Sequential Circuits BIST Synthesis from Signal Specifications

Jaan Raik¹, Maksim Jenihhin¹, Rain Adelbert²

¹Department of Computer Engineering, Tallinn University of Technology
Raja 15, 12618 Tallinn, Estonia
{jaan | maksim}@pld.ttu.ee

²Artec Design OÜ
Türi 10c, Tallinn, Estonia
Rain.Adelbert@artecdesign.ee

Abstract

Efficient Built-In Self-Test (BIST) solutions for certain cryptographic applications have been known for a long time. However, area-efficient methods for the general case of sequential circuits are missing. Current paper proposes a novel approach to BIST synthesis for synchronous sequential circuits. The approach is based on designer's specification of the control signal modes. Relying on that formal specification BIST structures are created and fault simulated in order to assess the fault coverage. An optimal solution is selected and corresponding BIST hardware is synthesized from the signal specification. Additional benefit of the approach is to provide for high-quality test patterns for cores lacking scan infrastructure as commercial ATPG tools able to efficiently handle sequential cores exceeding couple of thousands logic gates are currently missing. The experiments show that high quality tests and BIST structures can be obtained by implementing the proposed method for non-scan cores.

1. Introduction

Scan chain architectures to handle test of digital cores is widely supported by commercial CAD tools. While inserting scan chains to a design allows to easily generate high quality tests, there are also a number of limitations:

- 1) Performance and routing overhead;
- 2) At-speed test is possible but hard to implement;
- 3) Excessive amount of test data necessary, usually to be saved to external tester's expensive and limited memory;
- 4) Field test not supported;
- 5) Relying on external ATE means often 'testing today's technology with yesterday's one';
- 6) Scan-based tests are known to cover defects that would never influence cores behavior in the normal functional mode. Thus, there could be yield loss caused by 'overtesting'.
- 7) The core could be obtained from a third party provider and it is often too risky, costly or even impossible to modify it.

Current paper proposes a global Built-In Self-Test (BIST) approach to sequential cores. The method does

not require any modifications to the structure of the core under test.

The problem of BIST for combinational and full-scan circuits has been thoroughly researched in the past. Implementing weighted random test patterns [1, 2] and bit-flipping [3] have been among the most efficient solutions. Nachman, Saluja et al. [4] propose a method where the values are kept unchanged at primary inputs and scan registers while a certain number of clock pulses are applied. This technique, called freezing, requires preliminary testability analysis of the circuit structure. Furthermore, the above approach is applicable for circuits containing scan-chains only.

However, rather limited amount of work is available on BIST for non-scan sequential designs. Pomeranz and Reddy present a solution for the general case of sequential circuits [5]. The main problem is an excessive hardware overhead since dedicated test pattern generators are to be tailored for each individual primary input. In [6], Chakrabarty proposes a method similar to the reseeding approach [7]. Here, deterministic patterns are embedded to a sequence generated on-chip by using twisted ring counters (or Johnson counters, as they are also referred to).

Global BIST approaches, where test pattern generator is inserted before the primary inputs and signature analysis performed at the primary outputs has been a known solution for a long time. Anheier et al. [8] prove that most of the cryptographic algorithms are easily pseudorandom testable, thus, crypto-cores can be easily tested by implementing a global BIST approach. In addition, many arithmetic and DSP cores are easily testable. This fact is also supported by the experiments carried out in Section 4.

Current paper proposes a novel approach to global BIST synthesis for synchronous sequential circuits. The approach is based on designer's specification of the control signal modes. Relying on that formal specification constrained-random BIST structures are created and fault simulated in order to assess the fault coverage. An optimal solution is selected and corresponding BIST hardware is synthesized from the signal specification. Additional benefit of the approach is to provide for high-quality test patterns for cores lacking scan infrastructure as commercial ATPG tools able to efficiently

handle sequential cores exceeding couple of thousands logic gates are currently missing (e.g. [9]). The experiments presented in Section 4 show that high quality tests and BIST structures can be obtained by implementing the proposed method for non-scan cores.

It is important to note that current paper does not address the problem of signature analysis for sequential cores, which is a separate research topic.

The paper is organized as follows. Chapter 2 explains how core input signals can be categorized and gives a general overview of the test generation flow. Chapter 3 presents the BIST architecture synthesis from the signal specification. In Chapter 4, experimental results on some industrial and public domain cores are provided. Finally, conclusions and future work are given.

