Methods of information systems synthesis

УДК 681.142 DOI: https://doi.org/10.20998/2522-9052.2022.1.07

Victor Krasnobayev, Sergey Koshman, Dmytro Kovalchuk

V. N. Karazin Kharkiv National University, Kharkov, Ukraine

THE CONCEPT OF PERFORMING THE ADDITION OPERATION IN THE SYSTEM OF RESIDUAL CLASSES

Abstract. The **subject** of the article is the development of a method for implementing the arithmetic operation of adding the residuals of numbers, which are represented in the system of residual classes (RNS). This method is based on the use of positional binary adders. The **purpose** of the article is to improve the performance of computer systems (CS) and their components by introducing new ways of organizing calculations based on the use of RNS. **Tasks**: to analyze and identify the shortcomings of the existing number systems that are used in the construction of computer systems and components; explore possible ways to eliminate the identified deficiencies; explore the structure of binary positional adders, taking into account the scheme for adding two residues of numbers modulo RNS; to develop a method for constructing adders modulo RNS, which is based on the use of a set of binary single-digit positional adders. **Research methods**: methods of analysis and synthesis of computer systems, number theory, coding theory in RNS. The following **results** are obtained. The paper shows that one of the promising ways to improve the performance of the CS is the use of RNS. The mathematical basis of RNS is the Chinese remainder theorem, which states that an integer operation on one large modulus can be replaced by a set of operations on coprime small modules. This opens up broad prospects for optimizing calculations. On the one hand, it is possible to significantly simplify the performance of complex and cumbersome calculations, including on low-resource computing platforms. On the other hand, calculations for different modules can be performed in parallel, which increases the performance of the CS. **Conclusions**. The article considers the operation of adding two numbers. This operation is the basis for both traditional positional number systems and RNS, i.e. forms the computational basis of all existing CS components. A new method for calculating the sum of the residuals of numbers modulo an arbitrary is proposed, and examples are given that clearly demonstrate the effectiveness of the proposed method. This method can be used in various computer applications, including for improving computing performance, ensuring fault tolerance, etc.

Keywords: number system; residue number system; non-positional code structure; operation of adding; positional binary adder.

Introduction

One of the components of a computer system (CS) in the positional binary number system (PNS) is an adder of two numbers [1–3]. In particular, the components of the CS are also adders modulo m_i of two numbers. This type of modulo adders is widely used both in the PNS and in the non-positional number system in the residual classes (RNS) [4, 5]. So, the adder of numbers in the RNS will consist of a set of k $n=[\log_2(m_i-1)+1]$ - bit adders modulo m_i $(i=\overline{1,k})$.

In this aspect, an urgent scientific and technical problem is the problem of constructing adders operating according to an arbitrary modulus m_i of the RNS.

If the residues a_i and b_i of numbers A and B in the RNS are represented in a binary PNS, then the adder of two residues a_i and b_i modulo m_i is a sequential collection of n binary one-bit adders.

The purpose of the article is improving the performance of computing systems and components through the introduction of new ways of organizing calculations based on the use of RNS. This is done by developing a method for performing the operation of modular addition $(a_i+b_i)\bmod m_i$ of two residues of numbers modulo m_i arbitrary, based on the use of a binary adder modulo $M=2^n-1$. The article provides examples of concrete execution of the method for performing the modular addition operation for various values of the residues a_i and b_i. The analysis of the considered examples showed the practical applicability of the method proposed in the article for performing the operation of modular addition $(a_i+b_i)\bmod m_i$ of two residues of the numbers A and B modulo m_i. The method proposed in the article for performing the operation of modular addition of two residues can be used both in the PNS and in the RNS.

Related work

The main theoretical provisions of the number system in the residual classes were laid in works [6, 7], as well as in [8–11] and many others [4, 5]. Computational algorithms for integer operations in RNS were substantiated on the basis of mathematical methods of number theory and the fundamental Chinese remainder theorem [12]. In subsequent works, various applied issues were investigated. In particular, mathematical models of integer operations in RNS are investigated in [13]. Papers [8] and [9] are devoted to error control in RNS. Improving the reliability and fault tolerance of computer systems was studied in [10, 12] and also in [14]. The construction of fault-tolerant computing architectures, including those based on FPGAs, is studied in [15]. In [10], the computational aspects of integer arithmetic are investigated as applied to cryptography problems. In works [16], new directions are investigated related to the computational processes of artificial neural networks and intelligent computing.

