Synthesis of a Test Generator for a Built-In Self-Test Scheme

Oleg V. Khomich ,Vyacheslav N. Yarmolik,
Department of Computer Science,
Belorussian State University of Informatic and Radioelectronic,
6, Brovki Str., Minsk, 220027, Belarus
helgi@belcaf.minsk.by, yarmolik@yvn_poit.minsk.by

Abstract

This paper presents a new algorithm for the automated synthesis of pseudo-random test patterns generators for Built-In Self Test schemes with a mixed test mode. The experimental results show an opportunity of using the given method on a design stage of circuits producing. In this paper it is shown that an appropriate selection of test pattern generator can significantly reduce the hardware requirements of deterministic part.

1. Introduction

Using the Built-In Self Test schemes is one of major methods for reliable functioning of circuits. The efficiency of Built-In Self Test is estimated by duration of the test and hardware overhead for a test scheme realization.

Nowadays most approaches to diagnosing circuits based on exhaustive testing, pseudorandom testing, weighed random testing, hardware patterns generators of deterministic tests and mixed-mode test pattern generation frequently are used [1-4]. Practically in all named methods wide application finds the Linear Feedback Shift Register (LFSR) as the generator of pseudo-random test patterns. LFSR has simple structure which requier small area overhead, and its can also be used as output response analyzer thereby serving a dual purpose.

Unfortunately, the majority of the combinational circuits contain random pattern resistant faults. In such cases for achievement a maximal fault coverage both pseudo-random and deterministic patterns are used. [5-7].

In paper[3] a scheme for Built-In Test with multiple-polynomial LFSR is offered. The presence of the mechanism for choice necessary polynomial for realization of the generator of pseudo-random patterns allows considerably improve fault overhead in comparison with use traditional LFSR as the generator. However, the management of a choice of a particular polynomial requires in this case additional hardware expenses.

In article [8] the method of updating generated pseudo-random test patterns is offered with the purpose of increase a fault coverage. This way requires the additional analysis of a test generator work, and also additional hardware expenses for realization a modifying logic.

The most effective decision of the problem represents a method consisting in updating of an initial pseudo-random test patterns, offered in work [1]. The given method uses a condition of the test generator as entrance value for the logic function which modify specified bits of pseudo-random test patterns, that allows to receive the necessary deterministic test cube.

The basic lack of known methods [1 - 4] is the increase of hardware expenses at realization of the generator of test patterns, that in some cases is inadmissible for BIST schemes.

2.Technique of BIST, based on change of generated pseudo-random test patterns

Is shown, that with using LFSR as a test generator for pseudo-random patterns the majority of generated sequences are useless for faults detection in the combinational circuit [8] . At the same time, the detection of specific fault of circuits frequently is possible only with the deterministic test cubes, which are specified depending on the tested circuit [1]. As a rule, in pseudo-random test pattern number of bits requiring change is not large in comparison with common length of a test cube. In table 1 the statistical data on expected amount of bits in a pseudo-random patterns are given, which change is necessary for reception of the deterministic test cubes, depending on length of the test and amount of specified bits [1]. For example, with length of the test in 10000 patterns and with total specified bit equal 40, is necessary to change 7.83 bits in an

initial pseudo-random test patterns, that will allow to generate the necessary deterministic test cubes and, accordingly to ensure required fault coverage of tested circuits.

Table 1

Expected number of bits to be changed

Number of patterns	Number of specified bits						
	10	20	30	40	50	60	70
1000	0.02	2.78	6.09	9.54	13.32	17.17	21.11
10000	0.00	1.79	4.66	7.83	11.39	15.03	18.74
100000	0.00	0.90	3.53	6.50	9.65	13.19	16.64
1000000	0.00	0.05	2.54	5.21	8.29	11.52	14.89

The given data are submitted in work [1], in which the effective method of the polinomial analysis is offered. The method allows to synthesize an additional logic function for updating of generated pseudo-random test patterns for receiving necessary deterministic test cubes.

