

UDC 004.75

DOI <https://doi.org/10.32689/maup.it.2025.3.10>

**Dmytro KOVALCHUK**

PhD in Computer Science, Senior Lecturer, V. N. Karazin Kharkiv National University,

kovalchuk.d.n@ukr.net

ORCID: 0000-0002-8229-836X

## THE CONCEPT OF BUILDING HIGH-PERFORMANCE REAL-TIME SYSTEMS USING THE RESIDUE NUMBER SYSTEM

**Abstract.** The article examines the concept of building high-performance real-time information processing systems using the Residue Number System (RNS).

**The aim of the research** is to improve the performance and fault tolerance of modern computing systems through the application of a non-positional number system, which enables parallel data processing, dynamic error correction, and adaptive regulation of accuracy and computational speed. The work emphasizes the key properties of RNS – independence of residues, equality of residues, and small digit length – and their impact on the performance and reliability of real-time systems.

**The research methodology** is based on the principles of systems analysis, number theory, computational process theory, and systems modeling, as well as the simulation of modular arithmetic operations and error correction mechanisms. The study analyzes mathematical models of distributing informational and control residues to optimize the balance between speed, accuracy, and reliability of computations. The implementation methods of modular arithmetic are considered, including adder-based, table-based, logical, and circular-shift principles, which improve processing speed and allow single-cycle execution of computations in real time.

**The scientific novelty of the work** lies in the proposed methodology of dynamic redistribution of informational and control residues in RNS to ensure high fault tolerance and adaptability of the system. The study proposes the use of control residues to maintain operability even in the case of multiple computational path failures, as well as the application of small-digit modular arithmetic to enhance speed and reduce hardware complexity. It is shown that such a structure enables the simultaneous realization of three types of redundancy – structural, informational, and functional – which is critical for real-time systems.

The conclusions of the study demonstrate that systems based on RNS provide significant acceleration of computations through parallel execution on independent computational paths and operand decomposition, increase reliability due to error localization, and enable dynamic regulation of accuracy and computational speed. The implementation of such systems makes them effective for processing large data sets, digital signal and image processing, cryptography, neurocomputing, and streaming computation tasks, ensuring continuous operation even in the case of partial component failures.

**Key words:** Residue Number System, high-performance computing, real-time systems, fault tolerance, modular arithmetic, parallel computing.

## Дмитро КОВАЛЬЧУК. КОНЦЕПЦІЯ ПОБУДОВИ ВИСОКОПРОДУКТИВНИХ СИСТЕМ РЕАЛЬНОГО ЧАСУ З ВИКОРИСТАННЯМ СИСТЕМИ ЗАЛИШКОВИХ КЛАСІВ

**Анотація.** У статті розглянуто концепцію побудови високопродуктивних систем обробки інформації в режимі реального часу з використанням системи залишкових класів (СЗК).

**Метою роботи** є підвищення продуктивності та відмовостійкості сучасних обчислювальних систем шляхом застосування непозиційної системи числення, яка дозволяє реалізувати паралельну обробку даних, динамічну корекцію помилок та адаптивне регулювання точності та швидкодії обчислень. У роботі зроблено акцент на властивостях СЗК – незалежності залишків, рівноправності та малорозрядності – та їхньому впливі на продуктивність і надійність систем реального часу.

**Методологія дослідження** базується на принципах системного аналізу, теорії чисел, теорії обчислювальних процесів і систем, а також на моделюванні модульних арифметичних операцій та механізмів корекції помилок. У роботі аналізуються математичні моделі розподілу інформаційних та контрольних залишків для оптимізації співвідношення між швидкістю, точністю та надійністю обчислень. Розглядаються способи реалізації модульної арифметики за допомогою суматорного, табличного, логічного та кільцевого принципів, що дозволяє підвищити швидкість та забезпечити однотактне виконання обчислень у реальному часі.

