I. J. Computer Network and Information Security, 2023, 1, 1-13
Published Online on February 8, 2023 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2023.01.01

Synthesis of the Structure of a Computer System Functioning in Residual Classes

Victor Krasnobayev
Department security information systems and technologies of the V. N. Karazin Kharkiv National University, 4 Svobody Sq., Kharkiv, 61022, Ukraine
E-mail: v.a.krasnobaev@gmail.com
ORCID iD: https://orcid.org/0000-0001-5192-9918

Alexandr Kuznetsov*
Department of Political Sciences, Communication and International Relations, University of Macerata, 62100, Macerata, Italy
Department security information systems and technologies of the V. N. Karazin Kharkiv National University, 4 Svobody Sq., Kharkiv, 61022, Ukraine
E-mail: kuznetsov@karazin.ua
ORCID iD: https://orcid.org/0000-0003-2331-6326
*Corresponding author

Kateryna Kuznetsova
Department security information systems and technologies of the V. N. Karazin Kharkiv National University, 4 Svobody Sq., Kharkiv, 61022, Ukraine
E-mail: kate7smith12@gmail.com
ORCID iD: https://orcid.org/0000-0002-5605-9293

Received: 29 August 2022; Revised: 20 October 2022; Accepted: 07 November 2022; Published: 08 February 2023

Abstract: An important task of designing complex computer systems is to ensure high reliability. Many authors investigate this problem and solve it in various ways. Most known methods are based on the use of natural or artificially introduced redundancy. This redundancy can be used passively and/or actively with (or without) restructuring of the computer system. This article explores new technologies for improving fault tolerance through the use of natural and artificially introduced redundancy of the applied number system. We consider a non-positional number system in residual classes and use the following properties: independence, equality, and small capacity of residues that define a non-positional code structure. This allows you to: parallelize arithmetic calculations at the level of decomposition of the remainders of numbers; implement spatial spacing of data elements with the possibility of their subsequent asynchronous independent processing; perform tabular execution of arithmetic operations of the base set and polynomial functions with single-cycle sampling of the result of a modular operation. Using specific examples, we present the calculation and comparative analysis of the reliability of computer systems. The conducted studies have shown that the use of non-positional code structures in the system of residual classes provides high reliability. In addition, with an increase in the bit grid of computing devices, the efficiency of using the system of residual classes increases. Our studies show that in order to increase reliability, it is advisable to reserve small nodes and blocks of a complex system, since the failure rate of individual elements is always less than the failure rate of the entire computer system.

Index Terms: Reliability, Fault Tolerance, Non-Positional Number System, Residual Classes, Computer System.

1. Introduction

The modern stage in science and technology is distinguished by the need to implement increasingly complex computational problems that require timely and reliable real time solutions [1-3]. However, a volume and complexity of the tasks outstrips the pace of increasing power of existing computer systems (CS) of general and special purpose, functioning in positional binary number system (PNS). In this aspect, the main directions of improving CS in PNS are increasing user productivity and, primarily, reliability of their functioning [4-7]. Ensuring the fault-tolerance property of

a computing system can increase the reliability of its operation.

The reserves for increasing the user performance (speed) of computing in PNS is the use of CS created on the principle of parallelizing the problem (algorithm) at the level of micro-operations [8]. The concept of parallelism potentially increases the performance of CS. Theoretical, experimental and industrial researches in this area have made it possible to substantiate basic principles for constructing parallel computing systems, e.g. the creation of super-parallel computing systems based on the principles of data processing in non-positional number system in residue number system (RNS). Further increasing the computing power of the CS is currently associated with the use of such non-positional code data structures.

The fault tolerance property provides an ability to perform specified computational functions after failures, both by reducing the functioning quality within acceptable limits (e.g. by gradual degradation) and without deteriorating the functioning quality of computer system. Thus, taking into account the foretold, it is a relevant task to develop methods for improving fault tolerance in the process of functioning CS.

Currently, the main methods that are widely used in the construction of fault-tolerant CS in PNS is structural redundancy. There is a large number of different redundancy methods, but any of them is characterized by significant structural redundancy. Even when correcting single errors, it is most often necessary to increase the number of CS equipment by at least three times. Such a high structural redundancy is explained by the following: using redundancy leads to ignoring all the properties of specific types of CS [8-10].

Let make a brief view in the modern literature [4,7,9]. Firstly, the use of positional binary number system as the number system does not allow a radical increasing in the performance and fault tolerance of computer system. Secondly, there are results of basic research and specific technical developments that show the possibility of significantly increasing the speed of implementing integer arithmetic operations of addition, multiplication and subtraction by using non-positional number system residue number system. This is achieved by using the properties of RNS, i.e. independence, equality and low bit depth of the residues that determine non-positional code structure (NPCS). It allows to parallelize arithmetic calculations at the level of decomposition of the remainders of numbers; implement spatial diversity of data elements with the possibility of their subsequent asynchronous independent processing; perform tabular execution of arithmetic operations of the basic set and polynomial functions with a single-cycle sample of the result of modular operation. The above significantly improves the performance of the computer system [11-13].

However, the lack of the results of basic research on the use of RNS to increase fault tolerance hinders finding a solution to significantly increase the reliability of the operation of the computer systems. The main problem solved in this article is to increase the reliability and fault tolerance of a computer system operating in residual classes. This is an actual field of computer science research, which has many practically important applications.

The goal of the article is to increase an operation reliability of the CS by using the result of synthesizing fault tolerant computing structures in RNS.

2. Background

Suppose that at the design stage it is necessary to provide the necessary (predetermined) level of reliability of the computing system. It is possible to increase (ensure) reliability if the computer system will have a certain property, the use of which will allow it to be done. Such a property is defined and called fault tolerance [4,5,7]. The concept of fault tolerance can be understood as the property of the computer system to ensure its operational state in case of failures of the elements included in their composition.

In the definition of the term fault tolerance there are three main aspects of its use [8,9,14,15]:

- The fault tolerance property is laid down by the developers during the design of the computer system in order to increase its reliability; at the same time, the necessary level of fault tolerance is achieved mainly when using redundant (additional) technical means (introducing artificial structural and (or) other redundancy) in comparison with the necessary minimum to perform all the required functions of the computer system and components in full;
- The use of fault-tolerance properties allows to save the full or partial performance of the computer system;
- It is believed that the failure of the elements of the computer system is not associated with exposure not provided for by the operating conditions.