More attractive then known methods of making an additional logic for both generating pseudo-random and deterministic patterns is method of the analysis of primitive polinomial, allowing to prove a presence of the deterministic test cubes in all amount of generated pseudo-random patterns, and also, when such generation is impossibal to find a primitive polinomial which can generate needed deterministic test cubes. Such analysis of the generator is possible, because the quantity of primitive polinomials of a large value of degree is rather great and is defined as L = Ö (2m - 1) /m, where Ö - Eyler function.

Thus, definition of presence of the needed test cubes with the specified bits in all amount of generated test patterns, and also, if necessary, find satisfying to the certain requirements primitive polinomial, is a solution of a problem of maximal fault coverage.

In the article the method of the primitive polinomial analysis for generator of pseudo-random test pattern is presented. The method allows to define an opportunity to generate of deterministic test patterns in all set of pseudo-random test cubes.. Described method is possible to use for circuits on BIST scheme design stage . The advantage of the offered method consists in absence of additional hardware overhead for generation of the deterministic test cubes.

3. Algorithm of the analysis polinomial for the test generator

The initial data for search of start conditions of the test generator ensuring presence of deterministic test paterns in all test cubes of generated pseudo-random test patterns is primitive polinomial $\varphi(x)$, length of a scan chain and set of the deterministic test patterns with specified bits.

The algorithm of the analysis for the test generator contains of the following stages.

1. The test generator provides producing periodically pseudo-random test patterns. In a consequence it, for each bits of a scan chain is possible to put in conformity both bits of pseudo-random patterns and bits of the deterministic test cubes.

2. On the basis of the analysis which has been made on a step 1, systems of equations for each deterministic test patterns are made. The given systems include equations for each specified bits of test patterns. The number of systems is defined by number of the deterministic patterns, and the number of the equations in each systems is defined by number of specified bits in the appropriate test cube.

3. Find primitive polinomial for generator of test patterns received by decimation of an initial pseudo-random sequence. Index of decimation should have a condition of mutual simplicity with a period of the test generator and the length of the scan chain.

4. Each bit of an initial sequence is put in conformity with bit of decimation sequence. In view of the received relations transform systems of equations received on a step 2 to systems for decimation sequence.

5. Calculate the coefficients for reception of the shifted copies of a M-sequence for shift value equal to numbers of bits of decimation test sequence rather which the systems of the equations on a step 4 are made.

6. The systems of the equations, received on a step 4, are transformed of rather start condition of the test generator. Coefficients in the equations are the appropriate value of coefficients for the shifted copies of the M-sequence.

7. Solve the systems of equations received on a step 6. The decisions of the systems of

equations are condition of the test generator in different steps of its work. It means, that they should differ from each other. The same decision of the systems will mean impossibility to generate needed deterministic test cubes. However, in practice the situation, when one deterministic sequence will include all specified bits of other patterns, will not take place. The given fact allows do not make the analysis of the decisions of the systems of equations.

8. If there is not decision of any system other primitive polinomial should be taken with equivalent or greater degree and make its analysis since a step 3 .

As it is clear from the algorithm, most work expenses and in a greater degree efficiency of the offered analysis is the step connected to evaluate of coefficients for evaluating of a shifted M-sequence. In the following unit the high speed alghoritm of getting coefficients of the shifted copies of a M-sequence is described.

4. Calculating of coefficients for getting the shifted copies of M-sequences

As was shown above, for the analysis of primitive polinomial of the test generator ensuring presence of deterministic test patterns in all set of generated pseudo-random cubes, it is necessary to evaluate coefficients for formation of the shifted copies of M-sequences.

There are many methods of the decision the problem distinguished on the efficiency [9]. The highest speed characterizes a method using the analytical analysis of test generator work.

