Research Article

Victor Krasnobayev, Alexandr Kuznetsov*, and Anastasiia Kiian

Designing of fault-tolerant computer system structures using residue number systems

https://doi.org/10.1515/comp-2020-0171
received May 4, 2020; accepted February 2, 2022

Abstract: This article discusses computing systems that operate in residue number systems (RNSs). The main direction of improving computer systems (CSs) is increasing the speed of implementation of arithmetic operations and the reliability of their functioning. Encoding data in RNS solves the problem of optimal redundancy, i.e., the creation of such computing systems provides maximum reliability with restrictions on weight and size characteristics. This article proposes new structures of fault-tolerant CSs operating in RNS in the case of the application with an active fault-tolerant method. The use of the active fault-tolerant method (dynamic redundancy) in the RNSs provides higher reliability. In addition, with an increase in the digits of CSs, the efficiency of using the proposed structures increases.

Keywords: fault-tolerant computer system, dynamic redundancy, optimal redundancy, residue number system

1 Introduction

Currently, the main directions of computer systems and components (CSC) development in a positional number system (PNS) are improving the speed of implementation of arithmetic operations (including integer arithmetic operations) and reliability of their functioning [1–3].

*** Corresponding author: Alexandr Kuznetsov,** Department of Information Systems and Technologies Security, V. N. Karazin Kharkiv National University, 4 Svobody Sq., Kharkiv, 61022, Ukraine; JSC "Institute of Information Technologies," Bakulin St., 12, Kharkiv, 61166, Ukraine, e-mail: kuznetsov@karazin.ua
Victor Krasnobayev: Department of Information Systems and Technologies Security, V. N. Karazin Kharkiv National University, 4 Svobody Sq., Kharkiv, 61022, Ukraine, e-mail: v.a.krasnobaev@gmail.com
Anastasiia Kiian: JSC "Institute of Information Technologies," Bakulin St., 12, Kharkiv, 61166, Ukraine; Department of Information Systems and Technologies Security, V. N. Karazin Kharkiv National University, 6 Svobody Sq., Kharkiv, 61022, Ukraine, e-mail: nastyak931@gmail.com

The complexity, scale and volume of various computational tasks and tasks of managing large and complex information systems which are solved by computer system (CS) necessitate the expansion of functions and capabilities of the CS. It leads to an increase in the number and complexity of computing equipment and system equipment, and in the complexity of their software [4,5]. This, in turn, necessitates the adoption of additional measures to ensure the high reliability of the CS and, first of all, the need to take additional measures to ensure the high fault tolerance of the CSC [6,7].

2 State-of-the-art

There are two main principles of increasing the reliability of the CS that operates in the PNS [8–10]: increasing reliability of individual elements of the CS (using a new base element) and introducing various types of redundancies. Since the reliability of logical elements is mainly determined by the current level of technology development, it is obvious that the introduction of redundancy when using any available element base is the most effective way to increase the reliability of the computer-controlled system [11,12].

The listed measures for introducing redundancy ensure the availability of fault-tolerant properties in CS. The presence of fault-tolerance properties in the CS allows us to increase the reliability of its functioning. The fault-tolerant property [7] provides the ability to perform specified computational functions after failures, both by reducing, within acceptable limits, any indicators of the quality of functioning (e.g., by gradual degradation), and without deteriorating the quality of functioning of the computer-controlled system [2,13,14].

The following is shown in the literature on this subject [2,15,16]. First, the use of binary PNS as the number system of the CS does not allow us to dramatically increase productivity and reliability. Second, there are results of basic research, which show possibility of a significant increase in the speed of implementation of integer arithmetic operations: addition, multiplication and subtraction by using a non-positional residue number system (RNS). This is achieved through the use of the properties of numbering systems, namely, independence, equal rights and low rate of transfer of residuals, which determine the non-positional code structure (NCS). The first and second properties of RNSs allow us to parallelize arithmetic calculations at the level of decomposition of residue numbers. Also, they provide an opportunity to realize spatial diversity of data elements with the possibility of their subsequent asynchronous independent processing. The use of the third properties of residue classes allow us to implement tabular methods for performing integer arithmetic operations of the basic set and implement polynomial functions with a single-cycle sample of the result of a modular operation. Thus, the use of RNS as number system of CS significantly (compared with PNS) increases the speed of the CSC.

Suppose that at the design stage it is necessary to provide the needed (predetermined) level of fault-tolerance of a CS. It is possible to increase (ensure) the reliability if the system has a certain property, the use of which will allow it to be done [6,17]. Such property is defined and called fault-tolerance [7,18,19]. The presence of the fault tolerance property of the CS can be ensured both through the application of the method of passive fault tolerance (using the method of constant structural redundancy) [6,20] and through the method of active fault tolerance [21] (dynamic backup method) by introducing, in addition to the main computing paths CS in RNS, backup computing paths.

