DOI: https://doi.org/10.15276/hait.09.2026.03
UDC 004.05

A method for constructing non-basic GL-models by combining models with arbitrary graph structures

Vitaliy A. Romankevich[1)]
ORCID: https://orcid.org/0000-0003-4696-5935; zavkaf@scs.kpi.ua. Scopus Author ID: 57193263058
Kostiantyn V. Morozov[1)]
ORCID: https://orcid.org/0000-0003-0978-6292; mcng@ukr.net. Scopus Author ID: 57222509251
Alexei M. Romankevich[1)]
ORCID: https://orcid.org/0000-0001-5634-8469; romankev@scs.kpi.ua. Scopus Author ID: 6602114176
Daniil V. Halytsky[1)]
ORCID: https://orcid.org/0009-0004-4421-3443; zipper135401@gmail.com. Scopus Author ID: 58553487600
Lefteris Zacharioudakis[2)]
ORCID: https://orcid.org/0000-0002-9658-3073; l.zacharioudakis@nup.ac.cy. Scopus Author ID: 57422876200
[1)] National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Peremogy Ave. Kyiv, 03056, Ukraine
[2)] Neapolis University Pafos, 2, Danais Ave. Pafos, 8042, Cyprus

ABSTRACT

The paper proposes a method for constructing GL-models of failure behavior for a special class of non-basic fault-tolerant multiprocessor systems. The considered class includes systems whose failure behavior is determined by the fulfillment of one of several conditions that depend on combinations of processor states and are specified by the corresponding Boolean expressions. When any of these conditions is satisfied, the system failure behavior can be described by an individual auxiliary model constructed using known methods. The objective of this study is to develop a generalized method for constructing models of such non-basic systems by combining multiple auxiliary models of arbitrary types into a single model that correctly represents the system failure behavior. The proposed method is based on modifying the edge functions of the auxiliary models using the Boolean expressions of the corresponding conditions, after which the modified models are combined by merging arbitrarily selected vertices of their graphs. It is shown that the resulting model reproduces the behavior of the corresponding auxiliary model under each condition and thus correctly describes the failure behavior of the original system. The scientific novelty of this work lies in the proposed approach to combining auxiliary models of a general form, which imposes no constraints on the structure of their graphs and, unlike existing approaches, does not require these models to be based on cycle graphs. This makes it possible to construct models for a wide class of non-basic fault-tolerant multiprocessor systems with complex operability conditions. A comparative analysis of the complexity of edge-function expressions in models constructed using the proposed method and those obtained by existing approaches is carried out on a set of representative examples, revealing a significant reduction in the complexity of such expressions. The case of parallel evaluation of edge-function values is considered separately, and a reduction in the maximum expression complexity is demonstrated, which is important for decreasing computation time. The practical significance of the work lies in the applicability of the proposed method to constructing models of complex non-basic fault-tolerant multiprocessor systems, as well as in reducing the complexity of analyzing their failure behavior. The presented examples and experimental results confirm the correctness of the constructed models and demonstrate the possibility of further simplification by eliminating edges with identically unit edge functions. The proposed approach can be used in the analysis and design of critical control systems, as well as in automated reliability assessment of non-basic fault-tolerant multiprocessor systems with a large number of components.

Keywords: fault-tolerant multiprocessor systems; GL-models; non-basic systems; control systems; reliability evaluation; statistical experiments

For citation: **Romankevich V. A., Morozov K. V., Romankevich A.M., Halytsky D. V., Zacharioudakis L. "A method for constructing non-basic GL-models with arbitrary graph structures".** ***Herald of Advanced of Information Technology*** **2026; Vol.9 No.1: 30–45. DOI: https://doi.org/10.15276/hait.09.2026.03**

INTRODUCTION

Modern technical systems are characterized by the widespread adoption of automated solutions across various fields of activity. Automation makes it possible to reduce dependence on human involvement in the execution of routine operations and to increase the efficiency and reliability of process operation by minimizing the impact of the human factor. At the same time, in a number of applied problems, direct human involvement in function execution is either impossible or significantly limited due to physiological constraints, including limited reaction speed, restricted capability for parallel information processing, and the ability to control only a limited number of objects. Additional limitations arise from

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

increased risks to life and health, for example, during disaster response to natural and man-made emergencies or in the execution of combat tasks, as well as from the fundamental impossibility of human presence under certain conditions, in particular in the case of long-duration space missions.

One of the key components of automatic and automated systems is the control system (CS), which performs the acquisition of input data from sensors, their processing, and the generation of control actions for the actuators of the controlled object [1], [2]. In modern technical systems, control systems are typically implemented using microprocessor-based solutions. A failure of the control system may lead to a violation of the normal operating mode of the controlled object.

For critical application systems (CAS), whose failures may cause significant material losses, pose threats to human life and health, or lead to adverse consequences for national security, control systems are subject to increased reliability requirements [3], [4], [5]. Accordingly, the control system of such objects must ensure a high level of reliability. Along with stringent reliability requirements, CAS objects are typically characterized by complex structures and algorithmically sophisticated control rules, which necessitate the execution of computationally intensive procedures and, consequently, high computational performance.

In this regard, control systems for CAS are often implemented using fault-tolerant multiprocessor systems (FTMSs) that comprise a large number of processors and are capable of maintaining operability in the presence of failures of individual computing components [6], [7], [8]. Thus, FTMSs provide a combination of high reliability and high performance.

The design of fault-tolerant multiprocessor systems requires the application of formalized methods for evaluating their reliability. Such evaluation is necessary both to verify compliance of the developed systems with specified requirements and to identify potential bottlenecks for subsequent system improvement. At the same time, control systems based on fault-tolerant multiprocessor systems may be characterized by complex and heterogeneous structures, which significantly complicates the procedures for reliability analysis.

LITERATURE REVIEW

The assessment of reliability parameters of fault-tolerant multiprocessor systems can be performed using various approaches, which can be conditionally divided into two groups [9], [10]. Methods of the first group are based on deriving analytical expressions for calculating the corresponding parameters [9], [11], [12], [13]. As a rule, these methods provide more accurate assessments; however, they are not universal, since the development of a dedicated calculation method may be required for each specific system type. In cases where a system simultaneously combines features of several types, the application of existing analytical methods may prove to be infeasible.

Methods of the second group are based on conducting statistical experiments with models of the failure behavior of fault-tolerant multiprocessor systems [14], [15], [16], [17]. Such methods are universal provided that a model of system behavior is available; however, they typically yield reliability parameter assessments with limited accuracy, which, in the general case, is determined by the number of experiments performed.

GL-models can be effectively used as models of the failure behavior of fault-tolerant multiprocessor systems [17]. A GL-model is based on an undirected graph, in which each edge is associated with a Boolean edge function. The arguments of the edge functions in a GL-model are typically the elements of the so-called system state vector – a Boolean vector whose components correspond to the states of individual processors in the system: a value of 1 indicates an operational state, whereas a value of 0 corresponds to a failure. If the corresponding edge function evaluates to zero, the associated edge is removed from the graph. Graph connectivity, in turn, is interpreted as the system state: a connected graph corresponds to an operational system state, whereas a disconnected graph corresponds to a failed state.

Existing methods for constructing GL-models are generally oriented toward so-called basic systems, which are tolerant to the failure of no more than a specified number of arbitrary processors. A basic system and its corresponding basic GL-model, consisting of n processors and tolerant to the failure of no more than m of them, are denoted as $K(n, m)$. The methods described in [18], [19] make it possible to construct GL-models of basic fault-tolerant multiprocessor systems that are based on cycle graphs, which to some extent simplifies the analysis of their connectivity. At the same time, it should be noted that for basic systems, the failure behavior can in many cases be determined relatively easily even without the use of a formalized model.

However, real-world systems, including fault-tolerant multiprocessor control systems, are often non-basic. Such systems include, in particular, consecutive-k-out-of-n systems [13], [20], [21], [22], [23], [24] consecutive-k-out-of-r-from-n systems [24], [25], m-consecutive-k-out-of-n systems [27], [28], [29], consecutive-k-within-m-out-of-n systems [30], [31], consecutive-k_c-out-of-n systems [29], [32], consecutive-k_r-out-of-n_r systems [33], consecutive-(k, l)-out-of-n systems [34], m-consecutive-k,l-out-of-n systems [35], [36], (r, s)-out-of-(m, n) systems [32], [37], [38], [39] (n, f, k) systems [24], [40], [41], $\langle n, f, k\rangle$ systems [41], [42], as well as a number of hierarchical systems [43]. All the systems listed above do not belong to the class of basic systems. Moreover, in practical applications, fault-tolerant multiprocessor systems may be characterized by even more complex operability conditions than those considered above.

The construction of GL-models for non-basic systems can be performed by modifying basic models. Such modifications may be carried out either by changing the structure of the model graph (for example, by introducing additional edges) or by transforming the expressions of the edge functions. However, in certain cases, these modifications prove to be excessively complex, since the system behavior differs substantially from that of a basic system. In particular, under some combinations of operational and failed processors, the system may operate according to one scenario, whereas under other combinations it may follow a different one.

In such situations, it is reasonable to combine several simpler models within a single GL-model, where these simpler models can be constructed using known methods and correspond to different operating conditions of the system. For example, in [44], a method for combining basic GL-models based on cycle graphs is proposed.

On the other hand, a complex system operability condition may be represented as a set of several simpler conditions that must be satisfied simultaneously. In this case, a separate GL-model can be constructed for each such condition. In [45], a method is proposed for constructing a GL-model of the corresponding fault-tolerant multiprocessor system by combining such auxiliary GL-models built for each operability condition of the system.

