

Hardware redundancy architecture based on reconfigurable logic blocks with persistent high reliability improvement

Štefan Krištofík*, Marcel Baláž, Peter Malík

Institute of Informatics, Slovak Academy of Sciences Dúbravská cesta 9, 845 07 Bratislava, Slovakia

ARTICLE INFO

Keywords:

Digital logic reliability
Fault tolerance
Hardware redundancy
Fault compensation
Reconfigurable logic

ABSTRACT

On-chip digital system reliability is an important concern today in many critical applications. To achieve high reliability, hardware redundancy architectures are often employed. One of the most frequently used architectures is the triple modular redundancy due to its simplicity and good reliability improvement in the early stages of a product lifetime. However, one of its main drawbacks is the high area overhead, which presents a problem especially in non-time-critical applications. An alternative approach based on reconfigurable logic blocks is proposed in this paper for non-time-critical applications. The aim is to reduce the area overhead below the triple modular redundancy levels while also improving the overall system reliability over the entire operational stage of the product lifetime. Experimental results show that using reconfigurable logic blocks instead of triple modular redundancy the area overhead of redundancy can be significantly reduced up to 71% while also increasing the system reliability over the entire lifetime.

1. Introduction

High reliability is necessary for on-chip digital systems used in critical real world applications such as transportation, health, space missions or national security. Now it is becoming a necessity also in non-critical applications, e.g., consumer electronics, to achieve acceptable levels of manufacturing yield of these devices [1,2]. Their yield is negatively impacted by the continuous reduction of feature sizes of VLSI designs and the resulting susceptibility to radiation effects and faults. Also despite many advancements in VLSI circuit production techniques, the fabrication processes are still not perfect and faults may occur. Insufficient levels of reliability can potentially lead to field failures that are very expensive to repair and can also damage the company reputation [3,2].

The required levels of reliability are achieved by making systems fault-tolerant, providing functionality despite the presence of hardware faults. Fault tolerant architectures utilize redundancy. Hardware faults are usually handled by time, information or hardware redundancy. The focus of this paper is hardware redundancy which is based on adding extra hardware into the design to detect, isolate and negate the effects of faults.

Two main types of hardware redundancy are static and dynamic [1]. The static redundancy focuses on fault masking by having

multiple identical functional units performing the same computations at the same time. The dynamic redundancy activates spares to replace faulty units if a fault is detected and isolated. The static redundancy is best used in time-critical applications, where the overall reliability has the highest priority and the inherent high area overhead can be accepted [4], e.g., space missions. In other non-time-critical areas, where the chip area and production cost are key concerns, e.g., mass produced consumer electronics, the high area overhead of these approaches is not tolerable. To reduce the chip production cost of common digital systems in non-time-critical applications, new dynamic hardware redundancy approaches are needed with low area overhead and at the same time offering high reliability improvement. The paper provides a contribution to this area of research.

1.1. Novel contributions and motivation of the paper

This paper offers the following contributions in the field of fault tolerant digital system design:

- A new dynamic hardware redundancy architecture is proposed for arbitrary digital logic in non-time-critical applications as an alternative to existing static redundancy approaches.
- A method is developed for the proposed architecture, enabling

abbreviations: TMR, triple modular redundancy; NMR, n-modular redundancy; GDR, generic dynamic redundancy; RLB, reconfigurable logic block; FU, functional unit; BU, backup unit; FDRP, fault detection and repair procedure; FI, fault indication; FDL, fault detection and localization; FR, fault repair

* Corresponding author.

E-mail addresses: stefan.kristofik@savba.sk (Š. Krištofík), marcel.balaz@savba.sk (M. Baláž), p.malik@savba.sk (P. Malík).

<https://doi.org/10.1016/j.microrel.2018.04.010>

Received 30 October 2017; Received in revised form 12 April 2018; Accepted 14 April 2018

Available online 25 May 2018

0026-2714/ © 2018 Elsevier Ltd. All rights reserved.

adjustment of its parameters: reliability improvement and area overhead.

The main motivation of the paper is to provide a solution to reduce the chip production costs in non-time-critical applications by proposing a new design approach as an alternative to the existing static redundancy approaches. To achieve this, two main design objectives of the proposed architecture (when compared to static redundancy) can be stated as (a) area overhead reduction and (b) persistent reliability improvement over the entire operation stage of a product lifetime. An architecture fulfilling these design objectives would no longer suffer from the disadvantages of the static redundancy and would offer a solution for reducing the chip production cost in non-time-critical applications.

1.2. Fault types considered

Fault tolerance against both permanent and transient faults is seen as one of the important aspects of the modern digital system design [2]. In contrast to the lasting effects of permanent faults, transient faults may persist in a circuit for indefinitely amounts of time, spanning single or multiple cycles (multi-cycle transient faults [5,6]). Further, shrinking feature sizes allow a single fault to affect multiple units in a design [7]. This paper considers logical fault models [8,9] of both permanent and multi-cycle transient faults [10], caused by radiation, particle strikes or other sources. All faults are handled equally by the proposed architecture as long as their effects propagate to the outputs of the functional units where they are detectable.

