Volume 01, No. 01

International Journal of Scientific Engineering and Technology Research

Jul-Dec 2012, P.P. 67-74

www.semargroups.org

Synthesis Techniques for Pseudo-Random Built-In Self-Test Based on the LFSR

S.SRAVANTHI[1], C. HEMASUNDARA RAO[2]

[1]*M.Tech Student of* CMRIT, Ranga Reddy(Dt), AP-India,e-mail: sakamurisravanthi@gmail.com,

[2]Professor, ECE Dept, CMRIT, Ranga Reddy(Dt), AP-India.

Abstract: The structure of test system based on application built-in self-test (BIST) circuitries has been proposed. The main idea is oriented on minimization of hardware overheads and dealt with automatization of BIST-circuitries generation. Test generator based on linear feedback shift register (LFSR) provides two types of testing pseudorandom and deterministic. The proposed modified Berlekamp–Massey algorithm is used for generation the LFSR polynomial coefficients. The experimental results of technique application for some ISCAS'89 benchmark circuits have been shown. The entire design is modelled using Verilog language and simulation is done using Xilinx ISE 12.1 tool and synthesis is done using XST synthesis tool.

Keywords: *IC test, BIST, LFSR, MISR, polynom synthesis.*

I. INTRODUCTION

Both the development of up-to-date integrated technologies and the increasing complexity of designed electronic devices provide the growth of testing process complication. The testing ensures required reliability and quality of produced electronic devices and systems. But testing has influence on growth both product cost and time-to-market. The use of design-for-testability and in particular built-in selftest (BIST) approach should allow reducing time and cost expenses. The BIST-approach relies on realization special testing subcircuits, providing test pattern generation (TPG) and output responses analysis, on the same chip together with original circuit. One of the efficient solutions for TPG realization and compact representation of output signals is LFSR (Linear Feedback Shift Register) or MISR (Multiple Input Shift Register). The LFSR/MISR is more efficient in comparison with simple binary counters because requires less combinational logic per one flip-flop and can works on higher frequencies. The generated output signals of LFSR can be considered as pseudorandom. The structure of test system based on LFSR includes generator of test signals, signature analyzer, test controller, and also register with internal or external "gold" signature and comparator. All elements of structure are realized as separate modules and interact with circuit under test using primary inputs and outputs. The structure of test system based on LFSR/MISR is represented in Fig. 1.

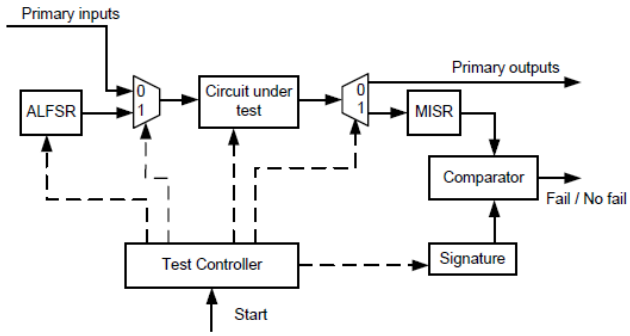

Fig. 1. The structure of test system

The LFSR without external input which called as Autonomous LFSR (ALFSR) is used for test generator (TG) construction. The signals of feedback are sum by modulo 2 and apply to LFSR input (Fig. 2). The generated output signals are repeated periodically. The maximum number of different state is equal to $2^n - 1$, where n – is a LFSR digit capacity.

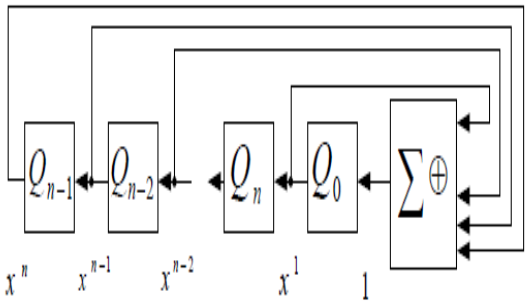

Fig. 2. Structure of Autonomous LFSR (ALFSR)

The LFSR is described by characteristic polynomial as following:

$$P_n(X) = h_n x^n + h_{n-1} x^{n-1} + h_{n-2} x^{n-2} + \ldots + h_1 x + h_0 = \sum_{i=0}^{n} h_i x^i \,, \quad (1)$$

where n is a LFSR digit capacity; h_i the coefficient of feedback signal presents, equal to 1, if the i-th stage has feedback, and 0 in opposite case.

