ISSN 0005-1179, Automation and Remote Control, 2022, Vol. 83, No. 2, pp. 291–302. © Pleiades Publishing, Ltd., 2022.
Russian Text © The Author(s), 2022, published in Avtomatika i Telemekhanika, 2022, No. 2, pp. 154–168.

OPTIMIZATION, SYSTEM ANALYSIS, OPERATIONS RESEARCH

Synthesis of Test Sequences with a Given Switching Activity

V. N. Yarmolik[1*] and N. A. Shevchenko[2]**

[1] *Belarusian State University of Informatics and Radioelectronics, Minsk, 220013 Belarus*
[2] *Lichtenbergschule Gymnasium, Darmstadt, 64285 Germany*
e-mail: *yarmolik10ru@yahoo.com, **nik.sh.de@gmail.com

Received April 7, 2020; revised June 3, 2021; accepted August 29, 2021

Abstract—The relevance of using test sequences with a given switching activity is discussed. As a mathematical model for generating the tests, a modification of the Antonov–Saleev method for generating Sobol sequences is used. It is based on the use of maximum-rank generating matrices the form of which determines the main properties of the sequences. It is shown that the construction of a generating matrix is reduced to the problem of partitioning an integer, and an algorithm for splitting into summands of a given form is proposed. Procedures for modifying the partition of an integer into summands and for modifying the value of switching activity are introduced. Three problems are stated for the synthesis of generators of test sequences with a given switching activity. Examples of using the proposed methods and experimental results are considered.

Keywords: test sequence, self-testing of computing systems, switching activity

DOI: 10.1134/S0005117922020114

1. INTRODUCTION

The efficiency of test sequences for modern computing systems is largely determined by the properties of test objects [1, 2]. It is important to concisely represent tests in the form of algorithms or hardware structures for their generation when implementing self-testing of *embedded systems*, *systems-on-a-chip*, *nets-on-a-chip* [2, 3], and, first of all, for self-testing of their storage devices, the share of which reaches 90% of the crystal area occupied by the system [3, 4]. An essential role for test sequences is played by *switching activity*, which affects the switching activity of the digital devices to be tested. The set of test sequences in the self-test architectures used includes sequences with different switching activity [5, 6]. The set of such sequences includes: *linear counting sequences*, *Gray code sequences*, sequences with maximum switching activity (*address complement sequences*); sequences with Hamming distance equal to one for all pairs of addresses (2^i *counting sequences*), and a number of other sequences [3, 5, 6].

Switching activity is of decisive importance in the field of designing digital devices with low energy consumption [7], including the development and application of means for their testing and self-testing [8, 9]. A large number of studies in this area are aimed at obtaining estimates of the values of the switching activity of the poles of the designed devices, which make it possible to predict their energy consumption [8, 10].

The mean values of switching activity can be interpreted as the average values of the Hamming distance, which are widely used to construct controlled probabilistic test sequences [11–15]. Changes in the values of these characteristics permit one to construct controlled probabilistic tests with given values of the Hamming distance.

It should be noted that the study of the synthesis of various types of devices with variable values of switching activity for testing computer systems is only at the initial stage. In particular, the methods for synthesizing address sequence generators, discussed in a number of sources [6, 16–18], make it possible to construct such devices described by fixed values of switching activity. The problem of synthesizing devices for generating test sequences with a given switching activity and forming controlled probabilistic test sequences remains practically open.

The present paper provides a solution to the problem of synthesizing generators of test sequences of maximum length consisting of 2^m m-bit sets, called address sequences, with a given switching activity of both separate bits of test sets and with the total switching activity of their sequences.

2. MATHEMATICAL MODEL

A mathematical model of a universal generator of sequences consisting of 2^m m-digit sets, called address sequences, was considered in [19]. By an address sequence one means a sequence $A(n) = a_{m-1}a_{m-2}\ldots a_2a_1a_0$, $n \in \{0, 1, 2, \ldots, 2^m - 1\}$, where $a_i \in \{0, 1\}$, $i \in \{0, 1, 2, \ldots, m-1\}$, which consists of all possible 2^m m-digit binary vectors $a_{m-1}a_{m-2}\ldots a_2a_1a_0$ generated in an arbitrary order, with each vector formed only once [2, 6, 19]. For example, the linear counting address sequence for $m = 4$ consists of 16 4-digit binary vectors 0000, 0001, 0010, 0011, ..., 1111, each formed only once [6]. As a basis for the mathematical model we use a modified method of forming Sobol sequences [20–22]. According to the model indicated, the nth element of the Sobol sequence $A(n)$, which is an m-digit binary vector $a_{m-1}a_{m-2}\ldots a_0$, is formed in accordance with the recurrence relation

$$A(n) = A(n-1) \oplus \mathrm{v}_i \quad n = 0, \ldots, 2^m - 1, \quad i = 0, \ldots, m-1. \tag{1}$$

in which only one modified direction number v_i, $i \in \{0, 1, 2, \ldots, m-1\}$ is added to the previous element $A(n-1)$ of the Sobol sequence, which is also an m-digit binary vector [19–21]. The value of the index i of the direction number v_i used as a summand in the expression (1) depends on the switching sequence (*transition sequence*) T_{m-1} of the reflected Gray code [21–23]. In the Gray code, the transition from the previous state of the code to the next one is carried out by inverting only one of its digits. The sequence of indices of these digits is the switching sequence [23]. In the case of the most commonly used version of the Gray code, namely the reflected Gray code, the switching sequence is easily formed from the linear counting sequence [23]. The index of the most significant variable bit of the linear counting sequence during the formation of its next code will be an element of the switching sequence. For example, when generating the code of the linear counting sequence $A(n) = a_3a_2a_1a_0 = 0001$, only the least significant bit is changed from the previous one 0000; accordingly, the first element of the switching sequence T_3 will be the index 0. For $m = 4$, the switching sequence of the reflected Gray code is $T_3 = 0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0$. This sequence forms the index sequence $i \in \{0, 1, 2, 3\}$ when generating $A(n) = a_3a_2a_1a_0$ 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) yields all $2^m - 1$ other values of $A(n)$ [19, 22].