It should also be noted that a direct analysis of Boolean system state vectors is applicable not only to basic systems; however, in practice it is generally feasible only for systems of very small dimensionality, since the number of possible states grows exponentially with the number of components. Moreover, such an approach does not provide a structured representation of system behavior and does not allow reuse or combination of partial results obtained for different operating conditions. In contrast, an approach based on GL-models represents system behavior in a compact and structured form by means of graphs with edge functions, where system operability is reduced to a graph connectivity problem. This enables the use of well-established graph-theoretic techniques, supports modular construction and combination of models, and simplifies both analysis and further model transformations.

PROBLEM STATEMENT

The method described in [44] enables the construction of GL-models for non-basic systems that, depending on specified conditions, are tolerant to the failure of different numbers of processors. In this sense, under different conditions, the system can be regarded as operating in a manner analogous to a corresponding *conditional* basic system. In the general case, such *conditional* systems may be arbitrary, that is, they do not necessarily belong to the class of basic systems. It is easy to see that the approach described in [44] can be extended to the case of arbitrary *conditional* systems, provided that the corresponding *auxiliary* GL-models are based on cycle graphs, for example, if they are obtained from basic models solely by modifying the expressions of the edge functions.

However, in a number of cases, GL-models of non-basic fault-tolerant multiprocessor systems must be based on graphs other than cycle graphs. In particular, situations in which modifying a basic model solely by transforming edge functions proves to be inefficient and leads to a significant increase in the complexity of the corresponding expressions require the use of more general graph structures. This observation also applies to *auxiliary* GL-models constructed to describe the aforementioned *conditional* systems. In such cases, the application of the method proposed in [44] becomes infeasible.

Thus, the problem of developing a generalized method for constructing GL-models of non-basic systems by combining multiple *auxiliary* models, which imposes no constraints on the structure of the graphs underlying these models, is highly relevant.

RESEARCH AIM AND OBJECTIVES

The objective of this work is to develop a method for constructing GL-models of complex non-basic fault-tolerant multiprocessor systems whose failure behavior, under a set of simple conditions determined by combinations of processor states, can be described as the behavior of simpler systems (not necessarily basic ones). For each such system, the corresponding *auxiliary* GL-models with arbitrary graph structures can be constructed using known methods. It is assumed that each condition is associated with a Boolean expression that evaluates to 1 when the condition is satisfied and to 0 otherwise. It is also assumed that exactly one of the specified conditions holds at any given time, and, accordingly, the failure behavior of the fault-tolerant multiprocessor system is described by the corresponding *auxiliary* GL-model.

To achieve the stated objective, the following tasks are formulated in this work:

1) to develop a method for constructing GL-models of non-basic fault-tolerant multiprocessor systems by combining *auxiliary* GL-models of a general form using Boolean expressions of the corresponding conditions;

2) to develop an algorithm for constructing such GL-models based on the proposed method;

3) to perform experimental verification of the correctness of the GL-models constructed using the proposed method.

METHOD FOR CONSTRUCTING A GL-MODEL OF A NON-BASIC FTMS

Consider a fault-tolerant multiprocessor system whose failure behavior depends on the fulfillment of certain conditions $C_1, C_2, \dots, C_K$, with exactly one of these conditions holding at any given time. When condition C_i is satisfied, the system operates in a manner analogous to a certain *conditional* fault-tolerant multiprocessor system, which is associated with an *auxiliary* GL-model M_i, $i = 1,2, \dots, K$. The conditions $C_1, C_2, \dots, C_K$ depend exclusively on the states of the system processors, and Boolean expressions $c_1(\mathbf{v}), c_2(\mathbf{v}), \dots, c_K(\mathbf{v})$ can be constructed for them such that $c_i(\mathbf{v}) = 1$ for all system state vectors $\mathbf{v}$ corresponding to the fulfillment of condition C_i, and $c_i(\mathbf{v}) = 0$ for all other state vectors.

Note that, to construct a GL-model of the considered fault-tolerant multiprocessor system, it is sufficient to construct a model that, under condition C_i, reproduces the behavior of the *auxiliary* GL-model M_i for all $i = 1,2, \dots, K$.

Let us modify the edge functions of the *auxiliary* GL-models $M_1, M_2, \dots, M_K$ according to the formula

$$\tilde{f}_j^i = f_j^i c_i \vee \bar{c}_i = f_j^i \vee \bar{c}_i, \qquad (1)$$

where i denotes the index of the *auxiliary* GL-model, j denotes the index of the edge function within the model, f_j^i is the *j*-th edge function of the *i*-th *auxiliary* model, and $\tilde{f}_j^i$ is the corresponding modified edge function. The GL-models obtained as a result of this modification will hereafter be referred to as *modified auxiliary* models and denoted by $\widetilde{M}_1, \widetilde{M}_2, \dots, \widetilde{M}_K$.

Consider the *modified auxiliary* GL-model $\widetilde{M}_i$. If condition C_i is satisfied, then $c_i = 1$ and $\bar{c}_i = 0$, and consequently $\tilde{f}_j^i = f_j^i$ for all values of j. Thus, the *modified auxiliary* model $\widetilde{M}_i$ reproduces the behavior of the *auxiliary* GL-model M_i.

If, on the other hand, condition C_i is not satisfied, then $c_i = 0$ and $\bar{c}_i = 1$, and consequently $\tilde{f}_j^i = 1$ for all j. In this case, the model $\widetilde{M}_i$ does not lose any edges and remains connected. It is assumed that each *auxiliary* GL-model is based on an initially connected graph whose connectivity may be violated only as a result of edge removal according to the values of the edge functions, since otherwise such a model would a priori correspond to a failed system state, which would be incorrect given its intended purpose.

Let us combine the graphs of the *modified auxiliary* GL-models $\widetilde{M}_1, \widetilde{M}_2, \dots, \widetilde{M}_K$ according to the method described in [45]. In accordance with this method, one arbitrary vertex is selected from each graph in a pair of such model graphs, after which these vertices are merged into a single common vertex. As a result of this merging, a unified graph is formed. The merging procedure is repeated until all models are combined into a single GL-model M. As shown in [45], the graph of model M is connected if and only if the graphs of all models $\widetilde{M}_1, \widetilde{M}_2, \dots, \widetilde{M}_K$ are connected for the given system state vector.

As shown above, when an arbitrary condition C_i is satisfied, the graphs of all *modified auxiliary* models $\widetilde{M}_j$ for which $i \neq j$ remain connected, whereas the model $\widetilde{M}_i$ reproduces the behavior of the *auxiliary* GL-model M_i. It follows that, under condition C_i, the combined GL-model M also reproduces the behavior of model M_i.

Indeed, if for a certain system state vector under which condition C_i is satisfied the graph of model M_i is connected, then, accordingly, the graph of model $\widetilde{M}_i$ is also connected. In this case, the graphs of all models $\widetilde{M}_1, \widetilde{M}_2, \dots, \widetilde{M}_K$ are connected, and therefore, according to the properties of the model-combination operation described in [45], the graph of the combined model M is connected as well.

If, on the other hand, for this system state vector the graph of model M_i is disconnected, then the graph of model $\widetilde{M}_i$, also becomes disconnected, as a result of which the graph of the combined model M likewise loses connectivity.

Thus, under condition C_i, the combined GL-model M reproduces the behavior of the *auxiliary* model M_i for all $i = 1,2, \ldots, K$. Therefore, model M correctly describes the failure behavior of the considered fault-tolerant multiprocessor system.

ALGORITHM FOR CONSTRUCTING A GL-MODEL OF A NON-BASIC FTMS

Based on the method proposed above, the following algorithm can be formulated for constructing a GL-model of a non-basic fault-tolerant multiprocessor system.

1. Construct Boolean expressions $c_1, c_2, \ldots, c_K$ corresponding to the conditions $C_1, C_2, \ldots, C_K$.
2. Construct *auxiliary* GL-models $M_1, M_2, \ldots, M_K$ that describe the failure behavior of the *conditional* fault-tolerant multiprocessor systems corresponding to conditions $C_1, C_2, \ldots, C_K$, respectively.
3. Modify the edge functions of the GL-models $M_1, M_2, \ldots, M_K$ according to (1), obtaining the *modified auxiliary* GL-models $\widetilde{M}_1, \widetilde{M}_2, \ldots, \widetilde{M}_K$.
4. Combine the *modified auxiliary* GL-models $\widetilde{M}_1, \widetilde{M}_2, \ldots, \widetilde{M}_K$ according to the method described in [45].
5. The GL-model M obtained as a result of the previous steps describes the failure behavior of the original fault-tolerant multiprocessor system.

EXAMPLES AND EXPERIMENTAL RESULTS

Example 1. As an example, let us construct a GL-model of a fault-tolerant multiprocessor system consisting of eight processors that is tolerant to any failures of multiplicity not exceeding two, as well as to a set of failures of multiplicity three, which is described by the following set of system state vectors: B = {00111011, 00111101, 01010111, 01011101, 01011110, 01100111, 01101110, 01111010, 01111100, 10011011, 10011101, 10101011, 10110110, 10111010, 10111100, 11000111, 11001011, 11001101, 11001110, 11011001, 11011100, 11100110}.

First, using the approach described in [19], we construct the basic GL-model $K_1(2, 8)$, which is based on a cycle graph with seven vertices (α_1, α_2, α_3, α_4, α_5, α_6, α_7) and seven edges (Fig. 1), and has the following edge functions:

$f_1^1 = x_1 \vee x_2;$
$f_2^1 = x_1x_2 \vee x_3x_4;$
$f_3^1 = x_3 \vee x_4;$
$f_4^1 = x_1x_2x_3x_4 \vee x_5x_6x_7x_8;$
$f_5^1 = x_5 \vee x_6;$
$f_6^1 = x_5x_6 \vee x_7x_8;$
$f_7^1 = x_7 \vee x_8.$

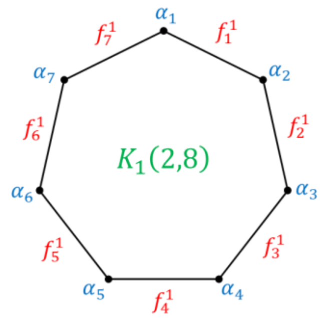

Fig. 1. **GL-model $K_1(2, 8)$ from *Example 1***
***Source:* compiled by the authors**

Next, we modify this model by introducing two internal edges, $\alpha_1\alpha_5$ and $\alpha_2\alpha_4$, associated with the edge functions f_8^1 and f_9^1, respectively, where

$f_8^1 = x_1x_3x_6x_7;$
$f_9^1 = x_2x_6x_7.$

The model obtained as a result of this modification (Fig. 2) will hereafter be denoted as M_1.

