Microprocessors and Microsystems 52 (2017) 251–260

Contents lists available at ScienceDirect

Microprocessors and Microsystems

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

Parity driven reconfigurable duplex system

Jaroslav Borecký, Martin Kohlík*, Hana Kubátová

Department of Digital Design, Faculty of Information Technology Czech Technical University in Prague, Technická 9, Prague, Czechia

ARTICLE INFO

Article history:
Received 9 January 2017
Revised 16 June 2017
Accepted 19 June 2017
Available online 20 June 2017

Keywords:
Availability
Fault tolerance
Fault tolerant systems
Field Programmable Gate Arrays
Reconfigurable architectures
Reliability

ABSTRACT

This paper proposes a method improving the fault-coverage capabilities of (FPGA) designs. Faults are mostly (SEUs) in the configuration memory of SRAM-based (FPGA) and they can change the functionality of an implemented design. These changes may lead to crucial mistakes and cause damage to people and environment. The proposed method utilizes Concurrent Error Detection techniques and the basic architectures of actual modern (FPGA) – the Look-Up Table (LUT) with two outputs. The main part of the paper is the description of the proposed method (PWtf) based on a cascade – waterfall – of several waves of inner parity generating the final parity of outputs of the whole circuit. The proposed PWtf method utilizes the (mostly) unused output of a two-output LUT to cover any single possible routing or LUT fault with a small area overhead. The encapsulation of the proposed PWtf method into a Duplication with Comparison scheme is presented in the second part of the paper. This encapsulation allows us to create a system containing two independent copies of all parts able to detect and localize any single fault (like common Triple Modular Redundancy method). Experiments are performed on the standard set of IWLS2005 benchmarks in our simulator. The results demonstrate differences between our proposed method and a similar existing technique – Duplication with Comparison (DwC), and between the encapsulated PWtf method and TMR. The proposed method has a lower relative overhead and requires a lower number of inputs and outputs.

1. Introduction

(FPGA) are widely used especially (but not only) due to their flexibility, very good price/performance ratio and rapid design process which shortens and streamlines "time to market". The universality of FPGA chips is based on possible alternations of their configurations. The development is cheaper and faster, because the price of the application based on an FPGA does not include costs of the development of the FPGA chip itself (unlike ASIC). Moreover, the possible reconfiguration without any hardware redesign can be used for either the recovery from a fault state or a less area overhead for pre-designed function changes (the whole functionality should not be implemented on a chip in the same time).

These properties predetermine their use in many areas, even as a control parts in mission critical systems. (FPGA) may occur in many different areas (e.g. aviation, medicine, space missions, and railway applications, etc.) with different impacts to people and environment in a case of their failure. Therefore such systems have to guarantee the determined level of safety and reliability parameters. but there may appear problems, especially when RAM-based (FPGA) are utilized. The main disadvantage is their sensitivity to many effects that can change their programmed function [1]. These changes are most unwelcome in systems, where financial losses, serious injuries or casualties can be caused because of a failure. The improvement of dependability parameters of the final design is required to minimize the impact of such effects.

Dependability of a system is the ability to avoid *service failures* (situations where the behaviour of the system deviates from the correct behaviour) that are more frequent and more severe than is acceptable [2].

One of the most important techniques allowing improvements of dependability is redundancy. This means that if one part of the system fails, there is an alternative functional part. However, redundancy can have a negative impact on a system performance, size, weight, power consumption, and others [3]. There are many redundancy techniques including hardware, information, time, software redundancy, etc. [3]. We focus on hardware redundancy by replication in this paper.

But each type of hardware redundancy means some space (area) overhead, therefore our aim is to find such methods which will minimize this area overhead with the focus to the lowest hardware/structural level when some types of redundancy will be used. It means that our method must depend on the FPGA type.

* Corresponding author.
E-mail addresses: borecjar@fit.cvut.cz (J. Borecký), kohlimar@fit.cvut.cz, martin.kohlik@fit.cvut.cz (M. Kohlík), hana.kubatova@fit.cvut.cz (H. Kubátová).

http://dx.doi.org/10.1016/j.micpro.2017.06.015
0141-9331/

The Fault Tolerant method designed for combinational circuits of newer (FPGA) is proposed in this paper. The PWtf is an architecture-specific and FPGA-specific method – it is based on the architecture of newer (FPGA) (applied and partially tested on the Virtex-5 family) and the technique of (TSC) circuits.

The proposed method has been also encapsulated into a Duplication scheme. This modification – DwPWtf – allows to localize detected faults, and the system may be fully operational during the fault repair (i.e. partial reconfiguration, repair and recovery processes are not part of this paper). This modification allows us to compare DwPWtf method to TMR system directly, because both methods are able to mask a single fault in one part of the design.

Both the plain PWtf and the encapsulated modification DwPWtf are experimentally verified using the standard set of International Workshop on Logic Synthesis IWLS2005 [4] benchmarks. The results demonstrate that the PWtf covers all possible routing and logic faults. The area overhead is smaller than the overhead of the (DwC) in 100% of the tested circuits. The results also demonstrate that the overhead of the DwPWtf is smaller than the overhead of the TMR.