In refs [14,22], it was noted that the use of passive fault-tolerant methods provides increased reliability of the CS in RNS in comparison with the majority turned computing system widely used in the PNSs. However, the lack of research results on the possibility of using residue classes to increase the reliability of the CS based on the application of active fault-tolerant methods limits the possibility of a comprehensive consideration of the task of ensuring the reliability of the functioning of the CSC. The purpose of the article is to design structures of a fault-tolerant CS in RNS in the case of applying the active fault-tolerant method.

3 Statement of the optimal redundancy problem

Based on the properties of the RNS, the independence, equality and low number of digits of the residues representing the number of the structure of the CS in RNS are based on the use of the idea of structural redundancy in the PNS. In this case, the computing paths, according to the corresponding modules of the RNS, play the role of the main elements of the redundant system, and the control computing paths of the RNS play the role of the backup elements of the redundant system. Theoretically, the increase in the reliability of the CS in the RNS is explained as follows. First, based on the properties of independence and equality of the remainders, the totality of which determines the NCS, the structure of the CS in the RNS is a kind of a structural redundant system, consisting of computing paths according to the corresponding RNS modules. Each of the elements (computing paths) of this redundant system (CS) operates independently of each other and in parallel in time and each in its own separate module. This in the PNS is equivalent to the fact that smaller nodes of the CS are reserved. Second, in accordance with the provisions of the general reliability theory (from the point of view of increasing reliability), it is advisable to reserve small nodes and blocks of a complex system, since the failure rate of elements is always less than the failure rate of large nodes or the entire large system as a whole. The probability of fault-tolerant operation of the CS in the RNS is higher than the probability of fault-tolerant operation of the methods of increasing the reliability of duplicated and tripled majority computer structures widely used in practice. For the active fault-tolerant methods, the task of designing a fault-tolerant CS operating in the RNS is verbally formulated as follows: it is necessary to determine the optimal set $\{m_j\}_{\mathrm{OPT}}$ of modules from possible control $m_{n+1}, m_{n+2}, \ldots, m_{n+k}$, at which reliability (fault tolerance) $P_{\mathrm{RC}}^{(n/k)}(t)$ CS in the RNS will be maximum. However, there are limits on the amount of equipment $V_{\mathrm{RC}}^{(n+k)}$ CS in the RNS.

To get the optimal set $\{m_j\}_{\mathrm{OPT}}$ of bases, it is necessary to formulate and solve the inverse problem of optimal reservation in RNS. If necessary, for the application of the active fault-tolerant method (application of the dynamic structural redundancy method), the inverse optimal reservation problem in RNS is mathematically formulated as follows (equation (1)):

$$\begin{cases} P_{\mathrm{RC}}^{(n+z)}(t)[t = \mathrm{const}] \to \max; \\ \quad (V_{\mathrm{PNS}}^{(l)} \geq V_{\mathrm{RC}}^{(n+k)}), \end{cases} \tag{1}$$

where k is a maximum possible number of control bases $m_{n+1}, m_{n+2}, \ldots, m_{n+k}$ (the number of CS control computing paths in residue classes); $\{m_{(n+z)}\}_{\mathrm{OPT}}$ $(z = 1 \ldots k)$ is an optimal (providing a minimum amount of equipment $V_{\mathrm{RC}}^{(n+k)}$ CS in RNS) set of control bases (for a given value l of the CS character grid), the simultaneous use of which

Table 1: The sets of information n and control k bases of the residue classes, 2PNS

$l(n)$	$\{m_i\}$, $i = 1 \dots n$	$V_{RC}^{(n)}$	$V_{2PNS}^{(l)}$	ΔV	$\{m_j\}$, $j = n+1 \dots n+k$	$V_{RC}^{(k)}$	k
1(4)	$m_1 = 3, m_2 = 4,$ $m_3 = 5, m_4 = 7$	10	16	6	$m_5 = 11$	4	1
2(6)	$m_1 = 2, m_2 = 5,$ $m_3 = 7, m_4 = 9,$ $m_5 = 11, m_6 = 13$	19	32	13	$m_7 = 17, m_8 = 19$	10	2
3(8)	$m_1 = 3, m_2 = 4,$ $m_3 = 3, m_4 = 7,$ $m_5 = 11, m_6 = 13,$ $m_7 = 17, m_8 = 19$	28	48	20	$m_9 = 23, m_{10} = 29,$ $m_{11} = 31, m_{12} = 37$	21	4
4(10)	$m_1 = 2, m_2 = 3,$ $m_3 = 5, m_4 = 7,$ $m_5 = 11, m_6 = 13,$ $m_7 = 17, m_8 = 19,$ $m_9 = 23, m_{10} = 29$	37	64	27	$m_{11} = 31, m_{12} = 37,$ $m_{13} = 41, m_{14} = 43$	23	4

ensures the maximum fault-tolerance value $P_{RC}^{(n/z)}(t)$ CS in residue classes (RC).