The functioning of the generator of test sequences can be described as follows:

(1) $$a_1(k+1) = \sum_{i=1}^{m}{}^{\oplus} \alpha_i * a_i(k);$$

(2) $$a_i(k+1) = a_{i-1}(k);\ i = \overline{2,m};\ k = 0, 1, 2, \ldots,$$

where $a_i(k) \in \{0, 1\}$ – value of register bits on a k-step of work; $m = \deg \varphi(x)$ and $\varphi(x) = 1 \oplus \alpha_1 * x^1 \oplus \alpha_2 * x^2 \oplus \alpha_3 * x^3 \oplus \ldots \oplus \alpha_m * x^m$.

Using property of shift and addition of a M-sequence, it is possible to write down expression for definition of a M-sequence value shifted on any number of steps:

(3) $$a_1(k+u) = \sum_{i=1}^{m}{}^{\oplus} \delta_i(u) * a_i(k);\ k = 0, 1, 2, \ldots$$

Where u - value of shift; $\delta_i(u) \in \{0, 1\}$- coefficients determining using of value an i-bit of the shift register for formation an element of a shifted M-sequence. From the point of view of realization the coefficient $\delta_i(u)$ set a connection topology of the bits of the shift register with additional XOR elements, on which output the shifted sequence is formed.

The method of definition coefficients $\delta_i(u)$ consists in definition of required value by the decision of some logic equations, which members are $\delta_i(h)$ and $\delta_i(s)$, where $h + s = u$. Let's assume, that the coefficients $\delta_i(h)$ and $\delta_i(u)$, allowing to receive M-sequence shifted on h and s steps, are known. Then according to (3) it is possible to write down

(4) $$a_1(k+h) = \sum_{n=1}^{m}{}^{\oplus} \delta_n(h) * a_n(k)$$

And

(5) $$a_1(k+s) = \sum_{r=1}^{m}{}^{\oplus} \delta_r(s) * a_r(k)$$

From (4) the validity of the next equality follows:

(6) $$a_1(k+h+s) = \sum_{i=1}^{m}{}^{\oplus} \delta_n(h) * a_n(k+s)$$

Where $a_n(k+s)$ with the (5) is defined

(7) $$a_l(k+s) = \sum_{p=1}^{m}{}^{\oplus} \delta_p(s) * a_p(k),\ n = 1;$$

$$a_l(k+s) = \sum_{r=1}^{m-(n-1)}{}^{\oplus} \delta_r(s) * a_{r+(n-1)}(k) \oplus$$

$$\oplus \sum_{q=1}^{n-1}{}^{\oplus} \delta_{m-(n-1)+q}(s) * a_m(k-q),\ n = \overline{2,m}$$

With the (1) and (2) elements $a_m(k-q)$ can be submitted as follows:

(8) $$a_m(k-q) = a_q(k) \oplus \sum_{c=1}^{m-q}{}^{\oplus} \alpha_c * a_{q+c}(k) \oplus \sum_{d=1}^{q-1}{}^{\oplus} \alpha_{m-q+d}(s) * a_m(k-q),\ q = \overline{1,m-1}.$$

For reception expression (8) the fact, that for anyone m and primitive polinomial $\varphi(x)$ value of coefficient $\alpha_m = 1$ was used.

As a result of substitution of meanings $a_m(k-1)$, $a_m(k-2), \ldots, a_m(k-q+1)$ in (8) is received

(9) $$a_m(k-q) = \sum_{j=1}^{m}{}^{\oplus} \beta_{j,q} * a_j(k),$$

Where $\beta_{j,q} \in \{0, 1\}$ is defined as:

(10) $$\beta_{j,q} = 1,\ \text{when } j = q = 1;$$

$$\beta_{j,q} = \sum_{u=1}^{q-1}{}^{\oplus} \alpha_{m-u} * \beta_{j,q-u},\ \text{when } j = \overline{1, m-1},\ q = \overline{j+1, m-1};$$

$$\beta_{j,q} = \alpha_{j-1}\ \text{when } j = 2, m,\ q = 1;$$

$$\beta_{j,q} = \alpha_{j-q} \oplus \sum_{u=1}^{q=1}{}^{\oplus} \alpha_{m-u} * \beta_{j,q-u},\quad \text{when } j = 1, m,\ q = 2, j-1;$$

$$\beta_{j,q} = 1 \oplus \sum_{u=1}^{q-1}{}^{\oplus} \alpha_{m-u} * \beta_{q,q-u}, \qquad \text{when } j = q = \overline{2, m-1}.$$

