Generation of test sequences with a given switching activity

Mikalai Shauchenka
Weird Science Club, Lichtenbergschule Darmstadt
Ludwigshöhstraße 105, 64285 Darmstadt, Germany
nik.sh.de@gmail.com

Abstract: *The Proposed study is based on the universal method of quasi-random Sobol sequences generation, efficiently used for address test sequence formation. As the mathematical model, a modification of the economical method of Antonov and Saleev is used. The main idea of the suggested approach is the use of generating matrixes with not necessarily the maximum rank for the procedure of generating test patterns. The proposed approach allows the generation of significantly more different sequences with different switching activities of the individual bits as well as the sequences itself. Mathematical expressions are obtained that make it possible to estimate the limiting values of the switching activity, both of the test sequence itself and the individual bits. Examples of the application of the proposed methods are considered.*

Keywords*: COMPUTER SYSTEM TESTING, ADDRESS TEST SEQUENCES, SWITCHING ACTIVITY, MULTIPLE TESTING, MODIFIED SOBOL SEQUENCES*

1. Introduction

The testing of modern computing systems such as embedded systems, systems on a chip, and nets on a chip is important and is in demand at the present time [1, 2, 3]. There are numerous approaches and new solutions aimed to increase efficiency of modern computer system testing. Among them the deterministic tests play a crucial role, such as the *counting sequences*, *Gray sequences*, *anti-Gray sequences*, sequences with *maximal Hamming distance*, and many others [4, 5]. Such sequences are usually periodic and are often called *enumeration sequences*, *de Bruijn sequences*, or sequences of *maximum length* [5]. There are many kinds of maximum length sequences. Each of these test sequences is described by its unique algorithm that suggests a specific implementation and has characteristics that are common with the other sequences. As the general characteristic of a test sequence, the most commonly used one is the so-called *switching activity*, which affects the switching activity of the tested computer systems [4–8].

Address sequences as subset of periodic tests have been investigated within the framework of the memory built-in self-testing [9–13], and multi-run memory testing [14–16]. The main feature of this kind sequence is that it consists of the entire set of binary vectors including all possible 2^m binary combinations, where m is the address length in bit. The task of generating the desired address sequence can therefore be regarded as the generation of m-dimensional binary vectors in binary space.

There is no doubt that the efficiency of the test is a major design issue. To achieve higher efficiency, a characteristic like the switching activity very often plays the crucial role. At the same time, the restriction on the set of the patterns (which is always 2^m) in address sequences may reduce the efficiency of the memory test procedures. To overcome this tradeoff the new approach as the extension of the idea of address sequences generation for a general case of the test sequences with specified values of switching activities is proposed and analyzed in this paper. The motivation for this work is to design an efficient test sequence generator, which allows to generate significantly more different sequences with different switching activities of the individual bits of an address and of the whole sequence itself.

2. Mathematical Model

To significantly reduce hardware overhead needed to generate many different address sequences, a mathematical model of a universal generator was considered and investigated [15]. An address sequence is an ordered sequence of m-bit binary vectors $A(n) = a_{m-1}(n)\ a_{m-2}(n)a_{m-3}(n) \ldots a_1(n)a_0(n)$, $a_i(n)\in\{0, 1\}$, $i\in\{0, 1, 2, \ldots, m-1\}$ and $n\in\{0, 1, 2, \ldots, 2^m-1\}$ where the vectors take all possible values $\{0, 1, 2, \ldots, 2^m-1\}$ exactly once. As the basis of this model, a modified method of *Sobol sequence* generation is used [11, 16, 18]. According to this model, the nth element $A(n) = a_{m-1}(n)a_{m-2}(n)a_{m-3}(n) \ldots a_1(n)a_0(n)$, of the *Sobol* sequence is formed in accordance with the following recurrence relation

$$A(n) = A(n-1) \oplus v_i, \quad n = \overline{0, 2^m-1}, i = \overline{0, m-1}, \tag{1}$$