This mathematical model was generalized to include the case of sequences belonging not only to the set of Sobol sequences [22]. In the general case, for the generating matrix V of direction numbers v_i, $i \in \{0, 1, 2, \ldots, m-1\}$, one can use any binary $m \times m$ square matrix of the form

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

constructed of m linearly independent binary vectors $v_i = \beta_{m-1}(i)\beta_{m-2}(i)\ldots\beta_0(i)$, $i = 0, \ldots, m-1$. The linear independence condition makes it possible to ensure the formation of test sequences of maximum length [19].

3. SWITCHING ACTIVITY

To estimate the properties of the Sobol sequences $A(n) = a_{m-1}a_{m-2}\ldots a_0$ used for a test sequence in [22], a numerical parameter $F(a_j)$, $j \in \{0, 1, 2, \ldots, m-1\}$, was introduced that determines the number of switchings (changes) in the jth digit a_j of the code of the sequence $A(n)$. The numerical characteristic $F(a_j)$ is called switching activity [5, 19, 24] and determines the switching activity of the jth digit a_j of the test sets $A(n)$. In the general case, for an arbitrary value of j, the value of this characteristic for sequence (1) is determined by the formula

$$\begin{aligned} F(a_j) &= \beta_j(0) \cdot 2^{m-1} + \beta_j(1) \cdot 2^{m-2} + \cdots + \beta_j(m-2) \cdot 2^1 + \beta_j(m-1) \cdot 2^0 \\ &= \sum_{i=0}^{m-1} \beta_j(i) \cdot 2^{m-1-i}. \end{aligned} \tag{3}$$

Based on the switching activity $F(a_j)$, an integral switching activity measure

$$F(A) = \sum_{j=0}^{m-1}\sum_{i=0}^{m-1} \beta_j(i) 2^{m-1-i} = \sum_{i=0}^{m-1} 2^{m-1-i} \sum_{j=0}^{m-1} \beta_j(i) \tag{4}$$

was introduced in [19] for a sequence $A(n)$, where the second sum equals the number of ones in the ith row of matrix (2) and is the Hamming weight $w(v_i)$ of the binary vector $v_i = \beta_{m-1}(i)\beta_{m-2}(i)\ldots\beta_0(i)$, $i = 0, \ldots, m-1$.

As follows from the linear independence of the binary vectors v_i, the jth column of the matrix V cannot be zero; therefore, the minimum value of $F(a_j)$, $j \in \{0, 1, 2, \ldots, m-1\}$, is achieved for $\beta_j(m-1) = 1$ and $\beta_j(i) = 0$, $i \in \{0, 1, 2, \ldots, m-2\}$. The minimum value of $F(a_j)$ is possible for any jth digit, but only for one of them [19]. This constraint also follows from the linear independence of the rows of the generating matrix V. The maximum value of $F(a_j)$, $j \in \{0, 1, 2, \ldots, m-1\}$, is ensured by the formation of the jth unit column of matrix (2), i.e., for the values $\beta_j(i) = 1$, where $i = 0, \ldots, m-1$. This implies the equality

$$\max F(a_j) = \sum_{i=0}^{m-1} 2^{m-1-i} = 2^m - 1. \tag{5}$$

The characteristic $F(a_j)$ of sequences (1) possesses the following property.

Property 1. For any sequence $A(n)$ given in (1), the switching activity $F(a_j)$ takes m distinct values in the range from 1 to 2^{m-1}.

The switching activity $F(A)$ of the sequence $A(n) = a_{m-1}a_{m-2}\dots a_0$, $n \in \{0, 1, 2, \dots, 2^m - 1\}$, takes the minimum value for Gray code sequences [22]. For a matrix consisting of m distinct rows, each containing one unity, according to (4), we have $\min F(A) = 2^{m-1}$. The maximum estimate of $F(A)$ is also unambiguously determined by the form of the generating matrix [22], whose first row consists of ones and the remaining rows contain one zero value each and which is determined as

$$\max F(A) = m \cdot 2^{m-1} + (m-1)\sum_{i=1}^{m-1} 2^{m-i-1} = m \cdot 2^m - 2^{m-1} - m + 1. \tag{6}$$

The switching activity $F(A)$ given in (4) possesses the following property.

Property 2. The switching activity $F(A)$ of the sequence $A(n)$ given in (1) assumes values in the range from 2^{m-1} to $m \cdot 2^m - 2^{m-1} - m + 1$.

For the real values of $m > 10$, it is convenient to use the mean values $F_{\text{av}}(a_j)$ and $F_{\text{av}}(A)$ of the previously considered numerical parameters of the switching activity $F(a_j)$ and $F(A)$, which indicate the average value of switchings when forming the next test set. These characteristics are defined as $F_{\text{av}}(A) = F(A)/(2^{m-1})$ and $F_{\text{av}}(a_j) = F(a_j)/(2^{m-1})$, and their maximum and minimum values are

$$\begin{aligned}
\min F_{\text{av}}(a_j) &= \min \frac{F(a_j)}{(2^m - 1)} = \frac{1}{2^m - 1};\\
\max F_{\text{av}}(a_j) &= \max \frac{F(a_j)}{2^m - 1} = 1;\\
\min F_{\text{av}}(A) &= \min \frac{F(A)}{2^m - 1} = 1;\\
\max F_{\text{av}}(A) &= \max \frac{F(A)}{2^m - 1} = m - \frac{1}{2} + \frac{1}{2^{m+1} - 2}.
\end{aligned} \tag{7}$$

An important consequence of the above Properties 1 and 2 is the existence of a set of generating matrices V of maximum rank [19, 22].

