Современная наука и инновации.
2023. №2 (42). С. 41–49
Modern Science and Innovations.
2023;2(42):41-49

ТЕХНИЧЕСКИЕ НАУКИ /
TECHNICAL SCIENCE

ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ /
INFORMATICS, COMPUTER ENGINEERING AND MANAGEMENT

Научная статья / Original article

УДК 519.684.6
DOI: 10.37493/2307-910X.2023.2.4

Виктор Андреевич Кучуков
[Viktor A. Kuchukov]
Михаил Григорьевич Бабенко
[Mikhail G. Babenko]
Николай Николаевич Кучеров
[Nikolay N. Kucherov]

Исследование ранга числа в системе остаточных классов

Investigating the rank of the number in a residue number system

Северо-Кавказский федеральный университет,
Северо-Кавказский центр математических исследований, Ставрополь, Россия /
North-Caucasus Federal University, North-Caucasus Center for Mathematical Research, Stavropol, Russia,
vkuchukov@ncfu.ru, mgbabenko@ncfu.ru, nkucherov@ncfu.ru

Аннотация. *Ранг числа в системе остаточных классов показывает количество переходов через диапазон при переводе числа в позиционную систему счисления и позволяет повысить эффективность немодульных операций и определить выход значений за диапазон. Основным подходом к вычислению ранга является использование Китайской теоремы об остатках. В статье предложен подход, позволяющий вычислить ранг с использованием набора специальных чисел, для которых заранее вычислены ранги. Моделирование рассмотренных методов произведено на языке программирования Python. Проведен анализ полученных результатов и даны рекомендации дальнейшего использования предложенного метода.*

Ключевые слова: система остаточных классов, ранг числа

Финансирование: Работа выполнена при финансовой поддержке Совета по грантам Президента Российской Федерации в рамках стипендии Президента СП-3186.2022.5.

Для цитирования: Кучуков В. А., Бабенко М. Г., Кучеров Н. Н. Исследование ранга числа в системе остаточных классов // *Современная наука и инновации. 2023. №2 (42). С. 41–49. https://doi.org/10.37493/2307-910X.2023.2.4*

Abstract. *The rank of a number in a residue number system indicates the count of transitions through a range when a number is converted to a positional number system and allows for more efficient non-modular operations and detection of values out of range. The main approach to calculate the rank is the use of the Chinese Remainder Theorem. In this article the approach which allows to compute the rank using a set of special numbers for which ranks are computed in advance is proposed. The simulation of the considered methods is done in the Python programming language. The results are analyzed and recommendations for further use of the proposed method are given.*

Key words: residue number system, rank of the number

Funding: The research was supported by Russian Federation President Grant SP-3186.2022.5.

For citation: Kuchukov V. A., Babenko M. G., Kucherov N. N. Investigation of the rank of a number in the system of residual classes. *Modern Science and Innovations. 2023;2(42):41-49. https://doi.org/10.37493/2307-910X.2023.2.4*

Introduction. The Residual Class System (RCS) is a non-positional number system in which a number is represented as remainders on a set of coprime numbers called RCS modules. In this case, operations such as addition, subtraction and multiplication can be performed independently for each of the remainders, the size of which is much less than the size of the original number. In the case of large numbers, in particular, more than 32 bits, the size of the modules can be of the order of 7-9 bits. Thus, the system of residual classes finds its application in cryptography [1-2], digital filtering [3], and other areas in which the main operations are addition and multiplication. The corrective properties of the residual class system allow it to be used to detect and correct errors [4]. The system of residual classes is especially effective in the case of its implementation on specialized integrated circuits [5], which allow you to build a circuit taking into account the size of the modules and the required modular operations.

However, a number of operations in RCS, called non-modular, require knowledge of the positional characteristics of the number. Such operations include division, comparison of numbers, determination of the sign of a number. Increasing the efficiency of computing these operations can be solved by reducing the computational complexity of determining the rank of a number in RCS. Another application of the number rank function is to control arithmetic operations.

The article is further organized as follows. Section 1 discusses the basics of the residual class system and known methods for calculating the rank. Section 2 proves a number of assertions that make it possible to increase the efficiency of the rank calculation. Section 3 is devoted to modeling the considered methods in Python. In conclusion, recommendations are given for the further use of the proposed method.