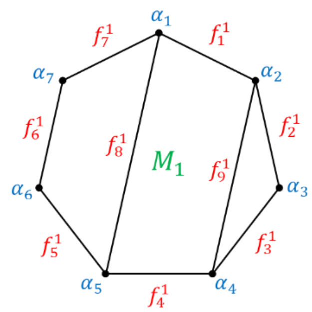

Fig. 2. **GL-model M_1 from *Example 1***
***Source:* compiled by the authors**

As a result of this modification, the behavior of the obtained model differs from that of the basic model $K_1(2, 8)$ on the following eight system state vectors with three zero components: B_1 = {01010111, 01011110, 01100111, 01101110, 10110110, 11000111, 11001110, 11100110}. For this set of vectors, model M_1 corresponds to an operational system state, whereas the basic model K_1 interprets the same vectors as a non-operational state.

Let us also construct the basic GL-model $K_2(2, 8)$ according to the method described in [18]. This model is based on a cycle graph with eight vertices (β_1, β_2, β_3, β_4, β_5, β_6, β_7, β_8) and eight edges

(Fig. 3), which are associated with the following edge functions:

$f_1^2 = x_1 \vee x_2x_3x_4;$
$f_2^2 = x_2 \vee x_3x_4x_5;$
$f_3^2 = x_3 \vee x_4x_5x_6;$
$f_4^2 = x_4 \vee x_5x_6x_7;$
$f_5^2 = x_5 \vee x_6x_7x_8;$
$f_6^2 = x_6 \vee x_7x_8x_1;$
$f_7^2 = x_7 \vee x_8x_1x_2;$
$f_8^2 = x_8 \vee x_1x_2x_3.$

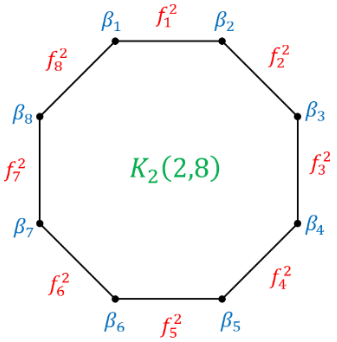

Fig. 3. **GL-model K_2 (2, 8) from *Example 1***
***Source:* compiled by the authors**

Next, we modify this model by introducing two internal edges, $\beta_3\beta_8$ and $\beta_4\beta_8$, associated with the edge functions f_9^2 and f_{10}^2, respectively, where

$f_9^2 = x_4x_5;$
$f_{10}^2 = x_1x_5x_8.$

The GL-model obtained as a result of this modification (Fig. 4) will hereafter be denoted as M_2.

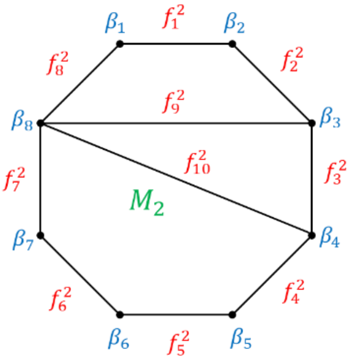

Fig. 4. **GL-model M_2 from *Example 1***
***Source:* compiled by the authors**

This modification results in the behavior of model M_2 differing from that of the basic model K_2(2, 8) on the following set of fourteen system state vectors with three zero components: B_2 = {00111011, 00111101, 01011101, 01111010, 01111100, 10011011, 10011101, 10101011, 10111010, 10111100, 11001011, 11001101, 11011001, 11011100}. As in the previous case, for this set of vectors, model M_2 corresponds to an operational system state, whereas the basic model K_2 interprets the same vectors as a non-operational state.

Note that $B = B_1 \cup B_2$. In addition, all vectors from set B_1 have unit values in the sixth and seventh components, whereas for all vectors from set B_2, at least one of the sixth or seventh components is equal to zero. Therefore, under the condition of simultaneous operability of the sixth and seventh processors, which corresponds to the Boolean expression $c_1 = x_6x_7$, the failure behavior of the system can be described using the GL-model M_1. In all other cases, which correspond to the Boolean expression $c_2 = \bar{c}_1 = \overline{x_6x_7} = \bar{x}_6 \vee \bar{x}_7$, the system behavior is described using the GL-model M_2.

Now let us apply the method proposed in this paper to construct a GL-model of the considered fault-tolerant multiprocessor system. To this end, we construct the *modified auxiliary* models $\widetilde{M}_1$ and $\widetilde{M}_2$. The edge functions of the *modified auxiliary* model $\widetilde{M}_1$ are as follows:

$\tilde{f}_1^1 = f_1^1 \vee \bar{c}_1 = x_1 \vee x_2 \vee \bar{x}_6 \vee \bar{x}_7;$
$\tilde{f}_2^1 = f_2^1 \vee \bar{c}_1 = x_1x_2 \vee x_3x_4 \vee \bar{x}_6 \vee \bar{x}_7;$
$\tilde{f}_3^1 = f_3^1 \vee \bar{c}_1 = x_3 \vee x_4 \vee \bar{x}_6 \vee \bar{x}_7;$
$\tilde{f}_4^1 = f_4^1 \vee \bar{c}_1 = x_1x_2x_3x_4 \vee x_5x_6x_7x_8 \vee \bar{x}_6 \vee$
$\vee \bar{x}_7 = x_1x_2x_3x_4 \vee x_5x_8 \vee \bar{x}_6 \vee \bar{x}_7;$
$\tilde{f}_5^1 = f_5^1 \vee \bar{c}_1 = x_5 \vee x_6 \vee \bar{x}_6 \vee \bar{x}_7 = 1;$
$\tilde{f}_6^1 = f_6^1 \vee \bar{c}_1 = x_5x_6 \vee x_7x_8 \vee \bar{x}_6 \vee \bar{x}_7 =$
$= x_5 \vee x_8 \vee \bar{x}_6 \vee \bar{x}_7;$
$\tilde{f}_7^1 = f_7^1 \vee \bar{c}_1 = x_7 \vee x_8 \vee \bar{x}_6 \vee \bar{x}_7 = 1;$
$\tilde{f}_8^1 = f_8^1 \vee \bar{c}_1 = x_1x_3x_6x_7 \vee \bar{x}_6 \vee \bar{x}_7 =$
$= x_1x_3 \vee \bar{x}_6 \vee \bar{x}_7;$
$\tilde{f}_9^1 = f_9^1 \vee \bar{c}_1 = x_2x_6x_7 \vee \bar{x}_6 \vee \bar{x}_7 =$
$= x_2 \vee \bar{x}_6 \vee \bar{x}_7.$

Similarly, the edge functions of the *modified auxiliary* model $\widetilde{M}_2$ are as follows:

$\tilde{f}_1^2 = f_1^2 \vee \bar{c}_2 = x_1 \vee x_2x_3x_4 \vee x_6x_7;$
$\tilde{f}_2^2 = f_2^2 \vee \bar{c}_2 = x_2 \vee x_3x_4x_5 \vee x_6x_7;$
$\tilde{f}_3^2 = f_3^2 \vee \bar{c}_2 = x_3 \vee x_4x_5x_6 \vee x_6x_7 =$
$= x_3 \vee x_6(x_4x_5 \vee x_7);$
$\tilde{f}_4^2 = f_4^2 \vee \bar{c}_2 = x_4 \vee x_5x_6x_7 \vee x_6x_7 =$
$= x_4 \vee x_6x_7;$
$\tilde{f}_5^2 = f_5^2 \vee \bar{c}_2 = x_5 \vee x_6x_7x_8 \vee x_6x_7 =$
$= x_5 \vee x_6x_7;$
$\tilde{f}_6^2 = f_6^2 \vee \bar{c}_2 = x_6 \vee x_7x_8x_1 \vee x_6x_7 =$
$= x_6 \vee x_7x_8x_1;$
$\tilde{f}_7^2 = f_7^2 \vee \bar{c}_2 = x_7 \vee x_8x_1x_2 \vee x_6x_7 =$
$= x_7 \vee x_8x_1x_2;$
$\tilde{f}_8^2 = f_8^2 \vee \bar{c}_2 = x_8 \vee x_1x_2x_3 \vee x_6x_7;$
$\tilde{f}_9^2 = f_9^2 \vee \bar{c}_2 = x_4x_5 \vee x_6x_7;$

$\tilde{f}_{10}^2 = f_{10}^2 \vee \bar{c}_2 = x_1 x_5 x_8 \vee x_6 x_7$.

Next, we combine the *modified auxiliary* models $\widetilde{M}_1$ and $\widetilde{M}_2$ using the method described in [45]. In particular, as an illustrative example, we merge vertices α_3 and β_8 of the graphs of these models; note that the choice of vertices does not affect the correctness of the method. The resulting model *M* is shown in Fig. 5.

