Security of Infocommunication Systems and Internet of Things

2025 Vol 3, No 2

https://doi.org/10.31861/sisiot2025.2.02016

Received 11 November 2025; revised 21 December 2025; accepted 29 December 2025; published 30 December 2025

Organization of Structural Means for Generating Sequences of Pseudorandom Equiprobable Binary Sets

Ihor Yermolenko* **and Anton Zhurba**

Department of System Programming and Specialized Computer Systems, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine

*Corresponding author (E-mail: yermolenkomail@gmail.com)

ABSTRACT The paper presents the results of research and development of structural methods for hardware generation of pseudorandom equiprobable binary sequences. The features and limitations of existing approaches to the synthesis of pseudorandom sequence generators are analyzed, particularly those associated with fixed probability distributions or predetermined set cardinality. A new structural generation method based on cascade dichotomic decomposition of the number of possible combinations in the output sequence is proposed. This approach enables the formation of a complete set of equiprobable binary patterns of a given bit length while reducing hardware complexity. An algorithm for constructing the generator has been developed, which includes the formation of the decomposition tree, the synthesis and minimization of combinational circuits implementing parity functions, and the determination of transition probabilities between the vertices of the decomposition graph. The algorithm is presented in a formalized form, allowing for simplified implementation and analysis. It is shown that the proposed method allows calculating transition probabilities through simple ratios between decomposition components, significantly simplifying the circuit implementation of the generator. The proposed generator structure provides the formation of equiprobable pseudorandom sets regardless of the set's cardinality. It is proven that the performance of such a generator depends not on the size of set *N* but on its binary logarithm, i.e., the bit width *m* of the generated code. The practical implementation of the method is illustrated by an example for, including the calculation of probabilities and construction of parity functions. The obtained results can be applied in the design of high-performance systems for digital device testing, modeling tools for fault-tolerant multiprocessor systems, and random data generators in digital computing technology.

KEYWORDS GL-models, fault-tolerant multiprocessor systems, generators.

I. INTRODUCTION

The relevance and importance of solving problems of forming probabilistic (including pseudorandom (PR)) sequences have led to the development of a wide class of specialized digital circuits, devices, and systems for obtaining such sequences. Statistical methods are widely applied in various fields of computer engineering. In particular, they form the basis for digital circuit diagnostics, probabilistic diagnostic methods, and reliability assessment of fault-tolerant multiprocessor systems by conducting statistical experiments using GL-models [1-3].

A GL-model represents an undirected graph in which each edge corresponds to a Boolean function. The Boolean function takes arguments x_i, each of which represents the state of the corresponding element in the system: $x_i = 0$ means that the element has failed, while $x_i = 1$ indicates that the element remains operational. If the Boolean function of an edge equals zero, the corresponding edge is removed from the graph. The loss of graph connectivity signals a system failure.

GL-models can be used to simulate the behavior of fault-tolerant multiprocessor systems within a failure flow and, accordingly, to calculate their reliability parameters through statistical experiments with models [4]. During the experiment, a sequence of binary vectors is generated, where 1 represents an operational element and 0 represents a failed one. The GL-model responds to each generated vector by determining whether the system remains operational (i.e., whether the graph retains its connectivity). The simpler the model, the more experiments can be performed, and consequently, the higher the statistical accuracy of the results.

To generate binary vectors modeling the system state, a specialized PR vector generator is used [5-9]. For a basic system, it is sufficient to test all equilibrium vectors in which the number of zeros corresponds to the system's fault tolerance condition. Generators can be implemented both in hardware and in software. They may be built on purely mathematical methods or developed using technical solutions that can later be adapted as software implementations. In systems with a small number of processors, this task can be solved by exhaustive enumeration of all possible vectors; however, for systems containing tens or hundreds of processors, statistical modeling becomes essential.

The existing approaches to the synthesis of structural means for obtaining sequences of PR sets are, as a rule, characterized by the equal probability of any *n*-bit binary set from the full set 2^n [10] or from the set of sets with a given (fixed) weight $k \in \{1, n\}$. In works [11], the theoretical aspects of using the sequential decomposition of arbitrary positive integers *N* specifying the number of distinct output sets have been examined in detail.

The decomposition procedure can be represented in the form of a tree-like graph whose vertices are integers – the results of successive dichotomic division of the graph components in the range from *N* to 1.

