UDC 681.142 **doi:** https://doi.org/10.20998/2522-9052.2023.2.06

Serhii Koshman[1], Victor Krasnobayev[1], Serhii Nikolsky[2], Dmytro Kovalchuk[1]

[1] V. N. Karazin Kharkiv National University, Kharkiv, Ukraine
[2] Kharkiv humanitarian-pedagogical academy, Kharkiv, Ukraine

THE STRUCTURE OF THE COMPUTER SYSTEM IN THE RESIDUAL CLASSES

Abstract. The **subject** of the article is the formulation and solution of the inverse problem of optimal redundancy in the system of residual classes (RNS) based on the use of the dynamic programming method. The solution of this problem makes it possible to improve the reliability of the operation of computer systems and components (CSC) in the RNS. The **purpose** of the article is to increase the reliability of the functioning of CSC, which are built on the basis of the use of RNS, without reducing the speed of calculations, as well as to calculate and compare the reliability, in terms of the probability of failure-free operation, of CSC in RNS and a tripled computing system that operates in a positional binary number system (PNS). **Tasks**: to analyze the influence of the number system used on the reliability of the CSC, taking into account the primary and secondary redundancy; to synthesize a computing system in RNS for a *l*-byte bit grid based on the use of the passive fault tolerance method (constant structural redundancy); formulate and solve the inverse problem of optimal redundancy in RNS based on the use of the dynamic programming method; to check the correctness of the results obtained, calculate the conditional amount of computer system equipment in the residual classes; evaluate the efficiency of using RNS to improve reliability when building a redundant CSC in relation to a redundant CSC in the PNS. **Research methods**: methods of analysis and synthesis of computer systems, number theory, coding theory in RNS, reliability theory. The following **results** are obtained. The paper shows that the use of PNS as a number system does not allow a radical increase in the performance and reliability of CSC. In this regard, the article developed the concept of using RNS as a number system for constructing a CSC. Based on this, the inverse problem of optimal redundancy in RNS is formulated and solved. **Conclusions**. As shown by the results of calculations and comparative analysis, the use of RNS provides a higher reliability of the CSC than the majority three-channel computing system in the PNS. The obtained research results can be used for the synthesis of fault-tolerant computer structures in RNS.

Keywords: non-positional code structure; system in residue classes; positional binary number system; reliability of computer systems and components.

Introduction

Modern trends in the development of computer technology cause its widespread introduction into various spheres of human activity. At the same time, large volumes of tasks to be solved require an increase in productivity and ensuring a given level of reliability of computing systems (CS). In addition, the content and complexity of such requests outstrip the pace of increasing power of existing computer systems and components (CSC) of general and special purpose, functioning in positional binary number system (PNS). In this aspect, the main directions of improving computer systems and components in positional number system are increasing user productivity and reliability (primarily reliability) of their functioning [1-3]. Ensuring the presence of a fault tolerance property in a CS can increase the reliability of its operation.

The fault tolerance property provides an ability to perform specified computational functions after failures, both by reducing, within acceptable limits, any indicators of the functioning quality (for example, by gradual degradation), and without deteriorating the functioning quality of computer system. Thus, considering the above, research in the field of developing methods for improving fault tolerance in the process of functioning of computer systems and components are relevant.

Reserves for increasing the speed of computing in positional number system are the use of computer system and components, created on the principle of problem (algorithm) parallelization at the level of microoperations. The concept of parallelism has long attracted the attention of specialists with its potential to increase the performance of computing systems. The theoretical, experimental and industrial developments in this direction have made it possible to substantiate the basic principles for constructing parallel computing systems. The prospect of further increasing a computing power of devices is currently associated with such systems [5-7].

The main methods that are widely used in the construction of fault tolerant computing devices and systems in positional number system are structural redundancy. There are a large number of different backup methods, but some of them is characterized by significant structural redundancy. Even with the correction of single errors, most often it is necessary to increase the volume of the computer system equipment at least three times. Such a high structural redundancy is explained by the fact that when applying redundancy, all the specific properties of specific types of computing systems and components are almost completely ignored [8-10].

