52. IWK

Internationales Wissenschaftliches Kolloquium
International Scientific Colloquium

PROCCEDINGS | **10 - 13 September 2007**

FACULTY OF COMPUTER SCIENCE AND AUTOMATION

COMPUTER SCIENCE MEETS AUTOMATION

VOLUME II

TECHNISCHE UNIVERSITÄT
ILMENAU

Bibliografische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der deutschen Nationalbiografie; detaillierte bibliografische Daten sind im Internet über **http://dnb.ddb.de** abrufbar.

ISBN 978-3-939473-17-6

Impressum

Herausgeber:	Der Rektor der Technischen Universität Ilmenau Univ.-Prof. Dr. rer. nat. habil. Peter Scharff
Redaktion:	Referat Marketing und Studentische Angelegenheiten Kongressorganisation Andrea Schneider Tel.: +49 3677 69-2520 Fax: +49 3677 69-1743 e-mail: kongressorganisation@tu-ilmenau.de
Redaktionsschluss:	Juli 2007
Verlag:	 Technische Universität Ilmenau/Universitätsbibliothek Universitätsverlag Ilmenau Postfach 10 05 65 98684 Ilmenau www.tu-ilmenau.de/universitaetsverlag
Herstellung und Auslieferung:	Verlagshaus Monsenstein und Vannerdat OHG Am Hawerkamp 31 48155 Münster www.mv-verlag.de
Layout Cover:	www.cey-x.de
Bezugsmöglichkeiten:	Universitätsbibliothek der TU Ilmenau Tel.: +49 3677 69-4615 Fax: +49 3677 69-4602

© Technische Universität Ilmenau (Thür.) 2007

Diese Publikationen und alle in ihr enthaltenen Beiträge und Abbildungen sind urheberrechtlich geschützt. Mit Ausnahme der gesetzlich zugelassenen Fälle ist eine Verwertung ohne Einwilligung der Redaktion strafbar.

Preface

Dear Participants,

Confronted with the ever-increasing complexity of technical processes and the growing demands on their efficiency, security and flexibility, the scientific world needs to establish new methods of engineering design and new methods of systems operation. The factors likely to affect the design of the smart systems of the future will doubtless include the following:

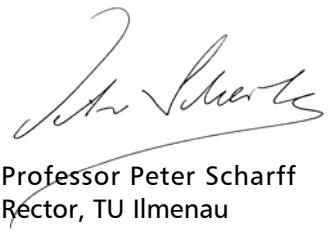
- As computational costs decrease, it will be possible to apply more complex algorithms, even in real time. These algorithms will take into account system nonlinearities or provide online optimisation of the system's performance.
- New fields of application will be addressed. Interest is now being expressed, beyond that in "classical" technical systems and processes, in environmental systems or medical and bioengineering applications.
- The boundaries between software and hardware design are being eroded. New design methods will include co-design of software and hardware and even of sensor and actuator components.
- Automation will not only replace human operators but will assist, support and supervise humans so that their work is safe and even more effective.
- Networked systems or swarms will be crucial, requiring improvement of the communication within them and study of how their behaviour can be made globally consistent.
- The issues of security and safety, not only during the operation of systems but also in the course of their design, will continue to increase in importance.

The title "Computer Science meets Automation", borne by the 52nd International Scientific Colloquium (IWK) at the Technische Universität Ilmenau, Germany, expresses the desire of scientists and engineers to rise to these challenges, cooperating closely on innovative methods in the two disciplines of computer science and automation.

The IWK has a long tradition going back as far as 1953. In the years before 1989, a major function of the colloquium was to bring together scientists from both sides of the Iron Curtain. Naturally, bonds were also deepened between the countries from the East. Today, the objective of the colloquium is still to bring researchers together. They come from the eastern and western member states of the European Union, and, indeed, from all over the world. All who wish to share their ideas on the points where "Computer Science meets Automation" are addressed by this colloquium at the Technische Universität Ilmenau.