Characteristic polynomial may be described in three possible ways:

- irreducible polynomial, which cannot be split on polynomials of less exponent;
- reducible polynomial, which can be represented by a product of simple irreducible polynomials;
- primitive polynomial, is irreducible polynomial of exponent n, which divides polynomial $x^{2^n-1}+1$ without remainder.

In practice the characteristic polynomial should be used in order to provide reliable operation of LFSR/MISR and increase the period of output signal repeating. The modification of Berlekamp–Massey algorithm is proposed to use for generation the LFSR/MISR polynomial Co-efficients. The resulting polynomial generates specified sequence of signals of length n or n working cycles. Important feature of generated polynomial consists in the fact what exponent and consequently digit capacity of obtained polynomial is minimal and less or equal n/2.

II. MODIFIED BERLEKAMP–MASSEY ALGORITHM FOR SYNTHESIS LFSR/MISR

A linear feedback shift register (LFSR) is a shift register whose input bit is a linear function of its previous state. The most commonly used linear function of single bits is XOR. Thus, an LFSR is most often a shift register whose input bit is driven by the exclusive-or (XOR) of some bits of the overall shift register value. The initial value of the LFSR is called the seed, and because the operation of the register is deterministic, the stream of values produced by the register is completely determined by its current (or previous) state. Likewise, because the register has a finite number of possible states, it must eventually enter a repeating cycle. However, an LFSR with a well-chosen feedback function can produce a sequence of bits which appears random and which has a very long cycle.

An LFSR is one of a class of devices known as state machines. The contents of the register, the bits tapped for the feedback function, and the output of the feedback function together describe the state of the LFSR. With each shift, the LFSR moves to a new state. (There is one exception to this -- when the contents of the register are all zeroes, the LFSR will never change state.) For any given state, there can be only one succeeding state. The reverse is also true: any given state can have only one preceding state. For the rest of this discussion, only the contents of the register will be used to describe the state of the LFSR.

A state space of an LFSR is the list of all the states the LFSR can be in for a particular tap sequence and a particular starting value. Any tap sequence will yield at least two state spaces for an LFSR. (One of these spaces will be the one that contains only one state -- the all zero one.) Tap sequences that yield only two state spaces are referred to as maximal length tap sequences.

The state of an LFSR that is n bits long can be any one of 2^n different values. The largest state space possible for such an LFSR will be 2^n - 1 (all possible values minus the zero state). Because each state can have only once succeeding state, an LFSR with a maximal length tap sequence will pass through every non-zero state once and only once before repeating a state.

One corollary to this behavior is the output bit stream. The period of an LFSR is defined as the length of the stream before it repeats. The period, like the state space, is tied to the tap sequence and the starting value. As a matter of fact, the period is equal to the size of the state space. The longest period possible corresponds to the largest possible state space, which is produced by a maximal length tap sequence.

The modification is based on realization of the Berlekamp Massey algorithm described in [8]. The algorithm is used for construction linear feedback shift register with minimal length, which generates specified

binary sequence. The algorithm takes n iterations for n-bit sequence and at N-th iterations (N < n) the LFSR polynom generating first N elements of sequence is calculated.

Definition. Lets s^{N+1} is a final binary sequence $s^{N+1} = s_0, s_1, \ldots, s_{N-1}, s_N$. Lets L,C(D) is LFSR described by polynom $C(D) = 1 + c_1 D + \ldots + c_L D^L$ and generating binary sequence $s^N = s_0, s_1, \ldots, s_{N-1}$. The next difference d_N is difference between s_N and (N+1)-th sequence element, generated by LFSR:

$$d_N = (s_N + \sum_{i=1}^{L} c_i s_{N-i}) \bmod 2. \qquad (2)$$

Lets $s^N = s_0, s_1, \ldots, s_{N-1}$ is final binary sequence and $\langle L, C(D) \rangle$ is LFSR generated this sequence, where L is LFSR length (L is linear complexity of sequence). Then $\langle L, C(D) \rangle$ generates sequence $s^{N+1} = s_0, s_1, \ldots, s_{N-1}, s_N$ if and only if the next difference d_N is equal to 0;