In the most cases, developers are interested in the fact of ensuring fault tolerance only while maintaining full operability, i.e. without reducing the quality of the computer system functioning. In the future, when considering the concept of fault tolerance, we will be interested only in such option for the operation of computer systems and components.

To provide the computer system with the fault tolerance property, at the design stage, it is necessary to provide not only an introduction and use of artificial redundancy (AR), i.e. use of various types of redundancy: structural, informational, functional, temporary and load, but also to identify and use the possible natural (“natural” available redundancy) redundancy (NR). In this regard, the main designer’s task to ensure the necessary level of fault tolerant operation of CS is to determine and use the existing internal reserves (IR) of computer systems for fault tolerance at the

pre-design stage, due to the used number system and, with this in mind, further select and apply the necessary reservation methods (introduction of IR). Accounting and use of NR will increase the reliability of computer systems and components.

There are the definitions of primary and secondary redundancy as applied to computing devices [16-18]. In this aspect, the primary redundancy is determined due to used number system in the computer system. Obviously, the secondary redundancy is determined due to the application of traditional backup methods widely used in various information systems to improve their individual characteristics. Primary redundancy for computer system coincides with the concept of natural redundancy of information processing systems, and secondary redundancy coincides with the concept of artificial redundancy. The need for the addition and uses of secondary redundancy is due to the requirements for the characteristics at the design stage of the computer system. Note that the selected and used number system significantly affects the following characteristics [13,18-21]:

- Structure (architecture) of CS;
- Principles of information processing (to a greater extent on methods and algorithms for performing arithmetic operations);
- Requirements for the use of the new element base;
- System and user performance of CS;
- Reliability, survivability, fault tolerance, operational characteristics and indicators of computer system, etc.

Quantitatively, a volume V_{PR} of computer systems and components equipment due to the presence of primary redundancy (the presence of redundancy only due to the used number system) is slightly less than a volume of equipment V_{NR} in the presence of natural redundancy (redundancy due not only to the used number system). The volume of additional equipment V_{SR} determined by the presence of secondary redundancy fully coincides with the volume of equipment V_{AR} due to the presence of artificial redundancy. An analysis of the number system influence on the structure and individual characteristics of computer system showed that it is completely correct to assume that $V_{PR} \approx V_{NR}$ [4,8,9].

In the traditional approach to the choice of the positional number system base, it is first necessary to ensure the following condition:

$$V_{PR} \to \min . \tag{1}$$

However, the fulfillment of the condition (1) is not always valid when developing computational structures, when a priori the problem arises of improving some characteristics of the computer system. It is possible that the option of constructing a computer system based on the fulfillment of condition (1), when solving the problem of increasing reliability, is not at all advisable. This feature is clearly manifested when used as number system, for example, residue number system.

It is known [11,13,21,22] that non-redundant computer system in residue number system contains ≈ (15-20)% more equipment V_{PR} than the computer system in positional binary number system with the same given bit grid without taking into account the addition of secondary redundancy. As preliminary studies have shown, in order to achieve a given level of fault tolerance of the computer system in residue classes, ≈ 50% less number of equipment is required than for system in positional binary number system. However, the lack of practical results of the synthesis of fault-tolerance computer system in RNS does not make it possible to show the efficiency of using non-position code structure to increase the reliability of computer system.

Thus, the results of a comparative analysis of well-known publications show that the known solutions do not allow to fully ensure the reliability and fault tolerance of computer systems. Known redundancy based approaches require very high excess. This requires unreasonably high hardware and software costs.

By studying and developing methods to improve the computer system reliability, it is expedient and convenient to divide the concepts of fault tolerance into two components, i.e. use two separate concepts: natural fault tolerance (NF) and artificial fault tolerance (AF). The introduced terms are conveniently used in the analysis and synthesis of reliable structures of computer systems [8,9,14]. These concepts quite fully reflect the essence of calculation methods and reliability increase.

Definition 1. Natural fault tolerance of computer system is a system property to maintain a workability by using only natural redundancy.

Definition 2. Artificial fault tolerance of computer system is a property incorporated in the design of computer system, the use of which allows to maintain a workability in the case of elements failures due to the use of both natural and artificial redundancy.

Obviously, natural fault tolerance determines an initial (existing) level of computer system reliability, and artificial fault tolerance determines the total (necessary) level of reliability. A general task of improving reliability can be formulated as the task of ensuring the fault tolerance of the computer system due to the simultaneous use of two types

of fault tolerance. Improving fault tolerance, as one of the properties of computer system reliability due to the use of natural and artificial redundancy can be implemented by the method of passive or active fault tolerance.

Definition 3. The method of passive fault tolerance (PFT) is a way to increase fault tolerance by using both natural and artificial redundancy without restructuring (permanent redundancy) of the computer system structure. This method is used at the design stage to increase the computer system reliability to a predetermined (necessary) level of reliability.

Definition 4. The method of active fault tolerance (AFT) is a way to increase fault tolerance by using both natural and artificial redundancy by restructuring (dynamic backup) the structure of the computer system. This method is also used at the design stage to increase the computer system reliability to a predetermined (necessary) level of reliability.

3. Synthesis of CS Structure in RNS

To calculate and comparative analyze the operational reliability of the computer system in PNS and RNS, we will conduct the synthesis of CS structure in RNS.

Let it be necessary to synthesize computer system in RNS for l - byte bit grid. Obviously, the results of solving the synthesis problem of computer system in residue classes will substantially depend on the type of structural reservation, i.e. constant or dynamic (AFT). Therefore, it is expedient to separately solve the problems of synthesizing the CS structure in RNS in the case of constant or dynamic structural redundancy. In [13,18], studies were carried out to increase the reliability of the CS based on the use of the method of active fault tolerance (dynamic structural redundancy) in RNS. The results of calculations and comparative analysis showed the high efficiency of the use of NCS in RNS to increase the reliability of the CS.