All the University's Faculties have joined forces to ensure that nothing is left out. Control engineering, information science, cybernetics, communication technology and systems engineering – for all of these and their applications (ranging from biological systems to heavy engineering), the issues are being covered.

Together with all the organizers I should like to thank you for your contributions to the conference, ensuring, as they do, a most interesting colloquium programme of an interdisciplinary nature.
I am looking forward to an inspiring colloquium. It promises to be a fine platform for you to present your research, to address new concepts and to meet colleagues in Ilmenau.



Professor Peter Scharff
Rector, TU Ilmenau

Professor Christoph Ament
Head of Organisation

Table of Contents

C O N T E N T S

10 Mobile Communications

11 Education in Computer Science and Automation

52nd Internationales Wissenschaftliches Kolloquium
Technische Universität Ilmenau
10 – 13 September 2007

A. Jutman

On LFSR Polynomial Calculation for Test Time Reduction

ABSTRACT

Testing of nanoscale semiconductor devices, nowadays, faces a strong shift from classical methodologies towards self-test, self-diagnosis, and self-repair solutions – all to improve manufacturability and reliability characteristics of the final product. Due to these efforts, it is possible to keep production costs at reasonable levels when moving to finer and less reliable manufacturing technologies. Linear Feedback Shift Registers (LFSR) and other Pseudo-Random Pattern Generators (PRPG) have become one of the central elements used in embedded testing and diagnosis of contemporary complex electronic systems like processors, controllers, and high-performance integrated circuits. Current paper presents a mathematical framework of LFSR polynomial calculation for fault coverage improvement and test cost reduction. The proposed technique allows shortening test runtimes and increasing fault coverage by embedding specific pre-calculated test patterns into the PRPG sequence.

INTRODUCTION

Accordingly to the International Technology Roadmap for Semiconductors (ITRS) [1], the increasing complexity of recent VLSI circuits and transition to multi-core System-on-Chip (SoC) and Network-on-Chip (NoC) paradigms has made testing (including planning, test generation and scheduling) one of the most complicated and time-consuming problems in the domain of digital design.

During the last several years, ITRS reports indicate that the semiconductor manufacturing industry is inevitably moving towards test compression and self-testing approaches that allow either to efficiently feed test data to individual system cores or to initially design self-testable cores. The main driving force of such a transition is the bandwidth gap between the I/O frequency and very high internal clock rates of modern semiconductor devices.

As the result, built-in self-test (BIST) becomes widely recognized as one of the promising approaches for testing modern nanoscale devices – especially SoC-s. Unlike external test pattern generators, embedded test facilities have a good access to any internal core or unit and can be used for self-test and self-repair purposes. Moreover, they can work at the same speed as the system providing detection of defects, which do not manifest themselves at lower clock rates.

Pseudo-random test pattern generation (TPG) techniques are the main instrument used in BIST. Linear feedback shift registers (LFSR) represent the simplest and most commonly used pseudo-random TPG (PRPG) hardware. However, in terms of fault coverage and testing runtimes, the efficiency of an LFSR is far from optimum. A test