**Наукова новизна роботи** полягає у запропонованій методології динамічного перерозподілу інформаційних і контрольних залишків у СЗК для забезпечення високої відмовостійкості та адаптивності системи. Запропоновано використання контрольних залишків для підтримки працездатності навіть при відмовах декількох обчислювальних трактів, а також застосування малорозрядної модульної арифметики для підвищення швидкодії та зниження апаратної складності. Показано, що така структура дозволяє реалізувати одночасно три типи резервування: структурне, інформаційне та функціональне, що критично для систем реального часу.

Висновки дослідження демонструють, що системи на основі СЗК забезпечують значне прискорення обчислень за рахунок паралельного виконання на незалежних обчислювальних трактах та декомпозиції операндів, підвищують надійність завдяки локалізації помилок і забезпечують можливість динамічного регулювання точності та швидкодії обчислень. Реалізація таких систем робить їх ефективними для обробки великих масивів даних, цифрової обробки сигналів і зображень, криптографії, нейрокомп'ютерної обробки та задач поточкових обчислень, гарантуючи безперервну роботу навіть у разі часткових відмов компонентів.

**Ключові слова:** система залишкових класів, високопродуктивні обчислення, системи реального часу, відмовостійкість, модульна арифметика, паралельні обчислення.

© D. Kovalchuk, 2025

Стаття поширюється на умовах ліцензії CC BY 4.0

**Introduction.** The rapid development of digital technologies and the growing volumes of data to be processed impose new requirements on the architecture and performance of computing systems. In the period of 2020–2025, a key trend in computing has been the integration of artificial intelligence methods, Internet of Things (IoT) systems, and real-time stream data processing. Such systems must ensure high performance, scalability, and fault tolerance, since delays or errors in information processing can lead to critical consequences – from failures in industrial complexes to risks for the security of cyber-physical systems [10].

Traditional approaches to performance improvement rely on parallel and pipeline architectures, multicore processors, and algorithm optimization. However, they do not always allow effective problem-solving under the constraints of real time and increased reliability requirements [3]. This issue becomes particularly acute in areas where failure or error is unacceptable: in transportation systems, energy, defense, financial technologies, and healthcare [4].

One of the promising directions for building high-performance systems is the use of the Residue Number System (RNS) as the basis for arithmetic computations. The non-positional nature of RNS provides natural parallelism in executing operations, reduces the risk of error accumulation, and enables the development of fault-tolerant algorithms for digital information processing. In addition, the application of RNS makes it possible to optimize addition, multiplication, and comparison operations, which is especially important in real-time applications.

The relevance of the study lies in the need to develop new conceptual approaches to building high-performance real-time computing systems that combine speed, scalability, and fault tolerance. The use of RNS in this context opens opportunities for creating next-generation architectures oriented toward working with large streams of digital data and mission-critical computations.

The purpose of the article is to substantiate and develop the concept of building high-performance real-time systems using the Residue Number System, which will improve the efficiency of digital information processing and ensure a specified level of reliability and fault tolerance under modern challenges [1; 6; 7; 8].

**Problem statement of the research.** Modern information technologies are characterized by the rapid growth of data volumes that must be processed in real time. This concerns a wide range of domains – from industrial automation systems and traffic management to high-frequency financial transactions, cybersecurity, medical monitoring systems, and intelligent IoT devices. Under such conditions, the primary requirements for computing systems become high performance, scalability, and fault tolerance [1; 5].

Traditional methods of improving performance, such as expanding hardware resources, using multicore processors, pipeline architectures, or distributed computing, have largely exhausted their potential. Moreover, their application is often complicated by high energy consumption, increased hardware costs, and synchronization constraints of parallel data streams. For real-time systems, where even millisecond delays can have critical consequences, these drawbacks become decisive [2; 4; 5].

Another significant challenge is ensuring the reliability of computations. In cases where data processing is performed with minimal latency, even a minor error or failure can cause the entire system to malfunction. Traditional approaches to error control (such as computation duplication, error-correcting codes, or hardware redundancy) increase the complexity of the architecture and resource consumption, which negatively affects efficiency [2; 3].