in which only one modified direction number $v_i = \beta_{m-1}(i)\beta_{m-2}(i)\beta_{m-3}(i) \ldots \beta_1(i)\beta_0(i)$, $i\in\{0, 1, 2, \ldots, m-1\}$ is added to the previous element $A(n-1)$ of the *Sobol* sequence [15]. The value of the index i of the direction number v_i that is used in expression (1) as a term depends on the so-called *switching sequence* T_{m-1} of the *reflected Gray code* [15, 18]. For example, for $m = 4$, the switching sequence has the form $T_3 = 0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0$, which represents the sequence of indices $i\in\{0, 1, 2, 3\}$ that are used for the generation of the sequence $A(n) = a_3(n)a_2(n)a_1(n)a_0(n)$ for $m = 4$ according to (1). Using an arbitrary initial value $A(0)\in\{0, 1, 2, \ldots, 2^m-1\}$, the recurrence relation (1) allows to obtain all other $2^m - 1$ values of $A(n)$ [15, 18]. This mathematical model was generalized for the case of sequences that are related to not only the set of quasi-random test sets [18]. In a general case, any binary square matrix of dimension $m \times m$ can be used as a generating matrix V consisting of direction numbers v_i, $i \in\{0, 1, 2, \ldots, m-1\}$

$$V = \begin{vmatrix} \beta_{m-1}(0) & \beta_{m-2}(0) & \beta_{m-3}(0) & \ldots & \beta_0(0) \\ \beta_{m-1}(1) & \beta_{m-2}(1) & \beta_{m-3}(1) & \ldots & \beta_0(1) \\ \beta_{m-1}(2) & \beta_{m-2}(2) & \beta_{m-3}(2) & \ldots & \beta_0(2) \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \beta_{m-1}(m-1) & \beta_{m-2}(m-1) & \beta_{m-3}(m-1) & \ldots & \beta_0(m-1) \end{vmatrix}, \tag{2}$$

constructed from m linearly independent binary vectors $v_i = \beta_{m-1}(i)\beta_{m-2}(i)\beta_{m-3}(i) \ldots \beta_1(i)\beta_0(i)$, $i = \overline{0, m-1}$. Linear independence is a necessary condition to generate all entries of the address sequence. In this case the matrix (2) has the maximal rank.

To evaluate the properties of modified *Sobol* sequences $A(n) = a_{m-1}(n)a_{m-2}(n)a_{m-3}(n) \ldots a_1(n)a_0(n)$ when used as a test sequence, the metric $F(a_j)$, $j\in\{0, 1, 2, \ldots, m-1\}$, which determines the number of switches (changes) of the jth digit a_j of the sequence $A(n)$ has been proposed and analyzed in [15]. This metric $F(a_j)$ is called *switching activity* [6, 8, 18] and determines the total number of switches of the jth digit of the test patterns $A(n)$ when all 2^m patterns are generated.

3. Proposed method

As was shown earlier [11, 15–16, 18], the use of generating matrices (2) of maximum rank allows to generate a wide range of sequences having a maximum length of 2^m. Such sequences were often called address sequence because they are composed from the whole set of non-repeating binary *m*-bit patterns. The requirement of maximizing the rank of the generating matrix V allows to obtain all 2^m binary combinations in the generated sequence $A(n)$ (1), however, they impose several restrictions on the properties of such sequences [18].

The proposed solution is based on the extension of the mathematical model (1) in terms of generating matrix V, which in this case does not have to have the maximal rank. This modification will allow to generate a wider spectrum of test sequences with a desired switching activity than the known solutions [15, 18]. As the examples of such an approach, Table 1 contains some sequences $A(n)$ obtained according to (1) with multiple generating matrixes V for $m = 4$ with the ranks less than 4.