1. Investigation of the rank of a number in the system of residual classes

Let a system of residual classes with modules be given $\{p_1, p_2, \dots, p_n\}$, which allows representing the number $X \in [0, P)$, where $P = \prod_{i=1}^{n} p_i$, in a unique way in the form of remainders after dividing the number $X$by modules p_i, i.e. in the form $X = (x_1, x_2, \dots, x_n)$[6].

At the same time, to obtain a representation of the number Xgiven in the form of remainders $(x_1, x_2, \dots, x_n)$in the positional number system, we use the Chinese Remainder Theorem (CRT):

$$X \equiv \left| \sum_{i=1}^{n} x_i B_i \right|_P = \sum_{i=1}^{n} x_i B_i - r_X P \quad (1)$$

where $B_i = P_i \cdot \left|P_i^{-1}\right|_{p_i}$ are RCS bases, $P_i = P/p_i$, $\left|P_i^{-1}\right|_{p_i}$ is multiplicative inversion. $r_X$is a positive integer, called the rank of the number X, showing how many times the dynamic range of the system $P$was surpassed when moving from the representation in the residual class system to the positional representation of the number.

Using the rank of a number allows you to establish the fact of the exit or non-exit of the result of arithmetic operations from the range $[0, P)$, i.e. range overflow. Also, the rank of a number is used in the error correction of a modular code, to determine the sign of a number.

From (1) the rank can be found by the formula

$$r_X = \left\lfloor \sum_{i=1}^{n} \frac{\left|P_i^{-1}\right|_{p_i}}{p_i} x_i \right\rfloor. \quad (2)$$

We transform formula (1) as follows:

$$X = \left| \sum_{i=1}^{n} P_i \cdot \left| \left|P_i^{-1}\right|_{p_i} \cdot x_i \right|_{p_i} \right|_P = \sum_{i=1}^{n} P_i \cdot \left| \left|P_i^{-1}\right|_{p_i} \cdot x_i \right|_{p_i} - \hat{r}(X) \cdot P.$$

In this case $\hat{r}(X) = \left\lfloor \sum_{i=1}^{n} \frac{1}{p_i} \cdot \left| \left|P_i^{-1}\right|_{p_i} \cdot x_i \right|_{p_i} \right\rfloor$, is the normalized rank of the number.

Consider an example of calculating the rank for an RCS {2,3,5}for which $P = 30$, $P_1 = 15$, $P_2 = 10$, $P_3 = 6$, $|P_1^{-1}|_{p_1} = 1$, $|P_2^{-1}|_{p_2} = 1$, $|P_3^{-1}|_{p_3} = 1$.

Then from formula (1) for the number $X = 14 = (0,2,4)$we obtain

$$15 \cdot 1 \cdot 0 + 10 \cdot 1 \cdot 2 + 6 \cdot 1 \cdot 4 - r_X \cdot 30 = 14 \Rightarrow r_X = 1.$$

Similarly, from formula (2):

$$r_X = \left\lfloor \frac{1}{2} \cdot 0 + \frac{1}{3} \cdot 2 + \frac{1}{5} \cdot 4 \right\rfloor = 1.$$

Obviously, calculations on a computer with fractional values according to formula (2) are difficult to implement and can lead to rounding errors. In the article [7], an approximate implementation of the rank of a number is proposed, for which the function

$$\mathrm{R_X} = \left\lfloor \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{k_i\, x_i}/2^{\mathrm{N}} \right\rfloor, \quad (3)$$

where $k_i = \left\lfloor \left|P_i^{-1}\right|_{p_i} 2^N / p_i \right\rfloor$. The rank value is calculated from Theorem 1.

Theorem 1. If $N = \lceil \log_2 \rho \rceil$, then $r_X = R_X$or $r_X = R_X - 1$, where $\rho = \sum_{i=1}^{n} p_i - n$.

For SOC {2,3,5}value $N = \lceil \log_2 (1 + 2 + 4) \rceil = 3$, then

$k_1 = \left\lceil \frac{1 \cdot 2^3}{2} \right\rceil = 4$, $k_2 = \left\lceil \frac{1 \cdot 2^3}{3} \right\rceil = 3$, $k_3 = \left\lceil \frac{1 \cdot 2^3}{5} \right\rceil = 2$.