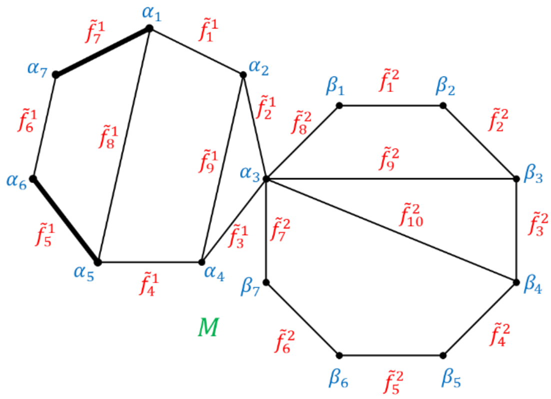

Fig. 5. **GL-model *M* from *Example 1***
Source: **compiled by the authors**

Experimental studies of the constructed GL-model show that it corresponds to an operational system state for all state vectors with no more than two zero components, as well as for all vectors from set *B*. For all other vectors, the graph of model *M* loses connectivity, which is interpreted as a non-operational system state. Thus, model *M* correctly represents the failure behavior of the considered fault-tolerant multiprocessor system.

Note also that the edge functions $\tilde{f}_5^1$ and $\tilde{f}_7^1$ of model $\widetilde{M}_1$ are identically equal to one. Consequently, the corresponding edges are permanently present in the graph of this model. This makes it possible to further simplify model $\widetilde{M}_1$ by merging the vertices connected by these edges, namely, the pairs of vertices α_5 and α_6, as well as α_7 and α_1. As a result of this simplification, the model $\widetilde{M}_1'$, shown in Fig. 6, is obtained.

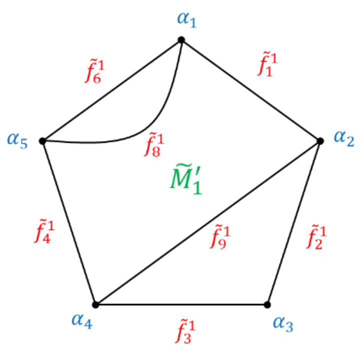

Fig. 6. **GL-model $\widetilde{M}_1'$ from *Example 1***
Source: **compiled by the authors**

In addition, model $\widetilde{M}_1'$ contains two parallel edges between vertices α_1 and α_5, which are associated with the edge functions $\tilde{f}_6^1$ and $\tilde{f}_8^1$. It can be shown that such a pair of edges is equivalent to a single edge with the edge function

$\tilde{f}_{10}^1 = \tilde{f}_6^1 \vee \tilde{f}_8^1 = x_5 \vee x_8 \vee \bar{x}_6 \vee \bar{x}_7 \vee x_1 x_3 \vee$
$\vee \bar{x}_6 \vee \bar{x}_7 = x_1 x_3 \vee x_5 \vee x_8 \vee \bar{x}_6 \vee \bar{x}_7$

The model $\widetilde{M}_1''$ obtained as a result of this transformation is shown in Fig. 7.

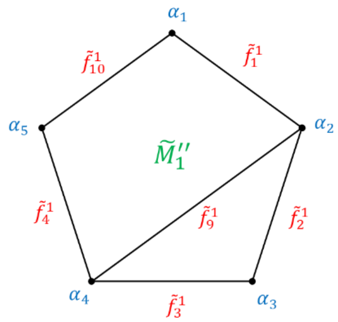

Fig. 7. **GL-model $\widetilde{M}_1''$ from *Example 1***
Source: **compiled by the authors**

Next, we combine models $\widetilde{M}_1''$ and $\widetilde{M}_2$ by selecting vertices α_2 and β_7 for merging. The resulting GL-model M' (Fig. 8) correctly represents the failure behavior of the considered fault-tolerant multiprocessor system, as confirmed by experimental studies.

Example 2. Let us also consider the system presented as an example in [44]. It consists of nine processors and is characterized by the following fault-tolerance properties. The system is tolerant to the failure of any three processors provided that processors 1 and 2 are simultaneously operational, or that processor 5 is operational. In addition, it is tolerant to the failure of any four processors provided that processors 4, 5, and 7 are simultaneously operational. In the general case, the system is 2-fault-tolerant.

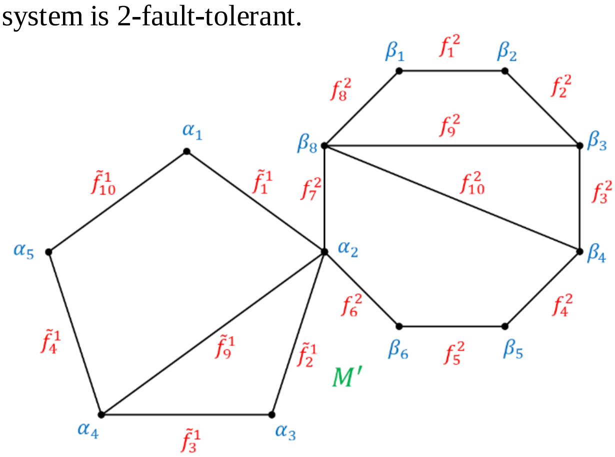

Fig. 8. **GL-model M' from *Example 1***
Source: **compiled by the authors**

For the considered fault-tolerant multiprocessor system, three *auxiliary* models $K_1(3, 9)$, $K_2(4, 9)$, and $K_3(2, 9)$ were constructed in [44]. These models are based on cycle graphs with seven, six, and eight edges, respectively. For brevity, the expressions of the edge functions of these models are not presented in this paper. Hereafter, these models will be used as *auxiliary* GL-models M_1, M_2, and M_3, respectively..

Also in [44], a procedure for *orthogonalizing* the Boolean expressions of the conditions was performed, which ensures that for any system state vector exactly one of the corresponding expressions evaluates to one. These expressions can be directly used in the present work, yielding:

$$c_1 = x_1 x_2 \bar{x}_5 \vee x_5(\bar{x}_4 \vee \bar{x}_7);$$
$$c_2 = x_4 x_5 x_7;$$
$$c_3 = \bar{x}_5(\bar{x}_1 \vee \bar{x}_2).$$

Note that in [44] these orthogonalized expressions were denoted by s_1, s_2, and s_3, whereas the symbols c_1, c_2, and c_3 were used to denote the corresponding conditional Boolean expressions prior to the application of the orthogonalization procedure.

We also derive the complementary expressions $\bar{c}_1$, $\bar{c}_2$, and $\bar{c}_3$, which will be used in forming the edge-function expressions of the *modified auxiliary* GL-models:

$$\bar{c}_1 = \overline{x_1 x_2 \bar{x}_5 \vee x_5(\bar{x}_4 \vee \bar{x}_7)} = \overline{x_1 x_2 \bar{x}_5} \wedge$$
$$\wedge\, \overline{x_5(\bar{x}_4 \vee \bar{x}_7)} = (\bar{x}_1 \vee \bar{x}_2 \vee x_5)(\bar{x}_5 \vee x_4 x_7);$$
$$\bar{c}_2 = \overline{x_4 x_5 x_7} = \bar{x}_4 \vee \bar{x}_5 \vee \bar{x}_7;$$
$$\bar{c}_3 = \overline{\bar{x}_5(\bar{x}_1 \vee \bar{x}_2)} = x_5 \vee x_1 x_2.$$

Next, based on the edge-function expressions f_j^i of models K_1, K_2, and K_3 presented in [44], and using transformation (1), we derive the edge-function expressions of the *modified auxiliary* GL-models.

As a result of the described transformations, the *modified auxiliary* GL-model $\widetilde{M}_1$ is defined by a graph with seven vertices (α_1, α_2, α_3, α_4, α_5, α_6, α_7), seven edges (Fig. 9), and the following edge functions:

$$\tilde{f}_1^1 = f_1^1 \vee \bar{c}_1 = x_1 \vee x_2 \vee x_3 \vee$$
$$\vee\, (\bar{x}_1 \vee \bar{x}_2 \vee x_5)(\bar{x}_5 \vee x_4 x_7);$$
$$\tilde{f}_2^1 = f_2^1 \vee \bar{c}_1 = (x_1 \vee x_2)(x_1 x_2 \vee x_3) \vee x_4 x_5 \vee$$
$$\vee\, (\bar{x}_1 \vee \bar{x}_2 \vee x_5)(\bar{x}_5 \vee x_4 x_7);$$
$$\tilde{f}_3^1 = f_3^1 \vee \bar{c}_1 = x_1 x_2 x_3 \vee x_4 \vee x_5 \vee$$
$$\vee\, (\bar{x}_1 \vee \bar{x}_2 \vee x_5)(\bar{x}_5 \vee x_4 x_7);$$
$$\tilde{f}_4^1 = f_4^1 \vee \bar{c}_1 = (x_1 \vee x_2)(x_1 x_2 \vee x_3) \wedge$$
$$\wedge\, (x_1 x_2 x_3 \vee x_4 x_5)(x_4 \vee x_5) \vee x_6 x_7 x_8 x_9 \vee$$
$$\vee\, (\bar{x}_1 \vee \bar{x}_2 \vee x_5)(\bar{x}_5 \vee x_4 x_7);$$
$$\tilde{f}_5^1 = f_5^1 \vee \bar{c}_1 = x_1 x_2 x_3 x_4 x_5 \vee (x_6 \vee x_7) \wedge$$
$$\wedge\, (x_6 x_7 \vee x_8 x_9)(x_8 \vee x_9) \vee (\bar{x}_1 \vee \bar{x}_2 \vee x_5) \wedge$$
$$\wedge\, (\bar{x}_5 \vee x_4 x_7);$$
$$\tilde{f}_6^1 = f_6^1 \vee \bar{c}_1 = x_6 \vee x_7 \vee x_8 x_9 \vee$$
$$\vee\, (\bar{x}_1 \vee \bar{x}_2 \vee x_5)(\bar{x}_5 \vee x_4 x_7);$$
$$\tilde{f}_7^1 = f_7^1 \vee \bar{c}_1 = x_6 x_7 \vee x_8 \vee x_9 \vee$$
$$\vee\, (\bar{x}_1 \vee \bar{x}_2 \vee x_5)(\bar{x}_5 \vee x_4 x_7).$$