Consistently by substituting expression for $a_m(k-q)$ from (9) in (7) and $a_m(k+s)$ from (7) in (6) we shall receive

$$(11)\ \delta_j(h+s) = \sum_{v=1}^{j}{}^{\oplus} \delta_v(h) * \delta_{j+1-v}(s) \oplus$$

$$\oplus \sum_{q=1}^{m-1}{}^{\oplus} \beta_{j,q} \sum_{c=q+1}^{m}{}^{\oplus} \delta_c(h) * \delta_{m+q+1-c}(s), \qquad j = 1, m$$

The expression (11) shows an opportunity of definition of coefficients $\delta_i(h+s)$ on set of meanings $\delta_i(h)$ and $\delta_i(h)$. Thus the given procedure is purely analytical.

With h = s and even j the first sum in expression (11) is equal to zero, when j is odd the first sum is $\delta_{(j+1)/2}$. In the case the ratio $\delta_v(h) * \delta_{j+1-v}(h) \oplus \delta_v(h) * \delta_{j+1-v}(h) = 0$ was taken.

With even m the system of the equations for $\delta_j(2h)$, including $\delta_j(2)$, is:

$$(12)\ \delta_j(2h) = \delta_{(j+1)/2}(h) \oplus \sum_{n=1}^{m/2}{}^{\oplus} \beta_{j,2n-1}(h) * \delta_{m-1+n}(h), \quad j = 2k-1;$$

$$\delta_j(2h) = \sum_{n=1}^{m/2}{}^{\oplus} \beta_{j,2n-1}(h) * \delta_{m/2+n}(h), \quad j = 2k, \ k = 1, 2, \ldots, i = \overline{1,m}$$

For odd m we have:

$$(13)\ \delta_j(2h) = \delta_{(j+1)/2}(h) \oplus \sum_{n=1}^{(m-1)/2}{}^{\oplus} \beta_{j,2n}(h) * \delta_{(m+1)/2+n}(h), \quad j = 2k-1;$$

$$\delta_j(2h) = \sum_{n=1}^{(m-1)/2}{}^{\oplus} \beta_{j,2n}(h) * \delta_{(m+1)/2+n}(h), \quad j = 2k, \ k = 1, 2,, \ldots, i = \overline{1,m}$$

Let's note, that $\delta_j(1) = \alpha_j$, where $\alpha_j \in \{0, 1\}$- coefficients of primitive polinomial of initial M - sequence. Thus, on a basis (12) and (13), using meanings of coefficients $\delta_j(1)$ easily to calculate $\delta_j(1)$, further $\delta_j(1)$ and so on. If it is necessary to receive a copy of a M-sequence shifted on $u \neq 2^k$ of takts (k = 1, 2, 3, ...), in the beginning it is expedient to take advantage of expression (12) or (13), as having smaller computing complexity, and then to apply (11).

5.Example of the analysis a primitive polinomial of a test generator

The generator is given by polinomial $\varphi(x) = 1 \oplus x \oplus x^3$, length of a scan chain is equal to 5, the deterministic test patterns have the following kind: 1.{11xxx}; 2.{0xx1x}; 3. {11x01}.
The schema of the test generator and a scan chain is given in a fig. 1.

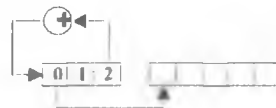

Fig. 1 The test generator and a scan chain used in the example

Step 1.According to the initial data, condition of a scan chain will be the following: 1. $\{a_4 a_3 a_2 a_1 a_0\}$; 2.$\{a_2 a_1 a_0 a_6 a_5\}$; 3. $\{a_0 a_6 a_5 a_4 a_3\}$; 4. $\{a_5 a_4 a_3 a_2 a_1\}$; 5. $\{a_3 a_2 a_1 a_0 a_6\}$; 6. $\{a_1 a_0 a_6 a_5 a_4\}$; 7. $\{a_6 a_5 a_4 a_3 a_2\}$.
Starting from various start condition, we shall receive identical patterns from seven test cubes. The choice of an initial condition will define only order of following of a test sequence in a set.