This article solves the problem of increasing the reliability of the CS by taking into account the results of the synthesis of CS structure in RNS, based on the use of the method of passive fault tolerance (permanent structural redundancy). In order to solve the problem of synthesizing the CS structure in RNS, in the case of constant structural redundancy with a loaded reserve without restoring a failed element, we determine a concept of the state vector $X^{(n)}{}_{RNS} = (x_1, x_2, \text{K}, x_i, \text{K}, x_n)$ of a redundant computer system in RNS. In this case, the role of the elements of the redundant system is played by computing paths (CP) for each module $m_i (i = \overline{1,n})$ of the residue number system, and the values $x_i = 0,1,2,\text{K}$ indicate the multiplicity of the reservation of a separate CP of computer system in residue classes by the corresponding module (when the value of the main CP equals to $x_i = 0$ by the module m_i, there are no redundant computing paths).

The task of synthesizing computer system in residue classes is formulated as follows: from the whole set $X^{(n)}{}_{RNS}$ of possible values of the state vector, it is necessary to determine the only reserve composition vector at which the reliability of the computer system in residue classes $P^{(n)}{}_{RNS}[X^{(n)}{}_{RNS}]$ would reach the maximum possible value.

Obviously, the solution to this synthesis problem is directly related to the formulation and solution of the inverse optimal reservation problem in residue classes. The inverse problem of optimal reservation in residue classes is formulated as follows: it is necessary to determine a vector $X^{(n)}{}_{RNS} = (x_1, x_2, \text{K}, x_i, \text{K}, x_n)$ of reserve composition for which $V^{(l)}{}_{add}$, at acceptable costs, the maximum probability of fault tolerant operation $P^{(n)}{}_{RNS}[X^{(n)}{}_{RNS}]$ would be achieved. Mathematically, this problem can be represented as follows:

$$\begin{cases} P^{(n)}{}_{RNS}\left[X^{(n)}{}_{RNS} = X^{(n)}{}_{RNS}\left(x_1, x_2, \text{K}, x_i, \text{K}\right)\right] = P^{(n)}{}_{RNS}\left[\prod_{i=1}^{n} P_{x_i}(t)\right] \to \max, \\ \left(V^{(n)}{}_{RNS} \le V^{(l)}{}_{add}\right), \end{cases} \tag{2}$$

where the x_i is an i-th component of the vector $X^{(n)}{}_{RNS}$ of the redundant computing system with modulo m_i of residue classes, which numerically characterizes a number of redundant computing paths connected to the main (working) computing path for this module (base) of residue classes; n is a number of bases m_i in residue number system; $P_{x_i}(t) = 1 - (1 - e^{-\alpha_i \cdot \lambda_{FR} \cdot t})^{x_i + 1}$ is a probability of fault tolerant operation of the redundant system with modulo m_i in residue number system; λ_{FR} is a failure rate of a conventional unit of equipment of the computer system, assigned to one binary digit of the bit grid of the computer system; $\alpha_i = \left\lceil \log_2 (m_i - 1) \right\rceil + 1$ is a number of binary digits needed to represent a module (the relative "cost" of one computing path in absolute value, expressed in binary digits; a value $V^{(n)}{}_0$ is a set limit on the cost of the system when solving the inverse optimal reservation problem in residue number system).

As the secondary redundancy in positional number system, to calculate and compare CS in RNS, we take the majority three-channel computing system, consisting of three same types l – discharge computing systems. This is the most widely used at present to increase the reliability of the computer system. In this case, without considering the reliability of the majority part, the amount of equipment is equal to $V_{add} = 3 \cdot 8 \cdot l$ conventional units. We note that the

probability of fault tolerant operation of a three-channel computing system in positional number system (without taking into account the reliability of the majority part) is equal to $P^{(l)}{}_{PNS}(t) = 1-[1-P^{(l)}{}_0(t)]^3$, where $P^{(l)}{}_0(t) = e^{-8\cdot l\cdot\lambda_{FR}\cdot t}$ is a probability of the fault tolerant operation of a $l-$ byte computer system in residue classes. Note that $V^{(n)}{}_0 = V^{(l)}{}_{add} - \sum_{i=1}^{n}\alpha_i$ is a difference between the permissible costs in positional number system and the costs necessary to build a fault tolerant computer system in residue classes (the $V^{(n)}{}_0$ values of the permissible restrictions on the creation of a redundant computer system in residue classes).

To solve the inverse problem (2) of optimal reservation formulated in the article in residue number system, a dynamic programming method is recommended in the literature. The approach using this method is very flexible for solving problems associated with multi-stage selection. In addition, the dynamic programming method due to the fact that the solutions are recurrence relations is very convenient for performing calculations for large numbers on a computer. To solve the inverse optimal reservation problem, when using dynamic programming, it is necessary to leave the main functional equation in the form:

$$\begin{cases} \max P^{(n)}{}_{RNS}\left[x_1, x_2, \text{K}, x_i, \text{K}, x_n\right] = \max \prod_{i=1}^{n} P_{x_i}(t), \\ \begin{pmatrix} 0 \le \sum_{i=0}^{n} a_i \cdot x_i \le V^{(n)}{}_0 \\ x_n = 0,1,2,\text{K} \end{pmatrix}. \end{cases} \tag{3}$$

We introduce into consideration some function $F_n(V^{(n)}{}_0)$, index n at which means the dimension of the maximized function $\prod_{i=0}^{n} P_{x_i}(t)$, and its argument is a permissible restriction imposed on the arguments of the function $V^{(n)}{}_0$. In this case, functional equation (3) can be written as:

$$\begin{cases} F_n(V^{(n)}{}_0) = \max P_{x_n} \cdot F_{n-1}(V^{(n)}{}_0 - a_n \cdot x_n), \\ \begin{pmatrix} 0 \le a_n \cdot x_n \le V^{(n)}{}_0 \\ x_n = 0,1,2,\text{K} \end{pmatrix}. \end{cases} \tag{4}$$

In view of the foregoing, a functional equation giving a recursive solution for the inverse optimal reservation problem in residue classes will be presented as follows:

$$\begin{cases} F_n\left[3\cdot 8\cdot l - \sum_{i=1}^{n} a_i\right] = \max P_{x_n} \cdot F_{n-1}\left[V^{(n)}{}_0 - a_n \cdot x_n\right], \\ \begin{pmatrix} 0 \le a_n \cdot x_n \le (3\cdot 8\cdot l - \sum_{i=1}^{n} a_i) \\ x_n = 0,1,2,\text{K} \end{pmatrix}. \end{cases} \tag{5}$$

The algorithm for solving the inverse optimal reservation problem in residue number system:

- The optimal two-dimensional vectors of the reserve composition are determined for the first and second computing path of computer system in residue classes, corresponding to the modules m_1 and m_2, for all values of the cost indicator, not exceeding the value $V^{(n)}{}_0$.
- The optimal three-dimensional reserve composition vectors for the third computing path are determined by modulo m_3 and the corresponding vectors (x_1, x_2) for all values of the cost indicator not exceeding the value $V^{(n)}{}_0$.
- A similar process continues until the optimal $(n-1)-$measured vector $X^{(n)}{}_{RNS} = (x_1, x_2, \text{K}, x_i, \text{K}, x_{n-1})$ of the reserve composition and the corresponding optimal vector for the value of the conditional cost indicator equal to $V^{(n)}{}_0$ are found.
- An optimal value x_n is determined, which, together with the value of the vector $(x_1, x_2, \text{K}, x_i, \text{K}, x_{n-1})$, gives the optimal solution to the problem.