The first (V_1) is the trivial solution to generate the sequence $A(n)$ with the maximum switching activity $F_{av}(A) = 4$, which equals m for general case. It should be mentioned that the sequence $A(n)$ consists from two patterns, namely arbitrary $A(0)$ and its negation, as presented in Table 1, where $A(0) = 0\ 1\ 0\ 0$ and $\overline{A(0)} = 1\ 0\ 1\ 1$. The rank of the matrix V_1 equals to 1, matrix V_2 has rank 1, and matrix V_3 has rank 3. In all these cases the test sequence $A(n)$ is a periodic sequence with the period being less than $2^m = 2^4$. At the same time the matrix V_4 with rank 2 used in (1) for test sequence generation provides the output sequence with the maximum period of 2^4 but does not include all possible 4-bit binary combinations.

A brief analysis of the examples presented in Table 1, allows to make conclude that in a case of a random matrix V with an arbitrary rank there are different types of test sequences $A(n)$ generated by equation (1). The type of test sequences called address sequence in

V	V_1	V_2	V_3	V_4	V_5
$\beta_3(0)\beta_2(0)\beta_1(0)\beta_0(0)$ $\beta_3(1)\beta_2(1)\beta_1(1)\beta_0(1)$ $\beta_3(2)\beta_2(2)\beta_1(2)\beta_0(2)$ $\beta_3(3)\beta_2(3)\beta_1(3)\beta_0(3)$	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0	0 0 1 0 1 0 0 0 1 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 1	0 1 1 0 1 1 0 0 0 1 1 0 1 1 0 0
$A(0)$	0 1 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
$A(1) = A(0) \oplus v_0$	1 0 1 1	1 1 1 1	0 0 1 0	0 0 0 0	0 1 1 0
$A(2) = A(1) \oplus v_1$	0 1 0 0	1 1 1 1	1 0 1 0	0 0 0 0	1 0 1 0
$A(3) = A(2) \oplus v_0$	1 0 1 1	0 0 0 0	1 0 0 0	0 0 0 0	1 1 0 0
$A(4) = A(3) \oplus v_2$	0 1 0 0	0 0 0 0	0 1 1 1	1 0 0 1	1 0 1 0
$A(5) = A(4) \oplus v_0$	1 0 1 1	1 1 1 1	0 1 0 1	1 0 0 1	1 1 0 0
$A(6) = A(5) \oplus v_1$	0 1 0 0	1 1 1 1	1 1 0 1	1 0 0 1	0 0 0 0
$A(7) = A(6) \oplus v_0$	1 0 1 1	0 0 0 0	1 1 1 1	1 0 0 1	0 1 1 0
$A(8) = A(7) \oplus v_3$	0 1 0 0	0 0 0 0	0 0 0 0	1 1 0 0	1 0 1 0
$A(9) = A(8) \oplus v_0$	1 0 1 1	1 1 1 1	0 0 1 0	1 1 0 0	1 1 0 0
$A(10) = A(9) \oplus v_1$	0 1 0 0	1 1 1 1	1 0 1 0	1 1 0 0	0 0 0 0
$A(11) = A(10) \oplus v_0$	1 0 1 1	0 0 0 0	1 0 0 0	1 1 0 0	0 1 1 0
$A(12) = A(11) \oplus v_2$	0 1 0 0	0 0 0 0	0 1 1 1	0 1 0 1	0 0 0 0
$A(13) = A(12) \oplus v_0$	1 0 1 1	1 1 1 1	0 1 0 1	0 1 0 1	0 1 1 0
$A(14) = A(13) \oplus v_1$	0 1 0 0	1 1 1 1	1 1 0 1	0 1 0 1	1 0 1 0
$A(15) = A(14) \oplus v_0$	1 0 1 1	0 0 0 0	1 1 1 1	0 1 0 1	1 1 0 0

a case of maximal rank matrix V, have been thoroughly investigated in [11, 15–16, 18]. Therefore, further analysis will be focused on the general case of the sequences $A(n)$ obtained by (1) using any generating matrix V regardless of its rank.

First, let us to formulate the property of the sequences $A(n)$ generated based on relation (1) using a random generating matrix V (2).