In this case, the following condition must be met:

$$V_{RC}^{(k)} \leq V_{PNS}^{(l)} - V_{RC}^{(n)}. \quad (2)$$

It is convenient to solve the inverse optimal reservation problem in RNS with restriction by the method of coordinate wise steepest descent or by the method of dynamic programming [23].

4 Structures of CSs functioning in RNS

Let us consider examples of constructing CS structures in RNS for $l = 1 \dots 4$. Tables 1 and 2 show the sets of information n and control k bases of the residue classes for the values of l of the corresponding characters grids $l = 1 \dots 4$ of the CS. The data on the conditional relative amount of equipment of the computer station in RNS are presented, as well as the data on the conditional relative quantity of the equipment of CS in the PNS, reduced to one binary digit of the length l of the grid, where

$$V_{RC}^{(n)} = \sum_{i=1}^{n} \alpha_i,$$

is a conditional relative amount of the equipment of the CS, reduced to one binary digit without excess CS in the RNS.

$$V_{RC}^{(k)} = \sum_{i=n+1}^{n+k} \alpha_i,$$

is an additional conditional relative amount of CS equipment, reduced to one binary digit of CS in RNS, formed by

Table 2: The sets of information n and control k bases of the residue classes, 3PNS

$l(n)$	$\{m_i\}$, $i = 1 \dots n$	$V_{RC}^{(n)}$	$V_{2PNS}^{(l)}$	ΔV	$\{m_j\}$, $j = n+1 \dots n+k$	$V_{RC}^{(k)}$	k
1(4)	$m_1 = 3, m_2 = 4,$ $m_3 = 5, m_4 = 7$	10	24	14	$m_5 = 11, m_6 = 13,$ $m_7 = 17$	13	3
2(6)	$m_1 = 2, m_2 = 5,$ $m_3 = 7, m_4 = 9,$ $m_5 = 11, m_6 = 13$	19	48	29	$m_7 = 17, m_8 = 19,$ $m_9 = 23, m_{10} = 29,$ $m_{11} = 31$	25	5
3(8)	$m_1 = 3, m_2 = 4,$ $m_3 = 3, m_4 = 7,$ $m_5 = 11, m_6 = 13,$ $m_7 = 17, m_8 = 19$	28	72	44	$m_9 = 23, m_{10} = 29,$ $m_{11} = 31, m_{12} = 37,$ $m_{13} = 41, m_{14} = 43,$ $m_{15} = 47, m_{16} = 53$	45	8
4(10)	$m_1 = 2, m_2 = 3,$ $m_3 = 5, m_4 = 7,$ $m_5 = 11, m_6 = 13,$ $m_7 = 17, m_8 = 19,$ $m_9 = 23, m_{10} = 29$	37	96	59	$m_{11} = 31, m_{12} = 37,$ $m_{13} = 41, m_{14} = 43,$ $m_{15} = 47, m_{16} = 53,$ $m_{17} = 59, m_{18} = 61,$ $m_{19} = 67$	54	9

the introduction of control bases. $\alpha_i = [\log_2(m_i - 1)] + 1$ is a number of bits needed to represent the m_i of the residue classes. $\lfloor X \rfloor$ is an integer part of the number X that does not exceed it (rounding down). For example, $[7.9] = 7$. $V_{0\text{PNS}}^{(l)}$ is a conditional relative amount of equipment without CS excess in the PNS. $V_{2\text{PNS}}^{(l)}$ is a conditional relative amount of equipment of duplicated CS in the PNS. $V_{3\text{PNS}}^{(l)}$ is a conditional relative amount of equipment of triple CS in the PNS.

The set $\{m_j\}_{\text{OPT}}$ from k control bases of residue classes, determined from condition (3):

$$\Delta V \geq V_{\text{RC}}^{(k)}. \quad (3)$$

In conventional units, the number of CS equipment $V_{\text{PNS}}^{(n+k)}$ in the RNC is determined by the following equation:

$$V_{\text{PNS}}^{(n+k)} = V_{\text{PNS}}^{(n)} + V_{\text{PNS}}^{(k)},$$

where $V_{\text{PNS}}^{(n)} = \sum_{i=1}^{n} \alpha_i$ is a conditional relative number of CS equipment reduced to one binary digit of CS in the RNS without redundancy (the number of binary bits in the representation of all RNS information modules).

