Microprocessors and Microsystems 39 (2015) 909–918

Contents lists available at ScienceDirect

Microprocessors and Microsystems

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

Functional self-test of high-performance pipe-lined signal processing architectures

Maksim Gorev *, Raimund Ubar, Peeter Ellervee, Sergei Devadze, Jaan Raik, Mart Min

Tallinn University of Technology, Akadeemia tee 15A, 12618 Tallinn, Estonia

ARTICLE INFO

Article history:
Received 21 April 2014
Revised 9 October 2014
Accepted 4 November 2014
Available online 15 November 2014

Keywords:
Pipe-lined signal processing architectures
Built-in self-test
At-speed testing
Design for testability

ABSTRACT

We propose a new methodology for Built-In Self-Test (BIST) where contrary to the traditional scan-path based Logic BIST, the proposed solution for test generation does not need any additional hardware, and will not have any impact on the working performance of the system. A class of digital systems organized as pipe-lined signal processing architectures is targeted. The on-line generated signal data used for processing in the system serve as test pattern sources. Testing under normal working conditions and with typically processed data, allows exercising of the system on-line and at-speed, facilitating the detection of dynamic faults like delays and cross-talks to achieve high test quality. The proposed new self-test method is free from the negative aspect of over-testing, compared to the traditional Logic BIST approaches, and uses minimal amount of added hardware. Experimental research was based on the case study of specialized bio-signal processor architecture. The experiments showed promising results in reducing the cost of testing and achieving high fault coverage.

1. Introduction

The technology advancements impose new challenges to testing modern chips as device geometries shrink, and deep-submicron delay defects are becoming more and more important requiring more accurate dynamic tests than before [1]. Therefore testing of chips closer to real working conditions by so called at-speed test is becoming the must.

Increasing size and complexity of digital systems directly reflects in more demanding test generation and application strategies. The use of scan chains has proven to be often inadequate increasing the cost in terms of additional hardware and testing time [2], excessive power dissipation during test [3] and leading to yield loss because of over-testing [4].

A lot of research has been carried out to relieve the burden of external testers by introducing system self-test approaches like hardware-based Logic Built-in Self-Test (LBIST) which typically use Linear Feedback Shift Registers (LFSR) [5]. In LBIST, typical functions of external test equipment like test generation and response analysis are carried out on-chip, so that the tester should not handle high-speed signals externally and its role should remain only to send the test enable signals to the chip under test, and to receive the pass/fail signals. For example, scan-based and Logic BIST solutions such as [6] relax the requirements on testers and considerably reduce the overall testing cost.

An important trend today is the at-speed test [7] having additional benefit of the ability to test circuits under conditions that are as close as possible to normal circuit operation. This factor has a direct impact on the number of chips that are found defective during system operation but still pass all manufacturing and functional tests. At-speed testing can be used for characterization and can also expedite test application time.

The question is whether a self-test sequence running in the system can adequately exercise its hardware components satisfying the targeted fault coverage requirements. Achieving the test quality target requires application of proper test sequences that is the focus of the current paper. It should also be pointed out that the quality of a test is measured not only by its fault coverage, but also by its code size (to be stored in the memory of the chip), hardware overhead, and by the test execution time.

The goal of the paper is to propose an approach which combines the ideas of traditional LBIST with at-speed testing to improve the test quality at less testing overhead and avoiding performance loss compared to the traditional self-test approaches. The feasibility and efficiency of the new method is demonstrated for a particular class of pipe-lined processing architectures which are easily adaptable for at-speed on-line self-testing by inherent functional sequences.

* Corresponding author.

E-mail addresses: maksim.gorev@ati.ttu.ee (M. Gorev), raimund.ubar@ati.ttu.ee (R. Ubar), peeter.ellervee@ati.ttu.ee (P. Ellervee), sergei.devadze@ati.ttu.ee (S. Devadze), jaan.raik@ati.ttu.ee (J. Raik), min@elin.ttu.ee (M. Min).

http://dx.doi.org/10.1016/j.micpro.2014.11.002

The rest of the paper is organized as follows. In Section 2, an overview about state-of-the-art is given. Section 3 presents the general idea of the proposed method, followed with the description of the evaluation bench in Section 4, which is needed for exploration of solutions for implementing the method. Section 5 presents the description of the representative case study, and in Section 6 the results of experimental research are discussed. Section 7 concludes the paper.

2. State-of-the-art of self-test techniques

In traditional LBIST, test pattern generation is mostly performed by Linear Feedback Shift Registers (LFSR) [5], cellular automata [8] or multifunctional registers like BILBO (Built-in Logic Block Observer) [5] to apply pseudorandom patterns to the Unit Under Test (UUT) and to analyze its output responses. However, many circuits contain random-pattern-resistant faults, which limit the fault coverage that can be achieved with this approach.

One method to improve the fault coverage for LBIST is to modify the UUT by either inserting test points [4] or by redesigning it to improve the fault coverage [6]. The drawback of these techniques is that they generally add additional logic levels to the circuitry that can degrade system performance. Another possibility to improve the fault coverage is to use weighted pseudorandom test sequences [9]. The disadvantage of this approach is in the need of storing of the weight sets on chip, and also dedicated control logic is required to switch between weights, so the hardware overhead may become large.