Then

$$R_X = \left\lfloor \frac{4 \cdot 0 + 3 \cdot 2 + 2 \cdot 4}{2^3} \right\rfloor = 1.$$

From Theorem 1, the rank of the number $r_X = R_X = 1$or $r_X = R_X - 1 = 0$. Clearly, clarification is needed to determine the exact value.

If we use the estimate $N = \lceil \log_2 P\, \rho \rceil$from [8], then the resulting rank is $R_X = r_X$, but in this case the dimension of the operand increases significantly.

Consider methods for calculating the rank in the case when the ranks of a series of numbers are known.

In [6], Theorem 2 on the rank of a sum was introduced.

Theorem 2. If in the system of residual classes with modules $\{p_1, p_2, \ldots, p_n\}$and range Ptwo numbers $X = (x_1, x_2, \ldots, x_n)$and $Y = (y_1, y_2, \ldots, y_n)$with ranks $r_X$and $r_Y$respectively are given, then the rank $r_{X+Y}$of the sum of these numbers is equal to

$$r_{X+Y} = r_X + r_Y - \sum_{i=1}^{n} \left\lceil \frac{x_i + y_i}{p_i} \right\rceil \cdot \left|P_i^{-1}\right|_{p_i}. \quad (4)$$

Expression (4) allows us to calculate the rank of the sum over the ranks of the terms. In this case $\left\lceil \frac{x_i + y_i}{p_i} \right\rceil = 1$, if $x_i + y_i \geq p_i$and $\left\lceil \frac{x_i + y_i}{p_i} \right\rceil = 0$otherwise, i.e. there is an addition of those $\left|P_i^{-1}\right|_{p_i}$for which $x_i + y_i \geq p_i$, i.e. $\sum_{x_i + y_i \geq p_i} \left|P_i^{-1}\right|_{p_i}$.

Consider an example. Let's take two numbers $X = 14 = (0,2,4)$and $Y = 15 = (1,0,0)$in SOC {2,3,5}. It is shown above that $r_X = 1$. For $Y$by formula (2) we obtain

$$r_Y = \left\lfloor \frac{1}{2} \cdot 1 + \frac{1}{3} \cdot 0 + \frac{1}{5} \cdot 0 \right\rfloor = 0.$$

When adding numbers $x_i$and $y_i$there was no transition through the module, therefore $\sum_{x_i + y_i \geq p_i} \left|P_i^{-1}\right|_{p_i} = 0$.

Then $r_{X+Y} = 1 + 0 + 0 = 1$. Let's check the value, find the rank of the sum $X + Y = (1,2,4)$.

$$r_{X+Y} = \left\lfloor \frac{1}{2} \cdot 1 + \frac{1}{3} \cdot 2 + \frac{1}{5} \cdot 4 \right\rfloor = \left\lfloor \frac{59}{30} \right\rfloor = 1.$$

Theorem 2 allows us to simplify the algorithm for finding the rank of a number. Thus, in [6] it is proposed to use the numbers $M_1 = 1$, $M_2 = p_1$, $M_3 = p_1 p_2$, …, $M_n = p_1 \cdot p_2 \cdot \ldots \cdot p_{n-1}$and their ranks to calculate the rank of the number A. To do this, $A$the number is added to the number M_1so many times that the digit of the number Ain the base $p_1$becomes equal to zero. Similar actions are carried out with other bases until a number is obtained $P = (0,0, \ldots, 0)$for which the rank is -1.