Step 2.For each deterministic patterns, being based on a condition of a scan chain, it is possible to write down the following equality:

1. {11xxx}: $a_{(4+k*5) \bmod 7} = 1$; $a_{(3+k*5) \bmod 7} = 1$;
2. {0xx1x}: $a_{(4+j*5) \bmod 7} = 0$; $a_{(1+j*5) \bmod 7} = 1$;
3. {11x01}: $a_{(4+j*5) \bmod 7} = 1$; $a_{(3+j*5) \bmod 7} = 1$; $a_{(1+j*5) \bmod 7} = 0$; $a_{(0+j*5) \bmod 7} = 1$.

Where $i \neq j \neq k \in \{0, 1, 2, \ldots, 6\}$.

Step 3.Let's examine a sequence received by decimation an initial sequence with decimation coefficient equal to length of a scan chain, in the case equal to 5: $b_i = a_{5*i \bmod 7}$. Primitive polinomial for decimation sequence has a kind: $\varphi(x) = 1 \oplus x^2 \oplus x^3$.

Step 4. $b_i = a_{5*i \bmod 7} \Rightarrow b_0 = a_0$, $b_1 = a_5$, $b_2 = a_3$, $b_3 = a_1$, $b_4 = a_6$, $b_5 = a_4$, $b_6 = a_2$.
The accordingly equality concerning a M-sequence received by decimation, will accept a kind:

1. {11xxx}: $b_{(5+k) \bmod 7} = 1$; $b_{(2+k) \bmod 7} = 1$;
2. {0xx1x}: $b_{(5+j) \bmod 7} = 0$; $b_{(3+j) \bmod 7} = 1$;
3. {11x01}: $b_{(5+j) \bmod 7} = 1$; $b_{(2+j) \bmod 7} = 1$; $b_{(3+j) \bmod 7} = 0$; $b_{(0+j) \bmod 7} = 1$.

Let's rewrite these equalities in view that any bit of a M-sequence can be received from an initial condition of the test generator by the coefficients of formation the shifted copies.

Step 5. The value of shifted coefficients of M-sequences for polinomial $\varphi(x) = 1 \oplus * x^2 \oplus * x^3$ for the appropriate meanings of shift have the following

meanings: 1.δ (0) = {1, 0, 0} ; 2. δ (2) = {1, 1, 0} ; 3. δ (3) = {1, 1, 1}; 4. δ (5) = {0, 0, 1}.

Step 6 and 7. Now it is possible to copy equality for specifióâ bits of a M-sequence of rather initial state of the generator:

$$0 * b_0 + 0 * b_1 + 1 * b_2 = 1$$
$$1 * b_0 + 1 * b_1 + 0 * b_2 = 1 \Rightarrow$$

$$b_0 = 0, b_1 = 1, b_2 = 1$$
$$b_{0+j} = 1, \; b_{1+j} = 0, \; b_{2+j} = 1$$

$$0 * b_{0+k} + 0 * b_{1+k} + 1 * b_{2+k} = 0$$
$$1 * b_{0+k} + 1 * b_{1+k} + 1 * b_{2+k} = 1 \Rightarrow \; k \neq 0$$

$$b_{0+k} = 0, b_{1+k} = 1, b_{2+k} = 0$$
$$b_{0+k} = 1, b_{1+k} = 0, b_{2+k} = 0$$

$$0 * b_{0+j} + 0 * b_{1+j} + 1 * b_{2+j} = 1$$
$$1 * b_{0+j} + 1 * b_{1+j} + 0 * b_{2+j} = 1$$
$$1 * b_{0+j} + 1 * b_{1+j} + 1 * b_{2+j} = 0$$
$$0 * b_{0+j} + 0 * b_{1+j} + 1 * b_{2+j} = 1 \Rightarrow \; j \neq k \neq 0$$

$$b_{0+j} = 0, \; b_{1+j} = 1, \; b_{2+j} = 1$$
$$b_{0+j} = 1, \; b_{1+j} = 0, \; b_{2+j} = 1$$