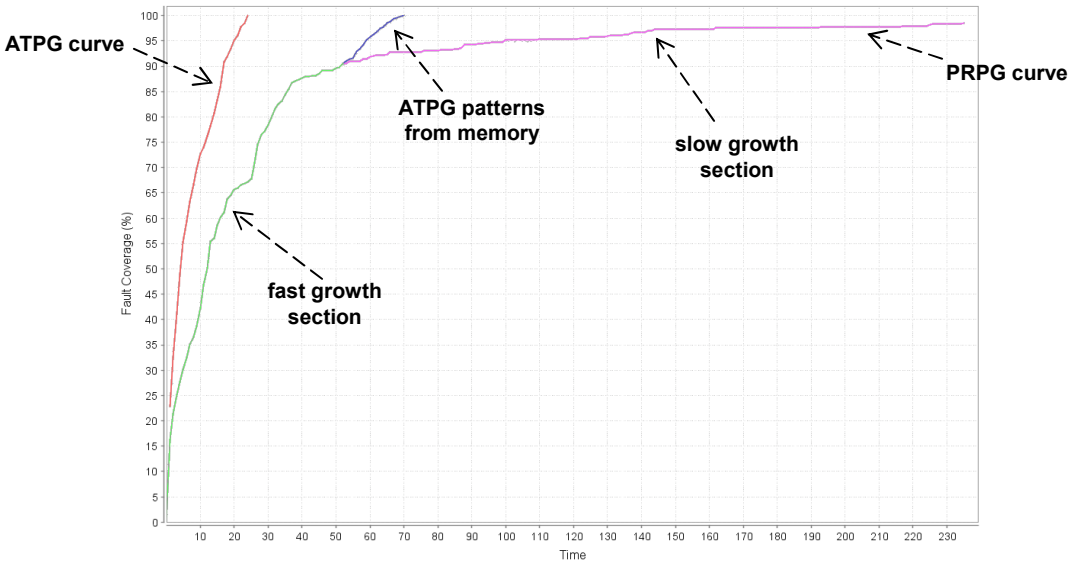


Figure 1. PRPG patterns and Fault Coverage trends

generated by an LFSR is usually an order of magnitude longer (see Fig. 1) than the test that would be calculated externally by a model-based automated test pattern generator (ATPG). In general, PRPG fault coverage trend is characterized by such peculiarities like fast initial growth and too long time to complete. Figure 1 illustrates this fact clearly showing two sections of PRPG curve. The slow growth section of PRPG curve in Figure 1 is mostly caused by existence of hard-to-test faults (HTTF) which are usually very hard to handle by PRPG-based methods (see Fig. 2.a)

As the result, there are many works that target improvement of PRPG efficiency. A big portion of research is devoted to study of alternative PRPG types [2,3] that have better saturation properties compared to the one of LFSR.

Much more gain is provided, however, when combining PRPG and ATPG patterns together. For example, one can get much shorter test sequence if he applies ATPG patterns right after the breakpoint between fast and slow sections of PRPG curve (Fig. 1 and Fig. 2.b). In case of self-testing, these patterns have to be stored in memory (Hybrid BIST) [4]. Hence, the more ATPG patterns we use, the bigger the memory overhead is.

This way of combining PRPG and ATPG patterns is not the only one possible. For instance, the Bit-Flipping BIST method adds extra circuitry to PRPG outputs. This circuitry modifies selected bits of selected PRPG patterns in such a way that these modified patterns become equivalent to ATPG patterns [5] (Fig. 2.c). However, the size of the bit-flipping controller represents a serious drawback of this method. Sometimes it occupies up to 30-40% of the circuit under test (CUT) area.

Another approach, called Reseeding, is considered as a promising one [6]. It allows for generation of several PRPG sequences where each one is optimized for covering a certain portion of HTTFs (Fig. 2.d). In terms of the test time, hardware cost, and fault coverage, the efficiency of this method is very close to the one of the Hybrid BIST.

This paper proposes a novel solution where ATPG patterns are used to form initial PRPG sequence. In this sequence, HTTF patterns are initially placed as close to each other as possible (Fig. 2.e). Then a special polynomial is calculated so that this sequence becomes feasible. In its pure form, this approach does not require nor additional memory for test patterns neither extra circuitry, but still it improves HTTF coverage and test length of PRPG sequence. For achieving extreme results, this method can be even combined together with all the methods mentioned above.