Both methods (PWtf and DwPWtf) have been presented in conference papers. The plain PWtf method and its comparison to a common method (DwC) has been presented in [5]. The encapsulated DwPWtf method and its comparison to a TMR method has been presented in [6]. This paper mainly summarizes both papers and their results. A short overview of the future work has been added – it introduces a new architecture based on a PWtf method encapsulated to an Upgraded Modified Duplex System (UMDS). This modular architecture will utilize a partial reconfiguration that is significantly faster than a reconfiguration of the whole FPGA. The speedup of the reconfiguration should lead to a significant improvement of the Availability parameter.

The paper is organized as follows: Section 2 provides the theoretical background and introduces related methods. The proposed parity waterfall method and its encapsulation is described in Section 3. The results are shown in Section 4 and Section 5 concludes the paper.

2. Background

2.1. Basic primitive element of FPGA

The proposed method utilizes the properties of the basic primitive of modern FPGA chips – the LUT with two outputs shown in Fig. 1. This primitive (LUT6_2) [7] can implement a 6-input logic function or two 5-input logic functions with shared inputs. A logical function of LUT is specified by 64-bit hexadecimal value stored in an INIT attribute. The upper half (bits 63:32) of the INIT values is used for the upper LUT5 and the lower half (bits 31:0) for the lower LUT5. The logic function of the O5 output correspond only the lower LUT5 value, but the O6 output can use both LUT5 values, which depends on a value of the I5 input.

2.2. Self-checking circuit

The main goal of the proposed method is to create a self-checking circuit – a circuit which is able to detect any fault caused by a bit-flip in the configuration memory that may affect it. A recovery from a bit-flip fault can be performed by a reconfiguration process. Transient faults affecting flip-flops containing data are detected as well and can be recovered by a recovery method.

The self-checking circuit is mostly based on a predictor of some kind of error detection code (i.e. parity predictor, a copy of the original circuit, etc.). The outputs of this predictor – the check bits – are connected (together with the outputs of the original circuit)

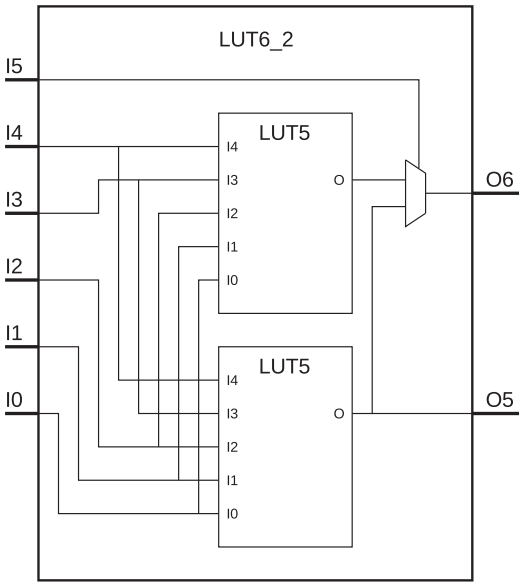

Fig. 1. Six-input, two-output Look-Up Table.

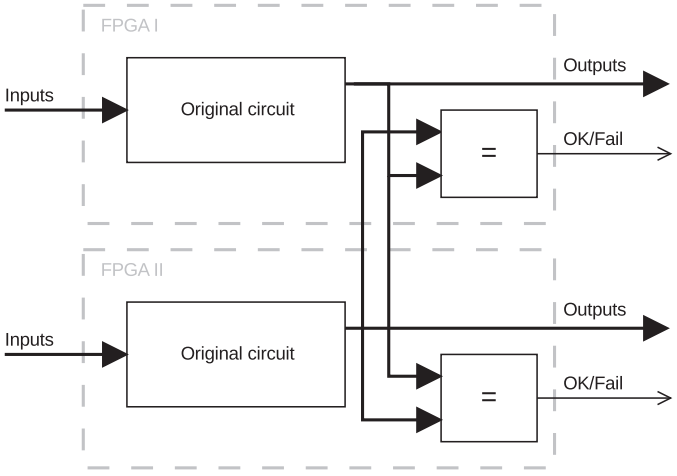

Fig. 2. Duplication with Comparison system.

to the checker that is able to determine whether the original circuit or the predictor is faulty or not. The checker must be able to detect faults inside itself to achieve a (TSC) system.

2.3. Common hardware redundancy types

In this section, two well-known redundancy types are compared. The first one – a (TSC) – is able to detect any single fault affecting the system, but it is not able to localize it. The duplication with comparison is not fully functional when a fault is detected. The second system is a Triple Modular Redundancy (TMR) that is able to mask and localize any single fault affecting the system. The Triple Modular Redundancy system is fully functional when a single fault is present.

2.3.1. Duplication with Comparison

A totally self-checking (DwC) is shown in Fig. 2. A (DwC) contains two copies of the original circuit, two independent sets of inputs, two independent checkers (comparators), and two independent sets of outputs. If any of these parts is present in one copy only, the system is not a (DwC) system anymore, because a fault in such part cannot be detected.