It is known that the central, basis of the operation performed by a computer system (CS) in a positional binary number system (PNS) is the operation of adding two numbers.

In a non-positional number system in residual classes (RNS), the main operation is the operation of adding $(a_i + b_i) \bmod m_i$ the residues a_i and b_i of numbers A and B according to a given modulus m_i.

In this case, the adder of numbers in the RNS will consist of a set of k $n = [\log_2(m_i - 1) + 1]$- bit adders modulo m_i $(i = \overline{1,k})$. Modular binary adders have a fixed modulus $M = 2^n - 1$, $M = 2^n$ or $M = 2^n + 1$.

This circumstance, in the general case, does not allow their direct use as modules m_i of RNS.

If the modulus m_i of the adder of residues a_i and b_i of numbers A and B differs from the value of M by a small amount, then the implementation of the modular addition operation can be relatively simple.

In the PNS, the most widespread are two options for the implementation of the operation of modular addition [16]. For each option, the following modulus ratios take place.

First variant. There is a ratio of modules $M = m_i + 1$. For the second variant, the ratio of modules takes place $M = m_i - 1$. For the variants considered, the implementation of the modular addition of residuals is quite simple. At the same time, in the general case of choosing the RNS modulus m_i , the operation of modular addition of two residues is a rather laborious task. In this aspect, an important problem arises of constructing adders for an arbitrary RNS modulus.

Method for performing the addition operation in the residual class system

The existing [16] method of adding $(a_i + b_i) \bmod m_i$ the residues of a_i and b_i numbers modulo m_i, based on the use of binary adders, consists of a combination of the following actions.

– *First action*. The structure of the adder modulo m_i is determined.

– *Second action*. The operation of bitwise addition modulo two residues a_i and b_i is performed.

– *Third action*. The result $S_n S_{n-1} ... S_2 S_1$ of the operation of bitwise addition modulo two residues a_i and b_i is placed in the corresponding binary one-bit adders (OBA) of the adder structure modulo m_i RNS.

– *Fourth action*. Based on the modular addition scheme, an algorithm for adding two residues a_i and b_i of numbers modulo m_i RNS is implemented.

The presented method has limited application. So, in the case of equality of two residues a_i and b_i, the result of modular addition will not always be accurate. This is due to the fact that the result $S_n S_{n-1} ... S_2 S_1$, does not always take into account the relationship between the values of the values of the modules m_i, M and the value of the sum $a_i + b_i$ of positional addition.

When developing a new method for adding the residues a_i and b_i it is necessary to take into account the options for the relationship between the values of the modules m_i, M and the value $a_i + b_i$ of the result of positional addition of the residues of numbers.

Consider a new method for adding the residues of a_i and b_i numbers modulo m_i.

A new method for implementing the addition operation from a combination of the following actions.

– *First action*. For the value of the modulus m_i, the synthesis of the adder modulo m_i is performed.

– *Second action*. The result of summation $a_i + b_i$ is determined.

– *Third action*. The result of the positional comparison of the result $a_i + b_i$ of addition and the value of the modulus of the adder m_i is determined.

– *Fourth action*. If the condition $a_i + b_i \le m_i$, is fulfilled, the result of the operation is the value of the sum $a_i + b_i$.

– *Fifth action*. When the condition $a_i + b_i > m_i$ is satisfied, the result $a_i + b_i$ of positional adding is bitwise entered into the corresponding $n = [\log_2(m_i - 1) + 1]$ bits of the OBA.

– *Sixth action*. In accordance with the algorithm for adding two residues a_i and b_i of numbers modulo, the operation of adding two residues a_i and b_i of numbers modulo is implemented.

Results of implementation of the method for performing the addition operation in the residual class system

Let's give examples of the implementation of the modular addition operation by the new method. We will consider examples of the implementation of the modular addition operation in two modes. First mode $a_i + b_i \le m_i$ and second mode $(a + b) > m_i$. Let $m_i = 11$.