4. Examples

Let's consider an example of solving the inverse optimal reservation problem in residue classes for a single-byte ($l=1$) bit network of computer system. Redundant computing paths of computer system in residue classes, in relation to the main (working) computing path, are in the load reserve, and failed computing paths are not restored. For the case when $l=1(n=4)$, RNS consists of four modules (bases). Table 1 presents a data for solving the inverse problem of optimal reservation in RNS of various l values of the bit $(l=\overline{1,4,8})$ grids of the CS.

Table 1. Initial Data for Solving the Inverse Optimal Reservation Problem in Residue Number System $(l=\overline{1,4,8})$.

The value of the $l-$ bit grid of computer system	The number of bases of residue number system	Set of bases $\{m_i\}$ of residue number system $i=\overline{1,n}$	$V^{(l)}{}_{add}$	$\sum_{i=1}^{n} a_i$	$V^{(n)}{}_0$
1	4	$m_1=3, m_2=4, m_3=5, m_4=7$	24	10	**14**
2	6	$m_1=3, m_2=4, m_3=5, m_4=7, m_5=11, m_6=13,$	48	19	**29**
3	8	$m_1=3, m_2=4, m_3=5, m_4=7, m_5=11, m_6=13,$ $m_7=17, m_8=19$	72	28	**44**
4	10	$m_1=3, m_2=4, m_3=5, m_4=7, m_5=11, m_6=13,$ $m_7=17, m_8=19, m_9=23, m_{10}=29$	96	37	**59**
8	16	$m_1=3, m_2=4, m_3=5, m_4=7, m_5=11, m_6=13,$ $m_7=17, m_8=19, m_9=23, m_{10}=29, m_{11}=31,$ $m_{12}=37, m_{13}=41, m_{14}=43, m_{15}=47, m_{16}=53$	192	72	**120**

For the value $l=1$, the inverse optimal reservation problem in residue classes is formulated as follows: it is necessary to determine a reserve composition vector $X^{(n)}{}_{RNS}=X^{(4)}{}_{RNS}=(x_1,x_2,x_3,x_4)$ for which at acceptable costs $V^{(l)}{}_{add}=3\cdot 8\cdot l=3\cdot 8\cdot 1=24$ of conventional units, and the fault tolerant operation (probability of fault-tolerance operation) $P^{(4)}{}_{RNS}[X^{(4)}{}_{RNS}]$ of the computer system in residue classes would reach the maximum possible value. Note that the cost of one element of the $i-$ type of the redundant system (the conditional volume V_{m_i} of computing paths equipment operating by modulo m_i) is determined by the number of binary digits $\alpha_i=[\log_2(m_i-1)]+1$ required to represent the number m_i. The indicator of the necessary conditional costs $V^{(n)}{}_0$ (specified restrictions on the cost of the reserved system) is defined as the difference between the allowable costs $V^{(l)}{}_{add}$ in positional number system and the costs $\sum_{i=1}^{n} a_i$ necessary to build breakeven computer system in residue classes, i.e.

$$V^{(n)}{}_0=V^{(l)}{}_{add}-\sum_{i=1}^{n}\alpha_i$$

For the value $l=1$, we can write (Table 1) the following:

$$V^{(n)}{}_0=14=24-10$$

where

$$V^{(l)}{}_{add}=3\cdot 8\cdot 1=24 \text{ and } \sum_{i=1}^{n} a_i=2+2+3+3=10$$

In order to obtain a solution to the inverse problem of optimal redundancy in residue number system while $l=1$ and conducting a comparative analysis of the reliability of the computer system in positional number system and in residue classes, we give an example of the calculation of reliability (expression (5)). As initial data, we take, for example, that the value λ_{FR} is equal $\lambda_{FR}=0,1[1/hour]$, and the given operating time of the computer system, assigned to one binary digit of the bit grid of the computer system, is equal to $t_N=0,1$ an hour. In this case, we obtain that $P_E(t_N)=e^{-\lambda_E\cdot t}=e^{-0,01}=0,99$ is a probability of fault tolerant operation of the equipment, assigned to one binary

discharge of the bit grid of the computer system. The probability of failure $P_i(t_N)(i=\overline{1,4})$ of one computing path by modulo m_i equal to:

$$P_1(t_N)=e^{-a_1\cdot\lambda_{FR}\cdot t_N}=e^{-2\cdot 0,01}\approx 0,98;$$

$$P_2(t_N)=e^{-a_2\cdot\lambda_{FR}\cdot t_N}=e^{-2\cdot 0,01}\approx 0,98;$$

$$P_3(t_N)=e^{-a_3\cdot\lambda_{FR}\cdot t_N}=e^{-3\cdot 0,01}\approx 0,97;$$

$$P_4(t_N)=e^{-a_4\cdot\lambda_{FR}\cdot t_N}=e^{-2\cdot 0,01}\approx 0,98.$$

Let's pre-calculate the values of the unreliability indicator, i.e. we calculate the values of the failure probability $q_{x_i}=1-P_{x_i}$ of each of the four subsystems of the computer system (computing path) in residue number system by modulo m_i for the number of backup elements x_i not exceeding five. For a large value of the values of the failure probability $q_{x_i}=1-P_{x_i}$ exceeding five, it is impractical to calculate q_{x_i}, since they will not be used. The results of the calculation of the values q_{x_i} are presented in table 2.