II. TREE-LIKE GRAPH OF DICHOTOMIC DECOMPOSITION

When solving circuit design problems of the generator of PR sequences of equiprobable m-bit binary sets from a given set of power N, where $m > \log_2 N > m-1$, it is sufficient to specify the procedure for performing transitions (selection of summands) in the tree-like graph. In Fig. 1, the initial part of such a graph with the corresponding transition probabilities is presented. Let us assume that:

$$(r_0 = 0) \rightarrow (P_{11} = P_{12} = 1/2),$$
$$(r_0 = 1) \rightarrow (P_{11} = a_{11}/a_0, P_{12} = a_{12}/a_0 = 1 - P_{11}).$$

For the initial decomposition level ($i = 1$, $a_0 = N$), we introduce a Boolean variable $b_0 = r_0$ as an indicator of the parity of the component a_0. Then, for the next stage (level) of decomposition $i = 2$, i.e., to obtain the summands $a_{21} - a_{24}$, we construct Table 1. If we introduce one more Boolean variable b_1 to define the transition probabilities $P_{21} - P_{24}$, then the following set of conditions can be written:

$$\begin{cases} [(b_0 = 0) \,\&\, (b_1 = 0)] \rightarrow [P_{21} = P_{22} = 1/2], \\ \dots \\ [(b_0 = 1) \,\&\, (b_1 = 1)] \rightarrow [P_{23} = a_{23}/a_{12}, P_{24} = 1 - P_{23}]. \end{cases}$$

TABLE 1. Transition Probability Conditions.

r_{11}	r_{12}	Probability
0	0	$P_{21}=P_{22}=1/2$, $P_{23}=P_{24}=1/2$
0	1	$P_{21}=P_{22}=1/2$, $P_{23}=a_{23}/a_{12}$, $P_{24}=1-P_{23}$
1	0	$P_{21}=a_{21}/a_{11}$, $P_{22}=1-P_{21}$, $P_{23}=P_{24}=1/2$
1	1	$P_{21}=a_{21}/a_{11}$, $P_{22}=1-P_{21}$, $P_{23}=a_{23}/a_{12}$, $P_{24}=1-P_{23}$

III. FUNCTION OF PARITY COMPONENTS

Each level i of the dichotomic decomposition of a given number N into summands can be associated with a certain Boolean variable b_i, as well as with a parity function f_i of the decomposition components, such that:

$$\forall i = 1, 2, \dots, m[f_i = f_i(b_0, b_1, b_2, \dots, b_{i-1})].$$

Let us take, for concreteness, $N = 7$. Fig. 2 shows the structure of the decomposition tree, the parity functions corresponding to its levels, as well as the Boolean variables b_0, b_1, and b_2, which form the output set of the generator and determine the transition (path selection) through the tree-like decomposition graph. Boolean variable b_i, as well as with a parity function f_i of the decomposition components, such that:

$$\forall i = 1, 2, \dots, m[f_i = f_i(b_0, b_1, b_2, \dots, b_{i-1})].$$

Let us take, for concreteness, $N = 7$. Fig. 2 shows the structure of the decomposition tree, the parity functions corresponding to its levels, as well as the Boolean variables b_0, b_1, and b_2, which form the output set of the generator and determine the transition (path selection) through the tree-like decomposition graph.

The beginning of the construction of the decomposition tree in general form (that is, for any positive integer $N > 1$) is shown in Fig. 3.

Under the assumption introduced above concerning the transitions in the graph depending on the parity of the decomposition components, we obtain:

$$(r(N) = 0) \rightarrow (P(a_1) = P(a_2) = ½),$$
$$(r(N) = 1) \rightarrow (P(a_1) = a_1/N, P(a_2) = 1 - P(a_1)).$$

Then the parity function f_0 will coincide with the value $r(N)$, and the bit b_0 can be obtained at the output of the switch S as shown in Fig. 4. According to Fig. 4, one can write:

$$\begin{cases} (b_0 = 0) \rightarrow (f_1 = r(a_1)), \\ (b_0 = 1) \rightarrow (f_1 = r(a_2)). \end{cases}$$

Two levels of a certain decomposition tree are shown in Fig. 5, where $i = 0, 1, 2, \dots m-1$, $k = 1, 2, \dots 2^i$, and also $a_{01} = N$.

The parity function f_i corresponding to the i-th level is obtained based on the data given in Fig. 5 and represented as a truth table (Table 2).