$V_{\text{RNS}}^{(k)} = \sum_{i=n+1}^{n+k} \alpha_i$ is an additional conditional relative quantity of the CS equipment, reduced to one bit of the digits of the redundant CS in the RNS, generated by introducing k control bases (the number of binary bits in the representation of all RNS control modules);

$\alpha_i = [\log_2(m_i - 1)] + 1$ is a number of bits required to represent the base by modulo m_i RNS. The values n, k denote the number of information and control bases of the RNS, respectively. Set of k control bases of the RNS, is determined from the following condition:

$$\Delta V \geq V_{\text{RNS}}^{(k)},$$

where

$$\Delta V = V_{2\text{PNS}}^{(l)} - V_{\text{RNS}}^{(n)} \; (\Delta V = V_{3\text{PNS}}^{(l)} - V_{\text{RNS}}^{(n)}),$$

or from the following condition:

$$V_{\text{PNS}}^{(l)} \geq V_{\text{RNS}}^{(n+k)}.$$

The conditional relative amount of equipment for duplicated and tripled l - byte ($l = 1 \ldots 4$) CS in the PNS is determined by the expression $V_{2\text{PNS}}^{(l)} = 2 \cdot 8 \cdot l$ and $V_{3\text{PNS}}^{(l)} = 3 \cdot 8 \cdot l$, respectively.

When conducting a comparative analysis of reliability for CS in the PNS, we will take duplicated and triple CS of the same type of CS. In this case, the probability of fault-tolerant operation for such CS will be determined, respectively, by the following well-known expressions:

$$P_{2\text{PNS}}^{(l)}(t) = 1 - [1 - P_{0\text{PNS}}^{(l)}(t)]^2, \quad (4)$$

$$P_{3\text{PNS}}^{(l)}(t) = 1 - [1 - P_{0\text{PNS}}^{(l)}(t)]^3, \quad (5)$$

where

$$P_{0\text{PNS}}^{(l)}(t) = \mathrm{e}^{-8l\lambda_E t},$$

is a probability of fault-tolerant operation of redundant l- byte positional CS; λ_E is a failure rate of a unit of the conditional equipment of the CS, assigned to one binary digit of the characters grid of the CS.

The amount of equipment for a duplicate CS in the PNS is determined by the following expression:

$$V_{2\text{PNS}}^{(l)} = 2 \cdot 8 \cdot l,$$

and for a triple CS is determined as

$$V_{3\text{PNS}}^{(l)} = 3 \cdot 8 \cdot l.$$

Based on the data of Tables 1 and 2 and using expressions (6)

$$P_{\text{RC}}^{(n/k)}(t) = \sum_{i=0}^{k} C_{n+k}^{i} P_i^{n+k-i}(t) \sum_{j=0}^{i} (-1)^i C_i^j P_i^j(t), \quad (6)$$

we obtain expressions for determining the probability of fault-tolerance operation of the CS in the RNS for the values $l = 1 \ldots 4$ of byte digit (Table 3).

$$C_{n+k}^{i} = \frac{(n+k)!}{i! \cdot (n+k-i)!}$$

is a number of combinations of $n + k$ elements by i.

Formulas similar to expression (6) are used to determine the probability of fault-tolerant operation of a CS in RNS in the case of a sliding redundancy mode with a loaded reserve with n information and k control computing paths. In this case, the information and control paths of the CS in RNS are considered equally reliable. For example, in a study of Polovko [24] there is a formula 7.5 (p. 319), as well as a similar formula in another study of Polovko [25].

It should be noted that in accordance with the data of Tables 1 and 2, Tables A1 and A2 (Appendix) presents the dynamic ranges (the volume of the range of possible numbers) of the set of information D_1 and control D_2 RNS modules.

Based on the data in Tables 1 and 2, the article synthesizes structures of fault-tolerance CS in RNS for $l = 1 \ldots 4$ byte characters grid (Figures 1–4).

Figure 1 presents the CS structure for a single-byte ($l = 1$) digit (8 binary digits). The structure of the CS contains four ($n = 4$) information computational paths and three ($k = 3$) control computational paths. In case of failure of the information computational paths, control computational paths are connected in its place.

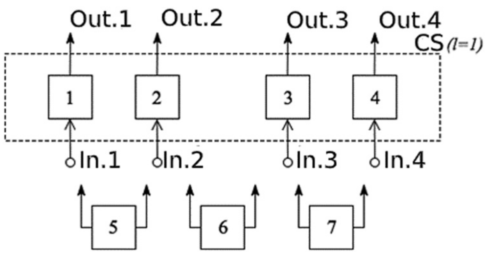

Figure 1: Block diagram of CS in RNS for value l = 1.

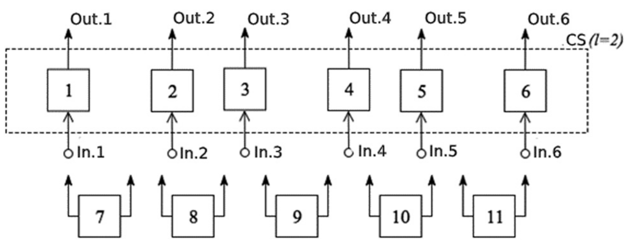

Figure 2: Block diagram of CS in RNS for value l = 2.