If $d_N = 0$, then linear complexity $L(s^{N+1}) = L$; Suppose d_N. Lets m is the largest integer less N, such what $L(s^m) < L(s^N)$. Lets $\langle L(s^m), B(D) \rangle$ is LFSR by length $L(s^m)$ generating s^m. Then $\langle L', C'(D) \rangle$ is LFSR by minimal length generating s^{N+1}, where

$$L' = \begin{cases} L, & L > N/2 \\ N + 1 - L, & L \le N/2 \end{cases}, \qquad (3)$$

and $C'(D) = C(D) + B(D) \cdot D^{N-m}$. Besides a minimal length there is additional requirement to test generator based on LFSR polynom describing LFSR should be primitive one. The structure of modified algorithm for synthesis LFSR/MISR is represented in Fig. 3. The CAD subsystem of TG realization based on the Berlekamp Massey algorithm has been prepared using C++ [6]. The checking of condition, what calculated polynom is primitive, is provided in software. The input data for software is represented by text-file with the following structure: first line contains the number of elements in sequence, second line contains the binary sequence.

For instance:

9

001101110

The output data is represented by text-file, where first line contains the degree of polynom, second line contains polynom as binary sequence and third line contains indicator of primitive polynom condition checking.

For instance:

Polynom degree: 5

1 0 0 1 0 1

Primitive

The sequence in example above corresponds to the follows primitive polynom $1 + D^3 + D^5$.

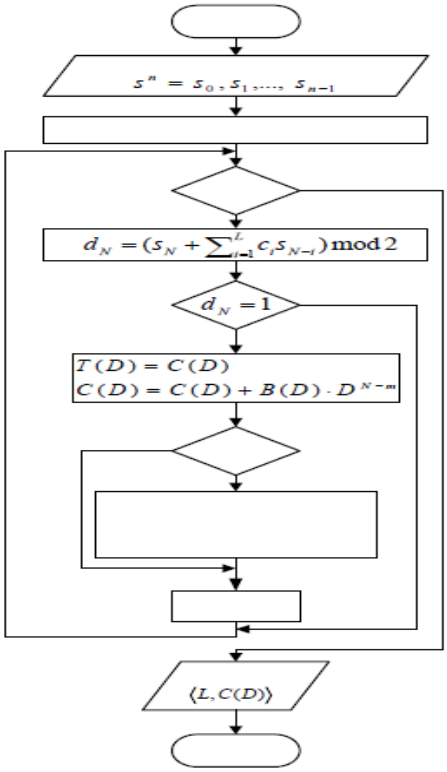

Fig.3. Modified Berlekamp–Massey algorithm for LFSR/MISR generation.

The experiment was provided for different input binary sequences. Some results are presented below:

Input: 12

001011110010

Output: Polynom degree: 6

1 1 0 0 0 0 1

Primitive

Input: 14

00101001001011

Output: Polynom degree: 7

1 0 0 0 1 1 1 1

Primitive

Input: 16

0010110100001101

Output: Polynom degree: 10

1 1 1 1 0 0 1 0 0 1 1

Primitive

Input: 16

1011011100011101

Output: Polynom degree: 8

1 1 1 1 0 0 0 0 1

Primitive

Input: 20

10110111000111010010

Output: Polynom degree: 10

1 0 0 0 0 0 1 0 1 1 1

Primitive

Calculated LFSR was described in VHDL and simulated in Mentor Graphics ModelSim tool. The results have shown absolute correspondence between generated sequence and input binary sequence.

The overwhelming majority of calculated polynoms is primitive one. Consequently, if the degree of polynom is equal L, then corresponding register is able to generate 2L patterns by width L. The length of sequence used for LFSR generation at the average is equal 2L. The rest (2L – 1) – L patterns can be used for pseudorandom testing. Thus, generated LFSR can be applied both for deterministic and pseudorandom testing without use additional combinational logic.

III. TECHNIQUE OF LFSR/MISR GENERATION

The process of LFSR generation with using the Berlekamp–Massey algorithm can be described by two main stages: selection of test generator (TG) structure for deterministic test sequence and LFSR generation [7].