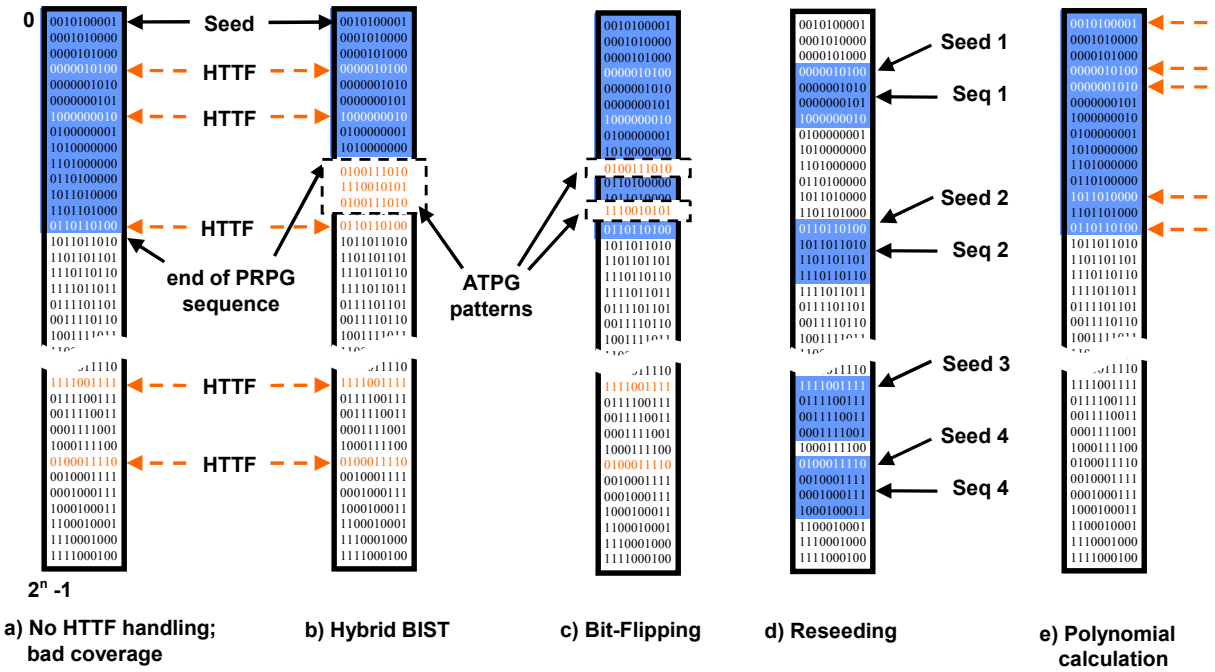


Figure 2. PRPG sequence and hard-to-test-fault (HTTF) coverage handling

DESCRIPTION OF THE METHOD

In this section we will first look deeper into LFSR structure and properties and then describe the main idea behind the method of polynomial calculation.

In Figure 3, a common internal structure of LFSR is given. It consists of D flip-flops connected in series and feedback loops collected by an XOR gate. This forms a simple shift register with a special kind of feedback. The presence or absence of the feedback loops is described by a so-called *generator polynomial*. The state of the LFSR at the beginning of test generation is determined by its initial state parameter called *seed*.

The main useful property of LFSR circuits is such that if clocked repeatedly, they go through a fixed sequence of unique states, which has a number of explicit properties of randomness and can be used, therefore, as a TPG in a BIST scheme [7]. The maximum number of such unique states is $(2^n - 1)$, where n is the length of the LFSR (i.e. no. of flip-flops). However, the actual length of this sequence depends on selected polynomial and seed. Figure 3 shows a configured LFSR and the sequence it generates. One can see that the sequence has 4 unique patterns only and the grey patterns are just a repetition of black ones. Hence, polynomial and seed have direct influence to the resulting test quality and, therefore, they play an important role in TPG. In most of modern approaches, a fully configured LFSR is used. Its configuration is based on a

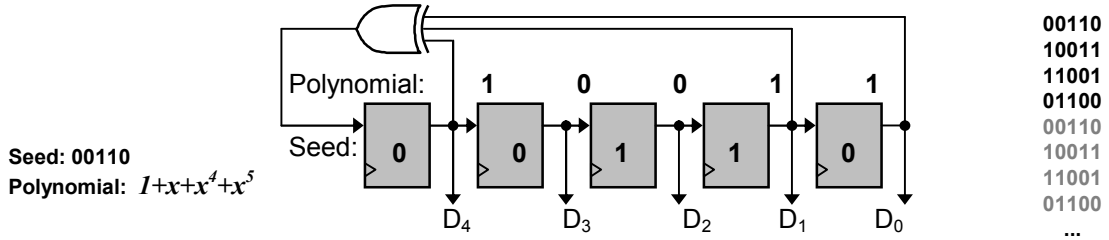


Figure 3. Internal structure of LFSR; the sequence it generates

primitive polynomial – a polynomial that guarantees $(2^n - 1)$ unique states.