TABLE 2. Parity Function Truth Table.

$b_0\ b_1 \dots b_{i-1}$	f_i
0 0 … 0	r_{i1}
0 0 … 1	r_{i2}
…	…
1 1 … 0	$r_{i(k-1)}$
1 1 … 1	r_{ik}

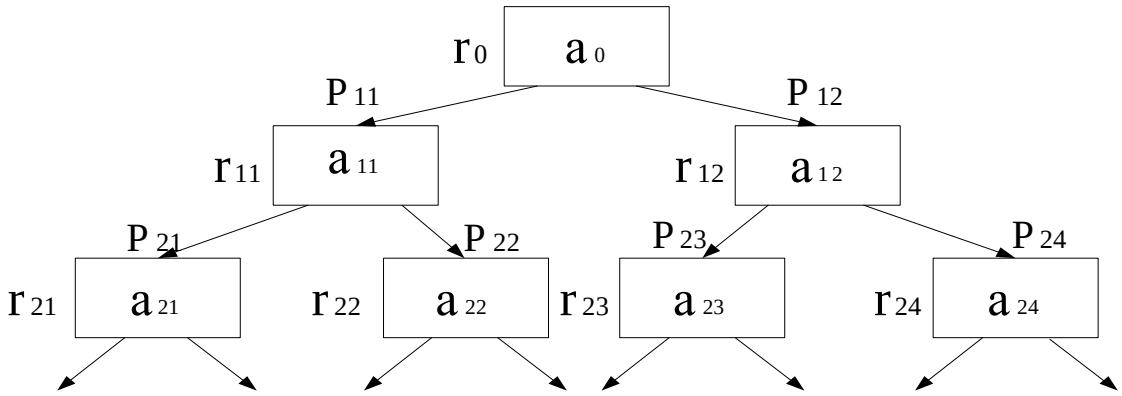


FIG. 1. Tree-like graph of dichotomic decomposition with transition probabilities.

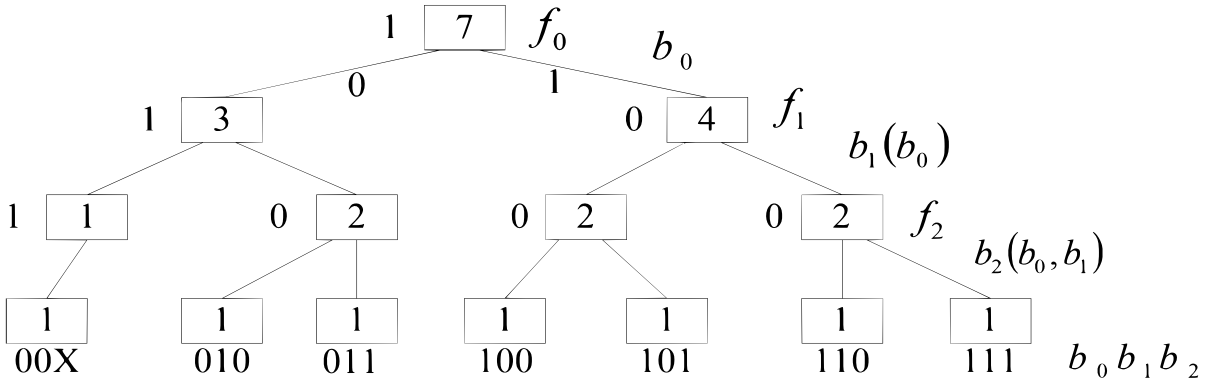


FIG. 2. Dichotomic decomposition tree.

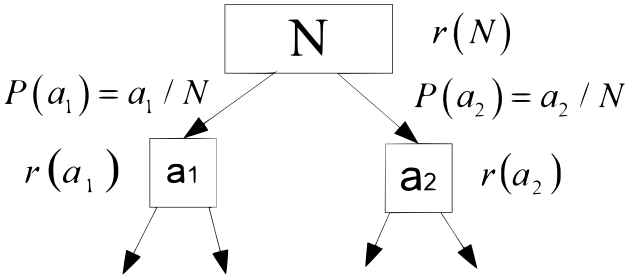


FIG. 3. Initial stage of the dichotomic decomposition tree.