LFSR generators produce what are called linear recursive sequences (LRS) because all operations are linear. Generally speaking, the length of the sequence before repetition occurs depends upon two factors, the feedback taps and the initial state. An LFSR of any given size m (number of registers) is capable of producing every possible state during the period $N=2^m-1$ shifts, but will do so only if proper feedback taps have been chosen. For example, such an an eight stage LFSR will contain every possible combination of ones and zeros after 255 shifts. Such a sequence is called a maximal length sequence, maximal sequence, or less commonly, maximum length sequence. It is often abbreviated as m-sequence. In certain industries m-sequences are referred to as a pseudonoise (PN) orpseudorandom sequences, due to their optimal noise-like characteristics. (Informally, even non-maximal sequences are often called pseudonoise or pseudorandom sequences.)

Technically speaking, maximal length generators can actually produce two sequences. The first--the trivial one--has a length of one, and occurs when the initial state of the generator is set to all zeros. (The generator simply remains in the zero state indefinitely.) The other one--the useful one--has a length of 2^m-1. Together, these two sequences account for all 2^m states of an m-bit state register. When the feedback taps of an LFSR are non-maximal, the length of the generated sequence depends upon the initial state of the LFSR. A non-maximal generator is capable of producing two or more unique sequences (plus the trivial all-zeros one), with the initial state determining which is produced. Each of these sequences is referred to as a state space of the generator. Together, every non-maximal sequence the generator can produce accounts for all 2^m states of an m-bit state register.

Properties of non-maximal sequences are generally inferior to those of maximal sequences. So the use of non-maximal sequences in real systems is usually avoided in favor of their maximal-length counterparts.

1) Deterministic sequence of test patterns can be defined using any well-known methods of test generation for sequential circuits [9]. The obtained test sequence is represented as matrix, which contains binary test patterns (Table 1).

TABLE 1

REPRESENTATION OF TEST SEQUENCE

Number of test pattern	Bit position in test pattern			
	0	1	…	*p*-1
0	1	0	…	1
1	0	0	…	0
2	1	1	…	1
…	…	…	…	…
n-1	0	0	…	1

There are two main ways to generate test patterns: sequential and parallel. The first way relies on the use only one LFSR generated each next pattern after p steps, where p is a number of bits in a test pattern. During parallel way the test patterns are generated at each step of TG functioning, which consists of LFSRs.

Sequential way of test pattern generation: Test sequence is specified in line (Table 2) and applied to processing in CAD subsystem for LFSR polynom calculation.

TABLE 2

LINEARIZED TEST SEQUENCE

Number of test pattern				
0	1	2	…	*n*-1
10…1	00…0	11…1	…	00…1

The set of coefficients for the LFSR polynom are calculated in results of subsystem execution. The length of polynom will be equal to the number of bits in test pattern. The initial bits of test sequence initiate the register.

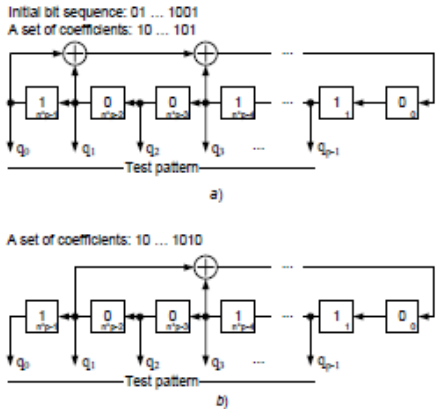

Fig. 4. Sequential test generator based on LFSR: a) MSB = 1, b) MSB = 0

A LFSR described by polynom with most significant bit equal 1 is shown in Fig. 4, a and a LFSR described by polynom with most significant bit equal 0 is represented in Fig. 4, b.

Parallel way of test pattern generation: Here the independent LFSR is used for generation each bit of a test pattern. The columns of Table 1 are used as input sequence for polynoms calculation of each LFSR. The structure of parallel test generator is represented in Fig. 5.