Consider this method for RCS {2,3,5}and numbers $X = 14 = (0,2,4)$in RCS {2,3,5}. Then the values $M_i$and their ranks are equal:

$M_1 = (1,1,1) = 1, r_1 = \left\lfloor \frac{15\cdot1+10\cdot1+6\cdot1}{30} \right\rfloor = 1, M_2 = (0,2,2) = 2, r_2 = \left\lfloor \frac{15\cdot0+10\cdot2+6\cdot2}{30} \right\rfloor = 1,$

$M_3 = (0,0,1) = 2 \cdot 3 = 6, r_3 = \left\lfloor \frac{15\cdot0+10\cdot0+6\cdot1}{30} \right\rfloor = 0.$

Thus,

$$r_{X+M_2} = r_X + r_2 - |P_2^{-1}|_{p_2} - |P_3^{-1}|_{p_3} = r_X + 1 - 1 - 1 = r_X - 1,$$
$$r_{X+2M_2} = (r_X - 1) + r_2 - |P_2^{-1}|_{p_2} = (r_X - 1) + 1 - 1 = r_X - 1,$$
$$r_{X+2M_2+M_3} = (r_X - 1) + r_3 = (r_X - 1) + 0 = r_X - 1,$$
$$r_{X+2M_2+2M_3} = (r_X - 1) + r_3 - |P_3^{-1}|_{p_3} = (r_X - 1) + 0 - 1 = r_X - 2 = r_P = -1.$$

Then $r_X - 2 = -1$from where $r_X = 1$.

The disadvantage of this method is the uncertainty of the required number of additions.

The development of this idea is proposed in the article [9], which considers the non-iterative calculation of multiplicities $\alpha_i$and the introduction of additional substitution tables for storing ranks $r(\alpha_i M_i)$and variables $w_i$for calculating the transition through the module. In this method, $M_i$multiplicative inversions are calculated for bases $\left|M_i^{-1}\right|_{p_i}$. In addition, for each module, p_ia substitution table of $p_i - 1$word length is calculated, each j-th line of which $(1 \le j \le p_i - 1)$,contains the rank $r(\alpha_i M_i)$, where $\alpha_i = \left| j \cdot \left|M_i^{-1}\right|_{p_i} \right|_{p_i}$ is the multiplicity of the number.

For a number, the multiplicity $X_0 = X = \left(x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}\right)$and product $\alpha_1 M_1 = \left(m_1^{(1)}, m_2^{(1)}, \dots, m_n^{(1)}\right)$are calculated $\alpha_1 = \left| \left(p_1 - x_1^{(0)}\right) \cdot |M_1^{-1}|_{p_1} \right|_{p_1}$, for which the sum $X_1 = X_0 + \alpha_1 M_1$will look like $X_1 = \left(0, x_2^{(1)}, \dots, x_n^{(1)}\right)$. When summing, the transition through the modulus $p_i$is fixed in the variable $w_i = \left\lfloor \frac{x_i^{(0)} + m_i^{(1)}}{p_i} \right\rfloor$. And the rank is selected from the lookup table $r(\alpha_i M_i)$.

The described steps are repeated for $i \in [2, n]$. The result is a number $X_n = P = (0,0, \dots,0)$whose rank is -1.

Then the rank of the original number

$$r(X) = \sum_{i=1}^{n} w_i \left|P_i^{-1}\right|_{p_i} - \sum_{i=1}^{n} r(\alpha_i M_i) - 1. \quad (5)$$

Consider an example for RCS {2,3,5}. Then the table for storing ranks $r(\alpha_i M_i)$will look like this:

$$\begin{pmatrix} 1 & & \\ 1 & 1 & \\ 0 & 0 & 0 \end{pmatrix}.$$

Let's consider a number $X = (0,2,4)$. Then $\alpha_2 = \left|(p_2 - x_2) \cdot |M_2^{-1}|_{p_2}\right|_{p_2} = |(3-2) \cdot |2^{-1}|_3|_3 = 2$, $\alpha_2 M_2 = (0,1,4)$, $r(\alpha_2 M_2) = 1$.

The amount is $X_2 = X + \alpha_2 M_2 = (0,2,4) + (0,1,4) = (0,0,3)$. In this case, there were transitions through the module according to the bases $p_2$and p_3, whence $w_1 = 0$, $w_2 = 1$, $w_3 = 1$.

Next, the multiplicity is $\alpha_3 = \left|(p_3 - x_3) \cdot |M_3^{-1}|_{p_3}\right|_{p_3} = |(5-3) \cdot |6^{-1}|_5|_5 = 2$ calculated.$\alpha_3 M_3 = (0,0,2)$, $r(\alpha_3 M_3) = 0$

The amount is $X_3 = X_2 + \alpha_3 M_3 = (0,0,3) + (0,0,2) = (0,0,0)$. In this case, there was a transition through the module according to the base p_3, from which $w_1 = 0$, $w_2 = 1$, $w_3 = 1 + 1 = 2$.