The switching activity $F(a_j)$, for the jth $j \in \{0, 1, 2, ..., m-1\}$ digit a_j of the sequence $A(n) = a_{m-1}(n)\, a_{m-2}(n) a_{m-3}(n) \ldots a_1(n) a_0(n)$ generated according to (1) takes values in the range from 0 to $2^m - 1$. The value of this switching activity $F(a_j) = 0$ for the jth bit of the $A(n)$ generated according to (1) is provided by the jth all zero column in the generating matrix V, as it was shown in Table 1 (see V_4 and V_5). Maximal switching activity $F(a_j) = 2^m - 1$ for the jth bit of the $A(n)$ corresponds to all ones within jth column in matrix V as it is for the cases of V_1 and V_5, shown in Table 1.

The switching activity $F(A)$ of the sequence $A(n)$, takes the minimum value $F(A) = 0$ in the case of an all zero generating matrix V. The maximum value of the switching activity $F(A) = m \times (2^m - 1)$ is achieved with the all ones matrix V (see V_1 in Table 1). The ranges of values of average switching activity values $F_{av}(A)$ and $F_{av}(a_j)$ are presented below (3).

$$\begin{aligned}
&\min\ F_{av}(a_j) = \min\ F(a_j)/(2^m - 1) = 0; \\
&\max\ F_{av}(a_j) = \max\ F(a_j)/(2^m - 1) = 1; \\
&\min\ F_{av}(A) = \min\ F(A)/(2^m - 1) = 0; \\
&\max\ F_{av}(A) = \max\ F(A)/(2^m - 1) = m.
\end{aligned} \tag{3}$$

A wide range of possible values of switching activity (3), as well as the absence of many restrictions on their mutual relationship allows us to generate a significantly larger number of test sequences with specified values of switching activities.

4. Maximal Length Test Sequences

As the test sequence $A(n) = a_{m-1}(n) a_{m-2}(n) a_{m-2}(n) \ldots a_1(n) a_0(n)$, where $a_i(n) \in \{0, 1\}$, $i \in \{0, 1, 2, \ldots, m-1\}$, and $n \in \{0, 1, 2, \ldots, 2^m - 1\}$, m-dimensional binary vectors in binary space were considered [18]. The generation of the specified test sequence has been regarded as the generation of m-dimensional binary vectors in binary space. The set of linearly independent vectors $v_i^* = \beta_{m-1}^*(i)\beta_{m-2}^*(i)\beta_{m-3}^*(i) \ \ldots\ \beta_1^*(i)\beta_0^*(i)$, $i \in \{0, 1, 2, ..., m-1\}$, generates the m-dimensional binary vectors $A(n)$ through all their possible linear combinations [19, 20]:

$$A^*(n) = v_0^* \times b_0(n) \oplus v_1^* \times b_1(n) \oplus v_2^* \times b_2(n) \oplus \ldots \oplus v_{m-1}^* \times b_{m-1}(n). \tag{4}$$

$B(n) = b_{m-1}(n) b_{m-2}(n) b_{m-3}(n) \ \ldots\ b_1(n) b_0(n)$; $b_i(n) \in \{0, 1\}$, $i \in \{0, 1, 2, \ldots, m-1\}$ is any binary vector set consisting of all possible 2^m binary combinations. Then the vector space $A^*(n)$ formed according to (4) is of dimension m and consists of all 2^m vectors, and that is why vectors $A^*(n)$ can be used as an address sequence [18]. For further investigations, the set of vectors $B(n)$ is regarded as the *Linear sequence* or simply the binary *Up-counter* sequence.

Application of the Gray code sequence in (4) was the productive idea for recursive generation of the quasi-random *Sobol* sequences according to (1) [21]. Relation between the output sequence $A^*(n)$ and $A(n)$ generated based on linear sequence (4) and Gray sequence (1) can be described in terms of corresponding generation matrixes V^* and V.

For this sequences $A^*(n)$ and $A(n)$ the next statement is true.