4. SYNTHESIS OF SEQUENCES WITH A GIVEN SWITCHING ACTIVITY

For an arbitrary m, the synthesis of a sequence generator $A(n)(1)$ with a given average switching activity $F_{\text{av}}(A)$ and its counterpart $F(A)$ consists in finding the generating matrix V. To do this, a binary $m\times m$ matrix of maximum rank is formed with constraints determined by the value of $F(A)$. Initially, the value of the switching activity $F(A)$ is represented as the partition [19]

$$F(A) = w(\mathrm{v}_0) \cdot 2^{m-1} + w(\mathrm{v}_1) \cdot 2^{m-2} + w(\mathrm{v}_2) \cdot 2^{m-3} + \dots + w(\mathrm{v}_{m-1}) \cdot 2^0. \tag{8}$$

This partition represents the value of $F(A)$ in a m-ary mixed number system, in which the weights of digits are represented as powers of two from 2^0 to 2^{m-1}, and the values of the digits $w(\mathrm{v}_i)$ range from 1 to m. Note that $w(\mathrm{v}_i)$ is the Hamming weight of the binary vector v_i of the desired generating matrix V of maximum rank. The absence of a zero value of $w(\mathrm{v}_i)$ is explained by the impossibility of constructing a square matrix of maximum rank with a zero row whose Hamming weight is equal to 0. The second constraint on the digits $w(\mathrm{v}_i)$ in partition (8) is that only one digit $w(\mathrm{v}_i)$ can take the value m. In terms of the generating matrix V of maximum rank, this means that only one row of the matrix can have a weight equal to m. There are other, not so

Table 1. Examples of partition (8) of number 37

$w(\mathrm{v}_0)$	$w(\mathrm{v}_1)$	$w(\mathrm{v}_2)$	$w(\mathrm{v}_3)$	$F(A) = w(\mathrm{v}_0) \cdot 2^3 + w(\mathrm{v}_1) \cdot 2^2 + w(\mathrm{v}_2) \cdot 2^1 + w(\mathrm{v}_3) \cdot 2^0$	$F(A) = 37$
3	2	2	1	$37 = 3{\cdot}8 + 2{\cdot}4 + 2{\cdot}2 + 1{\cdot}1$	88844221
2	4	1	3	$37 = 2{\cdot}8 + 4{\cdot}4 + 1{\cdot}2 + 3{\cdot}1$	8844442111
2	3	3	3	$37 = 2{\cdot}8 + 3{\cdot}4 + 3{\cdot}2 + 3{\cdot}1$	88444222111

obvious, restrictions on the digits $w(\mathrm{v}_i)$ (Hamming weights) of the partition (8), which are the basis for constructing the matrix V of maximum rank.

By way of example, consider the case of formation of the sequence $A(n)$ for $m = 4$ and switching activity $F(A) = 37$. The value $F(A) = 37$ belongs to the range from 15 to 53 defined by 2. Table 1 lists the partitions (8) of the number 37 for the case of $m = 4$.

Note that each partition (8) can be associated with a set of matrices V in which the weights of the rows correspond to the values of the digits $w(\mathrm{v}_i)$ of the specified partition. For example, for the partition $37 = 3{\cdot}8 + 2{\cdot}4 + 2{\cdot}2 + 1{\cdot}1$, the weight $w(\mathrm{v}_0)$ of the first row of the matrix is 3, the weights $w(\mathrm{v}_1)$ of the second and $w(\mathrm{v}_2)$ of the third row are 2, and the weight $w(\mathrm{v}_3)$ of the fourth row is equal to 1. The matrices V_1 and V_2 given in Table 2 are examples of maximum rank matrices with specified row weights, while the matrices V_3 and V_4 are maximum rank matrices for other partitions (8) of size 37. In Table 2, there are also examples of forming sequences $A(n)$ according to (1) for all four types of matrices V. The sequence T_3 previously given as an example was used as a switching sequence.

The procedure for obtaining the partition (8) for an arbitrary integer value of $F(A)$ can be interpreted as a solution of the problem of partitioning an integer into summands that are positive integers of the form 2^i, where $i \in \{0, 1, 2, \ldots, m-1\}$. Table 1 gives examples of such partitions, one of which is the partition $37 = 8 + 8 + 8 + 4 + 4 + 2 + 2 + 1$, represented as the sequence 88844221 of repeated terms 8, 4, 2, and 1 [25–27].

The simplest way to generate all partitions of an integer into summands, regardless of their order, is to partition in reverse lexicographic order, starting with the integer 'n' being partitioned, when the number itself is represented by one summand n, and ending with the representation '$111\ldots1$' of this number in the form of n summands equal to one [25].

For an integer value of $F(A) = 37$ and $m = 4$, taking into account the constrains on the summands, which in this case can only be 8, 4, 2, and 1, and their number, all possible partitions are as follows: 88844221, 888442111, 888422221, 8884222111, 884444221, 8844442111, 8844422221, 88444222111.

Partitioning the switching activity value is peculiar because of the restriction imposed on the number of summands 2^{m-1}, 2^{m-2}, $\ldots$, 2^0, the number of each of which should not be zero or exceed m, with only one summand being allowed to occur in the partition m times.