Step 8.Thus, all three systems have decisions. It means, that all needed deterministic test cubes can be generated. The concurrence of the decisions of the first and third system is a consequence that the first determined test patterns completely becomes covered by the third patterns, as was emphasized above, the probability of occurrence of such situation is small.

For example, if we shall take the decision of the first system as the start condition of the test generator, $b_0 = 0, b_1 = 1, b_2 = 1, \Rightarrow a_0 = b_0, a_1 = b_3, a_2 = b_6 \Rightarrow a_0 = 0, a_1 = 0, a_2 = 1$, according to the initial data, condition of a scan chain will be the following:

1. 0 1 1 1 0 ; 2. 1 1 0 0 1; 3. 0 0 1 0 1; 4. 1 0 1 1 1;
2. 5. 1 1 1 0 0; 6. 1 0 0 1 0; 7. 0 1 0 1 1.

As it is clear from the example needed deterministic patterns {11xxx} will be generated in the fifth case, the patterns {0xx1x} will be generated in first and in the seventh cases, and the patterns {11x01} in the second case.

6. Experimental results

The offered method of the analysis of primitive polinomial of the test generator of pseudo-random patterns was taken for a basis in realization of a software tools allowing to find primitive polimomial , ensuring an opportunity of generation of the deterministic test cubes. The developed software realizes two basic modes :

1. Mode of the analysis given primitive polinomial for an opportunity of generation of the needed deterministic test patterns with specified bits.
2. Mode of search optimum primitive polinomial $\varphi(x)$ with the minimal major degree m and minimal number of unzero factors α_i, ensuring of generation of the needed deterministic test cubes.

In table 2 data from work [4] are given about the deterministic test patterns necessary for achievement of the maximal fault coverage of the circuit. For example, the generator of pseudo-random test patterns is applied to the circuit s838 with a scan chain of length n = 66 bits on the basis of primitive polinomial of a degree m = 14. For achievement of the maximal fault coverage N = 159 deterministic test cubes with total specified bit equal S = 4415 are generated, and the maximal number of specified bit in one set is equal $s_{max} = 36$.

From tabel 2 follows, that quantity of specified bit in the deterministic test cubes makes an insignificant part in relation to common length of a scan chain. Experiments were made and established dependence between given deterministic patterns and number of specified bits in the patterns and ability of generating needed test cubes. The results of a various degree of polinomial are shown on fig.2. Length of a scan chain was 17 bits. The sets from N = 100 deterministic test cubes with random allocated specified bits equial s were used. The data given for polinomials with power m = 14 and m = 17.

On the basis that relative amount of specified bits in the deterministic pattrens insignificant (see tabel 2) given experimental data prove high efficiency of using this method of analysis for maintenance of the given maximal fault coverage of circuits.

7. Conclusion

In the article the new method of analysis of primitive polinomial for a test generator for BIST is presented. The method is allowing to define presence of the needed deterministic test patterns in all set of generated pseudo-random test cubes, and also, with impossibility of generation of the such cubes to find primitive polinomial, ensuring generation of needed deterministic patterns.

The analys of the experimental data, received with the developed software tools, allows to consider the offered method of the polynomial analysis as the method ensuring required fault coverage without increase hardware overhead by generation of deterministic test patterns, that is obvious advantage on compare with used nowadays methods of self-testing.