Statement 1. Sequence $A^*(n)$ generated as m-dimensional binary vectors according to (4) based on any generating m by m matrix V^* can also be obtained from the recursive relation (1) ($A(n) = A^*(n)$) with matrix V obtained from (7), and vice versa.

$$V=\begin{vmatrix} v_0 \\ v_1 \\ v_2 \\ \dots \\ v_{m-2} \\ v_{m-1} \end{vmatrix}=\begin{vmatrix} v_0^* \\ v_0^*\oplus v_1^* \\ v_0^*\oplus v_1^*\oplus v_2^* \\ \dots \\ v_0^*\oplus v_1^*\oplus v_2^*\oplus\dots\oplus v_{m-2}^* \\ v_0^*\oplus v_1^*\oplus v_2^*\oplus\dots\oplus v_{m-1}^* \end{vmatrix}; \quad V^*=\begin{vmatrix} v_0^* \\ v_1^* \\ v_2^* \\ \dots \\ v_{m-2}^* \\ v_{m-1}^* \end{vmatrix}=\begin{vmatrix} v_0 \\ v_0\oplus v_1 \\ v_1\oplus v_2 \\ \dots \\ v_{m-3}\oplus v_{m-2} \\ v_{m-2}\oplus v_{m-1} \end{vmatrix}. \tag{5}$$

For example, corresponding matrix V_5^* for the matrix V_5, shown in Table 1 has the form

$$V_5^* == \begin{vmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{vmatrix},$$

and it is obvious that the sequence $A^*(n)$ (4) generated based on V_5^* is identical to the sequence $A(n)$ (1) generated according to V_5.

Statement 1 allows to make the conclusion that the resulting sequence $A(n)$ obtained from the recursive relation (1) is the linear combination of the binary sequences $b_i(n)\in\{0,1\}$, $i\in\{0, 1, 2, \dots, m-1\}$ representing the counter sequence $B(n)$. The sequence $b_i(n)$ has period equals to 2^{i+1}, that is why the period of $A^*(n)$ will be equal to the maximal period when linear sum (4) includes $b_{m-1}(n)$, what follows from next statement [19].

Statement 2. The sum $A^*(n)$ (4) of the periodic binary counting sequences $b_i(n)\in\{0, 1\}$, $i\in\{0, 1, 2, \dots, m-1\}$ has the period 2^{j+1}, where j is the maximal index of the nonzero values of v_i^*.

Based on the last statement, it is easy to obtain a condition for the formation of a sequence $A(n)$ with the maximal period which consists in the fulfillment of the inequality $v_{m-2} \neq v_{m-1}$. This inequality follows from the following affirmation. According to statement the maximal length sequence $A^*(n)$ will be obtained in a case when $v_{m-1}^* \neq 0$. Then to get the sequence $A(n) = A^*(n)$ the next condition should be accomplished $v_{m-2}\oplus v_{m-1} \neq 0$ (see (5)).

Table 1 contains two examples of $A(n)$ with the maximal period 2^4 for the case of two generating matrixes V_4 and V_5, due to for both matrixes $v_2 \neq v_3$. There are three matrixes V_1, V_2 and V_3 in Table 1 with $v_2 = v_3$, which correspond to the sequences with a period less than 2^4.

5. Generation of Test Sequences with Specified Switching Activities

Considering the wide range of applications of test sequences with specified values of switching activities $A(n)$ [5–10, 18], the problems of finding a generator for such sequences can be formulated as follows:

Find the generating matrix V (2), which provides the required values of $F_{av}(A)$ and $F_{av}(a_i)$ of the sequence $A(n) = a_{m-1}(n)a_{m-2}(n)a_{m-3}(n) \dots a_1(n)a_0(n)$, $a_i(n)\in\{0, 1\}$, $i\in\{0, 1, 2, \dots, m-1\}$ and $n\in\{0, 1, 2, \dots, 2^m-1\}$, where the desired values $F_{av}(A)$ and $F_{av}(a_i)$ do not exceed their bounds (3).

There are three cases of this design problem. The first case is the test sequence with a desired $F_{av}(A)$ and the second is specified $F_{av}(a_i)$ for $k \le m$ digits $a_{\alpha1}, a_{\alpha2}, a_{\alpha3}, \dots, a_{\alpha k}$, $\alpha i\in\{0, 1, 2, \dots, m-1\}$ $i=\overline{1,k}$, of the sequence $A(n)$. The third case is the union of the first two.