Then from formula (5) we get:

$$r(X) = w_1|P_1^{-1}|_{p_1} + w_2|P_2^{-1}|_{p_2} + w_3|P_3^{-1}|_{p_3} - r(\alpha_1 M_1) - r(\alpha_2 M_2) - r(\alpha_3 M_3) - 1 =$$
$$= 0 \cdot 1 + 1 \cdot 1 + 2 \cdot 1 - 0 - 1 - 0 - 1 = 1 + 2 - 2 = 1.$$

2. Modification of the method for calculating the rank

Let us introduce a new approach that allows us to simplify the above calculations.

Let us take as bases $E_i$for calculating the rank the numbers consisting of i zeros and ni ones, i.e. $E_0 = (1,1,\dots,1)$, $E_1 = (0,1,\dots,1)$, …, $E_{n-1} = (0,0,\dots,1)$. To simplify obtaining the digits of a number from the basis, we introduce a statement about the rank of the basis multiplied by a constant.

Statement 1. For an RCS with modules $\{p_1, p_2, \dots, p_n\}$and bases of rank E_i, $i = 0, \dots, n-1$, where $E_i = (\underbrace{0, \dots, 0}_{i}, \underbrace{1, \dots, 1}_{n-i})$, then for any $a \in N$satisfying the condition $0 \le a < p_{k+1}$, the expression

$$r_{a \cdot E_k} = a \cdot r_{E_k} + \left\lfloor \frac{a}{M_k} \cdot \left| \frac{1}{\prod_{i=1}^{k} p_i} \right|_{M_k} \right\rfloor, \quad (6)$$

where for $k = 0$: $\prod_{i=1}^{0} p_i = 1$, $M_0 = P$, for $k = 1, \dots, n-1$: $M_k = \frac{P}{\prod_{i=1}^{k} p_i}$.

Consider an example for RCS {2,3,5}.

For $E_1 = (0,1,1)$we get $r_{E_1} = 0$, $M_1 = \frac{P}{p_1} = 15$, $\left| \frac{1}{\prod_{i=1}^{1} p_i} \right|_{M_1} = 8$.

Then $2E_1 = (0,2,2)$, $r(0,2,2) = 1$and from (6):

$$r_{2E_1} = 2 \cdot 0 + \left\lfloor \frac{2}{15} \cdot 8 \right\rfloor = 1.$$

For $E_2 = (0,0,1)$we get $r_{E_2} = 0$, $M_2 = \frac{P}{p_1 p_2} = 5$, $\left| \frac{1}{\prod_{i=1}^{2} p_i} \right|_{M_2} = 1$.

Then $4E_2 = (0,0,4)$, $r(0,0,4) = 0$and from (6):

$$r_{4E_2} = 4 \cdot 0 + \left\lfloor \frac{4}{5} \cdot 1 \right\rfloor = 0.$$

We apply Statement 1 and Theorem 2 to find the rank of a number. We use formula (5), in which the multiplicity $a_i = p_i - x_i$.

Let's find the rank of a number $X = 15 = (1,0,0)$in RCS {2,3,5}.

The remainder modulo $p_1$is 1. Calculate the multiplicity $a_1 = p_1 - x_1 = 1$. So, let's take it $E_0 = (1,1,1)$with the rank $r_{E_0} = 1$.

At the same time, $X + E_0 = (0,1,1)$in which there was a transition modulo p_1.

Since the remainder modulo $p_2$the number $X + E_0$is equal to 1, then the multiplicity $a_2 = 3 - 1 = 2$and for $E_1 = (0,1,1)$c rank $r_{E_1} = 0$according to formula (6) the rank of the product is equal to

$$r_{2E_1} = 2 \cdot r_{E_1} + \left\lfloor \frac{2}{M_1} \cdot \left| \frac{1}{p_1} \right|_{M_1} \right\rfloor = 2 \cdot 0 + \left\lfloor \frac{2 \cdot 8}{15} \right\rfloor = 1.$$

Then $X + E_0 + 2E_1 = (0,1,1) + (0,2,2) = (0,0,3)$, while there was a transition modulo p_2.