2. Related work

Numerous on-chip (or built-in) hardware redundancy architectures have been proposed and studied in literature, e.g. [1]. One of the most frequently used static hardware redundancy architecture types is the n -modular redundancy (NMR) also often called the m -of- n system, where n is the total number of functional units (FUs), comprising one original FU and $n - 1$ backup units (BUs), and m is the minimal number of FU that have to be fault-free to guarantee the validity of outputs. The well-known triple modular redundancy (TMR) architecture [11] is a subset of NMR for $m = 2, n = 3$. It is especially successful due to the simple implementation (e.g., in FPGAs [12]) and the significant reliability improvement over simplex (baseline FU with no redundancy) in the early stages of a product lifetime. However, the reliability decreases below simplex after a certain point in time, making it disadvantageous in the later stages of a product lifetime. The main disadvantage is the large amount of area overhead consisting of 200% needed for the two copies of the original FU and the additional area needed for the majority circuit. The majority voter is not protected against faults in the basic TMR version and the system fails if the voter fails. Moreover, the system cannot guarantee the validity of outputs when 2 or more units are faulty at a time.

Recently, there has been some progress in the field of dynamic redundancy using reconfigurable logic blocks (RLBs), e.g., [13,4,14]. Basic principles of RLB architectures were introduced in [13]. The main idea is to extract a group of existing, functionally identical FUs within a random digital logic (denoted by p) and add a number of hot BUs with the same function for them (denoted by q). FUs have independent inputs, i.e., each unit can process different data at a time. Each faulty FU can be replaced by one of BUs when needed. This is done by re-routing the inputs of faulty units through backups using input switches and filtering the outputs of faulty units using output switches and instead propagating the outputs of backups to the system output (see Fig. 1). Each state of switching circuitry corresponds to a unique RLB architecture state. Various types of RLB architectures can have different number of possible states. RLB states define the group of units that are currently connected to data by switching circuitry. For example, valid

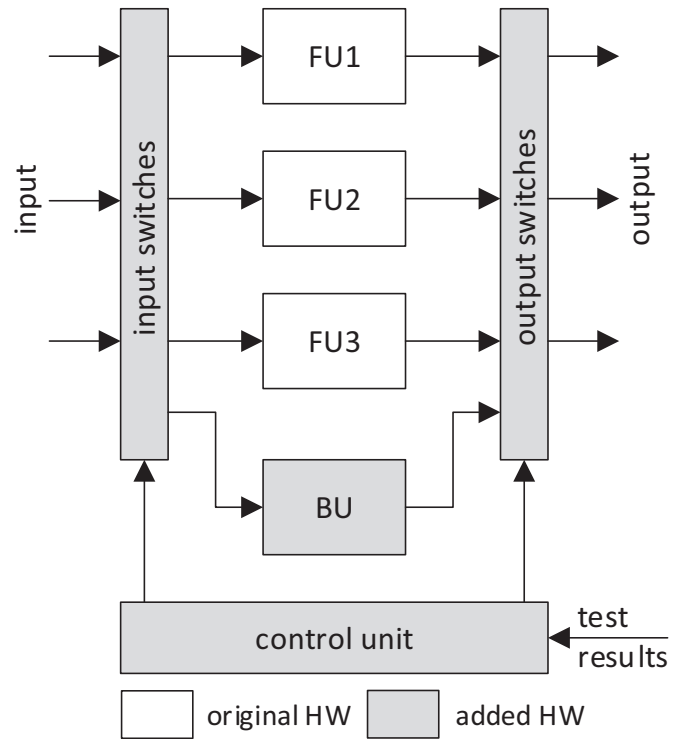


Fig. 1. Principal scheme of an RLB 3 + 1 system.

RLB states for $p = 3, q = 1$ include 'FU 1, FU 2, FU 3' or 'FU 2, FU 3, BU'. Fault detection is an important part of the RLB architecture. It is handled by a built-in tester which is already located on the chip. In terms of area overhead, an additional switching circuitry is needed to reroute the RLB inputs and outputs. Also a control unit is needed to drive the switching activity based on the fault information provided by the built-in tester. The group consisting of p FUs, q BUs and the support circuitry (switching and control circuitry), is referred to as an RLB of type $p + q$ or RLB $p + q$. The process of assembling such group of units is referred to as RLB integration. In such architectures, the correct operation is guaranteed as long as a minimum of p units (functional or backup) are working. If all BUs are working, then a maximum of q faulty FUs can be compensated. The architecture fails if more than q FUs are faulty at a time. There is also a possibility of one or more BUs becoming faulty. In this case, the number of faulty FUs that can be compensated is decreased by the number of faulty BUs. All of the above situations are detectable by the built-in tester and the appropriate actions can be executed subsequently, e.g., disconnecting the faulty RLB from data.

In this paper, RLB architectures of type $p + 1$ are of particular interest because of the least additional area required for BUs among all RLB types. The principal scheme of one of such systems, an RLB 3 + 1 system, is shown in Fig. 1. Such systems guarantee the validity of outputs as long as a total of 3 units are working. A maximum of 1 faulty FU can be compensated by utilizing the backup. If BU is faulty, the system will still work as long as all FUs are working. The system will fail when more than 1 units are faulty at a time. RLB 3 + 1 can be in 4 different states: i) FU 1, FU 2, FU 3, ii) FU 2, FU 3, BU, iii) FU 1, FU 3, BU and iv) FU 1, FU 2, BU. The area overhead consists of approx. 33% needed for BU and the additional area needed for support circuitry.

The main RLB advantage is the reliability improvement over simplex during the most of the product lifetime as opposed to TMR which only improves reliability in the early stages of a product lifetime. This means the maintenance repairs or replacements of faulty units are not needed as often as in TMR systems. Another advantage is the possibility of reduced area overhead compared to static redundancy architectures

Download English Version:

<https://daneshyari.com/en/article/6945470>

Download Persian Version:

<https://daneshyari.com/article/6945470>

[Daneshyari.com](https://daneshyari.com)