Consider an algorithm for partitioning an integer that determines the switching activity $F(A)$ of the sequence $A(n) = a_{m-1}a_{m-2}\ldots a_0$ (1) for a given value of m. The partition terms can only be integers of the form 2^i, where $i \in \{0, 1, 2, \ldots, m-1\}$, and their sum must be in the range from $2^m - 1$ to $m \cdot 2^m - 2^{m-1} - m + 1$ (see Property 2).

Table 2. Examples of generating matrices V and sequences $A(n)$ for $m = 4$

V	V_1	V_2	V_3	V_4
$\beta_3(0)$ $\beta_2(0)$ $\beta_1(0)$ $\beta_0(0)$	1 1 1 0	1 0 1 1	0 1 1 0	1 0 1 0
$\beta_3(1)$ $\beta_2(1)$ $\beta_1(1)$ $\beta_0(1)$	1 1 0 0	1 1 0 0	1 1 1 1	1 1 1 0
$\beta_3(2)$ $\beta_2(2)$ $\beta_1(2)$ $\beta_0(2)$	1 0 0 1	0 1 1 0	0 0 1 0	0 1 1 1
$\beta_3(3)$ $\beta_2(3)$ $\beta_1(3)$ $\beta_0(3)$	0 0 0 1	0 1 0 0	0 1 1 1	1 0 1 1
$A(0)$	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
$A(1) = A(0) \oplus \mathrm{v}_0$	1 1 1 0	1 0 1 1	0 1 1 0	1 0 1 0
$A(2) = A(1) \oplus \mathrm{v}_1$	0 0 1 0	0 1 1 1	1 0 0 1	0 1 0 0
$A(3) = A(2) \oplus \mathrm{v}_0$	1 1 0 0	1 1 0 0	1 1 1 1	1 1 1 0
$A(4) = A(3) \oplus \mathrm{v}_2$	0 1 0 1	1 0 1 0	1 1 0 1	1 0 0 1
$A(5) = A(4) \oplus \mathrm{v}_0$	1 0 1 1	0 0 0 1	1 0 1 1	0 0 1 1
$A(6) = A(5) \oplus \mathrm{v}_1$	0 1 1 1	1 1 0 1	0 1 0 0	1 1 0 1
$A(7) = A(6) \oplus \mathrm{v}_0$	1 0 0 1	0 1 1 0	0 0 1 0	0 1 1 1
$A(8) = A(7) \oplus \mathrm{v}_3$	1 0 0 0	0 0 1 0	0 1 0 1	1 1 0 0
$A(9) = A(8) \oplus \mathrm{v}_0$	0 1 1 0	1 0 0 1	0 0 1 1	0 1 1 0
$A(10) = A(9) \oplus \mathrm{v}_1$	1 0 1 0	0 1 0 1	1 1 0 0	1 0 0 0
$A(11) = A(10) \oplus \mathrm{v}_0$	0 1 0 0	1 1 1 0	1 0 1 0	0 0 1 0
$A(12) = A(11) \oplus \mathrm{v}_2$	1 1 0 1	1 0 0 0	1 0 0 0	0 1 0 1
$A(13) = A(12) \oplus \mathrm{v}_0$	0 0 1 1	0 0 1 1	1 1 1 0	1 1 1 1
$A(14) = A(13) \oplus \mathrm{v}_1$	1 1 1 1	1 1 1 1	0 0 0 1	0 0 0 1
$A(15) = A(14) \oplus \mathrm{v}_0$	0 0 0 1	0 1 0 0	0 1 1 1	1 0 1 1

Integer partition algorithm.

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

Step 2. Divide $F(A)$ by $2^m - 1$. The resulting quotient w determines the minimum number of occurrences of each of the terms 2^i in the partition of the integer $F(A)$. If the remainder q of the division operation is zero, the quotient w is the number of instances of each of the terms 2^i, $i \in \{0, 1, 2, \ldots, m-1\}$ in the partition of $F(A)$, and at this step the partitioning algorithm ends. Otherwise, go to the next step.

Step 3. Represent the remainder $0 < q < 2^{m-1}$ of the division operation in binary code, $q = b_{m-1} \cdot 2^{m-1} + b_{m-2} \cdot 2^{m-2} + \cdots + b_0 \cdot 2^0$, $b_i \in \{0, 1\}$.

Step 4. Construct the partition of the integer $F(A)$ into summands 2^i, where $i \in \{0, 1, 2, \ldots, m-1\}$, each of which is included in the partition $0 < w + b_i \leq m$ times, where the value $w + b_i$ determines the value of the digit $w(\mathrm{v}_{m-1-i})$ of the partition (8).

Applying this algorithm for the case of $m = 6$ and $F(A) = 189$, we conclude that the quotient w of dividing 189 by 63 is 3, the remainder is $q = 0$, and accordingly, the partition of the number 189 is $2^5 2^5 2^5 2^4 2^4 2^4 2^3 2^3 2^3 2^2 2^2 2^2 2^1 2^1 2^1 2^0 2^0 2^0$. The digits of the partition (8) take the values $w(\mathrm{v}_0) = w(\mathrm{v}_1) = w(\mathrm{v}_2) = w(\mathrm{v}_3) = w(\mathrm{v}_4) = w(\mathrm{v}_5) = 3$, and it is these values that determine the weights

(a)

8	8		
4	4		
2	2		
1	1		

(b)

8	8		
4			
2	2	2	2
1	1		

Fig. 1. Diagrams for (a) $w(\mathrm{v}_0) = w(\mathrm{v}_1) = w(\mathrm{v}_2) = w(\mathrm{v}_3) = 2$ and (b) $w(\mathrm{v}_0) = w(\mathrm{v}_3) = 2$, $w(\mathrm{v}_1) = 1$ and $w(\mathrm{v}_2) = 4$.

of the rows in the matrix V. In the case of finding the generating matrix V corresponding to the resulting row weights, which is used to implement relation (1), finding the next set of the sequence $A(n)$ will always be carried out by performing three switchings. In the general case, an important fact is the existence of a generating matrix V of maximum rank whose row weights correspond to the digits of the partition (8) [28].