Since the remainder modulo $p_3$number $X + E_0 + 2E_1$is 3, then the multiplicity $a_3 = 5 - 3 = 2$and for $E_2 = (0,0,1)$c rank $r_{E_2} = 0$according to formula (6) the rank of the product is equal to

$$r_{2E_2} = 2 \cdot r_{E_2} + \left\lfloor \frac{2}{M_2} \cdot \left| \frac{1}{p_1 \cdot p_2} \right|_{M_2} \right\rfloor = 2 \cdot 0 + \left\lfloor \frac{2 \cdot 1}{5} \right\rfloor = 0.$$

Then $X + E_0 + 2E_1 + 2E_2 = (0,0,3) + (0,0,2) = (0,0,0)$, while there was a transition modulo p_3.

By formula (5) we get

$$r(X) = |P_1^{-1}|_{p_1} + |P_2^{-1}|_{p_2} + |P_3^{-1}|_{p_3} - r(E_0) - r(2E_1) - r(2E_2) - 1 =$$
$$= 1 + 1 + 1 - 1 - 1 - 0 - 1 = 0,$$

which is consistent with previous calculations $r(15) = 0$.

The application of this approach makes it possible to replace the storage $\sum_{i=1}^{n} p_i - n$of values with the calculation of the multiplicity, which will significantly reduce the amount of equipment used for large modules.

For the case of hardware implementation E_i, the values r_i, , $\frac{1}{M_K} \cdot \left|\frac{1}{\prod_{i=1}^{k} p_i}\right|_{M_k}$ can be written to memory.

The rank of the number obtained as a result of arithmetic operations is the calculated rank of the number. In the case of correct operations, the calculated rank coincides with the true one, while if the range is overflowed, *P*these ranks will differ.

3. Modeling methods for calculating the rank

Rank calculation simulation was done on a MacBook Air with Apple chip M 1 and 16 GB of RAM using the Python programming language. As a measured indicator, the calculation time obtained using the time its library was chosen.

Modules were taken, the dynamic range of which exceeds 8, 16 and 32 bits. The modules of special form, 2^n, $2^n + 1$[10] have the greatest efficiency when working in the system of residual classes.$2^n - 1$

As methods for calculating the rank for modeling, the method based on the CRT, the approximate method based on the CRT, the method from the article [9], the proposed method was chosen.

The CTO-based method given by formula (2) requires division with a remainder by a large modulus P.

An approximate method based on CTO according to formula (3) with an estimate $N = \lceil \log_2 P\rho \rceil$requires operations with numbers of large capacity, however, the operation of finding the remainder by a large modulo is reduced to taking the least significant bits of the number.

The rank from [9] can be obtained from formula (5). At the same time, both to calculate the multiplicity $\alpha_i$and the product, $\alpha_i M_i$it is necessary to find the remainders modulo. The method considered in the article [11] based on the period and half-period of a number was taken as a method for finding the remainder. This method allows you to reduce the process of finding the remainder to the addition of numbers of a smaller dimension. All possible constants of this method were calculated in advance. For hardware implementation on integrated circuits, the authors recommend storing tables with ranks in memory, but this significantly increases the required area.

The proposed method based on Statement 1 makes it possible to avoid storing a large number of constants and finding residues by RCS modules, but the dimension of the operands increases.

Table 1. Simulation results

Module set	**Computation time, microseconds**			
	Chinese remainder theorem	**Approximate Chinese remainder theorem**	**Method from article [9]**	**Suggested method**
8 bit				
{5,7,8}	8 , 362	0.761	1 , 09 9	1 , 09 5
{5,9,16}	8,452 _ _	0.786	1.085	1.082
{7,15,16}	8.497	0.754	1,084 _	1.082
16 bit				
{5,7,9,17,32}	13.805	0.981	1.445	1.436
{15,17,31,32}	11.519	0.888	1.255	1.253
{31,63,64}	8.551	0.825	1.112	1.084
32 bits				
{3,7,17,31,65,127,128}	19.381	1.423	1.775	1,760
{9,17,31,65,127,128}	16,700	1.291	1.604	1.592
{1023, 2047, 4096}	8.713	0.929	1.089	1.084
64 bits				
{511,1023,1025,2047,4097,8192}	17.425	1.391	1.637	1.624
{2047,4097,8193,16385,32768}	14.857	1.236	1.461	1.447
{2097151,4194303,4194304}	9.541	0.974	1.091	1.089

