# AN EFFICIENT MIXED-MODE BIST TECHNIQUE

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

***Abstract.*** *We propose a new built-in self-test (BIST) method based on a combination of a pseudo-random test method with a deterministic test. This enables us to reach a high fault coverage in a short test time and with a low area overhead. The main feature of the method is that there are no memory elements to store the deterministic test patterns; the test patterns are being produced by a transformation of the non-testing pseudo-random patterns. This transformation is being done by a purely combinational block, while we try to keep this block as small as possible. Our method can be apprehended as a generalization of a bit-fixing method. We synthesize the transformation logic by a slightly modified column-matching algorithm proposed before.*

## 1. Introduction

The complexity of VLSI circuits rapidly grows, therefore their testing using only external test equipment (ATE) is becoming impossible. Huge amount of necessary test vectors prolongs the testing time and demands complicated and expensive testers. Built-in Self-Test (BIST) requires no external tester, since all the circuitry needed to conduct a test is included in the very circuit. This is paid by an area overhead, a long test time and often a low fault coverage. Many recent BIST methods have been trying to find some trade-off between these three aspects that are mutually antipodal. A high fault coverage means either a long test time (exhaustive test), or a high area overhead (ROM-based BIST). A pseudo-random testing established the simplest trade-off between all the three criteria. With an extremely low area overhead the circuit can be tested usually up to more than 90% in a relatively small number of clock cycles (thousands). To improve the fault coverage and to reduce the test time, many enhancements of this pseudo-random principle were developed. Some methods incorporate a memory into BIST to store deterministic test vectors. Since the chip area needed for a memory is often large, these patterns are being compressed somehow, and then they are being decompressed by a LFSR [1-2]. Other methods try to modify the pseudo-random pattern generator (PRPG) somehow to improve a fault coverage [3, 5] or modify some of the PRPG patterns by some combinational logic into deterministic tests [4, 5]. These methods are often referred to as a *mixed-mode* BIST.

Synthesis of the combinational logic transforming the pseudo-random patterns into deterministic tests is based on our column-matching algorithm [6]. In this paper we

describe an enhancement of this method to support a mixed-mode BIST, which significantly reduces the output decoder logic.

The paper is structured as follows: Section 2 describes the major principles, the experimental results are presented in Section 3, Section 4 concludes the paper.

## 2. Principles of the Mixed-Mode BIST Method

In order to improve the fault coverage we modify some pseudorandom vectors generated by a LFSR to obtain deterministic tests detecting the hard-to-detect faults. In a bit-fixing approach [4] the LFSR patterns are being modified by AND and OR gates driven by an additional logic. Here the transformation logic needs to keep track of the LFSR vectors and modify the appropriate bits in some vectors only. Vectors that do not detect any faults were being modified in previous approaches, while these vectors were picked up from any of the source vectors. However, praxis shows that some limited set of the initial pseudorandom vectors does detect faults and, after this initial sequence, the fault detection capability of the vectors quicky drops to zero. Thus, effectively dividing the test execution into two phases becomes a good solution. First the unmodified pseudorandom LFSR patterns are applied to the CUT to test the easy-to-detect faults, and then the succeeding vectors are transformed into deterministic tests precomputed by some ATPG tool.

Like in the bit-fixing and bit-flipping approaches, the additional combinational logic consists of two blocks: the pattern transformation logic itself (Decoder) and the logic switching between the two test phases. We propose a more generalized approach where the switching logic is not implemented as AND and/or OR gates, but by multiplexers. This principle has two advantages in general: firstly, the chip area needed to implement a multiplexer is comparable to a standard gate in the CMOS logic [7]. Moreover, these multiplexers can be driven together, preferably by an external signal, or by an additional pattern counter. Thus, the transformation logic needs not reflect the initial set of the LFSR patterns (these are taken as external don't cares), only the patterns that have to be transformed are considered. This fact significantly reduces the decoder logic.

The decoder is a purely combinational block transforming the LFSR patterns following after the pseudorandom test sequence into a deterministic test set. It is based on our column matching method [6]. Here we try to implement most of the outputs of the decoder logic by assigning them to the inputs, thus without any circuitry. If we succeed in matching two columns at the same position, we call it a *direct match*. Here no logic is needed, even for the switch. Any other matches involve a multiplexer, the unmatched columns have to be synthesized by the decoder. An artificial illustrative example is shown in Fig. 1. The 5-bit LFSR runs for 5 cycles first and the easily testable faults are detected. Then we run the fault simulation to find the undetected faults, for which the test vectors are generated by an ATPG. At the end the decoder logic is synthesized for these test vectors and the succeeding LFSR patterns. The resulting circuitry is shown in Fig. 2.