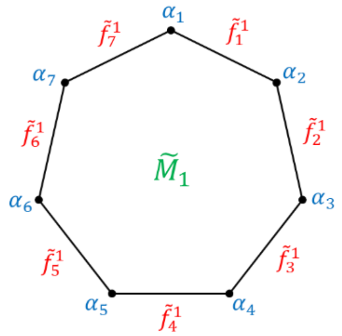

Fig. 9. **GL-model $\widetilde{M}_1$ from *Example 2***

***Source:* compiled by the authors**

The *modified auxiliary* model $\widetilde{M}_2$ is based on a graph with six vertices (β_1, β_2, β_3, β_4, β_5, β_6) and six edges (Fig. 10), and has the following edge functions:

$$\tilde{f}_1^2 = f_1^2 \vee \bar{c}_2 = x_1 \vee x_2 \vee x_3 \vee x_4 x_5 \vee$$
$$\vee\, \bar{x}_4 \vee \bar{x}_5 \vee \bar{x}_7 = 1;$$
$$\tilde{f}_2^2 = f_2^2 \vee \bar{c}_2 = (x_1 \vee x_2)(x_1 x_2 \vee x_3) \vee x_4 \vee x_5 \vee$$
$$\vee\, \bar{x}_4 \vee \bar{x}_5 \vee \bar{x}_7 = 1;$$
$$\tilde{f}_3^2 = f_3^2 \vee \bar{c}_2 = (x_1 \vee x_2 \vee x_3) \wedge$$
$$\wedge \big((x_1 \vee x_2)(x_1 x_2 \vee x_3) \vee x_4 x_5\big) \wedge$$
$$\wedge\, (x_1 x_2 x_3 \vee x_4 \vee x_5) \vee x_6 x_7 x_8 x_9 \vee \bar{x}_4 \vee \bar{x}_5 \vee \bar{x}_7 =$$
$$= (x_1 \vee x_2 \vee x_3)\big((x_1 \vee x_2)(x_1 x_2 \vee x_3) \vee x_4 x_5\big) \wedge$$
$$\wedge\, (x_1 x_2 x_3 \vee x_4 \vee x_5) \vee x_6 x_8 x_9 \vee \bar{x}_4 \vee \bar{x}_5 \vee \bar{x}_7;$$
$$\tilde{f}_4^2 = f_4^2 \vee \bar{c}_2 = (x_1 \vee x_2)(x_1 x_2 \vee x_3) \wedge$$
$$\wedge\, (x_1 x_2 x_3 \vee x_4 x_5)(x_4 \vee x_5) \vee (x_6 \vee x_7) \wedge$$
$$\wedge\, (x_6 x_7 \vee x_8 x_9)(x_8 \vee x_9) \vee \bar{x}_4 \vee \bar{x}_5 \vee \bar{x}_7;$$
$$\tilde{f}_5^2 = f_5^2 \vee \bar{c}_2 = x_1 x_2 x_3 x_4 x_5 \vee$$
$$\vee\, (x_6 \vee x_7 \vee x_8 x_9)(x_6 x_7 \vee x_8 \vee x_9) \vee$$
$$\vee\, \bar{x}_4 \vee \bar{x}_5 \vee \bar{x}_7 = x_1 x_2 x_3 \vee (x_6 \vee x_7 \vee x_8 x_9) \wedge$$
$$\wedge\, (x_6 x_7 \vee x_8 \vee x_9) \vee \bar{x}_4 \vee \bar{x}_5 \vee \bar{x}_7;$$
$$\tilde{f}_6^2 = f_6^2 \vee \bar{c}_2 = x_6 \vee x_7 \vee x_8 \vee x_9 \vee \bar{x}_4 \vee \bar{x}_5 \vee \bar{x}_7 =$$
$$= 1.$$

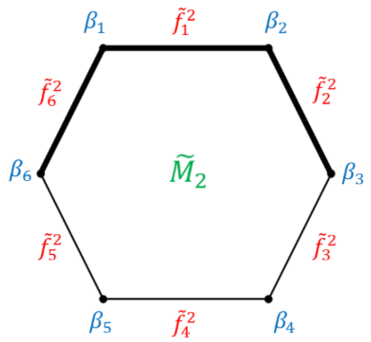

Fig. 10. **GL-model $\widetilde{M}_2$ from *Example 2***

***Source:* compiled by the authors**

Similarly, the *modified auxiliary* GL-model $\widetilde{M}_3$ is based on a graph with eight vertices (γ_1, γ_2, γ_3, γ_4,

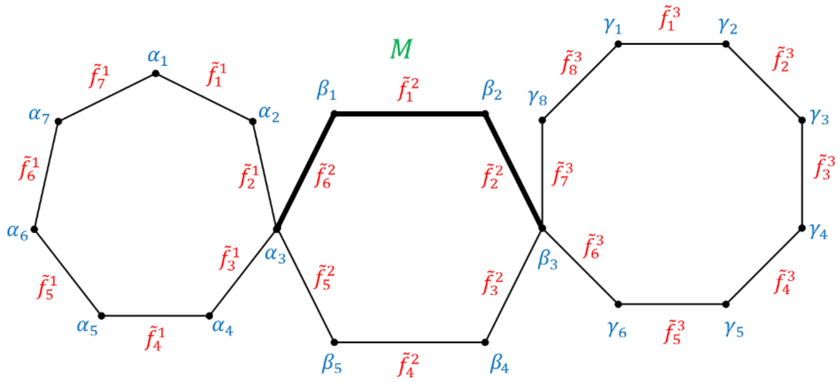

Fig. 12. **GL-model *M* from *Example 2***
***Source:* compiled by the authors**

γ_5, γ_6, γ_7, γ_8) and eight edges (Fig. 11), and has the following edge functions:

$\tilde{f}_1^3 = f_1^3 \vee \bar{c}_3 = x_1 \vee x_2 \vee x_5 \vee x_1x_2 =$
$= x_1 \vee x_2 \vee x_5;$
$\tilde{f}_2^3 = f_2^3 \vee \bar{c}_3 = x_1x_2 \vee x_3 \vee x_5 \vee x_1x_2 =$
$= x_1x_2 \vee x_3 \vee x_5;$
$\tilde{f}_3^3 = f_3^3 \vee \bar{c}_3 = x_1x_2x_3 \vee x_4x_5 \vee x_5 \vee x_1x_2 =$
$= x_1x_2 \vee x_5;$
$\tilde{f}_4^3 = f_4^3 \vee \bar{c}_3 = x_4 \vee x_5 \vee x_5 \vee x_1x_2 =$
$= x_4 \vee x_5 \vee x_1x_2;$
$\tilde{f}_5^3 = f_5^3 \vee \bar{c}_3 = x_1x_2x_3x_4x_5 \vee x_6x_7x_8x_9 \vee x_5 \vee$
$\vee x_1x_2 = x_1x_2 \vee x_5 \vee x_6x_7x_8x_9;$
$\tilde{f}_6^3 = f_6^3 \vee \bar{c}_3 = x_6 \vee x_7 \vee x_5 \vee x_1x_2;$
$\tilde{f}_7^3 = f_7^3 \vee \bar{c}_3 = x_6x_7 \vee x_8x_9 \vee x_5 \vee x_1x_2;$
$\tilde{f}_8^3 = f_8^3 \vee \bar{c}_3 = x_8 \vee x_9 \vee x_5 \vee x_1x_2.$

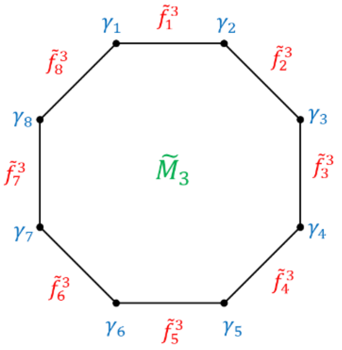

Fig. 11. **GL-model $\widetilde{M}_3$ from *Example 2***
***Source:* compiled by the authors**

Let us combine the *modified auxiliary* GL-models $\widetilde{M}_1$, $\widetilde{M}_2$, and $\widetilde{M}_3$ according to the method described in [45] by merging the arbitrarily selected pairs of vertices α_3 and β_6, as well as β_3 and γ_7. The GL-model obtained as a result of this combination is shown in Fig. 12. Experimental studies show that the behavior of this model coincides with that of the model constructed in [44]; therefore, it correctly represents the failure behavior of the considered fault-tolerant multiprocessor system.

As in the previous case, the *modified auxiliary* GL-model $\widetilde{M}_2$ contains a number of edges that are permanently present in the graph, since the corresponding edge functions are identically equal to one. Accordingly, such edges can be removed from the GL-model, and the vertices incident to them can be merged. As a result of this transformation, four vertices – β_1, β_2, β_3, and β_6 – are merged into a single vertex. Thus, the model $\widetilde{M}_2'$ is obtained, which is based on a cycle graph with three edges (Fig. 13).

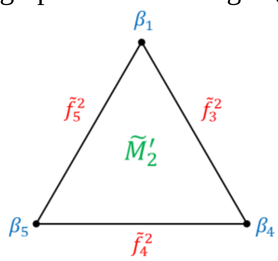

Fig. 13. **GL-model $\widetilde{M}_2'$ from *Example 2***
***Source:* compiled by the authors**

By combining models $\widetilde{M}_1$, $\widetilde{M}_2'$, and $\widetilde{M}_3$ using the method described in [45] by merging the arbitrarily selected pairs of vertices α_2 and β_5, as well as β_4 and γ_8, we obtain the GL-model M' (Fig. 14), which, similarly to the previous case, correctly represents the failure behavior of the considered fault-tolerant multiprocessor system, as confirmed by experimental studies.