From Table 1, it can be seen that with an increase in the dimension, the calculation time does not actually change, this is due to the dimensional grid of the processor, however, it becomes difficult to calculate the constants for these methods.

Conclusion

It can be seen that the approximate method based on the CRT has the highest efficiency, the method based on the CRT showed the worst time, while the proposed method and the method from article [9] have a similar calculation time. However, the simulation did not take into account the time of calculating the constants, which is also important when designing systems. So, for 128 bits, the calculations in the method from the article [9] turned out to be resource-intensive, and could not be completed in the allotted time.

In the case of implementing methods on a computer, the running time varies depending on the number of modules, but depends less on the size of the modules. At the same time, the proposed method has only a slight advantage, about 1%, in comparison with the method from [9].

Further studies of this problem can be directed to the implementation of methods using ASICs , which allow you to adapt circuits taking into account the size of the modules. At the same time, the proposed method requires significantly less equipment for storing constants, compared to the method from [9].

It is also possible to use the proposed method for determining the rank to determine the sign of a number.

ЛИТЕРАТУРА

1. Tchernykh A. et al. Cryptographic Primitives Optimization Based on the Concepts of the Residue Number System and Finite Ring Neural Network //Optimization and Learning: 4th International Conference, OLA 2021, Catania, Italy, June 21-23, 2021, Proceedings 4. Springer International Publishing, 2021. P. 241–253.

2. Chen S. et al. A Low Complexity and Long Period Digital Random Sequence Generator Based on Residue Number System and Permutation Polynomial // IEEE Transactions on Computers. 2022. T. 71. No. 11. P. 3008–3017.

3. Kaplun D. I. et al. Error correction of digital signal processing devices using non-positional modular codes //Automatic Control and Computer Sciences. 2017. T. 51. P. 167–173.

4. Gapochkin A. V. Using Redundant Modular Codes of the Residual Number System for Error Detection and Correction //Advances in Automation II: Proceedings of the International Russian Automation Conference, RusAutoConf2020, September 6-12, 2020, Sochi, Russia. Springer International Publishing, 2021. P. 653–663.

5. Bayoumi M. A., Jullien G. A., Miller W. C. A VLSI model for residue number system architectures // Integration. 1984. T. 2. No. 3. P. 191–211.

6. Акушский И. Я., Юдицкий Д. И. Машинная арифметика в остаточных классах. М: Сов. радио, 1968.

7. Chervyakov N. et al. AR-RRNS: Configurable reliable distributed data storage systems for Internet of Things to ensure security //Future Generation Computer Systems. 2019. T. 92. P. 1080–1092.

8. Chervyakov N. I. et al. Residue-to-binary conversion for general moduli sets based on approximate Chinese remainder theorem // International journal of computer mathematics. 2017. T. 94. No. 9. P. 1833–1849.

9. Исупов К. С., Завиялов А. А. Об эффективности нового алгоритма вычисления ранга в системе остаточных классов // Advanced Science. 2017. №. 4. С. 21–21.

10. Kuchukov V. et al. Performance Analysis of Hardware Implementations of Reverse Conversion from the Residue Number System //Applied Sciences. 2022. T. 12. No. 23. P. 12355.

11. Chervyakov N. I., Babenko M. G., Kuchukov V. A. Research of effective methods of conversion from positional notation to RNS on FPGA // 2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). IEEE, 2017. P. 277-281.

REFERENCES

1. Tchernykh A. et al. Cryptographic Primitives Optimization Based on the Concepts of the Residue Number System and Finite Ring Neural Network // Optimization and Learning: 4th International Conference, OLA 2021, Catania, Italy, June 21-23, 2021, Proceedings 4. Springer International Publishing, 2021. P. 241–253.