2. Signals specification for sequential test

Cores from third party providers include seldom any test related information. However, there is always a description of the functionality of the I/O signals, where clocking, reset signals and strobes are shown. The main goal of current paper is to allow synthesis of global BIST structures from an easy to specify formal description of the core input signal behavior.

The global BIST approach is based on a concept of test sequences. The user can specify the length of the sequence and the number of test sequences. In current implementation, both, sequence length and the number of sequences have to be powers of two.

At the beginning of each sequence global reset signal is activated. In addition to handling the global reset, five classes of signals are distinguished in the proposed approach. The five types are explained below and the underlying implementation in hardware is presented in Section 3.

1) Pseudo-random, update at every clock cycle (default)

By default, all signals are assumed to be of type 1, i.e. they are connected to a Linear Feedback Shift Register (LFSR) in the test mode. The LFSR generates pseudo-random values to the input signals at each clock cycle.

2) Pseudo-random, update at every test sequence

An alternative for clocking the LFSR by the test clock is to synchronize only at the beginning of each test sequence. For some cases this solution might provide for less fault coverage. However, it allows to keep power consumption of the test down. Furthermore, for some signals it might be required that they should not change during the test sequence. In current implementation, keyword 'HOLD' is used for signals of type 2.

3) Mode counter

Signals of type 3 implement an approach, where tests are generated by a counter. The toggling frequency of

each signal can be set. For example, 'PERIOD/2' means that half of the test sequences the signal is held constantly zero, while the rest of the test it is held one. 'PERIOD/4' will double the toggling frequency, 'PERIOD/8' will multiply it by four, etc.

This approach is useful when some control signal combinations have to be exhaustively covered by the test.

4) Fixed waveform

Applying different specific waveforms in order to guarantee some desired functional behavior of the core during test might be necessary. However, hardware implementation could be rather costly. As a solution it is possible to resort to methods described in [5] and [6].

5) Dependencies from other signals

Some signals might be required to hold values, which are logic functions of other input or output signals.

The authors have so far not experimented with signals of type 4 and 5. Arithmetic cores were tested most efficiently with signals of type 1. Signals of type 2 were useful when testing I2C controller and an industrial memory controller. The latter required also signals of type 3. More details about the experiments are presented in Section 4.

It is important to separate the stage of finding the most suitable signal specification from the stage of finding the optimum feedback polynomial structure and initial states for the LFSR when searching for the best global BIST solution. Otherwise, if signal specifications would be evaluated on specific LFSR configurations then less suitable polynomials and initial states could possibly void a very optimal spec. Large number of test iterations should then be made to evaluate each signal spec with different LFSR configurations. Therefore, current approach uses a sophisticated pseudo-random number generation algorithm based on L'Ecullier's method to obtain the test data.

General design flow of the proposed approach is as follows:

1. Create a signal specification for test.
2. Generate the constrained-random test.
3. Fault simulate the test patterns.
4. If fault coverage is not satisfactory then go to 1.
5. Select polynomials and initial states for the LFSRs.
6. Regenerate the test patterns.
7. Fault simulate.
8. If fault coverage obtained by LFSR is lower then go to 5, else optimal BIST structure is found.

Hardware implementation of the constrained-random global BIST scheme is introduced in the following Section.

Fig.1. Global BIST architecture for 2^n sequences of length 2^k clock-cycles

3. BIST synthesis from signal specifications

Figure 1 presents the hardware implementation of the proposed global BIST scheme. The presented architecture supports BIST for 2^n sequences of length 2^k clock-cycles.

The central component of the architecture is an $n+k$ bit resettable counter, which is used for, both, test pattern generation and test control. The counter and LFSR1, which is the unit for generating test data for signals of type 1, are clocked by the test clock.

OR-tree from k least significant bit positions of the counter clocks LFSR2 (i.e. the unit for generating patterns for signals of type 2). The output of the OR-tree is connected to the global reset signal in the test mode. If reset is active-high then an additional inverter is necessary.

Signals of type 3 are received directly from the $k+1$ -th to the $k+n$ -th bit of the counter.

Signals of type 4 and 5 have not yet been implemented in the proposed tool. However, a possible implementation could be a decoder buffered by a register (shown by dotted rectangles in Figure 1). Known sequential BIST methods [5, 6] based on e.g. twisted ring counters and reseeding [7] could also be considered.