Let´s show the solutions for all those three types of problems using the algorithm of partition of a positive integer number $F(A) = (2^m-1)\times F_{av}(A)$, that is the basis for matrix V construction as it was shown in [18].

Algorithm for partitioning an integer into terms:

Input data: An integer $F(A)$ belonging to the range from 0 to $m\times(2^m-1)$; The terms should belong to the set of in integer values 2^i, where $i\in\{0, 1, 2, \dots, m-1\}$ to obtain a partition of the integer $F(A)$.

1. Initially, the sum of all terms 2^i is formed, which is equal to the maximum m-bit binary number 2^m-1.

2. The division operation $F(A)$ by 2^m-1 is performed. The resulting quotient q determines the minimum amount of use of each of the terms 2^i in the partition of the integer $F(A)$. If the remainder r of the division operation is zero, the quotient q is the number of uses of each of the terms 2^i in the partition $F(A)$, and at this step the partition algorithm ends. Otherwise, the next step is performed.

3. The remainder $0 < r < 2^m-1$ from the division operation performed in the previous step is represented in the binary code $r = b_{m-1}\times2^{m-1} + b_{m-2}\times2^{m-2} + b_{m-3}\times2^{m-3} + \dots + b_0\times2^0$, $b_i\in\{0, 1\}$.

4. The partition of the integer $F(A)$ into terms 2^i is constructed, where $i\in\{0, 1, 2, \dots, m-1\}$, each of the terms is included in the partition $q + b_{m-1-i}$ times. The quantity $q + b_{m-1-i}$ determines the value of the Hamming weight $w(v_i)$ of the binary vector $v_i = \beta_{m-1}(i)\beta_{m-2}(i)\beta_{m-3}(i) \dots \beta_1(i)\beta_0(i)$, which is the result of applying this algorithm.

Problem 1. Find the generating matrix V for the sequence $A(n)$ for a given value of m and the desired value of $F_{av}(A)$.

The solution of the problem 1 will be as follows: obtaining $F(A) = \text{int}[2^m\times F_{av}(A)]$; partitioning the integer $F(A)$ into terms [18]; obtaining the values of the row weights $w(v_i)$ of the desired generating matrix V and finding the matrix with maximal rank corresponding to the given values of m and $w(v_i)$. The matrix with maximal rank is chosen from the matrix with the restriction that $v_{m-1} \neq v_m$, which guarantees the maximal length of the sequence $A(n)$. The maximal rank of the matrix V is the requirement to get maximal number of binary combinations $a_{m-1}(n)a_{m-2}(n)a_{m-3}(n) \dots a_1(n)a_0(n)$ within $A(n)$ [19].

*Problem 2***.** For a given value m find the generating matrix V for sequence $A(n)$, in which specified $F_{av}(a_j)$ for $k \le m$ digits $a_{\alpha1}, a_{\alpha2}, a_{\alpha3}, \dots, a_{\alpha k}$, $\alpha j\in\{0, 1, 2, \dots, m-1\}$ $j=\overline{1,k}$, are determined.

Initially, as in the case of Problem 1, the average values $F_{av}(a_{\alpha1})$, $F_{av}(a_{\alpha2})$, $F_{av}(a_{\alpha3})$, …, $F_{av}(a_{\alpha k})$ of switching activities are represented as total values of the number of switching bits $a_{\alpha1}, a_{\alpha2}, a_{\alpha3}, \dots, a_{\alpha k}$ of $A(n)$. These values $F(a_{\alpha1})$, $F(a_{\alpha2})$, $F(a_{\alpha3})$, …, $F(a_{\alpha k})$ are determined according to the relation $F(a_{\alpha j}) = \text{int}\,[F_{av}(a_{\alpha j})\times(2^m-1)]$. Then, $F(a_{\alpha j})$ is converted to an m-bit code represented in the binary number system $F(a_{\alpha j})_{(10)} = F(a_{\alpha j})_{(2)} = \beta_{\alpha j}(0)\times2^{m-1} + \beta_{\alpha j}(1)\times2^{m-2} + \beta_{\alpha j}(2)\times2^{m-3} + \dots + \beta_{\alpha j}(m-1)\times2^0$. The values of all $k \le m$ columns of the matrix V are calculated, which provides the switching activities $F_{av}(a_{\alpha1})$, $F_{av}(a_{\alpha2})$, $F_{av}(a_{\alpha3})$, …, $F_{av}(a_{\alpha k})$. The next step in solving Problem 2 is to randomly (equally likely and independently) generate the remaining columns of the binary matrix V. From the resulting matrixes, the matrix with the highest rank must be selected.