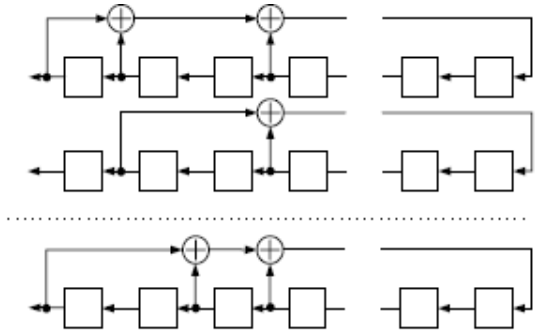

Fig. 5. Parallel test generator

Both parallel and sequential test generators use at the average the same number of flip-flops. If the length of test pattern is equal p and length test sequence is equal to n, then length of LFSR is (n - p)/ 2 .

The length of each LFSR of parallel generator is n / 2 . Consequently the same (n - p) 2 flip-flops will be required for realization of p parallel-working LFSR. Both types of test generator (sequential and parallel) are used in practice.

IV. IMPLEMENTATION OF TEST CIRCUITRIES AND SIMULATION RESULTS

Proposed technique was put on trial for benchmark circuits from ISCAS'89 [10]. The following frequency characteristics have been obtained for different implementation of BIST-circuitries (LFSR as well as BILBO – Built-In Logic Block Observer) in the basis of FPGA using proposed technique. The synthesis has been done for Xilinx FPGA Virtex-5 XC5VSX50T (Table 3):

TABLE 3

THE RESULTS OF SYNTHESIS FOR FPGA

Circuit	Maximum frequency, MHz
s386 + BILBO	482.207
s386 + LFSR	456.642
s386	815.661
s27 + BILBO	442.713
s27 + LFSR	563.222
s27	966.464

Test circuitries have been implemented also in the basis of standard cells using CMOS 0.35 um integrated technology. The Mentor Graphics CAD tools were used for ASIC standard cells synthesis. The aggregate results of synthesis for ISCAS'89 circuits s27 and s386 with two possible realization of test circuitries (LFSR and BILBO) are represented in Tables 4 and 5.

TABLE 4

THE RESULTS OF S27 SYNTHESIS

Parameters	*s27*	*s27+LFSR*	*s27+BILBO*
Number of pins	34	19	34
Number of elements	118	158	333
Die area (um^2)	13013	24861	45573

TABLE 5

THE RESULTS OF S386 SYNTHESIS

Parameters	*s386*	*s386+LFSR*	*s386+BILBO*
Number of pins	26	20	23
Number of elements	165	269	528
Die area (um^2)	27018	46010	81146

The results of test circuitries synthesis show the reducing maximum working frequency of original circuits, increasing both a number of components and a die area. The essential increasing a complexity and a die area of circuit deals with simplicity of original circuits, when initial number of components is proportional to a number of components in test circuitries. Test generators for both circuits generate test patterns which provide coverage 100 % s-a faults using only deterministic test. The structure of generators in both cases is optimal and compact.

Comparing results of test circuitries implementation in basis LFSR and BILBO the following conclusion can be obtained – the efficiency of LFSR for simple circuits is higher in contrast to a BILBO. But BILBO architecture more efficient for test generator and signature analyzer implementation than LFSR/MISR architecture for complex circuits which contain many flip-flops and also in the case when original circuits can be split on two subcircuits and their testing is realized independent by changing working mode of BILBO-blocks.