Example 1. Residues are $a_i = 5$ and $b_i = 4$. The adder implements the operation of positional addition of the residues $a_i = 0101$ and $b_i = 0100$ in the form

$$a_i + b_i = 1001.$$

The value of the sum $a_i + b_i = 1001$ determines the result of the operation.

Check: $(0101 + 0100) = 1001 (\bmod 11)$.

Example 2. Let $a_i = 5$ and $b_i = 6$. The adder implements the operation of positional addition of the residues $a_i = 0101$ and $b_i = 0110$ in the form

$$a_i + b_i = 1011.$$

The value of the positional sum $a_i + b_i = 1011$ of the residues $a_i = 0101$ and $b_i = 0110$ is bitwise fed to the corresponding inputs of the OBA $5_4 - 5_1$. Thus, OBA $5_4 - 5_1$ of the adder contain respectively the values 1011. The modular addition operation is implemented according to the modular addition scheme modulo $m_i = 11$ [16]. The algorithm for implementing a modular operation is presented in Table 1 and in Fig. 1. The unit of the binary digit is supplied to the input OBA 5_3 and 5_1. Fig. 1 shows a scheme of the addition of residues $a_i = 0101$ and $b_i = 0110$ modulo $m_i = 11$.

Check: $(0101 + 0110) = 0000(\text{mod}11)$.

Table 1 – **Algorithm for executing the result of the operation for a_i = 5 and b_i = 6**

OBA $5_4 - 5_1$	**OBA content** $5_4 - 5_1$	**The presence of a unit at the DOS inputs** $5_4 - 5_1$	**The result of the modular addition operation**
5_1	1	+1	0
5_2	1	–	0
5_3	0	+1	0
5_4	1	–	0

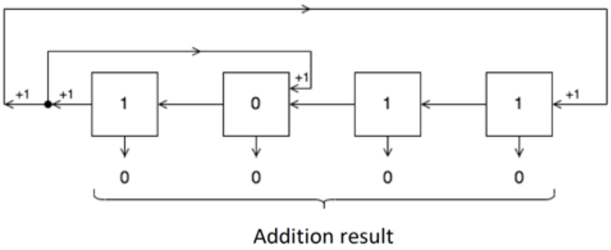

Fig. 1. Scheme for adding residues a_i = 0101 and b_i = 0110 modulo m_i = 11

Example 3. Let $a_i = 5$ and $b_i = 7$. The adder implements the operation of positional addition of the residues $a_i = 0101$ and $b_i = 0111$ in the form $a_i + b_i = 1100$.

The value of the positional sum $a_i + b_i = 1100$ of the residues $a_i = 0101$ and $b_i = 0111$ is bitwise fed to the corresponding inputs of the OBA $5_4 - 5_1$. Thus, OBA $5_4 - 5_1$ of the adder contain respectively the values 1100. The modular addition operation is implemented according to the modular addition scheme modulo $m_i = 11$.

The algorithm for implementing a modular operation is presented in Table 2 and in Fig. 2. The unit of the binary digit is supplied to the input OBA 5_3 and 5_1.

Check: $(0101 + 0111) = 0001(\text{mod}11)$.

Example 4. Let $a_i = 5$ and $b_i = 9$. The adder implements the operation of positional addition of the residues $a_i = 0101$ and $b_i = 1001$ in the form $a_i + b_i = 1110$. The value of the positional sum $a_i + b_i = 1110$ of the residues $a_i = 0101$ and $b_i = 1001$ is bitwise fed to the corresponding inputs of the OBA $5_4 - 5_1$. Thus, OBA $5_4 - 5_1$ of the adder contain respectively the values 1110. The algorithm for implementing a modular operation is presented in Table 3 and in Fig. 3. The unit of the binary digit is supplied to the input OBA 5_3 and 5_1.

Check: $(0101 + 1001) = 0011(\text{mod}11)$.