A “mixed mode” approach, where deterministic patterns will be added to detect hard-to-test faults, has been developed in [10–13]. In [10] a technique based on reseeding LFSR was proposed that reduces the storage requirements. In [11], multi-polynomial LFSR for encoding a set of deterministic test cubes was introduced, and in [12] a technique called bit flipping for generating deterministic test cubes using BIST control logic was proposed. Further, in [13] a mixed-mode approach was presented in which deterministic test cubes are embedded in the pseudorandom sequence of bits itself.

Established BIST solutions use special hardware (typically LFSR) for test pattern generation (TPG) and test response evaluation (TRE) on chip [5], but this in general introduces significant area overhead and performance degradation. To overcome these problems, specialized methods were proposed which exploit specific functional units such as arithmetic units for on-chip test pattern generation [14,16], which may afford to reach similar fault coverage like traditional LFSR-s. These methods are called Arithmetic BIST (ABIST), since they essentially adopt the additive congruential generation scheme of pseudo-random numbers [17].

In [18,19], a mixed-mode or hybrid BIST approach was proposed, where a test set is assembled from two parts, from pseudorandom test patterns that are generated on-line, and deterministic test patterns that are generated off-line and stored in the system. A combination of both test sources in an optimized fashion allowed improving the traditional LBIST in targeting hard-to-test faults. A similar approach called Hybrid Functional BIST (HyFBIST), where instead of LBIST the inherent functional sequences were used, was proposed in [20,21] for testing digital systems, and particularly micro-programmed data-paths.

In this paper we generalize and combine the ideas of using inherent functional blocks for test generation [14,15] and the inherent working sequences produced by the UUT itself for self-testing purposes. We propose an overall functional self-test concept for pipelined architectures where the working sequences are produced on the primary inputs of the system and the internal signals are monitored in selected test-points by Multiple Input Signature Analysers (MISR). We propose a systematic procedure for selecting the test-points to achieve the best overall fault coverage at minimum testing overhead and cost.

To our knowledge, the usage of digital representation of analog signal sequences as a functional test for testing digital circuits (signal processing architectures) is investigated in our paper the first time. Main idea is to take the input data, which is close to what the circuit-under-test would most probably have during its normal operation and apply this data as an at-speed test. In our case this input data is digital representation of the sine signal. It will be shown in results, that such a signal could yield better fault coverage in comparison to traditional pseudo-random LFSR sequence. This can also be considered as one step further compared to the Arithmetic BIST (ABIST), since the source for the first stage of UUT is stimulated using more complicated equation (sine wave), than traditionally used in ABIST. The next stages of the UUT can be considered as test generators similar to ABIST. The Functional test strategies (e.g. software based self-test) used for example in microprocessors, are traditionally using dedicated software test routines, which have to be stored in the memory. In our case, there is no need to store in the memory such test routines or other test data.

3. General description of the method

Consider a digital system as a network of sub-circuits (blocks) where all the blocks may play simultaneously two roles: on one hand, each block will be itself UUT, and on the other hand, it will serve as the test pattern generator for the subsequent blocks it is feeding. As the overall test source, selected input working sequences (as functional test) will be used.

Two main problems arise: (1) how to find the best functional test sequences, and (2) how to find the minimal set of test-points for monitoring to achieve the highest fault coverage of testing.

In some cases, the first problem can be solved straightforwardly like in the instruction set architectures or in signal processing units. In the first case, the instructions can be exercised one by one where the problem recedes to finding only proper data (operands) as test patterns [22,23]. In case of signal processing units, the analog signals to be processed can be used as candidates for exploiting in testing purposes as well.

We are investigating the possibility of using given digital representation of analog signals as stimuli for testing signal processors. The idea is similar to random (LFSR based) testing where the critical point is analysis of the test quality as the function of test length.

For example, in bio-impedance spectroscopy, for measuring the bio-impedance typically the following signals are generated and processed as shown in Fig. 1: *sine* [28] and *chirp* [29]. These signal sequences may be used as well in the role of stimuli (i.e., functional test sequences) for self-testing purposes for the same signal processor itself. The quality of the listed signals as test stimuli can be compared with popular *saw-tooth* analog signal and pseudorandom LFSR sequences which are traditionally used in the Logic BIST solutions. *Saw-tooth* is easy to generate digitally; this is the reason why it is widely used in signal generation and processing. It can also be thought of as an additive generator of exhaustive patterns.

The second problem of selecting test-points for monitoring the test process depends how well can the faults in different blocks be detected by the given functional test sequence.

In Fig. 2, an example of a pipe-lined signal processing unit is given which is partitioned into 6 blocks.

Two solutions are demonstrated for monitoring the behavior of the circuit with MISRs. The solution in Fig. 2a shows the case where all blocks are monitored whereas in the solution depicted in Fig. 2b, only three MISR are used: the first is monitoring the behavior