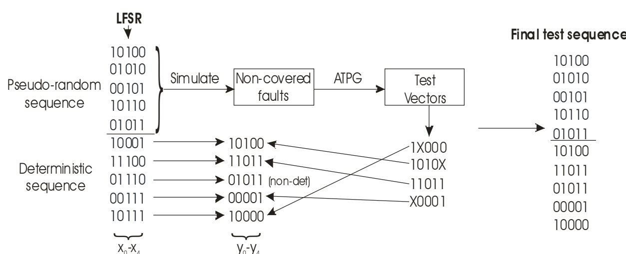


Fig. 1: Test sequence generation

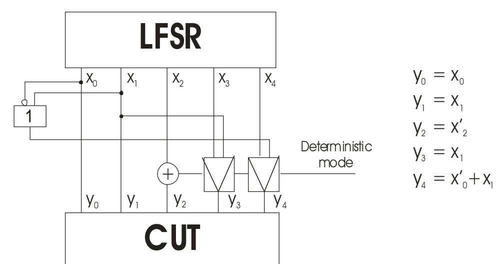


Fig. 2: BIST circuitry

## 3. Experimental Results

The method was extensively tested on standard ISCAS benchmarks, both on the combinational ones [8] and the full-scan versions of the sequential benchmarks [9]. To show the tradeoff between the test time and area overhead, the BIST for all of the benchmarks was synthesized using two different test lengths. We have used the FSIM program as a fault simulator and ATALANTA ATPG tool [10].

For all the benchmarks 100% coverage of detectable single stuck-at faults is assumed. The results of the experiments are shown in Table 1. After the benchmark circuit name the number of its inputs is indicated (*"inps"*). The number of pseudorandom LFSR patterns needed to reach 100% fault coverage (without using any additional logic) is shown in the *"PR"* column, just to illustrate the pseudorandom testability of the circuit. The *"rand / det."* column describes the number of cycles used to test the circuit using our mixed-mode BIST; the first number indicates the length of the pseudorandom phase, the second indicates the length of the deterministic one. Thus, the total number of LFSR cycles needed to complete the BIST is equal to the sum of these two numbers. The *"ud."* column shows the number of undetected faults after the fault simulation using the *"rand"* vectors. To cover these faults *"ATPG"* test vectors were generated by ATALANTA.

The *"mtch."* column gives the total number of column matches found after applying the modified column-matching algorithm. From these matches *"d. mtch."* were direct matches. After that, the decoder logic was synthesized using BOOM [11, 12, 13], the number of gate equivalents [7] obtained is indicated in the *"GEs"* column. In the cases where all the columns were matched, there is no output decoder needed, thus the number of GEs is equal to 0 (fields in bold). The total area overhead of the BIST combinational logic, thus of both the output decoder and the switching logic is shown in the *"tot. GEs"* column. This number can be computed as follows: *tot. GEs = GEs + 1.5(inps - d. matches)*, since MUXes have to be present at the outputs, that are not directly matched. The GEs for a MUX is equal to 1.5.

The last column shows the total computational time in seconds needed to obtain the result on a 900 MHz Athlon CPU. In all the cases the algorithm was run iteratively 10 times, to improve the result.

It is obvious that the BIST area overhead rapidly decreases with increasing test time. Two aspect play role here: the longer the pseudorandom phase runs, the more faults are detected here, thus fewer deterministic vectors are needed to test the rest of the faults. Secondly, the number of column matches reached (both the direct and non-direct) increases with increasing the number of the LFSR patterns to be transformed and with a decrease of the number of the deterministic tests they are to be transformed to.

Table 1. Experimental results

| *bench* | *inps* | *PR* | *rand / det.* | *ud.* | *ATPG* | *mtch.* | *d. mtch.* | *GEs* | *tot.GEs* | *time* |
|---|---|---|---|---|---|---|---|---|---|---|
| c880 | 60 | 7 K | 500 / 500 | 9 | 4 | 60 | 53 | **0** | 10.5 | 0.3 |
| | | | 1000 / 1000 | 5 | 4 | 60 | 59 | **0** | 1.5 | 0.2 |
| c1355 | 41 | 2 K | 500 / 500 | 31 | 12 | 24 | 7 | 19 | 70 | 11.5 |
| | | | 1000 / 1000 | 2 | 1 | 41 | 31 | **0** | 15 | 0.2 |
| c1908 | 33 | 4 K | 1000 / 1000 | 46 | 30 | 29 | 12 | 15 | 46.5 | 15 |
| | | | 2000 / 1000 | 19 | 10 | 33 | 28 | **0** | 7.5 | 0.2 |
| c3540 | 50 | 15 K | 1000 / 1000 | 33 | 22 | 50 | 40 | **0** | 15 | 5.6 |
| | | | 2000 / 1000 | 8 | 8 | 50 | 45 | **0** | 7.5 | 1.4 |
| s641 | 54 | 2 M | 1000 / 500 | 12 | 9 | 54 | 40 | **0** | 21 | 2.7 |
| | | | 4000 / 1000 | 8 | 7 | 54 | 44 | **0** | 15 | 1.1 |

We have compared our method with the bit-fixing [4] and row-matching [5] methods. The results of this comparison are listed in Table 2. The *TL* column indicates the number of test clock cycles. The empty cells indicate that the data for the respective circuit was not available to us.