Table 2 – **Algorithm for executing the result of the operation for a_i = 5 and b_i = 7**

OBA $5_4 - 5_1$	**OBA content** $5_4 - 5_1$	**The presence of a unit at the OBA inputs** $5_4 - 5_1$	**The result of the modular addition operation**
5_1	0	+1	1
5_2	0	–	0
5_3	1	+1	0
5_4	1	–	0

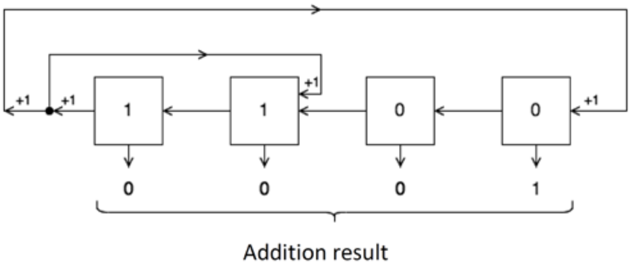

Fig. 2. Scheme for adding residues a_i = 0101 and b_i = 0111 modulo m_i = 11

Table 3 – **Algorithm for executing the result of the operation for a_i = 5 and b_i = 9**

OBA $5_4 - 5_1$	**OBA content** $5_4 - 5_1$	**The presence of a unit at the OBA inputs** $5_4 - 5_1$	**The result of the modular addition operation**
5_1	0	+1	1
5_2	1	–	1
5_3	1	+1	0
5_4	1	–	0

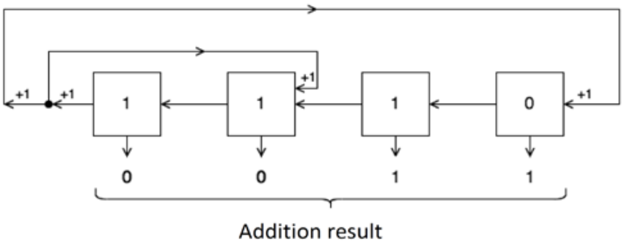

Fig. 3. Scheme for adding residues a_i = 0101 and b_i = 0110 modulo m_i = 11

Conclusion

The number system in the residual classes is a promising direction for the qualitative improvement of modern computer systems. In particular, the execution of integer computational operations can be significantly accelerated by reducing the bit depth of the computations.

Performing operations on a large module can be replaced by several operations on small modules. And this is at the heart of the computational operations in RNS. We have considered the simplest operations of addition of the remainders modulo an arbitrary. These calculations are used both in the traditional positional number system and in the number system in residual classes.

The article describes a new method for performing the addition operation $(a_i + b_i) \bmod m_i$ of two residues a_i and b_i of numbers modulo.

The addition operation $(a_i + b_i) \bmod m_i$ is implemented based on the use of the structure of a binary adder modulo m_i of RNS, and also taking into account the scheme for adding two residues a_i and b_i of numbers.

The method for constructing binary adders modulo m_i is based on the structure of an adder modulo $M = 2^n - 1$, which consists of a set of sequential binary one-bit adders.

Examples of executing the method of performing the addition operation for various values of the residues a_i and b_i are given.

The analysis of the considered examples showed the practical applicability of the method proposed in the article for performing the modular addition operation.

The proposed method can be used in various computer applications, including for improving computing performance, providing fault tolerance, etc.