Problem 3. For a given value m find the generating matrix V for sequence $A(n)$, in which specified $F_{av}(a_j)$ for $k \le m$ digits $a_{\alpha1}, a_{\alpha2}, a_{\alpha3}, \dots, a_{\alpha k}$, $\alpha j\in\{0, 1, 2, \dots, m-1\}$ $j=\overline{1,k}$, are determined, and the switching activity $A(n)$ equals $F_{av}(A)$.

The correct formulation of Problem 3 assumes that $F_{av}(a_{\alpha1}) + F_{av}(a_{\alpha2}) + F_{av}(a_{\alpha3}) + \dots + F_{av}(a_{\alpha k}) < F_{av}(A) \le m$. At the initial stage, the solution of Problem 3 repeats the solution to Problem 2. Next, the steps of the procedure for solving Problem 1 are performed. The difference is the partition on the terms have to be performed on the integer $F^*(A) = \mathrm{int}[F_{av}(A) \times (2^m - 1)] - \mathrm{int}[F_{av}(a_{\alpha 1}) \times (2^m - 1)] - \mathrm{int}[F_{av}(a_{\alpha 2}) \times (2^m - 1)] - ... - \mathrm{int}[F_{av}(a_{\alpha k}) \times (2^m - 1)]$. In addition, when obtaining the row weights $w(v_i)$ of the desired generating matrix V, it is necessary to consider the row weights of the previously generated k columns.

The partition of the integer $F^*(A)$ into terms 2^i is constructed, where each term is included in the partition $q + b_{m-1-i} = 1 + 0$ times. Since $r = 0000$, the terms 2^3, 2^2, 2^1 are included in the partition of 15 only once. The value $q + b_{m-1-i}$ determines the Hamming weight of the rows of the desired matrix V, consisting of four rows and four columns, excluding the first and third columns. Then, randomly generated values of six two-digit binary vectors with Hamming weights equal to $w(v_0) = w(v_1) = w(v_2) = w(v_3) = 1$, which will be determine the values of the remaining columns (except the first and third) of the desired matrix. For the matrixes thus obtained, the maximal of its rank is determined. One of the possible solutions of this example may be the matrix V_5, shown in Table 1.

6. Conclusion

The use of a modified mathematical model for the *Sobol* sequences generation has allowed to expand the capabilities of the test sequence generator in terms of a significant increase in the number of different types of such sequences. The paper describes a method for constructing a test sequences with given values of switching activity, of the test patterns and the activity of their bits. The essence of the method consists in the synthesis of the required generating matrix providing specified values of switching activity. The limitations of the previously proposed and investigated techniques associated with possible conflicting requirements for the values of the weights of the rows of the matrix and their linear independence are shown. The main idea that distinguishes the results obtained in this article is the use of arbitrary matrices for which the condition that ensures the maximum period of the test sequence is fulfilled. Of the many matrices for which this condition is satisfied, it is necessary to use matrices with a maximum rank. In this case, not only the required values of the switching activity are provided, but also a larger number of test patterns in the sequence. Examples of the use of such sequences for the purpose of constructing a test pattern generator are considered.

7. References

[1] M.L. Bushnell and V.D. Agrawal, New York: Kluwer Academic Publishers, *Essentials of Electronic Testing for Digital, Memory & Mixed-Signal VLSI Circuits* (2000).