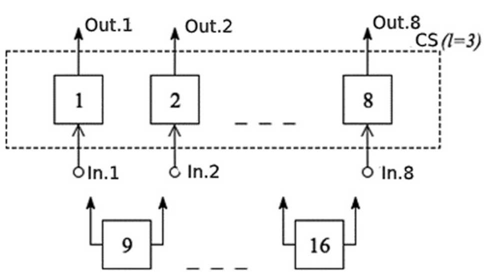

Figure 3: Block diagram of CS in RNS for value l = 3.

Tables 1 and 2 present sets of specific information and control modules of the RNS. Figure 2 presents the CS structure for a two-byte (l = 2) digit (16 binary digits). The structure of the CS contains 6 (n = 6) information computational paths and 5 (k = 5) control computational paths. In case of failure of the information computational paths, control computational paths are connected in its place. Tables 1 and 2 present sets of specific information and control modules of the RNS. Figure 3 presents the CS structure for a three-byte (l = 3) digit (24 binary bits). The structure of the CS contains eight (n = 8) information computational paths and eight (k = 8) control computational paths. In case of failure of the information computational paths, control computational paths are connected in its place. Tables 1 and 2 present sets of specific information and control modules of the RNS. Figure 4 presents the CS structure for a four-byte (l = 4) character grid (32 binary bits). The structure of the CS contains eight (n = 10) information computational paths and eight (k = 9) control computational paths. In case of failure of the information computational paths, control computational paths are connected in its place. Tables 1 and 2 present sets of specific information and control modules of the RNS.

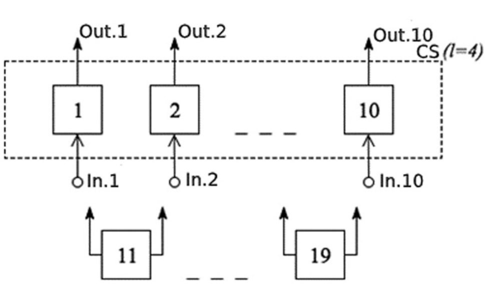

Figure 4: Block diagram of CS in RNS for value l = 4.

The structure of the CS in the RNS contains n information modules and k control modules. In Figures 1–4, a simplified structure of CSs in RNS for a one-byte (l = 1) character grid (8 binary digits), two-byte (l = 2) characters grid (16 binary digits), three-byte (l = 3) bit grid (24 binary bits) and a four-byte (l = 4) character grid (32 bits). Tables 1 and 2 present sets of specific n informational (number of information computing paths) and k control modules of the RNS. In the article, the CS in the RNS is considered from the standpoint of the backup system in the PNS operating in the sliding redundancy mode with a loaded reserve. In this case, information and control computing paths play the role of the main and backup elements of the redundant system in the PNS respectively. The following basic assumptions are made for the functioning of the CS in the RNS.

- Failures of computing paths satisfy the conditions of the simplest flow of failures. This is due to the fact that the simplest flow of CS failures was theoretically substantiated, confirmed experimentally and provided with information about the failure rate of the existing CS elements.
- A switching device that performs the operations of determining failed computing paths and connecting in their place ideally working ones, i.e., the probability of failure-free operation of the switch is equal to one. On simplified structural diagrams (Figures 1–4), the CS in the RNS does not show the switching device.
- Since the article deals with an ordered ($m_i < m_{i+1}$) RNS, an arbitrary operable control computing path can be connected instead of any failed information computing path.
- Recovery of the failed computing paths of the CS in the RNS is not provided.

Figures 1–4 show the simplified structural diagrams of the CS in the RNS. In the general case, the CS in the

Table 3: Mathematical expressions for calculating the values of the fault-tolerant operation probability of the CS in RNS

In relation to the duplicated CS in the PNS	In relation to the triple CS in the PNS
$P_{\substack{RC\\l=1}}^{(4/1)}(t)=\sum_{i=0}^{1}C_5^iP_i^{5-i}\sum_{j=0}^{i}(-1)^jC_i^jP_i^j(t)$	$P_{\substack{RC\\l=1}}^{(4/3)}(t)=\sum_{i=0}^{3}C_7^iP_i^{7-i}\sum_{j=0}^{i}(-1)^jC_i^jP_i^j(t)$
$P_{\substack{RC\\l=2}}^{(6/2)}(t)=\sum_{i=0}^{2}C_8^iP_i^{8-i}\sum_{j=0}^{i}(-1)^jC_i^jP_i^j(t)$	$P_{\substack{RC\\l=2}}^{(6/5)}(t)=\sum_{i=0}^{5}C_{11}^iP_i^{11-i}\sum_{j=0}^{i}(-1)^jC_i^jP_i^j(t)$
$P_{\substack{RC\\l=3}}^{(8/4)}(t)=\sum_{i=0}^{4}C_{12}^iP_i^{12-i}\sum_{j=0}^{i}(-1)^jC_i^jP_i^j(t)$	$P_{\substack{RC\\l=3}}^{(8/8)}(t)=\sum_{i=0}^{8}C_{16}^iP_i^{16-i}\sum_{j=0}^{i}(-1)^jC_i^jP_i^j(t)$
$P_{\substack{RC\\l=4}}^{(10/4)}(t)=\sum_{i=0}^{4}C_{14}^iP_i^{14-i}\sum_{j=0}^{i}(-1)^jC_i^jP_i^j(t).$	$P_{\substack{RC\\l=4}}^{(10/9)}(t)=\sum_{i=0}^{9}C_{19}^iP_i^{19-i}\sum_{j=0}^{i}(-1)^jC_i^jP_i^j(t).$