In this context, the RNS offers an alternative approach to organizing arithmetic computations. Due to its non-positional nature, RNS enables natural parallelism: operations on numbers in different moduli are performed independently, which significantly increases performance. At the same time, this approach prevents intermediate result overflow and reduces the likelihood of error accumulation [4].

Accordingly, the research problem lies in creating a conceptual model and architecture of high-performance real-time systems based on RNS that will provide:

- increased speed of digital information processing through natural parallelism [4];
- guaranteed reliability and fault tolerance of computations;
- scalability for working with large data streams;
- integration with modern hardware and software platforms.

Solving this problem has important theoretical and practical significance for the development of next-generation computing systems capable of operating under the challenges of the digital economy and global digital infrastructure.

**Literature Review.** The literature review indicates that RNS are considered one of the most promising approaches for building high-performance and fault-tolerant computing architectures. Contemporary research primarily focuses on three key aspects: computational performance, reliability and redundancy, and issues related to integration with real-time systems [1–3; 6; 7].

The enhancement of performance and optimization of hardware resources in RNS has been actively studied in recent years. Works [2; 3] dedicated to the hardware implementation of the reverse transformation from RNS to positional number systems demonstrate that the use of specialized sets of modules, in combination with the Chinese Remainder Theorem or mixed radix methods, can significantly reduce latency and hardware costs [9]. Furthermore, research on methods for constructing module sets, such as the use of diagonal functions, shows that appropriate module selection can substantially improve computational efficiency. Notably, experiments on the application of RNS in highly parallel environments, particularly on GPUs [8], reveal that the large number of independent operations allows for significant acceleration in tasks involving large data sets [4].

Another important research direction concerns the reliability and fault tolerance of RNS [8]. Works [7; 9; 10] in this area emphasize that the inherent redundancy of the residue number system can be leveraged to build fault-tolerant architectures. Special attention is given to the concept of dynamic redundancy, which allows the level of fault tolerance to be adjusted depending on system operating conditions, thereby balancing resource consumption and reliability. Comparisons of reliability models for RNS and traditional binary systems show that, in certain scenarios, RNS can provide a higher probability of fault-free operation with lower hardware costs. This is particularly relevant for critical real-time systems.

At the same time, the literature notes several challenges that hinder the widespread adoption of RNS. Among the most complex tasks are magnitude comparison, sign determination, division, and reconstruction of numbers from residues [10; 11].

A separate line of research focuses on integrating RNS into high-performance real-time systems [3; 6; 12]. The use of FPGA [3], GPU [9; 10], and ASIC [11] technologies is considered a means to practically realize the potential of residue number systems in tasks such as digital signal processing, streaming data processing, and machine learning components. Interest is also growing in adaptive systems capable of dynamically modifying redundancy or reservation parameters depending on current operating conditions. This approach allows achieving the necessary balance between performance, energy efficiency, and fault tolerance [8; 9].

Thus, the literature highlights the significant potential of residue number systems in building high-performance and fault-tolerant real-time systems. At the same time, several fundamental and applied challenges remain, the resolution of which will pave the way for their widespread adoption in critical domains where speed, accuracy, and operational stability are crucial.

**Methods of Research.** The research methods are based on the principles of systems analysis, number theory, and the theory of computational processes and systems. To analyze the performance and fault tolerance of high-performance real-time systems, the foundations of computer architecture design and the principles of hardware component organization were applied. For the formalization and analysis of parallel data processing, approaches from the theory of computational processes were employed, allowing the evaluation of the temporal characteristics of arithmetic operations and the provision of guaranteed performance bounds [1; 6].

**Presentation of the Main Research Material.** High-performance real-time information processing systems are critically important in modern computing environments, where delays and failures can have irreversible consequences for control systems, telecommunications, medical equipment, and autonomous computing platforms. Achieving high performance while simultaneously ensuring reliability and fault tolerance requires a specialized numeric representation in the form of a RNS. In this system, a number  $A$  is represented as a set of residues over a modular set  $\{m_i\}_{i=1}^{n+k}$ :

$$A \equiv (a_1, a_2, \dots, a_n, \dots, a_{n+k}) \pmod{m_i}, i = 1, \dots, n+k,$$

where  $n$  is the number of information modules carrying the main computational data, and  $k$  is the number of check modules that provide redundancy and error correction. This structure defines the key properties of the RNS: residue independence, residue equivalence, and low-digit residues, each of which confers specific advantages in high-performance real-time systems [7].