[2] L.-T. Wang, C.-W. Wu and X. Wen, New York: Elsevier Inc. *VLSI Test Principles and Architectures: Design for Testability*. (2006).

[3] I. A. Grout, Springer-Verlag, *Integrated Circuit Test Engineering. Modern Techniques* (2006).

[4] E.J. Marinissen, B. Prince, D. Keitel-Schulz and Y. Zorian, Proc. of Design, Automation and Test in Europe Conference and Exhibition, Munich, Germany, *Challenges in Embedded Memory Design and Test*, pp. 722–727 (2005).

[5] V.N. Yarmolik, Bestprint, *Kontrol' I diagnostika vuchislitel'nuh system* (2019).

[6] I. Pomeranz, IEEE Trans. Comput., *An Adjacent Switching Activity Metric under Functional Broadside Tests*, **vol. 62**, **№ 4**, pp. 404–410 (2013).

[7] P.A Girard, L. Guiller, C. Landrault, and S. Pravossondovitch, Proc. Ninth Great Lakes Symposium on VLSI, *A test vector ordering technique for switching activity reduction during test operation*, pp. 24–27 (1999).

[8] S. Wang and S.K. Gupta, IEEE Trans. Comput.-Aided Design of Integr. Circuits and Systems, *An automatic test pattern generator for minimizing switching activity during scan testing activity*, **vol. 21**, **№ 8**, pp. 954–968 (2002).

[9] S. Saravanan, M. Hailu, G.M. Gouse, M. Lavanya, and R. Vijaysai, Proc. of 6th EAI International Conference, ICAST, Bahir Dar, Ethiopia, *Design and Analysis of Low-Transition Address Generator* (2018).

[10] P.A. Pavani, G. Anitha, J. Bhavana, and J.P. Raj, Inter. J. of Scien. & Eng. Research, *Novel Architecture Design of Address Generators for BIST Algorithms*, **vol. 7**, **№ 2**, pp. 1484–1488 (2016).

[11] V.N. Yarmolik, and S.V. Yarmolik, Autom. Control Comput. Sci, *Address sequences*, pp. 207–213, **vol. 48**, **№ 4** (2014).

[13] B. Singh, S. Narang, and A. Khosla, IJCSI Int. J. Comput Sci. Issues, *Address Counter / Generators for Low Power Memory BIST*, **vol. 8**, **Issue 4**, **№ 1**, pp. 561–567 (2011)

[14] I. Mrozek and V.N. Yarmolik, Journal of Electronic Testing: Theory and Applications, *Iterative Antirandom Testing*, **vol. 9**, **№ 3**, pp. 251–266

[15] Yarmolik, V.N. and Yarmolik, S.V., Generating Modified Sobol Sequences for Multiple Run March Memory Test, *Autom. Control Comput. Sci*, vol. 47, № 5, pp. 242–247 (2013).

[16] V.N. Yarmolik, and S.V. Yarmolik, Autom. Control Comput. Sci, *Address sequences for multiple-run March tests of random-access memory*, **vol. 40**, **№ 5**, pp. 42–49. (2006)

[17] A.J. Goor, H. Kukner, and S. Hamdioui, , Proc. of 2011 6th Int. Conf. on Design & Tech. of Integrated Systems in Nanoscale Era, *Optimizing memory BIST Address Generator implementations*, pp. 572–576 (2011).

[18] V.N. Yarmolik, and M.A. Shauchenka, Informatics, *Generation of address sequences with a given switching activity* **vol. 17**, **№ 1**, pp. 47–62 (2020).

[19] S. Boyd, Cambridge, UK: University Printing House, *Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares* (2018).

[20] P. Ferreira, B. Jesus, J. Vieira, and A.J. Pinho, IEEE Communic. Let., *The Rank of Random Binary Matrices and Distributed Storage Applications*, **vol. 17**, **№ 1**, pp. 151–154 (2013).

[21] I.A. Antonov, and V.M. Saleev, Zh. Vychisl. Matem. Mat. Fiz, *Economical method of LP sequences calculation*, **vol. 19**, pp. 243–245 (1979).