In the modern literature, it is shown, that, 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 systems and components. 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 a non-positional number system in residue classes (RNS). This is achieved through the use of the following properties of residue system: independence, equality and low bit depth of the residues that determine a non-position code structure (NCS), which allows the following: parallelize arithmetic calculations at the level of decomposition of the remainders of numbers; realize 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 computer system and components performance.

However, the lack of the results of basic research on the use of systems in a residue classes to increase fault tolerance hinders the solution of the problem of a significant increase in the reliability of the operation of the computer systems and components.

The purpose of the article is to increase the reliability of the functioning of CSC, which are built on the basis of the use of non-positional number system in residue classes, without reducing the speed of calculations, as well as to calculate and compare the reliability, in terms of the probability of failure-free operation, of CSC in RNS and a tripled computing system, which operates in a positional binary number system.

Problem statement

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 [11-13]. With respect to the computer system, 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: 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 is to identify (determine) and use the existing internal reserves (IR) of computer systems and components for fault tolerance at the pre-design stage, due to the number system used and, with this in mind, in the future, select and apply the necessary reservation methods (introduction of IR). Accounting and use of NR will increase the reliability of computer systems and components.

In [14-16], as applied to computing devices, particular definitions of primary and secondary redundancy were introduced. In this aspect, it is believed that the primary redundancy is due to used number system in the computer systems and components. Obviously, secondary redundancy is redundancy 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 - 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: structure (architecture); 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; 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 $V_{PR} \approx V_{NR}$. In the traditional approach to the choice of the positional number system base, it is first necessary to ensure the following condition:

$$V_{PR} = \min . \qquad (1)$$

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 [17-19] that redundant computer system with residue number system contains (15-20)% more equipment V_{PR} than a computer system in a 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 volume of equipment is required than for system in a positional binary number system. However, the lack of practical results of the synthesis of fault-tolerance computer system in residue classes does not make it possible to show the efficiency of using the non-position code structure to increase the reliability of the computer system functioning without reducing the performance of solving problems.

Statement and solution of the inverse problem of optimal redundancy

For the purpose of calculating and comparative analysis of the operational reliability of the computer system in positional number system and in residue classes, we will conduct the synthesis of the fault tolerant computer system with residue number system. Let it be necessary to synthesize a computer system in residue number system for a 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 used: constant or dynamic.

Therefore, it is advisable to separately solve the synthesis problem in the case of constant or dynamic structural redundancy.

This article solves the problem of synthesizing the structure of the CS, based on using the method of passive fault tolerance (permanent structural redundancy).

In order to solve the problem of synthesis of computer system in residue classes in the case of constant structural redundancy with a loaded reserve without restoring a failed element, we introduce a concept of the state vector

$$X^{(n)}{}_{RNS} = (x_1, x_2, \ldots, x_i, \ldots, x_n)$$

of a redundant computer system in residue classes.

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 $x_i = 0, 1, 2, \ldots$ 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). Verbally, 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

$$X^{(n)}{}_{RNS} = (x_1, x_2, \ldots, x_i, \ldots, x_n)$$

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:

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

Mathematically, this problem can be represented as follows where:

the x_i is an $i$ts 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;

$\alpha_i = \left[\log_2\left(m_i - 1\right)\right] + 1$ 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) [20, 21].

As the computer system in positional number system, used for comparison with the computer system in residue classes, we take the majority three-channel computing system, consisting of three same type $l-$ 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 $V_0^{(n)}$ 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 [6]. 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 numerical calculations 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:

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

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

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

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

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

The algorithm for solving the inverse optimal reservation problem in residue number system is as follows.

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

$$X^{(n)}{}_{RNS} = (x_1, x_2, \ldots, x_i, \ldots, x_{n-1})$$

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 $(x_1, x_2, \ldots, x_i, \ldots, x_{n-1})$, gives the optimal solution to the problem.

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 $l = 1 (n = 4)$, the RNS consists of four modules (bases). Table 1 presents a data for solving the inverse problem of optimal reservation in residue number system of various l values of the bit $(l = \overline{1, 4}, 8)$ grids in the computer system.