DISCUSSION OF RESULTS

GL-models constructed using the method proposed in this paper correctly represent the failure behavior of fault-tolerant multiprocessor systems. This is confirmed by the results of experimental verification performed both for the models presented in the paper and for a number of additional models constructed using the proposed method. In particular, the models were tested on various system state vectors; for systems of small dimensionality (with the number of processors not exceeding 20), all possible state vectors were examined. The

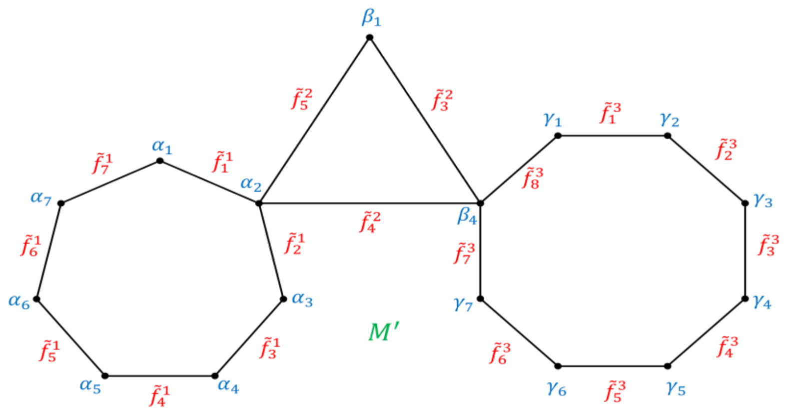

Fig. 14. **GL-model M' from *Example 2***
***Source:* compiled by the authors**

obtained results were compared with the expected states of the corresponding systems, which made it possible to confirm the correctness of the constructed models.

It should be noted that the method proposed in this paper leads to the construction of relatively complex GL-models which, as a rule, are based on graphs that do not belong to the class of cycle graphs. This somewhat complicates the procedure for determining graph connectivity, since in this case it is not possible to restrict the analysis to a simple counting of the number of edges removed from the graph, as is typical for basic models. However, as shown in [45], the procedure for assessing the connectivity of a GL-model graph usually does not result in a significant increase in the overall computational complexity.

Moreover, it is assumed that the original GL-models used as *auxiliary* models for constructing the GL-model of the entire fault-tolerant multiprocessor system may themselves be based on graphs other than cycle graphs, which, in particular, removes the constraints imposed by the method described in [44]. Therefore, the fact that the graphs of the resulting models do not belong to the class of cycle graphs does not, in many cases, lead to additional difficulties associated with the procedure for assessing GL-model graph connectivity.

A comparison of the complexity of the edge-function expressions of GL-models constructed using the proposed method with those obtained by existing approaches shows that the models built by the proposed method are characterized by significantly simpler edge-function expressions. In this comparison, existing methods for constructing non-basic GL-models were represented by approaches based on adding the corresponding unit and zero constituents to the edge-function expressions of a given basic model.

The results of the comparison of the complexity of the edge-function expressions of GL-models M and M' for *Example 1* are presented in Table 1. The corresponding results for models M and M' from *Example 2* are given in Table 2. In addition, Table 2 includes a comparison with the GL-model constructed using the method described in [44].

Table 1. **Number of logical operations in the edge-function expressions of the GL-models for *Example 1***

Model	Disj.	Conj.	Inv.	Binary ops.	Total ops.
M	33	29	14	62	76
M'	32	29	12	61	73
Modified $K(2, 8)$	29	164	66	193	259
Modified $K(3, 8)$	254	52	170	306	476

***Source:* compiled by the authors**

Since the system in *Example 1* is tolerant to certain subsets of failures of multiplicity 2 and 3, the basic GL-models $K(2, 8)$ and $K(3, 8)$ were selected as initial models for further modification. Similarly, the system in *Example 2* is characterized by tolerance to certain subsets of failures of multiplicity 2, 3, and 4; therefore, the basic models $K(2, 9)$, $K(3, 9)$, and $K(4, 9)$ were used.

The obtained results indicate that the method proposed in this paper indeed enables the construction of simpler models of non-basic fault-tolerant multiprocessor systems, at least in terms of the complexity of the edge-function expressions.

Table 2. **Number of logical operations in the edge-function expressions of the GL-models for *Example 2***

Model	Disj.	Conj.	Inv.	Binary ops.	Total ops.
M	96	76	30	172	202
M'	96	76	30	172	202
Proposed in [44]	99	138	48	237	285
Modified $K(2, 9)$	109	821	333	930	1263
Modified $K(3, 9)$	139	158	138	297	435
Modified $K(4, 9)$	1127	168	711	1295	2006

Source: **compiled by the authors**

It is also worth noting that the evaluation of the edge-function expressions of GL-models can be performed in parallel. In this case, the determining factor is not the total number of logical operations across all edge-function expressions of the model, but the maximum number of logical operations among individual edge functions. Table 3 presents the maximum number of logical operations among the edge-function expressions of models M and M' from *Example 2*, as well as of the corresponding GL-model constructed using the method described in [44]. As can be seen from the presented results, although models M and M' contain a larger number of edge functions compared to the model obtained by the method of [44], their maximum complexity is lower, which, under parallel evaluation of edge-function values, makes it possible to reduce the overall computation time.

Table 3. **Maximum number of logical operations in the edge-function expressions of the GL-models from *Example 2***

Model	No. of edge functions	Binary ops.	Total ops.
M	21	22	25
M'	18	22	25
Proposed in [44]	8	49	55

Source: **compiled by the authors**

It should be noted that the actual computation time depends on the specific implementation and hardware platform. However, in the general case, the computation time is largely determined by the number of logical operations required to evaluate the edge-function expressions, which allows the complexity measures reported in the tables to be used as an implementation-independent indicator of computational efficiency.

CONCLUSIONS

The paper proposes a method for constructing GL-models of the failure behavior of a special class of non-basic fault-tolerant multiprocessor systems. The considered systems are those whose behavior, under the fulfillment of certain conditions specified by the corresponding Boolean expressions, can be described using simpler auxiliary GL-models constructed by known methods. At the same time, no constraints are imposed on these auxiliary models, in particular with respect to the structure of their graphs.

The application of existing methods for constructing GL-models for systems of this type often results in excessively complex models, which, in particular, are characterized by a high complexity of edge-function expressions. Other approaches impose constraints on auxiliary models by requiring them to be constructed exclusively on cycle graphs, which significantly complicates the application of such methods to systems with complex failure behavior. In contrast, GL-models constructed using the proposed methods are characterized by simpler edge-function expressions. In particular, their complexity is at least several times lower compared to models obtained by blocking the corresponding system state vectors through the addition of zero and unit constituents to the edge-function expressions of basic models.

The presented examples demonstrate the applicability of the proposed method for constructing GL-models to systems of various types. Experimental studies confirm the adequacy of the constructed models in representing the failure behavior of the corresponding systems.

It is also shown that the GL-models obtained using the proposed method, in certain cases, admit further simplification, in particular by eliminating edges that are permanently present in the graph due to the corresponding edge functions being identically equal to one.

It is worth noting that the examples considered in this paper were intentionally chosen to be of moderate size in order to allow clear illustration and experimental verification of the proposed method.

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

While for such systems a direct analysis of system state vectors may still be applicable in principle, it already lacks structural flexibility. In real-world fault-tolerant multiprocessor systems, the number of components and the complexity of operability conditions are typically much higher, making direct state-space analysis impractical due to combinatorial explosion. In such cases, the advantages of the proposed graph-based approach become particularly significant.

REFERENCES

1. Nazarova, O. S., Osadchyy, V. V. & Rudim, B. Y. “Computer simulation of the microprocessor liquid level automatic control system”. *Applied Aspects of Information Technology*. 2023; 6 (2): 163–174. DOI: https://doi.org/10.15276/aait.06.2023.12.

2. Kotov, D. O. “A generalized model of an adaptive information-control system of a car with multi-sensor channels of information interaction”. *Applied Aspects of Information Technology.* 2022; 5 (1): 25–34. DOI: https://doi.org/10.15276/aait.05.2022.2.

3. Drozd, O., Ivanova, O., Zashcholkin, K., Romankevich, V. & Drozd, Yu. “Checkability important for fail-safety of FPGA-based components in critical systems”. *CEUR Workshop Proceedings*. 2021; 2853: 471–480, https://www.scopus.com/pages/publications/85104838273.

4. Kovalev, I. S., Drozd, O. V., Rucinski, A., Drozd, M. O., Antoniuk, V. V. & Sulima, Y. Y. “Development of computer system components in critical applications: problems, their origins and solutions”. *Herald of Advanced Information Technology.* 2020; 3 (4): 252–262. DOI: https://doi.org/10.15276/hait.04.2020.4.

5. Antoniuk, V. V., Drozd, M. O. & Drozd, O. V. “Power-oriented checkability and monitoring of the current consumption in FPGA projects of the critical applications”. *Applied Aspects of Information Technology*. 2019; 2 (2): 105–114. DOI: https://doi.org/10.15276/aait.02.2019.2.

6. Joshi, H. & Sinha, N. K. “Adaptive fault tolerant control design for stratospheric airship with actuator faults”. *IFAC-PapersOnLine*. 2022; 55 (1): 819–825, https://www.scopus.com/pages/publications/85132157930. DOI: https://doi.org/10.1016/j.ifacol.2022.04.134.

7. Abbaspour, A., Mokhtari, S., Sargolzaei, A. & Yen, K. K. “A survey on active fault-tolerant control systems”. *Electronics*. 2020; 9 (9): 1–23, https://www.scopus.com/pages/publications/85090902832. DOI: https://doi.org/10.3390/electronics9091513.

8. Nedeljkovic, J. N., Dosic, S. M. & Nikolic, G. S. “A survey of hardware fault tolerance techniques”. *2023 58th International Scientific Conference on Information, Communication and Energy Systems and Technologies, ICEST 2023 – Proceedings*. 2023. p. 223–226, https://www.scopus.com/pages/publications/85167870342. DOI: https://doi.org/10.1109/ICEST58410.2023.10187275.