V. SIMULATION RESULTS

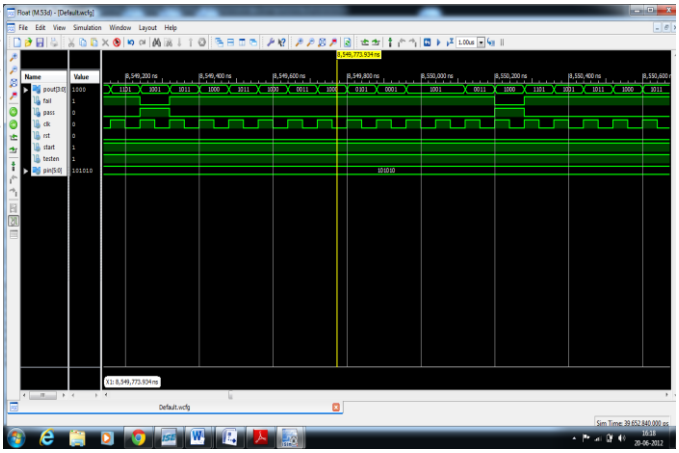

Fig6. Output Waveform of Top level Module BIST

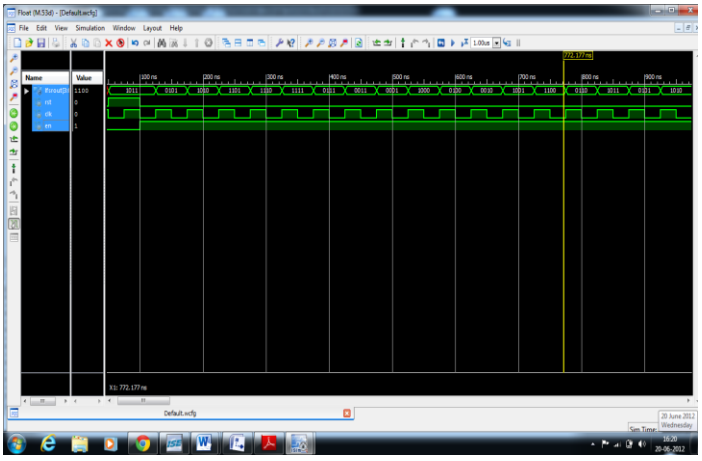

Fig6. Output Waveform of LFSR

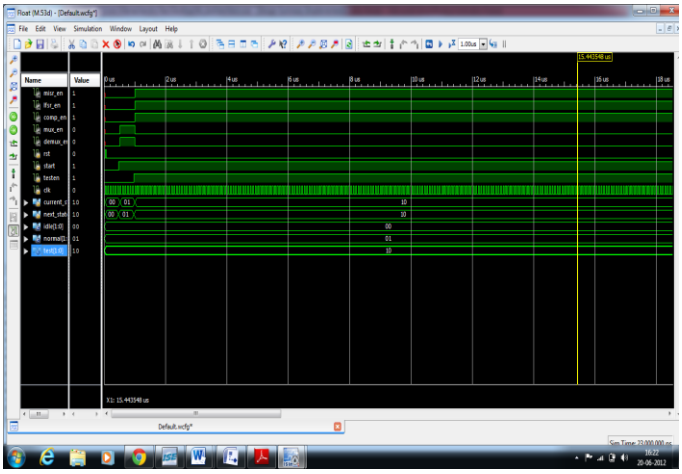

Fig7. Output waveform of controller

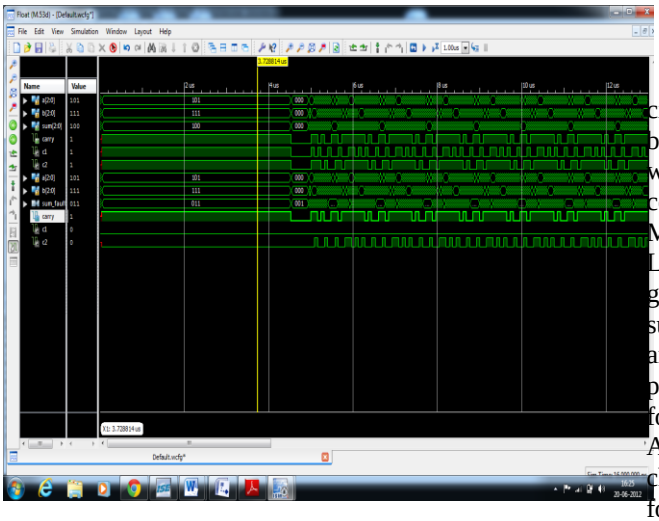

Fig8. Output Waveforms of Cut-Fault free and Cut Fault

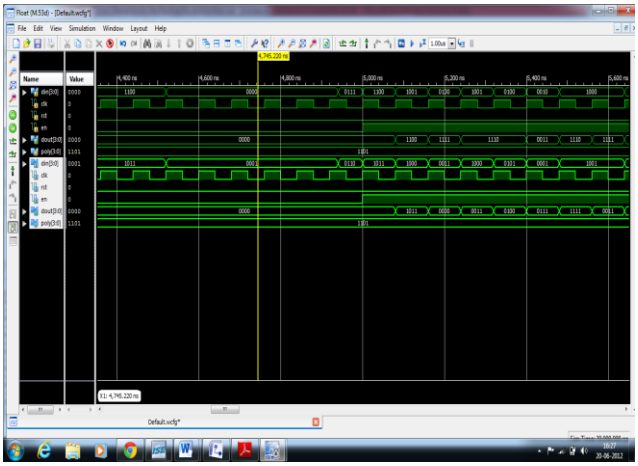

Fig9. Output Waveforms of MISRS

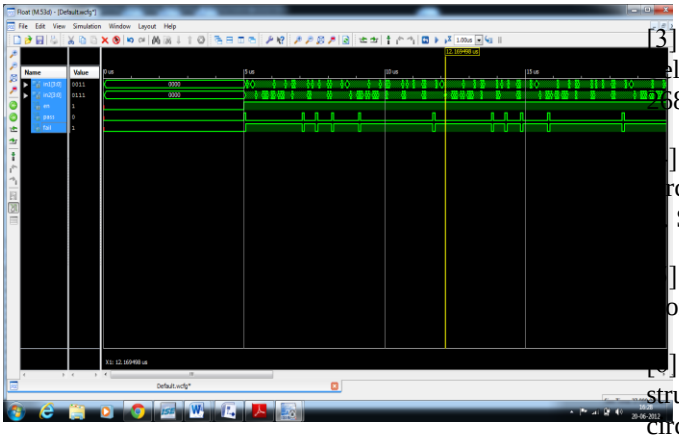

Fig10. Output waveform of Comparator

VI. CONCLUSIONS

Proposed technique of realization optimal BIST circuitries allows to implement test generator combining both deterministic and pseudorandom test approaches without changing structure or inclusion some additional components in TG circuit. The modified Berlekamp–Massey algorithm is suggested for calculation optimal LFSR polynom providing implementation of test generator. The technique has been realized as a CAD subsystem supporting the design-for-testability of IC and electronic devices. Practical experiments indicate possibility to use the proposed technique and algorithms for circuits which are implemented in both FPGA or/and ASIC basis. But BIST circuitries can impair frequency characteristics and die area of original circuits. The following arrangements making in the framework of proposed technique allow to improve the efficiency of BIST circuitries application:

→ To generate the deterministic test sequence only for some subset of faults, and the rest faults to detect by pseudorandom test sequence;