RNS consists of $n+k$ computing paths, which operate independently of each other in parallel in time (RNS properties). For a specific case, we will make explanations using the example of Figure 1. Figure 1 presents CS in RNS implements calculations for a single-byte ($l=1$ or 8 binary digits) character grid. The CS operates with numbers in the numerical range from 0 to 2^8-1 or from 0 to 255. Information ($n=4$) computing paths by modules $m_1=3$, $m_2=4$, $m_3=5$ and $m_4=7$ (Tables 1 or 2) ensure the operation of the CS in the RNS in the given 0 to 2^8-1 numerical range, as $2^8<m_1m_2m_3m_4=420$. Figure 1 shows four ($n=4$) information inputs (In. 1 – In. 4) of the CS to the information computational paths of which, through the corresponding RNS modules, the processed numbers are received. At the exits Out. 1-Out. 4 KS we obtain the result of the operation. Reserve computing paths 5–7 modulo $m_5=11$, $m_6=13$ and $m_7=17$ (Tables 1 or 2), respectively, provide an efficient state of the CS in the RNS in case of failure of any information computing path. In this case, the sequence of connecting backup computing paths 5–7 corresponds to the number of the control computing paths, as $m_i<m_{i+1}$. The fulfillment of the condition $m_i<m_{i+1}$ makes it possible to ensure the possibility of the functioning of the CS in a given numerical range from 0 to 2^8-1.

In Figure 1, information and control computational paths are naturally included in the structure of the CS in the RNS. The location of the computing path of the CS in the RNS (Figure 1) only emphasizes the differences between information and control modifications.

In accordance with this path mathematical expressions were first derived for calculating the values of the fault-tolerant operation probability of the CS in the RNSs (Table 3).

Using the probability formulas for the fault-tolerant operation of the CS in the RNS with respect to the triple CS in the PNS (Table 3), as well as expression 6, the

Table 4: The results of the calculations of the reliability of CS in the PNS and the RC

		PNS		RC	
l	$\lambda_E t$	One CS	Triple	Permanent reservation	Dynamic reservation
$l=1$	10^{-3}	0.992032	0.999999	0.999983	0.999999
	10^{-2}	0.923116	0.999546	0.998317	0.999995
	10^{-1}	0.449329	0.833015	0.856338	0.915746
	2×10^{-1}	0.201897	0.491633	0.574597	0.521476
$l=2$	10^{-3}	0.984127	0.999996	0.999967	1.000000
	10^{-2}	0.852144	0.996767	0.996717	0.999999
	10^{-1}	0.201897	0.491632	0.732831	0.937141
	2×10^{-1}	0.040762	0.117369	0.322572	0.471169
$l=3$	10^{-3}	0.9761286	0.999986	0.999961	1.000000
	10^{-2}	0.786628	0.990285	0.996029	0.999999
	10^{-1}	0.90717	0.248211	0.633995	0.917094
	2×10^{-1}	0.008229	0.024486	0.176643	0.261531
$l=4$	10^{-3}	0.968506	0.999968	0.999948	1.000000
	10^{-2}	0.726149	0.979462	0.994618	1.000000
	10^{-1}	0.040762	0.117369	0.529579	0.924419
	2×10^{-1}	0.001661	0.004976	0.089611	0.209170

reliability of the CS in RNS was calculated. Table 4 presents the results of the calculations of the CS reliability in the PNS and CS reliability in the in RNS.

5 Conclusion

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

(1) Based on the application of the active fault-tolerant methods, the article presents the designed structures of the CS in the RNS for $l = 1 \ldots 4$ byte system (Figures 1–4). In accordance with which (on the basis of the initial data, Tables 1 and 2), mathematical expressions were first derived for calculating the values of the fault-tolerant operation probability of the CS in the RNS (Table 3). Calculation and comparative analysis of the reliability of the CS in the RNS and the widely used in the PNSs with triple computing structure, which confirmed the high efficiency of the use of the NCS (Table 4). Studies have shown that, first, the use of the active fault-tolerant method (dynamic reservation) in the in RNS provides higher reliability than that widely used in the PNS method of tripling of the same type of CS, as well as provide higher reliability than the passive fault-tolerant method (continuous redundancy) in the in RNS. Second, with an increase in the CS digits, which is characteristic of the modern tendency in the development of computing facilities, the efficiency of using NCS in RNS increases.