Table 2. Comparison result

| | Col.-match. | | Bit-fixing | | Row-match. | |
|---|---|---|---|---|---|---|
| *Bench* | *TL* | *GEs* | *TL* | *GEs* | *TL* | *GEs* |
| c880 | 1 K | **10.5** | 1 K | 27 | 1 K | 21 |
| c1355 | 2 K | 15 | 3 K | 11 | 2 K | 0 |
| c1908 | 3 K | **7.5** | 4 K | 12 | 4.5 K | 8 |
| c2670 | 5 K | 172 | 5 K | 121 | 5 K | 119 |
| c3540 | 3 K | 7.5 | 4.5 K | 13 | 4.5 K | 4 |
| c7552 | 8 K | 586 | 10 K | 186 | 8 K | 297 |

| | Col.-match. | | Bit-fixing | | Row-match. | |
|---|---|---|---|---|---|---|
| *Bench* | *TL* | *GEs* | *TL* | *GEs* | *TL* | *GEs* |
| s420 | 1 K | **24.5** | 1 K | 28 | - | - |
| s641 | 4 K | 15 | 10 K | 12 | 10 K | 6 |
| s713 | 5 K | 16.5 | - | - | 5 K | 4 |
| s838 | 6 K | 130 | 10 K | 37 | - | - |
| s1196 | 10 K | **6** | - | - | 10 K | 36 |

## 4. Conclusions

In this paper we have proposed an extension of the column-matching method to a mixed-mode BIST. First the circuit is pseudorandomly tested by LFSR patterns and then the deterministic test patterns detecting yet uncovered faults are being produced. This implies some additional logic that switches between these two modes. We try to minimize this logic as well, particularly by introducing the direct column matches.

The method was tested on the standard ISCAS benchmarks. We have shown the tradeoff between the test time and the area overhead. Longer test time means smaller area overhead and vice versa. For these benchmarks we have compared our method to the other state-of-the art methods. The BIST synthesis tool based on this method is available at [14].

## Acknowledgement

This research was supported by a grant GA 102/04/2137 and MSM 212300014.

## References

[1] Hellebrand, S. et al.: Built-In Test for Circuits with Scan Based on Reseeding of Multiple-Polynomial Linear Feedback Shift Registers. IEEE Trans. on Comp., vol. 44, No. 2, February 1995, pp. 223-233

[2] Hellebrand, S., Liang, H., Wunderlich, H: A Mixed Mode BIST Scheme Based on reseeding of Folding Counters, Proc. IEEE ITC, 2000, pp.778-784

[3] Hartmann, J., Kemnitz, G.: How to Do Weighted Random Testing for BIST, Proc. of International Conference on Computer-Aided Design (ICCAD), pp. 568-571, 1993

[4] Touba, N.A.: Synthesis of mapping logic for generating transformed pseudo-random patterns for BIST, Proc. of International Test Conference, pp. 674-682, 1995

[5] Chatterjee, M., Pradhan, D.K.: A BIST Pattern Generator Design for Near-Perfect Fault Coverage, IEEE Transactions on Computers, vol. 52, no. 12, December 2003, pp. 1543-1558

[6] Fišer, P., Hlavička, J., Kubátová, H.: Column-Matching BIST Exploiting Test Don't-Cares. Proc. 8th IEEE Europian Test Workshop (ETW'03), Maastricht (The Netherlands), 25.-28.5.2003, pp. 215-216

[7] De Micheli, G.: Synthesis and Optimization of Digital Circuits. McGraw-Hill, 1994.

[8] Brglez, F., Fujiwara, H.: A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortan, Proc. of International Symposium on Circuits and Systems, pp. 663-698, 1985

[9] Brglez, F., Bryan, D., Kozminski, K.: Combinational Profiles of Sequential Benchmark Circuits, Proc. of International Symposium of Circuits and Systems, pp. 1929-1934, 1989

[10] Lee, H.K., Ha, D.S.: Atalanta: an Efficient ATPG for Combinational Circuits, Technical Report, 93-12, Dep't of Electrical Eng., Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 1993

[11] Hlavička, J., Fišer, P.: BOOM - a Heuristic Boolean Minimizer. Proc. International Conference on Computer-Aided Design ICCAD 2001, San Jose, California (USA), 4.-8.11.2001, pp. 439-442

[12] Fišer, P., Hlavička, J.: BOOM - A Heuristic Boolean Minimizer, Computers and Informatics, Vol. 22, 2003, No. 1, pp. 19-51

[13] http://service.felk.cvut.cz/vlsi/prj/BOOM

[14] http://service.felk.cvut.cz/vlsi/prj/ColMatch