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.

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**

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

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

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*t 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 also presents the results of solving the inverse optimal reservation problem in residue classes for the values l_j while $j = 2,3,4,8$. 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 4-8.

Table 3 – **The result of solving the optimal reservation problem** $j = 1,2,3,4,8$

$l\ (n)$	$X^{(n)}{}_{OPT} = (x_1, x_2,\ldots,\ x_i,\ldots,x_n)$	$P_{RNS}^{(n)}(t_н)$	$P_{PNS}^{(l)}(t_н)$	$K_H(t_H)$
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
8 (16)	(2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1)	0,9800	0,9101	4,5101

Table 4 – **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 5 – **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 6 – **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 7 – **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 8 – **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**

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 4-8) 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.

We evaluate an effectiveness of the computer system in residue classes and in positional number system 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.

Conclusions

A new concept is proposed 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. When implementing PFT or AFT, the essence of which is to identify (determine) a natural redundancy of the computer system through the use of the applicable number system. With the combined use of natural and artificial redundancy, on the basis of well-known methods for increasing reliability, the maximum value of the operational reliability of the computer system due to the total redundancy can be achieved. Note that in residue number system, primary structural redundancy is significantly manifested only in the presence of secondary structural redundancy. As an example of the use of the proposed concept, the computer system is considered in residue classes. For this, the inverse problem of optimal redundancy in residue classes for *l*-byte bit grids is formulated and solved.

REFERENCES

1. Kamaraj, A. and Marichamy, P. (2019), “Design of integrated reversible fault-tolerant arithmetic and logic unit”, *Microprocess Microsyst 69*, pp. 16-23, doi: https://doi.org/10.1016/j.micpro.2019.05.009.
2. Jain, S. and Gajjar, S. (2023), “Design and Implementation of Fault Tolerance and Diagnosis Technique for Arithmetic Logic Unit (ALU) in Soft-Core Processor”, Darji, A.D., Joshi, D., Joshi, A., Sheriff, R. (eds) *Advances in VLSI and Embedded Systems. Lecture Notes in Electrical Engineering*, vol. 962, Springer, doi: https://doi.org/10.1007/978-981-19-6780-1_10.
3. Krasnobaev, V., Kuznetsov, A., Kiian, A. and Kuznetsova, K. (2021), “Fault Tolerance Computer System Structures Functioning in Residue Classes”, *11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, pp. 471-474, doi: 10.1109/IDAACS53288.2021.9660919.
4. Ankit, K.V., Narasimham, S.M. and Prakash, V. (2015), “Design approach for fault recoverable ALU with improved fault tolerance”, *Int J VLSI Des Commun Syst* 6(4), doi: https://doi.org/10.5121/vlsic.2015.6402.
5. Vallero, A. (2016), “Cross-layer system reliability assessment framework for hardware faults”, *2016 IEEE International Test Conference (ITC)*, Fort Worth, TX, USA, pp. 1-10, doi: 10.1109/TEST.2016.7805863.
6. Krasnobaev, V., Kuznetsov, A., Popenko, V. and Kuznetsova, T. (2021), “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”, *IEEE 4th International Conference on Advanced Information and Communication Technologies (AICT)*, pp. 225-229, doi: 10.1109/AICT52120.2021.9628929.
7. Egwutuoha, I P, Levy, D, Selic, B. and Chen, S. (2013), “A survey of fault tolerance mechanisms and checkpoint/restart implementations for high performance computing systems”, *The Journal of Supercomputing*, 65(3), pp. 1302–1326, available at: https://link.springer.com/article/10.1007/s11227-013-0884-0.
8. Krasnobayev, V. A., Koshman, S. A. and Mavrina, M. A. (2014), “A method for increasing the reliability of verification of data represented in a residue number system”, *Cybernetics and Systems Analysis*, vol. 50, Issue 6, pp. 969-976, available at: https://link.springer.com/article/10.1007/s10559-014-9688-3.
9. Bravo-Montes, J.A., Martín-Toledano, A. and Sánchez-Macián, A. (2022), “Design and implementation of efficient QCA full-adders using fault-tolerant majority gates”, *The Journal of Supercomputing* 78, pp. 8056–8080, doi: https://doi.org/10.1007/s11227-021-04247-9
10. Moroz, S. A. and Krasnobayev, V. A. (2011), “A data verification method in a non-positional residue number system”, *Control, Navigation, and Communication Systems* No. 2(18), pp. 134–138.
11. Ahmadpour, S.-S., Mosleh, M. and Heikalabad, S.R. (2020), “An efficient fault-tolerant arithmetic logic unit using a novel fault-tolerant 5-input majority gate in quantum-dot cellular automata”, *Computers & Electrical Engineering*, Vol. 82, 106548, doi: https://doi.org/10.1016/j.compeleceng.2020.106548.
12. Krasnobayev, V. A., Kuznetsov, A. A., Koshman, S. A. and Kuznetsova, K. O. (2020), “A method for implementing the operation of modulo addition of the residues of two numbers in the residue number system”, *Cybernetics and Systems Analysis*, Vol. 56, No. 6, pp. 1029-1038, doi: https://doi.org/10.1007/s10559-020-00323-9.
13. Valvano, J. (2017), “Embedded Systems: Real-Time Operating Systems for Arm Cortex M Microcontrollers”, *CreateSpace Independent Publishing Platform*, 486 p., available at: https://faculty.uobasrah.edu.iq/uploads/1631127575.pdf.