Table 2. Initial Data for Solving the Optimal Reservation Problem for the Value $l=1$.

x_i	q_{x_1}	q_{x_2}	q_{x_3}	q_{x_4}
0	$2\cdot10^{-2}$	$2\cdot10^{-2}$	$3\cdot10^{-2}$	$3\cdot10^{-2}$
1	$4\cdot10^{-4}$	$4\cdot10^{-4}$	$9\cdot10^{-4}$	$9\cdot10^{-4}$
2	$8\cdot10^{-6}$	$8\cdot10^{-6}$	$27\cdot10^{-6}$	$27\cdot10^{-6}$
3	$16\cdot10^{-8}$	$16\cdot10^{-8}$	$81\cdot10^{-8}$	$81\cdot10^{-8}$
4	$32\cdot10^{-10}$	$32\cdot10^{-10}$	$243\cdot10^{-10}$	$243\cdot10^{-10}$
5	$64\cdot10^{-12}$	$64\cdot10^{-12}$	$729\cdot10^{-12}$	$729\cdot10^{-12}$

In the further calculations of reliability, we use an approximate expression of the form: $P_{x_i}\cdot P_{x_{n-1}}\approx 1-q_{x_n}-q_{x_{n-1}}$.

In accordance with the above algorithm for solving the inverse optimal reservation problem in residue classes and based on the initial data shown in Tables 1 and 2, for the value $l=1$ we obtain the desired optimal vector $X^{(4)}{}_{RNS}=(1,1,1,2)$, the value of the i-th coordinate of which $(i=\overline{1,4})$ is equal to the number of backup computing paths connected to the operating computing path for this base m_i of the residue classes (table. 3).

Table 3. The Result of Solving the Optimal Reservation Problem $j=1,2,3,4,8$.

l(n)	$X^{(n)}{}_{OPT}=(x_1,x_2,\text{K},x_i,\text{K},x_n)$	$P^{(n)}_{COK}(t_n)$	$P^{(l)}_{IICC}(t_n)$	$K_H(t_H)$
1(4)	(1, 1, 1, 2)	0,9983	0,9995	-
2(6)	(1, 2, 2, 1, 1, 2)	0,9966	0,9963	1,0983
3(8)	(1, 2, 1, 1, 1, 2, 2, 2)	0,9959	0,9902	2,3809
4(10)	(1, 2, 1, 1, 1, 1, 2, 2, 2, 2)	0,9944	0,9787	3,7973
5(16)	(2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1)	0,9800	0,9101	4,5101

Table 3 also presents the results of solving the inverse optimal reservation problem in residue classes for the values l_j while $j=2,3,4,8$. Based on the obtained set of optimal vectors $\left\{X^{(n)}{}_{OPT}\right\}$ (Table 3), it is easy to obtain analytical ratios $P^{(n)}{}_{RNS}$ for assessing the reliability of computer system in residue classes (Table 4).

In order to verify the correctness of the results of solving the inverse optimal reservation problem in residue classes, the article presents calculated values $V_{calc}{}^{(n)}{}_0$ of the conditional amount of the equipment of the computer system in residue classes, presented in tables 5-9. In accordance with the necessary condition $V_{calc}{}^{(n)}{}_0\le V^{(n)}{}_0$ of the inverse optimal reservation problem in residue classes, a comparative analysis of the initial values $V^{(n)}{}_0$ (Table 1) and the calculated values $V_{calc}{}^{(n)}{}_0$ (Table 5-9) was performed. The results of the comparative analysis showed the correctness of the obtained results of solving the inverse optimal reservation problem in residue classes.

Table 4. Analytical Ratios $P^{(n)}_{RNS}$ for Assessing the Reliability of the Computer System in Residue Classes for $j = 2,3,4,8$.

$P^{(4)}_{RNS}(t) = \prod_{i=1}^{4} P_{x_i}(t) = \left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_1-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^2\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_2-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^2\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_3-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^2\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_4-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)$
$P^{(6)}_{RNS}(t) = \prod_{i=1}^{6} P_{x_i}(t) = \left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_1-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^2\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_2-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_3-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_4-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^2\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_5-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^2\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_6-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)$
$P^{(8)}_{RNS}(t) = \prod_{i=1}^{8} P_{x_i}(t) = \left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_1-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^2\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_2-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_3-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^2\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_4-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^2\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_5-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^2\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_6-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_7-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_8-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)$
$P^{(10)}_{RNS}(t) = \prod_{i=1}^{10} P_{x_i}(t) = \left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_1-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^2\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_2-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_3-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^2\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_4-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^2\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_5-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^2\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_6-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^2\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_7-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_8-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_9-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)\times$ $\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_{10}-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)$

$$P^{(16)}_{RNS}(t)=\prod_{i=1}^{16}P_{x_i}(t)=\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_1-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)\times$$
$$\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_2-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)\times$$
$$\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_3-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)\times$$
$$\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_4-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)\times$$
$$\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_5-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)\times$$
$$\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_6-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)\times$$
$$\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_7-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)\times$$
$$\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_8-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)\times$$
$$\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_9-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)\times$$
$$\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_{10}-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)\times$$
$$\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_{11}-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)\times$$
$$\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_{12}-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^3\right)\times$$
$$\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_{13}-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^2\right)\times$$
$$\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_{14}-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^2\right)\times$$
$$\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_{15}-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^2\right)\times$$
$$\left(1-\left\{1-\exp\left[-\left\{\left[\log_2(m_{16}-1)\right]+1\right\}\cdot\lambda_{FR}\cdot t\right]\right\}^2\right)$$

Table 5. Data to Verify the Solution of the Inverse Optimal Reservation Problem in Residue Classes for $l=1$.

m_i	3	4	5	7	$V_{calc}{}^{(4)}{}_0$
α_i	2	2	3	3	10
x_i	1	1	1	2	–
$\alpha_i \cdot x_i$	2	2	3	6	**13**

Table 6. Data to Verify the Solution of the Inverse Optimal Reservation Problem in Residue Classes for $l=2$.

m_i	2	5	7	9	11	13	$V_{calc}{}^{(6)}{}_0$
α_i	1	3	3	4	4	4	19
x_i	1	2	2	1	1	2	–
$\alpha_i \cdot x_i$	1	6	6	4	4	8	**29**

Table 7. Data to Verify the Solution of the Inverse Optimal Reservation Problem in the Residue Classes for $l=3$.