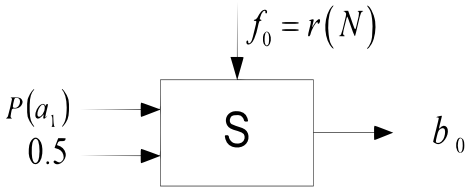


FIG. 4. Scheme of generating bit b_0 using parity function f_0.

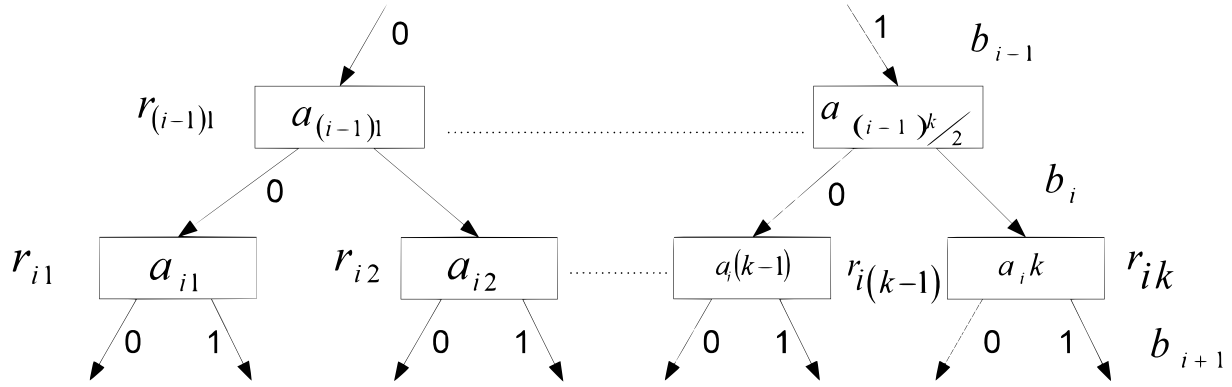


FIG. 5. Two consecutive levels of the dichotomic decomposition tree.

Fig. 6 shows the scheme for obtaining the bit b_i of the output set. As shown above, the probability P_i of transition in the graph can be calculated according to the relation: $P_i = aij / a_{(i-1)j/2}$, where $i = 1, 2, \ldots m$, $j = 1, 2, \ldots, 2^i$. Thus, each bit is formed based on the set of values of bits b_0, b_1, …, b_{i-1}, that is:

$$\forall i = 1, 2, \ldots m-1\,[b_i = f(b_0, b_1, \ldots, b_{i-1})].$$

The general structure of the generator output unit is shown in Fig. 7, where C_1, C_2, …, $C_{(m-1)}$ are combinational circuits implementing the corresponding Boolean parity functions.

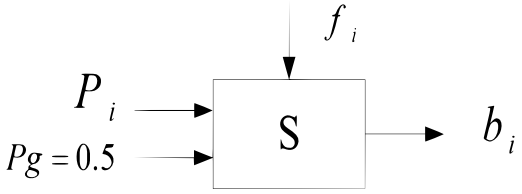


FIG. 6. Scheme of generating bit b_i using parity function f_i.

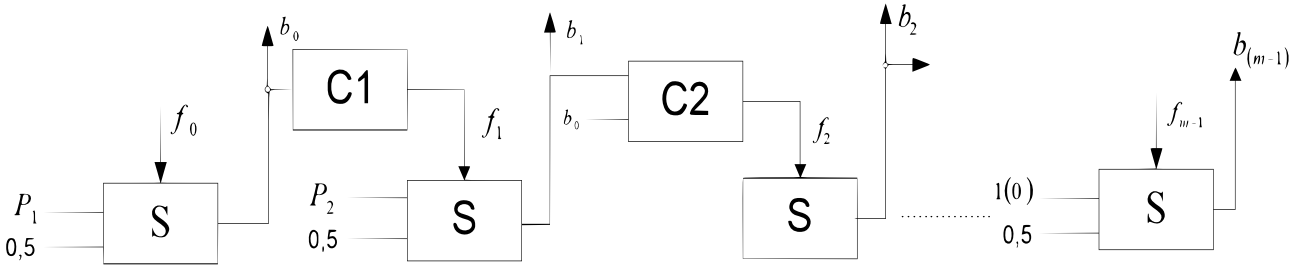


FIG. 7. General structure of the generator output unit.