9. Kuo, W. & Zuo, M. “Optimal reliability modeling”. *John Wiley & Sons*. 2002.

10. Billinton, R. & Allan, R. N. “Reliability evaluation of engineering systems. concepts and techniques”. *Springer New York*. 1992. DOI: https://doi.org/10.1007/978-1-4899-0685-4.

11. Hu, B. & Seiler, P. “Pivotal decomposition for reliability analysis of fault tolerant control systems on unmanned aerial vehicles”. *Reliability Engineering & System Safety*. 2015; 140: 130–141, https://www.scopus.com/pages/publications/84928741612. DOI: https://doi.org/10.1016/j.ress.2015.04.005.

12. Yi, H., Cui, L. & Gao, H. “Reliabilities of some multistate consecutive-k systems”. *IEEE Transactions on Reliability*. 2020; 69 (2): 414–429. DOI: https://doi.org/10.1109/TR.2019.2897726.

13. Jianu, M., Daus, L., Dragoi, V. F. & Beiu, V. “Reliability polynomials of consecutive-k-out-of-n:F systems have unbounded roots”. *Networks*. 2023; 82 (3): 222–228, https://www.scopus.com/pages/publications/85163222383. DOI: https://doi.org/10.1002/net.22168.

14. Dong, H., Xie, H. & Xu, H. “Reliability evaluation of motor controllers based on FMEA and DFTA”. *Hangkong Dongli Xuebao/Journal of Aerospace Power*. 2025; 40 (9): 20230656, https://www.scopus.com/pages/publications/105011205376. DOI: https://doi.org/10.13224/j.cnki.jasp.20230656.

15. Wu, J., Chen, C., Ma, Y., Xiu, Z., Cheng, Z., Pan, Y. & Song, S. “Integrated fault tree and case analysis for equipment conventional fault IETM diagnosis”. *Sensors*. 2025; 25 (17): 5231, https://www.scopus.com/pages/publications/105015822431. DOI: https://doi.org/10.3390/s25175231.

16. Dagal, I. “Probabilistic fault tree analysis and dynamic redundancy optimization for next-generation avionic flight control systems”. *Reliability Engineering and System Safety*. 2026; 266: 111841, https://www.scopus.com/pages/publications/105020942559. DOI: https://doi.org/10.1016/j.ress.2025.111841.

17. Romankevich, A., Feseniuk, A., Maidaniuk, I. & Romankevich, V. “Fault-tolerant multiprocessor systems reliability estimation using statistical experiments with GL-models”. *Advances in Intelligent Systems*

and Computing. 2019; 754: 186–193, https://www.scopus.com/pages/publications/85047465084. DOI: https://doi.org/10.1007/978-3-319-91008-6_19.

18. Romankevich, A. M., Romankevich, V. A., Kononova, A. A. & Rabah Al Shbul. "On some features of GL-models K(2, n)". *Visnyk NTUU "KPI". Informatyka, Upravlinnia ta Obchysliuvalna Tekhnika*. 2004; 41: 85–92.

19. Romankevich, V. A., Potapova, E. R., Bakhtari, Hedayatollah & Nazarenko, V. V. "GL-model of the behavior of fault-tolerant multiprocessor systems with the minimal number of lost edges". *Visnyk NTUU "KPI". Informatyka, Upravlinnia ta Obchysliuvalna Tekhnika*. 2006; 45: 93–100.

20. Yin, J., Balakrishnan, N. & Cui, L. "Efficient reliability computation of consecutive- k-out-of-n: F systems with shared components". *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*. 2024; 238 (1): 122–135, https://www.scopus.com/pages/publications/85142023916. DOI: https://doi.org/10.1177/1748006X221130540.

21. Eryilmaz, S. & Kan, C. "The mean number of failed components in discrete time consecutive k-out-of-n: F system and its application to parameter estimation and optimal age-based preventive replacement". *Reliability Engineering and System Safety*. 2025; 263: 111229, https://www.scopus.com/pages/publications/105005869605. DOI: https://doi.org/10.1016/j.ress.2025.111229.

22. Yin, J. & Cui, L. "Reliability for consecutive-k-out-of-n:F systems with shared components between adjacent subsystems". *Reliability Engineering & System Safety*. 2021; 210: 107532, https://www.scopus.com/pages/publications/85100736521. DOI: https://doi.org/10.1016/j.ress.2021.107532.

23. Chopra, G. & Ram, M. "Linear Consecutive-k-out-of-n:G system reliability analysis". *Journal of Reliability and Statistical Studies*. 2022; 15 (2): 669–692. DOI: https://doi.org/10.13052/jrss0974-8024.15211.

24. Romankevich, V. A., Yermolenko, I. A., Morozov K. V. & Romankevich A. M. "Graph-logical models for (n,f,k) – and consecutive-k-out-of-n–systems". *Herald of Advanced Information Technology*. 2024; 7 (3): 296–308. DOI: https://doi.org/10.15276/hait.07.2024.21.

25. Amirian, Y., Khodadadi, A. & Chatrabgoun, O. "Exact Reliability for a Consecutive Circular k-out-of-r-from-n:F System with Equal and Unequal Component Probabilities". *International Journal of Reliability, Quality and Safety Engineering*. 2020; 27 (1), https://www.scopus.com/pages/publications/85069848232. DOI: https://doi.org/10.1142/S0218539320500035.

26. Chachra, A., Ram, M. & Kumar, A. "A Pythagorean fuzzy approach to consecutive k-out-of-r-from-n system reliability modelling". *International Journal of System Assurance Engineering and Management*. 2024, https://www.scopus.com/pages/publications/85200035598. DOI: https://doi.org/10.1007/s13198-024-02435-3.

27. Nashwan, I. "Reliability and failure probability functions of the m-Consecutive-k-out-of-n: F linear and circular systems". *Baghdad Science Journal*. 2021; 18 (2): 430, https://www.scopus.com/pages/publications/85100161580. DOI: https://doi.org/10.21123/bsj.2021.18.2.0430.

28. Triantafyllou, I. "m-Consecutive-k-out-of-n: F structures with a single change point". *Mathematics*. 2020; 8 (12): 2203, https://www.scopus.com/pages/publications/85097534209. DOI: https://doi.org/10.3390/math8122203.

29. Triantafyllou, I. S. "Combined m-Consecutive-k-Out-of-n: F and Consecutive k_c-Out-of-n:F Structures with cold standby redundancy". *Mathematics*. 2023; 11 (12): 1–13, https://www.scopus.com/pages/publications/85164204439. DOI: https://doi.org/10.3390/math11122597.

30. Torrado, N. "Tail behaviour of consecutive 2-within-m-out-of-n systems with nonidentical components". *Applied Mathematical Modelling*. 2015; 39 (15): 4586–4592, https://www.scopus.com/pages/publications/84937631892. DOI: https://doi.org/10.1016/j.apm.2014.12.042.

31. Zhu, X., Boushaba, M. & Reghioua, M. "Reliability and joint reliability importance in a Consecutive-k-Within-m-out-of-n: F system with Markov-dependent components". *IEEE Transactions on Reliability*. 2016; 65 (2): 802–815, https://www.scopus.com/pages/publications/84943426081. DOI: https://doi.org/10.1109/TR.2015.2484079.

32. Romankevich, V. A., Yermolenko, I. A., Morozov, K. V., Romankevich, A. M. & Melnyk, O. O. "Generalization of the method for constructing GL-models of complex fault-tolerant multiprocessor systems with additional failure conditions". *Applied Aspects of Information Technology*. 2025; 8 (2), 216–230. https://doi.org/10.15276/aait.08.2025.15.

33. Lin, C., Cui L. R., Coit D. W. & Lv, M. "Reliability modeling on consecutive-k_r-out-of-n_r:F linear zigzag structure and circular polygon structure". *IEEE Transactions on Reliability*. 2016; 65 (3): 1509–1521, https://www.scopus.com/pages/publications/84973885922. DOI: https://doi.org/10.1109/TR.2016.2570545.

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

34. Yin, J., Cui, L. & Balakrishnan N. "Reliability of consecutive-(k,l)-out-of-n: F systems with shared components under non-homogeneous Markov dependence". *Reliability Engineering & System Safety*. 2022; 224: 108549, https://www.scopus.com/pages/publications/85129557917. DOI: https://doi.org/10.1016/j.ress.2022.108549.

35. Zhu, X., Boushaba, M., Coit, D. W. & Benyahia, A. "Reliability and importance measures for m-consecutive-k,l-out-of-n system with non-homogeneous Markov-dependent components". *Reliability Engineering and System Safety, Elsevier*. 2017; 167 (C): 1–9, https://www.scopus.com/pages/publications/85019236438. DOI: https://doi.org/10.1016/j.ress.2017.05.023.

36. Özbey, F. "Reliability evaluation of m-consecutive-k, l-out-of-n: F system subjected to shocks". *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*. 2022; 236 (6): 1135–1146, https://www.scopus.com/pages/publications/85116069008. DOI: https://doi.org/10.1177/1748006X211048992.

37. Lu, J., Yi, H., Li, X. & Balakrishnan, N. "Joint reliability of two consecutive-(1, l) or (2, k)-out-of-(2, n): F Type systems and its application in smart street light deployment". *Methodology and Computing in Applied Probability*. 2023; 25 (1): 33, https://www.scopus.com/pages/publications/85148634528. DOI: https://doi.org/10.1007/s11009-023-09984-3.

38. Nakamura, T., Homma, I. & Yamamoto, H. "Birnbaum importance-based simulated annealing algorithm for solving the component assignment problem of linear connected-(r, s)-out-of-(m, n):F lattice systems". *International Journal of System Assurance Engineering and Management*. 2024; 15 (4): 1407–1414, https://www.scopus.com/pages/publications/85144872424. DOI: https://doi.org/10.1007/s13198-022-01848-2.