m_i	3	4	5	7	11	13	17	19	$V_{calc}{}^{(8)}{}_0$
α_i	2	2	3	3	4	4	5	5	28
x_i	1	2	1	1	1	2	2	2	–
$\alpha_i \cdot x_i$	2	4	3	3	4	8	10	10	**44**

Table 8. Data to Verify the Solution of the Inverse Optimal Reservation Problem in Residue Classes for $l=4$.

m_i	2	3	5	7	11	13	17	19	23	29	$V_{calc}{}^{(10)}{}_0$
α_i	1	2	3	3	4	4	5	5	5	5	37
x_i	1	2	1	1	1	1	2	2	2	2	–
$\alpha_i \cdot x_i$	1	4	3	3	4	4	10	10	10	10	**59**

Table 9. Data to Verify the Solution of the Inverse Optimal Reservation Problem in Residue Classes for $l = 8$.

m_i	2	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	$V_{calc}{}^{(16)}{}_0$
α_i	1	2	3	3	4	4	5	5	5	5	5	6	6	6	6	6	72
x_i	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	–
$\alpha_i \cdot x_i$	2	4	6	6	8	8	10	10	10	10	10	12	6	6	6	6	**120**

We evaluate an effectiveness of the computer system in RNS and PNS as a ratio of the reliability of two redundant computing systems. In reliability theory, there is a criterion for evaluating the effectiveness of redundancy. This criterion is a coefficient $K_N(t_N)$ of reliability increase, and it is defined as the ratio of the failure probabilities of two redundant computer systems at a given operating time t_N, i.e.

$$K_N(t_N) = \frac{1 - P^{(l)}{}_{PNS}(t_N)}{1 - P^{(n)}{}_{RNS}(t_N)}.$$

The coefficient $K_N(t_N)$ characterizes the decrease in the failure probability of the computer system in residue classes compared to the computer system in positional number systems. The results of the calculation of values $K_N(t_N)$ are summarized in the table 3, which contains the results of solving the inverse optimal reservation problem in residue classes for $l-$byte $(l = \overline{1,4,8})$ bit grids. The results of solving this problem showed that the use of residue number system for $l \geq 2$ provides a higher value of the probability of fault tolerant operation $P^{(l)}{}_{RNS}(t)$ than the method of tripling of the computer system widely used in positional binary number system. Note that with an increase in the value of the computer system $l-$bit grid, the efficiency of using residue classes increases. Structures of computer system in residue classes for $l-$byte bit grids are presented in the Fig. 1 – 5.

Fig. 6 is a graphical representation of the main result obtained. These are the values of the coefficient of reliability improvement for all considered cases of $l-$ byte bit grids of the CS.

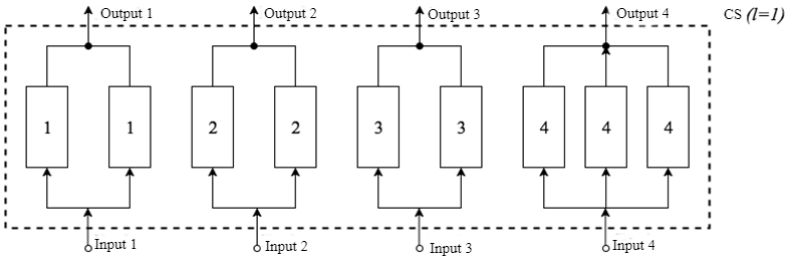

Fig.1. Structural Diagram of Computer System in Residue Classes with *l*=1.

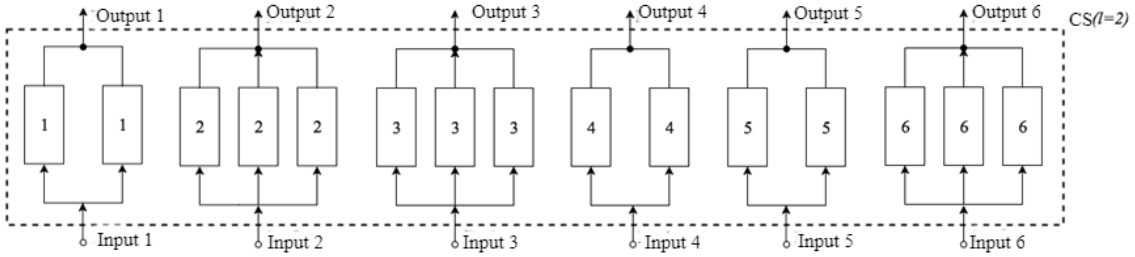

Fig.2. Structural Diagram of Computer System in Residue Classes with l=2.

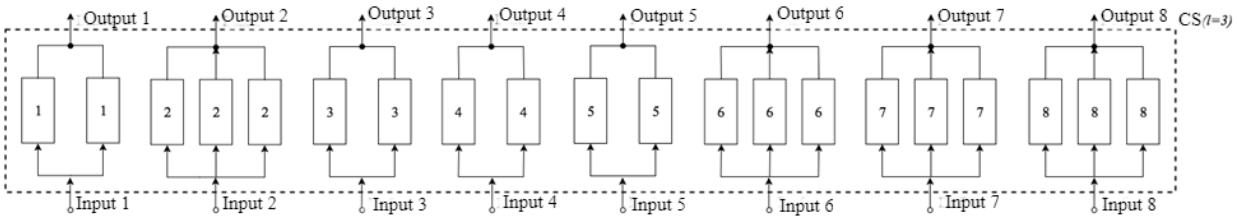

Fig.3. Structural Diagram of Computer System in Residue Classes with *l*=3.

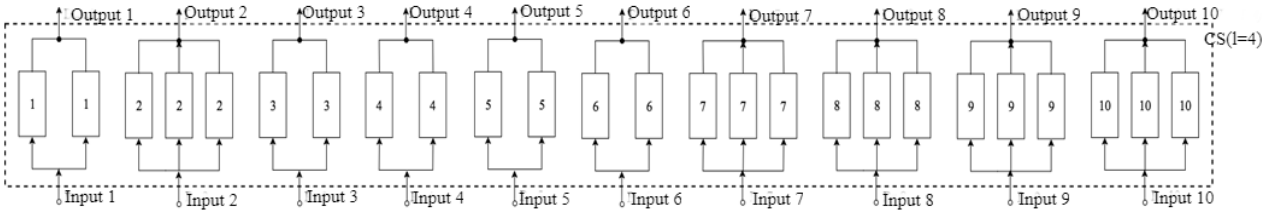

Fig.4. Structural Diagram of Computer System in Residue Classes with l=4.