(2) 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 a type of structural redundant system from the computing path according to the corresponding modules m_i of the residue classes. Each of the elements (computing paths) of this redundant system (CS) operates independently of each other and in parallel in time according to its own separate module. This in the PNS is equivalent to reserving the smaller nodes of the CS. Second, in accordance with the provisions of the general theory (from the point of view of increasing reliability), it is advisable to reserve small nodes and blocks of a complex system, since the failure rate of elements is always less than the failure rate of large nodes or the entire large system. The results of a comparative reliability analysis showed that the probability of a fault-tolerant operation of the CS in RNS is higher than the probability of the fault-tolerant operation of the tripled majority structure in the PNS.

(3) There is an assumption that the data encoding in RNS solves the problem of optimal backup, i.e., the creation of CS based on the use of residue classes provides it with maximum reliability under restrictions on its overall dimensions. However, this assumption requires a separate strict proof.

Conflict of interest: Authors state no conflict of interest.

Data availability statement: All data generated or analyzed during this study are included in this published article.

References

[1] J. O. Tuazon, *Residue Number System in Computer Arithmetic*, Ames, Iowa, United States, Iowa State University, 1969, DOI: 10.31274/rtd-180816-2270.

[2] P. V. A. Mohan, *Residue Number Systems*, Bangalore, India, Birkhäuser, 2016. DOI: 10.1007/978-3-319-41385-3.

[3] L. Hanzo, L.-L. Yang, E.-L. Kuan, and K. Yen, "Residue number system arithmetic," in: *Single- and Multi-Carrier DS-CDMA*, L. Hanzo, L.-L. Yang, E.-L. Kuan, and K. Yen, eds., Hoboken, New Jersey, John Wiley & Sons, Ltd, 2005, DOI: 10.1002/0470863110.ch17.

[4] J. D. Dumas, "Parallel and high-performance systems [WWW Document]," *Computer Architecture*, Boca Raton, Florida, CRC Press, 2016, DOI: https://doi.org/10.4324/9781315367118-6.

[5] A. Elahi, "Introduction to computer architecture," in: *Computer Systems: Digital Design, Fundamentals of Computer Architecture and Assembly Language*, A. Elahi, ed., Cham, Springer International Publishing, 2018, pp. 115–136, DOI: 10.1007/978-3-319-66775-1_6.

[6] E. Bauer, R. Adams, and D. Eustace, "Understanding redundancy," in: *Beyond Redundancy*, Hoboken, New Jersey, John Wiley & Sons, Ltd, 2011, pp. 35–58, DOI: 10.1002/9781118104910.ch3.

[7] 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.

[8] D. Radhakrishnan and T. Pyon, "Fault tolerance in RNS: an efficient approach," in: *Proceedings on 1990 IEEE International Conference on Computer Design: VLSI in Computers and Processors*, Cambridge, MA, USA, IEEE, 1990, DOI: 10.1109/iccd.1990.130156.

[9] T. F. Tay and C.-H. Chang, "Fault-tolerant computing in redundant residue number system," in: *Embedded Systems Design with Special Arithmetic and Number Systems*, Cham, Springer, 2017, pp. 65–88, DOI: 10.1007/978-3-319-49742-6_4.

[10] K. Phalakarn and A. Surarerks, "Alternative redundant residue number system construction with redundant residue representations," in: *2018 3rd International Conference on Computer and Communication Systems* (*ICCCS*), Nagoya, Japan: IEEE, 2018, DOI: 10.1109/ccoms.2018.8463305.

[11] N. I. Cherviakov, M. G. Babenko, and M. N. Shabalina, "Development of a secure system for distributed data storage and processing in the clouds based on the concept of active security in RNS," in: *2017 XX IEEE International Conference on Soft Computing and Measurements* (*SCM*). *Presented at the 2017 XX IEEE International Conference on Soft Computing and Measurements* (*SCM*), St. Petersburg, Russia, IEEE, 2017, pp. 558–560, DOI: 10.1109/SCM.2017.7970649.

[12] J. Wang, S. Ma, Z.-G. Yang, and J. Hu, "A systemic performance evaluation method for residue number system," in: *2016 2nd IEEE International Conference on Computer and Communications* (*ICCC*). *Presented at the 2016 2nd IEEE International Conference on Computer and Communications* (*ICCC*), Chengdu, IEEE, 2016, pp. 321–325, DOI: 10.1109/CompComm.2016.7924716.