4. Experimental results

The goal of the authors was to find realistic benchmark cores of possibly wide range of applications and types. So far, three cores from OpenCores library and one industrial example have been synthesized and tested by the approach. Two of the cores were arithmetic units and two of them controllers. Experimental results are presented in Table 1.

Experiments show, as expected, that very good fault coverage can be obtained by global BIST by using pure pseudo-random data for the arithmetic cores. Constrained-random approach did not have any benefits in testing such cores. Thus, more complex, control-dominated circuits were necessary to evaluate the method proposed in the paper.

An I2C controller and an industrial memory controller design were selected for this purpose. The fault coverage achieved by pseudo-random tests was just 40.87 % for the I2C core. It was raised to 67.95 % by creating a signal specification from the core's functional spec. Similarly, memory controller's BIST fault coverage increased to 61 % from just 16 % of stuck-at fault coverage. This shows the potential of the global BIST for the more complex class of sequential cores.

Table 1. Global BIST experiments on sequential cores

Circuit	HCSA_ALU	CORDIC	I2C controller	Memory controller
# faults	1686	3626	2440	24242
# flip-flops	100	120	114	639
# inputs	40	121	20	300
pseudo-random fault coverage, %	96.18	84.35	40.87	16.07
constrained-random fault coverage, %	96.18*	84.35*	67.95	61.36
CPU time, s	9.3	165.6	207.5	**

* Constrained random did not improve the fault coverage

** Fault simulation took several hours of CPU time

5. Conclusions and future work

The paper presented a novel method for sequential circuit BIST, where designer's specification of core input signals is used for generating high quality test patterns. The same specification is then synthesized into a global BIST architecture implementing a constrained-random approach. Preliminary experimental results on industrial and public domain cores have shown the feasibility of the proposed method, both, for control- and data-dominated circuits. For the former case, four times fault coverage improvement was achieved based on a case study of an industrial memory controller. For the data-dominated cases, high quality area efficient solutions were synthesized.

As a future work it is planned to experiment with more control-oriented hard-to-test cores. In addition, the authors plan to implement parallelized solutions (grid computing, etc.) and efficient fault emulation approaches [10] in order to speed up the BIST simulation step.

Acknowledgments

The work has been supported in part by Estonian Science Foundation grant 5637 and Enterprise Estonia funded ELIKO Technology Development Center.

References

- [1] F. Brglez, et al., "Hardware-Based Weighted Random Pattern Generation for Boundary-Scan", *Proc. IEEE Int. Test Conf.*, pp. 264-274, 1989.
- [2] H.-J. Wunderlich. „Multiple distributions for biased random test patterns”, *Proc. ITC*, 1988, pp. 236-244
- [3] H.-J. Wunderlich, G. Kiefer. „Bit-Flipping BIST”, *Proc. ICCAD*, November 1996, pp. 337-343
- [4] L. Nachman, K.K. Saluja, S. Upadhyaya, R. Reuse, „Random pattern testing for sequential circuits revisited”, *Proc. of FTCS-26*, 1996.
- [5] I. Pomeranz, S.M. Reddy, „Built-in test generation for synchronous sequential circuits”, *ICCAD 1997*: pp. 421-426.
- [6] K. Chakrabarty, B.T. Murray, V. Iyengar, "Deterministic built-in test pattern generation for high-performance circuits using twisted-ring counters", *IEEE Trans. VLSI Systems*, Vol.8., No.5, Oct. 2000.
- [7] S. Hellebrand, S. Tarnick, J. Rajski, and B. Courtois, "Generation of Vector Patterns through Re-seeding of Multiple-Polynomial Linear Feedback Shift Registers", *Proc. IEEE Int. Test Conf.*, pp. 120-129, 1992.
- [8] A. Schubert, W. Anheier, "On random pattern testability of cryptographic VLSI cores", *Proc. of IEEE European Test Workshop*, May 1999, pp. 15-20.
- [9] G. Jervan, Z. Peng, O. Goloubeva, M. Sonja Reorda, M. Violante, "High-level and hierarchical test sequence generation", *IEEE High-Level Design Validation and Test Workshop*, Oct. 2002 pp. 169 – 174.
- [10] J.Raik, P.Ellervee, V.Tihhomirov, R.Ubar, "Improved Fault Emulation for Synchronous Sequential Circuits", *Proc. of the EUROMICRO Conference on Digital Systems Design*, Porto, Portugal, 2005