In our work, we neglect this common practice due to following reasons. At first, a longer LFSR with a big number of flip-flops can generate enough unique patterns to fully test a CUT even if it is not based on a primitive polynomial. Secondly, our method allows calculating a good polynomial that targets most of HTTFs by a possibly shorter sequence.

The overall strategy of the proposed method is the following:

1. Define set of HTTFs;
2. Select ATPG patterns for given HTTFs;
3. Form a continuous sequence;
4. Calculate polynomial;
5. Let configured LFSR run until all the rest faults (non-HTTF) covered.

The first and the second steps can be done in many different ways. The choice of a particular method depends on many factors and it is, therefore, left behind the scope of the current paper. Both steps are preparational for the main phase, that is calculation of sequence and polynomial. However, the efficiency of the proposed method depends dramatically on the choice of proper HTTFs and the way the corresponding ATPG patterns were generated or selected. Selection of most difficult HTTFs and usage of patterns with don't cares is highly recommended. The method limitations with respect to these facts are further considered in the next section.

The novelty of the method mainly lies in the next two steps of the procedure: forming a continuous feasible LFSR sequence and calculation of the corresponding polynomial.

The former task means adaptation of existing patterns into a form that in principle can be generated by an LFSR. We will look at this task deeper.

If LFSR of size n generates a sequence of m states, then this sequence can be represented as a binary m-by-n Toeplitz matrix A with repeating entries a_i, where $1 \le i \le k$ and $k = n + m - 1$.

$$A = \begin{bmatrix} a_n & a_{n-1} & \dots & a_3 & a_2 & a_1 \\ a_{n+1} & a_n & \dots & a_4 & a_3 & a_2 \\ a_{n+2} & a_{n+1} & \dots & a_5 & a_4 & a_3 \\ \dots & \dots & & \dots & \dots & \dots \\ a_{n+m-1} & a_{n+m-2} & \dots & a_{m+2} & a_{m+1} & a_m \end{bmatrix}$$

The LFSR seed is represented by the first row of the matrix $(a_n \dots a_2\, a_1)$. The polynomial is not directly visible but it can be calculated by solving a system of linear equations. This task is solved in the next step of our procedure.

Let us denote LFSR feedbacks by vector $(x_1, x_2 \dots x_n)$, where $x_j = 1$ if the corresponding feedback exists, otherwise $x_j = 0$. From the definition of the LFSR structure, we can derive a system of $m-1$ linear equations that describes the LFSR operation.

$$\begin{array}{lllllll} a_n x_1 & \oplus & a_{n-1} x_2 & \dots & a_2 x_{n-1} & \oplus & a_1 x_n = a_{n+1} \\ a_{n+1} x_1 & \oplus & a_n x_2 & \dots & a_3 x_{n-1} & \oplus & a_2 x_n = a_{n+2} \\ a_{n+2} x_1 & \oplus & a_{n+1} x_2 & \dots & a_4 x_{n-1} & \oplus & a_3 x_n = a_{n+3} \\ \dots & & \dots & & \dots & & \dots \\ a_{n+m-2} x_1 & \oplus & a_{n+m-3} x_2 & \dots & a_m x_{n-1} & \oplus & a_{m-1} x_n = a_{n+m-1} \end{array}$$

This set of equations can be unambiguously constructed from the matrix A. In other words, if we have a sequence of LFSR states of sufficient length (in general: $m \ge n$) we can unambiguously calculate the generator polynomial of that LFSR by solving the corresponding system of linear equations. Despite the fact that the system is based on modulo-2 operations, we still can use Gaussian elimination to solve it. For faster results, special techniques developed for Toeplitz matrices can be used, e.g. Levinson

recursion, which complexity is $O(n^2)$.