Residue independence allows computational pipelines to be built as modular, autonomous subsystems that can operate in parallel without mutual interference. This ensures a high degree of parallelism in computations, which, in real-time systems, reduces latency and increases the throughput of the computing system [8]. The execution time of operations in such a system is determined by the largest modulus among the selected set:

$$T_{op} \sim \max_i(m_i),$$

where  $T_{op}$  is the execution time of the arithmetic operation defining the performance of the specific computational pipeline. Localizing errors within a single pipeline eliminates the "error propagation" effect characteristic of classical positional number systems, thus enhancing computational reliability [8].

Residue equivalence allows for the dynamic replacement of a non-functioning pipeline operating with modulus  $m_i$ , with another pipeline using modulus  $m_j$  ( $i \neq j$ ) without stopping computations. This property implements the principle of graceful degradation: if several computational pipelines fail, the system continues calculations with reduced precision while maintaining operational capability. Mathematically, this can be expressed through a new distribution of modules:

$$n' + k' = n + k = \text{const}, n' < n, k' > k,$$

where  $n'$  and  $k'$  are the new numbers of information and check modules, respectively. Reducing the number of information modules  $n'$  increases computational speed, while increasing the number of check modules  $k'$  enhances reliability. To improve accuracy, the redistribution is performed conversely:

$$n'' + k'' = n + k = \text{const}, n'' > n, k'' < k,$$

which increases result accuracy by reducing redundancy, slightly decreasing execution speed but providing more precise outcomes. Such dynamic adaptation allows the system to adjust to specific real-time task requirements, balancing speed, accuracy, and fault tolerance [8; 9].

Low-digit residues are another important characteristic of the RNS. They allow a reduction in the hardware resources of computational pipelines and increase their reliability. Additionally, they enable arithmetic operations to be implemented using table-based arithmetic, where the result of a modular operation can be obtained practically in a single clock cycle. Formally, addition and multiplication operations in RNS can be represented as:

$$C_i = (A_i \pm B_i) \bmod m_i, i = 1, \dots, n + k,$$

$$D_i = (A_i \cdot B_i) \bmod m_i, i = 1, \dots, n + k,$$

where  $A_i, B_i, C_i, D_i$  are the residues of the respective numbers. The inverse transformation from RNS to a positional number system is performed using the Chinese Remainder Theorem:

$$A = \left( \sum_{i=1}^n a_i M_i M_i^{-1} \right) \bmod M, M = \prod_{i=1}^n m_i,$$

where  $M_i = M/m_i$ , and  $M_i^{-1}$  is the multiplicative inverse of  $m_i$  modulo. Thanks to parallel processing of residues, the execution time of operations is reduced to the maximum of the execution times for individual modules, which significantly enhances performance compared to the classical binary system.

The combination of residue independence, equivalence, and low-digit representation in RNS enables the implementation of high-performance systems with the following advantages [7; 9]:

- parallelization of computations at the operand decomposition level;
- spatial separation of data elements with the possibility of asynchronous processing;
- single-cycle execution of modular operations in the table-based variant;
- efficient detection and correction of errors during computations;
- dynamic addition of small redundant blocks to increase fault tolerance;
- rapid reconfiguration of computational structures;
- reduced computational complexity for specific classes of tasks;
- elimination of the error propagation effect in modular pipelines;
- adaptability for real-time diagnostics of blocks and nodes.

The use of RNS is particularly effective for classes of tasks that are traditionally resource-intensive in positional number systems, such as cryptography and modular transformations, digital signal and image processing, high-bit integer data processing in real time, vector and matrix processing of large data arrays, neurocomputing, optoelectronic table-based processing, fast Fourier transform algorithms, and other parallel computational tasks. For such tasks, the execution time of modular operations is determined by the number of information modules  $n$ , while reliability is determined by the number of check modules  $k$ , enabling adaptive management of the balance between accuracy and fault tolerance.