When a matrix V with a rank different from the maximum is obtained, the random formation of a matrix with fixed Hamming weights of its rows is repeated. Each row of this matrix is a random binary vector with a given weight $w(\mathrm{v}_i)$. Then the rank of the matrix is checked again. Obviously, the requirement for the weights of the matrix rows and at the same time the need for its maximum rank can significantly worsen the probabilistic estimate of a positive outcome of finding the generating matrix V [28]. The inconsistency of these requirements may lead to the impossibility of finding such a matrix, as illustrated by the following example.

Example 1. Determine the row weights of the generating matrix V to form the sequence $A(n) = a_3a_2a_1a_0$ (1) with switching activity $F(A) = 30$.

The value of m is equal to 4; accordingly, $F(A) = 30$ belongs to the required range from 15 to 53. Applying the algorithm described above, we obtain the values of the digits $w(\mathrm{v}_3) = 2$, $w(\mathrm{v}_2) = 2$, $w(\mathrm{v}_1) = 2$, and $w(\mathrm{v}_0) = 2$ of the partition (8) corresponding to the partition 88442211 of the integer 30. However, an attempt to find the corresponding matrix of maximum rank in which all rows contain two ones fails for $m = 4$. This is due to the fact that in this case the requirement for the linear independence of the vectors v_3, v_2, v_1, and v_0 and their weights $w(\mathrm{v}_3)$, $w(\mathrm{v}_1)$, $w(\mathrm{v}_2)$, and $w(\mathrm{v}_0)$ are inconsistent. At the same time, the partition 884222211 of the number 30 determines the digits $w(\mathrm{v}_3) = 2$, $w(\mathrm{v}_2) = 4$, $w(\mathrm{v}_1) = 1$, and $w(\mathrm{v}_0) = 2$ of the partition (8) for which matrices with maximum rank already exist.

The above example shows that obtaining one partition of an integer into summands is not a difficult task. In turn, the generation of the generating matrix V may require the presence of a larger number of integer partitions obtained by modifying the original one. By analogy with Young diagrams [25], to formalize the procedure for modifying the partition of a number into summands, we will determine the diagram of the partition (8) that takes into account all the previously formulated constraints.

Definition 1. The diagram of the expansion (8) of an integer belonging to the range from 2^{m-1} to $m \cdot 2^m - 2^{m-1} - m + 1$ is a matrix consisting of $m \times m$ cells, where each filled cell in the ith row, $i \in \{0, 1, 2, \ldots, m-1\}$, corresponds to the integer 2^{m-1-i}. There are no flush-left empty rows in the matrix, and their filling corresponds to the partition of the integer.

Figure 1 shows two partition diagrams for the case of integer 30.

The presented diagrams show that the sum of values of the filled cells in both cases equals the number 30, and their filling corresponds to the partitions 88442211 and 884222211 into summands 8, 4, 2, and 1. An analysis of the example shows that diagram 1b can be obtained from diagram 1a by removing one cell 4 and filling in two empty cells with 2. This procedure is equivalent to removing the term 4 from the partition 88442211 of the number 30 and adding two terms equal to 2 to obtain the partition 884222211 of the same number; this allows us to define the operation of modifying the partition corresponding to Definition 1.

Operation of modification. For the ith ($i = 0, \ldots, m-2$) row of the diagram of partition (8) containing more than one filled cell, deleting a filled cell is associated with filling 2^j free cells in the $(i+j)$th ($i+j = 1, \ldots, m-1$) row of the diagram, where $j \le \log_2(m-1)$.

This operation is symmetric with respect to the delete and fill operations. This means that deleting 2^j filled cells in the ith ($i = 1, \ldots, m-1$) row of the diagram containing more than 2^j filled cells is associated with filling one cell in the $(i-j)$th row ($i-j = 0, \ldots, m-2$), where $j \le \log_2(m-1)$.

5. PROBLEMS OF SYNTHESIZING SEQUENCES WITH GIVEN SWITCHING ACTIVITY

Taking into account the wide range of application of test sequences $A(n)$ [19, 21, 22, 29, 30], we will formulate the problems of synthesis of generators of such sequences. The result of the synthesis, as mentioned earlier, will be a generating matrix V that provides the values of switching activity $F_{\mathrm{av}}(A)$ and $F_{\mathrm{av}}(a_j)$ of the sequence $A(n) = a_{m-1}a_{m-2}\ldots a_0$, $a_i \in \{0,1\}$, $i \in \{0,1,2,\ldots,m-1\}$ and $n \in \{0,1,2,\ldots,2^{m-1}\}$.

Problem 1. Synthesize a device that generates a sequence $A(n)$ for a given value of m and a required value of $F_{\mathrm{av}}(A)$.

An example of such a problem can be the problem of generating a double Gray sequence, i.e., a sequence for which only two switchings are performed during the transition from the current test set to the next one. The solution to Problem 1 will be:

(i) Obtain the value of $F(A) = \mathrm{int}\,[F_{\mathrm{av}}(A) \times (2^{m-1})]$, where int means the operation of obtaining the integer part of the number in brackets.

(ii) Partition the integer $F(A)$.

(iii) Obtain the values $w(\mathrm{v}_i)$ of row weights of the desired generating matrix V and find the maximum-rank matrix with the row weights $w(\mathrm{v}_i)$.