→ To determine the best order of test patterns generation for BILBO architecture with means to minimize the structure of test generator.

VII. REFERENCES

[1] Mosin S.G. “State-of-the-art tendencies and technologies of IC design”, Information technologies, № 1, pp. 28 – 33, 2009.

[2] Ondrej Novak, Elena Gramatova, Raimund Ubar and collective. “Handbook of testing electronic systems”. Czech Technical University Publishing House., 395 p. 2005.

[3] Janusz Rajski, Jerzy Tyszer. “Arithmetic Built-in elf-Test for Embedded Systems”. Prentice Hall PTR. 268 p. 1997.

] Morgan Kaufmann. “System-on-chip Test rchitectures”. Edited by Laung-Terng Wang, Charles Stroud, Nur A. Touba. 893 p. 2008.

] Berlekamp E. R. “Algebraic Coding Theory”. New ork: McGrow Hill, 1968.

] Chebykina N.V., Mosin S.G. “Investigation of structural solutions for built-in self-test of digital circuits”. Proc. of XVI Int. conference “Information

Systems and Technologies (IST-2010)". –N.-Novgorod, 2010. – p. 274.

[7] Mosin S., Chebykina N. "A Technique of Optimal Built-In Self-Test Circuitries Generation". Proc. of IEEE East-West Design and Test Symposium (EWDTS'2010). P. 145 – 148. 2010.

[8] A. Menezes, P. van Oorschot, S Vanstone. Handbook of Applied Cryptography. CRC Press, pp. 191-212. 1996.

[9] Bareisa E., Jusas V., Motiejunas K., Seinauskas R. "Functional Digital Systems Testing". Monografija. Kaunas Technological University. 282 p. 2006.

[10] Brglez F., Bryan D., Koiminski K. "Combinational profiles of sequential benchmark circuits" ISCAS'89: IEEE International Symposium on Circuits And Systems, Portland, OR (USA), pp. 1929-1934, May 1989.