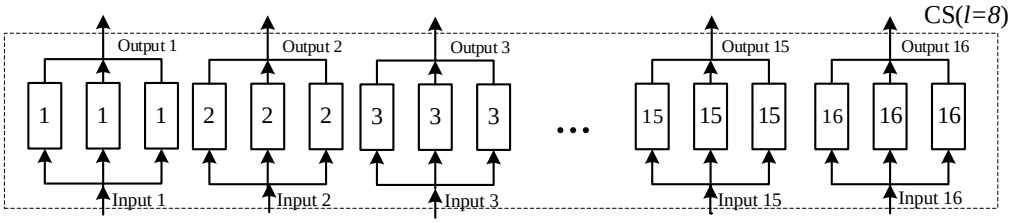

Fig.5. Structural Diagram of Computer System in Residue Classes with l=8.

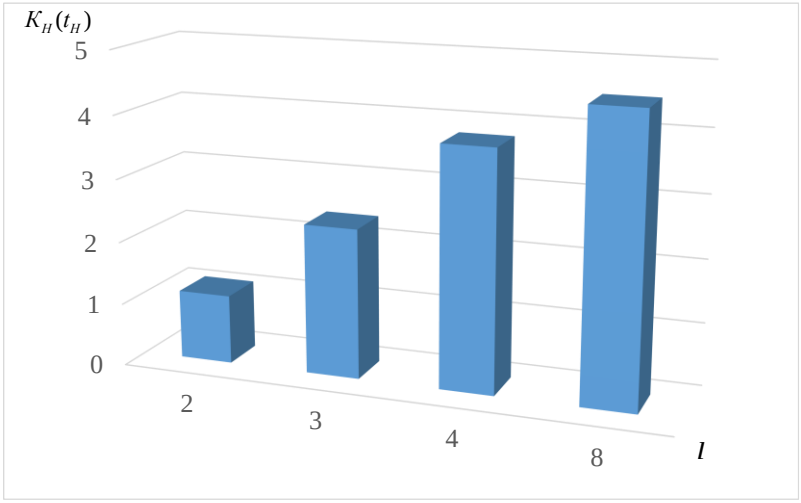

Fig.6. Values of the Coefficient of Reliability Improvement for Different Cases.

The conducted studies show that the proposed approach really makes it possible to increase the reliability of computing systems and components. At the same time, with an increase in the bit grid, the coefficient of reliability increase also increases.

5. Conclusion

Based on the research results presented in this article, the following main conclusions can be drawn.

1. This article presents a new method to increase the reliability of the computer system by using the available redundancy of the number system. The concept assumes that in the process of designing computer systems and components, accounting and possibility of using natural redundancy (account of used number system) and artificial redundancy (reservation methods) are made. The basis of these methods is PFT and AFT, which are based on the joint use of natural and artificial redundancy. This fact allows to set and solve the problem of achieving the required level of reliability at the design stage of the computer system for any applicable number system. With the combined use of natural and artificial fault tolerance, on the basis of known methods for improving reliability, the maximum value of the reliability of the CS, due to the total redundancy, can be achieved.

2. As an example of using the proposed concept, the CS in RNS is considered. For this purpose, the inverse problem of optimal redundancy in RNS for l-byte bit grids of the CS is formulated and solved. Based on the result of solving the problem, analytical relationships are obtained for calculating the probability of failure-free operation of the CS in RNS for l-byte $(l = \overline{1,4,8})$ bit grids, and the structures of the CS in RNS are synthesized.

3. The calculation and comparative analysis of the reliability of the CS in the RNS and the CS in PNS was made, which confirmed the high efficiency of using the NCS in RNS. The conducted studies have shown that, firstly, the use of non-positional code structures in RNS provides a higher reliability than the method of majority triaging of the same

type of CS, which is widely used in PNS, and with a smaller, additional input amount of equipment; secondly, with an increase in the bit grid of the CS, which is typical for the modern trend in the development of computing systems and tools, the efficiency of using RNS increases. It should be noted that in RNS the primary structural redundancy is significantly manifested only in the presence of secondary structural redundancy.

4. Theoretically, the increase in the reliability of the CS in RNS is explained as follows. First, based on the properties of independence and equality of the remainders of the number, the totality of which determines the NCS, the structure of the CS in RNS is already initially a type of structural redundant system. Each of the elements (computing paths) of this redundant system operates independently of each other and in parallel in time according to its separate module m_i. In PNS, this is equivalent to the fact that, as it were, smaller nodes of the CS system are reserved. Secondly, in accordance with the provisions of the general theory of reliability, in order to increase the reliability of the CS, it is advisable to reserve small units and blocks of a complex system, since the failure rate of the CS elements is always less than the failure rate of the entire CS. In this aspect, the CS in RNS, consisting of a set of individual secondary redundancy, will be more reliable than the equivalent CS in PNS.

The obtained research results can be used to improve the reliability of computer systems and components. In particular, the article shows that the redundancy of elements at the level of computational operations can reduce the probability of failure. This can be used in practice to improve computing devices operating in the system of residual classes. Thus, modeling of new computational components with increased operational reliability is a promising direction for further research.

References

[1] R. F. Hodson, *Real-Time Expert Systems Computer Architecture*. CRC Press, 2018. doi: 10.1201/9781351076203.

[2] R. S. Alford, *Computer Systems Engineering Management*. CRC Press, 2018. doi: 10.1201/9781351070829.

[3] A. Yadin, *Computer Systems Architecture*, 1st ed. Boca Raton : Taylor & Francis Group, CRC Press, 2016. | Series: Chapman and Hall/CRC, 2016. doi: 10.1201/9781315373287.

[4] I. Koren and C. M. Krishna, "Fault-Tolerant Networks," in *Fault-Tolerant Systems*, Elsevier, 2021, pp. 115–159. doi: 10.1016/B978-0-12-818105-8.00014-0.

[5] Q. Hu, B. Xiao, B. Li, and Y. Zhang, "Fault-tolerant velocity-free attitude control," in *Fault-Tolerant Attitude Control of Spacecraft*, Elsevier, 2021, pp. 125–173. doi: 10.1016/B978-0-32-389863-8.00015-0.