Special attention is given to the dynamic reconfiguration of information and check modules during computations. When higher speed is required, the system can reduce the number of information modules  $n$  and increase the number of check modules  $k$ , thereby maintaining fault tolerance. Conversely, when higher accuracy is needed,  $n$  can be increased while  $k$  is reduced. This adaptation implements the principle of hardware-software flexibility, allowing the system to remain effective in real-time conditions despite changing workload characteristics or computational requirements.

Mathematically, the minimum code distance  $d_{\min}$ , which characterizes the error-correcting capability of the code, is defined as:

$$d_{\min} = \min_{i \neq j} \left| \sum_{l=1}^{n+k} (a_l - b_l) M_l M_l^{-1} \right| \bmod M,$$

where  $a_l, b_l$  are the residues of two distinct numbers. Increasing the number of check modules or adding new modules with larger moduli  $m_l$  increases  $d_{\min}$ , directly enhancing error correction capabilities and improving system fault tolerance. In real-time conditions, this allows “on-the-fly” error correction without interrupting computations, which is critical for control systems and streaming data processing.

Furthermore, RNS enables efficient implementation of modular arithmetic through various methods: the adder-based method (for low-digit modules), table-based method (using precomputed ROM tables), direct logic method at the level of Boolean functions, and circular shift method for registers. This provides a wide range of architectural solutions depending on requirements for speed, power consumption, and hardware complexity [2; 9].

Overall, the application of RNS in high-performance real-time systems allows the creation of computational structures that combine parallelism, adaptability, high reliability, and dynamic error correction capability. Such an architecture enables efficient processing of large volumes of data at high speed and guarantees continuous operation even in the event of partial component failures. This makes RNS a promising tool for building modern real-time computing systems capable of solving complex tasks in digital information processing, cryptography, neurocomputing, and streaming computations.

**Conclusions.** The conducted research has shown that the use of the RNS is an effective approach for building high-performance real-time information processing systems. The main advantages of this approach are determined by the properties of RNS: independence, uniformity, and low-digit residues. Independence of residues allows parallel data processing at the level of modular computational units, significantly increasing the system’s speed and throughput. Uniformity of residues provides the ability to dynamically replace failed units without stopping computations, implementing the principle of graceful system degradation, which is critical for real-time systems. Low-digit residues reduce the hardware complexity of units, enhance the speed of operations, and enable efficient error correction during computations.

Analysis of arithmetic operations in RNS has shown that execution time is determined by the largest modulus among the chosen set, while system accuracy and reliability can be dynamically adjusted through redistribution of information and check modules. This approach allows adaptive balancing between speed, accuracy, and fault tolerance depending on the specific task requirements, which is a key factor for real-time systems.

Moreover, RNS enables single-cycle execution of modular operations using table-based arithmetic, adder-based, logic-based, or circular-shift implementation principles, making these systems versatile for various architectural solutions. Due to these properties, RNS is effectively applied for processing large data arrays, digital signal and image processing, cryptographic computations, neurocomputing, and other tasks requiring high performance and fault tolerance.

Thus, the concept of building high-performance real-time systems based on RNS allows combining parallelism, adaptability, high reliability, and dynamic error correction capability. Implementing such systems ensures continuous computations even in the event of partial component failures, increases system performance and flexibility, making this approach promising for modern real-time computing platforms across various fields of science and engineering.