IV. ALGORITHM OF GENERATOR CONSTRUCTION

The sequence of synthesis of a generator circuit for producing equiprobable PR sets for an arbitrary positive integer N can be represented in the form of the following algorithm:

1. Set (select) the value of N;
2. Construct the decomposition tree of the number N, as shown, for example, in Fig. 2;
3. Perform the synthesis and minimization of m combinational circuits that implement the parity functions $f_0 - f_{m-1}$ according to the truth tables (see Table 2);
4. Compute the probabilities P_i, $i = 1, 2, \ldots m$, by determining for each i the value $max(a_{i1}, a_{i2}, \ldots, a_{ik})$, after which find $P_i = a_{ij} / a_{(i-1)j/2}$.

As an example, consider $N = 11$. The decomposition tree for this case is shown in Fig. 8.

Since $f_0 = r(N) = 1$, then $P(b_0 = 1) = 6/11$. Considering that $(b_0 = 0) \rightarrow (f_1 = 1)$ and $(b_0 = 1) \rightarrow (f_1 = 0)$, we can find the next probabilities. Then $(f_1 = 0) \rightarrow [P(b_1 = 1) = 1/2]$, $(f_1 = 1) \rightarrow [P(b_1 = 1) = 3/5]$, where $P(b_i = 1)$ is the probability of the unit state of bit b_i.

The parity function f_2 is determined from the truth table (Table 3) and the corresponding combinational circuit can be constructed based on the Boolean equation. After that, bit b_2 is determined by the conditions:

$$(f_2 = 0) \rightarrow \left[P(b_2) = 1/2\right],$$
$$(f_2 = 1) \rightarrow \left[P(b_2) = 2/3\right].$$

TABLE 3. Parity function f_2 truth table.

b_0	b_1	f_2
0	0	0
0	1	1
1	0	1
1	1	1

For the function f_3, we construct a truth table (Table 4) and obtain:

$$\overline{f_3} = \overline{b_0} \cdot b_1 \cdot b_2 \vee b_0 \cdot \overline{b_1} \cdot b_2 \vee b_0 \cdot b_1 \cdot b_2 =$$
$$= \overline{\overline{b_2} \vee \overline{b_0} \cdot \overline{b_1}} = b_2 (b_0 \vee b_1),$$

from which, finally, it follows that:

$$f_3 = \overline{b_2} \vee \overline{b_0} \cdot \overline{b_1} .$$

TABLE 4. Parity function f_3 truth table.

b_0	b_1	b_2	f_3
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

The value of bit b_3 is determined from the following conditions:

$$(f_3 = 0) \rightarrow \left[P(b_3) = 1/2\right],$$
$$(f_3 = 1) \rightarrow \left[P(b_3) = 1\right].$$

V. GENERAL STRUCTURE OF THE GENERATOR

Practical interest during the synthesis of the general structure of the generator shown in Fig. 9, in addition to the characteristics already considered, is also represented by the circuit implementation of the set of probabilities P_i, $i = 1, 2, \ldots m - 1$ for a specific value of N.

As shown above:

$$\forall i = 1, 2, \ldots m-1 \left[P_i = \frac{a_{ij}}{a_{(i-1)\,j/2}} \right],$$

where $2a_{ij} \neq a_{(i-1)j/2}$, and also $a_{ij} = max\,(a_{i1}, a_{i2}, \ldots a_{ik})$, $k = 2^i$.

Let $P_i = ½ + P_{iД}$. It is assumed that the signal of the PR pattern generator Pg and the state of the output of the circuit producing $P_{iД}$ are statistically independent.

Consider the decomposition sequence of the number $N = 11$ into a set of summands (Figure 8):

$$\{11\} \rightarrow \{5, 6\} \rightarrow \{2, 3\} \rightarrow \{1, 2\} \rightarrow \{1\}.$$

Then $P_1 = 6/11$, $P_2 = 3/5$, $P_3 = 2/3$, $P_4 = 1$, and in addition $P_{2Д} = 1/22$, $P_{2Д} = 1/10$, $P_{3Д} = 1/6$, that is $\forall_i = 1, 2, \ldots m - 1$ [$P_{iДmin} = P_{1Д} = 1/2N$]. Therefore, the minimal value $P_{1Д}$ may serve as the basis for defining the precision of representation of other $P_{iД}$ values. For $N = 11$ in binary format $P_{1Д} = 0.000010111\ldots$. We take the