References

1. R.S. Alford, "Computer Systems Engineering Management", *CRC Press*, 2018. https://doi.org/10.1201/9781351070829.
2. P.V. Ananda Mohan, "Residue Number Systems", *Springer International Publishing, Cham*, 2016. https://doi.org/10.1007/978-3-319-41385-3.
3. P.V. Mohan, CDAC, Bangalore, "Implementation of Residue Number System Based Digital Filters", *A Quarterly Publication of ACCS, (n.d.).* https://journal.accsindia.org/implementation-of-residue-number-system-based-digital-filters/ (accessed August 16, 2020).
4. J.O. Tuazon, "Residue number system in computer arithmetic", *Doctor of Philosophy, Iowa State University, Digital Repository*, 1969. https://doi.org/10.31274/rtd-180816-2270.
5. F. Barsi, P. Maestrini, "Error Correcting Properties of Redundant Residue Number Systems", *IEEE Transactions on Computers*. (1973) 307–315. https://doi.org/10.1109/T-C.1973.223711.
6. M.G. Arnold, "The residue logarithmic number system: theory and implementation", in: *17th IEEE Symposium on Computer Arithmetic (ARITH'05)*, 2005: pp. 196–205. https://doi.org/10.1109/ARITH.2005.44.
7. M.Z. Garaev, A.A. Karatsuba, "The representation of residue classes by products of small integers", *Proceedings of the Edinburgh Mathematical Society*. 50 (2007) 363–375. https://doi.org/10.1017/S0013091505000969.
8. S. Timarchi, K. Navi, "Efficient Class of Redundant Residue Number System", in: *2007 IEEE International Symposium on Intelligent Signal Processing*, 2007: pp. 1–6. https://doi.org/10.1109/WISP.2007.4447506.
9. P.V. Ananda Mohan, "Error Detection, Correction and Fault Tolerance in RNS-Based Designs", in: *P.V. A. Mohan (Ed.), Residue Number Systems: Theory and Applications, Springer International Publishing, Cham*, 2016: pp. 163–175. https://doi.org/10.1007/978-3-319-41385-3_7.
10. V.M. Amerbaev, R.A. Solovyev, A.L. Stempkovskiy, D.V. Telpukhov, "Efficient calculation of cyclic convolution by means of fast Fourier transform in a finite field", in: *Proceedings of IEEE East-West Design Test Symposium* (EWDTS 2014), 2014: pp. 1–4. https://doi.org/10.1109/EWDTS.2014.7027043.
11. T.-C. Huang, "Self-Checking Residue Number System for Low-Power Reliable Neural Network", in: *2019 IEEE 28th Asian Test Symposium (ATS)*, 2019: pp. 37–375. https://doi.org/10.1109/ATS47505.2019.000-3.
12. V. Krasnobayev, A. Kuznetsov, A. Yanko, K. Kuznetsova, "Correction Codes in the System of Residual Classes", *in*: *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC&ST)*, 2019: pp. 488–492. https://doi.org/10.1109/PICST47496.2019.9061253.
13. D.I. Popov, A.V. Gapochkin, "Development of Algorithm for Control and Correction of Errors of Digital Signals, Represented in System of Residual Classes", in: *2018 International Russian Automation Conference (RusAutoCon)*, 2018: pp. 1–3. https://doi.org/10.1109/RUSAUTOCON.2018.8501826.
14. P.V. Ananda Mohan, "Specialized Residue Number Systems", in: P.V.A. Mohan (Ed.), *Residue Number Systems: Theory and Applications, Springer International Publishing, Cham*, 2016: pp. 177–193. https://doi.org/10.1007/978-3-319-41385-3_8.
15. M. Karpinski, S. Ivasiev, I. Yakymenko, M. Kasianchuk, T. Gancarczyk, "Advanced method of factorization of multi-bit numbers based on Fermat's theorem in the system of residual classes", in: *2016 16th International Conference on Control, Automation and Systems (ICCAS)*, 2016: pp. 1484–1486. https://doi.org/10.1109/ICCAS.2016.7832500.
16. V. Krasnobaev, A. Kuznetsov, V. Babenko, M. Denysenko, M. Zub, V. Hryhorenko, "The Method of Raising Numbers, Represented in the System of Residual Classes to an Arbitrary Power of a Natural Number", in: *2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, 2019: pp. 1133–1138. https://doi.org/10.1109/UKRCON.2019.8879793.

Received (Надійшла) 14.12.2021
Accepted for publication (Прийнята до друку) 02.02.2022

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.

Кошман Сергій Олександрович – доктор технічних наук, доцент, доцент кафедри безпеки інформаційних систем і технологій, Харківський національний університет імені В. Н. Каразіна, Харків, Україна;

Sergey Koshman – Doctor of Technical Sciences, professor, asc. professor, 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.

Ковальчук Дмитро Миколайович – аспірант кафедри безпеки інформаційних систем і технологій, Харківський національний університет імені В. Н. Каразіна, Харків, Україна;

Dmytro Kovalchuk– PhD student of Information Systems and Technologies Security Department, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine

e-mail: kovalchuk.d.n@ukr.net; ORCID ID: https://orcid.org/0000-0002-8229-836X.