14. Shahid, M.A. (2021), “Towards resilient method: an exhaustive survey of fault tolerance methods in the cloud computing environment”, *Computer Science Review*, Vol. 40, 100398, doi: https://doi.org/10.1016/j.cosrev.2021.100398.
15. Zhang, Y., Mandal, A., Koelbel, C. and Cooper, K. (2009), “Combined fault tolerance and scheduling techniques for workflow applications on computational grids”, *Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*. CCGRID’09, pp 244–251, doi: https://doi.org/10.1109/CCGRID.2009.59.
16. Pascucci, F. (2023), “Fault Tolerance in Cyber-Physical Systems”, Jajodia, S., Samarati, P., Yung, M. (eds) *Encyclopedia of Cryptography, Security and Privacy*. Springer, Berlin, Heidelberg, doi: https://doi.org/10.1007/978-3-642-27739-9_1724-2.
17. José Rodrigo, Azambuja, Fernanda, Kastensmidt and Jürgen, Becker, (2014), *Hybrid Fault Tolerance Techniques to Detect Transient Faults in Embedded Processors*. Springer Cham, doi: https://doi.org/10.1007/978-3-319-06340-9.
18. Ozturk, Z., Topcuoglu, H.R. & Kandemir, M.T. (2022), “Studying error propagation on application data structure and hardware”, *The Journal of Supercomputing*, 78, pp. 18691–18724, doi: https://doi.org/10.1007/s11227-022-04625-x.
19. Jia, J., Liu, Y. and Zhang, G. (2023), “Software approaches for resilience of high performance computing systems: a survey”, *Frontiers of Computer Science*. Sci. 17, 174105, doi: https://doi.org/10.1007/s11704-022-2096-3.
20. Amiri, Z., Heidari, A. and Navimipour, N.J. (2022), “Resilient and dependability management in distributed environments: a systematic and comprehensive literature review”, *Cluster Computing*, doi: https://doi.org/10.1007/s10586-022-03738-5.
21. Thakral, S. and Bansal, D. (2020), “Novel high functionality fault tolerant ALU”, *TELKOMNIKA (Telecommunication, Computing, Electronics and Control) Formal Verification of a Fully IEEE Compliant Floating-Point Unit, Christian Jacobi*, Vol. 18, pp. 234-239, doi: https://doi.org/10.12928/TELKOMNIKA.v18i1.12645.