2. Chen S. et al. A Low Complexity and Long Period Digital Random Sequence Generator Based on Residue Number System and Permutation Polynomial // IEEE Transactions on Computers. 2022. Vol. 71. No. 11. P. 3008–3017.

3. Kaplun D. I. et al. Error correction of digital signal processing devices using non-positional modular codes // Automatic Control and Computer Sciences. 2017. T. 51. P. 167–173.

4. Gapochkin A. V. Using Redundant Modular Codes of the Residual Number System for Error Detection and Correction // Advances in Automation II: Proceedings of the International Russian Automation Conference, RusAutoConf2020, September 6-12, 2020, Sochi, Russia. Springer International Publishing, 2021. P. 653–663.

5. Bayoumi M. A., Jullien G. A., Miller W. C. A VLSI model for residue number system architectures // Integration. 1984. Vol. 2. No. 3. P. 191–211.

6. Akushskij I. YA., YUdickij D. I. Mashinnaya arifmetika v ostatochnyh klassah. – M.: Sov. radio, 1968.

7. Chervyakov N. et al. AR-RRNS: Configurable reliable distributed data storage systems for Internet of Things to ensure security // Future Generation Computer Systems. 2019. Vol. 92. P. 1080–1092.

8. Chervyakov N. I. et al. Residue-to-binary conversion for general moduli sets based on approximate Chinese remainder theorem //International journal of computer mathematics. 2017. Vol 94. No. 9. C. 1833–1849.

9. Isupov K. S., Zaviyalov A. A. Ob effektivnosti novogo algoritma vychisleniya ranga v sisteme ostatochnyh klassov //Advanced Science. 2017. No. 4. P. 21–21.

10. Kuchukov V. et al. Performance Analysis of Hardware Implementations of Reverse Conversion from the Residue Number System // Applied Sciences. 2022. Vol. 12. No. 23. P. 12355.

11. Chervyakov N. I., Babenko M. G., Kuchukov V. A. Research of effective methods of conversion from positional notation to RNS on FPGA // 2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). IEEE, 2017. P. 277–281.

ОБ АВТОРАХ / ABOUT THE AUTHORS

Кучуков Виктор Андреевич, младший научный сотрудник отдела теоретико-числовых систем Регионального научно-образовательного математического центра "Северо-Кавказский центр математических исследований" ФГАОУ ВО «Северо-Кавказский федеральный университет», 355000, Россия, г. Ставрополь, e-mail: vkuchukov@ncfu.ru

Kuchukov Viktor, Associate Researcher, Department of Number-Theoretical Systems, Regional Scientific and Educational Mathematical Centre “North Caucasus Centre for Mathematical Research” North-Caucasus Federal University, 355000, Russia, Stavropol, E-mail: vkuchukov@ncfu.ru

Бабенко Михаил Григорьевич, заведующий кафедрой вычислительной математики и кибернетики факультета математики и компьютерных наук имени профессора Н.И. Червякова ФГАОУ ВО «Северо-Кавказский федеральный университет», 355000, Россия, г. Ставрополь, E-mail: mgbabenko@ncfu.ru.

Babenko Mikhail, the head of the department of computational mathematics and cybernetics, Faculty of mathematics and computer science named after Professor N.I. Chervyakov, North-Caucasus Federal University, 355000, Russia, Stavropol, E-mail: mgbabenko@ncfu.ru

Кучеров Николай Николаевич, старший научный сотрудник учебно-научного центра «Вычислительной математики и параллельного программирования на суперЭВМ»

факультета математики и компьютерных наук имени профессора Н.И. Червякова ФГАОУ ВО «Северо-Кавказский федеральный университет», 355000, Россия, г. Ставрополь, e-mail: nkucherov@ncfu.ru

Kucherov Nikolay, Senior Researcher, Educational and Scientific Center "Computational Mathematics and Parallel Programming on Supercomputers", Faculty of mathematics and computer science named after Professor N. I. Chervyakov, North-Caucasus Federal University, 355000, Russia, Stavropol, E-mail: nkucherov@ncfu.ru

Дата поступления в редакцию: 19.04.2023
После рецензирования:13.05.2023
Дата принятия к публикации:07.06.2023