[6] S. X. Ding, "Performance Recovery and Fault-Tolerant Control Schemes," in *Advanced methods for fault diagnosis and fault-tolerant control*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2021, pp. 553–600. doi: 10.1007/978-3-662-62004-5_20.

[7] P. Castaldi, N. Mimmo, and S. Simani, "Fault diagnosis and fault-tolerant control techniques for aircraft systems," in *Fault Diagnosis and Fault-tolerant Control of Robotic and Autonomous Systems*, A. Monteriu, A. Freddi, and S. Longhi, Eds. Institution of Engineering and Technology, 2020, pp. 197–212. doi: 10.1049/PBCE126E_ch9.

[8] A. M. Polovko, *Fundamentals of reliability theory*. Published: New York - London: Academic Press., 1968.

[9] I. A. Ushakov, Ed., *Optimal Resource Allocation: With Practical Statistical Applications and Theory*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2013. doi: 10.1002/9781118400715.

[10] I. Koren and C. M. Krishna, "Chapter 2 - Hardware Fault Tolerance," in *Fault-Tolerant Systems (Second Edition)*, I. Koren and C. M. Krishna, Eds. San Francisco (CA): Morgan Kaufmann, 2021, pp. 11–57. doi: 10.1016/B978-0-12-818105-8.00012-7.

[11] V. Krasnobayev, A. Kuznetsov, A. Yanko, B. Akhmetov, and T. Kuznetsova, "Processing of the Residuals of Numbers in Real and Complex Numerical Domains," in *Data-Centric Business and Applications*, vol. 48, T. Radivilova, D. Ageyev, and N. Kryvinska, Eds. Cham: Springer International Publishing, 2021, pp. 529–555. doi: 10.1007/978-3-030-43070-2_24.

[12] V. Krasnobayev, A. Kuznetsov, S. Koshman, and S. Moroz, "Improved Method of Determining the Alternative Set of Numbers in Residue Number System," in *Recent Developments in Data Science and Intelligent Analysis of Information*, vol. 836, O. Chertov, T. Mylovanov, Y. Kondratenko, J. Kacprzyk, V. Kreinovich, and V. Stefanuk, Eds. Cham: Springer International Publishing, 2019, pp. 319–328. doi: 10.1007/978-3-319-97885-7_31.

[13] V. Krasnobaev, A. Kuznetsov, V. Popenko, and T. Kuznetsova, "Mathematical Model of the Reliability of a Computer System which is Functioning in the Residual Class System, Taking into Account the Reliability of Switching Devices," in *2021 IEEE 4th International Conference on Advanced Information and Communication Technologies (AICT)*, Sep. 2021, pp. 225–229. doi: 10.1109/AICT52120.2021.9628929.

[14] V. Piuri, "Fault-tolerant systolic arrays: An approach based upon residue arithmetic," in *1987 IEEE 8th Symposium on Computer Arithmetic (ARITH)*, May 1987, pp. 230–238. doi: 10.1109/ARITH.1987.6158712.

[15] M. H. Weik, "computer system fault tolerance," in *Computer Science and Communications Dictionary*, M. H. Weik, Ed. Boston, MA: Springer US, 2001, pp. 274–274. doi: 10.1007/1-4020-0613-6_3420.

[16] I. A. Ushakov, "Formulation of Optimal Redundancy Problems," in *Optimal Resource Allocation*, John Wiley & Sons, Ltd, 2013, pp. 33–47. doi: 10.1002/9781118400715.ch2.

[17] I. A. Ushakov, "Dynamic Programming," in *Optimal Resource Allocation*, John Wiley & Sons, Ltd, 2013, pp. 69–84. doi: 10.1002/9781118400715.ch5.

[18] V. Krasnobaev, A. Kuznetsov, A. Kiian, and K. Kuznetsova, "Fault Tolerance Computer System Structures Functioning in Residue Classes," in *2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Sep. 2021, vol. 1, pp. 471–474. doi: 10.1109/IDAACS53288.2021.9660919.

[19] Y. Zhang, "An FPGA implementation of redundant residue number system for low-cost fast speed fault-tolerant computations," Thesis, 2018. doi: 10.32657/10220/47113.

[20] Ya. M. Nikolaychuk, *Specialized computer technologies in information*. Ternopil: TzOV «Terno-graph», 2017.

[21] P. V. A. Mohan, *Residue Number Systems*. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-41385-3.

[22] I. Koren, “THE RESIDUE NUMBER SYSTEM,” *Computer Arithmetic Algorithms*, Oct. 08, 2018. https://www.taylorfrancis.com/ (accessed Aug. 16, 2020).

Authors’ Profiles

Victor Krasnobayev: Doctor of Sciences (Engineering), Full Professor, a Professor of the Department of Electronics and Control Systems of V.N. Karazin Kharkiv National University, Ukraine. Areas of scientific interests: the theory and practice of creating computer systems and components in a residue numeral system, non-positional number systems and technologies for increasing fault tolerance.
Email: v.a.krasnobaev@gmail.com

Alexandr Kuznetsov: Doctor of Sciences (Engineering), Full Professor, Academician of the Academy of Applied Radioelectronics Sciences. Laureate of the Boris Paton National Prize of Ukraine (2021). Visiting Professor of the Department of Political Sciences, Communication and International Relations, University of Macerata, 62100, Macerata, Italy. Professor of the Department security information systems and technologies of the V. N. Karazin Kharkiv National University, Ukraine. Areas of scientific interests: non-positional number systems, technologies for increasing fault tolerance, applied cryptology and coding theory.
Email: kuznetsov@karazin.ua

Kateryna Kuznetsova: She received her bachelor's degree in cybersecurity in 2022. The theme of the bachelor's thesis is devoted to the study of post-quantum cryptography algorithms. She is currently studying for a master's program at the Department security information systems and technologies of the V. N. Karazin Kharkiv National University, Ukraine. Her research topics include non-positional number systems, technologies for increasing fault tolerance, information security, block and stream ciphers.
Email: kate7smith12@gmail.com

How to cite this paper: Victor Krasnobayev, Alexandr Kuznetsov, Kateryna Kuznetsova, "Synthesis of the Structure of a Computer System Functioning in Residual Classes", International Journal of Computer Network and Information Security(IJCNIS), Vol.15, No.1, pp.1-13, 2023. DOI:10.5815/ijcnis.2023.01.01