39. Nakamura, T., Yamamoto, H. & Akiba, T. "Reliability of a toroidal connected-(r, s)-out-of-(m, n): F lattice system". *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*. 2022; 236 (2): 329–338, https://www.scopus.com/pages/publications/85077453315. DOI: https://doi.org/10.1177/1748006X19893533.

40. Amrutkar, K. P. & Kamalja, K. K. "Efficient algorithm for reliability and importance measures of linear weighted-(n,f,k) and (n,f,k) systems". *Computers and Industrial Engineering*. 2017; 107: 85–99, https://www.scopus.com/pages/publications/85015035867. DOI: https://doi.org/10.1016/j.cie.2017.02.011.

41. Cui, L., Wang, M. & Jiang, W. "Reliability analysis of a combination of (n, f, k) and <n, f, k> systems". *Reliability Engineering and System Safety*. 2024; 249: 110191, https://www.scopus.com/pages/publications/85193451324. DOI: https://doi.org/10.1016/j.ress.2024.110191.

42. Triantafyllou, I. S. "Reliability study of <n, f, 2> systems: A generating function approach". *International Journal of Mathematical, Engineering and Management Sciences*. 2021; 6 (1): 44–65, https://www.scopus.com/pages/publications/85097568436. DOI: https://doi.org/10.33889/IJMEMS.2021.6.1.005.

43. Romankevich, A. M., Morozov, K. V. & Romankevich, V. A. "Graph-logic models of hierarchical fault-tolerant multiprocessor systems". *IJCSNS International Journal of Computer Science and Network Security*. 2019; 19 (7): 151–156.

44. Romankevich, V. A., Morozov, K. V., Romankevich, A. M., Morozova, A. V. & Zacharioudakis, L. "On the method of building of non-basic GL-models which are formed on combination of edge functions of basic models". *Applied Aspects of Information Technology*. 2024; 7 (2): 175–188. DOI: https://doi.org/10.15276/aait.07.2024.13.

45. Romankevich, V. A., Morozov, K. V., Halytsky, D. V., Yermolenko, I. A. & Zacharioudakis, L. "A method for constructing GL-models of behavior under failure flow for complex non-basic fault-tolerant multiprocessor systems". *Applied Aspects of Information Technology*. 2025; 8 (4), 465–480. https://doi.org/10.15276/aait.08.2025.30.

Conflicts of Interest: The authors declare that they have no conflict of interest regarding this study, including financial, personal, authorship, or other interests, which could influence the research and its results presented in this article

Received 08.12.2025
Received after revision 12.02.2026
Accepted 18.02.2026

DOI: https://doi.org/10.15276/hait.09.2026.03
УДК 004.05

Метод побудови небазових GL-моделей шляхом комбінування моделей з довільною структурою графів

Романкевич Віталій Олексійович[1)]
ORCID: https://orcid.org/0000-0003-4696-5935; zavkaf@scs.kpi.ua. Scopus Author ID: 57193263058
Морозов Костянтин В'ячеславович[1)]
ORCID: https://orcid.org/0000-0003-0978-6292; mcng@ukr.net. Scopus Author ID: 57222509251
Романкевич Олексій Михайлович[1)]
ORCID: https://orcid.org/0000-0001-5634-8469; romankev@scs.kpi.ua. Scopus Author ID: 6602114176
Галицький Даніїл Володимирович[1)]
ORCID: https://orcid.org/0009-0004-4421-3443; zipper135401@gmail.com. Scopus Author ID: 58553487600
Лефтеріс Захаріудакіс[2)]
ORCID: https://orcid.org/0000-0002-9658-3073; l.zacharioudakis@nup.ac.cy. Scopus Author ID: 57422876200
[1)] Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37. Київ, 03056, Україна
[2)] Університет Неаполіс Пафос, Данаіс Авеню, 2, Пафос, 8042, Кіпр

АНОТАЦІЯ

У роботі запропоновано метод побудови GL-моделей поведінки в потоці відмов небазових відмовостійких багатопроцесорних систем спеціального типу. Розглядається клас систем, поведінка яких у потоці відмов визначається виконанням однієї з кількох умов, що залежать від комбінацій станів процесорів і задаються відповідними булевими виразами. За виконання кожної з таких умов поведінка системи в потоці відмов може бути описана за допомогою окремої допоміжної моделі, побудованої відомими методами. Метою роботи є створення узагальненого методу побудови моделей таких небазових систем шляхом комбінування кількох допоміжних моделей довільного виду в єдину модель, яка коректно відображає поведінку системи в потоці відмов. Запропонований метод ґрунтується на модифікації реберних функцій допоміжних моделей із використанням булевих виразів відповідних умов, після чого модифіковані моделі об'єднуються шляхом злиття довільно обраних вершин їх графів. Показано, що отримана в результаті модель відтворює поведінку відповідної допоміжної моделі за виконання кожної з умов і, таким чином, коректно описує поведінку вихідної системи в потоці відмов. Наукова новизна роботи полягає в запропонованому способі комбінування допоміжних моделей загального вигляду, який не накладає обмежень на структуру їх графів і, на відміну від відомих підходів, не вимагає, щоб ці моделі базувалися на графах-циклах. Це дозволяє будувати моделі для широкого класу небазових відмовостійких багатопроцесорних систем зі складними умовами роботоздатності. Проведено порівняльний аналіз складності виразів реберних функцій у моделях, побудованих запропонованим методом, і моделях, отриманих за допомогою існуючих підходів, на ряді характерних прикладів, що дозволило виявити істотне зменшення складності таких виразів. Окремо розглянуто випадок паралельного обчислення значень реберних функцій і показано зменшення максимальної складності їх виразів, що є важливим для скорочення часу розрахунків. Практична цінність роботи полягає в можливості застосування запропонованого методу для побудови моделей складних небазових відмовостійких багатопроцесорних систем, а також у зменшенні складності аналізу їх поведінки в потоці відмов. Наведені приклади та результати експериментальних досліджень підтверджують коректність побудованих моделей і демонструють можливість їх подальшого спрощення шляхом усунення ребер із тотожно одиничними реберними функціями. Запропонований підхід може бути використаний під час аналізу та проєктування систем керування критичного застосування, а також у задачах автоматизованої оцінки надійності небазових відмовостійких багатопроцесорних систем із великою кількістю компонентів.

Ключові слова: відмовостійкі багатопроцесорні системи; GL-моделі; небазові системи; системи керування; оцінка надійності; статистичні експерименти

ABOUT THE AUTHORS

Vitaliy A. Romankevich - Doctor of Engineering Sciences, Professor, Head of System Programming and Specialized Computer System Department. National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Peremogy Ave. Kyiv, 03056, Ukraine
ORCID: https://orcid.org/0000-0003-4696-5935; zavkaf@scs.kpi.ua. Scopus Author ID: 57193263058
Research field: Fault-tolerant multiprocessor systems reliability estimation; GL-models; Self-diagnosable systems; Diagnosis of multiprocessor systems; Discrete mathematics

Романкевич Віталій Олексійович - доктор технічних наук, професор, завідувач кафедри Системного програмування і спеціалізованих комп'ютерних систем. Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37. Київ, 03056, Україна

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

Kostiantyn V. Morozov - PhD, Assistant, System Programming and Specialized Computer System Department. National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Peremogy Ave. Kyiv, 03056, Ukraine
ORCID: https://orcid.org/0000-0003-0978-6292, mcng@ukr.net, Scopus Author ID: 57222509251
***Research field*:** GL-models; Fault-tolerant multiprocessor systems reliability estimation; Self-diagnosable multiprocessor systems

Морозов Костянтин В'ячеславович - кандидат технічних наук, асистент кафедри Системного програмування і спеціалізованих комп'ютерних систем. Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37. Київ, 03056, Україна

Alexei M. Romankevich - Doctor of Engineering Sciences, Professor, System Programming and Specialized Computer System Department. National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Peremogy Ave. Kyiv, 03056, Ukraine
ORCID: https://orcid.org/0000-0001-5634-8469; romankev@scs.kpi.ua. Scopus Author ID: 6602114176
***Research field*:** Fault-tolerant multiprocessor systems reliability estimation; GL-models; Self-diagnosable systems; Diagnosis of multiprocessor systems; Multi-valued logic.

Романкевич Олексій Михайлович - доктор технічних наук, професор, кафедра Системного програмування і спеціалізованих комп'ютерних систем. Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37. Київ, 03056, Україна

Daniil V. Halytsky - postgraduate student, System Programming and Specialized Computer System Department. National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Peremogy Ave. Kyiv, 03056, Ukraine
ORCID: https://orcid.org/0009-0004-4421-3443; zipper135401@gmail.com. Scopus Author ID: 58553487600
***Research field*:** GL-models building and processing optimization, Parallel computing and multi-core system optimization, Algorithm design for distributed and high-performance computing, Graph algorithms and their applications in complex systems, Computational efficiency in large-scale data processing.

Галицький Даніїл Володимирович - аспірант, кафедра Системного програмування і спеціалізованих комп'ютерних систем, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», пр. Перемоги, 37. Київ, 03056, Україна

Lefteris Zacharioudakis - PhD (Eng), Assistant professor. Neapolis University Pafos, 2, Danais Ave., Pafos, 8042, Cyprus
ORCID: https://orcid.org/0000-0002-9658-3073; l.zacharioudakis@nup.ac.cy. Scopus Author ID: 57422876200
***Research field*:** Principles of CyberSecurity, Operating systems, Information Security, Cryptography, Penetration Testing

Лефтеріс Захаріудакіс - кандидат технічних наук, доцент. Університет Неаполіс Пафос, Данаіс Авеню, 2, Пафос, 8042, Кіпр