Received (Надійшла) 15.03.2023
Accepted for publication (Прийнята до друку) 17.05.2023

Кошман Сергій Олександрович – доктор технічних наук, доцент, професор кафедри безпеки інформаційних систем і технологій, Харківський національний університет імені В. Н. Каразіна, Харків, Україна;
Serhii Koshman – Doctor of Technical Sciences, Assistant Professor, Assistant Professor of Information Systems and Technologies Security Department, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine;
e-mail: s.koshman@karazin.ua; ORCID ID: http://orcid.org/0000-0001-8934-2274.

Краснобаєв Віктор Анатолійович – доктор технічних наук, професор, професор кафедри електроніки і управляючих систем, Харківський національний університет імені В. Н. Каразіна, Харків, Україна;
Victor Krasnobayev – Doctor of Technical Sciences, Professor, Professor of Electronics and Control Systems Department, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine;
e-mail: v.a.krasnobaev@karazin.ua; ORCID ID: http://orcid.org/0000-0001-5192-9918.

Нікольський Сергій Борисович – кандидат технічних наук, доцент, доцент кафедри інформатики Комунального закладу "Харківська гуманітарно-педагогічна академія" Харківської обласної ради, Харків, Україна;
Serhii Nikolsky – Candidate of Technical Sciences, Associate Professor, Associate Professor of Informatics Department, Municipal establishment «Kharkiv humanitarian-pedagogical academy» of Kharkiv regional council, Kharkiv, Ukraine;
e-mail: s.nikoloss@gmail.com; ORCID ID: http://orcid.org/000-0002-3279-6459.

Ковальчук Дмитро Миколайович – аспірант, Харківський національний університет ім. В. Н. Каразіна, Харків, Україна;
Dmytro Kovalchuk– PhD student, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine
e-mail: kovalchuk.d.n@ukr.net; ORCID ID: https://orcid.org/0000-0002-8229-836X.

Структура комп'ютерної системи у залишкових класах

С. О. Кошман, В. А. Краснобаєв, С. Б. Нікольський, Д. М. Ковальчук

Анотація. **Предметом** статті є постановка та вирішення зворотної задачі оптимального резервування в системі залишкових класів (СЗК) на основі використання методу динамічного програмування. Вирішення поставленого завдання дає можливість підвищити надійність роботи комп'ютерних систем та компонентів (КСК) у СЗК. **Метою** статті є підвищення надійності функціонування КСК, які будуються на базі використання СЗК, без зниження швидкості обчислень, а також проведення розрахунку та порівняльного аналізу надійності, за ймовірністю безвідмовної роботи, КСК у СЗК та тройованої обчислювальної системи, яка функціонує у позиційній двійковій системі числення (ПСЧ). **Завдання**: провести аналіз впливу системи числення, що використовується, на надійність КСК з урахуванням первинної та вторинної надмірності; синтезувати обчислювальну систему в СЗК для l-байтової бітової сітки на основі використання методу пасивної відмовостійкості (постійної структурної надмірності); сформулювати та вирішити обернену задачу оптимального резервування в СЗК на основі використання методу динамічного програмування; для перевірки коректності отриманих результатів провести розрахунок умовної кількості обладнання обчислювальної системи у залишкових класах; оцінити ефективність використання СЗК для підвищення надійності при побудові резервованої КСК по відношенню до резервованої КСК у ПСЧ. **Методи дослідження**: методи аналізу та синтезу комп'ютерних систем, теорія чисел, теорія кодування в СЗК, теорія надійності. Отримано такі **результати**. У роботі показано, що використання ПСЧ, як системи числення, не дозволяє радикально підвищити продуктивність і надійність КСК. У зв'язку з цим у статті розроблено концепцію використання СЗК, як систему числення для побудови КСК. На підставі цього сформульована та вирішена зворотна задача оптимального резервування в СЗК. **Висновки**. Як показали результати розрахунків та порівняльного аналізу використання СЗК забезпечує більш високу надійність КСК, ніж мажоритарна триканальна обчислювальна система в ПСЧ. Отримані результати досліджень можуть бути використані для синтезу відмовостійких комп'ютерних структур у СЗК.

Ключові слова: непозиційна кодова структура; система залишкових класів; позиційна двійкова система числення; надійність комп'ютерних систем та компонентів.