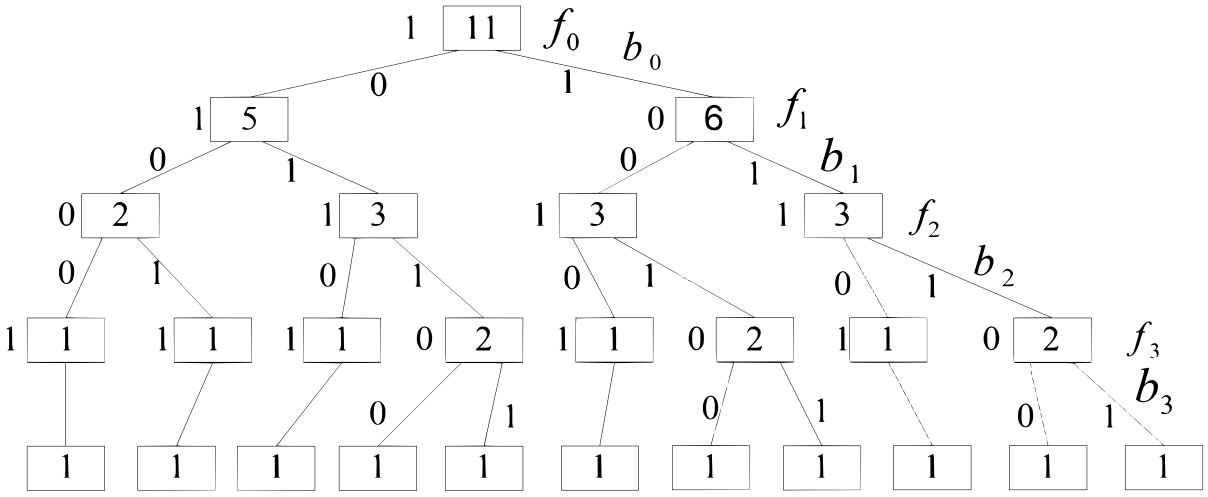


FIG. 8. Example of the decomposition tree for $N = 11$.

