was discussed [3]. The circuit implemented in an FPGA consists of individual memory elements (LUTs - look up tables). For circuits realized by LUTs the change caused by a single event upset (SEU) leads to an incorrect value on the primary output of the LUT. Therefore we can use the stuck-at fault model in our experiments to detect SEU.

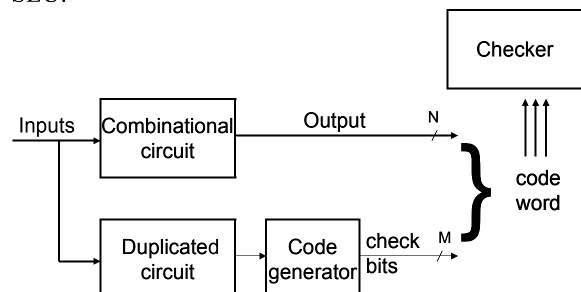

Figure 1. Structure of TSC circuit

Our previous results show, that in many cases is too difficult to reach TSC goals with minimal area overhead [3]. A solution, how to detect and localize fault part of the circuit has to be proposed. If we assume, that the TSC goals cannot be higher than 90%, we can rapidly decrease area overhead and use other methods to cover and localize fault. On-line testing methods can only detect faults. The localization process must exploit some other methods for off-line testing. But both on-line and off-line tests do not increase the reliability parameters. In many cases the reliability is decreasing due to larger area occupied than the original circuit. Therefore we propose the

reconfigurable system in order to increase these parameters. Every block in our design is TSC and we have been working on the methodology to satisfy TSC goals for the whole design and to design highly reliable systems. The solution how to connect all TSC block is shown in Figure 2. The main idea is based on the detection of the error code word generated in any block. The detecting process is moved from primary outputs to primary inputs of the following circuit. The interconnections of all individual blocks play an important role. The different connections between inner blocks can lead to lower fault coverage. Additional logic has to be included to control arrangement of the implemented blocks with respect to the way the automatic tools handle the interconnection.

All our experiments have been applied to the combinational circuits only. The same techniques can be used for sequential circuit due to the fact that these circuits can be divided into simple combinational parts separated by flip-flops. The finite state machine can be divided into two parts: the first part covers combinational logic from inputs to flip-flops (with feedback), the second one covers the combinational logic from flip-flops to outputs (and the parts connected directly from the input to the output). Therefore the restriction to the combinational circuits does not reduce the quality of our method and experimental results.

4. Conclusion and future work

It is possible to reach 100% coverage and TSC goals can be satisfied. However, in these cases the final area overhead of the TSC circuits is higher than 100%. It is too difficult to predict the final area at the time when we select detecting code. We cannot say that the less output nets correspond to smaller area. The area overhead strongly depends on the structure of the tested circuit. Due to the fact, that our methodology is applied to circuits with unknown structure, we must use some steps to reach minimal area overhead and maximum fault coverage.

Our future work is devoted to improving our solution, mainly in choosing the appropriate detecting code. The appropriate cooperation of on-line and off-line testing is under our intensive research, too. We have to discover more precise relations between real FPGA defects and the used fault models. Also the appropriate decomposition of the designed circuit is under our intensive research.

5. Acknowledgement

This research has been in part supported by the GA102/04/2137 grant, CTU0408913 grant and MSM 212300014 research program.

6. References

[1] Abramovici, M., C. Stroud, S. Wijesuriya, C. Hamilton, and V. Verma (1999). “Using Roving STARs for On-Line Testing and Diagnosis of FPGAs in Fault-Tolerant Applications,” Proc. IEEE Intn'l. Test Conf., pp. 973-982.

[2] Drineas, P., Y. Makris (2003). "Concurrent Fault Detection in Random Combinational Logic, In: Proceedings of the IEEE International." Symposium on Quality Electronic Design (ISQED), pp. 425-430.

[3] Kubalík, P. and H. Kubátová (2003). "Design of Self Checking Circuits Based on FPGA." In: Proc. of 15th International Conf. on Microelectronics, pp. 378-381. Cairo, Cairo University.

[4] Mitra, S., and E. J. McCluskey (2000a). "Which Concurrent Error Detection Scheme To Choose?" Proc. International Test Conf., pp. 985-994.

[5] Mohanram, K., E. S. Sogomonyan, M. Gössel, N. A. Touba (2003). "Synthesis of Low-Cost Parity-Based Partially Self-Cheking Circuits," Proceeding of the 9th IEEE International On-Line Testing Symposium, pp. 35.

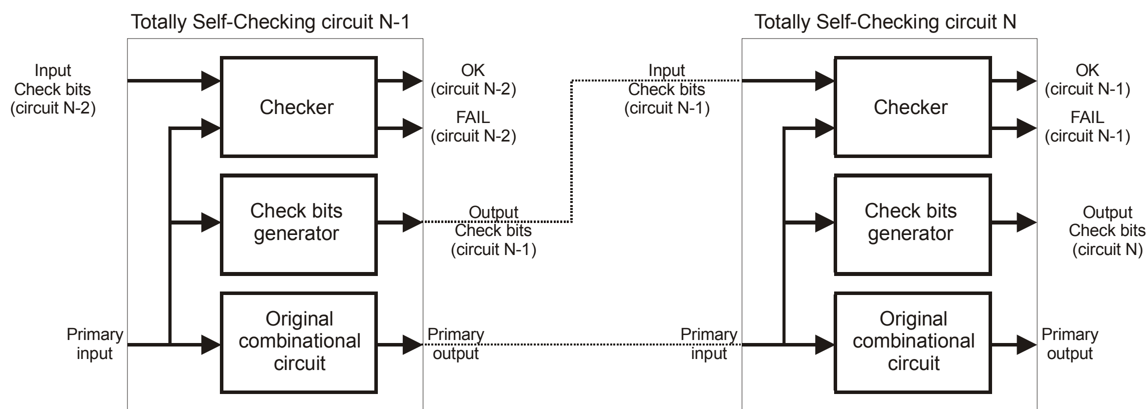

Figure 2. Proposed structure of TSC circuits implemented in FPGA