Table 2

Deterministic test patterns

Circuit	N	m	N	S	s_{max}
s641	54	14	8	167	22
s838	66	14	159	4415	36
s953	45	14	9	123	14
s5378	214	14	33	532	23

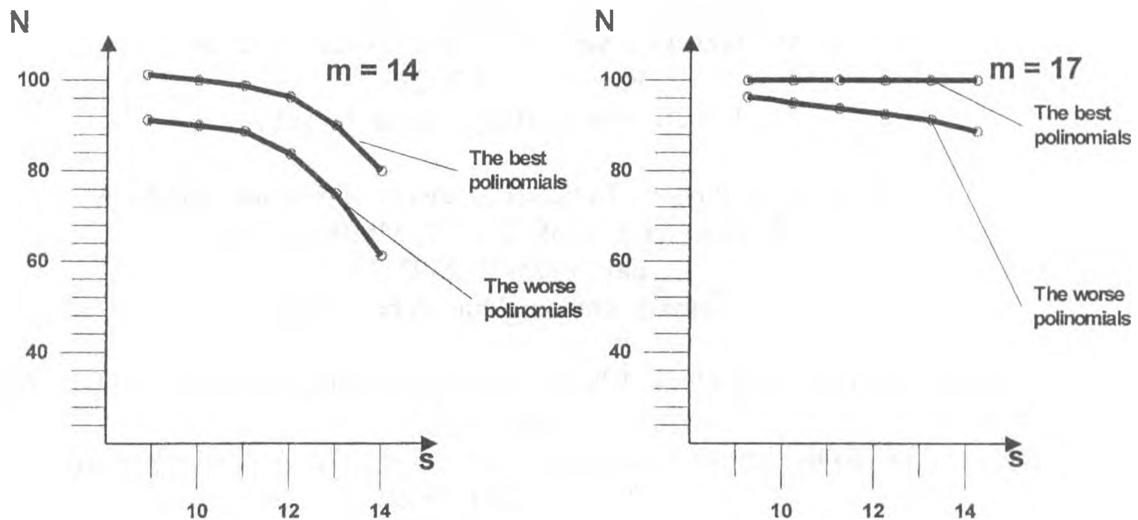

Fig. 2 The ability of generation deterministic patterns by polinomials with different power

References

1. Hans-Joachim Wunderlich, Gundolf Kiefer. "Bit-Flipping BIST". *IEEE Int. Test Conf.*, 1996. pp. 337-343.
2. Sybille Hellebrand, Hans-Joachim Wunderlich, Andre Hertwig. "Mixed-Mode BIST Using Embedded Processors". *IEEE Int. Test Conf.*, 1996.
3. Sybille Hellebrand, Janusz Rajski, Stefen Tarnick, Srikanth Venkataraman, Bernard Courtois. "Built-In Test for Circuits with Scan Based on Reseeding of Multiple-Polynomial Linear Feedback Shift Register" .*IEEE Int. Test Conf.*, 1995. pp. 223-233
4. Sybille Hellebrand, Birgit Reeb, Stefen Tarnick, Hans-Joachim Wunderlich. "Pattern Generation for Deterministic BIST Scheme" . *IEEE Int. Test Conf.*, 1995. pp. 88-94.
5. Stephen Pateras, Janusz Rajski. "Cube-Containsed Random Patterns and their Application to the Complete Testing of Synthesized Multi-level Circuits". *IEEE Int. Test Conf.*, 1991. pp. 473-482.
6. Christophe Fagot, Patrick Girard, Cristian Landrault. "A Novel Approach for Logic BIST Based on Machine Learning ". *IEEE Int. On-Line Testing Workshop*, 1997. pp. 170-174.
7. Gundolf Kiefer, Hans-Joachim Wunderlich. "Using BIST Control for Pattern Generation ".*IEEE Int. Test Conf.*, 1997. pp. 347-355.
8. Nur A.Touba, Edward J. McCluskey." Syntethis of Mapping Logic for Generating Transformed Pseudo-Random Patterns for BIST". *IEEE Int. Test Conf.*, 1995.
9. V.N. Yarmolik , S.N. Demidenko. *Generating and Using Pseudo-Random Signals in Test Systems* . Science and Engineering Publisher, Minsk,1986.