#### Bibliography:

1. Aliluiko A., Kasianchuk M. Arithmetic of Asymmetric Cryptosystems in the Field of Complex Numbers. *Ukrainian Information Security Research Journal*. 2024. Vol. 26, No. 1. P. 35–43. DOI: <https://doi.org/10.18372/2410-7840.26.18825>
2. Bovchalyuk S., Bovchalyuk N., Drozd O. Concept of “Modular Architecture” for Parallel-Action Control Devices. *Systems of Control, Navigation and Communication*. Poltava: PNTU, 2025. Vol. 2, No. 80. P. 47–53. DOI: <https://doi.org/10.26906/SUNZ.2025.2.046>
3. Bovchalyuk S. Ya., Piskaryov O. M., Radchenko S. S., et al. Definition of Directions for the Development of Control Devices with Parallel Architecture Based on FPGA. *Systems of Control, Navigation and Communication*. Poltava: PNTU, 2023. Issue 1 (71). P. 69–72. DOI: <https://doi.org/10.26906/SUNZ.2023.1.069>
4. Devi R. P., Neeraja P., Ajay V. K., Raj A. N., Reddy D. V. L., Ramachandran G. Analysis of Artificial Intelligence Hybrid Security Cloud System Intelligent Technology and its Applications. 2024 4th International Conference on Soft Computing for Security Applications (ICSCSA), Salem, India. 2024. P. 506–509. DOI: <https://doi.org/10.1109/ICSCSA64454.2024.00087>
5. Jency Rubia J., Sherin Shibi C., Balajishanmugam V., Babitha Lincy R. High-Performance Computing Based on Residue Number System: A Review. 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India. 2023. P. 639–647. DOI: <https://doi.org/10.1109/ICACCS57279.2023.10112959>

6. Koshman S., Krasnobayev V., Nikolsky S., Kovalchuk D. The Structure of the Computer System in the Residual Classes. *Advanced Information Systems*. 2023. Vol. 7, No. 2. P. 41–48. DOI: <https://doi.org/10.20998/2522-9052.2023.2.06>
7. Krasnobayev V., Koshman S., Kurchanov V., Zinevich D. Main Properties of Non-Positional Number System in Residue Classes and Their Influence on the Structure and Principles of Arithmetic Operations Implementation in Computer Systems. *Systems of Control, Navigation and Communication*. Poltava: PNTU, 2019. Vol. 2, No. 54. P. 114–118. DOI: <https://doi.org/10.26906/SUNZ.2019.2.114>
8. Krasnobayev V., Yanko A., Kovalchuk D., Fil I. Synthesis of a Mathematical Model of a Fault-Tolerant Real-Time Computer System Operating in Non-Positional Arithmetic in Residual Classes. *Mathematical Modeling and Simulation of Systems. MODS 2023. Lecture Notes in Networks and Systems*. Cham: Springer, 2024. Vol. 1091. P. 186–199. DOI: [https://doi.org/10.1007/978-3-031-67348-1\\_14](https://doi.org/10.1007/978-3-031-67348-1_14)
9. Radchenko S., Demchenko K., Hrytsenko S. Methods of Increase Reliability in Automated Control Systems. *SWorldJournal*. 2024. Vol. 1, No. 23-01. P. 111–115. DOI: <https://doi.org/10.30888/2663-5712.2024-23-00-033>
10. Shen S., Yang H., Liu Y., Liu Z., Zhao Y. CARM: CUDA-Accelerated RNS Multiplication in Word-Wise Homomorphic Encryption Schemes for Internet of Things. *IEEE Transactions on Computers*. 2023. Vol. 72, No. 7. P. 1999–2010. DOI: <https://doi.org/10.1109/TC.2022.3227874>
11. Shrimali D., Sharma L. An Extensive Review on Residue Number System for Improving Computer Arithmetic Operations, *International Research Journal of Engineering and Technology (IRJET)*. 2018. Vol. 5, Issue 12. P. 1617–1621
12. Yatskiv V., Kulyna S., Bykovyy P., Maksymyuk T., Sachenko A. Method of Reliable Data Storage Based on Redundant Residue Number System. 2020 IEEE 5th International Symposium on Smart and Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS), Dortmund, Germany. 2020. P. 1–4. DOI: [10.1109/IDAACS-SWS50031.2020.9297052](https://doi.org/10.1109/IDAACS-SWS50031.2020.9297052)

Дата надходження статті: 25.09.2025

Дата прийняття статті: 20.10.2025

Опубліковано: 04.12.2025