Концепція виконання операції додавання у системі залишкових класів

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

Анотація. **Предметом** статті є розробка методу реалізації арифметичної операції додавання залишків чисел, що представлені у системі залишкових класів (СЗК). Цей метод заснований на використанні позиційних двійкових суматорів. **Метою** статті є підвищення продуктивності комп'ютерних систем (КС) та їх компонентів за рахунок застосування нових методів організації обчислень, які засновані на використанні СЗК. **Задачі**: провести аналіз та виявити недоліки існуючих систем числення, що використовуються при побудові комп'ютерних систем та компонентів; дослідити можливі шляхи усунення виявлених недоліків; дослідити структуру двійкових позиційних суматорів з урахуванням схеми додавання двох залишків чисел за модулем СЗК; розробити метод побудови суматорів за модулем СЗК, який заснований на використанні набору двійкових однорозрядних позиційних суматорів. **Методи дослідження**: методи аналізу та синтезу комп'ютерних систем, теорія чисел, теорія кодування у СЗК. Отримано наступні **результати**. У роботі показано, що одним із перспективних напрямів підвищення продуктивності КС є застосування СЗК. Математичною основою СЗК є китайська теорема про залишки, яка стверджує, що цілочислова операція за одним великим модулем може бути замінена набором операцій за взаємно простими малими модулями. Це відкриває широкі перспективи оптимізації обчислень. З одного боку, можна спростити виконання складних і громіздких обчислень, у тому числі на малоресурсних обчислювальних платформах. З іншого боку, обчислення за різними модулями можуть виконуватись паралельно, що підвищує продуктивність КС. **Висновки**. У статті розглянуто операцію додавання двох чисел. Ця операція є основою як для традиційних позиційних систем числення, так і для СЗК, тобто складає обчислювальну основу всіх компонентів КС. Запропоновано новий метод обчислення суми залишків чисел за довільним модулем і наведено приклади, що наочно демонструють ефективність запропонованого методу. Цей метод можна використовувати у різних комп'ютерних додатках, зокрема для підвищення продуктивності обчислень, забезпечення відмовостійкості та інш.

Ключові слова: система числення; система залишкових класів; непозиційна кодова структура; операція додавання; позиційний двійковий суматор.

Концепция выполнения операции сложения в системе остаточных классов

В. А. Краснобаев, С. А. Кошман, Д. Н. Ковальчук

Аннотация. **Предметом** статьи является разработка метода реализации арифметической операции сложения остатков чисел, которые представлены в системе остаточных классов (СОК). Данный метод основан на использовании позиционных двоичных сумматоров. **Целью** статьи является повышение производительности компьютерных систем (КС) и их компонентов за счёт внедрения новых способов организации вычислений, основанных на использовании СОК. **Задачи**: провести анализ и выявить недостатки существующих систем счисления, которые используются при построении компьютерных систем и компонентов; исследовать возможные пути устранения выявленных недостатков; исследовать структуру двоичных позиционных сумматоров с учетом схемы сложения двух остатков чисел по модулю СОК; разработать метод построения сумматоров по модулю СЗК, который основан на использовании набора двоичных одноразрядных позиционных сумматоров. **Методы исследования**: методы анализа и синтеза компьютерных систем, теория чисел, теория кодирования в СОК. Получены следующие **результаты**. В работе показано, что одним из перспективных направлений повышения производительности КС является применение СОК. Математической основой СОК является китайская теорема об остатках, которая утверждает, что целочисленная операция по одному большому модулю может быть заменена набором операций по взаимно простым малым модулям. Это открывает широкие перспективы для оптимизации вычислений. С одной стороны, можно значительно упростить выполнение сложных и громоздких вычислений, в том числе на малоресурсных вычислительных платформах. С другой стороны, вычисления по разным модулям могут выполняться параллельно, что повышает производительность КС. **Выводы**. В статье рассмотрена операция суммирования двух чисел. Данная операция лежит в основе как для традиционных позиционных систем счисления, так и для СОК, т.е. составляет вычислительную основу всех существующих компонентов КС. Предложен новый метод вычисления суммы остатков чисел по произвольному модулю и приведены примеры, наглядно демонстрирующие эффективность предложенного метода. Данный метод может быть использован в различных компьютерных приложениях, в том числе для повышения производительности вычислений, обеспечения отказоустойчивости и др.

Ключевые слова: система счисления; система остаточных классов; непозиционная кодовая структура; операция сложения; позиционный двоичный сумматор.