If it is impossible to obtain the desired matrix owing to the inconsistency of the requirements stated for it, the modification of integer partition described earlier is first applied. The next and final step is the *correction* of the value of $F(A)$.

In general, the *correction* operation is used to ensure that $F_{\mathrm{av}}(A)$ is set to a given value with minimal error. To do this, the value of $F(A)$ is initially changed (*corrected*) by a minimum value (+1 or −1), and the corresponding generating matrix V (2) is sought. In case of a negative outcome based on the results of the search for the required matrix, the value of the deviation of $F(A)$ from the required value $\mathrm{int}\,[F_{\mathrm{av}}(A) \cdot (2^m - 1)]$ increases. It should be noted that the correction of $F(A)$ by one introduces an insignificant error, which for real values of m in percentage terms is $(1/(2^m - 1)) \times 100\%$.

Problem 2. Synthesize a device that generates a sequence $A(n)$ for a given value of m in which the specific values of switching activity $F_{\mathrm{av}}(a_{\alpha 1})$, $F_{\mathrm{av}}(a_{\alpha 2})$, $\ldots$, $F_{\mathrm{av}}(a_{\alpha k})$ are defined for $k \le m$ of its bits $a_{\alpha 1}$, $a_{\alpha 2}$, $\ldots$, $a_{\alpha k}$, $\alpha i \in \{0,1,2,\ldots,m-1\}$, $i = 1,\ldots,k$.

Just as in the case of Problem 1, the averages of $F_{\text{av}}(a_{\alpha 1})$, $F_{\text{av}}(a_{\alpha 2})$, $\ldots$, $F_{\text{av}}(a_{\alpha k})$ of switching activities are represented as the total values $F(a_{\alpha 1})$, $F(a_{\alpha 2})$, $\ldots$, $F(a_{\alpha k})$ of the number of switching bits $a_{\alpha 1}$, $a_{\alpha 2}$, $\ldots$, $a_{\alpha k}$ in the sequence $A(n)$. Then $F(a_{\alpha i})$ is converted into an m-digit binary code, $F(a_{\alpha i})_{(10)} = F(a_{\alpha c})_{(2)} = \beta_{\alpha i}(0) \cdot 2^{m-1} + \beta_{\alpha i}(1) \cdot 2^{m-2} + \cdots + \beta_{\alpha i}(m-1) \cdot 2^{0}$. Note that $\beta_{\alpha i}(0)$ is the most significant bit of the resulting binary code, and the code $\beta_{\alpha i}(0)\beta_{\alpha i}(1)\ldots\beta_{\alpha i}(m-1)$ uniquely determines the values of the αith column of the generating matrix V (2). Thus, we calculate the values of all $k \leq m$ columns of the matrix V that determine the switching activities $F_{\text{av}}(a_{\alpha 1})$, $F_{\text{av}}(a_{\alpha 2})$, $\ldots$, $F_{\text{av}}(a_{\alpha k})$.

If the necessary condition for $F(a_{\alpha 1})$, $F(a_{\alpha 2})$, $\ldots$, $F(a_{\alpha k})$ stated as Property 1 is not satisfied, then the *correction* operation is applied.

The next step in solving Problem 2 is to check whether the sufficient condition for the values of switching activities $F(a_{\alpha 1})$, $F(a_{\alpha 2})$, $\ldots$, $F(a_{\alpha k})$, which are the binary codes of the columns of the matrix V, is satisfied. This condition is the linear independence of the columns of the generating matrix. The failure of this condition necessitates the application of the *correction* operations.

Then the remaining columns of the binary matrix V are generated randomly (equiprobably and independently). The columns $\alpha 1$, $\alpha 2$, $\ldots$, αk in this matrix take the given values. The rank of the resulting matrix is determined. In the case of the maximum rank, this matrix is the desired one and is used to construct the generator of the sequence $A(n)$. When obtaining a matrix with a rank different from the maximum value m, the random generation of the remaining columns of the desired matrix V is repeated.

Problem 3. Synthesize a device that generates a sequence $A(n)$ for a given value m in which specific values of the switching activity $F_{\text{av}}(a_{\alpha 1})$, $F_{\text{av}}(a_{\alpha 2})$, $\ldots$, $F_{\text{av}}(a_{\alpha k})$ are defined for $k \leq m$ of its bits $a_{\alpha 1}$, $a_{\alpha 2}$, $\ldots$, $a_{\alpha k}$, $\alpha i \in \{0, 1, 2, \ldots, m-1\}$, $i = 1, \ldots, k$, and the switching activity $A(n)$ is $F_{\text{av}}(A)$.

The well-posedness of Problem 3 implies that $F_{\text{av}}(a_{\alpha 1}) + F_{\text{av}}(a_{\alpha 2}) + \cdots + F_{\text{av}}(a_{\alpha k}) < F_{\text{av}}(A) \leq m - 1/2 + 1/(2^{m+1} - 2)$. At the initial stage, the solution of Problem 3 repeats the solution of Problem 2. Next, the steps of the procedure for solving Problem 1 are executed. The difference is that we partition the integer $F^*(A) = \text{int}\,[F_{\text{av}}(A) \cdot (2^m - 1)] - \text{int}\,, [F_{\text{av}}(a_{\alpha 1}) \cdot (2^m - 1)] - \text{int}\,[F_{\text{av}}(a_{\alpha 2}) \cdot (2^m - 1)] - \cdots - \text{int}\,[F_{\text{av}}(a_{\alpha k}) \cdot (2^m - 1)]$ rather than the numbers $F(A)$. In addition, when obtaining the values $w(\mathrm{v}_i)$ of the row weights of the desired generating matrix V, it is necessary to take into account the row weights of the previously generated k columns.