[13] 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*, Cham, Springer, 2018, pp. 319–328, DOI: 10.1007/978-3-319-97885-7_31.

[14] V. Krasnobaev, S. Koshman, A. Kononchenko, K. Kuznetsova, and T. Kuznetsova, "The formulation and solution of the task of the optimum reservation in the system of residual classes," in: *2019 IEEE International Conference on Advanced Trends in Information Theory* (*ATIT*), Kyiv, Ukraine, IEEE, 2019, DOI: 10.1109/atit49449.2019.9030483.

[15] M. Welschenbach, "Modular arithmetic: calculating with residue classes," *Cryptography in C and C++*, pp. 67–79, 2013, DOI: 10.1007/978-1-4302-5099-9_5.

[16] V. Krasnobayev, A. Kuznetsov, I. Lokotkova, and A. Dyachenko, "The method of single errors correction in the residue class," in: *2019 3rd International Conference on Advanced Information and Communications Technologies* (*AICT*), Lviv, Ukraine, IEEE, 2019, DOI: 10.1109/aiact.2019.8847845.

[17] M. L. Shooman, "N-modular redundancy," in: *Reliability of Computer Systems and Networks*, Hoboken, New Jersey, John Wiley & Sons, Ltd, 2002, pp. 145–201, DOI: 10.1002/047122460X.ch4.

[18] F. P. García Márquez, ed., *Fault Detection, Diagnosis and Prognosis*, London, UK, IntechOpen, 2020, DOI: 10.5772/intechopen.77564.

[19] T. Chen, *Fault Diagnosis and Fault Tolerance*, Chongqing, China, Computer Research Institute, Chongqing University, 1992, DOI: 10.1007/978-3-642-77179-8.

[20] N. Kniazieva and T. Kunup, "Increasing the structural survivability of telecommunication networks," in: *2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science* (*TCSET*). *Presented at the 2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science* (*TCSET*), Lviv, Ukraine, IEEE, 2016, pp. 572–576, DOI: 10.1109/TCSET.2016.7452119.

[21] P. CaȄŹcaval and F. Leon, "Active redundancy allocation in complex systems by using different optimization methods," in: *Computational Collective Intelligence, Lecture Notes in Computer Science*, N. T. Nguyen, R. Chbeir, E. Exposito, P. Aniorté, B. Trawiński, eds., Cham, Springer International Publishing, 2019, pp. 625–637, DOI: 10.1007/978-3-030-28377-3_52.

[22] V. Yatskiv and T. Tsavolyk, "Improvement of data transmission reliability in wireless sensor networks on the basis of residue number system correcting codes using the special module system," In: *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering* (*UKRCON*), Kyiv, UKraine, IEEE, 2017, DOI: 10.1109/ukrcon.2017.8100376.

[23] I. A. Ushakov, ed., *Optimal Resource Allocation*, Hoboken, New Jersey, John Wiley & Sons, 2013, DOI: 10.1002/9781118400715.

[24] A. M. Polovko, *Fundamentals of Reliability Theory*, Moscow, Science, 1964, p. 446, (in Russian).

[25] A. M. Polovko, *Fundamentals of Reliability Theory*, New York, NY, Academic Press, 1968, p. 478.

Appendix

It should be noted that:

- $D_1 = \prod_{i=1}^{n} m_i$ is a dynamic range of information n modules of the RNS;
- $D_2 = \prod_{i=n+1}^{n+k} m_i$ is a dynamic range of control k modules of the RNS.

Table A1: Dynamic ranges of information D_1 and control D_2 modules of the RNS in the case of a duplicated computer system in the PNS

$l = 1$		$l = 2$		$l = 3$		$l = 4$	
D_1	D_2	D_1	D_2	D_1	D_2	D_1	D_2
3 · 4 · 5 · 7	11	2 · 5 · 7 · ·9 · 11 · 13	17 · 19	3 · 4 · 5 · 7 · ·11 · 13 · 17 · 19	23 · 29 · 31 · 37	2 · 3 · 5 · 7 · 11 · ·13 · 17 · 19 · 23 · 29	31 · 37 · 41 · 43

Table A2: Dynamic ranges of information D_1 and control D_2 modules of the RNS in the case of a tripled computer system in the PNS

$l = 1$		$l = 2$		$l = 3$		$l = 4$	
D_1	D_2	D_1	D_2	D_1	D_2	D_1	D_2
3 · 4 · 5 · 7	11 · 13 · 17	2 · 5 · 7 · ·9 · 11 · 13	17 · 19 · 23 · ·29 · 31	3 · 4 · 5 · 7 · ·11 · 13 · ·17 · 19	23 · 29 · 31 · ·37 · 41 · ·43 · 47 · 53	2 · 3 · 5 · 7 · ·11 · 13 · 17 · ·19 · 23 · 29	31 · 37 · 41 · ·43 · 47 · ·53 · 59 · 61 · ·67