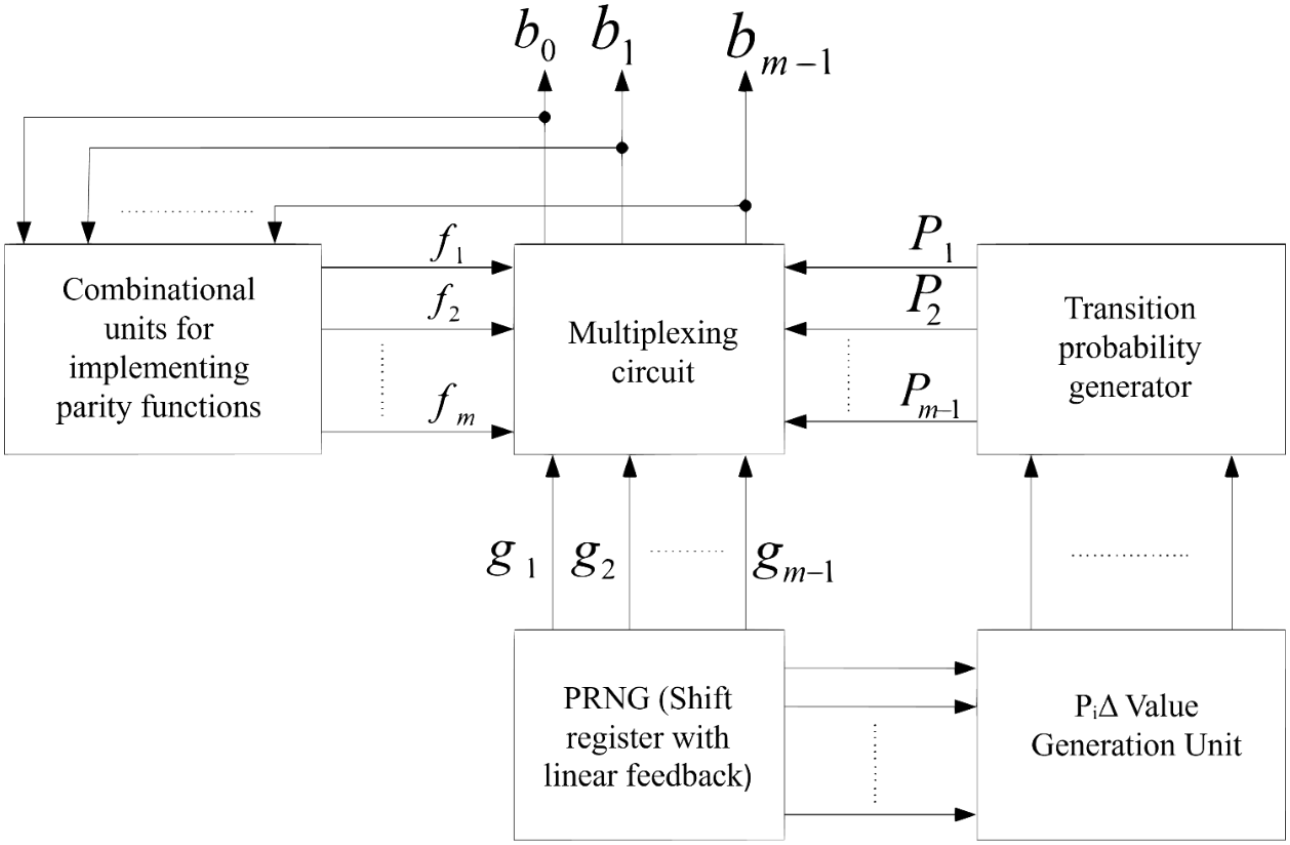


FIG. 9. General structure of the sequence generator.

rounded value $P_{1\Delta} = 0.000011$. Hence, with the same precision (up to the sixth digit) $P_{1\Delta} = 1/32 + 1/64$. Then, with the same precision $P_{2\Delta} \approx 0.0001100 = 1/16 + 1/32$ and $P_{3\Delta} \approx 0.001001 = 1/8 + 1/64$. In the general case, if the precision of setting the probabilities $P_{1\Delta}$ equals q binary digits, and $a_{imax} = m - 1$, then it is advisable to use a PR pattern generator [11] with a register length $R \geq q + m - 1$.

Obviously, any probability value of the form $1/2^j$, $j = 1,2, \ldots$ can be obtained by the conjunction of j independent outputs of the feedback shift register.

The circuitry for generating probabilities $P_{1\Delta}$ can be simplified either by rounding the exact value $P_{1\Delta}$ to the nearest value $1/2^r$ (r – integer positive number) or by replacing the arithmetic sum with a disjunction operation of the corresponding probability terms. At the same time, one must take into account the error introduced by such circuit implementation.

VI. CONCLUSION

The results obtained in this work make it possible at the design stage to implement specialized digital structures intended for hardware generation of PR sequences of binary sets forming a set with a predetermined cardinality.

Unlike known devices, the use of the method of successive dichotomic decomposition of the number determining the cardinality of the set of different output sets leads to improvement in several parameters of the generating structures.

For example, from the materials of the work it follows that the performance of the generator is determined not by the set cardinality N, but by the binary logarithm of this number, that is, by the bit width m of the generated sets.

AUTHOR CONTRIBUTIONS

I.Y. – writing (original draft preparation), conceptualization, methodology, investigation; A.Z. – writing (original draft preparation), conceptualization, methodology, investigation.

COMPETING INTERESTS

The authors declare no competing interests.

REFERENCES

[1] I. V. Maidanyuk, K. V. Morozov, E. R. Potapova, and A. V. Shuryga, “On one property of a GL-model with a minimum number of lost edges,” *Scientific Bulletin of Chernivtsi University. Series: Computer Systems and Components*, vol. 1, no. 2, pp. 31–34, 2010.

[2] K. V. Morozov, E. R. Potapova, and N. K. Kichigin, “GL-model of a hierarchical system with processors at all hierarchy levels,” *Proceedings of the All-Ukrainian Scientific and Practical Conference*, pp. 35–37, 2016.

[3] E. R. Potapova, A. V. Shuryga, and I. V. Maidanyuk, “On one method for modifying the edge functions of a GL-model,” *Information-Control Systems in Railway Transport*, no. 4, p. 48, 2012.

[4] I. A. Yermolenko, “Method for constructing GL-models for consecutive-k-within-m-out-of-n systems,” *Computer-Integrated Technologies: Education, Science, Production*, no. 61, pp. 6–11, 2025, doi: 10.36910/6775-2524-0560-2025-61-01.

[5] F. Panneton, P. L’Ecuyer, and M. Matsumoto, “Improved long-period generators based on linear recurrences modulo 2,” *ACM Transactions on Mathematical Software*, vol. 32, no. 1, pp. 1–16, 2006.