If it is impossible to obtain the desired matrix owing to the inconsistency of the requirements stated for it, the operation of modification of integer partition is first applied. Subsequently, the *correction* of the values of $F^*(A)$ is performed and lastly, the switching activities $F(a_{\alpha 1})$, $F(a_{\alpha 2})$, $\ldots$, $F(a_{\alpha k})$ are corrected starting from their maximum values. As an example, consider the solution of Problem 3 in a particular case.

Example 2. Synthesize a device that generates the sequence $A(n) = a_5a_4a_3a_2a_1a_0$ for $m = 6$, where the switching activities $F_{\text{av}}(a_1) = 1$, $F_{\text{av}}(a_3) = 1/63$, as well as the value of $F_{\text{av}}(A) = 2$, are defined for the bits a_1 and a_3.

The inequality $F_{\text{av}}(a_1) + F_{\text{av}}(a_3) = 1 + 1/63 < F_{\text{av}}(A) = 2 \leq m - 1/2 + 1/(2^{m+1} - 2) = 5.5079$ satisfied indicates the possibility of constructing a device with given switching activities. Based on the average values of the switching activities $F_{\text{av}}(a_1)$, $F_{\text{av}}(a_3)$, and $F_{\text{av}}(A)$, we obtain $F(a_1) = 63$, $F(a_3) = 1$ and $F(A) = 126$.

The values of $F(a_1)$ and $F(a_3)$ are represented as $F(a_1) = 63_{(10)} = 111111_{(2)}$ and $F(a_3) = 1_{(10)} = 000001_{(2)}$. Accordingly, the values of the first and third columns of the matrix V take the form $\beta_1(0)\beta_1(1)\beta_1(2)\beta_1(3)\beta_1(4)\beta_1(5) = 111111$ and $\beta_3(0)\beta_3(1)\beta_3(2)\beta_3(3)\beta_3(4)\beta_3(5) = 000001$. The value $F^*(A) = F(A) - F(a_1) - F(a_3) = 126 - 63 - 1 = 62$ is calculated.

Further, using the algorithm described above for partitioning the integer $F^*(A) = 62$, we obtain $w = 0$ and $q = 62_{(10)} = 111110_{(2)}$. Thus, $b_5b_4b_3b_2b_1b_0 = 111110$.

A partition of the integer $F^*(A)$ into summands 2^i is constructed, where $i = 0, \ldots, 5$, each of which is included in the partition $w + b_i = 1 + b_i$ times. Since $q = 111110$, the terms 2^5, 2^4, 2^3, 2^2, and 2^1 occur in the partition of 62 once each, and the term 1 is not included, because only $b_0 = 0$. The value $w+b_i$ determines the value of the digit $w(\mathrm{v}_{m-1-i})$ in the partition (8) of the number $F^*(A)$, which in this case is the Hamming weight of the rows of the desired matrix V, which consists of six rows and six columns excluding the first and third columns and hence allows the zero values of the partition digits.

Next, we randomly form the values of six four-bit binary vectors with Hamming weights equal to $w(\mathrm{v}_0) = w(\mathrm{v}_1) = w(\mathrm{v}_2) = w(v_3) = w(\mathrm{v}_4) = 1$, and $w(\mathrm{v}_5) = 0$, which will determine the values of the remaining (except the first and third) columns of the desired matrix. The maximum of the rank is determined for the matrix thus obtained. In the case of a positive outcome, the matrix is the basis for forming the sequences $A(n)$ (1) with switching activities specified in the statement of Problem 3. If the rank of the matrix is not 6, then the procedure for generating the matrix is repeated; i.e., six four-bit vectors are randomly generated that determine the values of the bits a_5, a_4, a_2, and a_0 of the sequence $A(n) = a_5a_4a_3a_2a_1a_0$. In the case where a result of a certain number of iterations does not result in the desired matrix of maximum rank, the operations of modification and *correction* are successively applied. A solution to the example in Problem 3 can be the matrix

$$V = \begin{vmatrix} \beta_5(0) & \beta_4(0) & \beta_3(0) & \beta_2(0) & \beta_1(0) & \beta_0(0) \\ \beta_5(1) & \beta_5(1) & \beta_3(1) & \beta_2(1) & \beta_1(1) & \beta_0(1) \\ \beta_5(2) & \beta_5(2) & \beta_5(2) & \beta_2(2) & \beta_1(2) & \beta_0(2) \\ \beta_5(3) & \beta_5(3) & \beta_3(3) & \beta_2(3) & \beta_1(3) & \beta_0(3) \\ \beta_5(4) & \beta_5(4) & \beta_3(4) & \beta_2(4) & \beta_1(4) & \beta_0(4) \\ \beta_5(5) & \beta_5(5) & \beta_3(5) & \beta_2(5) & \beta_1(5) & \beta_0(5) \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{vmatrix}.$$

This result was obtained by successive application of modification and correction operations.

6. CONCLUSIONS

A technique for synthesizing generators of test sequences with given switching activity is proposed. The definitions of modification and correction operations for finding the generating matrix of the test generator are given. The problems of synthesis of test sequences with given switching activity are stated, and the ways of their solution and the existing limitations are shown.

REFERENCES

1. Jha, N.K. and Gupta, S., *Testing of Digital Systems*, Cambridge, UK: Cambridge Univ. Press, 2003.
2. Yarmolik, V.N., *Kontrol' i diagnostika vychislitel'nykh sistem* (Control and Diagnostics of Computer Systems), Minsk: Bestprint, 2019.