In practice, in order to construct LFSR sequence from given ATPG patterns, one has to put them into a bit-stream of length k: $(a_k\ a_{k-1}\ \ldots\ a_1)$, then construct a system of linear equations and solve it in order to find variables $(x_1, x_2\ \ldots\ x_n)$ that represent the target LFSR polynomial.

Let us consider the small example in Figure 4. The leftmost part of this figure (4.a) shows the ATPG patterns to be processed. Here we assume that these patterns were generated for selected HTTFs (steps 1 and 2). Due to don't care bits, these patterns can be combined into a bit-stream of length *11* (Fig 4.b) and being then consequently transformed into a *7*-pattern LFSR sequence and a *6*-by-*5* matrix in Figure 4.d. The solved system of linear equations and the hardware implementation of this solution are given in Figure 4.e and Figure 5 correspondingly.

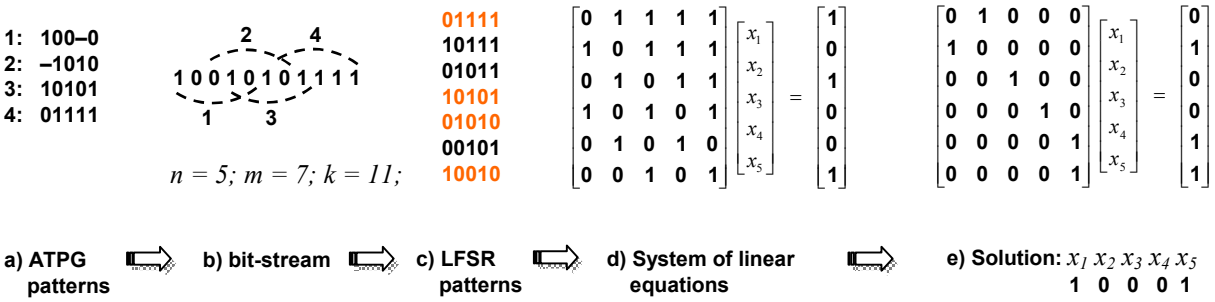


Figure 4. Solution flow: from ATPG to LFSR

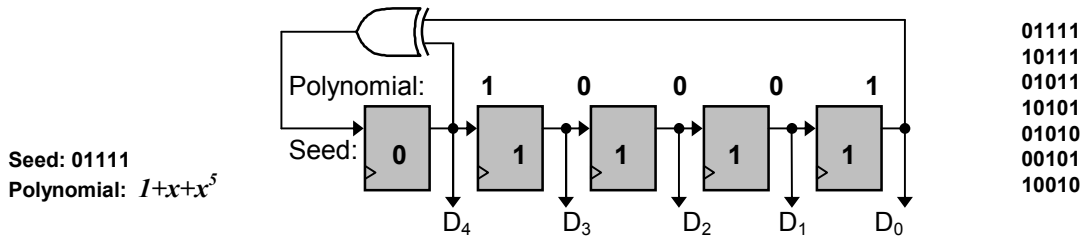


Figure 5. Hardware implementation of the solution from Figure 4

LIMITATIONS AND APPLICABILITY OF THE METHOD

Due to general properties of linear equation systems, a solution exists if rank of the extended matrix is equal to the rank of the main matrix. In our case, if this rank is also equals n, a single solution exists. In practice, it doesn't make any problem if rank is less than n. Then, one just has to choose a solution with lower hardware cost. The problem arises if the system of linear equations is inconsistent. This might happen if m is larger than n. Statistically, the expectation value for the border between system consistency and inconsistency is close to n. As the result, the polynomial calculation problem will reliably have a solution for input data that is compactable to a bit-stream of length $k \leq 2n$. If this is not the case, one either has to increase n or change the ATPG vectors. Since LFSR length increase, in general, is not desirable, a careful selection of target HTTF set and existence of don't-care bits in ATPG vectors are very important.