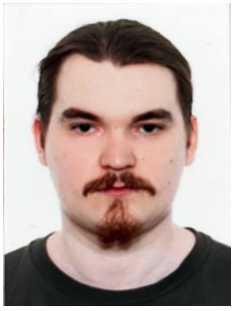
[6] S. Sánchez, R. Criado, and C. Vega, “A generator of pseudo-random number sequences with a very long period,” *Mathematical and Computer Modelling*, vol. 42,

nos. 7–8, pp. 809–816, 2005, doi: 10.1016/j.mcm.2005.09.009.
[7] V. A. Romankevich and I. V. Maidanyuk, “Structural method for forming binary pseudorandom vectors of a given weight,” *Upravlyayushchiye Sistemy i Mashin*, no. 5, pp. 28–33, 58, 2011.
[8] A. M. Romankevich, I. V. Maidanyuk, and V. A. Romankevich, “On the formation of control functions for a generator of binary vector sequences,” *Radio Electronic and Computer Systems*, no. 6, pp. 157–163, 2014.
[9] V. V. Grol, V. A. Romankevich, E. R. Potapova, and S. M. Moravedge, “Structural method of generating pseudorandom sequences of a special type,” *Radio Electronic and Computer Systems*, no. 5, pp. 230–236, 2010.
[10] R. P. Brent, “Some long-period random number generators using shifts and xors,” *The Proceedings of ANZIAM*, vol. 48, pp. C188–C202, 2006.
[11] R. Al Shboul and V. A. Romankevich, “Structural means generating pseudorandom sequences of fixed-weight binary patterns,” *International Journal of Computer Science & Network Security*, vol. 17, no. 10, pp. 62–66, 2017.

Ihor Yermolenko

PhD student, System Programming and Special Computer System Department. National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”. Research field: GL-models; Fault-tolerant multiprocessor systems reliability estimation.

ORCID ID: 0009-0008-5298-4888



Anton Zhurba

PhD student, System Programming and Special Computer System Department. National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”. Research field: GL-models; Fault-tolerant multiprocessor systems reliability estimation.

ORCID ID: 0009-0007-3375-2590

Організація структурних засобів формування послідовностей псевдовипадкових рівноймовірних двійкових наборів

Ігор Єрмоленко*, Антон Журба

Кафедра системного програмування і спеціалізованих комп’ютерних систем, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

*Автор-кореспондент (Електронна адреса: yermolenkomail@gmail.com)

АНОТАЦІЯ У роботі представлено результати дослідження та розроблення структурних методів апаратної генерації псевдовипадкових рівноймовірних двійкових послідовностей. Проаналізовано особливості існуючих підходів до синтезу генераторів псевдовипадкових наборів і визначено їхні обмеження, пов’язані з фіксованим розподілом імовірностей або жорстко заданою потужністю вихідної множини. Запропоновано новий метод побудови структурних засобів генерації, який ґрунтується на каскадному дихотомічному розкладанні числа можливих комбінацій вихідної послідовності. Такий підхід забезпечує формування повної множини рівноймовірних наборів заданої розрядності при зменшенні апаратних витрат. Розроблено алгоритм побудови генератора, який охоплює етапи формування дерева розкладання, синтезу та мінімізації комбінаційних схем, що реалізують функції парності, а також визначення ймовірностей переходів між вершинами графа розкладання. Опис алгоритму подано у формалізованому вигляді, що спрощує його подальшу реалізацію та аналіз. Показано, що використання запропонованого методу дозволяє визначати ймовірності переходів через прості відношення між компонентами розкладання, що значно спрощує схемотехнічну реалізацію генератора. Запропонована структура формувача забезпечує отримання рівноймовірних псевдовипадкових наборів незалежно від потужності вихідної множини. Доведено, що швидкодія такого генератора визначається не розміром множини N, а її двійковим логарифмом, тобто розрядністю вихідного коду m. Практичну реалізацію методу проілюстровано на прикладі, наведено способи обчислення ймовірностей та побудови функцій парності. Отримані результати можуть бути використані під час проектування високопродуктивних систем тестування цифрових пристроїв, засобів моделювання відмовостійких багатопроцесорних систем і генераторів випадкових даних у цифровій обчислювальній техніці.

КЛЮЧОВІ СЛОВА GL-моделі, відмовостійкі багатопроцесорні системи, генератори.