3. Bushnell, M.L.AgrawalV.D., *Essentials of Electronic Testing for Digital, Memory & Mixed-Signal VLSI Circuits*, New York: Kluwer, 2000.
4. Sharma, A.K., *Semiconductor Memories: Technology, Testing, and Reliability*, London: John Wiley & Sons, 2002.
5. Wang, S. and Gupta, S.K., An automatic test pattern generator for minimizing switching activity during scan testing activity, *IEEE Trans. Comp. Aided Des. Int. Cirs. Syst.*, 2002, vol. 21, no. 8, pp. 954–968.
6. Goor, A.J., Kukner, H., and Hamdioui, S., Optimizing memory BIST Address Generator implementations, *Proc. 6th Int. Conf. Des. & Tech. Integr. Syst. Nanoscale Era (DTIS)* (2011), pp. 572–576.
7. Pedram, M., Power minimization in IC design: principles and applications, *ACM Trans. Des. Autom. Electr. Syst.*, 1996, vol. 1, pp. 3–56.
8. Murashko, I.A. and Yarmolik, V.N., *Vstroennoe samotestirovanie. Metody minimizatsii energopotrebleniya* (Built-In Self-Testing. Energy Minimization Methods), Minsk: Bestprint, 2008.
9. Girard, P., Guiller, L., Landrault, S., et al., A test vector ordering technique for switching activity reduction during test operation, *Proc. Ninth Great Lakes Symp. VLSI* (1999), pp. 24–27.
10. Bellaouar, A. and Elmasry, M., *Low-Power Digital VLSI Design Circuits and Systems*, New York: Springer, 1996.
11. Huang, R., Sun, W., Xu, Y., et al., A survey on adaptive random testing, *IEEE Trans. Soft. Eng.*, 2015, vol. 14, no. 8, pp. 1–36.
12. Mrozek, I. and Yarmolik, V.N., Iterative antirandom testing, *J. Electr. Test: Theory Appl.*, 2012, vol. 9, no. 3, pp. 251–266.
13. Chen, T.Y., Kuo, F.C., Merkel, R.G., et al., Adaptive Random Testing: the ART of test case diversity, *J. Syst. Software*, 2010, vol. 83, no. 1, pp. 60–66.
14. Mrozek I., Yarmolik V.N.Antirandom test vectors for BIST in Hardware. Software systems, in *Fundamenta Informaticae,* 2012, no. 119, pp. 1–23.
15. Yarmolik, S.V. and Yarmolik, V.N., Controlled random tests, *Autom. Remote Control*, 2012, vol. 73, no. 10, pp. 1704–1714.
16. Du, X., Mukherjee, N., Cheng, W.T., et al., Full-speed field-programmable memory BIST architecture, *Proc. IEEE Int. Test Conf.* (2005), pp. 1173–1182.
17. Aswin, A.M. and Ganesh, S.S., Implementation and validation of memory built in self-test (MBIST)—survey, *Int. J. Mech. Eng. Tech.*, 2019, vol. 10, no. 3, pp. 153–160.
18. Yarmolik, V.N. and Yarmolik, S.V., The repeated nondestructive march tests with variable address sequences, *Autom. Remote Control*, 2007, vol. 68, no. 4, pp. 126–137.
19. Yarmolik, V.N. and Shevchenko, N.A., Formation of address sequences with a given switching activity, *Informatika*, 2020, vol. 17, no. 1, pp. 7–23.
20. Sobol, I.M., *Tochki, ravnomerno zapolnyayushchie mnogomernyi kub* (Points Uniformly Filling a Multidimensional Cube), Moscow: Znanie, 1985.
21. Antonov, I.A. and Saleev, V.M., An economic method of computing LP_τ-sequences, *USSR Comput. Math. Math. Phys.*, 1979, vol. 19, no. 1, pp. 252–256.
22. Yarmolik, S.V. and Yarmolik, V.N., Quasirandom testing of computing systems, *Informatika*, 2013, no. 3(39), pp. 65–81.
23. Savage, C., A survey of combinatorial Gray code, *SIAM Rev.*, 1997, vol. 39, no. 4, pp. 605–629.
24. Pomeranz, I., An adjacent switching activity metric under functional broadside tests, *IEEE Trans. Comput.*, 2013, vol. 62, no. 4, pp. 404–410.
25. Knuth, D.E., *The Art of Computer Programming. Vol. 4A. Part 1. Combinatorial Algorithms*, Upper Saddle River, NJ: Addison-Wesley, 2011. Translated under the title: *Iskusstvo programmirovaniya. T. 4A. Kombinatornye algoritmy. Ch. 1*, Moscow: Williams, 2013.

26. McKay, J.K.S., Algorithm 371: Partitions in natural order [A1], *Commun. ACM,* 1970, vol. 13, no. 1, p. 52.

27. Stojmenović, I. and Zoghbi, A., Fast algorithms for generating integer partitions, *Int. J. Comput. Math.,* 1998, vol. 70, no. 2, pp. 319–332.

28. Ferreira, P., Jesus, B., Armando, J.V., and Pinho, J., The rank of random binary matrices and distributed storage applications, *IEEE Commun. Lett.*, 2013, vol. 17, no. 1, pp. 151–154.

29. Shauchenka, M., Address sequence generator for memory BIST, *SSRG–IJCSE*, 2019, vol. 6, no. 11, pp. 22–26.

30. Shauchenka, M., Address sequence generator for memory BIST investigation, *Int. Sci. J. Sci. Tech. Union Mech. Eng.—Math. Model.*, 2020, vol. 4, no. 1, pp. 7–8.

This paper was recommended for publication by A.N. Sobolevskii, a member of the Editorial Board