The *7*-pattern LFSR sequence in Figure 4.c contains 4 ATPG vectors and 3 extra ones. Since in practice a test has to cover much more faults than just some HTTFs, the

existence of these extra vectors is beneficial for they are covering additional faults. In general, LFSR can be let generating even a longer test sequence for the sake of getting higher fault coverage. This fact can be also used in cases when the system of equations becomes inconsistent. Then such a solution should be found, that satisfies the largest part of the system. HTTFs left aside as the result of this simplification will be covered later in the LFSR sequence. Such an approach will still help to reduce the overall test time since it puts at least some of HTTF test vectors into the beginning of the test sequence.

There are other ways to cope with this issue. The simplest one is to split the system of linear equations into several consistent subsystems of length $h \leq n$. Each such subsystem has to be solved then separately producing a separate polynomial. This would result in more complicated multiple-polynomial LFSR hardware that, similarly to reseeding, has to change its configuration at each h^{th} step [6]. Instead of changing the state it would change the polynomial.

Another solution would be to correct the inconsistent equations using a correcting vector matrix. In practice, such a correcting vector can be implemented similarly to the bit-flipping hardware. The difference is such that in the proposed method a single bit needs to be flipped instead of flipping arbitrary bits in the test sequence as in case of [5].

The proposed method of polynomial calculation can be especially beneficial when combined with other techniques like e.g. reseeding or hybrid BIST.

CONCLUSIONS

This paper presents a novel theoretical approach to LFSR polynomial calculation for the purpose of test time reduction through fitting ATPG vectors into LFSR test sequence. The method allows to improve fault coverage characteristics of LFSR-based testing without using additional hardware resources. As the result, the method can be used together with other existing approaches of combining LFSR sequence with ATPG patterns (reseeding, bit-flipping, multiple-polynomial, etc.) for improving their efficiency.

Acknowledgement:
This work has been supported by the Estonian Science Foundation grants 5910 and 7068, Estonian IT Foundation grant 07-03-00-05, and Enterprise Estonia funded ELIKO project.

References:

[1] *The International Technology Roadmap for Semiconductors, 2006 Update: Test and Test Equipment.* http: //public.itrs.net/
[2] G. Mrugalski, J. Rajski, J. Tyszer, “Cellular Automata–Based Test Pattern Generators with Phase Shifters,” in *IEEE Trans. on CAD/ICAS*, vol. 19, no. 8, pp. 878–893, 2000.
[3] S. Chidambaram, D. Kagaris, D. K. Pradhan, "A Comparative Study of CA with Phase Shifters and GLFSRs," in *Proc. IEEE International Test Conference (ITC)*, Austin, USA, Nov 8-10, 2005, pp. 926-935.
[4] G. Jervan, Z. Peng, R. Ubar, O. Korelina, “An improved estimation methodology for hybrid BIST cost calculation,” in *Proc. Norchip Conference*, Oslo, Norway, Nov 8-9, 2004, pp. 297 – 300,
[5] H. Wunderlich, G. Kiefer, "Bit-Flipping BIST," in *Proc. Int. Conf. on Computer-Aided Design*, Nov 10-14, 1996, pp. 337-343.
[6] A. A. Al-Yamani, S. Mitra, E. J. McCluskey, “BIST reseeding with very few seeds,” in *Proc. 21st IEEE VLSI Test Symposium (VTS)*, Napa Valley, USA, Apr./May 2003, pp. 69–74.
[7] M. Bushnell, V. Agrawal, *Essentials of Electronic Testing for Digital Memory & Mixed Signal VLSI Circuits*, Kluwer Academic Publishers, 2000, 712 p.

Author:
Dr. Artur Jutman
Tallinn University of Technology
Raja 15, 12618 Tallinn, ESTONIA
Phone: +372 620 22 63
Fax: +372 620 22 53
E-mail: artur@pld.ttu.ee